

[0001] (4 lines) Clinton.Thea 03/21/90 1122.2 mst Wed midi
Subject: Reason for this meeting
In response to the amount of information being presented on the topic in
cb, and to try and localize the discussions in a central place, this
meeting is convened to contain the discussions concerning the
development of midi standards for controlling theatre equipment.
---[0001]---

[0002] (14 lines) Clinton.Thea 03/21/90 1134.8 mst Wed midi
Subject: MIDI Forum intro
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2296] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 19:55 mst

The following 17 entries are copies of pertinent transactions in the
MIDI Manufacturers Forum in which we have been discussing the
development of Theatrical MIDI Message standards within the MIDI
specification. As you can see things have just really begun. We are
expecting to use Callboard as our mode of communication for this from
now on so I am putting these on as a matter of record and a way of
bringing everyone up to speed. All contributions welcome - the official
members of the MMA working group on Theatre Messages know who they are
and we will be talking here.
---[0002]---

[0003] (85 lines) Clinton.Thea 03/21/90 1135.2 mst Wed midi
Subject: Re: MMA 487
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2299] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:09 mst

Sorry about that! Had my settings wrong for uploading. Here is the
real MMA 487 message:

487 27-DEC 21:22 General Interest
New topic: Theatrical Messages
From: RICHMOND To: ALL

Greetings and happy new decade to everyone. This intends to be the
beginning of a thread to develop a whole new set of theatrical messages:
communication protocols between dedicated intelligent controllers in
theatrical applications such as memory lighting systems, computer
controlled theatre sound systems, moving light motion controllers, laser
control systems, computer controlled rigging systems, pyrotechnics
controllers, micro based intercoms, etc.

If you are interested in this please participate - I have been
soliciting the participation of numerous performing arts equipment
manufacturers and understand that quite a few of them are joining the
MMA specifically to become part of this discussion.

This subject was prompted by the previous TSBB item76 "Proposed MIDI
Lighting Standard" by Ken Ypparila and my subsequent response to it.

As a result of this spurt of activity we were asked to put together a committee under the new revised way of dealing with submissions to the MMA and now here we are (or at least hope to be!)

Now in the interest of pushing things one more step before leaving everything wide open with a what now? kind of reaction, here's something to get us started:

I would like to suggest that a general header be created under which all types of "Theatrical Messages" will reside and that it be in this format:

F0 7F <chan> <sub id 1> <sub id 2> <function> <data> F7

F0 7F = Real Time Universal System Exclusive Header

<chan> = Channel number

<sub id 1> = Theatrical message (number to be assigned)

<sub id 2> = Message format, i.e:

00 = Special

01 = Lighting

02 = Sound

03 = Moving instruments

04 = Rigging

05 = Laser

06 = Pyrotechnics

07 = Intercom

08 = Hydraulics

09 = Gas

etc.

<function> = Type of message, i.e:

00 = Special

01 = Load cue

02 = GO cue

03 = GO cue and Jam clock

04 = Advance to next cue

05 = Back to previous cue

06 = Open new cue directory

07 = Close cue path

08 = MIDI Time Code chase ON

09 = MIDI Time Code chase OFF

0A = Start Clock

0B = Stop Clock

0C = Zero Clock

0D = Zero All

0E = Jump to next sequence

etc.

<data> = Additional info as required

F7 = EOX

Well, that's it for starters - hope to hear something on this soon (anti-apathy message) or at least after the New Year's gigs are finished.

Cheers.

-Charlie Richmond

---[0003]---

[0004] (24 lines) Clinton.Thea 03/21/90 1135.4 mst Wed midi
Subject: MMA 488
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2300] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:12 mst

488 28-DEC 21:05 General Interest
RE: New topic: Theatrical Messages (Re: Msg 487)
From: RICHMOND To: RICHMOND

I just received the latest TSBB today - the mail is as slow as molasses here in the frozen North. Now that I am privy to the latest Unversal System Exclusive thinking, we have to revise my previous suggestion to start:

F0 7F <device ID> <sub ID 1> etc., where
<device ID> = ID of target device (not channel)
also, generally the device ID will equal 7FH, which is the
'broadcast' ID meaning 'intended for all devices'

Follow the fascinating story of "Device Number" (Device ID) in the continuing saga of Item 79 in the current TSBB...

-Charlie Richmond
---[0004]---

[0005] (20 lines) Clinton.Thea 03/21/90 1135.6 mst Wed midi
Subject: MMA 490
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2301] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:13 mst

490 2-JAN 12:49 General Interest
RE: New topic: Theatrical Messages (Re: Msg 487)
From: CYBERPUNK To: RICHMOND

Re: Theatrical messages:

Starting off admitting that I don't know much about these applications plus I haven't had time thoroughly muse over your proposal, is there any way (or does it make any sense) to integrate this with the MTC Set-Up messages, which seem to already (try to) cover the music automation side of this, and have plenty of room left for extensions? (The "multimedia" people might also appreciate having these treated as equals).

- Chris
---[0005]---

[0006] (40 lines) Clinton.Thea 03/21/90 1135.8 mst Wed midi
Subject: MMA 491
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2302] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:14 mst

491 2-JAN 21:44 General Interest

RE: New topic: Theatrical Messages (Re: Msg 490)
From: RICHMOND To: CYBERPUNK

Good comment. I actually thought at one point that this should be where the messages reside. However, a couple of difficulties came to light: the first probably more philosophical, the second definitely practical.

1. Our messages are required on a "Real-time" basis - that is they are performance oriented and must be sent and received without any delay. The receiver must also know to act on them immediately, therefore assigning them to the category of "Non-Real Time..." seems definitely like a step in the wrong direction. Again, don't know if this really makes any difference in the reality of how these messages are dealt with, but if eventually a MIDI network evolves which tries to manage thousands of messages simultaneously I can well imagine that "Non-real time.." messages will be the first to be put on the back burner! Probably safer to start out the way we want to go..

2. I anticipate very few of our messages will require referencing themselves to a time code location, so to tie ourselves into a format which (seems to?) (requires?) 7 bytes of superfluous information seems like costly overhead.

But, who knows, maybe someone else has a more compelling argument in the other direction. Hope to get some more feedback on this before dismissing it entirely. Perhaps we can put some of our more appropriate (i.e. "Non-real time, Set-up" type) messages into this category so they can be dealt with appropriately and not contribute to the overhead of the real-time stuff.

Anyway, let's keep this in mind and thanks.

Charlie
---[0006]---

[0007] (28 lines) Clinton.Thea 03/21/90 1135.9 mst Wed midi

Subject: MMA 492

To: {forum >udd>Thea>Clinton>meetings>midi}

Forum-Transaction: [2303] in the >udd>USITT>KDHewitt>cb meeting

Transaction-Entered-By: CRichmond.USITT

Transaction-Entered-Date: 20 March 1990 20:14 mst

492 3-JAN 12:39 General Interest

RE: New topic: Theatrical Messages (Re: Msg 491)
From: CYBERPUNK To: RICHMOND

Charlie -

Ah; I was thrown off track by skimming your proposal and seeing mention of timecode - and therefore thought everything was pre-meditated. Of course, MTC Set-Up as it now sits does not lend itself towards real-time response.

On the other hand, I've kind of lamented that there wasn't a real-time

response provision in Set-Up messages. So, what do you (and everybody) think about duplicating the current MTC Set-Up messages in a set of real-time SysEx codes, with the main difference being the timecode portion is dropped (time=now)? And then rolling the theatrical messages into both the pre-meditated (non-real-time) and real-time versions of these messages? (Mind you, this is still off the top of my head and without studying your application closer - or the thread on the same subject going on over in the public Synth forum...)

- Chris
---[0007]---

[0008] (14 lines) Clinton.Thea 03/21/90 1136.1 mst Wed midi
Subject: MMA 493
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2304] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:14 mst

493 3-JAN 21:47 General Interest
RE: New topic: Theatrical Messages (Re: Msg 492)
From: RICHMOND To: CYBERPUNK

Yes, I expect we will basically come around to that and more -good idea. For a more complete idea (but not totally!) check out my public Synth forum comment. I'll look more closely at the MTC Set-Up format to see how easily it can be adapted. Thanks.
---[0008]---

[0009] (36 lines) Clinton.Thea 03/21/90 1136.2 mst Wed midi
Subject: MMA 497
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2305] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:15 mst

497 30-JAN 19:05 General Interest
RE: MMA Meeting (Re: Msg 496)
From: RICHMOND To: IMA (NR)

Some comments on this report:

1. My understanding (per TSBB13) is that Working Group 6) is properly entitled Lighting Control/Theatre Cue Proposals - if not, could you please clarify the current thinking on this subject? Are we going to separate lighting control from Theatre Cues or make it a subset? Please note that in Message 487 I amended my original theatre cue proposal to suggest that all technical theatre elements including lighting might be integrated into this global Real-time Universal Sysex header called Theatrical Messages.

2. Also, I have received confirmation from Jim Cooper that he wishes to be the official representative of the Technical Board in this Working Group. I also want to make sure you have myself (Charlie Richmond) included.

3. Re: my Theatre Control proposal from TSBB13. My intention as outlined above was to have this treated as a proposed amendment to the original Lighting Cue proposal rather than an independent proposal. Also, as explained in Message 487, that message really contains the most recent update of this proposal, effectively superceding the hard copy referred to in the meeting summary. I would prefer it if all members were to refer to Message 487 rather than the hard copy for current thinking on this subject.

Thanks very much, Charlie Richmond
---[0009]---

[0010] (21 lines) Clinton.Thea 03/21/90 1136.4 mst Wed midi
Subject: MMA 498
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2306] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:15 mst

498 31-JAN 23:54 General Interest
RE: MMA Meeting (Re: Msg 497)
From: JRONA To: RICHMOND

Charlie, p`~ My one question about the theatrical control proposal is this: While it opens up the possibility for sophisticated software for control with inexpensive hardware via MIDI, it doesn't address the needs that some people seem to need in being able to sequence controls from some kind of fader or trigger controller into a standard sequencer (possibly in combination with music tracks). The reason I say this is because not all sequencers will record Sys Ex in real time. Generally the proposal is very good and should help to set a needed standard, but does it cover all the bases?

jr
---[0010]---

[0011] (46 lines) Clinton.Thea 03/21/90 1136.6 mst Wed midi
Subject: MMA 499
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2307] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:16 mst

FORUM>Reply, Add, Read, "?" or Exit> 499 499 1-FEB 12:43 General Interest
RE: MMA Meeting (Re: Msg 498)
From: CYBERPUNK To: JRONA

There seems to have been a miscommunication to sequencer manufacturers about the Universal SysEx area and its purpose; perhaps we can clarify it now. It is understood that recording unknown SysEx into a sequencer is a pain - how long is the message, does it bump other messages around, how do you display it, how do you allow the user to edit it in a meaningful way, etc. However, there are some extensions to the basic MIDI spec that may convey useful performance information that can't be stuffed inside controller messages. So, the Universal Real-Time area

was created to make life on sequencers easier in two ways: One, the ID - 7F - is easily recognizable and tested for (and all others can be ignored), and two, since all accepted Universal Real Time SysEx messages are approved, publishes, and fixed in format (and there's not very many of them), it should now be possible for sequencers to know in advance how long these messages are, how to parse them, and what they mean so that they can be presented to the user as unique icon with special, meaningful editing windows without forcing the user to dive into hex programming. In other words, sequencers should treat approved Universal Real Time SysEx messages like 'known' real-time channel messages. Therefore, a sequencer could record (and display, and allow editing of) these special messages without worrying about the morass of dealing with all other generic, unknown SysEx messages.

So, how about it, sequencer-guys (and gals)? How about spotting, recognizing, and recording the FO 7F messages? The more comprehensive Bank Select message died because

-[missing text]-

-[missing text]-

the subsequent message. In the meantime your comments are very well taken, fit in precisely with our needs and really address the matter correctly in a more general context. Stand by for previously composed response.....

---[0012]---

[0013] (81 lines) Clinton.Thea 03/21/90 1137.0 mst Wed midi

Subject: MMA 502

To: {forum >udd>Thea>Clinton>meetings>midi}

Forum-Transaction: [2309] in the >udd>USITT>KDHewitt>cb meeting

Transaction-Entered-By: CRichmond.USITT

Transaction-Entered-Date: 20 March 1990 20:17 mst

502 1-FEB 21:12 General Interest

RE: MMA Meeting (Re: Msg 498)

From: RICHMOND To: JRONA

Jeff, I understand what you are saying. This aspect of the universal real time sysex extension is one which I think is going to have to be addressed by both sequencer developers as well as manufacturers of hardware controllers. What has evolved is a new set of real time messages which must be treated in most respects just like the "standard" musical MIDI messages we are all familiar with. A partial treatment of this is contained in messages 490 through 493 if you want to review these.

At this point, it is true that sequencers often do not handle sysex messages in an appropriate manner but as universal real time messages continue to be defined, sequencers should ultimately deal with them. I feel that specialized sequencers will be used for handling these more complex requirements and, in fact, that is exactly what we have developed. We are anxiously wanting to get on with the finalization of at least the first version of this theatrical message specification so we can include it in our sysex message editor in a librarian menu format. At the moment, users can use the editor to create any kind of sysex message but they must type in the hex data directly.

Likewise, as real time messages become more commonly implemented,

hardware controllers will have to accommodate the more popular messages - ones that users will need to trigger events in real time from physical handles. As you may have guessed, in the absence of such standardized messages, manufacturers such as ourselves who make MIDI controlled theatrical systems have had to use note/controller/song select/position pointer/etc. messages in order to take advantage of the features of standard controllers and sequencers. Of course, the disadvantages to this are that one has to be very careful in the channelization of the system and also the translation of musical MIDI commands to theatrical functions is not very intuitive.

Since MIDI is gaining such wide acceptance (and its protocols are uniquely applicable to the application) within the performing arts community as a "performance controller" we will have to seriously address the need to accommodate live performance requirements in other than strictly musical parameters or else another standard performance communication protocol must and will be developed to fill the need. I do not think this would be healthy for the industry but neither is it likely. However, we must address the problem of the time it takes to get these standards approved: we have applications which need to use these messages and something is going to have to be done soon. We were hoping to get some consensus initially within a couple of months and to get the sub id1 assigned so that we could start using legal messages to do the functions we need to do right away.

This brings me to a question that I had wanted to ask a while ago, but decided to wait (until now, I guess): would it be possible to have the IMA assign the sub id1 code and allow the working group to start developing a set of theatrical messages which may be used by mutual agreement until they are officially approved. I realize this goes against the general philosophy as currently espoused, except that it would be very similar to assigning a manufacturer id and then allowing the manufacturer to use any protocol they wish without prior approval, as long as they publish it.

There is a possibility that what could be done is to obtain a manufacturer id and do our development under that guise. The two basic problems with this are:

1. Political - attempting to create an industry standard protocol under a specific manufacturer's id number simply does not wash.
2. Sysex data sent as a specific manufacturer's message would never be treated by a sequencer or intelligent system node or network as it should, i.e. a universal real time system exclusive message!

At this point I will conclude for now - Thanks also for Cyberpunk's previous comments on this subject.

Thanks Charlie Richmond
---[0013]---

[0014] (16 lines) Clinton.Thea 03/21/90 1137.1 mst Wed midi
Subject: MMA 503
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2310] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:17 mst

503 1-FEB 23:11 General Interest

RE: MMA Meeting (Re: Msg 497)

From: MARKATOPCODE To: RICHMOND

In regards to your first question, yes, I agree that working group 6 should deal with the entire issue of theatrical queues, not just lighting queues. I realize that this might become a rats nest, but I think that if we could solve both of these with one stone, the better.

-mark

---[0014]---

[0015] (32 lines) Clinton.Thea 03/21/90 1137.3 mst Wed midi

Subject: MMA 515

To: {forum >udd>Thea>Clinton>meetings>midi}

Forum-Transaction: [2311] in the >udd>USITT>KDHewitt>cb meeting

Transaction-Entered-By: CRichmond.USITT

Transaction-Entered-Date: 20 March 1990 20:17 mst

515 2-FEB 12:54 General Interest

RE: MMA Meeting (Re: Msg 502)

From: CYBERPUNK To: RICHMOND

Right. Hmm... I can see your bind, but we must stay 'politically correct.' in theory, several members of the MMA boards are going to Japan in Feb or March to attend a JMSC meeting, and try to properly present (and get approved) several burning and/or overdue topics (such as machine control). If you can get a 'Level 1' proposal together and approved by a few companies in short order, we could try to place it on that agenda. I don't see the JMSC going for temporary approval of a Universal Sub-ID for experimentation, since they are supposed to be in the loop for approving the final use of things (and have not been quick to do so lately). In general, there has been some resistance from the JMSC in approving anything that isn't obviously 'music' in a note on/mod wheel/pitch bend sort of mode (care to update us on that, RolandMMA?). So, another idea that comes to mind is trying to set aside some more 'universal' IDs (such as 7C, 7B, etc.) for non-note-on things like machine control, theatrical, mixing, lighting, etc., so the JMSC can not be involved in the loop or worry about these non-musical things confusing the original intent of MIDI and the previously approved spaces, if they so desire. Ideas, folks? (P.S. - Didn't mean to rake MarkatOpcode over the coals - for those who don't know him, he IS a very enthused, open-minded, rather smart 'dude.' He'll be an asset -even if he does rip off my work verbatim and throws it back up on PAN <grin>.)

- CM

---[0015]---

[0016] (42 lines) Clinton.Thea 03/21/90 1137.4 mst Wed midi

Subject: MMA 517

To: {forum >udd>Thea>Clinton>meetings>midi}

Forum-Transaction: [2312] in the >udd>USITT>KDHewitt>cb meeting

Transaction-Entered-By: CRichmond.USITT

Transaction-Entered-Date: 20 March 1990 20:18 mst

517 2-FEB 22:17 General Interest

RE: MMA Meeting (Re: Msg 515)

From: RICHMOND To: CYBERPUNK

I guess the main problem I have in appraising your suggestion is that I don't know all the fine points of this approval process so I can only say that anything that will help get this underway with a minimum of delay would be great.

The only thing that seems like it might be a problem is the creation of a 'universal' id that will not be handled properly by sequencers or hardware. Music 'networks' make arbitrarily decide that since our new id is not music related they won't deal with it and that seems like a step backward since we want to be able to put this stuff out all over the MIDI interface throughout the theatre environment.

In other words, not handling the new id on a musical sequencer would defeat the whole purpose of creating these new standards in the first place. Of course, the more powerful software and hardware could advertise that it handles 'all the new stuff' correctly and there would be pressure to upgrade. The other thing is that we would want to make sure that any 'new' theatrical (especially) id is designated a 'real time' message and will never be relegated to the back of the queue.

I guess it just seems that real time universal sysex is the slot that this seems to belong in - perhaps you could enlighten us as to the reason that this area poses particular problems in gaining JMISC approval. Maybe you could also enlighten us about what other areas (new universal id's?) have less red tape involved -what rules apply to what types of messages require whose approval, etc. This would help us newcomers. A sort of MIDI welcome wagon service....

Well, chew on this while I'm out of town (till Thurs) and I'll come back to confuse things some more!

Thanks

---[0016]---

[0017] (67 lines) Clinton.Thea 03/21/90 1137.8 mst Wed midi

Subject: MMA 563

To: {forum >udd>Thea>Clinton>meetings>midi}

Forum-Transaction: [2313] in the >udd>USITT>KDHewitt>cb meeting

Transaction-Entered-By: CRichmond.USITT

Transaction-Entered-Date: 20 March 1990 20:18 mst

563 12-MAR 17:41 General Interest

Working Groups

From: MARKATOPCODE To: ALL

Working Groups -----

Here are my overly wordy ideas for working groups. Before you all jump all over me (hmmmm) for this appearing overly formal, I don't mean for this to be carved in stone or immutable. I am merely trying to get some guidelines down on paper (er, um, electrons?) so that we can keep going.

Guidelines: - A working group is set up when a proposal has been made by a member of the MMA and there is sufficient interest. This proposal should be submitted to the chairman of the TSB or sent to Lucky at the MMA office.

- The goal is to have, by the next MMA meeting, a polished proposal that can go to vote.

- Each working group should consist of at least three people: a working group chair person (presumably, although not necessarily the member who brought it up), a member of the TSB, and at least one other interested member. If a proposal is deemed important, but cannot get a third member interested, the first two shall be sufficient.

- There is a member of the TSB who is the working group co-ordinator. They are there to facilitate and track the actions of the working groups.

- Each working group shall communicate amongst itself at least once a month. It is the responsibility of the chairman of each working group to ensure that at least some progress gets made each month. Progress can be technical, or even just agreements on meeting times and places, anything (please). A copy also needs to go to the TSB working group co-ordinator.

Action Items: - We need a member of the TSB to be the working group co-ordinator. Volunteer, or I'll be calling...

- Each working group needs to have a member of the TSB on it. This is the status as I know it (see message 496 for the complete membership and chairpersons of these working groups):

- 1) General MIDI - Rob Lewis
- 2) Leveles of MIDI - Mark Lentczner
- 3) MIDI Testing - Alex Limberis
- 4) Implementation Charts - Alex L.
- 5) MicroTonality - Hal Chamberlin & Mark L.
- 6) Lighting Control - *** no TSB membr ***
- 7) MIDI Machine Control - Jim Cooper & Chris Meyer
- 8) User Bits - Gerry Lester & Chris M. 9) Timbre/Release Time - *** no TSB member ***
- 10) Bar Marker - lots'o'TSB members
- 11) Channel Change - Chris M. & Mark L.
- 12) Legatto Footswitch - Chris M.
- 13) Generic Voice Editing - Chris M.
- 14) Controller Etiquette - Mark L. & Chris M.
- 15) Bank Select - Alex L.

-mark

---[0017]---

[0018] (23 lines) Clinton.Thea 03/21/90 1137.9 mst Wed midi

Subject: MMA 564

To: {forum >udd>Thea>Clinton>meetings>midi}

Forum-Transaction: [2314] in the >udd>USITT>KDHewitt>cb meeting

Transaction-Entered-By: CRichmond.USITT

Transaction-Entered-Date: 20 March 1990 20:19 mst

564 12-MAR 17:43 General Interest

Numbers
during development
From: MARKATOPCODE To: RICHMOND

EOIf I understand what you want to do with an assigned ID, there is already a number for you. I believe that what you want is a number so that you can create software that operates, but the proposal is not yet approved. Use Sys Ex ID of \$7D. This is reserved for "non-commercial use and is not to be used on any product released to the public" (from the spec.) Just make the rest of the message look exactly as if the ID were \$7E or \$7F as needed with made up sub-ids. Now, it is true, \$7D does not distinguish between real-time and not, but for your purposes, just use the sub-ids different. As none of the software you write in this phase will ever see the public (right...) you can change all the numbers when the final proposal is approved.

-mark l.
---[0018]---

[0019] (36 lines) Clinton.Thea 03/21/90 1138.0 mst Wed midi
Subject: MMA 568
To: {forum >udd>Thea>Clinton>meetings>midi}
Forum-Transaction: [2315] in the >udd>USITT>KDHewitt>cb meeting
Transaction-Entered-By: CRichmond.USITT
Transaction-Entered-Date: 20 March 1990 20:19 mst

568 13-MAR 20:12 General Interest

RE: Working Groups (Re: Msg 563)
From: RICHMOND To: MARKATOPCODE (NR)

Thanks for the concise guidelines. With reference to 6) Lighting Control: I am trying to get this recognized as "Theatrical Messages" within which we will define a lighting category. I believe we have some general consensus on this so we are trying to stop calling it just 'Lighting'. This is going to be a very comprehensive project and it is slow getting started however Jeff has sent me a copy of the membership list as of the first of March and about 6 new members have recently joined who have been specifically solicited by me to join in this working group. So as soon as they get on PAN and start contributing to the forum things should begin in earnest.

Also, I spoke to Jim Cooper some time ago with the understanding that he has agreed to be our official member of the TSB. I haven't spoken with him recently so don't know if he has since become uninvolved. I suppose I should send him my monthly report but don't really have a great deal to report at this point nor am I sure he is the one to report to. We have no objection to having any other member be on this working group (ahem). My impression from communication with other theatre equipment manufacturers is that since they are basically unfamiliar with MIDI, they are trying to come up to speed with its current implementation before plunging in and developing!

Anyway, thanks for your patience and please let me know if we should be reporting to anyone besides Jim Cooper.

-Charlie

---[0019]--- (nref = [0020])

[0020] (36 lines) JLBr

-[missing text]-

2. the unit's equipment subtype were specified.
3. the particular unit's device ID were called

Upon validating the three previous conditions, the unit would read the command information. The execution of the command would be carried out in the particular unit's own internal "language", except that each device would be required to understand how to process the <function> cues other than GO as sent by the MIDI SysEx call.

In looking at this while writing, the first question that comes to mind is "Why is the <sub id 2> message format byte necessary?" If each device has an ID (and I'm assuming that it must), then why not proceed directly to the <function> byte? This I am asking for information only, since I'm also sure that there's a good reason that I just don't understand yet.

Personally, I'm enthusiastic about the possibilities that this MIDI extension offers. As a designer, I think this would make all kinds of interesting things possible. I frequently use music in production, sometimes off of a live keyboard. Further, there are an increasing number of processing devices that incorporate MIDI to enable them to work with music machines. Add to that the various editor and sequencer programs available for personal computers, plus a control system like Command Cue, and the limits are almost out of sight. This is exciting!
---[0020]--- (pref = [0019], nref = [0021])

[0021] (40 lines) CRichmond.USITT 03/22/90 1941.0 mst Thu midi
Subject: Re: MMA 568

As usual, John, you have described the situation in a nutshell better than I could. To answer your question, there are two basic reasons:

1. The command set for controllers in various media (lights, sound, pyro, rigging, etc.) is likely to be very similar but also have many differences. The idea was that if we had a single command set that included absolutely every command that every type of theatrical controller could have, we would have an extremely large list and one which would not apply for the most part to the system of immediate concern. This way, we can classify at least which media's command set the programmer can refer to to program responses in that area without making them look through a 10 page list. Also, I hoped we could keep the set of commands for each type of controller to fewer than 128 (the number we have available in a single function byte). The total number of commands for everything is (I suspect) larger than that, necessitating using two function bytes and negating the advantage of dropping the id2 byte.
2. I have been involved in productions which have a large number of intelligent controllers and can easily foresee having more than 128 devices to network on this system. We haven't used 128 yet, but in these venues if you tell them they can do it, they'll want 20 more. True 99% of the users will only use a few, but this thing will really get going once the standard is published and I could easily see the possibility that we may want to talk with more than 128 lighting consoles alone. My concern runs in this direction- that we should

perhaps provide for two bytes of device id, but of course we can get around this to a certain extent by creating a second message format number for the additional lighting consoles. (i.e. in the proposal, sub id 2 is 01 for "lighting" - we have 128 total "message format" numbers and will not define them all right away. We can make a sub id2 number to signify "lighting group 2" if it becomes necessary. Or there are other possibilities, of course. If

Thanks greatly for your contribution - I hope this gets everyone going and look forward to more.

Thanks also to Tim Clinton for setting up this forum without being asked - it will assist all of us greatly in coming up with this important new proposal for a MIDI standard!

---[0021]--- (pref = [0020], nref = [0022])

[0022] (13 lines) JLBracewell.USITT 03/23/90 0609.0 mst Fri midi
Subject: Re: MMA 568

Charlie, before some of the big guns log onto this discussion, why don't you give the rest of us a run-down on what you see as the outside limits on size of the network that might be involved. For example, I had thought of this only in terms of single-event staging. Granted, in a rock show that could involve a lot of color changers and motorized lights; but the number of physical entities to be addressed would still be relatively small. After reading the previous entry I realized that you are also thinking in theme park terms that might involve a great many "stages" plus the intervening "parade ways." Maybe even more than that. Some of the rest of us might be able to contribute a bit more if we expand our horizons beyond physical theatre walls. You've had the experience of having to think in large multi-stage terms. Why don't you write to what you see as potential extents?

---[0022]--- (pref = [0021])

[0023] (54 lines) CRichmond.USITT 03/23/90 1129.0 mst Fri midi
Subject: MIDI Control

Date: Friday, 23 March 1990 08:27 mst

From: rodney at HUNT

SAI.BOEING.COM (Rodney Daughtrey)

To: CRichmond at UNCAMULT

EOPRedistributed-Date: 23 March 1990 11:29 mst

Redistributed-From: CRichmond

Redistributed-To: {forum >udd>USITT>CRichmond>meetings>midi}

Date: Fri, 23 Mar 90 01:30:00 EST

From: Tim Clinton <Clinton%UNCAMULT.bitnet@ugw.utcs.utoronto.ca>

Subject: Midi control

To: huntsai!rodney@BEAVER.CS.WASHINGTON.EDU.ARPA NET

Hi. I'm just basically trying out mail paths to get back to you for someone else who is very interested in midi control. Actually, he is part of a committee looking into standards for the whole question. I

\cf

this gets through to you, you can correspond with him at "CRichmond@UNCAMULT.BITNET". Otherwise, just reply to me, and I'll forward to him with instructions on how to get through to you.

Ain't networks bizarre?

Hello! I got your mail address from Tim Richmond, per the message above.

I'm very interested in using MIDI for stage lighting applications, as you are probably aware. Some information about myself might be useful.

I don't have any real experience in stage lighting; I'm a computer scientist working in the field of artificial intelligence, and in my spare time, a musician. I am a drummer, and have an electronic drumset which has MIDI capabilities (a Simmons SDS9), which I have used to do the usual MIDI-type things with other musical instruments (changing programs on a keyboard connected up to the SDS 9, playing the voices of another synth, other synths playing my voices, etc). I often jam with a keyboard player friend of mine, and have lots of interesting ideas about what might be possible using MIDI to control visual parameters of a musical performance, such as changing stage lighting parameters (brightness, color, position in 3D-space with servo motors, etc).

If I can help in any way, I would be more than happy to...I would especially like to hear more about the committee you are on to develop standards for stage lighting. I have some ideas about that also...

Talk to you soon!

-- Rodney

Rodney Daughtrey E-mail: rodney@huntsai.boeing.com
Huntsville AI Center ...!uw-beaver!bcsaic!huntsai!rodney
Boeing Computer Services Voice: (205)-461-2352 Fax: (205)-461-2933
---[0023]--- (nref = [0024])

[0024] (6 lines) CRichmond.USITT 03/23/90 1136.5 mst Fri midi
Subject: Re: MIDI Control
I received this message and thought it presented some interesting ideas for us to consider even though it wasn't specifically pertinent to developing this protocol. We should probably try to make the standard allow for these types of advanced concepts if at all possible. I have responded to Rodney and suggested that he get on Callboard to fully participate in this forum.
---[0024]--- (pref = [0023])

[0025] (60 lines) BRodriguez.USITT 03/25/90 1549.1 mst Sun midi
Subject: ASCII Light Cues and MIDI
Here are some relevant excerpts from messages that Charlie Richmond and I have exchanged, on the subject of ASCII Light Cues and MIDI.

CRichmond 6 Mar 1990 -----

Yes, you have realized exactly what I was basically getting at with the MIDI file dump proposal. In essence we are talking about a proposal to standardize the physical layer and communication format protocol (within the network model) only. The application layer (the content) and the operating system of the host environment...can and will be completely independent of the MIDI aspects of the standard. All I'm saying

is that there should be a standard procedure for transferring these cues between machines and that it should follow the MIDI file dump format. Obviously, any cue data saved as simply an ASCII file can be fired out any serial port or modem, etc. or even converted with the appropriate utility to a MIDI file dump data stream and there's nothing we can do to stop people from doing this.

BRodriguez 10 Mar 1990 -----

I just received a query from Jim Fackert at CAE on the subject of using MIDI Files for the ASCII Cues project. Let me quote part of my reply: "the thrust of our group [the ASCII Light Cues group] is to define a 'medium-independent' format, which can be transmitted through either a disk file or a serial port (to name two examples). This may or may not be compatible with the MIDI proposal."

What you're telling me now is that they ARE compatible -- that the MIDI Files proposal defines only the ISO "physical", "data link", and "network" layers. (I'm a bit fuzzy on the ISO model, but then I think ISO themselves aren't sure what they mean.) The ASCII Files proposal is definitely not concerned with physical transfer -- I guess we're up at the ISO "application" and "presentation" layers.

So, great! I'll be happy to talk about how MIDI can be used to transfer ASCII Cues. However, I don't think we should say MIDI is THE standard procedure for transferring ASCII Cues between machines. Better that we should say machine X can produce USITT standard ASCII Cue Files, and machine X can transfer said files using USITT standard MIDI lighting protocols. (This leaves the door open for manufacturers of "gateway" products.)

CRichmond 12 Mar 1990 -----

I understand your concern about saying MIDI is the standard for transfer of ASCII Cues. I meant only that the MIDI standard for transfer of ASCII Cues must be defined by us, the Theatre Messages working group, whether its to simply define how the ASCII data is to be mapped into the MIDI file transfer format or whether we come up with a whole new file transfer format within the Theatre Message header. I do agree with you that this is more a non- real time function and that the non-real time function of the standard MIDI file transfer could probably be used without detriment BUT I did not want to presume to know enough about ASCII Cues and their use to make this judgment myself. Thank you for the additional information: it does indeed look this way.

BRodriguez ?? Mar 1990 -----

O.k., let's talk about mapping the ASCII Cues data stream into the MIDI transfer protocol. We're agreed then -- it's not a real time function.

---[0025]---

[0026] (35 lines) CRichmond.USITT 03/28/90 1928.6 mst Wed midi
Subject: LIEP Protocol

I just received a copy of Draft V4.5 of the "LIEP - Lighting Information Exchange Protocol" and have had an initial glance at it to see whether or not it might provide a good model for us to start with in this MIDI message protocol.

It seems to me as though there are a number of potential difficulties here:

The LIEP protocol makes a number of assumptions which will not necessarily be valid in the MIDI implementation.

First, it assumes there are two (only) units communicating - a "control Product" and an "external Device" - and it defines the mode of communication (19.2kb/8b/0p) and describes a very closely integrated set of messages and required responses to requests, the lack of which could hang the system and cause serious operational difficulties. Basically, it is very heavily dependent on a standard one-to-one "equipment/terminal" relationship over the RS-232 style null-modem or standard modem connection. This involves close attention to ACK's and error checking and really does not fit in with MIDI in its current approach to creating a performance system with a potentially open-loop configuration.

Secondly, although the lighting industry may feel this is adequate, the use convention used for cue numbering is quite rigid, with only a single decimal point place allowed (and automatically inserted) plus very specific assignment of cue numbers and ranges for specific purposes. I feel that we need to adopt a more open-end approach to cue numbering and will enter my proposal on this shortly.

In many other respects, however, the range and depth of the types of information to be transmitted or exchanged is quite wide in this document and may very well provide us with a good basis for ensuring that our project encompasses as much of this type of information exchange as possible within the performance style of networking.
---[0026]--- (nref = [0027])

[0027] (40 lines) CRichmond.USITT 03/29/90 0142.6 mst Thu midi
Subject: Re: LIEP Protocol

Well, if you understood what I was rambling on about in that last message, I'm sure you'll have no trouble with this one. As promised, this is my proposal for cue numbering. I originally sent this in hardcopy form to the MMA in response to the previously mentioned proposal from Ken Ypparila which was basically derived from Andy Meldrum's paper of a year ago which started the ball rolling in the MIDI direction in the first place. I spoke to Andy today and he will hopefully be joining us here soon and will be able to put this paper online for our reference.

Anyway, here is the idea I have for numbering cues in the <data> portion of the proposed Theatrical Message protocol:

Whenever the <function> byte makes reference to a cue in which the following data byte logically should contain the cue number, the cue number should be expressed as an open-ended length string with the high digit sent first (MSB) and then followed directly by F7H, which is the system exclusive end- of-message byte.

Several features are contained in this cue number string - digits are represented by 00H (0) to 09H (9); the placement of decimal points is specified by 0AH; 0BH indicates the beginning of another number specifying to which concurrent cue list the cue is assigned.

For example, to call cue 786.63.5 in cue list 76.2 on device number 1, for lighting would be F0 7F 01 xx 01 01 07 08 06 0A 06 03 0A 05 0B 07 06 0A 02 F7. Specifically, F0 means System Exclusive, 7F Real Time Universal, 01 Device Number 1 (or 2 if you prefer), xx is the yet-to-be-assigned A Theatrical Message id, 01 Lighting, 01 Load Cue Number, then the rest of the data represents the cue and cue list number as above.

I'm sure this could be improved and/or elaborated upon - the question is really how far to we want this to go? We will need at least this amount of detail to call up cues within our own theatre sound system controller, so I am
-[missing text]-

Again, I have to credit Andy Meldrum for the original impetus for this method of expressing cues - ideas?

---[0027]--- (pref = [0026], nref = [0028])
[0028] (5 lines) JNewstrom.USITT 03/30/90 1029.1 mst Fri midi
Subject: Re: LIEP Protocol

I still haven't read all the messages on this board yet, but is the LIEP protocol transmitted over a MIDI line or an RS-232?

John Huntington
---[0028]--- (pref = [0027], nref = [0029])

[0029] (7 lines) CRichmond.USITT 03/30/90 2351.1 mst Fri midi
Subject: Re: LIEP Protocol
The LIEP proposal was passed on to me by Brad Rodriguez and I am not certain of its origin but it seems to be dormant at the moment. Perhaps Brad could provide further information about its history.

At any rate, it seems to be transmitted over RS-232 (or possibly RS-485 or 422) - I didn't see specific reference to any standard but it read more like 232 than anything else. Definitely not MIDI. .A .]
---[0029]--- (pref = [0028], nref = [0030])

[0030] (2 lines) JNewstrom.USITT 04/01/90 1332.5 mst Sun midi
Subject: Re: LIEP Protocol
I guess that one of the RS-XXX interfaces would be better for the longer allowed transmission distance. John Huntington
---[0030]--- (pref = [0029], nref = [0031])

[0031] (45 lines) CRichmond.USITT 04/02/90 2326.5 mst Mon midi
Subject: Re: LIEP Protocol
It is correct that the allowed transmission distances of RS-422 and RS-485 are longer than MIDI. BUT there are tradeoffs: transmission of 422 and 485 is a compromise between data rate and transmission distance, with the allowable rate going down as the distance goes up. Even though MIDI is technically restricted to 50 feet, the real reason for this is cost. The idea was that very low cost devices could be produced which transmitted the minimum distance and higher cost transmitting devices could be used to send longer distances. Since the receiver is not a factor and the entire transmission system is a very simple one-to-one opto-isolated current loop with a reverse polarity diode to snub reflections, it is technically possible to transmit MIDI up to several

kilometres with just an enhanced transmitter circuit. In fact, most standard MIDI outputs are capable of this performance now simply because the technology is there.

A couple more points: Because every MIDI input is completely opto-isolated, there is no potential for ground loops (AC or DC) and the common-mode voltage range of the system is couple of thousand volts. This cannot be said for the RS-422 or 485 standard. Also, because the data rate and start/stop bits and basic protocol is precisely defined with no exceptions, every MIDI output can talk to every MIDI input without any doubt. These days, the ability to plug two things together right out of the box and have them talking to each other immediately is refreshing to say the least!

A side note on speed and distance: A company called Lone Wolf has developed a LAN for MIDI that provides up to 65000 channels as an extension of the MIDI protocol and allows the LAN to be configured (in a "setup mode") for any combination of MIDI inputs to talk to any combination of MIDI outputs (like a huge MIDI patch bay) and then to switch to "performance mode" to operate. The node interconnections are via fibre optic link and use a baud rate of up to 10Mhz to travel up to two kilometres. I don't know how successful this system is because I have not personally seen it, but there's no reason it can't be done.

This also brings up a new unit that we just designed for a Disney show that allows the creation of MIDI messages that can be converted directly into any type of RS-232, 422 or 485 transmission desired (any baud rate, protocol and data) and vice versa. The unit has one MIDI in, one THRU and one OUT; plus 4 independent serial ports using standard 25 pin D connectors. Up to 128 separate units like this can be used together, each with its own device number, so that a single MIDI controller (sequencer, show controller, etc.) can talk to up to 512 separate devices using their own serial protocols simultaneously. We are using them to talk to video disc players for this show..

---[0031]--- (pref = [0030], nref = [0032])

[0032] (11 lines) JNewstrom.USITT 04/03/90 1725.7 mst Tue midi
Subject: Re: LIEP Protocol

I think that MediaLink has phenomenal potential. I am including a chapter on it in my "Methods of System Synchronization and Interconnected for Live Performance" thesis. I was talking to a guy at Lone Wolf yesterday, and they just now are getting demo units to dealers. I'm trying to arrange a demo at a dealer in Hartford. The guy at Lone Wolf said that all the information they have released about the system (it is proprietary) is in the MediaLink article in the December 1989 Keyboard. When they get the SMPTE Tap and the Audio Tap, that will be a pretty interesting entertainment LAN.

John Huntington
---[0032]--- (pref = [0031])

[0033] (59 lines) CRichmond.USITT 04/04/90 0027.8 mst Wed midi
Subject: Suggestions

I offered to mail copies of transactions up to date to those who have not yet managed to get on Callboard to participate. Jim Fackert of CAE received one of these mailings and has the following suggestions to make:

1) To make it less complicated in the long run: what about calling "the code" NOT "lights", not "theater", but "Production Messages"A (real time)???

Theater is limited, and to some, a stodgy term. PRODUCTION fits shows, musical shows, theater, A.V. shows, industrials, etc.

2) I agree it's got to be REAL TIME, to accommodate synchronous cues.... but it also has to allow for dumps and loads. Even dumps should be high priority, as they may need to be done fast during a show!

3) USITT ASCII will help as a framework to organize the lighting cue format. There's a USITT forum planned on this.

You might should attend, and I will. They play a medium non-specific file transfer EOPformat which looks like it will work well for dumps and loads. I have read the current proposal over, and think it will work very well, and could be extended to allow for real time synch. Messages, too.

4) Re the message outline

a) NEXT CUE needs to be able to be a very big number, with or without decimal extension. (See Andy at Varilites recommendation, plus USITT ASCII which needs to be bigger too.)

b) With cues, chases, sets (or acts) etc., we need to be able to load and dump sequences, go to a next or a specific or a last cue in the sequence, start/stop an automatic retimestepping thru cues in the set, and go immediately to a specific cue, with or with out fade.

5) Why can't the "production arts consortium" buy a sysex mfg. and use it for development until a real number is accepted by MMA? This would insulate us from other experimenters, and also from the bureaucracy of the MMA, which appears to be considerable from the infor you sent!

-----End of quote-----

A couple of comments from me on this:

My only concern with "Production Messages" is that production is too often applied to film and video (and audio recording as well). How about "Live Performance" as a compromise?

Perhaps file dumps and loads can have some sort of priority but not quite as high as commands like "GO" "NEXT" etc.

Cue number references in data extension should also have multiple decimal point extension capability plus cue list number extensions, I think (see my previous proposal). Also, I just realised that I entered the "pound" sign in the above transcription as was used by Jim, forgetting that this will not print on this network - you'll have to read between the lines.

The MMA has suggested we could use the "Educational/Experimental" Universal sysex header until this is approved -- or we could agree upon one member's id for the time being (couldn't we???)

---[0033]--- (nref = [0034])

[0034] (7 lines) JLBracewell.USITT 04/04/90 1909.2 mst Wed midi

Subject: Re: Suggestions

Regarding priority of dumps and loads: These *usually* aren't as high a priority as a GO or NEXT; but if you're in the middle of a load in some circumstances, and a GO overrides it and your cue information was in the input you were in process of loading, you could be in real trouble -- possibly with future cue status uncertain. Unless I'm misunderstanding what's being suggested here, I think there's potential trouble if we head this way.

---[0034]--- (pref = [0033], nref = [0035])

[0035] (37 lines) CRichmond.USITT 04/05/90 0256.2 mst Thu midi

Subject: Re: Suggestions

I think that this is a valid point, however I had always envisioned a file transfer protocol that was intelligent and tough enough to recover from a higher priority interrupt. In fact, what actually happens in the standard MIDI file dump protocol is that the file is broken into packets and then transmitted one packet at a time (actually this is not prescribed but is in fact how most systems do it because of limits on buffer size). Each packet is pretty small so they get sent very quickly.

An interrupt occurs when a higher priority message manages to make itself known to the system, either by being generated on interrupt within the system's main controller computer (also the one managing the file dump, probably) or by being merged through a standard MIDI merge box into the data stream and the MIDI merge box sees a break in the file EOP dump message and sticks the waiting messages in. This process works in a fairly natural and non-intrusive manner even without merge boxes that "know" what priorities various messages have (i.e. they just send them all in the order in which they are received). Most MIDI messages and file dump packets are short enough that most incoming messages don't have to wait too long before getting the opportunity to be sent.

Obviously, this simplistic approach does not really address our concerns for priorities on a very busy system or one in which the file dump is not packetized or hogs the system. As a system designer, I would naturally want to avoid sending large amounts of data during a show anyway. It's much better to have all cue data stored in each intelligent controller's memory so I have a slightly cynical attitude toward those users who insist on going against the recommended procedures: something about getting what one deserves?

I talked to Jim Fackert today and we discussed the idea of two types of file dumps - real time and non-real time - in order to allow us to specify whether a dump is high or low priority. The standard MIDI file dump is non-real time already, so we could create a real-time dump protocol under the Theatre(?) Messages header. I would seriously suggest that if we do this it should be much simpler than the other one and leave out all the handshaking required - just fire it out and go.

---[0035]--- (pref = [0034])

[0036] (23 lines) BRodriguez.USITT 04/16/90 1706.5 mst Mon midi

Subject: MIDI Lighting Functions

At USITT in Milwaukee we spent some time during the MIDI session discussing what lighting control functions people want to perform with MIDI. The emphasis was not so much on what we could imagine wanting

to do, as what we had an immediate need or desire to do. This was to
{serve as a starting point for MIDI ymplementations. The following LONG
message is this list of functions.

I should note that the proposal described on this forum in message 0003,
and put forth by Charlie at that USITT meeting, was accepted in essence:
namely, the ideas of an extensible function set, and a different
function set for each class of controller. The people who contributed
to this particular function list were all lighting control peoplS {, so
the categories of rigging, laser, pyro, etc. etc. are not represented
here. Presumably, the interested groups will gather to produce their
own function lits, and then we'll look to see if there are common
functions which apply across the board....but I digress, and Charlie can
elaborate further on this.

My notes are skimpy, so if I misremember or misinterp{ret any of the
ideas from the meeting, someone]k please correct me.

--Brad
---[0036]--- (nref = [0037])

[0037] (68 lines) BRodriguez.USITT 04/16/90 1708.3 mst Mon midi
Subject: Re: MIDI Lighting Functions
THE MOST DESIRED MIDI LIGHTING FUNCTIONS as proposed at the 1990 USITT
MIDI session

GO /)STOP / RESUME GO starts a fade to the next sequential cue. STOP
halts the fade midway; RESUME resumes a fade which has been STOPped. It
was stressed that RESUME must be distinct from GO, since it may be
desirable to command the board to complete all current fades, but NOT to
start any new fades. (Unlike GO, RESUME never starts a fade.)

GO cue-number / STOP cue-number / RESUME cue-number GO cue-number starts
a fade to a given cue. (Another suggestion was for the function pair
"LOAD cue-number / GO" instead of "GO cue-number / GO next".) STOP
cue-number and RESUME cue-number are for the case of a cons~3le (e.g.
Palette) which can have several fades running simultaneously; they allow
a single, specific fade to be stopped and resumed, while leaving A7{hJo
others to run. (The implication is that STOP and RESUME affect ALL
running fades.) In all cases, cue-number is a null-terminated ASCII
string which may include decimal points.

SET generic-controller level This is used to bring up subs, channels,
etc. Rather than specific commands for each kind of controller, a
generic "bring up to level" command was proposed. The
generic-controller value is a two-byte (14-bit) number, and is treated
much like the MIDI continuous controllers: the use of this number is
manufacturer-defined, but values for several common controllers (e.g.
grandmaster, subs 1-12) will be recommended. The level value is one
byte (7 bits).

BLACKOUT on/off A separate BLACKOUT function, which doesn't affect the
setting of the grandmaster, was desired....especially because some
consoles don't have grandmasters, and some consoles have outputs which
are independent of their grandmaster. (BLACKOUT is a real blackout.)

FIRE macro-number Used to trigger a preprogrammed keyboard macro, in
those consoles which support this feature. Macro-number is a one-byte

value (0 to 127). The macros themselves are either programmed at the lighting board, or loaded via the file dump facility (below).

RESET-CONSOLE We weren't too clear on what this function encompasses, other than clearing all the timed faders and resetting the cue playback position to the top of the cue sheet. (Sort of like GOTO Q 0 on a Palette.)

Other functions needing more work....

Dump faders/buttons: a message to essentially "remote" all of the faders and pushbuttons of a console; i.e., this message would contain all of the handle positions and "pushed/not-pushed" state of all of the control panel buttons. Obviously, this is VERqY manufacturer-specific.

Effects: everyone wants MIDI effects control, but nobody knows a portable way to accomplish this. The best we were able to come up with was a "GO effect-number" function to start an effect, with level and rate control of the effect through the generic SET function described above.

File dumps: everyone seems happy with formatting file dumps in ASCII, in whatever format the ASCII Light Cues team comes up with, and using the MIDI File Transfer protocol to send it. Since the former is purely a "content" format, and the latter is purely a tra
-[missing text]-

Charlie's original suggestions for lighting functions (see midi message 0003) were also brought up, but caused some confusion due to unfamiliar terminology....I'll post my interpretations of these functions in a later message. Certainly they all seem useful.

--Brad

---[0037]--- (pref = [0036], nref = [0038])
[0038] (18 lines) CRichmond.USITT 04/17/90 0146.4 mst Tue midi
Subject: Re: MIDI Lighting Functions

Just to clarify my original proposal in message 0003, the list of functions which were provided were not intended to be the functions found in lighting consoles but rather the ones which we use for computerized sound controllers, since that area is the one in which I was most conversant. I really would not attempt to define a set of lighting commands and can really only observe the discussion regarding these since I have had so little experience with lighting controllers.

I don't want anyone to think that I am trying to singlehandedly change the entire scope of popular lighting commands!

Also, I am basically going to be out of town for the next 3 weeks and am trying to convince Brad Rodriguez to take over for that period of time (looks pretty hopeful so far, eh?) There is a possibility that I may participate if I have time since the computer will be with me - so you haven't gotten rid of the old curmudgeon completely.

In the meantime - party on dudes!
---[0038]--- (pref = [0037])

[0039] (11 lines) CRichmond.USITT 05/09/90 1819.1 mst Wed midi

Subject: Activity

Or you might say the lack of This is very encouraging (?) I was concerned about not being able to participate while I was away since everyone (not quite but you know what I mean) at the MIDI discussion at the USITT conference seemed so eager to get on Callboard to get this standard finalized. Anyway, the wait will hopefully be worth it.

In the meantime, I will provide in the following transaction (which will be long) the names of all who signed the sheet which was passed around at that discussion. San

Stand by...

---[0039]--- (nref = [0040])

[0040] (123 lines) CRichmond.USITT 05/09/90 1904.2 mst Wed midi

Subject: Re: Activity

Participants

The following names and replies were entered in response to the cryptic notation: Name/Address/Active on Working Group?/Callboard?/MMA? which was placed at the top of a pad of paper and passed around the room during the MIDI Session:

Jim Fackert, Leprecon CAE Inc. Box 430 Hamburg MI 48139, 313-231-3471, fax 231-1631/Yes/In Progress/Yes

Daryl Carson, CAE (above)/(above)

Ian Ibbitson, Strand UK/Hope to/Possibly/Yes

Ralph O. Weber/29 Adelaide St., Hudson WH[?] 030512

Scott Colstrom, 1113 Congress Apt. C, Emporia, KS 66801 / ?

Matt Kellen, 403 E. Main Apt. B, Vermillion, SD 57069

Steve Thompson, 5009 California, Omaha, NE 68132 /

Robert Neuerdof[?], 401 S. Weinbach, N214 Hughes Hall, Evansville IN 47714/N//

Chip Perry, 401 S. Weinbach, 208 Hughes , Evansville IN 47714/Like to/
/No

Peter Willis/Andera Ltd, 19 Glenfield Rd, London W13 9S2/Not at present/
No/No

Chris Heagel/1536 Hewitt Ave. St Paul MN 55104/HV Theatre/No/Like to

Thomas McDonough/ BR Box 1563 Edwardsville, IL 62026/No/Like to/No

Jonathan Darling/8040 Kostner Ave, Skokie IL 60076/Tech Prod/No/No

Pierre Gagnon (GALA) 930 Wellington, Montreal H3C 1V1 Canada

Ken Vannice 1707 Micheltorena418, LA, CA 90026/N/

Steve Terry/Production Arts/636 11th Ave NY NY 10036

Daniel Herman/804 Stewart Ave/Ithaca, NY 14850

Hanson Hsu/31-1/2 Maplewood Rd, Ithaca NY 14850/Please contact me

Terri Osborne/400 N. River Rd, Apt410/W.Lafayette IN

Jason Dunne/260 S. Chauncey6/West Lafayette, IN/Please contact

John Uthoff/4812 Lakewood Ridge/No/No/-

James D. Scott/PO Box 304 Storrs, CT 06268/U-Conn/No/No

Gregory Foulds - 80 Oak Ave Belmont MA 02178/ No/No

John Huntington/550 George2, New Haven, CT 06511/Interested in being in working group/CB=yes/MMA=no

Anthony Simone/St Olaf College, Northfield, MN 55057/Interested not active/ No CB/No MMA

Henry Hampton/7661 N. Sheridan12 Chicago, IL 60626/N/N/N

Brian Ress/253 Sheetz St12 West Lafayette, IN 47906/Purdue student

Frank Lartonoix Box 1777 Edwardsville, IL 62026 No/no/no

Eric Brackett/1745 Beeler, Speedway, IN 46224/Purdue Student

John Heil 111 S. Jefferson Millstadt, IL 62260-1549 No/NO/No/

Ed Prasser 9535 W Grange Ave Hales Corners WI 53130 No/No/No

Staci A. Bollinger 477 Warren Ave Pocatello ID 83201 No No NO

C A Milne PO Box 8625 Pocatello ID 83209 No/No/No

Rik Kaye 395 Riverside Drive, NY, NY 10025

Glenn Ragaishis 7252 W 78th St, Chicago IL 60652

Douglas Oeste 5455 N LaPorte Chicago IL 60630 N/N/N

Patrick Finelli 608 E River Dr. Temple Terrace FL 33617/Bitnet/

Rory Quayle 57 Foothill Blvd Pocatello ID 83204

Dale Chavie, 621 W 6th Ave, Anchorage, AK 99501

Thomas Utterback 704 W. Green2, Urbana, IL 61801 No/NO/No

John J. McFadden Jr., Department of Theatre, Memphis State University, Memphis, Tennessee 38152 MMA=NO CALLBOARD=NO

Sheila Heeke / 455-I N Grant W Lafayette, IN 47906/Purdue Student/

Melissa Marquis/110 S. 9th Apt 3 Lafayette, IN 47901/Purdue Student/

Alex Belden/831 Flume St Chico, CA 95928/NO/NO/NO

Matt Howell/1033 E. Loren, Springfield MO 65807/NO/NO/NO

Jim Christensen/1675 NW 216, Hillsboro, OR / yes-?-No

Louis-Philippe Demers/510 Jarvis10, Toronto ON M4Y 2M9/NNN

Brian Stuhlmacher 1602 E. Main Griffith IN N/N/N

Kyle Loudon/2140 Happy Hollow; W Lafayette, IN 47906 N/N/N

Matt Booty/1009 N 6th St., W Lafayette, IN 47904 N/N/N

Rick Thomas/919 N. Salisbury W. Lafayette, IN 47906 N/N/N

Brad Rodriguez/55 McCaul St.14 Toronto ON M5T 2W7 Y/Y/N

David S. Skok/6265 Hamilton Blvd. Allentown, PA 18106

Steve Norman 18111 So. Santa Fe Ave. Rancho Dominguez, CA 90221

That's all folks!!!

---[0040]--- (pref = [0039])

[0041] (14 lines) CRichmond.USITT 05/14/90 0108.0 mst Mon midi

Subject: Agreement!?

As you can see, I got caught by the dreaded "pound sign deletes the previous letter rather than prints itself" syndrome.

I would like to add here that my understanding of the Conference meeting consensus included the decision that cue numbers could be expressed as any ASCII character string (in the value range of 20 to 127 decimal) in order that unusual features such as slashes, letters, decimal points, etc. could be expressed in accordance with the convention of the console/system being spoken to. This still makes lots of sense to me. We also agreed that all command/data strings should have a length limit even though much longer strings may some times need to be sent. A convention to links packets together which contain related data should be part of the spec. I think we came up with a packet length of 127 bytes, but I'm not sure. Brad?

---[0041]--- (nref = [0042])

[0042] (6 lines) BRodriguez.USITT 05/20/90 1456.9 mst Sun midi

Subject: Re: Agreement!?

I have the same recollection -- packet lengths are 127 bytes or less -- although I haven't found it in my notes yet. (I'm still a bit disorganized since USITT, having much backlog to clear up.) This seems to be an "obvious" limit, given that a MIDI "byte" is 7 bits.

More when I dig ou{t from under... Brad

---[0042]--- (pref = [0041])

[0043] (71 lines) CRichmond.USITT 06/29/90 0024.9 mst Fri midi

Subject: Next TSBB

The next Technical Standards Board Bulletin will reprint as an action item a copy of the MMA Forum message 487 (our MIDI Forum message 0003)

for the technical standards board's approval. I have told the person (Stanley Junglieb) who is compiling this TSBB that we have achieved a EOP consensus of opinion that this proposal as it is expressed in the above message represents the basic approach which we wish to take for the transmission and reception of MIDI "Theatrical Messages"

Note that in this proposal the term <chan> and Channel Number is used whereas the current accepted use for this byte position in these types of System Exclusive messages is to denote "Device Numbers" from 0 to 127, where a value of 127 is normally used when all devices are to be addressed. To update this proposal, substitute <device> for <chan>.

I have also passed on the gist of our discussion regarding the phrase "Theatrical Messages" and our concern that this may be too antiquated, quaint or just plain restrictive and have talked with him about our interest in perhaps having this proposal eventually encompass a much wider range of technical performance devices more appropriately known as "multi-media equipment." My major concerns in this area were:

1. That we had not as yet embraced a wide enough range of players in the multi-media area in order that we could honestly say we represented both theatrical and media/film/video/commercial/industrial/what-have-you interests.

2. That this was very possibly a whole can of worms which I certainly had no particular interest in opening or taking responsibility for right now - but could actually see no reason why our proposed protocol could not indeed suffice for these expanded purposes for some time in the future.

At the same time, I enquired whether the rumour was true that a multi-media group had approached the IMA/MMA about forming a working group to define multi-media messages, many of which would probably duplicate the function of our own. Apparently this is not the case and we may be well advised to proceed with accommodating such an expanded protocol right away in the hopes that the multi-media people will become interested and lend support to our process by becoming involved. If they do, it will certainly add credence to the concept of using MIDI as a universal communication standard for live performance using technical devices of all types.

Stanley was quite taken with this idea and resolved to contact other movers and shakers in the MMA and elsewhere who may be in a position to forward or represent the interests of the multi-media industry (Apple being the primary target). Well, we'll see. If anybody reading this knows anybody in the multi-media industry interested in creating communication standards, please exhort them to climb aboard this process.

I will continue to report any further activity in this area as I become aware of it and those of you who are members of the MMA will be receiving your own copies of the TSBB. If anyone feels we have not achieved a consensus which is accurately represented by the proposal in message 3 of this forum, please let me know immediately. Also, please understand this proposal represents not the final form of our efforts but merely the framework on which we will build fully defined messages. The standards board will be approving this framework as an appropriate means to an end - not as a structure in final form to become part of the

new MIDI spec (but I sure hope we won't have to wait much longer before we can put forward our first Version of the "Theatrical Messages Spec" - Multi-Media Spec???? hmmm..L.A?m(Spec" - Multi-Media Spec???? hmmm....

Anyway, I'm doing this on-line and we seem to have picked up some garbage so maybe the phone company is trying to tell me something..

So much for now - more later!

-Charlie
---[0043]---

[0044] (28 lines) CRichmond.USITT 07/24/90 2227.6 mst Tue midi
Subject: Wrap up

OK, here we go again. I have been advised by those with more experience in these matters that the lack of participation in this forum probably means that everybody thinks that what is going on is just wonderful and they can't wait to have it all become finalized. Beyond that, people who might comment are hesitating to do so in case it creates any more delays in the process. So, with this assumption in hand, I have advised the MMA that we will be putting forward a final version of the proposal within the next 6 weeks in order that the Technical Standards Board will have a fully complete and technically correct document on which to vote.

This document will comprise "Version 1.0 of the MIDI Theatre Messages Proposal" and will include the currently espoused format with a few more technical areas added and a few more types of messages added. Please let me know if you want any specific technical areas represented in this first draft and if you want any specific types of commands added or deleted. If I don't hear from anyone, I will post a suggestion for the final draft on this forum shortly and then we'll see who howls.

One additional comment - in the message area which we have called "data" and "any additional comments", I have advised the people drafting the TSBB that we have come to a basic consensus that 7-bit ASCII data is adequate for our needs, since we agreed that cue numbers, for example, could be comprised of any ASCII string of any length up to 128 (?) characters. The TSBB should reflect this and all those on the MMA mailing list will be receiving the TSBB in any case. Perhaps this mailing will prompt some response.

Ta Ta For Now..
---[0044]---

[0045] (39 lines) CRichmond.USITT 07/27/90 0104.8 mst Fri midi
Subject: Lighting commands

I've been studying Brad's message (37) describing the "MOST DESIRED MIDI LIGHTING FUNCTIONS" and would like to get some comments on the two options proposed for calling up a specific cue number and making it "GO"

If you look at this proposal, there are two methods described for this:

1. LOAD cue-number / GO
2. GO cue-number / GO next

Brad described the differences in these two methods but did not indicate

that there was a preference for either one. Being the impartialo~
9%reporter that I am, I have no inherent preference for either and
wonder if we can utilize this marvel of instant electronic
communications to come to some concensus. If we can't, I'll probably
suggest that we incorporate four distinct commands so that people can do
it whatever way they want. i.e. different manufacturers may choose to
support one method or the other - or both, having them perform either
identical functions or possibly coming up with subtle differences in the
way the two command combinations work.

Anyway, comments and discussion would be appreciated.

Also, two other areas were mentioned which need more work:

1. Dump Faders/Buttons
2. Effects

Obviously, no more work has been done or you would have seen it here.
Brad did suggest that both of these could simply fall under the SET
command and since we have 14 bits available to define which generic
controller we're SETting, there should certainly be enough for now.
Perhaps in the next round we can make a further extension for these but
for now I think we should let it ride.

If anyone disagrees please say so. Thank you!!

Ahhh, ain't progress wonderful!
---[0045]--- (nref = [0046])

[0046] (8 lines) BillWilliams.USITT 08/04/90 2117.8 mst Sat midi
Subject: Re: Lighting commands
Re: the CUE GO FUNCTION, I think both of the following are important:
a.) GOTO/CU (/)GO b.) LOAD/CU (/)GO ..

..
---[0046]--- (pref = [0045], nref = [0047])

[0047] (1 line) CRichmond.USITT 08/08/90 2343.2 mst Wed midi
Subject: Re: Lighting commands
Thank you... Watch this space and see what you think.

---[0047]--- (pref = [0046], nref = [0048])

[0048] (28 lines) BRodriguez.USITT 08/13/90 1910.1 mst Mon midi
Subject: Re: Lighting commands
Re. message45: my comments on LOAD cue vs. GO cue are in a separate
message. Other comments:

SET generic-controller level: It's been pointed out to me that, if
you're planning to use this feature to perform fades, 7 bits may not be
enough resolution. Fade stepping can sometimes be seen at the low end,
with fast (low-wattage) loads, when using only 7 bits.

Dump Faders/Buttons: I really don't know if this is a useful function
or not. On the one hand, it's guaranteed to be "non-portable" -- it
won't work the same on any two different consoles. On the other hand,
all the manufacturer needs to do is provide a reference table in the
back of the manual, and the user of _his_ console can do nifty things.
Are we pushing manufacturer-independence here, or do we want to cover

all the bases?

Effects: Similar comments, although I think I have come up with three reasonably- portable functions which are likely to be desired in a MIDI environment: "GO effect-number", to start an effect, as I previously described. "STOP effect-number", to stop and clear the effect. And "STEP effect-number", as a real-time operation. Presumably, consoles which can run multiple effects will assign multiple GO commands to the different effects generators automatically (rather like Palette assigning cues to faders). Then a STOP command is needed to free an effects generator, and the STEP command should indicate which effect (_not_ which effects generator!) should be stepped. Comments, anyone?

- Brad

---[0048]--- (pref = [0047], nref = [0050])

[0049] (38 lines) BRodriguez.USITT 08/13/90 1911.7 mst Mon midi

Subject: GO cue vs. LOAD cue

In response to message45, and other queries I've received:

The distinction between GO and RESUME is that GO always starts a new fade, and RESUME never does. RESUME simply "re-starts" all suspended fades and allows them to run to completion. GO will start a fade to the next cue, even if the current fade was STOPped before completion. Note that some consoles use the GO button for both of these functions; e.g. on Palette you press STOP/BACK to halt a fade, GO to resume it, GO again to start the next fade. (As I recall.)

I'm afraid I don't know the philosophical underpinnings of the "GO cue" vs. "LOAD cue/GO" controversy. I do remember when this second function was added to Palette (around 1983 or so), which means that at least one vociferous customer somewhere wanted it. Again using Palette for example: when you type "GOTO CUE nnn ENTER" you get an IMMEDIATE fade to that cue, regardless of what was in progress; this will clear anything that was running on the faders. The new function "LOAD CUE nnn ENTER" simply set up a fade to that cue on one of the faders, WITHOUT disturbing what was in progress; then the next GO would do a fade to that cue. (This was somewhat confused by Palette's peculiar algorithm for assigning cues and parts to faders.) So, can someone tell us who first asked for this feature, and why it is desirable?

The functions "STOP cue-number" and "RESUME cue-number" were concocted at the Milwaukee MIDI session, as a hopefully console-independent way to grab control of multiple parts. Palette exempli gratia: pressing GO-GO-GO-GO for single-part cues Q1 through Q4 will cause the four cues to run on four faders. Depending on the outcome of previous actions, these may not always be put on the same four faders! An operator can look at the screen and know to press the fader3 "manual" button to take control of cue2, but -- short of downloading all kinds of configuration data -- a MIDI controller can't do this. It was finally decided that what the controller _really_ wants to do is to grab a certain CUE, not a certain fader, and so the protocol should make requests by cue number and let each individual console figure out how to get control of that cue. (Not all consoles can do this.)

- Brad

---[0049]--- (nref = [0051])

[0050] (46 lines) CRichmond.USITT 08/14/90 0141.2 mst Tue midi

Subject: Re: Lighting commands

This has come at a very opportune time - I have just uploaded to Tim Clinton the first draught of the Theatre Message proposal which I have titled "MIDI Show Control 0.01" (hope you like it). He will organized this into manageable downloadable segments and place it in >udd>USITT>common>standards> (I don't know what the name will be yet) as soon as possible. I hope you all get a chance to retrieve this and look at it as soon as possible. It is 22 pages long so don't try to print it to your screen.

Comments on Brad's message:

SET: see the proposal I have posted - I am suggesting more precision than originally agreed upon. What the hell, bits are cheap!

Dump: I think this is really the reverse of SET. i.e. when you push a button on the Controlled Device (the lighting console) it sends out a whole bunch of SET commands which represent all the functions, buttons, switches, faders, and do-dads and their respective settings at that instant (or even dynamically). The purpose of this is so that in an about-face of the normal connection, the Controlled Device sends data to the Controller not to control it but so that the Controller can quickly record all the current SET data as a specific set-up (or sort of Show Control 'Cue') which can then be sent back to the Controlled Device during the normal operation of a show to put the so console quickly into the SETup it was in when the whole thing was recorded. Of course, the data can be edited in the Controller so that changes can be made selectively.

The bottom line here is that no new commands need be created and that SET will work fine in either direction defining generic controller data for any kind of function.

Effects: This brings up the discussion of whether or not an Effects command really needs to exist as an entity separate from SET and from all the normal Cue oriented commands. Really, with these two commands any effect you like can be called up and controlled within any range of capabilities. The FIRE command was put in my draught simply because it had been agreed that it was necessary at the MIDI meeting at USITT. I don't really have a feel for it, but if indeed all the lighting people feel that the new effect commands Brad is suggesting are necessary, by all means they should be included. Lord knows we've got lots of room to breathe and lots of space for more commands!

Hear here!(?) it for activity on this hear forum.

Charlie.

---[0050]--- (pref = [0048])

[0051] (3 lines) CRichmond.USITT 08/14/90 0142.8 mst Tue midi

Subject: Re: GO cue vs. LOAD cue

Please take a look at the draught proposal and see if it meets these requirements. If not, how can we define the commands more appropriately? Yr fthfl srvnt.

---[0051]--- (pref = [0049], nref = [0052])

[0052] (4 lines) Clinton.Thea 08/14/90 0752.0 mst Tue midi

Subject: Re: GO cue vs. LOAD cue

Charlie's draft proposal is now in position in the standards directory.

The full pathname to it is:

>udd>USITT>common>standards>midi_show_control_0.01. For ease of typing when you download, it also bears the name MSC_0.01

---[0052]--- (pref = [0051])

[0053] (8 lines) STwose.USITT 08/23/90 1203.3 mst Thu midi

Subject: Sweden Online

Hi This is Anders Ekvall from AVAB. I have finally managed to connect through all the amazing network wires. I will soon give my comments to the discussion going on.

Thanks to Charlie for the great support trying to get me connected.

Stay tuned!

---[0053]--- (nref = [0054])

[0054] (1 line) BRodriguez.USITT 08/23/90 2010.7 mst Thu midi

Subject: Re: Sweden Online

Hi Anders! Glad to see you on line! - Brad

---[0054]--- (pref = [0053])

[0055] (127 lines) STwose.USITT 08/27/90 1238.8 mst Mon midi

Subject: Comments on the discussions

Hi, Sweden again. I am trying to copy a file into this system.

Hopefully it will work.

-[missing text]-

Here are some comments I have collected through the last months. I think that some of the topics have been discussed and somehow agreed on already. I have not had the possibility to follow these discussions very well, so please excuse me if I repeat old stuff.

The idea of a special sysex code for light sounds OK by me, and I think that such a code should be decided on and approved by MMA. However, I would NOT recommend it for sending loads of submaster information or other heavily used messages.

Cues AVAB has implemented MIDI in a big, professional lighting system. This means that we think it is unsatisfactory to access preset through Program Change numbers. We would like to see two methods for accessing cues: MTC and sysex. Each cue (or crossfade) could be linked to an absolute time in MTC. This makes it easy to connect to a SMPTE-synchronized system. If, however, the sysex message should be a standard for accessing cues I would like to see the cue number packed in some way. Why not use the same packing method that is used in the Standard MIDI File format for specifying the number of clock ticks between each message. This is a dynamic and compact format. Because the sysex message itself includes a lot of overhead (6 bytes) this can save some valuable time and bandwidth. In the specification I can see nothing about what will happen when a Program Change or a sysex message is used for accessing a cue. Will the cue be cut in on stage or will it fade? The normal situation for theater use is, of course, to FADE in a new light from a new position in the sequence. Therefore you will need another message as a complement - the Start Fade message. That could be one of the additional sysex messages - you send a position message followed by a GO message. Each crossfade should be possible to assign as a real time crossfade or a crossfade run by MTC. Then all fades will adjust to the external time base.

Bump Buttons There should be possible to connect a standard MIDI equipped keyboard to the lighting console and play with the light channels (or submasters) using the keys of the keyboard. If you have a velocity sensitive keyboard you shall also have velocity sensitive flash. To be able to work with most keyboards there shall be a setting in the console where the user can specify which light channel shall correspond to each note number. This can be made using a Start Note parameter which tells the system which note number that will access the first light channel. It can also be made through a Patch which connects any note to any channel. It shall be possible to select if the console shall Bump light channels OR submasters.

Channel Data I strongly dislike the idea of sending light data using standard Note On/Off messages. This can only be used by very small consoles (up to 24 channels) because the transmission speed is far too low to give an acceptable result on stage during a fade. For light data you will need something much quicker such as the DMX512 standard. Eventually, complete lighting frames can be sent through a sysex message.

Submasters First of all I don't agree on the opinion that we should avoid using reserved controller numbers not to interfere with musical information sent on the same MIDI channel. I think that this just complicates things, making the messages longer and the system slower. If you mix light consoles and instruments on the same MIDI line you should of course have them on separate MIDI channels! In this case it is very important that we try to keep the submaster messages as short as possible. Submaster messages are probably the most transmitted messages at all because of their continuous nature. I strongly recommend using the high byte of the continuous controller messages and not caring about if the controllers are assigned to a specific musical function. Of course, care should be taken to avoid using controllers that sequencers treat in a special way such as Modulation. Another important aspect is that the fader movements shall be graphically editable in a sequencer program. Thereby you can sculpture the fade profile in a computer. This is a very useful function for professional theater. I don't know how a sequencer program displays the lower byte of the controllers. Can they also be edited graphically? Most instruments only use the high byte of a controller so I am not sure that the manufacturers of sequencer programs have included handling of the low byte. Does anybody know? There should be at least 24 controllers for use by submasters. Seven bit resolution is OK.

Many sequencers can handle Controller chasing. This means that if we use a controller message for sending submaster information, the sequencer can, by itself, figure out the correct settings for each submaster after a jump. During rehearsal and recording of submasters into a sequencer this is very important. That can not be achieved if we use SysEx messages.

Other controllers I am not sure on which additional controllers we should agree on. The Grand Master should be included. Regarding Chase Speed I think that consoles have a more different approach. Our consoles can run many effects at the same with different Speed masters AND Effect master. Maybe we should have at least one Speed Master AND one Effect Master. An Effect master controls the total light output from an effect. Also there should be put aside controller numbers for at least two split crossfaders (that equals 4 controller numbers). You

must be able to record and graphically edit the progress of a manual crossfade in the same way as with submasters.

Function Keys There should also be a system for sending up to 128 function keys. This shall be used when two consoles shall be synchronized together or the actions of one console shall be recorded into a sequencer and then played back. Both key down and up should be possible to send and receive. This seems to be analog with the Note On/Off message. I suggest that the Note On/Off message can be used for this as well. If the user can select individual MIDI channels for the Flash function and this Synchronization function this will not be a problem.

MIDI Timing

Clock It shall be possible to synchronize effects with MIDI Clock. I am not sure on how it should work with Start/Stop/Continue because you should be able to run and control several effects at the same time. Maybe they should affect all effects currently active

MIDI Time Code MTC could be used for activating an lighting event at a specific time in the same way as you make SMPTE list in a sequencer for activating sound effects to video etc. The consoles should then have a possibility to store such event lists and respond on them.

SysEx There is a big problem with using SysEx messages: Many sequencers (even big computerbased ones) do not record or handle this type of data properly (or at all). This seems to me as a very big limitation and a drawback of the whole idea with using MIDI.

Anders Ekvall/AVAB

---[0055]--- (nref = [0058])

[0058] (56 lines) CRichmond.USITT 08/28/90 0130.5 mst Tue midi

Subject: Re: Comments on the discussions

I have deleted the latter two transactions, Anders, since your first upload was completely successful!

In general, I think your comments are all well taken, but we need to have you review the latest version of the current proposal which represents the concensus (or at least my understanding of it) to date. I think you will find that a number of your concerns have been acknowledged and dealt with, albeit not necessarily in the manner you may prefer. Ifr these directions are not to your liking, please let us know here how you feel they may be improved. In most ways, we have chosen to solve the problems and needs you identify by providing control capabilities which are very broad and generic by definition. We feel that convention may soon establish "normal" control functions for many of these generic controllers. If you feel the need to establish defined uses now, please suggest the control ranges and their respective descriptions.

Also, I don't think anyone is really trying to preclude the use of regular channel messages for lighting functions but merely to define a "parallel" dedicated world in which we may start developing powerful new systems with their own "sequencers," editors and controllers which are

specially oriented to Show Control and really do not have any of the compromises which exist as a result of using a system designed for music. Many of the software functions currently used in existing sequencers will probably be easily adapted to the kind of control needed by MIDI Show Control, and a lot more will have to be created but at least we will have our own SysEx area which can be tailor made by us to do what we need it to do, both in the Controller and the Controlled Device. I think we will see consoles with a switch allowing them to be controlled by either standard channel messages, MIDI Show Control messages, or both so we can have the flexibility of using a dedicated show controller or a keyboard or both.

-[missing text]-

To download the current version of the MIDI Show Control proposal, do the following:

1. At the Ready: prompt, type 'kermit';
2. Then type 'send >udd>USITT>common>standards>MSC_0.01
3. Then prepare your computer to receive a large file and store it
4. When finished, 'quit' then signoff and review the file with a text editor or print it.

We have received comment on this proposal from the MIDI Manufacturers' Association. They feel it is very well presented and looks quite complete and thoroughly described (what do they know?). At any rate, we need to get everybody to look at this and find out what's missing if anything or if there are any problems. Hopefully we're pretty close (at least for the first go round) and we can get this approved and usable! That will be very exciting for those of us who want to start using it. We have plans to create a full-featured controller system as soon as possible.

Thanks again Anders for your valuable input here - we look forward to getting your specific comments on the proceedings to date!

-Charlie

---[0058]--- (pref = [0055])
[0059] (88 lines) AEkvall.USITT 09/03/90 2328.7 mst Mon midi
Subject: Comments

Hi there

Finally under my own login. From now on it will be AEkvall and nothing else.

I have now read the MSC proposal. I have a few comments on it. Also I have some questions.

General reflexions

I agree that the sysex format is necessary for the type of commands suggested in the proposal. Still, I think there will be a big time problem partly because of this. We really have to make each message as short as possible. Maybe the list of possible devices has been made too extensive for this simple and slow transfer method.

About using sysex: I think that we have a problem here because there

are very few sequencers out there, for the moment, handling sysex in a proper way (if at all). This means that this whole idea is depending on stuff that does not exist now. We can not be sure that they will exist (or when) either. How soon can there be cheap sequencers available in the shops for doing this? As the proposal is today, we can easily implement it in our consoles within a few days. But when will there be other equipment ready to communicate with it? So far, AVAB has been using a very simple approach, using Notes and Controllers for all basic (and time critical) functions. We have been able to use any standard sequencer for this. Only in special cases we have used our sysex ID for doing data dumps etc. This has worked very well because of existing, well-tested third-party products. Do not get me wrong. I am not against standardization, I think it is necessary and good for all parts, but I can still envision a danger here.

Comments

I think that the suggested method for transferring Q-numbers in ASCII is general but very byte-inefficient. Do we really need this flexibility. The sysex message is long enough as it is. Can't we pack it in some way?

TIMED_GO: In the lighting business we use separate times for in- and out-going fades plus an optional delay. No provision seems to be made for this. Is this supposed to be programmed in the controller itself. If so, why do we allow a time to be specified at all. That one could also be programmed in the controller. A start fade command would be necessary.

SET: If you specify a time, does the transition start immediately? I would also like to propose some Generic Controller numbers to start with (Sorry for only proposing light oriented controllers.): 0-127 Sub masters. 128 should be enough.

128-129 Masters of the first playback. 130-131 Masters of the second playback. ... 191-192 Masters of the 32nd playback. 32 different playbacks should be enough.

193-224 Speed controller for the above 32 playbacks.

225 General speed controller for all fades.

226 Grand Master.

We have many controller numbers to choose from so I think that we should reserve quite a few for them for each function. Should we reserve some of these controller numbers for remote control of the pushbuttons of a controller? To be able to record actions done on a controller and play it back is a very important feature. Can't we use the Registered/Non-registered Parameter MIDI message for sending faders to speed it up. When making complex arrangements, not even regular controllers will be fast enough!

ALL_OFF: Is this a true Blackout?

GO/JAM_CLOCK, START_CLOCK, STOP_CLOCK, ZERO_CLOCK, SET_CLOCK: I do not really follow the discussions about Jam Clock and the Auto Follow clock

timer. Can someone please explain it more carefully for me. What will be the use of it?

OPEN_CUE_LIST: Is a Cue list the same as a sequence? We are using a slightly different vocabulary so I am not sure.

OPEN_CUE_PATH: Is this the same as a complete play or performance? This is what we have stored on disk.

Anders Ekvall

---[0059]--- (nref = [0060])

[0060] (134 lines) CRichmond.USITT 09/05/90 0105.1 mst Wed midi

Subject: Re: Comments

I'd like to respond to your comments as well as I can, but will have to leave some lighting specific responses to others, since I can not represent the original intent of all of these commands by the lighting community which created them since I don't pretend to be an expert in this area.

First speed: I have felt and the MIDI community has believed for some time that the delays one typically experiences when controlled devices respond to commands is due more to the lack of speed of the device's processor due to inefficient code, slow processing capability or both.

At the rate of 32kbaud, even a fairly complex message of, say 32 bytes will be completely transmitted in 8mS. Admittedly, this is not instant but since most of the messages we are dealing with in this context are extremely global ones which can cause immediate and total change of the scenic environment (by calling up and executing a cue sequence), 8mS is not much different than anything less than that.

I suspect that your concern lies more in the area of having many different real-time controls needing to send continuously changing data via MIDI from one controller to another. Obviously, in this context and in the similar one in which massive amounts of musical information try to be sent by many synthesizers, there is a distinct limit to this capability. But, just like musicians have discovered, we have to acknowledge the limitations of this medium. Some solutions are to limit this kind of use, to string separate communications links, or to link controller and controlled device with a dedicated high-speed interface rather than trying to piggyback onto MIDI.

In the future, with the existing commands in place, a new high-speed MIDI link will accommodate such applications without having to reinvent the wheel (only the axle on which it rotates).

On sequencers: I share your concern about the lack of proper handling of real-time sysex messages by popular sequencers, however this has been brought to the attention of the developers of these programs and they have been admonished by the MMA to get their software to deal properly with real-time messages as soon as possible. To be fair, so few devices have used real-time, the demand really hasn't been there. We must demonstrate the need by creating this powerful subset of the real-time sysex area.

No one is allowed to put a product into the public's hands which implem

implements any version of MIDI which is not 100% approved by the IMA and has been published as part of the MIDI standard. For this reason, it is impossible to purchase a product which supports this proposal. But... the moment it is approved and standardised believe me it will be available and it will be used immediately. We have several customers who are waiting anxiously for this to become a reality.

About ASCII Q-numbers: I agree to an extent, but we felt it was easier in terms of universal editability, plus it can eventually support other ASCII characters, should it be deemed necessary. Whatever the consensus here is OK, I think, for all of us, provided it allows the necessary sub-cue partitions.

TIMED_GO: the lighting people will have to deal with this, except that I had envisioned the CLOCK of our system (audio) to be the device which basically controls the optional delays described. These are indeed programmed in the controlled device normally when "automatic follows" to any particular cue are written in an unvarying manner. In our system, such follows can also be executed remotely by the controller via GO, or other specific cue referencing commands, since every follow is a cue with its own number and exists in its own right (but just happens to have a "auto-follow time" and an "auto-sequence number" If a different scheme is deemed necessary or more logical for lighting please elaborate.

SET: Yes, I think there should be no delay. Basically, this is simply saying the control should be changed now. If it is to be changed anything other than instantly, there is a time prescribed, but at any rate start now. As for the controller numbers suggested, since we do have plenty, we can do this. On the other hand, we can leave such specifics till the next round of this spec. Comments?

A manufacturer can use this standard or any other MIDI standard such as registered/non-registered parameter controls to send faders - the more flexibility the more universal the control system they can offer. We are not trying to limit flexibility by preventing such use as well.

ALL_OFF: What is the definition of a true Blackout? i.e. where all outputs get turned off but the cue still displays the current control values, or where everything goes straight to 0 and to recover you have to recue? If this can be further clarified, it shall be duly entered and set down in the book of the law.

CLOCK stuff: this really has to do with the Sound extensions of the command set and is not intended to be used with lighting, except that I had seen that it could possibly be used to time automatic follows (which is what we basically use it for). Beyond this, we can do all sorts of manipulations to the CLOCK in order to change the pacing and synchronization of the cues to the show. The corresponding functions in lighting are more closely related to the speed controllers for the various cue playback controls and the general speed controller. Since audio is a different beast from lighting, we have to manipulate sound cues differently and the CLOCK is the best way since it can be made to correspond on a frame-to-frame basis with audio source material if desired (or not, as the case may be).

The best I can suggest is that we get together at LDI and I can demo our software.

OPEN_CUE_LIST: Basically, a Cue List is a sequence with a numerical cue number relationship - low numbered cues to high numbered cues. They automatically arrange themselves in numerical order and execute in that order as GOes are received by that list. A Cue List may be anything from a short sequence with just a few cues to an entire show. Typically, each Cue List contains cues which control a specific range of performance devices, such as the pyro cues, the fireworks, the symphonic audio cues, the explosive audio effects, the gunshots, etc. but over the entire length of the show. In this way, all these performance devices may be dealt with for editing and control separately from the other devices in the show and they may even be teched on their own by running only their Cue List.

OPEN_CUE_PATH: Really this is a set of stored cues which may be opened with this command in order to place them into a cue list. The implication is that each Cue Path has as its inherent description information telling what Cue List it belongs in. Opening a new Cue Path in most cases means loading the cues found in that Cue Path (storage location) into a newly created Cue List which will be immediately made accessible to the current show. If a Cue Path is opened during a show, the Cue List will appear with the next cue in logical numerical order standing by according to the current or active cue number.

Again, with these two functions, I would have to show you the way our software works. If similar functions are required for lighting, we will have to determine whether new commands are necessary or whether these make sense.

Thank you very much for your prompt and considerable analysis of the MSC proposal!

-Charlie

---[0060]--- (pref = [0059], nref = [0061])

[0061] (17 lines) RClinton.USITT 09/09/90 1655.8 mst Sun midi
Subject: Re: Comments

Hello all. I will be joining callboard on a regular basis when I get the starting package but am reading off of a friends screen for now. I have been using sysex for a while now and have not seen a computer based system that does not support it. I am using a Mac system, but as far as I know my Atari-based friends are using sysex as well. This sounds like the beginnings of great things here... see where the midi spec. has led us. If there is a problem with resolution of the controllers in the continuous controllers it is possible to utilize two controllers together to send out a message with resolution far higher than the 128 point grid. One example of this is used in the pitch controller. There are(I believe) 31 controllers reserved for use as the second controller in such a pair. If there are not enough channels here for a large system the solution could be paralell midi wiring such as in the Mark of the Unicorn system that supports up to 512 independant midi channels. Talk to you all later!

Doug Blackley on Robert Clinton's computer ..

---[0061]--- (pref = [0060], nref = [0062])

[0062] (13 lines) CRichmond.USITT 09/09/90 2111.5 mst Sun midi
Subject: Re: Comments

Thanks for your comments, Doug. Hope you get a chance to study the MSC 0.01 proposal - the command 'SET' addresses the problem you are concerned with: This command provides as standard the ability to set any of 16000 controllers to any resolution within a range of 16000 values. It also includes an optional Time specification which defines the rate or speed at which the controller assumes the specified value, as long as the controlled device supports this (if it does not, it ignores the time spec).

Hope you find this a reasonable approach - if you think we need more, please let us know now!

Thanks...

---[0062]--- (pref = [0061])

[0063] (67 lines) AEkvall.USITT 09/10/90 1059.8 mst Mon midi
Subject: More questions
Charlie

-[missing text]-

More comments on our answers:

TIMED_GO: In/Out and Delay are programmed in each individual console. In lighting cases I think that the time in the TIMED_GO command should be ignored.

SET: Why can't we start defining the controller numbers? We also have products that we want to release following these standards! They should be defined A.S.A.P. so manufacturers can start using them as a standardized complement to their own methods.

You did not answer my request for pushbutton codes.

ALL_OFF: All_off should inhibit the complete light output from the console (setting all channels to level 0%). What is made internally does not matter so much, as far as all light output is inhibited. RESTORE will enable light output again.

Basically, I think that there is a somewhat confusing mixture of functions for completely different controllers. Also, I am a little bit suspicious to including all these very specialized functions that only can be used in your existing software. Correct me if I am wrong, but I get the feeling that you, Charlie, is making the standard fit your product and not the other way around. To understand the benefit of some of the functions, you say that I have to look at your software. Functions to be included in the standard should be of a more basic nature than that.

I am still trying to include the type of functions that I know many lighting consoles use when it comes to MIDI. They (and we too) use MIDI to be able to record and playback real time events including push buttons and faders. This is a new and very powerful way of working with light in a dynamic way. I think that there should be more provision for this type of control. We should reserve SET controller numbers for this as soon as possible.

OPEN_CUE_LIST and OPEN_CUE_PATH: I still think these functions are a bit strange and specialized. If we should allow more than one cue list to be active at the same time, we must have a way to assign which cue transition to be controlled from which crossfaders or controllers.

I think that there should be provision for having multiple sequences (cue lists) executing in parallel controlled by separate controllers. You can then execute fades manually and change the fade speed individually.

Do you mean that each cue list controls different consoles? Are they distinguished by use of the device ID in the header?

By the way, we do allow the user to arrange their sequence in any (even non-numerical) order. The same cue number can appear several times in a sequence if the same light appears on multiple occasions.

I do not like the idea of the CUE_PATH. It seems to be very specific to your product.

Excuse me for being a bit negative. I do want to see your software anyway.

Anders

---[0063]--- (nref = [0064])

[0064] (141 lines) CRichmond.USITT 09/11/90 0217.4 mst Tue midi

Subject: Re: More questions

Thank you, Anders. I am not put off by your concerns, but I feel I must respond to them in detail, so here goes.

Generally, all the commands which we agreed upon by consensus at the MIDI meeting at USITT in Milwaukee are the ones starting with 0. These commands are ones which were defined by Brad Rodriguez as a result of the discussions amongst all lighting manufacturers present. I was not involved since I really have little to do with this area. I attempted to put these commands into a general form of description for the purposes of this proposal and for that reason they read as generic commands, not necessarily just for lighting.

Again, the commands starting with 1 are ones which we consider to be essential to the operation of our audio control system. I invited several other manufacturers of similar systems to become involved in this process so we could achieve a consensus, but only Soundcraft actually got so far as to say they would - then they shut down their entire theatre sound development. So, I can only respond that we have had a successful theatre sound control system on the market for almost 5 years and it still remains the only one interested in using this standard, so we have to push for the commands we need.

The simple solution is to have lighting consoles respond only to commands starting with 0 and ignore commands starting with 1, and this is exactly what I had intended when this draught was written.

TIMED_GO is a command which came out of the lighting meeting - I have no plans to implement this in our system since we, also, program in/out and delay internally, so it remains firmly in the lighting area. Ob

Apparently a substantial number of lighting manufacturers feel the need for this. If we can get a consensus here to delete this command, we will do so, but with the vast amount of participation we are seeing currently, this remains doubtful. You can always ignore the command or respond to it as specified i.e. by simply GOing normally using your programmed time instead.

SET: No reason at all that we can't do this. I was simply hoping that someone else would get into this conversation to say either "yes let's define controller numbers" or "no, let's not" My recollection was that there was a consensus not to define controller numbers now, but to save it for the next round. There are lots of them so it really isn't a problem either way as far as I'm concerned. It would make me feel a little more comfortable having at least one other lighting manufacturer say "yes, this list of controllers seems reasonable". It's a little different from the fact that we are the only sound manufacturer involved - we are genuinely trying to create a usable sound standard without the assistance of others and I'm sure you have equally honourable intentions, but since there are a number of other lighting manufacturers on this network, it would be nice to hear from them!

I thought my response would also cover your comments about pushbutton codes. These could also be thought of as controllers, but with a very restricted range of setting values. Please put forward your suggestions and in the absence of any other opposing opinions, they will probably be put forward in the next draught proposal. (why not, indeed?)

ALL_OFF and RESTORE: This description certainly falls into the acceptable range of responses to this command. This really has less to do with this spec than it does the individual characteristics of each console. If you don't think the spec adequately conveys this response, or is ambiguous, please suggest alternate wording. I took the wording used from Brad Rodriguez synopsis of the meeting proceedings. Our own response to this would be to actually show the control settings as going to zero (i.e. showing the "on stage" condition of all off (equal to blackout), but to put all the settings just prior to the ALL_OFF command into a buffer which can be implemented upon the RESTORE command. This is slightly different from the description but meets the intent of the spec, I believe.

Again, I can only say that my considerable effort which has been and still is being expended in getting this proposal pushed forward is totally a result of the need to standardize the functions and commands which may seem foreign to lighting manufacturers but which are absolutely key to our system (as much so as the 'standard' lighting functions) as well as to other sound system designs which need to perform the same functions as ours. We are just as anxious as you are to release products which incorporate this standard, and we have customers who are anxiously awaiting their release. I can only try to assure you that all these functions are well considered and are extremely important to the implementation of our system in a very basic sense, with the very first project planned designed to utilize all these commands quite extensively. Our feeling is that these are indeed very basic in nature, since we are able to define them before they are even implemented. The ones that are less basic are those not yet considered. You must also understand that from our position, there are a number of lighting commands, controls and functions which seem just as pointless and esoteric from our perspective, but that is the nature of

this kind of thing. I'm quite pleased that we have come up with a basic list that is relatively short and digestible!

Being able to send and record real time events inclusion push buttons and faders is absolutely a goal of this proposal. If there is a flaw here which prevents this, please identify it to us and it will be rectified. We can standardise controller numbers as soon as there is some agreement amongst lighting people - perhaps we need to phone some of them, because so far only you have responded to the message I have sent to all the lighting people who have joined callboard so far. Hopefully, some of them will read this and feel that now is the time to respond if only to affirm their agreement.

OPEN_CUE_LIST and OPEN_CUE_PATH: I agree these seem a little strange and even we have not fully implemented this. We have a number of ways that cues and controllers are inter-assigned on a cue-to-cue basis, allowing any number of cues (including none) from any cue list to be able to control crossfaders or controllers simultaneously plus at the same time be able to dynamically assign control of manual controller devices to different programmed controls on a cue-to-cue basis. In sum, this fulfills your stated desire to have multiple sequences (cue lists) executing in parallel controlled by separate controllers. We go the one step further by defining within each cue which controllers will control which devices for which sequences. This allows executing fades manually and controlling crossfades individually and being able to program exactly which channels this all happens to on a cue-to-cue basis.

No, the device ID is separate for each console, any of which may have up to 128 individual concurrent cue lists active. Our system supports this.

We normally arrange each cue list in a completely numerical order. The only way the user can call up cues in a random numerical order is through this MIDI facility. (Actually, Det Norske Teatret has a custom made touch screen control system which allows random access through RS232, but this is not available to anyone else unless they pay royalties to them). The MIDI command to be used would be GO or LOAD and then GO.,

Finally, I also think CUE_PATH may be the most likely to be unused, or at least misused, command. At the moment, though, we feel it is necessary but if everyone were to insist it be removed, we could probably effect a work around by using an extension to OPEN_CUE_LIST. But I really don't agree with lighting people telling sound people what they should use or not use.

Hope this does not sound too belligerent, I can only argue for what I know must get done, one way or another. I also believe I am a reasonable person, provided one argues with me until I see the error of my ways!

I hope you understand my point of view and do not find it illogical. Please continue to pursue the goals you wish to obtain and together we'll make this right. Thank you, Charlie.

---[0064]--- (pref = [0063])

[0065] (5 lines) CRichmond.USITT 09/18/90 0132.6 mst Tue midi
Subject: Out of town

I will be out of town from the 19th through the 30th of September. We are showing our wares at the AES convention in Los Angeles so come by if convenient. If not, I'll meet with you again here on my return!

-Charlie.

---[0065]--- (nref = [0066])

[0066] (33 lines) CRichmond.USITT 10/01/90 0020.3 mst Mon midi

Subject: Re: Out of town

Well, that was quick! Maybe nobody noticed. Had some good feedback at the AES regarding the proposal. It was suggested that we provide the capability of recognizing errors and acknowledging transmissions.

I would like to propose that in the next update of this proposal we accommodate these requests. The method I would suggest is to create a new delimiter character, say 01 hex which could be placed at the end of the data being transmitted, no matter how many fields or 00 delimiters it contains. Following the 01 delimiter would be a checksum value which## would be calculated for the entire preceding message data. The received data could then be ignored or stored for future comparison by the controlled device. The controlled device could then acknowledge the transmission with an "error detected" message and then request retransmission if desired. Even when correct data is received, we could define "successful reception" messages which could be sent as a handshake.

In general, my major concern with these processes is that it tends to bog things down quite a bit, although I can certainly understand the need on the part of many to ensure accurate data transmission. The only thing that has to be is that all variations must be compatible for this to be a good standard. Also, the basic requirements of MIDI are that it be capable of operating open-loop and therefore not require handshakes. Basically, making the checksum and acknowledgement optional would meet this requirement since a wide variety of equipment with different levels of sophistication can communicate. The options would be usable, but would be ignored if not supported. Specifically, if a retransmission is requested but the sender does not support the command, the system should not hang. Also, a sophisticated controlled device should not hang if it cannot get a response to its handshake. ..etc.

Comments??

-Charlie

---[0066]--- (pref = [0065])

[0067] (95 lines) Clinton.Thea 10/01/90 2114.6 mst Mon midi

Subject: Forwarded from the Usenet (rec.arts.theatre)

Date: Monday, 1 October 1990 19:25 mdt

From: b-davis at CAI.UTAH.EDU

To: stagecraft at CAI.UTAH.EDU

Errors-To: stagecraft-request@cai.utah.EDU

Precedence: bulk

X-Envelope-To: Clinton@UNCAMULT.BITNET

Redistributed-Date: 1 October 1990 22:14 mdt

Redistributed-From: Tim Clinton <Clinton>

Redistributed-To: {forum >udd>Thea>Clinton>meetings>ascii},
{forum >udd>Thea>Clinton>meetings>midi}

Redistributed-Comment:

I'm forwarding this message that came through the stagecraft

listserver to both the midi and ascii meetings since I thought that both these places would be interested in this. Make of it what you will, and note the electronic addresses.

This was on the Usenet. I think this may be of some interest here.

>From hellgate.utah.edu!cs.utexas.edu!samsung!crackers!transfer!
lectroid!STRATUS.COM!csa Mon Oct 1 19:21:10 MDT 1990
Article: 890 of rec.arts.theatre
Path: hellgate.utah.edu!cs.utexas.edu!samsung!crackers!transfer!
lectroid!STRATUS.COM!csa
>From: csa@STRATUS.COM (Chris Arthur)
Newsgroups: rec.arts.theatre
Subject: computer-controlled lighting
Keywords: computer pc lighting Complecs
Message-ID: <1990Sep28.083337@STRATUS.COM>
Date: 28 Sep 90 12:33:37 GMT
Organization: Stratus Computer, Inc.
Lines: 76

I am developing a computer-based lighting system for theatre and concert use. The software runs on IBM PC compatibles, is written in Turbo C/C++, and will be distributed under the GNU license (which basically means that I will be giving away the source code). It currently requires the support of a fairly simple hardware device that plugs into the RS-232 port on one end and the dimmer packs on the other end; I intend to support the various RS-232-based digital control protocols as well, if I can, so that no custom hardware will be required. In any case, schematics will be made available for the hardware as well.

An early beta version of the software was successfully used to run all of Worcester Polytechnic Institute's New Voices 8 production this past spring; we expect to have the (approximately) final version available for beta testing in early 1991, and to be generally released around spring or mid-1991.

Highlights of the software include: number of scenes, cues, etc. effectively limited only by available memory (and the next version may lift even that restriction), a full-scale compiler to support a lighting control language (based on C and Modula-2), support for "bumps" via the function keys, various capabilities equivalent to the chase sequences supported by most computer lighting boards, and a real-time interrupt-driven kernel that ensures that cues and lighting programs continue to run smoothly regardless of what the operator is doing at the time. And, of course, you get the source code. The name of the system is COMPLECS (an acronym for COMputer Programmable Lighting Effect Control System). Complete documentation will be included, along with a complete on-line context-sensitive help system, and no more than elementary computer experience should be required to operate the system (although programming experience is helpful if you intend to use the compiler, you can do pretty much everything that you could on most computer lighting boards without using the compiler). More detailed information is available.

The main reason that I am posting this now is to see if anyone out there might be interested in using, beta testing, or alpha testing such a system. Also, I would greatly appreciate it if you could send

me (or at least point me to) specs for any control protocols that I can support directly through the RS-232 port (ie, without special hardware). (It might be best to tell me what you can send me before you send it, in case I get duplicates.)

If I am posting in the wrong place, I apologize, and please tell me where to go :-).

Chris Arthur
csa@pennies.sw.stratus.com
csa@es.stratus.com
csa@wpi.wpi.edu
...!uunet!lectroid!csa
csa@wpi.bitnet (may not work)

Chris Arthur
Complecs Engineering
48 Fruit Street, 2nd floor
Worcester, MA 01609
508-792-1585 (10am-10pm Eastern Time)

Brad Davis
Stagecraft List Maintainer
---[0067]--- (nref = [0068])

[0068] (7 lines) CRichmond.USITT 10/02/90 0213.2 mst Tue midi
Subject: Re: Forwarded from the Usenet (rec.arts.theatre)
Which RS-232 based digital control protocol was that? Sorry, but this sounds like a dnf to me.

On second thought, maybe Callboard should start another forum in which contributors can describe new versions of their software and offer beta versions to the professionals of the industry. We could call it Vapourtalk.

---[0068]--- (pref = [0067])
[0069] (220 lines) MDeakin.USITT 10/09/90 1209.3 mst Tue midi

Subject: comments from celco

Hi.
Matt Deakin from Celco in London calling! (At last).
First of all I would like to thank Charlie for keeping me up to date with hard copy, listening to my comments on the phone and for all the work he has obviously put into MSC_0.01.

I have now read ALL messages and the MSC_0.01 document and would like to put forward my comments.

Reading all the messages since this forum started I have felt that there was a risk of losing sight of the objective. But what was the objective?

- A. To produce a full blown show control system so any controller can control any possible piece of equipment.
- B. To produce a system for linking intelligent control systems

so that a master console can trigger slave units to execute pre-defined actions.

C. As in B but also include accurate time based control.

I feel the current proposal is type A. There are so many command types and commands that it will not be possible to keep track of them and keep the data up to date. The General Command section of MSC0_0.01 on its own is more like option C. I feel this is what we should be aiming for. If we do not 'keep it simple' we will end up with a non-standard standard where equipment which should interconnect not doing so.

Remember, the original MIDI spec consisted of little more than note on/off and velocity information with a basic timing system. This has grown since but the fundamental simplicity must have been a big factor in the success of MIDI.

If we can base MSC on the simple concept of commands to GO cues on or of with or without a time this will cover 99% of applications. After all, no matter what your controlling a show is made up of cues.

The SET command will allow more detailed control of specific controls, the MSC equivalent of pitch wheels Etc.

As I see the system will be used to connect intelligent devices together. One or more master or controlling devices will be defined. Each slave or controlled device will react to GO or SET commands as per preset definitions. E.G. GO cue 99 may result in a lighting, sound and videochange. Exactly what happens is up to the designer, why do we need messages specific to video, lighting or sound? The only requirement is that each console can map MSC cue numbers to its internal numbering system. We only need a large range of integer cue numbers 1-999999999... Surely this is simpler than having cue numbers like 333.67.45.99.3!

If there are to be less intelligent output devices, relay boxes, smoke machines etc. these can simply have an address select to define which SET generic control number it responds to.

The Sound Commands of Charlie's seem over complex to me but then I'm not a Noise Boy and don't understand these things!

I know it is a little late for this type of comment as the draft is already written but in fact the parts fully defined are very close to what I feel is needed. All I'm saying is do we need any more and do we need all the Command_Formats and the complex Q_number, Q_list, Q_path notation.

Now for some more practical comments.

1. GO and TIMED_GO commands.

I may have mislead Charlie here with my comments on timed Cues. In our lighting consoles we would normally want a GO ON and a GO OFF command.

These could be timed or un-timed. If we could add a ON/OFF Byte into both of these commands it would suite us. Those consoles/devices that only do cross or move fades or devices where

off is meaningless, like Pyros!, could simply ignore any GO OFF commands.

This would also overcome Anders objection to needing on, hold and off times. On times and off times would be supplied and the hold is the time between the two commands.

It would be nice to have specifically defined codes for instant and default times instead of having to use separate GO and Timed_GO commands.

00:00:00:00:00 would obviously be instant.

25:00:00:00:00 could be use default time.

2. SET command.

I would agree with Anders that some form of convention at least should be established for the basic controls. The ideas he came up with in his message 59 are fine as a starting point. It may be an idea to include some Sequence (chase) playbacks with speed controls, say 16 pairs.

That's all for now.

Matt Deakin.

---[0069]--- (nref = [0073])

[0070] (48 lines) AEkvall.USITT 10/10/90 0027.6 mst Wed midi

Subject: Comments

Anders here

I totally agree with Matt Deakins message no 69. We must try to keep it simple. Remove all the equipment specific codes (CD Players, Flys, Lifts etc.). Controllers responding to GO and SET commands should be enough for all different types of equipment.

Also, I again agree on what is said about the strange cue number method. Simplify! But restricting to integer number is a bad idea. Most lighting consoles use at least one decimal place.

About GO ON and GO OFF

Matt: What is a GO OFF? Most consoles I know have a GO facility to start the next transition. All the times are then programmed into that transition. Why GO OFF?

SET numbers

0-127 Sub masters. 128 should be enough.

128-129 Masters of the first playback. 130-131 Masters of the second playback. ... 191-192 Masters of the 32nd playback. 32 different playbacks should be enough.

193-224 Speed controller for the above 32 playbacks.

225-256 Chase sequence masters.

257-288 Chase sequence speed masters.

512 General speed controller for all fades.

513 Grand Master.

514-545 32 different inhibitive masters? Is there a need for this? We can have 25 different inhibitive masters active at the same time.

I am trying to keep the numbers in groups of 32. This will simplify the binary decoding of the commands. That is also why I put the single numbers (like G.M.) by themselves.

That is all for today.

Anders

---[0070]--- (nref = [0071])

[0071] (29 lines) ABall.USITT 10/10/90 0624.6 mst Wed midi

Subject: Re: Comments

I have been dutifully reading this area for the past few months, and the last two transactions are the first two that make any sense to me.

While I'm no techno-whiz, it seems to me that any "standard" you wish to bring into wide use must be simple, fast and lean. Matt's suggestions above surely qualify under all three of these parameters - even I can understand them (-;...

Without meaning to detract from the extremely difficult and technical work that the participants in this forum have done, might I offer a layman's view of Matt's proposal ?

If I understand Matt's thinking correctly, he is suggesting an approach similar to that found in distributed computing - rather than try to build a huge processor, turning over at billions of MFLOPS per second and housed in an air-tight room, etcetera, distribute the work load across a number of smaller processors, which, in turn are controlled by a central chip (whose function is to coordinate which processor gets which job). The analogy I'm trying to draw is this: rather than building a huge protocol incorporating every possible instruction now and forever, which, by virtue of its enormous packet-size requires and hour per instruction and must be transmitted on 0-nought welding cable (forgive my taking literary licence...(-:), why not leave the machine-specific instructions to the machines involved and allow the protocol to simply control/coordinate their actions ?

Am I over-simplifying this, just deeply confused, or completely "wide of the ma.....that should be "wide of the mark"..... ?

Andrew

---[0071]--- (pref = [0070], nref = [0072])

[0072] (41 lines) CRichmond.USITT 10/11/90 0131.3 mst Thu midi

Subject: Re: Comments

You are absolutely correct and, although it may not exactly seem as though this is what the present proposal embodies, what you have described is, in layman's terms, exactly what its intent is. There is no necessity by virtue of this proposal to have a packet size that will take any larger than a few milliseconds to transmit nor is the

transmission medium expensive or complex (\$50 synthesizers incorporate MIDI).

I'm glad you have simplified it and, although the proposal has a great deal of sophisticated capabilities built in, have expressed the essence of what we are attempting to standardize. The sophistication is completely optional and virtually all of the methods of simplified operation that both Matt and Anders have indicated they want to use can be utilized by simply using the command structure in its most fundamental way.

I fear that the need to create a complete and unambiguous technical specification with flexibility built into it in order that it may accommodate future needs has created the impression that every controller and every controlled device must completely handle all conceivable message concatenations. This is absolutely not the case- in fact I doubt if there ever will be any single device which completely utilizes all possible combinations. But the flexibility within each individual control area is necessary and the possibility of future extensions must be provided for, or else we will have a standard which will not be able to grow.

I have been contemplating a detailed response to both Matt's and Anders' recent comments, but have hesitated to do so until someone else has had a chance to communicate. This has happened and it seems as though there is the misunderstanding that their requirements cannot be accommodated through the current direction. I don't think this is the case, but certainly accept that some fine tuning may need to be done, especially in the area of the lighting commands, in which both Matt and Anders are acknowledged specialists.

In the meantime, rather than continuing unabated in the direction of attempting to defend a proposal singlehandedly, I hope more comments will be thrown into the ring.

Yr fthfl srvnt, Charlie

---[0072]--- (pref = [0071], nref = [0074])
[0073] (85 lines) CRichmond.USITT 10/11/90 0245.9 mst Thu midi
Subject: Re: comments from celco
It's great to have you join us and to get your constructive comments, Matt! I've been studying your message and would like to provide a few initial reactions:

1. I think you have defined very accurately with your type 'B' proposal description the initial intent of this specification. It is this type of control which the most basic use of MSC would provide. Accurate time based control is an optional function which would probably only exist in the more sophisticated implementations. Therefore, type 'C' is what we are attempting to provide for ultimately while supporting type 'B' in its basic form.

There was never any intent to create a type 'A' specification and I'd like to dispel this immediately. There is no implication intended

that the lighting devices using this specification must deal with anything more than the most basic command structures presented and especially not any of the dedicated sound commands. It did, however, seem logical to create a command structure which was simple, logical, extensible and fundamentally similar between all the different command_formats. That is to say that a GO in lighting would be formatted the same as a GO in sound, pyro, rigging or what have you. This means that the options that are needed within one discipline should be at least accommodated within the others, or else we end up with a large number of different rules for structuring commands within the different command_formats.

In this sense, sound is accommodating the requirement of lighting that a timed_go should be recognized even though our cues may contain hundreds of separate transition times relating to different audio channels and effects. As well, a lighting device needs to know in the simplest sense that additional data may follow the basic command and that it should simply ignore it if it means nothing in the context of that show or controlled device.

2. Your comment about mapping master cue numbers to internal cue numbers is well taken. I feel that the current proposal does not preclude this method of operation. If you wish to use all integer cue numbers and to have your console map new cue numbers from master MSC cues, there is no reason why it cannot be done. I do, however, know stage managers and designers who would rather not go through this exercise and prefer to use the numbering system they are most familiar with or which exists within each designer's cueing process. The proposal as it stands has the flexibility to allow all of these possibilities and more.

For example, at the meeting at USITT it was discussed that some manufacturers use cue numbers that contain ASCII characters other than numbers and decimal points. Although the consensus at this point was to allow only these particular characters, since we are using the entire ASCII character range to send only digits and decimal points, there is the possibility that more characters could be allowed in the future (hope this comes out OK).

3. I am slightly concerned about using up too many generic controllers with dedicated type descriptions and also with not allowing enough combinations of command_formats and generic controllers to fulfil the needs of more complex shows. For example, we have systems already in use which would use all the generic controller numbers available for a single addressed sound device (about 16,000). This is certainly not to say that anybody would want to have control of that many controls through MSC, but there is easily the need to pick one of those controls out of the multitude and adjust it when necessary. For this reason, it is important that sound generic controllers be just that and smoke, pyro, mechanical, etc. have their own range of controllers completely separately defined.

4. Again, your comments regarding the standard GO format are well taken. There is no reason that we could not adopt the convention that a TIMED_GO with 00:00:00:00:00 following it could be instant or that 25 hours could use default time. If you then wished to use nothing but this commands for your system, then all the variants would be at your disposal and those manufacturers which do not support TIMED_GO could

simply use the basic GO command. I think one reason we have GO as well as TIMED_GO is that when a number of TIMED_GOs are sent simultaneously, the time it takes to transmit all those bytes may start to add up and we could do it much faster if most of the GOs did not need to be timed.

5. With regard to all the lighting specific comments, I defer completely to the lighting majority concensus and only hope I will be able to understand the terminology in the end!

Thanks again - I hope I have not misunderstood anything you have said and look forward to getting your reply to these ramblings.

-Charlie
---[0073]--- (pref = [0069])

[0074] (72 lines) MDeakin.USITT 10/11/90 1122.8 mst Thu midi
Subject: Re: Comments
Hi.

First I would like to quote part of Andrew's message No. 71. "Why not leave the machine specific instructions to the machines involved and allow the protocol to simply control/co-ordinate their actions?"

That's exactly what I was TRYING to say! Thanks Andrew.

To reply to Anders question about GO OFF.

Our boards and most others in 'good old rock 'n roll' have lots of Sub masters, Pile-ons, additive masters, whatever you want to call them. These are the main controls used during the show. As each of these faders controls a single Cue if we are to trigger them from a GO CUE command this will set the fader to full level. Now how do we get it down again? That's what GO OFF is for. Remember we can have up to 30 playbacks active at a time with autofades. I know GO OFF its not required when using cross fader playbacks or move fades, much as Note Off in MIDI is not always relevant but is still defined and used.

Also on the question of Cue numbering. No matter what we come up with it will be wrong for some types of equipment. The only way I can see of doing this is to assign MSC cue numbers to each internal Cue number. Each manufacturer could have a default set-up but the user will need to be able to change this.

For example. If a Celco board (just for example you understand) is being used to control a colour change controller, one of Charlie's sound controllers and Slide projector controller.

We may want Cue fader 3 on page 14 of the Celco to trigger the following:

Cue 98 on the colour change controller Cue 333.55.765.88.999.0001 ! on Charlie's sound controller Cue 67.04 on the Slide projector controller.

How could any standard numbering system hope to cope with this lot?

However if each unit could internally assign MSC cue 99 to each of it's own cues we would be OK.

Next subject.

Anders: Your proposal for the SET commands looks OK. The only point I would like to clarify is what is the level format for the speed controls and what does it represent?

Does it set an absolute time in MTC format? With a fade time for the playback speed controllers and a step time for the Sequence speed masters. If so how does the General speed master work in relation to these? This would prevent sequences with different times for each step.

Another option is to send a relative scale percentage. So for a playback speed control 50% means go 50% faster than the original fade time. -50% means go at half speed.

This system would allow relative overall time control which would allow controllers which control outputs with very different speeds to be speeded up relative to the original settings. E.G. A moving truss and a laser are very different in their speeds of operation. So if the truss was moving in 30 seconds and the lasers were doing 0.5 second sweeps then slowing it all down by 50% would result in times of 60 seconds and 1.0 seconds.

That's all folks.

Matt.

PS. Charlie I will read your last 2 messages tonight and reply tomorrow.

---[0074]--- (pref = [0072], nref = [0075])

[0075] (59 lines) MDeakin.USITT 10/12/90 1124.2 mst Fri midi
Subject: Re: Comments

Hi.

Having read Charlie's messages 72:73 I have the following thoughts:

I fully agree that the specification should be expandable.

I do not feel that the current spec will not fulfil the needs of lighting, except for a few minor points. My objections are that by defining so many command_formats we will end up with a system where only sound controllers will talk to sound controllers, lighting to lighting, laser to laser Etc. Surely the aim is to produce a SHOW control protocol so different types of controller can interact. As per the example in my last message.

If we spec lots of command_formats with separate GO commands how will you ever get a lighting control board to trigger a moving light cue?

What we need is a set of General commands exactly as in MSC_0.01 that apply to ALL units if the broadcast ID is sent. You can get more specific by using the individual device IDs. As such a huge range of cue numbers is allowed for (120 digits?) there will be no shortage of cue numbers.

This brings me back to integer cue numbers. In a show situation there will be several people programming up different controllers.

The normal approach in the limited time available (there's never enough time!) will be to put all the looks, effects sounds Etc. into each controller and then tie them together afterwards. As no two types of equipment have the same cue numbering system it will be impossible to agree that cue 37 on each controller will be the same cue. Even worse sometimes you will need one button press on the master controller to trigger two or more cues on one of the other controllers (I'm talking about live situations in this case). All this can only be done by having assignable cue numbers. Once you do this they may as well be integers for simplicity.

With regard to your comments about the SET command.

If you have 16,000 controls, of which you way want to control a few by MSC surely this is another example where assignability is required. If anyone has a system of 16,000 controllable 'things', let alone 16,000 on one unit, that they need to control over MSC they will need about two years to programme it up! If however you need to pick a subset of all the controls then this can be done by assigning them SET numbers.

If Anders ideas are treated only as a guide line for the default SET numbering system for lighting controllers then these could be changed to suite each application. This would also help to overcome Charlie's concerns about using up all the SET numbers.

On Charlie's point about Timed_GO and GO. Yes, good point, lets leave them both in.

Cheers!

Matt.

---[0075]--- (pref = [0074], nref = [0076])
[0076] (94 lines) CRichmond.USITT 10/13/90 0157.9 mst Sat midi
Subject: Re: Comments
M Yes - this all sounds quite reasonable. I have a few specific comments:

1. We have tried to keep the machine instructions as generic as possible.

This means that the protocol for each command is the same no matter what command_format is being prescribed. In a practical sense, I think this will only be really useful for programming - both for the people creating the code to deal with MSC and for the people programming shows. i.e. one could create a series of goes for a number of different media by just taking a standard format the and changing the command_format byte for each GO or whatever. I know it looks like we have put in some sound specific commands which may not be needed by anyone else, but I have considered in every case the alternative of trying to make these functions assignable within the system program and have not really been

able to accommodate these requirements. There really aren't very many of them and nobody needs to pay any attention to them for the most part- these extended commands will only be used in the case where there is a

dedicated master show control system running the MSC network and this show controller will be completely oriented to providing full MSC capabilities for all types of command formats. Its purpose in life will be managing other intelligent systems through MSC and presenting a 'soft' user friendly stage manager oriented environment within which to call the show electronically. We would never expect a light board to know sound commands, but on the other hand, we would certainly be able to respond to 'light' commands (read general MSC format) sent under the 'sound' command_format unchanged. (we could also reassign our response table within our program to respond to 'light' command_formats if that's what the user wants to do)

2. You're right about cue numbering. What we've got is a consensus-based compromise. You're correct about needing to be able to assign MSC cue numbers to internal cue numbers and that's something we have not done within our system yet (but don't hold your ...) Just kidding; it is a good idea and should be standard for everyone. If you want to know why we use so many decimal points: it allows a parent/child cue arrangement with several levels of priority; it allows the user to insert virtually any number of cues between any two existing cues; and it allows several levels of nested loops to be run. Macros exist both within MSC (sound sub area) and on our keyboard which allow the operator to jump quickly to previous/next parent cues for purposes of live override of timed sequences and loops. Assigning internal cues to MSC cue numbers will simplify controlling our system from the outside world.

3. I'm glad you feel this will fulfil your basic needs. I'm sure we can fine tune it successfully. Actually, as you may have guessed, my perception of the way this is going is not to have dedicated controllers talking with each other so much as having a single master controller talking with and coordinating all the intelligent dedicated controllers (most of which currently exist, operating within their own little realm). As said before, 99% of the GO's will be simple, without extension and in completely standard format (the same for all command_formats).

One thing that does interest me is your comment about a broadcast ID for all units. The implication seems to be that you want to have an ID that will be recognized by all command_formats simultaneously. Would this mean that you could send a GO CUE 99 (for example) that would be recognized by every single command_format from lights to lasers without exception? This is something that we have specifically avoided because we feel it is fraught with danger: it would be much too easy to hook a pyro system on to a system programmed this way and all of a sudden have pyro going off all over the place where the programmer had "forgotten" that he put in an ALL command_format command. Perhaps I just misunderstood your comment, but one needs to make sure of these things (since we're putting it all down in black and pixel). (We are allowing an ALL device broadcast within each command_format, and nobody has objected to this although I expected someone to do so).

4. Please feel free to use nothing but integer cue numbers and ne'er a decimal point (nor full stop/period) shall cross your digitizing tablet. You have everybody's permission to do this as long as you let us mess up the world with tiny li'l dots.

5. Re all those controls. Yes! Again you're right about assignability and this is a very good suggestion (wish I'd thought of it - but oh

well) Right - controls shall be made assignable to MSC SET numbers. Now, where does that leave the concept of dedicating certain SET numbers to certain types of controls (and making it all lighting specific at that) Maybe it's my turn to complain about things becoming too format specific. No - on the contrary, I have no problem with this as long as the list isn't too long and doesn't leave room for (here's that word again) expansion. I'm not sure what you mean by 'If Anders ideas are treated only as a guide line for the default SET numbering system' - if this is what I think it is then we are talking 'recommended' rather than 'prescribed'

This suits me, but really it's the concensus (another favourite word) falling into disrepute in this forum - where are the multitudes? Hmm perhaps it's just as well - look at how much time this is taking!) By the way - I'm not terribly concerned about us ing up all the SET numbers - who has that kind of live control capability?

Well, it's been a pleasure -- ta ta for now. -Charlie
---[0076]--- (pref = [0075], nref = [0077])

[0077] (18 lines) ABall.USITT 10/13/90 0654.6 mst Sat midi
Subject: Re: Comments
Something else to kick around...why not attach a spare byte onto the end of each packet, to allow the manufacturers to "enhance" or expand the sorts of instructions that may be passed to their device. In this way, you allow "freedom of choice" among different devices. One manufacturer might use the last byte for motion control instructions or Artificial Intelligence ("wait until the central controller gives you a device exclusive, then check device 44 to see whether it has completed its moves and then execute your nest instruction"), or perhaps feedback to the controller ("This device has finished executing its next instruction"). The controller might could be configurable, using specific software (perhaps supplies by the manufacturer) that allow the controller to store one extra byte's worth of instruction per device. This implies some sort of "set-up" rather than a self-configuring system...

That's more than enough. Two cents, if not a nickel...

Andrew
---[0077]--- (pref = [0076], nref = [0078])

[0078] (38 lines) CRichmond.USITT 10/13/90 2355.8 mst Sat midi
Subject: Re: Comments
This is an interesting suggestion. My first reaction was to consider it to be going much too far in the direction of being too device/ manufacturer specific but then I considered the fact that the prescribed format of data transmission says that any data sent after the final delimiter of interest to the controlled device should simply be ignored.

In other words, this means that the current specification supports this suggestion as a future expansion capability, should we decide to implement it. The only catch I found upon rereading the spec is that some of the commands do not specifically allow a null delimiter character (00 hex) to be sent after them. These commands are SET, FIRE, ALL_OFF, RESTORE, and RESET. The way the general description of data transmission is worded in section 2.6 DATA LENGTH, it could be implied

that even these commands could have the null delimiter character transmitted after them and that the correct response by a controlled device should be to ignore them and anything after them. If we change the number of data bytes allowed for these commands to "variable" then we could allow the null delimiter to be sent along with your device specific enhancement (or any other enhancement we might dream up in the future).

SET presents a slightly complicated situation for this suggestion: it may have an optional time sent after the control data and this may contain several 00 hex bytes. The controlled device must assume that any data sent in the 6th through 10th bytes of the actual command will be a Standard Time Specification and that either an F7 or 00 will follow this to indicate end of exclusive or continuation of device specific information per this suggestion. In other words, to implement this proposal with SET, one must send the full optional message length, unless we rewrite the SET specification as it now stands.

If everybody agrees that this is a good suggestion, we need to deal with the rewriting of this area right away, so I would appreciate some feedback.

We're up to two bits now for sure.

-Charlie

---[0078]--- (pref = [0077], nref = [0079])

[0079] (10 lines) CRichmond.USITT 10/14/90 0001.4 mst Sun midi

Subject: Re: Comments

One more thought about SET: we have provided this command with the built in ability to set more than 16000 controls each to more than 16000 values. Furthermore this can be done for up to 127 devices within each command_ format (lighting, sound, etc.) so the possibilities of providing your controlled device with extremely unique responses to this command are not exactly limitless - but very close. At least in terms of the SET command, I think we have already given it a tremendous amount of flexibility.

(opens the cash drawer again)

---[0079]--- (pref = [0078], nref = [0080])

[0080] (4 lines) CRichmond.USITT 10/14/90 0006.9 mst Sun midi

Subject: Re: Comments

North American anachronism explained for our trans Atlantic participants:

1 bit = 12.5 cents

---[0080]--- (pref = [0079], nref = [0081])

[0081] (3 lines) CRichmond.USITT 10/14/90 0011.3 mst Sun midi

Subject: Re: Comments

I have long believed that this bizarre North American valuation for a single bit partially explains the inherent shortcomings of MSDOS. Not to say what it does for music tonality when played through an IBM.

---[0081]--- (pref = [0080], nref = [0082])

[0082] (8 lines) ABall.USITT 10/15/90 1717.5 mst Mon midi

Subject: Re: Comments

Charlie...I have a confession to make...I have not read the spec. Not

one word. I suspect that most of it would be beyond me. I approach your spec from the perspective of the end user. I leave it to you and the experts to sort out the technical problems.

Best of luck.

Andrew

---[0082]--- (pref = [0081], nref = [0089])

[0083] (44 lines) AEkvall.USITT 10/16/90 0022.9 mst Tue midi
Subject: Comments...

Hi,

Comments on Matt's messages 74 and 75.

A reflection to Matt's explanation of the GO OFF command: Why don't you use the SET command to control the faders. In this way you can easily get them both up and down. The GO command is more suited to start cues in a more crossfade/sequence oriented way. We will use the SET command for controlling our masters which is a fundamental part of our systems too. Crossfades will be started with the GO command.

I totally agree on your argumentation about cue numbering. Integers should be enough. Eventually one decimal place could be allowed for this to be able to insert new cues in between. You will always need to have a possibility to map global cue numbers to internal ones.

Speed Control: Relative percentage is what I thought of but didn't say.

Comments to Charlie's message 76

Point 5: Of course it should be possible to reassign the SET numbers in the controller. It is OK to say that some numbers is recommended for different uses. They should be given as default numbers for the functions that match any of the functions described. This means that the basic hookup of a system has a fair change of working decently without having to do a lot of setups. However, if the user wants to change the assignments, he should be free to do so.

Comments to Andrew's message 77

No, I think this will complicate MSC too much. Why don't use a normal manufacturer dependant SysEx message for sending this type of special information? Each manufacturers own MIDI ID could then be used. Manufacturer specifics should not be included in standard messages. I am totally against this. If there is a strong demand for this, we could make a special message type within MSC for manufacturer specifics. A SysEx within a SysEx!

And the work goes on....

Anders

---[0083]--- (nref = [0086])

[0086] (27 lines) ABall.USITT 10/16/90 1734.3 mst Tue midi

Subject: Re: Comments...
\Anders, mi

Maybe I misunderstood you, but I don't see why these "generic bytes" have to be specific to a manufacturer - indeed, they should *not* be. If what you are proposing is allowing each manufacturer to provide a manufacturer specific MIDI ID that could be used by the protocol, sure I like the sounds of that. How does that preclude the use of these extra information byte(s) ? I would

As for a "manufacturer dependant SysEx" ...if I knew what the heck it was, I might agree with you. (????)

Once again, I'm not talking about two byte of anarchic madness within the packet - sorting out who has control of those two bytes would be a problem, for example. Just allow those two bytes to act as "breathing room" for extra, device specific instructions, as in byte 1 = "Hey ! Device 4567! Here comes a specific instruction", byte 2 "Instruction 22", which device 4567 interprets as "Turn on the coffee maker !".

One last thought - is their a simple checksum included in each packet to verify data integrity ?

Brain swelling, but undaunted, I remain,

Andrew
---[0086]--- (pref = [0083], nref = [0087])

[0087] (7 lines) JLBracewell.USITT 10/16/90 2009.1 mst Tue midi
Subject: Re: Comments...
Andrew: Let me take a shot at giving you that explanation. The SysEx, or System Exclusive, command in MIDI is a reserved area of the protocol that lets a manufacture register a command that only his products will respond to. If I remember correctly from the beginnings of this debate, the whole Theatre Messages section of the MIDI protocol will operate under the SysEx section of the specification. Charlie, Matt, Brad, or Anders will correct me if I'm wrong, I'm sure.
---[0087]--- (pref = [0086], nref = [0088])

[0088] (14 lines) CRichmond.USITT 10/17/90 0026.4 mst Wed midi
Subject: Re: Comments...
John, you are indeed correct. The fact is that SysEx has grown like topsy and now really covers a vast amount of area. It was originally designed to be just for manufacturer specific commands but the spec basically needed to be expanded and they decided to create "Universal System Exclusive" which had its own "SysEx" header. Really, this should not have even been called "System Exclusive" because it was destined to contain many standard commands that everyone should be able to deal with and contains no manufacturer specific data (other than raw data which is being trans mitted via the file transfer protocol - but even this is usually stuff like sample data and follows some other standard format.)

Anyway, enough. Our proposed MIDI Show Control spec falls under this heading of Universal System Exclusive and actually is under the further sub-division of "Real-Time". Don't ask.
---[0088]--- (pref = [0087], nref = [0090])

[0089] (7 lines) CRichmond.USITT 10/17/90 0031.8 mst Wed midi

Subject: Re: Comments

Well... The spec is lurking here in the common>standards database if you want to read it (it does not have to be read in detail to glean the basics, so I recommend it). Also, it would be very useful to read the MIDI spec or some other explanation of it. Thanks for your suggestions- we need end user input.

-Charlie

---[0089]--- (pref = [0082])

[0090] (28 lines) CRichmond.USITT 10/17/90 0045.9 mst Wed midi

Subject: Re: Comments...

Anders, I do not feel like arguing one way or another on the cue numbering situation - my understanding of why we went the way we did has already been explained. I'd like to get some feedback on this from several of the people who originally argued at the USITT conference for decimal points (and advocated additional ASCII character transmission capability). Because of the assignability situation, I agree that we could deal with integers only (but having a cross reference chart would be very cumbersome) I still feel that if you want to use integers in your world, this spec allows that very nicely, while allowing others to use what they are used to. (Remember, our system is not operating in a vacuum, either).

Anders, re your comment on Point 5: I think you mean to refer to reassigning SET numbers in the controlled device rather than at the controller. It could be possible to allow the controller to send different SET numbers than the one ostensibly being controlled but this means two sets of conversions (one at the controller and potentially another at the controlled device). A moot point perhaps, since we cannot prevent manufacturers of controllers or controlled devices from reassigning SET numbers for convenience or necessity, but still it would be good to try to define a useful range of standard SET numbers that accommodates future expansion. Anyway, I think this is what you mean, anyway.

I do think, also, manufacturer specific data should be kept where it belongs: in their own sysex area. Mfr SysEx within MSC - yikes!!!

next..

---[0090]--- (pref = [0088], nref = [0091])

[0091] (8 lines) CRichmond.USITT 10/17/90 0052.2 mst Wed midi

Subject: Re: Comments...

Again, the MIDI spec allows manufacturers to send really specific instructions within their own SysEx ID number quite nicely and this is where turning on the coffee maker should reside. See previous reply for details.

Also, see recent comment on checksum: this is being considered as an enhancement that does not have to be supported (for potential economy/speed sake). Thanks again.

---[0091]--- (pref = [0090], nref = [0092])

[0092] (15 lines) JLBracewell.USITT 10/17/90 0924.1 mst Wed midi

Subject: Re: Comments...

I'm going to throw my "two-bits" worth in on this even though I haven't gone back to check older transmissions to see what's already been said. I wasn't able to stay all the way through the discussion in Milwaukee, but I think I know what the lighting people would have said. I'd have to argue that at least one decimal place in cue numbering makes the insertion of cues much easier. I've never seen a time (in lighting, anyway) when it wasn't necessary to insert at least one cue into the sequence. Unless we're all going to use somekind of integrated software that will renumber cues as integers FOR ALL APPLICATIONS CONCERNED, it's going to be a mess if we have to deal with only integer-numbered cues. Either that or we have to have the ability to link, and that could be as big a mess! For the sake of everybody's sanity, leave us the ability to insert fractional-numbered cues!

(Or am I missing something obvious?)

---[0092]--- (pref = [0091], nref = [0093])

[0093] (6 lines) ABall.USITT 10/17/90 1001.5 mst Wed midi

Subject: Re: Comments...

Charlie,

Thanks for the patient explanation. Mr. Bracewell is also owed a debt of thanks. Time to sit on the sidelines again. Thanks.

A.

---[0093]--- (pref = [0092], nref = [0094])

[0094] (113 lines) MDeakin.USITT 10/17/90 1113.8 mst Wed midi

Subject: Re: Comments...

Hi.

Charlie: We seem to be talking a cross purposes slightly. I understand that you will normally run a system consisting of a master show controller computer. If I understand MIDI correctly this will be similar to using a sequencer for music applications.

For Live use of MSC I would expect the configuration to be more as I described earlier with one Lighting (say) controller triggering a number of other controllers when the operator thumps the buttons! As I see it this is equivalent to having a couple of synths and expanders linked up. This is why I want a simple assignable, and YES global set of GO commands.

The big difference is that with a central controller you will programme up on that which cues on which unit of each type will be triggered at a specific time. For this arrangement the spec as written is fine. You can specifically address a number of devices of each type.

In the live set-up it should be possible to set-up simple assignments between internal cues and MSC cues. So if the internal cue is assigned an MSC cue number it will always trigger on that MSC cue. If there is no MSC assignment the internal cue will never be triggered via MSC. This will allow there to be operators on all the consoles operating manually but allow some cues to be mastered via MSC.

As there will normally be only 2-6 controllers interconnected there is no need for individual type or ID addressing.

Here comes another example.

1. Master controller console defaults to MSC cue number 1-999 being assigned to internal cues 1-999. The Master will send an appropriate GO ON and GO OFF when each internal cue is activated or de-activated.
2. Slave number one is programmed up as normal. Then each cue that is to be triggered from the master console has the appropriate MSC cue number assigned to it. E.G. Internal cue 56 is to be triggered by Cue 99 on the master controller. So MSC cue 99 is assigned to the internal cue 56. and so on.
3. The same is done for the second slave controller.

This approach allows the controllers (any one of which could a master) to have very simple MSC interfaces. Otherwise you would have to be able to assign multiple GO <command_format><ID><cue No.> to each internal cue.

Another advantage of course is speed. One message to trigger a number of slaves instead of one message per slave. This is not insignificant because any delay over 20mS becomes noticeable when you press a button and expect a light to come on.

One slight enhancement on this would be as well as having a broadcast ID and command_format you had say 15 broadcast channels as in MIDI. This would allow several masters to control subgroups but still allow an overall broadcast message to potentially control all units.

So here is my proposal.

Command_formats.

Define command format 7FH as ALL TYPES.

Any unit that has a matching device ID (see below) will respond to a message with this command_format. Only general commands should be sent using this command_format.

Device IDs.

Let 1-6FH represent individually addressed units for a specific command_format.

Let 70H-7EH represent Channels 1-15. These can either be channels within a specific command_format if command format is set. If command format is 7FH then they are universal channels controlling units of different command_formats.

Let 7FH be a global message which should be received by all units of the specified command_format or by ALL units if the command_format is 7FH.

Integer cues, AGAIN!

OK. You have your dots and I'll have a dot free world. However it would be more compatible if all broadcast or channel messages only used integer cue numbers and dotty cues were reserved for device specific messages.

Proscribed v recommended.

Does anyone have any comments on this? Obviously if we can have broadcast messages then there is a risk of interaction if SET commands are sent. However if SET numbers can be assigned internally and there is only a RECOMMENDED default for each controller type then it would be possible to directly inter-link similar controls on different types of console by editing the default assignment. E.G. The sequence speed control on a lighting board could control the sequence speed control on a laser controller.

Right that's my tuppence ha'penny's worth.

Matt.

Comments on the last 10 messages coming up tomorrow.
---[0094]--- (pref = [0093], nref = [0095])

[0095] (22 lines) RThomas.USITT 10/17/90 2151.6 mst Wed midi
Subject: Re: Comments...

Two thoughts: The decimal system for numbering cues that has been proposed is indeed cumbersome to look at and get used to. However Charlie is absolutely right in suggesting that it does allow nested loops and infinite insertion of cues--something that no other system really does. The problem seems to be in the size of the show you are running. If you need to run 1,000 cues in 20 minutes, I really haven't found another cue numbering system that will do it. If you need to run 20 cues in 1,000 minutes, there are a lot of systems that work great. Perhaps the difference in opinion expressed here has a lot to do with each end user's expectations of how they would use the spec? This will probably sound silly, but if the decimal system proposed is implemented, is it being suggested that manufacturer's won't use it because they will have to write their own translators to convert the numbering system into whatever they want their equipment to show? Also, please assure me, Charlie, that you are not seriously directing the specification towards a SINGLE MASTER CONTROLLER. The theatre of the future is full of artists that will wish to perform together both on stage and off. There will certainly be many situations where a single master controller will be necessary, but what about the "Cantata for Woodwinds, Fresnels, and Air Casters" some arT}IKTSQ (sory the computer just puked at the very thought of this) some of my more exotic friends are insisting that I get right to work on? Guts to go...

---[0095]--- (pref = [0094], nref = [0096])

[0096] (57 lines) CRichmond.USITT 10/18/90 0003.1 mst Thu midi
Subject: Re: Comments...

Rick- you've finally taken the plunge! I've been waiting for you to start on this work of a lifetime for so long I thought you'd never do

it. The only thing was I thought it was entitled "Cantata for Brass, Lekos, and Flys". Oh, well all we have to do is change a few command_ formats and we'll have whatever we want.

Seriously, the concerns you raise (cue numbering and master control) are all symptoms of the same thing: wanting to make a single standard be everything to everybody. Really, in 95% of the cases, the situation will be like Matt describes: several control boards with live operators and linked via MSC so that a few specific integer MSC cues can be passed around from unit to unit to provide an extra measure of coordination assistance and synchronization. It's only in the more exceptional situations that decimal point cues will have to be used and certainly in even fewer situations will a single controller be used. Rest assured _I_ am not the one attempting to define a show situation in which only one controller is to be used: however, we have customers whose fondest wish is to have everything (read: _everything_) controlled by a single control system if not a single controller.

We are currently designing a system for an application in which a single controller will be controlling the following all simultaneously as part of a single "show": 60 floats each with a light and sound control system controlling up to 10 loudspeakers, hundreds of lights, pyro effects, mechanical effects including animated figures (the control signals for each float are programmed into an on-board controlled device which receives MSC via radio); several dozen lasers, several dozen automated highpower searchlights on lifts and with hydraulic control systems, thousands of regular lighting instruments, pyrotechnic controllers, multiple synchronized fireworks controllers, fixed mechanical and animated effects plus (of course) several hundred loudspeaker zones each with programmable audio source capability from several dozen audio sources including digital audio multitracks, samplers and EPROM effects. All this is spread evenly over many acres of themepark and will control the general park lighting and background music as well as synchronizing with live actors, dancers, and other action as necessary. We want MSC to control this because we want to use as much standard off-the-shelf equipment as possible. The sooner this standard is established, the sooner standard equipment will talk MSC and the sooner we can do the above without having to completely reinvent the wheel.in

Of course, most applications are not this comprehensive and need not bother with dotted cues, cue_paths, cue_lists, complex master controllers and all that other stuff. But what I think is exciting about this spec is that it can easily meet the needs of the local community theatre with a little ambition and a willing young computer literate student (or maybe not even that- hopefully it will be understandable by complete technical novices - certainly this is the object) and at the same time be able to run a show such as I outlined. I don't think there is anything described in the above show description that cannot be handled by MSC as it is now proposed (hopefully we'll have the chance to find out - soon).

Resting easier? Hope so.

-Charlie

---[0096]--- (pref = [0095], nref = [0097])

[0097] (85 lines) CRichmond.USITT 10/18/90 0053.7 mst Thu midi

Subject: Re: Comments...

Re: Matt's transaction, message 94 -

I don't think cross purposes as much as different applications perhaps. Many times we will be using a master show computer for most cue operations but at virtually all times, there will probably be a live operator at least available to override or assist with cues on each individual controlled device. These will include totally live theatrical applications as well as completely pre programmed shows. The concept is that the stage manager is calling the show from an electronic desk using whatever they are comfortable with: keyboard, mouse, tablet, touchscreen, whatever. This is connected to the master show controller which is running an MSC program which, in our case, is called Stage Manager. The idea is not that the stage manager replaces the board operators but merely is able to push a button and receive electronic confirmation when they want to call a cue. This simply makes the old process of saying "Warning cue xxx, Standby cue xxx, Cue xxx - GO" basically obsolete. If the stage manager is really the one who decides what cue and when it should go, the process should be as direct and foolproof as possible. If it's the board operator that makes the decision, that capability certainly is not taken away!

For the majority of applications, I think the scenario you outline is entirely logical with one lighting controller talking to several slaves.

One question: in general, what is the difference between having a global command_format capability and simply assigning several lighting boards to the same device_ID with a different map of MSC cues to internal cues? It seems to me that if you did not want a board to respond to an MSC cue you could just not have any internal cue respond to it. This would be much the same as sending a non-global GO in order that specific boards would not respond. This could even be more flexible in that to a single MSC cue (short transmission time) some boards could respond with various internal cues (as programmed) and other boards would not respond.

Maybe I'm not understanding the objective of a global command_format.

Referring to your example, I think this would work fine under the current arrangement if every controller, master and slave were simply set to the same device_ID number. Slave number one would then have internal cue 56 be triggered by MSC cue 99 as you described. Other assignments would be done on all slaves so that each internal cue either was triggered by one, none or more MSC cues. Also you could consider setting up your consoles to be able to transmit and/or receive cue GOs selectively not only on the basis of a MSC to internal cue number table, but also on the basis of multiple device_ID's. In other words, you could program a slave to have cue 56 go when MSC cue 99 for device_ID 3 was received but have cue 78 go when MSC cue 99 for device_ID 4 was received. Then, going another step further, you could have cue 43 go when MSC cue 99 for device_ID 4, command_format 03 (colour changers) was received. If you explore the flexibility built into the proposal in terms of internal reassignment of data on a cue-to-cue basis, quite a lot can be done.

The capability of having a variety of different internal cues respond to the same MSC cue reduces the need to transmit many different MSC cues to get a lot of devices to respond.

Re the specific proposal:

I think I understand your proposal and don't actually have any complaint with it other than I think it is probably unnecessary but does add some flexibility.

Question- I don't really understand the difference between "Device_IDs 1-6FH and 70H-7EH. To me they seem like they would do the same thing. All devices responding to the same device_ID and command_format would respond to the same message. This would work like a "Channel" no matter what the device ID number is, from 01 through 7E, so I'm not sure of the distinction.

RE SET recommendations: by definition, MIDI is a recommendation. We don't have the ability nor the desire to police manufacturers' responses to MIDI commands. The marketplace decides whether a particular approach is a viable one or not. By all means, makes the responses to SET as programmable as you wish. If you make it easy to program and understand what you've done, your customers will love you for it. If not, well....

What you suggest seems fine regarding having the ability to make one controller of one type control one of another. You'll have to give the user the ability to define the response range as well. And the control law/slope, too.

Where'd you find a ha'penny - haven't seen one of them in years.

Regards, Charlie

---[0097]--- (pref = [0096], nref = [0099])

[0099] (102 lines) MDeakin.USITT 10/18/90 1127.5 mst Thu midi

Subject: Re: Comments...

Hi.

1.

In reply to Ander's message 83.

In our consoles and many others of the same type there is no crossfade playback facility. If we do not allow the Sub Masters to be triggered and reset by GO ON and GO OFF commands we would not be able to respond to GO commands at all.

Another point. If on a theatre console you have two crossfade playbacks how do you know which one to use for a GO command? As I see it you can only do crossfades on a single playback because the GO command must remove the previous cue and replace it with the new one. This means you can only ever have one cue active at a time. Or have I missed the point.

I would have thought that not all the different types of controller MSC will operate have a single crossfade type playback. If this is the case then we need to be able to activate and de-activate multiple cues on a controller.

I feel there is some confusion over what the GO command and the SET command will be used for. Are we saying that the GO command is only for

use on a single playback per controller that will always replace the last cue with the new one? The SET commands are then to be used for all other types of operation?

My feeling is that the GO should be used as the main triggering command and allow multiple cues to be active at one time by defining GO ON and GO OFF commands. Then the SET commands are used for secondary functions like speed control and interlinking controls on similar types of equipment. (We can use RESET to turn off all active cues or define GO OFF 00 as all cues off).

I'm not saying you could not control sub masters via SET commands Anders, the difference is that on your consoles Subs are a secondary control where on ours they are the main playbacks.

2.

NOT INTEGER CUES AGAIN!

Yep!

To take the example of MIDI again, the operator is not normally aware of the note numbering system. When the 'musician' hits he doesn't know or care that it's MIDI note number XX. This should be the same in MSC, If the operator calls up cue 78.55 on the master controller he need not know that this will send MSC cue 99. This gets round the problem of inserting cues.

Here comes another example.

Master controller MSC Cue No. Slave 1 Cue No. Slave 2 Cue No.
Cue Number.

78.55	99	78	5.2
79.00	2	77	88.4
79.5	976	9	43.0

Now insert a cue on the master controller between cues 79 and 79.5.

Master controller MSC Cue No. Slave 1 Cue No. Slave 2 Cue No.
Cue Number.

78.55	99	78	5.2
79.00	2	77	88.4
79.25	977		
79.5	976	9	43.0

The next free MSC cue number (977) can be assigned to the inserted cue and now any cues in the slave controllers can be assigned to MSC cue 977.

It may sound complicated but in lighting at least its no different from dimmer patching.

3.

SysEx in MSC. YUCK!

4.

Checksums.

I'm not against the idea but MIDI gets along OK without them and they do take up time. Without a return path the only thing a unit can do if it gets a dud checksum is throw the message away. This is better than firing off the wrong cue though! So I'm happy to make it optional.

TTFN.

Matt.

No time to reply to the last 3 messages now, off to the pub!

---[0099]--- (pref = [0097], nref = [0100])

[0100] (11 lines) ABall.USITT 10/18/90 1419.8 mst Thu midi

Subject: Re: Comments...

The chance of a controller misfiring, given the scope of devices Charlie is proposing to control, warrants careful investigation of checksums. I have played some awfully small MIDI synth set-ups and there is nothing quite as disturbing as having a power spike send an ALL_NOTES_ON, requiring me to pull the plug in order to shut the silly things off.

Ask some musicians - they carry little black boxes with BIG red buttons which send ALL_NOTES_OFF when pushed.

There. That's enough.

A.

---[0100]--- (pref = [0099], nref = [0101])

[0101] (52 lines) CRichmond.USITT 10/19/90 0036.8 mst Fri midi

Subject: Re: Comments...

High. (This is in acknowledgment of Matt having departed to the pub after sending the last message)

1. I've been secretly wondering in my own quiet (high signal-to-noise ratio you understand) way when this subject was going to come up. Could I suggest a method of defining which playback facility (submaster, cross fade playback, what ever you want to call it) you are making the specified cue GO to (become active on, controlled by or whatever)? If you look at the proposal, following the Q_number specification there is an optional 00 hex delimiter which can then be followed by an optional specifier called Q_list. In our system, this Q_list data actually specifies in which "play back" position the cue GOing is placed. We don't actually put a whole cue on a single fader as is done in lighting, but the situation is analogous. Perhaps you could use Q_list to define into which cue playback location the desired cue is to appear for the operator.

To go one (perhaps unnecessary) step further, the next specifier "Q_path" additionally prescribes from which separate storage area the requested cue is derived or loaded. This is to both allow mass storage

of cue data as well as being able to differentiate between cue 79 in path 3 and cue 79 in path 23. Why this would be of use to the average lighting plan, only imagination could tell.

Of course, as you suspected, controllers could also come into play here in some obscure and yet undefined way. I do agree with your statement that the GO command should by itself be the main triggering command and that is why these multiple sub-data delimiter areas have been included: to ensure that all parameters regarding each cue GO can be defined with the command.

2. Yep! I fully comprehend your instructive demonstration of the complexities of multiple renumbering procedures. If there is a consensus, this can be the way it is to be done. Somehow, I suspect most users will not want to go through this complicated process and I, for one hope to be able to have the same number represent the same cue calling function at every place in the communication chain. The amount of time saved with this capability seems to rank up there with the advantage of having the wheel. Honestly, I am not trying to be petulant here - it's just that it seems easier for us to do things the way we had planned (logical, eh?) just like it seems easier for you to do things the way you're used to doing them. A compromise really seems to be to go with a system which allows both integer and non-integer so we can both use the same standard without compromise. And that does seem like a pleasant thing to do. This is probably not the last thing I'll say about this but it is the last for now.

4. Checksums look like they'll definitely be included as options in the next version of the proposal. Why not?

One more message then off to bed!

---[0101]--- (pref = [0100], nref = [0102])

[0102] (25 lines) CRichmond.USITT 10/19/90 0050.9 mst Fri midi

Subject: Re: Comments...

Andrew- I agree with the advisability of checksums, but interestingly we probably won't use them soon. In our experience with MIDI, most of these problems have been entirely attributable to poorly designed equipment, a ground loop or some other system fault and not with MIDI itself. In fact, because MIDI by definition is electrically isolated between devices with an opto-isolator, it cannot be responsible for such system problems. The electronic design of MIDI communication is extremely reliable, using a low-voltage current loop with snubbing diodes and large measure of both timing and current error tolerance. Also, only one receiver can be connected to each transmitter so fan-out is not abusable (new word for the spell checker, folks) and the better systems reshape, reform and retime the signals each time they are regenerated and retransmitted.

In actual use, we have MIDI running video disc players (4 of them) with a cable length of 10 metres, and looping on to a bank of digital audio delay/reverb units in a show which runs an average of 30-40 times a day. There has not been a single fault attributable to bad MIDI data since it opened at the beginning of June this year and it has run like this every day. So that makes it no errors in about 2-3 Megabytes since the average length of each cue is 10-15 bytes long and there are about 150-200 MIDI cues in the show. Not fantastic statistics, so far, but that's what we know about firsthand. We still share your concern.

-bedtime for Charlie

---[0102]--- (pref = [0101], nref = [0103])

[0103] (5 lines) ABall.USITT 10/19/90 1412.0 mst Fri midi

Subject: Re: Comments...

Good to know Charlie. Nothing like "ackshul use under ackshul condishuns" to verify the "bomb-proofness" (put THAT in yer spell checker...) of a protocol. Thnanks.

A.

---[0103]--- (pref = [0102], nref = [0104])

[0104] (3 lines) CRichmond.USITT 10/19/90 2336.5 mst Fri midi

Subject: Re: Comments...

And Thnanks to you, too!

- Charlie

---[0104]--- (pref = [0103], nref = [0105])

[0105] (25 lines) MDeakin.USITT 10/26/90 1025.5 mst Fri midi

Subject: Re: Comments...

Hi.

In reply to Charlie's message 97.

My idea for the device ID's is very similar to your example of having slave units respond to multiple IDs. But instead of setting a slave to respond to 2 or more specific IDs you could set a unique ID and a Channel ID so selecting groups of slaves by setting there channel address the same but still allowing unique addressing as well. This really applies more to systems controlled by a central MSC computer. This would give three modes of addressing, individual, by Channel grouping or global. This is all getting a bit involved but it would help keep the number of messages down.

The main idea was to allow a very simple global addressing system that as a default would allow two or three units to be plugged up with the minimum of complication. By using a global ID and command_format the user only has to assign simple cue numbers to get the thing going.

Nice and short this time.

Bye.

Matt.

---[0105]--- (pref = [0104], nref = [0106])

[0106] (26 lines) CRichmond.USITT 10/27/90 1520.8 mst Sat midi

Subject: Re: Comments...

Thanks for the explanation, Matt - I understand the application and it seems as though the channel method might increase flexibility. Perhaps we could have the best of both worlds without changing the spec by setting up masters and slave so that they use the ID's you have suggested as a "channel" type of response and also be able to use individual ID's as well.

In a sense, what could happen is that the ID numbers could be used for multiple, channel, or what-have-you in various combinations as determined by the built-in default capabilities of the masters and slaves or user programmed if necessary. This means you could set up your systems to user (use) the ID numbers as individual, channel, or global as you have suggested until we can arrive at a concensus here. Just set things up so they can be changed easily if the standard becomes slightly different.

In a sense, having the ability of the master to send messages on several ID's and the slaves to receive from several ID's, all on a cue-specific basis will give you this capability with even more flexibility but possibly without the ease in quick plug-it-in-and-it-works functionality. But really, this is more a result of how user-friendly the system is rather than exactly how the spec is written. At any rate, the spec does allow us to do this one way or the other at the moment. Please correct me if I'm wrong.

- Charlie
---[0106]--- (pref = [0105], nref = [0107])

[0107] (45 lines) MDeakin.USITT 10/31/90 1239.2 mst Wed midi
Subject: Re: Comments...
Hi.

This is going to be a very 'agreeable' message.

In reply to Charlie's message 101.

1. I agree with your point about using the Q_List defined after the Q-Number to select the playback. Our system only allows certain Cues to be put on each playback but we can cover all this in our documentation.

2. I agree completely that the ideal solution is to have all units operate the same Cue numbering system. However the practicalities of different operating practices and the vast range of equipment we hope will implement MSC will make a patching arrangement necessary. In the long run as new equipment is designed to operate with MSC the Q_numbering conventions way converge on a standard. However at first with MSC interfaces being bolted onto existing equipment we will need to patch the MSC Q_Numbers to the internal Cues.

As for allowing both Integer and Dot Q_Numbers I agree. (I said this would be agreeable).

The only remaining problem from my point of view is how to handle the Go On and Go Off problem. There are two solutions I can think of. One is to define a Go Off message with the same Data structure as the Go On message. The other is to define another byte in the Go message. We define an optional byte to follow the Q_Path, delimited by another 00. This could be used as an on/off switch, if omitted the default would be On.

The disadvantage of the second option is that even if you did not need Q-List and Q_Path data you would have to send these to allow you to send the ON/OFF flag. One way round this would be . This could then follow

the Q_Number if no Q_List or Q_Path was required.

On balance I prefer the GO Off option because it gives a slightly shorter message. As I see this being used when the LD is bashing away on his flash buttons at 5 beats per second every millisecond counts!

Well you didn't expect me to agree to everything did you?

Matt.

---[0107]--- (pref = [0106], nref = [0108])

[0108] (40 lines) CRichmond.USITT 10/31/90 2357.3 mst Wed midi
Subject: Re: Comments...

Oh _very_ agreeable, definitely! I must admit I have not really been following all the subtleties of the discussion on GO On and GO Off since my attitude toward it was more along the lines of passive observation of a lighting related subject of which I was studiously trying to remain well you understand. Now, it occurs to me (foolishly, perhaps) that if you call GO and specify a cue which previously had not been called up at all (or whose last command was GO Off) then it goes (excuse) without saying that this would mean "GO On". After that point, if the command GO were ever to be sent again with that same cue number, since the last command was effectively GO On then this command would be interpreted as GO Off.

To say this briefly, maybe the console should just interpret repeated GO commands specifying the same cue number as a toggle, alternately turning that cue On and Off. I don't know what operational difficulties would ensue with this plan, and I know it does not create a system where the cue can be positively engaged On or Off independent of its previous status (this is the biggest drawback) but I'm not sure if you require this in your scheme of things. Certainly, it is perhaps a logical response of a controller to toggle the cue if it receives more than one GO for the same cu (although this is definitely not what our system would do in response to repeated GO commands).

Anyway, this is just an idea - and after all isn't that what all this electronic scratchwork is for? In fact, I can see no reason that we can't add a new general command to the list which would be represented thusly:

Hex Command No.	of data bytes	R	M	S
0B	GO_OFF	variable		-23

as far as the detailed command and data description, you are invited to draught it in the following space:

(Thank you in advance, yr fthfl srvnt)

- Charlie

---[0108]--- (pref = [0107], nref = [0109])

[0109] (5 lines) CRichmond.USITT 11/01/90 0003.6 mst Thu midi
Subject: Re: Comments...

Multics very kindly condensed my header, but I think you get the message

about as well as I get the message from Multics about condensing my messages. (Around here I have a reputation for putting people to sleep while explaining how things work - callboarders [is that the word?] have the ability to abort the message and they should be thankful for that)
---[0109]--- (pref = [0108])

[0110] (17 lines) AEkvall.USITT 11/01/90 1248.5 mst Thu midi
Subject: Comments...
Hi.

Some comments on Matts message 0094.

Integer cue numbers are Ok by us.

Still I do not understand the philosophy of GO On/GO Off. I understand that you have a problem due to your working method but I do not understand when a normal (sorry, otherworking) system should send the GO Off. Especially not when you start new cues before the previous ones have finished to build on the first ones. Maybe, I have missed something fundamental, but please tell me if so.

Will anyone of you go to LDI in Orlando? I will be there. Hopefully we can take the change and meet and discuss eye-to-eye.

Anders
---[0110]---

[0111] (38 lines) AEkvall.USITT 11/01/90 1317.0 mst Thu midi
Subject: Comments...
Hi again

I also agree on using the Q_list for specifying which playback that should be affected by a GO command.

Matts message 0107

I can see that there is a big difference in thinking when Matt describe the GO command as a result of a user banging on the Flash keys. In theatre control much more is emphasized on the sequence of cues. Also, Rock n roll is more and more turning into this type of lighting with a recorded sequence of cues and effects instead of manually operated bumps. Therefore I think it is a bit strange that this flashing of masters should generate GO commands. This will conflict with the way control systems from companies like Richmond work. They seem to be made for starting things without having to bother to take them away afterwards. One state will replace an earlier state automatically when the next GO for that specific playback is issued. No matter if it is light or sound or anything. This is a nice and clean way to work that suits at least most theatre lightboards very well.

I dislike the idea of having a toggle each time a GO command for the same playback is issued. This means that the result will depend on the previous state of a specific playback. This can not be accepted. It is even worse than Move Fades!

I would like to be able to connect to or activate Celco boards with the same method as others, not having to add extra GOs to remove cues after

use.

It all seem to me like Celco is the only one having problems with this approach because they seem somewhat stuck in their method of activating/deactivating masters instead of having lighting states replacing each other on each GO signal. If anyone else has problem with this, please scream now.

Anders

---[0111]--- (nref = [0112])

[0112] (14 lines) JLBracewell.USITT 11/01/90 2141.8 mst Thu midi

Subject: Re: Comments...

I keep throwing the odd comment in from time to time, but this is one time and I have to agree with Anders. This whole GO ON/GO OFF business makes no sense whatsoever to me! As far as I'm concerned, sensibly designed lighting consoles replace one look with another (or scene, if you prefer.) I understand generally the problems involved in a memory system of having to find a way of comparing old and new levels and working out some percentage of each to accomodate a crossfade, etc., but that should be up to the particular console to work out. The theatre messages link should only say GO, and as a default the next numbered cue should run. If the GO is followed by a number, then that could jump to another cue other than the next in sequence; but otherwise

If I ever get to use this beastie, I'd sure hate to have to program in a GO OFF for each cue number I want to get rid of. Apologies if I've misunderstood the whole point of this.

---[0112]--- (pref = [0111], nref = [0113])

[0113] (23 lines) CRichmond.USITT 11/03/90 0047.5 mst Sat midi

Subject: Re: Comments...

Hi. Well, after reading the previous 3 messages, I do tend to agree with Anders as well. Perhaps Celco could set up an internal convention within the board to treat any cue number with a decimal point following it as the equivalent to GO Off for the parent number. For example, sending GO cue 14 would mean GO On cue 14 and GO cue 14. (or 14.1, etc.) would mean GO Off cue 14 within the Celco. In this way, it would basically remain compatible in that normally (in theatrical boards) cue 14.1 or some similar cue would supercede cue 14 and calling up cue 14 again afterward should not be a problem for anyone. Therefore cue 14 and 14.1 could be repeatedly sent one after the other with consistent results in everybody's board.

Re: LDI. I will be there from Nov. 13 through the 20th and look forward to meeting at any agreeable time. There is a session scheduled for discussion of various standards in which I hope to present at least briefly a synopsis of where we have come with this. I will be at our booth most of the time (Richmond Sound Design) and will post the name of the hotel (which I have forgotten) where I am staying in a subsequent message.

EOPp n

Hope to see all interested parties there!

-Charlie

---[0113]--- (pref = [0112], nref = [0114])

forum:

[0114] (90 lines) MDeakin.USITT 11/05/90 1218.4 mst Mon midi

Subject: Re: Comments...

Hello and good evening.

Well I really stirred up some comments about Go Off!

I understand that in theatre and multi media shows there is always a pre-defined script and cue list. The only factor that may change is the timing of the cues. However this is not the case in many other areas.

Consider the following:

The small club operator who has to light 3 bands a night, none of whom he has ever seen before. He can't use a pre-defined list of cues, changing from one to the next on a single playback.

The multi-purpose venue that has to light a fashion show with only 2 hours of programming time (if their lucky). All they can do is put in a number of cues and use them in various orders and combinations during the show.

Lighting a major rock act that changes the running order every night, puts in extra choruses without warning and does one or two or THREE encores depending....

What about Charlie's "Cantata for Brass, Lekos, and Flys". where is the artistic input for the lighting designer/operator if the only freedom he has is when to press the Go button?

Do we want to exclude these types of application from using MSC? I am sure they would all benefit from being able to control lighting, moving lights, colour changers, pyros, smoke machines and lasers all together. The important point is that there is no EXACT cue list for these types of show. Therefore the operator needs to be able to choose from a number of cues in various combinations at any time. This is why we need Go Off.

After all why does MIDI have note on and note off commands, if the music is all pre-defined why not programme up the keyboard in advance and just press a GO button during the 'performance'?

To clear up the point on Toggling cues on and off. This is definitely not what I had in mind, sorry if I confused you Charlie. If only GO ON commands are received then ON is all you get. Only a GO OFF or another GO ON for a different Cue on the same playback will remove the first Cue.

Relative as opposed to absolute commands are a very bad idea!

How all this will operate if some consoles need GO OFF and some don't is a problem. How about this:

Command Direction. Console needing Go Off. Console not need Go Off.

Go On In Activate Cue Activate Cue

Go Off In Remove Cue No Change

Go On. Out Sent when cue activated Sent when cue activated

Go Off Out Sent when Cue removed Sent when cue removed
or replaced **

** This output could be optional or be enabled and disabled on the console.

If no Go Off commands were received by a console needing them then only a Go On command with a new cue number for the same playback would remove the old cue. One option here would be to send GO ON cue 00 for a playback as a sort of Go Off. I know this could be used instead of defining a specific GO OFF but it's a bit of a bodge. We don't have any shortage of Command numbers so lets define a GO OFF. If you don't need it, ignore it and only send it if the option is turned on.

So Charlie, how about this:

Hex command	Number of data bytes	Recomm'd Min Sets
02 GO OFF	variable	123

Sorry but it will mean moving the numbers 02-0A to 03-0B but what are word processors for?

LDI, Yes I'll be there Anders so we can argue, sorry discuss this further then.

Logout Matt.

---[0114]--- (pref = [0113], nref = [0115])

[0115] (52 lines) MDeakin.USITT 11/05/90 1220.9 mst Mon midi

Subject: Re: Comments...

Hi.

This is Keith Dale using Matt's password.

(Keith is Celco's Marketing Director and a Lighting Designer. Matt)

Prior to those last Halloween messages regarding the MSC protocol I was an innocent bystander content to view all this from the fence. However, some of the views aired have prompted me to put digit to word processor and stick my two penneth worth in.

It would appear to me that those from the 'theatre' background have little comprehension of a loosely structured show (and by that I don't mean only Rock 'n' Roll) and would like to promote their method of control as being the base line by which every type is judged.

Let us consider two types of show environment. Firstly the 'theatre' type - generally performed in the same venue, with the same lighting and sound cues every night. Here it is possible that all cues are relative. Go Cue 96 replaces Cue 95 and triggers the smoke machine on (amongst

other things). The smoke machine stays on until Cue 97 is started so turning it off. All very nice in a controlled environment. It's much the same as sound EQ which will not change much from the first show to the last.

Now lets consider the 'rock' (for want of a better word) approach. In this situation the venue is likely to change every night. Here cues are absolute, Cue 96 triggers the smoke, Cue 97 triggers the down stage wash - Cue 96 is still smoking away until the atmosphere is saturated to the satisfaction of the Lighting Designer (who is usually but not always the operator). When the LD is happy with the level of smoke he removes Cue 96 (Go Off). Why does he do it this way? Because if the smoke was left to it's own devices until Cue 97 is triggered it might be too little or too much - depending on the venue, crowd capacity Etc. These variations have a similar effect on the sound engineer. Are his EQ settings the same every night? I think not.

To conclude then, this debate must not develop into an 'us and them' situation. Anders must surely see the light (pun intended) now. Celco consoles are not unique (we'll forget the superlatives here). What must be clearly understood is that our type of 'hands on' console is not a quirk of the Celco marketing and R&D departments but a response to the needs of the market. If you want to exclude what you may call the 'Celco approach' you are, in effect, excluding that sector of the industry. If so you will end up with the on-stage tech. unplugging the dammed smoke machine because Cue 97 hasn't been sent yet and the band are fast disappearing under a cloud. And who is responsible for choosing this system?

I'll be at LDI too - and I'm bringing my gloves!

Keith Dale.

---[0115]--- (pref = [0114], nref = [0116])

[0116] (24 lines) CRichmond.USITT 11/06/90 0307.1 mst Tue midi

Subject: Re: Comments...

Well, I certainly noticed how studiously everyone has avoided commenting on my compromise proposal... The assumption is that since we're dealing with a pack of vehement integer cue number types, the suggestion involving little dots was simply beyond the pale.

So, undaunted, here's another try at an appeasement to break this critical deadlock in international negotiations:

How about setting up a 'standard' option of making any even numbered cue be the 'GO Off' corresponding command for the previous 'GO' of an odd numbered cue. Although this probably rates as the poorest worded proposal in the recently known world, let me explain by way of example:

If GO cue 14 is sent, the console may be optionally configured to recognize this as GO Off cue 13. (Sorry about the decimal point but I wanted to end the sentence there) Controllers could even be made smart enough to say to their operators "GO cue 14" or "GO Off cue 13" depending on whether they were running one type of show or another. This could easily be a user chosen menu item and the message sent would of course be the same.

Oh well, just trying to help...

-Charlie

---[0116]--- (pref = [0115])

[0117] (50 lines) AEKvall.USITT 11/07/90 1344.5 mst Wed midi

Subject: Reply

Matt

What happens if you say Go On on the same playback that already has a cue On? You say that it is removed. Will the old cue fade out and the new cue fade in dipless or will it cut?

I dislike the idea of needing to make a lot of setups to take care of all the possible GO On/Off actions as you propose in 0114. The GO Off stuff really seems to complicate the connection of different systems.

Keith Dale

Keith, you talk about us as having Utheatre onlyU background and forgetting about hands-on control. I do not agree. We are not restricted to one market. We are making light controls for all types of users, including 'rock' and real-time performance. You do not have to explain what that means to me.

We have put a lot of effort in making our faders or playbacks as multi-purpose as possible. I would dare to say that they are more powerful for TrockU than your faders. We are also proud of our faders but we don't make a lot of fuzz for that.

Our faders (that already can be MIDI activated) are a big part of any show due to their advanced features and live performance capabilities. Our systems is not intended strictly as preprogrammed show executers but as excellent live tools. However, we have understood that the SET command should be used for moving faders, not the GO command. GO is used for activating cues in a preprogrammed sequence (which we also have). Please accept that and use the SET command too. Then our consoles (Celco and AVAB) would speak nicely to each other (if you want them too?!) without having to make complicated setups.

Conclusion: We are not at all against the hand-on feeling. In fact, to be able to be really real-time, live preformance useful, we have added real-time MIDI control for all faders and keys since more than one year.

We are not trying to exclude your market segment. Please stop talking about Uus-and-themU. I do not understand where this giant misunderstanding is coming from. We are talking about the same thing

We are talking about the same thing but you have gotten stuck with the idea that you should be able to use the GO command instead of the intended SET. Or as you would describe it: See the light!

Why bring your gloves? IsnUt this an intellectual fight?!

Anders

---[0117]---

[0118] (8 lines) CRichmond.USITT 11/10/90 0158.3 mst Sat midi

Subject: LDI

I will be staying at the Orlando Heritage Inn from November 13th through the 19th - leaving the 20th. If anyone wants to get in touch please leave a message for me there; hope we can get together in a peaceable fashion sometime during this event!

Phone in Orlando 352-0008. See you then...

-Charlie

---[0118]--- (nref = [0119])

[0119] (11 lines) RThomas.USITT 11/11/90 1107.7 mst Sun midi

Subject: Re: LDI

As usual, I am not certain about what is technically being discussed here, just have some feelings (as an end user) from what I am able to sift out of this discussion that I can't resist venting: The comments regarding MIDI note messages having go on and go off get many hurrahs from this end of the world. Any manufacturer should be able to simply (or they won't do it) configure their software and hardware so that turning a controller on (e.g. pressing a "nybka(1 [E: is a GO CUE, and releasing the same controller (lifting the finger off the key) is a GO OFF cue. How this gets accomplished in the spec is for everybody else to figure out. I just can't imagine the spec without the capability of performing this simple task.

---[0119]--- (pref = [0118], nref = [0120])

[0120] (23 lines) CRichmond.USITT 11/11/90 2316.3 mst Sun midi

Subject: Re: LDI

Rick, I think you are basically correct except that, as you correctly identify, the need to turn something off that has been turned on before really applies to the controller area, not in fact the "Cue" area. In other words, if you set a controller ON using our proposed SET command syntax, then that controller will stay on indefinitely until it is SET to a new value. In this way, our controllers are very similar to the notes you refer to in the original MIDI spec.

Cues, however, are somewhat different - they are analogous to the original "Program change" of the MIDI spec. The way most MIDI devices respond to Program Change is exactly the way we expect devices responding to MIDI Show Control Cue GO's - that each new Program Change or GO Cue replaces the settings defined by the previous Program or Cue. In essence the new Program/Cue replaces the old one. Our proposed spec goes a little further and allows the user to define a specific Cue List in the device in which the new Cue is to be placed when it GO's and it is in that Cue list only that the GO causes the old cue to be replaced (leaving extant cues in other lists untouched until they receive a new GO Cue).

EOPp n

I hope this clarifies the situation....

-Charlie.

---[0120]--- (pref = [0119], nref = [0121])

forum:

[0121] (51 lines) MDeakin.USITT 11/12/90 1147.5 mst Mon midi

Subject: Re: LDI

Hi.

Anders in reply to your message 117.

Go On messages to a playback which already has an active Cue on it can crossfade or cut depending on any times stored.

As I understand the proposal the SET commands are for interlinking Controllers. I.E. specific front panel knobs and buttons. Go commands are for controlling Cues which have no physical presence on the front panel. The problem of using Go commands only for preprogrammed sequences of Cues is the lack of flexibility in live applications as I pointed out in my last message.

It seems to me that your suggestion means the following:

GO commands are for preprogrammed shows only.

SET commands are for Live shows and ancillary functions.

EOPp

On our boards the Playback Faders (sub masters) can each have one of a number of Cues assigned to it. However the SET command only allows us to say SET fader X to Level Y regardless of which Cue is assigned to the Playback fader. This means we would have to define other SET commands to change the Cue assignments Etc. Now we are back in a situation where specific commands relate to specific front panel controls on each type of control unit. There is no standardisation and we might as well be using MIDI note and controller numbers and let chaos rule.

I am not trying to force some special need on the rest of you. I simply feel that Celco and other manufacturers will need a Go Off command. If we don't define one now then a number of unspecified short cuts will arise like using Cue 00, or Odd and Even Cues or dotty and integer cues Etc. to fulfil this requirement. These will be incompatible and simply create confusion.

To clarify a point. There is only one set-up required in the suggestion in my last message. That was if a control unit does not need Go Off commands it could be set-up to send a Go Off for an outgoing Cue before sending the Go On for the incoming Cue.

E.G. Go On Cue 4 on Playback 1
Go Off Cue 4 on Playback 1 (optional)
Go On Cue 9 on Playback 1

This is my last message before LDI so see you all there.

Matt.

---[0121]--- (pref = [0120])

forum:

[0121] (51 lines) MDeakin.USITT 11/12/90 1147.5 mst Mon midi

Subject: Re: LDI

Hi.

Anders in reply to your message 117.

Go On messages to a playback which already has an active Cue on it can crossfade or cut depending on any times stored.

As I understand the proposal the SET commands are for interlinking Controllers. I.E. specific front panel knobs and buttons. Go commands are for controlling Cues which have no physical presence on the front panel. The problem of using Go commands only for preprogrammed sequences of Cues is the lack of flexibility in live applications as I pointed out in my last message.

It seems to me that your suggestion means the following:

GO commands are for preprogrammed shows only.

SET commands are for Live shows and ancillary functions.

On our boards the Playback Faders (sub masters) can each have one of a number of Cues assigned to it. However the SET command only allows us to say SET fader X to Level Y regardless of which Cue is assigned to the Playback fader. This means we would have to define other SET commands to change the Cue assignments Etc. Now we are back in a situation where specific commands relate to specific front panel controls on each type of control unit. There is no standardisation and we might as well be using MIDI note and controller numbers and let chaos rule.

I am not trying to force some special need on the rest of you. I simply feel that Celco and other manufacturers will need a Go Off command. If we don't define one now then a number of unspecified short cuts will arise like using Cue 00, or Odd and Even Cues or dotty and integer cues Etc. to fulfil this requirement. These will be incompatible and simply create confusion.

To clarify a point. There is only one set-up required in the suggestion in my last message. That was if a control unit does not need Go Off commands it could be set-up to send a Go Off for an outgoing Cue before sending the Go On for the incoming Cue.

E.G. Go On Cue 4 on Playback 1
Go Off Cue 4 on Playback 1 (optional)
Go On Cue 9 on Playback 1

This is my last message before LDI so see you all there.

Matt.

---[0121]--- (pref = [0120])

[0122] (10 lines) Meldrum.USITT 11/15/90 0955.7 mst Thu midi

Subject: Cue's

The Artisan Console, to pick one at random, supports cue numbers in the range 000.01 to 999.99. I believe that Strand console allow a x.x numbering scheme. I don't think anyone is being overly realistic to think we could make a world wide, do it all, standard by saying 'use

integer cue's'. Please could everyone review their reasons for asking for this. I'd VERY much like to see a more arbitrary range.

Greetings to all

Andy

---[0122]---

[0123] (27 lines) CRichmond.USITT 11/22/90 0120.0 mst Thu midi
Subject: Revisions

Talked with the MMA today and it has been confirmed that if we move quickly, we will have no difficulties getting a revised version of MSC before the TSB before they have to vote on it. This will be good news for those participants in this forum who met in Orlando this past weekend and decided to agree on some new additions to the spec. You will be seeing each proponent submit their agreed-upon additions herein and forthwith.

If there is no further hue and cry against these upcoming proposals, I will incorporate the additions into a new draught proposal and title it MIDI Show Control 0.2 then send it off to the MMA.

A note to Matt- for the purposes of confusion reduction I think it would definitely be advisable to make the new command "GO Off" have a hex code of 0B. This will eliminate any confusion between the two draught proposals since all other commands will retain the same codes.

Also, for both Matt and Anders - PLEASE provide in your submissions detailed and unambiguous descriptions of the Command(s) and Controller Ranges(?) you will be proposing (a' la the descriptions in Section 5. of the first document)

Thank you all very much (I did enjoy very much meeting all of you, the "players" of this pageant) and look forward to the future

- Charlie

---[0123]--- (nref = [0124])

[0124] (116 lines) MDeakin.USITT 11/26/90 1146.3 mst Mon midi
Subject: Re: Revisions
Hi.

OK Charlie you asked for it!

2.2 Device identification.

ADD:

Device_ID Value.

01-6FH Individual unit addresses.

70H-7EH Channels addresses 1-15

7FH Broadcast to all units of Command_format Type

Every type of device must be able to respond to both an individual address and the broadcast messages. The channel addressing mode is

optional.

A device may respond to one or more individual addresses and one or more channel addresses.

Both the Device_ID and the Command_format of a message must match the Device_ID address and Command_format type of a device before the message is accepted.

4.1 Command_Formats.

ADD:

Command_format 7FH = All Types.

This is equivalent to the Device_ID 7FH broadcast address.

Use of this Command_format along with the broadcast Device_ID allows a complete system to be RESET with a single message.

4.3 General Commands.

ADD:

Hex command	Number of data bytes	Recomm'd Min Sets
0B GO_OFF	Variable	123

5. DETAILED COMMAND AND DATA DESCRIPTIONS

ADD:

0B GO_OFF

Starts a transition or fade of a cue to the off state.
Transition time is determined by the cue in the
Controlled Device.
If no Cue Number is specified, the last cue GOes OFF.
If a Cue Number is specified, that cue GOes OFF.

0B GO_OFF
<Q_number> optional; required if Q_list is sent
00 delimiter
<Q_list> optional; required if Q_path is sent
00 delimiter
<Q_path> optional

Compatibility of GO_OFF between devices.

Not all types of device will need to receive GO_OFF commands for correct operation as slaves. However to maintain compatibility when these devices are used as masters they should send GO OFF commands as follows:

Command	Direction.	Console needing	Console not needing
		Go Off.	Go Off.

Go On In (slave) Activate Cue Activate Cue

Go Off In (slave) Remove Cue No Change

Go On.	Out (master)	Sent when cue	Sent when cue
		activated	activated

Go Off	Out (master)	Sent when Cue	Sent when cue
		removed	removed or replaced **

** This output could be optional or be enabled and disabled on the console.

Notes on implementations.

For lighting applications the Q_list should be used to define the Playback the cue will be (or is) loaded onto. This applies to all occurrences of Q-list in GO, TIMED_GO, GO_OFF, LOAD Etc. A playback number is optional but if used should be in the range 1-120.

This is my understanding of the "Orlando Accord" I hope you agree Anders.

Cheers!

Matt.

---[0124]--- (pref = [0123], nref = [0125])

[0125] (8 lines) CRichmond.USITT 11/28/90 0050.2 mst Wed midi

Subject: Re: Revisions

Sounds lovely at first blush! Hope everyone else agrees and I can put it into the document at get it off to the MMA pronto. I will study this over the next day or so and see if any details need changing.

Thanks very much, folks. I'm glad you had a constructive meeting on this.

- Charlie.

---[0125]--- (pref = [0124])

[0126] (5 lines) CRichmond.USITT 11/29/90 0122.1 mst Thu midi

Subject: Controller ranges

Just a reminder, Anders, that you were going to upload a revised(?)

list of recommended standard controller ranges as well as confirming your agreement of the "Orlando Accord".

Thanks, Charlie
---[0126]---

[0127] (31 lines) AEkvall.USITT 12/01/90 1106.3 mst Sat midi
Subject: Revisions

Hi folks!

Revisions

Everything sounds OK to me, Matt.

Here is my updates to the list of SET parameter numbers.

SET numbers

0-127 Sub masters.

128-129 Masters of the first playback. 130-131 Masters of the second playback. ... 190-191 Masters of the 32nd playback. 32 different playbacks should be enough.

192-223 Speed controller for the above 32 playbacks.

224-255 Chase sequence masters.

256-287 Chase sequence speed masters.

510 Grand Master for all channels.

511 General speed controller for all fades.

512-1023 Channel levels (8 or 16 bits can be used) for slow transmission of actual light levels.

Anders
---[0127]--- (nref = [0128])

[0128] (30 lines) CRichmond.USITT 12/02/90 1726.7 mst Sun midi
Subject: Re: Revisions

Thank you all very much! I will incorporate these additions into the current spec and forward it to the MMA. They have already mailed out the first version and are receiving feedback and comments on it from the Technical Standards Board until the 10th of December. This revision will form an official response to the proposal now that it is already "on the street". What will happen will be one of the following: A. The proposal will be rejected B. It will be accepted with additional changes made by the TSB C. It will be accepted with our new additions

D. It will be accepted with our new additions plus changes made by the TSB

If it is accepted in any form, it will be forwarded to the Japan MIDI A Standards Committee for approval or further recommendations. At that point it will probably be accepted but may be returned with changes for

more consideration. This process could take a while but we can try to push them to move on it as quickly as possible. I'll keep you posted and also will upload the next version to Callboard for your reference as soon as possible, but it will basically just contain Matt's and Anders' recommendations unless you hear otherwise in this forum.

There will be more recommendations put forward here in the next while, but I suspect we will not have the time to discuss them and get them to the MMA before the 10th, so they will have to wait for the next round of standards voting. I do think we have a workable proposal here that can be built upon without difficulty in the future, so not to worry!

Thanks again - on again soon.

-Charlie
---[0128]--- (pref = [0127])

[0129] (291 lines) CRichmond.USITT 12/04/90 0008.8 mst Tue midi
Subject: New draught

I have put together "Version 0.2" of MIDI Show Control (proposed) and have uploaded a complete copy of it to the proposals database of the MMA proprietary area of the PAN Network for access by all members of the MMA. I will also do the same putting it into the common/standards area of Callboard for your reference.

I have also excerpted the sections of the new draught which have been changed since the first version and uploaded the following message to the MMA Forum on the PAN network. This will appear on the fax machines of most MMA members tonight and will be read by anyone following the MMA Forum:

The following are excerpts from the latest draught (Version 0.2) of MIDI Show Control. These are the only sections that have been changed since Version 0.01. Unchanged sections of the text are represented by a short line of doublespaced dots, as follows:

.....

2.2. DEVICE IDENTIFICATION

<device_ID> is always a DESTINATION device address.

Commands are most often addressed to one device at a time. For example, to command two lighting consoles to GO, transmit:

```
F0 7F <device_ID=1> <msc> <command_format=lighting> <GO> F7  
F0 7F <device_ID=2> <msc> <command_format=lighting> <GO> F7
```

<device_ID> values:

- 00-6FH Individual ID's
- 70H-7EH Group ID's 1-15 (optional)
- 7FH "All-call" ID for system wide broadcasts

Every device must be able to respond to both an individual and the "all-call" (7F) ID. The group addressing mode is optional. A device may respond to one or more individual ID and one or more group ID. Both <device_ID> and <command_format> of a message must match the device_ID and command_format of a controlled device before the message is recognized.

If two separate controllers responding to the same command_format are set to respond to the same device_ID then only one message need be sent for both to respond.

The "all-call" device_ID (7F) is used for system wide "broadcasts" of identical commands to devices of the same command_format (or to all devices when used with <command_format=all-types>; see 4.1, below.)

Before fully interpreting the <device_ID> byte, parsing routines will need to look at <msc> and <command_format>, both of which follow <device_ID>, in order to first determine that the Sysex contains Show Control commands in the appropriate format.

.....

Controlled Devices which support Q_list and/or Q_path will normally default to the current or base Q_list and Q_path if these fields are not sent with Q_number.

For lighting applications, Q_list optionally defines the Playback or Submaster Controls (0 to 127) with which the cue corresponds.

It is highly recommended that every manufacturer publish a clear and concise description of their equipment's response to the above conditions.

3.2. TIME CODE NUMBERS

.....

4.1. COMMAND_FORMATS

Command_formats fall into the categories of General, Specific and All-types. General command_formats have a least significant nibble equal to 0, except for lighting which is 01H. Specific command_formats are related to the General command_format with the most significant nibble of the same value, but represent a more restricted range of functions within the format.

Command_format "All-types" (7F) is used for system wide "broadcasts" of identical commands to devices of the same device_ID (or to all devices when used with <device_ID=All-call>; see 2.2, above.)

For example, use of the All-types command_format along with the All-call device_ID allows a complete system to be RESET with a single message.

Controlled Devices will normally respond to only one command_format

besides All-types. Occasionally, more complex control systems will respond to more than one command_format since they will be in control of more than one technical performance element. Controllers, of course, should normally be able to create and send commands in all command_formats, otherwise their usefulness will be limited.

Hex	command_format
00	reserved for extensions
01	Lighting (General Category)
02	Moving Lights
03	Colour Changers
04	Strobes
05	Lasers
06	Chasers
10	Sound (General Category)
11	Music
12	CD Players
13	EPROM Playback
14	Audio Tape Machines
15	Intercoms
16	Amplifiers
17	Audio Effects Devices
18	Equalisers
20	Machinery (General Category)
21	Rigging
22	Flys
23	Lifts
24	Turntables
25	Trusses
26	Robots
27	Animation
28	Floats
29	Breakaways
2A	Barges
30	Video (General Category)
31	Video Tape Machines
32	Video Cassette Machines
33	Video Disc Players
34	Video Switchers
35	Video Effects
36	Video Character Generators
37	Video Still Stores
38	Video Monitors
40	Projection (General Category)
41	Film Projectors
42	Slide Projectors
43	Video Projectors
44	Dissolvers
45	Shutter Controls
50	Process Control (General Category)

- 51 Hydraulic Oil
- 52 H2O
- 53 CO2
- 54 Compressed Air
- 55 Natural Gas
- 56 Fog
- 57 Smoke
- 58 Cracked Haze

- 60 Pyro (General Category)
- 61 Fireworks
- 62 Explosions
- 63 Flame
- 64 Smokepots

- 7F All-types

.....

4.3. GENERAL COMMANDS

The following commands are basic to the current implementation of Memory Lighting systems and probably apply to all dedicated theatrical show control systems in a general sense. Although it is not required that Controlled Devices incorporate all of these commands, it is highly EOPrecommended:

Hex command	Number of data bytes	Recomm'd Min Sets

00 reserved for extensions		
01 GO	variable	123
02 STOP	variable	123
03 RESUME	variable	123
04 TIMED_GO	variable	-23
05 LOAD	variable	-23
06 SET	4 or 9	-23
07 FIRE	1	-23
08 ALL_OFF	0	-23
09 RESTORE	0	-23
0A RESET	0	-23
0B GO_OFF	variable	-23

.....

06 SET

Defines the value of a Generic Control. The Generic Control and its value are each specified by a 14 bit number. A Controlled Device may treat virtually any of its variables, attributes, rates, levels, modes, functions, effects, subs, channels, switches, etc. as Generic Controls which may be sent values via SET. Optionally, the time it takes the Generic Control to

achieve its value may be sent.

Time is Standard Time Specification with subframes (type {ff}), providing anything from "instant" to 24 hour transitions. If a Controlled Device does not support times in SET, it should ignore time data.

Standard Generic Control Numbers for Lighting:

- 0-127 Sub masters
- 128-129 Masters of the first playback
- 130-131 Masters of the second playback
- ...
- etc.
- ...
- 190-191 Masters of the 32nd playback
- 192-223 Speed controllers for the 32 playbacks
- 224-255 Chase sequence masters
- 256-287 Chase sequence speed masters
- 510 Grand Master for all channels
- 511 General speed controller for all fades
- 512-1023 Individual channel levels

- 06 SET
- cc cc Generic Control Number, LSB first
- vv vv Generic Control Value, LSB first
- hr mn sc fr ff Standard Time Specification, optional

.....

0B GO_OFF

Starts a transition or fade of a cue to the off state. Transition time is determined by the cue in the Controlled Device.

If no Cue Number is specified, the current cue GOes OFF. If a Cue Number is specified, that cue GOes OFF.

In Controlled Devices with multiple Cue Lists, if no Cue Number is Specified, all currently active cues in Open Cue Lists GO OFF. If Q_number is sent without Q_list, all cues with a number identical to Q_number and which are in Open Cue Lists GO OFF.

For compatibility with Controlled Devices which do not automatically replace an existing cue with a new cue upon receipt of the GO command, Controllers should optionally prompt the programmer to simultaneously create a GO_OFF command.

- 0B GO_OFF
- <Q_number> optional; required if Q_list is sent
- 00 delimiter
- <Q_list> optional; required if Q_path is sent
- 00 delimiter
- <Q_path> optional

.....

I am confident that I have accurately incorporated both Matt's and Anders' recommendations into the proposal even though the wording in some areas is not identical to that proposed. The changes have been made to bring the terminology in line with the expressions already established in the first version and to smoothly incorporate the new concepts and extensions into the existing document. If I have screwed up in any way, please let me know as soon as possible. Time is of the essence here and I was requested by the MMA to get this out immediately since they are already behind schedule and need to see the revisions in order to vote on the document by the 10th of December. If there are any serious omissions or mistakes we still have time to fix it (and will have for a while) but I wanted to make sure everyone had the latest info as soon as possible.

That's it for now....

- Charlie

---[0129]--- (nref = [0130])

[0130] (5 lines) Clinton.Thea 12/04/90 1139.7 mst Tue midi

Subject: Re: New draught

I have made the new draft available in the standards directory. Look under:

>udd>USITT>common>standards>MSC_0.2

I've also given it the name "midi_show_control_0.2"

---[0130]--- (pref = [0129], nref = [0131])

[0131] (49 lines) CRichmond.USITT 12/05/90 0138.4 mst Wed midi

Subject: Re: New draught

Would that be the new draft or the new draught? These details are m important, you know! (Thank you, Tim)

On a more sombre (or is it somber) note - I have been informed that the Japan MIDI Standards Committee could drag their feet for a while even after this version has been officially adopted by the Technical Standards Board of the MMA (which is a virtual certainty, according to our inside contacts). The reason for this is that the Japanese manufacturers are much more oriented to consumer applications than professional ones and will tend to ignore this proposal rather than take the time to understand it. It does not mean it will be rejected by the JMSC but that it may take a while to be approved.

If we wish to start using this protocol as we have agreed upon it, we do have the option open to us to do so almost immediately. The way this was suggested by the MMA is as follows:

1. Have USITT front the refundable \$200 deposit to the MMA and obtain a "manufacturer's system exclusive number"
2. Publish the specification as an interim measure using the USITT sysex number in place of the unique universal real time system exclusive number (sub-ID 1) which must be assigned when the JMSC finally approves the spec.

3. All manufacturers who wish to start using this protocol may do so using the USITT sysex number and may at a later date upgrade their software or firmware to also recognize the sysex number assigned when the spec becomes a MIDI standard.
4. As soon as any manufacturer actually produces a product using the interim protocol, USITT will get their deposit back. Perhaps a manufacturer would be able to even front the deposit for USITT.

I don't see any drawback in this plan myself since it does have the blessing of the MMA, although I do think we should at least wait the month or so until the proposal is actually approved by the TSB. If we agree that we want to proceed as soon as possible, we could ask USITT to apply for the sysex number now so that it will be assigned by the time we need to start using it.

If we get positive responses to this suggestion, I will proceed with asking USITT to apply for the number. You may watch this space for reports, as usual.

Yr fthfl srvnt

- Charlie
---[0131]--- (pref = [0130])

[0132] (20 lines) CRichmond.USITT 12/06/90 0052.1 mst Thu midi

Subject: MMA Update

The responses we have received to the MSC proposal have been very positive so far. There has been one suggestion that universal sysex protocol adopt the "group" numbering scheme developed at the Orlando Accord for all universal sysex messages! - congrats to Matt and Anders on this.

In the next little while, Ralph Weber will be uploading a proposal which is some 19 pages long and forms an extension to the current spec. We do not expect to be able to debate this sufficiently before this first version gets approved, so it will have to form part of the next round of proposals if at all. I don't want to speak for Ralph at this stage except to say that it defines a protocol which can be used in systems which provide for a "closed-loop" communications network. I have told the MMA that Ralph has been working on this and they are encouraging us to support this direction as a very important and logical extension to the MSC spec for high reliability installations.

Enjoy -

Charlie
EOPp n
---[0132]---

forum:

[0133] (46 lines) Weber.USITT 12/06/90 0423.5 mst Thu midi

Subject: Optional MSC two-phase commit

I have taken Charlie up on his offer to accept additional information for the draft of the MIDI Show Control (MSC) specification. My contribution is a draft of the additions required to include an optional two-phase commit capability in MSC.

Two-phase commit is a data base system and transaction processing methodology that allows multiple computer systems or data processing applications to coordinate their activities such that a group of functions are all processed or none of them are processed. Two-phase commit is very similar to the standby - go cue calling methodology commonly used in vocal theatrical cue systems.

One of the immediate benefits of two-phase commit methodology is the ability to use the checksums that some have suggested be added to MSC. My draft proposal includes checksums as well as details of their computation and usage.

I have asked Tim Clinton to place my draft proposal in:

>udd>USITT>common>standards>MSC_2PC_0.01
EOPp n

The propasal is approximately 47 Kbytes long (around 19 typewritten pages).

At this time, I can think of only one section that needs to be added. There should be a discussion of error recovery that is outside the examples section. This is because I have specifically stated that the examples section is not a part of the specification definition. Since the error recovery methodology needs to be defined, a strict interpretation of the document requires that it be defined outside the examples section.

There is nothing wrong with the error recover example information that is present, and it gives good guidance into how error recovery is handled in two-phase commit. So, you can use that information as a guide to what will appear in the separate error recovery section.

Regardless of how complete I think this proposal may be, I expect that many important additions will be described by the MIDI Show Control community (in this forum). If nothing else, I may have assumed to much in some of the existing discussions. Additional clarification may be necessary to accomodate someone not immediately familiar with the two-phase commit concept. I may also have made grossly incorrect assumptions about the technical details of MIDI.
EOPp n

I look forward to a lively debate on this propasal. Hopefully, we can enhance it to the point where it is ready for inclusion in MSC over the course of the next several months.\032
---[0133]--- (nref = [0134])

forum:
[0134] (2 lines) Clinton.Thea 12/06/90 0950.1 mst Thu midi
Subject: Re: Optional MSC two-phase commit
Ralph's proposal is now in the standards directory, with the name he suggested.
---[0134]--- (pref = [0133], nref = [0135])

forum:
[0135] (34 lines) MDeakin.USITT 12/07/90 1024.8 mst Fri midi
Subject: Re: Optional MSC two-phase commit
Hi.

Charlie. Yes, great the additions are as I intended, thanks.

The only addition I would suggest is to add a little explanation of the Group IDs in the Device Identification section 2.2. For example:

The optional Group IDs can be used in addition to the Individual and All-Call ID. The Group ID allows sets of similar equipment to be addressed as a group without having to address each one individually. This can be thought of in similar terms to Channels in the MIDI specification.

I can't decide if the GO_OFF command should be included in the number 1 minimum set. I think it should, does anyone else have any opinions?

As far as the the idea of having a USITT Sysex number is concerned I think this is fine for development but no products should be launched using this number. If this happens then there is the risk of incompatible systems being out in the market. I know software can be upgraded but this quickly becomes a major problem if even a small number of units are sold. This is made worse if products are sold through a distribution network making it hard to contact end users. I know this from experience!

If this means the USITT can't get it's \$200 back then I'm sure a number of manufacturers would contribute, Celco would.

I shall be down loading Ralph's proposal, provided my modem doesn't overheat, and will read it with interest.

TTFN

Matt.

---[0135]--- (pref = [0134])

forum:

[0136] (69 lines) AEkvall.USITT 12/08/90 0426.1 mst Sat midi

Subject: Comments

Hi there

I think that we need some kind of clarification on how some of the SET controllers should be used.

All masters (subs, playback, chase, Grand Master) should have a resolution of 8-bits. Full should be represented by an internal 255 and zero by an internal 0.

The masters of a playback:

The masters are grouped in pairs. The first should control the channels fading down in a crossfade, the second one the channels fading up. This also means that normally (when no fade is in progress) the first master is at full and the second master is at 0. To make a crossfade, the first master is moved down (fading down the old channels) at the same time as the second master is moved up (fading up the new channels). When

the first master reaches 0 and the second master reaches full, the crossfade will be completed and the sequence will be advanced one step to be ready for the next cue. The next crossfade can not be started until the first master is moved back to full and the second master is moved back to 0. Then they grab the next crossfade.

The speed controllers for the playbacks, chase speed master and general speed control:

Here we should decide a middle value which could be 128 in 8-bit resolution or 32768 in 16-bit resolution. This middle value means that the speed should be the preprogrammed one. A higher value will increase the speed and a lower value will decrease the speed. I do not know which scale and time base we should use for speed changes. Suggestions?

Grand Master

This is an 8-bit value. When at 255 it should allow all channels to go untouched to the output. At 0, no light should be output.

Chase Masters

This is an 8-bit value. When at 255, a chaser should be at its maximum programmed level. At 0, no light from the chaser should be produced.

Channel levels

This is also an 8-bit value. Is there a need for a higher resolution for channel levels?

Shall we provide a 16-bit format as a complement. This could be used when giving new targets to a motorized lantern.

1024-1535 could be defined as 16-bit channel levels.

Maybe a better solution would be to have the same approach as for the standard MIDI 14-bit controllers. There you normally use only the higher 7-bits when you need low resolution. If we normally put an 8-bit channel level in the highest byte of MSC SET values, this will work fine when connected to a unit requiring 16-bit values (the LSB is zero). You will lose resolution but it will work directly. A controller capable of sending 16-bit values may use the LSB as well. Then we only have format and no confusion.

Anders

---[0136]--- (nref = [0138])

[0137] (23 lines) CRichmond.USITT 12/09/90 0009.0 mst Sun midi
Subject: Re: Matt's comments

Thanks for the input. Your clarification of the Group ID numbers is beneficial. I don't think at this point there is any misunderstanding of the Group ID concept since it is so closely allied to the original MIDI Channel idea (I changed it from 'Channel' to Group because of the

slight difference in implementation and different numbers)

I must admit that I did not feel it was initially advisable to make GO_OFF be part of the minimum implementation requirements since there had been so much discussion of whether or not it should even be part of the spec at all! Not to denigrate its value, but we are really talking here about what the very cheapest boards should have and I'm not sure a lot of manufacturers would be able to do much with it in a very low end product. I could also easily be wrong.

We are all really hoping that all will go smoothly and the spec will be approved without undue delay. My dissertation was probably somewhat premature, but I did want us to start thinking about this possibility and to prepare you for the worst if it happened.

EOPp n
Thanks again, Matt.

Cheerio, Charlie.
---[0137]---

forum:
[0138] (23 lines) CRichmond.USITT 12/09/90 0027.2 mst Sun midi
Subject: Re: Comments
This is a very worthwhile process and the direction you are going seems to be quite logical except for one detail: because all sysex data must use bytes that always start with 0 as the first bit, we only have 7 bit bytes available to use and a maximum of 14 bits for controller data in SET. It would be valuable here to review your suggestions and see if substituting 7 for 8 and 14 for 16 would make sense or would compromise resolution too much. I know that we would feel uncomfortable in audio restricting specific controls to 7 bits only.

The way that I felt this command would be used would be that both the controller and the controlled device would simply implement as many bits of resolution that they could: i.e. if a device were 7 bit, then it would only send/receive meaningful data in the MSB and send zeros/ones/whatever in the LSB (controller) and toss away or disregard the LSB (controlled device). Variations on this would be logical for any number of bits a device could handle. This, admittedly, is a much more cavalier approach to the standard than yours and a good workable standard with a more defined approach will be welcome (so please continue) but we don't have more than 14 bits to work with.

Thanks, Anders

- Charlie
---[0138]--- (pref = [0136], nref = [0141])

[0139] (22 lines) CRichmond.USITT 12/10/90 0138.9 mst Mon midi
Subject: 2PC

I have had a brief opportunity to scan Ralph's proposal but not to study it in any great depth, so I only have a couple of comments:

1. The "master" and "slave" terminology seems to duplicate the already established "controller" and "controlled device" concepts (unless there's something I'm missing). Would it be possible to simply replace

"master" with "controller" and "slave" with "controlled device"? (I never have been big on these S&M relationships anyway)

2. We need to rework the data ranges and definitions of the Checksums, Sequence Numbers, and Status Codes (Sections 6.2-6.4) keeping in mind that MIDI Sysex data is restricted to 7 bit bytes because the first bit always has to be 0. (Sorry - this was originally a very limited concept communications vehicle when devised in 1984). Ralph - do you think you could fit all you need to do within 14 bit words or will we have to expand the size of some of the defined data ranges? I suspect we can make do with proportional adjustments.

That's it for now, more later. Ta Ta

- Charlie
---[0139]--- (nref = [0140])

forum: pn
forum: Unknown request "pn". Type "?" for a request list.

[0140] (9 lines) Weber.USITT 12/11/90 0725.9 mst Tue midi
Subject: Re: 2PC
I will work on changing the master/slave terminology.

The status code idea is easily amenable to 7-bit bytes. Likewise for sequence numbers. Timeout times (in the STANDING_BY message) and checksums are a little nastier. I will work on this too.

Thanks.

Ralph:
---[0140]--- (pref = [0139])

[0141] (9 lines) Weber.USITT 12/12/90 1749.5 mst Wed midi
Subject: Re: Comments
Charlie's comments in 138 & 139 apply to the Generic Controller Number field of the SET command too. Because of the restriction to 7 bits, a Generic Controller Number of 128 is not valid... unless some kind of numerical manipulation of the bits is being applied to move the on bit from the sign position of the LSB to the low order position of the MSB.

Thanks.

Ralph:
---[0141]--- (pref = [0138], nref = [0142])

forum: pn# n

[0142] (33 lines) CRichmond.USITT 12/13/90 0126.4 mst Thu midi
Subject: Re: Comments
The Generic Controller Number (and Value as well) is a 14 bit number which is defined by two 7 bit bytes, sent LSB first. Therefore controller 128 set to a value of 128 would be represented by the following Controller Number: 00 01h , followed by a Value of 00 01h. These 7 bit bytes are actually contained in 8 bit bytes of which the first bit must always be zero because of restrictions in System Exclusive (by definition the first bit being a one signals the end of sysex). So 16 bits have to be sent in order to give us 14 bits of

usable data and therefore the previous values truly indicate 128 since the highest number the LSB alone can represent is 127.

Is everyone still with us? I realize this may be a bit (pardon) confusing and I'm quite willing to elaborate if necessary. Or if someone else feels they can better un-muddle the situation, please feel free!

How about this: A 14 bit number looks like this -

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
EOPp n
```

```
-----
least significant byte      most significant byte
_ this must always be zero  _ this one, too (most sig. bits)
_ most sig. bit of LS 7bit byte  _ most sig bit of MS 7bit byte
least sig bit of LS7bit byte_  least sig bit of MS7bit byte_
```

Therefore the number represented above, written in traditional binary is: 00000010000000 (14 bit precision, decimal=128)

There, that's all I can manage at this point (hope I'm right)

-Charlie
---[0142]--- (pref = [0141], nref = [0143])

forum:
[0143] (16 lines) Weber.USITT 12/19/90 0604.7 mst Wed midi
Subject: Re: Comments
Charlie,

Thanks for the information. It is important for me to know the "standard" way of building multiple byte values in MIDI. I understand the concept of "14 bit values, sent least significant byte first." So, I think I know where to proceed from here.

However, something in the example mystifies me. 128 is a power of two. So any binary (or hex) representation of 128 should have only a single bit set. Therefore, the value sent for 128 being 00 01h followed by 00 01h seems wrong. It has two bits set. Should the value be 00 00h followed by 00 01h? Or, am I still missing something?

Thanks.

Ralph:
---[0143]--- (pref = [0142], nref = [0144])

forum:
p n

[0144] (25 lines) MDeakin.USITT 12/19/90 1108.6 mst Wed midi
Subject: Re: Comments
Hi.

The main point I want to raise this time is the one of resolution in the SET commands.

I think Charlie's suggestion in his message 138 of always sending 14 bits of data and using the number you need/want is ideal. By always starting with the most significant bit you will maintain compatibility between devices of differing resolutions.

I can't think of any lighting applications where 14 bits will not be enough. However there are many where 7 bits won't.

Anders, do we need to define the use of speed controllers more exactly? Are these to be speed changes relative to a stored speed or an absolute speed setting? In either case how are the settings to be interpreted?

Charlie, on the subject of the minimum sets. It was only a thought that GO_OFF should be included in the minimum set. On reflection if a unit needs it, and I think many of the small and simple units will be the type to use GO_OFF, then the manufacturer will implement it!

Cheers.

Matt.

---[0144]--- (pref = [0143], nref = [0145])

[0145] (7 lines) CRichmond.USITT 12/20/90 0028.2 mst Thu midi

Subject: Re: Comments

OOPS! Yes, you are right: 128 should be 00 00h followed by 00 01h.

I'd like to say this was intentional just to see if you were on your toes, but the truth is that this slipped by as I rambled. You will notice, however, that I did do it correctly in the elaboration of the example displaying the exact bit stream. Good work, Holmes!

- Watson

---[0145]--- (pref = [0144], nref = [0146])

[0146] (1 line) CRichmond.USITT 12/20/90 0031.0 mst Thu midi

Subject: Re: Comments

Matt- Sounds good - if you're happy, well at least we're all happier!

---[0146]--- (pref = [0145])

[0147] (10 lines) CRichmond.USITT 12/22/90 0058.0 mst Sat midi

Subject: Holidays

And the best of them to everyone attending this scintillating platform for profound dissertation!

I will be away until the 30th so will bid adieu for the holidays at this point and with luck will return for the New Year.

Thank you all for the very constructive work everyone has done in this forum in 1990 - I look forward to '91!

- Charlie Richmond

---[0147]---

[0148] (31 lines) AEkvall.USITT 01/03/91 0015.4 mst Thu midi

Subject: Comments

Charlie

There is no problem in having 7 and 14 bits instead of 8 and 16. The approach of sending as many bits as you need, leaving the others at 0, is fine.

I do not follow your correction of the 14-bit example, Watson. Your example was to set controller 128 to a value of 128. That must be 00 01 for controller 128, followed by 00 01 for the value as you said in your example. Everything in hex.

Matt

Yes, I think that we should define the speed control more precisely. Do you have an opinion on if it should be absolute or relative? In all our systems, the speed control works relatively, changing the speed from a preprogrammed value. How should a speed change be defined? In percentage up/down from the preprogrammed speed? +/- 1000 percent would be OK, I think. In our systems, when you move the speed controller down to the slowest speed, it will actually stop the fade completely. This is to provide a way of temporary stopping a fade. Is this a feature that should be defined? A special value for stopping could be -1000 percent (is that system is adopted). This means that when you move the the speed controller up from the stopped position, the fade will slowly start again without jumping.

That all folks

Anders

---[0148]--- (nref = [0149])

[0149] (35 lines) CRichmond.USITT 01/03/91 0105.6 mst Thu midi
Subject: Re: Comments

Well, are we all now adequately confused? I know I had to go back and look at the previous entries a few times before I figured out what was going on. I should have used a different controller number to set to a value of 128 (than 128) but did so for convenience of illustrations because it only necessitated fully printing out the bit map for one value (128). Then the whole thing began to unravel when the confusion began (besides, I have a lot of trouble thinking in hex because I'm not a programmer).

Anyway, Anders, you are correct - my original example was correct and the "correction" was incorrect. And I may just go back and delete the "correction" and this message at a later time to try to help keep things as straightforward as possible. But then that wouldn't be nearly as much fun or entertainment as leaving it all in here for everyone to see, would it?

How about the duplication of -1000% being defined as the same as STOP? I think you should have -1000% always truly mean "run the fade at 1/10 (?) the normal rate and if you want to have it stop completely then use the STOP command (with RESUME cancelling the STOP command and causing the transition to resume at the already established rate). Nevertheless, a large variation in rates may be defined and the boards _may_ respond in whatever manner they wish, but I think STOP should be the only command to completely halt the transition.

As an aside, we have two different ways of halting a transition: one is to stop the automatic sequence timing without stopping the continuation of the fade(s) and the second is to stop the fades without changing the automatic sequence timing (cue-to-cue). Obviously a third way combines these two. Therefore, we have two commands to perform these functions: STOP_CLOCK for the former and STOP for the latter. Adjustable rate controls are independent of these commands and exist for each function.

Cheers, Charlie.

---[0149]--- (pref = [0148])

[0150] (77 lines) Weber.USITT 01/13/91 1320.4 mst Sun midi

Subject: V0.02 of 2PC proposal

I have constructed a second draft of the two-phase commit enhancements to MIDI Show Control. This draft addresses Charlie's comments (found in entry 139) and my own concerns about the incompleteness of some of the discussions in the first draft. The new draft deals with the following major concerns:

1. The master/slave terminology has been changed to "controller" and "controlled device." I have made every effort to make the usage of the preferred terminology unambiguous.
2. The issues around 14-bit data values (as opposed to 16-bit data values) have been addressed. In most cases, I simply adopted the common MIDI usage of 14-bit data values. But, there are two exceptions:
 - i. Checksums are still computed using 16-bit arithmetic. However, before being transmitted the 16-bit checksum value is logically anded with 7F 7Fh. This last step eliminates the bits that cannot be transmitted in MIDI.

I debated between reducing the error detection coverage afforded by the checksum and increasing the number of MIDI bytes transmitted in the checksum. Reducing the coverage won.

- ii. Status code values are still 16-bit values with the high-order bit set for command_format independent values. In order to meet the MIDI 14-bit data value restrictions I have required that the low-order 2 bits of a status code value always be zero. Based on this change, I can use 2 MIDI bytes to transmit a status code value and reconstruct the value using the formula:

$$\text{status} = (\text{LSB} * 4) + (\text{MSB} * 512)$$

3. While working on the status code values problem, I discovered several errors in the first draft text. Incorrect status code values appeared some of the examples. The status code ranges (defined in Section 6.3) were not constructed so that all possible status code values were accounted for. These errors have been corrected in the new draft. Because the low-order two bits in a status code value must now be zero, all status code values have been changed (except 00 00). Also, two new status code values have been defined.

4. I have addressed my concern that error recovery methodology was not properly described in the first draft. Actually, this was done by adding Section 6.2, which covers several aspects of the two-phase commit methodology. Subsections of 6.2 cover the following topics:

- Normal Message Sequences
- Response Timeouts
- Exceptional Condition Handling
- Manual Override Processing
- Waiting for Messages

Some of this material is new, but some of it has been moved from other sections to the new Section 6.2.

I have asked Tim Clinton to place the new draft in:

>udd>USITT>common>standards>MSC_2PC_0.02

The new draft is approximately 62 Kbytes long (around 23 typewritten pages).

To aid those who have already read the first draft, I have placed change bars in the left hand column. The change bars mark lines whose contents are substantially different from the first draft. Insertion of the change bars was a totally manual process, and the distinction between substantial and insubstantial changes was based on my judgement. So, you may find places where you think the change bars are inaccurate.

Two symbols are used as change bars, | and !. | is used to mark new or revised material. ! is used to mark material that was reorganized as a result of adding Section 6.2.

\032

---[0150]--- (nref = [0151])

[0151] (1 line) Clinton.Thea 01/13/91 1956.3 mst Sun midi

Subject: Re: V0.02 of 2PC proposal

I have placed the file as requested, and it is now available to you.

---[0151]--- (pref = [0150], nref = [0153])

[0153] (28 lines) MDeakin.USITT 01/18/91 1059.4 mst Fri midi

Subject: Re: V0.02 of 2PC proposal

Hi.

1. Speed controls.

I agree Anders that this should be a relative change. The range of change you suggest (+/-1000%) is possibly a bit small. It all depends on the range of times allowed. If this speed control format is to be used for sequence step times as well as fade times then the range could be from 33mS (one frame at 30 frames per second) to several hours. So as I understand it +/- 1000% gives you a range of speed of 11 times the original speed to 1/11 times the original speed.

Is this enough? In theory if you wanted to speed up a fade time of 1 hour to a time of 0.5 seconds you would need a change of +719900%. Oh look its more than 14 bits! At least we should allow changes in the range +/- 8000% using the 14 bit number as a signed integer.

When it comes to Stopping a Fade or Sequence (and as I understand it we are talking about controlling speed changes to both here) the STOP command as currently defined only applies to Cues so would not be used for Sequences controlled by specific SET commands. We could use say -8001% as a stop for a Sequence or we could define yet more SET commands for Stop and Step functions for each Sequence. What do you think Anders?

That's all for now.

Matt.

---[0153]--- (pref = [0151])

[0154] (15 lines) AEkvall.USITT 01/23/91 1330.7 mst Wed midi
Subject: Comment

Hi,

some comments on Matts message 153.

+/- 8000% percent is OK.

I think that a common situation is that you slow down a fade to a complete stop and then release it again to start fade again. This should be possible to do with only one SET command. You should not have to switch to another SET command to stop the fade.

-8001% is a little bit odd, but why not?!

Anders

---[0154]---

[0155] (17 lines) CRichmond.USITT 01/26/91 1545.0 mst Sat midi
Subject: Approval!

I am pleased to announce that the MIDI Manufacturers Association has voted to approve the current proposal of MSC (V0.2) and has passed it on to the Japan MIDI Standards Committee for confirmation. This has been known to take some time in the past but we will be pushing them constantly to process the proposal and then it will be a standard!

Please note that there is a small change in the way MIDI Time Code is to be transmitted within MSC messages and this will be detailed when the final standard is published. Until then, we can't use it but this is good news and we can all get prepared. I'll let you know the moment it is finalized and the full standard will be uploaded to the common area here.

Thank you all for your hard work - this will be a major step in the development of technical control for live performance!

- Charlie.

---[0155]--- (nref = [0156])

[0156] (11 lines) KDHewitt.Thea 01/29/91 1018.2 mst Tue midi
Subject: Re: Approval!

Charlie... congratulations to you and the entire committee (on both side of the Atlantic).

I realize that everyone is busy, but it would be great if both CITT and USITT could get a short article summarizing the work of this committee so far, and how it used CallBoard to help develop this proposed standard, as well as what the standard might mean for "average" performing arts technicians.

I believe that Paul Court is working on the next issue of the CITT newsletter right now (hint, hint).

---[0156]--- (pref = [0155], nref = [0157])

[0157] (1 line) CRichmond.USITT 01/29/91 1546.6 mst Tue midi
Subject: Re: Approval!

Why, I don't know _what_ you mean.

---[0157]--- (pref = [0156], nref = [0158])

[0158] (10 lines) Brunelle.USITT 01/31/91 1554.3 mst Thu midi
Subject: Re: Approval!

Congratulations to everyone involved. I've been following this forum with great interest, although not understanding a word most of the time. My feeling is that this is a great moment for all of us who will have to deal with the new technologies of Show Control sooner or later. And this is definitely a great day for Callboard, as a major tool in this very "democratic" process.

Thank y Many thanks to all who participated, and again, congratulation.

Steph

---[0158]--- (pref = [0157], nref = [0159])

[0159] (26 lines) CRichmond.USITT 01/31/91 2310.0 mst Thu midi
Subject: Re: Approval!

I feel I must say this, even though MSC is not strictly involved at this point: we are pleased to announce that our "Stage Manager" (TM) show control software will be running the Vari*Lites for Miss Saigon on Bdway. This software will eventually be the vehicle with which we will provide users all the MIDI Show Control commands and tools which have been developed by this forum. Since it is not quite yet available (as we all know) we cannot use it, but other more cryptic MIDI messages can be created which are unique to Vari*Lites and inserted into the cue lists just like MSC commands will soon be able to be. Also, of course, Vari*Lites do not and cannot yet support MSC so must use their own commands.

But... the day will soon come where Stage Manager, standard software from one company, will be able to automatically command lighting systems from another company through easy to understand menus using regular English phrases and show designers won't have to become computer programmers to make things "jump through hoops"

Now, to partly answer a previous question, this is one thing that will be significant to the average theatre technician. We expect to be able EOPp n

to sell this software in its most basic form as a complete package including the computer and monitor for less than \$1000 every thing in. End of ad. Hope this is interesting to some of you - if not, we're doing the wrong thing and that's for sure.

TTFN, Charlie

---[0159]--- (pref = [0158], nref = [0160])

forum:

[0160] (1 line) Brunelle.USITT 02/01/91 1734.4 mst Fri midi

Subject: Re: Approval!

1000\$????????????

---[0160]--- (pref = [0159], nref = [0161])

[0161] (15 lines) CRichmond.USITT 02/01/91 1836.6 mst Fri midi

Subject: Re: Approval!

Yes, including a monochrome monitor, mouse and MIDI interface - however

I was inadvertently speaking in US funds since I was in Orlando mode.

By the way, this brings up the fact that we have opened a US office.

The address is:

Richmond Sound Design Inc.

5750 Major Blvd. Suite 320

Orlando, FL 32819

Phone 407/352-0458 Fax 407/345-1621

This is our US base and all US sales will be done from here. Canadian price is approximately \$1200 (plus GST, of course!) RUSH your orders to the appropriate location.

Cheers, Charlie.

---[0161]--- (pref = [0160], nref = [0162])

[0162] (49 lines) Brunelle.USITT 02/02/91 1532.4 mst Sat midi

Subject: Re: Approval!

Let me get this straight, Charlie. Are we talking about a dedicated machine, or a personal computer with proprietary software? If it's a computer, is it one of your beloved Amigas or what? How can you make such a deal possible? It just sounds incredible.

I have a few questions for you. Now that the MSC protocol has been approved, is there some kind of lobbying being done so that the manufacturers of lighting, sound, rigging, pyro and other live shows gadgets implement the protocol in their machines? I know that the Japanese association still has to approve it, but by the time it's done, it would be nice if the applications could come out on the market. I guess the manufacturers will need time to implement this, so why not start the R&D now?

I have the feeling that up to now, sophisticated show control has been used mainly in performances that were not dependant on actors or musicians, but rather on video or film that made the show be exactly the same every time. As you know, we will need a lot of flexibility in the theatre business. We will want automated luminaires cues to be triggered by the lighting board, lighting cues triggered by a limit switch on a door, pyro cues triggered by a sequencer to be exactly on the beat, and there will be complex sequences of lighting, sound and motors cues that we will want triggered by the stage manager. I guess this is all possible, but how difficult will it be to program? How difficult will it be to edit that program? What will be the part of the controller in that process, and what will be left to the controlled devices? Do you think that the average light boards will look almost the same, or will they be more complicated to program? In other words, will the process be transparent for the operators of the controlled

devices? Do you think that we will have to add an other operator on the show crew to operate the controller? I'd also like to have your opinion on the reaction of the crews to this technological change. The guy that 20 years ago used to work with his two hands and two feet to handle his huge 4 preset board will now just sit behind his desk and watch the controller do the job? How safe would it be to leave everything to the controller?

That's a long list of questions... I'm anxious to hear your answers not because I'm skeptic, but because I'm an enthusiast. I believe that with this new technology, and because a standard is set and adopted (or will be soon), in a very near future we will be able to do MORE and BETTER with our equipments, giving to the directors and designers what they always wanted to get but were afraid to ask for.

Anyway, if you think that this discussion is too philosophical, I'll see you in Boston and we can discuss this in the appropriate setting... A bar.

Steph

---[0162]--- (pref = [0161], nref = [0163])

[0163] (110 lines) CRichmond.USITT 02/03/91 0020.3 mst Sun midi

Subject: Re: Approval!

Yes - so many questions! Let's see now...

Yes it is a beloved Amiga. The hardware package is a standard A500 with 512K RAM and single floppy drive (list CDN\$899), a mono monitor (list CDN \$100) and Stage Manager 500 software (list CDN \$200). I guess at list price you can't include the MIDI interface (RSD model MD-101, list CDN\$150) but you don't have to buy the hardware from us. At the moment, Stage Manager does not include MIDI Show Control Commands because it can't - but it will the instant MSC is approved (and the price will go up). I know a number of other manufacturers (lighting exclusively) are working now at having their systems completely ready for MSC when it arrives. So much for lobbying.

The most effective form of lobbying for a manufacturer is a customer who says "If you make xxx, I will buy it" So it's primarily up to the end user to get things going at this stage.

Actually film and video production have not used these techniques. Their modus operandus normally falls into the category of "have a whole bunch of special effects technicians work like hell to make something happen just once exactly the way the director likes it and hope the cameras are rolling when it does" Repeating it after that is not only pointless but wouldn't be asked for unless the shot was unusable. The places that have worked hardest on show control have traditionally been the theme parks - and indeed, until lately, they have not used live action or actors. The approach has been to mechanize a ride or attraction to the Nth degree and synchronize everything with time code so it all happens exactly the same way every time - and of course this has been completely unworkable when applied to live shows. That's where the kind of techniques used in live theatre technology come in and are now being used by theme parks for these types of shows.

The specific effects you describe are in fact being done right now by computer controlled systems running live shows at theme parks. The

difference between how they currently do it and the way they will do it once more manufacturers support MSC is in the ease of designing a custom control system, the ease of programming and editing it and in the cost of such an undertaking. Right now, these systems must be custom engineered from scratch and programmed by computer technicians because all the equipment must be assembled from many different sources all using different communication standards and they must be made to talk to each other. When technical theatre equipment starts talking the same language and uses standard connectors and cables, then the average theatre technician will be able to hook up fairly complex systems using equipment "right out of the box"

As far as our software is concerned, programming and editing will be done almost exclusively with pull-down menus and screens displaying commands and options in plain English (sorry, French isn't available yet) and will require virtually no knowledge of the nuts and bolts approach currently required to make two different devices talk to each other.

The controller will basically assume the role that the stage manager's prompt script in combination with the headset/squawk box system. In other words, the cues the stage manager normally "stores" in their script will be stored electronically and when the stage manager presses the correct button, the cue in standby will be transmitted electronically to the console or controlled device handling that kind of cue. The information transmitted will be the electronic version of "Light Cue 39 - GO" NOT "Lights - put Channels 3,5,6,18 at a level of 68 in a 15 count (or some such similar data)". Therefore what I am saying is that initially the controlled device will continue to do exactly what it has always done: store all cue data and do the dedicated control of its media and simply execute cues according to preprogrammed information when told to via an operator's button push or via a MSC go.

MSC does have the capability built in to ultimately transfer actual cue data including individual channel, dimmer, fade rate, time, chase, etc. information but I do not see much equipment talking via MSC in this manner very soon (except that single manufacturers will start using MSC to allow 2 or more consoles of their own make to act like a single large console when they are connected together - but the response of other manufacturers consoles may not be quite as predictable when used together) WE are however, attempting to define the commands in a manner specific enough that this capability will eventually be quite possible. Much like MIDI works with music - middle C is usually middle C - beyond that, experimentation is required to find out how things respond but once found out can be repeated reliably.

Initially, the light board will look much the same - except for an increasing number of software functions which will be viewed and controlled via the small window and keypad found on more and more consoles these days. This will expand as the entire industry gets more and more comfortable with the power available. I think the person running the controller will be same one as controls the show today - the stage manager (hence the name of our software) and I also think the technicians will be quite enthusiastic about the advent of all of this. After all, it takes a lot of the pressure off the operator. They will be able to assume the role of quality controller or process monitor and will be able to concentrate on any special live effects and cues which must be taken directly for one reason or another. The rote, boring and

mechanical cues will be performed automatically and without error whereas the ones requiring special feel or live interaction can be focused upon without distraction.

I always felt I made more errors as an operator when I was bored or over worked. The intent of this is to eliminate the performing of boring cues and reduce the total number of cues the operator has to actually concentrate on. If things go as planned, the computer systems should perform more reliably and therefore be safer and the operators should too.

This does not mean that, because I have attempted to answer your questions here in the forum, we can't also talk in Boston (over chowder).

Your comments are thought provoking and valuable. Thank you.

- Charlie

---[0163]--- (pref = [0162], nref = [0164])

EOPp n

forum:

[0164] (14 lines) Brunelle.USITT 02/04/91 1546.3 mst Mon midi

Subject: Re: Approval!

Thank you very much for taking the time to answer my questions. While we are at it, here are some other ones. You say that lighting manufacturers only have started working on MSC. Are you working on sound devices such as mixing boards or digital recorders that would talk MSC? That is one area of application that would be very interesting for us. We've been looking at the automation of our sound devices for a long time. Are there already sound automation devices on the market that could easily implement MSC and therefore take cues from the controller? Since most sound devices already have MIDI, is it easier to implement MSC in these?

Thank you in advance.

Steph.

---[0164]--- (pref = [0163], nref = [0165])

[0165] (17 lines) ABall.USITT 02/04/91 1719.2 mst Mon midi

Subject: Re: Approval!

Uh, Charlie...just a couple of questions...any plans to port the software to other commercially available machines and sound boards (One begins with an "I", the other with an "M"...or is that an "A" ?) ? I'd be interested if you had a demo disk that ran on something other than a computer that I had to *buy* in order to run the demo...

Also, since I have a stable of "I"s and "M"s in-house, if one machine goes down, I simply swap out the MIDI controller card and boot the software on another machine.

I know that the Amiga has a MIDI port, but I'm curious about other machines none-the-less...

Congrats on the approval ! Your discussions have been an endless source of entertainment, if not profound confusion (on my part). Thanks again.

Andrew

---[0165]--- (pref = [0164], nref = [0166])

[0166] (22 lines) CRichmond.USITT 02/04/91 2242.9 mst Mon midi

Subject: Re: Approval!

Answers to Steph: We have been manufacturing a virtual audio mixing console system (to describe it one way) that can have up to 4000 channels of audio control for the last 4 years. It currently talks its own brand of MSC as well as all other versions of other manufacturers' MIDI implementation so that you can control MIDI devices (and many other kinds as well) on a cue- to-cue basis during the show. It is used in many applications where it controls DAT machines, video disc players, hard disk recorders, synthesizers, samplers, etc. as well as non-audio systems such as lighting, Vari*Lites, film projectors, slide projectors, stage mechanics, pyro, fireworks, smoke, and other special effects. If you would like to get more information on this please let me know. I can get your address from the info file here on Callboard if it is current. I apologize for not letting you know before this but I thought you had seen our previous "commercial" message right around the time you started logging in.

In short, COMMAND/CUE will be able to respond to MSC commands as well as be the controller simultaneously as soon as it is officially approved.

Thanks for asking.

-Charlie

---[0166]--- (pref = [0165], nref = [0167])

[0167] (53 lines) CRichmond.USITT 02/04/91 2309.1 mst Mon midi

Subject: Re: Approval!

Answers to Andrew: Surprisingly enough, we've had this question before. As far as COMMAND/CUE is concerned- no. This software could not exist if the Amiga did not exist. The multi-tasking operating system provides this machine with the power and flexibility required for a complex real-time show control system. The cost of the computer is slight compared to the overall value of most systems and the computer is not in a position in most venues to be useful for anything else other than a dedicated controller of the show control system. I have to say that this question continues to bug the hell out of me. How many people complain because the processor inside their memory lighting board hasn't been made accessible to do word processing on; How many people complain because their memory lighting board doesn't have an IBM PC running it? Really, I'm getting a little tired of people complaining just because we happened to choose the best independent platform of the moment. As far as Stage Manager goes

As far as Stage Manager goes, here's the situation: we may port this to another platform, but if we do the cost of the program will be several times higher than the Amiga version. We feel that most people will continue to choose to buy the software and the computer that is going to

EOPp n
be tied up running the show for the same price as just the software will be that will run on the office machine. I know the grammar is terrible but I hope you get the idea. I also know that the question has not gone away despite the fact that it should have. Not content with ignoring the realities of the world, we will probably bend once we've got some time to breathe. A warning about IBM's: to talk MIDI they require an

internal card which alone typically costs almost as much as an Amiga (and, horrors, neither IBM or Apple make it!). I fully acknowledge the market position of these makes, but do not feel in this day and age it should constitute such an overriding reason for choosing a piece of hardware. Operating systems are neither here nor there in this argument since the software that does the job is designed for one particular platform: just like the powerful video production and animation software available for the Amiga which has put the Amiga in the position of being the dominant PC platform in that particular area (this is true - just ask anyone who knows this area of specialty).

For the record as well: regarding the stability of Commodore. The US market is the only one in which it has a small percentage of the total sales. In Germany and Japan, thought to be the most sophisticated consumer markets in the world, Commodore is2 and6, respectively, and only IBM has higher consumer PC sales in all of Europe.

Finally, I'm sure there is a store not too far from you that has an Amiga in it, so if you really want to see the demo disk, you could take it there. I'm sure the salespeople and other customers would also find it quite interesting....

As usual I've overdone it; but then what did you expect????

Thanks for asking.

-Charlie

---[0167]--- (pref = [0166], nref = [0168])

forum:

[0168] (6 lines) Brunelle.USITT 02/05/91 1535.4 mst Tue midi

Subject: Re: Approval!

Charlie, one last question before Boston.

What is the physical link between your virtual audio mixing board and the real life sound? When I do something on the Amiga screen, what kind of hardware does the job? A real board? Some mysterious rack-mounted black box? O.K., I'm through, don't shoot.

---[0168]--- (pref = [0167], nref = [0169])

[0169] (21 lines) JLBracewell.USITT 02/05/91 2151.2 mst Tue midi

Subject: Re: Approval!

I really should keep my yap shut and let Charlie answer this one, but I've been a good boy and stayed quiet for a long time. I have an early version of Command Cue. The Amiga screen represents controls for a group of electronics which actually handles the audio signal. The relationship really is the same as the control board on any other mixing console. It's just harder for people to think of that way.

In my case, I laid out the configuration of hardware -- Autopan's (Charlie's patented name for the device that actually accomplishes fades, level controls, routing switches, and the grouping of output level controls to form presets -- and the electronics "black box" and the Amiga screen (to some extent) were tailored to my specified configuration.

The biggest difference between the representation on the Amiga screen

and a "real" mixing console is that you don't get to see all of the controls at once, and you have to get to each control's operating screen before you can manipulate that control. This can take a little bit of time in my version of the system, but the repeatability and the range of actions that the system can accomplish really is phenomenal.

---[0169]--- (pref = [0168], nref = [0170])

[0170] (6 lines) Brunelle.USITT 02/05/91 2209.6 mst Tue midi

Subject: Re: Approval!

Since you opened the door, Mr. Bracewell, I'll put my foot in. If i get you right, the physical link between your Amiga and the device you call "Autopan" is a data cable... right? And all your actual input and output are physically plugged into this "Autopan" device... right? Now, what type of input do you usually have? Tapes, mics, CDs, DAT? How many inputs, how many outputs?

---[0170]--- (pref = [0169], nref = [0171])

[0171] (19 lines) ABall.USITT 02/06/91 0830.6 mst Wed midi

Subject: Re: Approval!

I think that you may have misinterpreted the motive behind my questions, Charlie. I have no brand loyalty when it comes to PCs. And I'm sure that the Amiga is a fine machine - after all, you chose to run your product on it.

My point is, and perhaps this is not an appropriate forum for this but you are the techno-whiz after all (-; , why not choose a broadly-based computer platform to run your innovative software on. Repair, parts and servicing costs are then kept to a minimum, availability is wide-spread, and if your machine breaks (or the road-crate falls from the back of the truck...) you can rent/steal/borrow the machines necessary to get your show up and running.

I should point out that I often wonder the same thing about lighting consoles...

Just wonderin'

Andrew

---[0171]--- (pref = [0170], nref = [0172])

EOPp n

forum:

[0172] (31 lines) CRichmond.USITT 02/07/91 0105.1 mst Thu midi

Subject: Re: Approval!

I'll try to do us all a favour and stick to the subject. The physical link between the computer and the black box (not very mysterious - all schematics are supplied) may take the form of the standard 34 conductor ribbon cable (similar to a printer cable but flat) for short (up to a maximum of 15-30 metres in optimal conditions) runs or we can provide a fibre optic link which in effect extends the ribbon cable by 2km per segment. Any number of black boxes may be connected to the computer either in close or distant proximity since the ribbon cable system can be simply paralleled to talk to more units. This means that the control computer may be located where ever it is most easily operated and the audio control hardware (consisting not only of Auto-Pans but also, faders, switches, assignors, selectors plus non-audio outputs such as TTL data outputs, 0-10V dimmer outputs and motor servo controls) may be located anywhere convenient. This reduces the wiring runs and any

propensity to noise pickup because of long audio cables. The fibre optic connection also has the side benefit of completely electrically isolating the computer from the audio hardware (but this is usually no problem anyway when the grounding system is properly managed according to instructions.)

EOPp n

A side note: a "real" console can also be operated by the system in two ways: 1. An audio board may be interfaced with the black box hardware on a channel-to-channel basis so that the computer controlled functions may be applied to audio signals that are controlled with conventional facilities as well. 2. A console with MIDI controlled muting or other capabilities may be controlled by the MIDI cues that can be interspersed with regular ones throughout the cue lists.

I may bark but I don't bite. Ask more - you'll only get answers here!

- Charlie.

---[0172]--- (pref = [0171], nref = [0173])

forum:

[0173] (29 lines) CRichmond.USITT 02/07/91 0118.9 mst Thu midi

Subject: Re: Approval!

For Andrew's question, again I'll try to stick to the facts, man.

Strictly speaking, we required an innovative platform for our innovative software. The broadly-based platforms just didn't have the guts and we felt that we'd rather do it than not. Call it gutsy but really the Amiga is not that hard to find nor is it expensive to service. Two of them (one to run the show and one as a spare) have been supplied to several nervous customers (without necessity to date) and together they are still cheaper than a super powered 386.

Have you asked your questions of lighting people? I would be curious about their answers, not really knowing anything about the rationale involved in their marketing or engineering. Would you feel more comfortable if we put the computer we used into our own box and put our name on it without telling you anything about it - like the lighting people do? Strangely enough, we rejected that idea because we felt our customers would appreciate the fact that they could just run out and get it fixed or replaced much more easily than if it were only available from us. Instead, we get all these questions about why it isn't a PC. Having done this first in 1985 and now seeing some lighting controllers become available on standard PC's, we think we may have started something (not that we'll get credit for it) but we also note that the PC lighting software is not particularly high power.

Anyway, you now know everything I do about this decision...

Thanks again for asking..

-Charlie

---[0173]--- (pref = [0172], nref = [0174])

[0174] (39 lines) JLBracewell.USITT 02/07/91 0735.8 mst Thu midi

Subject: Re: Approval!

Charlie and I had a rather protracted go-round on this subject when the Command Cue system was in early stages, Andrew. Also, we had arguments

about ability to take manual control of any portion of a cue (which is still a problem for me with my version of the system, by the way.) After much discussion, I had to (grudgingly) admit that Charlie had done his homework. Macs and IBMs aren't set up to handle the kinds of things Charlie does. At least, I've never seen a system set up -- certainly not for the price of an Amiga -- that would do anything comparable.

Would I rather Charlie would crate everything up in a proprietary box and put his name on it? No, probably not; but I have been thinking one thing since we got our new ETC Expression: I wish ETC had essentially given us a PC as part of their board, so that you could have all of the capabilities of a PC available in the system. It's so close! In the case of Command Cue, I think I still wish that there were some hardware controls as part of the interface that allowed the operator to get to something fast. I'm still taking flak at Ithaca College about a rather celebrated cow noise. The problem was that no matter how I tried to set it before we got into rehearsal, it never came out right once we ran the effect as part of the performance. Usually it was just too loud, and there's NO way to react quickly enough to pull the level of a short cue down with the mouse -- at least not in my system. Finally one night the director stood up and screamed "I never want to hear that sound again! Cut it!" We did, but I was known as Dr. Moo for a while.

I understand that the next version of the system, with assignable masters in software, makes some of those problems easier. In all fairness, Charlie tells me that when they want to know how far they've come, they go back and look at the version of the system that I'm still using. Would that I could persuade Ithaca College to let me upgrade. I support Command Cue as a very useful tool for sound. I've just often thought that there must be some better way, but everytime I try to think it out any other way, I become convinced that Charlie was pretty thorough.

Mr. Brunelle, I'm sorry that I wasn't totally clear in my response to your question. Evidently I left you with the impression that the black box was called the AutoPan. As Charlie explained, AutoPan is only one of the modules contained in the Command Cue hardware "black box."
---[0174]--- (pref = [0173], nref = [0175])

[0175] (25 lines) ABall.USITT 02/07/91 0922.1 mst Thu midi
Subject: Re: Approval!
To Mssrs. Brunelle and Richmond: Thanks for your concise answers.

Re: PC-based control. the only information that I have been able to e

I used to think that an 8088 or 68000 would have a difficult time keeping up with the instructions from complex lighting software - Silicon Graphics et al have realized that it's easier to drive machine-based instructions at a very low level than write high-level software which runs about as fast as your average glacier. But with so many new processors to choose from, some of which are more than capable of complex multi-tasking (Intel 80386, Motorola 68030) why do lighting control companies leave the computer R&D to computer what they do best companies (where there is a sufficient user base to pay for the expenses of R & D) and concentrate on better, faster interface boards and excellent software.

Just re-read the last paragraph. That should be "why don't lighting

control...".

I have not yet heard the reasoning behind the desire to use a dedicated processor that would convince me. Anyhow...

...Just askin'...

Andrew

---[0175]--- (pref = [0174], nref = [0176])

[0176] (80 lines) CRichmond.USITT 02/08/91 0035.0 mst Fri midi

Subject: Re: Approval!

Andrew, when Dr. Bracewell talks about "the next version" he is referring to a software version which is now about 2 years old. This has both on- screen assignable submasters and the ability to control these submasters through MIDI remote control devices such as the JLC Cooper Fader Master and the Lexicon MRC. This gives the system an expandable complement of "hardware controls" and the version we are currently working on will use MIDI Show Control to be able to control virtually any parameter of the system with a MIDI remote control device.

To be fair, when a programmer writes machine code on a raw microprocessor system, it is often easier than when they try to do so on a more complex generalized PC which uses an "operating system" whose primary purpose in life is to provide convenient ways to make common system requests without having to generate all the machine code from scratch. Since the programmer knows exactly what the system is required to do, they can write very efficiently, putting in only the functions needed by the lighting program (or whatever). Multi-tasking can even be written in if the programmer is up to it and it could be a better choice than using a multi-tasking third party shell. The problem programmers have when writing in an environment which was created by other programmers is that they were not the ones who created everything the program does and therefore it can be much harder to debug and to really figure out what the results will be.

Operating systems and environments are primary intended to allow programmers to more quickly write powerful applications using high level languages where the cost/speed tradeoff is weighted more toward cost. In the types of programs that run lighting consoles and show/sound control systems, speed is absolutely the primary goal. For example, to this end we have written COMMAND/CUE entirely in assembler and sometimes even written our own operating system calls instead of using the resident ones simply to speed things up. The most current version of the code compiles and links into a primary file of over 300K, with several other secondary modules in the range of 50K to 100K that loaded in as necessary. Anyone who is familiar with assembly language programming will acknowledge this as a major effort. The result is a very speedy real-time system that can change the virtual and actual volume levels in over 4000 audio channels in less than 10 milliseconds-something that would probably be impossible to do even in the "C" language on the very fastest computers around.

It is the constantly changing environment of the PC operating system that has probably driven most developers of lighting systems to stay with a proprietary system over which they have total control. Within the computer community, complaints about bugs in various versions of even the most popular operating systems are legendary - and MSDOS is one

of the best known offenders, if not the worst. The last thing a lighting mfr. (or anyone running a show) wants to have happen is have the system crash because of an operating system bug over which they have little control. All we can do is produce the most rock-solid system possible using extensive beta-testing and providing updates to anyone experiencing any difficulties. Such is the nature of the beast. Luckily, the Amiga operating system is a good one and the multitasking, while not easy to write for, provides a really powerful and well thought out toolkit.

Enough.

Brad Rodriguez, you must have something to say about this! Ralph Weber, I'll bet you could contribute a lot here (and tell me if I'm wrong, not actually being a programmer myself). Also, Ralph we are looking forward to your latest incarnation of your proposal. There are more and more people waiting to see if it will be something we can continue to expand upon for audio equipment intercommunication (as a better substitute for the "PA422" application").

Finally, I have good news about MIDI Show Control. The relationship between the MMA and the Japan MIDI Standards Committee has been redefined. Now, unless the JMSC has specific concerns with proposals put to them by the MMA (and vice versa), such proposals automatically are approved after 60 days. This means we could see MSC become a standard in little over a month. In fact, the JMSC has said they will specifically present the MSC proposal to the "Japanese equivalent of the USITT" for comment and possible suggestions. It will be interesting to see what transpires.

Anyway, everybody better get ready for this - it's happening!

- Charlie

---[0176]--- (pref = [0175], nref = [0177])

[0177] (9 lines) ABall.USITT 02/09/91 1009.3 mst Sat midi

Subject: Re: Approval!

Thanks Charlie. Indeed, I can appreciate the difficulty of programming 300K in assembler. Ted Nelson once said that programming in BASIC was like juggling bowling pins - programming in assembler was like juggling straight razors. And as usual, you have eloquently answered my questions.

Thanks again.

Andrew

---[0177]--- (pref = [0176])

[0178] (116 lines) BRodriguez.USITT 02/09/91 1357.6 mst Sat midi

Subject: PCs vs dedicated hardware

Well, I haven't been able to keep up with this discussion, beyond making sure that I download it regularly for future consumption. But I did pick up on the PC debate. Charlie has covered many of the issues quite well in his last message. Maybe I can bring up a few more, and provide more background on the use of PCs for lighting systems.

First, PLEASE dispel the belief that an "off-the-shelf" PC can run

lighting control. All PCs require expansion boards to handle the funny protocols (AMX, DMX) we've invented for lighting control. And these boards are not stocked at your local PC service center. (I sometimes wonder if they void the warranty.)

Now, let me bring up some hardware reasons, software reasons, and operating/marketing reasons why PCs may not be suitable for lighting control. The validity of these points I leave to your judgment.

Hardware reason1: superfluous hardware. Most PCs contain electronics on the motherboard which is of no use to a lighting controller. (Granted, on a PC clone the cost of this hardware is negligible, but it does take up space and power.)

Hardware reason2: inappropriate hardware. For example, most PCs use dynamic RAM, which can't be preserved through a power failure. (At least, not without using a car battery to supply full system power.)

Hardware reason3: missing hardware. No PC has all of the hardware and I/O needed for lighting control -- such as AMX or DMX outputs, or inputs for a large number of slidepots.

Hardware reason4: inappropriate CPU. You may find this hard to believe, but the 8086 and 68000 are awful processors to use for lighting control. Mainly this is because of their slow multiply instruction. (BTW, The 80286 is much better at this, which I suspect is why all of the "higher" Prestige boards use an AT motherboard.)

Hardware reason5: badly documented. This mainly applies to "PC clone" motherboards and peripheral boards. The manufacturers of these don't provide timing information, or any specifications at all except "compatible with IBM PC". (And that's not even always true.) This makes it hard to design add-in hardware that will work reliably.

Hardware reason6: not stable. In addition to being badly documented, timing and electrical specifications for "clones" are subject to change EOPp without notice, or from manufacturer to manufacturer. (I've been burned by this in the past.)

Hardware reason7: difficult to service. Mainly because of the lack of documentation, and lack of stability, but also because of the fact that most clones use custom silicon these days. You can't replace a chip. You can't even probe a chip to see if it's causing the problem. (This is more a concern for the factory, since you don't usually do this kind of service in the field. But wouldn't you like schematics for your board?)

Software reason1: inappropriate operating system. If you choose to use the built-in BIOS, you may find annoyances (such as the power-up memory check) which have no place in a control system.

Software reason2: operating system overhead. Charlie mentioned this: a badly designed operating system will slow down an application. I've seen an application running on an IBM PC under MS-DOS, without MS-DOS but using the BIOS, and without even the BIOS. You wouldn't believe the difference in speed. (Of course, you CAN program around this -- but this can require just as much programming effort as starting from

scratch.)

Operating/marketing reason1: lighting boards shouldn't look like PCs -- or at least this has been the verdict of the marketplace. People want their lighting boards to look like old-time lighting boards, and with the exception of a techno-literate few, recoil in horror from anything resembling a computer. (Does anyone appreciate the risk Strand took when they introduced the first Palette?)

Operating/marketing reason2: inappropriate user interface. Even if you're not afraid of PCs, you don't use a keyboard to control lights, you use slidepots and perhaps wheels. So most of the PC gets shoved under the desk. (Charlie took a big risk himself with Command Cue, and, in my experience, sound people are much more receptive to technological innovation, and to changing their ways, than are lighting people.)

Operating reason3: the joy of dedicated computers. It sounds great in principle to be able to use your lighting system for word processing. But wait until someone is using it for one thing, and you want to use it for another. (I wish everyone here could have experienced the Old Days of computer science -- before practical timesharing or PCs -- when we had to sign up for computer time in 1-hour slots.) Be thankful that you don't need to share your lighting board.

Charlie has avoided some of these problems by making his machine look like a computer, and expecting the user to make the adjustment. (A gutsy move, Charlie.) He also picked a machine very well suited to his application -- with MIDI ports, a clean hardware design, and a multitasking operating system. His application exploits the strengths of the operating system -- such as the graphical user interface -- and avoids the weaknesses of the hardware -- such as lousy eight-bit math. Since he didn't use a clone, he can get good service info, and the design is quite stable. A few problems (such as nonvolatile memory) I suspect he sidestepped, but perhaps I should let him comment about that.

Such is the difference between sound and lighting.

As you must have noticed by now, I favor dedicated hardware for lighting control. It all boils down to this: in my engineering judgment, hardware specifically designed for this application does a better job than PCs pressed into service.

- Brad

P.S. Charlie, multitasking operating systems aren't hard at all to write. (I've done a few.) It's the user interface which is the killer.

P.P.S. I'd be delighted to continue this discussion with anyone who's interested, but let's move it to a different forum and let this one get back to MIDI.

---[0178]--- (nref = [0179])

forum:

[0178] (116 lines) BRodriguez.USITT 02/09/91 1357.6 mst Sat midi

Subject: PCs vs dedicated hardware

Well, I haven't been able to keep up with this discussion, beyond making

sure that I download it regularly for future consumption. But I did pick up on the PC debate. Charlie has covered many of the issues quite well in his last message. Maybe I can bring up a few more, and provide more background on the use of PCs for lighting systems.

First, PLEASE dispel the belief that an "off-the-shelf" PC can run lighting control. All PCs require expansion boards to handle the funny protocols (AMX, DMX) we've invented for lighting control. And these boards are not stocked at your local PC service center. (I sometimes wonder if they void the warranty.)

Now, let me bring up some hardware reasons, software reasons, and operating/marketing reasons why PCs may not be suitable for lighting control. The validity of these points I leave to your judgment.

Hardware reason1: superfluous hardware. Most PCs contain electronics on the motherboard which is of no use to a lighting controller. (Granted, on a PC clone the cost of this hardware is negligible, but it does take up space and power.)

Hardware reason2: inappropriate hardware. For example, most PCs use dynamic RAM, which can't be preserved through a power failure. (At least, not without using a car battery to supply full system power.)

Hardware reason3: missing hardware. No PC has all of the hardware and I/O needed for lighting control -- such as AMX or DMX outputs, or inputs for a large number of slidepots.

Hardware reason4: inappropriate CPU. You may find this hard to believe, but the 8086 and 68000 are awful processors to use for lighting control. Mainly this is because of their slow multiply instruction. (BTW, The 80286 is much better at this, which I suspect is why all of the "higher" Prestige boards use an AT motherboard.)

Hardware reason5: badly documented. This mainly applies to "PC clone" motherboards and peripheral boards. The manufacturers of these don't provide timing information, or any specifications at all except "compatible with IBM PC". (And that's not even always true.) This makes it hard to design add-in hardware that will work reliably.

Hardware reason6: not stable. In addition to being badly documented, timing and electrical specifications for "clones" are subject to change without notice, or from manufacturer to manufacturer. (I've been burned by this in the past.)

Hardware reason7: difficult to service. Mainly because of the lack of documentation, and lack of stability, but also because of the fact that most clones use custom silicon these days. You can't replace a chip. You can't even probe a chip to see if it's causing the problem. (This is more a concern for the factory, since you don't usually do this kind of service in the field. But wouldn't you like schematics for your board?)

Software reason1: inappropriate operating system. If you choose to use the built-in BIOS, you may find annoyances (such as the power-up memory check) which have no place in a control system.

Software reason2: operating system overhead. Charlie mentioned this:

a badly designed operating system will slow down an application. I've seen an application running on an IBM PC under MS-DOS, without MS-DOS but using the BIOS, and without even the BIOS. You wouldn't believe the difference in speed. (Of course, you CAN program around this -- but this can require just as much programming effort as starting from scratch.)

Operating/marketing reason1: lighting boards shouldn't look like PCs -- or at least this has been the verdict of the marketplace. People want their lighting boards to look like old-time lighting boards, and with the exception of a techno-literate few, recoil in horror from anything resembling a computer. (Does anyone appreciate the risk Strand took when they introduced the first Palette?)

Operating/marketing reason2: inappropriate user interface. Even if you're not afraid of PCs, you don't use a keyboard to control lights, you use slidepots and perhaps wheels. So most of the PC gets shoved under the desk. (Charlie took a big risk himself with Command Cue, and, in my experience, sound people are much more receptive to technological innovation, and to changing their ways, than are lighting people.)

Operating reason3: the joy of dedicated computers. It sounds great in principle to be able to use your lighting system for word processing. But wait until someone is using it for one thing, and you want to use it for another. (I wish everyone here could have experienced the Old Days of computer science -- before practical timesharing or PCs -- when we had to sign up for computer time in 1-hour slots.) Be thankful that you don't need to share your lighting board.

Charlie has avoided some of these problems by making his machine look like a computer, and expecting the user to make the adjustment. (A gutsy move, Charlie.) He also picked a machine very well suited to his application -- with MIDI ports, a clean hardware design, and a multitasking operating system. His application exploits the strengths of the operating system -- such as the graphical user interface -- and avoids the weaknesses of the hardware -- such as lousy eight-bit math. Since he didn't use a clone, he can get good service info, and the design is quite stable. A few problems (such as nonvolatile memory) I suspect he sidestepped, but perhaps I should let him comment about that.

Such is the difference between sound and lighting.

As you must have noticed by now, I favor dedicated hardware for lighting control. It all boils down to this: in my engineering judgment, hardware specifically designed for this application does a better job than PCs pressed into service.

- Brad

P.S. Charlie, multitasking operating systems aren't hard at all to write. (I've done a few.) It's the user interface which is the killer.

P.P.S. I'd be delighted to continue this discussion with anyone who's interested, but let's move it to a different forum and let this one get back to MIDI.

---[0178]--- (nref = [0179])

[0179] (29 lines) CRichmond.USITT 02/10/91 0027.2 mst Sun midi
Subject: Re: PCs vs dedicated hardware
Only one subject left open on that last excellent transaction - I'll
close it and then we _should_ move on (somewhere else?):

Fortunately the combined power requirements of the computer and a very substantial sound control system (the "black box") are minimal, usually less than 500 Watts and never more than 1000. Therefore, in installations which are critical, the entire system is put on a low cost UPS (uninterruptible power supply) and lightning, glitches, brown-outs and temporary black outs are easily handled. Additional audio equipment such as effects equipment, EPROM playback, disc players, crossovers, delay units, MIDI remotes, etc. generally tend to have built in battery-backed up RAM so that they can recover quickly and independently. We don't attempt to keep power amplifiers on during black outs, but they generally recover instantly (there is not a general need to continue performing the show in the dark) and a couple of amplifiers for emergency evacuation or paging can be put on the UPS circuit.

One other solution: to avoid down time in the event of a hard disk crash (which does happen with the best of them occasionally), a complete second computer with the show loaded is kept standing by. The current show cue location can be loaded within seconds on the standby machine and the entire system switched to that unit with a couple of generic changeover switches (parallel printer style and RS-232 style for MIDI).

Thank you for an excellent synopsis of the technology, Brad. See you in Boston!

- Charlie

---[0179]--- (pref = [0178])

[0180] (36 lines) Weber.USITT 02/10/91 1245.2 mst Sun midi
Subject: MSC 2PC and other things
Charlie,

I have been following the discussion of your various systems. I have not commented previously because I never quarrel with someone's choice of computer and operating system platform. So long as someone does not claim software portability when it does not exist (and you definitely have not done that), I think everybody is entitled to use the system that they think best.

I can also appreciate the requirements that COMMAND/CUE places on its platform. I accept at face value your statement that the Amiga system best satisfies these needs. At the same time, having such a system that was also portable to other hardware platforms would be a significant advantage.

I would like to believe that the lighting console folks provide important added value to their systems by including specialized operator interfaces (such as fader wheels) on their systems. This added value makes exposing the basic system interface more difficult (or more risky). Since I like the specialized interface, I would not buy a lighting system that used a basic PC, with mouse and keyboard.

As regards my second draft of the 2-Phase Commit work, I announced it on

31 January, in entry number 150 of this forum. A summary of the changes in the second draft can be found in entry 150. The second draft text can be found in:

>uud>USITT>common>standards>MSC_2PC_0.02

I have received no comments on the second draft yet. However, I have privately conversed with Matt Dni bout the work. I have promised Matt an attempt to work GO_OFF into a third draft. My major difficulty with that promise is a desire to add GO_OFF in a form that differs from what has been done in the basic MSC work.

Regards...

---[0180]--- (nref = [0181])

[0181] (6 lines) CRichmond.USITT 02/11/91 0129.6 mst Mon midi

Subject: Re: MSC 2PC and other things

Ralph, you're correct - I apologize for forgetting your announcement of this version. I was monitoring the forum from out of town and had decided to download it when I got back, then forgot about it. Thank you for your comments on the other matters.

- Charlie

---[0181]--- (pref = [0180], nref = [0182])

[0182] (8 lines) ABall.USITT 02/11/91 0904.4 mst Mon midi

Subject: Re: MSC 2PC and other things

And many thanks to you Brad for expanding on Charlie's information. You have "armed" us all with the information necessary to avoid making poorly engineered hardware choices without having to possess an electrical engineering degree.

Indeed, on to other topics.

Andrew

---[0182]--- (pref = [0181], nref = [0183])

[0183] (8 lines) Weber.USITT 02/13/91 0417.8 mst Wed midi

Subject: Re: MSC 2PC and other things

Charlie,

Don't worry about it. I spent six months last year traveling around for Digital. I could barely keep track of what month it was.

See you in Bean Town.

Ralph:

---[0183]--- (pref = [0182])

[0184] (23 lines) AEkvall.USITT 02/16/91 0442.4 mst Sat midi

Subject: SET chasing...

Hi,

one subject that has not been discussed is how to manage SET command chasing. In many sequencers there is a function called controller chasing. This will run though the recoded material each time you make a jump to find out where the different controller should be. It will scan from the beginning to the cureent point and accumulate the latest value

for each controller. This value will then be sent out to update the controllers.

This should be available for SET controllers too, if possible. It means that a unit using controllers (or in our case SET commands) for moving the faders of a lightboard will always be updated correctly.

The problem with SET is that only some of the SET numbers are specified as controller equivalents, the rest can be push buttons etc. This means that we should have some way to find out if a SET controller should be chased or not. Maybe we should specify that only SET controller above 8192 should be the ones that are allowed to be chased.

Suggestions? I think that Matt could be interested in this question.

Anders

---[0184]--- (nref = [0185])

[0185] (12 lines) CRichmond.USITT 02/28/91 0120.6 mst Thu midi
Subject: Re: SET chasing...

If I understand you correctly, my response to this would be that this function would (should) be a function of the software that is being used as the controller - and not necessarily part of the spec. In other words, this as you have described makes perfect sense, but it is a function which would be done as an option within the sequencer program and is under operator control or choice.

Why would you not, therefore, want to be able to have the SET controls below 8192 be chaseable? Seems to me this would be useful for submasters and other types of controls in that range...

-jes wonderin'

---[0185]--- (pref = [0184])

[0186] (31 lines) CRichmond.USITT 02/28/91 0125.2 mst Thu midi
Subject: 2 Phase Committ

There has been some discussion on the PAN network/MMA Forum about the idea of defining a function for the two currently unused pins on the DIN socket used for MIDI IN, OUT and THRU. There has been a lot of discussion but definitely no consensus so far. That has begun to get me thinking so I came up with a preliminary proposal which I formulated and uploaded to that forum this evening. It also affects our work with 2PC so the following is a copy of the transaction:

-INTRODUCTION

This is a most interesting time for this discussion to appear

(reappear?) We (the MIDI Show Control Working Group - you

thought you may have heard the last of us? Not likely...) are

currently working on the next version of MSC which includes a

very thorough implementation of Two-Phase Commit protocol including full handshaking, parity checking and checksum error correction (all of which remains completely compatible with the basic MIDI protocol since it is optional and self-invoking with automatic time-out resets, et
---[0186]--- (nref = [0188])

[0188] (83 lines) CRichmond.USITT 02/28/91 0130.0 mst Thu midi
Subject: Re: 2 Phase Committ
(got cut off, so here goes again:)

-INTRODUCTION

This is a most interesting time for this discussion to appear (reappear?) We (the MIDI Show Control Working Group - you thought you may have heard the last of us? Not likely...) are currently working on the next version of MSC which includes a very thorough implementation of Two-Phase Commit protocol including full handshaking, parity checking and checksum error correction (all of which remains completely compatible with the basic MIDI protocol since it is optional and self-invoking with automatic time-out resets, etc.) Anyway, this all requires operation within a closed-loop system for proper response and, since these systems will include a number of controllers/controlled devices, the closed loop will have to assume the topology of a full network.

Using regular MIDI connections (IN, OUT, THRU) would mean that the user would have to incorporate a fair number of MIDI mergers and know their network configuration in order to hook up such a network. If we were to use the two additional pins on the MIDI IN and MIDI OUT connectors to provide a complementary output and input, respectively, we could easily construct a full duplex network by simply connecting MIDI IN's to MIDI OUT's in the most direct and convenient manner.

-A PROPOSAL:

MIDI IN of any given device designed to network would receive messages in the normal manner on the normal pins and a built-in merger would properly merge all incoming messages with any messages created by the device and output them via MIDI OUT in the normal way and on the normal pins. In this respect, MIDI IN, MIDI OUT and MIDI THRU are completely compatible with the MIDI 1.0 spec since messages are received in the normal manner and passed to the MIDI THRU jack unchanged, plus any messages created by the device are output via MIDI OUT.

Also, the two extra pins on the MIDI OUT jack would form another standard MIDI input connection to receive messages coming back from the device this jack is connected to - this might be called the "MIDI RETURN" circuit. The messages received here would be merged together properly with any additional messages the host device wants to add in the return direction and would be output

via the two extra pins on the MIDI IN jack. In this manner, a system comprised of properly designed devices can automatically and instantly network regardless of the order in which they are interconnected and without any expertise required on the part of the installer.

-BENEFITS

This would certainly meet the needs of the next version of MIDI Show Control and would make it much easier for us to define its operation. I also suspect this would make MIDI much more attractive to the audio equipment manufacturers who are looking for a better alternative to PA-422, the maverick pseudo-standard which Bob Moses of Rane Corp. (among others) is rebelling against. Since this proposal is functionally similar to PA-422 but with many superior characteristics, including the fact that it is MIDI compatible, means that existing "PA-422" equipment can be connected to the "MIDI NETWORK" with a simple converter box.

Also, it seems to me that this proposal might even make current musical, studio and concert MIDI interconnection schemes much easier to implement. Think about how much faster it will be to set up and tear down if all you have to do is just connect OUT to IN in the most direct and physically convenient way and all the gear will automatically be talking with itself. Troubleshooting may even be easier. -CONS- (it might also be more difficult, but that depends).

-WRAPUP

Anyway - please think about this. It would help us and it would make MIDI a much more respectable, complete and sophisticated system in this era where networking has become so important. (end of pitch: who's next?)

- Thank you,
Charlie

---[0188]--- (pref = [0186], nref = [0189])

[0189] (14 lines) Weber.USITT 03/11/91 0515.9 mst Mon midi
Subject: Re: 2 Phase Committ
Charlie,

I assume that the body of 188 is a proposal that you are sending to the MMA. I cannot imagine how our little band here on callboard could unilaterally make such a change to MIDI.

If my assumption is correct, then my response is that the document appears to be well thoughtout. It seems to contain the strongest arguments that I can think of regarding why MMA should adopt the proposal.

I hope MMA approves.

Ralph:

---[0189]--- (pref = [0188], nref = [0190])

[0190] (57 lines) CRichmond.USITT 03/12/91 0124.0 mst Tue midi

Subject: Re: 2 Phase Committ

Well, Ralph, this particular proposal seems to be getting shot down in flames. In a very definite way, I really can only agree. I don't have the text here but the MMA does not feel that MIDI should adopt a networking approach that is based on such a poor model as PA-422. I could only agree. I just thought in a rash moment that if they didn't know what to do with the extra 2 pins they could at the very least use them for duplex communication. They don't seem to feel at the moment that this is the best way to use them..

On a slightly different subject they have now invited interested members to get involved with a group to discuss porting PA-422 over to MIDI in some fashion in order to establish "compatibility" of some sort. The following is a copy of my response to this request for input:

"

OK, I'm interested in the PA-422 thread. This is also evidenced by the fact that I wrote a previous proposal suggesting a use for the extra connector pins that might be compatible with PA-422. That suggestion was responded to in an appropriate manner by Stanley J.

My own feelings about PA-422 alternate between total revulsion at the current proposal (we have so far refused to produce any equipment which specifically embraces it, although we have a "bridge box" which allows us to convert from MIDI to PA-422 in a rather cumbersome manner) and a grudging acknowledgment that there is a substantial market for equipment compatible with it. Just like RS-232 itself is a pretty stupid specification with all of its variability, the mere phrase "serial" has come to mean a specific subset of the RS-232 spec.

One thing is certain - PA-422 has nothing to do with Show Control and should not be confused with it in the least. Another thing - PA-422 is basically a hardware spec and has inherently no built-in intelligence and very little in the way of complex communication protocol. All that is really required to produce a "bridge box" between the two worlds is make a unit like ours which defines PA-422 data transmission and handshaking in terms of MIDI messages. It is entirely possible that a committee could do a better job than we have done (plus ours uses our Mfr Sysex number, which just won't do), but it's still a cumbersome job to create the software which talks "PA-422" on a MIDI network, no matter how the conversion is done.

If the "network experts" here start to look too closely at PA-422, they will probably come to the same conclusion we have: stay away from it as much as possible, a position that flies in the face of the reality that it is being widely used. What to do? First of all - let's get our own house in order. The fact is that PA-422 is more of a true network than MIDI is right now even though it has worse structural flaws. Let's create a good MIDI network protocol independent of PA-422 (please!) and even independent of the current MIDI structure if necessary (probably).

My vote at the moment goes toward piggybacking any network design onto FDDI, but of course that's still pretty costly (but nice....) Nuf said for now.

- Charlie

"

---[0190]--- (pref = [0189], nref = [0192])

[0191] (23 lines) CRichmond.USITT 03/15/91 0054.4 mst Fri midi
Subject: Now guess what
So this week we got our first request to build a stand-alone MSC box and it was absolutely typical!

What they want it exactly everything that we did not put into the spec yet. First, it is a command_format which was the most obvious and for that reason, probably, the most easily overlooked: cue lights. I suppose this category should go under lighting, but to be honest, I thought about this category briefly and thought that a cue light device should be able to be set or programmed to respond to the command_format that it was communicating on behalf of rather than its own category (ie flies, wagons, etc.) so rationalized the absence of cue lights on the list....

Next, the customer wants it to fully conform with the two-phase commit procedure which we are (Ralph is?) currently working on. Obviously to me we are going to have to hurry along and get this 2PC thing underway or this guy's cue lights/standby/ready switches box is going to be an unfriendly kludge. So...k

I hereby promise to study the 2PC document carefully and get some good EOPp n constructive feedback in on it. Anyone else? Brad, where are you?

- Charlie
---[0191]--- (nref = [0194])

forum:
[0192] (29 lines) Weber.USITT 03/16/91 0503.4 mst Sat midi
Subject: Re: 2 Phase Committ
Charlie,

My first reaction appears to be invalid, after having read the entire message. However, it might still give you some ground to work from. So, I'll toss it in anyway.

My reading of your proposal was not as an attempt to make MIDI like PA-422 (or was it PA-442, I forget). Rather, I saw your proposal as an attempt to give MIDI a chance to compete with the PA-xxx spec on a somewhat level playing field.

What makes my initial reaction invalid are some of the statements near the end of your message. From those statements, it is clear that MMA does not want a level playing field vis-a-vie PA-4x2. MMA wants a playing field that is unequivocally tilted in their favor. The only response to this is, "Go get'em tigers!!!"

This may be seen as plugging one of my company's (Digital) products, but I think Ethernet is a less costly substitute for FDDI. I can appreciate your enthusisam for FDDI. It is the latest and greatest. But, that also makes it the most expensive. For a few years yet, only those folks who cannot afford to use anything technologically inferior to FDDI are going to be willing to pay for it. At the same time, the presence of FDDI at the high-end combined with the fairly large number of Ethernet

manufacturers will keep pushing Ethernet prices down.

Have lots o fun.

Ralph:

---[0192]--- (pref = [0190])

[0194] (22 lines) Weber.USITT 03/16/91 0538.7 mst Sat midi

Subject: Re: Now guess what

I would like to believe that cue lights fit into 2PC in some fashion similar to the electric eye used in the example (in Section 6.6 of the V0.02 draft). Like the electric eye, a cue light does really know cue numbers. It always returns the success and never an abort. When the cue light box gets a STANDBY_BY, it iluminates the light and returns a STANDING_BY. When the cue light box gets a GO_ACKNOWLEDGED, it extinguishes the light and returns a COMPLETE.

Of course, this is not as good as 2PC with a computer. The computer would be expect to know both whether the cue can be executed, for the STANDING_BY, and when the cue is really complete, for the COMPLETE. But, such is the nature of low-tech interfaces.

Happy reading on the 2PC spec. Please try to think about things that can go wrong during 2PC transactions. That is the hardest part to get right, and I will not believe that it is right until several other folks have found the things I forgot.

Thanks.

Ralph:

\032

---[0194]--- (pref = [0191])

[0195] (37 lines) Weber.USITT 03/16/91 2009.5 mst Sat midi

Subject: 2PC V0.03 (Go Off)

Thinking about Charlie's cue light requirements produced the realization that Charlie will soon need the Go Off support in MIDI Two-Phase Commit. So, I've gotten off my duff and added the Go Off description of the MIDI Two-Phase Commit document.

I have asked Tim Clinton to place the new draft in:

>udd>USITT>common>standards>MSC_2PC_0.03

The new draft is approximately 70 Kbytes long (around 25 typewritten pages).

To aid those who have read or are reading V0.02, change bars have been placed in the left-hand column. If you are currently reading V0.02, I recommend completing your reading in that copy and then looking at the changed parts of V0.03. The changes between V0.02 and V0.03 are quite minor.

Support for Go Off has been added in the form of 4 7-bit cue data values that are present in every STANDBY and GO_ACKNOWLEDGED message. Go Off and Go On use 2 of these cue data values to set a Go Level. GO_ACKNOWLEDGED with a Go Level of 255 is equivalent to Go On. GO_ACKNOWLEDGED with a Go Level of 0 is equivalent to Go Off. All of this is described in the newly added Section 6.6.

Charlie is going to need the 4 7-bit cue data values to control the cue lights multiplexer box that he is going to want to build. Such a box is needed to amortize the cost of a MIDI processor over a reasonably large number of cue lights. This too is explained in Section 6.6. To get that cost amortization really high, I stuck switch open/close control in the box description too.

With execution of this promise to Matt and Charlie, I have completed all outstanding work requirements in the MIDI Two-Phase Commit space. So now, you all really have to dig in and find those bugs. Or, V0.03 will be the last MSC_2PC draft.

Ralph:\032
---[0195]--- (nref = [0196])

[0196] (2 lines) CRichmond.USITT 03/16/91 2154.3 mst Sat midi
Subject: Re: 2PC V0.03 (Go Off)
Thank you, Ralph! Now our work is cut out for us - I just hope I'll be able to adequately analyze your thorough document. Onward!
---[0196]--- (pref = [0195])

[0197] (15 lines) Cotten.USITT 03/30/91 1340.7 mst Sat midi
Subject: MSC status ?
Quick Intro: Hi i'm Lary Cotten with Lightwave Research. I've spent the last month or so trying to catch up with MSC activity and approval. Our controllers are just begging for a central control protocol, so after LDI 90 I joined the callboard in hopes of not creating yet another unique one company protocol.

Somehow I seem to be missing one minor little piece of information. What is the current status of approval for MSC. I'd like to write an article for "Show Technology" about what MSC is and where it stands. I'd also like to know because I need to implement MSC or some equivalent (but less complete) protocol of my own in the next couple of months, and I'd much rather have the all the people who interface with out stuff follow a standard like MSC.

?? Lary Cotten.
---[0197]--- (nref = [0198])

[0198] (38 lines) CRichmond.USITT 03/31/91 0114.8 mst Sun midi
Subject: Re: MSC status ?
Hi Lary, I was the one who spoke with you on the phone about joining - You have come to the right place! The current approval status is that we have submitted the MSC 0.2 proposal to the MMA and it has been voted on and approved by them, with one small change in the way MIDI Time Code is a representation of SMPTE Time Code (something to do with the user bits). They have sent it off to the Japan MIDI Standards Committee to be ratified. The arrangement is that unless the JMSC has any specific objections to the MMA approved proposals, such proposals automatically become part of the MIDI Standard after 60 days. This proposal was sent to the JMSC almost exactly 60 days ago and I don't know whether they have responded to it or not. I expect we will know very soon now what the JMSC response to this is, if any.

The JMSC initially stated when they received the MSC proposal that they thought it was beyond their ability to appraise its merits and would be

giving it to the Japanese version of the USITT (!) to come up with a response. I have not heard if they have even found such an organisation much less what their response is. At any rate, we should be hearing something from the MMA Real Soon Now (RSN), but if we don't you can be sure I'll be rattling some modems. The unofficial word is, of course, that it is almost a certainty that it will be approved and on the street imminently - and even if the JMISC has some specific suggestion, we will be able to deal with it quickly and get an amendment approved. That will delay things a couple of months, though. The MMA still has to assign the Real Time Universal Sysex byte for MSC before we know exactly what we're going to be sending and receiving of course! So standby and watch this space - when it is completely finalized I will upload the new MSC V1.0 Standard to the database here.

In the meantime, Larry, the proposal is here in the USITT<Common>Standards file area and can be perused at will. Also for your edification is the TTwo Phase Commit proposal which we need input on (review the last few messages from Ralph Weber to update yourself). All comments are gratefully accepted right here.....

- Welcome again,

- Charlie

---[0198]--- (pref = [0197], nref = [0199])

[0199] (12 lines) Weber.USITT 04/03/91 0405.2 mst Wed midi

Subject: Re: MSC status ?

Details... details... details...

Charlie typoed the specification for the standards area. It's (at least the one I've been using is):

>udd>USITT>common>standards

Do give the two phase commit spec a through review, if you can.

Thanks.

Ralph::

---[0199]--- (pref = [0198], nref = [0200])

[0200] (6 lines) CRichmond.USITT 04/05/91 0132.7 mst Fri midi

Subject: Re: MSC status ?

Oh well, I knew someone could be counted on to be absolutely accurate... (It wasn't exactly a typo; more likely a "memo" - a memoryographical error) Thanks, Ralph! (see how we help each other out around here? Soo friendly)

- Charlie

---[0200]--- (pref = [0199], nref = [0201])

[0201] (15 lines) Cotten.USITT 04/07/91 2153.3 mst Sun midi

Subject: Re: MSC status ?

Thank's for the Update. I've already read the MSC proposal, but havent had a chance to read the 2 phase commit stuff. What I've read, I like. The spec covers about 10 times what I'll need to make an effective working system, but hey!, that's this week. I don't normally write articles for the magazine ("Show Technology"), but if we're going to

have a name like that, I think we ought to cover MIDI Show Control. Can you Imaging a "Keyboard" issue without the letters "M.I.D.I.", or "LD" without the d-word.

Also, where should I point the general public (actually I'm referring to our readers.) for Information about the spec? here? Actual details are beyond the scope of the article, so somebody's bound to ask.

Later Guys (a generic referen

\cce

to both sexes)

---[0201]--- (pref = [0200], nref = [0202])

[0202] (17 lines) CRichmond.USITT 04/07/91 2354.7 mst Sun midi

Subject: Re: MSC status ?

Right - and now the d-word is soon to be the s-word! (and better for it)

As for where to point the general public for detailed technical information:

the group is called the International MIDI Association and has the same address, phone no., fax number and managers as the MMA. In fact, the IMA is the same operation as the MMA but it serves the general public instead of the manufacturing community. Simply put the IMA's particulars in as a reference (you might want to check with them just to make sure that all info is indeed still the same as the MMA, but that's my understanding).

Later, much later....

- Charlie

---[0202]--- (pref = [0201], nref = [0203])

[0203] (3 lines) Cotten.USITT 04/20/91 1111.6 mst Sat midi

Subject: Re: MSC status ?

Yes, the two still have the same address and phone number, etc. And they each want you to join (and pay) separately based on the stuff they've sent me. Thanks for all the info.

---[0203]--- (pref = [0202])

[0204] (4 lines) Cotten.USITT 04/20/91 1116.4 mst Sat midi

Subject: MSC Compatability

Does anyone know if Any or Many of the major sequencer packages (ie:Master Tracks, Performer, Opcode's sequencer (i forgot the name)..Dr. T's, etc...) will currently record realtime sys-ex in realtime? If not, has anyone discussed this with them?

---[0204]--- (nref = [0205])

[0205] (16 lines) CRichmond.USITT 04/22/91 1154.7 mst Mon midi

Subject: Re: MSC Compatability

My admittedly vague understanding of these "general sequencers" is that they all handle this task in different manners and some less well than others. There has been some discussion on the MMA Forum about the need to update these packages to deal with real time SysEx in a better manner than they now do. The acknowledgment is that "real time" does indeed mean that and that most sequencers treat Sysex in a rather cavalier manner, not in a "real time" way. It seems as though it should be

fairly simple to make these improvements for most packages. Of course, we are not so interested in getting others to make these improvements because we prefer the way our show control software deals with it all....

BTW- I'm away in Orlando for a week but will be checking in if I have the time. Otherwise, meet you all back here next week!

- Charlie
---[0205]--- (pref = [0204])

[0206] (39 lines) CRichmond.USITT 05/02/91 2359.9 mst Thu midi
Subject: Approval!

I am pleased to announce that MIDI Show Control has been ratified by the Japan MIDI Standards Committee! This makes this standard development process by far the quickest one I have ever been involved with! Thanks to everyone and let's get on with the next version.....

For everyone's reference: the newly assigned sub-id 1 (to be used where "msc" is placed in the current document) is 02 hex. In other words, the hex string for a lighting "GO" command for "all-call" devices would be:
F0 7F 7F 02 01 01 F7

Also, a couple of editing errors have been discovered in the 0.2 version of the proposal which probably should be corrected before interpreting it as the definitive 1.0 standard. They are:

1. in the third full paragraph of section 2.2, the first sentence should read: If two separate controlled devices responding to the same command_ format etc. (rather than controllers, and pardon the typo here)
2. the entire section of 3.2.2 up to "DROP FRAME NOTES" should be deleted.

This is because we have not included any commands up till now which allow the use of Standard Short Time Code or Standard User Bits so this section is therefore only confusing (besides, the User Bits format is now different)

I am also pleased to say that the first use of MSC will be at the Stardust Hotel in Vegas where we will be controlling a Pan-Command console with the COMMAND/CUE sound and show control system. Watch for this marvellous event to change the face of top-quality theatre momentarily.....

But seriously, folks, this is very exciting and it will start making things much better and easier from now on as far as we're concerned.

As the MMA said: "Uncork the Champagne!"

- Charlie
---[0206]--- (nref = [0207])

[0207] (3 lines) Weber.USITT 05/03/91 0451.6 mst Fri midi
Subject: Re: Approval!
Congratulations!!!

Ralph:

---[0207]--- (pref = [0206], nref = [0208])

[0208] (4 lines) KDHewitt.Thea 05/03/91 0858.2 mst Fri midi

Subject: Re: Approval!

CONGRATULATION TO THE ENTIRE MIDI GROUP!... This deserves some wider publicity, I think. Charlie... would have time to bash together a quick "press release"... I think that USITT and CITT between them could take care of circulation.

---[0208]--- (pref = [0207], nref = [0209])

[0209] (7 lines) JLBracewell.USITT 05/03/91 2021.2 mst Fri midi

Subject: Re: Approval!

Charlie, I'm sure that I can speak for everybody in Sound Design Commission to add a hearty congratulations to everyone involved for this achievement. And you're absolutely right: this should make a LOT of changes. For any skeptics around, I wish you could have heard the discussion this afternoon in my LIGHTING class.

Again, congratulations.

---[0209]--- (pref = [0208], nref = [0210])

[0210] (10 lines) CRichmond.USITT 05/04/91 2303.5 mst Sat midi

Subject: Re: Approval!

Thank you all on behalf of those who have been giving their input and continue to do so! Yes, Ken you are right - this definitely deserves a press release and I will compose one in the next couple of days and upload it here, if that is appropriate. (I will double check the deadline for the last CITT newsletter...)

Thanks again all participants. (Now back to the grindstone to get this all implemented).

- Charlie

---[0210]--- (pref = [0209])

[0211] (13 lines) AEkvall.USITT 05/25/91 0107.1 mst Sat midi

Subject: Logo

Hi

I think that we in the working group should propose a logo for MIDI Show Control.. This logo should be used by any company that follows the standard. I can propose such a logo if the working group wishes so. My wife works as graphical designer so she can help me with this.

I think it is important that we try to standardize how a manufacturer tells a user that he is following this standard. We should state how it should be marked. Which version of the standard etc. It must be very clear for a customer what he is actually buying.

Anders

---[0211]--- (nref = [0212])

[0212] (21 lines) CRichmond.USITT 06/02/91 2231.0 mst Sun midi

Subject: Re: Logo

Yes, I thiwe all agree that this is a good id. I don't know what the MMA's policy is on this but I will try to find out. In the meantime, please feel free to try some ideas on logo design but don't spend too

much time on it till we find out if we're even allowed to use such a thing. A preliminary opinion, though is that we can, as an independent group do anything we want to do in this way since the MMA and the IMA only coordinate the development of standards and do not police the use of them, although they have discussing the matter of testing MIDI compatibility lately. So they will probably not careat we do.

It seems like this would be a good way to both promote the use of MSC plus remind manufacturs and users alike as to the origins of this standard and where to go to get authoritative assistance (the USITT/CITT/OISTAT<?> of course!). Which brings up the other questions - do we want to install a procedure within the institute for dling with this? It only seems as though this may happen if we continue to promote it independent from the MMA.

Ideas?

- Charlie
---[0212]--- (pref = [0211], nref = [0213])

[0213] (4 lines) MHefters.USITT 06/03/91 1919.3 mst Mon midi
Subject: Re: Logo
The Standards policy committee is probably one place to pursue the logo idea, and I'm sure publications will want a hand in it. I'll let the standards people know as we are progressing slowly towards a policy. Once a decision is made, we can then bother Publications. -Mitch
---[0213]--- (pref = [0212], nref = [0214])

[0214] (6 lines) CRichmond.USITT 06/04/91 0833.5 mst Tue midi
Subject: Re: Logo
Great - I knew we could get some of the bureaucracy involved! Ain't delegation wunnerful.

Thanks, Mitch.

- Charlie
---[0214]--- (pref = [0213])

[0215] (11 lines) Cotten.USITT 07/08/91 2319.1 mst Mon midi
Subject: Midi Stardust
Sorry, I haven't been on in two months, but Oh well. I just spent the last few days at the Stardust in Vegas. The show is still being programmed, so there's no MIDI Show Control of lighting just yet, But I did get the Intellabeams runningoff Charlie's Software using an RS-232 protocol I made up while my head was too burried in my computer to check this board to know that MSC had been approved. I will be changing the protocol to MSC as soon as the New rev of our controllerscomes out with the ubiquitous 5 pin DIN jacks. So Be watching the Lido Showroom for "Enter the Night" with Lightwave Research Intellabeams, Morpheus PC-Beams, and an ETC Expression (maybe two) all taking orders from Command-Cue+ hopefully soon to be all using MSC.
---[0215]--- (nref = [0216])

[0216] (4 lines) CRichmond.USITT 07/09/91 0020.9 mst Tue midi
Subject: Re: Midi Stardust
Great Lary! Keep us posted how it goes and let us know if you run into any great discoveries, revelations or difficulties for that matter.

Regards, Charlie
---[0216]--- (pref = [0215])

[0217] (77 lines) CRichmond.USITT 07/12/91 0044.2 mst Fri midi
Subject: 2PC comments...

OK, Ralph, I finally had some time to start looking over your proposal (decided it was more important than driving to Cabo San Lucas for the eclipse - not a major decision) but have only been able to peruse the commands so far. Anyway, a few ideas have surfaced and I want to pass them on in small doses so we can start considering this lengthy document without further ado.

1. Yes, I think we should add a fourth category of minimum sets of which all of the 2PC commands should be a part. Not sure which of the others should be, but we'll have to look into that.
2. 20 STANDBY - add "Normally sent by a controller" to the beginning of the description of this command since it helps clarify which device is talking right from the start. Typo: "or an ABORT"
3. 21 STANDING_BY - add "Normally sent by a controlled device."
4. GO_ACKNOWLEDGED - I have some difficulty with this name and would like to suggest EXECUTE for several reasons, not the least of which is that this command's actions are repeatedly referred to in subsequent descriptive writing as 'execution' 'executing' or some other derivative of execute. See if you don't agree. Anyway, the basic problem I have with this is that it sounds like a message which a controlled device would send in response to a controller's "GO" or some such thing. Perhaps you would prefer something like GO_CONFIRM GO_SAFE or GO_CHECK but upon some consideration I still like EXECUTE - sounds pretty ultimate, doesn't it?
5. STANDING_BY - add "Normally sent by a controlled device."
6. EXECUTE or GO_ACKNOWLEDGED or ? - add "Normally sent by a controller"

Also, suggest rewording last sentence of descriptive text before data detail: "If neither of these is received by the controller in this period of time, the controller proceeds as if an ABORT message was received."

7. 23 COMPLETE - add "Normally sent by a controlled device."
8. 24 CANCEL - add "Normally sent by a controller." Suggest rewording last sentence of paragraph 4: "Manufacturers should document the actions taken when a CANCEL message is received by their controlled EOPp n devices." Suggest rewording second sentence in paragraph 7 (provided my interpretation is correct!): "The only valid reason for an ABORT message is "checksum error.""
9. 25 CANCELLED - add "Normally sent by a controlled device." Suggest rewording last sentence in paragraph before data detail: "No optional parameters are permitted at this time."
10. 26 ABORT - add "Normally sent by a controlled device." Suggest

rewording of last sentence in first paragraph: "Thus, correcting the error condition reported by the status code may not be sufficient to permit execution." First sentence of second para: "Status code severity is related to the ease with which the condition can be corrected." Third para: "The "manual override in progress" status code indicates that the local operator at the controlled device has taken over control of cue execution. Devices may be designed to ignore all two-phase commit MIDI show control messages whenever the local console operator is manually initiating cue actions. See Section 6.2.4 for additional information on this design feature." Fifth para: "The sequence number found in the message that initiated the transaction must be used to identify the STANDBY, GO_ACKNOWLEDGED, or CANCEL message being ABORTed."

11. Also suggest rewording para 4 of CANCELLED to be same as suggested wording of para 3 of ABORT, above. Finally, what do you think of deleting the word "console" from both of these paragraphs? I'm not sure it is necessary or helps particularly.

Anyway, there you have it for now. Overall this looks excellent so far and I'm looking forward to some more time tomorrow to continue reading SECTION 6! This is slow going for me because I want to make sure I understand everything you are saying and am sure it is going to be exactly what we need! Looks good... Thanks again Ralph,

- Charlie
---[0217]--- (nref = [0218])

forum:
[0218] (48 lines) Weber.USITT 07/14/91 0727.5 mst Sun midi
Subject: Re: 2PC comments...
Charlie,

Thanks for the comments. I have written your suggestions onto a paper copy of the spec that I have here. Generally, I agree with your thoughts. I will note exceptions to that below. Since you believe that more are coming, I will delay editing the docuemnt until you are done.

+++

RE: GO_ACKNOWLEDGED...

I can understand your concern about the mistaken implication that GO_ACKNOWLEDGED is a controlled device to controller command. On the other hand, I would prefer to keep the word GO in the command name. I want the commands names to follow the existing verbal cue calling protocol as much as possible. (If GO was not already taken in the basic MSC command set, I would have used that.) Of the options you propose, I like GO_CONFIRM best. But, let me propose some additional alternatives: GO&ACKNOWLDEGE, GO&CONFIRM, GO_&_ACKNOWLEDGE, or GO_&_CONFIRM.

+++

RE: CANCEL --Suggest rewording second sentence in paragraph 7 (provided my interpretation is correct!): "The only valid reason for an ABORT message is "checksum error.""

Your interpretation is correct. However, I believe the current wording is weaker than your comment suggests. What do you think of:

"A CANCEL message must be responded to with a CANCELLED message or an ABORT message. An ABORT message response is allowed only when a "checksum error" is detected in the CANCEL message. ..."

+++

Yep! The phrase "local console operator" in CANCELLED and ABORT should be "console operator."

+++

On the whole, I'd say you've got the hang of two-phase commit. Have fun with Section 6.

Ralph::

\032

---[0218]--- (pref = [0217], nref = [0219])

[0219] (14 lines) CRichmond.USITT 07/15/91 2341.7 mst Mon midi
Subject: Re: 2PC comments...

Well, I'd really like to get some other people's input here before attempting to finalize some version of GO_***** - perhaps this discussion will get things going again.

Yes, your rewording of CANCEL Para 7 is entirely correct and more specific as far as I can tell. Sounds good.

Regarding local console/controller operator - this is used in 4 places and on second thought, I'd like to put forward "controlled device operator" at this time for all 4 places. How about that?

Having fun with section 6 - not quite ready to spiel.

- Thanks, Charlie

---[0219]--- (pref = [0218], nref = [0220])

[0220] (101 lines) CRichmond.USITT 07/16/91 1806.4 mst Tue midi
Subject: Re: 2PC comments...

Well, that was faster than I expected. Nice, very nice - and a very good touch with the elaborate example. OK, here goes:

Section 6., para 2, sent 2: typo "to" to "two"

Section 6.1, para 1, sent 1: suggest "Two types of MIDI show control systems participate....." Sent 3: suggest "This type is called the controller." Para 2, sent 1: suggest "The other type acts upon MIDI show control....." Sent 2: suggest: "This is called a controlled device."

Section 6.2.1, para 1, sent 4: typo "forth" to "fourth"

Last para, sent 1: suggest "STANDBY messages should normally be sent at least 2 seconds....." Sent 2: typo "their" to "there"

Section 6.2.3, chart 1: suggest: "CANCELLED con

CANCELLED controlled device to confirm discarding of
to controller previous instructions

Section 6.2.3, para 7, sent 2: suggest "The simplest possibility is that the human operators at the controlled devices"

Para 11, sent 1: suggest "If a currently executing cue is moving ..."

Sent 2: suggest "If the currently executing cue is moving something..."

Section 6.2.5, para 1, sent 4: typo "in" to "it"

Para 2, sent 2: suggest "...to accept manual override instructions from their local controls."

Section 6.4, para 1, sent 2: suggest "For each STANDBY, GO_****, and CANCEL message sent, a controller constructs...."

Section 6.5, para 3, sent 2: suggest "The exact meaning of these status codes depends on the type of device that sent it." also note my suggestion further on to change the meaning of status code 10 04 for sound.

Para 4, sent 2: suggest "The exact meaning of these status codes depends both on the type and manufacturer of the device that sent it."

Para 7, sent 1: suggest "Manufacturers are free to use status codes ..."

Section 6.5, chart 'Status codes in ABORT messages': timeout's s,g,c should be S G C; suggest separating code 80 50 into separate numbers for each of unknown <Q number>, <Q_list> and <Q_path>.

Para 11, sent 3: is sequentiality a word? maybe reword i.e. "...when its rules on sequential cueing are violated."

Para 13, sent 1: suggest "The table below lists status codes that are..."

Table of command_format dependent codes: suggest making 10 04 for sound the same as for lights, 10 08 amplifier failure, 10 0C amplifier overload. For machinery, add 10 10 servo failure. For video, typo "synch" to "sync" and add 10 08 time code lost. For projection, add 10 08 lamp failure.

Section 6.6, para 2, sent 3: suggest "The examples in this section are simply that: examples."

Example cue: suggest changing the cue number for fly cues to something other than "22" and "22.1" simply because the command_format is also "22" and could possibly be a little confusing.

Notes for "Error-free....", note B, sent 4: suggest "In this case, the flys controlled device returns its two STANDING_BY messages back-to-back, even though other messages separated the STANDBY messages when they were sent." Note D, sent 3: suggest "But this is unnecessary since the controlled device does not require it."

Section 6.6.2, para 1, sent 3: typo "wench" to "winch" also in last para, last sentence.

That's it! Very entertaining reading - needs a wrapup section at the end, or had you planned to add even more by way of examples? Anyway, seems quite good and I'd like to come up with a first version of this to upload to the MIDI forum on the PAN network for the MMA people to peruse. There is a new forum on PAN being organized by the AES and MMA: it is under the SIG (Special Interest Group) "Multimedia Networking" and has been created as a result of the attempts by certain manufacturers to use the new semi- standard "PA-422" as a be-all and end-all sound (and multi-media) equipment networking protocol. It is generally acknowledged that PA-422 is hopelessly inadequate for this purpose but it is also felt that MIDI is as well. I'm not at all sure that this proposal will necessarily do anything to dispel some of the concerns that the critics have of MIDI in general but we can try. Incidentally, most of the criticisms of MIDI are from extremely biased sources with serious vested interests to promote and the 'problems' quoted are usually as a result of misuse of the medium. With the new fibre optic multi-drop MIDI network capability now available through the Lone Wolf system, we now have a complete error-free and reasonably fail-safe bidirectional network available for transmission of MIDI Show Control c/w 2PC.

So let's forge ahead! Hope to hear from you soon, Ralph (and others...)

- Charlie

---[0220]--- (pref = [0219], nref = [0221])

[0221] (22 lines) Weber.USITT 07/16/91 1828.2 mst Tue midi

Subject: Re: 2PC comments...

Well, I'm going to have to read that 101 lines of comments (probably on section 6). But, I do have some thoughts on "controlled device operator."

English is an incredibly flexible language. But, some of that flexibility comes at the cost of ambiguity. Such ambiguity is beginning to creep in here. With "controlled device" meaning an entirely automated system (devoid of a defined human component), I fear that "controlled device operator" might fail to be comprehended as the human operating the "controlled device." My sense of the phrase, "controlled device operator" is that we could use it in the specification, but only if we took the effort to define it prior to the first usage. Making such a definition might force us to overly describe what is a "controlled device>'

I think we should try to find a self defining phrase that means, "the human being responsible for local operation of the system containing a controlled device."

Thanks.

Ralph::

---[0221]--- (pref = [0220], nref = [0222])

[0222] (89 lines) Weber.USITT 07/16/91 2025.7 mst Tue midi

Subject: Re: 2PC comments...

... Charlie,

... I'm going to use a practice that is very familiar to me, but

... perhaps unfamiliar to you to work through your comments. Your
... text will appear on lines that do not begin with "...". My
... responses will appear on lines that begin with "..."

...
... Only those comments with which I have problem will appear here.
... All other comments have been incorporated into the specification.

Nice, very nice - and a very good touch with the elaborate example.

...
... Actually, it was the simplest example I could construct that
... demonstrated all the aspects of 2PC that I thought relevant.

Section 6.2.3, chart 1: suggest:

CANCELLED controlled device to confirm discarding of
to controller previous instructions

...
... Actually, I used:

... CANCELLED controlled device confirms discarding of
... to controller previous instructions

Para 2, sent 2: suggest "...to accept manual override instructions
from their local controls."

...
... The only change here is from "local console" to "local controls."
... I don't see why the word "console" is so burdensome in this context.

Section 6.5, chart 'Status codes in ABORT messages': timeout's s,g,c
should be S G C

...
... I used lower case "s g c" here because the timeout status code can
... only be used internally within a controller. This status code should
... never appear in a transmitted ABORT message. I have added a note to
... this affect but left the letters in lower case.

Para 11, sent 3: is sequentiality a word?

...
... Its definitely a computer jargon word. Also, its in my unabridged
... dictionary. I changed the specification anyway. The differences
... in meaning were nugatory.

needs a wrapup section at the end,

...
... What should go into such a wrapup section?

I'd like to come up with a first version of this to upload to the MIDI
forum on the PAN network for the MMA people to peruse.

... I have made all the changes you've suggested in my copy of the
... specification file, except the following:

...
... GO_ACKNOWLEDGED --> GO_*****
... local operator --> controlled device operator
... local console --> local controls (Section 6.2.5)

...
... Do you want to wait for these issues to be resolved before uploading
... to PAN? Or, do you want to upload with the issues still open and
... let them debate there too? It makes very little difference to me.
... But, since I don't have access to PAN, you will have to relay any
... debate results to me using CallBoard.

...
... I should also note that you must have reviewed an old copy of the
... specification. A new Section 6.6 was added to the last draft.
... The basic concept involves adding d1, d2, d3, and d4 as cue data
... values to the STANDBY and GO_ACKNOWLEDGED messages. The new Section
... 6.6 describes usage of these values. They are intended for things
... like cue lights.

...
... The latest version of the specification is described in transaction
... [0195] in this forum. Fortunately for you, I marked the changes
... with change bars. So you will not have to reread lots of stuff.
... Compared to what you've already read, catching up to the latest and
... greatest should be a piece of cake.

...
... Ralph::

\032

---[0222]--- (pref = [0221], nref = [0225])

[0225] (79 lines) CRichmond.USITT 07/19/91 2356.0 mst Fri midi
Subject: Re: 2PC comments...

...You're a good teacher and I'm determined to learn (and fast,
...I hope) so here are my responses to your previous 2
...transactions in order:

I think we should try to find a self defining phrase that means,
...I'm all for this and what you have written seems fine to use
...as a clear definition for 'local operator.' I agree this is
...potentially confusing considering the control topology and
...such a description is necessary as a definition that can be
...introduced along with the definitions of 'controller' and
... 'controlled device.'

Actually it was the simplest example I could construct
...And very nicely constructed using all the aspects of 2PC -
...that's all I meant, I guess

Actually, I used:
...Perfect

I don't see why the word "console" is so burdensome in this
...Perhaps I'm being overly sensitive but this word connotes to
...me a very specific piece of hardware that is used less and
...less these days as a control input. Often, one encounters
...such variations as trackballs, touchscreens, capacitive or
...pressure sensitive membranes, mice, or other types of devices

...which translate human response into electrical signals
...without looking like the traditional 'console'. That's
...really the only problem with that word here for me.

I used lower case "s g c" here because the timeout status code
...Perfect again

The differences in meaning were nugatory.
...I won't argue with your dictionary again! Actually, my
...comment really was a question - I was at the office and my
...unabridged is at home, so I just asked for the record.
...Please feel free to return to the original wording! (I'm not
...up on the latest computer jargon, either, I'm sure)

What should go into such a wrapup section?
...I really don't know - probably your new version (which I have
...not yet read) will work fine.

GO_ACKNOWLEDGED --> GO_*****
...After perusing our previous discussion on this I propose we
...use GO_CONFIRM for now since you have indicated you prefer
...this and I definitely have a problem with ACKNOWLEDGED.

local operator --> controlled device operator
...As above, keep local operator but add your definition in the
...introduction.

local console --> local controls
...Again per the above discussion, I advocate making this
...replacement.

Do you want to wait for these issues to be resolved before
...I think you and I are so close at this point that we should
...resolve our own differences here and then I will upload. If
...we get input from someone else, we should try to resolve that
...as well before uploading, but the way things are going it
...doesn't look like it's going to be long before we get total
...agreement between all active parties in this here forum. My
...experience is that there will be very little discussion of
...this on PAN since it is pretty much out of their areas of
...expertise and familiarity. There is a likelihood that it will
...generate more discussion as a possible MIDI-based complement
...to the PA-422 protocol, but that is not the basic reason for
...its uploading at this time. I suspect the PAN people will
...fundamentally accept the proposal as we present it and then
...debate its merits as a good use for this other application.

...Sorry I overlooked your updated spec. Inexcusable, really
...since I'm supposed to be hosting this forum.... Oh, well
...nobody's perfect as they say.

...Thanks Ralph - look forward to hearing from you again soon!

- Charlie
---[0225]--- (pref = [0222], nref = [0226])

[0226] (84 lines) Weber.USITT 07/20/91 1234.6 mst Sat midi
Subject: Re: 2PC comments...

>>> Well Charlie, you've set me a noticable chunk of work.
>>>
>>> I have a commitment to a production of West Side Story
>>> tonight (and a put in for Pirates of Penzance tomorrow).
>>> So, I'm not sure whether the work will be completed before
>>> next week.

I think we should try to find a self defining phrase that means,
...I'm all for this and what you have written seems fine to use
...as a clear definition for 'local operator.' I agree this is
...potentially confusing considering the control topology and
...such a description is necessary as a definition that can be
...introduced along with the definitions of 'controller' and
... 'controlled device.'

>>>
>>> Sounds good. Will do.

I don't see why the word "console" is so burdensome in this
...Perhaps I'm being overly sensitive but this word connotes to
...me a very specific piece of hardware that is used less and
...less these days as a control input. Often, one encounters
...such variations as trackballs, touchscreens, capacitive or
...pressure sensitive membranes, mice, or other types of devices
...which translate human response into electrical signals
...without looking like the traditional 'console'. That's
...really the only problem with that word here for me.

>>>
>>> Once "local operator" is defined, then this can become
>>> "the local operators device control interface," or some
>>> reasonable equivalent.

The differences in meaning were nugatory.
...I won't argue with your dictionary again!

>>>
>>> Please! Always question anything that doesn't make sense.
>>> Just between us, I had a very hard time with high school
>>> grammer. There is always a possiblity that my wording of
>>> something is not correct (or not the best it can be). Since
>>> I always want the best, I am always open to criticism.
>>>
>>> BTW (by the way): Nugatory happens to be one of the favorite
>>> words used by DEC's storage subsystems master architect.
>>> I wouldn't know it otherwise.

What should go into such a wrapup section?

...I really don't know - probably your new version (which I have
...not yet read) will work fine.

>>>
>>> I cannot think of anything new to put in a wrapup section. If
>>> I did, I'd probably put it in the introductory part of Section 6.
>>> The existing MSC specification seems to be similarly spartan.

GO_ACKNOWLEDGED --> GO_*****

...After perusing our previous discussion on this I propose we
...use GO_CONFIRM for now since you have indicated you prefer
...this and I definitely have a problem with ACKNOWLEDGED.

>>>
>>> Will do.

...Sorry I overlooked your updated spec. Inexcusable, really
...since I'm supposed to be hosting this forum.... Oh, well
...nobody's perfect as they say.

>>>

>>> No problem.

>>> I will revise the spec based on the agreement reached thusfar.

>>> Then, I will upload two copies to CallBoard. One will have

>>> change bars, suitable for posting in the CallBoard specs

>>> area. The other will be just raw text, suitable for posting

>>> on PAN. I'll ask Tim to put the second copy in Charlie's

>>> private disk area.

>>>

>>> I am also going to post both MSG and 2PC in a DEC forum on

>>> theatre. Maybe some computer hacker there will find some

>>> detailed errors in 2PC that I've over looked.

>>>

>>> As noted above, other commitments may slow the execution time

>>> for this work.

>>>

>>> Thanks.

>>>

>>> Ralph::

\032

---[0226]--- (pref = [0225])

[0227] (12 lines) Weber.USITT 07/20/91 1415.4 mst Sat midi

Subject: GO_*****

Charlie<

I was thinking about GO_CONFIRM and several questions occurred to me.

Did you reject GO&CONFIRM (which is a little bit more indicative of the function required)?

Are either of the following any better: 2PC_GO or GO_2PC?

Thanks.

Ralph::

---[0227]--- (nref = [0228])

[0228] (59 lines) CRichmond.USITT 07/21/91 2306.6 mst Sun midi

Subject: Re: GO_*****

Hmmmmmm..... offhand, GO_2PC seems to have a nice condensed ring to it.

I have been trying to think about commands that are easy to say as well as type plus have a sort of ring to them. No decision yet for me....

In the meantime (ha, ha) I have looked over v0.03 and this has now brought up some more ideas and questions.

Question 1: Could you please elaborate a bit on how you propose to transmit a "value" of '0' and '255' using d1-d4 to provide the equivalent "GO" and "GO_OFF". I'm not sure this technique of letting users provide their own meanings to d1-d4 values is a good one (or perhaps you intend to define all d1-d4 values eventually and give them meanings). At any rate, to have d values give "GO_CONFIRM" a meaning

that says it is actually not "GO" but "GO_OFF" seems possibly a little obscure. There are a lot of basic commands of which GO_OFF is but one which I could see users wanting to obtain a 2PC version of using d1-d4. I do understand the value of these d bytes and why you put them in but I don't think it is necessarily a good idea to use them to change "GO_***" into a somewhat different command.

I am now wondering whether or not it might be a better approach to provide a new set of commands starting 2* hex which are in essence a superset of the basic ones (and possibly even the Sound commands as well - I'll have to think about that, but probably not for now.) These commands would be:

- 21 GO_2PC
- 22 STOP_2PC
- 23 RESUME_2PC
- 24 TIMED_GO_2PC
- 25 LOAD_2PC
- etc....

Well, you see what I mean. Obviously this will require some rather serious looking in to and perhaps it is not logical to give all of these a 2PC version. But we definitely have identified GO_OFF as one and I think it could be easily argued that STOP, RESUME, TIMED_GO, LOAD, and FIRE might be necessary. None of these except GO_OFF, TIMED_GO, and RESUME would really receive exactly the same operational treatment between controller and controlled_device as GO_2PC does now (even then, there are slight differences with these).

Anyway, this is all starting to threaten to be the rat's nest I hoped it would not become - but after all we are attempting to create a fool proof standard, so fools need not apply. Seriously, though, I think the basic question we need to answer now is are we going to allow d1-d4 to be able to so seriously change the nature of the command that it actually could mean a different command and if so, how far will this go? I think we must at least keep the process of 'standardization' firmly enough in hand that no values of d1-d4 can be used that are not clearly defined.

If we do decide to go to the superset of basic commands moved up to the 2* hex range, should we still allow the d1-d4 bytes?

Your thoughts? (in between production duties, of course - we won't hold our breath!)

Thanks, Ralph - hope you read this before all the uploading...

- Regards, - Charlie
---[0228]--- (pref = [0227], nref = [0229])

[0229] (160 lines) Weber.USITT 07/22/91 1910.0 mst Mon midi
Subject: Re: GO_*****
Charlie,

I didn't upload the spec. I was hoping you'd like either GO_2PC or 2PC_GO. That would mean that you and I have reached closure on all the major issues. That hope worked out pretty good, I'd say.

Of course, now there's the d1-d4 matter. But, I hope that will be resolved

after another transaction (or two).

The next version of the spec will use GO_2PC. (That might even reduce the page count by one or two pages. :-) [The last symbol :-) is a sideways smile, use to indicate a humorous remark. See the two eyes, nose, and smiling mouth.]

The remainder of this transaction will discuss d1-d4. Please do not take any of the following as an indication that I am offended by the questions or proposals. The folks that taught me to write credible English taught me to write emphatically. Sometimes this ends up reading stronger than I intend.

=====

I think that there are more than a few problems with STOP_2PC, LOAD_2PC, etc. First, I think STANDBY_STOP_2PC, STANDBY_LOAD_2PC, etc. are required in addition to the basic do-it equivalent messages. STANDING_BY_STOP_2PC might also be required. Second, I cannot imagine how one could get acceptable response out of a system where the STANDBY_STOP_2PC message must proceed the STOP_2PC message by at least 2 seconds.

I think the original MSC message set is based on a very different operational concept than 2PC. MSC is dynamic, do this now, this way. 2PC is planned, scripted, do this sequence just like it was done yesterday.

MSC will be at home at rock concerts, where there is no script. 2PC will be at home at Miss Saigon, where the goal is giving the audience the same crisp performance every night.

Trying to combine them in such a direct way will only produce a monumental mess. Please let me know if you don't agree.

=====

Now that I've said 2PC is not for rock and roll, you might ask, "Why add support for GO_OFF (a rock and roll console concept) to 2PC?" I did it because the same support also provided for cue lights (which even Miss Saigon might find handy). When a methodology serves multiple needs, then I start believing that it might be useful in a general purpose specification.

Personally I think that, if Matt Deakin ever gets a chance to read the 2PC spec, he'll say, "Standby, wait 2 seconds, then go ... you've got to be kidding me!" Then, he'll whip out his MSC spec and start building his consoles accordingly.

Still, I promised Matt that I would work GO_OFF into 2PC. As I will show later, that's what I did. But, the really gave me a reason I added d1-d4 was your (Charlie's) cue lights. If those cue lights had never been mentioned, d1-d4 might still be on my to do list. (Sorry Matt.)

=====

Regarding the purpose of d1-d4, I want to start by quoting the specification:

"The d1, d2, d3, and d4 data values allow controlled devices

to rely on the controller to remember a small amount of information about how a cue is to be executed."

Think the controller and controlled device as two elements in a larger system. The purpose of the overall system is to operate the mechanical (and electrical) equipment associated with a theatrical production. Within the overall system, the controller and controlled device are partners.

The V0.2 specification required the controlled device to maintain ALL information about how cues will be executed. The d1-d4 values allow the controller maintain up to 28 bits of such information as a service to the controlled device. In effect, d1-d4 transfer 28 bits of the work associated with remembering how to do a cue from the controlled device to the controller. The relative memory workloads in the production partnership have changed, nothing more.

I also disagree with the idea that d1-d4 somehow allow users to provide cue data that they should not (or would not otherwise) be allowed to provide. When viewed from the perspective of the overall system, d1-d4 may move the location where the user's cue data is stored. But, one way or another that data must be stored somewhere in the system in order to operate the show correctly.

In cases where a cue requires lots of memory, d1-d4 will be nearly useless. The most they can do is supply one very small piece for a much larger puzzle. However, in cases where each cue requires very little memory d1-d4 might be very important.

=====

I think that d1-d4 make building a Cue Light Control Box (CLCB) practical. Without them, the CLCB would have to contain enough memory to associate every <Q_number> it receives with what cue light is to be flashed. In addition, the CLCB would have to have a programming interface through which all this data could be entered. Using d1-d4, the CLCB needs only to inspect the "which cue light" data maintained and sent by the controller. Since the controller already must have a pretty sophisticated programming interface, the added cost of loading and storing the CLCB data is very small.

=====

As for GO_OFF: What Matt really wanted was a GO[to a level] message. On his consoles, cues are somewhat like what some other consoles call submasters. Each cue has a 0-100 slider associated with it. Making a cue GO is accomplished by moving the slider to a higher value. Since Matt's envisioned usage of MSC is connecting some number of sub-consoles to a master console, what he wants is for movement of a single slider on the master console to affect similar movement of the equivalent sliders on all the sub-consoles. Thus, ideally, moving a master slider to 50 would send GO[to 50%] messages for the appropriate cues to all the sub-consoles.

Matt was willing to settle for GO[to 100%] (a.k.a. GO) and GO[to 0%] (a.k.a. GO_OFF) in MSC. But, he was very clear about his true desires in a private discussion at LDI. So, despite the fact that I think the 2 second rule in 2PC will make him sick, I've given Matt the full 0% to 100% range in the 2PC GO_OFF support methodology.

=====

Regarding the notion that d1-d4 can change a GO_2PC message into some other message, GO_OFF_2PC (or whatever):

This is true only if one thinks of GO_2PC in the same way that one thinks of GO. In MSC, GO is one of a collection of action messages (messages that effect some kind of mechanical or electrical action in the equipment operated by a controlled device). Some other action messages are: STOP, TIMED_GO, LOAD, SET, and FIRE.

Among the 2PC messages, only GO_2PC (and in error cases CANCEL) actually effect some action in the mechanical or electrical equipment operated by a controlled device. All of the other messages ONLY change the memory state in either the controlled device, the controller, or both. N.B. I consider this to be a fundamental feature of the 2PC proposal.

So, I contend that GO_2PC is the one and only non-error-case do-it message in the 2PC message set. When viewed in this way, d1-d4 DO NOT change the do-it meaning of GO_2PC. They only change the specifics of what is to be done.

Again, d1-d4 functionally move 28 bits of per cue memory requirement from the controlled device to the controller. They do not affect the controller - controlled device partnership in any other way.

=====

Now... I feel somehow too close to the problem to know what (if any) of these thoughts need to be incorporated into the 2PC specification. I am also unsure about how or where to add them. So, I will await comments on these matters. I'll also accept pointed barbs about how these concepts are all wet.

Ralph::

\032

---[0229]--- (pref = [0228], nref = [0230])

[0230] (33 lines) CRichmond.USITT 07/23/91 0854.1 mst Tue midi
Subject: Re: GO_*****

You are a very convincing presenter - I was primarily fishing for more of the 'inside' rationale behind some of the decisions that were made. (plus acting as a bit of a devil's advocate in order that we could sit for a moment and question the long-range wisdom of this direction.)

You have responded to my concerns quite adequately and the explanation of the origin of d1-d4 was very enlightening. As much as I understand why, I have to say I'm still a little apprehensive about providing 4 bytes of data (28 bits) to the 'user' to do with essentially what he pleases. I know this is not the intent but in fact it is very much the way it is simply because these data bits are not firmly defined by the spec as to what they do and what the various ranges of data will be used as. This is a relatively minor problem and I will look over the spec again to see if in fact it isn't more closely proscribed.

Finally, if the idea was to provide Matt with a range of 0-100 level data, should we not define how this data is to be obtained with the

d1-d4 data? Also, could you please answer my final question - exactly how are the 'values' 0 and 255 sent to mean "GO" and "GO_OFF" (so to speak). These two questions are really one in the same and their explanation should indeed be part of the spec by way of illustration at least. I also agree that you should include most of your above explanation to me (us?) as part of the introductory/d1-d4 spec as appropriate: it is very helpful in understanding the 2PC rationale plus it means your excellent dissertation (forceful, but pleasantly so! :-)) will not go to waste. It would be a pity to have it be resigned to the bit bucket for eternity. Please do engage in colorful expository explanation for the benefit of those brave souls taking the plunge into deep dark spec reading territory....

Thanks, again (yes we are very close)...

- Charlie

---[0230]--- (pref = [0229], nref = [0231])

[0231] (81 lines) Weber.USITT 07/23/91 1816.4 mst Tue midi

Subject: Re: GO_*****

Charlie,

This has been an excellent discussion. Your questions have caused me to notice weaknesses in both the 2PC overview and the cue data value discussion. I will be working to improve both of these (hopefully tonight).

Your continuing concern about 'user' freedom with the d1-d4 values is important. Here is a better response than the one I wrote last night.

(Not that last night's response is incorrect. Just that it ignores exactly who has the freedom to define the usage of the d1-d4 values.)

Also, I will be adding some text like what follows to the Cue Data Values section.

The usage of the d1-d4 values is defined by the designer of each controlled device. This definition must either 1) conform to one of the definitions shown in the MSC-2PC specification or 2) be clearly stated in the documentation that describes the controlled device. Option 1 is preferred, and we will amend the specification as necessary in order to make it practical for all major controlled device manufacturers.

When a controlled device receives a d1, d2, d3, or d4 value that is incorrect, it must respond with an ABORT message containing one of the "invalid dn cue data value" status codes. (Note: I have replaced the single "invalid dn data" status code with four individual status codes.) Thus, an incorrect d1 value must result in an ABORT message containing the "invalid d1 cue data value" status code. In those cases where the error cannot be isolated to a single d1-d4 value, the ABORT message must contain the status code that is appropriate for the lowest numbered data value involved. For example, if a compound datum is constructed from the d3 and d4 values and that compound datum is incorrect, then an ABORT response message containing the "invalid d3 cue data value" status code must be sent.

Regarding d1-d4 and go levels:

Here again, a bad choice of characters in my description of the feature has caused you to believe that the methodology is undefined. In section 6.6, I wrote:

Lighting consoles that operate on the "Go On, Go Off" concept use ll as a "Go Level" (where $ll = d1 + (d2 * 128)$).

I now realize that ll was a bad choice of symbols. It looks too much like the number eleven. I have already changed ll to gl, which hopefully makes the formula by which d1 and d2 are converted to a value ranging from 0 to 255.

The next interesting question is, "Exactly how do numeric values between 0 and 255 become level percentages between 0% and 100%?" The answer to that one varies from one lighting console to the next. The 0 to 255 range is what the a console feeds into its digital to analog converters to get (directly, or indirectly) a 0 to 10 volt signal. But, as Karl G. Ruling shows in his Lighting Dimensions article "Mixed Messages," each lighting console uses a different function to match 0% to 100% to the 0 to 255 D-to-A input range.

ref> Karl G. Ruling, "Mixed Messages," Lighting Dimensions, March 1991, Volume XV, Number 2, pages 106-107.

I'd like to codify the exact relationship between gl and percentage. Rumor has it that Gordon Pearlman is writing a table that gives exact digital values (between 0 and 255) for every whole number percentage between 0% and 100%. I am supposed to get a copy of this table to put in the ASCII Lighting Cues specification (which I'm trying to bring up to snuff). But, I haven't received that yet.

Given the obvious mess described by Karl, I'd rather have someone having Gordon's reputation laying down the law. So, I prefer waiting over acting on this one.

I hope this helps.

Ralph::

\032

---[0231]--- (pref = [0230], nref = [0233])

[0232] (69 lines) Weber.USITT 07/23/91 1938.9 mst Tue midi

Subject: New 6.2 & 6.3

Here is the text for two new sections that appear between the old Section 6.1 and 6.2. What do you think? Ralph::

6.2. HUMAN OPERATORS

This specification assumes that a human operator normally is present at every controller and controlled device. A person serving in this capacity is called the "local operator." The local operator may only monitor operations. Or, the local operator may perform safety verification functions, such as activating a deadman

interlock switch. The local operator also may initiate a manual override of a controlled device as described in Section 6.4.4.

Presumably, the local operator effects his or her control via some type of console or other user interface. This interface is called the "local operator interface."

6.3. RELATING TWO-PHASE COMMIT TO OTHER MIDI SHOW CONTROL MESSAGES

There are several important differences between Two-Phase Commit (2PC) and the other Midi Show Control (MSC) messages. Understanding these differences is important to successfully building a product to use either message set. Also, the differences strongly suggest that any one product should NOT support both message sets. So, understanding the differences is important to choosing the best message set to use in any given product.

The MSC/2PC differences start with fundamental principles. MSC is dynamic: do this now, this way. 2PC is planned, scripted: do this sequence just like it was done yesterday. MSC is well suited to rock and roll concerts, where there is no script. 2PC is better suited to Broadway musicals, where the goal is giving the audience the same crisp performance every night.

The MSC message set includes many messages that cause some kind of mechanical or electrical action by the controlled device, for example: GO, SET, FIRE, STOP, and RESUME. This large collection of action messages is required to communicate all the various desired results, with consistent behavior across a large variety of controlled devices. Appropriate behavior is ensured through explicit conformance to each action message defined in the MSC specification.

The 2PC message set contains just one action producing message, GO_2PC. Precisely what results are produced by any given GO_2PC message is determined by a script of cues stored in the controlled device, and how the controlled device interprets that script of cues. Ensuring the desired behavior is a matter of selecting appropriate controlled devices and loading them with cue scripts that produce the correct results. Then, the controller must be programmed to issue 2PC messages in ways that properly coordinate the cue scripts in all the controlled devices.

The major advantage of MSC over 2PC is faster responses to requested actions and simpler system configuration. The faster response comes from two facts. First, MSC requires just one message to produce an action. 2PC requires at least three messages. Second, 2PC includes an up to 2 second delay between the request for an action and initiation of that action.

2PC has two advantages over MSC. First, 2PC can coordinate the actions of multiple controlled devices with an extremely high degree of certainty. Second, 2PC has error detection and recovery semantics built in to the protocol.

---[0232]---

[0233] (27 lines) CRichmond.USITT 07/24/91 0018.3 mst Wed midi
Subject: Re: GO_*****

All right! This all sounds good to me - not sure whether we should set the 0-100% range into stone (since it probably varies in practice anyway) but I'm not a lighting person and really don't feel qualified to comment knowledgeable. When do you think Gordon will have this info?

I'm also not a cyberpunk so you'll have to forgive me on this one - it must be painfully obvious to others - but I really thought I understood your formula for converting the value generated by d1 and d2, both in terms of what it should be and how you have expressed it; yet it seems to me that you are saying that these two bytes can produce a maximum decimal value of 255 - but I see 14 bits available to be used. What gives and I apologize in advance for being dense in the field of these numerical expressions.

Finally, in the second message - the text is great - there's one typo at the end: "to" instead of "two". I know you're really busy this week so there's no need to blow your brains out doing too many things at once just because I happened to finally get around to dealing with this! :-)
(I like that...) I'm also not sure we should recommend that a single product not support both MSC and 2PC (unless you are referring to controlled devices only, perhaps) because we are planning to make a controller that supports bot simulataneously and this seems quite logical.

Have fun in the sun (it's finally hot here now)!

=-Charlie

---[0233]--- (pref = [0231], nref = [0234])

[0234] (23 lines) Weber.USITT 07/24/91 1622.6 mst Wed midi
Subject: Re: GO_*****
Oh! That 0-255 problem. :-)

I've been assuming all along that d2 will be limited in value to 0 or 1. That's enough to guarantee that g1 is in the range 0 to 255.

I'm in the last stages of editing the spec. I just finished passing it through the grammar checker. Upload should be occuring later tonight, baring unforeseen complications.

For all I know Gordon has his tables done already. Since he's not a USITT member (and I'm not a theatre insider), I don't know how to contact him. I've been waiting for the ANSII Cues person who promised to get the tables to do so. If you have info on how to contact Gordon, I'd be glad to use it.

As for the busy part... That died down after Sunday. The director decided that a follow spot is not necessary in Pirates. So, I don't have a job for this show.

Stand by to warm up your data modem... (The new spec draft is 25 pages, typewritten.)

Ralph:

---[0234]--- (pref = [0233])

[0235] (19 lines) Weber.USITT 07/25/91 0422.5 mst Thu midi
Subject: 2PC V0.5 Available
I have uploaded V0.5 of the MIDI Show Control - Two-Phase Commit specification. This draft incorporates all the ideas discussed in the last several transactions. Please read the transactions for the overview of of the new draft's contents.

I have asked Tim Clinton to place this draft in:

>udd>USITT>common>standards>MSC_2PC_0.5

WARNING: This document is 75 Kbytes in length and occupies 26 pages,
type written.

Most of the changes from the previous draft are marked with change bars.
The exception is the changes from GO_ACKNOWLEDGED to GO_2PC. That change was too pervasive for me to handle with change bars.

Ralph::

\032

---[0235]--- (nref = [0236])

[0236] (10 lines) CRichmond.USITT 07/26/91 0021.5 mst Fri midi
Subject: Re: 2PC V0.5 Available
So.... I only have one question: why should d2 be limited to 0 or 1 only???? Should we not define the conditions of the range of possible values it could take and restrict its use specifically if that is what is required at this stage?

Is this getting boring? Am I only entertaining myself? Tell me when to stop so it won't hurt so much... :-><=| (yes I'm dancing, dancing dancing the night away..!)

- yoo no hoo

---[0236]--- (pref = [0235], nref = [0237])

[0237] (21 lines) Weber.USITT 07/28/91 1324.6 mst Sun midi
Subject: Re: 2PC V0.5 Available
Since the particular usage of d1-d2 in question involves lighting consoles and since lighting consoles already use D-to-A converters bases on digital values ranging from 0 to 255 (in almost a standard way), I think the specified range for gl (go level) is reasonable. Since we cannot use the sign bit in MIDI values, we must use a combination of d1 and d2 to get the 8-bit range (0 to 255). This does leave some unused bits in d2. But, that's life.

The only other option would be to spec gl in level percentages (i.e. with a range of 0 to 100). Since the ASCII cues format folks (perdominately lighting console types) are getting very up-tight about the inaccuracy of 0 to 100, I doubt that they would prefer this solution. BTW the reason 0 to 100 is becoming inadequate concerns lighting consoles that control moving luminaries. 0 to 100 means that every step moves the fixture 3.6 degrees. That's a lot!

Dance till you drop Charlie :-) I love a good debate. I used to attend

these three-day spec debating sessions every six months at DEC. Eight hours (or more) of debating for three days straight. Great fun!!!

Ralph:

---[0237]--- (pref = [0236], nref = [0239])

[0239] (14 lines) CRichmond.USITT 07/29/91 2234.0 mst Mon midi
Subject: Re: 2PC V0.5 Available
Well, I may have met my match for now... Let's just say we'll go with what we've got for the time being - hi ho hi ho it's off to PAN we go

But I still think that we need to reserve the additional bit values available for yet undefined applications and meanings. In a sense we are since we have only specified 0-255 so far but perhaps we should specifically say that higher values are reserved for the future... Does this go without saying? Only if we specifically say 0-255 are the only currently usable values.

Beat me with a ...? Got to keep this up or our quantity discount rates on this forum will be cancelled.

- C. You Later

---[0239]--- (pref = [0237], nref = [0240])

[0240] (4 lines) CRichmond.USITT 07/29/91 2245.4 mst Mon midi
Subject: Re: 2PC V0.5 Available
So I just checked the standards directory and it does not contain V0.5 yet - has Time said he would put it there or am I just too impatient?

- C. You (its later)

---[0240]--- (pref = [0239], nref = [0241])

[0241] (17 lines) Weber.USITT 07/30/91 1819.0 mst Tue midi
Subject: Re: 2PC V0.5 Available
I think adding verbage to the effect that values above 255 are reserved for possible future expansion is reasonable. Also, remember that all the bits in d3 and d4 are available for future enhancements.

I sent Tim mail about copying the spec to the common directory. I also asked Tim to copy a version of the spec without change bars to your private directory (Charlie). I doubt that PAN wants to see our revision history in action.

Lastly, I asked Tim about sending membership info to a colleague here who wants to participate in these spec writing fora.

I have heard nada from Tim on any of this. Either I sent it to the wrong name (I used Clinton, which is where I got mail from). Or, Tim is on vacation. I'm hoping the latter.

Ralph:

---[0241]--- (pref = [0240], nref = [0242])

[0242] (16 lines) CRichmond.USITT 07/31/91 2351.8 mst Wed midi
Subject: Re: 2PC V0.5 Available
Your cue, Tim.

Actually, Ralph, to expedite things it would be easiest if you just

uploaded the spec to me as you want to present it to PAN direct via sdm.
I usually do this in cases like this and all is fine ~TN{?E8pYG9{N*T`\$
04uqR}(ok i.e. article copy going to the TD&T Editors is all sent via
sdm - my protocol settings may be different from yours but I send via an
XMOdem variant called "ASCII" (meaning no non-ascii characters to be
transmitted) parity of 8N1, xON/xOFF handshake, Text mode, CRC on - and
it seems to simulate sitting at the old keyboard pounding away quite
nicely. (That's simulating nicely, not pounding away.... :->-<= ...
still dancing...)

Standing by sans Tim...

- Charlie .,
---[0242]--- (pref = [0241], nref = [0243])

[0243] (3 lines) Weber.USITT 08/01/91 0527.4 mst Thu midi
Subject: Re: 2PC V0.5 Available
Tim is back. I just got mail from him.

Ralph:.
---[0243]--- (pref = [0242])

[0244] (14 lines) Weber.USITT 08/01/91 0530.4 mst Thu midi
Subject: "Reading cues..." Status
Charlie,

I realized this morning that I made a mistake in agreeing to put the
"reading new show cues from disk" status in both lighting and sound. If
both lighting and sound need that status, others may need it as well.
Even if they don't need it now, they might in the future.

Therefore, the "reading new show cues from disk" status code should be
in the command_format independent group of status codes. Baring
objections, I want to make that change.

Thanks.

Ralph:.
---[0244]--- (nref = [0245])

[0245] (2 lines) Clinton.Thea 08/01/91 1231.9 mst Thu midi
Subject: Re: "Reading cues..." Status
Sorry for the delay, gentlemen. The file is now in place and available
to all interested.
---[0245]--- (pref = [0244], nref = [0246])

[0246] (9 lines) CRichmond.USITT 08/02/91 0101.7 mst Fri midi
Subject: Re: "Reading cues..." Status
I have no objections, Ralph. Do you want to make the change and upload
a complete new file or just the correction or can you just outline the
change here. I vote for here because we can then 'all' see it and I can
just make the change in the the V0.5 which I just downloaded (If you
trust me) and then send it off to PAN. (promises, etc.)

So.... here's 2PC in your aye yi yi...

- Charlie
---[0246]--- (pref = [0245], nref = [0247])

[0247] (19 lines) Weber.USITT 08/04/91 1705.7 mst Sun midi
Subject: Re: "Reading cues..." Status
Charlie,

I had thought to treat the "reading cues" status code change and the "256 and up reserved on lighting GO" change as edits to the V0.5 draft. Presumably, there will be more comments received and a V0.6 draft will be necessary sometime in the next couple of months. These changes would appear then.

I don't think either change is substantial enough to mislead reviewers (either here or on PAN). If someone else in either location notes the need for these changes, well then, we've already handled that comment.

Having said all that, I also recognize that you may need to have the "latest and greatest" for PAN. If that is the case, then I'll construct and upload a V0.6 draft.

Thanks.

Ralph:
---[0247]--- (pref = [0246], nref = [0248])

[0248] (8 lines) CRichmond.USITT 08/07/91 0137.1 mst Wed midi
Subject: Re: "Reading cues..." Status
OK, I promise this will be the last time I suggest you change and update the files (unless you come up with another important consideration....) V0.6 will be the one for PAN and I will send it to them as soon as it appears!

- Onward ever onward

(hmm, that's a funny name)
---[0248]--- (pref = [0247])

forum: p b#n

[0249] (20 lines) Weber.USITT 08/08/91 1708.3 mst Thu midi
Subject: 2PC V0.6 Uploaded
I have uploaded the V0.6 draft of the MIDI Show Control - Two Phase Commit specification. This draft incorporates all the ideas discussed in the last several transactions in this forum. Please read the transactions for an overview of the new draft's contents.

I have asked Tim Clinton to place this draft in:

>udd>USITT>common>standards>MSC_2PC_0.6

WARNING: This document is 75 Kbytes in length and occupies 26 pages, type written.

Most of the changes since the V0.3 draft (including both the V0.5 and the V0.6 changes) are marked with change bars. The exception is the changes from GO_ACKNOWLEDGED to GO_2PC. That change was too pervasive for me to handle with change bars.

Ralph:

\032

---[0249]---

[0250] (16 lines) CRichmond.USITT 08/31/91 0106.7 mst Sat midi

Subject: MMA TSBB

The latest Technical Standards Board Bulletin of the MIDI Manufacturers Association includes the item for consideration "2PC MSC V0.6" as submitted by Ralph here in the common standards database. Because of its length they have not printed it in the TSBB but it is in the PAN network database and can be ordered from the MMA in hardcopy. Also, the current release of MSC 1.0 is now available from the MMA for a fee of \$7.00. Also, the MMA has requested that we refer all enquiries for copies of this standard to them rather than us providing our "own" version or contraband photocopies. This is one way for them to build their mailing list, find out who's out there and also bring in a little cash to help out.

More as late-breaking news develops and the republic declare independence...

- Charlie

---[0250]--- (nref = [0251])

[0251] (10 lines) Weber.USITT 09/04/91 1714.1 mst Wed midi

Subject: Re: MMA TSBB

Charlie,

I'm a member of the IMA but not the MMA. Can I order a copy of MSC V1.0 from the IMA? The IMA appears to have a pretty comprehensive specification order-by-mail capability (based on what I see in the newsletter they send me).

Thanks.

Ralph::

---[0251]--- (pref = [0250], nref = [0252])

[0252] (14 lines) CRichmond.USITT 09/04/91 2336.7 mst Wed midi

Subject: Re: MMA TSBB

Yes, absolutely - the IMA and the MMA are practically the same thing; same address, phone, fax, staff etc. The only reason for the distinction is that the MMA is specifically for manufacturers, the IMA is for everyone else (MMA is also more expensive...). All that the MMA provides that the IMA does not is access to the decision making process in the development of the MIDI standard (which we are all witnessing here for our own little area of interest - other discussions take place on the MMA Forum of the PAN - Performing Arts Network).

If you have any difficulty please let us know here! (They can be a bit quirky at times - run by musicians, you see - and if you thought Theatre was bad....)

- C

---[0252]--- (pref = [0251])

[0253] (53 lines) CRichmond.USITT 10/29/91 0019.7 mst Tue midi

Subject: LDI/CyberArts

Do we need to organize a meeting for this group at LDI or CyberArts? (in other words - Ralph, do you want to get together with yourself at either of these events?) Seriously, I will be at both of these (CyberArts -14-17 Nov, Pasadena; LDI in Reno shortly afterward) and would be quite willing to schedule any amount of time required to accomplish anything felt to be necessary at this stage.

A brief update on other matters - the AES "PA-422 Committee" have basically decided that PA-422 is not the right thing for networking audio equipment although there is a move afoot to port PA-422 protocols to MIDI using a Sysex code for the various commands and data simply so it can be merged into a MIDI system. The companies that are active on this committee have recently been looking carefully at 3 different low-cost network alternatives: Echelon's LONworks Local Operating Network using Motorola's and Toshiba's "Neuron Chips" built under license (this is a very interesting system built mostly on huge amounts of hype and PR and desintined to fill a large niche in the industry but probably not right for our 'real-time' needs); Intelix' MIND Net network which uses good networking principles, has lots of useful development tools for the PC available but uses only a non-standard implementation of the relatively slow RS-485 protocol; and Lone Wolf's Media Link network.

This last one seems to be the most promising since it is really a basically generic real-time communications network with lots of good implementations of hard-learned lessons gleaned from its creators' experience with older style networks. It is speed and communication medium independent and has the ability to define intercommunication protocols between nodes and ports of different standards (such as MIDI to RS-232, parallel to serial, SCSI to you-name-it) and reads like the ideal system on paper. It has become a sort-of defacto MIDI networking standard since it has always supported MIDI as its basic communication protocol. Lone Wolf seems quite willing to divulge technical information to anyone who signs a non-disclosure agreement so I don't exactly know what that means yet since no other manufacturers are really supporting this protocol within their own software. The systems in use are being used primarily as interconnection devices for lots of MIDI ports in a relatively fixed manner, with some repatching done live as far as I know. I will be getting further info soon. Major drawback with LW is that they are small and not getting the publicity out in volume so are suffering in perception.

This is starting to change since QSC, the second-largest amplifier manufacturer (behind Peavey) have retained LW to design an intelligent amplifier networking system to compete with other proprietary systems such as Crown, Crest, BGW and IED. The obvious crossover between MIDI and pro audio just with this arrangement will be very significant. Where does this leave us? Right where we are for the moment - MSC messages will be treated like any other MIDI message and can be transported over MediaLink like any other (increasingly more) type of data.

TTFN, Charlie.
---[0253]--- (nref = [0254])

[0254] (19 lines) Weber.USITT 10/29/91 1945.4 mst Tue midi
Subject: Re: LDI/CyberArts
Charlie,

I will be at LDI, and look forward to seeing you there.

I will be doing something in the way of a review for the ASCII Text Representation for Lighting Console Data specification, which I am about to upload to CallBoard (tomorrow morning).

I do not have any useful formal presentation ready for Two-Phase Commit MSC. I've promised something for USITT. I don't think I can be ready sooner than that. But, I'm always ready to wave my arms wildly and pretend I know what I'm talking about.

I've never heard of CyberArts. But, then again, there's probably a lot going on in the industry that I've never heard of.

See you at LDI.

Ralph:

---[0254]--- (pref = [0253], nref = [0255])

[0255] (1 line) CRichmond.USITT 10/29/91 2334.2 mst Tue midi

Subject: Re: LDI/CyberArts

Waving arms always works for me.

---[0255]--- (pref = [0254])

[0256] (18 lines) Cotten.USITT 11/09/91 0005.3 mst Sat midi

Subject: EcheLON LON works

Just thought I'd open a discussion on LON works wnd the Echelon Chips.

We (3 of us in electronic/software R&D) expressed interest in Echelon, and now both Motorola & Toshiba started chasing us when they found out we design lighting equipment in volume. We know we may use Echelon for our own highspeed data links, probably using rs-485, but since the device is rather programmable we're looking to see if it can be made into a MIDI or MSC intelligent UART of sorts. Imagine having all the parsing handled in hardware, so to speak. If these chips are what they're hyped up to be, and stay at the \$10 or less range, we may be seeing a lot of them in this industry.

I don't really have a specific point to make, but just wanted to see what y'all have to say on the subject.

See you at the big, loud, obnoxious booth at LDI. (that's the Lightwave Research / High End Systems booth for those of you who are lucky enough to spend most of your time on the quiet (Non-Smoking, or is that Non-Fogging) side. Come say Hi.

---[0256]--- (nref = [0257])

[0257] (64 lines) CRichmond.USITT 11/11/91 0320.2 mst Mon midi

Subject: Re: EcheLON LON works

Hi Lary - You can count on seeing me there - we also will have a booth (in the "quiet" area) so don't hesitate to escape!

Echelon is quite interesting. The AES committee looking for a standard (that isn't its official name but it could be) has been considering them along with MIND-NET from Intelix and MediaLink from Lone Wolf and have had a "shoot-out" between these three in LA in September. None of them seem to be ideal. My understanding of the alternatives is as follows:

Intelix is based on an RS-485 variant (stock hardware, different message data) which allows a couple hundred nodes on each system but is both bound by the RS-485 spec and their own implementation. It is actually a good one but doesn't promise to be more than just a good medium speed network protocol for control/command exchanges. They are pretty well developed in the area of software applications but not as well as the 'new' PA-422 audio standard which is based on an atrocious hardware spec.

Lone Wolf has some amazing hype going with it but it is hard to believe, especially since there is little real networking substance in their literature and mostly just buzzwords and lots of charts with latency, delay and throughput figures calculated on 'typical' and 'worst-case' situations but not apparently measured on any real systems. They also purport to have a \$10 chip in the works which can communicate quite generically at up to 20MBaud. This is attractive to many people because it is very open and supposedly can be implemented over fibre optics, twisted pair, telco, etc. and automatically handles different media and communication speeds, handshakes, etc. They are promoting its used specifically into the media and entertainment industry, having produced a MIDI/MIDI version already which also includes some ability to translate between MIDI and serial, parallel, etc. but the actual transfer codes don't seem to be available easily... The AES people seem to like this the most even though there is virtually no software developed for it (except the resident handler stuff of Lone Wolf which one apparently has to use to get in and out of the system - I can't really tell from the lit, but will be seeing them in LA next week so hope to find out more..)

Echelon certainly has the most elaborate advertising and biggest promo budget but they have some serious glitches in presentation (you must see their promo video!) and it seems to be really industrial in nature. They are not be any means trying to make media types very interested, being highly elitist and not really answering many pertinent questions.

The major drawback of Echelon seems to be its real-time response. They themselves only quote 'about a second' for most commands and we can't really determine its capabilities in terms of large amounts of data throughput. (I don't think there is much - it is basically not fast and relies on short cryptic command exchanges with lots of error checking, correcting and resending/retrying) So, they seem to have pissed the AES people off with their noncaring attitude, but I still think they have the best chance for a truly universal industrial standard. But this does not mean its for us - after all you can buy a nifty little dimmer that goes in a duplex box for \$3.95 but it doesn't mean the same design is being used in the theatre!

My vote for the ultimate answer in multimedia networking still remains with FDDI (or even FDDI-II with a 1.25 GigaBaud transmission rate - truly real-time!) In the meantime, we will stay with 32bit parallel or 125Mbit fibre optic TAXI/FOXI chip data transmission....

See you there!

- Charlie
---[0257]--- (pref = [0256], nref = [0258])

[0258] (41 lines) BRodriguez.USITT 11/17/91 0752.3 mst Sun midi

Subject: Re: EcheLON LON works

Well, I just received some of the LONworks literature, so I haven't had a chance to review it yet. But, based on what I've been told, there are three strikes against it:

First, you have to use their development system, and I understand that it costs \$10,000. There's no way that I'll be able to afford to develop with LONworks.

Second, it's too much of a "black box" for me to really trust it. They provide no details of the ROM support routines, and in particular are very proprietary about their network protocol. This means that either it works or it doesn't, and if it doesn't you are up the creek. No putting a 'scope or logic analyzer on the line or on the chip. And you thought DMX problems were hard to diagnose?

Third, although they've neatly handled the single-source problem for hardware, they are still a single source for software. You have to use their language and their compiler.

BTW, Charlie, if the AES wants to standardize on this, they should investigate the proprietary nature of the protocol. Echelon won't release it. Or would the AES simply standardize on the Echelon chip? (Remember the debate on "who owns SMX?")

On the flip side, my understanding is that the Echelon chip is quite fast, capable of over 1 MBaud. They also claim nearly 100% bus utilization in a "masterless" network, which I flatly refuse to believe until they cough up some protocol details.

My feel for LONworks right now was neatly captured by your summary of Lone Wolf: "some amazing hype going with it but it is hard to believe, especially since there is little real networking substance in their literature..." In fact, most of your description of Lone Wolf sounds like what I've heard of LONworks.

See you all at LDI. I'll try to make it by Richmond Sound and Lightwave Research to say hi. Mostly, I'll be at the Teatronics booth...where, by one of those cosmic coincidences, I'll be demonstrating another product named Echelon.

- Brad

---[0258]--- (pref = [0257], nref = [0259])

[0259] (6 lines) MHefters.USITT 11/18/91 1958.9 mst Mon midi

Subject: Re: EcheLON LON works

We've been subjected to the Echelon hype as well. Yes, it looks like it could be very promising. But, if we were on a LON with several other manufacturers who do some communicating with each other, and something goes awry, who's problem is it??? Our R & D man says the Echelon folks did quite a soft shoe around that and never really answered it. So we're taking a wait & see approach for the time being. -Mitch...

---[0259]--- (pref = [0258])

[0260] (21 lines) CRichmond.USITT 12/11/91 2359.3 mst Wed midi

Subject: Strand's comments

At LDI, Ralph Weber mentioned to me that Strand had some comments and suggestions on the current 2_PC proposal. Ralph, have you had a chance

to put these together or should we pursue Strand to get on board and enter them in themselves?

Also, a number of European companies who were concerned about being left out of these discussions and the difficulty of joining Callboard from Europe approached me about setting up their own E-mail network and conducting discussions amongst themselves. I encouraged them to do this, telling them who was already involved over there and confirming that I would be willing to upload the summaries of their discussions to this forum while providing them with the results of ours.

It will be interesting to see what develops on these fronts in the next little while....

By the way, we had, I think, a very good panel discussion about Show Control and MSC specifically at LDI - this is definitely creating a lot of positive interest (plus a little controversy).

- Charlie
---[0260]--- (nref = [0261])
[0261] (11 lines) Dignan.USITT 12/13/91 0112.2 mst Fri midi
Subject: Re: Strand's comments
Charlie,

I think it is grossly counterproductive to encourage the development of more than one standards and technical discussion forum worldwide. It is THE great feature of this system that it is genuinely international and allows roughly equal easy access to the process for everyone.

Anders and Matt do OK, why not these others? If there is a technical problem, let's solve that; not compromise the usefulness of this system.

Joe /
---[0261]--- (pref = [0260], nref = [0262])

[0262] (49 lines) CRichmond.USITT 12/13/91 0139.1 mst Fri midi
Subject: Re: Strand's comments
Thanks so much for your comments, Joe. I must say I agree but you don't know how difficult it has been to get most of the other world wide manufacturers to join us here. Also, the heat put on me for not trying harder (if that was possible) made me think these lighting guys had found some new high temperature filaments.

To be fair(er), I must also explain that there is some truth to their complaints about difficulties in getting onto Callboard. Risking the wrath now of those reading this, I got repeated major complaints voices (voiced) about not getting phone calls returned from USITT, the MMA and also (yes -) Callboard people. Seems these organizations just don't seem to be all that interested in signing people up from the far reaches of the earth. Not only that, but (and I know this personally because I have tried to assist) it is extremely difficult to login to Callboard from a lot of places - and once done it is unbelievably expensive. The only real facts I have are from a customer of ours whom I convinced would find it worth his while to participate: After many months of searching out various ways of getting on from Oslo, he had to phone long distance to Sweden to tie into an expensive system which then charged connect time to somewhere else and then paid to somewhere in the US that then provided access to the Tymnet port (he could not actually get

access directly to Canada). The 3 or 4 times he logged in over the first month netted him a bill of over \$100US so he quit. Not only that, but he is an experienced modem user and simply downloaded newer messages as quickly as possible then logged out. Needless to say, I had to apologise for the time and money wasted.

So - put yourself in their shoes: Just to do this, you have to join USITT, join MMA, join Callboard, then figure out how to login and then start spending the big bucks. I'd seriously start asking someone to just copy me with faxes, too. Things are a lot easier for us homebodies.

Finally, given this perspective, I have to say in my defence that I did not consider it grossly counterproductive to suggest a way in which these people (who were quite adamant about getting involved) could participate without these difficulties. After all, the objective is to get more involved and general agreement over a wide range of manufacturers, not to get more people onto Callboard (if CITT wants to do that, I have gone on record as saying they should make it easier, but that's the most I can do). The MMA has their own network which is marginally easier to access internationally, but after six months I couldn't get anyone to join that forum, so we decided to start things here (see all the early messages here - by anyone I mean the people that ended up making this happen here)

Enough. Any more suggestions? We're all ears (electronic style).

- Charlie

---[0262]--- (pref = [0261], nref = [0263])

[0263] (7 lines) Weber.USITT 12/20/91 0611.8 mst Fri midi

Subject: Re: Strand's comments

I have Strand's comments. Most are minor concerns. I've got to go soon, so I cannot type in the full list of six comments now. The major concern is a request for formalizing error recovery for checksum errors. I've thought about this some and it has problems for controlled devices. They have no options in some cases.

Ralph::

---[0263]--- (pref = [0262], nref = [0267])

[0264] (11 lines) Weber.USITT 12/20/91 0615.1 mst Fri midi

Subject: Foreign CallBoard access

Regarding European access to CallBoard:

I must note that Anders Ekvall recently sent me paper mail rather than logging into CallBoard. By using paper mail, he risked my not receiving the information before LDI. And, he was fully aware that he could contact me on CallBoard; he had done so many times previously.

So, I must conclude that Anders had some significant motivation to use paper mail, despite its disadvantages.

Ralph::

---[0264]--- (nref = [0265])

[0265] (8 lines) CRichmond.USITT 12/21/91 0102.6 mst Sat midi

Subject: Re: Foreign CallBoard access

Yes - I suspect the motivation is the difference in cost between postage (expensive as it is) and that of European login just to send a brief(?) message.

But... enough of the suspense: the real question here is, "Did he make it?"

- Pony (or in the case of Anders, Unicorn) Express
---[0265]--- (pref = [0264], nref = [0266])

[0266] (3 lines) Weber.USITT 12/21/91 1504.4 mst Sat midi
Subject: Re: Foreign CallBoard access
The mail got through before I left for LDI. :-)

Ralph:
---[0266]--- (pref = [0265])

[0267] (90 lines) Weber.USITT 12/21/91 1505.0 mst Sat midi
Subject: Re: Strand's comments
> The following is the complete list of issues raised by Strand.
>
> My tentative responses appear on lines beginning with ">"

The issues raised by Strand are:

- 1) Suggest adding formal definition for recovery from checksum and cue number errors. Recovery from these errors should be possible without operator assistance.
>
> I will reconsider the existing recovery procedures in this area.
> However, I'm confident that a complete solution is not possible.
> In the case were a controller gets a message with a bad checksum
> from the controlled device, the controller must act as if the
> controlled device sent an ABORT message. The specification contain
> no mechanism for the controller to ask the controlled device to
> retransmit the message. Adding explicit retransmit capabilities
> to the protocol would increase its complexity.
>
2) Section 4.5, STANDING_BY message: Suggest specifying the "Maximum number of seconds required to execute the cue" parameter in the standard time specification format. See TIMED_GO for an example.
>
> This looks like the MIDI think to do. But, it's big time overkill.
> I'll make this change, unless Charlie tells me not to.
>
3) Section 4.5, STANDING_BY message, regarding the following statement:

"In the previously send STANDING_BY message, the controlled device has informed the controller of the maximum number of seconds required to execute this cue."

Does this override cue timing?

- >
> No. It provides the controller with a way of knowing whether the
> controlled device broke during execution of the cue. If the controlled
> device does not send a COMPLETE or an ABORT message in this time, then

> the controller must assume that the controlled device is broken, which
> is the concept behind the remainder of this paragraph.
>

4) Section 6.4.3, regarding the 4th paragraph:

"Note: controlled devices cannot send ABORT messages except in response to STANDBY or GO_2PC messages. From the point of view of the larger presentation, the exceptional condition does not become significant until its existence prevents proper cue execution. However, controlled devices may still initiate local actions the instant the exceptional condition is detected."

Why not CANCEL?

>
> The omission of CANCEL appears to have been an oversight.
> However, I must also note that the only status code allowed in
> such an ABORT message would be "checksum error."
>

5) Section 6.5, regarding the 3rd sentence in the 2nd paragraph:

"Next, the <device_ID> byte is added to the sum."

Why not add both the <device_ID> and <msc> bytes to the two byte sum?

>
> If the <msc> byte was garbled, then the presence of the MIDI Show Control
> protocol was not recognized and we never got to the checksum validation
> logic. Therefore, including the <msc> byte adds no relevant validation
> information. Finally, I can justify omission of <msc> by noting that
> every datum included in the checksum forces some validation of the other
> data to be lost.
>
> The only justification I can see for adding <msc> is algorithmic
> simplicity. But, unless somebody makes a strong case to the contrary,
> I cannot justify the loss of data validation based on a need for
> algorithmic simplicity.
>

6) Section 6.7: Why are status code values constructed differently from
sequence number and checksum values?

>
> Someday, I want MSC 2PC transmitted on some transport layer other than
> MIDI. (SMX was my choice at the time this spec was written.) When that
> happens, I expect that what are 14 bit values in MIDI will become 16 bit
> values. In the case of status code values, I want the two extra bits
> available in a way that keeps the code values easy to read. Also, I want
> the MIDI and non-MIDI status code values identical in the 14 MIDI bits.
> This arrangement satisfies my goals.
>

---[0267]--- (pref = [0263], nref = [0268])

[0268] (7 lines) CRichmond.USITT 12/22/91 0127.4 mst Sun midi

Subject: Re: Strand's comments

On initial reading, Ralph, everything you have said sounds quite logical. Let's sit on it for a while and see what comes to mind, and what other comments we get....

Thanks

- Charlie

---[0268]--- (pref = [0267], nref = [0280])

[0269] (6 lines) AEkvall.USITT 12/27/91 1052.2 mst Fri midi

Subject: am back again

I am back again! I have been out of the Callboard for a while because we had some trouble to get the membership fee to USITT. Thats why I have been so quiet for a while and also the reason for sending paper mail (aaarg) to Ralph.

Anders

---[0269]---

[0270] (13 lines) AEkvall.USITT 12/27/91 1116.5 mst Fri midi

Subject: Overseas connection

I am surprised to hear about the big problems on getting connected to Callboard from Europe. The service I use from Sweden is called Datapak and it should be avilable from most countries in the world.

Regarding the price, we have paid about USD100-150 for a whole year of connection. And also I have been talking quite a lot duriong that year. The datapak is as I can see both faster and cheaper than anything else I have seen.

Thats all for this time.

Anders ..

---[0270]--- (nref = [0271])

[0271] (4 lines) CRichmond.USITT 12/31/91 0134.6 mst Tue midi

Subject: Re: Overseas connection

Thank you Anders for giving us this information - it is going to be very helpful in assisting others to join us (when we hear from them!)A

- Charlie

---[0271]--- (pref = [0270])

[0274] (187 lines) CRichmond.USITT 02/03/92 0030.2 mst Mon midi

Subject: Pep talk

I was supposed to go to the MMA meeting at the NAMM show in Anaheim but didn't. In my stead, I produced the following document which was provided to all there and subsequently (by request) put on the PAN MMA Forum so that all MMA members could see (and possibly benefit?) from it. Since things have been a littel dead around here lately, here it is just to liven things up a bit.....

MIDI SHOW CONTROL and MULTIMEDIA - A marketing position

CHARLIE RICHMOND, Richmond Sound Design Ltd., Vancouver, Canada

MIDI Show Control (MSC) has evolved quickly, becoming a standard which is partly misunderstood, even by members of the MMA. In order to dispel this misunderstanding we must promote MSC both for what it is now

and for its future potential. Although born of humble roots, MSC has the power, universality and flexibility to be the backbone of a complete MIDI multimedia communication and control standard.

INTRODUCTION

Ongoing entertainment for me these days is to watch the computer industry grapple with the meaning of the phrase "multimedia." Just a little while ago I read yet another editorial which began "Ask anybody what Multimedia is and you'll get a different answer....." Perhaps the computer wizards trying to convert us to their brand really believe that multimedia is composed of only what can be displayed by a video projector (or slides of high resolution monitor graphics or videotapes made frame-by-frame) along with sound from a 3" speaker amplified by a microphone (or in some cases, a cassette deck playing "Dolby Mono" from one track and triggering frame changes with the other.) But I think those of us who have been involved with live performance production know better.

I guess a lot of those "computer people" have never been to a good live show or else they might realize there is a lot of media involved in performance technology. It's only my opinion, of course, but surely any multimedia specification should be able to include virtually all current and future technical elements involved with presentation and show technology. I suspect the reason MSC has not been identified by some MMA members as something which could do this is that its development began under the uninspiring title "Theatre Lighting Cues Proposal." The reason it has not been identified as a control solution by the multimedia industry may be just as much a lack of publicity as the lack of awareness of the real components of multimedia. Finally, the reason the public has not identified MSC as the key to multimedia production is because few of the industry gurus have discovered it and are touting it.

We need to change this state of affairs!

MIDI SHOW CONTROL TODAY

MSC is an almost open-ended protocol that can be easily expanded. It currently defines 55 media elements (called Command_Formats) within seven general categories and has the room to add an almost unlimited number. It can address 112 individual devices and 15 groups within each Command_Format. Twenty-six commands with varying complexity and multiple extensions have been defined, again with the ability to expand almost without limit. Cues, Cue Lists and Cue Paths may be identified by numbers containing as many as 50 digits.

Systems incorporating MSC are currently lighting big Broadway shows including "Miss Saigon," "Will Rogers at the Ziegfield Follies" and, briefly, "Nick and Nora." They are running lavish productions in Las Vegas including "Siegfried and Roy," "Into the Night" and the new audio system for the Mirage Hotel's 'Volcano.'

At Disney World, they are controlling Disney/MGM Studios' "Sorcery in the Sky," "Indiana Jones Epic Stunt Theatre" and "The Little Mermaid Show," Magic Kingdom's "SpectroMagic Parade," "Anniversary Parade" and "Surprise Celebration Parade" and Pleasure Island's "New Year's Eve Celebration." At Universal Studios they run "The Riot Act" (Hollywood) and "The Wild, Wild, Wild West Show" (Florida). They were even used to

control the production elements for Nintendo's touring commercial display entitled "PowerDome Tour 1991" and will soon be doing multidimensional sound playback control for an IBM industrial road show. And Nintendo and IBM are probably totally unaware of how their shows are being controlled!

But as impressive as these credits are, these are not the clients that represent the bulk of the potential market for MSC. The people we need to reach are the same ones who helped MIDI redefine the musical world - the users - and the way we have to reach them is the same way it worked back then. When I say all of the above shows were done with systems incorporating MSC, what that really means is that MSC is currently used to a very minor extent. We couldn't use MSC to communicate with most show equipment because it doesn't support it yet. This is understandable since MSC has just been released but from now on we have to make sure everyone puts MIDI in their equipment - not just manufacturers of musical gear!

For example, instead of controlling slide projectors with custom interfaces, solid state relays and PLC's, (or even the "black magic" of expensive A-V controllers), we should be able to simply plug MIDI cables into the projectors, set their 'Device Number' switches as appropriate and start sending MSC 'Slide Projector' messages. When it becomes this easy to control all these previously obscure devices at the same time and using the same cables as we use to run our musical equipment, we will see a renaissance in show technology and experimentation with new media forms by people who previously had neither technical expertise nor the money to buy it.

CURRENT MSC PROTOCOLS

COMMAND_FORMATS: (112 Devices/15 Groups addressable within each Format)

Lighting	Machinery	Projection
Moving Lights	Rigging	Film Projectors
Color Changers	Flys	Slide Projectors
Strobes	Lifts	Video Projectors
Lasers	Turntables	Dissolvers
Chasers	Trusses	Shutter Controls
	Robots	
	Animation	
Sound	Floats	Process Control
	Breakaways	
Music	Barges	Hydraulic Oil
CD Players		H2O
EPROM Playback		CO2
Audio Tape Machines	Video	Compressed Air
Intercoms		Natural Gas
Amplifiers	Video Tape Machines	Fog
Audio Effects Devices	Video Cassette Machines	Smoke
Equalizers	Video Disc Players	Cracked Haze
	Video Switchers	
	Video Effects	
Pyro	Video Character Generators	

Video Still Stores
Fireworks Video Monitors
Explosions
Flame
Smoke Pots

COMMANDS: (Most commands can specify Cues, Cue Lists & Cue Paths - each labeled with up to 50 digit numbers)

```
GO    FIRE    GO/JAM_CLOCK START_CLOCK MTC_CHASE_OFF
STOP  ALL_OFF STANDBY_+  STOP_CLOCK  OPEN_CUE_LIST
RESUME  RESTORE STANDBY_-  ZERO_CLOCK  CLOSE_CUE_LIST
TIMED_GO RESET  SEQUENCE_+  SET_CLOCK  OPEN_CUE_PATH
LOAD   GO_OFF SEQUENCE_-  MTC_CHASE_ON CLOSE_CUE_PATH
SET
```

You can see why, with all these media and commands along with tremendous expansion capability, we believe this is the protocol with which "multimedia" control can flourish. If there is a category above which lacks a logical relationship with any device into which a manufacturer wishes to integrate MSC, that company is invited to join the on-going MSC working group discussions in the "MIDI Forum" on the Callboard Network just as other MIDI proposals are discussed on PAN. Any proposal to add to these categories can certainly be seriously considered.

FUTURE PLANS

We are in the final stages of defining the next proposed addition to MSC which adds 'Two-Phase Commit' commands that are designed to operate potentially dangerous equipment in a completely fail-safe manner. We are currently prescribing how this equipment should respond if communication errors occur or if an E-stop condition is invoked, and we invite all interested parties to participate. Involved in this process are some of the most qualified people in the theatre industry plus a highly capable network software designer from Digital Equipment Corporation (DEC). I have contemplated proposing a few additions to the Commands list because we have occasionally not had a standard Command available to do exactly what we wanted. We have so far managed by using the next most logical one but obviously this won't be the case forever.

It will be exciting and rewarding to get some "Multimedia Types" involved to improve, expand and refine the MSC specification. We need Intensive Care as well as active and influential members of the MMA to get some interest and excitement generated for this new standard which holds the key to the future of true Multimedia!

TTFN

- Charlie
---[0274]--- (nref = [0276])

[0275] (36 lines) CRichmond.USITT 02/03/92 0045.6 mst Mon midi

Subject: CD Players

An associate/friend in LA (Ken Hirsch) has been retained by Philips to create a MIDI-based control language for a remote control unit to talk with a consumer CD Player (audio variety, not CD-ROM or other variant) and he wants to incorporate MSC in its fullest implementation. I told him that my ideal MSC implementation for CD Players was to have at least GO, LOAD (equivalent to go to a specific location and pause, waiting to play), STOP (equivalent to Pause on), RESUME (equivalent to Pause off), and RESET (equivalent to Stop).

I described a cue number hierarchy where the cue number defines the cut number then the index number then the frame number, for example cue no. 5.3.2654 means "cut 5, index 3 (within cut 5), frame 2654 (within index 3). Incidentally, he informs me that CD's apparently have 75 frames per second.

Cue list would refer to the CD position within players which can hold more than 1 CD, such as a 'jukebox' player; Cue_path would refer to the number of the CD which can be physically shuffled into the playing position.

For example, if a player can hold 20 CD's for random access and can mechanically exchange CD's from holders into the 20 positions from say, 50 storage slots, then the unit can hold a maximum of 1000 CD's and would have a total of 20 cue lists and 50 cue paths (per list).

Ken says they are committed to doing this and it will be on the market within 6 months! (the consumer unit, that is - not the jukebox version as described)

Let us know here if I have overlooked something important in this proposed implementation, or if you think there is a better way to do this!

Thanks,

Charlie

---[0275]---

[0276] (14 lines) Cotten.USITT 02/28/92 2354.1 mst Fri midi

Subject: Re: Pep talk

Just to let you know where MSC might just be headed soon, New controllers for ALL of Lightwaves products (officially due this summer) will be supporting MSC, and I spoke with Avo Lites, and they seemed interested. Also, I will be meeting with ETC in a couple of weeks to discuss compatibility issues, and I intend to push MSC as the means to this end.

I've managed to make Show Control a buzzword with all of our sales staff. I d't know if this is good or bad, but now everyone at least thinks they want MSC.

I'm in the renewal process, so I may not be responding to email in the next few weeks. If you want to contact me use your desktop voice network terminal and transmit to address 512 832 0363.

---[0276]--- (pref = [0274], nref = [0277])

[0277] (9 lines) CRichmond.USITT 02/29/92 0136.7 mst Sat midi

Subject: Re: Pep talk

Great work - we are having another (!) panel discussion about MSC at the USITT conference next week and the head cheese of ETC is on it apparently to be the devil's advocate (against) in the argument. Seeing as how I have never managed to get him to respond to my requests to talk with them about MSC nor have I been able to get him to say why he doesn't like it, this promises to be perhaps enlightening and hopefully instructive to all of us....

- Charlie (ps, try to make it to the forum)

---[0277]--- (pref = [0276], nref = [0278])

[0278] (3 lines) Clinton.Thea 02/29/92 1741.1 mst Sat midi

Subject: Re: Pep talk

I anyone is in contact with Larry over the "voice network terminal" they can let him know I've advanced his renewal date on the strength of his "the cheque is the mail".

---[0278]--- (pref = [0277], nref = [0279])

[0279] (1 line) CRichmond.USITT 03/08/92 2335.8 mst Sun midi

Subject: Re: Pep talk

We'll see what we can do....

---[0279]--- (pref = [0278], nref = [0284])

[0280] (4 lines) Weber.USITT 03/09/92 1958.3 mst Mon midi

Subject: Re: Strand's comments

The next transaction will be my final response to Strand's comments.

You will notice that I've softened a bit on some points.

Ralph:

---[0280]--- (pref = [0268], nref = [0281])

[0281] (105 lines) Weber.USITT 03/09/92 1958.6 mst Mon midi

Subject: Re: Strand's comments

- 1) Suggest adding formal definition for recovery from checksum and cue number errors. Recovery from these errors should be possible without operator assistance.

Response:

I am adding a new section that discusses automated recovery procedures. It will immediately follow the exceptional condition handling section (6.4.3) and will be referenced by that section. N.B. Simple retransmission is possible only when the controller receives an ABORT message from a controlled device. When the controller receives a garbled message (checksum error) from a controlled device, there is no way for it to ask for a retransmission. I will cover this case by instructing the controller to CANCEL outstanding activity in the controlled device and reinstate it using new STANDBY messages.

- 2) Section 4.5, STANDING_BY message: Suggest specifying the "Maximum number of seconds required to execute the cue" parameter in the standard time specification format. See TIMED_GO for an example.

Response:

This looks like the MIDI think to do. But, it's big time overkill.
I will make this change.

3) Section 4.5, STANDING_BY message, regarding the following statement:

"In the previously sent STANDING_BY message, the controlled device has informed the controller of the maximum number of seconds required to execute this cue."

Does this override cue timing?

Response:

No. It provides the controller with a way of knowing whether the controlled device broke during execution of the cue. If the controlled device does not send a COMPLETE or an ABORT message in this time, then the controller must assume that the controlled device is broken, which is the idea behind the remainder of this paragraph.

I also must note the importance of the word maximum in this statement. This time must be the absolute maximum time required to execute the cue. For example, if operator confirmation is required to execute the cue, the maximum time must include time for the operator to provide confirmation.

I will update this text slightly.

4) Section 6.4.3, regarding the 4th paragraph:

"Note: controlled devices cannot send ABORT messages except in response to STANDBY or GO_2PC messages. From the point of view of the larger presentation, the exceptional condition does not become significant until its existence prevents proper cue execution. However, controlled devices may still initiate local actions the instant the exceptional condition is detected."

Why not CANCEL?

Response:

The omission of CANCEL appears to have been an oversight. However, I also must note that the only status code allowed in such an ABORT message would be "checksum error."

5) Section 6.5, regarding the 3rd sentence in the 2nd paragraph:

"Next, the <device_ID> byte is added to the sum."

Why not add both the <device_ID> and <msc> bytes to the two byte sum?

Response:

If the <msc> byte was garbled, then the presence of the MIDI Show

Control protocol was not recognized and we never got to the checksum validation logic. Therefore, including the <msc> byte adds no relevant validation information. Finally, I can justify omission of <msc> by noting that every datum included in the checksum forces some validation of the other data to be lost.

The only justification I can see for adding <msc> is algorithmic simplicity. But, unless somebody makes a strong case to the contrary, I cannot justify the loss of data validation based on a need for algorithmic simplicity.

6) Section 6.7: Why are status code values constructed differently from sequence number and checksum values?

Response:

Someday, I want MSC 2PC transmitted on some transport layer other than MIDI. (SMX was my choice at the time this spec was written.) When that happens, I expect that what are 14 bit values in MIDI will become 16 bit values. In the case of status code values, I want the two extra bits available in a way that keeps the code values easy to read.

Also, I want the MIDI and non-MIDI status code values identical in the 14 MIDI bits. This arrangement satisfies my goals.

---[0281]--- (pref = [0280])

[0282] (12 lines) Weber.USITT 03/09/92 1959.8 mst Mon midi

Subject: 2PC V0.7 Uploaded

I have uploaded the V0.7 draft of the MIDI Show Control - Two Phase Commit specification. This draft incorporates the changes resulting from Strand's comments. (See the previous forum transaction.)

I have asked Tim Clinton to place this draft in:

>udd>USITT>common>standards>MSC_2PC_0.7

WARNING: This document is 75 Kbytes in length and occupies 27 pages, type written.

Ralph:

---[0282]--- (nref = [0283])

[0283] (1 line) Clinton.Thea 03/10/92 1248.0 mst Tue midi

Subject: Re: 2PC V0.7 Uploaded

The MSC_2PC_0.7 document is in place in the standards directory.

---[0283]--- (pref = [0282])

[0284] (5 lines) Cotten.USITT 03/31/92 2327.6 mst Tue midi

Subject: Re: Pep talk

i'm back, i think?

charlie, how'd USITT go? I met with ETC (actually they came here), and they seemed to be committed to MSC. They wanted to choose a medium or method for attachg our equipment together. MSC, of course.

---[0284]--- (pref = [0279], nref = [0285])

[0285] (18 lines) CRichmond.USITT 04/01/92 0211.4 mst Wed midi

Subject: Re: Pep talk

Bravo, Lary! Actually, I had a panel discussion with Anne Valentino who confirmed at that time that they were committed to MSC - they just weren't sure how soon since they were somewhat committed in the R&D department to the development of other things and were a little stretched. Seems as though this was really the only hesitation they had had all along.... but now everything seems to be a GO (or possibly a TIMED_GO)

Incidentally, we have been getting a lot of inquiries from users about High End's support of MSC in its various products and we could only confirm that you were working on it - would you be able to let us know here (or even in the commercial_meeting) what products you now have on line that use MSC. We need to start a database of equipment that has it for the use of our customers!

Thanks and welcome back...

- Charlie
---[0285]--- (pref = [0284], nref = [0286])

[0286] (3 lines) JLBracewell.USITT 04/01/92 1218.7 mst Wed midi
Subject: Re: Pep talk
Charlie, is it at all a possibility that that database might be made public here? I think a lot of us would like to know who's got what when.
---[0286]--- (pref = [0285], nref = [0287])

[0287] (5 lines) CRichmond.USITT 04/02/92 0148.4 mst Thu midi
Subject: Re: Pep talk
Sure - just as soon as we have the db put together ourselves!

Watch this space....

- MSC
---[0287]--- (pref = [0286])

[0288] (14 lines) CRichmond.USITT 04/15/92 0008.9 mst Wed midi
Subject: new eyes
A developer of CDTV, Hedley Davis, expressed considerable interest in the MSC_2PC proposal, asking to view the document and promising some comment. I am looking forward to his comments, which are the only ones we are likely to get from an 'outsider,' based on the amount of interest we've received to date.....

All will be posted here as received. Also, I would suggest everyone get a copy of the new MIDI Machine Control 1.0 standard from the MMA- it is similar to MSC in some respects, especially in that it is both open-loop and closed-loop compatible. There are considerably more commands and the controller/controlled_device are intended to work more closely in time than in MSC, but it is a good study.

- Charlie
---[0288]--- (nref = [0291])

[0289] (7 lines) CRichmond.USITT 04/24/92 0117.9 mst Fri midi
Subject: Dabase
We are compiling a database of equipment that is MSC compatible (mostly lighting gear at this point) since we are getting asked a lot these

days. If you know of manufacturers and their model numbers that are specifically MSC (not just MIDI) standardized, please let us know here - Thanks!

- Charlie
---[0289]---

[0290] (5 lines) CRichmond.USITT 04/24/92 0120.1 mst Fri midi
Subject: Out of town
I will be out of town for two weeks starting tomorrow so feel free to take over this forum as anarchically as you wish. All will be put to rights upon returning...

- Charlie
---[0290]---

[0291] (273 lines) CRichmond.USITT 05/14/92 2351.4 mst Thu midi
Subject: Re: new eyes

As you will remember, I sent a copy of the proposal to Hedley Davis at Commodore/Amiga since he was very interested. Here is his response:

25 (107 lines in body):
Delivery-Date: 11 May 1992 16:31 mdt
Delivery-By: Network_Server.Daemon)
Date: Friday, 8 May 1992 16:35 mdt
From: hedley at CBMVAX.CBM.COMMODORE.COM (Hedley Davis)
Subject: Re: Proposal
To: CRichmond at UNCAMULT, hedley at CBMVAX.CBM.COMMODORE.COM

Charlie,

Attached are comments to the proposal you forwarded to me. I just reread it, having written it a couple of weeks ago, and it sounds a little flippant. That is not the intent. I am just asking questions, and trying to poke a well intended hole or two.

Anyway, here it is.

Charlie,

Well you were correct, the spec is clearer once one takes the time to go all the way through it.

I understand the basic concepts of what the document you sent to me says (standby/standingby/go2pc/complete), and I appreciate the efforts to elaborate on the command sets, and on the error conditions.

I have two questions, a comment, and a recommendation.

1) Who or what is N.B. ?

2) How is the 'datapath back to the host' to be achieved ?

I assume you are layering some other network protocol underneath the two phase commit protocol which is suited somewhat to MIDI. What exactly do you have in mind ?

If this is true, then why are you specifying MIDI as the standard ? Is this simply a marketing/acceptance issue ? Or is the goal to allow these new 'safer' device to be intermixed with other MSC devices ?

If this is false, then how to plan to handle the interconnect problem ?

3) I was unhappy to read that the specification is foregoing Rock and Roll in favor of the more controlled environment of repetitive staged performances. Clearly, large set changes in even Rock and Roll tend to be repeated night after night in the same order. The lack of real time response is a hindrance to the utility of the standard.

The crux of the issue here is coordination between devices not originally designed to be coordinated with each other. For example, if a buyer purchases motor controllers, and requires four of them for a specific stage motion, then the buyer will also want them to act in a coordinated fashion in the event of failures of any single device.

The problem is one of coordination. A motor controller will take a finite time to decide that its motor is 'stuck'. It then needs to respond to the system controller, which then needs to inform the other three devices to stop. If communications to any of these controllers fails after things have started moving, (a very real possibility), then the all able controllers will happily complete their cue. Needless to say, if these devices are all attached to the same mechanical thing, then extremely dangerous stresses can be created.

Cues are a fine thing. All sergeants in an army should understand the 'plan'. But these same sergeants should know it is not safe to march ahead when another sergeant fails to cover his flank.

I guess what I think needs to happen is that some network bandwidth needs to be expended on assuring the integrity of the network, and that the specifications should contain a methodology to deal with losing 'critical' network sections.

The problem with that solution is similar to the problem you are already coming to terms with: "What does Abort really mean ?".

4) This brings me to my final question/comment/recommendation.

In order to deal with the problem of the various possible responses to exceptional conditions, it would seem logical

that each device would have a programmable response. This again is additional complexity for the controller.

I therefore recommend that you consider changing the standby command to explicitly state the action(s) the controller should take in the case of the CANCEL command. (If the controlled device is smart enough to understand what the cue is about in the first place, then it ought to be able to understand being told what to do with errors).

This way, there is never any doubt on the part of the system programmer as to what will happen in response to the things he really doesn't want to think about.

I admit this is not as well thought out as I or you would like. It's just an issue that needs to be addressed.

I'm really interested in the interconnect issue. How are you going to hook these puppies up ?

Best Regards,

Hedley

And here is my response to him:

Hedley,

Thanks very much for your interest, for taking the time to consider this proposal and for your thorough response. I am putting your comments along with this letter of mine in the forum so that others may reply to both. I will pass along all additional replies to you in due time...

I will attempt to respond to your questions as well as I can and perhaps Ralph Weber or others may be able to assist, so here goes (in order):

- 1) My unabridged dictionary gives the definition of this as "note well (from the Latin 'nota bene')". I would have said it means "important note:"
- 2) Although there are emerging networks and proposals which attempt to provide a proper physical network layer for MIDI, the standard way to provide a return path to the host is to connect the MIDI OUT's from the desired controlled devices to MIDI IN's of a device called a 'merger' or 'merge box.'

Mergers are designed to intelligently combine MIDI messages coming to it asynchronously and from various sources simultaneously into a single MIDI message stream. They usually can store fairly long messages from each input temporarily in RAM and then assemble them (in order) into a properly formatted message string.

Mergers can be cascaded to provide a very large number of inputs to be merged. The largest number of inputs usually

available in a single merger is eight. One of the advantages of using MIDI this way is that only the controlled devices which must use 2PC need to send back to the host, so the 'network' requirements are actually fairly minimal, since devices not using 2PC commands need only the one-way path of standard commands.

3) This actually has several parts: A. You are correct that the 2PC variant of MSC is designed for use within a very controlled environment. However, the first version of MSC (V1.0, which is already approved and being used) specifically addresses the need for real time, spontaneous 'rock & roll' response. In fact, it is such a freewheeling standard, without any error checking or prescribed responses, that the people who do things that are dangerous have expressed their concern that a 'safe' spec must be developed which can be used within the same show framework. This is specifically why 2PC is being developed - and the two approaches to show control are completely compatible provided the show controller can handle all variations of commands and responses.

B. In fact, this brings me to the response to your next concern - if the show controller is intelligent enough (or rather, sufficiently programmable) then the scenario which you describe can be dealt with quite nicely by creating a 'script' in the controller which will send an 'abort' or 'stop' or whatever you wish to all running motor controllers if it receives an 'error' or 'abort' or no response from a defective motor controller. The point is - the language for doing all of this now exists and is couched in normal show terms rather than cryptic computer language, so will hopefully be easier for show production people to deal with. The ability of a system to deal with a scenario such as you describe will be dependent only on the capabilities of the show controller and is not limited by the range of the MSC_2PC language (or at least we hope so, and will 'fix it' if it doesn't). As always, the qualities of systems response is dependant on the quality of the programming, not the inherent capabilities of the control language.

C. My belief is that certainly all but the most complex shows will be well served with existing MIDI bandwidth since all commands are fairly cryptic and the system should not be clogged because raw data must be exchanged - only commands. If network integrity is there, then all messages will get through. In 2PC, the protocol is such that if responses do not get received as expected an error occurs which gets reported by the controller to the operator and the show or certain portions of the show will stop. This is subject to the way the show is programmed, of course, but is as it should be since we are dealing with such a tremendous variety of performance media. To set in stone within the spec exactly how the system should respond to errors, aborts or other commands may unnecessarily limit the available range of acceptable responses, and we feel that the current proposal provides a reasonable trade off between mandatory responses and ones which remain under programmer control. There are also the standard 1.0 commands available for use which incur no penalties of this nature, as well. A further command set which addresses absolute system responses in the face of

prescribed errors may become the text of yet a further extension of MSC in the future...

4) This suggestion is quite a good one and I'd like to have Ralph Weber, who authored the proposal, respond to it. I have my own ideas (basically that it is worth doing in some form) but want to see what others have to say. Some of the options open to us are:

A. amend the CANCEL command to include data values (such as used in the GO_2PC command) prescribing specific response options (such as outlined in the CANCEL text plus future needs)

B. add a new command such as 'HALT' which is similar to CANCEL but includes the data values to define desired action.

C. add individual commands as necessary which specifically define action to be taken (using the examples 1-4 in the text for the CANCEL command, these could be defined as FINISH, WAIT, TERMINATE and REVERSE - to which the controlled device would send FINISHED, WAITING, TERMINATED or REVERSED - and COMPLETE or ABORT if appropriate.)

A brief discussion of show politics might be appropriate. At the moment, CANCEL is essentially a 'power neutral' command - meaning that the show control operator can send a CANCEL but the person who is in charge of or has programmed the controlled device has the sole ability to determine what their device's response to CANCEL is in each possible circumstance. This provides both people with an appropriate and reasonable amount of control over the situation.

It is possible that giving the show controller absolute control over the response of the controlled device when CANCEL is sent may be seen as having too much power by the person responsible for the programming of a controlled device that is in charge of dangerous equipment. I suspect the programmer of a rigging system would want to make sure a flat dropping in will definitely stop dead if CANCEL is received and would be extremely nervous if they knew that someone else could override the safety responses for which they are basically responsible.

Of course, even this could be accommodated by an intelligent controlled device: the device could have a way in which its programmer could tell it to ignore all but certain specific CANCEL options which comprise the set of acceptable responses to the CANCEL command. There's always going to be a way of doing the show to the satisfaction of all concerned and within the realm of compromise!

And people will hook these puppies up any way they like - what ends up happening will, as always, be completely up to them!

Thanks again - talk to you soon,

Charlie

And now - what about everyone here? Surely some comments are to

be had...

---[0291]--- (pref = [0288], nref = [0292])

[0292] (8 lines) LDalman.USITT 05/16/92 0015.2 mst Sat midi

Subject: Re: new eyes

Charlie:

In responding, you touched the nerve of "that guy who is responsible"
WRT when the shit hits the fan what the Hell do we do here?

How far can we second-guess every single possible scenario?

To the point of safety, we hope!

---[0292]--- (pref = [0291], nref = [0293])

[0293] (6 lines) CRichmond.USITT 05/17/92 0004.5 mst Sun midi

Subject: Re: new eyes

I'm sure your nerve was tweaked! You are right to be concerned and there is no intent to avoid creating the absolute safest scenarios. Hedley has also responded and I am writing a response - both will be put here soon, so standby (standing_by)....

- Charlie

---[0293]--- (pref = [0292], nref = [0294])

[0294] (163 lines) CRichmond.USITT 05/17/92 0158.1 mst Sun midi

Subject: Re: new eyes

OK - here is Hedley's latest message and my response:

27 (80 lines in body):

Delivery-Date: 16 May 1992 10:42 mdt

Delivery-By: Network_Server.Daemon)

Date: Saturday, 16 May 1992 10:09 mdt

From: hedley at CBMVAX.CBM.COMMODORE.COM (Hedley Davis)

Subject: Re: Proposal

To: CRichmond at UNCAMULT, bitnet!uncamult.hedley at PRINCETON.EDU

Hi Charlie,

Thanks for the response. Either I have failed to communicate something to you, or I do not understand your response to my hypothetical motor control scenario. Allow me to try again.

The scenario involves two motors which must work together. Anytime one motor moves, the other must also move in concert. All situations where only one motor moves are to be avoided in the name of safety. Each motor is a 2PC controlled device.

The problem occurs as follows:

- 1) Each motor receives a STANDBY, and responds correctly with a STANDING_BY. Each motor then receives GO_2PC. The motors then start moving.

- 2) At this point, there is no further network communication until such time as a CANCEL or COMPLETE or an ABORT.
- 3) Assume that one motor now, in the course of moving, causes its MIDI-OUT to be unplugged. If it then discovers that it can no longer move, and ABORTS, there is no way to see this, and therefore the second motor will continue to move making for the bad situation.
- 4) Alternatively, assume that either of the motors, in the course of moving, loses its MIDI_IN. Again, failure of the of the other motor, and a need to stop moving to avoid danger is not satisfied.

I do not see where the standard addresses these scenarios. If it does and I am just being blind, Tell me what I missed.

On the presumption that I am correct, then you have several options.

- 1) "Dr. It hurts when I do this", "Don't do that".
Dress the cables properly and avoid the problem.
- 2) Provide network health monitoring as part of the standard.
I know of no way to test network integrity short of using network bandwidth to do it. However, safety is more important in my mind than technical difficulty.

One way would be to implement short periodic polling messages where the controller would poll all devices for health, and be able to issue CANCELs appropriately.

Another way would be to have group polling for subsets of the network where several devices would be defined as belonging to a group. The controller would request that everyone in a group respond, and upon failure of any member of the group to respond, would CANCEL the group as a set. (Gee this sounds like canceling a cue).

Controlled devices would have to have the mindset that they expect to be polled, and if they are not polled, they cancel automatically. This is sort of like a network watchdog timer.

It might be possible, and more palatable, to use midi time code as the network verification means of controller to controlled device.

Then controlled devices could be expected to respond periodically to the time code as the verification means of controlled device to controller. One could further stipulate that controlled devices respond at various offsets in time relative to timestamps. Thus each controlled device would be told to respond to every fifth, or one hundredth midi time code event. In this way, network bandwidth could be fixed trading

loop time against network size for any given implementation.

A third, and final option would be to avoid this type of control altogether under the presumption that such control is beyond the scope of 2PC. This might be the only way to solve the problem without imposing excessive demands on the designed or less critical apparatus controllers.

So that's my beef Charlie. Did I miss something in the spec, or does the spec miss something in application ?

Hedley

Hedley,

I won't claim I didn't miss something the first time around, but I do think that, although your scenario is quite plausible and accurate, it is unlikely that one would want to rely on MSC_2PC for such a critical application. All of the proposed fixes you have presented are also accurate and possible but we have to remember that MSC has been created not to replace people and systems that are essential for proper operation and safety in all situations - especially ones such as you have outlined.

MSC is essentially an improvement on the stage manager's headset system: allowing faster and more accurate electronic communication between a central control location and sub-control stations while all is OK. In the event of faults, just as is done currently, there are people monitoring the situation and who are responsible for stopping unsafe conditions using any means within their ability. In the case of complex interdependent rigging operational requirements such as in your example, I would expect most theatres to use a dedicated rigging control system which has such interlocks and safety capabilities built in that can detect these types of error conditions and stop all motors at the moment such faults occur. Certainly they would be advised to do so. If they do not, then people must carefully monitor the action of the motors and stop them the moment errors are detected.

In summary, I would not recommend such a scenario to be controlled using the current MSC_2PC proposal since it is not intended to provide continuous network verification as you rightly suggest is the only way to do this properly. Furthermore, if we were to upgrade the 2PC spec as you suggest, I fear we may run the risk of being accused of trying to do everything including local safety and monitoring through MSC. We have been saying all along that this is not what we are trying to do with this and we do not believe this is the time to change. More robust networks would inherently provide a much better vehicle for such an application if and when it ever will be (needs to be?) done. These networks will automatically provide data error checking and perform automatic responses to messages as well as even constant network health monitoring -

all transparently, without involving show programmers.

Well, that's it for now - let's see what others have to say....

- Charlie.

PS - Just to clarify - the function of MSC in the case of a rigging control system being used is not eliminated: MSC commands can and will still be sent from the show control system being operated by the stage manager to the rigging system. Of course, these will be relatively simple commands such as "GO cue 23" because all safety elements will be handled by the rigging control system and its human operator.

- C ./

---[0294]--- (pref = [0293], nref = [0295])

[0295] (115 lines) CRichmond.USITT 05/19/92 2327.2 mst Tue midi
Subject: Re: new eyes
In

Here's a reply from Hedley to my last comments:

24 (49 lines in body):
Delivery-Date: 17 May 1992 11:39 mdt
Delivery-By: Network_Server.Daemon)
Date: Sunday, 17 May 1992 10:43 mdt
From: hedley at CBMVAX.CBM.COMMODORE.COM (Hedley Davis)
Subject: Re: Proposal
To: CRichmond at UNCAMULT, bitnet!uncamult.hedley at PRINCETON.EDU

There are two things I wish to discuss. Technical Issues, and motivational issues.

Technical:

After I sent you my last set of comments outlining a dangerous scenario, I did come across the MIDI real time message for network integrity checking. That should serve well for checking the channel to the controlled device. All you need is an analogous back channel message.

It seems to me a small sysex from every controlled device when prompted by the real time message would not overly kill your network. Perhaps only critical units require this level of error checking. Certainly, since the problem of dealing with coordinated actions falls on the responsibility of the CONTROLLER, it would probably only be programmed to deal with loss of communications events on critical channels. Therefore, only critical channels would need to be responded to.

I therefore suggest that there is sufficient bandwidth to

perform network integrity checking on a moderate number of critical devices even when embedded in a large network of non critical devices.

Motivational Issues:

- 1) Safety pretty much demands human watchers. Agreed. We agree. Honest.
- 2) Humans are unreliable. They are easily distracted, and often overloaded with other things to do. Stage management is less carefully defined than commercial aircraft management, and look how often they can screw up.
- 3) There is nothing wrong with redundant systems. In fact, this is a damn good thing where safety is involved.

We agree you should continue to say that 2PC is not for life threatening situations. For legal reasons if no other. I also understand why you cannot make 2PC all things to all men. Its just that this thing seems to be within the direct domain of what you are trying to do.

I don't want to piss you off. If you want to drop it, OK. I'm just offering a (perhaps unqualified) perspective. These are certainly the things I have worried about in the past when designing things for this application.

Hedley

And here's my response:

Hedley,

What is the MIDI Real-Time message you are identifying as being for network integrity checking? (there are a few potential ones to my recollection, but I haven't looked thoroughly for one that does this specifically...) Can you provide a proposed example of both forward and back channel messages? I suspect you are quite right that having only the critical units responding on a regular basis would not affect network latency. The controller definitely is the final arbiter of all these conditions, as you point out.

On to motivation - 1) Its great you're so agreeable; we are too. 2) Humans can be unreliable. Agreed. We agree. Honest. I don't like to think about air traffic controllers screwing up, but this may be a good analogy to our situation. Honest. 3) True. The only time redundant systems can be problematic is when they don't agree - at which time we're all glad they are there. (except when a computer disagrees with a stage manager, the person often wins...)

In many ways 2PC is very much centre to the domain of what we are doing and suspect it will be used in this way even sooner than we imagine. But your appraisal that we cannot legally accept the responsibility for such actions is also correct. We must, however, make the system as safe as it can possibly be (but cannot control the conditions in which people will use it.)

I personally support your proposed additions but want to make it clear as usual that this comprises a 'redundant safety' approach rather than a replacement for primary safety personnel.

Thanks again,
Charlie

And again, here in the MIDI Forum - comments?

- C
---[0295]--- (pref = [0294], nref = [0296])

[0296] (1 line) CRichmond.USITT 05/22/92 2354.3 mst Fri midi
Subject: Re: new eyes
The next transaction is Hedley's latest response -
---[0296]--- (pref = [0295], nref = [0299])

[0297] (30 lines) CRichmond.USITT 05/22/92 2355.0 mst Fri midi
Subject: Active Sensing
Date: Friday, 22 May 1992 09:13 mdt
From: hedley at CBMVAX.CBM.COMMODORE.COM (Hedley Davis)
To: CRichmond at UNCAMULT, bitnet!uncamult.hedley at PRINCETON.EDU
Redistributed-Date: 23 May 1992 00:55 mdt
Redistributed-From: Charlie <CRichmond>
Redistributed-To: {forum >udd>USITT>Clinton>Forums>midi}

Regarding the comment about network integrity checking, there is a standard MIDI system real time message. It is a single byte (\$FE), and is called ~Active Sensing". The details are on page 37 of the MIDI 1.0 detailed specification (Doc Version: 4.1.1 Feb 1990).

Basically, its optional for both receivers and transmitters. A transmitter may send this message when there is no other network activity. Receivers operate in one of two modes.

- 1) No active sense bytes yet received.
(Do nothing, active sensing not being used).
- 2) Some active sense bytes have been received.
(If there is no network traffic for 300 ms, assume communications is lost).

Active sense is nice because it is only used when the network is idle. Its ideal for the checking the controller to controlled devices link. Unfortunately, because there are no ID codes involved, it cannot be used for the back channel of communications (at least with the merger box implementation you outline) because the controller would not be able to discern from which controlled device the messages are coming from.

Hedley

---[0297]---

[0298] (77 lines) CRichmond.USITT 06/03/92 0025.4 mst Wed midi

Subject: Production Arts Letter

The following is the content of a letter which was recently sent by Production Arts Lighting Inc to the members of the ASTC and others in the theatre industry. I view this as a very positive statement and feel that it speaks for itself:

PRODUCTION ARTS LIGHTING INC.

MAY 20, 1992

Dear Colleague,

Last year, the International MIDI Association approved the MIDI Show Control standard, which is a method of using MIDI to interconnect various controllers in the show environment. The MIDI Show Control (MSC) standard was designed by and for the live entertainment industry; it is a reality today primarily due to the efforts of a USITT committee headed by Charlie Richmond, owner of Richmond Sound Design. We at Production Arts Lighting feel that MSC is an important step forward for our industry, and would like to tell you more about it.

MIDI (Musical Instrument Digital Interface) itself arrived in 1983 after a group of musical-instrument manufacturers got together to standardize the interface between keyboards and synthesizers. The resulting ability to connect synthesizers from many manufacturers opened up a vast array of possibilities; this has resulted in a tremendous increase in the range of tools available to composers and performers. This benefit from standardization is similar to that experienced by our own industry after the advent of DMX512 for console-to-dimmer connections.

To date, the use of MIDI in show control has been somewhat cumbersome. For instance, the use of a MIDI "Program Change" command to initiate a lighting cue, or a "Note On" command to activate a submaster requires a significant amount of translation on the part of the end user. Although many live entertainment control manufacturers have implemented MIDI ports in their gear, up until now they have decided on an individual basis what MIDI commands will access the features of their controllers. This has created considerable difficulty in interconnecting controllers using MIDI.

The advent of MIDI Show Control now makes controller interconnection a reality. MSC is a series of standardized MIDI commands for show applications, and includes such theatrical commands as "Go Cue 'n'", "Stop", "Resume", and a myriad of others. MSC can be used to connect any type of MSC-aware device in the show environment--lighting, sound, projections, etc. If enough manufacturers implement MSC, it may soon be possible to include everything from lighting consoles to CD players on the same MIDI network, and have all the systems communicate with each other, or be run by a master controller.

Just as SMPTE Time Code is being accepted by our industry as the standard for absolute, time-based applications, we feel that MSC is the best candidate for relative, event-based applications. MSC exists today, is economical and easy to implement, simple to use, and extremely

powerful. MSC and SMPTE can easily be used together to create very powerful hybrid systems; in fact, using MIDI Time Code, it is very easy to send SMPTE down a MIDI line.

Production Arts recently conducted an informal survey of major lighting manufacturers on their plans for MSC, and the results were encouraging. Most manufacturers are aware of the standard, and may have plans to implement it. If all goes well, we may see some lighting consoles which can be controlled with MSC as soon as LDI in November.

We urge you to ask manufacturers about their plans to implement MSC, and hope that you will specify MSC-aware products on your future projects. If you have any questions, comments or suggestions, or if you want more information about MSC or other show-control topics, please feel free to contact us.

Sincerely,

Steve Terry
Executive Vice President

John Huntington
Show Control Systems Manager
---[0298]---

[0299] (92 lines) Weber.USITT 06/14/92 1817.0 mst Sun midi

Subject: Re: new eyes

I have read transactions 291, 294, and 295, basically all the meaty discussions between Hedley Davis and Charlie Richmond. Charlie, I'm sorry you had to do all the talking here. But, I think you did an excellent job of it. The only thing I can think of to add is an elaboration on (or maybe it's a slightly different cut at) the two-motors problem.

MSC 2PC assumes a distributed intelligence network. That is, MSC 2PC assumes either:

1. The controlled device is smart. In particular, the controlled device is smart enough to handle all mechanically related aspects of the cue when it is told to GO.

or

2. The controlled device is dumb, but the thing being controlled is dumb too. This is actually an exception case, provided for really simple controlled devices, like cue lights.

The two-motors problem proposed by Mr. Davis falls under the first option above. This means that MSC 2PC assumes that it is not controlling two motors at opposite ends of a truss. Rather, MSC 2PC assumes that it is controlling an intelligent processor system that knows how to operate those two motors on the truss.

Since I know about at least two intelligent motor control systems that can do this (and could probably find the names of three or four without trying very hard), I think this is a reasonable design center for the MSC 2PC protocol. There are several other good reasons for designing MSC 2PC on this basis.

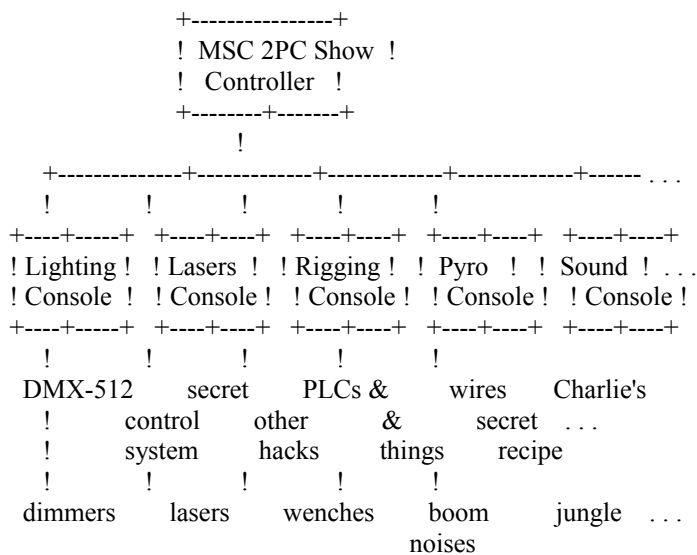
Different controlled systems require different controlling methodologies. Specific differences in methodologies include physical data interconnect properties required, such as shielding, bandwidth, latency, bidirectional vs. unidirectional, and data encoding. Another major difference would be control protocol specifics. In some cases, protocol standards already exist. For example, DMX-512 is almost universally the control protocol for lighting dimmers.

I doubt that anyone would suggest that MSC 2PC replace DMX-512 as the control methodology for individual dimmers in a cue. Personally, I think that the same argument applies to individual motors at opposite ends of a truss.

Also, the control electronics for lasers are different from those for motors, are different from those for pyrotechnics, . . . Furthermore, companies and individuals specializing in the safe usage of each of these technologies are already making appropriate control systems for each of them. Each system incorporates the feedback, redundancy, safety checking, and other features that these experts believe correct. I cannot presume to know better than these experts (with decades of accumulated experience) how to safely operate these systems.

So, the purpose of MSC 2PC is coordination of diverse presentation technologies. The coordination has two aspects. First, there is the coordinated initiation of cues. Second, there is the uniform reporting of problems from the failing subsystem to the other participating subsystems. These two problems are complex enough. I don't want to add weight to the MSC 2PC document by trying to do more.

So, here is my picture of a distributed intelligence MSC 2PC presentation system. Each box is a computer, capable of storing lots of cues and doing lots of computations per second. Each box contains specially designed software that knows how to talk MSC 2PC to the controller and how safely to manage the hardware it controls.



Hopefully, this description adds some clarity to the purpose and boundaries of MSC 2PC. Hopefully, I haven't been too pedantic. If you agree with this picture but think some words need to be added to the proposal, please give me a hint about the gist of what you think is needed. Then, I will

try to enhance the proposal.

Ralph:

---[0299]--- (pref = [0296])

[0300] (38 lines) Cotten.USITT 06/24/92 2323.5 mst Wed midi

Subject: hello

Sorry I haven't been around for a while, i've been locked in my office getting our new MSC compatible stuff ready to go. I wish I'd been in on the big discussion, but most of my comments were made by somebody or another. I've been familiar with active sensing for years, but never really considered how active sensing interacts with MIDI merging boxes. Unless there is a defacto standard of some sort that solves our problem, I think we should talk about it. Is active sensing just another message which a merger should faithfully pass along untouched? That seems useless for integrity checking; one good apple can spoil the bunch. Since Active Sensing is for integrity checking of a single line between two points, should each input of a merger box deal with each MIDI source independantly, and retransmit a new active sensing byte, only if all inputs are considered active? Since active sensing may not be sent if there is other MIDI data on the line, should the merger stop sending all data if just one input is considered inactive?

All these possibilities have some logic behind them (for the sake of discussion). Do we need "MSC_ACTIVE_SENSEn" or some such. Hmmmmmm. I'll stop there for now, i think y'all get the point.

For the Database: New Lightwave products with MSC:

All controllers are being replace with these hip new boxes with the hardware for MIDI, 0-10v, TTL, PCMCIA (JEIDA) memory cards, RS-4223285, and a nice big 40 * 2 LCD display for all kinds of user programmable options. Intellabeam controllers will ship first, then Emulator, Dataflash, Trackspot, and ColorPro. Don't hold me to the exact order. We also have a new, very simple (but quite cool, of course) 12 channel touch control dimming board.

All these device use just the basic minimal MSC, as that's all that seem to apply. Each controller knows all about it's own product, so GO_C is 99% of all that's really needed (for now at least). All of our non-microprocessor controlled products can work with ours, or any other dimming board, so really, our whole product line can be "controlled" by one MIDI cable. Hurrah!

/es .s .es /

---[0300]--- (nref = [0301])

[0301] (25 lines) CRichmond.USITT 06/25/92 0202.1 mst Thu midi

Subject: Re: hello

Hooray! With this kind of report you are allowed to stay away for weeks at a time! This sounds great and I will pass it along to Ken Bell, who is right now working with a big dealer of yours in Las Vegas who wants to put together a huge high profile show with lots of the se products all controlled by a single MSC computer (who gave him that idea...?) and he is working hard at the concept. Now the only question is - will the stuff be out by the time he needs it???

Shall we have him call you----

As far as active sensing goes - my understanding is that this is one of the least used and even less standardized functions in the MIDI(world. For reliable communications, we definitely need an MSC equivalent- and this has been made part of the MSC 2.0 proposal, by putting in the acknowledgments required for each confirmed command. This is also along the lines of what Hedley Davis was advocating and I believe we are responding appropriately by considering an on-going message type (such as "Cue in Progress") as an option to be sent on a regular basis.

Discussion of these matters is in previous transactions over the last 2 months, so feel free to peruse....

This is all good stuff and we hope to see your new stuff soon!!

- Charlie

---[0301]--- (pref = [0300], nref = [0302])

[0302] (29 lines) Cotten.USITT 06/25/92 2052.8 mst Thu midi

Subject: Re: hello

IntellabeaController will be shipping next week or so, rev 1.0 is in final testing, and is already being used on the Iron Maiden tour, and some big show in london. Emulator is so similar, it should only be about a month later, I hope. Dataflash is getting a major controller facelift to be much more similar in concept to Ibeam and Emulator, ColorPro should be relately easy, since Ibeam is just an overgrown ColorPro Controller to begin with. THE touch dimmer should be out within roughly a mon (Another Engineer is doing that, so i'm not completely sure). I don't build or sell or market, so all I can give are estimates!!!!!!

We've been sending out these two little programs, called LCC (Lightwave Control Center) to act as "show control" while working on the MIDI products, this way I've been "training" everyone on the concept of show control for about a year now. (two programs, one for mac, and one for PC) These programs are little more than cuelists which can transmit through the serial po to our controllers.

Active sensing- MSC style: perhaps this should be designed to operate at a fairly low level, directly between a transmitter and receiver. That is, the message are added and trapped just as they are transmitted and received, that way, the actual higher level control system doesn't deal with the message directly, . The lower level just tells the higher level if is still in contact with the transmitter or not. If the receiver doesnt receive a "I'm devi and I'm still here" message for some amount of time, it tells the higher level controll system that communication with devi has been lost. Operating at this lower level, there would be no interaction with specific commands to worry about, just "link from devi good", or not.

---[0302]--- (pref = [0301], nref = [0303])

[0303] (4 lines) Cotten.USITT 06/25/92 2057.5 mst Thu midi

Subject: Re: hello

boy was that garbled. to clarify, wherever you see "devi" substitute device number NN.

i.e. "I'm device number NN and I'm still here"

---[0303]--- (pref = [0302], nref = [0304])

[0304] (1 line) CRichmond.USITT 06/26/92 2237.1 mst Fri midi
Subject: Re: hello
Great - what method are you using to get the PC to talk MIDI serial?
---[0304]--- (pref = [0303], nref = [0305])

[0305] (16 lines) Cotten.USITT 06/30/92 2154.1 mst Tue midi
Subject: Re: hello
Let me clarify: the 2 programs use regular 9600 baud serial, and a "protocol" just sufficient for {"go cue nn" and "set master dim to nn" and a couple of other less significant commands. Since our old controller had no facility for accepting midi signals, this intermediate step was necessary. For most of the testing of msc on the new controllers, i've been using a MusicQuest MQx-32m, (i think that number is correct?) on the PC and an Opcode Studio 3 on the MAC. As I have no MSC software of any kind, i just manually sending bytes in the msc format. It's a little tedious, but with things like Hypercard, its not so bad, but i wouldn't want to do a show that way. on the other hand, it'll do in a pinch. The Command Cue + in the Stardust is currently sending that RS-232 9600 baud one of a kind protocol to our controllers, but within the month, I'm supposed to make sure they switch over to the new controllers using MSC. I have no idea who'll re-enter the cue's as MSC. Maybe I'll finally get to see the show. I keep missing it. I hope i answered your question somewhere in there.
---[0305]--- (pref = [0304], nref = [0306])

[0306] (11 lines) CRichmond.USITT 07/01/92 0008.5 mst Wed midi
Subject: Re: hello
Yes, the question was answered to my satisfaction, although others may be a bit puzzled. This serial stuff is not of champions! As far as the Stardust goes, the guy we recommend in LV is George Kindler, formerly of Acromedia and now of his own company, Thoughtful Designs. Also, Christian Hugenaar of Thomas Gregor Audio (who put in the original system and did most of the programming) could certainly help, although he normally is in LA.

Good luck and if all else fails - do it yourself as usual!

- Charlie
---[0306]--- (pref = [0305])

[0307] (38 lines) Weber.USITT 07/01/92 1947.4 mst Wed midi
Subject: "Cue in Progress" and "I'm still here"
Adding a "Cue in Progress" message to MSC V2.0 opens a nasty can of worms. There must be some previously agreed to timing that the controlled device must observe when sending the "Cue in Progress" message. This might be a fixed value, set by the standard. But, that might not be flexible enough for all MSC devices. So now, there's got to be a message exchange that negotiates the timeout value for "Cue in Progress" messages.

One way or the other, the controller has got to know the timeout value. Otherwise, it doesn't know when a missing "Cue in Progress" message really means something.

"I'm still here" messages implemented at a layer below the controller-to- controlled-device communications assumes a layering of software that I have not seen present in MIDI. Also, this approach

changes MIDI MSC from what the pros call a connectionless protocol to a connection-based protocol. The underlying layer that implements the "I'm still here" is the connection layer. There is a world of difference between connectionless and connection-based protocols. This is not something you change lightly.

The current MSC V2.0 assumption is that the controlled device will complete a cue in the time specified in the STANDING_BY message. Admittedly, there is a certain amount of trust implied between the controller and controlled device. Also, the level of trust increases proportionally to the length of time required to complete a cue.

To me the questions are:

"How paranoid must we be?"

and

"How much pain are we willing to endure to support that paranoia?"

Ralph:
---[0307]---

[0308] (33 lines) AEkvall.USITT 07/06/92 1004.5 mst Mon midi
Subject: Misc

Hi there,

I have been following the discussions but as I think most things have been said, I have not participated.

However, later messages indicate something that I myself has been longing for. That is some kind of simple MSC test software that can be used for trying out an MSC implementation in a product.

Is there such a software available out there? Charlie, would it be possible for you to provide a small subset of your software just for testing? Unfortunately (in this case) your software runs on an Amiga. PC or Mac programs would be better.

It should be possible to enter simple event lists and execute them.

I think that the availability of such a test routine should speed up the inclusion of MSC in existing products.

About the discussion of Active Sense and Network checking messages: There is a problem with merging (which we must use a lot) and that Active sense is an almost totally unused part of MIDI. This has been said on the forum and I totally agree.

Also, there is a discussion about the two motors that need to be tied together in some way. MSC is made for Controller to Controller communication. It is not meant to replace high-speed, high-security networks between a rigging system and its motors. It should communicate high-level commands and leave the rest for the specialized controller products that is made for each type of control system.

Anders

---[0308]--- (nref = [0309])

[0309] (7 lines) CRichmond.USITT 07/07/92 0043.8 mst Tue midi

Subject: Re: Misc

I am disappointed, Anders, that you declined to be on the panel at PLASA, but I do understand your concerns.. although it would have been quite nice.

Please stand by for Mac software - coming soon, we hope.

- Charlie

---[0309]--- (pref = [0308])

[0310] (27 lines) CRichmond.USITT 07/28/92 2129.4 mst Tue midi

Subject: New proposal

A new data transmission protocol has been proposed by a major manufacturer of show control systems. This protocol is similar to DMX 512 and has similar applications, but is more robust and fault tolerant as well as having a number of other advantages which are outlined in the document. Conversion between DMX and SDX (for Serial Data Transmission) is relatively easy and they are currently using SDX for their own inter-device control.

I told them that I would upload this protocol to this forum for perusal by those interested. I'm not sure whether we want to consider this as a formal suggestion to begin a new standard, but we should be open to this since it looks as though it will gain some ground. It is low cost and could be an upgrade for DMX. We will be making converters from SDX to RSD-32 so that their show control systems can address our audio hardware for things such as amusement park rides and other non-live show applications where our software is overkill.

It's pretty long, so this message is a warning that the next one is a killer. (the powers that be may want to put the entire proposal into the database, but since they are not physically present in our world at the moment this will have to do!)

- Charlie

---[0310]---

[0311] (320 lines) CRichmond.USITT 07/28/92 2132.3 mst Tue midi

Subject: SDX Protocol

Richard Gray writes in reference to the following specification:

"This code is designed for many show control applications. It is not intended to replace MIDI Show Control, but it is far superior to DMX 512. I have tried to build in simplicity - reliability - flexibility, etc. without ignoring communications theory. I hope you approve!"

"My plan is to get comments and suggestions from as many people as I can and then release this for general use after all reactions have been considered."

SYNCHRONOUS SHOW DATA TRANSMISSION PROTOCOL (SDX)

(C) R. A. GRAY, INC. 1992
ALL RIGHTS RESERVED WORLDWIDE

R.A. Gray, Incorporated
9181 Chesapeake Drive
San Diego, CA 92123
(619) 560-4162

Draft 1 June 1992

(SUMMARY)

The Synchronous Serial Data Protocol is designed for all communications associated with show data transmission. This protocol satisfies a broad range of external requirements including reliability, error checking, noise immunity, variable framing rates, synchronization, and time skews. The protocol is also fully compatible with all film, video, and recording time base standards.

The code is based on a superset of the Manchester code which is the standard used for all forms of digital communications. This code allows for synchronous detection of data, and offers reliability orders of magnitude greater than currently used asynchronous codes. The signals are symmetrical and have no D.C. component. Therefore isolation is easily accomplished with passive components such as transformers. The data protocol incorporates this code into data frames which are ideally suited to show communications and also retain the best features of asynchronous codes.

GENERAL DESCRIPTION

The fundamental concept of this protocol is a modal system where all show elements are updated each data frame. While this may not be necessary at all times, it is usually necessary at some time during a performance. This protocol minimizes the bandwidth for the worst case data pattern when everything changes. Furthermore, data reliability is enhanced orders of magnitude since, if a data block is corrupted, it will be repeated in full. Reliability is also a main concern. In critical show control environments errors can be catastrophic. This protocol optimizes signal to noise ratios, and minimizes bandwidth requirements to three spectral points. The code is blocked and designed to be double buffered. The whole data block is only acted upon when it is complete, and validated. This eliminates time skewed movement associated with incomplete messages.

The basic carrier frequency of 192,000 cycles per second is harmonically related to all film, video, and time code standards. Since the carrier is constantly being transmitted several benefits can be achieved. Foremost is reliability. All receivers can stay locked, and synchronously decode the signal. Noise rejection is extremely good, and reliability can be excellent. Another major benefit is all system

devices can have the same time base, assuring accuracy of even the most demanding servo systems. The carrier frequency can be used at the receiving end to provide a time base for speed control and servo position integration. The 192,000 baud data rate is also easily converted to a DMX512 data stream for compatibility with existing systems.

The inclusion of a parity bit allows the receiver to easily determine if the signal is corrupted. While more sophisticated error checking and correcting schemes are possible, parity is adequate, and simple to generate and validate. In blocked data, error detection approaches that of more complex formats.

Data is blocked into one of three standards, with 512 bytes per message the preferred format. Although a smaller block size of 256 base has been reserved for applications requiring fast service, and a 1024 byte block has also been specified for systems where the update rates are slow and data paths are scarce. Each data block contains systems information. Each interface and system element can be slaved to the master through performance time, as well as show data.

This protocol supports systems that can be built with smart elements that are programmed to respond at a particular time. A synchronous detection scheme would allow these elements to start even if the particular data block containing the time is corrupted. The protocol allows for a completely variable framing rate. The program can be played back or recorded at any rate, even frame by frame. The constantly running 192,000 cycle time base can be used to assure all servos still act in a uniform manner without concern for what the framing rate is.

The use of a special character for synchronization is a very important aspect of this code. Most Manchester codes use special "sync patterns". While this does somewhat reduce the bandwidth requirement, it forces data codes to be more complex. It also requires the synchronization period to be longer. The introduction of the F/2 code as "sync pattern" eliminates all these undesirable features without introducing drawbacks. The sync character is a cycle at a frequency of F/2 using four bit times. This code is easy to generate and detect. The transmitter is required to send this character whenever show data is not present, assuring all receivers stay locked.

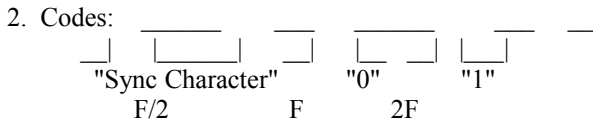
The signals used in this protocol are optimized for bandwidth and signal to noise ratios. Since the code is symmetrical and has no direct current components it can easily be isolated with linear passive components such as transformers and capacitors. In extreme cases less desirable non-linear components such as opto-isolators could be used. The code is low pass filtered at 250KHz. Unlike asynchronous codes, fast edges are not required or desirable. Therefore, line ringing is not nearly as big a concern as with other codes. This allows a high relative signal voltage to be used without excessive power requirements. Resistive and capacitive loading of wiring is less of a problem. Therefore, in most cases inexpensive standard twisted pair or audio wiring is adequate. Also the reduced harmonic content of the signal minimizes shielding requirements.

Every attempt has been made to make the code compatible with both discrete component decoding, and more sophisticated micro-processor

based playback systems. The code is downward compatible with DMX512 using a small handful of discrete parts yet maintains the sophistication necessary for the future. Features have been included to make the code as useful for off- line playback as live theatre applications. SDX should gain wide acceptance.

DATA DESCRIPTION

1. Bit Rate: A. 192,000 bits/second +/- 5%



3. Data Blocking:

A. Byte Definition

| "P" | "7" | "6" | "5" | "4" | "3" | "2" | "1" | "0" |
Least Significant Bit First

B. Odd Parity

C. Bytes/Frame:: 256; 512; 1024 (only 512 byte blocking currently being used, other blocking for future use)

D. Followed by any number of sync characters

4. Framing Rates:

A. Any rate that meets the above conditions.

B. Framing does not need to be at an even spacing or periodic.

C. Maximum Framing Rates:

Rate <21 frames per second, for 1024 byte blocking
Rate <42 frames per second, for 512 byte blocking
Rate <84 frames per second, for 256 byte blocking

5. Data frame blocking chart:

DATA FRAME BLOCKING (DECIMAL)

(1024)	(512)	(256)	
0	0	0	First user byte
1011	499	249	Last user byte
1012	500		System Byte 12 reserved for future use
1013	501		System Byte 11 reserved for future use
1014	502		System Byte 10 reserved for future use
1015	503		System Byte 9 reserved for future use
1016	504		System Byte 8 reserved for future use
1017	505		System Byte 7 reserved for future use
1018	506	250	System Byte 6

1019 507 251 System Byte 5
 1020 508 252 System Byte 4
 1021 509 253 System Byte 3
 1022 510 254 System Byte 2
 1023 511 255 System Byte 1

- Notes:
1. For 1024 and 512 byte blocking there are 12 system bytes and for 256 byte blocking there are 6 system bytes.
 2. If code is being converted to DMX-512, first user byte is used as null byte.
 3. System bytes 6-12 are reserved for future use.

Only 512 Byte Block is Currently Implemented.

6. A. Run time system byte definitions

7 6 5 4 3 2 1 0 Up to 83 frames/sec

SB6 F F F F f f f f "Frames"
 SB5 X S S S s s s s "Seconds"
 SB4 X M M M m m m m "Minutes"
 SB3 T T H H h h h h "Hours"
 SB2 X X X X X X X X "Status"
 SB1 S X U U U U U U System Info

X = Reserved for Future Use

B. SB6-SB3

1. Performance time in hours, minutes, seconds, frames.

C. SB2

1. System status, such things as setup record, playback, etc. to be defined later.
2. SB3 bits 7 and 8 (TT)
 - 00 Time stopped
 - 01 Time advancing at a periodic rate
 - 10 Time advancing at an aperiodic rate
 - 11 Time retarding at an aperiodic rate

These signals are very useful for setting servo damping constants

D. SB1

1. S: Packet type

0 = Normal run time data packet

1 = Special packet, or command packet (in which case all other definitions of all other data elements may take on any special meaning, such as down loading of data to an interface during setup)

2. UUUUUU:

User bytes. These signals are available in real time at the sending node and are not prerecorded. They could be used for such functions as wind scaling, etc.

7. No command or special data packets have been defined at this time. However, a protocol has been proposed.
 - A. The current packet marks where the next packet is going...as well as verifying the destination of the current packet.
 - B. Special packets would contain message headings containing packet counts and packet numbers.
8. Data Transmitters must as a minimum:
 - A. Provide the parity generation
 - B. Always send synchronizing frames whenever a data block is not being sent
 - C. System bytes are optional
9. Data Receivers should as a minimum:
 - A. Verify parity for each byte
 - B. Verify proper blocking
 - C. Validate the command bit is not set
 - D. Be double buffered and only respond if all the above criteria are met
 - E. All respond at end of each data block, avoiding time skews

ELECTRICAL DESCRIPTION

1. Carrier Frequency
 - A. 192KHz +/- 5%
 - B. Frequency driven from system time base such as: video sync; SMPTE; film framing rate; crystal clock
2. Transmitter
 - A. $5 < \text{P-P voltage} < 10$
 - B. Output impedance < 20 ohms
 - C. Rise times limited to > 500 nS or low pass filtered at 250KHz
 - D. Passive balanced using a transformer
3. Receiver
 - A. Responds to signals of > 1 volt P-P
 - B. Provides a minimum of .25 volts P-P hysteresis
 - C. Provide a nominal 2000 ohms resistive load
 - D. Passive balanced using a transformer
4. Termination

- A. There will be one or more terminations consisting of a 100 ohm resistor in series with a .1uF capacitor at the end of lines > 250 ft.
- B. No more than 50 receivers or 1500 ft. of Belden 8451, or equivalent, will be used without a repeater amplifier (Much longer wire runs could be attained with low capacitance, or non-shielded wiring).
- C. A shielded distribution cable is not a requirement. The signals are of sufficient amplitude to not require shielding to guard against data reception errors. However, shielding may be required to prevent EMI in some applications.
- D. The shield is tied at the transmitter only. The shield may pass through receivers, if loop through connectors are provided. If static build-up is a concern the center tap of receiver input transformers may be grounded to the chassis through a separate 1 Meg ohm resistor and the shield may be tied to the chassis through a separate 1 Meg ohm resistor.

---[0311]--- (nref = [0312])

[0312] (11 lines) CRichmond.USITT 07/28/92 2138.2 mst Tue midi

Subject: Re: SDX Protocol

Sorry, SDX refers to "Synchronous Show Data Transmission"... As you can see, this spec is not yet in a fully presentable form yet, but such as it is can probably be adequately understood. I feel it would be valuable to tighten it up to eliminate any potential technical ambiguities so that it can be used for its intended purpose by a variety of manufacturers without any of the kinds of problems that DMX encountered....

Comments?

- Charlie

---[0312]--- (pref = [0311], nref = [0322])

[0313] (8 lines) AEkvall.USITT 08/04/92 1018.3 mst Tue midi

Subject: Question

Embarrassingly Unfortunately, I seem to have lost the Sysex ID assigned to our beloved standard. Can anyone please give it to me?

Charlie, what do you mean that you are exspecting a Mac MSC program to arrive? Are you involved in that? Is it a third-party program?

Andersd

---[0313]--- (nref = [0314])

[0314] (11 lines) CRichmond.USITT 08/05/92 0046.9 mst Wed midi

Subject: Re: Question

Hi, Anders - the subID number 1 is 02H (i.e. F0 7F <device number> 02 <command_format> <command> <data> F7)

Yes, we are developing a Mac version of Stage Manager, but it is a little way off yet - although we will be announcing it at PLASA. It is third party in the sense that our stuff is all sub-contracted to independent programmers, but it is our design and owned by us. Are you interested in using this version, perhaps? We have been making

arrangements for special licensing with other manufacturers.

- Charlie

---[0314]--- (pref = [0313])

[0315] (15 lines) CBlackley.USITT 08/05/92 0811.7 mst Wed midi
Subject: MIDI mechanicals

Hi Charlie. Thanks very much for the demos (Thank Ken for me) and thanks for the binder of product info. As I was reading through it I noticed that Stage Manager has been set up for control of mechanicals. We are investigating automation in a small scale for a 6 drive system for a show next season. We have a local automation company working up a design for a system but we are having some communication problems (the concept of going over the same cue sequence several times in rehearsal is unfathomable to them). I realised the Stage Manager has a much friendlier interface and the system is already in place. My question is: when it comes to mechanical controls, where does your products end and the mechanical suppliers start. ie. at the motor controller. Phrased another way. I have motors and controllers. RDS has the Amiga and software. What else do we need in between the two and does RDS have that missing link or is it another supplier.

This is turning into a semi-business transaction so I probably should have put this in as mail instead of on a forum.

---[0315]--- (nref = [0318])

[0316] (18 lines) AEkvall.USITT 08/05/92 2331.0 mst Wed midi
Subject: Charlie

Thank you for the answer.

Yes, I am intrested in the Mac version of Stage Manager.

Still, I am more urgent on a way to test my MSC implementation of our system s against a third-party product or test program. Of course, I can try it internally but there is nothing like testing it agaist someone elses code!! You do not happen to have just a small routine that could be used validate my implementation? Anyway, this should be available as a tool for evaluating implementations. This would certainly increase the speed of a wide-spread adoption to MSC. What do you think? This is a general problem with all these standards. There are no tools for validation of the standard. Maybe, you know about something that I can use. I have implemented the code and I would like to release it within a month or so in a major software upgrade. As you can see I am on a tight time schedule.

Anders

---[0316]--- (nref = [0317])

[0317] (35 lines) CRichmond.USITT 08/05/92 2348.6 mst Wed midi
Subject: Re: Charlie

Goddard Design Co of New York mentioned to us at USITT that they were going to make an MSC 'tester' similar to the DMX512 unit they currently make. Perhaps you could contact them to see where they are with this and if it can send and receive or whatever you need. In the meantime, we use two ways to test our stuff: a. if it works with other stuff correctly, and b. scope the waveforms and use recognized MIDI monitoring software utilities for whatever computer platform is desired to determine whether the correct message codes are being sent/received. This is really just about it.

As far as your time schedule, the best I can suggest as a definite way to do it is buy or rent an Amiga (all you need is an A500 with 1Meg - about \$600US), an Amiga MIDI interface - our deluxe one is \$140 - and license Stage Manager for \$400US. This software fully supports all MSC commands by both sending and capturing any valid messages. It takes about 2 seconds to validate in plain English (sorry, we haven't got a Swedish editor done yet...) the contents of any captured MSC message.

The process goes like this: click on the Capture button, send the message you want evaluated, click on Capture again to turn it off, click on the Add button to put the message into the list, double-click on the message you wish to look at in detail and you are put immediately into the appropriate MSC editing screen with all the cue data, cue list, cue path, etc. filled into the blanks as captured. The reason for the double entry process is that you might send a number of different messages in a single capture process - adding the captured data to the message list automatically parses all captured messages and lists them separately so you can then select which one of those captured you want to look at.

Anyway, I'm sorry to say we don't have the Mac version ready yet, but if I had a dollar for everyone who's asked - well, I wouldn't be rich, but if I had \$400 for everyone - well, I still wouldn't be rich....

- Charlie
---[0317]--- (pref = [0316])

[0318] (80 lines) CRichmond.USITT 08/06/92 0012.8 mst Thu midi
Subject: Re: MIDI mechanicals
Perhaps you are right about this being a private transaction, but I will briefly explain what can be done for this interface.

We make a card which receives the MSC message to run a mechanical device to a certain position at a certain speed with a certain acceleration and deceleration factor. This message is created on the Stage Manager system using the MSC editor and can be done quite simply. When the card receives the message, it calculates where the mechanical device it is controlling is now and where it needs to go then commands the motor controller which direction to go and sends it speed commands. The speed outputs from the card constantly and dynamically change while the mechanical device is in motion in order that the speed of the device follow the acceleration, deceleration and maximum speed defined by the MSC message.

What you need to provide to make this all happen is the following:

1. An absolute position sensor that puts out up to 10 bits of binary data in TTL voltages that represents the actual position of the mechanical device at all times _or_ a quadrature position encoder that puts out two bits of leading/trailing signals that represent the direction and speed of travel. Scaling needs to be dealt with to a certain extent although it is really up to you since as many as 1024 position points are definable.
2. Limit switches that tell the card when the lower and upper limits have been reached. The card automatically remembers and recalibrates when it sees the limit switches close.

3. A 7 bit switch to define which of 127 devices the particular card is. This means you can talk to up to 127 cards on a single MIDI network.
4. A deadman switch that is normally closed during operation. If it opens, the motor is halted and the brakes are applied immediately. There is also an E-stop input which does the same thing when the contact is closed.
5. A few mode switches to select what you are seeing on the multi-digit display: position, target value, error number, speed, last error, etc.
6. A couple of other inputs that I don't remember at the moment. And that's it for inputs.

The outputs work like this:

1. 8 outputs control the motor speed binarily. This means that if all outputs are on, the motor should run at maximum speed, if only the LSBit is on, the motor should run at minimum speed. How you do this is up to you, but most people use solid state relays of appropriate value to drive the motor speed control unit. The simplest way to do this for the untechnical is to use the relays to simulate a varying resistance hooked up to the external speed control potentiometer connections of a standard motor speed control box.
2. 1 output controls the brakes.
3. 1 output controls the direction. Again these are normally interfaced via appropriate solid state relays.
4. Various outputs are provided to indicate correct operation, fault conditions, brakes on, etc.

This card is being used by a company in Pennsylvania who has adopted them as a standard moving scenery implementation for rock and roll acts like Cher and Michael Jackson, so it is quite bulletproof. They are, however, the exclusive licenser of this card for the next year and a half since they paid for our software development, so I will have to either refer you to them for purchasing or else contact them ourselves to see what they will sell them for. I recommend the former since you will get their full expertise in how to use them - and perhaps they will set you up with their whole little rack mount rig instead of just our card.

Their name is Tait Towers and they are in Lititz, PA but I don't have their phone number handy - if you have trouble getting it let me know.

Sorry about the length of this message, but I felt it would be useful.

- Charlie
---[0318]--- (pref = [0315])

[0319] (11 lines) CRichmond.USITT 08/06/92 0016.5 mst Thu midi
Subject: MSC_2PC_0.7
The MMA has published the latest TSBB but has only included MSC_2PC_0.6 due to some unspeakable oversight - I hope it isn't mine. At any rate,

this is just for purposes of informing the rest of the unwashed what we have been up to and it is not for voting on. I will point out the omission and make sure they have the latest. In the meantime, we may get some feedback from the MIDI world and we may come up some changes yet ourselves. Ralph???

I will, of course, keep you all posted....

- Charlie
---[0319]---

[0320] (41 lines) CRichmond.USITT 08/09/92 0014.7 mst Sun midi
Subject: Multimedia
The Interactive Multimedia Association has extended the following invitation to all interested parties:

You, or an appropriate person from your organization are invited to join the newly chartered MIDI Special Interest Group within the IMA Compatibility Project. Please pass this information along to anyone who may have an interest - you don't have to be an IMA member to participate. Please contact Peter Farrett of IBM in Austin, Texas. He can be reached at (512) 838-9819 or on Internet [send_mail, sdm] farrett-at ausvm1.vnet.ibm.com

The IMA has formed a MIDI Special Interest Group (SIG) as a forum for those interested in MIDI in the Multimedia computing environment. The charter of this SIG is:

- (1) Provide an open forum for the discussion of MIDI topics as they relate to multimedia computing
- (2) Act as the focal point for technical exchange between the IMA and the MIDI Manufacturers Association (MMA)
- (3) Represent the interests of MIDI users and vendors within the IMA and assure that MIDI is adequately considered in any relevant Compatibility Project activities.

Dr. Peter Farrett of IBM's Austin, Texas development laboratory has volunteered his time and effort to organize the SIG and act as chair until it is ready to elect its own chairperson. Peter has development responsibility for music and speech products for the IBM Personal Systems group and hold degrees on both music and computer science.

I have already asked to join this SIG and one of my primary objectives will be to make MSC become a standard part of the IMA communications protocols. I have written two 'white papers' on this subject in the past and they have been reprinted in the current MMA Technical Standards Board Bulletin 18. I am also writing an update on this same subject to be published in Lighting and Sound International for distribution at PLASA. The subject is, of course, why MSC should be part of a multimedia communication standard.

- Charlie
---[0320]---

[0321] (4 lines) CRichmond.USITT 08/16/92 1146.0 mst Sun midi
Subject: OOT (away)

I will be in the UK until the 14th of September. Phone number is +1-44/81-399-8351 in case of emergencies. Thanks.

- Charlie
---[0321]---

[0322] (24 lines) Weber.USITT 09/27/92 1828.7 mst Sun midi
Subject: Re: SDX Protocol
I'm not much on electronics. So, I may overstep my expertise here.
But...

The Manchester encoding sounds like a good idea. DEC used that in its Computer Interconnect (CI), which I developed software for many years ago. I've never heard anything bad about it.

We did have one very nasty problem with messages being sent out of sequence. Because we had used only a single-bit sequence number the problem was virtually undetectable. Care should be taken to avoid similar problems in SDX.

If one views the timestamp in frames SB6 -- SB3 as a sequence number, then clearly there is no problem with detecting out of sequence frames. However, I wonder if using the timestamp as a sequence number is consistent with the overall SDX design. This could be like trying to hammer a nail with a pair of needle-nose pliers.

Also, this design appears to be focused on uni-directional message delivery. This gives it the same problems as MIDI has with respect to a bi-directional protocol, such as the two-phase commit extensions to MIDI Show Control.

Ralph:
---[0322]--- (pref = [0312])

[0323] (4 lines) CRichmond.USITT 01/09/93 2020.1 mst Sat midi
Subject: Moving Day
Yes, it's true - this forum will also continue on in new 'discuss' guise on Callboard II. The latest report on MSC_2PC will be there soon!

- Charlie
---[0323]---

forum (print): There is no next transaction.