Pyro Network Control Protocol

Author: Leon S. Searl, January 2010
Contributors: PyroUniverse.com Users (mutski, pikoko, stuntborg)
Version 0.02

Introduction

The purpose of the open and publicly available Pyro Network Control Protocol is to provide a shared protocol enabling amateur builders and designers of Control Modules, Firing Modules and Bridge Modules to create modules that can be easily interconnected. This capability allows gatherings of amateur pyro-technicians for an organized shoot to aggregate their equipment into a much larger coordinated set-up than they can assemble individually. If manufactures also use the protocol then pyro equipment consumers have a wider selection of equipment to choose from to add to equipment that they already have that supports the protocol.

The Pyro Network Control Protocol is intended as an open and publicly available protocol to remotely fire pyrotechnic igniters using digital communications. The igniters are attached to 'smart' Firing Modules on a shared digital packet based communication network controlled by a Control Module with a possible backup Control Module. Portions of the network may use different communication mediums. The differing communications mediums are connected together using Bridge Modules. The collection of Firing Modules, Bridge Modules and Control Module(s) is a Pyro Network. The protocol that runs over the communications medium is the Pyro Network Control Protocol.

The protocol only applies to communication between a Controller Module and the Bridge and/or Firing Modules. A connection between a PC and a Controller Modules is not covered by this protocol.

The communication medium of the Pyro Network may be wired and/or wireless. Wired networks or segments of a network may use RS485, Ethernet, Power Line Communication (PLC) or other standard or custom medium. Wireless (RF) network segments may be Zigbee or other wireless RF medium. The underlying network medium does not matter as long as the Pyro Network Control Protocol can be applied over the top of the medium.

The Pyro Network is a broadcast network meaning that the Control Module, all Bridge Modules and all Firing Modules see all messages that are placed into the network. In order for the Pyro Network to be implemented on a medium, the medium must be capable of broadcasting messages. In other words all nodes on the network must be able to 'see' all messages on the network. Addressing is also used in the protocol to allow individual or groups of Firing Modules to be addressed.

Other features:

- Up to 255 groups of Firing Modules
- Simultaneous firing of cues within a module group.
- Simultaneous firing of any set of cues on a Firing Module
- A Firing Module does not have to implement the full protocol. The builder may pick and choose which capabilities to implement.
- 10 mS cue timing resolution
- Firing Modules may store cue firing schedule fired by an interruptable and resettable time code.

- Firing Module capabilities are discoverable by the Control Module.
- Firing Modules may report cue continuity and/or cue resistance.
- The protocol is designed to handle small private shoots, club shoots and large city shoots.

Overview

A Control Module is the head node (master) of the Pyro Network. Its primary purpose is to send firing commands to the Firing Modules. The Control Module may be fully self contained with a firing command sequence stored in it or its fire commands may come from manually actuated switches. A PC may send firing commands to the Control Module which then routes the commands to the Firing Modules. The Control Module may also be a PC that connects directly to the Pyro Network. An example of a PC that is a Control Module is a laptop with a WiFi card that sends commands to a WiFi - EIA-485 Bridge Module.

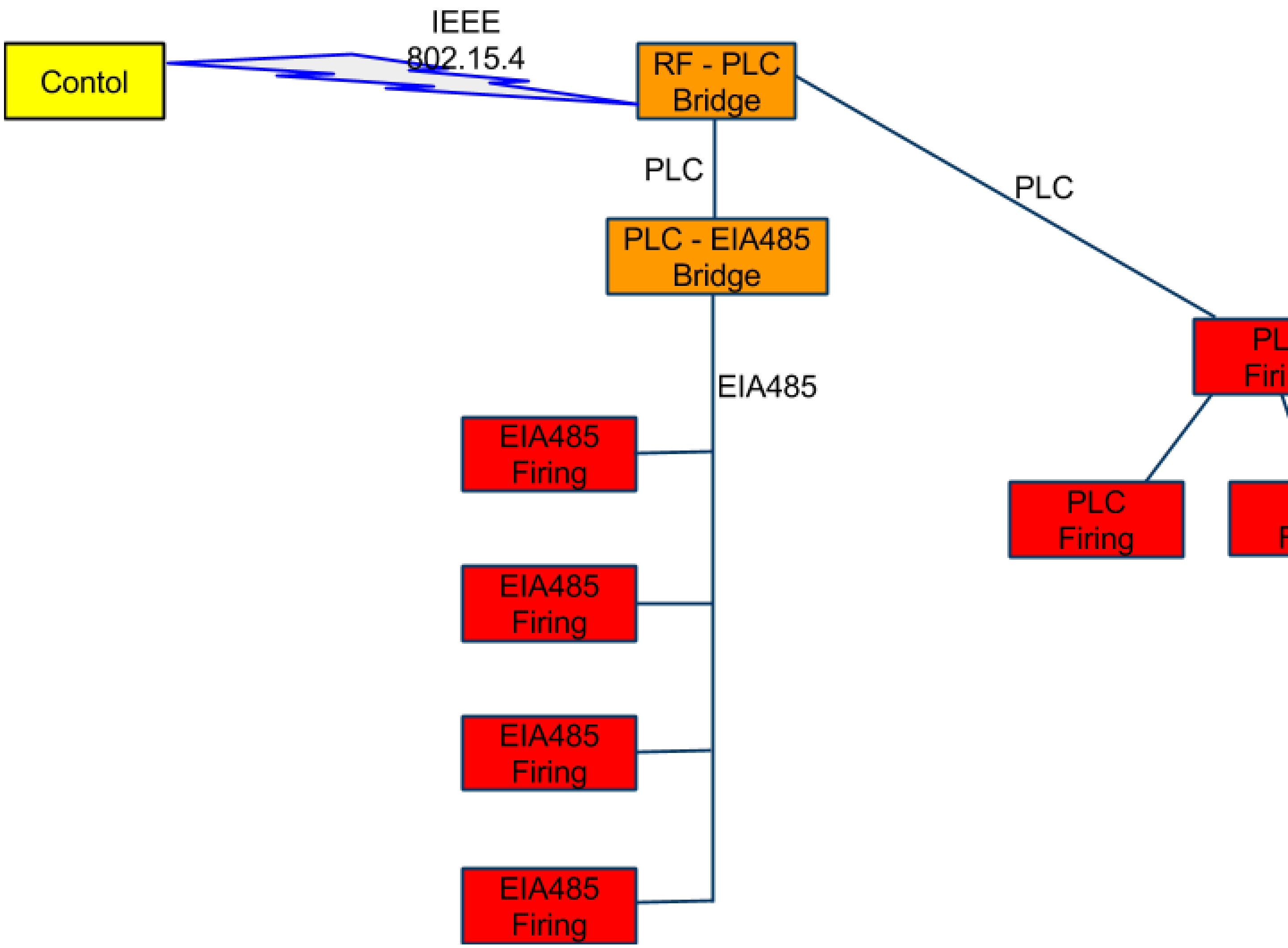
A Firing Module is electronic hardware that receives a command, over the Pyro Network, to fire an electrically ignited device (igniter) that in turn ignites a pyrotechnic display device or leader to a pyrotechnic display device (comet, mine, shell, cake, etc). Each electrical connection on a Firing Module for connecting igniters is call a cue. Firing Modules may have up to 255 cues or as few as 1 cue. Each Firing Module must have a network address so that firing commands can be directed to the desired Firing Module(s). Firing modules may range in complexity from simple to very complex. The simplest Firing Module implements the only required command (**Fire Cue**) and may only have a few cues. A complex Firing Module may have any or all of the following:

- Report its capabilities
- Fire any set of its cues simultaneously
- Report resistance of the cues (continuity can be determined from the resistance).
- Report input power voltage
- Fire a sequence of cues from one fire command.

In order for a command to reach the desired Firing Module the module must have an address. There are two Firing Module addressing mechanisms. One address is a long identifier unique to each Firing Module and may not be changed except by a firmware change. The other address is short, usually configurable and my be shared by more than one Firing Module. Firing modules my use either one or both of the address mechanisms.

The figure below shows a display set-up that could occur at a pyro club shoot where 3 different members have brought their equipment to use together. The Control Module uses a Zigbee medium to send firing commands to a Zigbee RF to Power Line Communication (PLC) Bridge module. The bridge pulls the Pyro Digitial Control Protocol packets from the Zigbee data packets and puts the PDCP protocol packets into a PLC packet and sends it on PLC line. PLC modules (and the PLC bridge outputs) can be connected into any configuration (daisy chain, star) except a loop. From the RF-PLC Bridge the PLC packets travel to both a PLC-EIA485 Bridge Module and a PLC Firing Module. The PLC-EIA485 Bridge pulls the Pyro Network Control Protocol data from the PLC packet and places it into EIA485 packets and forwards the packets onto the EIA485 line. There are 4 EIA485 Firing Modules that must be daisy chained (if no EIA485 hubs are used). The first PLC Firing Module receives the packet from the PLC Bridge and passes the packets on to any other PLC Firing Module or Bridge

Module connected to it (just like a electrical power strip distributes power). Any Firing Module that receives a command may respond to the command in the reverse path back to the Control Module. All modules in the network see every command and response. Firing Modules only respond to commands containing their address. Firing Modules ignore responses from other Firing Modules.



Notation

01011b - Any combination of '1' and '0' or 'X' followed by 'b' indicates a binary representation of a number. The 'X' represents either a '1' or a '0'.

7CD8h - Any combination of '0' through '9' and 'A' through 'F' or 'X' followed by 'h' is a hexidecimal representation of a number. The 'X' represents any digit value.

UByte - unsigned byte value

5bit UInt - 5 bit unsigned integer value.

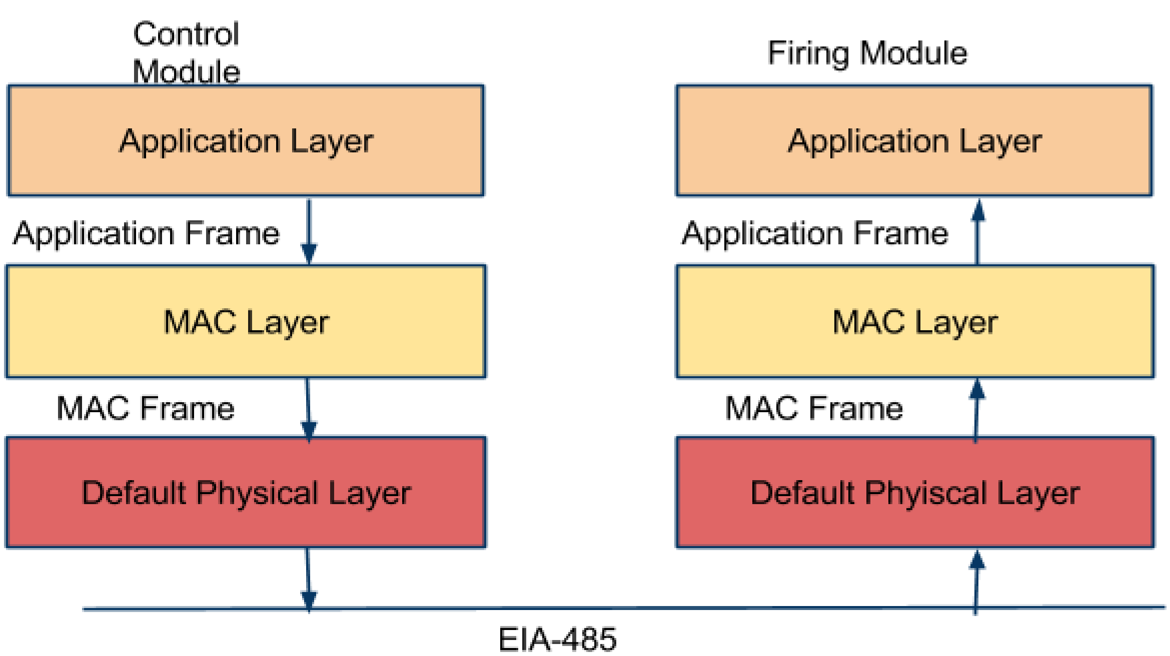
Protocol Layers

The protocol is divided into layers to segregate the handling of different aspects of the protocol. There may be several versions of the lower layers with a version for each communication medium.

Each layer is composed of a frame format and software/firmware for processing the contents of the frame it receives or transmits.

The protocol covers 3 layers:

- Application
- MAC
- Physical



The Application layer is concerned with the format of the pyro commands themselves. It is the top level with the responsibility of carrying out the commands. Commands to be transmitted are passed down to the MAC layer. Commands that are received are passed from the MAC layer to the Application Layer.

The MAC Layer is concerned with getting the commands to the correct Firing Module. This layer is build on top of the Physical Layer. Data that is transmitted is passed from this layer to the Physical Layer. Data that is received is passed from the Physical Payer to this layer. This protocol provides a default MAC layer but many other communication protocols can be used in place of the default MAC layer (example, TCP/UP).

The Physical Layer concerns the media that is used to transmit and receive the packets and it verifies the integrity of the packets. This is the lowest and most basic layer. The Physical Layer may also be a full network stack with an application layer, link/MAC layer and physical layer. In this case the pre-existing network stack is considered the Pyro Network Command Protocol's Physical Layer.

The Physical Medium is usually an existing standard but builders may create their own custom Physical Medium.

Physical Medium

The Physical Medium may be a simple electrical or radio frequency specification or a full blown protocol with its own MAC/Link and Application layers.

In its basic form the Physical Medium is the electrical specification or radio frequency specification for transporting information (bits of data) between nodes on a network.

EIA485 Serial Interface

EIA485 is strictly an electrical interface specification. It has no PHY, MAC or Application layers.

The PNCP specifies the following for EIA485 Physical Layer compatibility between Firing Modules of different builders.

- Baud Rate for connections using a bridge to a PC being used as the Control Module - 9600, 115200
- Connector Pins
  - D+
  - D-
  - GND

A Firing Module may use just the 9600 baud rate or both baud rates. When both baud rates are supported by the Firing Module, the baud rate to use must be selected by a front panel switch on the Firing Module.

CAN Bus

CAN bus is also the Physical Layer.

- Speed - TBD

Ethernet/UDP

Ethernet/UDP is also the Physical Layer.

Broadcast Address -TBD
Broadcast Port - TBD

Power Line Communication(PLC)

Power Line Communication is using the 2 line DC power line delivered power from the Control Module to Firing Modules to also send commands.

Impedence above TBS kHz: >= TBS ohms.

Modulation: TBS

TBD

RF - IEEE 802.15.4

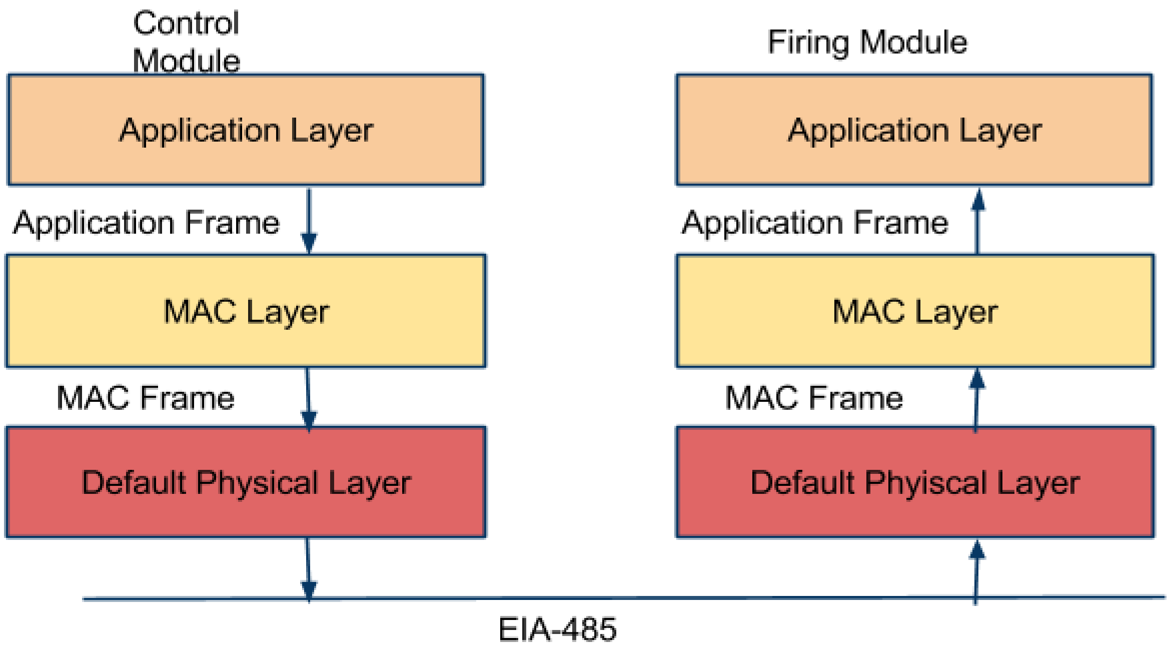
TBD

Physical Layer

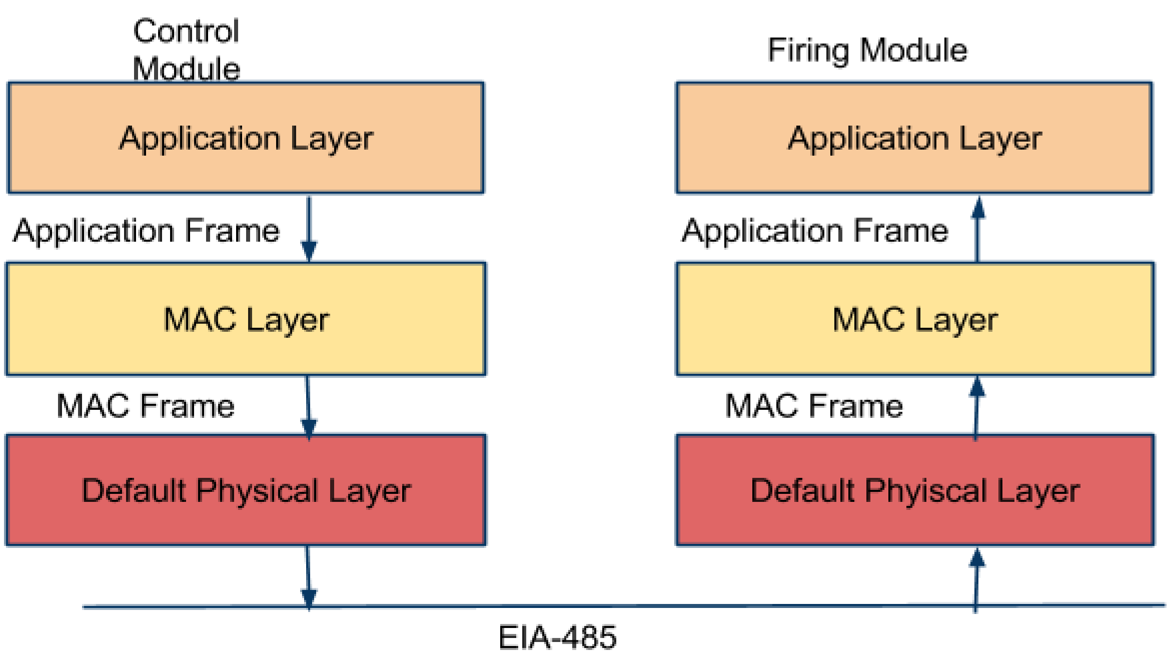
The Physical Layer is the Physical Frame and the firmware/software for processing the Physical Frame according to the Physical Layer protocol. The Physical Layer is also what controls the hardware used to transmit and receive data from other modules.

If the under laying communication mechanism is a full blown packet communication protocol then the Physical, MAC and possibly Application Layers of the under laying communication mechanism replaces the Default Physical Layer of the Pyro Network Control Protocol.

As an example the Ethernet/ Internet /User Datagram Protocol can be used as the PNCP Physical Layer. The protocol stack would look as follows for a IP/UDP Physical Layer:



The Default Physical Layer is used when there is no protocol defined to run on the chosen Physical Medium. As an example the Default Physical Layer is used when the Physical Medium is EIA-485. The protocol stack looks as follows:



Default Physical Layer

The Default Physical Frame is for use by Physical Layers over Physical Mediums that do not have their own frame format or protocol. Examples are:

- EIA-485
- Custom Power Line Communication
- Custom RF

The goal of the Default Physical Frame is to provide a frame that can be at least delineated and have a high confidence of determining the presence of bit errors. Frame delineation is important since some of the network mediums allow packet collision where a packet from one Firing Module can interfere with part of packet from another module. The delineation provides a mechanism for determining when a packet starts and stops. The Start of Frame Delimiter and Frame Length provide delineation for the frame.

It is important to detect when bit errors occur in a packet. Bit errors are very possible in both wired and wireless networks at pyro club events due to Electro-Magnetic Interference (EMI) from nearby high power transmissions lines, broadcast AM, FM and other RF antenna towers, and thunder storms within 30 miles. Undetected bits errors could cause a command to be interpreted as a Fire Cue command when it is some other command and thus cause an unexpected firing of a cue. To detect bit errors with high
confidence a 16 bit CRC is used.

Default Physical Layer Frame Format

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2 thru MFL+1</td><td colspan="8">MFL+2</td><td colspan="7">MFL+3</td></tr>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td></tr>
<tr><td>Label</td><td colspan="8">SOF</td><td colspan="5">MFLI</td><td colspan="3">RSV</td><td colspan="8">MF</td><td colspan="15">CRC</td></tr>
<tr><td>Value</td><td colspan="8">10101011b</td><td colspan="5">0 thru 31</td><td colspan="3">000b</td><td colspan="8"></td><td colspan="8">MSb</td><td colspan="7">LSb</td></tr>
</table>

Label	Type	Value	Units	Description
SOF		10101011b		Start of Frame Symbol
MFLI	5bit UInt		Index	MAC Frame Length Index, Index into a table containing MAC Frame Length values.
RSV				Reserved Flag bits. May be used to indicator type of CRC used in the future.
MF	Data			MAC Frame bytes.
CRC				CRC type is TBD

The value used for Start of Frame Delimiter could occur naturally in any other part of the frame so escape sequences is needed to prevent false Start frame detection. The escape byte is 0xFF. When preparing to transmit a frame, any Start of Frame Delimiter or Escape byte is replace with two bytes as follows:

- Start of Frame Delimiter (0x01)-> 0xFF 0xFE
- 0xFF -> 0xFF 0xFF

Start of Frame Delimiter
A byte that indicates the start of a frame of data. This byte is needed when packets may be transmitted back to back. It allows the receiver to determine where a packet starts when there is no empty space between the packets.

MAC Frame Length Index - An index to the total number of bytes in the MAC Structure. Some receivers may be programmed to only allocate enough buffer space to hold a packet. The *Frame Length* tells the receiver how much space to allocate for the MAC Frame.

The MAC Frame Length is an index into a table that contains the actual length number.
NOTE: The length numbers start at 3 since that is the smallest frame size possible

Index	MAC Frame Length
0	3
1	4
2	5
3	6
4	7
5	8
6	9
7	10
8	11
9	12
10	13
11	14
12	15
13	16
14	20
15	24
16	28
17	32
18	40
19	48
20	56
21	64
22	80
23	96
24	112
25	128
26	160
27	192

28	224
29	256
30	reserved
31	reserved

To convert from a number of bytes into the length index:

```
char length;
char mfli;
if(length < 17) {
  mfli = length - 3;
}
else if (length < 33){
  mfli = (length / 4) + 9
}
else if (length < 65) {
  mfli = (length / 8) + 13
}
else if (length < 129) {
  mfli = (length / 16) + 75
}
else if (length < 257) {
  mfli = (length / 32) + 165
}
else {
  // This is an error.
}
```

Note: The Frame Length is the length of the MAC Frame before escape sequences are added.

MAC Frame
The *MAC Frame* is a variable length frame containing control and address information plus the **Application Frame**.

CRC

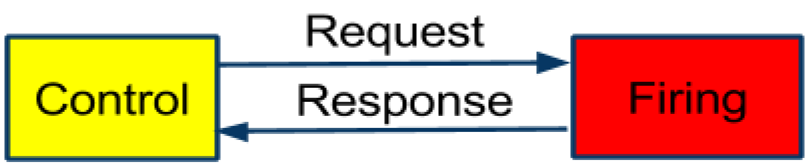
CRC of the *MAC Frame Length* and MAC structure up to but not including the two bytes of the CRC. The CRC type used is TBD.

NOTE: The CRC is made before the escape sequences are added.

MAC Layer

The MAC layer is a piece of software and a frame formate between the Physical Layer and the Application Layer. It fills in the MAC Frame with information from the Application Layer for transmitted frames and passes the MAC frame to the Physical Layer. For received frames it extracts information from MAC Frame provided by the Physical Layer and passes the extracted information to the Application Layer.

In the Pyro Network all Firing Modules have addresses. The Control Module does not have an address since there can be only one Control Module in the network at any time. The first two bits of the frame determines if a message is going to a Firing Module or a Control Module. Th module of message flow is request-response. The Control Module is the only module on the network that sends requests. Firing Modules are the only modules on the network then send responses. Firing Modules never initiate communication.



Not all requests from a Control Module will cause a Firing Module to respond. It is up to the Application Level to determine if a Request causes a Response.

MAC Frame Format

There are 4 types of MAC Frame Formats. Two formats are used to send messages from a Control Module to Firing Modules. One message type is used to send responses from Firing Modules to the Control Module. The fourth message type is reserved for future use.

The first two bits of a MAC Frame determine the type of the MAC Frame format and the direction of the frame.

Value	**Description**
00b	Broadcast Frame Format - Sent from Control Module to Firing Modules. For a Broadcast Frame all Firing modules are expected to extract the Application Frame and pass it to the Application Layer.
01b	Addressed Frame Format - Sent from Control Module to Firing Modules. An Addressed Frame is a frame that is dropped if the address in the frame does not match an address of the Firing Module.
10b	Response Frame Format - Sent from Firing Modules to Control Module.
11b	Reserved for future use.

Broadcast MAC Frame

Byte	0			1	2 + i
Bit	7 6	5 4 3 2	1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Label	MT	RSV	AT	NSLOT	APPF
Value	00b	000b	xxb	0 thru 255	

Broadcast MAC Frame Description

Label	Type	Value	Units	Description
MT		00b		Message Type: Broadcast from Control Module
RSV		000b		Reserved
AT		xxb		Address Type is don't care.
NSLOT	UByte			Number of response slots
APPF				Application Frame Command

The Broadcast MAC Frame is used to send a command to all Firing Modules in a display setup at the same time. Typically the only commands that are broadcast are the **Report Capabilities** command (because the Firing Controller does not yet know what modules are in the network) and the **Time** command.

NSLOT - The number of slots field is used to tell the Firing Modules the number of time slots that they have to use to put their command response into if the command requires a response. With the exception of the **Fire Cue** and **Time** command, commands sent in Broadcast Msg Type mode will have many modules trying to send response messages back to the Control Module. If all modules responded at the same time then the packets would collide on the medium and the Control Module would get CRC errors (if it was able to understand the packet header at all). Collision is not a problem if the Physical Layer is something like Ethernet/UDP using Ethernet Switches which can avoid most collisions. In the case of a collision proof medium the Firing Modules should ignore the *Number of Slots* field and send responses immediately.

For the cases where the Physical Layer is something like EIA485 or a custom medium where packet collisions are possible then see **MAC Broadcast Mst Type Response**.

APPF - The Application Frame Command message bytes.

Group Addressed MAC Frame

Byte	0			1	2 + i
Bit	7 6	5 4 3 2	1 0	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
Label	MT	RSV	AT	GADD	APPF
Value	01b	000b	00b	0 thru 255	

Group Addressed MAC Frame Description

Label	Type	Value	Units	Description
MT		01b		Message Type: Addressed
RSV		000b		Reserved

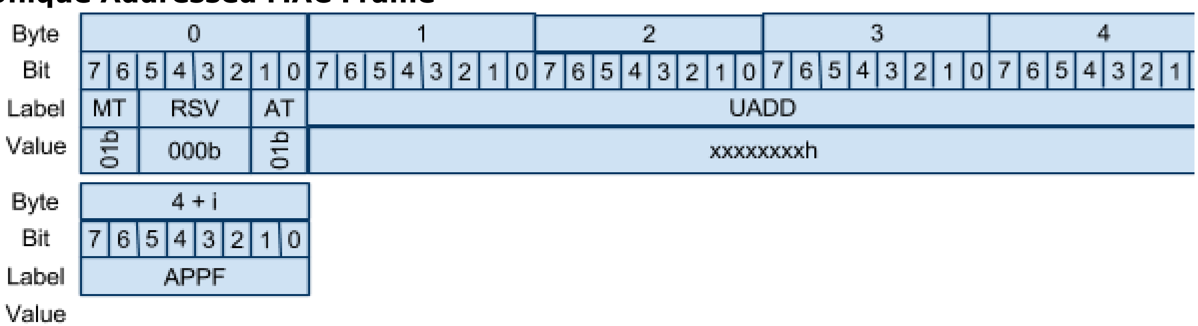
AT		00b		Group Address Type indicator
GADD	UByte	XXh		Group Address
APPF				Application Frame Command

The Group Address MAC Frame is more versatile then the Unique Address MAC Frame. The address is shorter making for a smaller frame and thus faster command rate. The Group Address is also typically set on each Firing Module by a front panel switch making setup similar than with a fixed address like the Unique Address. A module can go into any firing location in a display setup and its address can be set by the module's front panel address switch to match the show script. The same Group Address can be set on more than one module (thus the name Group Address). All modules with the same Group Address process a Group Addressed MAC Frame simultaneously if the frame contains their Group Address.

GADD - The byte value of the Group Address.

APPF - The Application Frame Command message bytes.

Unique Addressed MAC Frame



Unique Addressed MAC Frame Description

Label	Type	Value	Units	Description
MT		01b		Message Type: Addressed
RSV		000b		Reserved
AT		01b		Unique Address Type indicator
UADD	UInt	XXXXXXXXh		Unique Address
APPF				Application Frame Command

The Unique Addressed MAC Frame is used to ensure that a command from the Control Module is received and processed by only one specific Firing Module in the Pyro Network. Each Firing Module produced by commercial vendors and private builders must have a 4 byte address that is highly likely to be unique among the whole population of Firing Modules in existence. This protocol does not have a formal organization to ensure that Unique Addresses assigned to Firing Modules are truly unique such as the IEEE for assigning vendor IDs for Ethernet MAC Address.

UADD - The Unique Address of the Firing Module.

Typically a shoot operator will not have to be aware of the Unique Address of the Firing Modules in a set-up with the exception of more than one module using the same Group Address.

The Unique Address must be retained by the Firing Module while power is NOT being applied. Since the protocol is a mildly complex digital packated protocol most Firing Modules will contain a micro-controller to process the protocol packets. Most micro-controllers contain Flash memory or One-Time-Write memory that can be used to store the Unique Address.

The Unique Address of each Firing Module should be written or labelled on the Firing Module in a visible location.

The following proposal is to create a mechanism that ensure a high likely-hood of Unique Addresses being truly unique. There are two forms of Unique Addresses. One form is for use by commercial vendors producing Firing Modules. The second form of Unique Address is for private builders producing Firing Modules for their own use. Each form of Unique Address has an ID field to identify a specific vendor or builder and a unit ID field to identify a specific Firing Module of the vendor or builder.

If in the future the proposed method for the two Unique Address formats proves to be insufficient there are two more currently unused values for the Address Type in the MAC header that can be used for additional (longer) Unique Address formats.

APPF - The Application Frame Command message bytes.

Commercial Vendor Unique Address

Byte	0								1								2								3							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	T	VID											UID																			
Value	0	0 thru 2047											0 thru 1,048,575																			

The first bit is 0 to identify the address as a Commercial Vendor type of Unique Address.

VID - The VID field is a Vendor ID field. It identifies the commercial vendor that produced the Firing Module. With 11 bits used for the VID there can be 2048 different VIDs. With no formal organization to control issuing VIDs there must be an informal process to have vendors pick their VID. A collaboration web site like a Wiki or Google Group should be setup so that the members (vendors) of the site can publish the Vendor ID that they have chosen. There should be only one VID per vendor unless the vendor has used up all of their UIDs. The VID list web page must also have the vendor's name, link to the vendor's web site and name of a contact person and the contact person's contact information (email, phone number).

No vendor should pick a VID in the expectation that some day they will produce a Pryo Network compatible product. A vendor should only pick a VID once a product is in production.

UID - The UID field is a number that identifies a specific Firing Module produced by the Commercial Vendor. This is very much like a serial number. The vendor is free to do with this field as he/she wishes. A recommended practice is to use the first 2 or 3 bits to identify a Firing Module model number and the remaining bits as a serial number.

Private Builder Unique Address

Byte	0								1								2								3							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	T	BID																							UID							
Value	1	0 thru 8388607																							0 thru 255							

The first bit '1' identifies the address as a Private Builders Unique Address.

BID - The BID field is a Builder's ID. Each private builders has their own unique BID. With 24 bits of BID there are 8,388,608 possible BID's. Since there are so many possible BIDs the mechanism for a private builder to choose their BID is different than for VIDs. A fingerprint algorithm is used on a text string that identifies the builder. The fingerprint algorithm produces the BID. If the builder is a Pyro Universe member then the string would be the Pyro Univers Username of the builder. A string of only a first name and last name should not be used since there is too high a chance of duplicate strings used to generate the BID and thus two different builders having the same BID.

The recommended algorithm for generating a BID is Rabin's Fingerprint algorithm with a irreducible polynomial p(x) of degree 23 = 96AE17h. See **Appendix A** for more information on generating the BID.

If the builder is confident that the modules the he/she builds will never be used in conjunction with another builder's modules then the builder may chose any random BID.

UID - The UID field is a number that identifies a specific Firing Module produced by the private builder. This is very much like a serial number. The builder is free to do with this field as he/she wishes. A recommended practices is to use 0 for the first module built and then increment the number for each successive module.

Response MAC Frame

Byte	0								1								2								3								4						
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1
Label	MT		RSV				AT		UADD																														
Value	10b		000b				01b		xxxxxxxxh																														

Byte	4 + i							
Bit	7	6	5	4	3	2	1	0
Label	APPF							
Value								

Response MAC Frame Description

Label	Type	Value	Units	Description
MT		10b		Messge Type: Response
RSV		000b		Reserved
AT		01b		Address Type of a response is always a Unique Address.

UADD		XXXXXXXXh		Unique Address
APPF				Application Frame Response

UADD - The address in the response is always a Unique Address so that controller knows which module the message came from. If a Group Address where to be used and there was more than one Firing Module then the Control Module would not know which Firing Module a response came from. That could cause problems when reporting a continuity and there was a problem with a cue. The operator would not know which of the Firing Modules in the group had the cue problem.

APPF - The Application Frame Response message bytes.

MAC Broadcast Msg Type Response TBR

The algorithm presented in this section for responses to Broadcast commands only applies to Physical Medium where it is possible for packets to collide.

When the Firing Module is to send a response to a broadcast command the Firing Module calculates a random number (R1) from 0 to N-1 where N = Number of Time Slots. The Firing Modules waits for the R1 slot time and then transmits the reply. With this algorithm it is possible for 2 or more Firing Modules to transmit in the same time slot and thus have their packets collide. It is the responsibility of the Control Module to determine that there was a collision in a time slot and act to try to have the Firing Modules send replies again. The method to do this is to send the Application Layer Ignore Next command to all Firing Modules that successfully responded (did not have collisions in their time slot and had a good CRC check) and then resend the broadcast command that had packet collisions. Only those Firing Modules that did not receive a Ignore Next command will respond to the broadcast command. The reduced number of responding Firing Modules increases the probability of successful responses from the Firing Modules. The Control Module should repeat this processes until there are no Firing Modules responding because all of the Firing Modules have been heard from at some point in the procedure.

Time slot duration is the length of time it takes to transmit the reply (based on the frame length and baud rate) plus 0.5 mS

The time from when the last bit of the broadcast command has been received until time slot 0 begins is 0.5 mS.

Typically the **Broadcast Command MAC Frame** is only used with the preceding slot response procedure for the **Report Capability Application Command** so that the Control Module can find the addresses of all of the Firing Modules on the Pyro Network. Once the Control Module knows the addresses of the Firing Modules on the Pyro Network the **Broadcast Command MAC Frame** is no longer needed except for the **Time Application Command** which does not have responses from the Firing Modules.

Application Layer

This level of the protocol is common to all types of Pyro Networks that adhere to the protocol regardless of MAC Layer and Physical Layer. It contains the actual commands and responses between the Control Module and the Firing Modules.

Not all Firing Modules will be built to the same level of complexity. The protocol is designed to be simple, complex or anywhere in between depending on the complexity of the Firing Module hardware.

There is only one command message that all Firing Modules must implement: the **Fire Cue Now Command Message**. This one command is sufficient to execute a pyrotechnic show but the display operator will have to enter the Firing Module's addresses into the Control Module manually.

If the builder wishes to have any other capability of the Firing Module accessible to the Control Module then the Report Capability Command Message must be implemented by the Firing Module. This gives the Firing Module information about all of the capabilities of the Firing Modules.

Note that this layer does not deal with addressing except for the Assign Group Address Command Message. Addressing is part of the MAC layer.

Messages from the Control Module always contain a leading command value. The command value is 2 or 4 bits for commands that are used during a display so short messages for faster message processing is important. Commands that are primarily used during set-up prior to a display are 8 bits (1 Byte) long. There may be additional succeeding bytes of the message depending on the message type.

Application Frame

Short Length Command Frames are used for commands that must be send very fast.

Short Length Command

Byte	0								1..N							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CD		PRM						PRM							

Medium Length Command Frames are used for commands where many different modules have to be addressed in a short period of time.

Medium Length Command

Byte	0								1..N							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CMD				PRM				PRM							

Long Length Command Frames are used for commands that don't fit into the short or medium length command categories.

Long Length Command

Byte	0								1..N							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CMD								PRM							

Command Summary:

Command Name	**Value**	**Cmd Size Bytes**	**Rsp Size Bytes**	**Description**
Fire Cue	00b	1	0	Fire a cue immediately.
Fire Multiple Cues	01b	1+(N +2)/8	1	Fire multiple cues at the same time.
Time	1000b	3	0	The Current Show Time
Report Capabilities	1001b	1	4	The Firing Module is to report its capabilities.
Report Cue Continuity	1010b	1	1 + (N+4) /8	Have the Firing Module report is cue continuity.
Report Cue Resistance	1011b	1	1 + N	Report the ohm resistance of each cue.
Charge Cues	1100 0000b	1	1	Charge Capacitive Discharge cues that do not charge automatically at power up.
Set Fire Pulse Witdh	1100 0001b		1	Set the Cue Firing Pulse width.
Report Input Voltage	1100 0010b	1	2	Measure the Firing Module internal battery or voltage input lines.
Cue Schedule	1100 0011b	2+4* N	1	Load a time schedule for cues to be fired into a Firing Module. N = Number of Cues to schedule.
Reserved	1100 0101b - 1111 1111b -			

Note that the leading bits of the command determine how many total bits are in the command.

- 0 - A leading zero indicates a two bit command. There can only be two of these. With such a short command ID the remainder of the first byte can be used for command parameters making the commands as short as possible. Short commands can be send faster than long commands. For low bit rate physical layers this is important.
- 10 - A leading 1 followed by a 0 indicates a 4 bit command. There can only be 4 of these.
- 110 - This indicates an 8 bit command. There can be

Response Summary:

Response Name	**Value**	**Description**
Acknowledge (Aak)	0x0	The command is being executed.
Not Acknowledge (Nak)	0x1	The command can not be executed for some reason.
Bad Parameter (BPR)	0x2	The command contained a value that the Firing Module could not handle.
Unimplemented (Ump)	0x3	The command is not implemented in the Firing Module and will be ignored.

Fire Cue Command Message (00b)

Tell a Firing Module(s) to fire one of its cues immediately. This command must be implemented by all Control Modules and Firing Modules. This is the only command that must be implemented by Pyro Network Modules.

Command Frame

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	CD		QN					
Value	00b		1 thru 63					

QN - Cue Number To Fire - The Firing Module cue to fire. Cue numbers start at 1. Most cue numbering systems on Firing Module front panels start at 1. Cue number 1 from the command matches cue 1 on the Firing Module front panel. Cue number 2 from the command is cue 2 on the Firing Module front panel. With a 6 bit size cue number there can be up to 63 cues on a Firing Module. The maximum number of cues is not 64 since cue 0 is special. If the cue number in the command is greater than the number of cues in the Firing Module then the command is ignored. If the cue has been fired before then the Firing Module should try to fire the cue again if possible.

In this protocol Firing Modules can have up to 255 cues. This command is unable to fire cue numbers above 63. The **Fire Multiple Cues** command must be used if the Firing Module has more than 63 cues.

If the cue number is 0 then the Firing Module is to fire all cues simultaneously if it has the capability. If it is not capable of firing all cues then the Firing Module does nothing.

Because of the structure of the MAC layer, if Group Addressing in the MAC Frame is used, all modules that have the same Group Address receive and execute the message at the same time and fire the same cue number.

How fast can **Fire Cue** commands be handled by a Pyro Network. Assume the following:
- EIA485 at 9600 Baud
- No Bridge Modules for the commands to pass through.
- MAC Short Addressing

With the EIA485 network medium the **Default Physical Frame** is used. The **Default Physical Fram**e has a 2 byte header and 2 byte trailer. The Group Addressing in the MAC Frame makes a MAC Frame header of 2 bytes. The **Fire Cue New** command application message is 1 byte. Total bytes for the message is 7 bytes.

With one start bit and one stop bit of EIA485 and a mark space of one bit between EIA485 symbols, there are 70 bits in the message. At 9600 Baud the message is transmitted in 7.39 mS.

Fire Cue Response

None

In order to make processing this command as simple as possible no response to the command is required.

Fire Multiple Cues Command Message (01b)

Tell a Firing Module to fire a specified set of cues. The cues are fired immediately.

Command Frame

Byte	0								1..N							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CMD		Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q n*8-1	Q n*8	Q n*8+1	Q n*8+2	Q n*8+3	Q n*8+4	Q n*8+5	Q n*8+6
Value	01 b															

Each cue of the module is represented by a single bit flag. Cue 1 is bit {0[5:5]}, Cue 2 is bit {0[4:4]} and so on.

A flag of '1' indicates a cue to be fired. A flag value of '0' indicates that the cue should not be fired.

If the Firing Module has 6 or fewer cues then only 1 byte is required for the Application Frame. For Firing Modules with more than 6 cues there are as many bytes in the Application Frame as needed to hold all of the the cue bits.

If there are more cue flags in the command than the number cues the Firing Module has the extra cues are ignored by the Firing Module.

If there are fewer cue flags in the command than the number cues the Firing Module has the cues of the Firing Module not flaged by the command are not fired.

Fire Multiple Cues Response

None - No response is sent for a Broadcast Command. In a set-up with many Firing Modules it would take too much time to receive all of the responses at the Control Module. This is the only command command that responds to an addressed command but will not send a response if the command is broadcast.

None - No response is sent if the only command implemented is the "Fire Cue" command.

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	STS				RSV			

STS=Ack - The command was accepted and execution started. The Ack must be sent as soon possible to keep the Control Module from a Timeout while waiting for the Ack.

STS=Ump - the Firing module does not support this command.

Time Command Message (1000b)

The Time Command Message tells the Firing Module the current point in the module's stored cue schedule. The time specified in the message represents the amount of time that has passed since the start of a pyrotechnic display (show). This command is normally sent using the MAC Broadcast Msg Type so that all modules receive the current show time simultaneously. Firing Modules must NOT send a response back to the Control Module for this command. When the command is received by a Firing Module, if the module has a stored **Cue Schedule**, the module firmware looks into the **Cue Schedule** to see if there is a cue or set of cues to fire at the time specified in the Time Command Message. If there are cues in the Cue Schedule to fire at the specified time then the modules fires those cues. If there are no cues in the Cue Schedule to fire at the specified cue time then the module does nothing. If the Firing Module does not have a stored **Cue Schedule** then the Time Command Message is ignored. Firing Modules are given a Cue Schedule using the **Cue Schedule** Command Message.

Note that the time specified in the message is not Time of Day and it is not necessarily real time although in normal operation is should be very close to real time from the time the show started.

The normal operational usage of this command is for the Command Module to send this command out every 10mS with the first TIME value = 0 to indicate the start of the show. Each successive Time Command Message should have the TIME value incremented by 1. The Command Module should continue sending the Time Command Messages until the show has completed.

It is perfectly valid to have the Command Module stop sending the Time Command Message for a unspecified period and then resume sending the Time Command Message. This would happen if the control panel dead-man switch was released and then at a later time engaged again.

It is valid for a Command Module to send the Time Command Message with time values that are not in order. This is useful if during a show someone tripped over a cable and unplugged some Firing Modules. A number of Time Command Messages will have been sent by the Command Module by the time the shooter could stop the show. The shooter could stop the show, reconnect the cable, reset the show time to a point before the cable had been unplugged and then restart the show. This will cause the Control Module to resend some Time Command Messages. Firing Modules will attempt to re-fire cues that have already been fired by a previous arrival of a Time Command Message with the same specified time.

Since Firing Modules do not have a response to this message it is not possible for a Control Module to know if the message was received by the Firing Modules. In order to compensate for possible dropped Time Command Messages cue to CRC errors if the currently received Time Command Message time is less than 100mS more than the last received Time Command Message time and there are unfired cues in the Firing Module's Cue Schedule between the last Time Command Message time and the current Time Command Message time then the Firing Module is to fire the unfired cues that have been skipped in the Cue Schedule since the last Time Command Message time.

What is the minimum bit rate required to be able to send the Time Command Message every 10mS. With a Application Frame size of 3 bytes, MAC Frame Header of 2 bytes (Default MAC Layer) and a PHY Frame Header plus CRC of 4 bytes (Default PHY Layer) there are 3 + 2 + 4 = 9 bytes required to send the Time Command Message. Assuming a 25% overhead for transmitting bytes over the medium (EIA-485 start and stop bits) then the minimum bit rate is (9 bytes * 8 bits * 125% / 10 mS) = 9000 bits/sec.

Application Frame Format

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1</td><td colspan="8">2</td></tr>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Label</td><td colspan="4">CMD</td><td colspan="20">MSb TIME LSb</td></tr>
<tr><td>Value</td><td colspan="4">1000b</td><td colspan="20">0 thru 1048575</td></tr>
</table>

Application Frame Content Description

Label	**Type**	**Value**	**Units**	**Description**
CMD	4bit UInt	1000b		The Command bits that indicate a Time Command Message. Leading 10b indicates 4 bit command value.
TIME	20bit UInt	All Values	10 mS	The current time since the start of the show in 10 mS resolution. The maximum value of 1048575 is 2.91 hours.

Time Response

None.

Firing Modules are not to respond to the Time Command Message. Since this command is normally sent using the MAC Broadcast Msg Type and there may be many Firing Modules in the Pyro Network, if Firing Modules were to send a response to the message it is very likely that not all of the responses would be received by the Control Module in time to send the next Time Command Message (in 10 mS). This would cause delays in sending Time Command Messages. The delays would quickly add up or cause too many Time Commands to be skipped by the Controller causing synchronization problems for choreographed displays.

Report Capabilities Command Message (1001b)

The purpose if this command is to tell the Control Module what the hardware capabilities and commands implemented by the Firing Module are. This is the second most important command for a Firing Module to implement. The **Fire Cue Now** command is the most important command to implement.

This command is almost always used in MAC Broadcast addressing to discover what Firing Modules are in the Pyro Network. The Slots field of the MAC Frame should be set to 0 when Broadcast is used to send this command.

Report Capabilities Command Frame:

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	CMD				RSV			
Value	1001b							

Report Capabilities Response

None - No response is sent if the only command implemented is the "Fire Cue" command.

Byte	0								1								2								3							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	STS				CONT	REST	VOLT	QSCH	RSV	CHGF	CHGT						PLSL				RSV				NCUE							

Response Description

Label	Type	Value	Units	Description
STS	4bit			Status - Ack
CONT	Flag			If '1' then the Firing Module implements the Report Cue Continuity Command.
REST	Flag			If '1' then the Firing Module implements the Report Cue Resistance Command.
VOLT	Flag			If '1' then the Firing Module implements the Report Input Voltage Command.
QSCH	Flag			If '1' then the Firing Module supports the **Time** and **Cue Schedule** commands.

CHGF	Flag			If '1' then the Firing Module supports that **Charge Cues** command. If 0 then either there are no charge capacitors in the Firing Module or the charge capacitors start charging when power is applied to the Firing Module.
CHGT	6bit Uint	0, 63, 1 thru 62	Seconds	Defines how long it takes the Firing Module to charge is firing capacitors. If 0 then the Firing Module does not have cue firing capacitors.
PLSL	4bit Uint			A value of 0 indicates that the cue firing pulse length can not be set. The value determines the maximum amount of time that the cue firing pulse length can be set. The time is calculated as (1mS *(2 ^ Value)). Example: value=0001b -> 1mS. Example: value=0101b -> 32mS. Example: value=1010b -> 1.024S The maximum time is 1111b -> 32.768 S.
RSV				Reserved
NCUE	Ubyte	1 thru 255		The Number of Cues on the Firing Module.

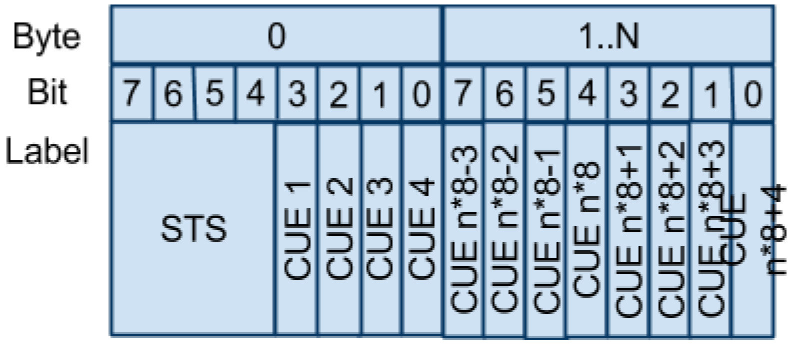
Report Cue Continuity Command Message (1010b)

The Control Module tells a Firing Module to report the continuity of each of its cues. This message is the third most important command for a Firing Module to implement after the **Fire Cue Now** and **Report Capabilities** commands.

This Firing Module applies a current limited voltage to the cue to determine if there is a measurable current through the cue.

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	CMD				RSV			
Value	1010b							

Report Cue Continuity Response



Name	Description
STS	Status - Ack, UMP. If STS == UMP then there is only the 0 Byte in the response.
CUE X	In each Cue bit '0' is NO continuity, '1' is GOOD continuity.

The Control Module will expect the number of bits in the response to be equal to the number of cues reported by the "Report Hardware Command Message" response plus enough bits to fill that last byte. Bits in the last byte that do no correspond to cues must be 0.

The bytes in the message are filled MSB first.

Report Cue Resistance Command Message (1011b)

The Control Module tells the Firing Module to report the resistance of each cue. This command can be very useful during continuity check time to determine if there are short circuits or some cues are in the wrong position.

On some igniter the insulation of the copper wires near the head is bare for up to 2 mm. During the igniter installation of the igniter into a pyro-technic device the pyrotechnician may accidently apply a twist to the igniter causing the two bare leads near the head to cross and create a short circuit. A normal 1/0 continuity test will show continuity but the igniter will not fire due to the short circuit. With a resistance measurement test the short circuit will be measured as a very low resistance. If a Firing Module has the hardware capability and implements this command the shorted cues can be located and then replaced.

The resistance measurement can also be used to verify the work of pyro-technicians that have wired up multiple igniters in series or parallel. With a known igniter resistance the number of igniters on the cue can be estimated from the measured cue resistance.

<table>
<tr><td>Byte</td><td colspan="8">0</td></tr>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Label</td><td colspan="4">CMD</td><td colspan="4">RSV</td></tr>
<tr><td>Value</td><td colspan="4">1011b</td></tr>
</table>

Report Cue Resistance Response

The response from the Report Cue Resistance Command Message is the longest of the protocol. There is one byte for each cue resistance.
The byte value is an unsigned number (N) where various ranges of the number represent various precisions of the resistance value (R).

<table>
<tr><td>Byte</td><td colspan="8">0</td><td colspan="8">1 - M</td></tr>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Label</td><td colspan="4">STS</td><td colspan="4">RSV</td><td colspan="8">RES</td></tr>
</table>

<table>
<tr><td>Name</td><td>Description</td></tr>
<tr><td>STS</td><td>Status - ACK, UMP</td></tr>
<tr><td>RSV</td><td>Reserved</td></tr>
<tr><td>RES</td><td>Resistance
<table>
<tr><th>Byte Value Range</th><th>Units</th><th>Ohm Value Rante</th><th>Ohm Calculation</th></tr>
<tr><td>0 - 99</td><td>1/100 Ohm</td><td>0 - 0.99</td><td>R = N / 100.0</td></tr>
<tr><td>100 - 189</td><td>1/10 Ohm</td><td>1.0 - 9.9</td><td>R = 1 + ((N - 100) / 10.0)</td></tr>
<tr><td>190 - 255</td><td>1 Ohm</td><td>10 - 75</td><td>R = 10 + (N - 190)</td></tr>
</table>
A value of 255 indicates an open circuit.</td></tr>
</table>

Examples, N=0 ==> R=0 ohms, N=99 ==> R=0.99 ohms.

Examples, N=100 ==> R=1.0 ohms, N=151 ==> R=6.1 ohms. N=189 ==> R=9.9 ohms.

Examples, N=190 ==> R=10 ohms, N=200 ==> R=20 ohms, N=227 ==> R=47 ohms.

Charge Cues Command Message (1100 000b)

Tell a Firing Module to charge its discharge capacitors. Some Firing Modules will start charging their discharge capacitors (if they have them) as soon as power is applied to the Firing Module. On some Firing Modules the charge current is large enough that if many Firing Modules start charging at the same time (power up for instance) from a shared power source the resitance will pull the voltage at the Firing Modules so low that the digital logic circuits will not function properly. For these Firing Modules the cues are not charged until the Charge Cue Command Message is sent to them to start charging. Software on the Control Module should be written to handle sending this command to the Firing Modules as fast as possible but still not put too large of a current draw on the power lines.

Firing Modules that have their own internal batteries doe not need to implement this command.

The Long Address should always be used to send this command message so that only one Firing Module charges at a time.

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	CMD				RSV			
Value	1100 0000b							

Charge Cues Response

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	STS				RSV			

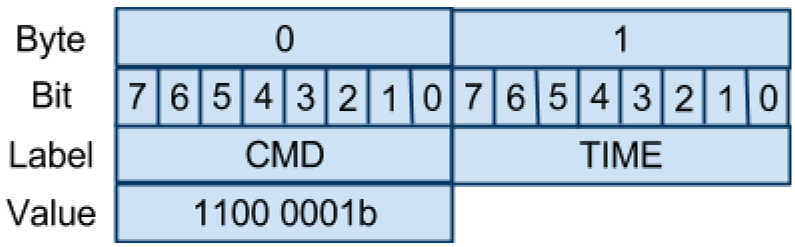
Name	Description
STS	Status - Ack, Ump
RSV	Reserved

Ack - message accepted and charging will start immediately.

Ump - The message is not supported by the Firing Module.

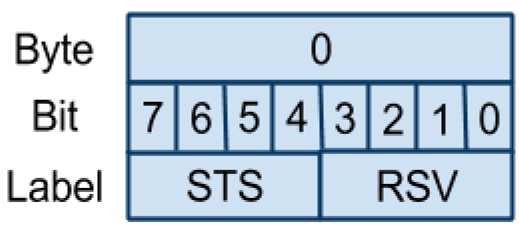
Set Fire Pulse Width Command Message (1100 0001b)

Sets the amount of time that the firing current/voltage is applied to a cue when it is fired.



Name	**Description**
CMD	1100 0001b Report Input Voltage Command Message
TIME	Multiple of 10mS for cue firing pulse width. 0 = 10mS. 255 = 2.56 Seconds.

Set Fire Pulse Width Response



Name	Description
STS	Status - Ack, Ump
RSV	Reserved

Ack - message accepted and charging will start immediately.

Ump - The message is not supported by the Firing Module.

Report Input Voltage Command Message (1100 0010b)

This command is used to determine the voltage being used by the Firing Module. For Firing Modules that have internal batteries this voltage gives an indication of the power left in the battery. For cable powered Firing Modules this gives and indication of the quality of the connections between the power source and the Firing Module. Poor power connections should be eliminated before a show.

Report Input Voltage Response

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	STS				VFRAC				VOLTS							

Name	Description
STS	Status - Ack, Ump
VFRAC	Tenths of a volt, 0 - 9
VOLTS	Whole volts, 0 - 255

Cue Schedule Command Message (1100 0011b)

The Cue Schedule Command Message is used to store on the Firing Module a schedule of what time to fire the cues of the Firing Module. Time is a number in units of mS from a time 0 that represents the start time of the pyrotechnic display. Time is set on the Firing Module using the **Time Application Message**. Time can be advanced by the **Time Application Message** or a separate time signal not covered by this protocol. When time reaches a scheduled cue time on the Firing Module the Firing Module must fire the cue. More than one cue on a Firing Module may be scheduled for the same time.

Cue Schedule Application Frame Format

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CMD								C	R	NUMC					
Value	1100 0011b										1 thru 63					

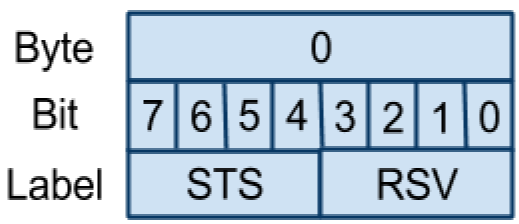
Byte	1+i*4								2+i*4								3+i*4								4+i*4							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CUEi								RSV				MSb										TIMEi									LSb
Value	1 thru 255								0000b				0 thru 1048575																			

Cue Schedule Application Frame Content Description

Label	Type	Value	Units	Description
CMD	8bit UInt	1100 0100b		The Command bits that indicate a Cue Schedule Command Message.
C	Flag			'1' indicates that all old Cue Schedules must be cleared before adding the Cue Schedules from this message. '0' indicates that Cue Schedules from this message must be merged with already stored Cue Schedules.
R	Flag			Reserved.
NUMC	5bit UInt	1 thru 63		Number of Cue Schedules in this message.
CUEi	UByte	1 thru 255		The Firing Module Cue number to schedule.
RSV				Reserved
TIMEi	20bit Uint		10mS	This is the time in mS from time 0 that the cue number CUEi is to be fired.

At a 9600 baud rate the time to transmit this message with 63 Cue Schedules is (260 bytes * 8 bits * 125% / 9600 baud) = 0.271 S.

Cue Schedule Response



Name	Description
STS	Status - Ack, Ump
RSV	Reserved

Ack - The command was accepted and execution started. The Ack must be sent as soon possible to keep the Control Module from a Timeout while waiting for the Ack.

Ump - the Firing module does not support this command.

Ack Command Response (0x0)

The Ack message is used by a Firing Module to indicate that it successfully complied with a command message. There may be other bytes also returned in the response.

BPR Command Response (0x2)

This is the response a Firing Module sends to a Control Module when the Firing Module encounters parameter/argument in a command from the Control Module that is out of range for the Firing Module.

Ump Command Response (0x3)

This is the response a Firing Module sends to a Control Module when the Firing Module does not know how to handle a command received from the Control Module. In this case there should only be 1 byte in the response.

Nak Command Response (0x1)

The Nak message is used by a Firing Module to indicate that it could not comply with a command message. There should only be 1 byte in the response.

Issues to Resolve

- How long should a Control Module wait for a response from a Firing Module before timing out and retrying.
- How many times should Control Module retry a command message when there is no response from a Firing Module.
- Should Bridge Modules announce themselves with information on baud rate on each side of the bridge so that Control Module timeout periods can be calculated correctly.
- The time slot response mechanism does not work well if there are slow bit rate (<100kbps) subnets in the network.

Appendix A

Generation of the Builders ID used for the BID field of the 4 byte Unique Address in the MAC Frame is described here.

Wikipedia has information about the Rabin Fingerprint that is specifically recommended for generating BIDs and there is also information in general about fingerprint computation. The irriducible polynomial p(x) to use for generating BID's has been randomly chosen to be '96AE17h'. All BID generators should use this number. This is a degree 23 polynomial (24 bits) that is used to create a 23 bit BID.

Java code for working with polynomials in general and the Rabin's polynomials specificly is found at http://code.google.com/p/rabinfingerprint/source/checkout. You will need collections-generic-4.01.jar to run the code.

Examples of generated fingerprints are:

PU Name	BID

LSearl	02772F
pikoko	14CAC7
stuntborg	27CFC7

An example of using the code to generate a fingerprint follows:

```
package org.pncp;

import java.math.BigInteger;

import org.bdwyer.polynomial.Polynomial;

/** Main routine that uses the Rabin method if fingerprint generation.

 *  Change the String 'builderPUname' to a string that identifies the builder
of a

 *  Pyro Network Firing Modules (like the builder's Pyro Universe user id).

 *  Run the main to see the output of the generated fingerprint.

 * @author searl

 *

 */

public class GenBIDfingerprint {

      /** Static main of the application for generating a Pyro Network Control
Protocol BID fingerprint.

       * @param args

       */

      public static void main(String[] args) {

            // Change the
```

```
            String builderPUname = "pikoko";


            long degree = 23; // 24 bit polynomal. 23bit fingerprint.

            String polyPhex = "96AE17"; // 23 degree polynomial to modulo the
strings with.


            // create a P(x) irreducible polynomial from the hex value

            Long polyPlong = Long.parseLong(polyPhex, 16);

            Polynomial polyP = Polynomial.createFromLong(polyPlong);

            polyP.setDegree(BigInteger.valueOf(degree));

            System.out.println(String.format("p(x)=%sh%n",
polyP.toHexString()));


            // Convert the name string to an array of bytes

            Polynomial namePoly =
Polynomial.createFromBytes(builderPUname.getBytes());

            System.out.println(String.format("String identifying builder
is='%s'%n", builderPUname));


            // generate the fingerprint

            Polynomial fingerPrint = namePoly.mod(polyP);

            System.out.println(String.format("23 bit fingerprint=%sh%n",
fingerPrint.toHexString()));


      }


}
```

Changes

Version 0.01

- Change name Short Address to Group Address
- Removed End of Frame byte from PHY Frame
- Changed MAC Frame Length from 8 bits to 4 bits that represent a mapping to the MAC Frame Length.
- Change MAC Command Type name to Msg Type.
- Removed MAC NAK Command Type
- Removed Address Parity from MAC Frame since MAC NAK has been removed.
- Changed MAC Address Type to be 2 bits instead of 1 bit for future addressing needs.
- Removed Sequence Number field from MAC Header. Message segmenting will be handled at the Application by commands that need it.
- Added Time Command Message.