Pyro Network Control Protocol: Data Link Layer - Serial

Author: Leon S. Searl, April 21, 2017
Contributors: PyroUniverse.com Users (mutski, pikoko, stuntborg)
Version 0.01

1 Introduction

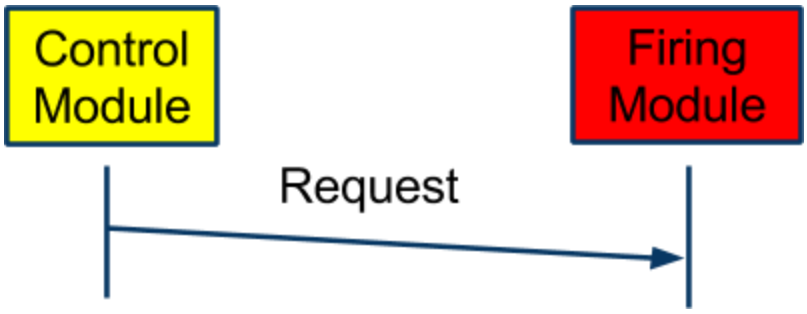
This document is the Pyro Network Control Protocol (PNCP) Data Link Layer (DLL) specification for implementation on an underlying serial digital data transport with no Data Link Layer of its own. Examples of this type of digital data transport are EIA-485, EIA-422, EIA-232, etc. This layer corresponds to the ISO Model layer 2. Note that the Pyro Network Control Protocol does not have the Network, Transport, Session or Presentation Layers of the ISO model.

The PNCP DLL is a piece of software and a frame format between the Physical Layer and the Application Layer of the Pyro Network Control Protocol. The DLL software fills in the DLL Frame with information from the Application Layer for transmitted frames and then passes the DLL frame to the Physical Layer. For received frames it extracts information from DLL 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 Start of Frame (SOF) byte of the frame determines if a message is going to a Firing Module or a Control Module. The SOF also determines if messages from the Control Module are Broadcast, Group Addressed or Unique Addressed. Broadcast frames are processed by all Firing Modules in the network. Group Addressed frames are processed by all Firing Modules that have the same Group Address as the Group Address in the message. Frames with a Unique Address are processed by only the one Firing Module in the network with the Unique Address.

The method 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 that send responses. Not all Control Module requests require a Firing Module response. Firing Modules never initiate communication.

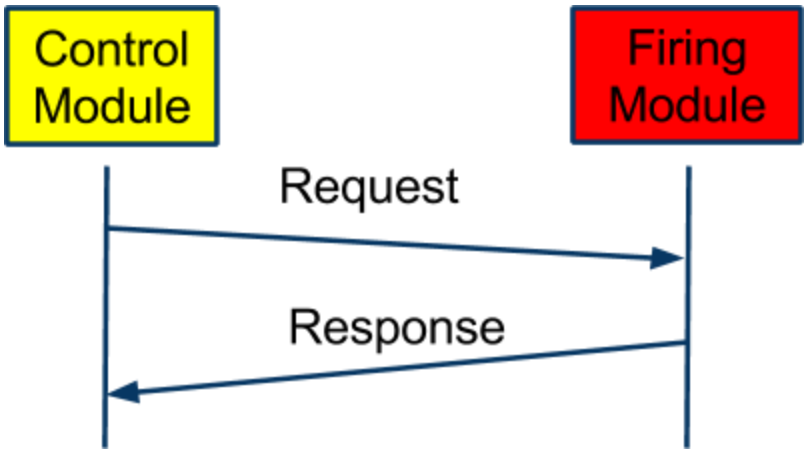
Request only Communication



Request only communication occurs for any command from the Control Module that does

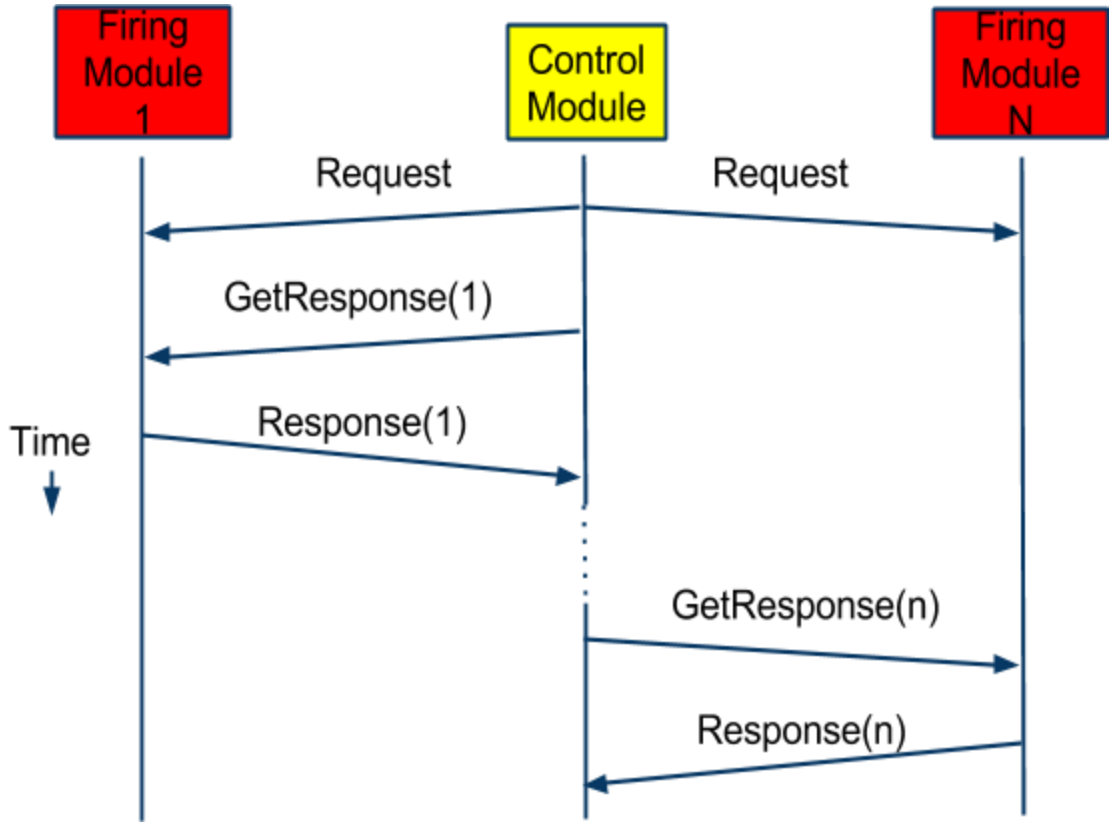
not require a response from the Firing Module. The Application level request 'Time' is such a request. Request only communication can occur with all 3 addressing modes (Broadcast, Group Address and Unique Address).

Request-Response Communication



The Request-Response communication mechanism occurs when the request from the Control Module will be processed by only one Firing Module. This is the case with Unique Address addressing and with Group Address addressing when there is only one Firing Module in the group.

Request-GetResponse-Response Communication



The Request-GetResponse-Response communication mechanism is used when multiple Firing Modules are to reply to a request. The initial Request is received by all addressed Firing Modules (Broadcast or Group Address). The GetResponse(i) request is sent to each Firing Module individually to prompt the Firing Module to send its response to the Request.

2 Notations

0x59 - Hexadecimal numbers begin with a '0x'

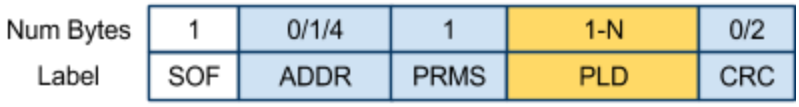
0000 1011b - Binary numbers end with a 'b'. Spaces are added every 4 bits to make it easier to read.

3 Data Link Layer Frame Formats

This section of the document describes the format of the Data Link Frames. The frames are byte oriented with 8 bits per byte. In all descriptions the MSb is shown on the left of the byte. Which bit occurs first in the Physical Layer (MSb/LSb) is dependent on the Physical Layer used.

3.1 DLL Frame General Format

This section give an overview of the Data Link Layer data frame formats.



Num Bytes	1	0/1/4	1	1-N	0/2
Label	SOF	ADDR	PRMS	PLD	CRC

Name	**Description**
SOF	**Start of Frame** - The Start of Frame byte is 1 byte with 4 possible values. Each value indicates the type of the Data Link frame. The 4 values of the Data Link frame SOF are: • SOFB(0x55) - Broadcast - Control Module transmits message to be processed by all Firing Modules. • SOFG(0x47) - Group Address - Control Module transmits message to be processed by a specified group of modules. • SOFU(0x78) - Unique Address - Control Module transmits message to be processed by a single specified module. • SOFR(0x6A) - Response - A Firing Module sends a response back to the Control Module. Note that the values chosen for the SOF bytes are also the 4 control values that are used in 6b/8b encoding so that the same SOF values can be used if 6b/8b line code encoding is used on the Data Link Frame. If 8b/10b is to be used to encode the data stream then a new version of the DLL layer should be written that uses different values for Start Of Frame.
PRMS	**Frame Parameters** - There are several parameters in the DLL header that determine the structure of the frame and thus how the frame is processed. This byte must be the first byte in the frame so that firmware will know what the structure of the rest of the frame will be. This is particularly true if a new version of the protocol is developed (FV=1b) causing the address lengths or address types to change. 

Bit	7	6	5	4	3	2	1	0
Label	FV	PLI				PT	RSV	CF
Value	0b	0 - 15				Xb	0b	Xb

<table>
<tr><td></td><td>
<table>
<tr><td>Name</td><td>Description</td></tr>
<tr><td>FV</td><td>Frame Version - For version 1.0 of the protocol this value is 0. If a future version of the protocol is developed this value will be 1b. Firing Modules that only know how to processes v1.0 of the protocol must ignore frames where FV=1b.</td></tr>
<tr><td>PLI</td><td>Payload Length Index. The value of this field is used as an index into a table the contains the actual length of the payload. Using this method of length specification requires far fewer bits in the frame at the expense quantized frame lengths.</td></tr>
<tr><td>PT</td><td>Payload Type - This value indicates the type of the payload. 0b indicates a Data Link Layer Sub Frame. 1b indicates an Application Layer Frame.</td></tr>
<tr><td>RSV</td><td>Reserved</td></tr>
<tr><td>CF</td><td>CRC flag. 0 indicates CRC is omitted from message. 1b indicates that there is a 16bit at the end of the message.</td></tr>
</table>
</td></tr>
<tr><td>ADDR</td><td>Address - The address is 0 bytes for the Broadcast message and the Response message. Messages addressed to a group of Firing Modules use a 1 byte Group Address. A message that is addressed to one specific Firing Module uses a 4 byte Unique Address.</td></tr>
<tr><td>PLD</td><td>Payload - The payload is a variable length frame. The payload may be an Application Frame or a DLL Sub Frame.</td></tr>
<tr><td>CRC</td><td>Cyclic Redundancy Check of the Data Link Frame excluding the SOF byte and the two bytes of the CRC. The CRC is used to determine if there are bit errors in the frame header and payload. The CRC may be omitted if the physical layer is very robust (unlikely to incur any bit errors) or has its own CRC. If the CRC is omitted the CF flag of the PRMS byte must be = 0 to indicate that the CRC has been omitted. The CRC occupies the bottom 15 bits of the 2 bytes. The CRC-CCITT-Kermit which is 16 bits with a polynomial of x^16 + x^12 + x^5 + 1 (0x8408) .

NOTE: This CRC function is built into some hardware and is in avr-libc http://www.nongnu.org/avr-libc/user-manual/group__util__crc.html

NOTE: On received frames the CRC is calculated after the escaped sequence have been converted into their 'normal' value. On transmitted frames the CRC is calculated before the escape sequences are added.</td></tr>
</table>

3.1.1 DLL Frame Byte Encoding/Decoding

The Pyro Network Control Protocol uses a special Start of Frame byte value to indicate to

receivers that a new Data Link frame is starting. To prevent byte values within the frame from being one of the SOF values and thus indicating a false Start of Frame an escape mechanism is used where the SOF value within a frame is replaced with two values. The first value is an escape byte and the following byte is a code to indicate which SOF value the escape sequence has replaced. If the escape byte value occurs in a frame it is also escaped.

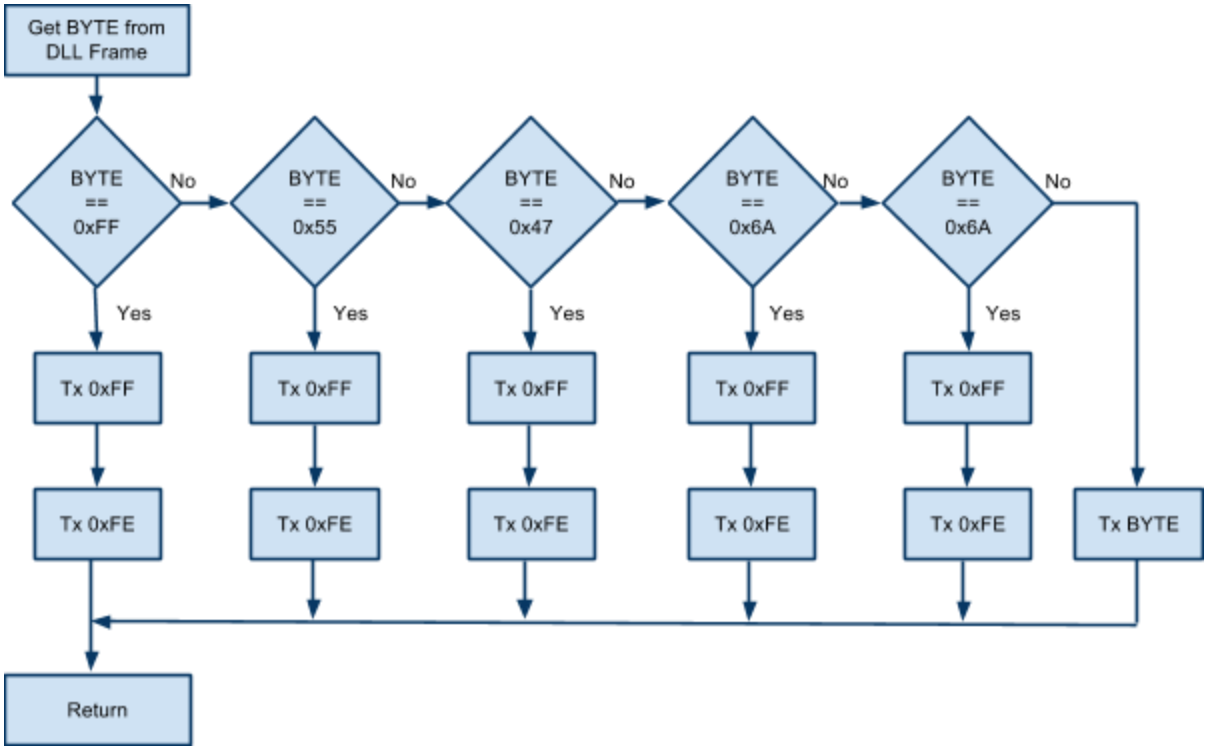
The value chosen for the SOF so that at least 2 bit errors would be required to change on SOF value into another SOF value. In addition, each SOF has 4 1's and 4 0's (same number of 1's and 0's). This helps with preventing DC bias in transmitting data.

The following table shows the sequence mappings for bytes within the DLL frame (the decoded and encoded values).

SOF Frame Type	**Decoded Value Hex**	**Decoded Value Binary**	**Encoded Values**
Escape value	0xFF	1111 1111b	0xFF 0xFE
Broadcast Data Link Frame	0x55	0101 0101b	0xFF 0xFD
Group Address Data Link Frame	0x47	0100 0111b	0xFF 0xFC
Unique Address Data Link Frame	0x78	0111 1000b	0xFF 0xFB
Response Data Link Frame	0x6A	0110 1010b	0xFF 0xFA

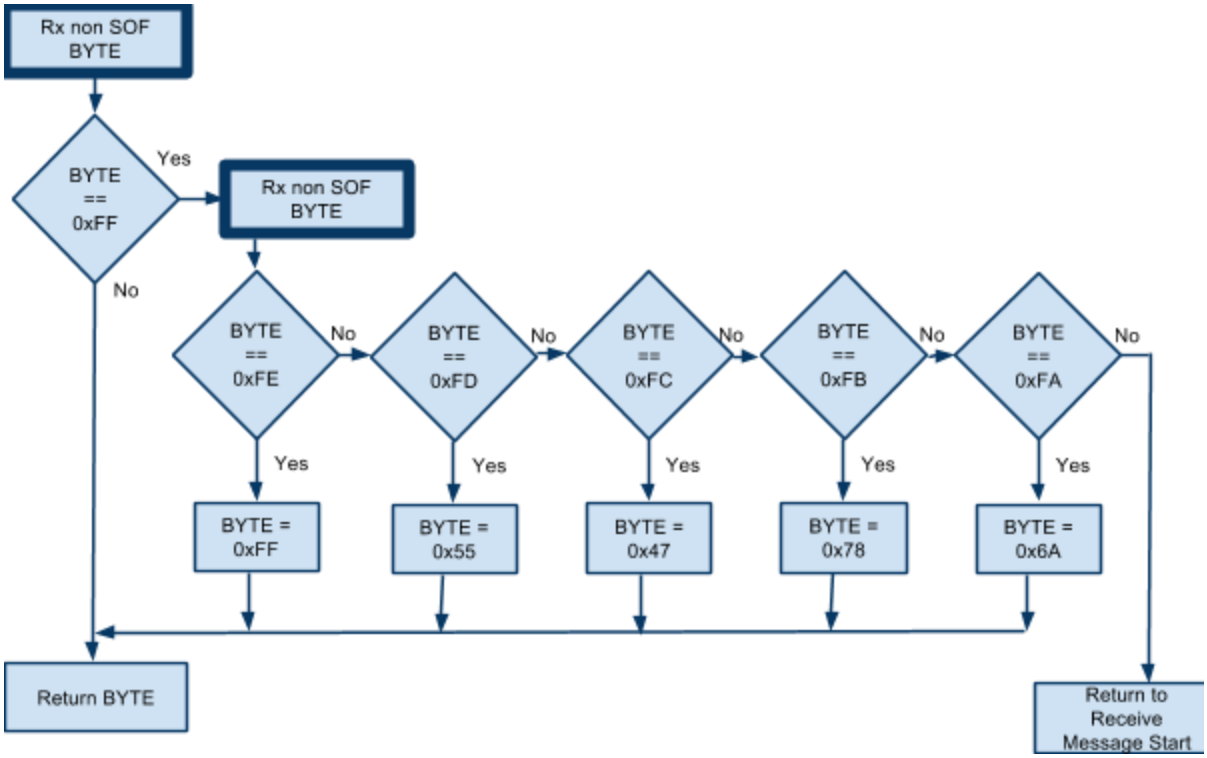
The following are flow charts for encoding and decoding the Data Link Frame bytes (excluding the Start of Frame byte).

Encoding Byte Flow Chart



The Encoding Byte Flow Chart shows the process for grabbing bytes from a DLL frame that is to be transmitted and encoding it to escape the SOF values that can occur in the DLL frame.

Decoding Byte Flow Chart



The Decoding Byte Flow Chart shows the process of decoding bytes from the received byte stream after a SOF byte has been detected. The decoded bytes are then CRC checked and placed into a DLL frame.

3.1.2 Payload Length Index

An index to the total number of bytes in the Application Frame. 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 Application Frame.

The Payload Length is an index into a table that contains the actual length number.

PLI Table

Index	Payload Length
0	1
1	2
2	3
3	4
4	5
5	6
6	8
7	12
8	16

9	24
10	32
11	48
12	64
13	96
14	128
15	255

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

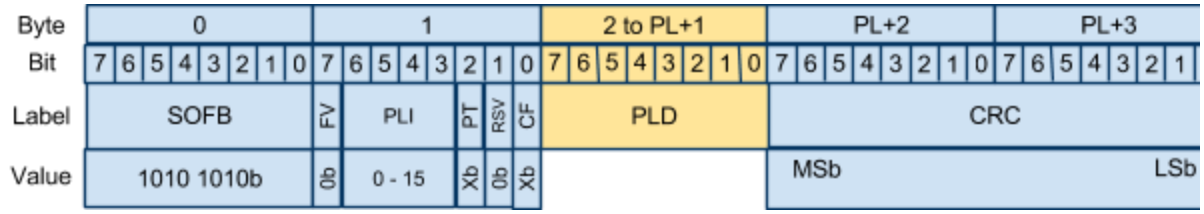
```
char length;
char afli;
if(length < 7) {
  afli = length - 1;
}
else if (length < 17) {
  afli = ((length - 1) / 4) + 5
}
else if (length < 33) {
  afli = (length / 8) + 6
}
else if (length < 65) {
  afli = (length / 16) + 8
}
else if (length < 129) {
  afli = (length / 32) + 10
}
else if (length < 256) {
  afli = 15
}
else {
  // This is an error.
}
```

Note: The Frame Length is the length of the App Frame before escape sequences are added for SOF values found in the MAC frame.

3.2 DLL Broadcast Frame Format

The Data Link Broadcast Frame Format has two uses:

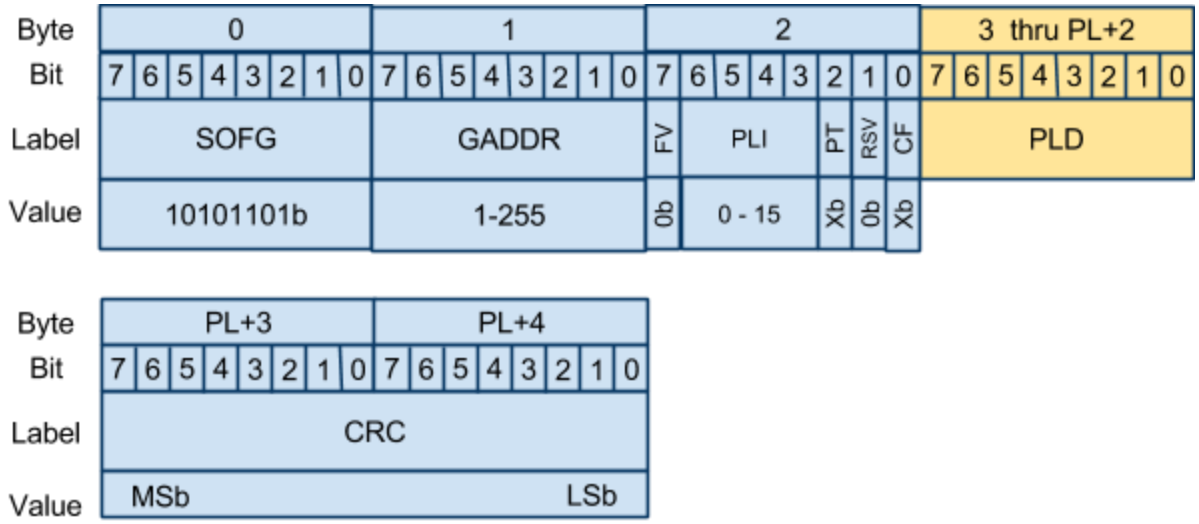
- Discover the Unique and Group address of the Firing Modules in the Pyro Network.
- Send the Time Application Frame to all modules in the Pyro Network.



Label	Type	Value	Units	Description
SOFB		0101 0101b		Start of Frame Symbol for a Broadcast frame.
FV	1bit			Frame Version.
PLI	4bit UInt			Payload Length Index
PT	1bit			Payload type. 0=Data Link Sub Frame, 1=Application Frame
RSV	2bit	00b		Reserved Flag bits.
CF	Flag			CRC Flag, '0' indicates no CRC, '1' is a 16 bit CRC.
PLD				Payload data.
CRC				CRC type is CAN protocol CRC. The CRC is omitted if CF is '1'.

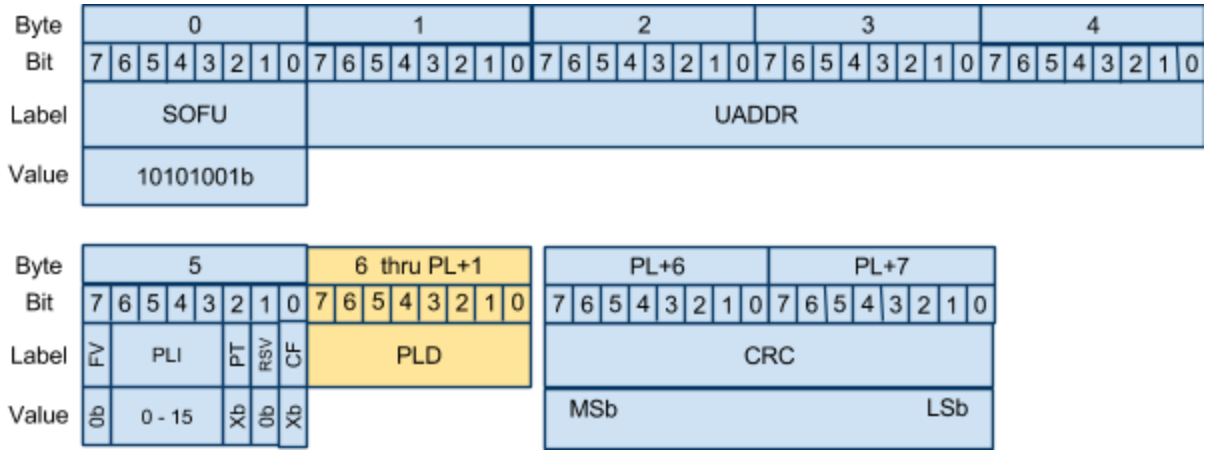
Firing Modules can not all respond to a Data Link Broadcast Frame at the same time or all of the responses would collide and none would be received successfully by the Control Module. A slot numbering mechanism is used to separate the responses from the Firing Modules across time. When a Firing Module is to respond to a Broadcast message it randomly picks a number from 0 to 255. It remembers this number for when a A new random slot number is chosen for each Data Link Broadcast Frame that arrives. The Firing Module then waits for the Control Module to send a Data Link Frame

3.3 DLL Group Addressed Frame Format



Label	Type	Value	Units	Description
SOF2		0100 0111b		Start of Frame Symbol for a Group Address frame.
GADDR	UByte	1-255		Group Address of the recipient Firing Module. Group address ‘0’ is to never be used in a frame. It is reserved for application use to indicate an unknown group address in it’s memory.
PLI	4bit UInt		Index	Payload Length Index, Index into a table containing Payload data.
RSV				Reserved Flag bits.
CF				CRC Flag, '0' indicates no CRC, '1' is a 16 bit CRC.
PLD	Data			Payload bytes.
CRC				CRC type is CAN protocol CRC. The CRC is omitted if CF is '1'.

3.4 DLL Unique Address Frame Format



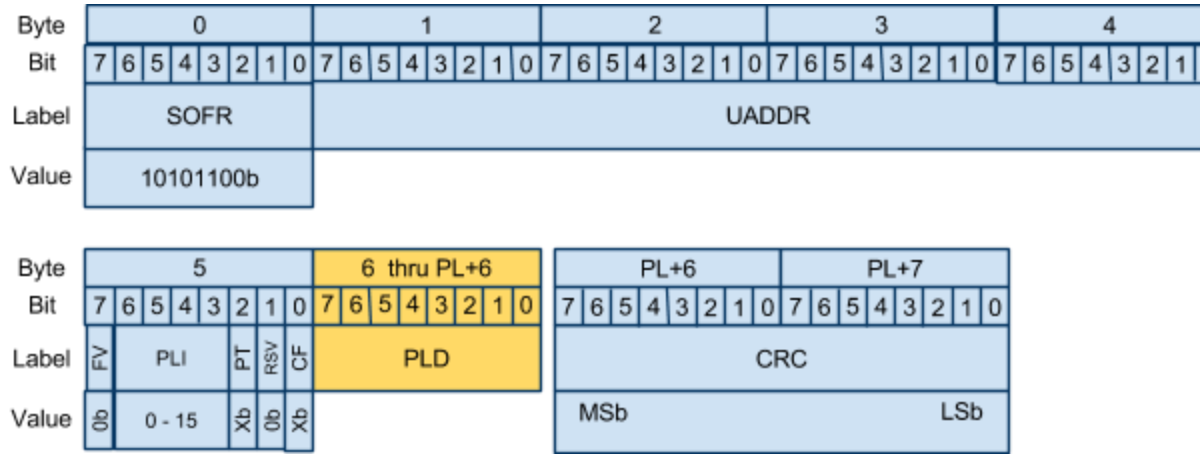
Label	Type	Value	Units	Description
SOFU		0111 1000b		Start of Frame Symbol for a Unique Address frame.
UADDR	UInt			Unique Address of the Recipient Firing Module
FV	1bit			Frame Version.
PLI	4bit UInt		Index	Application Frame Length Index, Index into a table containing Application Frame Length values.
RSV				Reserved Flag bits.
CF				CRC Flag, '0' indicates no CRC, '1' is a 16 bit CRC.
PLD	Data			Payload bytes.
CRC				CRC type is CAN protocol CRC. The CRC is omitted if CF is '1'.

3.4.1 Unique Address Composition

TBS

3.5 DLL Response Frame Format

The Data Link Layer Response Frame Format is used by the Firing Modules to send Responses from a Control Module command back to the Control Module.



Label	Type	Value	Units	Description
SOF4		0110 1010b		Start of Frame Symbol for a Response frame.
UADDR	UInt			Unique Address of the transmitting(responding) Firing Module.
FV	1bit	0b		Frame Version.
PLI	4bit UInt		Index	Payload Length Index, Index into a table containing Payload Length values.
PT	1bit			Payload type. 0=Data Link Sub Frame, 1=Application Frame
RSV				Reserved Flag bits.
CF	1bit		Flag	CRC Flag, '0' indicates no CRC, '1' is a 16 bit CRC. This CRC flag must match the CRC flag of the request message. In the case of a slot response the CRC flag is the flag of the 'Get Slot Response' request.
PLD	Data			Payload bytes.
CRC	2 Bytes			CRC type is CAN protocol CRC. The CRC is omitted if CF is '1'.

Application Frame
The *Application Frame* is a variable length frame containing control commands or responses.

CRC Flag
Some Physical Mediums that the Data Link uses may actually be fully functional network stacks that provide their own CRC handling. If this is the case the the CRC is not needed in the PNCP Data Link frame. If this flag is '0' then the CRC has been omitted from the Data

Link frame.

The Firing Module must follow the example of the Controller. If in the command frame the Controller omitted the CRC then the response from the Firing Module must also omit the CRC.

CRC

4 Data Link Layer Algorithms

4.1 Code/Decode Data Link Frame Bytes

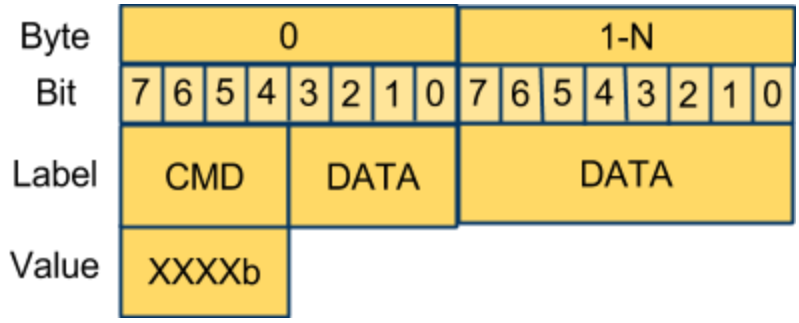
Bytes that are not the Start of Frame (SOF) must be coded and decoded. On a plain serial medium (example: EIA-485) the only coding done is to escape bytes in the frame that are the same value as one of the SOF values so that there are no false Start Of Frame bytes in the middle of a frame.

5 Data Link Sub Layer Commands

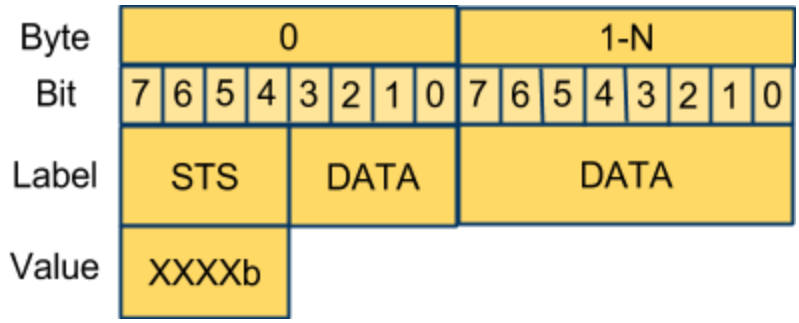
The Data Link Commands is used by the Control Module to establish orderly communication between the Control Module and the various Firing Modules on a network. Only the Control Module sends out commands. Firing Modules process commands and for some commands sends responses back to the Control Module. The commands for the Data Link Layer are:

- Get and Set Group Address
- Set Group Slot Response Number
- Request for Slot Response
- Ignore Next Command
- Get and Set Unique Address

The DLL Command is one type of payload in the Data Link Frame. The other type of payload is the Application Frame. Data Link Layer Commands are processed by the Firing Module Data Link Layer and are NOT passed up to the Application Layer of a Firing Module.



For the DLL commands that only return a status response the following are the valid response values.



Status responses:

ACK - 0x00 - There may be other bytes in the response.

NAK - 0x01 - There are no other bytes in the response.

BFL - 0x02 - Bad Frame Length - There are no other bytes in the response.

5.1 Get Group Address DLL Command

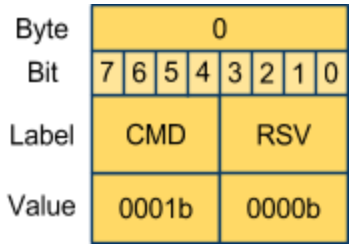
The Get Group Address Sub Command Sub Frame is usually the payload of a Broadcast Data DLL Frame. It's purpose is for the Control Module to discover the Group Addresses of the Firing Modules in the Pyro Network.

If the firing module implements Group Addressing then for Firing Modules without Front Panel Group Address switches the Group Address for the module after power up must be 0x00.

Frame Addressing this Sub Frame Command can be used in:

- Broadcast - **YES** - This will return the Groups Address for all Firing Modules.

- Group Address - **YES** - This is useful to determine which if any Firing Modules are using a specific Group Address. The specific use case for this is when Firing Modules use switches to set the Group Address instead of being programmable with the **Set Group Address DLL Command**.
- Unique Address - **YES** - This will return the Group Address for a specific Firing Module.

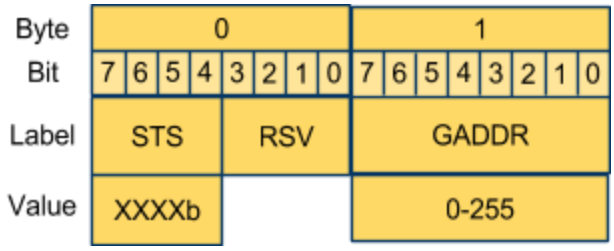


The value of the CMD for the Get Group Address Data Link Command is 0001b.

5.1.1 Get Group Address DLL Command Response

If the **Get Group Address DLL Command** request was by Broadcast or Group command then the Firing Module waits for the next **Get Slot Response DLL Command** that has its Broadcast slot or Group slot before responding. The Firing Module ignores all **Get Slot Response DLL Commands** that do not match its Broadcast Slot or Group slot.

While waiting for it's **Get Slot Response DLL Command** to arrive if a DLL Command arrives that is not a Get Slot Response DLL Command then the Firing Module abandons waiting for Get Slot Reponse DLL Command and processes the new DLL Command.



STS	ACK NAK
GADDR	The group address (GADDR) of the Firing Module. A GADDR value of 0 indicates that the Firing Module does not have a Group Address set. If the STS==NAK this byte must still be present but the value could be anything.

- ACK - The command was successfully processed and the current Group Address of the Firing Module was returned. GADDR field has the current Group Address of the

Firing Module.

- NAK - The Firing Module does not support Group Addressing. The GADDR field is not defined in this case.

Since the DLL Response frame contains the Unique Address of the Firing Module the Control Modules receives both the Group Address and the Unique Address of the Firing Module with this response.

5.2 Set Group Address DLL Sub Frame Command

The Group Address may set using a switch on the Firing Module front panel but some Firing Modules may not have a front panel address switch. The Set Group Address sub frame command can be used to set (for modules without front panel address switch) the group address for a Firing Module without a Group Address front panel switch.

Frame Addressing types that this DLL Command can be used in are:

- Broadcast - **NO** - Although this could be done there is no practical benefit. This would just cause all modules have the same group address. A firing module should consider this situation a bad command regardless of weather it has a front panel Group Address switch. The Firing Module must not respond and should show the user a Bad Command indication.

- Group Address - **NO** - Although this could be done there is no practical benefit. This would just cause all modules in a group to change their group address. A Firing Module should consider this a bad command regardless of weather it has a front panel Group Address switch. The Firing Module must not respond and should show the user a Bad Command indication.

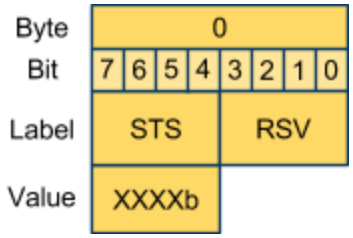
- Unique Address - **YES**

5.2.1 Set Group Address DLL Sub Frame Format

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CMD				RSV				GADDR							
Value	0010b				0000b				0-255							

The value of the CMD for the Set Group Address DLL Command is 0010b.

5.2.2 Set Group Data Link Sub Frame Response



STS - The valid status result values are:

- ACK - The Group Address was changed.
- NAK - The Group Address could not be changed. This result must only be sent if the Firing Module has a manual switch for setting the Group Address.

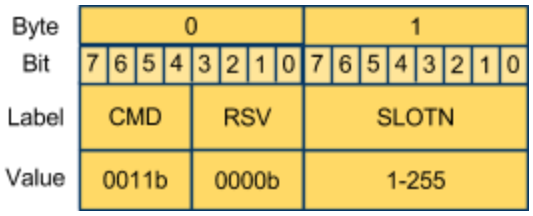
5.3 Get Slot Response DLL Command

The **Get Slot Response Data Link Command Command** is used to tell a Firing Module to send the response from a previously sent **Broadcast** or **Group Addressed Data Link Frame**. The Firing Module with the matching SLOTN responds by sending the response. Each Firing Module in a Group is assigned its own slot number by the Control Module using the **Set SlotN Data Link Command Sub Frame Format**. Slot Numbers for Broadcast Data Link Frames are calculated randomly for each response.

Firing Modules have both a Broadcast Slot and a Group Slot. The Broadcast Slot is used if the **Get Slot Response Data Link Command Sub Frame** arrives in a Broadcast Data Link Frame. The Group Slot is used of the **Retrieve Slot Data Link Command Sub Frame** arrives in a **Group Address Data Link Frame**.

Frame Addressing types that this DLL Command can be used in are:

- Broadcast - **YES** -
- Group Address - **YES** -
- Unique Address - **NO**



5.3.1 Get Slot Response DLL Command Response

The response from the previous Broadcast or Group Address Data Link Frame (or either Application or Data Link Command Sub Frame).

5.4 Set Group Slot DLL Command

The **Set Group Slot DLL Command** is used to set an individual Firing Module's Group Slot Number. This is the slot number that the Firing Module is to respond to when the **Retrieve Slot Command** for the Firing Module' group is received.

If a Firing Module has never received a **Set Group Slot DLL Command** then it is to use random slot numbers as in Broadcast slotting. Once the Set Group Slot DLL Command is received by a Firing Module it will not change its group slot until the next **Set Group Slot DLL Command**.

Frame Addressing types that this DLL Command can be used in are:

- Broadcast - **NO** -
- Group Address - **NO** -
- Unique Address - **YES**

Byte	0								1							
Bit	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Label	CMD				RSV				SLOTN							
Value	0100b				0000b				0-255							

5.4.1 Set Group Slot Data Link Command Sub Frame Response.

There is no response for this command.

5.5 Ignore Next Command DLL Command

The Command Module sends this command to a specific Firing Module (using the SOFU unique address) to tell it to ignore the next command that it would have otherwise processed.

This command is typically only used during the initial process of locating the Firing Modules in the network. A Command Module repeatedly sends the Get Group command and asks for responses from Firing Modules until no corrupted responses (due to packet collisions) are detected. To reduce packet collisions the Command Module tells each known Firing Module not to respond to the next command (Get Group).

The Firing Module is to ignore the next command that is addressed to it (Broadcast command, Group command that matches the Firing Modules Group Address, Unique Address command that matches the Firing Module's Unique Address). Commands to other Group Addresses or other Unique Addresses should be ignored as normal and do not count as an ignored command.

If a Firing Module receives more than one consecutive Ignore Next Commad

Frame Addressing types that this DLL Command can be used in are:

- Broadcast - **NO** - There is no known situation that can utilize telling all modules on the network to ignore the next command.
- Group Address - **NO** - There is no known situation that can utilize telling all modules with with a specific group address to ignore the next command.
- Unique Address - **YES -** The Firing Module with the specified address will ignore the next broadcast command, group command with it's group address or command with it's unique address.

Byte	0							
Bit	7	6	5	4	3	2	1	0
Label	CMD				RSV			
Value	0101b				0000b			

The value of the CMD for the Get Group Address Data Link Command is 0101b.

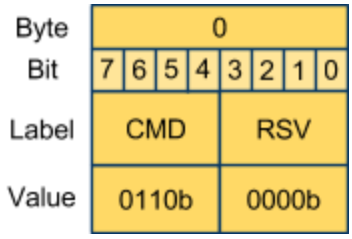
5.5.1 Ignore Next DLL Command Response

There is NO response to this command. This is partly to reduce the amount of time it takes to send out to possibly of sending the command to hundreds of Firing Modules. It is not critical if a Firing Modules fails to receive the command because of corrupted bits and then responds to the next command since this command is intended to only be used for discovering Firing Modules.

5.6 Get Unique Address Command Data Link Command Sub Frame

This command is used to program the Unique Address of a Firing Module. The command can be used to change an already set Unique Address of a firing module.

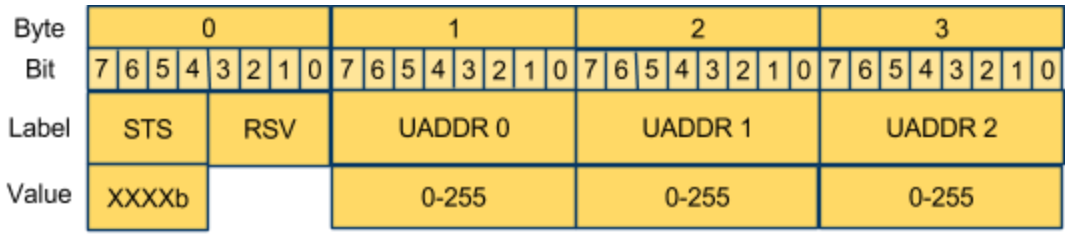
The command must only be used with 1 Firing Module in the network. If more than 1 Firing Module receives this command then the address is not longer unique (more than one module has the same address) and will cause problems with the system.



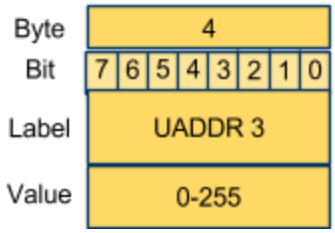
Byte	0	
Bit	7 6 5 4 3 2 1 0	
Label	CMD	RSV
Value	0110b	0000b

The value of the CMD for the Get Unique Address Data Link Command is 0110b.

5.6.1 Get Unique Address Command Data Link Command Sub Frame Response



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	RSV	UADDR 0	UADDR 1	UADDR 2
Value	XXXXb		0-255	0-255	0-255



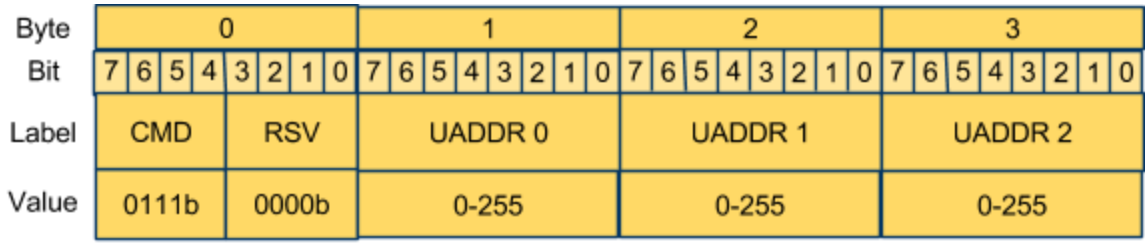
Byte	4
Bit	7 6 5 4 3 2 1 0
Label	UADDR 3
Value	0-255

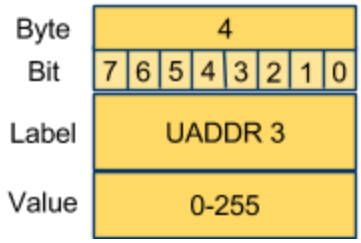
5.7 Set Unique Address Command Data Link Command Sub Frame

This command is used to program the Unique Address of a Firing Module. The command can be used to change an already set Unique Address of a firing module.

The command must only be used with 1 Firing Module on the network. If more than 1 Firing Module receives this command then the address is no longer unique (more than one module has the same address) and will cause problems with the system.

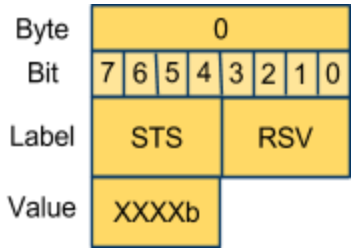
The Unique Address must be stored into nonvolatile memory on the Firing Module so that it is remembered after being powered down and back up again.





The value of the CMD for the Set Unique Address Data Link Command is 0111b.

5.7.1 Set Unique Address Command Data Link Command Sub Frame Response



6 User Visual Indications

Talk about flashing leds for various indications.

7 Control Module DLL Protocol Flow Charts

Control Modules need a mechanism to discover the Unique Address and Group Address of Firing Modules on the Pyro Network. Without the Firing Module Address discovery mechanism display operators would be required to write down the Unique Address and Group Address of each Firing Module and then enter the addresses into the control software on a PC or Control Module. This would be a tedious and error prone process.

TBS

8 Firing Module DLL Protocol Flow Charts

This section provides flowcharts to illustrate the expected behaviour of Firing Modules at the Data Link Layer level and to provide a starting point for implementations. It is not expected that implementations will follow the Flowcharts exactly. Differing languages, micro controller architectures and coding techniques will necessarily cause implementations to be different from the flowcharts.

Firing Modules typically sit and wait for request messages from the Control Module. To have immediate responses to Application Requests like the Report Continuity or Report Cue Resistance request the Firing Module may periodically check cue continuity or cue resistance and store the result for quick response when the report request arrives. The flowcharts in this section do not deal with actions taken at the application level such as periodic tasks.

Blocks in the Flowcharts with a thick border indicate that there is another flowchart that shows the details inside the block.

'==' indicates an equivalence test.

'=' indicates an assignment.

8.1 Firing Module Received Message Flow Chart

The Firing Module Received Message Flow Chart is the top level flowchart for the Firing Module (excluding microcontroller register setup and application level periodic tasks). The main purpose of this flowchart is to determine if a received message is destined for the Firing Module the code is running on and to determine if the message CRC matches the calculated CRC.

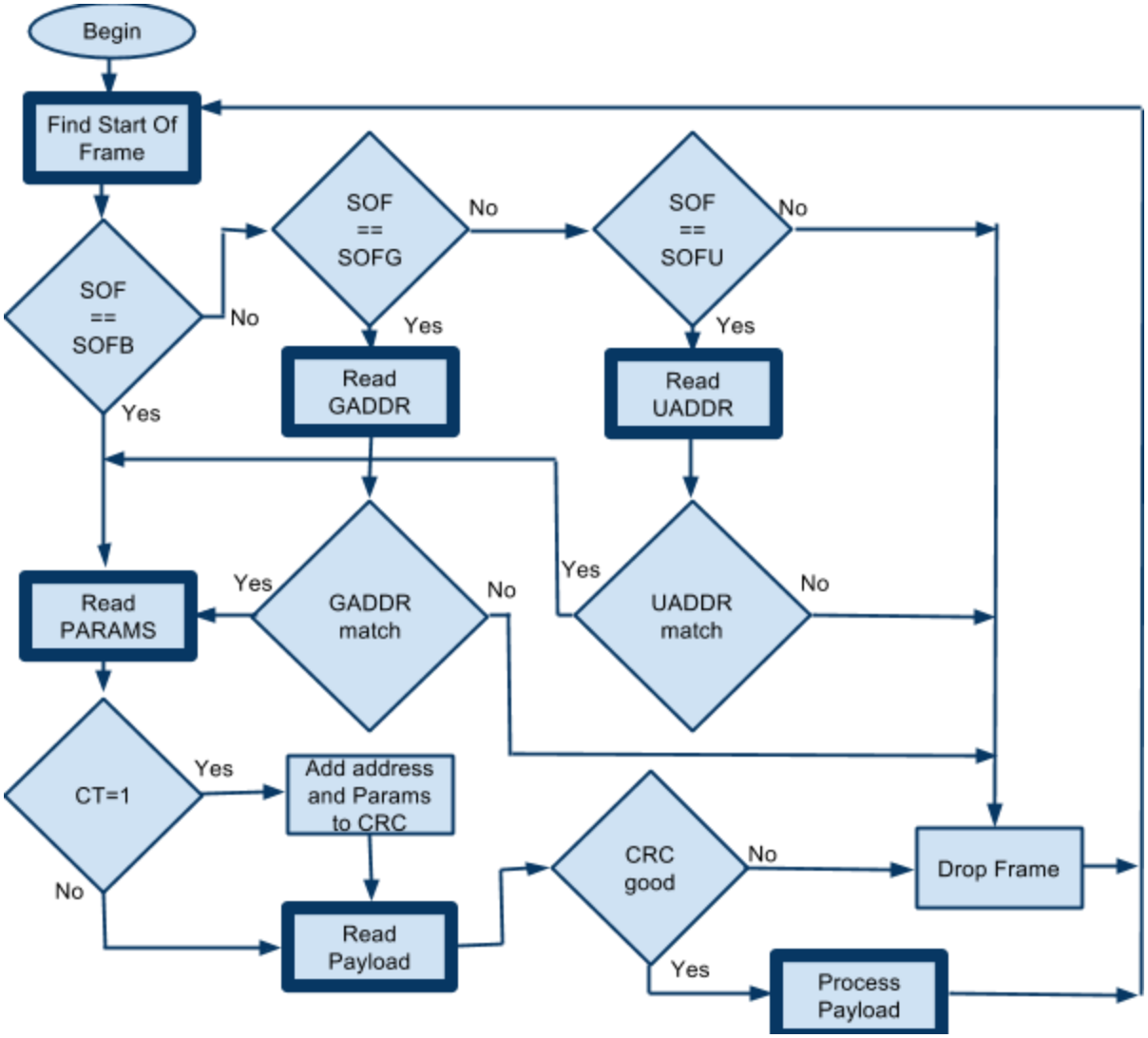
The following are explanations for block labels that may not have an apparent meaning.

GADDR match - Does the GADDR in the message match the Group Address of the Firing Module.

UADDR match - Does the UADDR in the message match the Unique Address of the Firing Module.

CRC good - Does the calculated CRC of the frame match the CRC found at the end of the frame.

Drop Frame - We don’t want the frame so forget about it and go back to looking for a SOF.



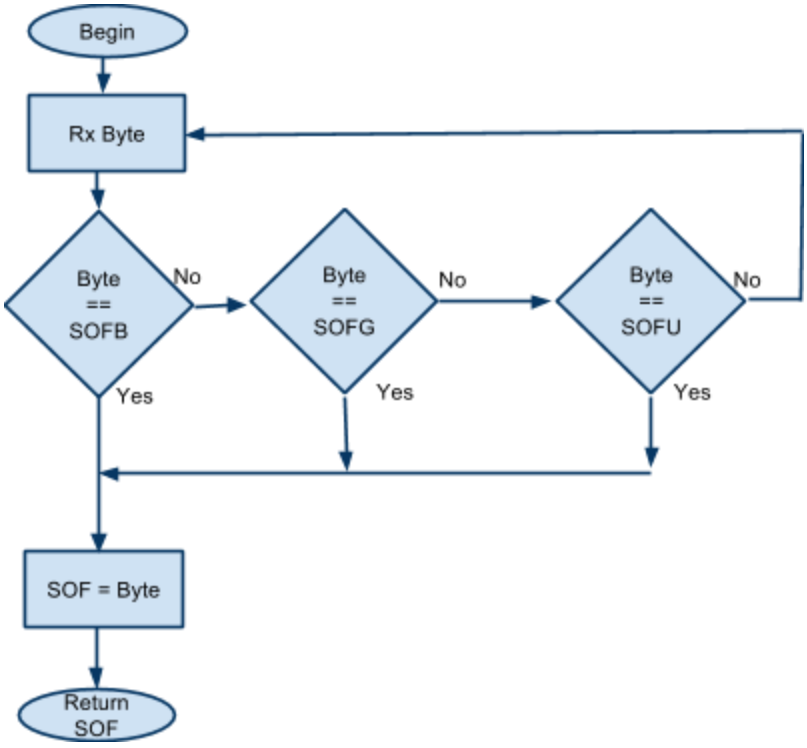
8.2 Find Start of Frame Flow Chart

The Find Start of Frame Flow Charts reads bytes from the UART of the Firing Module until one of the 3 inbound SOF bytes is recognized.

The value of the SOF is returned since later processing depends on the value of the SOF.

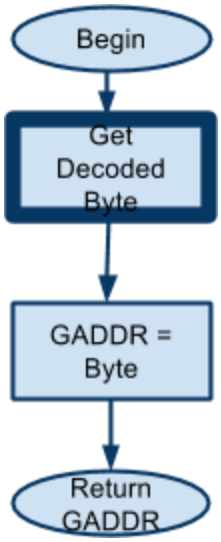
The following are explanations for block labels that may not have an apparent meaning.

Rx Byte - Read a byte directly from the UART.



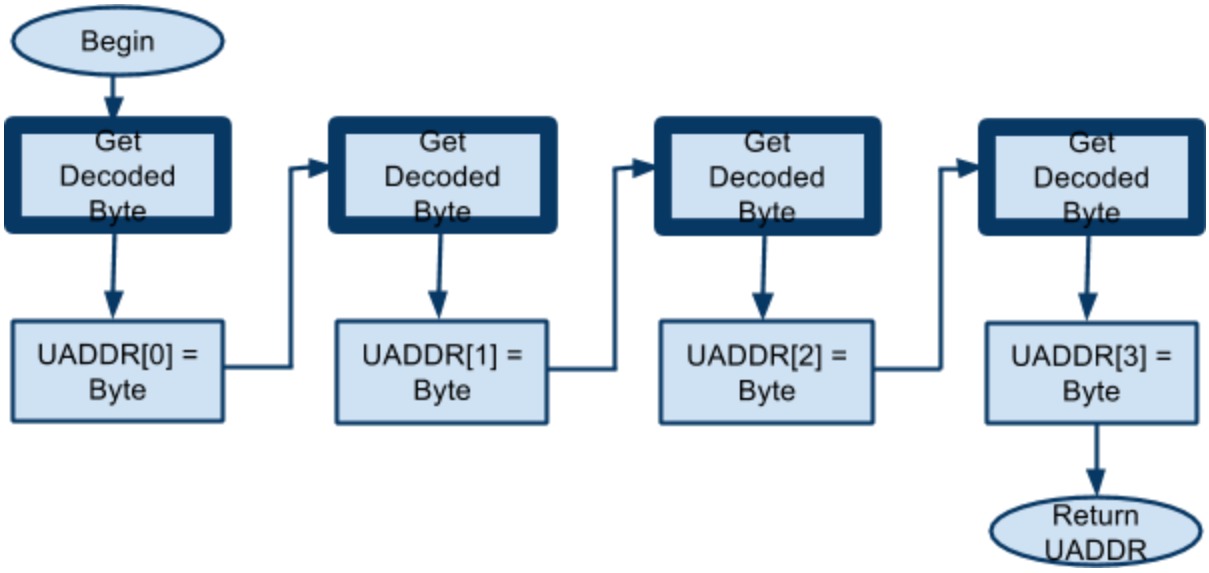
8.3 Read GADDR Flow Chart

The Read GADDR Flow Chart shows the Group Address byte being read, CRC checked and assigned to a temporary variable GADDR for use in a later flow chart.



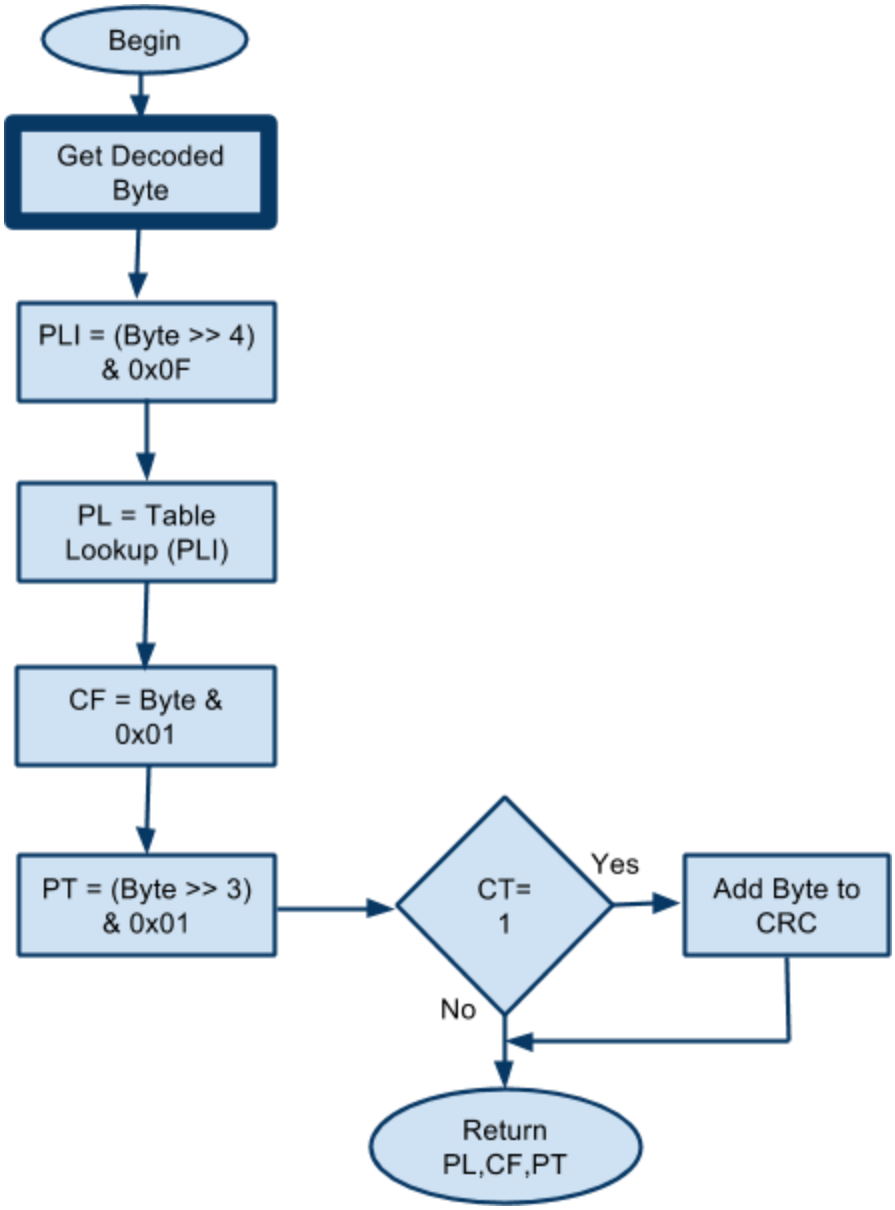
8.4 Read UADDR Flow Chart

The UADDR Flow Chart shows a Unique Address being obtained from a DLL frame. Each byte of the address is CRC checked and then assigned to a temporary variable UADDR. The flow chart shows UADDR being a byte array but a 4 byte integer could also be used.



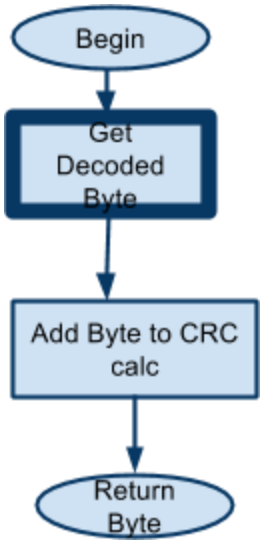
8.5 Read Params Flow Chart

The Read Params Flow Chart shows the 1 byte parameters value being read from the DLL frame. The byte is CRC checked and then the values from the byte are extracted. The Payload Length Index (PLI) is extracted and the the Payload Length is determined by table lookup using the PLI as the index into the table. The CRC flag (CF) is extracted. The Payload Type (PT) is extracted.



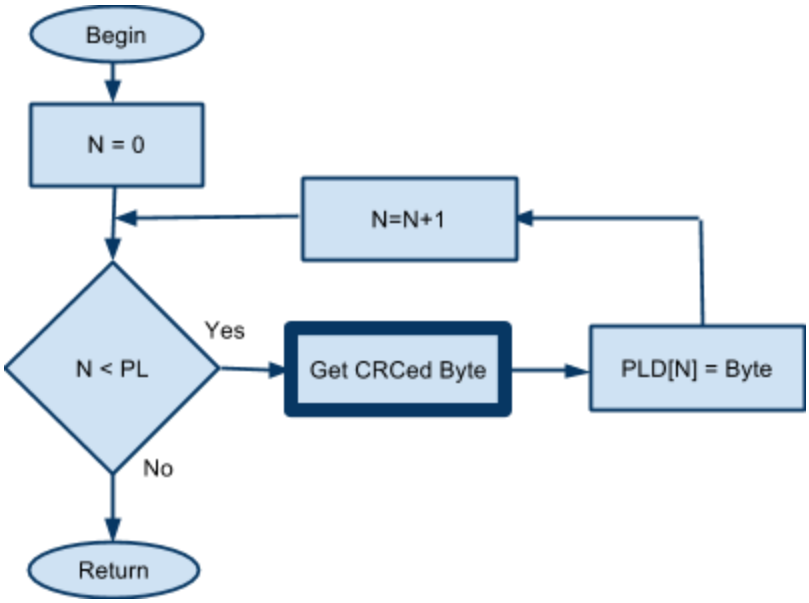
8.6 Get CRCed Byte Flow Chart

Bytes read for the payload are added to the CRC as they are read.



8.7 Read Payload Flow Chart

This flow chart shows that a loop is used to read (Get CRCed Byte) through the number of bytes designated by the Payload Length (PL) and put the bytes into an payload array. Since none of the payloads sent out by the Control Panel are large except for the Application Layer Cue Schedule command, small micro controllers should have enough RAM to put the payload into a RAM stored array.

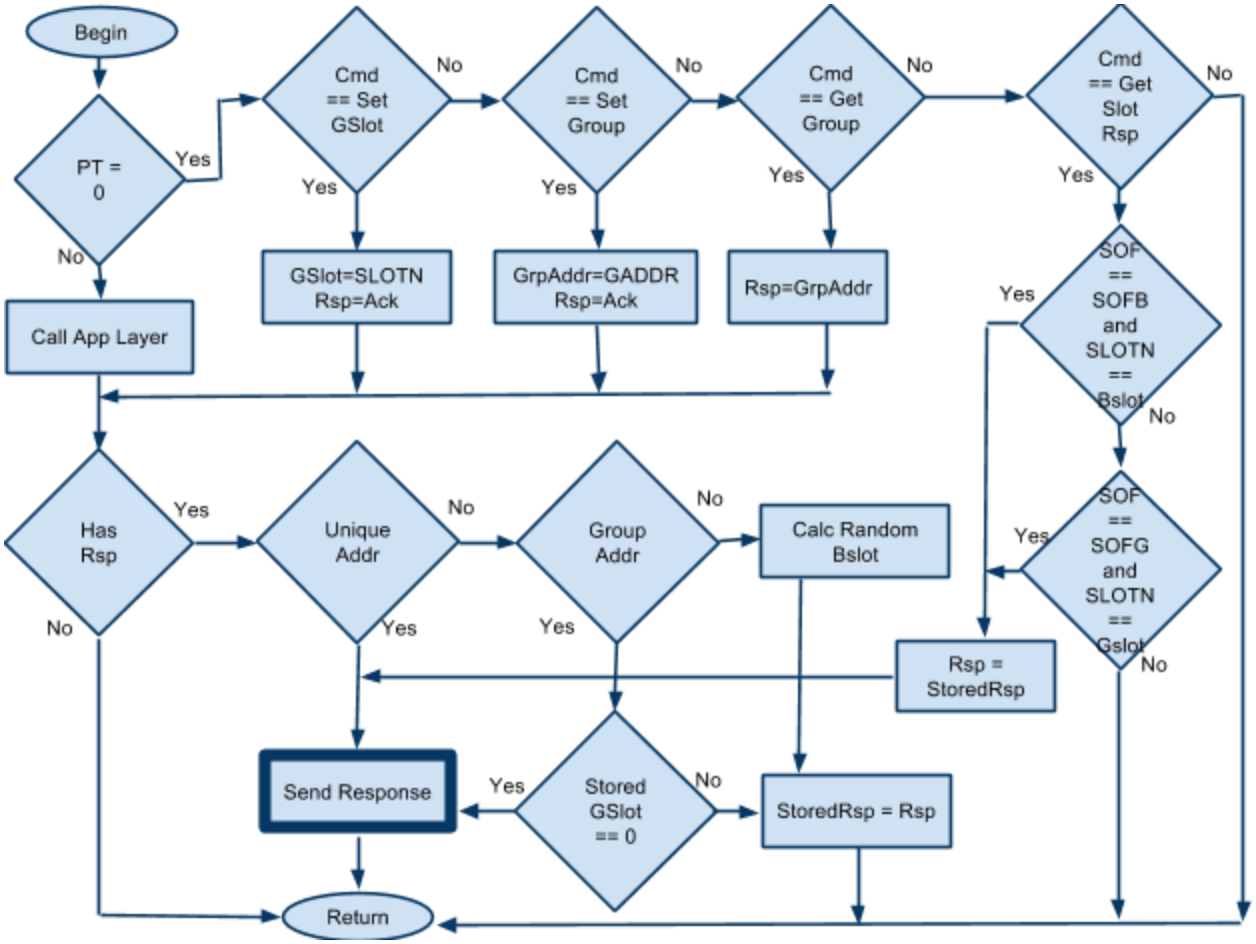


8.8 Process Payload Flow Chart

The Process Payload Flow Chart shows the algorithm for processing the DLL Sub Frame payload (Payload Type/PT=0) and the DLL portion of processing Appliation Layer payloads (Payload Type/PT=1).

The DLL Sub Frame payloads are handled significantly different than the Application Layer payloads. In the DLL Sub Frame only certain commands are allowed with certain addressing modes.

DLL Sub Frame Command	**Broadcast**	**Group Address**	**Unique Address**
Get Slot Response	X	X	
Set Group	X		
Set SlotN			X
Set Group			X


Begin
PT = 0
Yes
No
Call App Layer
Cmd == Set GSlot
Yes
GSlot=SLOTN Rsp=Ack
No
Cmd == Set Group
Yes
GrpAddr=GADDR Rsp=Ack
No
Cmd == Get Group
Yes
Rsp=GrpAddr
No
Cmd == Get Slot Rsp
Yes
SOF == SOFB and SLOTN == Bslot
Yes
No
SOF == SOFG and SLOTN == Gslot
Yes
No
Has Rsp
Yes
No
Unique Addr
Yes
No
Group Addr
Yes
No
Calc Random Bslot
Rsp = StoredRsp
Send Response
Stored GSlot == 0
Yes
No
StoredRsp = Rsp
Return

9 Appendix CRC

The optimum CRC bit size and polynomial is dependant on the message size that the CRC covers. See Koopman http://www.ece.cmu.edu/~koopman/roses/dsn04/koopman04_crc_poly_embedded.pdf.

Number of bits covered by CRC in frame

Description	Min Bits Typical	Max Bits Typical
Data Link Frame Sub Frame	16	17
Group Address Data Link Frame	24	56-64
Unique Address Data Link Frame	48	80-88
Application Frame Cue Schedule (Group Address), 12 cue & 45 cue	408	1465

Of the known 16 bit CRC polynomials described in Koopman, the 0xC86C polynomial is the best choice for the frame lengths in the protocol that need the best error detection (24 - 60 bytes). The polynomial in 'normal' form ix 0x90D9 written a the polynomial (x16 + x15 + x12 + x7 + x6 + x4 + x3 + 1).

The initial value of the register (before computing the CRC) is 0x0000.

Since this version of the protocol is geared toward EIA-485 the CRC algorithm uses LSb first into the register since EIA-485 transmits bytes LSb first.

There is no final XOR of the result.

The CCITT-16, CRC-16 and ANSI-16 polynomials are very poor CRCs for messages of less than 241 bits.

10 Appendix Random Number Generation

11 Revision History

V0.04 (April 21, 2017) Updated revision number.

V0.03 (December 2016)

- Added Appendix on Random Number Generation.
- Added Appendix Revision History.
- Added get/Set Unique Address sub frame commands.