Layout Control Specification	Specification	LCS-9.10.1
CMRInet Protocol	Title	Communications Specification for CMRInet
	Version	1.1 December 2014

Introduction

This Layout Control Specification defines the format of the information sent via the Computer Model Railroad Interface, C/MRI[1], referred to hereafter as CMRInet. This protocol defines the format and rules for sending and receiving data over a network system, in a Master/Slave communication environment. A digital computing device acting as a Master (or Host) communicates to one or more Slave devices, known as Nodes, using the defined CMRInet message protocol. A sequence of 8 bit bytes, called a message, is used to transmit and receive CMRInet information. The content of a message is precisely defined to ensure that the intended instructions can be properly encoded and decoded by the participating devices.

In this specification, the Master will be referred to as the **Host**. The Slave will be referred to as the **Node**.

The C/MRI system was created by Dr. Bruce Chubb in 1985, and introduced to the model railroad community through a 16-part series of articles in Model Railroader magazine. A C/MRI update article was published by MR in 1992. Applications flourished with the C/MRI-based Chubb authored book, “BUILD YOUR OWN UNIVERSAL COMPUTER INTERFACE” published by McGraw-Hill 1989-1997. Significant C/MRI enhancements were conveyed in the 4-part series “SIGNALING MADE EASIER” published in Model Railroader beginning with their January 2004 issue. Subsequent publications, in the form of comprehensive C/MRI User’s Manuals and Railroader’s C/MRI Application Handbooks, continue to advance the protocol with an ever broadening base of C/MRI users. In summary, over the past three decades, multi-thousands of layout control projects using the CMRI protocol have been implemented. These systems use a variety of Host computers, programming languages and employ a variety of operating systems.

The reference document for this specification is - “THE COMPUTER/MODEL RAILROAD INTERFACE (C/MRI) USER'S MANUAL VERSION 3.1 SUPPLEMENT - SERIAL PROTOCOL SUBROUTINES”, By Dr. Bruce A. Chubb, MMR, 2013.

CMRInet Protocol Overview

The CMRInet protocol is designed to operate in a polled, serial communication network environment. A digital computer or equivalent dedicated device operates as the Host. A Node is a CMRInet compatible device, which has assigned unique addresses within the network. The CMRInet protocol is a byte-oriented communication system definition, which provides command and control for model railroad layout devices, such as block occupancy detectors, turnout switch motors, push buttons, toggle switches, trackside signals and panel LEDs.

[1] **The Computer/Model Railroad Interface can be written as C/MRI or CMRI. CMRInet is one word with net in lower case.**

Under software control, the Host polls each active Node on the network, requesting data. The addressed Node responds with data bytes representing the state of the Node's input port bits. Node input ports are typically connected to devices on the layout such as block occupancy detectors or push buttons mounted on fascia panels.

Data bytes transmitted by the Host to a Node will cause the Node to set the state of the output ports. Node output ports are typically connected to controlled devices on the layout, for example, a turnout switch motor or lighting a signal head.

A. CMRInet Message Format

Each message consists of framing characters and data bytes. Framing separates one message from another and defines the type of data, which needs to be processed. CMRInet character framing consists of 10 bits; 1 Start bit, 8 Data bits, 1 Stop bit, asychronous mode.

Three standard ASCII control characters are used in framing the message: STX (start-of-text), ETX (end-of-text), and DLE (data-link-escape). CMRInet adds a non-ASCII character SYN (hex 0xFF, decimal 255) as the message synchronization character.

Software, which implements the CMRInet protocol, creates messages of information to be transmitted. It also inserts special control characters to mark the beginning and end of transmissions, and to define which bytes are addresses and which are pure data.

A CMRInet message consists of a *message header*, *message body*, and *message trailer*. Any sequence of bits not meeting the full specification of this general message format is not, for the purpose of this specification, a valid “message”. Message bytes are 8 binary bits. Protocol characters conform to the American Standard Code for Information Interchange (ASCII) character encoding. There are four types of messages: Initialization, Poll Request, Receive Data, and Transmit Data.

The header consists of two synchronization bytes (SYN), a start-of-text (STX), the addressed Node address, and the message type, which is a single upper case alpha character.

The message body (data) is 0 to 256 data bytes in length.

The message trailer is a single end-of-text (ETX) character. The ETX character signals the end of the message and is used by the software to transition from protocol parsing to message data processing, either in the Host or the Node. For total backward compatibility, the STX/ETX message framing must be preserved.

A properly formed CMRInet protocol message is shown in Figure 1.

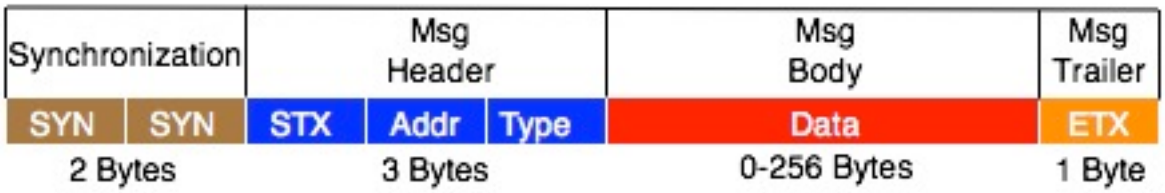

Figure 1 CMRInet Message Format

Synchronization A message starts with two SYN (hex 0xFF, decimal 255) characters. These characters provide data receivers to synchronize the start of a message.

Start-of-Text An ASCII STX (hex 0x02, decimal 2) character defines the start of the message.

Node Address (UA) Node address is in the range of 0 to 127 decimal. A value of 65 decimal is added to the Node address value to create an 8-bit data value outside of the value range of ASCII CONTROL characters.

Message Type (MT) Message Type defines the type of message and is represented by a single UPPER case alpha character. Message types are unique.

Data A sequence of 0 to 256 data bytes representing the body of the message.

End-of-Text An ASCII ETX (hex 0x03, decimal 3) character signifies the end of the message data.

CMRInet Protocol Characters

STX (hex 0x02, decimal 2) – Start of Text
ETX (hex 0x03, decimal 3) – End of Text
DLE (hex 0x10, decimal 16) – Data Link Escape
SYN (hex 0xFF, decimal 255)– Synchronization

Software implementations which process protocol messages may choose to process the message in sections (e.g. header, followed by data, followed by trailer) or parse the protocol characters using software state machine techniques.

Classic C/MRI Nodes (SMINI, SUSIC, USIC) may co-exist in networks with New Technology (NT) Nodes such as the cpNode, by ignoring messages not addressed to them, flushing received characters to the ETX. This is the current implementation in Classic Node software.

B: CMRInet Protocol Sequence

The Host establishes a connection with a Node by first sending an Initialization message with parameters set by the Host. Parameters in the initialization message are used by the Host and Node to negotiate the operating characteristics and parameters of the connection.

Once the Host has made the connection, the Host will then request any input data from the Node by issuing a Poll Request. The addressed Node responds with a Receive Data message containing data bytes representing the state of the Node input port bits.

If the Host has output data for the Node, the Host issues a Transmit Data message with data bytes representing the state to set the Node output port bits. Figure 2 is the message sequence for CMRInet.

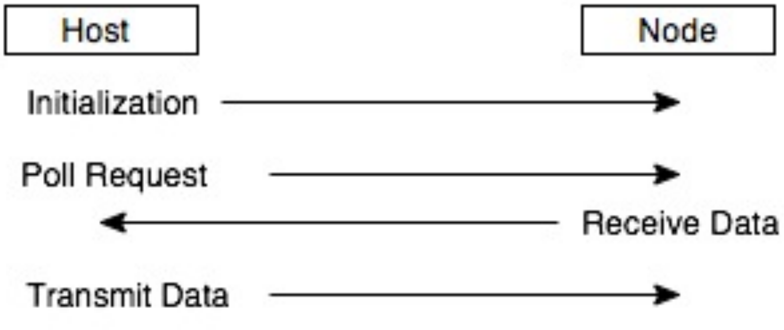

Figure 2 CMRInet Protocol Sequence

C: CMRInet Network Specification

A CMRInet network is defined as four-wire, half-duplex, EIA RS-422/485[7]. The serial communication connection for CMRInet Nodes is through a four-wire cable. With respect to the HOST, one pair of wires is for transmission and one pair for reception. CMRInet Nodes are connected daisy chain fashion, Node to Node.

The supported network speeds for serial communications are 9600, 19200, 28800, 57600, and 115200 bits per second (BPS).

Each CMRInet compatible Node has two connectors for connecting the Node to the network. The onboard connector circuit paths are in parallel on the board. Either connector can be used for input or output. Cables with a shield or drain wire, should have the cable shields tied together, but not tied to any point on the Node.

The connection from the control computer (Host) to CMRInet is through a communication interface device, which converts RS-232 or USB (Universal Serial Bus) signals to RS-422/485. Commercially available interfaces, commonly known as a "dongle", are used where USB ports are available on the Host computer.

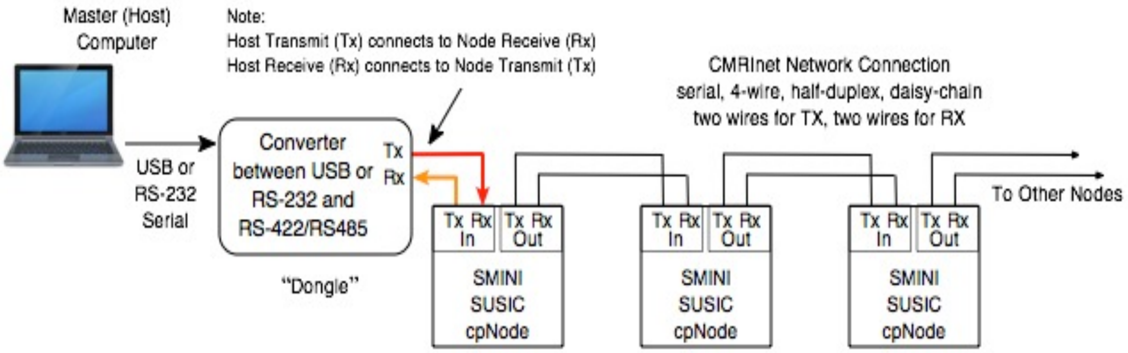
Master (Host) Computer
Note:
Host Transmit (Tx) connects to Node Receive (Rx)
Host Receive (Rx) connects to Node Transmit (Tx)
CMRInet Network Connection
serial, 4-wire, half-duplex, daisy-chain
two wires for TX, two wires for RX
USB or RS-232 Serial
Converter between USB or RS-232 and RS-422/RS485
Tx
Rx
"Dongle"
Tx Rx In
Tx Rx Out
SMINI SUSIC cpNode
To Other Nodes

Figure 3 CMRInet Network Connections

D: CMRInet Messages

It is the intent of this specification, that in order to conform to the CMRInet protocol specification, a computer acting as a Master must encode and decode the contents of the defined messages in conformance with this specification.

A CMRInet message is a series of bytes with each bit of each byte sent in serial order, one bit after another, and each new byte following right behind its predecessor.

Each CMRInet message begins with two bytes of all ones SYN (hex 0xFF, decimal 255) to synchronize the hardware receivers. Next comes an STX (hex 0x02, decimal 2) to indicate that the body of the message begins with the next byte.

The first byte following the STX is the Node address (0-127). An offset of decimal 65 (hex 0x41) is added to the Node address, which forms the Unit Address, known as the UA. The UA maps the transmitted Node address into a range of characters outside of the protocol character values.

Following the Unit address (UA), is the message type (MT), an alpha character defining the type of message. Defined message types are: "I" for Initialization, "P" for Poll Request, "R" for Receive Data, and "T" for Transmit Data.

In a distributed serial communication system, every Node on the network receives each message sent by the Host. A Node first checks the UA contained in the message to see if the message is addressed to it. If there is no address match, the Node discards all the remaining bytes until an ETX is seen. The addressed Node continues to process the message bytes that follow the address byte.

a. Data Link Escape (DLE) Processing

A data byte in the message body can be any binary value, 0 to 255 decimal. Three of those values are CMRInet protocol characters, decimal values 2 (STX), 3 (ETX), and 16 (DLE). Software in the Host and Node use these values in processing a CMRInet message. In order to deal with protocol character values in a data stream, the data-link-escape character (DLE, decimal 16) is used. Protocol management software must insert a DLE in front of any of these data values when forming a Transmit Data or Receive Data message. The receiver, when seeing a DLE in the message body, ignores the DLE and takes the next character in the data stream for processing. This approach provides transmission of pure binary data between Host and Node.

b. Protocol Description Notation

Notations used in the definitions of the protocol messages are as follows:

- Message framing characters are specified using their ASCII mnemonic symbols.
  e.g. `SYN`

- Words and phrases enclosed in < > define protocol values or description of protocol values, text, or options.
  e.g. `<UA> or <I>`

- Hexadecimal values are in the form of 0xhh.
  e.g. `0xCC`

1. Initialization Message

Message Type - "I"
Direction - *HOST to NODE*

Message Format

```
SYN SYN STX <UA> <I> <NDP> <dH><dL> <NS> <CT(1)>...<CT(NS)> ETX
```

Description
The initialization message is sent from the Host to the Node, specifying the Node definition parameter (NDP), which sets specific Node configuration parameters, based on Node type. Each Node on the CMRInet network must have an initialization message sent.

Node Definition Parameters `<NDP>` are defined as:
N - Classic USIC and for SUSIC using 24 bit input/output cards.
X - SUSIC using 32 bit input/output cards.
M - SMINI with fixed 24 inputs and 48 outputs
C - CPNODE with 16 to 144 input/outputs using 8 bit cards.

Transmission Delay, `<dH><dL>`. Each unit of transmission delay represents 10 microseconds (µs) of delay. These two values are set to zero for modern Host computers. A non-zero transmission delay value will cause SMINI, SUSIC and USIC Nodes to delay between transmissions to the Host.

NS (Number of Sets)
For Nodes defined as "X" (SUSIC using 32 bit I/O) or "N" (USIC and SUSIC using 24 bit I/O), **NS** specifies the number of card sets of four plugged into a motherboard

For Nodes defined as “M” (SMINI)
NS specifies the number of output bit pairs used for creating a Yellow signal aspect by oscillating a 2-lead bi-color LED. There are six total output card elements. If NS is zero, no CT() bytes are included in the Initialization message.

Card Element Array CT(1)...CT(NS)
For Nodes defined as “X” (SUSIC using 32 bit I/O) or “N” (USIC and SUSIC using 24 bit I/O) **CT(1)...CT(NS)** specifies bytes, which define input, and output card types and their address locations on a motherboard. The algorithm for computing these values is found in Reference [2] Pages 12-1 through 12-8 and summarized herewith in Table 1.

Table 1. Card-element definition array variables for SUSIC/USIC nodes

Card Set	CT( )	Card Set	CT( )	Card Set	CT( )
OXXX	2	IOOX	41	OIIO	150
IXXX	1	IOIX	25	OIII	86
OOXX	10	IIOX	37	IOOO	169
OIXX	6	IIIX	21	IOOI	105
IOXX	9	OOOO	170	IOIO	153
IIXX	5	OOOI	106	IOII	89
OOOX	42	OOIO	154	IIOO	165
OOIX	26	OOII	90	IIOI	101
OIOX	38	OIOO	166	IIIO	149
OIIX	22	OIOI	102	IIII	85

Note: I = input card, O = output card and X = no card

For Nodes defined as “M” (SMINI)
CT(1)...CT(ns) specifies which output bits are assigned to yellow oscillate 2-lead bi-color LEDs.
The algorithm for computing these values is found in Reference [2] Pages 9-14 through 9-16.

2. *Poll Request*

Message Type - “P”
Direction - *HOST to NODE*

Message Format

```
SYN SYN STX <UA> <P> ETX
```

Description
The Host transmits a Poll Request message to each Node, requesting input data bytes. The Node responds with a Receive Data (R) message containing the number (NI) of input data bytes defined.

The Host waits for a response from the Node for a specified time, known as the timeout interval. If a Receive Data message is not received from the polled addressed Node within the timeout interval, the Host software handles a timeout error, and the next Node in the poll list is polled.

3. Receive Data

Message Type - "R"
Direction - *NODE to HOST*

Message Format

```
   SYN SYN STX <UA> <R> <IB(1)>...<IB(NI)> ETX
```

Description
The Node transmits a Receive Data message in direct response to a Poll (P) message from the Host. The number of bytes sent, NI, is the total number of input bytes. Data Link Escape (DLE) processing must be applied to the message data.

4. Transmit Data

Message Type - "T"
Direction - *HOST to NODE*

Message Format

```
SYN SYN STX <UA> <T> <OB(1)>...<OB(NO)> ETX
```

Description
The Host transmits a Transmit Data message to set output card byte values. The number of bytes sent, NO, is the total number of output card bytes. Data Link Escape (DLE) processing must be applied to the message data.

D: CMRInet Message Summary

Initialization

```
    SYN SYN STX <UA><I> <NDP><dH><dL><NS><CT(1)>...<CT(NS)> ETX
```

Poll Request

```
    SYN SYN STX <UA><P> ETX
```

Receive Data

```
    SYN SYN STX <UA><R> <IB(1)>...<IB(NI)> ETX
```

Transmit Data

```
    SYN SYN STX <UA><T> <OB(1)>...<OB(NO)> ETX
```

E: References

[1] “THE COMPUTER/MODEL RAILROAD INTERFACE (C/MRI) USER'S MANUAL VERSION 3.1 SUPPLEMENT - SERIAL PROTOCOL SUBROUTINES”, By Dr. Bruce A. Chubb, MMR, 2013

[2] “THE COMPUTER/MODEL RAILROAD INTERFACE (C/MRI) USER’S MANUAL VERSION 3.1” by Dr. Bruce A. Chubb, MMR ©2013 Bruce A. Chubb

[3] “BUILD YOUR OWN UNIVERSAL COMPUTER INTERFACE, SECOND EDITION” Bruce A. Chubb, McGraw-Hill ©1989, 1997

[4] “SIGNALING MADE EASIER” Bruce Chubb, Model Railroader, January – April 2004

[5] “RAILROADER'S C/MRI APPLICATIONS HANDBOOK VOLUME 1 - SYSTEM EXTENSIONS VERSION 3.0” by Dr. Bruce A. Chubb, MMR ©2010 Bruce A. Chubb

[6] “RAILROADER'S C/MRI APPLICATIONS HANDBOOK VOLUME 2 - SIGNALING SYSTEMS VERSION 3.0” by Dr. Bruce A. Chubb, MMR ©2010 Bruce A. Chubb

[7] “RS-422 AND RS-485 STANDARDS OVERVIEW AND SYSTEM CONFIGURATIONS”, SLLA070D–June 2002–Revised May 2010 Copyright © 2002–2010, Texas Instruments Incorporated