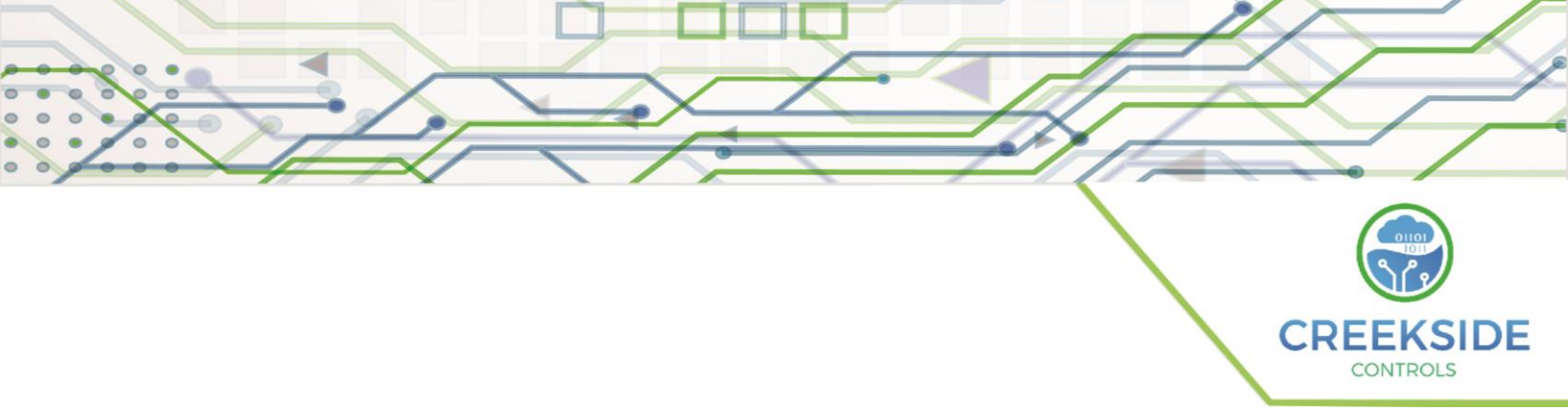

CREEKSIDE
CONTROLS

Reference Manual
DM-719-0

Creekside Serial Packet Protocol

Table of Contents

Scope

This document describes packet and messaging protocol used for serial communication in Creekside projects

General Notes

Serial packet / message delivery is not guaranteed. It can be added at a higher level if desired.

All messages are routed immediately and with no priority.

Definitions

- Node - a location that can send and receive serial data. Each node has a unique address used to determine location.
- Message - a specific set of data with a header that defines specifics of the data
- Message Header - Information attached to the message used to identify and route the data.
- Packet - the serial form of a Message being sent over UART
- Data Payload - The data inside a message.
- Node Address - This is a hard-coded address is assigned to each node by the Message Router
- Message Router - a software routine inside the system which handles the incoming and outgoing data to all Nodes.
- Route - a path for a message to follow from one Node to another. Also, a verb for the act of sending a message from one Node to another.

- Direct Message - A message whose destination is a specific Node
- Broadcast Message - A message whose destination is not specified and is determined using the Routing Table.
- Message Type - The Message Header includes an enumeration for the type of message. This can be used to with the Routing Table to Route certain types to specific Nodes.

Message Structure

The following table defines the structure of a message in the order presented. The highest component in the table is the first seen in the message.

Component	Length	
Message Length	1 byte	The length of the entire message including the header and data payload in bytes. Maximum message size is 255 bytes.
Source Node Address	1 byte	Shows where the message came from.
Destination Node Address	1 byte	Shows where the message is going.
Message Type	1 byte	An enumeration used to show what type of information is present in the message.
Data Payload	n bytes	The data payload of the message. Length of payload is determined by message length subtracted by 4 for the header bytes.

Table 1 – Message Structure

Node Addressing

Node addresses are predefined or assigned by a master at startup. The 0xFF node addresses is reserved. It is used to designate a broadcast message.

Message Type

Message Types are defined per project. It is up to the application to decide how best to organize these.

Packet Structure

All *Messages* are encoded as *Packets* before being sent over a UART. The following table defines the structure of a *Packet* sent over a UART. The designation of *Message* is used to refer to the formatted bytes used by the application firmware. The designation of *Packet* is used to refer to the same message in its serial UART form.

Component	Length	Description
Start Character	1 byte	Always 0xCB
Message	n bytes	Entire *Message*, as defined in above section
CRC	2 bytes	CRC provides bit error checking of the packet
End Character	1 byte	Always 0xCE

Table 2 - Packet Structure

Start, End and Escape Characters

The start and end characters are always 0xCB and 0xCE, respectively. However, it may be that some bytes within the message may need to be one of these values. To accomplish this, an escape character is used. The escape character is 0xC9. Every time an escape character is seen in a message, the escape character is thrown away, and the next character gets 1 added to its value. Escape characters and their following characters are still used in CRC calculation.

The escape character is used whenever the start character '0xCB', the end character '0xCE' or the escape character '0xC9' need to be in a message. They will look like this:

Message Byte	will turn into message Bytes
0xCB	0xC9, 0xCA
0xCE	0xC9, 0xCD
0xC9	0xC9, 0xC8

Table 3 - Escape character examples

For instance, this message has a command ID of 0xCB:

{0x80, 0x04, 0x01, 0xD0, 0xCB, 0x00…}

Which will be translated into a packet that looks like this:

{0xCB, 0x80, 0x04, 0x01, 0xD0, 0xC9, 0xCA, 0x00…}

CRC

A 16-bit CRC is performed on all UART messages. The CRC type is identical to the CRC-16 method used by Modbus.

Revision History

Rev	Date	Changes
0	7/2/2019	Initial Release
1	7/12/2019	Fixed typo with message data length and added data length calculation

IMPORTANT NOTICE – PLEASE READ CAREFULLY

Creekside Controls and its subsidiaries ("Creekside") reserve the right to make changes, corrections, enhancements, modifications, and improvements to their products and/or this document at any time without notice. Purchasers should obtain the latest relevant information on Creekside products before placing orders. Creekside products are sold pursuant to Creekside's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of Creekside products and Creekside assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by Creekside herein.

Resale of Creekside products with provisions different from the information set forth herein shall void any warranty granted by Creekside for such product.

Creekside and the Creekside and StreamUX logo are trademarks of Creekside. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.