SERIAL PROTOCOL

DATA STRUCTURE

This serial protocol is a binary, packet-based serial communication protocol for master to slave topologies.

Communication is always initiated by the master (controlling system) with a control/request packet, followed by a slave reply/data packet.

Both host and slave use the same binary packet format, illustrated in figure 1. The packet consists of:

1. Start Byte (0x2B)
2. Destination Address (1 byte)
3. Source Address (1 byte)
4. Packet type identifier (1 byte)
5. Data bytes
6. Checksum (calculated on 2, 3, 4 and 5)
7. End Byte (0x2D)

Sections 2-6 of the packet require byte stuffing using escape byte 0x23 to hide bytes 0x2B and 0x2D within the data. At the transmitter, bytes 0x2B, 0x2D and 0x23 found in these sections are preceded by the 0x23 escape byte which is discarded at the receiver.

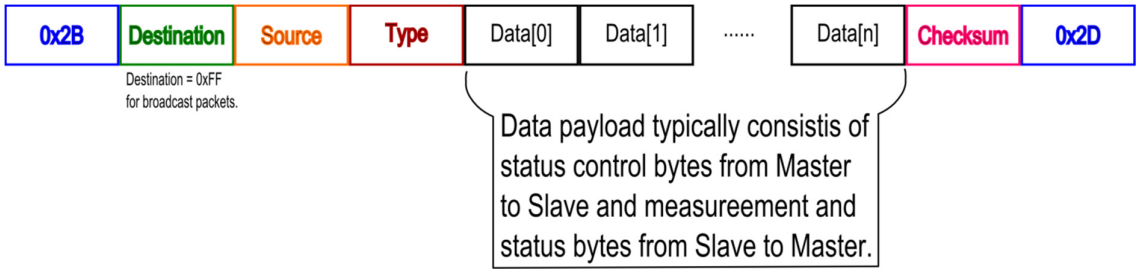

Figure 1: Packet Format

ADDRESSING

The serial protocol includes a device address. This is intended for RS485 based systems with multiple slave devices. In this case of a single slave topology, the address is not required but is maintained for standardization of the serial protocol.

BYTE STUFFING

Sections 2-6 of the packet require byte stuffing using escape byte 0x23 to hide bytes 0x2B and 0x2D within the data. At the transmitter, bytes 0x2B, 0x2D and 0x23 found in these sections are preceded by the 0x23 escape byte which is discarded at the receiver.

Byte Stuffing is performed after checksum calculation.

Example:

Original data bytes: **0x02 0x34 0x2B 0x12 0x23 0x43**

Transmitted bytes: **0x02 0x34 0x23 0x2B 0x12 0x23 0x23 0x43**

CHECKSUM

The 1 byte checksum is calculated on sections 2 to 5 of the packet as follows:

1. Checksum initialized to 0
2. Checksum value XORed with all bytes in sequence from the first to the last byte
3. Checksum value set to 255 minus the checksum value in (2)

Example algorithm in C:

```
uint8_t data_packet[N];
uint8_t checksum = 0;

      for(uint8_t i = 0; i<N; i++)
      {
            checksum ^= data_packet[i];
      }

      checksum = (255 - checksum);
```