OPI Link Protocol

Version	Date	Author	Comments
1.00	20130503	mpeng	1. Original

Introduction

This document defines the protocol used for sending byte data through serially streamed asynchronous links. It provides a mechanism for a transmitter and receiver to exchange data asynchronously and faithfully. The major advantage of using this kind of link is that it is tolerant to data corruption and byte loss through the link. The disadvantage of this kind of link is that packets can be corrupted and thus will be discarded since there is no mechanism for recovery.

Packet Structure

The transmitter sends packets in this protocol through the asynchronous links with specific byte order definition.

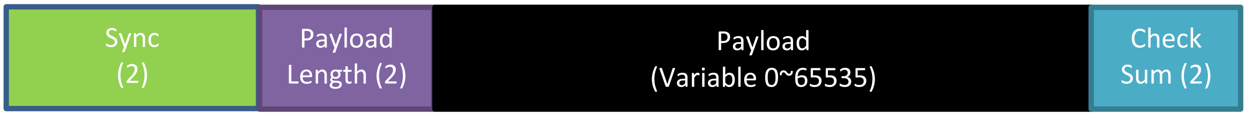

Figure 1:Packet structure with field size in bytes.

1. Sync (2 bytes) – The sync is two bytes, both with value 0x33 that denotes the beginning of a packet. When the receiver receives these bytes, it can take appropriate action and receive the packet. While it is possible that two consecutive 0x33 bytes may be in the payload and the receiver might interpret them as sync bytes, the combination of the payload length and checksum will ensure that mis-interpreted packet is discarded.
2. Payload Length (2 byte) – The length in bytes of the payload is specified in this field. The minimum payload size is 0 and the maximum payload size is 65535. Practically, payload size is never larger than ~1k bytes due to RAM limitations in embedded systems. The high byte and low bytes are sent in Big Endian format. That is high byte first, and then low byte.
3. Payload (Variable Length) – The payload contains the data being transmitted. It can contain frames that can be specified by another protocol.
4. Check Sum (2 byte) – This check sum is derived only from the payload data. The bytes of the payload are summed and the lower 16 bits of the sum are used as the checksum. The high byte and low bytes are sent in Big Endian format. That is high byte first, and then low byte.

The maximum valid packet length is 2+2+65535+2 = 65541 bytes. Packet sizes of greater than 1k bytes are not really practical.