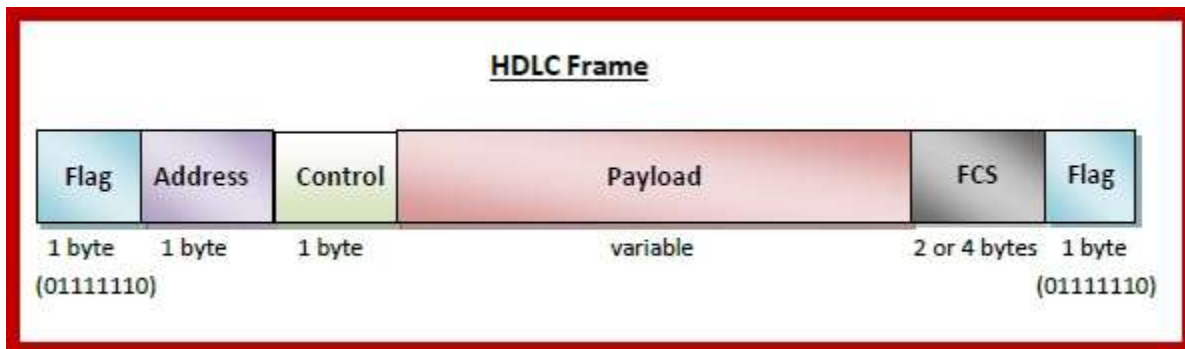


# HDLC Package

## HDLC Frame

HDLC is a bit - oriented protocol where each frame contains up to six fields. The structure varies according to the type of frame. The fields of a HDLC frame are –

- **Flag** – It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address** – It contains the address of the receiver. If the frame is sent by the primary station, it contains the address (es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.
- **Control** – It is 1 or 2 bytes containing flow and error control information.
- **Payload** – This carries the data from the network layer. Its length may vary from one network to another.
- **FCS** – It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)



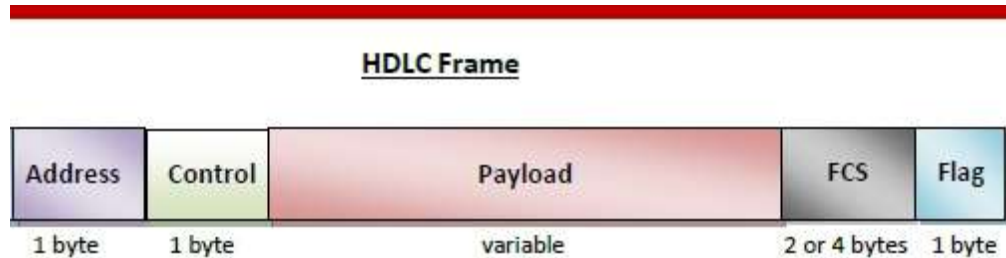
## Usage of this Package:

After Acquiring data from Serial Port or any DIO card on Host make sure data is in bit oriented format as it is requirement of HDLC.

1. If data is Synchronous which means if external clock input is available, then place [HDLC] Synchronous Acquire.vi after acquiring data from any DIO card make sure clock and Data is in Boolean format.  
*Synchronous Serial Receiver cards can also be used for this process.*
2. If data is Asynchronous which means Transmission and reception is accomplished independently on each channel with specific baud rate, then sample it at that baud rate.
3. After Sampling Data extract message with in flag, place [HDLC] Flag Check.vi and specify minimum length of message in order to make sure no consecutive Flag detected.  
*Currently this is configured for detecting (01111110) 0x7E Pattern. It can be configured for any other flag.*
4. Next Step is bit stuffing which occurs if data is in Synchronous mode. Flag has a unique binary pattern and, in order to avoid that the data field to transmit contains the same pattern, "zero-bit insertion" technique is used. [HDLC] Zero Bit Stuffing.vi VI does both Receiving and

transmitting part. Place this VI in transmitting chain after complete message created including all four fields of HDLC and Wire False to "Read Operation(T)" button. For Reception After extracting message within flag (by using [HDLC] Flag Check.vi) place this VI and wire True to "Read Operation(T)" button.

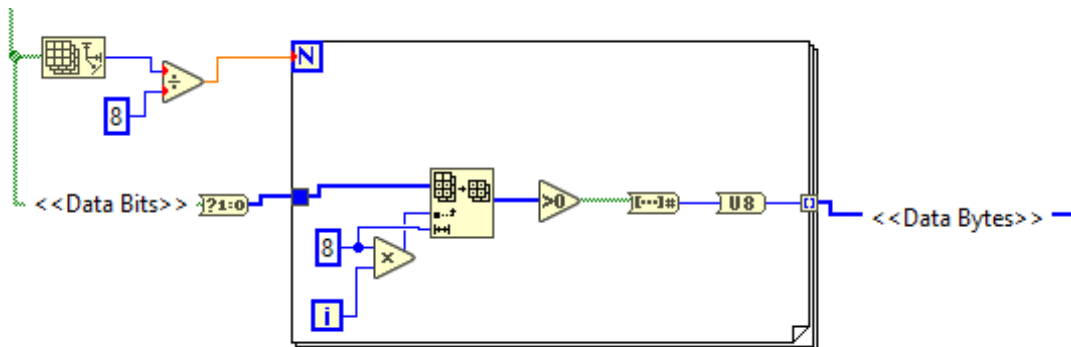
Now extracted data is in this format



**FCS:**

Get Generic CRC calculation VI from this [link](#) (from [Matthew Kelton](#)) specify desire polynomial and other parameters. Input to this VI should be in Byte Format and it should include Address Control and Payload Field not the FCS field.

For Conversion of bit to byte format refer to this snippet



Link: <http://forums.ni.com/t5/LabVIEW/Computing-CRC/m-p/825331#M375052>