ComIf

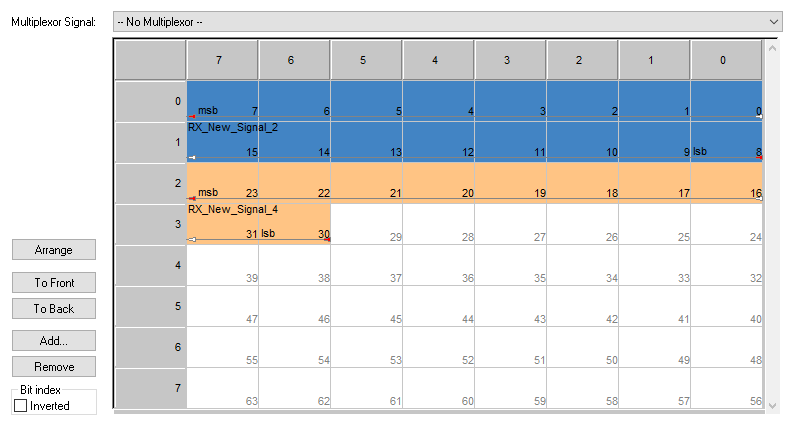
Communication Interface Design

# Frames

To be updated.

# Signals

Signals are just the representation of values (bits or bytes) within the frame payload.



## Signal Properties

### Name

Name of the Signal adhering to the C Variable naming convention. This is a mandatory parameter

### Datatype & Length

Signals can range between 1 bit up-to the entire payload of the frame. However, the data type of the signals are confined to one of the below: int8, uint8, int16, uint16, int32, uint32, uint8[].

* Any signal with any bit length shall select the next possible data type. Example: 1 to 8-bit signals shall choose int8/uint8, 9 to 16 bit signals shall choose int16/uint16, etc.
* If the data length is less than the confined data type, the masking of data will be handled in the ComIf code itself.
* Any signal greater than 32 bits should be mentioned as uint8[] array data type.
* A signal with signed data type with length less than the confined data types, the most significant bit will be considered as the signed bit and conversion will to and fro will be handled. So, the data beyond the length – 1 bits will be masked out and if the value is –ve, then the MSB bit will be set.

Length property differs based on Data Type selected. If the data type is singular, then this parameter is defined in bits. However, if the data type is uint8[], then this parameter represents number of bytes in the payload.

### Start Bit Position

Start bit of the signal in the Frame. Since ComIf is being used in serial communication protocols, it must be noted that the ComIf ONLY support Big Endian Bit-Endianness in the ComIf output serial data, which means the Most-Significant-Bits of the signals will be sent first as depicted in the image above.

The start bit position is the position of the MSB of signal in the Message payload. For example: in the image above, the start bit position of Rx\_NEW\_Signal\_2 is 7 and the start bit position of RX\_New\_Signal\_1 is 23.

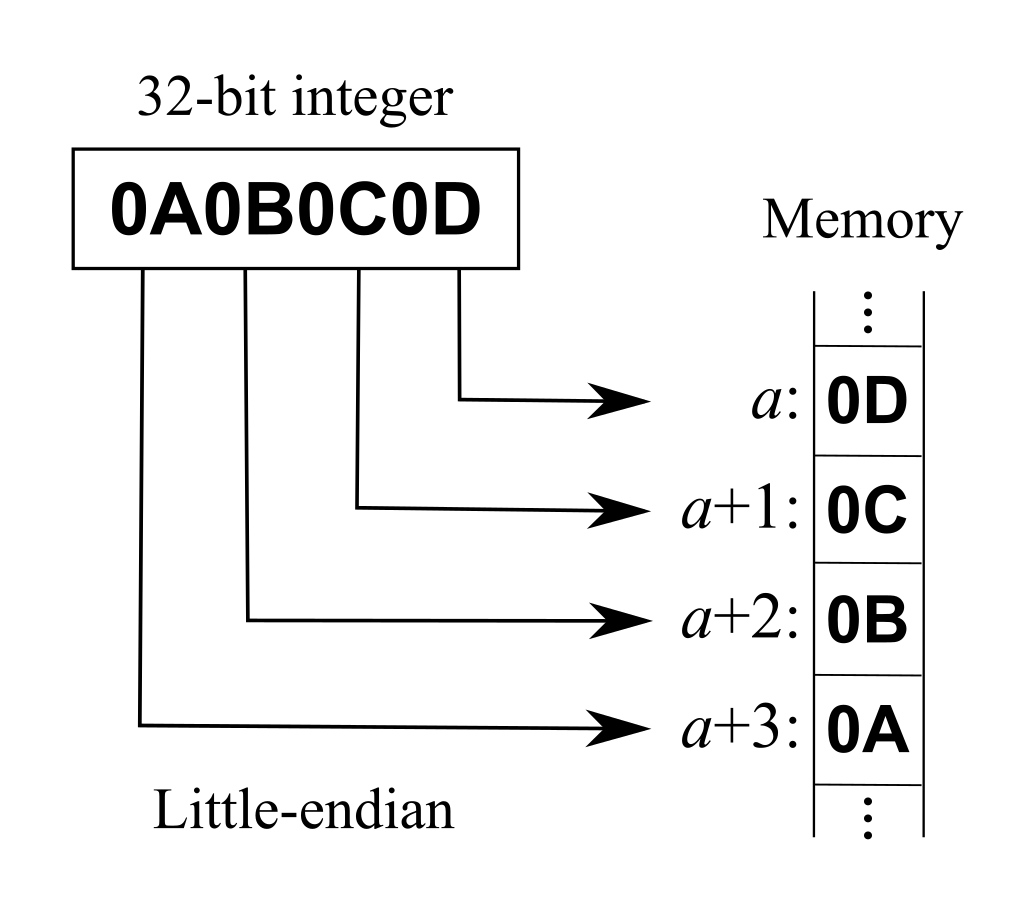
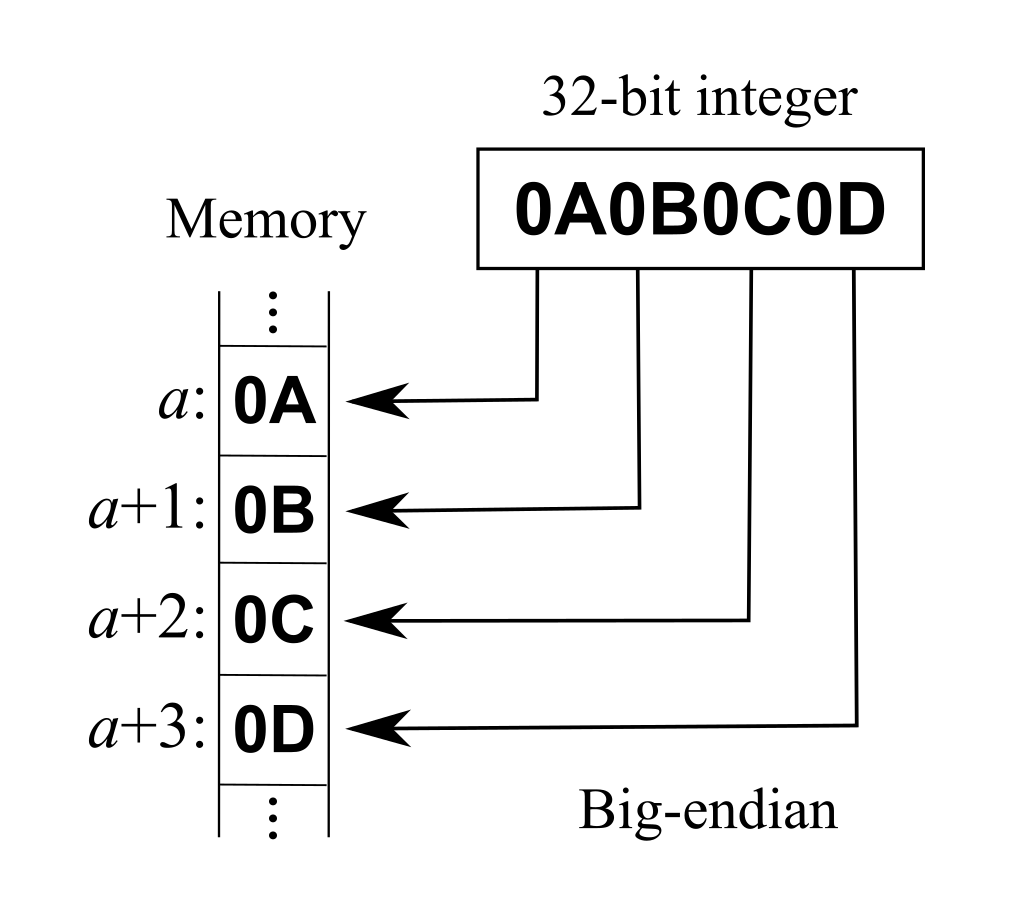
So the signals in the serial data will be:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Byte 0 | | | | | | | | Byte 1 | | | | | | | | Byte 2 | | | | | | | | Byte 3 | | | | | | | | Byte 4 | | | | | | | | … | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RX\_NEW\_Signal\_2 | | | | | | | | | | | | | | | | RX\_New\_Signal\_1 | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### Endianness

This is applicable for data conversion from the serial input to the global variable representation in the software code.

The parameter shall either be Big-Endian or Little-Endian. The difference is depicted below:



Since the message data is always in Big endian format, this shall be represented based on the system which this information is processed, like the micro-controller or compiler etc.

Since there are multiple data types being handled in the ComIf signals, the APIs for transmit or receive the signals will contain the void pointer data type only. So, this parameter plays a significant role in representing the data and vice versa.

If the signal is configured as Big-Endian, then the content in the pointer will be considered as MSB byte, content in the pointer + 1 is considered as the next byte and so on up-to LSB. If the signal is configured as Little-Endian, then vice-versa.

### Tx Transfer Property

Tx Transfer property is used to trigger the transmission when the signal is updated. Below are the possible values:

|  |  |
| --- | --- |
| Values | Comment |
| None | Does nothing |
| OnChange | Triggers the transmission of the frame on value change |
| OnUpdate | Triggers the transmission of the frame on every time the signal is being called |

### Rx Notification

Rx Notification Call-back function which will be called when the new signal is received.