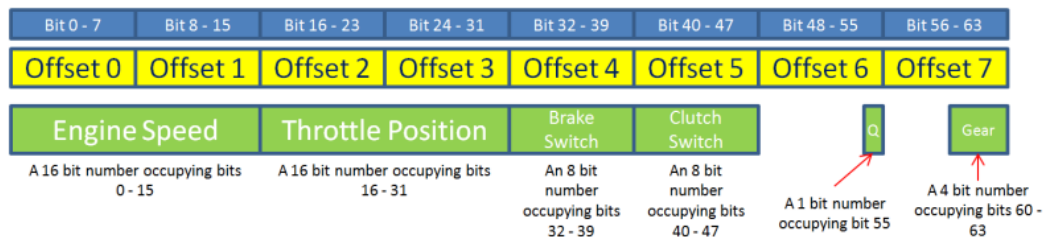


How we generate the final CAN message is this:

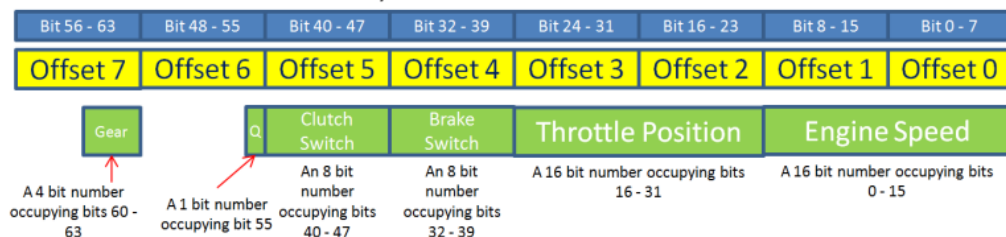
- For normal messaging, the buffer is populated from bit 0 to bit 63 as per normal practice, ie. if you want a two byte value at offset 2 and 3 you put it at bit 16 and it loads from bit 16 to bit 31.
- For normal messaging, when you transmit the buffer, it reads out each BYTE in order from bit 0 to bit 56. In other words, Byte 0, then Byte 1, then Byte 2, up to Byte 7.
- For little endian messaging, when you transmit the buffer, it reads out each BYTE but it starts from Byte 7 and goes backwards from there: Byte 7, byte 6, byte 5, down to byte 0.

CAN Buffer is loaded:



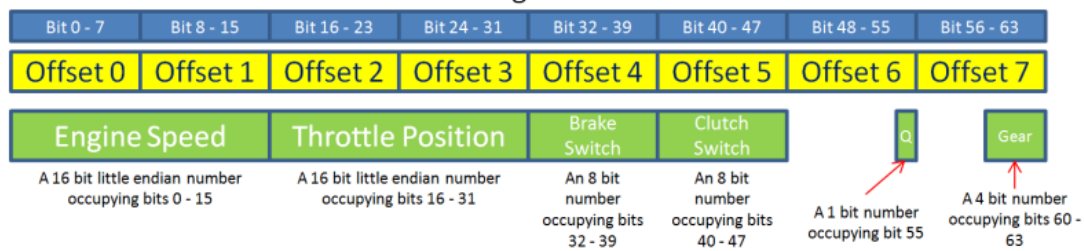
Normal transmission: exactly as above

Little endian transmission: byte order reversed



In the example above, if you want little endian throttle position to appear in the final message from bits 16 to 31 (ie in the exact same position as the first example) , you need to load the buffer from bits 32 to 47. This graphic shows how to load the buffer:

Little endian transmission :CAN message as transmitted



Little endian transmission: buffer is loaded as follows

