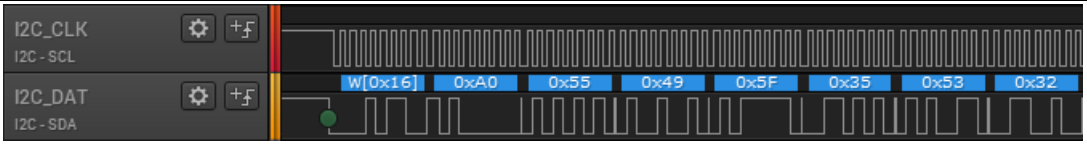
UI I2C Firmware Update Process

1. **Read first 32 bytes of FW bin file and send with I2C command 0xA0.**

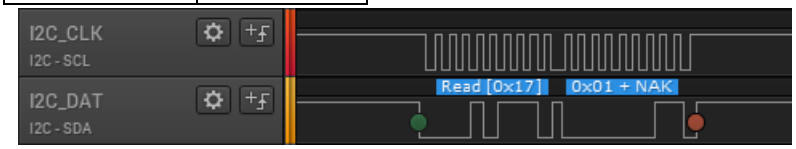
Address(W)	Command	Byte 0	Byte 1	…	Byte 31
Write[0x16]	0xA0	FW data	FW data	…	FW data



BMS will verify the information and jump to bootloader if this is valid FW bin file.

2. **Delay around 100ms**
3. **Read 1 byte which indicates status**

Address(R)	Data
Read[0x17]	Status byte



0x01: BMS already jumped to BSL, ready for next step

0x00: Command not received.

0xE0: Incorrect bin file.

0xE1: Incorrect MCU type.

4. **Read next 32 bytes of bin file and send with command 0xA1 and packet number, start from 0x01.**

Address(W)	CMD	Packet Number (high byte)	Packet Number (Low byte)	Byte 0	…	Byte 31	CRC Byte
Write[0x16]	0xA1	0x00	0x01	FW data	…	FW data	CRC8

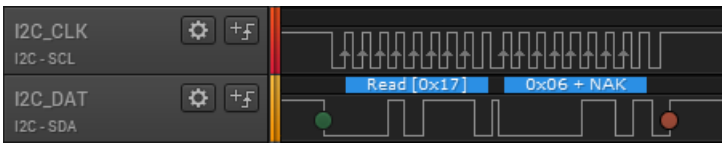
Packet number: from 0x01 ~ 0x180

CRC8: Calculate from address byte to FW byte 31, total 36 bytes

(CRC initial value: 0x00, poly: 0x07)

5. **Read status**

Address(R)	Data
Read[0x17]	Status byte



0x06: ACK, continue next packet.

0xE2: CRC error

0xE3: Packet number out of range, packet number should be 0x01 to 0x180.

0xE4: Wrong packet number, packet number must be transmitted in order.

※ <u>If BMS does not return ACK, need to re-start update process from packet 1.</u>

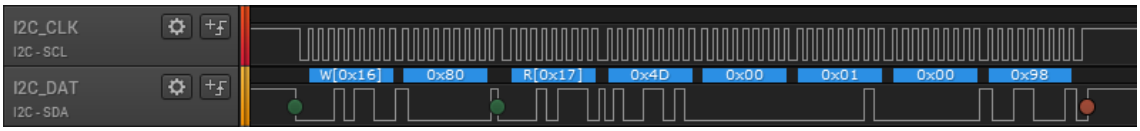
6. **Continue to send remaining FW data as step 4 and 5 until end of bin file, last packet number should be 0x180.**

7. **Send finish command 0xA2 with data 0x00.**

Address(W)	Command	Data
Write[0x16]	0xA2	0x00

8. **BMS will jump to main code after receiving last packet and 0xA2 command, can use command 0x80 to read BMS FW version to confirm it's already running in main code:**

Address(W)	Command	Address(R)	Byte 0	Byte 1	Byte 2	Byte 3	CRC
Write[0x16]	0x80	Read[0x17]	0x4D	0x00	0x01	0x00	0x98



Byte 0: 0x4D('M') indicates in main code, 0x42('B') indicates in BSL code.

Byte 1: Major version

Byte 2: Minor version

Byte 3: Test version

CRC byte: Calculate from address(w) byte to byte 3.