UI 5S2P BMS FW Functionalities (07/14)

1. Power Modes

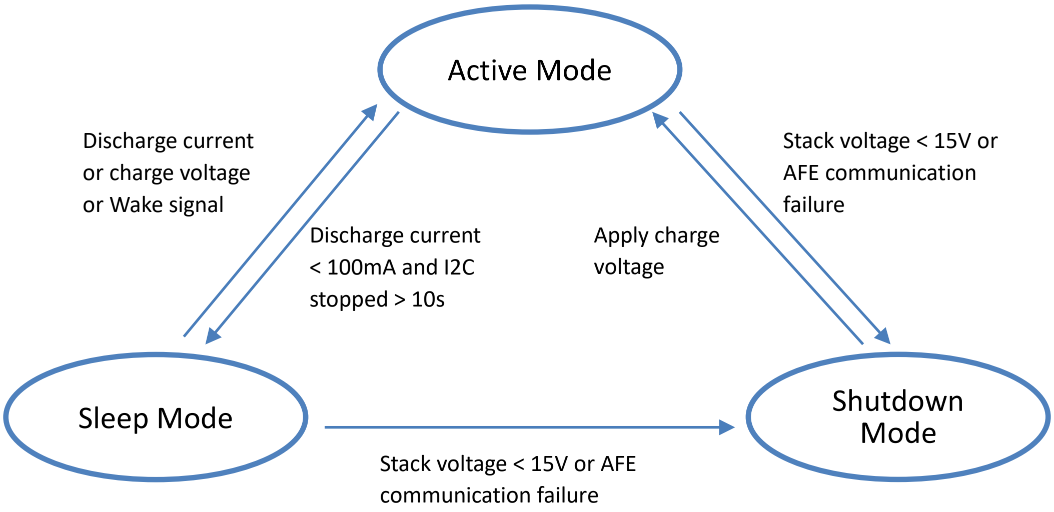
1.1. Active Mode: BMS is fully functional, monitor battery voltage, current, temperature and cell voltages, calculates RSOC, check protections to enable/disable MOSFET, controls charge process and handles host I2C commands.

1.2. Sleep Mode: BMS sleeps and wakes up periodically to reads stack voltage, current and input voltage. Host I2C communication is disabled at this state.

- **Enter:** Discharge current < 100mA and host I2C communication stopped for over 10 seconds.
- **Exit:** Charge voltage is applied, discharge current or WAKE pin low signal is detected

1.3. Shutdown Mode: BMS shuts down AFE and disable it's VCC to achieve lowest power consumption

- **Enter:** Stack voltage is lower than 15V, or AFE communication failure for over 10 seconds.
- **Exit:** Apply charge voltage.



2. I2C Communication

2.1. I2C 7-bits slave address: 0x0B, frequency up to 100kHz.

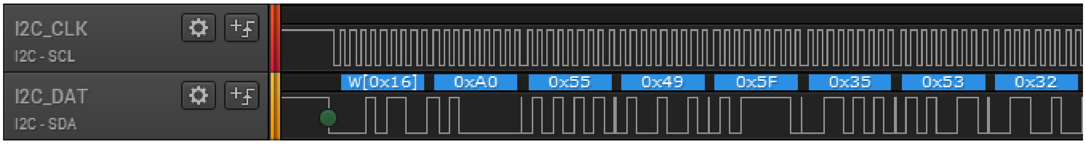
2.2. Command list:

Item	Command	Description	Unit
Temperature	0x08	Read word, unsigned int.	0.1K
Stack voltage	0x09	Read word, unsigned int.	10mV
Current	0x0A	Read word, signed int.	10mA
RSOC	0x0D	Read word, unsigned int.	%
Battery Status	0x16	Read word. Bit 4 : Fully Discharged Bit 5 : Fully Charged Bit 7 : Initialized	Hex
Cell 1 Voltage	0x31	Read word, unsigned int.	mV
Cell 2 Voltage	0x32	Read word, unsigned int.	mV
Cell 3 Voltage	0x33	Read word, unsigned int.	mV
Cell 4 Voltage	0x34	Read word, unsigned int.	mV
Cell 5 Voltage	0x35	Read word, unsigned int.	mV
FW Version	0x80	Read 4-Bytes: **Byte 0**: 0x4D(‘M’) means running in main code, 0x42(‘B’) means in BSL. **Byte 1**: Major version **Byte 2**: Minor version **Byte 3**: Test version	Hex
FW Update Start	0xA0	Write with first 32-Bytes of FW data.	Hex
FW Packet	0xA1	Write with packet number and 32-Bytes FW data.	Hex
FW Update Finish	0xA2	Write with data 0x00.	Hex

3. I2C Firmware Update Process

3.1. 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

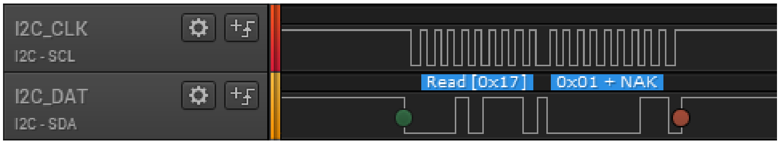


BMS will verify the information and jump to bootloader if this is valid FW bin file.

3.2. Delay around 100ms

3.3. Read 1 byte which indicates status

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



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

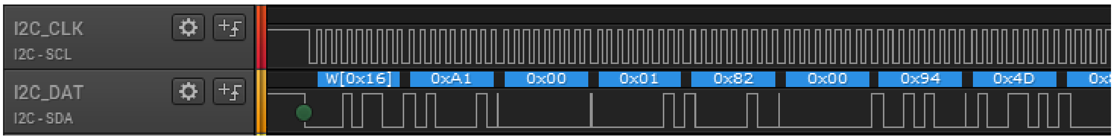
0x00: Command not received.

0xE0: Incorrect bin file.

0xE1: Incorrect MCU type.

3.4. 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

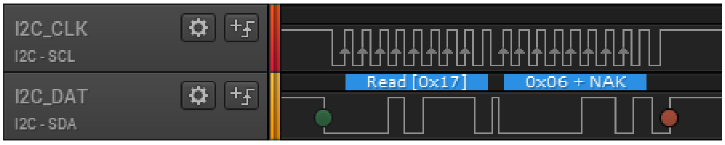
(CRC initial value: 0x00, poly: 0x07)

For example, the CRC8 of packet 16 A1 00 01 82 00 94 4D 82 00 B0 8C 82 00 B0 8C 82 00 99 B4 82 00 B0 8C 82 00 B0 8C 82 00 B0 8C 82 00 99 AF

is 0x6F

3.5. Read status

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



0x06: ACK, continue next packet.

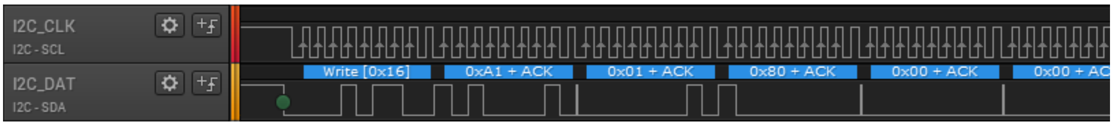
0xE2: CRC error

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

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

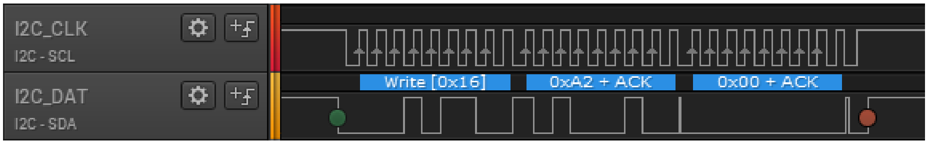
※ If BMS does not return ACK, need to re-start update process from packet 1.

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



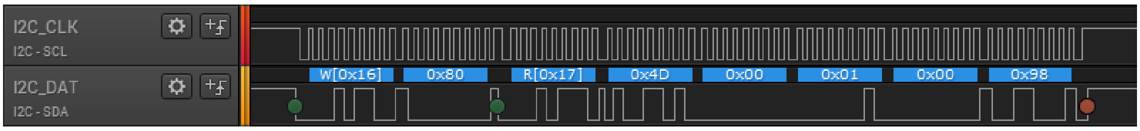
3.7. Send finish command 0xA2 with data 0x00.

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



3.8. 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.