

# **3S Gauge**

# **User Application Note**

## Catalog

<b>CATALOG .....</b>	<b>2</b>
<b>1. INTRODUCTION .....</b>	<b>3</b>
<b>2.HW HARDWARE .....</b>	<b>4</b>
<b>2.1 EVM.....</b>	<b>4</b>
<b>2.2 COMMUNICATION BOARD .....</b>	<b>5</b>
<b>3. HARDWARE CONNECTION INSTRUCTIONS.....</b>	<b>6</b>
<b>4.SOFTWARE GUI.....</b>	<b>7</b>
<b>4.1 CONNECTION.....</b>	<b>7</b>
<b>4.2 CELL INFORMATION READING.....</b>	<b>8</b>
<b>5. CALIBRATION INSTRUCTIONS AND PROCEDURES.....</b>	<b>10</b>
<b>5.1 CONNECTION.....</b>	<b>10</b>
<b>5.2 SOFTWARE GUI CALIBRATION PROCESS .....</b>	<b>11</b>
<b>5.3 COMMUNICATION PROTOCOL (I2C).....</b>	<b>16</b>
<b>5.3 FLOWCHART .....</b>	<b>18</b>
<b>6. MCU PARAMETER SPACE.....</b>	<b>21</b>
<b>6.1 CONNECTION.....</b>	<b>21</b>
<b>6.2 COMMUNICATION PROTOCOL (I2C).....</b>	<b>21</b>
<b>6.3 FLOWCHART .....</b>	<b>21</b>

## **1. Introduction**

3-string gauge solution includes EVM hardware, graphical user interface (GUI), and communication boards for communication to computer.

Through the functions of the GUI software, users can obtain the running status of the batteries connected to the hardware board through the communication board, and adjust the best system performance through appropriate data processing and analysis, optimal parameter configuration, and experimentation. This article will illustrate the overall circuit connection of the solution and the basic operation functions of the GUI.

Hardware EVM combined with software GUIs can provide product development including, power tools, such as electric drills / electric boarders / saber saws, etc., cordless vacuum cleaners, lights, waxing machines, distance measuring instruments, car rescue mobile power, etc. 3 series lithium battery protection boards.

## 2.HW hardware

### 2.1 EVM

TH251A EVM is shown in the figure below and the main interface interfaces are:

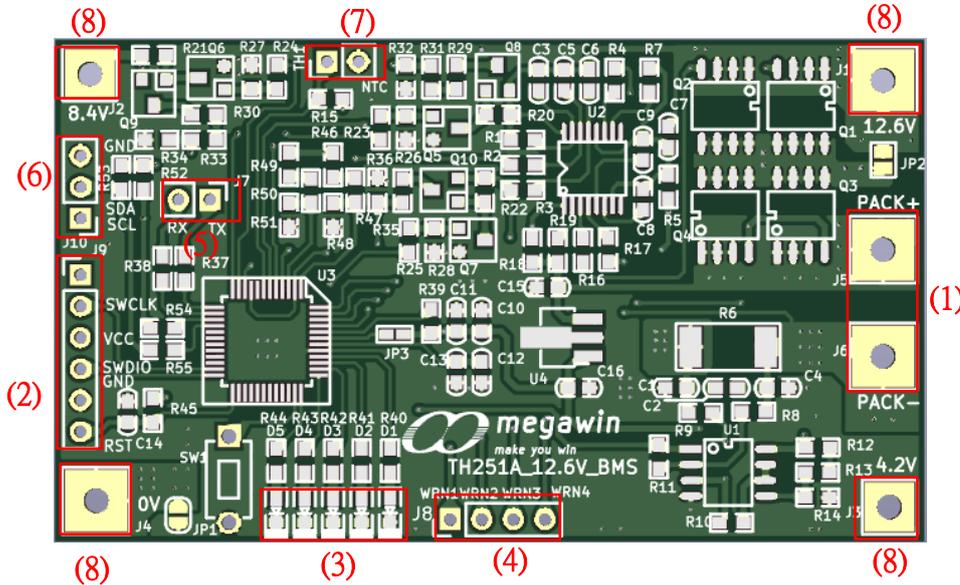


Figure 2-1 MSE03GM1AD48 for 3S Gauge EVM introduction

- (1) Charge and Discharge Port : Charging and discharging with the same hole interface, connected to the power supply to charge the battery cell, connected to the load to do the power output behavior.
- (2) Sequence debugging interface: Used by system development, e.g. to update firmware.
- (3) Gauge and health status indicator: Provides a reminder of the remaining capacity of the battery cell in use and the health status of the battery cell if the voltage difference between the cells is too large.
- (4) Warning status interface: Provides IO outputs for power (charge full, discharge empty), usage behavior (charge, discharge), and status (over-voltage, under-voltage, over-temperature, and over-current), so that the user can know the operation behavior of the battery cell by using this interface and then make other applications.
- (5) Serial interface interface: system development use, such as: code debugging.
- (6) I2C interface: Connects to the communication board, allowing the host computer to know the information of the core (user can use your own system device to know the information of the core through this interface).
- (7) NTC interface: connect with negative temperature coefficient thermistor to measure temperature.
- (8) Cell Connector port: Connects to the core module as shown in Figure below:

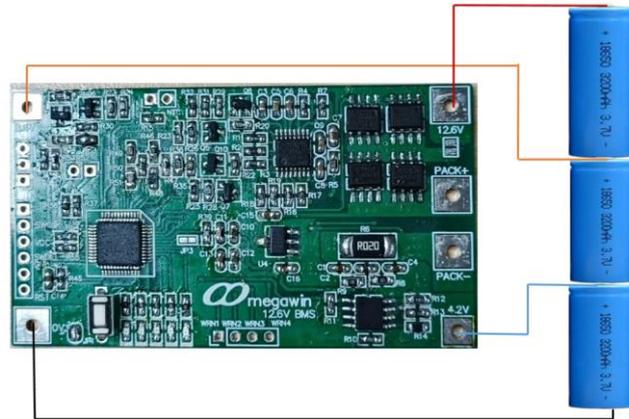


Figure 2-2 EVM connect cell figure

TH251A EVM size : 41 x 70 (mm) , Single lithium battery size 18 x 65 (mm)



Figure 2-3 EVM actual connection of the cells figure

## 2.2 Communication Board

As shown in figure below, it is main for connect the USB interface of the computer and the I2C interface of the hardware EVM, and the signal lines from top to bottom are GND, SCL, SDA, and VCC. The purpose is to convert the commands of the GUI into the commands received by the I2C interface of the hardware EVM, in order to read or run the information of the core, or to write changes to the parameter settings, and so on.

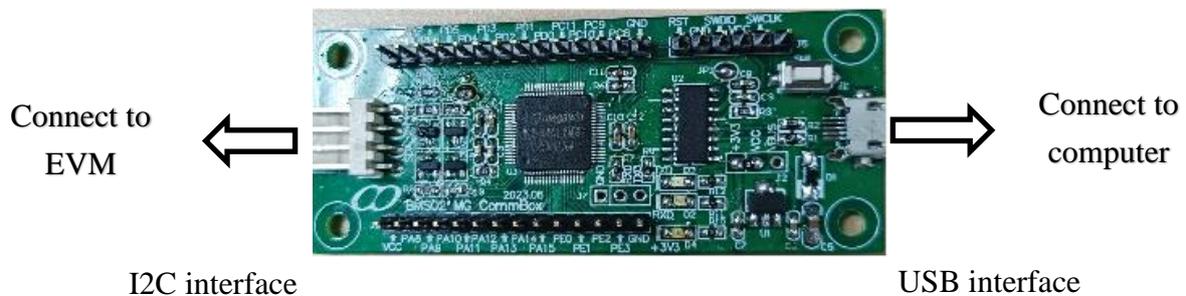


Figure 2-4 Communication Board

### 3. Hardware Connection Instructions

(1) **TH251A EVM** : I2C interface

(2) **TH254A communication board** : I2C interface to USB interface

Medium of communication between EVM and GUI, translates GUI commands to EVM.

Note: user can also communicate with EVM via I2C (your own device) without the need of GUI and communication adapter board.

(3) **megawin 3S Gauge software GUI** : USB interface

GUI to display readings, instantly update parameters and confirm results.



Figure 3-1 Hardware connection overview figure

## 4. Software GUI

### 4.1 Connection

After the whole hardware circuit is connected successfully, open the installed host computer software, the screen shown in figure below will appear.

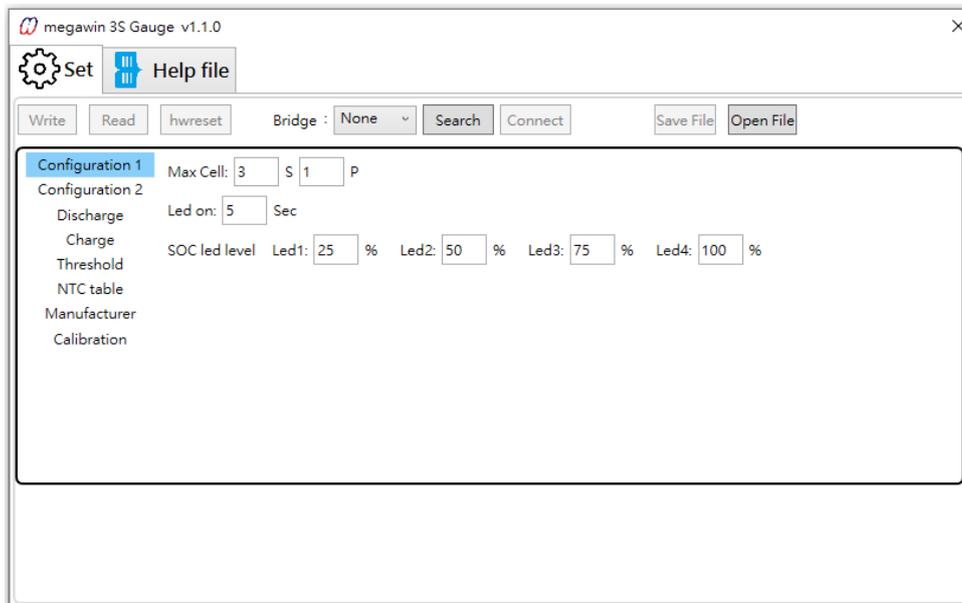


Figure 4-1 Open the host computer software

After clicking Search and selecting the port where the communication board is located on the left side of the screen (user can find it from your computer's hardware administrator), then click on the Connect button. If no error message appears, the Disable buttons (highlighted) on many parts of the screen will be restored to the Control Mode, as shown in the following figure.

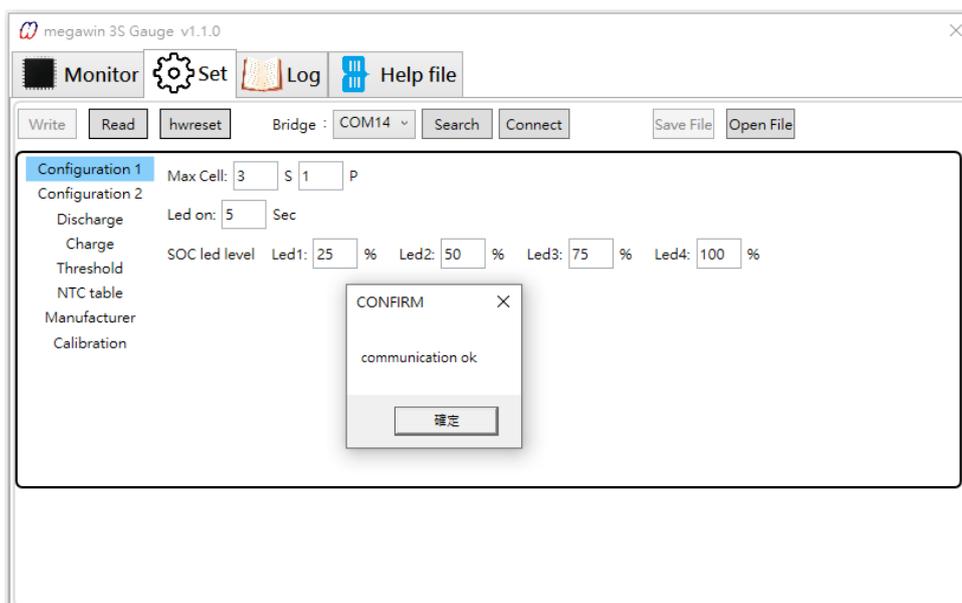


Figure 4-2 The host computer software connection successful

## 4.2 Cell information reading

By clicking the Monitor button (below figure label 1), the GUI will change from the Setup Display page to the Monitor Cell Information page (Figure 4-4).

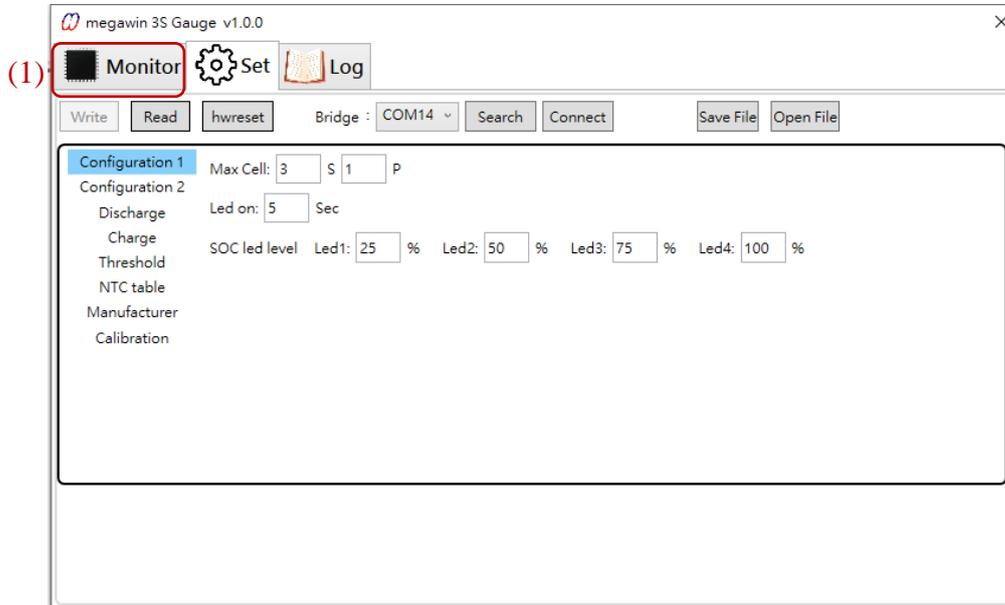


Figure 4-3 The host computer software setting page

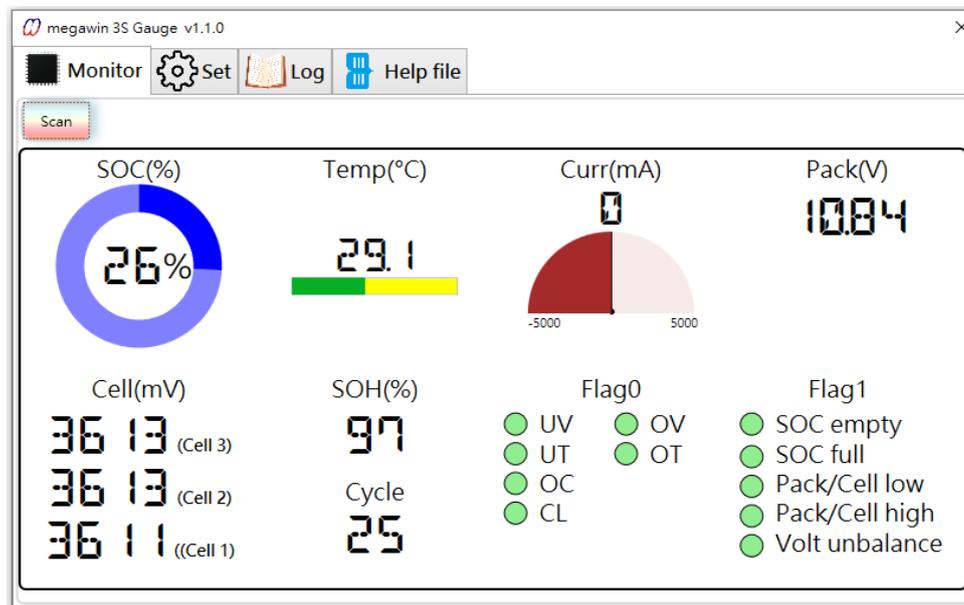


Figure 4-4 Monitor Cell Information Display page

After clicking the Scan button, the button will change to the Stop button (below figure label 1), which will update the display data every 1.5 seconds and display the number of scans (below figure label 2), so that you can know the full status of the battery cell operation in real time from the display page.

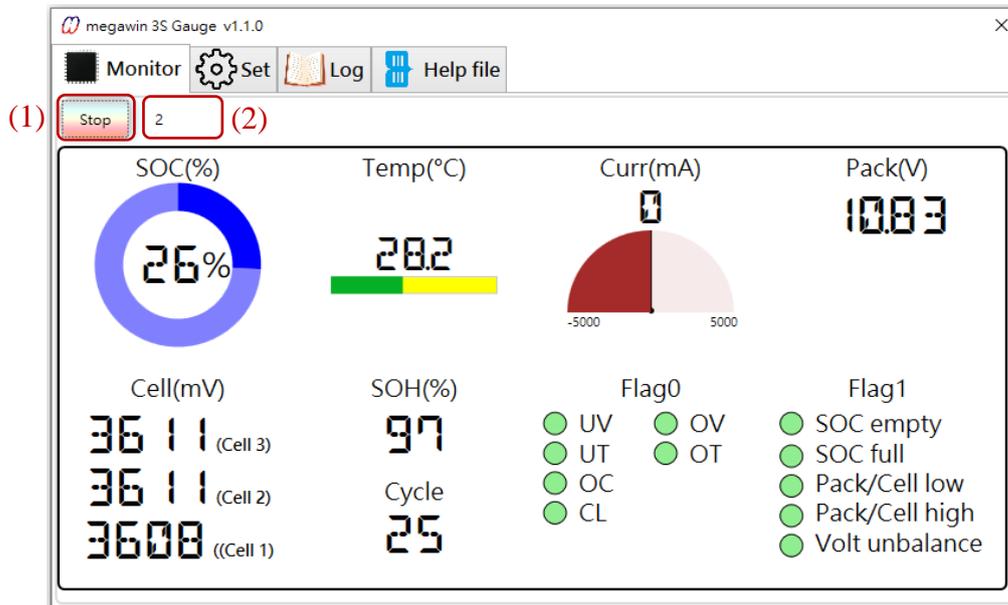


Figure 4-5 Run scan page

## 5. Calibration Instructions and Procedures

Calibrate the voltage, current, and temperature of EVMs to improve the accuracy of system measurement.

### 5.1 Connection

The overall calibration connection is shown in figure below. The computer (console) reads the channels to be calibrated (voltage, current, temperature) on the EVM and compares them with the values of the precision meter connected to the channel to be calibrated. If the error is too large, the amplification gain or offset is adjusted appropriately, so that the difference between the channels to be calibrated on the EVM read by the computer (console) and the measurements made by the precision meter will be within a reasonable range, thus completing the overall channel calibration procedure.

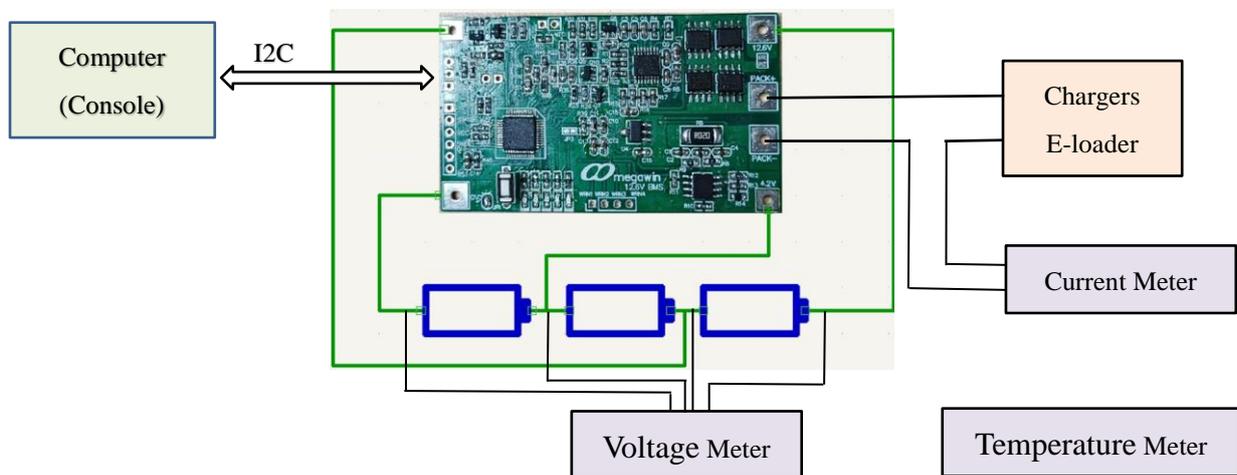


Figure 5-1 Calibration connection overview figure

## 5.2 Software GUI Calibration Process

Provide voltage, current, temperature calibration and message display functions, the calibration operation steps are as follows:

Press the Scan (below figure label 1) button to start the program of reading voltage and temperature from MCU, the screen will be shown as Figure 2-12.

Note: The current must be offset corrected before displaying.

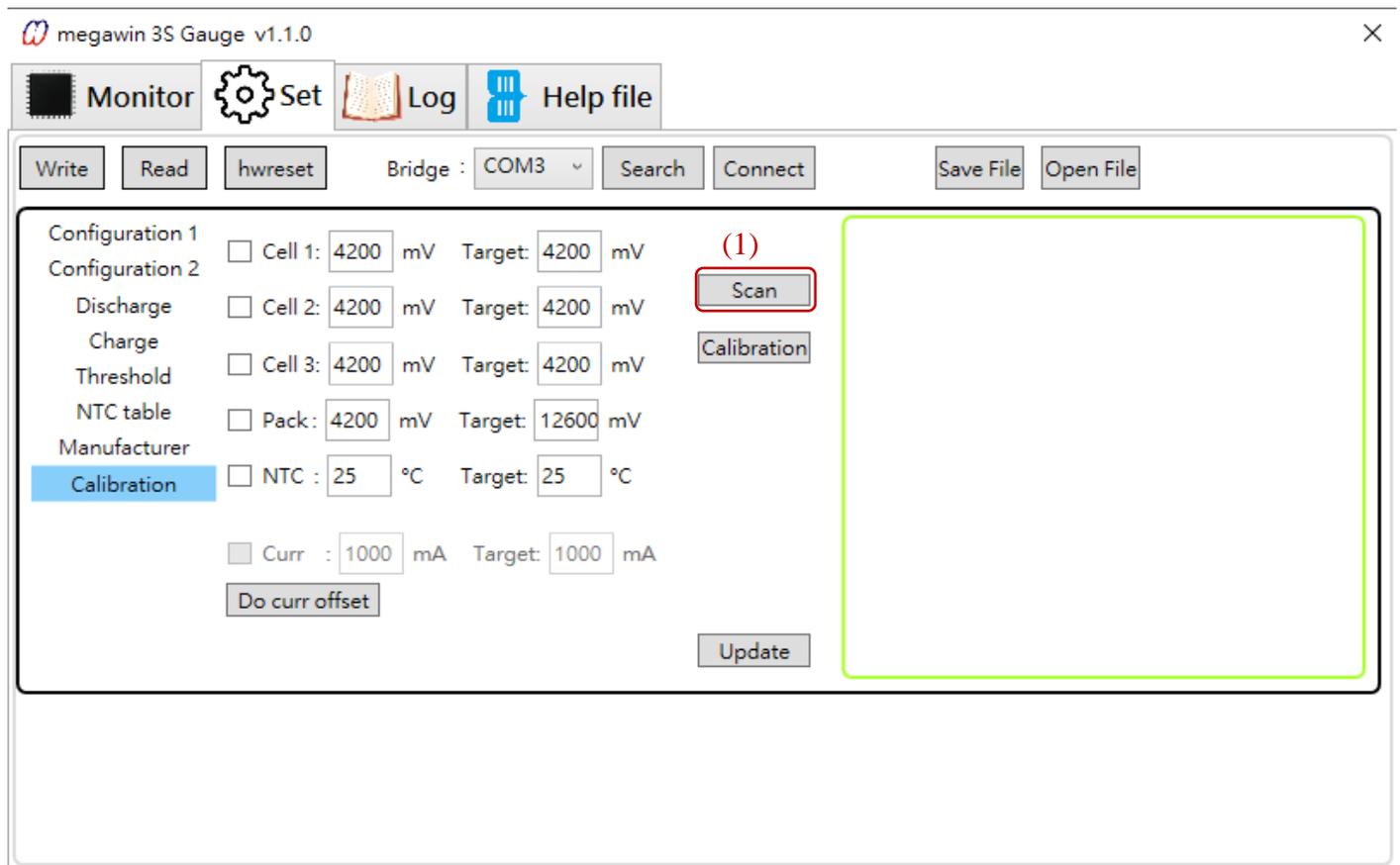


Figure 5-2 Calibration Page

The operating scan button has been changed to a stop button (below figure label 1), and the reading status is displayed (below figure label 2), and the MCU reads back the respective voltage and temperature values of the serial cell (below figure label 3).

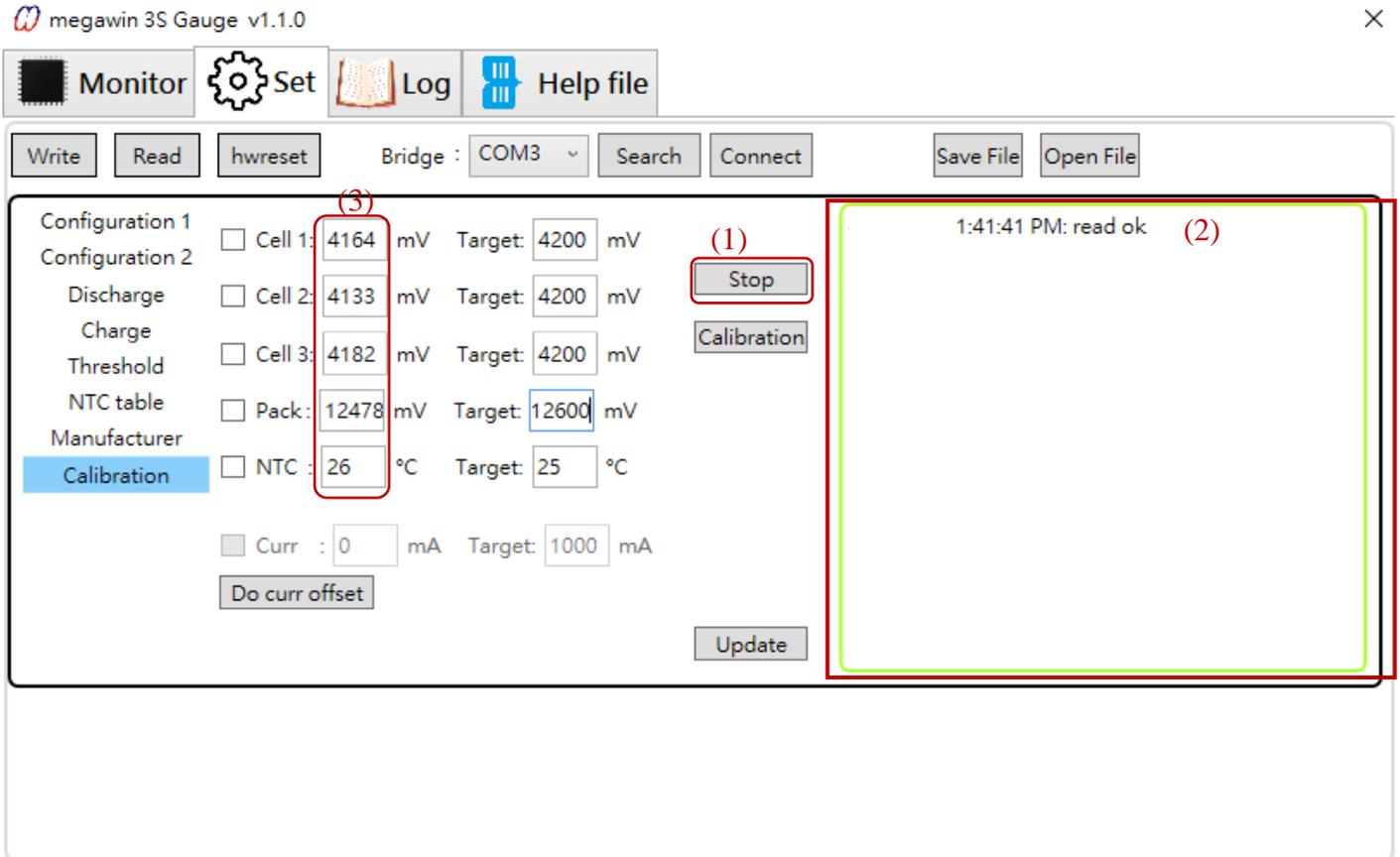


Figure 5-3 Calibration Page - Scan

Use the meter to measure the voltage and temperature of each series-connected cell, fill in the corresponding column with the measured value of the meter (below figure label 1), and check the check box of the channel to be calibrated (below figure label 2), at this time, the voltage of each series-connected cell and the total voltage of the series-connected cell need to be calibrated while the temperature does not need to be calibrated, at this time, the text of the channel to be calibrated will be inverted and the value of MCU will not change (below figure label 3), after confirming that the MCU reading and measured value are correct, press Calibrate (below figure label 4), and then press Calibrate (below figure label 4), after confirming that the MCU reading and measured value are correct, press Calibrate (below figure label 3), and then press Calibrate (below figure label 4). The value of MCU will no longer change (below figure label 3), after confirming that the read value of MCU and the measured value are correct, press the Calibrate (below figure label 4) button to start the calibration, after the calibration is finished, the check box option will be unchecked again, and the system will resume scanning again, at this time, the read value of MCU will be close to the measured value, indicating that the calibration is successful and there is no need to calibrate again. At this time, the system can stop scanning and complete the calibration procedure.

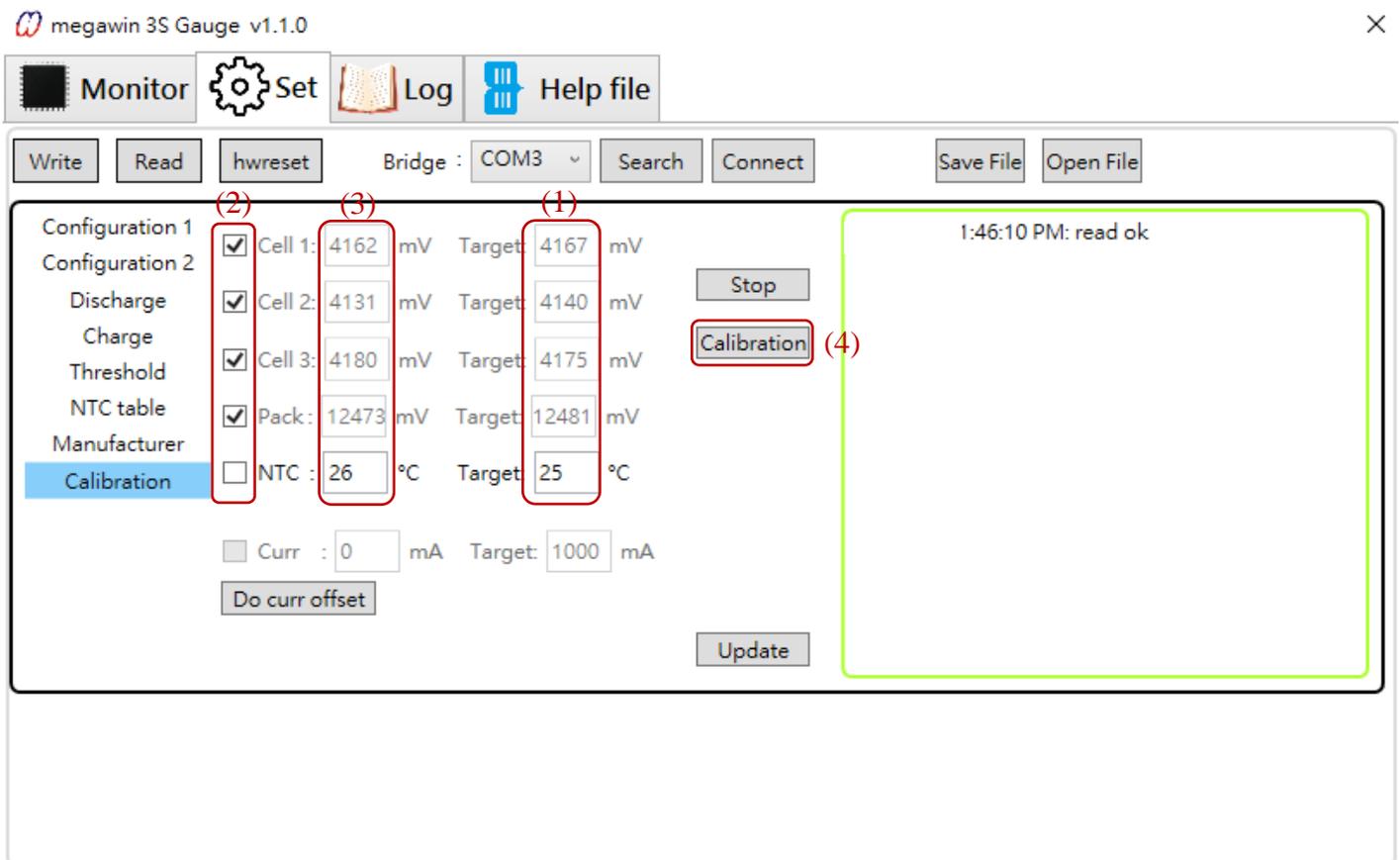


Figure 5-4 Calibration Page-Calibration

If user want to activate the current calibration, user need to do the current offset calibration first, at this time, there should not be charging or discharging behaviors, to avoid the current state, resulting in the wrong current offset calibration, press the offset adjustment (below figure label 1), the current field will be restored from the masked state to the normal display Figure 2-15, allowing the current gain calibration.

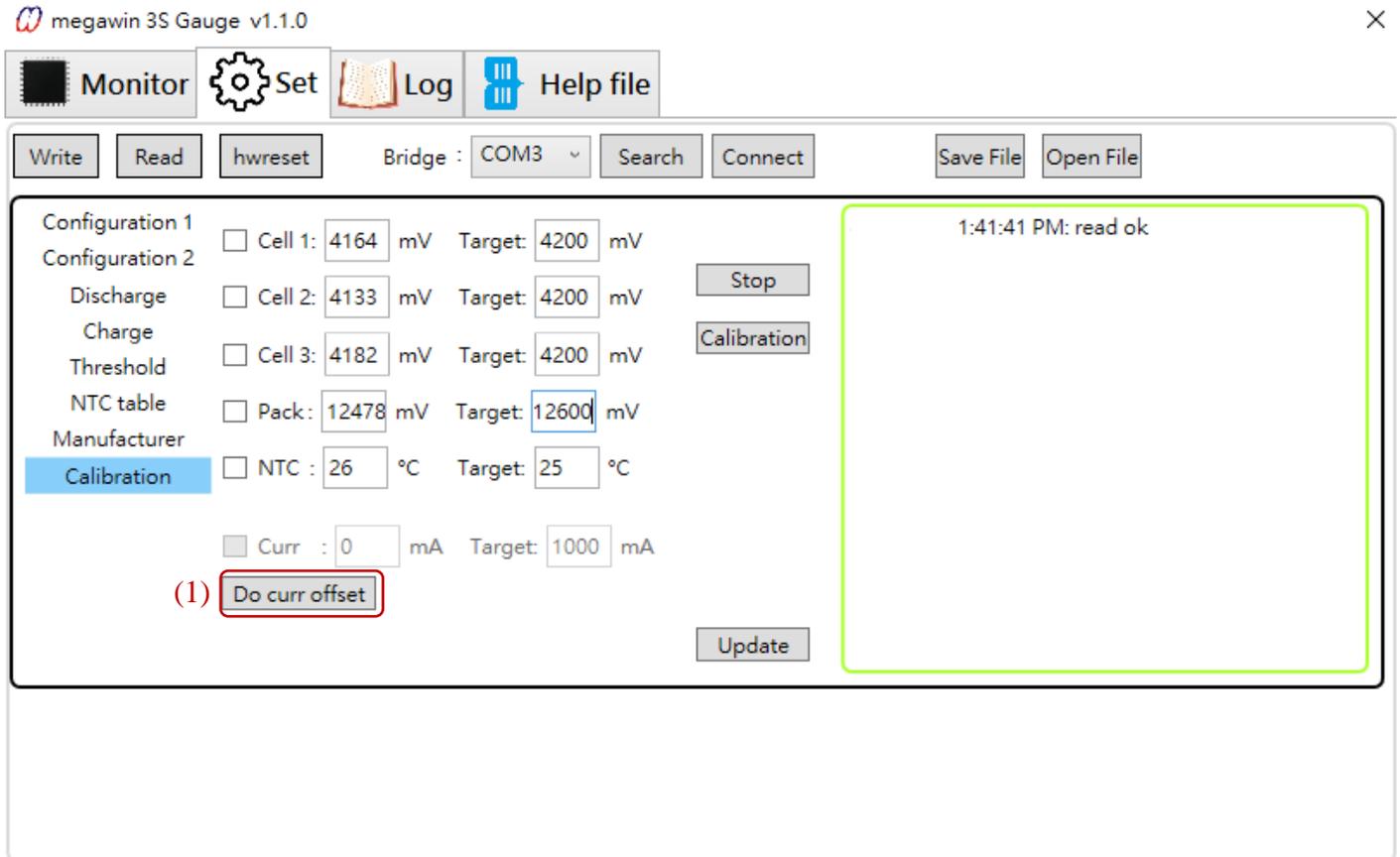


Figure 5-5 Calibration Page - Current Offset

As with the voltage calibration program, the current calibration program can be completed by charging or discharging the MCU so that the current reading of the MCU is the same as the actual current value measured by the instrument, which is filled in the target field (below figure label 1).

At this time, the overall calibration parameters are stored in the MCU parameter temporary space, if the MCU resets the calibration parameters at this time, all the parameters will be lost, so it is necessary to press the Update button (below figure label 2) to save them in the MCU parameter space to ensure that they will not be lost.

Note: Before pressing the Update button, make sure there is no charging or discharging of the system.

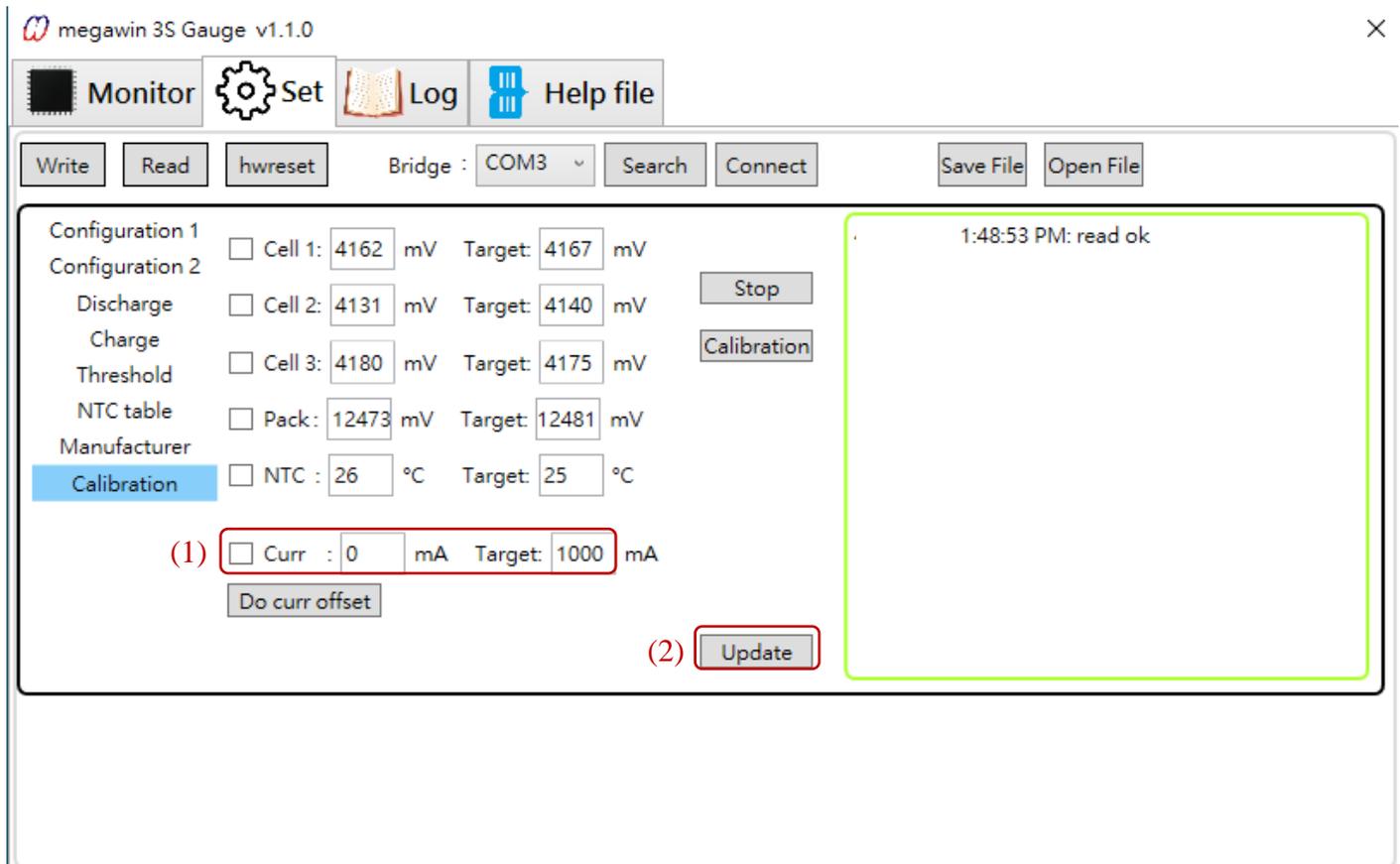


Figure 5-6 Calibration Page-Current Calibration

### 5.3 Communication Protocol (I2C)

EVB mainly communicates with the computer (console) through the I2C interface as a slave device, and its main acceptable slave addresses are two, and its main packet formats are as follows:

(1) Format 1 :

Slave address is 0x55 (7 Bit mode), command/address is 1 Byte long, mainly used for register (RAM) block read/write.

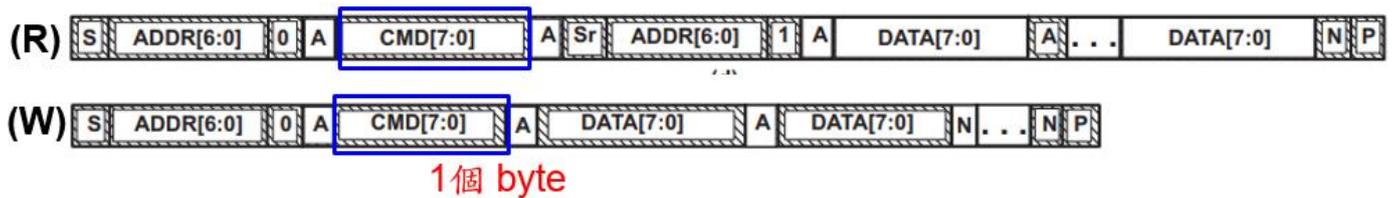


Figure 5-7 I2C format 1 packet Figure

The commands that support reading are listed in the following table :

Table 1 Format 1 reading command table

Command /Address	Name	(Read) Data Length	Description
0x00	Cell_1 voltage	2 bytes	The first string of voltage
0x02	Cell_2 voltage	2 bytes	The second string of voltage
0x04	Cell_3 voltage	2 bytes	The third string of voltage
0x06	Pack voltage	2 bytes	Total Cell Module Voltage
0x08	NTC temperature	2 bytes	Temperature
0x0A	Average current	2 bytes	Current
0x0C	Current offset	2 bytes	Current Correction Offset
0x0E	State of charge	2 bytes	Remaining Capacity Level
0x10	State of health	2 bytes	Health Status Level
0x12	Cycle	2 bytes	Number of discharge cycles
0x14	Status flag	2 bytes	Flag
0x16	Warning flag	2 bytes	Flag
0x18	Tick counter	2 bytes	Total power up time (unit: 2 hours)
0x1C	Force power-up enable	2 bytes	Forced not to enter power saving mode

The commands that support writing are listed in the following table :

Table 2 Format 1 writing command table

Command	Name	(Write) Data Length	Description
0x1E	Force power-up disable	2 bytes	Restore normal power saving mode
0xE0	Adc control	2 bytes	Forcing ADC scanning on/off
0xE1	Current 0A offset calibration	2 bytes	Startup current offset correction
0xE2	Flash page erase	2bytes	Clear internal flash data
0xF1	Update cell_1 gain to memory	2bytes	Update first string gain to memory
0xF2	Update cell_2 gain to memory	2bytes	Update second string gain to memory
0xF3	Update cell_3 gain to memory	2bytes	Update third string gain to memory
0xF4	Update pack gain to memory	2bytes	Updating Total Module Gain to Memory
0xF5	Update ntc offset to memory	2bytes	Update Temperature Offset to Memory
0xF6	Update current gain to memory	2bytes	Update current gain to memory
0xF7	Update current offset to memory	2bytes	Update current offset to memory
0xF8	Update calibration memory data to flash	2bytes	Calibration Memory Data Write to MCU
0xFE	Software reset	2bytes	Reset

(2) Format 2:

The slave address is 0x14 (7 Bit mode), the command is 2 Bytes long, and the main application is to read/write MCU setting data.

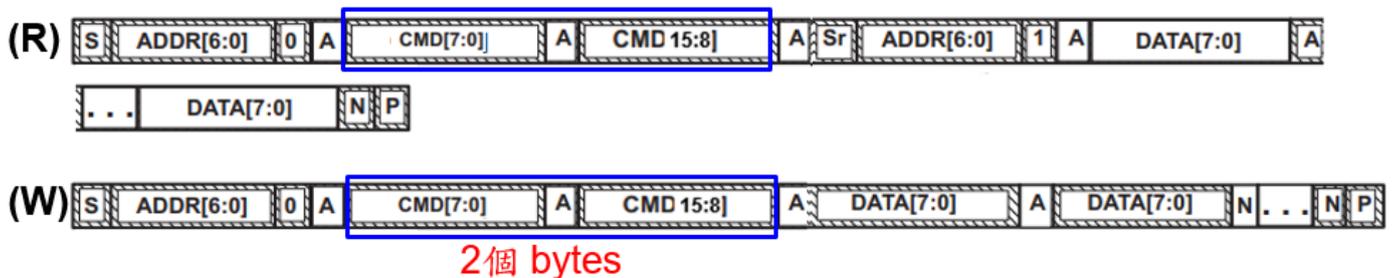


Figure 5-8 I2C format 2 packet Figure

### 5.3 Flowchart

The computer (console) must control the overall calibration process, which is described in the following steps :

#### (1) System Wakeup and Startup ADC Scanning

Wake up system: must use format 1, give 0x1C read command, if read back data is 0x55, 0xAA, means the command has been received, the system will be forced to wake up, in normal mode, waiting for new command.

Initiate ADC scanning: Format 1 must be used to give the 0xE0 write command with 0x00, 0x01 write data to start the internal ADC auto scanning action (forced).

Turn off ADC scanning: must use format 1, give 0xE0 write command, write data to 0x00, 0x00, turn off internal ADC auto scanning (forced).

#### (2) Reading of current calibration parameters

When the internal ADC auto-scan is activated, the system will set the current calibration parameters in register (RAM) space, and the commands are shown in the following table:

Table 3 Calibration Parameters Table

Command /Address	Name	(Read) Data Length	Description
0x22	Cell1_gain	2 bytes	The first string of voltage gain
0x24	Cell2_gain	2 bytes	The second string of voltage gain
0x26	Cell3_gain	2 bytes	The third string of voltage gain
0x28	Pack_gain	2 bytes	Total Cell Module Voltage gain
0x2A	Ntc_offset	2 bytes	Temperature offset
0x2C	Curr_gain	2 bytes	Current gain

User can use Format 1 to give the 0x22 read command to read 12 Bytes of data consecutively to read out all the calibration parameters.

#### (3) Read the Pending Calibration Channel

According to the requirement, select the channel from Table 1, e.g. to read the third string voltage to be calibrated, use Format 1, give 0x04 read command, and read back 2 Bytes consecutively, that is the requirement.

Since the system updates all the data every second, it is recommended to read n consecutive seconds and n data averages to improve the accuracy. Moreover, in order to save the calibration time, it is possible to

To save calibration time, it is possible to calibrate different channels at the same time (read out different channels to be calibrated consecutively) instead of calibrating only one channel at a time.

(4) Calculation of new calibration parameters

Calculation of gain calibration :

Assuming that the gain of reading the current calibration parameter is old\_gain

The reading of the calibration channel is now\_value, the values measured by the meter is target\_value

new\_gain calculation is :

$$\text{new\_gain} = (\text{old\_gain} * \text{target\_value}) / \text{now\_value}$$

Calculation of offset calibration :

Assuming that the offset of reading the current calibration parameter is old\_offset

The reading of the calibration channel is now\_value, the values measured by the meter is target\_value

new\_offset calculation is :

$$\text{new\_offset} = \text{target\_value} - \text{now\_value} + \text{old\_offset};$$

(5) Update calibration data to MCU parameter space.

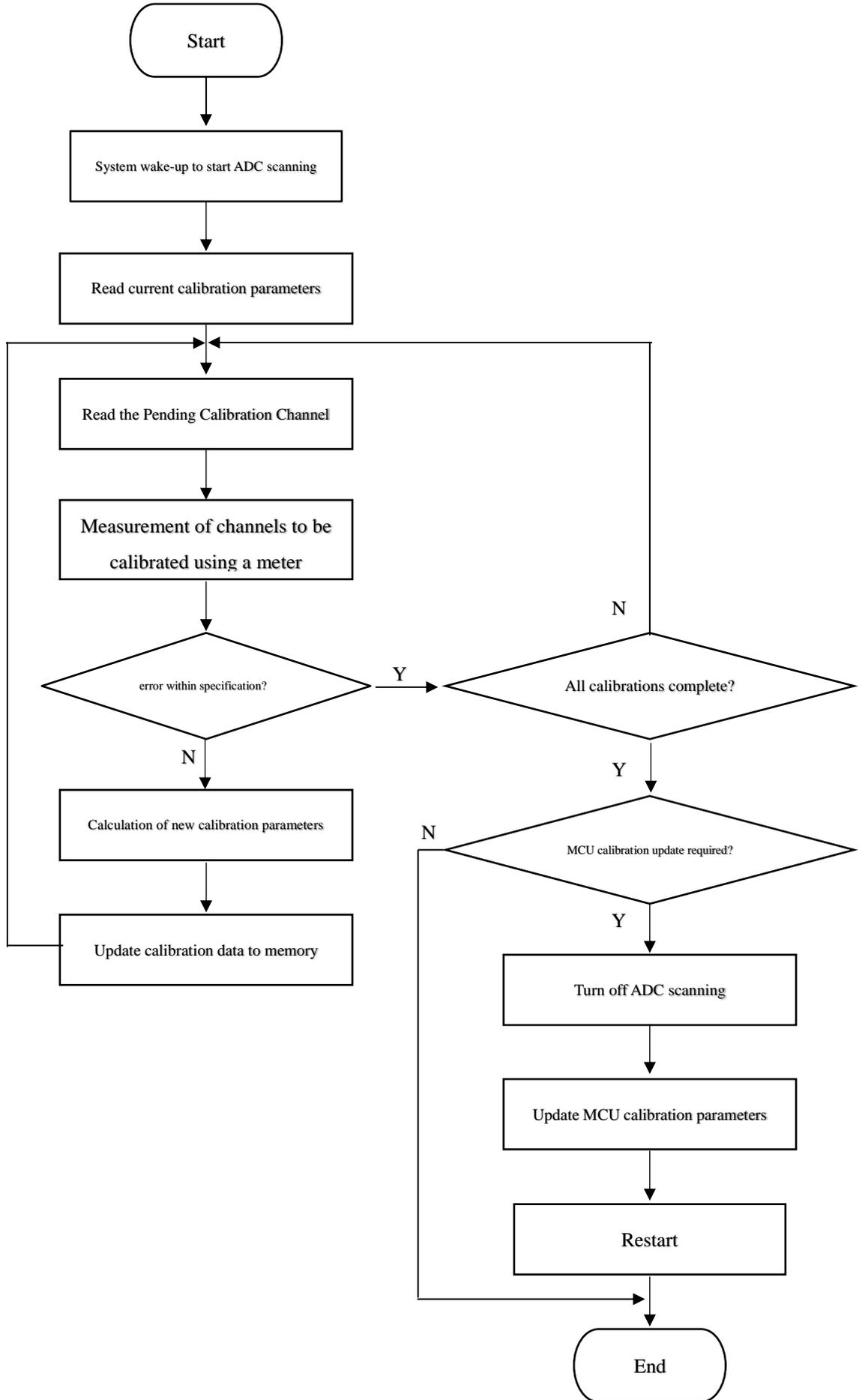
When a new calibration parameter is calculated, the system must be notified to make changes to the calibration parameter in order to confirm whether the new calibration parameter is correct.

For example, to update the temperature offset to the MCU parameter space, a 0xF5 (see Table 2) write command must be issued using Format 1 with the new calibration parameter calculated previously.

(6) Update the calibration parameters of MCU parameter space.

When all the channels to be calibrated have been calibrated and the error is within the acceptable range, then the system can be asked to write the calibration parameters from the memory to the MCU parameter space to avoid the loss of calibration data when the system loses power.

At this time, the system can be asked to write the calibration parameters in the memory to the MCU parameter space to avoid the loss of calibration data when the system is powered off. Therefore, it is necessary to use Format 1 to give the command 0xF8 (Table 2) to write the data 0xA5, 0xA5, and then the system will update the calibration data in the MCU parameter space when it receives the command.



## 6. MCU Parameter Space

This section describes how to update the cell and system parameters on the MCU parameter space and conform to the system usage.

### 6.1 Connection

The overall connection diagram is shown in figure below. Read the message on the EVM through the computer (console) and confirm the I2C communication is normal, then start to update all the data on the MCU parameter space, after updating, please read back again to make sure the written data is correct, if there is no error, please reboot the computer to complete the correct update.

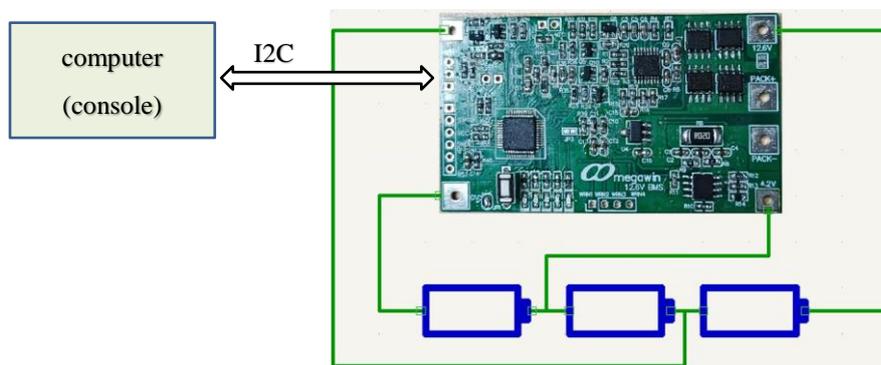


Figure 6-1 connection overview figure

### 6.2 Communication Protocol (I2C)

與 5-3 章節一致。

### 6.3 Flowchart

The computer (console) must be operated to complete the overall program, and the process steps are described below:

#### (1) System Wake-up and ADC Scan Shutdown

Wake up system: must use format 1, give 0x1C read command, if read back data is 0x55, 0xAA, means the command has been received, the system will be forced to wake up in normal mode, waiting for new command.

Turn off ADC scanning: must use format 2, give 0xE0 write command, write data to 0x00, 0x00, turn off internal ADC auto scanning action (forced).

#### (2) Erase the MCU parameter space page.

There are two pages to be processed, 0x00 and 0x01, 0x00 is mainly for placing parameters, 0x01 is for placing calibration, if the update parameters, including the correct calibration data, you must erase the 0x01 page data, so as to prevent the system from re-running when the use of the original first page of the

calibration data, and similarly, if the update parameters, does not contain calibration data, you do not need to erase the 0x01 page data. Similarly, if the parameter update does not contain calibration data, then there is no need to erase the 0x01 page.

You have to use Format 2 to write 0xE2 command to write 0x00 or 0x01 data to erase the corresponding page.

(3) Update all parameters

According to the requirement, select the channel from Table 1, e.g. to read the voltage of the third string to be calibrated, use Format 1.

For example, to read the voltage of the third string to be calibrated, we need to use Format 1 and give the command 0x04 to read, and then read back 2 Bytes consecutively, that is the requirement.

Since the system updates all the data every second, it is recommended to read n consecutive seconds and n data averages to improve the accuracy. Moreover, in order to save the calibration time, it is possible to To save calibration time, you can calibrate different channels at the same time (read out different channels to be calibrated consecutively) instead of calibrating only one channel at a time.

