

# **3S Gauge**

# **User Application Note**

## 目录

目录 .....	2
1. 简介 .....	3
2. 硬件 HW .....	4
2.1 EVM .....	4
2.2 通讯转接板 .....	5
3. 硬件连接说明 .....	6
4. 软件 GUI .....	7
4.1 连线 .....	7
4.2 电芯讯息读取 .....	8
5. 校准说明与流程 .....	10
5.1 连接 .....	10
5.2 软件 GUI 校准流程 .....	11
5.3 沟通协定(I2C) .....	16
5.3 流程图 .....	18
6. MCU 参数空间 .....	21
6.1 连接 .....	21
6.2 沟通协定(I2C) .....	21
6.3 流程图 .....	21

## 1. 简介

3 串电量计解决方案，包含了 EVM 硬件、图形化使用者介面(GUI)、讯息沟通所需要之通讯连接板，使用者藉由 GUI 软件之丰富功能，透过通讯连接板，进而获取连接在硬件板之电芯运行状态，透过适当的资料处理与分析，最佳的参数配置与实验，调适出最好的系统效能，本文将说明解决方案之整体线路连接，与 GUI 之基本操作功能。

硬件 EVM 与软件 GUI 搭配，可提供的产品开发包括：电动工具类，如电钻/电板手/军刀锯等、无线吸尘器、照明灯、打蜡机、测距仪、汽车救援行动电源等 3 串锂电池保护板。

## 2.硬件HW

### 2.1 EVM

TH251A EVM 如下图所示，主要介面接口有：

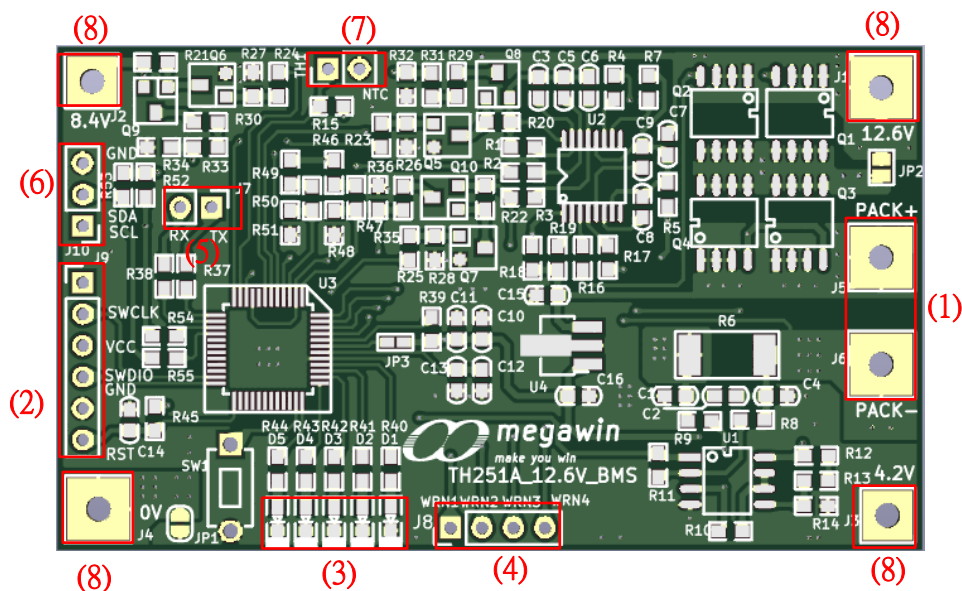


图 2-1 MSE03GM1AD48 for 3S Gauge EVM 介绍

- (1) 充放电介面接口：充放电同孔接口，接电源对电芯做充电的动作，接负载做电力输出的行为。
- (2) 序列除错介面：系统开发使用，例如：更新固件。
- (3) 电量与健康状态显示灯号：提供使用中电芯的剩余容量灯号显示与电芯间之电压差距过大之健康状态异常灯号提醒。
- (4) 警告状态介面接口：提供电量(充饱、放空)与使用行为(充电、放电)及状态(过压、欠压、过温、过流) 等 IO 输出，方便使用者利用此接口，就可得知电芯的运行行为，进而做出其他应用。
- (5) 串口介面接口：系统开发使用，例如：代码除错。
- (6) I2C 介面接口：连接通讯板，让上位机得知电芯讯息(可使用自己系统装置，透过此介面接口，得知电芯讯息)。
- (7) NTC 介面接口：连接负温度系数热敏电阻，量测温度使用。
- (8) 电芯介面接口：连接电芯模组，接法如下图所示：

注意:电池芯安装完成后，保护功能是作动的，因此于 Pack+与 Pack-量测不到电压，取一条电线碰触 Pack+ 与电池组正端，以解除保护。

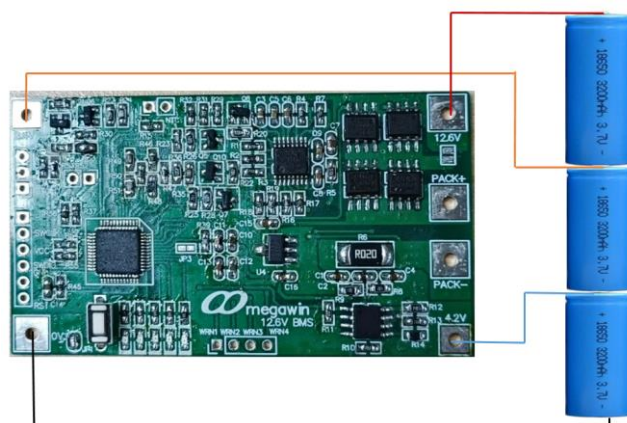


图 2-2 EVM 连接电芯示意图

TH251A EVM 尺寸：41 x 70 (mm)，单颗锂电池尺寸 18 x 65 (mm)



图 2-3 EVM 实际连接电芯图

## 2.2 通讯转接板

如下图所示，负责连接电脑 USB 接口与硬件 EVM 之 I2C 接口，从上到下之信号线依序是 GND、SCL、SDA、VCC，目的是将 GUI 之命令，转换成硬件 EVM 之 I2C 接收的命令，以读取或运行状态之电芯讯息，或是写入更改参数设定等等。



图 2-4 通讯转接板

### 3. 硬件连接说明

(1) TH251A EVM : I2C 接口

(2) TH254A 通讯转接板 : I2C 接口转 USB 接口

EVM 与 GUI 沟通的媒介，转译 GUI 命令给 EVM

注：也可自行透过 I2C(自有装置)与 EVM 沟通，无需透过 GUI 及通讯转接板

(3) megawin 3S Gauge 软件 GUI : USB 接口

图形化介面显示读取资料、立即更新参数确认结果



图 3-1 硬件连接示意图

## 4.软件GUI

### 4.1 连线

当整体硬件线路连接成功后，开启安装的上位机软体，将出现如下图所示画面。



图 4-1 开启上位机软体

点选搜寻后，并在其左边选择通讯板所在的埠口(可从电脑之硬体管理员查知)，接着点选连线键，如无错误提示讯息出现，画面多处禁止操作键(反白)将被恢复成操控模式，并显示连线成功讯息，如下图所示。



图 4-2 上位机软体连线成功

## 4.2 电芯讯息读取

点选监看键(下图标示 1)，GUI 将从设定显示页面，变成监看电芯讯息页面(下图 4-4)。

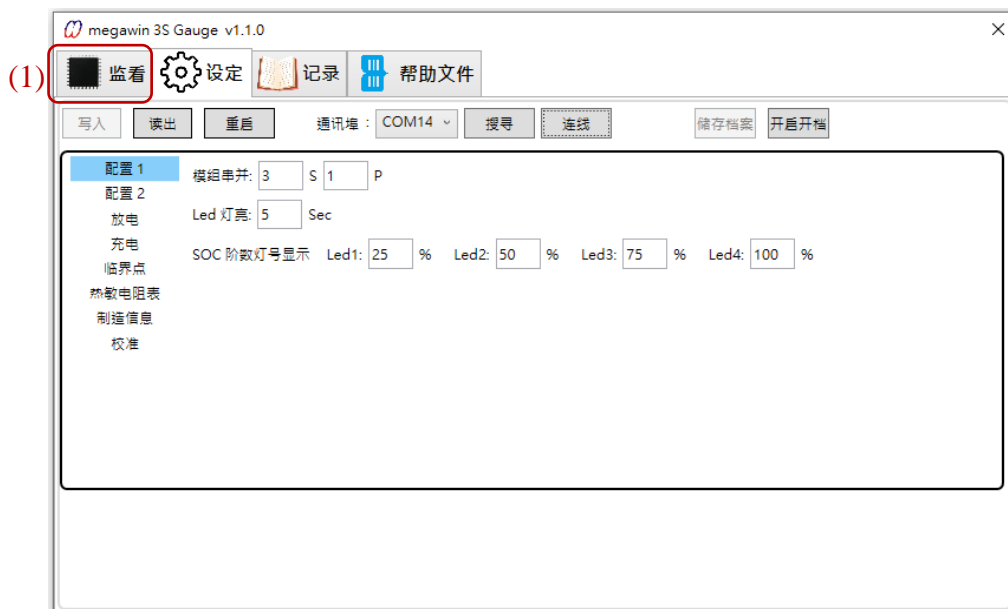


图 4-3 上位机设定显示页面

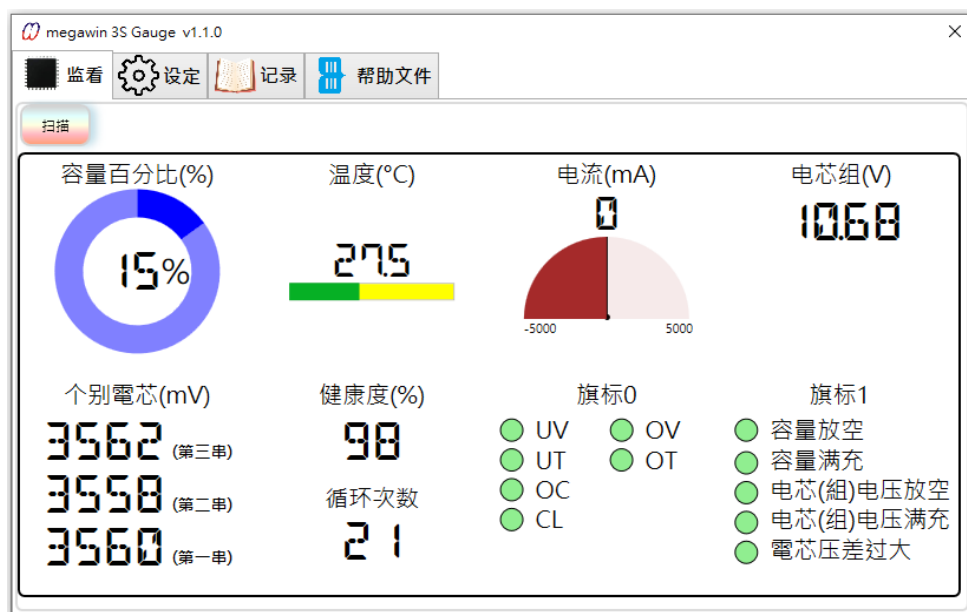


图 4-4 监看电芯讯息显示页面

点选扫描键后，此键将变停止键(下图标示 1)，此时每 1.5 秒更新一次显示资料，并且显示扫描次数(下图标示 2)，从显示页面可即时得知电芯运行的全部状态。

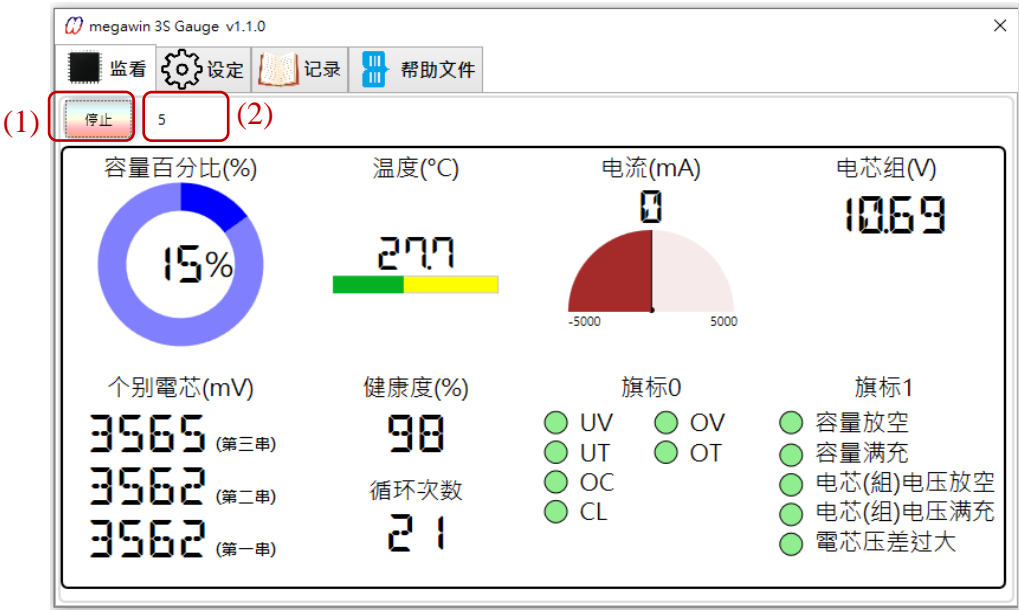


图 4-5 执行扫描页面

## 5. 校准说明与流程

校准 EVM 之电压、电流、温度以提升系统量测的精度。

### 5.1 连接

整体之校准连接示意图如下图所示，透过电脑(主控台)读取 EVM 上之待校准通道(电压、电流、温度)，并与连接在待校准通道上之精密仪器电表值做一比较，如果误差过大时，透过适当的放大增益或偏移量调适，使电脑(主控台)读取 EVM 上之待校准通道与精密电表之量测，其误差落在合理范围内，完成整体通道校准程序。

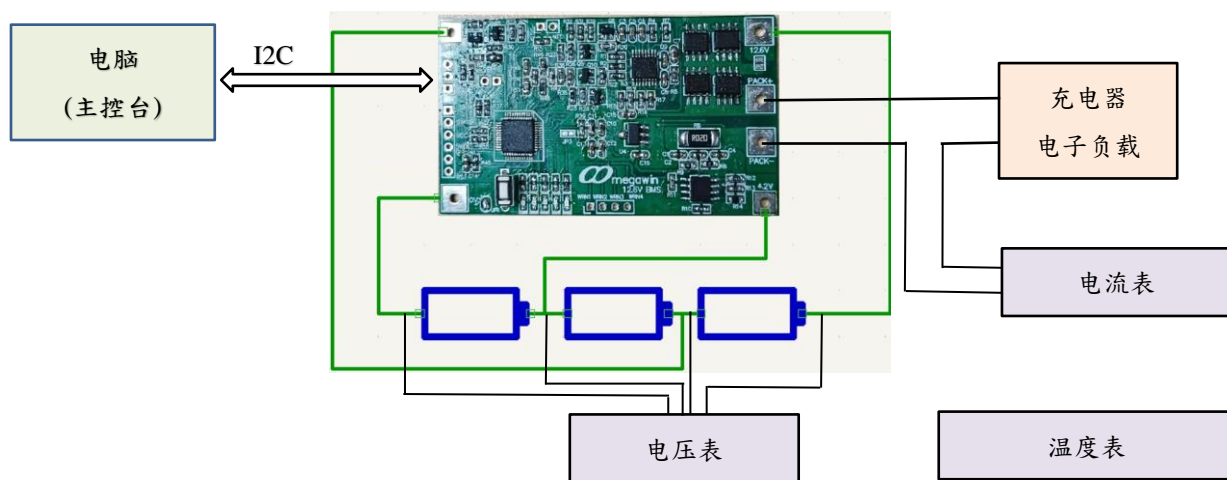


图 5-1 校准连接示意图

5.2 软件GUI校准流程

提供电压、电流、温度校准与讯息显示的功用，校准操作步骤如下：  
先按扫描(下图标示 1)按钮，启动读取 MCU 电压、温度程序，此时画面将如图 2-12 所示。  
注：电流必须在做完偏移校正才会显示。



图 5-2 校准页面

操作的扫描按钮已经变成停止按钮(下图标示 1)，并且显示读取状态(下图标示 2)，MCU 读回的串接电芯的各别电压与温度数值(下图标示 3)。



图 5-3 校准页面-扫描

使用仪表量测各个串接电芯的各别电压与温度，将仪表量测数值填于对应栏位(下图标示 1)，并将所需校准的通道的检查盒选项打勾(下图标示 2)，此时包含串接电芯的各别电压与串接电芯的总电压需校准，而温度不需校准，此时需校准的通道的文字将反白，并且 MCU 的数值将不再改变(下图标示 3)，确认 MCU 读值与量测数值无误后，按校准(下图标示 4)按钮开始启动校准，校准完毕后，前面检查盒选项将回复取消勾选，系统再一次回复扫描状态，此时 MCU 的读值将会与量测的数值接近，表示校准成功，无需再校准一次，此时可停止扫描，完成校准程序。



图 5-4 校准页面-校准

如果要启动电流校准，需先做电流偏移量校准，此时不能有充电或放电的行为，避免在有电流状态，产生错误的电流偏移量校准，按下偏移量调整后(下图标示 1)，电流栏位将从遮蔽状态回复到正常显示图 2-15，允许电流增益校准。



图 5-5 校准页面-电流偏移量

如同电压校准程序，此时可使用充电或放电，让 MCU 的电流读值与仪表量测之实际电流值(填于目标栏位(下图标示 1)完成电流校准程序。

此时整体校准参数储存于 MCU 参数暂存空间，如果此时 MCU 重置校准参数将全部遗失，因此需按更新按钮(下图标示 2)，储存于 MCU 参数空间保证不遗失。

注：按下更新按钮前，需先确认无对系统做充放电的动作



图 5-6 校准页面-电流校准

5.3 沟通协定(I2C)

EVB 主要通过 I2C 接口与电脑(主控台)沟通，做为从机装置，其主要可接受的从机地址有两个，而其主要的封包格式如下：

(1) 格式 1：

从机地址为 0x55 (7 Bit 模式)，命令/地址为 1Byte 长度，主要应用为 register(RAM)区块的读写。

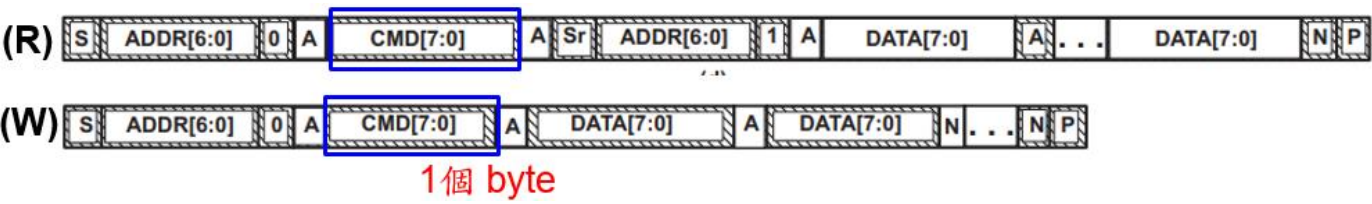


图 5-7 I2C 格式 1 封包图

支援读的命令如下表所示：

表 1 格式 1 读取命令表

命令/ 地址	名称	(读取)资料长度	描述
0x00	Cell_1 voltage	2 bytes	第一串电压
0x02	Cell_2 voltage	2 bytes	第二串电压
0x04	Cell_3 voltage	2 bytes	第三串电压
0x06	Pack voltage	2 bytes	电芯模组总电压
0x08	NTC temperature	2 bytes	温度
0x0A	Average current	2 bytes	电流
0x0C	Current offset	2 bytes	电流校正偏移
0x0E	State of charge	2 bytes	剩余容量阶数
0x10	State of health	2 bytes	健康状态阶数
0x12	Cycle	2 bytes	放电循环次数
0x14	Status flag	2 bytes	旗标
0x16	Warning flag	2 bytes	旗标
0x18	Tick counter	2 bytes	上电总时间(单位:2 小时)
0x1C	Force power-up enable	2 bytes	强迫不进入省电模式

支援写的命令如下表所示：

表 2 格式 1 写入命令表

命令	名称	(写入)资料长度	描述
0x1E	Force power-up disable	2 bytes	恢复正常省电模式
0xE0	Adc control	2 bytes	强迫开启/关闭 ADC 扫描
0xE1	Current 0A offset calibration	2 bytes	启动电流偏移校正
0xE2	Flash page erase	2bytes	清除内部 flash 资料
0xF1	Update cell_1 gain to memory	2bytes	更新第一串增益到记忆体
0xF2	Update cell_2 gain to memory	2bytes	更新第二串增益到记忆体
0xF3	Update cell_3 gain to memory	2bytes	更新第三串增益到记忆体
0xF4	Update pack gain to memory	2bytes	更新串接模组增益到记忆体
0xF5	Update ntc offset to memory	2bytes	更新温度偏移到记忆体
0xF6	Update current gain to memory	2bytes	更新电流增益到记忆体
0xF7	Update current offset to memory	2bytes	更新电流偏移到记忆体
0xF8	Update calibration memory data to flash	2bytes	校正记忆体资料写入 MCU
0xFE	Software reset	2bytes	重置

(2) 格式 2:

从机地址为 0x14(7 Bit 模式) ，命令为 2 Bytes 长度，主要应用为 MCU 设定资料的读写。

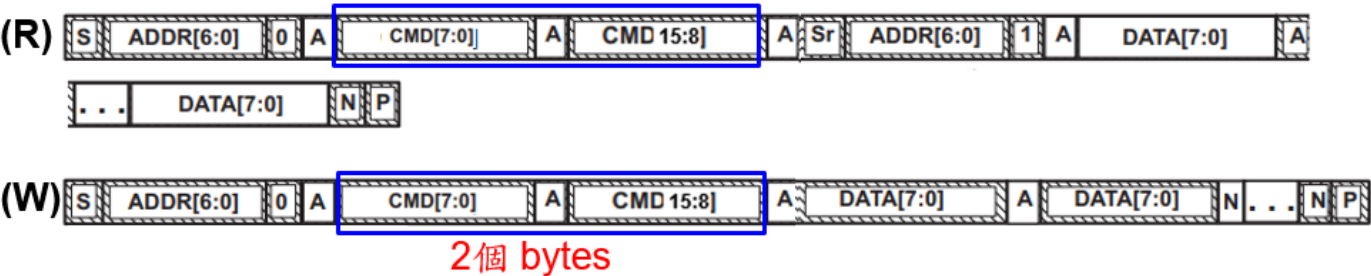


图 5-8 I2C 格式 2 封包图

## 5.3 流程图

电脑(主控台)必须操控完成整体的校正程序，其流程步骤说明如下：

### (1) 系统唤醒与启动 ADC 扫描

唤醒系统：必须使用格式 1，下达 0x1C 读取命令，如果读回资料为 0x55、0xAA，代表命令已被接收，系统将被强迫唤醒，处于正常模式，等待新命令。

启动 ADC 扫描：必须使用格式 1，下达 0xE0 写入命令，写入资料为 0x00、0x01，启动内部 ADC 自动扫描动作(强迫)。

关闭 ADC 扫描：必须使用格式 1，下达 0xE0 写入命令，写入资料为 0x00、0x00，关闭内部 ADC 自动扫描动作(强迫)。

### (2) 读取现今校准参数

当启动内部 ADC 自动扫描时，系统将会将现今校准参数置于 register(RAM)空间内，其命令如下表所示：

表 3 校正参数表

命令/地址	名称	(读取)资料长度	描述
0x22	Cell1_gain	2 bytes	第一串电压增益
0x24	Cell2_gain	2 bytes	第二串电压增益
0x26	Cell3_gain	2 bytes	第三串电压增益
0x28	Pack_gain	2 bytes	电芯模组总电压增益
0x2A	Ntc_offset	2 bytes	温度偏移
0x2C	Curr_gain	2 bytes	电流增益

此时，可使用格式 1，下达 0x22 读取命令，连续读取 12 Bytes 资料，将全部校准参数读出。

### (3) 读取待校准通道

依据需求，从表 1 选择所用通道，例如：读取待校准之串接第三串电压，则需使用格式 1，下达 0x04 读取命令，连续读回 2 Bytes，即为所求。

由于系统每一秒更新全部数据，因此建议连续读取 n 秒，n 笔数据平均以提高精度。并且为了节省校正时间，可同时做不同通道的校正(连续读出不同的待校准通道)，而不用一次只校正一个通道。

(4) 计算新校准参数

增益校准的计算方式：

假设读取现今校准参数之增益为 old\_gain

校准通道之读值为 now\_value，电表量测之值为 target\_value

新的增益(new\_gain)计算为：

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

偏移计算方式：

假设读取现今校准参数之偏移为 old\_offset

待校准通道之读值为 now\_value，电表量测之值为 target\_value

新的偏移(new\_offset)计算为：

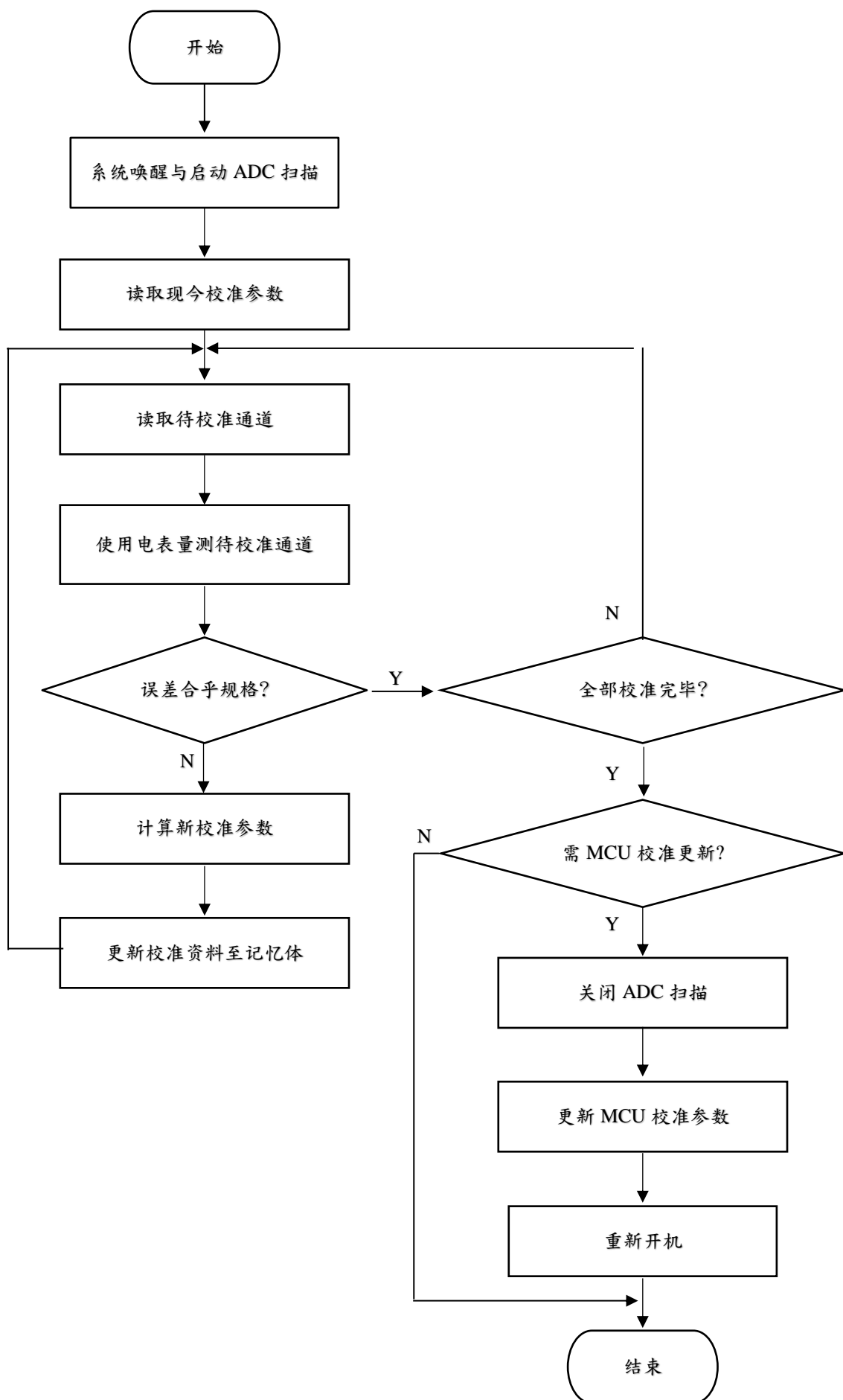
$$\text{new\_offset} = \text{target\_value} - \text{now\_value} + \text{old\_offset};$$

(5) 更新校准资料至 MCU 参数空间

当计算出新校准参数，必须通知系统对校正参数做出变更，以便确认新的校正参数是否正确，例如：更新温度偏移到 MCU 参数空间，则必须使用格式 1，下达 0xF5 (参阅表 2) 写入命令，写入资料为之前所计算出的新校准参数。

(6) 更新 MCU 参数空间的校准参数

当全部待校准通道都已校正完毕后，误差都落在可接受范围内，此时可要求系统将记忆体内的校准参数写入至 MCU 参数空间内，以免系统断电后造成校准资料的流失，因此必须使用格式 1，下达 0xF8(表 2) 写入命令，写入资料为 0xA5、0xA5，此时系统收到命令后，将会更新 MCU 参数空间内的校正资料。



## 6. MCU参数空间

介绍如何更新 MCU 参数空间上之电芯与系统参数，并合乎系统使用。

### 6.1 连接

整体之连接示意图，如下图所示。透过电脑(主控台)读取 EVM 上之讯息，确认 I2C 通讯正常后，开始更新全部 MCU 参数空间上之资料，更新完毕后，请再一次读回确认写入的资料是正确的，如无错误，请重新开机，完成此次正确更新。

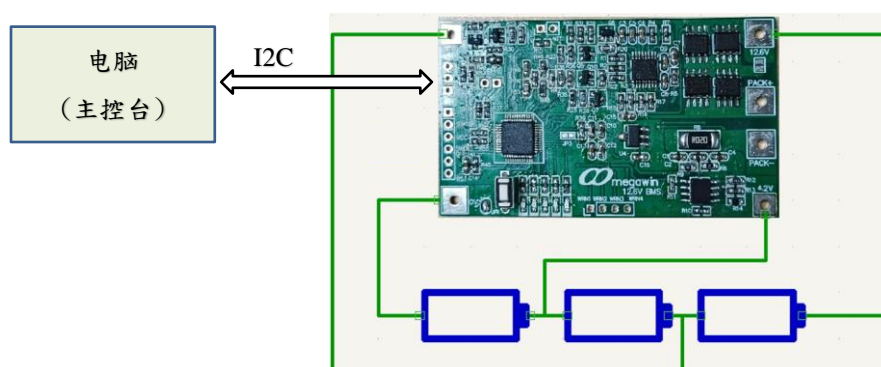


图 6-1 连接示意图

### 6.2 沟通协定(I2C)

与 5-3 章节一致。

### 6.3 流程图

电脑(主控台)必须操控完成整体的程序，其流程步骤说明如下：

#### (1) 系统唤醒与关闭 ADC 扫描

唤醒系统：必须使用格式 1，下达 0x1C 读取命令，如果读回资料为 0x55、0xAA，代表命令已被接收，系统将被强迫唤醒，处于正常模式，等待新命令。

关闭 ADC 扫描：必须使用格式 2，下达 0xE0 写入命令，写入资料为 0x00、0x00，关闭内部 ADC 自动扫描动作(强迫)。

#### (2) 抹除 MCU 参数空间页面

有 2 个页面需处理，分别为 0x00 与 0x01，0x00 主要为放置参数的地方，0x01 为放置校准的地方，如果此次更新参数，包含了正确校准资料，则必须抹除 0x01 页面资料，以免系统重新跑起来时，使用到原本第一页的校准资料，同理，如果此次更新参数，不包含校准资料，则不必抹除 0x01 页面资料。

必须使用格式 2，下达 0xE2 写入命令，写入资料为 0x00 或 0x01，抹除相应的页面。

### (3) 更新全部参数

依据需求，从表 1 选择所用通道，例如：读取待校准之串接第三串电压，则需使用格式 1，下达 0x04 读取命令，连续读回 2 Bytes 即为所求。

由于系统每一秒更新全部数据，因此建议连续读取 n 秒，n 笔数据平均以提高精度。并且为了节省校正时间，可同时做不同通道的校正(连续读出不同的待校准通道)，而不用一次只校正一个通道。

