BMS Getting Started Guide

Power Application Controller®

Table of Contents

1 INTRODUCTION 3
2 DEVELOPMENT SETUP 3
2.1 Battery Pack Connection 5
2.2 Power Supply and Electronic Load Connection 5
2.2.1 Discharging Condition 5
2.2.2 Charging Condition 5
2.3 PC UART and Debugger Connection 5
2.3.1 Option 1: ET-UARTSWD 6
2.3.2 Option 2: ET-UARTISO-1 with ET-IARISO-1 or jumper SWD connection 9
3 BUILD AND FLASH THE FIRMWARE 11
3.1 Choose Device Family and IDE Project 11
3.2 Configure Firmware 11
3.3 Build and Flash the Firmware 12
4 BMS GUI CONFIGURATION 12
4.1 COM Port Configuration 16
5 OPERATING THE BMS ON BATTERY PACK 17
5.1 Load The Configuration File 17
5.2 Load Battery SOC Profile File 17
5.3 Enable Auto FETs control 18
5.4 Next Steps 19
6 ADDITIONAL GUI INFORMATION 19
6.1 Load/Save Parameters and Configuration 19
6.2 Read / Write Parameters 19
6.3 Calibration 20
6.3.1 Cells Voltage Calibration 20
6.3.2 Current Calibration 21
6.4 Datalogger 21
6.5 Auto Gain Current DPGA 23
7 LEGAL INFORMATION 24

1 INTRODUCTION

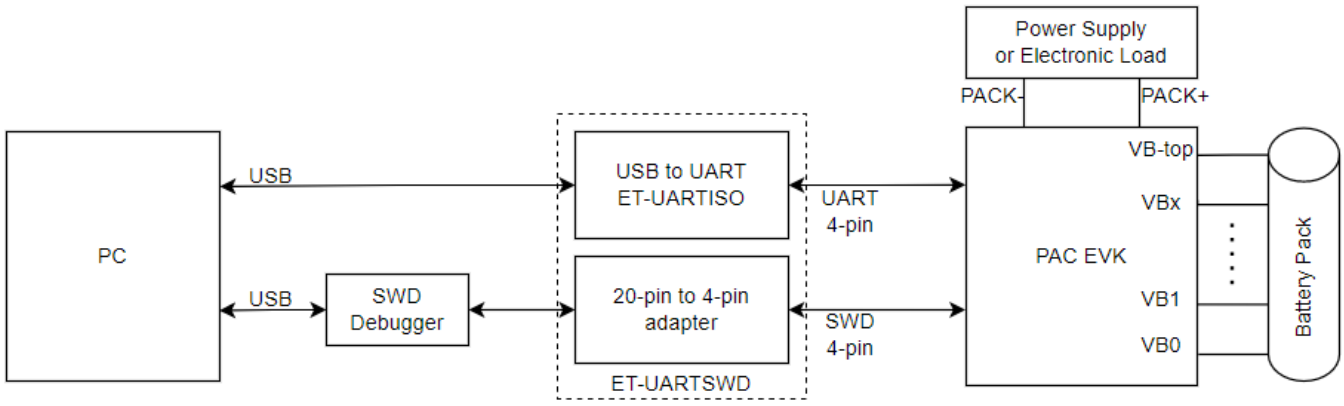
The PAC BMS Getting Started Guide is designed to get the PAC BMS FW up and running with a PAC EVK and battery pack. It assumes that the desired IDE has been installed and the appropriate PAC22xx IDE support files have been added.

2 DEVELOPMENT SETUP

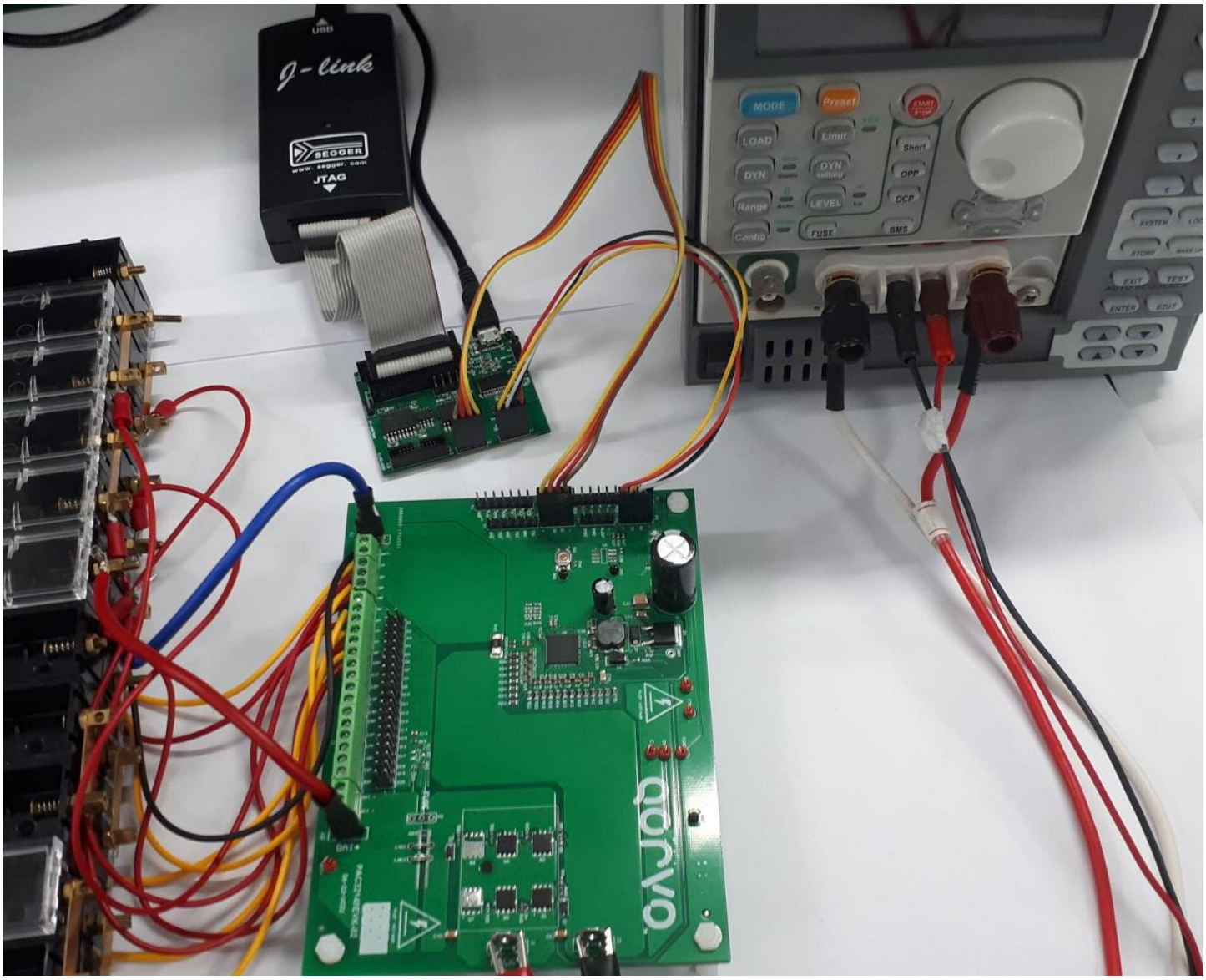
The first step is to setup the development environment to monitor the desired 10-series to 20-series Li-Ion, Li-Polymer and LiFePO4 battery packs as shown in the figure below. The setup must include the following:

1. PAC EVK board
2. Power Supply
    a. Should be capable of supplying Vin rated for the battery pack charging and the particular PAC EVK
    b. Should be able to supply enough current as required by the battery pack.
    c. Should be able to sink current as required Load (Optional)
3. Electronic Load for using if the Power Supply is only able source output.
4. SWD debugger (not in kit, must purchase separately)
5. Adapter Board(s) for connecting the SWD debugger and UART for BMS GUI communications
6. PC with desired IDE.

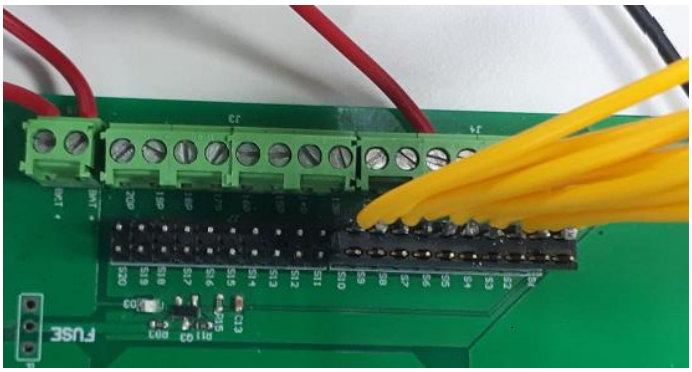
The figure below illustrates the different components.



This picture is a close-up of the EVK connections for the PAC22140 EVK



VBx Connection note: The VBx cells sense may be wrapped and plug to J7 and J8 header connectors instead of using screw connectors.



2.1 Battery Pack Connection

A normal Battery Pack includes BAT+ and BAT- terminals and a number of series cells monitoring connectors. The top and bottom of the cell stack are connected to BAT+ and BAT- spade tab connector, cells voltage are connected by corresponding screw terminal connectors. The Qorvo PAC22140 is a Smart Battery Monitoring System that can monitor a 10-series to 20 series battery pack. In the actual application, the number of used cells may be less than 20 cells. PAC22140EVK1 cell count can be configured by using the selection resistor which connects the corresponding top cell to the BAT+ connector. The bottom resistors from R70 to R80 are used to connect VB20 to VB10 respectively or may be shorted directly by wire and screw connectors J3, J4, J9. When using less than 20 cells, the unused cells corresponding VBx pins should be shorted together.

2.2 Power Supply and Electronic Load Connection

The PAC2140 contains a Configurable Analog Front-End (CAFETM) that can be used to sense battery pack current in charging and discharging condition. A bipolar sink source power supply may be used to charge or discharge battery pack.

A single Power supply and Electronic Load are able used corresponding to charging and discharging operation.

2.2.1 Discharging Condition

In Discharging operation, the battery pack supplies power to outside or output load via PACK+ and PACK- spade connectors. Connect positive of Electronic load to PACK+ and ground of Electronic Load to PACK- connector and adjust load to change discharging current.

2.2.2 Charging Condition

In Charging operation, the battery is charged from a Power via PACK+ and PACK- connectors according to the rating of the PAC EVK and maximum desired charging current. Please take notice of limit of Power Supply on charging to prevent the battery from explosion.

2.3 PC UART and Debugger Connection

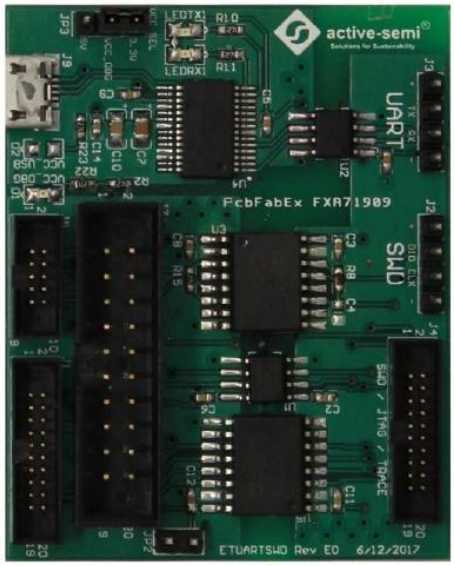
To control the BMS FW, a Windows based GUI has been created that requires a UART connection to the PAC device. And, a debugger is required to flash the BMS FW to the PAC device. Two primary options are available for the UART and debugger connectivity:

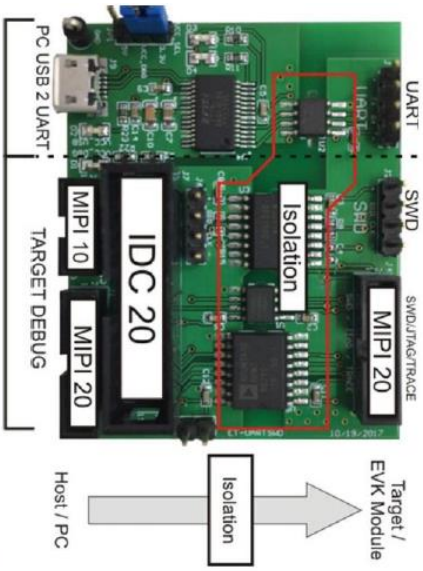
1) ET-UARTSWD combination adapter board

2) ET-UARTISO-1 with either ET-IARISO-1 or jumper SWD debugger connection

2.3.1 Option 1: ET-UARTSWD

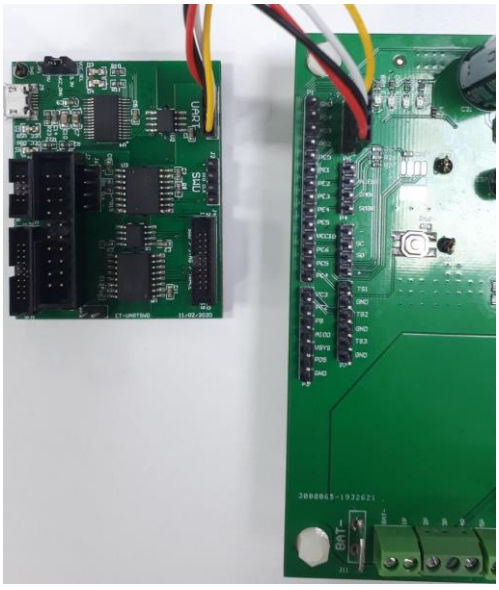
Using the ET-UARTSWD adapter board is the best method for UART and Debugger connectivity. New PAC EVKs are now shipping with the ET-UARTSWD combination adapter board that includes isolation for high voltage applications (see image below on left). The ET-UARTSWD has both a USB to UART function and connectors that adapt various debugger connector types to the 4-pin SWD header found on all PAC EVKs. It also includes an optional target side 20-pin MIPI connector for PAC EVKs. Various aspects of the board are highlighted in the image below on the right.

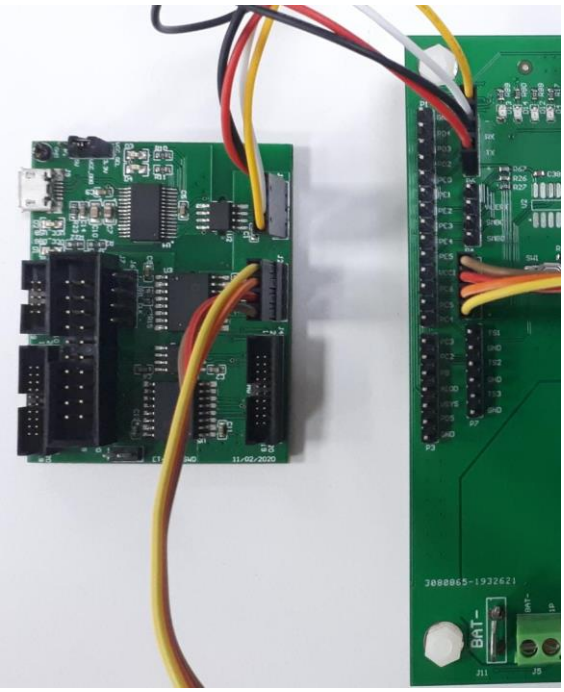




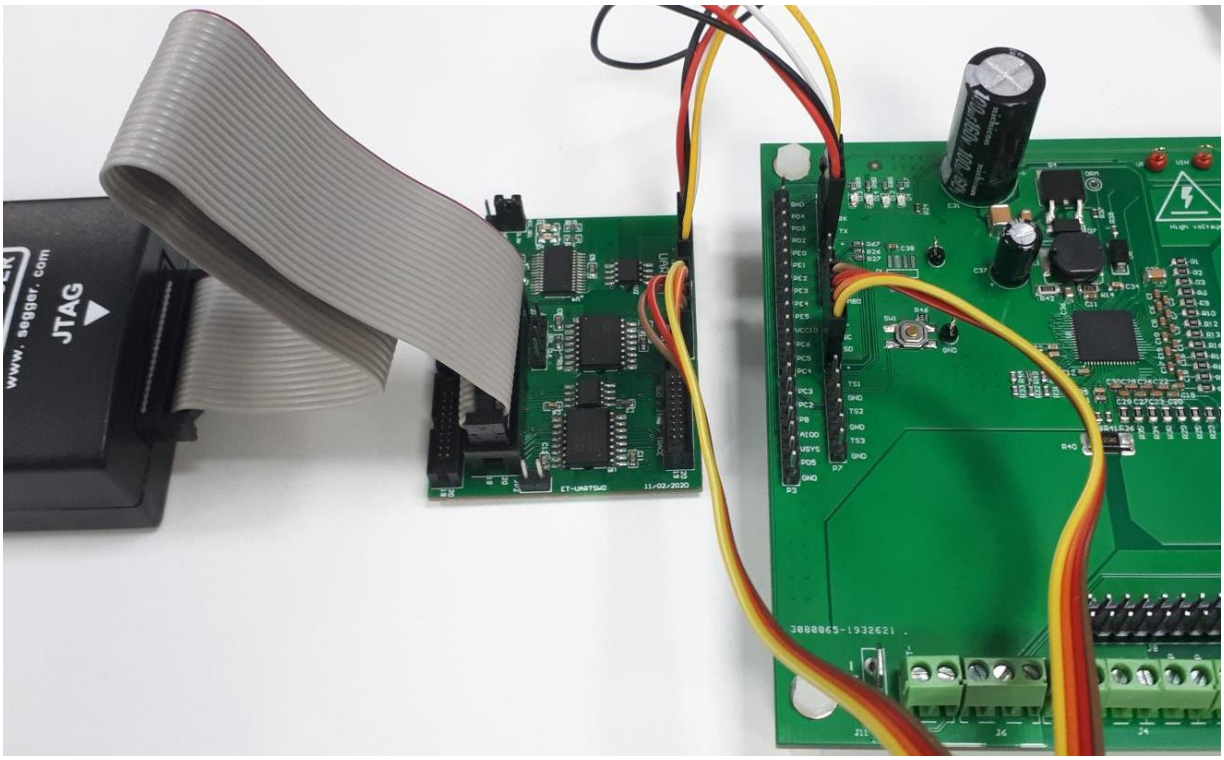
To connect the UART, use a 4-wire cable to connect the PAC EVK 4-pin UART header labelled [+ TX RX -] to the ET-UARTSWD 4-pin UART header labelled [+ TX RX -]. The image on the right shows the connections for the PAC22140 EVK.

Next, use a 4-wire cable to connect the PAC EVK 4-pin SWD header labelled [+ SD SCL -] to the ETUARTSWD 4-pin UART header labelled [+ DIO CLK -]. The image below shows the connection for the PAC22140 EVK.





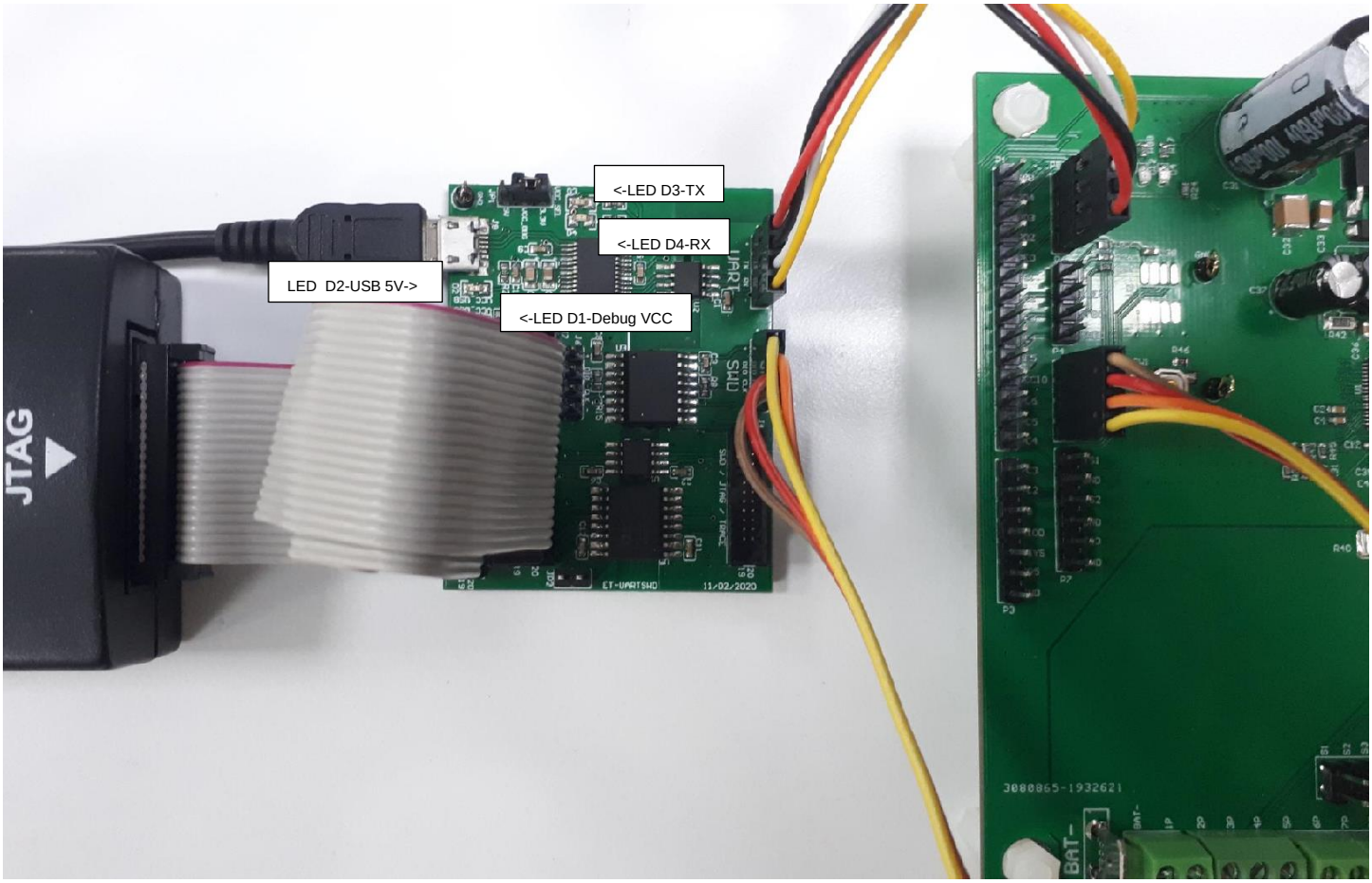
Now connect the 20-pin IDC ribbon cable of the SWD debugger to the appropriate PC/Host side connector on the ET-UARTSWD board. The image below shows a J-Link 20 pin IDC ribbon cable connection.



⚠ Notes: J-Link SWD debugger not in EVK kit box, must be purchased separately

1. The majority of debuggers sense the target hardware Vcc on pin 1 and can handle a 3.3v input. So, the most typical configuration for the ET-UARTSWD is where the power for the PC/Host side of the isolation is supplied by the USB to UART circuitry. For this configuration, the ETUARTSWD jumper JP3 should be connecting 3.3v to VCC_DBG, which is the default position. If the debugger can't support this because
   a. it supplies a voltage on pin 1,
   b. it doesn't support 3.3v input, or
   c. if a USB connection to the PC is not being used, then consult the ET-UARTSWD User's Guide for powering the debugger Vcc.
2. Debuggers that have built in isolation are not supported, because the 2 isolation circuits will interfere with each other.
   a. The ET-COLINK-1 debugger for CooCox includes isolation and so is not supported with ET-UARTSWD. ET-COLINK-1 can be directly connected to the 4-pin SWD header.
3. For debuggers with built in isolation, connect the 4 pins [Vcc, SD, SC, Gnd] directly to the SWD 4 pin header on the PAC EVK using jumpers.

Next connect a USB cable from the PC to the ET-UARTSWD adapter board as illustrated in the image below. If USB power is present, then LED D2 should light up. LED D1 should also light up indicating the debug Vcc is active. D3 and D4 LEDs will turn on and off based on UART TX and RX activity respectfully.



Finally, plug the debugger's USB cable into the PC.

Now proceed to paragraph 3 to build and flash the FOC firmware image.

2.3.2 Option 2: ET-UARTISO-1 with ET-IARISO-1 or jumper SWD connection

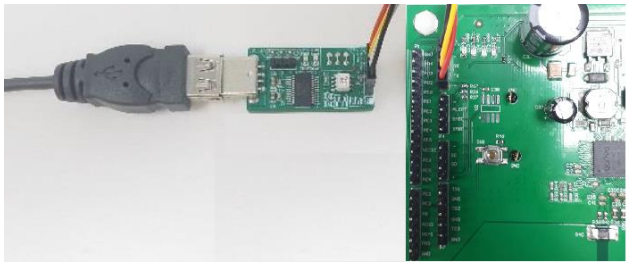
Another option for UART connectivity is to use the ET-UARTISO-1 adapter for USB to UART communications. In this case, SWD connectivity to the PAC EVK is handled separately. The ET-UARTISO-1 is shown below.

To connect the UART, use a 4-wire cable to connect the PAC EVK 4-pin UART header labelled [+ TX RX -] to the ET-UARTISO-1 4-pin UART header labelled [V T R G]. Note that V T R G stands for Vcc, Tx, Rx, and Gnd respsectively. Then, connect the USB port to a PC.

⚠ Note:

1. The ET-UARTISO-1 can be plugged directly into a USB port on the PC, or used with a USB extension cable.

The image on the right shows the connection of the ET-UARTISO-1 to the PAC22140 EVK using a USB extension cable.



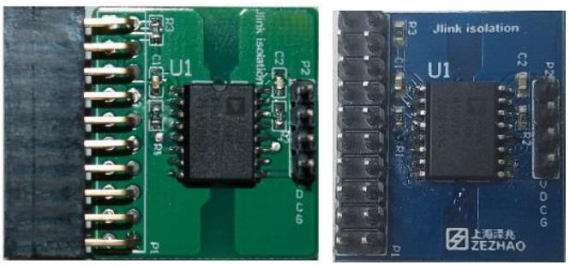
LEDTX and LEDRX LEDs will turn on and off based on UART TX and RX activity respectfully.

When using the ET-UARTISO-1, there are 2 options for SWD connectivity to the PAC EVK:

1. Use the ET-UARTISO-1 SWD adapter board
2. Connect the SWD debugger to the PAC EVK using jumper wires

2.3.2.1 ET-IARISO-1

For SWD debugger connectivity, the ET-IARISO-1 isolated SWD adapter board can be used to adapt the 4 pin SWD connector of PAC EVKs to most SWD debuggers that have a standard ARM 20-pin IDC connector. Two versions of the ET-IARISO-1 adapter board are shown on the right. The green one has a right-angle connector, which can be directly plugged into the SWD debugger without a cable. The blue one has a straight header, which the debugger 20-pin ribbon cable can be plugged into.



Because the debugger side of the ET-IARISO-1 requires power, either

1. The SWD debugger must be cable of supplying 3.3V power on Pin 1 of the 20-pin IDC connector.
2. Or, an external 3.3v supply must be connected to Pin 1 of the 20-pin IDC connector

If these two options aren't possible, then consider the SWD debugger jumper method in paragraph 2.3.2.2.

⚠ Note:

1. Debuggers that have built in isolation are not supported, because the 2 isolation circuits will interfere with each other.
2. Note that Pin 1 on the ET-IARISO-1 has a square solder connection on the bottom of the adapter board.

Now, use a 4-wire cable to connect the PAC EVK 4-pin SWD header labelled [+ SD SCL -] to the ET-UARTISO 4-pin UART header labelled [V D C G]. V D C G stands for Vcc, SD, SCL, Gnd respectively.

Now connect the 20-pin ribbon cable of the SWD debugger to the ET-IARISO-1 board. Pin 1 of the ET-IARISO board is at the top and represented by the blue wire of the ribbon cable in the figure above. Connect the other end of the 20-pin ribbon cable to the 20 pin IDC connector on the SWD debugger.

Now proceed to paragraph 3 to build and flash the FOC firmware image.

2.3.2.2 SWD Debugger Connection Using Jumper Wires

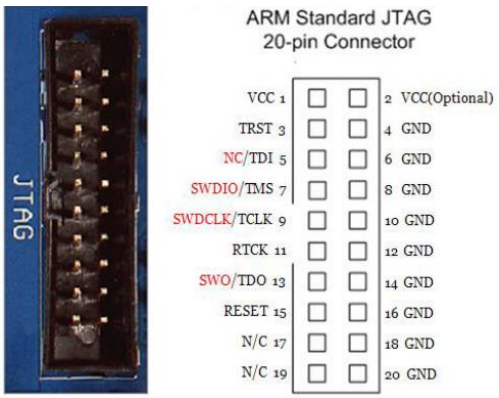
Use of jumper wires to connect the SWD Debugger should be considered if any of the following are true:

1. an ET-UARTSWD adapter is not available
2. an ET-UARTISO-1 is not available
3. an ET-UARTISO-1 is available, but 3.3V can't be supplied to pin 1 of the adapter
4. The SWD Debugger has built in isolation

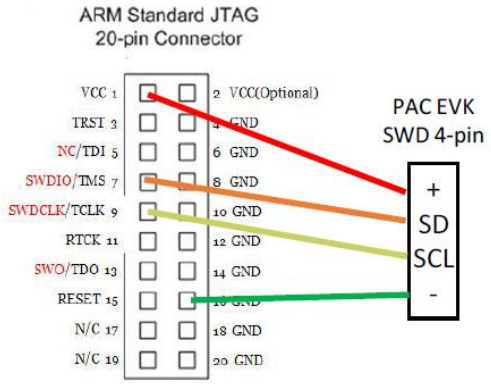
⚠ IMPORTANT Notes:

1. When connecting a non-isolated SWD Debugger to a high voltage PAC EVK using jumpers, care must be taken or injury and PC damage could occur. For PAC22140 EVK, the default SWD signal voltage is 5V.
2. If the SWD debugger doesn't support 5V operation, do NOT connect the debugger via this jumper method.

Connection of the SWD Debugger using jumper wires involves connecting 4 jumpers to the PAC EVK. A standard ARM JTAG/SWD 20-pin IDC connector pin out is shown below.



Connect VCC, SWDIO, SWCLK, and a GND using jumper wires to the 4 signals of the PAC EVK 4-pin SWD header labelled [+ SD SCL -]. This is illustrated in the block diagram on the right.



The picture below shows the actual connection of a J-Link debugger and the PAC22140 EVK. The jumper wires in the picture are as follows:

- Pin 1 - Vcc = Red

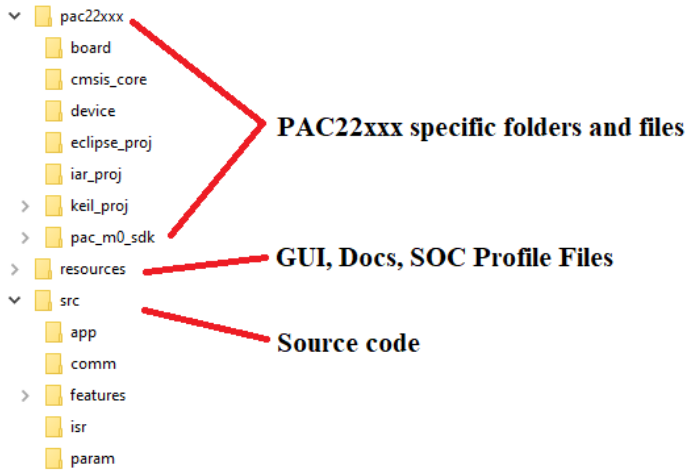
- Pin 7 - SWDIO = Orange
- Pin 9 - SWCLK = Yellow
- Pin 15 - Gnd = Green

3 BUILD AND FLASH THE FIRMWARE

The next few steps will configure, build, and flash the firmware to the device on a PAC EVK

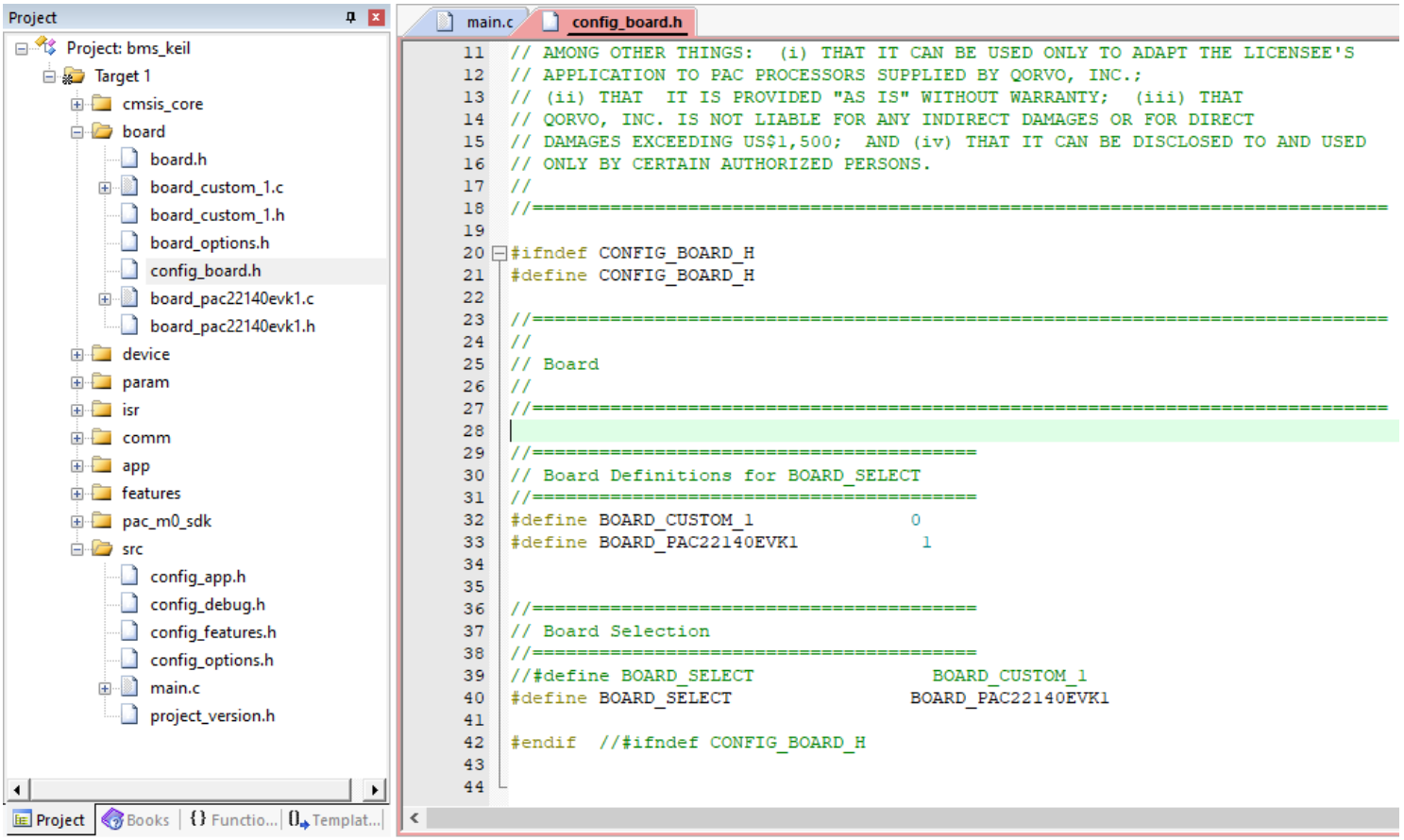
3.1 Choose Device Family and IDE Project

The PAC BMS FW contains common source used for various PAC devices and several IDEs; this code is stored in the src folder. The device dependent code and IDE projects are stored inside the device family folders. The figure to the right shows the folder hierarchy of the PAC BMS FW. First choose the device family, either PAC22xx or PAC25xx. Then, navigate down to the project folder for one of the IDEs: Keil, IAR, or Eclipse. Using the desired IDE, open the project contained in the associated folder.



3.2 Configure Firmware

Navigate within the folder view of the IDE project to the src folder as shown in the figure below. Several configuration header files are



provided to configure the BMS FW. Open the config_board.h file in board folder and navigate to the “Board Selection” portion of the code. Uncomment the BOARD_SELECT for the desired PAC EVK and comment out all other BOARD_SELECT lines. In the example below, the PAC22140EVK1 has been chosen.

3.3 Build and Flash the Firmware

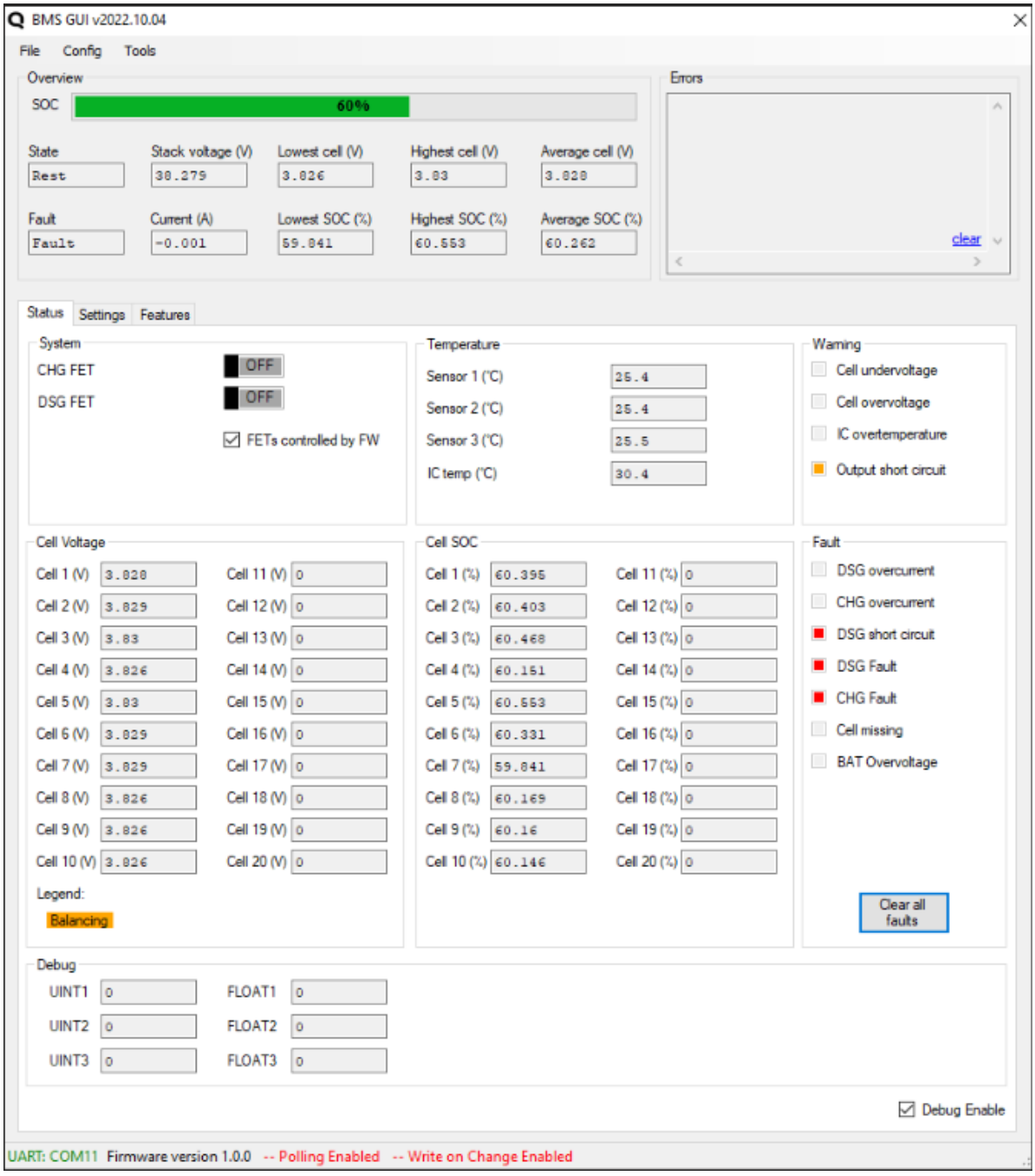
- From the IDE build the firmware image
- Verify the build is successful with no errors before moving to the next step.
    - Some IDEs will produce some warnings which are ok to ignore.
- Make sure the Power Supply is turned on and the IDE is configured for the debugger being used
- Flash the firmware image to the device

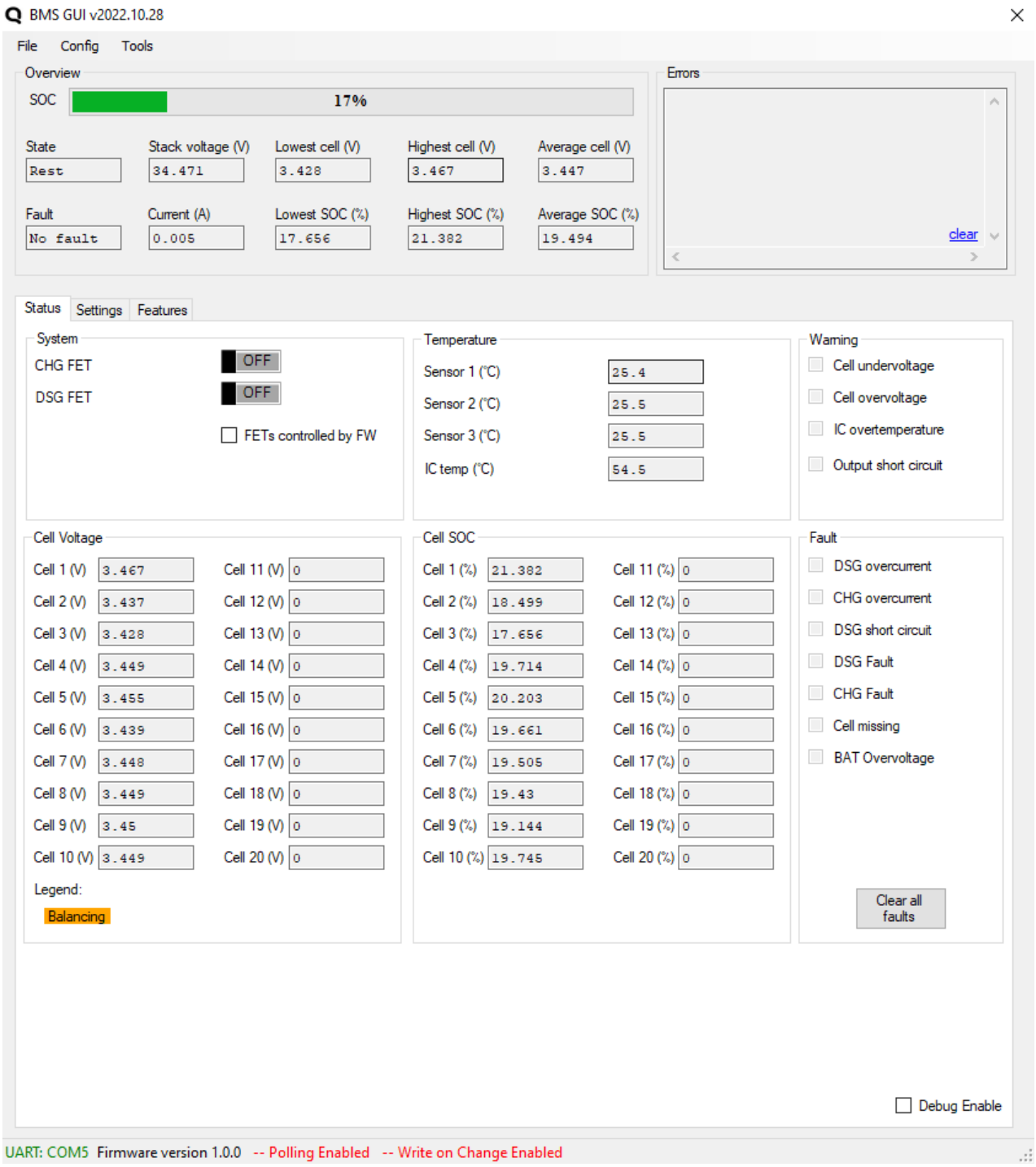
Some IDEs won’t reset the PAC device after Flashing and this has to be done manually. Turn the power supply off and back on to reset the PAC device.

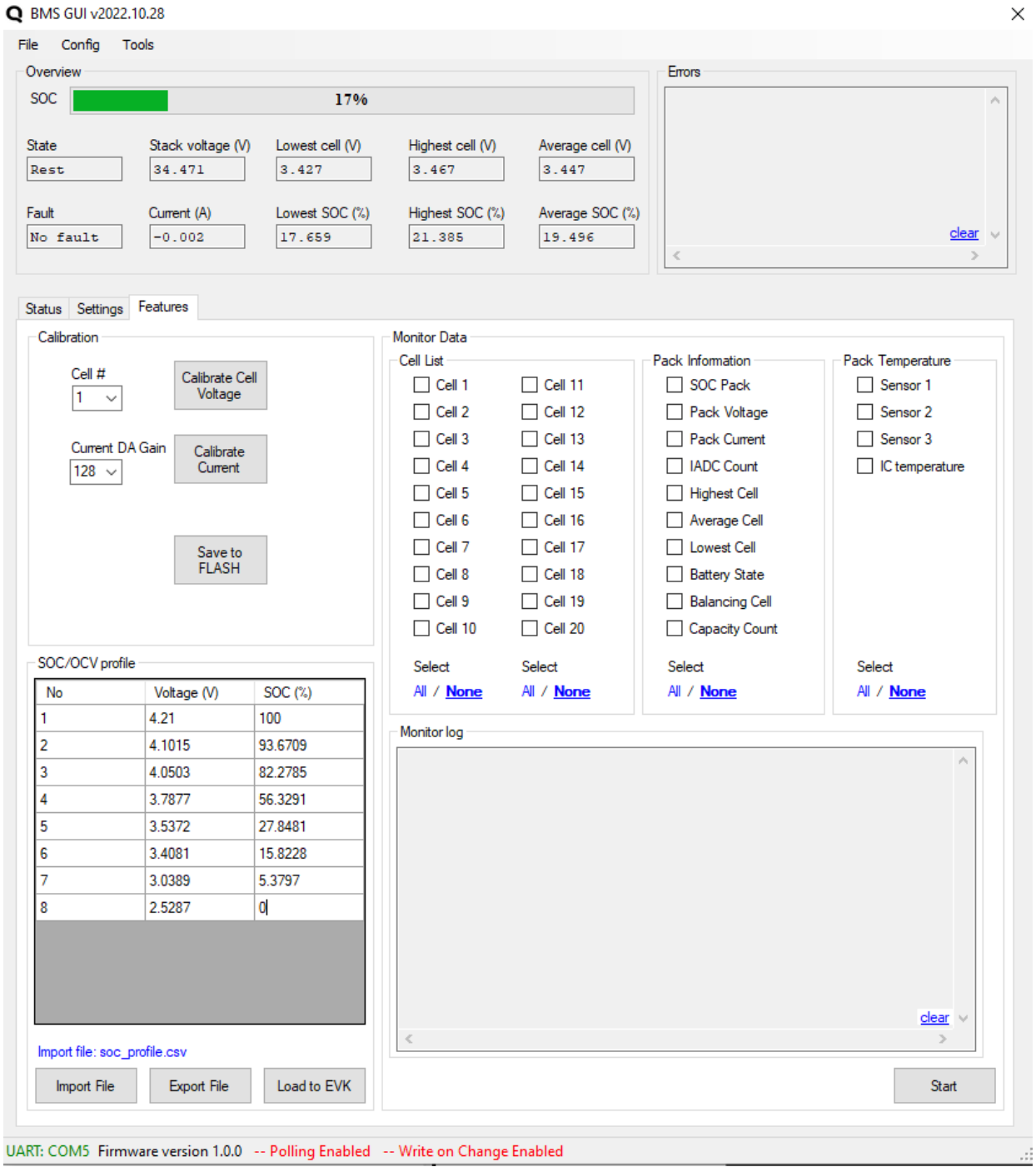
Now the PAC will be running and waiting for communication from the GUI.

4 BMS GUI CONFIGURATION

In the resource folder, start PAC BMS GUI by executing the file pac_bms_gui.exe. The application will open and be displayed as shown below.



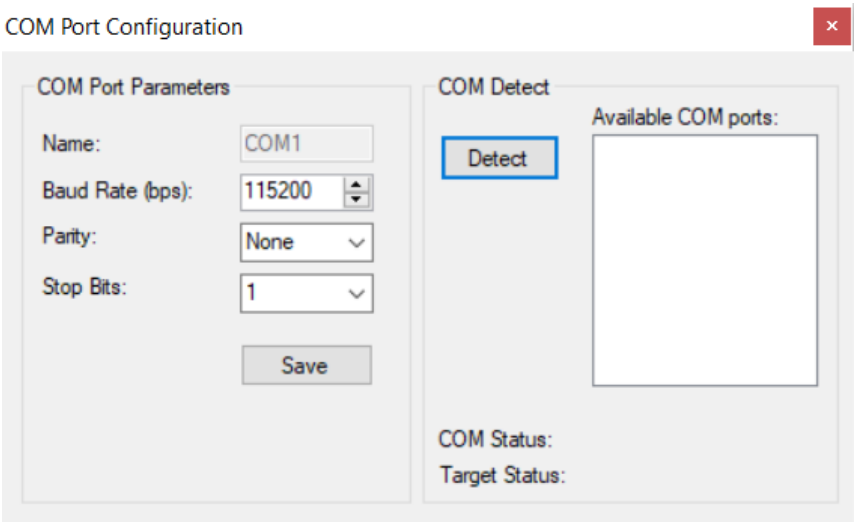




4.1 COM Port Configuration

The UART is the interface to communicate between GUI and PAC FW. Before the COM port is configured and the GUI connects to the PAC device, the GUI will display a “NO COMM” message in the bottom left corner of the GUI as shown on the right.

Configure the COM port by selecting **Config->Com Port Config**…”. A dialog box will be displayed as shown below.



The PAC BMS FW is configured to use the default UART parameter values shown in the **COM Port Parameters** dialog box, namely 115200 bps, no Parity, and 1 Stop Bit.

Select the correct COM port from the **Available COM ports** window on the right. If the correct COM port is selected, the **COM Status** will change to read **COM port open**. Additionally, if the GUI is able to communicate with the firmware, the **Target Status** will change to display the firmware version. If the firmware version is not displayed after several seconds, reset the device by turning the power supply off and back on. Some IDEs won’t reset the device after Flashing and this has to be done manually.

Once the Firmware Version is displayed, click the **Save** button to save the COM port configuration so that it will be used automatically the next time the GUI is started.

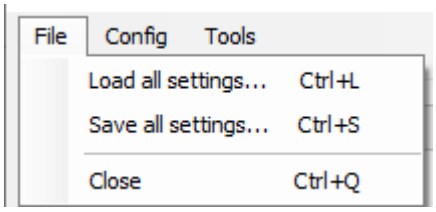
After exiting the **COM Port Parameters** dialog box, the GUI will display the COM port configuration, communication status in the bottom left corner and will automatically enable the **“Poll Status”** and **“Write on Change Enabled”** features.

When **Poll Status** is enabled, some data, such as firmware status and cells voltage, current and SOC of battery pack, will be automatically read from the firmware periodically. When **“Write on Change Enable”** is active, new parameter values are automatically written to the firmware as soon as they are changed by the user.

5 OPERATING THE BMS ON BATTERY PACK

This section will describe the process to operate a BMS from the GUI using a characterized SOC curve file for the desired battery package and PAC EVK. If the SOC curve file has not been provided, the BMS will use a SOC curve profile of an INR21700-40T Li-Ion battery cell as an example while operating.

5.1 Load The Configuration File

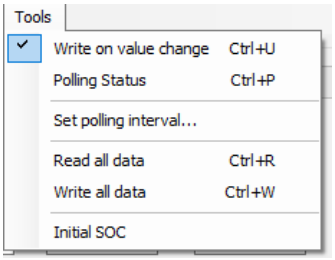


The BMS FW is designed so that most new battery package parameters and configuration can be written from the GUI without having to recompile the firmware. If a pre-configured parameter file is available for the desire battery and PAC EVK, load it by clicking **File->Load all settings**… and navigating to the file.

To view the parameters and configuration for the battery package and EVK, click on the various Tabs in the upper right portion of the GUI such as **Setting**, and **Features**.

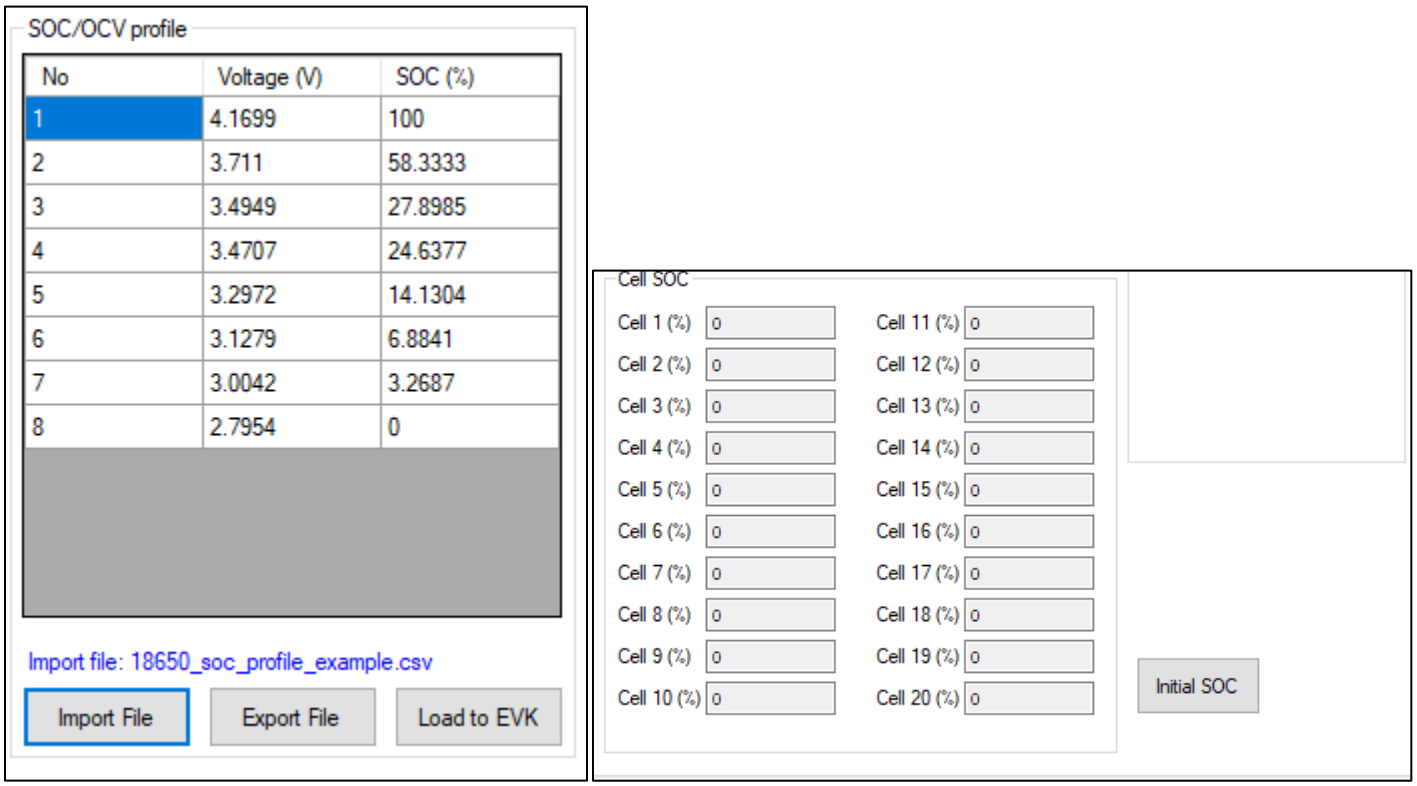
Note: Since "**Write on Change Enable**" is active, new parameter values entered from the GUI will be automatically written to the firmware as soon as they are changed by the user.

5.2 Load Battery SOC Profile File



The BMS SOC operates on OCV and Columb Counting algorithm. In application, there are many types of cell battery, and they have a corresponding SOC curve. The BMS FW will use these SOC profiles for initialization then keep monitoring the power capacity which is charged/discharged to battery pack. The particular SOC profile may be imported to GUI from a file which are exported from GUI already, load the file by clicking **Import File** button in **SOC/OCV profile** panel of **Features** tab. After file is loaded into data grid view, it's able to edit and export for next using. Finally, click **Load to EVK** button to send this reference points of SOC profile to device FW.

To initialize SOC starting point which is match to SOC profile curve, click **Initial SOC** in **Tools** before enable FETs control.

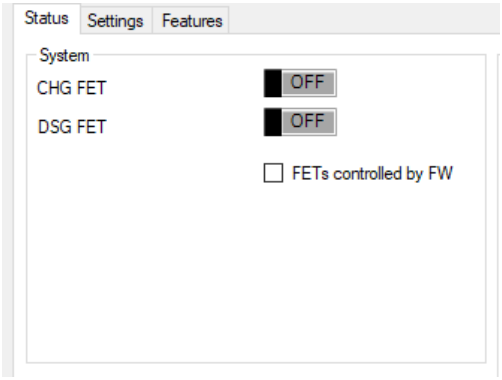


5.3 Enable Auto FETs control

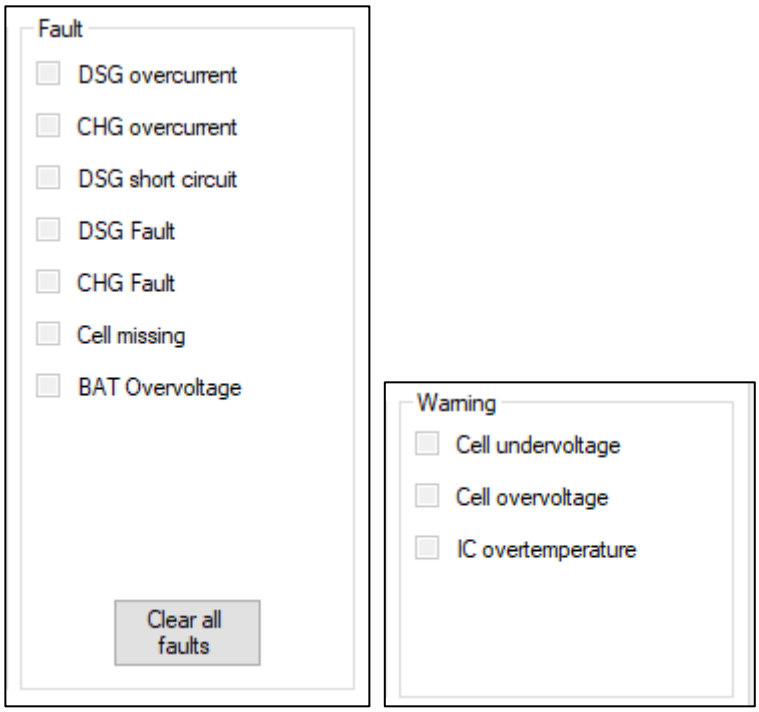
When device startups and GUI connecting successfully, the battery package are still not connected to outside via FETs. All current parameters and setting in FW will be read and shown on GUI. The cells voltage of battery pack, NTCs, die temperature and all any Fault ore Warning status are displayed in **Status** tab.

The BMS GUI default set CHG FET and DSG FET in disabling on startup, all FETs are controlled automatically by FW when user ticks the tick box **FETs controlled by FW** in **Status** tab.

In operating, the status and error will displayed in **Errors** log message



The GUI will also show Faults indicator in **Status\Fault panel**. The occurred faults can be cleared by clicking the **Clear All Faults** button.



5.4 Next Steps

Now that the BMS battery is operating, consult the following documents for additional information:

- Guide_BMS_Firmware_Overview.pdf – describes the functionality and architecture of the firmware
- PAC22140EVK1 User Guide.pdf- describe the setup and evaluation guide.
- Consult the PAC hardware User's Guide for your specific PAC hardware.

6 ADDITIONAL GUI INFORMATION

6.1 Load/Save Parameters and Configuration

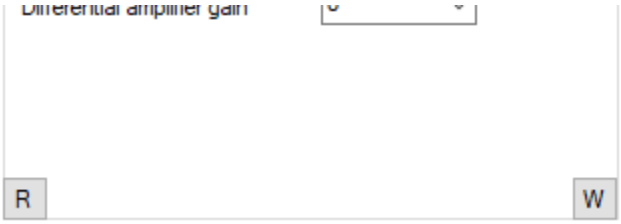
The GUI XML parameters files can be loaded or saved using the Load all settings or Save all settings features, respectively, accessed from the File menu.

6.2 Read / Write Parameters

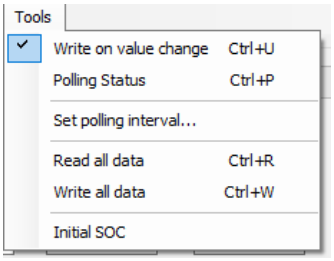
If the GUI is connected to the firmware, it will automatically enable the **Poll Status** and **Write on value change** features and indicate this in the bottom left corner.

When Poll Status is enabled, some data, such as firmware status and motor speed, will be automatically read from the firmware periodically. When Write on value change is enabled, new parameter values are automatically written to the firmware immediately.

An entire parameter group can be read or written using the **R** or **W** buttons, respectively, located at the bottom of each parameter group.



All of the parameters can be read or written using the **Read all data** or **Write all data** features, respectively, accessed from the Tools menu. The GUI will indicate the operation is in progress in the bottom left corner. Any problems reading or writing any parameter will be reported in the Errors message box.

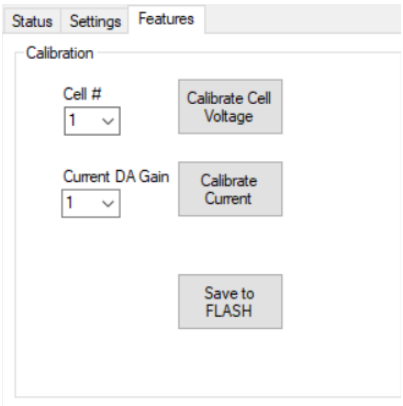


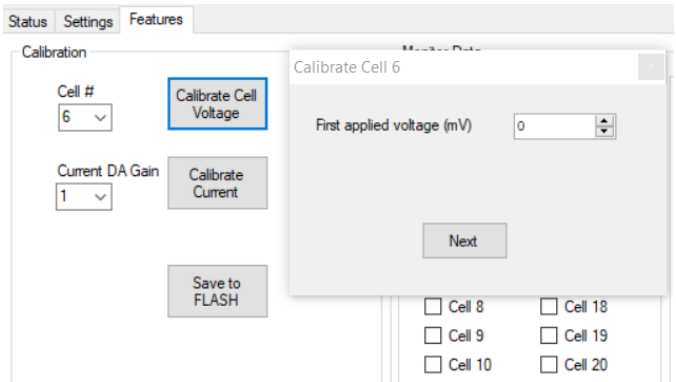
6.3 Calibration

The PAC devices for BMS contain a Configurable Analog Front-End (CAFETM) that can be used to sense battery pack current using an integrated Differential Amplifier (DA) and 16-bit Sigma-Delta ADC. Moreover. It can also sense voltage and temperature internally through a MUX. The gain and offset calibration are written in INFOR Flash already to instant using. However, the GUI features calibration function to get more accuracy Gain and Offset calibration factors due to it includes all sense resistor and EVK components error. The new calibration factors are stored in defined Flash area.

6.3.1 Cells Voltage Calibration

To implement the recalibration for a cell, click and select Cell index in **Cell#** drop box. A dialog is shown to fill a voltage which is measured at this cell by a good DMM in mV unit then click **Next** button and follow until complete. The calibration procedure requires 2 points, 3.2V and 4.2V are recommended.

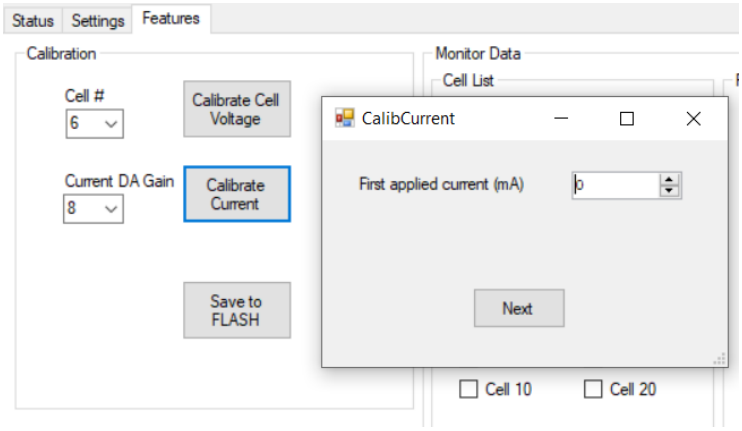




Click **Save to FLASH** button if you have no more other cell calibration.

6.3.2 Current Calibration

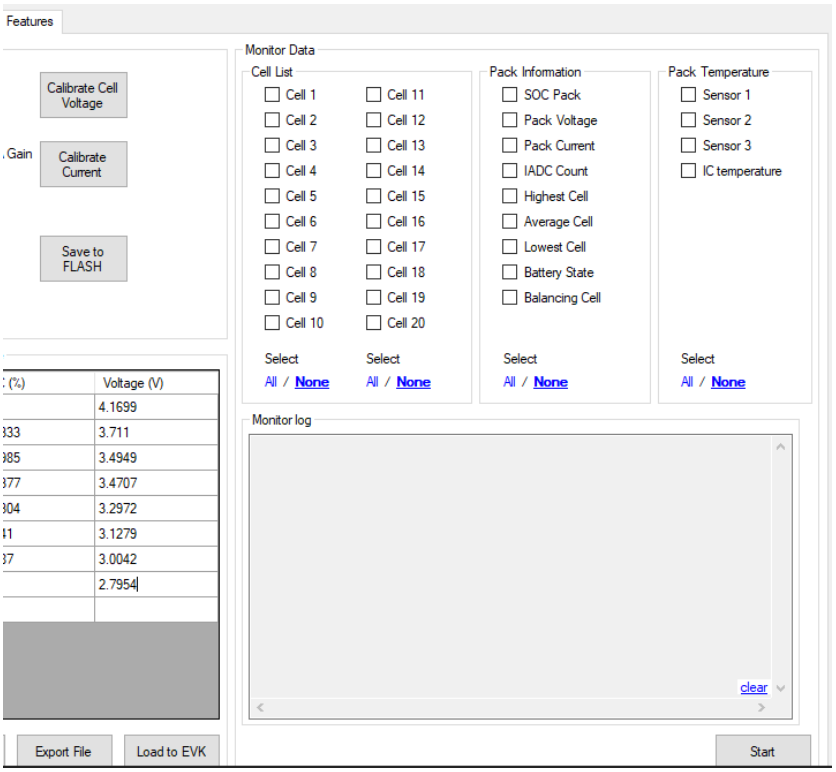
To implement the recalibration for current ADC at a DPGA gain setting, click and select Gain setting of DPGA in **Current DA Gain** drop box. A dialog is shown to fill a current which is measured at this cell by a good DMM in mA unit then click **Next** button and follow until complete. The calibration procedure requires 2 points, 0A and a charging/discharging are recommended. The current should be high and NO over permitted DMM. The input of IADC is I*R_shunt*Gain and the signal input of IADC is better if the Load current is enough. The negative value is the discharging condition.

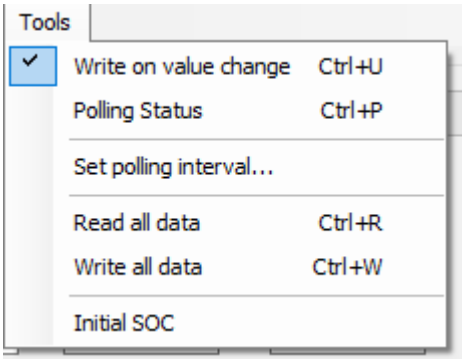


Click **Save to FLASH** button if you have no more other cell calibration.

6.4 Datalogger

The BMS GUI supports the monitoring and storing the monitoring result in datalog file in .csv format. The Datalog function is in **Features** tab. The available parameters for logging are Cells voltage as **Cell List** group, Pack Current and other parameters of battery pack in **Pack Information** group, the sensors temperature and die temperature are in **Pack Temperature** group.
Click to select the desired parameters which will be stored in log file. Click to select **All/None** text select to quick select/unselect all corresponding parameters.
The time interval of logging data is the polling time interval which is set as 1000ms as default. Open **Set polling interval** in **Tool** menu to change this period in millisecond.



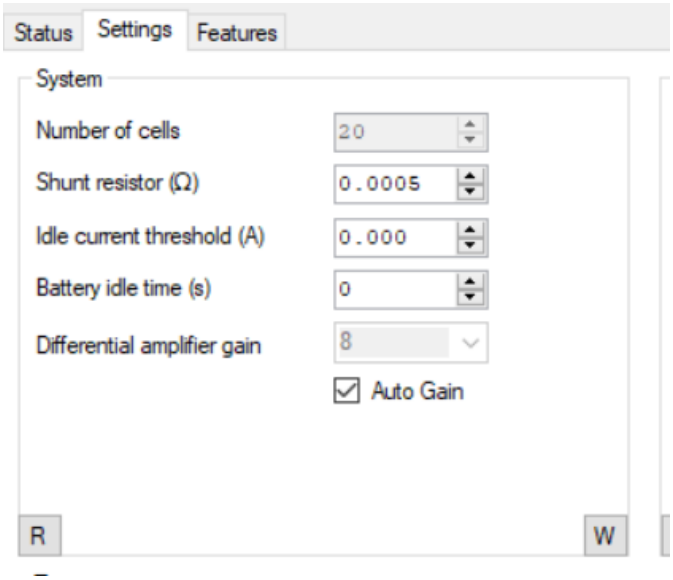


Click **Start** button to start logging file after all expected parameters are selected, the button **Start** will change to **Stop** button. The log file is saved in folder which contains execute BMS_GUI.exe. To stop the logging process, click **Stop** button. The image below is an example output format of data log file.

	A	B	C	D	E	F	G
1	Dat	24/08/2022					
2	Tim	9:39:54					
3	Inte	1000					
4	No	CELL_01 [V]	SOC_01 [%]	CELL_02 [V]	SOC_02 [%]	CELL_03 [V]	SOC_03 [%]
5	1	3.902	75.637	3.893	74.87	3.896	75.089
6	2	3.902	75.633	3.893	74.866	3.896	75.084
7	3	3.902	75.627	3.893	74.86	3.896	75.079
8	4	3.902	75.621	3.893	74.853	3.896	75.073
9	5	3.902	75.615	3.893	74.847	3.896	75.067
10	6	3.902	75.609	3.893	74.841	3.896	75.061
11	7	3.902	75.603	3.893	74.835	3.896	75.054
12	8	3.902	75.597	3.893	74.829	3.896	75.049
13	9	3.902	75.591	3.893	74.823	3.896	75.042

6.5 Auto Gain Current DPGA

As default, the DPGA of current sensing signal is fixed at a Gain setting (x8 typical) during the current measurement. To improve the accuracy of IADC at low current value, the BMS supports the *Auto Gain* feature. When this feature is enabled, the gain of DPGA is configured automatically such that the voltage at sensing resistor is gained as much as possible in valid range of IADC.



Click to select tick box **Auto Gain** in **System** Group of **Settings** Panel to enable or disable Auto Gain feature. While the BMS operates with Auto Gain setting, the OCDDAC and OCCDAC are auto changed. The OCC and OCD protects may be disabled if the desired protection threshold is out of DAC range at this DPGA gain. These protections are re-enabled with corresponding DAC setting when the threshold is valid on OCCDAC/OCDDAC range. Don't set the OCC, OCD protection current threshold at so small due to hit fault by Com protection offset.

7 LEGAL INFORMATION

For the latest specifications, additional product information, worldwide sales and distribution locations:

Web: www.qorvo.com

Tel: 1-844-890-8163

Email: customer.support@qorvo.com

The information contained in this Application Note and any associated documents ("Application Note nformation") is believed to be reliable; however, Qorvo makes no warranties regarding the Application Note Information and assumes no responsibility or liability whatsoever for the use of said information. All Application Note Information is subject to change without notice. Customers should obtain and verify the latest relevant Application Note Information before placing orders for Qorvo® products. Application Note Information or the use thereof does not grant, explicitly, implicitly or otherwise any rights or licenses to any third party with respect to patents or any other intellectual property whether with regard to such Application Note Information itself or anything described by such information.

APPLICATION NOTE INFORMATION DOES NOT CONSTITUTE A WARRANTY WITH RESPECT TO THE

PRODUCTS DESCRIBED HEREIN, AND QORVO HEREBY DISCLAIMS ANY AND ALL WARRANTIES WITH RESPECT TO SUCH PRODUCTS WHETHER EXPRESS OR IMPLIED BY LAW, COURSE OF DEALING, COURSE OF PERFORMANCE, USAGE OF TRADE OR OTHERWISE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Without limiting the generality of the foregoing, Qorvo® products are not warranted or authorized for use as critical components in medical, lifesaving, or life-sustaining applications, or other applications where a failure would reasonably be expected to cause severe personal injury or death.

Applications described in the Application Note Information are for illustrative purposes only. Customers are responsible for validating that a particular product described in the Application Note Information is suitable for use in a particular application.