LTC6813-1
18-Channel Battery-Stack Monitor with Daisy-Chain Interface

DESCRIPTION

Demonstration circuit 2350B features the LTC®6813-1, 18-channel battery-stack monitor. Multiple boards can be linked through a 2-wire isolated serial interface (isoSPI™) to monitor a long series of cells in a stack. The DC2350B demo board also features reversible isoSPI enabling a redundant communication path. The PCB, components, and DuraClik connectors are optimized for low EMI susceptibility and emissions.

The DC2350B can communicate to a PC by connecting a DC2792B dual master isoSPI together with DC2026C Linduino® One. The DC2026 must be loaded with the appropriate program (called a **sketch**) to control the battery-stack monitor IC and receive data through a USB serial port. The DC2026 provides a standard SPI interface which can be translated to isoSPI and then connected to a DC2350B isoSPI port (J4 or J5 connector). The DC2792B companion board provides two SPI-isoSPI channels for reversible operation.

Design files for this circuit board are available.

PERFORMANCE SUMMARY

Specifications are at T_A = 25°C

PARAMETER	MIN	TYP	MAX	UNITS
V^+ Supply Voltage	16	60	90	V
V^+ to C18 Voltage	–0.3			V
V^+ to C12 Voltage			40	V
C13 Voltage	2.5			V
C7 Voltage	1			V
V_{REG} Supply Voltage	4.5	5	5.5	V
V_{REF1} 1st Reference Voltage, No Load	3.0	3.15	3.3	V
V_{REF2} 2nd Reference Voltage, 5k Load to V^-	2.993	3	3.007	V
V_{BIAS} I_{BIAS} Voltage, READY/ACTIVE State	1.9	2.0	2.1	V
CPIN Input Range	0		5	V
Cell Count			18	

HARDWARE SETUP

Wiring J1 Connector

The DC2350B demo board connector pinout is critical; correct wiring must be followed to avoid the risk of damaging the DC2350B demo board.

When connected to a battery-stack, power for the DC2350B is provided by the cell group being monitored. To connect the cell group, separate the screw-terminal block section from the J1 connector. Then, insert the cell-voltage connections or resistors into the screw-terminal clamping contacts. These connections provide the power and input stimulus for the battery-stack monitor IC.

Cell-voltages are wired to J1 starting from position 1 (most negative potential of the group). Please reference the appropriate demo board J1 connector pinout in Table 1.

Alternatively, resistors can be used to simulate battery cell-voltages. 100Ω 2W or equivalent resistors are recommended because 100Ω (or lower values) typically will not induce measurement errors and the 2W (or greater rating) will keep the resistor temperatures low preventing power dissipation damage.

DC2350B Eighteen Resistor Connections

Carefully connect eighteen 100Ω resistors onto the screw-terminal block between each CPIN input clamping contact from position 1 to position 19 as shown in Table 1, DC2350B J1 pinout. Provide a stack-equivalent power supply connection to position 19 (positive) and position 1 (negative). The power supply may be adjusted to provide the desired nominal cell-voltage (i.e., 59.4V will be 3.3V per cell).

Table 1. DC2350B J1 Pinout

J1 PIN	CPIN INPUT
1	C0/V$^-$
2	C1
3	C2
4	C3
5	C4
6	C5
7	C6
8	C7
9	C8
10	C9
11	C10
12	C11
13	C12
14	C13
15	C14
16	C15
17	C16
18	C17
19	C18

DC2350B SERIAL INTERFACE OPTIONS

The isoSPI is the only communication option to DC2350B. Due to the custom EMI optimized isoSPI cable with DuraClik connectors, it is highly recommended to use DC2792B dual master isoSPI demo board or equivalent for easy plug-and-play operation. The DC2792B dual master isoSPI demo board can be connected as a typical single-ended isoSPI bus master or to both ends of a reversible configuration with two isoSPI bus masters. Refer to demo manual DC2792B for usage details.

DC2792B to DC2350B Typical isoSPI Connection

A typical isoSPI connection begins with the isoSPI master connected to the first (or **bottom**) DC2350B. Additional DC2350B boards can be daisy-chained onto the isoSPI bus. Communication begins from the first (or **bottom**) DC2350B then to the next **upper** DC2350B and, finally, to the last (or **top**) DC2350B.

Figure 1 shows the following connections for two boards on a stack interfaced to a PC.

1. Connect a USB cable from the PC USB port to the DC2026 J5 connector.
2. Connect the DC2026 to the DC2792B dual master isoSPI demo board.
   a. Connect a 14-pin ribbon cable from the DC2026 J1 header to the DC2792B J1 header.
3. Connect the DC2792B to the DC2350B. This DC2350B is the first (or **bottom**) board of the stack.
   a. Connect a 2-wire twisted-pair patch cable from the DC2792B J2 MAIN DuraClik connector to the **bottom** DC2350B J4 isoSPI A DuraClik connector.

   CAUTION! The 2-wire twisted-pair patch cable with the DuraClik end plugs have 1mm thick center locking tabs on the wiring side that must be pressed down to release from the DuraClik receptacles. Failure to do so may damage the cable and prevent board-to-board isoSPI communication.
4. Connect or daisy-chain the DC2350B to another DC2350B in isoSPI mode. This DC2350B is the last (or **top**) board of a two-board stack. More DC2350B

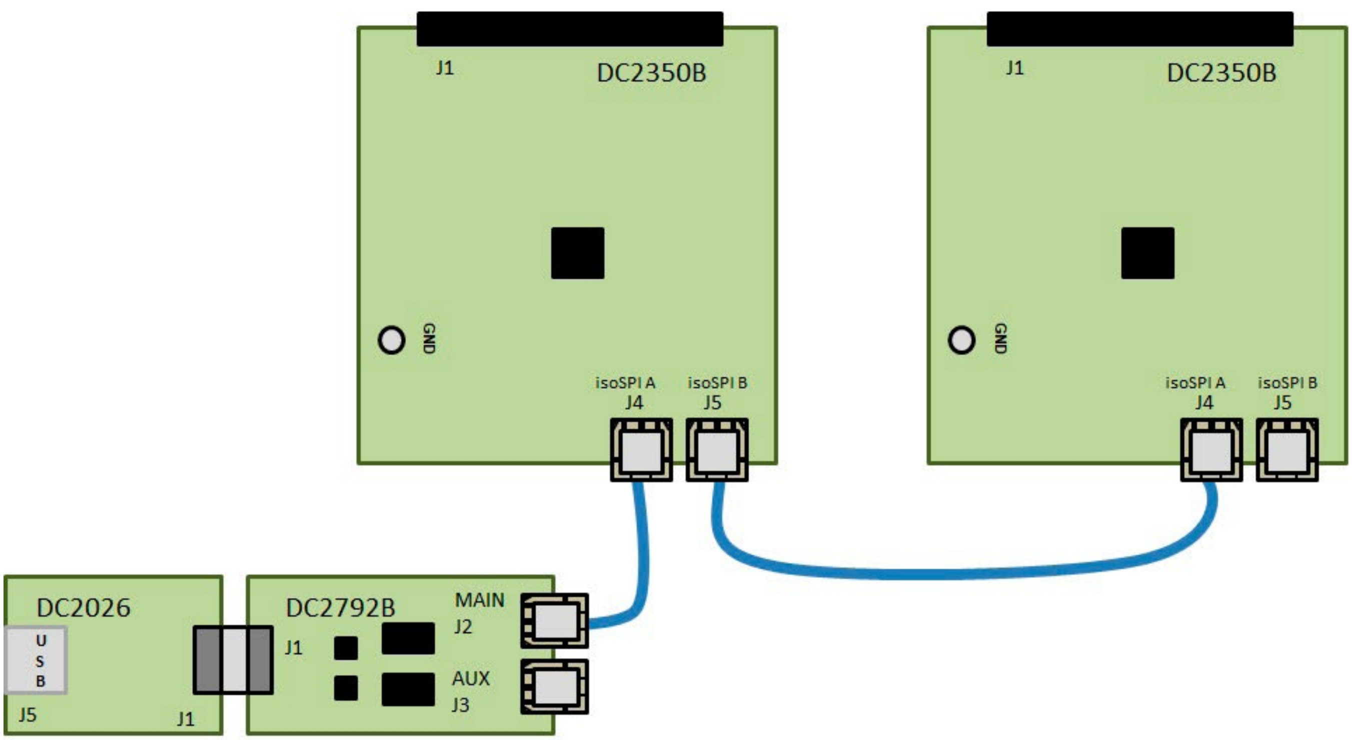

Figure 1. DC2792B Typical isoSPI Connection to the *Bottom* DC2350B in a Two-Board DC2350B Stack

DC2350B SERIAL INTERFACE OPTIONS

upper boards can be daisy-chained together in the same manner.

a. Connect a 2-wire twisted-pair patch cable from the **bottom** DC2350B J5 isoSPI B DuraClik connector to the next **upper** or **top** DC2350B J4 isoSPI A DuraClik connector.

5. **CAUTION!** Prevent damage to the DC2350B. Refer to Table 1 and confirm that the cell-voltage connections to the screw-terminal block matches the DC2350B J1 pinout.

   a. Plug the screw-terminal blocks into the J1 cell-voltage connectors.

6. Refer to the Software Setup section of this demo manual to properly setup the PC with the Arduino IDE software to allow communication to the DC2350B boards.

DC2792B to DC2350B Reverse isoSPI Connection

A reverse isoSPI connection begins with the isoSPI Master connected to the last (or **top**) DC2350B. Additional DC2350B boards can be daisy-chained onto the isoSPI bus. Communication begins from the last (or **top**) DC2350B then to the next **lower** DC2350B and, finally, to the first (or **bottom**) DC2350B.

Figure 2 shows the following connections for two boards on a stack interfaced to a PC.

1. Connect a USB cable from the PC USB port to the DC2026 J5 connector.
2. Connect the DC2026 to the DC2792B dual master isoSPI demo board.

   a. Connect a 14-pin ribbon cable from the DC2026 J1 header to the DC2792B J1 header.

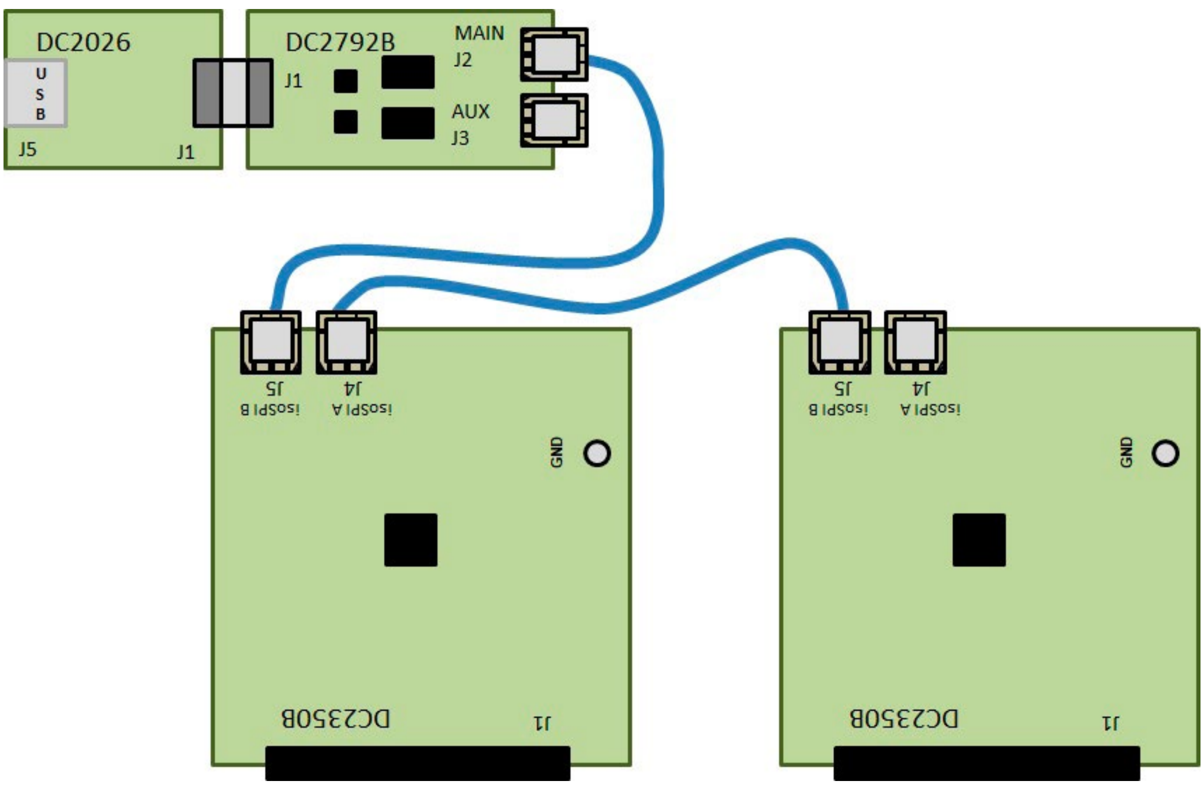

Figure 2. DC2792B Reverse isoSPI Connection to the *Top* DC2350B in a Two-Board DC2350B Stack

Rev. 0

DC2350B SERIAL INTERFACE OPTIONS

3. Connect the DC2792B to the DC2350B in isoSPI mode. This DC2350B is the last (or **top**) board of a two-board stack.

   a. Connect a 2-wire twisted-pair patch cable from the DC2792B J2 MAIN DuraClik connector to the **top** DC2350B J5 isoSPI B DuraClik connector.

   CAUTION! The 2-wire twisted-pair patch cable with the DuraClik end plugs have 1mm thick center locking tabs on the wiring side that must be pressed down to release from the DuraClik receptacles. Failure to do so may damage the cable and prevent board-to-board isoSPI communication.

4. Connect or daisy-chain the DC2350B to another DC2350B in isoSPI mode. This DC2350B is the first (or **bottom**) board of a two-board stack. More DC2350B **lower** boards can be daisy-chained together in the same manner.

   a. Connect a 2-wire twisted-pair patch cable from the **top** DC2350B J4 isoSPI A DuraClik connector to the next **lower** or **bottom** DC2350B J5 isoSPI B DuraClik connector.

5. **CAUTION!** Prevent damage to the DC2350B. Refer to Table 1 and confirm that the cell-voltage connections to the screw-terminal block matches the DC2350B J1 pinout.

   a. Plug the screw-terminal blocks into the J1 cell-voltage connectors.

6. Refer to the Software Setup section of this demo manual to properly setup the PC with the Arduino IDE software to allow communication to the DC2350B boards.

DC2792B to DC2350B Redundant isoSPI Connection

A redundant isoSPI connection begins with the primary (or **main**) isoSPI Master connected to the first (or **bottom**) DC2350B and has a backup auxiliary (or **aux**) isoSPI master connected to the last (or **top**) DC2350B. Additional DC2350B boards can be daisy-chained between the two isoSPI masters on the isoSPI bus. Primary (or **main**) communication begins from the first (or **bottom**) DC2350B then to the next **upper** DC2350B and, finally, to the last (or **top**) DC2350. The backup auxiliary (or **aux**) communication begins in the reverse direction to provide coverage when a possible isoSPI daisy-chain break occurs.

Figure 3 shows the following connections for two boards on a stack interfaced to a PC.

1. Connect a USB cable from the PC USB port to the DC2026 J5 connector.

2. Connect the DC2026 to the DC2792B dual master isoSPI demo board.

   a. Connect a 14-pin ribbon cable from the DC2026 J1 header to the DC2792B J1 header.

3. Connect the DC2792B primary (or **main**) isoSPI master to the first (or **bottom**) DC2350B board of the stack.

   a. Connect a 2-wire twisted-pair patch cable from the DC2792B J2 MAIN DuraClik connector to the **bottom** DC2350B J4 isoSPI A DuraClik connector.

   CAUTION! The 2-wire twisted-pair patch cable with the DuraClik end plugs have 1mm thick center locking tabs on the wiring side that must be pressed down to release from the DuraClik receptacles. Failure to do so may damage the cable and prevent board-to-board isoSPI communication.

4. Connect or daisy-chain the DC2350B to another DC2350B in isoSPI mode. This DC2350B is the last (or **top**) board of a two-board stack. More DC2350B **upper** boards can be daisy-chained together in the same manner.

   a. Connect a 2-wire twisted-pair patch cable from the **bottom** DC2350B J5 isoSPI B DuraClik connector to the next **upper** or **top** DC2350B J4 isoSPI A DuraClik connector.

5. Connect the DC2792B auxiliary (or **aux**) isoSPI Master to the last (or **top**) DC2350B board of the stack.

   a. Connect a 2-wire twisted-pair patch cable from the DC2792B J3 AUX DuraClik connector to the **top** DC2350B J5 isoSPI B DuraClik connector.

DC2350B SERIAL INTERFACE OPTIONS

6. **CAUTION!** Prevent damage to the DC2350B. Refer to Table 1 and confirm that the cell-voltage connections to the screw-terminal block matches the DC2350B J1 pinout.

a. Plug the screw-terminal blocks into the J1 cell-voltage connectors.

7. Refer to the Software Setup section of this demo manual to properly setup the PC with the Arduino IDE software to allow communication to the DC2350B boards.

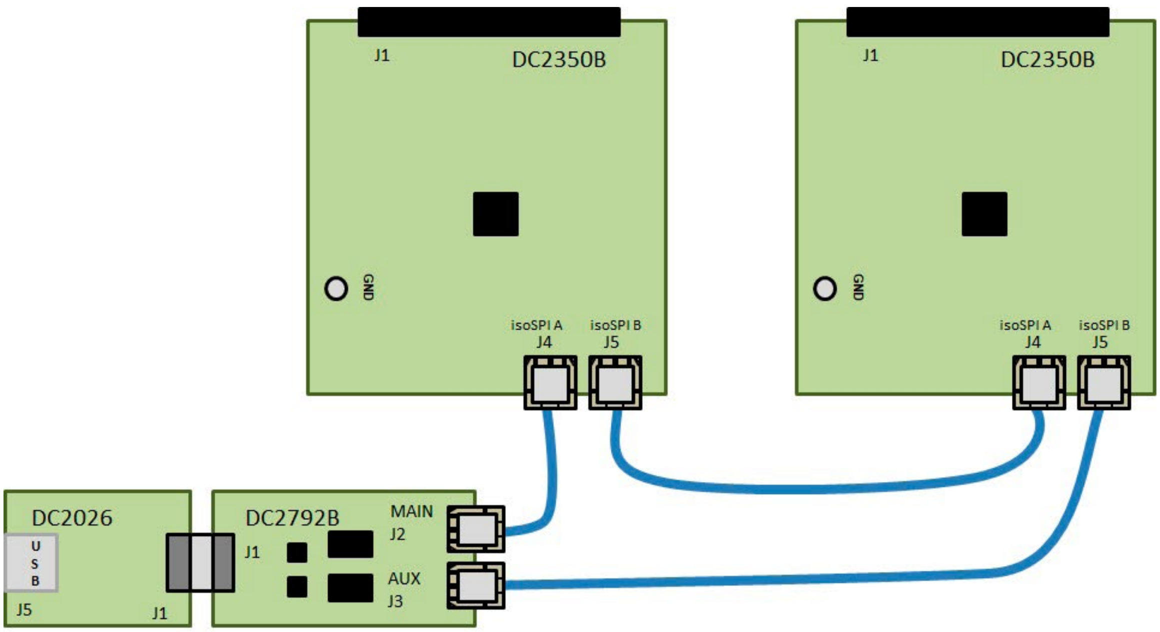

Figure 3. DC2792B Redundant isoSPI Connections to the *Bottom* and *Top* DC2350B in a Two-Board DC2350B Stack

SOFTWARE SETUP

The DC2350B can be controlled with the DC2026 Linduino One board together with DC2792B dual isoSPI Master or equivalent isoSPI transceiver. The DC2026 is part of the Arduino compatible Linduino platform that provides example code that will demonstrate how to control the multicell battery-stack monitor ICs. Compared to most Arduino compatible microcontroller boards, the DC2026 offers conveniences such as an isolated USB connection to the PC, built-in SPI MISO line pull-up to properly interface with the battery-stack monitor IC open drain SDO, and an easy ribbon cable connection for SPI communication through the DC2792B 14-pin QuikEval™ J1 connector.

Arduino IDE Setup

1. Download then install the Arduino IDE onto the PC. Detailed instructions can be found under the quick start tab.
2. Set the Arduino IDE to open BMS Sketchbooks. From within the Arduino IDE, click on **File** menu select **Preferences**. Then under **Sketchbook location:** select **Browse** and locate the path to the extracted **LTSketchbook.zip** file that was downloaded (see Figure 4).
3. Close then re-open the Arduino IDE to enable the use of the Sketchbook Location that was previously set.

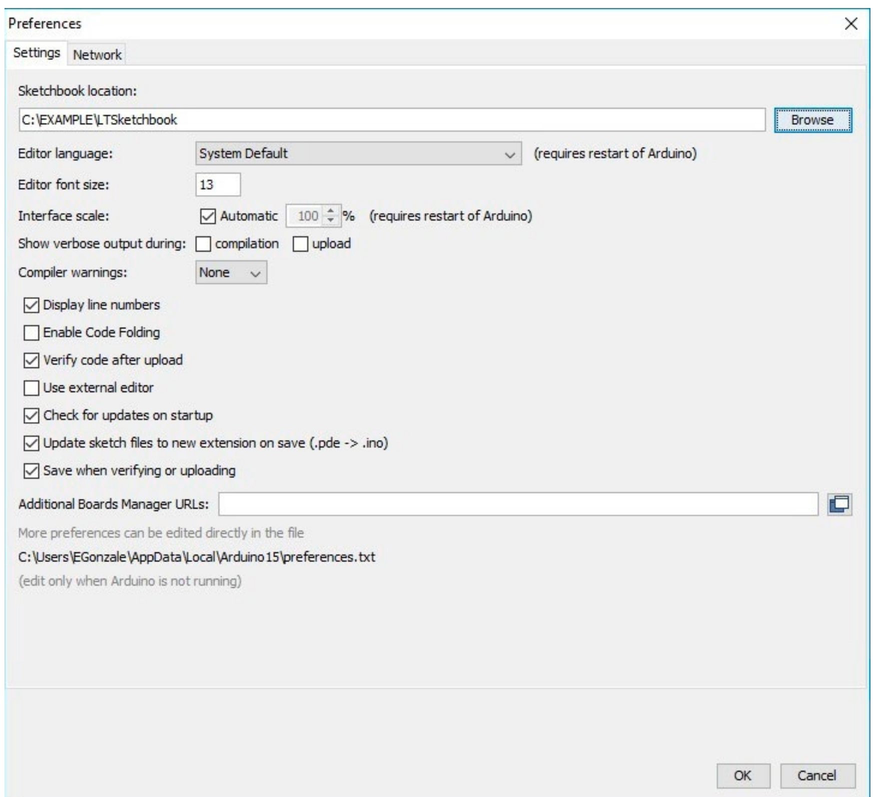

Figure 4. Sketchbook Location Path

SOFTWARE SETUP

4. Select the correct COM port to allow communication to DC2026 through USB. Under the **Tools** menu, select **Port** → Select the highest number **COMxx** with the √ check mark symbol. There may be more than one option; DC2026 is usually the highest COM port number. The PC screenshots (Figure 5) used in this example show the DC2026 connected to COM6.

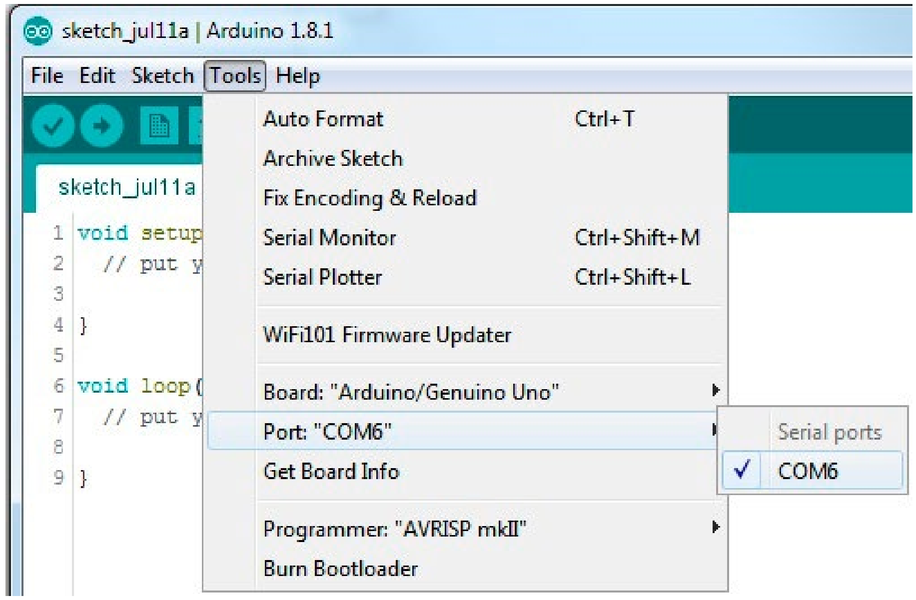

Figure 5. Selecting Correct COM

5. Select the correct Arduino compatible microcontroller board. Under the **Tools** menu, select **Board** → **Arduino/Genuino Uno** with the "•" black dot symbol (see Figure 6).

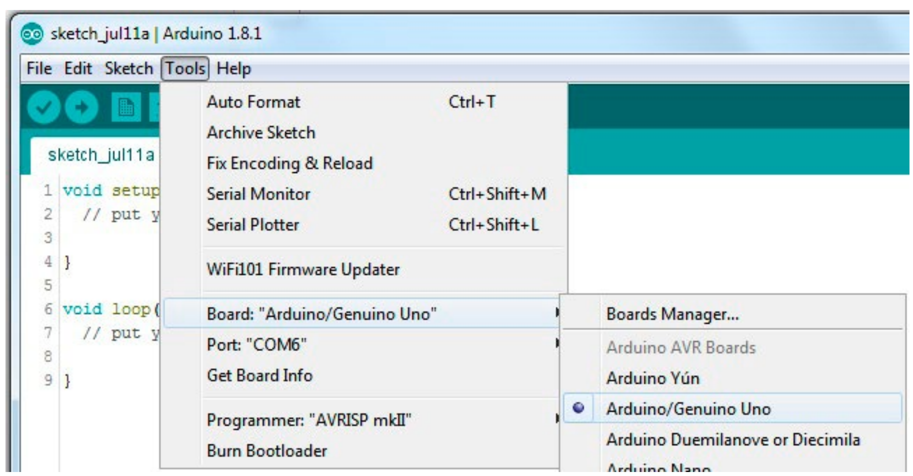

Figure 6. Selecting the Compatible Microcontroller Board

SOFTWARE SETUP

6. Open one of the programs or **sketches** associated with the DC2350B. In this example LTC6813 sketch will be opened. Under the **File** menu, select **Sketchbook → Part Number → 6000 → 6813 → DC2350AB** (see Figure 7).

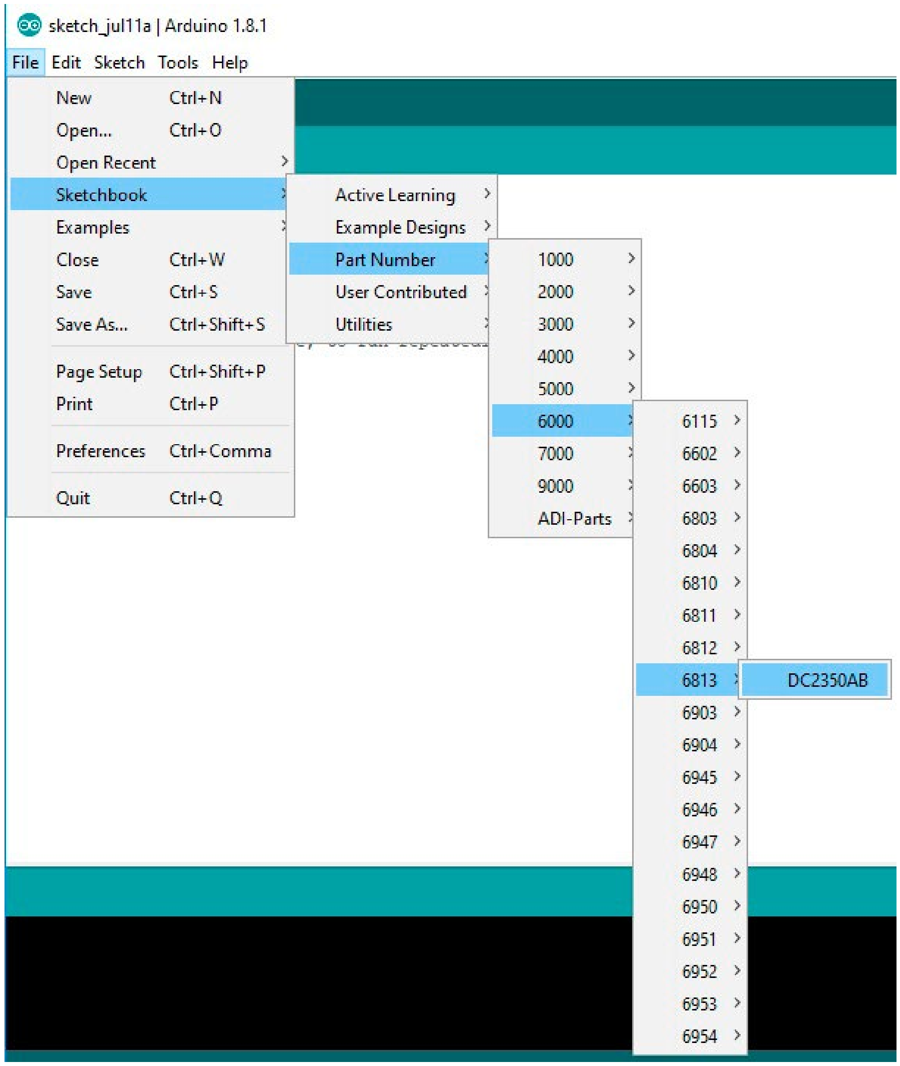

Figure 7. Selecting the LTC6813 Sketch

SOFTWARE SETUP

7. Upload the **DC2350AB** sketch onto the DC2026 by clicking on the **Upload** button on the top left corner. When this process is completed there will be a **Done Uploading** message on the bottom left corner (see Figure 8).

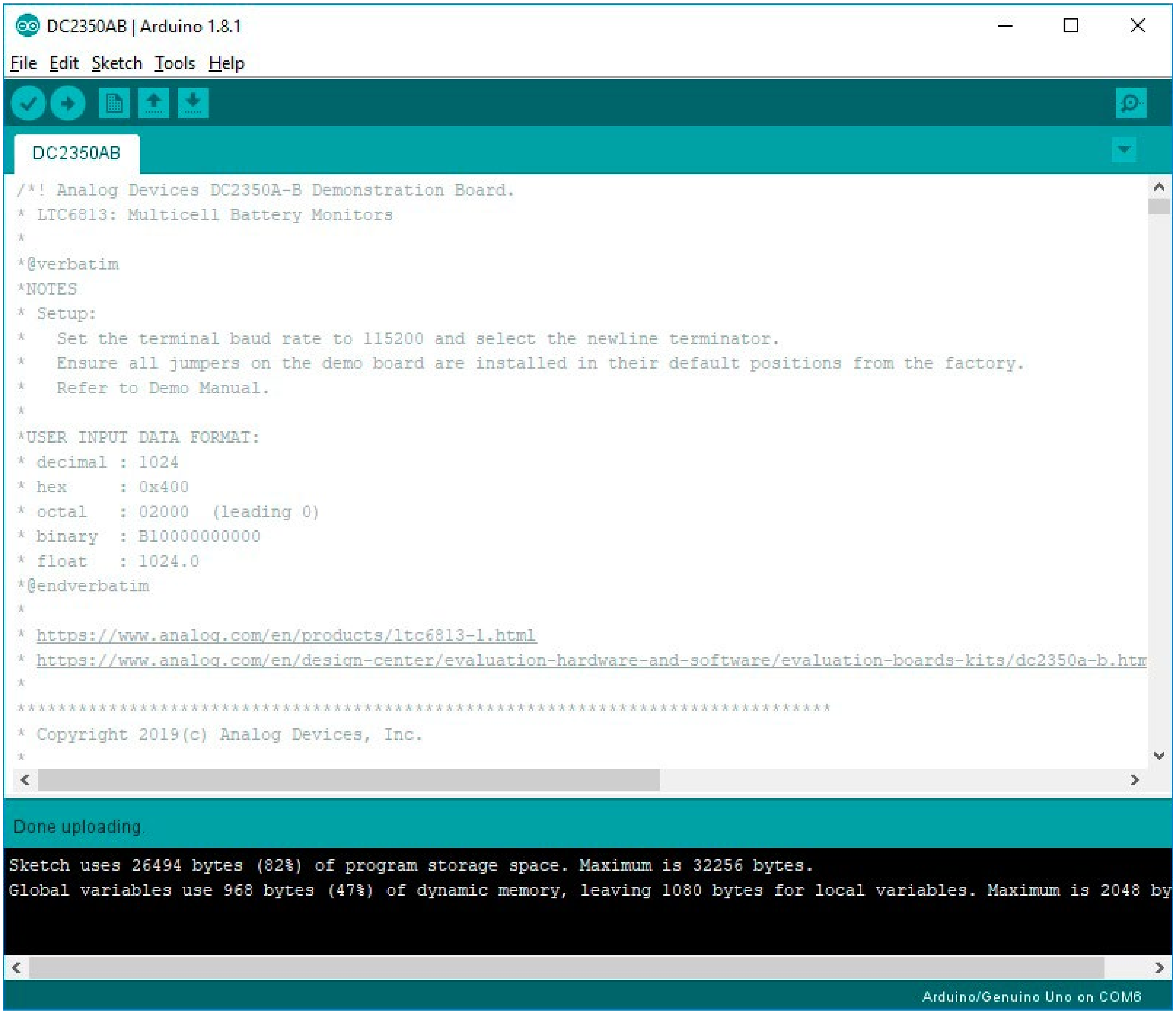

Figure 8. Uploading DC2350AB Sketch

SOFTWARE SETUP

8. Open the Arduino **Serial Monitor** (Figure 9) tool. Click on the Serial Monitor button on the top right corner then the Serial Monitor window will open and show on the top left corner the **COMxx** used.

Figure 9. Arduino Serial Monitor Tool

9. Configure the **Serial Monitor** to allow communication to the DC2026 through USB. On the bottom of the Serial Monitor window, set the following starting from bottom left to bottom right:
   a. Click on the **Autoscroll** checkbox for the √ check mark symbol.
   b. Select **Both NL & CR** on the left dropdown menu.
   c. Select **115200** baud on the right dropdown menu.
   d. As shown in Figure 10, when configured correctly the DC2350AB sketch menu will appear.

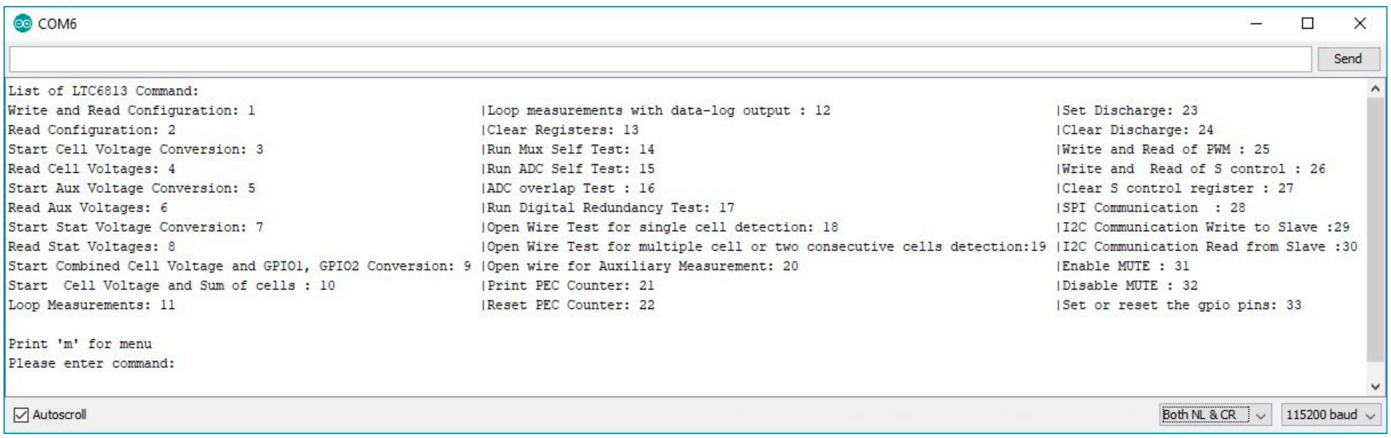

Figure 10. DC2350AB Sketch Menu

APPENDIX A: THE SKETCHBOOK CONTENTS

The LTSketchbook will generally contain the following folders: Libraries, Part Number, Documentation and Utilities.

Libraries Directory: Contains a subdirectory for each IC in the sketchbook. Each subdirectory contains a *.cpp* and *.h* file. These files contain all of the constant definitions and low-level IC command implementations. Porting to a different microcontroller requires changes to some library files.

Part Number Directory: Contains example control programs for each IC. Inside the 6000 folder of the Part Number folder, each 68xx folder is a BMS IC with a sketch(*.ino*) file that implements a control program to evaluate the functionality of the IC. This sketch allows the user to control the IC through a serial terminal and make all primary measurements. This sketch also allows for evaluation of self-test and discharge features of the IC. Generally, the name of a sketch relates to the IC's demo board. For example, the sketch for LTC6804 is *DC1942.ino*, for LTC6811 it is *DC2259.ino*, and for LTC6813 it is *DC2350AB.ino*.

Utilities Directory: Contains support programs, including a program that emulates a standard Analog Devices DC590 isolated USB to serial controller.

Documentation Directory: Contains *html* documentation for the provided code base. Documentation for all of the BMS ICs can be accessed by opening the *Linduino.html* file, as found in the main sketchbook directory (Figure 11) and in the Documentation directory.

APPENDIX A: THE SKETCHBOOK CONTENTS

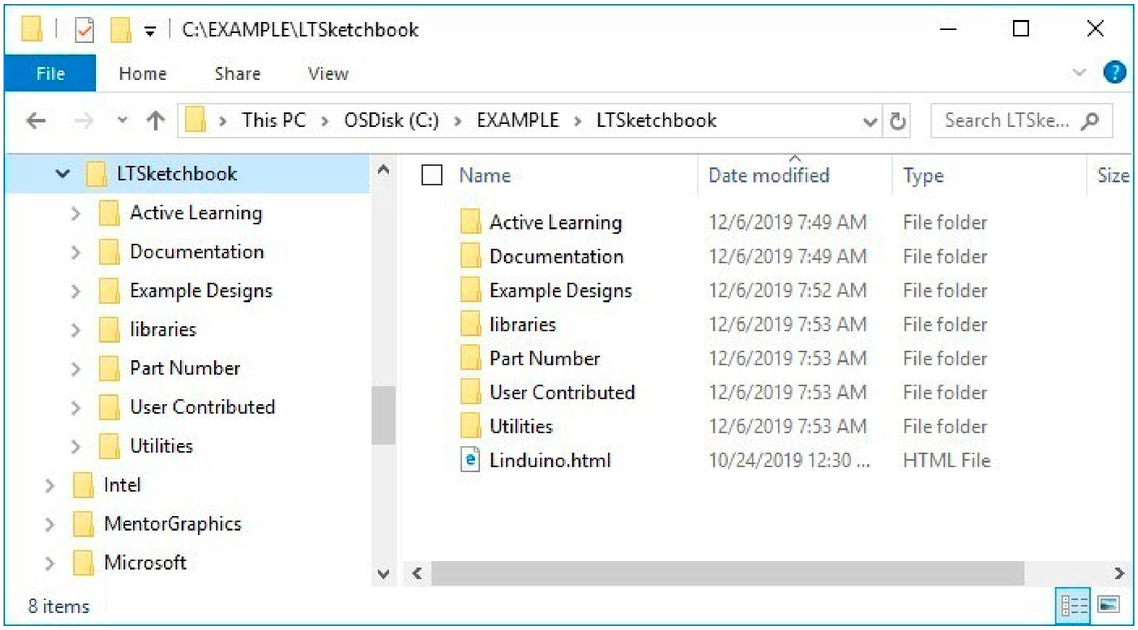

Figure 11. Documentation Directory

What Is a Sketch

A **sketch** is simply another word for a microcontroller/ Linduino program. The term is generally only used when referring to Arduino-based programs, as sketches have several abstractions that remove some of the complexity of a standard microcontroller(MCU) program. All sketches contain two primary functions, the *setup()* and the *loop()* function. These are in fact the only functions that are mandatory in a sketch and are almost always implemented in some form in a typical MCU program. The *setup()* function is run once at power on or after the MCU is reset. The *setup()* function generally is used to initialize the MCU peripheral circuits and to initialize all of the control variables. The *loop()* function is similar to a *main()* function that has implemented an infinite loop inside a standard C program. The code within the *loop()* function is typically where the primary program code is placed. The code within the *loop()* function will repeat infinitely.

Sketch Modifications

Sketches can be modified to a set of applications specific requirements. All sketches are written such that the most common modifications can be made by changing the variables listed in the /**Setup Variables* */table at the top of the sketch. For reference, example modifications to a DC2259 (LTC6811) sketch are shown below. These modifications are applicable to most of the available BMS ICs in the sketchbook.

Common modifications can be made by changing the *Setup Variables*. The most common application changes are listed below. After the variables are changed, the sketch will need to be recompiled and uploaded to the Linduino.

1. To change the number of ICs in the isoSPI network, change the TOTAL_IC variable. A number between 1 and 4 should be entered. In an application that has 2 devices in the network the modified line will look like the following.

   ```
   const uint8_t TOTAL_IC = 2;
   ```

2. Often an application may need to sample data at a rate faster than the default 500ms (2Hz). To modify the loop/sample rate the MEASUREMENT_LOOP_TIME variable should be changed. The loop time must be entered in milliseconds and should be a number larger than 20mS. To change the loop rate to roughly 10 measurements a second the loop rate should be changed to 100mS. The modified line will look like the following.

   ```
   const uint16_t MEASUREMENT_LOOP_TIME = 100;
   ```

APPENDIX A: THE SKETCHBOOK CONTENTS

3. It is possible to modify which measurements fall within the loop during the Loop Measurements command. The following list are the measurements that can be looped.

```
const uint8_t MEASURE_CELL = ENABLED;
// This is ENABLED or DISABLED
const uint8_t MEASURE_AUX = DISABLED;
// This is ENABLED or DISABLED
const uint8_t MEASURE_STAT = DISABLED;
//This is ENABLED or DISABLED
```

   By default, only a cell measurement is done, as noted by MEASURE_CELL = ENABLED. What measurements are made can be changed by setting what the Measure field is equal to. To Measure Cells and the Status register but not the AUX register, the variables would be setup as shown below:

```
const uint8_t MEASURE_CELL = ENABLED;
// This is ENABLED or DISABLED
const uint8_t MEASURE_AUX = DISABLED;
// This is ENABLED or DISABLED
const uint8_t MEASURE_STAT = ENABLED;
//This is ENABLED or DISABLED
```

4. ADC conversion settings can also be modified in the *Setup Variables* section. The default setup is to run the ADC in **Normal** mode, which has a 7kHz filter code; in this mode the ADC_OPT bit is disabled. Typical choice for which cell to convert is ALL. Full ADC conversion programming requires setting ADC_OPT, ADC_CONVERSION_MODE, CELL_CH_TO_CONVERT, AUX_CH_TO_CONVERT, and STAT_CH_TO_CONVERT. These variables are programmed with constants listed in the LTC68xy_daisy.h file. For simplicity they are also listed below.

```
MD_422HZ_1KHZ
MD_27KHZ_14KHZ
MD_7KHZ_3KHZ
MD_26HZ_2KHZ
ADC_OPT_ENABLED
ADC_OPT_DISABLED
CELL_CH_ALL
CELL_CH_1and7
CELL_CH_2and8
CELL_CH_3and9
CELL_CH_4and10
CELL_CH_5and11
CELL_CH_6and12
```

   To set the ADC to have a 1kHz filter corner the ADC_OPT and ADC_CONVERSION_MODE variables would be changed as follows.

```
ADC_OPT = ADC_OPT_ENABLED;
ADC_CONVERSION_MODE = MD_422HZ_1KHZ;
```

   To convert only cells 2 and 8,

```
CELL_CH_TO_CONVERT = CELL_CH_2and8;
```

5. In another example, the user may wish to change the undervoltage and overvoltage thresholds. Each number is based on an LSB of 100μV.

```
//Under Voltage and Over Voltage Thresholds
const uint16_t OV_THRESHOLD = 41000;
// Over voltage threshold ADC Code.
// LSB = 0.0001
const uint16_t UV_THRESHOLD = 30000;
// Under voltage threshold ADC Code.
// LSB = 0.0001
```

ESD Caution

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.