LTC6806 isoSPI Fuel Cell Monitor

DESCRIPTION

Demonstration circuit 2192B is a fuel cell monitor featuring the LTC®6806, a 36-channel fuel cell 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 DC2192B demo circuit also features reversible isoSPI enabling a redundant communication path. The PCB, components and DuraClik connectors are optimized for low EMI susceptibility and emissions.

The DC2192B can communicate to a PC by connecting a DC2792B dual master isoSPI together with DC2026 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 DC2026C provides a standard SPI interface which can be translated to isoSPI and then connected to a DC2192B isoSPI port (J3/J4 connector). The DC2792B provides two SPI-isoSPI channels for reversible operation.

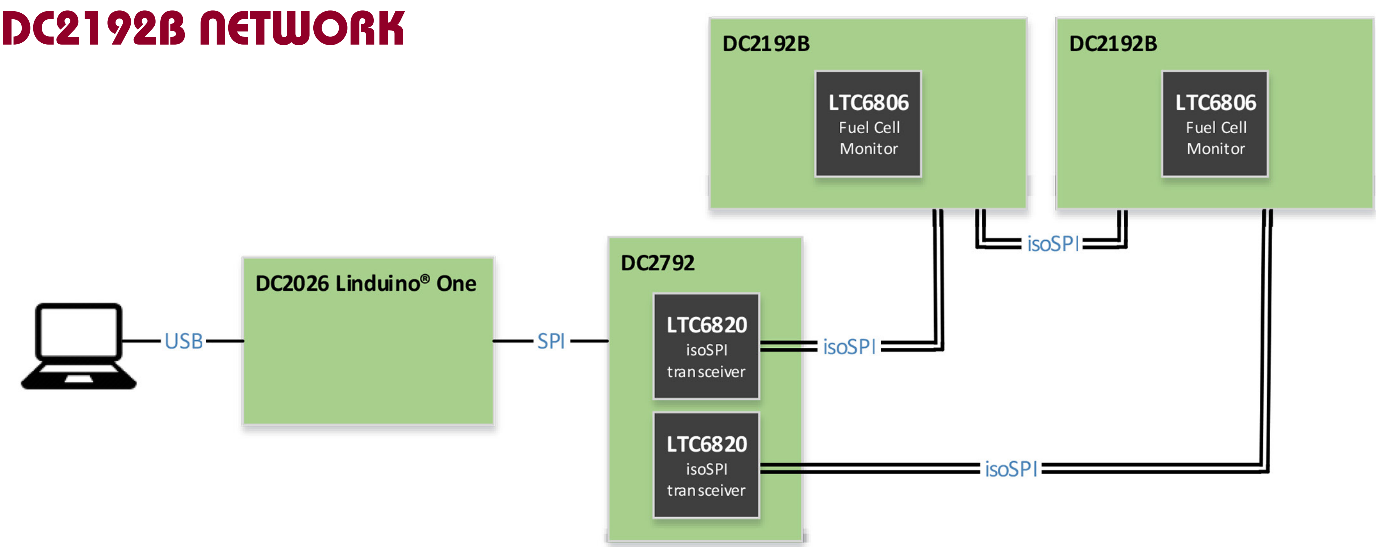
Design files for this circuit board are available.

PERFORMANCE SUMMARY

Specifications are at T_A = 25°C

PARAMETER	MIN	TYP	MAX	UNITS
V^+ Supply Voltage (Relative to V^-) where TME Specifications are Met	4.75	5.0	5.5	V
Input Voltage Range, C25 – C36 to V^-	–5.0		140	V
Input Voltage Range, C14 – C24 to V^-	–5.0		95	V
Input Voltage Range, C0 – C13 to V^-	–5.0		60	V
V_{REF1} 1st Reference Voltage, No Load	2.700	3.058	3.300	V
V_{REF2} 2nd Reference Voltage, 1mA Load to V^-	2.200	2.500	2.700	V
V_{BIAS} I_{BIAS} Voltage, READY/ACTIVE State	1.9	2.0	2.1	V
CPIN Input Measurement Range with HIRNG = 0	–2.5		2.5	V
CPIN Input Measurement Range with HIRNG = 1	–5.0		5.0	V
Cell Count			36	

DC2192B NETWORK



HARDWARE SETUP

Table 1. J1 Pinout

J1 PIN	CPIN INPUT
19	C0/GND
37	C1
18	C2
36	C3
17	C4
35	C5
16	C6
34	C7
15	C8
33	C9
14	C10
32	C11
13	C12
31	C13
12	C14
30	C15
11	C16
29	C17
10	C18
28	C19
9	C20
27	C21
8	C22
26	C23
7	C24
25	C25
6	C26
24	C27
5	C28
23	C29
4	C30
22	C31
3	C32
21	C33
2	C34
20	C35
1	C36

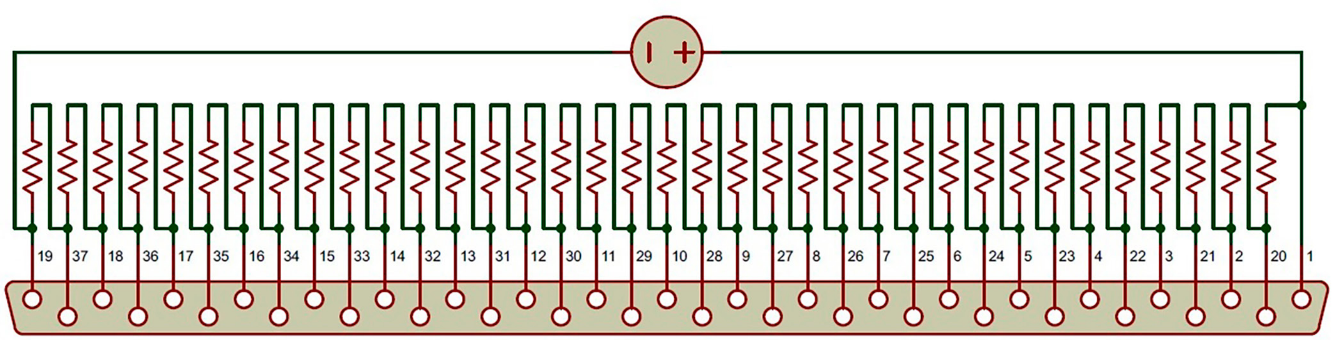


Figure 3. Resistor Divider String Connection D-SUB37 (DB-37) Female Receptacles Schematic of Figure 2. Includes the Power Supply Connection

DC2192B SERIAL INTERFACE OPTIONS

isoSPI is the only communication option to DC2192B. 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 DC2792B demo manual for usage details.

DC2792B to DC2192B Typical isoSPI Connection

A typical isoSPI connection begins with the isoSPI master connected to the first (or **bottom**) DC2192B. Additional DC2192B boards can be daisy-chained onto the isoSPI bus. Communication begins from the first (or **bottom**) DC2192B then to the next **upper** DC2192B and, finally, to the last (or **top**) DC2192B.

Figure 4 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 DC2026 to 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 DC2792B to DC2192B in isoSPI mode. This DC2192B 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** DC2192B J3 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.
   b. Connect an external 3.3V – 15V power supply to the **bottom** DC2192B VEXT$^+$ E2 and VEXT$^-$ E1 turrets.

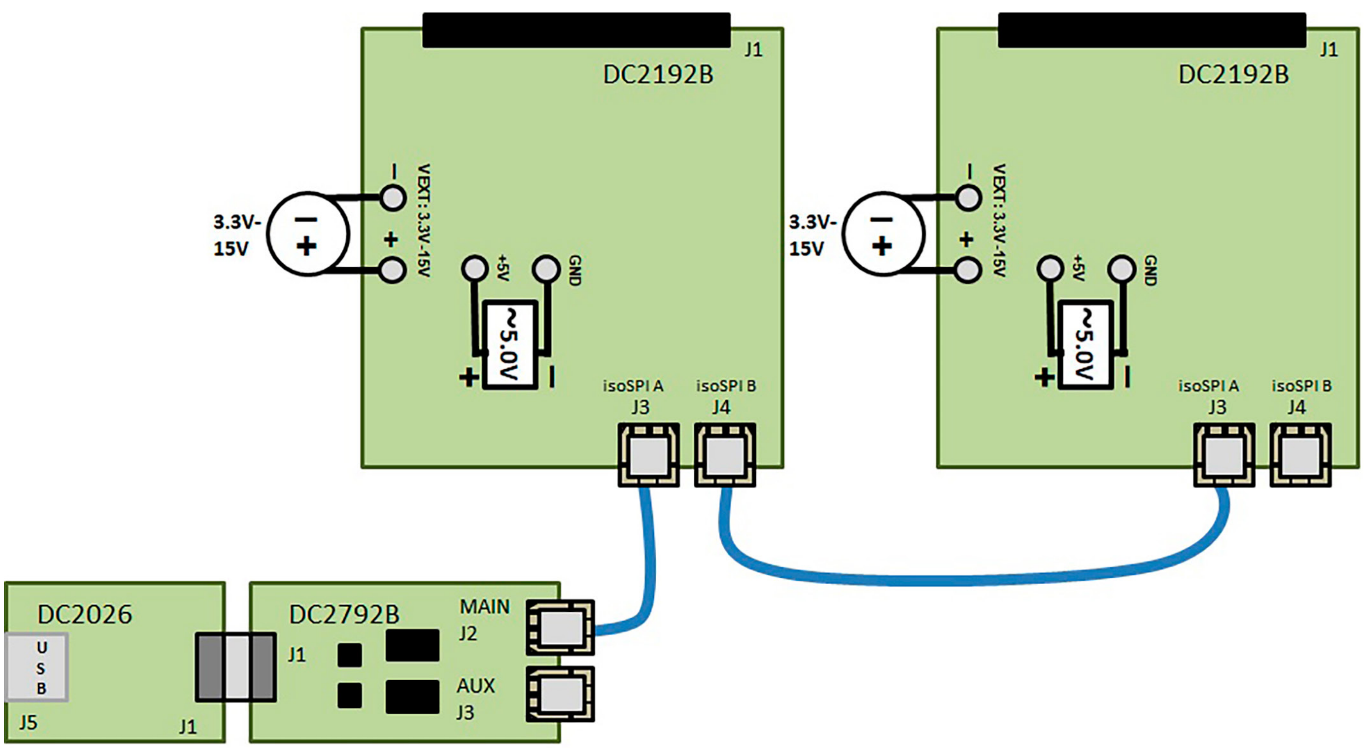


Figure 4. DC2792B Typical isoSPI Connection to the *Bottom* DC2192B in a Two-Board DC2192B Stack

DC2192B SERIAL INTERFACE OPTIONS

4. Connect or daisy-chain the DC2192B to another DC2192B in isoSPI mode. This DC2192B is the last (or **top**) board of a two-board stack. More DC2192B **upper** boards can be daisy-chained together in the same manner.

   a. Connect a 2-wire twisted-pair patch cable from the **bottom** DC2192B J4 isoSPI B DuraClik connector to the next **upper** or **top** DC2192B J3 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.

   b. Connect an external 3.3V – 15V power supply to the next **upper** or **top** DC2192B VEXT$^+$ E2 and VEXT$^-$ E1 turrets.

5. **CAUTION!** The fuel cell voltage must be connected correctly to prevent damage to the DC2192B. Refer to Table 1 and Figure 2 then confirm that the fuel cell voltage connections to the DB-37 female matches the DC2192B J1 pinout.

   a. Plug the DB-37 female receptacle connector onto the J1 fuel cell voltage DB-37 male plug connector.

6. Refer to Software Setup section of this demo manual to properly setup the PC with the Arduino IDE software to allow communication to the DC2192B boards.

DC2792B to DC2192B Reverse isoSPI Connection

A reverse isoSPI connection begins with the isoSPI Master connected to the last (or **top**) DC2192B. Additional DC2192B boards can be daisy-chained onto the isoSPI bus. Communication begins from the last (or **top**) DC2192B then to the next **lower** DC2192B and, finally, to the first (or **bottom**) DC2192B.

Figure 5 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.

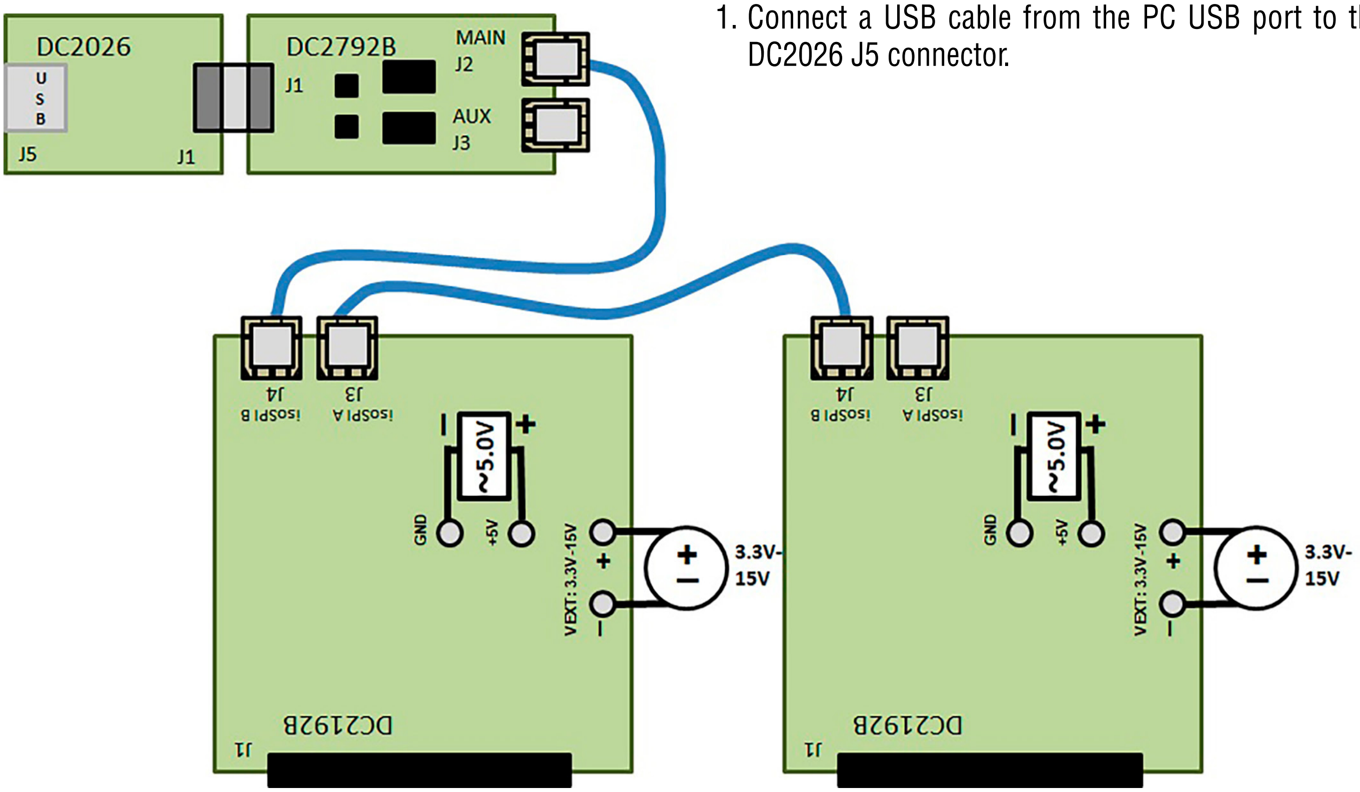


Figure 5. DC2792B Reverse isoSPI Connection to the *Top* DC2192B in a Two-Board DC2192B Stack

DC2192B SERIAL INTERFACE OPTIONS

2. Connect DC2026 to 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 DC2792B to DC2192B in isoSPI mode. This DC2192B 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** DC2192B J4 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.
   b. Connect an external 3.3V – 15V power supply to the **top** DC2192B $VEXT^+$ E2 and $VEXT^–$ E1 turrets.
4. Connect or daisy-chain the DC2192B to another DC2192B in isoSPI mode. This DC2192B is the first (or **bottom**) board of a two-board stack. More DC2192B **lower** boards can be daisy-chained together in the same manner.
   a. Connect a 2-wire twisted-pair patch cable from the **top** DC2192B J3 isoSPI A DuraClik connector to the next **lower** or **bottom** DC2192B J4 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.
   b. Connect an external 3.3V – 15V power supply to the next **lower** or **bottom** DC2192B $VEXT^+$ E2 and $VEXT^–$ E1 turrets.
5. **CAUTION!** The fuel cell voltage must be connected correctly to prevent damage to the DC2192B. Refer to Table 1 and Figure 2 then confirm that the fuel cell voltage connections to the DB-37 female matches the DC2192B J1 pinout.
   a. Plug the DB-37 female receptacle connector onto the J1 fuel cell voltage DB-37 male plug connector.
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 DC2192B boards.

DC2792B TO DC2192B Redundant isoSPI Connection

A redundant isoSPI connection begins with the primary (or **main**) isoSPI Master connected to the first (or **bottom**) DC2192B and has a backup auxiliary (or **aux**) isoSPI Master connected to the last (or **top**) DC2192B. Additional DC2192B boards can be daisy-chained between the two isoSPI masters on the isoSPI bus. Primary (or **main**) communication begins from the first (or **bottom**) DC2192B then to the next **upper** DC2192B and, finally, to the last (or **top**) DC2192B. The backup auxiliary (or **aux**) communication begins in the reverse direction to provide coverage when a possible isoSPI daisy-chain break occurs.

Figure 6 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 DC2026 to 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 DC2792B primary (or **main**) isoSPI master to the first (or **bottom**) DC2192B board of the stack.
   a. Connect a 2-wire twisted-pair patch cable from the DC2792B J2 MAIN DuraClik connector to the **bottom** DC2192B J3 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.
   b. Connect an external 3.3V – 15V power supply to the **bottom** DC2192B $VEXT^+$ E2 and $VEXT^–$ E1 turrets.

DC2192B SERIAL INTERFACE OPTIONS

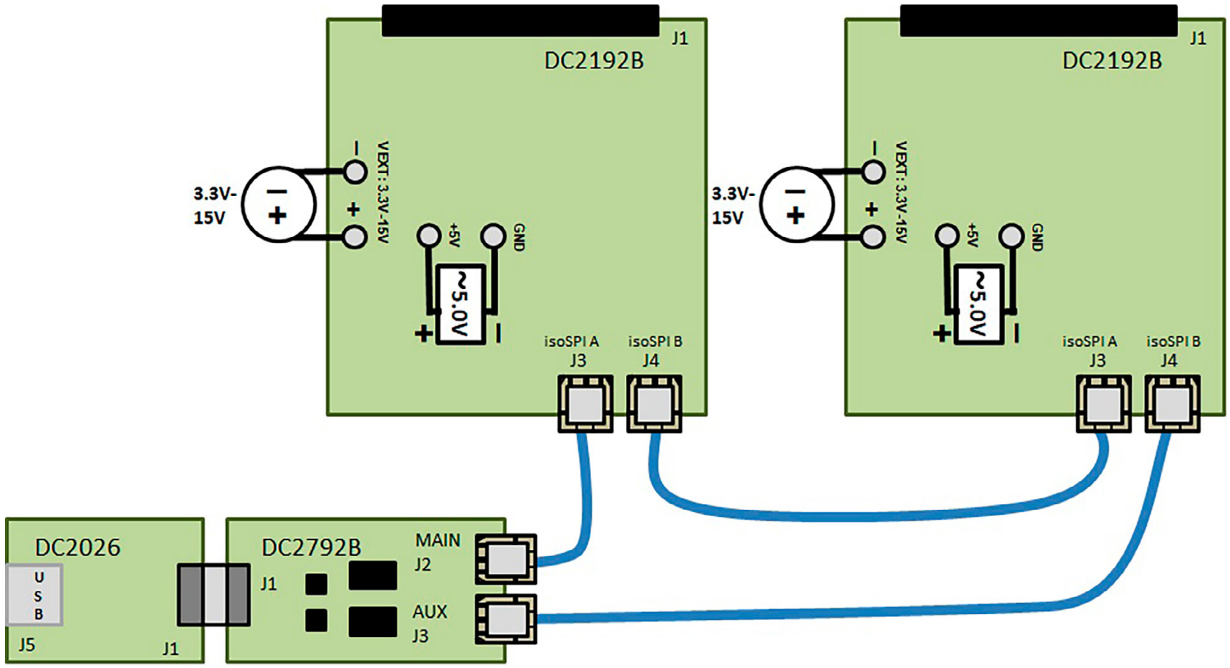


Figure 6. DC2792B Redundant isoSPI Connections to the *Bottom* and *Top* DC2192B in a Two-Board DC2192B Stack

4. Connect or daisy-chain the DC2192B to another DC2192B in isoSPI mode. This DC2192B is the last (or **top**) board of a two-board stack. More DC2192B **upper** boards can be daisy-chained together in the same manner.

   a. Connect a 2-wire twisted-pair patch cable from the **bottom** DC2192B J4 isoSPI B DuraClik connector to the next **upper** or **top** DC2192B J3 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.

   c. Connect an external 3.3V – 15V power supply to the **upper** or **top** DC2192B VEXT+ E2 and VEXT– E1 turrets.

5. Connect the DC2792B auxiliary (or **aux**) isoSPI Master to the last (or **top**) DC2192B board of the stack.

   a. Connect a 2-wire twisted-pair patch cable from the DC2792B J3 AUX DuraClik connector to the **top** DC2192B J4 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.

6. **CAUTION!** The fuel cell voltage must be connected correctly to not damage the DC2192B. Refer to Table 1 and Figure 2, then confirm that the fuel cell voltage connections to the DB-37 female matches the DC2192B J1 pinout.

   a. Plug the DB-37 female receptacle connector onto the J1 fuel cell voltage DB-37 male plug connector.

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 DC2192B boards.

SOFTWARE SETUP

The DC2192B can be controlled by 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 7).
3. Close then re-open the Arduino IDE to enable the use of the Sketchbook Location that was previously set.

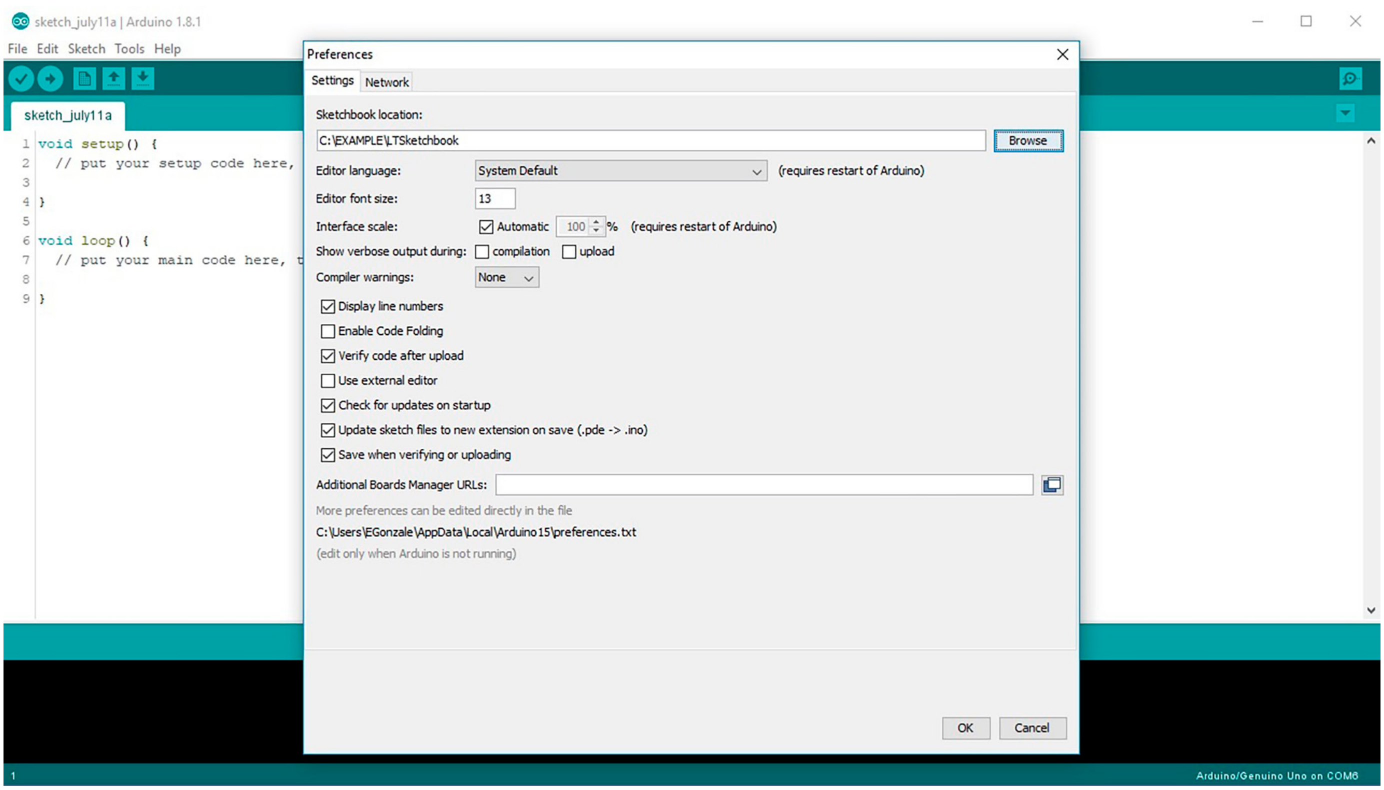


Figure 7. **Sketchbook Location Path**

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**. There may be more than one option—DC2026 is usually the highest COM port number. The PC screenshot (Figure 8) used in this example shows the DC2026 connected to COM6.

5. Select the correct Arduino compatible microcontroller board. Under the **Tools** menu, select **Board** → **Arduino/Genuino Uno** with the "•" black dot symbol (see Figure 9).

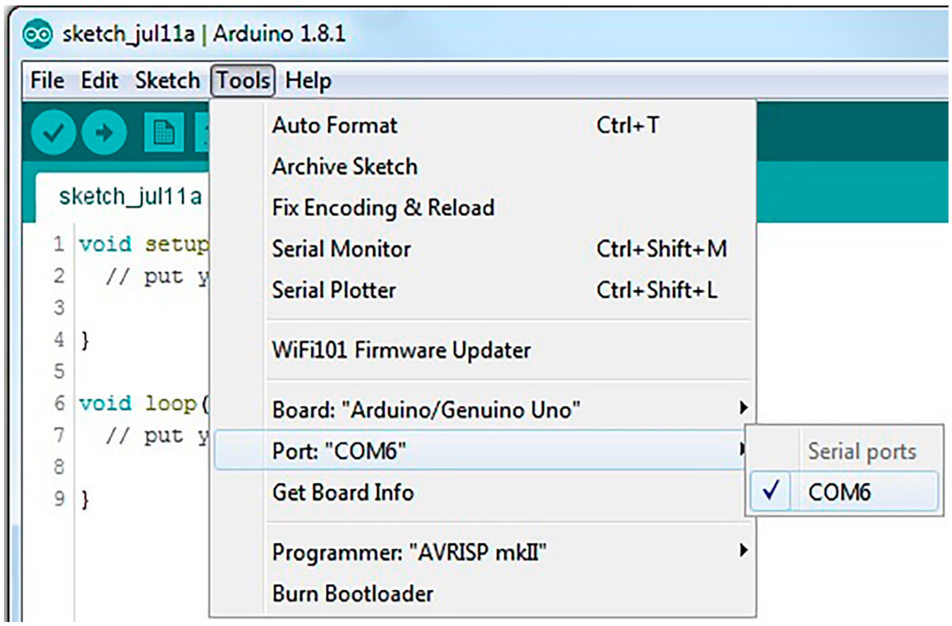


Figure 8. Selecting the Correct COM

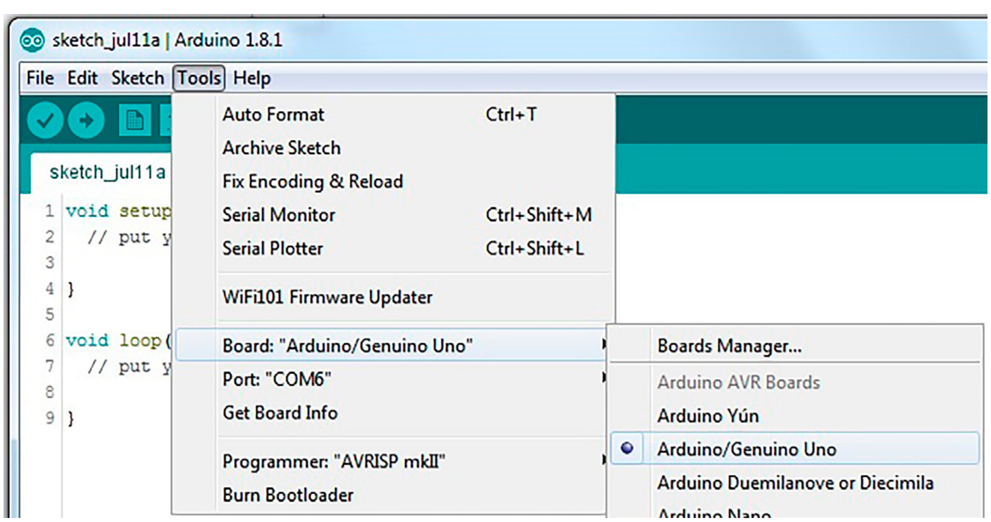


Figure 9. Selecting the Compatible Microcontroller Board

SOFTWARE SETUP

6. Open one of the programs or **sketches** associated with the DC2192. In this example LTC6806 sketch will be opened. Under the **File** menu, select **Sketchbook** → **Part Number** → **6000** → **6806** → **DC2192** (see Figure 10).

7. Upload the **DC2192** 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 11).

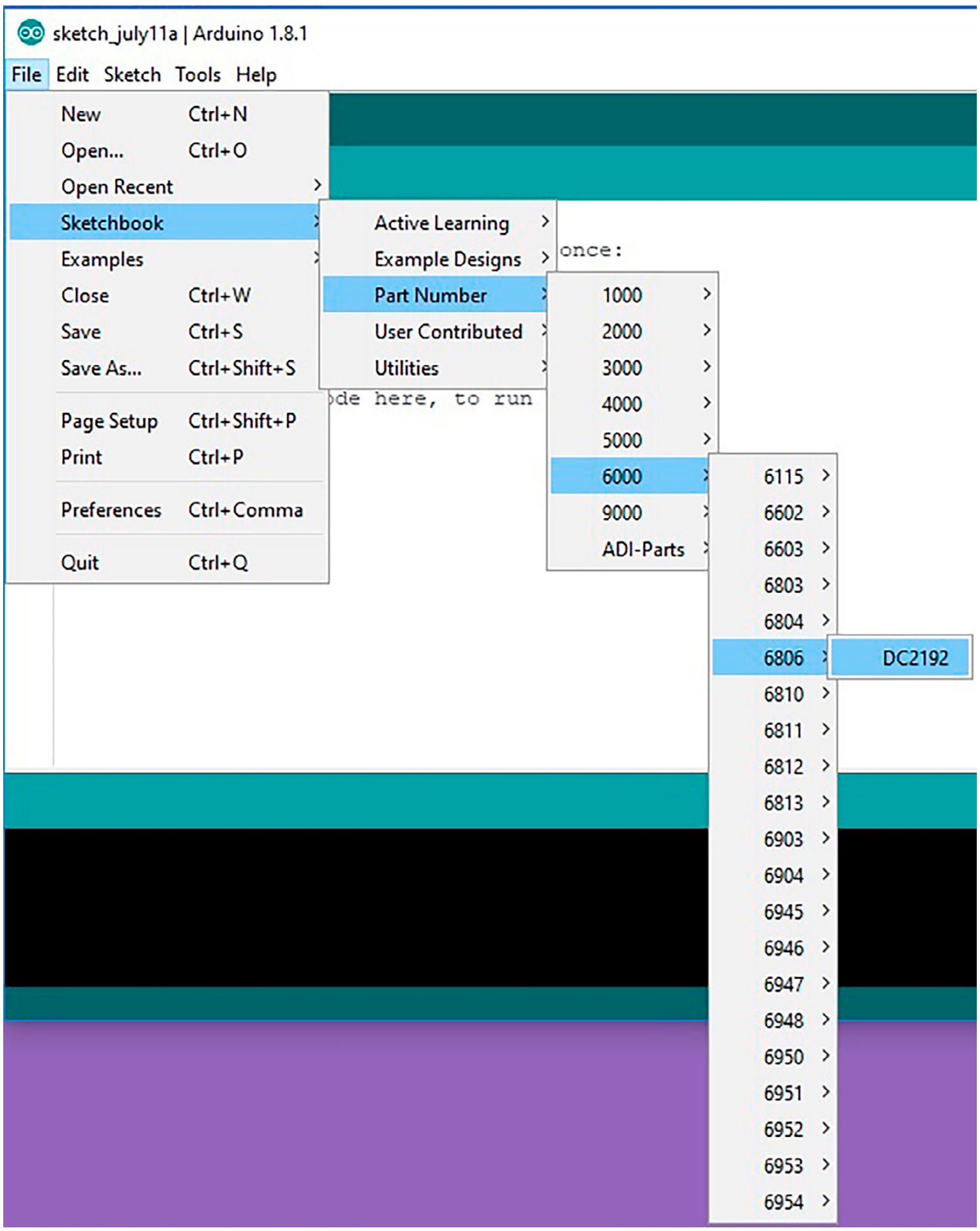


Figure 10. Selecting the LTC6806 Sketch

SOFTWARE SETUP

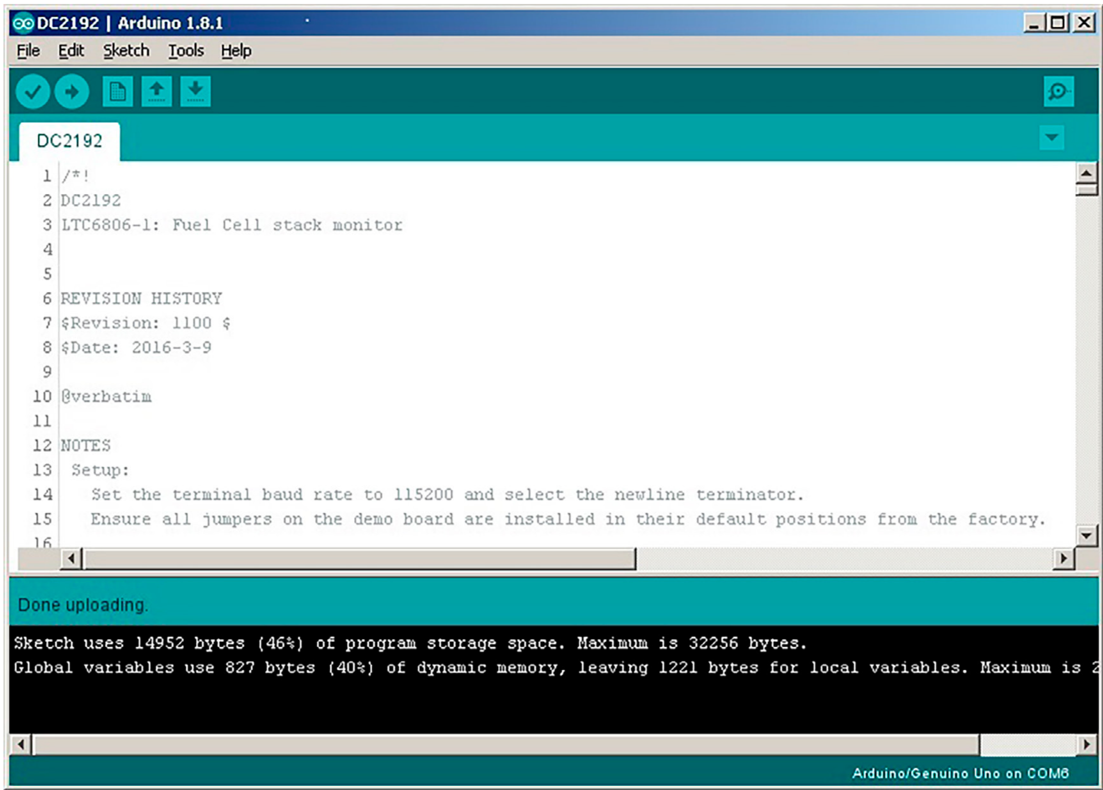


Figure 11. Uploading DC2192 Sketch

8. Open the Arduino **Serial Monitor** tool (Figure 12). 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.

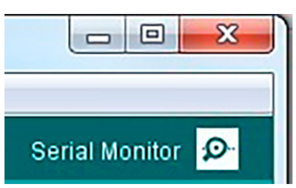


Figure 12. Arduino Serial Monitor Tool

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 to select it.
   b. Select Both **NL & CR** on the left dropdown menu.
   c. Select **115200** baud on the right dropdown menu.
   d. As shown in Figure 13, when configured correctly the DC2192 sketch menu will appear.

SOFTWARE SETUP

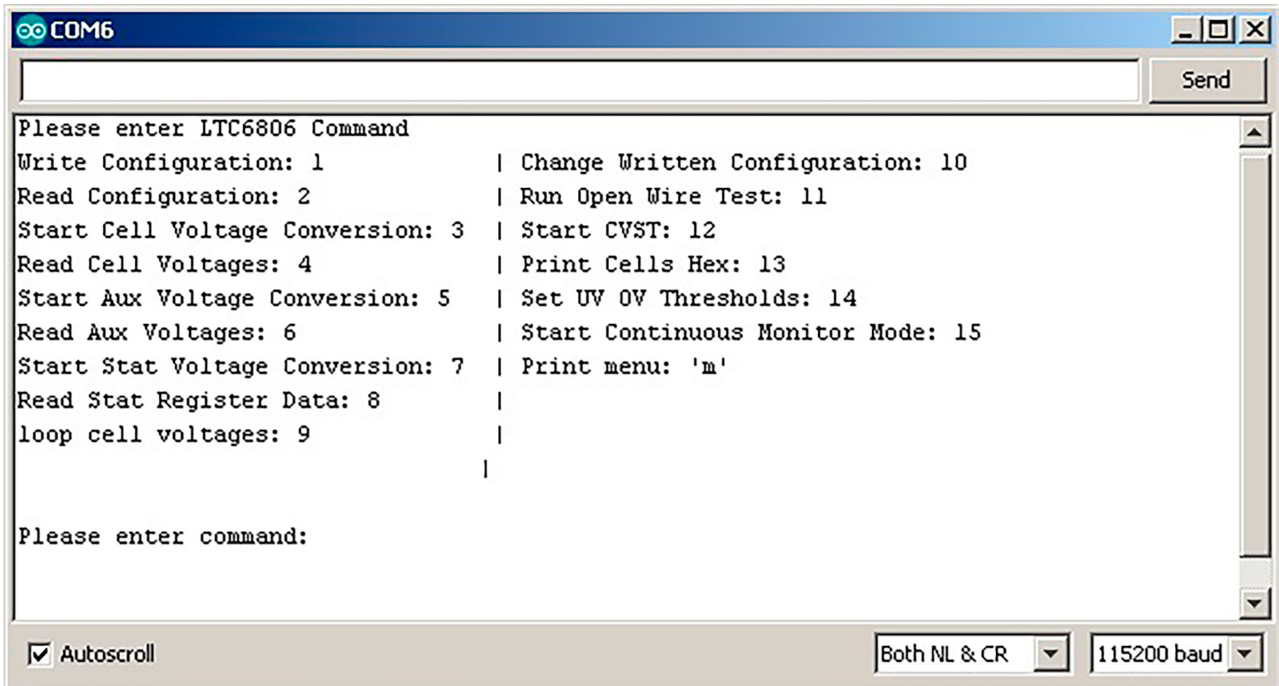


Figure 13. DC2192 Sketch Menu

APPENDIX A: THE SKETCHBOOK CONTENTS

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

Libraries Directory contains a subdirectory for each IC in the sketchbook. Each subdirectory contains a *.cpp* and *.h* files. These files contain all 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 Part Number folder, each BMS IC has 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 LTC6806 it is *DC2192.ino*.

Utilities Directory contains support programs, including a program that emulates the 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 (see Figure 14) and in the 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 contains 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.

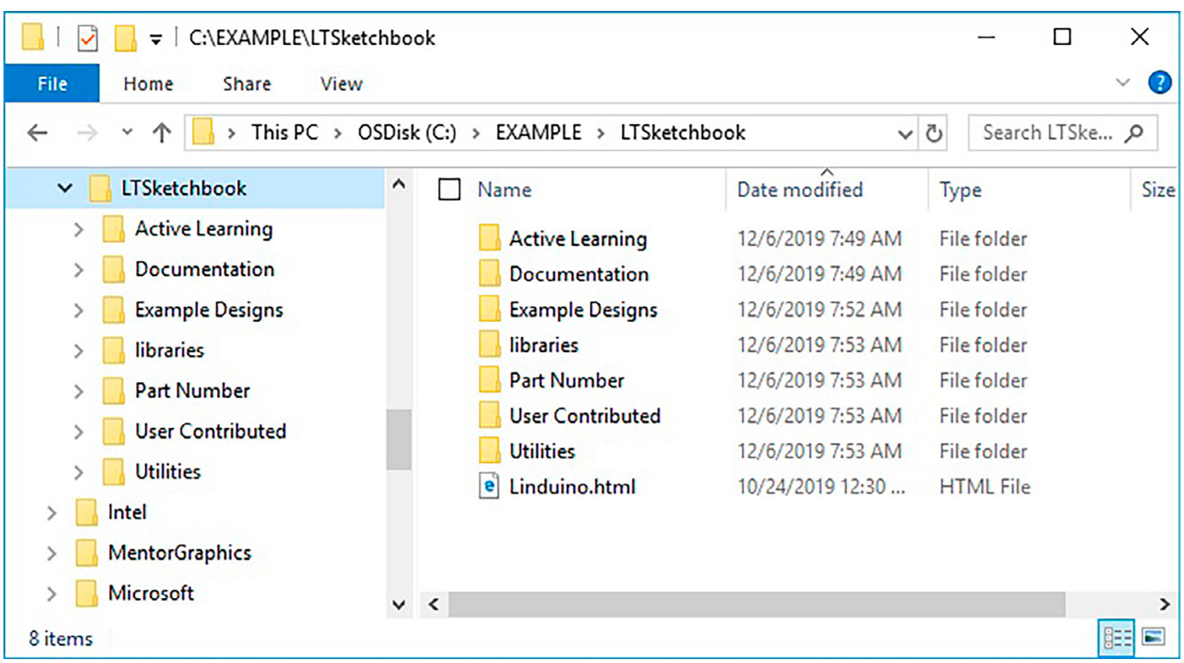


Figure 14. Documentation Directory

APPENDIX A: THE SKETCHBOOK CONTENTS

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:

```
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:

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

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
```

APPENDIX A: THE SKETCHBOOK CONTENTS

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

```
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.