

**Description**

This document is intended for use by individuals engaged in the development of hardware for a 4- to 7-series connected Li-ion battery pack using the ISL9208.

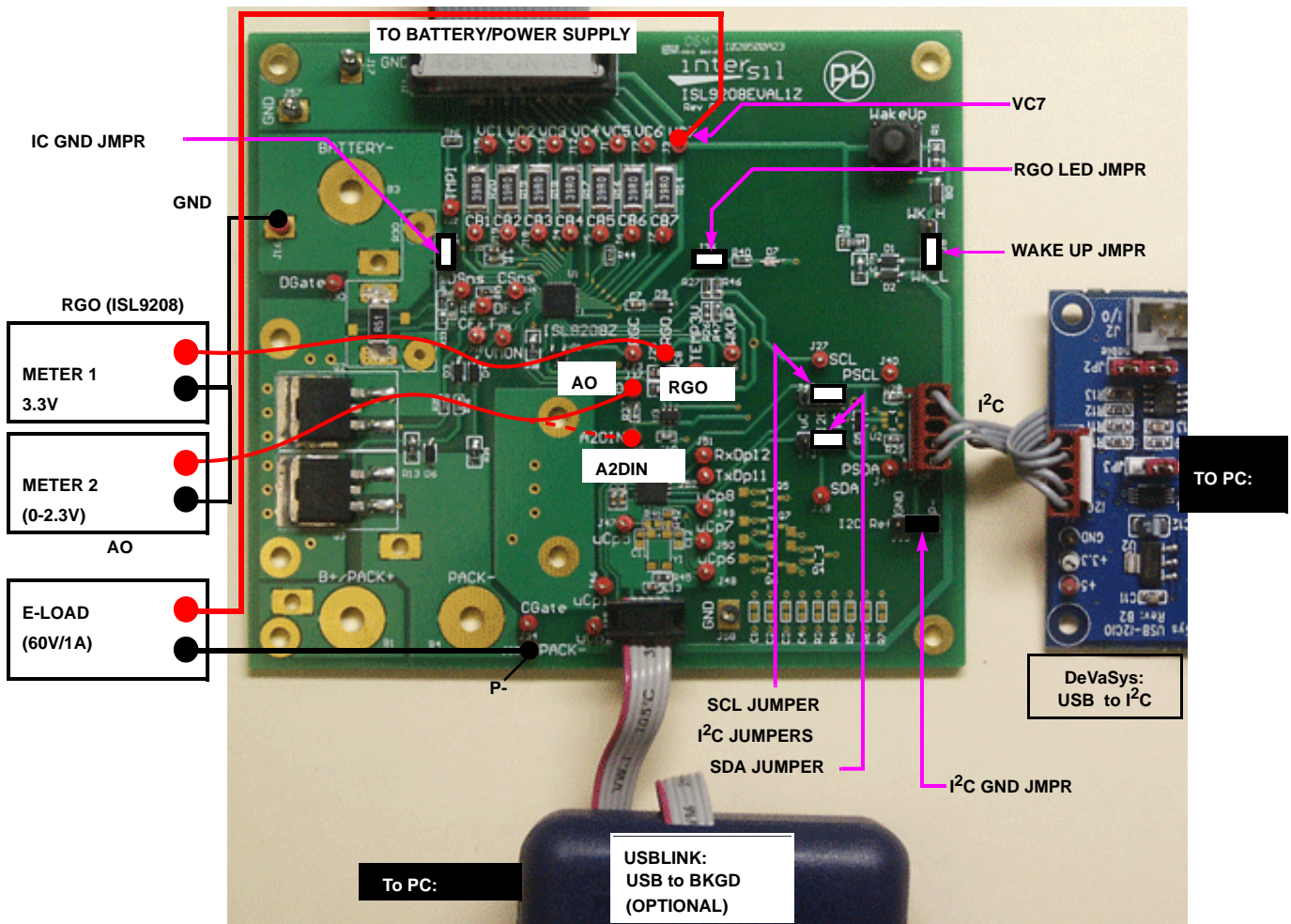
The evaluation kit consists of a main board and a USB to I<sup>2</sup>C board. An optional link between the PC and the microcontroller BKGD connector is available from Freescale for monitoring and debugging the microcontroller code.

Prior to powering the ISL9208 board, it is advised that the USB to I<sup>2</sup>C board software be installed and the DeVasys board be connected to the PC (see "Appendix 1" on page 13). Don't connect the DeVasys board to the ISL9208EVAL1 (Rev C) board, yet. With these pieces in place, the PC interface can quickly be used to monitor the operation of the board once power is applied.

**Initial Testing**

**Setup**

- Make sure the following jumpers are on (See Figure 1): IC GND JMPR, RGO LED JMPR, WKUP JMPR, I<sup>2</sup>C GND JMPR, and the I<sup>2</sup>C Jumpers.
- For initial testing, set the I<sup>2</sup>C jumpers (SCL and SDA) to the PC position. This configures the board such that the PC communicates directly with the ISL9208.
- Before connecting the PC to the ISL9208EVAL1 (Rev C) board (through the USB to I<sup>2</sup>C interface) connect the power supply to the ISL9208EVAL1 (Rev C) board.
- The power supply should consist of a string of 4 to 7 batteries, or a string of 4 to 7 resistors and a power supply, or 4 to 7 individual power supplies (see Figure 2 or Figure 4).
- Once power is turned on (or Li-ion cells are connected to the ISL9208 board,) the RGO LED should light. Use meter 1 to measure the RGO voltage. It should read about 3.3V.



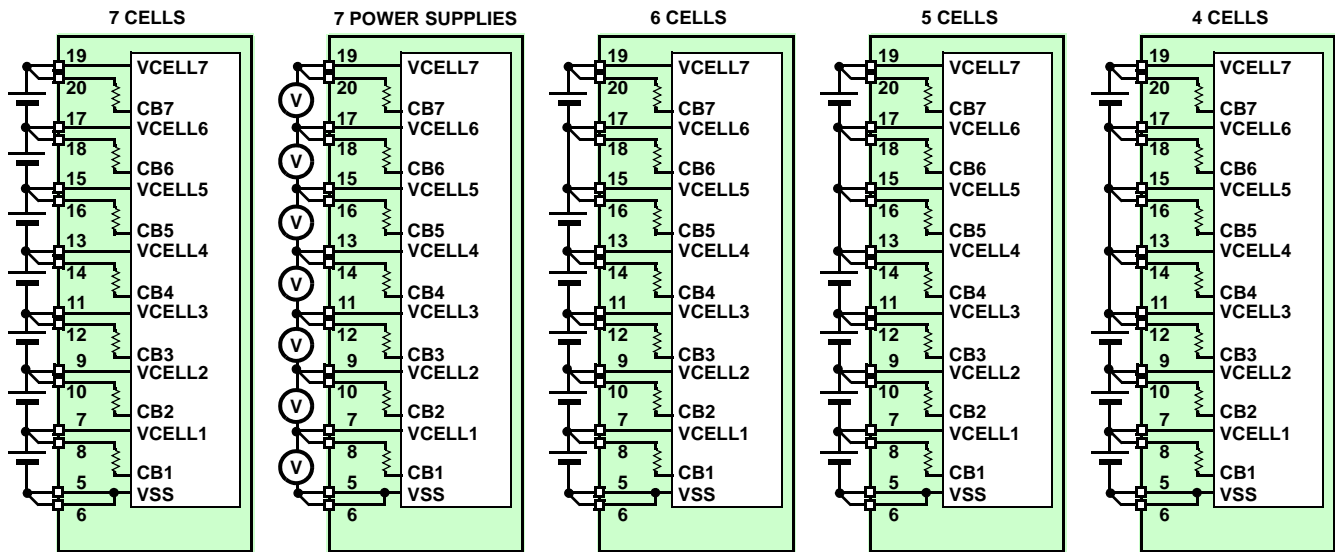
**FIGURE 1. BOARD CONNECTION DIAGRAM**

**Battery/Power Supply Connection**

When connecting battery packs or power supplies, use the connections of Figure 2 and Figure 3. If individual power supplies are being used to replace battery cells, then connect the power supplies identically to the battery connections (see Figure 2). Also, make sure that the individual power supply voltages do not exceed the ISL9208 maximum input voltage differential of 5V per cell.

If using a string of resistors to emulate the battery cells, then use the connection in Figure 4 and Figure 5. In this case, limit the power supply voltage so that the resistor divider outputs do not exceed the ISL9208 input maximum ratings.

It is recommended that the series resistors be  $20\Omega$  or less and 2W minimum. Resistors with higher resistance can be used, but when activating the ISL9208 cell balance outputs, the  $39\Omega$  cell balance resistor will lower the voltage across that series power supply resistor, while raising the voltage on all of the other series resistors. Turning on multiple cell balance outputs could then result in one or more of the VCELLN input voltages exceeding their maximum specified limit. If using series resistors greater than  $20\Omega$ , the cell balance resistors ( $R_{14}$  through  $R_{20}$ ) should be replaced with higher values, with  $1k\Omega$  resistors recommended.



NOTE: Multiple cells can be connected in parallel

FIGURE 2. BATTERY CONNECTION OPTIONS

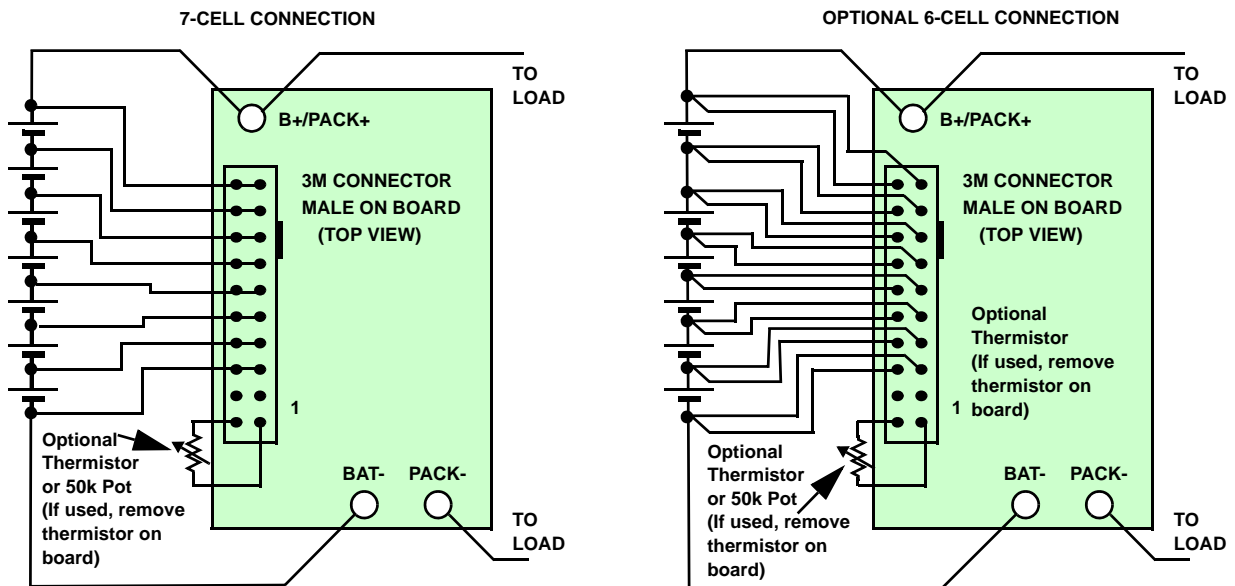
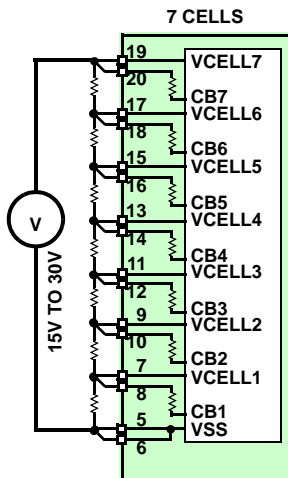


FIGURE 3. BATTERY CELL CONNECTION TO ISL9208 PCB



NOTES:

1. For the battery simulation resistors, use 20Ω/5W units. If the resistors are more than 100Ω, then turning on the cell balance resistors cause fluctuations in the cell input voltages that can violate the ISL9208 max specifications.
2. Switch the power supplies on at the same time, or if this cannot be guaranteed, turn them on from bottom to top.

FIGURE 4. USING RESISTOR/POWER SUPPLY COMBINATION TO EMULATE A STRING OF BATTERIES

RECOMMENDATION:  
ALL RESISTORS = 20Ω/2W

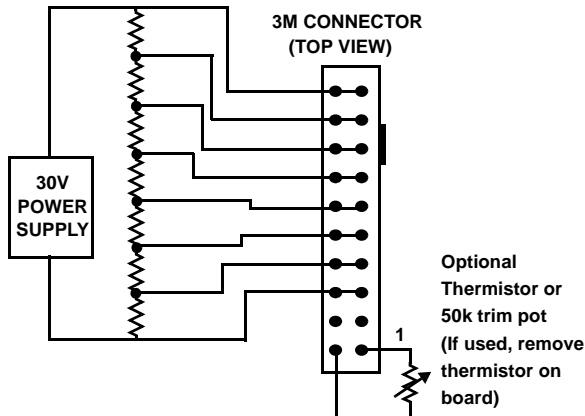


FIGURE 5. POWER SUPPLY/RESISTOR CONNECTION TO ISL9208 PCB

USB to I<sup>2</sup>C Interface

- Once the power supply connections are verified, power down the ISL9208EVAL1 (Rev C) boards and make the PC connection to the board, via the USB port, the DeVaSys board, and the I<sup>2</sup>C cable. Before making this connection, make sure that the USB to I<sup>2</sup>C interface

software is installed (See “Installing the DeVaSys USB to I<sup>2</sup>C Board Software” on page 13).

- Connect the I<sup>2</sup>C cable from the interface board to the ISL9208EVAL1 (Rev C) as in Figure 6. The kit may come with a 5-pin to 5-pin cable or a 5-pin to 4-pin cable.

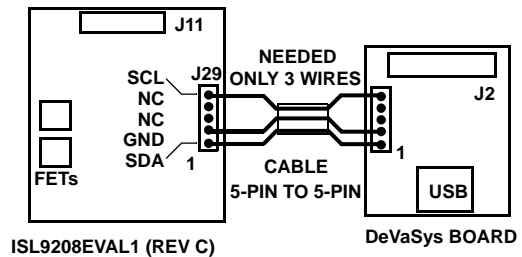


FIGURE 6. I<sup>2</sup>C CONNECTION TO ISL9208 PCB

Testing without the Microcontroller

Cell Voltage Monitor Accuracy Check

- For this test, make sure the SCL and SDA jumpers are set to the PC position. In this case, the PC has full control of the board and the microcontroller function is disabled (see Figure 7). Except for the ISL9208 automatic response to overcurrent and over temperature, all other actions of the board are manual and controlled through the GUI.

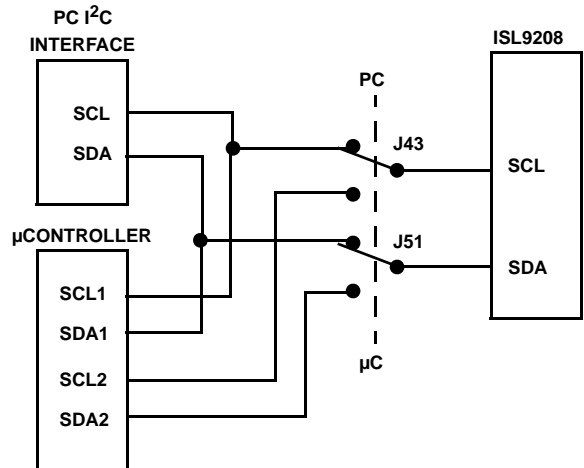


FIGURE 7. I<sup>2</sup>C JUMPERS: PC or μC CONNECTION TO THE ISL9208

- Make the I<sup>2</sup>C port connection to the PC.
- Power up the board and re-check the RGO voltages. Since RGO is the voltage reference for the on-board A/D converter, this voltage may be needed in the accuracy calculations.
- Start the GUI. Execute the program BATTERYPACK.EXE from the “Software” directory.
- The GUI should power up with some color. That is, the FET controls should be RED and the indicators should be green or red. If the GUI is all gray, then there is a

communication problem. If there is a communication problem, see the troubleshooting guide in the Appendix.

- Use the GUI to read register 0 from the ISL9208. The ISL9208 should return the value 20H. This verifies communication to the device.
- Next, move to the “MONITOR” tab of the GUI.
- Set the ISL9208 to monitor the VCELL1 input by choosing VCELL1 in the Monitor drop down box. Execute this command by clicking “refresh.” This operation connects the VCELL1 input to the AO output (through a level shifter and divider). Any changes on VCELL1 appear on AO.
- Using a meter, measure the CELL1 voltage (from test point VC1 to GND) and measure the voltage on the ISL9208 analog output (from AO to GND). The AO voltage, times 2, should equal the CELL1 voltage. Any errors in this measurement are due to the ISL9208. (Note: make sure that all of the cell balance outputs are off, because cell balance current will cause inaccurate measurements).
- Also, read the GUI value for CELL1. In this configuration (without the  $\mu C$ ) the cell voltage is converted to digital using a 15-bit A/D converter. This A/D converter is an ADS1100 from Texas Instruments. Its output is determined by Equation 1:

$$\frac{\text{DigValue}_D}{32768} \times 3.3 = \text{A2DIN} \quad (\text{EQ. 1})$$

Since, the reference for the A/D converter is supplied by the ISL9208 RGO voltage, any difference between the RGO voltage and 3.3V turns up as an accuracy error.

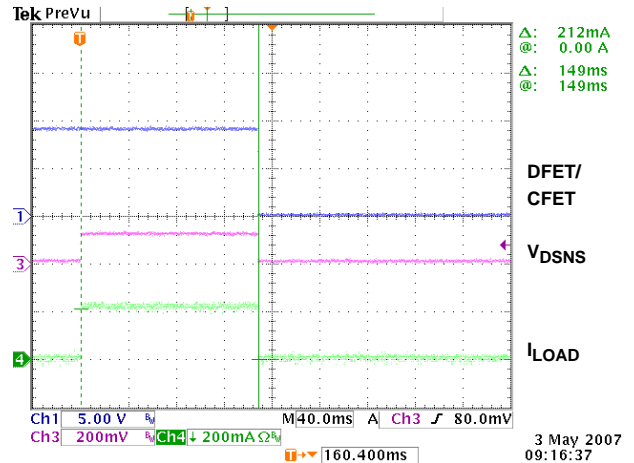
- Proceed, in sequence, to read the AO voltage for each cell connected to the ISL9208.

### Discharge Overcurrent Testing

- Select an E-load that is able to handle up to 30V and sink 1A minimum.
- With the e-load output off, connect the positive terminal to Test Point VC7 (Battery + terminal) and the negative terminal to test point P- (Battery - terminal).
- Use the GUI “CONFIGURATION” screen to set the desired discharge overcurrent and short circuit levels and time delays in the ISL9208.
- To test overcurrent, a pulse load or a continuous load can be used. A continuous load has the advantage of showing the load monitor operation (see below).
- Set the e-load current such that it will exceed the expected overcurrent threshold for more than the selected time delay interval.
- Turn on both FETs by clicking on the FET buttons in the GUI. When the FETs are on, the GUI indicators will turn GREEN. Periodically click on the “Status Refresh” button on the lower right of the screen to make sure that the GUI reflects the latest status of the device. (An automatic scan can also be started that updates all parameters every 1, 5,

10, or 30s, however, this might cause an update when not expected).

- Turn on the e-load output. This should cause the FETs to turn off (see Figure 8).



**FIGURE 8. DISCHARGE OVERCURRENT TEST (0.1V THRESHOLD, 160ms TIME DELAY, 0.5 $\Omega$  SENSE RESISTOR)**

- Do a refresh of the GUI and the FET buttons should have gone to RED. Also, the “Discharge Overcurrent” indicator should now be red.
- Leave the load on and click on the “Enable Load Monitor” button in the lower right corner of the screen. This turns on the load monitor output.
- Click on the “Status Refresh” button and the “Load Fail” indicator should now also be red.
- Turn off or remove the load and again click on “Status Refresh”. The “Load Fail” indicator should go to green. Click on the “Reset Overcurrent” button to reset the “Discharge Overcurrent” indicator. It should also go to green. If the indicators are still red, it is because the remaining resistance on the load keeps the voltage on the ISL9208 load monitor (VMON) pin above its input threshold. Try disconnecting the load.
- Note: In the GUI, the discharge overcurrent, discharge short circuit, and charge overcurrent indicators are latched by the GUI. Internal to the ISL9208, the bits are reset by a read (if the condition has been resolved). The GUI latch is provided, because the overcurrent condition goes away as soon as the FETs turn off and the bits in the ISL9208 are reset by reading the registers. So, without the latch, the indicator would not stay on long enough for the user to monitor. Reset the latch by clicking on the “Clear Overcurrent” button.

## Charge Overcurrent Testing

- Turn off the power to the board.
- Remove any load on the board Pack+ and Pack- pins.
- Turn on the ISL9208 board power supply (or connect the Li-ion cells to the pack).
- Use the GUI “CONFIGURATION” screen to set the desired charge overcurrent level and time delay.
- Turn on both FETs by clicking on the FET buttons in the GUI. When the FETs are on, the GUI indicators will indicate GREEN. Periodically click on the “Status Refresh” button on the lower right of the screen to make sure that the GUI reflects the latest status of the device.
- Use another power supply for charge emulation. With the output off and not connected to the board, set the output to just over the chosen overcurrent detect voltage threshold. (This supply should have a 2.5A limit, but will only need to provide 1.5V max).
- Connect the charge emulation power supply positive terminal to the board GND pin and connect the charge emulation power supply negative terminal to the board P- pin (see Figure 9). A current probe can be used to monitor the overcurrent details.

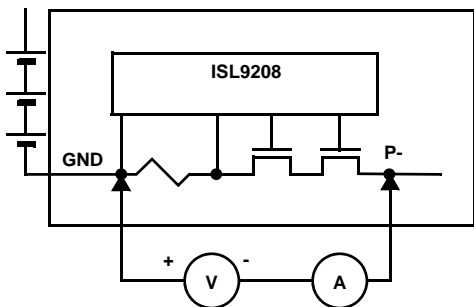


FIGURE 9. CHARGE OVERCURRENT TEST CONNECTION

- Turn the charge emulation power supply output on. This causes the ISL9208 to detect an overcurrent condition, which turns the FETs off. Figure 10 shows a charge

overcurrent condition where the charger turns on with current too high.

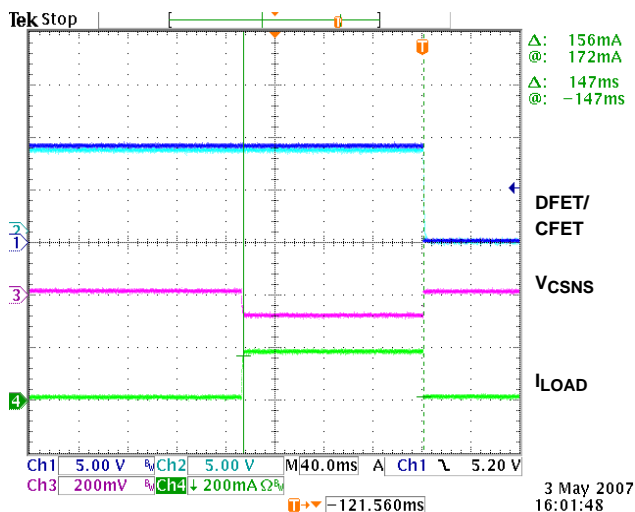


FIGURE 10. CHARGE OVERCURRENT TEST (0.1V THRESHOLD, 160ms TIME DELAY, 0.5Ω SENSE RESISTOR)

- The charge emulation power supply could have been connected across the Pack output pins - as in a “real world” operation. However, doing this requires that both the “charger” and “battery” power supplies sink current, that the “charger” supply needs to be floating when turned off (not shorted), and that the “charger” supply needs to handle a higher voltage than the input.

## Sleep/Wake Testing (Default Setting - WKPOL = 0)

The ISL9208 board can be put to sleep via commands from the PC. This sequence is described in the following paragraphs.

- To put the ISL9208 into the sleep mode, use the Register Access window to write an 80H to the ISL9208 register 4. This turns off the ISL9208 RGO output and LED.
- To wake up the ISL9208 requires that the ISL9208 WKUP pin go below its wakeup threshold. Normally, a charger would connect to the pack terminals to generate a wake-up signal. This works because the unloaded charger voltage is equal to, or greater than, the voltage on a fully charged pack. This makes the charger voltage higher than the voltage on the cells in any charge state. So, connecting the charger forces the WKUP pin low, causing the part to wake up. However, in a test setup, it is not always desirable to connect the charger. In the test setup, it is possible to force a wake-up by connecting a jumper from GND to the P- pin. Alternatively, a jumper can be connected between GND and the WKUP pin to wake the ISL9208. When using these techniques, don't leave the jumper in place.
- When the WKUP pin is pulled low, the ISL9208 wakes up and turns on the RGO output. This turns on the RGO LED.

## Sleep/Wake Testing (WKPOL = 1)

- Set the WKUP jumper to the active high position (shunt on the side closest to the push-button switch).
- Use the GUI to set the “WKUP Pin Active High” in the Configure Tab, feature set window.
- Put the ISL9208 in sleep mode as before.
- This time, the device can be waken by the press of the WKUP button on the board.

## Testing with the Microcontroller

- To operate the board using the microcontroller, power down the board
- Set the I<sup>2</sup>C jumpers to the  $\mu$ C position.
- Power up the board and restart the GUI. Now, the PC will be communicating with the microcontroller and the microcontroller will be communicating with the ISL9208.
- The GUI should power up with some color. In this case, the FET controls should be GREEN and the indicators should be green or red. (Note: there is a 5s delay after power-up before the microcontroller turns on the FETS, so the indicators may be RED when initially powered up. Clicking on the “refresh button” after 5s should change the FET indicators to GREEN). If the GUI is all gray, then there is a communication problem. If there is a communication problem, see the troubleshooting guide in “Appendix 2” on page 14.
- If the FET indicators are remain RED after 5s, then it is likely that at least one input voltage is out of range.

With the microcontroller in place, the board performs a number of automatic functions. These are:

1. The cell inputs are monitored for too high or too low voltage. If any of the cell voltages go too high, the charge FET is turned off. If any of the cell voltages go too low, the discharge FET turns off. When the voltage recovers from these excursions, back into the normal range, the FETS automatically turn on.
  2. After an overcurrent condition, the microcontroller monitors the load and turns the FETs back on when the load is released.
  3. The microcontroller monitors the temperature and turns off the cell balance if the temperature is too high or low.
  4. The microcontroller performs cell balancing (once it is enabled through the GUI).
  5. The microcontroller monitors the cell voltages and reports these voltages to the GUI. The microcontroller A/D converter accuracy is only 10-bits, so the voltage reading are not as accurate as when using only the PC interface.
- Test the overvoltage and undervoltage conditions by:
    - If Li-ion cells are being used, discharge the pack until one or more of the cells reach the undervoltage limit and the discharge FET turns off. Then, charge the

pack until the FETs turn on again and continue charging until a cell overvoltage condition is reached.

–If one power supply is being used, lower the voltage on the power supply until one or more of the cells reach the undervoltage limit and the discharge FET turns off. Then, increase the voltage until the FETs turn on again and continue increasing the voltage until a cell overvoltage condition is reached.

–If seven power supplies are used, then simply decrease or increase any individual supply until the thresholds are reached and the FET turns off (or on).

- Test the overcurrent in the same way as before, but this time, when the load is removed the FETs should automatically turn back on. In this case, with the microcontroller operating, the status indicators in the GUI may not prove to be very useful, because the microcontroller is often doing things too quickly to display on the screen. To get an indication of the operation, monitor the voltage at the VMON test point with a scope.
- Testing the cell balance operation requires the use of Li-ion cells or requires modifying the board to use 1000 $\Omega$  cell balancing resistors. (With 7-cells, a string of 20 $\Omega$  voltage divider resistors, and 39 $\Omega$  cell balancing resistors, turning on one cell balance output drops the voltage on that cell to less than 2.5V. At this voltage, the microcontroller puts the ISL9208 to sleep).
- Start the cell balance test by first observing if the cell with the maximum cell voltage exceeds the cell with the minimum voltage by more than 30mV. If so, note the cell number of the maximum voltage cell.
- Next, select “CB Max #” to be “1”. This limits the balancing to only one cell (the one with the maximum voltage).
- Use the CB refresh button (or start auto update) to update the indicators to see which cell is being balanced, it should be the maximum voltage cell. Be patient, because the microcontroller will balance for 10s, then turn off balancing for 2s<sup>1</sup>, then balance again. Also, if the maximum voltage cell is very close to the next highest voltage cell, or if there are many cells within a narrow voltage range, then any of these cells could be balanced, due to the limited accuracy of the microcontroller A/D converter.
- Next, select “CB Max #” to be “2”. This limits the balancing to two cells (the highest two voltage cells). Again refresh the CB screen periodically to see the operation of the cell balance code.
- Open the pack tab in the GUI and change some of the settings for overvoltage, undervoltage, or cell balance and re-test. Remember to click on “Write” to send the new parameters to the microcontroller.

1. If this is too long to wait, go to the “Pack Tab” and change the cell balance on/off times. Setting 1s on and 1s off is the minimum setting for cell balancing, but 2s on, 2s off is the recommended minimum, only because an autoscan of 1s can cause confusion due to the asynchronous nature of cell balance and autoscan.

Further tests on the board will likely follow the lines of battery pack testing, so can become quite involved and be very specific to the application. So, before setting up the tests, see the "GUI user Manual" for information on using the interface and see the "Microcode Reference Guide" for information about how the software works.

### Other Board Features/Options

#### Sense Resistor

The ISL9208EVAL1 (Rev C) board has three basic sense resistor "footprints" for different types of sense resistors. The basic footprint is a standard 2512 surface mount. The board uses a 0.5Ω resistor in this form factor as the default.

A second footprint is for an axial lead sense resistor. In this case, remove the resistor that was shipped with the board.

The third footprint is for a 4 terminal sense resistor with Kelvin connections. These are higher precision sense resistor devices, but are more costly.

In summary, in addition to a standard 2512 footprint, the board is laid out to handle the following resistors:

Isotek:

- SMV-R005-1.0
- SMR-R005-1.0
- LMSR005-5.0
- BVS-A-R004-1.0

TT electronics (IRC):

- OAR-5 0.005 5% LF (Mouser 66-OAR5R005JLF)

#### Additional FETs

The board has pads on the bottom of the board to handle additional power FETs. As shown in the schematic, these parallel the ones on the top of the board. While this can be used to test very high discharge current applications, the primary purpose of adding these optional FETs to the PCB was to test the performance of the FET drive circuitry in applications where higher capacitance FETs or multiple FETs were used.

If the FETs are added to the bottom of the board, monitor the FET gate drive voltage during FET turn on and turn off to determine if the response times are suitable for the application.

No tests have been made to determine if the board traces can handle the amount of current supported by the FETs on the board.

#### Microcontroller Options

The BKGD connection on the ISL9208 board allows development of new or modified code for the Freescale MC9S08QG8 microcontroller, which is supplied on the board.

The PCB provides optional connections for an external crystal for the microcontroller and brings all microcontroller terminals out for use in other application modes.

The external, 15-bit, A/D converter is not used by the microcontroller, but it could be, if changes are made to the microcontroller code. To use this, refer to the Texas Instruments ADS1100 data sheet. The version of the A/D converter used on the ISL9208 board is the ADS1100A2, which has an I<sup>2</sup>C address of "1001 010x."

#### Use with an External Microcontroller

The ISL9208 board can be used as a platform for developing code for a microcontroller other than the Freescale microcontroller on the board. To do this, set the I<sup>2</sup>C jumpers in the PC mode and use the I<sup>2</sup>C interface for communication with the external microcontroller.

To get good communication between the external microcontroller and the ISL9208, connect the "I<sup>2</sup>C GND jumper" to the GND position. However, in this case, make sure that the external microcontroller is isolated from earth ground before connecting a load or charger to the pack. This is because the board GND terminal and the P- terminal are not the same when the FETs are off.

### Related Documentation

#### From Intersil

ISL9208, ISL9216, ISL9217 Microcode Reference Guide

*ISL9208 FN6446 Data Sheet*

*ISL9208 Application Note*

*ISL9208, ISL9216 GUI User Guide*

*DeVaSys USB-I<sup>2</sup>C Software Installation*

#### From Texas Instruments

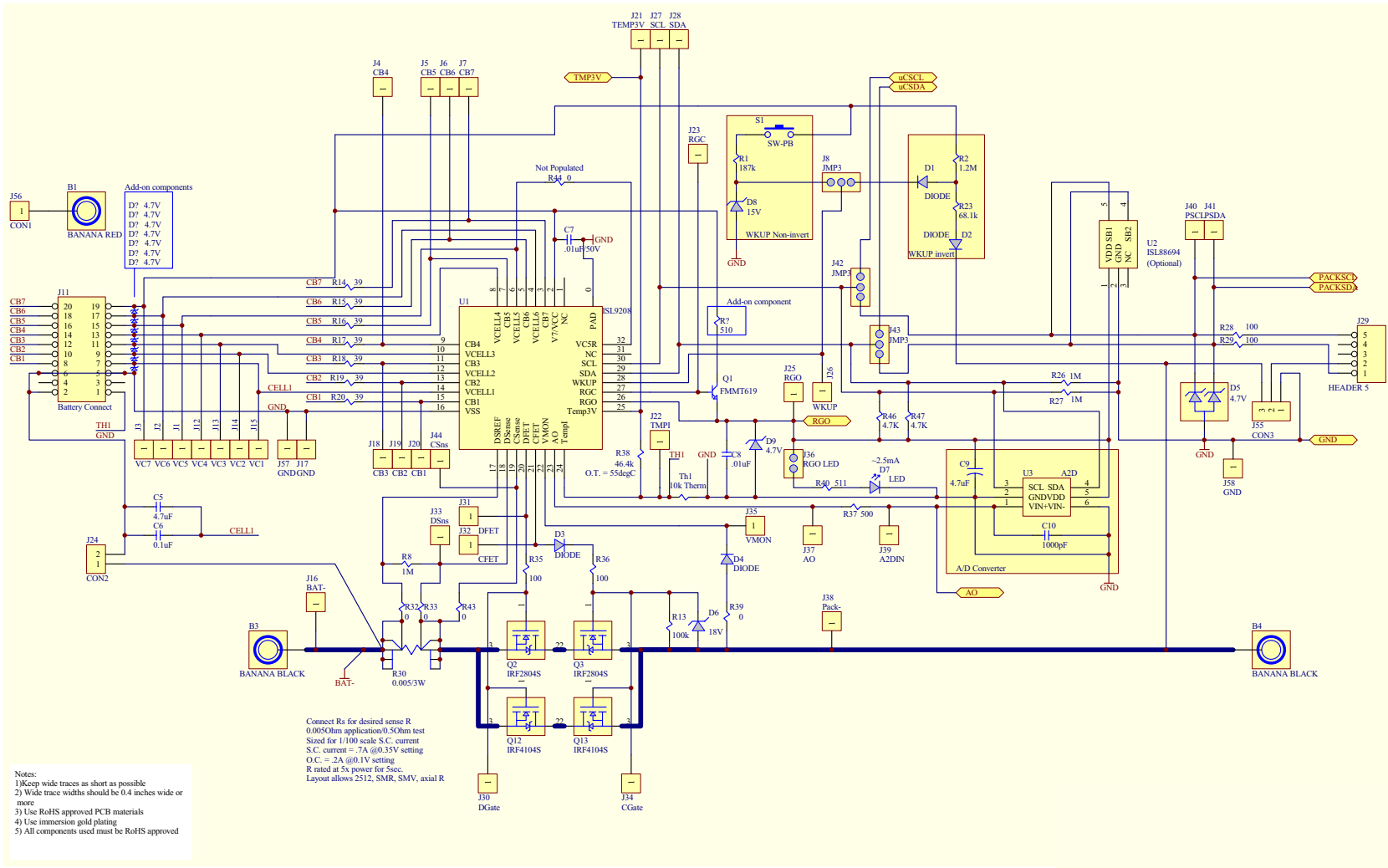
*ADS1100 Data Sheet*

#### From Freescale

*MC9S08QG8 Microcontroller Data Sheet*

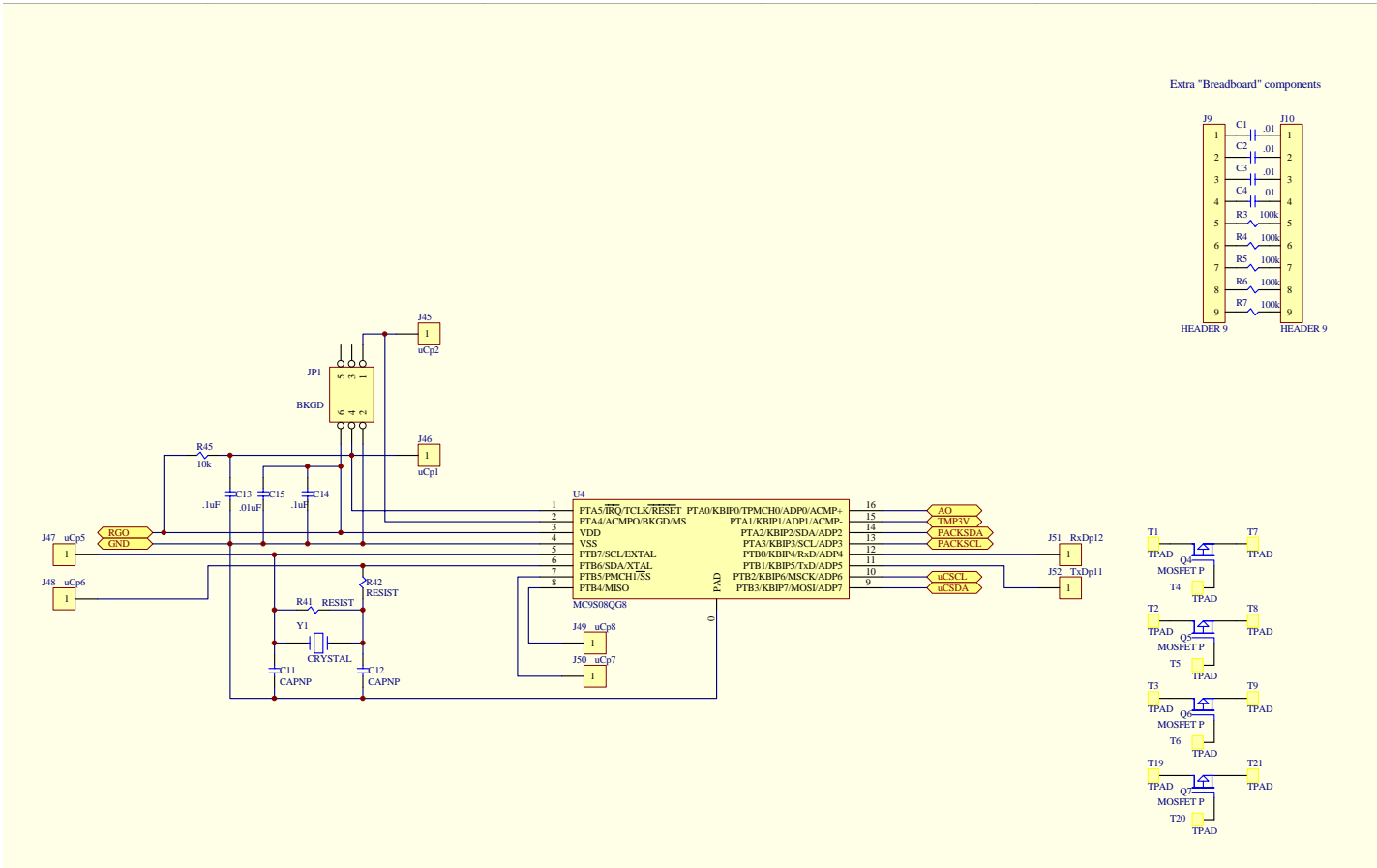
*HCS08 Microcontrollers Family Reference Manual*

# AFE Schematic

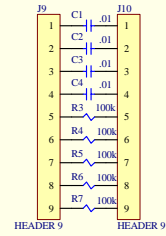




# Microcontroller Schematic



Extra "Breadboard" components



## Application Note 1334

### Bill of Materials

ITEM	QTY	PART TYPE	DESIGNATOR	FOOTPRINT	DESCRIPTION	PART FIELD 1
1	3	0	R32, R33, R43	603		*
2	1	0.51	R30			
3	7	39	R4, R15, R16, R17, R18, R19, R20	2512		Digikey: PT39AFCT-ND
4	4	100	R28, R29, R35, R36	603		*
5	3	511	R40, R37, + add-on (Q1 collector)	603		*
6	2	0.01 $\mu$ F	C15, C16	603		*
7	1	0.01 $\mu$ F	C8	805		*
8	1	0.01 $\mu$ F/50V	C7	603		*
9	3	0.1 $\mu$ F	C6, C13, C14	603		*
10	1	1.2M	R2	805		*
11	1	100k	R13	805		*
12	1	10k	R45	603		*
13	1	10k Therm	Th1	603		*
14	1	15V	D8	SOD-123		Digikey: MMSZ4702T10SCT-ND
15	1	187k	R1	805		*
16	1	18V	D6	SOD-123		Digikey: MMSZ4705T10SCT-ND
17	2	1M	R26, R27	603		*
18	1	249k	R39	805		*
19	1	1000pF	C10	603		*
20	2	4.7k	R46, R47	603		*
21	1	4.7 $\mu$ F	C5	603		*
22	1	4.7 $\mu$ F	C9	805		*
23	1	4.7V	D5	SOT23		Digikey: AZ23C3V6-FDICT-NDD
24	8	4.7V	D9 + 7 add-on (VCn inputs)	SOD-123		Digikey: BZT52C4V7-FDICT-ND
25	1	46.4k	R38	805		*
26	1	68.1k	R23	805		*
27	1	A2D	U3	SOT23-6		Digikey: 296-14299-1-ND
28	41	A2DIN, AO, CB1, CB2, CB3, CB4, CB5, CB6, CB7, CFET, CGATE, CSns, DFET, Dgate, DSns, VC5, VC4, VC3, VC2, VC1, VC6, TEMP3V, TMPI, RGC, RGO, WKUP, SCL, SDA, VC7, VMON, Pack-, PSCL, PSDA, $\mu$ Cp2, $\mu$ Cp1, $\mu$ Cp5, $\mu$ Cp6, $\mu$ Cp8, $\mu$ Cp7, RxDp12, TxDp11	J39, J37, J20, J19, J18, J4, J5, J6, J7, J32, J34, J44, J31, J30, J33, J1, J12, J13, J14, J15, J2, J21, J22, J23, J25, J26, J27, J28, J3, J35, J38, J40, J41, J45, J46, J47, J48, J49, J50, J51, J52	TP	Connector	Digikey: 5000K-ND
29	1	Battery Connect (Female)		HEADER 10X2 3M		Digikey: MSD20K-ND

## Application Note 1334

### Bill of Materials (Continued)

ITEM	QTY	PART TYPE	DESIGNATOR	FOOTPRINT	DESCRIPTION	PART FIELD 1
30	1	Battery Connect (Male)	J11	HEADER 10X2 3M		Digikey: MHC20K-ND
31	1	BKGD	JP1	HEADER 3X2		
32	2	CON2, RGO_LED	J24, J36	JP_2		*
33	1	CON3, JMP3	J55, J42, J8, J43	JP_3	Connector	*
34	4	DIODE	D1, D2, D3, D4	SOD-123		Digikey: B0540WDICT-ND
35	1	FMMT619	Q1	SOT23 - NPN		Digikey: FMMT619CT-ND
36	4	GND, BAT-	J58, J57, J17, J16	TP SM	Connector	Digikey: 5011K-ND
37	1	HEADER 5	J29	HEADER 5X1		*
38	2	IRF4104S	Q2, Q3	D2PAK		Digikey: IRF2804S-ND
39	1	ISL9208	U1	QFN32		Intersil Provided
40	1	LED	D7	LED_GW		Digikey: 490CT-ND
41	1	MC9S08QG8	U4	QFN16		Digikey: MC9S08QG8FFE-ND
42	1	SW-PB	S1	B3WN-6002		Digikey: SW425TB-ND
	4					
<b>COMPONENTS NOT POPULATED</b>						
		BANANA RED	B1	BANANA	(Not Populated)	
		BANANA BLACK	B3	BANANA	(Not Populated)	
		BANANA BLACK	B4	BANANA	(Not Populated)	
		0.01	C1	603	(Not Populated)	
		CAPNP	C11	603	(Not populated)	
		CAPNP	C12	603	(Not Populated)	
		0.01	C2	603	(Not Populated)	
		0.01	C3	603	(Not Populated)	
		0.01	C4	603	(Not Populated)	
		HEADER 9	J10	HEADER9	(Not Populated)	
		CON1	J56	TPAD	(Not Populated)	
		HEADER 9	J9	HEADER9	(Not Populated)	
		IRF4104S	Q12	D2PAK	(Not Populated)	Digikey: IRF4104S-ND
		IRF4104S	Q13	D2PAK	(Not Populated)	Digikey: IRF4104S-ND
		MOSFET P	Q4	SOT23	(Not Populated)	
		MOSFET P	Q5	SOT23	(Not Populated)	
		MOSFET P	Q6	SOT23	(Not Populated)	
		MOSFET P	Q7	SOT23	(Not Populated)	

## Application Note 1334

### Bill of Materials (Continued)

ITEM	QTY	PART TYPE	DESIGNATOR	FOOTPRINT	DESCRIPTION	PART FIELD 1
		100k	R3	603	(Not Populated)	
		100k	R4	603	(Not Populated)	
		RESIST	R41	603	(Not Populated)	
		RESIST	R42	603	(Not Populated)	
		0	R44	603	(Not Populated)	
		100k	R5	603	(Not Populated)	
		100k	R6	603	(Not Populated)	
		100k	R7	603	(Not Populated)	
		1M	R8	603	(Not Populated)	
		TPAD	T1	TPAD	(Not Populated)	
		TPAD	T19	TPAD	(Not Populated)	
		TPAD	T2	TPAD	(Not Populated)	
		TPAD	T20	TPAD	(Not Populated)	
		TPAD	T21	TPAD	(Not Populated)	
		TPAD	T3	TPAD	(Not Populated)	
		TPAD	T4	TPAD	(Not Populated)	
		TPAD	T5	TPAD	(Not Populated)	
		TPAD	T6	TPAD	(Not Populated)	
		TPAD	T7	TPAD	(Not Populated)	
		TPAD	T8	TPAD	(Not Populated)	
		TPAD	T9	TPAD	(Not Populated)	
		ISL88694	U2	SOT23-5	(Not Populated)	*
		CRYSTAL	Y1	32k XTAL	(Not Populated)	

**Appendix 1**

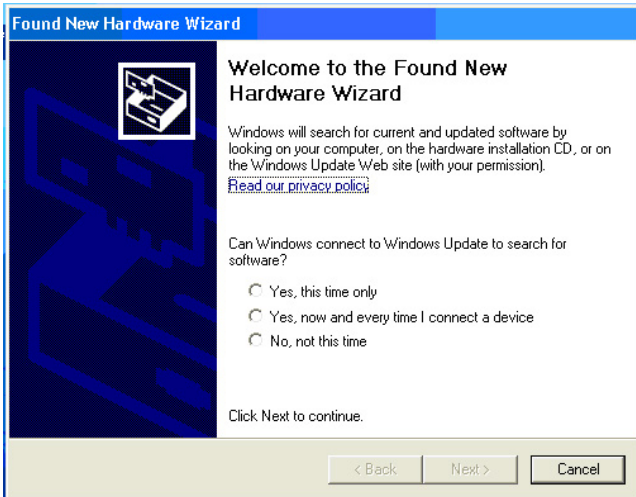
**Installing the DeVaSys USB to I<sup>2</sup>C Board Software**

Copy and extract the files from the “PC\_software.zip” to the PC at whatever location is desired.

Disconnect the DeVaSys board from the ISL9208/ISL9216 board.

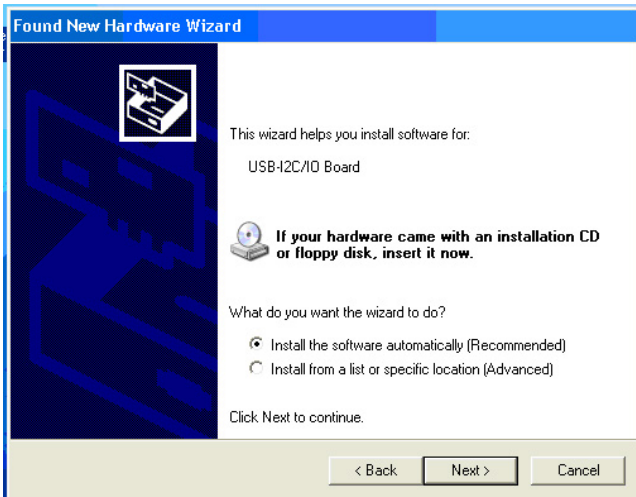
Then, plug in the DeVaSys board into the USB port.

The following screen should pop up.



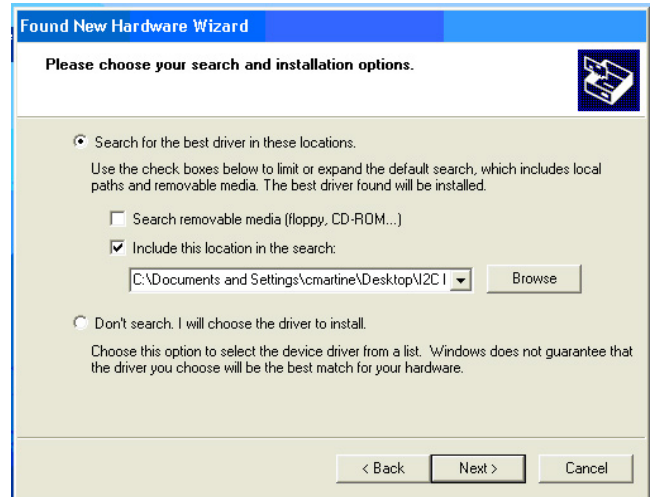
Select “Yes, this time only” and click “Next”.

Then, this screen will come up:



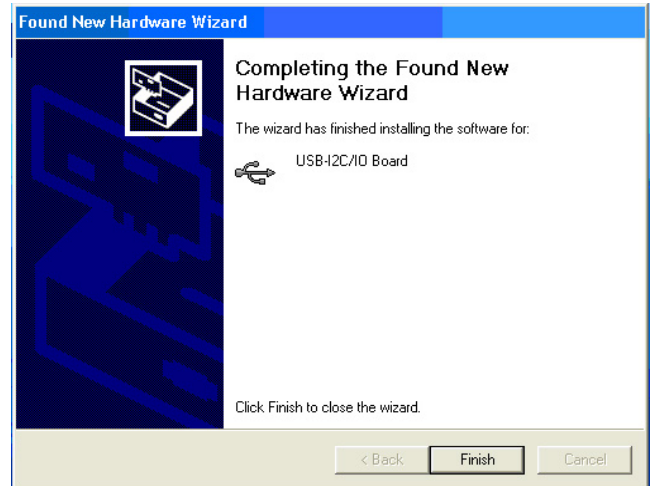
Select “Install from a list or specific location” and click “Next”

A screen like the following will come up:



Browse for the “Software” directory in the “ISL9208, ISL9216 Eval Kit SW and docs” folder then click “Next”.

This should install the software, eventually bringing up the following screen:



Click “Finish” and you’re done.

## Appendix 2

### Communication Troubleshooting

#### IF THE GUI STARTS UP WITH ALL ITEMS “GRAYED OUT”

1. Check that the I<sup>2</sup>C cable is connected properly.
2. Check that the board is powered and that the RGO voltage is ~3.3V.
3. If the RGO voltage is not powered to the right voltage, move to the power supply troubleshooting section.
4. Make sure that the board drivers are installed correctly. On the DeVaSys USB to I<sup>2</sup>C interface board, there should be one red LED and one green LED lighted.
5. Use a scope to see that the I<sup>2</sup>C communication is correct at the board. Monitor the SCL and the SDA lines while initiating a read of the ISL9208 status register. Set the scope to single trigger on the falling edge of SCL.
6. Check that the SCL and SDA jumpers (J42 and J43) have shunts on the “PC” side.
7. Check to see that the “I<sup>2</sup>C GND” jumper is in place in the “GND” position.
8. Check that the “IC GND” jumper (J24) is in place.

### Power Supply Troubleshooting

#### IF RGO DOES NOT HAVE THE CORRECT VOLTAGE

1. Check that the voltage on each of the input terminals are correct.
2. Check that there is no unexpected load on the RGO output.

### ISL9208 Troubleshooting

#### IF THE AO VOLTAGES ARE READING INCORRECTLY AT THE AO PIN

1. Make sure that the I<sup>2</sup>C jumpers are in the “PC” position.
2. Check that all cell balance outputs are off.
3. Make sure that there is no series resistance between the battery and the input of the ISL9208 and that the input voltage is between 2.3V and 4.3V.

#### IF THE AO VOLTAGES ARE READING INCORRECTLY ON THE GUI

4. Check that the RGO output is 3.3V. GUI and microcontroller calculations assume the RGO voltage is 3.3V. Any variation translates directly into errors in the GUI screen value.
5. Power down the board and stop the GUI. Power up the board and restart the GUI. This should clear any communication problems.
6. If operating with the I<sup>2</sup>C Jumpers in the  $\mu$ C position, make sure that the “Partition” setting in the Pack Tab matches the battery connection on the board.