

## ISL94208 GUI Description

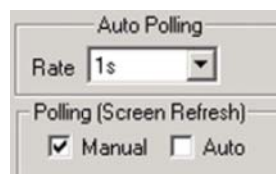
The ISL94208 Graphical User Interface (GUI) provides readout and control of all registers in the ISL94208, provides indicators of the pack status, a readout of cell voltages, a simple method of configuring the ISL94208 parameters and pack limits, and a mechanism for board calibration. See Figure 1.

### System Operating Modes

There are two states for the GUI. One state provides for automatic polling of the ISL94208EVZ board. This polling occurs at preselected rates. The other (default) state provides for manual polling of the board under the control of the user.

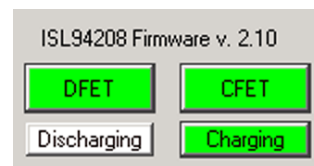
There are also two states for the evaluation board microcontroller operation. In the default state, the microcontroller on the board automatically scans the ISL94208 every 256ms and makes decisions about the pack operation. The other state is a manual scan mode where the microcontroller on the board is not scanning the voltages and no pack decisions are made. In this mode the microcontroller is waiting for communication from the GUI.

### Main Screen



### POLLING

The user selects either Manual Polling or Automatic Polling mode using the Manual and Auto buttons (lower right of the screen.) The Auto Polling rate is selectable with refresh periods of 1s, 2s, 5s, and 10s. The auto scan updates all values in the main window and in the Monitor tab.



### FIRMWARE VERSION

At the top of the screen (above the FET indicators) is the device part number and the version number for the microcontroller firmware on the board.

### CFET/DFET

The CFET and DFET buttons show the status of the Charge FET and Discharge FET on the board. When the microcontroller is in the auto scan mode, these buttons are only indicators.

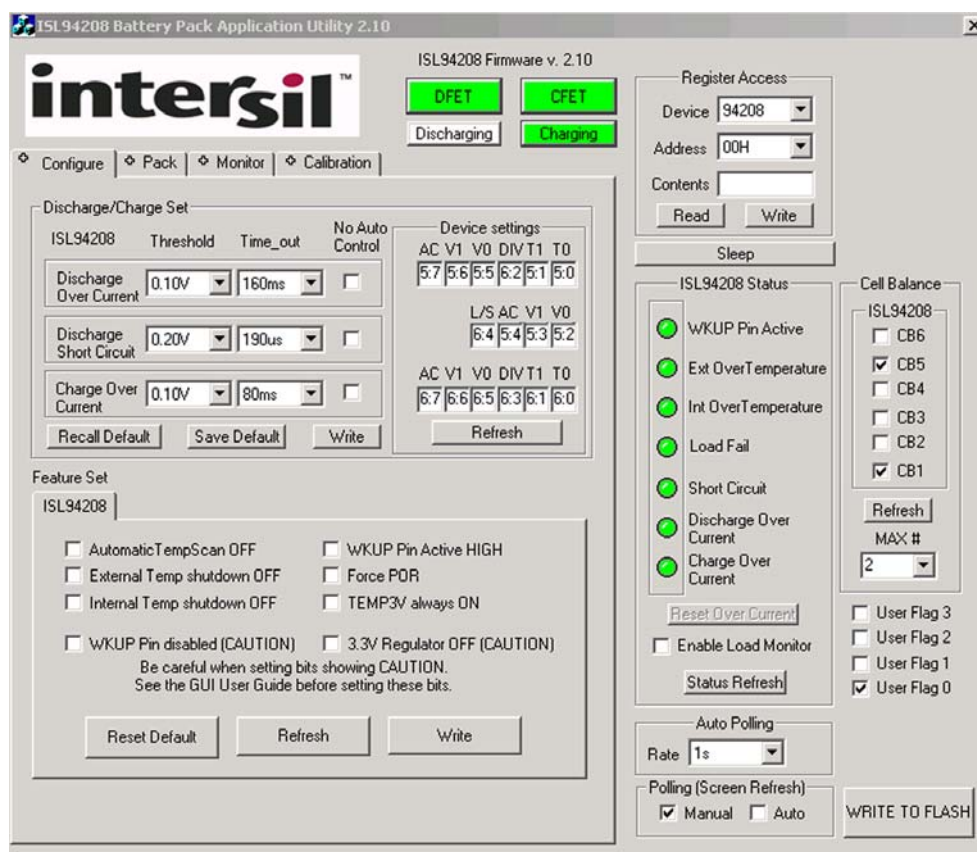


FIGURE 1. ISL94208 GUI Main and Configure Screens

Clicking on a button changes the color, but the microcontroller over-rides the setting. Check this by clicking on the “Status Refresh” button.

When the board is set for manual scan mode (select this mode by monitoring any single voltage, not ALL, in the Monitor Tab), then the CFET and DFET buttons are not controlled by the microcontroller, so the system allows the FETs to be turned on and off by the user. Click on either of these buttons to turn the respective FET on or off. The indicator is Green to indicate that the FET is on. The default state in manual scan mode is OFF. With the microcontroller running auto scan, the default state is ON (if all pack conditions are within normal operating limits).

## CHARGE/DISCHARGE INDICATORS

Below the CFET/DFET buttons are the charging and discharging indicators. These glow GREEN when the hardware indicates that there is charge or discharge current.

To test charge/discharge operation requires a charge or load current across the sense resistors. A load current of up to 1A is needed between the B- terminal and P- terminal for the board to detect load current. A charge current of 250mA between B- and CHG- should be high enough current for the board to detect. A good way to test this when applying the load or charger current is to set the GUI to Auto Polling mode so the status indicators are automatically updated. In Manual Polling mode, it is necessary to periodically click on the “Status Refresh” button.



## REGISTER ACCESS

The register access box allows the user to select the address to read and write. During a read, the value is returned in the “Contents” window. During a write, the value in the “Contents” window is sent to the device at the address specified. The operation takes place when the read or write buttons are pressed.

The GUI is set up to handle a single ISL94208EVZ board.

Note that when using this Register Access, and writing to registers 5, 6, and 7 of any device, a write must first be made to register 8 to set an enable (80H). Then, once the write is complete, register 8 should be reset to prevent accidental writes.

Do not write to registers 9H or AH.

## SLEEP BUTTON

A sleep button (below the Register Access controls) has been added to the GUI. Clicking on this button causes the microcontroller to put the pack to sleep. To wake the pack it is necessary to do one of the following.

- Press the WKPO\_L button (WKPOL = 0; default).
- Press the WKPO\_C button (WKPOL = 0; default).
- Press the WKP1 button (WKPOL = 1).
- Connect a charger to the board between VBAT and CHG-.
- Connect a load to the board between VBAT and P-.



## STATUS

The status box is simply a graphical representation of the contents of the two status registers of the ISL94208. In auto scan, these values are updated at the scan rate. In a manual scan, they are updated when the refresh button is pressed.

In this area is also a box that allows the user to turn on or off the load monitor function of the ISL94208. When the box is checked, the load monitor is on.

The “Reset Over Current” button in this area clears the discharge over current, charge over current, and short circuit flags after one of these events occurs. This latch is part of the microcontroller code on

the board and the GUI and is not part of the ISL94208 functionality. Without the latch, these indicators do not stay on long enough after an event for the user to detect. This is because the FETs open in response to the over current event. Once the FETs open, the over current condition goes away and the ISL94208 devices immediately clear the over current indicator bit in the device. When the microcontroller is operating, it manages the over current conditions.



## CELL BALANCE

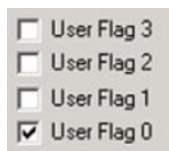
In the cell balance block, the status of the cell balance outputs are shown, plus the user has control of turning cell balance outputs on or off. A change to the cell balance output occurs immediately on clicking on one of the cell balance boxes.

There is a box that limits the number of cells that can be balanced at the same time. When this limit is reached, all other cell balance controls are disabled.

When the microcontroller is operating and running automatic scan operations and controlling the pack, the user can still turn on or off any of the cell balance outputs.

When the automatic cell balance routine is being executed by the microcontroller, the “Refresh” button updates the screen indication of which balance outputs the microcode algorithm has turned on or off. These cell balance indicators are also updated when the automatic polling is active.

The cell balance routine is always active. However, balancing does not start unless all conditions allow it. The primary limit to cell balance is that balance only happens when balancing is set to balance during charging and a charge current is detected by the hardware. This is the default condition. The board can also be set to balance during discharge when the board detects a discharge current. This setting is controlled in the Pack tab (See Section , “Pack TAB,” on page 5.)



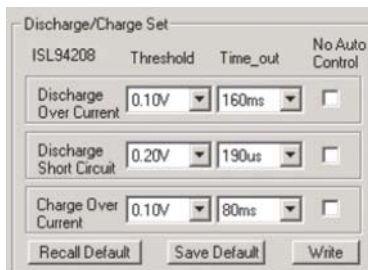
## USER FLAGS

There are 4 buttons on the main screen to indicate the status of four user flags in the ISL94208. These simply control and indicate the condition of four bits in the device register space. When the microcontroller powers up, it

sets User Flag 0 to indicate that the code has been installed and is operating. The other three tabs are for general purpose use.

## Tabbed Selections

There are four tabs on the main window. These are Configure, Monitor, Pack, and Calibration.



## CONFIGURE TAB

The Configure tab provides simplified on screen access to the configuration registers of the ISL94208. These registers are also controlled by the Register Access block, but the Configure Tab shows real values, making it easier to make selections.

The discharge and charge thresholds and time-outs are selected by drop down menus, which have all of the settings in one drop down menu, even though the settings span multiple bits and even multiple registers. For example: the Discharge Over Current time-out setting shows 2.5ms, 5ms, 10ms, 20ms, 160ms, 320ms, 60ms, and 1280ms. The value is set by programming bits 5 and 6 in register 05H and bit 2 in register 06H. The auto control bit for Discharge Over Current is set by bit 5 in register 05H.

When “Write” is pressed, the values for all configuration registers and the Auto Control bits are written to the ISL94208. This entire procedure includes setting the enable bits in register 08H prior to writing the individual values and resetting the write control bits when the operation is over. This procedure is also used for changing the threshold levels and the other time out conditions.

Note: It appears that in Rev 2.01 of the GUI code, the software does not properly set the write enable bits in register 08H prior to changing the configuration values. To more reliably change the configuration, write E0H to register 08 prior to writing the new configuration values. To be sure of the configuration setting, click on the Refresh button to read back the value in the ISL94208.



At the conclusion of the write operation, the registers are read and the settings returned from the ISL94208 are displayed in the “Device Settings” area. This display is a simple binary representation of the registers. During an auto-polling operation, these values are automatically updated.

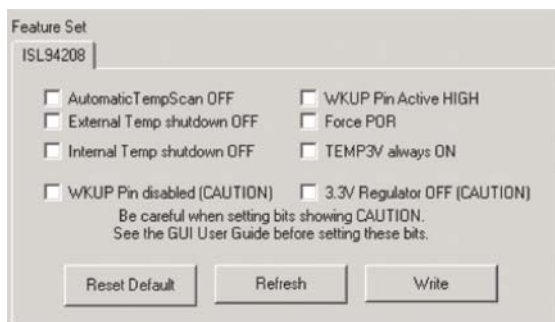
In each of the device settings bits, there is a value that shows where the bit is located. For example, the T0 bit for discharge over current is labeled as 5:0.

This means that the bit is in register 5, bit 0. The device status display is read only.

The normal default setting is “all bits equal to zero.” So, clicking on “Recall Default” sets all of the bits to zero. Then, clicking on “Write” loads the values into the device.

There may be cases where other default values are desired. If so, the user can click on “Save Default.” This stores (in the GUI program) the current settings of the configuration registers. (This is a volatile setting, so it is lost when the GUI is closed.) After clicking on “Save Default” these are the configuration values that will be recalled when clicking the “Recall Default” button.

In either manual or automatic polling, clicking the “Refresh” button processes a read of the device and the values are updated based on the settings of the ISL94208.



Also in the configure tab is a screen for setting the feature bits. The check boxes indicate the current setting of the bit and allow a change to be made. The box is checked if the feature is active.

As with the Charge/Discharge parameters, it is recommended to write E0H to Register 08H prior to changing any of these settings. Clicking on a setting two times (leaving enough time for the command to complete) also works. To be sure about the setting, click on Refresh to read back the status of the ISL94208.

For bits labeled “CAUTION”, the user is advised to consider the response to system operation when these bits are set.

- Clicking on the “3.3VRegulator OFF” button turns off the ISL94208 regulator, which turns off the microcontroller as well as some functions within the ISL94208. This bit should only be set if there is an external 3.3V supply connected to RGO.
- Clicking on the “WKUP pin disabled” will turn off the wake up circuit of the ISL94208. If the part goes to sleep with this bit set, the device cannot be waken by hardware. The only way to wake the part in this case is to supply an external RGO and use an I2C command to reset the bit.

The “Refresh” button re-reads the register and sets the check boxes accordingly. The “Reset Default” button clears the check boxes to reset the device default values. This does not write the values to the device until the “Write” button sends the screen values to the ISL94208.

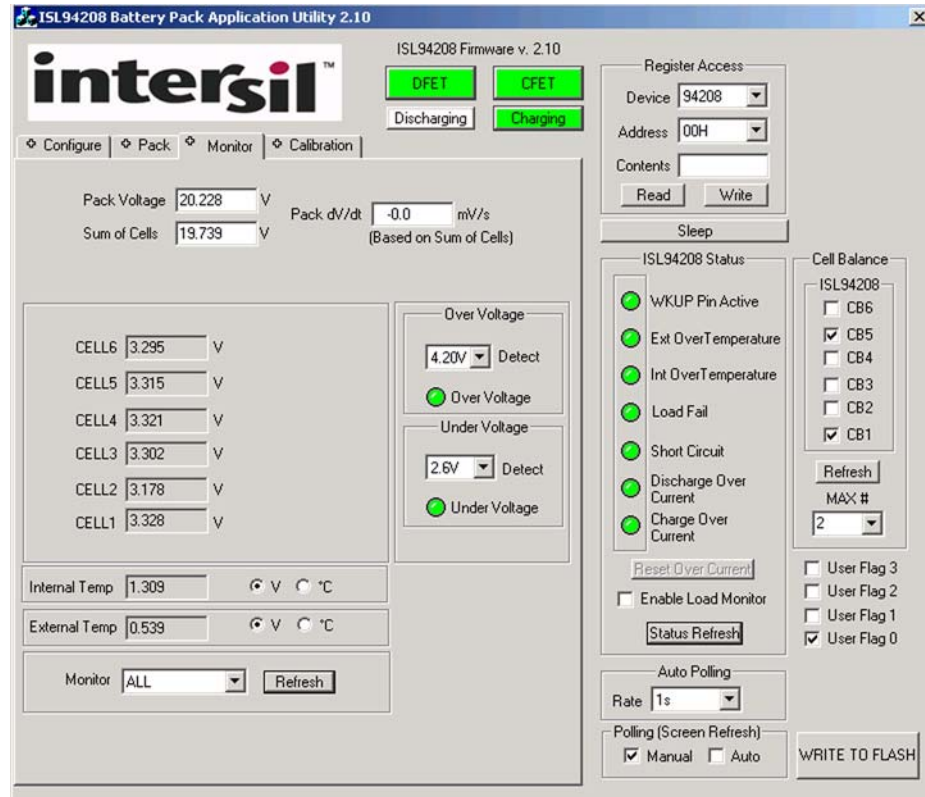


FIGURE 2. ISL94208 GUI WITH MONITOR TAB

## MONITOR TAB

CELL6	3.295	V
CELL5	3.315	V
CELL4	3.321	V
CELL3	3.302	V
CELL2	3.178	V
CELL1	3.328	V

The monitor tab provides information on the analog voltages returned from the board. This reading is based on the AO of the ISL94208. The AO voltage is converted to digital by the 10-bit A/D converter in the microcontroller.

Internal Temp	1.309	V	°C
External Temp	0.539	V	°C

The internal and external temperatures are reported as either a voltage or a temperature.

The user chooses which units are to be displayed. In the case of internal temperature, the temperature value is calculated converted from voltage to °C using data sheet values for 25°C(1.31V) and the voltage/temperature slope (-3.5mV/°C.)

The calculation for internal temperature is:

$$C = 85 + (\text{intTempVoltage} - 1.1)/(-0.0035)$$

Where intTempVoltage is the internal temperature value converted from the ISL94208 AO output. Note, the AO output is 1.1V at 85°C.

For the external temperature, the conversion is based on the values of the resistor divider on the board, the RGO voltage, and a table of values for a typical thermistor from Murata.

The calculation for external temperature is:

$$R = (46400 \times \text{extTempVoltage}) / (3.3 - \text{extTempVoltage})$$

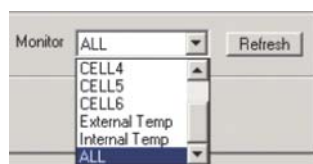
Where 46400 is the value of the top resistor in the divider, R is the resistance of the thermistor, and extTempVoltage is the external temperature value converted from the ISL94208 AO output.

The “R” value in the equation is then divided by 1000 and compared to Resistance values in the Table 1 to get the corresponding temperature value. The GUI algorithm does a linear interpolation between two points to get the final result.



TABLE 1. EXTERNAL TEMPERATURE LOOKUP TABLE

Temperature (°C)	Resistance (kΩ)	Temperature (°C)	Resistance (kΩ)
-20	68.2	35	6.95
-15	53.6	40	5.83
-10	42.5	45	4.92
-5	33.9	50	4.16
0	27.2	55	3.54
5	22	60	3.01
10	17.9	65	2.59
15	14.7	70	2.23
20	12.1	75	1.92
25	10	80	1.67
30	8.31	85	1.45



The user can select to Monitor “ALL” voltages or only a specific cell or temperature. When monitoring only one voltage, that voltage is constantly placed on the AO of the ISL94208, making it easy to monitor the

cell voltages with a meter. In Monitor “ALL”, the voltages are sequentially obtained and automatically updated in the microcontroller. The values in the Monitor screen update when clicking the Refresh button. The values are automatically updated when Auto Polling is selected.

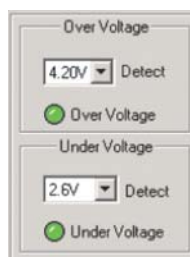


The monitor tab shows two pack voltage values. The top one, Pack

Voltage, is obtained by the microcontroller from hardware on the board. The ISL94208EVZ Evaluation board is equipped with a switched voltage divider ( $\div 16$ ) that the software uses to read the pack voltage. The Pack Voltage window shows a voltage equal to 16 times the voltage from the divider.

The “Sum of Cells” value in the Monitor Tab shows the sum of all voltages shown for the active cells. Both are included to give an indication of the performance of the board.

The dV/dt indicator in the monitor window shows the change in the “Sum of Cells” over time. It detects the difference in the reading every Automatic Polling cycle and shows the result in mV/s. The value is for indication only, as there is no functionality tied to it. The purpose was to give the user the ability to experiment with various charge and discharge algorithms.

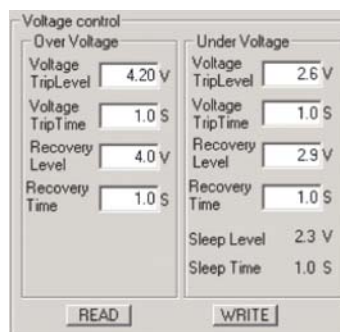


The monitor window also shows an Over Voltage and Under Voltage indicator. By default, these are set to the same values as programmed into the microcode on the ISL94208EVZ board (see board thresholds in Section , “Pack TAB,” on page 5.)

Otherwise, these values are not associated with the over voltage or under voltage conditions on the ISL94208 board. If the settings are kept at the same thresholds as

the on the board, then these provide simple visual indicators that the voltage has exceeded an over voltage level or dropped below an under voltage level to verify that the ISL94208 board is at or near a threshold level. The indicators are normally green, but show red in a fault condition.

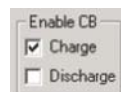
## PACK TAB



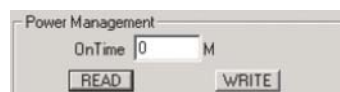
This tab provides inputs to the microcontroller for setting battery pack parameters controlling the operation of the pack by the microcode. The use of these parameters by the microcontroller are detailed in the “ISL94208 Microcode Reference Guide.”

These values are updated on the screen with the various

“read” buttons and any changed values are written back to the microcontroller by the “Write” buttons. The write button stores the value back to the microcontroller and the values are saved only in RAM.



The Cell balance window of the Pack Tab allows the user to set the pack such that balancing will not happen, will happen during charge, will happen during discharge, or will happen during both charge and discharge. Balancing only happens if the board detects that current is flowing (and in the proper direction.) There is no balancing if a box is checked, but there is no current detected.



The Pack tab also has a Power Management window. This allows the user to set a time, from 1 minute to 15

minutes for the pack to go to sleep when there is no current detected flowing into or out of the pack. When the value is “0”, the pack does not go to sleep.

The Sleep Level is “hard coded” in the software to be 2.3V and the sleep time is hard coded to be 1.0s.

Since the GUI changes for Pack parameters are made in the microcontroller RAM, inducing a sleep mode on the ISL94208 causes these values to be lost when the microcontroller is turned off. The “Write to Flash” button in the main window allows the user to store changes back to the microcontroller Flash for permanent storage. (The procedure is described in Section “This may not be the preferred approach, but any change would require an update to the microcode. Another solution to this

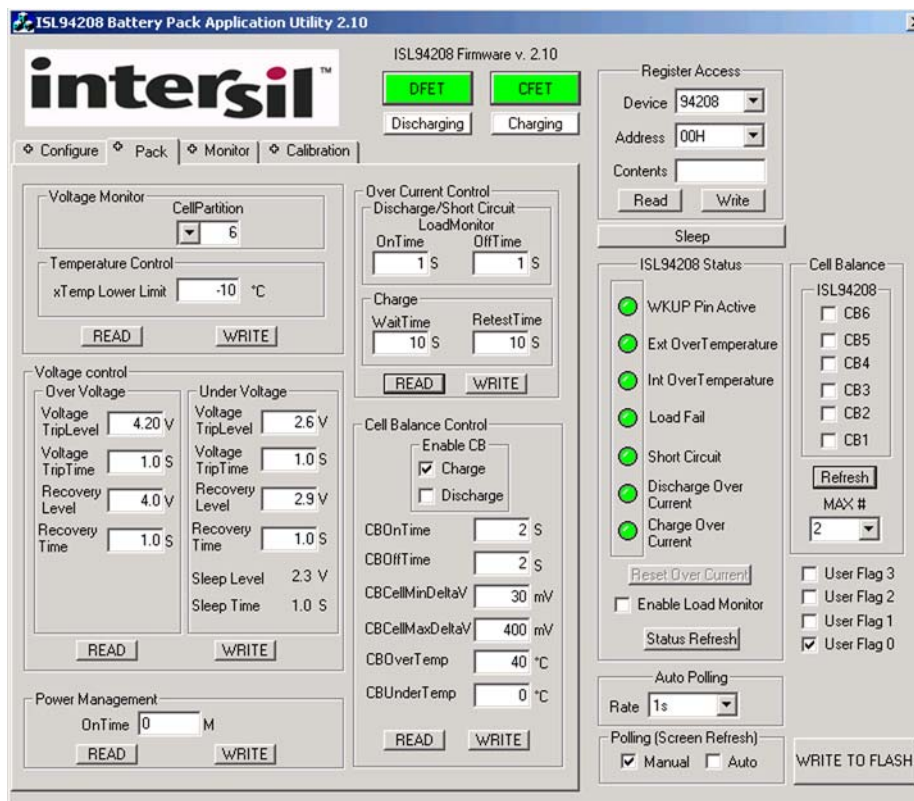
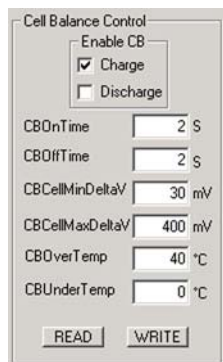


FIGURE 3. ISL94208 GUI WITH PACK TAB

condition, for example, would be to turn off the discharge FET, but leave the charge FET on. Then, when there is charge current, the lower voltage cells could balance faster than the other cells to try to recover the pack capacity. But, this approach might not be the proper response to other conditions that show up with the same symptoms, which would prompt other conditionals. Since each application is different, the microcode provided, takes a simple approach.” on page 9).



The Pack Tab also contains settings for the Cell Balance Control. As with the other sections, the values are read from the microcontroller with the “Read” button and written to the microcontroller with the “Write” button.

The cell balance control values are the Cell balance on and off times, the minimum voltage differential required before a cell is balanced, the maximum deviation between the lowest and highest voltage cell, and temperature limits on the balancing.

More information about the balancing operation is presented in Section , “ISL94208 Cell Balancing,” on page 8.

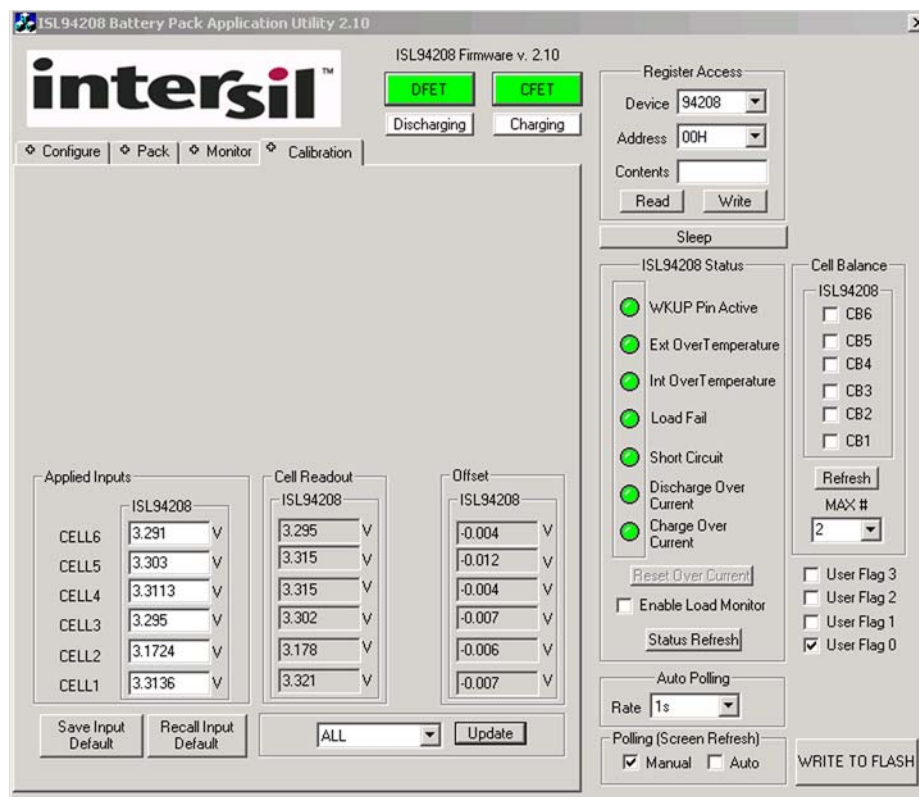


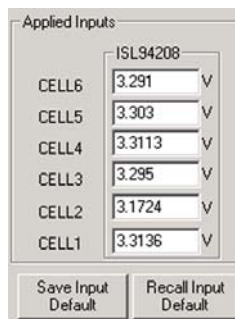
FIGURE 4. ISL94208 GUI WITH CALIBRATION TAB

## CALIBRATION TAB

The configuration Tab input column allows the user to define the voltage at which the board is calibrated. This single point calibration provides very good accuracy. It is usually performed at a critical voltage, such as at a voltage comparable to the “flat” part of a discharge curve or near the over voltage detection.

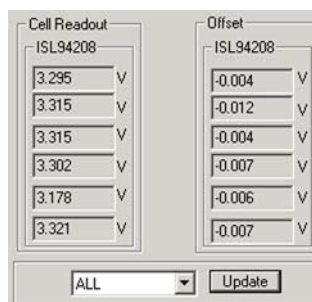
The GUI powers up with a default input voltage of 3.3V, so the user could apply a power supply with exactly 3.3V per cell to perform the calibration. However, getting a supply that can support 6 input cells to near 1mV accuracy is often difficult. So, the user can instead provide a supply that is close to the desired voltage and use a meter to provide an accurate measurement of the input voltage for calibration (as shown in Figure 4).

The input voltage can be measured at the pins of the evaluation board or at the terminals of a battery stack or power supply. Normally the voltage at the pins of the board is the preferred method for repeatability in manufacturing.



## Saving the Applied Input Values

During calibration of multiple boards, the measurement of input values can be performed once, then saved using the “Save Input Default” button in the Configuration Tab. This saves the data to a “.CIV” file. For the second and subsequent boards, if the supply voltages do not change, then the previously entered power supply voltages can be loaded using the “Recall Input Default” button to load the previously saved “.CIV” values.



## Programming the offset

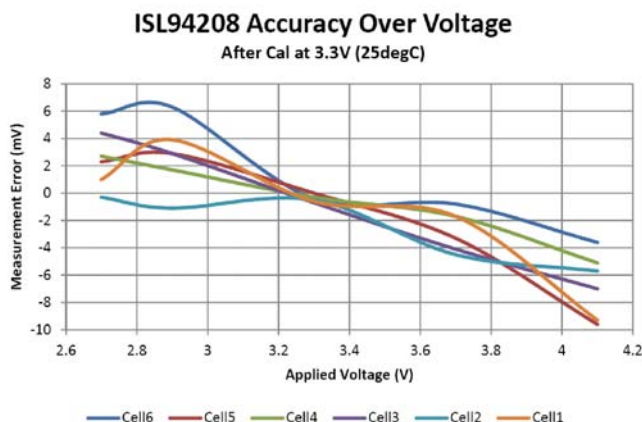
Once the input voltages are entered into the GUI screen, the user clicks on the Update button. This instructs the microcontroller to do a scan of the available cells and enter the cell voltage values into the Cell Readout column. An offset is automatically calculated and saved to the microcontroller RAM. The calibrated value remains in RAM as long as the microcontroller is powered. The calibration values can be programmed into the

microcontroller flash using the procedure defined in the Section , “Saving Microcode Changes,” on page 9.

This single point calibration removes errors caused by input voltage drops, the ISL94208 input level shifters, AO voltage drops, ADC conversion errors, and errors in the voltage reference.

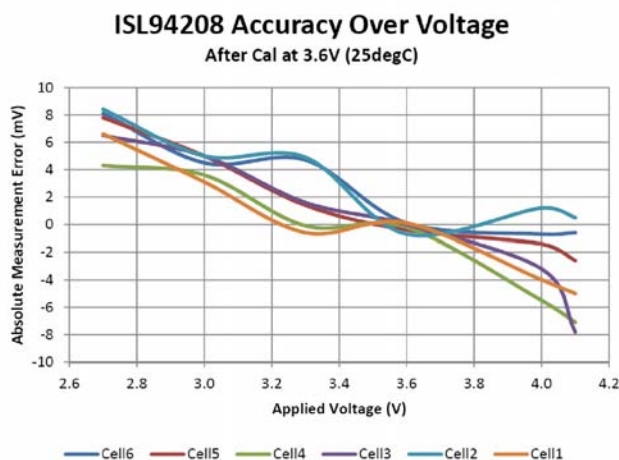
As an example of the calibration procedure and accuracy, a board was calibrated at about 3.3V per cell, as shown in Figure 4. This shows that the un-calibrated error was about 30mV.

Then the cells were measured over the range of 2.7V to 4.1V, with the results shown in Figure 5.



**FIGURE 5. SINGLE POINT CALIBRATION ACCURACY RESULTS, ROOM TEMP**

The same board was re-calibrated at 3.6V, and the accuracy measured again, with mostly the same results.



**FIGURE 6. SINGLE POINT CALIBRATION ACCURACY RESULTS, ROOM TEMP**

## Resetting Calibration

If a board needs to be calibrated again, then the offsets must be set to zero first. Do this by setting the Applied input values to match the Cell Readout values. Then click the “Update” button. This should set the Offset values to zero. If it does not, then

repeat until they are all zero. Now, you can use the procedure above to set new offset values.

## ISL94208 Cell Balancing

The operation of the cell balance algorithm is described in more detail in the “ISL94208 Microcode Reference Guide”. However, a quick description is provided here.

The microcontroller code essentially runs the cell balancing algorithm constantly. However, cells are actually balanced only when all conditions allow it. Conditions that control cell balancing are differential cell voltage, temperature, open circuit detection, and current sense. Of course, if there are events that turn off the charge/discharge FETs, such as under voltage detection, then current will go to zero and balancing is also turned off. For a graphical depiction of the cell balance routine, see Figure 7.

In general there is no balancing, if:

- the cell balance off timer is counting,
- the temperature is too high or too low for balancing,
- the difference between the highest voltage cell and the lowest voltage cell is lower than the specified minimum voltage differential, meaning that the cells are already balanced,
- the difference between the highest voltage cell and the lowest voltage cell exceeds a specified maximum,
- an input open circuit is detected,
- balance during charge is selected and there is no charge current
- balance during discharge is selected and there is no discharge current.
- the maximum number of cells are already being balanced.

Once the general conditions are met, then balancing for each cell proceeds, starting with the cell with the highest voltage differential from the lowest voltage cell.

If there is no hardware that detects current flow, then balancing is disabled.

The conditions for Cell Balance that are programmable through the GUI are described below.

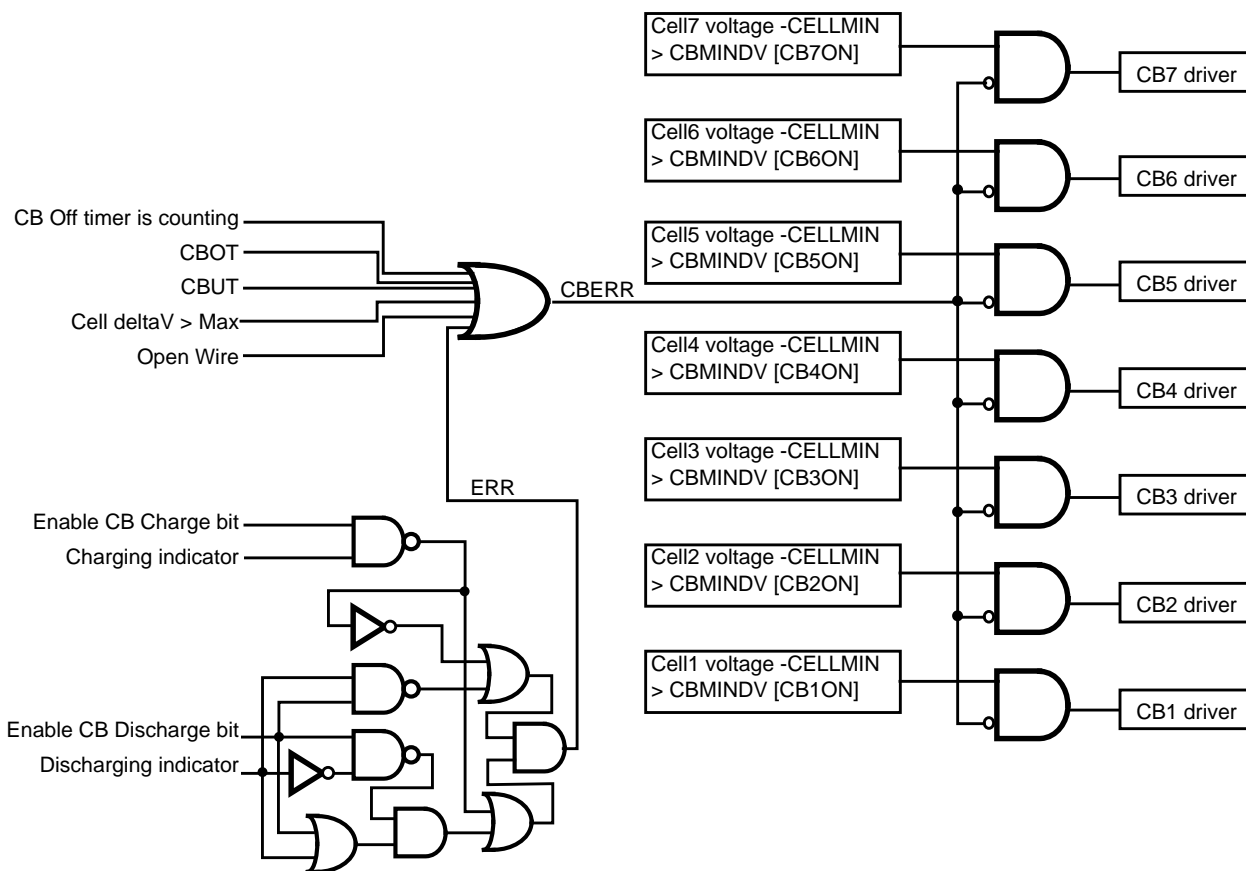
### CELL BALANCE MIN/MAX

The CBCellMinDeltaV parameter specifies the differential between the highest voltage cell and the lowest voltage for a

CBCellMinDeltaV	<input type="text" value="30"/>	mV
CBCellMaxDeltaV	<input type="text" value="400"/>	mV

balanced pack. Ideally, this value is zero to achieve cells with the same capacity. However, due to variations in cell voltage measurement through inaccuracy in the ISL94208 level translation and the resolution of the microcontroller this is the value chosen for the default. The 30mV value could be a little smaller, based on the calibrated results above, but overall, the system should allow for a minimum of 15mV variation between cells over temperature and voltage and the microcontroller minimum resolution of 3mV. So, if changing this value, the recommended minimum CBCellMinDeltaV value is 20mV. If the value is set too low, then it is possible for the system to balance the wrong cell.





**FIGURE 7. Cell balance operation**

The CBCellMaxDeltaV value is the maximum allowable differential between the minimum voltage cell and the maximum voltage cell. The most likely cause for this condition is that one or more cells reaches minimum capacity during discharge, so that the voltage on those cells drops very fast.

The ISL94208EVZ kit microcode treats a large differential between the high voltage cell and the low voltage cell like a cell open circuit condition, where the microcontroller turns off the charge and discharge FETs and turns off cell balancing.

This may not be the preferred approach, but any change would require an update to the microcode. Another solution to this condition, for example, would be to turn off the discharge FET, but leave the charge FET on. Then, when there is charge current, the lower voltage cells could balance faster than the other cells to try to recover the pack capacity. But, this approach might not be the proper response to other conditions that show up with the same symptoms, which would prompt other conditionals. Since each application is different, the microcode provided, takes a simple approach.

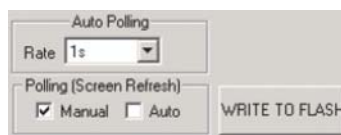
## ISL94208 Flash Programming

Because of memory limitations in the microcontroller, it is not possible to include EEPROM re-programming through the I2C port. So, programming changes to Flash is a multi step process.

The microcontroller contains the base code when the board is assembled, so it is not necessary to make any changes for basic operation of the board.

Whether it is desired to change the operation of the microcontroller code or change the calibration coefficients in the microcontroller Flash, the process of loading the modified code is the same. The following procedure describes how to modify the Flash memory for calibration and pack configuration parameters.

First, the board is calibrated using the procedure described in the section "Calibration Tab" on page 7 above. This procedure sets the calibration values into the microcontroller RAM. If there are any changes to the operation of the pack (for example different over voltage or under voltage thresholds) make these changes as described in Section , "Pack TAB," on page 5.



## Saving Microcode Changes

Once calibration values, changes to Pack settings, or device parameters are

saved in RAM, click on the "Write to Flash" button. This brings up a window as shown in Figure 8.

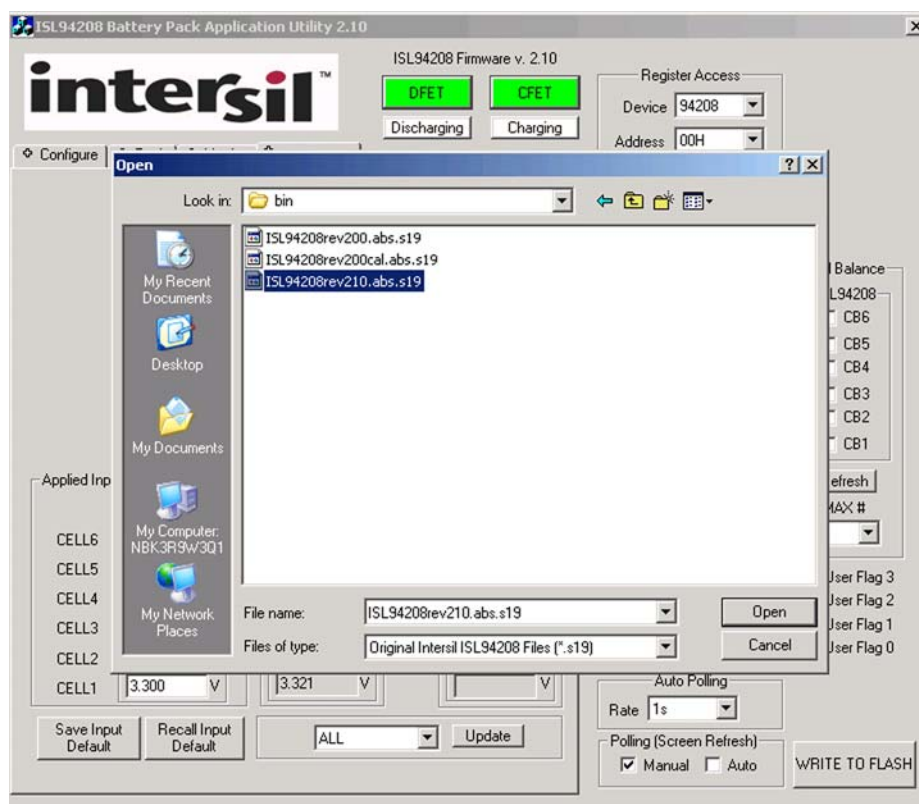


FIGURE 8. ISL94208 GUI WITH “WRITE TO FLASH” OPEN FLASH DIALOG BOX

In this window, select the “base” code and click on “OPEN”. The base code will be:

- ISL94208rev210.abs.s19.

Make sure that you pick the correct file.

The next window that appears will look similar, but you now change the file name to something appropriate and click “SAVE” to save the new code. As an example, Figure 9 shows a modified file name of ISL94208rev210cal.abs.s19. But, you can name the file anything else, but with a suffix of “.abs.s19”. If you want to keep the released, default, base code separate, do not click SAVE without changing the name of the file.

## Loading code into the microcontroller

The following procedure is used for loading the microcontroller with new, factory provided updates of the microcode as well as loading modified or calibrated code.

The code is loaded into the microcontroller using the Freescale USB/Multilink module and Code Warrior tools. So, these must be available prior to starting the programming sequence.

1. Start the Simulator.  
Use the program called HIWAVE.EXE, located in the prog folder in the CodeWarrior installation folder (c:\Program Files\Freescale\CW08 v5.0\prog). The Simulator appears as shown in Figure 10.
2. Select *Set Connection* from the Component menu. The *Set Connection* dialog box appears (see Figure 11).

3. The CPU derivative for this project is in the HCS08 subfamily, so select HCS08 from the Processor pull-down menu. Select P&E Multilink/Cyclone Pro in the Connection pull-down menu. Press OK.
4. The *Connection Manager* dialog box appears (See Figure 12.) Press Connect.
5. From the File menu, select *Load Application....* The *Load Executable File* dialog box appears. (See Figure 13.)
6. Browse to find and select the base code (the default code initially programmed on the board), a newly revised program (provided by Intersil or your own new code), or the new configuration and calibrated code in the modified “.abs.s19” file.  
Press Open.

This brings up a dialog to start the flash memory write. Select “Yes”. This begins the write of the new flash values. When this process completes, the USB/Multilink cable can be removed from the board and the board power cycled to begin the new code operation.

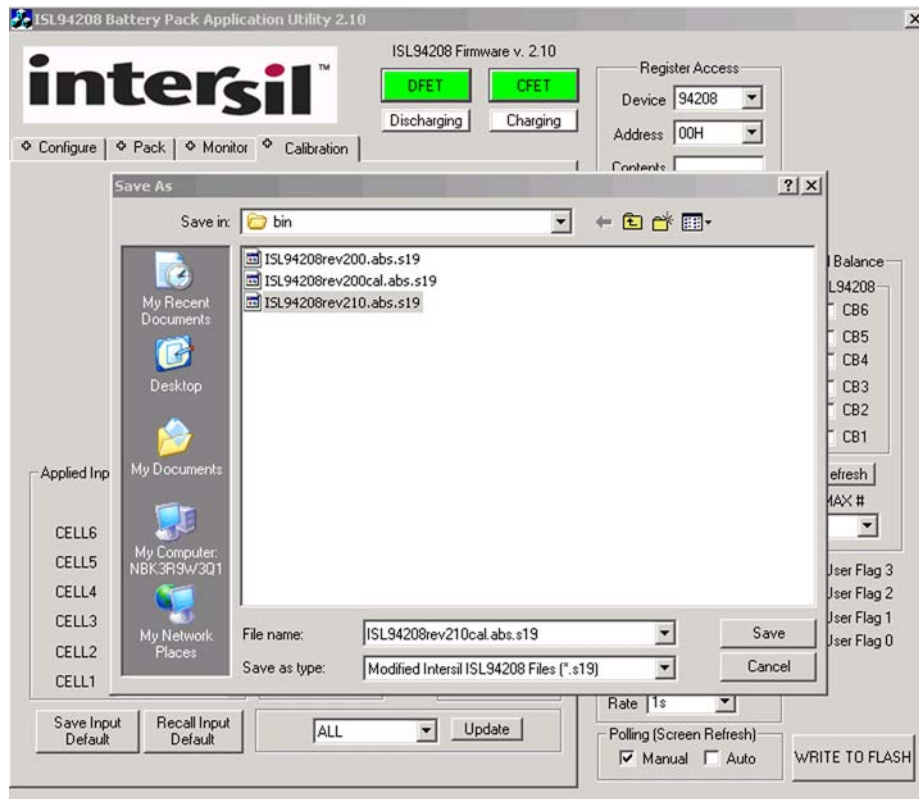


FIGURE 9. ISL94208 GUI “WRITE TO FLASH” WITH SAVE TO FLASH FILE DIALOG BOX

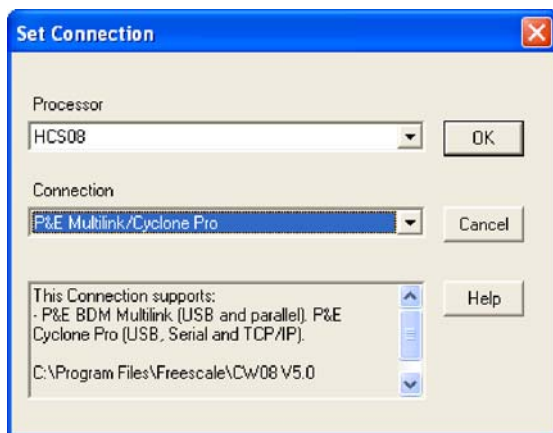


FIGURE 11. CODE WARRIOR SIMULATOR SET CONNECTION

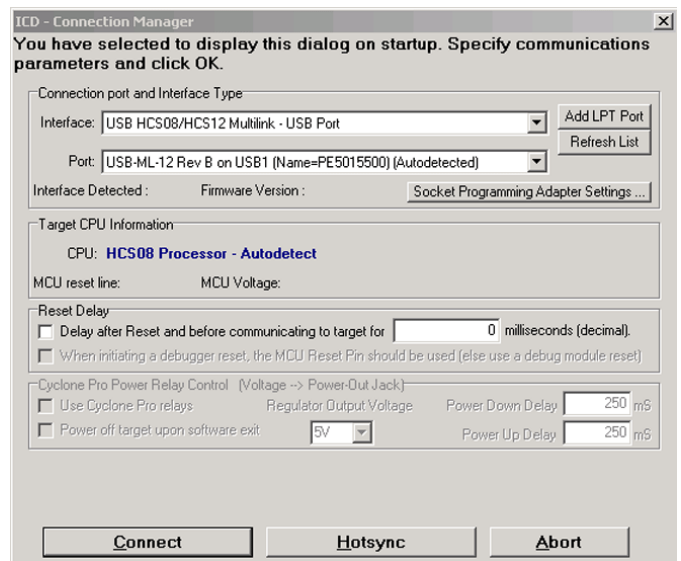


FIGURE 12. CODE WARRIOR SIMULATOR CONNECTION MANAGER

## ISL94208 GUI User Guide V2.10

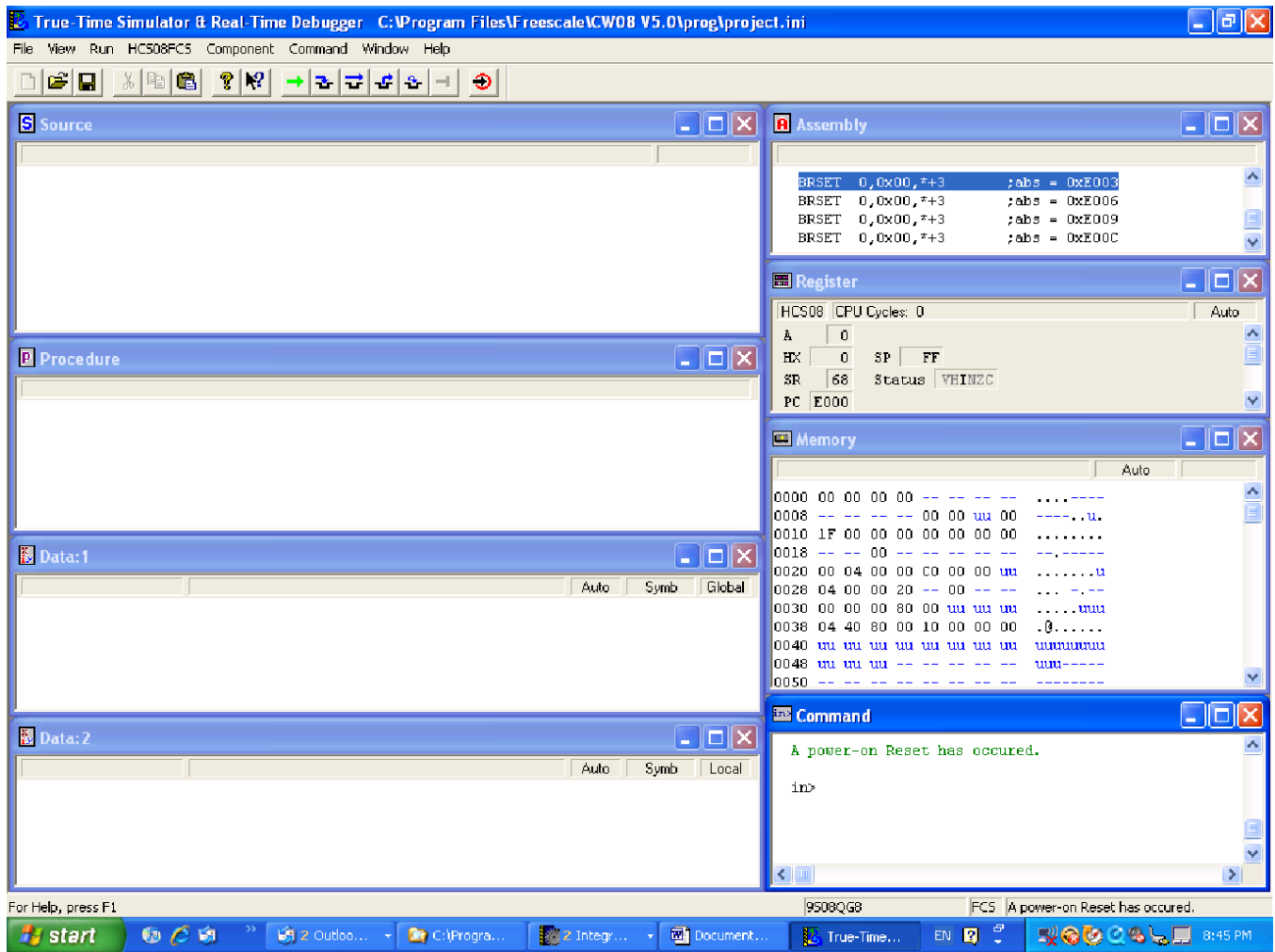
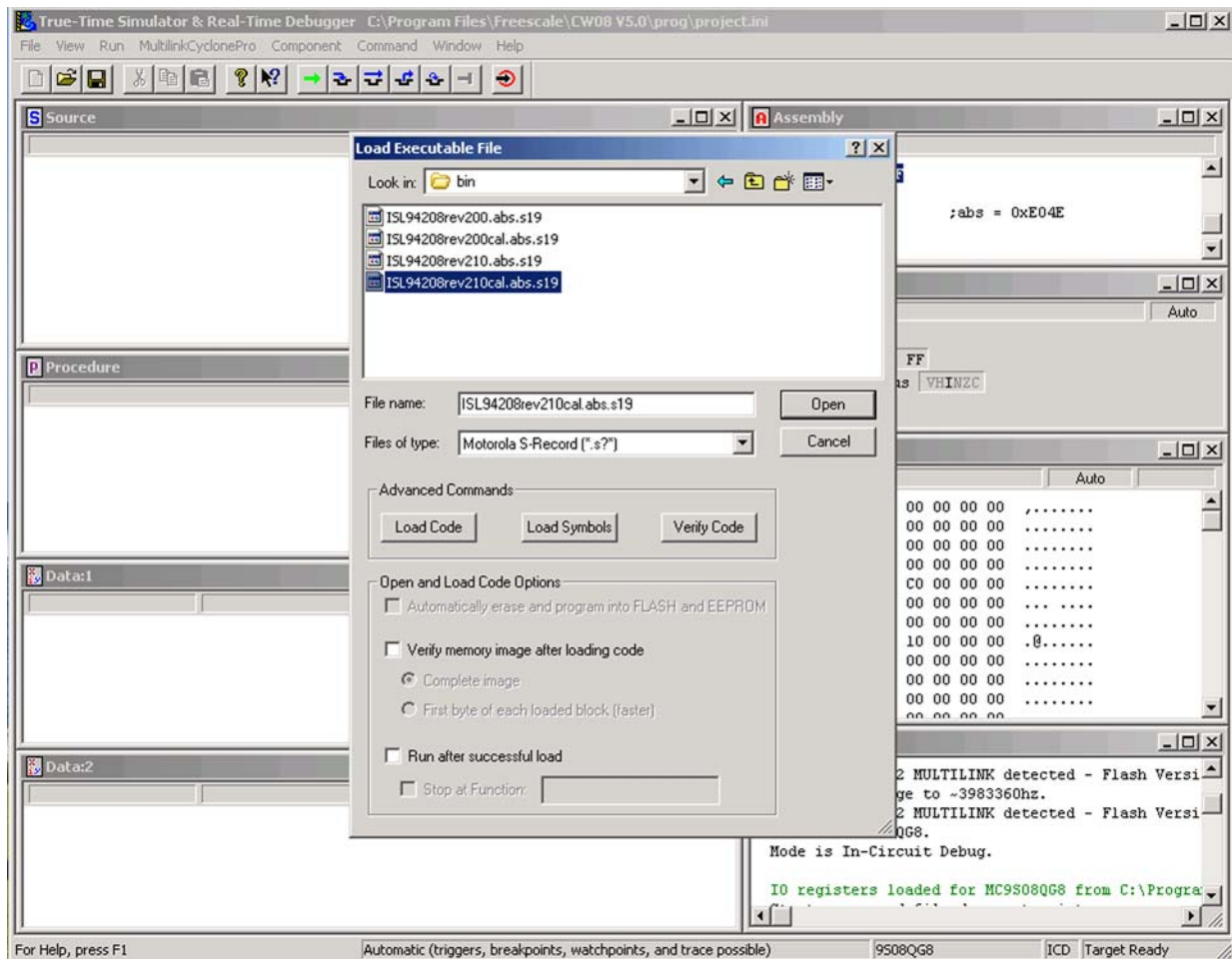


FIGURE 10. CODE WARRIOR SIMULATOR (HIGHWAVE.EXE)





**FIGURE 13. CODE WARRIOR SIMULATOR LOAD APPLICATION**