LTC2986, LTC2986-1 and LTM2985 Demo Software Manual

DESCRIPTION

The LTC®2986 demo software is designed to help configure, program and run the LTC2986 (LTC2986-1) and LTM®2985. It can configure, save the configuration, check for configuration errors, run, output the conversion results into a text file, and create Linduino-ready C code based on the configuration.

The software can be used by itself, or used in conjunction with the DC2508 (LTC2986 demo board, included in DC2531) or DC2618 (LTC2986-1 demo board, included in DC2608) or DC2872 (LTM2985 demo board) or demo circuits. For more information about these circuits, please see
DC2531A
DC2608A
DC2872A

INSTALLING THE PROGRAM

The LTC2986/LTM2985 demo software can be installed either through QuikEval™ or manually.

QuikEval provides a single interface that installs and launches the appropriate demo software based on the hardware connected to the computer. However, QuikEval requires connected demo hardware (DC2508/DC2618/DC2872A) to install or run.

A manual install and/or launch, by contrast, works with or without any of the accompanying demo hardware.

To install through QuikEval, follow the steps below:

1. Obtain the DC2508/DC2618/DC2872A demo circuit, one sensor-specific demo circuit (DC2210, DC2211, DC2212, DC2213 or DC2214), and a DC2026 Linduino demo board.
2. Before connecting the hardware to the computer, install the latest version of the QuikEval software. This can be downloaded here.
3. Connect the hardware together. Connect the DC2508/DC2618/DC2872A to the DC2026 using the supplied 14-conductor ribbon cable. Connect the sensor-specific demo circuit directly to the DC2508/DC2618 demo board through the 40-pin connector. Connect the DC2026 to the computer using a standard USB cable.
4. Launch the QuikEval software. The LTC2986/LTM2985 demonstration software will install and launch automatically.

To launch through QuikEval, repeat steps 3 to 4 above.

To install manually, follow the steps below.

1. Download the demo software files from https://www.analog.com/en/products/ltc2986.html#product-requirement and https://www.analog.com/en/products/ltm2985.html#product-requirement.
2. Right click the file and choose “Extract All.” Choose a suitable location for the extraction folder, and check the “Show extracted files when complete” check box. Click “Extract” to extract the files.
3. In the extracted folder, double-click the .msi file and follow the prompts to install the LTC2986/LTM2985 QuikEval module.
4. In the extracted folder, double-click the file CDM20824_Setup.exe to install the USB drivers. (Note: If you are sure that you have up-to-date FTDI drivers, you can skip this step.)

To launch manually, from the Start menu, select “Analog Devices” → “LTC2986” or “LTM2985.”

Regardless of the installation method, the user can later launch the program manually or, if installed, through QuikEval.

QUICK START PROCEDURE

When the program is started, a screen similar to Figure 1 appears.

There are menus for Configuration, Evaluation, C code generation, and Help. Moving the mouse pointer over the toolbar buttons will show a mouseover box describing the button's function.

There is also a log window, which can give useful information (particularly with unexpected events).

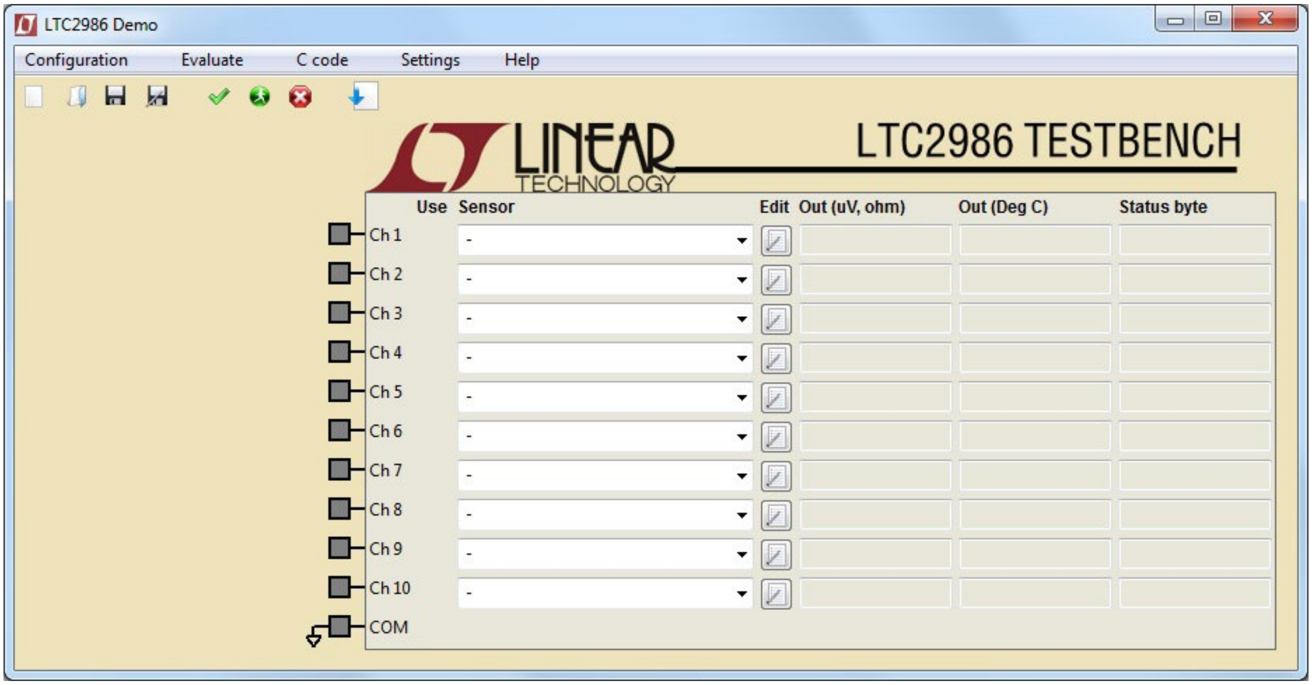


Figure 1. Main Screen

CONFIGURING THE LTC2986/LTM2985

The user can either create a new configuration or load an existing configuration.

Creating a New Configuration

In this example, the LTC2986/LTM2985 will be configured for a thermocouple on Channel 3 with a RTD PT-100 cold junction on Channel 9, using a sense resistor on Channel 7. When this example is finished, the configuration should look like Figure 2.

Additionally, the type K thermocouple on Channel 3 will use the LTC2986/LTM2985's open circuit checking feature, with an open circuit detect current of 100μA. The RTD PT-100 on Channel 9 will use an American standard. It will also use a 4-wire configuration, without rotation or sharing, and will use 100μA excitation current. The sense resistor for this specific daughter board was measured precisely at 1999.1Ω.

To set up this configuration, first add a Type K thermocouple on Channel 3. Click on the Sensor box on Channel 3, and choose the Type K thermocouple, shown in Figure 3.

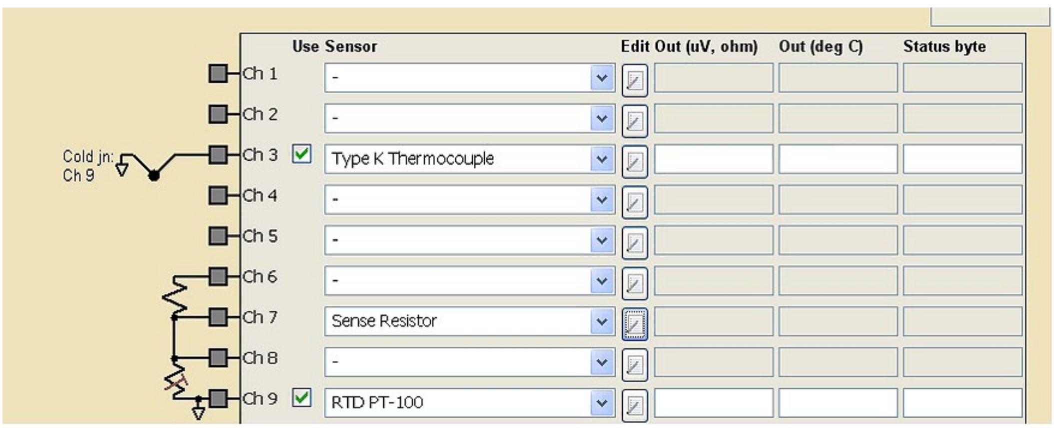


Figure 2. Desired Configuration, After Setup, with a Thermocouple and RTD Cold Junction

CONFIGURING THE LTC2986/LTM2985

After selecting the thermocouple, the demo software will draw a thermocouple on Channel 3, shown in Figure 4. This shows how the thermocouple should be wired to the LTC2986/LTM2985 (for thermocouples, protection resistors and filter capacitors can also be added). There is also a checkbox under the "Use" column. If it is checked, the demo software will make a measurement on this channel while it is running.

To remove this device, select the "-" choice (top selection in Figure 3).

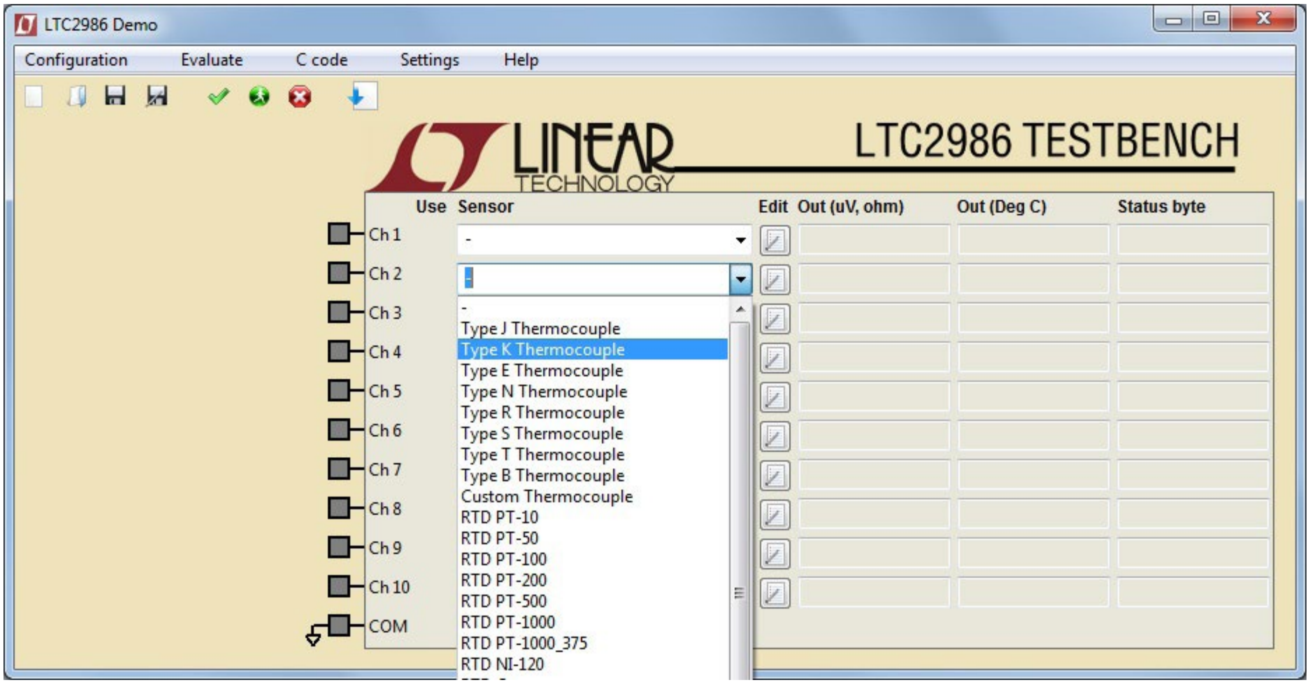


Figure 3. Selecting the Type K Thermocouple

Figure 4. Display After Thermocouple Is Selected

CONFIGURING THE LTC2986/LTM2985

To configure the thermocouple, click the corresponding "Edit" button to the right on Channel 3. The dialog in Figure 5 will appear. Change the choices to match Figure 5.

In this case, the cold junction sensor is on Channel 9, the thermocouple measurement is single-ended, open-circuit detection is generated by the LTC2986/LTM2985, and the open detect current is 100µA.

When satisfied with the changes, click "Accept Changes."

Note that the channel assignment data, shown at the bottom of the window, has updated. This channel assignment is the 32-bit word that is sent into the memory location for the given channel in order to configure it. This is shown in Figure 6.

Close this dialog box.

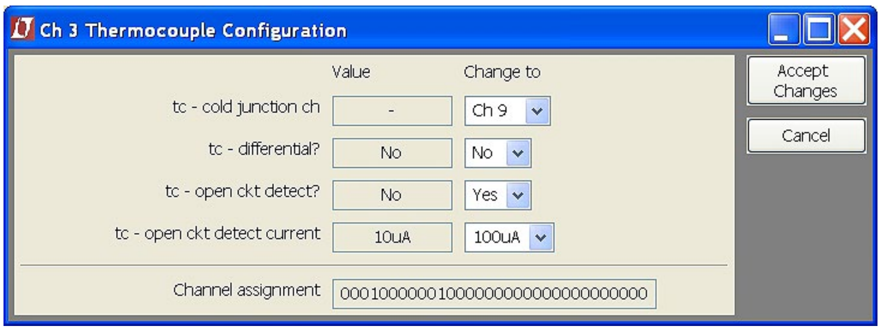


Figure 5. Changing the Thermocouple Configuration

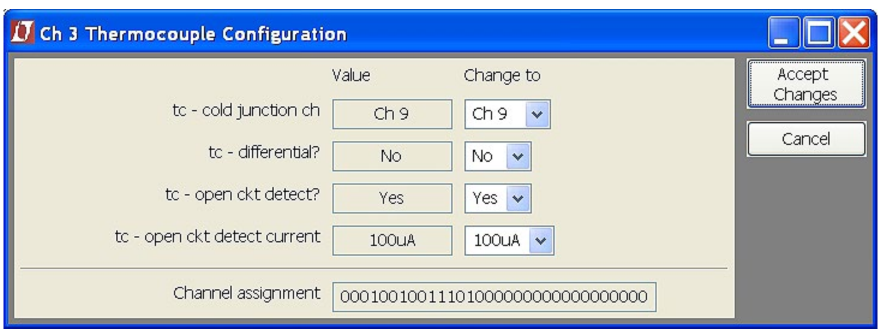


Figure 6. Configuration Screen After "Accept Changes" Button Is Pressed

CONFIGURING THE LTC2986/LTM2985

Next, add a cold junction sensor on Channel 9. Similar to how the thermocouple was selected, go to the Sensor box on Channel 9, and select the RTD PT-100. The screen should now look like Figure 7.

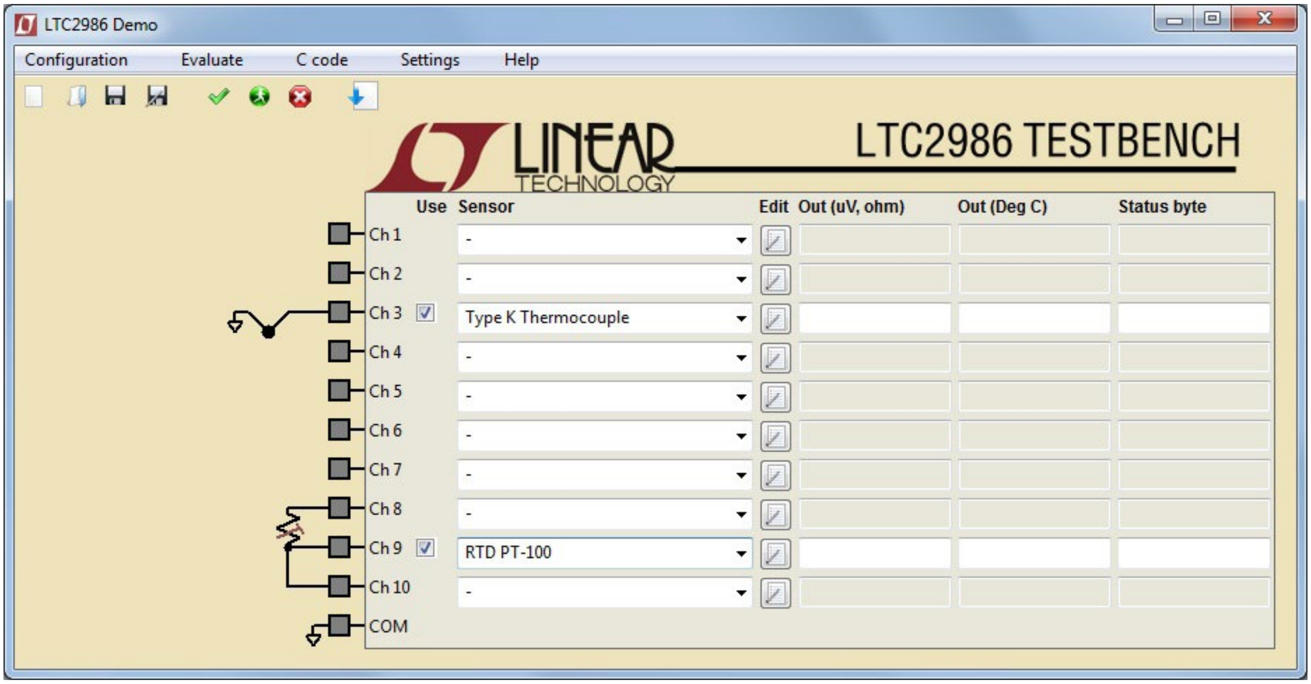


Figure 7. Software After RTD Is Selected on Channel 9

CONFIGURING THE LTC2986/LTM2985

To configure the RTD, click the Edit button on Channel 9, and fill the form out as shown in Figure 8. Then click Accept Changes.

Close this dialog box.

The main screen should match Figure 9. The RTD is now grounded on Channel 9 (which happens when there is no rotation or sharing).

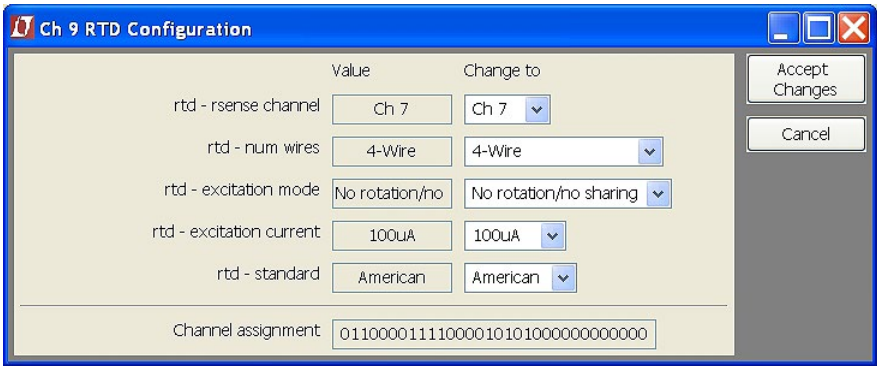


Figure 8. Configuring the RTD on Channel 9

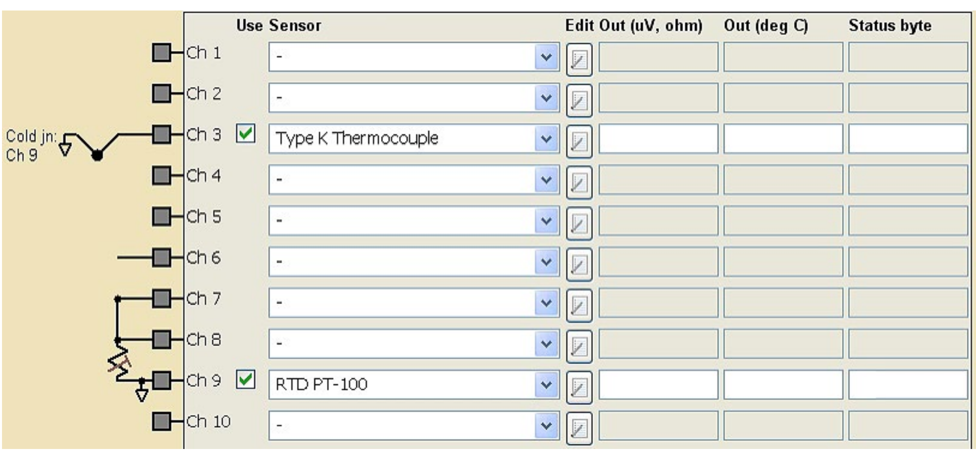


Figure 9. Main Screen After RTD Configuration Was Changed

CONFIGURING THE LTC2986/LTM2985

Setting Global Parameters

As mentioned in the LTC2986/LTM2985 data sheet, the temperature unit, rejection frequency, RTD and thermistor Kelvin mode settings, and delay between individual conversions are adjustable. To change these parameters, in the menu bar, go to Configuration → Set global parameters. The dialog box in Figure 10 should appear.

The default values are fine here, so close this box.

Checking The Configuration

To check the configuration, either go to Evaluate → Check configuration in the menu bar, or press the check mark button in the toolbar. The demo software then looks like Figure 11:

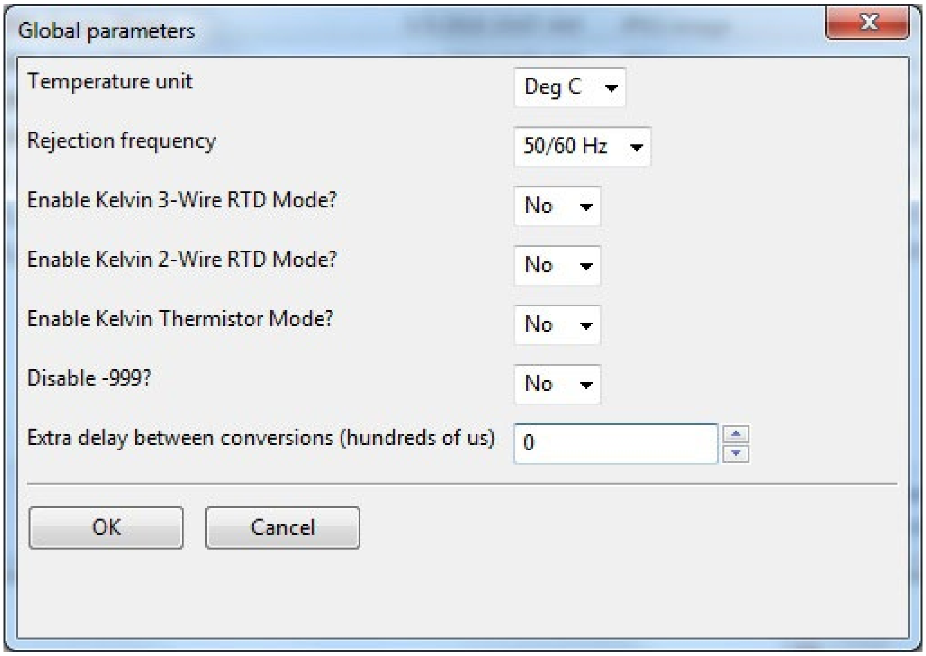


Figure 10. Configuring Global Parameters

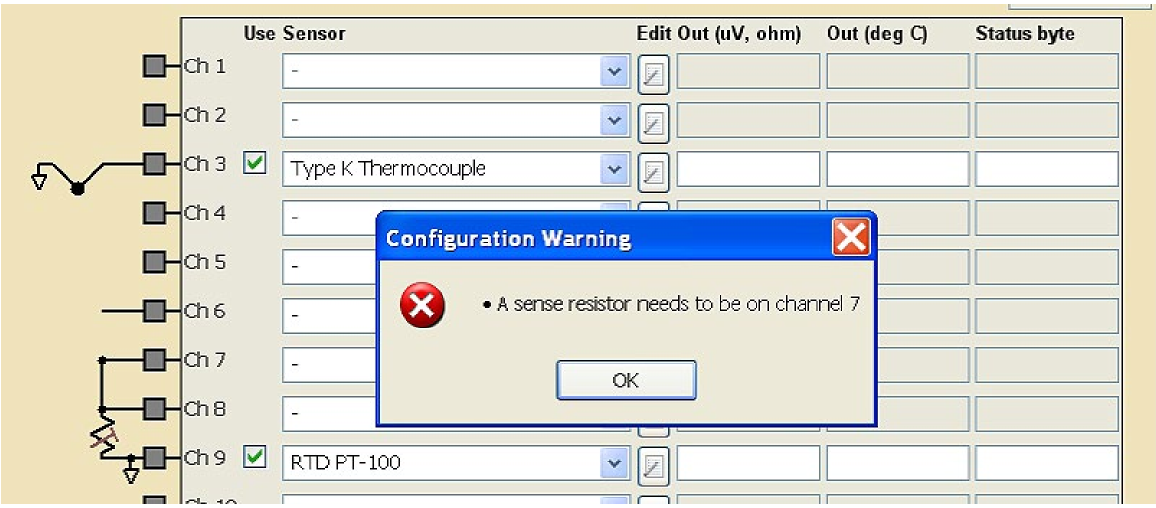


Figure 11. Main Screen After Running the Configuration Checker

CONFIGURING THE LTC2986/LTM2985

The demo software includes an extensive configuration checker, which in this case shows that a sense resistor was never added on Channel 7. To add the sense resistor, select the sense resistor from the Sensor box. Click the edit button and enter the value of the sense resistor (1999.1 in this example). The resulting dialog box should look like Figure 12.

The software rounded 1999.1 to the closest value that can be represented in the LTC2986/LTM2985's 27-bit format.

Close this dialog and check the configuration again – there should be a message that no errors were found.

Save Configuration

To save this configuration, go to Configuration → Save, or press the save button, and pick a name for the file (demo_manual1.cfg). The configuration can now be opened from the menu by selecting configuration → open.

The configuration should look as shown in Figure 13.

Every channel with a device is checked, except for the sense resistor. Every sensor that is checked will be measured, but since the sense resistor is part of the RTD measurement and is not measured separately, there is no option to check it.

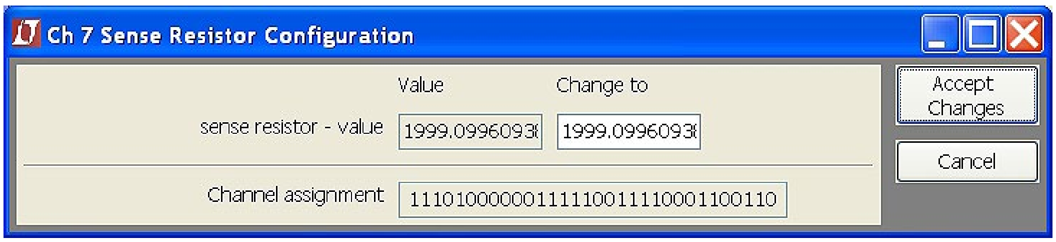


Figure 12. Changing the Sense Resistance

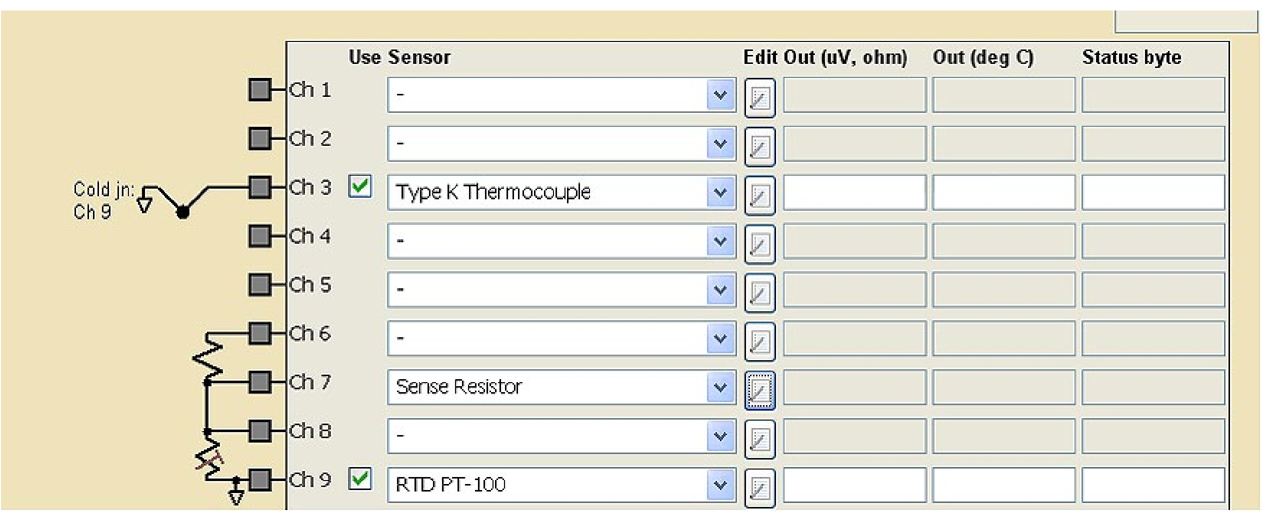


Figure 13. Final Configuration

CONFIGURING THE LTC2986/LTM2985

Load an Existing Configuration

A number of example configurations, from both the data sheet and from the demo boards, are also available. For example, to load the configuration for the DC2212 demo board, from the menubar select Configuration → Load Example → Demo Board → DC2212_THERMOCOUPLE_BOARD. The circuit in Figure 14 will load.

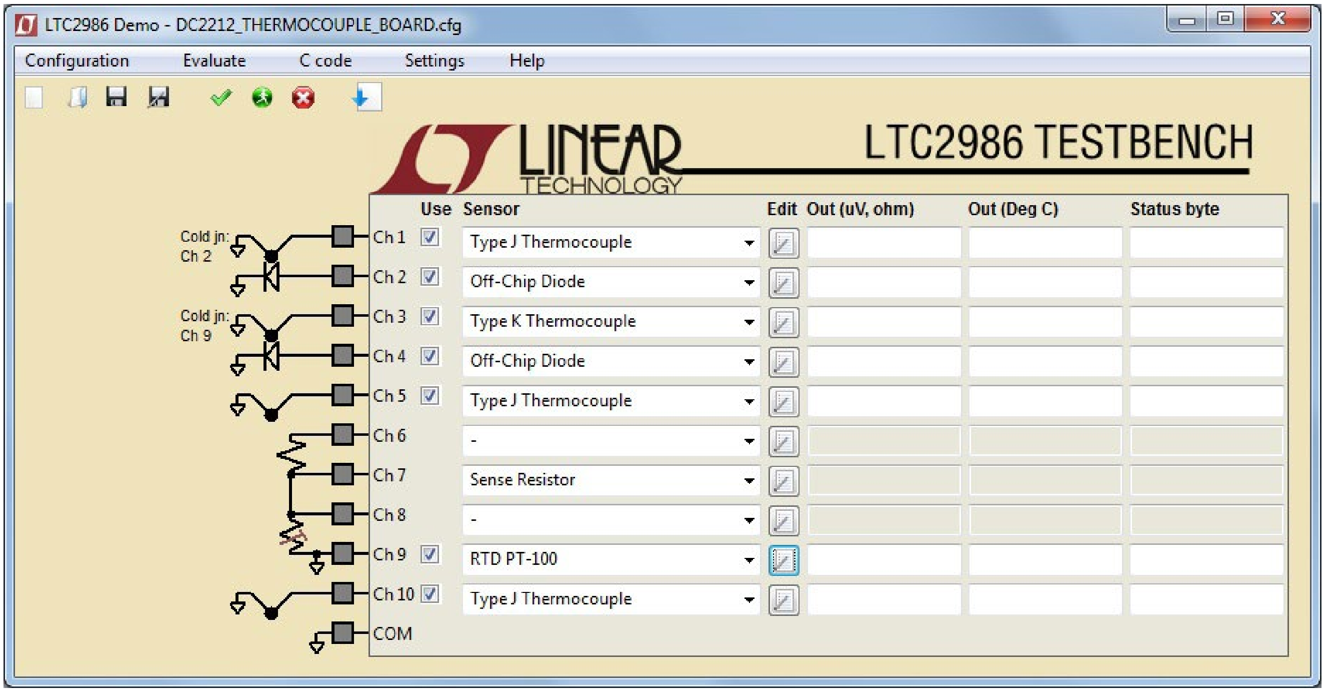


Figure 14. Opening DC2212_THERMOCOUPLE_BOARD.cfg

CONFIGURING THE LTC2986

Loading The Configuration from a Daughter Board

The demo software has a corresponding configuration for each daughter board. Figure 15 shows such a setup.

The DC2211 Universal Temperature Measurement board is shown at the top right of Figure 15. It includes a sense resistor and a diode. The measured sense resistance was stored on an EEPROM chip when the DC2211 was tested.

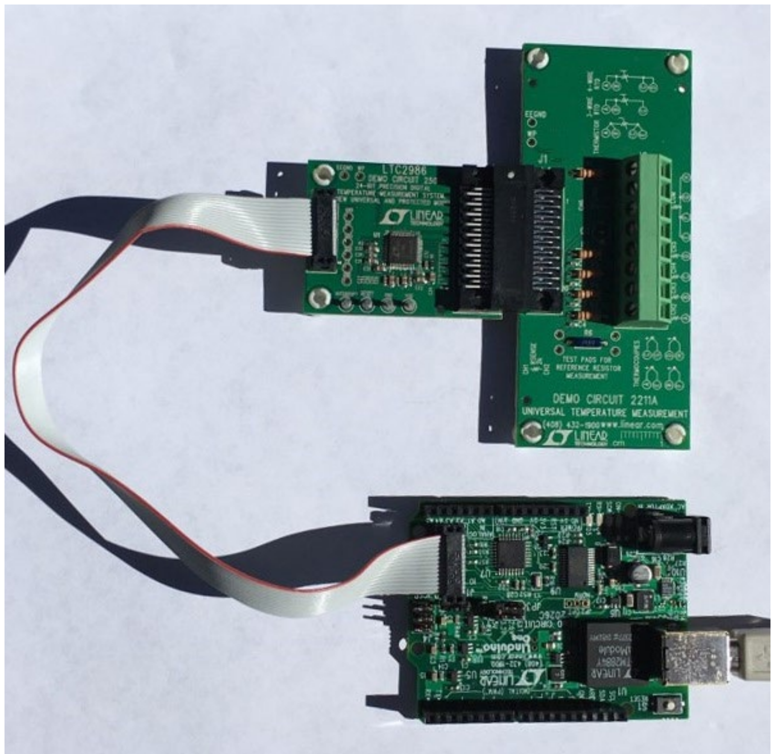


Figure 15. Setup with DC2211 Daughter Board

Note: This section on Loading the Configuration from a Daughter Board only applies to the LTC2986 and LTC2986-1.

CONFIGURING THE LTC2986

To load the configuration from the daughter board, do the following:

1) Plug the setup, which should look similar to what is shown in Figure 15, into a computer.

2) In the Configuration menu, select "Load from daughter board."

The software will search for the daughter board, find the appropriate configuration, and load the device values from the EEPROM. The results of loading the DC2211 daughter board are shown in Figure 16.

Click on the Ch 2 Edit button to see the loaded value for the sense resistor.

Refer to the DC2531 and DC2608 demonstration manuals for details on interfacing the DC2211 to Thermistors, RTDs, and Thermocouples.

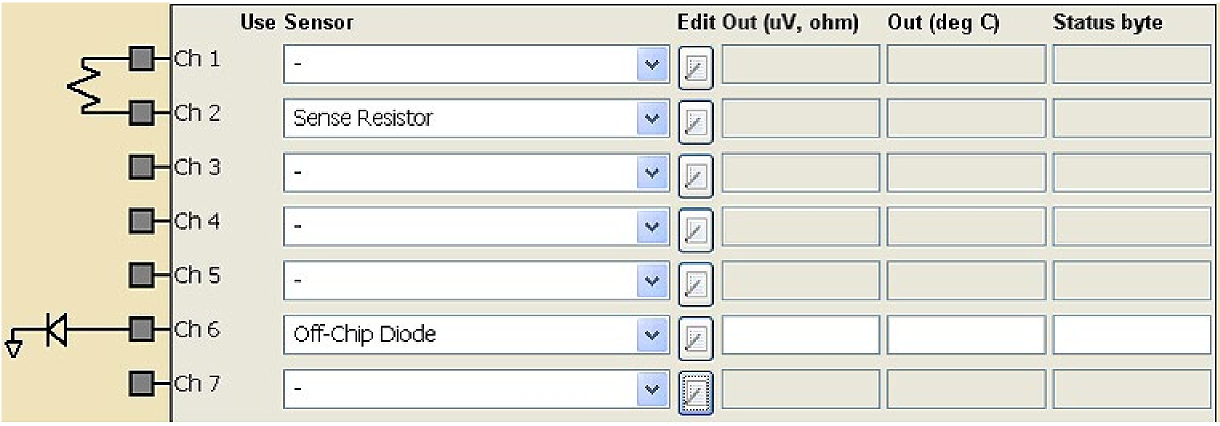


Figure 16. Main Screen After Loading from Daughter Board

RUNNING THE TESTBENCH

To run the program from the menu bar, go to Evaluate → Run, or press the run button in the toolbar. An example output for a given configuration is shown in Figure 17.

The LTC2986/LTM2985 demo software continuously scans and measures the sensors from top to bottom. The sensor just measured is highlighted. To turn off a sensor, uncheck the corresponding "use" check box.

Seeing Output Errors

In order to demonstrate the fault reporting capabilities of the LTC2986/LTM2985, the DC2212 thermocouple board is used. Its configuration automatically loaded using the "Load from Daughter Board" command in the configuration menu. The thermocouple on Channel 1 was removed from the board. The channel with an error is then shown in red, as shown in Figure 18. By hovering the mouse over the status byte, the user can see the corresponding errors, as shown in the figure.

To stop the run, either go to Evaluate → Stop, or press the red "x" button on the toolbar. The program will scan down to the last sensor and then stop.

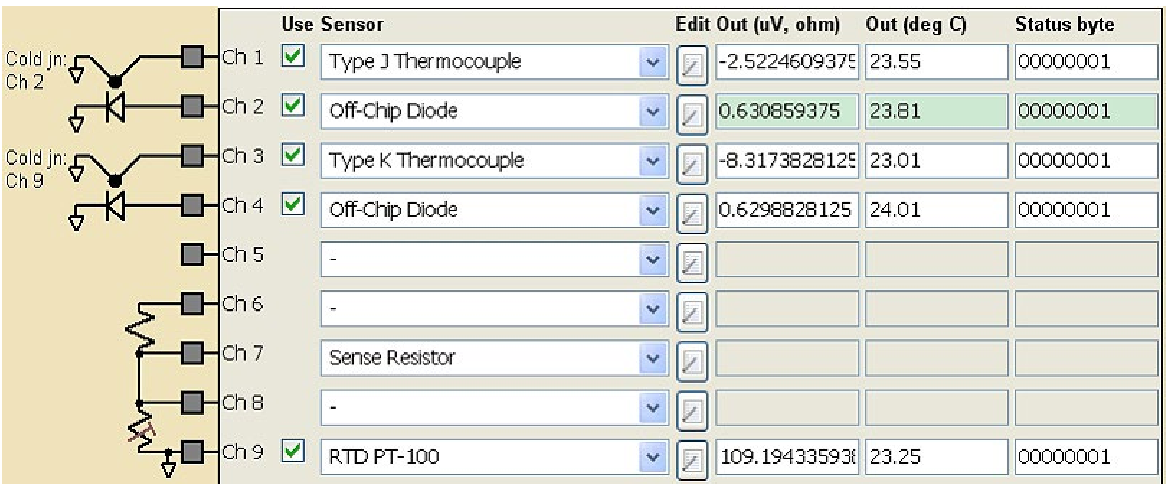


Figure 17. Main Screen When Running the Program

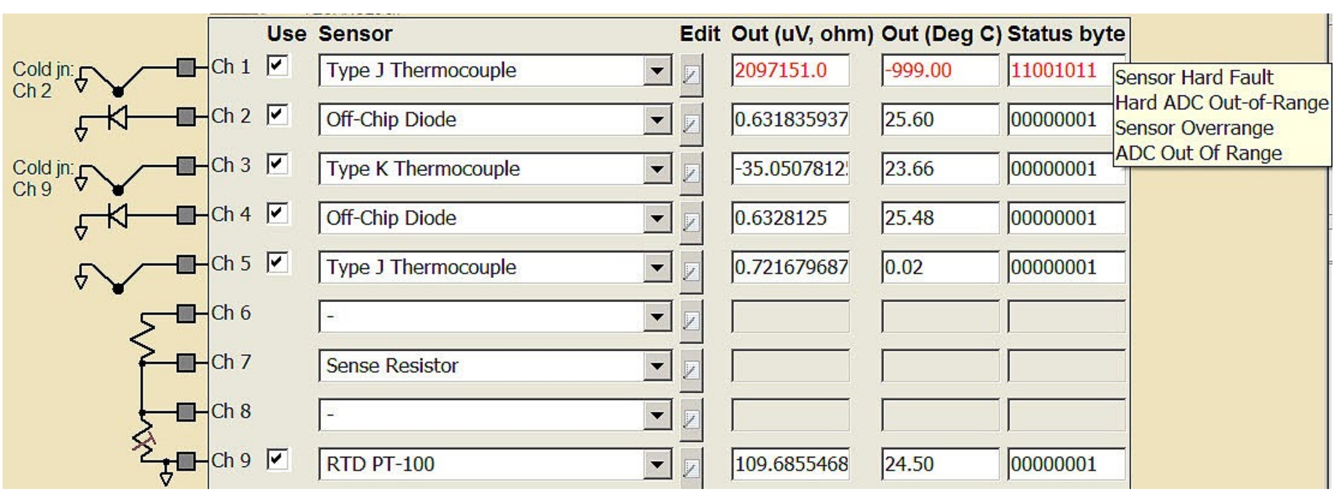


Figure 18. Main Screen After the Thermocouple on Channel 1 Is Disconnected. While Not Shown in the Screenshot, the Mouse Is Hovering Over the Channel 1 Status Byte

RUNNING THE TESTBENCH

Storing the Output

Data can be logged from the TestBench by selecting Evaluate → Run and save output:

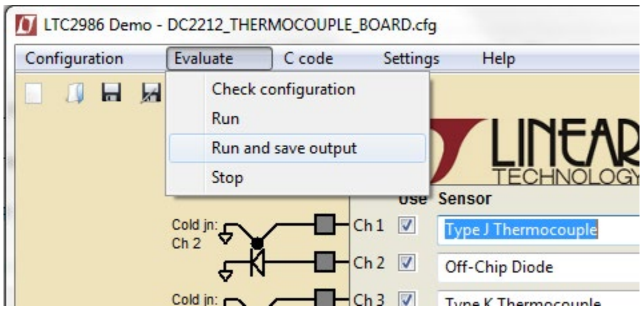


Figure 19.

Once selected, the software will ask where to store the results (Figure 20).

The output filename can also be set here.

After stopping the run, the output file contains all the results from the run, including the measured voltages or resistances, temperatures, and status codes. The file is comma separated, which means the data can be imported into Excel. A part of this output is shown in Figure 21.

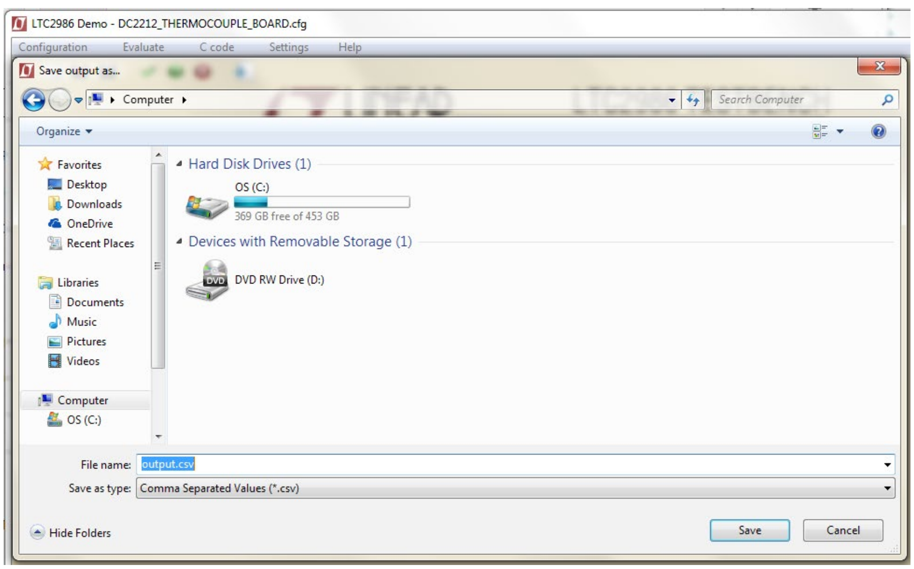
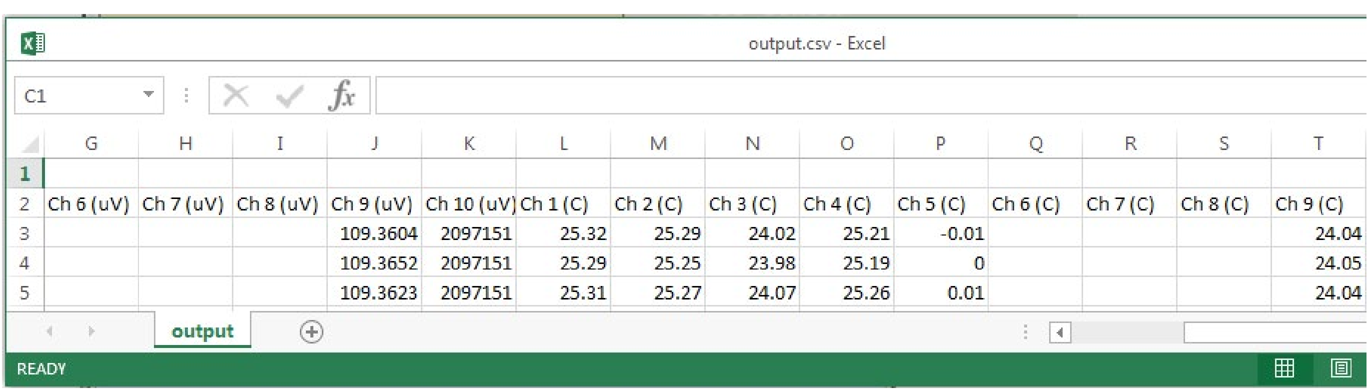


Figure 20.



	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1														
2	Ch 6 (uV)	Ch 7 (uV)	Ch 8 (uV)	Ch 9 (uV)	Ch 10 (uV)	Ch 1 (C)	Ch 2 (C)	Ch 3 (C)	Ch 4 (C)	Ch 5 (C)	Ch 6 (C)	Ch 7 (C)	Ch 8 (C)	Ch 9 (C)
3				109.3604	2097151	25.32	25.29	24.02	25.21	-0.01				24.04
4				109.3652	2097151	25.29	25.25	23.98	25.19	0				24.05
5				109.3623	2097151	25.31	25.27	24.07	25.26	0.01				24.04

Figure 21. One Part of the output.csv File (Copied Into Excel)

CREATING C CODE

The LTC2986/LTM2985 demo software can create C code for the configuration in the main screen. This code can be directly loaded into a Linduino (download here) and executed.

To see an example, go to Configuration → Open, navigate to example_config_files, and select DC2211_universal_temperature_measurement_daughter_board.cfg. Then, from the menu bar, go to C Code → Create C Code and select an appropriate folder. There should be a message similar to that shown in Figure 22:

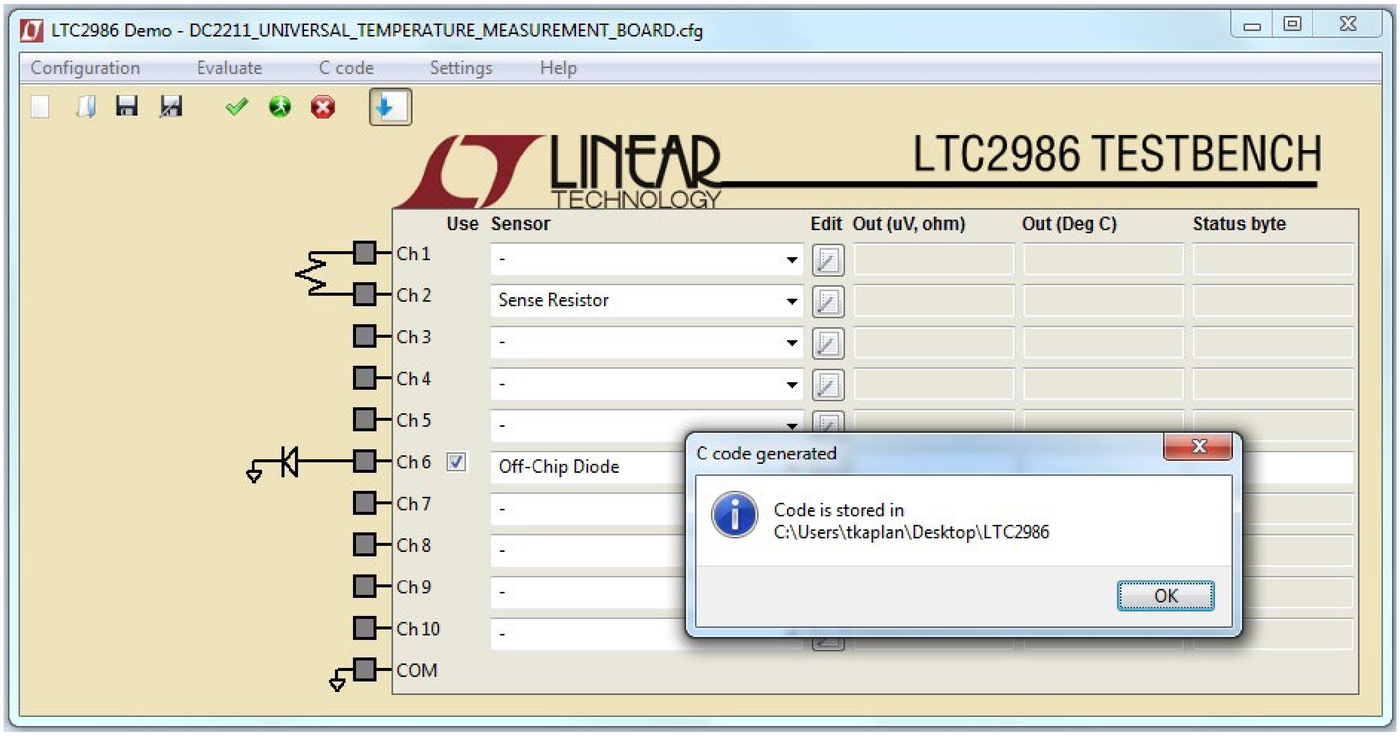
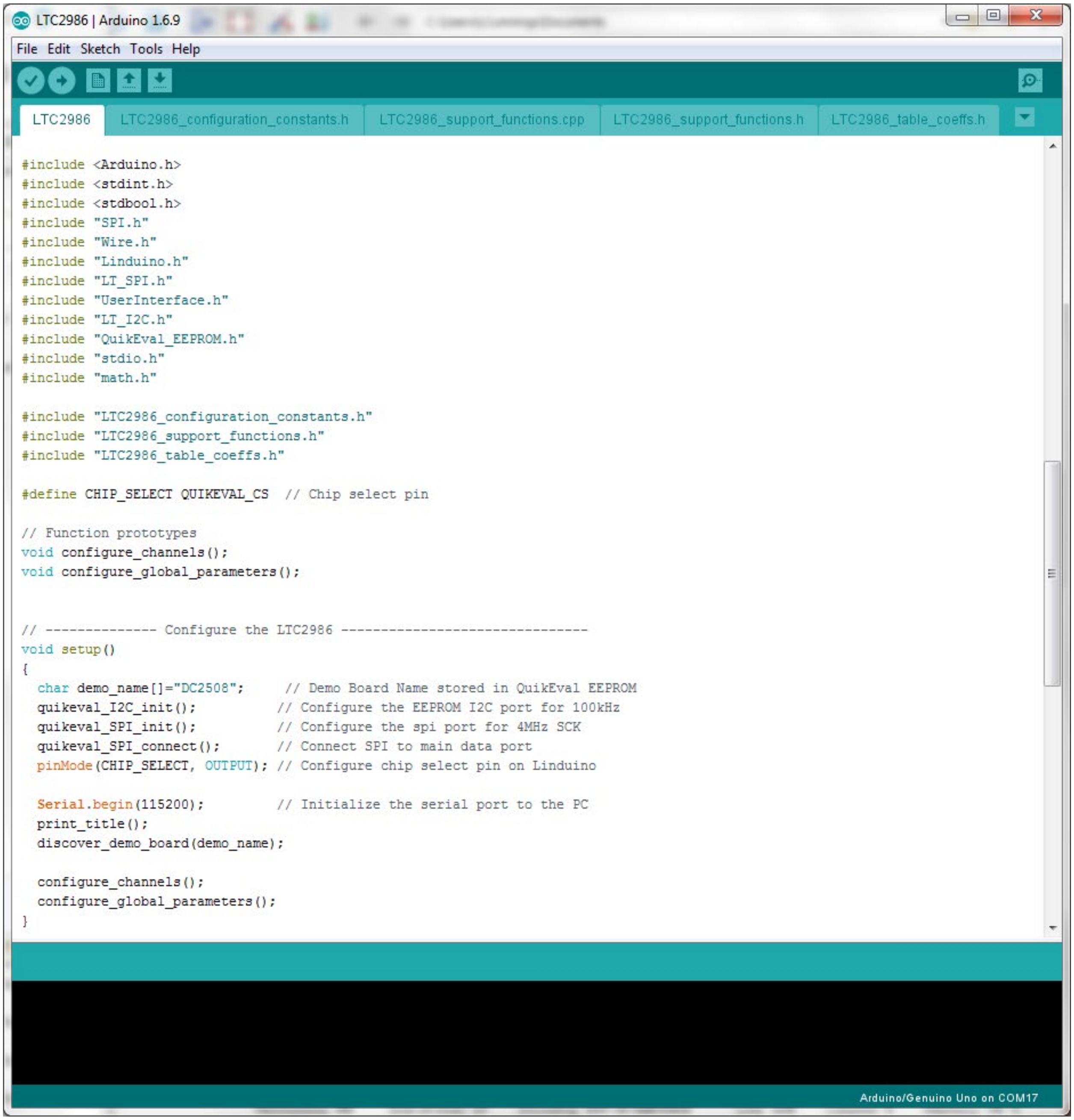


Figure 22. Main Screen After the "Generate C Code" Toolbar Button Was Pressed

CREATING C CODE

Navigate to the folder selected, and go into the "LTC2986" or "LTM2985" folder in order to see a file with a .ino extension, as well as support functions and configuration constants. Sample screenshots are shown in Figure 23a and Figure 23b.




```
#include <Arduino.h>
#include <stdint.h>
#include <stdbool.h>
#include "SPI.h"
#include "Wire.h"
#include "Linduino.h"
#include "LT_SPI.h"
#include "UserInterface.h"
#include "LT_I2C.h"
#include "QuikEval_EEPROM.h"
#include "stdio.h"
#include "math.h"

#include "LTC2986_configuration_constants.h"
#include "LTC2986_support_functions.h"
#include "LTC2986_table_coeffs.h"

#define CHIP_SELECT QUIKEVAL_CS  // Chip select pin

// Function prototypes
void configure_channels();
void configure_global_parameters();


// -------------- Configure the LTC2986 -------------------------------
void setup()
{
  char demo_name[]="DC2508";     // Demo Board Name stored in QuikEval EEPROM
  quikeval_I2C_init();          // Configure the EEPROM I2C port for 100kHz
  quikeval_SPI_init();          // Configure the spi port for 4MHz SCK
  quikeval_SPI_connect();       // Connect SPI to main data port
  pinMode(CHIP_SELECT, OUTPUT); // Configure chip select pin on Linduino

  Serial.begin(115200);         // Initialize the serial port to the PC
  print_title();
  discover_demo_board(demo_name);

  configure_channels();
  configure_global_parameters();
}
```

Figure 23a. Top Part of Generated C Code

CREATING C CODE

```
void configure_channels()
{
  uint8_t channel_number;
  uint32_t channel_assignment_data;

  // ----- Channel 2: Assign Sense Resistor -----
  channel_assignment_data =
    SENSOR_TYPE__SENSE_RESISTOR |
    (uint32_t) 0x1F4000 << SENSE_RESISTOR_VALUE_LSB;    // sense resistor - value: 2000.
  assign_channel(CHIP_SELECT, 2, channel_assignment_data);
  // ----- Channel 6: Assign Off-Chip Diode -----
  channel_assignment_data =
    SENSOR_TYPE__OFF_CHIP_DIODE |
    DIODE_SINGLE_ENDED |
    DIODE_NUM_READINGS__2 |
    DIODE_AVERAGING_OFF |
    DIODE_CURRENT__20UA_80UA_160UA |
    (uint32_t) 0x100C49 << DIODE_IDEALITY_FACTOR_LSB;   // diode - ideality factor(eta): 1.00299930572509765625
  assign_channel(CHIP_SELECT, 6, channel_assignment_data);

}



void configure_global_parameters()
{
  // -- Set global parameters
  transfer_byte(CHIP_SELECT, WRITE_TO_RAM, 0xF0, TEMP_UNIT__C |
    REJECTION__50_60_HZ);
  // -- Set any extra delay between conversions (in this case, 0*100us)
  transfer_byte(CHIP_SELECT, WRITE_TO_RAM, 0xFF, 0);
}

// -------------- Run the LTC2986 -------------------------------------

void loop()
{
  measure_channel(CHIP_SELECT, 6, TEMPERATURE);      // Ch 6: Off-Chip Diode
}
```

Figure 23b. Bottom Part of Generated C Code

CREATING C CODE

Even without a Linduino, one can get a good idea how to program the LTC2986/LTM2985 by examining the generated C code.

C-code for all demonstration boards and each example shown in the data sheet are included in the Linduino sketchbook.

However, the file should be ready to load and run on a Linduino. For further information on how to set up a Linduino, download here. Once user generated C-code is uploaded into the Linduino, the demonstration software will no longer run. In order to run the demonstration software, the default DC590 code needs to be reloaded into the Linduino board. This is accomplished by opening the Arduino IDE, selecting from the sketchbook Utilities → DC590B, and uploading to the Linduino (see Figure 25).

Figure 24. Warning to Load Original Linduino Code Back in

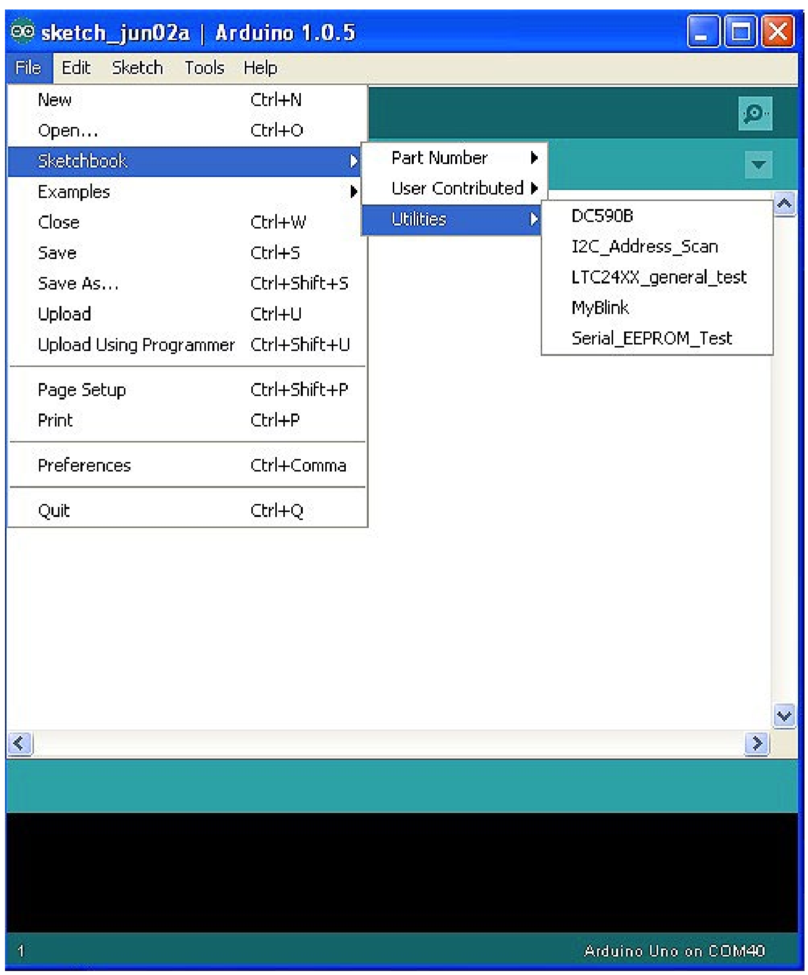


Figure 25. Reloading DC590 Back Into the Linduino

ERROR ESTIMATION

Noise and errors are an unavoidable part of measuring sensors. While measuring a RTD temperature with the LTC2986/LTM2985 family, for example, errors can come from many sources—from the temperature coefficient (tempco) of the sense resistor to the INL of an ADC inside the LTC2986/LTM2985.

It can be difficult to derive system-level error and noise estimations based on these error sources. The Error Estimator, inside the demo software, helps model these errors.

To see an example, perform the following:

1) Open the Demo Software

2) Open the configuration for the 2212 thermocouple board. To do this, go to the configuration menu and do the following:

   Configuration → Load example → Demo board → DC2212_THERMOCOUPLE_BOARD.cfg.

   The main screen should look like Figure 26:

3) Edit Channel 9 (The RTD PT-100 thermocouple).

4) Click the "Estimate Errors" button on the right side.

   A window (Figure 27) appears. Mouse over the graphs to see the error contributions of the different error sources.

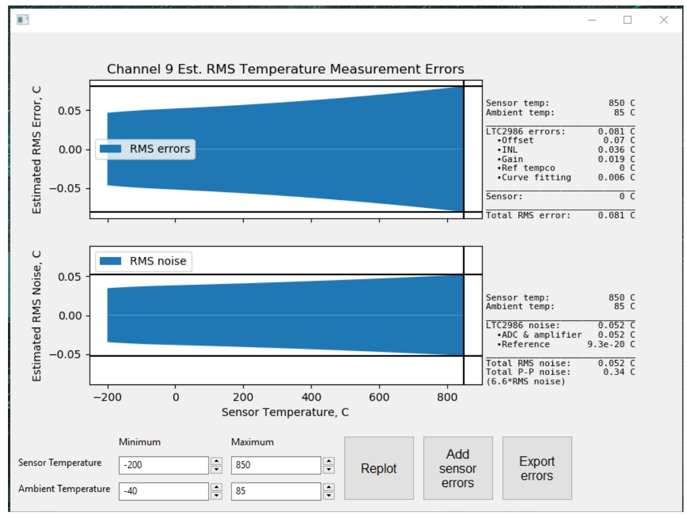


Figure 27. Error Estimation Diagram

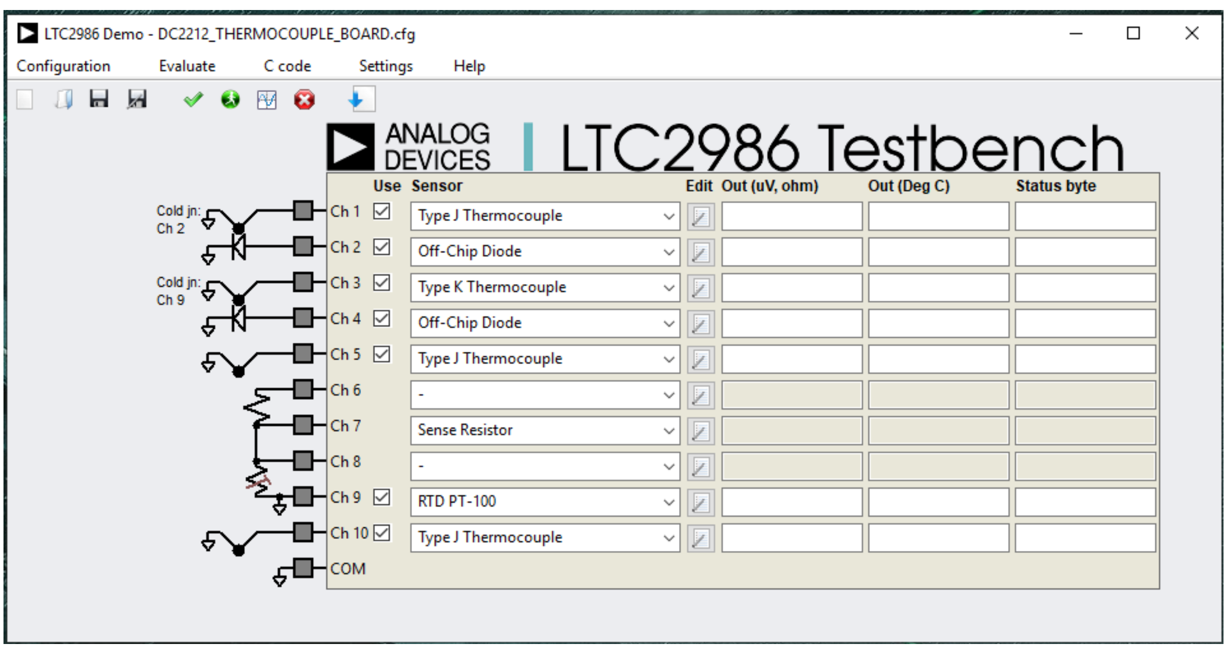


Figure 26. DC2212 Thermocouple Board Configuration

ERROR ESTIMATION

Error Estimation Detail

To better understand the information in Figure 27, it is useful to know how the error is estimated.

The top graph in Figure 27 shows the estimated RMS error. To generate this graph, the errors are estimated as follows:

1) For each error source that has a data sheet spec (for example, the INL), the worst-case data sheet spec is used.
2) For each error source without a data sheet spec (for example, current source noise), estimates were informed by measuring production parts.
3) External components are assumed to contribute zero errors until the user supplies sensor errors. This will be discussed in the next section.
4) The error sources are assumed to be uncorrelated and are added using RMS to generate the total error.
5) Worst-case ambient temperatures are used to generate these graphs. The user can change the ambient temperature range.

The bottom graph in Figure 27 shows the RMS noise. To create this estimate, only ADC noise and current source noise, where applicable, are considered. The corresponding peak-peak noise is shown on the right.

Mouse over the graph to view more detail on the right for specific temperatures.

The top item, "Sensor temp", is the temperature of the main sensor. For this measurement, it is the RTD temperature. For a thermocouple measurement, it is the thermocouple temperature. The sensor temperature range can be set by the text controls at the bottom of the figure.

The next item, "Ambient temp", is the temperature of the ADC and all other external components. In the case of an RTD measurement, the ADC and sense resistor are at the ambient temperature. For a thermocouple measurement with cold junction compensation, the entire cold junction and ADC are assumed to be at the ambient temperature.

To compute the two graphs, errors are calculated at the extremes of the ambient temperatures (–40°C and 80°C here), and the worst-case errors are plotted. The noise graph is similarly calculated. In this example, an ambient temperature of 85°C generated both the worst-case noise and worst-case errors across the sensor temperature range.

Below the sensor and ambient temperatures, the display shows the LTC2986/LTM2985 errors. The LTC2986/LTM2985 offset, INL, gain, and the tempco of the reference all can contribute errors. However, because the RTD measurement is ratiometric, reference inaccuracies cancel out and do not contribute to the total error. The error estimator assumes no sensor errors by default; the next section shows how to add sensor errors.

The sensor temperature and ambient temperature ranges are changeable at the bottom of the window. For example, let us assume the ambient temperature will only range from 20°C to 30°C. In the bottom of the window, change the minimum ambient temperature from –40°C to 20°C, and change the maximum ambient temperature from 85°C to 30°C. Click the "Replot" button. The display should look like Figure 30.

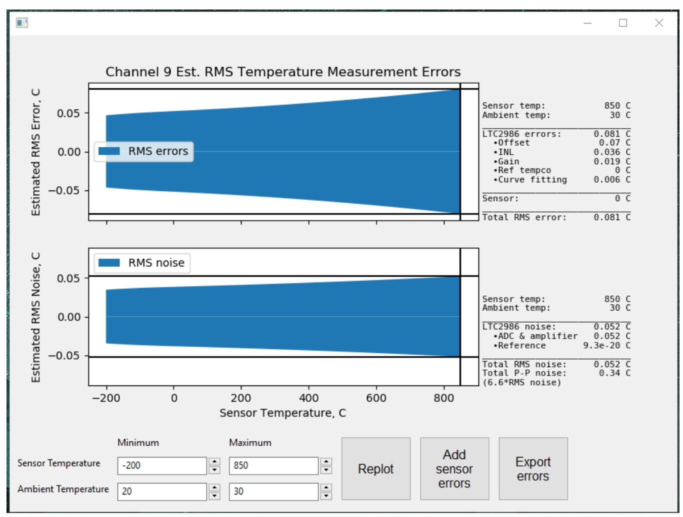


Figure 28. Estimated Measurement Errors with A Narrower Ambient Temperature Range

Adding Sensor Errors

The user can add error specifications to sensors and sense resistors. To add these errors, click "Add sensor errors" on the bottom. A worksheet for adding sensor errors appears (Figure 29). All relevant sensor and sense resistor errors can be added here.

ERROR ESTIMATION

The first table shows a temperature-error table for the RTD on Channel 9. Enter the following information as shown in Figure 29.

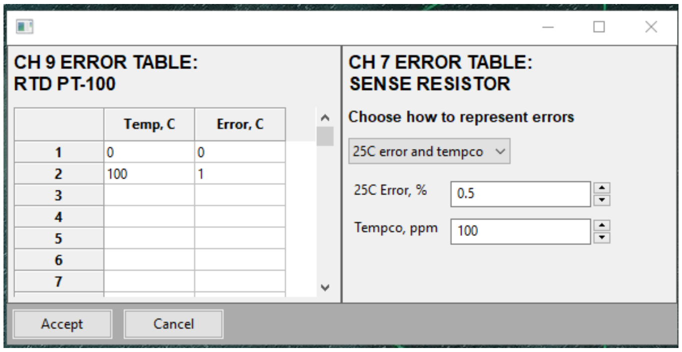


Figure 29. Sensor Error Table

The error table for Channel 9 in Figure 29 shows an error of 0°C at 0°C and an error of 1°C at 100°C. The demo software linearly interpolates between the points, and extrapolates beyond these points using the slope at the boundary. The demo software will then interpolate the error to 0.5°C at 50°C and will extrapolate the error to 2°C at 200°C.

Sense resistor specs are oftentimes given as a maximum error or tolerance at 25°C and tempco in ppm. Enter an error of 0.5% at 25°C and a tempco of 100ppm, as shown in Figure 29. To enter a more complex error table, simply change the selection on top from “25°C error and tempco” to “full error table”. Note that the error table for the sense resistor lists errors in percent, not degrees C as they are for other sensors.

Click “Accept” at the bottom to enter this information. The calculated errors now include RTD and sense resistor contributions. The errors are replotted and the new errors are shown in Figure 30.

The errors for each channel will propagate—in other words, the same sense resistor errors will be used for every measurement that utilizes the Channel 7 sense resistor. As an example, Channel 3 has a thermocouple with the RTD on Channel 9 as its cold junction. Because the thermocouple measurement accuracy depends on the cold junction measurement accuracy, errors in the RTD and sense resistor should be reflected in the thermocouple measurement errors.

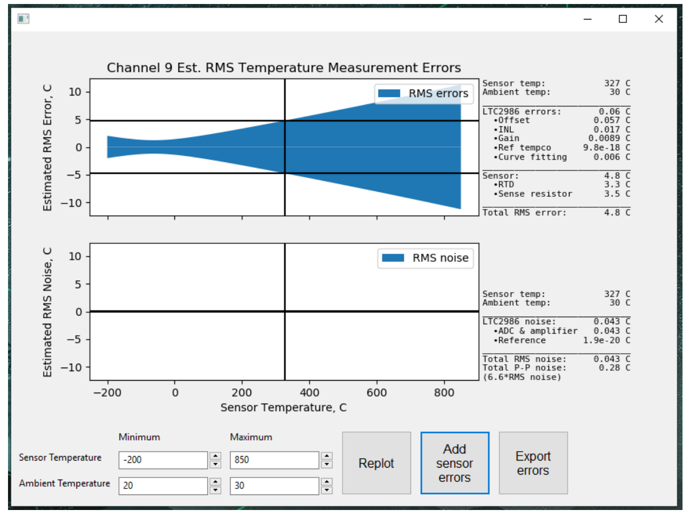


Figure 30. Measurement Errors with User-Supplied RTD and Sense Resistor Errors

To test this hypothesis, generate the estimated errors for Channel 3, using the same procedure used to generate the Channel 9 error estimates. The graph in Figure 31 should appear:

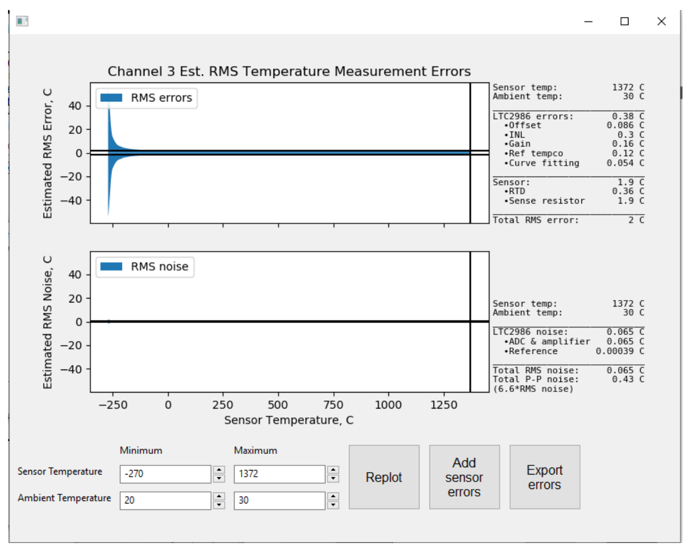


Figure 31. Estimated Errors for Channel 3 Thermocouple

As expected, resistor errors on Channel 7 and RTD errors on Channel 9 contribute to, and in fact dominate, the total estimated error.

ERROR ESTIMATION

It is also notable that there is now an error contribution from the reference tempco. This contribution occurs because a thermocouple creates a voltage, and the accuracy of measuring this voltage depends on the accuracy of the LTC2986/LTM2985's reference voltage.

Exporting and Saving Error Estimates

The estimated errors can be exported into a .csv file, To do so, click "Export errors". To save these errors within the configuration, click Configuration → Save. The sensor errors are then stored with the configuration and will be reloaded the next time the configuration is opened.

CUSTOM SENSORS

Loading In Custom Coefficients and Tables

The LTC2986/LTM2985 demo software allows custom coefficients for thermocouples, RTDs and thermistors. The basic method is shown below: Create a new configuration on the menu by choosing Configuration → New. Select a custom thermocouple on Channel 7, and click the Edit button on Channel 7. The form in Figure 33 will open.

The thermocouple form now has three new entries: The custom address, the custom length (number of coefficients -1) and the custom values. The custom address is where the user wishes to store the thermocouple table, the length is the length of the data, and the values open a text editor to input the custom values (Figure 32).

The data has comma-separated x, y values. The x unit is mV (for thermocouples) and Ohms (for RTDs and thermistors). The y unit is always kelvin. The x values must be monotonically increasing.

The first table entry for thermocouples is the mV value corresponding to 0 kelvin. The first table entry for RTDs and thermistors is the temperature corresponding to 0Ω output. This first table entry is used for extrapolation when readings are below the normal range of the sensor. The remaining entries are the valid sensor specific data.

In this example a custom thermocouple is used. The first entry –10mV, 0 kelvin is the point used for extrapolation of data below the first valid data point (100mV, 199

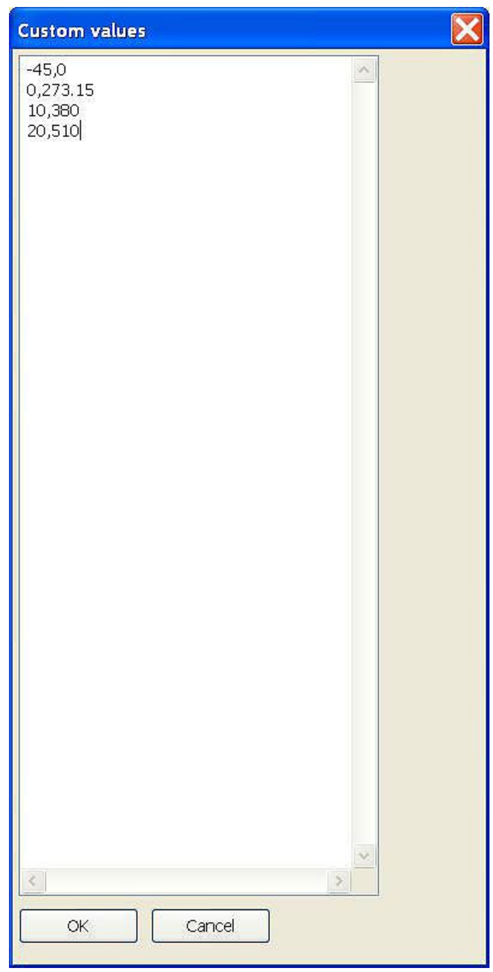


Figure 32. Custom Value Editor

CUSTOM SENSORS

kelvin). The valid data consists of three table entries. Combined with the extrapolation point, four pairs of data are entered into the device. The value for tc–custom length–1 should be 3 in this example (length-1). If the user inputs an incorrect number this results in an error (see Figure 34). Once the correct length is entered and Accept Changes is clicked, the LTC2986/LTM2985 is programmed with the custom table data.

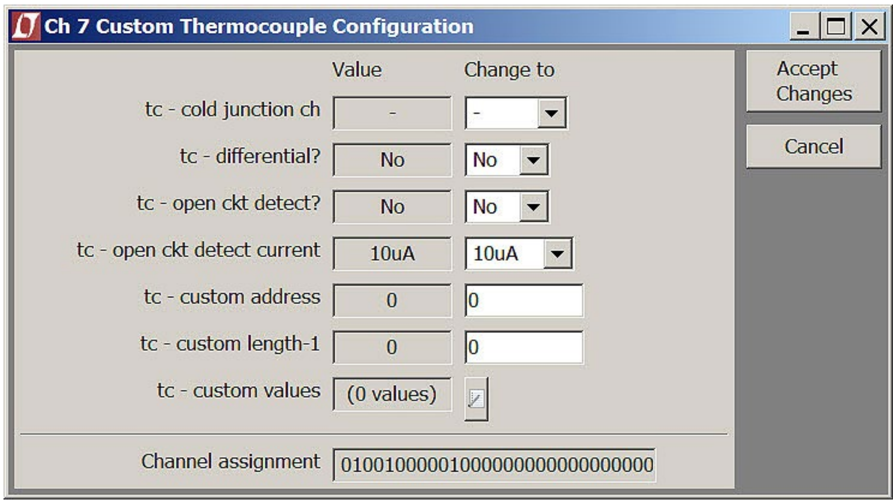


Figure 33. Configuring a Custom Thermocouple. The Window Is Similar When a Custom RTD or Thermistor Is Selected

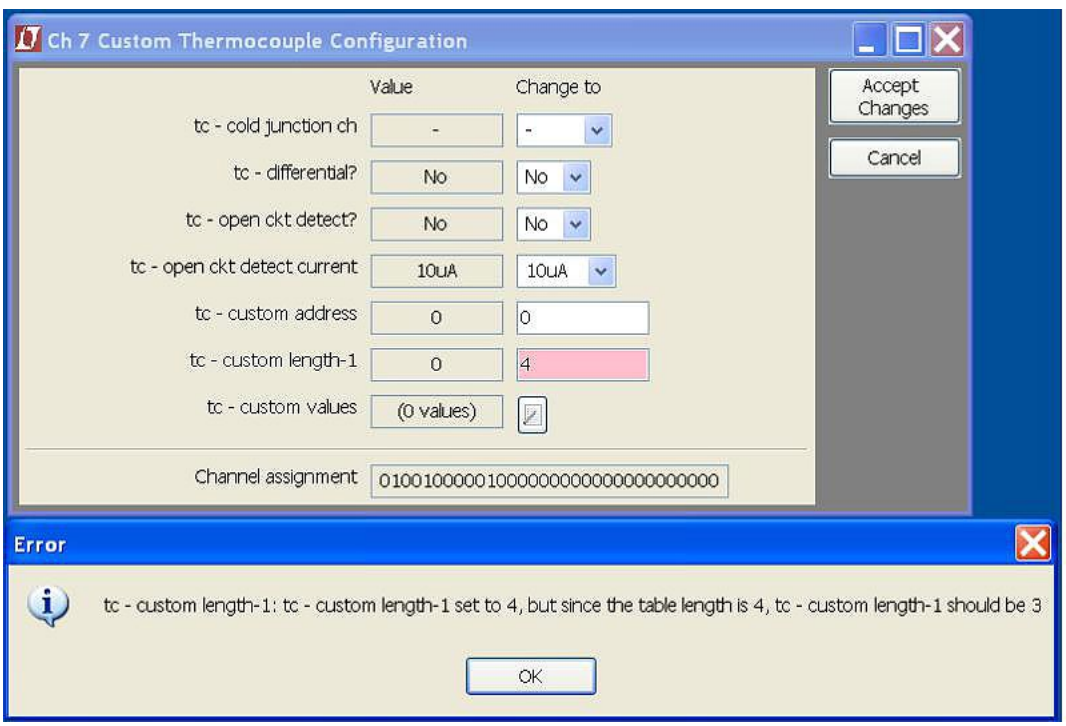


Figure 34. Error Message When Choosing the Wrong Custom Length

CUSTOM SENSORS

A Thermistor Custom Steinhart-Hart table file should look like Figure 35. There are always six Steinhart-Hart coefficients to enter, from A through F.

As was the case before, the LTC2986/LTM2985 can create C code for this configuration, including the custom coefficients. This code can be directly loaded into a Linduino and run.

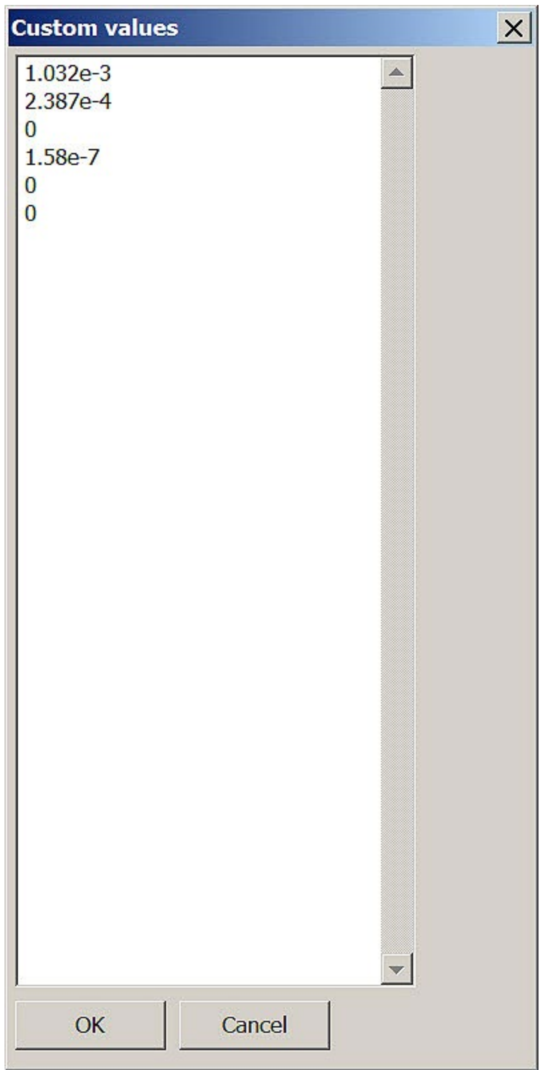


Figure 35. Entering Custom Steinhart-Hart Coefficients

LTC2986-1/LTM2985: USING THE EEPROM

The LTC2986-1 and LTM2985 allow the user to easily load and save configuration data into its EEPROM. To show this feature in action, do the following:

1) Choose a demo board from the configuration: for example, "Configuration" → "Load example" → "Demo board" → "DC2212_THERMOCOUPLE_BOARD.cfg." The screen will look like that in Figure 36.

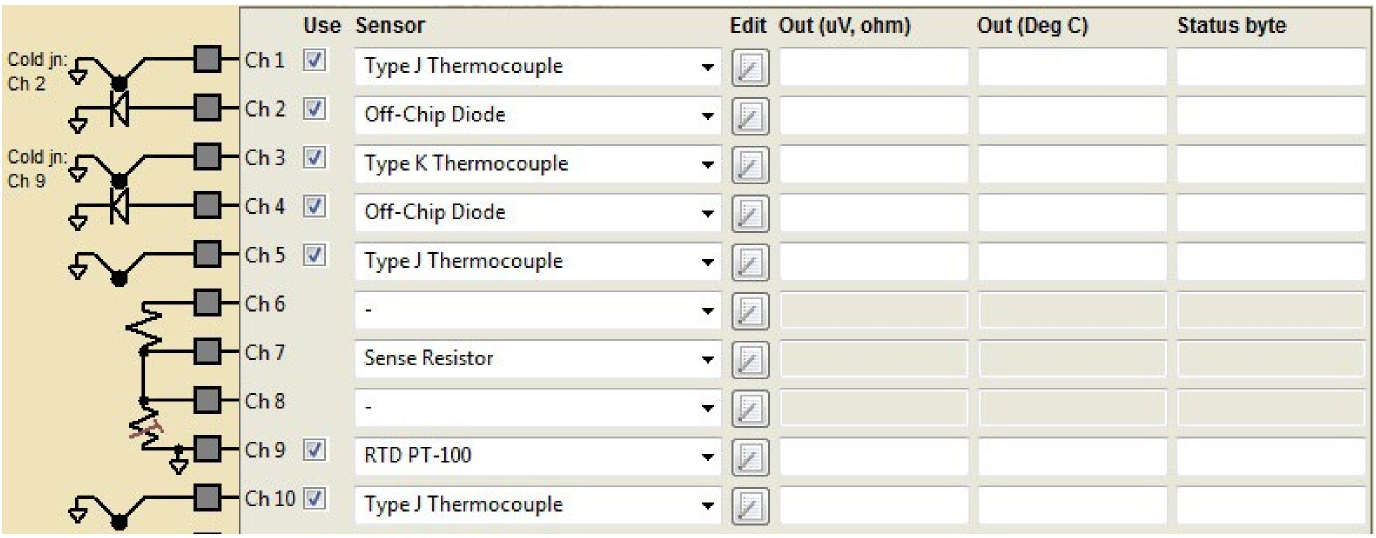


Figure 36. Thermocouple Board Configuration

USING THE EEPROM

2) Attach the setup as shown in Figure 15 into a computer. For this exercise, a daughter board is not needed.
3) Load the configuration into EEPROM: Go to "Configuration" → "EEPROM" → "Load into EEPROM." The message in Figure 37 appears if everything worked correctly.

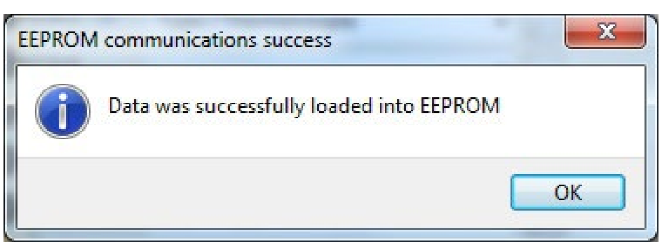


Figure 37. Message Showing Successful Load into EEPROM

4) Go to "Configuration" → "New." The configuration should now be empty.
5) Let the LTC2986-1/LTM2985 go through a power cycle by unplugging and replugging in the USB connection to the DC2026 Linduino demo board.

Load the EEPROM back into the LTC2986-1/LTM2985 by going to "Configuration" → "EEPROM" → "Retrieve from EEPROM." After a successful operation, the original configuration in Figure 36 will reappear, and the software shows the message in Figure 38.

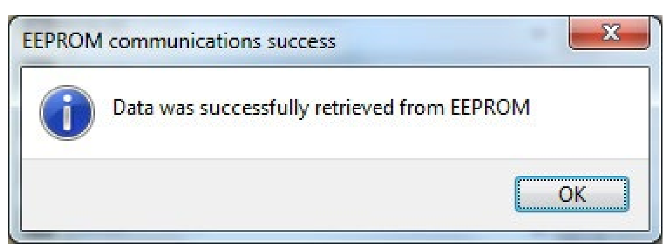


Figure 38. Message Showing Successful Retrieval from EEPROM

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.

Legal Terms and Conditions