RL78 Family

RL78 Internal Temperature Sensor Calibration (Using IAR Toolchain)

Introduction

The following document describes a method to improve the internal temperature sensor accuracy of the RL78 MCU byperforming a calibration. A sample project and evaluation data is also provided.

Target Device

RL78 Family

(The sample project was developed for RL78/G13 RSK (Renesas Starter Kit) using IAR Embedded Workbench.)

Contents

1. Introduction

The RL78 MCU family has an integrated temperature sensor. The temperature sensor provides a voltage that is proportional to the temperature. The sensor is internally connected to the ADC block to allow monitoring the sensor voltage. The internal voltage reference of the RL78 allows using the temperature sensor without requiring an external voltage reference to the MCU. The temperature sensor is fairly linear; however, it does have a significant offset which a single point calibration can easily compensate for. This document will discuss the expected performance of the temperature sensor and provide a method to perform a simple single point calibration.

2. Software environment

The sample code contained in this application note has been checked under the conditions listed in the table below.

Item	Description
Microcontroller used	RL78/G13(R5F100LE)
Operating frequency	● High-speed on-chip oscillator (HOCO) clock: 32 MHz ● CPU/peripheral hardware clock: 32 MHz
Operating voltage	5.0 V (can run on a voltage range of 1.6 V to 5.5 V.) LVD operation (VLVD): OFF
Integrated development environment (IAR)	IAR Systems IAR Embedded Workbench for Renesas RL78 V4.21.3
C compiler (IAR)	IAR Systems IAR C/C++ Compiler for Renesas RL78 V4.21.3.2447
FDL	FDL RL78 Type04 Ver1.05

Tabel 1 Sample Software projects versus environment

3. Temperature Sensor Characteristics

The internal temperature sensor of the RL78 MCU provides a voltage that is proportional to temperature. The nominaloutput voltage of the sensor is 1.05V at 25 degrees C. The sensor voltage changes -3.6 mV/°C. The sensor has an internal connection to the analog-to-digital converter (ADC) of the RL78 MCU. Since the output of the sensor is a voltage there must be a known reference voltage provided to the system. There are two options that can be used when using the sensor. The first option is to use an external reference voltage as shown in Figure 1.

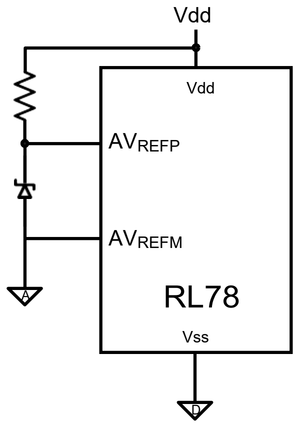


Figure 1

The figure shows a reference diode connected to the external ADC reference inputs of the MCU. In this configurationthe ADC reading of the temperature sensor can be converted to temperature using the formula below:

Temp = [Vref *(ADCreading/ADCfull_scale) – 25Cnominal Voltage]/ Sensor_gain +25C

For the RL78/G13 temperature sensor using a 10 bit conversion this would simplify to

Temp = [Vref *(ADCreading/1024) – 1.05V]/-3.6 mV/degC + 25C

In figure 1 the AVrefm input is used for the negative side of the reference diode. The ADC could be configured to use Vss as the negative side reference for the ADC converter then the AVrefm pin could be used for other purposes. In thatcase the negative side of the diode would be connected to Vss

An external reference diode is an extra cost that is not necessary using the RL78 since it has an internal reference voltage. However, this reference voltage cannot be used as Vref when converting the temperature sensor. Therefore Vdd is used as Vref when converting the voltage sensor and the temperature sensor. Since the ADC can be configuredto provide an internal connection using Vdd as Vref this also frees up the AVrefp pin.

This does not really make the calculation that much more complicated . When converting the internal voltage reference we have the following relationship

ADCvolt = Vint/Vref * 1024

where:

ADCvolt = adc conversion result when measuring internal reference voltage

Vint = Internal reference voltage (1.45V nominal)

Vref = Vdd

When converting the temperature sensor we have:

ADCtemp = Vtemp/Vref * 1024
where:
ADCtemp = adc conversion result when measuring internal temperature sensor
Vtemp = Actual temperature sensor voltage
Vref = Vdd

Solving the two equation for Vref and substituting provides the relationship:

Vtemp/ADCtemp = Vint/ ADCvolt or Vtemp = (Vint * ADCtemp)/ADCvolt

To provide the full conversion to temperature we use the nominal values for the voltage and temperature sensor and theresulting equation is :

Temperature = [(1.45 * ADCtemp/ADCvolt)-1.05]/-3.6mV] + 25C

So to find the temperature a conversion is done on the internal voltage sensor and the internal temperature sensor thenthe equation above is applied.

4. Sensor Measurements System Considerations

In the previous calculations the full scale counts of the ADC were used to determine voltage levels. However, the actual voltage could be anywhere from that level to the next step level. To minimize this error the calculation could add
½ LSB to the calculated value but this typically not done since other errors swamp out this quantization error. Since the RL78/G13 ADC is a 10 bit converter if the MCU power supply is 3.3 volts and that voltage is used for the ADC reference each step of the ADC is equal to 3.3V/1024 or 3.22 mV. This means that the finest resolution that could be provided for temperature is 1 degree C. With a 5V Vdd the step size is actually 4.88 mV so the situation is even worse. To improve on this resolution oversampling is implemented in the sample code. The temperature and voltage sensors are each sampled 16 times. The 16 samples are summed then the result is decimated (divided) by 4. Sampling theory indicates that this technique can improve the effective resolution of the ADC. The number of oversamples and the decimation factor are related to the number of bits, n, to be added by the following formula:

Number of bits added (n) = 4^n /2^n

By oversampling 16 times and decimating by a factor of 4 an additional 2 bits of resolution can be added. The sample code does not use interrupts, the data transfer controller (DTC) or DMAC. This consumes quite a bit of processor time but makes it easier to follow the logic of the conversion in the code. In most systems these conversions would take place using either the DTC or DMAC and the summing and decimation would take place only after all the data was collected.

Another system consideration is the calculation of the temperature from the ADC readings of the temperature sensorand internal voltage. We previously showed that the temperature could be calculated using the formula below

Temperature = [(1.45 * ADCtemp/ADCvolt)-1.05]/-3.6mV + 25C

Though the equation is not overly complex if it was used as is it would require floating point math to be used in the system. In a device like the RL78 that does not have a hardware floating point unit this requires calling libraries which can take quite a bit of time to perform the floating point calculation. In some systems that may be acceptable but in many cases the extra time for the float calculations would not be acceptable. In those cases fixed point math is preferred. The sample code implements both a floating point and a fixed point calculation so the computation time and accuracy can be compared. In the fixed point implementation the fractional reference voltages are scaled by 10,000 andthe temperature sensor gain is scaled by 1000. This provides an effective gain of 10 for the temperature calculation so the displayed resolution can be tracked to 0.1 degree. Notice the formula reference temperature is also scaled by 10 (250 instead of 25C). The portion of the code is shown below for reference

```
#define SENSOR_REF_TEMP_SCALED (250)
#define INT_REF_V_SCALED (145000L) #define
INT_REF_TEMP_SCALED (105000L)#define
TEMP_SENSOR_GAIN_SCALED (36)

g_tempv_int = (int16_t)(((INT_REF_V_SCALED) *g_adc_temperature /g_adc_int_ref)-
(INT_REF_TEMP_SCALED)) ;
g_measured_tempi = (uint16_t)( -(g_tempv_int/ (TEMP_SENSOR_GAIN_SCALED)) +
SENSOR_REF_TEMP_SCALED);
```

The result of the calculation is an integer which represents temperature * 10. This calculation is much faster than the floating point version. In many cases a binary scaling is used for calculation efficiency, however, in this case only constants were scaled and the result did not have any additional scaling so performance was not affected by the base 10scaling.

5. Sensor Errors

For sensors, like the temperature sensor, there are generally three types of error that must be considered.

1. Non-linearity error
2. Gain error
3. Offset error

A sensor transfer function typically has some degree of each type of error. Compensation for each is easier by discussing the different errors individually. Figure 2 shows the ideal transfer function for the RL78 temperature sensor.In the figure linearity error would be seen as a variation in the step size from one degree step to another, in other words,the gain is not constant. The gain error relates to the slope of the line, which is nominally -3.6 mV for RL78 sensor.
The offset error is the variation from the nominal 1.05V at 25C. Figures 3, 4 and 5 show the effect of each individualerror.

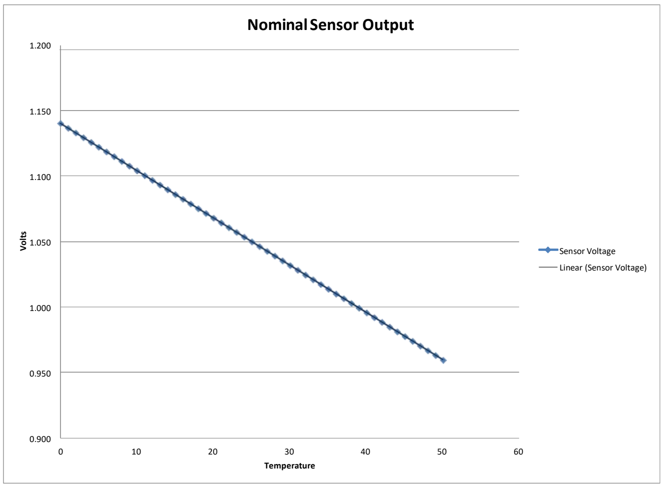


Figure 2

Non-Linearity Error

When non-linearity error is present the actual readings do not fall on the nominal output line, however, the overall slopeand reference points are correct. In figure 3 the overall slope and 25C points are correct but a few readings do not fall on the nominal output line represented in black.

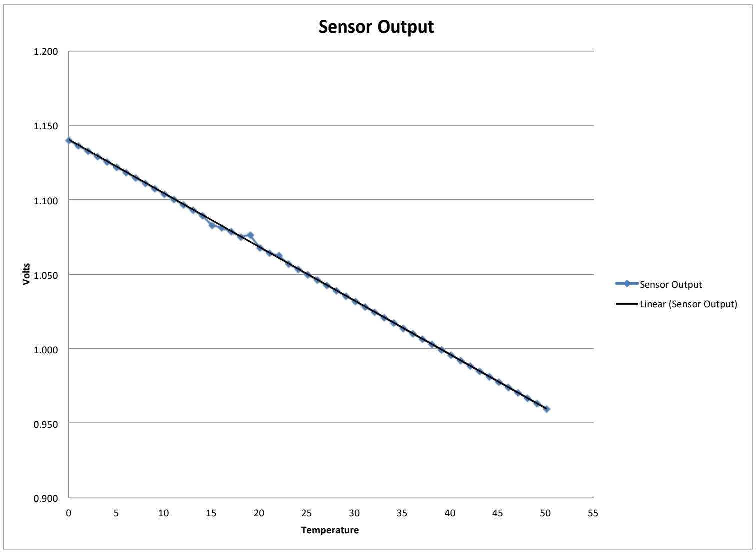


Figure 3

Calibration for non-linearity error is usually impractical, especially with temperature sensors. The non-linearity must be characterized and this requires quite a few calibration points and linear interpolation between the points or non-linear curve fits.

Gain Error

Gain error is present when the overall slope of the sensor output does not match the nominal slope. Figure 4 shows a sensor output where the gain is low. The actual output shown in blue does not match the nominal slope shown in red.

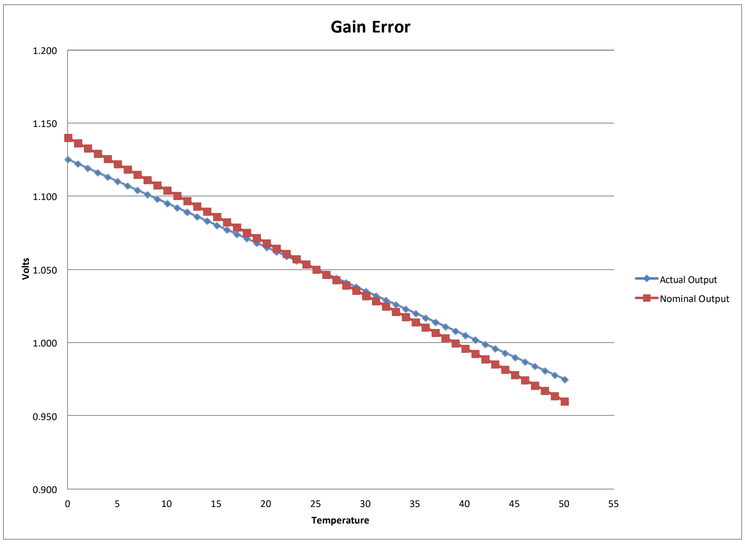


Figure 4

Correcting for gain errors requires a minimum two point calibration so the correct slope can be determined. Typicallythe points are taken at the ends of the range that will be used in the application. Since the gain is not always constant some systems use more points and best fit algorithms to determine the appropriate gain equation.

Offset Error

With offset error the slope of the output voltage is correct but the actual line is "offset" by some DC voltage from the nominal output. Figure 5 shows an output voltage with a positive offset, meaning the actual readings are higher than the nominal readings

Figure 5

Offset error can be calibrated using a single point calibration method. Since the offset is a constant at all outputs it only needs to be measured once then added to each reading. To determine the value for the offset error there are two options. The first would be to force the system to a calibration temperature, for example 25C. Once the system is stabilized at that temperature the indicated temperature from the sensor can be read and the difference between the readings is the required offset compensation that must be added to the readings. This is the method that is most commonly used in calibrations because it ensures the measured parameter is stable and is in the normal range. However, since temperature is typically slow to change and fairly constant in many environments it is possible to just allow the system to stabilize tothe ambient temperature and measure the actual temperature using another calibrated system. This is the method used in the sample program. Room ambient works well as a calibration point in this example since it falls within the expected usage range.

6. RL78 Temperature Sensor Calibration

Figure 6 is a plot of a typical RL78 temperature sensor voltage versus the nominal voltage. Notice the slope is very close to the desired slope and fairly linear, however, the actual sensor output voltage is less than the nominal output.

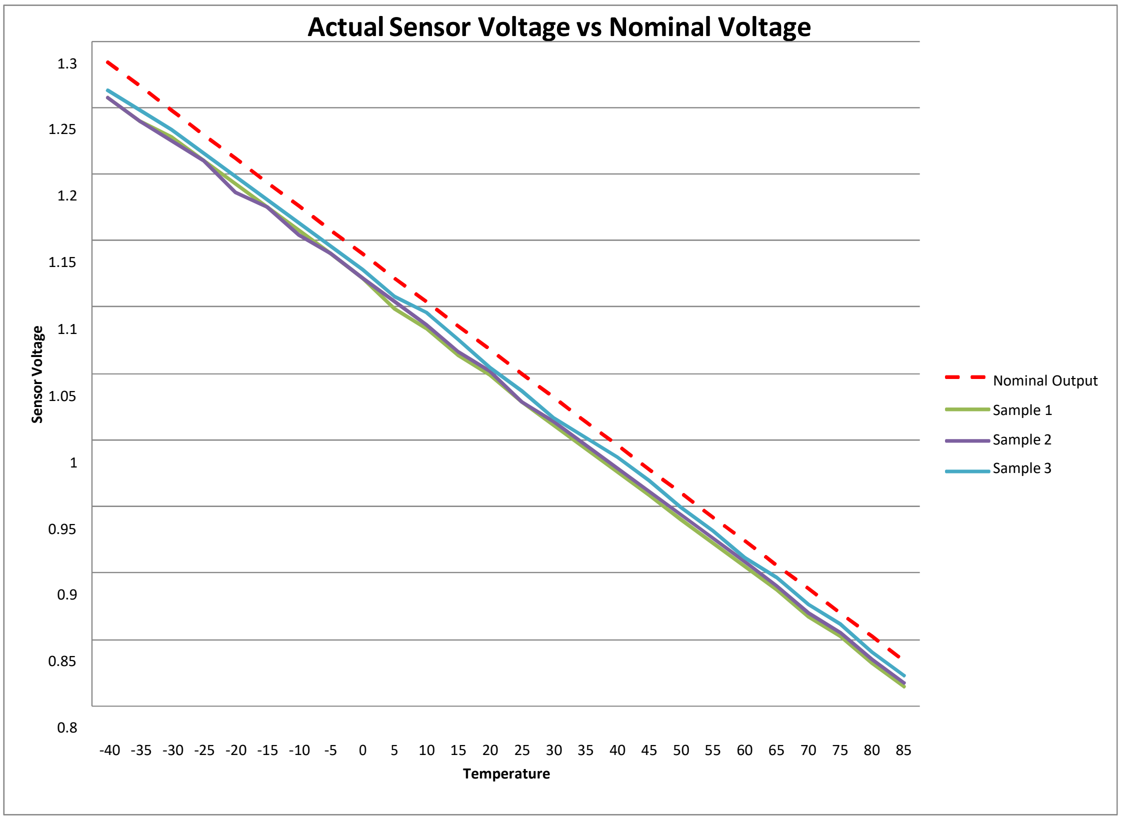


Figure 6

At 25 degrees C the nominal output voltage of the sensor is 1.05V (reference the Hardware manual). The actual outputof the sensor for these devices is closer to 1.03V. The sensor provides a -3.6mV/°C output so the 0.02V difference between the nominal 25C output and the actual 25C output represents a +5.5 degree error in the reading.

Figure 7 shows plots of the indicated temperature from one of the sensors shown above and the error in the reading. Notice the error is fairly constant which means that a single point calibration will typically provide very good results. If the error plot showed a noticeable slope over temperature then a two point calibration would provide better results sinceit would compensate for the “gain” error of the sensor

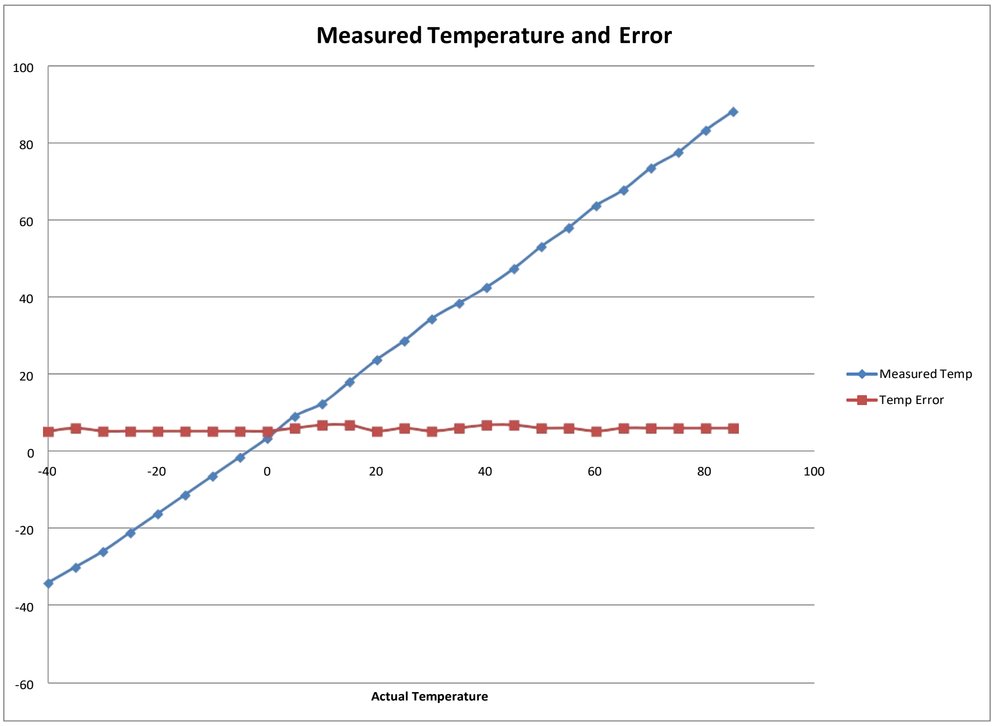


Figure7

The RL78 temperature sensor should typically provide ± 3°C accuracy using a single point calibration method. Two point calibration does not typically provide better results and may result in worse results depending on the actual pointsmonitored for a device.

7. Sample Project

The sample project was developed for RL78/G13 RSK (Renesas Starter Kit) using IAR Embedded Workbench.

This sample project uses FDL. Please refer to 9.How to import FDL for detailed instructions.

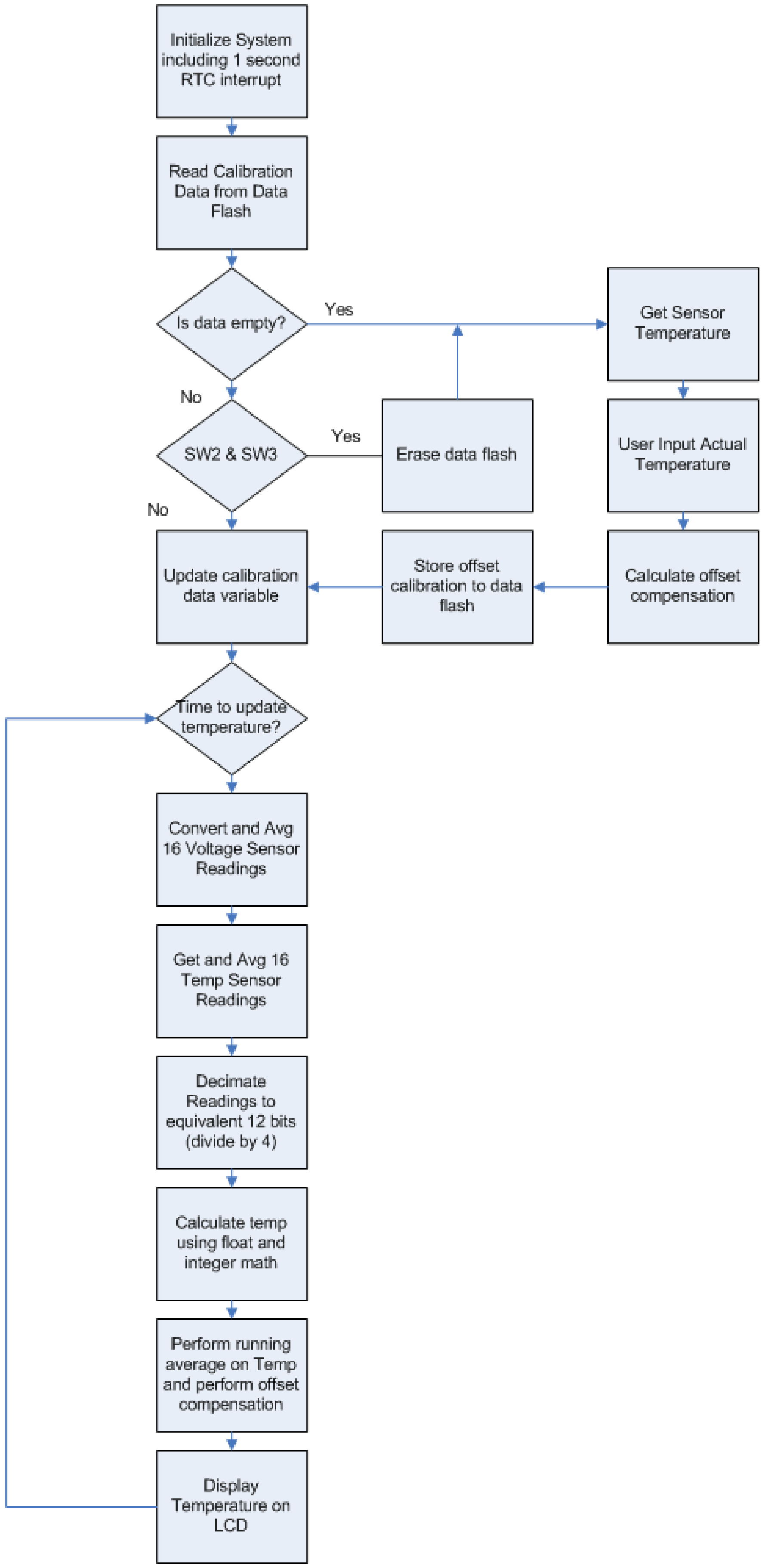
The structure of the sample code is shown in the flowchart on the following page. The calibration data in this project is stored in the data flash of the RL78. The system checks the value of the data flash on startup and if it is blank (0XFFFFFFFF) the code branches to a calibration routine. The user can also force the MCU to the calibration routineby holding SW2 and SW3 when Reset is released.

The calibration routine performs sixteen temperature calculations to determine an average measured temperature. The user is then prompted on the LCD screen to input the actual temperature. SW2 is used to decrease the reading, SW3 is used to increase the reading and SW1 is used to enter the value. After the actual value is entered the system calculates the calibration offset factor and stores it into the data flash.

When there is valid data in the data flash calibration value the system then performs a temperature conversion once a second. The ADC values for the temperature sensor and internal voltage sensor are oversampled to provide an equivalent 12 bit value. These values are used to calculate the measured temperature. The calculation is performed using both floating point and integer math to allow the user to compare the time and accuracy of the results. The integer result is averaged and displayed on the LCD.

Measured temperature using the sample code and RSK performed within ±3 °C of the actual temperature using a thermocouple probe bonded to the RL78 package. Testing of measured temperature to ambient provided similar results when the calibration was done using the ambient temperature actual reading.
Results from testing between 0° C and 50° C is shown below

Actual Temp	0	5	10	15	20	25	30	35	40	45	50
Measured Temp	7.4	12.2	16.8	20.9	28.2	34	38.6	43.4	49.4	54.7	59.6
Calibrated Temp	-1	3.8	8.4	12.5	19.8	25.6	30.2	35	41	46.3	51.2
Error	-1	-1.2	-1.6	-2.5	-0.2	0.6	0.2	0	1	1.3	1.2


Initialize System including 1 second RTC interrupt
Read Calibration Data from Data Flash
Is data empty?
Yes
No
SW2 & SW3
Yes
No
Get Sensor Temperature
User Input Actual Temperature
Calculate offset compensation
Store offset calibration to data flash
Erase data flash
Update calibration data variable
Time to update temperature?
Convert and Avg 16 Voltage Sensor Readings
Get and Avg 16 Temp Sensor Readings
Decimate Readings to equivalent 12 bits (divide by 4)
Calculate temp using float and integer math
Perform running average on Temp and perform offset compensation
Display Temperature on LCD

8. Summary

The RL78 internal temperature sensor provides a cost effective method to measure temperatures. The internal voltagereference of the RL78 allows the temperature sensor reading to be obtained without having to add an external voltage reference diode. Typical accuracy of ±3 °C is expected when a single point calibration of the sensor reading is performed. The data flash feature of the RL78 provides a convenient storage point for the calibration data.

9. How to import FDL

(1) "https://www.renesas.com/jp/ja/software-tool/data-flash-libraries#overview"

Download the latest FDL from the URL above.

(2) Copy the downloaded FDL folder to the root directory of the project.

(3) Select "C/C++Compiler" -> "Preprocessor" from the project options in IAR.

(4) Specify folders copied to the root directory to subfolders in "Additional include directories". "

(e.g., : $PROJ_DIR$\FDL\IAR_210\lib)

(5) "https://github.com/IARSystems/ewrl78-linker-config-flashlibs"

Download the linker script for IAR from the GitHub URL above.

(6) Copy the following folder in the download to the user function folder.

- trio_lnkR5F100xE.icf
- common.icf
- self_ram.icf

(7) From IAR project options, check "Linker" -> "Config" -> "Override default".

(8) Specify "trio_lnkR5F100xE.icf".

(9) In the "Configuration file symbol definitions," enter the corresponding FDL symbols from the "RAM reservation symbols" table at the bottom of the GitHub URL.

(e. g : __RESRVE_T04_FDL=1)

(10) Add FDL libraries to "Linker" -> "Library" -> "Additional libraries" from the IAR project options.

(e. g : $PROJ_DIR$\FDL\IAR_210\lib\pfdl.a)

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	June.28.2012	-	First edition issued
1.10	June.24.2022	2	Update Software environment
		10	Update Sample Project
		12	Add How to import FDL

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on

   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state

   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins

   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals

   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin

   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses

   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products

   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/.