Texas Instruments

Application Report

SWRA415C–June 2014

CC112X/CC120X On-Chip Temperature Sensor

Bjarte Nystøyl and Torstein Ermesjø

ABSTRACT

This application report provides the necessary information to use the temperature sensor of the CC112X and CC120X families. The temperature sensor is based on a proportional to absolute temperature (PTAT) current from a bandgap cell fed to a resistor to generate a PTAT voltage. It is possible to read out the temperature information either as an analog voltage on a general-purpose input/output (GPIO) pin or using the on-chip analog-to-digital converter (ADC) to convert the voltage to a digital readout. This document is divided into two parts: the first part covers the analog readout and the second covers the digital readout.

Contents

List of Figures

List of Tables

1 Analog Readout

1.1 Operation

The temperature sensor is activated using the register settings of Table 1, which makes the GBIAS output a single-ended voltage measurement on GPIO1.

Table 1. Register Settings for Temperature Sensor

Register	Value
IOCFG1	0x80
ATEST	0x2A
ATEST_MODE	0x0C
GBIAS1	0x07

Setting IOCFG1 to 0x80 configures the GPIO1 pad into analog mode (digital GPIO input and output is disabled). The remaining registers set up the ATEST (analog test) module to output the temperature value as a PTAT voltage on the GPIO1.

1.2 Temperature Sensor Parameters

General Information	Value	Unit
Temperature sensor fitted from	-40 to +85	°C
Effect of supply voltage deviance	1.17	mV/VDD-V
Effect of supply voltage deviance	0.44	°C /VDD-V

Changes in the supply voltage affect the voltage of the GPIO pin, and the supply voltage must be stable in order to get accurate temperature sensor readings.

Table 2. Typical Temperature Sensor Parameters

Technical Information	Value	Unit
VDD - 2 V		
Typical output voltage @ 0°C	727.42	mV
Typical output voltage @ 25°C	793.73	mV
Temperature coefficient	2.6598	mV/°C
VDD - 3 V		
Typical output voltage @ 0°C	728.55	mV
Typical output voltage @ 25°C	794.78	mV
Temperature coefficient	2.6733	mV/°C
VDD - 3.6 V		
Typical output voltage @ 0°C	730.62	mV
Typical output voltage @ 25°C	796.94	mV
Temperature coefficient	2.6773	mV/°C

1.3 Calibration

As seen in Figure 1, the CC112X/CC120X temperature sensor voltage is highly linear, but for some devices there is an offset in the GPIO1 voltage from the typical (average) value that could potentially give an error of up to ±10°C in the temperature reading. In order to ensure accurate temperature sensor measurements, the sensor must be calibrated. There are two simple approaches depending on the required accuracy level: single- and two-point calibration.

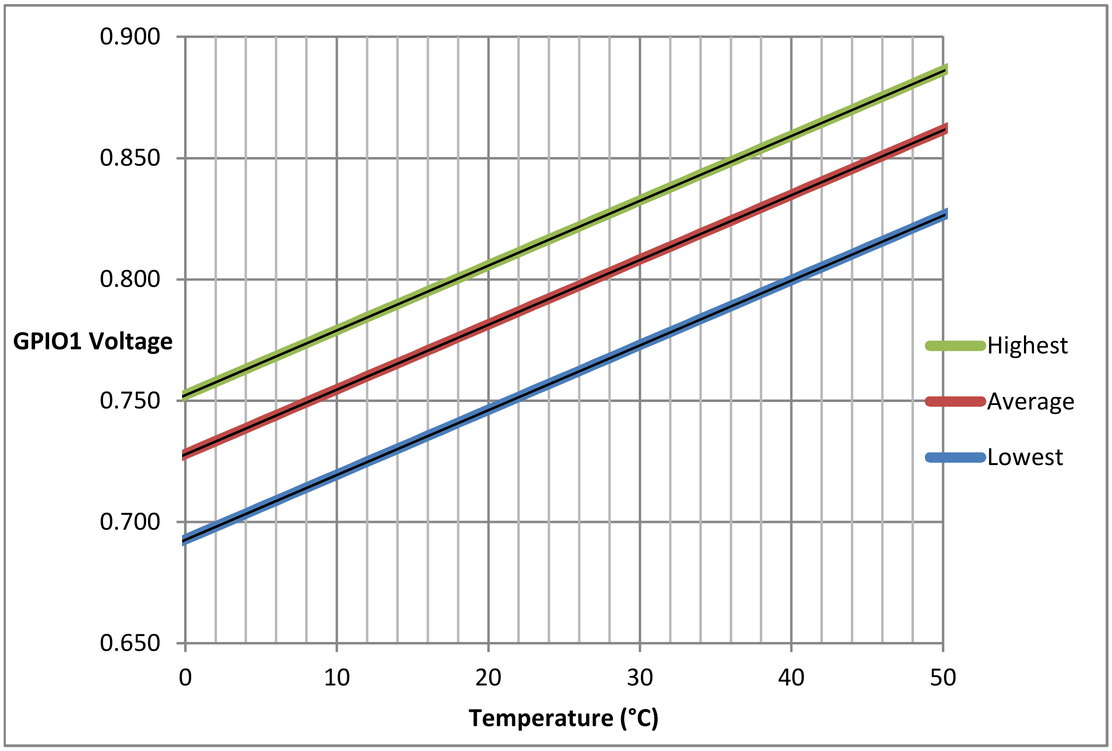

Figure 1. GPIO1 Voltage vs Temperature

1.4 Single-Point Calibration

This is a simple and fast approach that can be applied for applications targeting approximately ±1°C accuracy within a limited temperature range around the temperature used for the single-point calibration, or approximately ±2°C accuracy across the -40°C to +85°C temperature range.

1.4.1 Performing Single-Point Calibration

The calibration should be performed at the center of the temperature range in which the device will operate. A given temperature, *T*, will be given as:

$$T = T_{CALIBRATION} + \frac{(V_{MEASURED} - V_{CALIBRATION})}{t_C} \quad (1)$$

- $T_{CALIBRATION}$ is the temperature when the calibration is performed
- t_c is the temperature coefficient for the given supply voltage (see the typical temperature parameters in Table 2)
- $V_{MEASURED}$ is the voltage of the GPIO1 pin at a given temperature
- $V_{CALIBRATION}$ is the GPIO1 voltage at the calibration temperature

Performing a single-point calibration removes the error caused by the device-specific voltage offset seen in Figure 1. The temperature reading accuracy is then limited by the accuracy of the individual temperature coefficients as the typical temperature coefficient is used in Equation 1.

Figure 2 shows the maximum error in the temperature reading when using the lowest and highest temperature coefficients out of 30 devices from different processing corners.

- Approximately ±2°C accuracy is possible across the -40°C to +85°C temperature range with single-point calibration and using the typical temperature coefficient in Table 2.
- Approximately ±1°C accuracy is possible across the temperature range defined by $T_{CALIBRATION}$ ± 25°C with single-point calibration and using the typical temperature coefficient in Table 2.

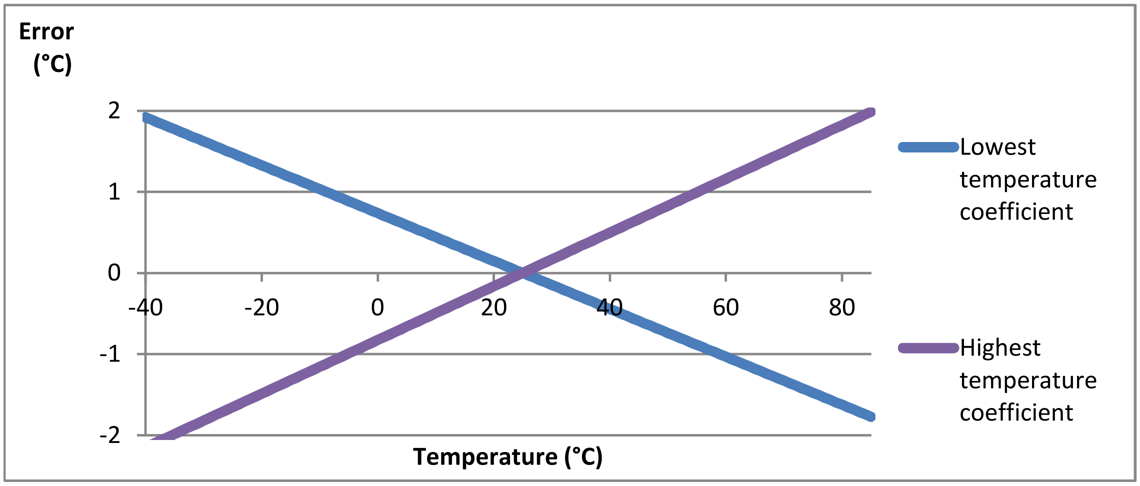

Figure 2. Temperature Error Due to Different Temperature Coefficients After Single-Point Calibration

1.4.2 Single-Point Calibration Example

A CC112X/CC120X device is operated using at 3 V supply voltage. The temperature coefficient is typically 2.673 mV/°C and for each degree Celsius increase in temperature the GPIO1 voltage increases by 2.673 mV.

The device is calibrated at room temperature (25°C), and the GPIO1 voltage is measured to be 793.0 mV. After changing the temperature, the GPIO1 voltage is measured to be 830.0 mV. This corresponds to a temperature T of:

$$T = 25°C + \frac{(830\ mV - 793\ mV)}{2.673\ mV / °C}$$
$$T = 25°C + 13.84°C = 38.84°C \quad (2)$$

1.5 Two-Point Calibration

If the application requires better accuracy than given by the single-point calibration, a two-point calibration must be used to correct for chip-to-chip variations in the temperature coefficients. As the sensor is highly linear, a two-point calibration will ensure high accuracy across the full temperature range of the chip.

1.5.1 Performing Two-Point Calibration

Choose two calibration temperatures more than 10°C apart, called T0 and T1, and set the reference voltage (V_{DD}) to what it will be in the final product.

NOTE: Changes in the voltage supply will influence the temperature sensor output.

Measure the output from the GPIO1 pin (V0 and V1) at the corresponding temperatures.

The temperature coefficient has a typical value of 2.673 mV/°C. The exact coefficient (t_c) for a given device is calculated as:

$$t_c = \frac{V_1 - V_0}{T_1 - T_0} \quad (3)$$

Using the exact coefficient, the measured voltage of the GPIO1 pin ($V_{MEASURED}$), the temperature (T0) and the GPIO1 voltage (V0) of the first calibration, the temperature, *T*, can be found as:

$$T = T_0 + \frac{(V_{MEASURED} - V_0)}{t_C} \tag{4}$$

1.5.2 Two-Point Calibration Example

A CC112X/CC120X device is operated using a 3 V supply voltage, and will have a typical temperature coefficient of 2.673 mV/°C.

The device is calibrated at two temperatures: 0°C and 25°C (T0 and T1). The respective GPIO1 voltages are measured to be 743.379 mV and 808.312 mV (V0 and V1). The exact temperature coefficient t_c is given as:

$$t_C = \frac{808.312\ mV - 743.379\ mV}{25°C - 0°C} = 2.5973\ mV / °C \tag{5}$$

At a given temperature *T*, the GPIO1 voltage is measured to be 921.465 mV. This corresponds to:

$$T = 0°C + \frac{(921.465\ mv - 743.379\ mv)}{2.5973\ mv / °C} = 68.57°C \tag{6}$$

NOTE: Single-point calibration at 25°C, using the typical t_c of 2.673 mV/°C, would in this case give a temperature reading of 67.33°C, which would have an error of 1.24°C.

1.6 *Change in Supply Voltage (V_{DD})*

As seen in Figure 3, the voltage measured on the GPIO1 pin depends on the supply voltage. Changing the supply voltage affects the measured voltage on the GPIO1 pin by typically 1.17 mV/V. This means that if the supply voltage is decreased by 1 V, the voltage measured at the GPIO1-pin is typically 1.17 mV lower.

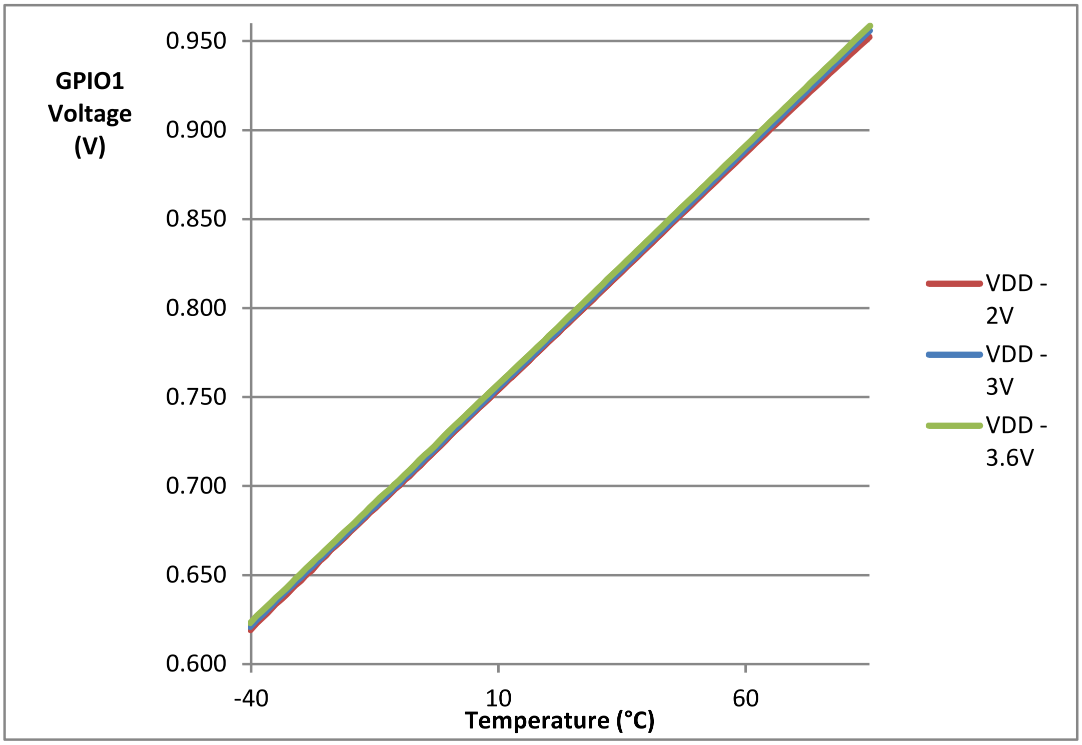

Figure 3. Typical GPIO1 Measurements vs Supply Voltage

2 Digital Readout

2.1 Operation

The digital readout uses the IFADC in the receive chain to convert the analog temperature sensor voltage to a digital value, which is read from the CHFILT register.

In order to get a digital readout, the following must be done:

- The chip must be in receive mode (RX)
- The DC filter must be disabled and the IF frequency set to zero. Otherwise, the DC information is filtered out.
- The IFAMP must be off; this is done with the chip set in debug mode.

Table 3 lists the registers used to activate digital readout for CC112x and CC120x, respectively.

Table 3. Register Settings for Digital Readout

CC112x		
Register	**Value**	**Comment**
DCFILT_CFG	0x40	Turn off DC filtering
MDMCFG1	0x47	Single ADC on, I channel
CHAN_BW	0x81	Bypass channel filter
FREQ_IF_CFG	0x00	Zero IF
ATEST	0x2A	Temp sensor on
ATEST_MODE	0x07	Temp sensor on
GBIAS1	0x07	Temp sensor on
PA_IFAMP_TEST	0x01	Route voltage into the ADC
CC120x		
Register	**Value**	**Comment**
DCFILT_CFG	0x40	Turn off DC filtering
MDMCFG1	0x47	Single ADC on, I channel
MDMCFG0	0x85	Bypass channel filter
CHAN_BW	0x81	Lowest decimation factor
IF_MIX_CFG	0x00	Zero IF
ATEST	0x2A	Temp sensor on
ATEST_MODE	0x07	Temp sensor on
GBIAS1	0x07	Temp sensor on
PA_IFAMP_TEST	0x01	Route voltage into the ADC

Figure 4 shows the digital readout is not linear as a function of temperature. A second order equation was chosen to best fit the measured data:

$$\text{Digital readout} = t_{c1} \times T^2 + t_{c2} \times T + t_{c3} \quad (7)$$

The analog temperature sensor voltage depends on the supply voltage (V_{DD}) and the digital readout will, therefore, have a supply voltage dependency. Table 4 gives typical digital readout values @0°C and 25° as well as temperature coefficients for different supply voltages.

Table 4. Typical Parameters for Register Readout

Technical Information	Value
VDD - 2 V	
Typical readout data @ 0°C	100
Typical readout data @ 25°C	21047
Temperature coefficient (t_{c1})	-3.72
Temperature Coefficient (t_{c2})	957.65
Temperature Coefficient (t_{c3})	385.21
VDD - 3 V	
Typical readout data @ 0°C	-2059
Typical readout data @ 25°C	19874
Temperature coefficient (t_{c1})	-3.32
Temperature Coefficient (t_{c2})	992.1
Temperature Coefficient (t_{c3})	-2629.91
VDD - 3.6 V	
Typical readout data @ 0°C	-3402
Typical readout data @ 25°C	19133
Temperature coefficient (t_{c1})	-3.29
Temperature Coefficient (t_{c2})	1010.87
Temperature Coefficient (t_{c3})	-3945.8

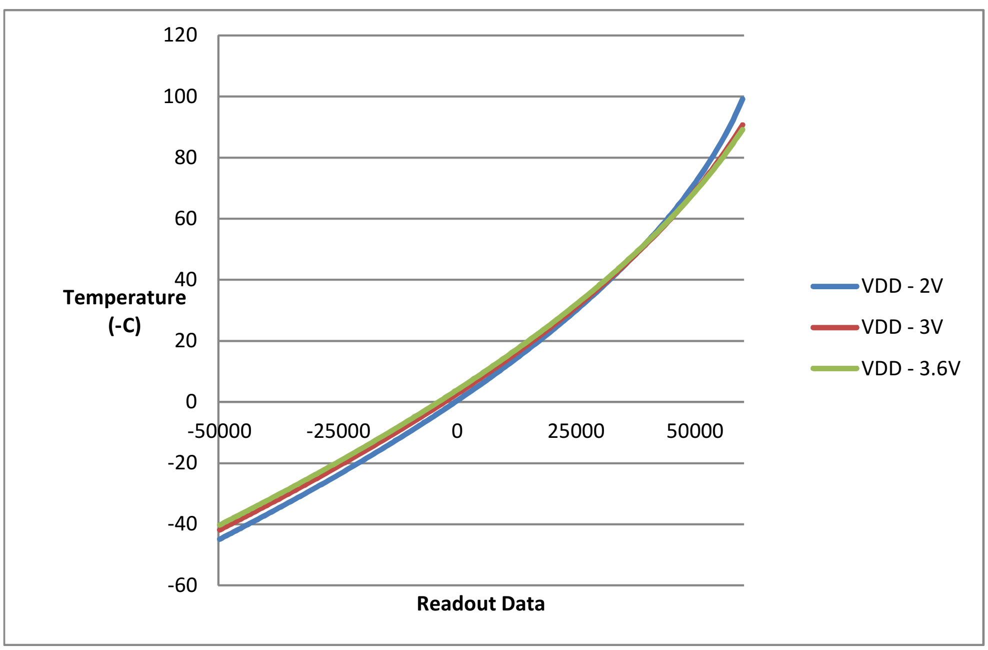

Figure 4. Register Values vs. Temperature

2.2 Calibration

To eliminate the offset discussed in Section 1, it is recommended to execute a single-point calibration before using the temperature sensor. The temperature, *T*, can then be estimated by:

$$T = \frac{t_{c2} + \sqrt{4 * t_{c1} * (t_{c3} - D_{MEASURED})}}{2 * t_{c1}} - \frac{t_{c2} + \sqrt{4 * t_{c1} * (t_{c3} - D_{CALIBRATION})}}{2 * t_{c1}} + T_{CALIBRATION} \quad (8)$$

- $D_{MEASURED}$ is the readout data at the given temperature
- $D_{CALIBRATION}$ is the readout data at the calibration temperature
- $T_{CALIBRATION}$ is the calibration temperature
- t_{c1}, t_{c2}, t_{c3} are the temperature coefficients for the given V_{DD} presented in Table 4

The following accuracy can be achieved using single-point calibration:

- Approximately ±5°C accuracy is possible over a 90°C range centered at $T_{CALIBRATION}$ when the single-point calibration is used with the coefficients given in Table 4.
- Approximately ±2°C accuracy is given over the range defined by $T_{CALIBRATION}$ ±10°C by the single-point calibration with the coefficients shown in Table 4.

2.3 Code Implementation

The function tempRead shown in Example 2, Example 3 and Example 4 shows how to read the temperature on a CC112x. The same code can be used with CC120x by changing the registers shown in Table 3.

After temperature is read the chip must be reset and all radio configurations have to be redone like shown in Example 1.

Example 1. Radio Must be Reset After Temperature Reading

```
//Read temperature, chip must be reset after this!
temp = tempRead();

//Reset chip
trxSpiCmdStrobe(CC112X_SRES);

// Write radio registers
registerConfig();
```

In Figure 2, a function for reading temperature and returning it as a Celsius value is shown. The specific registers are set and the chip is put in debug mode to be able to turn off the IFAMP and read the digital value from the CHFILT register.

Example 2. tempRead Function – Write Relevant Registers and Put Chip in RX

```
static int8 tempRead(void) {
//Variables
  uint8 RegValue = 0;
  uint8 marcStatus;
  uint8 writeByte;
  uint32 ADCValue_I = 0;
  int8 celsius = 0;
  //String to put radio in debug mode
  uint8 txBuffer[18] =
{0x0F,0x28,0x02,0x90,0x42,0x1B,0x7E,0x1F,0xFE,0xCD,0x06,0x1B,0x0E,0xA1,0x0E,0xA4,0x00,0x3F};
  //Constants for temperature calculation
  float a = -3.3;
  float b = 992;
  float c = -2629.9;

  //Register settings specific for temp readout.
  writeByte = 0x40;
  cc112xSpiWriteReg( CC112X_DCFILT_CFG, &writeByte, 1);  //Tempsens settings, bit 6 high
  writeByte = 0x47;
  cc112xSpiWriteReg( CC112X_MDMCFG1, &writeByte, 1);  //Tempsens settings, single ADC, I channel
  writeByte = 0x81;
  cc112xSpiWriteReg( CC112X_CHAN_BW, &writeByte, 1);  //Tempsens settings, bit 7 high Bypass ch filt.
  writeByte = 0x00;
  cc112xSpiWriteReg( CC112X_FREQ_IF_CFG, &writeByte, 1);  //Tempsens settings, 0-IF
  writeByte = 0x2A;
  cc112xSpiWriteReg( CC112X_ATEST, &writeByte, 1);  //Tempsens settings
  writeByte = 0x07;
  cc112xSpiWriteReg( CC112X_ATEST_MODE, &writeByte, 1); //Tempsens settings
  writeByte = 0x07;
  cc112xSpiWriteReg( CC112X_GBIAS1, &writeByte, 1); //Tempsens settings
  writeByte = 0x01;
  cc112xSpiWriteReg( CC112X_PA_IFAMP_TEST, &writeByte, 1); //Tempsens settings

  //Set chip in RX
  trxSpiCmdStrobe(CC112X_SRX);

  //Read marcstate and wait until chip is in RX
  do {
    cc112xSpiReadReg(CC112X_MARCSTATE, &marcStatus, 1);
  } while (marcStatus != 0x6D);
```

The radio must be put in debug mode to be able to turn the IFAMP off and read the digital value from the CHFILT register.

Example 3. tempRead Function – Put the Radio in Debug Mode

```
//#### Set radio in debug mode ####
// Write debug init to tx fifo
cc112xSpiWriteTxFifo(txBuffer,sizeof(txBuffer));
// Run code from FIFO
writeByte=0x01;
cc112xSpiWriteReg( CC112X_BIST, &writeByte, 1);
// Strobe IDLE
trxSpiCmdStrobe(CC112X_SIDLE);
// Set IF AMP in PD
writeByte=0x1F;
cc112xSpiWriteReg( CC112X_WOR_EVENT0_LSB, &writeByte, 1);
// Strobe SXOFF to copy command over
trxSpiCmdStrobe(CC112X_SXOFF);
//#### Radio in Debug Mode ####
```

With the radio in debug mode, wait until the data in CHFILT is valid and read the data. Then, the value is converted to Celsius and returned.

Example 4. tempRead Function – Wait Until Data From CHFILT is Valid, Read Data and Convert to Celsius

```
  //Wait until channel filter data is valid
  do {
    cc112xSpiReadReg(CC112X_CHFILT_I2, &RegValue, 1);
  } while (!RegValue&0x08);

  //Read ADC value from CHFILT_I registers
  cc112xSpiReadReg(CC112X_CHFILT_I2, &RegValue, 1);
  ADCValue_I = ((uint32)RegValue) << 16;
  cc112xSpiReadReg(CC112X_CHFILT_I1, &RegValue, 1);
  ADCValue_I |= (((uint32)RegValue) << 8) & 0x0000FF00;
  cc112xSpiReadReg(CC112X_CHFILT_I0, &RegValue, 1);
  ADCValue_I |= (uint32)(RegValue) & 0x000000FF;

  //Convert ADV value to celsius
  celsius = (int) ( (-b+sqrt(pow(b,2)-(4*a*(c-ADCValue_I)) ) ) / (2*a));

  //Return degrees celsius
  return celsius;
}
```

3 References

1. *High Performance RF Transceiver for Narrowband Systems Data Sheet* (SWRS112)
2. *High Performance Low Power RF Transceiver Data Sheet* (SWRS111)
3. *Ultra-High Performance RF Narrowband Transceiver Data Sheet* (SWRS120)
4. *CC112X/CC1175 Low-Power High Performance Sub-1 GHz RF Transceivers/Transmitter User's Guide* (SWRU295)

Revision History

Changes from B Revision (September 2013) to C Revision **Page**

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.