AN1413

Temperature Compensation of a Tuning Fork Crystal Based on MCP7941X

Author: Alexandru Valeanu
Microchip Technology Inc.

INTRODUCTION

An increasing number of applications that involve time measurement are requiring a Real-Time Clock device. The MCP7941X is a feature-rich RTCC that incorporates EEPROM, SRAM, unique ID and timestamp.

This application note describes how to compensate the parabolic thermal drift of some crystals in RTCC-based projects, using the Calibration register (OSCTRIM register).

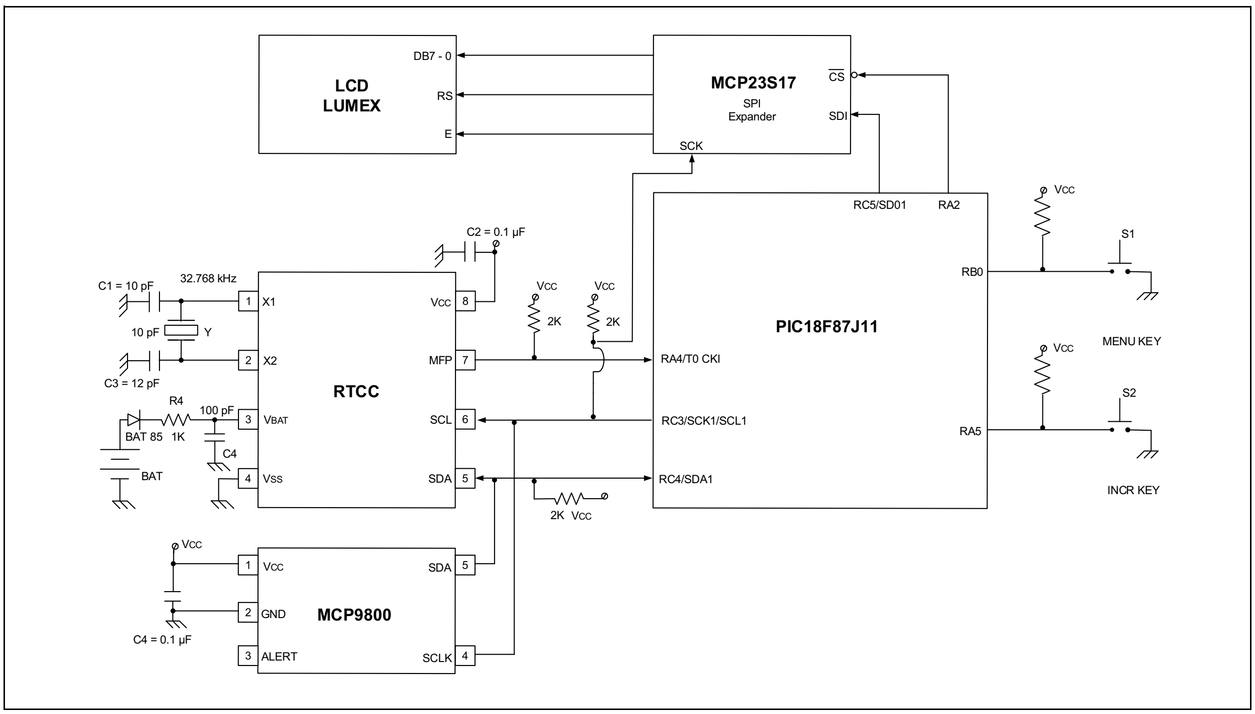
FEATURES OF THE RTCC

- I^2C Bus Interface
- RTCC with Time/Date Registers: Year, Month, Date, Day of Week, Hours, Minutes, Seconds
- Support for Leap Year
- Low-Power CMOS Technology
- Input for External Battery Backup
- 64 Backed-up Bytes of SRAM
- On-Board 32.768 kHz Crystal Oscillator for the RTCC
- On-Chip Digital Trimming/Calibration of the Oscillator
- Operates down to 1.8V
- Backup Voltage down to 1.3V
- Operating Temperature Range:
  - Industrial (I): -40°C to +85°C
- Multifunction Pin:
  - Open-Drain configuration
  - Programmable clock frequency output
- Interrupt Capability (based on the two sets of Alarm Registers, ALM0 and ALM1)
- Time Saver Function
- Timestamp Registers for Holding the Time/Date of Crossing:
  - from VCC to VBAT
  - from VBAT to VCC

SCHEMATIC

The schematic includes the PIC18 Explorer demo board and the I^2C RTCC PICtail™ AC164140 daughter board, as shown in Figure 1.

FIGURE 1: SCHEMATIC



The resources used on the demo board are:

- LCD module
- 16-bit I/O expander with SPI interface (MCP23S17)
- PIC18F87J11 microcontroller
- two push buttons

The I^2C RTCC PICtail™ AC164140 daughter board includes:

- an I^2C temperature sensor (MCP9800), which is connected to the I^2C bus
- an I^2C RTCC
- a 32,768 Hz crystal driving the internal clock of the RTCC
- a 3-volt battery sustaining the RTCC when VCC is not present on the demo board

To access the LCD through a minimum of pins, the SPI on the MSSP1 module is used, in conjunction with a 16-bit I/O expander with SPI interface (MCP23S17).

The two on-board push buttons S1 and S2 are connected to RB0, RA5 GPIOs. The I^2C RTCC is part of the PICtail daughter board and is directly connected to the MSSP1 module of the microcontroller. All connections between the I^2C RTCC and the microcontroller (SDA, SCL, MFP) are open-drain and use pull-up resistors.

FUNCTIONAL DESCRIPTION

MCP7941X is an I^2C slave device, working on the related bidirectional 2-wire bus. SDA is a bidirectional pin used to transfer addresses and data in and out of the device. It is an open-drain pin, therefore, the SDA bus requires a pull-up resistor to VCC (typically 10 kΩ for 100 kHz, 2 kΩ for 400 kHz). For normal data transfers, SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the Start and Stop conditions. SCL input is used to synchronize the data transfer from and to the device. The related internal structures have the following device address/control bytes (the RTCC is included in the SRAM bank):

- RTCC + SRAM: 0xDE for writes, 0xDF for reads
- EEPROM: 0xAE for writes, 0xAF for reads

The chip can support speeds up to:

- 400 kHz at 2.5V to 5V
- 100 kHz at 1.8V to 2.5V

The MCP9800 temperature sensor has the following I^2C addresses/control bytes:

- 0x90 for writes
- 0x91 for reads

DETAILS ABOUT IMPLEMENTATION

The application is designed around the PIC18 Explorer demo board, running on a PIC18F87J11 microcontroller. The code is written using the C18 compiler. The firmware shows how to compensate a parabolic thermal drift of some crystals, using the Calibration register, included in the RTCC structure, at address 08h.

The operation of this register is described in the MCP7941X data sheet (DS20002266):

"The MCP7941X features digital trimming to correct for inaccuracies of the external crystal or clock source, up to roughly ±129 ppm when CRSTRIM = `0`. In addition to compensating for intrinsic inaccuracies in the clock, this feature can also be used to correct for error due to temperature variation. This can enable the user to achieve high levels of accuracy across a wide temperature operating range.

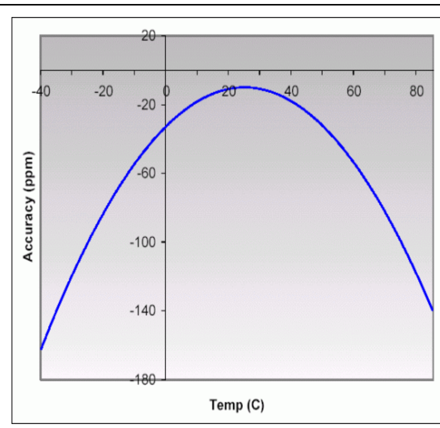
Digital trimming consists of the MCP7941X periodically adding or subtracting clock cycles, resulting in small adjustments in the internal timing. The adjustment occurs once per minute when CRSTRIM = `0`. The SIGN bit specifies whether to add cycles or to subtract them. The TRIMVAL<6:0> bits are used to specify by how many clock cycles to adjust. Each step in the TRIMVAL<6:0> value equates to adding or subtracting two clock pulses to or from the 32.768 kHz clock signal. This results in a correction of roughly 1.017 ppm per step when CRSTRIM = `0`. Setting TRIMVAL<6:0> to 0x00 disables digital trimming. Digital trimming also occurs while operating off the backup supply."

For additional information, please refer to the following application note, available at the corporate website (www.microchip.com):

- AN1365 – "*Recommended Usage of Microchip Serial RTCC Devices*" (DS00001365)

Tuning fork crystals are the most common type of crystals and are traditionally used with RTCC devices due to availability and low cost. The typical temperature curve for tuning fork crystals is shown in Figure 2.

FIGURE 2: PARABOLIC CURVE FOR TUNING FORK CRYSTALS



The accuracy of the crystal is acceptable around the 25°C temperature. Moving away from this point, the frequency deviation changes drastically. It is recommended that the internal calibration be used to improve the accuracy at other temperatures.

The following crystals have been tested and found to work with the MCP7941X family. The table below is given as design guidance and a starting point for crystal and capacitor selection.

Manufacturer	Part Number	Crystal Capacitance	CX1 Value	CX2 Value
Micro Crystal	CM7V-T1A	7 pF	10 pF	12 pF
Citizen Holdings Co., Ltd.	CM200S-32.768KDZB-UT	6 pF	10 pF	8 pF

Note: For more details on crystal solutions, refer to AN1519, "*Recommended Crystals for Microchip Stand-Alone Real-Time Clock/Calendar Devices*" (DS00001519).

The temperature effect on crystals, illustrated in Figure 2, is also detailed on Wikipedia® website (http://en.wikipedia.org).

"A crystal's frequency characteristic depends on the shape or 'cut' of the crystal. A tuning fork crystal is usually cut such that its frequency overtemperature is a parabolic curve centered around 25°C. This means that a tuning fork crystal oscillator resonates close to its target frequency at room temperature, but slows when the temperature either increases or decreases from room temperature. A common parabolic coefficient for a 32 kHz tuning fork crystal is 0.04 ppm/°C^2."

The parabolic dependence frequency vs. temperature is described in the following relations:

Relation 1A

$$f = f0 \times [1 - 0.04\ ppm/°C^2 \times (T-T0)^2]$$

Where:

f = frequency of the crystal

f0 = frequency at room temperature

T = ambient temperature

T0 = turnover point (room temperature(1))

Note 1: room temperature = 25°C usual value

The particular coefficient should be replaced in a more general manner:

Relation 1B

$$f = f0 \times [1 - Tc \times (T-T0)^2]$$

$$Tc = [0.030 \ldots 0.050]\ ppm/°C^2$$

Where:

Tc = thermal coefficient

Or, in another form:

Relation 2A

$$df = -Tc \times [(T-T0)^2]$$

Where:

df = frequency deviation

To calculate more easily inside the code (firmware), a particular thermal coefficient can be used, as indicated in Relation 2B.

Relation 2B

$$df = -TC \times [(T-T0)^2]/1000$$

$$TC = 1000 \times Tc$$

Where:

TC = thermal coefficient

The deviation of the frequency is shown in Relation 2C.

Relation 2C

$$df = -[(T-T0)^2]/KT$$

Where:

KT = thermal coefficient

The Wikipedia article on crystal oscillators (https://en.wikipedia.org/wiki/Crystal_oscillator) describes further:

"In a real application, this means that a clock built using a regular 32 kHz tuning fork crystal keeps good time at room temperature, but loses two minutes per year at 10°C above or below room temperature and loses 8 minutes per year at 20°C above or below room temperature due to the quartz crystal."

The code offers through `#define` directives the ability to choose the most correct value for both variables: turnover point and thermal coefficient.

T0	~	25°C
Tc	=	[0.030 - 0.050] ppm/°C^2
TC	=	[30 - 50] ppm/°C^2
KT	=	[20 ... 30] °C^2/ppm

Since one bit of the Calibration register adds 2 clocks/minute, it means that one bit will be approximately 1 ppm:

$$2/(60 \times 32{,}768) = 2/1{,}966{,}080 \sim 1 \text{ ppm}$$

Note: Keep in mind that the Calibration register must have negative values to increase the frequency and subtract pulses.

Considering the above relations and the previous Note, the value introduced in the Calibration register is obtained as indicated in Relation 3A.

Relation 3A

calib = -Tc x (dT^2)

Where:

dT = temperature drift

For accuracy and ease of use, Relation 3B will be used in the firmware.

Relation 3B

calib = -[TC x (dT^2)]/1000

Or, in another form, the calibration can be calculated based on the formula in Relation 3C.

Relation 3C

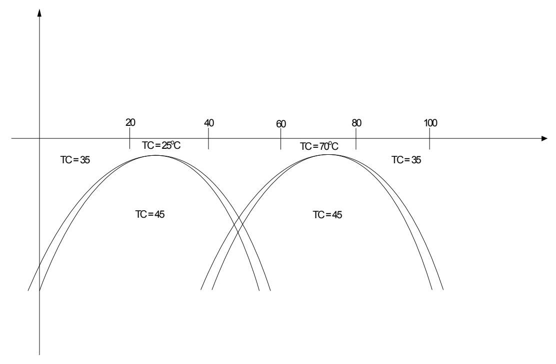
calib = -(dT^2)/KT

MATH RELATIONS AND THE PRECISION OF THE METHOD

As shown in Relation 2A, Relation 2B and Relation 2C, the math rule (frequency deviation versus temperature) describes for some crystals (tuning fork crystals) a parabolic curve in which the main coefficient of the parabola is negative. This means that the frequency reaches a maximum value in the turnover point (room temperature) and decreases for any other temperature value.

The dependence between the frequency deviation and temperature is illustrated in Figure 3, that shows parabolas related to several values for the turnover point and the thermal coefficient.

FIGURE 3: PARABOLIC CURVE



Since this parabola has a negative coefficient, the Calibration register must be set with a negative value (bit 7 = `1`, the last six bits are significant) in order to finally obtain the frequency versus temperature flat curve.

In other words, the Calibration register must have negative values in order to compensate the decrease of frequency of the crystal.

Accordingly,

calibration < 127

dT < $(127 \times 25)^{1/2}$ ~ 56°C (KT = 25°C^2/ppm)

Therefore, the ambient temperature must be in the range: [-30...+80]°C

More information about the compensation mechanism of the Calibration register can be found in the MCP7941X data sheet (DS20002266).

QUANTIZATION ERRORS

Approximation of numbers on small machines creates quantization errors. 8-bit machines are especially affected.

E_t represents the total error and is the sum of several possible errors, as shown in Relation 4.

Relation 4

$$E_t = E1 + E2 + E3 + ...$$

Where:

E1 = quantization of the thermal coefficient (KT)

E2 = truncation or rounding of the calibration value

E3 = error due to the imprecision of the MCP9800 temperature sensor

APPROXIMATION OF THE THERMAL COEFFICIENT (KT)

The first version of the code uses KT = 1/TC, where TC is given by manufacturers in the formula:

$$dF = -Tc \times dT^2$$

TC is expressed in ppm and is in the range [0.03... 0.05] ppm/°C^2.

Consequently, Relation 3A turns into Relation 3C:

calibration = -dT^2/KT

Where:

KT = [20 ... 30] °C^2/ppm

KT is calculated manually and only once by the user, starting from the constant TC = [0.03 ... 0.005] ppm/°C^2.

A source of errors is represented by the truncation at 'unsigned char' of the thermal coefficient (KT).

Tc = 0.039, therefore KT = 1/Tc = 25.6. (25 or 26)

A typical error can be seen in Equation 1:

EQUATION 1:

$$d\_cal = \frac{dT \times dT}{KT1} - \frac{dT \times dT}{KT2} = dT \times dT \times \left[\frac{1}{KT1} - \frac{1}{KT2}\right]$$

$$or$$

$$d\_cal = dT^2 \times \left[\frac{1}{25} - \frac{1}{25.5}\right] = dT^2 \times \frac{0.5}{625} = \frac{dT^2}{1250}$$

This means that for a 50°C temperature deviation, the error could reach 2 ppm.

(50^2/1250 = 2500/1250 = 2 ppm)

In order to better this deviation, another math algorithm could be used. The TC constant will be used instead of KT and dT is an unsigned long:

EQUATION 2:

$$dF = -Tc \times dT^2$$

$$calib = -Tc \times dT^2$$

$$Tc = [0.030 \ldots 0.050] \ ppm/^oC^2$$

$$calib = -\frac{(-TC \times dT^2)}{1000}$$

$$TC = [30 \ldots 50]$$

where TC = 1000 x TC

By using TC, the quantization of KT is avoided and a better precision is obtained. The two examples below use both methods.

EXAMPLE 1:

TC = 39 dT = 20

Method 1 (using TC)

calib1 = 39 x 400/1000 = 15.6 ~ 16

Method 2 (using KT) KT= 1/0.039 = 25.6 ~ 26

calib2 = 400/26 = 15.38 ~ 15

EXAMPLE 2:

TC = 39 dT = 50

Method 1 (using TC)

calib1 = 39 x 2500/1000 = 97.50 ~ 98

Method 2 (using KT) KT= 26

calib2 = 2500/26 = 96.15 ~ 96

CALIBRATION VALUE – TRUNCATION VERSUS ROUNDING

Truncation implies a constant negative offset of approximately 0.5 ppm, while rounding offers a flat curve of the error on the whole range of temperatures (+/- 0.5 ppm).

Accordingly, the rounding method is used in the firmware.

ERRORS DUE TO THE MCP9800 TEMPERATURE SENSOR

The last possible source of error is represented by the temperature sensor.

As stated in the related data sheet, the error for each temperature range is listed below:

25°C	=	+/-0.5°C
[-10...+085]	=	+/-1.0°C
[-10...+125]	=	+/-2.0°C
dT	=	Deviation of the ambient temperature
eT	=	Temperature error of the I^2C sensor

The error of the calibration value `d_cal` is shown in Equation 3:

EQUATION 3:

$$d\_cal = \left(\frac{TC}{1000}\right) \times [(dT + eT)^2 - dT^2]$$
$$= \left(\frac{TC}{1000}\right) \times [eT^2 + 2 \times dT \times eT]$$
$$= \left(\frac{40}{1000}\right) \times [1 + 100] \sim 4\ ppm$$

Where TC = 40 ppm/°C^2
dT = 50°C
eT = 1°C

The value of the error due to the temperature sensor is higher than the error due to the quantization of KT.

FIRMWARE DESCRIPTION

The new functions introduced by the application are:

- `void temp_compensation(void)` – may be included in any RTCC project to compensate a parabolic thermal drift of a tuning fork crystal. Starting from the basic variable '`unsigned int ADC_temp`' it calculates the final values for:
  - temperature
  - sign of temperature
  - temperature drift
  - calibration

This function is based on two methods, truncation or rounding.

- `unsigned int MCP9800_rdtemp(void)` – reads the ambient temperature on an I^2C bus. The format of the temperature sensor is the complement of 2 on 9, 10, 11, 12 bits.
- `void ini_MCP9800(void)` – initialization of the temperature sensor.
- `void per_mfp (void)` – measures (based on TMR0) the duration of one minute (value expressed in microseconds).

The most important of these functions is the temperature compensation function. The Firmware Code can be found in **Appendix B: "Firmware Code"**.

As stated in the math relations paragraph, there are slight differences between the two basic methods of calculation for the calibration value, truncation and rounding.

Truncation will offer a permanent negative offset (1 bit = 1 ppm), with an average value (in the whole temperature range) of -0.5 ppm.

Rounding will give an offset of +/- 0.5 ppm with an average value (in the whole temperature range) of approximately 0 ppm. Based only on this statement, it seems that rounding is better than truncation.

Slightly different values for the calibration are obtained through the two methods discussed earlier, truncation and rounding.

Drivers

Drivers are divided into 3 classes:

- LCD drivers
- RTCC's registers access drivers
- Temperature compensation functions

LCD Drivers

The application is implemented on a specific hardware, the PIC18 Explorer demo board. On this board it was important to reduce the number of GPIO pins used to access the LCD. Accessing the LCD is performed on a SPI bus (included in the MSSP1 module) through an auxiliary chip, the MCP23S17 SPI expander. The related drivers are:

- `wrcmnd_lcd (unsigned char cmnd_lcd)` (write command to LCD)
- `wrdata_lcd (unsigned char data_lcd)` (write data byte/character to LCD)
- `wrstr_lcd (const rom unsigned char *str_lcd)` (write to LCD a string stored in the flash)

Drivers to Access RTCC's Registers

Since MCP7941X is an I^2C RTCC, it will use the I^2C bus of the microcontroller (the MSSP1 module). Accordingly, the related drivers will be divided into two categories: basic I^2C drivers and RTCC drivers. They use as a control method the SPP1IF bit (flag) in the PIR1 register (interrupt flag of the MSSP1 module), read through polling and not through interrupts. The method represents an alternative to the classical "`i2c.h`" library, included in the C18 compiler.

FIGURE 4: FLOWCHART FOR A TYPICAL WRITE OPERATION (FOR A RANDOM BYTE ACCESS)

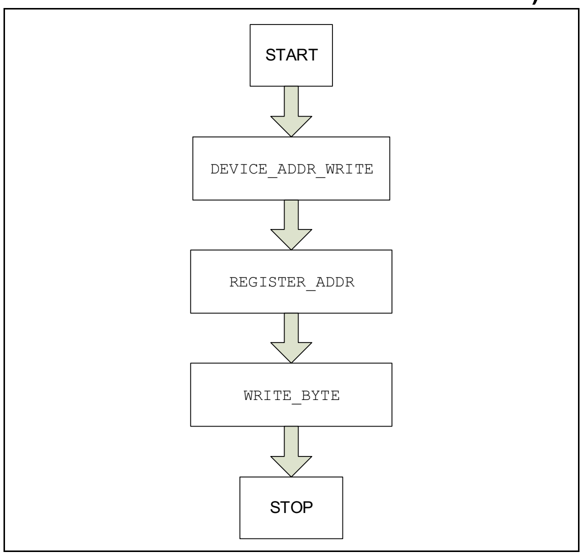
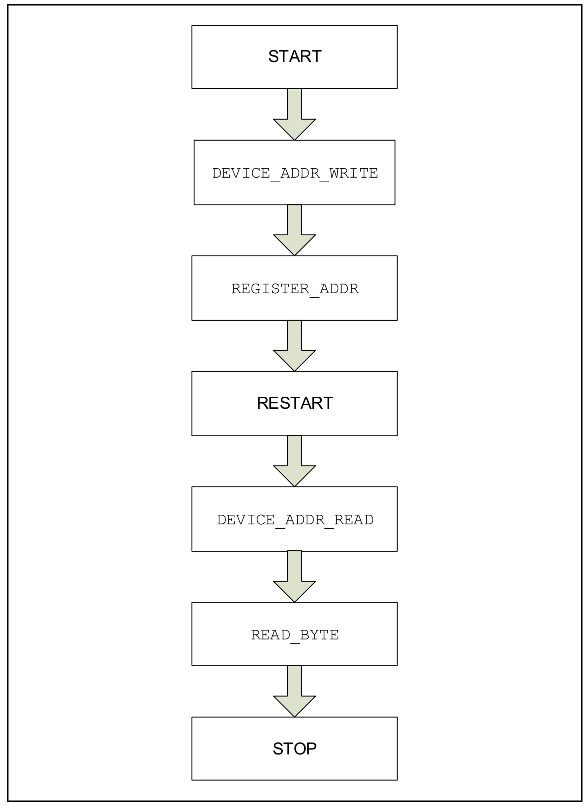


FIGURE 5: FLOWCHART FOR A TYPICAL READ OPERATION



ACCESSING THE RTCC'S REGISTERS

There are two basic functions for accessing the RTCC: one for writes and one for reads. They can be defined as: `void rtcc_wr (unsigned char time_var, unsigned char rtcc_reg), unsigned char rtcc_rd (unsigned char rtcc_reg).` Each of these two functions include error messages displayed on LEDs, which could signal when an operation is not acknowledged by the slave (RTCC).

EXAMPLE 3: FLOWCHART FOR WRITES TO THE RTCC

```
i2c_start()           ;//start I2C communication: SDA goes down while SCL remains high
i2c_wr(ADDR_RTCC_WRITE);//send the RTCC's address for write = 0xde
i2c_wr(rtcc_reg)      ;//send the register's address
i2c_wr(time_var)      ;//send data byte to the RTCC
i2c_stop()            ;//stop I2C communication: SDA goes high while SCL remains high
```

EXAMPLE 4: FLOWCHART FOR READS FROM THE RTCC

```
i2c_start()           ;//start I2C communication: SDA goes down while SCL remains high
i2c_wr(ADDR_RTCC_WRITE); //   send the RTCC's address for write = 0xde
i2c_wr(rtcc_reg)      ;  //   send the register's address
i2c_restart()         ;  //   switch to reads
i2c_wr(ADDR_RTCC_READ) ;//send the RTCC's address for read = 0xdf
i2c_rd()              ;//read the byte from the RTCC (register's content)
i2c_nack()            ;//NoACK from MCU to the RTCC (no more bytes to read)
i2c_stop()            ;  //   stop I2C communication: SDA goes high while SCL remains high
```

TESTS AND SIMULATIONS

During the development, the correctness of the math relations was tested through a simulation project. In the Temperature column there are pairs of temperatures. The frequency describes a parabola and accordingly the frequencies are symmetric around the turnover point. The turnover point (temp0) is usually 25°C. The results can be found in Table 1 and Table 2.

TABLE 1: CALIBRATION VERSUS TEMPERATURE – THE TRUNCATION METHOD

Temperature (°C)	Compensation -(ppm)		
25/25	0	0	0
26/24	0	0.03	0
27/23	0	0.16	0
28/22	0	0.35	0
29/21	0	0.62	0
30/20	0	0.98	0
31/19	1	1.40	1
32/18	1	1.91	1
33/17	2	2.49	2
34/16	3	3.16	3
35/15	3	3.90	3
36/14	4	4.72	4
37/13	5	5.62	5
38/12	6	6.59	6
39/11	7	7.64	7
40/10	8	8.78	8
41/09	9	9.98	9
42/08	11	11.27	11
43/07	12	12.64	12
44/06	14	14.08	14
45/05	45	15.60	15
46/04	17	17.12	17
47/03	18	18.88	18
48/02	20	20.63	20
49/01	22	22.46	22
50/00	24	24.38	24
51/-01	26	26.36	26
52/-02	28	28.43	28
53/-03	30	30.58	30
54/-04	32	32.80	32
55/-05	35	35.10	35
56/-06	37	37.48	37
57/-07	39	39.94	39
58/-08	42	42.47	42

TABLE 1: CALIBRATION VERSUS TEMPERATURE – THE TRUNCATION METHOD

Temperature (°C)	Compensation -(ppm)		
59/-09	45	45.08	45
60/-10	47	47.78	47
61/-11	50	50.54	50
62/-12	535	53.39	53
63/-13	56	56.32	56
64-14	59	59.32	59
65/-15	62	62.40	62
66/-16	65	65.56	65
67/-17	68	68.80	68
68/-18	72	72.11	72
69/-19	75	75.50	75
70/-20	78	78.98	78
71/-21	82	82.52	82
72/-22	86	86.15	86
73/-23	89	89.86	89
74/-24	93	93.64	93
75/-25	97	97.50	97
76/-26	101	101.44	101
77/-27	105	105.46	105
78/-28	109	109.56	109
80/-30	117	117.98	117
81/-31	122	122.30	122
82/-32	126	126.71	126

TABLE 2: CALIBRATION VERSUS TEMPERATURE – THE ROUNDING METHOD

Temperature (°C)	Compensation -(ppm)		
25/25	0	0	0
26/24	0	0.03	0
27/23	0	0.16	0
28/22	0	0.35	0
29/21	1	0.62	1
30/20	1	0.98	1
31/19	1	1.40	1
32/18	2	1.91	2
33/17	2	2.49	2
34/16	3	3.16	3
35/15	4	3.90	4
36/14	5	4.72	5
37/13	6	5.62	6
38/12	7	6.59	7
39/11	8	7.64	8
40/10	9	8.78	9
41/09	10	9.98	10
42/08	11	11.27	11
43/07	13	12.64	13
44/06	14	14.08	14
45/05	16	15.60	16
46/04	17	17.12	17
47/03	19	18.88	19
48/02	21	20.63	21
49/01	22	22.46	22
50/00	24	24.38	24
51/-01	26	26.36	26
52/-02	28	28.43	28
53/-03	31	30.58	31
54/-04	33	32.80	33
55/-05	35	35.10	35
56/-06	37	37.48	37
57/-07	40	39.94	40
58/-08	42	42.47	42
59/-09	45	45.08	45
60/-10	48	47.78	48
61/-11	51	50.54	51
62/-12	53	53.39	53
63/-13	56	56.32	56
64-14	59	59.32	59
65/-15	62	62.40	62

TABLE 2: CALIBRATION VERSUS TEMPERATURE – THE ROUNDING METHOD

Temperature (°C)	Compensation -(ppm)		
66/-16	66	65.56	66
67/-17	69	68.80	69
68/-18	72	72.11	72
69/-19	76	75.50	76
70/-20	79	78.98	79
71/-21	83	82.52	83
72/-22	86	86.15	86
73/-23	90	89.86	90
74/-24	94	93.64	94
75/-25	98	97.50	98
76/-26	101	101.44	101
77/-27	105	105.46	105
78/-28	110	109.56	110
79/-29	114	113.72	114
80/-30	118	117.98	118
81/-31	122	122.30	122
82/-32	127	126.71	127

SETUP OF THE APPLICATION

First of all, choose from the data sheet of the crystal's manufacturer the correct values for the turnover point and the parabolic coefficient. Some of the values should be tested for the following range of temperatures:

- <0°C
- [0°C - turnover point (25°C)]
- >turnover point (25°C)

These actions will test the MCP9800 temperature sensor. A final simulation test should include measurements of the clock frequency (MFP) delivered by the RTCC in order to observe the correct operation of the calibration mechanism. As mentioned in the data sheet, the calibration module adds or subtracts two pulses (in order to obtain a 1 ppm precision) of the main frequency of the crystal (32,768 Hz), with every bit of the Calibration register. The calibration module performs it only once per minute. The related test results can be found in Table 3. The column titled 1MIN T32K shows how many pulses of 32,768 Hz are in one minute. The column titled DIFF_1MIN (25°C) T32K shows how many pulses of 32,768 Hz will be subtracted from one minute compared to the number of pulses contained in one minute at 25°C.

TABLE 3: TEST RESULTS

TEMP	calib (-)	1MIN (µsec)	1MIN T32K	TEMP	calib (-)	1MIN (µsec)	1MIN T32K	DIFF_1MIN (25°C) T32K
0	24	59,992,369	1,966,963	50	24	59,992,376	1,966,963	48 = 2 x 24
1	22	59,992,492	1,966,967	49	22	59,992,495	1,966,967	44 = 2 x 22
2	21	59,992,554	1,966,969	48	21	59,992,559	1,966,969	42 = 2 X 21
3	19	59,992,677	1,966,973	47	19	59,992,681	1,966,973	38 = 2 x 19
4	17	59,992,800	1,966,977	46	17	59,992,803	1,966,977	34 = 2 x 17
5	16	59,992,862	1,966,979	45	16	59,992,863	1,966,979	32 = 2 x 16
6	14	59,992,984	1,966,983	44	14	59,992,985	1,966,983	28 = 2 x 14
7	13	59,993,046	1,966,985	43	13	59,993,047	1,966,985	26 = 2 x 13
8	11	59,993,169	1,966,989	42	11	59,993,170	1,966,989	22 = 2 x 11
9	10	59,993,230	1,966,991	41	10	59,993,232	1,966,991	20 = 2 x 10
10	9	59,993,291	1,966,993	40	9	59,993,294	1,966,993	18 = 2 x 09
11	8	59,993,352	1,966,995	39	8	59,993,352	1,966,995	16 = 2 x 08
12	7	59,993,415	1,966,997	38	7	59,993,413	1,966,997	14 = 2 x 07
13	6	59,993,475	1,966,999	37	6	59,993,474	1,966,999	12 = 2 x 06
14	5	59,993,534	1,967,001	36	5	59,993,535	1,967,001	10 = 2 x 05
15	4	59,993,595	1,967,003	35	4	59,993,597	1,967,003	08 = 2 x 04
16	3	59,993,655	1,967,005	34	3	59,993,657	1,967,005	06 = 2 x 03
17	2	59,993,717	1,967,007	33	2	59,993,715	1,967,007	04 = 2 x 02
18	2	59,993,716	1,967,007	32	2	59,993,715	1,967,007	04 = 2 x 02
19	1	59,993,775	1,967,009	31	1	59,993,775	1,967,009	02 = 2 x 01
20	1	59,993,775	1,967,009	30	1	59,993,775	1,967,009	02 = 2 x 01
21	1	59,993,775	1,967,009	29	1	59,993,775	1,967,009	02 = 2 x 01
22	0	59,993,840	1,967,011	28	0	59,993,840	1,967,011	00 = 2 x 00
23	0	59,993,840	1,967,011	27	0	59,993,841	1,967,011	00 = 2 x 00
24	0	59,993,840	1,967,011	26	0	59,993,840	1,967,011	00 = 2 x 00
25	0	59,993,839	1,967,011	25	0	59,993,839	1,967,011	00 = 2 x 00
-1	26	59,992,253	1,966,959	51	26	59,992,253	1,966,959	52 = 2 x 26
-2	28	59,992,132	1,966,955	52	28	59,992,131	1,966,955	56 = 2 x 28
-3	31	59,991,948	1,966,949	53	31	59,991,947	1,966,949	62 = 2 x 31

TABLE 3: TEST RESULTS (CONTINUED)

TEMP	calib (-)	1MIN (µsec)	1MIN T32K	TEMP	calib (-)	1MIN (µsec)	1MIN T32K	DIFF_1MIN (25°C) T32K
-4	33	59,991,826	1,966,945	54	33	59,991,825	1,966,945	66 = 2 x 33
-5	35	59,991,705	1,966,941	55	35	59,991,703	1,966,941	70 = 2 x 35
-6	37	59,991,583	1,966,937	56	37	59,991,581	1,966,937	74 = 2 x 37
-7	40	59,991,400	1,966,931	57	40	59,991,399	1,966,931	80 = 2 x 40
-8	42	59,991,278	1,966,927	58	42	59,991,277	1,966,927	84 = 2 x 42
-9	45	59,991,094	1,966,921	59	45	59,991,094	1,966,921	90 = 2 x 45
-10	48	59,990,912	1,966,915	60	48	59,990,912	1,966,915	96 = 2 x 48
-11	51	59,990,729	1,966,909	61	51	59,990,729	1,966,909	102 = 2 x 51
-12	53	59,990,606	1,966,905	62	53	59,990,607	1,966,905	106 = 2 x 53
-13	56	59,990,424	1,966,899	63	56	59,990,424	1,966,899	112 = 2 x 56
-14	59	59,990,241	1,966,893	64	59	59,990,240	1,966,893	118 = 2 x 59
-15	62	59,990,057	1,966,887	65	62	59,990,057	1,966,887	124 = 2 x 62
-16	66	59,989,815	1,966,879	66	66	59,989,813	1,966,879	132 = 2 x 66
-17	69	59,989,631	1,966,873	67	69	59,989,629	1,966,873	138 = 2 x 69
-18	72	59,989,449	1,966,867	68	72	59,989,446	1,966,867	144 = 2 x 72
-19	76	59,989,204	1,966,859	69	76	59,989,203	1,966,859	152 = 2 x 76
-20	79	59,989,022	1,966,853	70	79	59,989,020	1,966,853	158 = 2 x 79
-21	83	59,988,777	1,966,845	71	83	59,988,774	1,966,845	166 = 2 x 83
-22	86	59,988,593	1,966,839	72	86	59,988,592	1,966,839	172 = 2 x 86
-23	90	59,988,350	1,966,831	73	90	59,988,347	1,966,831	180 = 2 x 90
-24	94	59,988,105	1,966,823	74	94	59,988,102	1,966,823	188 = 2 x 94
-25	98	59,987,862	1,966,815	75	98	59,987,859	1,966,815	196 = 2 x 98
-26	101	59,987,679	1,966,809	76	101	59,987,676	1,966,809	202 = 2 x 101
-27	105	59,987,435	1,966,801	77	105	59,987,430	1,966,801	210 = 2 x 105
-28	110	59,987,130	1,966,791	78	110	59,987,125	1,966,791	220 = 2 x 110
-29	114	59,986,886	1,966,783	79	114	59,986,881	1,966,783	228 = 2 x 114
-30	118	59,986,642	1,966,775	80	118	59,986,637	1,966,775	236 = 2 x 118
-31	122	59,986,397	1,966,767	81	122	59,986,393	1,966,767	244 = 2 x 122
-32	127	59,986,091	1,966,757	82	127	59,986,088	1,966,757	254 = 2 x 127

CONCLUSION

This application note presents how to compensate the parabolic thermal drift of tuning fork crystals using the Calibration register of Microchip's I^2C RTCC, MCP7941X.

The project is performed on a PIC18 Explorer demo board, using the on-board resources: LCD (accessed through the SPI bus) and push buttons. The AC164140 PICtail daughter board (including an I^2C RTCC and an I^2C temperature sensor) is used. The code (drivers and main function) is written in C, using the C18 compiler. The target microcontroller is PIC18F87J11.

REFERENCES

1. MCP7941X Data Sheet – "*Battery-Backed I²C Real-Time Clock/Calendar with SRAM, EEPROM and Protected EEPROM*" (DS20002266)
2. AN1365 – *"Recommended Usage of Microchip Serial RTCC Devices"* (DS00001365)
3. AN1519 – *"Recommended Crystals for Microchip Stand-Alone Real-Time Clock/Calendar Devices"* (DS00001519)
4. http://en.wikipedia.org

APPENDIX A: REVISION HISTORY

Revision A (11/2011)

Initial release of this document.

Revision B (09/2016)

Updated bit and register names to match new data sheet format; Updated some paragraphs referring to the latest version of the MCP7941X data sheet; Updated overall content for improved clarity.

APPENDIX B: FIRMWARE CODE

```
    void temp_compensation(void) {      //  SETS THE INTERNAL FREQUENCY ACCORDING THE TEMP,
                                        //  THROUGH THE CALIBRATION REGISTER
                                        //  this is the most important function of the code.
                                        //  it obtains the 4 main values : the ambient
                                        //  temperature 'temp', it's sign , the difference
                                        //  |dT|=|temp-temp0| and the calibration value
                                        //  the 'calib' will be always negative, in order to
                                        //  increase the frequency around the turn over point
                                        //  |temp| and sign will be printed on the LCD,
                                        //  |dT| will help to compensate the temp drift, through
                                        //  the calibration register.
   unsigned int ADC_res               ; //  reserve variable to store ADC_temp
// ADC_temp  = MCP9800_rdtemp()       ; //  obtain the 16bit temperature from the sensor
   ADC_res = ADC_temp                 ; //  store the ADC result
   if((ADC_temp&0x8000)==0x0000)        //  if temp = plus,
   { sgntemp = 0x00 ; }                 //  build the extended sign
   else { sgntemp = 0x01 ;              //  if temp = minus : build the extended sign,
          ADC_temp = (~ADC_temp)+1 ; }  //  2 is complement of the ADC value
   temp = (ADC_temp>>7)&0xff          ; //  build the 8bits temperature variable
   if(!sgntemp) {                       //  if a positive temperature
    if(temp >= temp0)
    { dT = temp - temp0            ;  }
    else { dT = temp0 - temp   ;  }  }  //  build | temp - temp0 |
    else { dT = temp0 + temp   ;  }     //  if a negative temperature, dT = temp0+temp
                                        //  once dT is calculated, the final formula
   calib = (TC*(dT*dT))/1000          ; //  unsigned char calibration value
   if(((TC*(dT*dT))%1000)>=500)
         { calib++  ; }                 //  rounding instead truncation
   rtcc_wr(calib+0x80,OSCTRIM)        ; //  write in the calibration register the
                                        //  compensation value = -(TC/1000)*dT^2(always '-')
   ADC_temp = ADC_res                 ; //  restore the ADC value for further use :
                                      } //  LCD functions & WHILE LOOP
```

Note: The function above belongs to the simulation projects, which replace the reads from the temperature sensor by virtual temperature samples. The real drive of the function will use real samples of temperature, taken from the MCP9800.

ISBN: 978-1-5224-0932-8

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Guangzhou
Tel: 86-20-8755-8029

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

06/23/16