TERIDIAN SEMICONDUCTOR CORP.
A Maxim Integrated Products Brand

71M6521
Energy Meter IC

APPLICATION NOTE

AN_6521_035 **APRIL 2009**

Real Time Clock Compensation

This document describes how to use software to compensate the real time clock (RTC) in Teridian meter chips. The sample code discussed is from the demonstration code for the 71M6521FE, but similar principles may be used with the 71M6521DE, 71M6523, 71M6511, 71M6513 and other Teridian Meter products.

The Task

The real-time clock (RTC) in Teridian energy meter ICs is based on the external crystal connected to the XIN and XOUT pins of the on-chip oscillator. Most applications require accuracy that is better than an uncompensated clock crystal can provide. Commercially available crystals will have a slight deviation from the target frequency at room temperature. In addition, the crystal frequency will change with temperature following a mostly quadratic function. Therefore, practical real-time-clocks require some sort of compensation to get adequate accuracy.

The RTC in the 71M6521DE/FE

The real-time clocks in the Teridian 71M6521 Energy Metering chips consist of an oscillator and divider chain that keeps track of the current time of day and date, as long as main power or battery power is applied to the chip. Once set, the RTC tracks real time and date at the accuracy provided by the 32-kHz oscillator.

The hardware of the 71M6521DE/FE permits the seconds counter to be corrected by incrementing or decrementing it using the *RTC_INC_SEC* or *RTC_DEC_SEC* registers of the I/O RAM.

Theory of Operation

Typical quartz crystals are available in various accuracy grades. For example, a 20 PPM crystal will not have more than ±20 PPM deviation from the nominal frequency at room temperature. Electricity meters will usually be exposed to warm and cold temperatures, especially when mounted outside buildings. At temperatures deviating from room temperature, even the best crystals change their frequency, as shown in Figure 1. Regardless of the initial accuracy, the crystals will always have a pronounced quadratic characteristic of the frequency over temperature. The coefficient describing this relationship is in the range from 3.5 to $4.0 * 10^{-8}$/°C for most crystals.

Another crystal parameter is the inversion temperature (T_i), i.e. the temperature where the frequency peaks. T_i is normally 25°C.

Teridian offers a solution that uses software to compensate the RTC, permitting quadratic compensation for temperature-induced drift. This solution is purely digital, with excellent accuracy and stability. It saves space on the integrated circuit, providing very good value compared to more expensive solutions, such as temperature-compensated oscillators, digital drift compensation logic, or programmable loading capacitors.

The digital compensation uses the *RTC_INC_SEC* or *RTC_DEC_SEC* registers of the I/O RAM to occasionally correct the seconds counter. This correction occurs when the calculated deviation of the clock has reached one second. The compensation code continuously adds up the expected fractional error, until it equals one second, and then adds or subtracts one second of error to or from the RTC by incrementing or decrementing the seconds counter. This mode of operation is shown in Figure 2. While this compensation method works fine, it is better to correct the RTC when the error exceeds ½ of a second. That way, the deviation from ideal time is always less than 500ms (see Figure 3). Recent Demo Codes for the 71M6521 correct the RTC as soon as the projected deviation of the time reaches +½ or -½ seconds.

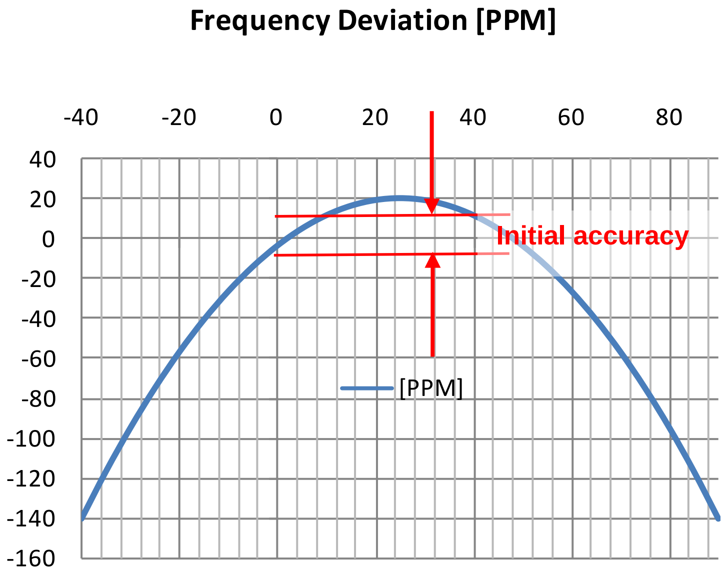

Figure 1: Crystal Frequency over Temperature

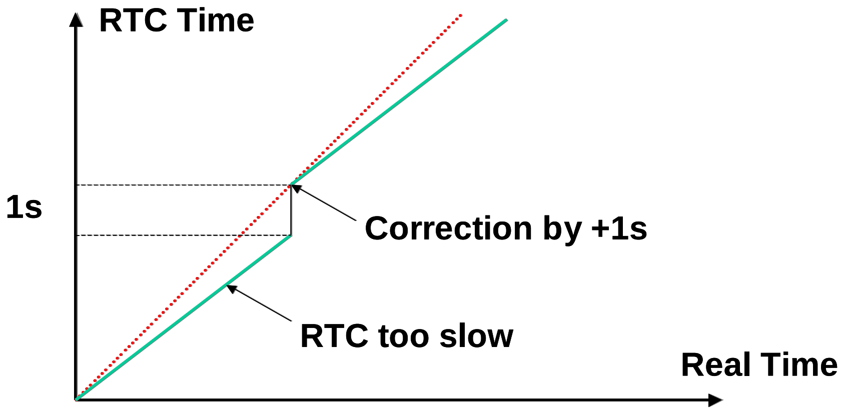

Figure 2: RTC Correction when Error Exceeds 1 Second

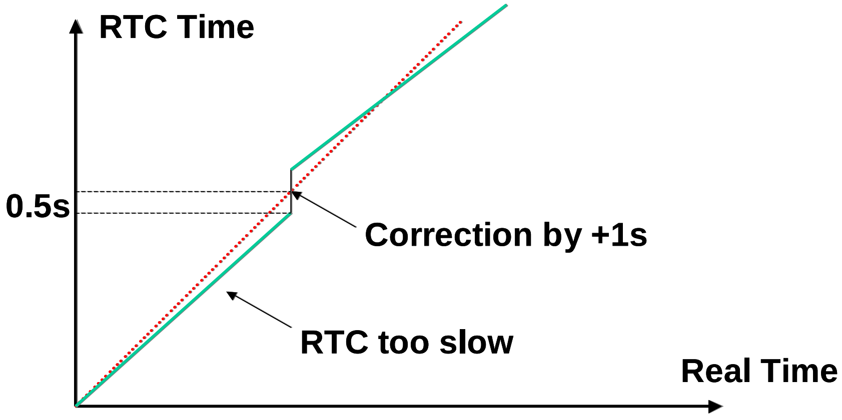

Figure 3: RTC Correction when Error Exceeds 0.5 Second

Digital compensation is active while the meter is operating with mains power on. But even when a meter has been completely turned off, and power to the oscillator and RTC has been supplied by a battery, the compensation software can calculate the time period that the meter was off, and apply the base drift compensation.

During manufacturing, an automated calibration process should be used to obtain good quality clock calibrations.

Design and Coding

Calculating the Error

Due to the quadratic behavior of the crystal over temperature, the Demo Code calculates the error using a quadratic equation:

`trim_value` (one second of correction in parts per billion) = $100*Y\_CAL0 + T*10*Y\_CAL1 + T^2 * 100 *Y\_CAL2$.

The coefficient *Y_CAL1* is included in order to compensate for linear deviations with temperature that can be caused by components other than the crystal, such as the capacitors at the XIN and XOUT pins.

This calculation is located in `RTC_Compensation()` in io\rtc.c. `trim_value` is in meter\meter.h in the nonvolatile register data (`Totals.Acc.T_trim_value`). `RTC_Compensation()` is called by `meter_run ()` when the CE has new data, and thus a new temperature. `meter_run()` is in meter\meter.c and is called very frequently by the main loop in main\main.c.

The error is the signed nanoseconds of error per second (parts per billion), so `trim_value` is a signed 32-bit number.

Storing `trim_value` must be done when interrupts are disabled. Otherwise, the one-second interrupt of the RTC (which adds up the error) can get an incorrect value. Therefore, `trim_value` is stored only in `RTC_Adjust_Trim()` of io\rtc.c.

T is the difference between the current, measured temperature and the temperature at which the temperature measurement and clock was calibrated. In the Demo Code, the variable is "`deltaT`" in meter\meter.h, and the LSB is 0.1C. `deltaT` is calculated by measuring the output of the on-chip temperature sensor with the ADC of the chip. Once calibrated, this output is very accurate and repeatable.

Y_CAL0 is the base crystal drift at every temperature. This is usually measured and set when the meter is calibrated (see Calibration, below). It is located in `Totals.Parms.T_Y_Cal_Deg0`, of meter\meter.h

Y_CAL1 and *Y_CAL2* describe the quadratic curve of a crystal's frequency drift over temperature. This may be established by measuring a statistically significant selection of units assembled with the same production lot of crystals. Alternatively, the temperature curve may be specified and controlled by the crystal manufacturer. Both variables are in `Totals.Parms.T_Y_Cal_Deg1` and `Totals.Parms.T_Y_Cal_Deg2`, of meter\meter.h

Adding the Error

The error accumulates in two variables: `second_count`, and `trim_count.` Normally, the adjustments are so slow that `second_count` is not needed. However, when the meter has been turned off for a long time, many seconds of adjustment must occur after it powers up.

`trim_count` is the fraction of a second of error. The long-term error needed in the unit can be expressed in parts per billion ($1/1x10^{-9}$), so `trim_count` is a 32-bit counter. `trim_count` is nonvolatile, saved by the Demo Code activated in case of a sag event. The variable is in `Totals.Acc.trim_count`, of meter\meter.h

`second_count` contains the accumulated seconds of error. It is also nonvolatile, and in `Totals.Acc.second_count` in meter\meter.h

The error must be added at equal intervals in real time. In the Demo Code, it is added once per second, in the one-second interrupt of the RTC (See `rtc_isr()` in io\rtc.c).

When `trim_count` has more than a half billion ($500*10^6$) or less than a negative half billion ($–500*10^6$) nanoseconds of error, it adds or subtracts one to `second_count`, and subtracts $+1*10^9$ or $–1*10^9$ from `trim_count`.

When `second_count` is nonzero, the RTC one-second interrupt increments or decrements the RTC's second counter, and subtracts or adds one to the `second_count`.

Adjusting when the Meter is off

The Demo Code can compensate for the time when the meter was turned off. To do this, it must measure the number of seconds that the meter was turned off.

To record the time that power fails, the meter frequently reads the clock and stores this in memory that is saved with the power registers.

When the meter restarts, the difference between the power-failure time and the current time is the number of seconds that the meter was off. The code must cope with situations in which the EEPROM is empty, the power-failure time is invalid, or the current time is invalid. The Demo Code for this is `RTC_reset()` in io\rtc.c. This is called from start-up code in `main()` before the RTC is read anywhere else.

The number of seconds that the meter was off is multiplied by `trim_value` and divided by a billion. This yields the number of seconds of adjustment, and is stored to the variable `second_count`. The remainder is multiplied by a billion and stored in `trim_count`.

If a calibrated meter is turned off and shelved, several thousand seconds of error from crystal drift may accumulate. When power returns, the clock must be incremented no more often than once every second. The Demo Code has logic that accumulates 32,767 (2^{15}) seconds of corrections, and then increments the real-time-clock every other second (to be conservative). This permits a storage time greater than five years.

Coping With Power Failures

Since the main power can fail at any point, the time of power failure, RTC error and accumulated error have to be non-volatile. Two copies have to be saved, so that if power fails while calculating one copy, the other copy remains valid.

In the Demo Code, the logic to save data is handled by the power-register logic in that calls `LRC_Calc_Nvr()` in `meter_run().` The clock variables are stored in the same data as the power registers, `Totals.Acc`, and saved during a sag event: See `ce_busy_isr()` in meter\ce.c The variables are validated and restored by `meter_initialize()` in meter\meter.c, called from the initialization in `main()`.

The nonvolatile variables for the clock are:

- `RTC_COPY` (This is the copy of the real time clock made each time the CE's data is added to the registers. After a power-failure, this is the time of the power failure).
- `trim_value` (one second of error in parts per billion, signed),
- `trim_count` (the accumulated fraction of one second in parts per billion, signed)
- `second_count` (the accumulated seconds of error, signed)

Calibration

The clock, meter and temperature measurement should all be calibrated at the same time, at the same temperature. This is because the temperature measurement is used to compensate both the RTC and the metrology portion (Wh, VARh) of the meter.

The calibration should occur in a location with a known and constant temperature, after the newly manufactured unit has been sufficiently soaked at that temperature. The IC uses its ADC to measure an internal junction temperature. This is very repeatable, but has an offset that varies based on manufacturing parameters.

A one-second square wave is the recommended method to measure the clock frequency drift. The basic scheme is to toggle a DIO as the first instruction of the RTC interrupt. To eliminate timing uncertainties, the calibration for the clock should set a special mode: The RTC's one-second interrupt should become the highest priority in the system. This is accomplished by setting *IPH* and *IPL* to 0x20, so that external interrupt 6, the RTC and xfer_busy, have the highest priority. Then, the CE should be disabled so that no xfer_busy interrupt can occur.

This one-second square wave can be emitted by a DIO normally used for LCD or pulse output, so that no I/O needs to be permanently committed.

The square wave from the DIO should be measured with a precision frequency counter. The measurement can then be used to calculate and set the *Y_CAL0* value. Most frequency counters with sufficient accuracy need at least an internal crystal oven, and a periodic recalibration to a national standard to adjust for aging drift in the frequency counter. See your frequency counter's documentation for more information.

Y_CAL1 and *Y_CAL2* describe the quadratic curve of a crystal's frequency drift over temperature. This may be established by measuring a statistically significant selection of units constructed with the same lot of crystals. Alternatively, the temperature curve may be specified and controlled by the crystal manufacturer.

After the square-wave is measured, the meter can then be reset to leave the clock calibration mode.

Depending on the exact design of the test and calibration system, there may be systematic errors that cause asymmetry or offset in the square-wave measurement. The designer can compensate for these by manually measuring and qualifying a meter, and then comparing the test system's clock calibration to the manual clock calibration. The calibration system must then be qualified by assuring that the values it assigns actually result in accurate clock adjustments in a statistically significant sample of production meters.

Precautions for Meter Production

At first sight it seems like meter RTC calibration is trivial. This is far from true. Process, part selection and other criteria have to be considered carefully when planning the RTC calibration process. This section lists a few criteria that are important for RTC accuracy.

Crystal Accuracy

When examining the parabolic coefficient of crystals, one will notice that it is often specified with a certain tolerance. For example, a well-known manufacturer specifies K as 3.5 ± 0.8 * 10^{-8}/°C^2, which is roughly 22%, for some of their crystals. This rather large range is a lot-to-lot variation. When examining crystals of the same lot, the K coefficient can be far more repeatable, as shown in Figure 4.

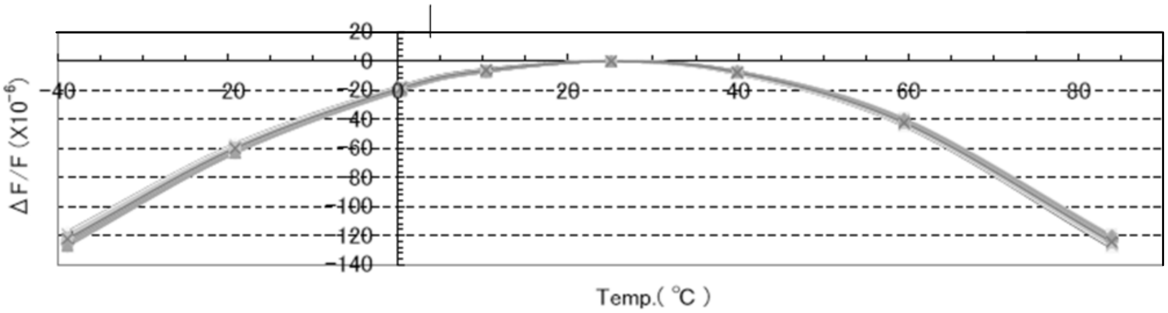

Figure 4: Crystal Accuracy over Temperature

Figure 4 shows that the deviation between individual crystals does not exceed ± 10 PPM, even at temperature extremes, which would make very good accuracy possible for a manufactured meter. At 10 PPM, the daily deviation of the RTC would only 0.86 seconds. This accuracy is achievable because the parabolic coefficient of this crystal lot varies only between 3.271 * 10^{-8}/°C^2 and 3.332 * 10^{-8}/°C^2, which is only a ± 0.93% deviation from the average of 3.3015 * 10^{-8}/°C^2. In addition, the inversion temperature T_i has very low variation (maximum 25.82 °C, minimum 23.41 °C).

Reality looks a bit different for meters made with commercially available crystals. Generally, they will most likely not be available as lot-sorted parts, and in addition, the inversion temperature (T_i) can vary by as much as ± 5°C.

Figure 5 shows the effect of a mismatch between nominal crystal parameters and real parameters. For this graph, the K coefficient was assumed to be 15% higher than nominal, and the inversion temperature was shifted by 3°C. The difference between the crystal frequency based on compensation using the nominal parameters and the frequency generated by the real crystal is plotted over temperature. The largest deviation is at -40°C, where -41.4 PPM result in an error of roughly 3.6 seconds per day.

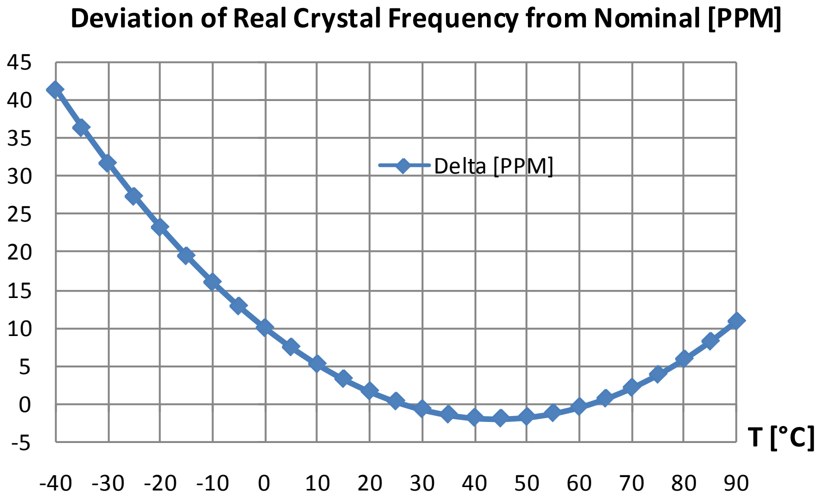

Figure 5: Deviation of Crystal Accuracy over Temperature

Handling and Soldering Precautions

Below is a list of several precautions that apply to the mounting/soldering of crystals:

- Do not bend leads. If bending the leads is absolutely required, it should be done by securing the leads close to the crystal body with pliers.
- Do not solder the case (of can-type crystals) since it will reduce the quality of the vacuum seal (which means the part could pull moisture).
- Do not pull the leads when handling the crystal.
- Some manufacturers strongly discourage ultra-sonic cleaning, since the frequency used by the cleaner can be close to the crystal resonance and can cause damage.
- At the same time, removal of all solder and flux residue is a must, which can be done with alcohol.
- So called "no clean solder" should not be used. The residue left by this material is conductive and may inhibit proper crystal operation.
- Some crystals tend to absorb moisture when washed in water. The way to tell this is to measure the resistance across the pins. After absorbing enough water, the measurement will show well below 1 MΩ. This can be fixed by baking the parts.

Handling and process precautions specified by the individual crystal manufacturers should be carefully considered in order to assure good RTC accuracy.

Revision History

Revision	**Date**	**Description**
Rev. 1.0	01/08/2007	First publication.
Rev. 1.1	05/02/2007	Added Figures 1 and 2.
Rev. 1.2	08/25/2008	Corrected caption for Figure 2. Added description for *Y_CAL1* coefficient. Added Revision History Table. Changed references to 71M6521DE and FE. Updated Teridian street address.
Rev. 1.3	04/17/2009	Added Figure 1. Added explanation of initial accuracy and quadratic characteristics. Added new section “Precautions for Meter Production”.

TERIDIAN Semiconductor Corp., 6440 Oak Canyon Rd., Suite 100, Irvine, CA 92618

TEL (714) 508-8800, FAX (714) 508-8877, http://www.teridian.com