KEELOQ® Encoders Oscillator Calibration

Author: Lucio Di Jasio
Microchip Technology Inc.

OVERVIEW

Several KEELOQ Encoders of recent introduction, offer the ability to calibrate the internal RC clock oscillator, upon which all the device timings are based. At the time of writing this Application Note, there are five KEELOQ devices offering this feature:

- HCS101 - fixed code encoder
- HCS201 - low cost hopping code encoder
- HCS362 - advanced hopping code encoder
- HCS410 - hopping code transcoder
- HCS412 - hopping code transcoder

While the respective Encoder Data Sheets show "where" the calibration value is stored in the device EEPROM, this Application Note will concentrate on "how to" determine the best calibration value.

WHY CALIBRATION?

One of the main advantages of using a KEELOQ Encoder (or Transcoder) in a remote control application has always been the extremely high level of integration offered. In fact, in most remote control applications, the actual circuit is reduced to:

- The HCS encoder (typically in a small 8-pin SOIC package)
- Several buttons
- An LED
- The battery
- A single stage (single transistor) RF transmitter

The key to this simplicity and convenience lies in:

- HCS encoders that incorporate EEPROM memory
- Voltage regulation and low voltage detect circuitry
- Debouncing logic
- A convenient RC clock oscillator.

Typically, this oscillator provides the clock frequency with a wide tolerance (up to -30%/+50%) and that reflects in ample tolerances on the transmission baud rate, LED flashing and, in general, on most device timings. Further, being a simple RC clock circuit, it is affected by battery voltage and temperature. In most applications, however, the convenience of being able to dispense with the cost (and space) of a crystal or a ceramic resonator far exceeds the inconvenience of such loose timing tolerances.

In fact, all KEELOQ Decoders and Application Notes presented so far have been designed to overcome this problem without limitation.

Still, there are special reasons to look for a better oscillator tolerance. These reasons are:

- wider baud rate tolerance means wider RF bandwidth
- in low power applications (decoders), the receiver software calibration abilities imply higher frequency clocks; therefore, higher power consumption
- in multi-tasking applications, an interrupt based receiver requires more time wasted in the thread
- in general, more code and testing is required in the development of a decoder to consider all possible operating conditions when the tolerance range is (so) wide

CALIBRATION BASICS

The basic calibration mechanism implemented in the new KEELOQ encoders is quite straightforward.

Four bits of the 16-bit device Configuration Word directly control the internal RC oscillator frequency.

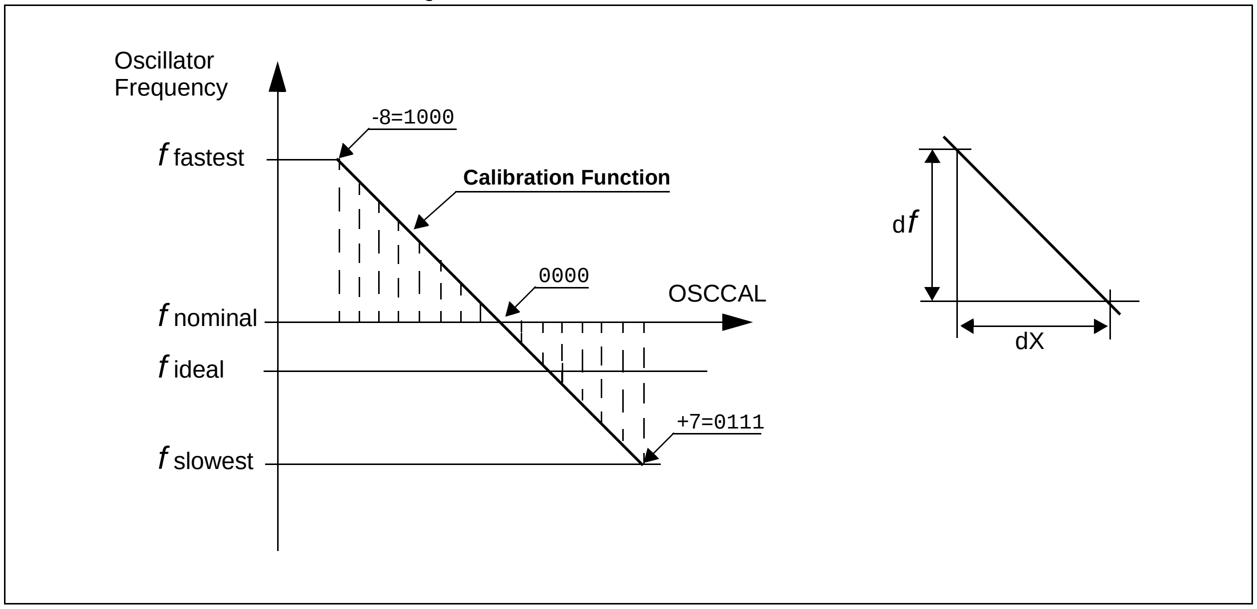
These four bits [OSC_0 through OSC_3] can be interpreted as a small signed integer (often referred to as OSCCAL) covering a range of 16 possible values from -8 to +7 as illustrated in Table 1.

TABLE 1: OSCCAL BITS

Decimal	Binary	Frequency
+7	`0111`	slowest
+6	`0110`	—
...	`...`	...
+2	`0010`	—
+1	`0001`	slower
0	`0000`	nominal
-1	`1111`	faster
-2	`1110`	—
...	`...`	...
-7	`1001`	—
-8	`1000`	fastest

Setting the oscillator calibration bits [OSC_0 through OSC_3] to `0000` makes the internal oscillator operate at a frequency that is considered the "nominal" value, corresponding to the frequency that a bulk erased (blank) part would present. Higher positive values have the effect of reducing the oscillator frequency, while lower, negative values correspondingly increase it. The graph of OSCCAL versus Frequency is shown in Figure 1.

FIGURE 1: OSCCAL vs. FREQUENCY GRAPH



The graph in Figure 1 represents a simplification. In fact, the relationship between the OSCCAL value and the oscillator frequency is not exactly linear, but for all practical purposes (when aiming at an accuracy of ±10%), we will consider it to be so.

There are a few more considerations to make before proceeding into the specific details of the devices. First of all, neither the graph in Figure 1 nor the Data Sheets show any absolute frequency reference values. Neither the *f* fastest frequency nor the *f* slowest frequency are given (in fact, we would not need any calibration if the two values were fixed and known). Further, the slope of the Calibration Function (d*f*/dX) is not given. In other words, d*f* is not known (when dX = 1 bit).

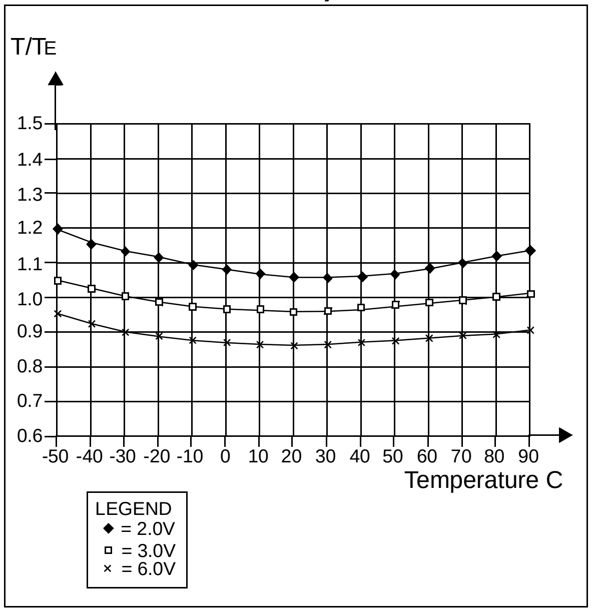
Finally, all our frequency references will vary slightly with temperature and voltage, as shown in Figure 2.

There are however, some things we do know . Production testing screens parts guarantee us that *f* ideal (our optimal calibration point) is, in fact, achievable:

f slowest < *f* ideal < *f* fastest

It is also given that d*f* is sufficiently small to allow us to achieve a calibration of ±10% or better.

FIGURE 2: TYPICAL PERIOD vs. TEMPERATURE (AND VOLTAGE) GRAPH



BRUTE FORCE

The simplest method to achieve calibration, is by 'brute force'. That is, by repeatedly trying out all possible values of OSCCAL. This can be simply illustrated in the following pseudo code segment:

```
OSCCAL = -8;
loop
    program part
    measure oscillator frequency F
    if F<Fideal then exit loop; //success
    increment OSCCAL;
    if OSCCAL>7 then exit loop; //failure
endloop
```

The main problem with this approach is that it takes an average of 8 programming and measurement loops to get a single part calibrated. The worst case requires 16 loops. It is obvious that this method is very inefficient and, therefore, probably unacceptable for volume production.

BINARY SEARCH

Taking inspiration from search algorithms theory, we can try and apply another simple technique: binary search. Starting from a midpoint, we repeatedly program the part and verify the oscillator frequency. We compare it with the optimal value, and if the frequency is not close enough to the optimal value, we increment or decrement OSCCAL by an amount (STEP), and loop. The STEP length is reduced in half at every loop. The method can be well illustrated by the following pseudo code segment:

```
OSCCAL = 0;//start from mid point
STEP = 8;
loop
    program part
    measure oscillator frequency F
    Delta = F-Fideal;
    if abs(Delta) < Tolerance then
        exit loop; // success
    else if Delta <0 then
        subtract STEP from OSCCAL
    else
        add STEP to OSCCAL
    endif
    if STEP = 1 then
        exit loop
    else
        divide STEP by two
end loop
```

where `Tolerance` is the actual frequency tolerance target that we want to achieve (i.e., 10% Fideal).

This algorithm offers a worst case of 4 programming/measurement loops to achieve calibration. This represents a vast speed improvement (4x) over the brute force method, but is still far from optimal.

LINEAR INTERPOLATION

Looking back at Figure 1, at any given temperature (and voltage) point, we have a simple plain geometry problem to solve: that of determining offset (*f* nominal) and slope (d*f*/dx) of a line (the Calibration line). When the Calibration line is defined, we can then easily interpolate and determine the OSCCAL value for *f* ideal to achieve calibration.

TWO POINT CALIBRATION

To fully define a line in a two-dimensional space, we need to know at least two points. That, in our case, translates into determining two frequencies for two given values of OSCCAL. Since the further apart the two points are, the more accurate the interpolation will be, we can specify the two points to be at the extreme ends of the OSCCAL value range: -8 and +7. Again, a few lines of pseudo code well describes the algorithm:

```
OSCCAL = -8;
program part
measure oscillator frequency Fh
OSCCAL = +7;
program part
measure oscillator frequency Fl
interpolate: OSCCAL = 16*(Fideal-Fl)/(Fh-Fl)
program part
```

The overall efficiency is considerably increased since the total number of programming steps has been reduced further to only three. There are also only two steps where we actually measure the oscillator frequency.

ONE POINT CALIBRATION

To speed things up further, we need to cut some corners. That is, we have to assume that not only the calibration function can be represented as a line, but also that the slope is known and constant. This is a relatively risky assumption. In fact, with the normal variation of silicon manufacturing processes over time, a variation in slope should be expected, although over a very long period of time. So it can be relatively safe to assume that for an homogeneous lot of devices (small production run), the slope will be constant.

With this assumption, a single point linear calibration method can be devised, where the sole remaining unknown is the offset of the calibration line. The interpolation problem for such a simple case is significantly simplified: OSCCAL = int(f NOMINAL*K1 + K2)

This is again a linear function that can be conveniently reduced to the definition of a lookup table with only 16 entries.

The lookup table can be prepared with data collected over a short characterization series of measurements (over a sample run of parts).

A typical lookup table would resemble the one shown in Table 2:

TABLE 2: SINGLE POINT LINEAR INTERPOLATION

F (Hz)	Period (μs)	X
3144	318	+7
...	...	...
2577	388	+1
2500	400	0
2427	412	-1
...	...	...
2000	500	-8

We can enter such a table with the measured frequency (or period) of the blank device, find the closest matching row, and extract X, the optimal OSCCAL value.

The single point linear calibration algorithm, using such a lookup table, can be expressed in pseudo code as follows:

```
OSCCAL = 0
program part or just bulk erase
measure oscillator frequency F
find closest match in the look up table: X
set OSCCAL = X
program part
```

To minimize the risk of the constant slope assumption, a final consistency check can be performed after calibration, providing a final trimming option of ±1 bit, or raising a flag to force the lookup table to be updated.

```
OSCCAL = 0
program part or just bulk erase
measure oscillator frequency F1
find closest match in the look up table: X
set OSCCAL = X
program part
// final consistency check
measure oscillator frequency F2
Delta = F2-Fideal
if abs(Delta) > Tollerance then
    //  set a FLAG
    if Delta > 0 then
        increment OSCCAL
    else
        decrement OSCCAL
    endif
endif
```

Overall, this method is certainly the fastest, with a bulk erase followed by a single programing step (occasionally a second step might be required) and only two frequency measurements. Although a proper implementation, including the look-up table generation and update process, is not necessarily the easiest way to go.

TEMPERATURE OPTIMIZED CALIBRATION

So far, we have been ignoring the effects of temperature and supply voltage over the oscillator frequency. We have been operating with the sole objective of making the oscillator frequency as close as possible to a given ideal value, 'at room temperature and a given programming voltage.'

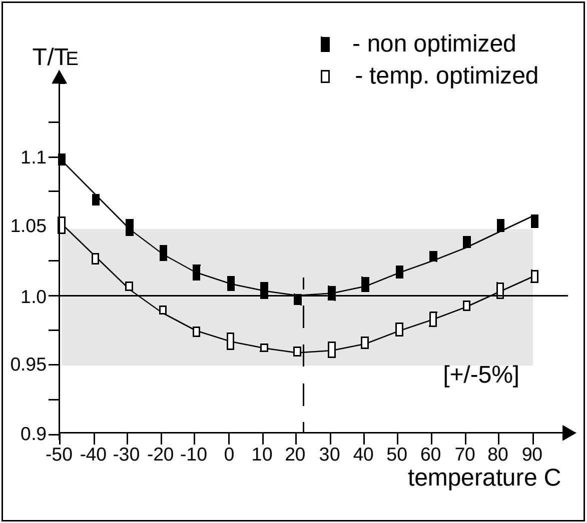
Unfortunately, over the entire life of a remote control application, most encoders will experience wide variations of these two parameters. Looking at Figure 2, we can observe how temperature seems to affect the device up to a maximum of ±5% over the entire Industrial temperature range (-40°C to +85°C). We can also notice how this curve is nicely U-shaped, and how at temperatures close to 25°C, it reaches a minimum.

Regarding voltage, we can notice how the effect of a decreasing supply voltage decreases the oscillator frequency (increases period). We can also appreciate how this translates into a modest shift up of the period/temperature curve, without modifying its shape (see Figure 2).

From these observations we can learn how to devise a better calibration strategy. In a temperature/voltage optimized calibration technique, we would not aim at getting the exact match of the oscillator frequency with *f* ideal, but rather, we would aim at *f* ideal + 5%.

In this way, small temperature variations from room temperature (positive and negative) would decrease frequency (increase period), with the effect of helping us to get closer to *f* ideal. Larger variations would push the period further up, but the overall excursion above T_E would be limited to a maximum of +5%, as shown in Figure 3.

FIGURE 3: TEMPERATURE OPTIMIZED CALIBRATION



Further, knowing at what voltage the device will be operating most of the time (and further considering a medium voltage over the application life) can help us establish an even better compromise, taking into account the "shift" effect on the oscillator curve consequent to battery depletion.

With respect to all the previously mentioned calibration methods, all these considerations translate simply into the selection of a different *f* ideal value (reduced by an appropriate percentage). In the case of the single point method, they translate into the adoption (construction) of a pre-compensated lookup table.

MEASURING OSCILLATOR FREQUENCY

None of the KEELOQ encoders offer a direct output pin where the oscillator frequency can be measured, nor is the absolute value of the internal oscillator frequency ever declared on a Data Sheet. However, there are indirect ways to determine proper calibration based on two methods:

1. TE measurements
2. Calibration Pulses measurements

Measuring TE

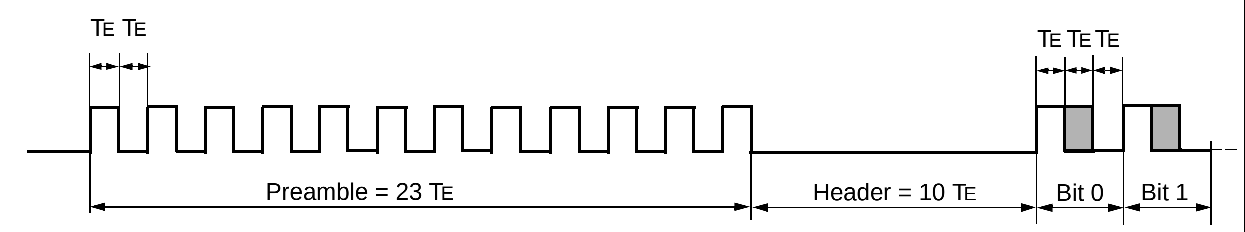
The first method is generic and works for all KEELOQ encoders capable of oscillator calibration. It is also the only option for the HCS362 and HCS410 devices. It consists of measuring some transmitted code word timings to derive TE, that is, the Basic Pulse Element (see Figure 4).

TE is, in fact, well defined in the Data Sheets and its measurement poses no serious difficulty. A transmission can be easily triggered activating a button input, S2 typically, that is the same pin used as a programming clock input during the programming process. The Code Word transmission output is available on the DATA pin. The following details must be considered in order to obtain reliable and repetitive information from the measurement:

- When triggering a transmission, after activating a button input (Sx), it is necessary to wait for the debouncing time, plus the standard transmission setup time of the encoder. These times can add up to 50 ms or more, before any preamble bit is output.
- TE varies with the selected baud rate, therefore, its value can be usefully compared against the nominal value from the Data Sheet, only when the baud rate is known (i.e., only after a bulk erase, BSEL=0, or after programming the part at least once).
- A single TE pulse in the preamble can be too short to guarantee adequate accuracy for the calibration. Adding up multiple preamble pulse times yields better results.
- The Header length changes with the modulation format: 10 TE for PWM vs. 4 TE for Manchester.
- On the HCS362, the Header length can also be reduced to 3 TE, depending on the HEADER configuration bit value in the SEED_3 Configuration word.

Again, knowing the modulation format is essential to deriving the correct value of TE, and the measurement is meaningful only after a bulk erase, or after programming the device at least once.

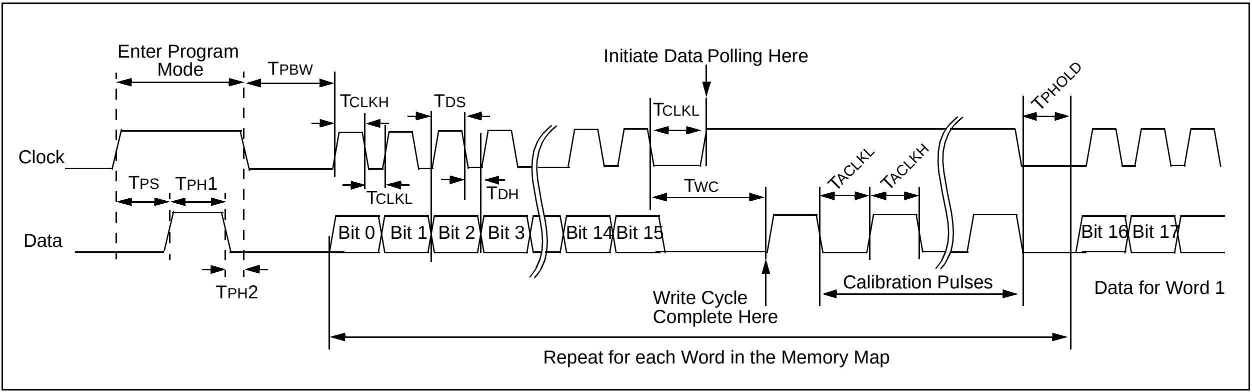
FIGURE 4: PWM CODE WORD TRANSMISSION TIMINGS



Measuring Calibration Pulses

The HCS101, HCS201 and HCS412, offer a second indirect method of measurement of the internal oscillator frequency. Special Calibration Pulses are provided during the programming sequence (see Figure 5), after each word write cycle.

FIGURE 5: PROGRAMMING WAVEFORMS WITH CALIBRATION PULSES



The device Data Sheets specify the nominal length of such calibration pulses (typically 800 µs) and their measurement is conveniently nested in the programming sequence of the part. So, the whole process results in a single and efficient continuous sequence.

There are some details on the calibration pulses measurement method that deserve attention:

- The calibration pulses length is independent of the baud rate or any other configuration parameter.
- Calibration pulses are output on the data pin when the word writing cycle is completed, and they repeat, until the clock line is lowered. Adding up multiple pulse times is recommended in order to increase the accuracy of the measurement.
- Calibration pulses are available after writing each word of the device memory and they are immediately affected by changes in the OSCCAL calibration bits. So, after writing the Configuration word containing OSCCAL, it is immediately possible to verify proper calibration.

HCS101 AND HCS201 CALIBRATION

Both the HCS101 and HCS201 offer calibration pulses for a convenient (indirect) measurement of oscillator frequency. Their programming and calibration are the simplest since there are only 12 words in the memory map. The Configuration word containing the OSCCAL bits is the last word (12th). Selecting almost any of the methods illustrated in the previous sections, from binary to single point linear calibration, will produce good results.

The following example shows the steps required to perform a single point linear calibration:

```
enter program mode
// this provides a bulk erase
// and also set the OSCCAL bits to 0000
loop 11 times
    write Nth word
    measure Calibration Pulses: Tcal
    // while waiting for each write
end loop
find closest match in look up table for Tcal
set OSCCAL in config word (last)
    write last word
    measure new calibration Pulses: TcalV
    // only for verification
```

In a single programming sequence, the part is calibrated and programmed. A new measurement of the Calibration Pulses length (`TcalV`) is also immediately available, at the end of the procedure to verify the effectiveness of calibration (and eventually to flag a request of lookup table update).

HCS412 CALIBRATION

The HCS412 also offers the convenience of the calibration pulses for the oscillator frequency measurement. There are 18 words in the memory map and the Configuration word containing the OSCCAL bits is located at address 8 (CONFIG_1).

All the previously presented methods can be applied, producing extremely efficient programming and calibration sequences. For example, a two-point linear calibration method can be expressed as follows:

```
enter program mode
// this provides a bulk erase
// and also set the OSCCAL bits to 0000
    write a dummy word
    measure calibration Pulses: Tcal0
cycle power (Vdd = 0; pause; Vdd=5V)
// this terminates first prog. seq.
enter program mode;
loop 7 times
    write Nth word
    measure Calibration Pulses: Tcal
    // while waiting for each write
end loop
    write config word(8th) with OSCCAL=+7
    measure calibration Pulses: Tcal7
cycle power (Vdd = 0; pause; Vdd=5V)
// this terminates second prog. seq.
// now we know Tcal0 and Tcal7
// we can interpolate and determine
// the perfect value for OSCCAL
enter program mode;
loop 18 times
    write Nth word
    measure Calibration Pulses: TcalV
    // while waiting for each write
end loop
```

Beside the apparent complexity, this method is actually very fast, since it requires only two partial programming sequences and one full programming sequence. There is no lookup table to construct and/or maintain. There is no assumption made on the slope of the calibration function. Further, the calibration pulses length value available at the end (`TcalV`) can be used to refine the algorithm, compensating even for eventual nonlinearity of the calibration function (allowing for a ±1 step final correction).

HCS362 CALIBRATION

This device requires the use of one of the TE measurement methods to determine the oscillator frequency. The memory map is composed of 18 words of which:

- word 11 (SEED_3) contains the modulation control (MOD) bit and the Synchronization Header length control (HEADER) bit.
- word 12 (CONFIG_0) contains the OSCCAL bits and the baud rate (BSEL) bits.

If the preamble pulse length is to be used to measure TE, then special attention must be given to the contents of the baud rate select bits.

The contents of `MOD` and `HEADER` bit in the Configuration word SEED_3 is important if measuring the synchronization header length.

All methods illustrated in the previous sections can be adapted to the HCS362. In the following we illustrate an example based on the single point linear calibration technique.

```
enter program mode;
// this provides a bulk erase
// and also set OSCCAL bits to 0000
// and also set BSEL to 00 = 100us
cycle power (vdd=0; pause; Vdd=5V)
// this terminates the programming seq.
trigger transmission (S2 = 5V);
wait for and measure preamble pulses;
// this can take some time
// count at least 8 pulses (1600us)
terminate transmission (S2 = 0)
find closest match in look up table
set OSCCAL in config word (CONFIG_0)
program part
```

This programming process is fast, but the additional time associated with the actual obtaining of the transmission and the initial bulk erase step increases the overall programming and calibration time of the part.

HCS410 CALIBRATION

These devices require the use of one of the TE measurement methods to determine the oscillator frequency. The memory map is composed of 18 words of which, word 5 (CONFIG), contains:

- the OSCCAL bits
- the baud rate (BSEL) bits
- the modulation format control (MOD) bit.

If the preamble pulses length method is to be used to measure TE, then special attention must be given to the contents of the baud rate select bits.

The contents of MOD is also important if measuring the synchronization header length.

All methods illustrated in the previous sections can be adapted to the HCS410. In the following code we will illustrate an example based on the single point linear calibration technique:

```
enter program mode;
// this provides a bulk erase
// and also set OSCCAL bits to 0000
// and also set BSEL to 00 = 400us!!!
cycle power (vdd=0; pause; Vdd=5V)
// this terminates the programming seq.
trigger transmission (S2 = 5V);
wait for and measure preamble pulses;
// this can take some time
// count at least 2 pulses (1600us)
terminate transmission (S2 = 0)
find closest match in look up table
set OSCCAL in config word (CONFIG)
program part
```

This programming process is fast, but the additional time associated with the actual obtaining of the transmission and the initial bulk erase step increases the overall programming and calibration time of the part.

Also, note how the bulk erase step clears all the configuration bits to 0, and produces a different baud rate setting for the part (400 µs compared to 100 µs for the HCS362). The lookup table must take this into account, or the number of preamble pulses measured for the two parts must be adapted.

UNIFIED CALIBRATION ALGORITHM

Because of all the differences presented so far, the reader might be of the impression that there cannot be a unified method of calibration. In reality, it is possible to devise such an algorithm with the following limitations:

- It is not possible to make use of the calibration pulses, since they are not available on all devices.
- It is not acceptable to use a single lookup table for all devices (unless the method prescribes the dynamic creation of such table at the beginning of every new programming session for every device type change).
- Probably, the safest method to employ would be a two-point linear interpolation, since it does not make any assumption on the calibration function slope (that can differ sensibly between device types).
- Measuring the preamble pulses will also require special attention for parts like the HCS101 and HCS201, that can output an extra initial START bit (that must be discarded).
- Different baud rates after bulk erase will have to be taken into account (same value of BSEL bits produces different results on different devices).

The source code presented in the Appendix will present one possible implementation of a Unified Calibration Algorithm, expressed in C language.

CONCLUSIONS

Although the oscillator calibration mechanism implemented in these KEELOQ encoders is very simple, the best frequency measurement and programming techniques to be used can differ sensibly from device to device. In this Application Note, we offered several simple algorithms to perform the task, as well as several considerations specific to each device. Desired programming speed and accuracy, as well as flexibility, will be important factors to guide the reader through the choice of the best calibration technique to employ.

REFERENCES

DS91002	(TB003) Introduction to KEELOQ Technology
DS41115	HCS101 Data Sheet
DS41098	HCS201 Data Sheet
DS41097	HCS362 Data Sheet
DS40158	HCS410 Data Sheet
DS41099	HCS412 Data Sheet
DS00218	HCS30X, HCS200 Stand-alone Programmer

MICROCHIP®

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: 480-792-7627
Web Address: http://www.microchip.com

Rocky Mountain
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966 Fax: 480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200B
Atlanta, GA 30350
Tel: 770-640-0034 Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848 Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071 Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423 Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250 Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, Indiana 46902
Tel: 765-864-8360 Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888 Fax: 949-263-1338

New York
150 Motor Parkway, Suite 202
Hauppauge, NY 11788
Tel: 631-273-5305 Fax: 631-273-5335

San Jose
Microchip Technology Inc.
2107 North First Street, Suite 590
San Jose, CA 95131
Tel: 408-436-7950 Fax: 408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699 Fax: 905-673-6509

ASIA/PACIFIC

Australia
Microchip Technology Australia Pty Ltd
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

China - Beijing
Microchip Technology Consulting (Shanghai) Co., Ltd., Beijing Liaison Office
Unit 915
Bei Hai Wan Tai Bldg.
No. 6 Chaoyangmen Beidajie
Beijing, 100027, No. China
Tel: 86-10-85282100 Fax: 86-10-85282104

China - Chengdu
Microchip Technology Consulting (Shanghai) Co., Ltd., Chengdu Liaison Office
Rm. 2401, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-6766200 Fax: 86-28-6766599

China - Fuzhou
Microchip Technology Consulting (Shanghai) Co., Ltd., Fuzhou Liaison Office
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506 Fax: 86-591-7503521

China - Shanghai
Microchip Technology Consulting (Shanghai) Co., Ltd.
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700 Fax: 86-21-6275-5060

China - Shenzhen
Microchip Technology Consulting (Shanghai) Co., Ltd., Shenzhen Liaison Office
Rm. 1315, 13/F, Shenzhen Kerry Centre,
Renminnan Lu
Shenzhen 518001, China
Tel: 86-755-2350361 Fax: 86-755-2366086

Hong Kong
Microchip Technology Hongkong Ltd.
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200 Fax: 852-2401-3431

India
Microchip Technology Inc.
India Liaison Office
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Microchip Technology Japan K.K.
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
Microchip Technology Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5934

Singapore
Microchip Technology Singapore Pte Ltd.
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-334-8870 Fax: 65-334-8850

Taiwan
Microchip Technology Taiwan
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Denmark
Microchip Technology Nordic ApS
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45 4420 9895 Fax: 45 4420 9910

France
Microchip Technology SARL
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79

Germany
Microchip Technology GmbH
Gustav-Heinemann Ring 125
D-81739 Munich, Germany
Tel: 49-89-627-144 0 Fax: 49-89-627-144-44

Italy
Microchip Technology SRL
Centro Direzionale Colleoni
Palazzo Taurus 1 V. Le Colleoni 1
20041 Agrate Brianza
Milan, Italy
Tel: 39-039-65791-1 Fax: 39-039-6899883

United Kingdom
Arizona Microchip Technology Ltd.
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44 118 921 5869 Fax: 44-118 921-5820

01/18/02