AN1328

KEELOQ® with XTEA Microcontroller-Based Transmitter with Acknowledge

INTRODUCTION
This application note describes the design of a microcontroller-based KEELOQ® Hopping transmitter with receiver acknowledge using the XTEA encryption algorithm. This transmitter is implemented on the Microchip PIC16F636 microcontroller. Descriptions of the encoding process, the encoding hardware and description of the software modules are included within this application note. The software was designed to emulate an HCS365 dual transmitter. This design can be used to implement a secure system transmitter that has the flexibility to be designed into various types of KEELOQ receiver/decoders. The acknowledge is achieved by using an MRF49XA transceiver.

BACKGROUND
XTEA stands for Tiny Encryption Algorithm Version 2. This encryption algorithm is an improvement over the original TEA algorithm. It was developed by David Wheeler and Roger Needham of the Cambridge Computer Laboratory. XTEA is practical both for its security and the small size of its algorithm. XTEA security is achieved by the number of iterations it goes through. The implementation in this KEELOQ transmitter uses 32 iterations. If a higher level of security is needed, 64 iterations can be used.

TRANSMITTER OVERVIEW
The transmitter has the following key features:

Security
- Two programmable 32-bit serial numbers
- Two programmable 128-bit encryption keys
- Two programmable 32-bit user values
- Each transmitter is unique
- 96-bit transmission code length
- 64-bit hopping code

Operation
- 2.0-5.5V operation
- Four-button inputs
- Automatic packet retry feature
- Nonvolatile synchronization data
- FSK modulation (handled internally by the MRF49XA)
- Dual transmitter functionality

DUAL TRANSMITTER OPERATION
This firmware contains two transmitter configurations with separate serial numbers, transmitter keys, user values, counters and seed values. This means that the transmitter can be used as two independent systems. The SHIFT (S3) input pin is used to select between transmitter configurations. When the dual transmitter feature is disabled, the button acts as a local status request, displaying the last received status on the LEDs.

RECEIVER ACKNOWLEDGE
On any button press, a data packet is sent over the air. The transmitter then goes to Listening mode for a period of time. During this time, the MRF49XA transceiver is in Listening mode and waits for a data packet coming back from the receiver. If no packet is received from the receiver end, then the transmitter has the ability to re-send the data packet (if the feature is enabled). The acknowledge indication is done using the two LEDs on the transmitter board.

SAMPLE BUTTONS/WAKE-UP
Upon power-up, the transmitter verifies the state of the buttons inputs and determines if a button is pressed. If no button press is detected, the transmitter will go to Sleep mode. The transmitter will wake-up whenever a button is pressed. Wake-up is achieved by configuring the input port to generate an interrupt-on-change. The button input values are then placed in the transmission buffer, in the appropriate section.
LOAD SYSTEM CONFIGURATION

After waking up and debouncing the input switches, the firmware will read the system Configuration bytes. All the system Configuration bytes are stored in the EEPROM. Below is the EEPROM mapping for the PIC16F636 transmitter showing the Configuration and data bits stored.

FIGURE 1: SOFTWARE FLOW DIAGRAM
MRF49XA RADIO CONFIGURATION

The radio link parameters in the MRF49xA are set to a default configuration that is adequate for the majority of applications. The baud rate is 9600 bps, using an FSK modulation with deviation of 60 kHz. For a more detailed description on how to setup the MRF49xA, please refer to AN1252, “Interfacing the MRF49XA Transceiver to PIC® Microcontrollers”.

The following considerations were made to select the MRF49XA Configuration Words.

The configuration considers the use of standard 30ppm crystal accuracy. Such a crystal will generate a frequency error of:

**EQUATION 1:**

\[
\Delta f_0 = \frac{30ppm}{10^6} \times 915 \times 10^6 = 27.45kHz
\]

The deviation can now be calculated:

**EQUATION 2:**

\[
\Delta f_{FSK} = 9600 + 2 \times \Delta f_0 + 10^3
\]

For the above values, we get a result of 74.5 kHz. The closest deviation supported by the MRF49XA transceiver is 75 kHz. For a maximum power output and a 75 kHz deviation, a value of 0x9840 is loaded into the TXCREG register.

Now, we can calculate the baseband bandwidth:

**EQUATION 3:**

\[
BBBW = \text{deviation} \times 2 - 10^3 \text{ Hz}
\]

For the above values, we get a result of 140 kHz. Picking a BBBW of 200 kHz, an RSSI of minus 97 dBm, and a maximum LNA gain, we get a value of 0x9481 to be loaded into the RXCREG register.

This code to configure the transceiver is contained in module MRF49XA.c.

**TABLE 1: EEPROM MAPPING FOR THE PIC16F636 TRANSMITTER**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Description</th>
<th>MNEMONIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Sync counter, byte 0, Transmitter 0, Copy A</td>
<td>EE_CNT0A</td>
</tr>
<tr>
<td>0x01</td>
<td>Sync counter, byte 1, Transmitter 0, Copy A</td>
<td></td>
</tr>
<tr>
<td>0x02</td>
<td>Sync counter, byte 2, Transmitter 0, Copy A</td>
<td></td>
</tr>
<tr>
<td>0x03</td>
<td>Sync counter, byte 3, Transmitter 0, Copy A</td>
<td></td>
</tr>
<tr>
<td>0x04</td>
<td>Sync counter, byte 0, Transmitter 0, Copy B</td>
<td>EE_CNT0B</td>
</tr>
<tr>
<td>0x05</td>
<td>Sync counter, byte 1, Transmitter 0, Copy B</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>Sync counter, byte 2, Transmitter 0, Copy B</td>
<td></td>
</tr>
<tr>
<td>0x07</td>
<td>Sync counter, byte 3, Transmitter 0, Copy B</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>Sync counter, byte 0, Transmitter 0, Copy C</td>
<td>EE_CNT0C</td>
</tr>
<tr>
<td>0x09</td>
<td>Sync counter, byte 1, Transmitter 0, Copy C</td>
<td></td>
</tr>
<tr>
<td>0x0A</td>
<td>Sync counter, byte 2, Transmitter 0, Copy C</td>
<td></td>
</tr>
<tr>
<td>0x0B</td>
<td>Sync counter, byte 3, Transmitter 0, Copy C</td>
<td></td>
</tr>
<tr>
<td>0x0C</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>0x0D</td>
<td>Sync counter, byte 0, Transmitter 1, Copy A</td>
<td>EE_CNT1A</td>
</tr>
<tr>
<td>0x0E</td>
<td>Sync counter, byte 1, Transmitter 1, Copy A</td>
<td></td>
</tr>
<tr>
<td>0x0F</td>
<td>Sync counter, byte 2, Transmitter 1, Copy A</td>
<td></td>
</tr>
<tr>
<td>0x10</td>
<td>Sync counter, byte 3, Transmitter 1, Copy A</td>
<td></td>
</tr>
<tr>
<td>0x11</td>
<td>Sync counter, byte 0, Transmitter 1, Copy B</td>
<td>EE_CNT1B</td>
</tr>
<tr>
<td>0x12</td>
<td>Sync counter, byte 1, Transmitter 1, Copy B</td>
<td></td>
</tr>
<tr>
<td>0x13</td>
<td>Sync counter, byte 2, Transmitter 1, Copy B</td>
<td></td>
</tr>
<tr>
<td>0x14</td>
<td>Sync counter, byte 3, Transmitter 1, Copy B</td>
<td></td>
</tr>
<tr>
<td>0x15</td>
<td>Sync counter, byte 0, Transmitter 1, Copy C</td>
<td>EE_CNT1C</td>
</tr>
<tr>
<td>0x16</td>
<td>Sync counter, byte 1, Transmitter 1, Copy C</td>
<td></td>
</tr>
<tr>
<td>0x17</td>
<td>Sync counter, byte 2, Transmitter 1, Copy C</td>
<td></td>
</tr>
<tr>
<td>0x18</td>
<td>Sync counter, byte 3, Transmitter 1, Copy C</td>
<td></td>
</tr>
<tr>
<td>0x19</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>Address</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>---------------------------------------</td>
<td>-----</td>
</tr>
<tr>
<td>0x1A</td>
<td>Serial Number, Byte 0, Transmitter 0</td>
<td>EE_SER0</td>
</tr>
<tr>
<td>0x1B</td>
<td>Serial Number, Byte 1, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x1C</td>
<td>Serial Number, Byte 2, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x1D</td>
<td>Serial Number, Byte 3, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x1E</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x1F</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x20</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x21</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x22</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x23</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x24</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x25</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x26</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x27</td>
<td>User value, Byte 0, Transmitter 0</td>
<td>EE_USER0</td>
</tr>
<tr>
<td>0x28</td>
<td>User value, Byte 1, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x29</td>
<td>User value, Byte 2, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x2A</td>
<td>User value, Byte 3, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x2B</td>
<td>Encryption Key, Byte 0, Transmitter 0</td>
<td>EE_KEY0</td>
</tr>
<tr>
<td>0x2C</td>
<td>Encryption Key, Byte 1, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x2D</td>
<td>Encryption Key, Byte 2, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x2E</td>
<td>Encryption Key, Byte 3, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x2F</td>
<td>Encryption Key, Byte 4, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x30</td>
<td>Encryption Key, Byte 5, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x31</td>
<td>Encryption Key, Byte 6, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x32</td>
<td>Encryption Key, Byte 7, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x33</td>
<td>Encryption Key, Byte 8, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x34</td>
<td>Encryption Key, Byte 9, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x35</td>
<td>Encryption Key, Byte 10, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x36</td>
<td>Encryption Key, Byte 11, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x37</td>
<td>Encryption Key, Byte 12, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x38</td>
<td>Encryption Key, Byte 13, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x39</td>
<td>Encryption Key, Byte 14, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x3A</td>
<td>Encryption Key, Byte 15, Transmitter 0</td>
<td></td>
</tr>
<tr>
<td>0x3B</td>
<td>Serial Number, Byte 0, Transmitter 1</td>
<td>EE_SER1</td>
</tr>
<tr>
<td>0x3C</td>
<td>Serial Number, Byte 1, Transmitter 1</td>
<td></td>
</tr>
<tr>
<td>0x3D</td>
<td>Serial Number, Byte 2, Transmitter 1</td>
<td></td>
</tr>
<tr>
<td>0x3E</td>
<td>Serial Number, Byte 3, Transmitter 1</td>
<td></td>
</tr>
<tr>
<td>0x3F</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x40</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x41</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x42</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x43</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x44</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x45</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x46</td>
<td>—</td>
<td></td>
</tr>
<tr>
<td>0x47</td>
<td>—</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 1: EEPROM MAPPING FOR THE PIC16F636 TRANSMITTER (CONTINUED)

<table>
<thead>
<tr>
<th>Address</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x48</td>
<td>User value, Byte 0, Transmitter 1</td>
</tr>
<tr>
<td>0x49</td>
<td>User value, Byte 1, Transmitter 1</td>
</tr>
<tr>
<td>0x4A</td>
<td>User value, Byte 2, Transmitter 1</td>
</tr>
<tr>
<td>0x4B</td>
<td>User value, Byte 3, Transmitter 1</td>
</tr>
<tr>
<td>0x4C</td>
<td>Encryption Key, Byte 0, Transmitter 1</td>
</tr>
<tr>
<td>0x4D</td>
<td>Encryption Key, Byte 1, Transmitter 1</td>
</tr>
<tr>
<td>0x4E</td>
<td>Encryption Key, Byte 2, Transmitter 1</td>
</tr>
<tr>
<td>0x4F</td>
<td>Encryption Key, Byte 3, Transmitter 1</td>
</tr>
<tr>
<td>0x50</td>
<td>Encryption Key, Byte 4, Transmitter 1</td>
</tr>
<tr>
<td>0x51</td>
<td>Encryption Key, Byte 5, Transmitter 1</td>
</tr>
<tr>
<td>0x52</td>
<td>Encryption Key, Byte 6, Transmitter 1</td>
</tr>
<tr>
<td>0x53</td>
<td>Encryption Key, Byte 7, Transmitter 1</td>
</tr>
<tr>
<td>0x54</td>
<td>Encryption Key, Byte 8, Transmitter 1</td>
</tr>
<tr>
<td>0x55</td>
<td>Encryption Key, Byte 9, Transmitter 1</td>
</tr>
<tr>
<td>0x56</td>
<td>Encryption Key, Byte 10, Transmitter 1</td>
</tr>
<tr>
<td>0x57</td>
<td>Encryption Key, Byte 11, Transmitter 1</td>
</tr>
<tr>
<td>0x58</td>
<td>Encryption Key, Byte 12, Transmitter 1</td>
</tr>
<tr>
<td>0x59</td>
<td>Encryption Key, Byte 13, Transmitter 1</td>
</tr>
<tr>
<td>0x5A</td>
<td>Encryption Key, Byte 14, Transmitter 1</td>
</tr>
<tr>
<td>0x5B</td>
<td>Encryption Key, Byte 15, Transmitter 1</td>
</tr>
<tr>
<td>0x5C</td>
<td>—</td>
</tr>
<tr>
<td>0x5D</td>
<td>System configuration</td>
</tr>
<tr>
<td>0x5E</td>
<td>—</td>
</tr>
<tr>
<td>0x5F</td>
<td>—</td>
</tr>
</tbody>
</table>

### TABLE 2: TRANSMITTER CONFIGURATION OPTIONS

<table>
<thead>
<tr>
<th>BIT</th>
<th>Field</th>
<th>Description</th>
<th>Values</th>
</tr>
</thead>
</table>
| 0   | MRT   | Maximum number of transmission retries | 00 – None  
|     |       |                                   | 01 – Once  
|     |       |                                   | 10 – Twice  
|     |       |                                   | 11 – Three times |
| 1   | INDESEL | Dual Transmitter Enable         | 0 = Disable  
|     |        |                                   | 1 = Enable |
| 2   | Not used |                                 | — |
| 3   | TSEL  | Time-out Select                  | 00 – 300 ms  
|     |       |                                   | 01 – 500 ms  
|     |       |                                   | 10 – 1000 ms  
|     |       |                                   | 11 – 2000 ms |
| 5   | Not used |                                 | — |
| 6   | Not used |                                 | — |
| 7   | Not used |                                 | — |
EE_SER0 AND EE_SER1

These locations store the 4 bytes of the 32-bit serial number for transmitter 1 and transmitter 2. There are 32 bits allocated for the serial number, and the serial number is meant to be unique for every transmitter.

EE_USER0 AND EE_USER1

These locations store the 4 bytes of the 32-bit user code for transmitter 1 and transmitter 2. There are 32 bits allocated for the user code, and the user code is meant to be unique for every transmitter.

EE_KEY0 AND EE_KEY1

The 128-bit encryption key is used by the transmitter to create the encrypted message transmitted to the receiver. This key is created using a key generation algorithm. The inputs to the key generation algorithm are the secret manufacturer’s code and the serial number. The user may choose to use the algorithm supplied by Microchip or to create their own method of key generation.

SYNCHRONIZATION COUNTER STORAGE

The following addresses save the counter and the checksum values. The counter value is stored in the Counter locations (EE_CNT0A, EE_CNT0B and EE_CNT0C for transmitter 1 and EE_CNT1A, EE_CNT1B and EE_CNT1C for transmitter 2) described in the EEPROM table. This code is contained in module counter.c.

For reliability, three copies of the synchronization counter are being stored. When reading counter value from the EEPROM memory, the counter is being verified against two additional copies of the same counter. If the values match, the counter value is considered correct. If the values do not match, the counter value is reconstructed from the additional counter copies. The firmware flow diagram is shown in Figure 2.
FIGURE 2: COUNTER CHECK DIAGRAM

START

Read Copy A

Read Copy B

A=B ?

YES

Use Copy A

NO

Read Copy C

B=C ?

NO

Use Copy A

YES

Use Copy C + 1
AUTOMATIC RETRY

Upon transmission of a data packet, the transmitter waits for reception of acknowledge from the receiver. The acknowledge reception can occur after the transmission of a radio packet. A time-out period is used and, if the acknowledge is not received, the reception is aborted. The time-out period is set according to the TSEL field of the Configuration register. If a packet acknowledge is not received, the transmitter has the ability to resend the data packet and wait for another acknowledge. The number of retries is defined in the MRT field of the Configuration register. This feature can be enabled, with a maximum of three retries, or it can be completely disabled. The sequence can be one of the following scenarios (see Figure 3).

FIGURE 3: DIFFERENT ACKNOWLEDGE SCENARIOS

In Figure 3, we see a total of six different acknowledge scenarios.

The first one is the most simple and will occur for the majority of time under normal conditions. Immediately after a transmission, the transmitter goes to Listening mode waiting for acknowledge. In this case, acknowledge is received on time and no time-out event occurs.

The second case represents a transmitter that has the automatic retry feature disabled. After a time-out event, the transmitter is not sending a new transmission.

In cases 3 and 4, we can see the transmitter’s automatic retry feature. After a time-out event, the transmitter sends a new data packet. In case 4, no acknowledge is received, even though the transmitter retried three times – the maximum allowed by the MRT setting.
CODE TRANSMISSION FORMAT

The following is the data stream format transmitted:

<table>
<thead>
<tr>
<th>Plain Text (32 bits)</th>
<th>Encrypted (64 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial number (32 bits)</td>
<td>Function code (8 bits)</td>
</tr>
</tbody>
</table>

A KEELQ/XTEA transmission consists of 64 bits of hopping code data and 32 bits of fixed code data.

HOPPING CODE PORTION

The hopping code portion is calculated by encrypting the function code, user code, and the counter with the Transmitter Key (KEY). A new hopping code is calculated every time a button is pressed. The user code can be programmed with any fixed value to serve as a post decryption check on the receiver end. This code portion is transmitted in encrypted format.

FIXED CODE PORTION

The fixed code portion consists of 32 bits of serial number and, therefore, is transmitted in non-encrypted format (plain text).

FIRMWARE MODULES

The following files make up the KEELQ transmitter firmware:

- main.c: this file contains the main loop routine, as well as the wake-up, debounce, read configuration, load transmit buffer and transmit routines.
- packet.c: this file loads the transmit buffer according to the encryption algorithm.
- MRF49XA.c: this file contains all the functions that control the MRF49XA transceiver.
- counter.c: this file loads the synchronization counter, checks its validity and automatically corrects any errors.
- encryption.c: this file contains the functions that provide the encryption algorithm. Because of statutory export license restrictions on encryption software, the source code listings for the XTEA algorithms are not provided here.

These applications may be ordered from Microchip Technology Inc. through its sales offices, or through the corporate web site: www.microchip.com.

FIRMWARE CONFIGURATION

The transmitter firmware is fully configurable. The encryption algorithm can be changed very easily. All the necessary functions and definitions are contained in the encryption.c and encryption.h modules.

Changing the encryption algorithm is as simple as replacing the above module and recompiling the source code.

CONCLUSION

This KEELQ/XTEA transmitter firmware has all the features of a standard hardware transmitter. What makes this firmware implementation useful is that it gives the designer the power and flexibility of modifying the encoding and/or transmission formats and parameters to suit their security system. In addition, this system allows the user to receive acknowledge from the intended receiver.

REFERENCES

C. Toma, AN1252, “Interfacing the MRF49XA Transceiver to PIC® Microcontrollers” (DS01252) Microchip Technology Inc., 2009
ADDITIONAL INFORMATION

Microchip's Secure Data Products are covered by some or all of the following:
Code hopping encoder patents issued in European countries and U.S.A.
Secure learning patents issued in European countries, U.S.A. and R.S.A.

REVISION HISTORY

Revision B (June 2011)
- Added new section Additional Information
- Minor formatting and text changes were incorporated throughout the document
Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip’s Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Trademarks
The Microchip name and logo, the Microchip logo, dsPIC, KEELOG, KEELOG logo, MPLAB, PIC, PICmicro, PICSTART, PIC32® logo, rPIC and Uni/O are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
FilterLab, Hampshire, HI-TECH C, Linear Active Thermistor, MXDEV, MXLAB, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.
Analog-for-the-Digital Age, Application Maestro, chipKIT, chipKIT logo, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, dsSPEAK, ECAN, ECONOMONITOR, FanSense, HI-TIDE, In-Circuit Serial Programming, ICSP, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, mTouch, Omniscient Code Generation, PICC, PICC-18, PICDEM, PICDEM.net, PICkit, PICtail, REAL ICE, rFLAB, Select Mode, Total Endurance, TSHARC, UniWinDriver, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.
SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.
All other trademarks mentioned herein are property of their respective companies.
© 2010-2011, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
Printed on recycled paper.

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company’s quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOG® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip’s quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV
ISO/TS 16949:2009

© 2010-2011 Microchip Technology Inc.
# 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](http://www.microchip.com/support)  
Web Address: [www.microchip.com](http://www.microchip.com)

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

**Boston**  
Westborough, MA  
Tel: 774-447-0087  
Fax: 774-447-0088

**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**  
Farmington Hills, MI  
Tel: 248-538-2250  
Fax: 248-538-2260

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

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

**Santa Clara**  
Santa Clara, CA  
Tel: 408-961-6444  
Fax: 408-961-6445

**Toronto**  
Mississauga, Ontario, Canada  
Tel: 905-673-0699  
Fax: 905-673-6509

## ASIA/PACIFIC

**Asia Pacific Office**  
Suites 3707-14, 37th Floor  
Tower 6, The Gateway  
Harbour City, Kowloon  
Hong Kong  
Tel: 852-2401-1200  
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 - Hangzhou**  
Tel: 86-571-2819-3180  
Fax: 86-571-2819-3189

**China - Hong Kong SAR**  
Tel: 852-2401-1200  
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: 86-755-8203-2660  
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

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

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

## ASIA/PACIFIC

**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: 91-20-2566-1512  
Fax: 91-20-2566-1513

**Japan - Yokohama**  
Tel: 81-45-471-6166  
Fax: 81-45-471-6122

**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: 886-3-6578-300  
Fax: 886-3-6578-370

**Taiwan - Kaohsiung**  
Tel: 886-7-213-7830  
Fax: 886-7-330-9305

**Taiwan - Taipei**  
Tel: 886-2-2500-6610  
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 - Munich**  
Tel: 49-89-627-144-0  
Fax: 49-89-627-144-44

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

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

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

**UK - Wokingham**  
Tel: 44-118-921-5869  
Fax: 44-118-921-5820

05/02/11