INTRODUCTION

This document describes a secure transponder system. The system is suitable for use in security applications such as cars, motor bikes, and scooters (two-wheelers). Microchip’s secure HCS410 KeeLoq® code hopping transponder is used. The decoder is implemented on a Microchip PIC16C56 microcontroller. The software can be used to implement a stand-alone decoder or can be integrated into a security system. The maximum operating range of this particular application circuit is 25 millimeters (one inch).

KEY FEATURES

• Stand-alone transponder decoder
• Compatible with KeeLoq HCS410 transponder
• Twelve learnable transponders
• Two function outputs
• XT oscillator

TYPICAL APPLICATIONS

• Automotive/scooter/motorcycle
• Access control
• Gate and garage door openers
• Identity tokens
HARDWARE

Overview

The hardware for this application note consists of a microcontroller circuit, a transponder, and a base station circuit. Figure 1 shows an overview of the hardware and the interface between each block. The base station is shown in Figure 2. The transponder and microcontroller are shown in Figure 3.

FIGURE 1: TRANSPONDER SYSTEM BLOCK DIAGRAM

FIGURE 2: READ/WRITE BASE STATION
FIGURE 3: MICROCONTROLLER, POWER SUPPLY, AND TRANSPONDER

MICROCONTROLLER

+5V
U7 TCM809L
3 VCC 2 RESET

1

+5V
U2 IMMOMB

1

17

18

1 LRN_IN

DATA_B2T

DATA_B2T

LRN_OUT

DATA_T2B

DATA_T2B

IGN/POLL

FIELD

EE_DIO

EE_CLK

EE_CS

VALID

PIC16C56

+5V
R15 4

10K

MCLR

RA0

Vcc

C8

22 pF

C9

22 pF

4 MHz
Y1

RA1

RA2

RA3

CLKOUT

OSC1

16

15

R16

10K

1

2

R17

1K

1

2

+5V
U3 LM7805

8 Vcc

C10

22 µF

1

2

GND

1

2

Vcc

NC

NC

NC

DI

DIO

1

2

3

4

93LC46B

R19

1K

1

2

D4 LED

D3 LED

D2 LED

POWER SUPPLY

+5V
U4 LM7805

1 Vcc

C10

22 µF

1

2

GND

1

2

VO

1

2

+5V
C11 10 nF

1

2

C12 10 µF

1

2

+5V

TRANSPONDER

U5

1

8

C14

0.1 µF

15V

1

2

3

4

5

7

8

C13

1.5 nF

1

2

1.0 mH
**Microcontroller**

The microcontroller consists of the following components:

- Microchip PIC16C56 microcontroller
- A Microchip 93LC46B serial EEPROM used to store all the information of learned transponders
- A 5V supply voltage regulator
- A supply supervisor that inhibits the microcontroller during low voltage events
- Two push buttons used for user inputs
- Three indicator LEDs used for user feedback and indication of function outputs

The microcontroller interfaces to the base station circuit by means of two wires: DATA_T2B used to read data from the transponder to the base station. The carrier enable line of the read/write base station (DATA_B2T) used by the base microcontroller to send data to the transponder.

Table 1 lists the I/O pin assignment for this application.

**Read/Write Base Station**

The base station is designed to operate from supply Vcc between 9V and 12 V.

The 14-bit binary counter U1 divides the 4MHz microcontroller clock CLKOUT to produce a 125kHz clock to transistor Q1. This transistor drives the resonant circuit formed by R1, C1 and L1 to produce a magnetic field. The microcontroller can switch the magnetic field on and off via signal DATA_B2T for communication to the transponder.

The transponder is powered by coupling with the magnetic field produced by L1. The transponder also modulates this field for communication back to the microcontroller. The field modulation is detected at the junction of C1 and L1 by the envelope detector circuit input at diode D5. The envelope detector output signal is amplified by U6A and the data is recovered by bandpass filter U6B. The filter output DATA_T2B is fed directly to the microcontroller.

Table 8 lists the components used in the decoder circuit as shown in the schematics, in Figure 2 and Figure 3. Table 9 lists the components used for the transponder in Figure 3.

### TABLE 1: PIC16C56 I/O PIN ASSIGNMENT

<table>
<thead>
<tr>
<th>Pin Number</th>
<th>Pin Function</th>
<th>Device Pin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>IMMOB</td>
<td>RA0</td>
<td>Immobilize function output</td>
</tr>
<tr>
<td>1</td>
<td>LRN_IN</td>
<td>RA2</td>
<td>Input to initiate learning</td>
</tr>
<tr>
<td>2</td>
<td>LRN_OUT</td>
<td>RA3</td>
<td>Output to show the status of the learn process (in an integrated system this will be combined with the system status indicator).</td>
</tr>
<tr>
<td>6</td>
<td>DATA_B2T</td>
<td>RB0</td>
<td>Data from base station to transponder</td>
</tr>
<tr>
<td>7</td>
<td>DATA_T2B</td>
<td>RB1</td>
<td>Data from transponder to base station</td>
</tr>
<tr>
<td>8</td>
<td>IGN/POLL</td>
<td>RB2</td>
<td>Input to activate transponder polling</td>
</tr>
<tr>
<td>9</td>
<td>FIELD</td>
<td>RB3</td>
<td>Magnetic field is active</td>
</tr>
<tr>
<td>10</td>
<td>EE_DIO</td>
<td>RB4</td>
<td>Interface lines to external serial EEPROM</td>
</tr>
<tr>
<td>11</td>
<td>EE_CLK</td>
<td>RB5</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>EE_CS</td>
<td>RB6</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>VALID</td>
<td>RB7</td>
<td>Valid token pulse function output</td>
</tr>
</tbody>
</table>
INTRODUCTION TO THE HCS410 KeeLoq TRANSPONDER

The HCS410 is a KeeLoq code hopping transmitter/transponder designed for secure entry and identification system. The device combines the circuitry required for Remote Keyless Entry (RKE) and inductively coupled Identify Friend or Foe (IFF). This section describes software which uses the inductive coupled IFF functions of the HCS410.

IFF Activation

IFF mode is activated when the HCS410 senses a signal on its LC0 pin. After the HCS410 verifies application of power and elapse of the normal debounce time, the device starts to acknowledge IFF activation by loading the LC pins with continuous acknowledge pulses as shown in Figure 4. This is an indication that the HCS410 is ready to receive a command. All the communication timing is done in multiples of the basic time element TE.

IFF Commands

The HCS410 transponder responds to 5-bit IFF commands or opcodes. The opcodes are sent to the HCS410 with the least significant bit (LSb) first. Depending on the command, additional data may be required for the HCS410 to respond. A list of IFF commands can be found in the HCS410 data sheet (DS40158).

FIGURE 4: IFF ACTIVATION WAVEFORM

<table>
<thead>
<tr>
<th>Pulse width = TE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>36 ms</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>[3 TE][3 TE]</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>255 TE</td>
</tr>
</tbody>
</table>
IFF Communication Protocols and Waveforms

All communication to and from the HCS410 during IFF is done in asynchronous Pulse Position Modulation (PPM) format. The format differs when sending commands and data to the HCS410 and when receiving data from the HCS410. After a complete transaction, the HCS410 is ready for the next command and will continue to send out acknowledge pulses. Commands to the HCS410 start with a pulse of 2 T_E. Time is measured from rising edge to rising edge with a logic 1 being 6 T_E and a logic 0, 4 T_E.

Data coming from the HCS410 starts with a start pulse of 1 T_E. Again, time is measured from rising edge to rising edge with a logic 1 being 3 T_E and a logic 0, 2 T_E. All data words are preceded by two preamble bits with the logic value 01₂ before the data is sent out.

FIGURE 5: IFF COMMUNICATION WAVEFORM

<table>
<thead>
<tr>
<th>Logic 0</th>
<th>Logic 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Decoder Commands</strong></td>
<td><strong>Encoder Response</strong></td>
</tr>
<tr>
<td>Logic 0</td>
<td>Logic 1</td>
</tr>
<tr>
<td>4 T_E</td>
<td>6 T_E</td>
</tr>
<tr>
<td>2 T_E</td>
<td>3 T_E</td>
</tr>
<tr>
<td>T_PMH</td>
<td>T_PMH</td>
</tr>
</tbody>
</table>
Identify Friend or Foe (IFF)

Identify Friend or Foe (IFF) is a procedure used to authenticate a transponder. IFF challenges the transponder with a random 32-bit value and then verifies the response.

HCS410 Commands Used

This application uses the following transponder commands: The IFF READ command (Figure 6) is used to read the two portions of the 32-bit serial number (SER1 and SER0).

The IFF CHALLENGE command (IFF1 using key-1 and HOP algorithm) is used to validate the transponder. The microcontroller generates a 32-bit random challenge and then validates the transponder's 32-bit response by decrypting the response using the KEELOG decryption algorithm.

**FIGURE 6: IFF READ COMMAND**

```
Ack pulses
Start pulse 2 T
RT Response 18 bits
```

**FIGURE 7: IFF CHALLENGE COMMAND**

```
Ack pulses Opcode TOTD Challenge 16/32 bits Response 18/34 bits
```
SOFTWARE DESCRIPTION

Overview (Figure 8)

After reset, the decoder enters the main loop. The main loop checks the learn button and if pressed (TST_LEARN) enters the learn mode. The decoder also checks the IGN/POLL input and if low it starts polling for transponder acknowledge pulses for up to 30 seconds. If a transponder is detected, it is validated by means of a 32-bit challenge/response IFF. The decoder pulses the VALID output pin for 500 ms and asserts the IMMO output for the duration that the IGN/POLL input is held low if the transponder is authentic.

Transponder Validation Flow

The decoder reads the transponder’s 32-bit serial number after it detects the acknowledge pulses. It then calculates the 16-bit serial number checksum value. The decoder then searches through all the EEPROM memory blocks for a matching checksum value. Then, it challenges the transponder with a 32-bit random challenge. The decoder validates the transponder by decrypting the 32-bit response with the 64-bit transponder key and comparing it to the 32-bit challenge.

Transponder Learn Flow

The 64-bit Manufacturer’s Code is read from the ROM table after the decoder enters learn mode. The decoder then starts polling the field to check if there is any transponder in the field for up to 30 seconds. The decoder reads the transponder’s 32-bit serial number after it detects acknowledge pulses. The transponder’s decryption key is then calculated using the 64 bit Manufacturer’s Code and the 32-bit serial number. The decoder then challenges the transponder with a 32-bit random challenge and validates the 32-bit response by using the newly calculated 64-bit transponder key. The decoder calculates the 16-bit serial number checksum then stores both the 16-bit checksum value and the 64-bit transponder key in EEPROM.

Calibration on Acknowledge Pulses

The WAIT_ACK function determines if there is a transponder in the field. The routine also calibrates on the acknowledge pulses of the transponder, thereby determining the basic elemental periods $T_E$, which is used for communication to the HCS410 transponder. The routine switches on the inductive field and waits for 30 ms for the transponder to activate. It then waits for up to 100 ms for a falling edge on the data output line of the read/write base station. The decoder calibrates on the time between the two rising edges. This time, which is equal to $2 T_E$, is used by the WAIT_TE routine during communication to the HCS410. The decoder waits for three acknowledge pulse pairs before it indicates that there is a transponder in the field by setting the zero flag and returning E_OK.

Capturing Data from the HCS410

The REC_PPM function is used to receive PPM data from the transponder. The decoder waits for the start bit, after which it starts measuring the time from rising edge to rising edge. The decoder then checks if this value is less than $2 T_E$ in which case, the result bit value is set to logic 0. Otherwise, the bit value is set to a logic 1. The function receives either 18- or 34-bits, depending on the initial value of the loop counter (CNT2). The incoming data is stored in a 32-bit temporary shift register (TMP3:TMP0). The two preamble bits are rotated though the shift register and their values are ignored.

Source Code

A floppy disk containing the source code for this application note is available under no fee license from your Microchip distributor. The disk order number is DS40149.
FIGURE 8: PROGRAM FLOW DIAGRAM

Start

Initialize

Activate Learn Sequence

LEARNPressed

POLL

Pressed?

Yes

No

No

Switch Field On

Valid ACKs?

Yes

No

No

Timeout Reached

Yes

No

Read Serial Number and Calculate Checksum

Compare with EEPROM’s Checksum

Equal?

Yes

No

Increase Block Index

Read 64 Bit Decryption Key

Challenge HCS410 with a 32 bit Challenge

Response Valid

Yes

No

Generate Function Outputs

Stop

Yes

No
ADDING/LEARNING TRANSPONDERS

Overview
Adding/learning a transponder involves calculating the transponder’s decryption key, then challenging the transponder and verifying the response using the newly derived key. If the learn was successful, the key and a serial number checksum will be stored in EEPROM. The decoder reads the transponder’s 32-bit serial number, forces the upper 4 bits to 6h or 2h to calculate the two input seed algorithms. Then, using these two input seeds and the decryption algorithm, the 64-bit transponder key is calculated. The Manufacturer’s Code is stored in a ROM table in program memory.

Generating the 64-bit Key
SEED1 = 6h + 28 bit Serial Number
SEED2 = 2h + 28 bit Serial Number

The transponder key is derived using the KeELOQ decryption algorithm and the 64-bit Manufacturer’s Code as follows:

Key Upper 32 bits = F KEeLOQ Decrypt (SEED1) | 64-Bit Manufacturers Code
Key Lower 32 bits = F KEeLOQ Decrypt (SEED2) | 64-Bit Manufacturers Code

Calculating the 16-bit Checksum From 28-bit Serial Number
The serial number checksum value stored in EEPROM is calculated by as follows:
Checksum = [(SER_3 | SER_1) << 8] + (SER_2 | SER_0)

Note: If the calculated checksum is zero the value is changed to 5AA5h.
## APPENDIX A: MEMORY ALLOCATIONS

### TABLE 2: MEMORY FOR EACH TRANSPONDER

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Checksum</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Key</td>
<td>8</td>
</tr>
</tbody>
</table>

### TABLE 3: COMBINED EEPROM MEMORY ALLOCATION

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Block 1 Scratch pad</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Block 2 16-bit seed counter used by random generator</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Block 3 Stored data for transponder #1</td>
<td>10</td>
</tr>
<tr>
<td>4</td>
<td>Block 4 Stored data for transponder #2</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>Block 5 Stored data for transponder #3</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Block 6 Stored data for transponder #4</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>Block 7 Stored data for transponder #5</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>Block 8 Stored data for transponder #6</td>
<td>10</td>
</tr>
<tr>
<td>9</td>
<td>Block 9 Stored data for transponder #7</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>Block 10 Stored data for transponder #8</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>Block 11 Stored data for transponder #9</td>
<td>10</td>
</tr>
<tr>
<td>12</td>
<td>Block 12 Stored data for transponder #10</td>
<td>10</td>
</tr>
<tr>
<td>13</td>
<td>Block 13 Stored data for transponder #11</td>
<td>10</td>
</tr>
<tr>
<td>14</td>
<td>Block 14 Stored data for transponder #12</td>
<td>10</td>
</tr>
</tbody>
</table>

**Total** 128
<table>
<thead>
<tr>
<th>Address</th>
<th>Mnemonic</th>
<th>Address</th>
<th>Mnemonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>CHAL_LW</td>
<td>20</td>
<td>KEY6_2</td>
</tr>
<tr>
<td>01</td>
<td>RESP_LW</td>
<td>21</td>
<td>KEY6_3</td>
</tr>
<tr>
<td>02</td>
<td>RESP_HI</td>
<td>22</td>
<td>CHKSUM_7</td>
</tr>
<tr>
<td>03</td>
<td>CHAL_SEED</td>
<td>23</td>
<td>KEY7_0</td>
</tr>
<tr>
<td>04</td>
<td>CHKSUM_1</td>
<td>24</td>
<td>KEY7_1</td>
</tr>
<tr>
<td>05</td>
<td>KEY1_0</td>
<td>25</td>
<td>KEY7_2</td>
</tr>
<tr>
<td>06</td>
<td>KEY1_1</td>
<td>26</td>
<td>KEY7_3</td>
</tr>
<tr>
<td>07</td>
<td>KEY1_2</td>
<td>27</td>
<td>CHKSUM_8</td>
</tr>
<tr>
<td>08</td>
<td>KEY1_3</td>
<td>28</td>
<td>KEY8_0</td>
</tr>
<tr>
<td>09</td>
<td>CHKSUM_2</td>
<td>29</td>
<td>KEY8_1</td>
</tr>
<tr>
<td>0A</td>
<td>KEY2_0</td>
<td>2A</td>
<td>KEY8_2</td>
</tr>
<tr>
<td>0B</td>
<td>KEY2_1</td>
<td>2B</td>
<td>KEY8_3</td>
</tr>
<tr>
<td>0C</td>
<td>KEY2_2</td>
<td>2C</td>
<td>CHKSUM_9</td>
</tr>
<tr>
<td>0D</td>
<td>KEY2_3</td>
<td>2D</td>
<td>KEY9_0</td>
</tr>
<tr>
<td>0E</td>
<td>CHKSUM_3</td>
<td>2E</td>
<td>KEY9_1</td>
</tr>
<tr>
<td>0F</td>
<td>KEY3_0</td>
<td>2F</td>
<td>KEY9_2</td>
</tr>
<tr>
<td>10</td>
<td>KEY3_1</td>
<td>30</td>
<td>KEY9_3</td>
</tr>
<tr>
<td>11</td>
<td>KEY3_2</td>
<td>33</td>
<td>CHKSUM_10</td>
</tr>
<tr>
<td>12</td>
<td>KEY3_3</td>
<td>32</td>
<td>KEY10_0</td>
</tr>
<tr>
<td>13</td>
<td>CHKSUM_4</td>
<td>33</td>
<td>KEY10_1</td>
</tr>
<tr>
<td>14</td>
<td>KEY4_0</td>
<td>34</td>
<td>KEY10_2</td>
</tr>
<tr>
<td>15</td>
<td>KEY4_1</td>
<td>35</td>
<td>KEY10_3</td>
</tr>
<tr>
<td>16</td>
<td>KEY4_2</td>
<td>36</td>
<td>CHKSUM_11</td>
</tr>
<tr>
<td>17</td>
<td>KEY4_3</td>
<td>37</td>
<td>KEY11_0</td>
</tr>
<tr>
<td>18</td>
<td>CHKSUM_5</td>
<td>38</td>
<td>KEY11_1</td>
</tr>
<tr>
<td>19</td>
<td>KEY5_0</td>
<td>39</td>
<td>KEY11_2</td>
</tr>
<tr>
<td>1A</td>
<td>KEY5_1</td>
<td>3A</td>
<td>KEY11_3</td>
</tr>
<tr>
<td>1B</td>
<td>KEY5_2</td>
<td>3B</td>
<td>CHKSUM_12</td>
</tr>
<tr>
<td>1C</td>
<td>KEY5_3</td>
<td>3C</td>
<td>KEY12_0</td>
</tr>
<tr>
<td>1D</td>
<td>CHKSUM_6</td>
<td>3D</td>
<td>KEY12_1</td>
</tr>
<tr>
<td>1E</td>
<td>KEY6_0</td>
<td>3E</td>
<td>KEY12_2</td>
</tr>
<tr>
<td>1F</td>
<td>KEY6_1</td>
<td>3F</td>
<td>KEY12_3</td>
</tr>
</tbody>
</table>

**CHAL_LW** Temporary storage of lower 16 bits of challenge  
**RESP_HI** Temporary storage of upper 16 bits of HCS410’s response  
**RESP_LW** Temporary storage of lower 16 bits of HCS410’s response  
**CHAL_SEED** 16 bit seed counter used by random generator to calculate a 32 bit random seed  
**CHKSUM** Transponder’s serial number 16 bit checksum storage  
**KEY** These bytes contain the 64 bit decryption key for each transponder
### TABLE 5: RAM MEMORY MAP (8-BIT BYTES)

<table>
<thead>
<tr>
<th>Address</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0D</td>
<td>FLAGS</td>
<td>Decoder flags</td>
</tr>
<tr>
<td>0E</td>
<td>ADDRESS</td>
<td>Address register – points to address in EEPROM</td>
</tr>
<tr>
<td>0F</td>
<td>TXNUM</td>
<td>Current transponder’s block index</td>
</tr>
<tr>
<td>10</td>
<td>XP_CNT</td>
<td>Transponder loop counter</td>
</tr>
<tr>
<td>08</td>
<td>OUTBYT</td>
<td>General data register, mask register used in decryption</td>
</tr>
<tr>
<td>09</td>
<td>TMP0</td>
<td>Temporary registers</td>
</tr>
<tr>
<td>0A</td>
<td>TMP1</td>
<td></td>
</tr>
<tr>
<td>0B</td>
<td>TMP2</td>
<td></td>
</tr>
<tr>
<td>0C</td>
<td>TMP3</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>CNT0</td>
<td>General loop counters</td>
</tr>
<tr>
<td>12</td>
<td>CNT1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CNT2</td>
<td></td>
</tr>
<tr>
<td>07</td>
<td>CNT3</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>CSR0</td>
<td>32-bit Code shift register used in decryption and key generation</td>
</tr>
<tr>
<td>15</td>
<td>CSR1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>CSR2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>CSR3</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>KEY7</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>KEY6</td>
<td></td>
</tr>
<tr>
<td>1A</td>
<td>KEY5</td>
<td></td>
</tr>
<tr>
<td>1B</td>
<td>KEY4</td>
<td></td>
</tr>
<tr>
<td>1C</td>
<td>KEY3</td>
<td></td>
</tr>
<tr>
<td>1D</td>
<td>KEY2</td>
<td></td>
</tr>
<tr>
<td>1E</td>
<td>KEY1</td>
<td></td>
</tr>
<tr>
<td>1F</td>
<td>KEY0</td>
<td></td>
</tr>
</tbody>
</table>

Many of the memory locations in RAM are used by multiple routines. A list of alternate names and functions are given in the table below.

<table>
<thead>
<tr>
<th>Address</th>
<th>Mnemonic</th>
<th>Also known as</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>DTA1</td>
<td>TEMP0</td>
<td>32-bit hop code register</td>
</tr>
<tr>
<td>19</td>
<td>DTA2</td>
<td>TEMP1</td>
<td></td>
</tr>
<tr>
<td>1A</td>
<td>DTA3</td>
<td>TEMP2</td>
<td></td>
</tr>
<tr>
<td>1B</td>
<td>DTA4</td>
<td>TEMP3</td>
<td></td>
</tr>
<tr>
<td>0D</td>
<td>EHOP3</td>
<td>ADDRESS</td>
<td></td>
</tr>
<tr>
<td>1C</td>
<td>EHOP2</td>
<td>TXNUM</td>
<td>Extended 32-bit buffer used during key generation as a 32-bit buffer</td>
</tr>
<tr>
<td>1D</td>
<td>EHOP1</td>
<td>TE_CNT</td>
<td></td>
</tr>
<tr>
<td>1E</td>
<td>EHOP0</td>
<td>CNT2</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>SER_0</td>
<td>CSR0</td>
<td>Shift register for unencrypted 32 bits received from transponder</td>
</tr>
<tr>
<td>16</td>
<td>SER_1</td>
<td>CSR1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>SER_2</td>
<td>CSR2</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>SER_3</td>
<td>CSR3</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE 6: MANUFACTURER’S CODE IN PROGRAM MEMORY (RETLW TABLE)

<table>
<thead>
<tr>
<th>Address</th>
<th>Mnemonic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>09B</td>
<td>MKEY_0</td>
<td></td>
</tr>
<tr>
<td>09C</td>
<td>MKEY_1</td>
<td></td>
</tr>
<tr>
<td>09D</td>
<td>MKEY_2</td>
<td></td>
</tr>
<tr>
<td>09E</td>
<td>MKEY_3</td>
<td></td>
</tr>
<tr>
<td>09F</td>
<td>MKEY_4</td>
<td></td>
</tr>
<tr>
<td>0A0</td>
<td>MKEY_5</td>
<td></td>
</tr>
<tr>
<td>0A1</td>
<td>MKEY_6</td>
<td></td>
</tr>
<tr>
<td>0A2</td>
<td>MKEY_7</td>
<td></td>
</tr>
</tbody>
</table>

64-Bit Manufacturer’s Code
(Used to generate decryption keys)

### TABLE 7: TIMING PARAMETERS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Typical</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output activation duration</td>
<td>500</td>
<td>ms</td>
</tr>
<tr>
<td>Transponder validation duration</td>
<td>330</td>
<td>ms</td>
</tr>
<tr>
<td>Erase all duration</td>
<td>4.2</td>
<td>s</td>
</tr>
<tr>
<td>Learn mode time-out</td>
<td>25</td>
<td>s</td>
</tr>
</tbody>
</table>
### TABLE 8: BILL OF MATERIALS FOR BASE STATION

<table>
<thead>
<tr>
<th>Item</th>
<th>Reference</th>
<th>Supplier</th>
<th>Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C1</td>
<td>Digi-Key</td>
<td>P3499</td>
<td>1.5nF, 630V Polypropylene Capacitor</td>
</tr>
<tr>
<td>2</td>
<td>C2</td>
<td>Digi-Key</td>
<td>P4797</td>
<td>10nF, B Series 100V Polyester Capacitor</td>
</tr>
<tr>
<td>3</td>
<td>C3</td>
<td>Digi-Key</td>
<td>P4787</td>
<td>1.5nF, B Series 100V Polyester Capacitor</td>
</tr>
<tr>
<td>4</td>
<td>C4</td>
<td>Digi-Key</td>
<td>P4773</td>
<td>100pF, B Series 100V Polyester Capacitor</td>
</tr>
<tr>
<td>5</td>
<td>C5</td>
<td>Digi-Key</td>
<td>P4797</td>
<td>10nF, B Series 100V Polyester Capacitor</td>
</tr>
<tr>
<td>6</td>
<td>C6, C7, C11, C14</td>
<td>Digi-Key</td>
<td>1210PHCT</td>
<td>0.1μF, 50V Axial Ceramic Capacitor</td>
</tr>
<tr>
<td>7</td>
<td>C8, C9</td>
<td>Digi-Key</td>
<td>P4016A</td>
<td>22pF, 50V Ceramic Disc Capacitor</td>
</tr>
<tr>
<td>8</td>
<td>C10</td>
<td>Digi-Key</td>
<td>P918</td>
<td>22μF 25V, KG Series Miniature Aluminum Electrolytic Capacitor</td>
</tr>
<tr>
<td>9</td>
<td>C12</td>
<td>Digi-Key</td>
<td>P6629</td>
<td>10μF 25V, Z Series Miniature Aluminum Electrolytic Capacitor</td>
</tr>
<tr>
<td>10</td>
<td>D1, D2, D3, D4</td>
<td>Digi-Key</td>
<td>P403</td>
<td>3mm Red Diffused High Brightness LED</td>
</tr>
<tr>
<td>11</td>
<td>D5, D6, D7</td>
<td>Digi-Key</td>
<td>IN4148DICT</td>
<td>100V, 500 MW Fast Switching Diode</td>
</tr>
<tr>
<td>12</td>
<td>L1</td>
<td>Coilcraft</td>
<td>DO5022P-105</td>
<td>1mH, DO5022 Series Surface Mount Power Inductors</td>
</tr>
<tr>
<td>13</td>
<td>Q1</td>
<td>Digi-Key</td>
<td>2N3904</td>
<td>Small Signal General Purpose Transistor</td>
</tr>
<tr>
<td>14</td>
<td>R1</td>
<td>Digi-Key</td>
<td>180R W-1</td>
<td>180R, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>15</td>
<td>R2, R6, R7, R13, R14, R15</td>
<td>Digi-Key</td>
<td>10K W-1</td>
<td>10k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>16</td>
<td>R3</td>
<td>Digi-Key</td>
<td>100R W-1</td>
<td>100R, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>17</td>
<td>R4, R5</td>
<td>Digi-Key</td>
<td>470K W-1</td>
<td>470k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>18</td>
<td>R8</td>
<td>Digi-Key</td>
<td>56K W-1</td>
<td>56k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>19</td>
<td>R9</td>
<td>Digi-Key</td>
<td>22K W-1</td>
<td>22k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>20</td>
<td>R10, R11, R12, R19</td>
<td>Digi-Key</td>
<td>1K W-1</td>
<td>1k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>21</td>
<td>R16</td>
<td>Digi-Key</td>
<td>120K W-1</td>
<td>120k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>22</td>
<td>R17</td>
<td>Digi-Key</td>
<td>390K W-1</td>
<td>390k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>23</td>
<td>R18</td>
<td>Digi-Key</td>
<td>2.7K W-1</td>
<td>2.7k, 5% Metal Oxide Film Resistor</td>
</tr>
<tr>
<td>24</td>
<td>SW1, SW2</td>
<td>Digi-Key</td>
<td>P8006S</td>
<td>Momentary Push-button Switch</td>
</tr>
<tr>
<td>25</td>
<td>U1</td>
<td>Digi-Key</td>
<td>MM74HC4060N</td>
<td>14 Stage Binary counter</td>
</tr>
<tr>
<td>26</td>
<td>U2</td>
<td>Digi-Key</td>
<td>PIC16C56-XP/P</td>
<td>8-Bit CMOS Microcontroller</td>
</tr>
<tr>
<td>27</td>
<td>U3</td>
<td>Digi-Key</td>
<td>93LC46B-I/P</td>
<td>2K CMOS Serial EEPROM</td>
</tr>
<tr>
<td>28</td>
<td>U4</td>
<td>Digi-Key</td>
<td>LM78L05ACH</td>
<td>+5V 100 mA Positive Regulator, TO-39</td>
</tr>
<tr>
<td>29</td>
<td>U6</td>
<td>Digi-Key</td>
<td>LM358N</td>
<td>Low Power Dual OP Amp</td>
</tr>
<tr>
<td>30</td>
<td>U7</td>
<td>Digi-Key</td>
<td>158-2021-2</td>
<td>IC 4.63V UP Reset Monitor SOT-23</td>
</tr>
<tr>
<td>31</td>
<td>Y1</td>
<td>Digi-Key</td>
<td>X911</td>
<td>4 MHz ZTA Series Ceramic Resonator</td>
</tr>
</tbody>
</table>

**Note:** Different value of the same order may have to be used to compensate for tolerance variations in R1, L1, and C1 to keep the peak-to-peak voltage across L1 at 100V.

### TABLE 9: BILL OF MATERIALS FOR TRANSPONDER

<table>
<thead>
<tr>
<th>Item</th>
<th>Reference</th>
<th>Supplier</th>
<th>Part Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U5</td>
<td>Microchip</td>
<td>HCS5410</td>
<td>KEELOQ Transponder IC</td>
</tr>
<tr>
<td>2</td>
<td>L2</td>
<td>Digi-Key</td>
<td>DN7437</td>
<td>1000 µH Power Axial Inductor</td>
</tr>
<tr>
<td>3</td>
<td>C13</td>
<td>Digi-Key</td>
<td>P4787</td>
<td>0.0015 µF 100V Poly B Series CAP</td>
</tr>
<tr>
<td>4</td>
<td>C14</td>
<td>Digi-Key</td>
<td>1210PHCT</td>
<td>0.1µF, 50V Axial Ceramic Capacitor</td>
</tr>
</tbody>
</table>
Note the following details of the code protection feature on PICmicro® MCUs.

- The PICmicro family meets the specifications contained in the Microchip Data Sheet.
- Microchip believes that its family of PICmicro microcontrollers is one of the most secure products 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 PICmicro microcontroller in a manner outside the operating specifications contained in the data sheet. The person doing so may be 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 product.

If you have any further questions about this matter, please contact the local sales office nearest to you.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip’s products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights.

Trademark

The Microchip name and logo, the Microchip logo, FilterLab, KEELOQ, microID, MPLAB, PIC, PICmicro, PICMASTER, PICSTART, PRO MATE, SEEVAL and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

dsPIC, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, microPort, Migratable Memory, MPASM, MPLIB, MPLINK, MPSIM, MXDEV, PICC, PICDEM, PICDEM.net, rPIC, Select Mode and Total Endurance are trademarks of Microchip Technology Incorporated in the U.S.A.

Serialized Quick Turn Programming (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.

© 2002, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

Printed on recycled paper.
AMERICAS
Corporate Office
2335 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
2335 West Chandler Blvd.
Chandler, AZ  85224-6199
Tel:  480-792-7966  Fax:  480-792-7456

Atlanta
500 Sugar Mill Road, Suite 200
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:  972-918-7423  Fax:  972-818-2924

Detroit
Tri-Atra Office Building
3255 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:  408-436-7955  Fax:  408-436-7955

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1XS, 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
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., Shanghai Liaison Office
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xing Road
Shanghai, 200001
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 Hong Kong Ltd.
Unit 901-6, Tower 2, Metropolis
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
Divoasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O’Shaunnessy 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, Shinya-kokoma
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, 189890
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
Lastrup høj 1-3
Ballersou 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:  49-89-6275-7500 Fax:  49-89-6275-7500

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