OVERVIEW
This application note describes a KEELoQ® code hopping receiver implemented on a Microchip mid-range microcontroller (PIC16F1827) using the Capture Compare mode. In previous implementations of the KEELoQ receiver, the receiving routine was developed around a fixed time base. The fixed-time-based method has the disadvantage of being very sensitive to variations in incoming data timing, which would often cause some received bits to be misread. In addition, having Timer0 used exclusively for this purpose can cause the firmware to be interrupted constantly every Timer0 period, which usually ranges around the tens of microseconds. Using the Capture mode eliminates these disadvantages by using the guard time to correctly adjust the bit timing and correctly identifying an incoming KEELoQ data transmission. Once the bit timing value is calculated, every change in level creates an interrupt and the time between interrupts is compared to the bit timing in order to identify if a valid bit has been received.

INTEGRATION WITH A KEELoQ SYSTEM
The module presented in this application note can be integrated with other modules, as described in AN744, “Modular Mid-Range PICmicro® KEELoQ® Decoder in C,” to complete the KEELoQ system as shown below in (Figure 1).

FIGURE 1: INTEGRATION WITH OTHER MODULES

Author: Enrique Aleman
Microchip Technology Inc.
The code hopping decoder as described in application note AN1248 and other KeeLoq decoder application notes is written in modular form. This makes it easier for the timer interrupt-based receiving routine to be directly replaced with the capture receiving routine.

DESCRIPTION

The capture receiving routine makes use of the CCP module available on various microcontrollers. The CCP module is configured in Capture mode so that it will generate an interrupt on a rising or falling edge of the incoming transmission. Timer1 counts the pulse duration between the rising and falling edge. The 16-bit value of Timer1 will be stored on the CCPR register when a falling or rising edge is detected at the capture pin. This strategy gives the advantage of having the exact measurement of the time of each pulse. Interrupts will only occur on each rising and falling edge.

A KeeLoq incoming PWM transmission normally looks as shown below in (Figure 2).

FIGURE 2: KeeLoq INCOMING PWM TRANSMISSION

The capture strategy is to detect the header portion of the transmission. The header length gives an exact measurement of what the Te timing is. After this, the Capture mode alternates between high-to-low and low-to-high detection to read each incoming bit length on the transmission. Once the receiver buffer is full, a flag is set that will prevent any more readings from the CCP until the flag is cleared by the main routine.

To prevent CCP interrupts from occurring when a valid transmission is not occurring, a separate timer (Timer 2) is used to measure and detect the guard time. This is the quiet time between transmissions. CCP interrupt is disabled during this time. When Timer2 overflows, the CCPIF flag is checked. If 4 Timer2 intervals have passed with no CCPIF flags set, then a valid transmission is about to come in and the CCP interrupt is enabled. If a guard time is detected, then the CCP interrupt is enabled.

The receiver routine consists of 4 states:

STATE 0: SYNC STATE

This state will measure the header portion of the incoming transmission. It will setup the capture to interrupt on a rising edge. In the interrupt routine, the capture time will be compared against a minimum and maximum header length. Once the length is determined to be valid, it is divided by 10 to get the exact value of Te (Timing element). This Te value will be used to determine the correct timing length of each bit.

STATE 1: MEASURE HIGH LEVEL

After the Sync state, the capture is set to interrupt on a falling edge. In the interrupt routine, the capture time will be compared against Te (+/- a tolerance) to determine if the bit is a 1, a 0 or invalid. If it is valid, then the bit is added to the data buffer, otherwise the state machine is reset (State 3).

STATE 2: MEASURE LOW LEVEL

After State 1, the capture is set to interrupt on a rising edge. This state will measure the time the transmission remains in low level. If a rising edge has been detected within the Te limits to determine if a valid bit has been received, then it is back to State 1 to read the next bit. Otherwise, the firmware checks for buffer full and determines if a complete packet has been received.

STATE 3: RESET

If any measurement during the previous states falls out of the range, then the buffer is cleared and the state machine is reset.

On the main routine, the firmware just waits for the Buffer Full flag. Once this is set, the main routine can decrypt and execute the decrypted message accordingly.
SUMMARY

The firmware supplied with this application note is precisely the same as supplied with AN1248, "PIC® MCU-Based KEELQ® Receiver System Interfaced Via I²C™", with the exception of the KEELQINT routine. This has been replaced with the Capture mode interrupt routine described in this application note, KLQRXINT_CCP.c. In addition, the same routine is supplied in assembly for use with previous KEELQ application notes.

Using the Capture mode provides an easier and a cleaner method of reading incoming bits on a transmission. The number of times the main process is interrupted is greatly reduced and the bit measurements are more exact and less sensitive to timing variations from transmitter to transmitter.
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.