

APPLICATION NOTE 3765

Remote Keyless Entry with the MAXQ3212

Abstract: The MAXQ family of microcontrollers is designed to be electrically quiet for the best integration with analog circuitry, including RF receivers. This article discusses the components needed to make a remote keyless entry (RKE) receiver using the MAXQ3212 microcontroller and the MAX1473 receiver.

Most automobiles these days ship with a factory-installed keyless entry system. But what if you wanted to add one to your older, hard-to-find-parts, 'classic' automobile?

The MAXQ® family of microcontrollers is designed to be electrically quiet for the best integration with analog circuitry, including RF receivers. This article discusses the components needed to make a remote keyless entry (RKE) receiver using the [MAXQ3212](#) microcontroller and the [MAX1473](#) receiver.

System Overview

A remote keyless entry system needs a key-fob transmitter and a receiver mounted somewhere in the car. **Figure 1** shows an overview of the system. (Note that the MAXQ3212 is a variant of the MAXQ3210, which could also be used for this project.)

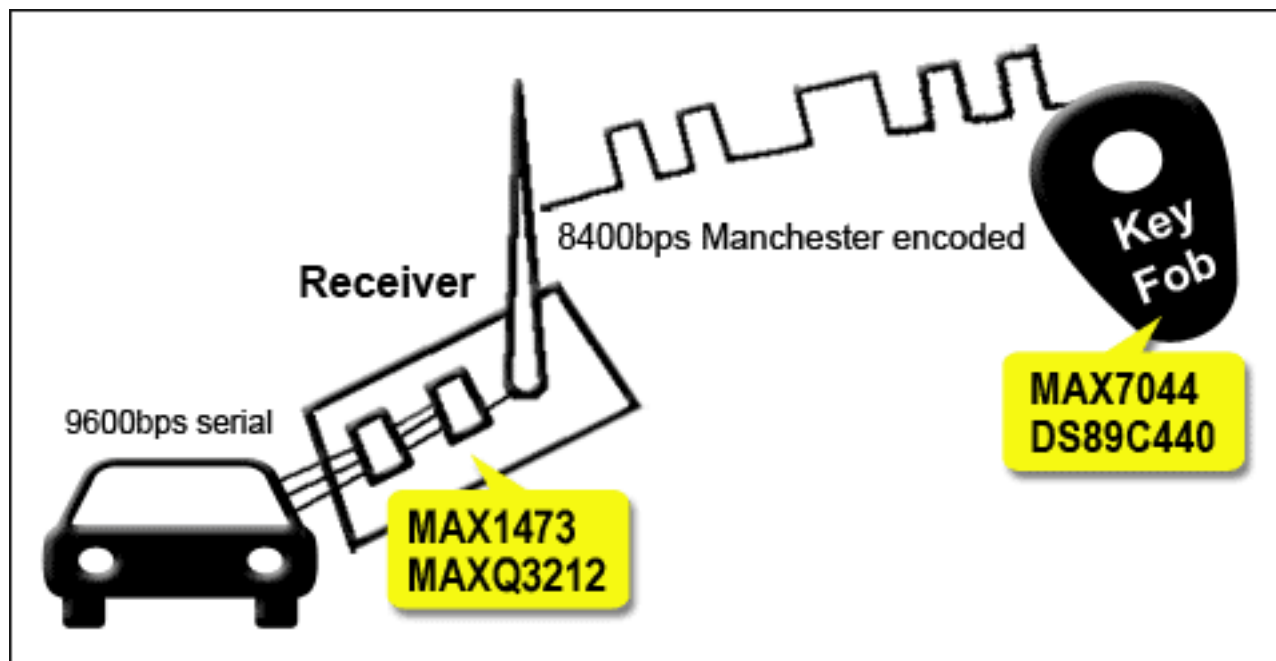


Figure 1. An RKE system needs both key fob and receiver.

Protocol

Keyless entry protocols differ vastly among car manufacturers, models, and model years; using a programmable microcontroller is therefore a good idea for this after-market project. For this article, we arbitrarily pick an 8400bps Manchester encoded digital data stream (see *Manchester Encoding* sidebar), transmitted at 433MHz using ASK. To use FSK or a different frequency, you must substitute a different receiver chip for the MAX1473.

Manchester Encoding

Each data bit is signified by at least one signal transition, making the data stream self-clocking. **Figure 2** shows the symbols for 0 and 1 if we choose a falling edge for 0 and a rising edge for 1.

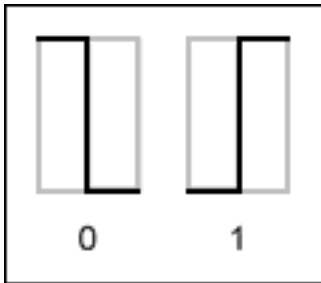


Figure 2. A falling edge encodes a 0, a rising edge encodes a 1.

Serial data is commonly transmitted LSB first. The ASCII character "A" (41h, 0100.0001b) is transmitted as 1000.0010b, as shown in **Figure 3**. The encoding can be derived by concatenating the symbols for 0 and 1 bits.

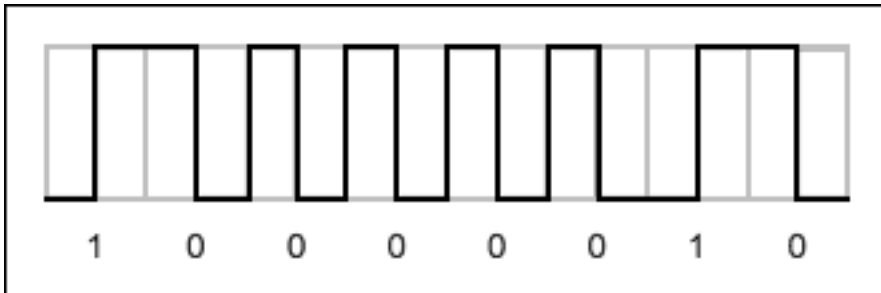


Figure 3. ASCII "A" is encoded by concatenating the symbols for 0 and 1.

Key Fob

Since we are concentrating on the receiver for this RKE, we will use two evaluation kits (EV kits) for our transmitter: DS89C450-KIT and [MAX7044EVKIT](#). These kits can be fitted into a case side by side with rechargeable batteries underneath (**Figure 4**). The key fob is somewhat oversized, but will serve well as a demonstration transmitter.

With an antenna, the range can be orders of magnitude higher than that of a standard key fob. *Note: Use this transmitter in a shielded lab environment only and follow standard procedures such as limiting the output power.*



Figure 4. The key-fob transmitter uses two EV kits side by side.

Data Stream

When a button is pressed on the key fob, it sends a synchronization preamble, followed by a transmitter ID, a counter, and button data (**Figure 5**). The transmitter repeats this sequence until the button is released, requiring a software debounce routine. In our example code, this is achieved by simply disabling the receiver for a short period.

Real-world systems also encrypt parts of the data to prevent vehicle theft. Decryption is usually handled by the car's body control module (BCM).

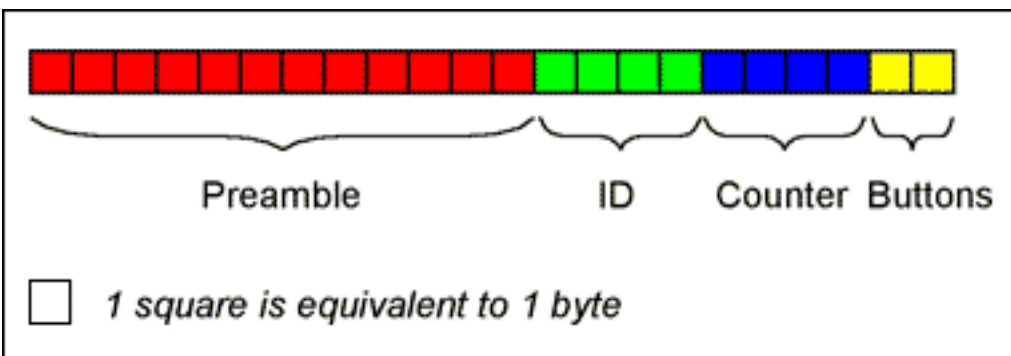
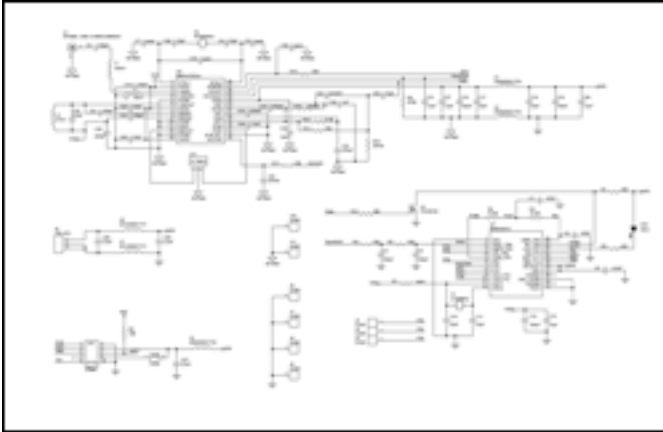


Figure 5. The key fob transmits a preamble, ID, counter, and the key code.

Receiver

The receiver consists of a MAXQ3212 8-bit microcontroller and a MAX1473 receiver mounted side by side, as shown in **Figure 6**. **Figure 7** shows the populated PC board. The wires on the side connect to the car's BCM. For this demonstration, we dedicated a port pin on the MAXQ3212 to transmit asynchronous serial data at 9600bps. **Figure 8** shows the MAXQ code for a simple bit-banged serial port.



[More detailed image \(PDF, 146kB\)](#)

Figure 6. The receiver consists of a MAXQ3212 and a MAX1473.

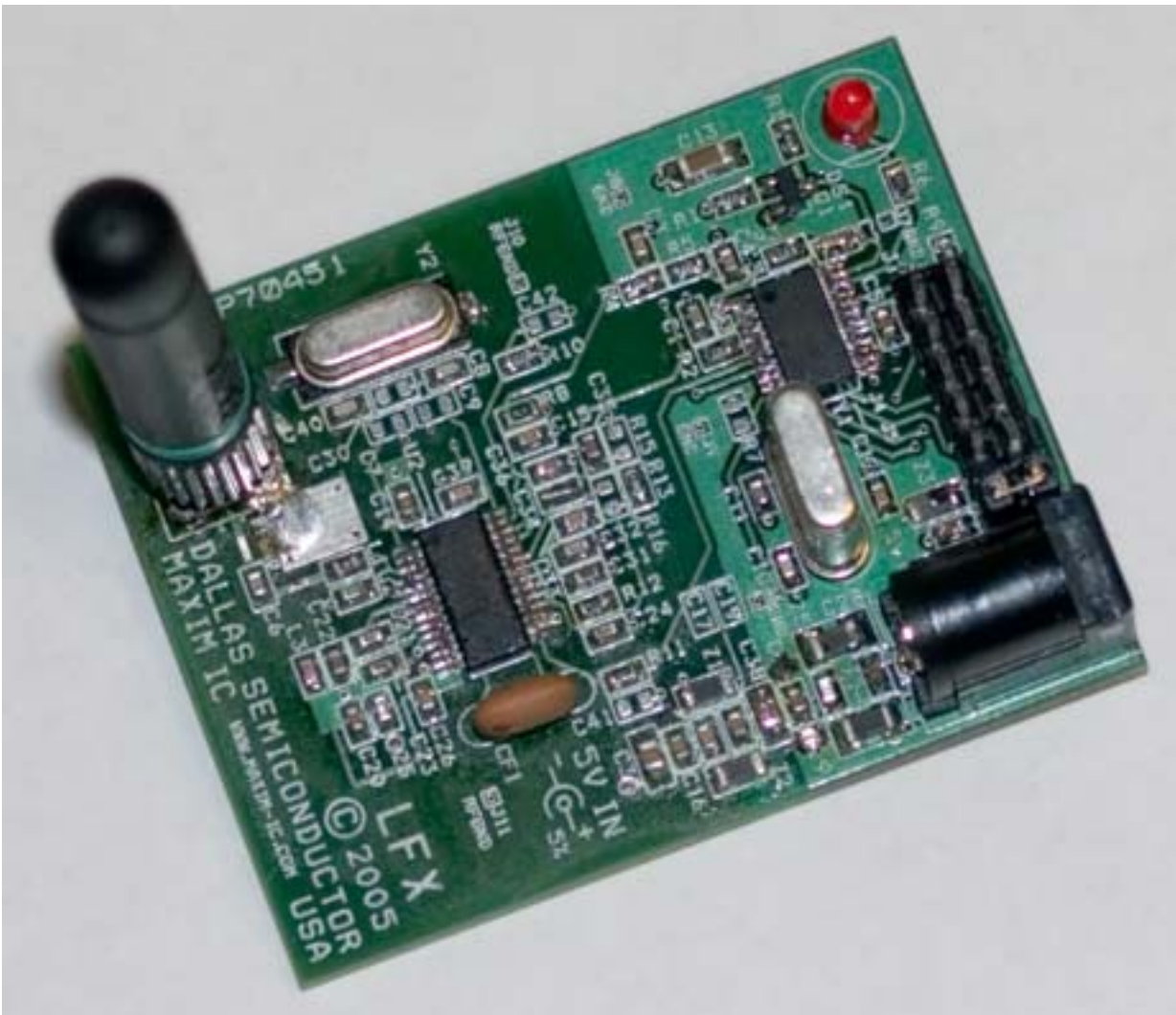


Figure 7. The RKE receiver board is populated with a MAXQ3212 and a MAX1473.

```

;
;*****
; Transmit a byte over the bit-bang serial port on P0.0.
; Baud rate 9600, 3.2768 MHz ==> 341 clock cycles per bit
;
SerialPortOutput:
    move LC[0], #9                ; Start bit + 8 data bits
    move C, #0                    ; First output bit to 0
serport_nextbit:
    move LC[1], #335             ; 6 cycles + delay loop == 335
    sjump C, serport_onebit      ; Is this a one bit?
    move PO0.0, #1              ; Set a zero bit
    sjump serport_delay         ; Jump to the delay
serport_onebit:
    move PO0.0, #0              ; Set a one bit
    nop                          ; Even out the timing
serport_delay:
    djnz LC[1], $               ; Single bit delay
    sr                           ; Shift for next bit, current in c
    djnz LC[0], serport_nextbit ; Next bit
    move PO0.0, #0              ; Send a stop bit
    move LC[1], #600            ; Extra long stop time
    djnz LC[1], $
    ret

```

Figure 8. Serial port output can be generated using a simple port pin.

Software

The receiver software measures the receive signal strength, waits for and synchronizes on the preamble, decodes the data stream, and transmits the values through the serial port.

Signal Strength Measurement

Measuring the signal strength is independent of our main task, the Manchester decoding. The MAX1473 receiver has an analog received signal-strength indicator (RSSI), which we measure. The MAXQ3212 features an analog comparator (comparing VREF and CMPI inputs), and can generate pulse-width modulation (PWM) on the timer output pins.

Figure 9 shows how to construct an ADC from the comparator and PWM. We feed the RSSI signal into the MAXQ3212 comparator's VREF. We then program the timer to PWM that, when suitably filtered, yields a DAC output. This DAC is connected to the other comparator input, CMPI. The comparator then compares the signal levels; if they match, we have a successful analog-to-digital conversion without a dedicated hardware ADC.

Rather than implementing successive approximation in software (which causes repeated large swings of the DAC signal, thus requiring longer settling times), we choose a slope-ADC. Starting at a reasonable minimum, the DAC output is ramped up until the comparator indicates a match.

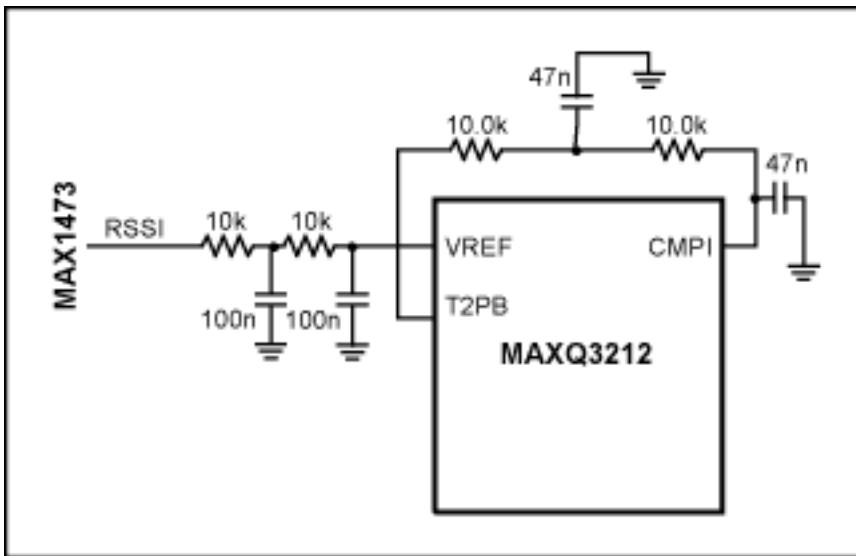


Figure 9. The MAXQ3212 comparator is used to measure the analog signal strength.

RF Signal Decoder

The MAX1473 supplies a digital signal output (DATAOUT). Due to always-present RF noise, this pin is continuously toggling, regardless of whether the key fob is actually transmitting or not. To distinguish this noise from a signal, the MAXQ microcontroller must implement a small state machine to measure the times between rising and falling signal edges, and to recognize the synchronization preamble.

The most efficient way to measure edge distances is by using interrupts. The MAXQ can be programmed to trigger an interrupt on a rising or a falling edge. To start a measurement, we set the interrupt to "rising edge." Once that edge is detected, we reset and start the timer and change the interrupt edge to "falling." On the falling edge, the interrupt handler reads the timer value. **Figure 10** shows a code fragment that reads and resets the timer and then toggles the interrupt sense.

If the edge distances match the data rate of 8400bps (plus/minus a reasonable tolerance), and the protocol-specific number of synchronization pulses is detected, the microcontroller's software state machine switches to receive mode and starts interpreting the rest of the packet data.

```

;
; We have an edge on P0.0. Grab timer value and toggle interrupt sense.
;
  move  a[T2_CURR], T2H      ; Read shifted by ~ 4 cycles
  move  T2H, #0              ; Reset timer
  move  acc, EI0
  xor   #5                   ; Flip edge phase and clear interrupt
  move  EI0, acc

```

Figure 10. Edge detection and timing can be entirely interrupt driven.

Conclusion

MAXQ microcontrollers are designed to be electrically quiet and integrate very well with Maxim RF components, without significantly degrading the RF signal. Code and schematics for the demonstration transmitter and receiver described in this article can be requested by email at micro.software@dalsemi.com.

More Information

For technical support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Automatic Updates

Would you like to be automatically notified when new application notes are published in your areas of interest?

[Sign up for EE-Mail™.](#)

Related Parts

MAX1473: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAXQ3210: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

MAXQ3212: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN3765, AN 3765, APP3765, Appnote3765, Appnote 3765

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal