QST

Official Journal of ARRL The national association for AMATEUR RADIO

Devoted entirely to Amateur Radio

www.arrl.org

June 2007

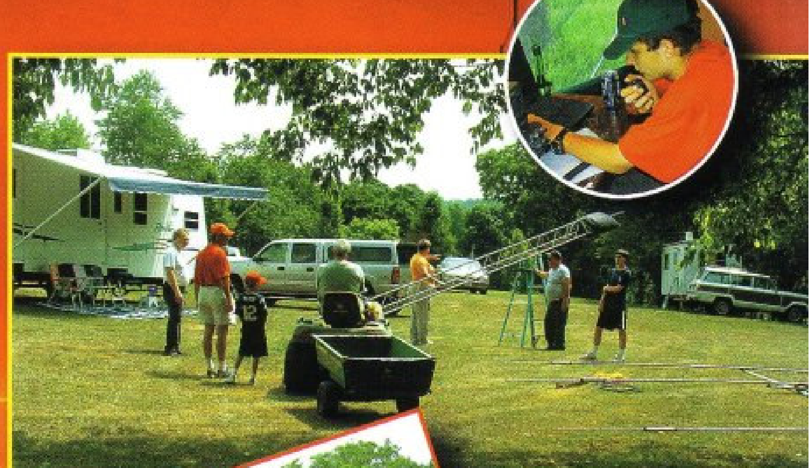
QST reviews:

ICOM IC-V85 2 Meter Handheld Transceiver

National RF Vector-Finder VHF Direction Finding System

Battery Powered Soldering Tools

Field Day is June 23-24

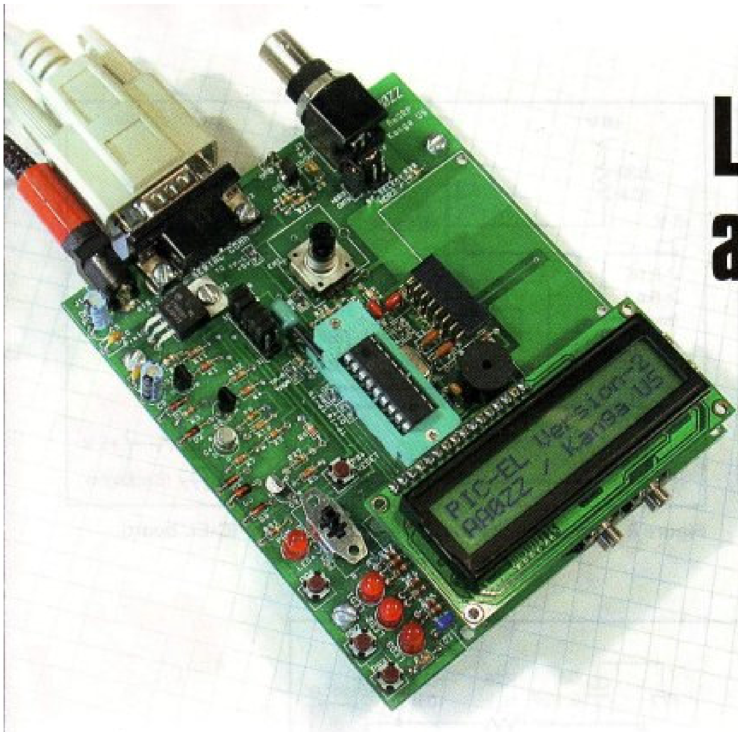


Inside:

A Field Day Logging Alternative

Tune Your Antenna System Audibly

Special 8-Page Field Day Section Inside!



Learning to PIC with a PIC-EL — *Part 2*

This easy to duplicate PIC proto board will make learning how to use these useful components easy!

Craig Johnson, AAØZZ

Last month I described the PIC-EL board that I developed for use as a companion board for an online course to teach beginners how to use peripheral interface controllers (PICs).[1] The course is written by John McDonough, WB8RCR, and additional lessons are still being developed. This article is intended to show how easy it is to understand how to use a PIC microcontroller. By looking at the various PIC-EL hardware components and the sample code or the Elmer-160 lessons, the amateur can learn how to use microcontrollers in many interesting applications.[2] The rewards are great.

PIC-EL Hardware

We looked at the PIC-EL computer interface last time and we also took a preliminary look at the various hardware components attached to the PIC microcontroller in the PIC-EL. Now we will take a closer look and show how you can make use of these components in your own projects. The project and demonstration portion of the PIC-EL board was designed to allow the experimenter to understand how a PIC microcontroller can be used in a variety of applications. It allows a person to progress from controlling very basic components to more advanced components and projects. PIC experimenters have an opportunity to use and understand the following hardware functions:

- An 18-pin PIC microcontroller (such as a 16F84/A, 16F628/A or 16F88).
- A 4 MHz crystal controlled clock.
- A two line (16 characters each) LCD display.
- A rotary encoder.
- Three general-purpose push buttons.
- A dedicated push button for master clear or reset of the PIC microcontroller.
- Three light emitting diodes (LEDs).
- A speaker with a transistor driver.
- All connections necessary to drive the NJQRP DDS daughtercard.[2]
- A stereo jack for connection to CW paddles.
- A stereo jack with transistor driver for transmitter keying.
- A transistor signal conditioner for converting low-level signals to levels required for PIC input detection.
- A multi-purpose BNC connector, jumper selectable to allow DDS output or signal input.
- A 2 × 6 pin header block to allow attachment of a foreign programmer.

What's Inside?

The PIC-EL schematic (see Figure 1 in Part 1) may look quite complicated because many of the PIC pins have multiple usages.[3] However, we can break down the schematic into its core pieces to understand the individual functions. This will also show how to use these basic components in other projects.

Last time we discussed the computer programming interface (the left side of the schematic). Now let's take a look at the various hardware components that are attached to the PIC and see how they work.

PIC System Clock

The system clock is generated by a 4 MHz crystal coupled with two 22 pF capacitors. A simple RC oscillator or the 16F628's internal oscillator could have been used instead. Since we are going to be experimenting with several timing-sensitive

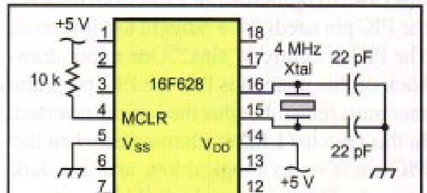


Figure 5 — The basic components necessary to run a PIC.

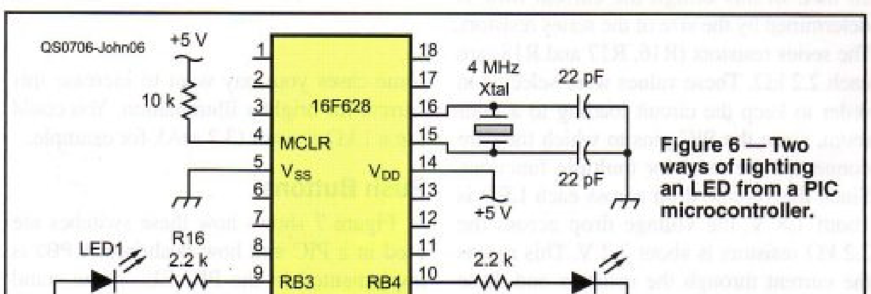


Figure 6 — Two ways of lighting an LED from a PIC microcontroller.

[1]Notes appear on page 36.

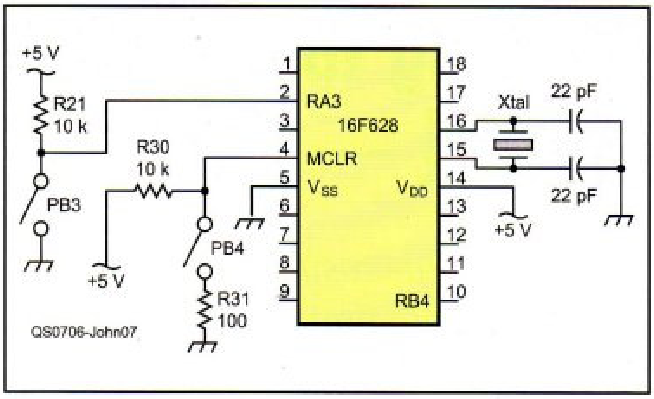


Figure 7 — The manner in which switches are used with a PIC and how PB3 is implemented in the PIC-EL.

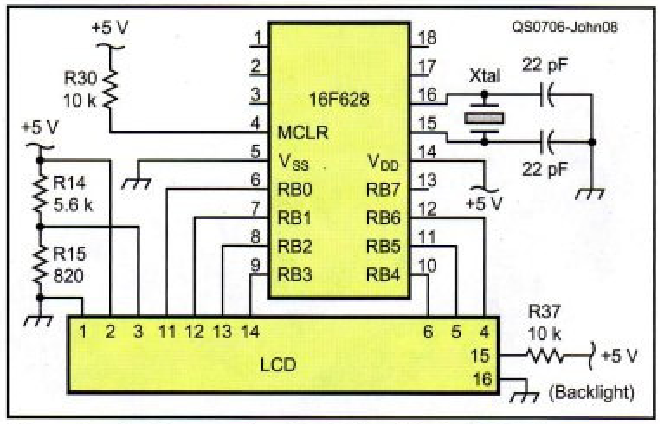


Figure 8 — Implementation of an LCD with the PIC-EL board.

projects such as frequency counters, an accurate clock is essential so a crystal was used.

The basic components necessary to run a PIC are shown in Figure 5. As you can see, it's really very simple.

LEDs

Two direct ways of lighting an LED from a PIC microcontroller are illustrated in Figure 6. The first is to connect a PIC output pin to a resistor and then to the anode of the LED with the cathode grounded. To light the PIC, the program needs to assert a logical high (+5 V nominal) on the output PIC pin. The PIC then provides the current to light the LED.

The other way is to connect a PIC output pin to a resistor and then to the cathode of the LED with the anode connected to +5 V. In this case, to illuminate an LED from the PIC, the PIC pin needs to be brought to a low level. The PIC is a current "sink." One minor drawback of this method is that the PIC programmer must remember that the logic is reversed. In this case the LED is illuminated when the PIC pin is set to a logical low, and it is dark when the PIC pin is at a logical high.

The method used in the PIC-EL board is to "sink" current with a PIC rather than to "source" the current.

Ideally, to illuminate an LED, the current flow through it should be between 1 mA and 20 mA. In this design the current flow is determined by the size of the series resistors. The series resistors (R16, R17 and R18) are each 2.2 kΩ. These values were selected in order to keep the circuit loading to a minimum, since the PIC pins to which they are connected are used for multiple functions. Since the voltage drop across each LED is about 1.8 V, the voltage drop across the 2.2 kΩ resistors is about 3.2 V. This means the current through the resistors and these LEDs is about 1.4 mA. This amount of current illuminates the LEDs sufficiently. In

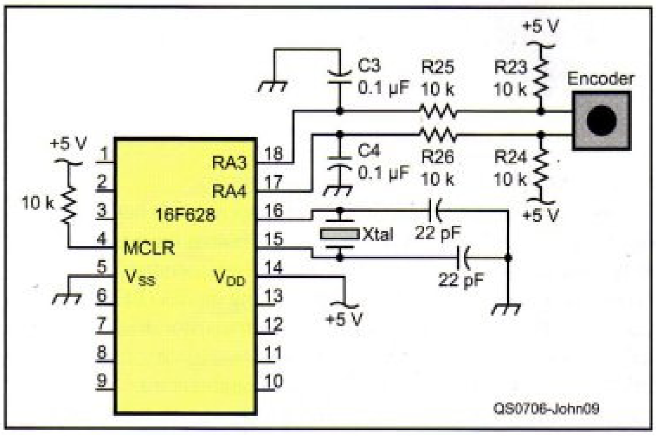
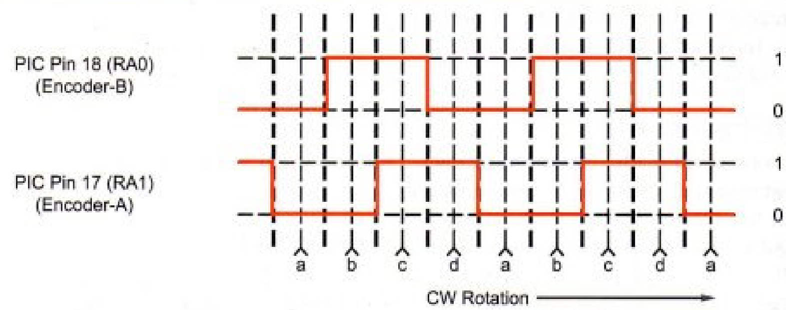


Figure 9 — A mechanical rotary encoder is attached to two PIC pins.



	RA0	RA1
At time a:	0	0
At time b:	1	0
At time c:	1	1
At time d:	0	1

Figure 10 — A graphical description of Gray code functionality.

QS0706-John10

some cases you may want to increase this current for brighter illumination. You could use a 1 kΩ resistor (3.2 mA), for example.

Push Buttons

Figure 7 shows how these switches are used in a PIC and how push button PB3 is implemented in the PIC-EL. Three stand alone normally open SPST push buttons (PB1, PB2 and PB3) are connected to PIC pins RA4, RA3 and RA2 in the PIC-EL. They can be used for any type of control functions that the programmer wants to use them. One other normally open SPST push button (PB4) is connected to the PIC's MASTER CLEAR pin and is used to reset the PIC program (make it start over). The three PIC pins that have general-purpose push buttons (PB1, PB2 and PB3) also have 10 kΩ pull-up resistors (R22, R21 and R34 respectively) attached to V_{DD}

(+5 V). In general, using pull-up resistors is a good design principle and provides a good "stiff" pull-up. In some cases, no pull-up resistor is used because some PIC pins (Port B in a 16F628) can have internal weak pull-ups activated via PIC software instructions. (This is done by executing a PIC instruction that clears bit 7 of the PIC's OPTION register.) In this mode, the PIC in effect puts a 50 kΩ resistor between each of these pins and +5 V. This means the PIC is able to source 0.1 mA of current on each of those pins. This is sufficient for a simple push button operation.

Note that the PIC's MASTER CLEAR pin (pin 4) has a 10 kΩ pull-up resistor (R30) to +5 V and is switched via a normally open SPST push button (PB5) to "near" ground. This is also illustrated in Figure 8. The pull-up resistor is essential here, since the PIC needs +5 V on MCLR for normal PIC operation. The 10 kΩ resistor is sufficient here, since the MASTER CLEAR pin draws very little current. Push button PB4 also has a 100 Ω resistor to prevent voltage transients from locking up the PIC.

Liquid Crystal Display (LCD)

The LCD panel used in the PIC-EL demonstration board has two rows of 16 characters each. It is a standard 5 × 10 dot matrix LCD that uses a Hitachi 44780 controller. It is attached in such a way that it minimizes interaction with other functions of the PIC-EL. In particular, the PIC programmer (also using PIC pins 12 and 13 — RB6 and RB7) still works properly when the LCD is connected in this manner.

The values of the voltage divider resistors (R14, R15) were selected to put the proper voltage on the LCD's contrast pin (pin 3). Also, the LCD backlight is activated with the resistor to +5 V connected to LCD pin 15 along with the ground connection to LCD pin 16. The backlight of the 2 × 16 LCD used in the PIC-EL kit draws about 75 mA. If an LCD with a different level of backlight current is used, the size of this resistor must be adjusted. Figure 8 shows how the LCD is implemented in the PIC-EL board.

Rotary Encoder

A mechanical rotary encoder is attached to two PIC pins, RA3 and RA4, as shown in Figure 9. R23 and R24 are typical pull-up resistors, since the rotary encoder is essentially just a pair of switches that open and close as the shaft rotates. Capacitors C3 and C4 are filters for removing noise that comes from contact bounce. The series resistors, R25 and R26, help in the signal filtering. Without the noise filtering, operation could be erratic.

For the mechanical encoder included in the PIC-EL kit, each of the signal lines produce 24 pulses per revolution, so a total of 96 up or down voltage transitions per revolution are generated that can be detected by the PIC microcontroller. The pulses of the two data lines are encoded in an overlapping Gray code such that an algorithm allows the PIC program to determine which direction the shaft is being turned. Figure 10 provides a simple explanation of how Gray code works. Table 1 illustrates one way to determine the rotation direction.

Speaker

A miniature speaker (SPKR-1) is attached to a PIC pin by way of a simple transistor (Q5) driver, as shown in Figure 11. The transistor driver gives more "punch" to the speaker than could be attained by directly attaching it to the PIC speaker output pin. The capacitor and diode in the path to the base of the driver transistor would be optional in most PIC speaker applications but are very important in the PIC-EL board because they prevent the speaker from being inadvertently left "on" if the PIC-EL application happens to leave that pin in a high state. Q5 acts as a switch, allowing current to flow through the speaker when Q5 is turned on and not flow when Q5 is turned off.

Pulses are generated by the PIC software and pass through capacitor C11 to turn Q5 on and off. The PIC program produces different tones by changing the frequency of the pulses it generates. Since audio tones are relatively low frequency and the PIC executes an instruction every microsecond, accurate delay loops can be designed to produce pulses with the desired frequencies.

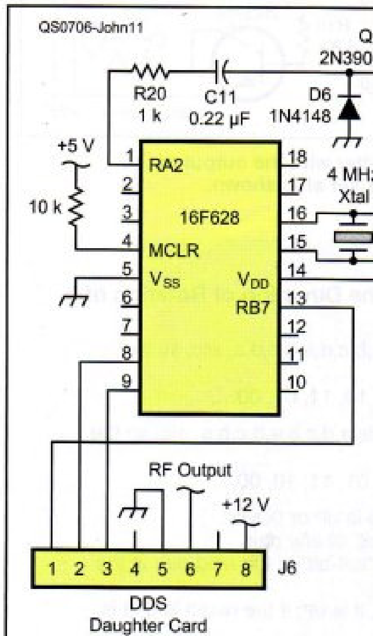


Figure 11 — Speaker connected to a PIC through a transistor driver. Daughter card connections also shown.

Signal Generation with the NJQRP DDS Daughtercard

The AmQRP DDS daughtercard can be plugged into the PIC-EL board by way of socket J6. Appropriate connections are made to the PIC and the required +12 V is also supplied to the daughtercard socket. Details of how the daughtercard operates can be found on the AmQRP Web page at **www.amqrp.org/kits/dds60/**. The PIC connections to the DDS daughtercard are illustrated on the bottom of Figure 11.

The output of the daughtercard is supplied back to pin 6 of socket J6. The PIC microcontroller can drive the DDS daughtercard to produce an amplitude of approximately 600 mV with a frequency within the range of 0 to 30 MHz or 60 MHz, depending on which version of the DDS daughtercard you have.

Signal Conditioner

A signal conditioner, shown in Figure 13, is provided to increase small amplitude signals to voltage levels detectable by the PIC. The output amplitude of the DDS daughtercard is too low to be fed directly back into a PIC pin for the demonstration of frequency counting. To make this work, the amplitude is increased by the signal conditioner circuitry. Notice that this conditioner is not a linear amplifier in that it does not attempt

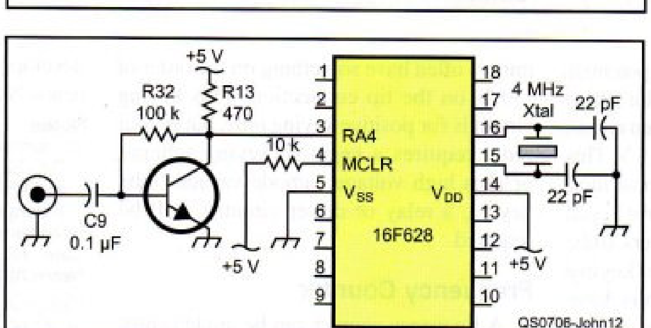


Figure 12 — Signal conditioner to increase small amplitude signals to voltage levels detectable by the PIC.

to keep a distortion-free sinewave output. For purposes of frequency measurement, a square wave is just as good as a sine wave.

Note that header HDR2 is used to select the source of the signal that goes into the conditioner. In one position, the output of the DDS daughtercard is fed into the conditioner while in another configuration a signal from an external source can be brought into the PIC-EL board via BNC connector J7 and routed through the conditioner before going to the PIC.

CW Paddle Input Via a Stereo Jack

CW paddles can be attached to the PIC by way of a ⅛ inch stereo jack as shown in Figure 13. The jack connects one of the paddle connections to the stereo plug's tip and the other to the ring. Both pins have pull-up resistors (R21 and R22) connected to +5 V. The PIC is then able to detect the paddle closures just as if they were two SPST switches. A demonstration example of a CW keyer is available on my Web site or the FILES section of the PIC-EL YAHOO group.

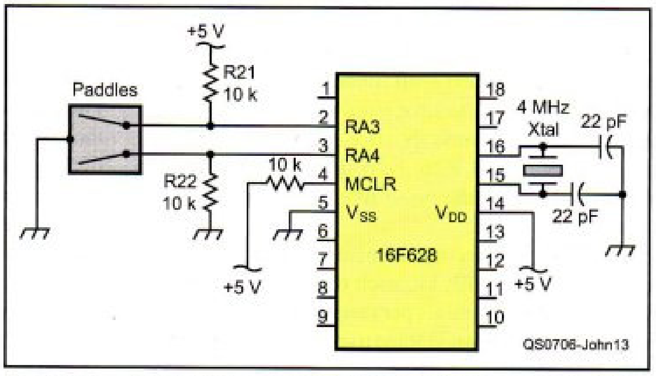


Figure 13 — CW paddles attached to the PIC by way of a ⅛ inch stereo jack.

Transmitter Keying Via a Stereo Jack

Figure 14 shows how to key a transmitter with the output of the demonstration keyer. Another ⅛ inch stereo jack is provided for this purpose. The output of a PIC pin goes to a transistor driver which then goes to the tip connection of the stereo jack. When keyed, the transistor driver drives the voltage at the tip connection from approximately 5 V to ground potential. When the PIC pin is not keyed the tip-to-ground connection looks like an open circuit so the tip remains at approximately 5 V. This keying mechanism will work for most modern rigs because they employ positive keyed transmitters. Some early transmitters (tube type in particular) used negative keying. Modern positive keyed transmitters have approximately +3 to +5 V on the tip connection with the radio keyed by connecting this pin to ground. Negative keying transmitters often have something on the order of –30 V on the tip connection. This keying circuit is for positive keying only, but if your radio requires a negative keying scheme, or uses high voltage cathode vacuum tube keying, a relay or driver circuitry will be required.

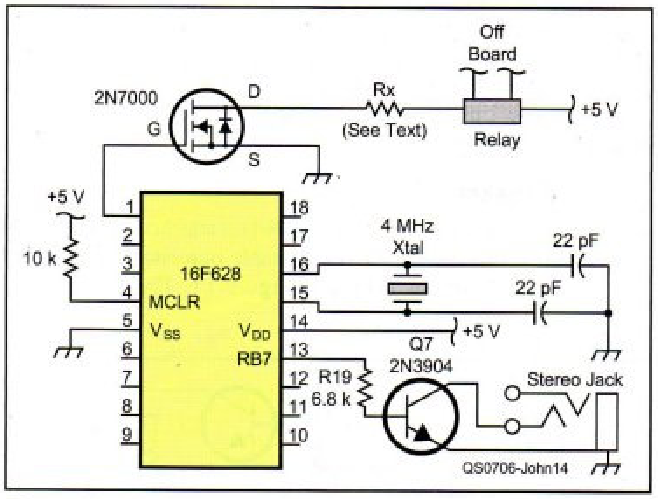


Figure 14 — Keying a transmitter with the output of the demonstration keyer. Relay driver also shown.

Table 1

A Method to Determine the Direction of Rotation of a Rotary Encoder

Going UP, the sequence is a,b,c,d,a,b,c,d,a, etc, so the sequence is:

00, 10, 11, 01, 00, 10, 11, 01, 00....

Going DOWN, the sequence is a,d,c,b,a,d,c,b,a, etc, so the sequence is:

00, 01, 11, 10, 00, 01, 11, 10, 00....

To determine if the sequence is UP or DOWN:

1) Take the "right-bit" of any pair.
2) XOR it with the "left-bit" of the next pair in the sequence.
3) If the result is 1, it is UP; if the result is 0, it is DOWN.

Frequency Counter

A frequency counter can be implemented in the PIC-EL by using the signal conditioner, described earlier. The conditioner feeds its output into PIC pin 3 (RA4/T0CKI). This PIC pin may be configured to be a general purpose input/output pin, but also has the unique characteristic of being configured as a counter input to PIC register TMR0. The TMR0 register is used by frequency counter applications.

How to Drive a Relay from a PIC

The PIC-EL is not set up to demonstrate relay control. In order to drive a device that requires more current than the 20 mA or so a PIC pin can deliver, a driver circuit is required. The top of Figure 14 shows an example of how it can be done. Resistor R_x in series with the relay coil must be sized to pass the proper amount of current. The 2N7000 MOSFET is a good general purpose device but in some applications, such as for driving relays or LCD backlight activation, an IRLML2502 is an even better choice, since its drain to source resistance when turned on is about 0.045 Ω. In contrast, the on R_{DS} of a 2N7000 is between 2 and 5 Ω.

Questions, Support?

For up-to-date details and documentation regarding this project, please see my Web page, **www.cbjohns.com/aa0zz**, the YAHOO group PIC-EL or e-mail me directly at **aa0zz@cbjohns.com**.

Conclusion

I hope you can see that getting started with PIC programming is not terribly difficult and that there are many useful things you can do with them. We have only scratched the surface, of course. The PIC-EL is a very convenient platform for experimenting but after that, it's up to you to develop your own projects with the components. Now it's time for you to try it!

Notes

[1]C. Johnson, AAØZZ, "Learning to PIC with a PIC-EL — Part 1," *QST*, May 2007, pp 37-42. Available on the ARRLWeb at **www.arrl.org/files/qst-binaries/Johnson0507.pdf**.

[2]**www.amqrp.org/elmer160**.

[3]See Note 1.

[4]**www.njqrp.org/**.

Craig Johnson, AAØZZ, can be reached at 4745 Kent St, Shoreview, MN 55126 or **aa0zz@cbjohns.com**. *His Web site is* **www.cbjohns.com/aa0zz**.