

The PIC-EL (Version 2)

A PIC Programmer and Test/Demonstration Board

By Craig Johnson, AA0ZZ

aa0zz@cbjohns.com

www.cbjohns.com/aa0zz



TABLE OF CONTENTS

1	Introduction	2
2	Hardware Description.....	3
2.1	PIC programmer	3
2.2	Project/Demonstration.....	4
2.2.1	PIC system clock (crystal).....	5
2.2.2	LEDs.....	5
2.2.3	Pushbuttons.....	6
2.2.4	LCD	7
2.2.5	Rotary Encoder.....	8
2.2.6	Speaker	9
2.2.7	Signal generation with the NJQRP DDS Daughtercard.....	10
2.2.8	Signal Conditioner.....	10
2.2.9	CW paddle Input via a Stereo Jack.....	11
2.2.10	Transmitter Keying via a Stereo Jack.....	11
2.2.11	Frequency Counter	12

2.2.12	How to Drive a Relay from a PIC	12
3	PIC-EL Assembly.....	13
3.1	Parts Identification.....	13
3.2	Test Points	13
3.3	Assembly	14
3.4	Final installation and Basic PIC-EL Test	16
4	PIC-EL Computer Interface	17
4.1.1	Installing FPP	17
4.1.2	Installing FPPA	17
4.1.3	Using FPP / FPPA	17
5	PIC-EL Diagnostic Program.....	19
5.1	Test LEDs.....	19
5.2	Test Pushbuttons.....	19
5.3	Test Speaker	19
5.4	Test Encoder.....	19
5.5	Test Paddles.....	20
5.6	Test Transmitter Keying.....	20
5.7	Test DDS-30.....	20
5.8	Test DDS-60.....	20
6	Elmer-160 Lessons	20
7	Support	20
8	Appendix A – Schematic.....	21
9	Appendix B – Parts List	22
10	Appendix C – Parts Layout	24

1 Introduction

PIC microcontrollers have been around for many years and are still very popular, in spite of the arrival of a number of new microcontroller families. One of the reasons PICs are so popular is because there are so many examples of projects that use them. Unfortunately, much of the code is complicated and hard to understand, especially for beginners. On the other hand, with a bit of determination and perseverance, trademarks of amateurs today, many have found PICs to be very useful in a variety of applications such as CW keyers, frequency counters, Direct Digital Synthesis (DDS) controllers, receivers and/or transmitters controllers, repeater controllers, power/SWR meters, antenna control, and station control. The list is endless.

I developed the PIC-EL board in late 2004, with consulting help from AmQRP club members, George Heron, N2APB, Joe Everhart, N2CX, John McDonough, WB8RCR, Earl Morris, N8ERO, and Jim Kortge, K8IQY. This was soon after John McDonough volunteered to teach an on-line course to help beginners learn how to use PICs. We wanted to make a project board that the students could use to load the code they just wrote into the PIC and then to immediately try it out. (Immediate feedback does wonders in keeping the experimenter motivated.) Within a three-month period I developed several prototypes, tested them, and finally produced a printed circuit board that the AmQRP club could use in a kit. Hundreds of these kits were distributed by the club in early 2005 and the reception was very positive.

Since John called his on-line PIC course Elmer-160, I named this board the PIC-EL. The standard PIC for the PIC-EL is a 16F628, a very common, mid-range PIC, selected for this project because of its balance of architectural simplicity, power, and low cost. It is one of the most common PICs used in amateur applications these days. The Elmer-160 lessons start with an even lower-end PIC, the 16F84,

because it is even easier to for beginners to understand. The 16F628 is “nearly compatible” with the 16F84 in that only a few lines of code need to be changed. The PIC-EL can use either of these PICs interchangeably. However, the diagnostic program, pre-loaded on a 16F628 that is supplied with the kit, is slightly too large for a 16F84.

PIC-EL kit sales by the AmQRP club ended after a few months. Recently I updated the board to replace obsolete parts, correct a few deficiencies, and make the kits available on a long-term basis. The result is the PIC-EL Version 2.

2 Hardware Description

Appendix A has the schematic of the PIC-EL board. The PIC-EL board has two parts - a PIC programmer and a test / demonstration portion.

2.1 PIC programmer

The PIC serial port programmer design is an adaptation of the classic programmer by David Tait. See Tait’s PIC-related home page on the Internet at: <http://people.man.ac.uk/~mbhstdj/piclinks.html>

The PIC programmer has an RS232 (DB9F) serial port connection. Pin assignments are as follows:

- Data-Out is on RS-232 pin 4 (DTR)
- Clock is on RS-232 pin 7 (RTS)
- Data-In is on RS-232 pin 8 (CTS)
- MCLR is on RS-232 pin 3 (TD)

The RS-232 lines representing Data-Out and Clock are inverted and converted to PIC signal levels via NPN (2N3904) transistor switches (Q1 and Q2) before going to the PIC microcontroller. Thus, for these two computer output signals passed via the serial port’s DTR and RTS lines, low serial port signal levels (–12v) get converted to high PIC signal levels (+5v) and high serial port signal levels (+12v) get converted to low PIC signal levels (0v).

The MCLR signal is another PC output signal, and it is sent to the programmer via the serial port’s TD line. This signal is generated by the PC programming software in order to put the PIC in high-voltage programming mode. In this case, the PIC needs the two levels to be zero and approximately +12v. Transistor Q3 operates in a manner that is very similar to Q1 and Q2, except its collector voltage is higher. As before, it inverts the serial port signal levels, but this time the low level (–12v) results in a MCLR signal of approximately 12.5v (depending on the power supply voltage) and the serial port high level (+12v) results in a MCLR signal of approximately zero volts.

One 2N2907 (PNP) transistor switch (Q4) is used in the programmer. It brings the Data-In signal from the PIC microcontroller to the serial port connector. Again, its high and low levels are inverted. Since the serial port expects to see voltages which are in the range of +4v to +12v (high) and –4v to –12v (low), Q4, two resistors (R7 and R8), a diode (D4), and a negative voltage (TD), are used to adjust the levels. When the PIC generates a low level (0v), the voltage at the base of Q4 will be about 5v, as supplied by the 5v regulator.

The voltage at RS-232 pin 4 (TD) is –12v when in active programming mode. This level gets inverted by transistor Q2 to produce +12.5v as Vp_{gm} to the HDR1 header and then to PIC pin 4 (MCLR). This meets the PIC requirement for high-voltage programming. High voltage programming requires the programmer to quickly raise the voltage on the MCLR pin from zero to at least 4.5v higher than the PIC’s run voltage (V_{dd}) of 5v.

No provision is made at this time for connecting to a USB port or a parallel port to the programmer. However, external USB or parallel port programmers can be attached to the PIC-EL board via HDR-1. The PIC-EL board cannot be run directly from a PC parallel port because of the difficulty is getting the Data In signal to work properly. The parallel port's voltages are 0v for a low level and +5v for a high level. Since the TD signal never goes negative, the circuitry that was used to pull the Q4 collector below zero for a low level does not work. In fact, the low level only goes down to approximately 3.2v when TD is asserted (0v). Additional circuitry could be added, but it is not included in this PIC-EL board.

How about a USB-to-serial converter between the computer and the PIC-EL's serial interface? It would be nice, but so far we have not found one that works in this application. The reason is because RS232 ports are intended to perform asynchronous serial communications while we are directly controlling the serial port's control lines and doing synchronous communications to the PIC-EL. The PIC-EL requires a perfect 1-to-1 correspondence between signals generated by the computer programmer software and the signals presented to the PIC-EL serial interface. Since USB-to-serial interfaces do not allow this closely controlled synchronous activity, it cannot interface properly with the PIC programmer and does not work properly.

2.2 Project/Demonstration

The project / 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 the person to progress from controlling very basic components to more advanced components and projects.

In RUN mode, PIC experimenters have an opportunity to use and understand the following hardware functions:

1. A 18-pin PIC microcontroller (16F84/A, 16F628/A, 16F88, etc)
 - Includes 4 MHz crystal
2. A 2x16 LCD (two lines of 16 characters)
3. A rotary encoder (ENC-1)
4. Three general-purpose pushbuttons (PB1 through PB3)
5. A dedicated pushbutton (PB4) for master clear (reset) of the PIC microcontroller
6. Three LEDs (LED1 through LED3)
7. A speaker (SPKR-1) with transistor driver.
8. All connections necessary to drive the NJQRP DDS Daughtercard (DDS-30 or DDS-60)
9. A stereo jack for connection to CW paddles.
10. A stereo jack with transistor driver for transmitter keying
11. A transistor "conditioner" for converting low-level signals to levels required for PIC input detection.
12. A multi-purpose BNC connector
 - Selectable via a jumper at header HDR-2
 - Allow DDS output to be routed to the BNC
 - Allow DDS output to be routed to a "conditioner" and then to a PIC input pin

- Allow an outside signal source to be brought in to the “conditioner” and then to the PIC input pin 13. A 2x6 pin header block (HDR1 - CONFIG)

- Allows attachment of a “foreign programmer” to this PIC project board
- Allows attachment of this programmer to a “foreign project board”

The PIC-EL Schematic (Appendix A) may look quite complicated because many of the PIC pins have multiple usages. 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.

NOTE: The PIC pin connections for components in the following sections DO NOT NECESSARILY correspond to the PIC pins used in the PIC-EL. PIC pins were selected here for ease of illustration. Refer to the schematic in Appendix A for the PIC-EL PIC pin usage.

2.2.1 PIC system clock (crystal)

The system clock is generated by a 4 MHz crystal with two 22 pf capacitors. A simple RC oscillator or the 16F628’s internal oscillator could have been used instead; however, since we are going to be experimenting with several timing-sensitive projects such as frequency counters, an accurate clock is essential so a crystal was used.

Figure 1 shows the basic components that are necessary to run a PIC. As you can see, it’s really very simple.

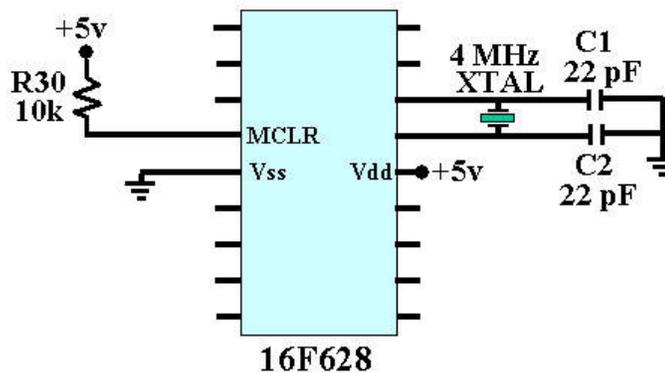


Figure 1 – Basic PIC connections

2.2.2 LEDs

Two direct ways of lighting an LED from a PIC microcontroller are illustrated in Figure 2. 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 (+5v nominal) on the output PIC pin. The PIC “sources” 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 +5v. 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; i.e., the LED is illuminated when the PIC pin is set to a logical low, and it is dark when the PIC pin is logical high.

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

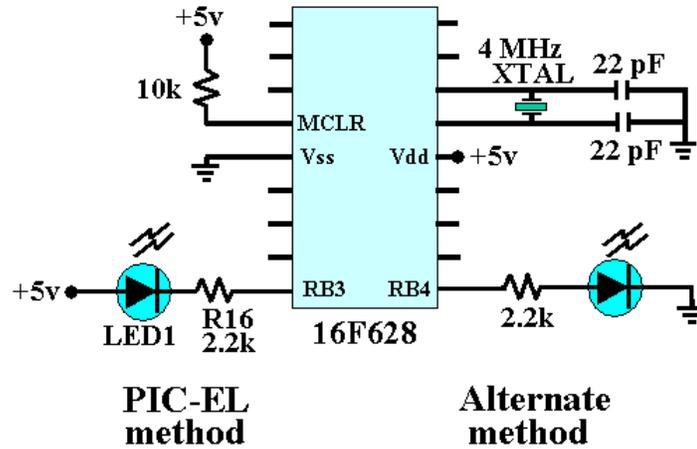


Figure 2 – Lighting an LED

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 2.2k ohms. 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 1.8v, the voltage drop across the 2.2k resistors is about 3.2v. This means the current through the resistors and these LEDs is about 1.4 ma. This amount of current illuminates the LEDs sufficiently. In some cases you may want to increase this current for brighter illumination. You could use a 1k resistor (3.2 ma), for example.

2.2.3 Pushbuttons

Figure 3 shows how these switches are used in a PIC and how PB3 is implemented in the PIC-EL.

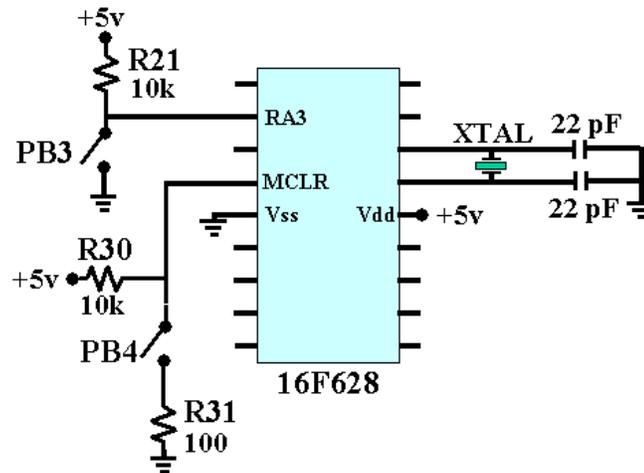


Figure 3 – General Usage and Master Clear Pushbuttons

Three stand-alone normally-open SPST pushbuttons (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 for. One other normally-open SPST pushbutton (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 pushbuttons (PB1, PB2 and PB3) also have 10k pull-up resistors (R22, R21 and R34 respectively) attached to Vdd (+5v). 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 which clears bit 7 of the PIC’s OPTION register.) In this mode, the PIC in effect puts a 50k ohm resistor between each of these pins and +5v. This means the PIC is able to source .1 ma of current on each of those pins. This is sufficient for a simple pushbutton operation.

Note that the PIC’s Master Clear pin (pin 4) has a 10k ohm pull-up resistor (R30) to +5v and is switched via a normally-open SPST pushbutton (PB5) to “near” ground. This is also illustrated in Figure 5. The pull-up resistor is essential here, since the PIC needs +5v on MCLR for normal PIC operation. The 10k resistor is sufficient here, since the Master Clear pin draws very little current. Pushbutton PB4 also has a 100 ohm resistor to prevent voltages transients from locking up the PIC.

2.2.4 LCD

The LCD used in the PIC-EL demonstration board has two rows of 16 characters. It is a standard 5x10 dot matrix LCD that has a standard 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 +5v connected to LCD pin 15 along with the ground connection to LCD pin 16. The backlight of the 2x16 LCD used in the PIC-EL kit draws about 75 ma; if the LCD used in a different application draws a different amount of backlight current, the size of this resistor must be adjusted.

Figure 4 shows how the LCD is implemented in the PIC-EL board.

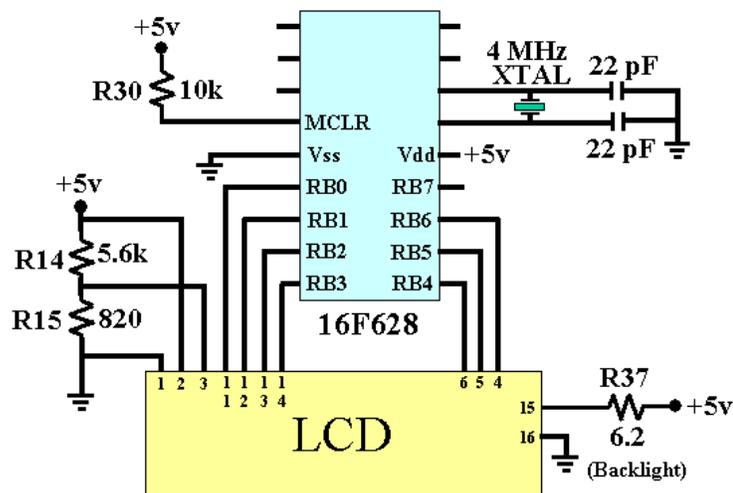


Figure 4 – LCD

2.2.5 Rotary Encoder

A mechanical rotary encoder is attached to two PIC pins, RA3 and RA4, as shown in Figure 5. 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 which comes from contact bounce. The series resistors, R25 and R26, help in the signal filtering. Without the noise filtering, operation is very erratic.

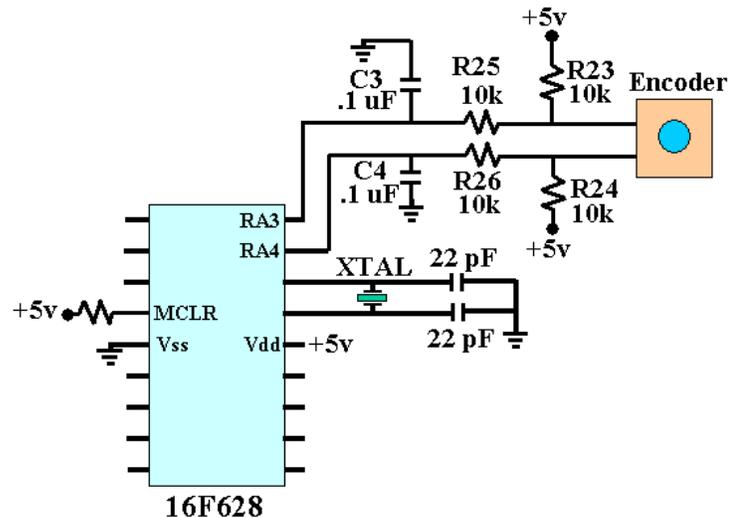
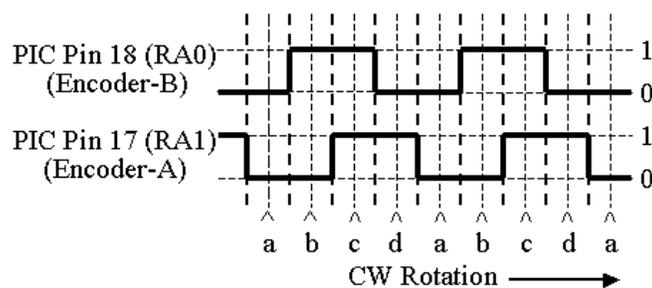


Figure 5 – Rotary Encoder

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 and can be detected by the PIC microcontroller. Since the pulses of the two data lines overlap (gray code), the PIC program can use an algorithm to determine which direction the shaft is being turned. See Figure 6 for a simple explanation of how this works.



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

Figure 6– Gray Code

Determining the direction the encoder is being turned is done by a variety of software algorithms. Figure 7 illustrates one way to determine the direction.

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 etc.

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, etc.

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

Figure 7– Determining Encoder Direction

2.2.6 Speaker

A miniature speaker (SPKR-1) is attached to a PIC pin by way of a simple transistor (Q5) driver, as shown in Figure 8.

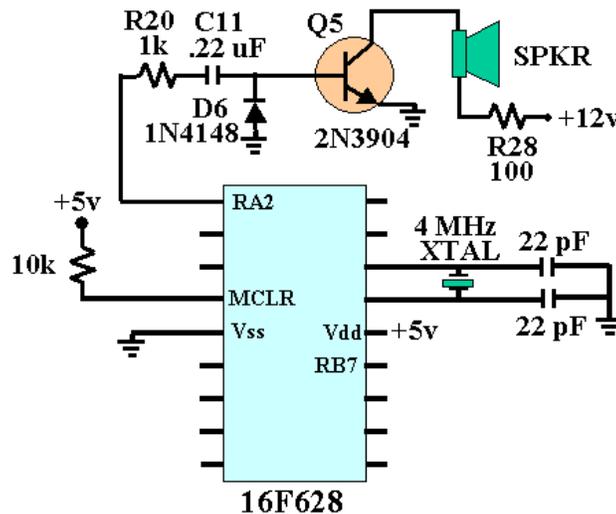


Figure 8- Speaker

The transistor driver gives more “punch” to the speaker than could be attained by directly attaching it to the PIC pin to the speaker. 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 duration 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 durations.

2.2.7 Signal generation with the NJQRP DDS Daughtercard

The AmQRP DDS Daughtercard can be plugged into the PIC-EL board by way of a socket (J6). Appropriate PIC connections are made to the PIC and the required +12v is also supplied to the Daughtercard socket. Details of how the Daughtercard operates can be found on the AmQRP web page at: <http://www.amqrp.org/kits/dds60/>.

The PIC connections to the DDS daughtercard are illustrated in Figure 9

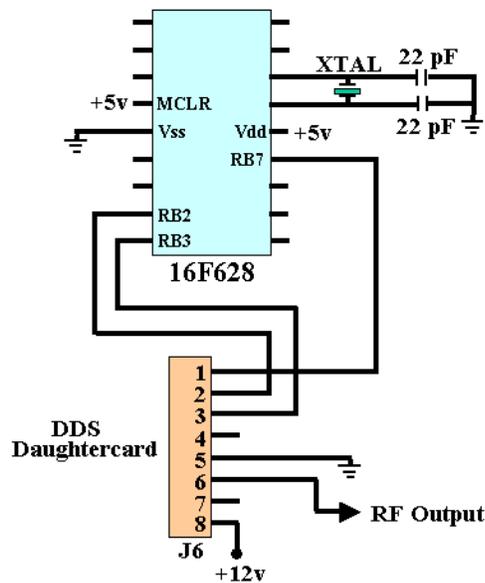


Figure 9- DDS Daughtercard

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

2.2.8 Signal Conditioner

A signal conditioner, shown in Figure 10 is provided to increase small amplitude signals to voltage levels which are 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 to keep a distortion-free sine-wave output. For purposes of frequency measurement, a square wave would be just as good as a sine wave.

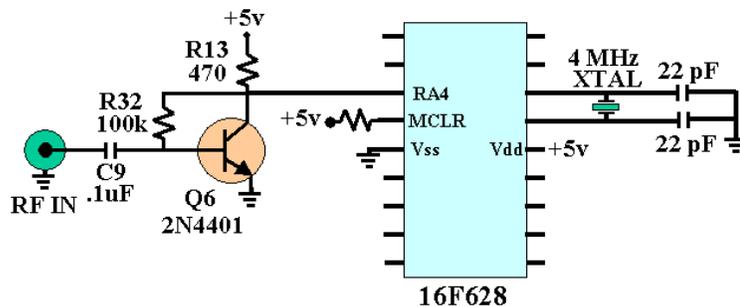


Figure 10– Signal Conditioner

Note that a header HDR2 is used to select the source of the signal which 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 the BNC connector (J7) and routed through the “conditioner” before going to the PIC.

2.2.9 CW paddle Input via a Stereo Jack

The CW paddles are attached to the PIC by way of a 1/8” stereo jack. See Figure 11

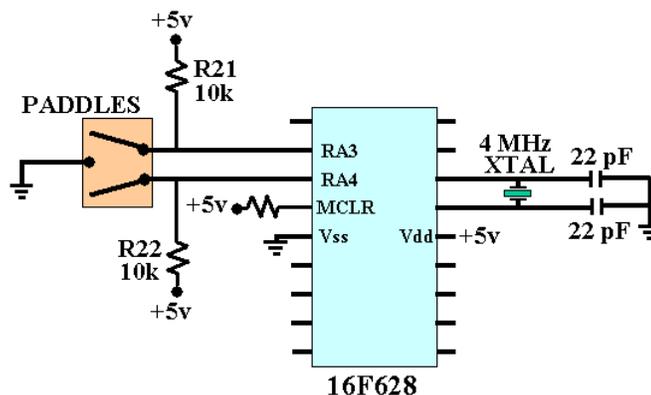


Figure 11– CW Paddles

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 +5v. 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.

2.2.10 Transmitter Keying via a Stereo Jack

Figure 12 shows how to key a transmitter with the output of the demonstration keyer. Another 1/8” stereo jack is provided for this purpose.

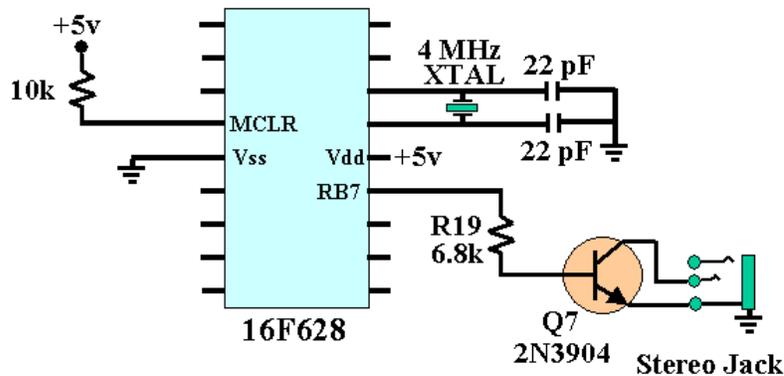


Figure 12 – Transmitter Keying

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 5v 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 5v. This keying mechanism will work for most modern rigs because they are positive-keyed transmitters. Some older style transmitters (tube style in particular) used negative keying. Positive keying means that the radio has approximately +3 to +5 volts on the tip connection and the radio is keyed when this pin is shorted to ground. Negative keying transmitters often have something on the order of -30v on the tip connection and are keyed when this connection is shorted to ground. This keying circuit is for positive keying only, but if your radio requires a negative keying scheme, it can be accomplished by adding a few extra components.

2.2.11 Frequency Counter

A frequency counter can be implemented in the PIC-EL by using the signal “conditioner” which was mentioned in a previous section. 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 the as a counter input to a PIC register, TMR0. The TMR0 register is used by frequency counter applications.

2.2.12 How to Drive a Relay from a PIC

Although the PIC-EL does not demonstrate this item, people often wonder how they can drive a relay or some other device which requires more current than a PIC pin can deliver (about 20 ma). Figure 14 shows how it can be done. The resistor (Rx) in series with the relay coil must be sized to pass the proper amount of current. The 2N7000 MOSFET is a good general-purpose part but in some applications, such as LCD backlight activation, an LRLML2502 is a good choice since its Drain-to-Source resistance when “turned on” is very low (about 0.045 ohms). This Rds(on) resistance is between 2 and 5 ohms in the 2N7000.

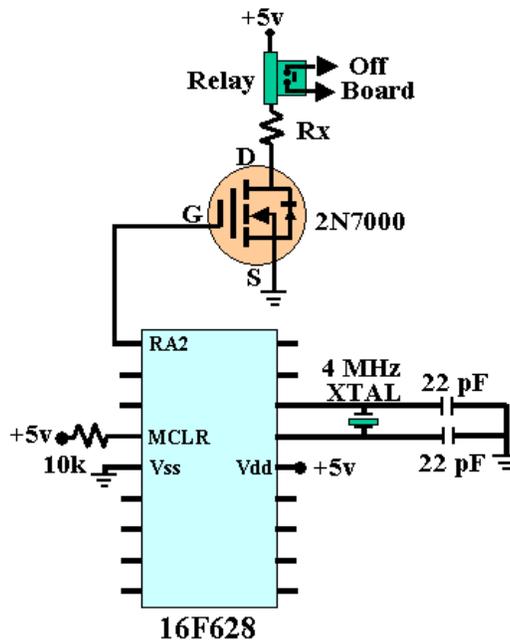


Figure 14 - Driving a Relay

3 PIC-EL Assembly

3.1 Parts Identification

The first thing you should do is to identify all the parts in the kit and make sure they are all there. The parts are listed and described in Appendix B. Every builder seems to have his own preferences for how to do this. You may want to separate like components (resistors, capacitors, etc) and put them in parts groups in trays, plastic cups, paper envelopes, etc. Some builders like to arrange the parts in order, sticking them in the edge of a corrugated cardboard box. Others like to tape them to a piece of paper next to their part number so they are easy to identify and find when needed. Reading the banded color codes on resistors may be a challenge to some and a simple ohm meter may be helpful in this regard. You really need to make sure you have them all identified properly. If you have questions, ask.

We make every effort to include all parts in the kit; however, mistakes do happen. If you find you are missing a part, notify Bill Kelsey at Kanga and he will send the missing parts to you. It may be faster to get common components from your own “junk box” or at your local Radio Shack.

3.2 Test Points

One rather unusual feature of the PIC-EL board is the existence of ten “test points” on the board. They are very useful for accessing key signals during initial test - supply voltages, encoder signals, I/O line, and more. They are little loops of wire and are very convenient for attaching clip leads for a volt meter or oscilloscope probe. You can use a small scrap of wire or a scrap from a trimmed component lead. I like to cut up small paper clip for this purpose since they are very strong and stiff and don’t bend when the lead clip or probe is attached. **Be VERY careful of your eyes** when using a wire cutter to cut up a paper clip! Hold one hand over the pieces so they don’t go flying into your eyes or across the room.

Since paper clip leads are very stiff, you may need two pairs of needle-nose pliers to form the small loops. It takes a bit of effort but the result is very nice.

The ten test points are:

1. **TP-A** = V_{pgm} (programming voltage), 12V is needed here for PIC programming Requires +V (supply voltage) of 12-14v. V_{pgm} will be less than 12V if supply voltage is less than 12V.
2. **TP-B** = +V (supply voltage), whatever your main supply voltage is plugged into the board. Should be 9-14v for PIC operation and 12-14v for PIC programming.
3. **TP-C** = V_{dd} (logic voltage), +5V for the PIC and associated logic
4. **TP-D** = GND, ground
5. **TP-F** = RF, signal going out (from DDS) or coming in (to be counted), depending on shunt position in HDR2
6. **TP-Ga** = ENC-A, encoder A signal, moves between 0 and 5v in quadrature with the B signal below when encoder is turned
7. **TP-Gb** = ENC-B, encoder B signal, moves between 0 and 5v in quadrature with the A signal above when encoder is turned
8. **TP-H** = TONES, can be high or low signal (depending on PIC program) but has audio freq square wave from 0-to-5V when speaker is sounding
9. **TP-Ia** = DIT or PB1, normally 5V, goes to zero when DIT paddle or pushbutton PB1 is pressed
10. **TP-Ib** = DAH or PB2, normally 5V, goes to zero when DAH paddle or pushbutton PB2 is pressed

3.3 Assembly

The silk-screen part names on the board will be your guide. Builders have different techniques and preferences for assembling kits. Some like to mount several parts and then solder them all at once. Some like to solder each individual part after it's mounted, to make sure you don't forget to solder something. Whatever your style, here is a suggested list of steps. Notice that there are tests interspersed along the way. This is designed to help you diagnose problems before you proceed to the next steps. It's also fun to see various parts of that the board coming to life.

1. **Install** the jumpers in the ten Test Points. Use wire scraps or piece of paper clip. (See above.)
2. **Install** 2 x 6 CONFIG header, HDR1. This header has double row of pins in plastic body. Put the shorter pins into the PC board.
3. **Install** 5 shunts across pins of the CONFIG header, HDR1. 1-to-2, 3-to-4, 5-to-6, 7-to-8, and 9-to-10. You don't need a shunt from 11 to 12 because these pins are already shorted on the board.
4. **Install** jack J1.
5. **Install** the voltage regulator, U2. Secure with 6-32 x 3/8" bolt and nut on bottom.
6. **Install** capacitors C5 - C8. Note that C5 and C8 are electrolytic capacitors so polarity must be observed. The bodies of these capacitors have the negative side marked "-". The negative side of the capacitor goes to the ground (at edge of board). C6 and C7 are the yellow, monolithic capacitors with "104" marked on the bodies.

7. **Test.** Apply 12-14V (supply voltage) to jack J1 and make sure you have +5V at TP-C.
8. **Test.** Make sure you have 12-14V (same as supply voltage) at TP-B
9. **Install** the two 22 pF capacitors, C1 and C2. These are capacitors with “220” on the bodies. Be careful to distinguish these from C11, the .22 uF monolithic capacitor with “224” markings on the body.
10. **Install** the four .1uF capacitors, C3, C4, C9, C10. These are the yellow, monolithic capacitors with “104” marked on the bodies.
11. **Install** the .22 uF capacitor, C11. Body marked with “224” on the body.
12. **Install** the 37 resistors, R1 – R37. Look at the color codes on the bodies of the resistors and match to the color codes indicated on the parts list. Be very careful with identification. Use an ohm meter to be sure.
13. **Install** the 4 MHz crystal. Keep it just a little (1/32 – 1/16”) above the board. Possibly use a piece of paper as a “washer” to hold it up while you solder it in place.
14. **Install** the Zener diode, D5. Be sure to mount with the banded end as indicated on the PCB.
15. **Install** the glass diodes, D1- D4, D6. Again, be sure to observe the position of the banded ends. Mount them as indicated on the PCB.
16. **Install** the DIP socket, J5
17. **Install** slide switch, S1
18. **Test.** Apply supply voltage to jack J1. Make sure you have +5V at pin 14 of DIP socket.
19. **Test.** Apply supply voltage to jack J1. Switch S1 into RUN position (UP – toward J1). Make sure you have +5V at pin 4 of DIP socket. Also you should see +12v on pin 4 of the DIP socket if S1 is in the PGM position (down) or +5v if S1 is in RUN position (up).
20. **Install** transistor Q4. This is the TO-18 (“metal can”). Make sure you have the tab in position indicated on the PCB.
21. **Install** the five 2N3904 transistors Q1 – Q3, Q5, Q7. Make sure they are mounted with the flat side of the transistor bodies as shown on the PCB.
22. **Install** the 2N4401 transistor, Q6. Again, make sure flat side of transistor is mounted as shown on PCB.
23. **Install** the four LEDs, LED1 – LED4. The cathode (negative) side of the LED is the lead with the shorter leg. It also has a slight flat spot on the side of the plastic body. Because there are “bumps” on the LED leads, you will have to clip them off just above the bumps before they will go in the holes in the PCB. After clipping, check again to make sure the flat side of the plastic body is as shown on the PCB.
24. **Test.** Apply supply voltage to jack J1. Put switch S1 into PGM (DN) position. Observe LED4 goes on. The LED will go off when switch S1 switched back to RUN (UP) position. The other LEDs will be off when no PIC is in the J5 socket.
25. **Install** pushbuttons, PB1 – PB4.
26. **Install** the 8-pin, 90-degree, SIP jack. Solder into position with socket centers about 3/8” above the board surface.

27. **Install** 16-pin SIP header, J4. These headers have a single row of 16 pins coming out of both sides of the plastic body. Solder the shorter pins into the PCB and leave the longer pins pointing up. Solder one pin and make sure it's straight. If necessary heat the soldered connection and straighten it. Then solder the remaining 15 pins.
28. **Install** 16-pin SIP socket, P4, on the back side of the LCD. This is the single row of 16 sockets with short pins coming out the other sides of the plastic bodies. Looking at face of LCD, make sure the sockets are mounted on the other (back) side of the LCD. Again, solder one pin and make sure it's straight. If necessary straighten it before soldering the remaining 15 pins.
29. **Install** 2x2 pin header, HDR2. This is double row of double pins. Install with shorter pins into the PCB.
30. **Install** 1x2 pin header, HDR3. This has two pins with a plastic body. Install with shorter pins into the PCB.
31. **Install** encoder ENC.
32. **Install** the speaker, SPKR. Observe polarity. See the "+" marked on the bottom of the speaker body and on the PCB. Be careful that you don't apply too much heat because it could damage the speaker.
33. **Install** RF jack J7
34. **Install** two stereo jacks, J3 and J8.
35. **Install** DB9F connector, J2

This completes the assembly of the PIC-EL. Congratulations!

3.4 Final installation and Basic PIC-EL Test

Before you plug your 16F628 into the socket (J5) please make these basic checks once again.

1. **Check the voltage at TP-B.** It should be equal to your supply voltage which between +9 and +14V. (See TP-B description above.)
2. **Check the voltage at TP-A.** It should be +12V, assuming your TP-B voltage is between +12 and +14V, as required for PIC programming.
3. **Check the voltage at TP-C.** It should be 5V.
4. **Check the voltage at DIP Pin 14.** It should be 5v.
5. **Check the voltage at pin 2 of LCD jack, J4.** It should be +5V.
6. **Check the voltage at pin 3 of LCD jack, J4.** It should be approximately +0.3V.
7. **Check the voltage at pin 8 of Daughtercard jack, J6.** It should be +12V.

Now turn off power to your PIC-EL. Then plug the PIC into the DIP socket, J5, and plug the LCD into jack J4. Make sure switch S1 is in RUN (UP) mode. Apply power to the board again and you should see the LCD come to life running the pre-loaded diagnostic program. (See Section 5 for information regarding the diagnostic program.) You should be able to run all the diagnostic tests. (You will need a DDS-30 and/or DDS-60 daughtercard plugged into jack J6 to run the DDS tests, of course.)

4 PIC-EL Computer Interface

Now you are ready to program a PIC using the PIC-EL. Before you can do this you need to have a program installed in your PC. There are many good options available (including IC-PROG and WinPIC) for the Windows Operating System but we will concentrate on one good one here, FPP by David Tait, since it is readily available and very easy to use. Similar packages are available for the Linux and Macintosh operating systems but will not be described here.

4.1.1 Installing FPP

David Tait's web page is no longer active on the Internet so I have moved the files to my web site with his permission. You can download FPP from the "DOWNLOADS" section of my web site: <http://www.cbjohns.com/aa0zz> .

If you are using Windows 95 or Windows 98 you can run FPP without any additional drivers. However, if you are using Windows 2000 or Windows XP you need to install two drivers, directio and loaddrv, to allow FPP to access the COM port. These two drivers are available many Internet sites but they are also available by following links from the AmQRP Elmer-160 web site or in the DOWNLOADS section of my web page.

Detailed instructions for installing FPP are available by downloading John's Elmer-160 Lesson 10 from the AmQRP Elmer-160 web site.

4.1.2 Installing FPPA

David Tait's FPP does not support some of the newer PICs and David is no longer making updates to the program. Thankfully, David published the source code for public use so I was able to modify it to support some of the newer PICs:

- 16F84A
- 16F627A
- 16F628A
- 16F648A
- 16F877A

In particular, I wanted to support the 16F628A because it's more readily available than the 16F628 and I wanted to use the 16F628A in the PIC-EL.

I called the new version FPPA. You can download it from the "DOWNLOADS" section of my web site: <http://www.cbjohns.com/aa0zz> .

The installation and operation is exactly the same as FPP. The new PICs are selected in the device selection drop-down list.

4.1.3 Using FPP / FPPA

A serial cable is needed to connect the PC to the PIC-EL board. You need a standard straight-through 9-pin DB9M (male) to 9-pin DB9F (female) connector. One end of the cable plugs into the RS-232C serial (COM) port on your PC and the other end plugs into the PIC-EL board. You may need to "disable" other programs also using that same serial port.

Be sure that you have a sufficiently high voltage power supply connected to your PIC-EL board. Although the board can run off a 9V battery, you will need to have at least 12V at the PIC-EL CONFIG header (HDR1) in order to generate the minimum "programming voltage", called Vpgm.

1) **Set up**

After the serial cable is connected, you've got FPP loaded and turned on, your PIC-EL board is powered by at least 12V, and you've manually wiggled the lines as described in Lesson 10, it should be a piece of cake to program a PIC on the PIC-EL board.

2) **Obtain the .HEX program**

The HEX file is the new software you will be burning into the PIC. You can download the PIC-EL V2 Diagnostic program from my web page or the FILES section of the PIC-EL YAHOO group. The PICELv2diag.HEX file contains the Test Program in "HEX ASCII" format. This file format is the "standard output" produced by PIC assembly programs and is a specific data format which is expected by the FPP program.

3) **Load the PICELv2diag.HEX file into FPP / FPPA.**

Click on the **LOAD** button and navigate to wherever you unzipped the PIC-EL V2 Diagnostic files on your PC. You will see the PICELv2diag.HEX file listed there. Double-click on it and the HEX ASCII code will load into the FPP buffer. You will see that code in the FPP window.

4) **Slide mode switch S1 DOWN to PGM MODE.**

You need to move the slide switch S1 to the **DOWN** position in order to put the PIC-EL board into the **PGM Mode**. LED4, next to the switch, will turn on when you do this.

5) **Erase the PIC currently plugged into the PIC-EL board.**

You **MAY** want to "erase", or clear out the software program currently in the PIC's flash memory before you burn a new program into the PIC. Click the **ERASE** button to do this, and if successful you will see a simple message pop up saying "PIC is erased".

6) **Burn the new code into the PIC.**

Now that the PIC memory is empty and you have the new program (HEX file) in the FPP buffer window, you are all set to burn the program into the PIC. Click **PROGRAM** on the FPP application window and confirm your desire again in the pop-up window. It will take a few moments for this short program to be burned into the PIC, but when complete FPP will display "Device programmed!". If it says "Programming failure", you obviously have a problem like the PIC was not first erased, power supply wasn't connected or sufficiently high, cable was not plugged in, etc.

7) **Slide the mode switch UP to go into RUN MODE.**

Now that the programming is complete, you next need to put the PIC-EL board back into **RUN Mode**. Do this by sliding Mode Switch S1 **UP**, and the PGM LED will turn off once again.

8) **RESET the board to start up the new program.**

Although not always necessary, just press the **RESET** pushbutton on the PIC-EL board to start up the new program just programmed into the PIC.

5 PIC-EL Diagnostic Program

A PIC-EL diagnostic program is pre-loaded on the 16F628 PIC that comes with the PIC-EL kit. The test symbolics file is also available on my web site and on the PIC-EL YAHOO group FILES section. By looking at these symbolics and running the various tests you will be able to get an idea about how to activate the various elements of the PIC-EL. Note that the diagnostic program was written for demonstration purposes and not necessarily the way a finished program would be written.

When the test is started, the various test items will appear, one at a time, on line 2 of the LCD. When the test you want to run is displayed, press pushbutton PB3. When you want to end a particular test you again press PB3 and hold it for ½ second. The main menu of tests will again start to appear, one at a time.

5.1 Test LEDs

When this test is selected the diagnostic will light the three LEDs in this sequence.

1. Turn LED1 ON and then OFF
2. Turn LED2 ON and then OFF
3. Turn LED3 ON and then OFF
4. Turn LED1 and LED2 and LED3 ON and then turn them all OFF.

The PGM/RUN LED is not exercised in this test. It is turned ON when switch S1 is in Program mode and OFF when in Run mode.

5.2 Test Pushbuttons

This test will verify the operation of the three general-purpose pushbuttons. Pressing the pushbuttons in sequence will result in the following results:

1. Pressing PB1 will cause LED1 to light. It will remain on until PB1 is released.
2. Pressing PB2 will cause LED2 to light. It will remain on until PB2 is released.
3. Pressing PB3 will cause LED3 to light briefly but the test will then end.

This test does not exercise the Reset pushbutton, PB4. Pressing PB4 resets the PIC and makes the test start over.

5.3 Test Speaker

Selecting this test will cause a series of tones to come from the PIC-EL speaker. The test plays 8 tones of a one-octave diatonic scale starting with C and ending at the next higher C. By the way, this test uses classical music theory, starting with “middle C”, ending with the next higher C, and tuned such that the A above “middle C” is tuned to 440 Hz. The accuracy is limited by the tolerance of the 4 MHz crystal. By my ear it’s pretty close.

5.4 Test Encoder

When this test is selected, the mechanical encoder is exercised. The test will start at zero with an 8-digit number displayed on the LCD. Then, turning the encoder clockwise will increment the displayed number and turning the encoder counter-clockwise will decrement the displayed number. The test uses the “gray code” signals generated by the encoder (see Figure 8) and determines the direction by the mechanism described.

5.5 Test Paddles

CW paddles can be connected to the PIC-EL via a patch cord which plugs into jack J3. The jack connects one of the paddle connections to the “tip” and the other to the “ring”. (In the absence of CW paddles, you can exercise the equivalent functionality by pressing PIC-EL pushbuttons PB1 and PB2.) When the “Dit” paddle is pressed (or PB1) the test will light LED1. When the “Dah” paddle is pressed (or PB2) the test will light LED2.

5.6 Test Transmitter Keying

This test will demonstrate how CW keying is accomplished via a PIC microcontroller. An oscillator or transmitter is connected to the PIC-EL board via a patch cord with 1/8” mono connectors. One end is plugged into PIC-EL jack J8 and the other is plugged into the oscillator/transmitter. The test activates transistor Q7 which effectively “shorts” the key line (“tip” connection) to ground and activates the oscillator/transmitter.

5.7 Test DDS-30

If you have a DDS-30 DDS Daughtercard from AmQRP organization you can test it by selecting this test. The DDS-30 is installed in Jack J6 of the PIC-EL. The test program generates three frequencies – 7.040 MHz, 7.041 MHz and 7.042 MHz. (They are displayed on the LCD in Hz as 7040000, 7041000 and 7042000.) The duration of each frequency is about ½ second and then the sequence repeats. You can observe the output on a spectrum analyzer (or oscilloscope) or by tuning your HF receiver to 7.040 MHz.

5.8 Test DDS-60

The DDS-60 DDS Daughtercard from AmQRP is slightly different from the DDS-30 in that the DDS-60 uses an AD9851 DDS while the DDS-30 uses an AD9850. The DDS-60 uses a 30 MHz clock oscillator and expects the 6x multiplier to be activated by driver software. Once again, the DDS-60 is installed in PIC-EL jack J6. When selected, this test generates the same three frequencies, 7.040 MHz, 7.041 MHz, and 7.042 MHz. You can observe the output on a spectrum analyzer (or oscilloscope) or by tuning your HF receiver to 7.040 MHz.

6 Elmer-160 Lessons

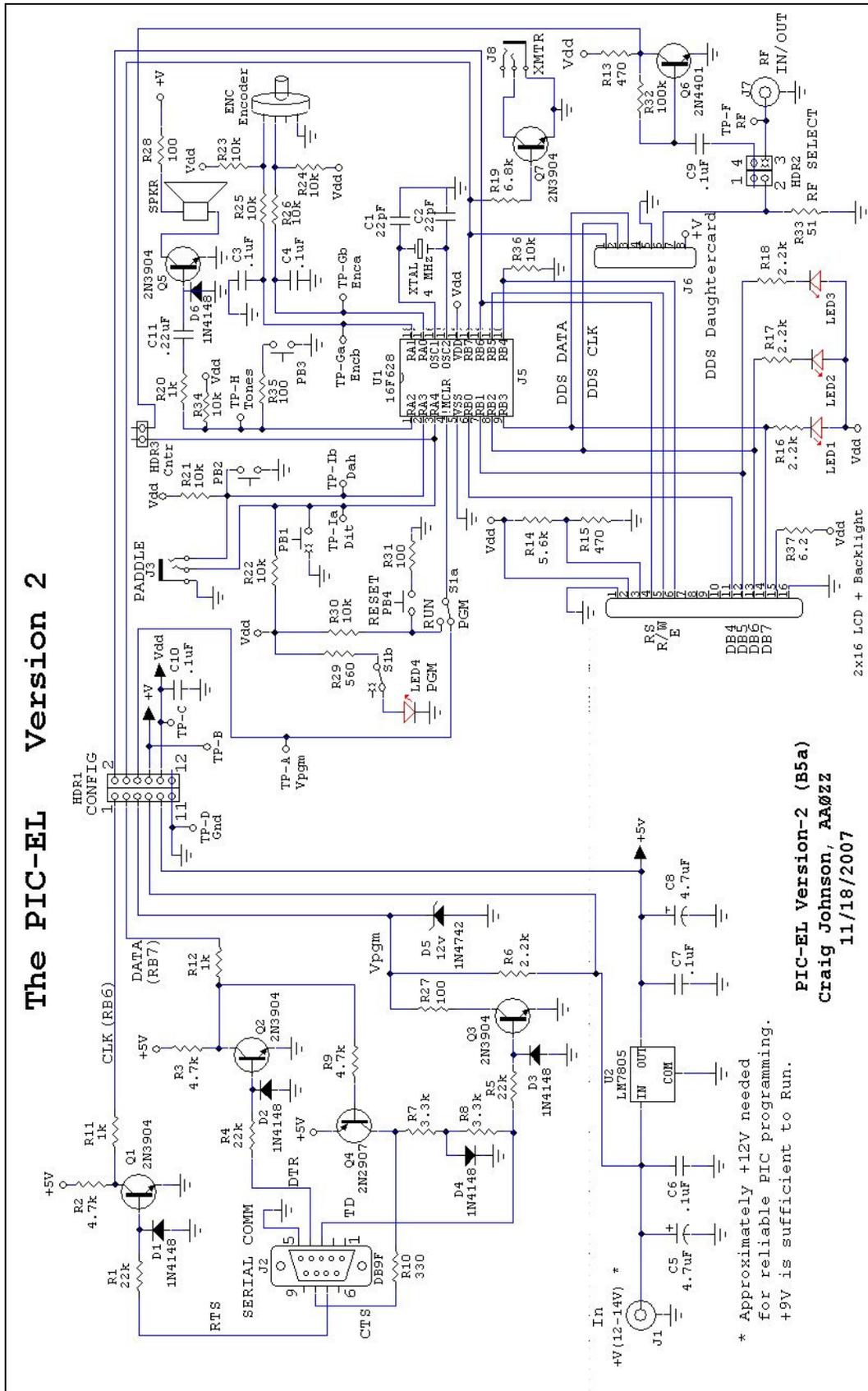
If you are interested in John McDonough’s (WB8RCR) Elmer-160 lessons, you can find them at the AmQRP web site: <http://www.amqrp.org/elmer160>. John’s lessons are available for downloading and you can contact him by using links provided on that web site.

7 Support

For up-to-date details and documentation regarding this project, please see my web page, www.cbjohns.com/aa0zz.

For additional support questions, see the YAHOO group PIC-EL or Email me directly.

8 Appendix A – Schematic



9 Appendix B – Parts List

PIC-EL (Version 2) Parts						11/29/2007
Quantity	Designator	Description	(D) DigiKey (M) Mouser or Other	Small Quantity Price	Total at Small Quantity Price	
Resistors (1/8 or 1/4 watt, 5%)						
1	R37	6.2 (blue, red, gold)	(D) 6.2EBK-ND (M) 291-6.2-RC	0.10	0.10	
1	R33	51 ohms (green, brown, black)	(D) 51EBK-ND (M) 291-51-RC	0.10	0.10	
4	R27, R28, R31, R35	100 ohms (brown, black, brown)	(D) 100EBK-ND (M) 291-100-RC	0.10	0.40	
1	R10	330 (orange, orange, brown)	(D) 330EBK-ND (M) 291-330-RC	0.10	0.10	
2	R13, R15	470 ohms (yellow, violet, brown)	(D) 470EBK-ND (M) 291-470-RC	0.10	0.10	
1	R29	560 ohms (green, blue, brown)	(D) 560EBK-ND (M) 291-560-RC	0.10	0.10	
3	R11, R12, R20	1K (brown, black, red)	(D) 1.0KEBK-ND (M) 291-1K-RC	0.10	0.30	
4	R6, R16, R17, R18	2.2K (red, red, red)	(D) 2.2KEBK-ND (M) 291-2.2K-RC	0.10	0.40	
2	R7, R8	3.3K (orange, orange, red)	(D) 3.3KEBK-ND (M) 291-3.3K-RC	0.10	0.20	
3	R2, R3, R9	4.7K (yellow,violet, red)	(D) 4.7KEBK-ND (M) 291-4.7K-RC	0.10	0.30	
1	R14	5.6K (green, blue, red)	(D) 5.6KEBK-ND (M) 291-5.6K-RC	0.10	0.10	
1	R19	6.8K (blue,gray, red)	(D) 6.8KEBK-ND (M) 291-6.8K-RC	0.10	0.10	
9	R21, R22, R23, R24, R25, R26, R30, R34, R36	10K (brown, black, orange)	(D) 10KEBK-ND (M) 291-10K-RC	0.10	0.90	
3	R1, R4, R5	22K (red, red, orange)	(D) 22KEBK-ND (M) 291-22K-RC	0.10	0.30	
1	R32	100K ohms (brown, black, yellow)	(D) 100KEBK-ND (M) 291-100K-RC	0.10	0.10	
Capacitors						
2	C1, C2	22 pF, ceramic disc ("22")	(D) 495-1004-1-ND (M) 140-50N2-220J-RC	0.06	0.12	
2	C5, C8	4.7 uF, electrolytic (blue)	(D) BC1148CT-ND (M) 140-XRL16V4.7-RC	0.23	0.46	
6	C3, C4, C6, C7, C9, C10	0.1 uF, monolithic, (small yellow, "104")	(D) P4910-ND (M) 80-C317C104M5U	0.07	0.42	
1	C11	.22 uF, monolithic	(D) 445-2849-ND	0.22	0.22	
Diodes						
5	D1, D2, D3, D4, D6	1N4148	(D) 1N4148FS-ND (M) 625-1N4148	0.07	0.35	
1	D5	1N4742, 12V zener DO-41	(D) 1N4742ADICT-ND (M) 78-1N4742A	0.07	0.07	
Transistors						

5	Q1, Q2, Q3, Q5, Q7	2N3904 transistor, NPN TO-92	(D) 2N3904D26ZCT-ND	0.19	0.75
1	Q4	2N2907A transistor, PNP TO-18	(D) 497-2577-5-ND (M) 610-2N2907A	0.94	0.94
1	Q6	2N4401 transistor, NPN TO-92	(D) 2N4401-ND (M) 610-2N4401	0.12	0.12
Other					
1	U2	L7805 voltage regulator, 5V	(D) 497-1442-5-ND (M) 511-L7805 ABV	0.81	0.81
1	LCD	Liquid Crystal Display, 2x16 char	08LCD9 from www.ElectronixExpress.com	8.95	8.95
4	LED1, LED2, LED3, LED4	LED, T1-3/4 (red)	(D) P374-ND	0.17	0.67
1	XTAL	Crystal, 4 MHz	(D) X405-ND (M) 520-HCU400-20	0.58	0.58
1	J1	coaxial power jack, 2.1mm	(D) CP-102AH-ND	0.42	0.42
1	J2	DB9F serial connector	(M) 152-3409	1.09	1.09
2	J3, J8	audio jack, 1/8", stereo	(D) CP1-3513N-ND (M) 161-3507-E	1.26	2.52
1	P4	SIP socket, 16-pos'n (for LCD)	(D) ED7150-ND (M) 517-974-01-16	1.54	1.54
1	J4	SIP header, 16-pos'n (for PCB)	(M) 571-16404526	1.25	1.25
1	J5	DIP socket, 18-position (for PIC)	(D) ED3118-ND	1.01	1.01
1	J6	SIP socket, 8-pos'n, 90-deg	SamTec SSQ-108-04-T-S-RA	0.80	0.80
1	J7	BNC jack, pcb mount	(M) 523-31-5538-10-RFX	2.18	2.18
1	HDR-1	pin header, 0.1", 2x6 pos'n	(M) 571-1032406	0.90	0.90
8	Shunts for HDR1, shunt-1	shunt, 0.1", 2 pos'n	(M) 571-2-382811-1	0.12	0.96
1	HDR-2	pin header, 0.1", 2x2 pos'n	(M) 571-1032402	0.32	0.32
1	HDR-3	pin header, 0.1", 1x2 pos'n	(M) 571-1032392	0.20	0.20
4	PB1, PB2, PB3, PB4	SPST pushbutton, momentary	(D) P8079SCT-ND	0.38	1.52
1	S1	slide switch, DPDT	(D) SW102-ND	1.29	1.29
1	ENC	rotary encoder	(D) P10860-ND or (D) P12334-ND or (D) P12335-ND	1.52	1.52
1	SPKR	speaker	(D) 433-1028-ND	0.87	0.87
1	machine screw (6-32 x 3/8")	(for U2 regulator)	(D) H356-ND	0.02	0.02
1	machine nut (6-32)	(for U2 regulator)	(D) H220-ND	0.01	0.01
1	U1	PIC16F628 microcontroller with diagnostics pre-loaded	(D) D19PIC16F628A-I/P - ND with diagnostics	7.50	7.50
1	PCB	PC Board	(Kanga)	20.00	20.00
				=====	
				Single cost	\$63.98

10 Appendix C – Parts Layout

