# The PIC-EL III

A PIC Programmer and Test/Demonstration Board

**By Craig Johnson, AAØZZ**

aa0zz@cbjohn.com

www.cbjohn.com/aa0zz

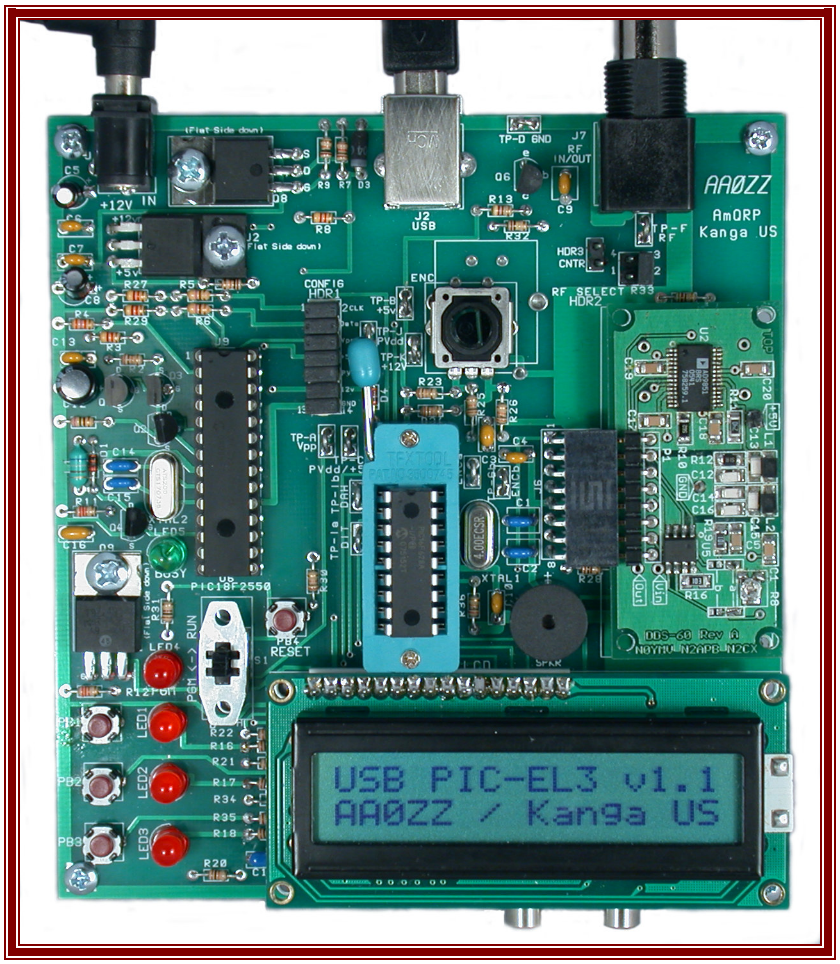



**TABLE OF CONTENTS**

1 Introduction ..... 2
2 Hardware Description ..... 3
2.1 PIC Programmer ..... 3
2.2 Project / Demonstration Portion ..... 4
2.2.1 PIC system clock (crystal) ..... 5
2.2.2 LEDs ..... 6
2.2.3 Pushbuttons ..... 7
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 ..... 11

2.2.9 CW paddle Input via a Stereo Jack ..... 12
2.2.10 Transmitter Keying via a Stereo Jack ..... 12
2.2.11 Frequency Counter ..... 13
2.2.12 How to Drive a Relay from a PIC ..... 13
3 PIC-EL Assembly ..... 14
3.1 Parts Identification ..... 14
3.2 Test Points ..... 14
3.3 Assembly ..... 15
3.4 Final Installation and Basic PIC-EL Test ..... 18
4 PIC-EL III Computer Interface ..... 18
5 PIC-EL III Programming Software ..... 18
5.1.1 Installing PICKit2 Programming Application (PICKit2 App) ..... 18
5.1.2 Programming a PIC with PICKit2 App ..... 19
6 PIC-EL Diagnostic Program ..... 21
6.1 Test LEDs ..... 22
6.2 Test Pushbuttons ..... 22
6.3 Test Speaker ..... 22
6.4 Test Encoder ..... 22
6.5 Test Paddles ..... 22
6.6 Test Transmitter Keying ..... 23
6.7 Test DDS-30 ..... 23
6.8 Test DDS-60 ..... 23
7 Elmer-160 Lessons ..... 23
8 Support ..... 23
Appendix A – Schematic ..... 24
Appendix B – Parts List ..... 25
Appendix C – PIC-EL III Parts Placement ..... 28
Appendix D – PIC-EL III PC Board ..... 29

# 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, receiver and/or transmitter controllers, repeater controllers, power/SWR meters, antenna control, and station control. The list is endless.

I developed the original 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 III is a 16F628A, 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 in mid-range group, the 16F84, because it is even easier to for beginners to understand. The 16F628A 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 16F628A 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. Then I updated the board to replace obsolete parts, correct a few deficiencies, and make the kits available on a long-term basis. The result was the PIC-EL Version 2.

The fact remained that many people were still looking for a PIC-EL that used a USB interface instead of the COM interface. I investigated many different methods but each had severe drawbacks, including the need for me to write new programming software to support it. Then, in late 2007, I bought a Microchip PICKit2, thinking it may be a way to let people have a way to quickly attach to the PIC-EL with a USB connection. The PICKit2 would be connected such that it would bypass the PIC-EL's hardware programmer and connect directly into the PIC-EL's configuration header. My initial attempts failed but several months later, by installing PICKit2 hardware updates suggested by Microchip, it worked perfectly on the PIC-EL II. I then found a number of "clones" on the web and, upon examining them, I decided that there is a lot of hardware in the Microchip PICKit2 that is nice but not really needed for a PIC-EL environment. I made a stripped-down version of the PICKit2 using ideas from other designs on the web but then putting my own spin on it. The PIC-EL III was born.

# 2 Hardware Description

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

## *2.1 PIC Programmer*

The PIC-EL programmer is a simplified version of the Microchip PICKit2 hardware and it uses the Microchip PICKit2 Programmer Application software (called PICKit2 App from here on) to drive it.

**NOTE: YOU MUST USE THE PICKIT2 APP (or MPLAB or similar) SOFTWARE WITH THE PIC-EL III. YOU CANNOT USE FPP OR WINPIC OR OTHER COM PORT PROGRAMMING SOFTWARE. This manual will assume you are using PICKIT2 APP.**

**PICKit2 App only runs on PCs with Microsoft Windows 2000 (Service Pack 3 or beyond) , XP or Vista operating system.**

The application software establishes a connection with the PIC-EL's programmer and then performs the read/write functions to the PIC in the PIC-EL III. The heart of the PIC-EL III programmer is a 18F2550 PIC which runs "operating system" code published by Microchip that connects to the PICKit2 App programming software. The 18F2550 PIC that is supplied with the PIC-EL III has this "operating system" code pre-loaded. If you want to start over with a blank 18F2550 PIC you will need to somehow load a base-level version of the code (file PK2V023200.hex on the CD) into the 18F2550 PIC with another programmer. Then, after inserting this 18F2550 PIC into the 28-pin socket in the PIC-EL III, the PIC-EL III will be fully functional.

The PIC-EL III hardware programmer uses MOSFETs to drive the programming lines. It does not draw 5v power from the USB connection but instead runs on 5v power from the PIC-EL's 12v-to-5v converter. The programmer hardware has charge pump circuitry to internally generate the +12v programming voltage (Vpp) to be applied to the MCLR pin. Note that the PIC-EL expects +12v (nominal) power supply power to be supplied at all times and it does not use the 5v power supplied by the USB. The PIC-EL's 12v-to-5v voltage converter supplies all the 5v power for the PIC-EL board. Note that when the PIC-EL is in "PROGRAM" mode, the programmer hardware generates Vdd (+5v) and Vpp (+12v) with the appropriate timing to program the target PIC. Depending on the type of PIC being programmed, the programmer sometimes raises Vpp before Vdd and sometimes raises Vdd before Vpp. It's tricky but very important. (This removes a limitation found in the previous PIC-EL versions as well as all other Tait-type programmers that use a COM interface. There simply aren't enough pins in a COM port to do this.)

The +12v programming voltage (Vpp) can be generated ("pumped up" from the input power supply voltage by the charge pump circuitry) to permit PIC programming to take place when the input power supply voltage is as low as approximately 7.5v.

The PIC-EL III has supports these 18-pin PICs:

- 16F627
- 16F627A
- 16F628
- 16F628A
- 16F648A
- 16F716
- 16F818
- 16F819
- 16F84A
- 16F87
- 16F88
- 18F1220
- 18F1230
- 18F1320
- 18F1330

Additional PICs can be supported by connecting adapters to the PIC-EL's 18-pin DIP socket or by connecting wires and adapters to the CONFIG Header.

## 2.2 Project / Demonstration Portion

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 HDR2
    - Allow DDS output to be routed to the BNC (by installing a jumper between HDR2 pins 2 and 3)
    - Allow DDS output to be routed to a "conditioner" and then to a PIC input pin (by installing a jumper between HDR2 pins 1 and 2)
    - Allow an outside signal source to be brought in to the "conditioner" and then to the PIC input pin (by installing a jumper between HDR2 pins 3 and 4)
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 III 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 III. PIC pins were selected here for ease of illustration. Refer to the schematic in Appendix A for the PIC-EL III 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.

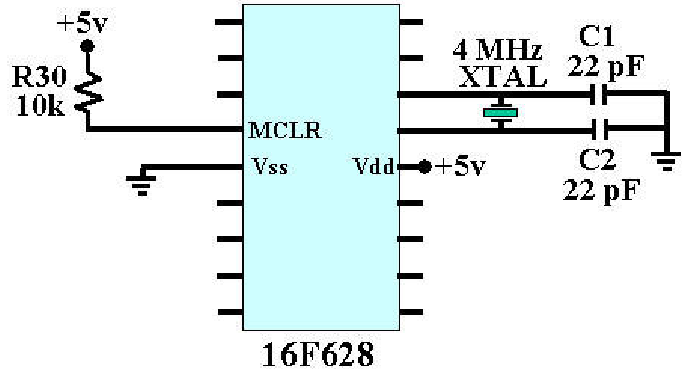



**Figure 1 – Basic PIC connections**

### 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 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.

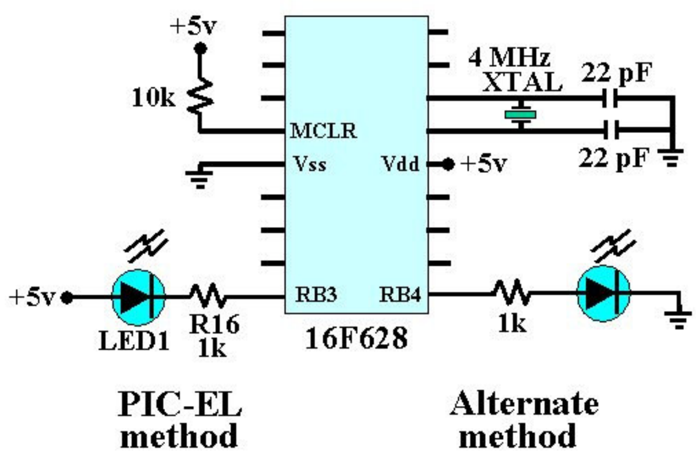



**Figure 2 – Lighting an LED**

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 1k ohms. This value was 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 1k resistors is about 3.2v. This means the current through the resistors and these LEDs is about 3.2 ma. This amount of current illuminates the LEDs sufficiently.

### 2.2.3 Pushbuttons

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

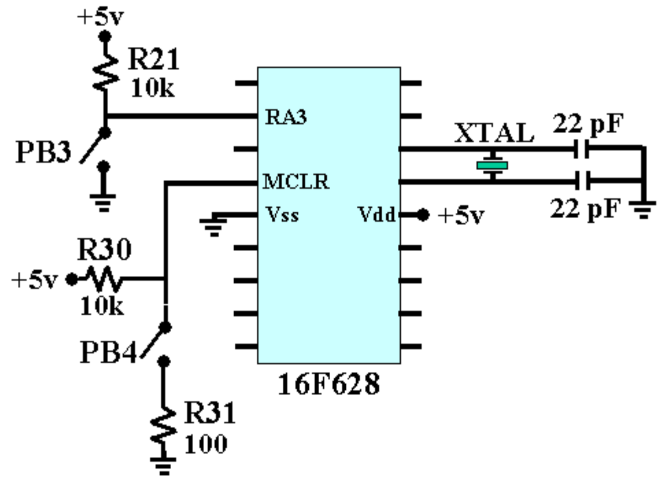



**Figure 3 – General Usage and Master Clear Pushbuttons**

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.

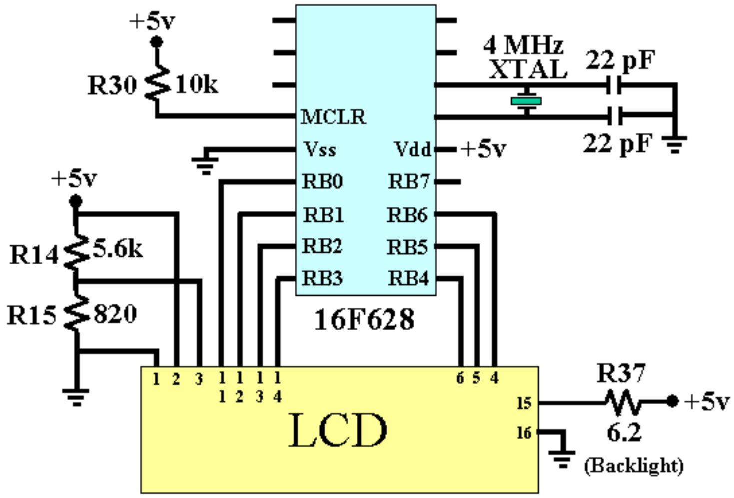



**Figure 4 – LCD**

### 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.

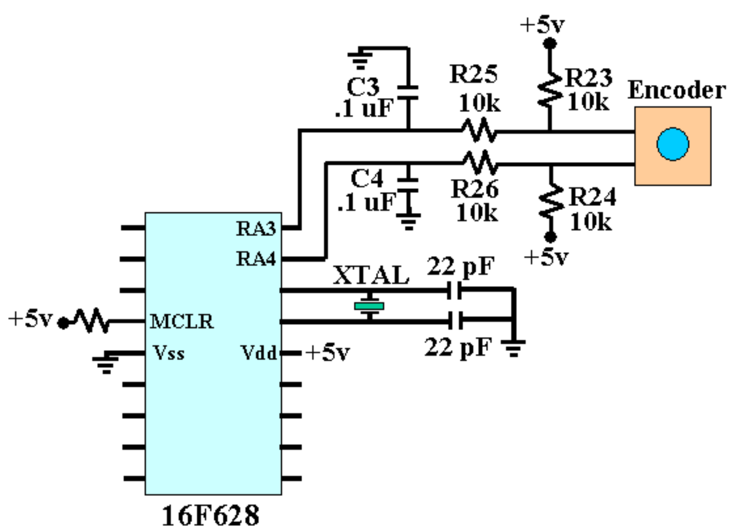
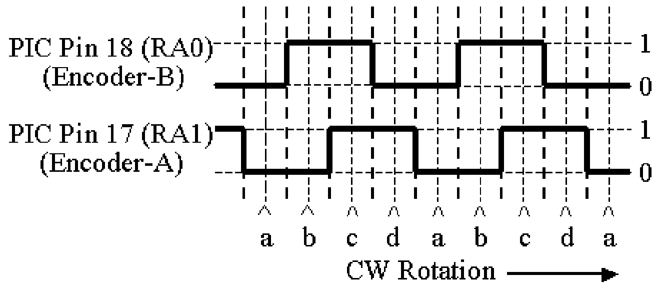



**Figure 5 – Rotary Encoder**

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 and Figure 7 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 7illustrates 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.

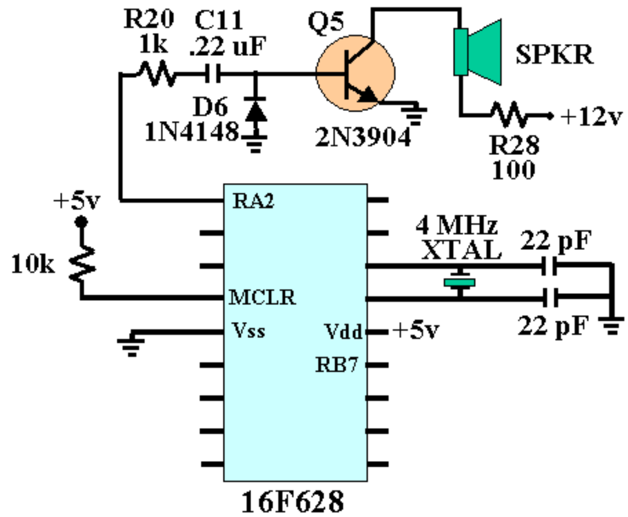



**Figure 8- Speaker**

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

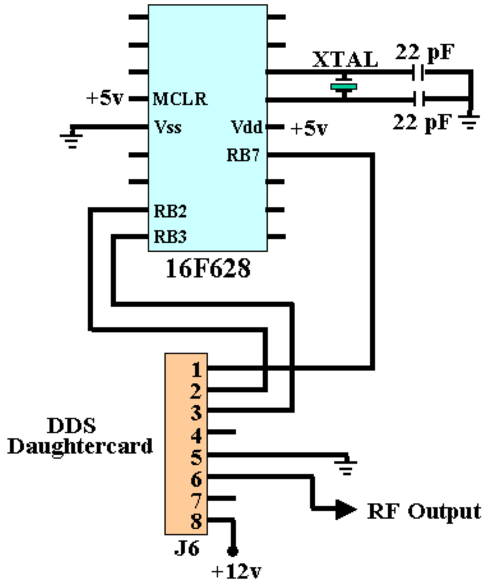



**Figure 9- DDS Daughtercard**

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.

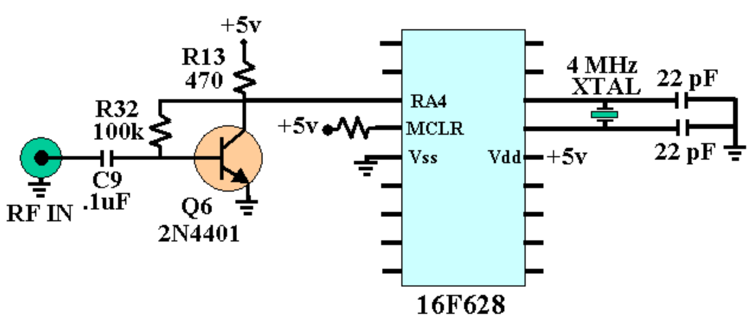



**Figure 10– Signal Conditioner**

Note that header HDR2 is used to select the source of the signal which goes into the “conditioner”. In one position (jumper from HDR2 pins 1 to 2), the output of the DDS Daughtercard is fed into the “conditioner” while in another configuration (jumper from HDR2 pins 3 to 4) 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. (Another option is to route the DDS output directly to the BNC, bypassing the "conditioner", by installing a jumper between HDR2 pins 2 and 3.)

The sensitivity of the signal conditioner on the PIC-EL III is about 100 mv P-P at 7 MHz.

### 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.

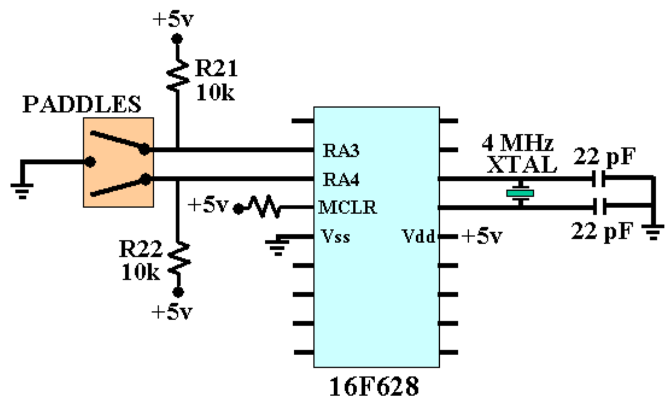



**Figure 11– CW Paddles**

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.

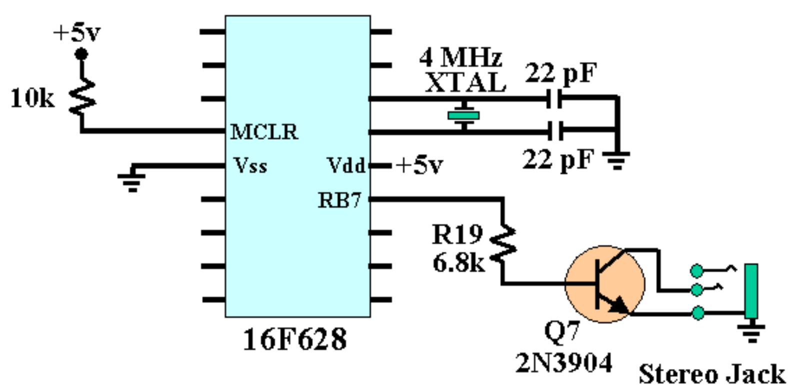



**Figure 12 – Transmitter Keying**

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 a 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.

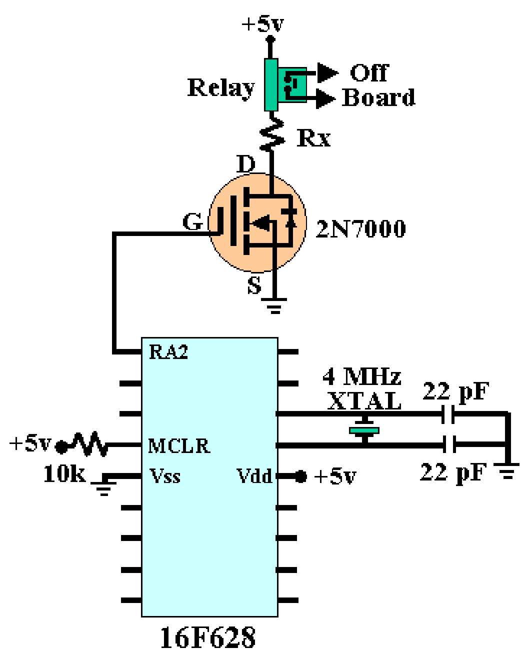



**Figure 14 - Driving a Relay**

# 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 eleven "test points" on the board. They are very useful for accessing key signals during initial test – to look at 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 metal paper clip for this purpose since they are very strong and stiff and don't deform 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 eleven test points are:

1. **TP-A** = Vpp (programming voltage) - +12V is needed here for PIC programming. Vpp is generated by the "charge-pump" mechanism in the programmer.
2. **TP-B** = +5V. Goes through MOSFET Q9 (when in RUN mode) to supply +5v for the LCD and all other components on the board EXCEPT the PIC. PIC power is supplied by PVdd.
3. **TP-C** = PVdd (Programming Vdd) - Supplies +5v to the PIC power (Vdd) pin at the appropriate time for proper PIC programming. (This timing is critical for some PICs.) Will see approximately 0v here when not programming! Same as TP-J except lowered by the D4 diode voltage drop (approximately 200 mv).
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 frequency 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
11. **TP-J =** PVdd (Programming Vdd) - Supplies +5v to the PIC power (Vdd) pin at the appropriate time for proper PIC programming. (This timing is critical for some PICs.) Will see approximately 0v here when not programming! Same as TP-C except higher by the D4 diode voltage drop (approximately 200 mv).
12. **TP-K =** +12v for the speaker.

## *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 four corner standoff insulators on the bottom side of each corner. Use 4-40 x ½" bolts and 4-40 nuts.
2. **Install** the jumpers in the twelve Test Points. Use wire scraps or piece of paper clip. (See above.)
3. **Install** 2 x 7 CONFIG header, HDR1. This header has double row of pins in plastic body. Put the shorter pins into the PC board.
4. **Install** 6 shunts across pins of the CONFIG header, HDR1. 1-to-2, 3-to-4, 5-to-6, 7-to-8, 9-to-10, and 11-to-12. You don't need a shunt from 13 to 14 because these pins are already shorted to the board.
5. **Install** jack J1.
6. **Install** the voltage regulator, U2 (L7805 regulator). Secure with 6-32 x 3/8" bolt and nut.
7. **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.
8. **Test**. Apply 12-14V (supply voltage) to jack J1 and make sure you have +5V at TP-B.
9. **Test**. Temporarily remove the jumper between pins 11 and 12 on HDR1. Make sure you have 12-14V (same as supply voltage) at pin 11. **Be careful not to allow your probe to make a short between Pin 11 and Pin 13 (ground)!** Then replace the jumper.
10. **Install** 28-pin DIP socket J9. Install with notched end as shown on silk screen to ensure the PIC will be installed properly.
11. **Install** the four 22pF capacitors, C1, C2, C14, C15. 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.

12. **Install** the five .1uF capacitors, C3, C4, C9, C10, C13. These are the yellow, monolithic capacitors with "104" marked on the bodies.
13. **Install** the .22 uF capacitor, C11. Body marked with "224" on the body.
14. **Install** capacitor C16 (**.**47uF). These are the yellow, monolithic capacitors with "474" marked on the bodies.
15. **Install** electrolytic capacitor C12. Observe polarity.
16. **Install** the 38 resistors, R1 – R38. 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.
17. **Install** the 4 MHz crystal, XTAL1. Make sure you have the right 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.
18. **Install** the 20 MHz crystal, XTAL2. Make sure you have the right 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.
19. **Install** Q8 – IRF9Z24 MOSFET. Mount with flat side down on board. Secure with 6-32 x 3/8" bolt and nut.
20. **Install** Q9 - IRF9Z24 MOSFET. Mount with flat side down on board. Secure with 6-32 x 3/8" bolt and nut.
21. **Install** the four diodes D1 (BAT85), D3 (1N4004), D4 (BAT85) and D6 (1N4148). Be sure to mount with the banded end as indicated on the PCB. (D2 and D5 do not exist.)
22. **Install** the inductor, L1 (680u).
23. **Install** the MOSFET Q3 (BS250). The body of this one is a little different than the other MOSFETS and the NPN transistors. Be very careful of the orientation with the "longest flat side" of the MOSFET lining up with the flat side marking on the PCB.
24. **Install** the MOSFETs, Q1 (BS170), Q2 (BS170) and Q4 (BS170). Be sure to observe markings. Make sure they are mounted with the flat side of the transistor bodies as shown on the PCB.
25. **Install** the 2N4401 transistor, Q6. Again, make sure the flat side of transistor is mounted as shown on PCB.
26. **Install** the 2N3904 transistors, Q5 and Q7. Make sure they are mounted with the flat side of the transistor bodies as shown on the PCB. Mount Q5 and Q7 as close to the PCB as possible, since the LCD will be mounted over it.
27. **Install** the DIP socket, J5 (machined pin or ZIF). Install with notched end as shown on silk screen, to ensure the PIC will be installed properly.
28. **Install** slide switch, S1.
29. **Test.** Apply supply voltage to jack J1. Make sure you have +5V at pin 14 of DIP socket J5 (for U1).
30. **Install** the four RED 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. Line up the flat spot (cathode) with the flat side marking on the PCB. **Note that the orientation of LED4 is opposite that of LED1 - LED3.**

31. **Install** the GREEN LED, LED5. 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. Line up the flat spot (cathode) with the flat side marking on the PCB.
32. **Install** the pre-programmed 18F2550 PIC, U6, in the 28-pin DIP socket, J8.
33. **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 J5.
34. **Test.** With supply voltage applied to jack J1, put switch S1 into PGM (DN) position. Observe LED4 goes on. LED4 will go off when switch S1 switched back to RUN (UP) position. The other LEDs (LED1 – LED3) will be off when no PIC is in the J5 socket.
35. **Install** pushbuttons, PB1 – PB4.
36. **Install** the 8-pin, 90-degree, SIP jack, J6. Solder into position with socket centers about 3/8" above the board surface.
37. **Install** 16-pin SIP socket, P4. This is the single row of 16 sockets with short pins coming out the other sides of the plastic bodies. Solder one pin and make sure the row is straight. If necessary, straighten it before soldering the remaining 15 pins.
38. **Install** 16-pin SIP header, J4, on the back (bottom) side of the LCD. These headers have a single row of 16 pins coming out of both sides of the plastic body. Solder the shorter pins into the back side of the LCD and leave the longer pins pointing outward. Solder one pin and make sure it's straight. If necessary heat the soldered connection and straighten it. Then solder the remaining 15 pins. If you want to support the LCD with bolts and nuts in the corners, you can do so. These bolts and nuts are not supplied with the kit.
39. **Install** 2x2 pin header, HDR2. This is double row of double pins. Install with shorter pins into the PCB.
40. **Install** 1x2 pin header, HDR3. This has two pins with a plastic body. Install with shorter pins into the PCB.
41. **Install** encoder, ENC. There are holes in the board for two different sizes of encoder. Use the holes that are appropriate for your encoder.
42. **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.
43. **Install** RF jack J7.
44. **Install** two stereo jacks, J3 and J8.
45. **Install** the USB receptacle, J2.
46. **Test.** With supply voltage applied to jack J1, put switch S1 into RUN (UP) position. Measure approximately 5v at Pin 4 of J5 (the 18-pin PIC socket). With switch S1 in PGM (DN) position, the voltage at Pin 4 of J5 is near zero.
47. **Test.** Turn the supply voltage off and then plug in the LCD. Observe that the LCD backlight is off when switch S1 is in PGM (DN) position and the LCD backlight is on when S1 is in RUN (UP) position.

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

### 3.4 Final Installation and Basic PIC-EL Test

Before you plug your 16F628A into the socket (J5) please make these basic checks once again.

1. **Check the voltage at TP-B.** It should be +5v. (See TP-B description above.)
2. **Check the voltage at TP-C**. It should be +5V.
3. **Check the voltage at DIP socket J5 Pin 14.** It should be +5v.
4. **Check the voltage at pin 2 of LCD jack, J4.** It should be +5V.
5. **Check the voltage at pin 3 of LCD jack, J4.** It should be approximately +0.3V.
6. **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 18-pin 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 III Computer Interface

Now you are ready to program a PIC using the PIC-EL III. Before you can do this you need to have a program installed in your PC. The PIC-EL III is connected to the PC via a cable connected to a USB port. **NOTE: It is *critical* that you install the programming software *before* you connect the USB cable between the PIC-EL III and your computer in for the first time**.

# 5 PIC-EL III Programming Software

### 5.1.1 Installing PICKit2 Programming Application (PICKit2 App)

The PICKit2 App software is on the CD that is supplied with the kit in the PICKit2 directory Otherwise you can download a workable version of PICKit2 App from the "DOWNLOADS" section of my web site: http://www.cbjohns.com/aa0zz or from the PIC-EL YAHOO group.

Note that PICKit2 App requires the installation of Microsoft **.**NET ("DotNet") Framework. This is a set of files that is freely distributed by Microsoft and is a basic element of many widely used applications so you may well have it installed already. (It takes about 90 MB of disk space.) DotNet is automatically included in Windows Vista but is easily installable on the earlier versions of Windows. You can see if you already have it installed on your computer by running the little application (on the CD) called **DetectDotNet.exe** . For more detailed look you can unzip and run the VersionCheck utility. Or you can manually check by doing the following:

- Open My Computer and navigate to ***C:\Windows\Microsoft.NET\Framework*** folder. If the folder cannot be found, you do not have any version of DotNet installed.

- If you have more than one version of DotNet installed you will see multiple directories listed. You need DotNet version 2.0 or later for PICKit2 App.

The CD contains two versions of the PICKit2 App installation files. **If you already have DotNet installed**, go to the "Microchip PICKit2 App" directory on the CD and then to the "PICKit2 only" directory. Unzip the files from the ZIP file called "PICkit2 Setup.zip". Then execute the **setup.exe** file and follow the installation prompts to install PICKit2 programming application on your PC. (The executable file for the PICKit2 App is called **PICkit2V2.exe**. You can ignore the other files in the installation directory.)

**If you want to install DotNet AS WELL as PICKit2 App**, go to the “Microchip PICKit2 App” directory on the CD and then to the “PICKit2 plus DotNet” directory and unzip the files from the ZIP file called “PICkit2 Setup plus DotNet.zip”. Go to the “dotnetfx” directory and execute the file called **dotnetfx.exe** . Follow the prompts to install DotNet on your PC. Then back up to the “PICKit2 plus DotNet” directory, execute the **setup.exe** file and follow the installation prompts to install PICKit2 programming application on your PC. (The executable file for the PICKit2 App is called **PICkit2V2.exe** . You can ignore the other files in the installation directory.)

**If you are using Windows 2000** you must be on Service Pack 3 or beyond. (An update to SP3 or beyond are downloadable from the Microsoft Update web site, www.update.microsoft.com.) Before installing DotNet you will have to install the Microsoft Windows Installer 3.0. It is also in the “dotnetfx” directory.

The CD also contains Microchip’s User guide for the PICKit2 application. You can refer to it for details regarding the usage of this application.

Although it is not necessary, please note that you can get updated versions of the PICKit2 App from the Microchip web page (www.microchip.com). However, please note that the version of the PICKit2 App that you use *must* be synchronized with the “operating system” code that is loaded into the 18F2550 PIC in the PIC-EL III. If you load a new version of PICKit2 App you will need to load the new “operating system” code also. Fortunately, you can update the PIC18F255 “in place” by clicking on the Tools tab of the PICKit2 App and then selecting “Download PICKit 2 Operating System”. You will then be prompted to select an “operating system” file that you have previously downloaded from Microchip and stored on your hard drive (for example, PK2V023200.hex). The PICKit2 App will load this code into the 18F2550 PIC in the PIC-EL, reset it to activate the new operating system, and then the programmer will be full functional.

### 5.1.2 Programming a PIC with PICKit2 App

The following steps should be followed to program a PIC in the PIC-EL.

1) **Install the PIC in the PIC-EL**

2) **Power up the PIC-EL**

3) **Slide mode switch S1 of the PIC-EL to PGM Position**

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

4) **Connect the USB cable from the PC to the PIC-EL**

Use an A-Male to B-Male USB cable to connect the PC to the PIC-EL.

If this is the first time you have connected the PIC-EL III to this computer, you will see a series of messages appear on the PC screen as the PICKit2 is detected and the appropriate drivers are automatically loaded. Wait until you see the message that says the hardware is ready to use. The next time(s) you connect the USB cable between the PC and the PIC-EL III you will not see the messages but you will simply hear the two tones (low then high) indicating the USB connection was made.

NOTE: If you do not hear the PC put out the two tones (low then high) indicating “USB Connected” you may have to reboot the PC. It’s rare but sometimes the USB interface of the PC gets hung up and can be fixed by a reboot.  You can check to see if the USB connection to the PICKit2 "clone" (i.e, the 18F2550 PIC and associated circuitry) in the PIC-EL III is working by going to your PC's Device Manager (Control Panel -> System -> Hardware -> Device Manager) and looking for the line called Human Interface Devices.  With the PIC-EL attached and properly "connected" you should see this item. Under Human Interface Devices, double-click on the USB Human Interface Device and the Location designator should say "Location 0 (PICKit 2 Microcontroller Programmmer)".  If you don't see this line, try rebooting your PC.

5) **Start the PICKit2 Application**

6) **Check the VDD Voltage on the PICKit2 App screen**

You should see 5.0v.  If not, the PIC-EL programmer is not working correctly and needs to be diagnosed.

Figure 15 shows the Microchip PICKit2 Application connected to the PIC-EL III and ready to read/write the PIC.

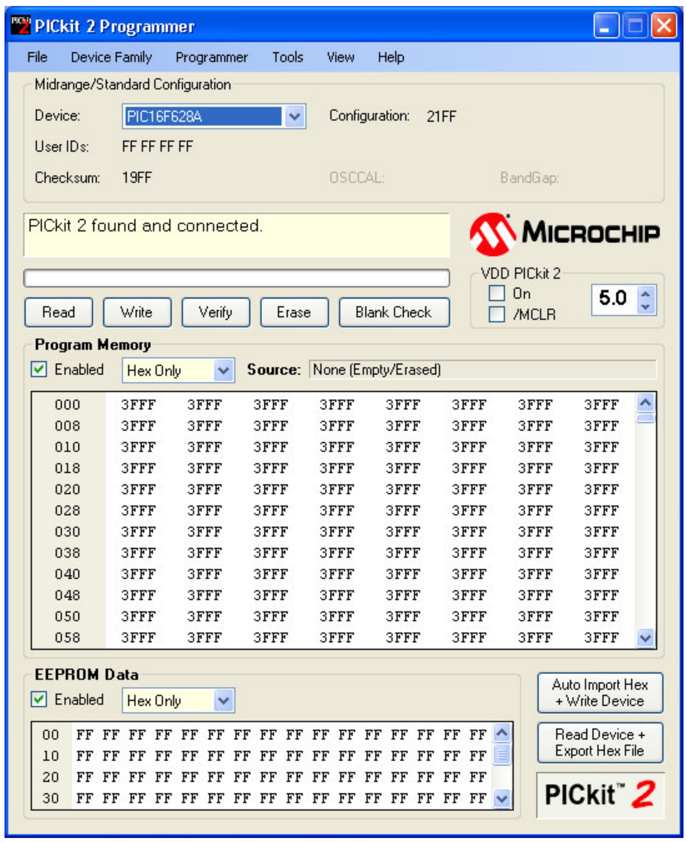



**Figure 15 – PicKit2 Programmer Application Connected to PIC-EL III**

7) **Erase the PIC currently plugged into the PIC-EL board (optional)**

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 and wait for the "Erasing device…. Complete" message.

8) **Import the .HEX file into PICKit2 Application**

The HEX file is the new software you will be burning into the PIC.  For example, you can get the PIC-EL III diagnostic program on the CD or you can download it from my web page or the FILES section of the PIC-EL YAHOO group.  The file "USBDiag1-1.HEX" 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 programming software.

Click on the **File** tab, then click on “Import Hex” and navigate to wherever you have the unzipped the PIC-EL III diagnostic file on your PC. Double-click on the USBDiag1-x.HEX file and the HEX ASCII code will load into the PICKit2 App buffer. You will see that code in the PICKit2 App window.

9) **Write the new code to the PIC**

Now you are all set to burn the program into the PIC. Click on the **Write** button and wait while status window displays “Writing Device: Program Memory…EE… UserIDs… Config… Done”. Then click on the **Verify** button and wait for the display to indicates “Verify Device… Program Memory…EE… UserIDs… Config”. Then the status box will turn green and display “Verification Successful”.

If it says “Programming failure”, you obviously have a problem that you need to diagnose. See the PICKit2 User Guide for details. If this is the first time you have tried to program a PIC in the PIC-EL, there are several diagnostics built into PICKit2 App that may be useful. For example, click on the **Tools** tab and then click on “Check Communications”. You should see a status message appear which says “PICKit 2 found and connected”. If it does not appear, you need to check the programmer components and solder connections.

Assuming you get the “PICKit 2 found and connected” message but get error messages when trying to Read or Write the target PIC in the PIC-EL, you can use some of the PICKit2 App diagnostic capabilities. Click on the **Tools** tab and then select “Troubleshoot”. You can click on Verify VDD and see that the VDD voltage is close to 5.0v. Then, on the next screen, you can check the VPP voltage by clicking on the test button. It should be close to 12v. You can check the MCLR voltage by clicking on the /MCLR ON button and checking the voltage on pin 4 of the PIC. It should be approximately near zero. The /MCLR OFF test is not important. Clicking on /MCLR OFF button will yield a non-zero voltage on Pin 4 but it will not be +5v because the PIC-EL doesn’t pull it up to 5v. That’s OK, since Pin 4 will get +5v when the PGM/RUN switch is switched to RUN position. (Try it.) On the next screen you will be able to toggle the clock line or the data line at a speed of 30 kHz. This is very handy for checking the connections to the target PIC. (You will need an oscilloscope to see these pulses at the target PIC.) You can also check to see if the voltages on the clock and data lines go from full +5v when turned ON to zero when turned OFF. If these pulses are not seen or the voltages are not correct on the appropriate PIC pins you need to find out why.

10) **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 to the RUN position and the PGM LED will turn off once again.

11) **RESET the PIC-EL to start up the new program**

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

# 6 PIC-EL Diagnostic Program

A PIC-EL diagnostic program is pre-loaded on the 16F628A PIC that comes on the CD with the PIC-EL III kit. This HEX code as well as the test symbolics file is on the CD and 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.

## 6.1 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.

## 6.2 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.

## 6.3 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.

## 6.4 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.

## 6.5 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.

### 6.6 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.

### 6.7 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, 7.0405 MHz and 7.041 MHz. (They are displayed on the LCD in Hz as 7040000, 7040500 and 7041000.) 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. If you tune to 7.041 MHz you can probably hear all three tones in sequence.

### 6.8 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, 7.0405 MHz, and 7.041 MHz. You can observe the output on a spectrum analyzer (or oscilloscope) or by tuning your HF receiver to 7.040 MHz. If you tune to 7.041 MHz you can probably hear all three tones in sequence.

## 7 Elmer-160 Lessons

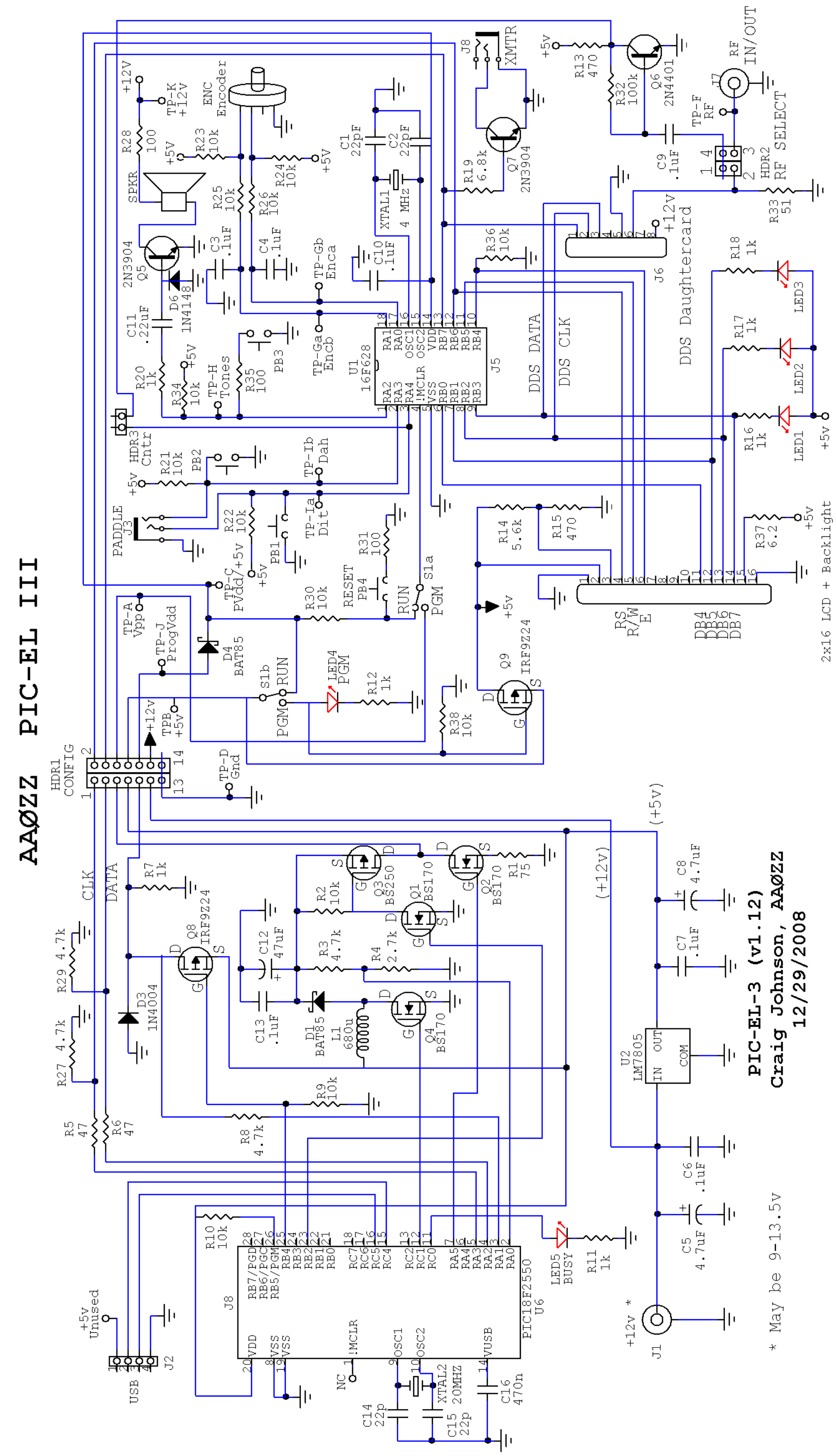
If you are interested in John McDonough's (WB8RCR) Elmer-160 lessons, you can find them on the AmQRP web site: http://www.amqrp.org/elmer160. John's lessons are available for downloading and you can contact him by using links provided on that web site.

## 8 Support

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

For additional support questions, see the PIC-EL YAHOO group: (http://www.groups.yahoo.com/group/PIC-EL) or Email me directly.

# Appendix A – Schematic



# Appendix B – Parts List

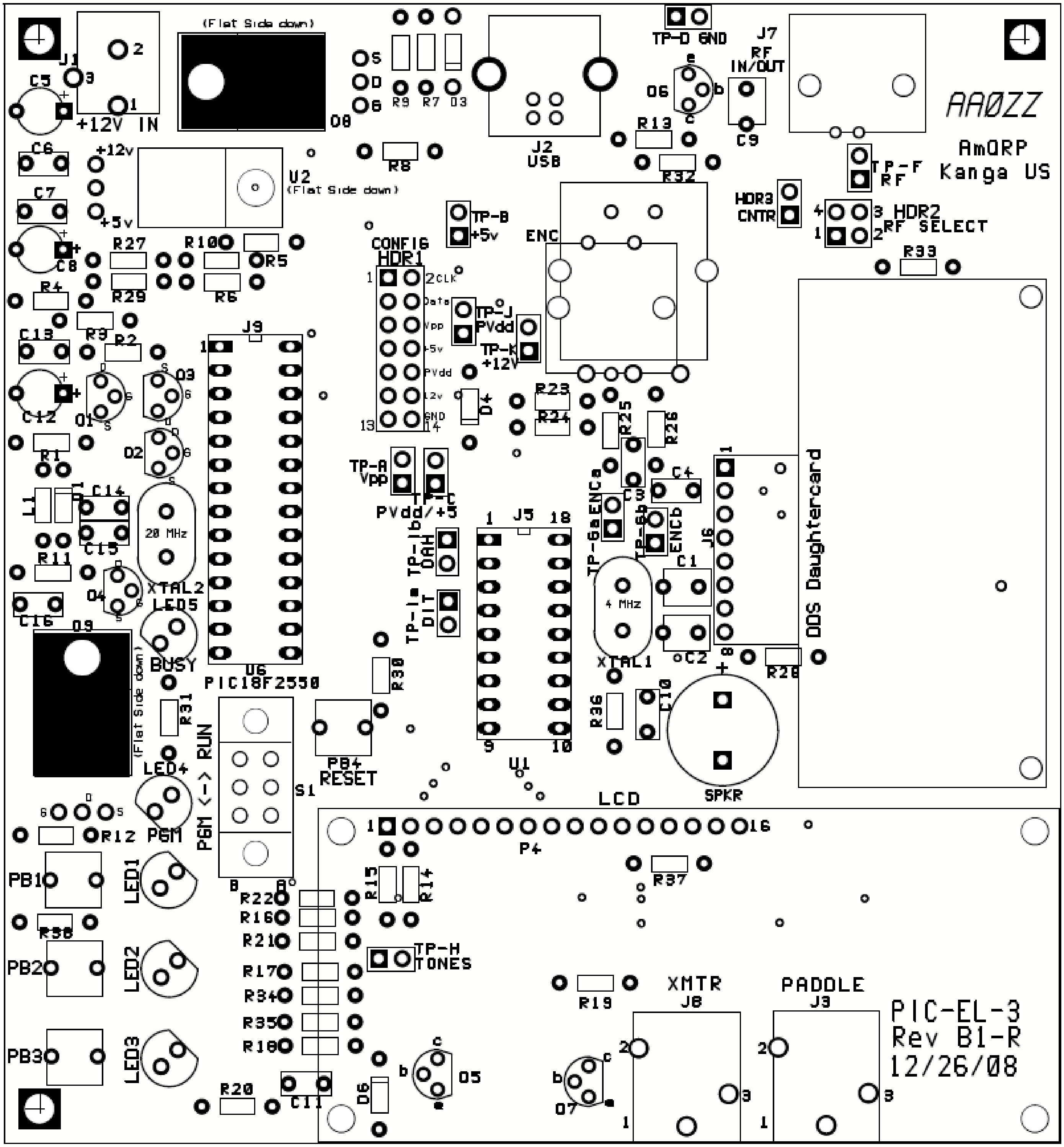
| | | PIC-EL (Version 3) Parts | | | 7/10/2009 |
|---|---|---|---|---|---|
| **Quantity** | **Designator** | **Description** | **(D) DigiKey (M) Mouser or Other** | **Small Quantity Price** | **Total at Small Quantity Price** |
| | | | | | |
| **Resistors** | **(1/8 or 1/4 watt, 5%)** | | | | |
| 1 | R37 | 6.2 (blue, red, gold) | (D) 6.2EBK-ND<br>(M) 291-6.2-RC | 0.05 | 0.05 |
| 2 | R5. R6 | 47 (yellow, violet, black) | (D) 47EBK-ND | 0.05 | 0.10 |
| 1 | R33 | 51 ohms (green, brown, black) | (D) 51EBK-ND<br>(M) 291-51-RC | 0.05 | 0.05 |
| 1 | R1 | 75 ohms (violet, green, black) | (D) 75EBK-ND | 0.05 | 0.05 |
| 3 | R28, R31, R35 | 100 ohms (brown, black, brown) | (D) 100EBK-ND<br>(M) 291-100-RC | 0.05 | 0.15 |
| 2 | R13, R15 | 470 ohms (yellow, violet, brown) | (D) 470EBK-ND<br>(M) 291-470-RC | 0.05 | 0.10 |
| 7 | R7, R11, R12, R16, R17, R18, R20 | 1K (brown, black, red) | (D) 1.0KEBK-ND<br>(M) 291-1K-RC | 0.05 | 0.35 |
| 1 | R4 | 2.7k (red, violet, red) | (D) 2.7KEBK-ND | 0.05 | 0.05 |
| 4 | R3, R8, R27, R29 | 4.7K (yellow, violet, red) | (D) 4.7KEBK-ND | 0.05 | 0.20 |
| 1 | R14 | 5.6K (green, blue, red) | (D) 5.6KEBK-ND<br>(M) 291-5.6K-RC | 0.05 | 0.05 |
| 1 | R19 | 6.8K (blue,gray, red) | (D) 6.8KEBK-ND<br>(M) 291-6.8K-RC | 0.05 | 0.05 |
| 13 | R2, R9, R10<br>R21, R22, R23, R24, R25,<br>R26, R30, R34, R36, R38 | 10K (brown, black, orange) | (D) 10KEBK-ND<br>(M) 291-10K-RC | 0.05 | 0.65 |
| 1 | R32 | 100K ohms (brown, black, yellow) | (D) 100KEBK-ND<br>(M) 291-100K-RC | 0.05 | 0.05 |
| | | | | | |
| **Capacitors** | | | | | |
| 4 | C1, C2, C14, C15 | 22 pF, ceramic disc ("22") | (D) 495-1004-1-ND<br>(M) 140-50N2-220J-RC | 0.06 | 0.24 |
| 7 | C3, C4, C6, C7, C9, C10<br>C13 | 0.1 uF, monolithic, (small yellow, "104")<br>(or blue) | (D) P4910-ND<br>(M) 80-C317C104M5U | 0.07 | 0.49 |
| 1 | C11 | .22 uF, monolithic (224) | (D) 445-2849-ND | 0.21 | 0.21 |
| 1 | C16 | 0.47 uF, ceramic radial ("474") | (D) BC1150CT-ND | 0.20 | 0.20 |
| 2 | C5, C8 | 4.7 uF, electrolytic | (D) P996-ND<br>(M) 140-XRL16V4.7-RC | 0.08 | 0.16 |
| 1 | C12 | 47 uF, electrolytic | (D) P969-ND | 0.1 | 0.10 |
| | | | | | |
| **Diodes** | | | | | |
| 2 | D1, D4 | BAT85 | (D) 568-1617-1-ND | 0.28 | 0.56 |
| 1 | D3 | 1N4004 | (D) 1N4004-E3/73GI-ND | 0.05 | 0.05 |
| 1 | D6 | 1N4148 | (D) 1N4148FS-ND<br>(M) 625-1N4148 | 0.07 | 0.07 |
| | | | | | |

| | | | | | |
|---|---|---|---|---|---|
| **Inductors** | | | | | |
| 1 | L1 | 680u Inductor (green body, blue, gray, brown) | (D) M8156-ND | 0.30 | 0.30 |
| | | | | | |
| **Transistors** | | | | | |
| 2 | Q5, Q7 | 2N3904 transistor, NPN TO-92 | (D) 2N3904D26ZCT-ND | 0.19 | 0.75 |
| 1 | Q6 | 2N4401 transistor, NPN TO-92 | (D) 2N4401-ND<br>(M) 610-2N4401 | 0.12 | 0.12 |
| | | | | | |
| **MOSFETs** | | | | | |
| 2 | Q8, Q9 | IRF9Z24 (TO-220) | (D) IRF9Z24PBF-ND | 1.58 | 3.16 |
| 3 | Q1. Q2, Q4 | BS170 (TO-92) | (D) BS170-ND | 0.34 | 1.02 |
| 1 | Q3 | BS250 (TO-92 but two sides flat) | (D) BS250P-ND | 1.27 | 1.27 |
| | | | | | |
| **Other** | | | | | |
| 1 | U2 | L7805 voltage regulator, 5V (TO-220) | (D) 497-1443-5-ND<br>(M) 511-L7805 ABV | 0.67 | 0.67 |
| 1 | LCD | Liquid Crystal Display, 2x16 char | 08LCD9 from ElectronixExpress.com | 8.95 | 8.95 |
| 4 | LED1, LED2, LED3, LED4 | LED, T1-3/4 (red) | (D) 67-1110-ND | 0.12 | 0.48 |
| 1 | LED5 | LED, T1-3/4 (green) | (D) 67-1109-ND | 0.12 | 0.12 |
| 1 | XTAL1 | Crystal, 4 MHz | (D) X405-ND<br>(M) 520-HCU400-20 | 0.58 | 0.58 |
| 1 | XTAL2 | Crystal, 20 MHz | (D) CTX416-ND | 0.80 | 0.80 |
| 1 | J1 | coaxial power jack, 2.1mm | (D) CP-102AH-ND | 0.42 | 0.42 |
| 2 | J3, J8 | audio jack, 1/8", stereo | (D) CP1-3513N-ND<br>(M) 161-3507-E | 0.79 | 1.58 |
| 1 | P4 | SIP socket, 16-pos'n (for PCB) | (D) ED7150-ND<br>(M) 517-974-01-16 | 1.69 | 1.69 |
| 1 | J4 | SIP header, 16-pos'n (for LCD) | (M) 571-16404526 | 1.10 | 1.10 |
| 1 | J5 | DIP socket, 18-position (for PIC) | (D) ED3118-ND | 1.01 | 1.01 |
| 1 | J2 | USB B receptacle | (D) WM17131-ND | 2.30 | 2.30 |
| 1 | J6 | SIP socket, 8-pos'n, 90-deg | SamTec<br>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 | J9 | DIP socket, 28-position (for PIC) | (D) ED3128-ND | 1.68 | 1.68 |
| 1 | HDR-1 | pin header, 0.1", 2x7 pos'n | (M) 571-4-103328-3 | 0.74 | 0.74 |
| 9 | Shunts for HDR1, HDR2, HDR3 | shunt, 0.1", 2 pos'n | (D) S9000-ND<br>(M) 571-2-382811-1 | 0.09 | 0.81 |
| 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<br>or (D) P12334-ND<br>or (D) P12335-ND | 1.52 | 1.52 |
| 1 | SPKR | speaker | (D) 433-1028-ND | 0.87 | 0.87 |
| 3 | machine screw (6-32 x 3/8") | (for U2, Q8, Q9 ) | (D) H356-ND | 0.02 | 0.06 |
| 3 | machine nut (6-32) | (for U2, Q8, Q9 ) | (D) H220-ND | 0.01 | 0.01 |
| 4 | machine screw (4-40 x 1/2") | (for corner standoff insulators) | (D) H346-ND | 0.02 | 0.08 |
| 4 | machine nut (4-40) | (for corner standoff insulators) | (D) H216-ND | 0.01 | 0.04 |

| | | | | | |
|---|---|---|---|---|---|
| 4 | Standoff insulator - 1/4" | (for four corners) | (D) 876K-ND | 0.15 | 0.60 |
| 1 | U1 | PIC16F628A microcontroller with diagnostics pre-loaded | (D) PIC16F628A-I/P -ND with AA0ZZ diagnostics | 7.50 | 7.50 |
| 1 | U6 | PIC18F2550 microcontroller with PICKIT2 code pre-loaded | (D) PIC18F2550-I/SP-ND with code | $15.00 | 15.00 |
| 1 | PCB | PC Board | AA0ZZ | 20.00 | 20.00 |
| 1 | CD | CD with documentation | AA0ZZ | 9.00 | 9.00 |
| | | | | | ======= |
| | | | | **Single cost** | **94.52** |

# Appendix C – PIC-EL III  Parts Placement



# Appendix D – PIC-EL III  PC Board