

*FOX2*  
*PIC Controlled*  
*PLL Synthesized*  
*Low Cost*  
*Fox Transmitter*

Designed by:  
James Lee  
N1DDK  
March 1999

## Table of Contents

Table of Contents .....	2
FOX2 programming instructions.....	3
Using the serial interface.....	3
Starting and connecting to the serial interface.....	3
Using the Push button interface.....	4
In delay set mode.....	4
In frequency select mode.....	4
In arm mode.....	4
Once Armed.....	5
Setting up in the field.....	5
Assembly and Alignment .....	6
Tools and other supplies needed.....	6
General notes.....	6
Initial assemble.....	6
Digital control assembly.....	7
Initial test.....	7
VCO assembly and alignment.....	8
Initial alignment.....	8
PA assembly and alignment.....	8
Final assembly.....	9
Suggested Modifications.....	9
Notes from first prototypes.....	9
LED Template.....	11
Coil Tables.....	12
Coils used in the first prototype.....	13
Alignment and other tricks in the software.....	14
Programming the PLL.....	14
PLL programming example.....	14
Suggested alignment procedure.....	14
Kit Levels.....	15
Schematic.....	16
About RS232 Level Converts.....	17
Sample RS232 Converter.....	17

## FOX2 programming instructions

- 1) First set the ID message and frequencies through the serial interface.
- 2) The Push button interface can be used in the field to set the powerup delay and chose one of three preprogrammed frequencies.

### *Using the serial interface*

The serial interface is used to program:

- The ID message
- The off time between transmissions
- The Hang time between transmissions
- The speed of the Morse ID
- The Three frequencies.

The interface can also be used to verify all of the settings.

### Starting and connecting to the serial interface.

To connect to the serial interface, you need a terminal or terminal program set to 300 baud, and a RS232 to TTL or CMOS level converter. A schematic for an inexpensive Level converter is shown in the appendix.

- 1) Set the terminal program to 300 baud, 8 data bits, no parity, no flow control.
- 2) Power the converter and the fox. The fox should echo the Morse code it is sending to the terminal. If you can hear the Morse code, but do not see anything on your terminal, check that you have not swapped the TXD and RXD between the fox and the converter or the converter and your terminal.
- 3) Once you can see the message being typed out slowly, you are ready to wake the interface by hitting <ENTER> or <RETURN>.
- 4) The message should stop and you should see a brief help message.
- 5) Chose from the commands explained below to configure your fox.

Command	Argument	Meaning
dl	number 1-255	Length of a DIT in 1/1200 of a second. 90 is about 13 words per minute. The effect is not seen until you eXit.
ht	number 0-255	hang time in seconds. After the ID is sent; the transmitter will have an unmodulated carrier for this number of seconds.
id	any message up to 40 characters	set the ID message. A leading space and trailing space are a good idea for some key-up and hang-time.
ot	number 0-255	off time in seconds. Between transmissions the transmitter will be off for this many seconds. A zero is a continuously on transmitter.
n1	number 0-255	set frequency 1 (see details on calculating values)
n2	number 0-255	set frequency 2
n3	number 0-255	set frequency 3
v	(no arguments)	Verify – Print out all the settings
x	(no arguments)	exit – exit the command mode and reboot; some changes such as DIT Length are not in effect until you exit and it reboots.

### ***Using the Push button interface***

If you want to change the frequency to one of the alternate frequencies you pre-programmed, or if you want a delay before the transmitter comes on, you can use the push button interface.

To activate the push buttons interface; Hold the MODE button when applying power. The time LED should blink, you are in delay set mode.

#### **In delay set mode**

Pressing SET will add 7.5 minutes to the start delay, the first 4 LEDS are 7.5, 15, 30, and 60 minutes a total delay of 112 minutes is possible. If you select too long a delay it will reset to zero by continuing to press the set button past lighting all the LEDS.

Pressing MODE will move you to frequency select mode. The Frequency select LED will blink.

#### **In frequency select mode**

Pressing SET will cycle between your three preset frequencies.

Pressing MODE will move you to arm mode. Both LEDS should blink.

#### **In arm mode**

The LEDS will show you the delay start time. The 60 minute indicator will be if the LEDS are flashing at the same time or alternatively. Same time, 60 minutes is not on, alternatively 60 minutes is on.

Pressing SET will start the delay counter counting down. So you can select a time less than what you need and wait in the arm mode until it is time to start it.

Pressing MODE will mode you to delay set mode.

### Once Armed

If there is no delay, the transmitter will come on, if a delay is programmed the LEDS will blink the time remaining until the transmitter will come on.

### ***Setting up in the field.***

If you want it to come on immediately on the default frequency, simply apply power. If you need to change from the default frequency or add a start up delay, hold the MODE button when applying power, and follow the directions under the push button interface.

## **Assembly and Alignment**

*Read the following sections on suggested modifications, and notes from the first prototype before proceeding.*

### ***Tools and other supplies needed***

- Soldering supplies
- Drill
- Power switch
- Battery and Connectors
- Hookup wire
- Spectrum analyzer
- Frequency counter
- Antenna analyzer or network analyzer
- Screwdrivers

### ***General notes***

The output filter as shown in the original schematic is was incorrect please disregard L3, L4, C8, C16, and C17 on the board layout; Follow the directions below. See the output filter schematic for correct circuit.

Since the board has no silk screen for the component location pay careful attention to attached drawings for component locations.

The instructions below are for reference, you may assemble it any way you want. The instructions to insert a component are a suggested order, solder and clip components as you insert them. You may want to insert many components in an area of the board before you solder them to be sure you have the proper components in the proper locations.

### ***Initial assemble***

Start with the power supply components.

1. Connect 18-22 awg wires to the holes marked b+ and gnd.
2. Insert C21 and C22, They can be any value around 10uf, I think I used 22's or 50's I had on hand. Note the polarity is backwards in the drawing, so insert the negative terminal of the caps where the + is on the drawings.
3. Insert U3, the 78L05 regulator, a low dropout regulator may be substituted if you expect to be running it on less than 7 volts.
4. Insert C1, near the corner of the board by C22,
5. Insert C23 near the top of U2, be careful not to use one of R15's holes. C23 is in line with U2, but a bit above it.
6. Insert a socket at U1.

Apply power, check for +5 at U1 pin 14 and ground at U1 pin5.

## **Digital control assembly**

Before completing this section read about some suggested modifications in case you decide to try any of them.

1. Insert R1. 10K
2. Insert R16, R17, R18, R19, and R20. 470  $\Omega$
3. Insert R21 and R22. 10K
4. Insert LED1, LED2, LED3, LED4, LED5. You may want to set the height of them to line up with your enclosure, or just install leads if you are going to remotely mount them. All of the LEDs share a common ground. So you can save some wires if you are going to remotely mount them. Check the polarity of your LEDs. The LED should light when the end you place the negative terminal or cathode (generally this is the side with the flat) towards the outside edge of the board is negative.
5. Insert SW1 and SW2. The supplied switches will mount in the board for access through a fairly thin enclosure. If you wish to remote mount some push buttons just install leads. The buttons also share the common ground. You may need to straighten the legs of the mounting tabs to make the fit more easily through the holes in the board. The mounting holes are grounded. Be careful when soldering in the buttons since the clearance between the mounting holes and the active pins is quite small.
6. Insert Y1. (read about the possible modifications) (the small holes close to the PIC)
7. Insert R4, R3, C2, C3. Do not install C4 at this time.
8. Insert a 330 K resistor for R5, (the original value was 100K but there was too much deviation)
9. Add a 470 ohm resistor from the C3,R3,R5 junction to ground. I found the best place to do this was the back of the board. Although you could simply put this resistor across C3, I found that it picked up RF from the LO if it was tacked across C3 on the top of the board.
10. Insert R2, (read about possible modifications) This is the current limiting resistor to the PTT transistor. Although this is connected to an open drain output there is a small leakage current that may keep the PTT transistor on. Add a 10K resistor, RE1, from VBAT to PTT to help the transistor turn off. The easiest way to do this to connect it to the two traces on the top of the board very near the B+ wire. (I folded one lead 180 degrees like inserting a standing resistor, and then cut the leads about ¼ inch from the bottom of the resistor, bent them 90 degrees the same way and lined the leads up on the traces that come from B+ and R2.)
11. Insert leads at the TXD RXD and Ground pins for the serial interface. I recommend running these to a stereo mini-jack. With the shield as ground, TXD connected to the ring, and RXD connected to the tip. The locations for these leads are in the 5 hole connector near the LEDs. The hole closest to the edge is ground, The holes farthest from ground are unused. RXD is the hole towards the top of the board, TXD is the hole towards the LEDs.

## **Initial test**

1. Plug in the pre programmed PIC and apply power. LED 5 should come on solid. Optionally connect and audio amplifier and speaker to R3 and listen for the Morse code message.
2. Remove power.
3. Hold the mode button while applying power. LED5 should blink. You can try some of the programming modes as outlined in the programming section. You could also build the RS232 level converter and play with programming it.

You should now have a working digital fox controller that can be connected to any transmitter. If you want to connect it to an external transmitter the audio is available at R5, and PTT is available at R2. Note that the audio level is quite low as it is used for direct modulation here.

## ***VCO assembly and alignment***

Insert all the VCO components. This is a tricky area of the board, so I suggest inserting them all before soldering.

1. Insert D1, Q3, and Q4. (read about possible modifications about Q3)
2. Insert C5, C6, C7, C9, C10, C11, R6, R8, R9, R10, R11
3. Insert a coil at L1. L1 is approximately 0.021uh, This is 2 ½ turns of #20 wire on a 8/32 screw. Consult the Coil chart at the end of these instructions.

## **Initial alignment**

1. Temporarily attach a capacitor to one end of C12, where it attaches to R11, and connect the loose end to a frequency counter or spectrum analyzer.
2. Temporarily connect a wire to the C11 / R10 junction. Connect this wire to the wiper of a potentiometer (5-100K), what every you have handy, connect the ends to the potentiometer to +5 and GND.
3. Apply Power, and run the potentiometer through it range and observe the useful range of the VCO. Adjust L1 accordingly. Target 146.5 Mhz for around 2-3 volts.

## ***PA assembly and alignment***

1. Attach a coax, connector, and dummy load to the output, or just a 50 ohm resistor across the output terminals.
2. Insert a 680pf capacitor at C17.
3. Insert a 20pf capacitor at L4.
4. Insert a 33pf capacitor at L3.
5. Add a 20pf capacitor from Q2's collector stripline just beyond the 90 degree bend to ground.
6. The two coils in the output filter are about 0.069 uh. Consult the coil charts. I wound them as 5 ½ turn on a 8/32 screw, spread out slightly. I found that they coupled so I mounted one to the back of the board. Insert a coil at C16, on the top of the board and C8, on the back of the board.
7. Insert C14 and R12.
8. Temporarily attach an antenna analyzer or network analyzer to the collector terminal of Q2 and ground. Tune the coils for a 50 ohm impedance at the desired center frequency. (the output of Q2 is about 50 ohms)
9. Insert c15.
10. L2 is about 0.12 uh, I had a 4.5 turn coil but I think this was a bit to small. Insert a coil at L2.
11. Tune L2 so you have the same impedance you had in the previous tuning of the LPF.
12. Disconnect the analyzer.
13. Insert Q1 and solder on Q2.
14. Insert C12 and C13.
15. Add RE3 a 10k resistor from Q2's base to ground. This is most easily done from the end of the stripline near C12 and then place the resistor parallel to the stripline.
16. Add RE2, *See Notes*, a 10K resistor from Q2's base to +5 volts. The is most easily done from then end of C10 nearest C12 to the same point as the other resistor you just added. The resistor cal lie down next to C12, or you can place it on the back of the board. These two resistors apply a bias voltage to the final to help it turn on. There are several notes about this bias voltage. The transistor takes quite a bit of bias current.

## **Final assembly**

Inset the remaining components, the order is unimportant as there are no test steps in this section.

1. Insert C4 to couple in the modulation audio.
2. Insert C20 and R15. To add the crystal drive to the PLL.
3. Add the components in the loop filter R7, R13, R14, C18, C19.
4. Insert U2, I would not recommend using a socket since there are RF signals here.
5. Trim the oscillator by adding a 20pf cap (CE1) from pin 5 to pin 15 of U1 on the back of the board.
6. Add a trim cap (CE2) from pin 5 to pin 16 of U1 on the back of the board.
7. (add any other components I may have forgotten to instruct)

## **Suggested Modifications**

Some of these ideas or others may improve the design.

1. Change Q3 to a gain stage of about 6 DB so the full 1.5 watts can be driven out of Q2.
2. Add trim capacitors to the crystal oscillator.
3. Change the crystal oscillator configuration: Move the crystal over to the PLL, and change it to 16Mhz and add trim capacitors. Take the REFOUT output from the PLL and drive through a resistor to the PIC. Since the PLL defaults to divide by 8, the PIC clock will come up at 2Mhz. The software currently disables REFOUT, so it will need to be changed to enable it at divide by 4.
4. Change the configuration of PTT keying. Apply B+ directly to the final (through its tank circuit). Use the keying transistor to turn on and off the VCO.
5. Decrease C6 to decrease the VCO gain
6. The junction of R10/C11/C4/R13/R14 can be used to inject AC coupled high impedance audio. An additional audio source such as a voice recorder or sound effects chip could be injected here.

## **Notes from first prototypes**

1. The VCO is sensitive to vibrations, so I might go with a factory coil.
2. Also the VCO "leaks" quite a bit so you can pick it up when the output stages are not keyed. I wonder if some of that is just noise coupled to the board and some well placed ferrite beads might help. I think a shielded factory coil might help.
3. I did not get the tank coil quite into resonance. I added a turn to it, having started with a 3.5 turn coil, but I think it may need to be 5.5 turns.
4. The coils in the LPF coupled to each other and the VCO, so I am tempted to use factory shielded coils here as well. Another thought is to use smaller coils throughout.
5. I was only about to get about 4V PP or about 320 mw out of the final, so there needs to be more drive from Q3 or a driver stage added.
6. For some reason my frequencies were 0.02% high, so I compensated for this in software, this is why I suggest either of the crystal oscillator modifications. I was able to trim all the oscillators we built with a 20 pf cap on the output and trimmer on the input, so these are added to the instructions and the kit.
7. JP finds that the bias on the final from RE2 and RE3 was insufficient. He found only about 0.2 volts, so he decreased re2 to 33k and connected it to R12 so it was also turned on and off with the PTT switch. He was unsure which side of the resistor he hooked it to, but had substantially higher output level.
8. JP finds the slight chip when the tone kicks on due to the dc offset in the output of the audio filter that the PLL need to compensate.

9. JP also finds that the modulation waveform is not well rounded due to the shunt resistor re4 I added to cut the modulation voltage.
10. Here are some suggestions for modifying the audio low pass filter
  - decrease C4 by a factor of 10, (even 100 will be ok) this will further reduce the dc offset voltage
  - R3 and R4 can be increased, replace R4 with three resistors, to create a voltage divider with a high output impedance.
11. I moved L3 and L4 near the end of the board and added a small shield between l4 and C9, this also shielded L1 from L3.
12. The first prototype I in fact had the values of RE2 and RE3 reversed, So I tried JP's Mod. It worked fine on either end of R12. Since the bias is now affected by the input voltage, the output power varies more with the B+ voltage. Perhaps a value of 10K -20K to the regulated +5 would be optimal.

## LED Template

The following template can be added to the enclosure to label the functions of the LEDs. A second template (without frequencies is provided for you to write in your own)

<b>Delay</b>	<b>Frequency</b>	<b>Ready</b>	<b>TX</b>
7.5 min	146.415		
15 min	146.490		
30 min	146.565		
60 min	<i>Blink</i>	<i>Blink</i>	
<i>Blink</i>		<i>Blink</i>	ON

<b>Delay</b>	<b>Frequency</b>	<b>Ready</b>	<b>TX</b>
7.5 min			
15 min			
30 min			
60 min	<i>Blink</i>	<i>Blink</i>	
<i>Blink</i>		<i>Blink</i>	ON

## Coil Tables

Coils on an 8/32 screw form

diameter	turns	length	l(uh)
0.17	1.5	0.046875	0.013176
0.17	2.5	0.078125	0.029204
0.17	3.5	0.109375	0.047616
0.17	4.5	0.140625	0.067383
0.17	5.5	0.171875	0.087994
0.17	6.5	0.203125	0.109166
0.17	7.5	0.234375	0.13073
0.18	6.5	0.203125	0.120449
0.17	6.5	0.203125	0.109166
0.17	6.5	0.15	0.134771
0.17	6.5	0.3	0.081077
0.17	6.5	0.203125	0.109166

Coils on an 6/32 screw form

diameter	turns	length	l(uh)
0.14	1.5	0.046875	0.010034
0.14	2.5	0.078125	0.021701
0.14	3.5	0.109375	0.034822
0.14	4.5	0.140625	0.048729
0.14	5.5	0.171875	0.063108
0.14	6.5	0.203125	0.077792
0.14	7.5	0.234375	0.092686
0.14	8.5	0.265625	0.107729
0.14	9.5	0.296875	0.122883
0.14	10.5	0.15	0.253627
0.14	11.5	0.3	0.178519
0.14	12.5	0.390625	0.168779

Coils on an 4/40 screw form

diameter	turns	length	l(uh)	
0.12	1.5	0.0375	0.008852	
0.12	2.5	0.0625	0.019313	
0.12	3.5	0.0875	0.031166	
0.12	4.5	0.1125	0.043784	
0.12	5.5	0.1375	0.056867	
0.12	6.5	0.1625	0.070254	
0.12	7.5	0.1875	0.083851	
0.12	8.5	0.2125	0.097598	
0.12	9.5	0.2375	0.111458	
0.12	10.5	0.2625	0.125403	
0.12	11.5	0.2875	0.139414	
0.12	12.5	0.3125	0.153479	
0.12	13.5	0.3375	0.167586	
0.12	14.5	0.3625	0.181729	
0.12	15.5	0.3875	0.1959	
0.12	16.5	0.4125	0.210096	
0.12	17.5	0.4375	0.224313	
0.12	18.5	0.4625	0.238548	
0.12	19.5	0.4875	0.252798	

Coils used in the first prototype

	diameter	turns	length	l(uh)	l(mm)
VCO	0.17	2.5	0.137795	0.021072	3.5
tank	0.17	4.5	0.161417	0.061495	4.1
LPF	0.17	5.5	0.23622	0.069889	6

## Alignment and other tricks in the software

With the terminal connected there are a few command that help with alignment

Command	Argument	Meaning
tf	none	Turn off the transmitter (unkey) Also tern off the tone
tx	none	Turn on the trnasmitter.
tn	none	Turns on the tone, does not affect the TX state.
ha	number 0-255	Hack subsequent EEPROM locations 9se with extreme care.

EEPROM Locations and definitions

0. Timer 0 value, this sets a 1200 HZ interrupt rate
1. DIT length also modifiable with the DL command
2. Hang time also set able with the HT command
3. Off time also set able with the OT command
4. CVAL the 8 bit control mode programmed into the PLL
5. RVALH The high 7 bits of the R value for the PLL
6. RVALL The low 8 bits of the R value for the PLL
7. NVALH1 The High 8 bits of the N value for frequency 1.
8. NVALL1 The Low 8 bits of the N value for frequency 1.
9. NVALH2 The High 8 bits of the N value for frequency 2.
10. NVALL2 The Low 8 bits of the N value for frequency 2.
11. NVALH3 The High 8 bits of the N value for frequency 3.
12. NVALL3 The Low 8 bits of the N value for frequency 3.
13. Start of ID string which is a Null terminated ASCII string, this is most easily set with the ID command.

### **Programming the PLL**

- The R value is programmed to 4Mhz/5Khz =800(decimal)
- N1, N2, and N3 are the frequencies for the PLL, They are the frequency divided by 5Khz, the frequency step.
- The default for the upper 8 bits of all of them is 72 hex or 114 decimal, this is a base frequency of 145.92Mhz.
- The low part of the frequency is the number of 5Khz above 145.92Mhz,
- The low part of each N may be 0-255 which allows for frequencies from 145.92 to 147.195 without reprogramming the high part of the N registers.

### PLL programming example

$146.565 - 145.92 = .645\text{Mhz}$   $645/5 = 129$ .

### **Suggested alignment procedure**

Hook up the terminal interface and set N1 to 116, this will set the frequency 146.500. Type x to exit or turn the power off and on again. Now you can easily check the frequency and adjust the trim cap.

Default Values N1, N2, N3.

Channel	Frequency	Value
N1	146.415	99
N2	146.490	114
N3	146.565	129

## Kit Levels.

The Fox kit is available in several levels shown in descending level of completeness.

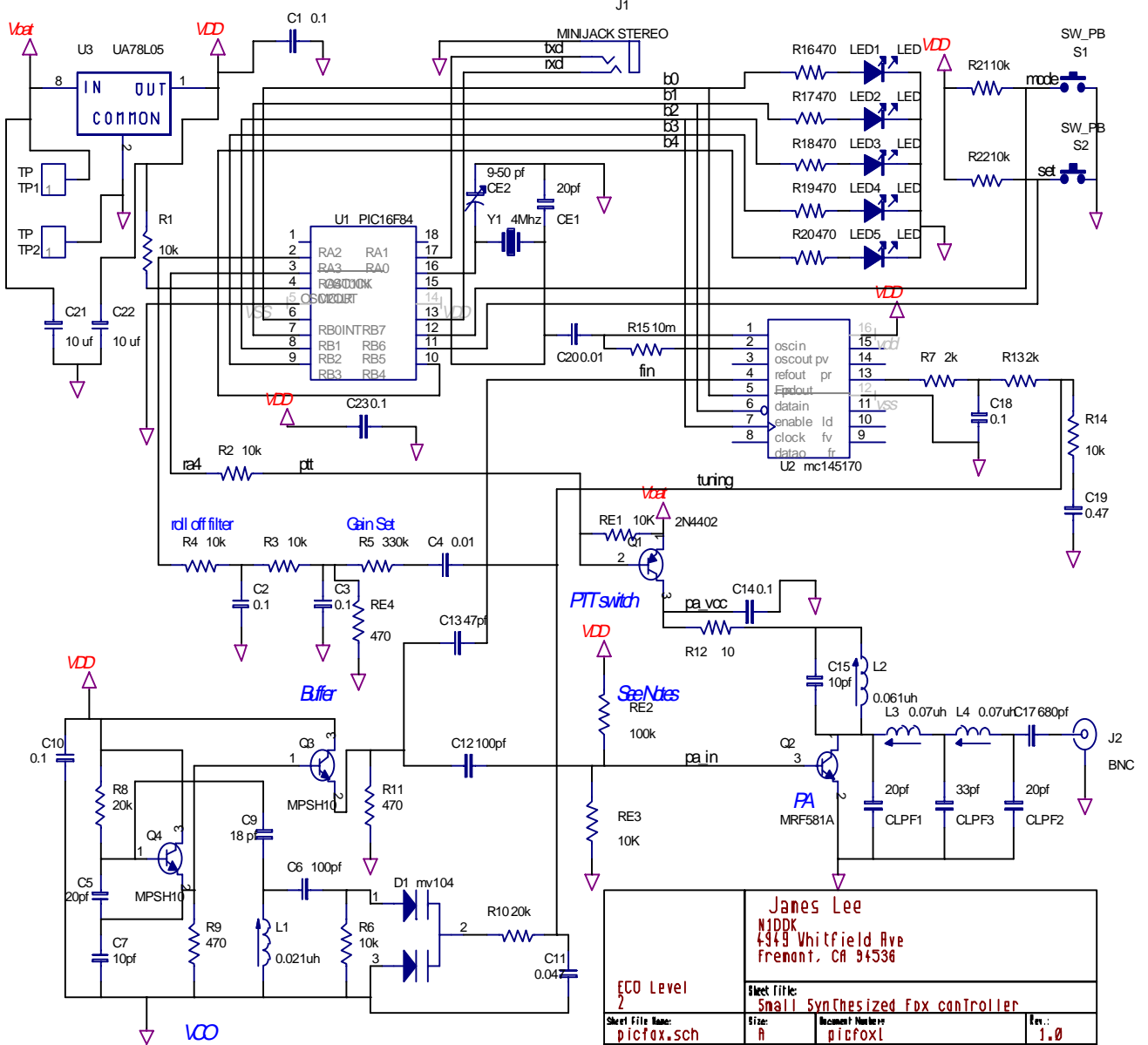
Complete Kit for "hand wound coils" \$45.

Kit minus Motorola PLL and Case \$35.

Board, preprogrammed PIC, and instructions only \$20.

QTY	Designations	Value
1	R12	10
8	R9,R11,R16,R17,R18,R19,R20,RE4	470
2	R7,R13	2k
11	R1,R2,R3,R4,R6,R14,R21,R22,RE1,RE2,RE3	10k
2	R8,R10	20k
1	R5	330k
1	R15	10m
7	C1,C2,C3,C10,C14,C18,C23	0.1
2	C4,C20	0.01
1	C19	0.47
2	C7,C15	10 pf
1	C9	18 pf
4	C5,CLPF1,CLPF2, CE1	20 pf
1	CLPF3	33pf
1	C13	47pf
2	C6,C12	100pf
1	C17	680pf
1	C11	0.047
2	C21,C22	10 uf
1	CE2	9-50 pf trim cap
1	J1	MINIJACK STEREO
1	J2	BNC
5	LED1,LED2,LED3,LED4,LED5	LED
1	Y1	4Mhz
1	D1	mv104
2	S1,S2	SW_PB
1	Q1	2N4402
2	Q3,Q4	MPSH10
1	Q2	MRF581A
1	U1	Programmed PIC16F84
1	U2	mc145170
1	U3	UA78L05
1	SOCKET1	18 PIN
1	L1	0.021uh
1	L2	0.061uh
1	L3,L4	0.07uh
1	mounting kit	
1	case	
1	pcboard	

# Schematic



<b>James Lee</b> N1DDK 4349 Whitfield Ave Fremont, CA 94536		
<b>ECO Level</b> 2	<b>Sheet title:</b> Small Synthesized Fox controller	
<b>Sheet File Name:</b> picfox.sch	<b>Size:</b> A	<b>Document Number:</b> picfox1
<b>Sheet Creation Date:</b> 11/18/98	<b>Modification Date:</b> 06/28/99	<b>Rev.:</b> 1.0
<b>SHEET 1 OF 1</b>		

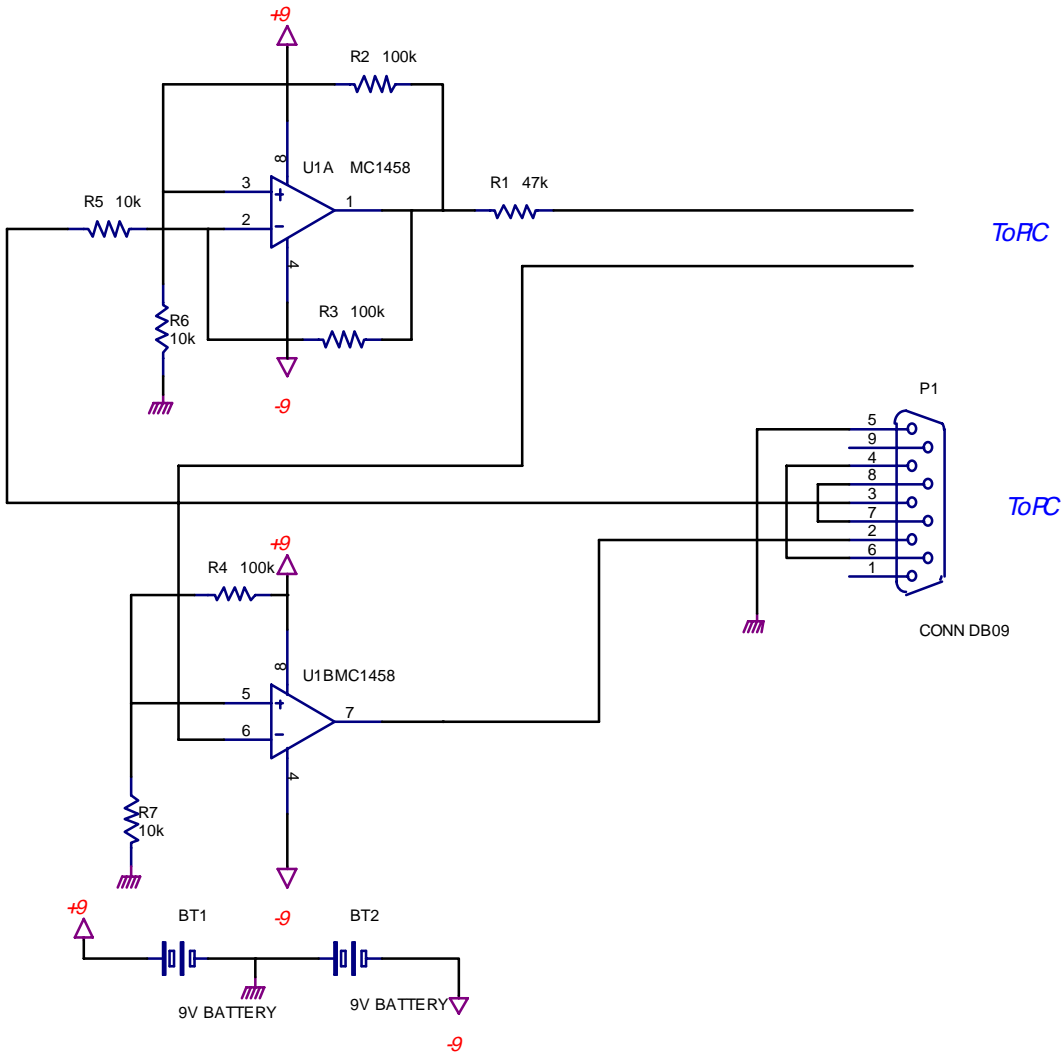
## About RS232 Level Converts

The Serial interface on the Fox controller works at TTL or CMOS levels. There are many circuits that will work to convert the levels for use with a computer. You may already have a level converter if you have an adapter to connect one of your other radios to a computer, so you may only need to create the proper cable adapter.

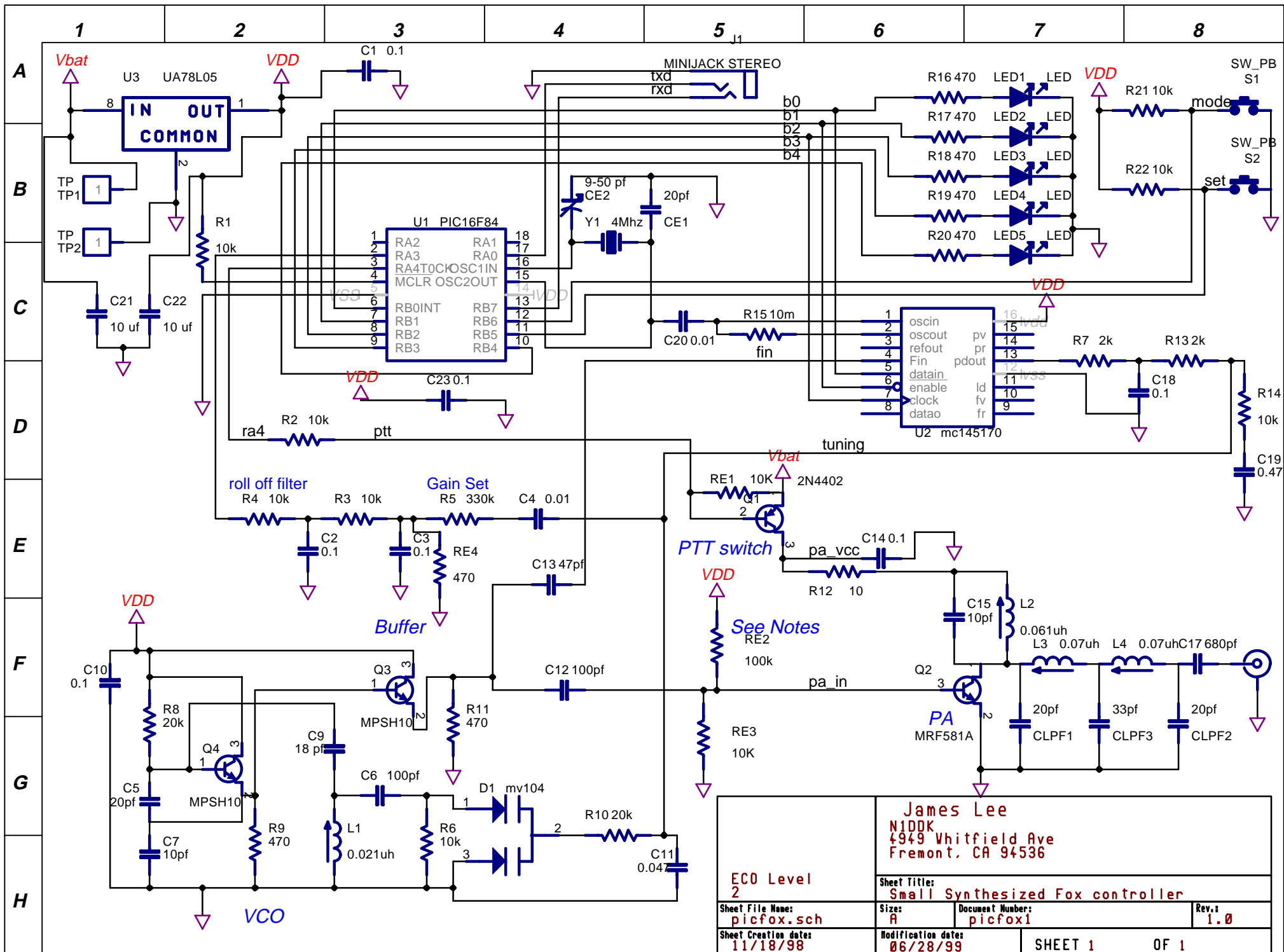
The levels used on the PIC controller is +5v for a logic 1, and 0v for a logic 0. Line Idle is +5v. The levels used in the RS232 standard are -3v to -12v for a logic 1 and +3v to +12v for a logic 0. Line Idle is -3 to -12 volts.

The Conversion can be done with a simple dual OPAMP or a Max202 or Max232 chip. A sample circuit is shown here.

### Sample RS232 Converter







<b>ECO Level</b> <b>2</b>		<b>James Lee</b> N1DDK 4949 Whitfield Ave Fremont, CA 94536	
<b>Sheet File Name:</b> picfox.sch		<b>Sheet Title:</b> Small Synthesized Fox controller	
<b>Sheet Creation date:</b> 11/18/98		<b>Size:</b> A	<b>Document Number:</b> picfox1
<b>Modification date:</b> 06/28/99		<b>Rev.:</b> 1.0	
<b>SHEET 1</b>		<b>OF 1</b>	

