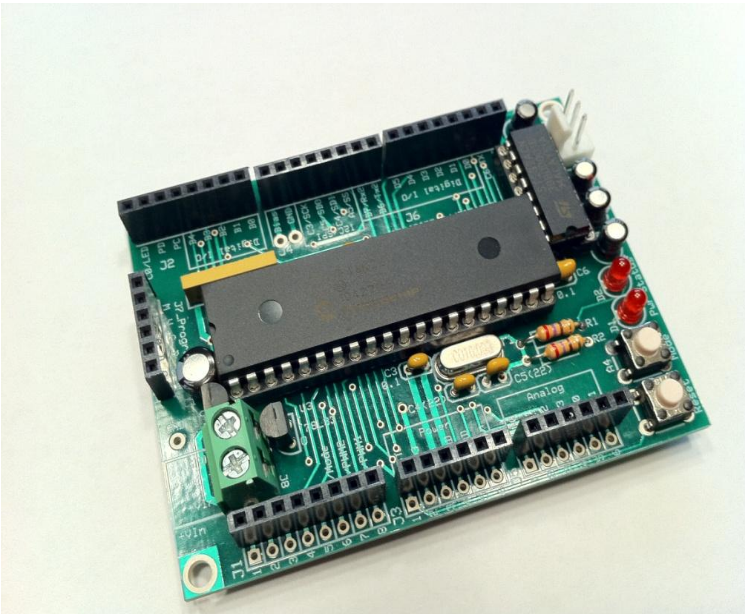
Making a Keyer with the

HamStack

Project Platform



Introduction

The HamStack Project Board includes primary interface elements needed to make a high quality CW keyer. Using the LCD display and a rotary encoder, it is fairly easy to implement a very nice interface to adjust the parameters of the keyer. These same concepts can be used for the user interface for any number of other projects.

This example is programmed in C, using the Microchip MPLAB integrated development environment and C18 compiler. The Hamstack C function library contains most of the low level elements needed to implement the keyer, including LCD support and interrupt driven timers, tone generation, keyer implementation in encoder support.

Refer to the HamStack Introduction and HamStack C Programming manuals for more information.

A precompiled hex file with the keyer program is included for those that just want to try out the keyer as is.

Setting up the Hardware

The prototype version of the keyer is shown below.

The following items are needed to complete the project:

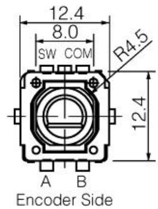
- HamStack CPU board
- HamStack project board
- 2 line x 16 character LCD module with interface board and cable
- Panasonic EVE-JBBF2020B rotary encoder (Digikey P12336-ND)
- H11D1 opto-isolator, or Vishay LH1540AT solid state relay (Mouser 782-LH1540AT)
- 0.5" diameter, 6mm shaft knob (Digikey 226-4128-ND)

If you want to enclose the keyer in one of the Hammond plastic boxes which the project board is designed to mount into, then you should ***not*** install the screw terminals on the project board. There is not enough room for the LCD module to be mounted in the front panel of the box if the screw terminals are in place. Even with the terminals removed, there is barely enough room for the LCD board above the project board – the hole in the panel must be cut so that the top of the LCD circuit board aligns with the top of the panel. For our prototype, we cut custom black front and back panels. It is also possible to carefully cut openings for the LCD, encoder and back connectors in the black panel supplied with the enclosure. The LCD board was attached to threaded nylon spacers that were subsequently super-glued to the back of the front panel.

Either of the opto-isolators will work. There are two sockets on the project board for the opto-isolator. The one closest to the output connector is wired for the solid state relay; the other is for the transistor switch opto-isolator. An H11D1 is included with the project board. It works with only one polarity. If you are keying a modern, solid state transmitter, then the center pin is the keyer pin, to be connected to the tip of the plug going into the transmitter.   If you are keying a negative grid block keyed tube transmitter, then the polarity must be reversed – connect the outside of the connector to the key line. The solid state relay is polarity independent and can be wired the same way for either +5V or -100+ V keying. When wiring the key output to the transmitter, do not connect the project board ground to the transmitter ground. We have had problems with both wall wart power supply noise getting into the receiver, and transmitter RF messing up the keyer when that was done. With the connection only through the opto-isolator outputs, there have been no problems.

The encoder wiring diagram is shown to the right. The wiring is as follows:

- One of the COM pins is connected to ground (far left screw terminal on the project board)
- SW pin to A4
- A pin to E2
- B pin to E1



Pin A is always off at the detented steady point. Pin B transitions at or near the detented steady point. You should check this with an ohmmeter. “Encoder Side” in the diagram from the datasheet means the view from the shaft end.

The key is connected with the dot switch connected to the tip of the 3.5 mm stereo plug. Left handed operation is accommodated in the software.

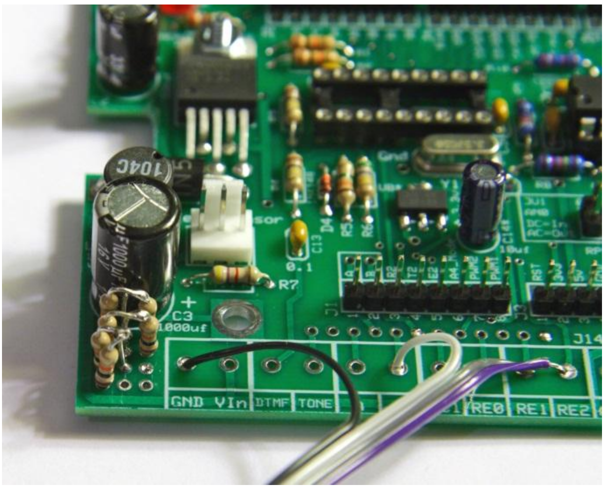
Project Board Modifications

There are a few modifications to the project board that are needed.

1. The switching power regulator on the board will generate significant noise if it is not properly bypassed. In rev. 1.0 of the board, two 0.1 uF monolithic ceramic capacitor should be soldered directly to the pins of the regulator, so that they are in parallel with the large electrolytics that are already in place.
2. The encoder A and B switches will not work without pull-up resistors from the processor input pin to 5 V. There is a tiny prototyping area in the corner of the board with room for a few vertically mounted 10k Ohm 1/4 watt resistors. Jumpers can connect these to regulated 5V and to the E1 and E2 processor pins. The encoder push button switch does not need an additional pull-up because it is wired in parallel with the CPU board “mode” switch which already has a pull-up.
3. A jumper is wired from the A4/mode pin to the B input (6^{th} from the left along the screw terminal connections) on the project board. This provides a connection on the project board edge for the push to select switch on the encoder.

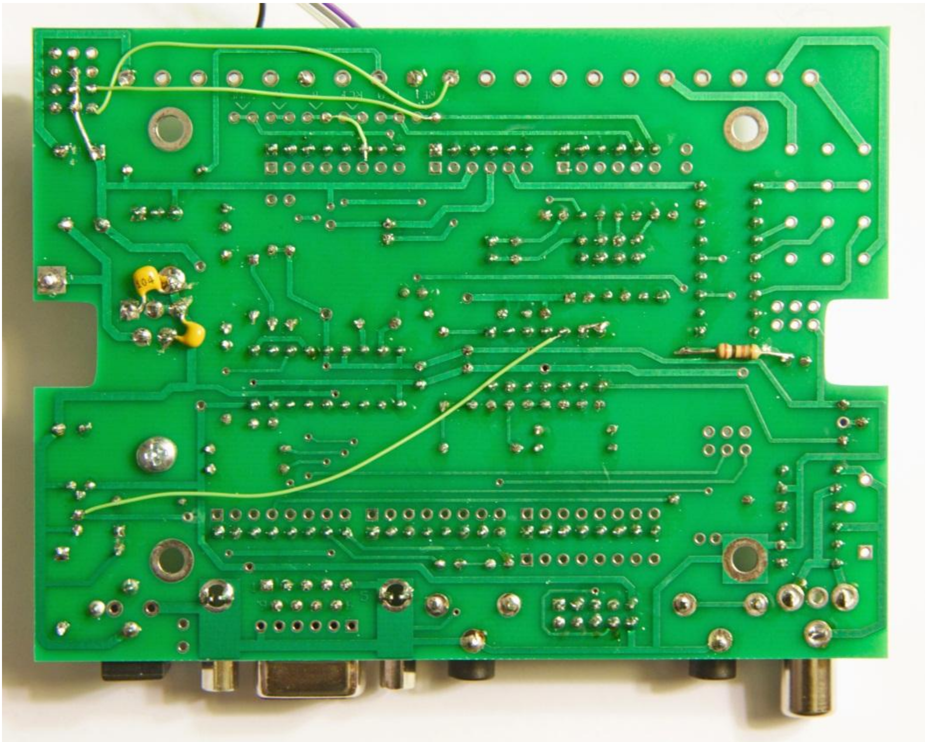
4. The op amp output of the tone generation filter cannot drive loads of less than 100 Ohms without severe distortion unless the level is very low. The tone generator runs at a full duty cycle swing, generating nearly 5V p-p tone amplitude in order to minimize quantization distortion from the 8 bit tone generation algorithm that is used. You can cut a trace from the output potentiometer and insert a 100 Ohm resistor to solve this problem. If you do not drive low impedance or a long cable with lots of capacitance, this may not be necessary.
5. One of the op amps in the MCP6004 quad op amp package is not used. The microchip datasheet states that unused op amps should have their inputs tied either to the supply rails or configured as a unity gain amplifier with the input ties to a bias voltage at ½ of the supply voltage. There is a 2.5V reference regulator on the board. The op amp should be wired to satisfy this condition unless you are wiring it to be used for something else. We did not detect any significant noise problem before this is done, so you might get away with ignoring this advice.
6. The location marked C11 on the project board should be populated with a jumper wire, not a 0.1 uF capacitor. This is necessary to set the correct DC bias level inside the tone generator filter.

The photo below shows the 10k Ohm resistors added for digital input pull-ups.



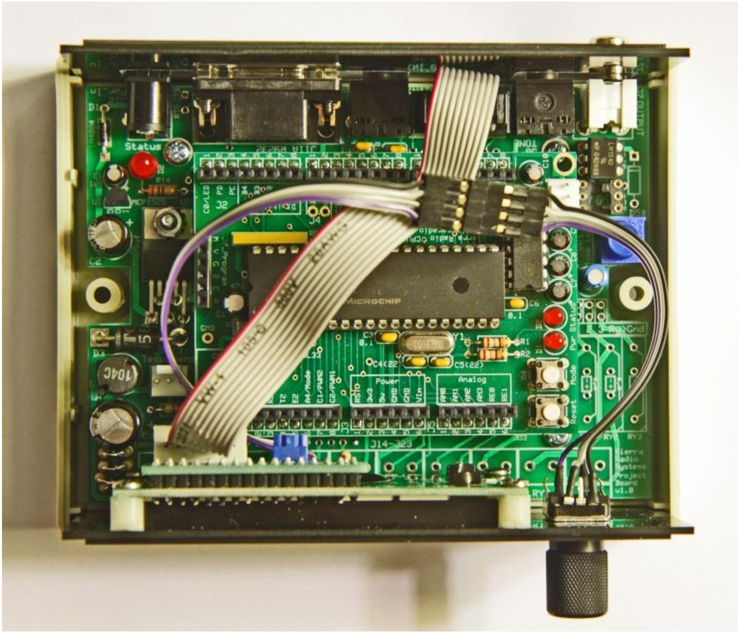
The photo below shows the bottom of the project board, with all of the modifications added. Note that the project board screw terminals were not installed. This necessary to allow room for the LCD panel tp project through a hole in the project board case front panel.

Project board bottom view, with modifications



- The top left is the area with the pull-up resistors, with wires added to +5V (uninsulated), and to the E1 and E2 pins. There is also a jumper from the project board A connector to the CPU board A4 pin.
- The two capacitors are the bypass capacitors at the switching regulator. Note the very short leads.
- The long jumper from the lower left to the middle of the board connects between the 2.5V reference output and the +input of the spare op amp (pin 5). There is also a short jumper between pins 6 and 7.
- The 100 Ohm resistor jumpers across a gap cut in the trace at the output of the tone volume potentiometer.

Additional Packaging Photos



Operation

On power up, the prototype code shows the CW speed on the bottom line of the display. It can be changed by rotating the encoder knob. You can adjust it while actively sending.

By pushing on the encoder button, you can change the display and encoder operation to set other keyer parameters. In the prototype code, successive presses of the button will cycle through setting speed, keyer mode, left handed operation, keyer weight, sidetone frequency, minimum character WPM (Farnsworth spacing), and back to speed. Rotate the encoder to change these parameters. If you do nothing for a few seconds, the display will revert back to showing the CW speed. Changes in parameters are automatically saved to on-chip data EEPROM, so that the last settings used are restored on power up.

The keyer modes are simple keyer, iambic a, iambic b, and iambic b2. Iambic b inserts an extra element if both paddles are pressed while the dot or dash is active. Iambic b2 is between iambic a and iambic b, recognizing the extra element only during the gap after the element.

More elaborate code is planned to capture, send, and display memory contents, to adjust the keying weight (in milliseconds additional relative dot and dash length, not percent, which is mostly useless), and maybe eventually to read incoming CW.

Software

Hex files compiled for the 18F4620 with a 10 MHz crystal and the 18F46K22 with a 16 MHz crystal are provided. Use any PIC programmer to load those into your CPU.

For other situations, or to modify the code to suit your own taste, you must edit and compile the code yourself.

You will need to download and install MPLab and the C18 compiler from the Microchip web site in order to edit and compile the program yourself.

Download and unzip the keyer_prototype.zip file from the Hamstack web site. Unzip it to a location of your choice.

Start MPLAB, go to File/Open Workspace, navigate to the directory where you put the keyer_project files, and open keyer.mcw.

- Click on keyer_example.c in the project window to edit the main keyer program.
- The main keyer code is in the file keyerint.c in the hamstacklib directory. Be careful if you edit this. Interrupt code can be difficult to debug.
- If you just need to change clock rate click on the user_options.h file and edit the CLOCKFREQ entry.

Recompile and program your controller to test your new version.

Details on using MPLab and C18 to do your own projects are available in the HamStack C Programming Manual.