INTRODUCTION

This document details how to duplicate an Arduino based electronic keyer that I developed in June 2014 since I wasn't happy with a commercial unit I was using. This unit has the following features.

1. Memory buttons on top of the case to avoid slipping on the operating desk.
2. The memory is not lost when the power is turned off.
3. A keyboard interface that permits keyboard CW and also a means to temporarily change the memory.
4. And thumb screws to open the case instead of screws to replace the battery.

I did borrow some code to get started from others. I chose the Arduino microcontroller because it was available at our local Radio Shack stores. Arduino's are used heavily in robotics and rocketry and thousands of completed various projects are available on Make.com. Furthermore, the development software is free.

If you are not familiar with Arduino, I highly recommend the book "Getting started with Arduino" with sells are Radio Shack for $15. Try some of the simple examples provided such as blinking an LED or reading a switch status.

On this disc are six files.

1. This "READ_ME_FIRST1.DOC" file.
2. The source code for the keyer "W9KTP_Keyer_With_Memory.INO".
3. The program listing "W9KTP_Keyer_Listing.Doc".
4. The overall circuit design "Arduino_Keyer_Main_Circuit.jpg".
5. The interface board circuit "Arduino_Keyer_Interface.jpg".
6. Lastly an include file for the keyboard "PS2Keyboard.h."

SOFTWARE

This README cannot teach you how to program in the 'C' language which is used by the Arduino. My objective is walk you the process of compiling the program and downloading it to the Arduino through a USB cable.

The first thing you will need is copy of the development system (DS) which is free online at http://www.arduino.cc/en/main/software. Once you have downloaded and installed the DS, you should have Desk Top icon on your main page for Arduino which an infinity symbol inside a circle.

Generally you will need to transfer the source code (referred to as a sketch in Arduino lingo) to the "examples" subfile under the Arduino directory. As an alternative, you can copy/paste from the file on the disc directly into the DS.

Before you try to compile (verify) the program, you will need to move the keyboard include file, PS2Keyboard.h into the "library" subfile in the Arduino directory. Also, there are two steps to installing the DS on your system, installing drivers and

setting the serial port number. Read the instructions on the Web Site under the ‘Learning” tab for your system.

Now that everything should be in place, click on the Desktop Icon for Arduino and bring up the DS. Click on FILE and then OPEN. Click on the keyer source code and it should load into the DS. In the upper left you should see an icon with a check mark in it. Click the icon. After a few seconds you should be rewarded with a message in the lower left announcing “Binary Sketch size xxxx bytes” indicating a clean compile. Since there are several things that could go wrong, you might get error messages instead of the Sketch size. My best advice is to review/repeat the earlier steps.

Assuming you got a clean compile, it’s time to hook up the Arduino with a USB cable and download the code to the board. As soon as both ends of the cable are connected, the lights on the board will blink. If it doesn’t, you may have to check your serial port settings.

Once the board is communicating with the computer, click on the ICON in the upper left with the arrow in the middle and shortly you will be message at the bottom of the screen that the sketch has been downloaded. Most likely, you will want to change the messages to include your call and possibly change some other things. Take a look at the Listing in the Arduino development system. About two pages down you’ll find the messages like char msg1[25] = {‘W’,’9’,’K’,’T’,’P’,’\0’}; You only want to or should change what is between the braces unless ou know what you are doing.

Notice that the call letters and other messages are in all caps and surrounded by single quotes. You need commas between them. To end the call or message you need to insert what is called a NULL Character. That’s what the back slash, zero combo is called. For a Beginner this is where many compile errors come from. Miss a single quote or a comma and you’ll get an error. Follow the instructions above EXACTLY. Re-compile and download the software with your changes to your Arduino. If you have put together the hardware you are ready to go.

HARDWARE

I used the Arduino Uno for my keyer which is a common model. If for some reason you chose a different model, you may have to change the pin assignments. The interface board is built on a protoboard which is available at Radio Shack. I stuck the switch for the battery/AC adapter selection on the boards as well. There is nothing special on wiring the circuit. I did use headers to connect the wires to the Arduino I/O sockets. Simply shoving wires into the sockets is only good for temporary service. Headers are just strips of plastic with pins through them at the same spacing as the Arduino sockets. Solder the wires to the top of the header and push the bottom pins into the sockets. Much more solid connection. I did label each wire with a small piece of tape

so if a wire came lose, I could fix it quicker. Depending on your transceiver, you may need to put a ferrite bead or two on the lead to the key.

KEYBOARD

Implementing the full PS/2 keyboard is a little more complicated than I expected. There's another chunk of software needed to use all the possible key combinations and I was unable to get it to compile correctly. It would have been better to use 'Function' keys instead of the ones I chose. I'll give it another try some time in the future. Since there is not a screen to indicate when a key is pushed, I used beeps to let the user know a key has been pushed. The keyboard functions are as follows.

PGUP – Enter Keyboard Mode (Dit, Dah Sound) –Send Morse w/KYBD

PGDN – Exit Keyboard Mode (Dah, Dit Sound) – Stop sending Morse w/KYBD

CHANGING STORED MESSAGES (one at a time!);

Enter Keyboard Mode

For message 1 Up Arrow (2 Dits), Type new message,

For message 2 Left Arrow (4 Dits), Type new message,

For message 3 Right Arrow (6 Dits), Type new message.

Exit Keyboard Mode

SOME FINAL NOTES

The listing is a M/S Word document allowing landscape printing. The "Print" function in the DS is very limited. I considered adding more illustrations and maybe some examples but I would have to guess which ones to include. If you have questions, sent me an Email at jerryhess@frontier.com.