Mortty – Morse Code and RTTY Keyer

Mortty is a Do-It-Yourself construction project that provides a miniature enclosure and a computing platform for K0SM's TinyFSK RTTY Keyer software or K3NG's CW Keyer software.

Unlike the full-featured - and more expensive - hardware solutions for CW and RTTY keying that are widely available to amateurs, Mortty is an inexpensive minimalist solution that depends upon the configuration and control features incorporated in popular logging programs. This tiny box, measuring only 2 inches long and 1 inch square (50mm by 25mm), has only two inputs (logging program, CW paddle) and two outputs (PTT, CW/FSK).

Designed by N8AR and K8UT, Mortty is available as a complete parts kit, or you can use this guide to build your own from "scratch."

Table of Contents

Mortty Overview

Key Features

- Inexpensive, off-the-shelf, readily available components (except for the Mortty board itself)
- Small, portable, USB-powered
- Rugged RFI-resistant metal enclosure
- CW Keyer emulates the widely supported Winkeyer protocol
- RTTY FSK keying via widely supported TinyFSK protocol
- Construction requires only common household tools

Setting Expectations

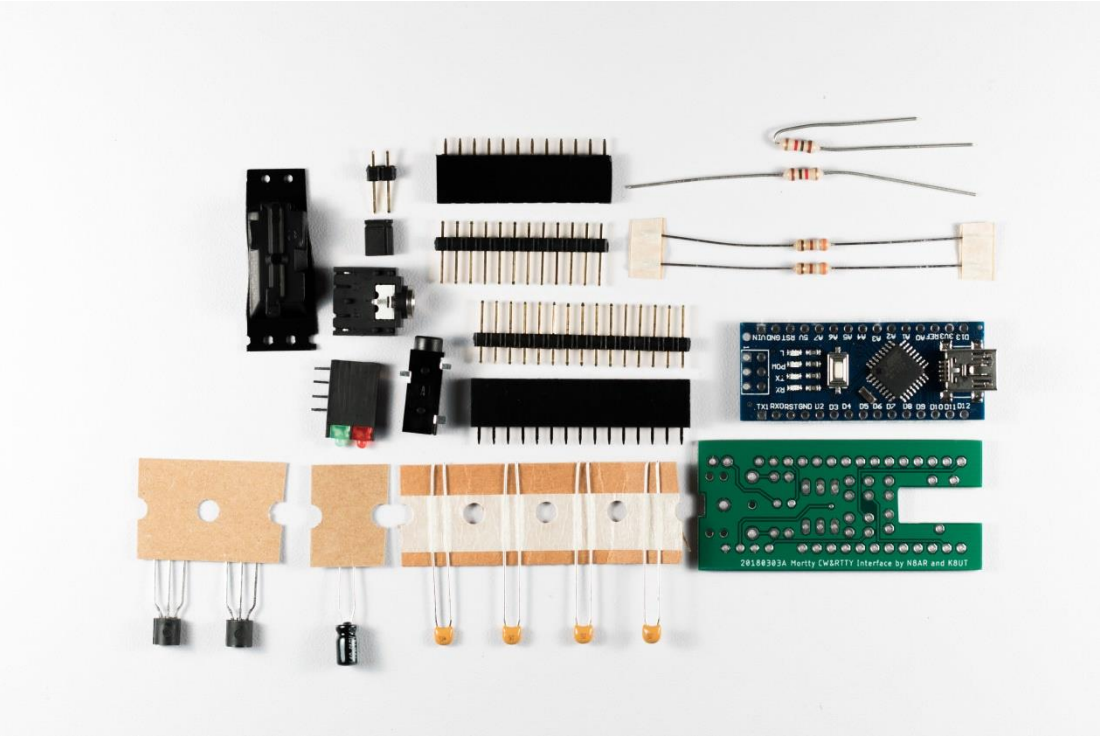
- **What's missing from this miniature design?**
  - For CW – No speed potentiometer, no macro pushbuttons, no monitor speaker
  - For CW – SO2R operation by re-purposing the ring connector (normally PTT) to CW Key rig 2
- **CW or FSK – but not both?**
  - Mortty can operate either CW or FSK
  - Switching between modes involves: opening the enclosure; removing the Mortty board program/operate jumper; uploading the desired sketch (CW or FSK); re-installing the program/operate jumper
    - Elapsed time for switching modes - about five minutes
    - Maybe you should build two Morttys? ;-)
- **Degree of mechanical difficulty – low (if you use the 3D printed plastic end caps)**
  - The Mortty board fits very tight in the metal enclosure. You may need a file or sandpaper to adjust the Mortty board for a snug fit in the case
  - *If you decide to use the metal end caps*, you will need to make five holes in the enclosure
    - Use this documentation or the plastic end caps as a drill guide
    - Drill four round holes for paddle input, keyed output, and two LEDs
    - Drill / notch / file one oblong hole for the Arduino Nano USB connector
- **Degree of soldering difficulty – medium**
  - To achieve Mortty's miniaturization, the components are tightly packed on the circuit board
  - There is only one surface mount part, the paddle jack. Its solder pads are as large as standard circuit board components and will be no more difficult than soldering the other components
  - You will need some type of circuit board holder and parts stabilizer to assist in soldering parts to the small Mortty board (small vice, tweezers, hemostats?)
- **Degree of computer expertise – low**
  - No programming required
  - The are many excellent resources on the Internet to assist you in loading the proper drivers for the Arduino Nano and uploading the desired sketch onto the board
  - If you indicate your desired sketch (TinyFSK or CW Keyer) on your PayPal order, we will preload the Arduino Nano for you

Required Materials

- Mortty project parts (see Appendix 1 - parts list and pcb layout)
- Tools
    - Low wattage (25 – 40 watt) solder pencil with small tip and solder
    - Phillips screwdriver, small side cutters, pliers
    - *If using metal end caps* – a drill with 1/4 inch, 7/64 inch, 5/64 inch bits, a small file
- Software sketch(es)
    - For FSK RTTY - download from K0SM TinyFSK website
    - For CW Keyer - download from K3NG CW Keyer website
    - Integrated Development Environment (IDE) from Arduino website
- Operation
    - Cable and connectors to interface Mortty output (3.5mm stereo jack) to your radio's PTT and CW or FSK inputs
    - Logging software compatible with K1EL Winkeyer (CW) or TinyFSK (RTTY)
    - OPTIONAL: For CW operation - paddle plugged into Mortty key input

Release Notes

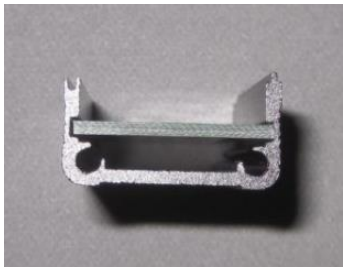
1.0 2018-04-02 Initial release



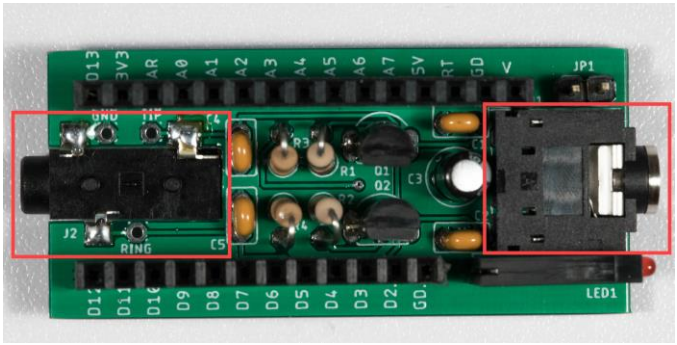
Step 1. Assembling the Mortty Circuit Board

Open the Mortty parts envelope

☐ Locate the Mortty board and gently try to slide the board into the guide slots of the metal enclosure. It might not fit. The board is sized to create a very tight fit – perhaps too tight in some of the enclosures. Use sandpaper or a file to evenly trim both sides of the board until it fits snugly in the metal enclosure.

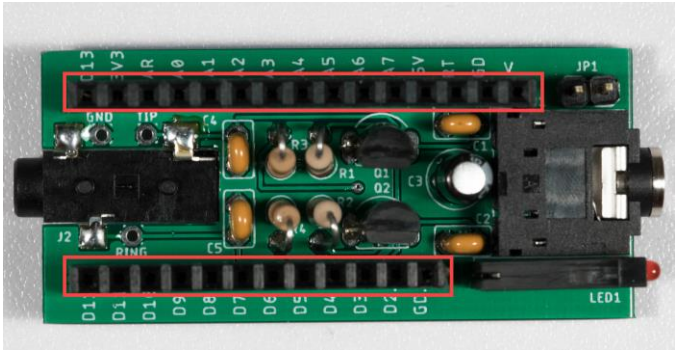


☐ Locate the 3.5mm surface mount jack. Insert the surface mount jack from the TOP (labeled) side of the Mortty board in the large notch indicated as J2. Press jack J2 against the three pads labeled GND, TIP, RING and apply solder **on the top of the board**. NOTE: This is the only component that will be soldered from the top.



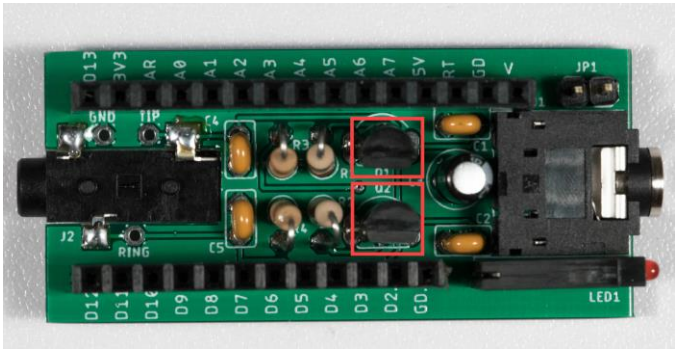
☐ Insert the other 3.5mm jack from the TOP (labeled) side of the Mortty board in the location marked J1. Hold jack J1 firmly against the Mortty board and apply solder to the bottom of the board.

SOLDERING HINT: The following connectors need to be perfectly vertical. **Begin by soldering only one pin on a connector**, then stop and examine/reheat/adjust that one pin for vertical alignment. When satisfied that the connector is positioned properly, solder the remaining pins. Avoid applying excessive solder as it will make it difficult to trim the pins as directed later in this procedure.



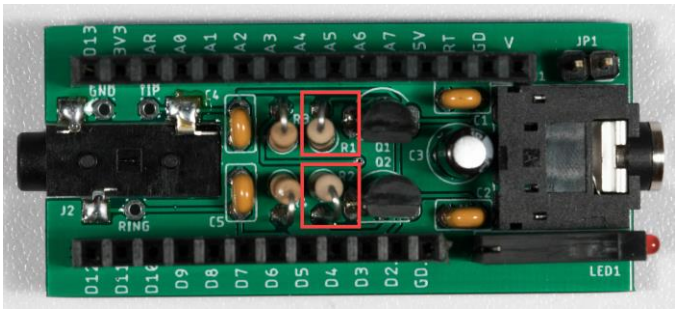
☐ Locate the 15 pin and 12 pin female Single-In-Line (SIL) connectors. Insert these connectors from the TOP (labeled) side of the board in the locations indicated in the photo. Apply solder to the bottom of the board.

☐ Locate the two NPN transistors and insert them from the TOP (labeled) side of the board in positions Q1 and Q2. Align the flat side of the transistors with the flat side in the outline on the Mortty board. Press them far enough onto the board so that the tops of the transistors are slightly lower than the height of the female SIP connectors. Apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.

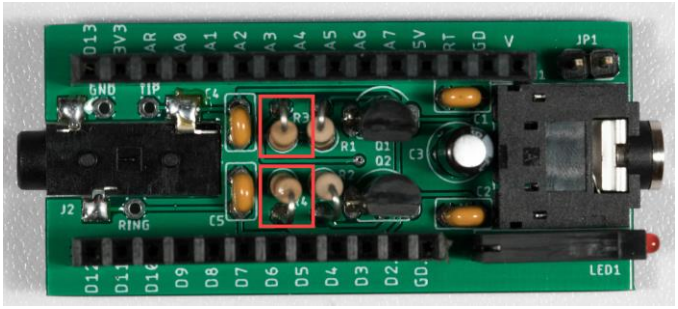


IMPORTANT! Due to space constraints on the Mortty board, resistors are installed vertically on the board.

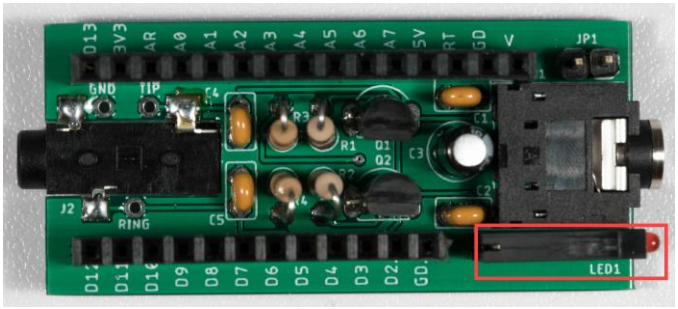
☐ Locate the two 1000 ohm (brown black red) resistors. Fold one lead over so that the leads are parallel to each other. Insert the 1000 ohm resistors on the TOP (labeled) side of the Mortty board in positions R1 and R2. Hold each resistor upright so that one end of the resistor is flush against the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



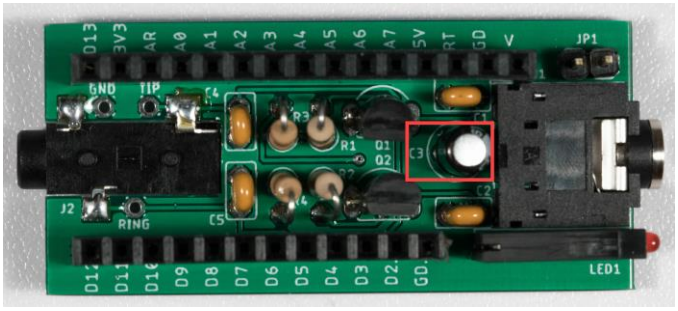
☐ Locate the two 390 ohm (orange white brown) resistors. Fold one lead over so that the leads are parallel to each to each other. Insert the 390 ohm resistors on the TOP (labeled) side of the Mortty board in positions R3 and R4. Hold each resistor upright so that one end of the resistor is flush against the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



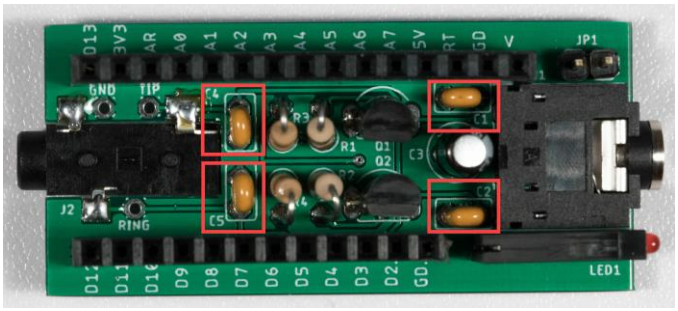
☐ Locate the dual red/green LED carrier. Insert the carrier on the TOP (labeled) side of the Mortty board in the position marked LED1. Keeping the carrier perfectly vertical, hold the LED carrier firmly against the Mortty board and apply solder to the bottom of the board.



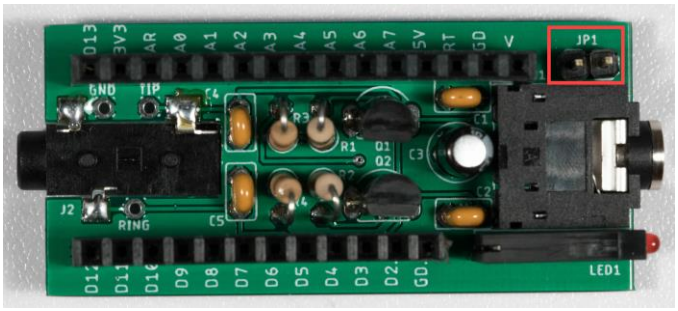
☐ Locate the 10 uF electrolytic capacitor. Insert the capacitor on the TOP (labeled) side of the Mortty board in the location marked C3. Ensure proper capacitor polarity by aligning the positive lead of the capacitor with the + indicator on the Mortty board. The stripe down the side of the capacitor identifies the negative lead. Hold the capacitor so that the bottom sits flush against the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



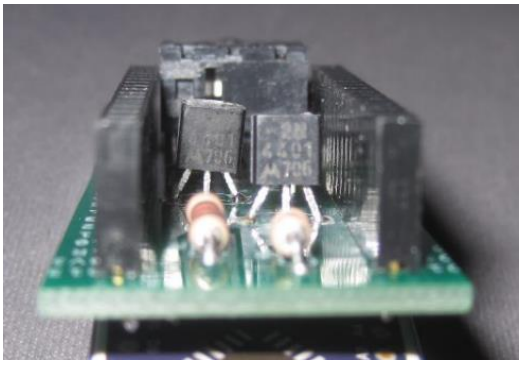
☐ Locate the four .001 uF ceramic capacitors. Insert the capacitors on the TOP (labeled) side of the Mortty board in positions C1, C2, C4 and C5. Hold each capacitor upright on the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



☐ Locate the 2 pin male Single-In-Line (SIL) connector. Insert the short pins on the TOP (labeled) side of the Mortty board in the location marked JP1. Hold the connector firmly against the Mortty board and apply solder to the bottom of the board.



This completes the assembly of the Mortty board. Examine the board to ensure that all of the resistors, capacitors and transistors are shorter than the two Single-In-Line (SIL) connectors on the sides of the board.



Step 2. Assembling the Arduino Nano

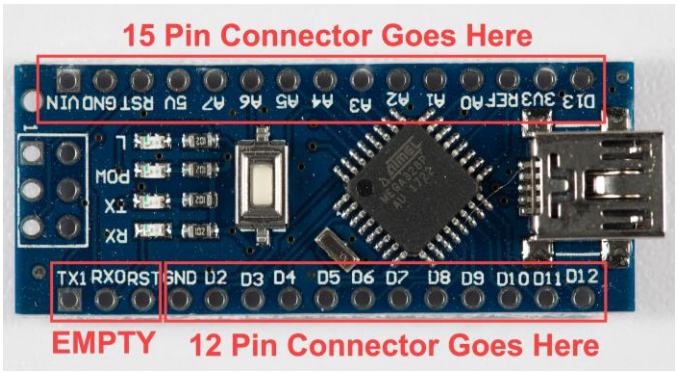
IMPORTANT! Due to space constraints in the miniature enclosure, the Arduino connectors are installed on the TOP of the Arduino circuit board, not the bottom. One of the connectors is shortened to only 12 pins. Do not solder any connectors to the Arduino Nano until examining the photos and carefully reading the following instructions.

Open the Arduino Nano package

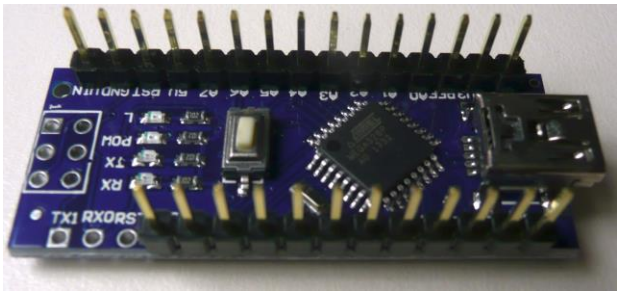
- ☐ **If your Arduino package** contains this 6 pin male Dual-In-Line connector, throw it away. **Do Not Install** this connector on the Arduino.

- ☐ **If your Arduino package** contains two 15 pin male Single-In-Line (SIL) connectors, shorten one connector to contain 12 pins. With a pair of needle-nose pliers, grasp the last three pins and carefully snap them from the end of one of the connectors.

SOLDERING HINT: The following connectors need to be perfectly vertical. **Begin by soldering only one pin on a connector**, then stop and examine/reheat/adjust that one pin for vertical alignment. When satisfied that the connector is positioned properly, solder the remaining pins. Avoid applying excessive solder as it will make it difficult to trim the pins as directed later in this procedure.

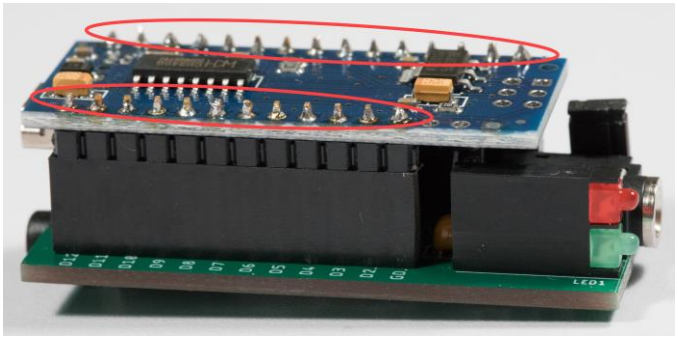


- ☐ Refer to the photo on the right. Insert the short pins of the 12 pin SIL connector from the TOP (labeled) side of the board, positioned so that the holes on the Arduino board labeled TX1, RX0, and RST remain empty. Hold the connector perfectly vertical and apply solder to the bottom of the board.



- ☐ Insert the short pins of the 15 pin SIL connector from the TOP (labeled) side of the board. Hold the connector perfectly vertical and apply solder to the bottom of the board.

☐ Inspect the bottom (solder side) of the Arduino Nano. Using side cutters, trim any excess lead length from the 12 and 15 pin connectors by clipping them as close to your solder connections as possible.



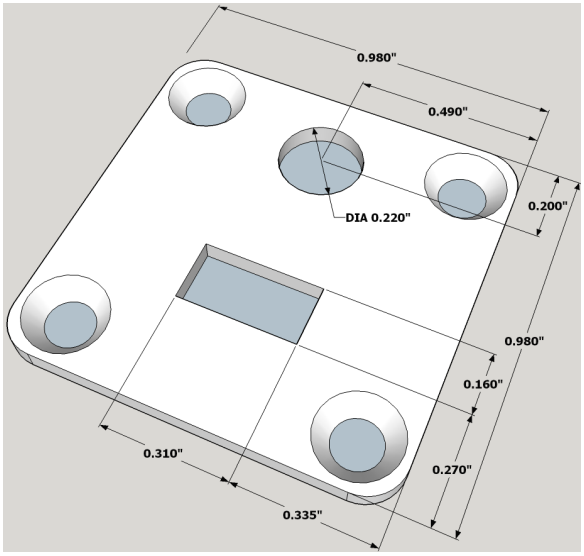
This completes assembly of the Arduino Nano. Set it aside for now.

Step 3. Cutting Holes in the Enclosure End Caps

The Mortty kit ships with 3D printed plastic end caps that align with the connectors and LEDs on the printed circuit board. The enclosure's original metal end caps are also included. The following steps are required only if you decide to use the metal end caps. It is probably easiest to use the plastic end caps as drilling templates.

The Mortty enclosure requires five holes for the USB serial port, CW paddle, keyed output and two LEDs. Designate one of the metal end caps as **Input** and the other as **Output**.

☐ See Appendix 3: End Cap Drilling Templates. Drill a ¼ inch (0.25) diameter round hole for the CW paddle jack in the **Input end cap**. Drill a small 7/64 inch (.11) pilot hole and then nibble/file an oblong 0.25 inch by 0.11 inch hole for the Arduino Nano USB port.

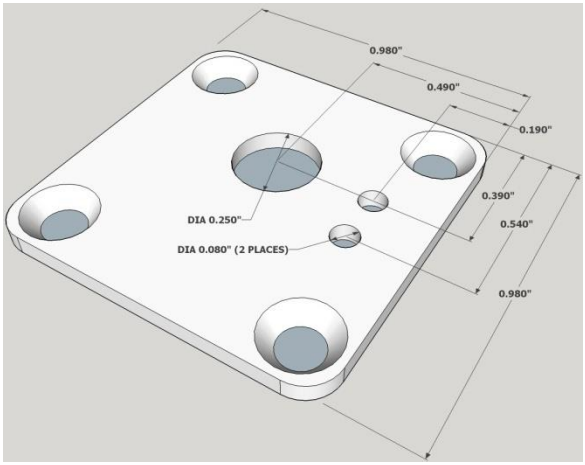


Input End Cap

- USB connector from computer
- 3.5mm (1/8") jack from CW paddle
  - Tip = Dash paddle
  - Ring = Dot paddle
  - Sleeve = Ground

3.5mm 3-Conductor (TRS) Connector

☐ See Appendix 3: End Cap Drilling Templates. Drill a ¼ inch (0.25) diameter round hole for the keyed output jack in the **Output end cap**. Drill two 5/64 inch (0.08) diameter round holes for the status LEDs.



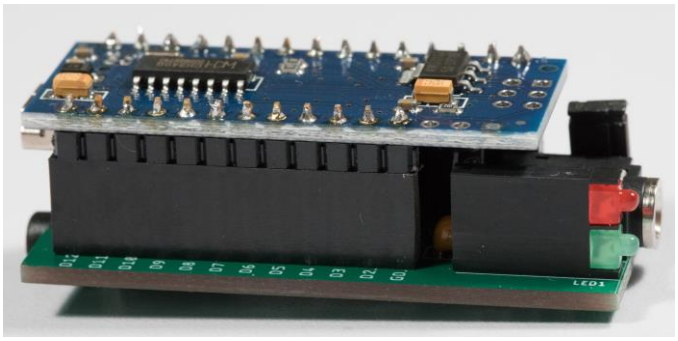
Output End Cap

- 3.5mm (1/8") jack to radio
  - Tip = CW or FSK Key output
  - Ring = Push To Talk (PTT) or *SO2R CW Key radio 2
  - Sleeve = Ground
- Status LEDs
  - Red = Push To Talk (PTT) or *SO2R CW Key radio 2
  - Green = CW or FSK key

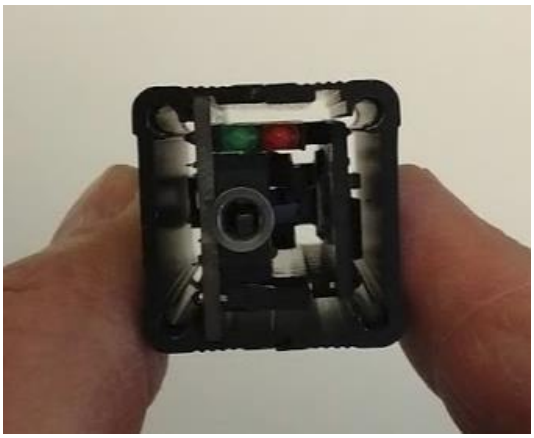
Step 4. Final Assembly

Be patient when performing the final assembly! The tiny end-cap screws are easy to cross-thread. The tolerances for placement of holes for USB connector, paddle, and Rig are very snug - an attempt to keep the Arduino and Mortty boards from shifting inside the case and shorting to ground. Take your time…

☐ Align the 15 pin and 12 pin connectors of the Arduino Nano with the corresponding pins on the Mortty board. Fully insert the SIL pins of the Arduino Nano into the corresponding sockets on the Mortty board. Check to ensure that all pins of the Nano are contained in matching sockets on the Mortty board.

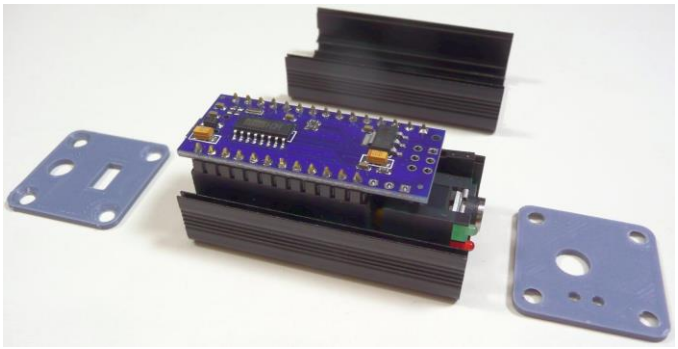


☐ Slide the Mortty board into the guides of the bottom of the enclosure. Place the cover on the enclosure. Stare through the case and examine the gaps between the enclosure and the adjoining boards. They will be very close, but the leads from the Mortty board and the Arduino Nano must not touch the case. If there is any contact with the case, use side cutters to trim the offending connection.



ASSEMBLY HINT: After trimming the leads of the connectors, if you still think they are too close to the case, use a small file or a fingernail emery board to *gently* file away the connector tips.

☐ Secure the cover by attaching the Input end cap (the end with the USB port) using four Philipps head screws. Temporarily set the Output end cap aside to provide access to JP1 during the sketch (software) upload.



TIGHT FIT? You may find the holes in the 3D printed end cap are too small. Use a 3/32 or 5/64 drill bit to enlarge the LED holes, 1/4” for the round connector holes.

Step 5. Installing Software

There are many on-line guides to assist in preparing the Arduino Nano for CW or RTTY operation. We've listed a few sources for you, or you can use Google to look for others.

WINDOWS UPDATE = OFF? Dave K6LL encountered problems because Windows did not automatically assign a USB com port to the Arduino Nano. He intentionally operates his PC with Updates turned off, and his computer had not received the Update that would have enabled the CH340 driver. There are good reasons to disable Updates on your computer, but in doing so you may need to temporarily enable Windows Updates or manually install the driver yourself.

Loading the CH340G Drivers into Windows

The Arduino Nano used by Mortty use a CH340G UART that requires a Widows driver for the USB interface. Upon initially connecting your Nano to your PC, Windows will attempt to install the correct driver. If you do not receive an error message, you're good-to-go. If you receive an error message, you may need to download/install the CH340G driver. If you are not sure if the driver installed correctly, open the Windows Device Manager and examine the **Ports (COM & LPT)** section for the new serial port. If the device is not listed, refer to these websites for instructions:

The latest version of the driver is CH341. Download the `CH341_SER.EXE` driver here:

http://www.wch.cn/download/CH341SER_EXE.html

Look for installation/configuration information here: (or just google "CH341G driver")

https://forum.arduino.cc/index.php?topic=397368.0
https://sparks.gogo.co.nz/ch340.html
https://www.youtube.com/watch?v=4YkXXNcNzh0

Is your CW Keyer or TinyFSK Sketch Already Installed?

Normally, the Mortty developers load a sketch on the Nano prior to shipping. By default, the CW Keyer sketch is installed, but the TinyFSK sketch may have been installed if you indicated a preference for RTTY.

- If the red LED is blinking slowly, then the Arduino is executing its default "Blink" sketch and you will need to use the Arduino IDE (see below) to upload either the CW Keyer or the RTTY TinyFSK sketch
- If the red LED blinks four times, pauses, then blinks two times (CW for "HI") when power is applied (immediately after USB cable plugged in), then the CW Keyer sketch is installed
- Otherwise – neither "Blink" for "HI" - then the TinyFSK Keyer sketch is probably installed

Selecting an Arduino Integrated Development Environment – the IDE

You can skip this step if the previous tests revealed that you have the preferred sketch installed.

After successfully connecting your Arduino Nano to a serial port on your PC, you need to decide how to manage and upload applications (called "Sketches" in Arduino land). You can read about using the Arduino Online IDE or the Arduino IDE for Windows here:
https://www.arduino.cc/en/Main/Software

Uploading Sketches – What Does the JP1 Jumper Do?

When the Mortty board JP1 jumper is removed, the Arduino is in Program Mode (for installing/switching sketches). When the jumper is installed, the Arduino is in Operate Mode (for running the uploaded application). To initially upload a sketch, or if you want to switch to another sketch (for example, for switching Mortty from CW to RTTY keying), remove the jumper before launching the IDE. When you want to run a sketch, install the jumper before launching the application program (writelog, n1mm, dxlab…).

Jumper Hint: If the red LED (PTT) blinks 7 times when attaching the USB cable, the JP1 jumper is not installed and Mortty is in Program Mode.

Arduino Sketches for CW Keyer or TinyFSK

TinyFSK Sketch

Description by its author, Andy Flowers, K0SM: http://www.frontiernet.net/~aflowers/tinyfsk/
DOWNLOAD link to TinyFSK: http://www.frontiernet.net/~aflowers/tinyfsk/TinyFSK.ino

Another reference, with some good step-by-step instructions: *Programming an Arduino for TinyFSk*, by Dave K6LL: http://lists.contesting.com/archives//html/RTTY/2016-02/msg00001.html

CW Keyer Sketch

Description by its author, Anthony Good, K3NG: https://blog.radioartisan.com/arduino-cw-keyer
DOWNLOAD link to the CW Keyer: https://github.com/k3ng/k3ng_cw_keyer

Another reference: *A Few Good CW Keyers*, by Jeff Blaine, AC0C: http://ac0c.com/main/page_home_page.html

After downloading K3NG's CW Keyer sketch

1. Create a new subdirectory and unzip the files into it
2. From within the Arduino IDE, or using a text editor (Notepad?), open the file **keyer_hardware.h** in the subdirectory **/k3ng_cw_keyer-master/k3ng_keyer**. Locate the line beginning with **//#define HARDWARE_MORTTY.** Uncomment that line by removing the two forward-slashes (//) at the beginning of the line. This inserts Mortty-specific instructions into the compiled and uploaded sketch.
3. Using the Arduino IDE
    a. Set the >Tools menu for: Board: "Arduino Nano"; Processor: "ATmega328P"; Port: <the serial port defined for this CH230G driver. Consult the Windows >Device Manager >Ports list if necessary>
    b. Press the "compile and upload" icon (you did remove the JP1 jumper, right?)
    c. A successful compile and upload will display "avrdude done. Thank you."
    d. Reinstall the JP1 jumper

Step 6. Operation

Install the JP1 Jumper!

The Mortty board JP1 program/operate jumper must be installed for CW or RTTY operation. Insert it now.

Connect Mortty to Your Radio

The only missing piece from your Mortty parts kit is the specific cable that connects Mortty to your radio. On the Mortty end of the connection is the three conductor 3.5mm "stereo headphone" jack that supplies ground on the jack sleeve, push-to-talk (PTT) on the jack ring, and the CW or FSK keying signal on the jack tip (see Mortty schematic). If you plan to switch Mortty between CW and RTTY, you need two cables – one with your radio's CW connections and the other with your radio's RTTY connections.

CABLE HINTS: #1 Consider sacrificing a standard stereo headphone cable with 3.5mm (1/8") connectors for this purpose. Cut off one end and replace it with the connectors required for your radio. #2 Buy a long headphone cable (6' or 12'), cut it in half and use one end as your CW cable, the other end as your RTTY cable. #3 Ed W0YK suggests buying one of the following pig-tail cables from Mouser: 172-7434-E (36" right angle connector) or 172-7435-E (72" straight connector)

- For CW operation, connect sleeve to ground, ring to push-to-talk (PTT) and tip to the CW key jack. The red LED indicates PTT, the green LED is CW key
- For RTTY operation, connect sleeve to ground, ring to PTT and tip to the FSK line. These signals are found on an auxiliary connector on the rear of most radios. The red LED indicates PTT, the green LED is FSK data

Adjust Your Logging Program

For CW Operation

K3NG's CW Keyer sketch emulates the popular K1EL Winkeyer, which is supported by most logging programs. Follow your logging program's instructions for using a Winkeyer.

For RTTY Operation

K0SM's TinyFSK sketch is supported by many logging programs and digital mode applications. The list includes (at least) WriteLog, DXLab WinWarbler, G3YYD's 2Tone, and N1MM Logger Plus. MMTTY can be used as a RTTY receive demodulator when used with WriteLog or N1MM directly driving TinyFSK as a RTTY modulator. There may be others - refer to your logging software's documentation for TinyFSK interface instructions.

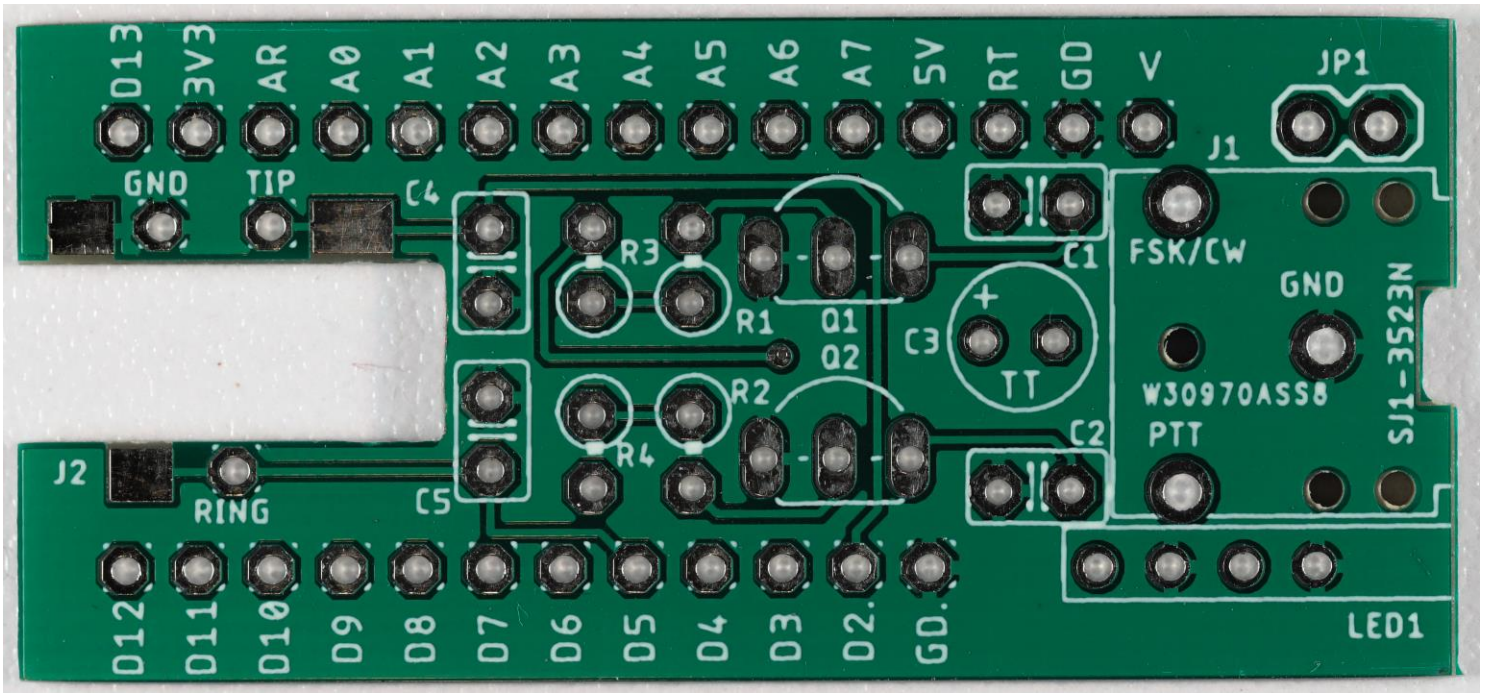
If you discover that your RTTY signals are "Upside Down"

When TinyFSK starts, notice the Configuration report in your RTTY Rx window. You might need the opposite polarity (for example, the Elecraft K3 needs Mark = HIGH and the TinyFSK default is Mark = LOW). To permanently adjust polarity, launch the Arduino IDE and select >Tools >Serial Monitor. In the text entry field at the top of the screen, enter ~0 (a tilde, then a zero, then <return>) for Mark = High; or ~1 (a tilde, then a one, then <return>) for Mark = Low. The settings will be written into the Arduino EEPROM memory.

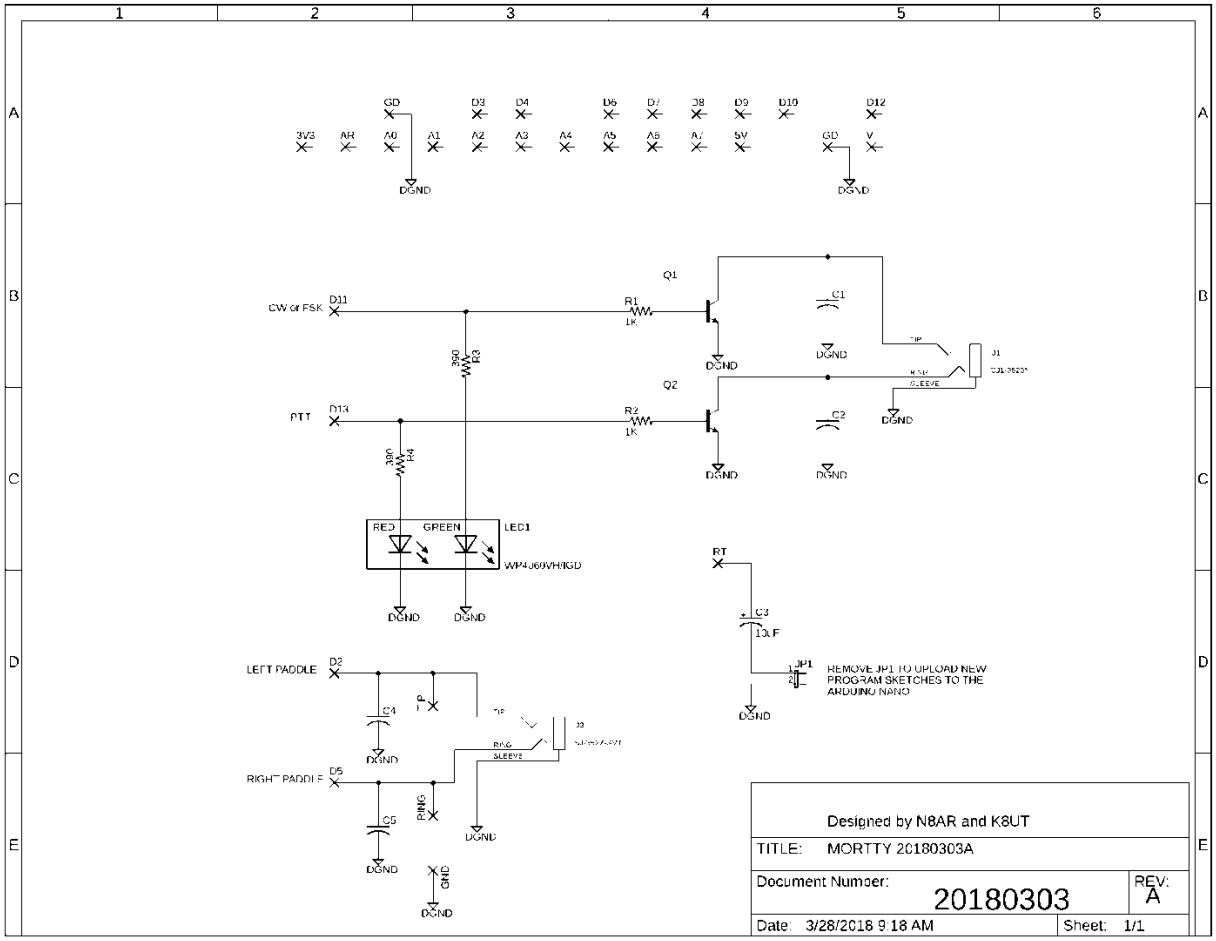
Appendix 1. Mortty Parts List and Board Layout

IMPORTANT: A specific model/manufacturer of Arduino Nano was selected for Mortty because it arrives without connectors soldered to the board. Other brands may be furnished with soldered connectors on the bottom of the board. A Bottom-of-Board connector arrangement will not fit within this small metal enclosure using Mortty's physical layout.

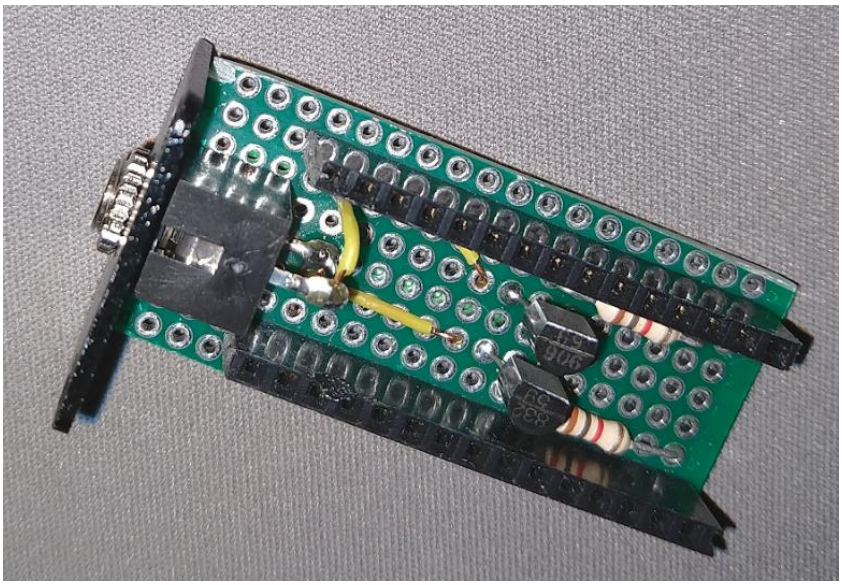
REF	NAME	PART NO	DESCRIPTION	MFG	QTY	SOURCE
Mortty	PC Board	20180303A	Mortty Interface Board	PCBway	1	N8AR/K8UT
R1,R2	Resistor	CFR-25JR-52-1K	Carbon film, 1K OHM, 5%	Yaego	2	Mouser
R3,R4	Resistor	CFR-25JR-52-390R	Carbon film, 390 OHM, 5%	Yaego	2	Mouser
C1,C2,C4,C5	Capacitor	K102K15X7RH5UH5	Ceramic, Radial Leads,1nF,10%	Vishay	4	Mouser
C3	Capacitor	ECE-A1CKA100I	Aluminum Electrolytic, 10uF, 20%	Panasonic	1	Mouser
Q1,Q2	Transistor	2N4401TF	NPN Transistor, 2N4401	ON Semi	2	Mouser
J1	Phone Jack	SJ1-3523N	Stereo Phone Jack, 3.5mm, RA	CUI	1	Mouser
J2	Phone Jack	SJ-3523-SMT-TR	Stereo Phone Jack, 3.5mm, RA, SMD	CUI	1	Mouser
LED1	Bi-level LED	WP4060VH/GID	1.8mm Bi-Level LED, (red,green)	Kingbright	1	Mouser
J3	Header	855-M20-7821546	Dupont female header, 1x15, 2.54mm	Harwin	1	Mouser
J4	Header	855-M20-7821246	Dupont female header, 1x12, 2.54mm	Harwin	1	Mouser
JP1	Header	855-M20-9990245	Male Header,RA,1x2, 2.54mm	Harwin	1	Mouser
	Jumper	855-M7581-05	Female 2 pin jumper (shunt), 2.54mm	Harwin	1	Mouser
	Enclosure		Extruded enclosure, 25x25x50mm	Eightwood	1	eBay/Amazon
	Arduino Nano		Arduino Nano, CH340/ATmega328P	Elegoo	1	Amazon
	USB Cable		USB 2.0, A-Male to Mini B, 3 feet	Amazon	1	Amazon



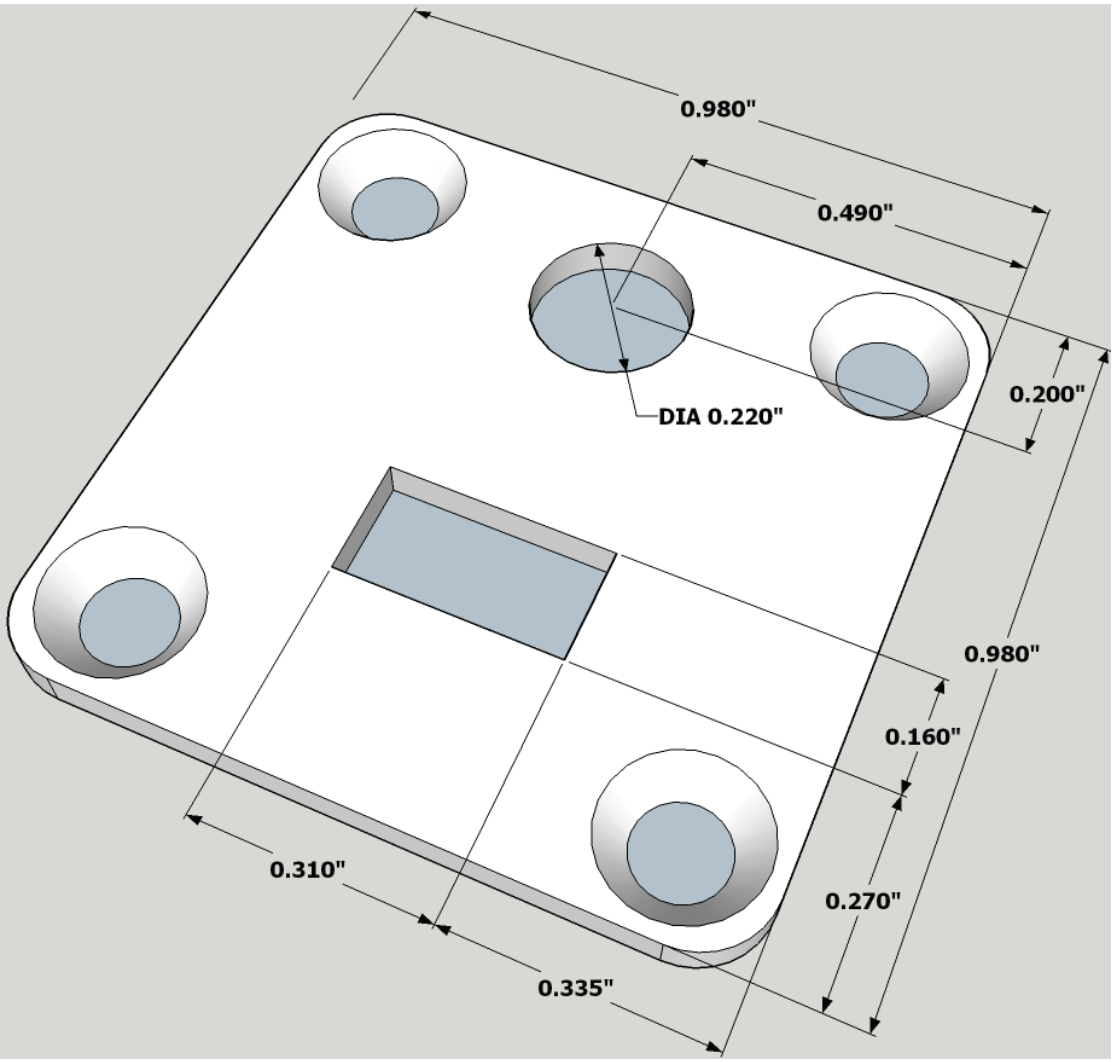
Appendix 2. Mortty Schematic Diagram



If building your keyer from scratch, here is an example of perf board construction

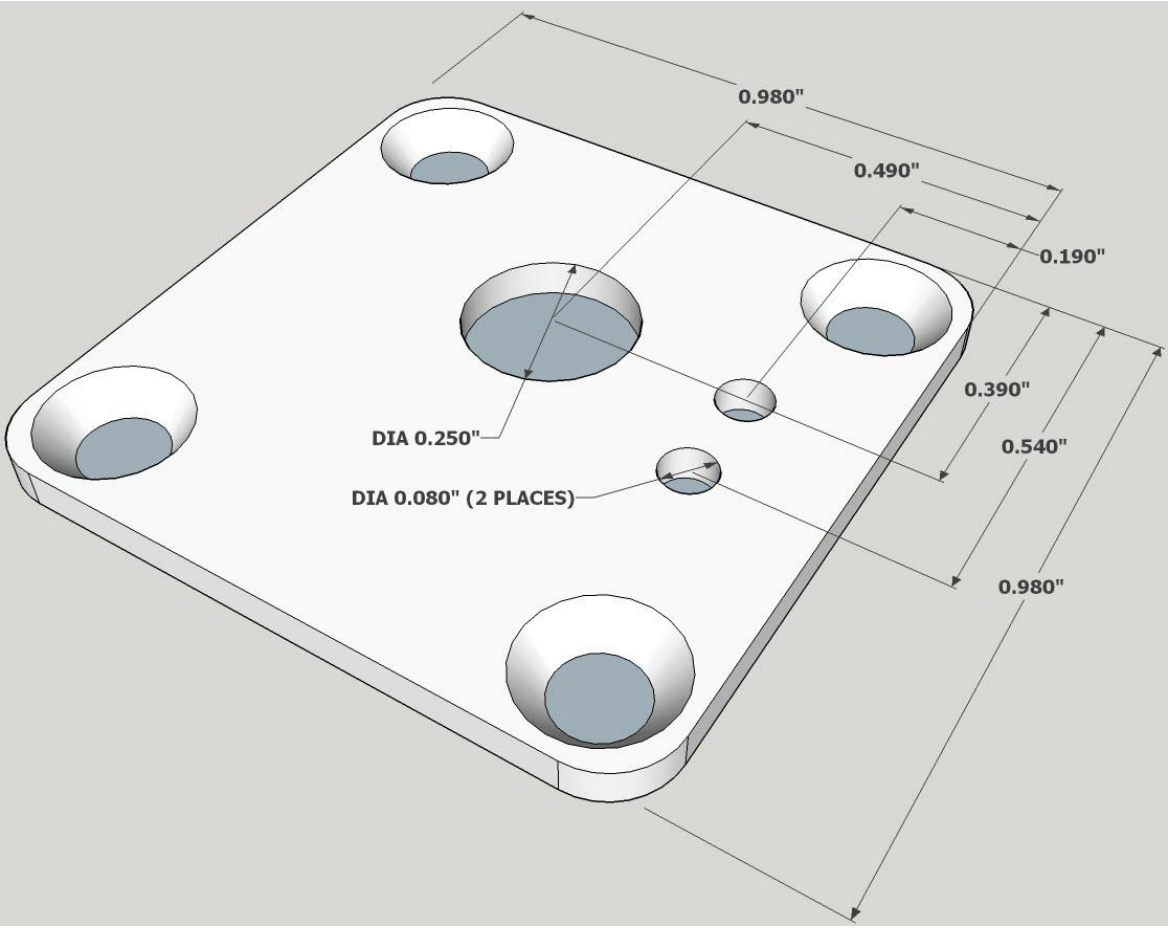


Appendix 3. End Cap Drilling Templates



Input End Cap

- USB connector from computer
- 3.5mm (1/8”) jack from CW paddle
    - Tip = Dash paddle
    - Ring = Dot paddle
    - Sleeve = Ground



Output End Cap

- 3.5mm (1/8”) jack to radio
    - Tip = CW or FSK Key output
    - Ring = Push To Talk (PTT) or SO2R CW Key radio 2
    - Sleeve = Ground

- Red LED = PTT (or Radio 2 with SO2R sketch)
- Greed LED = RTTY FSK or CW Key