

Mortty v3- 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. It optionally can be configured to run W1HKJ's nanoIO software using fldigi.

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 and control programs. This tiny box measures only 2 inches long and 1 inch square (50mm by 25mm). Mortty has two input connectors (a microUSB jack to connect a computer and a 3.5mm TRS jack to connect a CW paddle) and one output connector (a 3.5mm TRRS jack that provides PTT, CW and FSK signals to your transceiver).



Designed by N8AR and K8UT, Mortty v3 is available as a complete parts kit, or you can use this guide to build your own from "scratch."

Copyright:

The Mortty schematic and design are copyrighted freeware. You can use, modify, and distribute the design provided that you offer it and any derivative works as freeware. Any commercial use must be approved in writing by the authors.

The *TinyFSK*, *CW Keyer* and *nanoIO* programs are copyrighted works licensed by their respective authors.

Disclaimer:

This device controls equipment that could be damaged by said device. You are responsible for installing, configuring, testing and verifying that the device performs properly in your environment. The developers cannot be held liable for any direct, indirect, consequential or incidental damages to other pieces of software, equipment, goods or persons arising from the use of this device.

By constructing this device you accept the above terms of copyright and disclaimer.

Release Notes:

- | | | |
|-----|------------|--|
| 1.0 | 2018-04-02 | Initial TinyFSK release. RTTY operation only |
| 2.0 | 2018-06-13 | Mortty release. CW and RTTY operation |
| 3.0 | 2018-07-05 | Simplified sketch management. Support for single-mode and dual-mode sketches |

Table of Contents

Mortty Overview.....	4
Key Features.....	4
Setting Expectations.....	4
Required Materials.....	5
Step 1. Assembling the Arduino Nano	6
Step 2. Choose Between Standard Single-Mode and Optional Dual-Mode Configuration	9
Step 3. Assembling the Mortty v3 Circuit Board.....	10
Step 4. Final Assembly	13
Step 5. Installing Software	14
Loading the CH340G Drivers into Windows.....	14
Is your CW Keyer, TinyFSK, or nanoIO Sketch Already Installed?	14
Selecting an Arduino Integrated Development Environment - the IDE.....	15
Uploading Sketches - When configured for a Standard Single-Mode Sketch.....	15
Uploading Sketches - When configured for an Optional Dual-Mode Sketch.....	15
Arduino Sketches for CW Keyer, TinyFSK and nanoIO	15
Step 6. Building Custom Cables to Connect Mortty v3 to Your Rig	16
A Custom Cable for Standard Single-Mode Sketch Operation	16
A Custom Cable for Optional Dual-Mode Sketch Operation	17
Step 7. Operation.....	18
Adjust Your Logging Program.....	18
Appendix 1. Mortty Parts List and Board Layout	19
Appendix 2. Mortty Schematic Diagram	20
Appendix 3. Modifying Mortty v3 for Dual-Mode Configuration	21
Appendix 4. Cutting Holes in the Metal Enclosure End Caps.....	22

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
- Optional Dual-Mode Sketch CW and RTTY operation using nanoIO and fldigi
- Construction requires only common hamshack tools

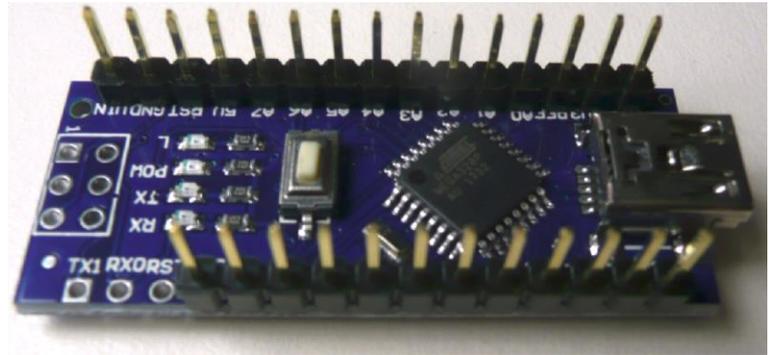
Setting Expectations

- **What's missing from this miniature design?**
 - For CW - No speed potentiometer, no macro pushbuttons, no monitor speaker
- **Standard Mortty v3 Single-Mode Sketch configuration, CW or FSK - but not both?**
 - Mortty operates either CW Keyer by K3NG or TinyFSK by K0SM
 - Switching between modes involves: closing the logging program, unplugging the Mortty output cable, launching the Arduino IDE, uploading the desired sketch (CW or RTTY); re-inserting the output cable
 - Elapsed time for switching modes - about one minute
 - Maybe you should build two Morttys? ;-)
- **Optional Mortty v3 Dual-Mode Sketch configuration using nanoIO and fldigi - both CW and FSK**
 - Optional assembly configuration provides for separate PTT, CW and RTTY outputs (three wires)
 - Requires using nanoIO by W1HJK with fldigi for CW and RTTY operation
 - Switching between modes involves selecting a different mode and modem in fldigi
- **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 end caps
 - 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 may 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
 - There 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
 - Based on your Mortty Kit order, we will preload the Arduino Nano with your desired sketch (TinyFSK, CW Keyer, or nanoIO)

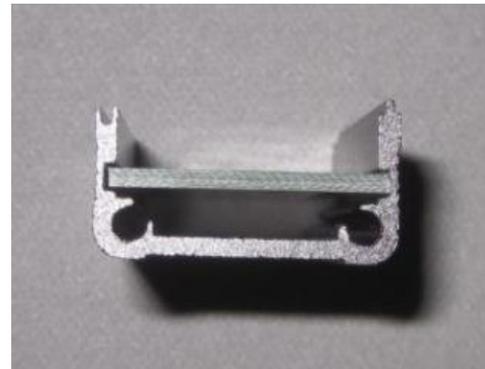
Step 1. Assembling the Arduino Nano

IMPORTANT! Due to space constraints in the miniature enclosure, the Arduino headers are installed on the TOP of the Arduino circuit board, not the bottom. One of the headers is shortened to only 12 pins. Do not solder any headers to the Arduino Nano until examining the photos and carefully reading the following instructions. Your Nano board should appear as shown below upon completion of Step 1.

SOLDERING HINT: In the following section, when soldering the header pins to the Arduino Nano, begin with only one pin on each header, then stop and examine/reheat/adjust each header for vertical alignment. When satisfied that the headers are 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.



Remove the Mortty v3 board from the Mortty plastic parts bag and gently try to slide the board into the card guide slots of one of the metal enclosure extrusions. 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.



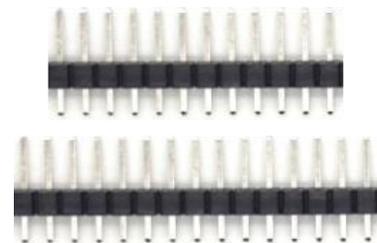
Open the Arduino Nano package

If your Arduino package contains this 6 pin male Dual-In-Line male header, recycle it. **Do Not Install this header on the Arduino.**



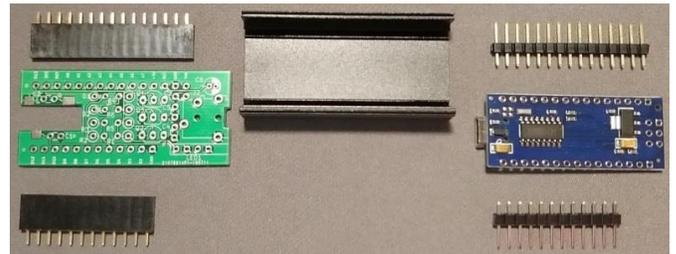
If your Arduino package contains two 15 pin male Single-In-Line (SIL) headers, shorten one header to contain 12 pins as follows:

Using a pair of needle-nose pliers, grasp the plastic of the third pin from a connector end and carefully snap the last three pins from the end of one of the headers.

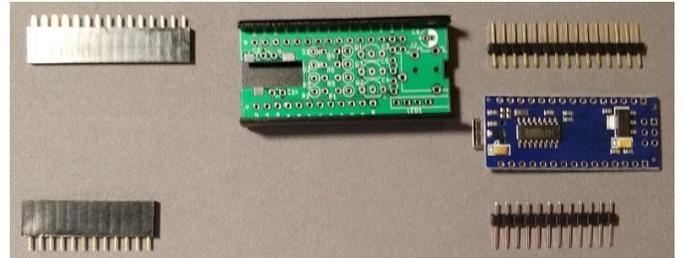


Locate the seven items shown in the photo to the right:

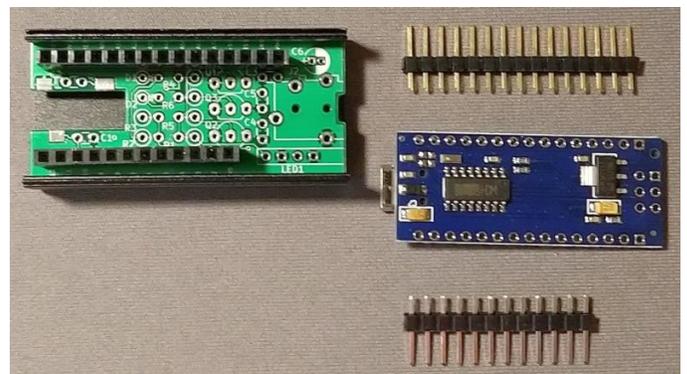
1. One of the enclosure extrusions
2. Mortty printed circuit board
3. Arduino Nano processor
4. Nano 12 pin male header
5. Nano 15 pin male header
6. Mortty 12 pin female header
7. Mortty 15 pin female header



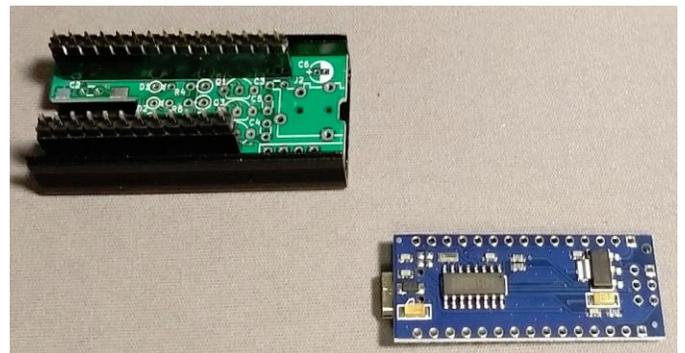
Insert the Mortty PCB into the slots in the extrusion so that the silkscreen on the Mortty PCB is visible (facing up)



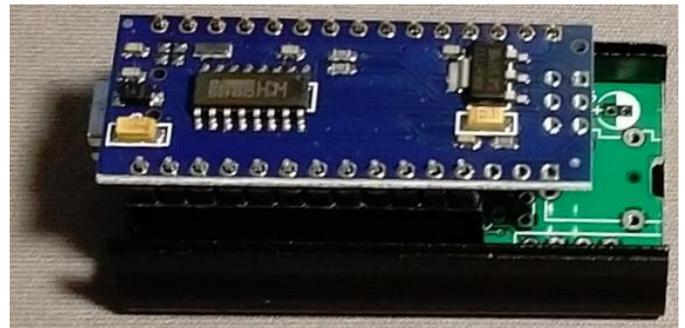
Insert the 12 and 15 pin female SIP headers into the corresponding holes on the Mortty printed circuit board



Insert the long pins of the 12 and 15 pin male SIP headers into the corresponding female headers on the Mortty printed circuit board

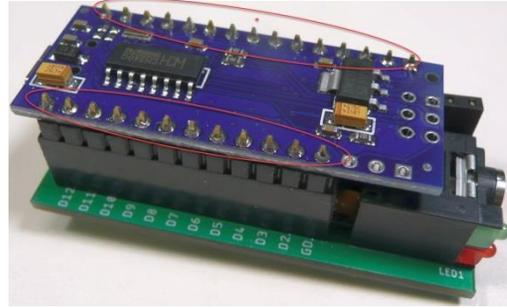


Place the Aduino Nano processor onto the short pins of the 12 and 15 pin male headers. Align the Nano so that the USB jack is on the BOTTOM (facing the Mortty PC board) and facing to the LEFT. Ensure that the headers are at right angles to the Nano and then solder the 27 pins of the two male headers to the Nano processor board. **Do not solder pins to the Mortty PCB.**



- Inspect the solder connections you just made on the Arduino Nano. To ensure that these pins will not accidentally short to the Mortty case after final assembly, use side cutters to trim any excess lead length above your solder joints from the 12 and 15 pin headers by clipping them as close to your solder connections as possible.

- Lift the Arduino Nano with the attached male/female headers from the Mortty PCB. Set it aside for now taking care not to bend or damage the exposed pins on the female headers. You will use this assembly as a fixture in Step 3 to solder the female headers to the Mortty PC board.



Step 2. Choose Between Standard Single-Mode and Optional Dual-Mode Configuration

The major improvement in Mortty version3 over version2 was the elimination of an internal jumper that switched the Arduino Nano between Program Mode and Operate Mode. With Mortty v3, the new output jack senses the insertion of an output cable and switches between Operate and Program modes, greatly simplifying the task of changing between the single-mode TinyFSK and single-mode CW Keyer sketches. **Most users will want to choose the Standard Single-Mode Sketch configuration for Mortty v3.**

Another improvement in Mortty version3 over version2 is its optional support of a dual-mode sketch program called nanoIO, developed by W1HKJ. This optional configuration uses a dual-mode sketch that allows fldigi to switch Mortty between CW and RTTY modes without changing sketches. **Users will want to consider the Optional Dual-Mode Sketch configuration for Mortty v3 if they: already operate fldigi; require constant and fast switching between CW and RTTY operation; or operate on non-Windows platforms (fldigi runs on Windows, Mac, and linux platforms).**

NOTE: See additional details about the differences between Standard Single-Mode and Optional Dual-Mode Sketch configurations in the documentation *Step 6: Operation, "Building a Custom Cable..."*

If you chose the Standard Single-Mode Sketch configuration, go to *Step 3. Assembling the Mortty v3 Circuit Board*

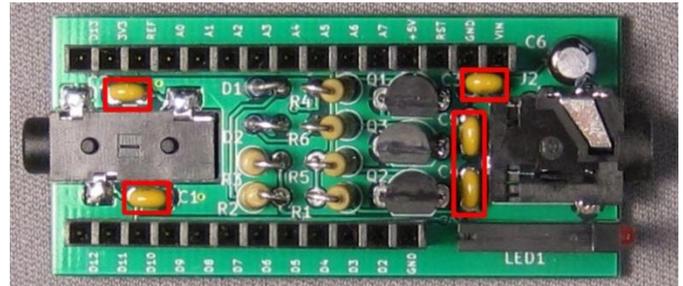
OR

If you chose the Optional Dual-Mode Sketch configuration, first go to Appendix 3. *Modifying the Mortty v3 Circuit Board*, then proceed to *Step 3. Assembling the Mortty v3 Circuit Board*

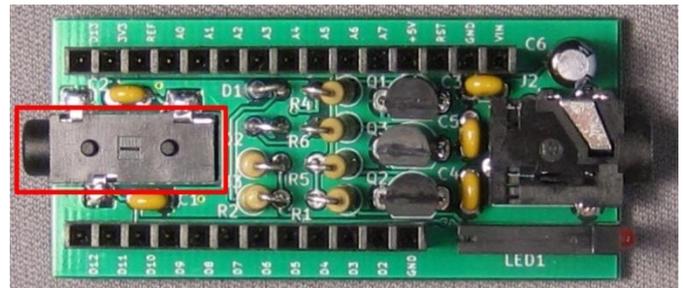
Step 3. Assembling the Mortty v3 Circuit Board

From the Mortty parts envelope

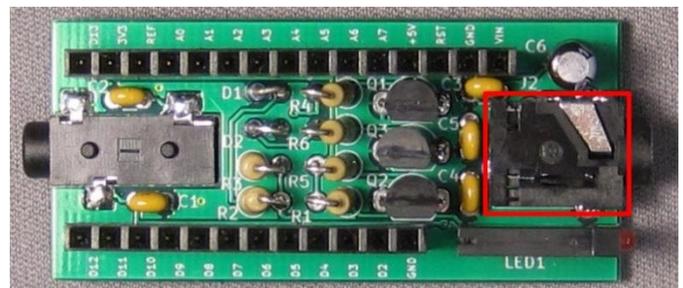
Locate the five 1 nF (1000 pF) ceramic capacitors. Insert the capacitors on the TOP (labeled) side of the Mortty board in positions C1, C2, C3, C4 and C5. Hold each capacitor upright on the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



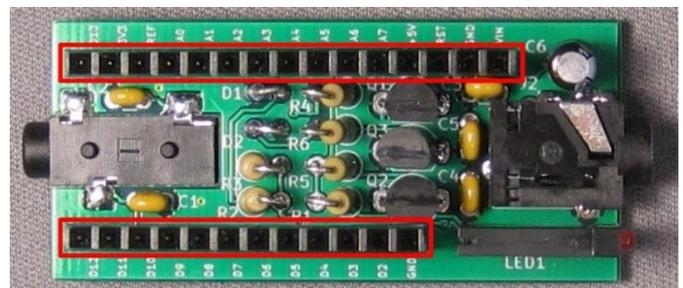
Locate the 3.5mm surface mount jack. Insert the surface mount jack from the TOP (labeled) side of the Mortty board in the large notch indicated as J1. Press jack J1 against the three surface mount pads and apply solder **on the top of the board**. NOTE: This is the only component that will be soldered from the top.



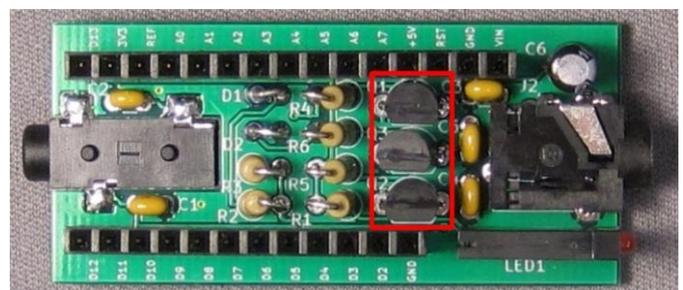
Insert the other 3.5mm jack from the TOP (labeled) side of the Mortty board in the location marked J2. Hold jack J2 firmly against the Mortty board and apply solder to the bottom of the board.



Place the Arduino Nano with the male/female headers still attached that you assembled in Step 1 on a flat surface with the pins of the female headers facing up. Place the Mortty PCB onto the female headers and solder the 27 pins of the female headers to the Mortty PCB. Unplug the Nano from the Mortty PCB after completing the solder joints

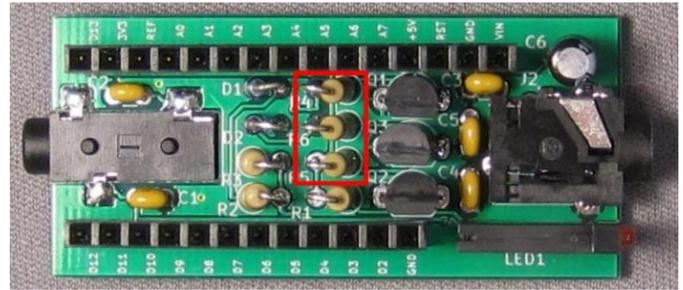


Locate the three transistors and insert them from the TOP (labeled) side of the board in positions Q1, Q2 and Q3. Align the flat side of the transistors with the flat side in the silkscreen on the Mortty board. Press them far enough onto the board so that the tops of the transistors are slightly lower than the height of the female headers. Apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.

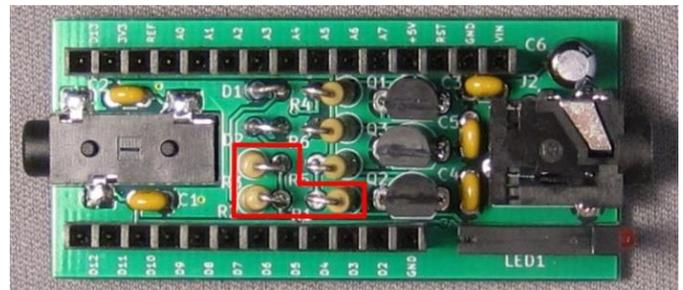


IMPORTANT! Due to Mortty space constraints, resistors and diodes are installed vertically on the board.

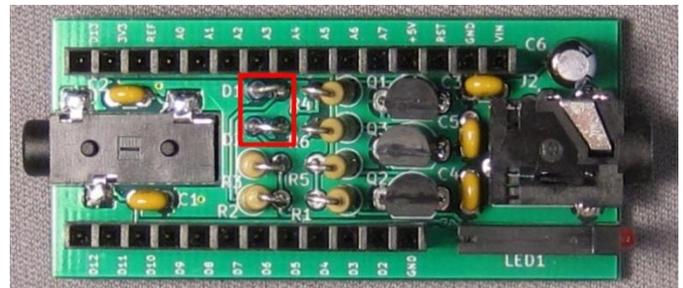
Locate the three 1000 ohm (brown-black-red) resistors. Fold one lead over so that the leads are parallel to each other. Insert the 1000 ohm resistors on the TOP (labeled) side of the Mortty board in positions R4, R5 and R6. Hold each resistor upright so that one end of the resistor is flush against the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



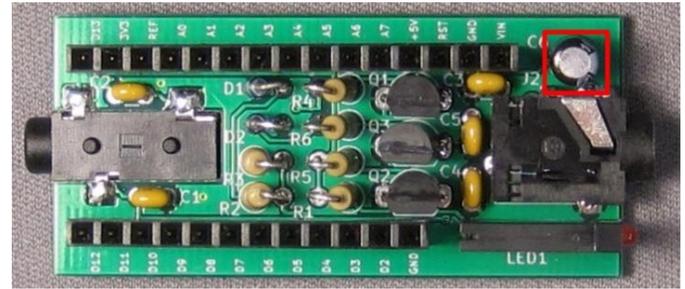
Locate the three 390 ohm (orange-white-brown) resistors. Fold one lead over so that the leads are parallel to each other. Insert the 390 ohm resistors on the TOP (labeled) side of the Mortty board in positions R1, R2 and R3. Hold each resistor upright so that one end of the resistor is flush against the Mortty board and apply solder to the bottom of the board. Trim any excess lead length from the bottom of the board.



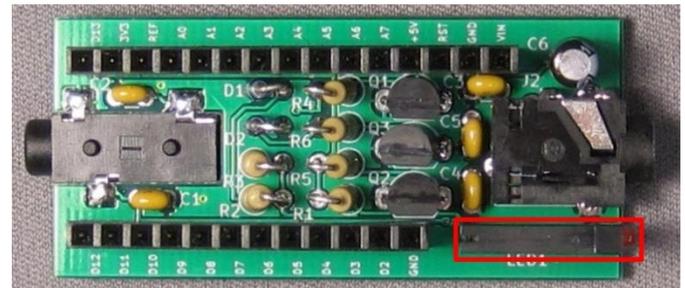
Locate the two Schottky diodes. Fold the lead at the cathode end of each diode over so that the leads are parallel to each other. (The cathode end of the diode is denoted by a black ring). Insert the two diodes on the TOP (labeled) side of the Mortty board in positions D1 and D2. The unbent lead of the diode (anode end, no band) **must** be inserted into the hole closest to the D1 and D2 designations silkscreened on the board. Hold each diode upright so that one end of the diode 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 10 uF electrolytic capacitor. Insert the capacitor on the TOP (labeled) side of the Mortty board in the location marked C6. 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 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.



This completes the assembly of the Mortty v3 board. Examine the board to ensure that all of the resistors, capacitors, diodes and transistors are below the height of the two female headers on the sides of the board.

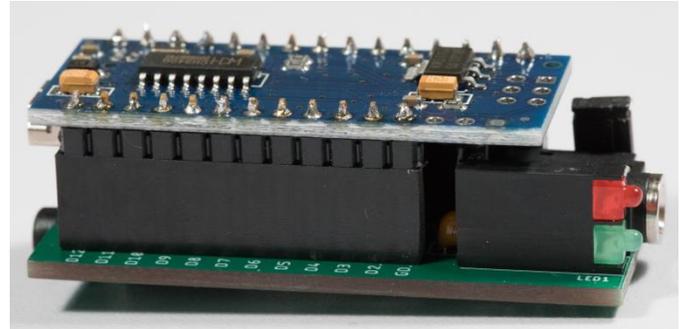
Step 4. Final Assembly

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...

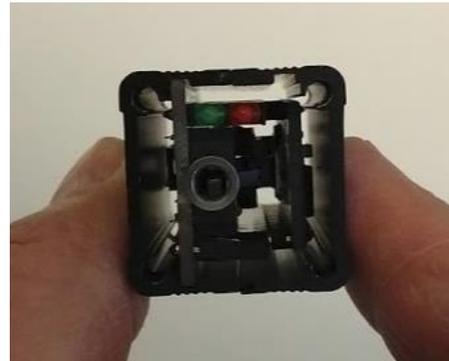
NOTE 1: If you prefer to use Mortty's optional metal end caps, see *Appendix 4: Cutting Holes in the Metal End Caps*

NOTE 2: If you chose to configure Mortty v3 in the Optional Dual-Mode Sketch configuration, the desired dual-mode sketch (*nanoIO*) should be uploaded to the Arduino Nano before final assembly.

- 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.



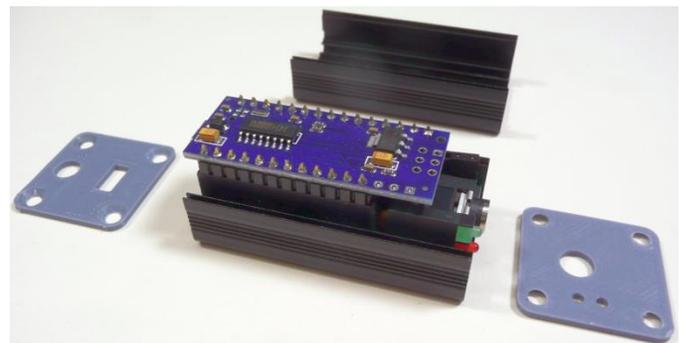
- Slide the Mortty board into the guides of the bottom of the enclosure. Place the cover on the enclosure. Stare through the case and examine the gaps between the enclosure and the adjoining boards. They will be very close, but the leads from the Mortty board and the Arduino Nano must not touch the case. 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 covers by attaching the 2 endcaps using eight Phillips head screws. Use care to avoid cross-threading the screws.
- **TIGHT FIT?** You may find the holes in the 3D printed end caps 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

Is your CW Keyer, TinyFSK, or nanoIO Sketch Already Installed?

Unplug your output cable from Mortty. Verify which sketch is installed on the Arduino Nano by observing Mortty's red and green LEDs when power is applied by plugging in the USB cable from your computer.

- If the red LED is continuously blinking slowly (1 Hz rate), then the Arduino is executing its default "Blink" sketch and you will need to use the Arduino IDE (see below) to upload the CW Keyer, the RTTY TinyFSK, or nanoIO sketch
- If the red led blinks eight times and stops then the RTTY TinyFSK sketch is installed.
- If the red LED blinks eight times, pauses, blinks four short , pauses, then blinks two short (CW for "HI") and stops then the CW Keyer sketch is installed
- If the green LED blinks two short then three long (CW for "IO") and stops - then the nanoIO sketch is installed

*The Mortty developers pre-load an Arduino Sketch on the Nano prior based on your order form. After loading the sketch on the Nano, K8UT writes **CW, RTTY, or nanoIO** on the Nano bag. **If the preceding LED test confirms that you have the desired sketch, you can skip the remainder of Step #5 instructions and proceed to Step #6.***

Please watch the video about software installation at <http://mortty.info/mortty/mortty-assembly> before proceeding. Also, 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 supplied with Mortty uses a CH340G UART that requires a Windows 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. Drivers are available here for Windows, Mac, linux, and android platforms. 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>

Installing the 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 install the Arduino IDE to manage and upload applications (called “Sketches” in Arduino land). You can read about using the Arduino IDE for Windows here: <https://www.arduino.cc/en/Main/Software>

Uploading Sketches - When configured for a Standard Single-Mode Sketch

If your Mortty v3 is configured for a Standard Single-Mode Sketch (CW Keyer **or** TinyFSK), insertion or removal of the output plug will switch Mortty from Operate Mode to Program Mode. To install a new sketch or to change between CW and RTTY sketches, unplug the output cable that leads to the rig, then launch the Arduino IDE and upload the sketch. After uploading the desired sketch, connect the appropriate cable (CW or RTTY) from the rig into the Mortty output jack and launch the application program (writelog, n1mm, dxlab...).

Uploading Sketches - When configured for an Optional Dual-Mode Sketch

If your Mortty v3 is configured for an Optional Dual-Mode Sketch (*nanoIO*), the case must be opened and the Arduino Nano separated from the Mortty PCB to switch Mortty between Program Mode and Operate Mode. To install a new nanoIO dual-mode sketch, disconnect the Arduino Nano from the Mortty printed circuit board to place the Nano in Program Mode, then launch the Arduino IDE and upload the new sketch. Once you have uploaded the desired sketch, re-assemble Mortty to place the Nano in Operate Mode and launch the application program (fldigi).

Arduino Sketches for CW Keyer, TinyFSK and nanoIO

TinyFSK Sketch

Description by its author, Andy Flowers, K0SM: <http://www.frontiernet.net/~aflowers/tinyfsk/>

Link to the TinyFSK download: <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>

Link to the CW Keyer download: 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

nanoIO Sketch and fldigi

Description by its author, David Freese: <http://www.w1hjk.com>

Link to fldigi download: <https://sourceforge.net/projects/fldigi/files/fldigi/>

Link to the nanoIO download: <http://www.w1hjk.com/files/nanoIO/>

Other references:

fldigi Users Manual <http://www.w1hjk.com/FldigiHelp/index.html>

nanoIO Interface <http://www.w1hjk.com/files/nanoIO/>

Step 6. Building Custom Cables to Connect Mortty v3 to Your Rig



A Custom Cable for Standard Single-Mode Sketch Operation

The Standard Single-Mode Sketch Configuration for Mortty v3 runs either *TinyFSK* or *CW Keyer*. This configuration requires a three conductor 3.5mm stereo (TRS) plug and cable to connect to your rig. If you intend to operate CW and RTTY you will need two cables: one cable will connect to your rig's PTT and CW key line; the other cable will connect to PTT and your rig's FSK input. Mortty v3 Standard Single-Mode Sketch configuration outputs are as follows:

- **TIP** - CW (K3NG CW Keyer Sketch) or RTTY (KOSM TinyFSK Sketch)
- **RING** - Push To Talk (PTT) for RTTY and CW
- **SLEEVE** - Ground

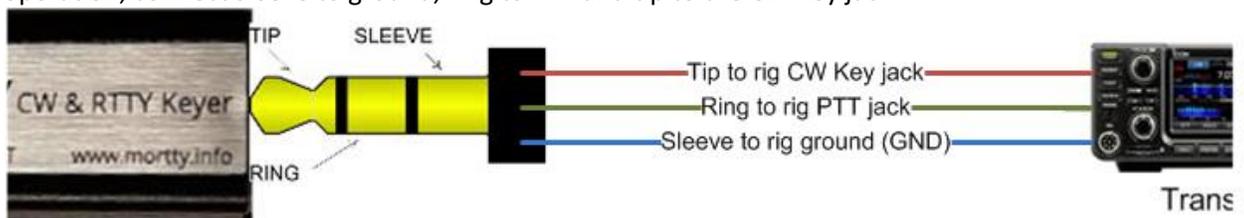


NOTE: In the Standard Single-Mode Sketch configuration, the elongated sleeve of the inserted TRS plug grounds the RING 2 pin of Mortty's internal TRRS jack. This short-to-ground switches the Arduino Nano from Program Mode to Operate Mode. (When a three conductor (TRS) 3.5mm stereo plug is inserted, RING 2 is internally grounded and Mortty v3 automatically switches to Operate Mode. When the TRS plug is removed, RING 2 is not grounded and the Arduino Nano is placed in Program Mode.)

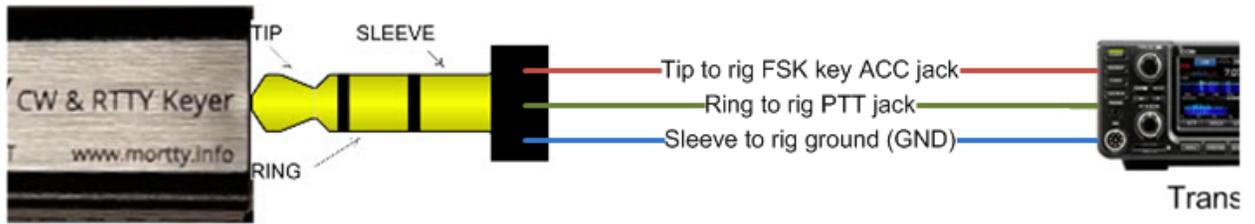
Build a custom cable that connects Mortty to your radio. On the Mortty end of the cable you use a standard three conductor 3.5mm TRS plug wired with radio ground on the plug sleeve, push-to-talk (PTT) on the plug ring, and the CW or FSK keying signal on the plug tip (see Mortty schematic). If you plan 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 HINT: Consider sacrificing a standard 3.5mm (1/8") stereo cable with a TRS connector for this purpose. Cut off one end and replace it with the connectors required for your radio. Ed WOYK 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 PTT and tip to the CW key jack



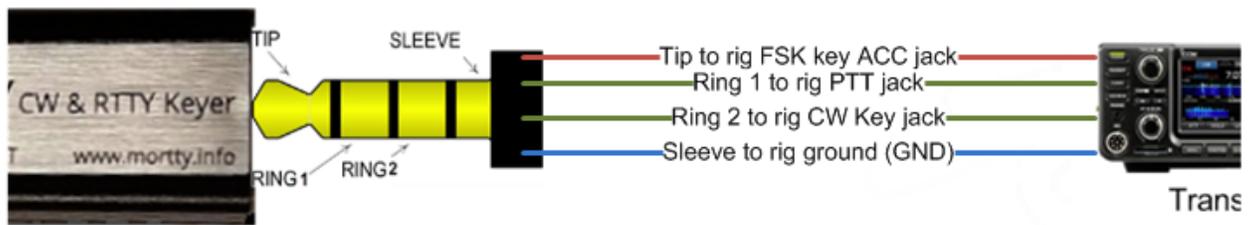
- For RTTY operation, connect sleeve to ground, ring to PTT and tip to the FSK line. These inputs are found on an auxiliary connector on the rear of most radios



A Custom Cable for Optional Dual-Mode Sketch Operation

The Optional Dual-Mode Sketch Configuration for Mortty v3 runs *nanoIO* controlled by *fldigi*. The optional configuration requires a four conductor (TRRS) 3.5mm plug and cable wired as follows:

- **TIP** - FSK input on your rig
- **RING 1** - Push To Talk (PTT) for RTTY and CW
- **RING 2** - CW key line
- **SLEEVE** - Ground



Build a custom cable that connects Mortty to your radio. On the Mortty end of the cable use a four conductor 3.5mm "TRRS" plug wired with radio ground on the plug sleeve, push-to-talk (PTT) on the plug ring 1, CW keying on ring 2, and RTTY keying on the plug tip (see Mortty schematic). You might want to consider buying a video adapter cable on Amazon or eBay and replacing the connectors with those required by your rig.

Step 7. Operation

Adjust Your Logging Program

For CW Keyer 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 TinyFSK Operation

KOSM's TinyFSK sketch is supported by many logging programs and digital mode applications. The list includes (at least) WriteLog, N1MM Logger Plus, DXLab WinWarbler, Ham Radio Deluxe, G3YYD's 2Tone, MMTTY... and probably others. Please refer to your logging software's documentation for TinyFSK interface instructions.

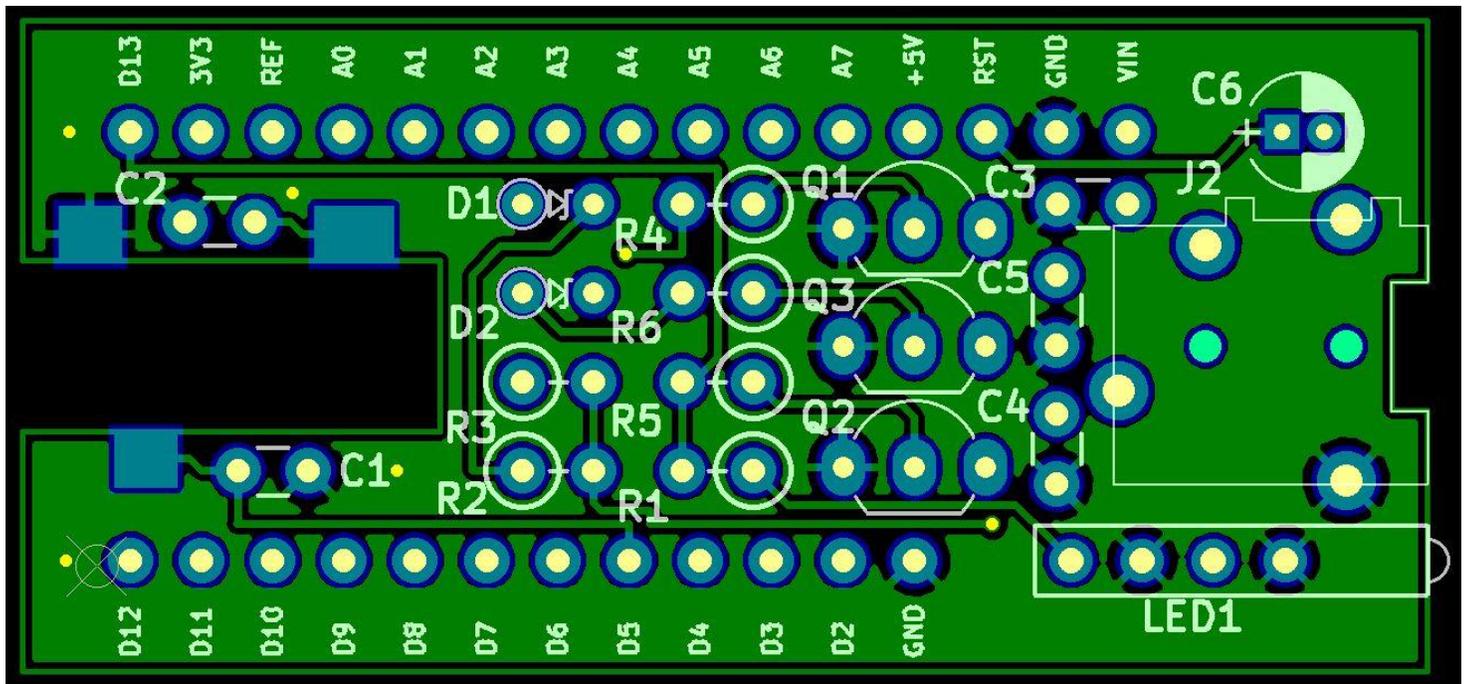
For nanoIO/flidigi Operation

Refer to the links provided for nanoIO and fldigi operation.

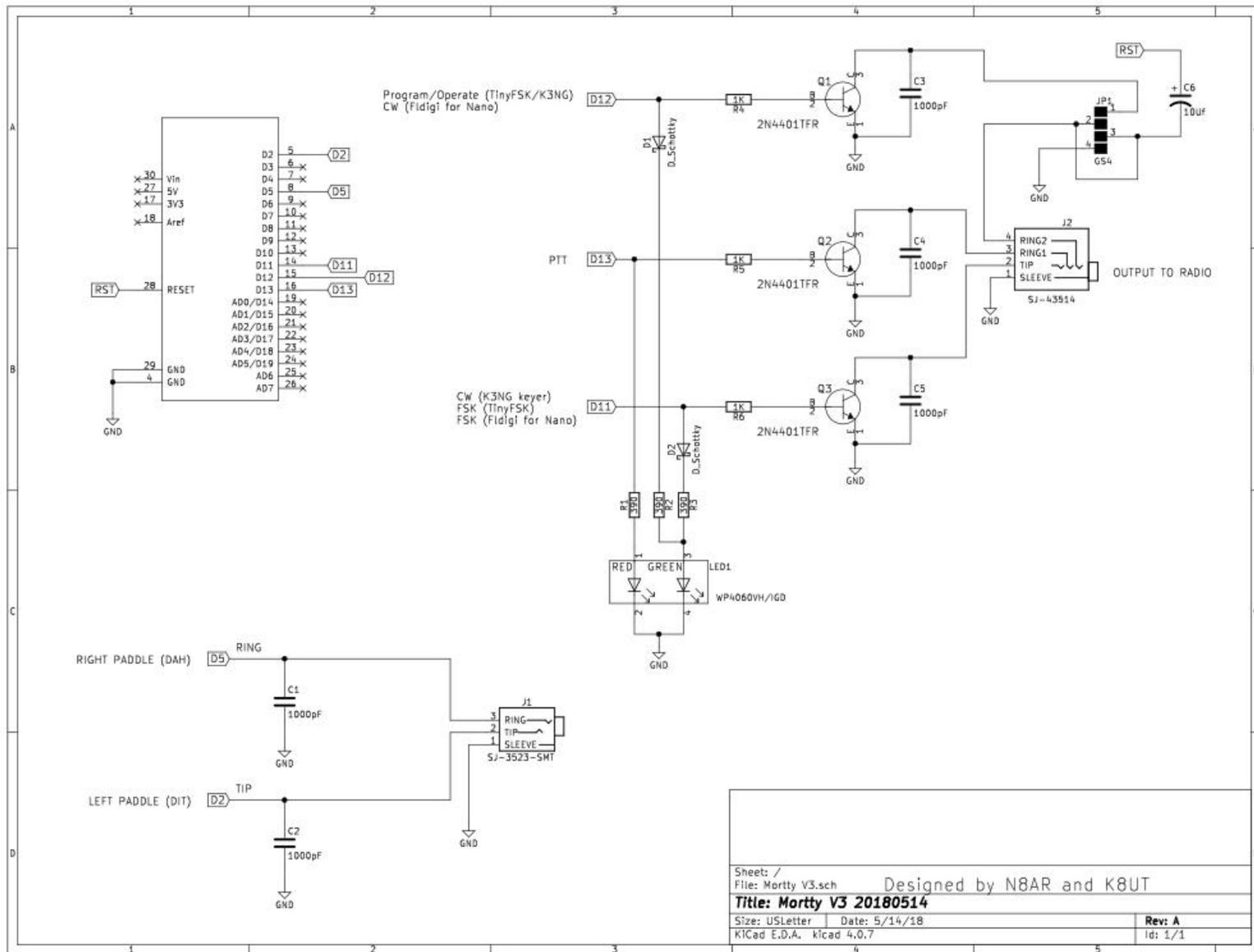
Appendix 1. Mortty Parts List and Board Layout

IMPORTANT: The furnished 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 v3	PC Board	20180514A	Mortty v3 Interface Board	PCBway	1	N8AR/K8UT
R1,R2,R3	Resistor	CFR-25JR-52-390R	Carbon film, 1K OHM, 5%	Yaego	3	Mouser
R4,R5,R6	Resistor	CFR-25JR-52-1K	Carbon film, 390 OHM, 5%	Yaego	3	Mouser
C1,C2,C3,C4,C5	Capacitor	K102K15X7RH53L2	Ceramic, Radial Leads,1nF,10%	Vishay	5	Mouser
C6	Capacitor	ECE-A1CKA100I	Aluminum Electrolytic, 10uF, 20%	Panasonic	1	Mouser
D1,D2	Diode	BAT43	Schottky Diode, BAT43	ST Micro	2	Mouser
Q1,Q2,Q3	Transistor	2N4401TF	NPN Transistor, 2N4401	ON Semi	3	Mouser
J2	Phone Jack	SJ-43514	4 Cond. Phone Jack, 3.5mm, RA	CUI	1	Mouser
J1	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
	Enclosure		Extruded enclosure, 25x25x50mm	Eightwood	1	eBay/Amazon
	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



Appendix 3. Modifying Mortty v3 for Dual-Mode Configuration

Configuring Mortty v3 for a Dual-Mode sketch requires minor alterations to the printed circuit board (PCB).

- On the back side of the PCB, use a sharp Exacto knife or razor blade to cut the connection between pads 2 and 3 of JP1. Use an ohmmeter to verify that the pads are no longer connected.



- Use a small-tipped solder pencil to melt enough solder to create a solder bridge between pads 1 and 2 of JP1. Similarly create a solder bridge between pads 3 and 4 of JP1. Verify both connections with an ohmmeter



This completes the optional modification of the Mortty v3 PCB for a Dual-Mode sketch. Proceed to this manual's *Step 2. Assembling the Mortty v3 Circuit Board.*

Appendix 4. Cutting Holes in the Metal Enclosure End Caps

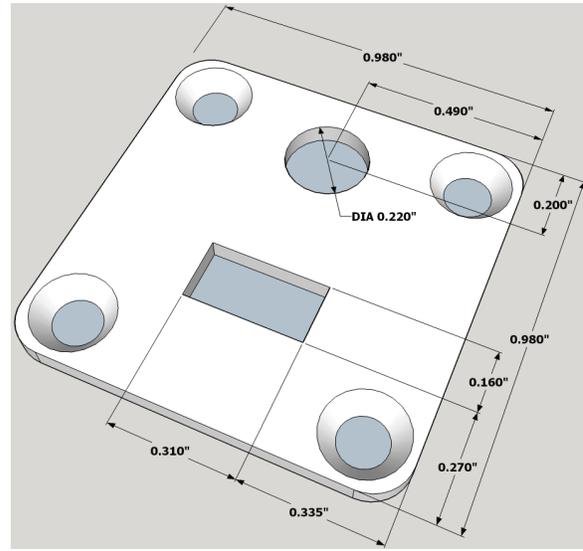
The Mortty kit ships with 3D printed plastic end caps that align with the connectors and LEDs on the printed circuit board. The enclosure's original metal end caps are also included. The following steps are required only if you decide to use the metal end caps. It is probably easiest to use the plastic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

- 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



- 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
- Status LEDs
 - Red = Push To Talk (PTT)
 - or, with SO2R sketch, CW Key radio 2
 - Green = CW or FSK key

