Python Radio 27: Work the World

Using a Raspberry Pi Pico to control a 1,500-watt transmitter

Simon Quellen Field

Simon Quellen Field

Follow

6 min read

·

Sep 21, 2024

Listen

Share

More

Press enter or click to view image in full size

Propagation map of the world

Screenshot by the author

Throughout this series, I have emphasized frequencies and power levels that are legal in the U.S. to use without a license. Those frequencies are so high (their wavelengths are so short) that they travel right through the ionosphere and out into space.

These frequency bands have names like VHF and UHF (very high frequency and ultra-high frequency).

Right below them is HF. High Frequency. These bands can refract and reflect off of the ionosphere and the earth to bounce around the globe.

Not surprisingly, when your signal can cross national boundaries you need a license recognized by international treaties. Such a license is very easy to get.

There is a written test of the rules you have to follow when transmitting. There are free simple study guides for the test. All the questions and answers are available online. You only need to get 74% right to pass (26 questions right out of 35). You can practice the test online as many times as you like (also here, and here, and here, and many more).

Having said all that, this project is about how to “key” a radio transmitter to send Morse code. The ideas here can be used to turn on and off many non-radio devices, up to 40 volts and half an ampere.

Our little Raspberry Pi Pico can only handle 3 volts, and only a few milliwatts. To switch on bigger things, like a 1,500-watt transmitter, we will use a transistor: the 2N4401 (although any NPN transistor will work).

2N4401 pinout

Screenshot by the author

Press enter or click to view image in full size

RP2040 with 2N4401 and mono plug

Image by author

The image above shows our entire hardware setup. The base of the transistor connects to pin 13 of the RP2040. The emitter connects to ground. The collector does not touch the computer. The transistor acts like a switch, connecting the collector to ground, and this completes the circuit to anything the mono plug is plugged into.

Press enter or click to view image in full size

A close view

A closer view (author’s image)

The code for this project is a slight modification of code we used in previous projects. The cwmorse.py module looks like this:

From machine import Pin

Class CWMorse:

Character\_speed = 18

Def \_\_init\_\_(self, pin):

Self.key = Pin(pin, Pin.OUT)

Def speed(self, overall\_speed):

If overall\_speed >= 18:

Self.character\_speed = overall\_speed

Units\_per\_minute = int(self.character\_speed \* 50) # The word PARIS is 50 units of time

OVERHEAD = 2

Self.DOT = int(60000 / units\_per\_minute) – OVERHEAD

Self.DASH = 3 \* self.DOT

Self.CYPHER\_SPACE = self.DOT

If overall\_speed >= 18:

Self.LETTER\_SPACE = int(3 \* self.DOT) – self.CYPHER\_SPACE

Self.WORD\_SPACE = int(7 \* self.DOT) – self.CYPHER\_SPACE

Else:

# Farnsworth timing from <https://www.arrl.org/files/file/Technology/x9004008.pdf>

Farnsworth\_spacing = (60000 \* self.character\_speed – 37200 \* overall\_speed) / (overall\_speed \* self.character\_speed)

Farnsworth\_spacing \*= 60000/68500 # A fudge factor to get the ESP8266 timing closer to correct

Self.LETTER\_SPACE = int((3 \* farnsworth\_spacing) / 19) – self.CYPHER\_SPACE

Self.WORD\_SPACE = int((7 \* farnsworth\_spacing) / 19) – self.CYPHER\_SPACE

Def send(self, str):

From the\_code import code

From time import sleep\_ms

For c in str:

If c == ‘ ‘:

Self.key.off()

Sleep\_ms(self.WORD\_SPACE)

Else:

Cyphers = code[c.upper()]

For x in cyphers:

If x == ‘.’:

Self.key.on()

Sleep\_ms(self.DOT)

Else:

Self.key.on()

Sleep\_ms(self.DASH)

Self.key.off()

Sleep\_ms(self.CYPHER\_SPACE)

Self.key.off()

Sleep\_ms(self.LETTER\_SPACE)

The lines self.key.on() and self.key.off() turn pin 13 high (3 volts) and low (0 volts) respectively.

Since pin 13 is connected to the base of the transistor, 3 volts turn the transistor fully on, as if the collector was connected directly to ground. Turning off pin 13 disconnects the collector, turning off whatever the mono plug controls.

Big 1,500-watt transmitters are expensive and heat the room. Let’s start with something affordable that can still reach over a hundred miles using a cheap wire antenna (in my case, an end-fed half-wave antenna for 40 meters for $7.05 at AliExpress.com).

The Pixie 2 transceiver is easy to find with a Google search. You can get it in kit form for $3.55 on AliExpress.com, or fully built and ready to use in a transparent acrylic case for $12.88 on eBay.

Press enter or click to view image in full size

The RP2040 connected to a Pixie 2 transceiver.

Photo by author

That’s it. You are on the air for about $30.

Our main.py module is simple:

From cwmorse import CWMorse

From time import sleep

Def main():

Cw = CWMorse(13)

Cw.speed(10)

Print(“CW keyer”)

Msg = “AB6NY testing RP2040 as a CW keyer.”

While True:

Print(msg)

Cw.send(msg)

Sleep(5)

Main()

This sets up a beacon for testing the range of your transmitter. Now you can use a good receiver and another EFHW (end-fed half-wave) antenna as you drive around (tossing one end of the antenna wire up into trees).

The module the\_code.py looks like this:

Code = {

‘A’: ‘.-‘,

‘B’: ‘-...’,

‘C’: ‘-.-.’,

‘D’: ‘-..’,

‘E’: ‘.’,

‘F’: ‘..-.’,

‘G’: ‘--.’,

‘H’: ‘....’,

‘I’: ‘..’,

‘J’: ‘.---‘,

‘K’: ‘-.-‘,

‘L’: ‘.-..’,

‘M’: ‘—‘,

‘N’: ‘-.’,

‘O’: ‘---‘,

‘P’: ‘.--.’,

‘Q’: ‘--.-‘,

‘R’: ‘.-.’,

‘S’: ‘...’,

‘T’: ‘-‘,

‘U’: ‘..-‘,

‘V’: ‘...-‘,

‘W’: ‘.—‘,

‘X’: ‘-..-‘,

‘Y’: ‘-.—‘,

‘Z’: ‘--..’,

‘0’: ‘-----‘,

‘1’: ‘.----‘,

‘2’: ‘..---‘,

‘3’: ‘...—‘,

‘4’: ‘....-‘,

‘5’: ‘.....’,

‘6’: ‘-....’,

‘7’: ‘--...’,

‘8’: ‘---..’,

‘9’: ‘----.’,

‘.’: ‘.-.-.-‘,

‘,’: ‘--..—‘,

‘?’: ‘..--..’,

‘\’’: ‘.----.’,

‘!’: ‘-.-.—‘,

‘/’: ‘-..-.’,

‘(‘: ‘-.--.’,

‘)’: ‘-.--.-‘,

‘&’: ‘.-...’,

‘:’: ‘---...’,

‘;’: ‘-.-.-.’,

‘=’: ‘-...-‘,

‘+’: ‘.-.-.’,

‘-‘: ‘-....-‘,

‘\_’: ‘..--.-‘,

‘”’: ‘.-..-.’,

‘$’: ‘...-..-‘,

‘@’: ‘.--.-.’,

}

An excellent walk-through of the Pixie 2 transceiver is here.

If the 1.2 watts of the Pixie 2 is not enough, you can go for 5 watts:

Press enter or click to view image in full size

The NS-40+ QRP transmitter.

Image by author

For $30 you can get the NS-40+ transmitter kit. The NS stands for “None Simpler” because all of the coils in the circuit are printed right on the circuit board. There are only 16 parts to solder. The kit goes together in minutes.

Connect to the same EFHW antenna and as much as 12 volts (shown above using a 9-volt battery, which works fine and gets over 3 watts out). That’s enough power to get anywhere in the world with good antennas.

You can buy a 1,500-watt amplifier to boost either of these transmitters to the full legal limit. But why, when you can already work the whole world?

None of the links in this article are affiliate links. I only make money when you clap for this article. ☺

Radio