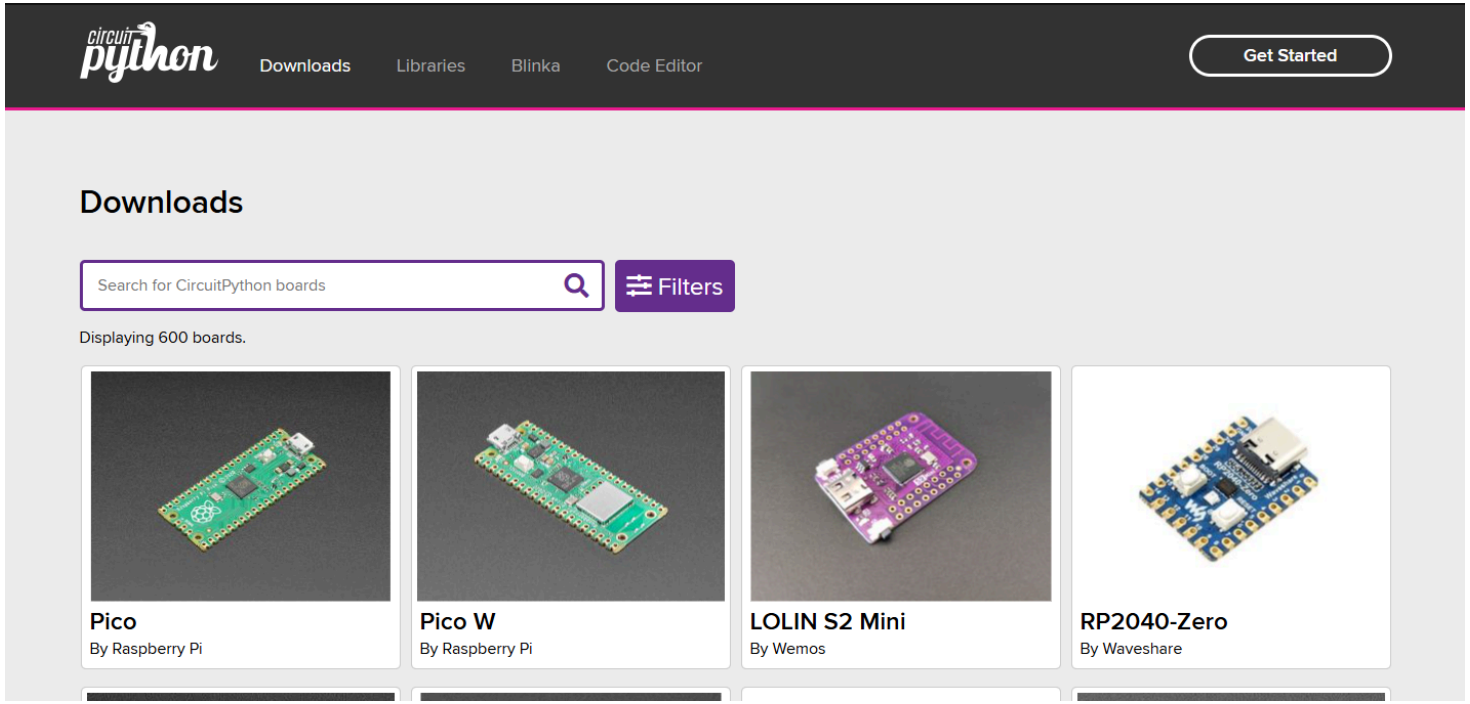
Czym jest CircuitPython?

Jest to fork MicroPythona. Składa się z kompilatora Pythona do bytecodu i interpretera, dzięki któremu możemy uruchamiać programy napisane w Pythonie na mikrokontrolerach. Dodatkowo udostępnia też CoreModules. W nich znajdziemy moduły przydatne do komunikacji z naszą płytką. Nie każdy moduł jest dostępny na każdym mikrokontrolerze, możemy sprawdzić to w tej tabelce. Może to wynikach z braku odpowiednich zasobów sprzętowych, czy z tego że jeszcze komuś nie chciało się tego zrobić 😉

Czy mogę w nim pisać Pythona jak na komputerze?

Składnia jest taka sama. Ale CircuitPython (jak na razie, mają w planach 😜) implementuje tylko pewien podzbiór biblioteki standardowej Pythona (CPython). Tutaj lista.

Dobra, ale jak go zainstalować?



Czy wspiera naszą płytkę?

Na tej stronie możemy sprawdzić czy nasza płytka jest wspierana przez CircuitPythona.

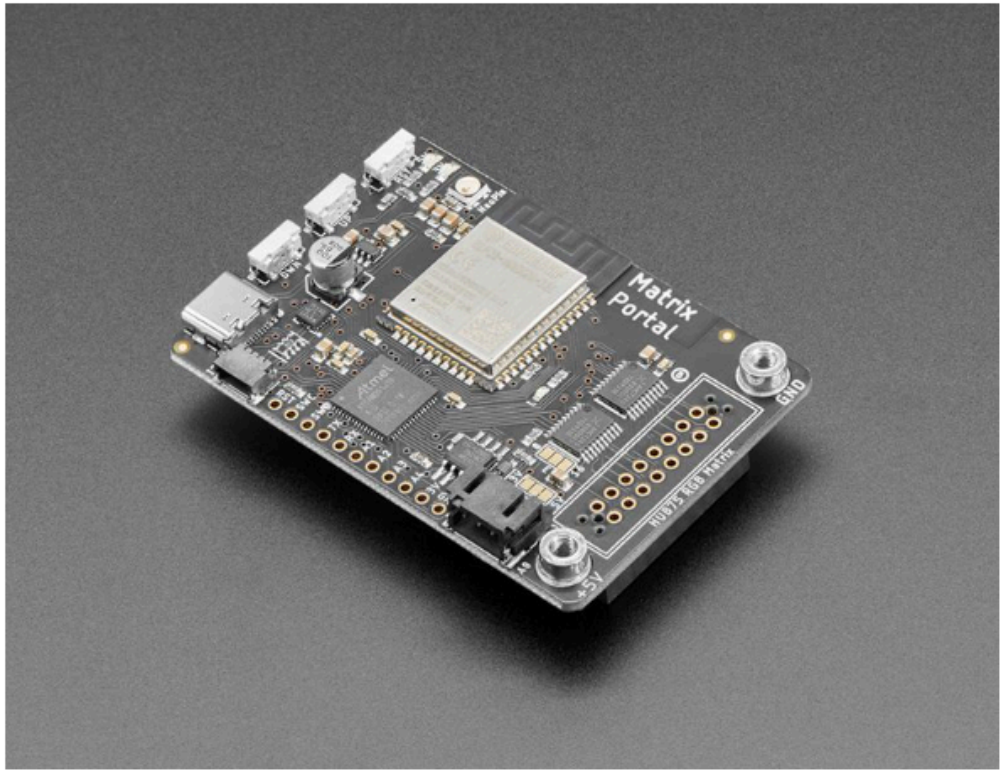
Pobieramy!

Gdy już znajdziemy nasz mikrokontroler, to musimy pobrać odpowiednią wersje i zainstalować na naszym sprzęcie.

Przykład dla Matrix Portal M4

MatrixPortal M4

by Adafruit



CircuitPython 9.2.5

This is the latest **stable** release of CircuitPython that will work with the MatrixPortal M4.

Use this release if you are new to CircuitPython.

Release Notes for 9.2.5

HOW TO INSTALL

POLISH

DOWNLOAD .UF2 NOW

Built-in modules available: _asyncio, _bleio, _pixelmap, adafruit_bus_device, adafruit_pixelbuf, alarm, analogio, array, atexit, audiobusio, audiocore, audioio, audiomixer, audiomp3, binascii, bitbangio, bitmaptools, board, builtins, builtins.pow3, busdisplay, busio, busio.SPI, busio.UART, codeop, collections, countio, digitalio, displayio, epaperdisplay, errno, fontio, fourwire, framebufferio, frequencyio, getpass, gifio, i2cdisplaybus, i2ctarget, io, jpegio, json, keypad, keypad.KeyMatrix, keypad.Keys, keypad.ShiftRegisterKeys, locale, math, microcontroller, msgpack, neopixel_write, nvm, onewireio, os, os.getenv, ps2io, pulseio, pwmio, rainbowio, random, re, rgbmatrix, rotaryio, rtc, samd, sdcardio, select, spitarget, storage, struct, supervisor, synthio, sys, terminalio, tilepalettemapper, time, touchio, traceback, usb_cdc, usb_hid, usb_midi, vectorio, warnings, watchdog, zlib

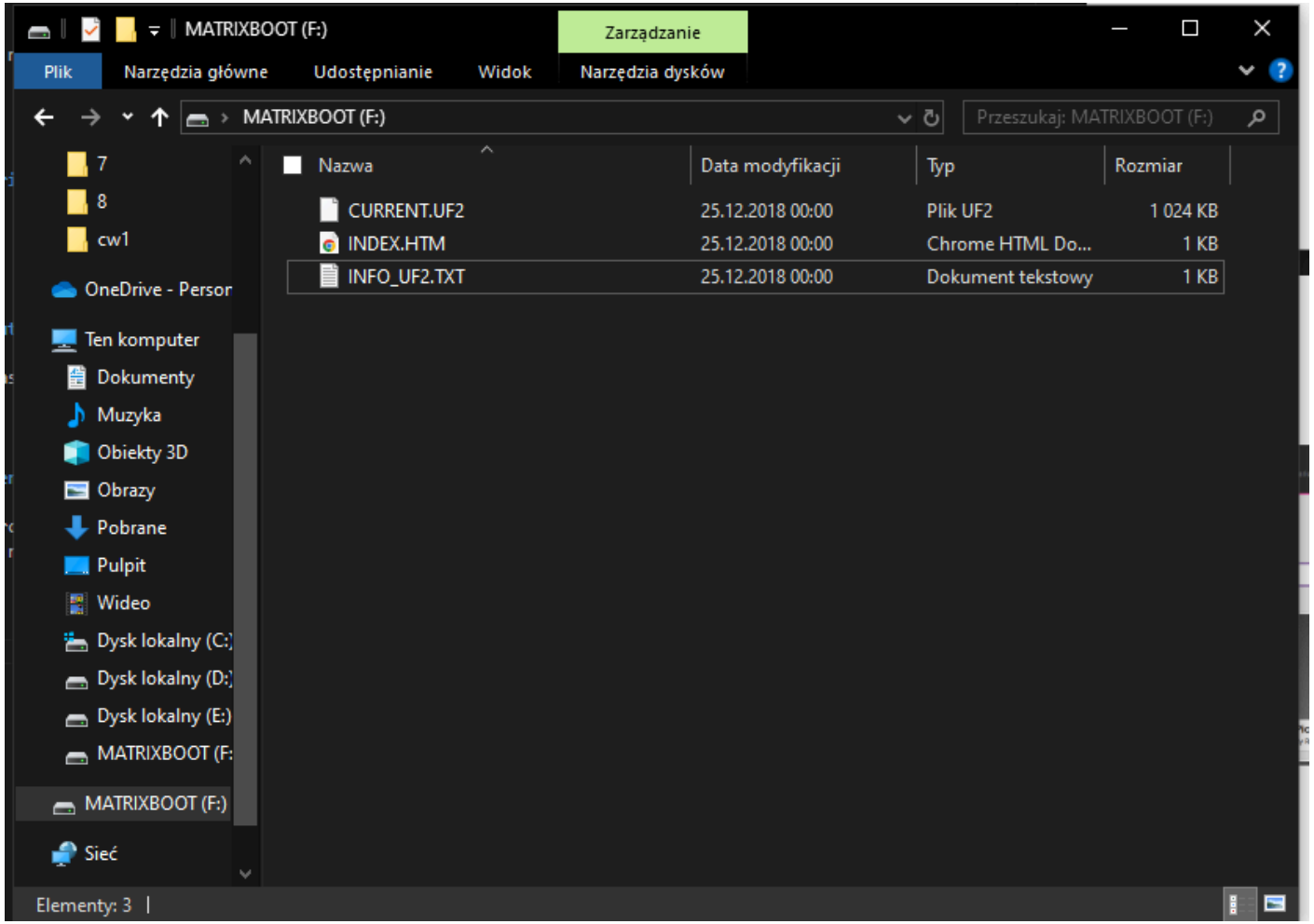
Podepnij do komputera

Upewnij się, czy masz dobry kabel 😆. Musi on przesyłać dane nie tylko zasilać.

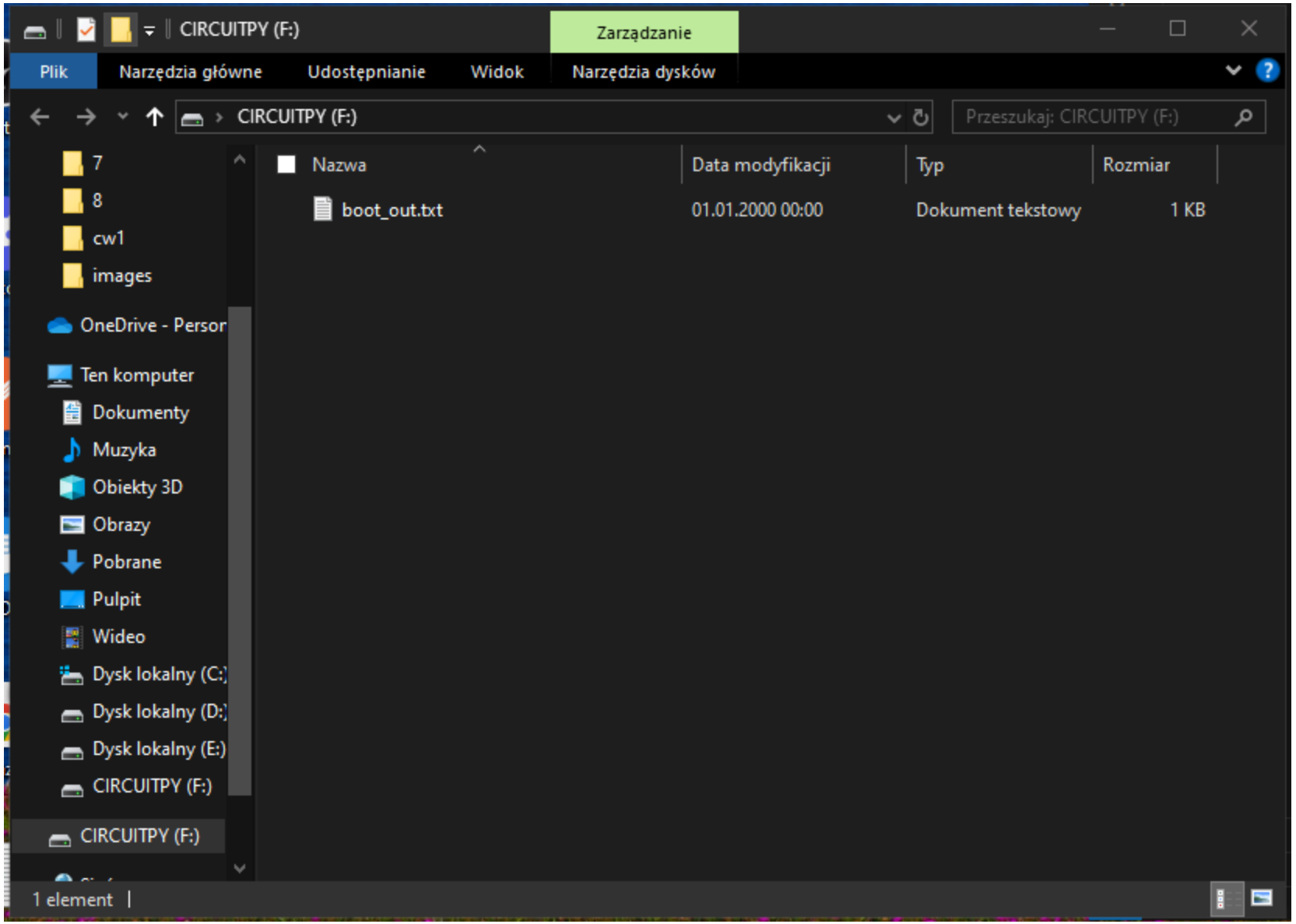
Uruchom Bootloader

By zainstalowac CircuitPython na naszym sprzęcie, musimy dostać się do Bootloadera.
W moim przypadku trzeba kliknąc dwa razy przycisk Reset na mikrokontrolerze.

Urządzenie powinno się pojawić jako dysk

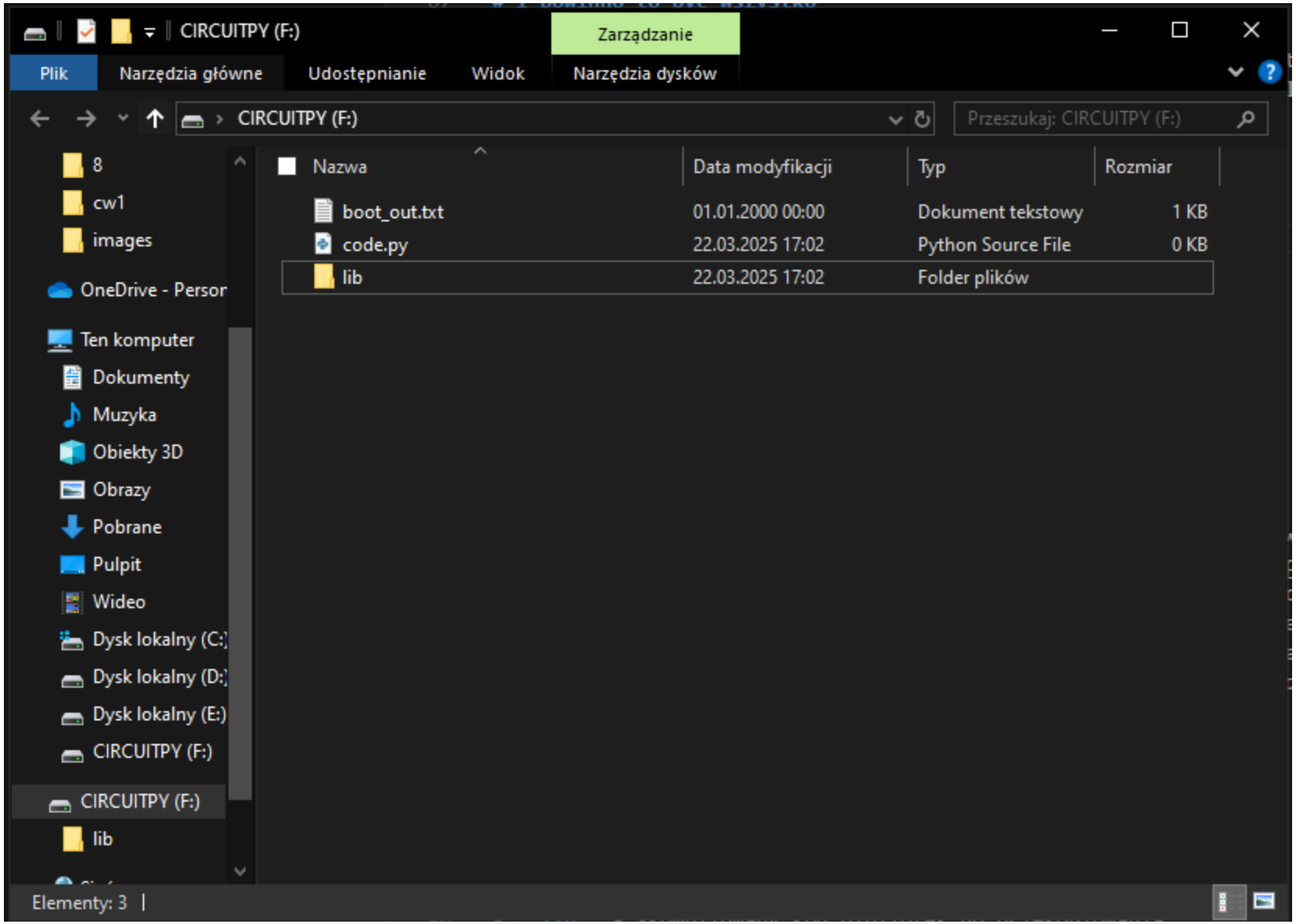


Przeciągnij pobrany plik


CIRCUITPY (F:)
Zarządzanie
Plik
Narzędzia główne
Udostępnianie
Widok
Narzędzia dysków
CIRCUITPY (F:)
Przeszukaj: CIRCUITPY (F:)
7
8
cw1
images
OneDrive - Persor
Ten komputer
Dokumenty
Muzyka
Obiekty 3D
Obrazy
Pobrane
Pulpit
Wideo
Dysk lokalny (C:)
Dysk lokalny (D:)
Dysk lokalny (E:)
CIRCUITPY (F:)
CIRCUITPY (F:)
Nazwa
Data modyfikacji
Typ
Rozmiar
boot_out.txt
01.01.2000 00:00
Dokument tekstowy
1 KB
1 element

I powinno to być wszystko

Polecam usunąć wszystko z dysku CIRCUITPY, bo mogły tam pozostać stare pliki. Po usunięciu wystarczy stworzyć plik code.py (jest to plik startowy dla naszego programu) oraz folder lib, gdzie będziemy wrzucać biblioteki.


CIRCUITPY (F:)
Zarządzanie
Plik
Narzędzia główne
Udostępnianie
Widok
Narzędzia dysków
CIRCUITPY (F:)
Przeszukaj: CIRCUITPY (F:)
8
cw1
images
OneDrive - Persor
Ten komputer
Dokumenty
Muzyka
Obiekty 3D
Obrazy
Pobrane
Pulpit
Wideo
Dysk lokalny (C:)
Dysk lokalny (D:)
Dysk lokalny (E:)
CIRCUITPY (F:)
CIRCUITPY (F:)
lib
Nazwa
Data modyfikacji
Typ
Rozmiar
boot_out.txt
01.01.2000 00:00
Dokument tekstowy
1 KB
code.py
22.03.2025 17:02
Python Source File
0 KB
lib
22.03.2025 17:02
Folder plików
Elementy: 3

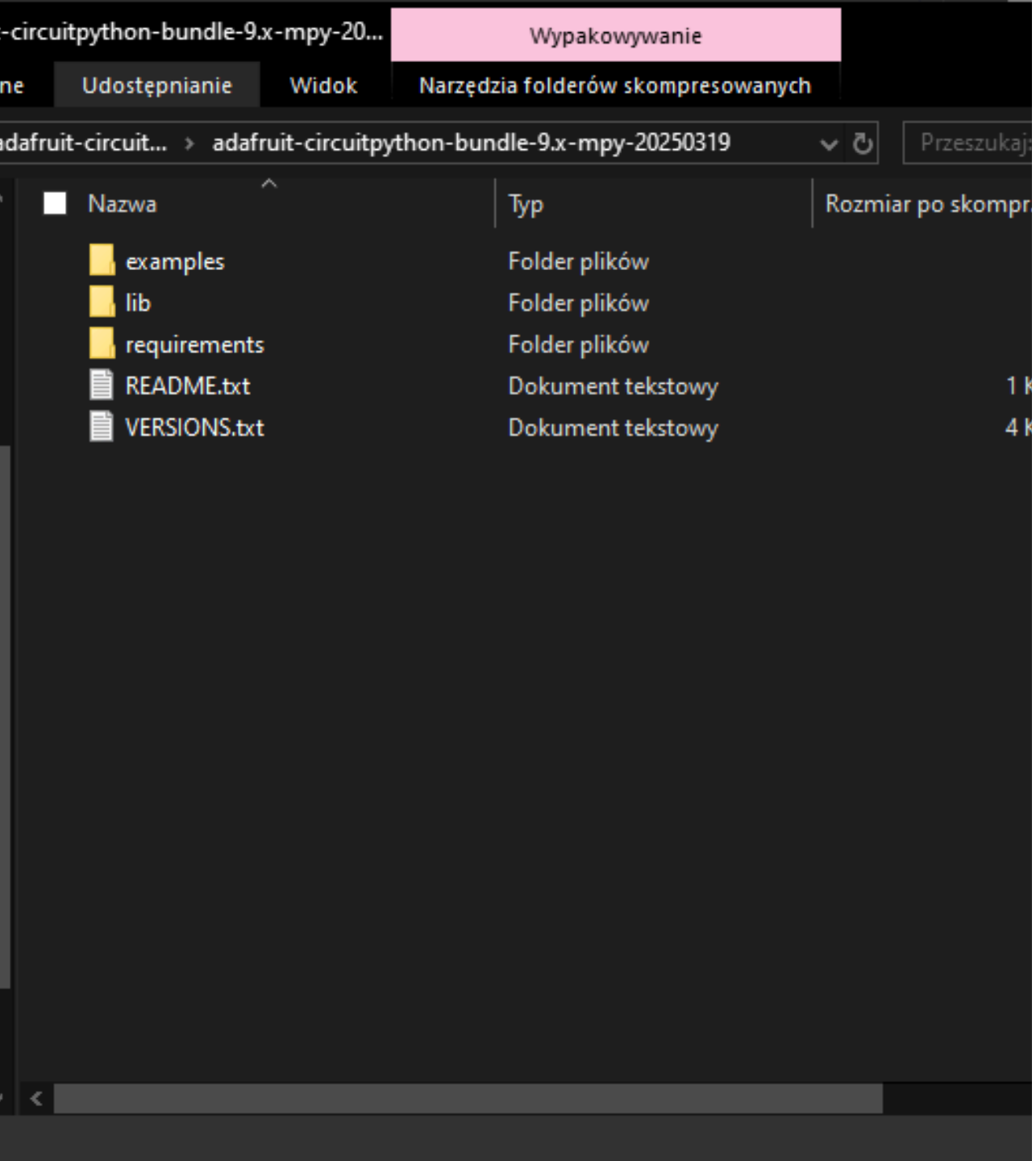
Mamy już CircuitPythona 🎉

Gdzie pisać kod?

Po zapisaniu pliku na dysku CIRCUITPY, powoduje przeładowanie, więc możemy pisać nawet w notatniku. Ale to nie jest wygodne, nie widzimy np. wyjścia naszego programu. Moglibyśmy go zobaczyć ale musimy się połączyć np. przez Putty. Możesz poczytać o tym tutaj. Ja skorzystam z *vscode* z wtyczką CircuitPython v2. Tutaj możemy poczytać o innych edytorach.

Biblioteki

Poza CoreModules i standarodwej bilioteki, czasami możemy potrzebować coś więcej, by ułatwić, przyspieszyć sobie prace. Na stronie CircuitPython mamy paczki biliotek oficjalnie wspieranych przez Adafruit oraz wersje community. Wystarczy skopiować wybraną biliotekę z folderu /lib paczki bibliotek do folderu /lib naszego urządzenia. Czasami jakaś biblioteka korzysta z innej i musimy ją wtedy też dorzucić. Możemy się o tym dowiedzieć z folderu requirements.



- **examples** - przykładowe użycia bibliotek
- **lib** - skompilowany kod bibliotek do przekopiowania
- **requirements** - folder z zależnościami dla bibliotek

Więcej o CircuitPythonie

- Welcome to CircuitPython! - wprowadzenie do CircuitPythona
- CircuitPython Essentials - mini kurs CircuitPythona
- Dokumentacja CircuitPythona
- Strona CircuitPythona