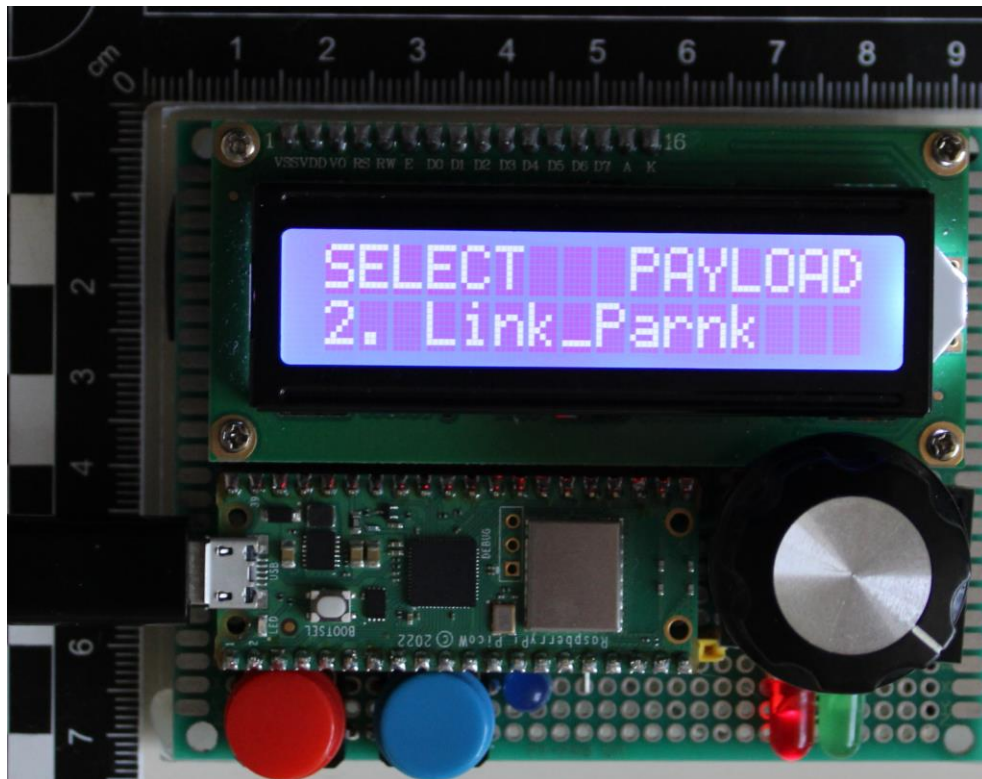


Raspberry Pi Pico W



Ducky



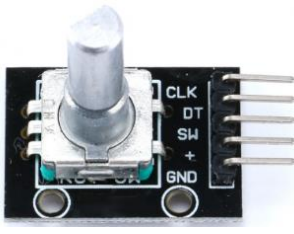

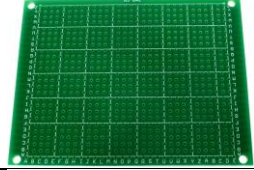

An educational project



A prototype of Ducky Pi Pico W project

Electronic component used

Raspberry pi pico w		
LCD 16X2 – Blue, I2C connection		LCD to Pico W connection LCD SDA ----→GP16 LCD SCL-----→GP17 Vin ----→3.3v Gnd ----→GGND

Push buttons with round caps		Buttons to Pico W connection Red button ----→GP0 Blue Button ----→GP9
Leds		Leds to Pico W connection Red led ----→GP13 Green led ----→GP11 Blue led ----→GP10 Gnd ----→GND
Rotary encoder		Rotary encoder to Pico W connection Clk ----→GP5 DT ----→GP6 Sw ----→GP12 Vin ----→3.3v Gnd ----→GND
Rotary encoder cap		
Proto board 90x70 mm		
Micro USB data cable		PC/Smatphone/tablet to Pico W connection

This educational project began from a cyber-security course on my University, when my professor told us very very fast about BAD USB. So I started looking about what is a bad USB, and WEB showed me examples about PI PICO W, that was very affordable. So, from information gathered, ideas start to evolve in a semi-complex project, very fun and very scary for my office colleagues☺.

My Ducky Pi Pico W has 3 (three) states of functionality

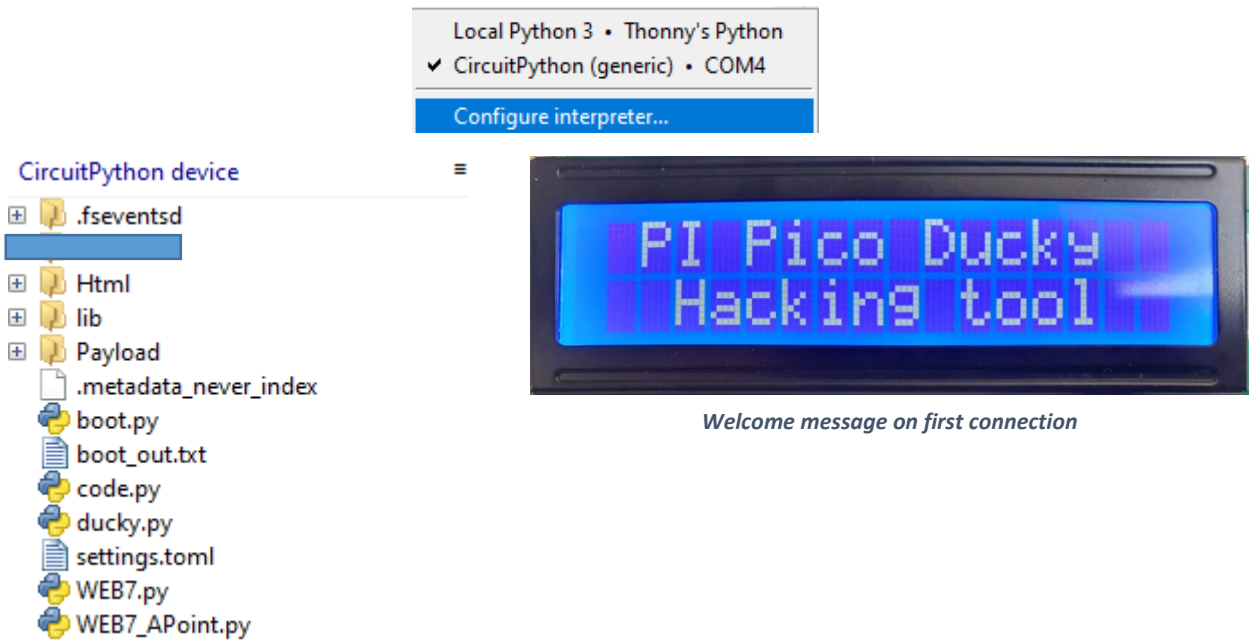
1. Programing mode – Simple connection with a usb micro USB cable to PC. In this mode Raspberry pi pico W is shown like a USB thumb drive in Computer O.S. and the RED led is blinking fast.

Thonny Python IDE screen shots then Raspberry pi pico W

```
0; 192.168.0.120 | REPL | 8.0.5\
```

```
Adafruit CircuitPython 8.0.5 on 2023-03-31; Raspberry Pi Pico W with rp2040
```

```
>>>
```



In this mode when we rotate the rotary encoder, on LDC 16x2 will be printed the name of payload and if we push the rotary, the payload will be executed

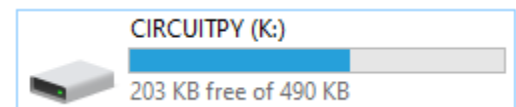
```

USB drive enabled
2 Link_Parnk
3 Notepad
2 Link_Parnk
3 Notepad
4 PSW
5 PwrShell
6 Scrip_1
7 While_Note
8 Wi_Fi_arch
9 You_tube_OCS

```

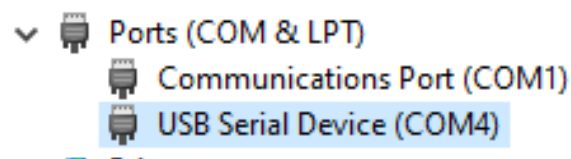


Name of payload printed on LDC 16x2



Thonny showing the name of Payload when rotate the encoder

2. HID (Human Interface Device) mode. To enter in this state the RED button must be pressed and after that cable micro USB will be connected to microcontroller, the Pico emulate a keyboard, and will not be showed like USB thumb drive and the GREEN led will blink slowly. Computer OS will see this device like USB SERIAL DEVICE.



Mouse, keyboard, & pen

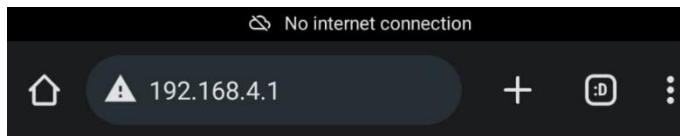
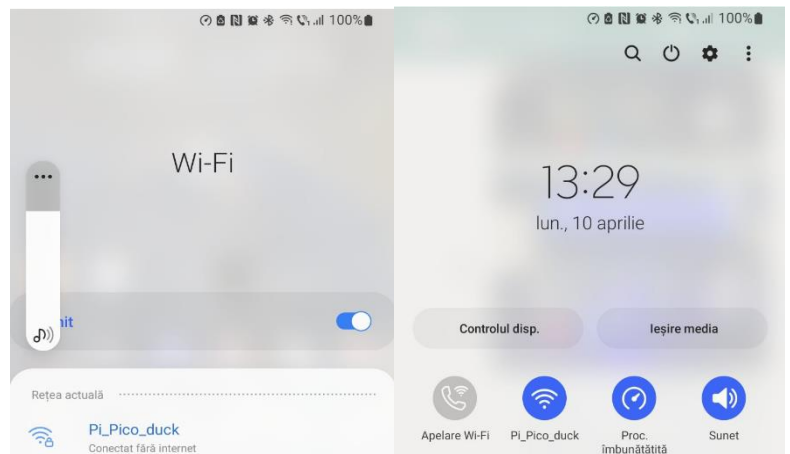
Game mouse

Pico W

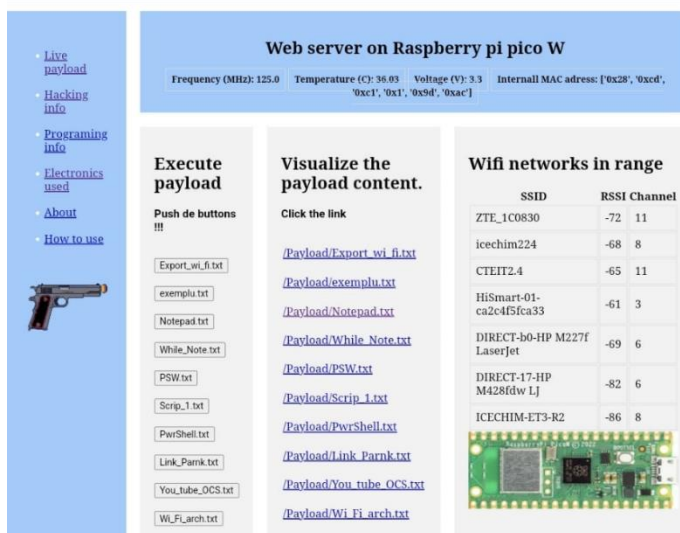
Remove device

USB Keyboard

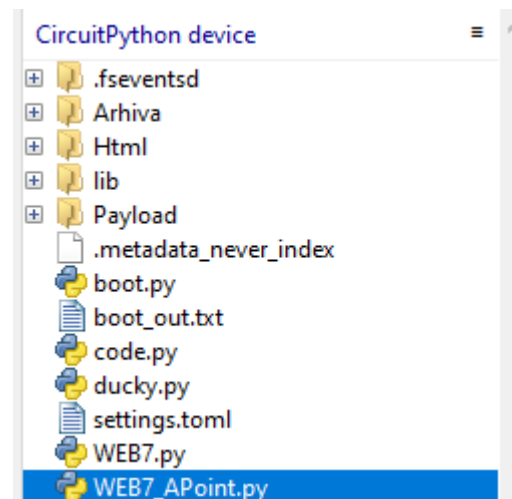
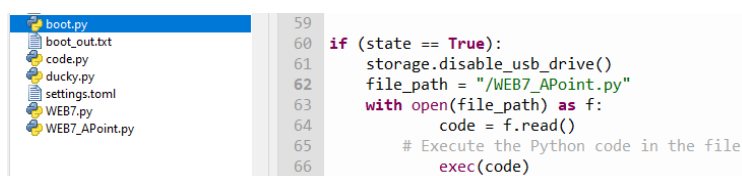
3. A.P.(Access Point) mode. To enter in this mode we must press the blue button before connect the device to de computer. The display will print the IP of the access point and the BLUE led will be always on. For acces point we must at least wait 10 seconds with the blue button pressed.



After inserting of IP in the browser the web page will be load.

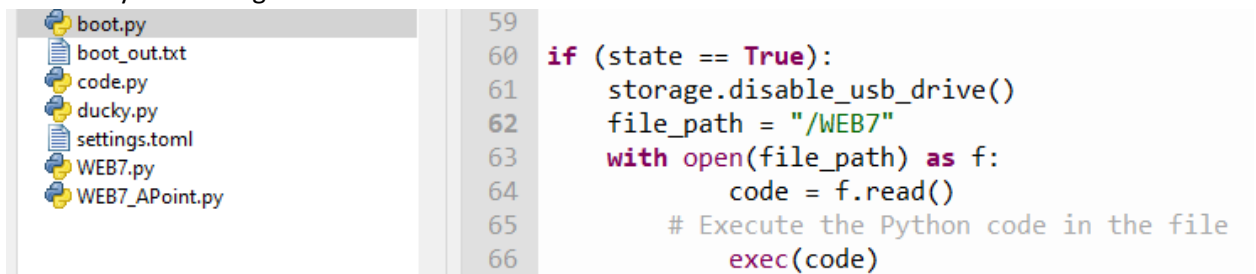


When payload was executed a message will be print on LCD 16x2

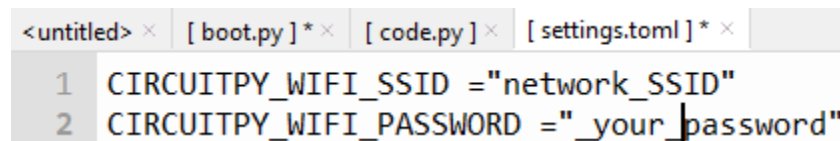


For access point to work the file WEB7_APoint.py must be executed in boot.py file on line 62. So we modify that line.

IF we want Raspberry pi Pico W to connect directly to a WI-Fi network or to Smartphone Hot Spot we must to modify 2 (two) files. A. the boot.py file on line 62 we must execute WEB7.py file, but before that we modify the settings.toml file with credentials of the network



```
59
60 if (state == True):
61     storage.disable_usb_drive()
62     file_path = "/WEB7"
63     with open(file_path) as f:
64         code = f.read()
65         # Execute the Python code in the file
66         exec(code)
```

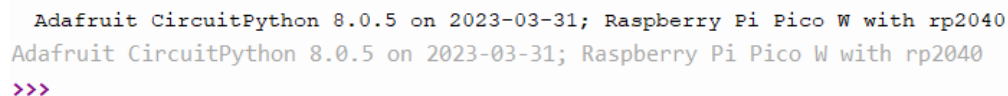


```
<untitled> × [ boot.py ] × [ code.py ] × [ settings.toml ] ×
1 CIRCUITPY_WIFI_SSID = "network_SSID"
2 CIRCUITPY_WIFI_PASSWORD = "_your_password"
```

Chapter II

So we have a good idea about how to use Pi Pico W educational ducky. Now lets make a resume about file structure.

- A. First of all raspberry pi Pico w must be flashed with Circuit Python version 8.0.5



```
Adafruit CircuitPython 8.0.5 on 2023-03-31; Raspberry Pi Pico W with rp2040
>>>
```

B. **Html** – Folder – where web pages are saved

Lib – Folder – libraries from Adafruit

Payload Folder – payloads Ducky to run

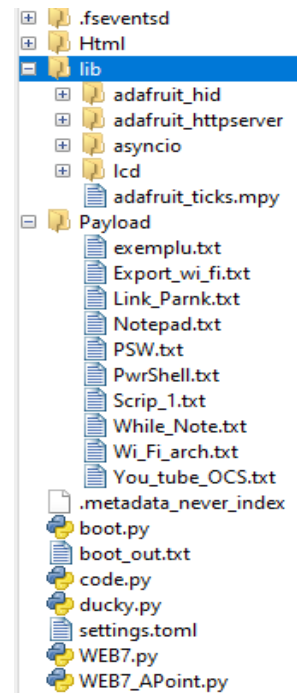
Boot.py – first file to execute when data cable is connected

Code.py second file to run end de execution of HID

Ducky.py conversion of Ducky payload to run on PC

Web7.py – Web page and direct connection wi-fi

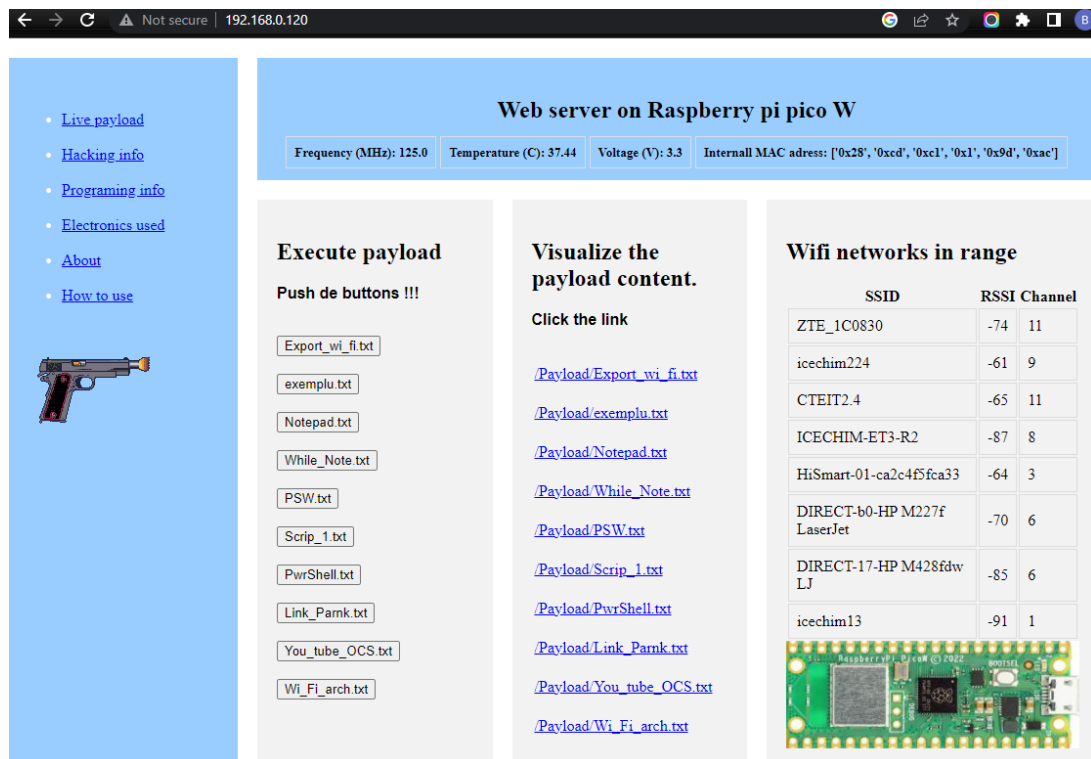
Web7_APoint.py Web page power on the Acces Point



Chapter III


I tried to make as user friendly as can, this is...

First page



Second page

- [HOME](#)
- [Live payload](#)
- [Hacking info](#)
- [Programing info](#)
- [Electronics used](#)
- [About](#)
- [How to use](#)



Create payload and run them live on target

Script

Run Script

Example

```
REM A slightly more
advanced "Hello, World!"
for Windows
DELAY 3000
REM Open the Run dialog
WINDOWS r
DELAY 1000
REM Open powershell with
our message
STRING powershell "echo
'Hello, World!'; pause"
ENTER
```

The commands in Ducky Script are usually short and simple, and they can be combined to create more complex scripts. Some common commands include:

DELAY: Pauses the script for a specified amount of time (in milliseconds).

STRING: Types a string of

On this page on the textarea element (https://www.w3schools.com/tags/tag_textarea.asp), the first line of ducky script will be the name of payload file

Exemple:

Lacrimosa

REM Title: Mozart - Lacrimosa

REM Author: krux

REM Description: start Mozart in browser.

REM Target: Windows 10 & 11

REM Version: 1.0

REM Category: Prank

REM -----

GUI R

DELAY 200

STRING https://www.youtube.com/watch?v=k1-TrAvp_xs&ab_channel=RosaMusic

DELAY 200

ENTER

```
Script
Lacrimosa
REM Title: Mozart - Lacrimosa
REM Author: krux
REM Description: start Mozart in browser.
REM Target: Windows 10 & 11
REM Version: 1.0
REM Category: Prank
REM -----
GUI R
DELAY 200
STRING https://www.youtube.com/watch?v=k1-TrAvp_xs&ab_channel=RosaMusic
DELAY 200
ENTER
```

Run Script

Execute payload

Push de buttons !!!

Export_wi-fi.txt

exemplu.txt

Notepad.txt

While_Note.txt

PSW.txt

Scrip_1.txt

Lacrimosa.txt

PwrShell.txt

Link_Parnk.txt

You_tube_OCS.txt

Wi-Fi_arch.txt

After running the script in the first page we will see a button with the name of first line of the script, hit run and the pico will execute the payload. Live payloads will always be automatically saved in internal folder /Payload of pico. Be **aware** of the first line, that will be the name of payload file.

Third page

Informational page

- [HOME](#)
- [Live payload](#)
- [Hacking info](#)
- [Programing info](#)
- [Electronics used](#)
- [About](#)
- [How to use](#)

Information about programing languages used

Circuit Python

CircuitPython is a programming language designed for beginners and experts alike to program microcontrollers easily and efficiently. It is a derivative of Python 3 that was created specifically for use with microcontrollers, and is gaining popularity due to its simplicity and ease of use.

CircuitPython is a powerful tool for creating interactive electronics projects, allowing users to quickly prototype and develop new ideas. It is built on top of the low-level C programming language, which means it can interact with hardware at a much lower level than other

H.T.M.L.

HTML, or Hypertext Markup Language, is a coding language used for creating websites and web applications. It is the standard markup language for creating web pages and documents on the internet, and is essential for creating and designing visually appealing and user-friendly websites.

HTML works by using tags and attributes to define the structure and content of a web page. Tags are enclosed in angle brackets, and provide information about the type of content being displayed, such as headings, paragraphs, images, and links. Attributes are used to provide additional

DuckScript 1.0

DuckScript 1.0 is a scripting language used by the Rubber Ducky, a USB device that can be used for both legitimate and malicious purposes. From a cybersecurity perspective, DuckScript 1.0 presents both opportunities and risks.

On one hand, DuckScript 1.0 can be used as a powerful tool for automating legitimate tasks and streamlining workflows. For example, it can be used to automate routine system administration tasks, simplify software installations, and even perform basic security assessments. In this context, DuckScript 1.0 can be a useful tool for system

Forth page

Informational page

- [HOME](#)
- [Live payload](#)
- [Hacking info](#)
- [Programing info](#)
- [Electronics used](#)
- [About](#)
- [How to use](#)

Information that is permissible for you to access

Ruber Ducky

Rubber Ducky is a small USB device that looks like a normal thumb drive but is actually a powerful tool used for penetration testing and security assessments. When plugged into a computer, Rubber Ducky can automatically inject pre-configured keystrokes and commands that can exploit vulnerabilities in the system, bypass security controls, and gain access to sensitive information.

However, from a blue team perspective, Rubber Ducky can also be used as a powerful tool for testing and improving the security of a system. By emulating the actions of a hacker, security teams can use

Payload

A payload is a piece of code that is executed on a system after a successful exploitation. In the context of Rubber Ducky, a payload is a set of commands and keystrokes that are injected into a target system to achieve a specific objective. For example, a payload could be used to steal login credentials, install malware, or gain access to sensitive data.

As a blue team or white-hat hacker, understanding how payloads work is crucial for detecting and responding to potential threats. By analyzing payloads, security professionals can identify the tactics, techniques, and procedures (TTPs) used by

Bad USB

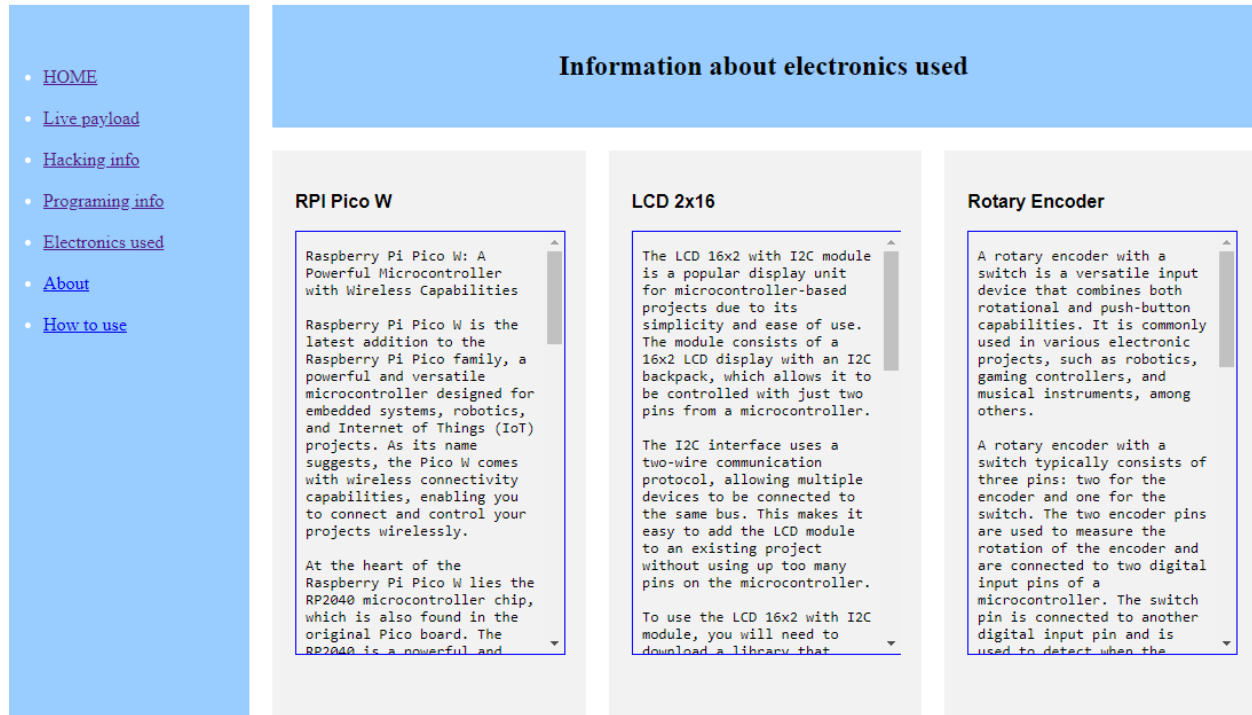
A Bad USB, also known as a malicious USB device, is a type of cyberattack that involves using a USB device to inject malware or other malicious code into a victim's computer or network. These attacks can be devastating, as they allow attackers to gain unauthorized access to sensitive data, steal passwords, and even take control of entire systems.

To protect against Bad USB attacks, there are a number of strategies that individuals and organizations can employ. Some of the most effective include:

Limiting USB access: One of

Fihth page

Informational page



Chapter IV

The Code

I tried to comment the code as much I can , like a reminder for what the duck I what to make and to be a remember, I hope that I clean all the comments from my maternal language, Romanian.

```

===== Import Library=====
import socketpool
import wifi
import microcontroller
import os
import digitalio
import time
import board
import storage

===== Import Web server Library=====
from adafruit_httpserver.mime_type import MIMEType
from adafruit_httpserver.request import HTTPRequest
from adafruit_httpserver.response import HTTPResponse
from adafruit_httpserver.server import HTTPServer
from adafruit_httpserver.methods import HTTPMethod

===== Import i2c and LCD 16x2=====
import busio
from lcd.lcd import LCD
from lcd.i2c_pcf8574_interface import I2CPCF8574Interface

===== Parsing the Payload code=====
from ducky import *

=====WEB APP=====
import json
import binascii
=====

```

Chapter V

The inspiration

1. <https://github.com/dbisu/pico-ducky>
2. <https://github.com/TheR1D/pico-ducky-ui>
3. <https://github.com/TheSavageTeddy/wirelessRubberDuckyPicoW>