Project Report: USB Rubber Ducky using Raspberry Pi Pico

1. Introduction

This project demonstrates how to create a **USB Rubber Ducky** using a **Raspberry Pi Pico**. The device acts as a Human Interface Device (HID) that mimics a keyboard and executes pre-defined scripts (payloads) when plugged into a computer. It is widely used for penetration testing, task automation, and testing keyboard-based exploits.

2. Objectives

- Create a USB Rubber Ducky using Raspberry Pi Pico.
- Load and execute Ducky Script 1.0 payloads.
- Implement **Setup Mode** to edit payloads without executing them.
- Enable/disable USB mass storage mode for stealth.
- Add functionality for multiple payloads using GPIO pins.
- Provide multiple payload storage and selection.
- Securely store and modify Ducky scripts.

3. Required Components

- Raspberry Pi Pico (Not Pico W or Pico 2)
- Micro USB cable
- Jumper wires
- Host computer (Windows, Linux, or Mac)
- MicroSD card (optional for future payload storage expansion)
- Push buttons (optional for payload selection)

4. Software and Libraries

- **CircuitPython 9.2.1** for Raspberry Pi Pico
- **Adafruit HID Library**

- **Adafruit CircuitPython Bundle**
- Required Python files:
  - `boot.py`
  - `code.py`
  - `duckyinpython.py`

5. Circuit Setup

5.1 Pin Configuration

- **Pin 1 (GP0)** – Setup mode activation
- **Pin 3 (GND)** – Ground for setup mode
- **Pin 18 (GND)** – USB mass storage disable
- **Pin 20 (GPIO15)** – USB mass storage disable
- **GP4, GP5, GP10, GP11** – Multiple payload selection

6. Circuit Diagram

- Connect a button between **GP0** and **GND** to enable setup mode.
- Connect another button between **GP18** and **GND** to enable/disable mass storage.
- Additional buttons can be connected to **GP4, GP5, GP10, GP11** for selecting multiple payloads.

7. Installation and Configuration

7.1 Step 1: Install CircuitPython

1. Download the latest **CircuitPython** release for Raspberry Pi Pico from CircuitPython Downloads.
2. Press and hold the **BOOTSEL** button while plugging the Pico into your computer.
3. Copy the `.uf2` file to the Pico, which will reboot as **CIRCUITPY**.

7.2 Step 2: Copy Required Files

1. Download the **pico-ducky repository** using:

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

2. Copy the following files to the root of the Pico:
    - `boot.py`
    - `code.py`
    - `duckyinpython.py`
3. Copy the `adafruit_hid` folder to the `lib` folder in **CIRCUITPY**.

7.3 Step 3: Load Payload Script

1. Create or download a **Ducky Script** payload.
2. Save the script as `payload.dd` in the root of the Pico.

7.4 Step 4: USB Enable/Disable Mode (Optional)

- Connect **GP18** to **GND** to disable USB mass storage mode.
- Remove the jumper to re-enable USB mass storage.

8. Working of Pico-Ducky

8.1 Entering Setup Mode

- Connect **GP0** to **GND** to stop payload execution and enter setup mode.
- This allows modification of payloads without executing them.
- While in setup mode, the `payload.dd` script can be edited.

8.2 Executing Payload

- Remove the jumper to exit setup mode.
- When plugged into a target computer, the Pico-Ducky executes the `payload.dd` script as a keyboard input.

8.3 Multiple Payloads with GPIO Selection

To select different payloads:

- **GP4** – Execute `payload1.dd`
- **GP5** – Execute `payload2.dd`
- **GP10** – Execute `payload3.dd`

- **GP11** – Execute `payload4.dd`

8.4 USB Mass Storage Enable/Disable

- **Default Mode:** USB mass storage is enabled.
- To disable:
  i. Enter setup mode.
  ii. Connect **GP18** to **GND**.
  iii. Unplug and re-plug the device.

9. Ducky Script Payload Structure

Basic Payload Example:

```
DELAY 500
STRING Hello, World!
ENTER
```

Advanced Payload Example:

```
DELAY 500
GUI r
DELAY 200
STRING cmd
ENTER
DELAY 500
STRING echo Pico-Ducky Activated!
ENTER
```

10. Security and Ethical Considerations

⚠️ **Warning:**

- Unauthorized use of USB Rubber Ducky devices may violate local laws and regulations.
- Use this device strictly for ethical and educational purposes.

11. Challenges and Solutions

Challenge 1: USB Mass Storage Disable

- **Problem:** Disabling USB storage for stealth.
- ✅ **Solution:** Use GPIO pins to toggle between enable/disable modes.

Challenge 2: Payload Modification Without Execution

- **Problem:** Editing payloads without automatic execution.
- ✅ **Solution:** Use setup mode with GPIO pins to pause execution.

Challenge 3: Multiple Payload Selection

- **Problem:** Managing multiple payloads dynamically.
- ✅ **Solution:** Configure multiple GPIO pins to select specific payloads.

12. Future Enhancements

- Support for Ducky Script 3.0.
- Integration of microSD for large payload storage.
- Web interface for managing and editing payloads.
- Encryption for secure payload management.
- Auto-detect target OS for customized payload execution.

13. Conclusion

The **Pico-Ducky** using **Raspberry Pi Pico** is a versatile USB Rubber Ducky tool that can execute predefined payloads, automate tasks, and simulate keyboard input. With features like **setup mode, multiple payload selection, and USB enable/disable functionality**, the Pico-Ducky is a valuable tool for penetration testing and automation tasks.

14. References

1. Adafruit Industries. (2024). *CircuitPython for Raspberry Pi Pico*. Available at: https://circuitpython.org/board/raspberry_pi_pico/

2. D. Biswas. (2022). *Pico-Ducky: A USB Rubber Ducky on Raspberry Pi Pico*. GitHub Repository. Available at: https://github.com/dbisu/pico-ducky

3. Hak5. (2023). *USB Rubber Ducky Documentation*. Available at: https://docs.hak5.org/usb-rubber-ducky

4. Raspberry Pi Foundation. (2023). *Raspberry Pi Pico Datasheet*. Available at: https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf