**Wi-Fi Duck**

|  |
| --- |
|  |
| Intro to Cyber securityLab manual |
| |  |  |  | | --- | --- | --- | | Muhammad Hayat | Submitted to: Sir Naveed Hussain | F2021408004 | |

## INTRODUCTION

The Wi-Fi Duck is a mini cyber security project. With it, you can plug the Wi-Fi Duck into a target computer that's exposed even for just a minute, then connect to it over Wi-Fi from another device to issue whatever payloads you have ready or can build before you have to disconnect.

The advantage is that you can connect to a slick web interface, save your own codes, and run them one by one, or write code on the fly to cause effects on the computer that you might not have intended before you knew what was on the computer. For example, if you didn't know the operating system. Or if there were other variables you couldn't account for before actually seeing the target computer.

**Components**

* Arduino Pro micro ATmega32U4
* ESP8266

**Software needed**

* Arduino IDE
* Python 3

## Assemble the Wi-Fi Duck

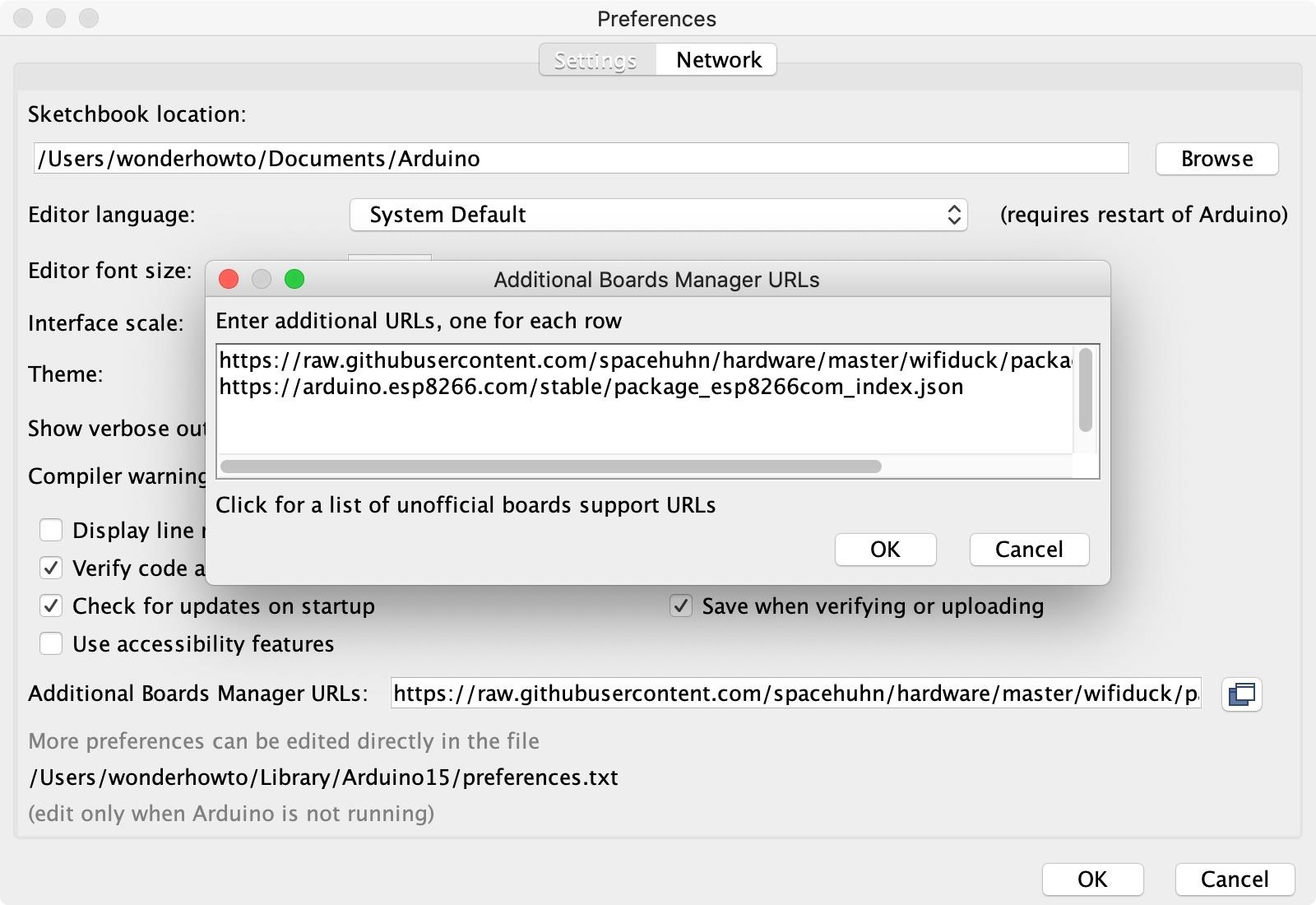
To connect the ESP8266 and ATmega32U4, we'll be working with a breadboard and jumper wires. So place each MCU on the breadboard, then use jumper wires to make the following pin connections.

* **D1** or **GPIO 5** (ESP8266) to **3** or **SCL** (ATmega32U4)
* **D2** or **GPIO 4** (ESP8266) to **2** or **SDA** (ATmega32U4)
* **GND** (ESP8266) to **GND** (ATmega32U4)
* **5V** (ESP8266) to **RAW** (ATmega32U4)

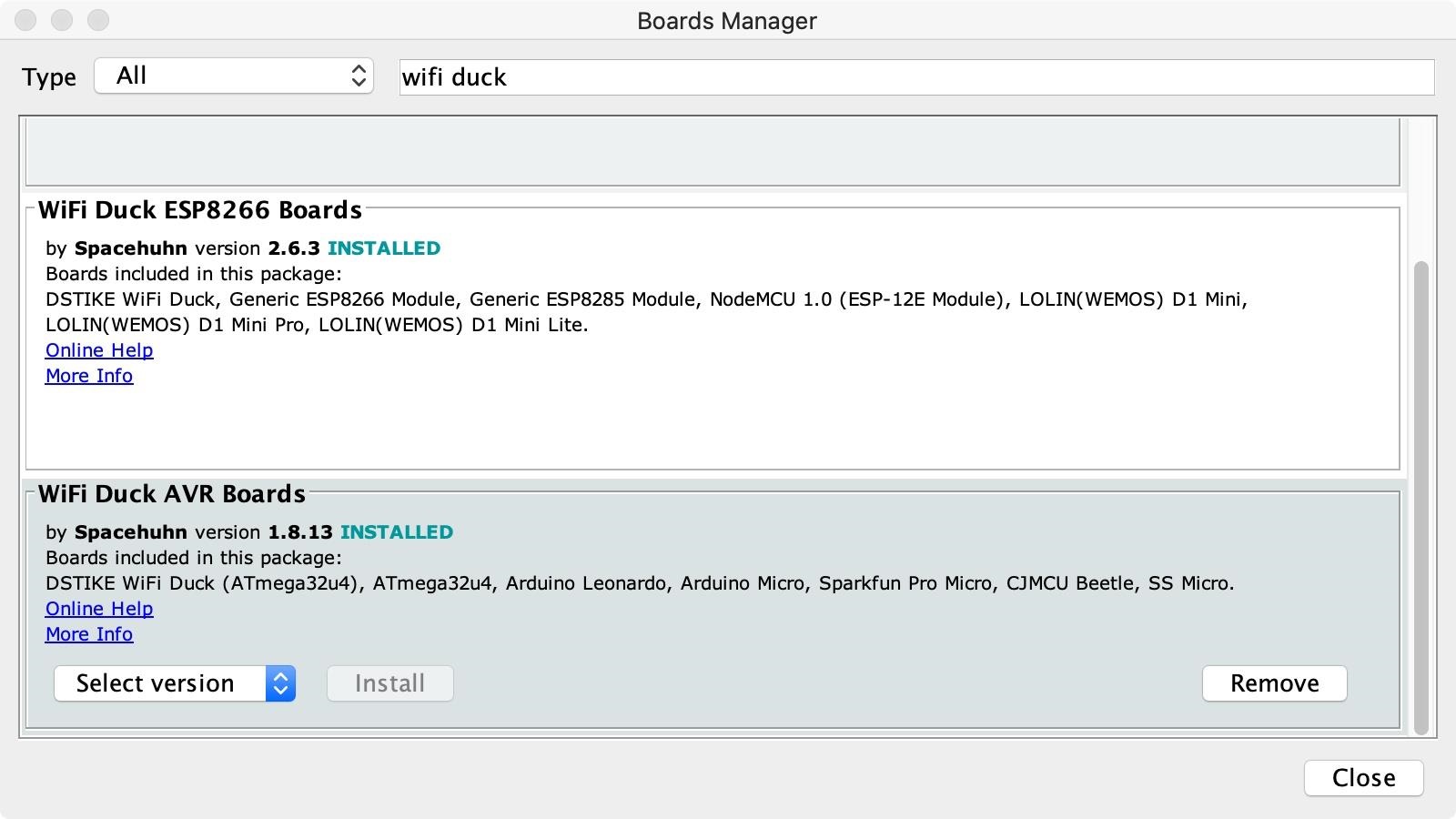
## Prepare the Arduino IDE

Next, we need to configure the Arduino IDE to work with both boards. Go to "Arduino" in the menu, then "Preferences." In the Additional Boards Manager URLs box, add the following two URLs, and click "OK."

* <https://raw.githubusercontent.com/spacehuhn/hardware/master/wifiduck/package_wifiduck_index.json>
* <https://arduino.esp8266.com/stable/package_esp8266com_index.json>

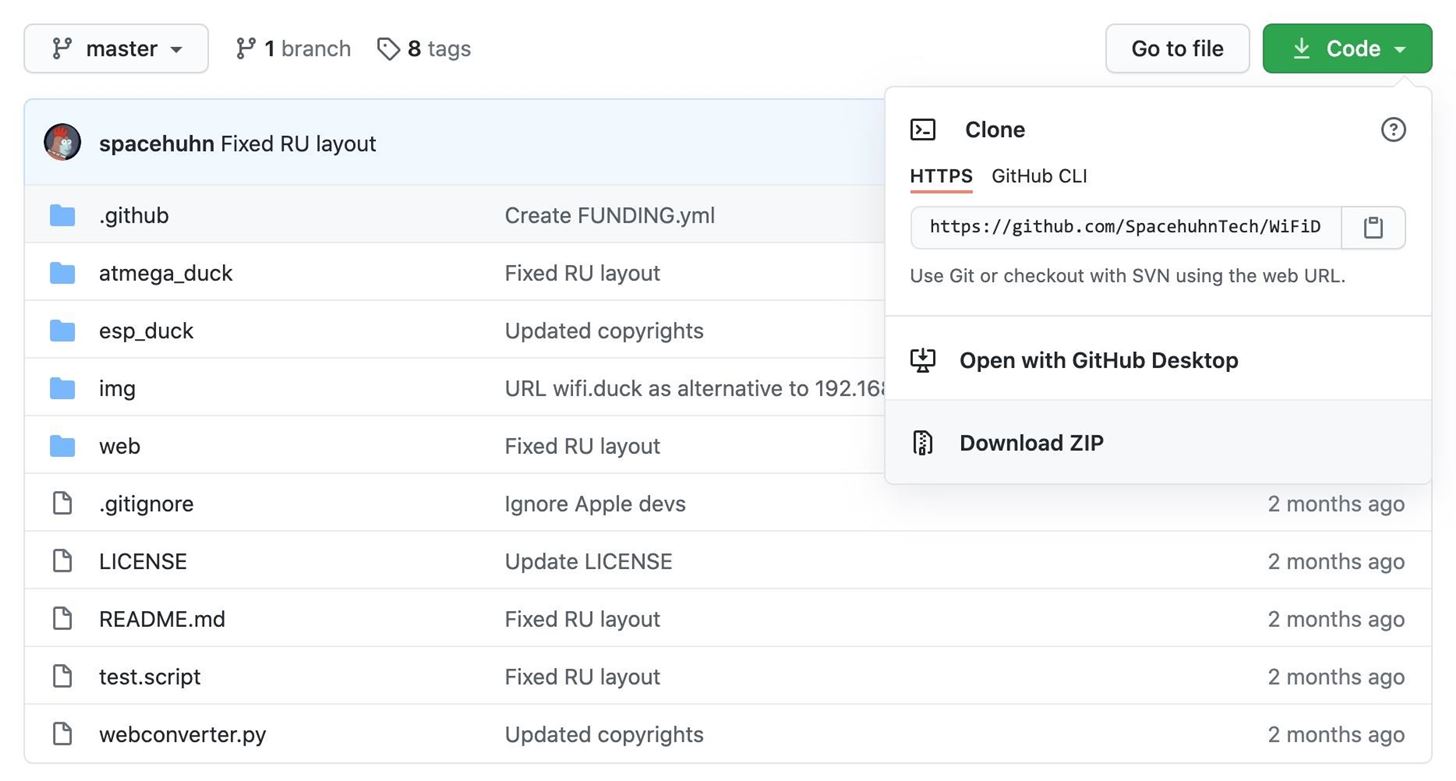


Now, go to "Tools" in the menu, hover over "Board," then select "Boards Manager." Perform a search for "Wi-Fi duck," then install both the Wi-Fi Duck AVR Boards and Wi-Fi Duck ESP8266 Boards options. If you already have them, make sure they're up to date. Click "Close" when done.



## Clone/Download the Wi-Fi Duck Repo

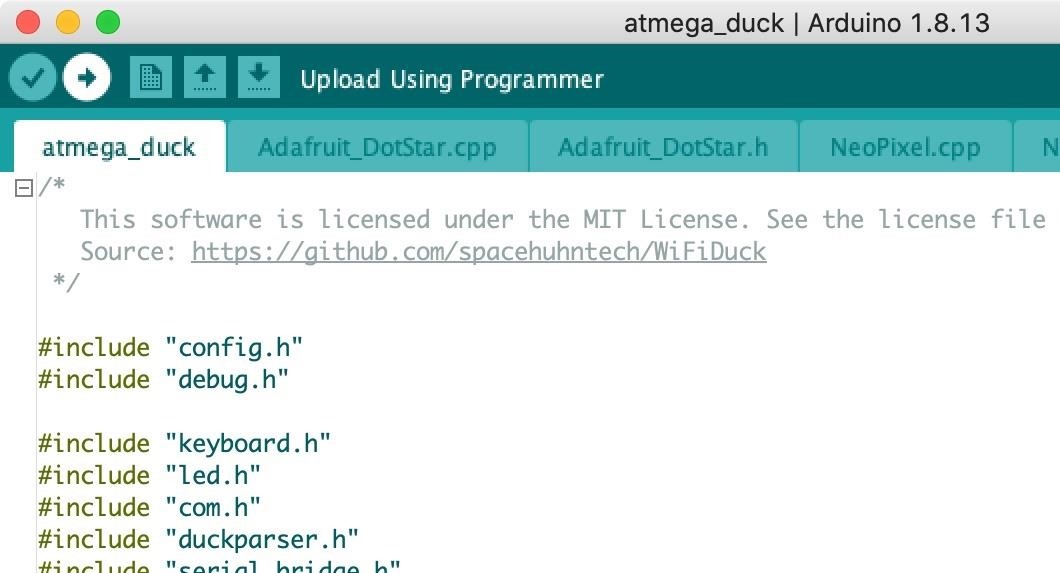
To get the code for both the ESP8266 and ATmega32U4, download the WiFi Duck repository as a zip file from GitHub. You can find it at the following link. Then unzip it on your computer.

* [](https://github.com/spacehuhn/wifiduck)github.com/spacehuhn/wifiduck

## [Flash Code to the ATmega32u4](https://github.com/spacehuhn/wifiduck)

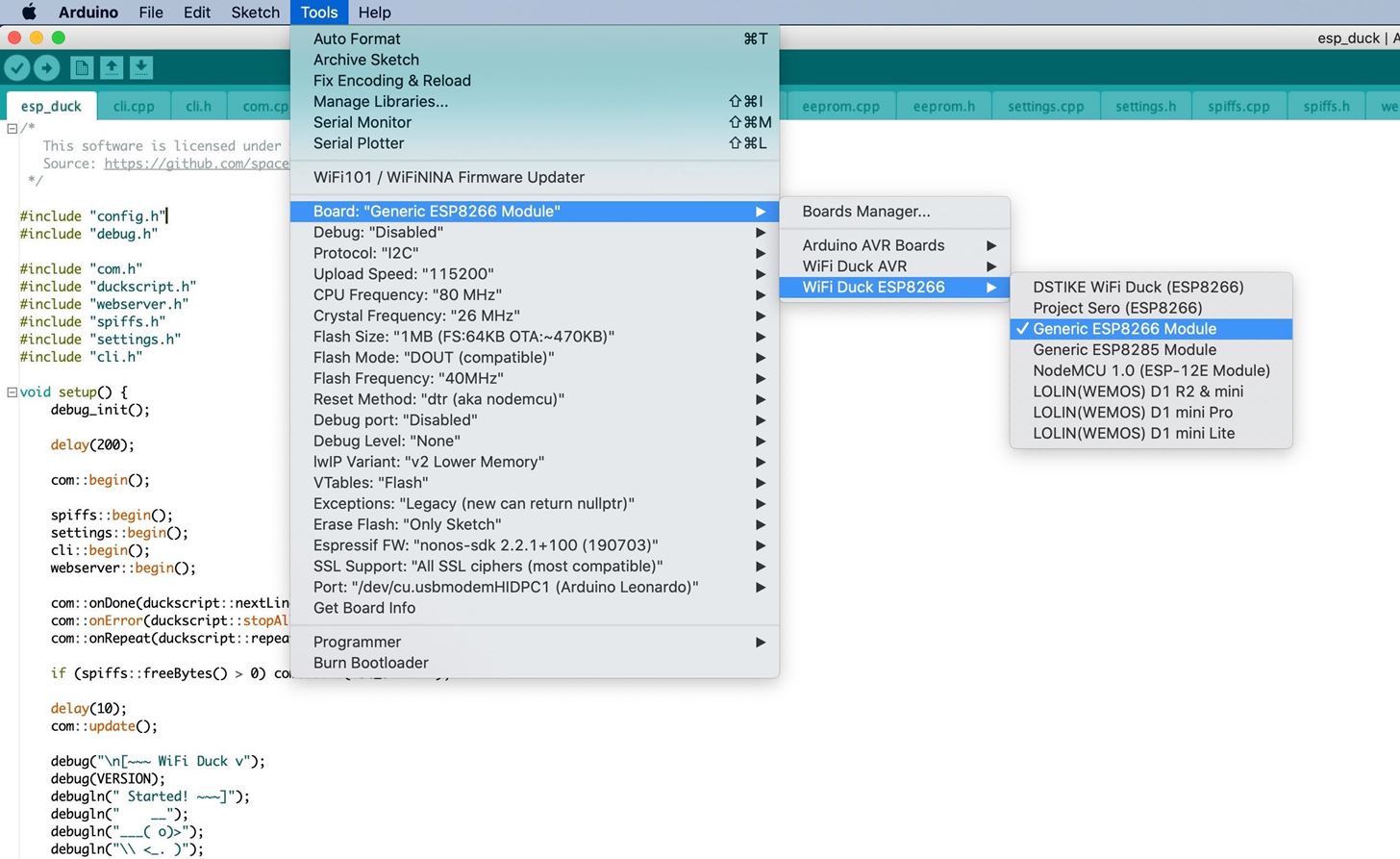
From the repo you just extracted, go into the **atmega\_duck** folder, then open the **atmega\_duck.ino** file in the Arduino IDE. No adjustments to the code are necessary. With it open in Arduino IDE, go to "Tools" in the menu, hover over "Board," then "Wi-Fi Duck AVR," and choose the board that you have.

Connect the ATmega32u4 board to your computer via your Micro-USB cable, then select its port in the "Port" selection in the "Tools" menu. If you don't see your board's serial port show up, the first thing you should do is make sure you're using a proper Micro-USB cable. I had about five cables, and only one of them ended up working with data transfers.



## Flash Code to the ESP8266

Now it's time to flash the code to your ESP8266. From the repo, go into the **esp\_duck** folder, then open the **esp\_duck.ino** file in the Arduino IDE. No adjustments to the code are necessary. With it open in Arduino IDE, go to "Tools" in the menu, hover over "Board," then "WiFi Duck ESP8266," and choose the board that you have.

After disconnecting the ATmega32U4, connect the ESP8266 board to your computer via your Micro-USB cable, then select its port in the "Port" selection in the "Tools" menu. Again, if you don't see your board's serial port show up, check that you're using a proper Micro-USB cable.

## Connect & Run First Script

With everything flashed, disconnect the ESP8266 from your computer, then connect ATmega32U4 back up with the Mini-USB cable. On your computer, change your Wi-Fi connection to the network called "wifiduck" and use "wifiduck" as the password.

You won't have any internet, but you will be communicating with your WiFi Duck setup. And if your target computer is the same as your computer, you can build your payloads ahead of time and try them out on your computer to make sure they work.

Once on the right network, open a browser and visit 192.168.4.1. This interface will let you do things like run the device, save scripts, and more. To change the WiFi network's name, click on "Settings," then change the SSID and password.