Thiết bị: **rubberducky không giây**

Chức năng: gõ phím không giây theo khối lệnh được tạo sẵn.

Công dụng:

+ Kết nối server nhận các gói dữ liệu

+ Gửi các gói sang cho **arduino pro micro** thực thi lên bàn phím.

**Cài đặt**

**Phần 1: sơ đồ nối chân vật lý**

|ESP8266 | arduino pro micro|

| D6 | 2 |

| D5 | 3 |

| D2 | 4 |

| VIN | RAW |

| GND | GND |

nguồn usb cấp vào con arduino pro micro

**Phần 2 : Hướng dẫn cài đặt.**

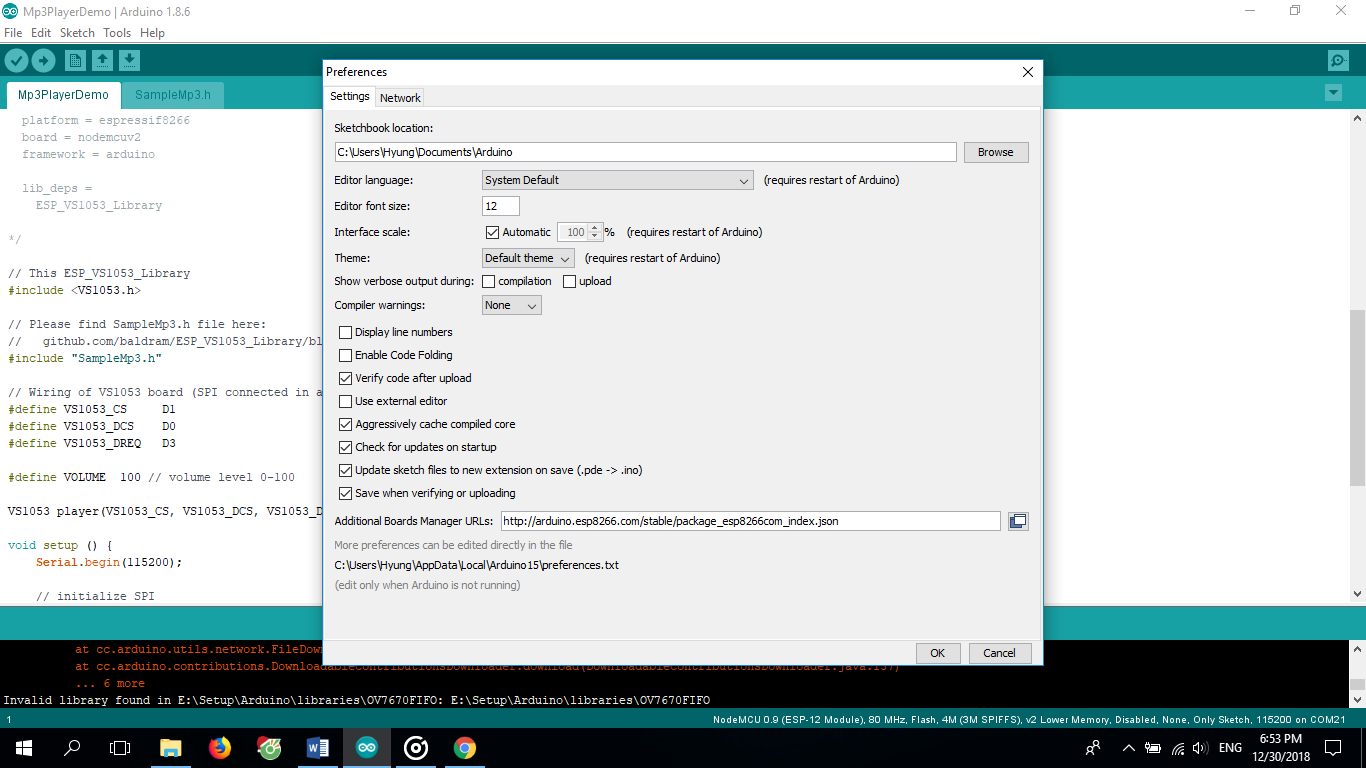
I. Cài đặt arduino

<http://arduino.vn/bai-viet/68-cai-dat-driver-va-arduino-ide>

II. Cài đặt thư viện liên quan

- Bước 1: Cài đặt thư viện cho ESP8266

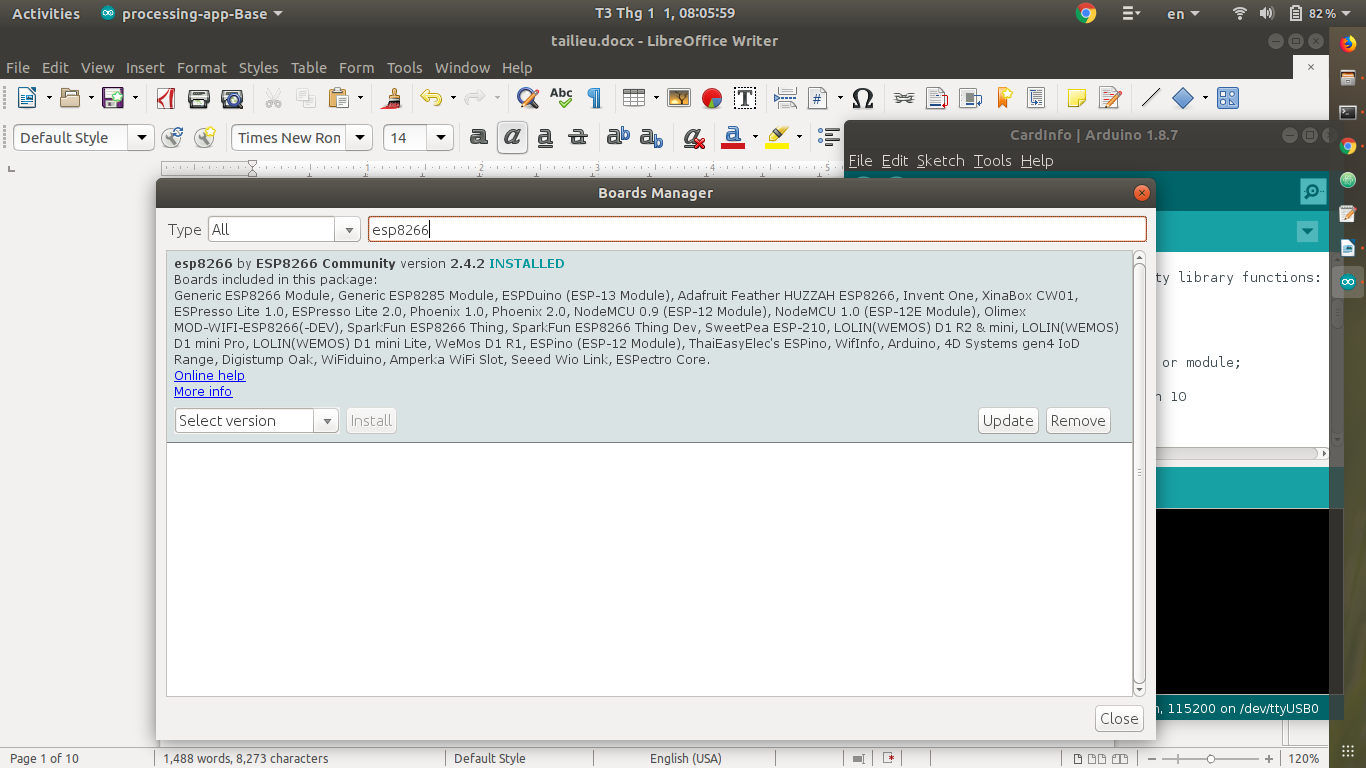
+ Đầu tiên vào *File - > References* cửa sổ sau hiện ra



+ Sketchbook location: chọn đường dẫn sẽ lưu trữ file và các thư viện.

+ Paste URL <http://arduino.esp8266.com/stable/package_esp8266com_index.json> vào ô *Additional Board Manager URLs* sau đó nhấn *OK*

*+ Tool > Board:… > Boards Manager > search esp8266*



Cài đặt board esp8266 by ESP8266 Communtity v2.4.2

Cài ESP8266 wifi:

<https://github.com/asdcxsd/ardnio/raw/master/ESP8266wifi-master.zip>

Sketch > include library > add zip library > chọn file nén thư viện.

Cài arduinojson 5.11:

link tải: <http://downloads.arduino.cc/libraries/github.com/bblanchon/ArduinoJson-5.11.0.zip>

Sketch > include library > add zip library > chọn file nén thư viện.

Cài websocketclient:

link tải: <https://github.com/Links2004/arduinoWebSockets/archive/master.zip>

Sketch > include library > add zip library > chọn file nén thư viện.

- Bước 2: Cài đặt SPIFFS:

<https://www.instructables.com/id/Using-ESP8266-SPIFFS/>

Bước 3: nạp code ESP8266

Tải code ESP8266

<https://github.com/phan123123/IOT/raw/device/Device/esp8266/uploadfile.zip>

Giải nén vào thư mục setup lưu trư file arduino

Chọn tools > board: .. > chọn NodeMCU 0.9 (ESP-12 Module)

Chọn port phù hợp.

Chọn Sketch > upload.

Upfile data : Tools > ESP8266 Sketch Data upload

Bước 4: nạp code ardunio pro micro

Tải chương trình về đầy đủ đặt trong thư mục rubberducky tại thư mục Arduino

<https://github.com/phan123123/IOT/tree/device/Device/pro_micro>

Chọn board Arduino/Genuino micro

Chọn port

Upload file

**Phần 3: Chi tiết code**

Thông số kỹ thuật:

Chương trình

Thư viện:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thư viện** | **Phiên bản** | **Công dụng** | **Ghi chú** |
| ArduinoJson.h | 5.11.0 | Cấu trúc các đoạn dữ liệu được gửi theo json | [Link](https://www.arduinolibraries.info/libraries/arduino-json) |
| ESP8266WiFi.h |  | Thao tác với wifi | [Thông tin](https://arduino.esp8266.vn/wifi/wifi-api.html)  [Tìm hiểu wifi](https://arduino.esp8266.vn/wifi/kn-wifi.html) |
| ESP8266WebServer.h |  | Tạo server để setup cho ESP8266 | [Thông tin thêm](https://github.com/esp8266/Arduino/tree/master/libraries/ESP8266WebServer) |
| FS.h |  | Thư viện SPIFFS để ghi vào bộ nhớ SPIFFS | [Tài liệu](http://esp8266.github.io/Arduino/versions/2.0.0/doc/filesystem.html) |
| WebSocketsClient.h |  | Thư viện tạo kết nối websocket với server | [Tài liệu](https://github.com/Links2004/arduinoWebSockets/tree/master/examples/esp8266/WebSocketClient) |
| serial\_v1.h |  | Gửi dữ liệu serial đến arduino pro micro | [Tài liệu](https://github.com/phan123123/IOT/tree/device/Device/gui_data) |
| data.h |  | Lưu lại dữ liệu gốc chương trình |  |
| File.h |  | Thao tác với file |  |
| setup\_wifi.h |  | Setup wifi |  |
| Server.h |  | server html cài đặt người dùng |  |
| Websocket.h |  | Thao tác với |  |

I. setup wifi

1. Đọc dữ liệu từ file: data/data\_wifi.tg

Được lưu trong bộ nhớ SPIFFS cấu trúc các thông số của wifi và thông tin websocket

Ví dụ:

{"NAME\_DRIVER":"Default driver","SSID\_STA":"InternetHV-ắt01","PASS\_STA":"12345678","ipWebSocket":"192.168.8.104","portWebSocket":8080,"linkWebSocket":"/Server\_java/actions","license":"KZSEIZGVQQEBCEQN"}

|  |  |
| --- | --- |
| Tên | Công dụng |
| NAME\_DRIVER | tên của thiết bị |
| SSID\_STA | tên wifi |
| PASS\_STA | cần kết nối |
| ipWebSocket | địa chỉ kết nối server websocket |
| portWebSocket | Cổng kết nối server websocket |
| linkWebSocket | link dẫn vào server websocket |
| license | key để tạo kết nối |

2. Dò wifi lân cận

Dò các wifi trong phạm vi có khả năng bắt được, nếu wifi cần bắt có trong phạm vi này sẽ tiến hành kết nối

3. Phát wifi

Luôn có 1 wifi được phát song song

Mặc định: Tên: Wifi chua

MK: 12345678

II. Mở công kết nối đến ardunio pro mico

Tài liệu kết nối: <https://github.com/phan123123/IOT/tree/device/Device/gui_data>

Tạo một kết nối có tên mySerial đến arduino pro mico qua 2 chân d5, d6

Tốc độ delay 100 micro giây

**III. Server html để cài đặt thiết bị cho người dùng**

1.Webserver

Webserver cung cấp cho người sủ dụng dễ dàng setup cho hệ thống theo hướng người dùng

**NOTE:** Nếu không kêt nối vào bất cứ wifi nào đều có thế vào wifi con đấy vừa phát ra thì truy cập vào wifi mà 2 con này phát

Dịa chỉ:

192.168.4.1: ip khi kết nối với wifi phát

IP khi kết nối với wifi

2. /

Trong gốc, khi đã đăng nhập thành công thì có quyền vào sửa các thông số cho esp

3. /login

Trang đăng nhập để setup thông số

username: admin

password: admin

Chương trinh sinh ra 1 đoạn string dài 32 ký tự, khi đăng nhập thành công thì đoán string sẽ được lưu lại trong cookie thành phiên làm việc hiện tại

3. /writefile

Ghi dữ liệu vào file mới

4. /send

Gửi các lệnh đến arduino pro micro không cần kết nối websocket

Cấu trúc thêm trường data vào phương thức get GET

VD: 192.168.4.1/send?data=nguyen manh thien

**IV WebSocket**

**1. Thư viện websocketClient**

Tài liệu :

<https://github.com/Links2004/arduinoWebSockets/tree/master/examples/esp8266/WebSocketClient>

<https://viblo.asia/p/websocket-la-gi-Ljy5VxkbZra>

Tạo kết nối websocket với server

**2. Quá trình tạo kết nối**

+ Khi thiết bị kết nối vào websocket của server. đồng thời sẽ gửi gói để bắt tay. Trong gói đó có key được lưu trữ.

+ Server nhận được key, check nếu đúng thì đổi key mới rồi gửi về cho thiết bị

+ Thiết bị nhận được key mới đổi lại rồi lưu lại vào file trên bộ nhớ SPIFFS

+ Lúc này 2 thiết bị có thể gửi dữ liệu qua lại cho nhau

sau khi kết nối ESP8266 sẽ gửi 1 gói đến server có cấu trúc sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Giá trị** | **Tính năng** | **Giá trị mặc định** |
| action | add | Giá trị để server biết và add thiết bị | add |
| kind | receive | Thể loại | receive |
| name | “Tên của thiết bị” | Tên thiết bị hiển thị trên server |  |
| license | “String 16 ký tự” | Key tạo kết nối với server | KZSEIZGVQQEBCEQN |
| type | “Loại thiết bị” | Loại thiết bị | 123 |
| description | “Mô tả” | Mô cả các đặc tình thêm | asdfsa |

**Chú ý:**

+ Các thông số sẽ được lưu trên server

+ Khi nào đúng key thì mới có thể kết nối

+Key mặc định luân được tạo kết nới với server

Server nhân được kiểu tra key:

- nếu key trùng với key mặc định, thiết bị sẽ được thêm vào

- nếu đúng key sẽ cho giao tiếp

- nếu sai sẽ nhận được gói có cấu trúc sau:

|  |  |
| --- | --- |
| action | Error |
| data | lỗi được gửi lại |

lúc này websocket được đưa vào trạng thái chết. cân reset để tiếp tục thực hiện.

**3. nhận dữ liệu được gửi về**

get dữ liêu từ trường action:

+setLicense: key đã được dổi trên server.

nhận action này thiết bị sẽ tiến hành lấy key từ trường license và lưu lại

+get: gói phản hồi của server để check kết nối

khi nhận dước 1 gói sẽ được gửi lại; cấu trúc:

|  |  |
| --- | --- |
| Action | post |
| data | receive |
| id | “Tên thiết bị” |

+error: lỗi được gửi về trong trường data

+TEXT: dữ liệu dược gửi về

Gói này gồm:

|  |  |  |
| --- | --- | --- |
| file | char\* | tên file server gửi |
| byte | Int | vị trí gửi |
| sizetext | int | Kích thước gửi |
| text | String | Dữ liệu gửi |
| end | int | 0: file vẫn còn |
| 1: file đã hết |

Sau khi nhận được dữ liệu sẽ được gửi sang arduino pro micro

**4. Yêu cầu gửi dữ liệu tiếp**

Sau khi nhận được dữ liệu mà file vẫn còn thì tiến hành yêu cầu server gửi gói tiếp theo

Gửi gói sau về server:

|  |  |
| --- | --- |
| action | GETTEXT |
| file | Tên file cần lấy |
| byte | Vị trí byte đầu tiên |
| sizetext | Số lượng byte có thể gửi |

**V Ardunio pro micro**

Chức năng:

Nhận dữ liệu từ esp8266 vào thực hiện gõ bàn phím

1. Thư viện

Keyboard.h: thư viện gọi chức năng bàn phím ảo của arduino pro micro

Tài liệu: <http://arduino.vn/bai-viet/1089-che-tao-keyboard-voi-arduino-pro-micro>

2. Tạo kết nối nhận dữ liệu từ ESP8266

Tài liệu: <https://github.com/phan123123/IOT/tree/device/Device/gui_data>

3.Cách thức nhận và xử lý số liệu

Dữ liệu được gửi dưới dạng HEX, hai ký tự HEX từ 00 → FF sẽ chuyển thành 0 → 255

sau khi chuyển thành số int sẽ đem đi xử lý.

char trong

Bảng tham số

|  |  |  |
| --- | --- | --- |
| Key | Hexadecimal value | Decimal value |
| KEY\_LEFT\_CTRL | 0x80 | 128 |
| KEY\_LEFT\_SHIFT | 0x81 | 129 |
| KEY\_LEFT\_ALT | 0x82 | 130 |
| KEY\_LEFT\_GUI | 0x83 | 131 |
| KEY\_RIGHT\_CTRL | 0x84 | 132 |
| KEY\_RIGHT\_SHIFT | 0x85 | 133 |
| KEY\_RIGHT\_ALT | 0x86 | 134 |
| KEY\_RIGHT\_GUI | 0x87 | 135 |
| KEY\_UP\_ARROW | 0xDA | 218 |
| KEY\_DOWN\_ARROW | 0xD9 | 217 |
| KEY\_LEFT\_ARROW | 0xD8 | 216 |
| KEY\_RIGHT\_ARROW | 0xD7 | 215 |
| KEY\_BACKSPACE | 0xB2 | 178 |
| KEY\_TAB | 0xB3 | 179 |
| KEY\_RETURN | 0xB0 | 176 |
| KEY\_ESC | 0xB1 | 177 |
| KEY\_INSERT | 0xD1 | 209 |
| KEY\_DELETE | 0xD4 | 212 |
| KEY\_PAGE\_UP | 0xD3 | 211 |
| KEY\_PAGE\_DOWN | 0xD6 | 214 |
| KEY\_HOME | 0xD2 | 210 |
| KEY\_END | 0xD5 | 213 |
| KEY\_CAPS\_LOCK | 0xC1 | 193 |
| KEY\_F1 | 0xC2 | 194 |
| KEY\_F2 | 0xC3 | 195 |
| KEY\_F3 | 0xC4 | 196 |
| KEY\_F4 | 0xC5 | 197 |
| KEY\_F5 | 0xC6 | 198 |
| KEY\_F6 | 0xC7 | 199 |
| KEY\_F7 | 0xC8 | 200 |
| KEY\_F8 | 0xC9 | 201 |
| KEY\_F9 | 0xCA | 202 |
| KEY\_F10 | 0xCB | 203 |
| KEY\_F11 | 0xCC | 204 |
| KEY\_F12 | 0xCD | 205 |

Bảng ký tự điều khiển

|  |  |  |
| --- | --- | --- |
| 233 | Delay | ký tự liền sau nó sẽ là delay ms  giá trị ký tự sau \*10  Phạm vi delay 200ms |
| 235 | dữ phím | Thực hiện số 1 vài phím  chức năng tổ hợp phím |
| 234 | Thả phím | Kết thúc ấn tổ hợp phím |