

# WiFi Garage Door Controller

Rev 1.1



Ken S.

11/2022

## 1. Description

This is an ESP32-WROOM based design for a WiFi enabled garage door controller capable of operating 2 garage doors. It includes illuminated physical pushbuttons wired in parallel with relays to allow the user to operate the garage doors even if the controller is unpowered or disconnected from WiFi. Included controls mimic those found on the manufacturer provided openers for the units I have – Open/Close, Light On/Off, and Lock. The ESP32 acts as a webserver hosting a website with the applicable controls / status indications for both doors. This project uses the WiFi manager library to aid in WiFi connection and to prevent the need to hard-code the WiFi network credentials into the program. OTA update functionality is also included.

Design files are included for the following:

1. Controller Electrical Schematic (w/ manufacturer and P/N component properties)
2. Controller PCB Layout
3. Interactive PCB Bill of Materials (BOM) – HTML format
4. VS Code / PlatformIO based code
5. 3D Printed Enclosure

## 2. Electrical / PCB

The controller PCB uses 4 reed style relays wired in parallel with physical LED illuminated pushbuttons to control garage door operation. Off board connections use JST-PH style connectors. Please mind polarity (LEDs, etc.) when terminating off-board devices using the male JST-PH connectors.

‘Auto Program’ transistors are included in the design to prevent the need to press the ‘reset’ button after the code is uploaded to the microcontroller. Indication LEDs are generic 5mm THT LEDs.

The manufacturer provided garage door control pad(s) that this design is intended to emulate uses DC voltage only and a voltage divider circuit to determine which button has been pressed / whether the lock switches are set. Please refer to the notes in the addendum or electrical schematic for details.

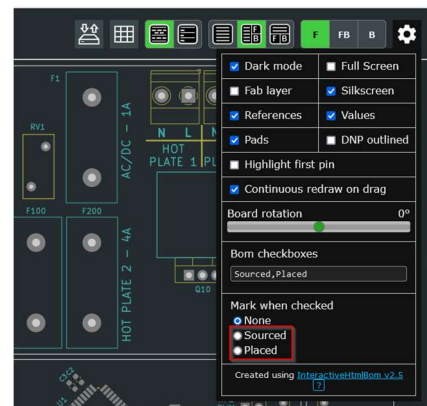
***Determination of how your opener works is REQUIRED and resistance values / entire circuit design may need to be changed to accommodate your specific garage door controller!***

Note that design accommodation for a UART pin header is included, but not required to be used due to the onboard USB – UART bridge IC.

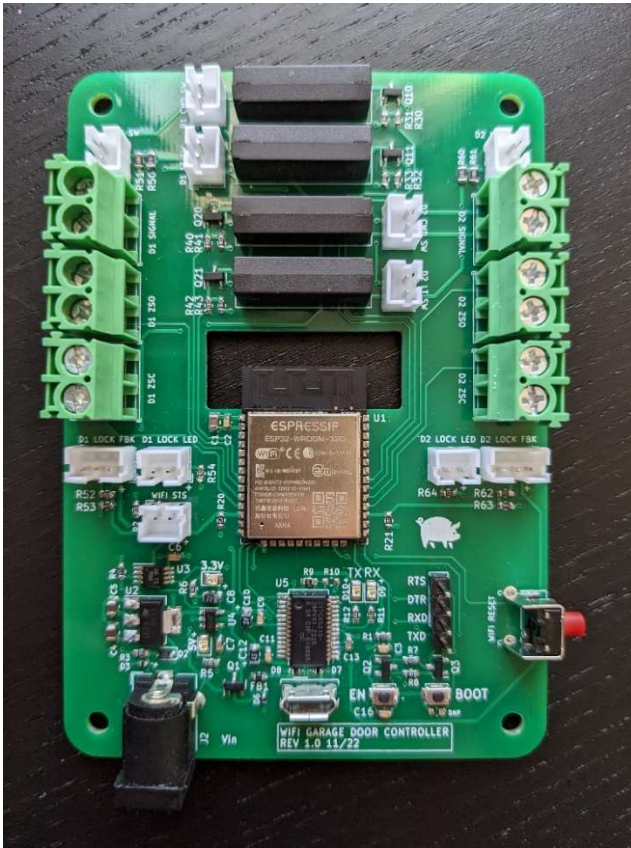
This design uses a Cypress / Infineon CY7C65213-28PVXI USB – UART bridge IC. The newer CY7C65213A-28PVXI may be used in its place. Please use the configuration utility that can be found at the following URL along with the included configuration file to program the USB – UART bridge:

<https://www.infineon.com/cms/en/design-support/tools/configuration/usb-uart-config-utility/>

Please refer to the included electrical schematic / PCB design / interactive BOM (html format) for further details. Note that selection of one of the options shown in the screenshot of the interactive BOM may be helpful during PCB assembly –



## Completed PCB / Installation Pictures:





### 3. Assembly / Construction

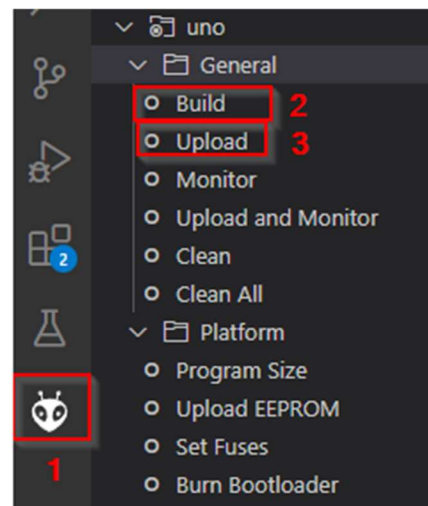
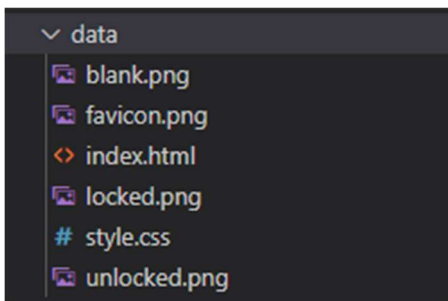
Please refer to the included design files for assembly / construction details. Various format 3D models (STEP, F3D, and STL) are included, along with assembly / enclosure component drawings.

Drill holes in top of enclosure bottom of appropriate size for incoming wiring prior to mounting enclosure to the wall. Mount enclosure to the wall prior to mounting PCB in enclosure. CAT5 ethernet cable may be used to provide 8 conductors (2 spare).

**NOTE:** M3 Threaded heat press inserts are required for PCB mounting holes and enclosure top mounting holes (shown in pictures from previous section). Holes for PCB mounting and enclosure top mounting are sized in the 3D model to accept these inserts – fasteners will not thread into these holes due to their diameters.

### 4. Microcontroller Code

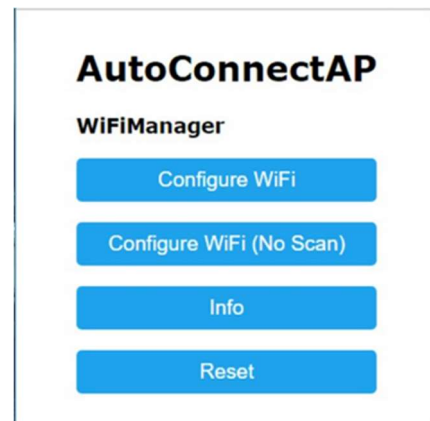
VS Code / PlatformIO must be used to upload the code to the microcontroller unless the user creates and Arduino IDE version. Note that for upload, there are additional files that must be uploaded to the ESP32 file structure (SPIFFS) – these can be found in the 'data' directory in the project. To upload to SPIFFS, select the PlatformIO icon on the left side bar, then select 'Build' and then 'Upload':



The code uses the following libraries:

1. Arduino.h
2. WiFi.h
3. ESPAsyncWebServer.h
4. WebSocketsServer.h
5. AsyncElegantOTA.h
6. ArduinoJson.h
7. SPIFFS.h
8. StreamString.h

If a WiFi network has not been configured, upon power up the ESP32 will operate in Access Point (AP) mode and hosts a website that scans for available WiFi networks and allows the user to select and provide credentials to connect. Once saved, this information is retained by the ESP32, even after a power cycle. **It is suggested to set a static IP address for the ESP32 in your router configuration to ensure the IP address always remains the same.**



The WiFi manager portal is configured to use the default http port of 80 for ease of use and connection.

There is a 'WiFi' reset button included in the design on the right side of the PCB that when pressed, will reset the WiFi connection configuration and re-enable WiFi manager.

Once connected to a WiFi network, WiFi manager will be disabled and the ESP32 will operate in station mode and automatically connect to the configured WiFi network upon power up.

The web page used for control and monitoring uses port 8080. This site is hosted by the ESP32 and can be modified by updating the index.html and style.css files contained in the project. It uses web sockets to pass JSON messages between the web page and the control code running in the ESP32. The web sockets server is set to use port 8081.

## 5. Operation / Usage

The controller can be powered via the 2.1mm barrel jack connector with a readily available 9VDC or 12VDC AC to DC converter - Ensure correct polarity of jack connection before purchase (positive on the inside).

The physical pushbuttons may be used in parallel with the web page to control garage door functions.

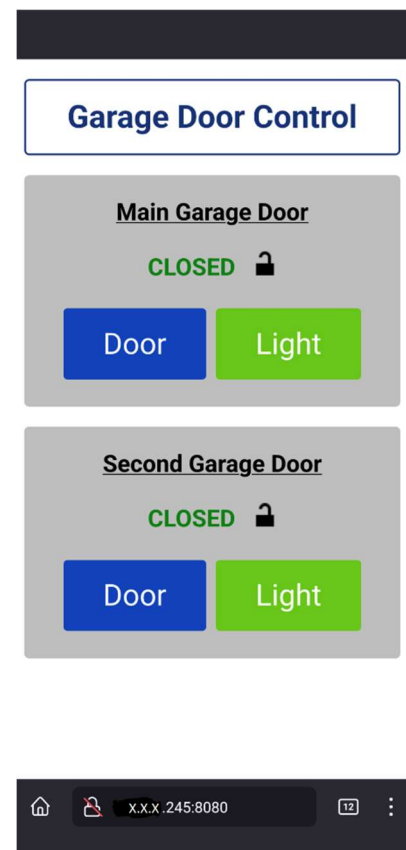
***Note that the lock functionality can only be toggled with the physical switch.***

### Control Over WiFi -

To connect to the control web page, navigate in a browser (computer or mobile device) to the IP address of the controller with the port suffix ':8080' (e.g. 192.168.1.245:8080 - see screenshot).

Indicators / controls on the web page allow the user to monitor the current open or closed state of the garage door, whether the lock switch is set, and buttons are provided to issue an open / close command ('Door' button) or turn the light on or off ('Light' button). If the limit switches are in a state where neither is made, the indicator will display 'Transition', if both switches are made simultaneously, it will display 'FAULT'.

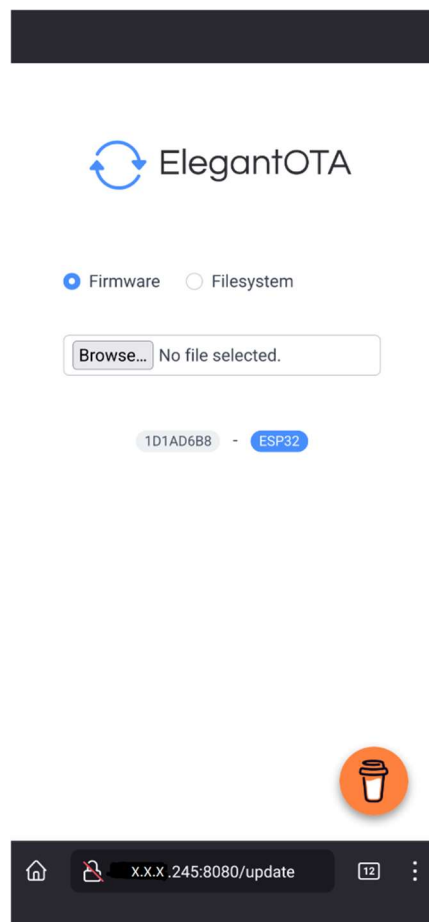
If your device has been disconnected from the WiFi network the garage door controller is connected to, a page refresh may be required for status to update / commands to issue to the controller.



## OTA Updates –

Program code as well as SPIFFS file system files can be updated ‘Over the Air’ (OTA) without the requirement to connect via USB. The file format required for OTA updates (both program and SPIFFS files) is .bin and must be created from the IDE before upload.

To use the OTA Update function, navigate to the IP\_Address:Port (8080)/update (e.g. 192.168.1.245:8080/update - see screenshot).



## 6. Addendum

### **Door Interface Notes:**

#### **Measured Voltages:**

Door 1: ~ 5 VDC (No AC)

Door 2: ~ 18 VDC (No AC)

#### **Measured Resistances (same for both doors):**

##### **Unlocked:**

No Buttons	Open Line
Command Button	< 1 ohm (short)
Light Button	~ 204 ohms

##### **Locked:**

No Buttons	~83 ohms
Command Button	< 1 ohm (short)
Light Button	~ 83 ohms

### **Links / Part Numbers for various off-board / supporting parts:**

#### **DC Barrel Jack Connector – 2.1mm:**

Adafruit PID 373

<https://www.adafruit.com/product/373>

#### **Door Switches (Green):**

Adafruit PID 1440

<https://www.adafruit.com/product/1440>

#### **Door Switches (Yellow / Amber):**

Adafruit PID 1441

<https://www.adafruit.com/product/1441>

#### **Lock Slide Switches:**

Nidec Copal - MFS201N-Z

<https://www.digikey.com/en/products/detail/nidec-copal-electronics/MFS201N-Z/5086556>

#### **Door Limit Switches (Generic):**

<https://a.co/d/92kAZzZ>