

# Antigneous Instruction Manual

Written by Lexzach

## What is this?

Antigneous is an affordable, open-source fire alarm control panel (FACP). This is targeted towards hobbyists or areas that do not have the privilege of having a fire alarm system from a name-brand company.

## Disclaimer!

Antigneous is **not** officially approved for use as a fire alarm system. Although the creator has done everything they can to make sure the panel will have extreme reliability, this cannot be guaranteed without extensive testing by official parties. If you do not accept the inherent risk of using a non-approved, homemade fire alarm system. **DO NOT USE THIS.**

## Table of Contents

Getting Started.....	2
What you need:.....	2
Flashing the firmware.....	3
1. Download the Antigneous firmware from the GitHub.....	3
2. Download the Arduino IDE.....	3
3. Add the ESP32 board definitions to Arduino IDE.....	4
4. Install the ESP32 board.....	4
5. Open the Antigneous firmware and Install the Liquid Crystal I2C library.....	5
6. Select your board.....	5
7. Select your port.....	5
8. Flash the firmware.....	5
Building the Panel.....	6
1. Breadboards.....	6
2. The Micro Controller.....	6
3. Connecting the LCD.....	6
4. Connecting Status LEDs and Buttons.....	8
5. Connecting the Relays.....	14
6. Creating the Zones.....	15
Using your New Panel.....	18
Entering the menu.....	18
The Testing and Settings menu.....	18
Fire Alarm Settings.....	18
Panel Settings.....	19

Fail-safe Mode.....	20
Error Codes.....	21

## BEFORE YOU GET STARTED:

If you need help with flashing, building, or operating your panel, join the Antigenuous Discord server. I am always happy to help!

<https://discord.gg/fwqYtBrfqs>

## Getting Started

What you need:

**REQUIRED** - **OPTIONAL**

Hardware:

- Wires, recommended that you get, Male to Male, Male to Female, *and* Female to Female
- DOIT ESP32 Dev Kit V1 (or other compatible board, but not guaranteed to work) [x1]
- Micro USB cable for programming and powering the board [x1]
- 16x2 LCD screen with IC2 interface [x1]
- LEDs [x3]
- Momentary push buttons [x3]
- Arduino compatible relays [x3]
- Piezoelectric buzzer [x1]

Software:

- A copy of the Antigenuous firmware
- Arduino IDE
- ESP32 board definitions
- LiquidCrystal I2C library

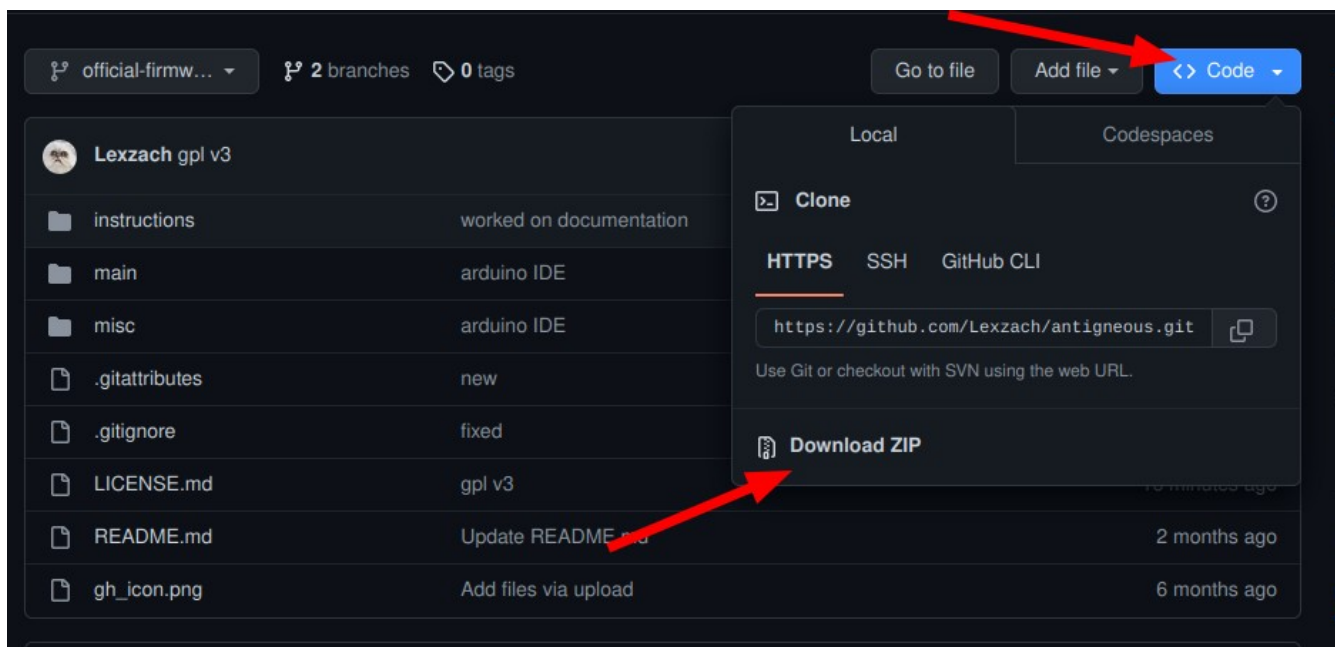


Figure 1: The first Antigenuous panel

## Flashing the firmware

### 1. Download the Antigneous firmware from the [GitHub](#).

If you are a *hobbyist* and wish to tinker with the firmware, and you want more recent builds, click the “code” button, then download as ZIP.



People who don’t want to experiment with the firmware or need reliability should download the firmware from the [releases](#).

### 2. Download the Arduino IDE

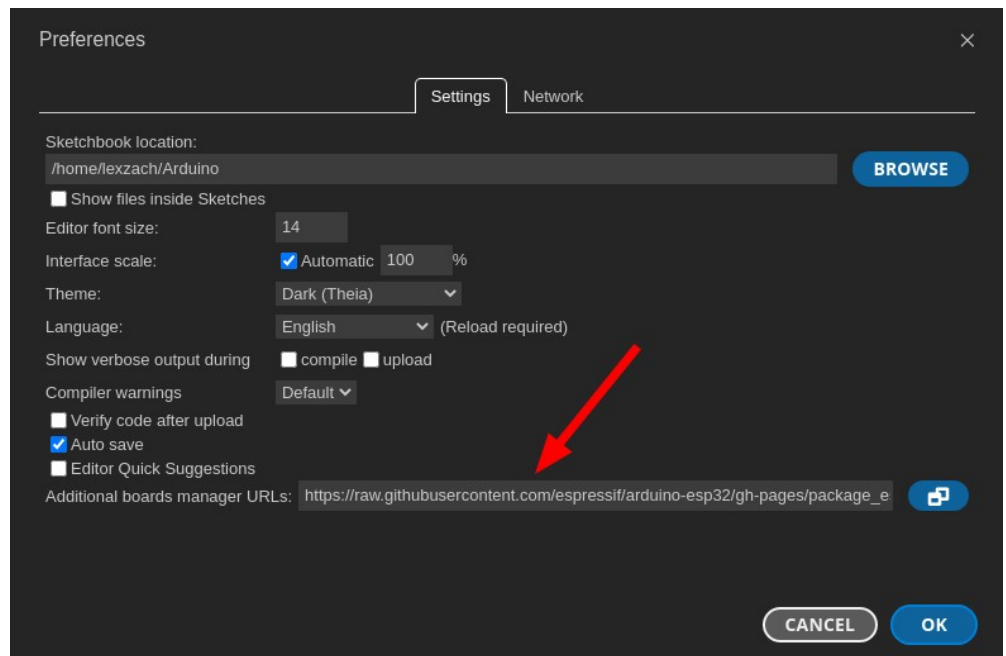
Navigate to the [Arduino](#) website in your web browser of choice, and download Arduino IDE.

### 3. Add the ESP32 board definitions to Arduino IDE

Navigate to File → Preferences, and paste

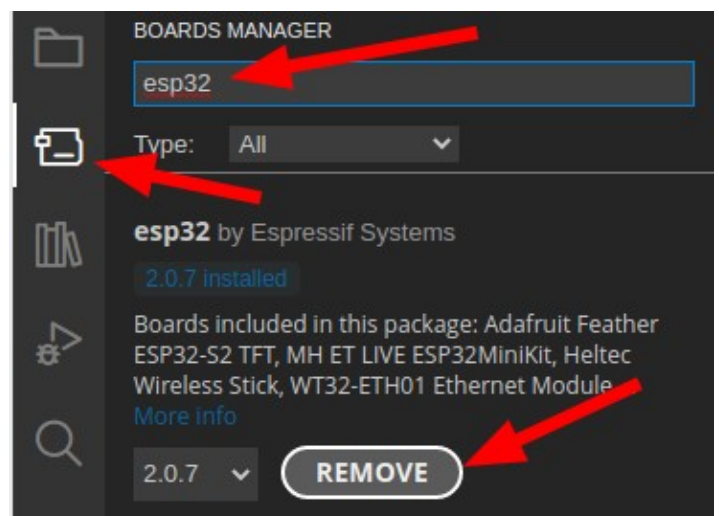
[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)

into the box labeled “Additional boards manager URLs”



### 4. Install the ESP32 board

Navigate to the board manager, and type in “esp32”  
Install the board by “Espressif Systems.”

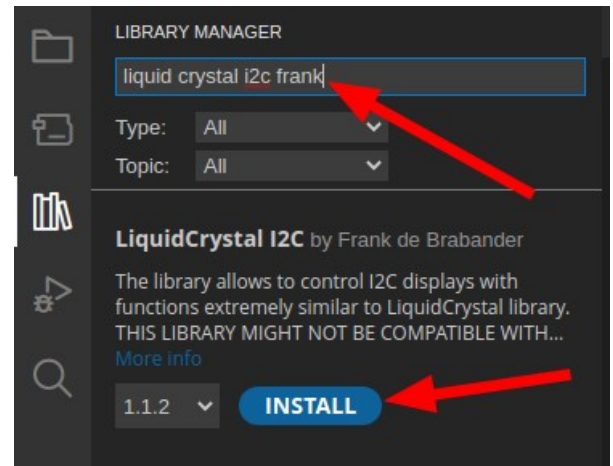


## 5. Open the Antigneous firmware and Install the Liquid Crystal I2C library

Navigate to the folder in which you downloaded the Antigneous firmware. Open the firmware file labeled “main.ino” inside the “main” folder.

Once opened, navigate to Sketch → Include Library → Manage Library

Type in “liquid crystal i2c frank” and install the library by “Frank de Brabander”



## 6. Select your board

Navigate to Tools. If you don’t see “Board: DOIT ESP32 DEVKIT V1,” you will need to navigate to Tools → Board → esp32 and click on “DOIT ESP32 DEVKIT V1.”

## 7. Select your port

Plug the ESP32 into your computer.

Navigate to Tools → Port

Select the port that appears. It will most likely start with “COM”

## 8. Flash the firmware

Click the forward arrow icon in the top left to flash the firmware to the device.



**You’re done with flashing the panel!**

# Building the Panel

## 1. Breadboards

For this panel, I am going to assemble it on a breadboard. This allows for prototyping, but isn't good for a panel built for reliability, since wires can freely go in and out.

At the top of the breadboard, you have the positive and negative rail. We call it a "rail" because it stretches across the entire board. You can see there are two rows of holes in between the negative line, marked in blue and a minus, and the positive line, marked in red and a plus. The row closest to the blue line are all negative. The other row of holes closest to the red line are all positive. In the middle of the breadboard are all the other holes that we use to connect components. Inside the breadboard, there are wires that connect all of these in columns. You can see that these columns are marked in white. This means that if you connect a component to "J1" it will be electrically connected to "F1." These connections do not go past the middle divider, meaning that there are only five holes that are electrically connected per column.

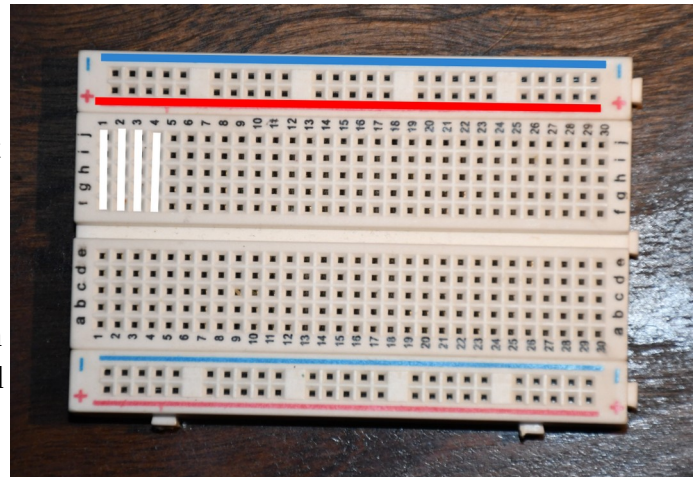


Figure 2: The electrical connections of a breadboard

## 2. The Micro Controller

The ESP32 is the brains of the control panel. You can see that there are 15 pins on each side of the device. These pins are labeled at the top of the device. Make sure you connect the right device to the right pin.

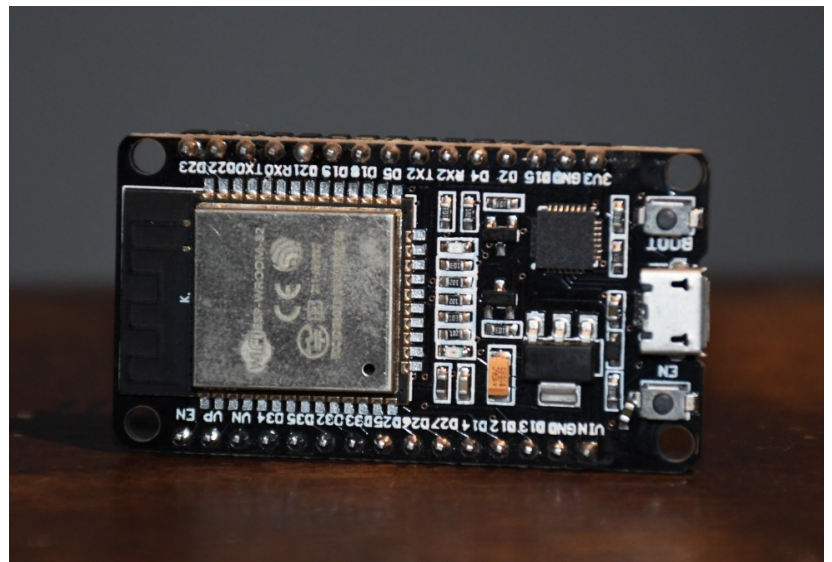


Figure 3: The ESP-32 board

## 3. Connecting the LCD

The first component we are going to connect is the LCD. This will verify that the panel is working.

On the back of the LCD, there should be a separate chip with four pins. This is the I2C interface, which allows us to interface with the LCD without needing a lot of pins.

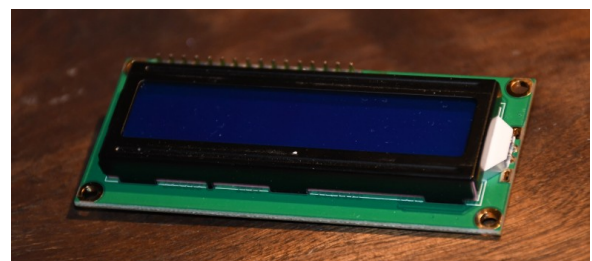


Figure 4: The LCD display



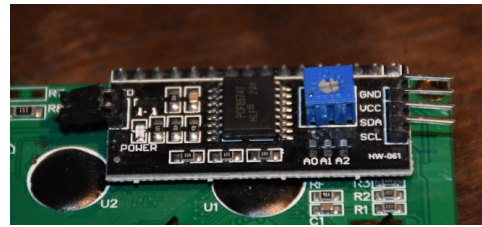


Figure 5: The I2C interface

First, we are going to connect four wires to those four pins.

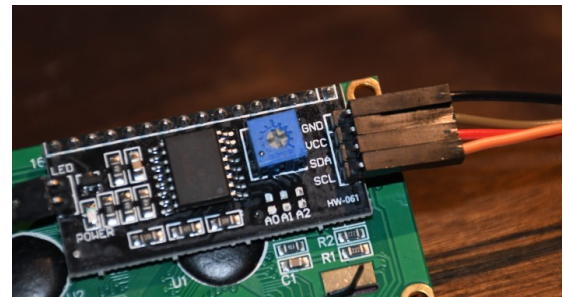


Figure 6: Connecting the four wires to the I2C interface

We will now plug those wires into the ESP32.

Plug the “GND” wire from the LCD into the “GND” pin on the ESP32. In this case, the black wire.

Plug the “VCC” wire from the LCD into the “VIN” pin on the ESP32. In this case, the brown wire.

Plug the “SDA” wire from the LCD into the “D21” pin on the ESP32. In this case, the red wire.

Plug the “SCL” wire from the LCD into the “D22” pin on the ESP32. In this case, the orange wire.

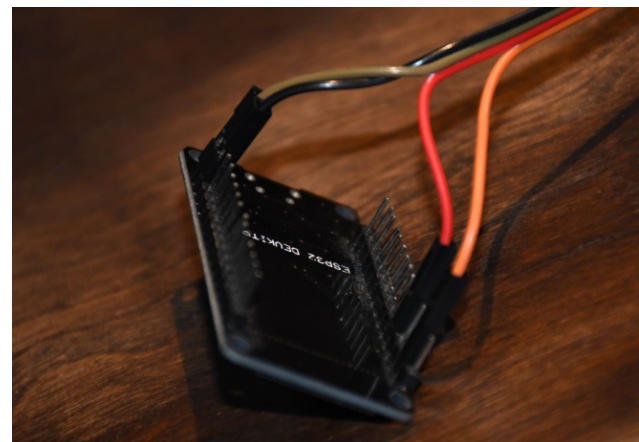


Figure 7: Connecting the four wires from the I2C interface into the four pins on the ESP-32

Now, plug in the ESP32 into power.

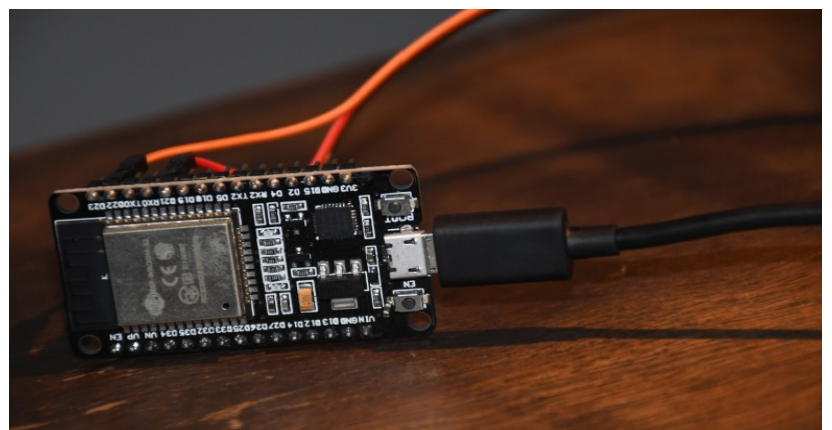


Figure 8: Connecting the ESP-32 to USB power

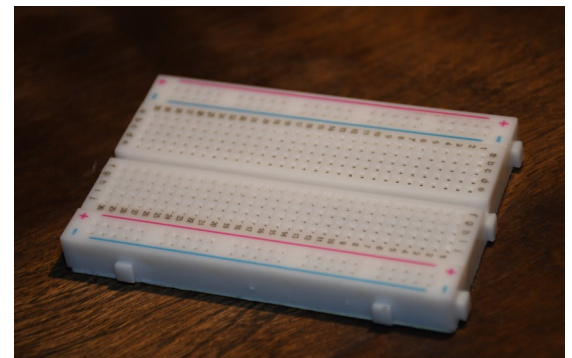
If the board has not had Antigneous installed before, there will be no settings stored, resulting in “ERROR 2” being displayed. THIS IS NORMAL, and actually means everything is configured correctly and is working. If your LCD display does not show anything, you may need to adjust the blue contrast knob found on the back.



*Figure 9: Successful bootup of the ESP-32. This will most likely NOT be what you see, as this board has already had Antigneous installed on it once, and is able to pull from stored settings, which won't likely be present on your board.*

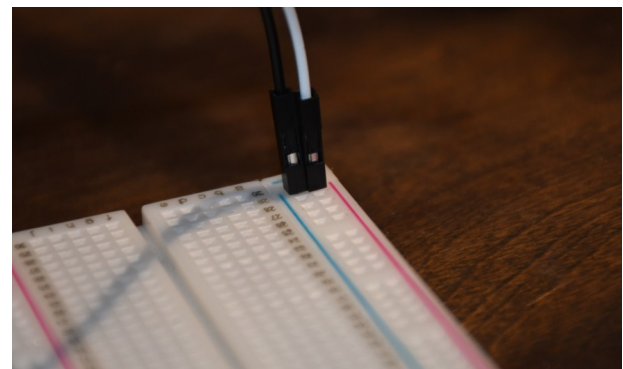
#### 4. Connecting Status LEDs and Buttons

We are going to use a breadboard for the LEDs and buttons.



*Figure 10: The breadboard*

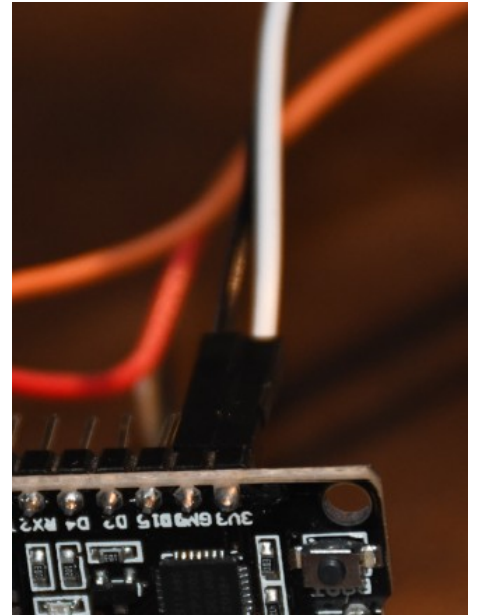
We first need to connect the positive and negative rails to the ESP32.



*Figure 11: Connecting the positive and negative rails*

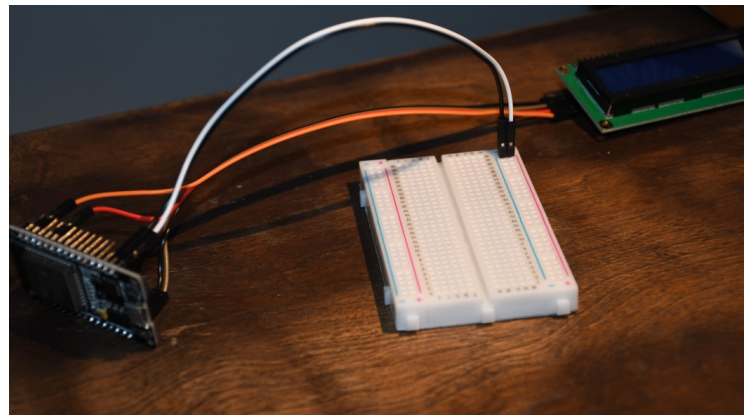


Connect the positive (in this case white) and negative wires (in this case black) to the “3V3” and “GND” pins respectively.



*Figure 12: The two wires from the breadboard connected to the ESP-32*

This is what your build should look like now.



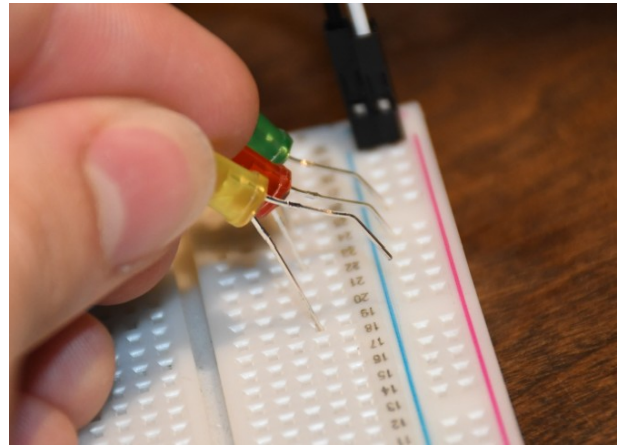
*Figure 13: The progress made on the panel so far*

Next, we will connect the LEDs to the breadboard.



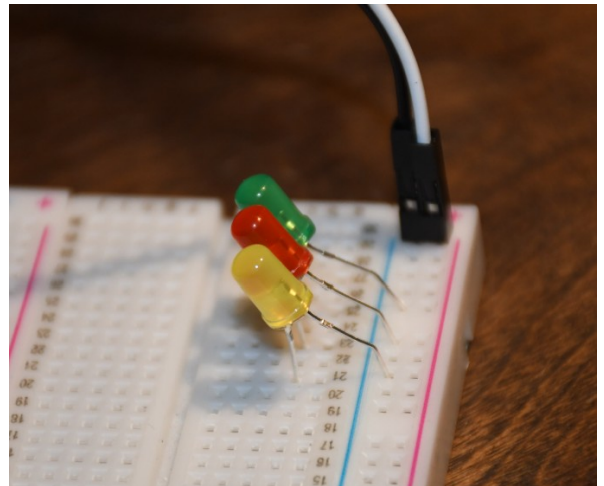
*Figure 14: The three LEDs. Green for ready. Red for alarm. Yellow for silence.*

Stick the shorter leg into the blue, negative rail. Then stick the longer leg into the component pins of the breadboards. Do not connect multiple LEDs to a single electronically connected column.



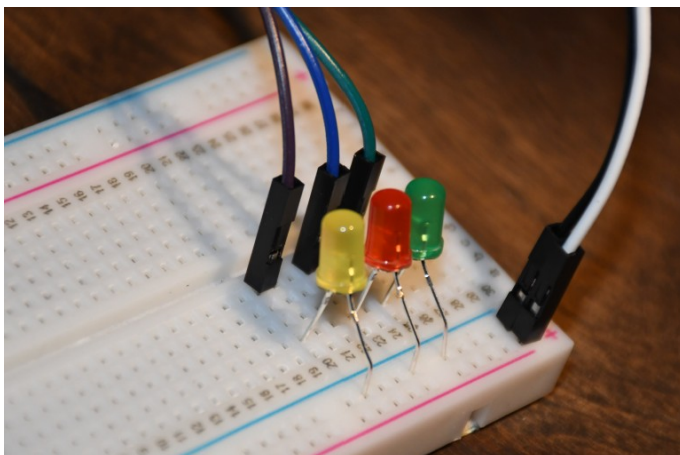
*Figure 15: Connecting the three LEDs to the breadboard*

This is what your breadboard should look like now.

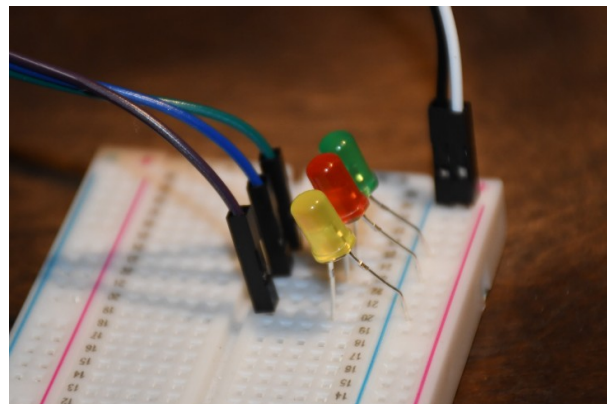


*Figure 16: Connected LEDs*

Connect three wires to the three channels of each LED.



*Figure 18: A different view of the connected LEDs*



*Figure 17: The three LEDs with a wire attached to each one*

Now, connect each LED to the board.

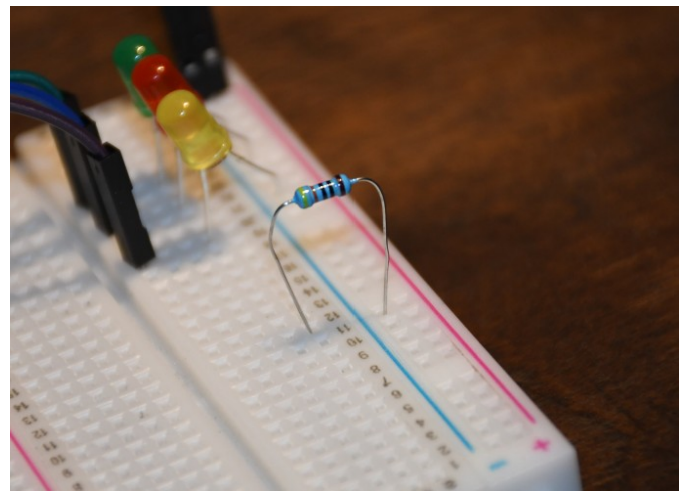
Connect the green ready LED to the “D27” pin on the ESP32. In this case, the green wire.

Connect the red alarm LED to the “D25” pin on the ESP32. In this case, the blue wire.

Connect the yellow silence LED to the “D26” pin on the ESP32. In this case, the brown wire.

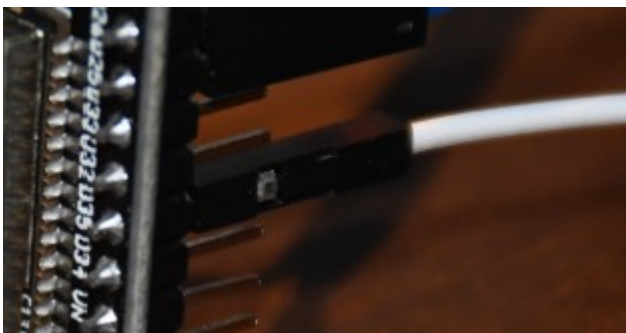
It is normal that none of the LEDs light up when power is connected, because ERROR 2 has still not been resolved.

Now lets wire up the buttons. Attach a resistor between the middle section of the breadboard and the negative rail. A 10K resistor will work fine.

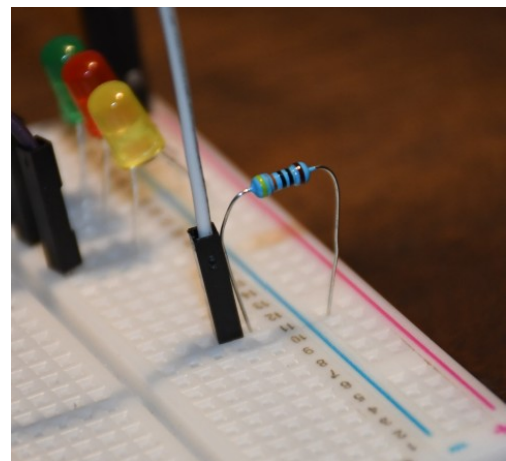


*Figure 19: The connected resistor*

Connect a new wire to the resistor. The first button being wired up is the silence button. Plug the wire from the resistor into the “D35” pin on the ESP32. In this case, that is the white wire.



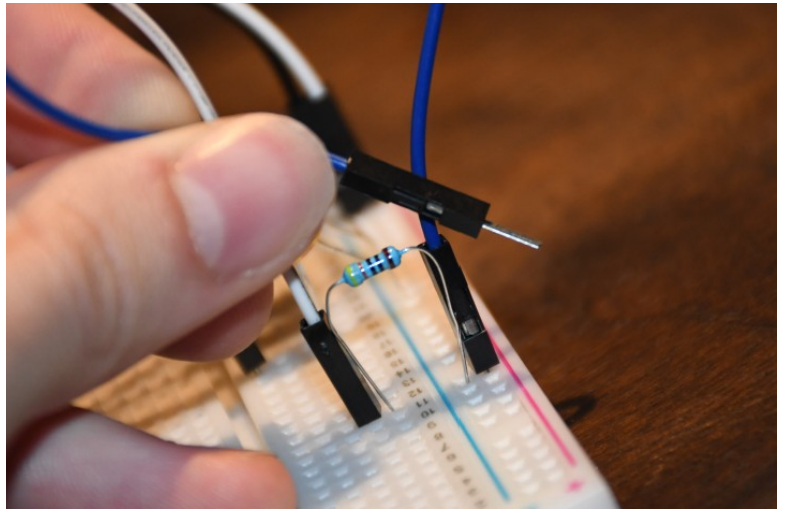
*Figure 20: The white wire from the resistor connected to the ESP-32*



*Figure 21: A new wire, electrically connected to the resistor as it is in the same column as it.*

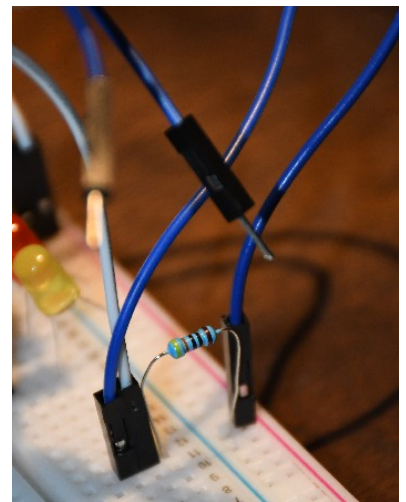


Connect a new wire to the positive rail of the breadboard. In this case, the blue wire. This first blue wire will be connected to one pin of your button.

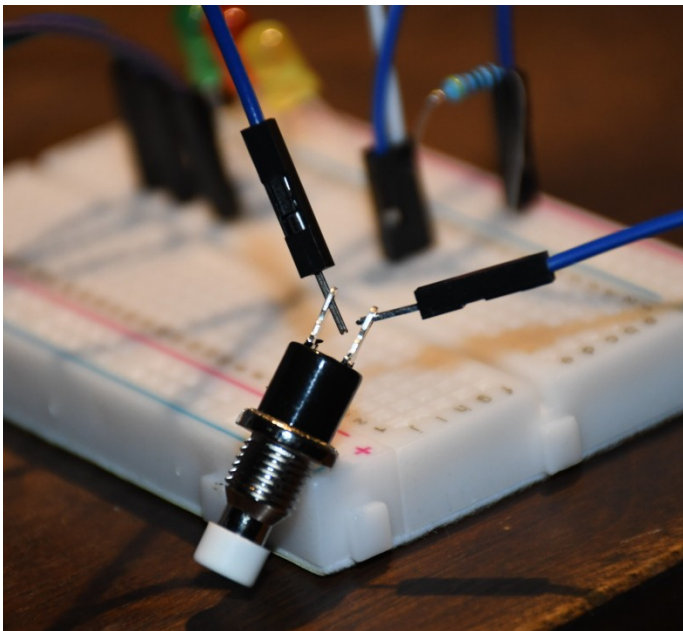


*Figure 22: The new wire attached to the positive rail. The other end is being held.*

Connect a second wire that is also electrically connected to the resistor and the “D35” pin of the ESP32. Ideally, you would want to connect these two blue wires to a button. I am too lazy to do so, but this would be what it would look like.



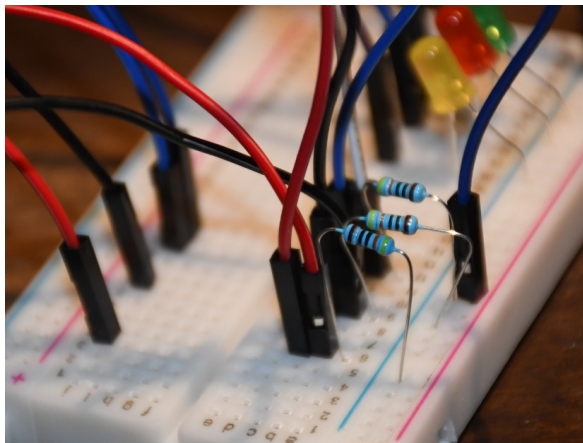
*Figure 23: The two blue wires. Both ends of these blue wires would ideally be connected to a button.*



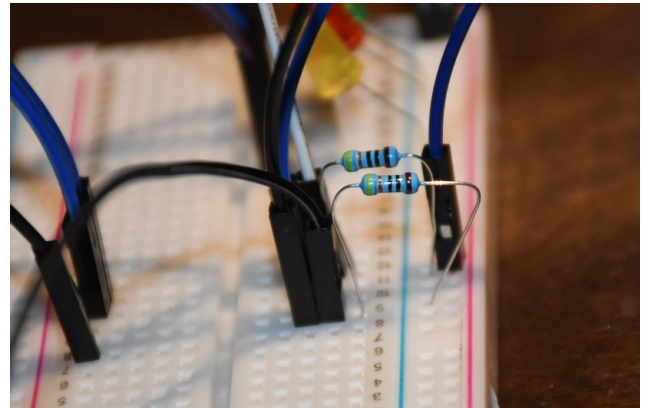
*Figure 24: The two blue wires connected to a button*

Repeat the previous four steps to create two more buttons. The reason that there is only one wire on the positive rail even though there are three buttons is because you can share the positive wire among all the buttons.

Something worth noting, the connections on the left side of the board lead to nowhere, I just needed a place to put them so they wouldn't go everywhere.

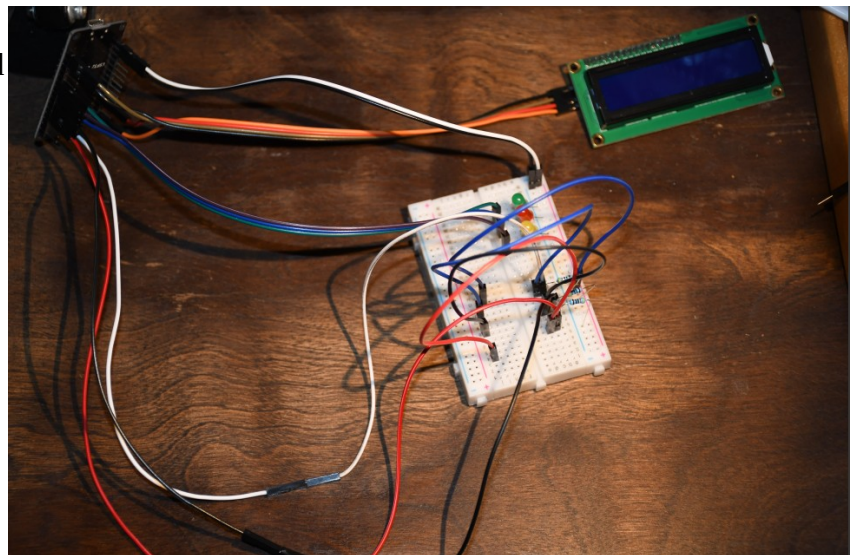


*Figure 26: The third button*



*Figure 25: The second button*

This is what your build should look like now. You most likely want the buttons to be ordered as Silence – Reset – Drill. This will make the menu controls more intuitive. At this point, you can boot up your panel and press the drill button to write the default settings to the panel. Although, since you haven't built the zones yet, the panel will almost immediately go into alarm.



*Figure 27: Progress on the build*



## 5. Connecting the Relays

The relay board is responsible for controlling the horns, strobes, and smoke detectors. This is a four channel relay board, because I couldn't find a three channel relay board.

Connect the "GND" pin of the relay board to the "GND" pin on the ESP32.

Connect the "VCC" pin of the relay board to the "VIN" pin on the ESP32. Since you already have the LCD taking up the "VIN" pin, simply replace that with a wire going to your breadboard. You can then connect both to the "VIN" pin.

Connect the "IN1" pin to the "D13" pin on the ESP32. This is the horn relay. You will want to wire the horn pin of your alarm to this relay. If you have a two wire alarm with the horn and strobe power integrated into two wires, you will also want to connect that device to this relay.

Connect the "IN2" pin to the "D18" pin on the ESP32. This is the strobe relay. You will want to wire the strobe circuit of your four wire alarms to this relay.

Connect the "IN3" pin to the "D14" pin on the ESP32. This is the smoke detector relay. You will want to wire the power of your four wire smoke detectors to this relay.

I do not have an extra relay board, so I will just use LEDs to represent the relay board. The white LED is the strobe relay, connected to the "D18" pin. The red LED is the horn relay, connected to the "D13" pin. The blue LED is the smoke detector relay, connected to the "D14" pin.

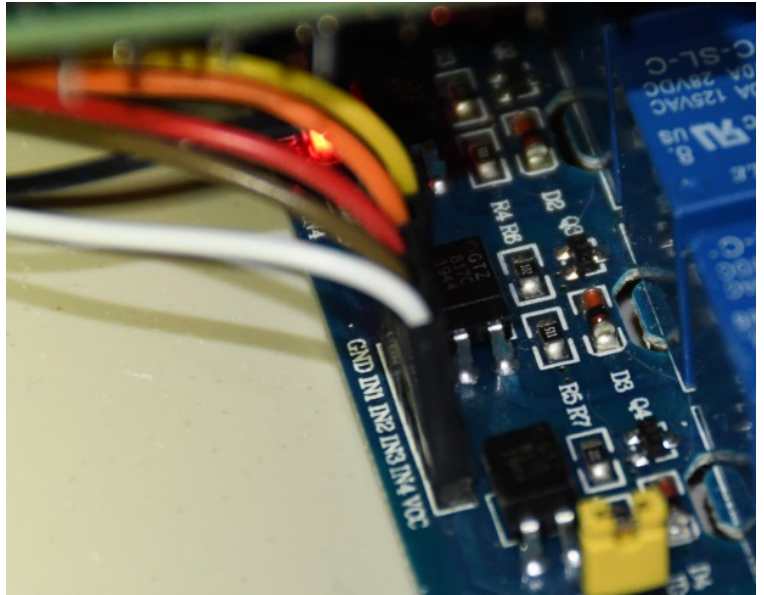


Figure 28: The relay board in the first Antigoneous panel

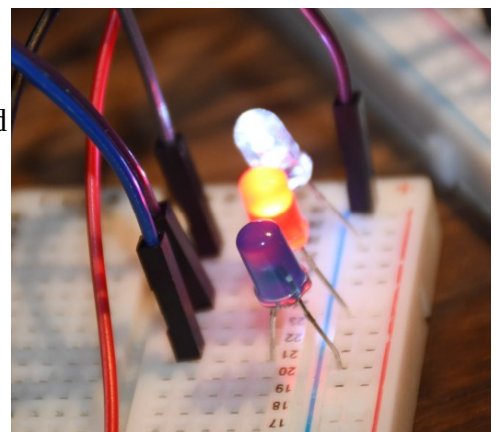


Figure 29: My substitute relay board

## 6. Creating the Zones

This is the final and most complicated step of the electronics. The wiring of this is slightly complicated, and I would recommend building it on a breadboard first, so you can see what's going on.

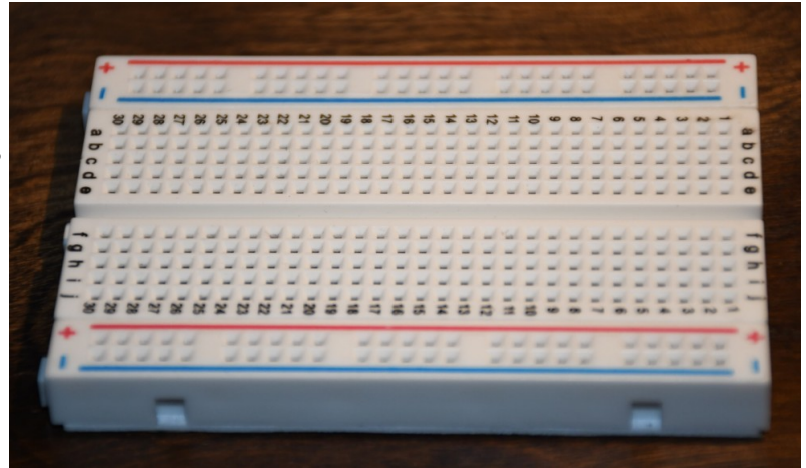


Figure 30: A new breadboard

Connect a new wire to the positive rail of the breadboard. Something worth noting, I have not connected the positive and negative wires attaching the breadboard to the ESP32 electronically. But that will be done in a later step to reduce confusion.

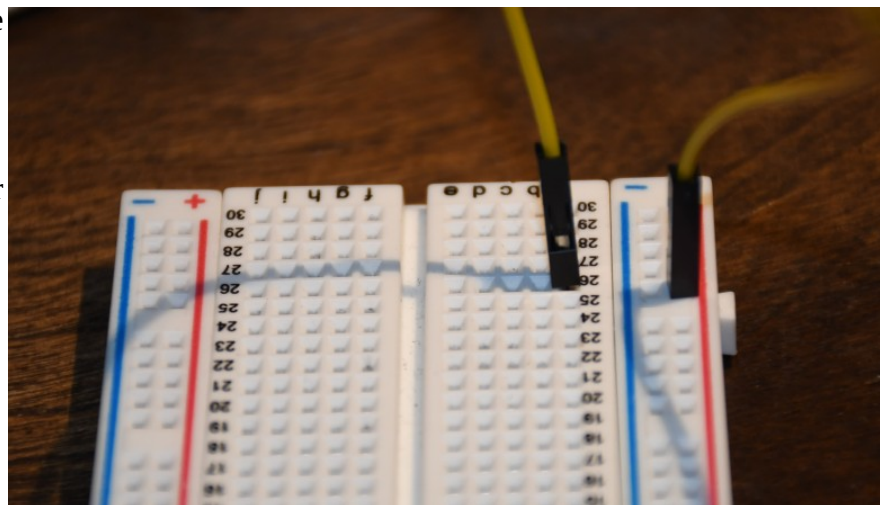


Figure 31: A new wire connected to the positive rail

Connect a new resistor to the breadboard as shown in the image.

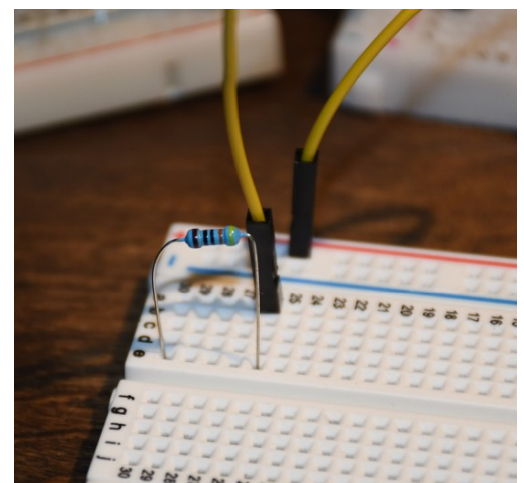
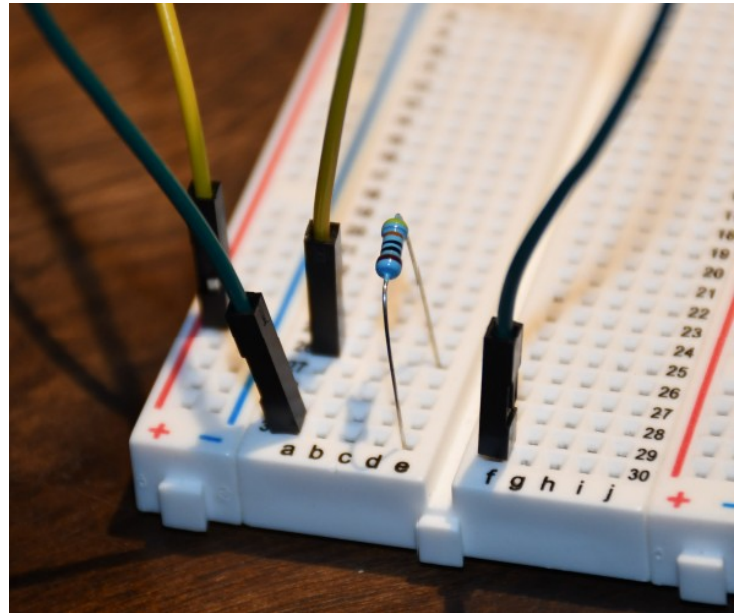


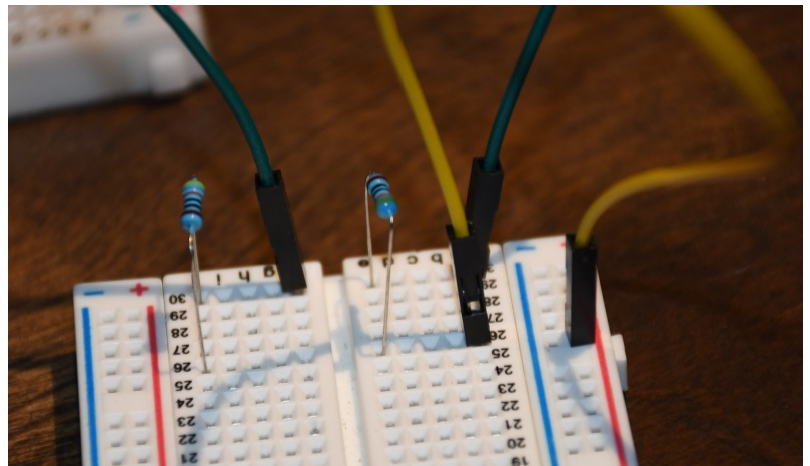
Figure 32: A new resistor attached to the breadboard, electrically connecting two columns

Connect a new wire from one side of the breadboard to the other. Connect it in such a way that it electrically connects the other side of the resistor to a new column of the other side.



*Figure 33: A new green wire connecting the resistor to the other side of the board*

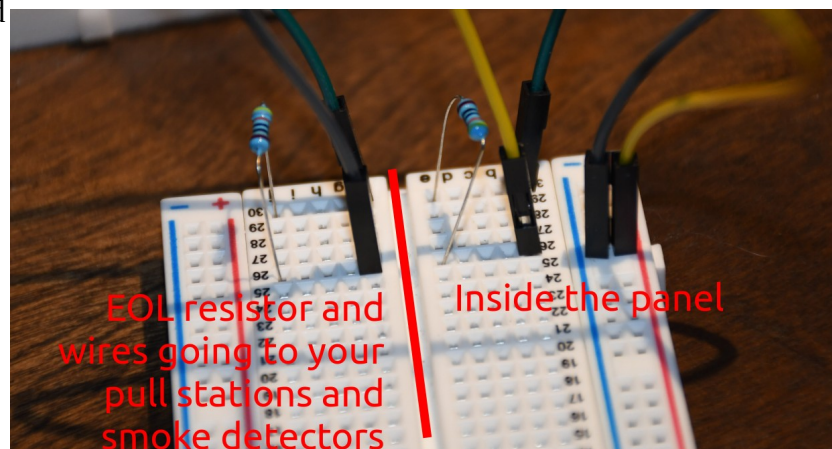
Add a new resistor. It should be electrically connected to the other resistor. In this case, through the green wire.



*Figure 34: The second resistor added to the other side.*

Connect the other side of the most recently placed resistor to the negative rail. In this case, through the gray wire.

**The left side of the breadboard contains the two wires leaving the control panel and the end-of-line resistor that you wire to your pull stations and smoke detectors. The right side of the breadboard contains the components that are inside the panel.**



*Figure 35: An explanation of what is inside or outside the panel*



Now here come the arrows. Place a new wire that is electrically connected to the green wire we placed a three steps ago. Plug the other side of that wire into pin “D15” on the ESP32. This will be our Zone 1.

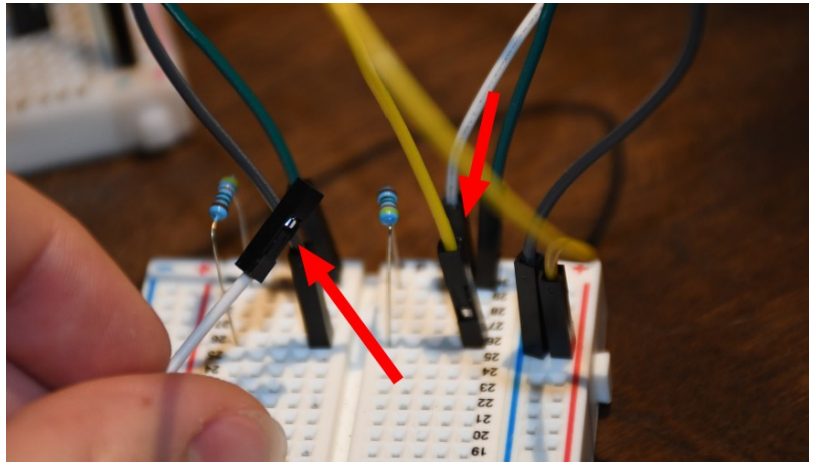


Figure 36: The white wire that will plug into the ESP-32

Repeat the previous six steps a second time to make Zone 2. This time, connect the final wire to pin “D39” on the ESP32, this pin is also labeled as “VN”.

You can optionally attach a buzzer to pin “D4”.

Connect the board to power, press the drill button to write default settings.

## YOU ARE DONE!

Now that you know the electrical connections of the panel, you can move things around however you see fit. Don't just stick with the design in this assembly guide, be creative, make your own enclosure for your new panel.

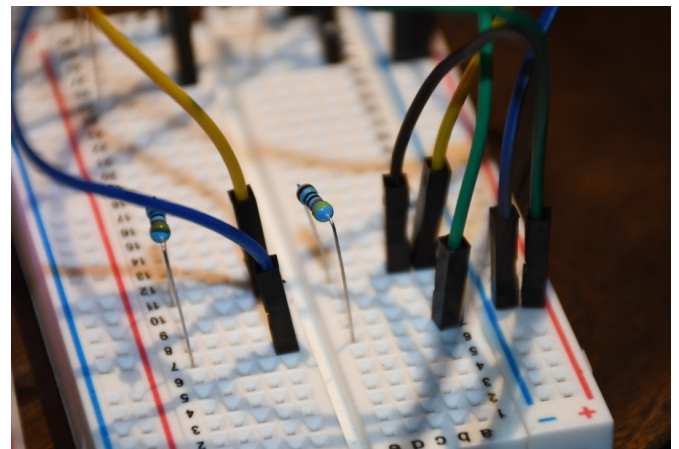


Figure 37: Zone 2, connected to pin D39 on the ESP-32

## Using your New Panel

Congratulations! You now have an Antigenuous panel. But what do you do next?

You need to configure the panel to work best with your setup. You also want to familiarize yourself with the interface.

### Entering the menu

By default, just-flashed Antigenuous panels have no electronic lock to prevent tampering. This can, of course, be enabled later. But for now you may find it easier to keep the lock off until you are done setting up the panel.

You can **enter the menu** by pressing the **silence button**.

From this menu, you can control the operation of the panel.

**The controls to navigate the menu are as follows:**

**Silence** – Return to previous screen

**Reset** – Scroll through current screen

**Drill** – Enter sub-menu **OR** perform action such as changing a setting

### The Testing and Settings menu

There are two options within the menu, **testing** and **settings**.

As the names imply, the **testing menu** allows you to run various tests such as **walk tests, silent walk tests, and strobe tests**.

The **settings menu** is where you can change the various **panel settings**.

There are two sub-menus within the settings menu, **Fire Alarm** and **Panel**.

The **Fire Alarm** sub-menu gives you access to settings that change how the notification appliances operate, such as the horns and strobes.

### Fire Alarm Settings

**Coding** – Changes the audible pattern of the alarms. If you use alarms that have their own audible pattern, you may wish to use “continuous,” as this does not pulse the alarm relay.

**Verification** – This is a sub-menu that controls how the panel acts regarding alarm verification. In other words, how the panel double-checks to make sure a pull station is pulled or a smoke detector isn’t reporting a false alarm.



**V. Time** – This controls how long the panel waits before double-checking the devices to see if it should really activate the alarms.

**Det. Verif.** – This enables or disables smoke detector verification. In short, smoke detector verification cuts power for a specific amount of time to a smoke detector that has triggered, then waits a specified amount of time, if the smoke detector re-alarms within that set period of time, the panel goes into alarm.

**Det. Timeout** – This changes the amount of time that the panel cuts power to the smoke detectors after an alarm is reported.

**Det. Watch** – This changes how long the panel will wait for a re-alarm before resetting the verification. If the smoke detector detects smoke during this period of time, the alarms will go off. If the time elapses with no smoke detected, then the smoke detector is assumed to have reported a false alarm.

**Pre-Alarm** – This is a sub-menu that controls how the panel deals with pre-alarm. Pre-alarm is an alarm condition that flashes the strobes as normal, but pulses the horns in small chirps for a specified amount of time, or until the **drill button** is pressed.

**Stage1 Time** – This changes how long stage 1 will last. Stage 1 is the stage in which the horns are sounded in small chirps. After the amount of time specific here, the panel will enter stage 2, and the horns will go into full alarm, using the coding specified by the user.

**Audible Sil.** – This changes if the panel will keep the strobes on once the panel is silenced. If this is disabled, power is cut to both the horn and strobe relays once the **silence button** is pressed. If this is enabled, pressing the **silence button** will disable the horns, but will keep the strobes on.

**No-Key Sil.** – This changes if the panel is able to be silenced without a key to electronically unlock the panel. If this is enabled, the panel can be silenced if the alarms are sounding. Nothing else can be done with the panel without the key, including resetting the panel and entering the menu. This setting *only* allows for it to be silenced.

**Strobe Sync** – This changes the strobe relay to use a strobe sync pattern to sync all the strobes on a circuit.

**2 Wire** – This changes if the panel uses four wires or two wires for the alarms.

## Panel Settings

**Panel Name** – This allows for a custom name to be set. This custom name displays on the home screen if no troubles or alarms are reported. Press the **reset button** to change the character of the selected box, then press the **drill button** to move forward once space. If you reach the end, the cursor will move to the beginning. Press the **silence button** to save and exit the panel name menu.

**Panel Security** – This allows the use of an electronic lock to be used. Be careful with this setting. Even though you have to reset the panel for this setting to take effect, if you enable the keyswitch lock

without having a keyswitch, you will either have to install a keyswitch or factory reset the panel to regain access.

**LCD Dim** – This setting changes if the panel LCD will automatically dim after a specified period of time.

**Factory Reset** – This allows the panel to be reset to factory defaults. When you press the **drill button** on this option, you will have to confirm the factory **reset** by pressing the **reset button**, or you can **cancel** by pressing the **silence button**.

**About** – Displays information about the panel such as the firmware revision.

## Fail-safe Mode

Fail-safe mode is a specific mode included with the Antigoneous panel. If your panel fails to boot, either from an error loading settings or something else. Fail-safe mode is intended to be a completely separate environment that is designed to boot up without fail. It does not require any interaction with the saved settings, and skips almost all of the boot-up code.

When the panel is in fail-safe mode, features are extremely limited. The only thing that works is the horn relay, strobe relay, smoke detector relay, silence button, and reset button. The reset button is the only thing that the user can interact with when the alarms are off, the reset button will restart the panel. When the alarms are on, the silence button can also be used to perform an audible silence. The coding on the alarms in fail-safe mode is continuous, and there is no verification or end-of-line resistor checks.

Fail-safe mode is not ideal to be in, and as the name suggests, is simply a mode that the panel can be in that ensures there is always fire detection, even if the panel is entirely unable to boot normally. Fail-safe mode should absolutely not be considered normal operation, and the reason for the panel not booting normally should be found and fixed immediately.

## Error Codes

Antigneous Error Codes		
Error Code	Meaning	Solutions
<b>1</b>	The panel's settings have failed a validity check and must be reset to factory settings.	<p>Press the <b>reset button</b> to boot into the fail-safe mode.</p> <p>Press the <b>drill button</b> to reset the panel to factory settings.</p>
<b>2</b>	The panel's firmware has been updated to a new version, and must be reset to factory settings in order to add new settings into the panel EEPROM.	<p>Press the <b>reset button</b> to boot into the fail-safe mode.</p> <p>Press the <b>drill button</b> to reset the panel to factory settings.</p>