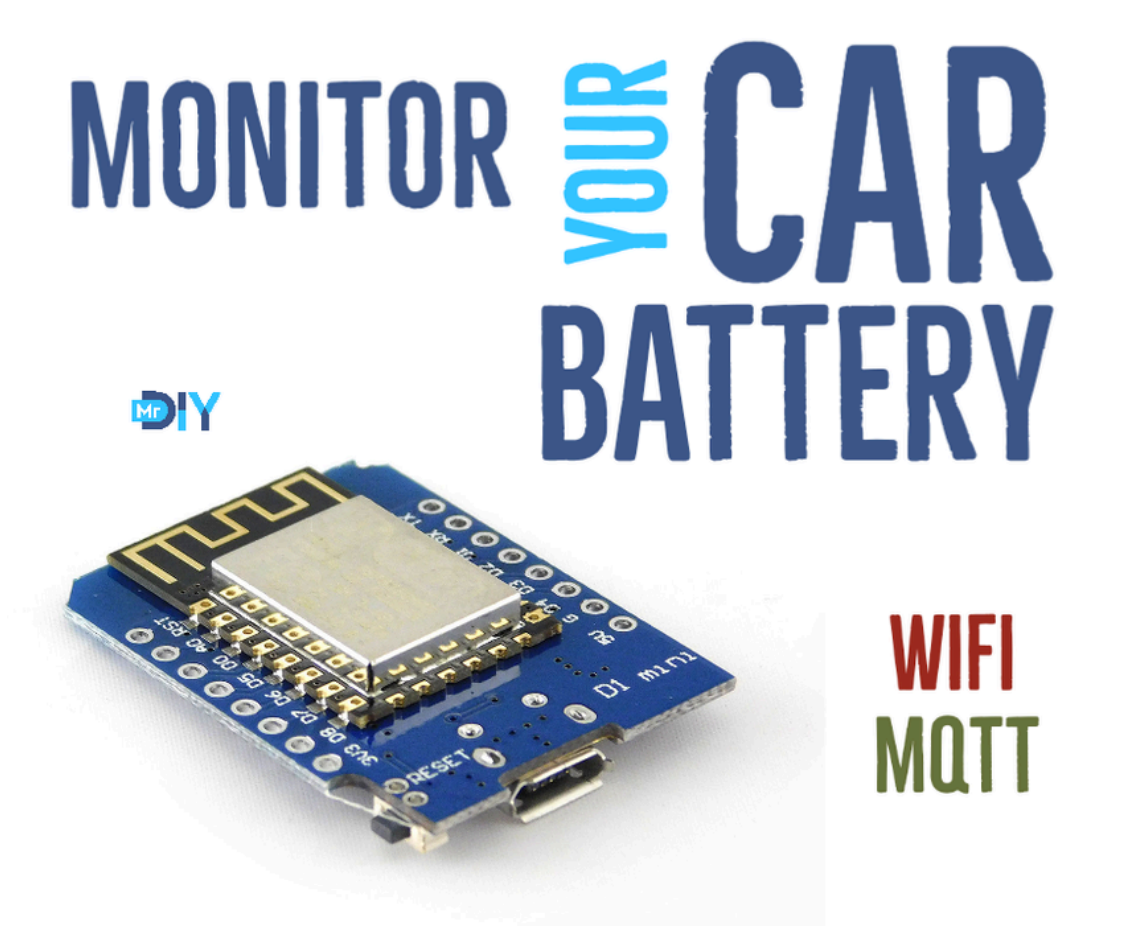
AUTODESK Instructables

DIY: Monitor Your Car Battery: Code & Setup

By MrDIYLab in CircuitsSensors

Introduction: DIY: Monitor Your Car Battery: Code & Setup



Having the ability to monitor your car battery can prevent some unpleasant surprises. I will show you how I assembled the hardware, loaded the software and installed the monitor in my car. I will be using the ESP8266 Board called Wemos D1 Mini.

New to ESP8266? Watch my Introduction to ESP8266 video first.

Step 1: Watch the Video

The video has step by step instructions that will guide you through the process. Feel free to add your questions in the comment section of the YouTube video if you need any futher assistance.

Step 2: Order the Components

Buy on Amazon.com

- Wemos D1 Mini - https://amzn.to/308Sxoy
- Power Shield - https://amzn.to/3gbfqxq
- Assorted Resistors - https://amzn.to/2PagR34

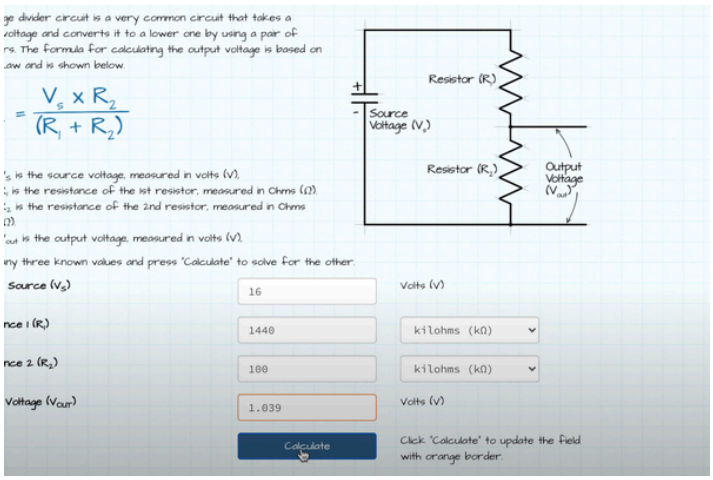
Buy on AliExpress:

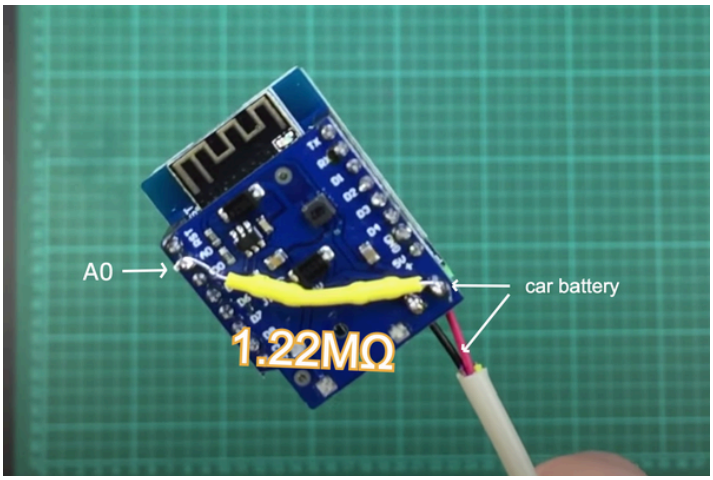
- Wemos d1 mini - https://s.click.aliexpress.com/e/_AEZGUP
- Power Shield - https://s.click.aliexpress.com/e/_AIE2SF
- Assorted Resistors - hhttps://s.click.aliexpress.com/e/_AAfyJV

Buy on Amazon.ca

- Wemos D1 Mini - https://amzn.to/3cpOxDe
- Power Shield - https://amzn.to/2XuoIxD
- Assorted Resistors - https://amzn.to/2AtfU1P

Step 3: Hardware



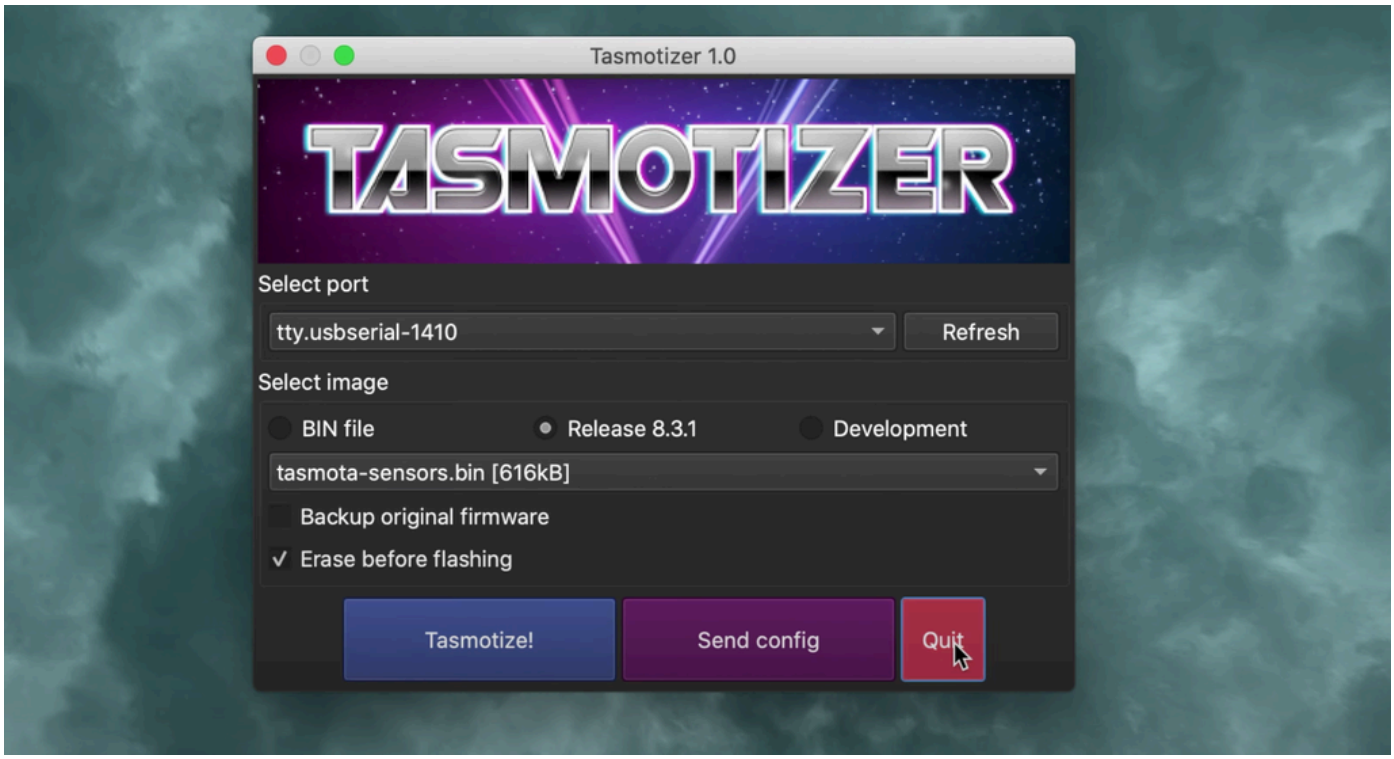


You will need a wemos d1 mini, a power shield and some resistors. First, I started by removing the power plug and installed the smaller connector to make the hardware more compact.

The D1 mini can measure external voltage up to 3.3v by using a voltage divider using R1=220KΩ & R2=100KΩ . This the voltage within the 0-1 Volt that the ADC can tolerate. To increase the 3.3v to 16v needed for the car battery, we need to increase R1 to 1.44MΩ. To do that we can add another 1.22MΩ in series to get the total 1.44MΩ. I did this by soldering a 1MΩ resistor to a 220KΩ resistor as shown here.

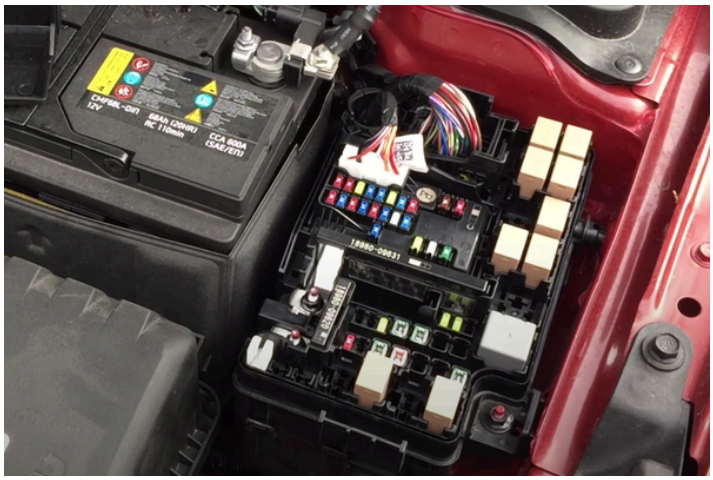
I attached a long wire to the power input terminals to be able to connect them to the car battery.

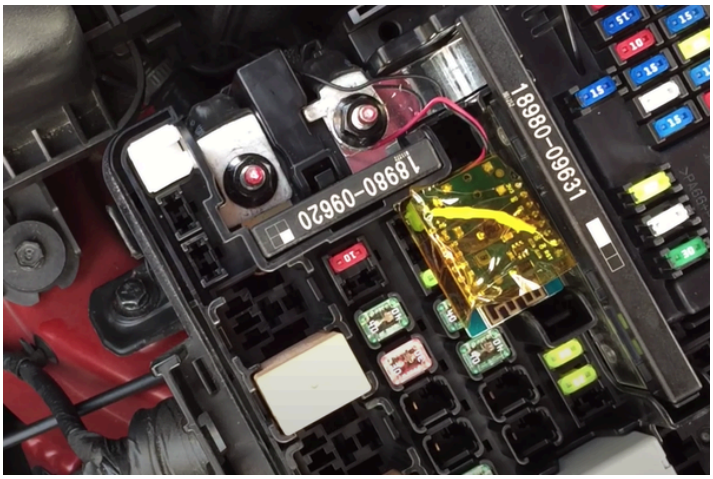
Step 4: Software



I then connected the D1 mini to my laptop and load the software. Make sure you select the sensor.bin version for the Analog input functionality. I then resumed with a typical Tasmota configuration.

Step 5: Car

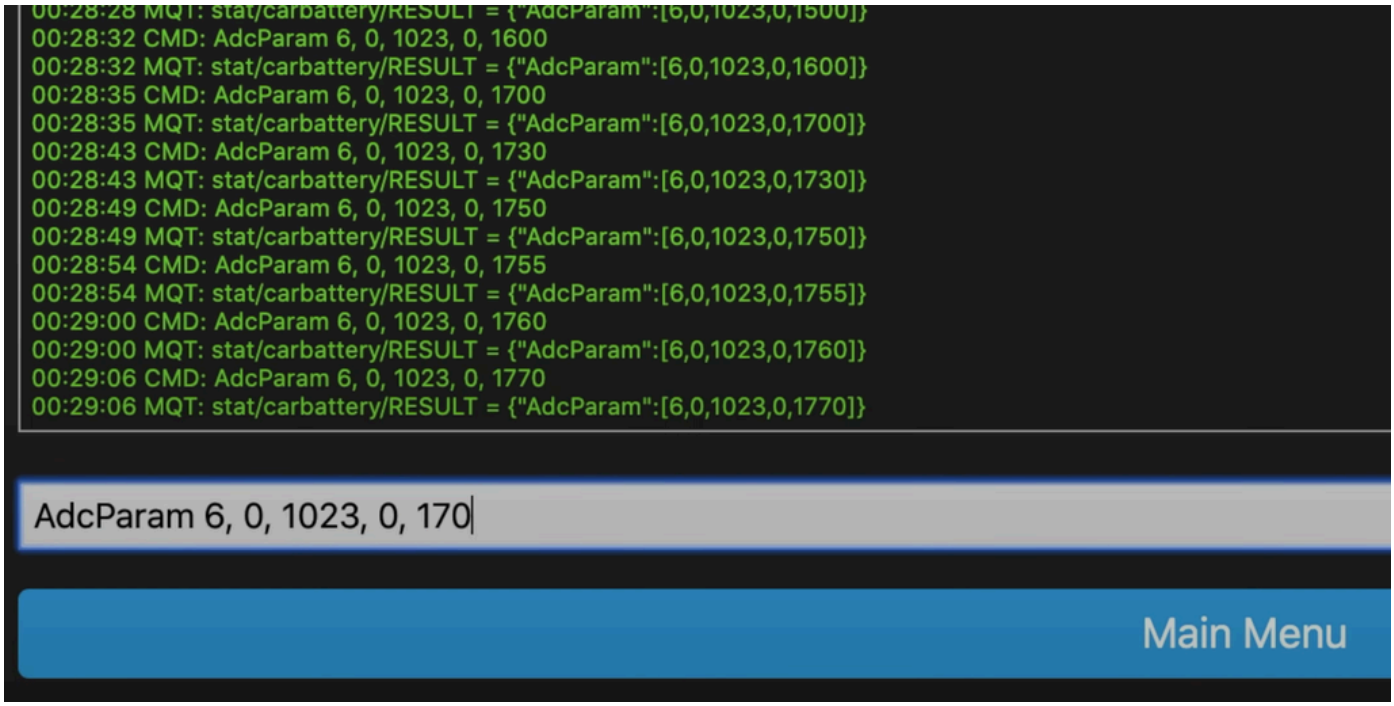




At the car, I opened the hood and located the fuse box. I found the fuse box to be a safe and secure place to install my device.

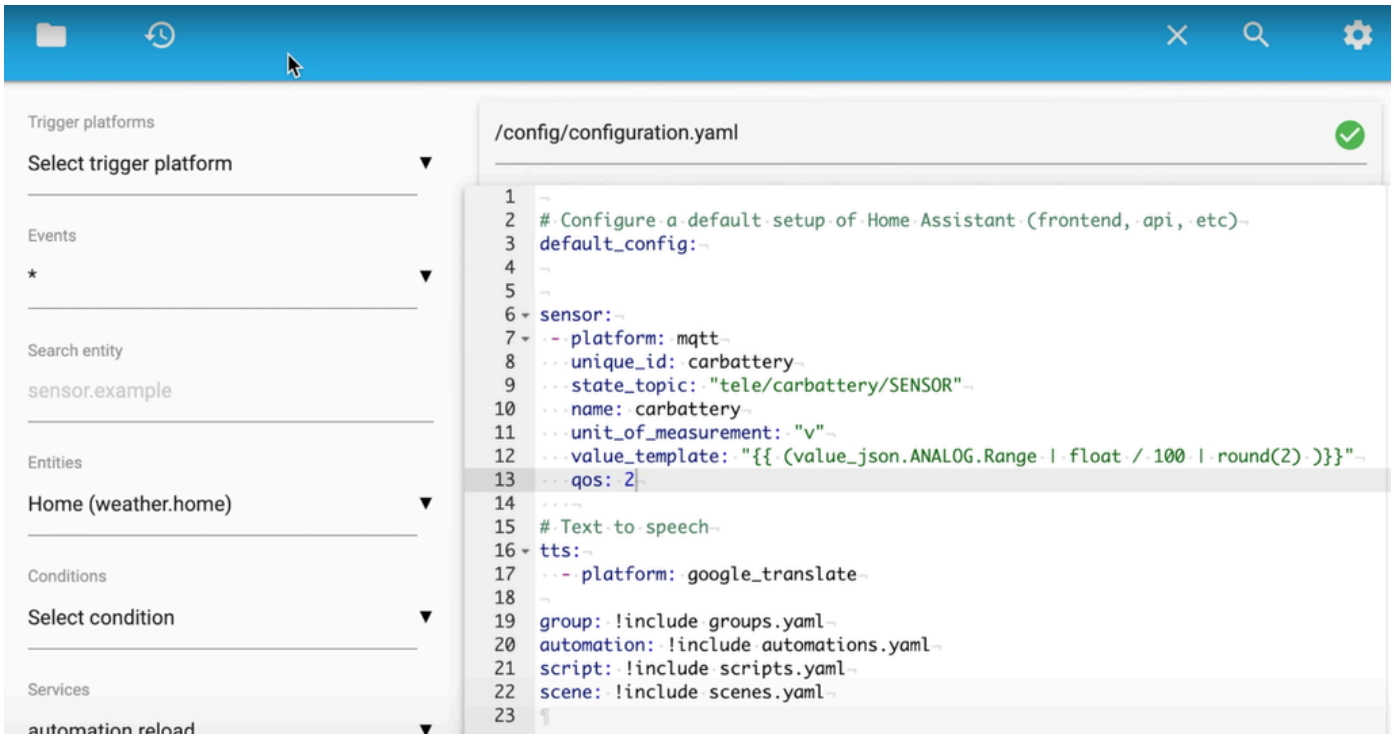
I first wrapped the device in heat resistant tape to cover any exported pin to withstand the engine heat. Since the entire car chase is ground, I found the nearest screw and connected my ground to it. Next, I located the nearest connection to the battery positive rail and connected my positive power input to it.

Step 6: Calibrate



I then proceeded with calibrating the range of the analog input. I did this by connecting a multimeter to the battery and read the current voltage (in my case it was 12.73 volts). I then did some trial and error until my reading was 1273 as the analog reading. The last step was closing the fuse box and the car hood.

Step 7: Home Assistant



Back in Home Assistant, I opened the **configuration** file and added a new MQTT sensor - the code is attached above.

After I saved, I restarted home assistant for it to take effect. When it came back online, I added the new sensor to the dashboard.

Step 8: Done

The integration is now complete. You can now use this sensor to trigger alerts!

If you found this useful, please consider subscribing to my YouTube channel - It helps me a lot.
If you are interested in supporting my work, you can check my Patreon page.

Much of the information contained is based on personal knowledge and experience. It is the responsibility of the viewer to independently verify all information.