PiJuice Weather Station Project

PiJuice Weather Station Project

This project will show you how to build a basic weather station using some easy-to-source components, the Raspberry Pi and powered by only the PiJuice portable battery solution and compatible solar panels. This project will be a truly off the grid solution to powering your projects all day and night, taking valuable sensor readings of the local atmospheric weather around you.

The project will be built on a breadboard to test that the circuit is working and you are able to get readings from the sensors and then it will be ported over to a fixed Pi Crust Pro prototyping HAT.

- What you will need
- Hardware Setup
- Installing the software
- Auto Installation
- Manual Installation
    - Adafruit DHT22 Library
    - Flask
    - Weather station project install
    - PiJuice Software
- Changing web page refresh rate
- JustGauge Javascript Plugin

What you will need

You will need the following parts to build this project. Some optional parts are required for the initial setup on the Raspberry Pi:

- Raspberry Pi Computer 2, 3 B or 3 B+
- 8GB MicroSD card with the latest Raspbian Stretch installed
- Mouse and Keyboard (Initial Setup)
- Power supply 2.5A
- Pi Crust Pro (optional)

- Breadboard
- PiJuice HAT
- PiJuice Solar Panel (optional)

Below is a list of electronic components required:

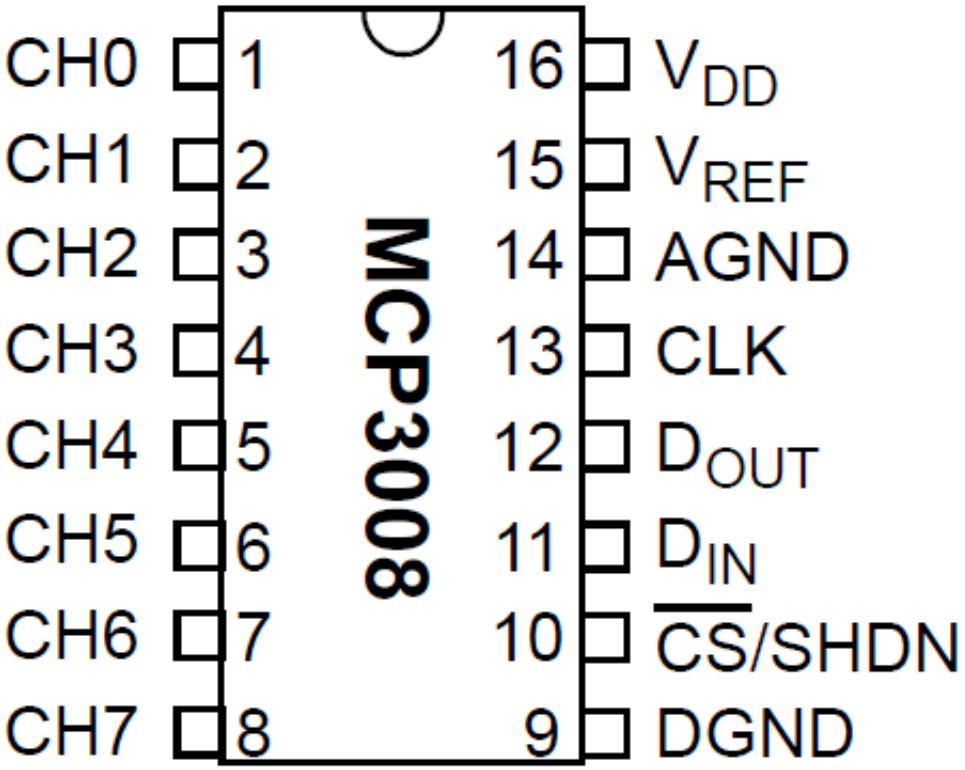
- MCP3008 ADC
- LDR Light Sensor
- LM35 Temperature Sensor
- DHT22 Humidity Sensor
- TGS2600 Air Quality Sensor (optional)
- Outdoor IP rated junction box (Optional)
- Jumper wires
- Single Gauge cable (Optional)

Hardware Setup

First we are going to prototype all the sensors on our breadboard to make sure that we have connected the sensors to the Raspberry Pi correctly and we are able to take accurate readings without any issues. We will connect each sensor in stages as it is easier to debug when things don’t go as planned.

Since the Raspberry Pi is primarily a digital system we need a way to interface with our analog sensors. To do this we are going to use a chip called MCP3008, which is an analog to digital convertor. To connect the MCP3008 take a look at the below circuit diagram:

The MCP3008 chip will have a small indent circle on the top, this will indicate which is pin 1. Make sure that you get the orientation correctly when inserting the MCP3008 into the breadboard. Below is the pinout for the MCP3008 IC:



You can follow the guide below for the pinout of the Raspberry Pi and the MCP3008:

MCP3008	**Raspberry Pi**
VDD	3.3V Pin 1
VRE	3.3V Pin 1
AGND	GND Pin 6
DGND	GND Pin 6
CLK	SCLK Pin 23
DOUT	MISO Pin 21
DIN	MOSI Pin 19
CS/SHDN	CE0 Pin 24

You will notice that the pins on the left most side of the MCP3008 are labelled CH0 to CH7 and these pins are used to connect your analog sensors, up to 8 in total.

Now let's connect the light sensor to the MCP3008 CH0, which is pin 1. You will need to connect one side of the LDR to ground and the other to 3V3 via a 2.2K resistor and CH0, as shown in the circuit diagram below.

Now let's connect the LM35 analog temperature sensor. The LM35 has three pins in total, looking at the flat side of the temperature sensor the left most pin must be connected to 5V, the middle pin to CH1 and the rightmost pin to ground.

Lastly, we can now connect the DHT22 humidity sensor. This sensor is a digital sensor, therefore we can connect it directly to the Raspberry Pi GPIO header. Insert the DHT22 sensor into the breadboard, looking at the grid of the plastic housing there are four pins in total on the underside. Connect pin 1, which is on the left most side to 3.3V. Pin 2 goes to GPIO 4 and 3V3 via a 10K resistor. Pin 3 is left disconnected and pin 4 connects to ground.

That's it for the circuit, we have connected all the components to the Raspberry Pi either directly using digital components or via the MCP3008 using analog components. In this next section we will test each sensor to make sure that we have connected it correctly and we are able to receive the data from each sensor.

Installing the Software

This project runs on the latest version of Raspbian OS for the Raspberry Pi. Make sure you run **sudo apt-get update** before installing the following libraries. You will need to run the following commands in the terminal window to install the libraries for the weather sensors.

Auto Installation

Just run the following line in the terminal to automatically install all the libraries and project files to the Raspberry Pi.

Raspbian OS Install Script:

Raspbian Lite Install script:

Manual Installation

Adafruit DHT22 Library

Flask

Flask is a lightweight web framework that runs using Python programming language. We will be using Flask to create a web server that can host a web page locally on the Raspberry Pi and then can be accessible over the network from any other device on that same network.

By default Flask is already installed on the latest version of Raspbian OS, however if the package is not there then you can type the following in the terminal window to install Flask:

Flask had a specific file structure that needs to be met in order for all the files to be located for the web server. Here is the file structure in its simplest terms:

- app.py
- config.py
- requirements.txt
    - static/
        - css/
            - style.css
        - javascript/
    - templates
        - index.html

For further information visit http://exploreflask.com/en/latest/organizing.html

Weather Station Project Install

To download the weather station project files to your Raspberry Pi type the following in the terminal window:

To run the Flask web server:

To view the webpage you will need to go to the Raspberry Pi's hostname address on your local network such as http://raspberrypi.local:5000 or using the Raspberry Pi's local IP address. You can find your IP address from the terminal window on the Raspberry Pi by typing in the following command:

PiJuice Software

Installing the PiJuice software will allow you to configure PiJuice HAT to do safe shutdowns when the battery is low and then boot the Raspberry Pi once charged. You can also do various other configurations such as timed shutdowns and boots depending on your circumstances.

PiJuice GUI:

PiJuice CLI:

Changing web page refresh rate

By default this project has the ability to refresh the web page every 5 seconds to get the latest update value from the sensors. You can change the refresh rate in the index.html page by changing the content value in the following line:

JustGauge Javascript Plugin

Justgauge is a handy JavaScript plugin for generating and animating nice & clean gauges. It is based on Raphaël library for vector drawing, so it's completely resolution independent and self-adjusting. This plugin is a nice clean way to display the values from our weather station in its simplest form.

The JavaScript files have already been added to the project files in static/javascript/ . To add JavaScript to the index.html file you must do so in the following format not the regular html format:

To add a new gauge you will need to create a new div with id and class.

Add the following to the css file to style each gauge element. The id element will change the individual gauge block and as such will require a unique id where as the class will style all gauge elements the same. Note: currently the id is not used in this project.

```
.gauge {
      width: 300px;
      height: 300px;
      display: inline-block;
      margin: 5px;
  }
```

Finally add these parameters changing the value to the variable that gets passed through from the Python script.

```
 var gauge = new JustGage({
   id: "gauge",
   value: {{temp}},
   symbol: '\u2103', #can be added as plain text or hex code
   levelColors: ['#1B94FF', '#FDDC00', '#FF9E00', '#FF3F00'],
   min: 0, #minimum range value
   decimals: 1, #decimal places
   max: 50, #maximum range value
   title: "Temperature" #text to be displayed above the gauge
 });
```

PiJuice Configuration

The weather station program also includes a battery level indicator to the web interface so you can keep track of when the battery level gets low. You can also configure the PiJuice to automatically shutdown and boot up when the battery gets as low as 5% and then boot back up when it charges back to 90%. You can configure these settings in the GUI or using the command line interface by running pijuice_cli .

PiJuice GUI

PiJuice CLI