

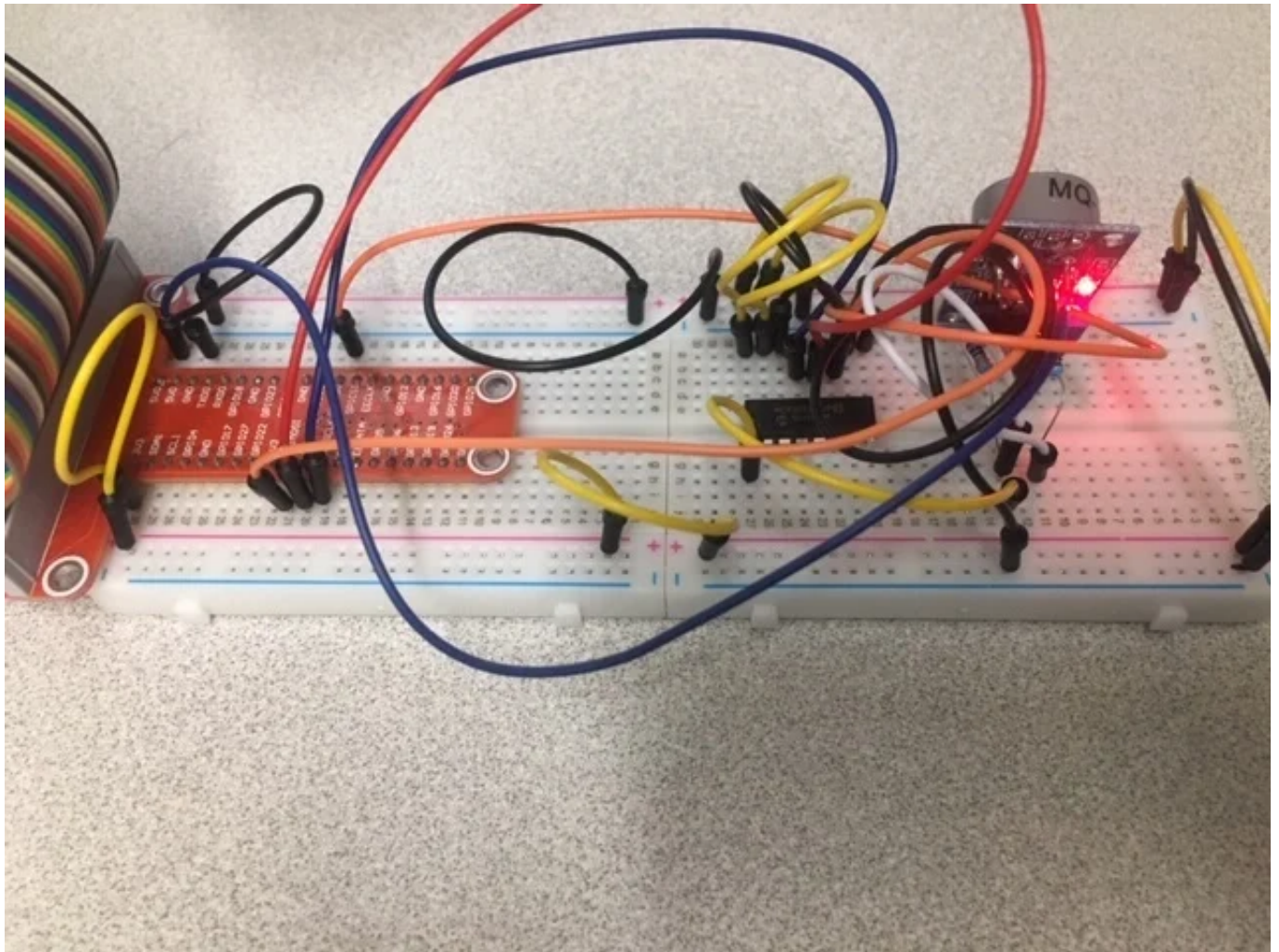
AUTODESK
Instructables

Raspberry Pi Carbon Monoxide Sensor

By [jeanet_g1](#) in [CircuitsRaspberry_Pi](#)



Introduction: Raspberry Pi Carbon Monoxide Sensor



Internet of Things are those devices which are connected to the internet and have the features of self-adaptability, self-configuring, Interoperable Communication protocols and have the unique physical and virtual entities. Sensors are those devices which measure some physical and environmental features and are mainly used to collect data in the IoT devices. For the purpose of this project we chose the Carbon Monoxide sensor to measure the Carbon Monoxide level present on the environment. The device we built can be used in cars to detect the presence of carbon monoxide inside the car, this is very important in real life because the

We use cookies to enhance your experience and personalize content. See our [cookie statement](#) for details. [Manage your cookie preferences.](#)

Decline

Accept

Supplies

Raspberry Pi 3

MQ-7 Carbon Monoxide Sensor

Breadboard

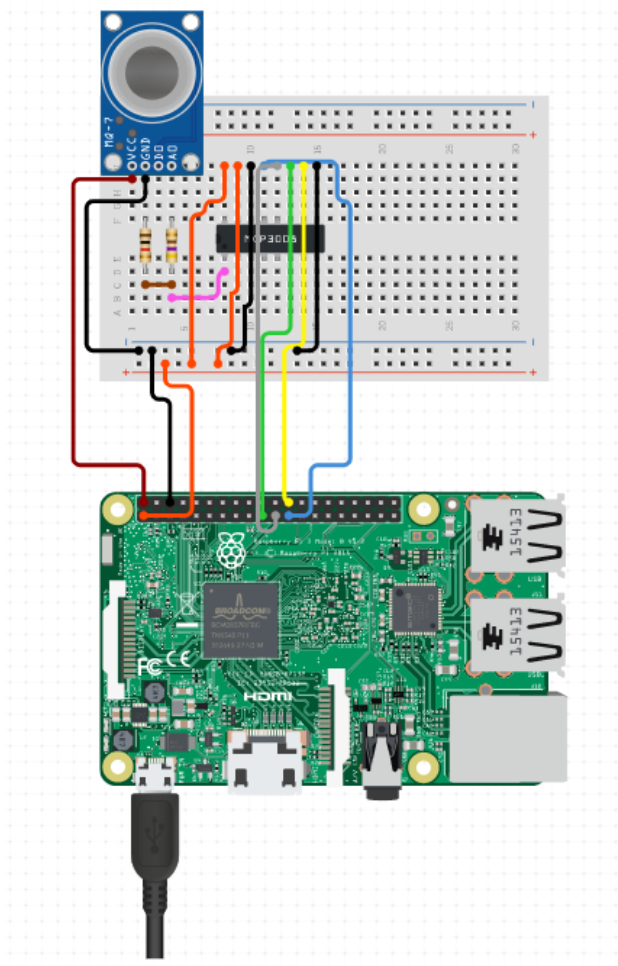
1K Ohm Resistor

470 Ohm Resistor

MCP3008 8 Channel, 10-bit ADC with SPI Interface

Computer

Step 1: Setting Up the Pi



Follow the schematic diagram to set up your circuit. For an interactive version of the picture, visit [circuitio.io's diagram](https://circuitio.io)

We use cookies to enhance your experience and personalize content. See our [cookie statement](#) for details. [Manage your cookie preferences.](#)

Decline

Accept

Step 3: Code

You can find all the source code for this project in our [github repository](#). Make sure when you download it to your local computer and to keep the file structure the same as you transfer it to the pi.

Open the `sensor.py` file and change the lines of code containing `localhost` to be the ip address of your computer. You can find instruction on how to find the ip address of your computer [here](#).

We need to move the **`sensor.py`** file to the raspberry pi, so run this command from the COSensor directory

```
scp sensor.py pi@"enter pi ip address here":.
```

We use cookies to enhance your experience and personalize content. See our [cookie statement](#) for details. [Manage your cookie preferences](#).

Decline

Accept

Step 4: Configuring Text Messages

Since our program alerts us through text messages when the Carbon Monoxide levels become too high, we need to be able to send messages from a central number. To do this, we will use a platform called Twilio. First, [sign up](#) for a free trial account. Next, we need to download a few packages. If you already have Node.js installed with version v8.0.0 or above, skip to step 2. You can check your version with:

```
node -v
```

The first package is Node.js which can be downloaded from their site [here](#). Make sure to install for your correct operating system, and then open it and follow the prompted instructions.

Next, we need to install the Twilio CLI. This is installed and updated with the Node package manager with the following commands:

```
npm install twilio-cli -g
npm install twilio-cli@latest -g
```

At this point, we need to connect the Twilio CLI to our account. To do this, we need two pieces of information: our Account SID and Auth Token from the [Twilio Console](#). Then run twilio login and enter the information prompted.

```
twilio login
```

So now we have connected our account but still need a phone number. You can purchase one through Twilio with the trial money you have been given. After typing the command below, a bunch of number will pop up; pick one.

```
twilio phone-numbers:buy:local --country-code US --sms-enabled
```

Now, in order for Twilio to work in our program, we need to install its packages. Type

```
pip install twilio
```

Inside the keys.py file, we need to enter in our Account SID and Auth Token for later use. There should already be a blank spot for you to copy and paste these values.

```
passwords = {"twilio":{"account_sid": "paste your sid here",
    "auth_token": "paste your token here"}}
```

With all this done, it is now time to set up email compatability with our program through the Gmail API.

We use cookies to enhance your experience and personalize content. See our [cookie statement](#) for details. [Manage your cookie preferences.](#)

Decline

Accept

Step 5: Configuring Gmail API

In order to configure the gmail API, first of all you need to visit the [google dashboard](#). Here you can register the new project using the 'create project' option. After the new project is created, there will be a prompt which says "You don't have any APIs available to use yet. To get started, please visit API Library".

Then visit [here](#). In the search box there select the Gmail API. After you click the Gmail API option, there will be an option to Enable this API. After you enable the Gmail API, you will need to create credentials for you to be able to use it. Therefore click on "Create Credentials", this will take you to the window asking you to select the API. There, select the Gmail API, then select the right option of where you will be calling this Gmail API from. After this you will need to select your role: something like product owner. Then the json file will be downloaded to your computer, which will be your service account, copy and paste this inside cred folder in your project directory. Hurray then your API is enabled and your account is registered to use this Gmail API. Now comes the fun part, say we want to send an email using your account registered with Gmail API. Visit [this website](#) for reference about code and how the code works to send email from the registered account. The first thing to remember is to define the SCOPE that allows you to send email. The scope to send email looks like: "https://www.googleapis.com/auth/gmail.send". You can find the list of authorization scopes [here](#)

Everything you do using the Gmail API like access the labels of email, or send the email, the new pickle token is created, this happens for the first time you run the application. Every time after that if you add a new scope the new token pickle is created, which allows for all the functions you can carry out using the gmail API. Each time you run your application changing the scope the new pickle token is created.

Step 6: Running the Sensor

Now we can finally run our program. Open an ssh sessions to your raspberry pi and in one run:

```
python3 sensor.py
```

On your computer, run

```
python api.py
```

Now, we can access the data and subscribe to receive notifications from the webpage. Open a web browser and type in <http://localhost:5000> to see current CO levels. Navigate to the subscribe page and enter in your information to receive notifications.

Now that everything is set up, you should receive notifications if CO is found, which hopefully doesn't happen.

We use cookies to enhance your experience and personalize content. See our [cookie statement](#) for details. [Manage your cookie preferences.](#)

Decline

Accept