Tutorial: Smart-Parking: IoT project

Requirements:

- Hardware
    - Raspberry pi (or Arduino)
    - Breadboard
    - LED's
    - Ultra-sound distance sensor
    - Resistors
    - Connector cables

- Software
    - A python script for reading the data on the pi and sending it to the cloud(script in repo)
    - A cloud account
    - An IoT platform and starter on the cloud
    - At least 1 cloud device
    - An application that receives the data and displays it

Connect hardware:

you could use an Arduino for this project but then you will need a power source since the sensor requires 5V.

1. first thing to do is connect sensor:
    - VCC is for your power attachment, connect this to pin 2 or 4 on your pi (Red)
    - Trig and Echo are for measurements so connect them to GPIO-pins
   - GND is grounding connect it to a grounding pin (Black)
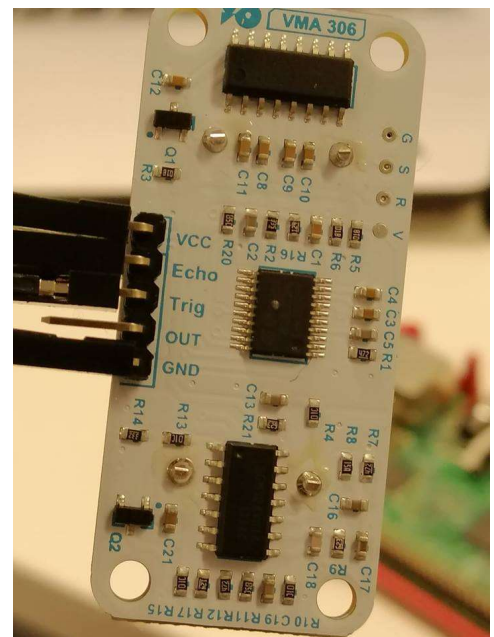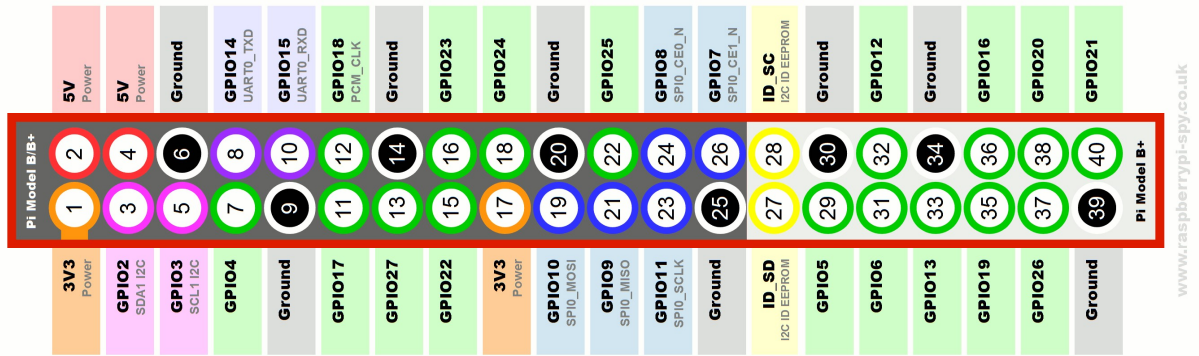
make sure the pins are placed correctly otherwise the script won't work. Modify the script based on what pins you use.

2. next connect the LED's to the breadboard:
    o   you will need resistors when attaching the LED's so put the LED's in series with the resistors.
    o   The sequence should be: GPIO(-in), resistor, LED, GND(-out)

Your hardware is now setup!
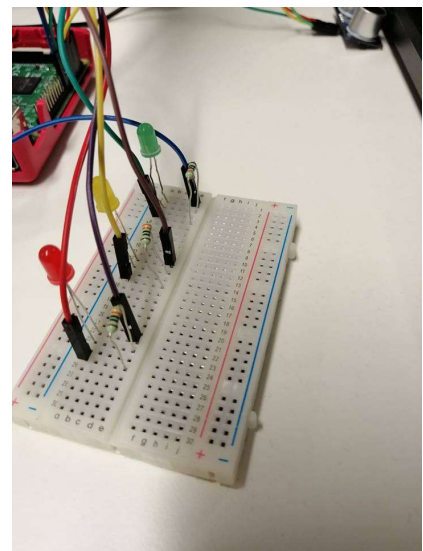
Connect your software

1. Get the script up and running

Connect to your pi and go to github to download the script(tutorialscript.py).

The script wont work yet, modify it so it will use your specific data. Some parts you will not yet be able to fill in before the cloud setup. Therefore, after the cloud setup, go back to your script for these parts.

Even then the script still needs some modules:

Download these modules, one you always need to download is 'ibmiotf'

        Pip3 install ibmiotf

Run your script with:

        Python3 tutorialscript.py

Now your sensors work but you still need the Cloud.

2. Cloud setup

- ○ make an IBM cloud account
- ○ login to the IBM cloud platform

- ○ make an IoT platform

o   make a device-type MQTT-client

o   make a device from the device-type

STORE THE TOKEN SO YOU CAN CONNECT!!!!! You will need to put it in the script.

Now you have a working cloud that receives the data.

3. make an android application and connect to cloud
    o You can choose whatever application you want to make, for connecting to the cloud there is a lot of (al be it somewhat aged) information out there for different types of application.
    o If working with android, the device-type should be 'Android' and not whatever type you made on the cloud.
    o The application should be subscribed to get the information
    o If you want to control LED's from the application, you will need to publish this to the cloud