

Introduction to an IoT Project – Smart Umbrella Stand

What is Internet of Things (IoT)

The Internet is built for doing data communications. I.e. It is for sharing data between parties. Usually, the parties are humans. While for IoT, on the other hand, the communication parties are “Things”, i.e. machine devices.

Hence IoT is -- machines (things) communicating on the Internet.

Why IoT?

Machines exchange data for -- enabling centralized data storage, data analysis and remote control.

Data analysis examples

- Cars can send out the road conditions as they travel to let other cars understand the road conditions.
- Public lamp posts can send over the real-time camera videos on the streets to help Police to fight crime.
- Machines in the factory can send out the operation data so that engineers can understand the working conditions of the machines and perform urgent inspections before the machines are really broken down.

How to make IoT?

1. The machine devices have Wi-Fi abilities.
2. Devices must have data collected from sensors to send over to central server.
3. In order to standardize the data communicate between devices, a data format called json is widely used.

Examples of IoT

- Public lights can be remotely controlled to turn on when it is getting dark.
- Video cameras on streets can pick up real-time videos for central computers to fight crimes.

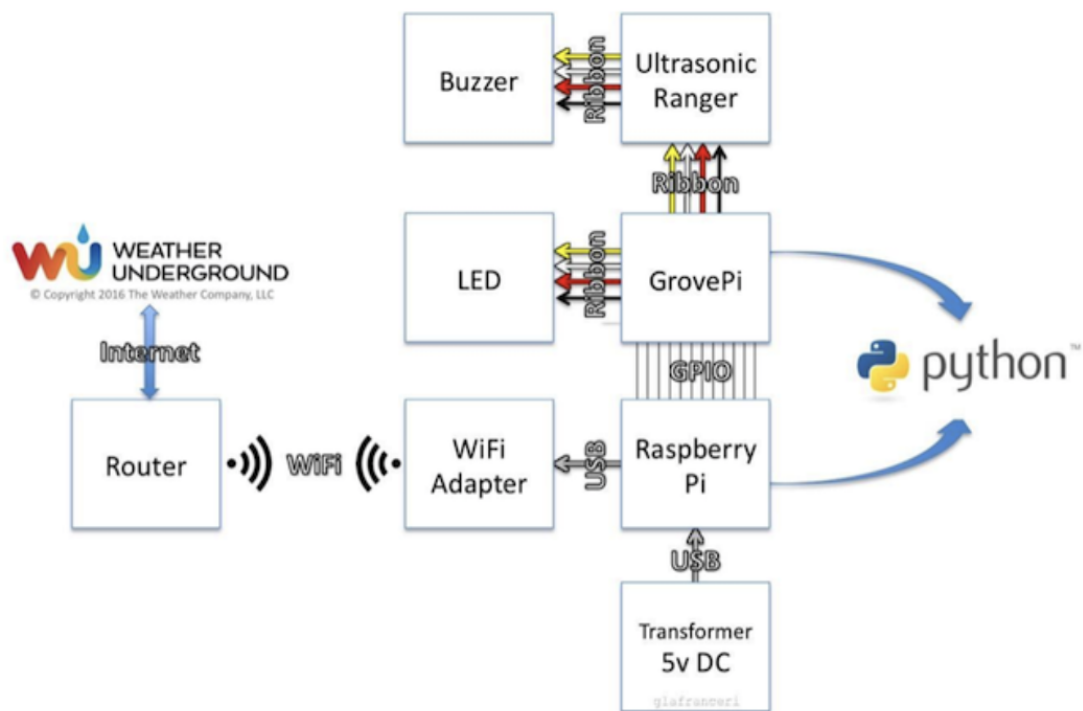
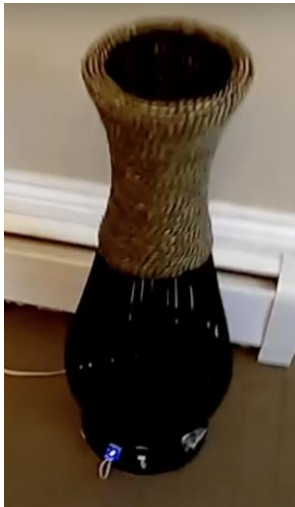
An project of IoT -- Smart Umbrella Stand

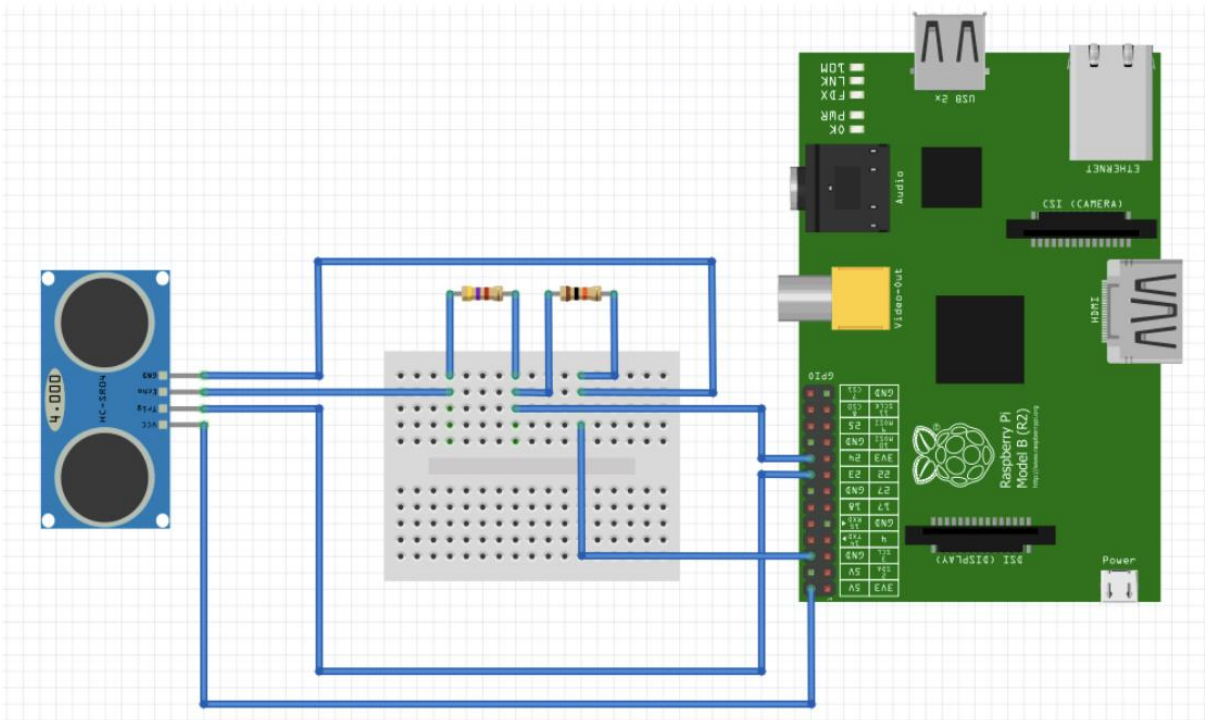
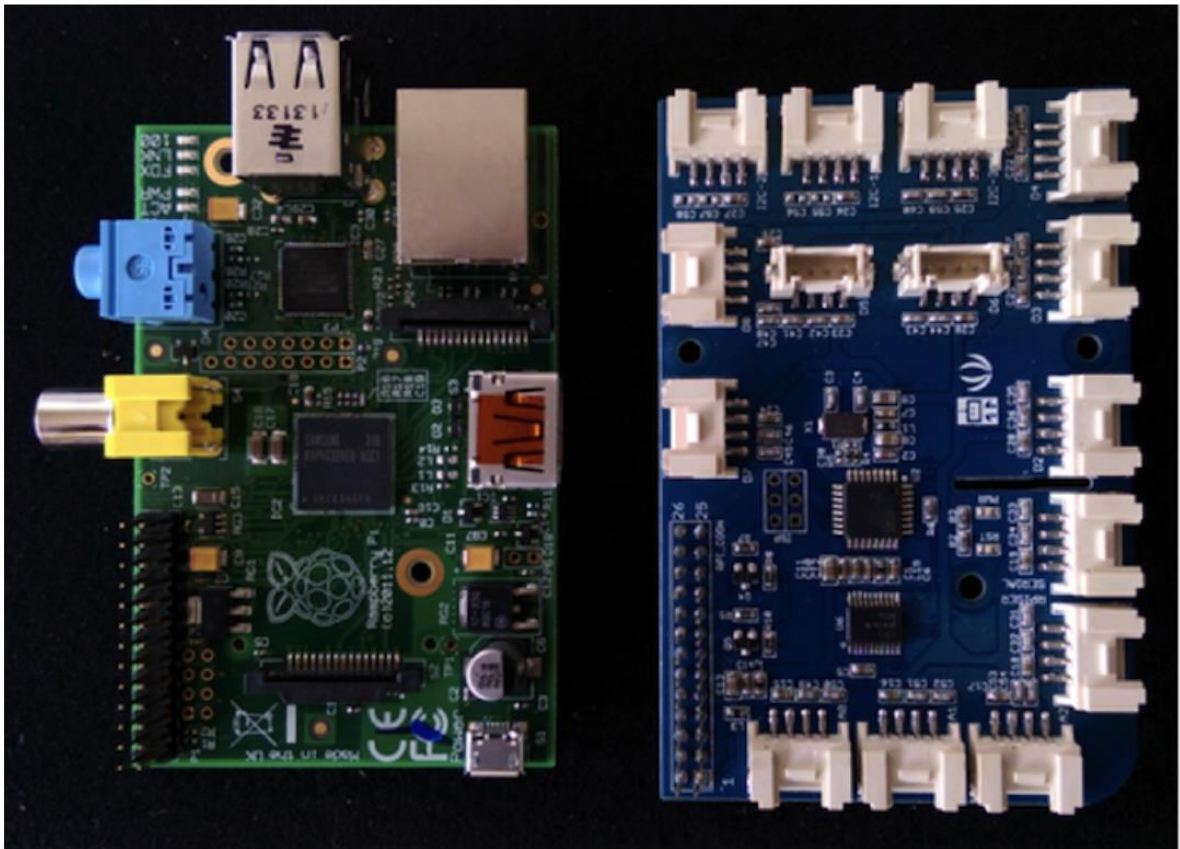
(Modified from the site: <https://www.deviceplus.com/raspberry-pi/looking-for-useful-raspberry-pi-projects-create-a-smart-umbrella-stand/>)

A smart umbrella stand placed by the main door can remind the owner to bring along an umbrella if it is going to rain at the moment when the owner left home.

Project details

In the stand, we use a Raspberry Pi CPU board to constantly read the weather information from an on-line web site to check out the current weather conditions. It also uses an ultrasonic sensor to monitor if anyone is nearby. Only when someone getting close and the weather is going to rain, it will alert by emitting sounds to remind the person to bring out the umbrella.



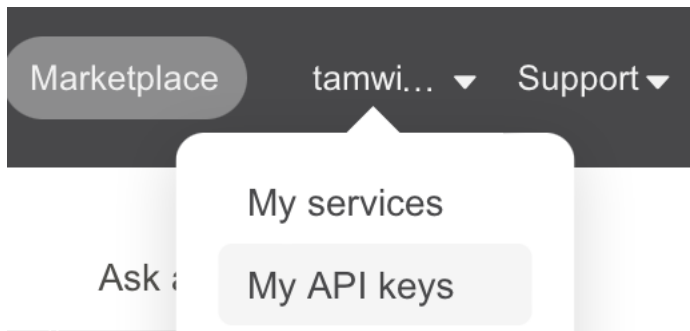


(Note: some pins may change)

Connection with the online weather server

Visit the site: <https://openweathermap.org/current>

Create an account. Login and press on your login name on the top-right hand corner. Then choose “My API keys”



To receive the json data of Hong Kong Weather, If the API Key is “f5464956d25632a90ffd7711fe1f49ab”, use a browser to visit:
<https://api.openweathermap.org/data/2.5/onecall?lat=22.285521&lon=114.157692&appid=f5464956d25632a90ffd7711fe1f49af>

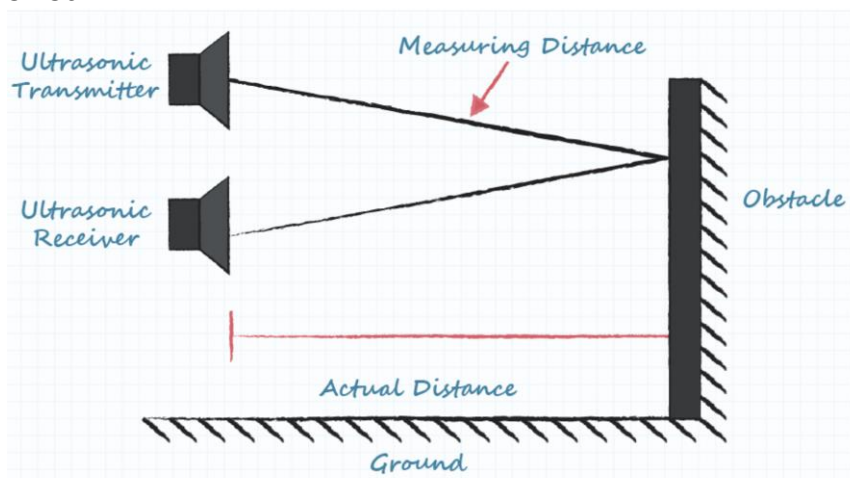
It will show something similar as the following:

```
{ "lat": 22.2855, "lon": 114.1577, "timezone": "Asia/Hong_Kong", "timezone_offset": 28800, "current": { "dt": 1650207372, "sunrise": 1650146520, "sunset": 1650192219, "temp": 295.09, "feels_like": 295.32, "clouds": 97, "visibility": 10000, "wind_speed": 4.92, "wind_deg": 80, "weather": [ { "id": 804, "main": "Cl
```

Then we can write a program to pick up the relevant data inside to get the current weather information and alarm the user when necessary.

Using ultrasonic ranger to check the proximity distance of the user

Ultrasonic ranger has two parts: ultrasonic transmitter and ultrasonic receiver, where the transmitter will emit ultrasound waves which will travel in air with speed 343 m/s and when run across an obstacle, it will bounce back and be detected by the receiver. Upon calculation of the time between transmission and receiving, it can find out the distance of objects in front of the ultrasonic ranger. For example, if the time between transmission and receiving ultrasonic signals is 1/1000s, then the distance should be $343/1000 \text{ m} = 0.343 \text{ m}$ or 34.3cm.



Basic operation of an ultrasonic transmitter and receiver pair

(Reference)

The Python program (run it on the Raspberry Pi)

Connections details:

It must connect LED (Pin7), ultrasonic sensor (Trigger input Pin18 and Echo output Pin24), and buzzer (Pin8), to the Raspberry Pi

```
# umbrella_stand.py
import urllib
import requests
import json
#from grovepi import *
import time
import sys

# Connect the LED to D7 pin
rainled = 7

# Connect the Ultrasonic Ranger to D4 pin
ranger = 4

# Connect the Buzzer to D8 pin
buzzer = 8

digitalWrite(rainled, 0) #turn off LED
digitalWrite(buzzer, 0). #turn off buzzer

def assign_rain(status):
    if status is True: # there will be rain
        # Light the RAIN LED on the umbrella stand
        led_blink(DELAY)
    else: # no rain in forecast
        # turn off the RAIN LED on the umbrella
        digitalWrite(rainled, 0)
        time.sleep(DELAY)

def led_blink(blinkdelay):
    count = 0.0
    blinkrate = 0.4
    while count < float(blinkdelay):
        digitalWrite(rainled, 1)
        time.sleep(blinkrate)
        distance=ultrasonicRead(ranger)
        if distance <= float(PROXIMITY):
            digitalWrite(buzzer, 1)
        else:
            digitalWrite(buzzer, 0)
        digitalWrite(rainled, 0)
        time.sleep(blinkrate)
        count += (2*blinkrate)
```

```

#####
#####

# openweathermap api url
url =
'https://api.openweathermap.org/data/2.5/onecall?lat=22.285521&lon=114.157692&appid=f5
464956d25632a90ffd7711fe1f49af'

pinMode(ranger,"INPUT")
pinMode(buzzer,"OUTPUT")
pinMode(rainled, "OUTPUT")

if len(sys.argv) > 1 and sys.argv[1] == 'test':
    try:
        led_blink(5)
    except KeyboardInterrupt:
        digitalWrite(rainled, 0)
        quit()

while True:
    f = urllib.request.urlopen(url)
    json_string = f.read()
    f.close()
    parsed_json = json.loads(json_string)

    if METRIC == 0:
        precip_today = parsed_json['current']['sunset']
    else:
        precip_today = parsed_json['current_observation']['precip_today_metric']

    print(precip_today)

    if float(precip_today) > float(RAIN_THRESHOLD):
        print("Rain today, take the umbrella")
        assign_rain(True)
    else:
        print("No Rain today")
        assign_rain(False)

```