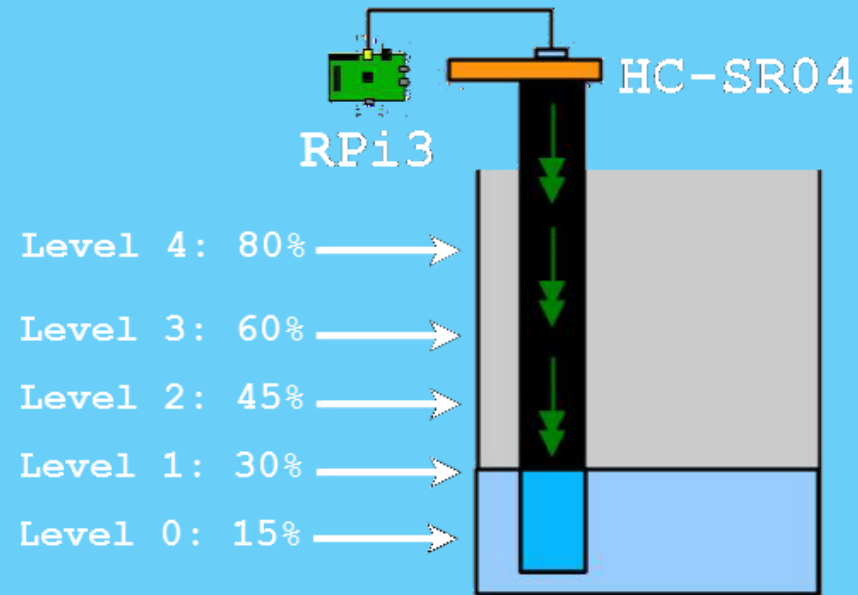


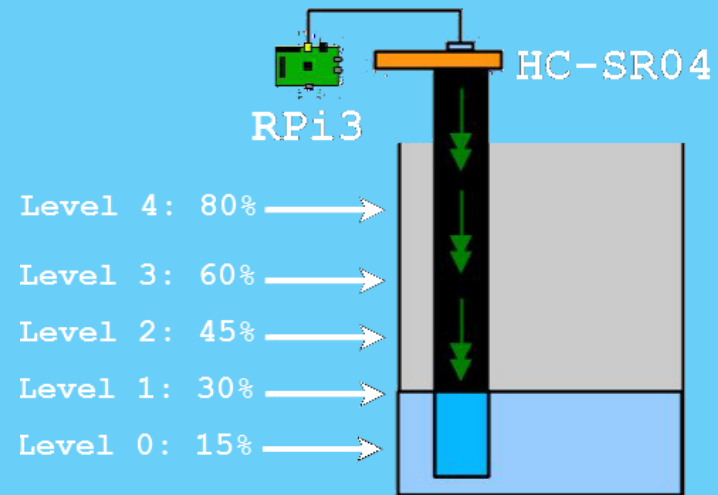
Water Level Monitoring



- The Water Level Monitoring system is a simple solution for continuous monitoring of water or other fluid levels.



Water Level Monitoring



- Monitoring based on 5 levels
 - Level 0: Storage on clod-based db
 - Level 1: Storage on clod-based db
 - Level 2: Storage on clod-based db
 - Level 3: Storage on clod-based db
 - Level 4: Storage on clod-based db + Email



Application's Purpose

- The aim of this project is to create a water level monitoring system, which works via an internet connection and wirelessly:
 - Save the measured values in a cloud database
 - Access this data via a cloud-based web application
 - View this data in an Android/iOS mobile application
 - Notify the user via email of any problems during the monitoring



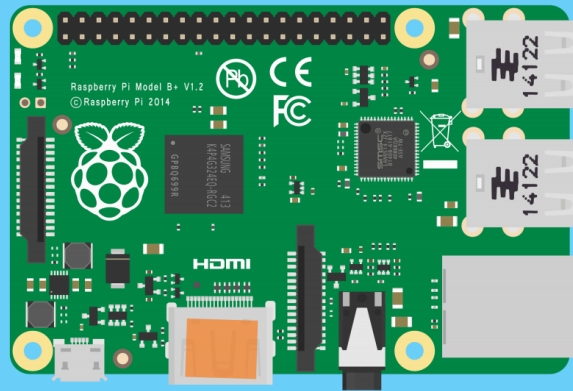
Hardware components

- Raspberry Pi 3 Model B+
- HC-SR04 Ultrasonic Module Distance Sensor
- Breadboard 400 Points Solderless
- Jumper Wires



Raspberry Pi 3 Model B+

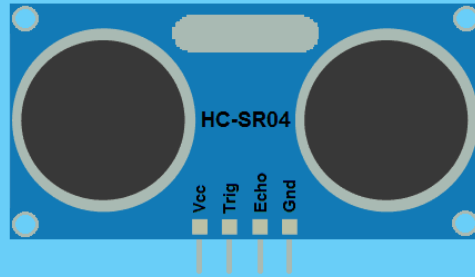
- **Raspberry Pi** is an **ARM** based credit card sized **SBC**(Single Board Computer). Raspberry Pi runs Debian based **GNU/Linux** operating system [Raspbian](#) and ports of many other OSes exist for this SBC.



- A powerful feature of the Raspberry Pi is the row of GPIO (general-purpose input/output) pins along the top edge of the board.
 - Any of the GPIO pins can be designated (in software) as an input or output pin and used for a wide range of purposes.



HC-SR04 Ultrasonic Module Distance Sensor



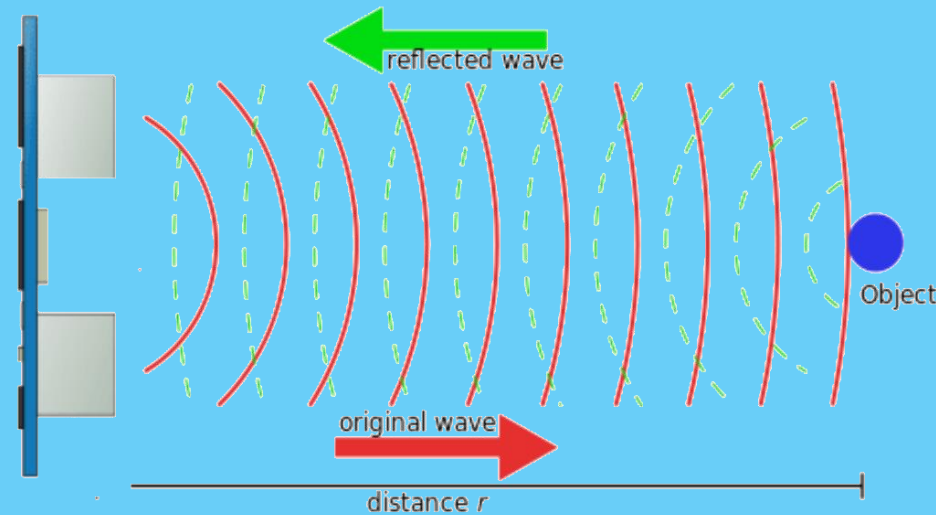
Operating voltage	+5V
Ultrasonic frequency	40 kHz
Min. distance	2cm
Max. distance	400cm
Accuracy	3mm
Measuring angle covered	15°

- The HC-SR04 is an ultrasonic sensor that uses sonar to determine distance to an object
- It includes 4 pins
 - VCC: for powers the sensor
 - Trig: is an Input pin
 - Echo: is an Output pin
 - Gnd: This pin is connected to the Ground of the system.

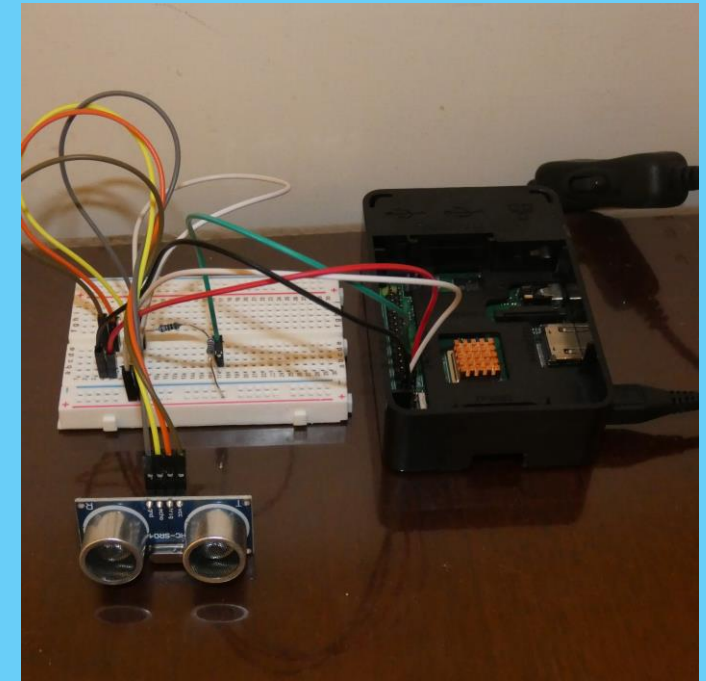
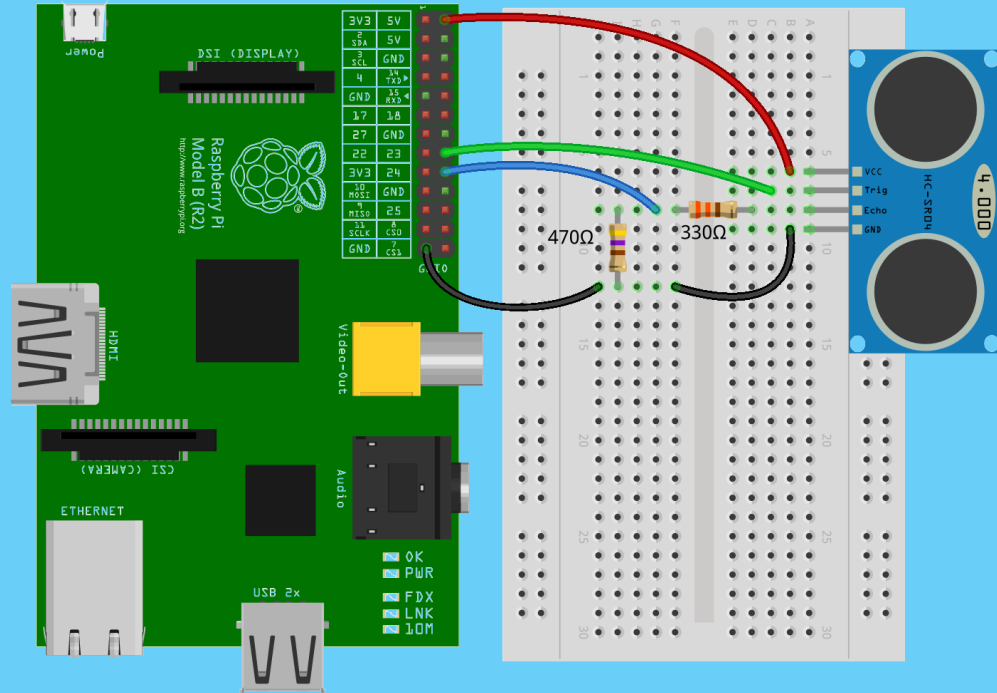


HC-SR04 - How it works

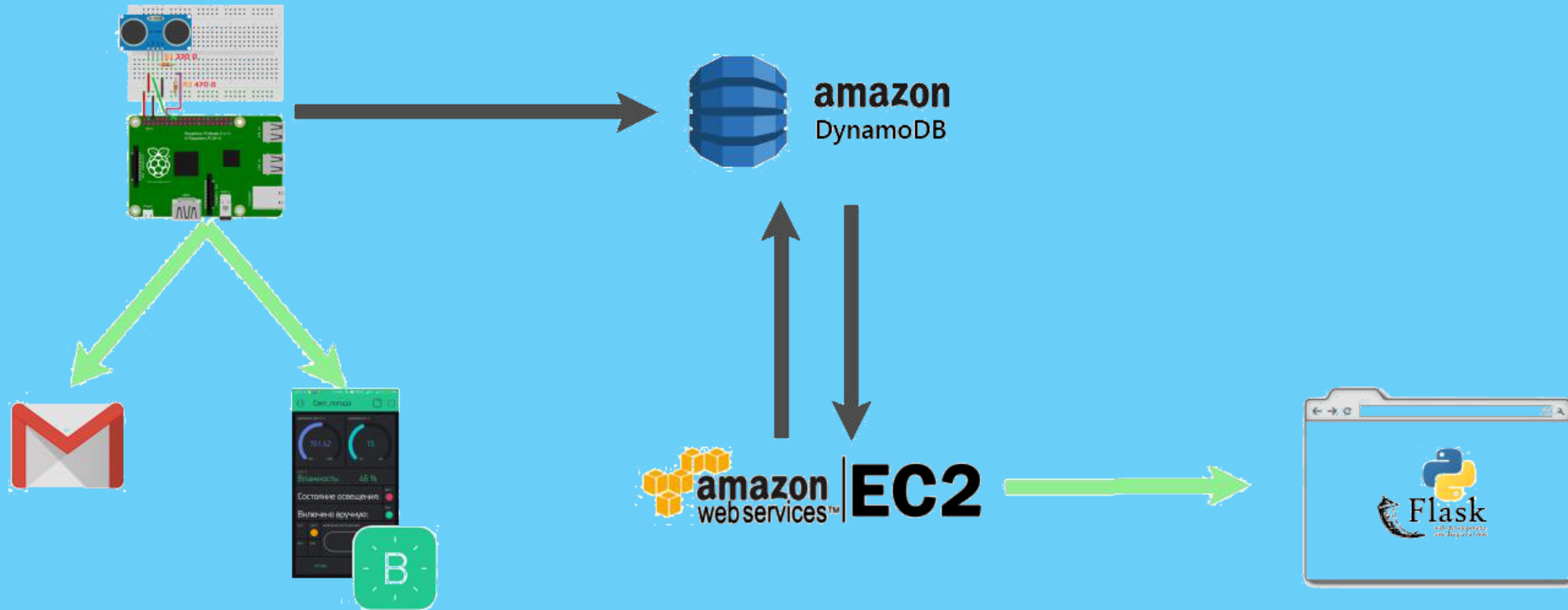
- Rough speed of ultrasonic sound = 34300 cm/s
- $\text{PulseDuration} = \text{StopTime} - \text{StartTime}$
- $\text{Distance in cm} = (\text{PulseDuration} * \text{Rough speed of ultrasonic sound}) / 2$



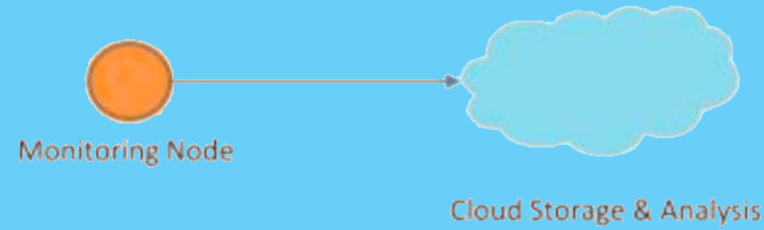
Component Connection



Software Architecture



IoT Level



- The developed project is a level-3 IoT system:
 - has a single node
 - data is stored and analyzed in the cloud
 - application is cloud-based.



RPi3 Development Environment



- Thonny is an integrated development environment(IDE) for Python 3.7
- In RPi3 IDE proceeded to:
 - Interact with HC-SR04(Startup, Start/Stop sampling)
 - Get the distance from water surface
 - Save data on cloud-based database
 - Send email(if necessary)
 - Send data to mobile app



DynamoDB



- DynamoDB is a hosted NoSQL database offered by Amazon Web Services (AWS)
 - It is advantageous and cheaper when many search queries are needed (and not join queries)



DynamoDB

- In dynamoDB are stored: data, distance, level, percentage

<input type="checkbox"/>	date ⓘ ▲	distance ▼	level ▼	percentage ▼
<input type="checkbox"/>	11-07-2019 23:22:05	15.42	0	7
<input type="checkbox"/>	11-07-2019 23:22:18	13.53	1	18
<input type="checkbox"/>	11-07-2019 23:22:36	11.37	2	31
<input type="checkbox"/>	11-07-2019 23:23:05	7.84	3	52
<input type="checkbox"/>	11-07-2019 23:23:42	3.29	4	80



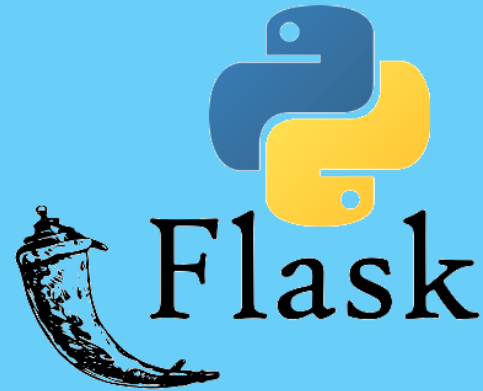
Amazon EC2



- Amazon Elastic Compute Cloud, EC2 is a web service from Amazon that provides re-sizable compute services in the cloud.
- It is based on instance concept: that is a virtual server for running applications
- Pricing options:
 - Spot Instances
 - On Demand Instances
 - Reserved Instances



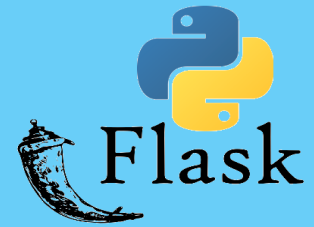
Flask framework



- Flask is a micro web framework written in Python.
 - It is classified as a microframework because it does not require particular tools or libraries
 - There are many extensions provided by the community that make adding new functionality easy.



Amazon EC2 - Flask



- To view data stored in the database a web app has been implemented
 - Flask is the development framework
 - An instance on Amazon EC2 is the development environment, and it is used to launch the application



Web Application: Data

```
[ec2-user@ip-172-31-95-24 distance_sensor_app]$ sudopython app.py
-bash: sudopython: command not found
[ec2-user@ip-172-31-95-24 distance_sensor_app]$ sudo python app.py
* Serving Flask app "app" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```

Amazon EC2 Instance

Non sicuro — ec2-3-95-153-102.compute-1.amazonaws.com

Water Level Monitoring: Data

Data

Percentage Chart

Level Chart

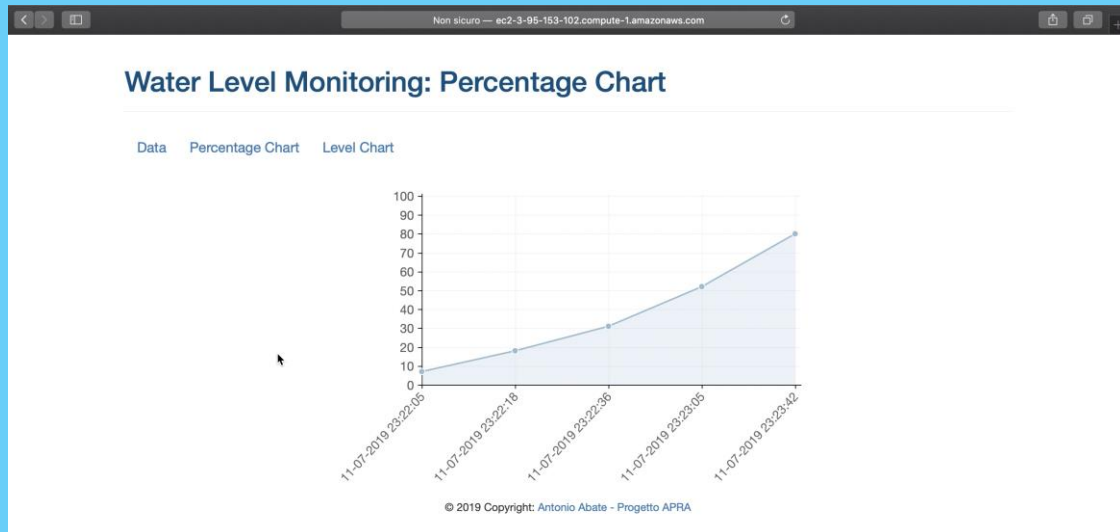
DATE	DISTANCE	PERCENTAGE	LEVEL
11-07-2019 23:23:42	3.29 cm	80%	4
11-07-2019 23:23:05	7.84 cm	52%	3
11-07-2019 23:22:36	11.37 cm	31%	2
11-07-2019 23:22:18	13.53 cm	18%	1
11-07-2019 23:22:05	15.42 cm	7%	0

© 2019 Copyright: Antonio Abate - Progetto APRA

Web Browser



Web Application: Charts



Percentage Chart



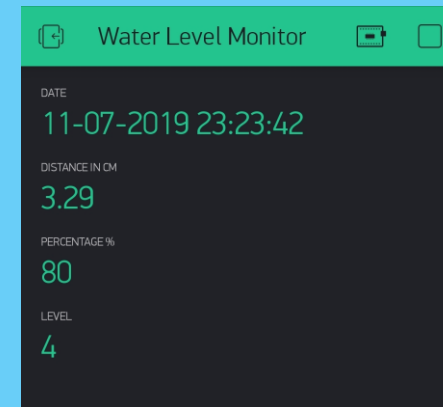
Level Chart



Blynk App

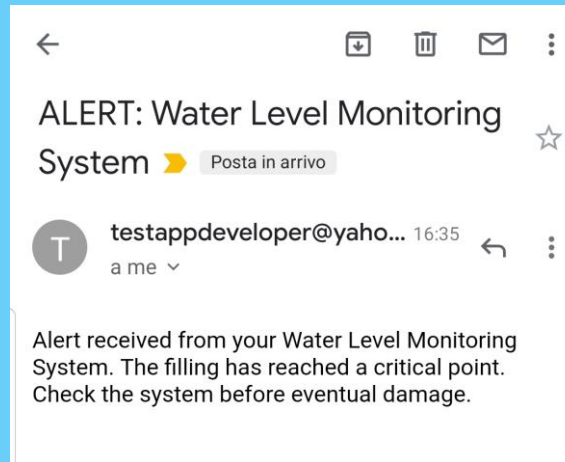


- Blynk is a popular IoT platform to connect to the cloud, design apps to control them and analyze telemetry data.
- It allows to:
 - Connect your devices to open and secure cloud
 - Drag-n-drop beautiful IoT apps in minutes
 - Manage connection in real-time
- It is available on Android/iOS device



Email

- An email is sent to the user if the water level exceeds 80% of the total.
 - This is a very useful service to alert the user in case of critical issues.
 - It allows to prevent any damage that may be caused to the system.
 - Allows not to continuously monitor the system, being alerted in case of danger



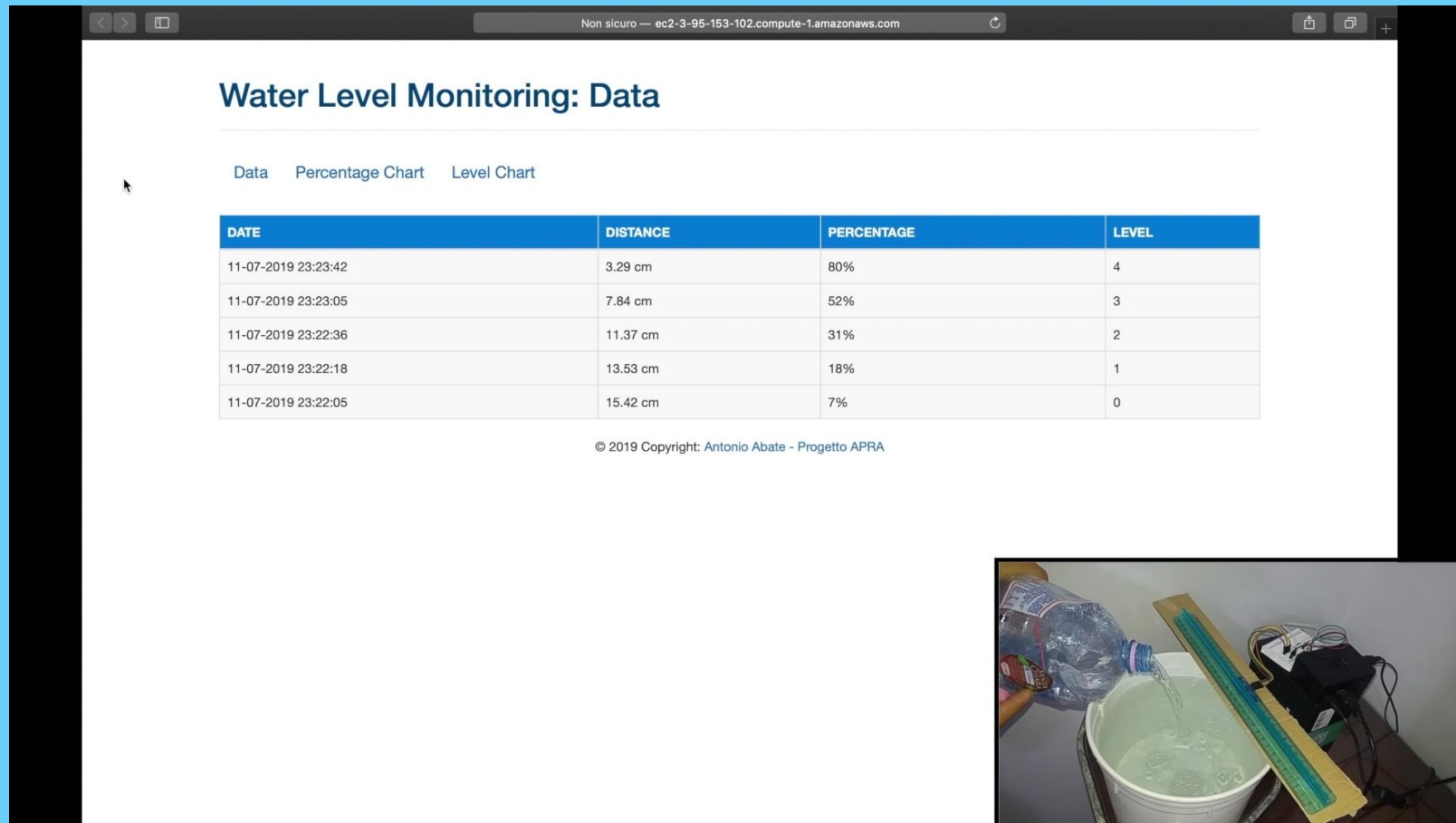
Dataplicity



- Recommended by the official documentation of Raspberry Pi [<http://www.raspberrypi.org/documentation>]
- Dataplicity is a remote terminal for Raspberry Pi.
 - It offers functionality similar to SSH, but it doesn't need any complex set-up to get it working
 - It allows remote control of RPi via the internet

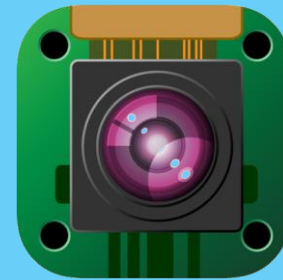
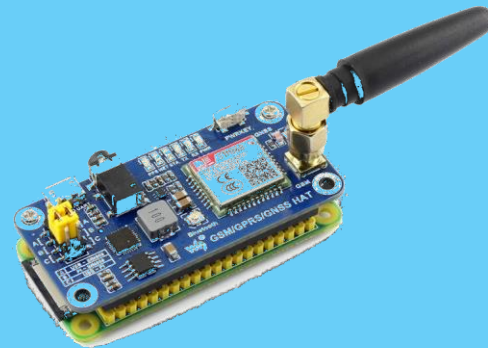
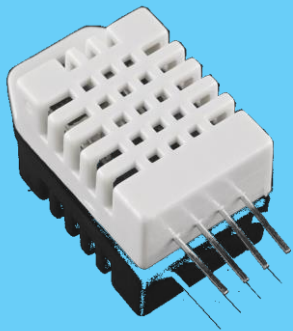


Test Case



Future developments

- Add temperature sensor for reducing distance error
- Add GPRS module to increase system independence
- Add Pi Camera Module for environment monitoring



THANKS FOR YOU ATTENTION

