USTH Smart Garden

Member

HOANG Duc Minh

DO Duy Huy Hoang

University of Science and Technology Hanoi

ICT Department

2019-05-06

Contents

I. Introduction

Gardening is very interesting, so We thought about making a project in gardening. Nowadays smart garden systems are very popular because people enjoy them. However our challenge is to make the project very simple so that anyone can implement this. As it turns out, projects based around the Arduino open hardware development board are an excellent place to start. A soil moisture sensor can read the amount of moisture present in the soil surrounding it. It's an ideal for monitoring an urban garden, or your pet plant's water level. This is a must have component for a IOT garden / Agriculture!

II. Sensors

We are now having two different type of sensor:

- Capacitive Soil Moisture Sensor - SKU_SEN0193
- Analog Soil Moisture Sensor - SKU_SEN0114

A. Capacitive Soil Moisture Sensor - SKU_SEN0193

Figure 1: Capacitive Soil Moisture Sensor

This product measures soil moisture levels by capacitive sensing, rather than resistive sensing like other types of moisture sensor. It is made of a corrosion resistant material giving it a long service life. Insert it into soil and impress your friends with the real-time soil moisture data! The product includes an on-board voltage regulator which gives it an operating voltage range of 3.3 ~ 5.5V. It is compatible with low-voltage MCUs (both 3.3V and 5V logic). To make it compatible with a Raspberry Pi, an ADC converter is required.

For example, we can use an Arduino to read the value directly from the sensor.

FEATURES

- Supports Gravity 3-Pin Interface
- Analog Output

APPLICATIONS

- Gardening & Farming
- Moisture Detection
- Intelligent Agriculture

Specification

- Operating Voltage: 3.3 ~ 5.5 VDC
- Output Voltage: 0 ~ 3.0VDC
- Operating Current: 5mA
- Interface: PH2.0-3P
- Dimensions: 3.86 x 0.905 inches (L x W)
- Weight: 15g

Schematic

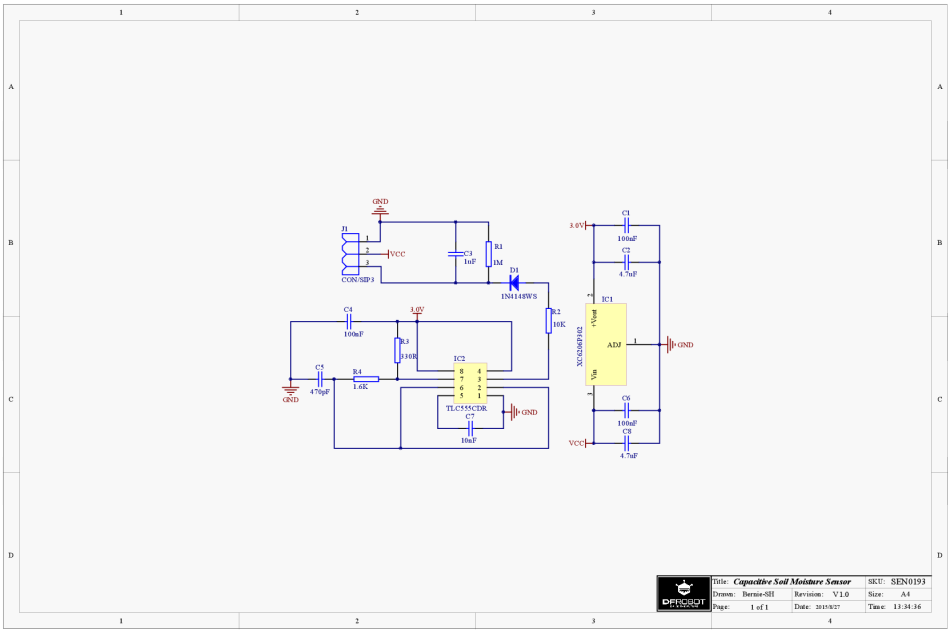

Figure 2: Schematic

Connection

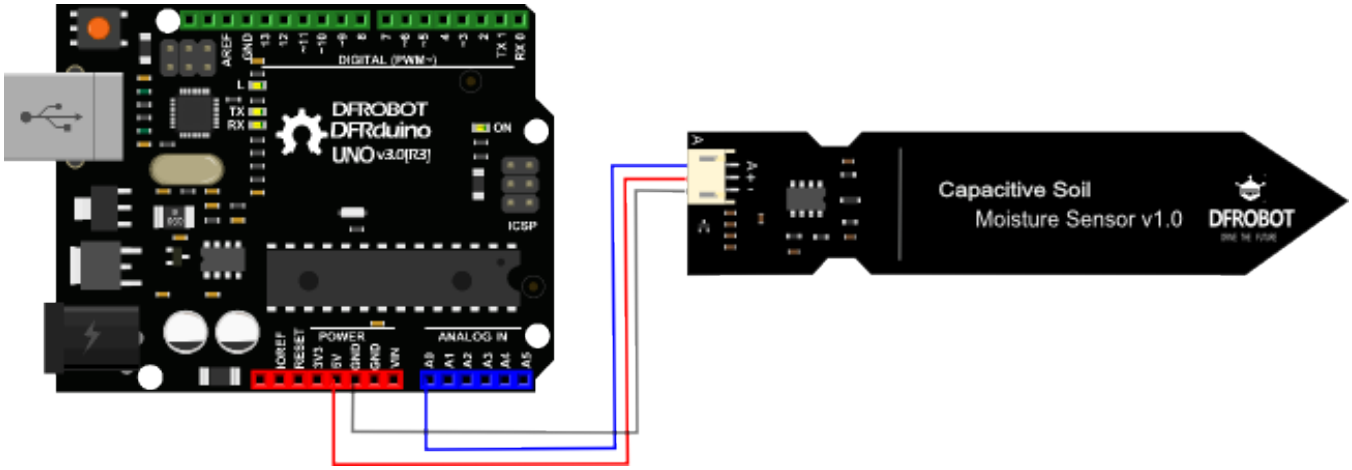

Figure 3: Connection Example with Arduino Uno

Aduino Uno	Capacitive Soil Moisture Sensor
GND	GND (Black wire)
5V	Red Wire
Analog output	Blue Wire

Calibration Range

The components on this board are NOT waterproof, do not expose to moisture further than the red line. (If you want to protect components from the elements, try using a length of wide heat shrink tubing around the upper-section of the board.) There is an inverse ratio between the sensor output value and soil moisture.

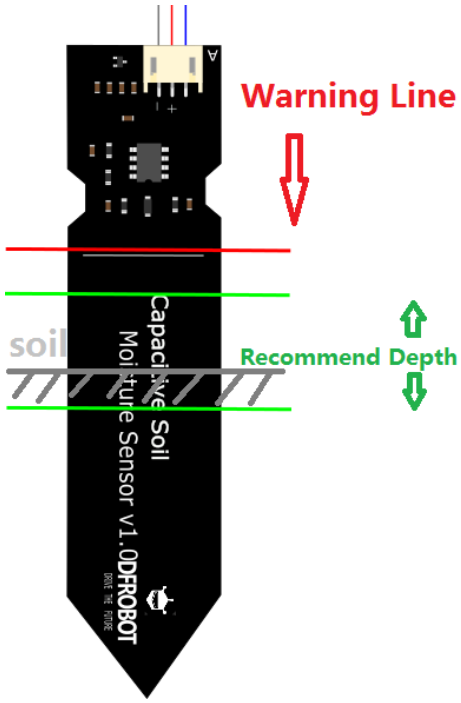

Figure 4: Calibration Range

- The final output value is affected by probe insertion depth and how tight the soil packed around it is.
- The range will be divided into three sections: dry, wet, water. Their related values are:
  - Dry: (520 430]
  - Wet: (430 350]
  - Water: (350 260]

Sample Code

Our Sample Code can be found **here**

B. Analog Soil Moisture Sensor - SKU_SEN0114

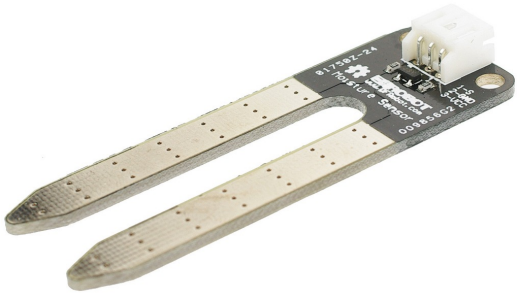

Figure 5: Analog Soil Moisture Sensor

This Gravity: **Analog Soil Moisture Sensor** For Arduino can read the amount of moisture present in the soil surrounding it. It's a low tech sensor, but ideal for monitoring an urban garden, or your pet plant's water level. This is a must have tool for a connected garden! The new soil moisture sensor uses Immersion Gold which protects the nickel from oxidation. Electroless nickel immersion gold (ENIG) has several advantages over more conventional (and cheaper) surface platings such as HASL (solder), including excellent surface planarity (particularly helpful for PCB's with large BGA packages), good oxidation resistance, and usability for untreated contact surfaces such as membrane switches and contact points.

This soil moisture arduino sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance). This sensor will be helpful to remind you to water your indoor plants or to monitor the soil moisture in your garden.

Note: If you are going to place the sensor to very humid soil, that would make it very easy to get the sensor surface erosion within several days, hence we suggest to use this one Analog Capacitive Soil Moisture Sensor- Corrosion Resistant instead.

Specification

- Power supply: 3.3V ~ 5V
- Output voltage signal: 0 ~ 4.2V
- Current: 35mA
- Pin definition:
  - Blue: Analog output
  - Black: GND
  - Red: Power
- Size: 60x20x5mm
- value range:
  - 0~300: dry soil
  - 300~700: humid soil
  - 700~950: in water

Schematic

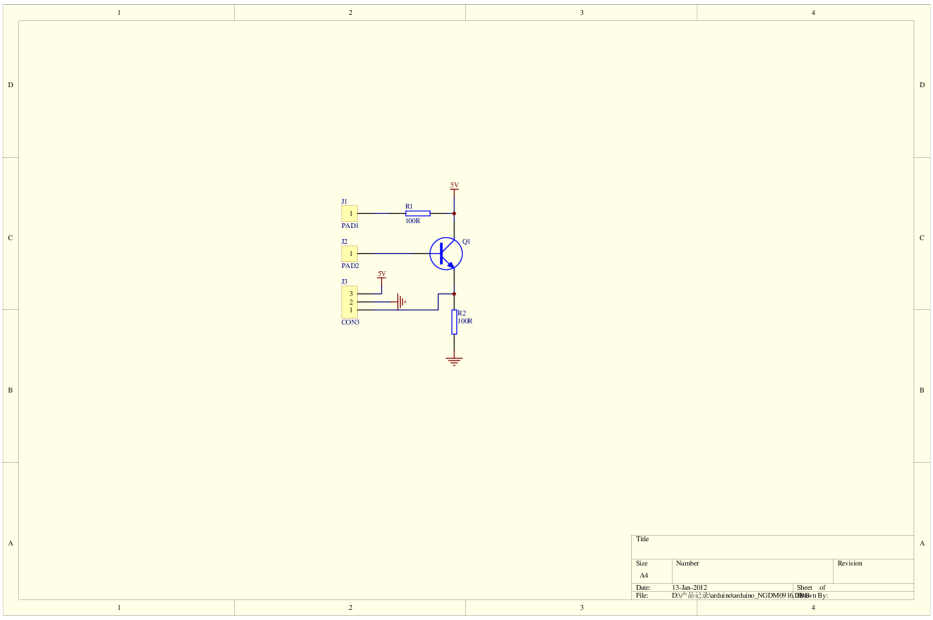

Figure 6: Schematic

Connection

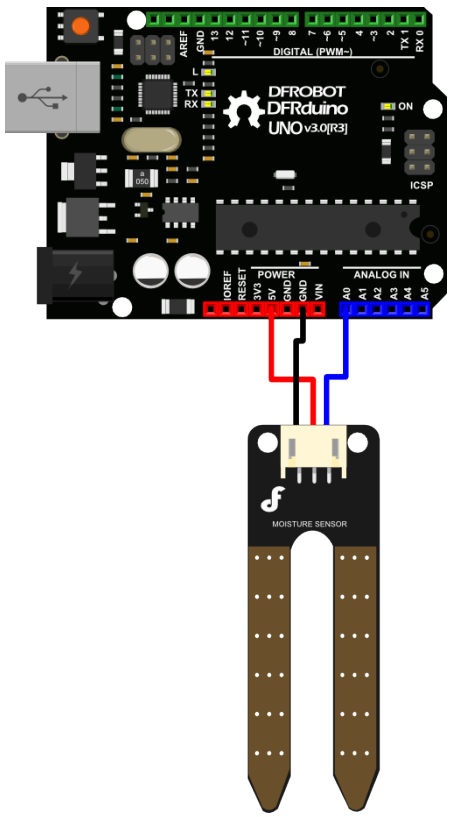

Figure 7: Connection Example with Arduino Uno

Aduino Uno	Analog Soil Moisture Sensor
GND	GND (Black wire)
5V	Red Wire
Analog output	Blue Wire

Calibration Range

- The range will be divided into three sections: dry soil, wet soil, in water. Their related values are:
  - 0 ~ 300 - dry soil
  - 300~700 - humid soil
  - 700~950 - in water

Sample Code

Our Sample Code can be found **here**

III. LoRa

Introduction

The term LoRa stands for Long Range. It is a wireless Radio frequency technology introduced by a company called Semtech. This LoRa technology can be used to transmit bi-directional information to long distance without consuming much power. This property can be used by remote sensors which have to transmit its data by just operating on a small battery. Typically Lora can achieve a distance of 15-20km and can work on battery for years.

Connecting Raspberry Pi with LoRa

Lora module communicates using SPI on 3.3V Logic. The Raspberry pi also operates in 3.3V logic level and also has in-built SPI port and 3.3V regulator. So we can directly connect the LoRa module with the Raspberry Pi. The connection table is shown below

Raspberry Pi	Lora Module
3.3V	3.3V
Ground	Ground
GPIO 10	MOSI
GPIO 9	MISO
GPIO 11	SCK
GPIO 8	Nss / Enable
GPIO 4	DIO 0
GPIO 17	DIO 1
GPIO 18	DIO 2
GPIO 27	DIO 3
GPIO 22	RST

You can also use the circuit diagram below for reference.

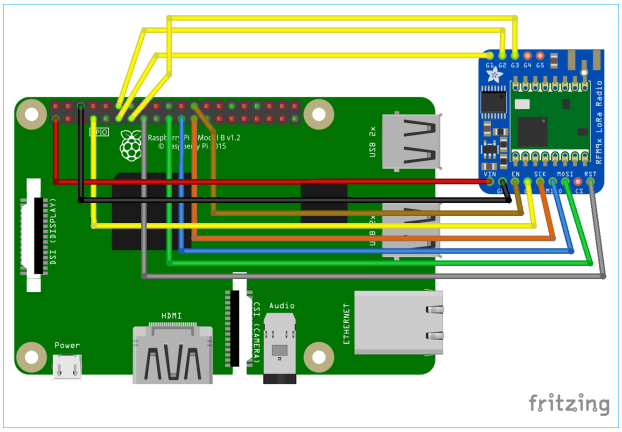

Figure 8: Connecting Raspberry Pi with LoRa

Connecting Arduino with LoRa

Transmitting Side- Connecting LoRa with Arduino UNO For the transmitting side we will use an Arduino UNO with our LoRa module. The circuit diagram to connect the Arduino UNO with LoRa is shown below. The module operates in 3.3V and hence the 3.3V pin on LoRa is connected to the 3.3v pin on the Arduino UNO board.

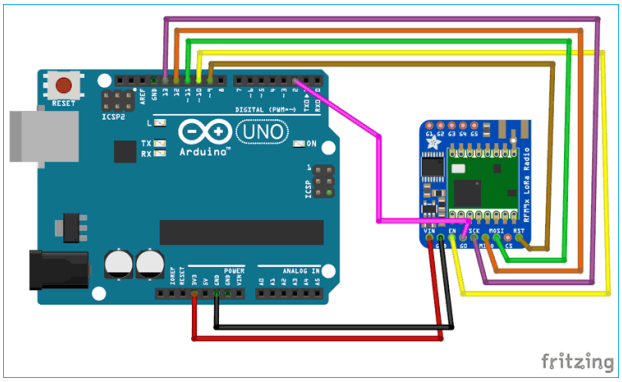

Figure 9: Connecting Arduino with LoRa

LoRa SX1278 Module	Arduino UNO Board
3.3V	3.3V
Gnd	Gnd
En/Nss	D10
G0/DIO0	D2
SCK	D13
MISO	D12
MOSI	D11
RST	D9

IV. Database

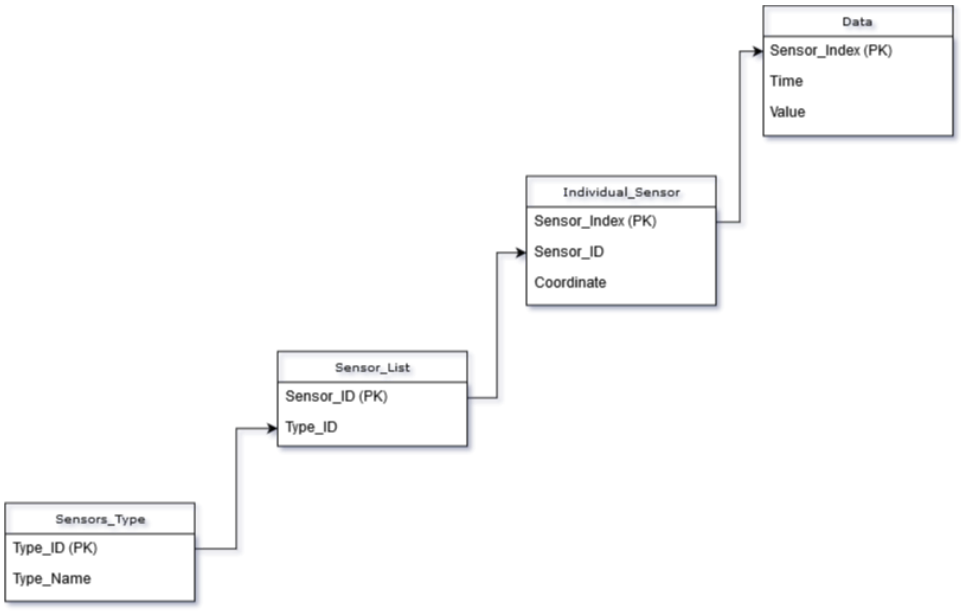

Figure 10: Database

Note: This database architecture was built up on our current knowledge of the system. It may change in the future due to practical usage

In the Database, It contains 4 table which is Sensors_Type, Sensor_List, Individual_Sensor, and Data table.

We created Sensor_Type table to classify different sensor like (Moisture, Heat, accelerator,..) with their own ID.
The table Sensor_List contains all the Sensor ID, e.g: SKU_SEN0114 - stand for Analog Soil Moisture Sensor.
The table Individual_Sensor contains all the sensors and each sensor have its own index value and also their location.
The table Data contains value and the time that we read that value from sensor.

V. HTTP Method

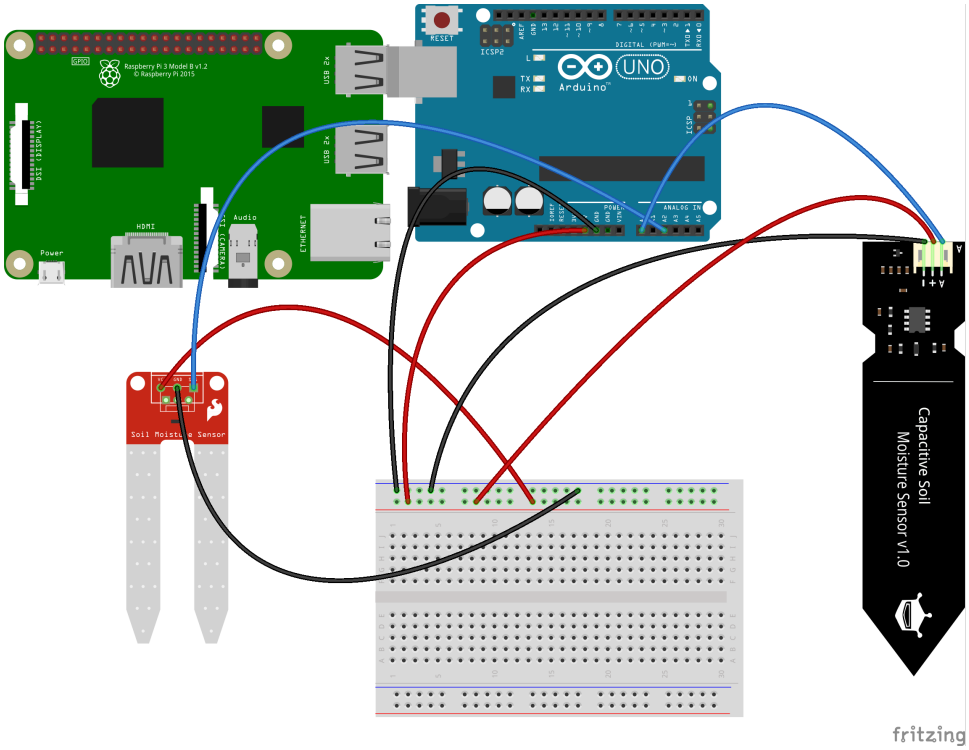

Figure 11: Demo Connection Using Raspberry Pi and Arduino Uno

Reads the value from the specified analog pin. Arduino boards contain a multichannel, 10-bit analog to digital converter. This means that it will map input voltages between 0 and the operating voltage(5V or 3.3V) into integer values between 0 and 1023. On an Arduino UNO, for example, this yields a resolution between readings of: 5 volts / 1024 units or, 0.0049 volts (4.9 mV) per unit. See the table below for the usable pins, operating voltage and maximum resolution for some Arduino boards. **1**

```
1 analogRead(pin)
```

NOTE: On ATmega based boards (UNO, Nano, Mini, Mega), it takes about 100 microseconds (0.0001 s) to read an analog input, so the maximum reading rate is about 10,000 times a second.
We use an simple library called pySerial and Raspberry Pi for reading data from Arduino.

```
# Connection
import serial
ser = serial.Serial(
    port = '/dev/ttyUSB0',
    baudrate = 9600
)
# Get data from Arduino
data = ser.readline()
```

A. Server

On the server side we use a small cronjob to automatically run the php script to receive data from client and update to the database.
We would have a website to display the graph of the data.

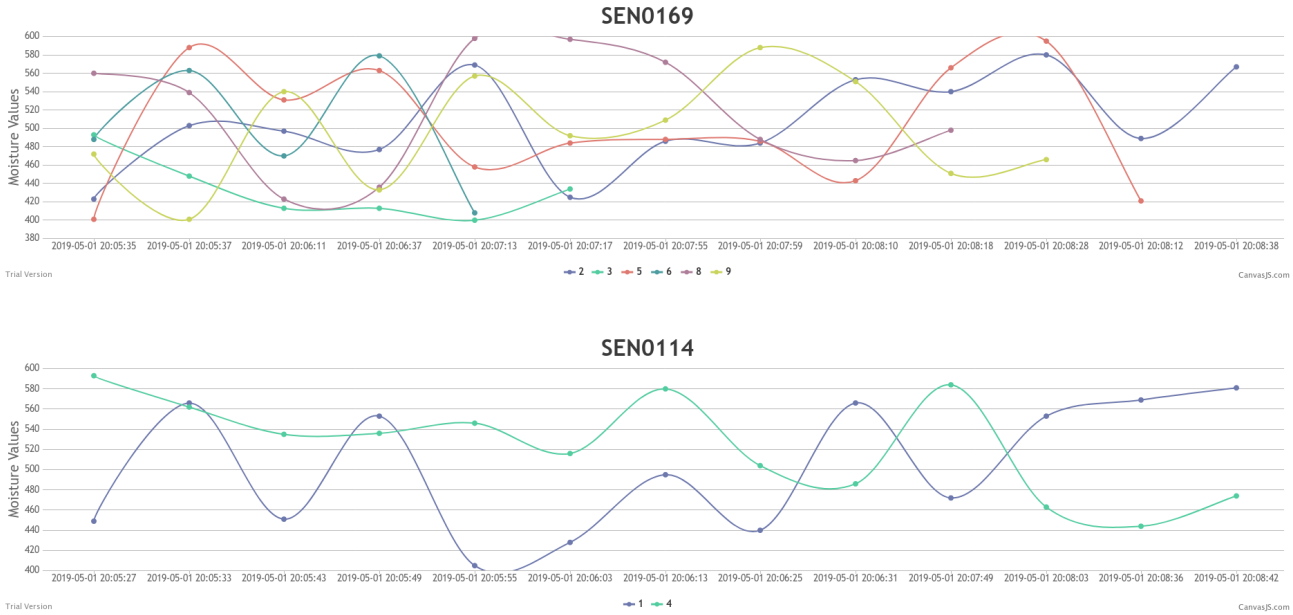

Figure 12: Data Visualization

B. Client

We have a small python script and It is controlled by a cron job. Basically It will read the directly output from the arduino and send back to the server by using HTTP GET request.

VI. Communication Protocol

A. Lora

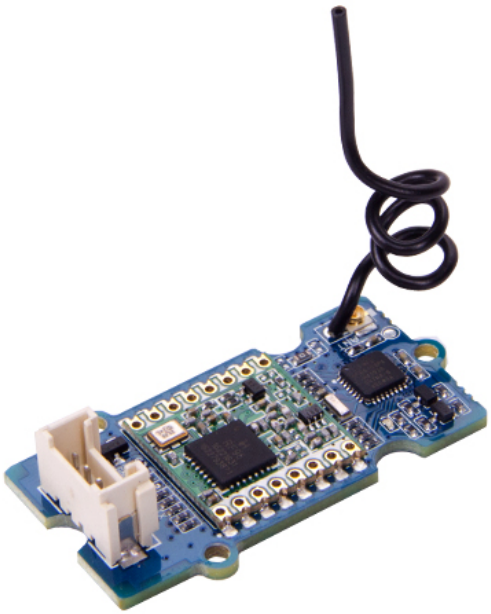

Figure 13: Grove LoRa Radio

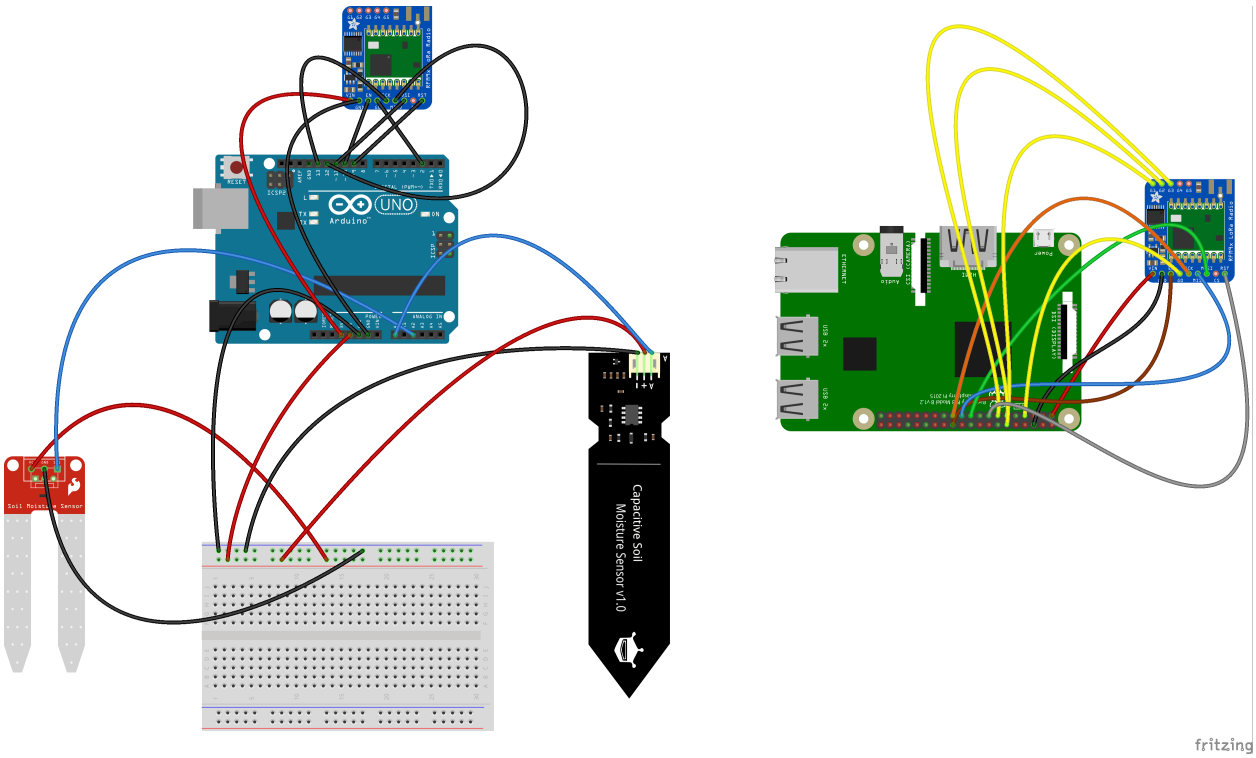

Figure 14: Demo Connection Using Raspberry Pi and Arduino Uno with LoRa

We are using a Lora module which is provided from Seeed Studio company. The main functional module in Grove - LoRa Radio 433MHz is RFM98, which is a transceiver features the LoRa long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst mini-missing current consumption. The heart of Grove - LoRa Radio 433MHz is ATmega168, a widely used chip with very high-performance and low power consumption, especially suitable for this grove module. This is the 868MHz version, which can be used for 868MHz communication.

Features

- Using RFM95 module based on SX1276 LoRa®
- Working Voltage:5V/3.3V
- ~28mA(Avg) @+20dBm continuous transmit
- ~8.4mA(Avg)@standby mode
- ~20mA(Avg) @receive mode, BW-500kHz
- Working Temperature:-20 – 70°C
- Interface:Grove - UART(RX,TX,VCC,GND)
- Simple wire antenna or MHF Connector for external high gain antenna
- Working Frequency:868MHz/433MHz
- +20dBm 100 mW Power Output Capability

- Size:20*40mm
- Rate:0.3kps~50kps
- Ready-to-go Arduino libraries
- Resered MHF antenna connector

From the official website, The Platforms Supported is only Arduino. To work with Raspberry Pi, we have to use an third-party library from **here**

B. Zigbee

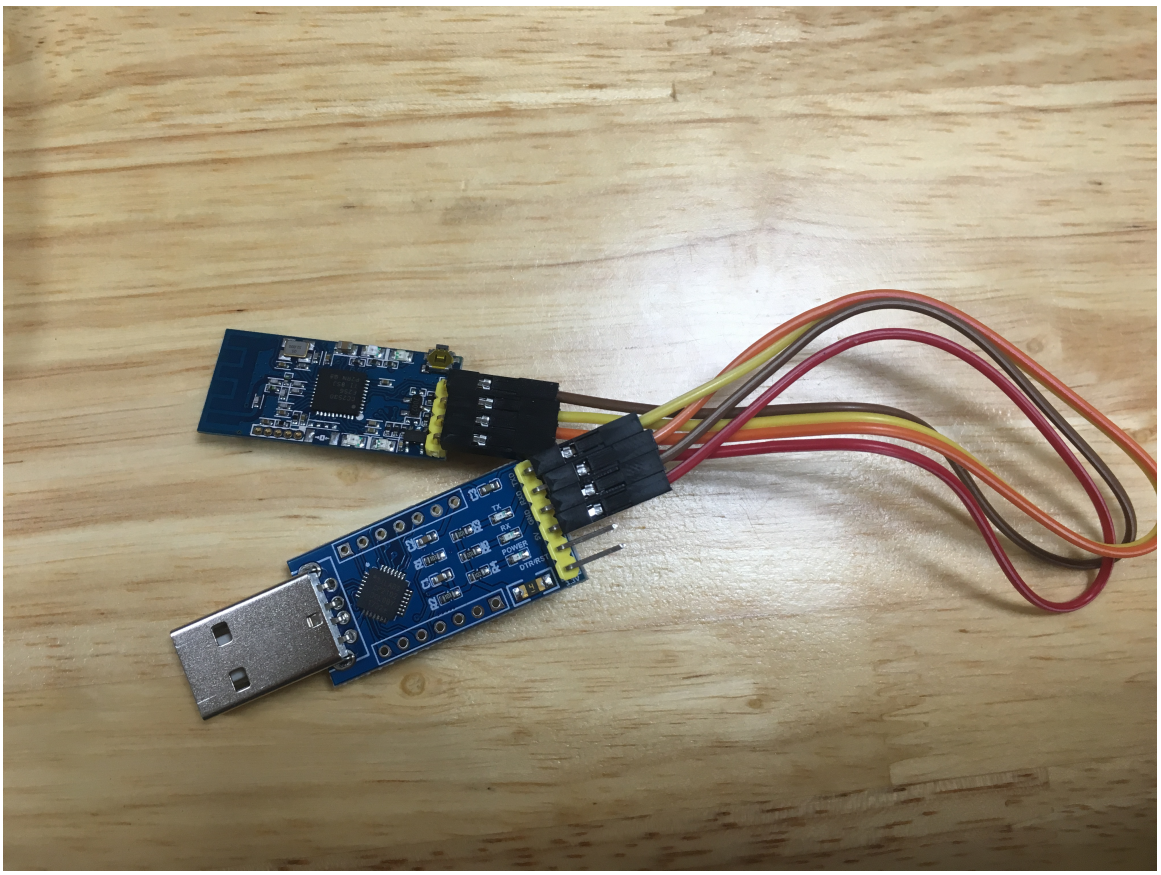

Figure 15: RF Zigbee UART CC2530

USB UART CP2102 This USB converter is using CP2102 from **Silicon labs** which is used to transfer USB to UART TTL and vice versa.

USB UART CP2102 switch can be used on all Windows, Mac, Linux, Android. Support transmission speeds such as 300, 600, 1200, 1800, 2400, 4000, 4800, 7200, 9600, 14400, 16000, 19200, 28800, 38400, 51200, 56000, 57600, 64000, 76800, 115200, 128,000, 153,600, 230,400, 250,000, 256,000, 460,800,

500,000, 576,000, 921,600 and other kinds of speeds.
To use this adapter, we need to install an driver from **Silicon Labs**
To use this adapter and zigbee cc2530:

USB UART CP2102	Zigbee cc2530
5V	5V
GND	GND
Rx	Tx
Tx	Rx

Introduction

Zigbee UART CC2530 V1 RF transceiver circuit uses CC2530 IC from TI, pre-programmed firmware to be easily used as a Zigbee standard wireless data transfer module with UART interface that is easy to connect with micro controller or computer (via USB-UART transfer cable) with only a few config steps using the push button.

Specification

- Power supply: 3 - 5.5VDC
- Current: < 30mA.
- Standard Zigbee 2.4Ghz wave transmission.
- Maximum transmission speed: 3300bps.
- Transmission capacity: 4.5dbm.
- Distance: 250m.
- UART TTL protocol connection (3.3VDC or 5VDC), maximum Baudrate 115200.
- Size: 15.5x31.5mm.

Connecting Arduino with LoRa

Arduino Uno	Lora
5V	5V
GND	GND
Rx	Tx
Tx	Rx

Connecting Arduino with Raspberry Pi

We use a UART-to-USB adapter to connect the zigbee module to our Pi because the UART pins on it are used for the Lora module.

VII. System Overview

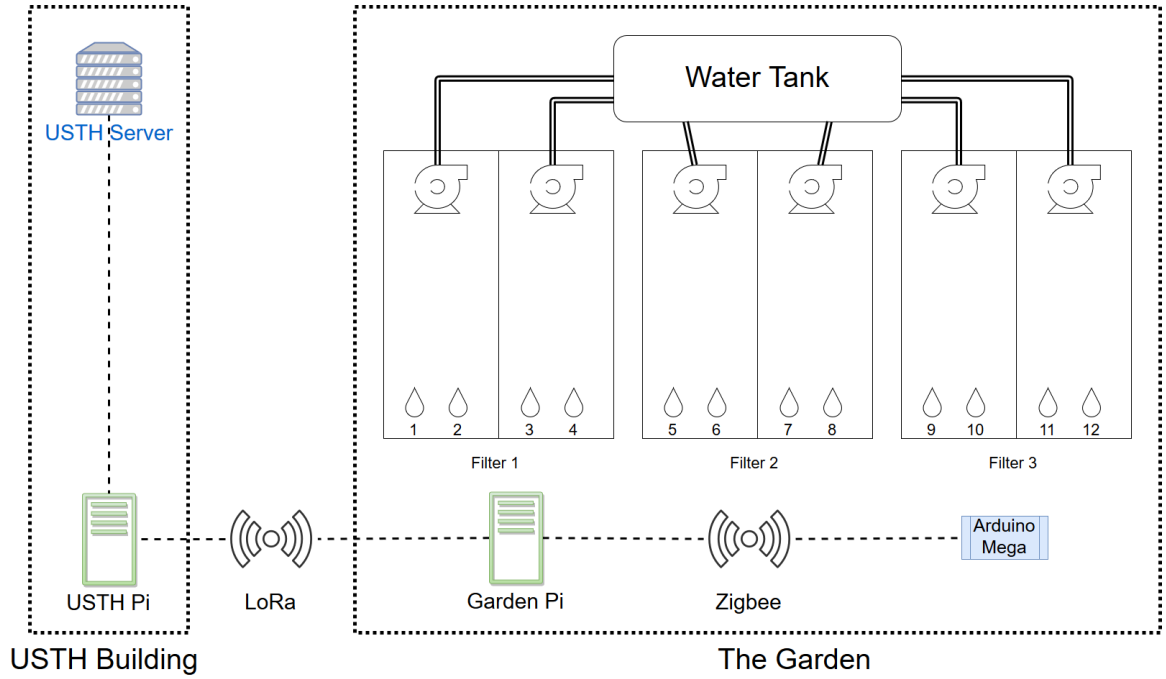

Figure 16: System Overview

Inside the garden, We are using twelve sensors total to measure the soil moisture data in each filter. We have three filters, and each filter has two more sub filters with two sensors. All sensors is connected together with an Arduino Mega.

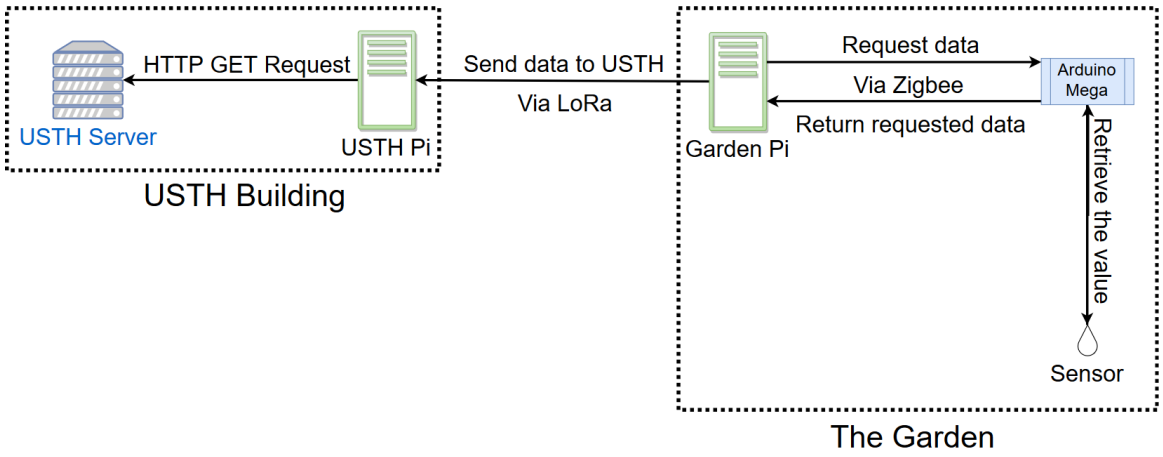

Figure 17: System protocol overview

To transfer data, we use zigbee and LoRa module. Thus the Raspberry Pi cannot read directly the analog value from the sensor so we use an Arduino Mega to do that. And we put one Raspberry Pi (which is attached with a Zigbee and Lora module), the raspberry pi will send a request to the Arduino Mega (which is also attached with a Zigbee module) to receive the data from the sensor. We also setup a cronjob to schedule the time request for the data depend on the time the filters will be supply with wastewaters.

After received the data from the Arduino, The raspberry pi inside the garden will send these data to another Raspberry Pi which is inside the USTH Building by LoRa. And the data will be send to USTH server by HTTP protocol.