Internet of Things

Project Report

Hello-water auto-watering smart IOT gear

with Mr.Massinissa HAMIDI ,Seyed MODARESI

18/1/2019

Participant：

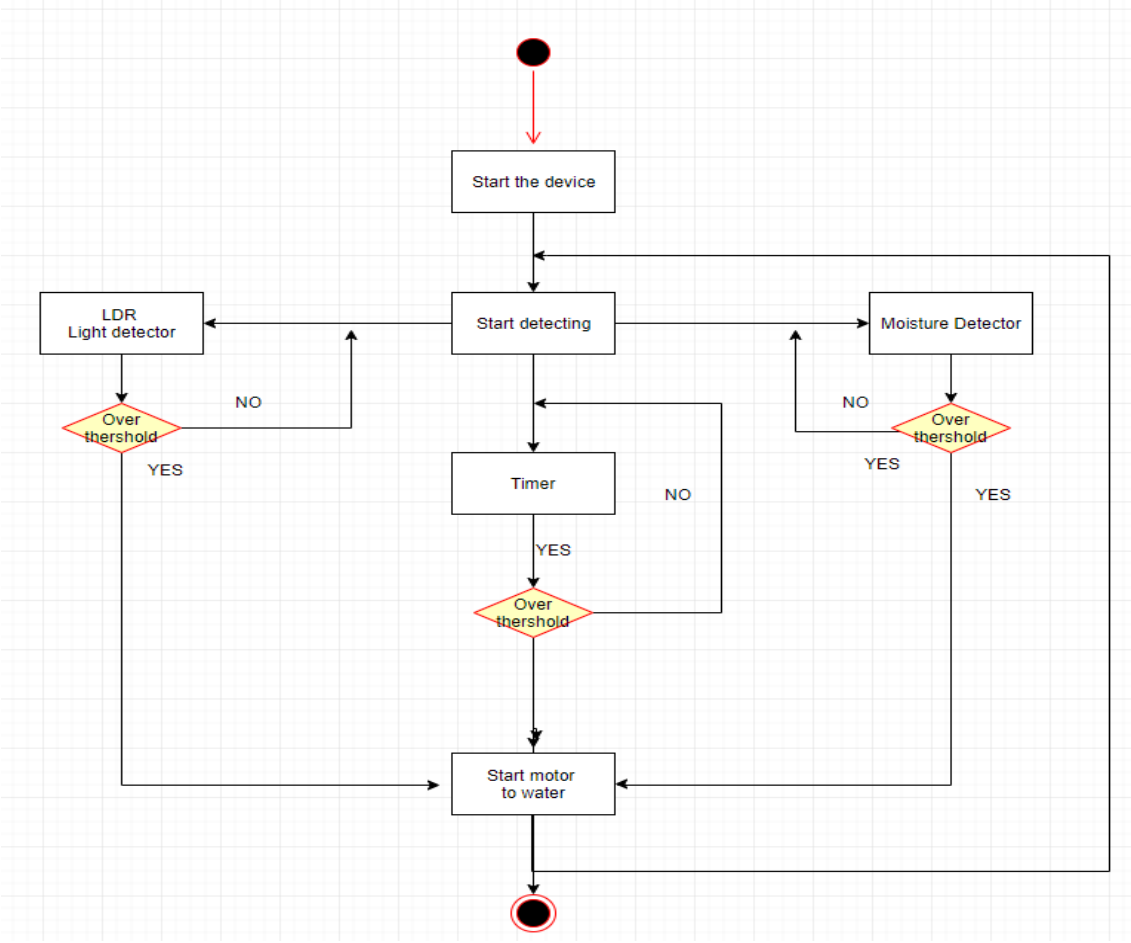
Yanwu ZHU	ISCE	2021
Yaoli MI	ISCE	2021
Ying LI	ISCE	2021

Index

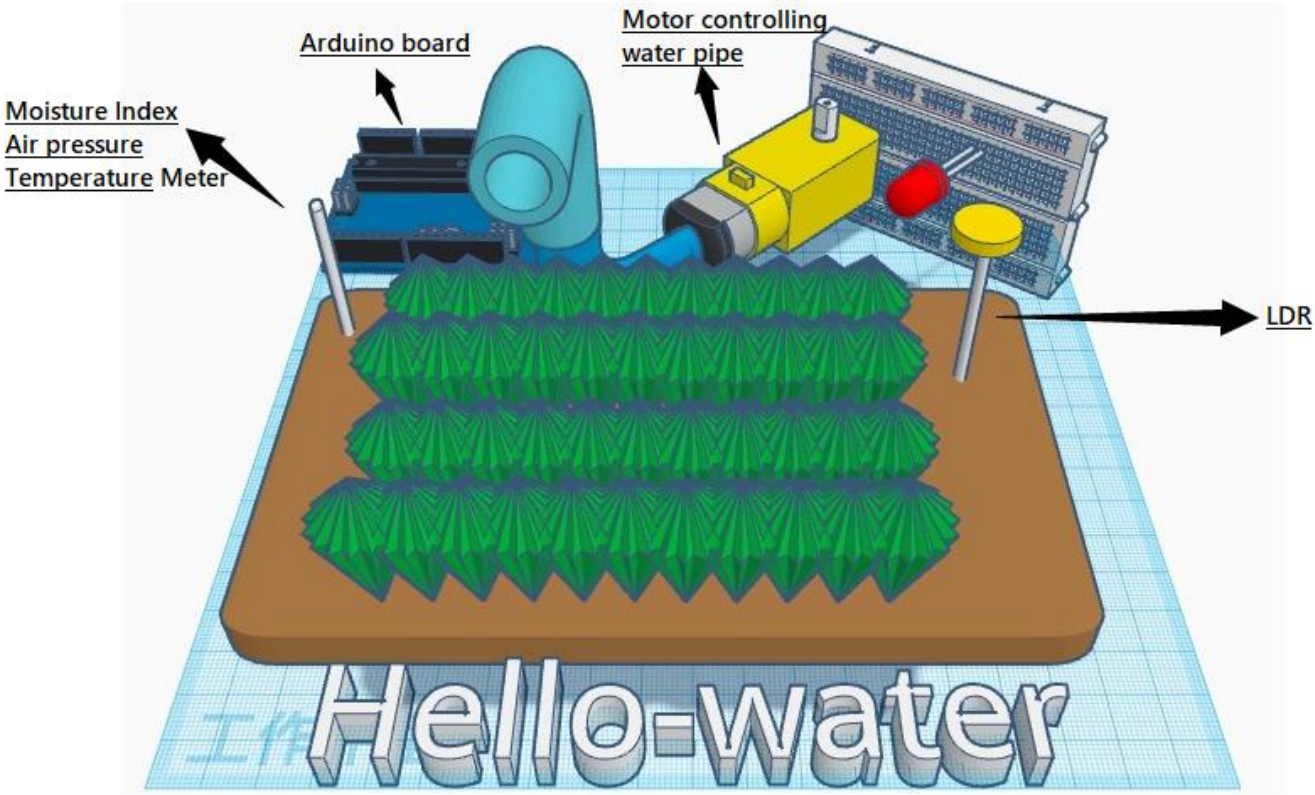
1.1 Project Description

With the development of economy and technology nowadays, we sometimes don't have enough time to tend our plants due to the quickening pace of life. Also, people with no experience may have a tendency to water plants at a wrong time or not sure about the amount of water they need.
Therefore a need of smart & automatic gardener emerges. Our goal is to design a 'gardener' as such with soil moisture detection, Sunlight detection. Automatic watering function in accordance with detection.
From the several detection devices and watering machine. we need Arduino board with different detector , also we need relay and motor (water pump) for the watering function. Also we need to learn how to code on the platform regarding control different components and read parameters like the moisture information and weather condition.

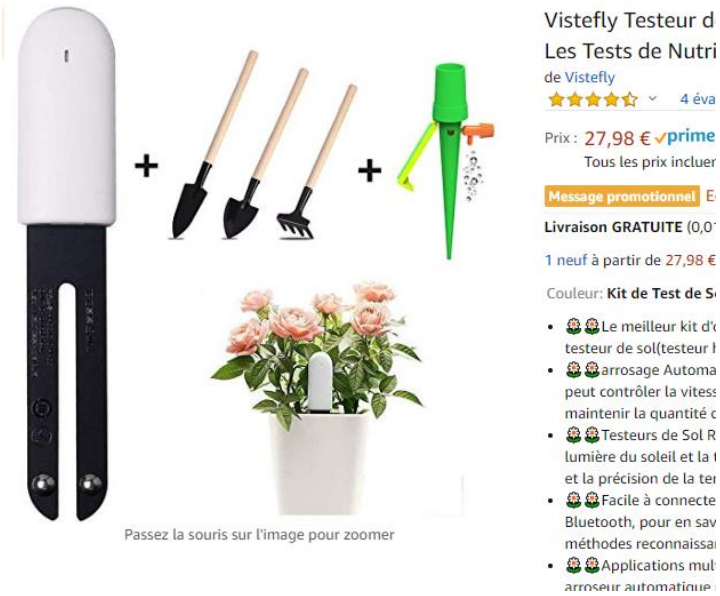
1.2 UML



2.Conceptional Product Image



3. State-of-art analysis



-On the left: Amazon best seller -On the right: Hello-water by our group

With some of the current apps /smart device checked on the Appstore /amazon and eBay. We found that there is not an integrated device or app that combine status detecting , displaying with watering offline.

Compared with this product or all the other products on the market. Our product does have some competitive advantages over others.

	Vistefly from Amazon	Hello-water
Sunlight detection	√	√
Moisture detection	√	√
PH detection	√	× （Expandable）
Water Volume	100 ml	Unlimited (Customized)
Price	28€	10-15€

4.Product component & Code

So far we have finished the part of LDR Sensor & moisture index detectors also weather API simulation completed, mostly verified by the professor of the lab part.

4.1 LDR Sensor

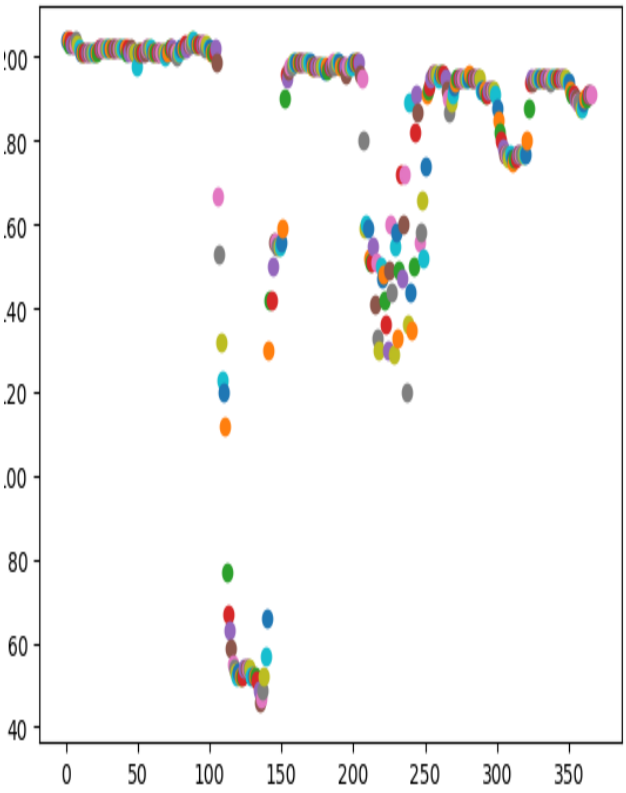


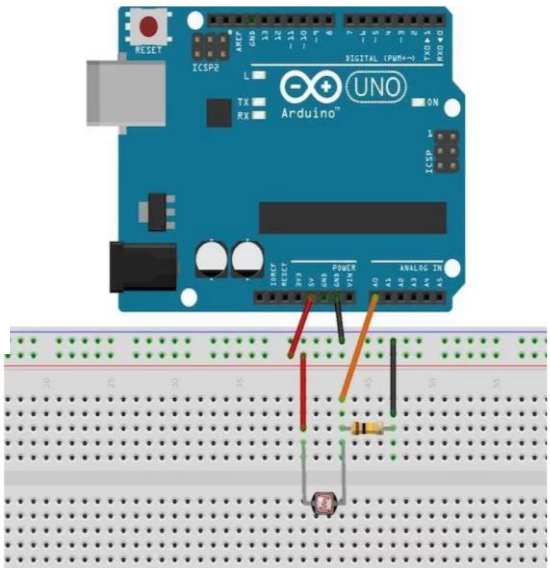
Figure 2 Plotting Output of Data from LDR

4.1.1 LDR Introduction

In order to detect the intensity of light or darkness, we use a sensor called an LDR (light dependent resistor).

The LDR is a special type of resistor that allows higher voltages to pass through it (low resistance) whenever there is a high intensity of light, and passes a low voltage (high resistance) whenever it is dark.

Figure 1 Connection Diagram of LDR circuit



The LDR gives out an analog voltage when connected to VCC (5V), which varies in magnitude in direct proportion to the input light intensity on it. That is, the greater the intensity of light, the greater the corresponding voltage from the LDR will be. See figure 1 to see how we connect LDR circuit.

Since the LDR gives out an analog voltage, it is connected to the analog input pin on the Arduino. The Arduino, with its built-in ADC (analog-to-digital converter), then converts the analog voltage (from 0-5V) into a digital value in the range of (0-1023). When there is sufficient light in its environment or on its surface, the converted digital values read from the

LDR through the Arduino will be in the range of 800-1023.See figure 2 to check the output signal of an LDR circuit.

4.1.2Code for LDR Sensor:

```
#define SensorPin A0

#include <Servo.h>
float sensorValue = 0;
int pumpin=2;

void setup() {
  Serial.begin(9600);
  pinMode(2,OUTPUT);
}

void loop() {
 for (int i = 0; i <= 100; i++)
 {
   sensorValuem = sensorValuem + analogRead(SensorPin);
   delay(1);
 }
  sensorValuem = sensorValuem/100.0;
   if(sensorValuem>650)
   {

digitalWrite(pumpin,LOW);
    delay(3000);

digitalWrite(pumpin,HIGH);
    }
   else
   {

digitalWrite(pumpin,HIGH);
    }

Serial.println(sensorValuem);
 delay(6000);
}
```

4.2. moisture detector

4.2.1 Moisture Detector Intro

Soil moisture meters can tell the resistance of the soil then calculate the moisture index. Nowadays moisture meters can be very accurate, A superb moisture meter can be accurate within 0.1% of the moisture content. Although Lower-grade moisture meters aren't very accurate as numbers are misleading and changing.

For the moisture detector, We successfully made it functional.
Instead of using the provided moisture detector, which is not suitable for soil moisture detection, we use the purchased moisture from online to specially detect the soil moisture .
We set the threshold to a certain number and if the soil moisture is below the threshold , suggesting it's too dry and need water, the system will pull water pump to high to start watering .

4.2.2 Code for moisture detection:

```
int sensorPin = A0; // select the input pin for LDR
int sensorValue = 0; // variable to store the value coming from the sensor
void setup() {
Serial.begin(9600); //sets serial port for communication
}
void loop() {
sensorValue = analogRead(sensorPin); // read the value from the sensor
Serial.println(sensorValueL); //prints the values coming from the sensor on the screen
   if(sensorValueL>650)
   {
digitalWrite(pumpin,LOW);
    delay(3000);
digitalWrite(pumpin,HIGH)
    }
   else
   {
digitalWrite(pumpin,HIGH);
    }
Serial.println(sensorValueL);
 delay(6000);
}
delay(100);

}
```

4.3 Weather API

4.3.1 Description

In order to realize the watering function regarding weather forecast. We connected to a weather API to fetch json structure text and then compare to pre-set index to judge the need of water.
To test this function, we run a simulation to a weather API website to fetch the pre-set weather text.

4.3.2Code for weather API:

```
=    if ((WiFi.status() ==
    WL_CONNECTED)) { //Check the
    current connection status
        HTTPClient http;
        DynamicJsonDocument doc(1024);


    http.begin("https://samples.openwea
    thermap.org/data/2.5/weather?q=Lond
    on,uk&appid=b6907d289e10d714a6e88b3
    0761fae22"); //Specify the URL
        int httpCode = http.GET();
    //Make the request

        String payload;
        if (httpCode > 0) { //Check for
    the returning code
          String payload =
    http.getString();
          Serial.println(httpCode);
          Serial.println(payload);
          DeserializationError error =
    deserializeJson(doc, payload);
          if (error) {

    Serial.print(F("deserializeJson()
    failed: "));

    Serial.println(error.c_str());
           delay(120000);
          return;
        }
        // Get the root object in the
    document
        JsonObject root =
    doc.as<JsonObject>();

        float temperature =
    root["main"]["temp"];
        int humidity =
    root["main"]["humidity"];
        int pressure =
    root["main"]["pressure"];

        Serial.println(humidity);
        delay(2000);

        Serial.println(pressure);
        delay(2000);
        temperature =
    round(temperature);
        Serial.println(temperature);
        delay(30000);
        } else {
          Serial.println("Error on HTTP
    request");
          delay(30000);
        }
        http.end(); //Free the
    resources
      }
    }
```

5.Problem Encountered

5.1 Multiple Device on one board

Problem Description: Since our product has the function of detecting multiple index for watering reference, there should be multiple devices connected to one Arduino board, However we didn't realize such function , instead we use one module to one board at a time and change the module to obtain different parameter regarding to different environmental aspects.

Possible solution: After thorough consideration, we found that we may use I2C technique to link different device to one board , detailed technique can be referred from the lab 4&5 we've accomplished . In regard to the unique main program , I suggest that we could use serval sub-loop program in the same main program . With different pin number different parameter set and I2C technique , we may well can realize 1 to more device connection.

5.2 Moisture detection

Problem description: The provided moisture detection module could satisfy our need for moisture detection.

Solution: We purchase another one from Amazon which is specialized in detecting moisture index in the soil.

6. Participation &Budget

6. 1 RACI Table for team participation

	Connection& assembling	LDR Code &Test	Weather API Code &Test	Moisture Code &Test
Yanwu ZHU	**C**	**C**	**C**	**R/A**
Ying LI	**R/A**	**R**	**R/A**	**C**
Yaoli MI	**C**	**R/A**	**C**	**C**

	Group Administration	PPT	Report	Video
Yanwu ZHU	**A**	**C**	**R/A**	**R/A**
Ying LI	**/**	**R/A**	**C**	**C**
Yaoli MI	**/**	**C**	**C**	**C**

6.2 Budget

Purchase from Amazon ：

Title：RUNCCI-YUN Kit d'arrosage Automatique pour Arduino DIY　From RUNCCI-YUN

Price:　10,99 €

Link :https://www.amazon.fr/gp/product/B0814HXWVV/ref=ppx_yo_dt_b_asin_title_o01_s00?ie=UTF8&psc=1