INSTITUTE OF
TECHNOLOGY OF CAMBODIA

DEPARTMENT OF
ELECTRICAL AND ENERGY ENGINEERING

REPORT of Student Project

“Auto Watering System”

Lecturer: **Mr. HEL CHANTHAN**

Student: **LOR CHHUNHENG** **(e20140370)**

Group: **I4-ET**

ACADEMIC YEAR: 2018 – 2019

Table of Contents

1. Introduction

Farming is an important part of the country development, the product of farming support all life in country.

Farming system is process according to the environment, there are many condition of a plant growing. The plant need two main supplies are water and fertilizer to grow. However, plant also need some environment condition like temperature, soil, air …etc. for its grown too.

2. Objective

While a plant is grown, the water supply is a supply that the plant need the most for growing condition. The water is supplied for plant one- two times a day in general. In this case, farmer should go to farm every day to do the water supply.

According to the plant biology, any type plants need a level of soil for growing, and that is why its need water in period. So, there is a relationship between the water supply and the growing of plant that connected by the soil condition of the ground.

Auto watering system, is made for automatic watering the plant in farm in condition of the change of percent of ground soil.

The system progress is divide into two phases:

- System design
- System development

3. Diagram of Process

To allow the connection between the soil data and the watering controller, we need a system for transmit the data and access on flow to make the system working smoothly.

In general, the basic of a system is work in three parts: Input, Operation, and Output.

For Auto watering system, Input part is the process of sensor detection, Output and Operation part is the work flow of the controller.

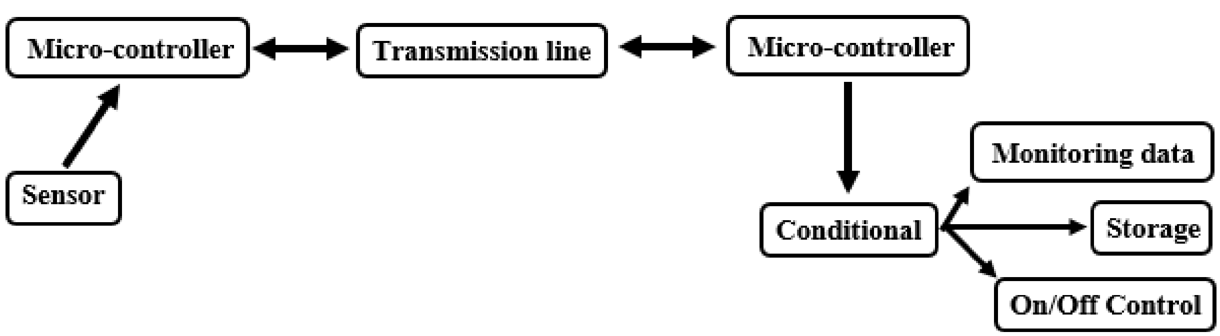


Figure 1 (Auto watering system communication diagram)

3.1.Sensor detection

The system of gathering data from sensors, we use the Arduino micro-controller board. With the abilities of programmable Arduino board could allowed many flow conditions to the system.

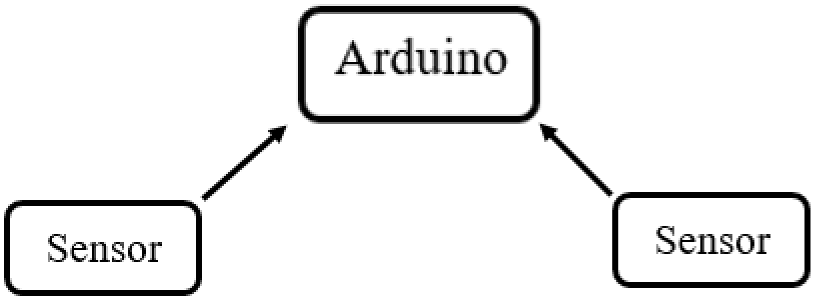


Figure 1 (Flow diagram of sensor data in Nodes)

3.2.Controller and nodes communication

To improve the data traffic for controller in collecting data, we need to develop the best topology for the system to avoid of delay and collusion of data package.

In this case, we choose a flow condition that the controller is the decider. The controller has a time division for request to get the sensor data from all nodes, the node is already contain the data and send it to the controller only when they got the request from the controller.

This communication system is working with the chosen wireless module is XBee module.

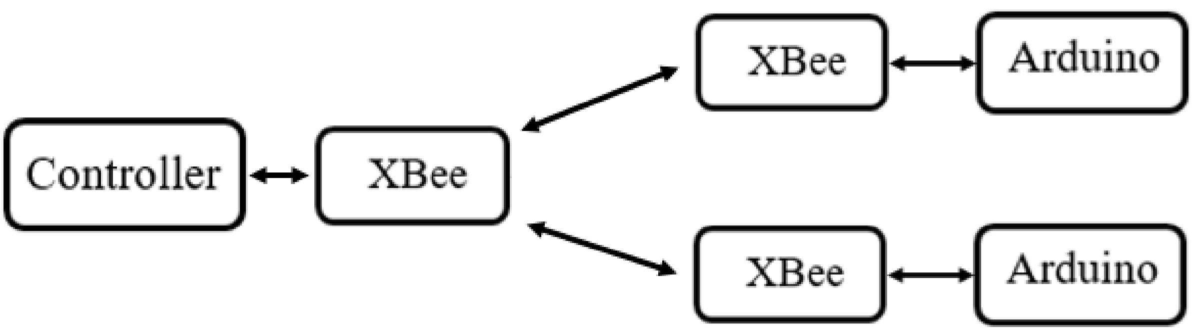


Figure 2 (Communication diagram between Controller and Arduino(s))

3.3. Power supply system

Power supply is also the problem of the system. In the experiment, we use 12v solar power supply with a supporter 12v battery and a converter device that can support two output sources: 5v and 15v. (All voltages are DC voltage)

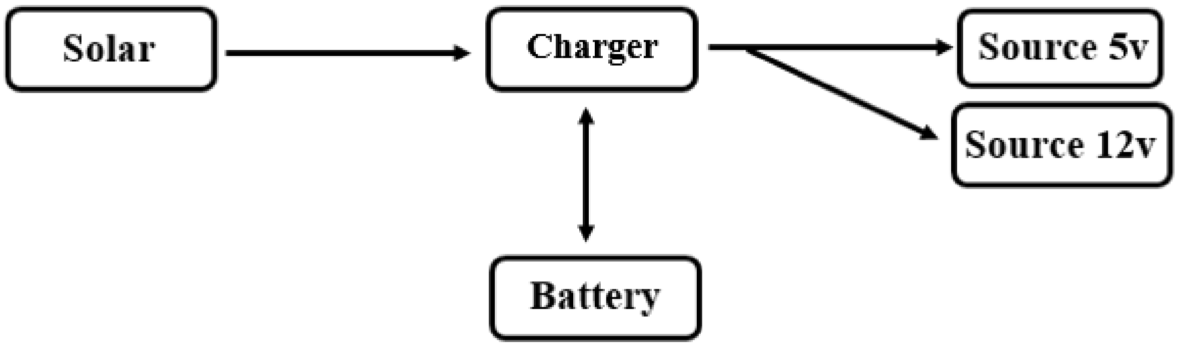


Figure 3 (Power suppy system)

4. System design phase experiment

4.1. Nodes communication diagram

There are Arduino board in every nodes and controller. So, the main programming for the flow of this system is working by Arduino programming. The communication of the controller, the flow is control by the main controller (coordinator) to timing and processing the data transfer as shown in Figure 4.1 In addition to the programming, we used some libraries for making the flow more comfortable. But, some devices are necessary to import its library to run in the system.

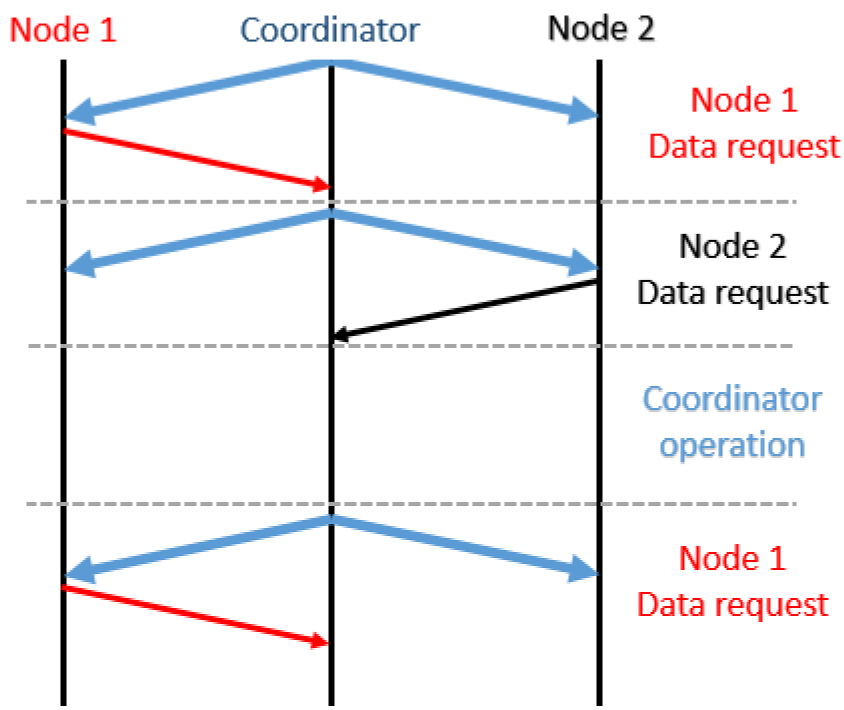


Figure 4.1 (Diagram of communication between controllers in system)

The main controller send request to every nodes one by one with a delay time to get the sensor data.

4.2.Node coding

The controller in node has function of gathering data from sensor then send the data to the main controller. In communication system, the synchronization between transmitter and receiver is necessary to avoid the miss receiving data at the receiver. In this case, the flow code of node should be ready for receiving the request, then do processing of sending data after received the request.

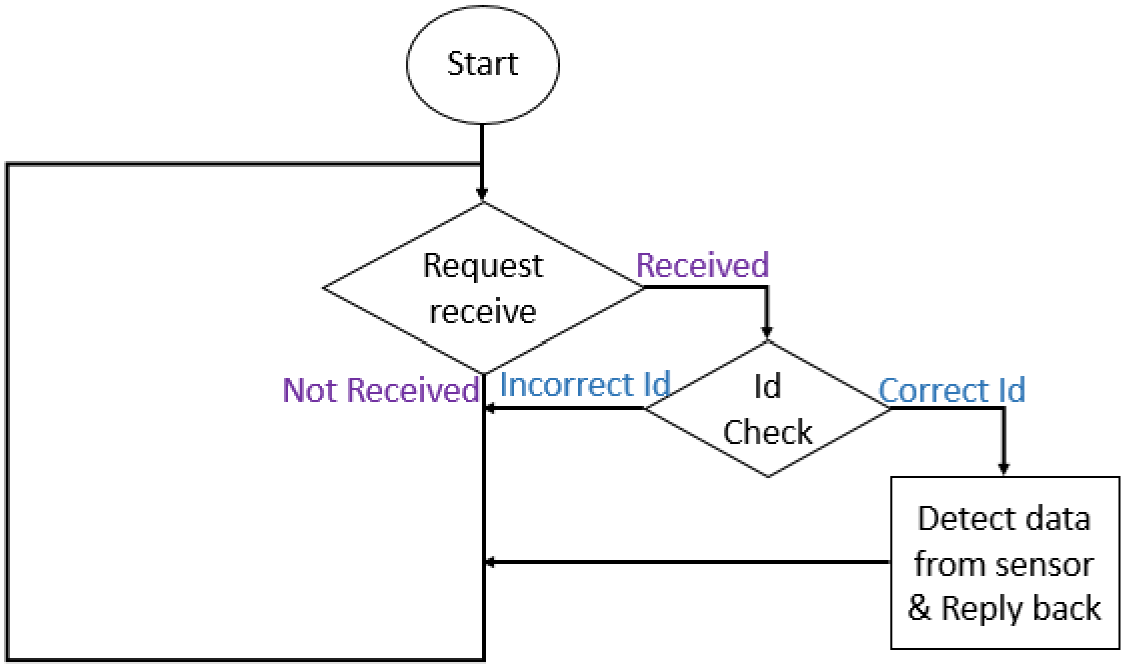


Figure 4.2 (Diagram of node coding)

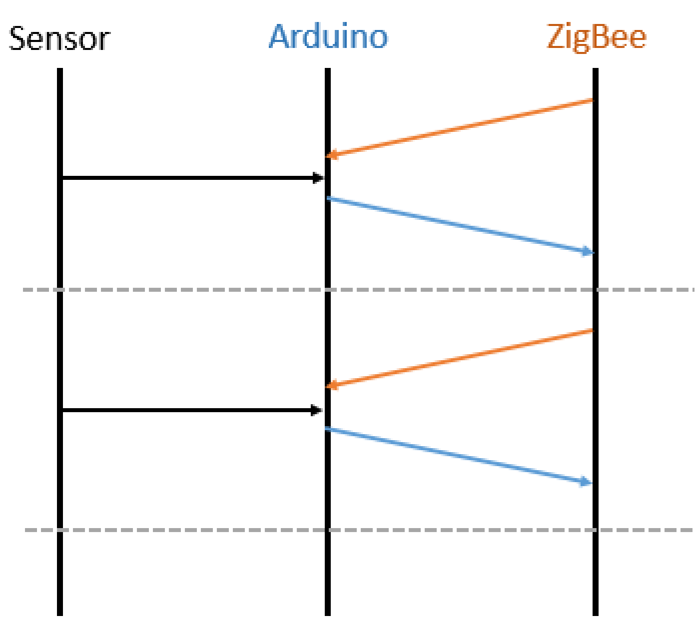


Figure 4.3 (Node communication diagram)

node_2 §

```
#include <SoftwareSerial.h>
#include <XBee.h>

SoftwareSerial XBee(2,3);
int data;
void setup() {
  Serial.begin(9600);
  XBee.begin(9600);
  pinMode(A0,INPUT);
}
void loop() {
  if (XBee.available()>0){
    int id = XBee.read();
    if (id == 202){ //Id detector
      data =(analogRead(A0)-305)/0.45;//operation for watering condition
      if (data <= 0){
        data = 0;
        }
      if (data >= 100){
        data = 100;
        }
      XBee.write(data);
      }
    }
}
```

Figure 4.4 (Node coding)

4.3. Coordinator coding

The main controller of the system is use for collect the data from many Nodes. The process of communication is the coordinator send ID to each node in order, and the Node will send back the data if the ID is correct with their ID.

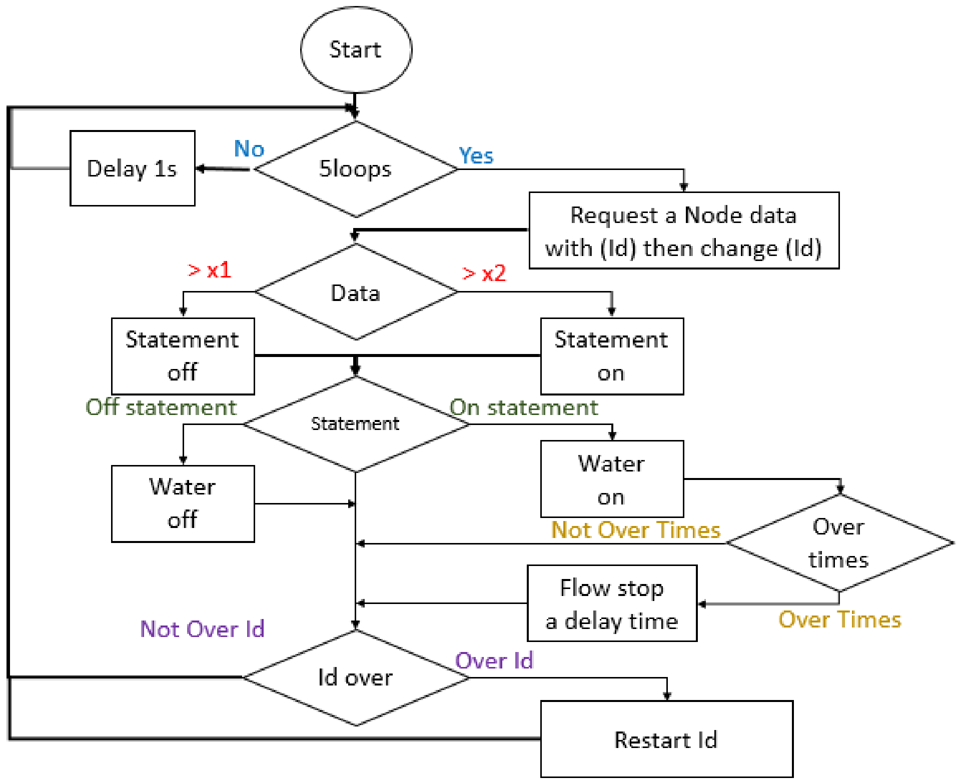


Figure 4.5 (Flow chart of Coordinator coding)

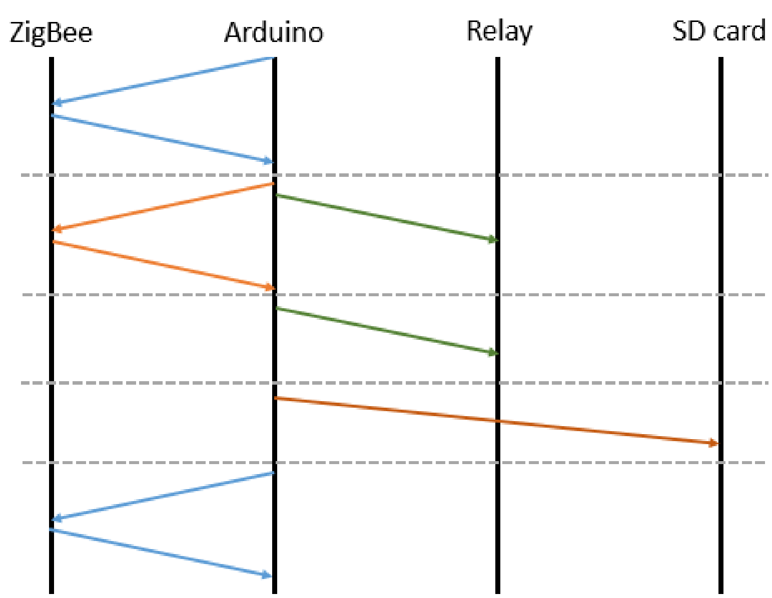


Figure 4.6 (Flow chart of Coordinator coding)

coordinator_code

```
#include <SoftwareSerial.h>
#include <XBee.h>
#include <SPI.h>
#include <SD.h>

SoftwareSerial XBee(2,3);//rx tx
int p = 0;//loop for data transfer
int id = 2;//id of node
int d;//for detect
int data[] = {0,0,0};//for save
int n = 1;//interation for data[]

int datal_nodel=0;int datal_node2=0;
int val;//additional for identify the id
int ti;//condition for error
int q;//water controller
void setup() {
  pinMode(6,OUTPUT);
  Serial.begin(9600);
  XBee.begin(9600);
}
```

```
void loop() {
  p++;
  //The request data from node condition
      if (p == 5){
        val = 200 + id;
        XBee.write(val);//Send request
        //Wait for replycondition
        for (int i =0; i <5000;i++){
          if (XBee.available()>0){
            d = XBee.read();
            data[1] = d;
            d = 0;
            }
          }
         if (id == 2){
          datal_nodel = data[1];
          data[1] = datal_node2;
          }
         if (id == 3){
          datal_node2 = data[1];
          data[1] = datal_nodel;
          }
         id++;
```

Figure 4.7 (Coordinator coding parameter configuration and requesting)

```
         //Watering condition
         if (datal_nodel == 0 | datal_node2 == 0){
           q = 1;
           }
         if (datal_nodel == 100 | datal_nodel == 100){
           q = 2;
           }

         //Watering control
         if (q == 1){
           digitalWrite(6,LOW);
           ti++;
           if (ti == 750){
             digitalWrite(6,HIGH);
             delay(900000);
             ti = 0;
             }
           }
         if (q == 2){
           digitalWrite(6,HIGH);
           ti = 0;
           }
         p = 0;
         if (id == 4){
           id = 2;
           delay(3000);
           //delay(299850);//time for run full cod is 2150ns
         }
         }
  delay(1000);
}
```

Figure 4.8 (Coordinator coding condition part)

4.4. Experiment

The Experiment is testing with the farming system, there is a system that install for automatic watering of plants using a water stock. At the design part, the system is installed follow the diagram of connection and the diagram of communication. The plan of the installation of system, is to set up Nodes inside the farm area and the coordinator is installed at a safety place.

Figure 5.1 (Node installation)

Figure 5.2 (Coordinator installation)

4.5. Result

The result of experiment after installed for few months, the plant is grown.

Figure 6 (Result of experiment)

5. Development phase (Raspberry as a coordinator instead of Arduino)

5.1. Raspberry pi communication

Raspberry pi is a type of controller that has performance closer to PC controller than Arduino controller. The communication of the system is change by the request function is changed to the system begin function, and this function is given to a Node to do.

With the Raspberry pi controller, there is a database function that exist on that device. So, the communication of the system has such different from the designed system.

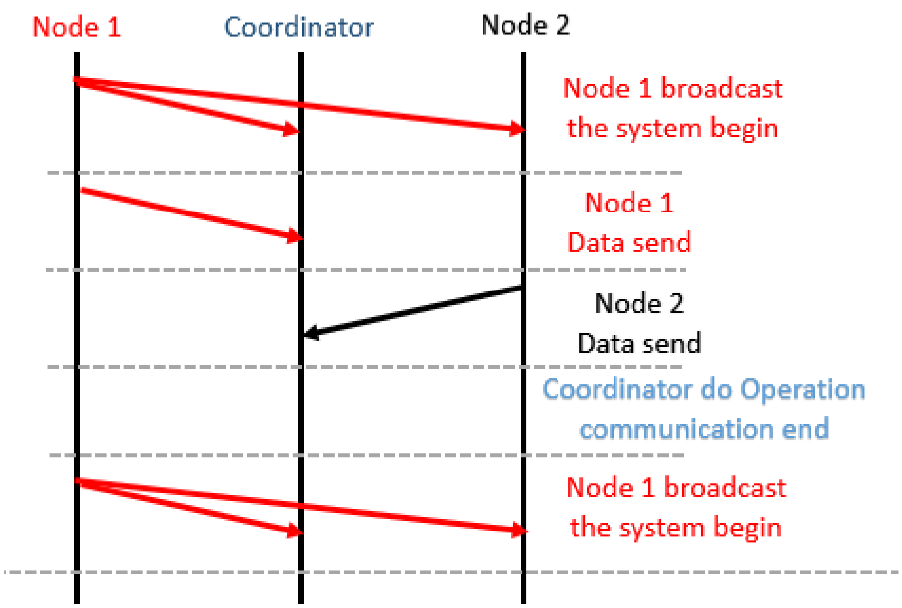


Figure 5.1 (Communication diagram of development system)

5.2. Database on Raspberry pi (MySQL database)

With the support of Raspberry pi device with the PC operation system, there are available database software that can be installed to use on the Raspberry pi.

MySQL is a database software, that able to save the list of data on a PC hardware. And this database is working on low specification PC. So, we can use this database on Raspberry pi.

By the way, to be easy to store and view the data on MySQL there are some software that installed for improve the system store and system view.

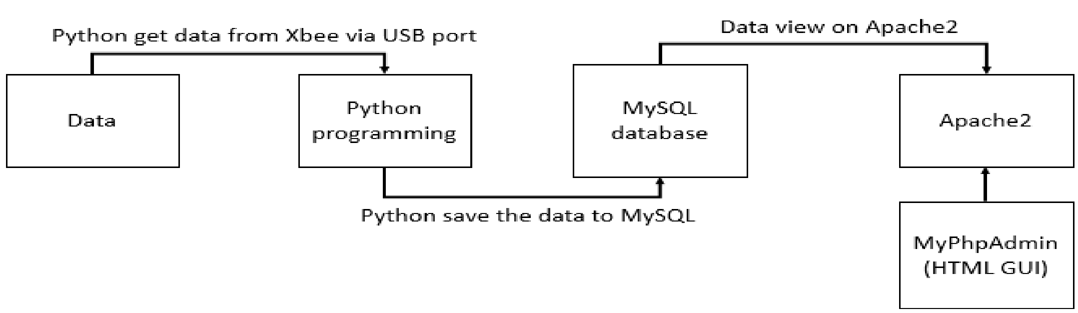


Figure 5.2 (MySQL database system)

5.3. Python coding

In the installation of database system, the view part is install follow the software process, and the saving system is install by the decided of the system communication with the python coding.

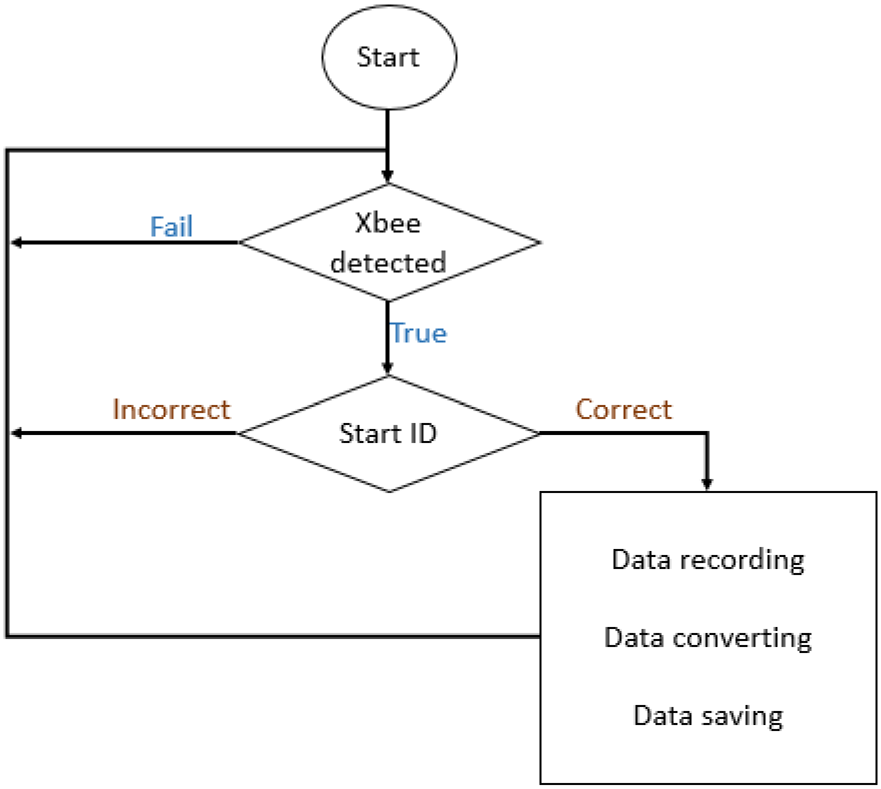
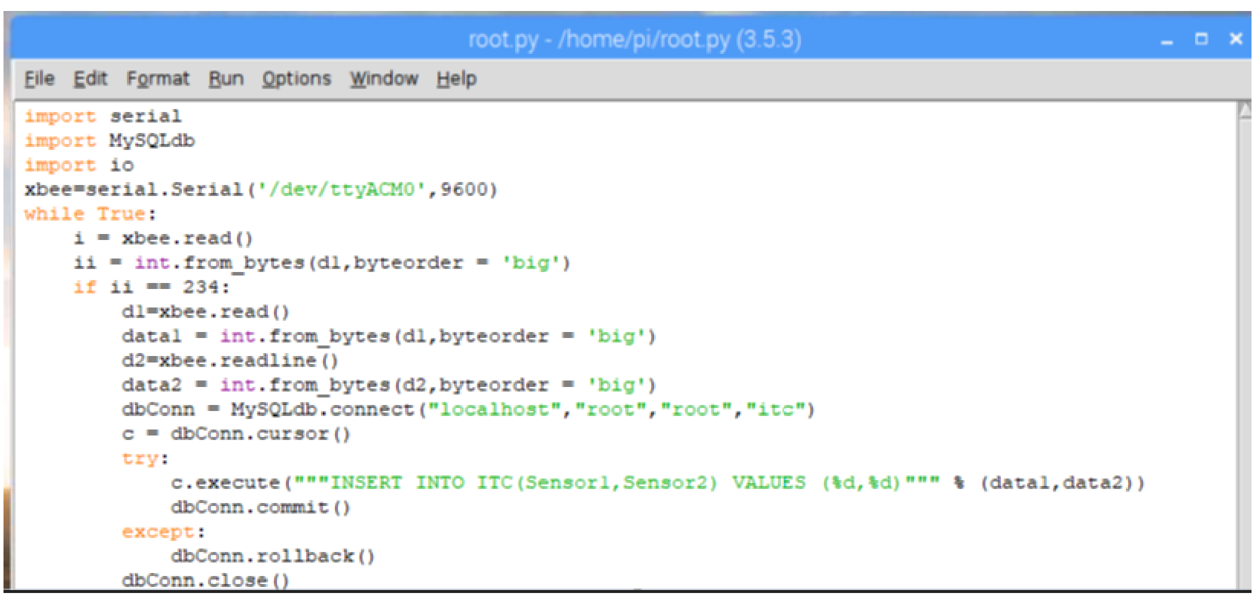


Figure 5.3.1 (Python coding flow chart)



root.py - /home/pi/root.py (3.5.3)

File Edit Format Run Options Window Help

```
import serial
import MySQLdb
import io
xbee=serial.Serial('/dev/ttyACM0',9600)
while True:
    i = xbee.read()
    ii = int.from_bytes(d1,byteorder = 'big')
    if ii == 234:
        d1=xbee.read()
        data1 = int.from_bytes(d1,byteorder = 'big')
        d2=xbee.readline()
        data2 = int.from_bytes(d2,byteorder = 'big')
        dbConn = MySQLdb.connect("localhost","root","root","itc")
        c = dbConn.cursor()
        try:
            c.execute("""INSERT INTO ITC(Sensor1,Sensor2) VALUES (%d,%d)""" % (data1,data2))
            dbConn.commit()
        except:
            dbConn.rollback()
        dbConn.close()
```

Figure 5.3.2 (Python Code)

5.4. Node coding

start_node

```
#include <SoftwareSerial.h>
#include <XBee.h>
int k = 0;
int start = 234;
SoftwareSerial XBee(2,3);
void setup() {
  Serial.begin(9600);
  XBee.begin(9600);
}

void loop() {
  delay(5000);
  XBee.write(start);
  XBee.write(k);
  k++;
  if (k == 256){
    k = 0;
    }
}
```

node2

```
#include <SoftwareSerial.h>
#include <XBee.h>
int k = 2;
SoftwareSerial XBee(2,3);
void setup() {
  Serial.begin(9600);
  XBee.begin(9600);
}

void loop() {
  if (XBee.available()>0){
    int j = XBee.read();
    if (j == 234){
      XBee.write(k);
      k++;
      if (k == 256){
        k = 0;
        }
    }
  }
}
```

Figure 5.7 (Arduino Node coding)

5.5.Experiment

The experiment is to test the transmission data from 2 nodes to MySQL on coordinator and view on MyPhpAdmin GUI.

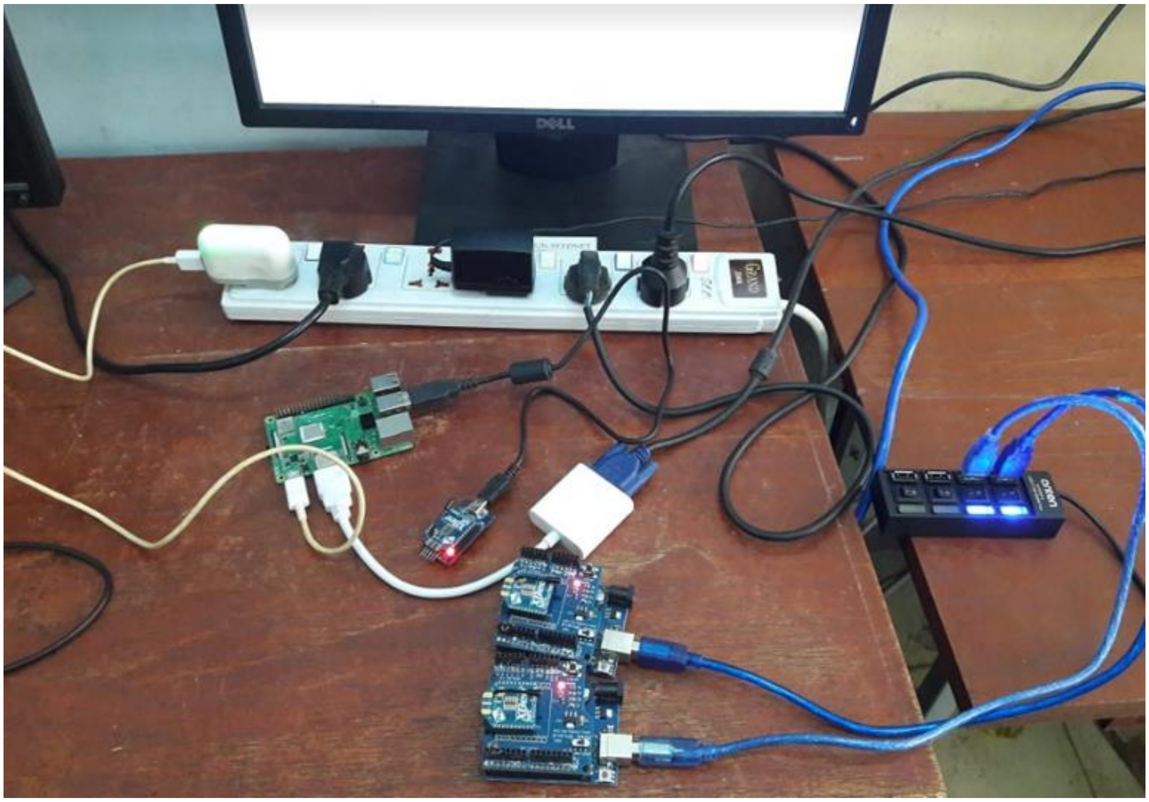


Figure 5.4.1 (the connection of the experiment)

5.6. Result

After the experiment the data transmission is correctly between the node data and MySQL data. The test data are node 1 data is the number that run from 0 to 255 in every transmission, node 2 data is the value of node 1 data plus 2.

Notice, MySQL is a database that working without a specific GUI. Apache2 is a server configuration software that create a web server as a localhost for a device (Apache2 is supported version for Raspbian). MyPhpAdmin is a popular GUI that create with PHP programming (PHP5 for Rasbian) the same as a HTML file that use for giving GUI to a web server.

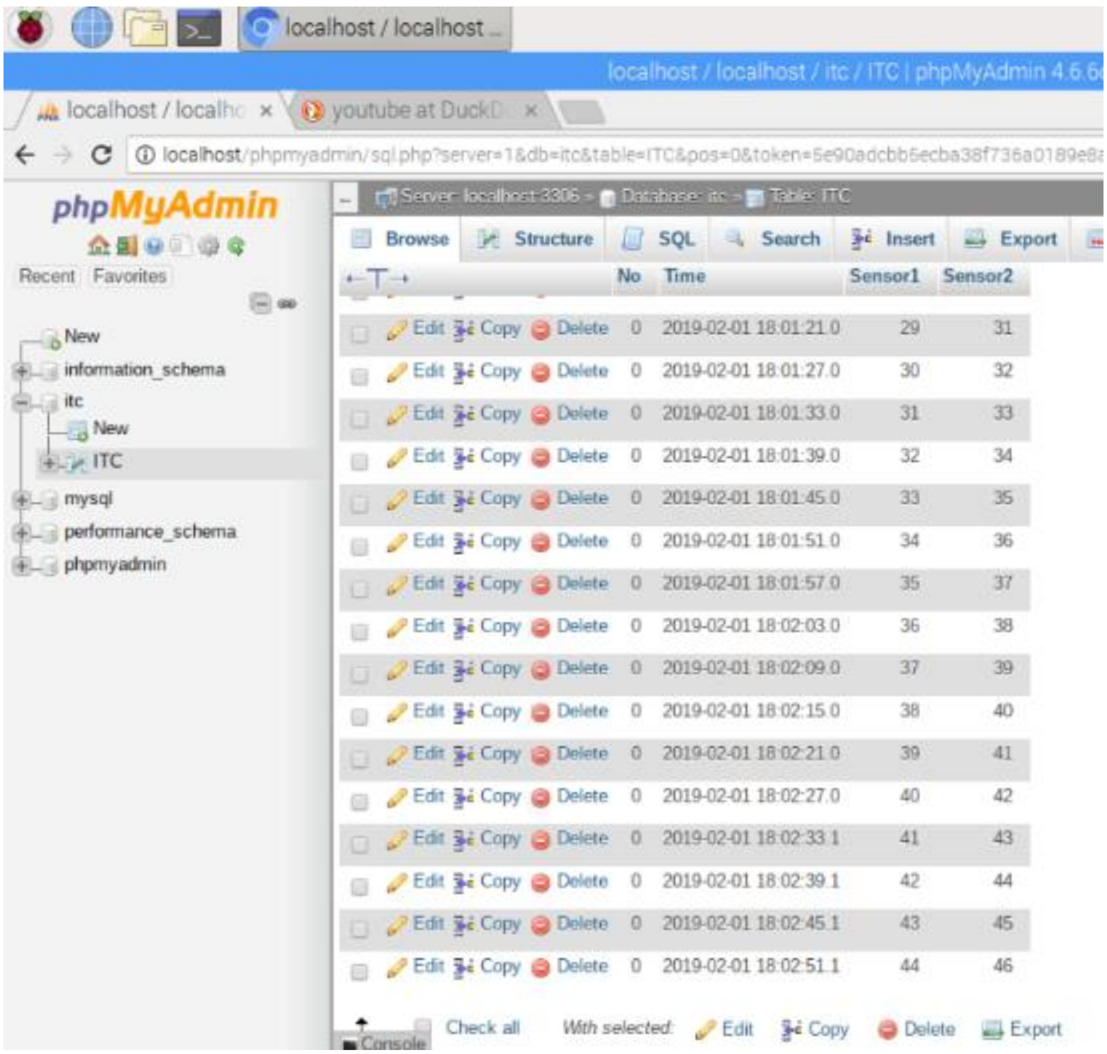


Figure 5.8 (Data collected on MySQL display with MyPhpAdmin GUI)

6. Conclusion

The phase 1 design system is able to use for farming system. But, the system is needed to develop for another condition to view the system process, to view the data collected, and to get the access from the user in different area.

For the second phase of system improvement is the development of upgrading the ability of the coordinator, there is a change of coordinator controller that increase the out boundary communication and the storage ability of the coordinator. The system in second phase is tested in data communication, but there is an output condition that need to be test for the watering control from the Raspberry pi controller by the GPIO pin.

7. Equipment for Experiment:

7.1. Arduino

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm

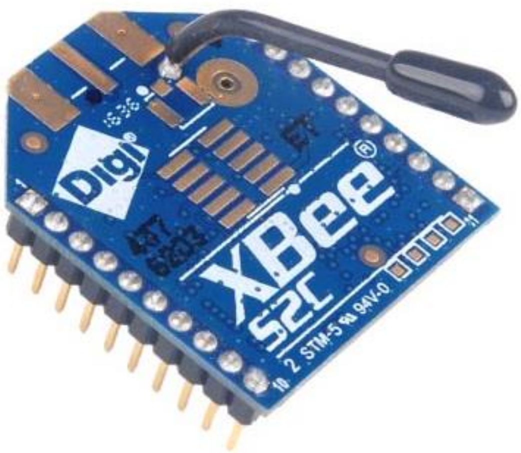
Figure 7.1 (Arduino Controller)

7.2.Raspberry pi

Raspberry Pi is the high performance board. Raspberry Pi 3 is good to use with the processor of 1.2GHz, 1GB ram, and the wireless adapter.

Figure 7.2 (Raspberry pi)

7.3.XBee (ZigBee module)



Specification	XBee Zigbee S2C
Indoor/urban range	Up to 60 m (200 ft)
Outdoor RF line-of-sight range	Up to 1200 m (4000 ft)
Transmit power output (maximum)	6.3 mW (+8 dBm), boost mode 3.1 mW (+5 dBm), normal mode channel 26 max power is +3 dBm
RF data rate	250,000 b/s
Receiver sensitivity	-102 dBm, boost mode -100 dBm, normal mode

Figure 7.3 (ZigBee module)

7.4. Power supply system

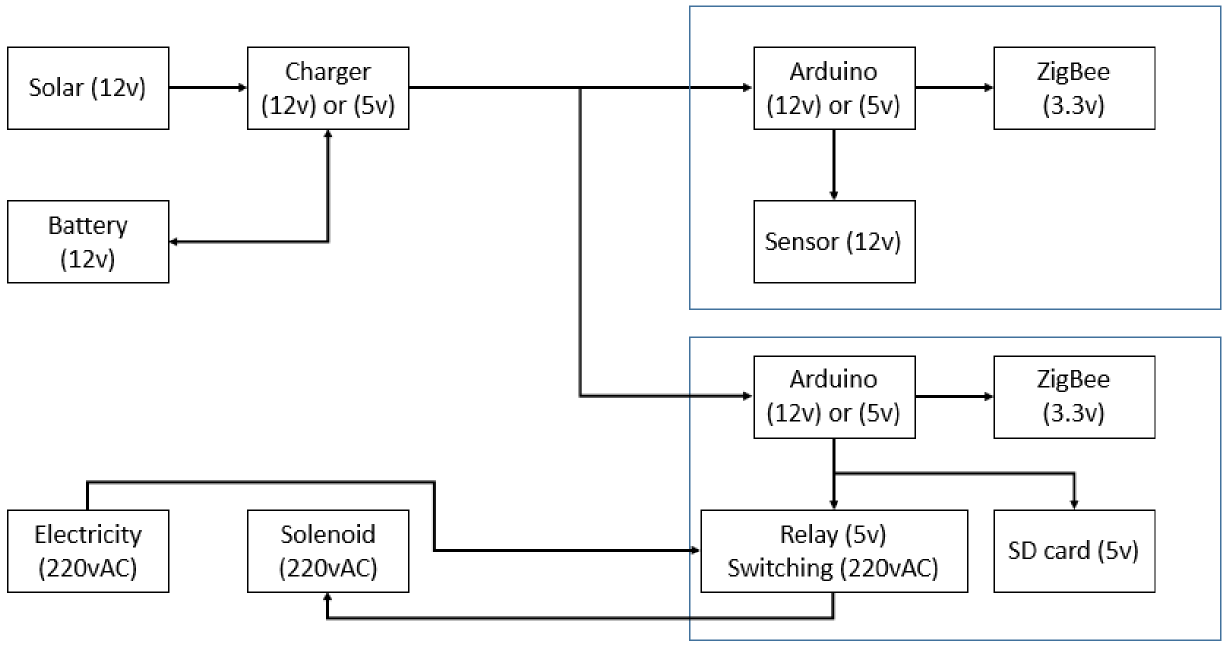


Figure 7.4 (Power supply diagram)