Plant Monitoring System
Robotics Design Report

Team 1, Group A&B
Phuong Hoang Nguyen (Leader) - 14939
Nguyen Dang Khoa - 14322
Ngo Phuc Linh - 14327
Dang Nhat Quang - 15606

Technical Writing
Foundation Year Studies
Vietnamese-German University
8 June 2019

Phuong Hong Nguyen
132/7A Nguyen Trong Tuyen Street,
Phu Nhuan District, HCMC, Vietnam

6 June 2019

Mr. Richard G. Bradley, MAL, MSM
Vietnamese German University
Foundation Year Studies
Nam Ky Khoi Nghia Street, Hoa Phu District
Binh Duong New City, Binh Duong Province

Dear Mr. Bradley,

Enclosed is our final report entitled "*Robotics Design Report: Plant Monitoring System*" submitted to partially fulfill the requirements of the course "English for Computer Science 4", as specified on 22 April 2019.

This report contains the context of a Plant Monitoring System in the embedded microcontroller project, gives a detailed of hardware and software description. Every single component assembled the system as well as schematics of the design are also included. Additionally, we have attached a Gantt chart to illustrate the progress of the whole project.

Your consideration of this report is greatly appreciated, and we hope you find it is satisfactory. However, if you have any further questions or concerns, please contact us without any kind of hesitation either by telephone 0906883192 or e-mail: jonnidip18@gmail.com.

Sincerely yours,

Phuong Hong Nguyen
Team Leader

Enclosure: "*Final Report_Team 1_Group A&B*" (1 Copy)

Vietnamese-German University
Foundation Year Studies
English for Computer Science 4

Project Design Report: Plant Monitoring System

Team 1
Phuong Hong Nguyen (Leader)
Nguyen Dang Khoa
Ngo Phuc Linh
Dang Nhat Quang

Instructor: Richard G. Bradley, MAL, MSM
Due date: 8 June 2019

Disclaimer

We declare that this report is a product of our own work, unless otherwise referenced. We also declare that all opinions, results, conclusions and recommendations are our own and may not represent the policies or opinions of the Vietnamese-German University.

Phuong Hong Nguyen (Leader)

Nguyen Dang Khoa

Ngo Phuc Linh

Dang Nhat Quang

The Vietnamese-German University

Foundation Year

E4CS4, Special Projects

Abstract

Plant Monitoring System, Version 1.0

Team 1, Group A&B

Due to the development of farming industry, the need of automatic farming is rising. Thus, coming up with an idea to contribute a solution for automatic agriculture is essential.

This report involves the design and coding of the plant monitoring system which includes sensor for recording the soil moisture using the Raspberry Pi, Arduino and Python programing language. The machine will automatically water the plants by using submersible pump together with relay module to supply water to the plant.

Furthermore, a user interface website was created in order to interact with the users. The website is programmed by programming languages such as Python for back-end, HTML and CSS for front-end of the website. The users can set the watering system to be automatic or they can turn on or off the water pump via website.

Acknowledgments

We would like to express the sincere gratitude to our course instructor Mr. Richard G. Bradley at the first place for providing his invaluable guidance, comments and suggestions throughout the course of the project. Moreover, his lecture particularly provided us a great deal about technical writing as an engineer which assisted us a lot to complete our final report.

We are also grateful to the technicians of the laboratory of the Vietnamese-German University department for their help in offering us the materials during the planning and development of the system.

This report also could not be completed without the help of our friends and classmates who have willingly helped us out with their abilities.

During this project, tasks were divided up as follows:

Phuong Hong Nguyen was a project leader, wrote abstract, drew block diagram and Gantt chart, helped write the reports and checked the report and PowerPoint.

Ngo Phuc Linh was a main report writer, structured this report, wrote the description, mechanical system, the controlling system, conclusion and glossary of terms.

Nguyen Dang Khoa was a main programmer, wrote the software, drew the flowchart and algorithms. He also checked the report and PowerPoint for the oral presentation.

Dang Nhat Quang was the speaker of presentations, wrote letter of transmittal, acknowledgements, introduction, background, requirements & constraints, checked the format of the report and PowerPoint.

Team 1

Table of Contents

Table of Figures

1. Introduction

1.1. Purpose of the Report

This report was created to fulfill the requirements of the course “English for Computer Science 4” at the Vietnamese-German University, as specified on 22 April 2019. The paper deals with the design of a plant monitoring system including a soil moisture sensor, Raspberry Pi 3 Model B+, and an Arduino microcontroller to receive and determine the data and then operate correspondingly the pump in order to provide water to the pot. The user interface website is created for the users to easily interact with the system via Internet. Every equipment, codes and the website are shown in this report.

1.2. Audience

The intended audience of this document is the course instructor, who will use it as the basis for the determination of a portion of our grade for the class “English for Computer Science 4”.

1.3. Scope of the Report

This report covers a section on the design stage and programming of the Plant Monitoring System on the small scale in the embedded microcontroller project, indicates an overall implementation, gives a detailed of hardware, software description and components.

2. Context of the Project

2.1. Requirement and Constraints

There are some requirements that we must meet during the project:

- Teams must come up with an idea for a project that uses a Raspberry Pi board as the main processor, the project also needs to include a graphic interface, either web based or on screen.
- Teams are free to use any programming language or graphic interface tools.
- Our project must comply with the reporting standards provided, including appropriate references for any ideas, events or images borrowed [1].

3. Overview of the Plant Monitoring System

3.1. Description

The project is to design and develop a plant monitoring system. This project will mainly focus on processing data collected from the system as well as publishing data to a website for users to see the status of the gardens. The key features of this system are sensors which sense the environment in order to control the growth of the plant, and an automatic watering system will lower the effort as well as reduce the time needed for taking care of the plant. This system, therefore, consists of a soil moisture sensor and a submersible water pump. The soil moisture sensor is used for measuring the moisture of the soil to determine whether to trigger the submersible pump to water the plant or not. The moisture sensor sends soil moisture statistics to Raspberry Pi for users to monitor their plants. The LED signal is used for notifying the status of the submersible pump which indicates the status of the pump “On” or “Off”. The users can control the Plant Monitoring System on the website.

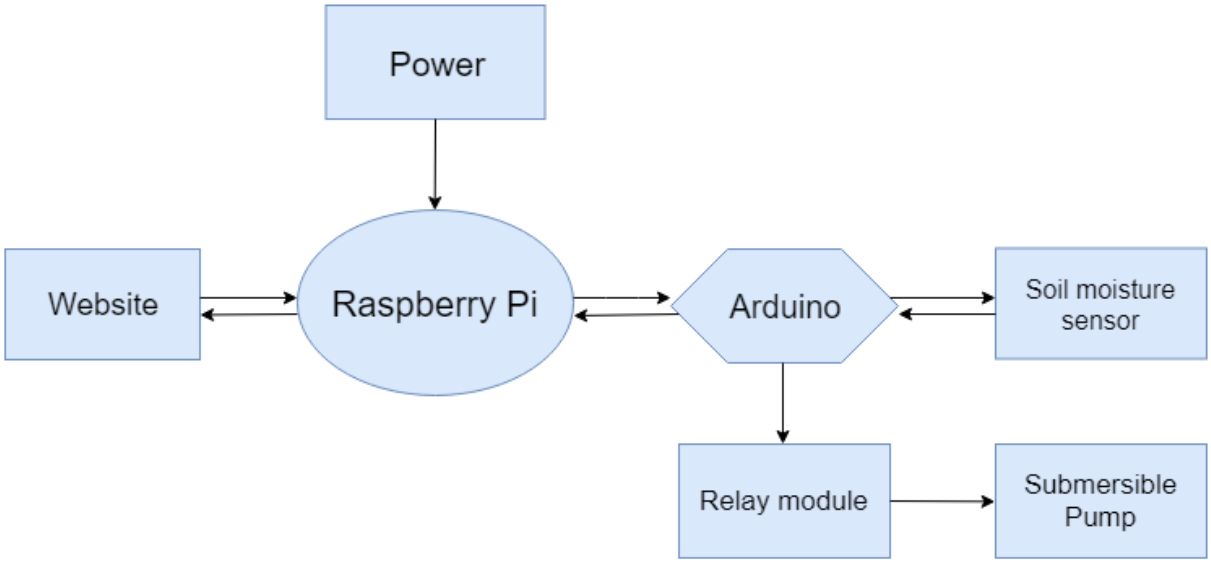


Figure 1: Block diagram for the system’s inputs and outputs

3.2. Mechanical system

The mechanical system of the plant monitoring system including four main parts including a submersible pump, a relay module, a soil moisture sensor and a LED indicator. For the purpose of controlling soil moisture, submersible pump triggered by relay module is utilized to provide adequate amount of water for the plants to prevent it from waterlogged. All the parts are securely connected to the system with wires.

3.2.1 Soil moisture sensor

The moisture sensor is used to measure the moisture of soil. When the soil is having water shortage, the module output is at high level, else the output is at low level. This sensor then sends signal to Raspberry Pi through Arduino's digital pin to adjust the water level of the plants as well as publishing data to the website [2].

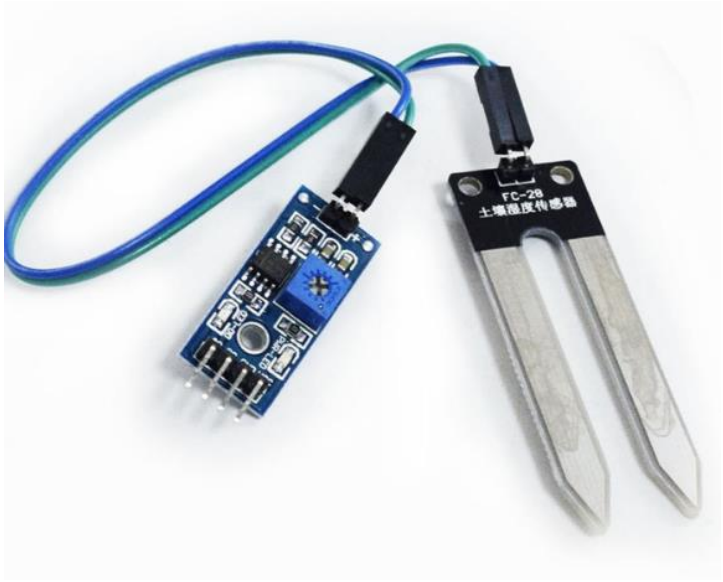


Figure 2: Soil moisture sensor

3.2.2 Relay module

Relay is an electrically operated switch which is composed of an electromagnet, an armature, a spring and a set of electrical contacts. The electromagnetic switch is useful to separate two different circuits or completing the circuit. They play an important role in the project since it allows one circuit to control another while they are completely isolated. Thus, this help Arduino microcontroller working at a low voltage to interface with the higher voltage submersible pump [3].

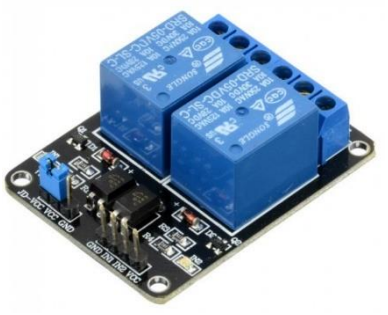


Figure 2: Relay module

3.2.3 Submersible Pump

Submersible pump is a type of water pump which is hermetically sealed operating by pushing water through the pipe during the pumping process. In contrast with others kind of pump, submersible pump pushes fluid to the pipe rather than pulling the fluids which results in more efficient during pumping process. Therefore, the submersible water pump along with relay module is used to supply water to the plants when needed [4].

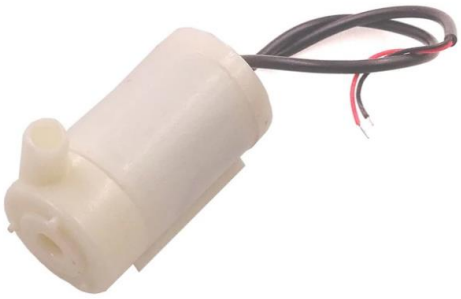


Figure 3: Submersible Pump

3.2.4. Communications interfaces

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. The LED signal is used to report the status of the submersible pump weather it is “On” or “Off” [5].

Figure 5: LED

3.3 Controlling system

The controlling system consists of a Raspberry Pi Model 3B+ to communicate with an Arduino microcontroller board to send signal to sensors.

3.3.1. Raspberry Pi Model 3B+

The heart of the project is Raspberry Pi Model 3B+ which was launched in 2018. This Model 3B+ is a microcomputer, compact size with capable of running several different operating systems including Linux-based OSes, Window 10... especially Raspbian which executes Python programming language and hosts the user interface website. It has 1GB RAM, 1.2GHz quad-core CPU, built-in wireless and Bluetooth connection, multiple USB ports, a micro-SD card reader, Ethernet, HDMI, audio outputs and a video camera connection. Moreover, it also has GPIO that allow you to control electronic components, e.g., buttons or LEDs [6].

Figure 6: Raspberry Pi 3 Model B+

3.3.2. Microcontroller board

The main controller in the robot is Arduino Uno with Third Revised edition. Arduino UNO R3 is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (6 PWM output pins), 6 analog input pins, a 16 MHz ceramic resonator, a USB type-B connection, a power jack, an ICSP header, and a reset button. The power supply can be provided via USB type-B or through DC female jack [7].

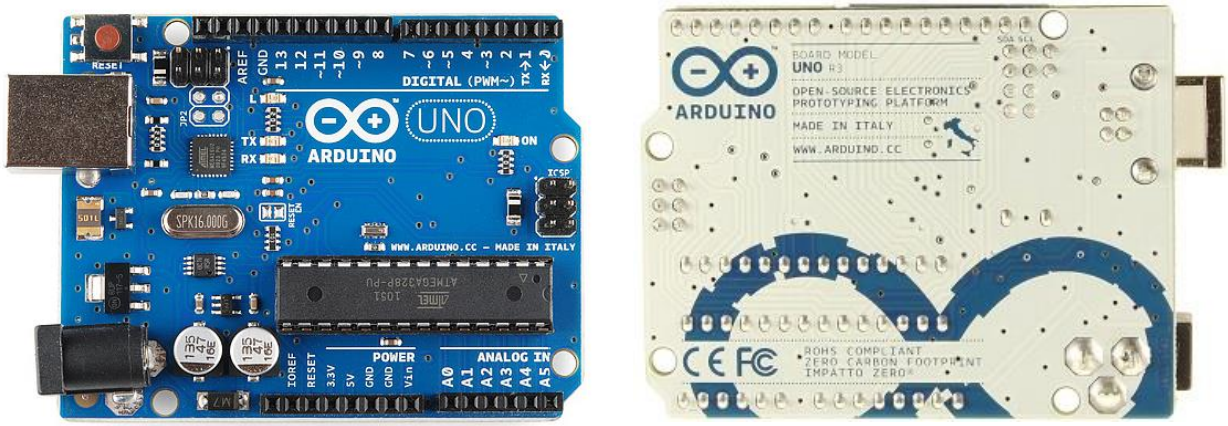


Figure 7: Front view and back view of Arduino

The Arduino is an open-source electronics platform and can be programmed with the Arduino Software (IDE). Arduino programming language is based on C/C++ language which is supported by a huge community and various type of library.

4. Software

4.1. Overall design

The plant monitoring system software is mainly programmed by four programming languages C/C++ (on Arduino), Python, HTML and CSS. All the process is implemented in Python by using "pySerial" library in order to send instructions to Arduino microcontroller to control the water pump and receive analog signals from soil moisture sensors. "Flask" – a web development library for Python combines with HTML and CSS to create a user interface website which provides users a controlling panel to easily control their plants. To summarize, Arduino is programmed as a slave device to receive and transmit signals to Raspberry Pi which allows user to interact with their plants.

The Plant Monitoring System software consists of the following elements:

- Automatic mode
- Manual mode
- Website module

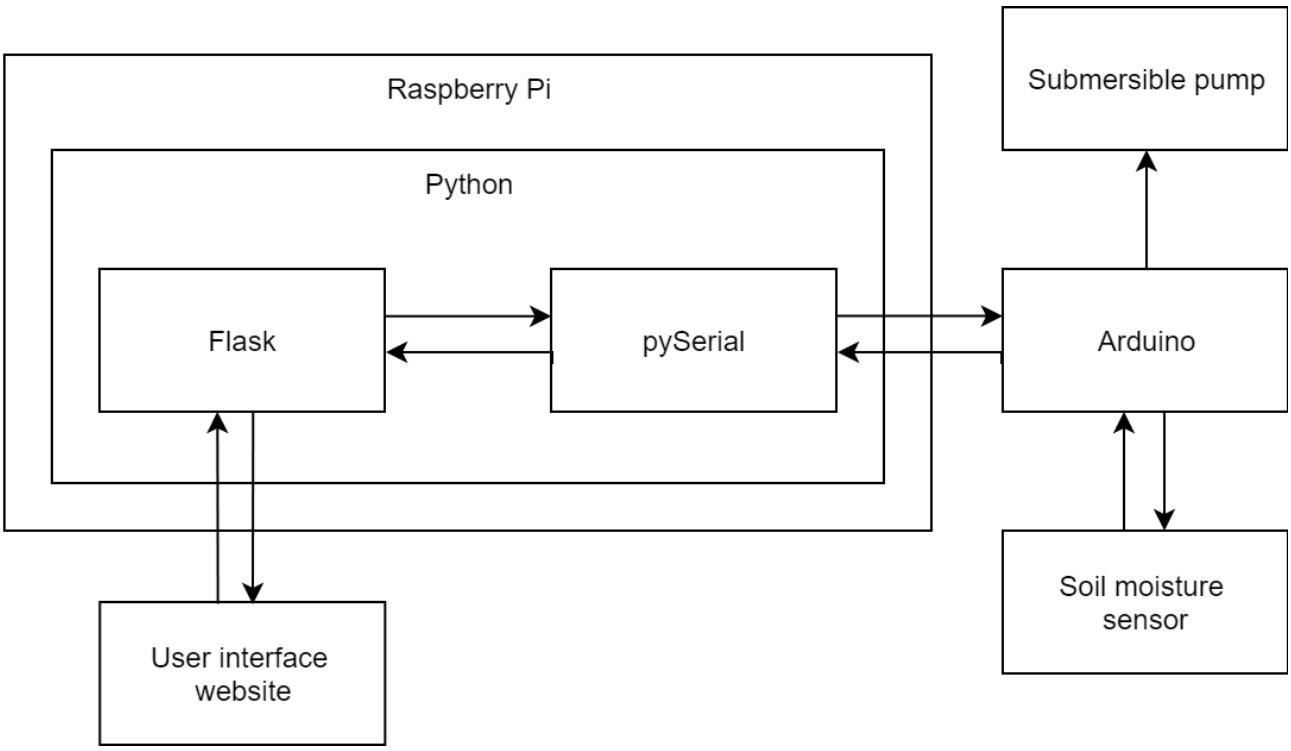


Figure 8: Software Functional Design

4.2. Automatic mode

In automatic mode, users are provided two options weather to triggers the automatic watering mode or not. Monitoring system will automatically check the soil moisture and water the plant if needed.

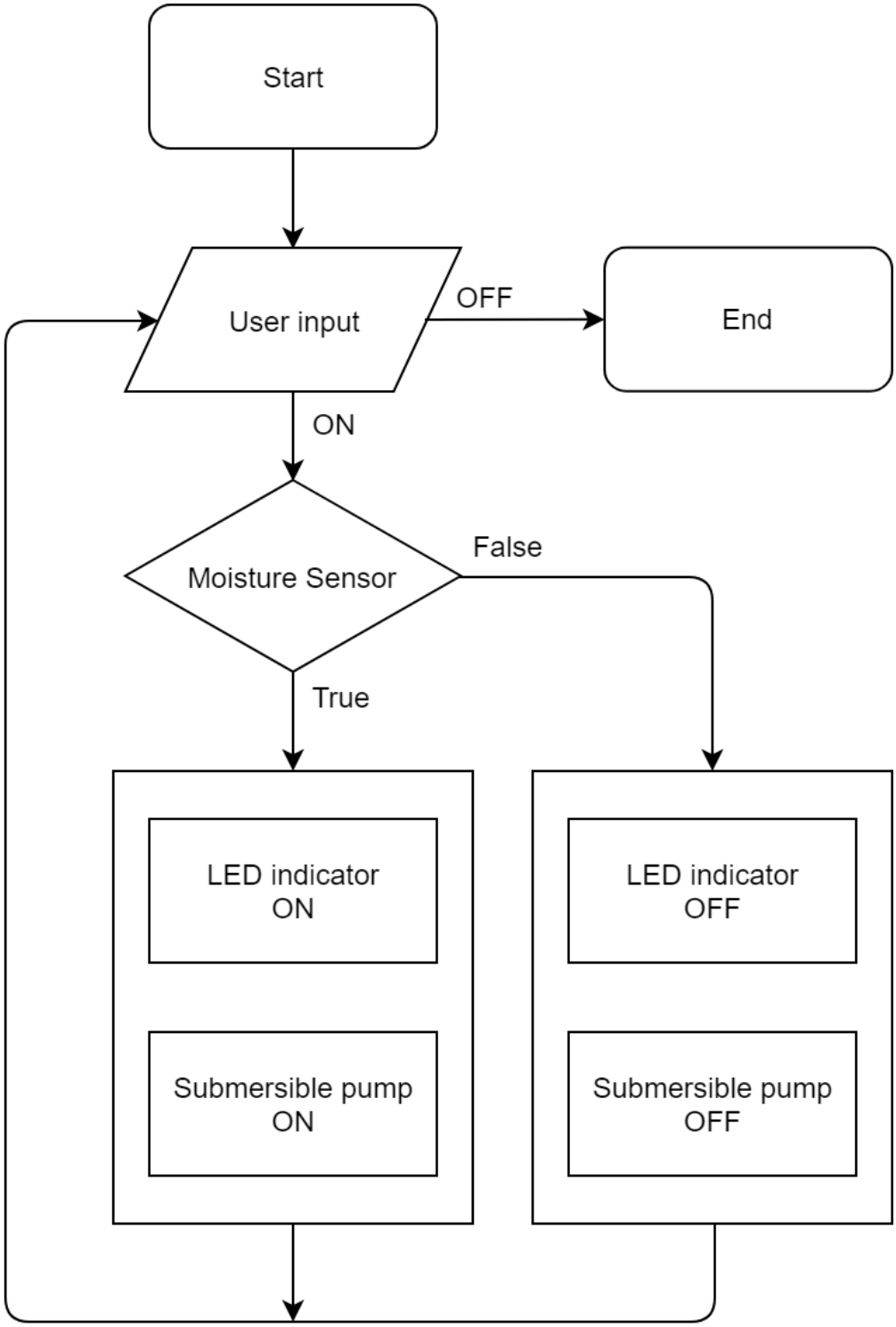


Figure 9: The functional design of automatic mode

4.3. Manual mode

In manual mode, three options are provided to users. Firstly, via user interface website, users can easily check last time watered or checking the soil moisture of the plants in the manual panel. Furthermore, users can control the water pump in this mode as well.

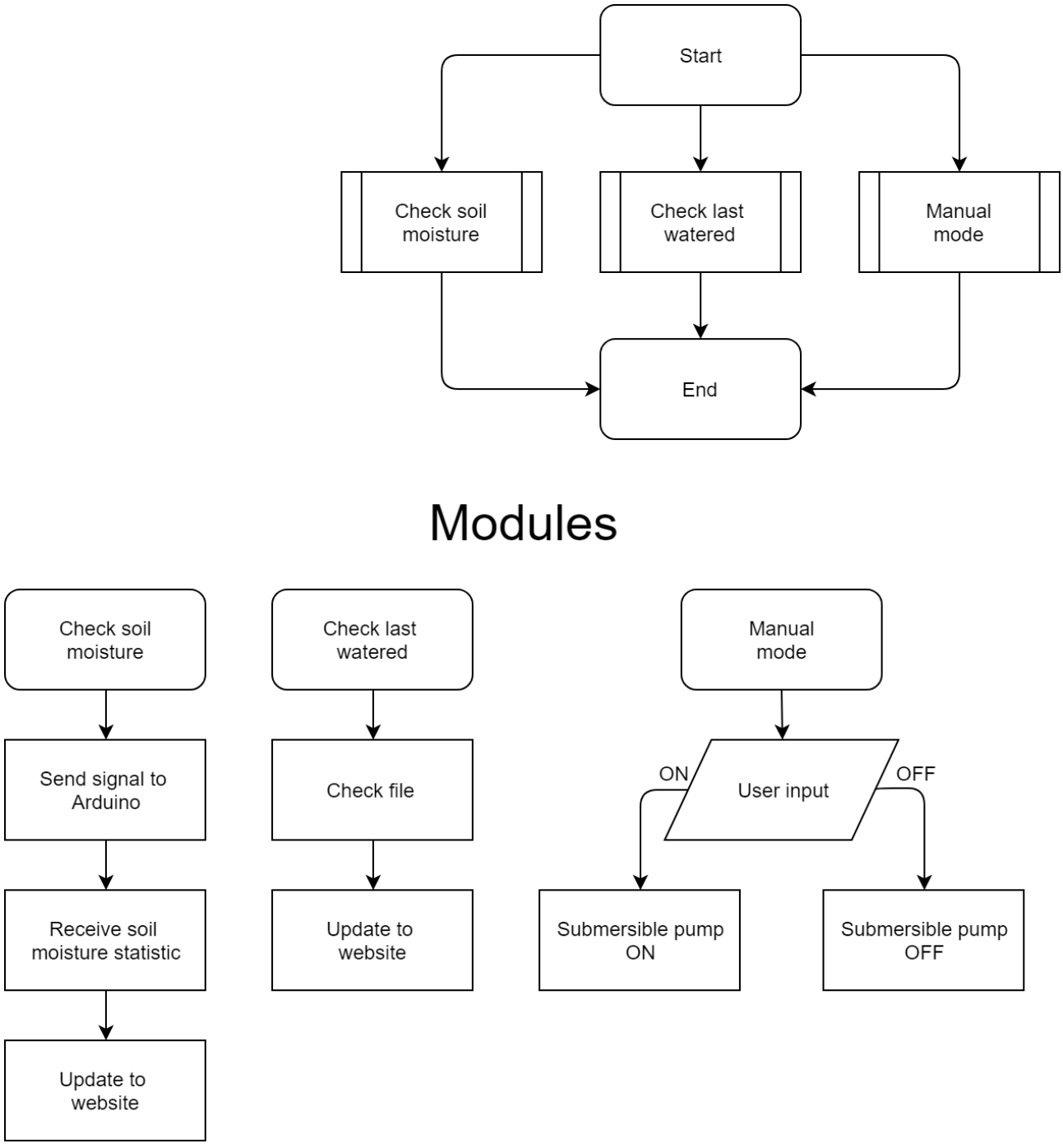


Figure 10: The functional design of manual mode

4.4. Website module

User interface website is developed and designed for user to easily monitoring and managing their plants. Website is programmed by HTML and CSS as front-end and Flask as back-end of the website.

User interface website is divided into four section including:

- System's time
- Memo
- Auto mode
- Manual mode

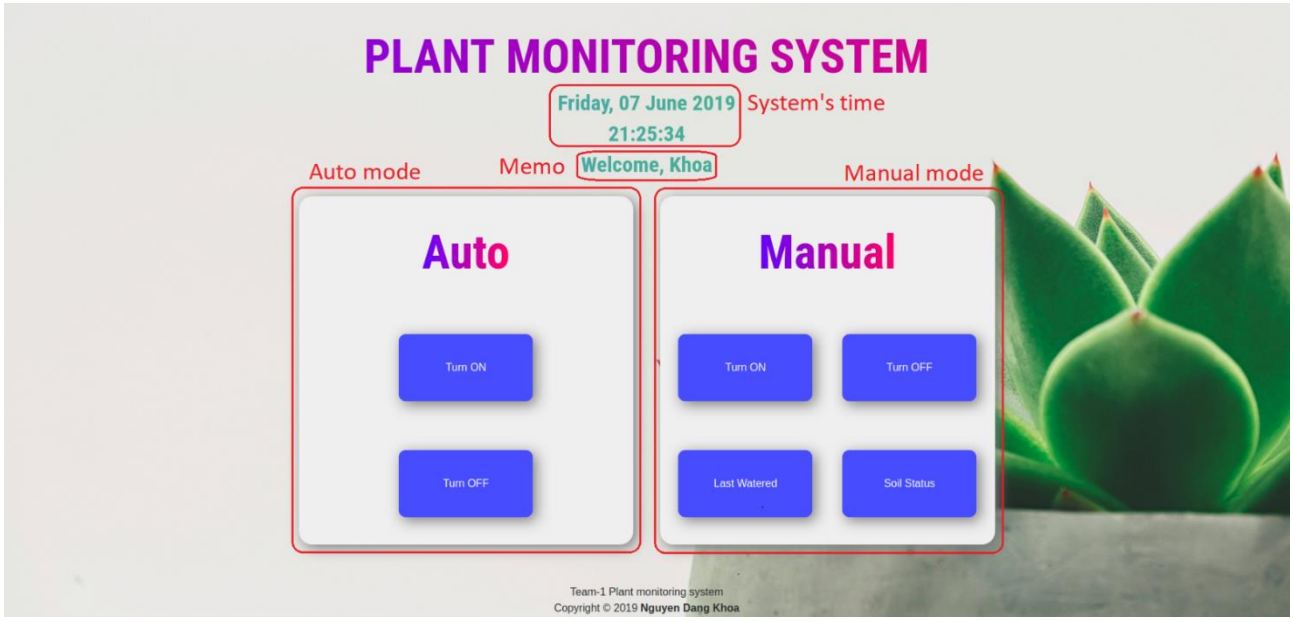


Figure 11: The website module

6. Conclusion

To sum up, the project works as planned, achieved the given requirements which involves create a plant monitoring system and a website for the user to monitor and interact with the system by using four main programing languages which are C/C++ programming language (Arduino), Python, HTML and CSS. The system can automatically check the moisture of the soil and water the plant to ensure it won't die or overwatering.

7. Reference

[1] R. G Bradley, “Specification for Raspberry Pi Project”, Vietnamese-German University, 14-Apr-2019.

[2] Wikipedia, Soil moisture sensor, [Online], Available: https://en.wikipedia.org/wiki/Soil_moisture_sensor/

[3] Core Electronics, Retrieved from 5V 2 Channel Relay Module 10A, [Online], Available:https://core-electronics.com.au/5v-2-channel-relay-module-10a.html/

[4] Amazon, Submersible DC 2.5-6V Low Noise Brushless Mini Water Motor Pump, [Online], Available: https://www.amazon.in/OnlineSRS-SUBMERSIBLE-2-5-6V-Noise-Brushless/dp/B07GJM2JY6/

[5] Wikipedia, Light-emitting diode, [Online], Available: https://en.wikipedia.org/wiki/Light-emitting_diode/

[6] Raspberry Pi Organization, RASPBERRY PI SPECS AND BENCHMARKS: 3A+, 3B+, ZERO W, [Online], Available: Available: https://www.raspberrypi.org/magpi/raspberry-pi-specs-benchmarks/

[7] Hshop, Arduino Uno R3, [Online], Available: https://hshop.vn/products/arduino-uno-r3/

Appendix A: Glossary of Terms

Arduino -	An opensource computer hardware and software project.
Clockspeed -	Measured in Hertz, used to measure a processor's speed.
CS -	Computer Science.
LED -	Light-Emitting Diode.
Microcontroller -	A small computer on single integrated circuit.
VGU -	Vietnamese – German University.
HTML -	Hypertext Markup Language.
CSS -	Cascading Style Sheets.
PySerial -	A module encapsulates the access for the serial port.
Flask -	A microframework for Python based on Werkzeug, Jinja 2 and good intentions.
Raspberry Pi -	A tiny and affordable computer that you can use to learn programming through fun, practical projects.

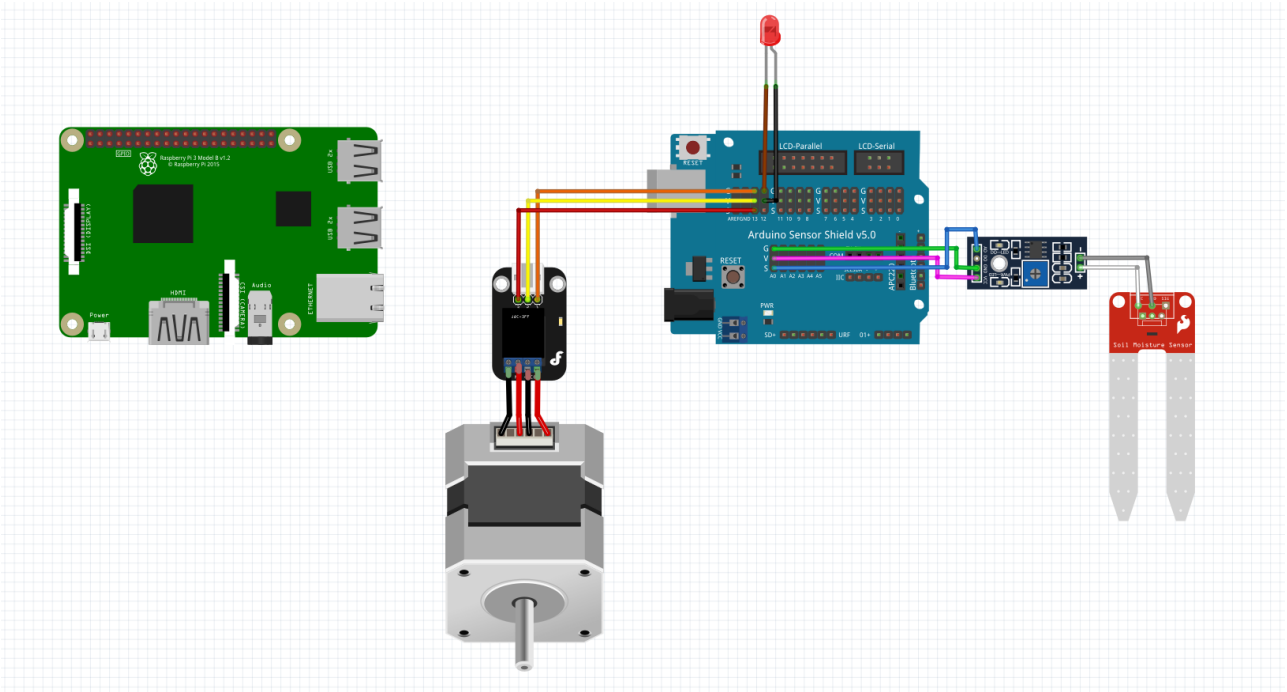
Appendix B: Grantt Chart

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7
Analyze constraints and requirements							
Research on exitsting designs							
Learn programming languages needed							
Test and work on the equipments							
Plant soybeans							
Complete the design of the system							
Connect all the hardware							
Optimize the code and design							
Complete a foundation for our website by using HTML and JS (JavaScript)							
Test out all the equipment and correct any errors							
Write report							

Deliverables	
Proposal	Week 3
Progress Report 1	Week 4
Progress Report 2	Week 6
Final report	Week 7

Legend	
Critical Path	
Non-Critical	

Appendix C: Circuit diagram



Appendix D: Program Listing

```
//main
import serial                   #Khai bao thu vien Serial
from time import sleep          #Khai bao lenh sleep tu thu vien time

ser = serial.Serial('/dev/ttyACM0',9600)
def sendAndReceive(reqData):
        while True:
                if (ser.in_waiting>0):
                        resData = str(ser.readline().decode('ASCII'))
#Get & decode data received
                        return resData
                else:
                        reqData += "\r"
                        ser.write(reqData.encode())
                        sleep(0.5)



//water

# External module import
import datetime
import time
import main

init = False

def get_last_watered():
    try:
        f = open("storage/last_watered.txt", "r")
        return f.readline()
    except:
        return "NEVER!"

def auto_water(delay = 2):
    print("Press CTRL+C to exit")
    try:
        while True:
            print("Entered the loop")
            time.sleep(delay)
            main.sendAndReceive('Led Off')
            main.sendAndReceive('Pump Off')
            storeLastWatered()
            dry = int(main.sendAndReceive('Moisture')) == 0 #True if dry
            print(dry)
            if dry:
                pump_on()
    except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
        main.sendAndReceive('Stop')

def pump_on(delay = 1):
    main.sendAndReceive('Led On')
    main.sendAndReceive('Pump On')
    storeLastWatered()
```

```
    time.sleep(delay)

def storeLastWatered():
    f = open("storage/last_watered.txt", "w")
    now = datetime.datetime.now()
    dateString = now.strftime("%A, %d %B %Y %X")
    f.write("Last watered: {}".format(dateString))
    f.close()



//auto_water

# External module import
import datetime
import time
import main

init = False

def get_last_watered():
    try:
        f = open("storage/last_watered.txt", "r")
        return f.readline()
    except:
        return "NEVER!"

def auto_water(delay = 2):
    print("Press CTRL+C to exit")
    try:
        while True:
            print("Entered the loop")
            time.sleep(delay)
            main.sendAndReceive('Led Off')
            main.sendAndReceive('Pump Off')
            storeLastWatered()
            dry = int(main.sendAndReceive('Moisture')) == 0 #True if dry
            print(dry)
            if dry:
                pump_on()
    except KeyboardInterrupt: # If CTRL+C is pressed, exit cleanly:
        main.sendAndReceive('Stop')

def pump_on(delay = 1):
    main.sendAndReceive('Led On')
    main.sendAndReceive('Pump On')
    storeLastWatered()
    time.sleep(delay)

def storeLastWatered():
    f = open("storage/last_watered.txt", "w")
    now = datetime.datetime.now()
    dateString = now.strftime("%A, %d %B %Y %X")
    f.write("Last watered: {}".format(dateString))
    f.close()
```

```
//website

from flask import Flask, render_template, redirect, url_for
import psutil
import datetime
import water
import main
import os

app = Flask(__name__)

def template(title = "Plant monitoring system", text = ""):
    now = datetime.datetime.now()
    dateString = now.strftime("%A, %d %B %Y")
    timeString = now.strftime("%X")
    templateDate = {
        'title' : title,
        'time' : timeString,
        'date' : dateString,
        'text' : text
        }
    return templateDate

@app.route("/")
def hello():
    templateData = template(text = "Welcome, Khoa")
    return render_template('main.html', **templateData)

@app.route("/last_watered")
def check_last_watered():
    templateData = template(text = water.get_last_watered())
    return render_template('main.html', **templateData)

@app.route("/sensor")
def action():
    status = int(main.sendAndReceive('Moisture'))
    message = ""
    if (status == 0):
        message = "Water me please!"
    else:
        message = "I'm a happy plant"

    templateData = template(text = message)
    return render_template('main.html', **templateData)

@app.route("/wateron")
def action2():
    water.pump_on()
    print('Pump on')
    templateData = template(text = "Watering...")
    return render_template('main.html', **templateData)

@app.route("/wateroff")
def action3():
    main.sendAndReceive("Pump Off")
    main.sendAndReceive("Led Off")
    print('Pump off')
    templateData = template(text = "Turned OFF")
```

```
    return render_template('main.html', **templateData)
@app.route("/auto/water/<toggle>")
def auto_water(toggle):
    running = False
    if toggle == "ON":
        templateData = template(text = "Auto Watering ON")
        for process in psutil.process_iter():
            try:
                if process.cmdline()[1] == 'auto_water.py':
                    templateData = template(text = "Already running")
                    running = True
            except:
                pass
        if not running:
            os.system("python auto_water.py")
    else:
        templateData = template(text = "Auto Watering OFF")
        os.system("pkill -f water.py")
        main.sendAndReceive("Pump Off")

    return render_template('main.html', **templateData)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=8000, debug=True)



//arduino

const int led= 12;

const int pump = 13;

const int moiSensor = A0;

String buff;

void setup()

{

  Serial.begin(9600);   //Bật cổng Serial Baudrate 9600

  pinMode(led, OUTPUT);  //Khai báo chân OUTPUT

  pinMode(pump, OUTPUT);

  pinMode(moiSensor, INPUT);

}

void loop()

{

  if (Serial.available()) //Nếu có tín hiệu từ Pi

  {

    buff = Serial.readStringUntil('\r'); //Đọc vào đến khi gặp \r (xuống
dòng)
```

```
    if (buff=="Led On")             //Nếu dữ liệu = "Led On"
    {
      digitalWrite(led,HIGH);       //Bật HIGH chân led
      Serial.println("Turned On");  //Trả ngược về "Turned On"
    } else

    if (buff=="Led Off")            //Nếu dữ liệu = "Led Off"
    {
      digitalWrite(led,LOW);        //Bật LOW chân led
      Serial.println("Turned Off"); //Trả ngược về "Turned Off"
    }

    if (buff=="Moisture")              //Nếu dữ liệu = "Hello"
    {
      Serial.println(!digitalRead(moiSensor));         //Trả ngược về
"Hi"
    }
    if (buff=="Pump On") {
      digitalWrite(pump, LOW);
      Serial.println("Turned On");
    }
    if (buff=="Pump Off") {
      digitalWrite(pump, HIGH);
      Serial.println("Turned Off");
    }

    if (buff=="Stop")
    {
      digitalWrite(led, LOW);
      digitalWrite(moiSensor, LOW);
    }
  }
}
```