Plant Selector

Lutfi Berker Evliyaoglu
Department of Computer Science
Binghamton University
New York/ USA
levliya1@binghamton.edu

ABSTRACT

Global warming has begun to affect and change the entire world and agriculture. Traditional cultivation is no longer efficient enough. We need to restore harmony, which is one of the greatest abilities of humanity in its existence. Thanks to Plant Selector, we can come more efficiently and easily in agriculture. By choosing the most suitable plant for changing conditions, we can show our survival knowledge on plants. Plant Selector chooses the suitable plants for us by looking at the type of soil, the amount of light, the rainfall amount, the temperature and humidity of the air and the moisture of the soil.

Using 4 different sensors and Raspberry Pi 4, I choose the most suitable plant for the area to be cultivated. Since this project is a long-term project, time will be required to observe the device and sensors. In order to find the right plants, he has to stay and work in that area for a long time. The longer it is left, the more accurate the result will be. Then the results will appear when the user manually terminates the program.

KEYWORDS

Global warming, harmony, IoT sensors, web scrapping, raindrop sensor, UV light sensor, DHT11 sensor, soil moisture sensor.

1 Introduction

Internet of things is getting more and more popular as the day goes on. With the help of sensors, we can use the internet of things everywhere, and agriculture is one of them. Agriculture will continue to exist as long as people exist. However, people inadvertently destroy agriculture in their own way. Climate change and global warming make agriculture more difficult. Sand storms, sudden freezing of plants, drastic changes in air temperature, changing humidity levels with newly built dams seriously affect the growth and productivity of plants. So, we will have a self-growing program by expanding our database with incoming data and feedback.

With the UV light sensor, we can see how much light the agriculture region is exposed to, and how moist the soil is with the soil moist sensor. We must also measure air temperature and air humidity, which is one of the most important factors. Since the weather conditions do not give enough specific information, it will not be possible for us to know the exact temperature of that region. The air temperature, which varies according to the incoming light and the wind, can vary throughout the agricultural area, so it becomes more important to find the exact temperature and humidity of the agricultural area. Finally, we can see the amount of rainfall in that area by web scrapping to find the amount. I also use sensors for this event. Instead of pulling in continuous information, we prevent the program from slowing down by pulling only after the rain had finished. By comparing this data with our database, we show the most efficient crops for the agricultural field to the user.

2 Design

2.1 Software

2.1.1 Raspbian

I used Raspberry Pi 4 for this project and its operating system is Raspbian. It is an operating system that looks between windows and Linux, but is easier to use than Linux and faster to integrate into sensors

2.1.2 Python

I used python as a programming language because it offers so much library for sensors and Raspberry Pi. I have developed a program faster using Python.

2.1.3 BeautifulSoup 4

It's a web scrapping library in Python. After the rain is over, I use this library to get the necessary information from the meteorology site.

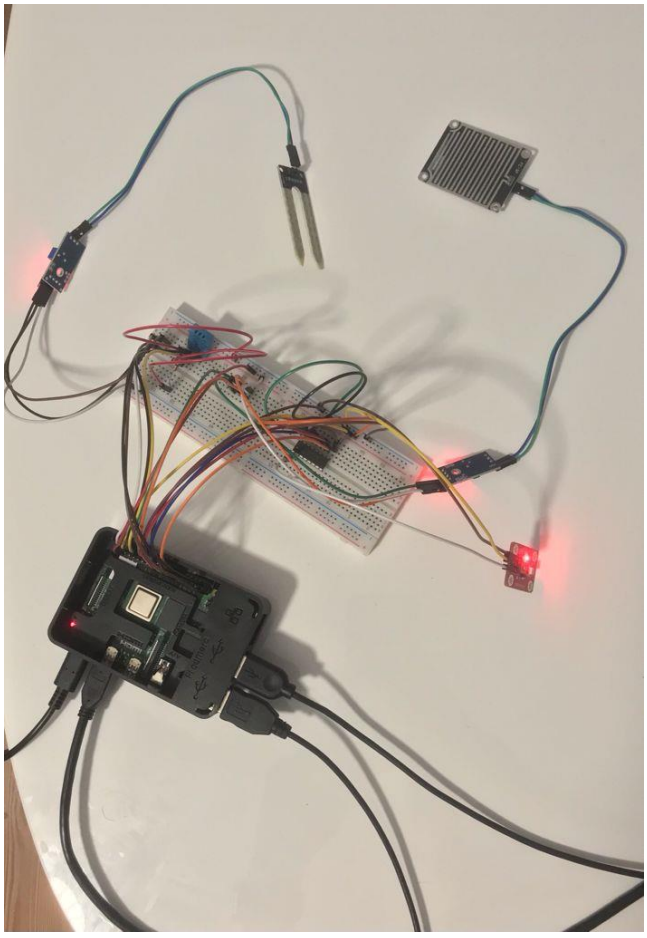
2.1.4 RPi.GPIO

RPi.GPIO is a Raspberry Pi library in Python to connect and use sensor.

2.1.5 Adafruit_Blinka[1]

A new library for adafruit blinka mcp3008 ADC (Analog to Digital Converter). Since Raspberry Pi does not use analog data, an ADC is required. We can run the mcp3008 by using Adafruit_blinka.

2.2 Hardware



2.2.1 Raspberry Pi 4

Raspberry Pi 4 [2] , has 1.5GHz 64-bit quad core ARM Cortex-A72 CPU, 1GB RAM, on-board wireless LAN, Bluetooth, Ethernet, USB ports and 40- pin GPIO header. So, we can use the sensor to connect it.

2.2.2 DHT11 Temperature and Humidity Sensor

The sensor comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. The sensor is also factory calibrated and hence easy to interface with other microcontrollers. The sensor can measure temperature from 0°C to 50°C and humidity from 20% to 90% with an accuracy of ±1°C and ±1%.[3]

2.2.3 GUVA-S12SSD UV Light Sensor

Light-measuring sensor. Cannot be used with Raspberry Pi without ADC as it uses analog data.

2.2.4 MCP3008 ADC

MCP3008 I / P Analog Digital Converter Integration is in DIP-16 case. MCP3008-I / P works with 2.7 - 5.5V supply voltage. This Analog Digital Converter Integration has Differential, Single Ended input type and Serial has SPI output. It also has a sampling rate of 200kSPS at 10 Bit resolution.[4]

2.2.5 Soil Moisture Sensor

It is for detect the soil is moisty or not. It is also using analog data so I used Control Board to convert to digital data.

2.2.6 Rain Drop Sensor

I used this sensor to detect rain is start or stop. In this way I can pull the rainfall amount data from meteorology website. It is also using analog data so I used control board to convert to digital data.

3 Implementation

I have completely used the python language in the code. Sensors are connected to Raspberry Pi 4. I did the sensor usage and calculations completely in raspberry pi, so the response time of the program was faster. After connecting the UV light sensor using mcp3008, I used adafruit_blinka library to turn it on. I used RPi.GPIO library for DHT11 temperature and humidity sensor, Rain drop sensor, and soil moisture sensor. I only use the soil type to compare with the database because there is no sensor available to detect the soil type. The purpose of using the rain drop sensor is that the program works faster and I take the average of the temperature for the DHT11 temperature and humidity sensor and compare it with the minimum and maximum value in the database. When the program is stopped by the user, it lists the most suitable plants for the measurements.

ACKNOWLEDGMENTS

Plant Selector works as described above. The program is designed to be usable in a region and for long-term operation. Since the program does not have an end, it must be measured continuously. If the user wants to finish the measurement, the user can easily see the results by doing CTRL + C. Since the program will be run in an open area, energy support must be required and the router must have a wide range or Personal Hotspot to be connected to the Internet.

REFERENCES

[1] https://pypi.org/project/Adafruit-Blinka/

[2] https://www.makeuseof.com/tag/raspberry-pi-4-overview/DOI: https://doi.org/10.1145/1188913.1188915.

[3] https://components101.com/sensors/dht11-temperature-sensor

[4] https://www.direnc.net/mcp3008-ip-dip16-analog-dijital-cevirici-entegresi