COMP4121 Fall 2021

Project Title: Soil Moisture Sensor for Plants

Team Members

Mert Atakan Duduoğlu, 214SE2295, mert.duduoglu@isik.edu.tr

Can Peksöz, 216SE2316, can.peksoz@isik.edu.tr

1 Problem Statement

Irrigation is a constant challenge in the agriculture industry. Under-watered crops can wither and die, wasting all the time, money and effort that went into growing them. Over-watering fields can lead to increased soil erosion.

2 Existing Solutions

We will be able to monitor the state of the soil by checking the data we have obtained via the soil moisture sensor for portable systems with deep-sleep function.

3 The Proposed System

The sensor will send data via Wi-Fi to the cloud database. We will monitor the data on mobile and laptop.

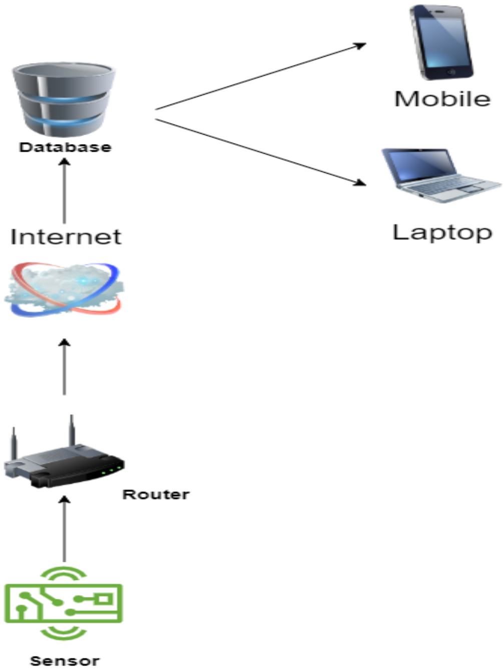
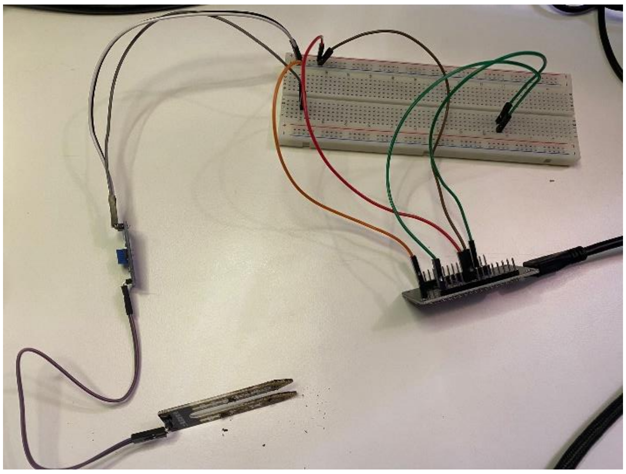


Figure 1: The proposed system overview

4 Hardware Specifications

- The Soil Moisture Sensor
- Node MCU ESP 8266
- 400 POINT BREADBOARD
- smart phone
- Wi-Fi Connection
- USB cable
- laptop
- jumper cables



COMP4121 Fall 2021

Project Title: Soil Moisture Sensor for Plants

Team Members

Mert Atakan Duduoğlu, 214SE2295, mert.duduoglu@isik.edu.tr

Can Peksöz, 216SE2316, can.peksoz@isik.edu.tr

5 Software Specifications

- Arduino IDE

- Google Realtime Database

- Android Studio

5.1 Use Cases

Use Case 1 :

Use case name: Soil is dry

Actors: User, Sensor

Flow: The user forgets the water the plant and the sensor measure the soil.

Entry Condition: The user saw that she/he needed to water the plant.

Exit Condition: The user watered the plant.

Use Case 2 :

Use case name: Soil is watered

Actors: User, Sensor

Flow: The user waters the plant and the sensor measures the soil.

Entry Condition: The plant must be watered.

Exit Condition: The user can monitor the plant doesn't need water.

Use Case 3 :

Use case name: Soil is dry and Button on app is visible

Actors: User, Sensor, Android App

Flow: User presses the button after watering.

Entry Condition: The button must be visible.

Exit Condition: The user click the button and data on cloud changed like a watered and button is invisible.

5.2 Functional Requirements

- The sensor can sense the moisture of the soil.

- The monitoring system can show alarm if water level is very low.

- The monitoring system can show alarm if plant is watered.

5.3 Non-functional Requirements

Usability:

-The monitoring should be easy for users.

-Users with poor sensor skills should understand the situation of the soil.

COMP4121 Fall 2021

Project Title: Soil Moisture Sensor for Plants

Team Members

Mert Atakan Duduoğlu, 214SE2295, mert.duduoglu@isik.edu.tr

Can Peksöz, 216SE2316, can.peksoz@isik.edu.tr

Availability:

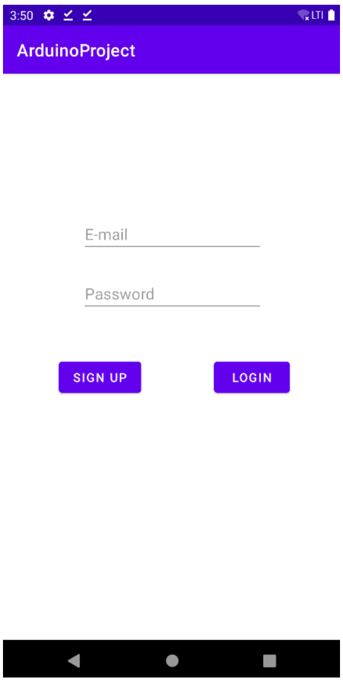
-Sensor and database should be available from any network and available all the time.

Performance:

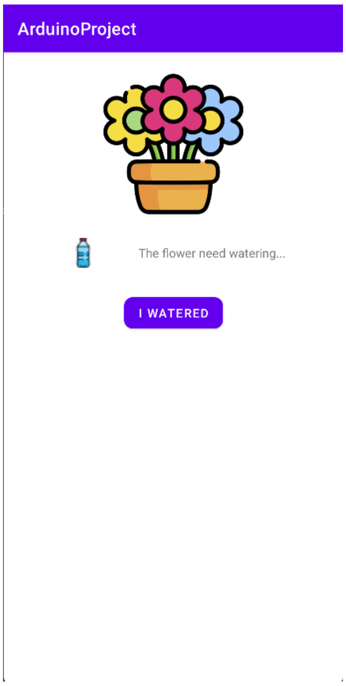
-The database should be reachable for any time for the sensor.

-The database must secure all data.

5.4 User Interface

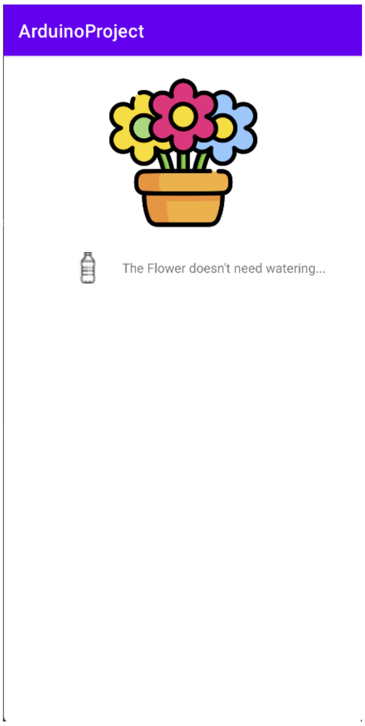


1.Login and Sign-up Page



The flower need watering...

1.2 Screen for if plant need water



The Flower doesn't need watering...

1.3 Screen for if plant doesn’t need water.

6 Test-bed Details

-Checking the connectivity

-Checking the required software

-Configuration of the sensor

7 Performance Tests & Results

Since the system should be portable, it will work on battery and the Wi-Fi connection will consume a lot of energy, so the system should work in deep-sleep and wake up only when it sends a signal.

System sends data in every 1 hour. The power consumption takes 5 seconds max in every cycle. When everything is powered on the board uses about 80mA. But in deep-sleep the current is 0.25mA. One day’s energy consumption calculation is approximately like this;

- In a day, the value is reported every hour.

(24times*5s*80mA/3600)

- The consumption in deep-sleep every day.

(24h*0.25mA)

If we use Duracell CopperTop AA Alkaline Battery(2850mAh)

2850/((24*5*80/3600)+(24*0.25))

=328.84 days

With those measurements we can see that system can run almost one year without changing battery.

8 Conclusion

We learned using ESP8266 with cloud database and using the system in deep-sleep mode for energy save.

We need to plug d0 and rst together for the awake the system. When uploading the code to ESP8266. GPI0(d0) and reset(RST) must be unplugged for the avoiding crash.

We saw that in ESP chip, deep-sleep function is unreliable and ESP8266 can’t sleep more than 71 minutes.

With the increasing number of IoT connections in the agriculture system, it can cooperate using soil moisture sensors together with energy-saving

COMP4121 Fall 2021

Project Title: Soil Moisture Sensor for Plants

Team Members

Mert Atakan Duduoğlu, 214SE2295, mert.duduoglu@isik.edu.tr

Can Peksöz, 216SE2316, can.peksoz@isik.edu.tr

References

[1]https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.680.1864&rep=rep1&type=pdf

[2]https://www.delmhorst.com/blog/whats-the-ideal-moisture-level-for-soil-to-grow-crops

[3]http://gardenbot.org/howTo/soilMoisture/

[4]https://panther.kapsi.fi/posts/2019-05-14_esp8266_deep_sleep