Automated Irrigation system

Team:

1. Abhinay Cherupally (21311A6624)
2. Prem Kiran Laknaboina (21311A6611)
3. Vishnu Vardhan Reddy (21311A6607)

Introduction

A System specially designed in order to minimize the water usage and increase the efficiency of crop growth in a well-maintained environment.

Our project is an automated drip irrigation system that helps farmers irrigate their fields with ease. It calculates the required moisture level for crops and turns on the water pump when the moisture level is not adequate. It works in fields with a power source and notifies the user if there is a power cut during watering or if the moisture levels drop below the specified level.

Problem:

Farmers often face the challenge of ensuring their crops receive adequate water, which is crucial for their growth and yield. However, manually irrigating crops is a time-consuming and labor-intensive process. In addition, traditional irrigation methods can lead to over or under watering, which can harm the crops.

Solution:

Automated Drip Irrigation system is developed for the benefit farmers so that irrigation becomes a job at hand for them.When the moisture level is not in the required level, then the water pump turns on and water is supplied to the soil. The amount of moisture required by a crop is calculated.When the moisture level is not in the required level, then the water pump turns on and water is supplied to the soil. This can be done in any field which has a current source

and it even pings the user if there is a power cut while watering or if the power cut has taken place for more than certain period of time or if the moisture levels drop down below the stipulated moisture levels and if the power was off when the watering was about to be done. This describes what our project does in a nutshell.

1.1 Existing System

- The existing systems uses cables for communication. Wired networks are mainly used to transfer data to base system by connecting sensor. It brings advantage as it provides reliable and stable communication system for instruments and controls.

The following are the drawbacks of the existing manual System:

- Cost of cables is high
- More chance of damage
- Limits mobility
- Loose cables can be hazardous

- Another Existing System using uses the internet to establish connection between the sensor and the user's hand held device. Thus, the system is independent of cables which make it an easier device to handle.

MERITS-

- User friendly interface
- Accessing of information becomes very easy
- Information sharing is easy.

DEMERITS-

- Connectivity to Internet with Phone.
- If the person isn't in a position to Check the notification, The plants will wilt.
- No Notification when Power is not available.

1.2 Proposed Model

Automated Drip Irrigation system is developed for the benefit farmers so that irrigation becomes a job at hand for them. When the moisture level is not in the required level, then the water pump turns on and water is supplied to the soil. The amount of moisture required by a crop is calculated.When the moisture level is not in the required level, then the water pump turns on and water is supplied to the soil. This can be done in any field which has a current source and it even pings the user if there is a power cut while watering or if the power cut has taken place for more than certain period of time or if the moisture levels drop down below the stipulated moisture levels and if the power was off when the watering was about to be done. This describes what our project does in a nutshell.

The Demerits in Previous Methods are Resolved in most Cases.

2. SYSTEM ANALYSIS

This System Analysis is closely related to requirements analysis. It is also "an explicit formal inquiry carried out to help someone (referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made". This step involves breaking down the system in different pieces to analyze the situation, analyzing project goals, breaking down what needs to be created and attempting to engage users so that definite requirements can be defined.

2.1 Functional Requirement Specification

The System after careful analysis has been identified to be present with the following module.

User Module:

User can access the temperature provided he has the app installed in his hand-held device. After connecting with the Wi-Fi, the user can access the moisture value using the app. In the app, on clicking the manual button, the user can on/off the motor and can also view the moisture value at that point of time. On clicking the automatic button, the user can provide a threshold value. If the moisture value is less than the required one, then the motor gets turned on automatically and vice versa.

2.2 Performance Requirements

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system.

Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

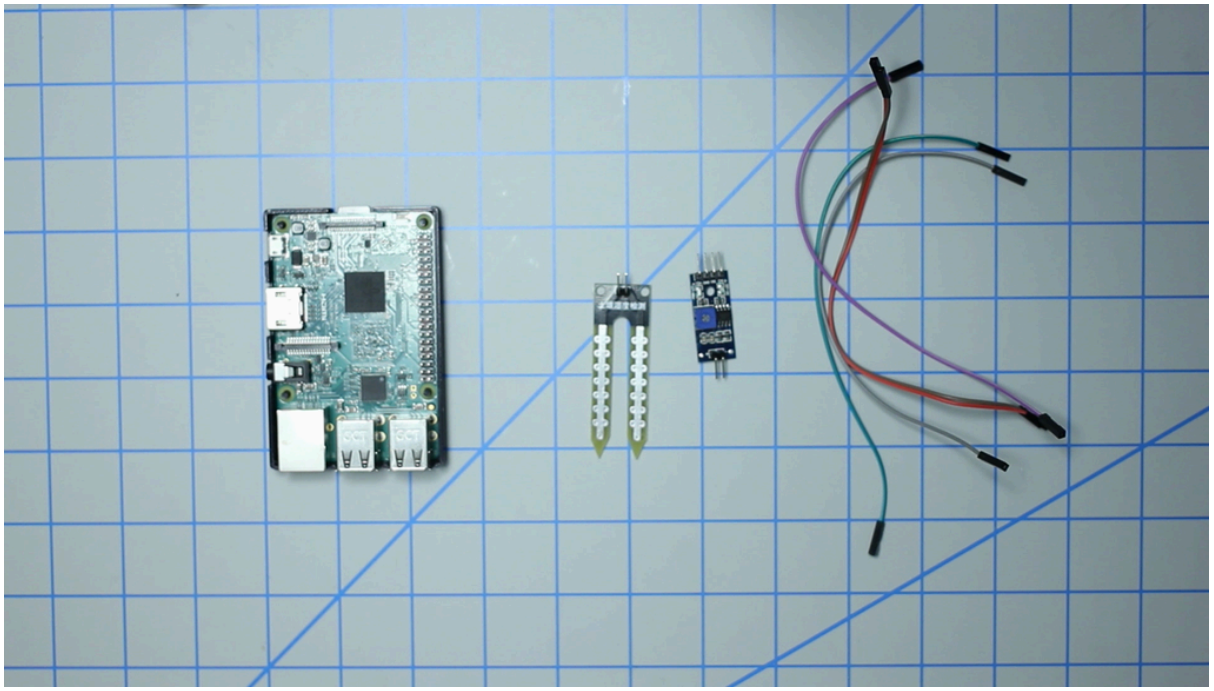
2.3 **Software Requirements**:

- **Technology**: Internet of Things
- **IDE:** Python IDE (such as Thonny, PyCharm, or IDLE)
- **Modules:** Relay Module
- **Libraries :** Required libraries such as GPIO, Time, and Adafruit_DHT.

2.4 Hardware Requirements:

- Raspberry Pi
- Moisture Sensor
- Water Pump
- Power Supply
- Breadboard
- Jumper Wires
- Hose and Drip System

2.5 Parts



3 RPI 3 - https://amzn.to/2VA9pQY

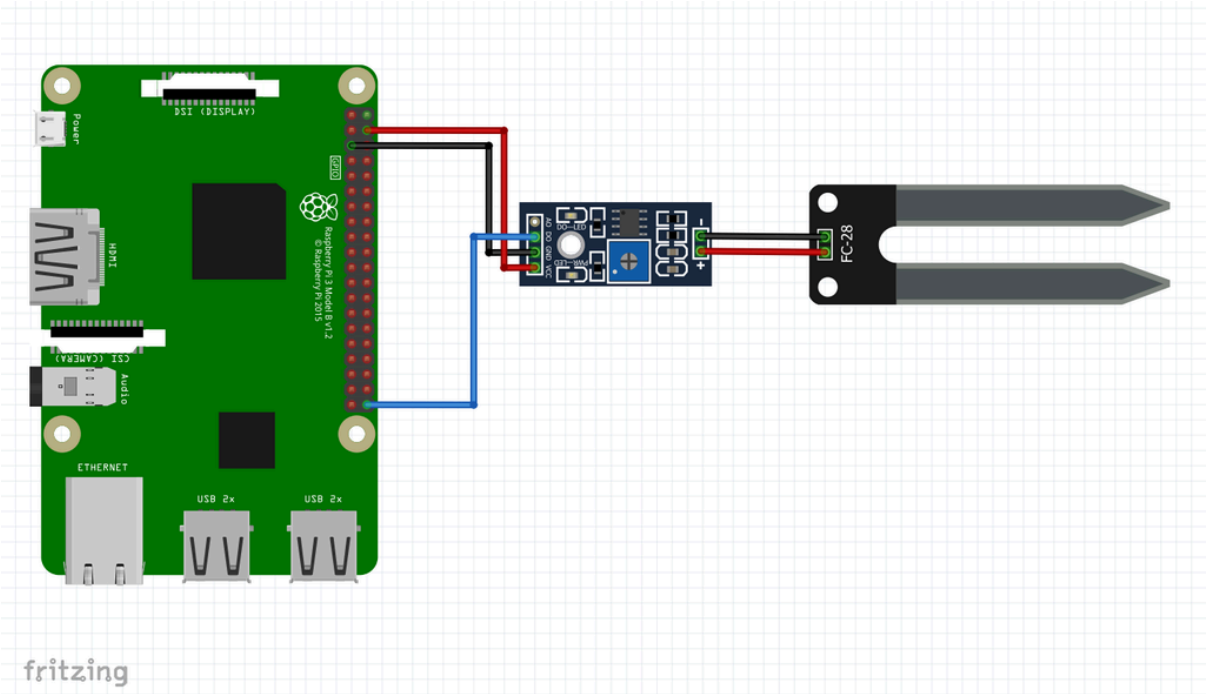
4 4 Amp Power Adapter - https://amzn.to/2CTptWu

5 16GB micro SD - https://amzn.to/2SFMwd3

6 120 pcs jumper cable: https://ebay.to/2VAb9cY

7 Soil Moisture Sensors: https://amzn.to/2DfX5hs

Step 2: Setup



8 VCC -> 5V

9 GND -> GND

10 SIG -> GPIO 21

Soil Moisture sensor :

The Spark Fun Soil Moisture Sensor is a simple breakout for measuring the moisture in soil and similar materials. The soil moisture sensor is pretty straight

forward to use. The two large exposed pads function as probes for the sensor, together acting as a variable resistor. The more water that is in the soil means the better the conductivity between the pads will be and will result in a lower resistance, and a higher SIG out.

To get the SparkFun Soil Moisture Sensor functioning all you will need is to connect the VCC and GND pins to your Raspberry Pi-based device (or compatible development board) and you will receive a SIG out which will depend on the amount of water in the soil. One commonly known issue with soil moisture sensor is their short lifespan when exposed to a moist environment. To combat this, we've had the PCB coated in Gold Finishing (ENIG or Electroless Nickel Immersion Gold). We recommend either a simple 3-pin screw pin terminal or a 3-pin jumper wire assembly (both can be found in the *Recommended Products* section below) to be soldered onto the sensor for easy wiring.

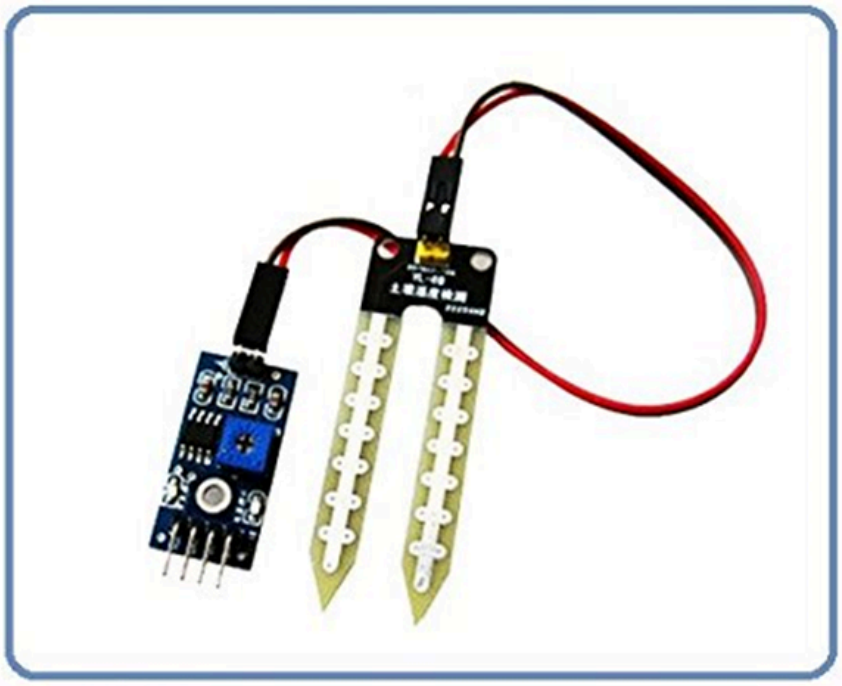
Moisture sensors are devices used to measure the moisture content of soil. They are a critical component of Automated Irrigation Systems, as they allow the system to determine when to water crops.

Moisture sensors work by measuring the electrical resistance of the soil, which is inversely proportional to the amount of water present in the soil. When there is more water in the soil, the electrical resistance decreases, and when there is less water, the resistance increases. This change in resistance is measured by the sensor and used to calculate the moisture level of the soil.

Moisture sensors are available in different types, including resistance-based sensors, capacitance-based sensors, and time-domain reflectometry sensors.

Each type has its advantages and disadvantages, and the selection of the type of sensor depends on the application.

It is essential to monitor soil moisture levels regularly to ensure that the plants receive adequate water, leading to improved yield and quality. Moisture sensors can be a valuable tool in achieving this goal



2.4.2 Soil Moisture Sensor

Features

- Operating Voltage: 3.3V ~ 5V
- Sensing Probe Dimensions: 60x30mm
- On-board power indicator LED

Water Pump:

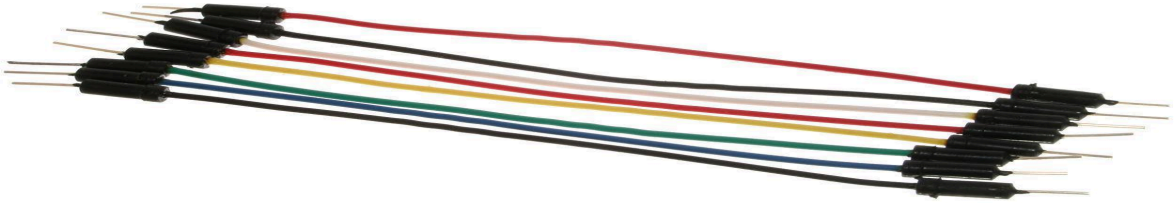
2.4.3 Water Pump

DC Voltage:2.5-6V

- Maximum lift:40-110cm / 15.75"-43.4"
- Outside diameter of water outlet: 7.5mm / 0.3"
- Inside diameter of water outlet: 4.7mm / 0.18"
- Size: Diameter : Approx. 24mm / 0.95" , Length:Approx. 45mm / 1.8", Height: Approx. 33mm / 1.30"

Jump wire

A jump wire (also known as jumper, jumper wire, jumper cable, DuPont wire, or DuPont cable – named for one manufacturer of them) is an electrical wire or group of them in a cable with a connector or pin at each end (or sometimes without them – simply "tinned"), which is



normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

Threshold Value Calculation:

Here are some test plant data for common crops:

Tomato plants: Optimum moisture level for tomatoes is between 60% and 80% of field capacity.

Lettuce plants: Lettuce requires soil moisture levels between 75% and 90% of field capacity.

Corn plants: Corn requires soil moisture levels between 50% and 70% of field capacity.

Strawberry plants: Strawberry plants require soil moisture levels between 70% and 80% of field capacity.

Depending upon the species of the plant, and based on such information as above mentioned, Available in various Government and non-Government Research Sources, the Threshold Values are Calculated for that species.

CODE RASPBERRY PI:

```
import RPi.GPIO as GPIO
import time
import requests
import json

# Define GPIO pins
motor_pin = 16
moisture_sensor_pin = 0

# Define ThingSpeak parameters
channel_id = '282680'
write_api_key = 'WDYQJ9R0EN83O5XP'
read_api_key = 'XXKEXH0HX6AXE1IV'
field_moisture = 'field1'
field_moisture_level = 'field2'
field_pump_status = 'field3'
field_mode = 'field4'

# Define constants
pump_on = 1
pump_off = 0
manual_mode = 0
auto_mode = 1

# Set up GPIO pins
GPIO.setmode(GPIO.BOARD)
GPIO.setup(motor_pin, GPIO.OUT)

# Define functions
def read_moisture():
```

```
    # Read moisture level from sensor
    return GPIO.input(moisture_sensor_pin)

def turn_pump_on():
    # Turn on the water pump
    GPIO.output(motor_pin, GPIO.LOW)
    print("Pump turned on")

def turn_pump_off():
    # Turn off the water pump
    GPIO.output(motor_pin, GPIO.HIGH)
    print("Pump turned off")

def update_thingspeak(field, value):
    # Update ThingSpeak channel with specified value
    url = 'https://api.thingspeak.com/update?api_key=' + write_api_key + '&field' +
str(field) + '=' + str(value)
    requests.get(url)

def read_thingspeak(field):
    # Read latest value of specified field from ThingSpeak channel
    url = 'https://api.thingspeak.com/channels/' + channel_id + '/fields/' + str(field) +
'.json?api_key=' + read_api_key + '&results=1'
    response = requests.get(url)
    data = json.loads(response.content)
    return data['feeds'][0]['field' + str(field)]

def main():
    # Initialize the system
    print("System started")
    update_thingspeak(field_mode, auto_mode)
```

```
# Loop indefinitely
while True:
    # Read moisture level from sensor
    moisture = read_moisture()
    print("Moisture level: ", moisture)
    update_thingspeak(field_moisture, moisture)

    # Read required moisture level from ThingSpeak
    moisture_level = int(read_thingspeak(field_moisture_level))
    print("Required moisture level: ", moisture_level)

    # Read mode from ThingSpeak
    mode = int(read_thingspeak(field_mode))
    print("Mode: ", mode)

    # Automatic mode
    if mode == auto_mode:
        # Check if moisture level is below threshold
        if moisture <= moisture_level:
            # Turn on the water pump
            turn_pump_on()
            update_thingspeak(field_pump_status, pump_on)

        # Check if moisture level is above threshold
        if moisture > moisture_level:
            # Turn off the water pump
            turn_pump_off()
            update_thingspeak(field_pump_status, pump_off)

    # Manual mode
    if mode == manual_mode:
        # Read pump status from ThingSpeak
```

```
        pump_status = int(read_thingspeak(field_pump_status))

        # Turn on or off the water pump based on pump status
        if pump_status == pump_on:
            turn_pump_on()
        else:
            turn_pump_off()

    # Delay for 20 seconds before looping again
    time.sleep(20)
```

Disclaimer: The code is Still in Construction and Developments are yet to be made.

Modes:

EXPLICIT MODE:

In the explicit mode, we have a text box where the user can enter the required moisture value and a button to submit. The input value is taken as the Threshold Value. When the moisture level of the soil decrease, below the Threshold Value The motor pumps the water to the plants Automatically up to a limit which is slightly Higher than the Threshold Value. The user can get the values by selecting the getvalues button.
The user also has data analysis button to get a graph of the data by time.

AUTOMATED/IMPLICIT MODE:

In this mode, depending upon the type of the crop entered, The moisture value is calculated and is set as Threshold Value. When the moisture level of the soil decrease, below the

Threshold Value The motor pumps the water to the plants Automatically up to a limit which is slightly Higher than the Threshold Value.

5.TESTING

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

6.1 Testing Strategies

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

- ✔ **Unit Testing:**

  Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements

- ✔ **Integrating Testing :**

  Integration testing ensures that software and subsystems work together a whole. It tests interface of all the modules to make sure that the modules behave properly when integrated together.

- ✔ **System Testing :**

It involves in house testing of the entire system before delivery to the user. Its aim is to satisfy the user the system meets all requirements of the client's specifications.

✔ **Acceptance Testing** :

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors.

✔ **Validation**:

The system has been tested and implemented successfully and thus ensured that all the requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

5.2 Test Cases

Table 5.1 Test Cases

TEST CASE NUMBER	TEST CASES	EXPECTED OUTPUT	IF ACTUAL OUTPUT	TEST PASS/FAIL
1.	CONNECTION ESTABLISHMENT	CONNECTED SUCCESSFULLY CONNECTED SUCCESSFULLY	CONNECTED SUCCESSFULLY NOT CONNECTED	PASS FAIL

2.	MOISTURE VALUE RETRIEVAL	RETURN VALUE RETURN VALUE	DISPLAY SUCCESSFUL DISPLAY FAIL	PASS FAIL
3.	MOTOR TURN ON	SWITCHED ON PROPERLY TURNED ON	SWITCHED ON PROPERLY TURNED OFF	PASS FAIL
4.	MOTOR TURN OFF	TURNED OFF TURNED OFF	TURNED OFF DID'NT TURN OFF	PASS FAIL

7. CONCLUSION AND FUTURE SCOPE

Conclusion

Nowadays, watering their plants is a major problem for farmers. This will make them stop
doing all other activities.

Here is an idea which helps the farmers' work easy by automatically sensing the moisture level and supplying water only when there is a need.

We hope that through this prototype we all can enjoy having plants, without being worried about absent or forgetfulness

When certain jobs can be done without human involvement, why to invest our precious time on such mundane works?

It would be wiser if we leave those works to the machines .This will help us to invest our time in wiser jobs.

This can be achieved by our project.

Future Scope

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used.
Some of the future enhancements that can be done to this system are:

- The level of water in the tank from which the water is being supplied may also be monitored.

- The data from the sensor can be further analysed to know the information like the time for which different soils are moisture can be analysed.

BIBLIOGRAPHY

http://www.decagon.com/en/soils/volumetric-water-content-sensors/

http://www.instructables.com/id/Automatically-water-your-small-indoor-plant-using-/

https://thingspeak.com/channels/299803/api_keys

https://thingspeak.com/channels/299803/private_show

http://easycoding.tn/tuniot/demos/code/

https://www.instructables.com/Soil-Moisture-Sensor-Raspberry-Pi/

https://robu.in/product/waveshare-moisture-sensor/

https://sixfab.com/smart-agriculture-projects-using-raspberry-pi/

Contact:

1. Abhinay Cherupally – 21311a6624@sreenidhi.edu.in
2. Prem Kiran Laknaboina - 21311a6611@sreenidhi.edu.in
3. Vishnu Vardhan Reddy - 21311a6607@sreenidhi.edu.in