

Monitoring and Control of Domestic Gardening with Feedback System using Internet of Things

Mr. Samuel Bryan Saynik¹, Mr. Sai Krishna K²

^{1, 2} UG Student, Dept. of Electronics and Communication, St. Joseph's Institute of Technology, Tamil Nadu, India

Abstract – This project “Monitoring and Control of Domestic Gardening with Feedback System using IoT”, proposes an automated system to modify the watering cycle. The system utilises ‘Raspberry Pi’ as a master controller with the ability to automatically monitor and control through a web-interface, the values from moisture level sensors to ensure adequate watering cycles. The benefit is to implement cost-effective automated systems in townships and homes to eliminate human intervention, reduce labour cost and regulate the usage of resources with the ability to be integrated with other home automation systems.

Key Words: water distribution, water management, IoT, cloud computing, moisture sensor, remote access

1. INTRODUCTION

Water is a resource racing towards extinction. If unchecked it can lead to disastrous consequences. The only way to counter this problem is by afforestation. People nowadays prefer green parks and kitchen gardens but harsh climatic conditions can cause damaging effects if the watering cycles are not regular. The main motivation behind the project is to maintain the green cover in townships and maintain kitchen gardens. Construction giants are into the trend of building townships spanning acres of land. They provide green cover as an aesthetic appeal. The governmental laws also emphasise on certain portion of the total area to be dedicated to greenery. This was a wonderful initiative started by the government. But after few years the association takes up the task of maintaining it. Several issues arise in the maintenance. Ground water is not abundantly available these-days. Purchasing water for day-to-day use has become expensive. Human labour to maintain the plants is also expensive. Thus due to mismanagement, these plants begin to wilt. Thus the green cover starts deteriorating. People also have various flowering plants in their homes. When people go out on long trips, these plants are not watered properly causing them to wilt. Thus to prevent the wilting of these precious plants and trees we proposed an automated watering system. It would modify the watering cycle based on the soil moisture values. This prevented the unwanted usage of water. Nowadays home automation systems have become a fancy trend. Thus we proposed to integrate this automated system into any home automation system. Thus by integrating the technology of IoT, we have proposed monitoring the moisture levels and the watering cycle. This also enables the access to the various systems from remote locations. With the implementation of a feedback system we have proposed to make the system fail proof.

2. COMPONENTS

The proposed automated system uses a wide range of low power systems to control the overall process. The components are integrated into the GPIO pins of the controller and the values obtained separately are processed to carry out the defined functions.

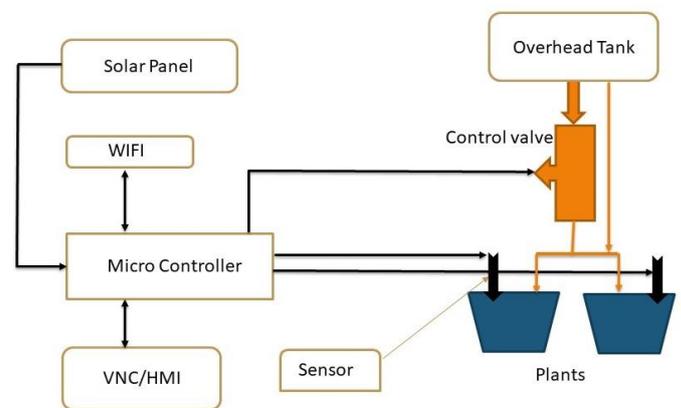


Fig -1: Block Diagram

2.1 Micro Controller

The core controller used in this system is a Raspberry Pi 3B model. Powered by a Raspbian JESSE OS it can work as a standalone computer. The raspberry pi enables it to be programmed directly without the need for another system to program it. It also makes integration other IoT systems into it very easy. The raspberry pi has an inbuilt Wi-Fi system with HDMI connectivity. With a 1 GB RAM and a 1.2 GHz processor the raspberry pi is an ideal controller for home automation system.



Fig -2: Raspberry Pi 3B

The controller can process 26 inputs simultaneously. Thus the raspberry pi is ideal to scale up the system as well. With in-built Bluetooth and Wi-Fi connectivity it also allows the implementation of wireless systems.

2.2 Soil Moisture Sensor

The basic component utilized here is the moisture sensor which is used to determine the amount of moisture present in the soil. It works on the principle of “soil conductivity varies according to the moisture levels”. Humans tend to over-water or under-water the plants. Thus using the soil sensor we determine when to initiate the watering cycle. By implementing this the water level is accessed and accordingly the watering cycle is modified. This enables prevention of over-watering also. This is done by pre-setting the moisture level to be maintained for the plants.

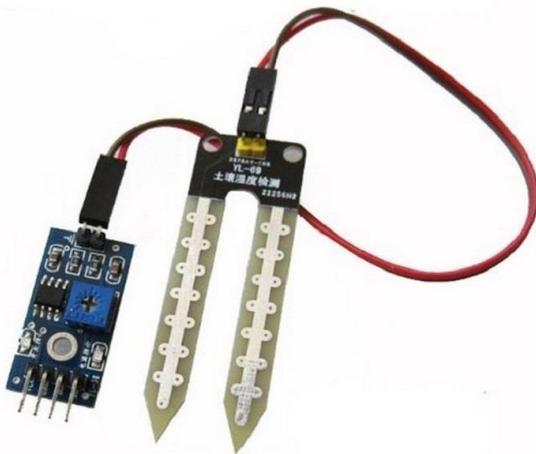


Fig -3: Moisture Sensor

The moisture sensor uses a LM393 Comparator to determine the conductivity of the soil and accordingly determine the soil moisture level. The potentiometer in the comparator is used to alter the reference value and thus set the required water moisture level. The sensor provides both analog and digital values. The analog values are more accurate but to reduce the amount of hardware we directly use the digital value as the controller can directly process it. The sensor provides data that can be input into any one of the controllers GPIO pins.

2.3 Real Time Clock (RTC)

IoT systems works with constant cloud updates. The proposed system is designed to work in offline connections and updates when reconnected. Thus to maintain the system time, we use an external timer. This maintains the time and updates the controller’s system time every time it reboots. The module used is a DTS-3231. It is a highly accurate module with an accuracy of ± 2 ppm.



Fig -4: RTC Module DS-3231

The RTC system is used to provide a time based system also. This system is used as a default when any abnormality is detected in the system.

2.4 Relay and Solenoid Valve

Relay is an electronic switch. The Raspberry Pi works on a power supply of 5V whereas the valve requires a minimum of 12V or 24V. Thus to drive the valve we need a relay. The relay can control 230V with just a 5V signal from the controller.



Fig -5: Relay

The Solenoid valve is an electrically controlled valve. This is used to control flow of liquids by electrically powering a solenoid which controls its opening and closing. The valve SVK-01 is used in the system. It is a normally closed valve. It can handle high levels of pressure. Based on inputs from the raspberry pi, the relay operates the valve to initiate the watering cycle.

The solenoid, which is a coil, activated by external voltage, pulls the mechanism that operates the valve. Thus the valve opens. The pressure handling capacity depends on the strength of the solenoid used in the valve and the input voltage.



Fig -6: Solenoid valve SVK-01

2.5 Feedback System

Automation systems are meant to reduce human intervention. To prevent any mishap we have implemented a feedback system. The water flow is gauged with a flow sensor. If the moisture sensor fails, it shows as dry always, thus watering continues infinitely. Thus if the system detects any abnormalities it immediately shuts down the valve.



Fig -7: Flow Sensor YF-S201

The flow sensor YF-S201 used in this works on the principle of "Hall Effect". This is to prevent dry runs (Valve operates with no water flow), or overruns (Failure to close signal from controller). The system directly shifts to the time based system which initiates the watering cycle according to a pre-set time.

3. WATERING CYCLE CONTROLLING SYSTEM

Watering cycle is the major area of interest in this system. This system was designed to work offline or online with cloud integration. The system provides the selection of any one system: Timer based or Moisture level based.

3.1 TIMER BASED SYSTEM

Certain types of plants require regular watering. Thus with the already integrated RTC, we implemented a timer based system. This is a fail proof system as the RTC module

has a lifetime of 10yrs for a small coin battery. The time to initiate the watering cycle can be pre-set. At the set time, the watering system is initiated and for a determined period of time the watering cycle is performed. This is irrespective of the values of the sensor. This system is used as the backup system, when the system detects any discrepancies in the functioning of the moisture based system. The time to initiate the watering system can be adjusted according to the location of implementation.

3.2 MOISTURE SENSOR BASED SYSTEM

Moisture is measured as volume or density of water in the soil. Depending on the moisture levels, the conductivity of the soil varies. The soil moisture sensor, detects the conductivity. The threshold is pre-set using potentiometers. Once the conductivity falls below the threshold, the controllers notifies the relay. Once all the sensor values drop below the threshold, the relay activates the solenoid valve, to initiate the watering cycle. The moisture sensor continues to monitor the moisture levels. Once the threshold is crossed, the watering cycle is terminated. Thus the automated system, maintains the moisture level at a pre-set level, thus optimising the usage of resources to the level needed.

The flow sensor detects the flow of water and provides the feedback for the system. If the water flow is not detected, the controlled sends an error message. This denotes that a dry run is taking place. Thus we can intervene and terminate the process, or take the necessary steps needed. Meanwhile, the program goes to sleep for a specific amount of time to prevent damage to the hardware.

If the moisture sensor fails or the controller fails to close the valve, the feedback system sends an error message as the flow sensor detects water flow for more than the specified time. Thus the feedback system shuts the program and jumps to the timer based system and closes the valve. Thus damage to the plants and wastage of resources is thus prevented.

4. RESULT

The system was successfully tested. The system can automatically detect moisture levels and activate watering cycles. The extent of human intervention can be drastically reduced.

The efficiency in reducing the consumption of water resource was drastic. The web-interface provides live values as well as stored data gathered over time. It also allows access to the system remotely from the same network as well as different networks.

The chart shows the data plotted for one watering cycle.



Chart-1: Working (plotted data)

5. APPLICATIONS

This system can be implemented in townships and kitchen gardens.

The system can be adapted to maintain large spread organic farms.

6. CONCLUSION

Thus our proposed project will revolutionize the overall maintenance of domestic gardening. It also reduces resource consumption which is the major issue in the present day scenario with a feed-back system to prevent damage to hardware.

7. FUTURE ENHANCEMENTS

The system can be run with the help of solar power. With a 10W solar panel and a 50VAh battery the system can be fully powered by a renewable power source. This makes it a stand-alone system with high efficiency and no operational cost.

The implementation of NodeMCU as a satellite node enables wireless data acquisition. Multiple nodes can be compiled under "master-slave" technology to get data from individual slave nodes, whereas the computation is done by master node. This eliminates the use of wires to connect the sensors to the controller.

By using Adafruit cloud, we can create a separate mobile application instead of the web-interface now being used

REFERENCES

- [1] [1] Ahmed Hussein Abbas, Maya Medhat Mohammed, Eman Adel Ahmed, Rania Ahmed Abdel Azeem Abul Seoud and Gehad Mostafa Ahmed, "Smart Watering System for Gardens using Wireless Sensor Networks"
- [2] Bo Cui and Qinghong Meng, "Smart Water Monitoring and Management System Based on the Architecture of Internet of Things"
- [3] Asha Saif and Lijesh L "Advanced Globally Accessible Machine Automation Using Raspberry PI"
- [4] Botta, Alessio, Walter De Donato, Valerio Persico, and Antonio Pescapé, "On the integration of cloud computing and internet of things", In Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, pp. 23-30. IEEE, 2014

BIOGRAPHY



Samuel Bryan Saynik is a highly enterprising student with a passion towards developing technologies in the field of electronics.