Smart Irrigation System

ELEG/CPEG 480- Capstone Design Project II

Project Members

Ali Ahmad (S0030910)

Dalal Al-Mshaweh(S00034482)

Fatima Al-Kuwaiti(S00038777)

Yousef Al-Saffar(S00031111)

Project Supervisor(s)

Dr. Hanan AlTabbakh

Co-Supervisor:

Dr. Detlef Hummes

Department of Electrical & Computer Engineering

AMERICAN UNIVERSITY *of* KUWAIT

JUNE 13, 2021

Smart Irrigation System

Project Members

Ali Ahmad (S0030910)

Dalal Al-Mshaweh(S00034482)

Fatima Al-Kuwaiti(S00038777)

Yousef Al-Saffar(S00031111)

The capstone project report is being submitted in partial fulfillment of the requirements for the degree of

Bachelor of Engineering in Electrical/ Computer Engineering

Project Supervisor(s):

Dr. Hanan AlTabbakh

Supervisor's Signature: ________________________________

Department of Electrical & Computer Engineering

AMERICAN UNIVERSITY *of* KUWAIT

JUNE 13, 2021

Declaration

We certify that this project work titled "*Smart Irrigation System*" is our own work. The work has not been presented elsewhere for assessment. The material that has been used from other sources has been properly acknowledged / referred.

<u>Project Members</u>

Ali Ahmad (S0030910)

Dalal Al-Mshaweh(S00034482)

Fatima Al-Kuwaiti(S00038777)

Yousef Al-Saffar(S00031111)

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by the Supervisor(s) is attached.

Project Members

Ali Ahmad (S0030910)

Dalal Al-Mshaweh(S00034482)

Fatima Al-Kuwaiti(S00038777)

Yousef Al-Saffar(S00031111)

Signature of Supervisor(s)

Copyright Statement

Acknowledgements

Ali Ahmad (S0030910)

Dalal Al-Mshaweh(S00034482)

Fatima Al-Kuwaiti(S00038777)

Yousef Al-Saffar(S00031111)

"A huge thanks for my team members for the effort of reaching this stage of accomplishment, and special appreciation for Rapid Manufacturing Technology for giving us the workspace in such time and sponsoring our smart irrigation system project".

Abstract

In the era of modern technology, development is a major wheel that is in motion. At first it was a computer that filled a room with wires and now, in hand with a single click that makes the difference of several commands. But to surpass the basic term of technology is to bring up the term IOT which is Internet of Things and the classification of it is to refer to the network of physical objects that are surrounded with sensors, software and plenty of technological basis in order to exchanging data with other devices and systems over the Internet. The limit of IOT is not only to exchange data, but to allow the system to communicate and control information by applying a meaningful purpose. IOT is based protocols and layers which present the major part of what is going to be applied and inserted for the required system. Which is why we bring up the regular Irrigation system that is mostly installed in home and factories. Such Irrigation system is controlled by a timer which requires adjustments that differs from season and depending on the weather condition. However, our aim as a team is to improve such mechanism and to make it a smart system that is self-adjusting no matter what the weather condition is and make it associated with an alternative energy source which is solar power.

Basically, the IOT is a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocol where physical and virtual things have identities and interact through intelligent interfaces with users and their environment. Thus, our Smart Irrigation system will solve itself by creating a system with a mind of its own.

Key Words: *Agriculture, IOT, Irrigation, Autonomous, System, Microcontroller, Sensors, Water. Technology.*

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

1.1 Background

When it comes to the term of irrigation system, it is a basic structure of the water sprinklers and water links and not to forget, the main controller which is the timer. The procedure of this basic system is to set the timer to a standard timing in which the water is on for it to sprinkle the soil. It requires human interference in order to set the timer from one season to another and whether the soil is a special type of soil which requires more water than the usual soil to be gardened. That is why adding the term smart to the irrigation system will make that problem solved. The solution is in the term IOT which will make the solution quite flexible and in the same time, more efficient.

1.2 Problem Statement

The underlying matter to be solved is that the irrigation system in its basic structure is not efficient and not adopting the advanced technology of today. Basic irrigation requires the human factor for it to be fully functional and maintenance on a regular basis for adopting to a certain parameter of soils and weather conditions and not only that, but also in case malfunction while being in a remote country.

1.3 Aims and Objectives of the Project

The aims of this project are to modify the irrigation system to another level. From the basic irrigation to smart irrigation with the relation to IOT definition to enhance its performance to a non-human interference system which will make the system flexible to self-configure itself without even being near the grid of the irrigation system.

.

1.4 Significance, Scope and Definitions

To highlight the project is to bring up our methodology, which is electrical component and circuits and their connections, agriculture and Internet of Things (IOT) and application platform which we will be explaining deeply in the report. Also, we will be covering alternative energy

Footnote may be given with Font size 10 of Times New Roman.

source and specifically, solar energy which is a field of interest that will make a difference it our project appliance.

1.5 SWOT Analysis

Our advantage in the project is that we are all Electrical students and took the required courses for us to apply the required connection between the components and understand them from several perspectives. Also, three of us took a course in IOT and one of us is still attending the course. The IOT course covers what we require to achieve the basis of self-configuring system and a smart structure from a standard irrigation system to an advanced flexible system. However, our issue relies in the adaptation of the application platform which requires more board base of knowledge.

1.6 Report Outline

In the following chapter we will cover the literature review of sources and come up with a solution that will guide us to build and establish a design to solve such question, which is how to make sure that the irrigation system is functional and more effective with the balance of agriculture parameters. So after that, we will explain our design in details and how the implementations we done and lastly, expand our scope to the market and other major field.

Footnote may be given with Font size 10 of Times New Roman.

CHAPTER 2: LITTERATURE REVIEW

2.1 Historical Background

To look at the smart irrigation system, we will start as we have stated before in the basic irrigation system. The first irrigation system was based on human factor where plants and roots were picked and placed manually by workers, then machines came to make a difference which made it more sufficient but still human factor was required to work the machines. After that all it became was to turn on the sprinklers on a particular timer machine and let the system operate under certain timing. While now, our purpose is to design an irrigation system that is more developed and not requiring a human factor, only the use of an ON/OFF button.

2.2 Agriculture

Agriculture is the field which deals with the plants based on their environment and what keeps it alive, is the irrigation process. A regular irrigation system at first was a worker, going in the plantation field, checking on the soil and harvesting or giving water. Then it became a timing machine along with automated water sprinklers. Now, it became by a click and an automated system that does all the previous. The department of mechanical engineering of the federal university of technology, Nigeria and the Federal Institute of Industrial Research developed a system that prevents such procedure based on sensory system and smart based to prevent faults with the accuracy of the PIC16f876A microcontroller. Thus such step to elevate the structure of the irrigation system was wise for it to be more efficient and the irrigation is based on quality first which will make quantity insured. Anneketh Vij published in Procedia Computer Science, that building a smart irrigation system resolves the problems faced by the farmers at a cheaper cost and building a machine learning algorithm that contradicts the weather and different situations to make irrigation at the best state. Therefore, transforming a regular irrigation to a smart, developed system make the process more efficient and more productive.

2.3 Water Consumption

The main purpose for a smart irrigation system is not only make thing simple by removing the human factor, but simple solution provide sometimes grant better performance and results. So water being used and making the main source for the system as Vij stated, it is being used by the

Footnote may be given with Font size 10 of Times New Roman.

gallons which should be minimized because the over use of water on a soil could lead the ruining of the plantation process. Also, in the same time we do not want to make the soil dry and not provide it with enough moisture. A survey is shown in 2019 by N, Sudharshan that 20 percent of the soil lands are wasted due to water shortage. Therefore, providing a balance for the agriculture stand is based mainly on the source of it, water which should be distributed fairly by not offering "too much or too little" (Moya, Tolentino B. 2018) So, our design will offer such solution.

2.4 Climate

Any system is controlled by its environment and so is the irrigation system. Clearly we have searched for the effects of climate on the plantation process and it showed quite the difference. In the Philippines, the government has been heavily investing in irrigation development to boost crop yield and to enlarge currently irrigated areas for many years now. However, the Philippine climate has been changing and the climatic variants and change present potential threats to the resilience of the Philippine irrigation systems and such information was provided by DLSU Business & Economics Review, 2018. Additionally, the choice of a proper irrigation system be subject to factors such as the crop to be irrigated, water, soil state, and geography. So designing a smart irrigation system will cover such parameter based on the sensors that will determine when and what to do comparatively.

2.5 IOT

Internet of things is presenting the self-configuring system concept as a definition. Our sources state that agriculture market being developed by the knowledge of IOT is making it sustainable and more productive (Anneketh, 2020). Simple solution such as Anneketh presented in the publication was to establish a Monitoring system whose main resolution is to elucidate over irrigation, soil destruction and crop specific irrigation problem will be developed to ease and efficiently manage Irrigation problems.

2.6 Summary and Implications

As we have stated in the previous topics, which showed what the proper irrigation system is should be created and how to make it efficient to the point it solves the problems faced in the irrigation process. Also, seeing that the adding an IOT field of application and making the smart

Footnote may be given with Font size 10 of Times New Roman.

irrigation system presentable in a manner of being more effective and saving water without the loss of the crops state is the main goal to even making it a highly developed irrigation system. Therefore, we have approached our design with the intent to save water based on a specific timing and relating it to the environment state and check on the condition of the soil by smart device.

Footnote may be given with Font size 10 of Times New Roman.

CHAPTER 3: METHODOLOGY, DESIGN AND ANALYSIS

3.1 Methodology

For our first try of the system we had the moisture sensors which collects data from the soil moisture for 24 hours and the water pumps would spray water whenever the sensors gave readings below 80, but this caused a problem with the readings because the water needs time to reach the bottom of the soil and for the sensor to give the correct reading therefore it lead for the sensors to give unrealistic readings and spray water unwantedly. To solve this issue, we made the sensors collect data at two times per day 8:00 a.m. and 6:00 p.m. and if the readings gave below 80% for the soil moisture, it would spray water accordingly since the plant do not need water consumption more than that.

3.2 Research Design

In our research design we concentrate on the quality of the soil moisture and our design is made specifically to give the plant the appropriate atmosphere it needs. To take care of the soil moisture we made the water pump sprays water for only 2 seconds whenever the sensors gave readings below 80 at two times per day therefore the plant does not intake more water than needed which can spoil the plant growth. Readings of the sensors will be taken at two times of the day and if the reading gave 80 and above the water pump will not spray water and waits till the second time of the day, and if still the moisture level was 80 and above it will wait till the next day. This helps the system to function effectively during summer and winter times since the soil dryness time differ in different conditions.

3.2.1 Design Alternative 1

An alternative design is CNC (Computer Numerical Control) farming machine that automatically grows plants while giving the user the total control of the plantation, in which the user can precisely plant the seeds in any pattern wanted and water each plant based on its type, age, soil conditions, etc. By using an onboard camera and advanced computer vision the system can scan the garden to detect weeds and take act immediately. The user also can monitor the garden and plant seeds from any phone, tablet or computer using an application in which there is

Footnote may be given with Font size 10 of Times New Roman.

no need for coding and can be used by any one even young and non-technical users. In this system raspberry pie and Arduino are the brain of the system.

3.2.2 Design Alternative 2

Another alternative design is similar to our irrigation system but instead of the three water valves supporting three water pipes each drilled through the pots, we could have a servo motor connected to one water pipe through one water valve which rotates according to the requirement. So for example if we have pot A and pot B the sensors detect which soil have less moisture, and if pot B has less soil moisture than pot A then the servo motor rotates toward pot B and starts watering till it finishes then it would rotates towards pot A.

3.3 Software and Hardware

The hardware part of our system consists of:

- Micro controller which is Arduino MKR 1010 connected through WIFI which consumes 5 volt, this variable is responsible for collecting data.
- Arduino MKR Environmental Shield allows the Arduino MKRE to acquire environmental data collected by an array of sensors that can detect atmospheric pressure, temperature and humidity, ultraviolet UVA intensity and ultraviolet UVB intensity, UV index and light intensity.
- A 12V DC waterproof water pump responsible for distributing water to the three water valves.
- A POLOLU 5V step down voltage regulator which can take up to 38V and efficiently reduces it to 5V.
- A 5V corrosion resistant Capacitive Moisture Sensor placed directly into the soil to detect the soil moisture.
- A VISHAY TEMT6000 light sensor sensitive to visible light similar to the human eye which is placed on the MKR Environmental shield.
- A Capacitive digital sensor for relative humidity and temperature STHTS221 which is placed on the MKR Environmental shield to detect the atmospheric humidity and temperature.

Footnote may be given with Font size 10 of Times New Roman.

- An air pressure sensor ST LPS22HB placed on the MKR Environmental shield to detect the atmospheric pressure.
- An ACS 712 current sensor module 5A placed on the MKR Environmental shield to detect the current.
- Smart device

Item Code (Source)	Item Description	Qty.	Cost - KD
Arduino Store USA	Arduino MKR WIFI 1010	1	10.5 KD
Arduino Store USA	Arduino MKR Environment shield	1	11.3 KD
TEMT6000X01	Ambient light sensor	1	1 KD
STHTS221	Humidity sensor	1	6.080 KD
ST LPS22HB	Air pressure sensor	1	2.50 KD
ACS 712 eva store	Current sensors	1	2 KD
RTC	Mousiture Capacitor sensor	3	9KD
	Solar System - PV	1	20KD
RTC Electronics	Water Pump		2.750 KD
Andriod Os phone	Smart phone	1	
Pololu	Stepdown Voltage Regulator	1	8 KD
Total Cost			73.13KD

Table 1: list of component to the smart Irrigation system

The software part includes:

- C++ coding language for implementing the coding algorithm
- Micro-controller IDE (Arduino) which is based for the micro-controllers but we mainly used the online IDE
- Arduino IOT cloud which we used the web editor to implement the code

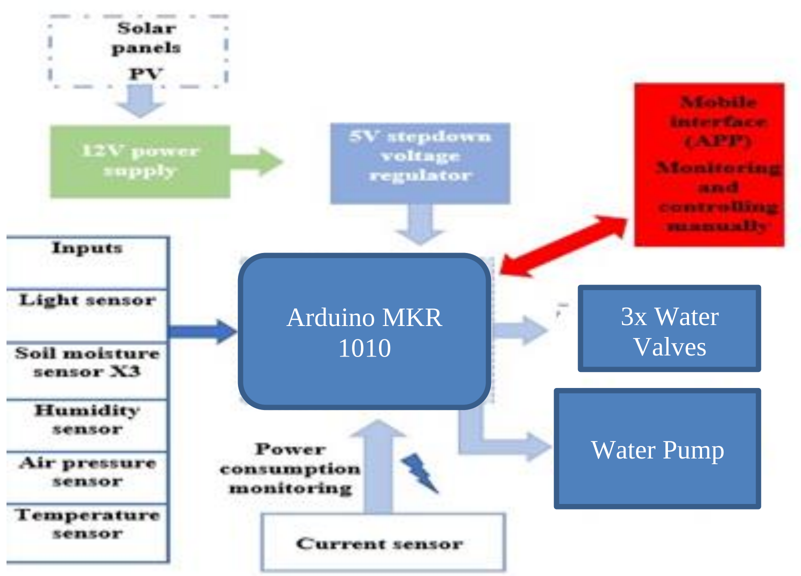

Figure 1: smart irrigation system chart

Footnote may be given with Font size 10 of Times New Roman.

3.4 Analysis

The data is collected through the Arduino MKR 1010 then transferred to a cloud where all the data is being saved and analysed. All recorded history of the plantation process can be found in the cloud where it can be helpful for the system to know exactly how much the plant needs water and how much the plant consumed water in the past three months. In our case, our system only detects the soil moisture at two times per day and if the soil moisture is below 80 percent then it would spray water for two seconds only and waits for the second times of the day and this process is repeated every day, in this way we can make that the plant does not consume more water than needed which can harm the plantation process.

3.5 Ethics and Limitations

The smart irrigation system we have built when applied for big farms will lead to less human labour leading to many farmers losing their jobs since the smart irrigation system can do all plantation process except for cutting the plants whenever needed, therefore human labour is not completely out of the picture. However, the smart irrigation system can solve a lot of issues in cases of farmers having to drive to far places where the farms are located sometimes not having a car to go the farms also working under bad weather conditions for example here in Kuwait temperature can rise to 55 degrees Celsius. The smart irrigation system can help these cases in which it can be monitored through an application. Our smart irrigation system we have built is a prototype, since it is just a prototype it cannot handle the severe conditions. However, to properly describe our design is that we place the micro-controllers and the remaining components in a controlled environment such as a control room while the remaining network grid of piping, will be distributed in any case scenario.

Footnote may be given with Font size 10 of Times New Roman.

CHAPTER 4: IMPLEMENTATION

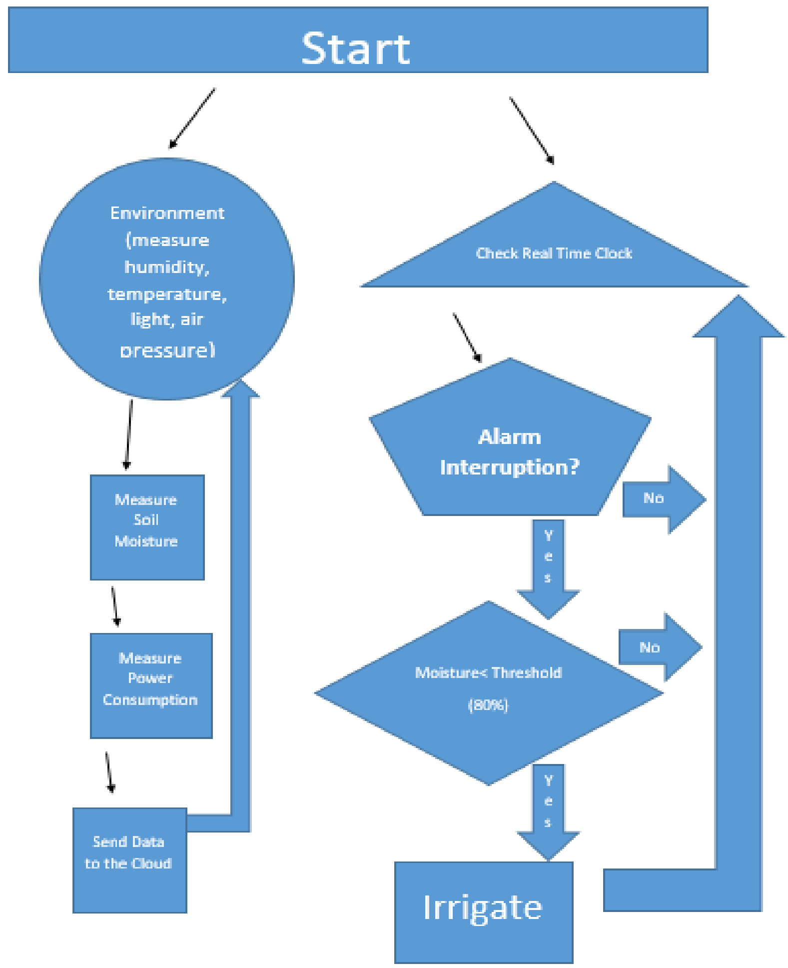

Figure 2: Smart Irrigation block Design

4.1 Hardware Implementation

First, we start our energy source which is a 12V solar or a standard 12V power source which if we applied the regular power source then the current sensor(ACS 712) will be used for feedback in matter of power consumption. Then the system is up for operation by a switch which starts the components to run the irrigation process. The pump is provided with 12V and as well the water valve which will distribute the water for the plants. The valve is configured on the parameter

Footnote may be given with Font size 10 of Times New Roman.

of the soil itself before water distribution. But in the same time we want to ensure the quality of the soil for it not to be wet or dry in order for the soil to be in top condition. Then the remaining is the relay and sensor and Micro-controller. These components are running on a lower voltage consumption which is 5V. Therefore, a step down voltage (Pololu 5V, 5A Step-Down Voltage Regulator D24V50F5) is placed on a breadboard and grants the components their required voltage to be functional. Also, the relay (JQC-3FF-S-Z) works on 5V and one channel relay module which operates as a switch for the system based on the receiving part of the system. While the Capacitive Moisture sensors are placed directly into the soil for reading and maintaining the condition of the moisture case of dryness which is based on the quality of the soil itself. While the Arduino MKR 1010 is placed on the breadboard for the controlling part of the system which grants also a Wi-Fi and Bluetooth based IOT which sends data to be analyzed and see the functionality of the complete system on a cloud based storage. Also, we placed an Arduino MKR environment shield which provides reading of air pressure, temperature, humidity, Ultra-violet index in calculation and intensity and light intensity which used ST LPS22HB sensor for atmospheric pressure, ST HTS221 sensor for temperature and humidity, VISHAY VEML6075 for UV wave length and VISHAY TEMT6000 for light ambient.

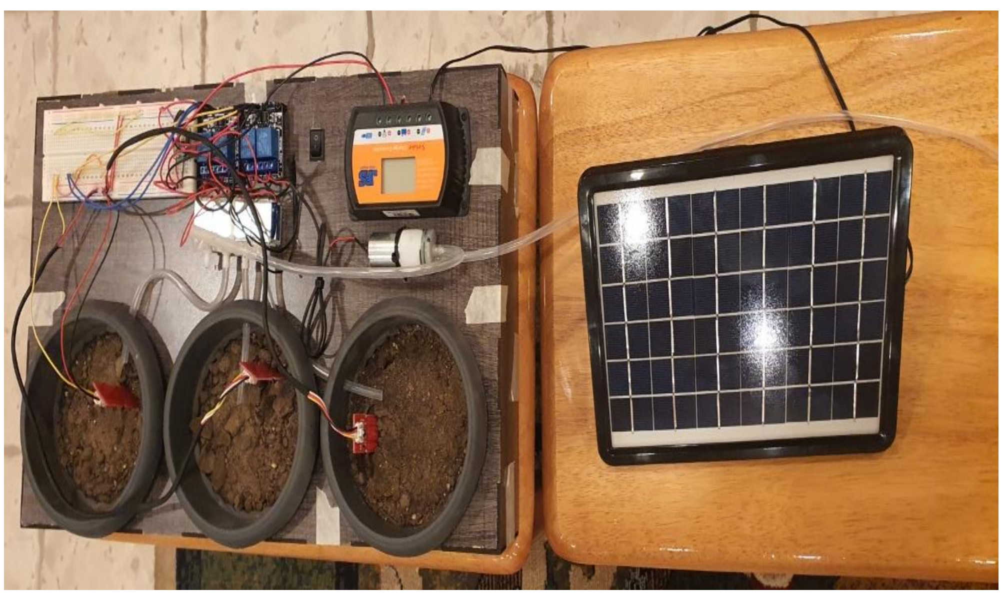

Figure 3: Smart Irrigation System

For the setup, we connect the system wires to a Solar panel or regular power source, then link it to a power consumption monitor, depending on the type of the power we either use the current

Footnote may be given with Font size 10 of Times New Roman.

sensor or not. Then we link the wires to the water pump which is linked to the piping and that will be taking the water from the tank to the pump. Then we have the relays which operate as a safety switch for the system. As for the interference we have the Arduino MKR 1010 coded to make the system operate under 8am and 6pm (Interference: 8am and 6pm) for the system to interact in checking the soil moisture if it needed irrigation (moisture<80%).

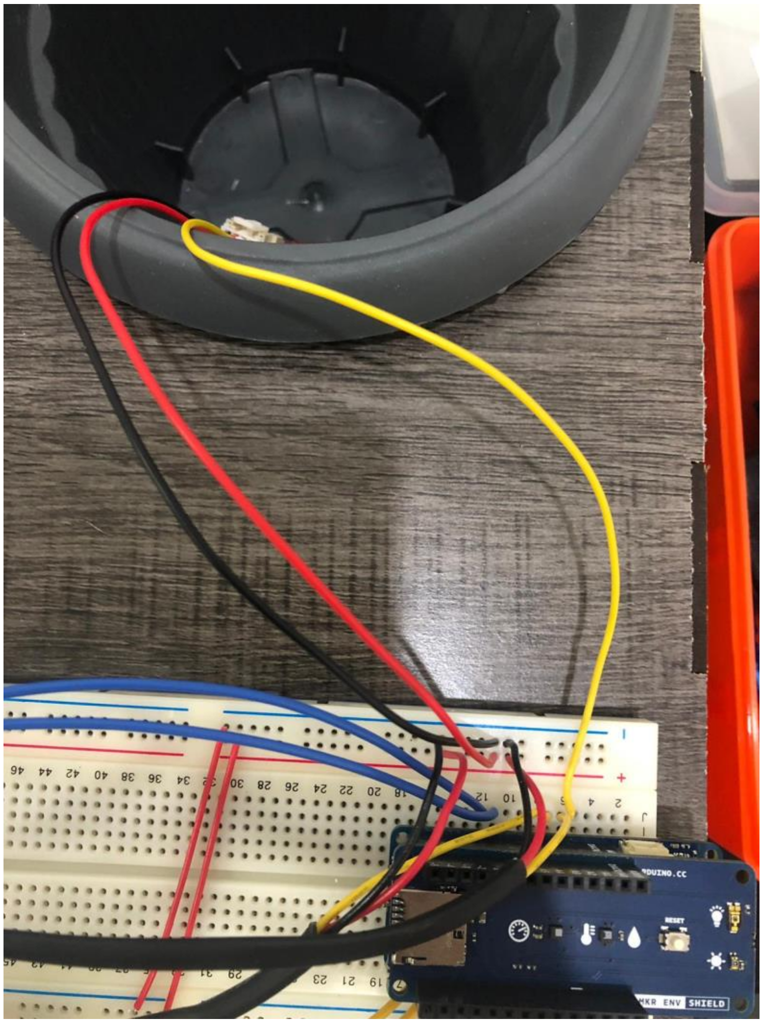

Figure 4: Arduino MKR 1010 with ENV Shield

But for the component to be operational, the water pump and valves work on 12V, while the rest of the components are operating on 5V, so we connect the voltage regulator for making the relays, sensors and micro-controllers working properly.

Footnote may be given with Font size 10 of Times New Roman.

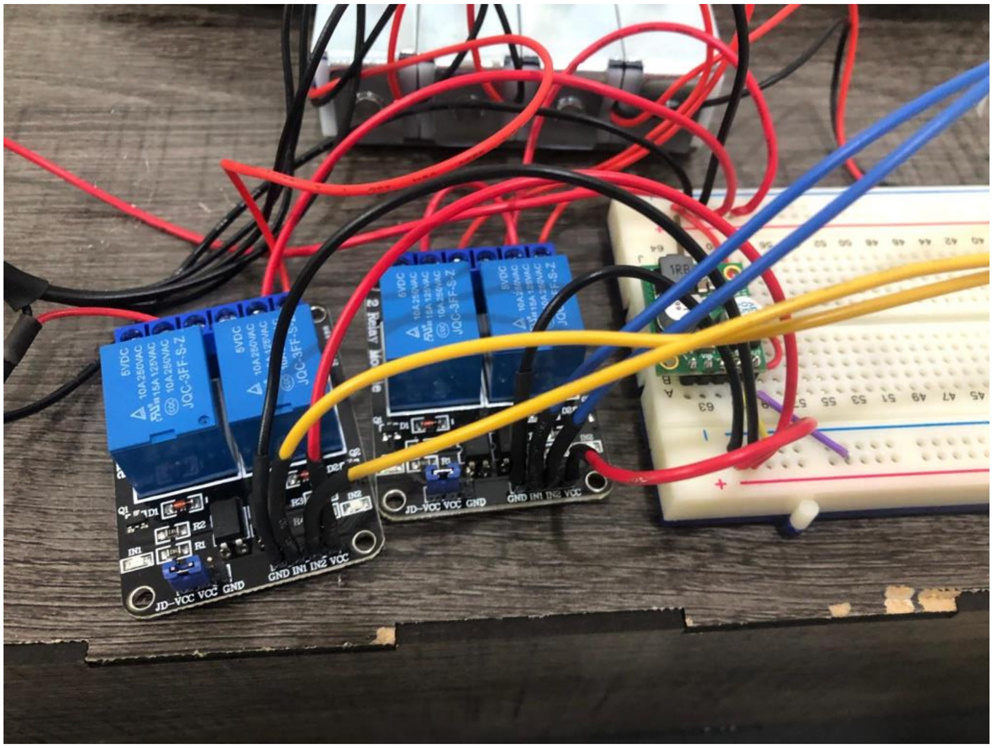
Figure 5: Step-down voltage regulator with the relay connection

4.2 Software Implementation

We mostly made our coding on the Arduino online editor due to the connection between the IOT cloud and some were on the regular software. First we must initialize our variables accordingly and name them properly such as the temperature, humidity, pressure with the integer as a type, cloud percentage of the soil moisture and also the life cycle of the soil, three integer type for the ultra violet light spectrum. And the timing for the system to be operation according to its command and the power of it. Then we used the real time clock (RTC) and we set their value manually based upon the time. And we set the date by initializing the constant of the day, month and year. The next of the software is based upon the components values which we used in the software platform based on testing the sensors on their normal state. Then created a method to setup our program based upon the data that will be given once the program has started. Then to run the program accordingly, first we had to make sure that the program is updated in a loop for it to keep up with the proper data. Now for the operation of the water valves to be under task, we came up with two timing which is the Interruption of the system, for the water to be pouring in the plants, if it was 8 am or 6 pm, these two timing are our pointers of water consumption but

Footnote may be given with Font size 10 of Times New Roman.

in the same time it will check if the soil moisture level is below 80 percent, then the plant will be given water which is found that in our research that the best moisture for the soil is to be at 80 percent. However, the reading of the sensors will not be on spot when the water is granted in the plants but it will waiting for a period of time then the reading will be taking after. Then the data is provided on the application with continues update and just by a click, operation of the system is displayed along with their values. While as for the cloud storage and analysis, the code is generated by the Arduino cloud according to the ‘Thing’ which is term used for the system at hand in IOT definition. And by that definition we used the Arduino web editor so that we can access the system and configure if needed without being near the “Thing”

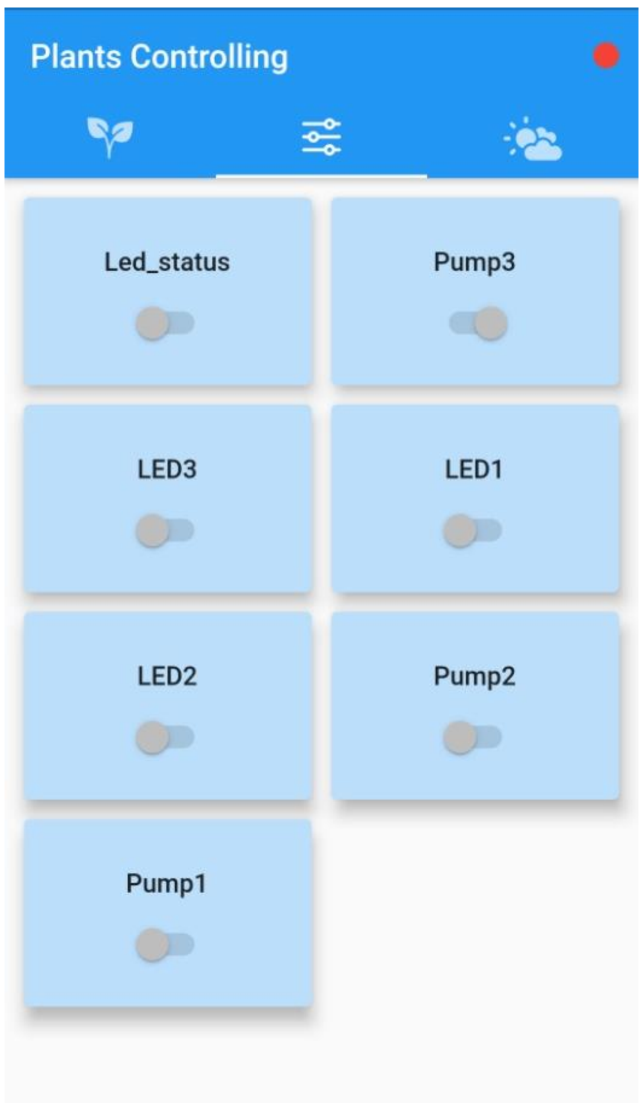

Figure 6: Application main page features

Footnote may be given with Font size 10 of Times New Roman.

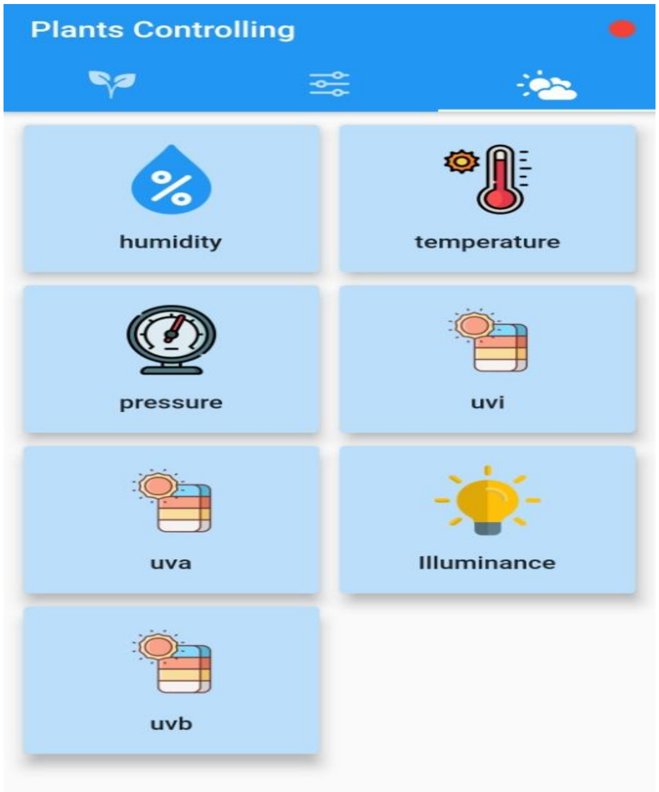

Figure 7: Application features of ENV reading

4.3 IEEE Standards

Our team had worked under the IEEE Standards for Guide for Installation, Maintenance, and Operation of Irrigation Equipment Located Near or Under Power Lines (1542-2007). Such Guide follows the industry field and irrigation equipment and we apply our smart irrigation following such pattern. Mainly this standard is for Transmission and Distribution and we deal with such application by using power to distribute water along the network based on a condition. Also, we are having two power source, either a regular 12V or alternative Solar panel that produce 12V and that is also part of the standard which they point to use an alternative power source for Transmission and Distribution (T&D). Additionally, we used IOT and that we followed the TIPPSS principles (P2933): Trust, Identity, Privacy, Protection, Safety, and Security. Our application is Arduino cloud based, which offer such principles through its features (2019-05-21). And not to forget, we followed the code of ethics of the IEEE policy 7.8: maintain the safety, health, and welfare of the public along with development practices (Part A - IEEE Policies). And

Footnote may be given with Font size 10 of Times New Roman.

the second code of ethics we followed is to expand by understanding individuals and society of the abilities and social implications of predictable and emerging technologies, including intelligent systems and by that code we made the system self-configuring and accessible via smart device with even being near the system itself.

Footnote may be given with Font size 10 of Times New Roman.

CHAPTER 5: EVALUATION

5.1 Business and marketing

When it comes to the market, the smart irrigation could cost more than the usual irrigation system but depending on the customer who is willing to conserve their plating and avoid technicality issue of the operation, then the smart irrigation is at the right price. Pay more in order to save more, and that phrase is focusing on the plant based system which even if the consumer is in a remote area, they are able to fix the issue with a single click.

5.2 Economics

As for the economic scale, Water is the main goal to reduce the amount because the system will provide proper amount to the soil while the regular irrigation system will continue to provide for a timeline. The smart irrigation could put at ease when it comes to maintenances issues because the smart irrigation system is self-configuring and self-adapting. So maintenance is not of an issue. The reason why we state the maintenance term is because it is not the price of the components that set a price, but it is their repairing service. But component wise, it is considered at the proper price and in the end, it will cost more if the coverage area is big.

5.3 Environmental

To talk about the environment, the smart irrigation will save more green life. Take into consideration when it comes to the human factor, if a person who forgot to switch the timer based on a particular season for a specific plant, the plant will die due to human error. Also, if the scheduled maintenance was forgotten, then the entire system might crash and the operation of the irrigation would fail. While for our smart irrigation, such catastrophe would be avoided if the system was installed and operated under its parameters and therefore, plants are safe in their environment. As will for the impact regarding the atmosphere, we provide minimum and proper amount of water for the soil which will save the evaporation matter

To compare our work with something similar is to bring up *OGEDENGBE* project of smart irrigation for which they stated that Farmers could waste thousands of cubic meters of water daily through Irrigation. As for our system and theirs, we will save more water quantity due to the proper installation of the smart system that is based on the soil itself, not the timing factor.

Footnote may be given with Font size 10 of Times New Roman.

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

As we have stated from what we have explained of our smart irrigation system, the aims of this project are to modify the irrigation system to another level. From the basic irrigation to smart irrigation with the relation to IOT definition to enhance its performance to a non-human interference system which will make the system flexible to self-configure itself without even being near the grid of the irrigation system. So, by a single click the irrigation system will operate under controlled circumstances which will save money by working in a more sufficient method.

6.2 Future Work

Our future work will be applying our system on Apple application rather than android Operating system and enhance the system by even making a robot that would do the maintenance in case of a malfunction in the system itself. And making our components are in a controlled environment and the irrigation network is distributed along the field.

Footnote may be given with Font size 10 of Times New Roman.

REFERENCES

[1] González-Briones, Alfonso, et al. "Intelligent Multi-Agent System for Water Reduction in Automotive Irrigation Processes." *Procedia Computer Science*, vol. 171, Mar. 2020, pp. 971–976.

[2] HEITKÄMPER, KATJA, et al. "Working Time Requirement for Different Field Irrigation Methods." *International Journal of Agricultural Management*, vol. 4, no. 2, Jan. 2015, pp. 50–54.

[3] Moya, Tolentino B. "Resilience of Irrigation Systems to Climate Variability and Change: A Review of the Adaptive Capacity of Philippine Irrigation Systems." *DLSU Business & Economics Review*, vol. 28, Apr. 2018, pp. 102–120.

[4] OGEDENGBE, T. I., et al. "Development of a Model and Sensor Based Smart Irrigation System." *Journal of Applied Sciences & Environmental Management*, vol. 24, no. 5, May 2020, pp. 909–915.

[5] Sudharshan, N., et al. "Renewable Energy Based Smart Irrigation System." *Procedia Computer Science*, vol. 165, Oct. 2019, pp. 615–623.

[6] Vij, Anneketh, et al. "IoT and Machine Learning Approaches for Automation of Farm System." Procedia Computer Science, vol. 167, Jan. 2020, pp. 1250–1257.

[7] XUE DONG. "Interpreting China's Dujiangyan Irrigation System." *Legacy (National Association for Interpretation)*, vol. 28, no. 6, Nov. 2017, pp. 16–18.

[8] https://www.pololu.com/file/0J1436/d24v50f5-step-down-voltage-regulator-dimensions.pdf

[9] https://content.arduino.cc/assets/MKRENVShieldV4.0_sch.pdf

[10] https://content.arduino.cc/assets/MKRWiFi1010V2.0_sch.pdf

Footnote may be given with Font size 10 of Times New Roman.

[11] https://www.allegromicro.com/-/media/Files/Datasheets/ACS712-Datasheet.ashx

Footnote may be given with Font size 10 of Times New Roman.

APPENDIX A

Microcontroller	SAMD21 Cortex®-M0+ 32bit low power ARM MCU (datasheet)
Radio module	u-blox NINA-W102 (datasheet)
Board Power Supply (USB/VIN)	5V
Secure Element	ATECC508 (datasheet)
Supported Battery	Li-Po Single Cell, 3.7V, 1024mAh Minimum
Circuit Operating Voltage	3.3V
Digital I/O Pins	8
PWM Pins	13 (0 .. 8, 10, 12, 18 / A3, 19 / A4)
UART	1
SPI	1
I2C	1
Analog Input Pins	7 (ADC 8/10/12 bit)
Analog Output Pins	1 (DAC 10 bit)
External Interrupts	10 (0, 1, 4, 5, 6, 7, 8,9, 16 / A1, 17 / A2)
DC Current per I/O Pin	7 mA
CPU Flash Memory	256 KB (internal)
SRAM	32 KB
EEPROM	no
Clock Speed	32.768 kHz (RTC), 48 MHz
LED_BUILTIN	6
USB	Full-Speed USB Device and embedded Host
Length	61.5 mm
Width	25 mm
Weight	32 gr.

Figure 8:Arduino MKR 1010 (SAMD21 microcontroller) specification

Footnote may be given with Font size 10 of Times New Roman.

ICs	LPS22HB TEMT6000 VEML6075
Input Voltage	3.3V
Operating Voltage	3.3V
Ranges	Pressure: 260 to 1260 hPa rH sensitivity: 0.004% rH/LSB Humidity accuracy: ± 3.5% rH, 20 to +80% rH UVA, UVB and UVBI measurment
Communication	I2C/Analog
Length	61 mm
Width	25 mm
Weight	32 gr.

Figure 9: Arduino MKR ENV Shield specification

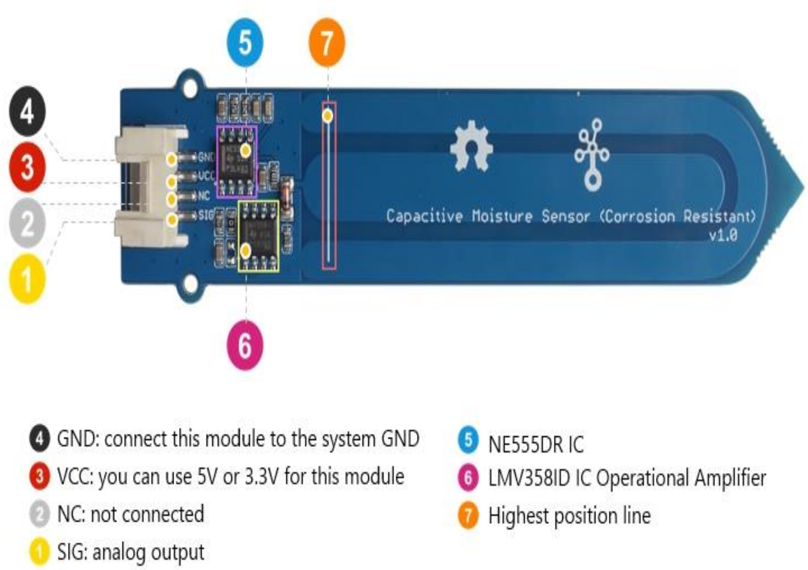

Figure 10: Grove capacitive moisture sensor specification and layout

Footnote may be given with Font size 10 of Times New Roman.

Dimensions

Size:	0.7″ × 0.8″ × 0.35″[1]
Weight:	3.0 g[1]

General specifications

Minimum operating voltage:	6 V[2]
Maximum operating voltage:	38 V
Continuous output current:	5 A[3]
Output voltage:	5 V
Reverse voltage protection?:	Y
Maximum quiescent current:	0.8 mA[4]

Figure 11: Step down voltage regulator 5V specification

- 80kHz bandwith
- 66 to 185 mV/A output sensitivity
- Low-noise analog signal path
- Device bandwith is set via the new FILTER pin
- 1.2 mΩ internal conductor resistance
- Total output error of 1.5% at TA = 25°C
- Stable output offset voltage.
- Near zero magnetic hysteresis

Figure 12: Current Sensor 712 ACS specification

Footnote may be given with Font size 10 of Times New Roman.

Code for the smart irrigation system

```
/*
  Sketch generated by the Arduino IoT Cloud Thing "Led_Project"
  https://create.arduino.cc/cloud/things/1ae5d761-4ce4-4f88-8656-
54b4bf4f07de

  Arduino IoT Cloud Properties description

  The following variables are automatically generated and updated when
changes are made to the Thing

  CloudPercentage humidity;
  float temperature;
  bool led_status;
  int pressure;
  bool pump3;
  bool lED3;
  int uvi;
  int uva;
  CloudPercentage soil_moisture3;
  bool lED1;
  float Illuminance;
  bool lED2;
  bool pump2;
  CloudPercentage soil_moisture2;
  CloudPercentage soil_moisture1;
  CloudPercentage life_cycle;
  float power_con;
  bool Pump1;
  int uvb;
  CloudPower tpower;
  CloudTime myTime;

  Properties which are marked as READ/WRITE in the Cloud Thing will
also have functions
  which are called when their values are changed from the Dashboard.
  These functions are generated with the Thing and added at the end of
this sketch.
*/
#include "thingProperties.h"
#include <Arduino_MKRENV.h>
#include <Wire.h>
#include <RTCZero.h>


RTCZero real_time_clock;
/* Change these values to set the current initial time */
const byte mseconds = 0;
```


Footnote may be given with Font size 10 of Times New Roman.

```
const byte mminutes = 1;
const byte mhours = 18;

/* Change these values to set the current initial date */
const byte mday = 3;
const byte mmonth = 6;
const byte myear = 21;


#define Moisture_1 A1 // 900 no water - 500 full water
#define Moisture_2 A2
#define Moisture_3 A3
#define my_pump 2
#define valve_1 3
#define valve_2 0
#define valve_3 1


const int AirValue1 = 900;
const int WaterValue1 = 480;
int soilMoistureValue1 = 0;
int soilmoisturepercent1 = 0;

const int AirValue2 = 315;
const int WaterValue2 = 227;
int soilMoistureValue2 = 0;
int soilmoisturepercent2 = 0;

const int AirValue3 = 892;
const int WaterValue3 = 480;
int soilMoistureValue3 = 0;
int soilmoisturepercent3 = 0;

bool Check;
int my_hour;
int gmt = 3;




void setup() {
  // Initialize serial and wait for port to open:
  Serial.begin(9600);
  real_time_clock.begin();
  // This delay gives the chance to wait for a Serial Monitor without
blocking if none is found
  Serial.println("welcome");
  delay(1500);
  pinMode(my_pump, OUTPUT);
  pinMode(valve_1, OUTPUT);
  pinMode(valve_2, OUTPUT);
  pinMode(valve_3, OUTPUT);
```


Footnote may be given with Font size 10 of Times New Roman.

```
  digitalWrite(my_pump, HIGH);
  digitalWrite(valve_1, HIGH);
  digitalWrite(valve_2, HIGH);
  digitalWrite(valve_3, HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN, HIGH);
  // Defined in thingProperties.h
  initProperties();
  delay(5000);
  ArduinoCloud.begin(ArduinoIoTPreferredConnection);
  delay(7000);
  setDebugMessageLevel(2);
  ArduinoCloud.printDebugInfo();

  //real_time_clock.setEpoch(1451606400); // Jan 1, 2016
  real_time_clock.setHours(mhours);
  real_time_clock.setMinutes(mminutes);
  real_time_clock.setSeconds(mseconds);

  // Set the date
  real_time_clock.setDay(mday);
  real_time_clock.setMonth(mmonth);
  real_time_clock.setYear(myear);
delay(1000);
  if (!ENV.begin()) {
    Serial.println("Failed to initialize MKR ENV shield!");
    while (1);
  }
  digitalWrite(LED_BUILTIN, HIGH);
  delay(5000);
  Serial.println("started");
}

void loop() {
  ArduinoCloud.update();
  humidity = int(ENV.readHumidity());
  Illuminance = int(ENV.readIlluminance());
  pressure = int(ENV.readPressure());
  temperature = int(ENV.readTemperature());
  uva = int(ENV.readUVA());
  uvb = int(ENV.readUVB());
  uvi = int(ENV.readUVIndex());


/*  print2digits(real_time_clock.getDay());
  Serial.print("/");
  print2digits(real_time_clock.getMonth());
  Serial.print("/");
  print2digits(real_time_clock.getYear());
  Serial.print(" ");

  // ...and time
```


Footnote may be given with Font size 10 of Times New Roman.

```
  print2digits(real_time_clock.getHours());
  Serial.print(":");
  print2digits(real_time_clock.getMinutes());
  Serial.print(":");
  print2digits(real_time_clock.getSeconds());

  Serial.println();*/
/*  my_hour = real_time_clock.getHours();
  Serial.println(my_hour);*/


  MeasureM1();
  MeasureM2();
  MeasureM3();

  /*if ((my_hour == 8) && (Check == 0) ) {
    //digitalWrite(my_pump, LOW);
    if (soil_moisture1 < 80) {
      digitalWrite(valve_1, LOW);
      delay(2000);
      digitalWrite(valve_1, HIGH);
    }
    if (soil_moisture2 < 80) {
      digitalWrite(valve_2, LOW);
      delay(2000);
      digitalWrite(valve_2, HIGH);
    }
    if (soil_moisture3 < 80) {
      digitalWrite(valve_3, LOW);
      delay(2000);
      digitalWrite(valve_3, HIGH);
    }
    Check = 1;
    }
    if ((myrtc.getHours() + 1 == 18) && (Check == 1)) {
    if (soil_moisture1 < 80) {
      digitalWrite(valve_1, LOW);
      delay(2000);
      digitalWrite(valve_1, HIGH);
    }
    if (soil_moisture2 < 80) {
      digitalWrite(valve_2, LOW);
      delay(2000);
      digitalWrite(valve_2, HIGH);
    }
    if (soil_moisture3 < 80) {
      digitalWrite(valve_3, LOW);
      delay(2000);
      digitalWrite(valve_3, HIGH);
    }
    Check = 0;
    }*/
```


Footnote may be given with Font size 10 of Times New Roman.

```
  delay(50);
}


void MeasureM1() {
  soilMoistureValue1 = analogRead(Moisture_1);  //put Sensor insert
into soil
  soilmoisturepercent1 = map(soilMoistureValue1, AirValue1,
WaterValue1, 0, 100);
  soil_moisture1 = soilmoisturepercent1;
  if (soilmoisturepercent1 >= 100)
  {
    Serial.println("M1 100 %");
  }
  else if (soilmoisturepercent1 <= 0)
  {
    Serial.println("M1 0 %");
  }
  else if (soilmoisturepercent1 > 0 && soilmoisturepercent1 < 100)
  {
    Serial.print(soilmoisturepercent1);
    Serial.println("% M1");

  }
}

void MeasureM2() {
  soilMoistureValue2 = analogRead(Moisture_2);  //put Sensor insert
into soil
  soilmoisturepercent2 = map(soilMoistureValue2, AirValue2,
WaterValue2, 0, 100);
  soil_moisture2 = soilmoisturepercent2;
  if (soilmoisturepercent2 >= 100)
  {
    Serial.println("M2 100 %");
  }
  else if (soilmoisturepercent2 <= 0)
  {
    Serial.println("M2 0 %");
  }
  else if (soilmoisturepercent2 > 0 && soilmoisturepercent2 < 100)
  {
    Serial.print(soilmoisturepercent2);
    Serial.println("% M2");

  }
}

void MeasureM3() {
  soilMoistureValue3 = analogRead(Moisture_3);  //put Sensor insert
into soil
```


Footnote may be given with Font size 10 of Times New Roman.

```
  soilmoisturepercent3 = map(soilMoistureValue3, AirValue3,
WaterValue3, 0, 100);
  soil_moisture3 = soilmoisturepercent3;
  if (soilmoisturepercent3 >= 100)
  {
    Serial.println("M3 100 %");
  }
  else if (soilmoisturepercent3 <= 0)
  {
    Serial.println("M3 0 %");
  }
  else if (soilmoisturepercent3 > 0 && soilmoisturepercent3 < 100)
  {
    Serial.print(soilmoisturepercent3);
    Serial.println("% M3");
  }
}


void onLedStatusChange() {

}
void onServoPosChange() {
  // Do something
}
void onPushButtonChange() {
  // Do something
}
void onPump1Change() {
  // Do something
}
void onPump2Change() {
  // Do something
}
void onPump3Change() {
  // Do something
}
void onLED1Change() {
  // Do something
}
void onLED2Change() {
  // Do something
}
void onLED3Change() {
  // Do something
}

void print2digits(int number) {
  if (number < 10) {
    Serial.print("0"); // print a 0 before if the number is < than 10
  }
  Serial.print(number);
```


Footnote may be given with Font size 10 of Times New Roman.

}

Footnote may be given with Font size 10 of Times New Roman.