Bachelor of Engineering in Computer Engineering

H3024 Project 2

Automated Plant Watering System

Author: Uzair Kamal

Supervisor: Dave Peyton

Date:

Department of Engineering

School of Informatics & Engineering

Institute of Technology, Blanchardstown

Dublin 15.

Declaration

The material contained in this assignment is the author's original work, except where work quoted is duly acknowledged in the text. No aspect of this assignment has been previously submitted in any other unit or course.

Signed: Date

Abstract

In daily operation related to watering the plants are the most important cultural practice and the most labour-intensive task. No matter whichever weather it is, either too hot and cold or too dry and wet it is very crucial to control the amount of water reaches to the plants. So, I will be effective to use an idea of automatic plant watering system which waters plants when they need it. An important aspect of this project is that: "when and how much to water". To reduce manual activities for the human to watering plant, an idea of plant watering system is adopted. The method employed to monitor the soil moisture level continuously and to decide whether watering is needed or not, and how much water is needed in plant's soil. This project can be grouped into subsystems such as; power supply, relays, solenoid valve, Arduino GSM shield, Soil moisture sensor and LCD.

Essentially, system is design and programmed in such way that soil moisture sensor senses the moisture level of plants at particular instance of time, if moisture level of sensor is less than the specified value of threshold which is predefined according to the particular plant's water need then the desired amount of water is supplied till it reaches to the predefined threshold value.

System reports its current states and sends the reminder message about watering plants and to add water to the tank. All this notification can be done by using Arduino GSM shield.

Acknowledgment

I would like to thank **Mr. Dave Peyton** who is my supervisor in this project. He was there to guide me for the project and keep me on the right track which was very important for me to be focused on my project.

Table of Content

Abbreviations, Symbols, Acronyms

LCD Liquid Crystal Display

GSM Global System for Mobile Communication

IDE Integrated Development Environment

RTD Resistive Temperature Sensor

PTC Positive Temperature Coefficients

SPI Serial Peripheral Interface

°C Centigrade

USB Universal Serial Bus

IMEI International Mobile Equipment Identity

SIM Subscriber Identity Module.

GPRS Global Packet Radio Service.

ISDN Integrated Services Digital Network

PSTN Public switched telephone network

TCP Transmission Control Protocol

UDP User Datagram Protocol

HTTP Hypertext Transfer Protocol

ADC Analogue to Digital Conversion

UART Universal Asynchronous Receiver-Transmitter

List of Figure Captions

List of Tables

Product Specification

Requirement ID	SRS-GSM-001
Title	GSM Module
Description	System includes the GSM module, which sends alert SMS to the recipient and receive a SMS from the user.
Version	Version 1.0

Requirement ID	SRS-Microcontroller -001
Title	ATmega328p
Description	System includes the microcontroller which usually comes with Arduino Uno. this microcontroller reads the reading from the sensor and controls the overall system.
Version	Version 1.0

Requirement ID	SRS-Temperature and Humidity-001
Title	DHT11
Description	System includes the temperature and humidity sensor, which keeps track of the current temperature and humidity values from the surrounding and sends the reading back to microcontroller.
Version	Version 1.0

Requirement ID	SRS-Moisture-001
Title	Grove Soil Moisture Sensor
Description	System includes the soil moisture sensor, which takes the reading from the moisture of the soil and sends the reading back to microcontroller.
Version	Version 1.0

Requirement ID	SRS-LCD-001
Title	Hitachi 16x2 LCD
Description	System includes the LCD interface for the user, which displays the reading taken by the different sensors in the system.
Version	Version 1.0

Chapter 1: Introduction

1.1. Aim of the project:

Since nowadays, in the age of advanced technology and electronics, the life style of the human should me smart, simpler, easier and much more convenient. So, therefore; there is a need for many automated systems in human's daily life routine to reduce their daily activities and jobs. Here an idea of one such system named as automatic plant watering system is very useful. As many people are facing a lot of problem watering the plants in the garden, especially when they away from the home. This model uses sensor technologies with microcontroller in order to make a smart switching device to help millions of people. (Agarwal, 2015), (Duzic&Dumic, 2017)

In its most basic form, system is programmed in such a way that soil moisture sensor which senses the moisture level from the plant at particular instance of time, if moisture level of the sensor is less than the specified value of threshold which is predefined according to the particular plant than the desired amount of water is supplied to plant till it's moisture level reaches to the predefined threshold value. System involves humidity and temperature sensor which keep tracks the current atmosphere of the system and has an influence when watering happens. Solenoid valve will control the water flow in the system, when Arduino reads value from moisture sensor it triggers the solenoid valve according to the desired condition. In addition, system reports its current states and sends the reminder message about watering plants and gets SMS from the recipient. All this notification can be done by using Arduino GSM shield.

Block diagram of the system is shown below in *figure1*:

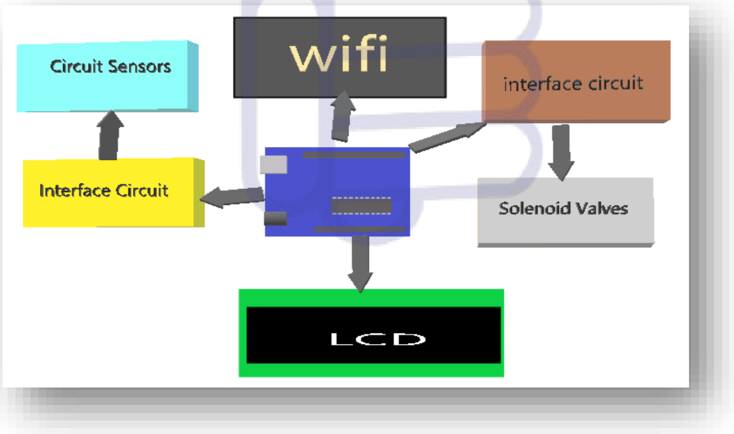

Figure 1 system block diagram

1.2. Motivation:

The motivation for this project came from, that plants are very beneficial to all human beings in many aspects. They produced oxygen and helps in keeping the environmental healthy. So, many people like to have plants in their homes and backyards. These plants are dependent on breeding conventionally – for instance provide the right amount of water supply to sustain life and growth. Many people forget to water their plant on a busy schedule of day and due to that many plants suffers disorder and ultimately died. Apart from that another big problem in the modern society is the shortage of water and the unplanned use of water inadvertently results in wastage of water. It is a big task to utilize water supply/ resources in a proper way thus, a system is required to handle this task automatically. In automatic plant watering system, the most momentous advantage is that water is only supplied when pre-set threshold value of the soil moisture sensor goes below. This saves lots of water even in the bigger irrigation systems. (D Divani, 2016)

All those problem can be rectified by using the automatic plant watering system in which watering can take place only when the threshold value from the moisture sensor matches with the pre-set value. Secondly, this system reduces the human effort and give them ease rather than watering the plants manually.

According to (Naidu, 2011)

"Due to unplanned use of water, significant amount of water is wasted.

For this purpose, we use Automatic Watering system."

1.3. Layout of Report:

Chapter 2 (Background Research):

A detailed literature review was conducted on an Automated plant watering system itself and related areas. A fundamental knowledge was obtained by the author on technologies and components which are utilised by the system later and how these constituents interface with one another. This chapter will outline the functions of the system and different sensor technologies including GSM module. This section of the report is basically the most important part of the report as it's based on the key aspect of the system.

Chapter 3 (Project Design):

This chapter goes into details that how the hardware and software design of the system was completed. It also includes the detail of why different components were chosen for the project how they perform in terms of functionality to build a fully functioning system. It also goes into further details of how the hardware design of the system was build which includes the different schematics and Veroboard design for the system. In the software design section, different approaches were adopted to design the software flow which can test the hardware. The last part of this section goes through different algorithm design of the system which allows easy flow of the system and makes decision making easy.

Chapter 4 (Construction and Testing):

This chapter of the report goes into the details of how the hardware design of the system was constructed and goes into the detail of how different approaches were adopted to test the hardware and software design of the system and to ensure that the system is doing what is supposed to do.

Chapter 5 (Safety and Ethical Consideration):

This chapter goes through the brief discussion of what safety and ethical consideration must be kept in mind while doing these types of project when surrounded by the heavy types of machinery.

Chapter 6 (Conclusion and Recommendations):

In this chapter, of the report author goes through the how the project was completed and what problems that author faced during the project and how all those problems solved. Any recommendation made for the future work which makes the system more reliable and efficient for the end users.

Chapter 2: Background Research

2.1. Background of the System:

It has been studied in the school from the science's books that the plants are very imperative for all the humanity in many aspects. As they keep the environmental clean by producing fresh oxygen time to time. Automatic plant watering system have been seen becoming much more with the rise in the everyday objects being connected to the advanced technologies, these systems are implemented at a growing rate. Places like homes as well as on industrial levels. The main use of these systems is efficiency and easy to use.

Plant watering system provides the ability to plant lovers to take of their home plant while they are away – through the use of efficient and reliable components such as different types of sensor technologies.

There are several different/uncomplicated types of indoor plant watering system, depending on the level of automation needed.

Plant Watering Globes and Spikes – it is one of the successful indoor system. The basic premise of the plant watering spike is the fact that it is a reservoir that basically waters the plant through capillary action. And the spike is made of unglazed ceramic. A small plastic tube goes from the spike to the nearby reservoir, and when soil moisture of the soil goes low it automatically draws water through the device into the soil. This basic system can be shown below in *figure 2*

Figure 2 plant watering Globes and Spikes (pots, p, 2016)

Indoor Drip Watering System – this system can take the guesswork out of watering the plant. As it is not necessary to remember when was the reservoir refilled last time on self-contained pots. Instead it is possible to run a drip system to those same pots and run a timer. However, for minimal maintenance watering this can be ideal. But one strategy to keep in mind when using those watering system is that there is no way for the dirt in the pot itself to regulate the water flow, as it does with most self-watering pots. Timer will turn on, and drips water for certain amount, then turns off. One disadvantage of this is that if plant do not require much water, there is a chance of over watering them.

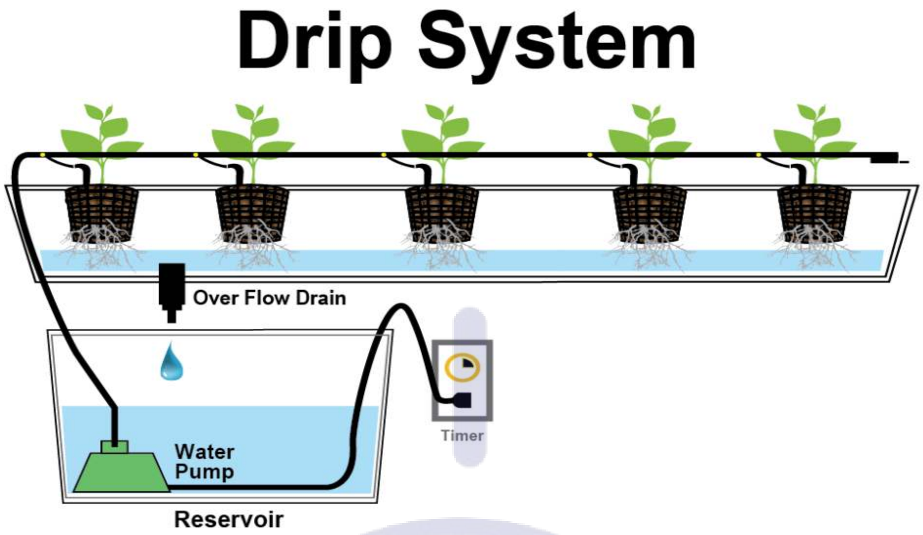

Figure 3 Drip watering System (Urban Organic Yield, n.d)

However, there are more different types of plant watering system including self-watering pots and container, self-watering hanging baskets, outdoor drip irrigation system and so on. As It can be seen from all those system which is described above is not very ideal. So, it is necessary to use sensor technologies here. The sensor provides real-time data that is used to run the overall system efficiently. Which keeps track the real-time values and update the system frequently. In addition, the key technologies behind the smart automated plant watering system will be discussed through the extensive research on them. These technologies include the frequently update of system according to the sensing technology used to allow the system to real time events such as soil moisture level of a plants goes below the threshold value and so on.

Some of the important smart watering system is illustrated below:

Arduino Based Automated Plant Watering System:

Block Diagram:

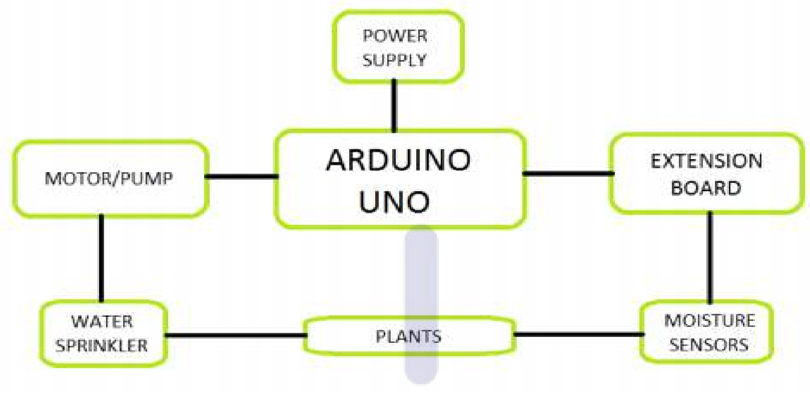

Figure 4 Automatic plant watering block diagram (Devika S, 2017)

According to this system there are two functional components in this project i.e. Moisture sensor and motor/water pump. In its most basic form moisture sensor senses level of the soil moisture. Then motor/water pump supplies water to plants.

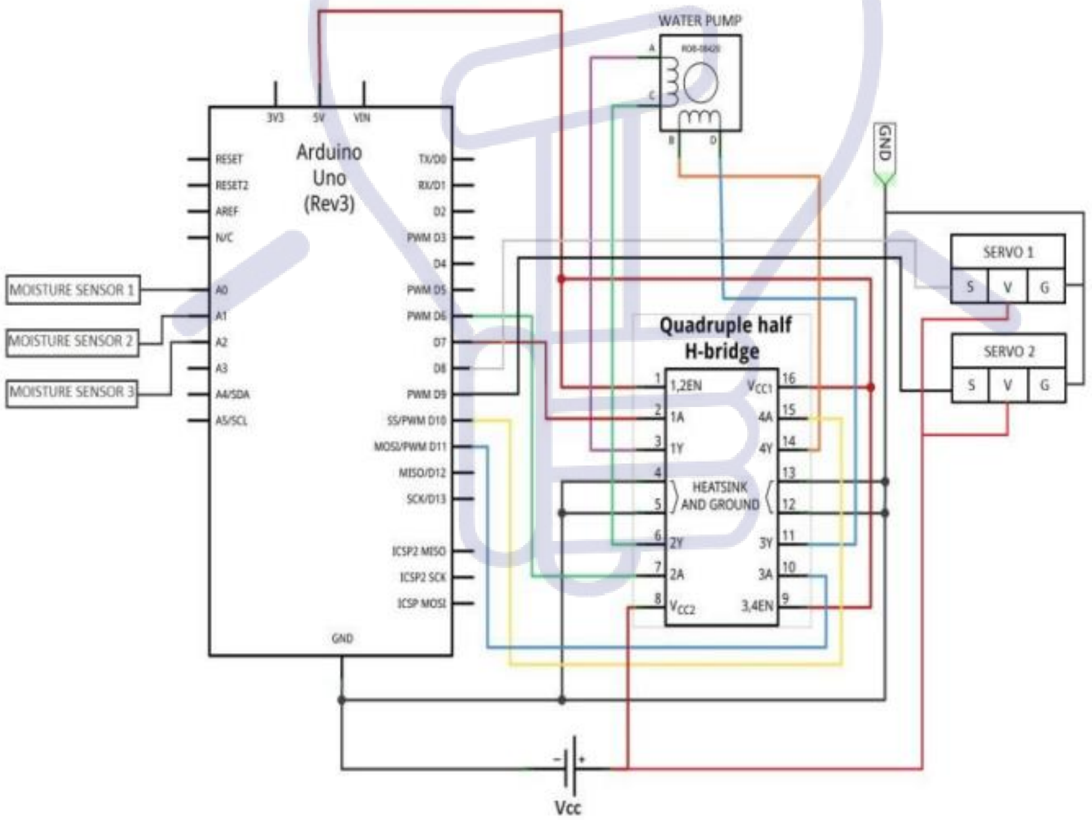

Figure 5 Schematic of the system (Devika S, 2017)

Schematic above in *figure 5* describes the overall behaviour of the system. Project uses Arduino Uno to controls the motor. It consists of H-bridge which controls the flow of servo motor i.e. clock or anti clock direction. Moisture sensor measures the level of soil and sends the signal to Arduino then Arduino will open the servo motor if watering is required. Then the motor/water pump supplies water to the plants until the desired moisture level is reached.

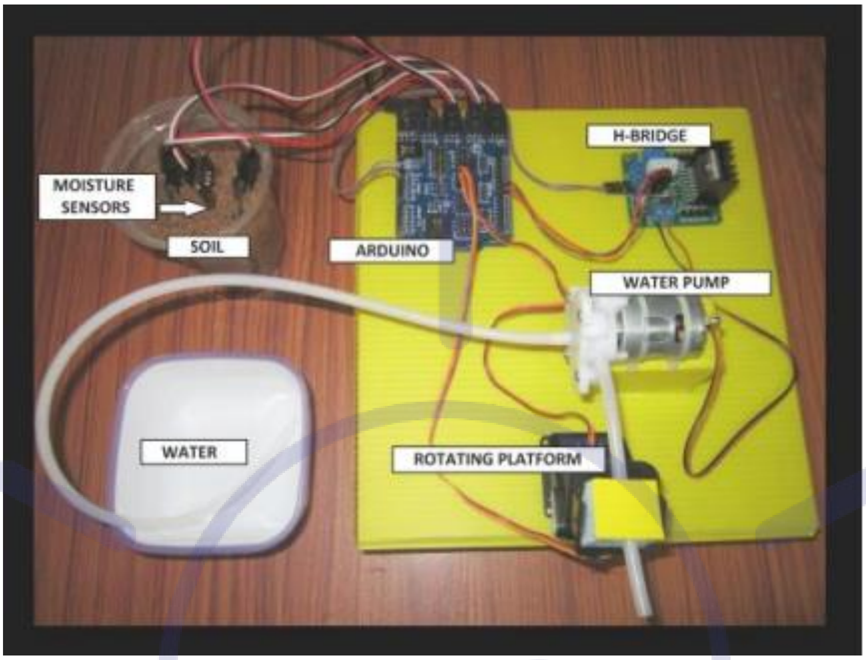

Figure 6 prototype of the system (Devika S, 2017)

Form the prototype above in *figure 6* moisture sensor senses the moisture level and sends the signal to Arduino and then Arduino opens water pump with the help of H-bridge and waters the particular plant. This is done by using Arduino IDE software. (Devika S, 2017)

Arduino Based Automated Plant Watering System with Message Alert:

Block diagram of the system is shown below in *figure 7*

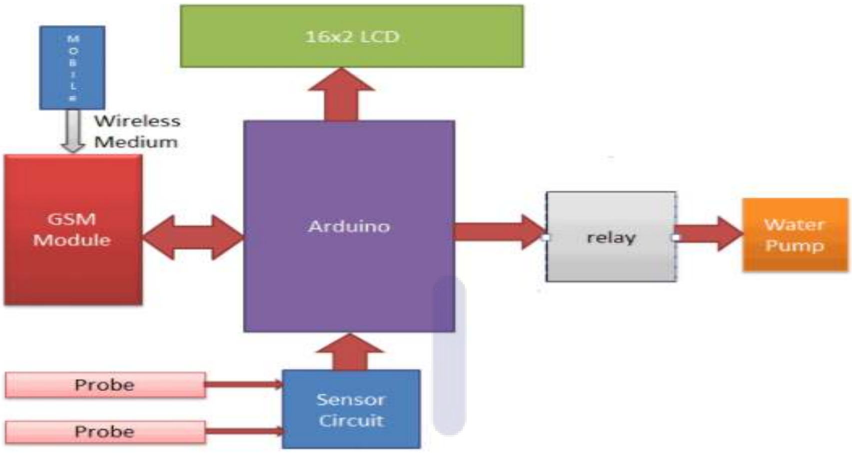

Figure 7 Block Diagram (Scribd, 2018)

Block diagram above gives the short presentation on what the framework will do in this specific undertaking. Arduino is utilized for controlling entire procedure of this Programmed Plant Watering System. The yield of soil sensor circuit is straight forwardly associated with a computerized stick of Arduino (digital pins). An idea for GSM module in this project is to notify the user by sending SMS (Scribd, 2018). Schematic of the overall system is shown below in *figure 8*

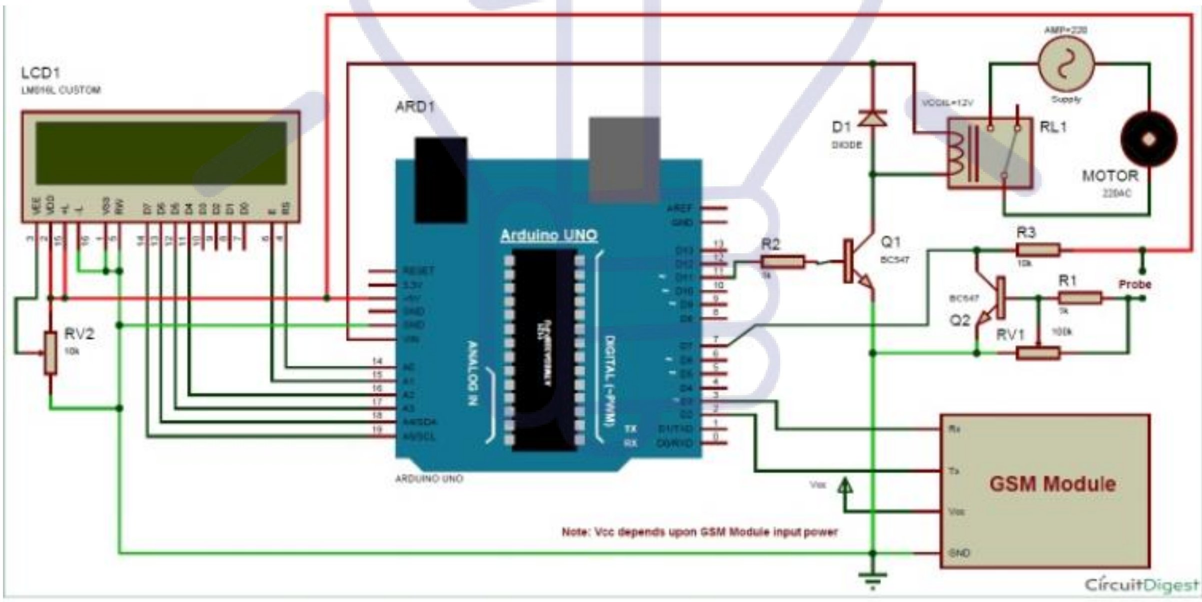

Figure 8 Schematic of the System (Scribd, 2018)

2.2. Sensor Technology:

"Sensor is a device which provides a usable output in respond to a specified measurement" (Engineering.nyu.edu, n.d.)

Sensor is a device that perceives and reacts to some sort of contribution from the physical environment. The specific data could be light, heat, motion, moisture, pressure or any of an incredible number of other natural marvels. The yield is general a banner that is changed over to understandable show at the sensor location or transmitted electronically completed a framework for examining or further taking care of.

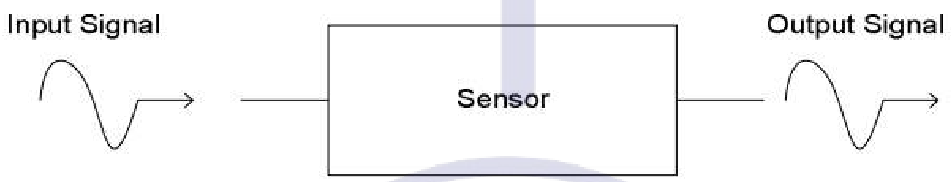

Figure 9 Sensor Block diagram (Engineering.nyu.edu, n.d)

"Without the use of the Sensors, there would be no automation !!"

(Engineering.nyu.edu)

Sensors obtains a physical amount and changes over it onto a flag reasonable for handling. These days sensor change over estimation of physical wonders into electrical flag. Sensors are inescapable. They are implanted in our bodies, automobiles, plans, cell phones, radios, chemical plants, mechanical plants and incalculable different applications. the graph beneath in *figure 10* indicates how Sensor innovation impacts day by day life routine and affects each part of life.

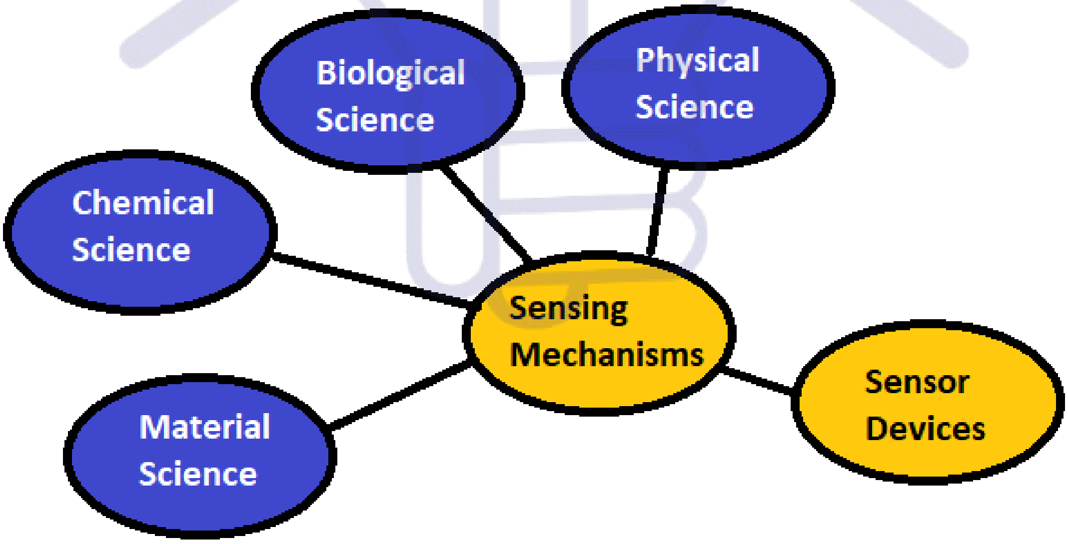

Figure 10 Sensors Importunate

The noteworthiness of sensor innovation is always developing. Sensors permits checking the surroundings in ways that nobody could scarcely envision a couple of years back. New sensor applications are being distinguished everyday which widens the extent of the innovation and extends its effect on regular day to day existence. (Discover Sensors, n.d.)

The figure underneath indicates where the sensors is utilized as a part of day by day life schedule in *figure 11*

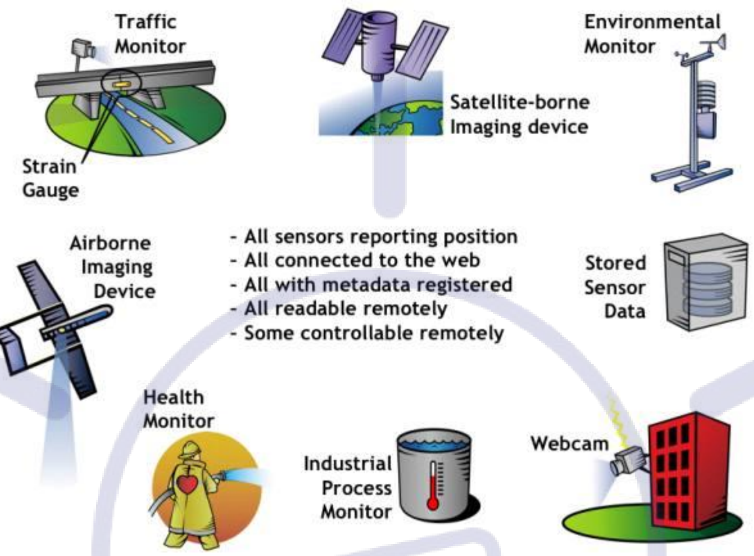

Figure 11 Use Of Sensors (opengespatial.org,n.d.)

2.2.1 Temperature and Humidity Sensors:

Temperature Sensor: Temperature Sensor measure the amount of heat that is generated by an object or system, permitting to "sense" or distinguish any physical change to that temperature delivering either an analogue or digital output.

There are many different types of temperature sensor available and all have distinctive qualities relying on their real application. A temperature sensor consists of two basic physical types:

- Contact Temperature Sensor Types.
- Non-Contact Temperature Sensor Types.

Contact Temperature Sensor Types: These sorts of temperature sensor are required to be in physical contact with the object being detected and utilize conduction to screen changes in temperature. They can be utilized to identify solids, liquid or gasses over an extensive variety of temperatures.

Non-Contact Temperature Sensor Types: These sorts of temperature sensor utilize convection and radiation to monitor changes in temperature. They can be utilized to distinguish liquids and gases that emanate brilliant vitality as heat rises and cold settles to the base in convection currents or recognize the radiant energy being transmitted from an object as infra-red radiation *from* (Electronic tutorials, n.d).

These two basic types of Temperature sensor can also be subdivided into the following:

Bi-Metallic Thermostat:

Bi-metallic thermostat is used as an electrical switch or as a mechanical way of operating an electrical switch in thermostatic controls and are used extensively to control hot water heating elements in boilers, hot water storage tanks as well as in vehicle radiator cooling system *from* (Electronic tutorials, n.d). Figure underneath *figure12* shows how bi-metallic thermostats works.

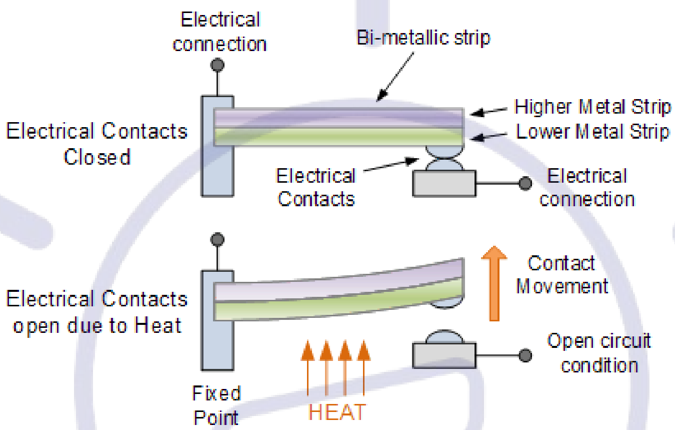

Figure 12 Bi-metallic Thermostat (Electronics tutorilas, n.d.)

Thermistor:

Thermistor is another type of temperature sensor. A thermistor is a special type of resistor which changes its physical resistance when disclosed to changes in temperature.

"the relation between thermistor temperature and its resistance is highly dependent upon the material from which it's composed" from (Omega.com , n.d)

Figure 13 Thermistor (Factory F , 2018)

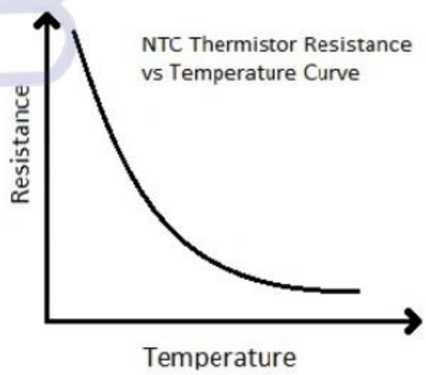

Figure 14 Resistance Vs Temperature curve (Khatri P , n.d.)

Resistive Temperature Detectors (RTD):

Resistive Temperature Detector sensor is another type of electrical resistance temperature sensor. RTD have positive temperature coefficients (PTC) but, dissimilar to the thermistor their output is extremely direct creating exceptionally precise estimations of temperature.

However, they have exceptionally poor thermal sensitivity, that is an adjustment in temperature only produce a very small output change, for instance, 1Ω/oC *from* (Electronic tutorials, n.d).

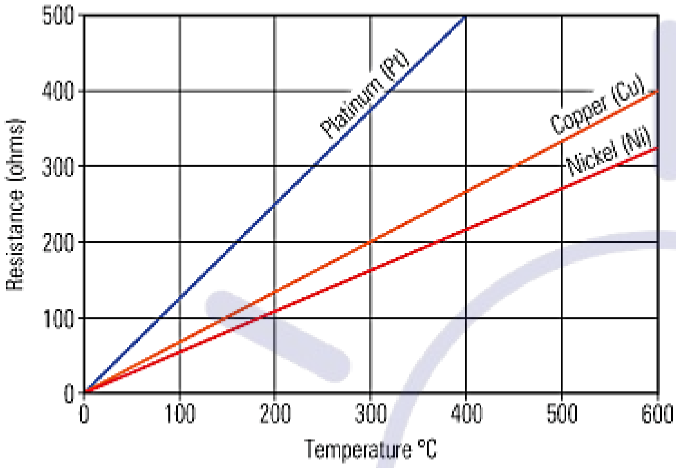

Figure 16 Resistance Vs Temperature in RTD (source: Pointing.spiraxsarco.com)

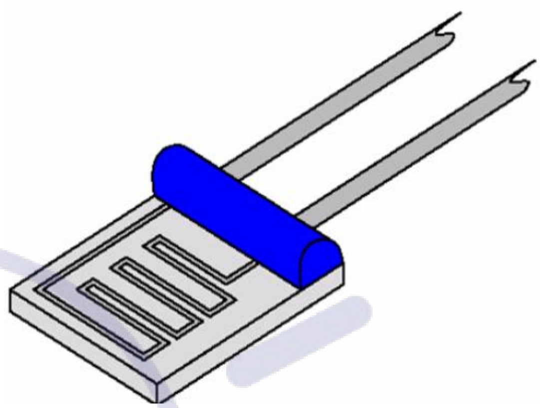

Figure 15 Resistive Temperature Detectors (source: Commons.wikimedia.org)

There are diverse types of the RTD's sensors are made from Platinum (Pt), Copper (Cu) and Nickel (Ni). And their graph; resistance against temperature can be seen in *figure 15*. Standard RTD element specifications such as; Element material, Resistance, Temperature coefficient, Operating Range and available accuracies are shown underneath in *table 1*.

Table 1 RTD Specifications (Source: thermos-kinetics.com)

Standard RTD Element Specifications				
Element Material *	Resistance at 0°C	Temperature Coefficient	Operatine Range **	Available Accuracies at 0°C
Platinum	100 Ohm	0.00385	-200°C to 600°C	±0.5% ±0.1% ±0.6% ±0.01%
Platinum	100 Ohm	0.00391	-200°C to 600°C	±0.1% ±0.6%
Copper	10 Ohm	0.00427	-200°C to 204°C	±0.2% ±0.5%
Nickel	120 Ohm	0.00672	-200°C to 204°C	±0.3% ±0.5%

Thermocouples:

"A thermocouple is a simple, robust and cost-efficient temperature sensor used in a wide range of temperature measurement processes". It comprises of two divergent metal wires, joined toward one side. At the point when appropriately arranged, thermocouples can provide measurements over a wide range of temperatures.

Known for their adaptability as temperature sensors, as they are manufactured in a variety of styles, such as thermocouple probes, thermocouple probes with connectors etc. they are commonly used in wide range of models and technical specifications. *from (Omega.com, n.d).*

figure underneath *figure 17;* shows how simple thermocouple circuit works,

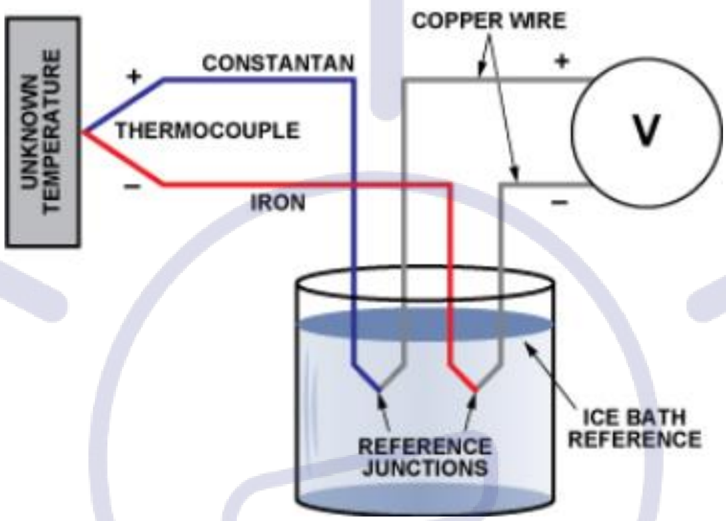

Figure 17 Basic thermocouple Circuit (Source: Duff, M and Towey, J, 2010)

Sensor ICs:

There is a wide assortment of temperature sensor ICs that are accessible to improve the broadest conceivable scope of temperature monitoring challenges. These silicon temperature sensors differ fundamentally from the previously mentioned types in two or three imperative ways. The first is operating temperature range. A temperature sensor IC can work over the ostensible IC temperature scope of - 55°C to +150°C. the second major difference is functionality.

Diverse types of temperature sensors which are used as an integrated circuit are describe below:

LM35:

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the °C temperature. These devices do not require any external calibration or trimming to provide typical accuracies of $\pm\frac{1}{4}$°C at room temperature and $\pm\frac{3}{4}$°C over a full - 55°C to +150°C temperature range. Diagrams underneath will show LM35 temperature sensor in *figure 18* and the circuit diagram of LM35 in *figure 19.*

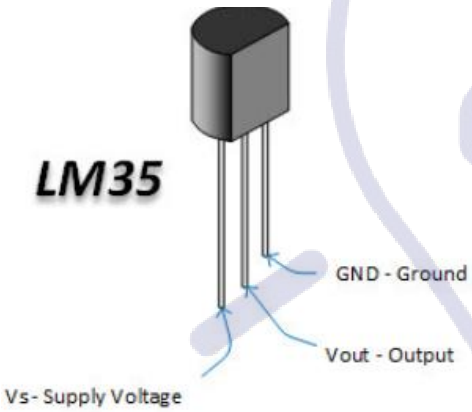

Figure 18 LM35 Temperature Sensor (Source: Agarwal, T, n.d)

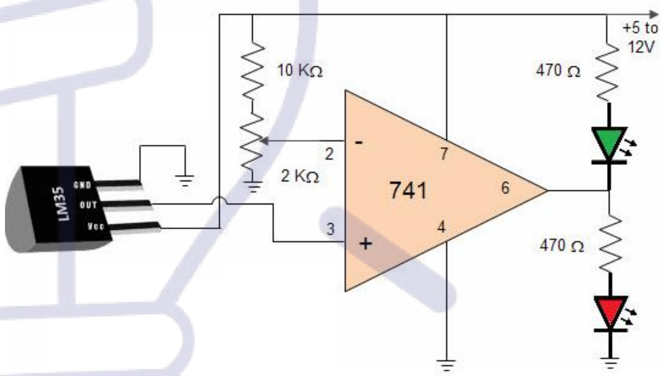

Figure 19 LM35 Circuit Diagram (Source: Agarwal, T, n.d)

Digital Temperature Sensor (ADT7301):

These sensors have a complete temperature monitoring system available in SOT-23 and MSOP packages. It contains a band gap temperature sensor and 13-bit ADC to screen and digitize the temperature perusing to a determination of 0.03125 °C. they have a malleable serial interface that allows easy interfacing to most microcontrollers. They have a wide supply voltage range, low supply current, and SPI-compatible interface make it perfect for an assortment of uses including personal computers, office gear, automotive, and local apparatuses. They are rated for operation over -40°C to +150°C temperature range.

Figure 21 Digital Temperature Sensor(Source: Digital temperature Sensor, n.d.)

Figure 20 Sensor's Pin Out (Source: Digital temperature Sensor, n.d.)

Table underneath *table2* shows the functional description of the Digital Temperature Sensor.

Table 2 Pin Functions Description (Source: Digital temperature Sensor, n.d.)

SOT-23 Pin No.	MSOP Pin No.	Mnemonic	Description
1	7	GND	Analog and Digital Ground.
2	6	DIN	Serial Data Input. Serial data to be loaded to the part's control register is provided on this input. Data is clocked into the control register on the rising edge of SCLK.
3	5	V_{DD}	Positive Supply Voltage, 2.7 V to 5.25 V.
4	4	SCLK	Serial Clock Input. This is the clock input for the serial port. The serial clock is used to clock data out of the ADT7301's temperature value register and to clock data into the ADT7301's control register.
5	3	$\overline{CS}$	Chip Select Input. Logic input. The device is selected when this input is low. The SCLK input is disabled when this pin is high.
6	2	DOUT	Serial Data Output. Logic output. Data is clocked out of the temperature value register at this pin. Data is clocked out on the falling edge of SCLK.
	1, 8	NC	No Connect.

DHT11 and DHT22:

These DHT sensors is a basic, low cost digital temperature and humidity sensor. they use a capacitive humidity sensor and a thermistor to gauge the encompassing air, and releases a computerized motion on the information pin (no analogue input pins needed). These sensors are genuinely easy to utilize, but requires careful timing to grab data. The only real flaw of these sensors is it gets new data once every 2 seconds.

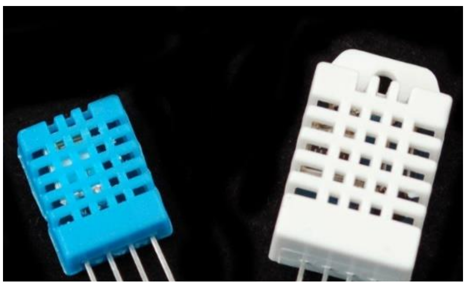

Figure 22 dht11 and dht22 (Source: Learn.adafruit.com, 2012)

Humidity Sensors: A humidity sensor senses, measures and reports the relative humidity in the air. It therefore measures both moisture and temperature in the air. The hotter the air temperature is, the more moisture it can hold. Humidity/dew sensors utilize capacitive estimation, which depends on electrical capacitance. Electrical limit is the capacity of two adjacent electrical transmitters to make an electric field between them. The sensor is made from two metal plates and contains a non-conductive polymer film between them. This film gathers moisture from the air, which makes the voltage between the two plates change. These voltage changes are converted in to digital readings demonstrating the level of moisture noticeable all around. *From* (futureelectronics.com, n.d.)

Figure underneath will give the brief introduction how the humidity sensor works.

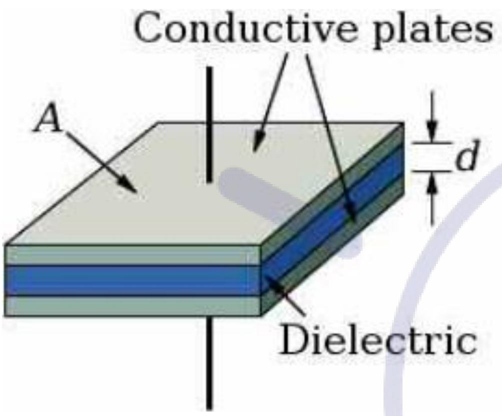

Figure 23 Inside View of Humidity Sensor (Source: Rotronic.com)

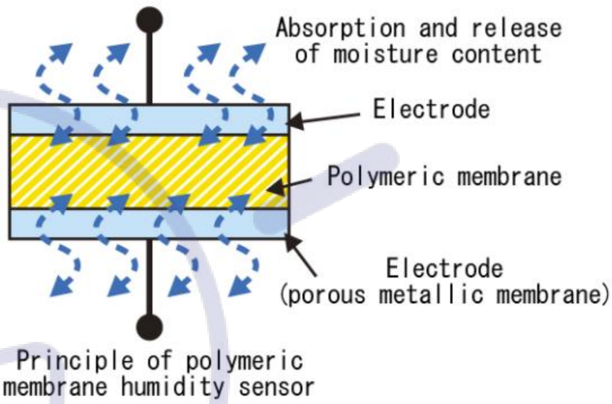

Figure 24 How Humidity Sensor Reacts (Source: Apsite-global.com, , n.d.)

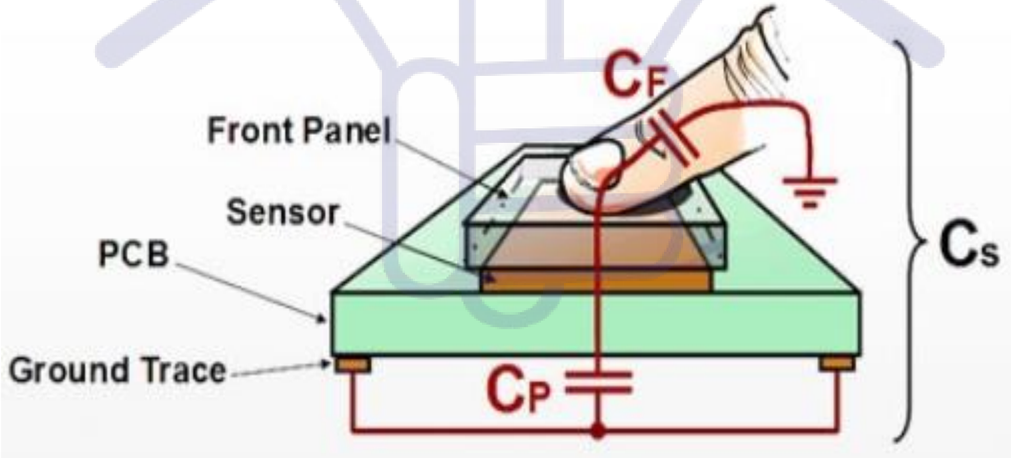

Figure 25 Working of Humidity Sensor (Source: EEE Projects, 2018)

Soil Moisture Sensors: The soil moisture sensor uses capacitance to quantify the water substance of soil (by estimating the dielectric permittivity of the soil, which is an element of the water content). Simply insert the sensor into the soil to be tested, and the volumetric water content of the soil is reported in percent. These sensors consisting of two electrodes and probes for estimating the soil resistance are frequently utilized for residential purposes. Time Domain Reflectometry (TDR) and Time Domain Transmission (TDT) are also used for measuring the soil moisture content. A higher average dielectric constant for the soil is caused by a higher water concentration. These sensors give real-time information and enhance the irrigation efficiency. The sensors are easy to install and require very less maintenance. *by* (Kalwinder, K, 2013)

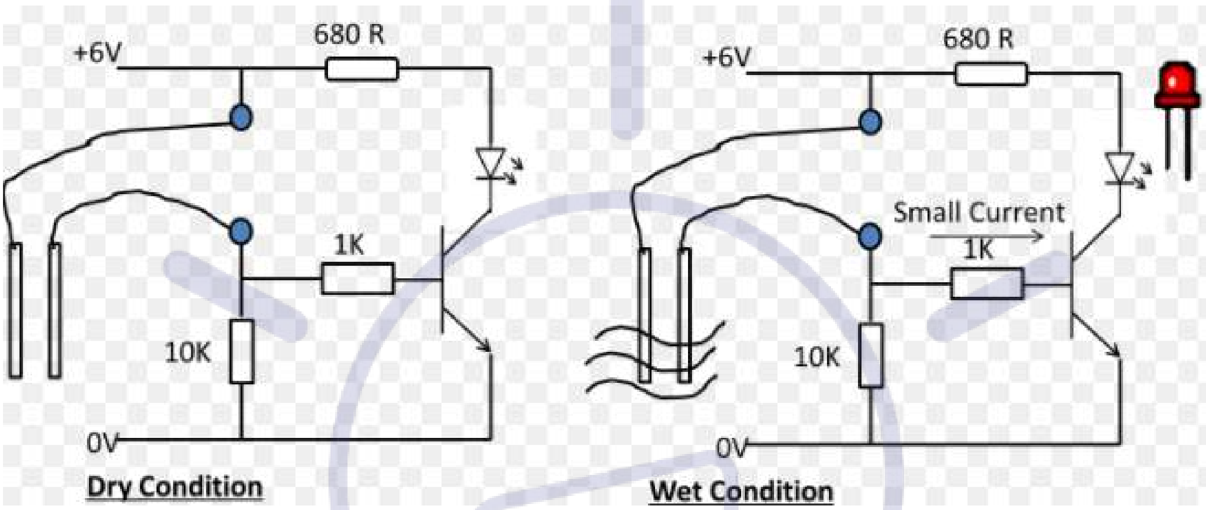

Figure 26 Basic Moisture Sensor Circuit (Source: Tsulo.com, n.d.)

figure above in *figure 26* shows how the basic moisture sensor circuit reacts in dry and wet conditions. During dry condition, when no moisture or water the probes are dry. Circuit is open, no current flows in to the base of the transistor, so the LED is OFF. During wet condition, when there is moisture or water, the probes gets short circuit when in contact with moisture/water. Circuit is closed, current will flow into the base of the transistor, LED turns ON.

Diagram underneath in *figure 27* shows how and where soil moisture sensor is utilized.

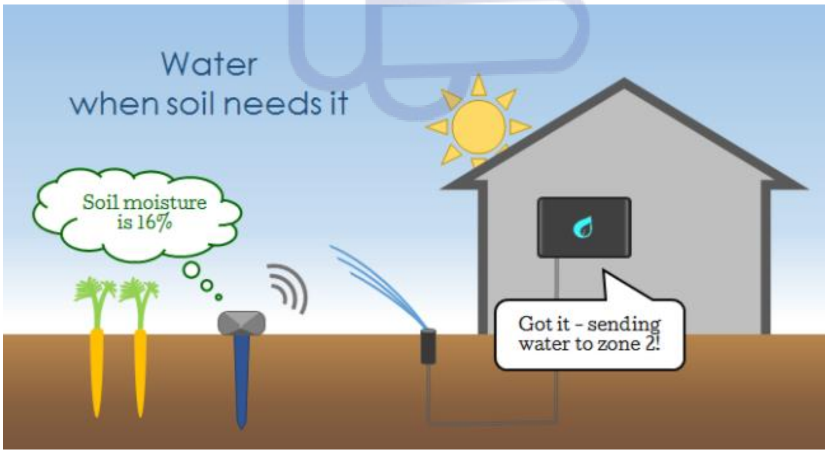

Figure 27 Use of Moisture sensor (Source: Robotics, S, 2017)

2.3. GSM Module:

GSM module is used to establish communication between a computer and a GSM-GPRS system. Global System for Mobile Communication GSM is a design utilized for mobile communication in most of the countries. "*GSM Module consist of a GSM modem assembled together with power supply circuit and communication interfaces like RS-232, USB, etc" by (Nikhil, A, 2010).*

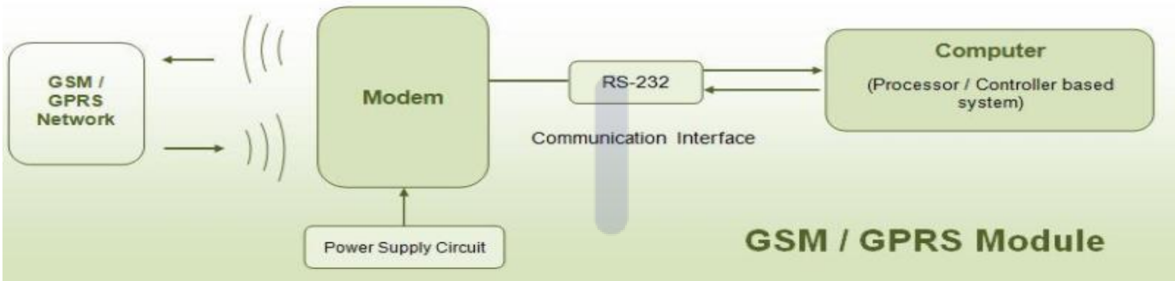

Figure 28 How GSM Works (Source: Nikhil, A, 2010)

GSM/GPRS Modem is a class of wireless modem gadgets that are intended for communication of a computer with the GSM and GPRS network. It requires a SIM card simply like cell phones to activate communication with the network. Additionally, they have IMEI number like cell phones for their recognizable proof. A GSM/GPRS Modem can perform the following operations:

- It can receive, send or delete SMS messages in a SIM.
- It can read, add, search phonebook entries of the SIM.
- Make, Receive or reject a voice call.

The modem needs AT commands, for interfacing with microcontroller, which are communicated through serial communication. Microcontroller sent these commands to the modem. The modem sends back a result soon after it receives the command. Different commands can be sent by the microcontroller to interface with GSM cellular network. The figure in *figure 29* below shows basic concept of how GSM SMS alerts works.

Figure 29 GSM SMS Alert (Source:)

2.3.1. GSM Architecture:

A GSM network consist of the following components

A Mobile Station (MS): it is a cell phones which comprises of the transceiver, the display and the processor and is controlled by a SIM card working over the system. (Agarwal, T, n.d.)

Base Station Subsystem (BSS): it acts as an interface between the mobile station and the network subsystem. It comprises of the *base transceiver station* which contains the radio transceiver and handles the protocols for communication with cell phones. It likewise comprises of the *base station controller* which controls the *base transceiver station* and acts as an interface between the mobile station and *mobile switching centre*. (Agarwal, T, n.d.)

Network Subsystem (NSS): it gives the basic network connection with the mobile stations. The essential piece of the network subsystem is the mobile service switching center which gives access to various system like ISDN, PSTN and so on. It additionally comprises of the *home location register* and the *visitor location register* which gives the call routing and roaming capabilities of GSM. It likewise contains the *equipment identity register* which keeps up a record of all the mobile equipment wherein every mobile phone is recognized by its own IMEI number. (Agarwal, T, n.d.)

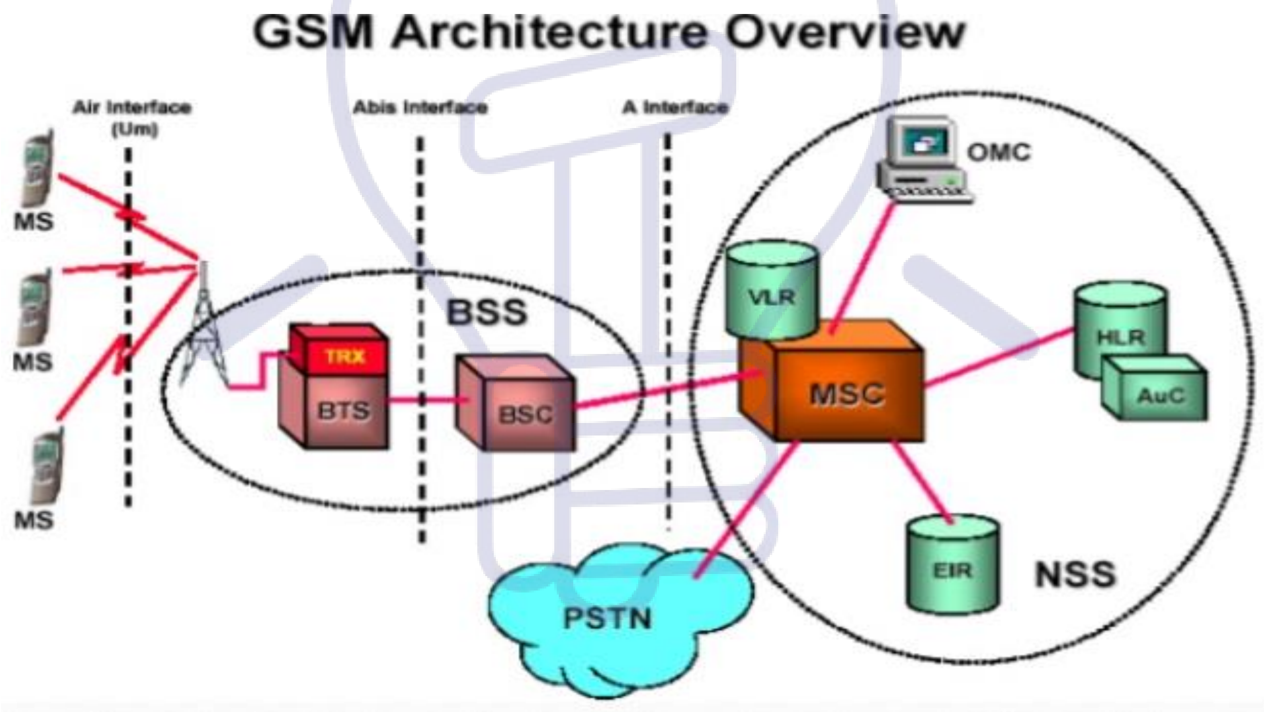

Figure 30 GSM Architecture (Source: Agarwal, S, 2014)

Chapter 3: Project Design

This section is talk about any work finished in software design and hardware design. It also goes into insight about what the system includes and why different components were chosen to make a fully completed Automated Plant watering system.

Consider a diagram below in *figure 31* shows the basic conceptual model of the system using chosen components.

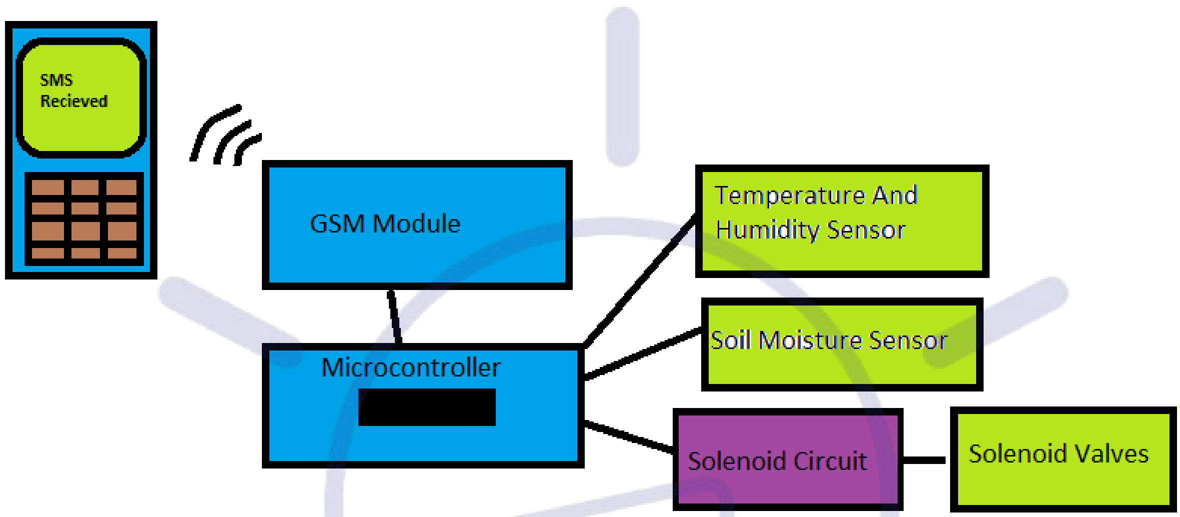

Figure 31 Functionality of the System

The functionality of above block diagram *figure 31* of the automated plant watering system is illustrated below:

- The system includes GSM module which sends SMS to the recipient and receives SMS from the recipient.
- It uses soil moisture, temperature and humidity sensors.
- It also includes solenoid valves along with solenoid circuit which controls the flow of water through the solenoid valves.
- The moisture sensor is used to determine the moisture level of the particular plant. This moisture level is read by the microcontroller, as it loops around and sees if the value from the sensor is above the threshold value or not. If the value is above the predefine value than GSM module is ready to send a SMS to the recipient.
- The temperature and humidity sensor is used to determine whether it is too hot for the plant or the humidity is too high to handle. This is read by the ATmega328p chip on Arduino Uno. these reading will be used to determine if all the solenoid in the system should be On or Off.

3.1. Theory and Reasoning for the Choice:

This section of the report talks about why different components were chosen and how they interfaced with a microcontroller. The idea was to build a fully functioning automated plant watering system. To do this, electronic components must be chosen very carefully and to ensure that all the chosen components interfaced correctly with the microcontroller.

After the generous research on the components and their specifications, various key parts were chosen:

Arduino GMS Module: Arduino GMS shield is one of the vital components in the system as it allows Arduino to make telephone calls, send SMS, interface with the websites, and making voice calls utilizing the GMS library. The shield utilizes a proportion modem M10 by Quectel (that associates with the client application and air interface). It is conceivable to speak with the board utilizing AT commands. GSM library has many methods for communication with the shield. GSM shield is open and advanced cellular technology utilized for transmitting mobile voice and information administrations works at the 850MHz, 900MHz, 1800MHz and 1900MHz recurrence band.

Shield utilizes digital pins 2 and 3 for software serial communication with M10 pin 2 is associated with M10's TX pin and pin 3 to its RX pin. It supports TCP/ UDP and HTTP protocol through a GPRS connection. To interface with the cellular network, the board requires a SIM card provided by the network operator.

Digitals pins 2,3 and 7 are reserved for communication for Arduino and modem and cannot be used for sketches. This communication between the modem and the Arduino is handled by the Software Serial library on pins 2 and 3. Pin 7 used for the modem reset.

The shield contains the number of status light:

- On: Shows the Shield gets power
- Status: turns on to when the modem is powered, and data is being transferred to/from GSM/GPRS network
- Net: blinks when the modem is communicating with the radio network.

Figure 32 Arduino GSM Shield (Arduino Corporation,1990)

Arduino Uno: Arduino Uno contains Atmega328p chip which acts as a main microcontroller of the system. As Arduino Uno has a number of facilities for communicating with a personal computer, another Arduino, or other microcontrollers. The atmega328p chip on Arduino provides UART TTL 5V serial communication, which is available on digital pins 0 RX and digital pin 1 TX. It also supports I2C and SPI communication. The biggest advantage of using an Arduino Uno Is its prepared to utilize structure. As Arduino arrives in an entire bundle shape which incorporates the 5V regulator, a burner, an oscillator, a microcontroller, serial communication interface and headers for connections. Another advantage of using an Arduino is its automatic unit conversion capability. Also, it very easy to interface different sensors and get them working with the Arduino. By default, it uses 16MHz clock speed which means microcontroller can execute 16 million instructions per second. It also possesses ADC, UART, and Timers.

Figure 33 Arduino Uno (Arduino Uno, n.d.)

Further specification of Arduino Uno can be seen in *table 3* below:

Table 3 Arduino Uno Specifications (Arduino Uno n.d.)

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) (0.5 KB used by bootloader)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

DHT11 Temperature and Humidity Sensor: This sensor in the system is used to detect the temperature and humidity readings from the surroundings. It is a composite sensor contains an adjusted digital signal output of the temperature and humidity. The sensor incorporates a resistive feeling of wet segments and NTC temperature estimation devices and connected with high performance 8 – bit microcontroller. Its sampling rate is around 1Hz or 1 reading every second. Operating voltage of the sensor is 3 to 5V while maximum current used when measuring is 2.5mA. They basically consist of a humidity sensing component, NTC temperature sensor and an IC on the back of the sensor. They are very easy to interface with the Arduino as they use single bus communication with the Arduino using one of the analogue pins. They have their own library called DHT which must be downloaded and used as an external library in Arduino software.

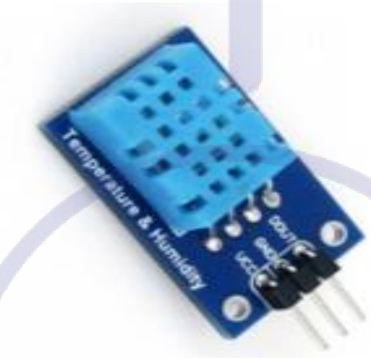

Figure 34 Humidity and Temperature Sensor (Waveshare, n.d.)

Grove Soil Moisture Sensor: This sensor is used to detect the moisture in the soil and give the readings back to the microcontroller. It is consisting of two probes which are used to measure the volumetric content of water around it. These two probes enable the current to go through the soil and afterward it gets the resistance reading to measure the moisture value. Normally, this sensor is consisting of three pins. Power 5V, Ground and analogue output pin which give the reading back to the microcontroller. The advantage of this sensor is during an interface with the Arduino extra hardware is not required that's why it is easy to use. It can also work with different architecture like raspberry pi etc.

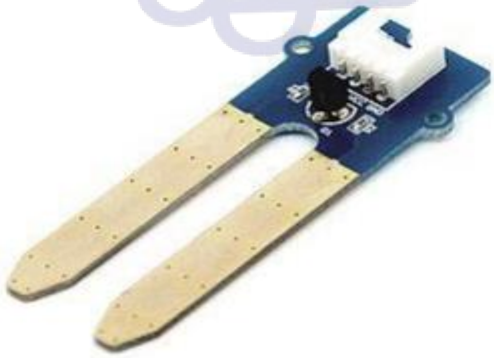

Figure 35 Grove Moisture Sensor (li, 2015).

Some of the technical specification of the sensor is shown below in *table 4:*

Table 4 Table for Specification of Moisture Sensor (li, 2015).

Item	Condition	Min	Typical	Max	Unit
Voltage	-	3.3	/	5	V
Current	-	0	/	35	mA
Output Value	Sensor in dry soil	0	~	300	/
	Sensor in humid soil	300	~	700	/
	Sensor in water	700	~	950	/

Solenoid Valve: A solenoid valve in the system prevents the water flow when it's not needed. It is an electromechanically operated valve. They can be normally closed or normally opened. It is a simple form of an electromagnet which consists of a coil of insulated copper wire. It is frequently used to control the flow of liquid or gas. They are found in many applications and commonly used in refrigerator and conditioning system. They offer fast and safe switching, reliability, long life and compact design.

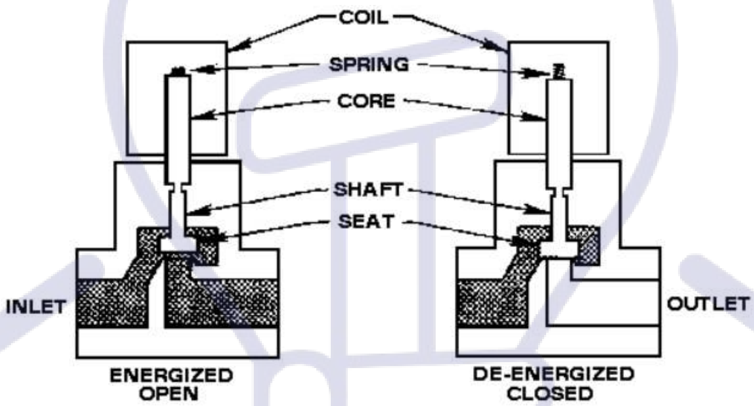

Figure 36 Solenoid valve (Plast-O-Matlic Valves, 2008)

Some of the technical specification of the solenoid valve is shown below in the *table 5:*

Table 5 Solenoid Valve Standard Specification table from (content.smcetech.com,2010)

Valve specifications	Valve construction	Direct operated poppet
	Valve type	N.C.
	Withstand pressure	5.0 MPa
	Body material	Brass (C37), Stainless steel
	Seal material	NBR, FKM, EPDM, PTFE
	Enclosure	Dusttight, Low jetproof (IP65)
	Environment	Location without corrosive or explosive gases
Coil specifications	Rated voltage	24 VDC, 12 VDC
	Allowable voltage fluctuation	±10% of rated voltage
	Allowable leakage voltage	2% or less of rated voltage
	Coil insulation type	Class B
	Surge voltage suppressor	Built-in surge voltage suppressor

Normally Closed (N.C.)
DC Specification

Model	Power consumption (W) (Holding)	Inrush current (A) (Inrush time: 200 ms)		Temperature increase (C°) Note)
		24 VDC	12 VDC	
VXE21	1.5	0.19	0.38	25
VXE22	2.3	0.29	0.58	25
VXE23	3	0.44	0.88	30

Relays: Relays in the system control the voltage flow for solenoid valve. Basically, it prevents the flow of voltage which is essential for the solenoid valve to get magnetized and let the water flow through it. They behave as electromechanical switches which open and close. They usually operated by the magnetic coil, when the coil of the relay gets energized it switches to normally closed position and allows 24V for solenoid valve pass through it. Relays are very useful to interface with high voltage motors and solenoid valve etc. As they simply act like a switch. In order to derive high voltage components such as motors, solenoid etc relays are very reliable and easy to use.

Figure 37 Relay diagram (finder)

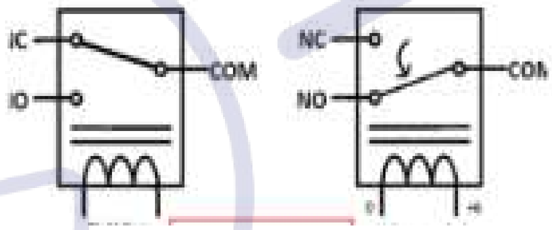

Figure 38 Relay circuit (Jayajant,2015)

Transistor: A transistor is a device that controls current and voltage stream and works as a switch for electronic signals. It is consisting of three layers of a semiconductor material, each fit for conveying a current. They are fundamental components in IC, which comprise of a substantial number of transistor interconnected with circuitry and baked into a single silicon microchip. One of the most common uses of the transistor in an electrical circuit is that they act as a simple switch. In short, a transistor conducts current across the collector-emitter path only when a voltage is applied to the base. When there is no base voltage, transistor seems to be in switch off mode. When there is base voltage present, transistor switch is on.

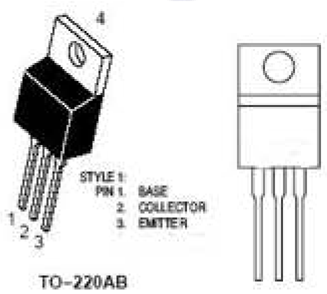

Figure 39 T0-220 Transistor (Vakits.com, 2012)

Hitachi 16x2 LCD: This 16x2 LCD screen displays a valuable information to the user about what's the current temperature and humidity value is and what's the current value reads by the soil moisture sensor. It also displays when the GSM module sends SMS and receive SMS from the recipient along with the mobile number.

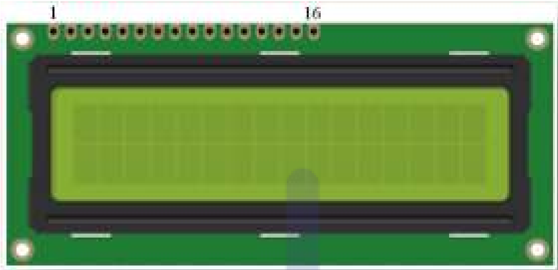

Figure 40 Hitachi 16x2 LCD Screen (Sumeeteshop)

3.2. Hardware Design:

Once all the necessary components for the system had been chosen, now it's time to move on to the hardware design of the system and get it tested. The first thing which must be done is the schematic diagrams of the circuit. All the schematic diagram was done using Cad star editor.

There are three different schematics drawing were made for the system which is as follows:

3.2.1 Sensor Schematic:

figure 41 shows the schematic of the Sensor circuit. As all the sensor connected with the Arduino analogue pin A0-A3. Pin A0 was reserved for the temperature and humidity sensor whereas pins A1-A3 were reserved for moisture sensors. All the sensors have common power 5V and Ground as shown in the schematic.

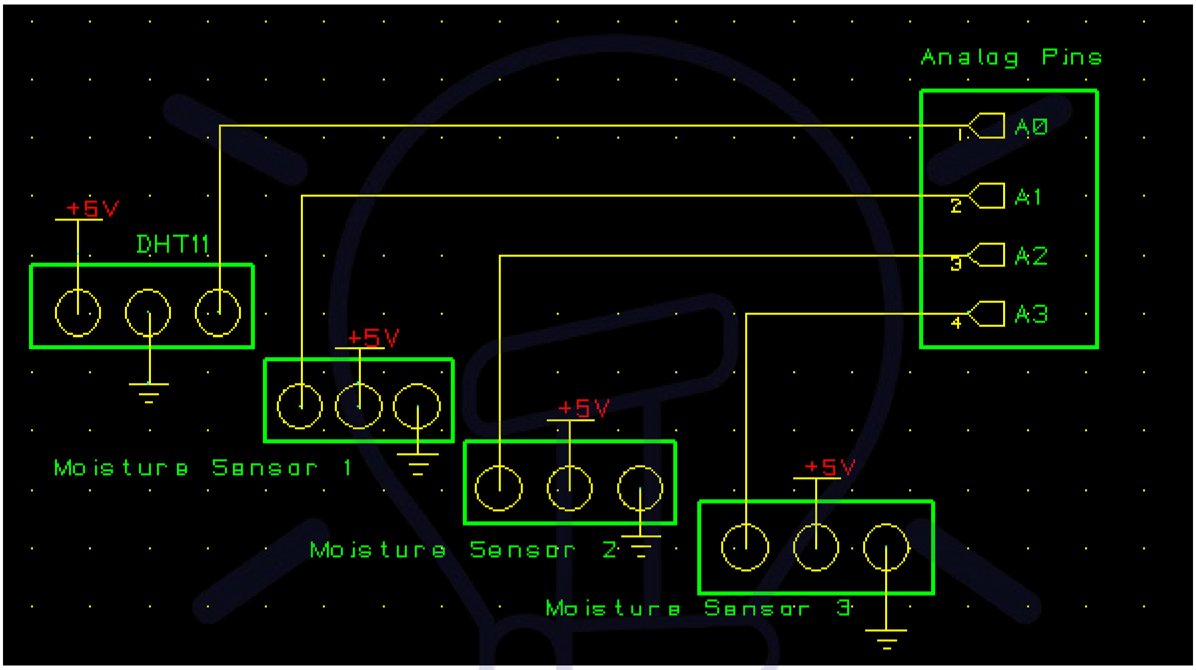

Figure 41 Sensor Schematics

3.2.2 LCD Schematic:

Figure 40 shows the schematic of the LCD. Digital pin 8 – 13 was reserved for the LCD as shown in the Schematic. Pin 1 and Pin 3 are the power and ground whereas pin 2 is the contrast pin on the LCD which controls the contrast and connected with the potentiometer. It must be kept in mind while connecting LCD is that digital pins from Arduino and data pins from LCD has to be connected in correct order otherwise LCD will display only garbage on the screen.

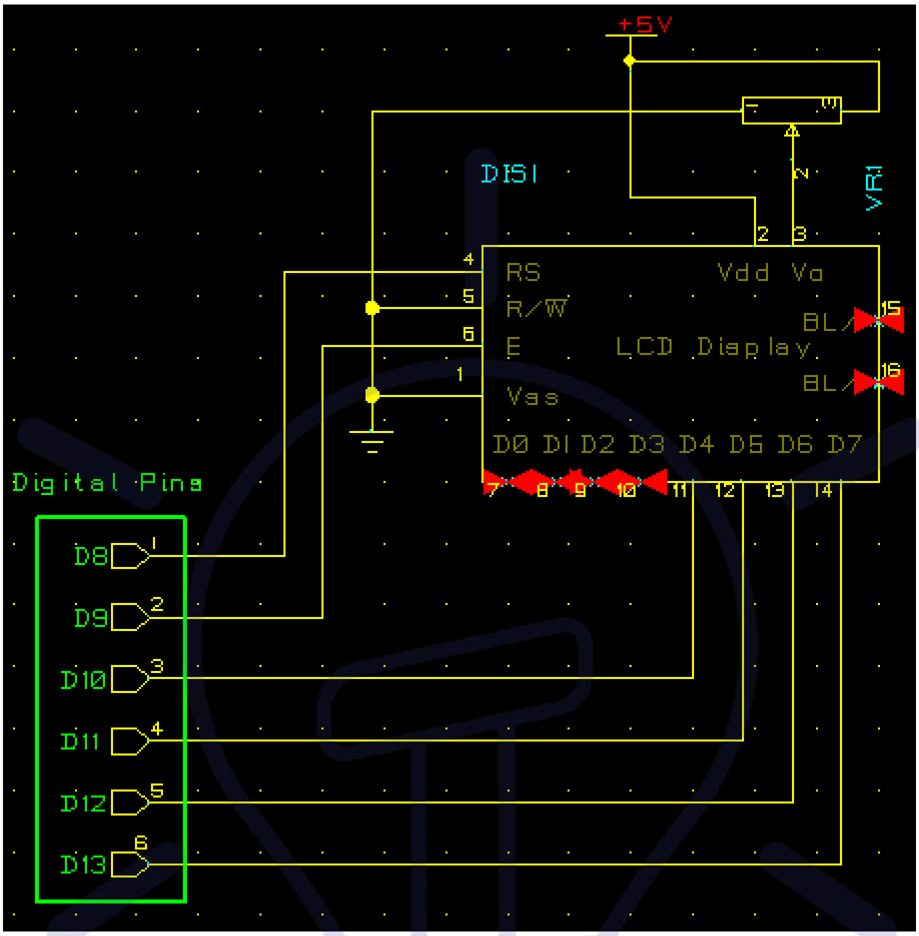

Figure 42 LCD Schematic

3.2.3. Solenoid Schematic:

Diagram underneath in *figure 43* shows the circuit diagram of the solenoid circuit. Digital pin 4 – 7 was reserved for the Solenoids. As the circuit consists of relays, transistors, resistors, and LEDs instead of the solenoid (CadStar doesn't have Solenoid symbol). In the schematic relays utilizes 5V. Whereas 5V also goes into NO channel of the relays as well this is because in the schematic LEDs replaces solenoid which works on (5V) followed by 220-ohm resistor.

So, when the voltage is applied to the base of the transistors. Transistor switches to ground allowing the coil of the relay to get magnetized and switches itself to Normally closed channel, due to which LED connected to that particular relay comes On and when the voltage applied on the transistor base is drops, transistor switches itself back to normal and the coil of the relay get de-magnetized and relay switches to NO channel again, due to which LED goes Off again.

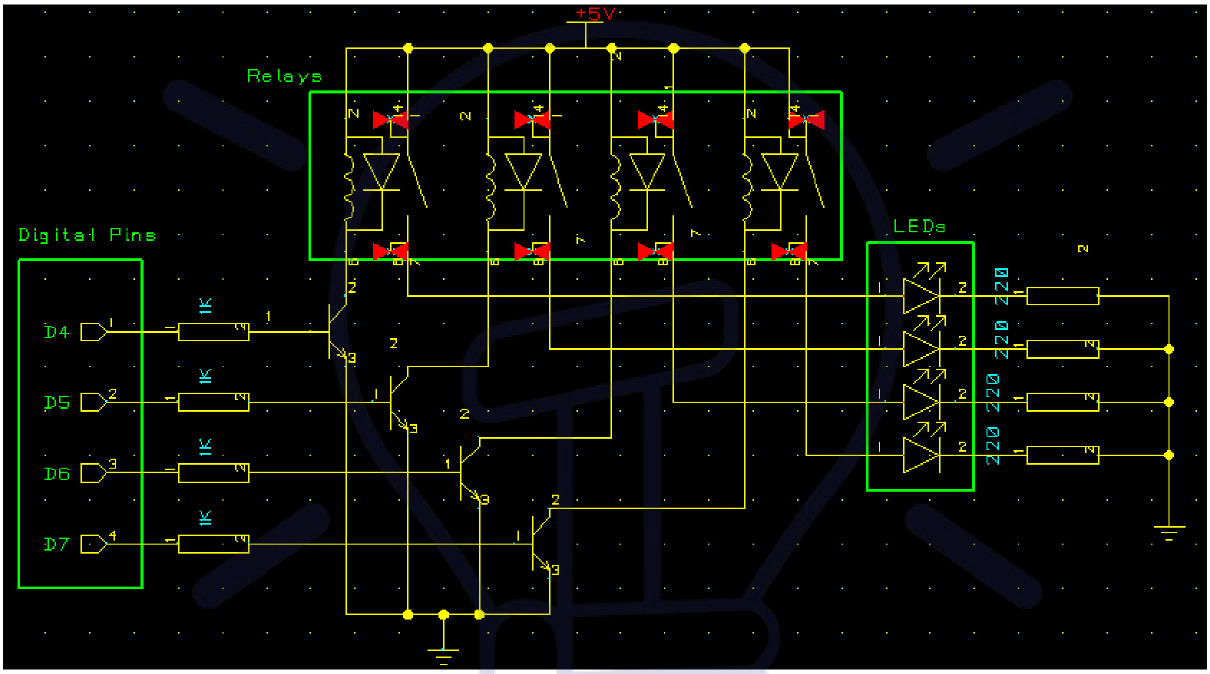

Figure 43 Solenoid Schematic

After finishing off all the Schematic of the circuit, next step is to build them on the Veroboard. It is important to design the circuit on stripboard layout planning sheet in advance because there are certain principles for designing a circuit on Veroboard which is as follows:

1. Mark out the Vs and GND power line first on the top right of the stripboard layout planning sheet.
2. Remember to cut the track between the pins of an IC. Mark the cuts on the diagram with an X.
3. Try to make resistor and axial capacitors lay flat on the stripboard. Resistors usually require a gap of 4 holes, capacitor a gap of 8 holes.
4. If possible numbers the pin of the ICs.

The bottom side of the Veroboard consist of the copper tracks in which voltage flows works horizontally. Different Veroboard design of the schematics above are shown below:

Veroboard Design for Sensors:

The design below in *figure 44* shows the Veroboard circuit for sensors schematic. On the top right of the circuit power 5V and GND were marked out, 5V flows horizontally on the board and gives power to grove moisture and DHT11 (temperature and humidity sensor). On the bottom right there are analogue pins from the Arduino which takes data reading from each of the sensors and sends it to the microcontroller.

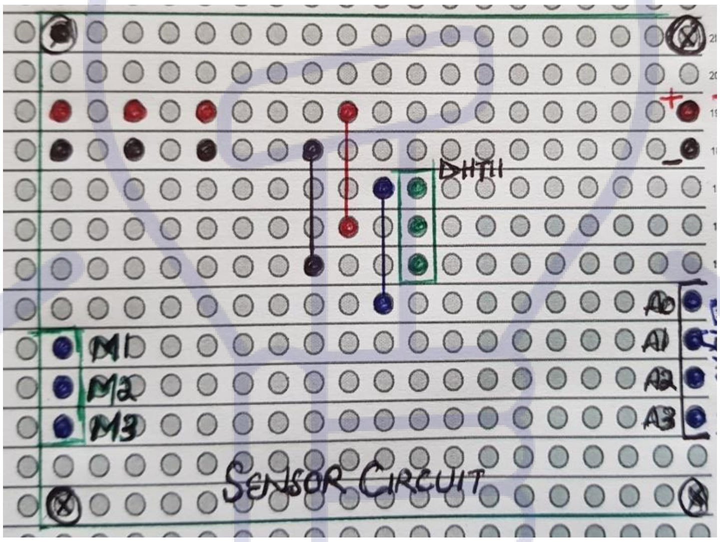

Figure 44 Veroboard Design for Sensor Circuit

Veroboard Design for LCD:

The figure below in *figure 45* shows the planning sheet for LCD circuit on stripboard. To get the LCD working properly it has be keep in mind that all the LCD pins must be connected in order with the Arduino for instance; LCD pins such as RS, E, D4, D5, D6, D7 must be connected with Arduino's digital pins 8, 9, 10, 11, 12, 13 respectively. Pin 1 and 2 from the LCD relate to power and GND whereas, pin 3 is a contrast pin which relates to a potentiometer as shown below, pin 5 from the LCD is also connected to GND as the LCD is in reading mode.

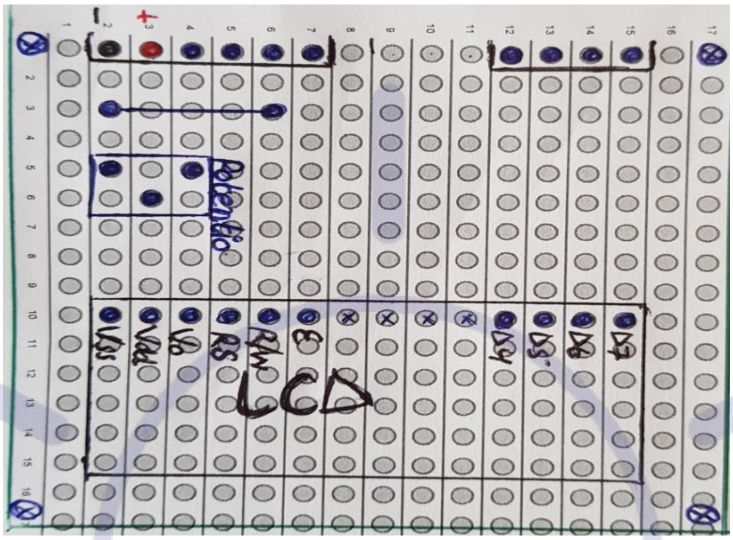

Figure 45 Veroboard Design for LCD

Veroboard Design for Solenoid:

Veroboard layout planning sheet for solenoid can be shown in the figure below in *figure 46* and the function/ description for this design has already been described under solenoid schematic section above.

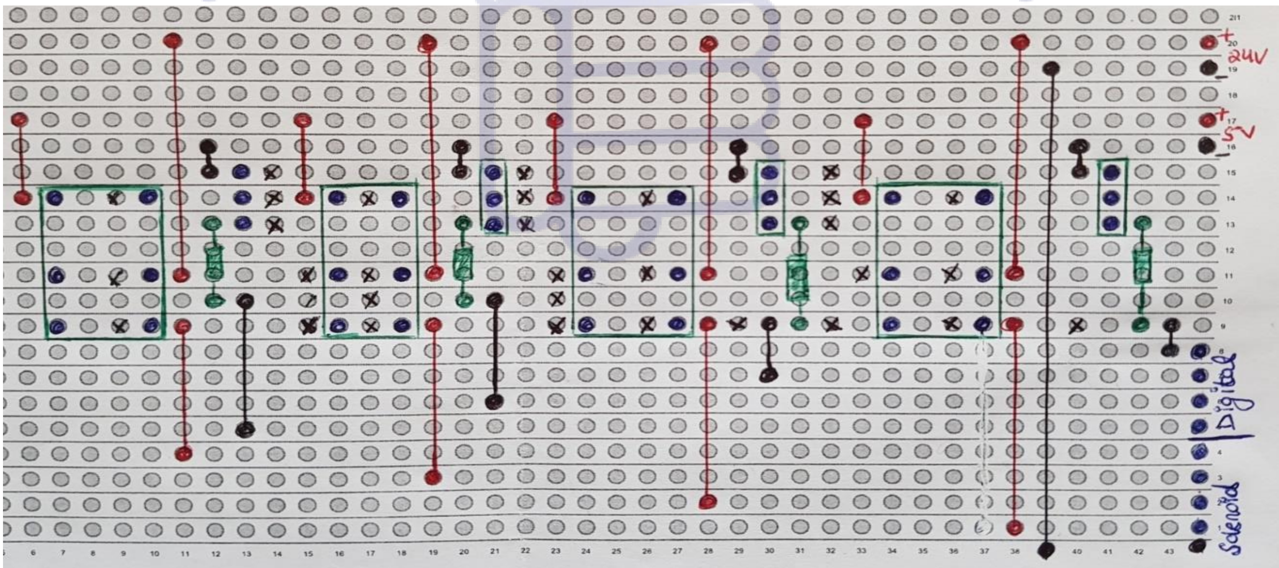

Figure 46 Veroboard Design for Solenoid Valve

3.2.4. Additional Circuitry:

This additional circuit is included in the system because due to GSM module which takes up 3 pins from the Arduino, digital pin 2 and 3 is used for communication between modem and Arduino. Whereas, digital pin 7 is used to reset the modem. Digital pin 8 - 13 is reserved for LCD display. Pin 0 and pin 1 are TX and RX pin on Arduino itself and can only be used for communicating purpose. So, Arduino is left with only 3 digital pins (4, 5 and 6). Instead of using other board such as Arduino mega this additional circuit was required to overcome this problem, which takes up 3 pins from the Arduino and spits out 8 pins which can be used to control the solenoid valve. 74HCT595N is an 8-bit shifter register was used in this circuit which basically expands the digital output from the microcontroller. It has 8 output pins (Pin 15 and Pin 1 – Pin 7). Whenever the signal on the clock pin (Pin 11) goes HIGH, all the values get shifted to the right and the new values get shifted in (whatever the data pin 14 is set to). After the new values get shifted in and to see the changes made Latch pin 12 must also set to HIGH in order to update the output-pins with the new data.

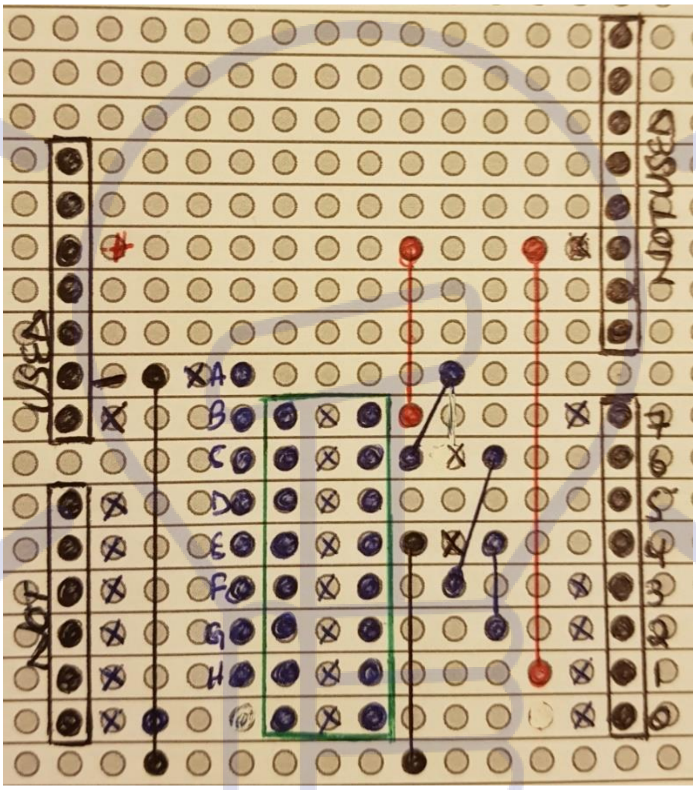

Figure 47 74HCT595N Circuit

In addition, there is no as such circuit diagram for the GSM module as it sits on top of the Arduino and uses only 3 digital pins as described above. Veroboard design above in *figure 45* design in such a way that it can also sit on top of GSM so there is no additional space required for this. After getting all the Veroboard designs built. The next step is to build the base of the system using aluminium. On this aluminium base, all the Veroboard design would be situated. It is important to decide the dimensions of the base, so all other design can mount over it without overlapping. After deliberating different ideas, a base of 440mm x 200mm was chosen with a right-angled bend of 35mm followed by another right-angled bend of 15mm in from either end.

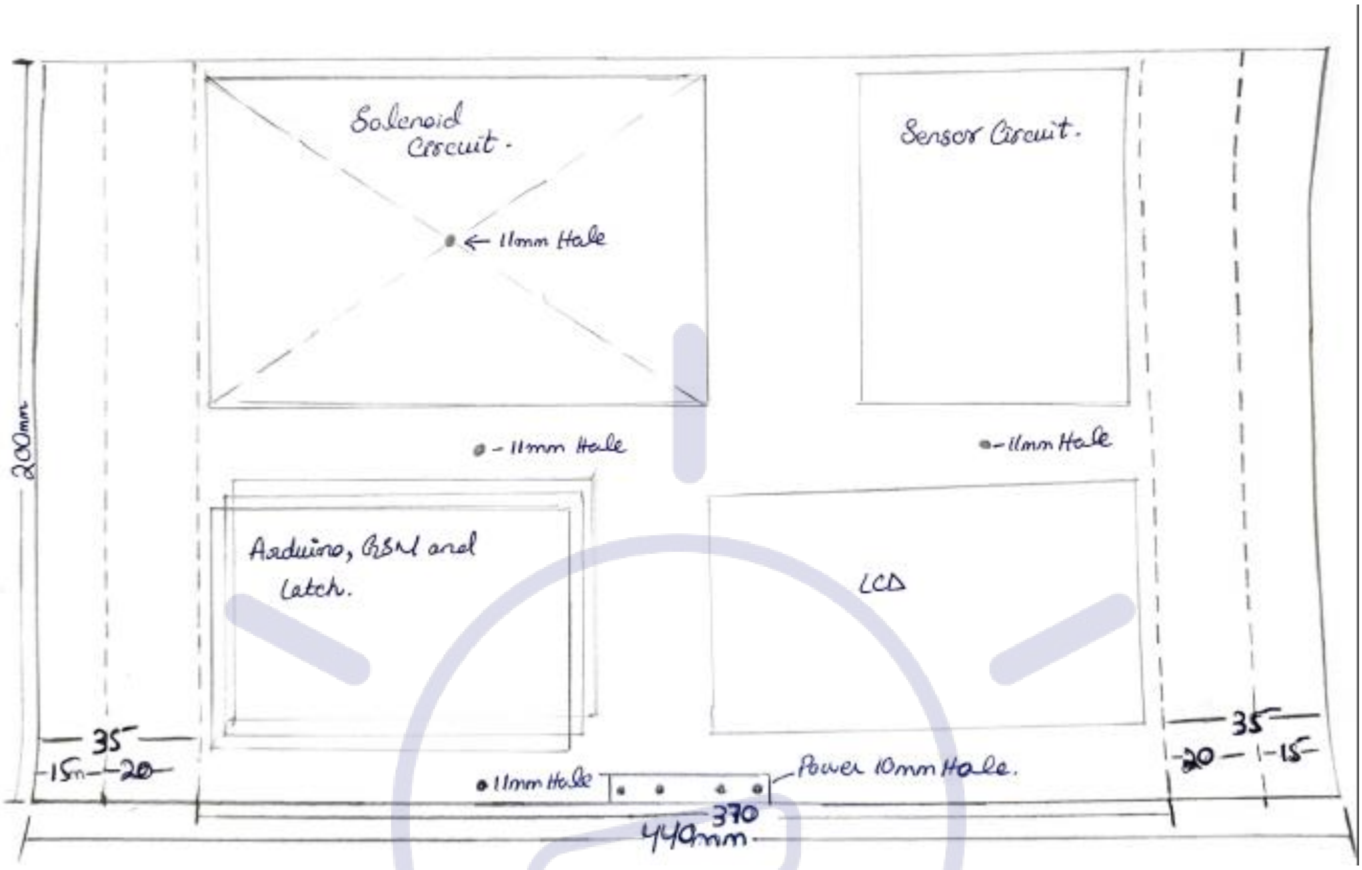

Figure 48 hardware aluminium sheet base and design

3.3. Software Design:

After getting the hardware done, it's time to test the hardware with the software. In this section, the implementation of the software design will be described in detail for each of the different automation/ technologies used within the system. This includes the Arduino code written and uploaded to the Arduino.

The first thing done was to get the solenoid circuit working and how the solenoid would act from the microcontroller perspective. For this, a small flow chart was done which can be seen under software flow section in *figure 50*.

Arduino IDE was used to get the upload the software on the Arduino. For the basic solenoid circuit, a simple program was written which basically blinks the LED every 1 sec. digital pin 4, 5, 6 and 7 was defined initially which test the program and the circuit. So, when the program runs it makes all basic initializations, defines all the output pins in ***void setup ()*** and then jumps into the ***void loop ()*** where it constantly runs and blinks LEDs on every 1sec.

After that, a small program was written and uploaded to the Arduino which gets the readings from the different sensor and prints them on the LCD. For this, a small flow chart was done which can also be seen under software flow section in *figure 51.* When the program goes into the ***void loop ()*** it gets the readings from the sensor and does all the basic calculation and prints them on LCD.

Next thing is to upload the software for GSM module to the Arduino, through which GSM could communicate with the microcontroller. Modem test was done initially which does all the basic initialization and libraries for the GSM and gets the IMEI number and see if the modem is functioning properly once start communicating with Arduino. The next step is network connection test which basically initialized the GSM and displays all the other network which GSM module can support.

Once the GSM module is tested and functioning properly it's time to use GSM module to communicate with the recipient, which means sends SMS to the recipient and receive SMS from them. In order to do that, another simple Arduino wiring program was written and uploaded to the Arduino. The program initialized the GSM and send SMS to the recipient in contrast another Arduino program was written in which GSM receive the SMS from the end user.

Finally, once all the software design was done it's time to merge all the software design together and build a final working software for the system. Different algorithm approaches were applied which can be seen under software flow section to get the final software working and does it what it supposed to do. *Figure 49* below shows the working of the final software where it takes a reading, send SMS, receive SMS and start doing what it was doing previously.

Note: *All the software code can be seen in the* ***appendix*** *below.*

```
Mositure 1: -10%
Mositure 2: -12%
Mositure 3: -17%
Current humidity = 51.00%temperature = 19.00C
Sending a message to mobile number: 0899506304
SENDING

Message:
Humidity is High. Open All Solenoids

COMPLETE!

O
p
e
n
A
l
l
OpenAll

END OF MESSAGE
MESSAGE DELETED
Mositure 1: -10%
Mositure 2: -13%
Mositure 3: -17%
Current humidity = 51.00%temperature = 19.00C
```

Figure 49 final Software test

NOTE: *all the software code can be viewed in the appendix. The output from the modem test and network connection test was not included in the report because the actual report was done after the submission of hardware.*

3.4. Software Flow Diagram:

3.4.1. Basic Solenoid Flowchart:

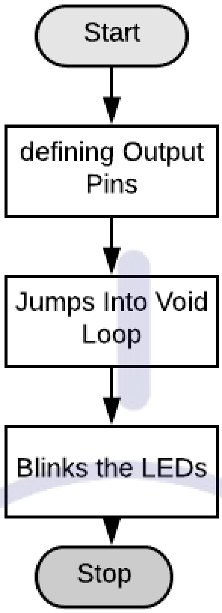

Figure 50 Solenoid Basic Flow

3.4.2. Sensor Flowchart:

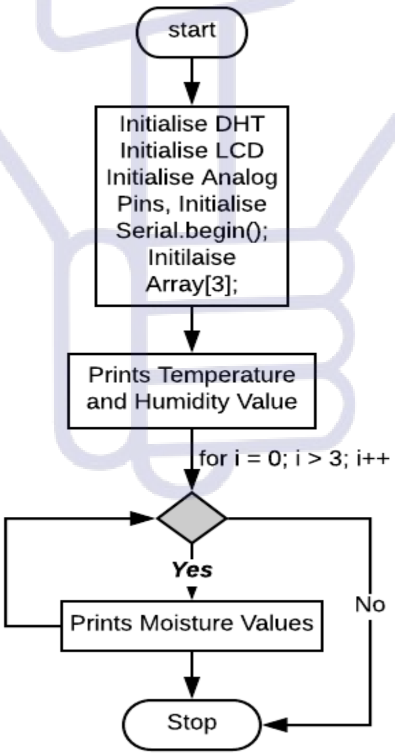

Figure 51 Sensors Flow

3.4.3. Solenoid Circuit Complete Flowchart:

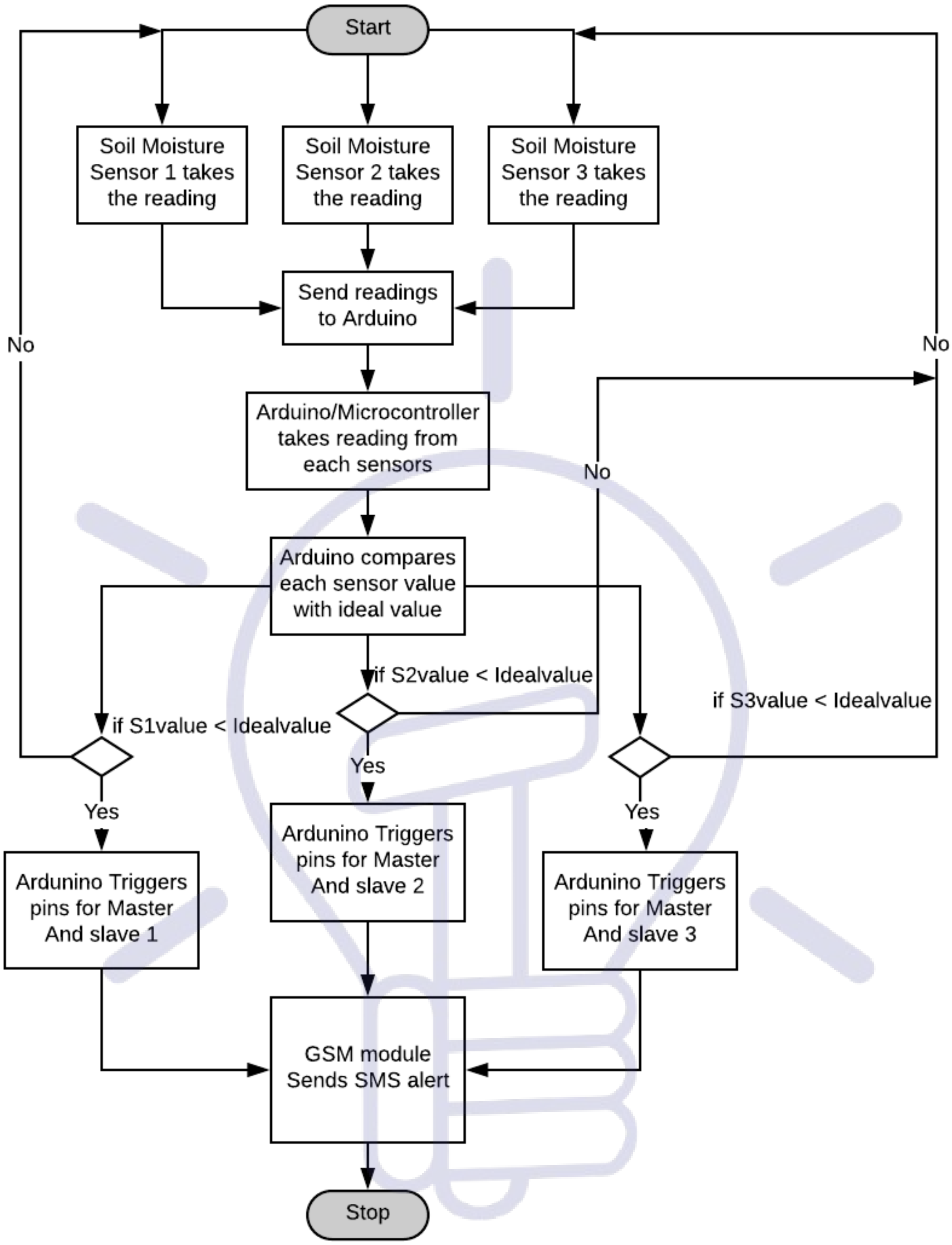

Figure 52 flowchart for Solenoid

3.4.4. Final Software Flowchart:

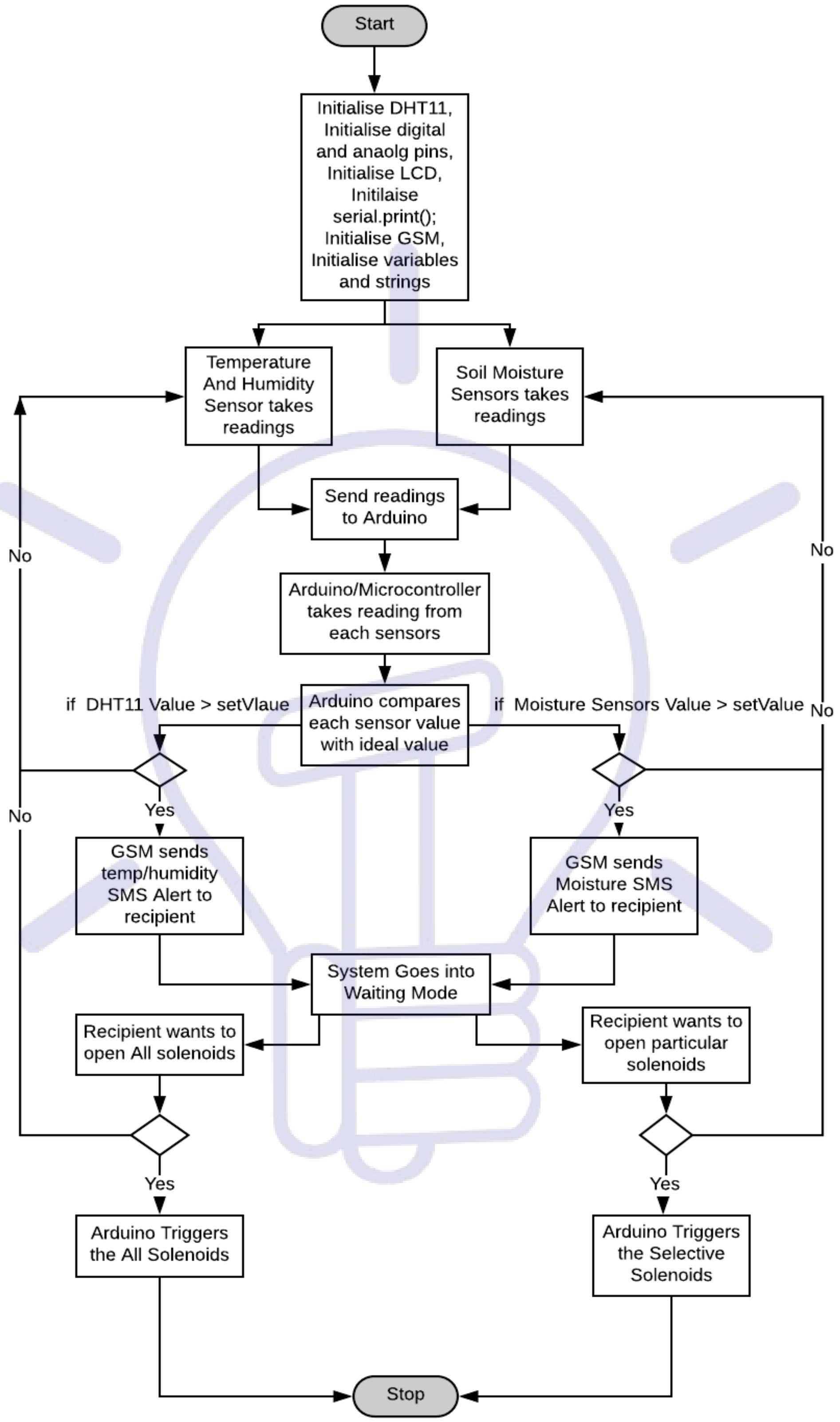

Figure 53 Software Main flow

Chapter 4: Project Construction and Testing

After getting all the hardware and software design done successfully it's time for the project construction and testing. In this section of the report, details will be given on how the different hardware design gets implemented and tested. This section also talks about if there was any hidden problem within the software code that was important to troubleshoot and evacuate and to build the project successfully.

4.1. Construction:

When the hardware design for all the schematics has been completed. It's time to build the design on the Veroboard itself. The first thing done was to solder all the components on the Veroboard. It has to be done with great care, if the soldering goes wrong it's hard to de-solder the components because de-soldering could damage the component and the copper tracks of the Veroboard. When soldering it has to be in mind that copper tracks on the Veroboard work horizontally, so this means all the components have to be soldered vertically on the board and if mistakenly two lines connect with each other then it is necessary to break the joint otherwise same voltage could flow on each line which ends up with a short circuit or the components will be damaged. An aluminium sheet was taken from the room and marked out according to the desired base calculations. Once the aluminium sheet marked out with all the necessary components on it. It's time to drill the holes in it. 3mm drill was used to mount the Veroboard circuits on the aluminium base, 10 mm drill was used for power and ground headers and 11mm drill was used for circuit wires which goes through them under the aluminium sheet and then connect with Arduino. The Arduino, GSM and latch circuit was mounted on the bottom left side of the aluminium sheet, on the top left of the sheet solenoid circuit mounted there which controls the solenoid project box resting on its left. On the top right, there was sensor circuit was placed which takes moisture and temp/humidity readings. reason for placing sensor circuit on the top right because the sensor can take readings from three different plants resting on top of the aluminium base. And then finally, the LCD was placed on the bottom right of the base which displays the readings taken by the sensors. The last step in the base construction is to take the aluminium sheet to the workshop and get it right-angled bend of 35mm followed by another right-angled bend of 15mm in from either end. Figure underneath in *figure 54* shows the completed aluminium base.

Figure 54 Aluminium base of the Circuit

4.2. Testing Procedure:

Testing procedure acquires an important section of the project report because it plays a vital role in the development of the project. Testing of the hardware and software is important because it founds any issues which can be present in the hardware or in software and can damage the important components of the system. Once all the Veroboard design was built with care it's time to test each individual Veroboard circuit. Testing of solenoid circuit board was the first test which was conducted because in this Veroboard design 24 volts and 5 volts was going to be flow at one time because solenoid valve works on 24V. Initially, circuit was tested using LEDs instead of the solenoid valve. When the power is connected to the circuit and basic code was uploaded to the Arduino. The testing of this circuit was successfully done as all the LEDs connected with the circuit blinks after every 1sec. Then LEDs was replaced with the solenoid valve and 24 V introduced in the circuit, same code was uploaded on the Arduino again and this time instead of LEDs solenoid valve goes on after every 1sec. figure underneath in *figure 53 – 56* shows the testing of solenoid circuit on LEDs.

The sample code was uploaded, after uploading the sketch green LED goes ON.

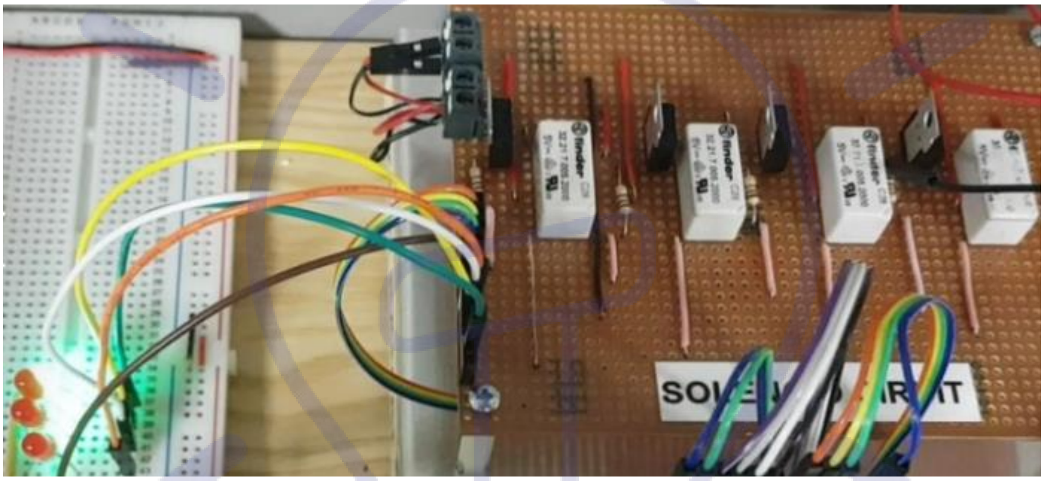

Figure 55 solenoid testing

After 1 sec green LED goes OFF and next LED goes ON and so on.

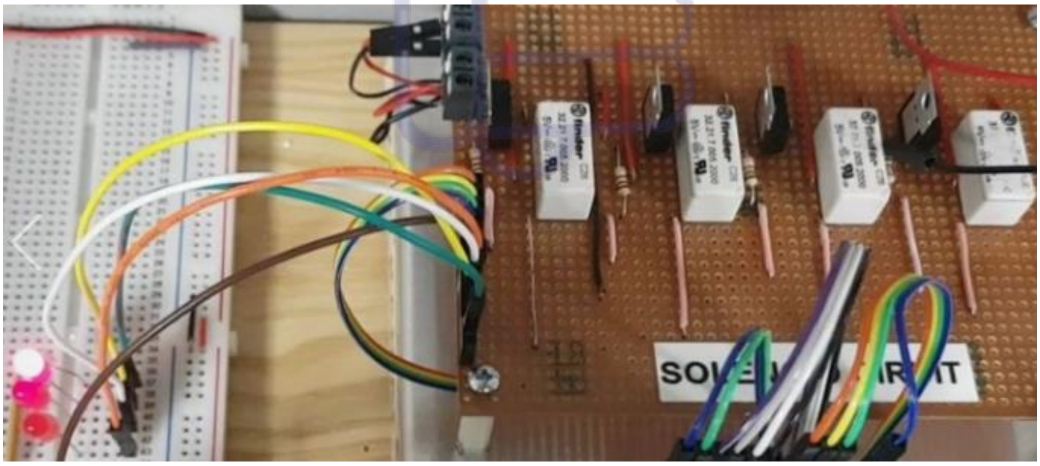

Figure 56 Solenoid Testing

The second test conducted was with the sensor circuit. In this test, sample code was uploaded on the Arduino which prints the reading on the serial printer and on LCD. This test included making sure that all the sensor on board is working properly and taking correct readings what they supposed to get. In order to test the readings from the sensor, another temperature and humidity sensor was introduced just to compare the measurements taken by both the sensors. readings from the sensor were close enough to say that the sensor was working properly. For the moisture sensor, it was clearly seen that all the sensor was working properly because they are taking the readings from the same pot and the readings taken by the sensor was close to each other. *Figure 57* shows the reading taken by the moisture sensor from the same pot.

```
Mositure 1: -10%
Mositure 2: -12%
Mositure 3: -17%
Current humidity = 51.00%temperature = 19.00C
```

Figure 57 sensor circuit testing

Note: *readings displayed on LCD wasn't captured because this report was completed after the hardware submission.*

Since there is no Veroboard circuit design for the GSM module and its testing can be seen in *figure 49 final software test.* The last test conducted was with the latch circuit, as it was the additional circuitry in the system which was due to the short of pins for solenoids. This circuit was designed in such a way that it can sit on the top of the GSM module. It was easy to test the circuit as it takes up 3 pins from the Arduino and spits out 8 pins which can be treated as digital pins. Description of this circuitry has already been described in the additional circuitry section. A code was uploaded to the Arduino which simply blinks a LED using the 74HCT595N shit register.

Figure 58 testing of the latch circuit

4.3. Software Testing:

Software testing phase is also an important aspect of the project development. Software testing is a procedure of executing a program or application with the goal of finding the software bugs. It can likewise be expressed as the process of *validating and verifying* that a software program or application meet its technical requirement, works as accepted and can be executed with a similar trademark. To do the software testing different approaches were adopted. A software requirement specification (SRS) document was written which fully addressed the expected behaviour of a software system.

Requirement ID	**SRS- Sensor -010**
Title	*Sensor*
Description	Sensors in the system take the readings and sends it back to the microcontroller.
Version	V1.0

Requirement ID	**SRS- Data -020**
Title	*Data Display*
Description	When the users try to get the reading from the system. Display should have displayed data to user for example: temperature and humidity value followed by moisture readings. .
Version	V1.0

Requirement ID	**SRS- Microcontroller -030**
Title	*Microcontroller*
Description	Microcontroller in the system act as a brain of the system which manages everything in the system
Version	V1.0

Requirement ID	**SRS- Latch -040**
Title	*Latch*
Description	Latch in the system expands the digital pins for the microcontroller
Version	V1.0

Requirement ID	**SRS- GSM-050**
Title	*GSM*
Description	System will react by sending a SMS alert to the recipient whenever microcontroller tells it to do so.
Version	V1.0

after writing the SRS document software design moved in to static testing phase which includes reviewing of the document. This is where verification of the requirements occurs. There are four diverse types of verification methods defined below:

- Inspection(I): control or visual verification.
- Analysis(A): verification based on analytical evidences.
- Demonstration(D): verification of operational characteristics, without quantitative measurement.
- Test(T): verification of quantitative characteristics with quantitative measurement.

For each requirement of the SRS document, a verification method is defined with abbreviation of I, A, D and T.

VEERIFICATION:

Requirement ID	**Requirement Title**	**Method**
REQ-010	Verify that the sensors of the system get readings	**I**
REQ-020	Verify that the Data is Displayed on the screen	**D**
REQ-030	It is verified that the microcontroller of the system is managing or working properly as it gives 100% result for every request.	**D**
REQ-040	Verify that latch circuit was doing what is supposed to do. That take sin 3 input and spits out 8 pins	**A**
REQ-050	Verify that the SMS has been sent and received by GSM	**D**

After the verification phase software testing moved in to the dynamic testing phase where unit testing and integration testing occurs. All these testing has already been done in testing procedure section.

4.4. Results:

As all the testing done with satisfactory result. Since there is no as such particular result that has to be documented. As the system works with moisture and DHT11 (temperature and humidity) sensor which takes reading according to the current room temperature and humidity. Readings from the moisture sensor in the circuit also depend on what the current moisture level is for the plant. Otherwise, overall result coming out from the circuit in terms of functionality was good for motivation.

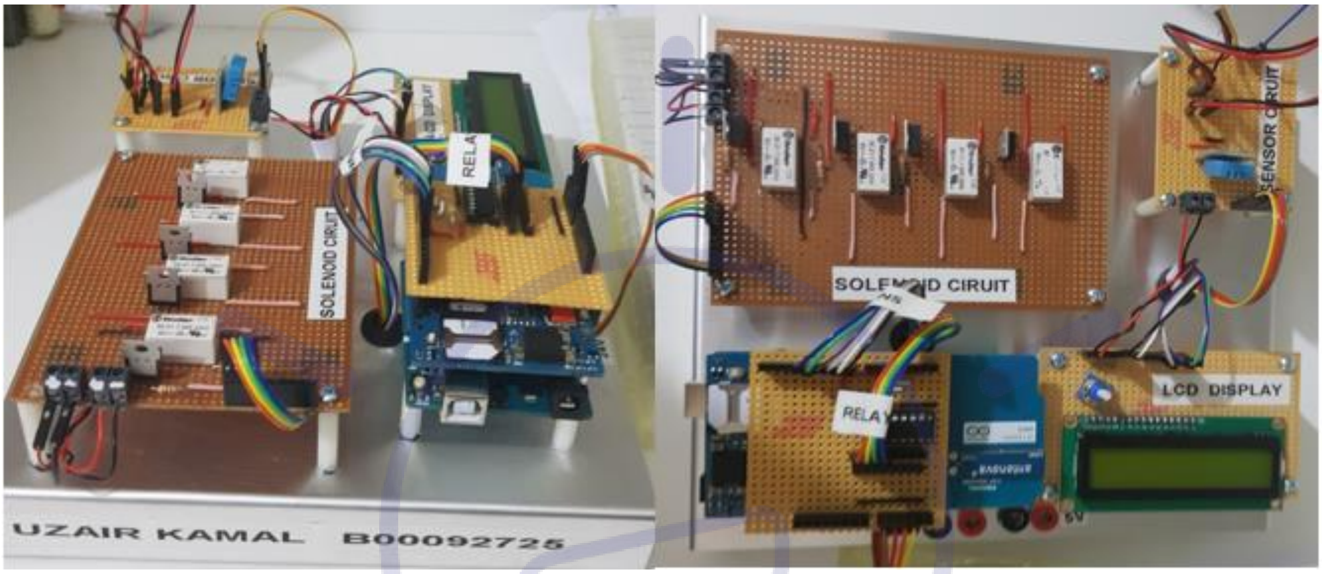

Figure 59 Final Circuit

Chapter 5: Safety and Ethical Consideration

Safety and ethical consideration are one of the important aspects of the project. Because engineers must keep this in mind that accidents can happen anywhere and with anyone. So, there is always a danger while doing this kind of project which includes heavy machinery. So, it is important for every engineer to keep this safety and ethical consideration in mind while working in the workshop or surrounded by heavy machinery which can cause severe damage. This chapter will go into the details about these safety considerations.

While doing such project safety is one of the important thing in terms of construction of the project. For instance, when drilling, safety glasses must be put on and safety guard must be used. This is because driller can produce a dust particle which can go in the eyes for this reason safety glasses is important for drilling. When drilling always start from the small drills because using bigger drills straightaway can damage the thing which needs to be drill and can also cause lots of noise, the worst-case scenario is that drill can stick into the board and start spinning it along with itself which can cause severe injury to the person who is drilling. The last thing which needs to be taken care of is to make sure that the vacuum is always on during drilling if possible which can absorb the dust particle and prevents them to get stuck in the eyes.

Another most important safety consideration is extreme caution must be taken while soldering because soldering iron gets hot and can easily burn ones' hand. When soldering makes sure the vacuum is on which absorbs all the harmful gasses from soldering. If possible, safety gloves must be always on when soldering. Make sure solder iron must be held from the top which prevents hands from burning and provides excellent quality soldering on the board.

While cutting a Veroboard it must also be kept in mind that all the edges of the Veroboard are smooth to touch. If possible filed the edges of the Veroboard and makes it smooth. If someone is making the PCB gloves must be worn which always protect from harmful liquids such as acid that is in the Etching process. While bending the aluminium sheet in the workshop it is necessary to hold the aluminium sheet from the middle and not from the edges as it could cause scrape to the skin. Safety shoes must be always on when working in the workshop because there is always a chance of falling a sharp piece of metal or a hammer on to the shoes. Loose clothing must be avoided while working in the workshop because in the workshop one is surrounded by different heavy types of machineries which can grab a loose cloth and cause severe damage to the worker.

When wire stripping makes sure it must be done gently and not to harm others by throwing the remains. When testing the LCD and PCB boards, make sure the area is clean from striped wires that could cause short circuits.

Chapter 6: Conclusion and Recommendations

This section of the report highlights the conclusion of the system and any recommendations that can make the system more efficient and useful for the end users. Or any additional circuitry that can be added into the existing system.

In this section, all the problems which were occurred during the project and the problem which still exist in the system will also be reported. Starting from the initial problems which were caused early in the project was choosing write components for the system, the most important choice was to choose a microcontroller which can perform several tasks at once and keeps the entire system on track. For this reason, Arduino Uno was selected which basically works on *Harvard Architecture* which means that it has separate buses for data transfer and instruction fetches, therefore there is no delay for fetching and executing within the Arduino. It has an Atmega328p onboard chip which has 32kb of flash memory, 2 kb of SRAM, 1kb of EPROM and operates on 16Mhz clock speed. As Arduino fulfils all the requirements needed for the project. For this reason, Arduino was selected for this project.

Another problem rises soon after when all the components were chosen and ordered. The Arduino was going to be short of digitals pins as most of the digital pins was occupied by the LCD screen and GSM which left Arduino with 3 usable digital pins, this problem was removed by using the latch circuitry which takes those 3 usable pins from Arduino and gives out 8 pins from its output these pins can treat exactly same as digital pins on Arduino.

As the system uses the Arduino GSM shield as a mode of communication between the Arduino and the recipient. The problem arises when the SIM card on the GSM is out of money and no longer sends SMS alert to the recipient which means that plants don't get water because system software design was written in such a way that plants do not get water until recipient sends SMS back to the GSM. This can be removed simply by changing the code like GSM sends the SMS alert no matter if is short of money. Arduino still does its job and plants get the water. But what if the water in the tank goes low and plants need water because of hot temperature GSM can only send an alert SMS. It doesn't have really care about the watering tank. The recommendation for this whole problem is to use a pressure sensor in the water tank and instead of using GSM get it replaced with any other board, for instance, a good recommendation is a raspberry pi or a Sigfox but to choose one of board massive amounts of research must be done and to see whether these architectures can solve this problem with ease. Another recommendation in terms of software is that software uses a ***delay ()*** function at the moment which is not as efficient from the engineer point of view. Use of the interrupts or timers is very crucial because it can save lots of the microcontroller work and does what it supposed to do when the interrupt occurs or when the timer expired.

References

Agarwal,T. (2015) Automatic Intelligent Plant Watering System Project Working, Available from: http://www.efxkits.us/automatic-plant-watering-system/ [Accessed 12 Dec. 2017].

Agarwal, T. (n.d.). Temperature Sensors – Types, Working & Operation. [Blog] Available at: https://www.elprocus.com/temperature-sensors-types-working-operation/ [Accessed 5 Apr. 2018].

Agarwal, T. (n.d.). Working Principle of Temperature Sensor and Its Application. [Blog] *Working Principle of Temperature Sensor and Its Application*. Available at: https://www.efxkits.us/lm35-temperature-sensor-circuit-working/ [Accessed 5 Apr. 2018].

Apiste-global.com. (n.d.). *Humidity sensor / Frequently Asked Questions (FAQ) / Apiste Corporation*. [online] Available at: http://www.apiste-global.com/faq/pau/detail/id=2502 [Accessed 9 Apr. 2018].

Agarwal, T. (n.d.). *What is GSM: Architecture and Working of GSM Module with Circuit*. [online] ElProCus - Electronic Projects for Engineering Students. Available at: https://www.elprocus.com/gsm-architecture-features-working/ [Accessed 14 Apr. 2018].

Agrawal, S. (2014) GSM Network Architecture [online], Available from: <https://tech.queryhome.com/51450/gsm-network-architecture> [Accessed Apr 14, 2018].

Arduino Corporation(1990). A000044 Reference Design | Antenna [online], Available from: <https://https//www.arrow.com/en/reference-designs/a000044-arduino-gsm-shield-integrated-antenna-allows-an-arduino-board-to-connect-to-the-internet-makereceive-voice-calls-and-sendreceive-sms-messages/f04948f3bfb192bedc1a06451b279de5> [Accessed Dec 8, 2017].

Arduino Uno (n.d.). *Arduino*. [image] Available at: https://www.rapidonline.com/pdf/73-4443.pdf [Accessed 8 Dec. 2017].

Content.smcetech.com. (2010). *Datasheet Solenoid valve pdf*. [online] Available at: http://content.smcetech.com/pdf/VXE_1090.pdf [Accessed 8 Dec. 2017].

Commons.wikimedia.org. (n.d.). *File:Thin Film PRT.png - Wikimedia Commons*. [online] Available at: https://commons.wikimedia.org/wiki/File:Thin_Film_PRT.png [Accessed 5 Apr. 2018].

Duzic, N. and Dumic, D. (2017). *Automatic Plant Watering System via Soil utomatic Plant Watering System via Soil Moisture Sensing by means of Suitable oisture Sensing by means of Suitable Electronics and its Applications for lectronics and its Applications for Anthropological and Medical Purposes*. [online] Collantropol.hr. Available at: http://www.collantropol.hr/antropo/article/viewFile/1564/1518 [Accessed 5 Feb. 2018].

Divani, D. (2016). *Automated plant Watering system - IEEE Conference Publication*. [online] Ieeexplore.ieee.org. Available at: http://ieeexplore.ieee.org/document/7557245/ [Accessed 13 Feb. 2018].

Devika, S. (2017). *Arduino Based Automatic Plant Watering System*. [online] Ijarcsse.com. Available at: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_4/10_October2014/V4I10-0228.pdf [Accessed 26 Feb. 2018].

Discover Sensors. (n.d.). *Discover Sensors*. [online] Available at: http://www.discoversensors.ie/sensors/sensors_in_everyday_life [Accessed 1 Mar. 2018].

Duff, M. and Towey, J. (2010). *Two Ways to Measure Temperature Using Thermocouples Feature Simplicity, Accuracy, and Flexibility | Analog Devices*. [online] Analog.com. Available at: http://www.analog.com/en/analog-dialogue/articles/measuring-temp-using-thermocouples.html [Accessed 5 Apr. 2018].

Digital Temperature Sensor. (n.d.). [ebook] pp.1 and 6. Available at: http://www.farnell.com/datasheets/2258678.pdf?_ga=2.131393803.544744636.1523300714-1174929237.1523300714&_gac=1.61922782.1523301015.Cj0KCQjwnqzWBRC_ARIsABSMVTOVbkZQBG4BIKKnzr7tuBAYa5VoDMsOB23QTghJOoMStpPiO_F-1VwaAoa7EALw_wcB [Accessed 9 Apr. 2018].

EEE PROJECTS. (n.d.). *Touch Sensor Working - EEE PROJECTS*. [online] Available at: https://eeeproject.com/touch-sensor/touch-sensor-working/ [Accessed 9 Apr. 2018].

Electronic tutorials. (n.d.). *Temperature Sensor*. [online] Available at: https://www.electronics-tutorials.ws/io/io_3.html [Accessed 3 Apr. 2018].

Engineering.nyu.edu. (n.d.). *Intro To Sensors*. [online] Available at: http://engineering.nyu.edu/gk12/amps-cbri/pdf/Intro%20to%20Sensors.pdf [Accessed 1 Dec. 2017].

Factory, F. (2018). *THERMISTOR NTC 10K PTH*. [online] FabLab Factory. Available at: https://www.fablabfactory.com/products/thermistor-ntc-10k-ohm-leaded [Accessed 3 Apr. 2018].

Futureelectronics.com. (n.d.). *what is a humidity, moisture sensors, temperature humidity sensor - Future Electronics*. [online] Available at: http://www.futureelectronics.com/en/sensors/humidity-dew.aspx [Accessed 9 Apr. 2018].

Finder (n.d.). *General Purpose Relay*. [image] Available at: http://ie.farnell.com/finder/32-21-7-005-2000/relay-spdt-250vac-6a/dp/1329651 [Accessed 7 Dec. 2017].

Jayjant (2015). *Automatic staircase lights using PIR sensor and Relays*. [image] Available at: https://circuitdigest.com/electronic-circuits/automatic-room-lights-using-pir-sensor-and-relay [Accessed 7 Dec. 2017].

Khatri, P. (n.d.). *Interfacing Thermistor with Arduino to Measure and Display Temperature on LCD*. [online] Circuitdigest.com. Available at: https://circuitdigest.com/microcontroller-projects/arduino-thermistor-interfacing-code-circuit [Accessed 3 Apr. 2018].

Kalwinder, K. 2013 'Soil Moisture Sensor Technology'. AZoSensors.Com,

Learn.adafruit.com. (2012). *Overview | DHT11, DHT22 and AM2302 Sensors | Adafruit Learning System*. [online] Available at: https://learn.adafruit.com/dht/overview [Accessed 9 Apr. 2018].

li, j. (2015). *Grove - Moisture Sensor User Manual*. 1st ed. [ebook] pp.3-4. Available at: http://www.mouser.com/ds/2/744/Seeed_101020008-838655.pdf [Accessed 6 Dec. 2017].

Naidu K (2011). Automatic Plant Irrigation System | Relay | Operational Amplifier [online], Available from: <https://www.scribd.com/doc/69745445/Automatic-Plant-Irrigation-System>,pg 3 [Accessed Dec 6, 2017].

Nikhil, A. (2010) 'GSM/GPRS Module'. 2. Available at: https://www.engineersgarage.com/articles/gsm-gprs-modules?page=2

Opengeospatial.org. (n.d.). *Sensor Web Enablement (SWE) | OGC*. [online] Available at: http://www.opengeospatial.org/ogc/markets-technologies/swe [Accessed 1 Mar. 2018].

Omega.com. (n.d.). *What is a thermistor? | Omega Engineering*. [online] Available at: https://www.omega.com/prodinfo/thermistor.html [Accessed 3 Apr. 2018].

Pots., P. (2016). *Plantpal Pack of 2 Large Self Watering Globes Plant Watering Spikes Aqua Spikes Automatic Potted Plant Watering System Vacation Watering. Use in 7 - 10 Inch Indoor Plant Pots. ~ Watering My Garden*. [online] Watering My Garden. Available at: http://www.wateringmygarden.com/product/plantpal-pack-of-2-large-self-watering-globes-plant-watering-spikes-aqua-spikes-automatic-potted-plant-watering-system-vacation-watering-use-in-7-10-inch-indoor-plant-pots/ [Accessed 26 Feb. 2018].

Pointing.spiraxsarco.com. (n.d.). *Controllers and Sensors : International site for Spirax Sarco*. [online] Available at: http://pointing.spiraxsarco.com/resources/steam-engineering-tutorials/control-hardware-el-pn-actuation/controllers-and-sensors.asp [Accessed 5 Apr. 2018].

Rotronic.com. (n.d.). *The Capacitive Humidity Sensor - ROTRONIC Measurement Solutions*. [online] Available at: https://www.rotronic.com/en/humidity_measurement-feuchtemessung-mesure_de_l_humidite/capacitive-sensors-technical-notes-mr [Accessed 9 Apr. 2018].

Robotics, S. (2017). Its watering time! – soil moisture sensor with arduino. [Blog] *Its watering time! – soil moisture sensor with arduino*. Available at: http://blog.srishtirobotics.com/blog/blog/2017/12/20/watering-time-soil-moisture-sensor-arduino/ [Accessed 9 Apr. 2018].

Scribd. (2018). *Arduino Based Automatic Plant Irrigation System With Message Alert | Arduino | General Packet Radio Service*. [online] Available at: https://www.scribd.com/document/371585956/Arduino-Based-Automatic-Plant-Irrigation-System-With-Message-Alert [Accessed 1 Mar. 2018].

Sumeeteshop (n.d.). *Serial LCD 2x16*. [image] Available at: http://shop.sumeetinstruments.com/index.php?route=product/product&product_id=413 [Accessed 7 Dec. 2017].

Technet.pnnl.gov. (n.d.). *PNNL: S & E - Biological Property Sensors Mechanism Capabilities*. [online] Available at: https://technet.pnnl.gov/sensors/biological/capabilities/ [Accessed 1 Mar. 2018].

Thermo-kinetics.com. (n.d.). *Resistance Temperature Detectors | RTDs | Thermo Kinetics*. [online] Available at: http://www.thermo-kinetics.com/resistance-temperature-detectors [Accessed 5 Apr. 2018].

Tsulo.com. (n.d.). *Tsulo.com*. [online] Available at: http://tsulo.com/g/0449_003/ [Accessed 9 Apr. 2018].

Urban Organic Yield. (n.d.). *5 Best Hydroponics Kits To Grow Your Indoor Herb Garden All Year Long*. [online] Available at: https://www.urbanorganicyield.com/best-hydroponics-kits/ [Accessed 26 Feb. 2018].

Vakits.com. (2012). *TIP32C PNP Power Transistor | NightFire Electronics LLC*. [online] Available at: http://vakits.com/tip32c-pnp-power-transistor-220 [Accessed 8 Dec. 2017].

Waveshare (n.d.). *DHT11 Temperature-Humidity Sensor Module*. [image] Available at: http://www.robotshop.com/uk/dht11-temperature-humidity-sensor-module.html [Accessed 7 Dec. 2017].

APPENDIX A

Sample Code to test Solenoid Valve:

```
#define relay1 3
#define relay2 4
#define relay3 5
#define relay4 6

void setup()
{
  // put your setup code here, to run once:
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);
  pinMode(relay3, OUTPUT);
  pinMode(relay4, OUTPUT);
  digitalWrite(relay1, LOW);
  digitalWrite(relay2, LOW);
  digitalWrite(relay3, LOW);
  digitalWrite(relay4, LOW);
}

void loop()
{
  digitalWrite(relay1, HIGH);
  delay(2000);

  digitalWrite(relay2, HIGH);
  delay(2000);
  digitalWrite(relay2, LOW);
  delay(2000);

  digitalWrite(relay3, HIGH);
  delay(2000);
  digitalWrite(relay3, LOW);
  delay(2000);

  digitalWrite(relay4, HIGH);
  delay(2000);
  digitalWrite(relay4, LOW);
  delay(2000);

  digitalWrite(relay1, LOW);
  delay(2000);

}
```

APPENDIX B

Code for testing System Sensors:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd (8, 9, 10, 11, 12, 13);

#include <dht.h>
#define dht_dpin A0
dht DHT;

int plantPotMoisture[3] = {A1, A2, A3};

void setup()
{
  // put your setup code here, to run once:
  lcd.begin (16, 2);
  lcd.setCursor(0,0);
  lcd.print("Readings");
  //lcd.print("Humidity and Temperature");
  delay(2000);
}

void loop()
{
  // put your main code here, to run repeatedly:

    DHT.read11(dht_dpin);
    lcd.setCursor(0, 0);
    lcd.print("Humidity=");
    lcd.print(DHT.humidity);
    lcd.print("%");
    lcd.setCursor(0, 1);
    lcd.print("temp=");
    lcd.print(DHT.temperature);
```

```
    lcd.print("C  ");
    delay(3000);
    lcd.clear();

  for (int i = 0 ; i < 3; i++)
  {
    lcd.clear();
    plantPotMoisture[i] = analogRead(i);
    plantPotMoisture[i] = map(plantPotMoisture[i], 550, 0, 0, 100);
    Serial.print("Mositure" + i );
    lcd.print("Mositure" + i);
    Serial.print(plantPotMoisture[i]);
    lcd.print(plantPotMoisture[i]);
    Serial.println("%");
    lcd.print("%");
    delay(1000);
  }
}
```

APPENDIX C

GSM Modem Test Code:

```
#include <GSM.h>

GSMModem modem;

String IMEI = "";


void setup() {

  Serial.begin(9600);

  while (!Serial)


  Serial.print("Starting modem test...");

  if (modem.begin()) {

    Serial.println("modem.begin() succeeded");

  } else {

    Serial.println("ERROR, no modem answer.");

  }

}


void loop() {

  Serial.println("Checking IMEI...");

  IMEI = modem.getIMEI();


  if (IMEI != NULL) {

    Serial.println("Modem's IMEI: " + IMEI);


    Serial.print("Resetting modem...");

    modem.begin();


    if (modem.getIMEI() != NULL) {

      Serial.println("Modem is functoning properly");

    } else {
```

```
      Serial.println("Error: getIMEI() failed after
modem.begin()");
    }
  } else {
    Serial.println("Error: Could not get IMEI");
  }
}
```

GSM Network Connection Code:

```
#include <GSM.h>
#define PINNUMBER ""

GSM gsmAccess(true);
GSMScanner scannerNetworks;
GSMModem modemTest;

String IMEI = "";
String errortext = "ERROR";

void setup()
{
  Serial.begin(9600);
  Serial.println("GSM networks scanner");
  scannerNetworks.begin();

  Serial.print("Modem IMEI: ");
  IMEI = modemTest.getIMEI();
  IMEI.replace("\n","");
  if(IMEI != NULL)
    Serial.println(IMEI);

  Serial.print("Current carrier: ");
  Serial.println(scannerNetworks.getCurrentCarrier());
```

```
  Serial.print("Signal Strength: ");
  Serial.print(scannerNetworks.getSignalStrength());
  Serial.println(" [0-31]");
}

void loop()
{
  Serial.println("Scanning available networks. May take some
seconds.");

  Serial.println(scannerNetworks.readNetworks());

  Serial.print("Current carrier: ");
  Serial.println(scannerNetworks.getCurrentCarrier());

  Serial.print("Signal Strength: ");
  Serial.print(scannerNetworks.getSignalStrength());
  Serial.println(" [0-31]");
}
```

GSM sends SMS alert Code:

```
#include <GSM.h>
#define PINNUMBER ""
GSM gsmAccess;
GSM_SMS sms;
char remoteNumber[20]= "0899506304";
String txtmsg="Hi Uzair!";

void setup()
{
  Serial.begin(9600);
  Serial.println("SMS Messages Sender");
  boolean notConnected = true;
```

```
  while(notConnected)
   {
    if(gsmAccess.begin(PINNUMBER)==GSM_READY)
      notConnected = false;
    else
    {
      Serial.println("Not connected");
      delay(1000);
    }
  }
  Serial.println("GSM initialized");
  sendSMS();
}
```

GSM Receive SMS code:

```
#include <GSM.h>
#define PINNUMBER ""
GSM gsmAccess;
GSM_SMS sms;
String messageBuffer = "";
char senderNumber[20];
void setup() {
   pinMode(13, OUTPUT);
  Serial.begin(9600);
  Serial.println("SMS Messages Receiver");
  Serial.println("GSM initialized");
  Serial.println("Waiting for messages");
}

void loop() {
 recieveSMS();
}
```

```
void recieveSMS()
{
 char c;
  if (sms.available()) {
    Serial.println("Message received from:");
    sms.remoteNumber(senderNumber, 20);
    Serial.println(senderNumber);
    while (c = sms.read()) {
      Serial.println(c);
      messageBuffer += c;
    }
    Serial.println(messageBuffer);
    if (messageBuffer == "yes")
    {
      digitalWrite(13, HIGH);
      delay(1000);
      digitalWrite(13, LOW);
      delay(1000);
    }
     messageBuffer = "";
    Serial.println("\nEND OF MESSAGE");
    sms.flush();
    Serial.println("MESSAGE DELETED");
  }
  delay(1000);
}
```

APPENDIX D

Final Software code:

```
#include <dht.h>

#define dht_dpin A0

dht DHT;

//-----------------------

#include <LiquidCrystal.h>

LiquidCrystal lcd (8, 9, 10, 11, 12, 13);

//----------------------------------

int plantPotMoisture[3] = {A1, A2, A3};

//---------------------

#include <GSM.h>

#define PINNUMBER ""

GSM gsmAccess; // include a 'true' parameter for debug enabled

GSM_SMS sms;

char remoteNumber[] = "0899506304";

String moistureMessage = "Moisture is Low on sensor: ";

String SMS_Alert = "Sending SMS!";

String humidityMsg = "Humidity is High. Open All Solenoids";

String tempMsg = "Temperature is too HIGH!..Open ALl Solenoids ";

String messageBuffer = "";

char senderNumber[20];


String stringOne = "Opens1";

String stringTwo = "Opens2";

String stringThree = "Opens3";

String stringFour = "OpenAll";


//---------------

#define solenoidData 5

#define solenoidClockster 4

#define solenoidLatch 6

//---------------
```

```
const int master = 0;
const int slave1 = 1;
const int slave2 = 2;
const int slave3 = 3;

boolean takeReadings = true;

int serialSolenoidOutput = 0;

void setup()
{
  pinMode(solenoidData, OUTPUT);
  pinMode(solenoidClockster, OUTPUT);
  pinMode(solenoidLatch, OUTPUT);

  digitalWrite(solenoidLatch, HIGH);

  digitalWrite(solenoidLatch, LOW);
  shiftOut(solenoidData, solenoidClockster, MSBFIRST, 0);
  digitalWrite(solenoidLatch, HIGH);

  //---------------------------
  Serial.begin(9600);
  lcd.begin (16, 2);
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Wait Until");
  lcd.setCursor(0, 1);
  lcd.print("GSM Initialized!");
  boolean notConnected = true;
  while (notConnected)
  {
    if (gsmAccess.begin(PINNUMBER) == GSM_READY)
```

```
        notConnected = false;
      else
      {
        Serial.println("Not connected");
        delay(1000);
      }
    }
}

void loop()
{
  if (takeReadings)
  {
    moistureSensor();
    TempAndHumidity ();


    if (DHT.humidity > 50 || DHT.temperature > 25 && takeReadings )
    {
      takeReadings = false;

      if (DHT.humidity > 50)
      {
        sendSMS(humidityMsg);
      }
      else if (DHT.temperature > 25)
      {
        sendSMS(tempMsg);
      }


      while (!takeReadings)
        recieveSMS();
    }
```

```
    if (plantPotMoisture[0] > 30 || plantPotMoisture[1] > 30 ||
plantPotMoisture[2] > 30  && takeReadings)
    {
      takeReadings = false;

      if (plantPotMoisture[0] > 30)
      {
        sendSMS(moistureMessage + "1");
      }
      else if (plantPotMoisture[1] > 30)
      {
        sendSMS(moistureMessage + "2");
      }
      else
      {
        sendSMS(moistureMessage + "3");
      }

      while (!takeReadings)
        recieveSMS();
    }

  }
}

void moistureSensor()
{
  for (int i = 0 ; i < 3; i++)
  {
    lcd.clear();
    plantPotMoisture[i] = analogRead(i);
    plantPotMoisture[i] = map(plantPotMoisture[i], 550, 0, 0, 100);
    Serial.print("Mositure" + i );
```

```
    lcd.print("Mositure" + i);
    Serial.print(plantPotMoisture[i]);
    lcd.print(plantPotMoisture[i]);
    Serial.println("%");
    lcd.print("%");
    delay(1000);
  }

}

void TempAndHumidity ()
{
  DHT.read11(dht_dpin);
  lcd.setCursor(0, 0);
  lcd.print("Humidity=");
  Serial.print("Current humidity = ");
  Serial.print(DHT.humidity);
  lcd.print(DHT.humidity);
  lcd.print("%");
  Serial.print("%");
  Serial.print("temperature = ");
  Serial.print(DHT.temperature);
  Serial.println("C");
  lcd.setCursor(0, 1);
  lcd.print("temp=");
  lcd.print(DHT.temperature);
  lcd.print("C  ");
  delay(1000);
  lcd.clear();
}

void sendSMS(String messageToSend)
{
  Serial.print("Sending a message to mobile number: ");
```

```
  Serial.println(remoteNumber);

  Serial.println("SENDING");
  lcd.print(SMS_Alert);
  Serial.println();
  Serial.println("Message:");

  Serial.println(messageToSend);

  sms.beginSMS(remoteNumber);
  sms.print(messageToSend);
  sms.endSMS();

  Serial.println("\nCOMPLETE!\n");
  lcd.clear();
  lcd.print("Completed!!!");
}


void recieveSMS()
{
  char c;

  if (sms.available())
  {
    lcd.clear();
    lcd.print("Message received from:");
    delay(800);
    lcd.clear();



    sms.remoteNumber(senderNumber, 20);
    lcd.print(senderNumber);
```

```
while (c = sms.read())
{
  Serial.println(c);
  messageBuffer += c;
}

Serial.println(messageBuffer);

if (messageBuffer == stringOne)
{
  toggleSolenoid1();

  takeReadings = true;
}
else if (messageBuffer == stringTwo)
{
  toggleSolenoid2();
  takeReadings = true;
}
else if (messageBuffer == stringThree)
{
  toggleSolenoid3();
  takeReadings = true;
}
else if (messageBuffer == stringFour)
{
  toggleAll();
  takeReadings = true;
}
else
{
  takeReadings = true;
}
```

```
    messageBuffer = "";

    Serial.println("\nEND OF MESSAGE");

    // Delete message from modem memory
    sms.flush();
    Serial.println("MESSAGE DELETED");
  }
  delay(1000);
}

void toggleSolenoid1()
{
  solenoidWrite(master, HIGH);
  delay(1000);
  solenoidWrite(slave1, HIGH);
  delay(1000);
  solenoidWrite(slave1, LOW);
  delay(1000);
  solenoidWrite(master, LOW);
  delay(1000);
}

void toggleSolenoid2()
{
  solenoidWrite(master, HIGH);
  delay(1000);
  solenoidWrite(slave2, HIGH);
  delay(1000);
  solenoidWrite(slave2, LOW);
  delay(1000);
  solenoidWrite(master, LOW);
  delay(1000);
}
```

```
void toggleSolenoid3()
{
  solenoidWrite(master, HIGH);
  delay(1000);
  solenoidWrite(slave3, HIGH);
  delay(1000);
  solenoidWrite(slave3, LOW);
  delay(1000);
  solenoidWrite(master, LOW);
  delay(1000);
}
void toggleAll()
{
  solenoidWrite(master, HIGH);
  delay(1000);
  solenoidWrite(slave1, HIGH);
  delay(1000);
  solenoidWrite(slave2, HIGH);
  delay(1000);
  solenoidWrite(slave3, HIGH);
  delay(1000);
  solenoidWrite(slave1, LOW);
  delay(1000);
  solenoidWrite(slave2, LOW);
  delay(1000);
  solenoidWrite(slave3, LOW);
  delay(1000);
  solenoidWrite(master, LOW);
  delay(1000);
}

void solenoidWrite(int pin, bool state)
{
  if ( pin >= 0 &&  pin < 8)
```

```
  {
    if (state)
      serialSolenoidOutput |= (1 << pin);
    else
      serialSolenoidOutput &= ~(1 << pin);
  }

  digitalWrite(solenoidLatch, LOW);
  shiftOut(solenoidData, solenoidClockster, MSBFIRST,
serialSolenoidOutput);
  digitalWrite(solenoidLatch, HIGH);
}
```

Timetable