AUTOMATIC PLANT WATERING SYSTEM USING ARDUINO

Khin Thandar Tun[1], *Hay Man Oo*[2], *Cho Thet Nwe*[3]

[1]*Lecturer, Department of Electronic Engineering, Technological University (Meiktila), Myanmar*
[2]*Lecturer, Department of Electronic Engineering, Technological University (Meiktila), Myanmar*
[3]*Assistant Lecturer, Department of Electronic Engineering, Technological University (Kyaukse), Myanmar*

Abstract

Nowadays gardening is one of the sophisticated hobby and also very important for environment balance. But the main problem is the time adjustment for watering the plant. Especially when people are away from home for few days, then lots of problem arises to watering the plants. In this technological era where people rely on the technology for almost everything to make their work easier and more efficient, then why not there has been a system that automatically waters the garden and plants without depending on someone when no one is there at home for longer period. So one of the objectives of this work is to observe how human activities are replaced from automated irrigation and also excess use of water is minimized. The method is employed, to continuously keep an eye on the soil moisture level to decide whether irrigation is needed, and how much water is needed for the soil. Program code for this paper is based on Arduino UNO with C++ language. If dryness level is high, moisture level is low. At that time, the water pump is switched ON to supply water to the plant and is automatically OFF when the soil moisture reaches to the minimum threshold level. The total work is done through six sub systems; soil moisture sensor, motor driver, servo motor, liquid crystal display, LCD I2C module and water pump. In this thesis, the micro water pump as pumping unit is constructed by using a small DC motor. An integrated Liquid Crystal Display (LCD) is also used for real time display of data acquired from the sensor and the status of the various devices.

Keyword: Arduino UNO, C Programming, Sensors, LCD, Water pump, Motors, L293D motor driver, LCD I2C module

1.INTRODUCTION

Irrigation is an artificial way of watering the soil for the proper growth of the plant. It is mainly used in the dry areas and the places where rainfall is less. Irrigation also helps to suppress the weeds growing in the agricultural fields. The old methods used for irrigation was manual irrigation using buckets and watering cans, by using sprinkler irrigation, localized irrigation, drip irrigation, etc. But by using these techniques people hasn't predicted the amount of water that is to be watered or the sufficient quantity of water that a crop needs. Due to this water logging occur while using these techniques to water the crops. So there is a need for the improvement on these existing techniques in order to conserve water. So to prevent the water which is being wasted during irrigation, an automatic plant watering system has been developed. This works by sensing the moisture content of the soil and decides whether the pump has to operate or not. The water supply needed for this irrigation is from any source like pond, stream, well etc. This system is not as expensive as people compare to other systems and is time saving, as the works done by the system is automatic. In the world of advance electronics, life of human beings is simpler.

Hence to make life simpler and more convenient, automatic plant watering system using Arduino has been made. Water is a resource that is needed by all living species. So it is necessary to check the water usage and preserve it for the future generation to come. A large amount of water resource is wasted also by this country. So in paper an efficient automatic plant watering system is designed which check the water usage and also reduce the human supervision required. The model implements sensor technology with Arduino microcontroller to make a smart switching device. The system displays the basic switching mechanism of motor and pump using sensors. This system avoid over watering of already saturated soil. This system is implemented easily and also it is cost effective. It is user friendly and has been easily reprogrammed for any requirement.

2.SYSTEM BLOCK DIAGRAM

The automatic plant watering system is designed to continuously check the soil moisture level. Firstly three most common soil samples (sandy, loamy and clayey soils) are used as a reference level of soil moisture. Probes are set up to the sensors. The sensor consists of two probes. This sensor measures the volumetric content of water in soil and gives the moisture level. Then stops the supplying water when the level of soil moisture is achieved which it desired. As a reference of voltage levels respective to the wet and dry states of the soil sample are computed by measuring the resistance through the moisture detector probes. This paper uses Arduino UNO board circuit for controlling whole the process of this automatic plant watering system. If moisture level is low then Arduino switches ON a water pump and rotates the servo motor to provide water to the plant. This board is connected with sensor, motor driver circuit, servo motor, LCD screen and power supply. The motor driver circuit uses as a L293D in this paper. This driver circuit controls the whole work. It sense the soil condition, it measures the time for supplying the water. This Arduino board, driver circuit is connected with power supply for active the circuit.

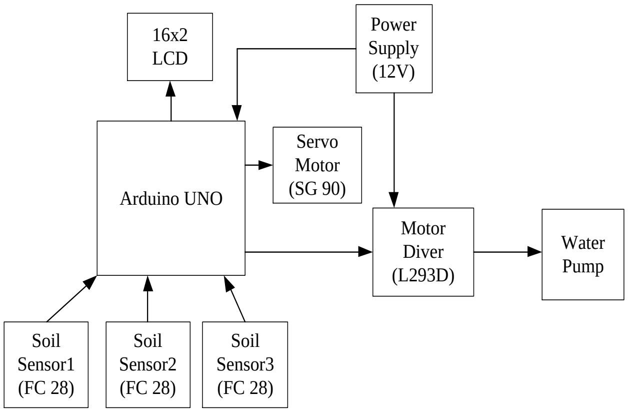


Fig 1: Block Diagram of Automatic Plant Watering System

The flowchart shown in Fig.2 is for the operation of the system.

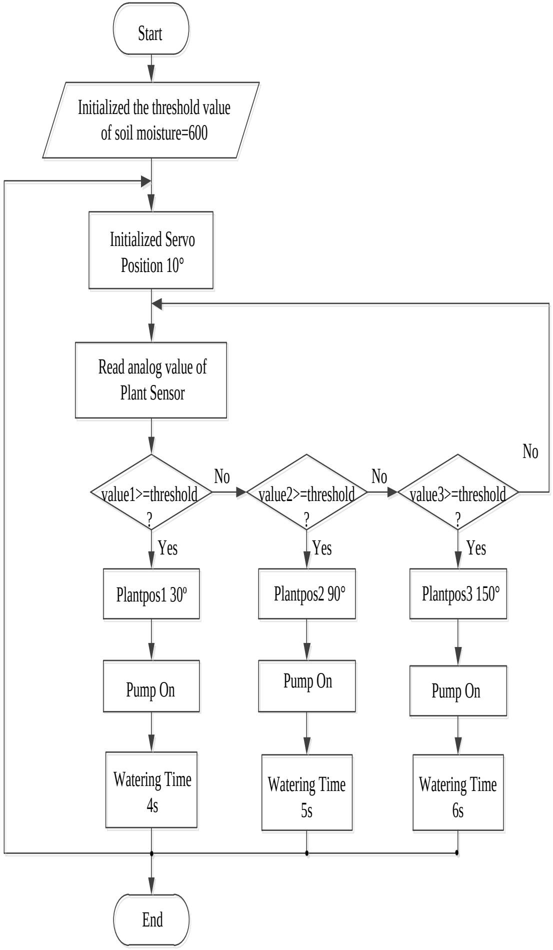


Fig 2: Flowchart of the System

The algorithm used in the flowchart.

- Step 1: Start.
- Step 2: Set the threshold value of soil moisture = 600 that means before three sensors are not working.
- Step 3: Read analog value of plant sensor from the sensors in the soil.
- Step 4: Check the sensor's value in the soil. If value1 is greater than the threshold value, it means go to step 5, 6 and step 7. If value1 is not greater than the threshold value, it goes to step 9.

- Step 5: Rotate the servo motor Plantpos1 at 30° for plant 1.
- Step 6: After rotate the servo motor for plant 1, the pump is on.
- Step 7: Defining watering time (4s) for plant 1 is working.
- Step 8: End.
- Step 9: Check the sensor's value in the soil. If value2 is greater than the threshold value, it means go to 10, 11 and step 12. If value2 is not greater than the threshold value, it goes to step 14.
- Step 10: Rotate the servo motor Plantpos2 at 90° for plant 2.
- Step 11: After rotate the servo motor for plant 2, the pump is on.
- Step 12: Defining watering time (5s) for plant 2 is working.
- Step 13: End.
- Step 14: Check the sensor's value in the soil. If value3 is greater than the threshold value, it means go to 15, 16 and step 17. If value3 is not greater than the threshold value, it goes to step3.
- Step 15: Rotate the servo motor Plantpos3 at 150° for plant 3.
- Step 16: After rotate the servo motor for plant 3, the pump is on.
- Step 17: Defining watering time (6s) for plant 3 is working.
- Step 18: End.
- Step 19: If the process wants to begin again, go to step 2.
- Step 20: If the process wants to finish, go to End.

3.IMPLEMENTATION

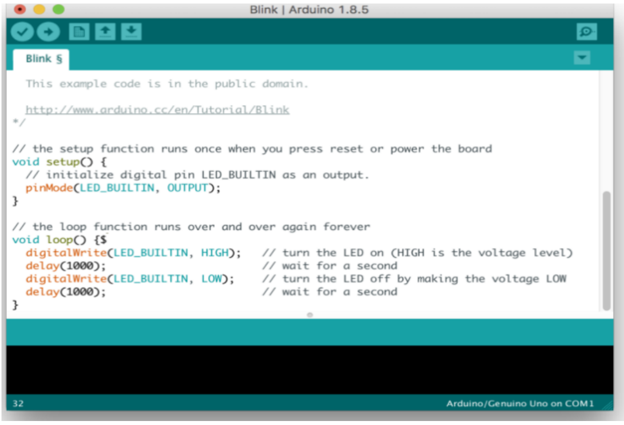
3.1. Implementation using Arduino UNO

With Arduino UNO and a USB A and B cable at hand, follow these steps to obtain and install the latest version of Arduino on version of Windows;

- Open the Arduino downloads page and click the Windows link to download the zip file containing a copy of the Arduino application for Windows.
- Plug the square end of the USB cable into the Arduino and the flat end into an available port on PC to connect the Arduino to computer.
- Open the Start Menu and type devmgmt.msc in the Search Programs and Files box; then press Enter.
- Right click Arduino UNO and select Update Driver Software in the list that appears; then click the Browse Computer for Driver Software option.
- Click Browse to find Arduino folder.
- Within Arduino folder, click the Drivers folder and then click the Arduino UNO file.
- Click Next, and Windows completes the installation.

The Arduino IDE supports the languages C and C++ using special rules of code structuring. The Arduino IDE supplies a software library from the Wiring project, which provides many common input and output procedures. User written code only requires two basic functions, for starting the sketch and the main program loop, that are compiled and linked with a program stub main () into and executable cyclic executive program with the GNU toolchain, also included with the IDE distribution.

- setup () : This function is called once when a sketch starts after power up or reset. It is used to initialize variables, input and output pin modes and other libraries needed in the sketch.
- loop() : After setup () function exits (ends), the loop () function is executed repeatedly in the main program. It controls the board until the board is powered OFF or is reset.



```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Blink
*/

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {$
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Fig 3: Architecture of Arduino UNO's IDE

The code for this paper, the schematics is fairly easy. It is heavily reliant on the servo.h library which comes pre-installed with the Arduino IDE. The goal implemented by the code is simple, turn to the servo motors in a different direction using different angles to demonstrate its capabilities.

To do an explanation of the code, as usual, the first thing do is included the libraries the servo.h library.

```
#include<Servo.h>
```

Next, an object of the servo library is created.

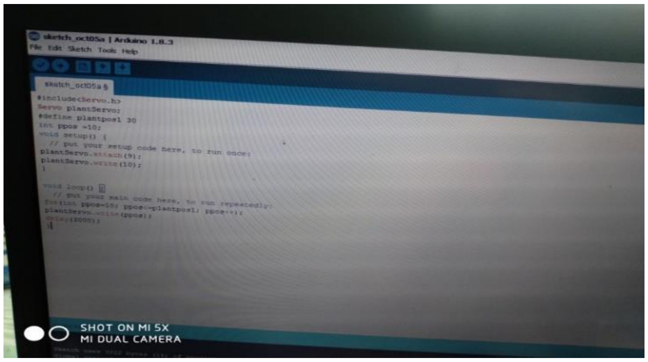
```
Servo plantServo;
```

Next, proceed to the void setup function is moved where the servo object is attached to a digital pin and set the servo angle to 10. Digital pin 9 to the servo object is attached. The servo object is now fully initialized and ready to control the servo.

```
void setup ()
{
plantservo.attach(9);
        plantservo.write(10);
}
```

Next is the loop function. To turn make the servo rotate to a particular degree, the plantservo.write () function is used to write the degree of rotation to the servo is desired. The function to turn the servo is used from one to the other and back. To demonstrate this, the "for" loop is used to turn the servo at several angles in one direction, and the another loop is used to turn the servo back to where it started.

```
void loop ()
{
        for (int ppos=10; ppos<=plantpos1; ppos++)
        plantServo.write(ppos);
        delay(3000);
}
```

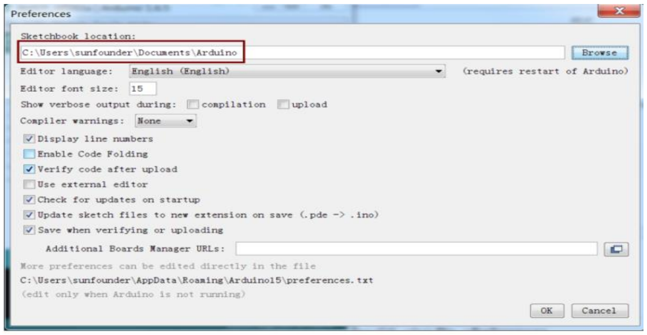


```
#include<Servo.h>
Servo plantServo;
#define plantpos1 30
int ppos =10;
void setup() {
  // put your setup code here, to run once:
plantServo.attach(9);
plantServo.write(10);
}

void loop() {
  // put your main code here, to run repeatedly:
for(int ppos=10; ppos<=plantpos1; ppos++);
plantServo.write(ppos);
delay(3000);
}
```

Fig 4: Program for Servo Motor

Since in some code, the libraries needed are not included in Arduino, so to add them before compiling. Unzip the downloaded file. Copy the folders under the Library folder to the libraries folder in Arduino, if the path in Arduino is not finded, open Arduino IDE, click File > Preference, and the path in the Browse box is seen, as shown in the following diagram. Compile the program.



Sketchbook location: C:\Users\sunfounder\Documents\Arduino
Editor language: English (English) (requires restart of Arduino)
Editor font size: 15
Show verbose output during: □ compilation □ upload
Compiler warnings: None
☑ Display line numbers
□ Enable Code Folding
☑ Verify code after upload
□ Use external editor
☑ Check for updates on startup
☑ Update sketch files to new extension on save (.pde -> .ino)
☑ Save when verifying or uploading
Additional Boards Manager URLs:
More preferences can be edited directly in the file
C:\Users\sunfounder\AppData\Roaming\Arduino15\preferences.txt
(edit only when Arduino is not running)

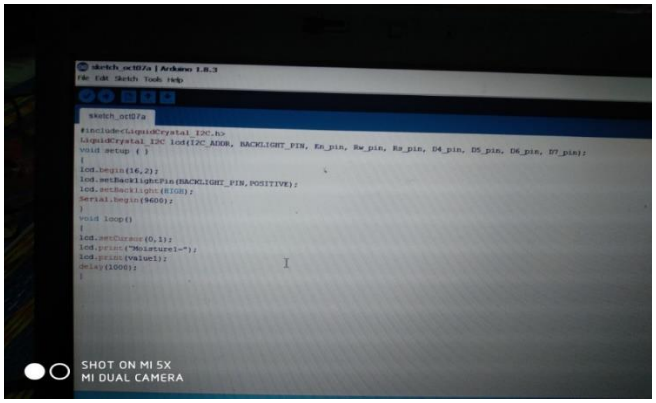
Fig 5: Arduino IDE

The default I2C bus address is 0x27 and the first parameter in the LiquidCrystal I2C function. This function sets the dimensions of the LCD. It needs to be placed before any other LiquidCrystal function in the void setup () section of the program. The number of rows and columns are specified as lcd.begin (columns, rows). For 16x2 LCD, lcd.begin (16, 2) is seen, and for a 20x4 LCD, lcd.begin (20, 4) is used. The backlight is also be turned on and off with lcd.setBacklight(HIGH) or lcd.setBacklight (LOW).

Similar, but more useful than lcd.home () is lcd.setCursor(). This function places the cursor and any printed text at any position on the screen. It is used in the void setup () or void loop () section of program. The cursor position is defined with lcd.setCursor (column, row). The column and row coordinates start from zero (0 to 15 and 0 to 1 respectively). For example, using lcd.setCursor(0,1) in the void setup () section of the "Moisture1=" program above prints "Moisture1=" to the lower line and shifts it to the right two spaces.

```
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(I2C_ADDR, BACKLIGHT_PIN, En_pin, Rw_pin, Rs_pin, D4_pin, D5_pin, D6_pin, D7_pin);
void setup ()
```

```
{
lcd.begin(16,2);
lcd.setCursor(0,1);
lcd.print("Moisture1=");
lcd.print(value1);
delay(1000);
}
```



```
sketch_oct07a | Arduino 1.8.3
#include<LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(I2C_ADDR, BACKLIGHT_PIN, En_pin, Rw_pin, Rs_pin, D4_pin, D5_pin, D6_pin, D7_pin);
void setup ( )
{
lcd.begin(16,2);
lcd.setBacklightPin(BACKLIGHT_PIN,POSITIVE);
lcd.setBacklight(HIGH);
Serial.begin(9600);
}
void loop()
{
lcd.setCursor(0,1);
lcd.print("Moisture1=");
lcd.print(value1);
delay(1000);
}
```

Fig 6: Program for LCD

The LiquidCrystal () function sets the pins the Arduino uses to connect to the LCD. Any of the Arduino's digital pins are used to control the LCD.

Put the Arduino pin numbers inside the parentheses in this order: LiquidCrystal (RS, E, D4, D5, D6, D7). RS, E, D4, D5, D6, D7 are the LCD pins.

This function is used to print text to the LCD. It is used in the void setup () section or the void loop () section of the program.

To print letters and words, place quotation marks (" ") around the text. For example, print "Moisture1=" use lcd.print ("Moisture1=").

This function clears any text or data already displayed on the LCD. If lcd.clear () with lcd.print () is used and the delay () function in the void loop () section, a simple blinking text program is made.

```
lcd.print("Moisture1=");
delay(2000);
lcd.clear( );
```

Two variables have been defined, one for the soil moisture sensor pin and the other for storing the output of the sensor.

```
int plantSensor1=A0;
int value1=0;
```

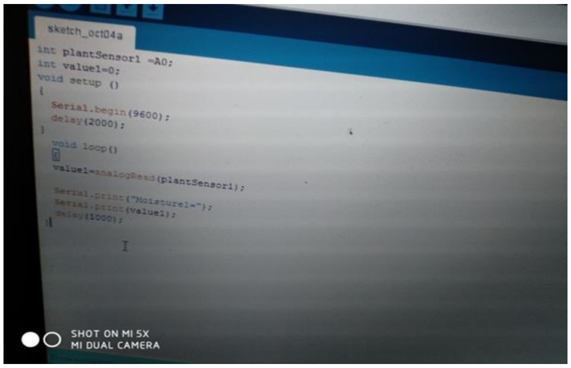
In the setup function, the "Serial.begin (9600)" command initializing the serial port of Arduino to display information in the serial monitor.

```
void setup () {
Serial.begin (9600);
}
```

In the loop function, the sensor analog pin is read and is store the values in the "value" variable. While monitoring the sensor ADC values are got from 0 to 1023.

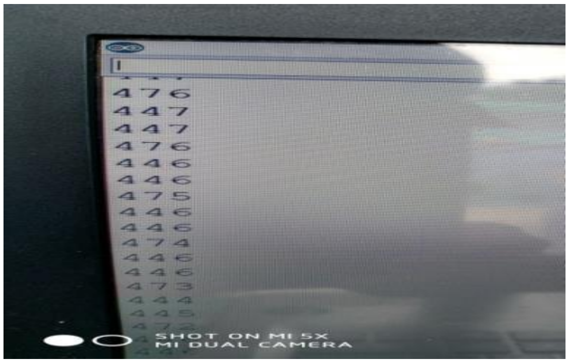
And the values display on LCD screen. Different ranges of the moisture values are set and turn ON or OFF the water pump according to it.

```
void loop ()
{
        value1=analogRead(plantSensor1);
        Serial.print("Moisture1=");
        Serial.print(value1);
delay(1000);
}
```



```
sketch_oct04a
int plantSensor1 =A0;
int value1=0;
void setup ()
{
  Serial.begin(9600);
  delay(2000);
}
  void loop()
  {
  value1=analogRead(plantSensor1);
   Serial.print("Moisture1=");
   Serial.print(value1);
   delay(1000);
}
```

Fig 7: Program for Soil Moisture Sensor of Plant 1



```
476
447
447
476
446
446
475
446
446
474
446
446
473
444
445
```

Fig 8: Serial Monitor for Soil Moisture Sensor of Plant 1

At initial state, all three sensors are not working and sensor's values are zero.

```
int value1=0;
int value2=0;
int value3=0;
```

The setup function is declared servo position at 10°.

```
plantServo.write (10);
```

And pump situation is LOW that means at initial state for pump is off. digitalWrite () programming language is used.

Outputs either logic level HIGH or LOW (turns ON or OFF) at a specified digital pin. The pin is specified as either a variable or constant (0 to13).

```
digitalWrite (pump, LOW);
```

3.2. Implementation by Hardware

Working process of automatic plant watering system is very simple. It is fully automated process. In this paper, it is working the five sections. The first section is power supply which is used 12V module. The second sections are sensors which is used soil moisture sensor. The third section is Arduino UNO. The four sections are LCD to display the sensor's value which is used together with LCD I2C module. The final sections are motor driver which is used IC L293D. It controls the servo motor which rotates and the water pump which is pumping unit. Arduino UNO and motor driver are supplied 12V power supply. Others soil moisture sensor, LCD, servo motor and water pump are supplied 5V from Arduino UNO. Initial position for servo motor horn is 10° position.

Soil moisture sensor has 3 connection points where V_{CC} is connected with the 5V of Arduino board, the GND pin is connected to the GND of Arduino and the output pins are connected to A0, A1 and A2 of Arduino. The first the moisture sensor senses the amount of moisture presence in the soil and accordingly it sends its output to the microcontroller through the input and output pins provided in the Arduino board. Then the sensor obtains the data from the soil and display on the LCD screen. The four bit mode LCD module is used. It has four connections namely VCC, GND, SDA, SCL. VCC pin of the LCD is connected to the 5V pin of the Arduino. GND pin is connected to the GND pin of the Arduino. SDA pin is connected to pin of Arduino which is the analog input pin (A4), and finally the SCL terminal is connected to the Arduino UNO board which is another analog input terminal (A5). The soil moisture sensor continuously monitors the volumetric moisture content in the soil and sends that information to the LCD and Arduino with values lying in the range 0 to 1023. And display on LCD " Moisture1 = " and "Watering Plant1".This data compared with the threshold value that means minimum soil moisture value from where soil completely dry and to need watering the soil. This value is already referred in the program and when the value reaches to the threshold value it sent the signal to motor driver IC.

The enable signal, pin 1 of L293D driver chip is connected to 5V. This driver chip main voltage means that pin 8 of the chip is connected to the positive terminal of the power supply. Pin 2 is connected to the negative terminal of the power supply. Pin 4, 5, 12, 13 are connected with ground and short to the GND pin of the Arduino UNO board. Pin 16 is connected to 5V. Pin 7 is connected to Arduino's digital pin 3. Then it rotates the servo motor and turns the water pump on simultaneously. It first moves the horn of the servo motor along with the pipe which is attached to it towards the plant whose moisture level decreased. The servo motor is connected to the 9th number output pin of Arduino board through which the microcontroller is giving instruction to the motor. Servo motor for plant 1 rotates the 30° position.

The positive terminal of the water pump is connected to pin 6 of the IC motor driver and the negative terminal is connected to pin 3 of the chip. The pump turn on about watering time which is also mentioned in the program and the process accordingly in the program for plant 1, plant 2, and then plant 3 step by step. It starts the water pump to supply water to the plant for 4s and then stops the water pump. It brings back the servo motor horn to its initial position. Sensor 2 and sensor 3 are also above principle. For sensor 2, display on LCD "Moisture2 = " and "Watering Plant 2", watering time is 5s and servo motor horn rotates 90° position. Then servo motor brings back to its initial position. For sensor 3, display on LCD " Moisture3 = " and "Watering Plant3", watering time is 6s and servo motor horn rotates 150° position. Then servo motor brings back to its initial position.

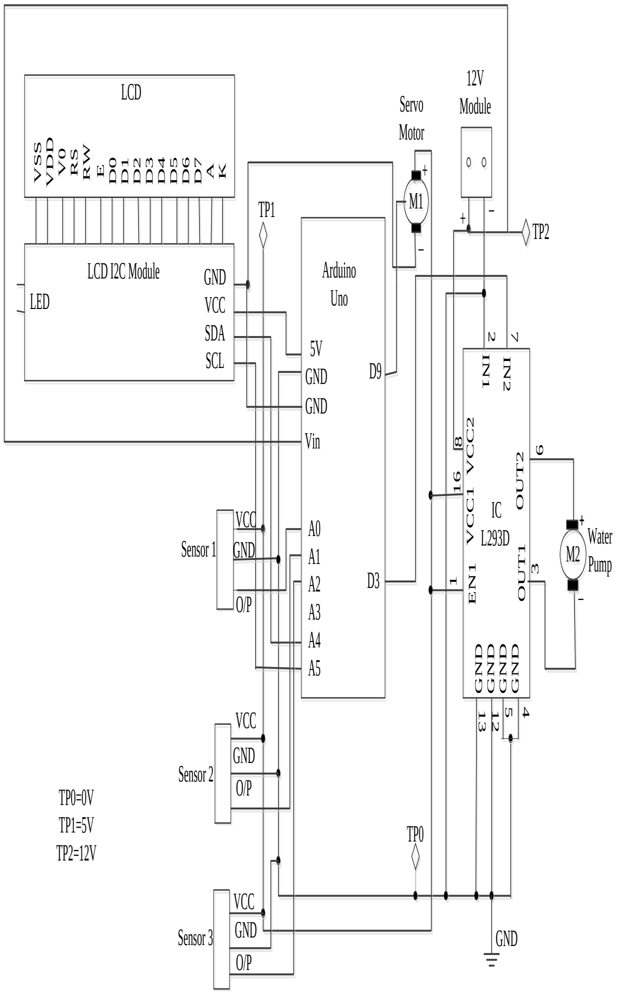


Fig 9: Circuit Diagram of Automatic Plant Watering System

4.RESULTS

The soil moisture sensor continuously monitors the volumetric moisture content in the soil and sends that information to the Arduino with values lying in the range 0 to 1023. Then the sensor obtains the data from the soil and display on the LCD screen. If the soil moisture content goes above a certain point, which is decided by the threshold value given in the Arduino code, the servo motor and the DC motor to start the watering mechanism. As soon as the Arduino decides that the soil needs watering, it tells the servo motor how much to rotate and along with it the water holding container is tilted to let the water flow. The water pump is used as the DC motor for running to maintain a steady flow of water.

The program in the Arduino reads the moisture value from the sensor every 20 seconds. If the value reaches the threshold value, the program runs the following three things;

- It moves the servo motor horn, along with the water pipe fixed on it, toward potted plant, whose moisture level is greater than the predetermined and threshold level.
- It starts the water pump to supply water to the plant for fixed period of time and then stops the water pump.
- It brings back the servo motor horn to its initial position.

At first the moisture sensor senses the amount of moisture present in the soil and accordingly it sends its output to the microcontroller through the input and output pins provided in the Arduino board. Then the sensor obtains the data from the soil and display on the LCD screen. Place the flower pots where the pipe from the servo motor horn is easily reach them. When the moisture level dips above 600, servo horn rotates at an angle of 30°. That is, after servo motor horn moves 30° toward the first pot, the water pump is ON for four seconds and then stops automatically. Then, the servo returns to its original position.

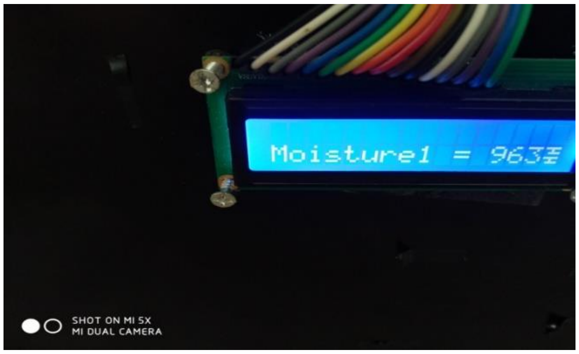


Fig 10: Display Moisture Value of Plant 1 on LCD

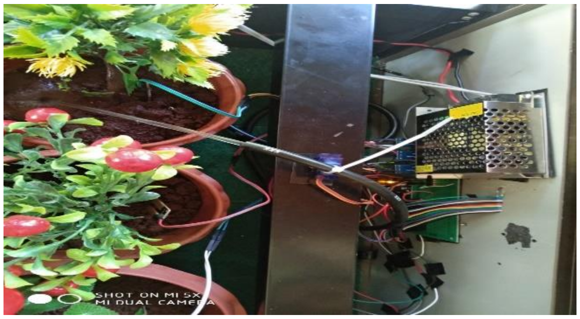


Fig 11: Result for Servo Motor of Plant 1 Rotates 30° Position

Similarly, if a second sensor is used, the moisture sensor senses the amounts of moisture presented in the soil and accordingly it sends its output to the microcontroller through the input and output pins provided in the Arduino board. Then the sensor obtains the data from the soil and display on the LCD screen. The servo motor horn rotates to 90° to the second pot, water pump is ON for five seconds and then stop automatically. The servo motor returns to its original position.

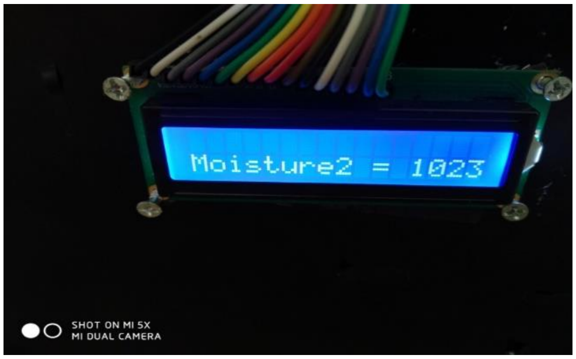


Fig 12: Display Moisture Value of Plant 2 on LCD

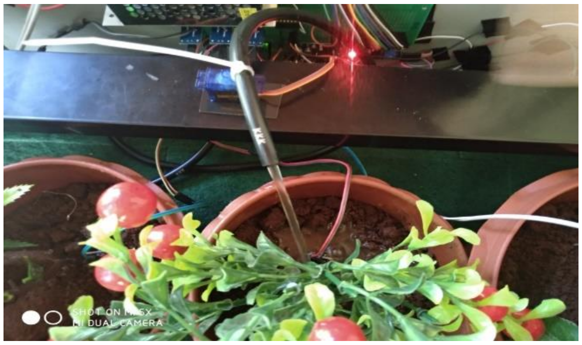


Fig 13: Result for Servo Motor of Plant 2 Rotates 90° Position

Finally, a third sensor is used, the moisture sensor senses the amount of moisture present in the soil and accordingly it sends its output to the microcontroller through the input and output pins provided in the Arduino board. Then the sensor obtains the data from the soil and display on the LCD screen. The servo motor horn rotates to 150° to the third pot, water pump is ON six seconds and then stop automatically. The servo motor returns to its original position.

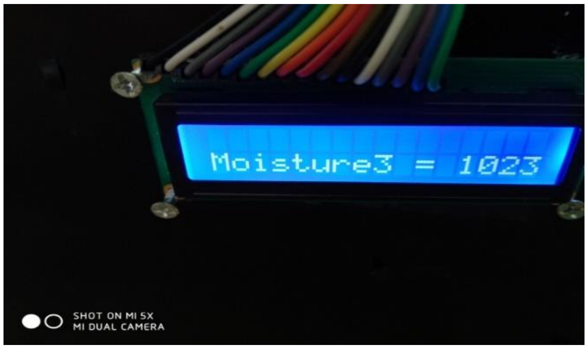


Fig 14: Display Moisture Value of Plant 3 on LCD

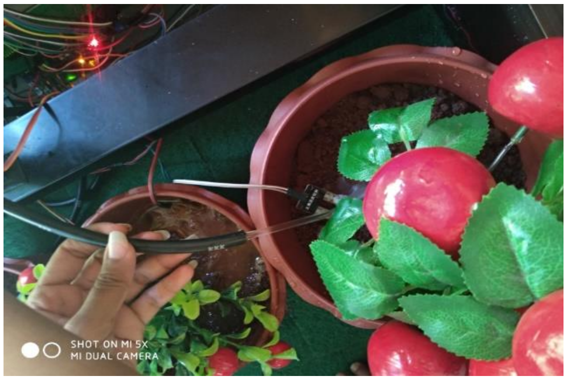


Fig 15: Result for Servo Motor of Plant 3 Rotates 150° Position

If the sensor value below the threshold value 600, nothing display on LCD screen. And servo motor horn its initial position at 10° has not changed any other degrees. And also, the motor driver L293D is not working, so the water pump is not pumped for plants. This situation is generally same as the operation of sensor 1, 2 and 3.

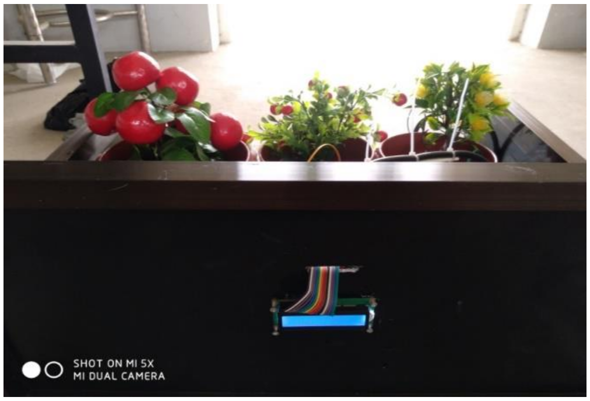


Fig 16: Result for Normal Situation

5.DISCUSSION

In this paper, the rotation of the servo motor is difficult to arrival the certain plant. Function of each module has been reasoned out, so the modules are placed carefully to the best working of the unit. The system has been tested to function automatically. The moisture sensor measures the moisture level of the different plant. Do not immerse the sensors fully inside the soil. If the moisture level is found to be above the desired level, the moisture sensor sends the signal to the Arduino board and LCD screen which triggers to rotate the servo motor and water pump to turn on and supply the water to the plants. When the desired moisture level is reached, the servo motor and water pump stop automatically. Thus, the functionality of the entire system is complete. In the present day, on conservation of water is of high importance. Present work is attempts to save the natural resources available which are used in very easily by human effort. By continuously monitoring the status of the soil, the flow of water is controlled and reduced the wastage. By controlling the pump in automatic plant watering system, the land is irrigated in all climatic conditions.

6.CONCLUSION

In this paper, automatic plant watering system by using Arduino was successfully established and tested. As the automatic plant watering system has a user friendly design so it is used by the household users as well as the commercial user. The best benefit is that it reduces the wastage of water during irrigation and helps in saving the precious water resource. Another are saving time, optimal water supply to plant or crop, automatic operation and protection against adverse weather conditions. The advantages of this paper are numerous. If this technology is implemented in the real world, a lot of water is saved. The world uses about 70% of freshwater for irrigation and on the other hand irrigation multiplies yield of most crops by almost 2 to 5 times. So irrigation seems to be a necessary evil. The best solution out of this paper is controlled usage of water for irrigation purpose.

REFERENCES

[1] Anonymous: “Arduino UNO tutorial begineers”. https://www.makerspaces.com/

[2] Anonymous: “Water Level Sensor”, (2018). http://osoyoo.com/2017/09/27/arduino-lesson-water-sensor/#2.2

[3] Jakob, M.: “lcd display 1602 i2c module”, January, (2017). https://wackydeveloper.wordpress.com/

[4] Saran A.: “Automatic Plant Watering and Soil Moisture Sensing”, (2017).

[5] Venkata, N. et.al.: “Developed automatic irrigation system used for plants”, (2017).

[6] Arelona, P. and Riya, R.: “Design and Implementation of automated plant watering system” Vol 4, issue 1, January, (2016).

[7] Ayan P.: “Automatic Plant Watering System”, October, (2016). https://electronicsforu.com/electronics-projects/hardware-diy/

[8] Rakesh R.: “L293D Motor Driver”, (2013). https://www.rakeshmondal.info/

[9] [9] Vellidis, Tucker, Kvien, Perry and Bednarz: “A Real Time Wireless Smart Sensor Array for Scheduling Irrigation”, National Environmentally Sound Production Agriculture Laboratory (NESPAL), (2007).

[10] [10] Castellano and Joseph, A.: Liquid Gold: “The Story of Liquid Crystal Displays and the Creation of an Industry World Scientific Publishing”, (2005).