

Automated Plant Irrigation System

ARDUINO BASED AUTOMATED PLANT Irrigation SYSTEM

System Fundamentals | 17/01/2018



# CONTENT

**1) ABSTRACT**

**2) OVERVIEW OF THE SYSTEM**

**3) MATERIALS NEEDED**

**4) LIMITATIONS OF THE EXISTING SYSTEM**

**5) ADVANTAGES OF THE PROPOSE SYSTEM**

**6) HARDWARE BLOCK DIAGRAM**

**7) MILESTONES**

**8) PROPOSED BUDGET**

**9) IMPLEMENTED ARDUINO CODE**

**10) FUTURE SCOPE OF SYSTEM**

**11) CONCLUSION**

**12) RESOURCES**

**13) PROJECT LOG**

# ABSTRACT

# Irrigation is the key to a successful garden. Long gone are the days of manual watering or relying on a friend to water when you are on vacation or away on business. The Project presented here waters your plants regularly when you are out for vacation. The circuit comprises sensor parts built using Arduino micro controller. Arduino micro controller is configured here as a comparator. Soil moisture sensor pins are inserted in the soil to sense the whether the Soil is wet or dry. The comparator monitors the sensors and when sensors sense the dry condition then the project will switch on the motor and it will switch off the motor when the sensors are in wet. The comparator does the above job it receives the signals from the sensors.

This prototype aims at saving time and avoiding problems like constant vigilance. It also helps in water conservation by automatically providing water to the plants/gardens depending on their water requirements. It can also prove to be efficient in Agricultural fields, Lawns & Parks. As technology is advancing, there is always a chance of reducing risks and making work simpler. Embedded and micro controller systems provide solutions for many problems. This application precisely controls water system for gardens by using a sensor micro controller system. It is achieved by installing sensor in the field to monitor the soil moisture which transmits the data to the microcontroller for estimation of water demands of plants.

# OVERVIEW OF THE PLANT IRRIGATION SYSTEM

This prototype monitors the amount of soil moisture. A predefined range of soil moisture is set, and can be varied with soil type or crop type. In case the moisture of the soil deviates from the specified range, the watering system is turned on/off. In case of dry soil, it will activate the irrigation system, pumping water for watering the plants. It consists of a microcontroller which is the brain of the system. The moisture sensor is connected to the input pins of the controller. The water pump is connecting with the output pins. If the sensors depart from the predefined range, the controller turns on the pump. Which ensures equal distribution of water to the soil. An LCD display indicates the moisture level like a percentage and indicates the status like a Dry, Moist and Soggy. An LED indicator indicates the status of the pump.

**Soil Moisture Sensor**

**Sensor Value<threshold value**

False True

**Stop Pump ()**

**Start Pump ()**

# MATERIALS NEEDED

## HARDWARE:

* Soil Moisture Sensor – A soil moisture sensor measures how much moisture is in soil, based on how much electricity the soil conducts. It pins will be put into soil. Our system’s key component.
* Arduino Uno Board
* DC Motor
* Relay Module
* LCD Display
* Bread Board
* Jumper Wires
* LEDs, Resistors

### SOFTWARE:

* Arduino IDE

# LIMITATIONS OF THE EXISTING SYSTEM

### **Cost:**

There are costs in purchasing, installing and maintaining automatic equipment.

### **Reliability:**

Can the irrigator trust an automatic system to work correctly every time? Sometimes failures will occur. Often these failures are because of human error in

setting and maintaining the systems. A re-use system is good insurance to collect any excess runoff when failures occur.

### **Increased channel maintenance:**

There is a need to increase maintenance of channels and equipment to ensure the system works correctly. Channels should be fenced to protect the automatic units from stock damage.

# ADVANTAGES OF THE PROPOSE SYSTEM

**Reduced employee and cost:**  
  
As the irrigator is not required to constantly monitor to check the progress of an irrigation, the irrigator is available to perform other tasks – uninterrupted. Because of that, the farmer is able to be away from the property, relax with the family and sleep through the night. In addition to, the reduces for the running costs of vehicles are used to constantly check progress down the bays being irrigated.  
  
**More timely irrigation:** Irrigators with automation systems are more inclined to irrigate when the plants need water, not when it suits the farmers. This the difference between hand watering and automatic system, it can take substantial time and early morning and evening watering rituals take away from family and effort. Automatic plant irrigation systems have timers that can be preset for daily or weekly watering so the farmers do not need to check the watering because the timer shuts the water off when it has finished.  
  
**Helping in the management of higher flow rates:** Many farmers are looking to increase the irrigation flow rates they receive through installing bigger channels and bay outlets. Such flow rates generally require an increase in employee as the time taken to irrigate a bay is reduced thus requiring more frequent change over. Automatic plant system irrigation allows for these higher flows to be managed without an increase in the amount of employees.  
  
**More accurate cut-off:** Automatic plant irrigation system allows cut-off of water at the appropriate point in the bay. This is usually more accurate than manual checking because human error can occur if the operator is too late or too early in making a change of water flow.  
  
**Reduced runoff of water and nutrients:** Automation can help keep fertilizer on farm by effectively reducing run off from the property. Using automatic plant irrigation system produces smaller droplets, helping to preserve nutrients and reducing runoff of water. Retaining fertilizer on farm has both economic and environmental benefits.

# HARDWARE BLOCK DIAGRAM

Probes to soil

LCD Display

Soil Moisture Sensor

Relay Module

Arduino Uno Board

(micro controller)

Sensor Amplifier

DC Motor

Power Supply

# MILESTONES

|  |  |
| --- | --- |
| **TASK** | **TIME** |
| Coming up with the project idea | 1 Week |
| Getting familiar with selected environment(Arduino) | 2 Week |
| Order required equipment | 2 Week |
| Test basics of the field | 1 Week |
| Get the sensor done(Soil moisture sensor) | 1 Week |
| Get the LCD Display done | 1 Week |
| Get the Relay Module & DC Motor done | 1 Week |
| Create the suitable prototype | 2 week |
| Testing and presentation | 1 week |
|  |  |
|  |  |

# PROPOSED BUDGET

|  |  |
| --- | --- |
| **EUIPMENT** | **PRICE** |
| Arduino Uno Board | 900.00LKR |
| Soil Moisture Sensor | 180.00LKR |
| DC Motor | 650.00LKR |
| Relay Module | 250.00LKR |
| LCD Display | 300.00LKR |
| Jumper Wires | 100.00LKR |
| Bread Board | 200.00LKR |
| 12V Power Adapter | 650.00LKR |
| LEDs 10mm (Red & Green) | 5.00LKR |
| Resistors 150ohm ¼ w | 5.00LKR |
|  |  |

Total – 3240.00LKR

# IMPLEMENTED ARDUINO CODE

*#include <Wire.h>*

*#include <LCD.h>*

*#include <LiquidCrystal\_I2C.h>*

*#define I2C\_ADDR 0x27*

*#define BACKLIGHT\_PIN 3*

*LiquidCrystal\_I2C lcd(0x27,2,1,0,4,5,6,7,3,POSITIVE);*

*int sensorPin = A0; //soil moisture sensor input pin*

*int sensorValue = 0; //variable to store the value coming from the sensor(sensor value)*

*int percent = 0; //convert to percent*

*int greenLED = 6;*

*int redLED = 7;*

*int motorpin=8;*

*int thresholdValue = 800; // you can adjust the threshold value*

*void setup() {*

*pinMode(motorpin,OUTPUT);*

*digitalWrite(motorpin,LOW);*

*pinMode(sensorPin, INPUT);*

*pinMode(greenLED, OUTPUT);*

*pinMode(redLED, OUTPUT);*

*digitalWrite(greenLED, LOW);*

*digitalWrite(redLED, LOW);*

*lcd.setBacklightPin(BACKLIGHT\_PIN,POSITIVE);*

*lcd.setBacklight(HIGH);*

*lcd.begin(16, 2);*

*lcd.setCursor(6,0);*

*lcd.print("Plant");*

*lcd.setCursor(0, 1);*

*lcd.print("Watering System.");*

*delay(4000);*

*lcd.clear();*

*lcd.begin(16, 2);*

*lcd.setCursor(0, 0);*

*lcd.println("Moisture : ");*

*Serial.begin(9600); //debug console*

*}*

*void loop() {*

*int sensorValue = analogRead(sensorPin);*

*lcd.setCursor(11, 0);*

*percent = convertToPercent(sensorValue);*

*lcd.print(percent);*

*lcd.setCursor(13, 0);*

*lcd.print("%");*

*lcd.setCursor(0, 1);*

*lcd.print("Status : ");*

*if(sensorValue>thresholdValue){*

*Serial.print("Sensor Value: ");*

*Serial.print(sensorValue);*

*Serial.print(" Percent: ");*

*Serial.print(percent);*

*Serial.print("%");*

*Serial.println(" - Time to water your plant");*

*lcd.println("Dry ");*

*digitalWrite(motorpin,HIGH);*

*digitalWrite(redLED, HIGH);*

*digitalWrite(greenLED, LOW);*

*}*

*if( sensorValue<thresholdValue && sensorValue>500){*

*Serial.print("Sensor Value: ");*

*Serial.print(sensorValue);*

*Serial.print(" Percent: ");*

*Serial.print(percent);*

*Serial.print("%");*

*Serial.println(" - Doesn't need watering");*

*lcd.println("Moist ");*

*digitalWrite(motorpin,LOW);*

*digitalWrite(redLED, LOW);*

*digitalWrite(greenLED, HIGH);*

*}*

*if(sensorValue<thresholdValue){*

*Serial.print("Sensor Value: ");*

*Serial.print(sensorValue);*

*Serial.print(" Percent: ");*

*Serial.print(percent);*

*Serial.print("%");*

*Serial.println(" - Doesn't need watering");*

*lcd.println("Soggy ");*

*digitalWrite(motorpin,LOW);*

*digitalWrite(redLED, LOW);*

*digitalWrite(greenLED, HIGH);*

*}*

*delay(500);}*

*//method of how to give sensor value like a percentage*

*int convertToPercent(int value)*

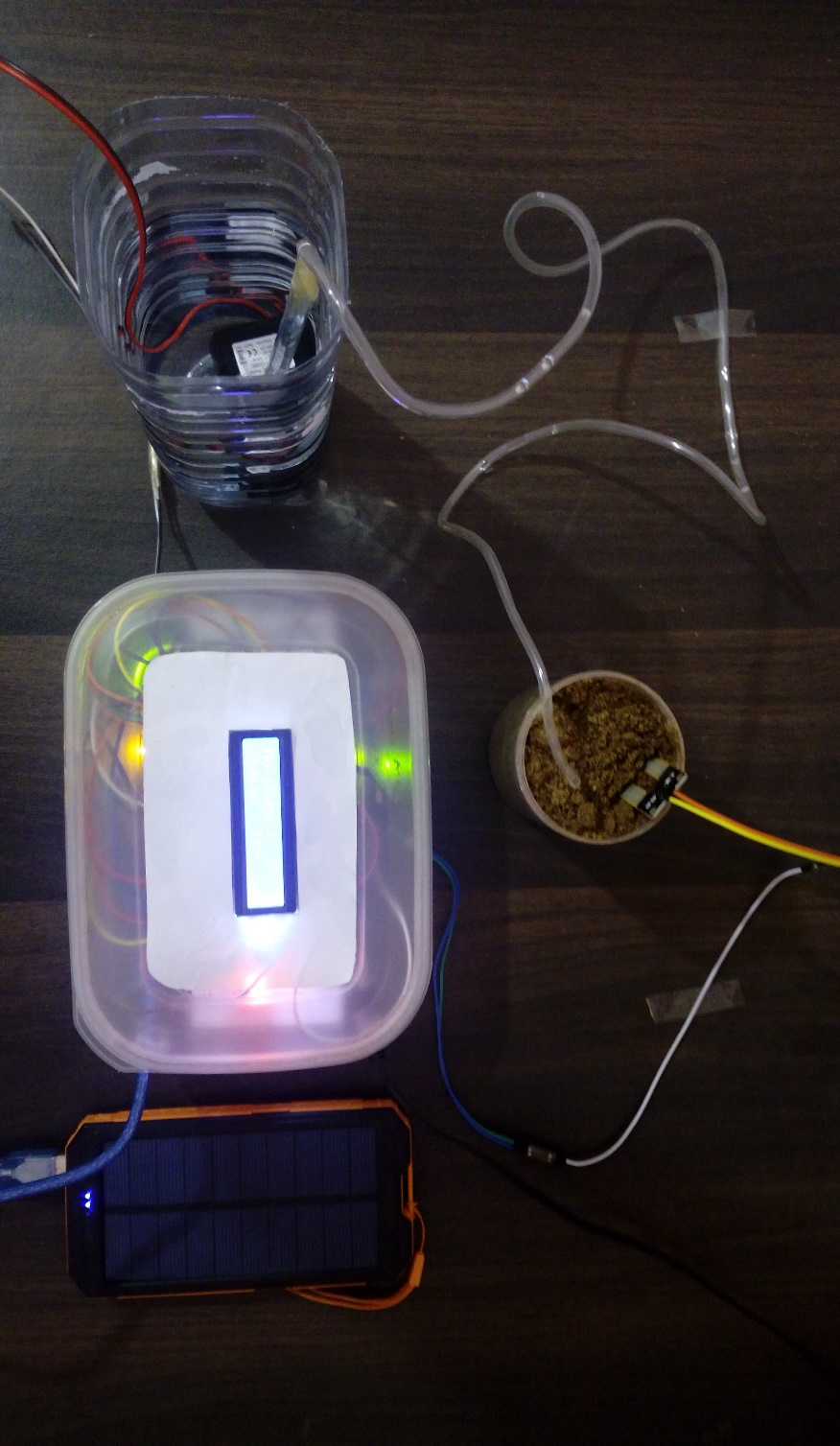
*{*

*int percentValue = 0;*

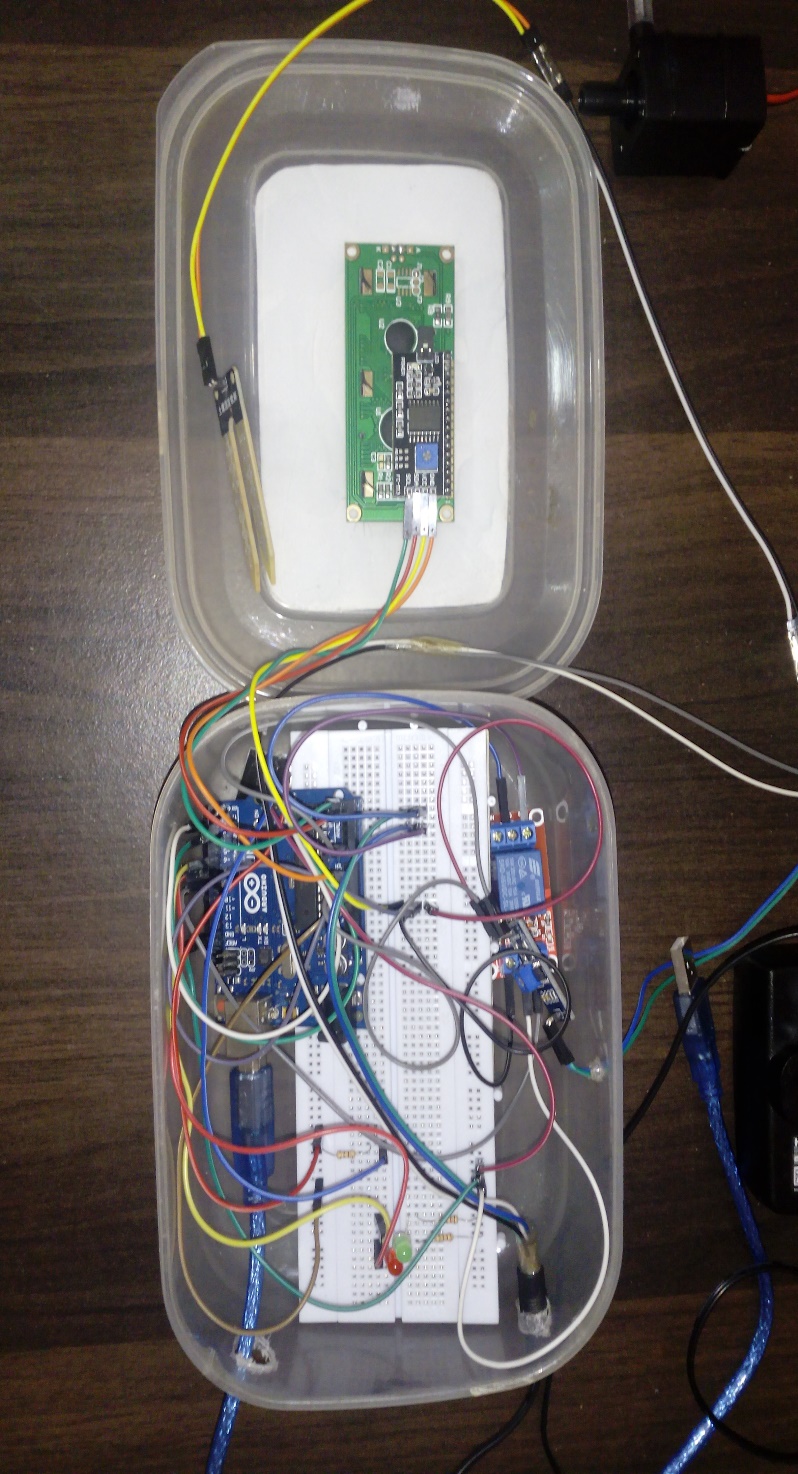
*percentValue = map(value, 1022, 403, 0, 100);*

*return percentValue;*

*}*



* Automated Plant Irrigation System





# FUTURE SCOPE OF SYSTEM

* Our project can be improvised by adding a Webs caper which can predict the weather and water the plants/crops accordingly. If rain is forecasted, less water is let out for the plants.
* Also, a GSM module can be included so that the user can control the system via smart phone.
* A water meter can be installed to estimate the amount of water used for irrigation and thus giving a cost estimation.
* A solenoid valve can be used for varying the volume of water flow. Furthermore, Wireless sensors can also be used.

# CONCLUSION

In present days especially farmers are facing major problems in watering their agriculture fields, it’s because they have no proper idea about when the power is available so that they can pump water. Even after then they need to wait until the field is properly watered, which makes them to stop doing other activities. Here is an idea which helps not only farmers even for watering the gardens also, which senses the soil moisture and switches the pump automatically when the power is ON.

# RESOURCES

1. Official Arduino site - <https://www.arduino.cc/>
2. You tube
3. <http://www.instructables.com/id/Arduino-Automatic-Watering-System-For-Plants/>
4. <https://greensense.github.io/Blog/2017/02/17/Arduino-Soil-Moisture-Sensor-Calibration/>

# PROJECT LOG

**01/10/2017 – Starting the project**

We talked about few topics and decided to build an Automated Irrigation system using Arduino Uno.

**04/10/2017 – Connecting Arduino Uno**

First, we installed the Arduino IDE and we connected Arduino Uno board and wrote a simple programs. Because we weren’t familiar with Arduino programming.

**17/10/2017 – Connecting Soil Moisture sensor with Arduino board**

We connected the soil moisture sensor to the Arduino board and we programmed it to get moisture level of soil.

**20/10/2017 – Calibrate the Soil Moisture Sensor**

We have to calibrate the soil moisture sensor because soil moisture measurements are not accurate and we understood the importance of soil moisture sensor calibration and done it.

**30/10/2017 – Connecting LCD Display with Arduino board**

We connect the LCD Display to the Arduino board using jumper wires. We have to use 16 jumper wires for the connect. So it is difficult and feeling trapped. Also we had to adjust the contrast of the LCD Display. To adjust the contrast we need potentiometer.

**05/11/2017 – Get the I2C Adapter for LCD Display**

So we have to buy I2C adapter to solve the above 2 problems. From this I2C adapter we can connect lcd display to the Arduino board using only 4 jumper wires and we can adjust the contrast for better vision.

**10/11/2017 –Program the LCD Display**

We connected the lcd display with I2C adapter with the Arduino board and we programmed it, but it didn’t work.

**17/11/2017 –Program the LCD Display**

We uploaded the code through Arduino IDE, but the code didn’t upload. We used different methods to fix this error, but couldn’t fix the upload error. After the web searching we found out that we didn’t add the ‘Newliquidcrystal’ libraries to the Arduino IDE. Then we searched about the libraries and added it to the IDE.

**22/11/2017 –Connect the DC Motor using Motor Controller**

First, we try to connect the dc motor using motor controller. But we understood we can do it using relay easily.

**27/11/2017 –Connect the DC Motor using Relay Module**

Then, we try to connect the dc motor using relay. It was possible to connect the relay with dc motor to the Arduino board. But we wanted to do it so easily. So we use relay module instead relay.

**04/12/2017 –Programed the Relay Module**

It is easy to the program the relay module and it work like switch we want.

**06/12/2017 –Connect the DC Motor to the Relay Module**

We done the connect the dc motor to the relay module. But motor doesn’t work better. So we checked our power supply.it gave output only 9v. To better work, dc motor wants 12v output power supply. So we brought 12v power supply and solve the problem.

**16/12/2017 –Code the Final Program and Fixed errors**

Code the full program and compile it to check some syntax errors. We get some syntax errors and resolve all syntax errors. Then upload the final code and we can make our system better.