Soil Humidity Level using Humidity Sensor for Smart Watering Mechanism (ACM:IA-I-RRL-PM-R-A)

Margarita Shiela Brinosa Timbal (Student)

Student of Department of Computer Engineering, College of Engineering, Architecture and Industrial Design, Bohol Island State University-Main Campus, [margaritashiela.timbal@bisu.edu.ph](mailto:margaritashiela.timbal@bisu.edu.ph)

MAX ANGELO DAPITILLA PERIN (INSTRUCTOR)

Bachelor of Science Computer Engineering, College of Engineering, Architecture and Industrial Design, Bohol Island State University-Main Campus, [maxangelo.perin@bisu.edu.ph](mailto:maxangelo.perin@bisu.edu.ph)

**Imaginative Abstract:**Home gardening had rapidly increased during the imposed lockdowns to suppress the spread of Covid-19. But nowadays, the restrictions of the lockdown are slowly being released and people are now going back therefore resulting into lack of time in nurturing their plants. We thought about this problem and find a solution to continue gardening and at the same time their jobs. With soil humidity level using humidity sensor, gardeners can monitor the soil moisture of a plant specifically Sweet Honey Plant since this type of plant requires delicate nurturing. After it senses that the soil of the plant is dry, it will send signals to let them know that the plant needs water. Therefore, the goal of this is to be able to create a system that informs the user if the plant needs watering based on its humidity level. Users can also make a watering schedule so that the prototype will automatically water the plant every day.

CCS CONCEPTS • mechanism • sensor• microcontroller • automated system

Additional Keywords and Phrases: humidity sensor, watering mechanisms, humidity level

ACM Reference Format:

**Margarita Shiela Brinosa Timbal**, **MAX ANGELO DAPITILLA PERIN.** 2022. **Soil Humidity Level using Humidity Sensor for Smart Watering Mechanism (ACM:IA-I-RRL-PM-R-A).** In Research Project Presentation for Bachelor of Science in Computer Engineering 3 in CPE 323 – Microprocessor Systems S.Y. 2021-2022, 2nd Semester, Bohol Island State University-Main Campus, Tagbilaran City, Republic of the Philippines. ACM, New York, NY, USA

1 INTRODUCTION

We are all aware that watering plants is one of the most important aspects when it comes to nurturing it. Plants give us oxygen which is very much essential to live that’s why it’s considered as exceptionally helpful to every person in every angles. But because also of the pandemic, there was an increase of individuals who have been into plants. However, as the restriction of the pandemic is slowly declining, people are also starting to get busy with their work which makes less time for them to look after their plants therefore resulting it to withered. Plants are heavily dependent with water thus; a smart watering mechanism would be convenient. Though the requirements of water of every plant depends on the soil properties such as soil moisture. An automated system like this would allow a control of timely watering activity to support the plant growth processes in plant. Therefore, resulting into the replacement of the old technique when it comes to watering your delicate plants. Also, the soil moisture sensor would be a promising solution for the people who want to take care of their plants without the need of their presence. And to create this automated system, Arduino microcontroller would be used.

Arduino is one of the educational product microcontrollers thus it meets the criteria as the control system of this smart watering mechanism. It is also a flexible programmable hardware platform and designed to control the circuit logically. In this system, a soil moisture sensor is used to detect and check the soil humidity of the plant. Based on the soil moisture level from the plant, the system will let it to automatically water the plant when it’s not damp enough and turn it off after.

2 REVIEW OF RELATED LITERATURE

Soil provides plants with a foothold for their roots as well as the nutrients they require to grow. It filters rainwater and regulates excess rainwater discharge, preventing flooding. It can store large amount or organic carbon and also acts as a buffer against pollutants, protecting groundwater quality.

An Arduino Based Automatic Plant Watering System is proposed in (Devika, 2014) where authors developed the Arduino microcontroller used to control two functional components which are the moisture sensors and the pump to automatically water the plant. The moisture sensor’s function is to sense the level of moisture in the soil whereas the water pump supplies water to the plants.

In a smart drip irrigation system using Raspberry Pi and Arduino is proposed for home automation system. A drip irrigation system makes the efficient use of water where the water is slowly dripped to the roots of the plant through narrow tubes and valves. The water flow from the system can be remotely controlled via email.

In this author (Laxmikant Jayprakash Goud) proposed “A Review and Proposed Automated Irrigation System using Soil Moisture Sensor and Android App”. This application will work with the help of hardware. It requires sensors, controller and GSM kit. First, the sensor will be place in the soil to detect its more. This produces a voltage level in terms of output. As machine will only understand the language of voltage, the voltage will be acquired by the controller and will then produce output. In this way, sensor and controller will work together to produce moisture level.

The small sphere of influence on the tested soil moisture parameters, combined with the high evaporative demands and temperatures used in this experiment, resulted in non-uniform drying of the tested containers (Raper, 2015). The study indicates that texture response for the 10HS sensors was not substantial and neither the 10HS nor 200SS was highly influenced by the drastic variations in soil temperature.

According to the study entitled “High-Sensitivity and Compact Time-domain Soil Moisture Sensor Using Dispersive Phase Shifter for Complex Permittivity Measurement”, soil moisture is a significant contributor to environmental issues such as climate change and natural disasters because of its impact on drainage, infiltration and fertilization, soil moisture is critical in precision agriculture. Agriculture sensors are planted at various depths in the ground to measure and transmit soil properties to a central location. Wireless Sensor Networks (WSN) and the Internet of Things (IoT) provide an optimal solution in this regard by rapidly recording necessary information

3 PROPOSED METHODOLOGY

The study will be conducted using an experimental technique by the researchers. This automated system created by the researchers will replace the traditional way of watering the plants. The accuracy of this watering mechanism is determined using an experimental approach. The data collected from the Soil Moisture Sensor is sent to the system to be able to utilize the efficient operation.

3.1 Hardware Overview

**ARDUINO UNO:** Arduino UNO is a microcontroller on the ATmega328. It has digital input/output pins, 6 analog inputs, a 16MHz ceramic resonator, a USB connection a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller, simply connect it to a computer with a USB cable or power it with a AC-DC adapter or battery to get started.

**SOIL HUMIDITY SENSOR:** The soil moisture sensor utilized has its voltage output proportional to the quantity of water in the soil. Its specified supply voltage is from 3.3V to 5V and with this supply, it gives an output voltage of between 0V to 2.3V for the full range of complete dryness to submersion in water. Its rating for maximum operating current is 0.15A. Its output is fed into the analog to digital converter input of the microcontroller.

**THE CONTROL UNIT:** This unit is basically the section that provides the control of the whole system. It consists of a microcontroller IC chip plus peripheral components and the control logic which the chip functions with. The microcontroller chip is the central hardware component while the program or code is the firmware component.

**THE DISPLAY UNIT:** The display unit is simply an output unit used for the purpose of giving the user the required information. The display unit is a simple 4x4 LCD module.

Table 1: Components used in this study

|  |  |
| --- | --- |
| Arduino UNO | Solenoid Actuator Valve |
| Soil Moisture Sensor | Jumper Cables |
| 8x2 LCD module | Aquarium Hose |
| Breadboard | Relay Module |
| Power Supply | Arduino Case |

**3.2 Components Used**

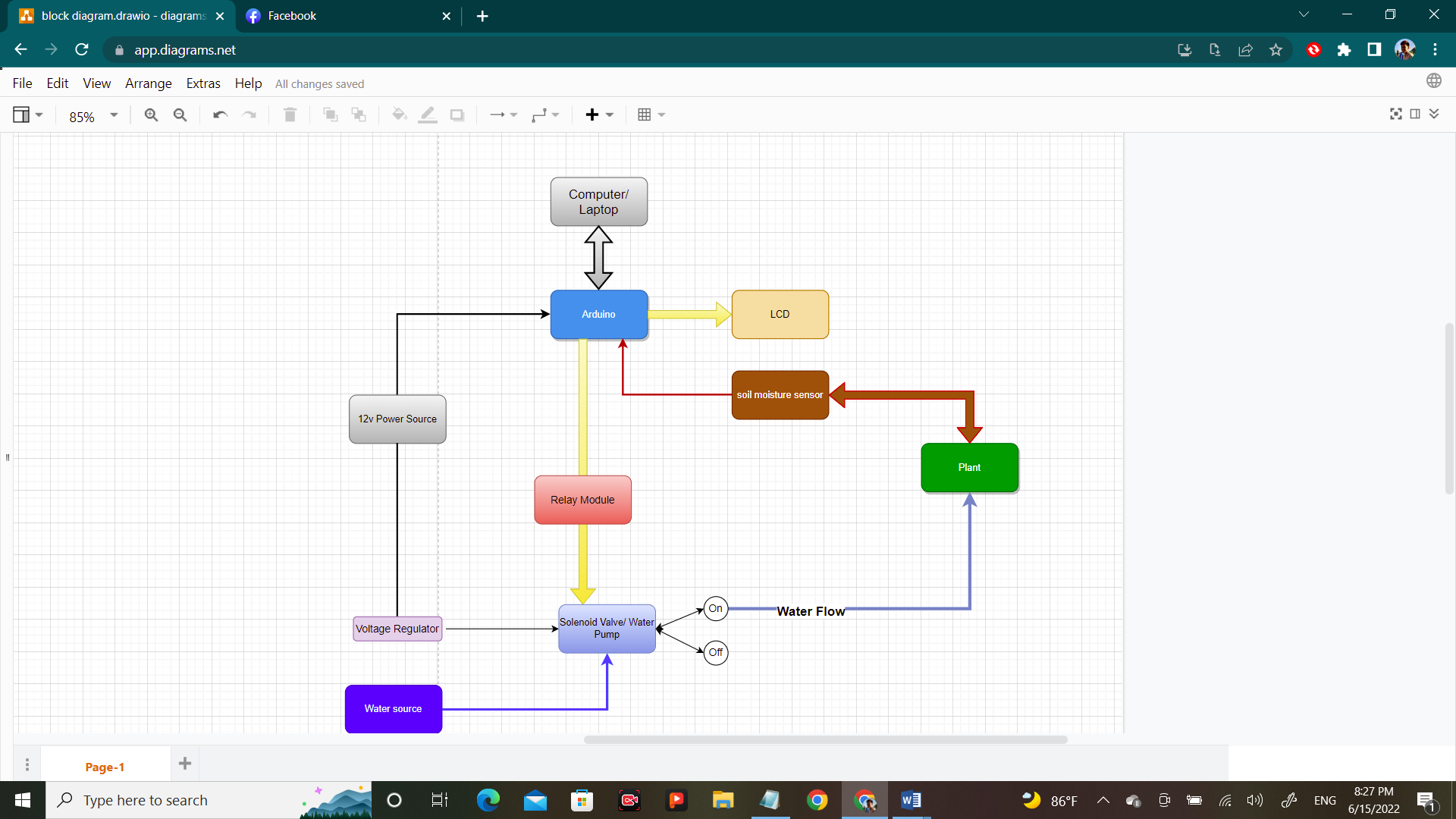
 In Table 1, you can see the different materials and components that the researchers used in this study. One of those is the Arduino Uno microcontroller. It also has the Soil Moisture Sensor that detects the level of the soil humidity. We also used jumper wires, power supply to be able to make this study function.

Figure 1: Block Diagram of Smart Plant Watering Mechanism using Arduino

**3.3 Block Diagram**

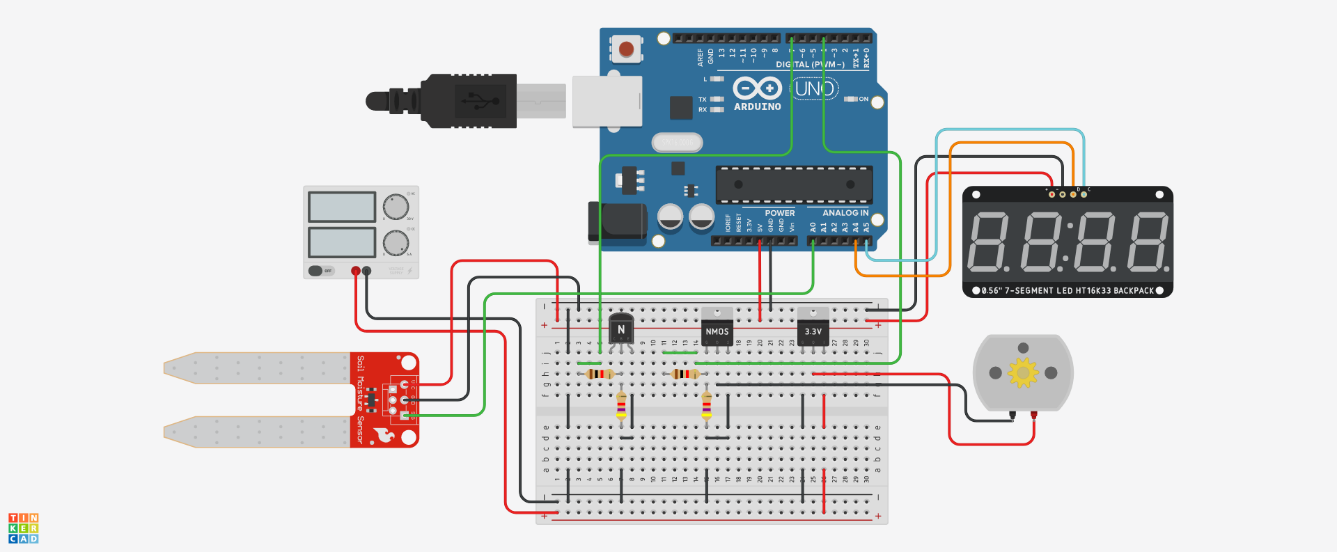
The block diagram shown in the Figure 1 is the system consist of power source, soil moisture sensor, Arduino, solenoid valve and LCD to be able to show the information about the humidity level of the soil.

Figure 2: Circuit Diagram for Smart Watering Mechanism using Arduino

**3.4 Circuit Diagram**

The circuit diagram shown in Figure 2 consists of Arduino Uno, jumper wires and Soil Moisture sensor. It is used to detect the soil humidity level of the plant through its soil.

3.5 Applications Used

The application used in this study is the Arduino IDE. It is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X and Linux. The IDE environment mainly consists two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module.

**REFERENCES**

<bib id="bib1"><number>[1]</number> Hassan. (2018, December). *ResearchGate.* Retrieved from ResearchGate: researchgate.net/publication/330212779\_An\_Automated\_Irrigation\_System\_Using\_Arduino\_Microcontroller</bib>

<bib id="bib2"><number>[2]</number> [2] Keshavarz. (2021, December). High Sensitivity and Compact Time-domain Soil Moisture Sensor using Dispersive Phase Shifter for Complex Permittivity Measurement. Retrieved from Research Gate: https://www.researchgate.net/publication/356763198</bib>

<bib id="bib3"><number>[3]</number> [3] Singhal, A. &. (2015). *Smart Drip Irrigation System using Raspberry Pi and Arduino.* Retrieved from Smart Drip Irrigation System using Raspberry Pi and Arduino: http://doi.org/10.1109/CCAA.2015.7148526</bib>

<bib id="bib4"><number>[4]</number> [4] *Why are Soils Important?* (2022). Retrieved from ISRIC - World Soil Information: https://www.isric.org/discover/about-soils/why-are-soils-importan **</bib>**

APPENDICES

Image of the Prototype

Internal Components.

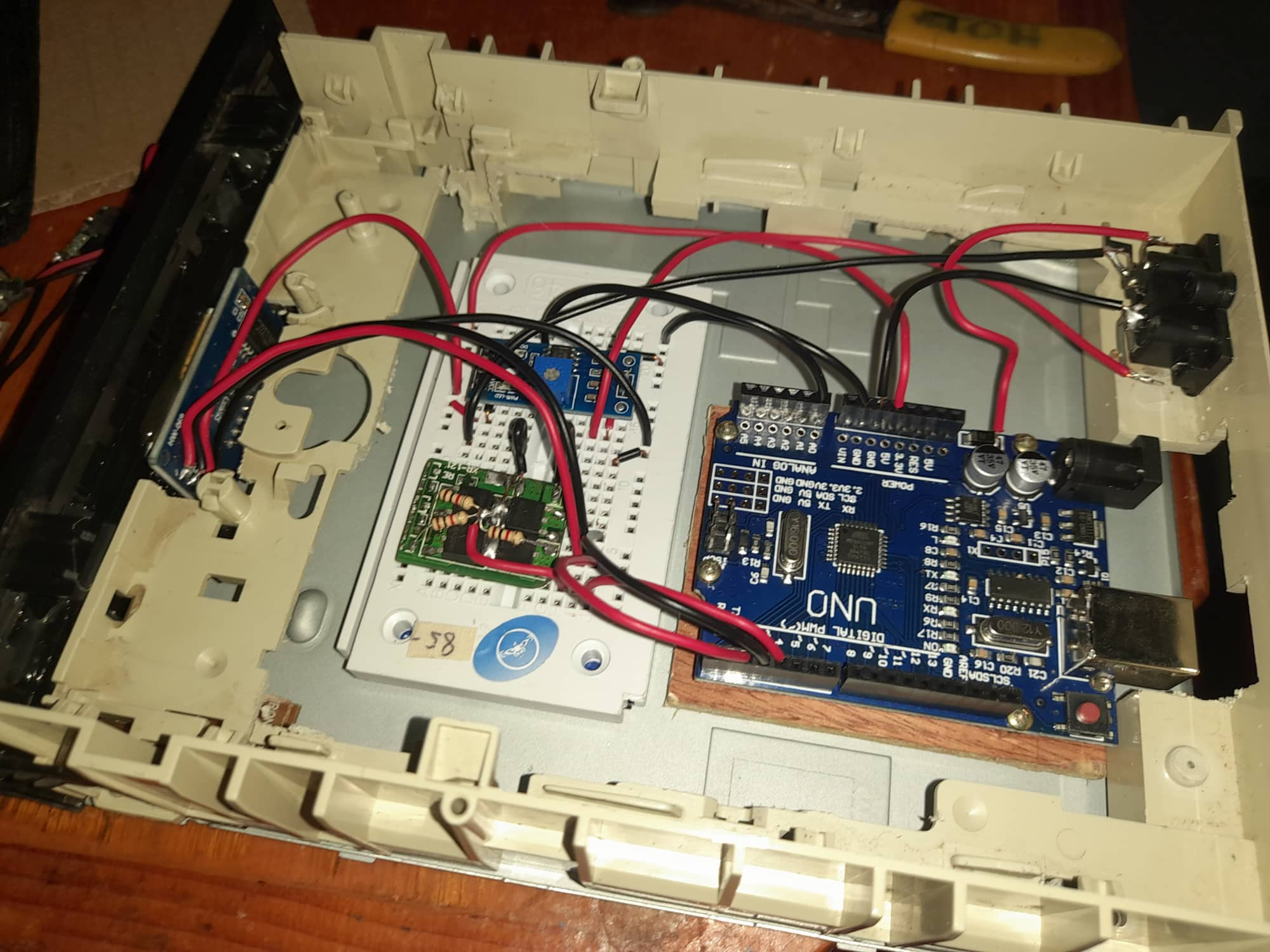


Figure 3: Image of the Smart Plant Watering Mechanism using Arduino

** Final Prototype.**

Figure 4: Image of the Final Prototype

**Soil Moisture Sensor using Humidity Sensor (Arduino) Sketch**

//Soil Moisture Sensor

#define soil Moisture A0 //connect analog pin of soil moisture sensor to A0 of Arduino

//calibration

//maximum value detected(dry) 1025

//minimum value detected(wet) 10

//percentage of moisture= 100-(detected/maximum\*100)

//100/maximum=100/1025=0.0975

//TM1637 Display

#include "TM1637.h"

#define CLK 5 //pins definitions for TM1637 and can be changed to other ports

#define DIO 6

TM1637 tm1637(CLK, DIO);

unsigned long moisture;

float percentage;

int pState = 0; //Indicates Pump/Motor State Off 0 On 1

float minM = 20;//Minimum moisture level

float maxM = 40;//Maximum moisture level

void setup() {

// put your setup code here, to run once:

// Initialize a serial connection for reporting values to the host

Serial.begin(115200);

pinMode(soilMoisture, INPUT);

pinMode(4, OUTPUT);

pinMode(7, OUTPUT);

tm1637.init();

tm1637.set(BRIGHT\_TYPICAL); //BRIGHT\_TYPICAL = 2,BRIGHT\_DARKEST = 0,BRIGHTEST = 7;

//Display Test

tm1637.display(3, 3);

tm1637.display(2, 2);

tm1637.display(1, 1);

tm1637.display(0, 0);

delay(3000);

tm1637.clearDisplay();

}

void loop() {

// put your main code here, to run repeatedly:

unsigned long tmp2 = 0;

int a;

int b;

int c;

int disP[] = {};

int dic;

digitalWrite(7, HIGH);//Turns on the soil moisture sensor

//Gets average soil moisture level

for (int i = 0; i <= 100; i++) {

unsigned long tmp1 = analogRead(soilMoisture); //read input from pin A0(soilMoisture)

tmp2 = tmp2 + tmp1;

delay(5);

}

moisture = tmp2 / 100;

displayNumber(moisture);

Serial.print("Moisture: ");

Serial.print(moisture);

Serial.print(" ");

percentage = 100 - (moisture \* 0.0975); //express as a percentage

Serial.print("Percentage: ");

Serial.print(percentage);

Serial.println("%");

displayNumber(percentage);

pump(percentage);

digitalWrite(7, LOW);//Turns of the moisture sensor to prevent probe damage

delay(3000);//Delay extend to every hour

}

//Display moisture level

void displayNumber(int num) {

tm1637.clearDisplay();

if (num == 100) {

tm1637.display(3, num % 10);

tm1637.display(2, num / 10 % 10);

tm1637.display(1, num / 100 % 10);

}

if (num < 100 && num > 10) {

tm1637.display(3, num % 10);

tm1637.display(2, num / 10 % 10);

}

if (num < 10 && num >= 1) {

tm1637.display(3, num % 10);

}

if (num < 1) {

tm1637.display(3, 0);

}

}