Princess Sumaya University for Technology

King Abdullah II Faculty of Engineering

Electrical Engineering Department



**Smart Irrigation System**

Supervisor: Dr. Belal H. Sababha

**Bushra Qutob 20190774**

**Meena Khaffaf 20190408**

**Mohammed Alzu’bi 20190323**

Contents

* [Abstract 3](#_Toc125388584)
* [Introduction 3](#_Toc125388585)
* [Project Description and Components 4](#_Toc125388586)
* [Flow Chart 15](#_Toc125388587)
* [Prototype 16](#_Toc125388588)
* [Problems and Recommendations 17](#_Toc125388589)
* [Conclusion 17](#_Toc125388590)

# Abstract

The main goal of this project is to use moisture and temperature sensors to monitor the moisture level of the soil in both dry and wet circumstances. The system determines whether water should be pumped based on data taken by the sensors. The system uses an ultrasonic sensor to determine the tank's water level and then switches an H-bridge to operate the water pump in accordance with the situation. It manages plant watering automatically, minimizing the need for human involvement. Additionally, it contributes to time savings, financial efficiency, environmental protection, cheap operation and maintenance costs, and effective irrigation services. The system also pays attention to the plant's life cycle and makes sure that all conditions are met for plant growth. It also helps to use water only and where needed which help in conservation of water.

# Introduction

Irrigation is the application of water to replenish the soil in plant root zones. According to data from AQUASTAT, FAO's Global Information System on Water and Agriculture, irrigation consumes 70% of the freshwater percentage on earth. Irrigation can have a serious negative impact on the environment if it is not properly managed. Due to obsolete and inefficient irrigation systems, the used water resources are not fully exploited. Poor irrigation is not the only problem that can arise from irrigation. Over-irrigation is also considered an issue. Today's world is experiencing a water shortage, making it urgent to implement clever irrigation techniques. This project, a smart irrigation system, was created to make farming easier for farmers who could also keep an eye on the moisture content of the water in the soil to make sure the plants were getting enough water to grow. By providing the required amount of water, this project aids the farmer in water optimization while reducing erosion. It has been noted that farmers occasionally are unable to control the plant's water consumption and invariably end up overwatering the plant. Additionally, many farmers are reported to lack the precise indicator needed to determine the rate of consumption for each plant. As a result, the farmer will be able to use this technique to consistently track the rate of plant consumption.

Our proposed design is implemented using temperature and moisture sensors to determines whether water should be sprinkled based on the conditions programmed using the microcontroller PIC16F877A. The system uses an ultrasonic sensor to determine the tank's water level and then switches an H-bridge to operate the water pump. If conditions for irrigation are meet and there’s enough water in the tank, the water pump will open and water will flow into the soil. If the water tank is empty, a buzzer will notify the farmer that the irrigation process failed and the water pump will close.

# 

# Project Description and Components

To implement our design, we will use the following components:

1. **PIC microcontroller (16F877A)**

One of the most well-known microcontrollers in the business which is easy to code or program. There are 33 input and output pins out of a total of 40 pins on it. It has an eight-channel, eight-bit ADC module which means 8 analog sensors are supported. The microcontroller has 3 timers which are timer0, timer1 and timer2. It is served as the brain of the circuit in this project, connecting and coordinating with all sensors. Mikro C pic compiler is used to program PIC16F877A.



Figure PIC16F877A

1. **Ultrasonic sensor HC-SR04**

An ultrasonic sensor is a piece of technology that uses ultrasonic sound waves to measure a target object's distance. The time elapsed between the emission and reception is measured by ultrasonic sensors to determine the target's distance. HC-SR04 has three parts: a transmitter, receiver, and control circuit. The HC-SR04 only has four pins that require: VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground). The sensor is easy to interface with a microcontroller using just two digital I/O pins. This affordable, simple sensor has accuracy that can reach up to 3mm. One of the main uses of HC-SR04 is to detect, monitor, and control liquid levels in closed containers.



Figure 2 HC-SR04 Ultrasonic Sensor

1. **Moisture sensor with the module**

Using a moisture sensor, you may find out how much water is in the soil or other things, like plants. It calculates the volumetric amount of water present outputs the moisture content. Moisture sensors typically work by measuring changes in electrical conductivity, resistance, or capacitance in response to the presence of moisture. The module contains a potentiometer to change the threshold level as well as digital and analog outputs. This moisture sensor has four pins: GND, DO, and AO. The analog pin is connected to the moisture sensor, while the digital out pin is connected to the LM393 comparator IC's output pin. The moisture sensor is made up of two probes to measure the moisture content of the soil. The sensor reads the resistance after passing the current through the soil using these two probes to determine the soil's moisture levels. To use the moisture sensor, connect the Analog/Digital output pin of the module to the Analog/Digital pin of Microcontroller and connect VCC and GND pins to 5V and GND pins of Microcontroller. After that insert the probe inside the soil. When there is more water presented in the soil, it will conduct more electricity that means resistance will be low and the moisture level will be high.

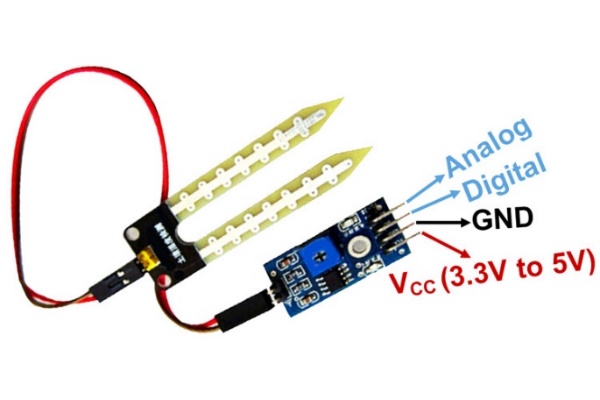


Figure 3 Moisture Sensor

1. **Temperature sensor (LM35)**

The LM35 is a highly accurate integrated-circuit temperature sensor that may be used in a variety of applications to measure temperature precisely. The sensor has an accuracy of 0.5°C and can monitor temperatures between -55°C and 150°C. LM35 has three pins: power (VCC), ground (GND), Output (ANALOG). LM35 Sensor operates in a very straightforward and simple-to-understand manner. To measure a known temperature value, the LM35 temperature sensor employs the fundamentals of a diode. The voltage across a diode grows at a known rate as the temperature rises, as we all know from semiconductor physics. It is simple to produce a voltage signal that is directly proportional to the ambient temperature by accurately amplifying the voltage change. LM35 gives the temperature as a voltage and then a conversion is done to find the value in Celsius. The sensor is a popular option in many temperature measurement systems since it is also reasonably priced.



Figure 4 LM35 Temperature Sensor

1. **Buzzer**

A buzzer is commonly used as an audible indicator or alarm. They can be used to signal the completion of a task, to alert the user of an error, or to provide other types of feedback. Buzzers are relatively simple to interface and control with a microcontroller and consume low power, making them an ideal choice for embedded systems where space and power are at a premium. The two wires of the buzzer are connected to the VCC and ground.



Figure 5 Buzzer

1. **LCD**

LCD (Liquid Crystal Display) is a type of display that is commonly used to display text and other information. It is composed of a matrix of dots, or pixels, that can be controlled to create different characters and symbols. The LCD used is called 2x16 which means it has 2 rows and 16 columns of characters. An LCD is made up of two layers of polarizing material with a liquid crystal solution between them. The liquid crystals can be manipulated to block or allow light to pass through, which creates the characters and symbols that are displayed on the screen. They are commonly used as a user interface in embedded systems, and also as an alternative to LEDs and 7-segment displays for displaying numbers and characters.

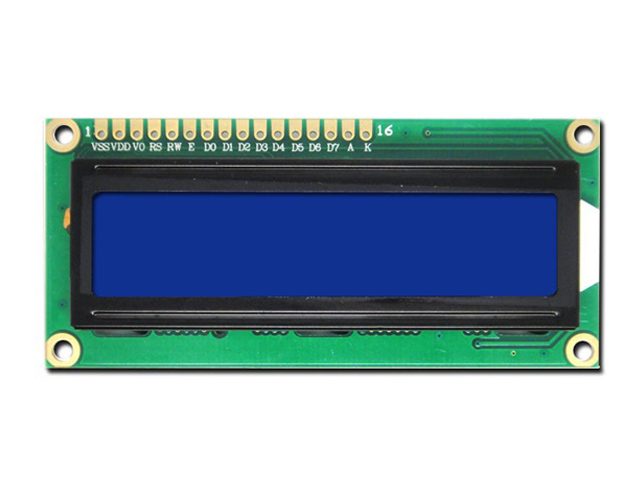


Figure 6 LCD

1. **Water Pump**

A water pump that can be submerged in water is known as a submersible water pump. They are frequently employed in situations when water needs to be pumped vertically from a low-lying location. Small submersible water pumps are frequently driven by DC motors that operate at low voltages like 3V or 5V. These pumps typically use a small amount of power and may pump water at low flow rates. Additionally, they are lightweight and tiny, making them simple to handle and install. Some DC submersible water pumps include an on/off switch and built-in thermal protection to guard against overheating.

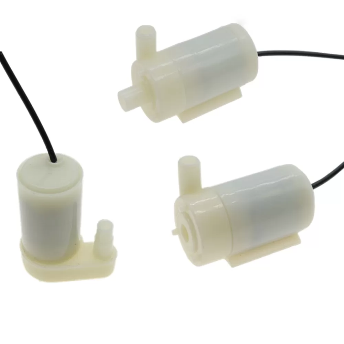


Figure 7 Water pumps

1. **Oscillator 8MHZ**

8MHz oscillator (8MHz) is a device that produces a continuous electrical signal at a frequency of 8 million cycles per second. For electrical circuits like microcontrollers, clocks, and other digital devices, it is often utilized as a timing source. Numerous technologies, including as quartz crystals, LC circuits, and RC circuits, can be used to build oscillators. They are frequently contained in tiny, low-cost gadgets called crystals or ceramic resonators. The oscillator has two pins (CLK in and CLK out) that are connected to the microcontroller.



Figure 8 8MHZ Oscillator

1. **Potentiometer**

Potentiometer is one kind of variable resistor that can be used to modify a circuit's resistance. The resistance of a 10k potentiometer is 10,000 ohms (10k). Potentiometers are frequently employed in a range of electronic applications, including circuit modifications, tone control, and volume control. Potentiometer has three terminals: the two outer terminals are the end terminals, and the middle terminal is the wiper terminal.



Figure 9 Potentiometer

The implementation of our Smart Irrigation System is divided into two stages:

1. Hardware design
2. Software design

The software design was accomplished using mikroC which is a proprietary integrated development environment (IDE) and programming language for microcontrollers. MikroC syntax is similar to C language. MikroC also has a built-in library of pre-written functions for a wide range of peripheral devices, including LCD displays, sensors, communication interfaces, and more. However, in our project we used only the following built-in libraries: Conversions, C\_String, Lcd, Lcd\_Constants. We call the functions that we read the sensors values from in the main function in the while loop that works forever as long as there is a power source connected to the system.

Another software tool used is Proteus Design Suite that is used for designing, simulating, and testing electronic circuits and microcontroller-based systems. Proteus helped us to design and test circuits and microcontroller systems in a virtual environment, before building and testing the design on real hardware.

Below is the code programmed using mikroC and the design simulated on Proteus:

// Lcd pinout settings

sbit LCD\_RS at RD0\_bit;

sbit LCD\_EN at RD1\_bit;

sbit LCD\_D7 at RD7\_bit;

sbit LCD\_D6 at RD6\_bit;

sbit LCD\_D5 at RD5\_bit;

sbit LCD\_D4 at RD4\_bit;

// Pin direction

sbit LCD\_RS\_Direction at TRISD0\_bit;

sbit LCD\_EN\_Direction at TRISD1\_bit;

sbit LCD\_D7\_Direction at TRISD7\_bit;

sbit LCD\_D6\_Direction at TRISD6\_bit;

sbit LCD\_D5\_Direction at TRISD5\_bit;

sbit LCD\_D4\_Direction at TRISD4\_bit;

void check\_moisture();

unsigned int ADC\_read1(void);

unsigned int ADC\_read2(void);

void read\_temp(void);

void measure\_distance();

void msDelay (unsigned int mscnt);

void usDelay(unsigned int mscnt);

float temp;

int Distance;

char txt\_dist[7];

char tempr[4];

unsigned int moisture;

char txt[7];

int i;

void main(){

TRISB=0x04;

PORTB=0x00;

ADCON1 = 0xC0;// All channels Analog, right justified, Fosc/16

TRISA= 0x03; //portA is input

Lcd\_Init();

Lcd\_Cmd(\_LCD\_CLEAR); // Clear display

Lcd\_Cmd(\_LCD\_CURSOR\_OFF); // Cursor off

Lcd\_Out(1,1,"HELLO");

msDelay(1000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

msDelay(1000);

while(1){

temp=Distance=moisture=0;

for (i =0 ; i<2; i++)

{

if (i == 0)

{

read\_temp();

}

if(i == 1)

{

check\_moisture();

}

}

measure\_distance();

if(20 - Distance > 7) //check the water tank level

{

if (temp < 25) //check the temperature

{

if (moisture > 150) //check the moisture value

{

FloatToStr(temp,tempr);

ltrim(tempr);

Lcd\_Out(1,1,"Temp");

Lcd\_Out(2,1,tempr);

msDelay(5000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

IntToStr(moisture,txt);

ltrim(txt);

Lcd\_Out(1,1,"Moisture: ");

Lcd\_Out(2,1,txt);

msDelay(5000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

IntToStr(Distance,txt\_dist);

ltrim(txt\_dist);

Lcd\_Out(1,1,"DISTANCE=");

Lcd\_Out(2,1,txt\_dist);

msDelay(5000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

PORTB = PORTB | 0x08; //turn on the water pump on RB3

msDelay(7000);

PORTB = PORTB & 0xF7; //turn off the water pump

continue;

}

}

}

else

{

IntToStr(temp,tempr);

ltrim(tempr);

Lcd\_Out(1,1,"Temp");

Lcd\_Out(2,1,tempr);

msDelay(5000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

IntToStr(moisture,txt);

ltrim(txt);

Lcd\_Out(1,1,"Moisture: ");

Lcd\_Out(2,1,txt);

msDelay(5000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

IntToStr(Distance,txt\_dist);

ltrim(txt\_dist);

Lcd\_Out(1,1,"DISTANCE=");

Lcd\_Out(2,1,txt\_dist);

msDelay(5000);

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Cmd(\_LCD\_CURSOR\_OFF);

PORTB = PORTB & 0xF7; //turn the water pump off

PORTB = PORTB | 0x80; //turn the buzzer on (RB7)

msDelay(2000);

continue;

}}}

unsigned int ADC\_read1(){

msDelay(50);

ADCON0 = ADCON0 & 0xC7;

msDelay(50);

ADCON0 = 0x49;// Power up, don't GO, channel AN1, Fosc/16 (500KHz)

ADCON0 = ADCON0 | 0x04; // GO ;

while(ADCON0 & 0x04); // until finish reading

return((ADRESH<<8)|ADRESL); //

}

unsigned int ADC\_read2(){

msDelay(50);

ADCON0 = ADCON0 & 0xC7;

msDelay(50);

ADCON0 = 0x41;// Power up, don't GO, channel AN0, Fosc/16 (500KHz)

ADCON0 = ADCON0 | 0x04; // GO ;

while(ADCON0 & 0x04); // until finish reading

return((ADRESH<<8)|ADRESL);

}

void check\_moisture(){

moisture=ADC\_read1();

}

void measure\_distance(){

TMR1H=0;

TMR1L=0;

PORTB =0x02; // Trigger is High

usDelay(10);

PORTB =0x00; // Trigger is Low

while (!(PORTB&0x04));

T1CON= T1CON | 0x01;

while ((PORTB.RB2));

T1CON = T1CON & 0xFE;

Distance = (TMR1L | (TMR1H << 8));

Distance = (Distance / 58.82)/2;

}

void read\_temp(){

temp=ADC\_read2();

temp=((temp\*4.88)/10)+1;

}

void msDelay(unsigned int mscnt) {

unsigned int ms;

unsigned int cnt;

for (ms = 0; ms < mscnt; ms++) {

for (cnt = 0; cnt < 155; cnt++);//1ms

}

}

void usDelay(unsigned int uscnt) {

unsigned int us;

for (us = 0; us < uscnt; us++) {

asm NOP; // Takes 0.5us

asm NOP; // Takes 0.5us

}

}

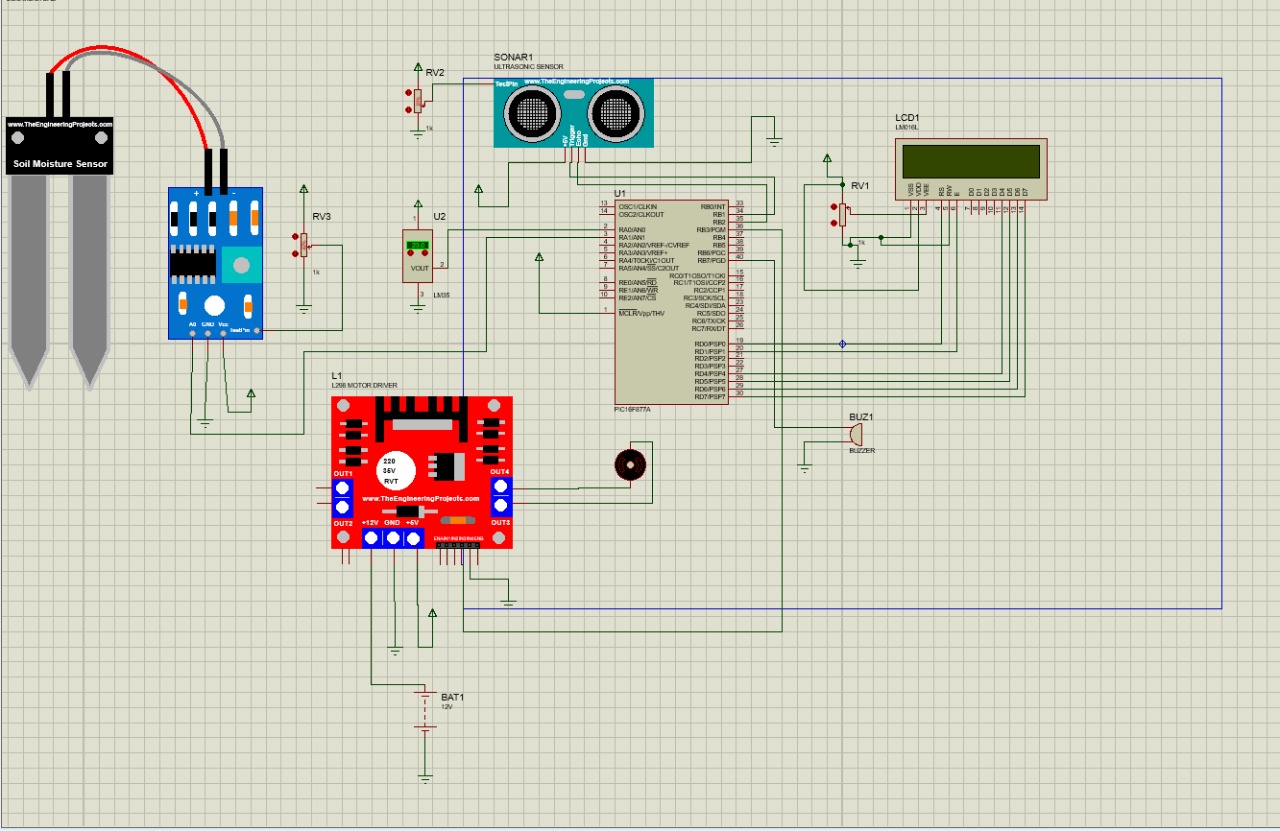


Figure 10 Proteus Electrical Design

The hardware design consists sensors that are used to implement our suggested design, which detects whether water should be pumped based on the parameters set using the microcontroller. The system measures the water level in the tank with an ultrasonic sensor before switching an H-bridge to run the water pump. To power the PIC, sensors, and H-Bridge, respectively, we required two separate power sources. A 5 volts power source from the Easy-pic was used to power the PIC microcontroller and the sensors also, a 12 volts power supply was used for the H-bridge as we needed higher power to power the water pump. The water pump will open and water will flow into the soil if the prerequisites for irrigation are satisfied and there is enough water in the tank. A buzzer will alert the farmer that the irrigation system failed if the water tank is empty, and the water pump will shut off.

First, the ultrasonic sensor HC-SR04 is used to check whether the water tank has enough water to irrigate the soil. If there is less than 7 cm of water in the water tank, the water pump will turn off and the buzzer will start. This will inform the farmer that the water tank needs to be refilled with water. On the other hand, if there is enough water the temperature sensor LM35 will check if air temperature is less than the suitable range. We found that the average air temperature for plants must not exceed 27 degrees Celsius. If the air temperature is above this value, the buzzer will start. If the temperature is less than 27 the moisture sensor will check whether the soil had enough moisture. After testing the sensor at different moisture levels, we found that a suitable moisture level must be less than 150.

So, to irrigate the soil the previous conditions must be achieved to turn on the water pump for 7 seconds.

# Flow Chart

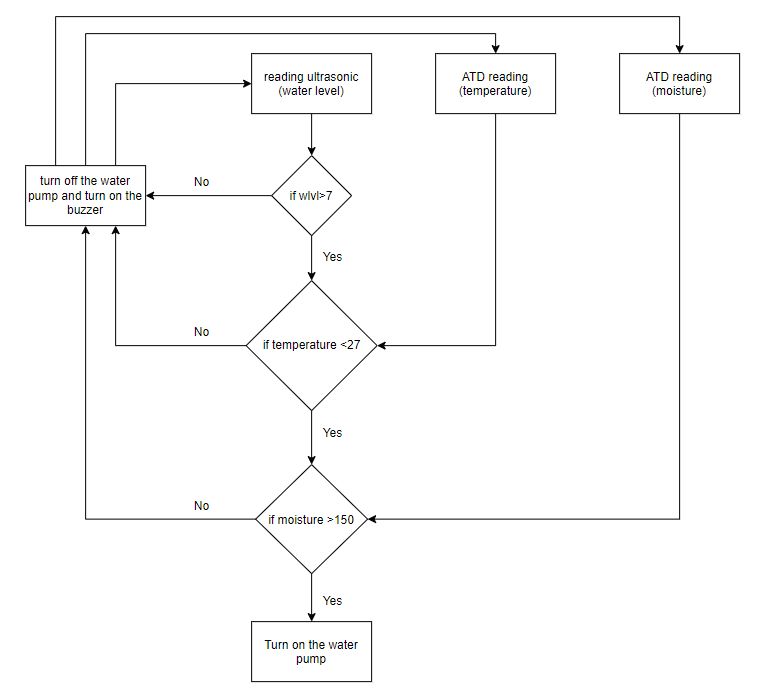
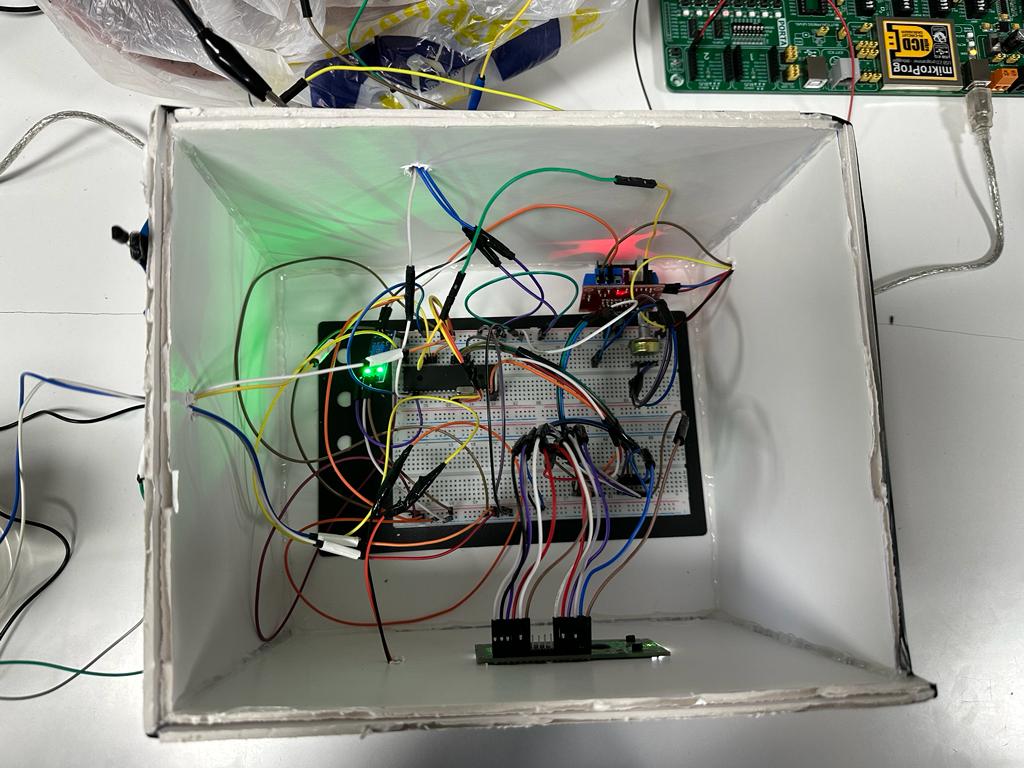


Figure 11 Flowchart diagram

# Prototype

A picture containing indoor, wall, cluttered

Description automatically generated



# Problems and Recommendations

We encountered a number of issues while developing and testing our design. One of the main problems faced was the ability to add two Analog pins without affecting both readings. At the beginning, when we added the moisture sensor and temperature sensor to the design, the temperature values drifted from the true values. This problem was from The ADC read function as we have to clear the ADCON0 register before changing the channel for the other analog sensor. After several trials, we were able to find the correct values for both analog sensors. We had difficulties in getting the temperature values because of the drift in temperature. The LM35's output can drift over time, causing it to give inaccurate readings if it is not calibrated regularly. Also, noise can affect the precision of temperature measurements when using the LM35. After trying different LM35 sensors we found that using the original sensor would minimize the drift in values. In addition, because PIC16F877A has a single ADC module we had to ensure that the timing of the ADC conversions is properly synchronized with the timing of the sensor readings to avoid errors. To avoid timing problems, we had to calculate our delays precisely. We have tried to do the three types of interrupts but our project fails every time we try one of them.

# Conclusion

The objective behind the proposed Smart Irrigation System is to assist farmers in watering their crops without the need for human intervention taking into consideration water conservation. There are a few suggestions for the project's future work that may be taken into consideration in order to improve the system and make it more effective such as connecting the system with a mobile application. Collected data can then be accessed through mobile applications. Another improvement could be adding a PH sensor analyze the amount of fertilizers found in the soil.

In conclusion we believe that we achieved our main goal which was to use what we learned about the 16F877A microcontroller, sensors, and how to integrate everything into an embedded system. This system can help governments cooperate with the implementation of afforestation projects by providing water conservation solutions.