 **ARDUİNO İLE OTOMATİK SERA SULAMA SİSTEMİ**

**Abdullah Erzin-170216014**

**Özgür Gökcük-170216015**

**Giriş**

Bu proje bir seranın su, nem, oksijen ihtiyaçlarını kontrol eden ve bunları değerlendirip eyleme geçen bir tasarımdır. Arduino, DHT11 sıcaklık ve nem sensörü, DC motor, fan ve BMP180 basınç sensörü kullanılarak tasarlanmıştır. Bu sistem geliştirilebilir ve güvenilir bir araç olarak gelecek sera sistemlerine öncü bir şekilde yol gösterebilir.

Projede toprakta bir adet DHT11 ve sera sisteminin dışında bir adet DHT11 ile BMP180 sensörleri bulunmaktadır. Dışarıda bulunan iki sensör ve sisteme daha önceden girilmiş rakım bilgisi ile havanın nem ve basınç oranlarına bakarak yağmur yağıp yağmayacağı kontrol edilir. Toprakta bulunan sensör yardımı ile toprağın sulanmaya ihtiyacının olup olmadığı kontrol edilir. Bu iki veri birleştirilerek ihitiyaç anında sulama sistemi devreye girer veya seranın çatısı açılır. Fanlar yardımıyla içeride ölçülen fazla nem istenilen düzeye gelene kadar içeriyi havalandırır ve ortamı stabil düzeyde tutar.

**Gerekli Malzemeler**

* 1 adet Arduino Uno
* 2 adet DHT11 sıcaklık ve nem sensörü
* 4 adet DC motor
* 1 adet BMP180 basınç sensörü
* 1 adet sera sistemi
* Yeterli uzunlukta boru

**Yapım Aşamaları**

1. **Sera Sisteminin Hazırlanması**
   1. **Gerekli Malzemeler**

* Taban yüzeyi hariç 5 yüzeyi açık dikdörtgen prizma kutu
* Taban yüzeyini örtecek kadar toprak
* Toprağa ekmek için tohum
* Sera sistemini üzerinin örtmeye yetecek kadar muşamba

1. **Akıllı Sistemin Kurulması**

DHT11 sensörlerinden bir tanesi hazırlanan toprağın içine gömülür. Hazırlanan sera sisteminin dışında üst kısma diğer DHT11 sensörü ile BMP180 sensörü yerleştirilir. Bir adet DC motor fana bağlanarak sera sisteminin en arkasında yere dayalı bir şekilde yerleştirilir. Bir adet DC motor da sulama sistemine yerleştirilir. Toprakta su ihtiyacı olduğu anda DC motor çalışarak sulama sistemi devreye girer. Diğer 2 adet DC motor ise tavandaki açılır kapanır muşambayla bağlantılı bir şekilde biri sistemin ön tarafına diğeri ise sistemin arka tarafına yerleştirilir. Yağmurun yağma durumunda toprağın su ihtiyacı varsa seranın üzerindeki muşamba açılır ve bir süre beklenir. Toprakta herhangi bir nem artışı olmadığı durumda muşamba kapanır ve yerine sulama sistemi devreye girer. Ve borular yardımıyla sulama işlemi gerçekleştirilir.

**Şematik Çizim**

**Kodlar**

**Kütüphane Ekleme ve Pin tanımlama Bölümü**

#include <DHT.h> //DHT sensörleri kitaplığını taslağımıza dahil ediyoruz.

#include <Wire.h> // I2C kütüphanesi

#include <SFE\_BMP180.h> //Basınç sensörü kitaplığını taslağımıza dahil ediyoruz.

#define DHTPIN1 2 //DHT11'in veri pinini Arduino'daki hangi pine bağladığımızı belirtiyoruz.

#define DHTTYPE1 DHT11 //Hangi DHT sensörünü kullandığımızı belirtiyoruz.

#define DHTPIN2 3 //DHT11'in veri pinini Arduino'daki hangi pine bağladığımızı belirtiyoruz.

#define DHTTYPE2 DHT11 //Hangi DHT sensörünü kullandığımızı belirtiyoruz.

#define ALTITUDE 40.0 //Bulunduğunuz şehrin rakım değeri.

DHT dht1(DHTPIN1, DHTTYPE1); //Toptak için belirttiğimiz değişkenleri kaydettiriyoruz.

DHT dht2(DHTPIN2, DHTTYPE2); //Dışarısı için belirttiğimiz değişkenleri kaydettiriyoruz.

SFE\_BMP180 bmp180;

**Değişkenleri Tanımlama Bölümü**

int gecenzaman=0;

int basinckarsilastirma[24] ={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}; //İçinde 24 adet değişken saklayan bir Array yaratıyoruz. Bu değişkenlerin her biri saat başı basınç değerlerini tutacak.

int toplam=0;

int ortalama=0;

int fark=0;

int tahmin;

float sicaklik1;

float sicaklik2;

float nem1;

float nem2;

float basinc;

int DC\_1=5;

int DC\_2=6;

int DC\_3=7;

int DC\_4=8;

**Void Setup Bölümü**

pinMode(DC\_1,OUTPUT);

pinMode(DC\_2,OUTPUT);

pinMode(DC\_3,OUTPUT);

pinMode(DC\_4,OUTPUT);

Serial.begin(9600);

dht1.begin(); //DHT11 ısı ve nem sensörünü ölçümlere başlattırıyoruz.

dht2.begin();

bmp180.begin(); // BMP180 basınç sensörünü ölçümlere başlattırıyoruz.

}

**Nem Ölçme Bölümü**

sicaklik1 = dht1.readTemperature(); //Detaylı bir şekilde göstermek için bir de float (küsüratlı) şeklinde değişken yapıyoruz.

nem1 = dht1.readHumidity(); //Detaylı bir şekilde göstermek için bir de float (küsüratlı) şeklinde değişken yapıyoruz.

sicaklik2 = dht2.readTemperature(); //Detaylı bir şekilde göstermek için bir de float (küsüratlı) şeklinde değişken yapıyoruz.

nem2 = dht2.readHumidity(); //Detaylı bir şekilde göstermek için bir de float (küsüratlı) şeklinde değişken yapıyoruz.

basinc = bmp180.readPressure(); //Detaylı bir şekilde göstermek için bir de float (küsüratlı) şeklinde değişken yapıyoruz.

toplam = basinckarsilastirma[0] + basinckarsilastirma[1] + basinckarsilastirma[2] + basinckarsilastirma[3] + basinckarsilastirma[4] + basinckarsilastirma[5] + basinckarsilastirma[6] + basinckarsilastirma[7] + basinckarsilastirma[8] + basinckarsilastirma[9] + basinckarsilastirma[10] + basinckarsilastirma[11] + basinckarsilastirma[12] + basinckarsilastirma[13] + basinckarsilastirma[14] + basinckarsilastirma[15] + basinckarsilastirma[16] + basinckarsilastirma[17] + basinckarsilastirma[18] + basinckarsilastirma[19] + basinckarsilastirma[20] + basinckarsilastirma[21] + basinckarsilastirma[22] + basinckarsilastirma[23]; //Belirli aralıklarla ölçülen basınç değerlerini topluyoruz.

ortalama = toplam / 24; //Sonra 3'e bölerek ortalamasını almış oluyoruz.

fark = basinc - ortalama; //Daha sonra güncel basınç değerini hesaplamış olduğumuz ortalamadan çıkararak aradaki farkı buluyoruz.

int nem\_anlik;

nem\_anlik = nem1; //Yağmurun yağdığını kontrol etmek ve topraktaki nem oranının artışına bakmak için o anın değerinin kaydeden değişken atadık.

**Değerlendirme Bölümü**

if (nem\_anlik < 50) // Topraktaki nem %50'nin altında ise

{

if (fark <= -4) // Olası bir yarğmur yağma durumu var ise

{

digitalWrite(4,HIGH);// Çatıyı açan DC motoru çalıştır

delay(3000); // çatının açılma süresi (DEĞİŞEBİLİR)

digitalWrite(4,LOW); // Çatı tamamen açıldığı anda DC motorun durur

if (nem1 > nem\_anlik && nem1 > 50 ) // Yağmurun yağıp yağmadığı ve topraktaki nem oranını olması gerektiği duruma gelip gelmediği kontrol ediliyor.

{

digitalWrite(5,HIGH); // Gerekli koşullar sağlandıysa çatıyı kapatacak olan DC motor çalışır

delay(3000); // Çatının kapanma süresi

digitalWrite(5,LOW); // Çatı tamamen kapandığı anda DC motor durur.

}

}

else {

digitalWrite(7,HIGH); // Gerekli koşullar sağlanmadıysa sulamı sistemi çalışır.

if(nem1 > 50) // Topraktaki nem tekrar kontrol edilir. Ve gerekli nem oranı sağlandıysa

{

digitalWrite(7,LOW); // Sulama sistemi kapatılır.

}

}

}

**Basınç Ölçme Bölümü**

switch (gecenzaman) //Saatlik ölçümler...

{

case 180:

basinckarsilastirma[0] = basinc;

break;

case 360:

basinckarsilastirma[1] = basinc;

break;

case 540:

basinckarsilastirma[2] = basinc;

break;

case 720:

basinckarsilastirma[3] = basinc;

break;

case 900:

basinckarsilastirma[4] = basinc;

break;

case 1080:

basinckarsilastirma[5] = basinc;

break;

case 1260:

basinckarsilastirma[6] = basinc;

break;

case 1440:

basinckarsilastirma[7] = basinc;

break;

case 1620:

basinckarsilastirma[8] = basinc;

break;

case 1800:

basinckarsilastirma[9] = basinc;

break;

case 1980:

basinckarsilastirma[10] = basinc;

break;

case 2160:

basinckarsilastirma[11] = basinc;

break;

case 2340:

basinckarsilastirma[12] = basinc;

break;

case 2520:

basinckarsilastirma[13] = basinc;

break;

case 2700:

basinckarsilastirma[14] = basinc;

break;

case 2880:

basinckarsilastirma[15] = basinc;

break;

case 3060:

basinckarsilastirma[16] = basinc;

break;

case 3240:

basinckarsilastirma[17] = basinc;

break;

case 3420:

basinckarsilastirma[18] = basinc;

break;

case 3600:

basinckarsilastirma[19] = basinc;

break;

case 3780:

basinckarsilastirma[20] = basinc;

break;

case 3960:

basinckarsilastirma[21] = basinc;

break;

case 4140:

basinckarsilastirma[22] = basinc;

break;

case 4320:

basinckarsilastirma[23] = basinc;

break;

}

**Basınç Değerlendirme Fonksiyonu**

//En başta girdiğimiz rakım bilgisini kullanarak, ham basınç değerleri deniz seviyesi basınç değerlerine dönüştürülüyor.

float readPressure()

{

char status;

double T,P,p0,a;

status = bmp180.startTemperature();

if (status != 0)

{

delay(status);

status = bmp180.getTemperature(T);

if (status != 0)

{

status = bmp180.startPressure(3);

if (status != 0)

{

delay(status);

status = bmp180.getPressure(P,T);

if (status != 0)

{

p0 = bmp180.sealevel(P,ALTITUDE);

return p0;

}

**Şematik Çizim**