

**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Bilgisayar Bilimi Dersi**

**Rapor**

<b>Rapor No</b>	Proje-2
<b>Rapor Tarih</b>	29.05.2018
<b>Proje Adı</b>	Arduino ile Meteoroloji İstasyonu

**Bilgisayar Bilimi Öğretmeni**

**Ersin TÜTÜNCÜ**

**2017-2018**



**T.C.**

**MİLLİ EĞİTİM BAKANLIĞI**

**SİVAS MERKEZ**

**Sivas Fen Lisesi**

**Proje Grup**

<b>Proje Görev</b>	<b>Numara</b>	<b>Ad-Soyad</b>
Proje Yönetimi	140	Muhammed Bera KOÇ
Doküman Yönetimi	38	Gizem BULUT
Lojistik Yönetim	140	Muhammed Bera Koç
Yazılım Geliştirme	325	Osman Sefa KILIÇ
Web ve GitHub Yönetimi	38	Gizem BULUT
Sunum Yönetimi	236,181	Şevval CEYLAN,Asya Dila SUBAŞI

## İÇİNDEKİLER

ÖZET .....	4
Anahtar Kelimeler .....	4
ABSTRACT .....	4
Key Words .....	4
Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması.....	5
• Görev Dağılımı ve Sorumlusu.....	5
• Görev süresince sürdürülen eylemler.....	6
• Görevlerin iş yükü şeması .....	7
• Yoklama Çizelgeleri .....	8
Haftalık İş Katkı Cetvelleri.....	9
GİRİŞ .....	9
1.Projenin Açıklaması .....	9
Donanım Yapısı:.....	10
a.Gömülü Sistemler Mimarisi ve Devre Tasarımı .....	10
b.Mekanik Sistem Mimarisinin Tanıtılması: .....	12
PID .....	17
4.Yazılım Yapısı .....	19
SONUÇ .....	36
1.Bilgi Düzeyine Katkıları: .....	36
2.Teknolojik Katkıları: .....	36
3.Ekip Çalışması Katkıları .....	37
4.Aksayan Yönler: .....	37
5.Görüş ve Öneriler: .....	37

## **ÖZET**

İkinci proje kapsamında kullanılan elemanlar ile meteoroloji istasyonu yapılacaktır. Bu projeden basınç, sıcaklık ve nem değerlerini ölçüp daha sonra ekrana yansıtması beklenmektedir. Bu amaçla kullanılan nem ve sıcaklık sensörü ve barometre sensörünün değerleri ölçebilmesi sayesinde gerekli kodlamalar ile robot değerleri ölçülecek ve daha sonra ekrana yansıtacaktır.

## **Anahtar Kelimeler**

Arduino, Meteoroloji İstasyonu, Nem ve Sıcaklık Sensörü, Barometre sensörü

## **ABSTRACT**

In the second project, the meteorological station will be constructed with the elements used in the project. It is expected to measure the pressure, temperature and humidity values from the project and then reflect it on the screen. Due to the measurement of the humidity and temperature sensor and the barometer sensor used for this purpose, the necessary codes and robot values will be measured and then reflected on the screen.

## **Key Words**

Arduino, Meteorology Station, Humidity and Temperature Sensor, Barometer sensor

## Proje Görev Dağılımı Listesi ve Görev Dağılımı Açıklaması

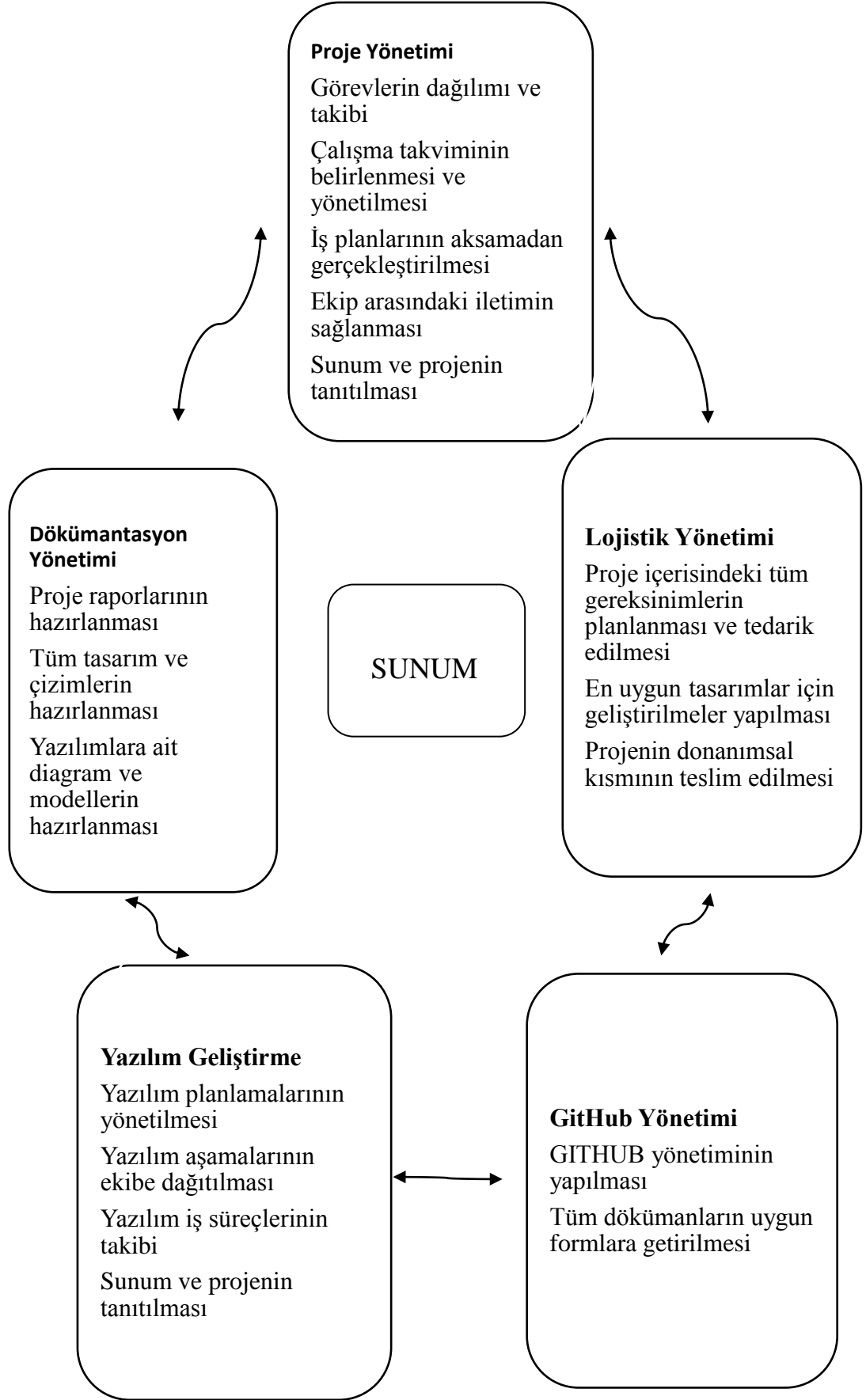
- Görev Dağılımı ve Sorumlusu

Proje Yönetimi	Görev dağılımı ve takibinden sorumlu kişi, aynı zamanda proje grubunun çalışma takvimini ve düzenini ayarlamaktadır. Grupta bulunan kişilerle iletişim halinde olup projenin yönetimini sağlar.	Muhammed Bera Koç
Döküman Yönetimi	Projenin tüm tasarım ve çizimlerinden, proje raporunun sunulmasından, dökümanların uygun forma getirilmesinden kodlamaya ait diagram ve modellerin hazırlanması ve web sitesi tasarımından sorumlu olan kişidir.	Gizem BULUT
Lojistik Yönetimi	Projede kullanılacak tüm elemanların, malzemelerin belirlenmesi ve temin edilmesi, en uygun tasarımın yapılması için geliştirmelerin yapılmasıyla ve projenin donanımsal kısmının tanıtılması ile ilgilenen kişidir.	Muhammed Bera Koç
Yazılım Geliştirme Yönetimi	Yazılım için araştırmaların yapılması, yazılım aşamalarının proje grubuna dağıtılması, Yazılım ile ilgili raporların hazırlanarak ilgili bölüme (döküman yönetimine) aktarılması yazılım ve süreç testlerinin gerçekleştirilmesi ile ilgilenen kişidir.	Osman Sefa KILIÇ
GitHub Yönetimi	Projenin GitHub yönetiminin yapılması, döküman yöneticisinden almış olduğu raporlar ile GitHub'a işlemekle sorumlu olan kişidir.	Gizem BULUT
Sunum Yönetimi	Proje teslim zamanında sunumun, yapılan tüm işlemlerin uygun bir biçimde anlatılmasından, rapor ve evrakların eksiksiz bir şekilde sunulmasından ve önerilere, sorulara uygun çözümler üretmekten sorumludur.	Asya Dila SUBAŞI Şevval CEYLAN

- **Görev süresince sürdürülen eylemler**

<b>1.Hafta</b>	Proje Hakkında Bilgi Edinme Görev Dağılımını Yapma
<b>2.Hafta</b>	Proje Hakkında Bilgi Edinme Malzeme Seçimi
<b>3.Hafta</b>	malzemelerin temin edilişi
<b>4.Hafta</b>	Mekanik ve Elektronik Tasarım
<b>5.Hafta</b>	Mekanik ve Elektronik Tasarım
<b>6.Hafta</b>	Yazılım
<b>7. Hafta</b>	Yazılım
<b>8. Hafta</b>	Test aşamaları ve ortaya çıkan sorunların çözümü
<b>9. Hafta</b>	Ortaya çıkan sorunların çözümü Grup elemanlarına ait iş yükünün tamamlanması (rapor,,github)
<b>10. Hafta</b>	Grup elemanlarına ait iş yükünün tamamlanması (rapor,,github) Sunuma hazırlık

- Görevlerin iş yükü şeması



## Yoklama Çizelgeleri

<b>GRUP</b> <b>TARİHLER</b>	<b>Bera KOÇ</b>	<b>Gizem BULUT</b>	<b>Asya Dila SUBAŞI</b>	<b>Şevval CEYLAN</b>	<b>Osman Sefa KILIÇ</b>
<b>20.02.2018</b>	+	+	+	+	+
<b>27.02.2018</b>	+	+	+	+	+
<b>06.03.2018</b>	+	+	+	+	+
<b>13.03.2018</b>	+	+	+	+	+
<b>20.03.2018</b>	+	+	+	+	+
<b>27.03.2018</b>	+	+	+	+	+
<b>03.04.2018</b>	+	+	+	+	+
<b>10.03.2018</b>	+	+	+	+	+
<b>08.05.2018</b>	+	+	+	+	+
<b>15.05.2018</b>	+	+	+	+	+



## Haftalık İş Katkı Cetvelleri

	Yapılan iş
20.02.2018	Proje araştırması
27.02.2018	Malzeme araştırması
06.03.2018	Malzeme Teminatı
13.03.2018	Elektronik ve Mekanik Tasarım
20.03.2018	Elektronik ve Mekanik Tasarım
27.03.2018	Yazılım
03.04.2018	Yazılım
10.03.2018	Deneme ve Test Aşamaları.
08.05.2018	Çıkan Sorunların düzeltilmesi Rapor,Github düzenlenmesi.
15.05.2018	Rapor,Github düzenlenmesi.Sunuma hazırlık yapımı.


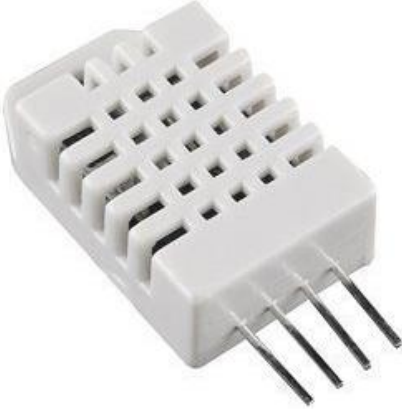
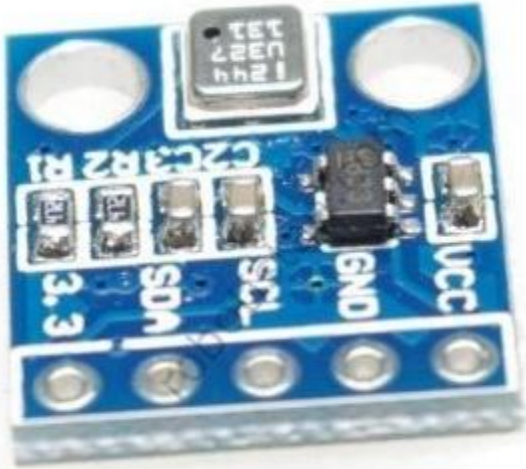
## GİRİŞ

### 1.Projenin Açıklaması

Yapılan projede amaç; ortamda bulunan meteorolojik değerleri ölçebilecek vedaha sonra ekrana yansıabilecek bir robot tasarımı geliştirebilmektir.Robot değerleri özdenetimli olarak ölçebilmelidir.robot gerekli kodlamalar ile sensörlerin verileri ölçüp ekrana yansıtması sayesinde çalışır.Nem ve sıcaklık,gaz ve basınç sensörlerinin çalışma prensibi direnç üzerine kuruludur.bu sensörler ortamda bulunan direnç farklılıkları ve değişimlerinden yararlanarak ölçümlemleri yaparlar.

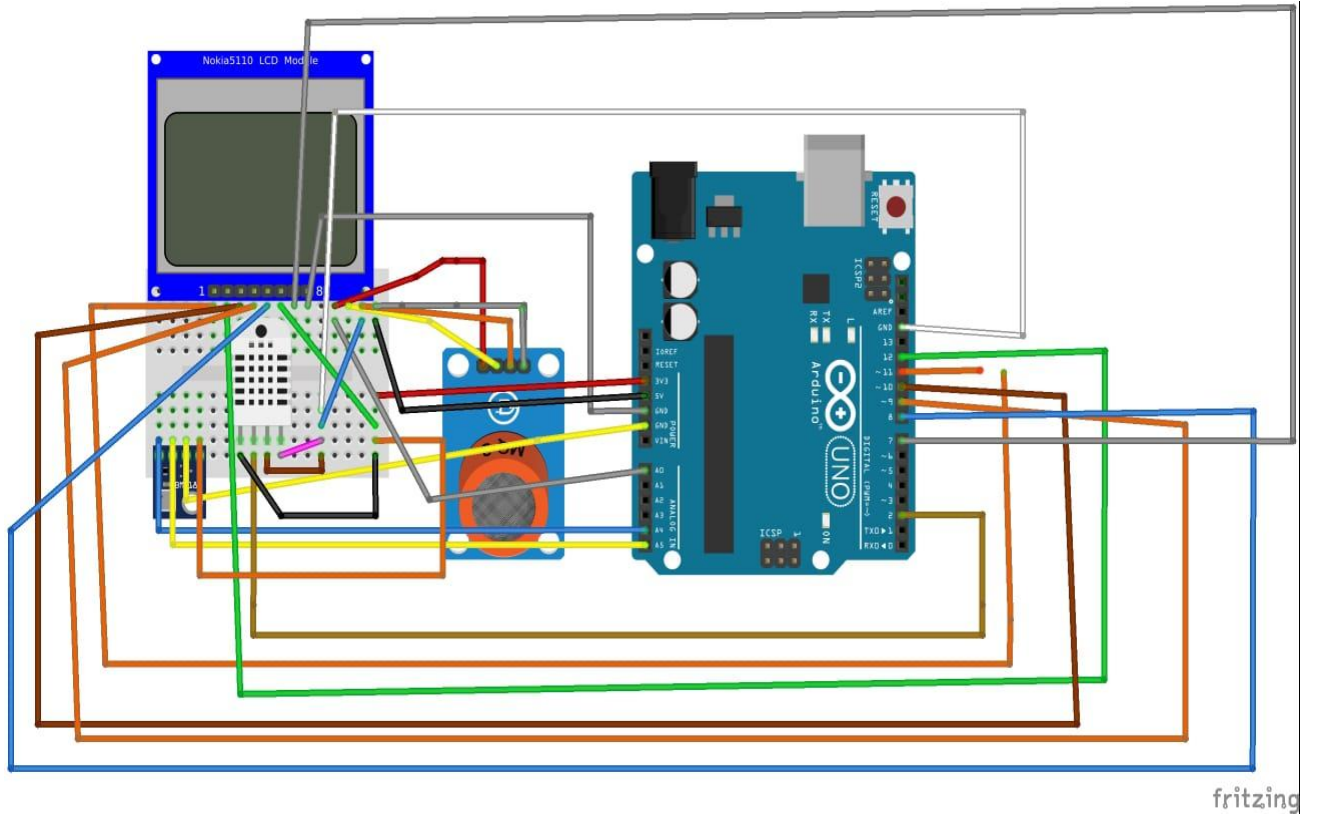
## Donanım Yapısı:

### a.Gömülü Sistemler Mimarisi ve Devre Tasarımı

	<p><b>Arduino Uno R3</b> Arduino Uno; Atmega328 temelli bir mikrodeneleyici kartıdır. Üzerinde 14 adet dijital giriş/çıkış pini (6 tanesi PWM çıkışı olarak kullanılabilir), 6 analog giriş, 16Mhz kristal, usb soketi, güç soketi, ICSP konektörü ve reset tuşu bulundurmaktadır. Kart üzerinde mikrodeneleyicinin çalışması için gerekli olan her şey bulunmaktadır. Kolayca usb kablosu üzerinden bilgisayara bağlanabilir, adaptör veya pil ile çalıştırılabilir.</p>
	<p><b>DHT22 Sıcaklık ve Nem Sensörü</b> DHT22 sıcaklık ve nem algılayıcı kalibre edilmiş dijital sinyal çıkışı veren gelişmiş bir sensör birimidir. Yüksek güvenilirliktedir ve uzun dönem çalışmalarda dengelidir. 8 bit mikroişlemci içerir, hızlı ve kaliteli tepki verir.</p>
	<p><b>BMP180 Basınç Sensörü</b> BMP180 dijital çıkışı basınç sensörü, ortamdaki hava basıncını ölçerek dijital şekilde çıkış veren oldukça küçük boyutlu ve çok kullanışlı bir üründür. Hava basıncına göre sensör yüksekliğinin anlaşılmasından dolayı multicopter projeleri başta olmak üzere bir çok sistemde kullanılabilir.</p>

	<p><b>Jumper Kablo (E-E/D-E)</b> Devre elemanlarının bağlantılarını gerçekleştirmek için kullanabiliriz.</p>
	<p><b>MQ-2 GAZ SENSÖRÜ</b> MQ-2 gaz sensörü 300 ile 10.000 ppm konsantrasyonlarda yanıcı gaz ve sigara dumanını algılayacak türden bir sensördür. Çıkış olarak ortamdaki gazın durumuna göre verdiği analog voltaj bilgisi sensörü kullanmayı oldukça kolaylaştırmaktadır.</p>
	<p><b>NOKIA 5110 LCD EKRAN</b> Arduino ve farklı mikrodenetleyici platformları ile beraber kullanabileceğiniz, kolay kullanımlı Nokia 5110 ekranı projeleriniz için güzel bir grafik LCD'dir. Bu projede sensörler ile ölçülen değerleri ekrana yansıtmak için kullanacağız.</p>

## Devre Tasarımı:



## b.Mekanik Sistem Mimarisinin Tanıtılması:

Yaptığımız bu projede yapılan basınç,sıcaklık,nem ölçümleri sensörler ile yapılmaktadır.Sıcaklık ve nem ölçümü DHT22 ;basınç ölçümü ölçümü BMP 180,gazların ölçümü ise MQ-2 gaz sensörü tarafından yapılmaktadır.Ölçülen bu veriler daha sonra gerekli kodlamalar ve bağlantılar yapılarak Nokia 5110 LCD ekrana yansıtılır .

## 1.Arduino UNO R3



### Özellikler

Mikrodenetleyici	AT mega328
Çalışma Gerilimi	5V
Giriş Gerilimi(önerilen)	7-12V
Dijital I/O Pinleri	14(6 tanesi PWM çıkışı)
Analog Giriş Pinleri	6
Her I/O için Akım	40mA
3.3V Çıkış için Akım	50 mA
EEPROM	1KB (ATmega328)
SRAM	2KB(ATMega328)
Flash Hafıza	32KB (Atmega32bootloader)
Uzunluk	68.8mm
Genişlik	53.4mm
Ağırlık	25gr

Arduino Uno gücünü usb üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör kart üzerindeki 2.1mm merkez-pozitif güç soketinden bağlanabilir. Batarya kart üzerindeki GND ve Vin pinleri üzerinden bağlanabilir.

Kartın çalışması için sürekli olarak usb'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılabilir.

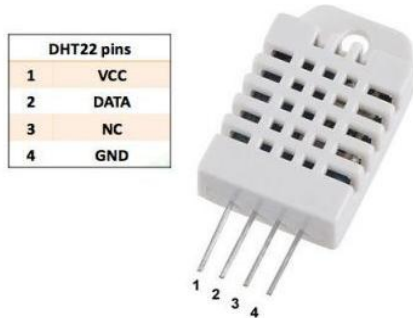
Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasındadır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir.

Uno kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur.

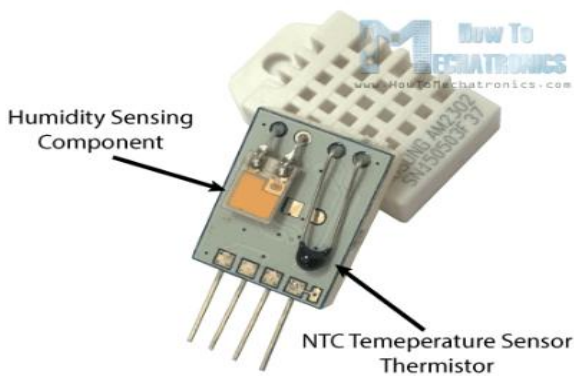
Güç pinleri:

- **VIN:** Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.
- **5V:** Bu pin regülatörden çıkan 5V çıkışı verir. Eğer kart sadece usb (5V) üzerinden çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Eğer karta güç Vin (7-12V) veya güç soketi (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
- **3V3:** Kart üzerinde bulunan 3.3V regülatörü çıkış pinidir. Maks. 50mA çıkış verebilir.
- **GND:** Toprak pinleridir.

## 2.DHT 22 SICAKLIK VE NEM SENSÖRÜ



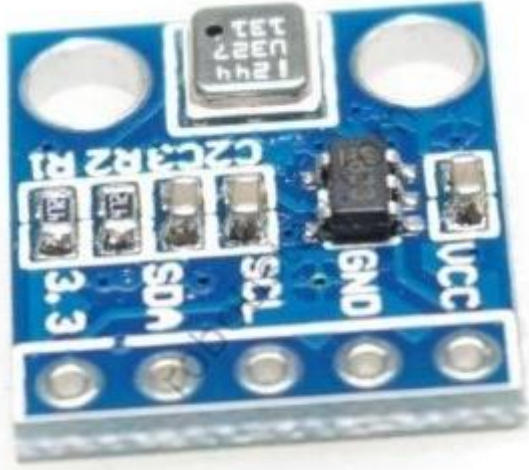
### Sensör İç Yapısı



Bir nem algılama bileşeni, bir NTC sıcaklık sensörü (veya bir termistör) ve sensörün arka tarafında bir IC'den oluşur. Nem ölçümü için, aralarında nem tutucu alt tabaka bulunan iki elektrotlu nem algılama bileşeni kullanılır. Nem değiştikçe, substratın iletkenliği değişir veya bu elektrotlar arasındaki direnç değişir. Dirençteki bu değişim, bir mikrokontrolör tarafından okunmaya hazır hale getiren IC tarafından ölçülür ve işlenir. Bir taraftan, ölçüm sıcaklıkları için bu sensörler bir NTC sıcaklık sensörü veya bir termistör kullanır. Bir termistör aslında sıcaklığın değişmesiyle direncini değiştiren değişken bir dirençtir. Bu sensörler, sıcaklıkta sadece küçük değişikliklerle dirençte daha büyük değişiklikler sağlamak için seramikler veya polimerler gibi yarı iletken malzemelerin sinterlenmesiyle yapılır. “NTC” terimi “Negatif Sıcaklık Katsayısı” anlamına gelir, bu da ısının yükselmesiyle direncin azaldığı anlamına gelir.

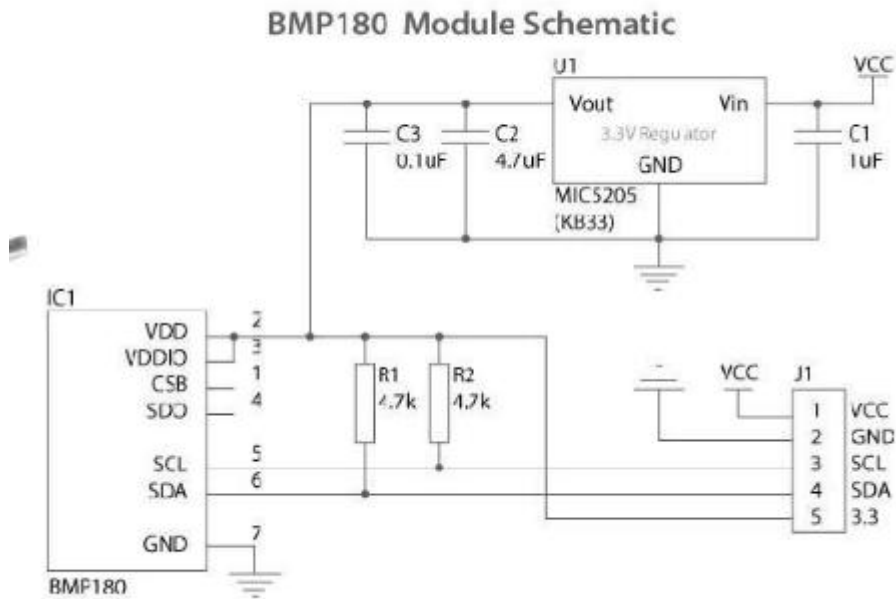


### 3.BMP 180 BASINÇ SENSÖRÜ



BMP180 birçok robot ve elektronik projesinde basınç ölçmek için kullanılan boyutları çok küçük olan bir sensördür. Hava basıncını ölçüp denetleyiciye dijital çıkış verir. Bu küçük sensör 300-1100hPa arasındaki basınç değerlerini ölçebilmektedir. Ayrıca 500 metre ile 9000 metre arasındaki yükseklik ile ilgili bilgi vermektedir. 3.3 V girişi bulunmakla birlikte 5V girişi de ayrıca bulunmaktadır.

BMP180, piezo dirençli bir sensör, dijital dönüştürücüden bir analog ve E2PROM ve seri I2C arabirimli bir kontrol ünitesinden oluşur. BMP180, kompanse edilmemiş basınç değerini verir. Mikrodenetleyici, bir basınç ölçümü başlatmak için bir başlangıç dizisi gönderir. Zaman dönüştürüldükten sonra, sonuç değeri I2C arayüzü üzerinden okunabilir. hPa cinsinden basınç hesaplamak için kalibrasyon verileri kullanılmalıdır. Bu sabitler BMP180 E2PROM'dan I2C arabirimi üzerinden yazılım başlatma sırasında okunabilir. Örnekleme hızı dinamik ölçüm için saniyede 128 örneğe (standart mod) kadar arttırılabilir. Bu durumda, sıcaklığı saniyede sadece bir kez ölçmek ve aynı zamanda tüm basınç ölçümleri için bu değeri kullanmak yeterlidir.

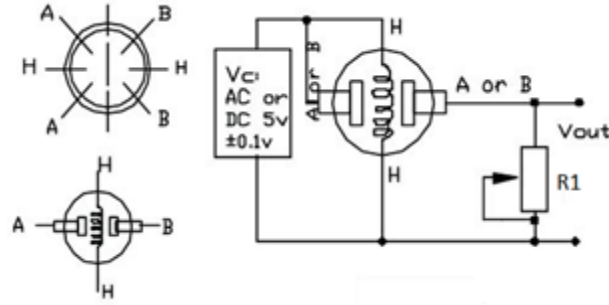


#### 4.MQ2 GAZ SENSÖRÜ



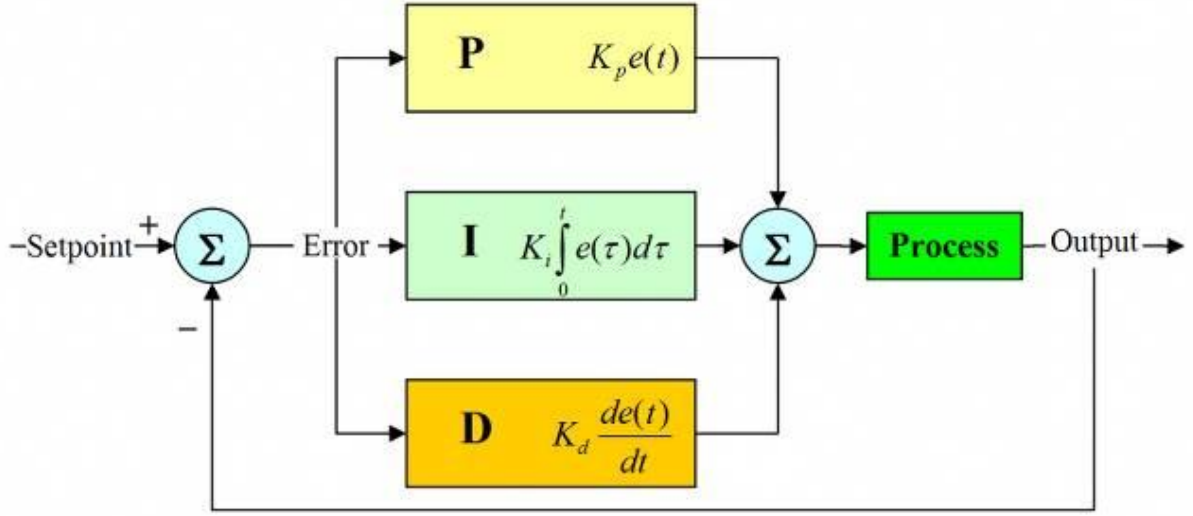
H pinlerine 5V besleyerek sensörün yeterince ısınmasını ve böylece tam olarak ve doğru ölçümlerle çalışmasını sağlarsınız. A veya B pinine 5V gerilim uygulamak sensörün diğer pinleri üzerinden analog voltaj vermesine sebep olur. Sensörün hassasiyetini, toprak ile sensörün çıkış pini arasına bağlayacağınız direnç değeri ile orantılı olarak değiştirebilirsiniz. Bu direnç değeri yapacağınız uygulamaya göre değişkenlik gösterebilir, bununla ilgili hesaplamalara datasheet'ten ulaşabilirsiniz. Örnek bir başlangıç değeri olarak 20 k $\Omega$  bir direnç kullanabilirsiniz.

MQ2, farklı yoğunluktaki farklı gazlar için direncini değiştiren bir elektrokimyasal sensöre sahiptir. Sensör, bir voltaj bölücü devresi oluşturmak için değişken bir direnç ile seri bağlanır ve değişken direnç, duyarlılığı değiştirmek için kullanılır. Gazlı elemanlardan biri, ısıtma işleminden sonra sensörle temas ettiğinde, sensörün direnci değişir. Dirençteki değişiklik sensördeki voltajı değiştirir ve bu voltaj bir mikro denetleyici tarafından okunabilir. Voltaj değeri, referans voltajı ve diğer direnç direncini bilmesiyle sensörün direncini bulmak için kullanılabilir. Sensör farklı tipte gazlar için farklı hassasiyete sahiptir.





## PID



**Şekil 9: PID kontrol Diyagramı**

PID, Oransal İntegral Türev için kullanılan bir kısaltmadır. En genel tanımıyla bir kontrol geri bildirim mekanizmasıdır. PID yönteminin en temel amacı hatayı minimize etmek, en aza indirmektir. PID kontrolünü uygulamak ve kavramak oldukça zordur. PID kontrolünde öncelikle hata tanımlaması yapılmalıdır. Hata ise referans değere olan uzaklık olarak tanımlanabilir. Açıklamak gerekirse;

Referans=İstenilen değer

Gelen=Şuan ki Konum

HATA=Referans-Gelen

### **Oransal Terim(P):**

Oransal terim, sistemden gelen hatayı bir katsayı ile çarparak hatayı küçültmeyi hedefler. Bozucu etkileri de mevcuttur. Projemizin yazılım kısmında oldukça ağırlık verdiğimiz PID kontrolünde deneme ve test aşamasında bu katsayıya büyük bir değer vermemiz gerektiğini anladık.

$$P = K_p * HATA$$

### **Integral Terimi(I):**

Integral hatanın alanını bulmak anlamına gelir. Integralin çok yükselmesini önlemek için sınırlandırmak gereklidir. Sürekli toplandığı için integral çok artarsa tekrar azalmasını beklemek zaman alır. Bu yüzden integrali sınırlamak sistemin çabuk toparlamasını sağlayacaktır.

$$I = I + (K_i * HATA * dt)$$

dt: PID fonksiyonuna her girdiğinde geçen zaman.

### **Türev Terimi(D):**

Türev sistemdeki iki örnek arasındaki zamanı hesaplar. Eğer hatada bir değişim olmadıysa türev sıfır olur.

EHata=Bir önceki hatanın değeri

$$HD = HATA - EHata$$

$$D = (K_d * HD) / dt$$

### **PID Algoritması**

$K_p$  ,  $K_i$  ve  $K_d$  katsayılarından oluşur. Bu katsayılar deneme yanılma yöntemiyle bulunur. Yapacağınız sistemde optimum katsayıları bulmak için değerde değişiklik yapıp sistemi gözlemlemeniz gerekmektedir.

$$HATA = Referans - Gelen$$

$$HD = HATA - EHata$$

$$P = K_p * Hata$$

$$I = I + (K_i * HATA * dt)$$

$$D = (K_d * HD) / dt$$

$$PID = P + I + D$$

$$EHata = Hata$$

## 4.Yazılım Yapısı

### a.Algoritmik olarak

Robot sensörlerin ölçümler yapıp ekrana yansıtması ile çalışmaktadır.Sensörlerin çalışma prensibi direnç farklılıklarına dayanır.DHT22 sıcaklık ve nem sensöründe nemin ölçümü için iki elektrodlu nem algılama bileşeni kullanılır.Nem değiştikçe, substratın iletkenliği değişir veya bu elektrotlar arasındaki direnç değişir. Sıcaklık ölçümü için ise sıcaklığın değişmesiyle direncini değiştiren bir sensör kullanılır.Isının yükselmesiile direnç azalır ve bu farklılıkla ölçüm yapılır.BP18 ise basınç ölçmek için bir başlangıç dizisi gönderir . Sonrad önüştürme zamanı, sonuç değeri arayüzü üzerinden okunabilir.MQ2 gaz sensörü de farklı gazlar için direncini değiştiren bir elektrokimyasal sensöre sahiptir.Gazlı elemanlardan biri, ısıtma işleminden sonra sensörle temas ettiğinde, sensörün direnci değişir. Dirençteki değişiklik sensördeki voltajı değiştirir ve bu voltaj bir mikro denetleyici tarafından okunabilir.Son olarak ise bu değerler gerekli kodlamalar ve bağlantılar ile Nokia LCD ekrana yansıtılır ve bu değerleri rahatlıkla görmemizi sağlar.

### b.Kod Yapısı

```
#include <DHT.h>

#include <MQ2.h>

#include <SFE_BMP180.h>.

#include <Wire.h>

#include <LCD5110_Graph.h>


#define DHTPIN 2

#define DHTTYPE DHT22

#define ALTITUDE 1285.0


DHT dht(DHTPIN, DHTTYPE);

LCD5110 myGLCD(8,9,10,11,12);

SFE_BMP180 bmp180;

extern uint8_t SmallFont[];

extern uint8_t MediumNumbers[];

extern uint8_t BigNumbers[];

extern uint8_t TinyFont[];

extern uint8_t nem_bitmap[];

extern uint8_t sicaklik_bitmap[];

extern uint8_t basinc_bitmap[];

extern uint8_t yagmurlu[];
```



```

int basinc = readPressure();

float sicaklikf = dht.readTemperature();

float nemf = dht.readHumidity();

float basincf = readPressure();


float* values= mq2.read(true);

//lpg = values[0];

lpg = mq2.readLPG();

//co = values[1];

co = mq2.readCO();

//smoke = values[2];

smoke = mq2.readSmoke();


toplama = basincarsilastirma[0] + basincarsilastirma[1] + basincarsilastirma[2] + basincarsilastirma[3] + basincarsilastirma[4] +
basincarsilastirma[5] + basincarsilastirma[6] + basincarsilastirma[7] + basincarsilastirma[8] + basincarsilastirma[9] +
basincarsilastirma[10] + basincarsilastirma[11] + basincarsilastirma[12] + basincarsilastirma[13] + basincarsilastirma[14] +
basincarsilastirma[15] + basincarsilastirma[16] + basincarsilastirma[17] + basincarsilastirma[18] + basincarsilastirma[19] +
basincarsilastirma[20] + basincarsilastirma[21] + basincarsilastirma[22] + basincarsilastirma[23];

ortalama = toplama / 24;

fark = basinc - ortalama;


if (firsttime == 0) {
    myGLCD.clrScr();

    {
        myGLCD.setFont(TinyFont);

        for (int i=84; i>=7; i--)

            myGLCD.print("METEROLOJI ISTASYONU", i, 37);

        delay(20);

        myGLCD.update();

    }


    for (int i=24; i>=19; i--) {

        myGLCD.clrScr();

        myGLCD.setFont(TinyFont);

        myGLCD.print("SENSORLER OKUNUYOR", 8, 37);

        myGLCD.print("METEROLOJI ISTASYONU", 5, i);
    }
}

```

```

delay(50);
myGLCD.update();
}

delay(7500);
firsttime= 1;

}
else {
}

//SICAKLIK GÖSTERİLİR...
myGLCD.clrScr();
myGLCD.drawBitmap(0, 0, sicaklik_bitmap, 84, 48);
myGLCD.setFont(BigNumbers);
myGLCD.printNumF(sicaklikf,1,12 ,18);
myGLCD.setFont(SmallFont);
myGLCD.print("C",72,27);
myGLCD.setFont(TinyFont);
myGLCD.print("o",70,22);
myGLCD.update();
delay(2500);

// NEM GÖSTERİLİR...
myGLCD.clrScr();
myGLCD.drawBitmap(0, 0, nem_bitmap, 84, 48);
myGLCD.setFont(BigNumbers);
myGLCD.printNumF(nemf,1,20, 20);
myGLCD.setFont(SmallFont);
myGLCD.print("%",9,28);
myGLCD.update();
delay(2500);
//GAZ GÖSTERİLİR
myGLCD.clrScr();
myGLCD.setFont(SmallFont);
myGLCD.print("GAZ",35,0);
myGLCD.setFont(SmallFont);
myGLCD.print("LPG:",0,10);

```

```

myGLCD.printNumI(lpg,25,10);
myGLCD.print("ppm",65,10);
myGLCD.print("CO:",0,25);
myGLCD.printNumI(co,25,25);
myGLCD.print("ppm",65,25);
myGLCD.print("SMOKE:",0,40);
myGLCD.printNumI(smoke,40,40);
myGLCD.print("ppm",65,40);
myGLCD.update();
delay(5000);

```

// BASINÇ GÖSTERİLİR...

```

myGLCD.clrScr();
myGLCD.drawBitmap(0, 0, basinc_bitmap, 84, 48);
myGLCD.setFont(BigNumbers);
myGLCD.printNumF(basincf,1,0,18);
myGLCD.setFont(TinyFont);
myGLCD.print("hPa",40,43);
myGLCD.update();
delay(2500);

```

// BASINÇ ÇİZELGESİ

```

myGLCD.clrScr();
myGLCD.drawRect(35, 10, 61, 47);
myGLCD.setFont(TinyFont);
myGLCD.print("BASINC (SON 24 SAAT)",6,0);
myGLCD.print("1000 hPa",0,42);
myGLCD.print("1050 hPa",0,11);

```

```

for (int i=36; i<61; i++)
{
  int koordinat;
  switch (i) {
  case 36:
    koordinat= map(basincarsilastirma[0],990,1050,46,11);
    break;
  case 37:

```

```
koordinat= map(basinckarsilastirma[1],990,1050,46,11);  
break;  
case 38:  
koordinat= map(basinckarsilastirma[2],990,1050,46,11);  
break;  
case 39:  
koordinat= map(basinckarsilastirma[3],990,1050,46,11);  
break;  
case 40:  
koordinat= map(basinckarsilastirma[4],990,1050,46,11);  
break;  
case 41:  
koordinat= map(basinckarsilastirma[5],990,1050,46,11);  
break;  
case 42:  
koordinat= map(basinckarsilastirma[6],990,1050,46,11);  
break;  
case 43:  
koordinat= map(basinckarsilastirma[7],990,1050,46,11);  
break;  
case 44:  
koordinat= map(basinckarsilastirma[8],990,1050,46,11);  
break;  
case 45:  
koordinat= map(basinckarsilastirma[9],990,1050,46,11);  
break;  
case 46:  
koordinat= map(basinckarsilastirma[10],990,1050,46,11);  
break;  
case 47:  
koordinat= map(basinckarsilastirma[11],990,1050,46,11);  
break;  
case 48:  
koordinat= map(basinckarsilastirma[12],990,1050,46,11);  
break;  
case 49:  
koordinat= map(basinckarsilastirma[13],990,1050,46,11);  
break;  
case 50:
```



```

koordinat= map(basinckarsilastirma[14],990,1050,46,11);
break;

case 51:
koordinat= map(basinckarsilastirma[15],990,1050,46,11);
break;

case 52:
koordinat= map(basinckarsilastirma[16],990,1050,46,11);
break;

case 53:
koordinat= map(basinckarsilastirma[17],990,1050,46,11);
break;

case 54:
koordinat= map(basinckarsilastirma[18],990,1050,46,11);
break;

case 55:
koordinat= map(basinckarsilastirma[19],990,1050,46,11);
break;

case 56:
koordinat= map(basinckarsilastirma[20],990,1050,46,11);
break;

case 57:
koordinat= map(basinckarsilastirma[21],990,1050,46,11);
break;

case 58:
koordinat= map(basinckarsilastirma[22],990,1050,46,11);
break;

case 59:
koordinat= map(basinckarsilastirma[23],990,1050,46,11);
break;
}

myGLCD.invPixel(i,koordinat);
myGLCD.update();
delay(20);
}

if (yonelim == 1) {

for (int c=0; c<4; c++) {

```

```

for (int i=20; i>10; i--){
myGLCD.drawBitmap(64, i, uparrow, 24, 24);
myGLCD.update();
delay(75);
}
for (int y=10; y<20; y++) {
myGLCD.drawBitmap(64, y, uparrow, 24, 24);
myGLCD.update();
delay(75);
}
}
}

```

```

if (yonelim == 0) {

```

```

for (int c=0; c<4; c++) {

```

```

for (int i=62; i<65; i++){
myGLCD.drawBitmap(i, 20, equal, 22, 17);
myGLCD.update();
delay(125);
}
for (int y=65; y>62; y--) {
myGLCD.drawBitmap(y, 20, equal, 22, 17);
myGLCD.update();
delay(125);
}
}
}

```

```

if (yonelim == -1) {

```

```

for (int c=0; c<4; c++) {

```

```

for (int i=10; i<20; i++) {
myGLCD.drawBitmap(64, i, downarrow, 24, 24);
myGLCD.update();
delay(75);
}
}
}

```

```

    }
    for (int y=20; y>10; y--) {
        myGLCD.drawBitmap(64, y, downarrow, 24, 24);
        myGLCD.update();
        delay(75);
    }
}
}
}

delay(2000);

```

//BASINÇ ORTALAMALARI

```

    myGLCD.clrScr();
    myGLCD.setFont(TinyFont);
    myGLCD.print("BASINC (SON 24 SAAT)",6,0);
    myGLCD.setFont(SmallFont);
    myGLCD.print("Ortalama: ",0,10);
    myGLCD.printNumI(ortalama,55,10);
    myGLCD.print("Guncel: ",0,25);
    myGLCD.printNumI(basinc,55,25);
    myGLCD.print("Fark: ",5,40);
    myGLCD.printNumI(fark,55,40);
    myGLCD.update();
    delay(5000);

    ++gecenzaman;
}

```

```

switch (gecenzaman){
    case 180:
        basincarsilastirma[0] = basinc;
        break;

    case 360:
        basincarsilastirma[1] = basinc;
        if ((basincarsilastirma[0]) > (basinc)) {
            yonelim=-1;
        }
}

```

```
if ((basinckarsilastirma[0]) < (basinc)) {  
yonelim=1;  
}  
if ((basinckarsilastirma[0]) == (basinc)) {  
yonelim=0;  
}  
break;
```

case 540:

```
basinckarsilastirma[2] = basinc;  
if ((basinckarsilastirma[1]) > (basinc)) {  
yonelim=-1;  
}  
if ((basinckarsilastirma[1]) < (basinc)) {  
yonelim=1;  
}  
if ((basinckarsilastirma[1]) == (basinc)) {  
yonelim=0;  
}  
break;
```

case 720:

```
basinckarsilastirma[3] = basinc;  
if ((basinckarsilastirma[2]) > (basinc)) {  
yonelim=-1;  
}  
if ((basinckarsilastirma[2]) < (basinc)) {  
yonelim=1;  
}  
if ((basinckarsilastirma[2]) == (basinc)) {  
yonelim=0;  
}  
break;
```

case 900:

```
basinckarsilastirma[4] = basinc;  
if ((basinckarsilastirma[3]) > (basinc)) {  
yonelim=-1;  
}
```

```

if ((basinckarsilastirma[3]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[3]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 1080:
basinckarsilastirma[5] = basinc;
if ((basinckarsilastirma[4]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[4]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[4]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 1260:
basinckarsilastirma[6] = basinc;
if ((basinckarsilastirma[5]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[5]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[5]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 1440:
basinckarsilastirma[7] = basinc;
if ((basinckarsilastirma[6]) > (basinc)) {
yonelim=-1;
}

```

```

if ((basinckarsilastirma[6]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[6]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 1620:
basinckarsilastirma[8] = basinc;
if ((basinckarsilastirma[7]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[7]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[7]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 1800:
basinckarsilastirma[9] = basinc;
if ((basinckarsilastirma[8]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[8]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[8]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 1980:
basinckarsilastirma[10] = basinc;
if ((basinckarsilastirma[9]) > (basinc)) {
yonelim=-1;
}

```

```

if ((basinckarsilastirma[9]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[9]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 2160:
basinckarsilastirma[11] = basinc;
if ((basinckarsilastirma[10]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[10]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[10]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 2340:
basinckarsilastirma[12] = basinc;
if ((basinckarsilastirma[11]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[11]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[11]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 2520:
basinckarsilastirma[13] = basinc;
if ((basinckarsilastirma[12]) > (basinc)) {
yonelim=-1;
}

```

```

if ((basinckarsilastirma[12]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[12]) == (basinc)) {
yonelim=0;
}
break;

```

case 2700:

```

basinckarsilastirma[14] = basinc;
if ((basinckarsilastirma[13]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[13]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[13]) == (basinc)) {
yonelim=0;
}
break;

```

case 2880:

```

basinckarsilastirma[15] = basinc;
if ((basinckarsilastirma[14]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[14]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[14]) == (basinc)) {
yonelim=0;
}
break;

```

case 3060:

```

basinckarsilastirma[16] = basinc;
if ((basinckarsilastirma[15]) > (basinc)) {
yonelim=-1;
}

```



```

if ((basinckarsilastirma[15]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[15]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 3240:
basinckarsilastirma[17] = basinc;
if ((basinckarsilastirma[16]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[16]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[16]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 3420:
basinckarsilastirma[18] = basinc;
if ((basinckarsilastirma[17]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[17]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[17]) == (basinc)) {
yonelim=0;
}
break;

```

```

case 3600:
basinckarsilastirma[19] = basinc;
if ((basinckarsilastirma[18]) > (basinc)) {
yonelim=-1;
}

```

```
if ((basinckarsilastirma[18]) < (basinc)) {  
yonelim=1;  
}  
if ((basinckarsilastirma[18]) == (basinc)) {  
yonelim=0;  
}  
break;
```

```
case 3780:  
basinckarsilastirma[20] = basinc;  
if ((basinckarsilastirma[19]) > (basinc)) {  
yonelim=-1;  
}  
if ((basinckarsilastirma[19]) < (basinc)) {  
yonelim=1;  
}  
if ((basinckarsilastirma[19]) == (basinc)) {  
yonelim=0;  
}  
break;
```

```
case 3960:  
basinckarsilastirma[21] = basinc;  
if ((basinckarsilastirma[20]) > (basinc)) {  
yonelim=-1;  
}  
if ((basinckarsilastirma[20]) < (basinc)) {  
yonelim=1;  
}  
if ((basinckarsilastirma[20]) == (basinc)) {  
yonelim=0;  
}  
break;
```

```
case 4140:  
basinckarsilastirma[22] = basinc;  
if ((basinckarsilastirma[21]) > (basinc)) {  
yonelim=-1;  
}
```

```

if ((basinckarsilastirma[21]) < (basinc)) {
yonelim=1;

}
if ((basinckarsilastirma[21]) == (basinc)) {
yonelim=0;
}
break;

case 4320:
basinckarsilastirma[23] = basinc;
if ((basinckarsilastirma[22]) > (basinc)) {
yonelim=-1;
}
if ((basinckarsilastirma[22]) < (basinc)) {
yonelim=1;
}
if ((basinckarsilastirma[22]) == (basinc)) {
yonelim=0;
}
break;
}

}

```

```

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;  
    }  
}  
}
```

## SONUÇ

### 1.Bilgi Düzeyine Katkıları:

Öncelikle grup olarak çalışmanın ve işbirliğinin katkılarını bir kez daha gördük.Bu süreçte çeşitli malzemeleri , ne işe yaradıklarını öğrendik.Bir grup olarak çalıştık ve robotumuzu meydana getirdik.Bu çalışma bize hem grup olarak çalışma konusundaki tecrübemizi artırmamızı ağırlarken,aynı zamanda robot yapma hakkındaki bilgi seviyemizi arttırdı.Proje sayesinde daha önceki projemizde kullanmadığımız çeşitli sensörleri kullandık ve bu sensörlerin kullanımı ve çalışma prensibini de öğrenmiş olduk.Devre tasarımında devre elemanlarının zarar görmemesi için yapılması gerekenleri devre tasarımının nasıl olması gerektiğini ve bağlantıları öğrendik. Devrenin kurulumu,gerekli kodlamanın yapılması, devrelerin şematik olarak gösterilmesi gibi konuları öğrendik.Robotun mekanik yapısını ve kodlamanın nasıl yapılacağını öğrendik.Bu proje hem önceki projemizdeki öğrendiklerimi pekiştirmemize yardımcı olurken hem de önceki projemizde öğrenmediğimiz bazı konular hakkında bilgisahibi olmamızı sağladı.

### 2.Teknolojik Katkıları:

Günümüzde meteoroloji alanında çok kullanılan meteoroloji istasyonu pek çok alanda bize kolaylık sağlamaktadır.günümüzde hava tahminleri gelişmiş meteorolojisi istasyonları tarafından yapılmaktadır. Atletizm, futbol ve golf turnuvalarını ilgilendiren hava tahminleri de bu istasyonlar tarafından yapılmaktadır. Gaz ve elektrik şirketlerin üretim, iletim ve planlamaları için soğuk ve sıcak havalarla ilgili bilgi verirler.Kısaca meteoroloji istasyonları bizim hayatımızı planlamamıza oldukça büyük bir katkı sağlarlar.Bizler de şimdiden olay ve basit programlanan robotlarımızla hayatı kolaylaştıran projeler geliştirerek kendimizi bu alanda ilerletmek ve ülkemize faydalı olmak istiyoruz.

### 3.Ekip Çalışması Katkıları:

Ekip çalışmasının bize kattıkları:

- İlk olarak hepimizin bu konudaki teknik bilgisi gelişti.Herkes yapabileceği görevleri üstlendi.Bu sayede de çalışmamız bir düzen ve bütünlük içinde tamamlandı.
- İşbirliği ve yardımlaşma içinde çalıştık ve bu çalışmamızın daha verimli oluşunu sağladı.
- Bu konu üzerinde hep birlikte uzun bir süre içinde çalıştık.Bu ilgilerimizin daha kalıcı olmasına katkı sağladı
- Karşılaştığımız sorunlarda birlikte çözüm aradık.Çözüm önerilerinde bulunduk.Grup olarak çalışınca sorunlar çok daha hızlı çözüm buldu.
- Birbirimizle iletişim alışanlığımız gelişti.
- Hatalarımızı gördüğümüzde bunları düzelttik.Bu çalışmamızın çok daha verimli olmasını sağladı.

### 4.Aksayan Yönler:

Projeyi gerçekleştirirken karşılaştığımız sorunlar:

### 5.Görüş ve Öneriler:

- Grup olarak iyi bir çalışma ortaya koyduğumuzu düşünüyor ve bu çalışmamızın bizim bu konudaki bilgimizi arttırdığını düşünüyoruz.
- Ortaya çıkan sorunlar bizi endişelendirdi ve nasıl çözüm bulabileceğimizi düşündük.Bu sorunlarla karşılaşmamız ise bize bir grup olarak sorunlarla başa çıkmayı öğretti.
- Edindiğimiz bu bilgiler sayesinde bir sonraki projede daha başarılı olacağımıza inanıyoruz.
- Bu projeyi internet üzerinden de değerleri görebilecek şekilde geliştirebiliriz.

## GİTHUB LİNKİ

<https://github.com/GizemBulut/sfl9eproje2.git>



