

T.C.
KOCAELİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ



Proje Adı : Sulama sistemi

Mehmet Baha Dursun

190223027

Ömer Halid Horat

190223052

İÇİNDEKİLER;

1) PROJENİN TANIMI.....	3
2) KULLANILAN MALZEMELER.....	3
2.1) Plastik Su Solenoid Vana	3
2.2) HC-05 Bluetooth Modülü	3
2.3) PIC16F887	4
2.4) 3.3V / 5V Breadboard Güç Kartı	4
2.5) DHT11 Isı ve Nem Sensörü Kart.....	5
2.6) HW 080 Toprak Nem Sensörü.....	5
2.7) PCF8574 I2C Giriş/Çıkış Çoklayıcı Modül	5
2.8) 2X16 LCD Ekran	6
2.9) LM35 Sıcaklık Sensörü.....	6
2.10) Pot, direnç, regülatör vb. (Elektronik Komponentler)	6
3) ÇÖZÜM YÖNTEMİ	6
4) AKIŞ DİYAGRAMI	7
5) KULLANILAN YÖNTEMLER	7
5.1) Matris Keypad.....	7
5.2) I2C LCD.....	8
6) MENÜLER.....	8
6.1) Ana Menü.....	8
6.2) Güncel Değer	8
6.3) Değer Ayarla	9
6.4) Sıcaklık.....	9
6.5) Nem.....	10
6.6) Olası Senaryolar.....	10
7) DEVRE ŞEMASI	11
8) KOD	12
9) KAYNAKÇA.....	33

1.) Projenin tanımı;

PIC mikroişlemci kullanarak kullanıcı girdisi ile ayarlanabilen sulama sistemi yapılacaktır. Sistem I2C ile kullanıcıya LCD üzerinden yakından kullanım veya bluetooth üzerinden UART kullanarak RoboRemo uygulaması aracılığı ile uzaktan kullanım sunabilecektir. 16x2 LCD ile kullanıcı menü üzerinden işlem yaparak istediği sıcaklık ve nem aralığını belirtebilecektir. Aynı işlemleri Android uygulaması üzerinden yapabilme opsiyonu da bulunmaktadır. Girilen parametreye göre devre; sensörlerden alınan analog değerleri istenilen değere getirmek için gerekli iletimleri sağlayacaktır. Sonuç olarak su pompasının kapağı açılacak ve istenilen nem değeri sağlanacak ya da bir ısıtma sistemi ile istenilen sıcaklık oluşturulacaktır.

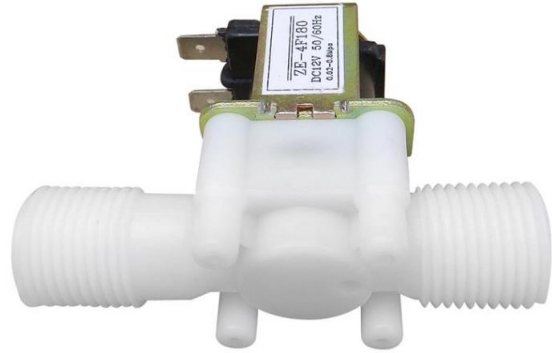
Kullanıcı istediği platform üzerinden girdilerini sağlayabilir.

2.) Kullanılan malzemeler;

2.1. Plastik Su Solenoid Vana

- Çalışma Gerilimi: 12 V 1A
- Çalışma Basıncı: 0.02 MPa – 0.08 MPa
- Valf Açılma: ≤ 0.15 sn
- Kapanma Süresi: ≤ 0.3 sn

Projemizde, verilen parametreler doğrultusunda suyun akışını sağlamak için kullandığımız vana.



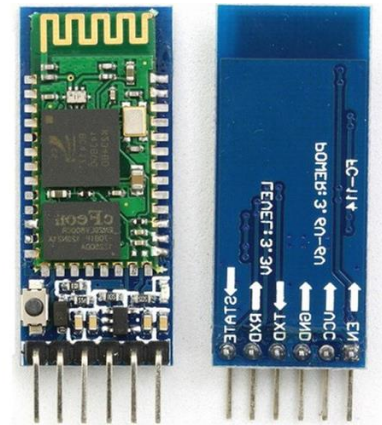
2.1.1.

2.2 HC-05 Bluetooth Modülü

- Bluetooth Protokolü: Bluetooth 2.0+EDR (Gelişmiş Veri Hızı)
- Çıkış Gücü: +4 dBm
- Çalışma Gerilimi: 1.8-5V

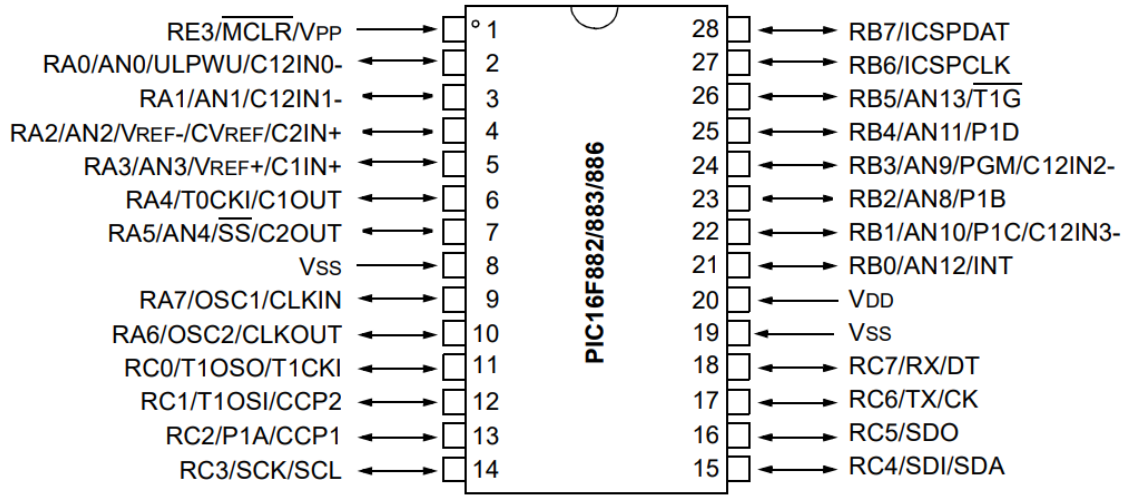
Android uygulamasından veri alıp göndermek için bluetooth bağlantısını sağlayan modül.

(Denenmiştir; çalıştırılmadığı için iptal edildi.)



2.2.1.

2.3 PIC16F887



2.3.1.

Projemizde mikroişlemci olarak PIC16F887 kullandık. Özellikleri;

- 35 kelimelik komut seti
- Çalışma frekansı: 0-20 Mhz
- Dahili osilatör
- 35 adet genel amaçlı I/O pini
- WDT
- Analog Karşılaştırıcı
- Gelişmiş USART modülü
- SPI ve I2C desteği
- 256 Bytes EEPROM hafıza
- 368 Bytes RAM hafıza
- 3 adet zamanlayıcı/sayıcı
- 8 KB FLASH hafıza
- A/D çevirici 14 kanal, 10 bit çözünürlük
- PWM çıkış özelliği
- Devre üzerinde programlama seçeneği bulunur.

The Data EEPROM and Flash program memory are readable and writable during normal operation (full VDD range). These memories are not directly mapped in the register file space. Instead, they are indirectly addressed through the Special Function Registers (SFRs). There are six SFRs used to access these memories:

- EECON1
- EECON2
- EEDAT
- EEDATH
- EEADR
- EEADRH (bit 4 on PIC16F886/PIC16F887 only)

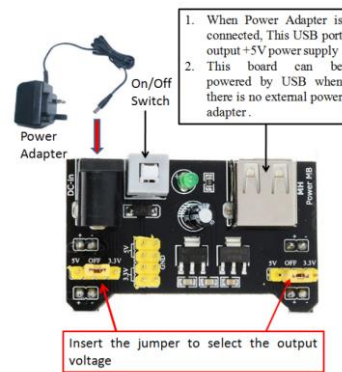
When interfacing the data memory block, EEDAT holds the 8-bit data for read/write, and EEADR holds the address of the EEDAT location being accessed. These devices have 256 bytes of data EEPROM with an address range from 0h to 0FFh.

When accessing the program memory block of the PIC16F886/PIC16F887 devices, the EEDAT and EEDATH registers form a 2-byte word that holds the 14-bit data for read/write, and the EEADR and EEADRH registers form a 2-byte word that holds the 12-bit address of the EEPROM location being read.

2.4 3.3V / 5V Breadboard Güç Kartı

- Giriş gerilimi: DC 6.5V – 12V
- Çıkış gerilimi: 3.3V – 5V
- Maksimum çıkış akımı: <700mA

Devremiz üzerinde karışıklık oluşturmaktan ihtiyacı duyduğumuz gerilimi sağlamak için kullandığımız güç kaynağı modülümüz.



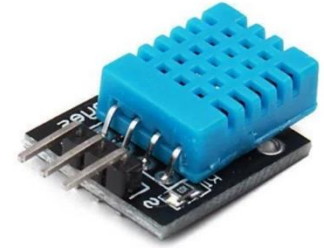
2.4.1.

2.5 DHT11 Isı ve Nem Sensörü Kart

- 8 bit mikroişlemci içerir
- 0- 50°C arasında 2°C hata payı ile sıcaklık ölçümü

Pin GND, 5 pini dijital sinyal çıkış ve ortadaki pin ise 5V gerilim pinidir. Devremizde sıcaklık değerini elde etmek için kullandık.

(Çalıştırılmadığı için LM35 kullanılmıştır.)

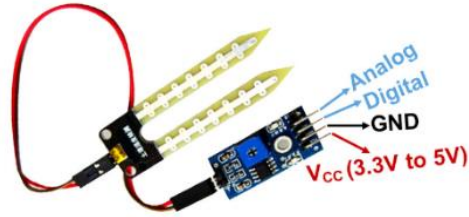
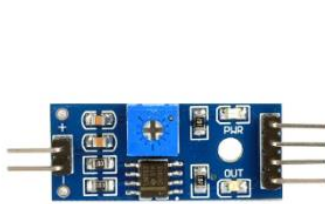


2.5.1

2.6 HW 080 Toprak Nem Sensörü

- Çalışma Gerilimi: 3.3V – 5V
- Çıkış gerilimi: 0- 4.2V
- Akım: 35mA
- Çıkış Türü: Dijital ve Analog

Toprak nem sensörü, toprağın içerisindeki nem miktarını veya ufak ölçekte bir sıvının seviyesini ölçmek için kullanabileceğiniz bir sensördür. Ölçüm yapılacak ortama nem ölçer problar batırılarak ölçüm yapılır. Toprağın veya içine batırılan sıvının meydana getirdiği dirençten dolayı prob uçları arasında bir gerilim farkı oluşur. Bu gerilim farkının büyüklüğüne göre de nem miktarı ölçülebilir. Topraktaki nem oranı arttıkça iletkenliği de artmaktadır.

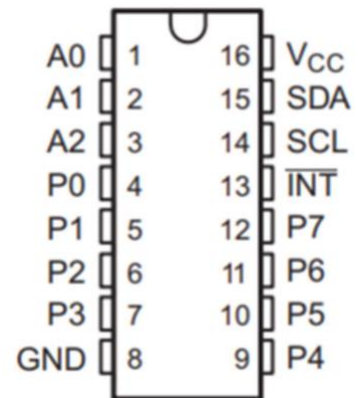


2.6.1.

2.7 PCF8574 I2C Giriş/Çıkış Çoklayıcı Modül

- Çalışma Gerilimi: 3.3V – 5V
- 80 mA'lık total alma kapasitesi
- 8-bit I/O Pini
- Düşük bekleme akımı
- Dahili çıkışlar doğrudan LED'leri kontrol edebilir.
- 100 kHz I2C-bus arayüzü

Mikrokontrolcü projemizde giriş/çıkış pin sayısını Arttırmak için kullandığımız modül.



2.7.1.

2.8 2X16 LCD Ekran

- Çalışma Gerilimi: 5V
- 16 Karakter kapasitesi x 2 Satır
- Built-in Control

INTERNAL PIN CONNECTION

Pin No.	Symbol	Level	Function
1	V _{SS}	—	0V
2	V _{DD}	—	+5V
3	V _O	—	—
4	RS	H/L	L: Instruction code input H: Data input
5	R/W	H/L	H: Data read (LCD module→MPU) L: Data write (LCD module←MPU)
6	E	H, H→L	Enable signal
7	DB0	H/L	Data bus line Note (1), Note (2)
8	DB1	H/L	
9	DB2	H/L	
10	DB3	H/L	
11	DB4	H/L	
12	DB5	H/L	
13	DB6	H/L	
14	DB7	H/L	

2.8.1.

MECHANICAL DATA (Nominal dimensions)

Module size	84W x 44H x 12D (max.) mm
Effective display area	61W x 15.8H mm
Character size (5 x 7 dots)	2.96W x 4.86H mm
Pitch	3.55 mm
Dot size	0.56W x 0.66H mm
Weight	about 25 g

ABSOLUTE MAXIMUM RATINGS

	min.	max.
Power supply for logic (V _{DD} –V _{SS})	0	7.0 V
Power supply for LCD drive (V _{DD} –V _O)	0	13.5 V
Input voltage (V _i)	V _{SS}	V _{DD} V
Operating temperature (T _a)	0	50°C
Storage temperature (T _{stg})	–20	70°C

ELECTRICAL CHARACTERISTICS

T_a = 25°C, V_{DD} = 5.0 V ± 0.25 V

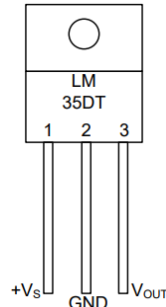
Input "high" voltage (V _{iH})	2.2 V min.
Input "low" voltage (V _{iL})	0.6 V max.
Output high voltage (V _{OH}) (–I _{OH} = 0.2 mA)	2.4 V min.
Output low voltage (V _{OL}) (I _{OL} = 1.2 mA)	0.4 V max.
Power supply current (I _{DD}) (V _{DD} = 5.0 V)	1.0 mA typ. 3.0 mA max.

Power supply for LCD drive (Recommended) (V_{DD}–V_O)
Du=1/16

at T _a = 0°C	4.6 V typ.
at T _a = 25°C	4.4 V typ.
at T _a = 50°C	4.2 V typ.

2.9 LM35 Sıcaklık Sensörü

- Analog sinyal çıkışı
- –55°C ile 150°C skala
- 10mV/K hassasiyet
- 5V besleme



2.9.1.

2.10 Pot, direnç, regülatör vb. (Elektronik Komponentler)

3.) Çözüm Yöntemi;

Projemiz 1 timer1, 1 WDT, 1 external kesme kullanılarak gerçekleştirilmiştir. Timer1 üzerinden 3 input kontrolü gerçekleştirilmiştir. Herhangi bir butona basıldığında Menu 0 1 2 #int_EXT kesmesi oluşarak menü durumlarında geçiş yapmaktadır. Diğer senaryolarda ise #int_EXT hafızası dolduğundan ötürü while(1) içinden RB0 butonunun basım şartı ile çağırılır. Menü modları switch-case yapıları ile sağlanmaktadır.

Karşılaştırılma işlemi yapıldıktan sonra, sistemin istenen değerlere getirilmesi sağlanacak ve bundan sonra timer1 ile düzenli olarak kontrol edilecektir. Sisteme x saniye input gelmezse şekil 3.1.'de görülen uyarıyı verir ve ardından sleep moduna girer. Sleep uyandırması için 2 farklı çözüm yöntemimiz vardır. Sleep de olduğu bilgisi Uyku_modu değişkeniyle hafızada tutulur. Sistemin ilk uyandırılması “OK” butonu ile sağlanır. #int_EXT'e gelen veri gerekli şartları sağladığında sleep modu O'lanır ve sleep modundan çıkılır. İkinci uyanması ise sistemin WDT ile uyandırılmasıdır. Bunun amacı sistemin enerji korunumudur. Uyandırıldıktan sonra **EEPROM**'dan gerekli bilgileri çekerek bahsedilen kontrolü gerçekleştirir ve gerekli ise sistemi tekrar istenen değerlere getirip uyur.

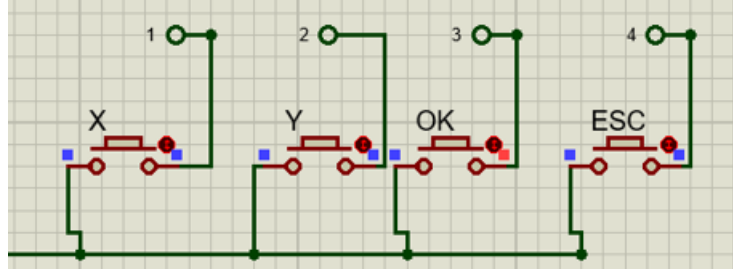


```

graph TD
    Start([Start]) --> I2C[I2C aktifleştirdi  
Değişkenler tanımlandı  
EEPROM bellekteki veriler  
okundu önceki açılıştan kalan  
önemli veriyi yazdırdı  
(Motor çalışma durumu son  
ister nem ve sıcaklık değeri  
aralığı)]
    I2C --> AnaMenu[Ana menu//Menu 0  
açıldı buton inputu  
bekleniyor]
    AnaMenu -- TRUE --> YButon[Y butonu senaryolar  
arası geçişi sağlar  
OK ise imlecin  
olduğu senaryoya  
girişi sağlar]
    AnaMenu -- FALSE --> Input[Input gelmedi  
sistem uyku  
moduna  
uyarsına girildi]
    YButon -- OK --> Case0[Case0 // Menu 1//  
Güncel menu  
Bu menu 2 diğer menüye  
daha acilir]
    YButon -- ESC --> Case1[Case1 //Menu 2//  
Değer ayarla  
Bu menu 2 diğer  
menüye daha acilir]
    Case0 --> Herhangi0[Herhangi bir butona  
basılmasız X saniye  
de bir uyarı sıcaklık  
ve nem değeri kontrol  
edilir.]
    Case1 --> OKButon[OK butonuna basılırsa  
sistem otomatik  
olarak(RB0) uyarır. Ve  
kalıcı menüden  
devam eder]
    Case1 --> NemAyar[Nem Ayarı  
Değer 1 ve Değer 2 seçeneği çıkar. %10'luk  
bir çözünürlüğü vardır (%10~%20~%30)  
Değer aralığı girilmezse sistem uyarı verir ve  
bir önceki menu'ye yollar girilen değerler  
kaydedilmez.  
değer 1 ya da değer 2 girilme ekranında ESC  
butonuna basılırsa yine değerler kaydedilmez  
buna dair bildirim verir ve bir önceki menüye  
geçer.]
    Case1 --> SıcaklikAyar[Sıcaklık Ayarı  
Değer 1 ve Değer 2 seçeneği çıkar. C olarak  
ayarlanır.  
Bu derece 15-25 C arasındır. Çözünürlüğü 1'dir.  
Değer aralığı girilmezse sistem uyarı verir ve bir  
önceki menu'ye yollar girilen değerler  
kaydedilmez.  
değer 1 ya da değer 2 girilme ekranında ESC  
butonuna basılırsa yine değerler kaydedilmez  
buna dair bildirim verir ve bir önceki menüye  
geçer.]
    Herhangi0 --> Fonksiyon0[Fonksiyon 1 eğer değer ayarları kısmından  
sıcaklık veya nem değeri girilirse. Sistem  
güncel nem veya sıcaklık ister değeri  
aralığında mı diye kontrol eder. İster değeri  
aralığında değil ise motorlar çalışır.  
Hangisi ister değeri aralığına ulaşmışsa o  
fonksiyonun motoru durdurulur ve  
sisteme ister değeri aralığına ulaşmıştır  
bilgisi gelir ve bu bilgi printlenir.]
    NemAyar --> Fonksiyon0
    SıcaklikAyar --> Fonksiyon0
    Fonksiyon0 --> Herhangi0

```

Menü ve işlem seçimleri matris keypad kullanılarak yapılmıştır. Timer1 ile matris keypadin 1,2,3,4 çıkışları sırayla toggledenmektedir. Bir butona basıldığında M çıkış işlemcinin B0 pini ile int_EXT kesmesini oluşturur ve o anda 1 2 3 4 çıkışlarından hangisinin toggled olduğuna bakılarak işlemler yapılır.



5.1.1.

5.2. I2C LCD

I2C haberleşme sistemi kullanılarak PCF8574 entegresiyle LCD kontrolü yapılmıştır. Böylece işlemciden 7 pin kullanmak yerine 2 pinle LCD kullanılmıştır.

6.) Menüler;

6.1. Ana Menü

Ana menü **“Güncel Değer”** ve **“Değer Ayarla”** menüleri arasında geçiş yapmamızı sağlar.



6.1.1.

6.2. Güncel Değer

Güncel Değer menüsü o sırada sistemin sensörden çektiği veriyi gösterir. Eğer istenilen sıcaklık ve nem değerleri daha önceden girilmişse ve kullanıcı bu menüye bakarken sistem istenilen değer aralığına getirilirse 6.2.2’de gösterildiği gibi bir bilgilendirme mesajı gösterilir. **“OK”** butonuna basılırsa ve önceden değer ayarlanmış ise hangi değer aralığını seçtiğinin hatırlatılması amaçlı sırasıyla girmiş olduğu sıcaklık ve nem değer aralığı gösterilir.

Kullanıcı **“Güncel Değer”** menüsünde olduğu sırada **“OK”** butonuna basarsa hatırlatıcı amaçlı daha önce hangi değer aralığının seçildiğini hatırlatan bir hatırlatıcı gösterir.



6.2.1



6.2.2

6.3. Değer Ayarla

Bu menüde kullanıcının karşısına ayarlamak için seçmesi amacı ile 2 seçenek çıkar. Kullanıcı bu sayede ayarlamak istediği parametreyi seçer.



6.3.1.

6.4. Sıcaklık

Kullanıcının karşısına sırası ile “Değer1” (6.4.1) ve “Değer2” (6.4.2.) ekranları çıkar ve bu ekranlarda kullanıcıdan gerekli istenen parametre girdileri alınır. 15 °C - 25 °C aralığında sıcaklık bilgisi girilebilir. 1 derece çözünürlüğe sahiptir.



6.4.1.



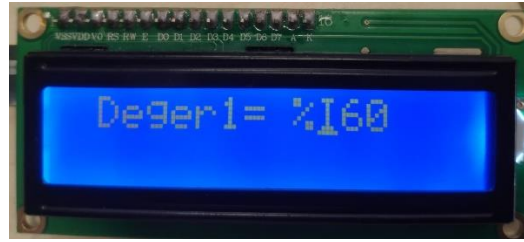
6.4.2.



6.4.3.

6.5. Nem

Kullanıcının karşısına sırası ile “Değer1” (6.5.1) ve “Değer2” (6.5.2.) ekranları çıkar ve bu ekranlarda kullanıcıdan gerekli istenen parametre girdileri alınır. 0 - %100 aralığında nem bilgisi girilebilir. 1 derece çözünürlüğe sahiptir. Çözünürlüğü %10’dur.



6.5.1



6.5.2



6.5.3

6.6. Olası Senaryolarda;

Eğer “Değer1” ve “Değer2” için girdi alınan ekranda “ESC” butonuna basılırsa “Değer Ayarla” ekranına geri dönülür.



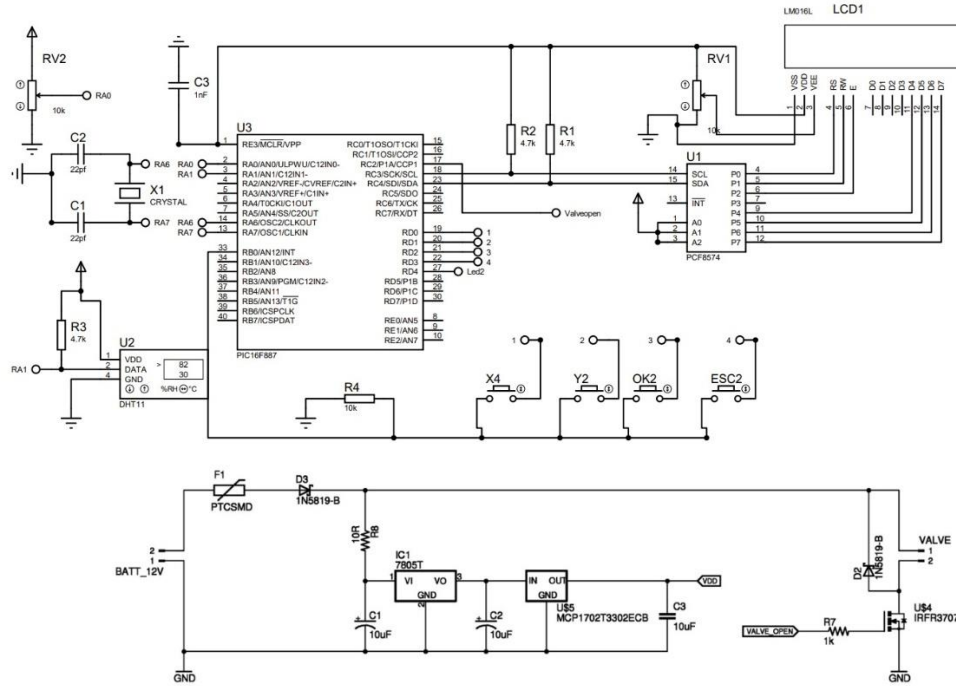
6.A.

Eğer “Değer1” ve “Değer2” aynı girilirse sistem değerleri kabul etmez ve “Değer Ayarla” ekranına geri dönlür.



6.B.

7.) Devre Şeması;



7.1.

8.) Kod;

```
#include <16f887.h>
#device ADC = 10
#use fast_io(A)
#use fast_io(B)
#use delay(clock=8M)
#use I2C(master, I2C1, FAST = 100000, STREAM = I2C_LCD,force_hw)

#priority timer1,timer2,ext
#include <I2C_LCD.c>

int8 i =0;
int8 X = 0;
int8 Y = 0;
int8 OK = 0;
int8 ESC =0;

int8 kontrol =0;

int8 Menu = 0;
int8 imlec_x = 1;
int8 imlec_y = 1;
int8 sayac = 0;

int8 Bir = 0;
int8 On = 5;

int8 ist_nem_on = 50;
int8 ist_nem_on1 = 0;
int8 ist_nem_on2 = 0;

int8 ist_sic = 15;
int8 ist_sic_1 = 0;
int8 ist_sic_2 = 0;

int8 veri_aktarimi=0;
int8 wdt_kesmesi = 0;

int8 olculen_nem_deger;
//int8 olculen_nem_deger=0;
int8 sicaklik_istenilen_deger_ulasti = 0;
int8 nem_istenilen_deger_ulasti = 0;
int8 gercek_olculen_sicaklik_degeri;

int8 olculen_sicaklik_deger;
//int1 olarak d?zenlenebilir -----
int8 sicaklik_girildi = 0;
```

```

int8 nem_girildi = 0;

int8 sicaklik_motoru = 0;
int8 su_motoru = 0;
int8 su_motoru_sayac;

int8 veri_girisi = 0;
int8 uyku_modu = 0;
int8 uyku_modu_yeri = 0;
// -----
int8 sleep_sayac = 0;
int8 son_veri_giris_zamani = 0;

unsigned long int bilgi; // iaretsiz 16 bitkil tam say
float voltaj;

int8 sic_bir[11] = {0x0F,0x10,0x11,0x12,0x13,0x14,0x15,0x16,0x17,0x18,0x19};
int8 nem_on[11] = {0x00,0x0A,0x14,0x1E,0x28,0x32,0x3C,0x46,0x50,0x5A,0x64};

void deger_gosteri();

void save_data()
{
    //write_eeprom(0xE7,i);
    //write_eeprom(0xE8,sayac);
    //write_eeprom(0xE9,On);
    // write_eeprom(0xEA,Menu);
    //write_eeprom(0xEB,X);
    //write_eeprom(0xEC,Y);
    write_eeprom(0xED,sicaklik_istenilen_deger_ulasti);
    write_eeprom(0xEE,nem_istenilen_deger_ulasti);
    //write_eeprom(0xEF,kontrol);
    // write_eeprom(0xF0,olculen_nem_deger);
    // write_eeprom(0xF1,gercek_olculen_sicaklik_degeri);
    write_eeprom(0xF2,ist_nem_on1);
    write_eeprom(0xF3,ist_nem_on2);
    write_eeprom(0xF4,ist_sic_1);
    write_eeprom(0xF5,ist_sic_2);
    write_eeprom(0xF6,nem_girildi);
    write_eeprom(0xF7,sicaklik_girildi);
    //write_eeprom(0xF8,wdt_kesmesi);
    write_eeprom(0xF9,su_motoru);
    write_eeprom(0xFA,sicaklik_motoru);
    //write_eeprom(0xFB,sleep_sayac);
    //write_eeprom(0xFC,uyku_modu);
    //write_eeprom(0xFD,uyku_modu_yeri);
    //write_eeprom(0xFE,veri_girisi);
    //write_eeprom(0xFF,Bir);

```

```

}

void imlec_temizle()
{
    LCD_Goto(imlec_x,imlec_y);
    LCD_Cmd(0x0C);
    LCD_Out(" ");
}

void imlec()
{
    LCD_Goto(imlec_x,imlec_y);
    LCD_Out("I");
    LCD_Cmd(0x10);
    LCD_Cmd(0x0F);
}

void su_motoru_kontrol()
{
    if (su_motoru == 1)
    {
        output_high(PIN_C2);
    }
    else
    output_low(PIN_C2);

    save_data();
}

void sicaklik_kontrolu()
{
    if ( sicaklik_motoru == 1)
    {
        output_high(PIN_D4);
    }
    else
    output_low(PIN_D4);

    save_data();
}

void olculen_nem_degeri()
{
    set_adc_channel(0);
    delay_us(20);
    bilgi = read_adc();
}

```

```

//giris degerimiz 5V 1023 deger okuyoruz o yzden hassasiyeti
// her %10 olarak baz aldığımızzdaan ttr 100 100 arttıracaız
bilgi = (1023 - bilgi); // veriyi alır iken su dikkatinde %100 su içinde
ise %30 a kadar alıyor bu sebepten ttr degeri tersliyorum
voltaj = bilgi/1023;
olculen_nem_deger = voltaj*100;

// voltaj = 0; 0 100
// bilgi=0;

if(nem_girildi ==1)
{
    if (ist_nem_on1 < olculen_nem_deger && olculen_nem_deger < ist_nem_on2)
    {
        delay_ms(300);
        su_motoru = 0;
        nem_girildi = 0;
        nem_istenilen_deger_ulasti = 1;

    }
    else
    {
        su_motoru = 1;
        nem_istenilen_deger_ulasti =0;

    }
    su_motoru_kontrol();
}
else
nem_girildi =1;
}

void olculen_sicaklik_degeri()
{
    set_adc_channel(1);
    delay_us(20);
    bilgi = Read_adc();

    voltaj = (0.0048828125*bilgi)*1000; // mV cinsinden
    olculen_sicaklik_deger = (voltaj/10); // her 10mv da 1 derece arttırma

    if(sicaklik_girildi==1)
    {

```



```

        if (ist_sic_1 < olculen_sicaklik_deger && olculen_sicaklik_deger <
ist_sic_2)
        {
            delay_ms(300);

            //deger_gosteri();
            sicaklik_girildi = 0;
            sicaklik_istenilen_deger_ulasti = 1;
        }
        else
        {
            sicaklik_motoru = 1;
            sicaklik_istenilen_deger_ulasti = 0;
        }
        sicaklik_kontrolu();
    }
    else
    sicaklik_girildi = 1;
}

void sleep_function()
{
    if (wdt_kesmesi ==0)
    {
        imlec_x=1;
        imlec_y=1;
        LCD_Cmd(0x01); //LCD temizleniyor
        delay_ms(10);
        uyku_modu = 1;
        LCD_Goto(1,1);
        LCD_Out(" Uyku Moduna");
        LCD_Goto(1,2);
        LCD_Out(" Aliniyor ");
        delay_ms(500);
        LCD_Cmd(0x01);
        imlec_temizle();
        Lcd_Goto(1,1);
        LCD_Out(" ");
    }
    output_high(PIN_D2);
    output_low(PIN_D0);
    output_low(PIN_D1);
    output_low(PIN_D3);

    save_data();
    wdt_kesmesi = 1;
}

```

```

    setup_wdt(WDT_ON);
    setup_wdt(WDT_2304MS|WDT_TIMES_4);    //~9,2 s reset

    sleep();

    restart_wdt();
    if(input(PIN_B0))
        wdt_kesmesi = 0;

    setup_wdt(WDT_OFF);
}

/*
void sleep_olcum()
{
    setup_wdt(WDT_ON);
    setup_wdt(WDT_2304MS|WDT_TIMES_4);    //~9,2 s reset

    restart_wdt();
    setup_wdt(WDT_OFF);
}
*/

void Main_menu()
{
    Menu = 0;
    Y = 0;
    OK=0;
    imlec_x=1;
    imlec_y=1;

    LCD_Cmd(0x01); //~LCD temizleniyor
    delay_ms(10);

    LCD_Goto(1,1);
    LCD_Out(" Guncel Deger");
    LCD_Goto(1,2);
    LCD_Out(" Deger Ayarla");
    imlec();
}

void Guncel_degerler_ok_senaryo()
{
    if(sicaklik_girildi == 1)
    {
        LCD_cmd(0x01);
        LCD_GOTO(1,1);
    }
}

```

```

        printf(LCD_Out,"%d - %d C arasi ",ist_sic_1,ist_sic_2);
        LCD_GOTO(1,2);
        LCD_Out("  ayarlaniyor ");
        delay_ms(1000);
    }
if (nem_girildi ==1)
{
    LCD_cmd(0x01);
    LCD_GOTO(1,1);
    printf(LCD_Out,"%c %d- %c %d arasi
",0b00100101,ist_nem_on1,0b00100101,ist_nem_on2);
    LCD_GOTO(1,2);
    LCD_Out("  ayarlaniyor ");
    delay_ms(1000);
    LCD_cmd(0x01);

}
}

void Guncel_degerler_menu()
{
    Menu = 1;
    Y = 0;
    imlec_temizle();
    //imlec_x = 1;
    //imlec_y = 1;
    //imlec();
    //LCD_Cmd(0x01);
    delay_ms(10);
    // nem ve sicaklik degerleri analog okunacak okunan degerler buraya
yazilacak
    olculen_nem_degeri();
    olculen_sicaklik_degeri();
    if(sicaklik_istenilen_deger_ulasti == 1)
    {
        lcd_cmd(0x01);
        LCD_Goto(1,1);
        LCD_Out(" Sicaklik ister");
        LCD_Goto(1,2);
        LCD_Out(" Deger araligi");
        sicaklik_istenilen_deger_ulasti =0;
        delay_ms(500);

    }
    if(nem_istenilen_deger_ulasti == 1)
    {
        lcd_cmd(0x01);
        LCD_Goto(1,1);
        LCD_Out(" Nem ister ");
    }
}

```

```

    LCD_Goto(1,2);
    LCD_Out(" Deger araligi");
    nem_istenilen_deger_ulasti =0;
    delay_ms(500);
    lcd_cmd(0x01);
}

    LCD_Goto(1,1);
    printf(LCD_Out,"  Nem = %c %d  ",0b00100101,olculen_nem_deger);
    LCD_Goto(1,2);
    printf(LCD_Out," Sicaklik= %d C",olculen_sicaklik_deger);

}

void Ayarlanacak_menu()
{
    Menu = 2;
    Y=0;
    LCD_Cmd(0x01);
    delay_ms(10);
    imlec_x = 1;
    imlec_y = 1;
    save_data();
    LCD_Out(" Sicaklik");
    LCD_Goto(1,2);
    LCD_Out(" Nem");
    imlec();
}

Void Sicaklik_ayari()
{
    Menu = 3;
    imlec_temizle();
    delay_ms(10);

    if(OK == 0)
    {
        imlec_x = 9;
        imlec_y = 1;
        LCD_GOTO(1,1);
        printf(LCD_Out," Deger1= %d C",ist_sic);
        ist_sic_1=ist_sic;
        imlec();
        veri_girisi = 1;
    }
}

```

```

else if (OK == 1)
{
    imlec_x = 9;
    imlec_y = 1;
    LCD_GOTO(1,1);
    printf(LCD_Out," Deger2= %d C",ist_sic);
    ist_sic_2=ist_sic;
    imlec();
    veri_girisi = 1;
}
else if(OK==2)
{
    veri_girisi = 1;
    imlec_temizle();
    LCD_Cmd(0x10);
    LCD_GOTO(1,1);

    if (ist_sic_1 < ist_sic_2)
    {
        LCD_GOTO(1,1);
        printf(LCD_Out,"%d - %d C arasi ",ist_sic_1,ist_sic_2);
        LCD_GOTO(1,2);
        LCD_Out(" ayarlaniyor ");
        delay_ms(1000);
        nem_girildi = 1;
        Ayarlanacak_menu();
        save_data();
    }

    else if (ist_sic_1 > ist_sic_2)
    {
        veri_aktarimi = ist_sic_2;
        ist_sic_2 = ist_sic_1;
        ist_sic_1 = veri_aktarimi;
        veri_aktarimi = 0;
        LCD_GOTO(1,1);
        printf(LCD_Out,"%d - %d C arasi ",ist_sic_1,ist_sic_2);
        LCD_GOTO(1,2);
        LCD_Out(" ayarlaniyor ");
        delay_ms(1000);
        nem_girildi = 1;
        Ayarlanacak_menu();
        save_data();
    }

    else
    {
        LCD_GOTO(1,1);
        LCD_OUT("Lutfen Deger ");
    }
}

```

```

        LCD_GOTO(1,2);
        LCD_OUT("Araligi giriniz");
        delay_ms(1000);
    }

    sicaklik_girildi = 1;
    delay_ms(1000);
    Ayarlanacak_menu();

}

}

void nem_ayari()
{
    Menu = 4;

    imlec_temizle();
    LCD_GOTO(1,1);
    if(OK == 0)
    {
        imlec_x = 11;
        imlec_y = 1;
        LCD_GOTO(1,1);
        printf(LCD_Out, " Deger1= %c %d ",0b00100101,ist_nem_on);
        ist_nem_on1=ist_nem_on;
        imlec();
        veri_girisi = 1;
    }

    else if (OK == 1)
    {
        imlec_x = 11;
        imlec_y = 1;
        LCD_GOTO(1,1);
        printf(LCD_Out, " Deger2= %c %d",0b00100101,ist_nem_on);
        ist_nem_on2=ist_nem_on;
        imlec();
        veri_girisi = 1;
    }
    else if(OK==2)
    {
        imlec_temizle();
        LCD_Cmd(0x10);
        veri_girisi = 1;
        if (ist_nem_on1 < ist_nem_on2)
        {
            LCD_GOTO(1,1);

```

```

        printf(LCD_Out,"%c %d- %c %d arasi
",0b00100101,ist_nem_on1,0b00100101,ist_nem_on2);
        LCD_GOTO(1,2);
        LCD_Out("  ayarlaniyor ");
        delay_ms(1000);
        nem_girildi = 1;
        Ayarlanacak_menu();
        save_data();
    }
    else if (ist_nem_on1 > ist_nem_on2)
    {
        veri_aktarimi = ist_nem_on2;
        ist_nem_on2 = ist_nem_on1;
        ist_nem_on1 = veri_aktarimi;
        veri_aktarimi = 0;
        LCD_GOTO(1,1);
        printf(LCD_Out,"%c %d- %c %d arasi
",0b00100101,ist_nem_on1,0b00100101,ist_nem_on2);
        LCD_GOTO(1,2);
        LCD_Out("  ayarlaniyor ");
        delay_ms(1000);
        nem_girildi = 1;
        Ayarlanacak_menu();
        save_data();
    }

    else
    {
        LCD_GOTO(1,1);
        LCD_OUT("Lutfen Deger ");
        LCD_GOTO(1,2);
        LCD_OUT("Araligi giriniz");
        delay_ms(1000);
    }
}

}

void senaryolar()
{
    switch (Menu)
    {
        case 0: //Main menu
            switch(Y%2)
            {
                case 0:
                    imlec_temizle();
                    imlec_x=1;
                    imlec_y=1;

```



```

    imlec();
    if(OK == 1)
    {
        LCD_CMD(0x01);
        Guncel_degerler_menu();
    }
    break;
case 1:
    imlec_temizle();
    imlec_x = 1;
    imlec_y = 2;
    imlec();
    if (OK == 1)
    {
        Ayarlanacak_menu();
    }
    break;
}
break;
case 1: //Güncel degerler menüsü

    imlec_temizle();
    imlec_x = 1;
    imlec_y = 2;
    imlec();

    if(ESC == 1 && uyku_modu_yeri == 0)
    {
        Main_menu();
    }

    else if(OK == 1 && uyku_modu_yeri == 0)
    {
        LCD_Cmd(0x01);
        delay_ms(10);
        imlec_temizle();
        imlec_x=1;
        imlec_y=1;
        LCD_Cmd(0x01);
        if(nem_girildi==1 || sicaklik_girildi ==1)
        Guncel_degerler_ok_senaryo();
        Guncel_degerler_menu();

    }
    break;

case 2: //Ayarlanacak menüsü
switch (Y%2)

```

```

{

case 0: //sicaklik ayari
imlec_temizle();
imlec_x = 1;
imlec_y = 1;
imlec();
if(ESC == 1 && uyku_modu_yeri == 0)
{
    Main_menu();
}
else if (OK == 1 && uyku_modu_yeri == 0)
{
    imlec_temizle();
    imlec_x = 1;
    imlec_y = 1;
    imlec();
    LCD_Cmd(0x01);
    delay_ms(10);
    imlec();
    OK=0;
    sicaklik_ayari();

}
break;

case 1: //nem ayari
imlec_temizle();
imlec_x = 1;
imlec_y = 2;
imlec();
if(ESC == 1 && uyku_modu_yeri == 0)
{
    Main_menu();
}
else if (OK == 1 && uyku_modu_yeri == 0)
{
    imlec_temizle();
    imlec_x = 1;
    imlec_y = 1;
    imlec();
    LCD_Cmd(0x01);
    delay_ms(10);
    imlec();
    OK=0;
    nem_ayari();

}
break;

```

```

    }
    break;
}
OK = 0;
ESC = 0;
veri_girisi = 1;
save_data();
}

void int_ext_ayar2()
{
    if (i == 2 && Bir != 0)
    {
        Bir--;
        veri_girisi = 0;
        son_veri_giris_zamani = 0;
    }
    else if (i == 1 && Bir != 10)
    {
        Bir++;
        veri_girisi = 0;
        son_veri_giris_zamani = 0;
    }
    else if (i == 4)
    {
        ESC=1;
        veri_girisi = 0;
        son_veri_giris_zamani = 0;
    }
    else if (i == 3)
    {
        OK++;
        veri_girisi = 0;
        son_veri_giris_zamani = 0;
        wdt_kesmesi = 0;
    }

    if (OK==1 && uyku_modu==1)
    {
        //veri_girisi=1;
        uyku_modu = 0;
        uyku_modu_yeri = 0;
        sicaklik_ayari();
    }

    if (ESC == 1)
    {

```

```

    veri_girisi = 1;
    ist_sic = 15;
    sicaklik_girildi = 0;

    lcd_cmd(0x01);
    lcd_GOTO(1,1);
    lcd_out("  Degerler  ");
    lcd_GOTO(1,2);
    lcd_out("  Kaydedilmedi  ");
    delay_ms(300);
    ist_sic_1= read_eeprom(0xF4);
    ist_sic_2= read_eeprom(0xF5);
    ESC = 0;
    Bir = 0;
    Ayarlanacak_menu();
}

else
{
    ist_sic= sic_bir[Bir];
    Sicaklik_ayari();
}
// save_data();
}
void int_ext_ayar3()
{
    if (i == 2 && ON != 0)
    {
        On--;
        veri_girisi =0;
        son_veri_giris_zamani = 0;
    }
    else if (i == 1 && ON != 10)
    {
        On++;
        veri_girisi =0;
        son_veri_giris_zamani = 0;
        wdt_kesmesi = 0;
    }
    else if(i == 4)
    {
        ESC=1;
        veri_girisi =0;
        son_veri_giris_zamani = 0;
    }
    else if(i == 3)
    {
        OK++;
    }
}

```

```

    veri_girisi =0;
    son_veri_giris_zamani = 0;
}

if(OK==1 && uyku_modu==1)
{
    //veri_girisi=1;
    uyku_modu = 0;
    uyku_modu_yeri = 0;
    nem_ayari();
}

if(ESC == 1)
{
    veri_girisi = 1;
    nem_girildi = 0;
    On = 6;
    ist_nem_on1= read_eeprom(0xF2);
    ist_nem_on2= read_eeprom(0xF3);
    lcd_cmd(0x01);
    lcd_GOTO(1,1);
    lcd_out("  Degerler  ");
    lcd_GOTO(1,2);
    lcd_out("  Kaydedilmedi  ");
    delay_ms(300);
    ESC=0;
    Ayarlanacak_menu();
}
else
{
    ist_nem_on = nem_on[On];
    nem_ayari();
}
// save_data();
}

#int_TIMER1
void sleep_time()
{
    // 1000 ms = 1 sn  10000 us = 10 ms
    // div_8 (osilatör 8mhz) 50ms*1/2*8 = 200 ms 200* 5 = 1 sn
    // 200*100 = 20 sn
    kontrol++;
    //Normalde hesaba göre 200*5 = 1 sn olmalıydır yani
    // her 1 de?i?imim 200 ms olmalıydır fakat di?er fonksiyonlarla
    //beraber bu de?er 540ms e kadar ????km???t???r Proteus kontrol ile
    // Kontrol 0-1'e ge?i?inin 540 ms s???rd???r g???r???lmektedir.
    if( su_motoru == 1)
    {

```

```

        su_motoru_sayac++;

        if (su_motoru_sayac<11) // kapak a?lma s?resi de dahil edilmi?ti 0.3
sn kapan? s?resi
        {
            su_motoru = 0;
            su_motoru_sayac = 0;
        }
    }

    if(kontrol == 5)
    {
        kontrol = 0;
        olculen_nem_degeri();
        olculen_sicaklik_degeri();
        if(Menu == 1 && uyku_modu_yeri == 0)
        guncel_degerler_menu();
    }

    if(veri_girisi == 1 && son_veri_giris_zamani<30)
    {
        son_veri_giris_zamani++;
        sleep_sayac = 0;
        if(uyku_modu_yeri==1)
        {
            LCD_CMD(0x01);
            uyku_modu_yeri = 0;
            if (Menu == 0)
            Main_menu();
            else if (Menu == 1)
            Guncel_degerler_menu();
            else if (Menu ==2 || Menu==3 || Menu==4)
            Ayarlanacak_menu();
        }
    }

    else if (veri_girisi == 0 && sleep_sayac <20)
    {
        sleep_sayac++;
    }

    else if (veri_girisi == 1)
    {
        son_veri_giris_zamani = 0;
        veri_girisi=0;
        uyku_modu_yeri = 1;
        imlec_temizle();
        imlec_x = 1;
        imlec_y = 1;
    }

```

```

        LCD_Cmd(0x01);
        LCD_GOTO(1,1);
        LCD_Out(" 10 sn sonra");
        LCD_GOTO(1,2);
        LCD_Out(" uyku modu aktif");
    }

    else
    {

        sleep_sayac=0;
        sleep_function();

    }

}
/*
#INT_TIMER2
void kontrol_et()
{
    clear_interrupt(INT_TIMER2);

}
*/
#int_EXT
void sorgulama(void)
{
    if( Menu == 0 || Menu == 1 || Menu ==2)
    {
        int8 sayac = 0;

        if(i == 4)
        {
            ESC=1;
            veri_girisi =0;
            son_veri_giris_zamani = 0;

        }
        else if(i == 3)
        {
            OK=1;
            veri_girisi =0;
            son_veri_giris_zamani = 0;
            wdt_kesmesi = 0;

        }
        else if(i == 2)

```



```

{
Y++;
veri_girisi =0;
son_veri_giris_zamani = 0;

}
else if(i == 1)
{
X++;
veri_girisi =0;
son_veri_giris_zamani = 0;

}
if(OK==1 && uyku_modu==1 && Menu == 0)
{
//veri_girisi=1;
uyku_modu = 0;
uyku_modu_yeri = 0;
Main_menu();
}
else if(OK==1 && uyku_modu==1 && Menu == 1)
{
//veri_girisi=1;
uyku_modu = 0;
uyku_modu_yeri = 0;
Guncel_degerler_menu();
}
if(OK==1 && uyku_modu==1 && Menu == 2)
{
//veri_girisi=1;
uyku_modu = 0;
uyku_modu_yeri = 0;
Ayarlanacak_menu();
}
if(X == 9)
X=1;
if(Y == 9)
Y=1;

senaryolar();

while(input(PIN_B0)&& sayac<250)
{
sayac++;
delay_ms(2);
}
}
// save_data();

```

```

}

void read_data()
{
    //i = read_eeprom(0xE7);
    //sayac = read_eeprom(0xE8);
    //On = read_eeprom(0xE9);
    //X = read_eeprom(0xEB);
    //Y = read_eeprom(0xEC);
    //Menu = read_eeprom(0xEA);
    sicaklik_istenilen_deger_ulasti = read_eeprom(0xED);
    nem_istenilen_deger_ulasti = read_eeprom(0xEE);
    //kontrol = read_eeprom(0xEF);
    //olculen_nem_deger= read_eeprom(0xF0);
    //gercek_olculen_sicaklik_degeri = read_eeprom(0xF1);
    ist_nem_on1= read_eeprom(0xF2);
    ist_nem_on2= read_eeprom(0xF3);
    ist_sic_1= read_eeprom(0xF4);
    ist_sic_2= read_eeprom(0xF5);
    nem_girildi= read_eeprom(0xF6);
    sicaklik_girildi= read_eeprom(0xF7);
    //wdt_kesmesi= read_eeprom(0xF8);
    su_motoru = read_eeprom(0xF9);
    sicaklik_motoru = read_eeprom(0xFA);
    //sleep_sayac = read_eeprom(0xFB);
    //uyku_modu = read_eeprom(0xFC);
    //uyku_modu_yeri = read_eeprom(0xFD);
    //veri_girisi = read_eeprom(0xFE);
    //Bir = read_eeprom(0xFF);
}

void main()
{
    LCD_Begin(0x4E);

    setup_wdt(WDT_1152MS|WDT_TIMES_16);      //~18,4 s reset
    setup_wdt(WDT_OFF);
    Main_menu();
    imlec();
    read_data();
    set_tris_a(0x03);

    //output_drive(PIN_B0);
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports(PIN_A0,PIN_A1);

    setup_timer_1(T1_INTERNAL | T1_DIV_BY_8);
    enable_interrupts(INT_EXT);

```

```

enable_interrupts(GLOBAL);
enable_interrupts(INT_TIMER1);
disable_interrupts(INT_TIMER2);
disable_interrupts(INT_CCP1);
disable_interrupts(INT_CCP2);

while(1)
{
    delay_ms(30);
    if( Menu == 3  && (input(Pin_B0) ==1))
    {
        int_ext_ayar2(); //nem_ayari
    }
    if( Menu == 4 && (input(PIN_B0) == 1))
    {
        int_ext_ayar3(); // sicaklik_ayari
    }

    if(wdt_kesmesi ==0)
    {
        output_low(PIN_D0);
        output_low(PIN_D1);
        output_low(PIN_D2);
        output_low(PIN_D3);
        i++;
        if(i==5)i=1;
        if(i==1)output_high(PIN_D0);
        else if(i==2)output_high(PIN_D1);
        else if(i==3)output_high(PIN_D2);
        else if(i==4)output_high(PIN_D3);
    }
    if(wdt_kesmesi ==1)
    {
        read_data();
        Guncel_degerler_menu();
        lcd_cmd(0x01);
        lcd_goto(1,1);
        save_data();
        delay_ms(500);
        sleep_function();
    }
}
}

```

9.) Kaynakça;

https://web.archive.org/web/20211203115438/https://www.nxp.com/docs/en/data-sheet/PCF8574_PCF8574A.pdf

<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>

<https://components101.com/modules/soil-moisture-sensor-module>

<https://pdf.direnc.net/upload/3-3v-5v-breadboard-guc-karti-datasheet.pdf>

<https://datasheetpdf.com/pdf-file/1462370/Hitachi/LM016L/>

https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf

<https://www.irf.com/product-info/datasheets/data/irfr3707.pdf>

<https://ww1.microchip.com/downloads/en/DeviceDoc/30277d.pdf>

<https://embedded-lab.com/blog/lab-17-sleep-and-wake-pic-microcontrollers/>

<https://simple-circuit.com/interfacing-pic16f877a-dht11-sensor-ccs-c/>

<https://www.theengineeringprojects.com/2017/08/dht11-arduino-interfacing.html>

<https://simple-circuit.com/pic16f887-dht11-sensor-ccs-c/>

<https://www.mikroe.com/ebooks/programming-dspic-mcus-in-c/introduction>