

Kocaeli Üniversitesi  
Mekatronik Mühendisliği

# Mikroişlemciler Dersi Proje Raporu

SD KART

İLYAS HÜSEYİN GÜVENÇ  
NO:170223025

## İçindekiler

1.Giriş.....	2
2. Kullanılan Komponentlerin Tanımı .....	2
2.1 Mikrodenetleyici Seçimi .....	2
2.2 FAT İmage Dosyası ve Gereksinimleri.....	2
2.3 Step Motorlar.....	3
2.4 GLCD ve Menü .....	4
3.Projenin Fotoğrafları ve Kod kısımlarının açıklaması .....	4
3.1 Fat dosya sisteminden bir G-code text dosyasını oluşturma .....	4
3.2 G-Code Text Dosyasını Diziye atmak ve Motorları Sürmek .....	8
4.KODUN TAMAMI.....	22
Kaynakça .....	40

# 1.Giriş

SD Kart projesinde FAT dosya sistemli bir sd kart içinden bir g-code text dosyasını okuyup en az 2 adet step motorun sürülmesi üzerine idi. İlk olarak FAT32 sistemini tercih ettim çünkü günümüzde bu sistem daha yaygın olarak kullanıyor ve kullanması kolay ve büyük hafızalı cihazlar tarafından destekleniyor.

Sırasıyla projenini kısımlarından bahsetmek istiyorum.

1. Bir FAT image dosyası üzerinden bir G-code text dosyasını okuma
2. Ardından bu text dosyasını işlemek için bir string dizisinin atma.
3. Bu diziyi en basit şekilde x ve y komutlarına göre ayrıştırma.
4. X ve Y değerlerine göre step motorları sürme
5. Motorlar sürülürken yapılan işlemlerin loglanması
6. GLCD üzerinden menü oluşturma
7. Motorlar sürülürken durdurma ve devam etme fonksiyonların eklenmesi
8. Diğer eklentilerin dahil edilmesi

## 2. Kullanılan Komponentlerin Tanımı

Bu bölümde projede neyi nasıl kullandığımı anlatacağım. İlk olarak FAT image dosyasından ve bunun için kullandığım kütüphanelerden bahsetmeliyim.

### 2.1 Mikrodenetleyici Seçimi

Projemde bana RAM ve ROM hafızalarının miktarı, portların sayısı, işlem hızı gibi faydalarından dolayı PIC18F4550 denetleyicisini kullanmaya karar verdim.

### 2.2 FAT İmage Dosyası ve Gereksinimleri

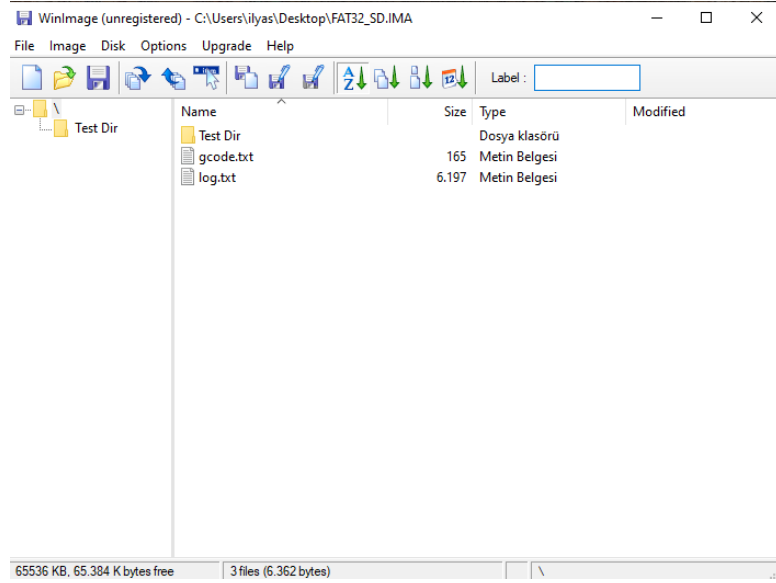
Öncelikle Proteus üzerinden bir simülasyon olacağı için Proteus'un kendi SD-CARD komponenti bir image dosyasını çalıştırıyor. Bunun için image dosya oluşturulabilir fakat ben bir kaynaktan onu indirdim.[1]

Ayrıca geliştirilmiş ve bazı hatalarından kurtarılmış olan iki kütüphaneyi yani mmcsd\_m.c ve fat\_m.c kütüphanelerini de kullandım. Onlarında linklerini ekliyorum.[2]

Proje de FAT İMAGE dosyasının bize çıkardığı birkaç sorun oldu. İlki, G – Code dosyasının kod text dosyasını kod ile oluşturabildim. Attığım dosyalara erişim söz konusu olmadı. Bunun nedenin simülasyon kaynaklı olduğunu düşünüyorum.

Sd kart İmage dosyasının boyutu 64Mb'dır. Dosyayı Pc'den açarken bir program kullanmamız gerekiyor. Ben Win İmage isimli programı kullandım. SD kart dosyasının görünümü aşağıdaki gibidir.

## Mikroişlemcileri Dersi Dönem Projesi



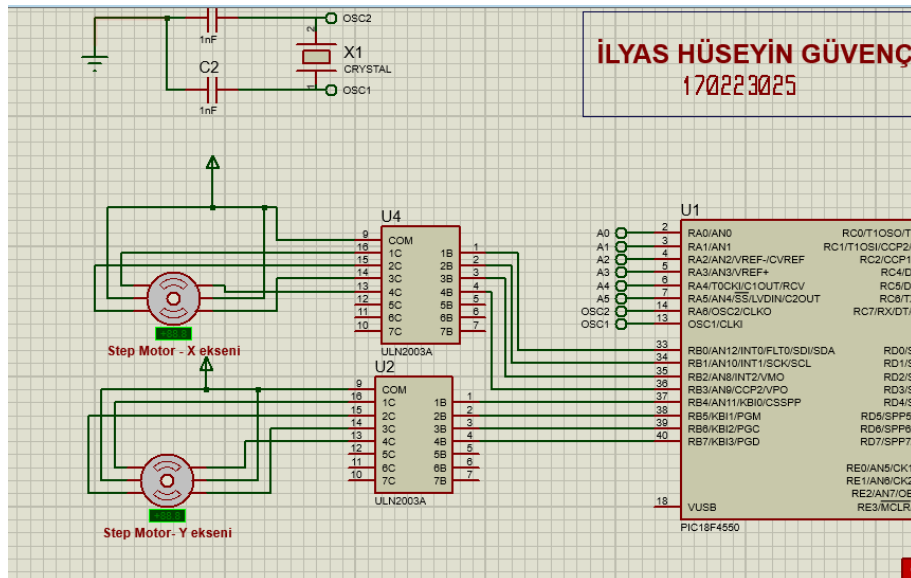
Dosyaların Directory(dir) kısmının tanımlanması için bir test dosyası oluşturmak zorunda kaldım aksi durumlarda program text dosyalarının içine okuma-yazma yapmaya izin vermedi.

### 2.3 Step Motorlar

İstenildiği üzere 2 adet step motoru sürme görevini gerçekleştirmek için ULN2003A entegresi üzerinden motorları sürerek gerçekleştirdim. Bunun için B portunu binary adımlar dizisi üzerinden kontrol ettim. (bk. BYTE const yarım\_adım\_x [ ], y [ ])

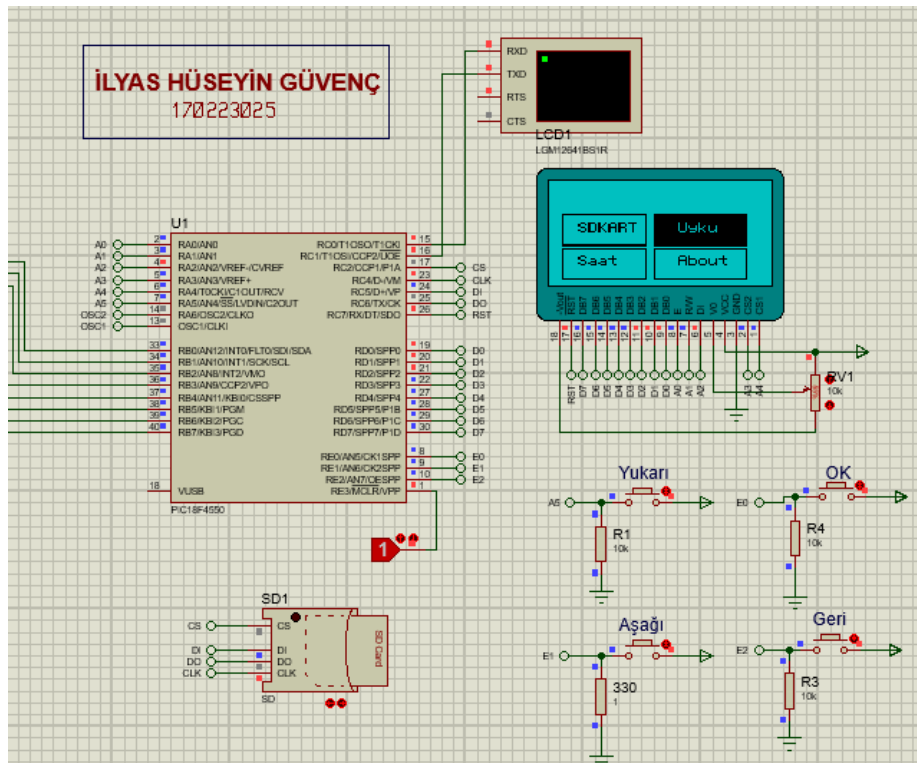
Projede 2 adet step motoru integer değerlerine göre sürme algoritması oluşturdum.

Burada şunu belirtmek istiyorum projedeki motor sürme fonksiyonu lineer interpolasyon yani G01 komutuna göre hareket ediyor. Açık söylemek gerekirse diğer kodları yapmaya kapasitem yetmedi lakin daha çok araştırma yaparak sağlam bir algoritma oluşturabileceğimi biliyorum.



## 2.4 GLCD ve Menü

GLCD seçmemin sebebi bazı özelliklerinin görünüm olarak daha açıklayıcı ve kullanımı basit olmasıdır.

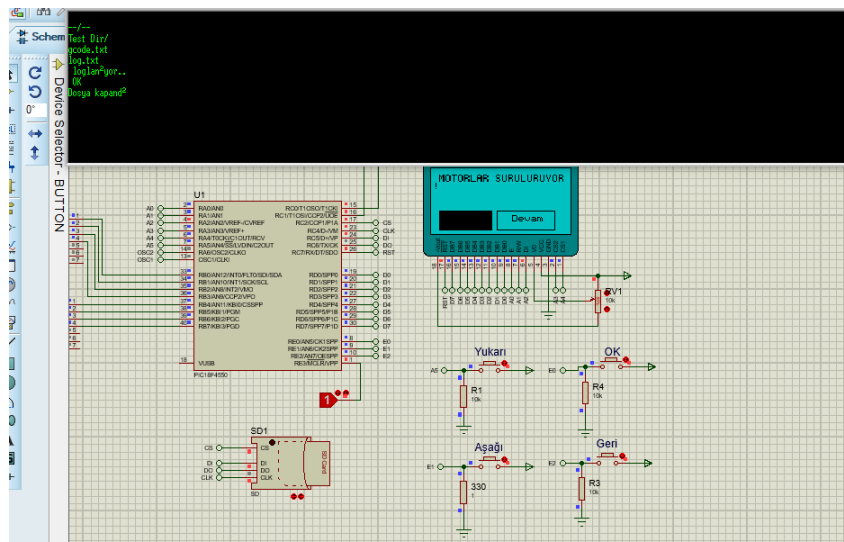


### 3.Projenin Fotoğrafları ve Kod kısımlarının açıklaması

Bu bölümde projenin ilk başta bahsettiğim kısımlarını anlatacağım. Ayrıca proje dosyaları ile birlikte bir adette simülasyon videosu oluşturacağım. Onun da Drive linkini proje dosyasına atıyorum.

### 3.1 Fat dosya sisteminden bir G-code text dosyasını oluşturma

Bunu oluşturmak için yukarda belirttiğim gibi text dosyası oluşturma dosyasını da kodda ilk olarak ben oluşturdum. Daha sonra onu comment olarak atadım.



Ekran görüntüsünde görüldüğü gibi simülasyonu daha rahat incelemek için bir Terminal’de kullandım. Terminalde görüldüğü gibi SD kartın içindeki dosya ve klasörler görülüyor ve işlemler tamamlandı mı tamamlanmadı mı onu da görebiliyoruz. Aynı zamanda motorlar sürülüyor ifadesinde de kodun okunduğunu gördük. Şimdi bu kısmın kodunu aşağıya kopyalıyorum. Daha rahat incelenebilir.

```
int8 ax,ay,i;

    unsigned int d=0;

    char logger1[]=" X konum: ";

    char logger2[]=" Y konum: \n";

// char saat[]={ 'Y konum' };

    glcd_init(ON);

    FILE mycode;

    FILE mylog;

    i=fat_init();

    delay_ms(200);

    if(i!= 0)

        { glcd_text57(5, 5, "DOSYA AÇILAMIYOR!", 1, ON);

        }

    glcd_text57(5, 5, "KART TIPI:", 1, ON);

    switch(g_card_type) {

        case MMC: glcd_text57(60, 5, "MMC", 1, ON);break;

        case SDSC: glcd_text57(65, 5, "SDSC", 1, ON);break;

        case SDHC: glcd_text57(65, 5, "SDHC", 1, ON);

    }
```

```
delay_ms(1000);

DisplayDirectory(g_CWD);
delay_ms(100);
/*if(mk_dir("/Test Dir/") == 0)
    printf("OK");
else
    printf("error creating folder");

    delay_ms(1000);
if(mk_file("/gcode.txt") == 0)
    printf("OK");
else
    printf("error creating file");

    if(mk_file("/log.txt") == 0)
        printf("OK");
    else
        printf("error creating file");

// if(fatopen("/log.txt", "w", &mylog) != 0);

    if(fatopen("/gcode.txt", "w", &mycode) != 0) // g code dosyasını oluşturma
    { printf("\n\n error opening file");}
else
    printf("\n OK");
    delay_ms(2000);
    printf("\n putting");
```

## Mikroişlemcileri Dersi Dönem Projesi

```
if(fatputs("%G00 X5 Y5 G01 X5 Y5 G01 X5 Y15 G02 X9 Y19 G01 X23 Y19 G01
X32 Y5 G01 X21 Y5 G01 X21 Y8 G03 X19 Y10 G01 X13 Y10 G03 X11 Y8 G01 X11 Y5
G01 X5 Y5 G01 X5 Y5 G28 X0 Y0%", &mycode) == 0){
    printf("OK");}
else
    printf("writing error");
delay_ms(2000);
// Now close the file
printf("\r\nClosing the file ... ");

if(fatclose(&mycode) == 0)
    printf("OK");
else
    printf("closing error"); */
    if(fatopen("/log.txt", "w", &mylog) != 0){
        printf("\r error");
    }
else
    printf("\r loglanıyor..");
    delay_ms(200);

if(fatopen("/gcode.txt", "r", &mycode) != 0)
    { printf("\n\n error opening file");}
else
    printf("\n\r OK");
    delay_ms(200);
    //fatprintf(&mycode);

while(fateof(&mycode)==NULL){ // text dosyasını bir diziye atıyoruz.
    dizi[d] = fatgetc(&mycode);
```

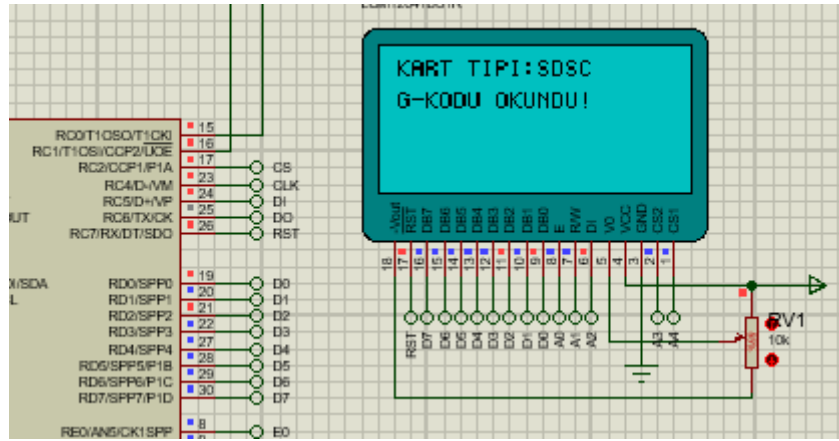


```

d++;
}
if(fatclose(&mycode) == 0)
    printf("\rDosya kapandı");
else
    printf("\r Kapatma hatası");

```

Baştan itibaren açıklayacak olursak ilk olarak tanımlamaları yaptım sonra SD kart fonksiyonunu çağırıp kart tipini ekrana yazdırdım.



Ve sonrasında G kodu doğrulama komutu koydum bu genel olarak bir işaret üzerinden sağlanır. G-code kontrol işareti ise text dosyasının başına ve sonuna koyulan “%” işaretidir.

```

if((dizi[0] == '%'))
    glcd_text57(5, 20, "G-KODU OKUNDU!", 1, ON);
delay_ms(2000);

```

Onun kodu da bu şekildedir. Okuyup glcd’ye yazdırıyoruz.

### 3.2 G-Code Text Dosyasını Diziye atmak ve Motorları Sürmek

G-code’u SD Karttan okuyup şu kodlar ile atıyoruz. Burada amaç dosyanın sonuna kadar okuyup onu sırasıyla bir diziye atmak. Bunu *fat\_m.c* kütüphanesindeki *fateof()* fonksiyonu ile gerçekleştirdim.

```
while(fateof(&mycode)==NULL){ // text dosyasını bir diziye atıyoruz.
dizi[d] = fatgetc(&mycode);

d++;
}
if(fatclose(&mycode) == 0)
printf("\rDosya kapandı");
else
printf("\r Kapatma hatası");
```

Görüldüğü gibi bir dosyayı önce bir açma izni vererek açıyoruz burada 'r' iznini kullanmamız gerekiyor. Sonra dosyadan gerekli metni okuyup diziye attık. Ve sonra dosyayı kapattık. Dosya kapanmazsa kod crash olabilir.

Daha sonra bu diziyi aşağıdaki gibi kodlarla motor sürem fonksiyonuna gönderiyoruz. Buradan önce motorsur() fonksiyonunu açıklamam gerekiyor.

```
void motorsur(unsigned int8 x, unsigned int8 y){
    if(((x>=konum_x) && (y>konum_y))|| ((x>konum_x) && (y>=konum_y))){ //X'in
konumu büyükse yeni x'ten bu döngüyü yap.
        for(int k=0;k=((x-konum_x)+(y-konum_y));k++ )
        {
            //Gelen konum farklıysa döngüye gir.
            delay_ms(100);
            if(x!=konum_x) {
                output_b(yarim_adim_x[in_x]);
                delay_ms(hiz);
                if(in_x>=7)
                {in_x= -1;}
                in_x++;
                konum_x++; } // X yeni konum

            if(y!=konum_y)
            { output_b(yarim_adim_y[in_y]);
                delay_ms(hiz);
```

```
if(in_y>=7)
    {in_y= -1;}

    in_y++;
    konum_y++;} // Y yeni konum

}

printf("\r1.Suanki in_x:%d in_y:%d",in_x, in_y);
printf("\r1.Suanki X konum:%d, Y:%d",konum_x, konum_y);delay_ms(500);
}

else if(((x>=konum_x) && (y <konum_y)) || ((x>konum_x) && (y<=konum_y))){
    for(int k=0;k=((x-konum_x)+(konum_y-y));k++ )
    {
        delay_ms(100);
        if(x!=konum_x) {

            output_b(yarim_adim_x[in_x]);
            delay_ms(hiz);

            if(in_x>=7)
                {in_x= -1;}

            in_x++;
            konum_x++; } // X yeni konum
        if(konum_y!=y){
            if(in_y<=0)
                { in_y=8;}

            in_y--;
            output_b(yarim_adim_y[in_y]);
```

```

        delay_ms(hiz);

        konum_y-- ; }// yeni konum

    }

    printf("\r2.Suanki in_x:%d in_y:%d",in_x, in_y);
    printf("\r2.Suanki X konum:%d Y Konum:%d",konum_x, konum_y);delay_ms(500);
}

else if(((x<=konum_x) && (y >konum_y)) || ((x<konum_x) && (y>=konum_y)) ){
//X'in konumu büyükse yeni x'ten bu döngüyü yap.
    for(int k=0;k=((konum_x-x)+(y-konum_y));k++ )
    {
        //Gelen konum farklıysa döngüye gir.
        delay_ms(100);
        if(konum_x!=x){
            if(in_x<=0)
                {in_x=8;}
            in_x--;
            output_b(yarim_adim_x[in_x]);
            delay_ms(hiz);
            konum_x-- ; }// yeni konum
        if(y!=konum_y){
            output_b(yarim_adim_y[in_y]);
            delay_ms(hiz);
            if(in_y>=7)
                { in_y= -1;}
            in_y++;
            konum_y++;} // yen}i konum
        }

    printf("\r3.Suanki in_x:%d in_y:%d",in_x, in_y);
    printf("\r3.Suanki X konum:%d Y Konum:%d",konum_x, konum_y);delay_ms(500);
}

else if(((x<konum_x) && (y<=konum_y)) || ((x<=konum_x) && (y <konum_y)) ){

```

```
for(int k=0;k=((konum_x-x)+(konum_y-y));k++ )
{
    delay_ms(100);
    if(konum_x!=x){

        if(in_x<=0)
            {in_x=8;}

            in_x--;
        output_b(yarim_adim_x[in_x]);
        delay_ms(hiz);

        konum_x-- ; // yeni konum
    if(konum_y!=y){

        if(in_y<=0)
            { in_y=8;}

            in_y--;
        output_b(yarim_adim_y[in_y]);
        delay_ms(hiz);

        konum_y-- ; // yeni konum
    }

    printf("\r4.Suanki in_x:%d in_y:%d",in_x, in_y);

    printf("\r4.Suanki X konum:%d Y Konum:%d",konum_x, konum_y); delay_ms(500);
}
}
```

Bu kodu kısaca şöyle açıklayabilirim:

1. İlk olarak fonksiyonlar bize (x,y) koordinatları şeklinde geliyor.
2. Bunları 4 duruma göre inceliyoruz.
3. Konum\_ x ve konum\_y değişkenleri önceki konumları gösteriyor.
4. Bu durumlar:
  - a. ((x>=konum\_x) && (y>konum\_y))|| ((x>konum\_x) && (y>=konum\_y))
    - i. Gelen koordinatların ikisinde şu anki konumdan büyük ve eşit olabilir
  - b. ((x>=konum\_x) && (y <konum\_y)) || ((x>konum\_x) && (y<=konum\_y))

- i. X gelen koordinatı büyük şu anki konumda, Y gelen koordinatı şu anki konumdan küçük olabilir
  - c.  $((x \leq \text{konum\_x}) \&\& (y > \text{konum\_y})) \parallel ((x < \text{konum\_x}) \&\& (y \geq \text{konum\_y}))$ 
    - i. Y gelen koordinatı büyük şu anki konumda, X gelen koordinatı şu anki konumdan küçük olabilir
  - d.  $((x < \text{konum\_x}) \&\& (y \leq \text{konum\_y})) \parallel ((x \leq \text{konum\_x}) \&\& (y < \text{konum\_y}))$ 
    - i. Gelen iki koordinat değeride küçük olabilir.
5. Sonra bu koşullardan geçce değerleri for döngüsüne attım burada da önemli olan şey for döngüsünün ne kadar döngü oluşturacağı bunu konum ile gelen x değerlerinin farkından hesaplattırmayı düşündüm ve başarılı bir şekilde çalıştı.

Not: Bu komutlar tam sayı değerlerine göre yazılmıştır. Float değerlere göre bu kod çalışmaz onun için farklı algoritmalar oluşturulmalı.

Şimdi de dizinin X ve Y koordinat değerlerinin ayrıştırmasını inceleyelim. Bunu için switch ,case ve dizideki eleman sayısı kadar dönen bir while döngüsü ile yaptım kodu aşağıdaki gibidir. Ayrıca kodun içinde loglama ve menü kodları da vardır onları da devamında açıklayacağım.

git1:

```
glcd_init(ON);  
glcd_text57(5, 5, "MOTRLAR SURULURUYOR!", 1, ON);  
//restart_wdt();
```

git:

```
glcd_text57(15,47,menu2, 1, ON);  
glcd_rect(60,42,115,62,NO,ON);  
glcd_text57(75,47,menu3, 1, ON);
```

```
if(k1==1){
```

```
glcd_rect(5,42,55,62,YES,ON);  
glcd_text57(75,25,menu2, 1, OFF);  
}
```

```
if(k1==2){
```

```
glcd_rect(60,42,115,62,YES,ON);
```

```
glcd_text57(75,47, menu3, 1, OFF);
```

```
}
```

```
while(j!=d){
```

```
do{
```

```
if(input(geri)==1) break;
```

```
if(input(ok)==1) {
```

```
if(k1==1){
```

```
// durdurma komutu
```

```
output_b(0x00);
```

```
glcd_init(ON);
```

```
glcd_text57(5, 15, "MOTORLAR DURDURULDU!", 1, ON);
```

```
delay_ms(500);
```

```
devam=1;
```

```
goto git;
```

```
}
```

```
if(k1==2){ // devam etme komutu.
```

```
devam=0;
```

```
goto git1;
```

```
}
```

```
}
```

```
if(input(yukari)==1){
```

```
k1++;
```

```
delay_ms(100);
if(k1>2) k1=1;
delay_ms(100);
goto git;
}
if(input(asagi)==1){
k1--;
delay_ms(100);
if(k1<1) k1=2;
delay_ms(100);
goto git;
} } while(devam==1);
switch(dizi[j]){
case 88:
    if(dizi[j+3]==32) //X elemanı okunduğu zaman 2 basamaklı ise
ilk koşul tek basamaklı ise 2 koşul devreye girecek
        { ax= 10*(dizi[j+1]-48)+(dizi[j+2]-48);
        if(fatputs(logger1, &mylog) == 0)
            fatputs(dizi[j+1], &mylog);
            fatputs(dizi[j+2], &mylog);} // dizinin o an ki elemanından '0' değerini
çıkarıyoruz. 2 basamak lı sayı işlemi yapıyoruz
    else
        { ax=(dizi[j+1]-48);
        if(fatputs(logger1, &mylog) == 0)
            fatputs(dizi[j+1], &mylog);}
        break;
case 89:
    if(dizi[j+3]==32) //X elemanı okunduğu zaman 2 basamaklı ise ilk
koşul tek basamaklı ise 2 koşul devreye girecek
        { ay= 10*(dizi[j+1]-48)+(dizi[j+2]-48);
        if(fatputs(logger2, &mylog) == 0)
            fatputs(dizi[j+1], &mylog);
```



```
fatputs(dizi[j+2], &mylog);} // dizinin o an ki elemanından '0' değerini  
çıkartıyoruz. 2 basamaklı sayı işlemi yapıyoruz
```

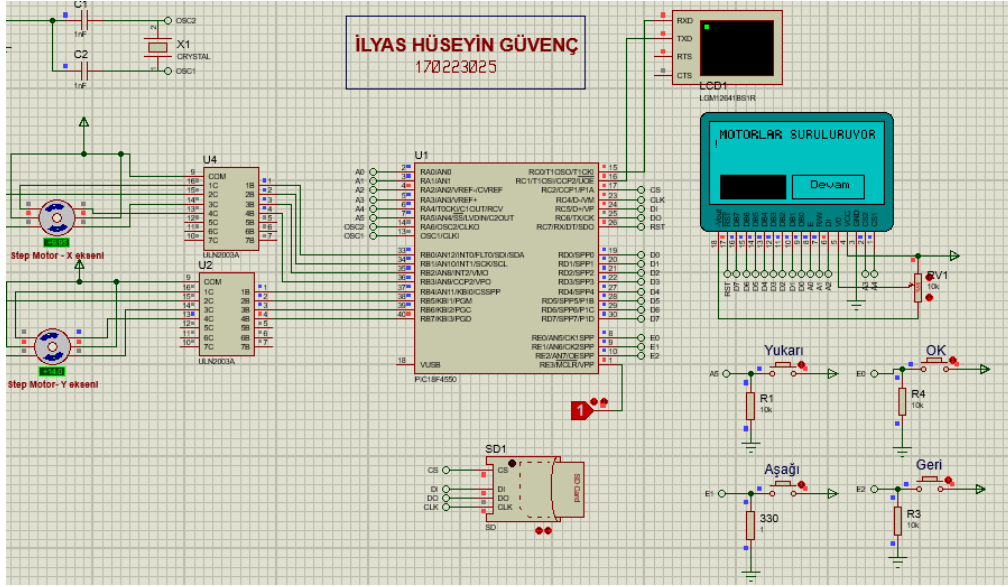
```
else  
{ay=(dizi[j+1]-48);  
if(fatputs(logger2, &mylog) == 0)  
fatputs(dizi[j+1], &mylog);}  
motorsur(ax,ay);  
break;  
}  
j++;  
}  
if(fatclose(&mylog) == 0)  
printf("\rDosya kapandı");  
else  
printf("\r Kapatma hatası");  
  
}
```

Burada önce d değerini çektiğimiz yeri hatırlayalım *fateof()* kısmında ki döngüden almıştık.

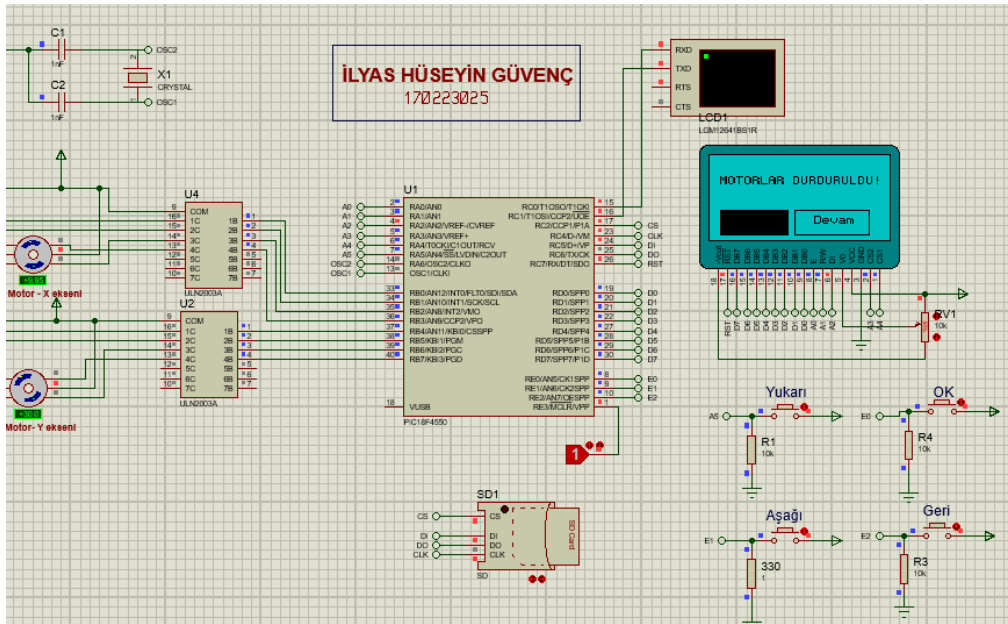
Burada case'leri oluşturduktan sonra onların dizideki yerine göre integer değerlerine atadım. Bunun sebebi karmaşık kodlara girmemektir. Koşul yapısı kullanarak X veya Y den sonra gelen değerlerin 2 mi yoksa tek basamaklı mı olduğuna göre işlem yaparak ax ve ay değişkenlerine attım sonra bunların motorsur() fonksiyonuna gönderdim bu şekilde iki motor aynı anda çalıştı ve gerçek zamanlı olarak işlem yapıldı.

Menu kodlarına gelecek olursak burada ilk while girince do ile her seferinde menüden durdurma veya devam etme komutu seçilmiş mi bunu kontrol ediyoruz. Sonra devam değişkeninin seçim yapıp yapılmadığına göre değeri belli oluyor eğer devam=1 ise do-while döngüsü dönecek ve X, Y komutu çalışmayacak eğer devam seçilirse devam = 0 olacak ve do döngüsünden çıkıp kaldığı yerden devam edecek.

## Mikroişlecileri Dersi Dönem Projesi

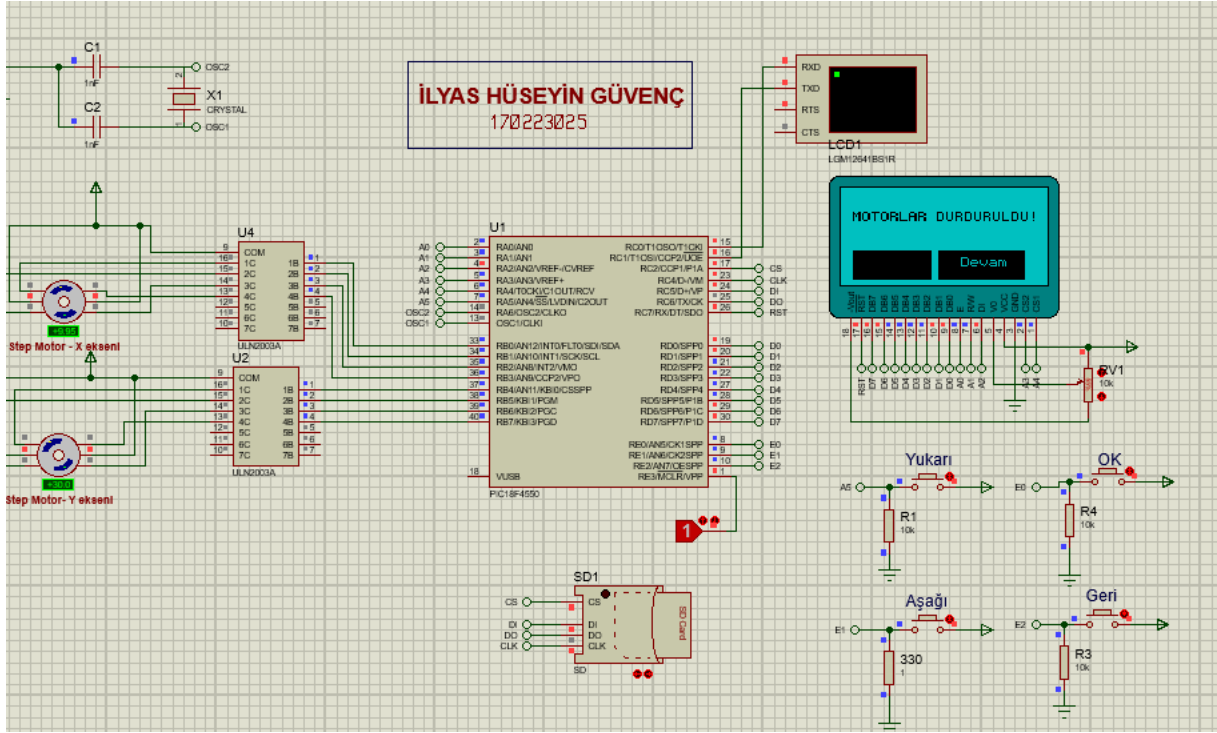


Motorlar sürülüyor bu sırada ok tuşuna basıp durduruyoruz.(bir hatadan dolayı durdur yazısı gözükmiyor.Kodda problem olmadığı halde)

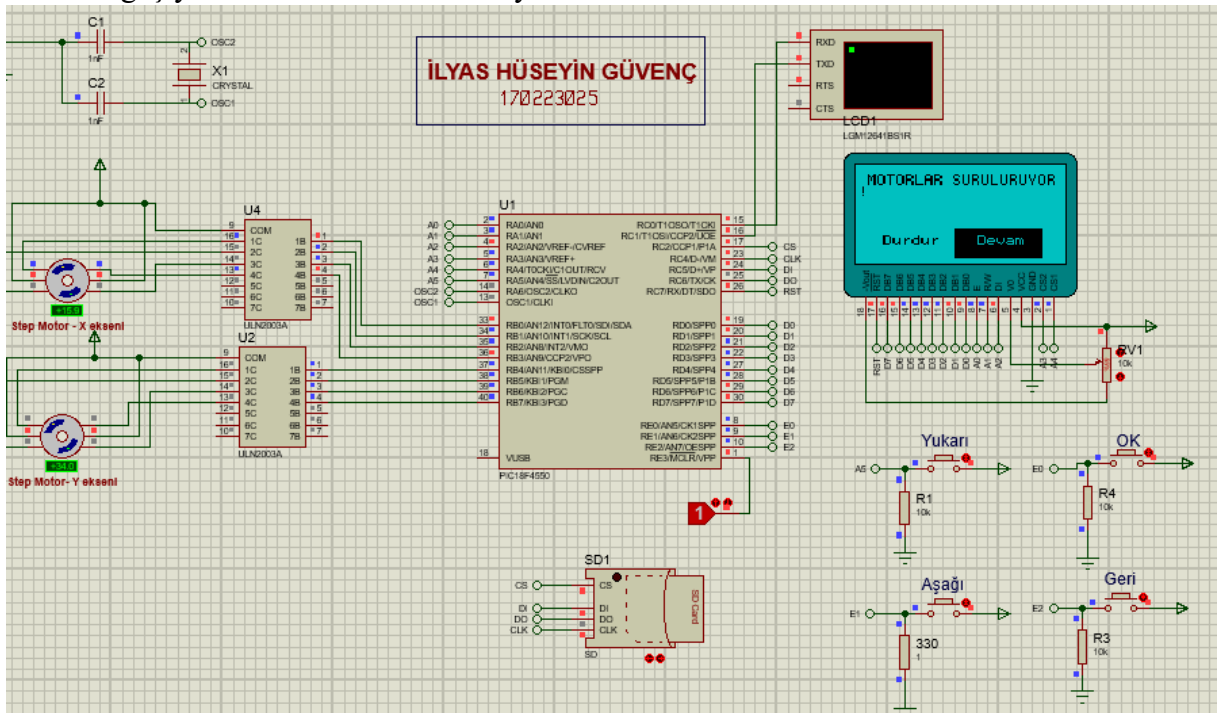


Durduruldu yazısı çıkıyor ve devam veya geri çıkılmasını bekliyor.

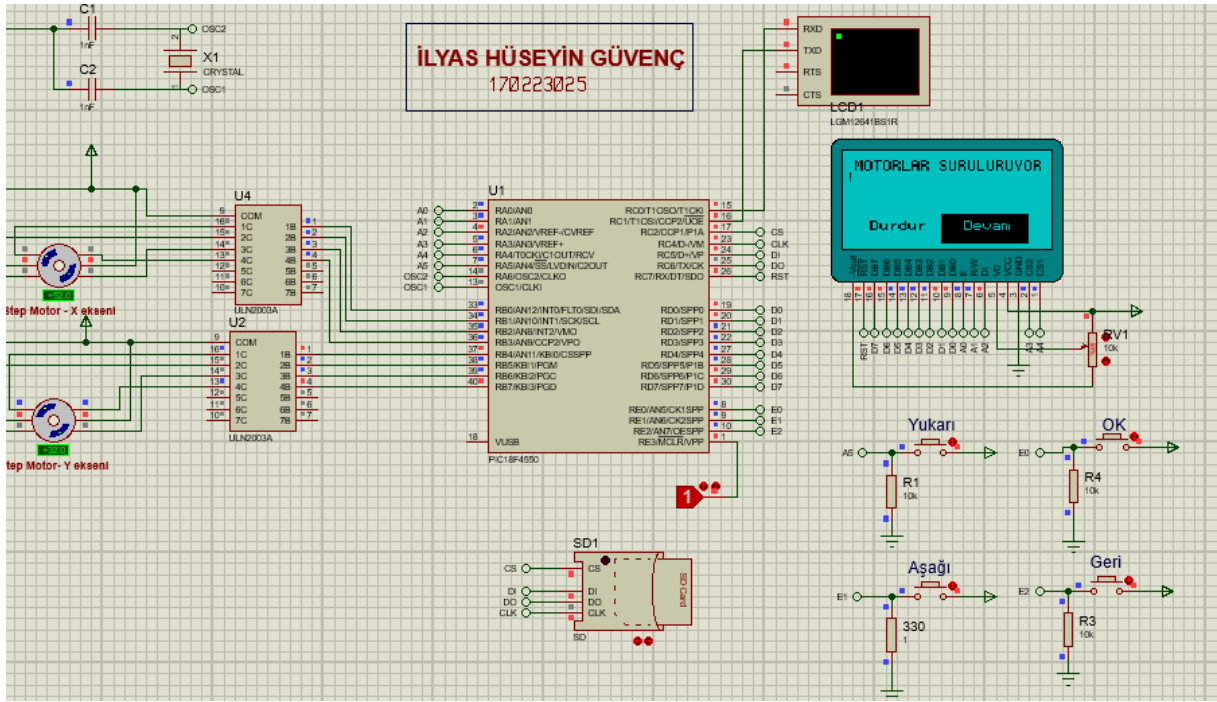
## Mikroişlecileri Dersi Dönem Projesi



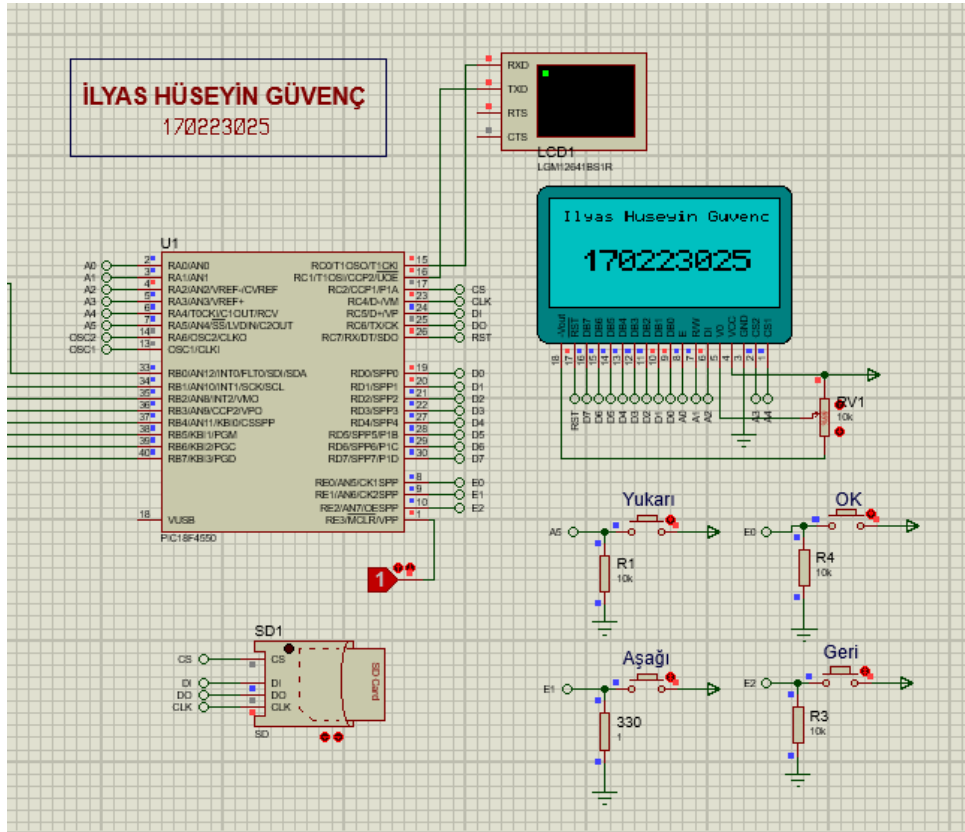
Devama geçiyoruz ve ok butonunu basıyoruz.



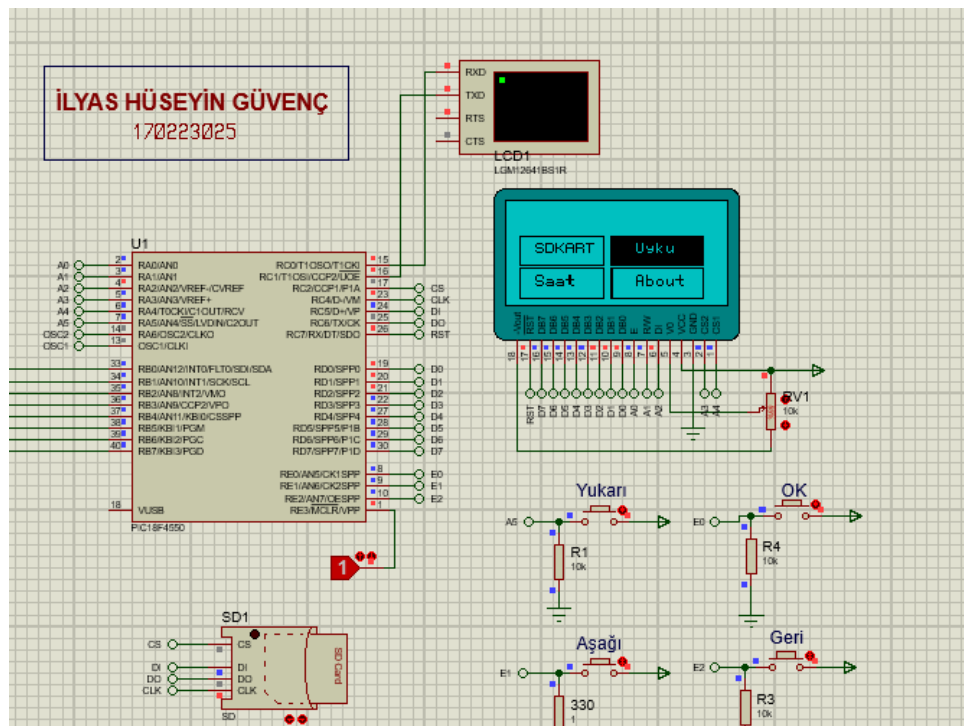
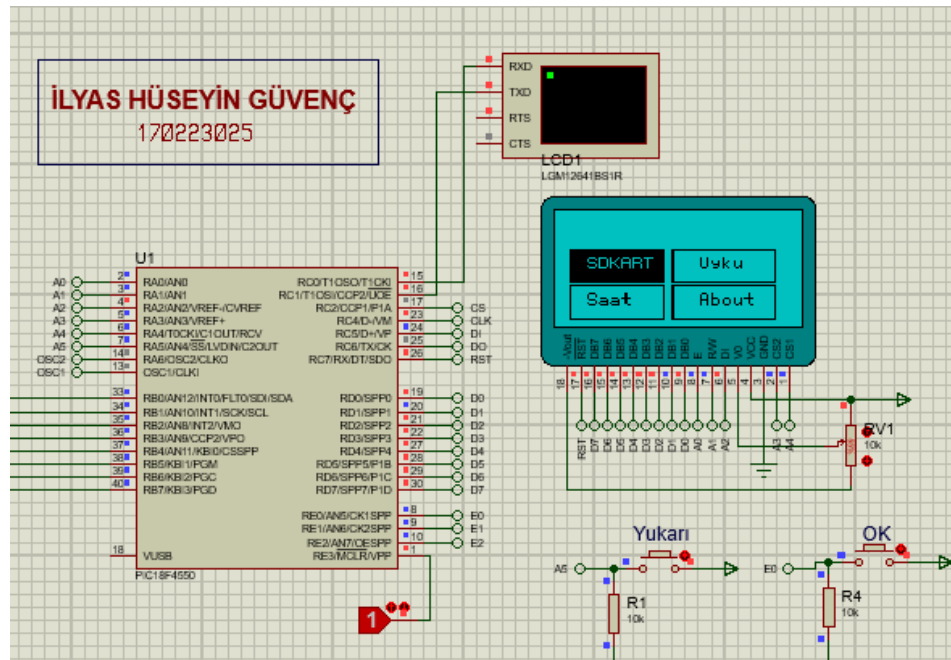
## Mikroişlecileri Dersi Dönem Projesi



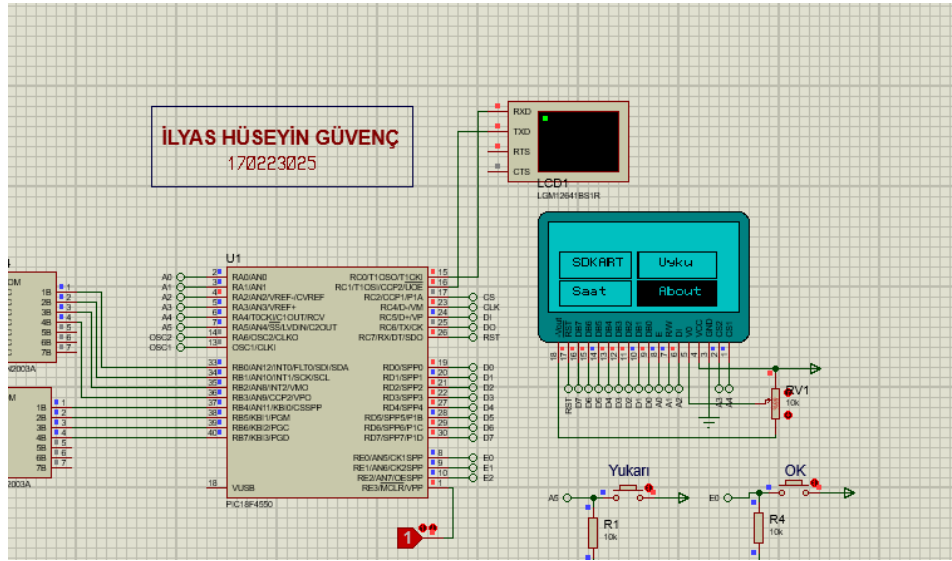
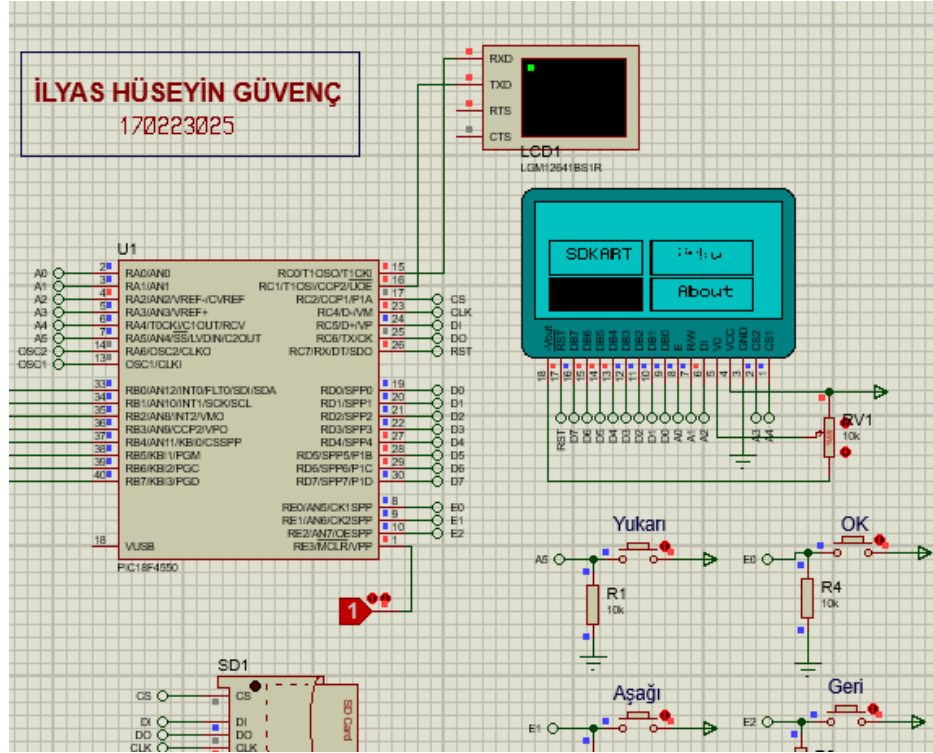
Menüde serbest gezinme:



## Mikroişlemcileri Dersi Dönem Projesi



## Mikroişlecileri Dersi Dönem Projesi



## 4.KODUN TAMAMI

.c Dosyasından incelemenizi tavsiye ederim.

```
#define MMCSD_PIN_SELECT PIN_C2
#define MMCSD_PIN_SCK    PIN_C4
#define MMCSD_PIN_MOSI   PIN_C5
#define MMCSD_PIN_MISO   PIN_C6
#include <18F4550.h>

#define PASS_STRINGS = IN_RAM
#define HSPLL PLL2 CPUDIV1
#define use delay(clock = 48MHz)
#define use rs232(baud=9600, xmit=PIN_C0, rcv=PIN_C1) // daha hızlı kontrol için consol
kullanalım.
#define use fast_io(D)
#define use fast_io(B)
#define use fast_io(E)

#define yukari PIN_A5
#define asagi PIN_E1
#define geri PIN_E2
#define ok PIN_E0

#include <stdio.h>
#include "HDM64GS12.c"
#include <graphics.c>
#include <mmcsd_m.c>

// Include FAT library source file
#include <fat_m.c>
```

```
BYTE const yarim_adim_x[] =  
{0b00000001,0b00000011,0b00000010,0b00000110,0b00000100,0b00001100,0b00001000,0  
b00001001};
```

```
BYTE const yarim_adim_y[] =  
{0b00010000,0b00110000,0b00100000,0b01100000,0b01000000,0b11000000,0b10000000,0  
b10010000};
```

```
int8 in_x=0, in_y=0, hiz=95, j=0;    // index tanımlaması   ve adımlar arası hız değeri  
  
unsigned int8 konum_x=0,konum_y=0; // konum değişkeni tanımlıyoruz. Motorun nerde  
olduğunu bilmek için  
  
unsigned char dizi[255];    // Gcode dosyasının içindeki verileri bir diziye atmak için  
  
char g_CWD[10] = "/";
```

```
/* GLCD TANIMLAMALARI*/
```

```
char menu1[]="SDKART";  
char menu2[]="Durdur";  
char menu3[]="Devam";  
char menu4[]="Uyku";  
char menu5[]="Saat";  
char menu6[]="About";
```

```
int1 devam=0; // motorların devamını sağlamak için ilgili komutu bekletme döngüsünü  
çalıştırma değeri. 0 ise devam edecek 1 ise ordaki while'ı döndürecek. satır:417 de.
```

```
int8 k=1,sec;
```



```
int k1=1;
```

```
//!int Ana_Menu();
```

```
//!void motorsur();
```

```
//!void about();
```

```
//!void sd_kart();
```

```
void DisplayDirectory(char *dir) // sd kartın içindekileri gösteren fonksiyon
```

```
{
```

```
    disp_folder_contents(dir);
```

```
}
```

```
int Ana_Menu(){
```

```
    basla:
```

```
        glcd_init(ON); // Ekran siliniyor
```

```
        glcd_rect(5,20,55,40,NO,ON);
```

```
        glcd_text57(15,25,menu1, 1, ON);
```

```
        glcd_rect(60,20,115,40,NO,ON);
```

```
        glcd_text57(75,25,menu4, 1, ON);
```

```
        glcd_rect(5,42,55,62,NO,ON);
```

```
        glcd_text57(15,47,menu5, 1, ON);
```

```
        glcd_rect(60,42,115,62,NO,ON);
```

```
        glcd_text57(75,47,menu6, 1, ON);
```

```
    if(k==1){
```

```
        glcd_rect(5,20,55,40,YES,ON);
```

```
        glcd_text57(15,25,menu1, 1, OFF);
```

```
}
```

```
if(k==2){
```

```
    glcd_rect(60,20,115,40,YES,ON);  
    glcd_text57(75,25,menu4, 1, OFF);  
}
```

```
if(k==3){
```

```
    glcd_rect(5,42,55,62,YES, ON);  
    glcd_text57(75,25,menu5, 1, OFF);  
}
```

```
if(k==4){
```

```
    glcd_rect(60,42,115,62,YES,ON);  
    glcd_text57(75,47, menu6, 1, OFF);  
}
```

```
while(TRUE){
```

```
    if(input(geri)==1) break;
```

```
    if(input(ok)==1) {  
        return k;}  
}
```

```
if(input(yukari)==1){
```

```
    k++;
```

```
delay_ms(100);
if(k>4) k=1;
delay_ms(100);
goto basla;
}
if(input(asagi)==1){

    k--;
    delay_ms(100);
    if(k<1) k=4;
    delay_ms(100);
    goto basla;
}
}
```

```
int uyku=0;
#INT_EXT
void uyku_modu(){
```

```
    uyku++;
    if(uyku==2)
        uyku=0;

}
```

```
int sayac=0;
#int_timer0
void timer0_kesme ()
{
```

```
set_timer0(60);  
sayac++;  
if (sayac==10) {  
    if(uyku==1) { sleep(); }  
}
```

```
void about(){  
    glcd_init(ON);  
    glcd_text57(10,5,"Mikro. Dersi", 1, ON);  
    glcd_text57(10,15,"HAZIRLAYAN", 1, ON);  
    glcd_text57(5,25,"ILYAS HUSEYIN GUVENC", 1, ON);  
    glcd_text57(10,35,"170223025", 1, ON);  
    while (TRUE){  
        if(input(geri)==1)  
            break;}  
}
```

```
void motorsur(unsigned int8 x, unsigned int8 y){
```

```
    if(((x>=konum_x) && (y>konum_y))|| ((x>konum_x) && (y>=konum_y)) ){ //X'in  
    konumu büyükse yeni x'ten bu döngüyü yap.
```

```
    for(int k=0;k=((x-konum_x)+(y-konum_y));k++ )
```

```
    { //Gelen konum farklıysa döngüye gir.
```

```
        delay_ms(100);
```

```
        if(x!=konum_x) {
```

```
            output_b(yarim_adim_x[in_x]);
```

```
            delay_ms(hiz);
```

```
            if(in_x>=7)
```

```
{in_x= -1;}

in_x++;

konum_x++; } // X yeni konum


if(y!=konum_y)
{ output_b(yarim_adim_y[in_y]);
  delay_ms(hiz);

if(in_y>=7)
  {in_y= -1;}

  in_y++;
  konum_y++;} // Y yeni konum

}

printf("\r1.Suanki in_x:%d in_y:%d",in_x, in_y);
printf("\r1.Suanki X konum:%d, Y:%d",konum_x, konum_y);delay_ms(500);
}

else if(((x>=konum_x) && (y <konum_y)) || ((x>konum_x) && (y<=konum_y))){

for(int k=0;k=((x-konum_x)+(konum_y-y));k++ )
{
  delay_ms(100);
  if(x!=konum_x) {

    output_b(yarim_adim_x[in_x]);
    delay_ms(hiz);

    if(in_x>=7)
      {in_x= -1;}
```

```
    in_x++;
    konum_x++; } // X yeni konum

if(konum_y!=y){

    if(in_y<=0)
    { in_y=8;}
    in_y--;
    output_b(yarim_adim_y[in_y]);
    delay_ms(hiz);
    konum_y-- ; }// yeni konum

}

printf("\r2.Suanki in_x:%d in_y:%d",in_x, in_y);
printf("\r2.Suanki X konum:%d Y Konum:%d",konum_x, konum_y);delay_ms(500);
}

else if(((x<=konum_x) && (y >konum_y)) || ((x<konum_x) && (y>=konum_y)) ){ //X'in
konumu büyükse yeni x'ten bu döngüyü yap.

    for(int k=0;k=((konum_x-x)+(y-konum_y));k++ )
    { //Gelen konum farklıysa döngüye gir.
        delay_ms(100);
        if(konum_x!=x){

            if(in_x<=0)
            {in_x=8;}

            in_x--;
```

```
output_b(yarim_adim_x[in_x]);
delay_ms(hiz);
konum_x-- ; }// yeni konum

if(y!=konum_y){

output_b(yarim_adim_y[in_y]);
delay_ms(hiz);
if(in_y>=7)
{ in_y= -1;}

in_y++;
konum_y++;} // yeni konum
}
printf("\r3.Suanki in_x:%d in_y:%d",in_x, in_y);
printf("\r3.Suanki X konum:%d Y Konum:%d",konum_x, konum_y);delay_ms(500);
}
else if(((x<konum_x) && (y<=konum_y)) || ((x<=konum_x) && (y <konum_y)) ){

for(int k=0;k=((konum_x-x)+(konum_y-y));k++ )
{
delay_ms(100);
if(konum_x!=x){

if(in_x<=0)
{ in_x=8;}
in_x--;
output_b(yarim_adim_x[in_x]);
delay_ms(hiz);
```

```
konum_x-- ; }// yeni konum

if(konum_y!=y){

    if(in_y<=0)
        { in_y=8;}
        in_y--;
    output_b(yarim_adim_y[in_y]);
    delay_ms(hiz);
    konum_y-- ; }// yeni konum
}

printf("\r4.Suanki in_x:%d in_y:%d",in_x, in_y);
printf("\r4.Suanki X konum:%d Y Konum:%d",konum_x, konum_y); delay_ms(500);
}

}

void sd_kart(){
    int8 ax,ay,i;
    unsigned int d=0;
    char logger1[]=" X konum: ";

    char logger2[]=" Y konum: \n";

    // char saat[]={ 'Y konum' };
    glcd_init(ON);
```



```
FILE mycode;
FILE mylog;

i=fat_init();

delay_ms(200);
if(i!= 0)
{ glcd_text57(5, 5, "DOSYA AÇILAMIYOR!", 1, ON);

}

glcd_text57(5, 5, "KART TIPI:", 1, ON);
switch(g_card_type) {
    case MMC: glcd_text57(60, 5, "MMC", 1, ON);break;
    case SDSC: glcd_text57(65, 5, "SDSC", 1, ON);break;
    case SDHC: glcd_text57(65, 5, "SDHC", 1, ON);
}
delay_ms(1000);

DisplayDirectory(g_CWD);
delay_ms(100);
    /*if(mk_dir("/Test Dir/") == 0)
        printf("OK");
    else
        printf("error creating folder");

    delay_ms(1000);
    if(mk_file("/gcode.txt") == 0)
```

```
printf("OK");
else
printf("error creating file");

if(mk_file("/log.txt") == 0)
printf("OK");
else
printf("error creating file");

// if(fatopen("/log.txt", "w", &mylog) != 0);

if(fatopen("/gcode.txt", "w", &mycode) != 0) // g code dosyasını oluşturma
{ printf("\n\n error opening file");}
else
printf("\n OK");
delay_ms(2000);
printf("\n putting");

if(fatputs("%G00 X5 Y5 G01 X5 Y5 G01 X5 Y15 G02 X9 Y19 G01 X23 Y19 G01
X32 Y5 G01 X21 Y5 G01 X21 Y8 G03 X19 Y10 G01 X13 Y10 G03 X11 Y8 G01 X11 Y5
G01 X5 Y5 G01 X5 Y5 G28 X0 Y0%", &mycode) == 0){
printf("OK");}
else
printf("writing error");
delay_ms(2000);
// Now close the file
printf("\r\nClosing the file ... ");

if(fatclose(&mycode) == 0)
printf("OK");
else
```

```
printf("closing error"); */
if(fatopen("/log.txt", "w", &mylog) != 0){
printf("\r error");
}
else
    printf("\r loglanıyor..");
    delay_ms(200);

if(fatopen("/gcode.txt", "r", &mycode) != 0)
    { printf("\n\n error opening file");}
else
    printf("\n\r OK");
    delay_ms(200);
    //fatprintf(&mycode);

while(fatfeof(&mycode)==NULL){
dizi[d] = fatgetc(&mycode);

d++;
}
if(fatclose(&mycode) == 0)
    printf("\rDosya kapandı");
else
    printf("\r Kapatma hatası");

if((dizi[0] == '%'))
```

```
    glcd_text57(5, 20, "G-KODU OKUNDU!", 1, ON);
delay_ms(2000);
git1:
glcd_init(ON);
glcd_text57(5, 5, "MOTORLAR SURULURUYOR!", 1, ON);
//restart_wdt();

git:
glcd_text57(15,47,menu2, 1, ON);
glcd_rect(60,42,115,62,NO,ON);
glcd_text57(75,47,menu3, 1, ON);

    if(k1==1){

glcd_rect(5,42,55,62,YES,ON);
glcd_text57(75,25,menu2, 1, OFF);
    }
    if(k1==2){

glcd_rect(60,42,115,62,YES,ON);
glcd_text57(75,47, menu3, 1, OFF);
    }

    while(j!=d){

do{

    if(input(geri)==1) break;

    if(input(ok)==1) {
```

```
if(k1==1){
    // durdurma komutu
    output_b(0x00);
    glcd_init(ON);
    glcd_text57(5, 15, "MOTORLAR DURDURULDU!", 1, ON);
    delay_ms(500);
    devam=1;
    goto git;

}

if(k1==2){ // devam etme komutu.
    devam=0;
    goto git1;
}

}

if(input(yukari)==1){

    k1++;
    delay_ms(100);
    if(k1>2) k1=1;
    delay_ms(100);
    goto git;
}

if(input(asagi)==1){

    k1--;
    delay_ms(100);
```

```
if(k1<1) k1=2;
delay_ms(100);
goto git;
}}while(devam==1);

switch(dizi[j]){

case 88:

    if(dizi[j+3]==32) //X elemanı okunduğu zaman 2 basamaklı ise
ilk koşul tek basamaklı ise 2 koşul devreye girecek
        {ax= 10*(dizi[j+1]-48)+(dizi[j+2]-48);
        if(fatputs(logger1, &mylog) == 0)
            fatputs(dizi[j+1], &mylog);
            fatputs(dizi[j+2], &mylog);} // dizinin o an ki elemanından '0' değerini
çıkarıyoruz. 2 basamak lı sayı işlemi yapıyoruz
        else
            {ax=(dizi[j+1]-48);

            if(fatputs(logger1, &mylog) == 0)
                fatputs(dizi[j+1], &mylog);}
            break;
case 89:

    if(dizi[j+3]==32) //X elemanı okunduğu zaman 2 basamaklı ise ilk
koşul tek basamaklı ise 2 koşul devreye girecek
        {ay= 10*(dizi[j+1]-48)+(dizi[j+2]-48);
        if(fatputs(logger2, &mylog) == 0)
            fatputs(dizi[j+1], &mylog);
            fatputs(dizi[j+2], &mylog);} // dizinin o an ki elemanından '0' değerini
çıkarıyoruz. 2 basamak lı sayı işlemi yapıyoruz
        else
            {ay=(dizi[j+1]-48);
```

```
        if(fatputs(logger2, &mylog) == 0)
            fatputs(dizi[j+1], &mylog);}

        motorsur(ax,ay);

        break;
    }
    j++;
}

if(fatclose(&mylog) == 0)
    printf("\rDosya kapandı");
else
    printf("\r Kapatma hatasi");

}
```

```
void main(void) {
    set_tris_b(0x00);
    setup_timer_0(RTCC_INTERNAL | RTCC_DIV_1);
    //! // setup_wdt(WDT_ON);
    //! ext_int_edge(L_TO_H);
    //! enable_interrupts (INT_TIMER0);
    //! enable_interrupts(INT_EXT);
    //! enable_interrupts(GLOBAL);
```

```
//! delay_us(10);  
//!  
  
glcd_init(ON); // Grafik LCD hazırlanıyor  
glcd_text57(5, 5, "Ilyas Huseyin Guvenc", 1, ON);  
glcd_text57(15, 30, "170223025", 2, ON);  
  
delay_ms(2000);  
  
while(TRUE){  
  
    sec=Ana_Menu();  
  
    switch(sec){  
        case 1: sd_kart(); break;  
        case 2:uyku_modu(); break;  
        case 3: break;  
        case 4: about(); break;  
    }  
  
}  
  
}
```



## Kaynakça

[1] FAT İmage dosyası İndirme linki,

[https://drive.google.com/file/d/1n\\_nSPxnHZh4qivpiT5WvRjHnKrsoEuuo/view?usp=sharing](https://drive.google.com/file/d/1n_nSPxnHZh4qivpiT5WvRjHnKrsoEuuo/view?usp=sharing)

[2] FAT ve Sd kart Kütüphaneleri,

<https://drive.google.com/file/d/18Pg3YEAUnJxdOVx3IZUtYBhdxaSajF1u/view?usp=sharing>

, <https://drive.google.com/file/d/140S8jiY1bSwy0Kz55Z7pBbwZ-EKvH0bE/view?usp=sharing>

[3] G-Code hakkında, <https://hayaletveyap.com/g-kodu-nedir-g-code-cnc-ve-3d-yazici-kod-ornegi/>

[4] FAT dosya sistemi, <https://simple-circuit.com/pic18f4550-sd-card-read-write-files/>