

TERMAL ETİKET YAZICISI

AHMET YILDIZ 100224002
HÜSEYİN TEKECİ 100223031
MAKSUT KAYA 090224040

Çalışmanın içeriği

- Bu çalışmada mikroişlemci kontrollü termal etiket/fatura yazıcısı yapılması amaçlanmıştır.
- Temin edilen panel yazıcının kontrolü ile-yazıcıya uygun mikroişlemci devresi yapılarak- istenilen metin ya da Matlab-Gui ile logonun çıktısının alınması sağlanacaktır.
- Bu işlemler yapılması için yazıcı mekanizmasının komut rehberinden yararlanılmıştır ve gerekli yazılım sağlanmıştır
- RS-232 bağlantılı bir mikroişlemci devresi kurulmuş ve gönderilen metin parçaları devredeki Lcd ekranda görüntülenmiştir.
- Lcd panelde görüntülenen karakterler numpad ile girilmiştir.
- Tuş takımında yazdırma tuşuna basıldığında uygun hex kodlarının yazıcıya transferi sağlanarak termal yazıcıdan çıktı alınmıştır.
- Son çıktı alınan metnin EEPROM belleğinde saklanmasıyla kolaylıkla tekrar çıktı alınması sağlanmıştır.
- Ayrıca kullanıcının bilgisayar aracılığıyla çıktı alabilmesi için Matlab-GUI de arayüz tasarlanmıştır.

Giriş

- Termal yazıcılar pek dikkat etmesek te günlük hayatımızda birçok yerde rastladığımız makinalardır.
- Marketlerden çıkışken çoğu zaman almadığımız fişler ya da banka sıralarında kiosk numaratörlerinden aldığımız sıra fişleri birer termal yazıcı ürünleridir.
- Görüldüğü gibi pos cihazları fişleri, ATM bilgi fişleri, numaratör çıktıları gibi birçok alanda başvurulan bu sistem günümüzde baskı sektöründe önemini hala korumaktadır.

Peki, neden diğer baskı yöntemleri bu alanlarda kullanılmamaktadır?

- baskı yönteminin ne olduğunu ve baskı mantığını
 - Kısaca belirtecek olursak termal baskı sistemlerinde mürekkep ihtiyacı bulunmamaktadır. Bu sistemler tercih için ilk sebep olabilir.
 - Anlık önemi olan bir çıktı için mürekkep harcamak oldukça gereksizdir. Bu safhada Termal yazıcılar çok önem kazanmaktadır.(örn: fiş, sıra numarası, fatura vb.)
 - sarf malzemesi sadece termal kâğıt olmaktadır. Kullanıcılar mürekkep, toner gibi masraflardan kurtulmuş olurlar.
 - Kâğıdın yazılacak kısımlarının termal iğneler sayesinde belli algoritmalarla planlı bir şekilde yakılmasıyla termal çıktı oluşmaktadır
 - Bu projemizde de bu amaçlar doğrultusunda termal baskı yöntemiyle hem etikete hem de termal kâğıda çıktı alınabilinen sistem tasarlanmıştır.

Yazıcı çeşitleri

Günümüzde farklı şekillerde baskı yapan yazıcılar mevcuttur

- Nokta vuruşlu yazıcılar (dot matrix printer)
 - yazma kafası bir matris şeklinde dizilmiş küçük iğneciklerden oluşmaktadır.
 - Bilgisayardan gelen sinyale bağlı olarak kafanın içindeki elektromıknatıslar yardımıyla bu iğnelerin bazıları öne çıkar, daktilodaki gibi nokta vuruşlarla şeillerin tanımlanması mümkündür.
 - Bir nokta vuruşlu yazıcıdan çıkan metinlerde, karakterler çeşitli noktaların yan yana getirilmesinden oluşmaktadır.

- Mürekkep Püskürtmeli Yazıcılar (Inject Printer)
 - Bunlar ince nozüllerden kâğıda direk olarak küçük mürekkep damlaları fişkirtma ilkesine göre çalışır.
 - . Nokta vuruş teknolojisinden farklı olarak, yazıcı kafası kâğıda değil.
- Lazer Yazıcı (Laser Printer)
 - Son geliştirilen yazıcı türüdür. Sessiz, yüksek baskı kalitesine sahip ve diğer yazıcılara göre daha hızlıdır
 - Baskı yöntemi fotokopi makinesindekine benzerdir. Satır satır yazmak yerine sayfa sayfa yazarlar.
 - Lazer yazıcılardaki ROM basılacak doküman tam sayfa bir haritasını oluşturur.
 - Bu bit haritası lazer ışını darbeleri ile aynadan yansıyarak ışığa duyarlı dönen bir silindir üzerine düşürülür. Lazer ışını silindiri tarayarak basılı alanları elektriksel olarak nötr hale getirir. Negatif yüklenmiş toner tozu nötr alanlara yapışır. Merdanenin sıcaklığı karakteri oluşturan noktaların kâğıda geçmesini sağlar.

Termal yazıcılar

- Termal yazıcılar ilk olarak 1960lı yılların sonunda, siyah-beyaz görüntü elde etmek amacıyla, faks makinelerinde ve gazete basımında kullanılmıştır.
- Termal yazıcılarda çizgisel dizili ısıtıcı elemanları, üzerlerinden (altlarından) geçen ısuya duyarlı kâğıda dokunarak kağıdı ısıtip baskı yaparlar. (termal kağıt)
- Mürekkep, şerit veya kokulu kimyasal madde kullanımı gerektirmez.
- Tek tek karakter yazımı tekniği ile işler.
- Ayrıca çok az mekanik parça gerektirdiği için düzenlenmesi kolaydır.

Termal baskı yöntemleri

□ Direkt baskı yöntemi

- Bu yöntemde baskı işlemi kâğıdı ısıtarak yapılır ve kağıt siyah renge dönüşür.
- Bu baskı yöntemi kısa sürelidir. Çünkü güneşten, ısından, yoğun ışıktan etkilenirler ve bozulmaya uğrayabilirler. Fakat kısa süreli kullanımlar için idealdir
- Ribon masrafı yoktur
- Termal yazıcının fiyatı termal transfer yazıcıya göre daha ekonomiktir
- Endüstriyel anlamda ana firmalar Seiko ve Fujitsu'dur

■ Termal Transfer Baskı Yöntemi

- Bu yöntemle yapılan baskılar daha uzun sürelidir
- Güneş, ısı ve yoğun ışiktan doğrudan etkilenmez.
- Bu yöntemde ribon denilen şeritler kullanılır.
- Ribon ısıtılarak kâğıt üzerine yapıştırılır.
- plastik, dokuma gibi değişik etiketlere baskı yapabilir
- Yazıcı fiyatı termal yazıcıya göre pahalıdır
- Son zamanlarda üretilen pek çok termal transfer yazıcı direkt termal baskında yapabilmektedir.
- En büyük dezavantajı ilk yatırım maliyetinin yüksek olmasıdır

Panel yazıcılar

Termal Rulo kağıda çıktı almak için panellere monte edilen termal yazıcı modelleridir. Daha çok ATM'lerden bilgi fişi almaya yada sıra numaratorlarında kullanılır.

Ürün Adı	Fiyatı	Sipariş İmkanı	Temin Süresi
WOOSIM PORTI-P40/PP40	95\$	<u>http://www.aliexpress.com/(ÇİN)</u>	7-12 gün
CUSTOM PLUS- II	100€+KDV	TÜRKİYE-ANKARA	2 Gün
Mini Thermal Receipt Printer	49,95\$	<u>www.adafruit.com</u>	17-30 Gün
58mm thermal printer	63,71\$	<u>http://www.dhgate.com/(ÇİN)</u>	7-12 GÜN

Pugh Karar Matris Tablosu		KONSEPT SEÇENEKLERİ			
Kriterler	Ağırlık	1	2	3	4
Baskı Yöntemi Uygunluğu	0,1	5	5	5	4
Temin Süresi	0,3	3	4	1	3
Command Manual Temini	0,2	4	5	3	2
Güvenilirlik	0,2	2	5	4	3
Bağlantı Portu çeşidi	0,1	3	3	2	3
Fiyat	0,1	2	1	5	2
AĞIRLIK HESABI SONUCU	1	3,1	4,1	2,9	2,8



Projedeki ana elemanları

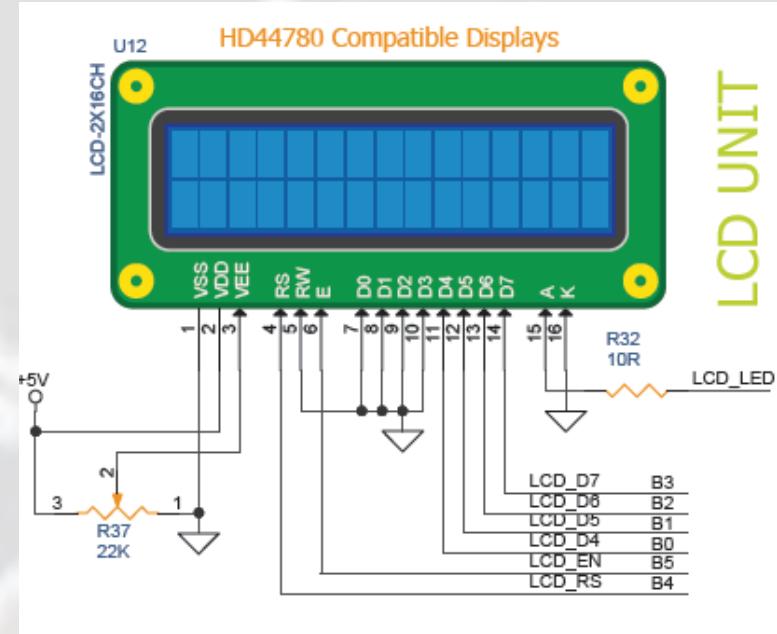
- 200 dpi high printing quality
- Same mechanics as PlusII (100% compatible)
- Paper width 58 mm
- Paper thickness 55-70gr/mq
- Roll dimension Ø 50mm
- Power supply 4-7,5Vdc
- 4MB internal flash memory for logos and font
- Interfaces: RS232/TTL (selectable with lever switch), miniUSB
- Font: European, International, Portuguese and Nordic, Chinese (GB18030) and Cyrillic
- Sensors: head temperature, paper presence, door open (opt.)
- Outdoor applications:-20 / +70°C working temperature



•2x16 LCD Ekran

Özellikleri;

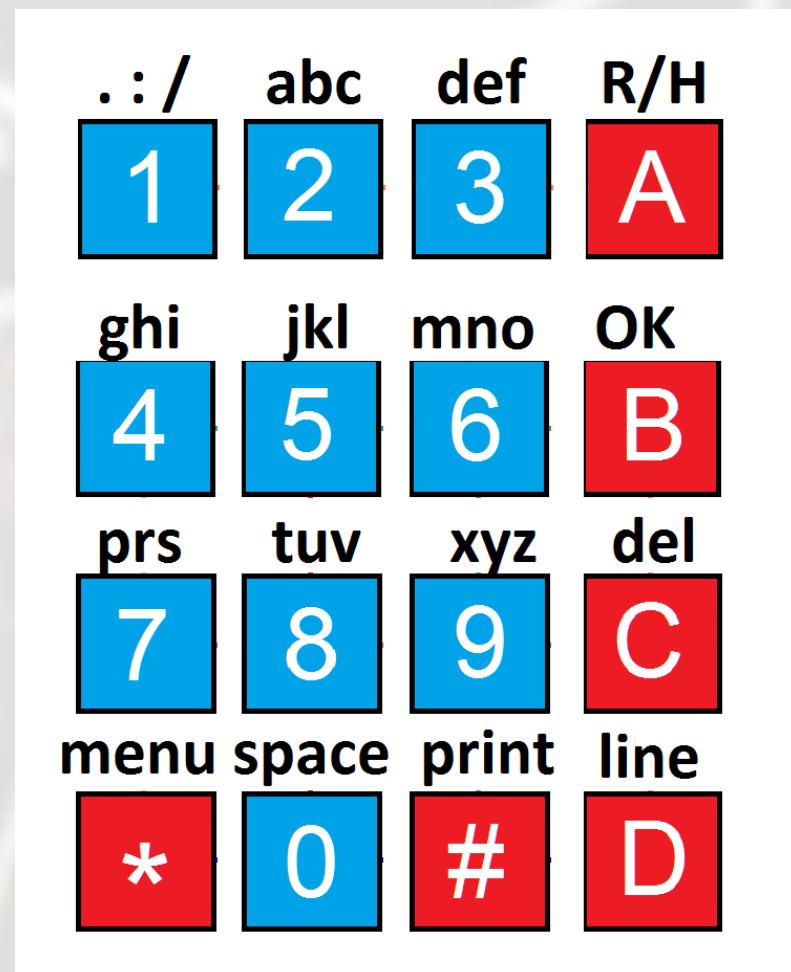
- +5V ile çalışmaktadır.
- Back Lighting özelliğine sahiptir.
- LCD arka fon ışığı olmadan 4mA akım çekmektedir.
- Boyutları 80x36x9.4mm'dir
- Çalışma sıcaklığı -20 ile +70 derece arasıdır.



LCD UNIT

4x4 Membran Tuş takımı (Keypad)

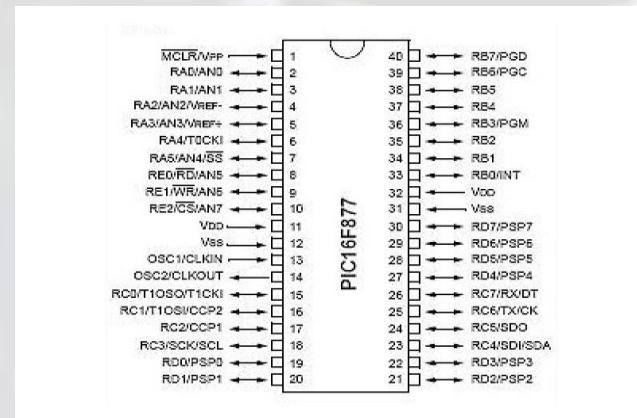
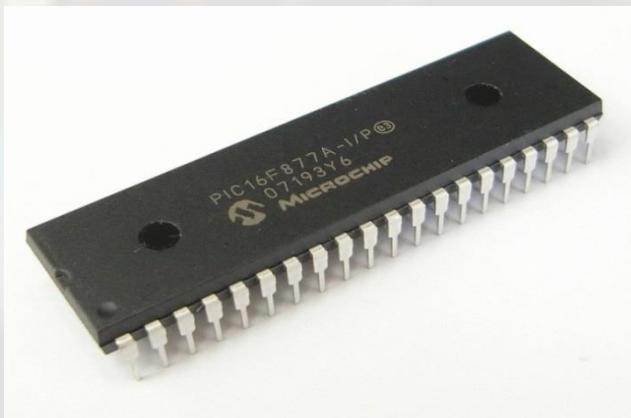
Kullanıcı girişleri için temel bir 16 butonlu tuş takımı. Butonlar matrix formatına uygun bir şekilde yerleştirilmiştir 16 butonlu membran keypad monteli bir şekilde kullanıcılara sunulur.



PIC16F877A

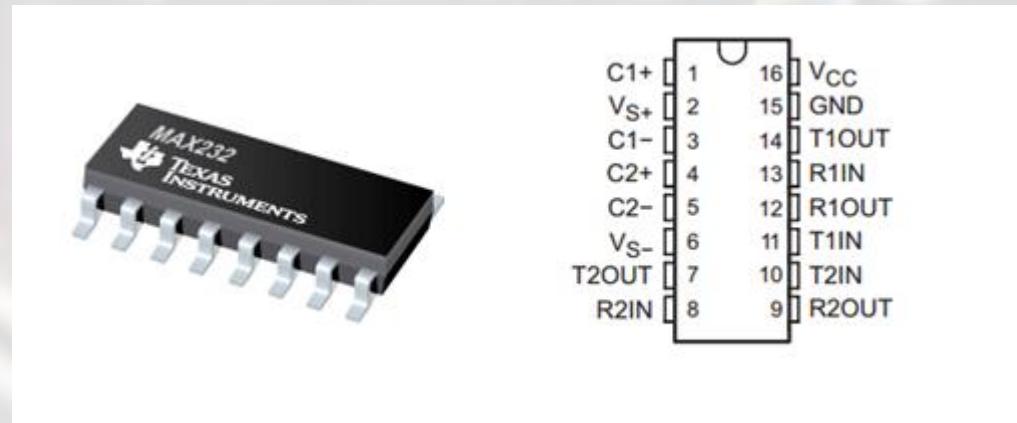
Temel Özellikleri

- *İşlem hızı DC-20 MHz dir. (bir komut DC-200 ns hızında çalışmaktadır.)
- *8 Kword Flash ROM programlama belleği (EEPROM özellikli program belleği), 368 Byte kullanıcı RAM belleği ve 256 Byte EEPROM belleği olmak üzere üç adet bellek bloğu vardır.
- *CPU azaltılmış komut setine sahiptir.
- *Öğrenilecek gereken her biri 14 bitlik 35 komut vardır.
- * 8 bitlik veri yolu (databus) vardır.
- *2,0 V ile 5,0 V arasında değişen geniş işletim aralığına sahiptir.
- *Kaynak akımı 25 mA dir.
- *Geniş sıcaklık aralığında ve düşük güçle çalışabilir.
- * Paralel ve seri haberleşme (spi, i2c, rs232 vb.)

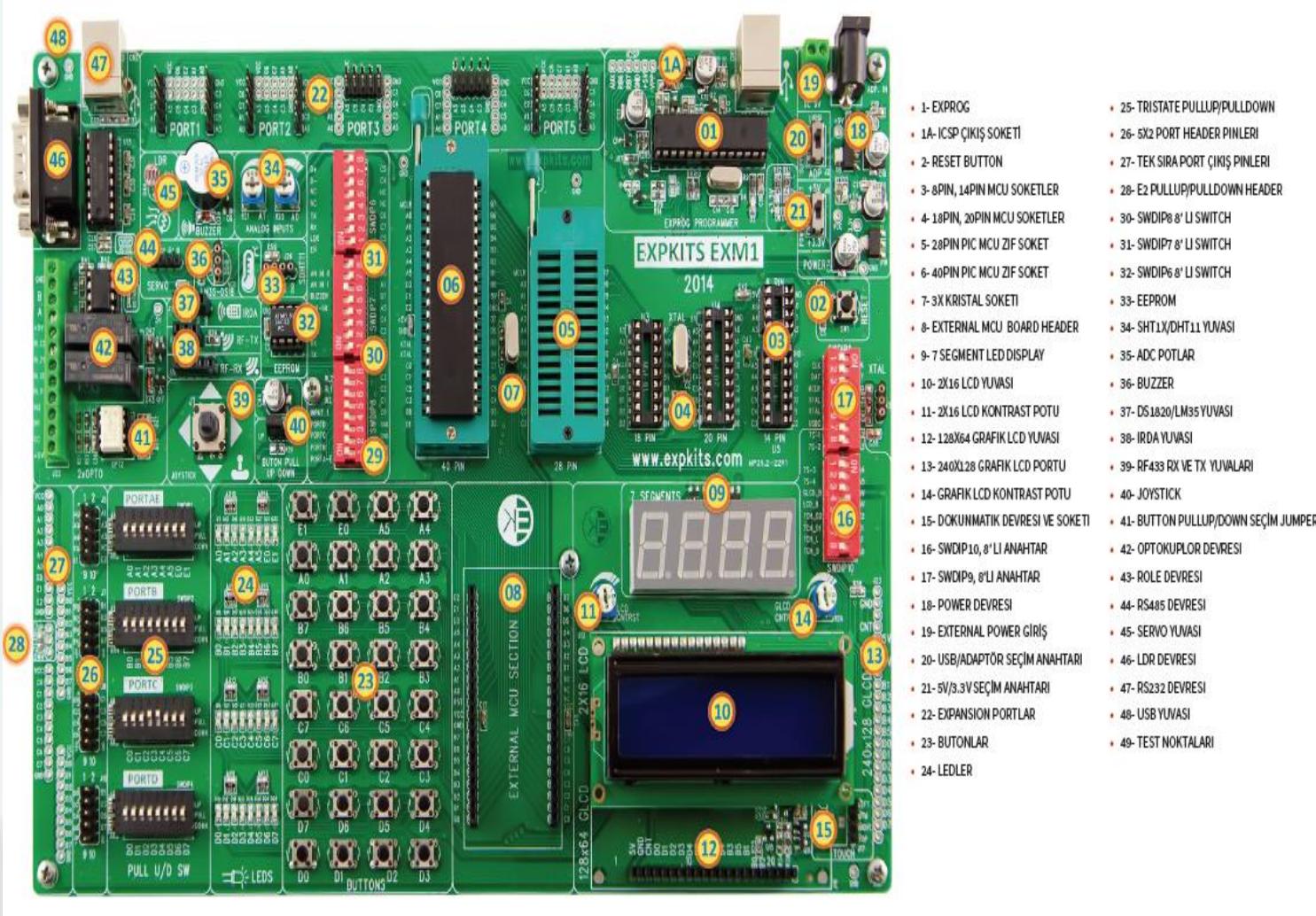


MAX232 DEVRESİ

Birçok mikrodenetleyici 5 volt ile çalışır. Bu mikrodenetleyicilerin high Kabul ettiği değer 5v, low Kabul ettikleri değer ise 0 volt yani gniddir. Buna TTL seviye denir. Bilgisayar seri portu için +12 V -12 V civarıdır. Bu durumda bilgisayar ile mcu devremizi direkt bağlamamız uygun değildir. Kullandığımız çevirimciler seri port ile TTL devrelerin çalışabilmesini sağlayan birer entegredir. Örnek olarak max232, 5volt ile çalışan cihazlarla bilgisayar arasındaki çeviriçi işini yapar. Bizim kullandığımız yazıcı modeli de RS232 seri haberleşmesini desteklemekte bu nedenle Projede TTL-RS232 dönüşümü yapılabilmesi için MAX232 devresi kullanılmıştır.



Proje yardımcı eleman



YAZICIDA KULLANILAN KOMUTLAR

- **0X02(Double height printing)**
- **0x0B <VT>(Forward feeds (n) lines)**
- **0x1D 0x57 0x30 <GS W>(Prints n byte of a 204 dpi graphic line)**

Detaylı tasarım

- Kullanıcının 3 satır olarak serbest karakterler girebilmesi sağlandı.
- Lcd de seçim yapabilmek için menü ekranı yapıldı. (BIL=1 PIC=2 HF=3)
- Lcd de ilk satırda girilecek olan karakterlerin görüntülenmesi, ikinci satırda ise kullanıcı bilgilendirilmesi sağlandı
- Hafıza ile son çıktısının kaydı sağlandı
- Matlab GUI de arayüz programı yapıldı

Ccs c ve lcd kütüphane kodları

```
□ #fuses HS
□ #use fast_io(a)
□ #use fast_io(b)
□ #use fast_io(c)
□ #use fast_io(d)
□ #use fast_io(e)
□ #use rs232 (baud=9600, xmit=pin_C6, rcv=pin_C7, parity=N, stop=1,
TIMEOUT=25)
□ #include <benim_LCD.c> // lcd.c dosyası tanıtılıyor
□ #define sut1 pin_d0 // sut1 ifadesi pin_d0 ifadesine eşitleniyor
□ #define sut2 pin_d1 // sut2 ifadesi pin_d1 ifadesine eşitleniyor
□ #define sut3 pin_d2 // sut3 ifadesi pin_d2 ifadesine eşitleniyor
□ #define sut4 pin_d3 // sut3 ifadesi pin_d2 ifadesine eşitleniyor
□ #define sat1 pin_d4 // sat1 ifadesi pin_d4 ifadesine eşitleniyor
□ #define sat2 pin_d5 // sat2 ifadesi pin_d5 ifadesine eşitleniyor
□ #define sat3 pin_d6 // sat3 ifadesi pin_d6 ifadesine eşitleniyor
□ #define sat4 pin_d7 // sat4 ifadesi pin_d7 ifadesine eşitleniyor
```

Tanımlanan değişkenler

```
char tus=' '; // karakter tipinde değişken tanımlanıyor
int i=0,j=0,k=1,k1=1,r=1,m=1,m1=1,n=1,n1=1,yzk=0,
int yzk1=0,yzm=0,sfr=0,bos=0,syc=0,syc1=0,ycz=0,q=0;
int ii=1;

int MENU=0; // ana menü seçiminin atıldığı değişken tanımlanıyor
int oku=0;
char t[]; // RDA dan gelen değişkenin saklandığı değişken tanımlanıyor
char mkt;

char abc[73]; // tuş takımı ile yazılan 3 satırın saklandığı değişken tanımlanıyor

char abc1[72]; // eepromdaki yazdırılan son yazının saklandığı değişken
tanımlanıyor
```

Eeprom yazma ve okuma fonksiyonları

```
void eeprom_yaz()
{
*****// girilen yazının 24*3satır
yazılma işlemi*****
    write_eeprom(0,abc[0]);
    write_eeprom(1,abc[1]);
    .
    .
    .
    write_eeprom(70,abc[70]);
    write_eeprom(71,abc[71]);
}
```

```
void eeprom_oku()
{
    syc=read_eeprom(75);
    k1=read_eeprom(74);
    m1=read_eeprom(73)
    n1=read_eeprom(72);
*****// eeproma yazılan 24*3satır
verinin okunması işlemi*****
    abc1[0]=read_eeprom(0);
    abc1[1]=read_eeprom(1);
    .
    .
    .
    abc1[70]=read_eeprom(70);
    abc1[71]=read_eeprom(71);
```

#int RDA kesmesi

```
mkt=getc();  
  
if(mkt=='Y') // yazı gelmesi durumunda  
{  
    for(yzk=0;yzk<72;yzk++)// 3 satır veri  
okuma  
    {  
        abc[yzk]=getc();  
    }  
    syc++;  
    if(syc==7) // kayma engelleyici sayacı  
    {  
        syc=0;  
        putc(0x31);  
        putc(0x0B);  
    }  
    putc(0x02);  
    for(yzk=0;yzk<72;yzk++)  
    {  
        putc(abc[yzk]);  
    }  
    putc(0x32);  
    putc(0x0B);  
}  
  
if(mkt=='M') // resim gelmesi durumunda  
{  
    oku++;  
    for(yzk=0;yzk<48;yzk++)  
    {  
        t[yzk]=getc();  
    }  
    putc(0x1D);  
    putc(0x57);  
    putc(0x30);  
    for(yzk=0;yzk<48;yzk++)  
    {  
        putc(t[yzk]);  
    }  
    if(oku==128)  
    {  
        putc(0x033);  
        putc(0x0B);  
        oku=0;  
    }  
}
```

```
if(mkt=='W') // resim ve 1. satır yazı  
gelmesi durumunda  
{  
    oku++;  
    if(oku<65) // 65 satır veri alımı  
    {  
        for(yzk=0;yzk<8;yzk++) // 8byte  
veri alımı  
        {  
            t[yzk]=getc();  
        }  
        putc(0x1D);  
        putc(0x57);  
        putc(0x08); // kaç byte veri  
gönderileceğinin seçimi  
  
        for(yzk=0;yzk<8;yzk++) // 8byte  
veri yazma  
        {  
            putc(t[yzk]);  
        }  
    }  
}
```

```
if(oku==65  
{  
    putc(0x1B);  
    putc('N');  
    putc(0x02);  
    for(yzk=0;yzk<24;yzk++)  
    {  
        abc[yzk]=getc();  
    }  
    syc++;  
    if(syc==7)  
    {  
        syc=0;  
        putc(0x31);  
        putc(0x0B);  
    }  
    putc(0x02);  
    for(yzk=0;yzk<24;yzk++)  
    {  
        putc(abc[yzk]);  
    }  
    putc(0x34);  
    putc(0x0B);  
    oku=0;  
}  
}
```

Tuş takımı okuma komutu

```
/*Port d'nin yüksek değerlikli bitlerinin hight olma durumuna göre  
taranan düşük değerlikli giriş bitlerinin hight olmasıyla hangi tuşa basıldığı  
kontrol ediliyor*/ // kaynak: Mikroişlemciler dersi  
/* tuşların birden fazla özelliği bulunmakta bu özellikler 0-9 haricindeki tuşlara  
ait olan sayıçların değerlerine göre seçilmektedir. */  
/* i=0 olması durumunda rakam yazılımı sağlanmaktadır*/  
/* i=1 olması durumunda harf yazılımı sağlanmaktadır */  
/* j sayacı mod3 e göre tuşların harici kaçinci özelliğinin kullanacağını  
belirlemektedir */  
/* j=0 1.özellik*/  
/* j=1 2.özellik*/  
/* j=2 3.özellik*/
```

TUŞ 1 kod içeriği

```
output_d(0x00);
output_high(sat1);
if (input(sut1))
{
while(input(sut1)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='1';
    if(i==1 && j==0) tus='.';
    if(i==1 && j==1) tus=':';
    if(i==1 && j==2) tus='/';
}
}
```

TUŞ A kod içeriği

```
if (input(sut4)) // 4. sütun okunuyor
{
while(input(sut4));
if (MENU==2) // harf/rakam seçimi
{
    i++;j=-1;
    i=(i%2);
}
}
```

TUŞ B onaylama kod içeriği

/* satırın sütunundan önceki kayıtlı harflerin tekrar ekranaya yazılma işlemi yapılmaktadır. Bunun nedeni ise tek satırda 3 ayrı satır kullanıldığından satırlar arası geçişte ekranda önceki harflerin görünmesini sağlamak
aynı işlemler satır 2 ve 3 içinde aşağıda sağlanmaktadır*/

```
if(r==1 && MENU==2) // satır 1 için tuş takımını veri girişi
{
    k++;
    lcd_hazırla();
    for(yzc=0;yzc<k;yzc++) printf(lcd_veri,"%c",abc[yzc-2]);
    //(abc[k-2]=tus for dan önce yazılmalı. Ekran sapıtmaması için)
    imlec(r,k);
    abc[k-2]=tus;
}
```

Örn; 3.sütun için veri girildiğinde be onaylandığında döngüden dolayı k=4 oluyor.
İmleç 4 e gitmeli fakat ekran ilk 3 terim için abc[] dizisinin 0,1,2 nolu elemanları
Yazdırılmalı bu yüzden abc[yzc-2] ekrana yazdırılıyor.

TUŞ C silme kod içeriği

```
tus=' ';
if(r==1 && MENU==2) // satır 1 için silme kodu
{
    k--;
    if(k==0)
    {
        k=1;
    }
    abc[k]=' ';
    imlec(1,k); // değiştirdin
    printf(lcd_veri,' ');
    imlec(1,k);
}
```

örn. 6. sütundaki eleman dizinin 5 indisine yazılır. Sayac 1 azaltığında Dizinin indisini sayacın kendisine eşit olur ve 'tus=' ile o karakter silinir

TUŞ * ana menü kod içeriği

```
/* yazıcının ana menüye dönüşünü sağlayan ana menü tuşu*/
if (input(sut1)) // 1. sütun okunuyor
{ while(input(sut1));r=1;k=1,n=1;m=1;MENU=0;lcd_hazirla();for(sfr=0;sfr<70;sfr++)
{abc[sfr]=' ';}tus=' ';
```

TUŞ # yazdırma kod içeriği

```
if(MENU==2)
{
lcd_hazirla();
printf(lcd_veri,"yazdiriliyor");
imlec(2,1);
printf(lcd_veri,"Adet %d: ",ii);
for(syc1=0;syc1<ii;syc1++)
{
syc=read_eeprom(75); // kaymayı önleyici daha önceki yazdırma sayısını okuma
syc++;
if(syc==7)
{
syc=0;
putc(0x31);
putc(0x0B);
}
write_eeprom(75,syc); // son yazdırılan verinin karakter sayılarını hafıza yazdır
write_eeprom(74,k); //komutu için kullanabilmek için eeproma kaydetme komutları
write_eeprom(73,m);
write_eeprom(72,n);
```

Önce 3. satır yazdırılıyor. (Sırası ile 3 - 2 - 1)

```
for(bos=0;bos<((25-n)/2);bos++) // ortalama için ön boşluk
{
    putc(' ');
}

for(yzm=0;yzm<n-1;yzm++) // veri yazımı
{
    putc(abc[48+yzm]);

}

for(bos=(((25-n)/2)+n);bos<25;bos++) // ortalama için son
boşluk
{
    putc(' ');
}
```

TUŞ D satır seçme kod içeriği

```
r++;  
if(r==5){r=1;}  
j=-1; // farklı bir tuşa basıldığında ilk özelliğindeki karakterin gelmesi için  
if(r==1 && MENU==2) // 1. satıra gelindiğinde girilen değerlerin aynen  
// kalmasını için if koşul komutları  
{  
if(abc[k-1]==' ') {tus=' '};  
else if (abc[k-1]!=' ') {tus=abc[k-1];}  
imlec(1,k);  
printf(lcd_veri,' ');  
imlec(1,k);  
}
```

Void main kod içeriği

```
void main ()  
{  
    setup_psp(PSP_DISABLED);      // PSP birimi devre dışı  
    setup_timer_1(T1_DISABLED);    // T1 zamanlayıcısı devre dışı  
    setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı  
    setup_adc_ports(NO_ANALOGS); // ANALOG giriş yok  
    setup_adc(ADC_OFF);          // ADC birimi devre dışı  
  
    //set_tris ayarları ile kullanılmayan bacaklar çıkış olarak ayarlandı  
    set_tris_b(0x00); // B portu komple çıkış  
    set_tris_d(0x0F); // Yüksek değerlikli 4 bit çıkış, düşük değerlikli 4 bit giriş  
    set_tris_a(0x00);  
    set_tris_e(0x00);  
    set_tris_c(0x80);  
    output_a(0x00);  
    output_e(0x00);  
    output_c(0x00);  
  
    eeprom_oku(); // fonksiyon olarak kullanılan eepromdan veri alma komutu
```

Menü seçim ekranı

MENUU:

```
while(MENU!=1 && MENU!=2 && MENU!=3)
{
    lcd_hazirla();
    printf(lcd_veri,"MENU SECINIZ:");
    imlec(2,1);
    printf(lcd_veri,"BIL=1 PIC=2 HF=3");

    if (keypad_oku()>9) // Eğer basılan tuş değeri 9'dan büyük ise
        // kaynak: mikroişlemciler dersi
    {
        .
        .
        .
    }
}
```



```
if(MENU==3)
{
    /* son yazdırılan yazının hafızadan alınıp tekrar
yazdırılması*/
    lcd_hazirla();
    printf(lcd_veri,"yazdiriliyor");
    eeprom_oku();
    .
    .
    .
    MENU=0;
    if (MENU==0) goto MENUU;
}
```



```
if(MENU==1)
{
    lcd_hazirla();
    printf(lcd_veri,"BAGLANTI TIPI");
    imlec(2,1);
    printf(lcd_veri,"BILGISAYAR");

    while(MENU==1)
    {
        keypad_oku(); // sadece ana menu tuşunu kontrol için vardır
    }
    if (MENU==0) goto MENUU;

}
```



```
if(MENU==2)
{
    lcd_hazirla() // LCD hazırlanıyor
    .
    .
    putc(0x02);
    while(1) // Sonsuz döngü
    {
        lcd_hazirla();
        imlec(2,1);
        printf(lcd_veri,"STR: ");
        printf(lcd_veri,"%d",r);
        imlec(2,8);
        printf(lcd_veri,"R/H");
        if(i==1)
            printf(lcd_veri," H");
        if(i==0)
            printf(lcd_veri," R");

        printf(lcd_veri," x");
        printf(lcd_veri,"%d",ii);
        imlec(1,1);
```

```
if(r==1) // satır 1 ekrana yazdırılıyor
{
    for(yzc=0;yzc<k-1;yzc++)
    {
        printf(lcd_veri,"%c",abc[yzc]);
    }
    imlec(1,k);
}
```



STR: 1 R/H R x1

LCD Kütüphane Kodları

```
// RB0 -> LCD'nin D4 ucuna  
// RB1 -> LCD'nin D5 ucuna  
// RB2 -> LCD'nin D6 ucuna  
// RB3 -> LCD'nin D7 ucuna  
// RB4 -> LCD'nin RS ucuna  
// RB5 -> LCD'nin E ucuna  
// R/W ucu direkt şaseye bağlanacak  
#define e pin_b5 // LCD'nin E ucu RB5 pinine bağlı  
#define rs pin_b4 // LCD'nin RS ucu RB4 pinine bağlı
```

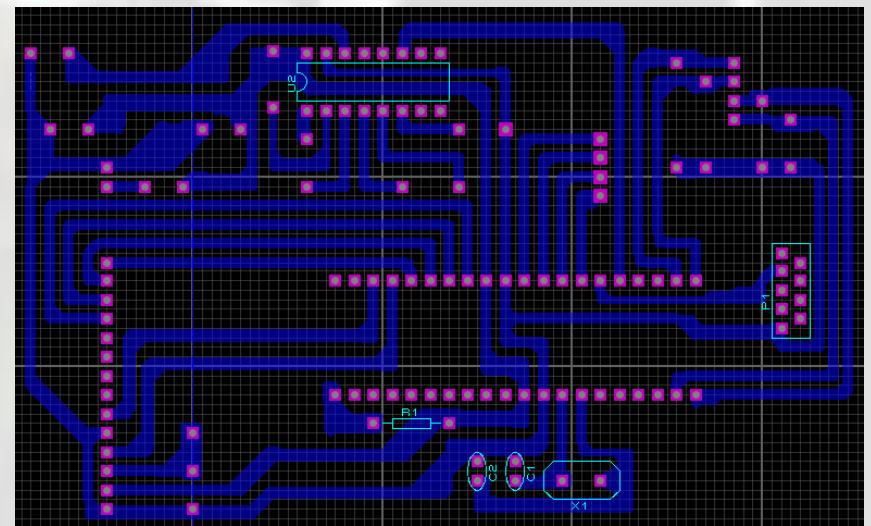
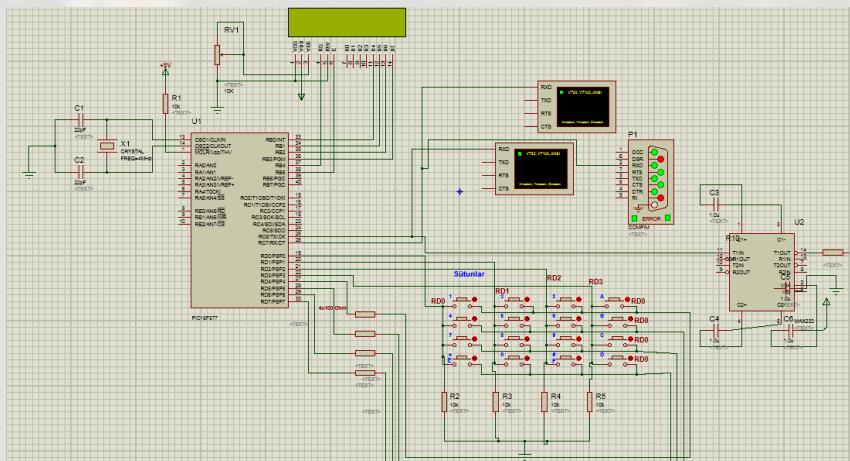
Komutlar

```
Lcd_komut();  
Lcd_hazirla();  
Lcd_veri();  
İmlec(satır,sütun)
```

Proteus 8.1

DEVRE SİMÜLASYON ÇİZİMİ

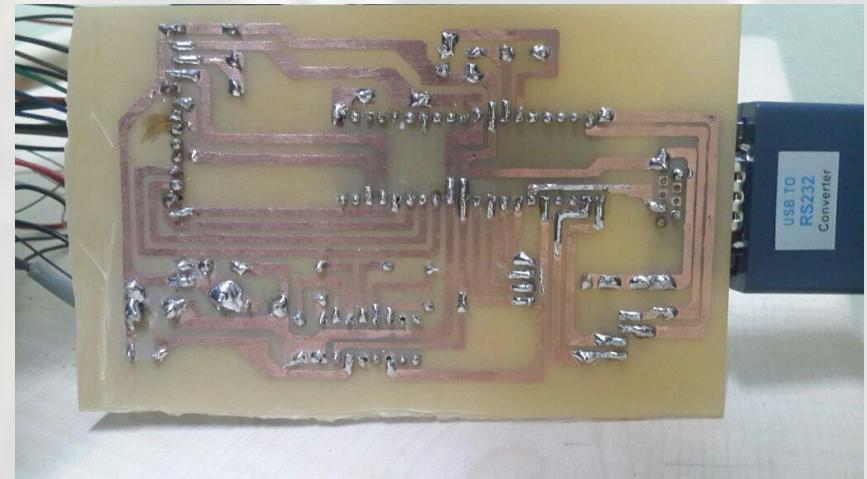
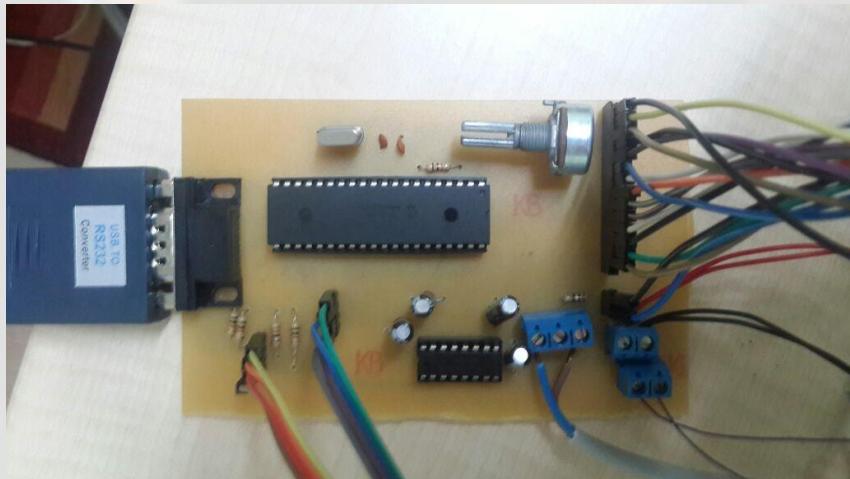
BASK DEVRE TASARIMI



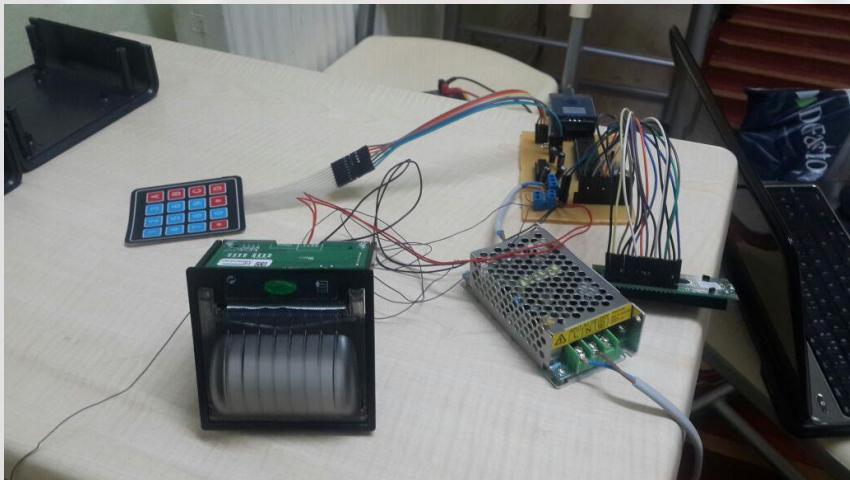
Devre fiziksel görünümü

DEVRE ELEMANLARI

YOL GÖRÜNÜMÜ



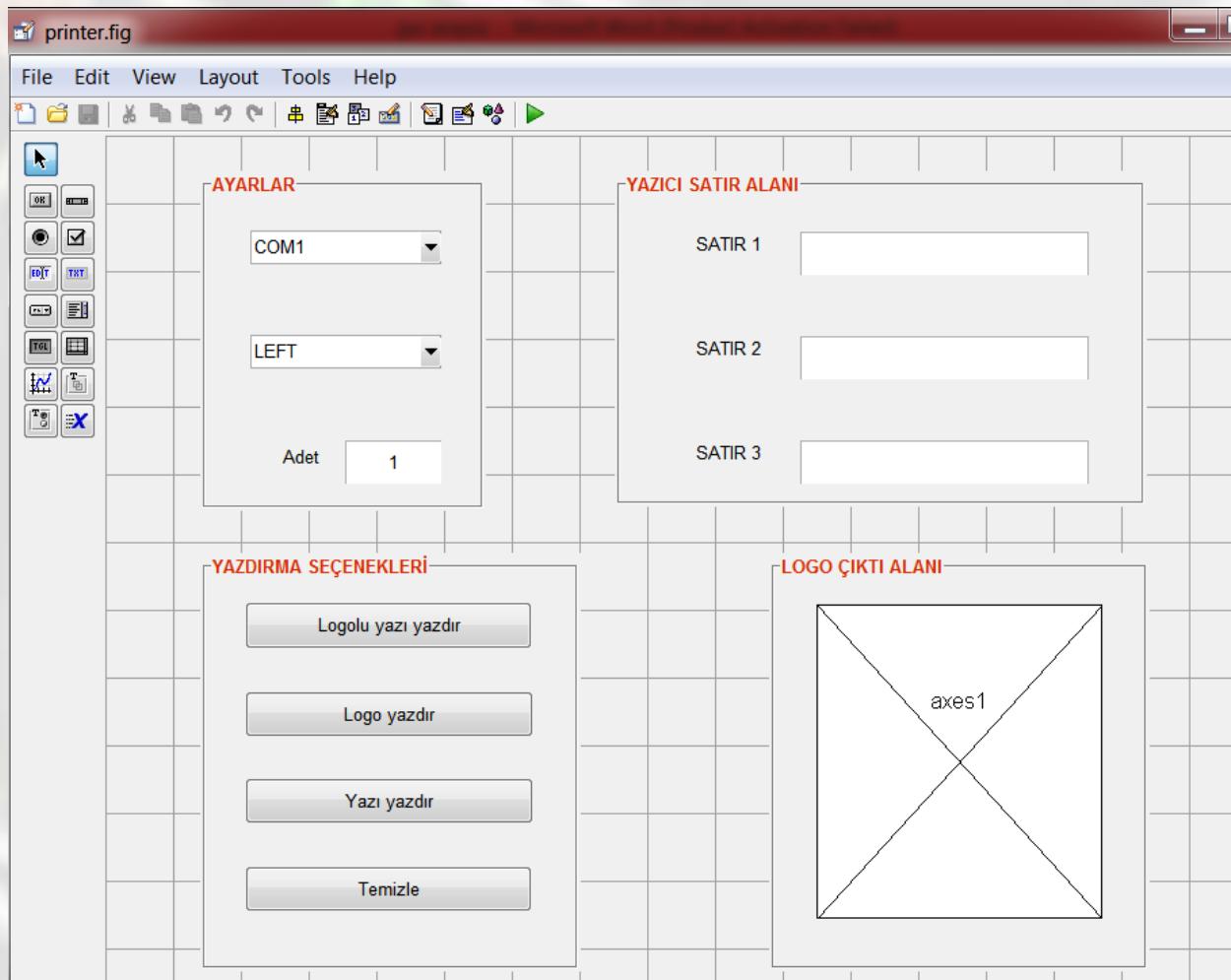
KUTULAMA ÖNCESİ



KUTULAMA SONRASI



Matlab Gui arayüz



Matlab Guide Kodları

Serial ayar komutları

```
if ss1== 1;  
s1=serial('COM1');  
End  
. . .  
if ss1== 10;  
s1=serial('COM10');  
End  
set(s1,'BaudRate', 9600,...);  
fopen(s1)  
fwrite(s1,'W'); // M -Y  
fclose(s1)  
delete(s1)
```

Bilgisayar port seçme komutları

```
global ss1;  
val=get(hObject,'Value');  
switch val  
case 1  
ss1=1;  
. . .  
case 10  
ss1=10;  
end
```

Edit text okuma komutları

```
adet=str2double(get(handles.edit5,'String'));
```

```
satir1=(get(handles.edit1, 'String'));
```

```
satir2=(get(handles.edit2, 'String'));
```

```
satir3=(get(handles.edit3, 'String'));
```

Yazının başlangıç konumu seçimi komutları

```
global s_yatik;
```

```
val1=get(hObject,'Value');
```

```
switch val1
```

```
case 1
```

```
s_yatik=1;
```

```
case 2
```

```
s_yatik=2;
```

```
case 3
```

```
s_yatik=3;
```

```
end
```

Görüntü işleme ve veri (bytes) gönderme kodları

```
[filename,pathname] = uigetfile('*.jpg','JPEG dosyasini seciniz');  
asd=imread([pathname,filename]);  
imshow(asd); // silmek için ise      cla reset  
asd_bw = im2bw(asd,125/255);
```

```
RGB = imresize(asd_bw, [64 64]);  
RGB=1-RGB;  
RGB180=imrotate(RGB,-180);
```

```
for i=1:64  
a(i,1)=bi2de(RGB180(i,1:8),'left-msb');  
a(i,2)=bi2de(RGB180(i,9:16),'left-msb');  
a(i,3)=bi2de(RGB180(i,17:24),'left-msb');  
a(i,4)=bi2de(RGB180(i,25:32),'left-msb');  
a(i,5)=bi2de(RGB180(i,33:40),'left-msb');  
a(i,6)=bi2de(RGB180(i,41:48),'left-msb');  
a(i,7)=bi2de(RGB180(i,49:56),'left-msb');  
a(i,8)=bi2de(RGB180(i,57:64),'left-msb');
```

Logo ve 1 satır yazı gönderme komutu içeriği

```
for j=1:64  
fopen(s1)  
fwrite(s1,'W');  
for i=1:8  
fwrite(s1,a(j,i));  
  
end  
fclose(s1)  
pause(0.001)  
end  
delete(s1)
```

Simülasyon ve sistemin çalıştırılması aşaması

BİZİ DİNLEDİĞİNİZ İÇİN TEŞEKKÜR EDERİZ...

AHMET YILDIZ 100224002
HÜSEYİN TEKECİ 100223031
MAKSUT KAYA 090224040