

**T.C**  
**KOCAELİ ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**MEKATRONİK MÜHENDİSLİĞİ**  
**BÖLÜMÜ**

**Bitirme Projesi**

**Etiket Makinesi**



AHMET YILDIZ 100224002  
HÜSEYİN TEKECİ 100223031  
MAKSUT KAYA 090224040

DANIŞMAN: YRD.DOÇ.DR. SELÇUK KIZIR

KOCAELİ

2015

## **TEŞEKKÜR**

Danışmanım ız Yrd. Doç. Dr. Selçuk KİZİR’e projenin ve tezin hazırlanması boyunca yaptığı katkılardan dolayı teşekkür ederiz.

## **ABSTRACT**

In this study, microprocessor controlled thermal label / bill is intended to make the printer. Available receipt of the panels made of any suitable microprocessor control circuitry in the printer's print text or logo will be output. These operations benefited from the instruction guide is planned to be sent to the printer mechanism to perform the appropriate code from the microprocessor to the system. For this purpose, a microprocessor circuit RS-232 connection is established and sent pieces of text displayed on the screen LCD. The characters displayed on the LCD panel are entered with numpad. Pressing the key on the keypad has been assigned the task of printing the appropriate hex code output from the thermal transfer printer has been providing printer. Again appeared to be easily stored in the EEPROM of the text received final output is provided. Also Matlab-GUI in order to get the output via the user's computer is also designed interface.

## İçindekiler Tablosu

TEŞEKKÜR .....	I
ABSTRACT .....	II
İçindekiler Tablosu .....	III
ŞEKİLLER LİSTESİ .....	V
TABLolar LİSTESİ .....	VI
BÖLÜM 1. LİTERATÜR TARAMASI .....	7
1.1 Giriş .....	7
1.2 Yazıcı Çeşitleri .....	8
1.3 Termal Yazıcılar ve Termal baskı Yöntemi .....	8
1.3.1 Termal Baskı Yöntemleri .....	9
1.4 Termal baskı kafasının yapısı ve tarihsel gelişimi .....	11
1.4.1 Termal Baskı kafası hakkında Bugüne kadar yapılmış çalışmalar .....	11
1.4.2 Termal yazıcı kafasının yapısı .....	13
1.4.3 ENDÜSTRİYEL ÜRÜNLER .....	14
1.4.4 PATENT .....	17
BÖLÜM 2. PROBLEM TANIMLANMASI .....	18
2.1 PROBLEM TANIMI VE TASARIM ÖZELLİKLERİ .....	18
2.1.1 Proje Örnekleri .....	18
BÖLÜM 3. PROJE PLANI VE YÖNETİM .....	20
3.1 PROJENİN ORGANİZASYONU .....	20
3.2 GÖREVLER VE AÇIKLAMALAR .....	20
3.2.1 Literatur Taraması .....	20
3.2.2 Kavramsal Tasarım .....	20
3.2.3 Ayrıntılı Tasarım ve Üretim(TM1,TM2,TM3) .....	21
3.2.4 Dökümantasyonlar(TM1,TM2,TM3) .....	21
3.3 ZAMAN TABLOSU .....	21
BÖLÜM 4. KAVRAMSAL TASARIM .....	22
4.1 KAVRAMSAL TASARIM VE SUNUM .....	22
4.1.1 Panel Yazıcılar .....	23
4.2 KONSEPT SÜRECİ .....	24
4.3 En İyi Kavramsal Tasarım .....	25
4.3.1 Tasarımda Kullanılacak olan malzemeler ve özellikleri .....	25
4.3.2 TERMAL YAZICIDA KULLANILAN KOMUTLAR VE İLK PROGRAM .....	36
BÖLÜM 5 DETAYLI TASARIM .....	39
5.1 Proteus Çizimi .....	39
5.2 Baskı Devre Tasarımı .....	40

5.3 Baskı Devre ve Fiziksel olarak Projenin Tamamlanması.....	40
BÖLÜM 6 MATLAB GUI İLE KULLANICI ARAYÜZÜ .....	43
6.1 MATLAB KULLANICI ARAYÜZÜ .....	43
6.1.1 GRAFİKSEL KULLANICI ARAYÜZÜ NASIL ÇALIŞIR.....	43
6.1.2 MATLAB’DA GUI OLUŞTURMA YÖNTEMLERİ .....	44
6.1.3 MATLAB GUIDE ARACI İLE GUI TASARIMI OLUŞTURMA .....	44
6.1.4 Komponentleri Çalışma Alanına Ekleme .....	45
6.1.5 Çalışma Alanının Boyutlarını Değiştirmek .....	46
6.2 GUI Tasarımını Kaydetme ve Çalıştırma .....	47
6.3 GUI Arayüzünün Programlanması .....	47
6.4 GUIDE ARAYUZU BAŞLANGIÇ .....	48
6.5 GUIDE ARAYUZU SON HALİ.....	48
EKLER .....	49
Ek 1. Deneme Amaçlı Yapılan ilk programın Kodları .....	49
Ek 2. Açıklamalı CCS C Kodları.....	55
Ek 3. LCD Kütüphane Kodları .....	73
Ek 4. Guide Kodları.....	75
Ek 5. KULLANIM KILAVUZU.....	102
Ek 5.1 Tuş takımı tabanlı çalışma .....	104
Ek 5.1.1 Tuş takımı tanıtımı ve fonksiyonu .....	104
Ek 5.2 Program kullanım örneği ve ana menü görüntüsü .....	105
Ek 5.3 Menü 1 seçimi .....	105
Ek 5.4 Menü 2 seçimi .....	105
Ek 5.5 Menü 3 seçimi .....	108
Ek 6 Matlab arayüzü.....	109
Ek 6.1 Ayarlar .....	109
Ek 6.2 Yazıcı satır alanı.....	109
Ek 6.3 Logo çıktıları.....	109
Ek 6.4 Yazdırma seçenekleri .....	109
Ek 7 Olası hatalar .....	110
Kaynakça .....	112

## ŞEKİLLER LİSTESİ

ŞEKİL 1: TERMAL YAZICI[2] .....	9
ŞEKİL 2: RİBONUN İÇ YAPISI[2] .....	10
ŞEKİL 3:TERMAL TRANSFER YAZICI ÇALIŞMA PRENSİBİ[2].....	11
ŞEKİL 4: YACIZI SİSTEMLERİNİN TARİHSEL GELİŞİMİ .....	13
ŞEKİL 5:TERMAL YAZICI KAFASININ YAPISI .....	14
ŞEKİL 6: LTP1245 PRINTER(SEIKO) [6] .....	14
ŞEKİL 7: FTP-628MCL101(FUJITSU)/ÇİN[7] .....	15
ŞEKİL 8: BİR DİREKT TERMAL YAZICI TEKNİK RESMİ[9].....	17
ŞEKİL 9:TERMAL YAZICININ ARDUİNO İLE BAĞLANTISI VE ENDÜSTRİYEL ÜRÜN .....	18
ŞEKİL 10:CUSTOM PLUS II TERMAL YAZICI .....	25
ŞEKİL 11: TERMAL YAZICI BAĞLANTI ŞEMASI .....	26
ŞEKİL 12:TERMAL YAZICI HABERLEŞME ŞEMASI .....	27
ŞEKİL 13:LCD EKRAN .....	29
ŞEKİL 14:4X4 TUŞ TAKIMI .....	30
ŞEKİL 15 PIC16F877A ENTEGRESİ .....	30
ŞEKİL 16:PIC16F877A BAĞLANTI ŞEMASI.....	32
ŞEKİL 17 :MAX232 ENTEGRESİ .....	33
ŞEKİL 18:MİKROİŞLEMCİ DENEME KİTİ .....	33
ŞEKİL 19:TÜM DEVRENİN PROTEUS ÇİZİMİ.....	39
ŞEKİL 20:TÜM DEVRENİN ARES ÇİZİMİ .....	40
ŞEKİL 21:BASKILANMIŞ DEVRENİN ALT GÖRÜNÜŞÜ.....	40
ŞEKİL 22:BASKILANMIŞ DEVRENİN ÜST GÖRÜNÜŞÜ .....	41
ŞEKİL 23:BÜTÜN KOMPANETLERİN BİRLEŞTİRİLMESİ .....	41
ŞEKİL 24: KUTU TASARIMI .....	42
ŞEKİL 25:MONTELENMİŞ ÜRÜNÜN ÖN,ARKA PERSPEKTİF GÖRÜNTÜSÜ .....	43
ŞEKİL 26 BASİT BİR GUİ ARAYÜZÜ TASARIMI .....	43
ŞEKİL 27 GUİ BAŞLANGIÇ EKRANI .....	44
ŞEKİL 28 GUİ ÇALIŞMA ALANI.....	45
ŞEKİL 29:GUİ KOMPANENT EKRANI.....	45
ŞEKİL 30:GUİDE ELEMAN SEÇİMİ .....	46
ŞEKİL 31 NESNELERİ HİZALAMA KOMUTU .....	46
ŞEKİL 32 GUİDE M-FİLE KOD ARAYÜZÜ .....	47
ŞEKİL 33:GUİDE TASARIMIN İLK HALİ .....	48
ŞEKİL 34:BİLGİSAYAR ARAYÜZÜNÜN SON HALİ .....	48
ŞEKİL 35: İLK PROGRAMIN YAZICI ÇIKTISI.....	55
ŞEKİL 36: YAZICININ ARKA KISMI.....	102
ŞEKİL 37: YAZICININ LCD VE KEYPADİN BULUNDUĞU ÖN KISIM .....	103
ŞEKİL 38:TUŞ TAKIMI.....	104
ŞEKİL 39:ANA MENÜ EKRAN GÖRÜNTÜSÜ .....	105
ŞEKİL 40:MENÜ 1 EKRAN GÖRÜNTÜSÜ .....	105
ŞEKİL 41: MENÜ 2 EKRAN GÖRÜNTÜSÜ .....	105
ŞEKİL 42:SATIR2 EKRAN GÖRÜNTÜSÜ .....	106
ŞEKİL 43: SATIR1 EKRAN GÖRÜNTÜSÜ .....	106
ŞEKİL 44: SATIR1 EKRAN GÖRÜNTÜSÜ .....	107
ŞEKİL 45 : SATIR1 EKRAN GÖRÜNTÜSÜ (+ TUŞ C) .....	107
ŞEKİL 46:YAZDIRMA EKRANI (+ TUŞ C) .....	107
ŞEKİL 47: ÇIKTI ÖRNEĞİ.....	107
ŞEKİL 48:MENÜ 3 YAZDIRILIYOR EKRANI.....	108
ŞEKİL 49: GUİDE KULLANICI ARAYÜZÜ .....	109
ŞEKİL 50:RESİM SEÇME ARAYÜZÜ .....	110

## **TABLÖLAR LİSTESİ**

TABLO 1: LPT1245 ÖZELLİKLERİ[6].....	15
TABLO 2: FTP-628MCL101(FUJİTSU) ÖZELLİKLERİ[7].....	16
TABLO 3: ZAMAN TABLOSU.....	21
TABLO 4:TERMAL BASKI YÖNTEMLERİ KARŞILAŞTIRILMASI.....	22
TABLO 5: ÇEŞİTLİ PANEL YAZICILARIN KARŞILAŞTIRILMASI .....	23
TABLO 6: PUGH KARAR TABLOSU.....	24

## **BÖLÜM 1. LİTERATÜR TARAMASI**

### **1.1 Giriş**

Termal yazıcılar pek dikkat etmesek te günlük hayatımızda birçok yerde rastladığımız makinalardır. Marketlerden çıkarken çoğu zaman almadığımız fişler ya da banka sıralarında kiosk numaratörlerinden aldığımız sıra fişleri birer termal yazıcı ürünleridir. Aynı zamanda market raflarında barkod, ürün adı ve fiyatının belirtildiği etiketler termal etiket makinalarından çıktı alınır. 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? Bu sorunun cevabı için termal baskı yönteminin ne olduğunu ve baskı mantığını bilmek gerekmektedir. Kısaca belirtecek olursak termal baskı sistemlerinde mürekkep ihtiyacı bulunmamaktadır. Mürekkep giderinin olmadığı bu sistemler tercih için ilk sebep olabilir. Diğer taraftan kullanım alanları insanlar için saklanacak diğer bir deyişle termal yazıcının çıktısı kişiler için kısa süreli bir önem taşımaktadır. Alınan fiş, sıra numarası ya da fatura belli bir süre sonra atılmakta ve kullanıcı tarafından önemini yitirmektedir. Anlık önemi olan bir çıktı için mürekkep harcamak oldukça gereksizdir. Bu safhada Termal yazıcılar çok önem kazanmaktadır. Kâğıdın yazılacak kısımlarının termal iğneler sayesinde belli algoritmalarla planlı bir şekilde yakılmasıyla termal çıktı oluşmaktadır. Bu durumda sarf malzemesi sadece termal kâğıt olmaktadır. Kullanıcılar mürekkep, toner gibi masraflardan kurtulmuş olurlar.

Bu projemizde de bu amaçlar doğrultusunda termal baskı yöntemiyle hem etikete hem de termal kâğıda çıktı alınabilen sistem tasarlanacaktır.



## 1.2 Yazıcı Çeşitleri

Bilgisayar ortamındaki bilgileri kağıt üzerine aktarmak için yazıcılar (printer) kullanılır. Günümüzde farklı şekillerde baskı yapan yazıcılar mevcuttur.

Örneğin:

Nokta vuruşlu yazıcılar (dot matrix printer): Genelde daktiloya benzerler. Fakat daktilodan farkı, yazma kafalarının elektronik bir kayışla hareket etmesidir. Bu yazıcıların yazma kafası bir matris şeklinde dizilmiş küçük iğneciklerden oluşmaktadır. Bilgisayardan gelen sinyale bağlı olarak kafanın içindeki elektro mıknatıslar yardımıyla bu iğnelerin bazıları öne çıkar, daktilodaki gibi nokta vuruşlarla şekillerin 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 fişkırtma ilkesine göre çalışır. Nokta vuruş teknolojisinden farklı olarak, yazıcı kafası kâğıda değmez. Bunun yerine yazıcı kafası kâğıda mürekkep damlacıkları püskürtür. Yazıcı kafası dikey olarak yerleştirilmiş birçok püskürtücü ucundan kâğıda minik noktalar halinde özel bir mürekkep püskürtür.

Lazer Yazıcı (Laser Printer): Son geliştirilen yazıcı türüdür. Sessiz, yüksek baskı kalitesine sahip ve diğer yazıcılara göre daha hızlıdır. Lazer yazıcılarda kullanılan baskı yöntemi fotokopi makinesindeki gibi benzer. Lazer yazıcılar satır satır yazmak yerine sayfa sayfa yazarlar. Lazer yazıcılar bütün sayfayı bir kerede basmak için geniş bir bellek kullanır. Lazer yazıcılardaki ROM basılacak doküman tam sayfa bir haritasını oluşturur. Bir bit haritası lazer ışını darbeleri ile bir sıra aynadan yansıyarak ışığa duyarlı dönen bir silindiri üzerine düşürülür. Lazer ışını silindiri tarayarak basılı alanları elektriksel olarak nötr hale getirir. Negatif yüklenmiş toner tozu nötr alanlara yapışır, negatif yüklü alanlara yapışmaz. Merdanenin sıcaklığı karakteri oluşturan noktaların kâğıda geçmesini sağlar.[1]

## 1.3 Termal Yazıcılar ve Termal baskı Yöntemi

Projenin asıl konusu olan termal yazıcılar ısıya duyarlı kâğıt üzerine baskı yapan yazıcılardır. Termal yazıcılar ilk olarak 1960'lı yılların sonunda, siyah-beyaz görüntü, elde etmek amacıyla faks makinelerinde ve gazete basımında kullanılmıştır. Termal yazıcılar elektronik olarak kaydedilen görüntülerin yüksek kaliteli işlenmesini sağlarlar. Termal yazıcılarda bilgisayardan gelen sinyaller ile kontrol edilen çizgisel dizili ısıtıcı elemanları, üzerlerinden (altlarından) geçen ısıya duyarlı kâğıda dokunarak ısıtıp baskı yaparlar. Hızlı ve sessizdirler, saklanması her zaman gerekli olmayan ve yüksek baskı kalitesi gerektirmeyen; ATM çıktıları, biletler, pos kâğıtları vb. baskı işlerinde

kullanılırlar.[2] Mürekkep, şerit veya kokulu kimyasal madde kullanımı gerektirmez. Tek tek karakter yazımı tekniği ile isler, böylelikle yazılan son karakter hemen girilebilir. Ayrıca çok az mekanik parça gerektirdiği için düzenlenmesi kolaydır.[3]



**Şekil 1: Termal Yazıcı[2]**

### **1.3.1 Termal Baskı Yöntemleri**

#### **1.3.1.1 Direkt Termal Baskı Yöntemi**

Bu yöntemde baskı işlemi kâğıdı ısıtarak yapılır. Bu yöntemde ısıya dayanıklı bir kağıt faks makinelerinde olduğu gibi ısıtılarak yakılır. Bu ısının etkisiyle kağıt siyah renge dönüşür ve böylece baskı işlemi yapılmış olur. Ancak bu yöntemle yapılan baskılarda çevresel faktörler önemli rol oynar. Bu baskı yöntemi kısa sürelidir. Çünkü güneşten, ısıdan, yoğun ışıktan etkilenirler ve bozulmaya uğrayabilirler. Fakat kısa süreli kullanımlar için idealdir.

- Ribon masrafı yoktur, yakarak basar.
- Termal etiket kullanılır. (Termal etiketler velium etikete göre pahalıdır).
- Etiket ömrü kısadır. Kısa sürede tüketilecek etiketler için uygundur.
- Termal yazıcının fiyatı termal transfer yazıcıya göre daha ekonomiktir.

### 1.3.1.2 Termal Transfer Baskı Yöntemi

Bu yöntemle yapılan baskılar daha uzun sürelidir. Bu yöntemde ribon denilen şeritler kullanılır. Termal baskıda ısıtılan kâğıdın yerini burada ribon alır. Ribon ısıtılarak kâğıt üzerine yapıştırılır. Daha sağlıklı ve dayanıklı bir yöntemdir. Güneş, ısı ve yoğun ışıktan doğrudan etkilenmez. Uzun ömürlüdür.

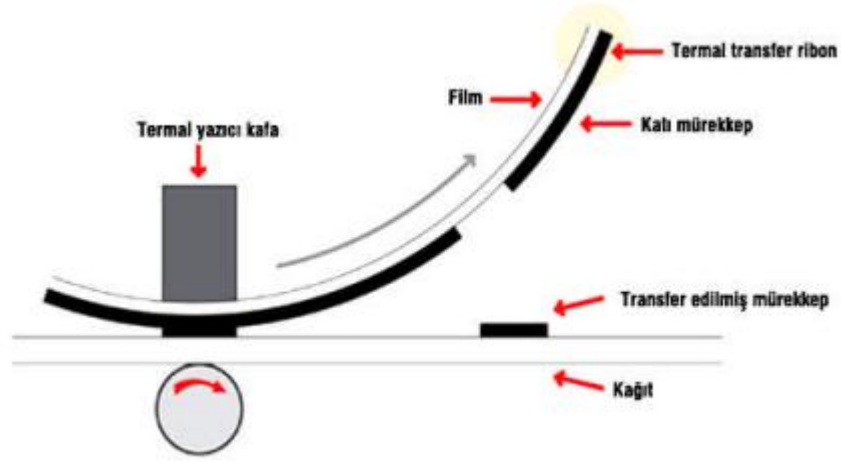
- Ribon masrafı vardır.
- Velium etikete veya plastik, dokuma gibi değişik etiketlere baskı yapabilir.
- Etiket ömrü çok uzundur.
- Yazıcı fiyatı termal yazıcıya göre pahalıdır. Son zamanlarda üretilen pek çok termal transfer yazıcı termal baskıda yapabilmektedir. Yani bu tip yazıcı aldığınızda her iki türlü de basabilme şansınız olur.

Termal transfer ribon üç katmandan oluşur. Ribonun orta kısmında film tabakası, bu tabakanın bir tarafında ısıyla sıvılaşan mürekkep, diğer tarafında da koruyucu tabaka bulunur.



Şekil 2: Ribonun İç Yapısı[2]

Polyester film üzerinde katı halde bulunan mürekkep, termal yazıcı kafa sayesinde ısıtılarak sıvı hale gelir ve etiket yüzeyine transfer edilerek baskı sağlanır. Mükemmel baskı kalitesinin sağlanabilmesi için yazıcı, ribon ve baskı yapılacak malzemenin uyumlu olması gerekmektedir.[2]



Şekil 3:Termal Transfer Yazıcı Çalışma Prensibi[2]

Termal transfer yazıcıların en büyük dezavantajı ilk yatırım maliyetinin yüksek olmasıdır. Bunlar, ofis yazıcıları gibi yüksek miktarlarda seri olarak üretilmeyen özel yazıcılar olduğu için fiyatları da daha yüksektir. Yüksek ilk yatırım maliyeti genellikle düşük birim etiket maliyetiyle dengelenir. Genel anlamda, eğer yüksek hacimli etiket üretimi gerekiyorsa termal transfer yazıcı en iyi ve uygun maliyetli seçenek olacaktır.

## 1.4 Termal baskı kafasının yapısı ve tarihsel gelişimi

### 1.4.1 Termal Baskı kafası hakkında Bugüne kadar yapılmış çalışmalar

İlk olarak 1967 de, W. H. Puterbaugh ve S. P. Emmons, “A New Printing Principle” adlı çalışmalarında bir termal yazıcının daha hızlı yazabilmesi için ne gibi bir düzenleme yapılabileceğini göstermişlerdir. Termal yazıcı başlığının kütlesinin küçük olması durumunda sistemin hızlı bir şekilde ısınıp soğuyacağını düşünerek silikon teknolojisi yardımıyla sistemin basitleştirilebileceğini öne sürmüşlerdir.[4]

1973 te T. R. Payne ve H. R. Plumlee katı durumlu termal yazıcıların kullanımıyla karakter yazım metodu önermişlerdir. Sistemde silikonun pozitif sıcaklık katsayısına ve düşük dirence sahip olması avantajından yararlanılmıştır. Fakat bu sistemin dezavantajı çok yüksek sıcaklıkların silikon yapısına zarar vermesidir.

Shibata vd. , 1976 yılında, “New Type Thermal Printing Head Using Thin Film” [5] adlı makalelerinde, tanımladıkları yeni ısı başlığını tanıtmışlardır. Isınma elementi olan Ta<sub>2</sub>N (tantal nitrit), 2 Ta<sub>2</sub>O<sub>5</sub> SiO<sub>2</sub> – (silikon dioksit-tantal oksit) çift tabakası ile kaplanmıştır. Böylece 2 SiO<sub>2</sub>, Ta<sub>2</sub>N deki oksidasyonu, Ta<sub>2</sub>O<sub>5</sub> ise sızdırmazlığı sayfa meydana gelebilecek aşınmayı engellemiştir. Ayrıca hesaplamalar sonucunda; termal zaman sabitinin, ısıyı izole eden tabaka kalınlığının değişimi ile kontrol edildiğini göstermişlerdir. Buradan cıvalı alüminyum tabakanın en iyi termal özelliklere sahip olduğunu bulmuşlardır. [1]

Shibata vd. , 1984 yılında, “Development of 16 dots/mm Thermal Printing Head” > adlı makalelerinde yüksek hıza sahip milimetrede 16 nokta içeren termal başlığın gelişimini tarif etmişlerdir. Başlıkta kıvrımlı direnç kullanmanın faydasını vurgulamışlardır. Ayrıca vuruş aralığı ile kıvrımlı şekle sahip ısıtıcı direncin yarattığı bozunmalar arasında ilişki olduğunu göstermişlerdir. [1]

Drees vd. , 1989 yılında, “Performances of Thermal, Transfer Printing and Their Improvement by Model Calculations” adlı makalelerinde termal başlıktan sayfaya harf basımında, başlıkta her bir vuruşta sıcaklık artışının benzer olması gerektiğini savunmuşlardır. Vuruşla sıcaklık artışı her bir vuruş için farklı seviyede olursa, baskı kalitesinin düşük olacağını, her bir vuruş için aynı seviyede sıcaklık artışı sağlandığında baskı kalitesinin artacağını göstermişlerdir. Webb ve Hann (1990), çalışmalarında, başlık yüzeyinde sıcaklık yayılımının, zamanla değişimini göstermişlerdir. Sıcaklık yayılımındaki düzensizliklerin renkli baskı da kaliteyi azalttığı sonucuna varmışlardır.[1]

Hodge ve Ross (1991), çalışmalarında, kütle transferi ile enerji girişi, vericideki boya kalınlığı, verici kalınlığı ve alıcı sıcaklığı arasındaki ilişkiyi araştırmışlardır. Sonuçta vericiden alıcıya boya transferinin difüzyonla gerçekleştiğine işaret etmişler, bunun alıcı sıcaklığı, verici kalınlığı gibi fiziksel faktörlere bağlı olduğunu göstermişlerdir.

Connolly (1994), “Optimization of the Thermal Performance of a Structure for Pulse-Count-Modulated Dye Diffusion Thermal Printing” adlı makalesinde, başlıktaki yalıtım tabakasının anahtar bir özelliğe sahip olduğunu, yalıtım tabakasının kalınlığının ve termal iletkenliğinin termal performansı etkilediğini göstermiştir.

Uyhan (1996), “The Temperature Distribution Produced by a Thermal Printer Head” adlı yüksek lisans tezinde bir boyutta tabakalardaki ısı iletimini ele almış, bundan yararlanarak iki boyutta altın levhaların ısı iletimi ve sıcaklık değişiminde etkilerini göstermiştir.[1]

Part of a series on the <b>History of printing</b>			
Woodblock printing	200		
Movable type	1040	Photostat and Rectigraph	1907
Printing press	c. 1440	Screen printing	1910
Etching	c. 1515	Spirit duplicator	1923
Mezzotint	1642	Xerography	1938
Aquatint	1772	Phototypesetting	1949
Lithography	1796	Inkjet printing	1951
Chromolithography	1837	Dye-sublimation	1957
Rotary press	1843	Dot matrix printing	1968
Hectograph	1869	Laser printing	1969
Offset printing	1875	<b>Thermal printing</b>	c. 1972
Hot metal typesetting	1884	3D printing	1984
Mimeograph	1886	Digital press	1993

Şekil 4: Yacız sistemlerinin Tarihsel Gelişimi

#### 1.4.2 Termal yazıcı kafasının yapısı

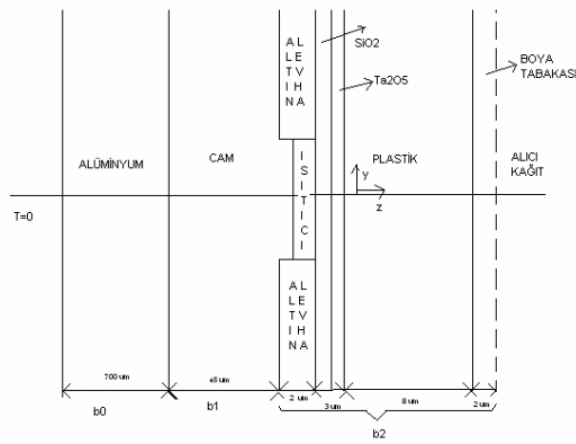
**Kalın Alüminyum Tabaka:** Başlıktaki ısıtıcının ürettiği ısı için bir ısı kanalı görevi görür. Akım kesildiğinde başlığın hızla soğuması gerekir, çünkü akım tekrar açıldığında bir sonraki noktanın üretilebilmesi için sistemin başlangıç sıcaklığına yakın olması gerekir. Bu olmazsa başlığa her defasında aynı miktarda enerji verilirse noktaların büyüklükleri artar. Pratikte pek çok ısıl başlıktan oluşan sistem birkaç saat çalıştırılırsa alüminyum tabakanın sol tarafına yerleştirilen soğutucu fanların kullanılmasına rağmen tabakanın sı- caklığı kademe kademe artar.

**Cam Tabaka:** Isıtıcı ile alüminyum tabaka arasında bulunur. Bu tabakanın kalınlığı sistemin işleyişi için önemlidir. Bu tabaka çok kalın ve düşük ısı iletkenliğine sahipse (iletkenliği az ise) ısıtıcıdan plastik tabakaya daha çok ısı geçişi olur. Bu durum plastik tabakada yüksek sıcaklığa neden olur ve akım kapatıldığında sistem hızlı biçimde soğumaz. Eğer cam tabaka çok ince ise plastik tabakaya çok az ısı geçişi olur. Bu da boyanın dağılım süresini artırarak yazım sürecinin yavaşlamasına neden olur.

Isıtıcı Tabaka: Cam tabaka ile plastik tabaka arasında bulunur. Isıtıcı eleman (direnç)  $Ta_2N$ 'den oluşur. Isıtıcı elemanın sağ tarafında koruyucu tabakalar vardır. Bunlar sırasıyla  $SiO_2$  ve  $Ta_2O_5$  'dir.  $SiO_2$  tabakası  $Ta_2N$  deki oksidasyonu engeller ve  $Ta_2O_5$  plastik tabakada oluşacak aşınmayı engeller. Çünkü ısının artmasıyla plastik tabaka ısıtıcıya doğru çekilir.  $Ta_2O_5$  aşınmaya karşı koyar.

Plastik Tabaka: Isıtıcı tabaka ile boya tabakası arasındadır. Isının boya tabakasına iletilmesinde yardımcıdır.

Boya Tabakası: Plastik tabaka nın sağındadır. Düşük sıcaklıkta erime özelliğine sahiptir. Isıtıcının açılmasıyla plastik tabakadan ısı iletimiyle boya tabakasının sıcaklığı artar, bunun sonucunda boya erir ve alıcı kâğıda yapışır.



Şekil 5:Termal Yazıcı kafasının yapısı

### 1.4.3 ENDÜSTRİYEL ÜRÜNLER

Günümüzde termal yazıcı alanında birçok ticari ürün bulunmaktadır. Başlıca firmalar seiko, fujitsu gibi firmalardır. Bu firmalara ait iki adet ürün Tablo-1 ve Tablo-2 de gösterilmiştir:



Şekil 6: LTP1245 Printer(SEIKO) [6]

### Ürün Özellikleri

Ürün	Ürün açıklaması	Ürün özellikleri
Model		LTP1245
Baskı	Methot	Direkt termal baskı
Algılama	Tepe sıcaklığı	Termistör aracılığı ile
	Ba yukarı algılama	Fotoğraf kesici aracılığı ile
	Kağıt algılama çıkışı	Fotoğraf kesici aracılığı ile
Güç kaynağı	İşlem voltu(Vdd)	4.5-5.5V
	İşlem voltu(Vp)	4.2V-8.5V
Tepe akımı	Tepe	2.7A(amper)
	Motor	0.46A(amper)
Ömür	Darbe aktivasyonu	100 milyon
	Maksimum baskı uzunluğu	50km
İşlem sıcaklığı		0-50(santigrad derece)
Boyutlar(WxDxH mm)		72x39.5x16
Ağırlık (g)		45g

Tablo 1: LPT1245 özellikleri[6]



Şekil 7: FTP-628MCL101(FUJITSU)/ÇİN[7]

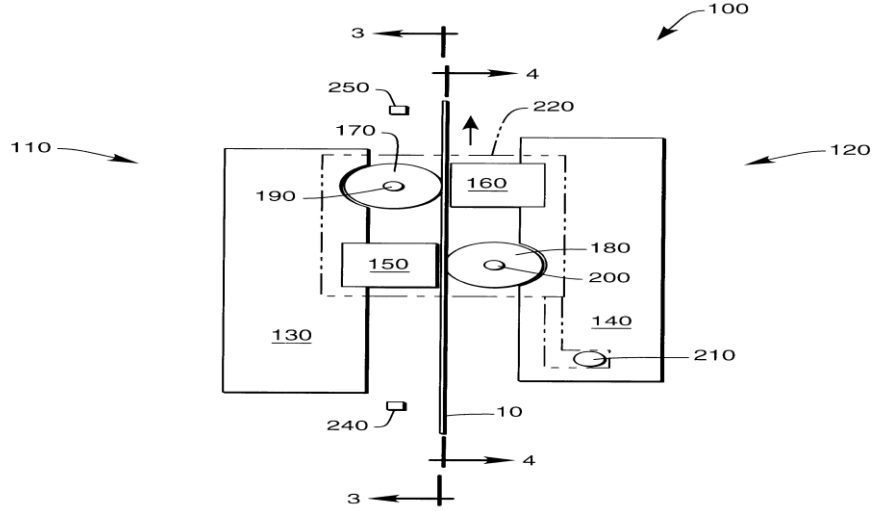


Ürün		Ürün Özellikleri
Ürün numarası		FTP-628MCL101/103
Baskı Metodu		Termal-çizgi nokta metodu
Nokta yapısı		384 nokta/çizgi
Nokta hattı(yatay)		0.125 mm(8 nokta/mm)-Nokta yoğunluğu
Nokta hattı(dikey)		0.125 mm(8 nokta/mm)-Çizgi besleme hattı
Etkili baskı alanı		48 mm
Kolonların sayısı		ANK 32 Kolon/Çizgi (maksimum 12*24 nokta yazısı)
Kağıt genişliği		58 mm+0 -1
Kağıt kalınlığı		60 ‘dan100 $\mu$ m (bu aralıktaki bazı kağıtlar kağıdın karakteristiğinden dolayı kullanılamayabilir.
Baskı hızı		Maksimum 80mm/sn. (640 nokta çizgi/sn.) 8.5V’ta
Karakter tipi		Alfanumerik, katakana: 159 tip Uluslararası ve özel karakterler: 195 tip JIS Kanji seviye 1, seviye 2, non-Kanji (sadece Kanji CG montajlı olduğunda destekliyor): yaklaşık 6800 tip
Karakter, Boyutlar(HxW), Kolonların sayısı		12 $\times$ 24 nokta, (1.5 $\times$ 3.0mm), 32 kolon:ANK 24 $\times$ 24 nokta, (3.0 $\times$ 3.0mm), 16 kolon:ANK, Kanji 8 $\times$ 16 nokta, (1.0 $\times$ 2.0 mm), 48 kolon: ANK 16 $\times$ 16 nokta, (2.0 $\times$ 2.0 mm), 24 kolon: ANK, Kanji
Haberleşme arayüzü		RS232C
Besleme Voltajı	Termal Başlık	4.2-8.5VDC,0.87A ortalama değer
	Motor	4.2-8.5VDC ,1A maksimum
	Lojik Giriş	3-5-5VDC,1A maksimum
Ölçüler	Yazıcı mekanizması	70.2x33x15.5(WxDxH)
	Arayüz bord	69.3x52x15.2(WXxH)
Ağırlık	Yazıcı Mekanizması	42kg
	Arayüz bordu	20kg
Çalışma Sıcaklığı	Ç.Sıcaklığı	0-50(cantıgrad derece)
	Ç.Nemi	20-80%
	Sıcaklık aralığı	-20-+60(cantıgrad derece)
	Nem aralığı	5-90%

**Tablo 2: FTP-628MCL101(FUJITSU) özellikleri[7]**

#### 1.4.4 PATENT

Termal printer Herman Epstein ve Tarbuck Robert R tarafından 22 Aralık 1959 yılında patenti alınmıştır. Haksahibi ise Burroughs Corp tur.Direkt termal yazıcı kafası ise Genzi Oshino, ve Masaaki Yoshikawa dır. Orijinal hak sahibi ise Tohoku Ricoh, Co., Ltd adlı bir Japonya şirketine aittir. Termal yazıcı kafa 8 Eylül 1997 yılında icat edilmiştir. Ayrıca direkt termal yazıcı çeşitlerinden bir tanesi ise John Long, Richard B. Moreland tarafından 18 Aralık 2001 de keşfedilmiştir. [8]



Şekil 8: Bir direkt termal yazıcı teknik resmi[9]

## BÖLÜM 2. PROBLEM TANIMLANMASI

### 2.1 PROBLEM TANIMI VE TASARIM ÖZELLİKLERİ

Bu proje de termal baskı yöntemi kullanılarak termal etiket makinesi yapılacaktır. Projenin gerçekleştirilmesi için birkaç problemin aşılması gerekiyor. Öncelikle uygun termal yazıcı kafası seçilmelidir. Seçilen termal kafaya uygun arayüz kartı tasarlanmalıdır. Bu kartın tasarlanması için endüstriyel örnekleri incelenip tasarımın yapılıp yapılamayacağı kararlaştırılmalıdır. Eğer kontrol kart tasarımı için gerekli devre elemanlarının temin edilemeyeceğine karar verilirse hazır kontrol kartı veya termal kafa-kontrol kartının bir bütün halinde olduğu panel ve kiosk printer modüllerinden biri temin edilmelidir. Bütün malzemelerin temin edildiğini varsayarsak geriye microcontrolör devresinin tasarımı kalmakta. Bunun için hangi programlama dilinin seçileceğine karar verilecektir. Karakter LCD veya grafik LCD kullanılarak görüntülenen metni rs-232 haberleşme protokolü ile yazıcıya göndererek çıktı alınması hedeflenmektedir. Tam bu safhada birkaç problemin çözülmesi gerekmektedir. İlk olarak alınan printer mekanizmalarının command manual rehberine göre mikrokontrolör kodu yazılmalıdır. Yazıcı mikroişlemci haberleşme mantığı öğrenilmeli ve projeye uygulanmalıdır.

#### 2.1.1 Proje Örnekleri

##### PROJE ADI: Termal Yazıcı Ve Arduino Kullanarak Baskı Yapma



Şekil 9:Termal yazıcının arduino ile bağlantısı ve endüstriyel ürün

**Problemin Tanımı:** Baskı makineleri birçok alanda yaygın olarak kullanılmaktadır. Bu makineler bankamatiklerde, ürün etiketlerinde, reklam firmalarında fax makinelerinde veya kişisel olarak vb alanlarda kullanılır. Birçok çeşidi bulunan bu makinelerin baskıyı yapma metoduna göre birçok çeşidi bulunmaktadır. Bu metotlardan birisi olan termal baskı yazıcı genellikle kullanılan kâğıdın önemsiz olduğu ve boyutlarının küçük olduğu alanlarda kullanılır.

Bu projede bilgisayar üzerinden gönderilen veriyi arduino işlemci üzerinden termal yazıcı ile haberleştirip çıktı alınması sağlanacaktır.

### **Proje Gereksinimleri**

- Termal yöntem ile baskılama yapan bir yazıcı temin etmek.
- Bilgisayara gelen veriyi yorumlayıp yazıcıya aktarmak için gerekli olan arduino işlemciyi temin etmek
- Bilgisayar üzerinde bir arayüz tasarlamak ve bu arayüz üzerinden çıktı almak
- Çıktıda kullanılan yazının font ayarlarını ,yazının sağ ,sol,merkez konum ayarlarını ve mesafeleri ayarlayabilmelidir.
- Bitmap formatın da resim çıktı alabilmelidir.
- QR kod ve barkod baskısı yapılabilmelidir.

### **Model Özellikleri**

- 2.25'' kağıta baskı yapabilmektedir.
- Büyük,orta,küçük boyutlarda baskı yapabilir.
- Yazıların kalınlığını ayarlama,altını çizme ve ters çevirme özelliğine sahiptir.
- Değişken satır aralığı girilebilmektedir.
- Yazının kağıdın sağ,sol,merkez konumlamasını yapabilir.
- UPC A, UPC E, EAN13, EAN8, CODE39, I25, CODEBAR, CODE93, CODE128, CODE11 and MSI formatlarında kod basabilmektedir.
- Bitmap formatında resim basabilmektedir.
- İşlemci giriş değerleri 3.3V-5V,1.5A(Arduino microcontrol input power),termal yazıcı giriş değerleri 5V-9V 2A dir.Bu güç kağıdın ısıtılması için optimum değerdir.

## BÖLÜM 3. PROJE PLANI VE YÖNETİM

Bu bölümde projenin işlem süreci ve görev dağılımından bahsedilecektir. Projemiz direkt termal yazma yöntemi ile mikrokontrolör ile kontrol edilen etiket makinesidir. Bu işlemi yaparken ilk etapta PIC16F877A ile tuş takımı kullanılarak panel termal yazıcı ile çıktı alınması amaçlanmıştır.

### 3.1 PROJENİN ORGANİZASYONU

- Maksut KAYA (TM1)
- Ahmet YILDIZ (TM2)
- Hüseyin TEKECİ(TM3)

### 3.2 GÖREVLER VE AÇIKLAMALAR

#### 3.2.1 Literatur Taraması

##### 1.aEtiket/Fatura makinesi nedir?

- Etiket makinesinin tarihçesi(TM1,TM2,TM3)
- Etiket makinesi baskı yöntemleri(TM1,TM2,TM3)

##### 1.bTermal baskı yöntemleri

- Direkt termal baskı yöntemi(TM2,TM3)
- Transfer termal baskı yöntemi(TM1,TM2)

##### 1.c Termal baskı mekanizmaları

- Termal yazıcı kafaları(TM2)
- Termal kontrol kartları(TM3)
- Termal panel yazıcılar (TM1)

#### 3.2.2 Kavramsal Tasarım

Piyasada bulunan termal yazıcı kafalar ,kontrol kartları,panel yazıcılar incelenecek ve bu anlamda projemize uygun komponentler temin edilip gerçekleştirilecektir.

##### 2.a Yazıcı seçimi(TM2,TM3)

- Nokta vuruşlu termal yazıcı
- Direkt termal yazıcı
- Termal transfer baskı yazıcı

##### 2.b Kontrolör Seçimi(TM1,TM2)

- Microchip (16F877A)
- STMicroelectronics (STM32F4)
- Arduino(UNOR3)

##### 2.c Diğer Komponentler(TM1,TM3)

- Karakter veya grafik lcd seçimi
- Güç kaynağının belirlenmesi
- Tuş takımının seçimi

## **2.d Yazılım(TM1,TM3)**

- CCS
- ARM
- Arduino

## **3.2.3 Ayrıntılı Tasarım ve Üretim(TM1,TM2,TM3)**

- Planlanan ürünlerin firmalardan temin edilmesi(TM1,TM2,TM3)
- Kontrol kodunun yazılması(TM1)
- Haberleşme kodunun yazılması(TM1)
- Yapılacak arayüzün hazırlanması(TM2,TM3)
- Baskı devre şemalarının çizilip plakete basılması(TM1,TM2,TM3)
- Devre komponentlerinin devreye lehimlenmesi(TM1,TM2,TM3)
- Diğer malzemeler ile ürünün genel montajının yapılması(TM1,TM2,TM3)

## **3.2.4 Dökümantasyonlar(TM1,TM2,TM3)**

- Literatür araştırma raporu(TM2,TM3)
- Problem tanımlama raporu(TM2,TM3)
- Ön tasarım raporu(TM2,TM3)
- Detay tasarım raporu(TM2,TM3)
- Sonuç raporu(TM1,TM2,TM3)

## **3.3 ZAMAN TABLOSU**

<b>Görev No</b>	<b>HAFTA</b>
<b>1.a</b>	<b>1.hafta</b>
<b>1.b</b>	<b>1.hafta</b>
<b>1.c</b>	<b>1.hafta</b>
<b>2.a</b>	<b>2-4.hafta</b>
<b>2.b</b>	<b>4.hafta</b>
<b>2.c</b>	<b>5.hafta</b>
<b>2.d</b>	<b>6.hafta</b>
<b>3</b>	<b>7-12.hafta</b>
<b>4</b>	<b>13-14.hafta</b>

**Tablo 3: Zaman Tablosu**

## BÖLÜM 4. KAVRAMSAL TASARIM

Projemiz de ilk olarak mikroişlemci kullanılarak çıktı alınması hedeflenmekte. Bu amaçla kontrol kartı tasarımı planlanan proje zamanında yetişmeyeceği düşünüldüğünden ve devrede kullanılan elemanların elektronik piyasada bulunamadığından iptal edildi. Kontrol kartı, yazıcı kafa ve rulo kağıt tutucu haznenin bir arada bulunduğu panel termal yazıcı modelleri proje için uygun olacağı düşünüldü.

### 4.1 KAVRAMSAL TASARIM VE SUNUM

Projede kavramsal tasarımın gelişimi en baştan ele alınırsa termal yazma yöntemlerinden hangisinin bizim projemize uygun olup olmadığına karar verilmesine dönmek gerekir. Daha önceki raporlarda bahsettiğimiz gibi termal ribonlu etiketler termal etiketler için daha kullanışlı ve uzun ömürlü olmasına karşın pahalı ve yapımı zor olduğu için seçeneklerden çıkartılmıştır. Diğer seçeneğimiz ribonsuz direkt termal baskı yönteminin kullanılmasına karar verilmiştir.

Termal Baskı yöntemleri	Avantajlar	Dezavantajlar
Direkt Termal Baskı Yöntemi	Ribon masrafı yoktur, yakarak basar. Termal etiket kullanılır Birçok proje uygulaması bulunmakta. Fiyatı transfer termal yazıcıya göre daha ekonomiktir.	Etiket ömrü kısadır. Kısa sürede tüketilecek etiketler için uygundur.
Transfer Baskı Yöntemi	Vellum etikete veya plastik, dokuma gibi değişik etiketlere baskı yapılabilir. Etiket ömrü çok uzundur. Bazı modellerinde her iki türlü de baskı yapabilme imkanı bulunmaktadır.	Ribon masrafı vardır. Yazıcı fiyatı direk termal barkod yazıcıya göre pahalıdır.

**Tablo 4:Termal baskı yöntemleri karşılaştırılması**

#### 4.1.1 Panel Yazıcılar

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

Daha önceki raporlarda kontrol kartı termal baskı kafası bir arada olan ürünler kullanılacağı belirtilmişti. Aşağıdaki tabloda bu amaç doğrultusunda çeşitli panel yazıcılara yer verilmiştir.

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

Tablo 5: Çeşitli Panel Yazıcıların Karşılaştırılması



Şekil 10: CUSTOM PLUS II ve WOOSIM PORTI-P40/PP40



Mikrokontrolör seçimine gelecek olursak elimizde birkaç seçenek bulunmaktadır. Mikrochip firmasının PIC16f877A veya 18F serisinden bir kontrolör seçilebilir. Diğer taraftan dokunmatik paneli bulunduğu için STMicroelectronics firmasının STM32F429/439, kullanım kolaylığı bakımından Arduino uno R3 modelleri seçeneklerimizde bulunmaktadır.

## 4.2 KONSEPT SÜRECİ

Yukarıda belirttiğimiz konsept seçeneklerinin değerlendirilmesi bu bölümde ele alındı. Kriterler ve önem ağırlıkları belirlendi. Pugh karar matris yöntemi kullanılarak seçenekler elendi.

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

**Tablo 6: Pugh karar Tablosu**

1. WOOSIM PORTI-P40/PP40
2. CUSTOM PLUS2
3. Mini Thermal Receipt Printer
4. 58mm thermal printer

Tabloda belirttiğimiz kriterlere göre yaptığımız puanlamaya göre 2 numaralı Custom Plus2 modeli seçilmiştir. Ayrıca puanlama tablosu kurulmadan projeye için grup üyelerinin diğerlerine göre içeriğine daha hakim olduğu PIC16F877A kullanılmasına karar verilmiştir.

Projede yazılacak metnin görüntüleneceği görsel elemanın grafik LCD mi yoksa karakter LCD mi olacağı belirlendi. İlk etapta grafik lcd de karakter belirlemenin diğerine göre zor olacağı düşünüldüğünden karakter LCD seçilmiştir.

Alınan panel yazıcı da Rs-232 girişi bulunmaktadır. Bu nedenle mikroişlemcinin TTL çıkışını RS-232 ye çevirebilmek için MAX232 devresi kurulacaktır.

### 4.3 En İyi Kavramsal Tasarım

Birkaç hafta süren araştırmalar ve literatür taramaları sonucu projede kullanılacak tüm elemanlar belirlendi. Bu seçimler yapılırken birçok kriter dikkat edildi. Kısaca maddeler halinde bu seçilen konseptin özelliklerinden bahsedeceğiz.

#### 4.3.1 Tasarımda Kullanılacak olan malzemeler ve özellikleri

##### 4.3.1.1 Projedeki Ana Elemanlar

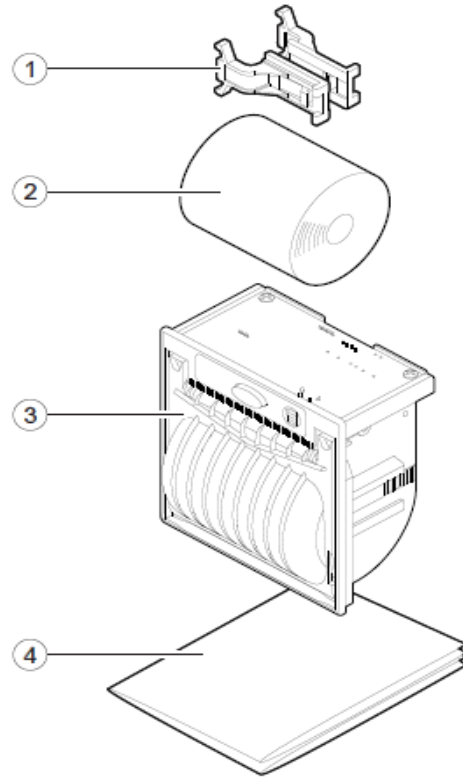
- CUSTOM PLUS II



Şekil 10:Custom Plus II termal yazıcı

- 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
- Codici a barre: 1D,2D e QR CODE
- Font: European, International, Portuguese and Nordic, Chinese (GB18030) and Cyrillic
- Sensors: head temperature, paper presence, door open (opt.)
- Approved for UL, CE, FCC
- Outdoor applications:-20 / +70°C working temperature

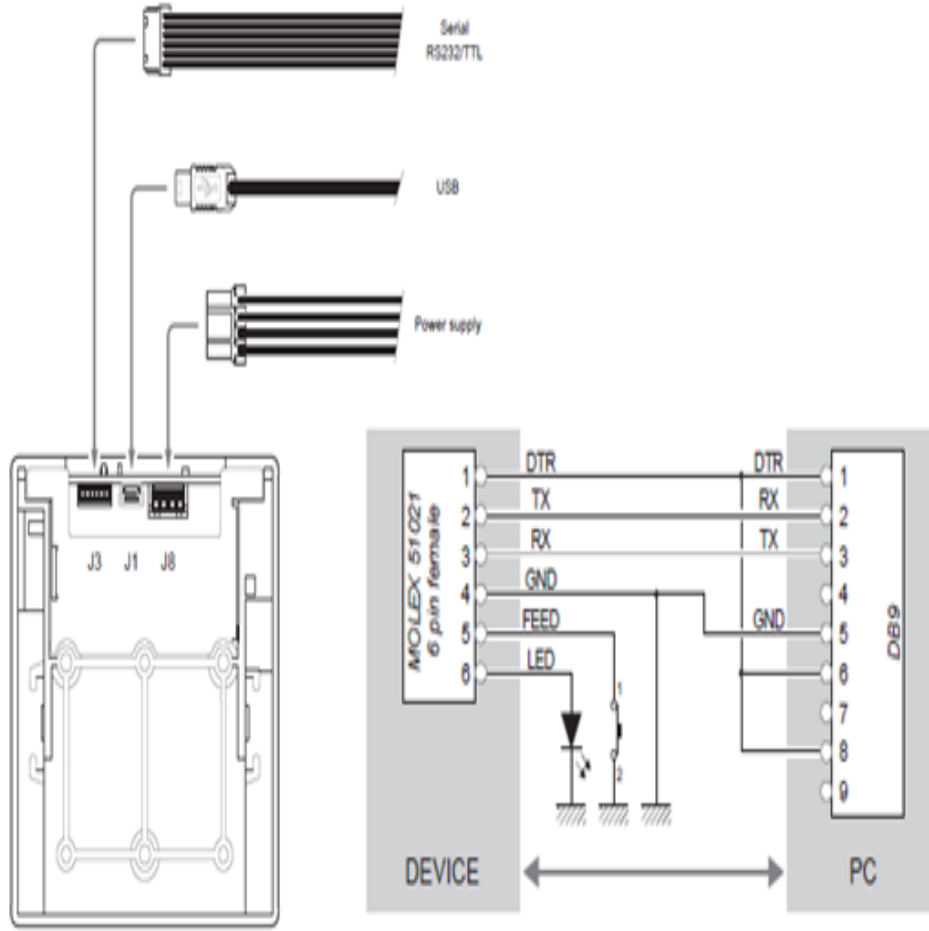
1. Fixing clips (no. 2)
2. Paper roll
3. Device
4. Installation instructions



**Şekil 11: Termal yazıcı bağlantı şeması**

CUSTOM PLUS II panel yazıcı RS232 veya TTL ile haberleşmektedir. Aşağıdaki şekilde yazıcının haberleşme ve güç bağlantılarının ne şekilde yapılacağı gösterilmiştir.

RS232 bağlantısının pinout şekli aşağıda gösterilmiştir. Bağlantı için 6 pinli MOLEX 51021 kullanılmıştır.



Şekil 12:Termal yazıcı haberleşme şeması

PRINTER EMULATION

Available emulations for the device:

PLUS<sup>D</sup>

CUSTOMPOS

RS232 BAUD RATE

Communication speed of the serial interface:

115200 19200 2400

57600 9600<sup>D</sup> 1200

38400 4800

NOTE: Parameter valid only with serial interface.

RS232 DATA LENGTH

Number of bit used for characters encoding:

7 bits/car

8 bits/car<sup>D</sup>

NOTE: Parameter valid only with serial interface.

RS232 PARITY

Bit for the parity control of the serial interface:

None<sup>D</sup> = parity bit omitted

Even = even value for parity bit

Odd = odd value for parity bit

NOTE: Parameter valid only with serial interface.

RS232 HANDSHAKING

Handshaking:

XON/XOFF<sup>D</sup> = software flow control

Hardware = hardware flow control (CTS/RTS)

NOTES:

Parameter valid only with serial interface.

When the receive buffer is full, if handshaking is set to XON/XOFF, the device sends the XOFF (0x13) on the serial port. When the receive buffer has cleared once again, if handshaking is set to XON/XOFF, the device sends the XON (0x11) on the serial port.

BUSY CONDITION

Activation mode for Busy signal:

OffLine/ RXFull = Busy signal is activated when the device is both in OffLine status and the buffer is full

RXFull<sup>D</sup> = Busy signal is activated when the buffer is full

NOTE: Parameter valid only with serial interface.

Yukarıdaki şekilde yazıcının haberleşme protokolü hakkında bilgiler verilmiştir. Baudrate hızı 115200 ile 4800baud arasında değişmekte ve fabrika ayarlı olarak 9600 baud hızında gelmektedir. Data uzunluğu 7 ve 8 bit uzunluğundadır.

- **2x16 LCD Ekran**



**Şekil 13: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 arasındır.

LCD Pin	Sembol	Bağlantı
1	V <sub>ss</sub>	GND
2	V <sub>dd</sub>	+5V
3	V <sub>o</sub>	Kontrast ayarı
4	RS	Register seçme pini
5	R/W	Data read/write pini.
6	E	Enable pini
7	DB0	Data Bus 0
8	DB1	Data Bus 1
9	DB2	Data Bus 2
10	DB3	Data Bus 3
11	DB4	Data Bus 4
12	DB5	Data Bus 5
13	DB6	Data Bus 6
14	DB7	Data Bus 7
15	A	LED arka ışık +5V
16	K	LED arka ışık gnd

- **4x4MembranTuş takımı(Keypad)**

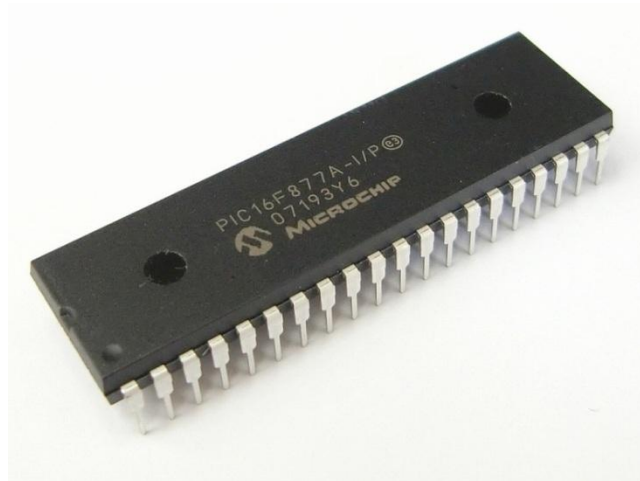


Şekil 14:4x4 tuş takımı

Kullanıcı girişleri için temel bir 16 butonlu tuş takımı. Butonlar matrix formatına uygun bir şekilde yerleştirilmiştir. Bu mikro denetleyicinin 16 butondan hangisinin basıldığını algılaması için 8 çıkış pinini ‘taramasına’ izin vermektedir. 16 butonlu membran keypad monteli bir şekilde kullanıcılara sunulur.

- **PIC16F877A**

PIC16F877 'nin 40 pininden 33 tanesi I/O ( input/output - giriş/çıkış ) pinleridir. PIC16F877; 6 bitlik A portu, her biri 8 bitlik B,C ve D portları ve 3 bitlik E portu olmak üzere 5 porta sahiptir. I/O pinlerinin gerekli konfigürasyonlar yapılarak başka amaçlarla kullanılması da mümkündür.



Şekil 15 PIC16F877A entegresi

## **Temel Özellikleri**

- \*İşlem hızı DC-20 MHz dir. (bir komut DC-200 ns hızında çalışmaktadır.)
- \*Dallanma komutları iki çevrim (cycle) sürede, bunların dışındaki komutlar ise bir çevrimlik sürede uygulanır.
- \*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.
- \*8 Kword 'lük Flash ROM programlama belleği 1000000 kez programlanabilir.
- \*CPU azaltılmış komut setine sahiptir.
- \*Statik RAM üzerinde 32 adet özel işlem kaydedicisi (SFR - Special Function Register) vardır.
- \*RISC temeline dayanır.
- \*CPU azaltılmış komut setine sahiptir.
- \*Öğrenilecek gereken her biri 14 bitlik 35 komut vardır.
- \*Pin çıkışları PIC 16C73B/74B/76 ve 77 ile uyumludur.
- \* 8 bitlik veri yolu (databus) vardır.
- \*14 kaynaktan kesme yapabilir.
- \*PIC16F877 dolaylı ve göreceli adresleme yapabilme özelliğine sahiptir.
- \*CMOSFlash EEPROM teknoloji ile düşük güçle yüksek hıza erişebilir.
- \*Enerji verildiğinde sistemi resetleme özelliği (Power-on Reset),
- \*Power-up zamanlayıcı (Power-up Timer)
- \*Osilatör başlatma zamanlayıcısı (Osilatör Start-up Timer)
- \*Özel tip zamanlayıcı (Watch-dog Timer), devre içi RC osilatör
- \*Hata ayıklamada kullanılanabilecek modül (devre içi Debugger)
- \*Seçimli osilatör özellikleri
- \*Statik tasarım
- \*Enerji tasarrufu sağlayan uyku modu ( Sleep Mode) özelliği vardır.
- \*Programla kod güvenliği sağlanabilir.
- \*Düşük gerilimli programalama özelliğine sahiptir.
- \*Sadece 5 V giriş ile devre içi seri programlanabilir.
- \*2 pinle programlanabilir.
- \*Program belleğine okuma/yazma özelliği ile erişilebilir.
- \*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.
- \*Devre içi iki pin ile hata ayıklama özelliği vardır.



## Çevresel Özellikleri

\*TMR0: 8 bitlik zamanlayıcı, 8 bit önbölücülü

\*TMR1: Önbölücülü, 16 bit zamanlayıcı, uyuma modundayken dış kristal zamanlayıcıdan kontrolü arttırılabilir.

\*TMR2: 8 bitlik zamanlayıcı, hem önbölücü hem de sonbölücü sabiti

\*İki Capture / Compare / PWM modülü

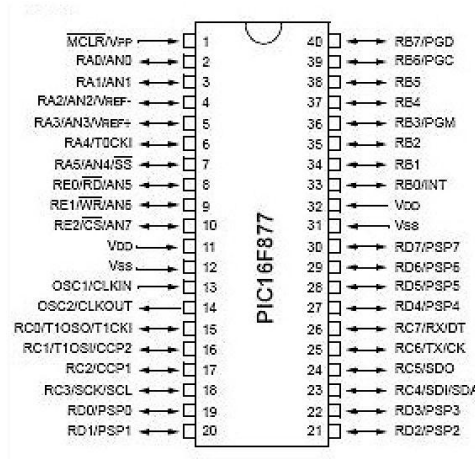
\*10 bit çok kanallı A/D çevirici

\*Senkron seri port (SSP), SPI (Master mod) ve I 2 C (Master Slave) ile birlikte

\*Paralel Slave Port, 8 bit genişlikte ve dış RD, WR, CS kontrolleri

\*USART/SCI, 9 bit adres yakalamalı

\*BOR Reset (Brown Out Reset) özelliği



Şekil 16: PIC16F877A bağlantı şeması

### • MAX232 DEVRESİ

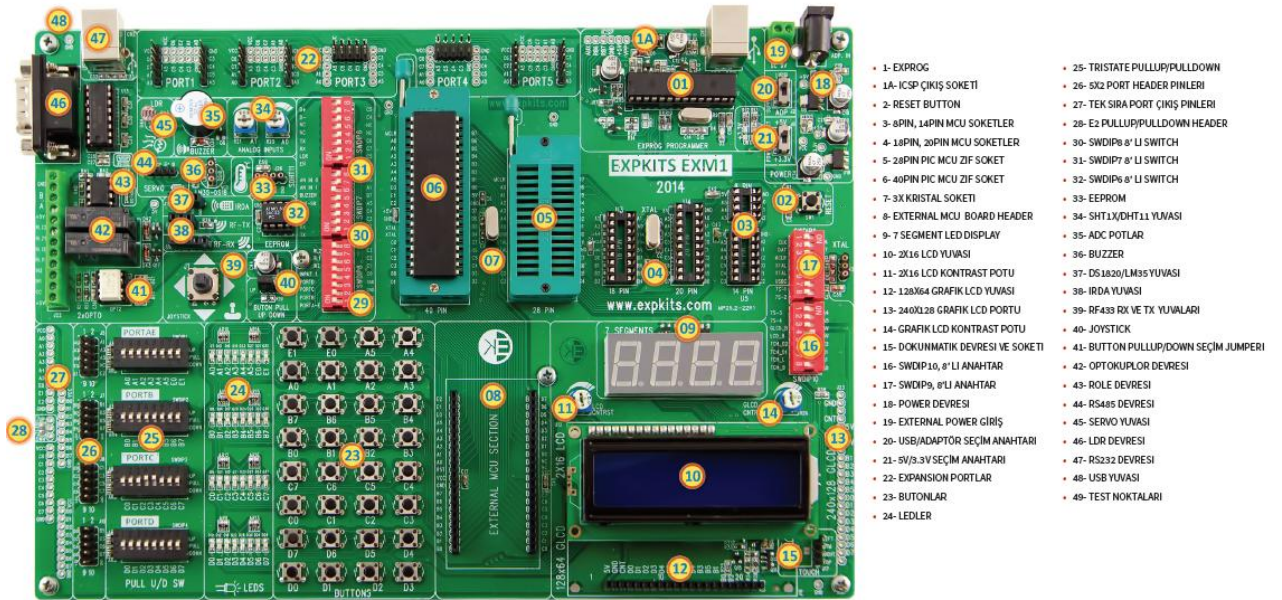
Birçok mikrodenetleyici 5 volt ile çalışır. Bu mikrodenetleyicilerin high Kabul ettiği değer 5v, low Kabul ettikleri değer ise 0 volt yani gnddir. Buna TTL seviye denir. Bilgisayar seri portu için high +12 V low -12 V civarındır. Bu durumda bilgisayar ile mcu devremizi direkt bağlamamız uygun değildir.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



Şekil 17 :Max232 entegresi

#### 4.3.1.2 Projedeki Yardımcı Elemanlar

- EXP KITS EXM1



Şekil 18:Mikroişlemci deneme kiti

EXM1 geliştirme kiti, mikrokontrolörlerin öğrenilmesi için tasarlanmış bir geliştirme platformudur.

4 Temel parçadan oluşmaktadır.

- MCU soketleri
- USB programlayıcı
- Çeşitli devre parçaları
- Ek bağlantı soketleri.

EXM1 mikrokontrolcü desteği 2 gruptadır;

Birinci grupta 8 bit Microchip 10Fxxx, 12Fxxx, 16Fxxx ve 18Fxxx ailesinden 40, 28, 20, 18, 14 ve 8 pinli PIC mikro kontrolörleri bulunmaktadır.

İkinci grupta “External mcu section” kısmı üzerinden takılabilecek diğer mcuları kapsar. Bu kit destek verdiği/vereceği mcular çok geniştir ve değişik firmaların ürünlerini kapsar. Tasarımından gelen bu yapısı sayesinde projelerin büyük bir çoğunluğuna destek verir.

Ayrıca harici devrelerin bağlanabileceği portlar bulunmaktadır. Expkits bu portlara bağlanacak devreler ile kitin kullanım sahasını geliştirecektir. Kartın üzerinde Expkits tarafından geliştirilmiş USB2.0 programlayıcı ile çok hızlı ve güvenli bir şekilde 40, 28, 20, 18, 14 ve 8 pinli Microchip PIC mikrokontrolcileri programlana bilinir.

Özellikler;

- Microchip 8 bit PIC mikro kontrolör desteği.
- Harici mcu soketi ile farklı firmalardan mikro kontrolörlere destek. (External mcu Section)
- 40 ve 28 pin ZIF soket desteği.
- 20, 18, 14 ve 8 pin entegre soket desteği.
- 3 adet kristal soketi.
- ExpKits Exprog USB 2.0 programlayıcısı ile picleri soketlerinden ayırmadan programlayabilme.
- ICSP modunda harici devrelerdeki picleri programlayabilme.
- 5 adet harici devre çıkış portu. Bu port yapısına uygun harici devre desteği verilecektir.
- 2x16 LCD desteği ve kontrast potu
- 128x64 GLCD desteği
- 240x128 GLCD desteği
- GLCD kontrast potu.
- Touch için sürücü devresi.
- 4 Adet 7 segment display
- 8’ li dipswitchler.Çeşitli küçük devrelerin ana devreye bağlantılarının yapılmasını sağlar.
- Mini joystick.
- USB haberleşme devresi.
- RS232 haberleşme devresi.
- RS485 haberleşme devresi.
- SHT1x Sıcaklık ve nem ölçme devresi.
- DHT11 Sıcaklık ve nem ölçme devresi.
- DS1820 ile sıcaklık ölçme desteği.
- LM35 ile analog sıcaklık ölçme desteği.
- Analog sinyal ölçüm için 2 adet pot .
- Buzzer devresi.
- 2 Adet 220V 3A röle çıkış devresi.

- 2 Adet dijital input devresi. ( 2x Opto input )
- Servo motor sürme desteği.
- IRDA desteği.
- RF-RX modül desteği.
- RF-TX Modül desteği.
- Fotosel LDR Devresi.
- I2C Seri EEPROM.
- USB Beslemesinden Çalışma. USB de iken harici kaynak gerekmez.
- Harici Güç Kaynağı ile Çalışma. (Maximum 9V DC Adaptör bağlanması normaldir.)
- USB Yada Harici Güç Seçim Soketi.
- 5V ve 3.3V voltaj devresi ve çalışma voltajı seçim soketi.
- Tüm pinler için 4 adet 5x2 IDC çıkış soketleri.
- Tüm pinler için tek sıra pin header çıkış soketi.
- 8' li Tristate dip switchler ile her pini ayrı ayrı Pull-Up ve Pull-Down yapabilme imkânı.
- Her pine bağlı 32 adet LED ve port olarak led gruplarını kontrol edebilme.
- 32 buton ile tüm mcu pinlerini kontrol edebilme.
- Butonların 1KR üzerinden 5V veya GND olarak seviye tanımlayabilme.
- Reset devresi.
- Opsiyonel pleksiglas altlık ile kitin korunmasının artırılması.
- 1. Sınıf kalitede PCB ve Smd malzemeler kullanılarak üretilmiştir.

#### **4.3.1.3 Projedeki Diğer Elemanlar**

- 20MHz Kristal osilatör
- 4x 1 mikroyarad elektrolit kondansatör
- 2x 22 pFarad kondansatör
- Erkek ve dişi headerlar
- 50 Kohm potansiyometre
- Klemens ve jumperlar
- 5v 5A metal kasa adaptör

## 4.3.2 TERMAL YAZICIDA KULLANILAN KOMUTLAR VE İLK PROGRAM

### 4.3.2.1 YAZICININ PROGRAMLAMA MANTIĞI

#### 0X02(Double height printing)

Valid for	mPLUS2 PLUS II (all models) PLUS2
[Format]	ASCII            STX Hex              02 Decimal                  2
[Range]	
[Description]	The printer prints in double height format
[Notes]	<ul style="list-style-type: none"> <li>• The commands from 0x00 to 0x09 do not cancel the print buffer.</li> <li>• The commands which modify the direction of the characters are only active at the beginning of the line.</li> </ul>
[Default]	Setting in option register by means of front keys.
[Reference]	0x00, 0x01, 0x03, 0x04, 0x1B 0x4D
[Example]	

#### 0x1D 0x57 <GS W> (Prints n byte of a 204 dpi graphic line)

Valid for	mPLUS2 PLUS II (all models) PLUS2
[Format]	ASCII GS W n d1...dn Hex 1D 57 n d1...dn Decimal 29 87 n d1...dn
[Range]	$1 \leq n \leq 48$ $0 \leq d1...dn \leq 255$
[Description]	Print n byte of a 200 dpi graphic line where: n specifies the number of byte to print; d1...dn specify the bytes to print.
[Notes]	<ul style="list-style-type: none"> <li>• If the bit image data input exceeds the number of dots to be printed on a line, the excess data are processed as printable characters.</li> <li>• d indicates the bit image data. Set a corresponding bit to 1 to print a dot, or 0 to not print the dot.</li> <li>• This command is not affected by the emphasized, double-strike, underline (etc.) print modes and the upside-down mode.</li> </ul>
[Default]	
[Reference]	
[Example]	For printing 12 bytes the command sequence is: 0x1D 0x57 0x0C 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00

**0x0B <VT>(Forward feeds (n) lines)**

Valid for	mPLUS2 PLUS II (all models) PLUS2		
[Format]	ASCII	(n)	VT
	Hex	(n)	0B
	Decimal	(n)	11
[Range]			
[Description]	Carries out the number of line feeds specied in (n).		
[Notes]	<ul style="list-style-type: none"> <li>• The number must be ASCII and between 0 and 9 (when n=0 the command is ignored)</li> <li>• This command clears the line buffer.</li> </ul>		
[Default]			
[Reference]	0x0A		
[Example]	To forward feed fast, 5 lines at a time: 0x35 0x0B (or 5 and the command 0x0B)		

**0x1D 0x24 <GS \$>Set absolute shift into a graphic line**

Valid for	mPLUS2 PLUS II (all models) PLUS2				
[Format]	ASCII	GS	\$	n	
	Hex	1D	24	n	
	Decimal		29	36	n
[Range]	$0 \leq n \leq 47$				
[Description]	Set the print beginning position into a graphic line based on the current value of n that indicate the byte number of shift from left margin.				
[Notes]	Settings outside the specied printable area are ignored.				
[Default]					
[Reference]					
[Example]					

**0x1D 0x55(Resets the printer parameters to default)**

Valid for	mPLUS2 PLUS II (all models) PLUS2		
[Format]	ASCII	GS	U
	Hex	1D	55
	Decimal	29	85
[Range]			
[Description]	Resets the printer parameters to the default con		
[Notes]	After executing this command the printer is initia		
[Default]			
[Reference]			
[Example]			

Valid for	mPLUS2 PLUS II (all models) PLUS2				
[Format]	ASCII	GS	W	n	d1...dn
	Hex	1D	57	n	d1...dn
	Decimal	29	87	n	d1...dn
[Range]	$1 \leq n \leq 48$ $0 \leq d1 \dots dn \leq 255$				
[Description]	Print n byte of a 200 dpi graphic line where: n species the number of byte to print; d1...dn specify the bytes to print.				
[Notes]	<ul style="list-style-type: none"> <li>• If the bit image data input exceeds the number of dots to be printed on a line, the excess data are processed as printable characters.</li> <li>• d indicates the bit image data. Set a corresponding bit to 1 to print a dot, or to 0 to not print the dot.</li> <li>• This command is not affected by the emphasized, double-strike, underline modes and the upside-down mode.</li> </ul>				
(etc.) print					
[Default]					
[Reference]					
[Example]	For printing 12 bytes the command sequence is: 0x1D 0x57 0x0C 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00 0xFF 0x00				

Valid for	PLUS II (all models)			
	PLUS2			
[Format]	ASCII	(dd)	ESC	a
	Hex	(dd)	1B	61
	Decimal	(dd)	27	97
[Range]				
[Description]	By using (dd) parameters it's possible to select the dot line number between one print line and another.			
[Notes]	(dd) are two ASCII characters (selected between '0', '1' ... '9', 'A', 'B' ... 'F') which identifies number from 0 to 127 in hexadecimal form and corresponds to the number of dot lines between one print line and another. The acceptable range is from 0x00 to 0x7F.			
[Default]	0			
[Reference]				
[Example]				

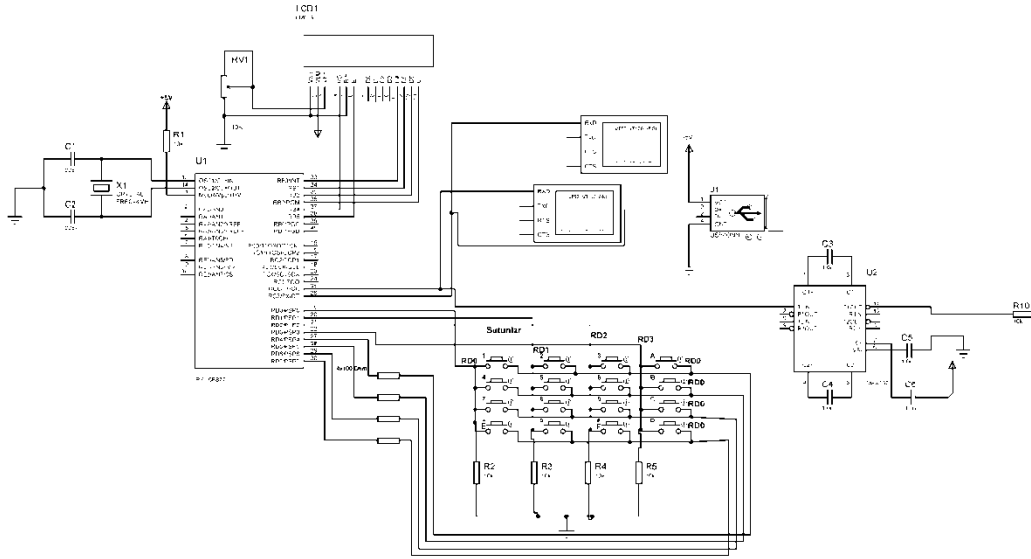
## BÖLÜM 5 DETAYLI TASARIM

Projenin detaylı tasarımında kavramsal tasarımda denem amaçlı yapılan ürün fiyatı ve ürün adı menüsü ikinci plana atılarak kullanıcının 3 satır olarak serbest karakterler girebilmesi sağlandı. Lcd de seçim yapabilmek için menü ekranı yapıldı. Menü ekranında bilgisayar ara yüzüne geçiş seçeneği sunuldu. Mikroişlemci modu seçildiğinde ise kullanıcı 3 satır metin girebilmesi sağlandı. Girilecek olan karakterler lcd de ilk satırda görüntülenmesi sağlandı. İkinci satırda kaçınıcı satırda olduğu ve hangi modda olduğu görüntülendi. En son hafızada olan metnin çıktısının tekrar yazılmadan alınması sağlandı.

Projemizde ek olarak Matlab GUI de arayüz programı yapıldı. Program arayüzünde kullanıcı 3 satırı ayrı ayrı girebilecek ve yazdır butonuna basarak çıktı alabilecektir. Ayrıca bilgisayardan seçilen logonun bitmap formatına çevrilmesi ve çıktısının alınması gerçekleştirildi. Programın haberleşmesi için 9 pinli USB-RS-232 kablo kullanıldı.

### 5.1 Proteus Çizimi

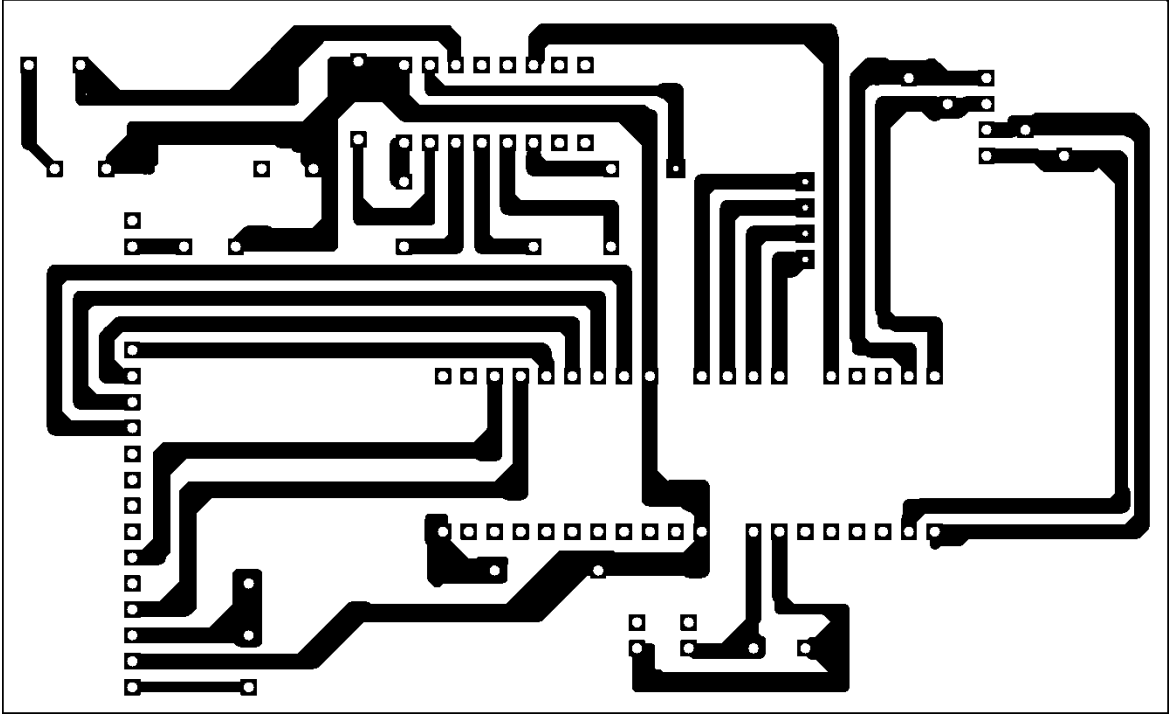
Şekil-22 ve Şekil-23 te labcenter electronics firmasına ait olan proteus programıyla çizilmiş devreler görülmektedir. Bu program sayesinde devre sanal ortamda denenmiş ve hatalar aykıldıktan sonra gerçek ortama aktarılmıştır.



Şekil 19:Tüm devrenin proteus çizimi



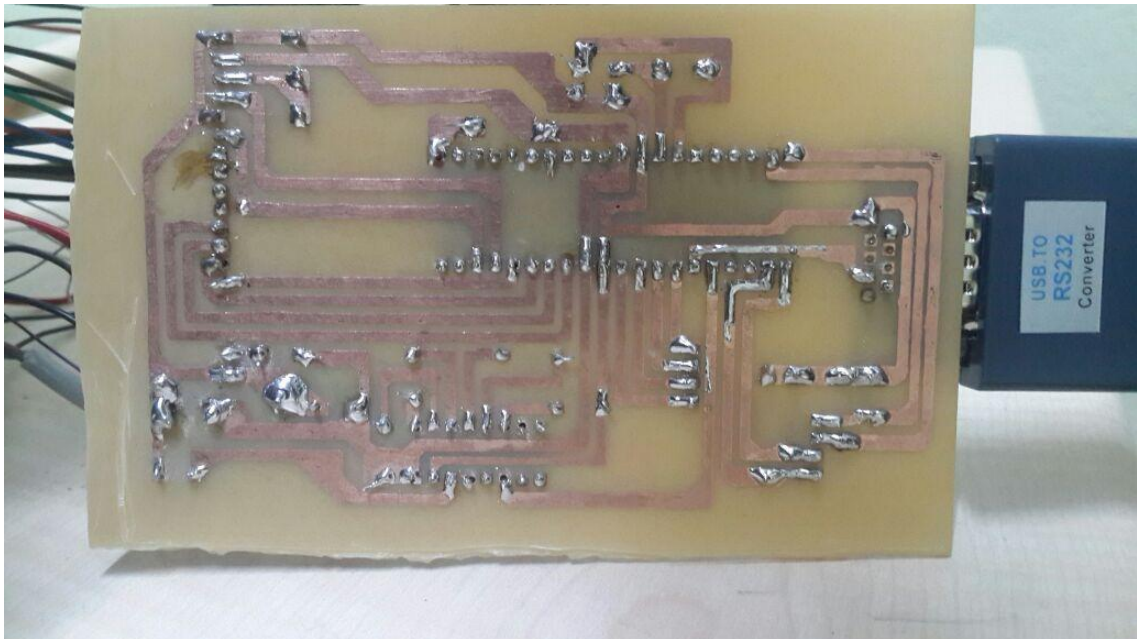
## 5.2 Baskı Devre Tasarımı



Şekil 20:Tüm devrenin ares çizimi

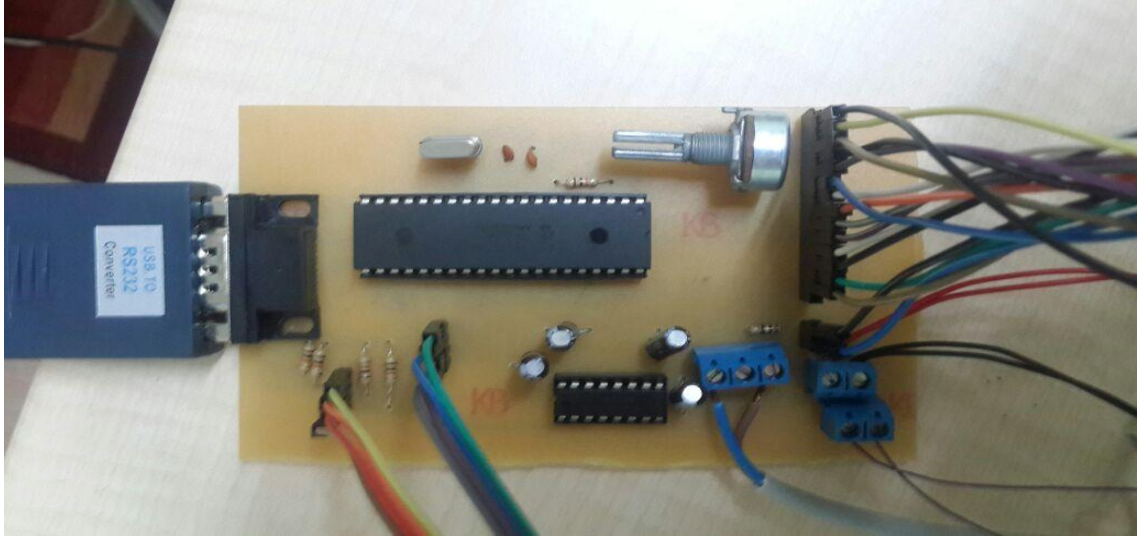
## 5.3 Baskı Devre ve Fiziksel olarak Projenin Tamamlanması

Şekil-21 de bakır plakette basılmış devrede lehimlenmiş devre elemanları bacakları görünmektedir. Burada mikroişlemci gibi entegrelerin direk lehimlenmesi entegrenin yanmasına sebebiyet vereceği için 40-pin soket kullanılarak lehimlenmesi gerekmektedir.

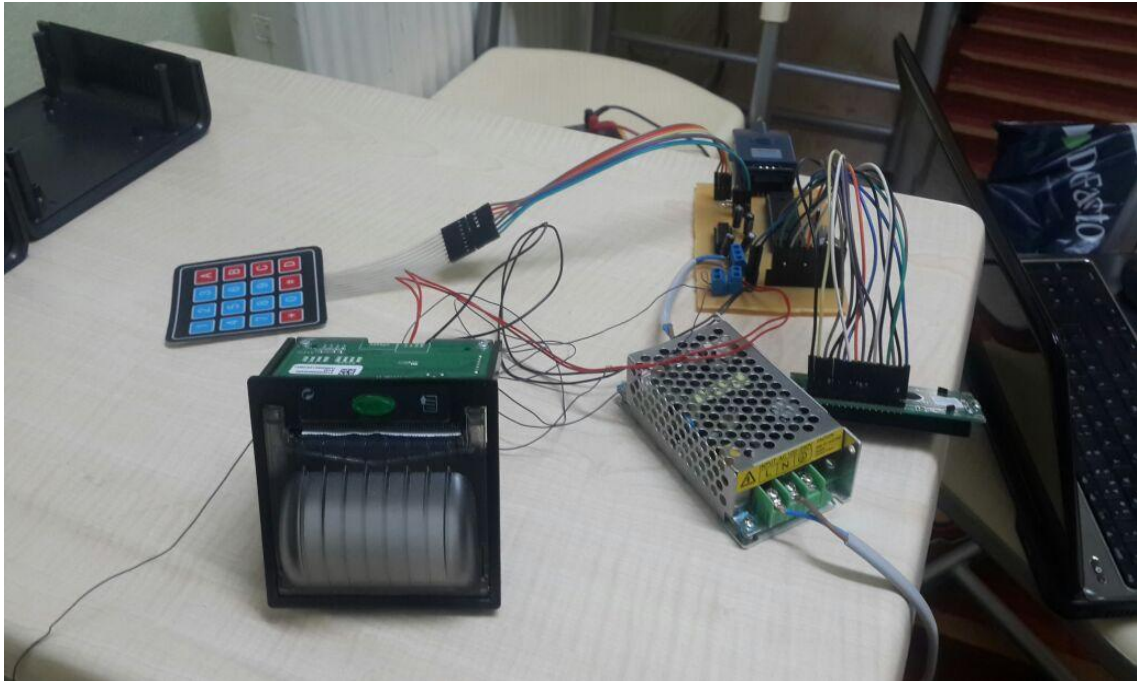


Şekil 21:Baskılanmış devrenin alt görünüşü

Şekil-22 ve Şekil-23 te daha önceki bölümlerde belirtilen malzemeler temin edildi. Ares çiziminin baskı devresi yapıldı. Tüm malzemeler yerlerine lehimlendi. Kısa devre gibi sorunlar kontrol edildi. İlk olarak devremiz rs 232 haberleşme modülü kullanılarak denenmiş ancak daha sonra haberleşme olarak TTL dönüştürücüye geçilmiştir.



Şekil 22: Baskılanmış devrenin üst görünüşü



Şekil 23: Bütün komponentlerin birleştirilmesi

Şekil-24 ve Şekil-25 te etiket makinesinin bitmiş hali görülmektedir. Güç kaynağı olarak devrede 5V 5A bir AC-DC çevirici kullanılmıştır.



Şekil 24: Kutu tasarımı

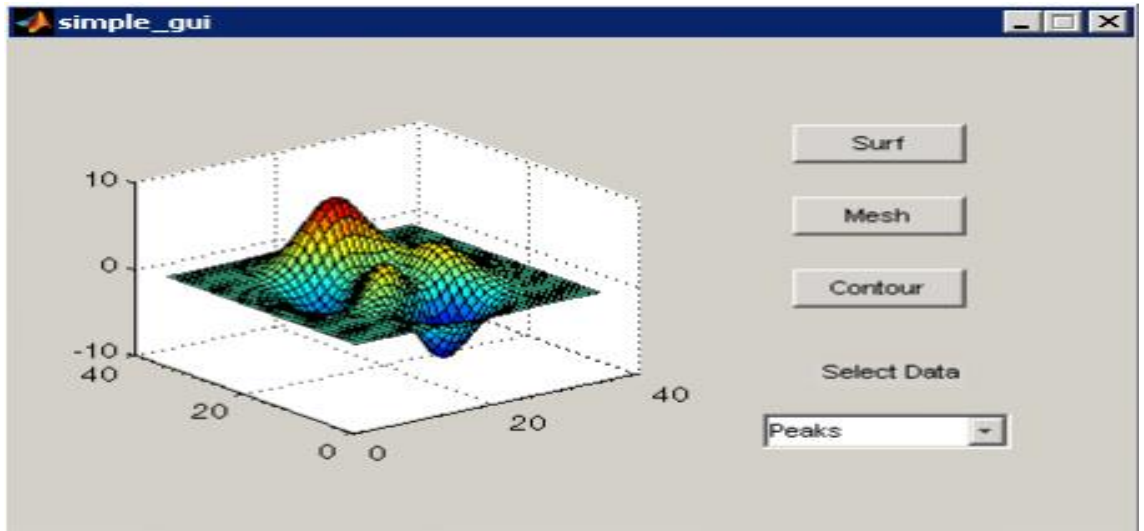


Şekil 25:Montelenmiş ürünün ön,arka perspektif görüntüsü

## BÖLÜM 6 MATLAB GUI İLE KULLANICI ARAYÜZÜ TASARIMI

### 6.1 MATLAB KULLANICI ARAYÜZÜ

İçeriğinde yer alan nesnelerin kullanılması ile kullanıcıya etkileşim sağlayan ve bir işin veya bir programın koşturulmasını sağlayan grafiksel bir program arayüzüdür. Açılımı Graphical User Interface (GUI) dir. GUI nesneleri menüler, araç çubukları, radio butonlar, liste kutuları veya kaydırıcılar olabilir. Bunların yanında MATLAB GUI ile MATLAB'ın sunduğu hesaplama imkânları kullanılarak da data alımı ve grafik çizimi gibi pek çok işlem gerçekleştirilebilir. Şekil-1'de basitçe bir GUI arayüzü görülmektedir



Şekil 26 Basit bir gui arayüzü tasarımı

#### 6.1.1 GRAFİKSEL KULLANICI ARAYÜZÜ NASIL ÇALIŞIR

Her bir nesne (veya komponent) GUI için tanımlanan programlama dosyasında callback diye adlandırılan ayrı alt rutin programlama parçalarına sahiptir. Bu şekilde her bir nesnede oluşan olaylara (örnek olarak bir buton nesnesinin tıklanması ile click event oluşması gibi) GUI o olaya ait callback rutinlerini icra ettirir. Yani, GUI hem bir arayüz hem de bir program çağrılarını icra ettirme mekanizması olarak çalışır.

Yukarıda bahsedilen programlama olay tabanlı programlama diye adlandırılır. Bu tür programlamada her bir olaylara ait alt program parçaları birbirinden bağımsız olarak MATLAB GUI tarafından çalıştırılır.



## 6.1.2 MATLAB'DA GUI OLUŞTURMA YÖNTEMLERİ

Matlab gui tasarımı iki farklı yol kullanılarak yapılabilir.

- **Matlab guide aracı kullanarak**
- **M-file programlama yöntemi kullanarak**

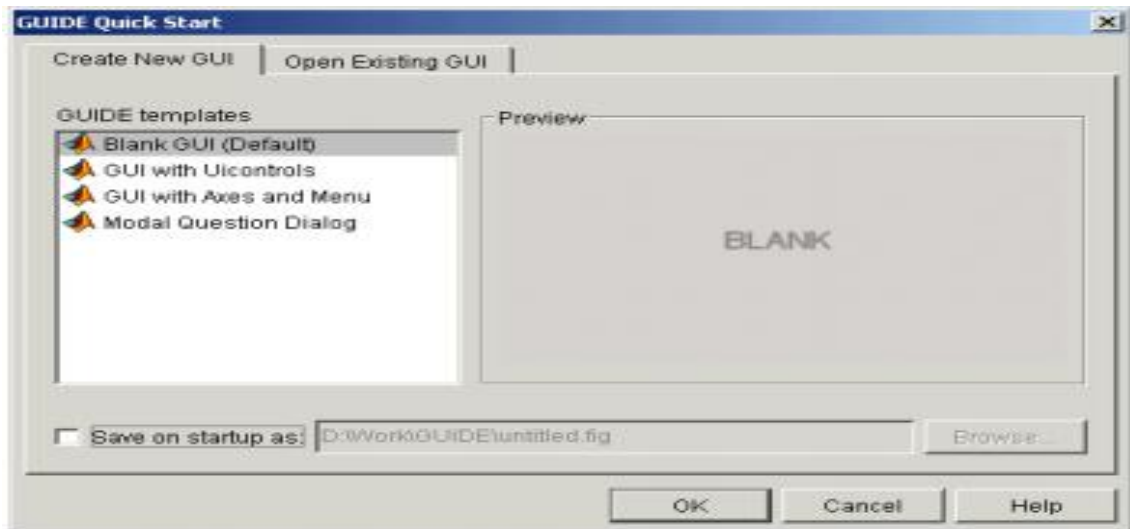
Özellikle GUI tasarımında hızlı arayüzler dizayn etmek ve bu işe ilk başlayan programcılar için MATLAB GUIDE aracının kullanılması büyük bir kolaylık sağlar. Bu aracın kullanılması ile GUI arabirimi kolaylıkla ve yorulmadan sürükleyip bırak ve açılan pencerelerde özelliklerin değiştirilmesine dayanan bir yöntem kullanılır. Ayrıca, bu yöntemi kullanmanın ileride var olan bir GUI'nin düzenlenmesi ve değişiklik yapılması bakımından da çok yararlıdır.

M-File programlama yönteminde tüm GUI tasarımları ve callback program parçalarının yazılması tamamıyla programlama kodları kullanılarak yapılır. Burada tasarımcı her şeye hakimdir ve bu teknik uzman bir programlama bilgisi gerektirir. Bu yöntem ile tasarım zamanı uzamasına rağmen programcı her türlü manipülasyonu yapabildiği için programcı açısından çok yararlıdır. Projemizde guide aracı kullanılarak arayüz tasarlanmıştır.

## 6.1.3 MATLAB GUIDE ARACI İLE GUI TASARIMI OLUŞTURMA

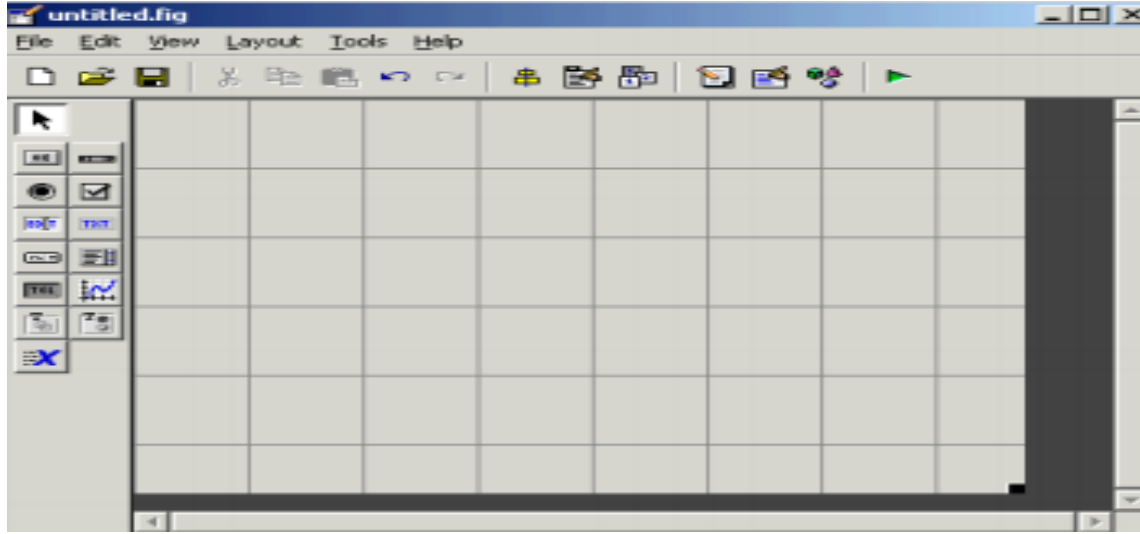
Burada projede kullandığımız yöntemden kısavca bahsedilecektir. GUIDE matlab'in GUI tasarımcılarına sunduğu içerisinde çeşitli araçlar içeren ve kolaylık sağlayan bir grafiksel GUI geliştirme ortamıdır. GUIDE kullanılarak tıklama ve sürükleyip bırak tekniği ile GUI arayüzüne nesneler (örneğin butonlar, text kutuları, liste kutuları, grafikler v.s.) kolaylıkla eklenebilir. Ayrıca, eklenen nesnelerin hizalanması, tab sırasının değiştirilmesi, görsel ayarlar üzerinde manipülasyonlar yapılması da bu ortamın tasarımcılara sunduğu imkânlardan bazılarıdır.

**Matlab guide aracını kısaca tanıyalım:** Bu aracını çalıştırmak için ya MATLAB komut satırından GUIDE komutu verilir ya da Start düğmesi tıklanarak MATLAB/GUIDE komutu verilir. Bu adımdan sonra karşımıza Şekil 2'deki gibi bir pencere gelir.



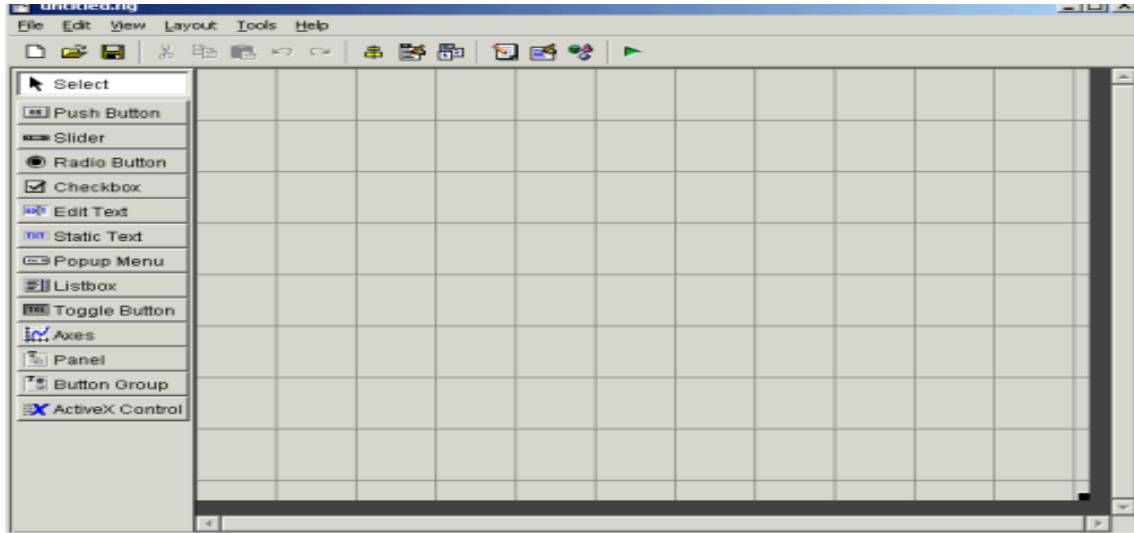
Şekil 27 Gui başlangıç ekranı

Bu pencereden eğer yeni bir GUI tasarımı yapacak isek Blank GUI seçeneğini seçeriz. Şayet önceden yapılmış bir tasarımı açmak istiyor isek Open Existing GUİ sekmesinden sonra istenilen dosyayı seçeriz. Burada yeni bir tasarım oluşturulacağını kabul edelim. Bundan sonra OK düğmesi tıklanılarak Şekil 3'teki GUIDE LAYOUT Editor (GUIDE Çalışma Alanı) penceresine ulaşırız.



Şekil 28 Gui çalışma alanı

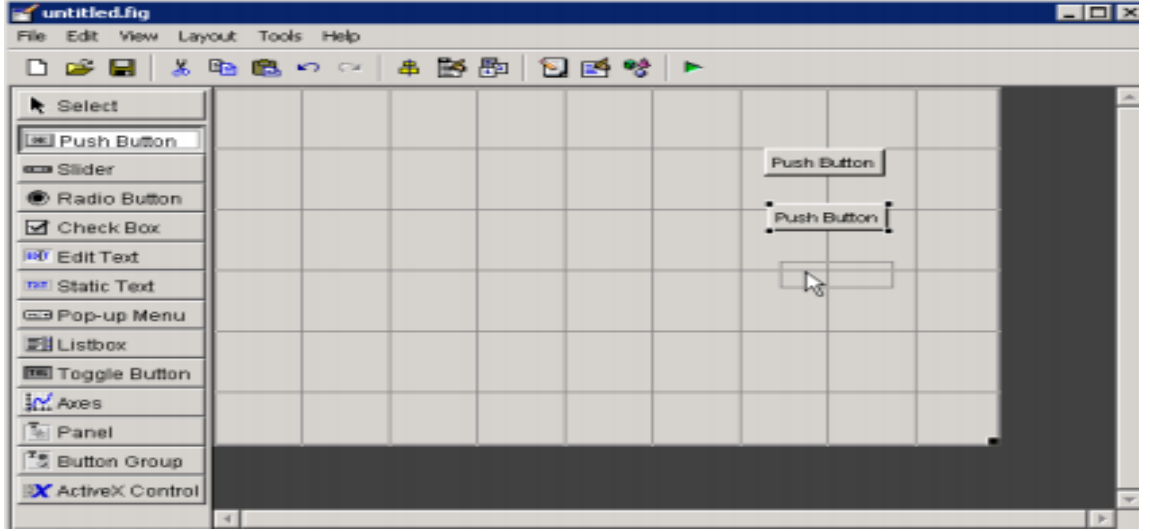
Bu adımdan sonra File/Prefences/Guide yolunu kullanılarak gelen pencereden “Show names in component palette” seçeneğini tıklayıp OK düğmesine basalım. Karşımıza Şekil 4'teki gibi bir pencere gelecektir.



Şekil 29:Gui komponent ekranı

#### 6.1.4 Komponentleri Çalışma Alanına Ekleme

Bunun için sol tarafta bulunan nesne butonlarından istenilen nesneye ait buton tıklanır ve daha sonra çalışma alanında uygun görülen bir noktaya tıklandığında o noktaya ilgili nesne eklenmiş olacaktır. İstenirse çalışma alanındaki bir nesne farenin sol tuşu ile tıklanıp bırakılmadan çalışma alanının herhangi bir yerine sürüklenebilir. Bu durum Şekil 34'te de görülmektedir



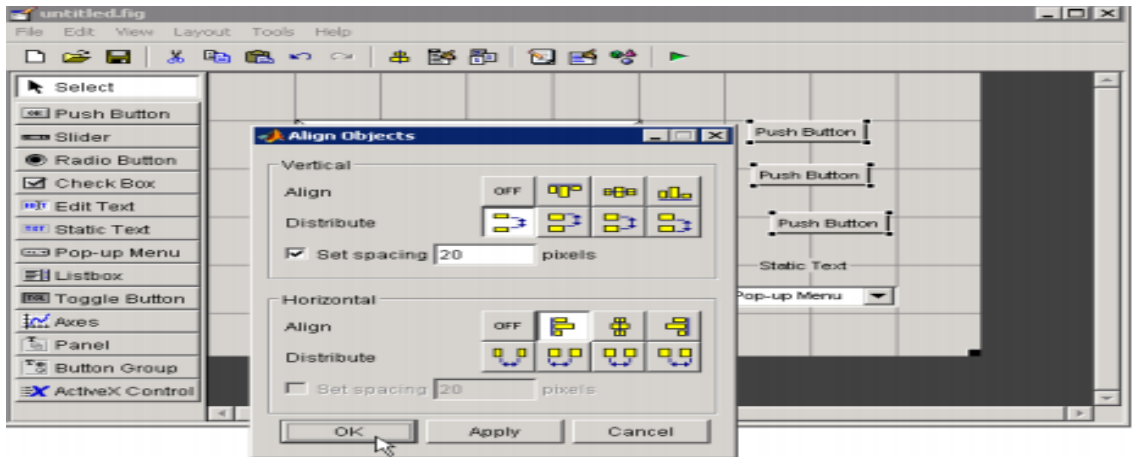
Şekil 30:Guide eleman seçimi

### 6.1.5 Çalışma Alanının Boyutlarını Değiştirmek

Burada da çalışma alanının sağ alt tarafında bulunan siyah karenin üzerine fare işaretçisi getirilir ve fare işaretçisi konum değiştirdiğinde farenin sol tuşu basılı tutularak çalışma alanı istenilen boyutlarda olacak şekilde düzenleme yapılabilir. Bu durum başlangıç anında ayarlanacağı sonradan da ayarlanabilir. Kullanıcı istediği boyutlarda bir arayüzü bu şekilde tasarlayabilmektedir

#### 6.1.5.1 Nesneleri Hizalamak

Bu işlemi yapmak için öncelikle hizalanacak nesneler seçilir. Topluca seçmek için çalışma alanında fare işaretçisini herhangi bir yere tıklayıp sürükleyerek açılan kesik kenarlı pencerenin içinde nesneler kalacak şekilde hareket ettirip, hizalanacak nesneler bu çerçeve içinde kalınca farenin sol tuşunu bırakın. Bu şekilde sadece o çerçeve içinde kalan nesneler seçilmiş olacaktır. Ayrıca, nesneleri Ctrl tuşunu basılı tutarak farenin sol tuşu ile teker teker de seçme imkânı bulunmaktadır. Hizalanacak nesneler seçildikten sonra Tools/Align Objects... yolunu kullanarak Aligment Tool (Hizalama Aracı) penceresini açınız. Şekil 6'deki gibi bir ekran ile karşılaşırız. Burada yatay ve dikey hizalamaları kendimize göre butonlardan seçip OK butonuna bastığımız zaman nesnelerimiz hizalanmış olacaktır. Eğer ki hizalama istenilen gibi olmadı ise Ctrl + Z kısayolu ile yapılan işlemler geri alınabilir.



Şekil 31 Nesneleri hizalama komutu

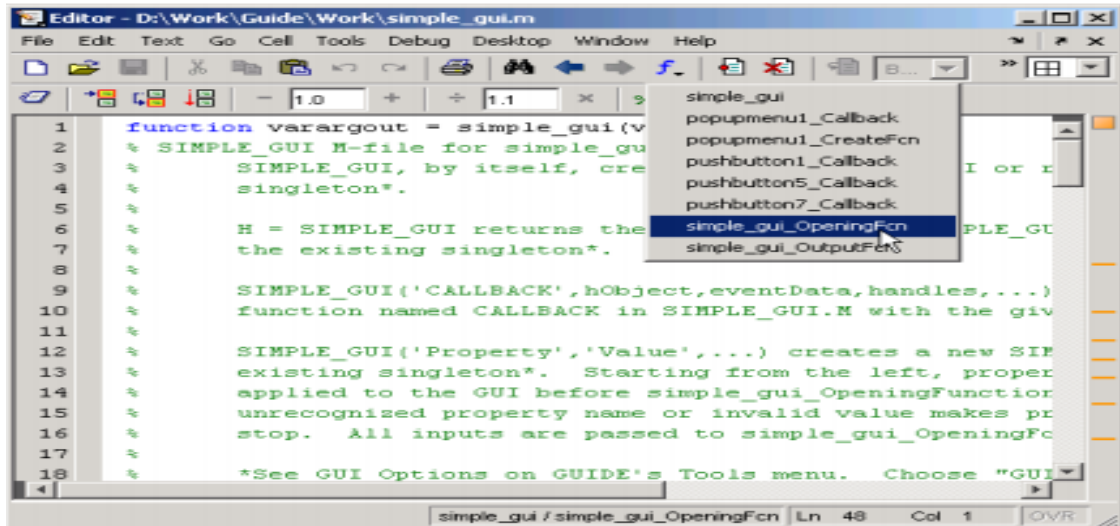
## 6.2 GUI Tasarımını Kaydetme ve Çalıştırma

Bundan sonra bitmiş olan bu GUI arayüzü çalıştırarak görmek için öncelikle Tools/Run yolundan Run (Çalıştır) komutu verilir. Daha sonra gelen pencereden çalışmamın Run edilebilmesi için kaydedilmesi gerektiğini bildiren bir pencere çıkar Burada Yes butonuna basarız. Bu adımdan sonra MATLAB GUIDE bize tasarımın kaydedileceği dosya isminisoran bir pencere getirir. Bu pencereden çalışmamıza bir isim vererek tasarımı kaydetmiş oluruz. Ardından karşımıza Change the MATLAB Directory gibi bir ekran gelirse burada bu ekranı OK tuşuna basarak kapatabilirsiniz. Bu ekran kaydedilen dosya MATLAB tanımlı dizinler dışında bir yere kaydedilme söz konusu olduğunda bizi uyarmaktadır.

## 6.3 GUI Arayüzünün Programlanması

Bir GUI arayüzünün programlanması demek o çalışmanın kaydedildiği isimle aynı zamanla oluşturulan .m uzantılı dosya içerisine kodlama satırlarının eklenmesi demektir. Bu dosyanın içine görebilmek, değişiklik yapabilmek için GUIDE çalışma ekranı penceresinden View/MFile Editor komutu işletilebilir. Ardından karşımıza Şekil 7'deki gibi bir pencere gelecektir. Şekil 7'deki pencerede hazırlamış olduğumuz GUI tasarımına ait kodlar gözükmemektedir. Burada pek çok kodun hazır eklenmiş olduğu görülecektir. Bu kodlar otomatik olarak MATLAB GUIDE tarafından eklenmiştir. Biz burada ilgili butonlara ve liste kutularına ya da istenilen bir nesneye ait callback isimli alt program parçalarına ilgili kodları yazacağız.

Bir nesneye ait callback in bulunduğu satıra gitmek için araç çubuğunda yer alan f simgeli butona tıklanır ve açılan listeden ilgili nesneye ait callback in ismi seçilir. Bu durum yukarıdaki pencerede de görülmektedir. Ayrıca, GUIDE çalışma ekranından da direk istenilen bir callback satırına gidilebilir. Bunun için ilgili nesne üzerinde sağ tıklanır ve açılan pencereden View Callbacks menüsünden ilgili callback tıklanması ya da ilgili nesne seçilip View/View Callbacks yolu üzerinden açılan listeden gidilmek istenilen callback tıklanması yeterlidir.



Şekil 32 Guide m-file kod arayüzü

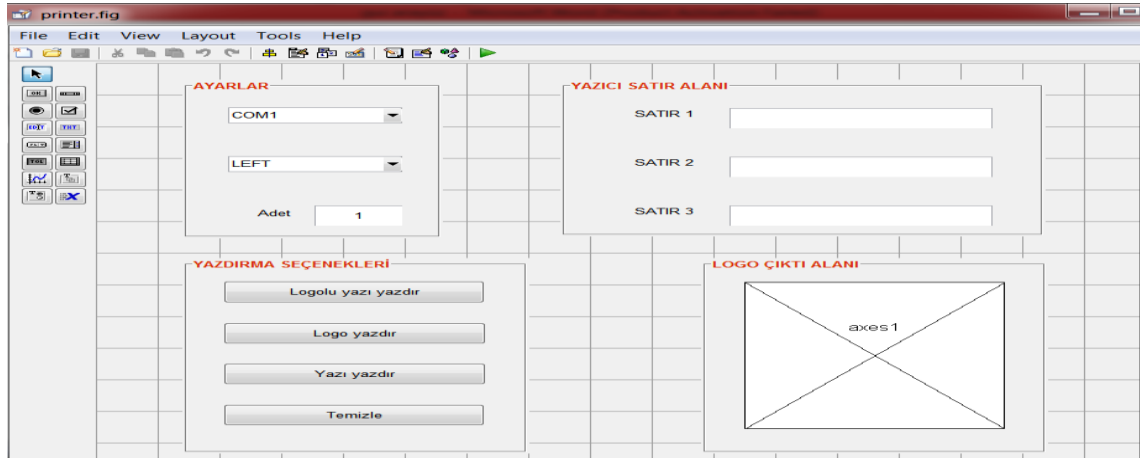
Buraya kadar matlab programı ile gui arayüzünün nasıl oluşturulacağından bahsettik.Şimdi ise kendi projemizde yapmış olduğumuz arayuzden bahsedilecektir.Etiket makinelerinde kullanıcı bilgisiyar üzerinden makinenin kullanılmasını isteyebilir.



Burada bilgisayar üzerinde arayüz oluşturma'nın birçok yolu bulunmaktadır. Biz de arayüz tasarımı olarak matworks firmasının yazılımı olan matlab guide programı ile yaptık.

## 6.4 GUIDE ARAYUZU BAŞLANGIÇ

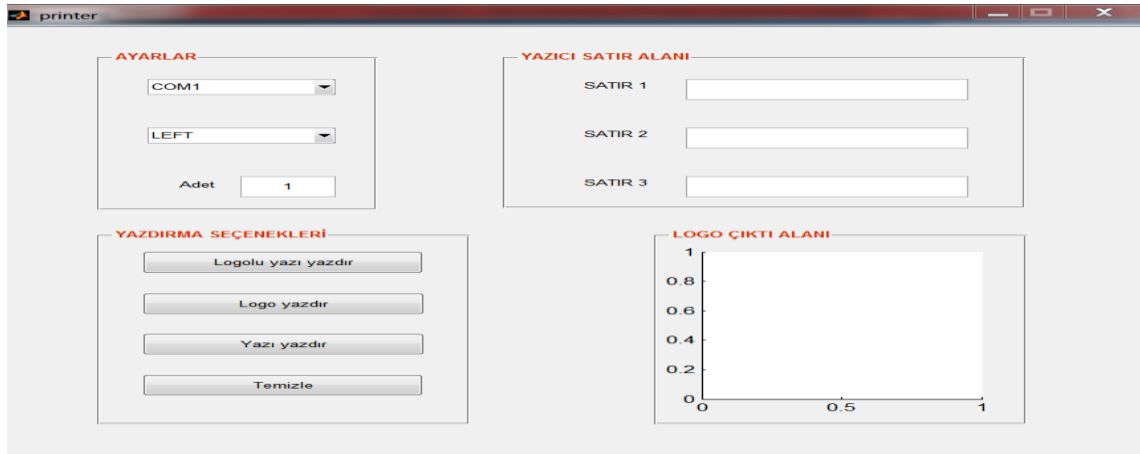
Guide oluşturduğumuz arayüzle kullanıcıya tuş takımından bağımsız bir şekilde bilgisayar üzerinden çıktı alması sağlanmıştır. Şekil-37 de arayüzde kullanıcı öncelikle ayarlar panelinden com seçeneklerini, yazı konumunu ve kaç adet çıktı alınacaksa çıktı miktarını ayarlayabilir. Ayrıca kullanıcı çıkartmak istediği yazıyı yazıcı satırından maksimum 3 satır yazı girebilmektedir. Kullanıcı yazdırma seçenekleri panelinden ise logo yazdırma, logo ile yazdırma, yazı yazdırma seçeneklerinden bir tanesini seçip çıktı alabilmektedir. Ayrıca kullanıcı temizle butonu ile ise girilen veriler silinebilmektedir. Sağ alt köşede bulunan grafikte ise kullanıcının seçmiş olduğu resim yeniden boyutlandırılıp grafikte gösterilmektedir.



Şekil 33:Guide tasarımının ilk hali

## 6.5 GUIDE ARAYUZU SON HALİ

Guide arayüzü çalıştırılınca karşımıza Şekil-38 deki gibi bir ekran gelecektir.



Şekil 34:Bilgisayar arayüzünün son hali

## EKLER

### Ek 1. Deneme Amaçlı Yapılan ilk programın Kodları

```
#include <16f877.h> // Kullanılacak denetleyicinin başlık dosyası tanıtılıyor.
// Denetleyici konfigürasyon ayarları
#define fusesHS,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODE
B UG,NOCPD
//use delay (clock=20000000) // Gecikme fonksiyonu için kullanılacak osilatör
frekansı belirtiliyor.
//use fast_io(b) //Port yönlendirme komutları B portu için geçerli
//use fast_io(d) //Port yönlendirme komutları B portu için geçerli
#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
char tus=' '; // karakter tipinde değişken tanımlanıyor
int i=0,j=0,k=1,r=1,m=1;

//***** Keypad Tarama Fonksiyonu *****

char keypad_oku() // Fonksiyon ismi
{
    output_d(0x00); // D portu çıkışı sıfırlanıyor
    output_high(sat1); // 1. satır lojik-1 yapılıyor
    if (input(sut1)) // 1. sütun okunuyor
    {
        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='/';
        }
    }
}
```

```

if (input(sut2)) // 2. sütun okunuyor
{
    while(input(sut2)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='2';
    if(i==1 && j==0) tus=65;
    if(i==1 && j==1) tus=66;
    if(i==1 && j==2) tus=67;
    }}
if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3)){if(i==1){delay_ms(500);j++;j=(j%3);}
    if(i==0) tus='3';
    if(i==1 && j==0) tus=68;
    if(i==1 && j==1) tus=69;
    if(i==1 && j==2) tus=70;
    }}
if (input(sut4)) // 4. sütun okunuyor
{
    while(input(sut4));
    i++;j=-1;i=(i%2);
    if(i==1)
    {
        imlec(2,16);
        printf(lcd_veri,"H");
        if(r==1) imlec(r,k);
        if(r==2) imlec(r,m);
    }
    if(i==0)
    {
        imlec(2,16);
        printf(lcd_veri,"R");
        if(r==1) imlec(r,k);
        if(r==2) imlec(r,m);
    }
}
output_low(sat1); // 1. satır lojik-0 yapıyor
output_high(sat2); // 2. satır lojik-1 yapıyor
if (input(sut1)) // 1. sütun okunuyor
{

```

```

while(input(sut1)){if(i==1){delay_ms(500);j++;j=(j%3);}
    if(i==0) tus='4';
    if(i==1 && j==0) tus=71;
    if(i==1 && j==1) tus=72;
    if(i==1 && j==2) tus=73;
    }}
if (input(sut2)) // 2. sütun okunuyor
{
    while(input(sut2)){if(i==1){delay_ms(500);j++;j=(j%3);}
        if(i==0) tus='5';
        if(i==1 && j==0) tus=74;
        if(i==1 && j==1) tus=75;
        if(i==1 && j==2) tus=76;
        }}
if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3)){if(i==1){delay_ms(500);j++;j=(j%3);}
        if(i==0) tus='6';
        if(i==1 && j==0) tus=77;
        if(i==1 && j==1) tus=78;
        if(i==1 && j==2) tus=79;
        }}
if (input(sut4)) // 4. sütun okunuyor
{
    while(input(sut4));
        if(r==1)
        {
            k++;
            if(k==17)
            {
                k=16;
            }
            imlec(r,k);
        }
        else if(r==2)
        {
            m++;
            if(m==15)
            {
                m=14;
            }
        }
    }

```

```

    }
    imlec(r,m);
    }
    tus=' ';
    j=-1;
}
output_low(sat2); // 2. satır lojik-0 yapılıyor
output_high(sat3); // 3. satır lojik-1 yapılıyor
if (input(sut1)) // 1. sütun okunuyor
{
    while(input(sut1)){if(i==1){delay_ms(500);j++;j=(j%3);}
    if(i==0) tus='7';
    if(i==1 && j==0) tus=80;
    if(i==1 && j==1) tus=82;
    if(i==1 && j==2) tus=83;
    }}
if (input(sut2)) // 2. sütun okunuyor
{
    while(input(sut2)){if(i==1){delay_ms(500);j++;j=(j%3);}
    if(i==0) tus='8';
    if(i==1 && j==0) tus=84;
    if(i==1 && j==1) tus=85;
    if(i==1 && j==2) tus=86;
    }}
if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3))
    {
        if(i==1){delay_ms(500);j++;j=(j%3);}
        if(i==0) tus='9';
        if(i==1 && j==0) tus=88;
        if(i==1 && j==1) tus=89;
        if(i==1 && j==2) tus=90;
    }
}
if (input(sut4)) // 4. sütun okunuyor
{
    while(input(sut4));
    if(r==1)
    {

```

```

    k--;
    if(k==0)
    {
        k=1;
    }
    imlec(r,k);
}
else if(r==2)
{
    m--;
    if(m==0)
    {
        m=1;
    }
    imlec(r,m);
}
tus=' ';
j=-1;
}

output_low(sat3); // 3. satır lojik-0 yapıyor
output_high(sat4); // 3. satır lojik-1 yapıyor
if (input(sut1)) // 1. sütun okunuyor
{
while(input(sut1));{r=1;k=1;m=1;lcd_hazirla();imlec(2,16);printf(lcd_veri,"R");imlec(2,15); printf(lcd_veri,"1");tus=' ';imlec(1,1);}}
    if (input(sut2)) // 2. sütun okunuyor
    {
        delay_ms(20);
        if(i==0) tus='0';
        if(i==1) tus=' ';

    }
    if (input(sut3)) // 3. sütun okunuyor
    { delay_ms(20); tus=0xF; }
    if (input(sut4)) // 4. sütun okunuyor
    {
        while(input(sut4));
        r++;
        if(r==3){r=1;}
        tus=' ';

```

```

j=-1;
if(r==1)
{
imlec(2,15);
printf(lcd_veri,"1");
imlec(r,k);
}
if(r==2)
{
imlec(2,15);
printf(lcd_veri,"2");
imlec(r,m);
}
}
output_low(sat4); // 3. satır lojik-0 yapılıyor
return tus; // Fonksiyon "tus" değeri ile geri döner
}
//***** ANA PROGRAM FONKSİYONU*****

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_b(0x00); // B portu komple çıkış
set_tris_d(0x0F); // Yüksek değerlikli 4 bit çıkış, düşük değerlikli 4 bit giriş

lcd_hazirla(); // LCD hazırlanıyor

imlec(2,16); // İmleç 3.sütun, 1.satıra konumlandırılıyor
printf(lcd_veri,"R");
//printf(lcd_veri,"URUN="); // LCD'ye string yazdırılıyor
imlec(2,15); // İmleç 3.sütun, 1.satıra konumlandırılıyor
printf(lcd_veri,"1");
while(1) // Sonsuz döngü
{
if(r==1) imlec(r,k);

```

```

if(r==2) imlec(r,m);// İmleç 3.sütun, 1.satıra konumlandırılıyor

if (keypad_oku()>9) // Eğer basılan tuş değeri 9'dan büyük ise
    printf(lcd_veri,"%c ",keypad_oku()); // Tuş değeri LCD'ye yazdırılıyor
else // Eğer basılan tuş değeri 9'dan büyük değilse
    printf(lcd_veri,"%c ",keypad_oku()); // Tuş değeri LCD'ye yazdırılıyor

}
}

```



**Şekil 35: İlk programın yazıcı çıktısı**

## **Ek 2. Açıklamalı CCS C Kodları**

```

#include <16f877.h> // Kullanılacak denetleyicinin başlık dosyası tanıtılıyor.
// Denetleyici konfigürasyon ayarları
#fuses HS,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPPD
#use delay (clock=2000000) // Gecikme fonksiyonu için kullanılacak osilatör frekansı belirtiliyor.
#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
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,yzk1=0,yzm=0,sfr=0,bos=0,syc=0,syc1=0,yzc=0,q=0;
//integer tipinde çeşitli sayaçlar için değişken tanımlama
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_yazdırma Fonksiyonu *****
void eeprom_yaz()
{
    *****// girilen yazının 24*3satır yazılma işlemi*****
    write_eeprom(0,abc[0]);
    write_eeprom(1,abc[1]);
    write_eeprom(2,abc[2]);
    write_eeprom(3,abc[3]);
    write_eeprom(4,abc[4]);
    write_eeprom(5,abc[5]);
    write_eeprom(6,abc[6]);
    write_eeprom(7,abc[7]);
    write_eeprom(8,abc[8]);
    write_eeprom(9,abc[9]);
    write_eeprom(10,abc[10]);
    write_eeprom(11,abc[11]);
    write_eeprom(12,abc[12]);
    write_eeprom(13,abc[13]);
    write_eeprom(14,abc[14]);
    write_eeprom(15,abc[15]);
    write_eeprom(16,abc[16]);
    write_eeprom(17,abc[17]);
    write_eeprom(18,abc[18]);
    write_eeprom(19,abc[19]);
    write_eeprom(20,abc[20]);
    write_eeprom(21,abc[21]);
    write_eeprom(22,abc[22]);
    write_eeprom(23,abc[23]);
    write_eeprom(24,abc[24]);
    write_eeprom(25,abc[25]);
    write_eeprom(26,abc[26]);
    write_eeprom(27,abc[27]);
    write_eeprom(28,abc[28]);
    write_eeprom(29,abc[29]);
    write_eeprom(30,abc[30]);
    write_eeprom(31,abc[31]);
    write_eeprom(32,abc[32]);
    write_eeprom(33,abc[33]);
    write_eeprom(34,abc[34]);

```

```

write_eeprom(35,abc[35]);
write_eeprom(36,abc[36]);
write_eeprom(37,abc[37]);
write_eeprom(38,abc[38]);
write_eeprom(39,abc[39]);
write_eeprom(40,abc[40]);
write_eeprom(41,abc[41]);
write_eeprom(42,abc[42]);
write_eeprom(43,abc[43]);
write_eeprom(44,abc[44]);
write_eeprom(45,abc[45]);
write_eeprom(46,abc[46]);
write_eeprom(47,abc[47]);
write_eeprom(48,abc[48]);
write_eeprom(49,abc[49]);
write_eeprom(50,abc[50]);
write_eeprom(51,abc[51]);
write_eeprom(52,abc[52]);
write_eeprom(53,abc[53]);
write_eeprom(54,abc[54]);
write_eeprom(55,abc[55]);
write_eeprom(56,abc[56]);
write_eeprom(57,abc[57]);
write_eeprom(58,abc[58]);
write_eeprom(59,abc[59]);
write_eeprom(60,abc[60]);
write_eeprom(61,abc[61]);
write_eeprom(62,abc[62]);
write_eeprom(63,abc[63]);
write_eeprom(64,abc[64]);
write_eeprom(65,abc[65]);
write_eeprom(66,abc[66]);
write_eeprom(67,abc[67]);
write_eeprom(68,abc[68]);
write_eeprom(69,abc[69]);
write_eeprom(70,abc[70]);
write_eeprom(71,abc[71]);
}

//***** Eeprom_yazdırma Fonksiyonu *****
void eeprom_oku()
{
    syc=read_eeprom(75); /* yazıcının kaymasını önlemek için...
                           kacinci yazdırmada olduğunu saklayan değişken */

    k1=read_eeprom(74); // yazının ortalanması için kullanıcının kaç karakter
    m1=read_eeprom(73); // girdiğini her satır için ayrı değişken atayıp tekrar
    n1=read_eeprom(72); // yazdırmada bu bilgilerin saklanıp tekrardan ortalanmasını
                        // sağlayan okuma işlemi

    *****// eeproma yazılan 24*3satır verinin okunması işlemi*****

```

```
abc1[0]=read_eeprom(0);
abc1[1]=read_eeprom(1);
abc1[2]=read_eeprom(2);
abc1[3]=read_eeprom(3);
abc1[4]=read_eeprom(4);
abc1[5]=read_eeprom(5);
abc1[6]=read_eeprom(6);
abc1[7]=read_eeprom(7);
abc1[8]=read_eeprom(8);
abc1[9]=read_eeprom(9);
abc1[10]=read_eeprom(10);
abc1[11]=read_eeprom(11);
abc1[12]=read_eeprom(12);
abc1[13]=read_eeprom(13);
abc1[14]=read_eeprom(14);
abc1[15]=read_eeprom(15);
abc1[16]=read_eeprom(16);
abc1[17]=read_eeprom(17);
abc1[18]=read_eeprom(18);
abc1[19]=read_eeprom(19);
abc1[20]=read_eeprom(20);
abc1[21]=read_eeprom(21);
abc1[22]=read_eeprom(22);
abc1[23]=read_eeprom(23);
abc1[24]=read_eeprom(24);
abc1[25]=read_eeprom(25);
abc1[26]=read_eeprom(26);
abc1[27]=read_eeprom(27);
abc1[28]=read_eeprom(28);
abc1[29]=read_eeprom(29);
abc1[30]=read_eeprom(30);
abc1[31]=read_eeprom(31);
abc1[32]=read_eeprom(32);
abc1[33]=read_eeprom(33);
abc1[34]=read_eeprom(34);
abc1[35]=read_eeprom(35);
abc1[36]=read_eeprom(36);
abc1[37]=read_eeprom(37);
abc1[38]=read_eeprom(38);
abc1[39]=read_eeprom(39);
abc1[40]=read_eeprom(40);
abc1[41]=read_eeprom(41);
abc1[42]=read_eeprom(42);
abc1[43]=read_eeprom(43);
abc1[44]=read_eeprom(44);
abc1[45]=read_eeprom(45);
abc1[46]=read_eeprom(46);
abc1[47]=read_eeprom(47);
abc1[48]=read_eeprom(48);
abc1[49]=read_eeprom(49);
abc1[50]=read_eeprom(50);
```

```

abc1[51]=read_eeprom(51);
abc1[52]=read_eeprom(52);
abc1[53]=read_eeprom(53);
abc1[54]=read_eeprom(54);
abc1[55]=read_eeprom(55);
abc1[56]=read_eeprom(56);
abc1[57]=read_eeprom(57);
abc1[58]=read_eeprom(58);
abc1[59]=read_eeprom(59);
abc1[60]=read_eeprom(60);
abc1[61]=read_eeprom(61);
abc1[62]=read_eeprom(62);
abc1[63]=read_eeprom(63);
abc1[64]=read_eeprom(64);
abc1[65]=read_eeprom(65);
abc1[66]=read_eeprom(66);
abc1[67]=read_eeprom(67);
abc1[68]=read_eeprom(68);
abc1[69]=read_eeprom(69);
abc1[70]=read_eeprom(70);
abc1[71]=read_eeprom(71);
}
*****// matlab ile gönderilen verinin algılandığı RDA kesmesi*****
#int_RDA
void seri()
{
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); // yazıcının ayar komutu
putc(0x0B); // satır olarak boşluk bırakma komutu
}

putc(0x02); // yazının fon büyüklüğünü ayarlama komutu

for(yzk=0;yzk<72;yzk++) // hafızaya alınan yazı yazıcıya gönderilimi
{
putc(abc[yzk]);
}
putc(0x32); // yazıcının ayar komutu
putc(0x0B); // satır olarak boşluk bırakma komutu
}
}

```

```

if(mkt=='M') // resim gelmesi durumunda
{
    oku++;

    for(yzk=0;yzk<48;yzk++)
    {
        t[yzk]=getc();
    }
    putc(0x1D); // standart komut olarak 48 byte veri gönderme komutu
    putc(0x57); // bu komut ile logo/resim çizimi esnasında gönderilen veri
    putc(0x30); // satırı 48 byte olacak şekilde satır satır gönderimi yapıyor

    for(yzk=0;yzk<48;yzk++)
    {
        putc(t[yzk]);
    }
    if(oku==128) // 128 satır olarak dönüştürülen resim verisinin bittiğini
    {
        // denetleyen komut
        putc(0x033); // boşluk bırakma
        putc(0x0B); //
        oku=0; // logo/resim alımında kayıt olmaması için sayaç sıfırlama
    }
}

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); // logo/resim için standart veri gönderme komutu 1D 57
        putc(0x57); //
        putc(0x08); // kaç byte veri gönderileceğinin seçimi

        for(yzk=0;yzk<8;yzk++) // 8byte veri yazma
        {
            putc(t[yzk]);
        }
    }
    if(oku==65) // logo/resim verisi bitişinin denetlenmesi
    {
        putc(0x1B); // yazıcıya karakter olarak yazı yazdırılacağı komutu
        putc('N'); // 1B 'N'
        putc(0x02); // fon büyüklüğü ayar komutu
        for(yzk=0;yzk<24;yzk++) // satır veri alım döngüsü
        {

```

```

    abc[yzk]=getc();
}

syc++;
if(syc==7) // kayma engelleyici sayac
{
    syc=0;
    putc(0x31);
    putc(0x0B);
}

putc(0x02);
for(yzk=0;yzk<24;yzk++) // satır yazdırma komutu
{
    putc(abc[yzk]);
}
putc(0x34);
putc(0x0B);
oku=0;
}

}
}
/*****Tuş takımı okuma komutu*****/
/* kontrol d portunun yüksek değerlikli bitlerinin high olma durumuna göre
taranan düşük değerlikli giriş bitlerinin high olmasıyla hangi tuşa basıldığı
kontrol ediliyor*/ // kaynak: Mikroişlemciler dersi

/* tuşların birden fazla özelliği bulunmakta bu özellikler 0-9 haricindeki tuşlara
ait olan sayaçların değerlerine göre seçilmektedir. */

/* i=0 olması durumunda rakam yazılımı sağlanmaktadır*/
/* i=1 olması durumunda harf yazılımı sağlanmaktadır */

/* j sayacı mod3 e göre tuşların harici kaçınıcı özelliğinin kullanacağını belirlemektedir */
/* j=0 1.özellik*/
/* j=1 2.özellik*/
/* j=2 3.özellik*/

char keypad_oku() // Fonksiyon ismi
{
    output_d(0x00); // D portu çıkışı sıfırlanıyor

    output_high(sat1); // 1. satır lojik-1 yapılıyor
    if (input(sut1)) // 1. sütun okunuyor
    {
        while(input(sut1)){if(i==1){delay_ms(250);j++;j=(j%3);}
            if(j==0) tus='1';
            if(i==1 && j==0) tus='.';
            if(i==1 && j==1) tus=':';

```

```

    if(i==1 && j==2) tus='/';
    }
    }
if (input(sut2)) // 2. sütun okunuyor
{
    while(input(sut2)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='2';
    if(i==1 && j==0) tus=65;
    if(i==1 && j==1) tus=66;
    if(i==1 && j==2) tus=67;
    }}
if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='3';
    if(i==1 && j==0) tus=68;
    if(i==1 && j==1) tus=69;
    if(i==1 && j==2) tus=70;
    }}
if (input(sut4)) // 4. sütun okunuyor
{
    while(input(sut4));
    if (MENU==2) // harf/rakam seçimi
    {
        i++;j=-1;
        i=(i%2);
    }
}
output_low(sat1); // 1. satır lojik-0 yapılıyor
output_high(sat2); // 2. satır lojik-1 yapılıyor
if (input(sut1)) // 1. sütun okunuyor
{
    while(input(sut1)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='4';
    if(i==1 && j==0) tus=71;
    if(i==1 && j==1) tus=72;
    if(i==1 && j==2) tus=73;
    }}
if (input(sut2)) // 2. sütun okunuyor
{
    while(input(sut2)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='5';
    if(i==1 && j==0) tus=74;
    if(i==1 && j==1) tus=75;
    if(i==1 && j==2) tus=76;
    }}
if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3)){if(i==1){delay_ms(250);j++;j=(j%3);}
    if(i==0) tus='6';
    if(i==1 && j==0) tus=77;

```

```

if(i==1 && j==1) tus=78;
if(i==1 && j==2) tus=79;
}}
/* harf onay tuşu*/
if (input(sut4)) // 4. sütun okunuyor
{
while(input(sut4));
if(r==1 && MENU==2) // satır 1 için tuştakımı veri girişi
{
k++;
if (k<17) // lcd 16 haneli olduğundan sınırlayıcı if koşulu
{
lcd_hazirla();
/* satırın sütunundan önceki kayıtlı harflerin tekrar ekrana
yazılma işlemi yapılmaktadır. Bbuun 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*/

for(yzc=0;yzc<k;yzc++) printf(lcd_veri,"%c",abc[yzc-2]);
imlec(r,k);
abc[k-2]=tus;

}
if(k==17)
{
k=16;

}

}
if(r==2 && MENU==2) // satır 2 için tuştakımı veri girişi
{
m++;
if(m==15)
{
m=14;
}
lcd_hazirla();
for(yzc=0;yzc<m;yzc++) {printf(lcd_veri,"%c",abc[24+yzc-2]);}
imlec(2,m);

if (abc[24+m-2]!=' ') {tus=abc[24+m-2];}
abc[24+m-2]=tus;

}

if(r==3 && MENU==2) // satır 1 için tuştakımı veri girişi
{
n++;
if(n==15)

```



```

    {
        n=14;
    }
    lcd_hazirla();
    for(yzc=0;yzc<n;yzc++) printf(lcd_veri,"%c",abc[48+yzc-2]);
    imlec(1,n);

    if (abc[48+n-2]!=' ') {tus=abc[48+n-2];}
    abc[48+n-2]=tus;

}

if(r==4 && MENU==2) // adet giriş satırı
{
    ii=tus-48;
}
tus=' ';
j=-1;
}

output_low(sat2); // 2. satır lojik-0 yapıyor

output_high(sat3); // 3. satır lojik-1 yapıyor

if (input(sut1)) // 1. sütun okunuyor
{
    while(input(sut1))
    {
        if(i==1){delay_ms(500);j++;j=(j%3);}
        if(i==0) tus='7';
        if(i==1 && j==0) tus=80;
        if(i==1 && j==1) tus=82;
        if(i==1 && j==2) tus=83;
    }
}

if (input(sut2)) // 2. sütun okunuyor
{
    while(input(sut2))
    {
        if(i==1){delay_ms(500);j++;j=(j%3);}
        if(i==0) tus='8';
        if(i==1 && j==0) tus=84;
        if(i==1 && j==1) tus=85;
        if(i==1 && j==2) tus=86;
    }
}

if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3))
    {
        if(i==1){delay_ms(500);j++;j=(j%3);}

```

```

        if(i==0) tus='9';
        if(i==1 && j==0) tus=88;
        if(i==1 && j==1) tus=89;
        if(i==1 && j==2) tus=90;
    }
}

/* girilen hatalı harf/rakam için silme kodu*/
if (input(sut4)) // 4. sütun okunuyor
{
    while(input(sut4));
    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);
    }
    else if(r==2 && MENU==2) // satır 2 için silme kodu
    {
        m--;

        if(m==0)
        {
            m=1;
        }

        abc[24+m]=' ';

        imlec(2,m);
        printf(lcd_veri,' ');
        imlec(2,m);
    }
    else if(r==3 && MENU==2) // satır 3 için silme kodu
    {
        n--;

        if(n==0)
        {
            n=1;
            n1=1;
        }
        abc[48+n]=' ';
    }
}

```

```

        imlec(1,n); // deđiřtirdin +1 di
        printf(lcd_veri,' ');
        imlec(1,n);
    }

    j=-1;
}

output_low(sat3); // 3. satır lojik-0 yapılıyor

output_high(sat4); // 3. satır lojik-1 yapılıyor

/* yazıcının ana menüye dönüşünü sađlayan ana menü tuřu*/
if (input(sut1)) // 1. sütun okunuyor
{
    while(input(sut1));r=1;k=1,n=1;m=1;i=0;MENU=0;lcd_hazirla();for(sfr=0;sfr<70;sfr++)
{abc[sfr]=' ';}tus=' ';}
if (input(sut2)) // 2. sütun okunuyor
{
    delay_ms(20);
    if(i==0) tus='0';
    if(i==1) tus=' ';
}
/* menu 2 de tuřtakımı ile girilen veriyi yazdıran yazdır tuřu komutları*/
if (input(sut3)) // 3. sütun okunuyor
{
    while(input(sut3));

    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);
            }

            // son yazdırılan verinin karakter sayılarını hafıza yazdır
            //komutu için kullanabilmek için eeproma kaydetme komutları
            write_eeprom(75,syc);

```

```

write_eeprom(74,k);
write_eeprom(73,m);
write_eeprom(72,n);

// ÖNCE ALT SATIR YAZDIRILIYOR
//sadır 3
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(' ');
}

//sadır 2
for(bos=0;bos<((25-m)/2);bos++) // ortalama için ön boşluk
{
    putc(' ');
}
for(yzm=0;yzm<m-1;yzm++) // veri yazımı
{
    putc(abc[24+yzm]);
}
for(bos=(((25-m)/2)+m);bos<25;bos++) // ortalama için son boşluk
{
    putc(' ');
}
// sadır 1
for(bos=0;bos<((25-k)/2);bos++) // ortalama için ön boşluk
{
    putc(' ');
}
for(yzk=0;yzk<k-1;yzk++) // veri yazımı
{
    putc(abc[yzk]);
}

for(bos=(((25-k)/2)+k);bos<25;bos++) // ortalama için son boşluk

```

```

{
    putc(' ');
}
putc(0x32);
putc(0x0B);

delay_ms(1000);

}
delay_ms(500);
lcd_hazirla();
k=1;
i=0;
m=1;
n=1;
r=1;
ii=1;
    tus=' ';
imlec(1,1);
q=1;

}

}

/* satırlar arası geçisi sağlayan satır değiş tuş kodları*/
if (input(sut4)) // 4. sütun okunuyor
{
    while(input(sut4));
    r++;
    if(r==5){r=1;}
    j=-1;
    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);
        }
    if(r==2 && MENU==2) // 2. satıra gelindiğinde girilen değerlerin aynen
        // kalmasını için if koşul komutları
        {
            if(abc[24+m-1]==' ') tus=' ';
            else if (abc[24+m-1]!=' ') tus=abc[24+m-1];
            imlec(1,m);
            printf(lcd_veri, ' ');
            imlec(1,m);
        }
    if(r==3 && MENU==2) // 3. satıra gelindiğinde girilen değerlerin aynen

```

```

        // kalmasını için if koşul komutları
    {
        if(abc[48+n-1]==' ') tus=' ';
        else if (abc[48+n-1]!=' ') tus=abc[48+n-1];
        imlec(1,n);
        printf(lcd_veri, ' ');
        imlec(1,n);
    }

    tus=' ';
}
output_low(sat4); // 3. satır lojik-0 yapılıyor

return tus; // Fonksiyon "tus" değeri ile geri döner
}

//***** ANA PROGRAM FONKSİYONU*****

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
    clear_interrupt(INT_RDA);

    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
        {

```

```

    lcd_komut(0x0C);
    putc(0x02);
    imlec(1,15);

    /*farklı menuler seçildiğinde bir diğer veriden (RDA) veri gelmesini
    global önlemek için kesmeler inaktif edildi */

    printf(lcd_veri,"%c",keypad_oku());// Tuş değeri LCD'ye yazdırılıyor
    if(tus=='3')
    {
        MENU=3;
        disable_interrupts(GLOBAL);
    }
    if(tus=='2')
    {
        MENU=2;
        disable_interrupts(GLOBAL);
    }
    if(tus=='1')
    {
        MENU=1;
        enable_interrupts(INT_RDA);
        enable_interrupts(GLOBAL);
    }

    tus=' ';
}

if(MENU==3)
{
    /* son yazdırılan yazının hafızadan alınıp tekrar yazdırılması*/

    lcd_hazirla();
    printf(lcd_veri,"yazdiriliyor");
    eeprom_oku();

    syc++;
    if(syc==7)
    {
        syc=0;
        putc(0x31);
        putc(0x0B);
    }
    write_eeprom(75,syc); // her yazdırmada syc güncellenmektedir

    /* normal yazdırma fonksiyonu aynen aktarılıp, burada hafızaya
    özgü değişkenler kullanılmıştır*/

```

```

for(bos=0;bos<((25-n1)/2);bos++)
{
    putc(' ');
}

for(yzm=0;yzm<n1-1;yzm++)
{
    putc(abc1[48+yzm]);
}

for(bos=((25-n1)/2)+n1;bos<25;bos++)
{
    putc(' ');
}

for(bos=0;bos<((25-m1)/2);bos++)
{
    putc(' ');
}
for(yzm=0;yzm<m1-1;yzm++)
{
    putc(abc1[24+yzm]);
}
for(bos=((25-m1)/2)+m1;bos<25;bos++)
{
    putc(' ');
}

for(bos=0;bos<((25-k1)/2);bos++)
{
    putc(' ');
}
for(yzk=0;yzk<k1-1;yzk++)
{
    putc(abc1[yzk]);
}

for(bos=((25-k1)/2)+k1;bos<25;bos++)
{
    putc(' ');
}
putc(0x32);
putc(0x0B);
delay_ms(500);
MENU=0;
if (MENU==0) goto MENUU;
// yazdırma işleminden sonra ana menüye dönülüyor

}

```



```

// tuş takımı ile yazdırma menü içeriği
if(MENU==2)
{
    lcd_hazirla(); // LCD hazırlanıyor
    for(sfr=0;sfr<72;sfr++) {abc[sfr]=' ';}

    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);

        /* yazdırma işlemi tamamlandığında eeproma atıldıktan sonra q değişkeni
        1 yapılıyor ve iş koşulu ile tüm yazılanlar sıfırlanarak boş ekrana
        dönüş yapılıyor */
        if (q==1)
        {
            eeprom_yaz();
            q=0;
            for(sfr=0;sfr<72;sfr++)
            {
                abc[sfr]=' ';
            }
        }

        if(r==1) // satır 1 ekrana yazdırılıyor
        {
            for(yzc=0;yzc<k-1;yzc++)
            {
                printf(lcd_veri,"%c",abc[yzc]);
            }
            imlec(1,k);
        }
        if(r==2) // satır 2 ekrana yazdırılıyor
        {

```

```

        for(yzc=0;yzc<m-1;yzc++)
        {
            printf(lcd_veri,"%c",abc[24+yzc]);
        }
        imlec(1,m);
    }
    if(r==3) // satır 3 ekrana yazdırılıyor
    {
        for(yzc=0;yzc<n-1;yzc++)
        {
            printf(lcd_veri,"%c",abc[48+yzc]);
        }
        imlec(1,n);
    }

    lcd_komut(0x0E);
    delay_ms(5);
    if (keypad_oku(>9)
    {
        lcd_komut(0x0C);
        putc(0x02);
        printf(lcd_veri,"%c",keypad_oku()); // Tuş değeri LCD'ye yazdırılıyor
    }
    if (MENU==0) goto MENUU;
    }
} //MENU SEÇİMİ MENÜ=0 ise

/* RDA kesmesi ile RS232 haberleşmesi menüsü */
if(MENU==1)
{
    lcd_hazirla();
    printf(lcd_veri,"BAGLANTI TIPI");
    imlec(2,1);
    printf(lcd_veri,"BILGISAYAR");

    while(MENU==1)
    {
        keypad_oku(>9; // sadece ana manu tuşunu kontrol için vardır
    }
    if (MENU==0) goto MENUU;

}
}

```

### Ek 3. LCD Kütüphane Kodları

```

////////////////////////////////////
//          benim_LCD.c          //
// Bu LCD sürücü dosyası 2x16 HD44780 uyumlu LCD'ler //
// için yazılmıştır. LCD'ye sadece yazma yapılacağı için //

```

```

// R/W ucu şaseye bağlanmalıdır. LCD bağlantısı aşağıdadır. //
////////////////////////////////////
// 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

//***** LCD'ye Komut Gönderme Fonksiyonu *****
void lcd_komut(byte komut)
{
    output_b(komut>>4); // Komutun yüksek değerli 4 bitini gönder
    output_low(rs); // LCD komut almak için ayarlandı
    delay_cycles(1); // 1 komut saykılı bekle
    output_high(e); // E ucu lojik-1'den lojik-0'a çekiliyor
    delay_cycles(1); // 1 komut saykılı bekle
    output_low(e);
    delay_ms(2); // 2 msn gecikme veriliyor
    output_b(komut&0x0F); // Komutun düşük değerli 4 bitini gönder
    output_low(rs); // LCD veri almak için ayarlandı
    delay_cycles(1); // 1 komut saykılı bekle
    output_high(e); // E ucu lojik-1'den lojik-0'a çekiliyor
    delay_cycles(1); // 1 komut saykılı bekle
    output_low(e);
    delay_ms(2); // 2 msn gecikme veriliyor
}

//***** LCD'ye Veri Gönderme Fonksiyonu *****
void lcd_veri(byte veri)
{
    output_b(veri>>4); // Verinin yüksek değerli 4 bitini gönder
    output_high(rs); // LCD veri almak için ayarlandı
    delay_cycles(1); // 1 komut saykılı bekle
    output_high(e); // E ucu lojik-1'den lojik-0'a çekiliyor
    delay_cycles(1); // 1 komut saykılı bekle
    output_low(e);
    delay_ms(2); // 2 msn gecikme veriliyor
    output_b(veri&0x0F); // Verinin düşük değerli 4 bitini gönder
    output_high(rs); // LCD veri almak için ayarlandı
    delay_cycles(1); // 1 komut saykılı bekle
    output_high(e); // E ucu lojik-1'den lojik-0'a çekiliyor
    delay_cycles(1); // 1 komut saykılı bekle
    output_low(e);
    delay_ms(2); // 2 msn gecikme veriliyor
}

//***** LCD'de İmlec Konumlandırma Fonksiyonu *****

```

```

void imlec(byte satir, byte sutun)
{
    if (satir==1)    // Eğer satır değişkeni "1" ise
        lcd_komut(0x80|(sutun-1));

    if (satir==2)    // Eğer satır değişkeni "2" ise
        lcd_komut(0x80|(0x40+(sutun-1)));
}

//***** LCD Başlangıç Ayarları Fonksiyonu *****
void lcd_hazirla()
{
    int i=0;
    output_low(rs); // RS ucu lojik-0
    output_low(e); // E ucu lojik-0
    delay_ms(30); // LCD enerjilendiğinde LCD'nin hazır olması için beklenen süre
    for(i=0;i<=3;i++) // LCD'ye 3 kez 0x03 komutu gönderiliyor
    {
        lcd_komut(0x03);
        delay_ms(5); // 5msn gecikme veriliyor
    }
    lcd_komut(0x02); // LCD'ye 4 bit iletişim komutu gönderiliyor
    lcd_komut(0x28); // 4 bit iletişim, 2 satır, 5x8 dot matris seçildi
    lcd_komut(0x08); // Display Kapalı
    lcd_komut(0x0E); // Display açık,imleç alt çizgi ve yanıp sönme yok
    // lcd_komut(0x06); // Her veri yazıldığında imleç bir sağa gitsin
    lcd_komut(0x01); // Display sil. İmleç 1.satır 1.sütunda
}

```

#### Ek 4. Gude Kodları

```

%***** AÇIKLAMA *****
% RS232 haberleşmesi sağlayan, 16f887A mikrodnetleyici kullanan panel
% yazıcı projesinin Matlab arayüzü tasarım mfile dosyası
% proje GUI ara yüzünde 2 adet popupmenu ile com ve yazının konumu seçimi
% sağlanmıştır.
% 3 satır veri girişi için 3 ayrı edittext konulmuş, ayrıca yazı adeti için
% de ayrı bir edittext kullanılmıştır.
% 3 ayrı yazdırma butonu ve 1 adet temizleme butonu vardır
% yazdırma butonu 1 --> 3 satır yazı yazdırma
% yazdırma butonu 2 --> 1 satır yazı diğer 2 satır alanına logo
% yazdırma butonu 3 --> tüm alana sadece logo

```

```

%***** mfile kodlarının açıklamaları hakkında bilgi*****

% dosya incelendiğinde (yukarıdan - aşağıya) karşımıza çıkan ilk kodun
% açıklaması yapılmış olup gerek görülmediği takdirde 2.bir açıklama
% yapılmamıştır. yapılan ilk açıklamanı artık biliniyor kabul edilerek
% diğer callback manulerinde tetrardan açıklanmamıştır.

function varargout = mesut(varargin)

% MESUT MATLAB code for mesut.fig

% MESUT, by itself, creates a new MESUT or raises the existing
% singleton*.

% H = MESUT returns the handle to a new MESUT or the handle to
% the existing singleton*.

% MESUT('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in MESUT.M with the given input arguments.

% MESUT('Property','Value',...) creates a new MESUT or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before mesut_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to mesut_OpeningFcn via varargin.

% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".

% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help mesut

% Last Modified by GUIDE v2.5 08-Jun-2015 04:35:59

% Begin initialization code - DO NOT EDIT

gui_Singleton = 1;

gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @mesut_OpeningFcn, ...

```

```

        'gui_OutputFcn', @mesut_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before mesut is made visible.
function mesut_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin   command line arguments to mesut (see VARARGIN)
% Choose default command line output for mesut
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
%***** AÇIKLAMA *****
% seçilen ayarların tüm callbacklerde kullanılması için değişkenler global
% tanımlanmıştır
global ss1;
ss1=1;

```

```

global s_yatik;

s_yatik=1;

global adet;

adet=0;

% UIWAIT makes mesut wait for user response (see UIRESUME)

% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.

function varargout = mesut_OutputFcn(hObject, eventdata, handles)

% varargout cell array for returning output args (see VARARGOUT);

% hObject handle to figure

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure

varargout{1} = handles.output;

% --- Executes on button press in pushbutton1.

function pushbutton1_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%***** AÇIKLAMA *****

global ss1;

global s_yatik;

global adet;

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

% girilen adet string olarak alındığından double formatına çevrildi

yukleme_cubugu = waitbar(0,'YAZDIRILIYOR. LÜTFEN BEKLEYİNİZ...');

% sadece yazı yazdırma bölümü için waitbar tanımlandı

% for i=0:adet % yazdırma adeti

```

```

% tüm satırlar strin olarak alınıyor
satir1=(get(handles.edit1, 'String'));
satir2=(get(handles.edit2, 'String'));
satir3=(get(handles.edit3, 'String'));

% yazıcı kapasitesi nedeni ile 24 karakter ile sınırlandırıldı
a1=length(satir1);

if a1>24

    a1=24;

end

a2=length(satir2);

if a2>24

    a2=24;

end

a3=length(satir3);

if a3>24

    a3=24;

end

for ii=1:adet

% Secilen COM ayarı ile serial için COM seçimi yapılıyor

if ss1 == 1;

s1=serial('COM13');

end

if ss1 == 2;

s1=serial('COM2');

end

if ss1 == 3;

s1=serial('COM3');

end

```



```

if ss1== 4;

s1=serial('COM4');

end

if ss1== 5;

s1=serial('COM5');

end

if ss1 == 6;

s1=serial('COM6');

end

if ss1 == 7;

s1=serial('COM7');

end

if ss1 == 8;

s1=serial('COM8');

end

if ss1== 9;

s1=serial('COM9');

end

if ss1== 10;

s1=serial('COM10');

end

%serial konfigürasyon ayarları

set(s1,'BaudRate', 9600, 'DataBits', 8, 'Parity', 'none','StopBits', 1, 'FlowControl', 'none');

fopen(s1);

% mikroişlemcinin yazı gönderildiğini algılaması için hatta öncelikle 'Y'

% konuluyor ve yazının solda sağda ve ortoda olma durumları sırası ile;

% ***** SOLA DAYALI *****

% yazı solda ise ve satır boş değil ise, ilk önce yazılan yazı yazdırılıp

```

```

% sonrasında ise, boş karakter gönderebilmek için tanımlanan satır_bos
% dizisinden-girilen karakter sayısı itibari ile- 24 karektere tamamlama
% işlemi yapılıyor.
% eğer o satır boş işe yazıcı kaymasın diye . karakterli dizi gönderiliyor.
% tüm stırlarda bu işlemler yineleniyor
%***** SAĞA DAYALI *****
% yazı sağda ise ve satır boş değil ise, ilk önce boş karakter dizisi aynı
% satırın sonuna girilen yazının yazdırılmasını sağlamak için,24-girilen
% karakter sayısı, adedince boş karakter gönderilip sonrasında
% girilen yazı hatta konuluyor
%***** ORTADA *****
% eger yazı ortada ise 24 ile sınırlı olan satırdan girilen karakter sayısı
% çıkartılarak sağdan ve soldan eşit boşluk bırakabilmek için hatta; önce
% toplam boşluk sayısının yarısı kadar boşluk, sonra girilen yazı, daha
% sonrasında ise 24 e tamamlamak için diğer yarım kadar boşluk
% konulmaktadır.
%****TÜM SATIRLARDA YUKARIDAKI AÇIKLAMALAR TEKRARLANMIŞTIR*****
% satır 3
fwrite(s1,'Y');
if s_yatik==1
    if a3 ~= 0
        satir3_bos='          ';
        fwrite(s1,satir3(1,1:a3));
        fwrite(s1,satir3_bos(1,(a3+1):24));
    end
    if a3 == 0
        satir3_bos='.';
        fwrite(s1,satir3_bos);

```

```

        end

    end

    if s_yatik==2

        if a3 ~= 0

            satir3_bos='          ';

            fwrite(s1,satir3_bos(1,1:24-a3));

            fwrite(s1,satir3(1,1:a3));

            end

        if a3 == 0

            satir3_bos='          ';

            fwrite(s1,satir3_bos);

            end

        end

    end

    if s_yatik==3

        if a3 ~= 0

            satir3_bos='          ';

            fwrite(s1,satir3_bos(1,1:fix((24-a3)/2)));

            fwrite(s1,satir3(1,1:a3));

            fwrite(s1,satir3_bos(1,(fix((24-a3)/2)+a3+1):24));

            end

        if a3 == 0

            satir3_bos='          ';

            fwrite(s1,satir3_bos);

            end

        end

    end

    % satir 2

```

```

if s_yatik==1

    if a2 ~= 0

        satir2_bos='          ';

        fwrite(s1,satir2(1,1:a2));

        fwrite(s1,satir2_bos(1,(a2+1):24));

    end

    if a2 == 0

        satir2_bos='          ';

        fwrite(s1,satir2_bos);

    end

end

if s_yatik==2

    if a2 ~= 0

        satir2_bos='          ';

        fwrite(s1,satir2_bos(1,1:24-a2));

        fwrite(s1,satir2(1,1:a2));

    end

    if a2 == 0

        satir2_bos='          ';

        fwrite(s1,satir2_bos);

    end

end

if s_yatik==3

    if a2 ~= 0

        satir2_bos='          ';

        fwrite(s1,satir2_bos(1,1:(fix((24-a2)/2))));

        fwrite(s1,satir2(1,1:a2));

        fwrite(s1,satir2_bos(1,(fix((24-a2)/2)+a2-1):24));

    end

end

```

```

end

if a2 == 0

    satir2_bos='          ';

    fwrite(s1,satir2_bos);

end

end

% satir 1

if s_yatik==1

    if a1 ~= 0

        satir1_bos='          ';

        fwrite(s1,satir1(1,1:a1));

        fwrite(s1,satir1_bos(1,(a1+1):24));

    end

    if a1 == 0

        satir1_bos='          ';

        fwrite(s1,satir1_bos);

    end

end

if s_yatik==2

    if a1 ~= 0

        satir1_bos='          ';

        fwrite(s1,satir1_bos(1,1:24-a1));

        fwrite(s1,satir1(1,1:a1));

    end

    if a1 == 0

        satir1_bos='          ';

        fwrite(s1,satir1_bos);

    end

end

```

```

end

if s_yatik==3

    if a1 ~= 0

        satir1_bos='          ';

        fwrite(s1,satir1_bos(1,1:(fix(((24-a1)/2)-2))));

        fwrite(s1,satir1(1,1:a1));

        fwrite(s1,satir1_bos(1,(fix((24-a1)/2)+a1-1):24));

    end

    if a1 == 0

        satir1_bos='          ';

        fwrite(s1,satir1_bos);

    end

end
end

```

```

waitbar(((100/adet)*i)/100); % waitbar güncelleme komutu

```

```

fclose(s1)

delete(s1)

pause(1)

end

close (yukleme_cubugu) % waitbar kapatma komutu

```

```

%kullanıcı bilgilendirme kutusu

msgbox('Yazdırma Tamamlandı','YAZI YAZDIRMA','help')

```

```

function edit1_Callback(hObject, eventdata, handles)

% hObject    handle to edit1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double
% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)

% hObject    handle to edit3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%    str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.

function edit3_CreateFcn(hObject, eventdata, handles)

% hObject    handle to edit3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.

function pushbutton3_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton3 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton4.

```



```

function pushbutton4_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

global ss1;
global adet;

%***** AÇIKLAMA *****
% .jpg formatındaki resmi logoyu seçme işlemi
[filename,pathname] = uigetfile('*.*','JPEG dosyasini seciniz');
asd=imread([pathname,filename]);
%seçilen resim verileri değişkene atılıyor
asd_bw = im2bw(asd,125/255);
% resim siyah-beyaz formata çevriliyor
RGB = imresize(asd_bw, [128 384]);
% yeniden ölçeklendiriliyor
imshow(RGB);
RGB=1-RGB;
% girilen değerler gönderilirken terslendiğinden biz de gerçek
% veriyi kaybetmemek için 1 den çıkartıyoruz
RGB180=imrotate(RGB,-180);
%yazıcı formatı gereği 180 derece döndürülerek veri gönderiliyor
adet=str2double(get(handles.edit5,'String'));
for ii=1:adet

%***** AÇIKLAMA *****
% 128*384 formatındaki resim 0-1 lere dönüştürüldükten sonra sol bitleri
% msb olarak çekilde byte byte gönderebilmek için bi2de kodu ile
% 128*48 byte veri elde ediliyor.

```

```

for i=1:128

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');

a(i,9)=bi2de(RGB180(i,65:72),'left-msb');

a(i,10)=bi2de(RGB180(i,73:80),'left-msb');

a(i,11)=bi2de(RGB180(i,81:88),'left-msb');

a(i,12)=bi2de(RGB180(i,89:96),'left-msb');

a(i,13)=bi2de(RGB180(i,97:104),'left-msb');

a(i,14)=bi2de(RGB180(i,105:112),'left-msb');

a(i,15)=bi2de(RGB180(i,113:120),'left-msb');

a(i,16)=bi2de(RGB180(i,121:128),'left-msb');

a(i,17)=bi2de(RGB180(i,129:136),'left-msb');

a(i,18)=bi2de(RGB180(i,137:144),'left-msb');

a(i,19)=bi2de(RGB180(i,145:152),'left-msb');

a(i,20)=bi2de(RGB180(i,153:160),'left-msb');

a(i,21)=bi2de(RGB180(i,161:168),'left-msb');

a(i,22)=bi2de(RGB180(i,169:176),'left-msb');

a(i,23)=bi2de(RGB180(i,177:184),'left-msb');

a(i,24)=bi2de(RGB180(i,185:192),'left-msb');

a(i,25)=bi2de(RGB180(i,193:200),'left-msb');

a(i,26)=bi2de(RGB180(i,201:208),'left-msb');

a(i,27)=bi2de(RGB180(i,209:216),'left-msb');

```

```

a(i,28)=bi2de( RGB180(i,217:224),'left-msb');
a(i,29)=bi2de( RGB180(i,225:232),'left-msb');
a(i,30)=bi2de( RGB180(i,233:240),'left-msb');
a(i,31)=bi2de( RGB180(i,241:248),'left-msb');
a(i,32)=bi2de( RGB180(i,249:256),'left-msb');
a(i,33)=bi2de( RGB180(i,257:264),'left-msb');
a(i,34)=bi2de( RGB180(i,265:272),'left-msb');
a(i,35)=bi2de( RGB180(i,273:280),'left-msb');
a(i,36)=bi2de( RGB180(i,281:288),'left-msb');
a(i,37)=bi2de( RGB180(i,289:296),'left-msb');
a(i,38)=bi2de( RGB180(i,297:304),'left-msb');
a(i,39)=bi2de( RGB180(i,305:312),'left-msb');
a(i,40)=bi2de( RGB180(i,313:320),'left-msb');
a(i,41)=bi2de( RGB180(i,321:328),'left-msb');
a(i,42)=bi2de( RGB180(i,329:336),'left-msb');
a(i,43)=bi2de( RGB180(i,337:344),'left-msb');
a(i,44)=bi2de( RGB180(i,345:352),'left-msb');
a(i,45)=bi2de( RGB180(i,353:360),'left-msb');
a(i,46)=bi2de( RGB180(i,361:368),'left-msb');
a(i,47)=bi2de( RGB180(i,369:376),'left-msb');
a(i,48)=bi2de( RGB180(i,377:384),'left-msb');
end

set(handles.edit1, 'String','LOGO BASIMINDA');

set(handles.edit2, 'String','SATIRLAR');

set(handles.edit3, 'String','KULLANILMAMAKTADIR');

if ss1 == 1;

s1=serial('COM13');

end

```

```

if ss1 == 2;

s1=serial('COM2');

end

if ss1 == 3;

s1=serial('COM3');

end

if ss1== 4;

s1=serial('COM4');

end

if ss1== 5;

s1=serial('COM5');

end

if ss1 == 6;

s1=serial('COM6');

end

if ss1 == 7;

s1=serial('COM7');

end

if ss1 == 8;

s1=serial('COM8');

end

if ss1== 9;

s1=serial('COM9');

end

if ss1== 10;

s1=serial('COM10');

end

set(s1,'BaudRate', 9600, 'DataBits', 8, 'Parity', 'none','StopBits', 1, 'FlowControl', 'none');

```

```

% elde edilen 128*48 byte 48byte lar halinde 128 kez gönderiliyor

for j=1:128

    fopen(s1)

    fwrite(s1,'M');

    for i=1:48

        fwrite(s1,a(j,i));

    end

    fclose(s1)

    pause(0.0001)

end

delete(s1)

end

cla reset % yazdırma sonrası axes temizleniyor

msgbox('Yazdırma Tamamlandı','LOGO YAZDIRMA','custom',asd)


% --- Executes on button press in pushbutton6.

function pushbutton6_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton6 (see GCBO)

% eventdata  reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global ss1;

global s_yatik;

global adet;

[filename,pathname] = uigetfile('*.jpg','JPEG dosyasini seciniz');

asd=imread([pathname,filename]);

imshow(asd);

asd_bw = im2bw(asd,125/255);

RGB = imresize(asd_bw, [64 64]);

```

```

RGB=1-RGB;

RGB180=imrotate(RGB,-180);

set(handles.edit2, 'String','LOGO İÇİN ');

set(handles.edit3, 'String','KULLANILMAKTADIR');

% 64*8byte veri dönüşümü

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');

end

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

for ii=1:adet

if ss1 == 1;

s1=serial('COM13');

end

if ss1 == 2;

s1=serial('COM2');

end

if ss1 == 3;

s1=serial('COM3');

end

if ss1== 4;

s1=serial('COM4');

```

```

end

if ss1== 5;

s1=serial('COM5');

end

if ss1 == 6;

s1=serial('COM6');

end

if ss1 == 7;

s1=serial('COM7');

end

if ss1 == 8;

s1=serial('COM8');

end

if ss1== 9;

s1=serial('COM9');

end

if ss1== 10;

s1=serial('COM10');

end

set(s1,'BaudRate', 9600, 'DataBits', 8, 'Parity', 'none','StopBits', 1, 'FlowControl', 'none');

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)

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

a1=length(satir1);

if a1>24

    a1=24;

end

if ss1 == 1;

s1=serial('COM13');

end

if ss1 == 2;

s1=serial('COM2');

end

if ss1 == 3;

s1=serial('COM3');

end

if ss1== 4;

s1=serial('COM4');

end

if ss1== 5;

s1=serial('COM5');

end

if ss1 == 6;

s1=serial('COM6');

end

if ss1 == 7;

s1=serial('COM7');

end

```



```

if ss1 == 8;

s1=serial('COM8');

end

if ss1== 9;

s1=serial('COM9');

end

if ss1== 10;

s1=serial('COM10');

end

set(s1,'BaudRate', 9600, 'DataBits', 8, 'Parity', 'none','StopBits', 1, 'FlowControl', 'none');

fopen(s1);

fwrite(s1,'W');

if s_yatik==1

    if a1 ~= 0

        satir1_bos='          ';

        fwrite(s1,satir1(1,1:a1));

        fwrite(s1,satir1_bos(1,(a1+1):24));

    end

    if a1 == 0

        satir1_bos='          ';

        fwrite(s1,satir1_bos);

    end

end

if s_yatik==2

    if a1 ~= 0

        satir1_bos='          ';

```

```

fwrite(s1,satir1_bos(1,1:24-a1));

fwrite(s1,satir1(1,1:a1));

end

if a1 == 0

satir1_bos='          ';

fwrite(s1,satir1_bos);

end

end

if s_yatik==3

if a1 ~= 0

satir1_bos='          ';

fwrite(s1,satir1_bos(1,1:((24-a1)/2)));

fwrite(s1,satir1(1,1:a1));

fwrite(s1,satir1_bos(1,(((24-a1)/2)+4):24));

end

if a1 == 0

satir1_bos='          ';

fwrite(s1,satir1_bos);

end

end

set(handles.edit2, 'String','LOGO İÇİN ');

set(handles.edit3, 'String','KULLANILMAKTADIR');

fclose(s1)

delete(s1)

end

cla reset

msgbox('Yazdırma Tamamlandı','LOGOLU YAZDIRMA','custom',asd)

% --- Executes on selection change in popupmenu1.

```

```

function popupmenu1_Callback(hObject, eventdata, handles)

% hObject    handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles    structure with handles and user data (see GUIDATA)

global ss1;

%***** AÇIKLAMA *****

% popupmenudeki seçilen değer switch case ile global ss1 e aktarılıyor. Bu

% işlem sonucu COM seçimi yapılmaktadır

val=get(hObject,'Value');

switch val

    case 1

        ss1=1;

    case 2

        ss1=2;

    case 3

        ss1=3;

    case 4

        ss1=4;

    case 5

        ss1=5;

    case 6

        ss1=6;

    case 7

        ss1=7;

    case 8

        ss1=8;

    case 9

        ss1=9;

```

```

    case 10

        ss1=10;

    end

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array
%     contents{ get(hObject,'Value')} returns selected item from popupmenu1
% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu2.
function popupmenu2_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global s_yatik;

%***** AÇIKLAMA *****
% popupmenudeki seçilen değer switch case ile global s_yatik a aktarılıyor.
% Bu işlem sonucu sağa yatık-sola yatık- merkez seçimi yapılmaktadır

val1=get(hObject,'Value');

switch val1

    case 1

        s_yatik=1;

```

```

case 2

    s_yatik=2;

case 3

    s_yatik=3;

end

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu2
% contents as cell array5
%     contents{ get(hObject,'Value')} returns selected item from popupmenu2
% --- Executes during object creation, after setting all properties.
function popupmenu2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of edit5 as text
%     str2double(get(hObject,'String')) returns contents of edit5 as a double
% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.

%    See ISPC and COMPUTER.

if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton7.

function pushbutton7_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%_***** AÇIKLAMA *****
% bu işlem ile edit textler temizlenmektedir.(not:adet değiştirilmemektedir)

set(handles.edit1, 'String','');
set(handles.edit2, 'String','');
set(handles.edit3, 'String','');

```

## Ek 5. KULLANIM KILAVUZU



Şekil 36: Yazıcının arka kısmı

- 1- Besleme Tuşu
- 2- Durum ledi ve Açma Tuşu
- 3- Çıktı Aparatı
- 4- Bağlantısı Kablosu
- 5- Güç Kablosu



**Şekil 37: Yazıcının LCD ve Keypadin bulunduğu ön kısım**

- 1- Yazdırma Ekranı
- 2- Tuş Takımı

Yazıcımız, üzerinde bulunan tuş takımı ve bağımsız olarak çalışan matlab gui arayüzü olmak üzere iki farklı şekilde çalışabilmektedir. ilk başlık altında tuş takımı tanıtımı ve yazdırma uygulaması yapılacaktır. Diğer bir başlıkta ise Matlab arayüzü tanıtılacaktır.



## Ek 5.1 Tuş takımı tabanlı çalışma

### Ek 5.1.1 Tuş takımı tanıtımı ve fonksiyonu

. : /	abc	def	R/H
1	2	3	A
ghi	jkl	mno	OK
4	5	6	B
prs	tuv	xyz	del
7	8	9	C
menu	space	print	line
*	0	#	D

Şekil 38:Tuş Takımı

TUŞ 1: 1 ya da . : /

TUŞ 2: 2 ya da a b c

TUŞ 3: 3 ya da d e f

TUŞ 4: 4 ya da g h i

TUŞ 5: 5 ya da j k l

TUŞ 6: 6 ya da m n o

TUŞ 7: 7 ya da p r s

TUŞ 8: 8 ya da t u v

TUŞ 9: 9 ya da x y z

TUŞ 0: 0 ya da boşluk

TUŞ A: R/H ( rakam/harf )

TUŞ B: OK (girilen her harften sonra onaylama yapılmalıdır)

TUŞ C: silme

TUŞ D: satırlar arası geçiş

TUŞ \*: ana menü

TUŞ #: yazdırma

## Ek 5.2 Program kullanım örneği ve ana menü görüntüsü

Yazıcı açıldığında karşınıza Şekil 39 deki gibi ana menü açılacaktır. Bilgisayar bağlantısı seçimi için tuş 1'e, pic ile elle yazı yazdırmak için tuş 2'ye, en son yazdırılan yazıyı hafızadan yazdırabilmek için tuş 3'e basılır.



Şekil 39: Ana menü ekran görüntüsü

### Ek 5.3 Menü 1 seçimi

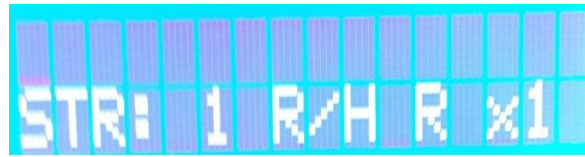
Menü 1 seçildiği zaman yazıcı bilgisayar bağlantısı ile alınan verinin yazılımı moduna geçer ve ana menü tuşu ile tekrar menü seçim ekranına dönlür. Menü 1 durumundaki ekran görüntüsü Şekil 40da gösterilmiştir.



Şekil 40: Menü 1 ekran görüntüsü

### Ek 5.4 Menü 2 seçimi

Menü 2 yazıcıya tuş takımı ile yazdırmak için kullanılmaktadır. Menü 2'nin ekran görüntüsü Şekil 41'ta gösterilmiştir.



Şekil 41: Menü 2 ekran görüntüsü

Şekil 40'ta gösterildiği gibi ilk satır girilecek yazı için ayrılmıştır. İkinci satır ise kullanıcıyı bilgilendirmek için kullanılmıştır. Bu satırdaki STR: kullanıcının o an bulunduğu satırı belirtir. R/H kullanıcının tuş takımı ile harf girişi mi, yoksa rakam girişi mi yaptığı gösterilmektedir. x1 ile gösterilen yer ise yazılacak yazının adetini göstermektedir. Aşağıda bu menüde yapılabilecek işlemlerin ekran görüntüleri ve açıklamaları bulunmaktadır.

- **Satir deęiřtirme (tuř D)**

Tuř D ile bulunduęunuz satırdan tekrarlı basmanız durumunda sıra ile satır1,satır2,satır3,satır4 geiřlerini gerekleřtirebilirsiniz.

Menü2 ekranında satırlar arası geiř örneęi;

Satır1’den satır2’e gemek için Tuř D’ye 1 defa basmak yeterlidir



řekil 42:Satır2 ekran görüntüsü

Dięer geiřler için örneğ;

Satır1’den satır3’e gemek için Tuř D’ye 2 defa basmak yeterlidir

Satır1’den satır4’e gemek için Tuř D’ye 3 defa basmak yeterlidir

(geiřler esnasında STR: 1 Tuř D’ye her basıldıęında deęiřmektedir)

- **Rakam/Harf seimi (Tuř A)**

Tuř A ile bulunduęunuz satırın istedięiniz sütununda harften rakama, rakamdan harfe geiř yapabilirsiniz.

Menü 2 ekranında harf rakam örneęi;

1+OK +2+OK +3+OK +A+OK +2(A) +OK +2(A) +2(B) +OK +2(A) +2(B) +2(C)



řekil 43: Satır1 ekran görüntüsü

- **x1 Adet yazdırma ayarı**

Yukarıda verilen örneğin 5 adet yazdırılabilmesi için ilave olarak satır 4'e gidilip 5 tuşunu onayladıktan sonra satır 1 e dönülmesi ile oluşan ekran görüntüsü ve tuş kombinasyonu;  
1+OK +2+OK +3+OK + TUŞ A+OK +2(A) +OK +2(A) +2(B) +OK +2(A) +2(B) +2(C) +Tuş D +Tuş D +Tuş D  
+ TUŞ A + 5 + OK + TUŞ D



Şekil 44: Satır1 ekran görüntüsü

- **Silme işlemi**

Yukarıda ki örnekten c harfini silmek için mevcut kombinasyona Tuş C'yi eklemek yeterlidir



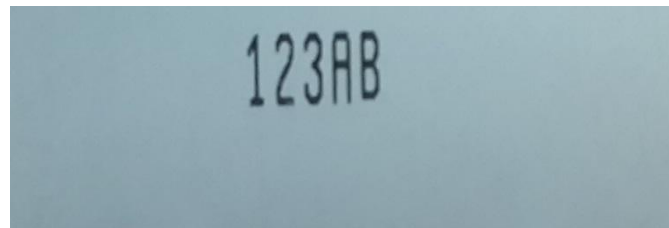
Şekil 45 : Satır1 ekran görüntüsü (+ Tuş C)

- **Yazdırma işlemi**

Mevcut tuş kombinasyonuna Tuş # eklendiğinde yazma işlemi yapılır. 5 adet çıktı alınır. Ekran görüntüsü ve çıktı örneği aşağıdaki gibidir .



Şekil 46:Yazdırma ekranı (+ Tuş C)



Şekil 47: Çıktı örneği

### **Ek 5.5 Menü 3 seçimi**

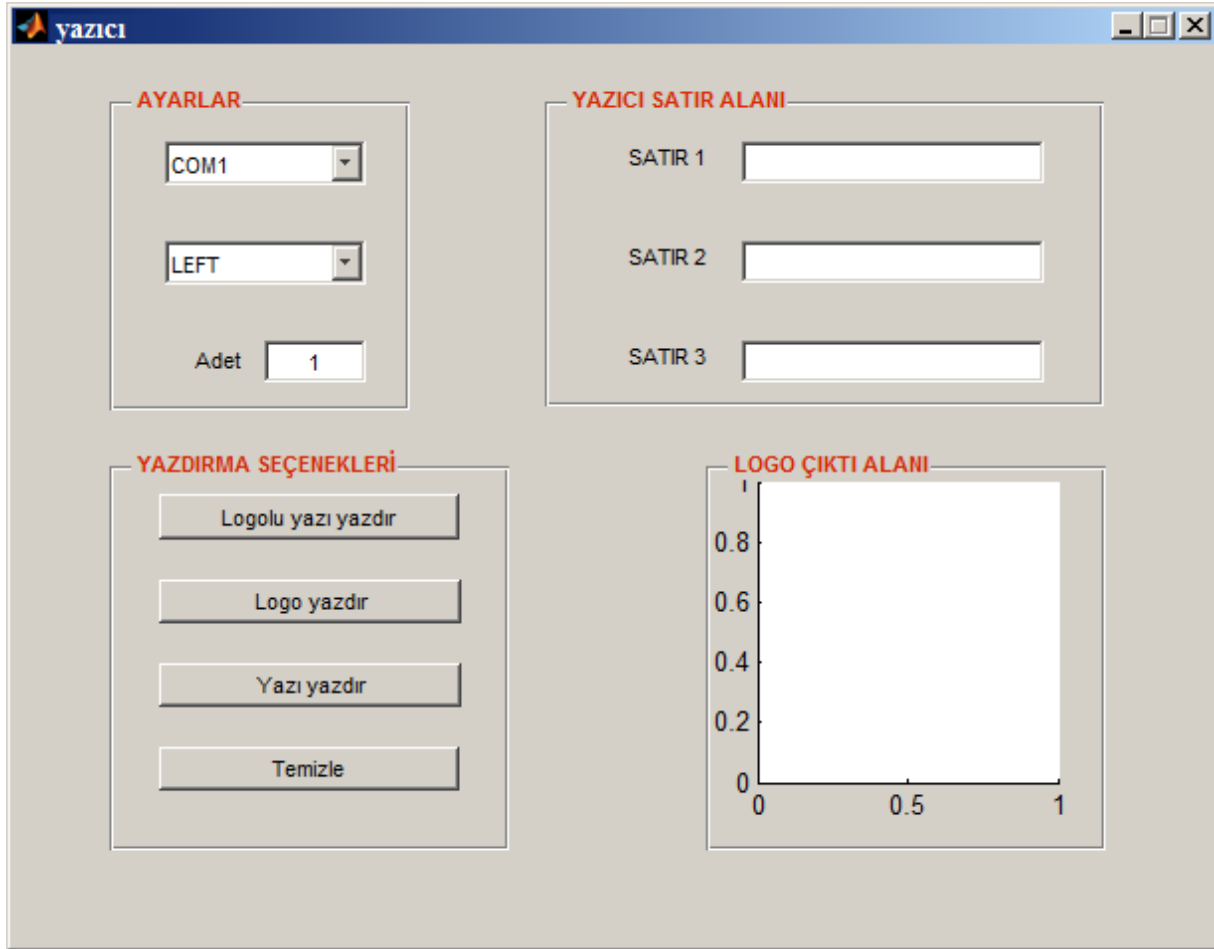
Menü 3 yazdırılan sonra yazının mikrodnetleyicinin dahili eepromuna atılması sonucu tekrardan yazdırabilmek için bulunmaktadır. Tuş 3'e basıldığı sürece tekrarlanabilir ve çıktı örneği yukarıdaki örneğimiz için yine Şekil 46'daki gibi olacaktır. Yazdırma esnasında kullanıcıya Şekil 49deki gibi yazdırılıyor ekranı gösterilir ve yazdırma bitiminde ana menüye dönülür.



**Şekil 48:Menü 3 yazdırılıyor ekranı**

## Ek 6 Matlab arayüzü

Yazıcımızın matlab arayüzü aşağıdaki şekilde verilmiştir.



Şekil 49: Guide Kullanıcı Arayüzü

### Ek 6.1 Ayarlar

Bu bölümde serial com seçimi, yazı adedi ve yazının başlangıç konumu seçilmektedir.

### Ek 6.2 Yazıcı satır alanı

Bu bölümde kullanıcıya sunulan yazı yazdırma alanları bulunmaktadır.

### Ek 6.3 Logo çıktıları

Bu bölümde basılacak logonun ekranda gösterimi yer almaktadır

### Ek 6.4 Yazdırma seçenekleri

Bu bölümde 4 ayrı buton bulunmaktadır.

### Ek 6.5 Temizle buton

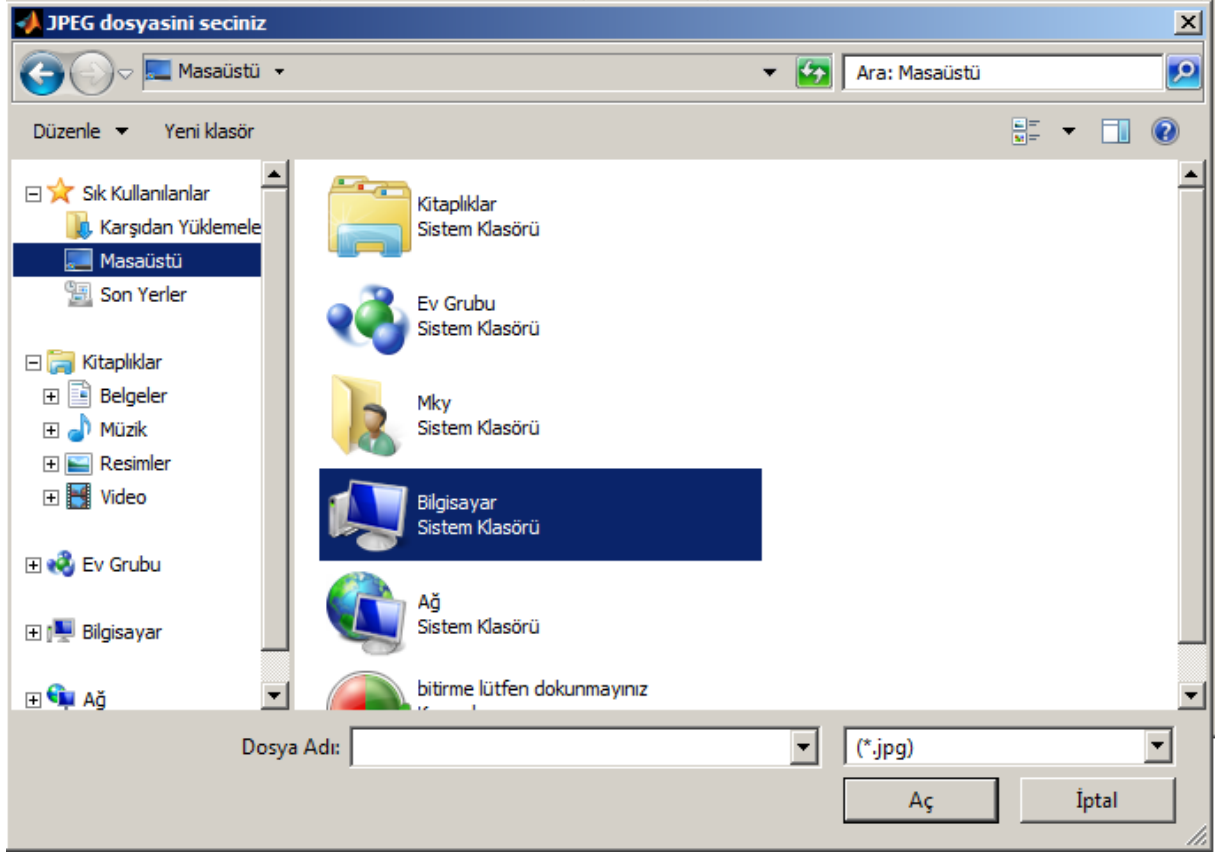
Temizle butonu kullanıcıya sunulan yazı yazdırma alanlarını silmek için kullanılmaktadır

## Ek 6.6 Yazı yazdır

Bu buton, kullanıcının yazdığı yazının direkt olarak çıktı alabilmesini sağlamaktadır

## Ek 6.7 Logo yazdır

Logo yazdırma butonuna tıklandığında ekrana aşağıdaki gibi .jpg uzantı resim seçme ekranı gelecektir. Seçim yapıp onayladıktan sonra yazım işlemi başlayacaktır.



Şekil 50:Resim Seçme Arayüzü

## Ek 6.8 Logolu yazı yazdır

Bu bölümde 1 satır yazı ve logo basımı yapılacaktır. Butona basmadan önce ilk satıra yazı girilmeli sonrasında ise butona basıp seçim ekranından istenen logo seçilmelidir. Seçim onaylandıktan sonra basım işlemi başkacaktır.

## Ek 7 Olası hatalar

1. Status led yanım sönmesi durumunda kağıdı kontrol ediniz ve kapağın tam kapalı olduğundan emin olunuz
2. Hazneye 57 mm genişlikten daha dar kağıt rulosu konulduğunda kaymalar yaşanabilir
3. Bilgisayar ara yüzünden yazım hatası alındığında yazıcının driverlerinin yüklü ve doğru com seçeneğinin seçildiğinden emin olunuz.
4. Tuş takımı kullanıldığında rakam/yazı geçişlerinden emin olunuz

## SONUÇ

Projemizde bir etiket makinesinin komple dizaynı yapılmıştır.Projede başlangıçta tasarlanan hedefe ulaşılmış ve gerekli tasarım yapılmıştır.Proje de asıl hedeflenen termal yazıcının da kendimiz tarafından yapılmasıydı ancak sürenin kısıtlı olması ve diğer bazı nedenlerden dolayı hazır alındı.



## Kaynakça

- [1] ISIL YAZICI BAŞLIKTA MATEMATİKSEL MODELLEME, Feyza KURBAN, T.C. SÜLEYMAN DEMİREL ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ, Isparta, 2007
- [2] MEGEP-Bilişim Teknolojileri-Yazıcılar, Ankara, 2007
- [3] ISIL YAZICI BAŞLIKLARIN OPTİMİZASYONU, Celâleddin SENÇİMEN, T.C. SÜLEYMAN DEMİREL ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ, Isparta, 2004
- [4] W. H. Puterbaugh ve S. P. Emmons, “A New Printing Principle”
- [5] New Type Thermal Printing Head Using Thin Film , SUSUMU SHIBATA, KEIJI MURASUGI, KATSUZO KAMINIS
- [6] <http://www.seikoinstruments.com/>
- [7] <http://www.fujitsu.com/>
- [8] Thermal printer/Payne, T.R. ; Plumlee, H.R.
- [9] Thermal printer/ Payne, T. ; Texas Instruments, Inc., Dallas, TX, USA ; Plumlee, H.

