



HAVA HARP OKULU KOMUTANLIĞI



HAVACILIK VE UZAY TEKNOLOJİLERİ ENSTİTÜSÜ YÜKSEK LİSANS TEZİ

AS-532 COUGAR HELİKOPTERİ PERFORMANS PLANLAMASININ ELEKTRONİK PLATFORM ÜZERİNDE HESAPLANMASI

FEVZİ YAZGAN
HAVA PİLOT ÜSTEĞMEN

YEŞİLYURT - İSTANBUL
HAZİRAN – 2009

HAVA HARP OKULU
HAVACILIK ve UZAY TEKNOLOJİLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

AS-532 COUGAR HELİKOPTERİ PERFORMANS PLANLAMASININ
ELEKTRONİK PLATFORM ÜZERİNDE HESAPLANMASI

Fevzi YAZGAN
Hava Pilot Üsteğmen

ELEKTRONİK ANABİLİM DALI

İSTANBUL
Haziran 2009

Her hakkı saklıdır.

Hv.Müh.Alb.Yrd.Doç.Dr. Serdar KARGIN danışmanlığında Hv. Plt. Ütğm. Fevzi YAZGAN tarafından hazırlanan bu çalışma 16/06/2009 tarihinde aşağıdaki jüri tarafından Elektronik Mühendisliği Anabilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Danışmanı : Yrd.Doç.Dr.Hv.Müh.Alb. Serdar KARGIN *İmza* :

Üye : Yrd.Doç.Dr. Murat BEKEN *İmza* :

Üye : Yrd.Doç.Dr.Hv.Müh. Ersin GÖSE *İmza* :

Yukarıdaki sonucu onaylarım.

Yrd. Doç. Dr. Sefer KURNAZ
Hava Mühendis Albay
Enstitü Müdürü

ÖZET

Yüksek Lisans Tezi

AS-532 COUGAR HELİKOPTERİ PERFORMANS PLANLAMASININ ELEKTRONİK PLATFORM ÜZERİNDE HESAPLANMASI

Fevzi YAZGAN
Hv. Plt. Ütğm.

Hava Harp Okulu
Havacılık ve Uzay Teknolojileri Enstitüsü
Elektronik Anabilim Dalı

Danışman : Hv.Yrd.Doç.Dr.Müh.Alb. Serdar KARGIN

Bu çalışmada, AS-532 Cougar helikopterinin performans grafiklerinin MATLAB yazılımı kullanılarak tek boyutlu matrisler halinde hazırlanmıştır. SD-CARD 'a yüklenen bu matrisler, Micropascal yardımı ile hazırlanan program ile PIC mikroişlemcisinin okuyabileceği bir hale getirilmiştir.

2007, ...60... sayfa

ANAHTAR KELİMELER : Performans Grafikleri, AS-532, Cougar, PIC ,
Micropascal.

ABSTRACT

Master of Science Thesis

CALCULATING THE PERFORMANCE CHARTS BY USING ELECTRONIC PLATFORM

Fevzi YAZGAN
1st Lieutenant

Air Force Academy
Aeronautics and Space Technologies Institutes

Supervisor: Assistant Prof. Dr. Col. Serdar KARGIN

In this study, AS-532 Cougar Helicopter Performance charts converted to the one dimension matrixes by using MATLAB. The matrixes, which is saved to the SD-CARD, becomes readable by PIC microprocessor by using Micro Pascal.

2007, ...60... pages

Key Words: Performance Charts, AS-532, Cougar, PIC , Micropascal.

ÖZET

Yüksek Lisans Tezi

AS-532 COUGAR HELİKOPTERİ PERFORMANS PLANLAMASININ ELEKTRONİK PLATFORM ÜZERİNDE HESAPLANMASI

Fevzi YAZGAN
Hv. Plt. Ütğm.

Hava Harp Okulu
Havacılık ve Uzay Teknolojileri Enstitüsü
Elektronik Anabilim Dalı

Danışman : Hv.Yrd.Doç.Dr.Müh.Alb. Serdar KARGIN

Bu çalışmada, AS-532 Cougar helikopterinin performans grafiklerinin MATLAB yazılımı kullanılarak tek boyutlu matrisler halinde hazırlanmıştır. SD-CARD 'a yüklenen bu matrisler, Micropascal yardımı ile hazırlanan program ile PIC mikroişlemcisinin okuyabileceği bir hale getirilmiştir.

2007, ...120... sayfa

ANAHTAR KELİMELER : Performans Grafikleri, AS-532, Cougar, PIC ,
Micropascal.

ABSTRACT

Master of Science Thesis

CALCULATING THE PERFORMANCE CHARTS BY USING ELECTRONIC PLATFORM

Fevzi YAZGAN
1st Lieutenant

Air Force Academy
Aeronautics and Space Technologies Institutes

Supervisor: Assistant Prof. Dr. Col. Serdar KARGIN

In this study, AS-532 Cougar Helicopter Performance charts converted to the one dimension matrixes by using MATLAB. The matrixes, which is saved to the SD-CARD, becomes readable by PIC microprocessor by using Micro Pascal.

2007, ...120... pages

Key Words: Performance Charts, AS-532, Cougar, PIC , Micropascal.

ÖNSÖZ VE TEŞEKKÜR

Bu çalışmanın amacı; tüm uçan makinaların aerodinamik, denge merkezi, motor performansı, yakıt kullanımı gibi verilerinin hesaplanmasında insan hatalarını yok etmek, aynı zamanda bu verilerin çok kısa sürede elde edilebilmesini sağlamaktır. Bu çalışmada AS-532 Cougar helikopterinin performans grafikleri ele alınmıştır. Uçuşun kritik anlarında (hover, iniş-kalkış ve emercensi durumlar) hayati önemi olan ve uçuş süresi boyunca sürekli değişen faktörlere (ağırlık, irtifa, sıcaklık, rüzgara) bağlı olan bu verilerin, kısa sürede ve havada hesaplanması gerekmektedir. Bu hesaplamada havada ancak kullanıcı dostu, taşınabilir ve hızlı bir cihaz ile yapılabilir. Bu çalışmada gerçekleştirilen elektronik devre AS-532 helikopteri için bu ihtiyacı karşılamaktadır. Aynı zamanda bu elektronik devre çok kısa süreli bir çalışma ile gerek Hava Kuvvetleri, gerekse sivil sektörde uçan herhangi bir uçan makinaya uyarlanabilir. Yaptığım bu çalışmanın gelecekte performanstan kaynaklı tüm kazaları önlemesi dileğiyle tez danışmanım Hv.Yrd.Doç.Dr.Müh.Alb. Serdar KARGIN'a desteğinden ötürü teşekkürü borç bilirim.

Fevzi YAZGAN
İstanbul, Temmuz 2009

İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
ÖNSÖZ VE TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ	v
ÇİZELGELER DİZİNİ.....	vi
1. GİRİŞ	1
2. PERFORMANS	6
2.1. WIND ENVELOPE.....	6
2.2. A NOKTASI.....	7
2.3. ÇİFT MOTOR MAKSİMUM KALKIŞ AĞIRLIĞI.....	8
2.4. ÇİFT MOTOR İGE HOVER AĞIRLIĞI.....	9
2.5. ÇİFT MOTOR OGE HOVER AĞIRLIĞI.....	10
2.6. 45 K İLE ÇİFT MOTOR TIRMANIŞ ORANI.....	11
2.7. VY SÜRATİ İLE ÇİFT MOTOR TIRMANIŞ ORANI.....	12
2.8. OTOROTASYONDA MINIMUM ROTOR SÜRATİ.....	13
2.9. FLY AWAY.....	14
2.10. TEK MOTOR İGE HOVER AĞIRLIĞI.....	15
2.11. TEK MOTOR OGE HOVER AĞIRLIĞI.....	16
2.12. 45K İLE TEK MOTOR TIRMANIŞ ORANI.....	17
2.13. VY SÜRATİ İLE TEK MOTOR TIRMANIŞ ORANI	18
3. IMAGE PROSESİNG	19
3.1. Grafiğin Hazırlanması	19
3.2. MATLAB.....	20
3.2.1 Matris Haline Getirilmesi.....	20
3.2.2 Matrisden A Noktasının Hesaplanması.....	20
4. ELEKTRONİK	22
4.1. PIC18F452 DATA SHEET.....	22
4.2. LM317T.....	29
4.3. SD-CARD.....	31
4.4. PCB1602BA LCD.....	34
4.5. Devrenin Çalışma Prensibi.....	39
5. MİKROPASCAL	43
5.1. Keypad Kullanımı.....	43
5.2. SD-CARD'dan Bilgi Okuma.....	45
5.3. ANA PROGRAM.....	47
6. SONUÇ	58
KAYNAKLAR	59
Özgeçmiş	60

ŞEKİLLER DİZİNİ

Şekil No	Şekil	Sayfa
Şekil 1.1	Proteus Similasyonu.....	3
Şekil 1.2	Devrenin Breadbord Üzerinde Gerçeklenmesi	4
Şekil 1.3	Isiste Çizilen Devre Şeması	4
Şekil 1.4	Bakır Plakete Basılmış Devre	5
Şekil 2.1	Hover Wind Envelope	6
Şekil 2.2	A Noktası	7
Şekil 2.3	ÇİFT MOTOR MAKSİMUM KALKIŞ AĞIRLIĞI	8
Şekil 2.4	ÇİFT MOTOR İGE HOVER AĞIRLIĞI	9
Şekil 2.5	ÇİFT MOTOR OGE HOVER AĞIRLIĞI	10
Şekil 2.6	45 K İLE ÇİFT MOTOR TIRMANIŞ ORANI	11
Şekil 2.7	VY SÜRATİ İLE ÇİFT MOTOR TIRMANIŞ ORANI	12
Şekil 2.8	OTOROTASYONDA MINIMUM ROTOR SÜRATİ	13
Şekil 2.9	FLY AWAY	14
Şekil 2.10	TEK MOTOR İGE HOVER AĞIRLIĞI.....	15
Şekil 2.11	TEK MOTOR OGE HOVER AĞIRLIĞI	16
Şekil 2.12	45K İLE TEK MOTOR TIRMANIŞ ORANI.....	17
Şekil 2.13	VY SÜRATİ İLE TEK MOTOR TIRMANIŞ ORANI.....	18
Şekil 3.1	Scanner İle Taranmış Resim.....	19
Şekil 3.2	Temizlenmiş Resim.....	19
Şekil 4.1	Pin Diyagramı.....	22
Şekil 4.2	Kristal Bağlantısı	27
Şekil 4.3	LM317 Devre Şeması	29
Şekil 4.4	Devrenin 3D Görüntüsü Ve LM317 Yerleri.....	29
Şekil 4.5	LM317 Proteus Devresi.....	30
Şekil 4.6	LM317 Önden Bacakları.....	30
Şekil 4.7	SD-CARD Çeşitleri.....	31
Şekil 4.8	SD-CARD Alttan Görünüşü Ve Çıkışları.....	32
Şekil 4.9	SD-CARD Microişlemci Arasındaki Bağlantı.....	33
Şekil 4.10	SD-CARD Mimarisi.....	33
Şekil 4.11	SD-CARD Adaptörü.....	33
Şekil 4.12	LCD Yapısı.....	35
Şekil 4.13	LCD Datasheet.....	36
Şekil 4.14	LCD Karakterleri.....	38
Şekil 4.15	Devrenin Çalışır Haldeki Görüntüsü.....	39
Şekil 4.16	Devrenin Üstten Görünüşü.....	40
Şekil 4.17	Devrenin Alttan Görünüşü.....	41
Şekil 5.1	Keypad Kullanımı Similasyonu.....	43
Şekil 5.2	SD-CARD'dan Bilgi Okuma Similasyonu.....	44

ÇİZELGELER DİZİNİ

Şekil No	Şekil	Sayfa
Çizelge 1.1	Performans Planı.....	2
Çizelge 4.1	Mikroişlemci Özellikleri	22
Çizelge 4.2	Mikroişlemci Çıkışları	23
Çizelge 4.3	Kondansatör Seçimi.....	27
Çizelge 4.4	SD-CARD Bağlantı Tablosu.....	32
Çizelge 4.5	LCD Bacakları	37
Çizelge 4.6	Devrede Kullanılan Elemanlar.....	40

1. GİRİŞ

1.1 Performans

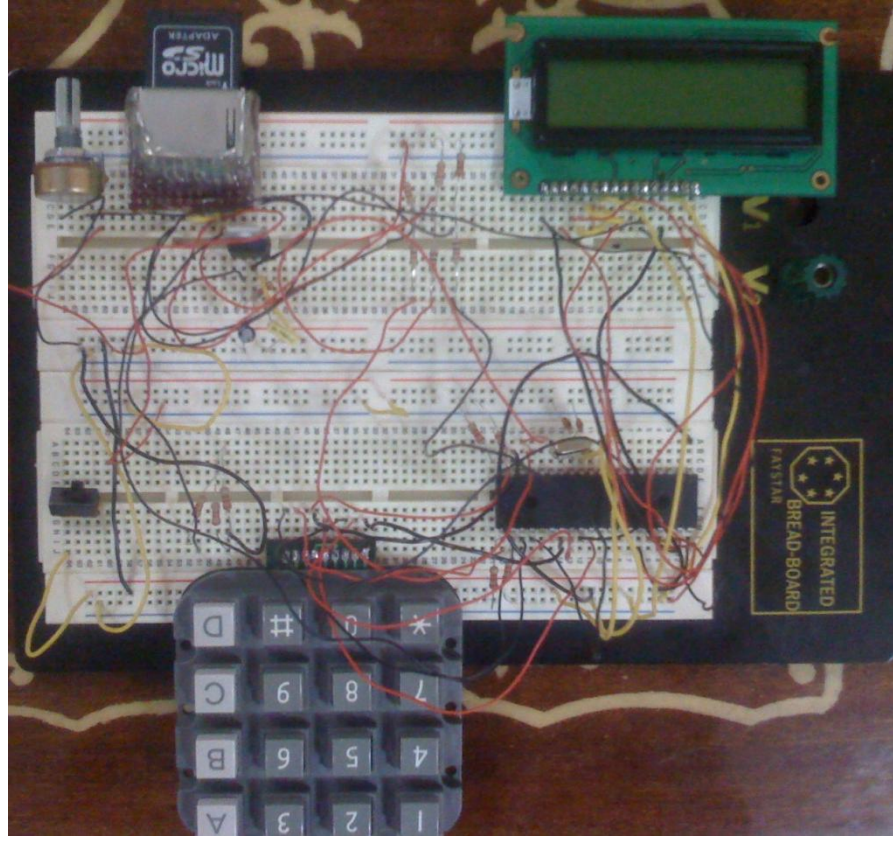
Tüm uçan makinalar doğa güçlerine karşı koyarak uçarlar. Bu güçlerde bazı faktörlere bağlıdır. Bu tezde kullanılan AS-532 helikopteri ile uçmadan önce bazı bilgilerin grafiklerden hesaplanması gerekir. Bu bilgiler bize helikopterin ağırlığına, o anki dış hava sıcaklığına, basınç irtifasına, rüzgar şiddetine bağlı olarak helikopterin performansını verir. Bu performans verileri helikopterin ne kadar ağırlık kaldırabileceği, hangi tımanış oranı ile tırmanabileceği gibi bazı verilerdir. Bu veriler 2 bölümde açıklanacaktır.

Performans verilerinin önemini anlamak için bir senaryo dahilinde yaşanabilecek bir uçuşu gözden geçirelim.

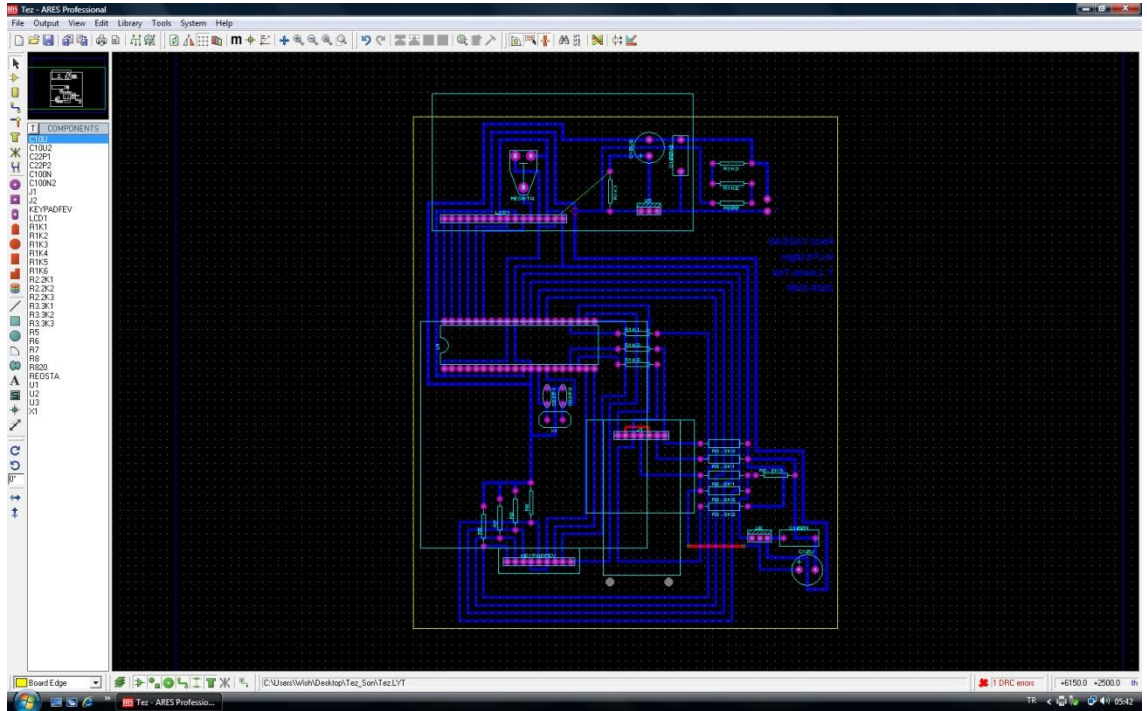
Saat 12:00'da Konya meydanından kalkan helikopter. 60 deniz mili (NM) batıda bulunan 6000 feetlik bir tepe üzerinden bir kazazede olacaktır fakat ne bir tepeye ineceği nede tepenin irtifası henüz belli değildir. Konya meydanında dış hava harareti (OAT) 20° C, irtifa 3381 feet, rüzgar ise 10K'tır. Helikopter ağırlığı ise 18800 libredir (lb). Helikopterin kazazede bölgesine gitmesi 120K ile 30dk sürer. Kazazede bulmasında 30dk sürer. Geçen zaman içinde helikopter (1 saatte) 150galon(990lb) yakıt harcamıştır. Kazazede bölgesinde rüzgar 20K ile esmektedir. İrtifası ise 6000 feettir. Sıcaklık ise 30° C tır. Aşağıda Konya meydanından kalkışındaki ve kazazede bölgesindeki performans verileri gösterilmiştir (Çizelge 1.1).

Çizelge 1.1 Performans Planı

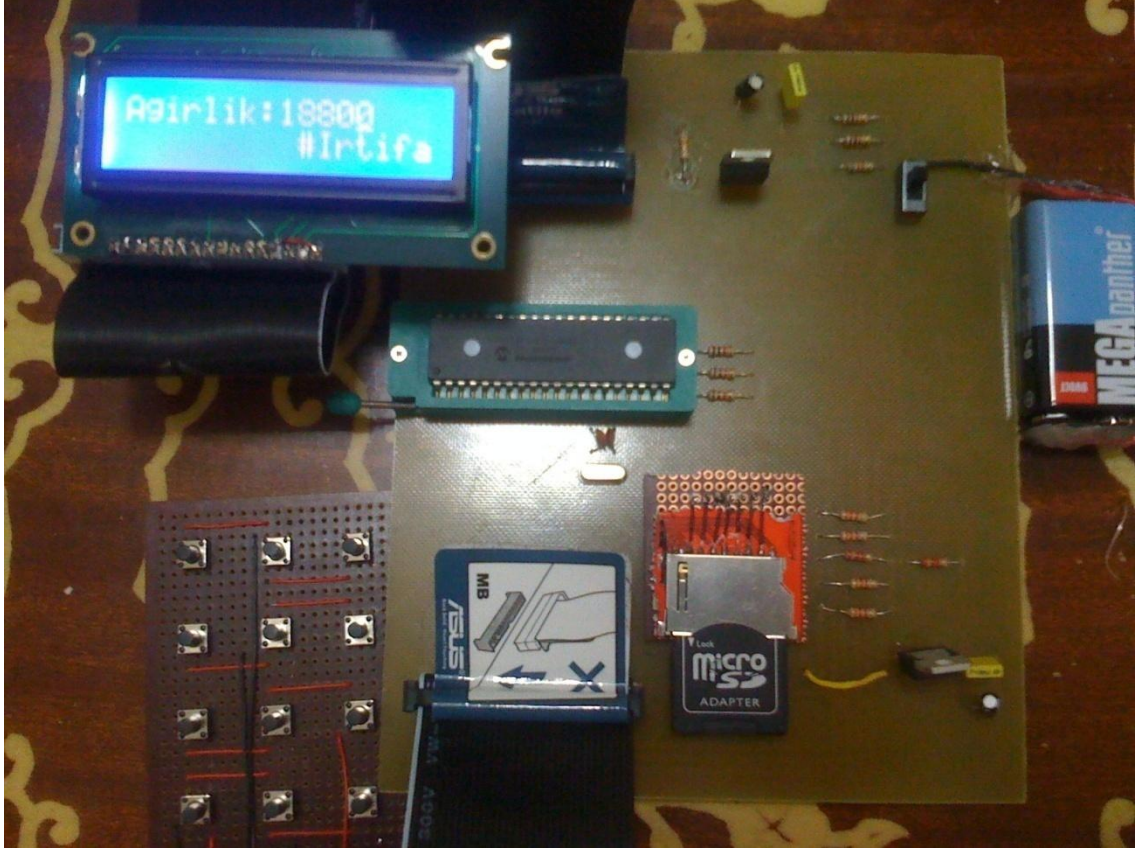
	KONYA	KAZAZEDE BÖLGESİ
AĞIRLIK	18800lb	17810lb
OAT	20C	30C
İRTİFA	3381Feet	6000Feet
RÜZGAR	10K	20K
A NOKTASI	322	391
FLY AWAY	166	124
MINIMUM ROTOR SPEED IN AUTOROTATION	297	310
TWIN-ENGINE MAX PERMISSIBLE TAKEOFF WEIGHT	19840	19049
TWIN-ENGINE HOVER IGE PERFORMANCE	19840	18610
TWIN-ENGINE HOVER OGE PERFORMANCE	19840	17536
TWIN-ENGINE RATE OF CLIMB AT 45KT IAS	1825	1270
TWIN-ENGINE RATE OF CLIMB AT VY	1362	1153
ONE-ENGINE HOVER OGE PERFORMANCE	13166	11359
ONE-ENGINE HOVER IGE PERFORMANCE	14065	12204
ONE-ENGINE RATE OF CLIMB AT 45KT IAS	-4	-256
ONE-ENGINE RATE OF CLIMB AT VY	88	-249



Şekil 1.2 Devrenin Breadbord Üzerinde Gerçeklenmesi



Şekil 1.3 İstite Çizilen Devre Şeması

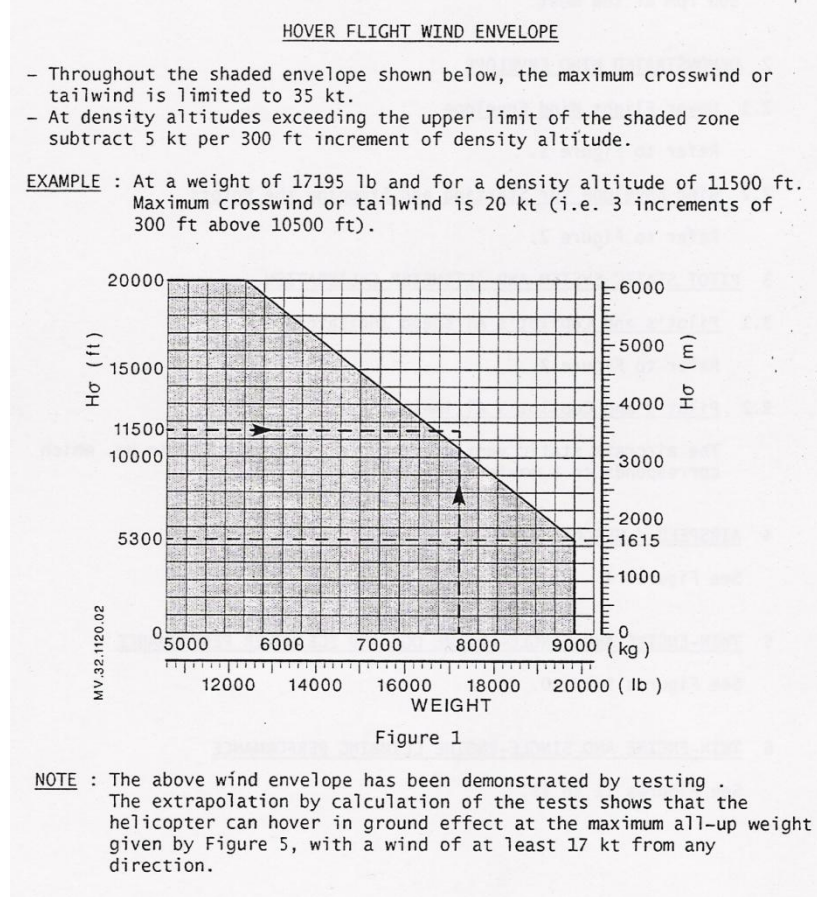


Şekil 1.4 Bakır Plakete Basılmış Devre

2. PERFORMANS

2.1 WIND ENVELOPE

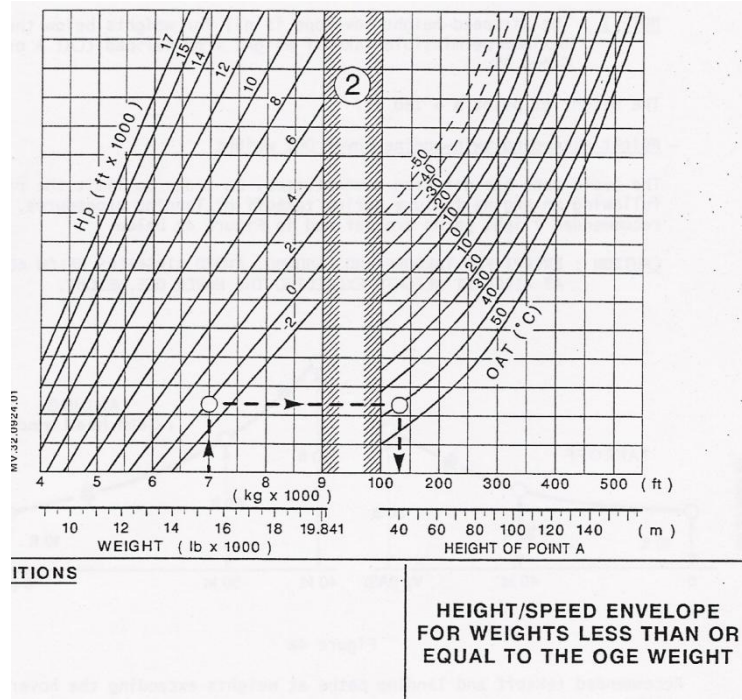
Rüzgar zarfı helikopterin hover pozisyonundaki maksimum arka ve yan rüzgar limitini gösterir. Bu değer aşağıdaki grafikten basınç irtifası ve ağırlığa göre hesaplanır.



Şekil 2.1 Hover Wind Envelope

2.2 A NOKTASI

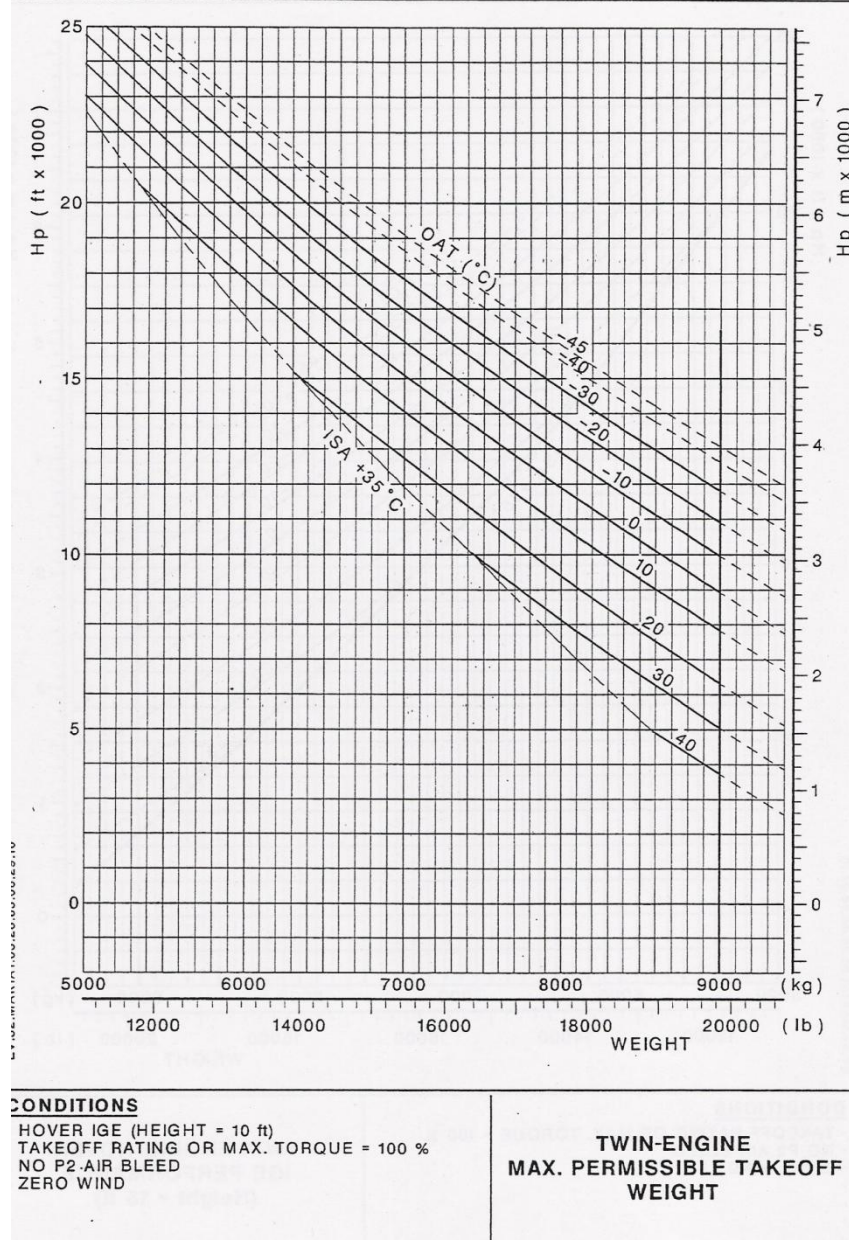
A noktası irtifası, helikopter hover pozisyonunda iken tek motor arızası olduğunda helikopterin bulunduğu pozisyondan uçuşa devam edebilmesi için gerekli olan minimum irtifadır. Bu değer ağırlık, basınç irtifası, dış hava hararetine bağlı olarak aşağıdaki grafikten hesaplanır.



Şekil 2.2 A Noktası

2.3 ÇİFT MOTOR MAKSİMUM KALKIŞ AĞIRLIĞI

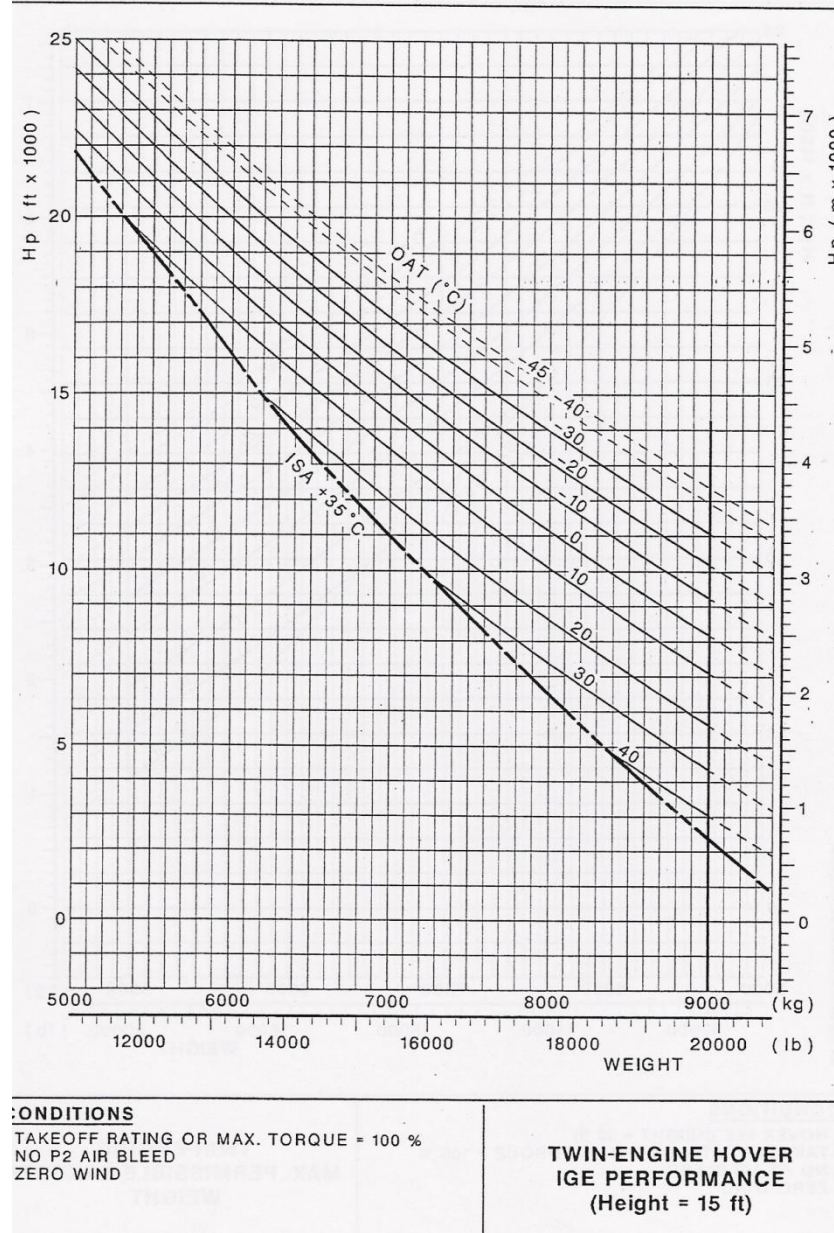
Çift motor maksimum kalkış ağırlığı helikopterin ilgili sıcaklık ve basınç irtifasındaki maksimum kalkış ağırlığıdır. Bu değer aşağıdaki grafikten basınç irtifası ve sıcaklığa göre hesaplanır.



Şekil 2.3 ÇİFT MOTOR MAKSİMUM KALKIŞ AĞIRLIĞI

2.4 ÇİFT MOTOR IGE HOVER AĞIRLIĞI

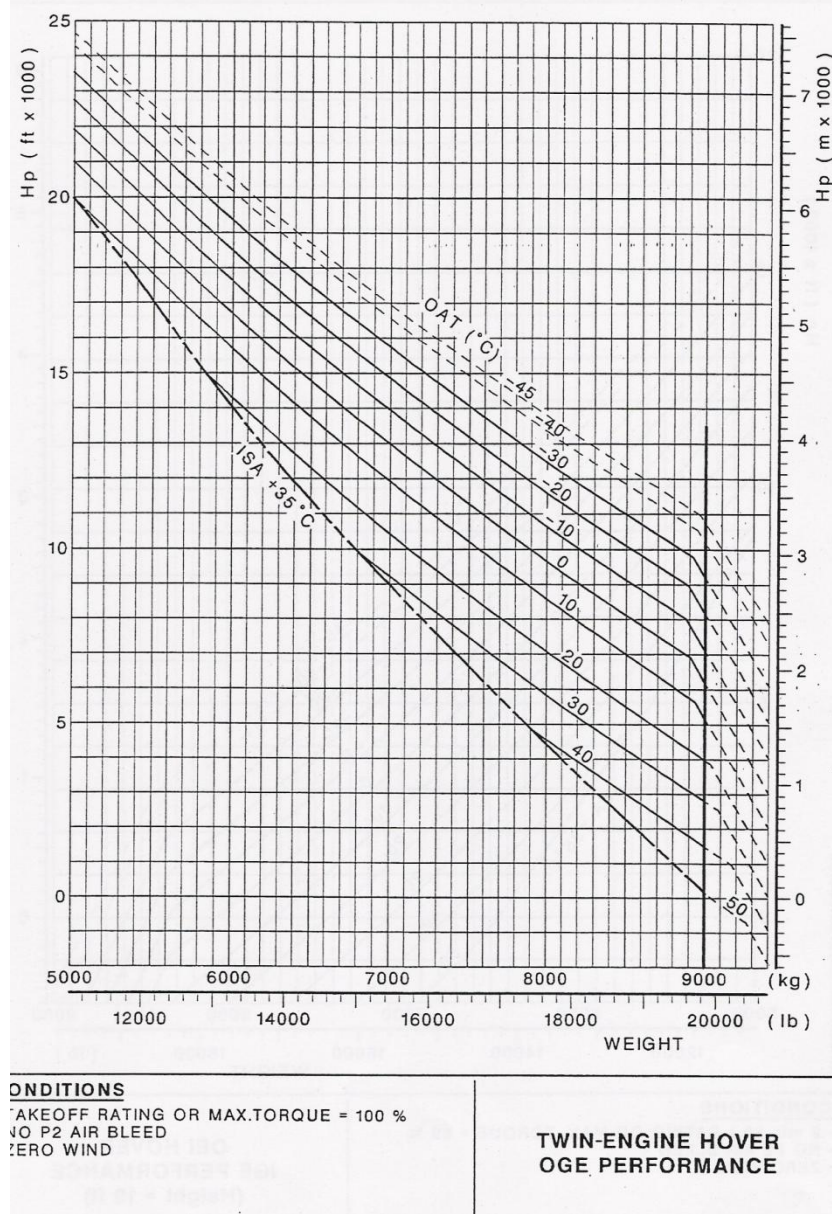
Çift motor IGE hover ağırlığı helikopterin ilgili sıcaklık ve basınç irtifasındaki yer etkisi içinde hover yapabilmesi için olabilecek maksimum ağırlıktır. Bu değer aşağıdaki grafikten basınç irtifası ve sıcaklığa göre hesaplanır.



Şekil 2.4 ÇİFT MOTOR IGE HOVER AĞIRLIĞI

2.5 ÇİFT MOTOR OGE HOVER AĞIRLIĞI

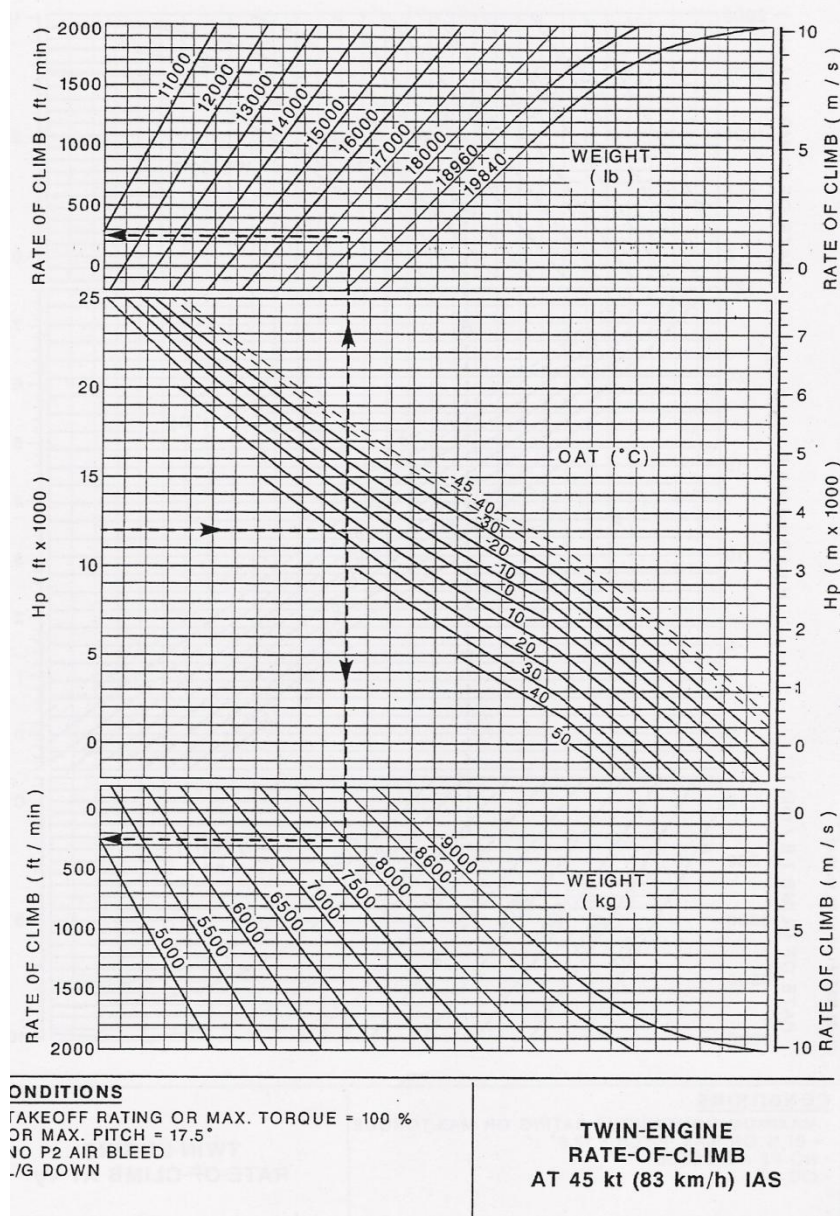
Çift motor OGE hover ağırlığı helikopterin ilgili sıcaklık ve basınç irtifasındaki yer etkisi dışında hover yapabilmesi için olabilecek maksimum ağırlıktır. Bu değer aşağıdaki grafikten basınç irtifası ve sıcaklığa göre hesaplanır.



Şekil 2.5 ÇİFT MOTOR OGE HOVER AĞIRLIĞI

2.6 45 K İLE ÇİFT MOTOR TIRMANIŞ ORANI

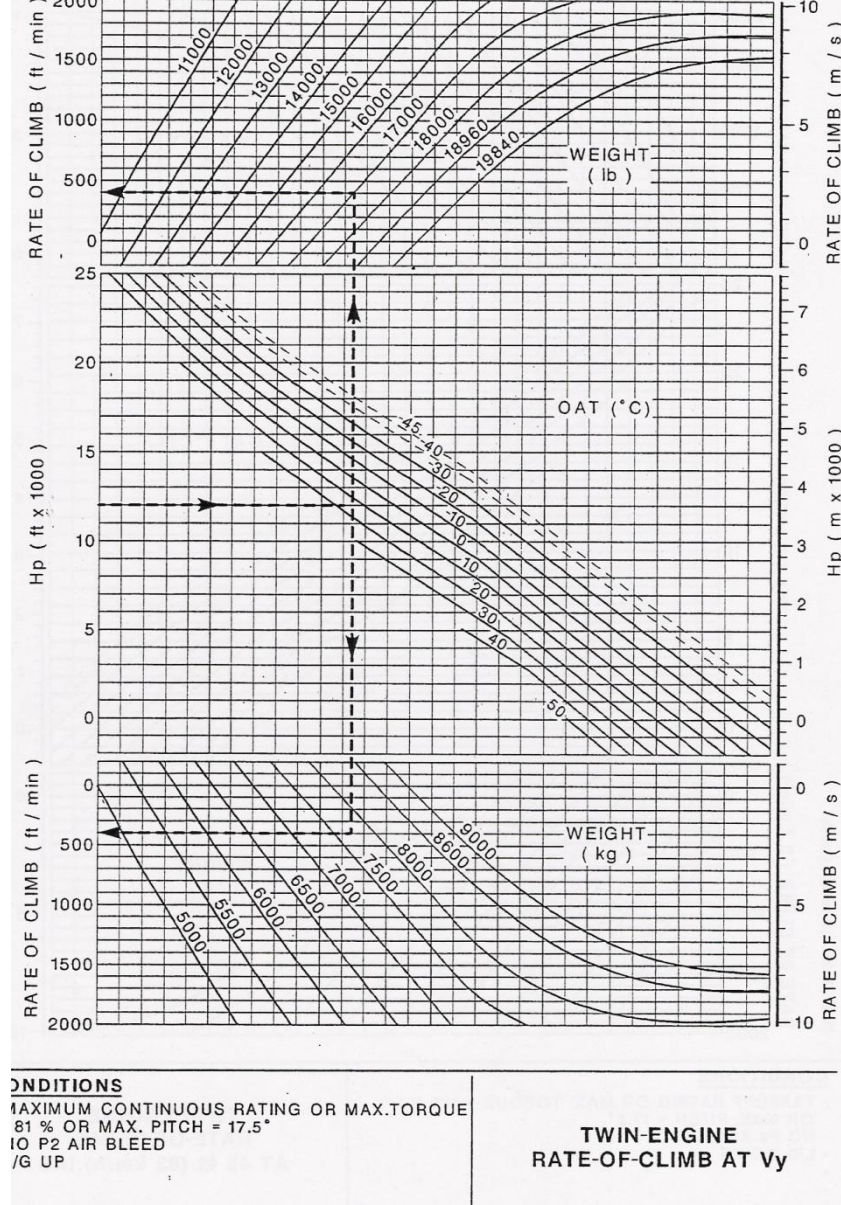
45 K ile çift motor tırmanış oranı helikopterin ilgili ağırlık, sıcaklık ve basınç irtifasında dakikada kaç feet irtifa alabileceğini gösterir. Bu değer aşağıdaki grafikten hesaplanır.



Şekil 2.6 45 K İLE ÇİFT MOTOR TIRMANIŞ ORANI

2.7 VY SÜRATI İLE ÇİFT MOTOR TIRMANIŞ ORANI

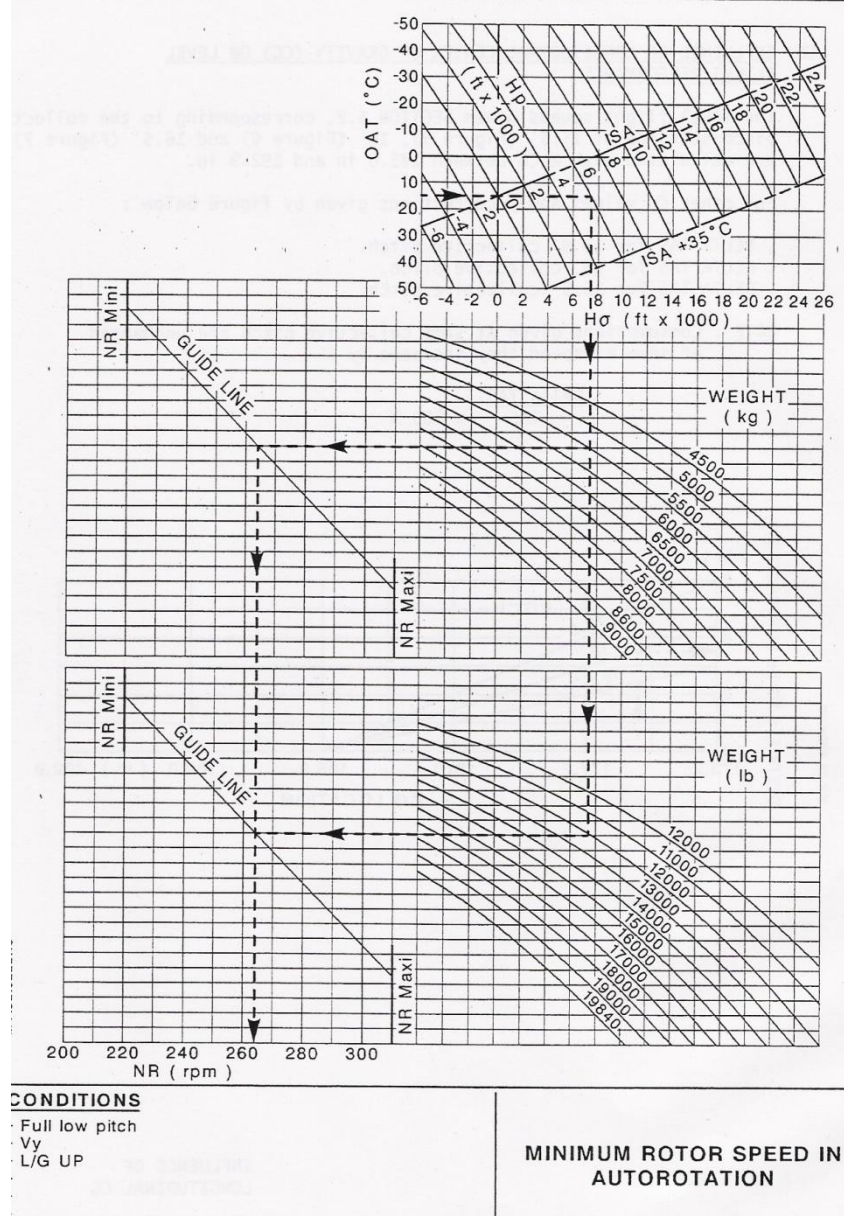
VY sürati ile çift motor tırmanış oranı helikopterin ilgili ağırlık, sıcaklık ve basınç irtifasında dakikada kaç feet irtifa alabileceğini gösterir. Bu değer aşağıdaki grafikten hesaplanır.



Şekil 2.7 VY SÜRATI İLE ÇİFT MOTOR TIRMANIŞ ORANI

2.8 OTOROTASYONDA MINIMUM ROTOR SÜRATI

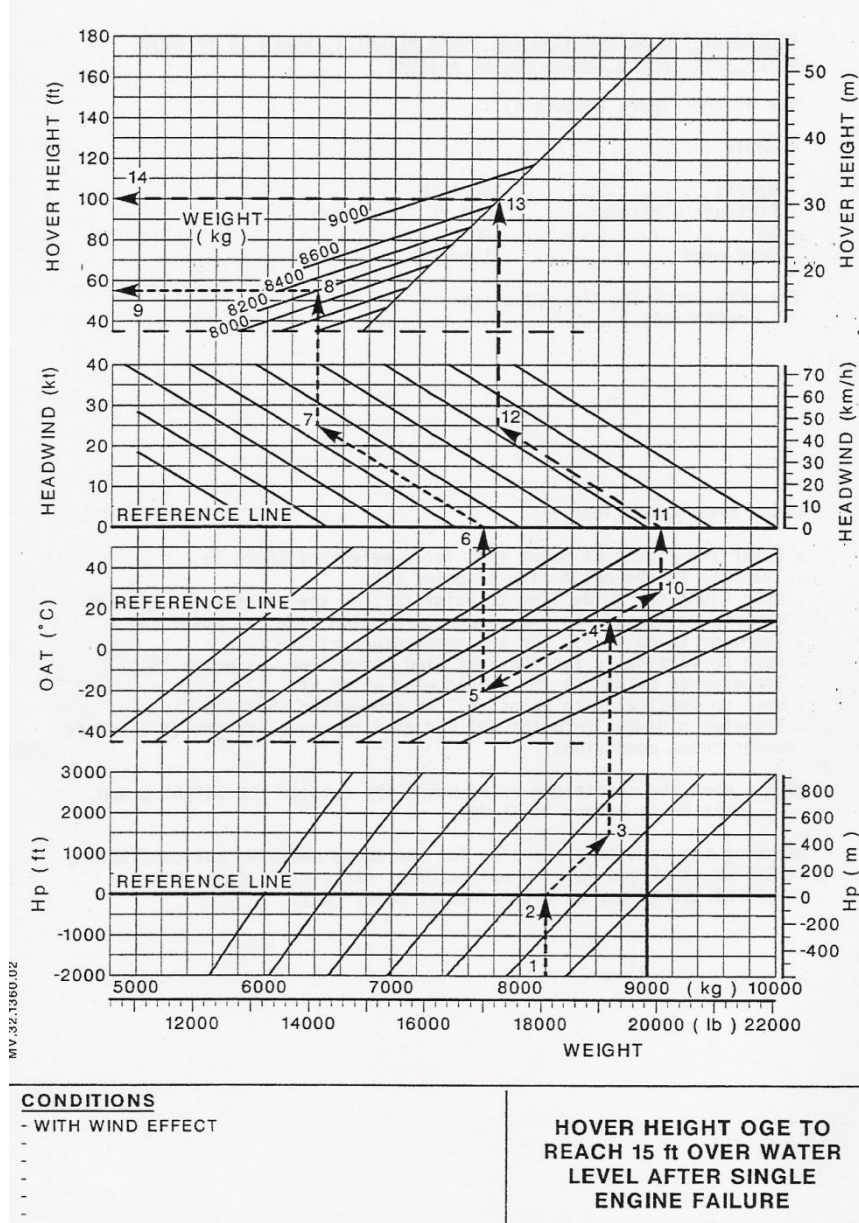
Otorotasyonda minimum rotor sürati her iki motor kaybında yere güvenle inebilmek için pallerin minimum hangi süratte dönmesi gerektiği aşağıdaki grafikten hesaplanır.



Şekil 2.8 OTOROTASYONDA MINIMUM ROTOR SÜRATI

2.9 FLY AWAY

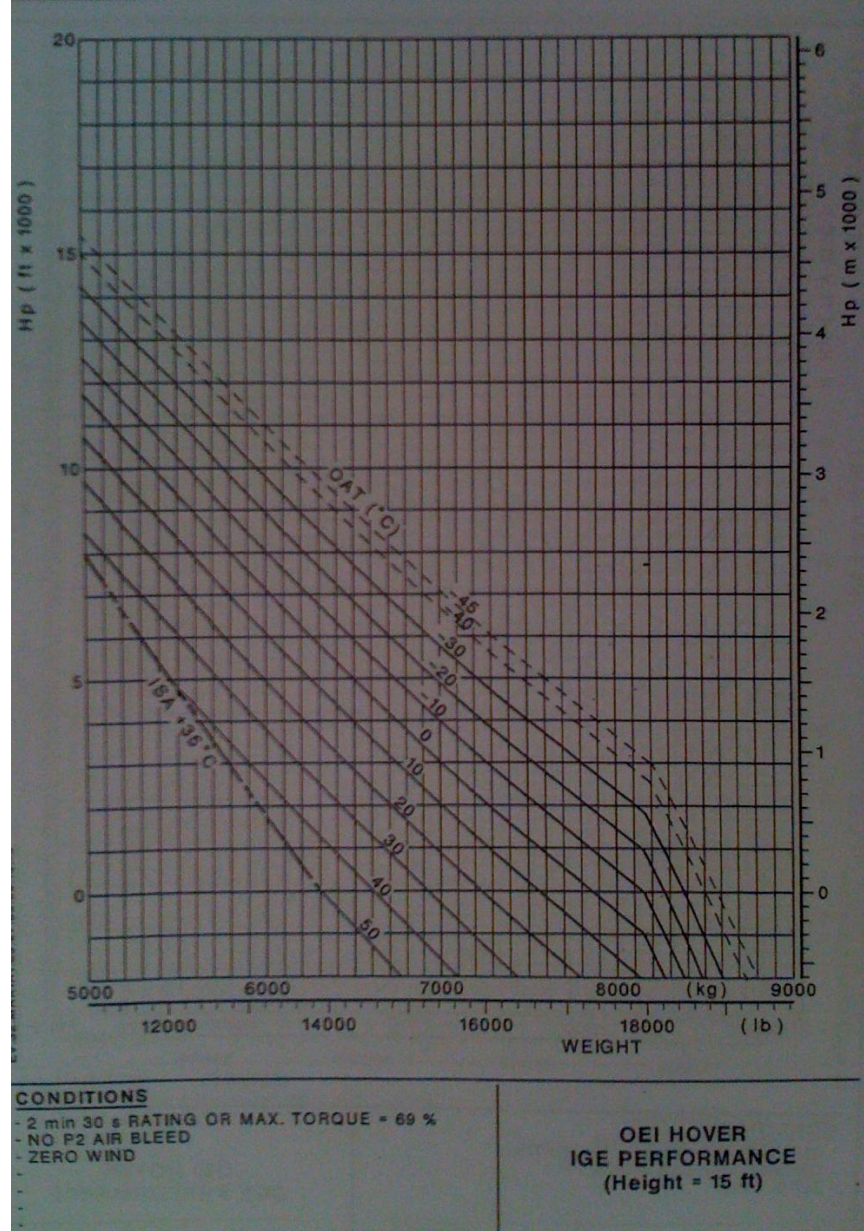
Su üzerinde tek motor arızası meydana geldiğinde pas geçerken 15 ft ulaşabilmek için gerekli olan irtifdadır. Aşağıdaki grafikten hesaplanır.



Şekil 2.9 FLY AWAY

2.10 TEK MOTOR IGE HOVER AĞIRLIĞI

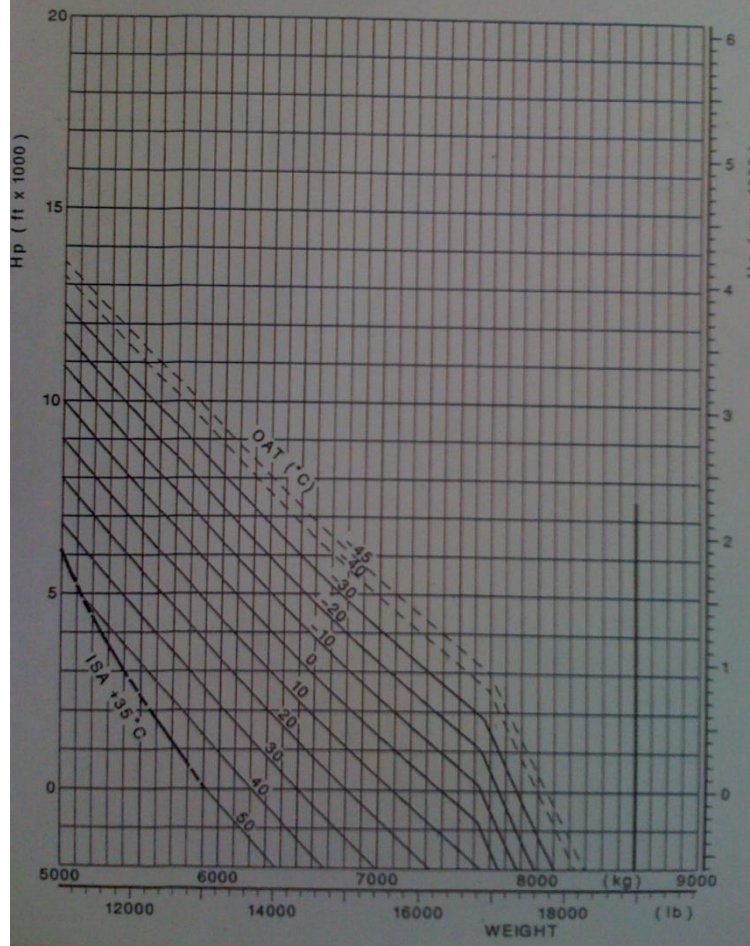
Tek motor IGE hover ağırlığı helikopterin ilgili sıcaklık ve basınç irtifasındaki yer etkisi içinde hover yapabilmesi için olabilecek maksimum ağırlıktır. Bu değer aşağıdaki grafikten basınç irtifası ve sıcaklığa göre hesaplanır.



Şekil 2.10 TEK MOTOR IGE HOVER AĞIRLIĞI

2.11 TEK MOTOR OGE HOVER AĞIRLIĞI

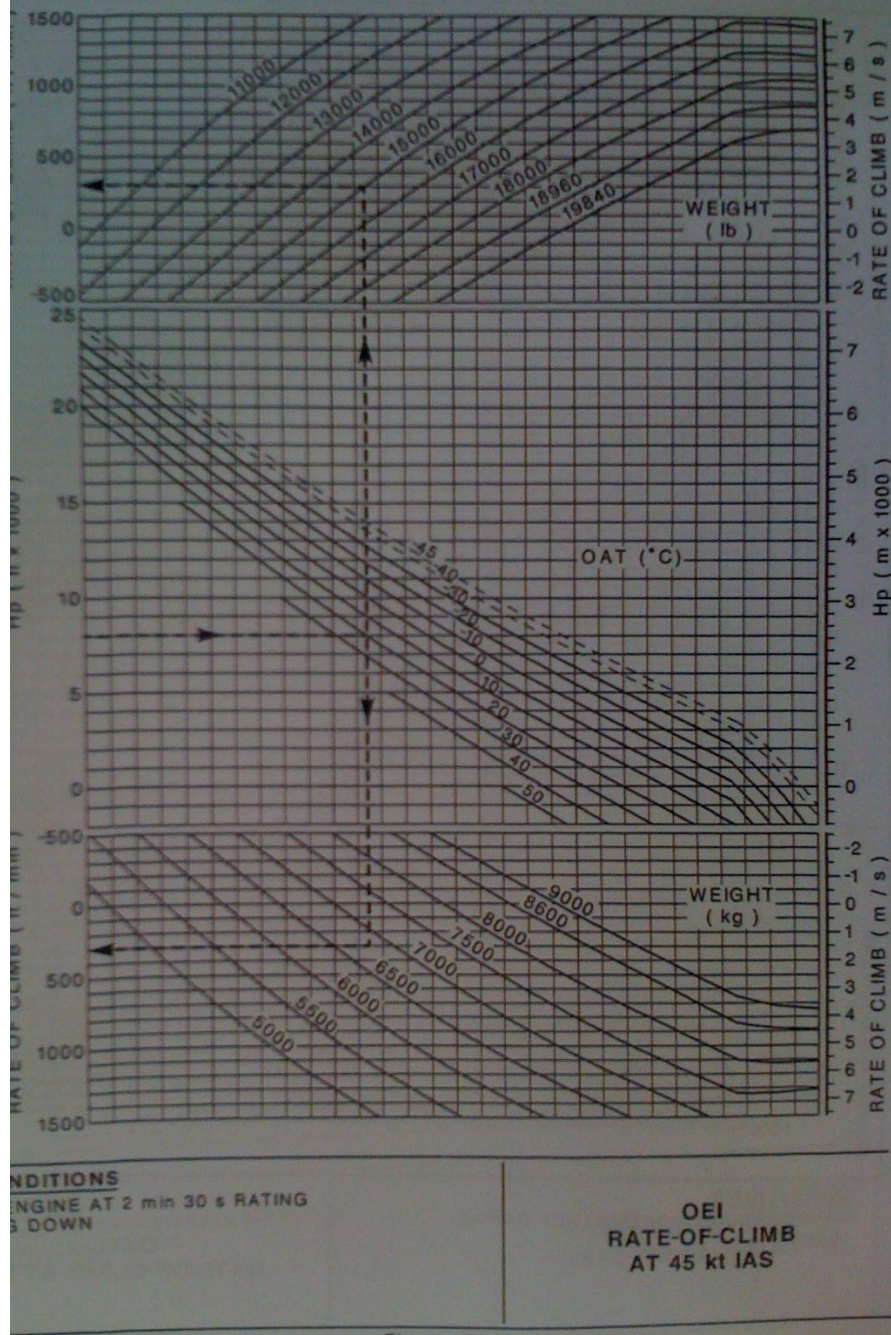
Tek motor OGE hover ağırlığı helikopterin ilgili sıcaklık ve basınç irtifasındaki yer etkisi dışında hover yapabilmesi için olabilecek maksimum ağırlıktır. Bu değer aşağıdaki grafikten basınç irtifası ve sıcaklığa göre hesaplanır.



Şekil 2.11 TEK MOTOR OGE HOVER AĞIRLIĞI

2.12 45K İLE TEK MOTOR TIRMANIŞ ORANI

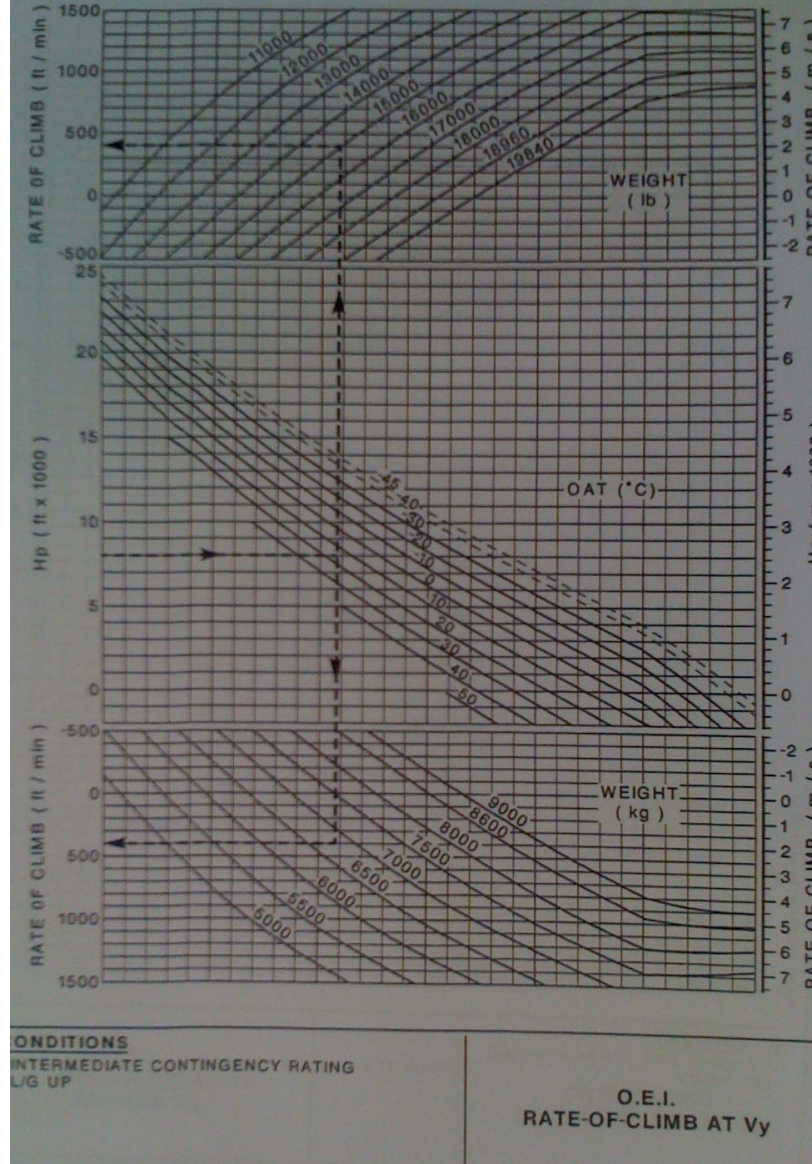
45K ile tek motor tırmanış oranı helikopterin ilgili ağırlık, sıcaklık ve basınç irtifasında dakikada kaç feet irtifa alabileceğini gösterir. Bu değer aşağıdaki grafikten hesaplanır.



Şekil 2.12 45K İLE TEK MOTOR TIRMANIŞ ORANI

2.13 VY SÜRATİ İLE TEK MOTOR TIRMANIŞ ORANI

VY sürati ile tek motor tırmanış oranı helikopterin ilgili ağırlık, sıcaklık ve basınç irtifasında dakikada kaç feet irtifa alabileceğini gösterir. Bu değer aşağıdaki grafikten hesaplanır.



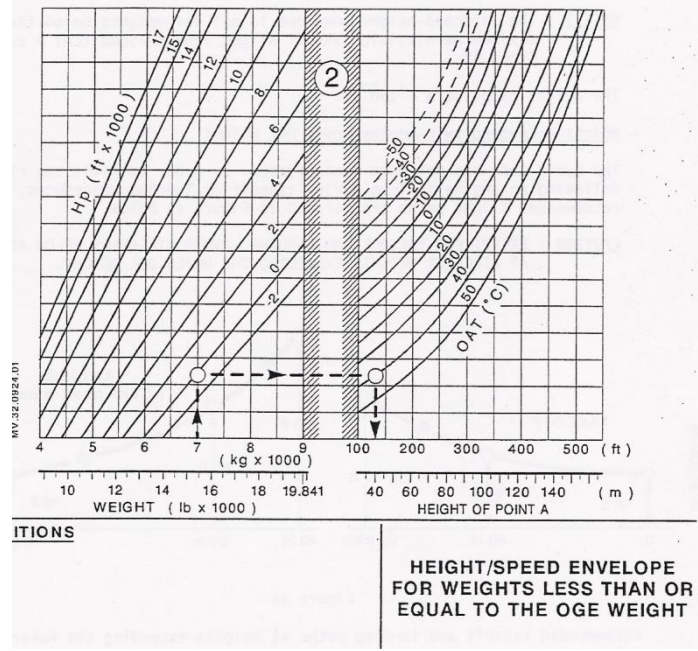
Şekil 2.13 VY SÜRATİ İLE TEK MOTOR TIRMANIŞ ORANI

3. IMAGE PROSSESING

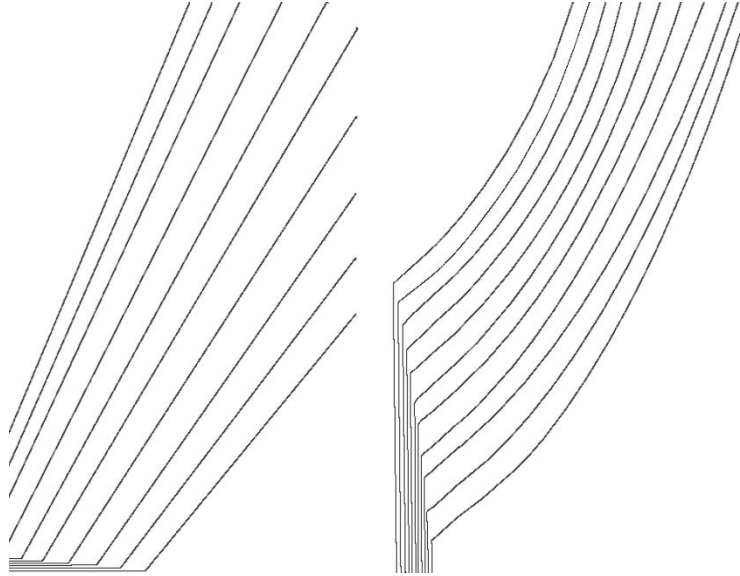
Bu bölümde örnek olarak A Noktasının nasıl hesaplandığı ele alınacaktır.

3.1 Grafiğin Hazırlanması

Scanner ile taranmış resim (Şekil 3-1) , Photoshop vasıtası ile temizlenir ve üzerinde işlem yapılabilecek hale getirir.



Şekil 3.1 Scanner İle Taranmış Resim



Şekil 3.2 Temizlenmiş Resim

3.2 MATLAB

3.2.1 Matris Haline Getirilmesi

Matlab' da yazılan aşağıdaki program vasıtası ile resim tek boyutlu bir matris haline getirilir.

```
clear all
c=80;
resim='AAA.bmp';
I = imread(resim);
r=imshow(resim);
q = imattributes(r);
for x = 1:str2num(cell2mat(q(2,2)))
for y = 1:str2num(cell2mat(q(1,2)))
if ((I(x,y,1)<c) & (I(x,y,2)<c)) | ((I(x,y,1)<c) & (I(x,y,3)<c)) |
((I(x,y,2)<c) & (I(x,y,3)<c)))
aa(x,y)=0;
else
aa(x,y)=1;
end;
end;
end;
```

3.2.2 Matrisden A Noktasının Hesaplanması

Tek boyutlu matrisden for döngüsü ve if komutları ile gerçekte grafikten bir insan nasıl A Noktasını hesaplıyor ise aynı şekilde hesaplatılır.

Aşağıdaki ana program irtifa, ağırlık, ve sıcaklıklara belli aralıklarla değer vererek tüm olasılıkların tek tek hesaplanmasını ve sonuc matrisine aktarılmasını sağlar.

```
%Ana Program
i=0;
for f=0:100:16700
for w=11000:50:19800
for s=-40:40
ss=50+s;
i=i+1;
sonuc(i) = ahesap(aa,f,ss,w);
end
end
end
```

Aşağıdaki fonksiyonlar sayesinde ana program hesaplamayı yapar.

```
function ap = ahesap(aa,f,ss,w)
xa=450/(19840-9000);
x=fix(xa*(w-9000))+1;
y1 = abul(aa,f,x);
```

```

x= abul2(aa,ss,y1);
ap=fix((x-548)*(450/(964-555)))+100;
if ap<100
    ap=100;
end

function y1= abul(a,f,x)
pp=0;
carpma=fix(f/2000)+1;
yy1=0;
yy2=0;
for y=737:-1:2
    if (a(y-1,x)==1 & a(y,x)==0)
        pp=pp+1;
        if carpma==pp
            yy1=y;
        end
        if (carpma+1)==pp
            yy2=y;
        end
    end
end
yy=yy1-yy2;
y1=round(yy1-(yy*((f/2000)-fix(f/2000))))+1;

function x1 = tmtobul(a,s,y)
pp=0;
carpma=fix(s/10)+1;
xx1=0;
xx2=0;
for x=490:964
    if (a(y,x-1)==0 & a(y,x)==1)
        pp=pp+1;
        if carpma==pp
            xx1=x;
        end
        if (carpma+1)==pp
            xx2=x;
        end
    end
end
xx=xx2-xx1;
x1=fix(xx1+(xx*((s/10)-fix(s/10)))));

```

Sonuçları harddiske .txt uzantısı ile aşağıdaki program ile kaydedilir.

```

fid = fopen('fly.txt','w');
fprintf(fid,'%4.3d',f);
fclose(fid)

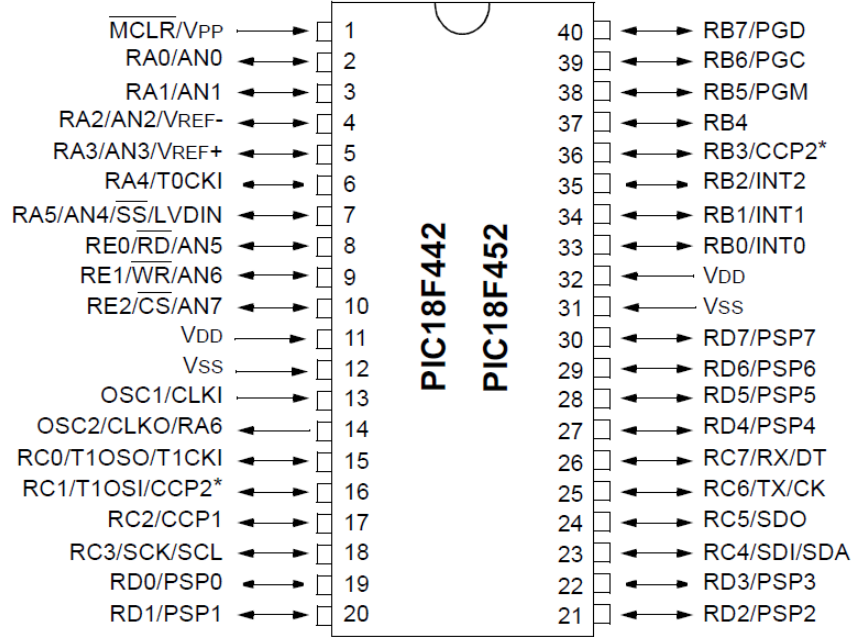
```

Diğer tüm hesaplamalar program algoritması değiştirilerek yapılmıştır. Tez ile birlikte verilen CD' de mevcuttur.

4. ELEKTRONİK

4.1 PIC18F452 DATA SHEET

DIP



Note: Pin compatible with 40-pin PIC16C7X devices.

Şekil 4.1 Pin Diyagramı

Çizelge 4.1 Mikroişlemci Özellikleri

Device	Flash (bytes)	Single Word Instructions	On-Chip RAM (bytes)	Data EEPROM (bytes)
PIC18F452	32K	16384	1536	256

Çizelge 4.2 Mikroişlemci Çıkışları

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	DIP	PLCC	TQFP			
MCLR/VPP MCLR VPP	1	2	18	I I	ST ST	Master Clear (input) or high voltage ICSP programming enable pin. Master Clear (Reset) input. This pin is an active low RESET to the device. High voltage ICSP programming enable pin.
NC	—			—	—	These pins should be left unconnected.
OSC1/CLKI OSC1 CLKI	13	14	30	I I	ST CMOS	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode, CMOS otherwise. External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKI, OSC2/CLKO pins.)
OSC2/CLKO/RA6 OSC2 CLKO RA6	14	15	31	O O I/O	— — TTL	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. General Purpose I/O pin.
RA0/AN0 RA0 AN0 RA1/AN1 RA1 AN1 RA2/AN2/VREF- RA2 AN2 VREF- RA3/AN3/VREF+ RA3 AN3 VREF+ RA4/T0CKI RA4 T0CKI RA5/AN4/SS/LVDIN RA5 AN4 SS LVDIN RA6	2 3 4 5 6 7	3 4 5 6 7 8	19 20 21 22 23 24	I/O I I/O I I/O I I I/O I I/O I I I	TTL Analog TTL Analog TTL Analog Analog TTL Analog Analog ST/OD ST TTL Analog ST Analog	PORTA is a bi-directional I/O port. Digital I/O. Analog input 0. Digital I/O. Analog input 1. Digital I/O. Analog input 2. A/D Reference Voltage (Low) input. Digital I/O. Analog input 3. A/D Reference Voltage (High) input. Digital I/O. Open drain when configured as output. Timer0 external clock input. Digital I/O. Analog input 4. SPI Slave Select input. Low Voltage Detect Input. (See the OSC2/CLKO/RA6 pin.)

Çizelge 4.2 (Devam)

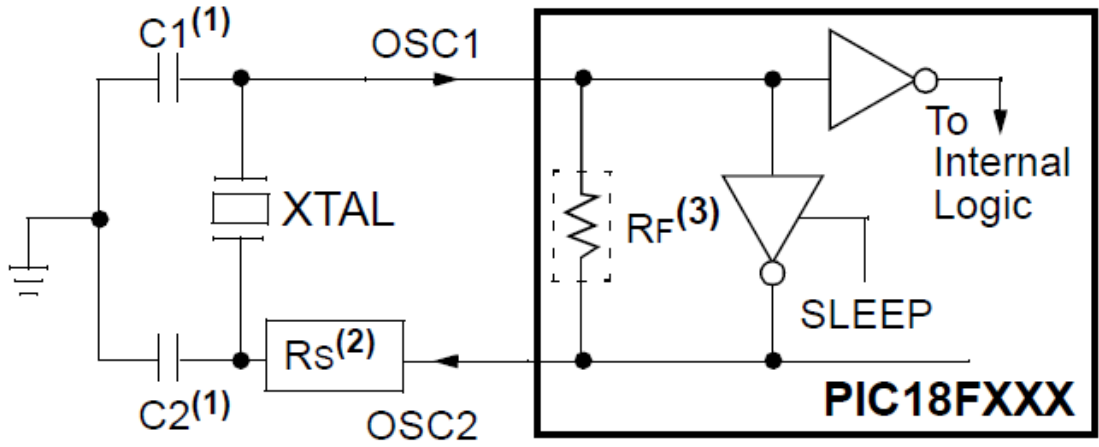
Pin Name	Pin Number			Pin Type	Buffer Type	Description
	DIP	PLCC	TQFP			
RB0/INT0 RB0 INT0	33	36	8	I/O I	TTL ST	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External Interrupt 0.
RB1/INT1 RB1 INT1	34	37	9	I/O I	TTL ST	External Interrupt 1.
RB2/INT2 RB2 INT2	35	38	10	I/O I	TTL ST	Digital I/O. External Interrupt 2.
RB3/CCP2 RB3 CCP2	36	39	11	I/O I/O	TTL ST	Digital I/O. Capture2 input, Compare2 output, PWM2 output.
RB4	37	41	14	I/O	TTL	Digital I/O. Interrupt-on-change pin.
RB5/PGM RB5 PGM	38	42	15	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. Low Voltage ICSP programming enable pin.
RB6/PGC RB6 PGC	39	43	16	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming clock pin.
RB7/PGD RB7 PGD	40	44	17	I/O I/O	TTL ST	Digital I/O. Interrupt-on-change pin. In-Circuit Debugger and ICSP programming data pin.

Çizelge 4.2 (Devam)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	DIP	PLCC	TQFP			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	15	16	32	I/O O I	ST — ST	PORTC is a bi-directional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2 RC1 T1OSI CCP2	16	18	35	I/O I I/O	ST CMOS ST	Digital I/O. Timer1 oscillator input. Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1 RC2 CCP1	17	19	36	I/O I/O	ST ST	Digital I/O. Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	18	20	37	I/O I/O I/O	ST ST ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	23	25	42	I/O I I/O	ST ST ST	Digital I/O. SPI Data In. I ² C Data I/O.
RC5/SDO RC5 SDO	24	26	43	I/O O	ST —	Digital I/O. SPI Data Out.
RC6/TX/CK RC6 TX CK	25	27	44	I/O O I/O	ST — ST	Digital I/O. USART Asynchronous Transmit. USART Synchronous Clock (see related RX/DT).
RC7/RX/DT RC7 RX DT	26	29	1	I/O I I/O	ST ST ST	Digital I/O. USART Asynchronous Receive. USART Synchronous Data (see related TX/CK).

Çizelge 4.2 (Devam)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	DIP	PLCC	TQFP			
RD0/PSP0	19	21	38	I/O	ST TTL	PORTD is a bi-directional I/O port, or a Parallel Slave Port (PSP) for interfacing to a microprocessor port. These pins have TTL input buffers when PSP module is enabled. Digital I/O. Parallel Slave Port Data.
RD1/PSP1	20	22	39	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RD2/PSP2	21	23	40	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RD3/PSP3	22	24	41	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RD4/PSP4	27	30	2	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RD5/PSP5	28	31	3	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RD6/PSP6	29	32	4	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RD7/PSP7	30	33	5	I/O	ST TTL	Digital I/O. Parallel Slave Port Data.
RE0/ $\overline{\text{RD}}$ /AN5 RE0 $\overline{\text{RD}}$ AN5	8	9	25	I/O	ST TTL Analog	PORTE is a bi-directional I/O port. Digital I/O. Read control for parallel slave port (see also $\overline{\text{WR}}$ and $\overline{\text{CS}}$ pins). Analog input 5.
RE1/ $\overline{\text{WR}}$ /AN6 RE1 $\overline{\text{WR}}$ AN6	9	10	26	I/O	ST TTL Analog	Digital I/O. Write control for parallel slave port (see $\overline{\text{CS}}$ and $\overline{\text{RD}}$ pins). Analog input 6.
RE2/ $\overline{\text{CS}}$ /AN7 RE2 $\overline{\text{CS}}$ AN7	10	11	27	I/O	ST TTL Analog	Digital I/O. Chip Select control for parallel slave port (see related $\overline{\text{RD}}$ and $\overline{\text{WR}}$). Analog input 7.
Vss	12, 31	13, 34	6, 29	P	—	Ground reference for logic and I/O pins.
Vdd	11, 32	12, 35	7, 28	P	—	Positive supply for logic and I/O pins.



Şekil 4.2 Kristal Bağlantısı

Çizelge 4.3’ deki XTAL, C1 ve C2 değerleri kullanılacaktır.

Çizelge 4.3 Kondansatör Seçimi

Ranges Tested:			
Mode	Freq	C1	C2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF
These values are for design guidance only. See notes following this table.			
Resonators Used:			
455 kHz	Panasonic EFO-A455K04B	± 0.3%	
2.0 MHz	Murata Erie CSA2.00MG	± 0.5%	
4.0 MHz	Murata Erie CSA4.00MG	± 0.5%	
8.0 MHz	Murata Erie CSA8.00MT	± 0.5%	
16.0 MHz	Murata Erie CSA16.00MX	± 0.5%	
All resonators used did not have built-in capacitors.			

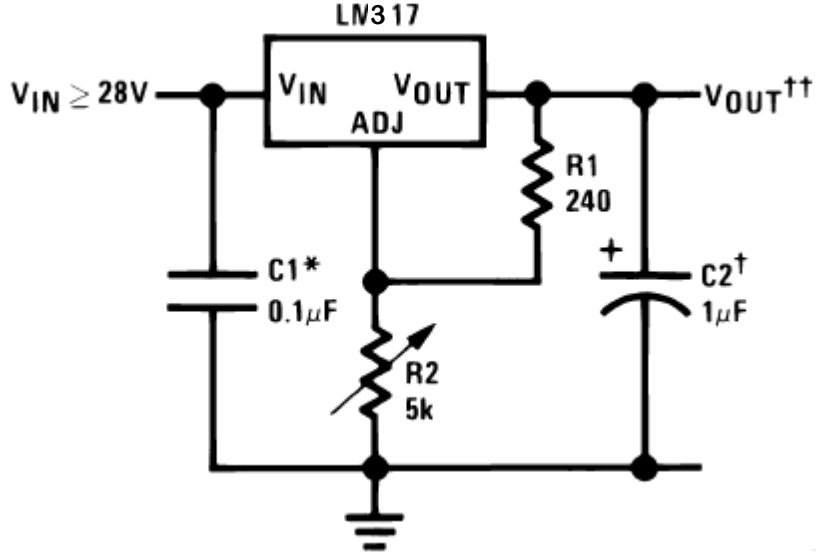
Çizelge 4.3 (Devam)

Ranges Tested:			
Mode	Freq	C1	C2
LP	32.0 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	22-68 pF	22-68 pF
	1.0 MHz	15 pF	15 pF
	4.0 MHz	15 pF	15 pF
HS	4.0 MHz	15 pF	15 pF
	8.0 MHz	15-33 pF	15-33 pF
	20.0 MHz	15-33 pF	15-33 pF
	25.0 MHz	15-33 pF	15-33 pF
These values are for design guidance only. See notes following this table.			
Crystals Used			
32.0 kHz	Epson C-001R32.768K-A	± 20 PPM	
200 kHz	STD XTL 200.000KHz	± 20 PPM	
1.0 MHz	ECS ECS-10-13-1	± 50 PPM	
4.0 MHz	ECS ECS-40-20-1	± 50 PPM	
8.0 MHz	Epson CA-301 8.000M-C	± 30 PPM	
20.0 MHz	Epson CA-301 20.000M-C	± 30 PPM	

4.2 LM317T

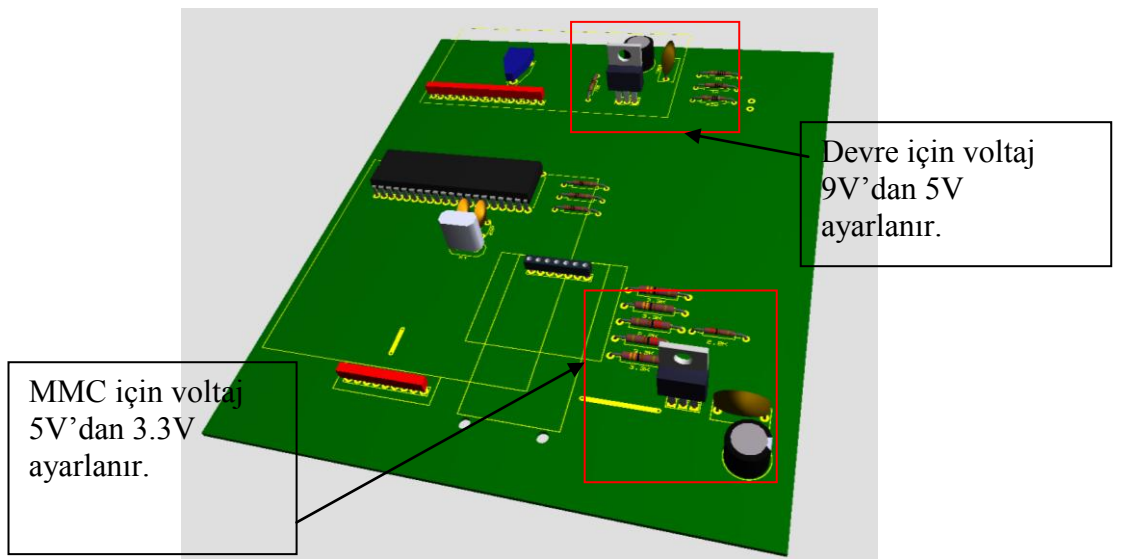
Giriş voltajı 9V olan devrede 2 adet regülatör kullanarak önce voltaj PIC18F452 mikroişlemcisinin kullanabileceği 5V' a sonra ise SD-CARD için 3.3V'a düşürülmüştür.

1.2V–25V Adjustable Regulator



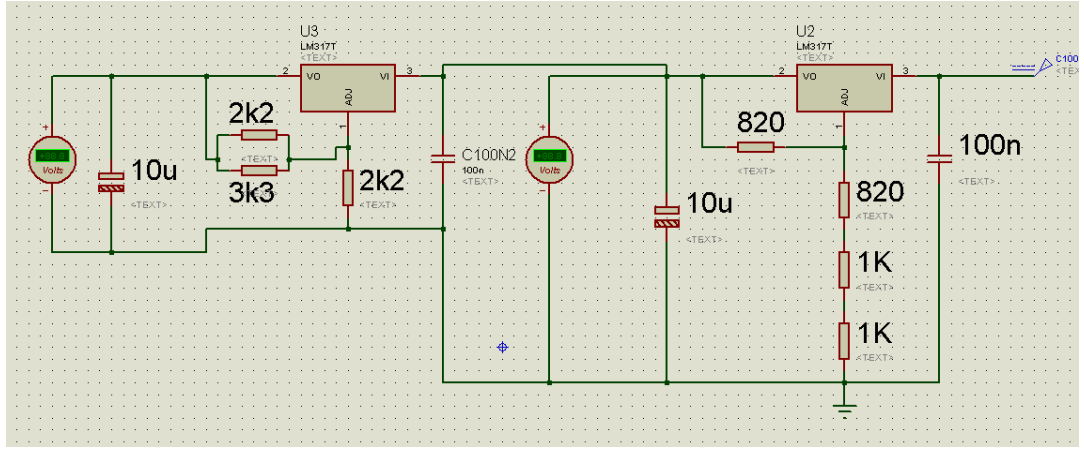
Şekil 4.3 LM317 Devre Şeması

$V_{out} = 1.25V \left(1 + \frac{R_2}{R_1} \right) + I_{ADJ}(R_2)$ formülü vasıtası ile gerekli V_{out} çıkışı direnç değerleri değiştirilerek hesaplanmıştır.



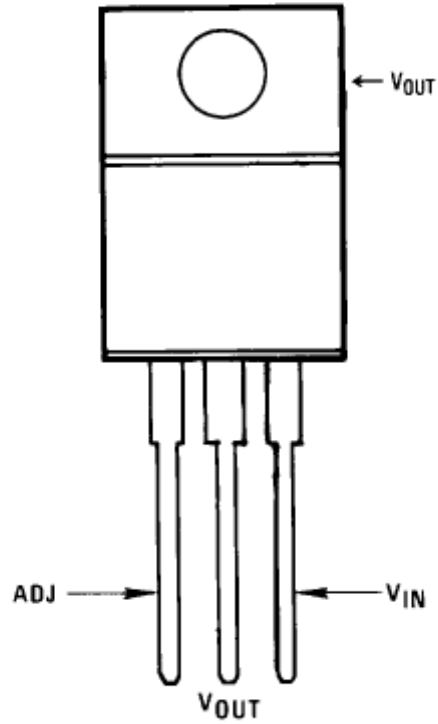
Şekil 4.4 Devrenin 3D Görüntüsü Ve LM317 Yerleri

Kullanılması gereken değerler devrede yerine konduğunda voljda değişiklikler görülmüş ve dirençler reosta yardımı ile ayarlanmıştır.



Şekil 4.5 LM317 Proteus Devresi

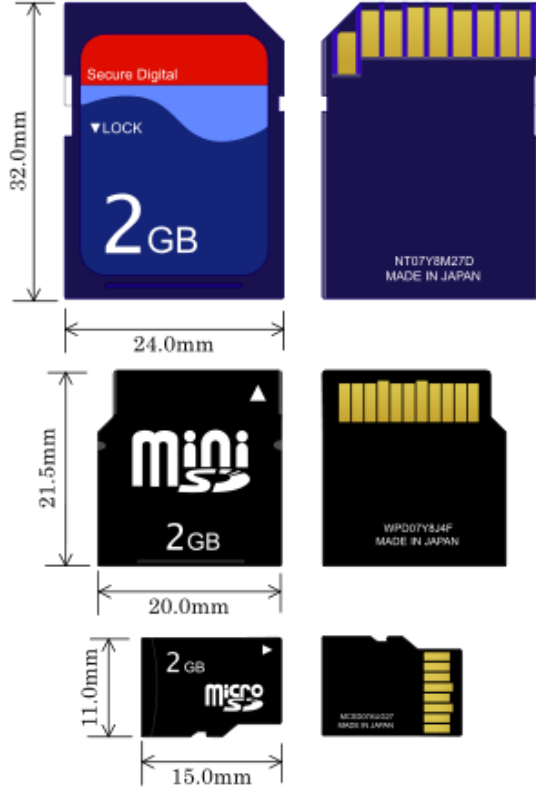
(TO-220)
Plastic Package



Şekil 4.6 LM317 Önden Bacakları

4.3 SD-CARD

SecureDigital Kartları 2001 yılında SanDisk tarafından, daha eski bir standart olan MMC kartının geliştirilmesiyle ortaya çıkmıştır. SecureDigital ismi, donanımsal olarak içerdiği Digital Rights Management (DRM) fonksiyonundan ileri gelmektedir. İçindeki kullanıcı tarafından görünemeyen bir hafıza alanı, yasal olmayan dosyaların kullanımını engellemek için ayrılmıştır.

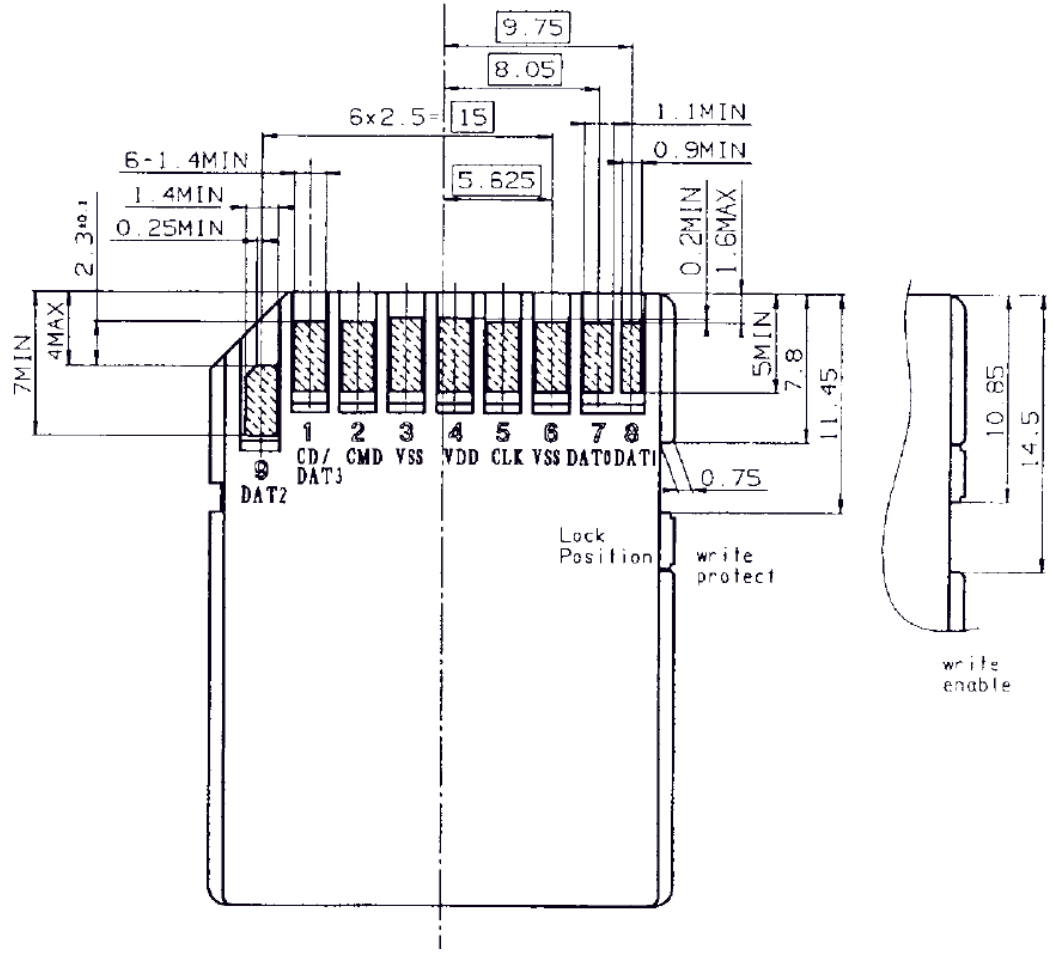


Şekil 4.7 SD-CARD Çeşitleri

Hafıza kartı dahili bir denetleyici içerir. 32 mm × 24 mm × 2,1 mm ebatlarındadır. 8 MB , 16 MB, 32 MB, 64 MB, 128 MB, 256 MB, 512 MB, 1 GB, 2 GB, 4 GB, 8 GB, 16 GB VE 32 GB kapasiteli kartlar mevcuttur. Ayrıca kartlar üzerinde bir adet yazım koruma tırnağı mevcuttur.

Daha ufak cihazlar için miniSD geliştirilmiştir. 20 mm × 21,5 mm × 1,4 mm ölçüleri ile ortalama SD-Card ın yarısı kadardır. Küçük bir çevirici yardımı ile tüm normal SD-Slotlarına takılabilmektedir. Bu ufak kartlar 32 MB ila 2 GB arasında bir kapasiteye sahiptirler.

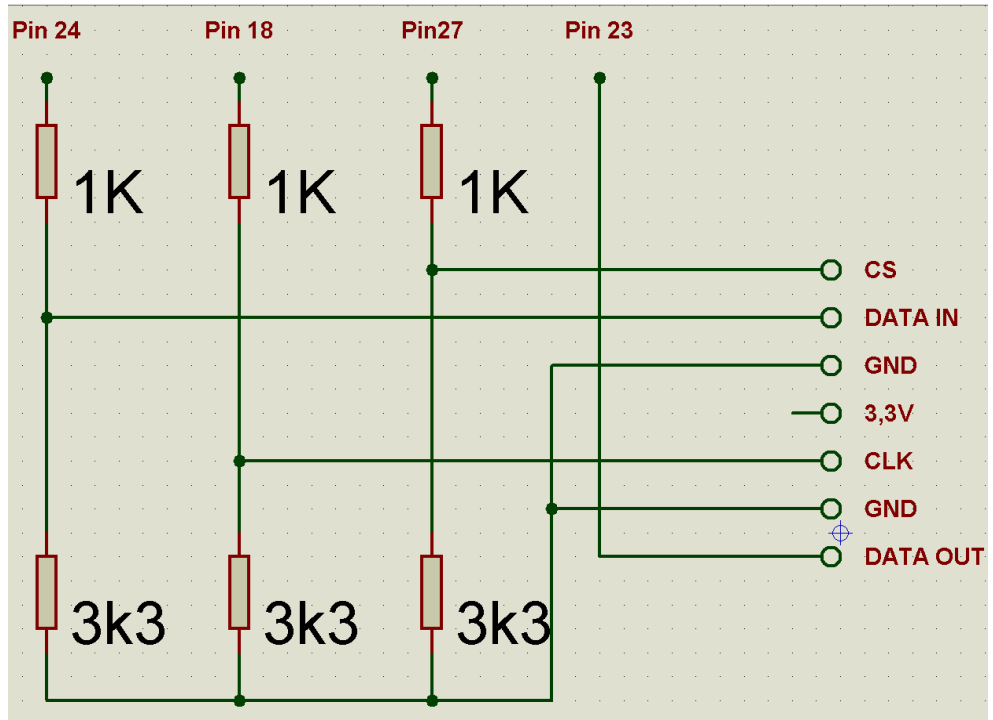
MicroSD-kartları SD Card ve miniSD den daha ufaktırlar. 11 mm × 15 mm × 1 mm ölçüleri ile 2005 te dünyanın en küçük Flash-RAM-hafıza kartıdır.SD kartlar farklı hızlarda olup hız değerleri CD-ROM sürücülerde olduğu gibi X harfiyle belirtilir ve 1X=150kB/s'dir.



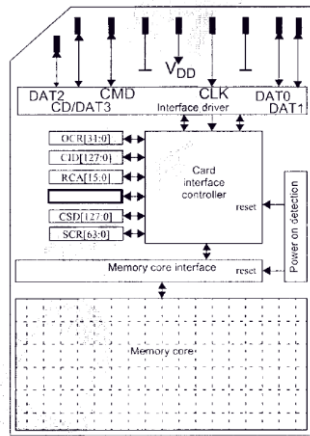
Şekil 4.8 SD-CARD Altın Görünüşü Ve Çıkışları

Çizelge 4.4 SD-CARD Bağlantı Tablosu

Pin #	Name	Type ¹	SPI Description
1	CS	I	Chip Select (Active low)
2	DataIn	I	Host to Card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card to Host Data and Status
8	RSV(2)	I	Reserved
9	RSV(2)	I	Reserved



Şekil 4.9 SD-CARD Microişlemci Arasındaki Bağlantı



Şekil 4.10 SD-CARD Mimarisi



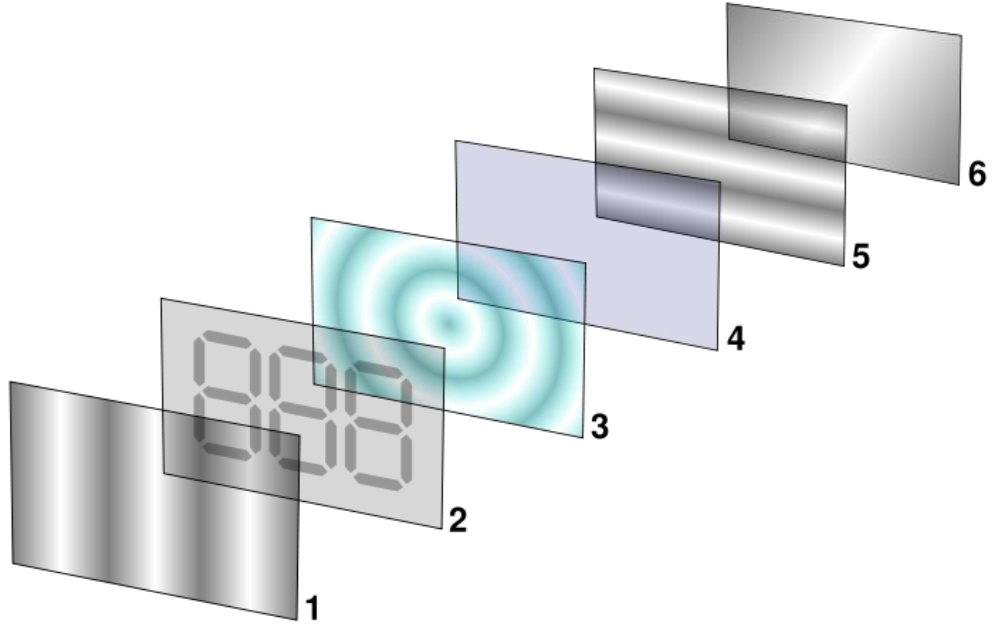
Şekil 4.11 SD-CARD Adaptörü

4.4 PCB1602BA LCD

Sıvı kristal ekran (Liquid Crystal Display), elektrikle kutuplanan (Polarization) sıvının ışığı tek fazlı geçirmesi ve önüne eklenen bir kutuplanma filtresi ile gözle görülebilmesi ilkesine dayanan bir görüntü teknolojisidir. Sıvı kristal ekranlar, düşük enerji tüketimleri ile eskiden kullanılan vakumlu fluoressan ekranların yerini almıştır.

Başlangıçta tek renkli ve çok düşük çözünürlüklü olan sıvı kristal ekranlar; hesap makineleri, saatler, cep telefonları vb. basit görüntüleme işlerinde kullanılmıştır. Nokta matrisli (Dot matrix) yüksek çözünürlüklü ve renkli sürümlerinin ortaya çıkması bu ürünlerin kullanımını yaygınlaştırmıştır.

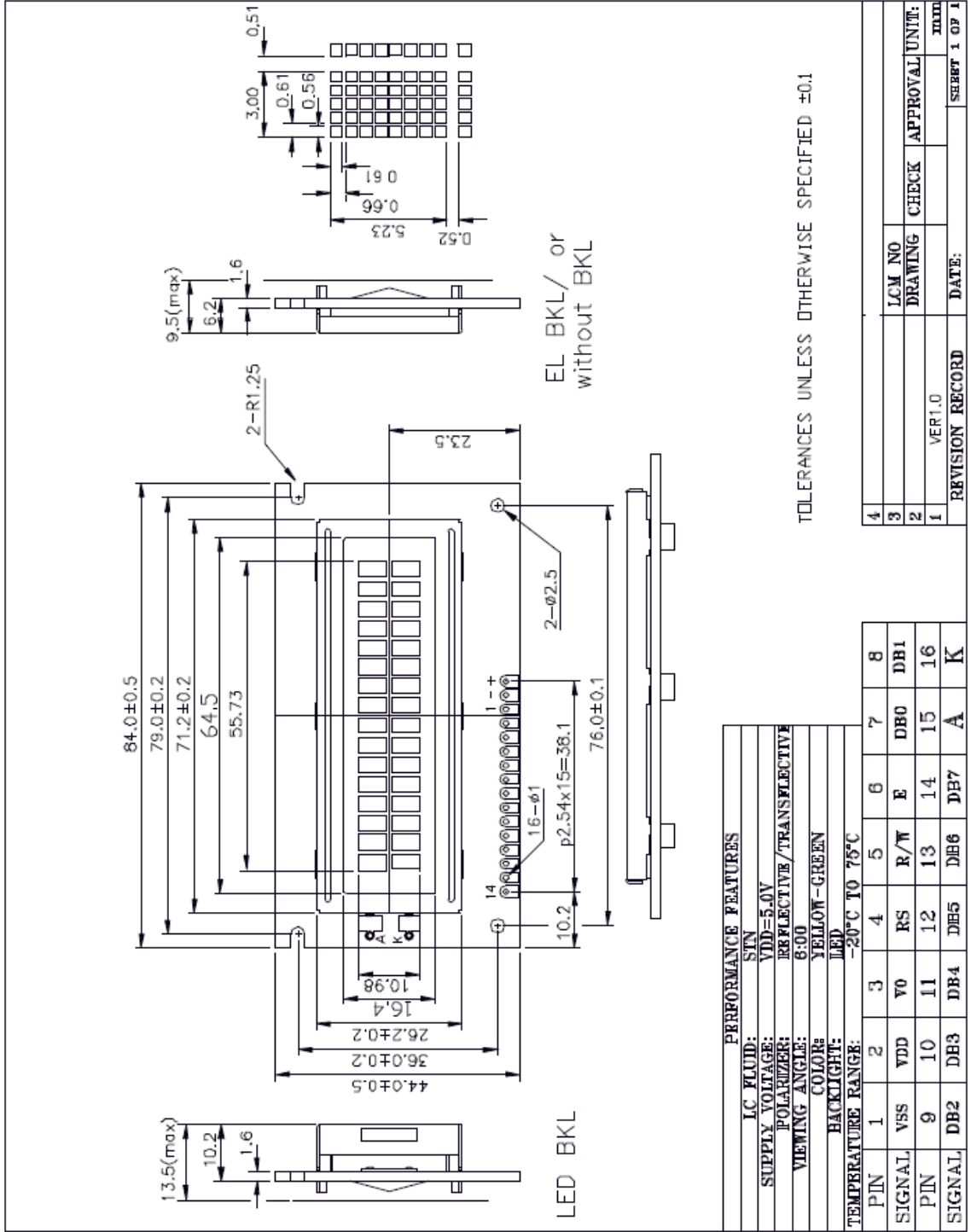
Az yer kaplamaları, düşük enerji tüketimleri ve katot ışınlu tüplere göre yok denecek kadar az radyasyon yayımları nedeni ile bilgisayar ekranları ve TV cihazlarında yoğun olarak kullanılmaktadır. Eskiden tazeleme hızları yavaş olan bu ekranlar hızlı görüntü değişiminde gölgelenme (ghosting) sorunu yaşamaktaydı. Teknolojinin ilerlemesi ile 2 milisaniye civarına kadar inen piksel hızı, bu sorunu ortadan kaldırmıştır.



Şekil 4.12 LCD Yapısı

Yasıtmalı çevrilmiş şematik sıvı kristal ekran (Şekil 4.12).

1. Dikey film filtresi giren ışığı kutuplar etmektedir.
2. ITO elektrodlu cam alt katman. Bu elektrotların şekilleri LCD çalışmadığında oluşan karanlık görüntüden sorumludur. Yüze çizilmiş dikey çizgiler sıvı kristalin kutuplanmış ışık ile aynı hizada olmasını sağlar.
3. Çevrilmiş şematik sıvı kristaller.
4. Yatay filtre ile aynı hizada yerleştirilmiş, yatay çizgili ortak elektrot filmlili (ITO) cam alt katman.
5. Geçen ışığı blok/geçirme özelliğine sahip yatay film filtre.
6. İzleyiciye ışığı geri gönderen yansıtıcı yüzey.



Şekil 4.13 LCD Datasheet

Çizelge 4.5 LCD Bacakları

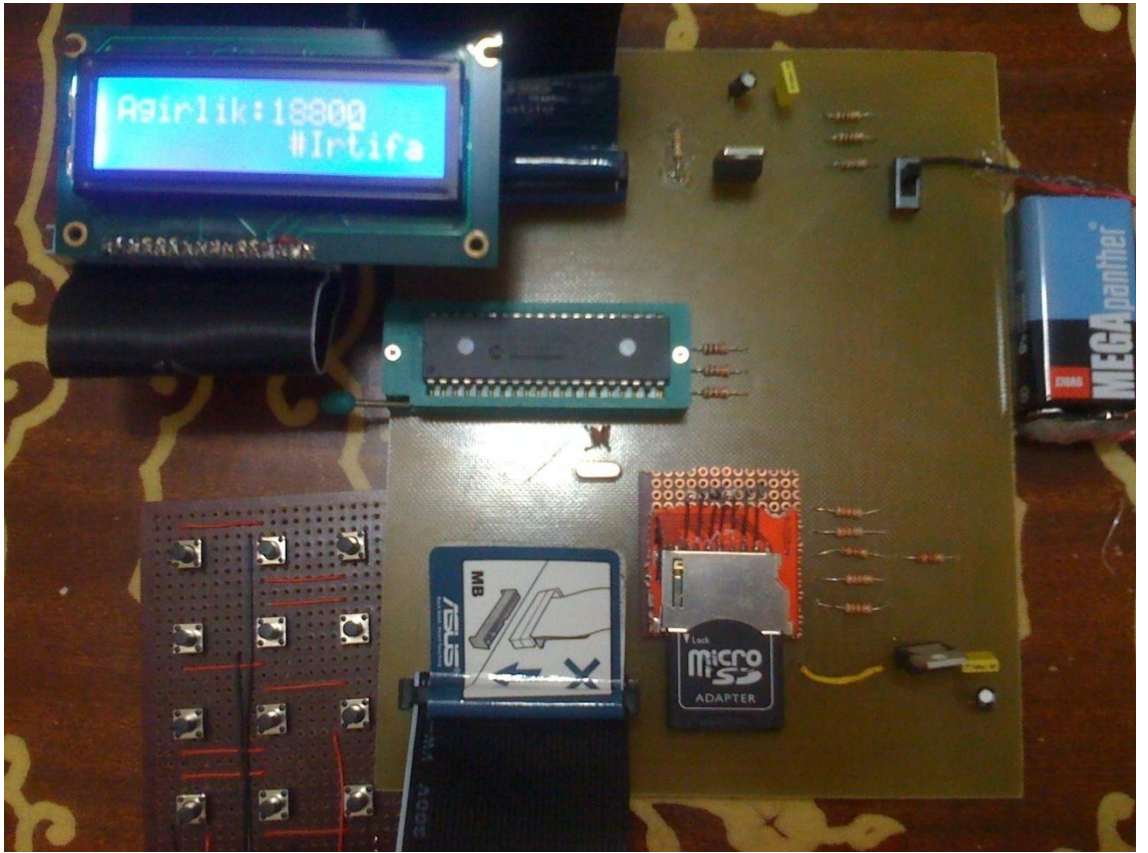
Pin No.	Pin Name	Input/ Output	External Connection	Function
1	VSS	—	Power Supply	VSS:GND
2	VDD	—		VDD: +5V
3	VO	—		V _{LCD} adjustment
4	RS	INPUT	MPU	Register select signal “0”:Instruction register (when writing) Busy flag & address counter (When reading) “1”:Data register (when writing & reading)
5	R/W	Input	MPU	Read/write select signal “0” for writing , “1” for reading
6	E	Input	MPU	Operation (data read/write) enable signal
7 / 10	DB0-DB3	Input	MPU	Low-order lines of data bus with 3-state, bi-directional function for use in data transaction with the MPU. These lines are not used when interfacing with a 4-bit microprocessor.
11 / 14	DB4-DB7	Input	MPU	High-order lines of data bus with 3-state, bi-directional function for use in data transactions with the MPU. DB7 may also be used to check the busy flag.
15 / 16	LED “+” LED “-”	Input	LED BACKLIGHT POWER SUPPLY	LED “+” VOLTAGE TYPE:4.2V MAX : 4.5V LED “-” : GND

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	a	P	`	P				-	夕	三	α	P
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	△	△	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	ε
xxxx0100	(5)		\$	4	D	T	d	t			、	エ	ト	カ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	1	ε	Ω
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ア	キ	ヌ	ラ	g	π
xxxx1000	(1)		(8	H	X	h	x			ィ	ク	ネ	リ	、	Σ
xxxx1001	(2))	9	I	Y	i	y			ウ	ケ	ル	、	、	γ
xxxx1010	(3)		*	:	J	Z	j	z			エ	コ	ハ	レ	j	κ
xxxx1011	(4)		+	;	K	[k	[オ	サ	ヒ	ロ	×	κ
xxxx1100	(5)		,	<	L	¥	l	l			カ	シ	フ	ワ	φ	π
xxxx1101	(6)		-	=	M]	m]			ユ	ス	ハ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	^			ヨ	セ	ホ	、	π	
xxxx1111	(8)		/	?	O	_	o	_			ッ	ソ	マ	、	ö	■

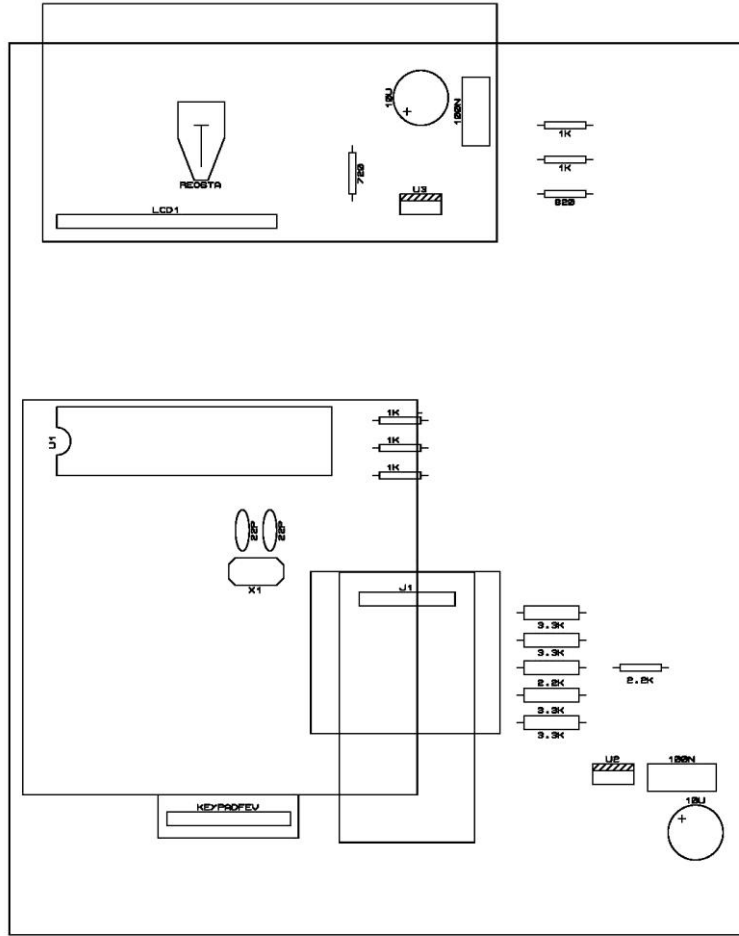
Şekil 4.14 LCD Karakterleri

4.5 Devrenin Çalışma Prensibi

9V pille çalışan devre içerisinde Pic18F452 mikroişlemcisi barındırmaktadır. Bunun yanında LCD display ve delikli plakete el ile hazırlanmış numpad devreye PC IDE kabloları ile bağlanmıştır. Devrede kullanılan SD-CARD modülü SD-CARD okuyucudan kesilerek çıkarılıp lehimle hazırlandıktan sonra devreye iştirilmiştir. Devre 8MHz hızda çalışmaktadır.



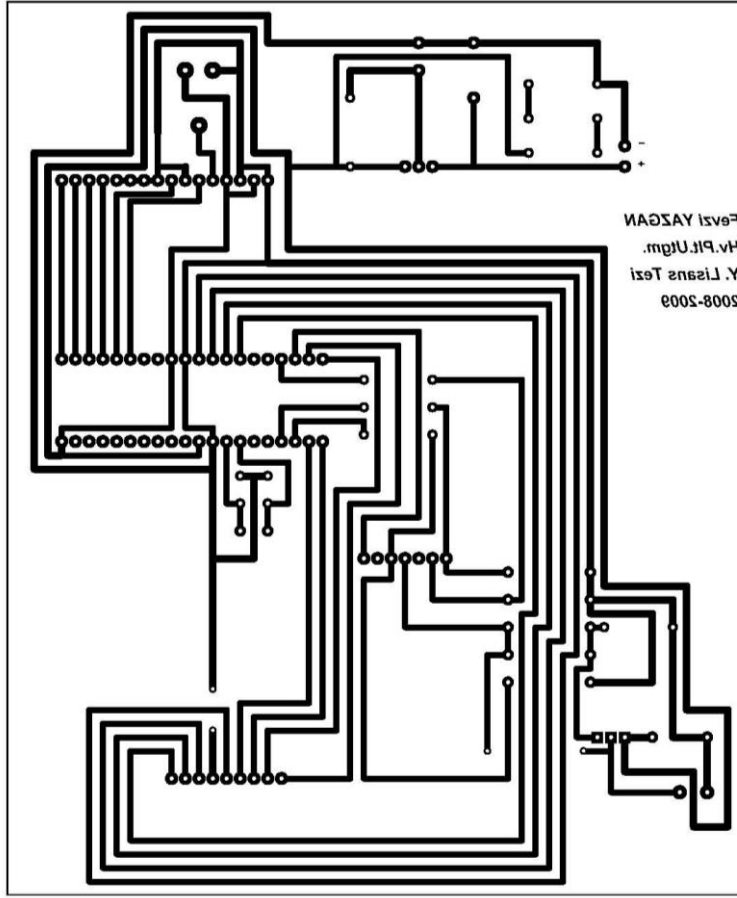
Şekil 4.15 Devrenin Çalışır Haldeki Görüntüsü



Şekil 4.16 Devrenin Üstten Görünüşü

Çizelge 4.6 Devrede Kullanılan Elemanlar

Devrede Kullanılan Elektronik Devre Elemanları			
KODU	TÜRÜ	DEĞERİ	ADEDİ
U1	MİKROİŞLEMCİ	-	1
U2	LM317T	-	2
U3	LM317T	-	
LCD1	PCB1602BA	-	1
J1	SD-CARD MODÜLÜ	-	1
X1	KRİSTAL	8MHz	1
KEYPAD	NUMPAD	-	1
-	DİRENÇ	720	1
-	DİRENÇ	820	1
-	DİRENÇ	1K	5
-	DİRENÇ	2.2K	2
-	DİRENÇ	3.3K	4
-	KONDANSATÖR	22Pf	2
-	KONDANSATÖR	10MicroF	2
-	KONDANSATÖR	100NanoF	2



Şekil 4.17 Devrenin Alttan Görünüşü

5. MİKROPASCAL

Mikropascal pic programlama için hazırlanmış alt seviye bir dildir. Kullanımı oldukça basittir, Pascal dilinden pek farklılığı yoktur.

5.1 Keypad Kullanımı

Aşağıdaki program vasıtası ile basılan tuş LCD ekranada gösterilir.

```
program Key;

var
kp, cnt : byte;
txt : string[5];

begin
  cnt := 0;
  Keypad_Init(PORTD);
  Lcd_Init(PORTB);      // LCD - PORTC
  Lcd_Cmd(LCD_CLEAR);  // LCD TEMİZLE
  Lcd_Cmd(LCD_CURSOR_OFF); // Cursor YOK ET

  Lcd_Out(1, 1, 'Key :');
  Lcd_Out(2, 1, 'Times:');

  repeat
  begin

    kp := 0;

    //--- TUŞA BASILMASILIP BIRAKILMASI İÇİN BEKLER
    while kp = 0 do
    begin
      kp := Keypad_Released();
    end;

    Inc(cnt);

    //--- DEĞERİ ÇIKIŞ İÇİN HAZIRLA
    case kp of
      5 : kp:='0';
      4 : kp:='1';
      8 : kp:='2';
      12 : kp:='3';
      3 : kp:='4';
```

```

7 : kp:='5';
0 : kp:='6';
2 : kp:='7';
6 : kp:='8';
10 : kp:='9';
1 : kp:='*';
5 : kp:='0';
9 : kp:='#';
end;

```

```

//--- LCD'ye yaz
Lcd_Chr(1, 10, kp);
WordToStr(cnt, txt);
Lcd_Out(2, 10, txt);

```

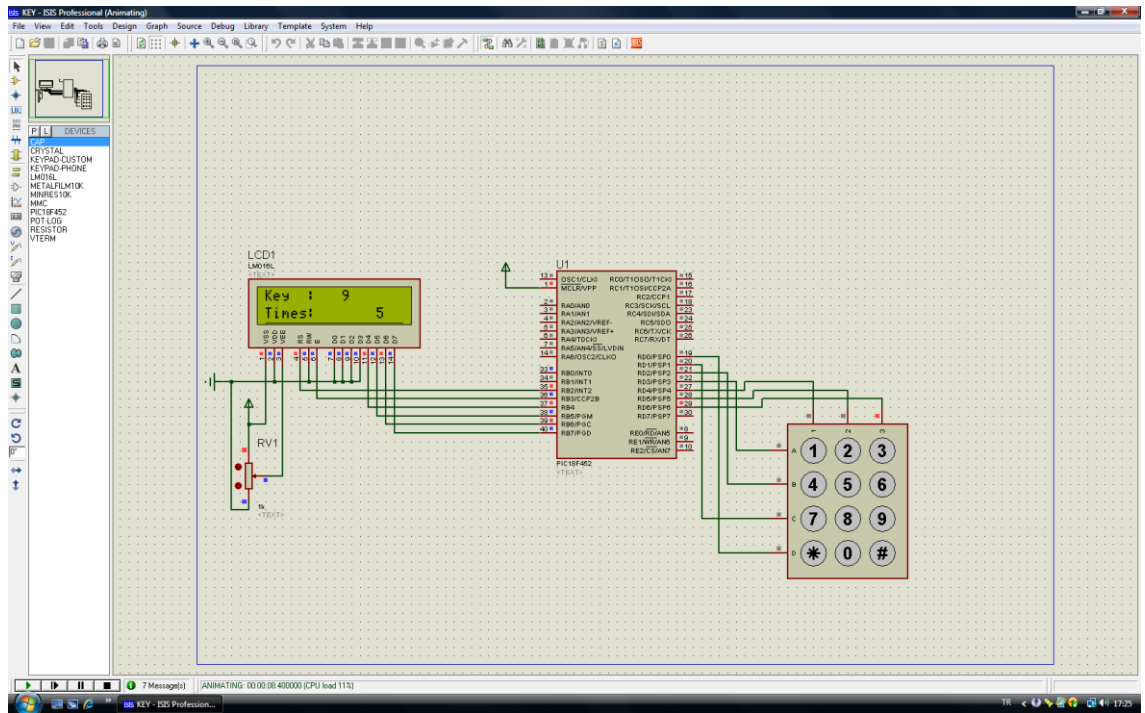
```

end;
until FALSE;

```

end.

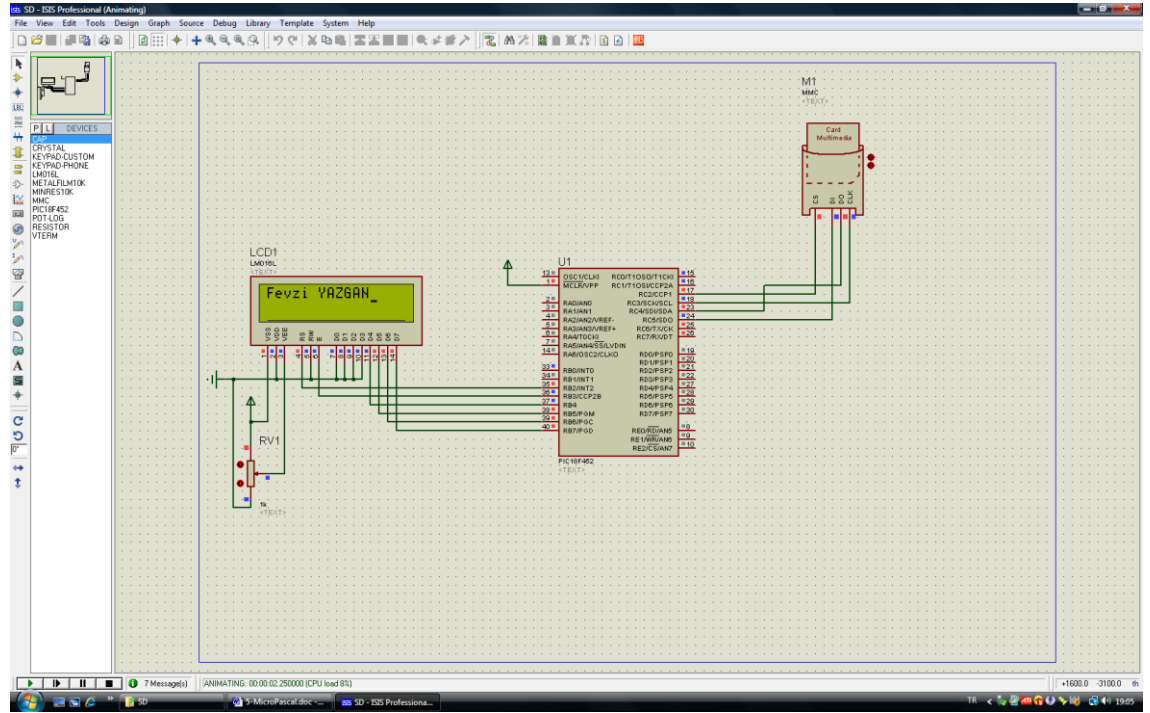
Similasyonu:



Şekil 5.1 Keypad Kullanımı Similasyonu

5.2 SD-CARD'dan Bilgi Okuma

SPI (Serial Peripheral Interface) vasıtası ile Mikroişlemci SD-CARD' dan veri okur. Mikropascal bu iş için hazır fonksiyonları vardır. Aşağıda SD-CARD' dan veri okuyan bir program verilmiştir.



Şekil 5.2 SD-CARD'dan Bilgi Okuma Similasyonu

```
program SD;
var
txt      :string[17];

function mmcaoku(veri:dword):char;
var
error    : word;
sectorNo : dword;
dataBuffer : array[512] of char;
secc,ss  : word;
ap       :string[3];
```

```

begin
  ap:=' ';
  secc:=floor(veri/512);
  sectorNo :=520+secc;
  error := Mmc_Read_Sector(sectorNo, dataBuffer);
  ss:=veri-(secc*512);
  result:=dataBuffer[ss];
end;

begin
  Keypad_Init(PORTD);
  Lcd_Init(PORTB);
  //mmc init işlemi yapılıyor
  PORTC := 0;
  TRISC := 0;
  Spi_Init_Advanced(Master_OSC_div64, DATA_SAMPLE_MIDDLE,
  CLK_IDLE_LOW, LOW_2_HIGH);
  if Mmc_Fat_Init(PORTC, 2) = 0 then
    begin
      // yüksek hızda reinitialize spi
      Spi_Init_Advanced(Master_OSC_div16, DATA_SAMPLE_MIDDLE,
      CLK_IDLE_LOW, LOW_2_HIGH);
      //start sinyali
      PORTC := $F0;
    end
  else
    begin
      Lcd_Out(1, 1, ' Mmc ');
      Lcd_Out(2, 1, ' Algılanamadi ');
      Vdelay_ms(1000);
    end;
    Lcd_Chr(1, 1, mmcaoku(0));
    Lcd_Chr(1, 2, mmcaoku(1));
    Lcd_Chr(1, 3, mmcaoku(2));
    Lcd_Chr(1, 4, mmcaoku(3));
    Lcd_Chr(1, 5, mmcaoku(4));
    Lcd_Chr(1, 6, mmcaoku(5));
    Lcd_Chr(1, 7, mmcaoku(6));
    Lcd_Chr(1, 8, mmcaoku(7));
    Lcd_Chr(1, 9, mmcaoku(8));
    Lcd_Chr(1, 10, mmcaoku(9));
    Lcd_Chr(1, 11, mmcaoku(10));
    Lcd_Chr(1, 12, mmcaoku(11));
  end.

```

5.3 ANA PROGRAM

Pic'e yüklenen program aşağıda belirtilmiştir.

```
program Ana;

var
  we,he,te,wi,hhh      :integer;
  Atxt,Ftxt,Rtxt,PPPP  :string[17];
  MTtxt,TItxt,TOtxt    :string[17];
  TR4txt,TRYtxt       :string[17];
  OItxt,OOtxt         :string[17];
  OR4txt,ORYtxt       :string[17];
  hveri                :dword;
  k                    :byte;
  we1,he1,te1         :word;
  we2,he2,te2         :word;
  m                    :byte;
label basla,a;
function mmcaoku(seccc,veri:dword):integer;
var
  error   : word;
  sectorNo : dword;
  dataBuffer : array[512] of char;
  secc,ss   : dword;
  ccc,i     : byte;
  ap       : string[10];
  scs      : string[10];
label sonnn;
begin
  ap:='      ';
  secc:=(floor(veri/512));
  ss:=veri-(secc*512);
  sectorNo:= secc+seccc;
  error := Mmc_Read_Sector(sectorNo, dataBuffer);
  i:=0;
  for ccc:=0 to 7 do
  begin
  if dataBuffer[ss+ccc]=' ' then
  begin
  ap[i]:=0;
  goto sonnn;
  end;
  ap[i]:=dataBuffer[ss+ccc];
  i:=i+1;
  end;
  ap[i]:=0;
```

```

sonnn:
  result:=StrToInt(ap);
end;
procedure gir(var weight,height,wind,temp :integer);
var
  kp,kpp  : byte;
  wei     : string[5];
  hei     : string[5];
  win     : string[2];
  tem     : string[3];
  temm    : string[3];
  ts      : char;
  cnt     : byte;
label turn,weightt,heightt,windd,temppt,calcc;
begin
// ağırlık bilgisi giriliyor
weightt:
  wei:='  ';
  Lcd_Cmd(LCD_CLEAR);
  Lcd_Out(1, 1, 'Agirlik:  ');
  Lcd_Out(2, 1, '      #Irtifa');
  Lcd_Cmd(LCD_FIRST_ROW);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
  cnt:=-1;
  kpp:=0;
  repeat
  begin
    inc(cnt);
    if kpp<cnt then kpp:=cnt;
    if cnt=5 then
      begin
        cnt:=0;
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        end;
        kp := 0;
        //--- Wait for key to be pressed
        while kp = 0 do

```

```

begin
  //--- un-comment one of the keypad reading functions
  kp := Keypad_Released();
  //kp := Keypad_Read();
end;
case kp of
4  : ts:='1';
8  : ts:='2';
12 : ts:='3';
3  : ts:='4';
7  : ts:='5';
11 : ts:='6';
2  : ts:='7';
6  : ts:='8';
10 : ts:='9';
1  : begin
      wei[kpp]:=0;
      goto weightt;
    end;
5  : ts:='0';
9  : begin
      wei[kpp]:=0;
      goto heightt;
    end;
end;

wei[cnt]:=ts;
Lcd_Chrcp(ts);
Vdelay_ms(200);

end;
until cnt=-1;
// irtifa bilgisi giriliyor
heightt:
hei:='  ';
kpp:=0;
Lcd_Cmd(LCD_CLEAR);
Lcd_Out(1, 1, 'Irtifa:  ');
Lcd_Out(2, 1, '*Agirlik #Ruzgar');
Lcd_Cmd(LCD_FIRST_ROW);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
cnt:=-1;

```

```

repeat
begin
  inc(cnt);
  if kpp<cnt then kpp:=cnt;
  if cnt=5 then
  begin
    cnt:=0;
    Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
    Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
    Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
    Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
    Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
  end;
  kp := 0;
  //--- Wait for key to be pressed
  while kp = 0 do
  begin
    //--- un-comment one of the keypad reading functions
    kp := Keypad_Released();
    //kp := Keypad_Read();
  end;

  case kp of
    4 : ts:='1';
    8 : ts:='2';
    12 : ts:='3';
    3 : ts:='4';
    7 : ts:='5';
    11 : ts:='6';
    2 : ts:='7';
    6 : ts:='8';
    10 : ts:='9';
    1 : begin
      hei[kpp]:=0;
      goto weightt;
    end;
    5 : ts:='0';
    9 : begin
      hei[kpp]:=0;
      goto windd;
    end;
  end;

  hei[cnt]:=ts;
  Lcd_Chrcp(ts);
  Vdelay_ms(200);

end;

```

```

until cnt=-1;
// wind bilgisi giriliyor
windd:
win:=' ';
kpp:=0;
Lcd_Cmd(LCD_CLEAR);
Lcd_Out(1, 1, 'Ruzgar:      ');
Lcd_Out(2, 1, '*Irtifa #OAT  ');
Lcd_Cmd(LCD_FIRST_ROW);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
cnt:=-1;
repeat
begin
inc(cnt);
if kpp<cnt then kpp:=cnt;
if cnt=2 then
begin
cnt:=0;
Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
end;
kp := 0;
//--- Wait for key to be pressed
while kp = 0 do
begin
//--- un-comment one of the keypad reading functions
kp := Keypad_Released();
//kp := Keypad_Read();
end;

case kp of
4  : ts:='1';
8  : ts:='2';
12 : ts:='3';
3  : ts:='4';
7  : ts:='5';
11 : ts:='6';
2  : ts:='7';
6  : ts:='8';
10 : ts:='9';
1  : begin
win[kpp]:=0;

```

```

        goto heightt;
    end;
5   : ts:='0';
9   : begin
        win[kpp]:=0;
        goto temp;
    end;
end;

win[cnt]:=ts;
Lcd_Chr_Cp(ts);
Vdelay_ms(200);

end;
until cnt=-1;
// sıcaklık bilgisi giriliyor
temp;
tem:=' ';
kpp:=0;
Lcd_Cmd(LCD_CLEAR);
Lcd_Out(1, 1, 'OAT : ');
Lcd_Out(2, 1, '*Irtifa #Hesap ');
Lcd_Cmd(LCD_FIRST_ROW);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
Lcd_Cmd(LCD_MOVE_CURSOR_RIGHT);
cnt:=-1;
repeat
begin
    inc(cnt);
    if kpp<cnt then kpp:=cnt;
    if cnt=3 then
    begin
        cnt:=0;
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
        Lcd_Cmd(LCD_MOVE_CURSOR_LEFT);
    end;
    turn:
    kp := 0;
    //--- Wait for key to be pressed
    while kp = 0 do
    begin
        //--- un-comment one of the keypad reading functions

```

```

    kp := Keypad_Released();
    //kp := Keypad_Read();
end;

case kp of
4   : if cnt=0 then ts:='+' else ts:='1';
8   : if cnt=0 then ts:='- ' else ts:='2';
12  : if cnt=0 then goto turn else ts:='3';
3   : if cnt=0 then goto turn else ts:='4';
7   : if cnt=0 then goto turn else ts:='5';
11  : if cnt=0 then goto turn else ts:='6';
2   : if cnt=0 then goto turn else ts:='7';
6   : if cnt=0 then goto turn else ts:='8';
10  : if cnt=0 then goto turn else ts:='9';
1   : begin
        tem[kpp]:=0;
        goto windd;
    end;
5   : if cnt=0 then goto turn else ts:='0';
9   : begin
        tem[kpp]:=0;
        goto calcc;
    end;
end;

tem[cnt]:=ts;
Lcd_Chrcp(ts);
Vdelay_ms(200);

end;
until cnt=-1;
calcc:
temm:='  ';
if tem[0]='+' then
begin
temm[0]:=tem[1];
temm[1]:=tem[2];
temm[2]:=0;
temp:=StrToInt(temm);
end
else
begin
temm[0]:=tem[0];
temm[1]:=tem[1];
temm[2]:=tem[2];
temm[3]:=0;
temp:=StrToInt(temm);
end;

```

```

weight:=StrToInt(wei);
height:=StrToInt(hei);
wind:=StrToInt(win);

if (weight>19800) or (weight<9000) or (height>10000) or (height<0) or (temp>50) or
(temp<=-50) or (wind>50) or (wind<0) then
begin
Lcd_Out(1, 1, ' Parametreleri ');
Lcd_Out(2, 1, 'Yanlış girdiniz!');
Vdelay_ms(1000);
goto weightt;
end;
end;
begin
Keypad_Init(PORTD);
Lcd_Init(PORTB);

//mmc init işlemi yapılıyor
PORTC := 0;
TRISC := 0;
Spi_Init_Advanced(Master_OSC_div64, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
if Mmc_Fat_Init(PORTC, 2) = 0 then
begin
// yüksek hızda reinitialize spi
Spi_Init_Advanced(Master_OSC_div16, DATA_SAMPLE_MIDDLE,
CLK_IDLE_LOW, LOW_2_HIGH);
//start sinyali
PORTC := $F0;
end
else
begin
Lcd_Out(1, 1, ' Mmc ');
Lcd_Out(2, 1, ' Algılanamadi ');
Vdelay_ms(1000);
end;
//program başlıyor
basla:
gir(we,he,wi,te);
//A NOKTASI
he1:=ceil((he+100)/100)-1;
we1:=ceil((we-10950)/50)-1;
te1:=(te+40);
hveri:=(((he1*(177*81)))+(we1*81)+te1)*8);
//579
IntToStr(mmcaku(579,hveri), Atxt);
//FLYAWAY
if he>3000 then he2:=30 else he2:=ceil((he+100)/100)-1;

```

```

we2:=ceil((we-12950)/50)-1;
hverl:=(((we2*31*81*41)+(he2*81*41)+(te1*41)+wi)*8)+296;
IntToStr(mmcaoku(23145,hveri), Ftxt);
//MINROTOR
hverl:=(he1*(137*81));
hverl:=hverl+(we2*81);
hverl:=hverl+te1;
hverl:=hverl*8;
hverl:=hverl+32;
IntToStr(mmcaoku(243584,hveri)+22, Rtxt);
//MAXTAKEOFFWEIGHT
te2:=41-te;
hverl:=(((he1*81)+(te2-1))*8)+264;
IntToStr(mmcaoku(261096,hveri), MTtxt);
//TWINIGEHOVER
hverl:=(((he1*81)+(te2-1))*8)+72;
IntToStr(mmcaoku(261309,hveri), Tltxt);
//TWINOGEHOVER
hverl:=(((he1*81)+(te2-1))*8)+296;
IntToStr(mmcaoku(261403,hveri), TOtxt);
//TWINRATE45 118953039
hverl:=(((he1*(177*81))+(we1*81)+te1)*8)+200;
IntToStr(mmcaoku(261734,hveri), TR4txt);
//TWINRATEVY 138221967
hverl:=(((he1*(177*81))+(we1*81)+te1)*8)+8;
IntToStr(mmcaoku(299369,hveri), TRYtxt);
//TOEIGEHOVER 149806263
hverl:=(((he1*81)+(te1))*8)+304;
IntToStr(mmcaoku(321994,hveri), OOttx);
//TOEOGEHOVER 149915127
hverl:=(((he1*81)+(te1))*8)+112;
IntToStr(mmcaoku(322207,hveri), OIttx);
//BTOERATE45 150023991
hverl:=(((he1*(177*81))+(we1*81)+te1)*8)+432;
IntToStr(mmcaoku(322419,hveri), OR4txt);
//BTOERATEVY 169292919
hverl:=(((he1*(177*81))+(we1*81)+te1)*8)+240;
IntToStr(mmcaoku(360054,hveri), ORYtxt);

//MENU
m:=1;
repeat

IF m=1 then
  BEGIN
  Lcd_Cmd(LCD_CLEAR);
  Lcd_Out(1, 1, 'A NOKTASI :');
  Lcd_Out(1, 11, Atxt);

```

```

    Lcd_Out(2, 1, 'FLYAWAY :');
    Lcd_Out(2, 11, Fttxt);
    END;
IF m=2 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'FLYAWAY :');
    Lcd_Out(1, 11, Fttxt);
    Lcd_Out(2, 1, 'MINROTOR :');
    Lcd_Out(2, 11, Rtxt);
    END;
IF m=3 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'MINROTOR :');
    Lcd_Out(1, 11, Rtxt);
    Lcd_Out(2, 1, 'MAXTAKEOFF:');
    Lcd_Out(2, 11, MTtxt);
    END;
IF m=4 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'MAXTAKEOFF:');
    Lcd_Out(1, 11, MTtxt);
    Lcd_Out(2, 1, 'TWIGEHOVER:');
    Lcd_Out(2, 11, TItxt);
    END;
IF m=5 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'TWIGEHOVER:');
    Lcd_Out(1, 11, TItxt);
    Lcd_Out(2, 1, 'TWOGEHOVER:');
    Lcd_Out(2, 11, TOtxt);
    END;
IF m=6 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'TWOGEHOVER:');
    Lcd_Out(1, 11, TOtxt);
    Lcd_Out(2, 1, 'TWRATE45 :');
    Lcd_Out(2, 11, TR4txt);
    END;
IF m=7 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'TWRATE45 :');
    Lcd_Out(1, 11, TR4txt);

```

```

    Lcd_Out(2, 1, 'TWRATEVY :');
    Lcd_Out(2, 11, TRYtxt);
    END;
IF m=8 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'TWRATEVY :');
    Lcd_Out(1, 11, TRYtxt);
    Lcd_Out(2, 1, 'OEOGEHOVER:');
    Lcd_Out(2, 11, OOttx);
    END;
IF m=9 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'OEOGEHOVER:');
    Lcd_Out(1, 11, OOttx);
    Lcd_Out(2, 1, 'OEIGEHOVER:');
    Lcd_Out(2, 11, OIttx);
    END;
IF m=10 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'OEIGEHOVER:');
    Lcd_Out(1, 11, OIttx);
    Lcd_Out(2, 1, 'OERATE45 :');
    Lcd_Out(2, 11, OR4txt);
    END;
IF m=11 then
    BEGIN
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Out(1, 1, 'OERATE45 :');
    Lcd_Out(1, 11, OR4txt);
    Lcd_Out(2, 1, 'OERATEVY :');
    Lcd_Out(2, 11, ORYtxt);
    END;

    k := 0;

    --- Wait for key to be pressed
    while k = 0 do
    begin
        --- un-comment one of the keypad reading functions
        k := Keypad_Released();
        //kp := Keypad_Read();
    end;

    case k of

```

```
1  :  
begin  
if m<1 then m:=2;  
m:=m-1;  
end;  
5  : goto basla;  
9  :  
begin  
if m>10 then m:=10;  
m:=m+1;  
end;  
end;  
//  
  
until k=4;  
  
//mmc kapatılıyor  
PORTC := $0F;  
Lcd_Cmd(LCD_CLEAR);  
  Lcd_Out(2, 1, '  KAPANDI  ');  
end.
```

6. SONUÇ

Bu tez çalışmasında, Hava Kuvvetlerinde Helikopter Pilotu olarak tecrübe ettiğim önlenebilir bazı riskleri minimuma indirmeyi amaçladım. Performans planı hazırlarken değerlerin hassas bir şekilde hesaplanması gerektiğini ve kesinlikle insan hatasına yer verilmemesi gerektiğini açıkça tecrübe etmişim. Bu da ancak yüksek teknoloji kullanılarak yapılabilir. El ile hesaplama gibi yöntemler günümüzde geçerliliğini yitirmiş ve yüksek teknoloji ile dolu olan araçlar için yetersiz kalmaktadır. Yaptığım çalışmada bilgisayar kullanarak bu performans grafiklerindeki kullanılan parametreler içerisine alacak tüm olasılıkları hesaplatarak hazır bir veri tabanı elde ettim. Microişlemci vasıtası ile bu verilere çok hızlı bir şekilde ulaşmayı sağladım. İleride yapılacak çalışmalarda yine aynı amaç güdülerek değişik yöntemler de uygulanabilir. Bu yöntemlere yapay sinir ağları, eğri uydurma yöntemleri gibi örnekler verilebilir.

Bu çalışma, tüm matematiksel grafik kullanan araçlar için pratik bir çözüm olabilir. Yaptığım bu elektronik devre oldukça modüler olup 1 gün içerisinde istenilen başka bir amaç doğrultusunda programlanabilir. Aynı zamanda ileride gerekli izinler uçan platformların üreticilerinden alınabilirse, platformun kendisine bu elektronik devre modüle edilebilir, bilgi girişi yapmaya gerek duymadan bilgiler platformun kendisinden güncellenerek elde edilir duruma getirilebilir. Bu sayede bilgi girişine harcanacak zaman ve dikkat de uçuşa verilebilir.

Öncelikle uçuş emniyeti için düşünülen bu devre, Hava Kuvvetlerindeki tüm uçan platformların uçuş emniyetine katkıda bulunması amacıyla yapılmıştır. Fakat diğer Kuvvetlerde ve sivil sektörde uçan platformlarda da kullanılabilir. Hatta aynı tarzda grafik kullanan yada grafik kullanmasada kompleks hesapların gerektiği her yerde bu tür taşınabilir bir cihaz vasıtası ile işlemler çok kolay ve hatasız yerine getirilebilir.

KAYNAKLAR

- [1] ALTINBAŞAK, Orhan. 2000. Mikrodenetleyiciler ve PIC Programlama. ALTAŞ YAYINCILIK, 240, İstanbul.
- [2] AK, Nursel. 2007. Pic Programlama. ALFA YAYINLARI, 220, İstanbul.
- [3] Gonzalez, R.E. Woods, S.L. Eddins and Rafael. 2009. Digital Image Processing Using MATLAB. Prentice Hall, 624, New Jersey.
- [4] AS-532 COUGAR FLIGHT MANUEL, Volume 1, Bölüm 5.1
- [5] http://tr.wikipedia.org/wiki/S%C4%B1v%C4%B1_kristal_ekran
- [6] <http://www.kontrolkalemi.com/forum/pic-programlama/>
- [7] <http://www.mikroe.com/en/books/picbook/picbook.htm/>
- [8] <http://www.elektrotekno.com/forum-7.html>

ÖZGEÇMİŞ

1981 yılında Balıkesir’de doğdu. İlk, orta öğrenimini Balıkesir’de tamamladı. 2000 yılında Maltepe Askeri Lisesini bitirdi. Aynı yıl Hava Harp Okulu, Elektronik Mühendisliği Bölümü’nde lisans eğitimine başladı. 2004 yılında teğmen rütbesiyle 2 nci Ana Jet Üs Uçuş Eğitim Merkezi Komutanlığı’na atandı. 2006 yılında pilot olarak 3 ncü Ana Jet Üssü Arama Kurtarma Komutanlığına atandı.

2008 yılında yüksek lisans eğitimine o yıl ilk defa uygulamaya giren Uzaktan Eş Zamanlı olarak başladı.

Bekar olup, iyi derecede İngilizce bilmektedir