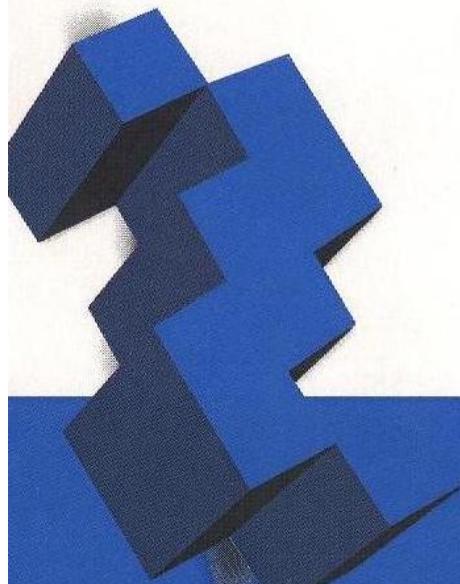
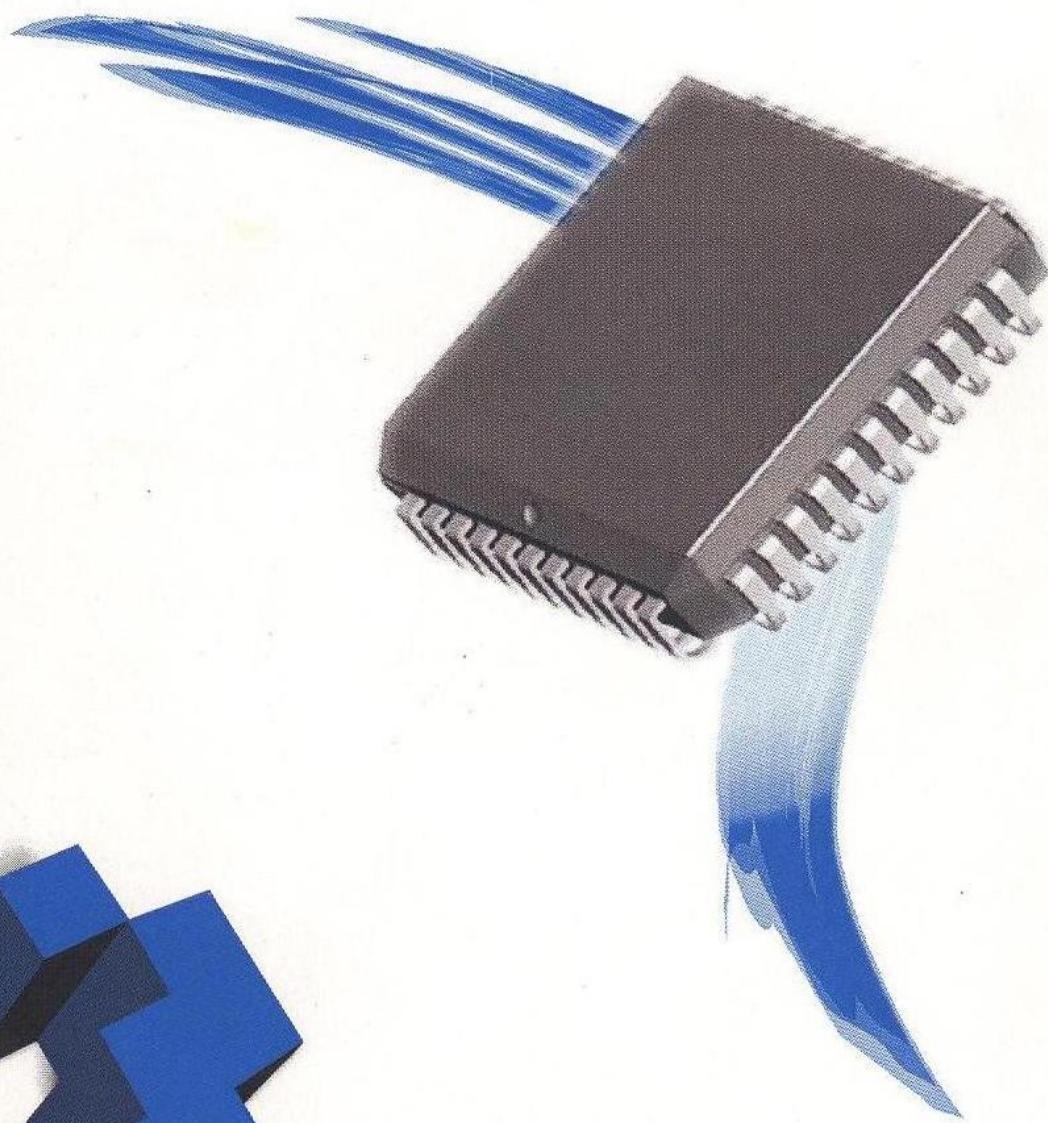


8051 Deney Seti



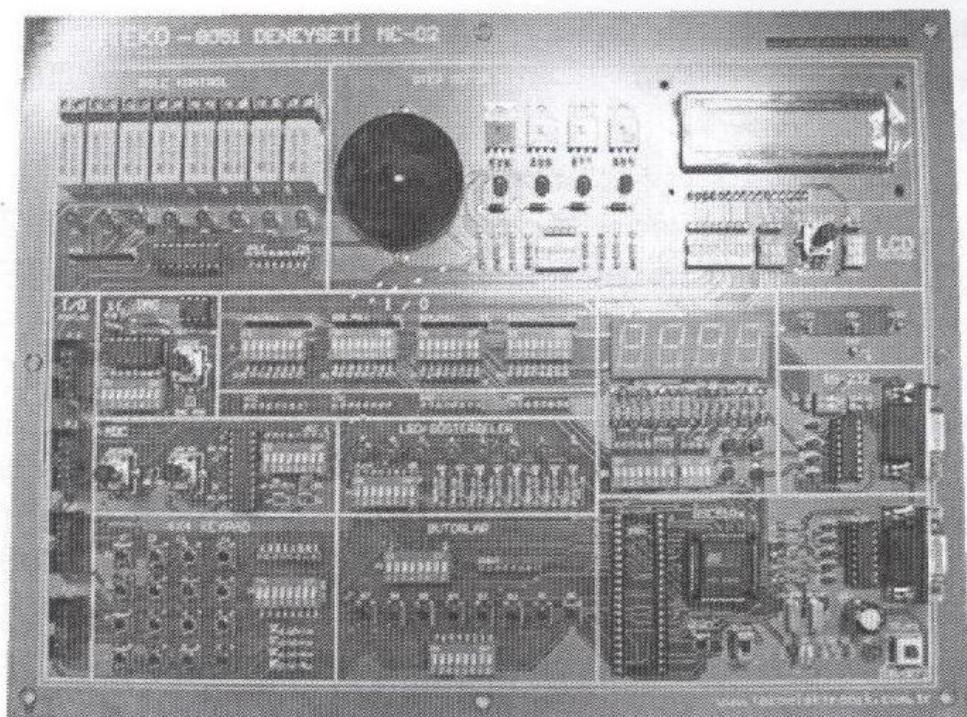
www.tekoelektronik.com.tr

 **TEKO**
ELEKTRONİK

---İÇİNDEKİLER---

8051 Deney Seti Genel Görünüm	2
Teknik-Donanım Özellikleri	3
Deney Seti Modülleri	4
Modül Görünümleri	5
Programlama Modülü	6
Atmel Flip	8
AT89C51RC2 Atmel Flip'le Programlanması	9
KEIL uVision Derleyicisi Tanıtımı	13
KEIL uVision ile Yapılan Örnekler	15
Örnek 1: Buton Kontrol	15
KEIL de Proje Oluşturma, Derleme	16
Örnek 2: Ledler	18
Örnek 3: Keypad	19
Örnek 4: Röleler	21
Örnek 5: Matrix Display	22
Örnek 6: DAC	25
Örnek 7: ADC + RS232	26
Örnek 8: LCD	27
Örnek 9: Step Motor	28

8051 DENEY SETİ GENEL GÖRÜNÜM





8051 DENEY SETİ TEKNİK ÖZELLİKLERİ:

Devre üzerinde programlamaya müsait, çok amaçlı **endüstriyel** bir deney setidir. İşlemciyi **soketinden çıkarmadan** yazılan programı yükleyebilir, tek bir switch ile devreye alabilirsiniz. Bu özellik laboratuar çalışmalarınızı kolaylaştıracaktır.

Set üzerindeki tüm modüller 8051'in portlarına dipswitchler sayesinde bağlıdır. Ancak istenirse giriş ve çıkış modüllerindeki eleman ve yükler CPU'nun istenilen I/O portuna 1mm deney kablolarıyla ya da her modüldeki dipswitchler ile sabit portlara bağlanabilmektedir. Bu kablo ve zaman tasarrufu sağlayacaktır.

Her port ve pinleri I/O modülündeki dip-switchler yardımıyla PULL-UP yapılmaktadır. Bu sayede open-drain olan çıkışlar active-high (lojik 1) çalıştırılabilmektedir. Set plastik kapalı bir kutu içinde muhafazalandır.

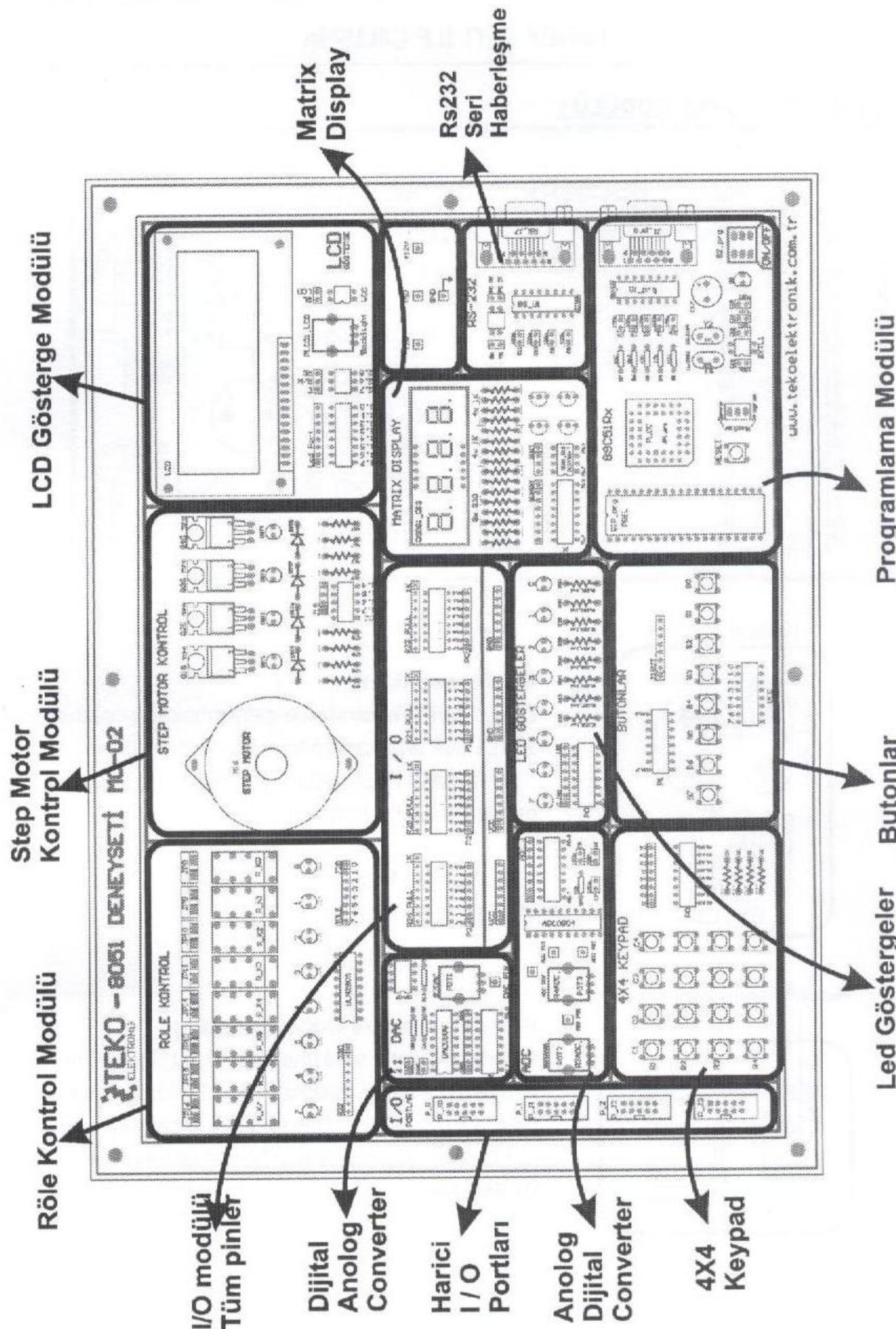
8051 Deney Seti sizin amatör veya profesyonel düzeyde uygulama geliştirme olanağını sağlamaktadır.

STANDART DONANIM :

- **Besleme Gerilimi** : 220V AC /50 Hz. (\pm % 10)
- **Güç Kaynağı** : SMPS (SWITCHMODE) +5V , +12V, 2A Kısa devre Korumalı. Deney setinin güç kaynağı deneylerin yapılabilmesi için gerekli olan tüm gerilimleri üretmektedir.
- **Güç Kablosu** : IEC / 1,5 M Topraklı, EMI Filitreli
- Deney setinin beslemesi açma kapama anahtarı ile kontrol edilip bir adet LED ile de gözlenebilir.
- Deney seti üzerindeki mikro denetleyici SERİPORT (RS232) aracılığıyla programlanabilir.
- Mikro denetleyici portları ile deney seti üzerindeki blokların bağlantıları **1mm** bağlantı kabloları ile istenildiği gibi değiştirilebilir yapıdadır.
- Deney seti üzerindeki deney bloklarının giriş ve çıkışları herhangi bir I / O portuna fix (sabit) bağlı değildir. Ancak istenilirse hemen yanında bulunan dipswitchler yardımı ile kablo kullanmadan belli bir porta hemen sabitlenebilirler.
- Deney seti üzerinde bulunan "I / O PORTLAR" modülü sayesinde dış ortamındaki ek donanımlarınızla haberleşmeniz, Mekatronik uygulamalarını yapmanızı sağlar.

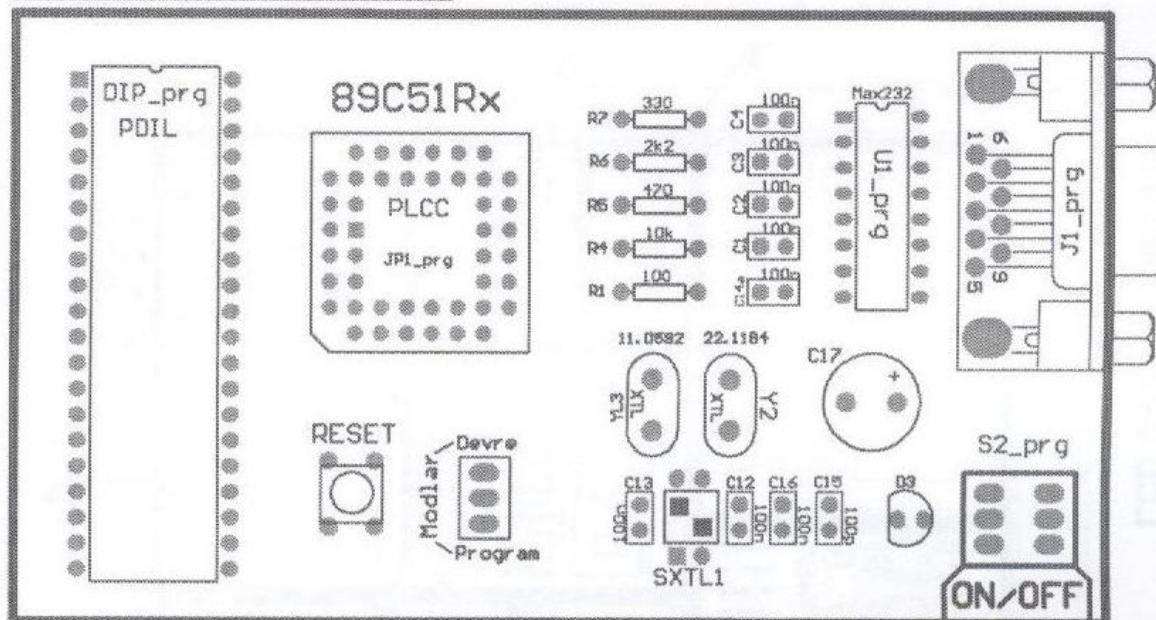
DENEY SETİ MODÜLLERİ :

- PROGRAMLAMA MODÜLÜ (89C51Rx işlemcileri PLCC + DIP Soket)
- BUTONLAR (8 ADET)
- 4x4 KEYPAD
- 4X IDC 16 (HARİCİ PORTLAR)
- RS232 MODÜL
- 8X LED GÖSTERGE
- ADC MODÜLÜ (ADC0804 + AYARLANABİLİR REF.)
- DAC MODÜLÜ (DAC0808)
- RÖLE KONTROL (8X 5V RÖLE + ULN2803)
- STEPMOTOR KONTROL
- MATRIX DISPLAY (4X 7 SEGMENT DISPLAY)
- LCD GÖSTERGE (2X16 KARAKTER LCD BACKLIGHTLI)
- RESET DEVRESİ
- SEÇİLEBİLİR KRİSTAL (11,0592 – 22,1184)
- AT89C51RC2 İŞLEMCI
- 8051 tabanlı 8 bit CISC Mimari, 32 I/O, 3 Timers/Counters, SPI, UART, 32K FLASH, 256 Bytes RAM

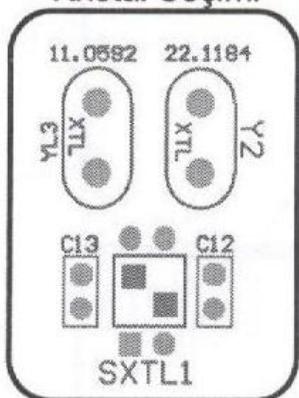


DENEY SETİ İLE ÇALIŞMA

1. PROGRAMLAMA MODÜLÜ :



Kristal Seçimi



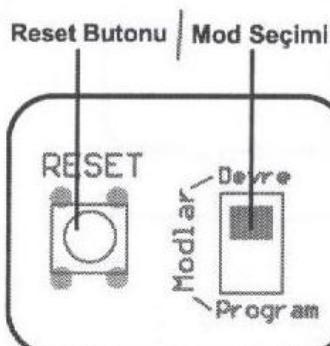
Kristal osilatör:

8051 'i farklı freksanslarda çalıştırılmak istiyorsanız SXTL1 DIP anahtarının konumunu değiştiriniz.

Varsayılan kristal değeri için resimde görülen bağlantıyı gerçekleştiriniz.

YL2: 22,1184 Mhz

YL3: 11,0592 Mhz



Programlama ve Deney Modları:

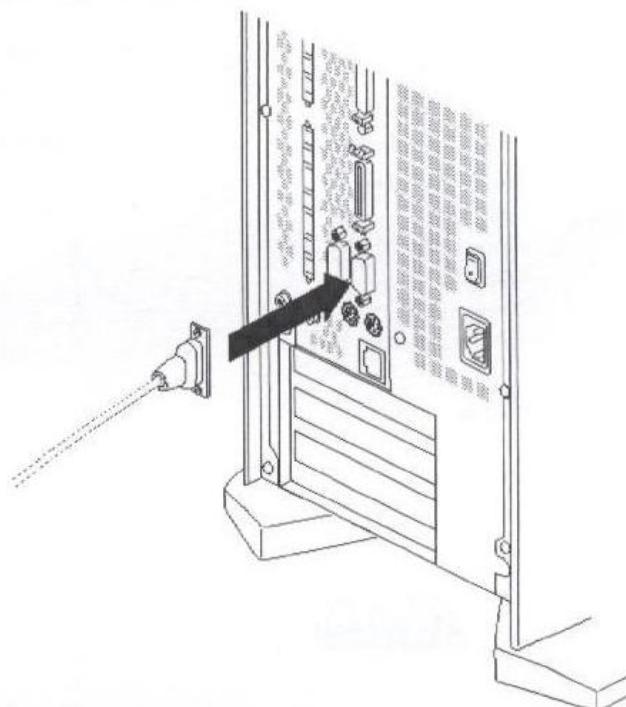
Programı CHIP e yüklemek için "Mod Seçimi" ni

Program a alınız. Program yüklendikten sonra programınızın 8051 Deney Setinde çalıştırılması için **Devre** konumuna alıp "Reset Butonu" ile **RESET** leyiniz.

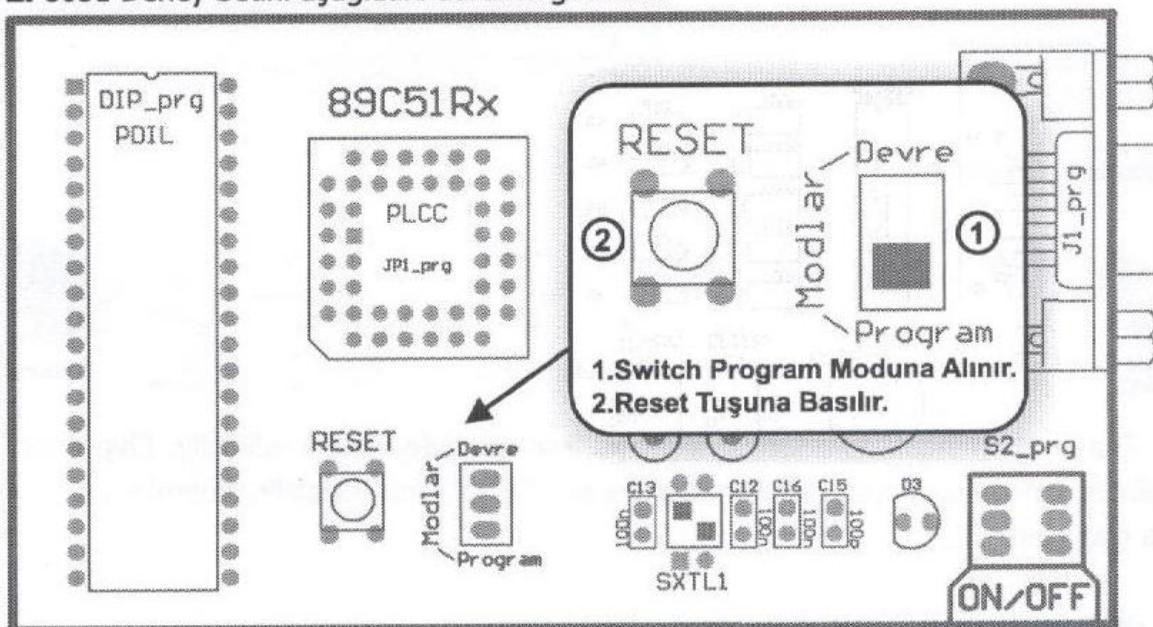
DENEY SETİNİN PROGRAMLAMA MODUNA ALINMASI

1. 8051 Deney setinin programlama modülündeki J1_prg (Seri Port) portunu Bilgisayarınızdaki COM1 (Seri Port) 'e bağlayınız.

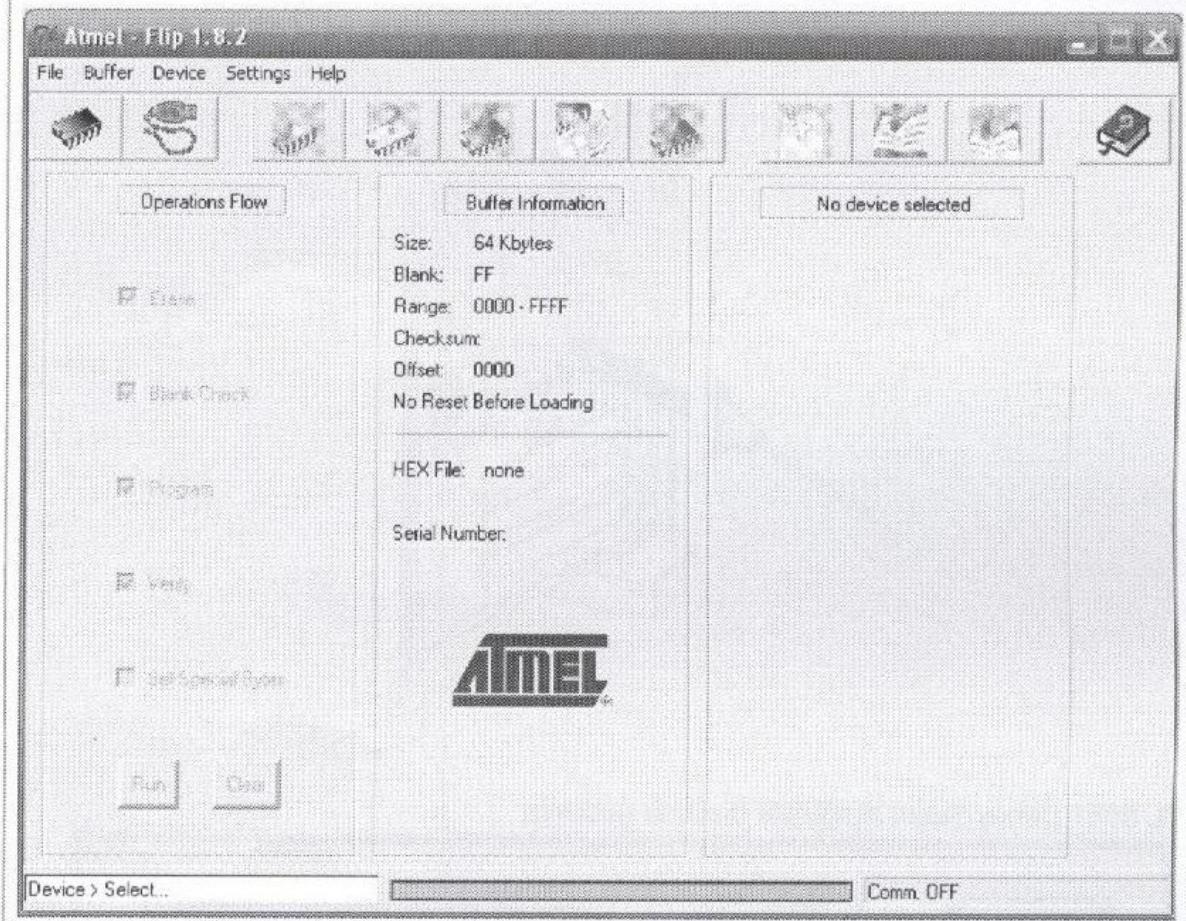
Örnek;



2. 8051 Deney Setini aşağıdaki duruma getiriniz.



Uyarı: Seri Portu programlama modülündeki RS232 Sokete (J1_Prg) takınız.

**FLIP Programı Tanımı: ATMEL FLIP (Flexible In-system Programmer)****Flip Ana Ekranı****Menu - İkonlar**

Flip programındaki tüm fonksiyonlar bu ikon menüde kontrol edilebilir. Chip seçip, haberleşmesini sağlayıp, programınızı seçip, chip e bunu yazabilir, doğrulayabilir ya da okuyabilirsiniz.

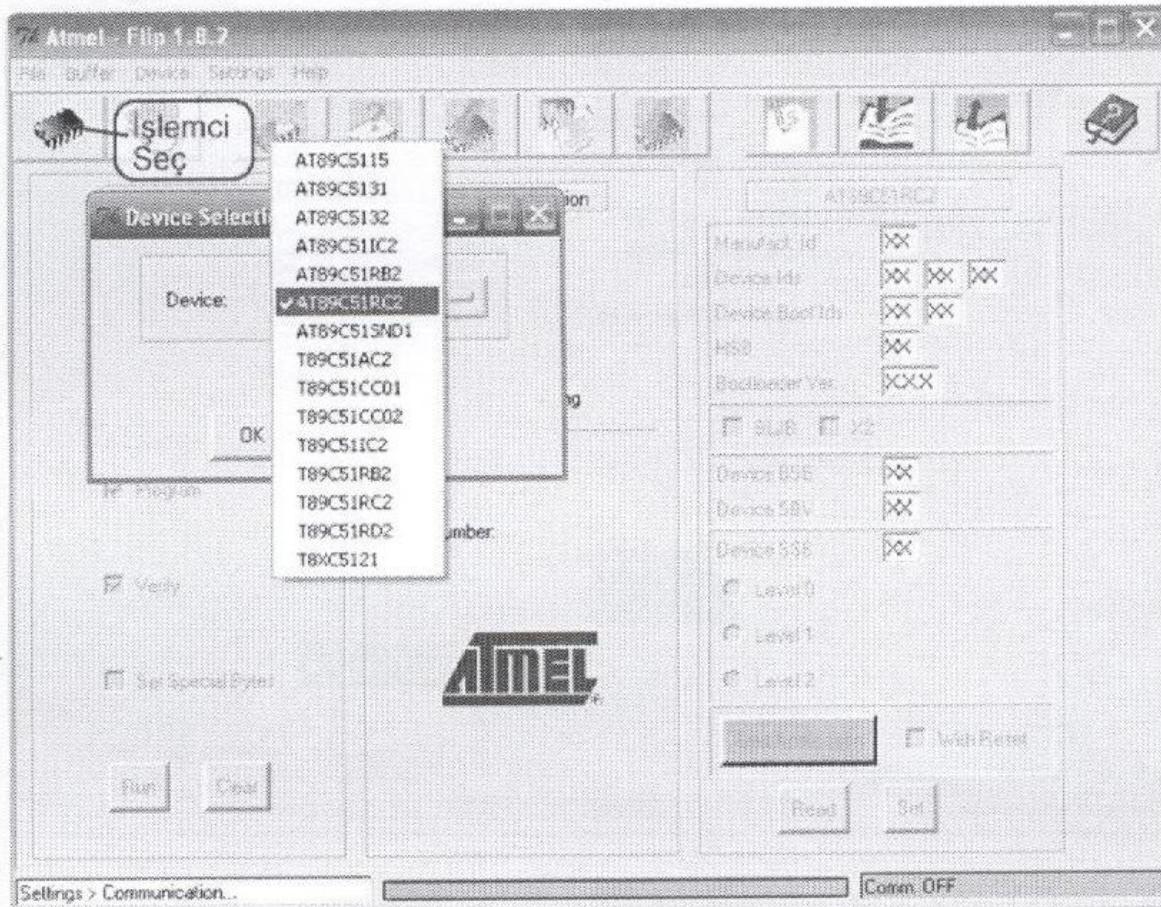
Basit yardım menüsünden de yararlanabilirsiniz.



AT89C51RC2 'nın FLIP 'le Programlanması:

ADIM - 1 (İşlemciyi Seçmek)

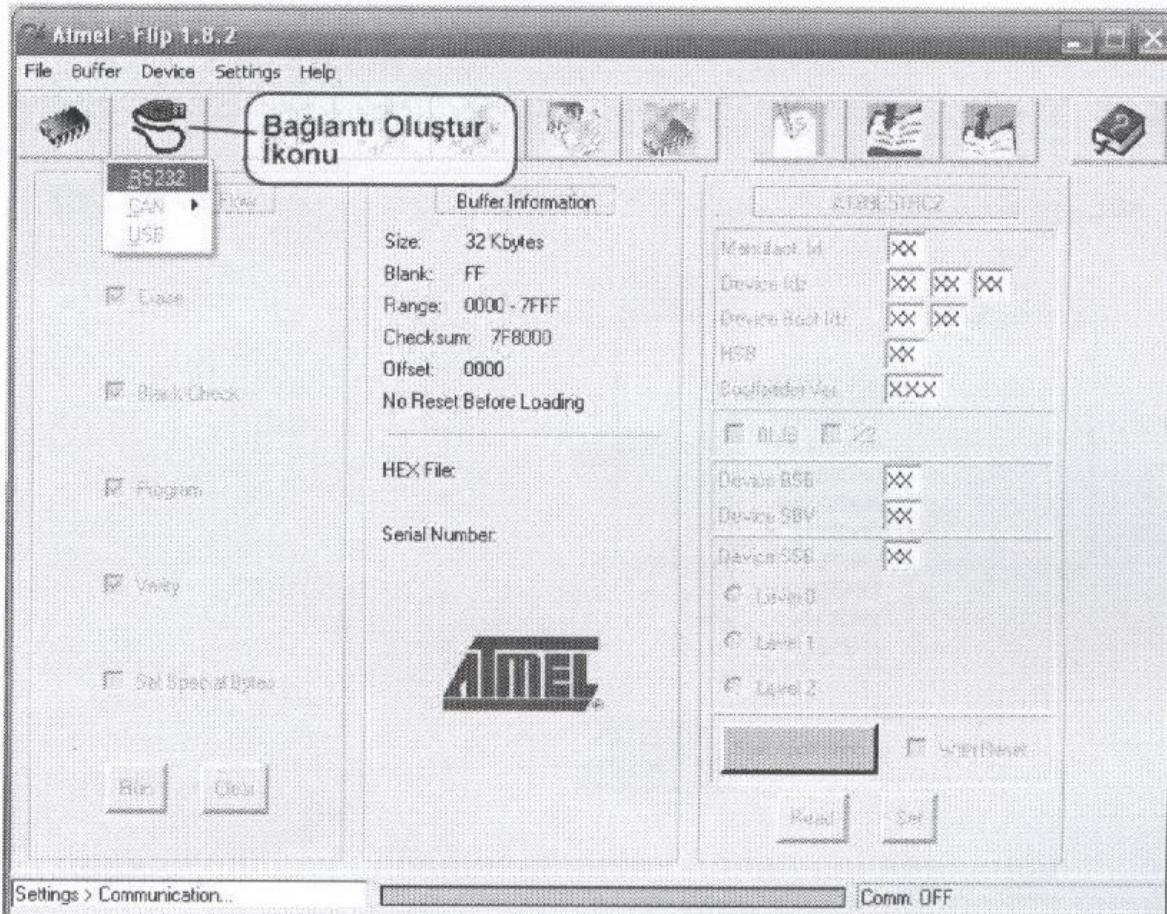
Flip Programını açılır \ İşlemci seç butonuna basılır. Daha sonra aşağıdaki şekilde kullanacağımız işlemci tanımlanır. Kullanacağımız işlemci **AT89C51RC2** dir.



Flip programı üzerinde dahili bootloader 'ı bulunan işlemcileri destekler. Yeni versiyonlarda USB, CAN arabirimleriyle de programlamaya müsaittir.

ADIM – 2 (İşlemci ile Deney Seti Haberleştmek.)

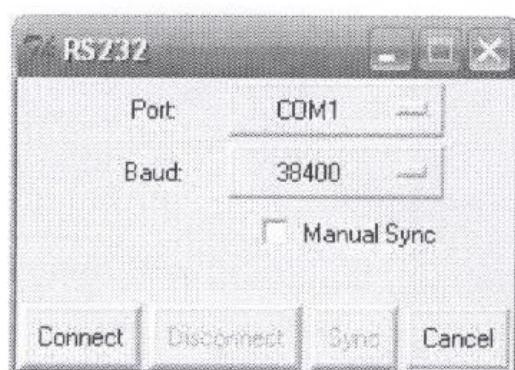
1. Arabirimden bağlantı oluştur ıkonuna tıklanır
2. Kullanılan arabirim olan RS232 seçilir.



3. RS232 ' de bağlantı kurulmak istenilen PC Seri portu seçilir. Şekilde Varsayılan "COM1" olarak görülmüyor.

4. İstenilen Baud (saniyede gelen bit hızı) seçilir.

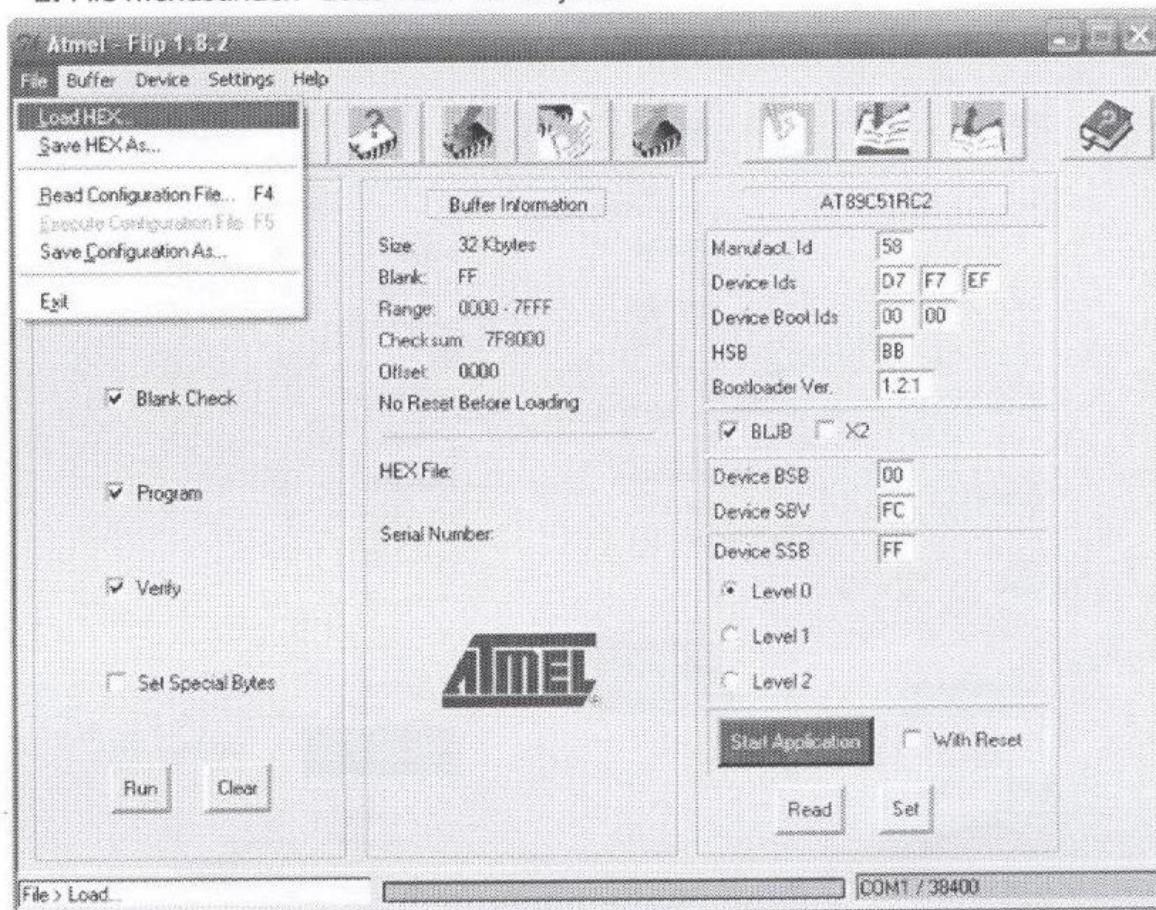
Not: Deney setleri 38400 kbps baud da başarıyla test edilmişlerdir. Sorunsuz haberleşme için düşük hızlarda baud seçilmesi önerilmektedir.



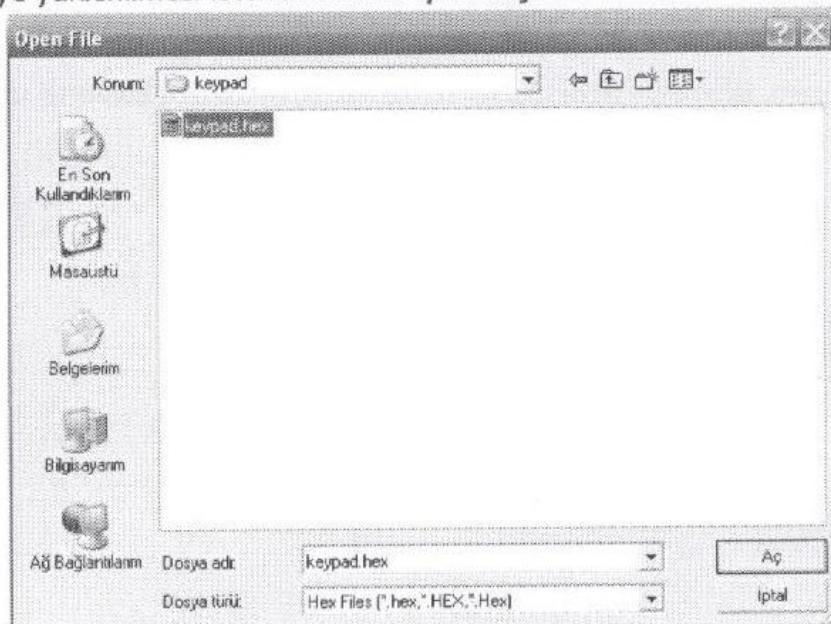
5. Connect (Bağla) butonuna basılır.
6. 8051 Deney Setindeki "RESET" butonuna basılır.

**ADIM – 3 (Üretilen HEX dosyasının seçimi)**

1. File menüsünden "Load HEX" e tıklayınız.

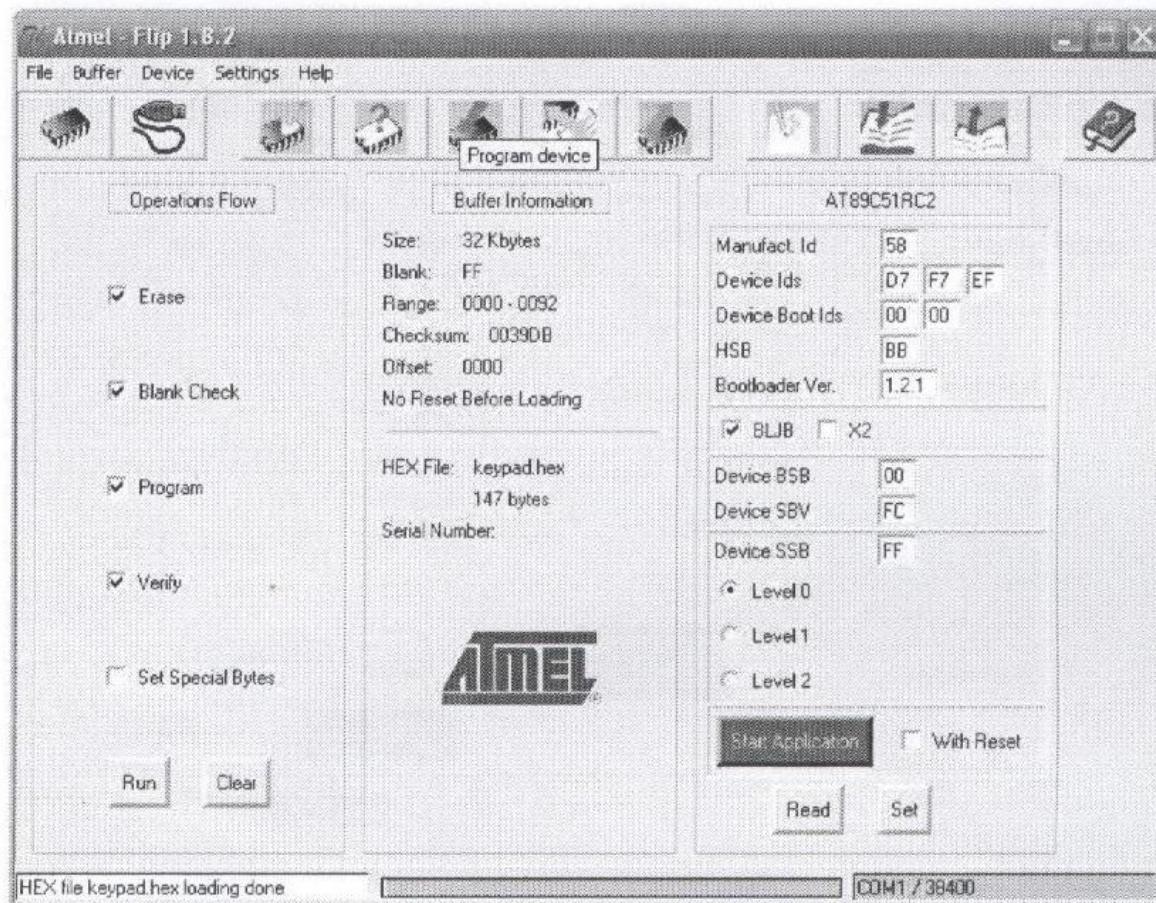


2. İşlemciye yüklenilmesi istenen hex dosyası seçiniz.





4. Seçilen hex dosyasını Programla ikonuna basınız.

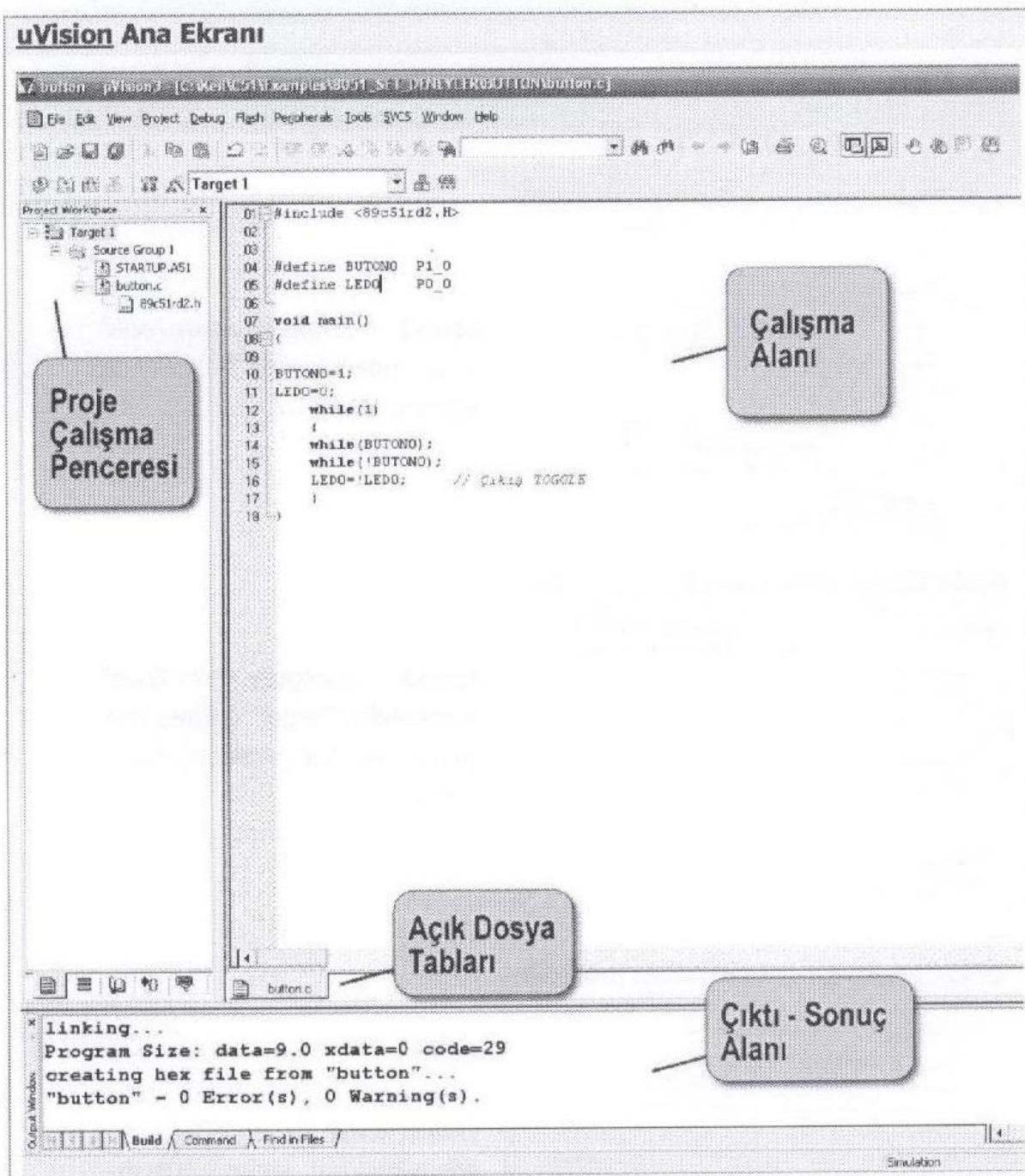


YAZILAN PROGRAM AT89C51RC İŞLEMCİMİZE AKTARILMIŞTIR.

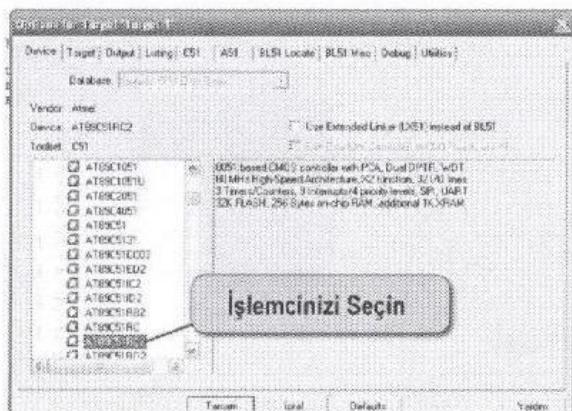
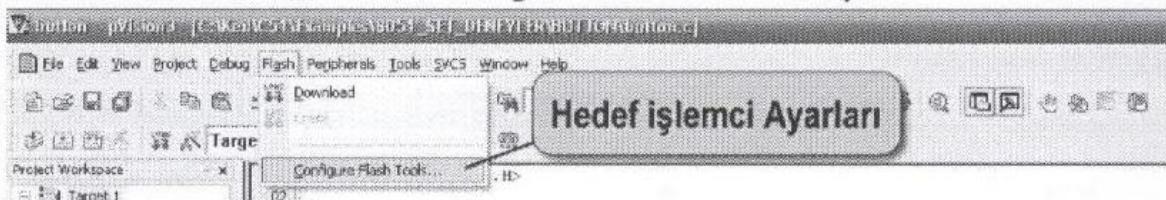
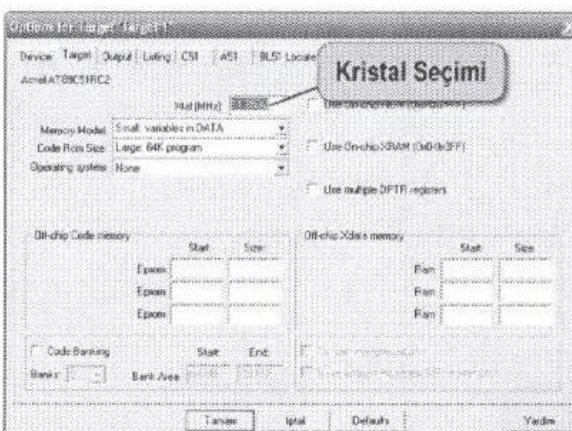
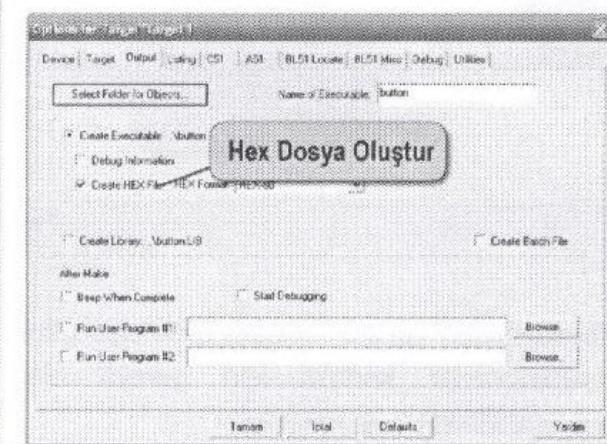
Uyarı: Programı işlemciye aktardıktan sonra Reset butonuna basmayı unutmayın.

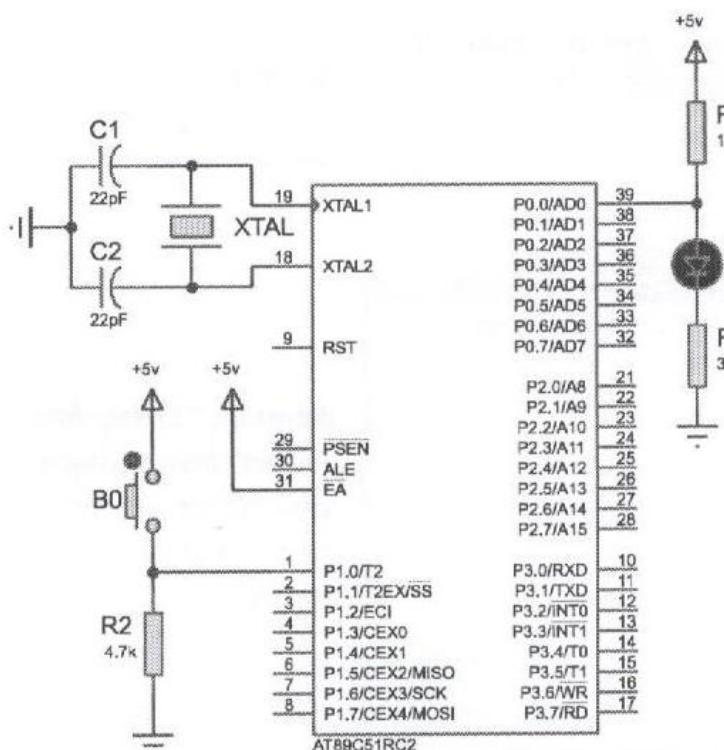
KEIL uVision Derleyicisi Tanıtımı:

Keil Elektronik GmbH firması tarafından geliştirilen bir derleyicidir.



C Derleyici, bağılayıcı, hata ayıklayıcı gibi özellikleri vardır. Seçilen işlemciye göre, kütüphane erişimi, yazılan kodla çevrebirimlerini kontrol için gerekli olan tanımlamaların kontrolü, kod dönüşümü gibi esneklikleri bulunmaktadır.

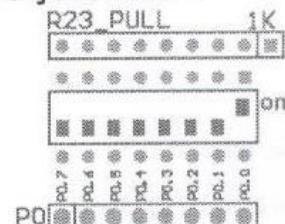
uVision 'da AT89C51RC2 işlemci tanımlama**Adım1 - "Flash" Menüsü'nden "Configure Flash Tools" u tıklayın.****Adım2 - "Configure Flash Tools" ayarlarında "Device" sekmesinden işlemcinizi seçin.****Adım3 - "Configure Flash Tools" ayarlarında "Target" sekmesinde işlemcinizle ilgili gerekli ayarları yapınız.****Adım4 - "Configure Flash Tools" ayarlarında "Output" sekmesinde, yazılan kodun derlendiğten sonra hex dönüşümü için, seçim kutusunu işaretleyin.**

KEIL uVision ile Yapılan Örnekler**Örnek 1: Buton Kontrol****Örnek 1 Hakkında**

Örneğimizde işlemcimizin P1.0 pinine bağlı olan buton ile P0.0 pinine bağlı olan LED'in kontrolü sağlanmıştır.

Program a göre LED toggle olarak çalıştırılmıştır. Butona her basıldığından led'in durumu terslenmiştir.

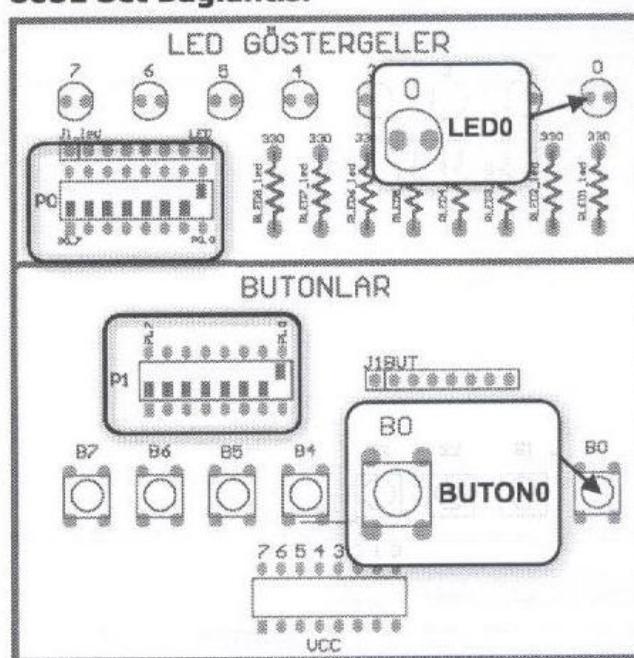
I/O modülünde P0.0 pull up a çekilmelidir.

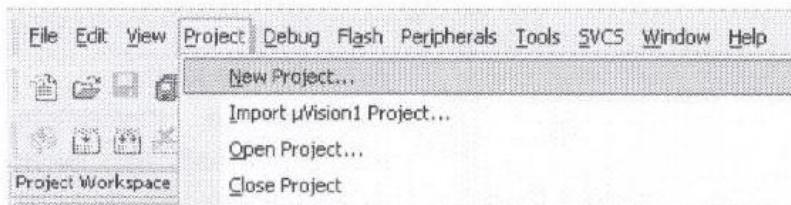
**Yazılan Kod: Buton.c**

```
#include <89c51rd2.H>

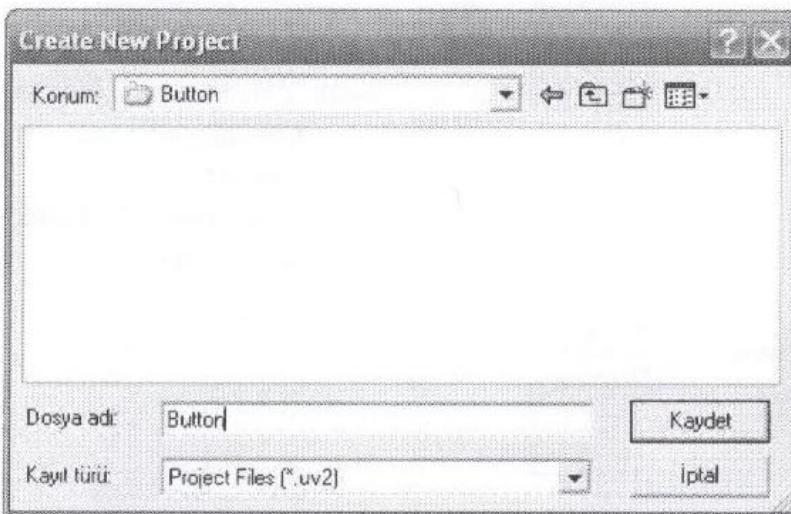
#define BUTONO P1_0
#define LEDO P0_0

void main()
{
    BUTONO=1;
    LEDO=0;
    while(1)
    {
        while(BUTONO);
        while(!BUTONO);
        LEDO=!LEDO; // çıkış TOGGLE
    }
}
```

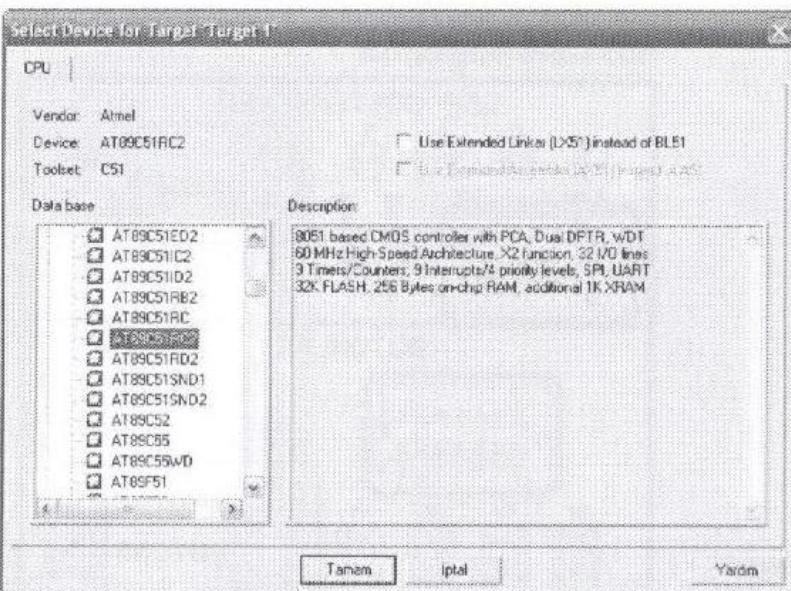
8051 Set Bağlantısı

KEIL de Proje oluşturma , Derleme

Adım 1 : "Project" menüsünden "New Project" seçilir.



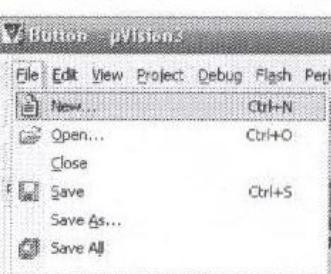
Adım 2 : "Create New Project" penceresinde yapacağınız projenin ismi ve konumu (path) belirlenerek kaydedilir.



Adım 3 :
Kullanacağınız işlemci seçilir.

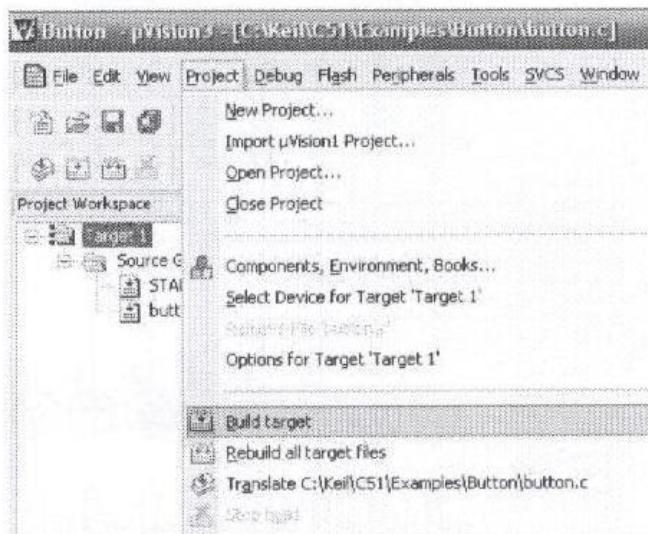
Not: Bu seçim yapıldıktan sonra program bir başlangıç dosyası oluşturacaktır.

STARTUP.ASL



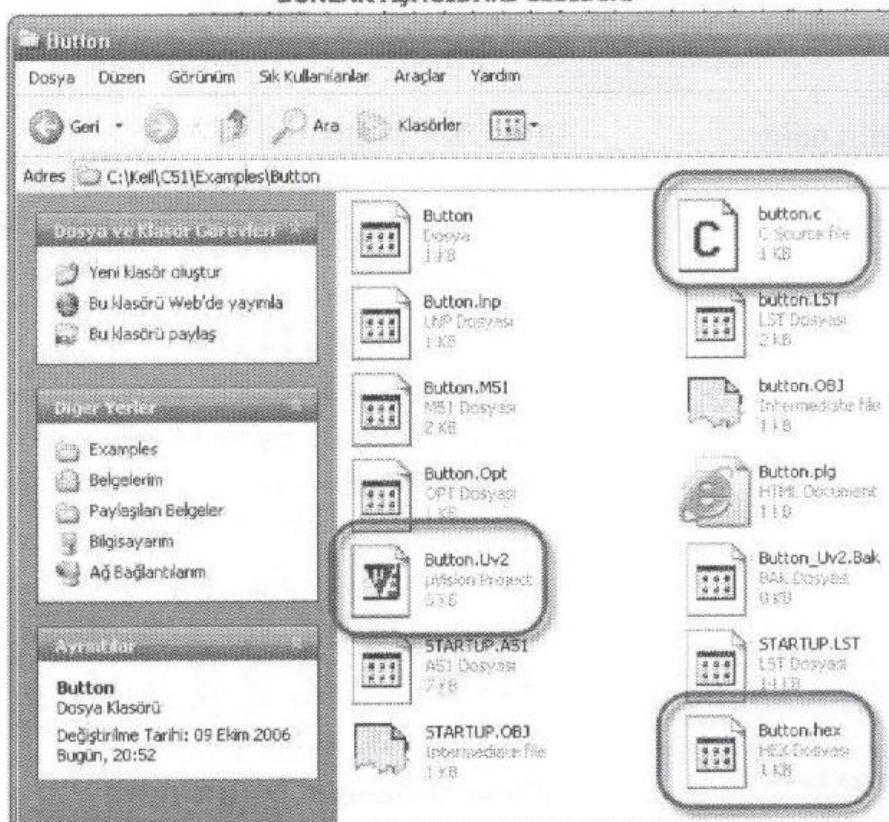
Adım 4 : "File"
Menüsünden "New" seçilerek yeni dosya oluşturulur.

Not: Oluşturduğunuz bu dosyanın ismini kayıt ederken "button.c" olarak belirleyin.

DERLEME,

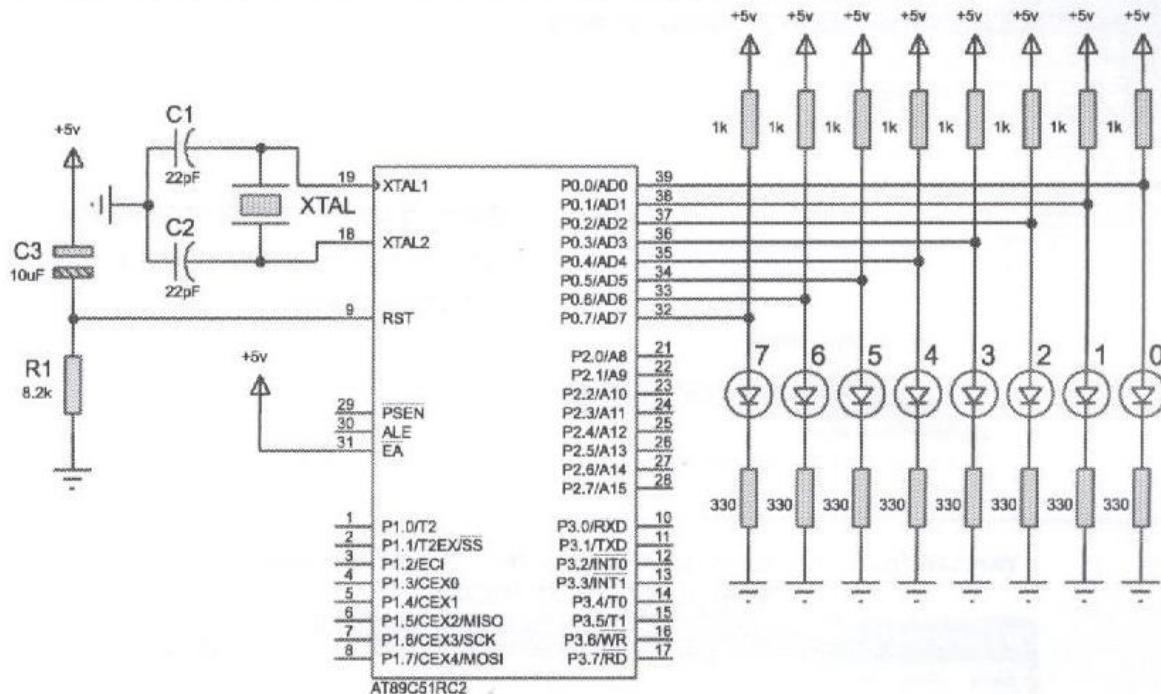
Adım 5 : Kayıt işlemi sonunda "Project" menüsünden "Build Target" a tıklanır.

**YAPILAN İŞLEMLER SONUCUNDA BİZE GEREKLİ 3 DOSYA OLUŞMUŞTUR.
BUNLAR AŞAĞIDAKİ GİBİDİR.**

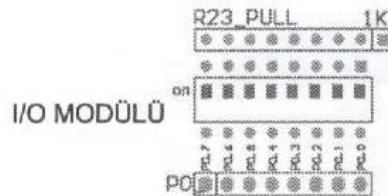


1. Button.Uv2 - Keil uVision Proje Dosyası
2. button.c - Projedeki "C" kodu
3. Button.hex - Proje sonunda oluşan ".hex" kodu (8051 e yollanabilecek dosya)

Üretilen HEX Dosyasını ATTEL FLIP Programı ile işlemciye yollayınız.

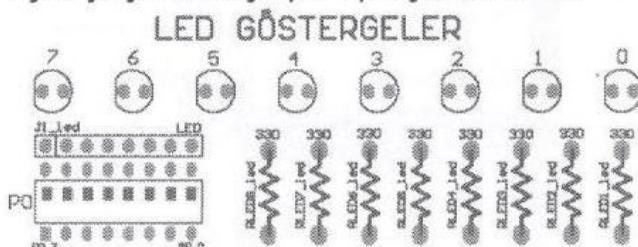
Örnek 2: Ledler**Yazılan Kod: Ledler.c**

```
#include <89c51rd2.H>
int k;
void zaman(void)
{
    int i;
    for (i=0;i<=20000;i++);
}
void main(void)
{
    P0=0x80;
    while(1)
    {
        for(k=0; k<7; k++){
            zaman();
            P0=P0>>1;
        }
        for(k=0; k<7; k++){
            zaman();
            P0=P0<<1;
        }
    }
}
```

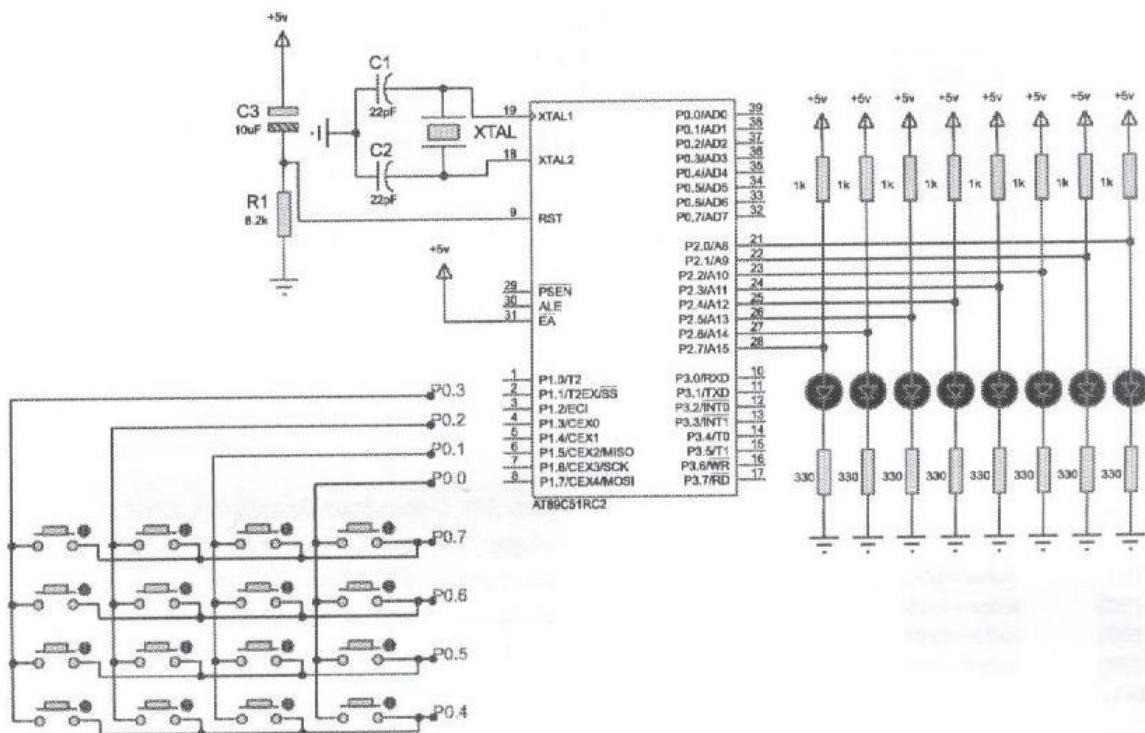
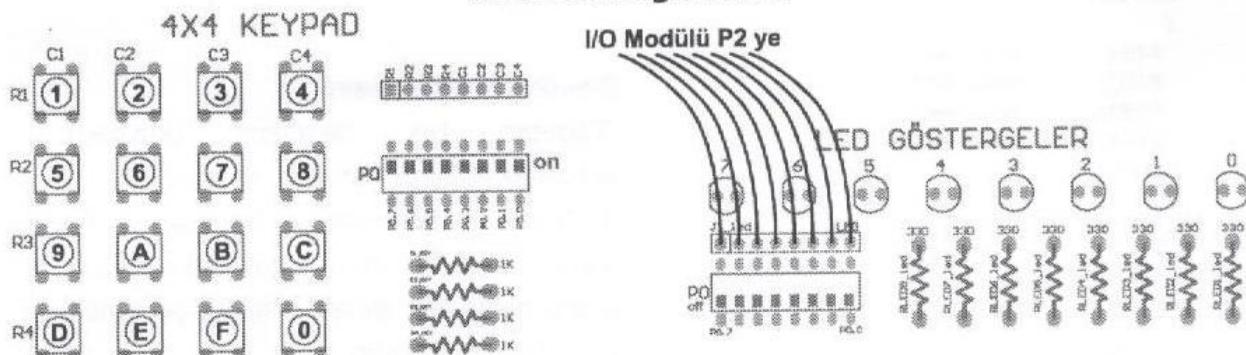
8051 Set Bağlantısı

I/O
modüllerinde
P0 dipswitch ini
“on” konumuna
alınız.

Led göstergeler modülünde de P0 dip switch ini ‘on’ konumuna alınız. Bu sayede P0 ledlere bağlanmış olup lojik1 çalıştırılmak için pull-up a çekilmektedir.

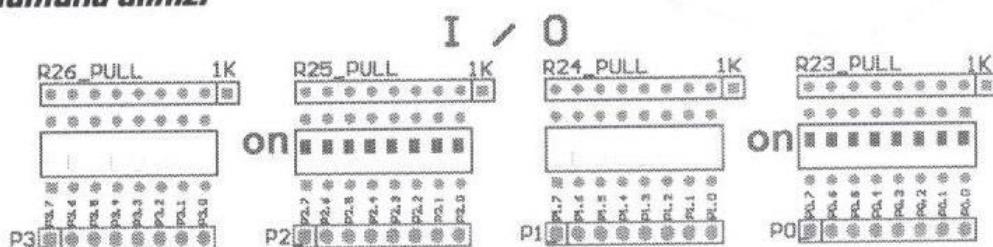
**Örnek 2 Deneyinin Çalışması**

P0 portuna bağlanılan ledler ilk anda “10000000” değeri ile başlar.(P0=0X80) Bu konum, zaman değişkeni ile bir gecikme (yaklaşık 100ms) sağlanarak sağa kaydırılmış. 7 durum sonunda ise tekrar sola kaydırılmıştır. Bu şekilde sonsuz bir döngü kurulmuştur.

Örnek 3: Keypad**8051 Set Bağlantıları**

* 4x4 Keypad modülü ve I/O modülündeki dipswichleri "on" konumuna alınız.

* Led göstergeler modülünde ise J1_Led bağlantısını jumperlarla (Zilteli) I/O modülündeki P2 portuna bağlayınız ve I/O modülündeki P2 ve PO dipswichlerini "on" konumuna alınız.



Yazılan Kod: Key.C

```
#include <89c51rd2.H>

#define C4      P0_0 //Sütun4
#define C3      P0_1 //Sütun3
#define C2      P0_2 //Sütun2
#define C1      P0_3 //Sütun1
#define R4      P0_4 //Satır4
#define R3      P0_5 //Satır3
#define R2      P0_6 //Satır2
#define R1      P0_7 //Satır1

#define ledbar  P2 //Ledler P2 Portuna Bağlı

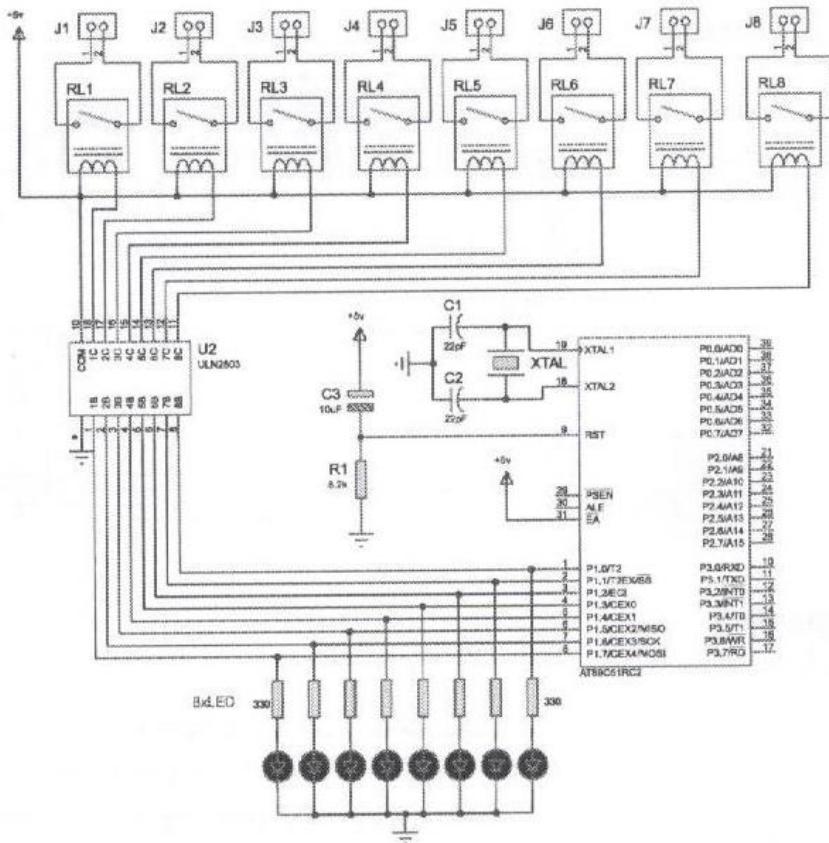
void main(void)
{
    while(1)
    {
        C1=0;
        for(;!C1;)
        {
            if (!R1)    ledbar=0x01;
            if (!R2)    ledbar=0x05;
            if (!R3)    ledbar=0x09;
            if (!R4)    ledbar=0x0D;
            C1=1;
        }
        C2=0;
        for(;!C2;)
        {
            if (!R1)    ledbar=0x02;
            if (!R2)    ledbar=0x06;
            if (!R3)    ledbar=0x0A;
            if (!R4)    ledbar=0x0E;
            C2=1;
        }
        C3=0;
        for(;!C3;)
        {
            if (!R1)    ledbar=0x03;
            if (!R2)    ledbar=0x07;
            if (!R3)    ledbar=0x0B;
            if (!R4)    ledbar=0x0F;
            C3=1;
        }
    }
}
```

```
C4=0;
for(;!C4;)
{
    if (!R1)    ledbar=0x04;
    if (!R2)    ledbar=0x08;
    if (!R3)    ledbar=0x0C;
    if (!R4)    ledbar=0x00;
    C4=1;
}
}
```

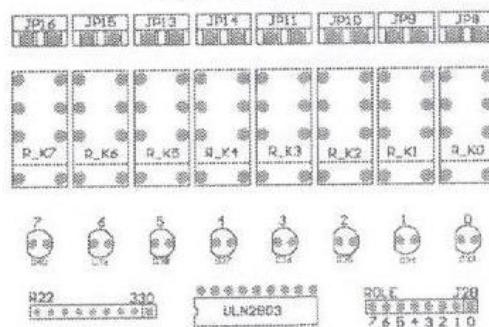
Not: Key.C dosyanızı derledikten sonra oluşan "key.hex" dosyasını ATTEL FLIP programıyla 8051 Deney Setine yollayınız. (Bknz: AT89C51RC2 'nin FLIP 'le Programlanması)

Devrenin Çalışması:

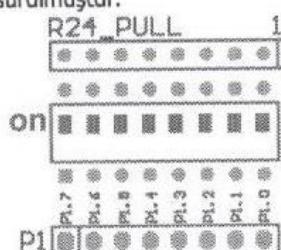
Taranan tuş takımının (keypad) sütunları P0'ın ilk 4 bitine, satırlar ise son 4 bitine bağlanmıştır. Sütunlar teker teker seçilerek her satırındaki buton bilgisi alınır. Bu alınan bilgi P2 portunda 2li sistemde görülür.

Örnek 4: Röleler**Yazılan Kod: Role.C****8051 Set Bağlantısı**

ROLE KONTROL

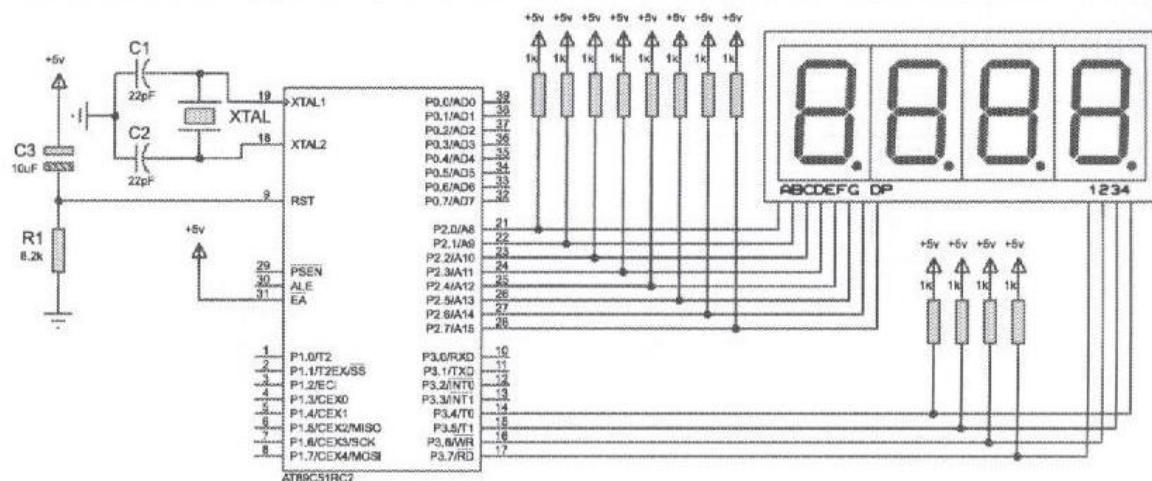
**Devrenin Çalışması:**

Röle Kontrol modülündeki ROLE (j28) soketini bağlantı kabloları ile (zil teli) P1 portuna bağlayınız. Program ledler.c programının rôle versiyonudur. Rölelerde aynı şekilde sürülmüştür.

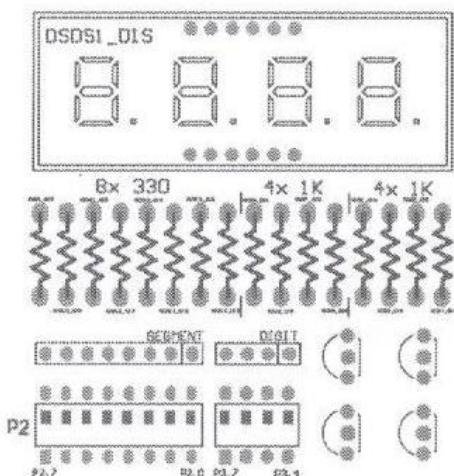


Not: I/O modülündeki P1 Portunu "on" konumuna almayı unutmayın.

Örnek 5: Matrix Display

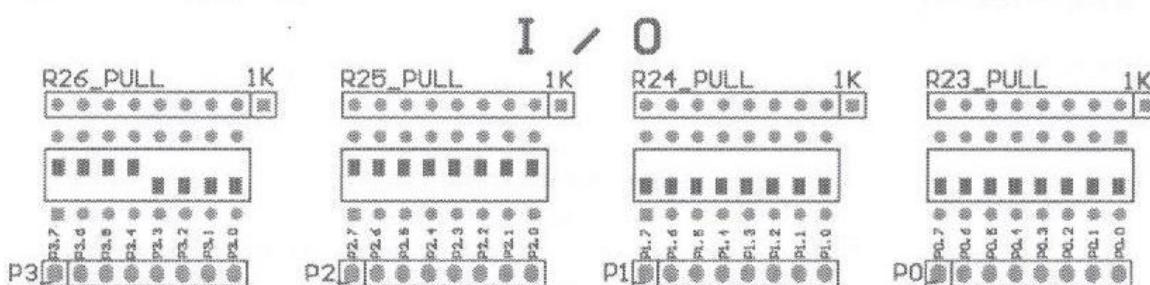


8051 Set Bağlantısı: MATRIX DISPLAY



8051 Deney Set'inde Matrix Display'in segmentleri dipswitch ile işlemcinin (AT89C51RC2) **P2 portuna statik olarak da bağlanabilmektedir. Digit seçme pinleri de istenilirse digit dipswitch i ile statik olarak **P3** MSB bitlerine (P3.7~P3.4) bağlanabilmektedir.**

Şekilde görüldüğü üzere bu dipswitchlerin konumu değiştirilerek pin-pin istenilen porta bağlanabilmektedir.



I/O Modülünde dipswitchleri şekilde görülen konuma alınız.



Yazılan Kod: Matrix.C

```
#include <89c51rd2.H>
#define segment P2          // Segment pinleri
#define SEGSEC P3           // Digit pinleri
//Digit pinleri P3.7~P3.4

void zaman(void)
{
    int i;
    for (i=0;i<=1000;i++);
}

void main(void)
{
    int sayi;
    unsigned char tablo[]={0xFC,0x60,0xDA,
0xF2,0x66,0xB6,0xBE,0xE0,0xFE,0xF6,
0xEE,0x3E,0x9C,0x7A,0x9E,0x8E};

    while(1)
    {
        sayi=1;
        segment = tablo[sayı];
        SEGSEC = 0x80;
        zaman();           //1 sn bekle
        sayı=2;
        segment=tablo[sayı];
        SEGSEC=0x40;
        zaman();           //1 sn bekle
        sayı=3;
        segment = tablo[sayı];
        SEGSEC = 0x20;
        zaman();           //1 sn bekle
        sayı=4;
        segment=tablo[sayı];
        SEGSEC=0x10;
        zaman();           //1 sn bekle
        sayı=1;
    }
}
```

Devrenin Çalışması

1. Matrix.C

Programda ilk olarak segment pinleri (P2) ve Digit pinleri seçilmiştir. "unsigned char tablo[]" ifadesinde segmentlerde karakter ifadesi oluşturmak için 8051 Deney Setimize göre bir tablo oluşturulmuştur. Bu tablo 0 ile F (16 karakter Hex) arasındaki karakterlerdir.

"while(1)" sonsuz döngümüzde ilk olarak bu tablodan istediğimiz karakterin değeri alınıp, bu digit seçerek ("SEGSEC" ile) "segment" değişkenine atılmıştır. Çok kısa zaman aralıklarında her digit e belli sayılar (1234) atanarak Matrix tarama işlemi gerçekleştirilmiştir.

1. Matrix2.C

Programda ek olarak 2 döngü hazırlanmıştır. 1.si Yavaşlama 2.si Hızlanma. "j" değişkeni döngülerde arttırılarak yavaşlama, azaltılarak da döngülerin hızlandırılması sağlanmıştır. Ayrıntılı bilgi için aşağıdaki "Matrix2.C" kodunu inceleyiniz.

Yazılan Kod: Matrix2.C

```

#include <89c51rd2.H>

#define segment P2
#define SEGSEC P3

void zaman(void)
{
    int i;
    for (i=0;i<=500;i++);
}

void main(void)
{
    int sayı;
    int k;
    int speed=1;
    int j = 0;
    unsigned char

tablo[]={0xFC,0x60,0xDA,0xF2,0x66,0xB6,0xBE,0xE0,0xFE,0xF6,0x
EE,0x3E,0x9C,0x7A,0x9E,0x8E};

while(1)
{
    ///////////////////////////////////////////////////////////////////
    if(speed==1){
        j++;                                //j değişkeni artır
        sayı=1;
        segment = tablo[sayı];
        SEGSEC = 0x80;                      // TABLODAN SAYIYI AL //
        for(k = 0; k < j; k++)              // SAYIYI YAZ //
        zaman();
        sayı=2;
        segment=tablo[sayı];
        SEGSEC=0x40;
        for(k=0; k<j; k++)
        zaman();
        sayı=3;
        segment = tablo[sayı];
        SEGSEC = 0x20;
        for(k=0; k<j; k++)
        zaman();
        sayı=4;
        segment=tablo[sayı];
        SEGSEC=0x10;
        for(k=0; k<j; k++)
        zaman();
        sayı=1;

        if(j>20) speed=0;
    }
    ///////////////////////////////////////////////////////////////////
    if(speed==0){
        j--;                                //j değişkeni azalt
        sayı=1;
        segment = tablo[sayı];
        SEGSEC = 0x80;
        for(k = 0; k < j; k++)
        zaman();

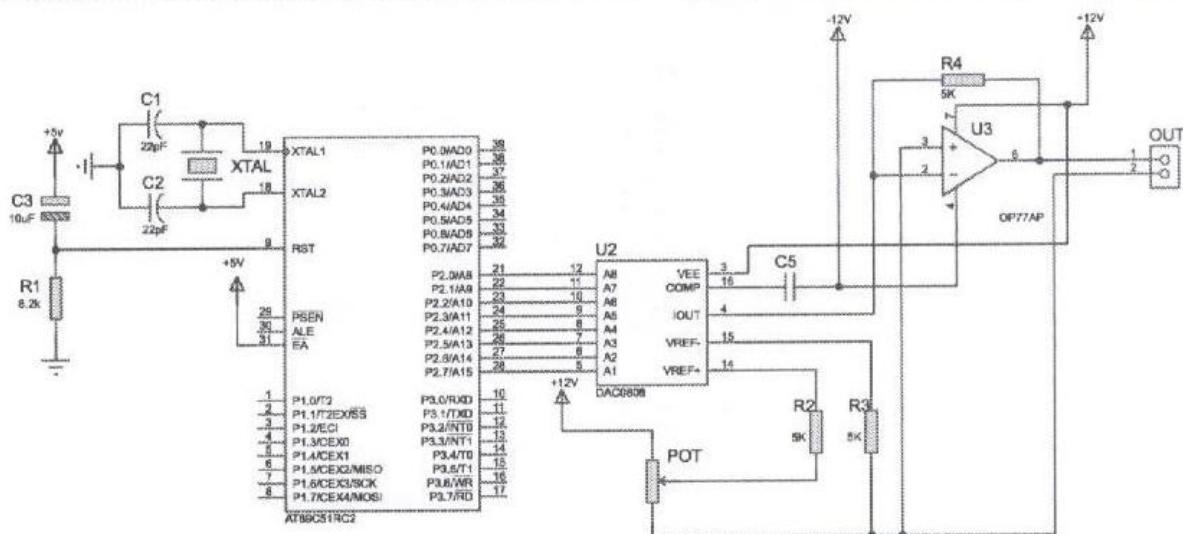
        sayı=2;
        segment=tablo[sayı];
        SEGSEC=0x40;
        for(k=0; k<j; k++)
        zaman();

        sayı=3;
        segment = tablo[sayı];
        SEGSEC = 0x20;
        for(k=0; k<j; k++)
        zaman();

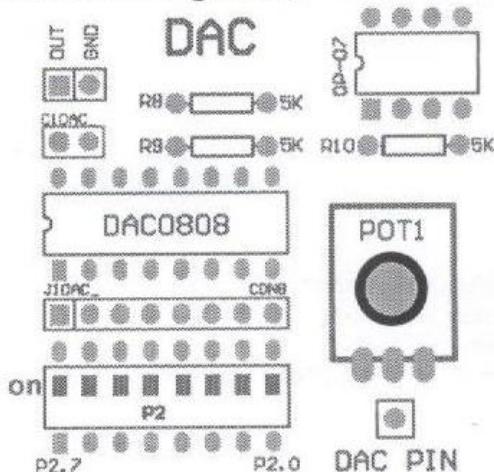
        sayı=4;
        segment=tablo[sayı];
        SEGSEC=0x10;
        for(k=0; k<j; k++)
        zaman();

        if(j<1) speed=1;                     //j
        değişkeni 0 landığında başa dön
    }
}

```

Örnek 6: DAC**Yazılan Kod: dac.c**

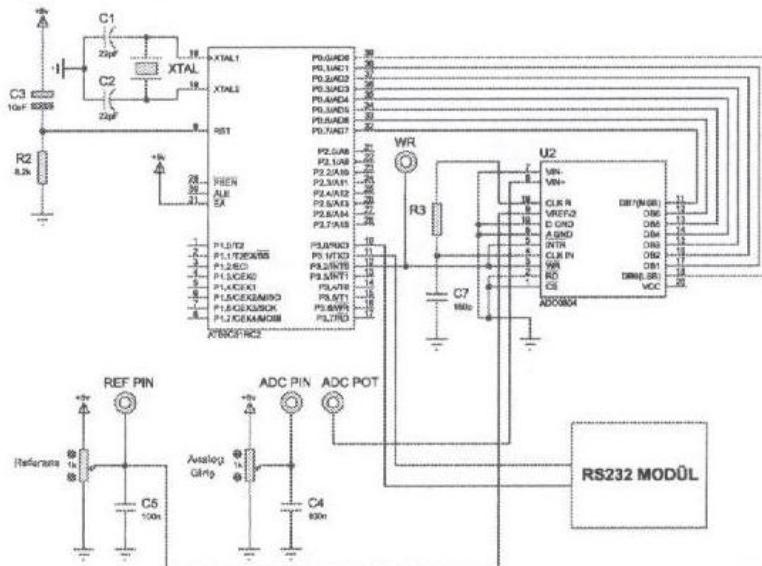
```
#include <89c51rd2.H>
#define DAC_PORT P2
void zaman(void)
{
    int i;
    for (i=0;i<=1000;i++);
}
void main(void)
{
    int sayı = 0;
    int k;
    int speed=1;
    int j = 0;
    while(1)
    {
        if(speed==1){
            j++;
            sayı++;
            DAC_PORT =sayı;
            zaman();
            if(j>254) speed=0;
        }
        if(speed==0){
            j--;
            sayı--;
            DAC_PORT=sayı;
            zaman();
            if(j<1) speed=1;
        }
    }
}
```

8051 Set Bağlantısı**Devrenin Çalışması:**

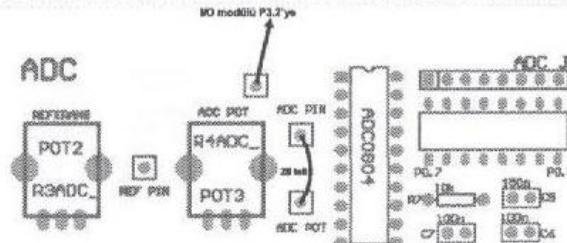
Bu uygulamada işlemcimizin P2 portunu DAC'ın dijital girişleri bağlanmaktadır (Bağlantı şeklinde P2 dipswitchi on konumuna alınmalıdır.)

Programda P2 portu 0'dan 255'e kadar dijital artar. 255'e ulaştığında tekrar 0'a doğru azalır. Bu artış yada azalışı "zaman" fonksiyonunda " $i \leq 1000$ " 1000 değerini değiştirerek DAC'ın çıkış frekansı değiştirilebilmektedir. Bu işlemin yavaş gerçekleştirilebilmesiyle voltaj değişimi görülmektedir.

Örnek 7: ADC + RS232



Yazılan Kod: adc.c



```
#include <89c51rd2.H>
#include <stdio.h>
#define VERI P0
#define WR P3_2 //ADC başlatma ucu
```

```
float Gerilim;  
float ADIM;  
int Maksimum=5;  
int orneklemme=256;
```

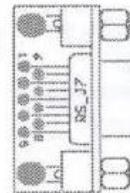
```

void main(void)
{
    SCON=0x40;
    TMOD=0x20;
    TH1=0xFD;
    TR1=1;
    TI=1;
    VERI=0xFF;
    WR=0;

    ADIM=(float)Maksimum/ornekleme;
    while(1)
    {
        WR=1;
        Gerilim=VERI*ADIM;
        printf("ADC OKU: %f\r", Gerilim);
    }
}

```

8051 Set Bağlantısı



ADC modülünde "P0" a bağlı dip switchi "on" konumuna alınız. WR pinini P3.2 ye bağlayınız.(Zilteli ile)

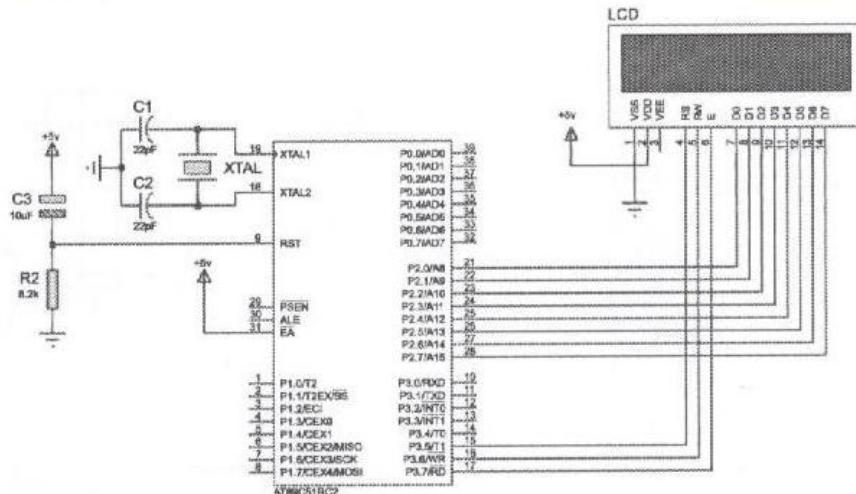
RS-232 modülündeki dip switch i "on" konumuna alınız. Hex dosyasını Atmel Flip programı ile islemeviye yükleyiniz.

Deney Seti Programlama Modülündeki "RESET" butonuna basınız.

Devrenin Çalışması:

Bu uygulamada ADC den gelen 8 bitlik bilgi (P0 portuna bağlı) işlemcimizde 256 lr örneklemeyeyle yapılarak okunabilmektedir. Her adımdaki voltaj değişimleri seri porttan gönderilebilmektedir. Programda ADC0804'ün WR pinine ilk anda 0 verilip , daha sonra 1 e çekilerek bu pinde yüksek empedans sağlanmıştır. Bu ADC 'ye start veren bir işlemmdir.

Örnek 8: LCD



Yazılık Kodu: lcd.c

```

#include <89c51rd2.H>
#include <STDIO.H>
#include <math.h>
//Zaman Gecikmesi
void gecikme(long int sure)
{
    long int i;
    for (i=1;i<sure;i++)
    {
    }
}
void datakomut(int komut) //LCD Komut
{
    P3_5=0;
    P3_6=0;
    P3_7=1;
    P2=komut;
    P3_7=0;
    gecikme(20000);
}
void dataveri(char veri[],long int hiz)
{
    int i;
    while(veri[i]!=0)
    {
        P3_5=1;
        P3_6=0;
        P3_7=1;
        P2=veri[i];
        P3_7=0;
        i++;
        gecikme(hiz);
    }
}

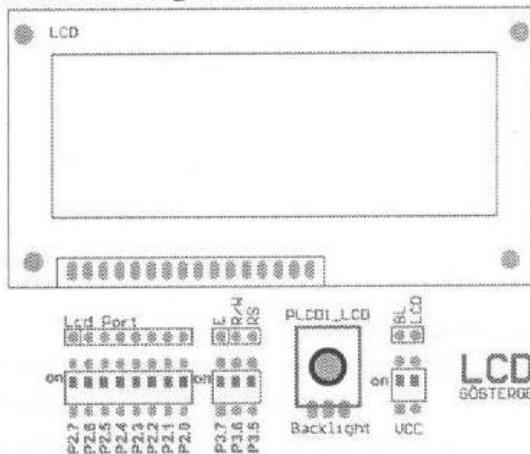
```

```

void LcdAc()//LCD'yi Aç
{
    int baslangic[]={12,52,4},t;
    for(t=0;t<3;t++)
        datakomut(baslangic[t]);
}
char dat[40];
void main (void)
{
    LcdAc();
    datakomut(1);
    sprintf(dat," TEKO ELEKTRONIK");
    dataveri(dat,1000);
    gecikme(60000);
}

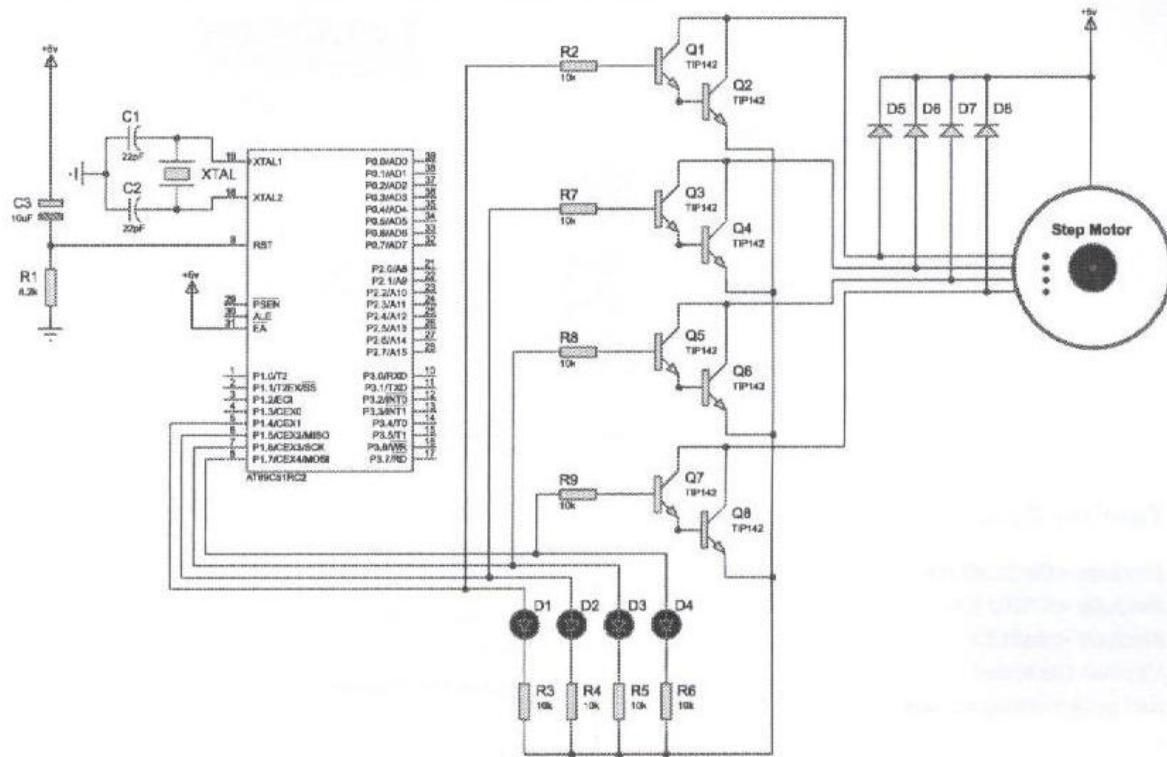
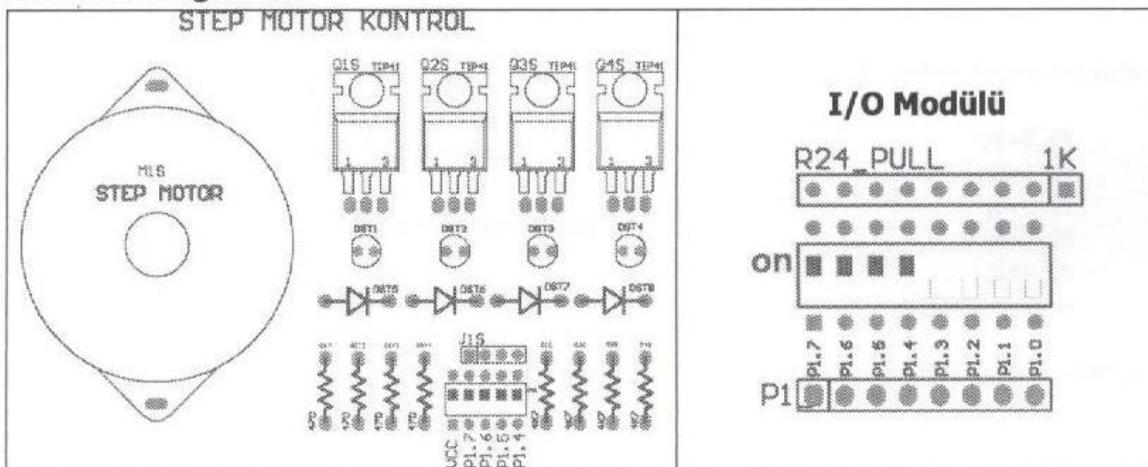
```

8051 Set Bağlantısı



LCD Gösterge modülündeki dipswitchleri "on" konumuna alın.

Lcd'ye karakter olarak tanımlanan "dat" değişkenine "TEKO ELEKTRONIK" stringi yazılmıştır. Bu "dataver" fonksiyonuyla LCD ayarlanarak ilk satıra yazılmaktadır.

Örnek 9: Step Motor**8051 Set Bağlantısı****Devrenin Çalışması:**

Step motor modülünde dipswitchleri "on" konumuna alın. Ve I/O modülündeki P1 portunu "on" konumuna alınız. İşlemcinin üst P1 Portundajı üst 4 biti (P1.7~P1.4) Step motoru süren bitlerdir. Step motoru doğru yönde çevirmek için P0 portuna 0x60,0xA0,0x90,0x50 değerleri verilmektedir. Ters yönde çevirmek için ise 0x50,0x90,0xA0,0x60 değerleri verilmektedir. Programda bu değerler sırayla porta yazılmaktadır.

Not: I/O Modülündeki P1 portunun üst 4 bitini şekilde görüldüğü gibi "on" konumuna almayı unutmayın..

**Yazılan Kod: step.c**

```
#include <89c51rd2.H>      // işlemci başlık dosyası
#define STEP P1                // 7 parçalı gösterge port tanımlaması
void zaman(void)
{
    int i;
    for (i=0;i<=20000;i++);
}
void main(void)
{
    int sayı;
    int forward = 1;
    int DirectionCounter;
    unsigned char tablo[]={0x60,0xA0,0x90,0x50};
    while(1)
    {
        if(forward==1){
            for(DirectionCounter = 0; DirectionCounter <4; DirectionCounter ++){
                sayı=0;
                STEP = tablo[sayı];
                zaman();          //1 sn bekle
                sayı=1;
                STEP=tablo[sayı];
                zaman();          //1 sn bekle
                sayı=2;
                STEP = tablo[sayı];
                zaman();          //1 sn bekle
                sayı=3;
                STEP=tablo[sayı];
                zaman();          //1 sn bekle
                sayı=3;
                forward=0;
            }
            else{
                for(DirectionCounter = 0; DirectionCounter <4; DirectionCounter ++){
                    sayı=3;
                    STEP = tablo[sayı];
                    zaman();          //1 sn bekle
                    sayı=2;
                    STEP=tablo[sayı];
                    zaman();          //1 sn bekle
                    sayı=1;
                    STEP = tablo[sayı];
                    zaman();          //1 sn bekle
                    sayı=0;
                    STEP=tablo[sayı];
                    zaman();          //1 sn bekle
                    sayı=0;
                    forward=1;
                }
            }
        }
    }
}
```