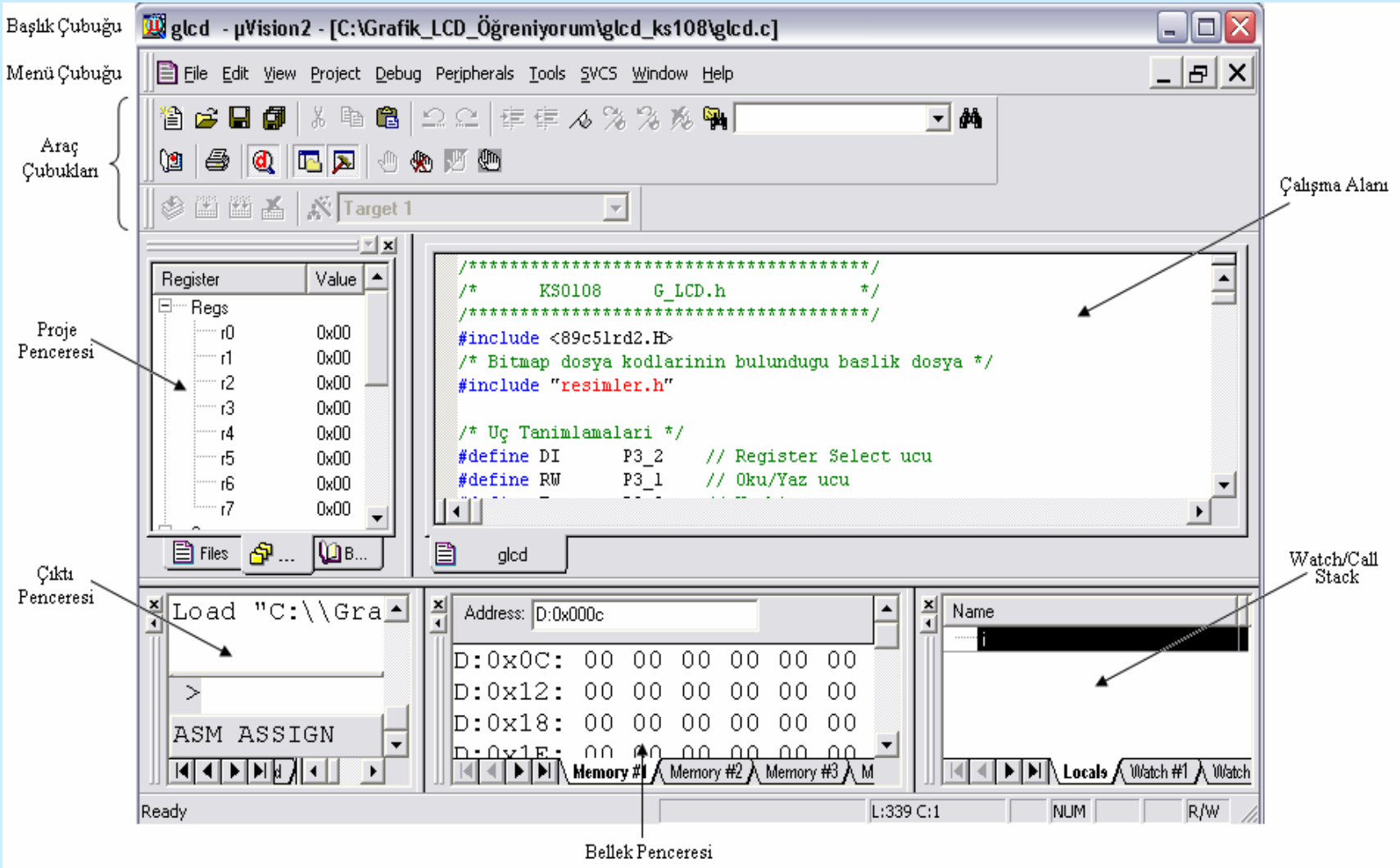
# İLERI MIKRODENETLEYICILER

Ege Üniversitesi Ege MYO
Mekatronik Programı

# EK-A

# IDE, Program Geliştirme Araçları

# Geliştirme Araçları – Keil C51 Yazılımı

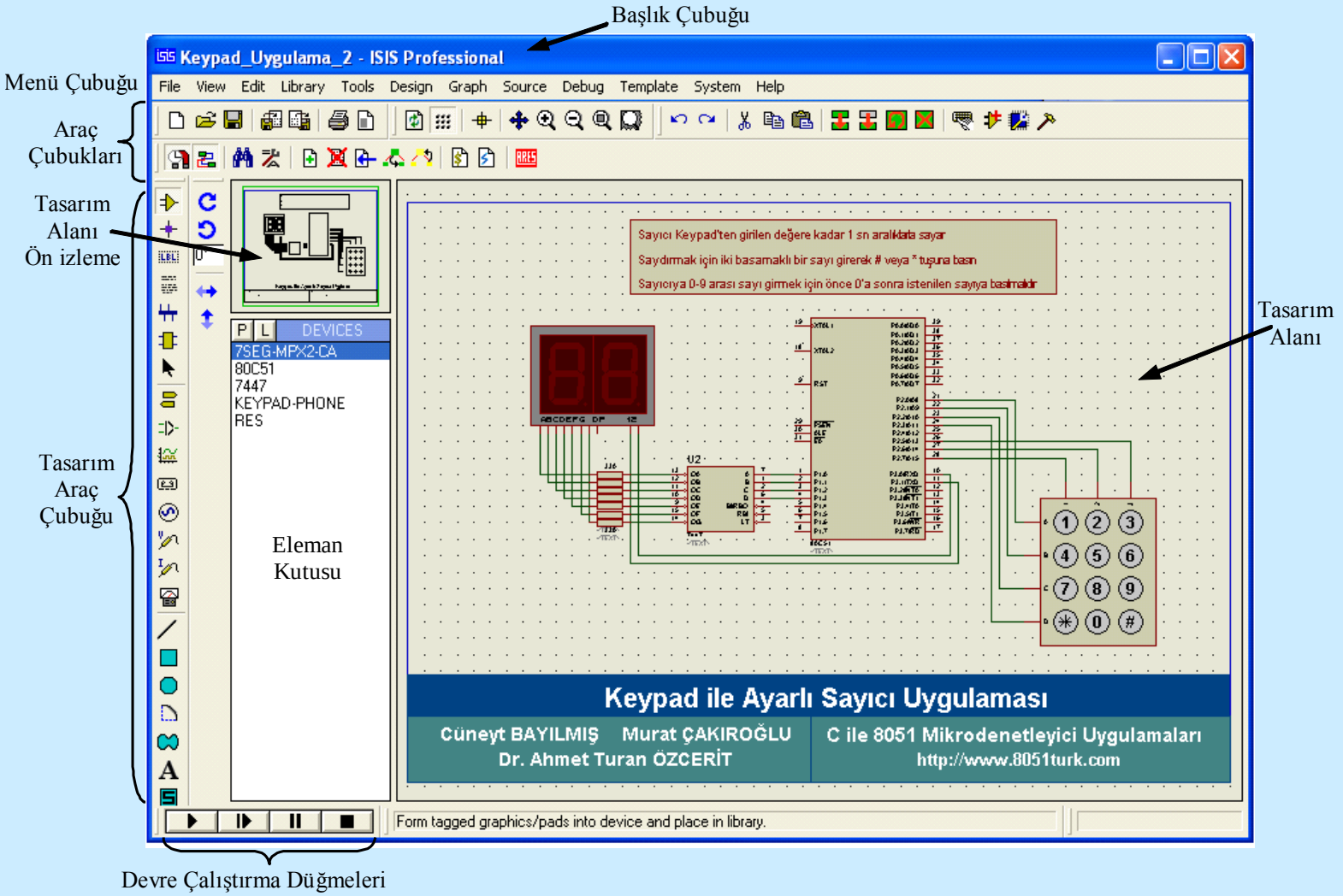


# Geliştirme Araçları – ISIS Programı

ISIS/Proteus programı:

- Hem sayısal hem de analog devreleri birlikte simüle edebilen, mikrodenetleyici tabanlı tasarımları destekleyen çok amaçlı simülasyon programıdır.
- Herhangi bir devreyi fiziksel olarak gerçeklemeksizin gerçek hayatta nasıl çalışacağını test etmeye imkan sağlamaktadır.
- Hem assembly hem de C dili desteği sağlamaktadır.
- Hata ayıklama (debug) işlemlerine imkan tanımaktadır.

# Geliştirme Araçları – ISIS Programı



# Keil C51 Derleyicisindeki Veri Tipleri

| Veri Tipi | Bit | Bayt | Değer Aralığı |
|---|---|---|---|
| Bit | 1 | | 0 to 1 |
| Signed char | 8 | 1 | -126 to +127 |
| Unsigned char | 8 | 1 | 0 to 256 |
| Enum | 16 | 2 | -32768 to +32767 |
| Signed short | 16 | 2 | -32768 to +32767 |
| Unsigned short | 16 | 2 | 0 to 65535 |
| Signed int | 16 | 2 | -32768 to +32767 |
| Unsigned int | 16 | 2 | 0 to 65535 |
| Signed long | 32 | 4 | -2147483648 to +2147483647 |
| Unsigned long | 32 | 4 | 0 to 4294967295 |
| Float | 32 | 4 | +/-1.7549E+38 to +/-3.402823E+38 |
| Sbit | 1 | | 0 to 1 |
| Sfr | 8 | 1 | 0 to 256 |
| Sfr16 | 16 | 2 | 0 to 65535 |

# Bit

- Tek bitlik (bir bit) bir değişken tanımlamak için kullanılır.
- Tanımlanan bütün *bit* değişenleri 8051 mikrodenetleyicisinin dahili hafızasında bulunan **bit adreslenebilir** bölgede tutulur.
- Bit adreslenebilir bölgenin toplam büyüklüğü 16 Bayt uzunluğundadır dolayısıyla bir uygulama içersinde toplam 128 adet bit değişkeni kullanılabilir.
- Bit veri tipi ANSI C standardında bulunmayan bir veri tipidir.

**Örnek:**

```
bit led;        /*led değişkeninin bit olarak
                 tanımlanması*/

led=0;          /*led değişkenine bir değer atılması*/
```

# signed char/unsigned char

- Bu veri tipleri, standart C dilinde de bulunmaktadır.
- 1 bayt (8 bit) uzunluğundadır.
- *signed char* veri tipinin aralığı -128 ile +127 arasındadır
- *unsigned char* veri tipinin aralığı 0 ile 255 arasındadır.

**Örnek:**

```
unsigned char veri; /* veri değişkeninin işaretsiz
                       karakter olarak tanımlanması*/

veri=0xFF;          /*veri değişkenine
                       0FFh değerinin atılması*/
```

# sbit

- SFR adres bölgesindeki her hangi bir bite erişmek için kullanılır.
- SFR hafıza bölgesinin tamamı bit adreslenebilir değildir.

**Örnek:**

```
sbit CY=PSW^7;          /* PSW kayıtçısının 7.bitinin CY
                           değişkeni olarak tanımlanması*/

sbit EA=0xAF;           /*EA değişkenin
                        tanımlanması */

sbit EA=IE^7;           /*EA değişkenin tanımlanması*/
```

# sfr

- SFR bellek bölgesinden 8 bitlik herhangi bir değişkeni tanımlamak için kullanılan veri tipidir.
- sfr veri tipi de standart C dilinde mevcut değildir.
- Yapı olarak diğer 8 bitlik veri tiplerine benzemektedir.

**Örnek:**

```
sfr P0=0x80;        /* Port-0'ın tanımlanması*/

sfr P1=0x90;        /* Port-1'in tanımlanması*/

sfr P2=0xA0;        /* Port-2'nin tanımlanması*/
```

# sfr16

- *sfr16* çoğu bakımdan *sfr* veri tipine benzemektedir.
- Aralarındaki tek fark *sfr16* veri tipi ile SFR adres bölgesinde bulunan 16 bitlik bir değişkeni tanımlayabilmenin mümkün olmasıdır.
- Bu sayede 16 bitlik bir saklayıcının düşük değerlikli bayt ile bir sonraki adres bölgesinde olan yüksek değerlikli baytı tek bir değişken olarak tanımlanabilir.

**Örnek:**

```
sfr16 T2=0xCC;    /* Düşük değerlikli baytı
                     T2L=0CCh, yüksek değerlikli baytı
                     T2H=0CDh olan Timer 2'nin sfr16
                     veri tipi ile tanımlanması */
```

# C51 Derleyicisinde Bellek Organizasyonu

| Bellek Tipi | Boyut | Adres Aralığı | İlgili Komut |
|---|---|---|---|
| Kod | 64 KB | 0000h-FFFFh | **code** |
| External Data (Harici) | 64 KB | 0000h-FFFFh | **xdata** |
| External Data (paged) | 256 B | 00h-FFh | **pdata** |
| Dahili Data (lower RAM) | 128 Byte | 00h-7Fh | **data** |
| Dahili Data (indirect) upper | 256 Byte | 00h-FFh | **idata** |
| Dahili RAM Bit adreslenebilir alan | 128 Bit (16 byte) | 20h-2Fh | **bdata** |

# Belirli Bellekte Değişken Tanımlama

[işaret] [data tipi] [bellek tipi] [değişken adı]

- Program (Kod) belleğinde değişken tanımlama
- **Örnek:**

```
code   int   sayi=55;   /*   sayi   değişkeninin   program
   belleğinde tanımlanması ve 55 değerinin yüklenmesi */

char code dizi[6]={0xFF,0xAA,0x56,0x87,0x45,0x89};
/* Program belleğinde dizi tanımlama */
```

# Belirli Bellekte Değişken Tanımlama

[işaret] [data tipi] [bellek tipi] [değişken adı]

- **Dahili veri belleğinde değişken tanımlama**

**Örnek:**

```
char data oku; /* ilk 128 baytlık dahili veri
                belleğinde oku değişkeninin tanımlanması */

int idata sicaklik;   /* sicaklik değişkeninin ikinci
                         128 baytlık dahili veri
                       belleğinde(upper RAM)tanımlanması */

int bdata bayrak;   /* bayrak değişkeninin bit
                     adreslenebilir tamsayı olarak tanımlanması */

char bdata bit_dizi[5]; /* bit adreslenebilir dizi */
```

# Belirli Bellekte Değişken Tanımlama

[işaret] [data tipi] [bellek tipi] [değişken adı]

- Harici veri belleğinde değişken tanımlama

**Örnek:**

```
unsigned char xdata veri;   /* veri değişkeninin
                               harici RAM belleğinde
                               (64 KB'lık bir alanda
                          her hangi bir yerde) tanımlanması */

int pdata yaz;          /* yaz değişkeninin harici RAM
                           belleğinde (256 bayt'lık bir
                           alanda her hangi bir yerde)
                           tanımlanması */
```

# C51 Derleyicisindeki Bellek Modelleri

- ***SMALL*** bellek modeli:
  - Tüm değişkenler standart olarak dahili veri belleğinde saklanır (Tüm değişkenlerin *data* tanımlaması kullanılarak ifade edilmesi gibi).
  - Mümkün olan en az kod üretilir ve en etkin kullanım sağlanır.

- ***COMPACT*** bellek modeli:
  - Tüm değişkenler harici veri belleğinin bir sayfasında tanımlanmaktadır (*pdata* ile)

- ***LARGE*** bellek modeli:
  - Tüm değişkenler, harici veri belleğinin 64 KBayt'lık her hangi bir bölgesinde tanımlanmaktadır (*xdata* ile)

# C51 Derleyicisinde Kesme Rutinlerinin Tanımlanması

| Kesme Kaynağı | Kesme Adresi | C51'deki Kesme Numarası |
|---|---|---|
| Harici Kesme 0 | 0003h | 0 |
| Zamanlayıcı / Sayıcı 0 | 000Bh | 1 |
| Harici Kesme 1 | 0013h | 2 |
| Zamanlayıcı / Sayıcı 1 | 001Bh | 3 |
| Seri Haberleşme | 0023h | 4 |

**void** Fonksiyon_Adı (void) **interrupt** kesme_no [**using** depo no]
{
  Kesme oluştuğunda yapılması gereken işlemler
}

**Örnek:**

```
void Kesme_T0 () interrupt 1
{
  Kesme oluştuğunda yapılması gereken işlemler
}
```

# Mikrodenetleyici Tabanlı C Program Yapısı

```
#include <89v51rd2.H>
```
kütüphane dosyalarının yazılması

```
#define ...
```

```
sfr ...
sbit ...
int…
```
Değişkenlerin tanımlanması

```
void kesme_0() interrupt 0
{
  ……
{
```
Kesme rutinlerinin tanımlanması

```
void alt_program()
{
…
}
```
Alt programların yazılması

```
main()
{
……
}
```
Ana programın yazılması

# Örnek 1 Buton ile LED denetimi

```
#include <REG52.H>
#include <stdio.h>
sbit buton = P2^0;                    //Butonu'i P2.0'a bağla
sbit led = P1^0;                      //LED'i P1.0'a bağla
void main(void)
{
led=0;
        while(1)
        {       while(!buton);
                while(buton);
                led=!led;
        }
}
```

# Örnek 2

```
/*Bu program bir LED'i bir saniye yakar ve söndürür*/

#include <reg52.h>                      // SFR tanımlamaları
#include <stdio.h>

#define SYSCLK    12000000              // SYSCLK frequency in Hz
sbit led = P1^0;                        //LED'i P1.0'a bağla

void msec(unsigned int Delay)           // 1 msaniye geçiktirir.
{    int i, j;
   for(j=0; j<Delay; j++)
   {
     for (i=0; i<500; i++);
    }
}
void main (void)                        // ana program
{     while (1)
     {  led=!led;
       msec (1000);
     }
}
```