//
//

//**Look up Table Tabanlı Sensor Doğrusallaştırılması**

```
#pragma config[1] = 0xF1 // Osilatör: XT
#pragma config[2] = 0xFE & 0xF9 // PWRT açık, BOR kapalı
#pragma config[3] = 0xFE // Watchdog Timer kapalı

void ayarlar();
void bekle(unsigned long t);  // t milisaniye gecikme sağlayan fonksiyon tanımı

void main()
{
        unsigned static const int lookuptable[11] = {1,2,3,4,5,6,7,8,9,10,11};
   unsigned int deger;
   //y= 2 * X + 1 sensor dogrulastirma islemi

   ayarlar();
   PORTC=0;
   anadongu:

        bekle(1);
        GO=1;
        while(GO);
        deger=ADRESH; //okunan analog degerin digital karsiligi, deger degiskenine atılıyor

  //PORTC=deger;
   if(deger == 0) //0V dogrulastirmasi
     PORTC=lookuptable[0];
   else if(deger == 25) //0.5V dogrulastirmasi
     PORTC=lookuptable[1];
   else if(deger == 51)//1V dogrulastirmasi
     PORTC=lookuptable[2];
   else if(deger == 76)//1.5V dogrulastirmasi
     PORTC=lookuptable[3];
   else if(deger == 102)//2V dogrulastirmasi
     PORTC=lookuptable[4];
   else if(deger == 127)//2.5V dogrulastirmasi
     PORTC=lookuptable[5];
   else if(deger == 153)//3V dogrulastirmasi
     PORTC=lookuptable[6];
   else if(deger == 179)//3.5V dogrulastirmasi
     PORTC=lookuptable[7];
   else if(deger == 204)//4V dogrulastirmasi
     PORTC=lookuptable[8];
   else if(deger == 230)//4.5V dogrulastirmasi
     PORTC=lookuptable[9];
   else if(deger == 255)//5V dogrulastirmasi
     PORTC=lookuptable[10];
```

```
goto anadongu;

}

////////////////////////////////////////////////////////////////////////////////////////////

void bekle(unsigned long t)   //t milisaniye gecikme sağlar
{
        unsigned x;

        for(;t>0;t--)
                for(x=140;x>0;x--)
                        nop();
}

////////////////////////////////////////////////////////////////////////////////////////////

void ayarlar()  // Bütün başlangıç ayarlarının tamamlandığı kısım
{
        GIE=0;                          // Bütün kesmeleri kapat
        TRISA=0xFF;
        TRISB=0xFF;
        TRISC=0x00;
        TRISD=0x00;
        TRISE=0xFF;

        PORTC=0x00;
        PORTD=0x00;

        ADCON0=0b.0100.0001;;
        ADCON1=0b.0000.0000;;
}
```

//
//

// **Bir DAC entegresi kullanarak sinüsoidal dalganın üretilmesi**

```
#pragma config[1] = 0xF1 // Osilatör: XT
#pragma config[2] = 0xFE & 0xF9 // PWRT açık, BOR kapalı
#pragma config[3] = 0xFE // Watchdog Timer kapalı

void ayarlar(); //PORT I/O ve Interrupt Ayarlamalarının Yapıldığı Fonksiyon
void bekle(unsigned long t);  // t milisaniye gecikme sağlayan fonksiyon tanımı
```

```
void main()
{

	ayarlar();

//-----------------------------------------------
anadongu:

	unsigned x; //For Döngülerinde Kullanılan Değişken
	static const unsigned y[10] =
{0x80,0x95,0xAD,0xC1,0xD3,0xE2,0xEF,0xF8,0xFD,0xFF}; //Sinüzoidal dalga üretmek
için kullanılan hexadecimel değerleri
	static const unsigned z[9] = {0x6A,0x54,0x40,0x2E,0x1E,0x11,0x08,0x02,0x00};
//tutan diziler...
	bekle(1);	// Acquisition Time(Sample & Hold kapasitörünün şarj olması için
//gerekli zaman)
	while(PORTB.0 == 0) //B portunun,0.bitine bağlı butona basıldığı sürece, sinyal
//üretilecek...
	{
		PORTC.0= 1 ; //Butona basıldığını belirten LED
		for(x=0;x<=9;x++)
		{
			PORTD = y[x];
			bekle(50); //Sinyalin daha net gözlemlenebilmesi için, ilgili değer
//PORT'a aktarıldıktan sonra, belirli bir süre beklenmektedir.
			PORTC.1 = 1; //For döngüleri içerisindeki bu satırlar,TRACE amacıyla
//konulmuş; dışarıdan, kodun hangi aşamada olduğunun
					//anlaşılması için kullanılmaktadır.
		}
		PORTC.1 = 0; //Döngü içerisinde yakılan LED,ilgili döngü sonlanınca,
//söndürülmektedir.
		for(x=9;x>0;x--)
		{
			PORTD = y[x];
			bekle(50);
			PORTC.2 = 1;
		}
		PORTC.2 = 0;
		for(x=0;x<=8;x++)
		{

			PORTD = z[x];
			bekle(50);
			PORTC.3 = 1;
		}
		PORTC.3 = 0;
		for(x=8;x>0;x--)
		{
			PORTD = z[x];
```

```
                bekle(50);
                PORTC.4 = 1;
            }
            PORTC.4 = 0;
        }
        PORTC.0 = 0;//Eğer sinyal üretme butonu bırakılmışsa, ilgili LED söndürülmekte ve
//kullanıcıya görsel bilgi verilmektedir.

        goto anadongu;
//---------------------------------------------
}

////////////////////////////////////////////////////////////////////////////////////////////

void ayarlar()  // Bütün başlangıç ayarlarının tamamlandığı kısım
{
        GIE = 0;            // Bütün kesmeleri kapat.
        TRISB = 0xFF;       // B portu giriş yapıldı.
        TRISC = 0;          // C portu çıkış yapıldı.
        TRISD = 0;          // D portu çıkış yapıldı.

        PORTC = 0;          // C portu çıkışları sıfırlandı.
        PORTD = 0;          // D portu çıkışları sıfırlandı.



}

////////////////////////////////////////////////////////////////////////////////////////////

void bekle(unsigned long t)   //t milisaniye gecikme sağlar
{
        unsigned x;

        for(;t>0;t--)
            for(x=140;x>0;x--)
                nop();
}


////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// Normalde High, basılınca Low olan bir butonu polling yöntemile algılanması, register
//tabanlı gecikme eklenmesi( pres delay and relase delay) ve bir LED'in yanması

#pragma chip PIC18f452

#pragma config[1] = 0xF1 // Osilatör: XT
#pragma config[2] = 0xFE & 0xF9 // PWRT açık, BOR kapalı
#pragma config[3] = 0xFE // Watchdog Timer kapalı
```

```
void ayarlar();
void bekle(unsigned long t);  // t milisaniye gecikme sağlayan fonksiyon tanımı
void kesme();

//-----------------------------------------------
void main()
{
        ayarlar();

anadongu:
        if (PORTB.0==0)
        {
        bekle(5000);
        PORTD.0=1;
 }
goto anadongu;
}
//-----------------------------------------------

////////////////////////////////////////////////////////////////////////////////////////////

void bekle(unsigned long t)   //t milisaniye gecikme sağlar
{
        unsigned x;
        for(;t>0;t--)
                for(x=100;x>0;x--)
                        nop();
}

////////////////////////////////////////////////////////////////////////////////////////////

void ayarlar()  // Bütün başlangıç ayarlarının tamamlandığı kısım
{

        TRISB=0xFF;
        TRISD=0x00;

        PORTD=0x00;

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Matris Klavyeden Veri Alarak Led Yakma

#pragma config[1] = 0xF1 // Osilatör: XT
#pragma config[2] = 0xFE & 0xF9 // PWRT açık, BOR kapalı
#pragma config[3] = 0xFE // Watchdog Timer kapalı
```

```
#pragma origin 0x8    //Aşağıdaki kesme fonksiyonunun hangi program satırından başlayacağı ayarlandı
                                        //(0x8 adresi yüksek öncelikli kesme başlangıç adresidir)
#pragma interruptSaveCheck n
void ayarlar();
void bekle(unsigned long t);  // t milisaniye gecikme sağlayan fonksiyon tanımı
void oku_yaz();


interrupt int_server(void)  // KESME SUNUCU FONKSİYONU
{
        if(INT0IF)                              //Gelen kesme, INT0 kesmesi mi?
        {
                oku_yaz();
                INT0IF = 0;
        }
}


void main()
{
        ayarlar();

//-----------------------------------------------
anadongu:

        bekle(1);       // Acquisition Time(Sample & Hold kapasitörünün şarj olması için gerekli zaman)
        goto anadongu;
//-----------------------------------------------
}

////////////////////////////////////////////////////////////////////////////////////////////

void ayarlar()  // Bütün başlangıç ayarlarının tamamlandığı kısım
{

        INT0IE=1;               // INT0 kesmesi açık
        INTEDG0=0;              // INT0 kesmesi düşen kenarda aktif olacak
        GIE=1;                          // Bütün kesmeler kullanılabilir
        TRISD=0x0F;             // PORTD giriş yapıldı(Buton)
        TRISC=0x00;             // PORTC çıkış yapıldı(LED)
        PORTC=0;                // PORTC çıkışları sıfırlandı
        PORTD=0;

}

////////////////////////////////////////////////////////////////////////////////////////////

void bekle(unsigned long t)   //t milisaniye gecikme sağlar
```

```
{
        unsigned x;

        for(;t>0;t--)
                for(x=140;x>0;x--)
                        nop();
}

////////////////////////////////////////////////////////////////////////////////////////////

void oku_yaz()
{
        unsigned x;
        int f0,f1,f2,f3,index;
        int b[4];
        static const v[8] = {0x01,0x02,0x03,0x00,0x04,0x05,0x06,0x00};
        static const n[8] = {0x07,0x08,0x09,0x00,0x0F,0x00,0x0F,0x00};

        b[0] = PORTD.0;
        b[1] = PORTD.1;
        b[2] = PORTD.2;
        b[3]  =PORTD.3;

        index = b[0] + (b[1]*2);
        index += (b[2]*4);
        index += (b[3]*8); // bu 3 satır tek bi satırda yapılabilirdi ancak compiler "syntax
hatası yok ancak yapılamıyor" uyarısı verdi.

        if( index>6 )
                PORTC = n[index-8];
        else
                PORTC = v[index];

}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//Tarama yöntemi kullanılarak matris klavyeden veri alınması ve 7 parçalı göstergeye
//aktarılması


#pragma config[1] = 0xF1 // Osilatör: XT
#pragma config[2] = 0xFE & 0xF9 // PWRT açık, BOR kapalı
#pragma config[3] = 0xFE // Watchdog Timer kapalı

char    keypad_oku();
char    TUS;
void    ayarlar();
void    bekle(unsigned long t);         // t milisaniye gecikme sağlayan fonksiyon tanımı
```

```
void main()
{
	ayarlar();

//-----------------------------------------------
anadongu:
	bekle(1);	// Acquisition Time(Sample & Hold kapasitörünün şarj olması için
gerekli zaman)
	while(1)
 {
	PORTD=keypad_oku();
 }
	goto anadongu;
//-----------------------------------------------
}

////////////////////////////////////////////////////////////////////////////////////////////

void ayarlar()  // Bütün başlangıç ayarlarının tamamlandığı kısım
{
	GIE=0;				// Bütün kesmeleri kapat
	TRISC=0xF0;		// C portu giriş yapıldı
	TRISD=0;		// D portu çıkış yapıldı

	PORTD=0;		// D portu çıkışları sıfırlandı
	PORTC=0;		// C portu çıkışları sıfırlandı



}

////////////////////////////////////////////////////////////////////////////////////////////

char keypad_oku()	// tarama keypad'ın okunduğu kısım
{
	PORTC.0=1;
	if(PORTC.4==1)
	{bekle(50);TUS=0X01;}
	if(PORTC.5==1)
	{bekle(50);TUS=0X02;}
	if(PORTC.6==1)
	{bekle(50);TUS=0X03;}
	if(PORTC.7==1)
	{bekle(50);TUS=0X0A;}
	PORTC.0=0;

	PORTC.1=1;
	if(PORTC.4==1)
	{bekle(50);TUS=0X04;}
```

```
        if(PORTC.5==1)
        {bekle(50);TUS=0X05;}
        if(PORTC.6==1)
        {bekle(50);TUS=0X06;}
        if(PORTC.7==1)
        {bekle(50);TUS=0X0B;}
        PORTC.1=0;

        PORTC.2=1;
        if(PORTC.4==1)
        {bekle(50);TUS=0X07;}
        if(PORTC.5==1)
        {bekle(50);TUS=0X08;}
        if(PORTC.6==1)
        {bekle(50);TUS=0X09;}
        if(PORTC.7==1)
        {bekle(50);TUS=0X0C;}
        PORTC.2=0;

        PORTC.3=1;
        if(PORTC.4==1)
        {bekle(50);TUS=0X0E;}
        if(PORTC.5==1)
        {bekle(50);TUS=0X00;}
        if(PORTC.6==1)
        {bekle(50);TUS=0X0F;}
        if(PORTC.7==1)
        {bekle(50);TUS=0X0D;}
        PORTC.3=0;

        return TUS;

}

////////////////////////////////////////////////////////////////////////////////////////////////

void bekle(unsigned long t)   //t milisaniye gecikme sağlar
{
        unsigned x;

        for(;t>0;t--)
                for(x=140;x>0;x--)
                        nop();
}

////////////////////////////////////////////////////////////////////////////////////////////////
```