



KOCAELİ ÜNİVERSİTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ
MKT-308 MİKRODENETLEYİCİLER DERSİ
PROJE RAPORU

MAKSUT KAYA-090224040

AHMET YILDIZ-100224002

TAYFUN ÇETİN-100224037

YRD.DOÇ.DR. SELÇUK KIZIR

25.12.2014

İçindekiler

PROJENİN TANITIMI	2
Projenin Amacı	2
Projenin Yapım Basamakları	2
Projede Kullanılan Malzemeler	3
PIC16F877A	3
Temel Özellikleri	3
Çevresel Özellikleri	3
GRAFİK LCD WG12864B	4
LM35	4
Proje Devre Şeması	5
Grafik LCD Kütüphanesi	5
PROJENİN CCS PROGRAM KODLARI	6

PROJENİN TANITIMI

Projenin Amacı

Dünya şehir saatlerinin ve ortam sıcaklığının grafik LCD de görüntülenmesi ve saat ayarının butonlar yardımıyla yapılması

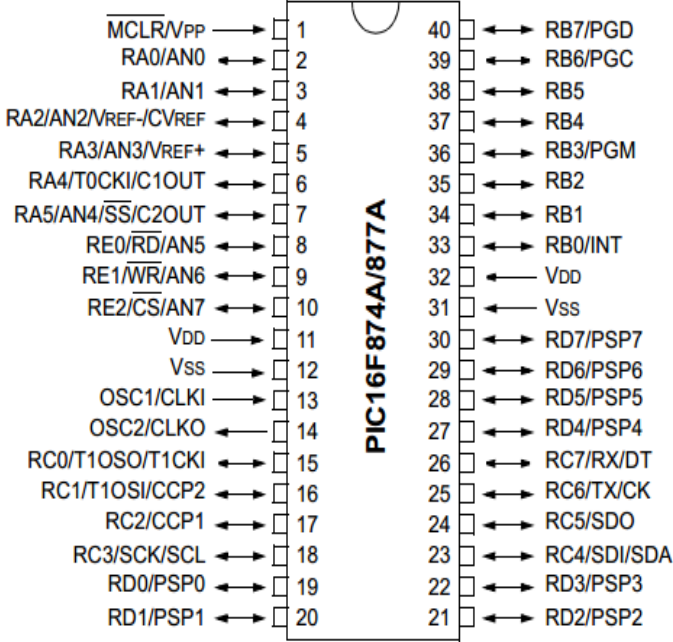
Projenin Yapım Basamakları

- Projemizdeki saat 20MHz lik osilatör ile timer1 kesmesi kullanılarak sağlanmıştır.
- Saat, dakika ayarları B portunun int_ext ve int_rb kesmelerinden yararlanılarak sağlanmıştır. Bu işlev için 5 buton kullanıldı.
- ADC modülü kullanarak sıcaklık ölçümü yapıldı.
- ADC modülünün çözünürlüğü 10 bit'tir.
- 5 volt ile beslenen 0-100 °C arası 10mV/°C hassasiyetle çalışan LM35 kullanıldı.
- Sıcaklık ve saat değerleri WG12864B grafik LCD'de gösterildi.
- Mevcut LCD kütüphanesindeki sürücü pinleri, saat ayarlarında int_ext ve int_rb kesmeleri kullanabilmek için C portları ile değiştirildi.
- Ölçülen sıcaklık her 4°C'de 1 kademe artacak şekilde termometre şekliyle görselleştirildi.
- Projenin Proteus programında simülasyonu gerçekleştirildi.



Projede Kullanılan Malzemeler

PIC16F877A



Temel Özellikleri

- İşlem hızı DC-20 MHz dir. (bir komut DC-200 ns hızında çalışmaktadır.)
- Dallanma komutları iki çevrim (cycle) sürede, bunların dışındaki komutlar ise bir çevrimlik sürede uygulanır.
- 8 K word Flash ROM programlama belleği (EEPROM özellikli program belleği), 368 Byte kullanıcı RAM belleği ve 256 Byte EEPROM belleği olmak üzere üç adet bellek bloğu vardır.
- 8 K word 'lük Flash ROM programlama belleği 1000000 kez programlanabilir.
- CPU azaltılmış komut setine sahiptir.
- Statik RAM üzerinde 32 adet özel işlem kaydedicisi (SFR - Special Function Register) vardır.

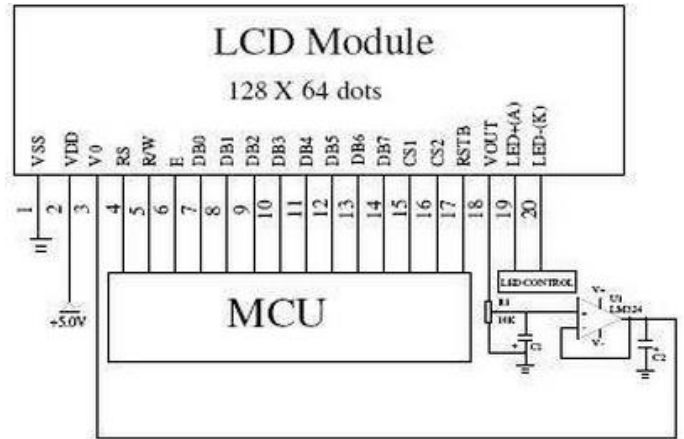
- RISC temeline dayanır.
- CPU azaltılmış komut setine sahiptir.
- Öğrenilecek gereken her biri 14 bitlik 35 komut vardır.
- Pin çıkışları PIC 16C73B/74B/76 ve 77 ile uyumludur.
- 8 bitlik veri yolu (databus) vardır.
- 14 kaynaktan kesme yapabilir.
- PIC16F877 dolaylı ve göreceli adresleme yapabilme özelliğine sahiptir.
- CMOSFlash EEPROM teknoloji ile düşük güçte yüksek hıza erişebilir.
- Enerji verildiğinde sistemi resetleme özelliği (Power-on Reset),
- Power-up zamanlayıcı (Power-up Timer)
- Osilatör başlatma zamanlayıcısı (Osilatör Start-up Timer)
- Özel tip zamanlayıcı (Watch-dog Timer), devre içi RC osilatör
- Hata ayıklamada kullanılabilecek modül (devre içi Debugger)
- Seçimli osilatör özellikleri
- Statik tasarım
- Enerji tasarrufu sağlayan uyku modu (Sleep Mode) özelliği vardır.
- Programla kod güvenliği sağlanabilir.
- Düşük gerilimli programlama özelliğine sahiptir.

- Sadece 5 V giriş ile devre içi seri programlanabilir.
- 2 pinle programlanabilir.
- Program belleğine okuma/yazma özelliği ile erişilebilir.
- 2,0 V ile 5,0 V arasında değişen geniş işletim aralığına sahiptir.
- Kaynak akımı 25 mA dir.
- Geniş sıcaklık aralığında ve düşük güçte çalışabilir.
- Devre içi iki pin ile hata ayıklama özelliği vardır.

Çevresel Özellikleri

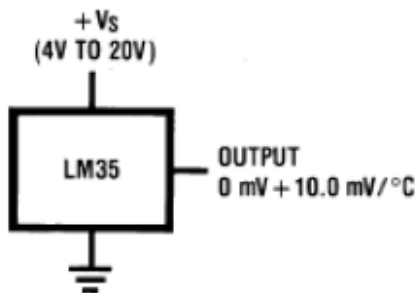
- TMR0: 8 bitlik zamanlayıcı, 8 bit önbölücülü
- TMR1: Önbölücülü, 16 bit zamanlayıcı, uyuma modundayken dış kristal zamanlayıcıdan kontrolü artırılabilir.
- TMR2: 8 bitlik zamanlayıcı, hem önbölücü hem de sonbölücü sabiti
- İki Capture / Compare / PWM modülü
- 10 bit çok kanallı A/D çevirici
- Senkron seri port (SSP), SPI (Master mod) ve I²C (Master Slave) ile birlikte
- Paralel Slave Port, 8 bit genişlikte ve dış RD, WR, CS kontrolleri
- USART/SCI, 9 bit adres yakalamalı
- BOR Reset (Brown Out Reset) özelliği

GRAFİK LCD WG12864B

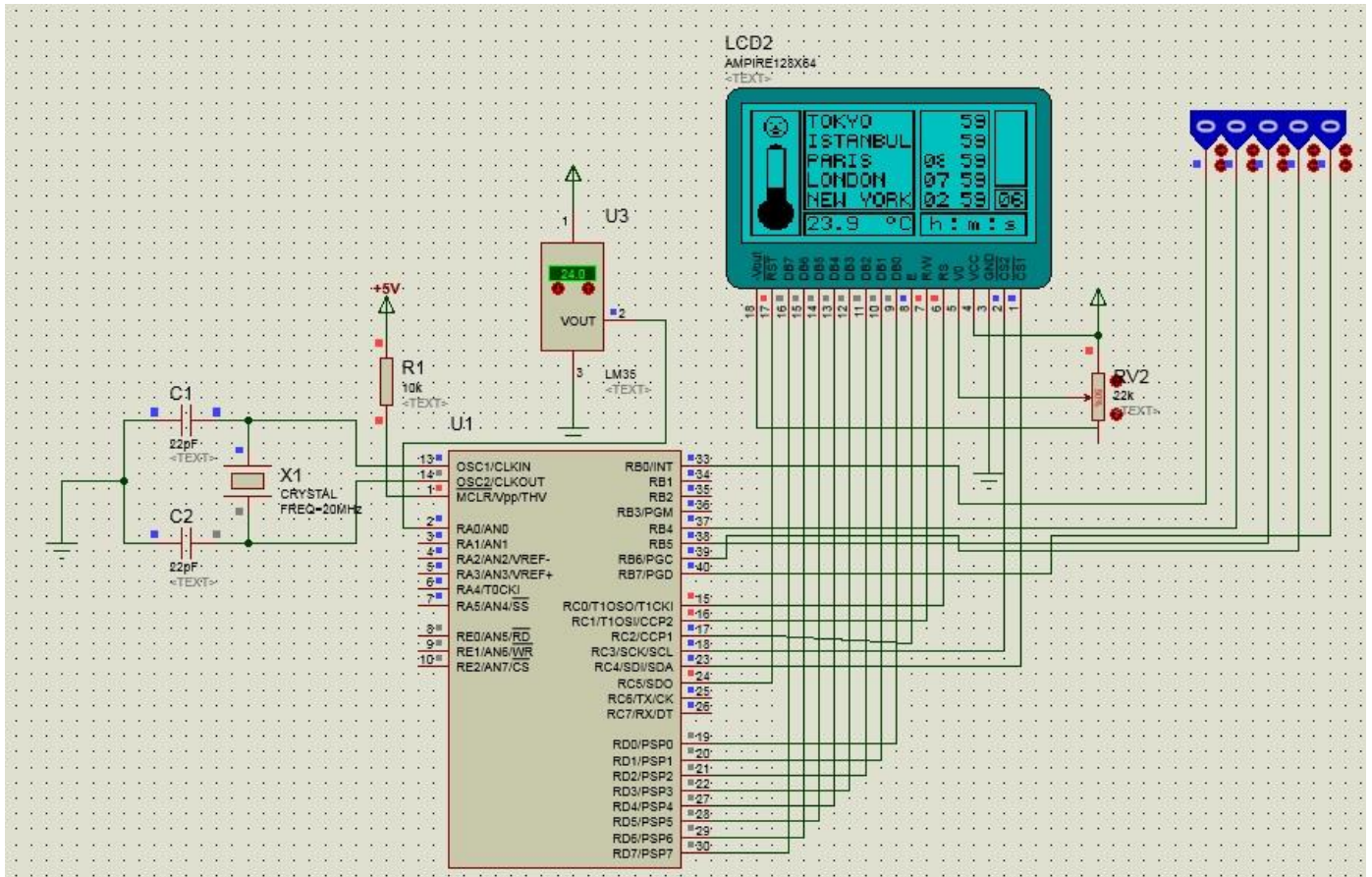


LM35

LM35 0 °C ile 100 °C arasında sıcaklık ölçümü yapabilen 3 bacaklı bir entegredir. Analog çıkış özelliği vardır. Analog çıkışından elde edilen gerilimler doğrusaldır. Bu sensörün çıkışı 10mv/°C(derece) olarak verilmiştir. 10 °C sıcaklık için yaklaşık olarak çıkışında 100 mV vermektedir.



Proje Devre Şeması



Grafik LCD Kütüphanesi

```
#ifndef HDM64GS12
#define HDM64GS12

#ifndef GLCD_WIDTH
#define GLCD_WIDTH 128
#endif

#ifndef GLCD_CS1
#define GLCD_CS1 PIN_C3 // Chip Selection 1
#endif

#ifndef GLCD_CS2
#define GLCD_CS2 PIN_C4 // Chip Selection 2
#endif

#ifndef GLCD_DI
#define GLCD_DI PIN_C0
// Data or Instruction input
#endif
```

```
#ifndef GLCD_RW
#define GLCD_RW    PIN_C1  // Read/Write
#endif

#ifndef GLCD_E
#define GLCD_E     PIN_C2  // Enable
#endif

#ifndef GLCD_RST
#define GLCD_RST   PIN_C5  // Reset
#endif

#define GLCD_LEFT  0
#define GLCD_RIGHT 1

#ifndef ON
#define ON          1
#endif

#ifndef OFF
#define OFF         0
#endif
```


PROJENİN CCS PROGRAM KODLARI

```
/******  
  
    PIC16F877 ile Grafik LCD Uygulaması  
  
*****/  
  
#include <16f877A.h>    // Kullanılacak denetleyicinin başlık dosyası tanıtılıyor.  
  
#fuses HS,NOWDT,NOPROTECT,NOBROWNOUT,NOLVP,NOPUT,NOWRT,NODEBUG,NOCPD  
  
// Denetleyici konfigürasyon ayarları  
  
#device ADC=10  
  
#use fast_io(a)  
  
#use fast_io(b)  
  
#use fast_io(c)  
  
#use delay (clock=20000000) // Gecikme fonksiyonu için kullanılacak osilatör frekansı belirtiliyor.  
  
#include <kit_c_glcd.c> // HDM64GS12.c dosyası programa ekleniyor  
  
#include <graphics.c> // graphics.c dosyası programa ekleniyor  
  
float bilgi;  
  
char yazi1[]="KOCAELI UNIVERSITY";    // Karakter dizisi tanımlanıyor  
  
char yazi2[]="Maksut Kaya";  
  
char yazi3[]="Ahmet Yildiz";  
  
char yazi4[]="Tayfun Cetin";// Karakter dizisi tanımlanıyor  
  
char yazi5[]="PIC PROJE";  
  
//char yazi6[]="Sicaklik:";  
  
char yazi7[2]={"h"};  
  
char yazi8[2]={"m"};  
  
char yazi9[2]={"s"};  
  
char yazi10[2]={"C"};  
  
char yazi11[2]={"0"};
```

```

/*-----saat tanımlamaları-----*/
int8 sayac=0; //8bitlik "sayac" isimli değişken tanımla.
int8 saat=0;
int8 i=1; //flash için tanımlanmıştır
int8 bar=0;
int8 dakika=0;
int8 saniye=0; //8bitlik "saat", "dakika", "saniye" değişkenleri
#define SIFIR pin_B0
#define SART pin_B4
#define SAZL pin_B5
#define DART pin_B6
#define DAZL pin_B7
/*-----*/
#INT_EXT
void sifirlama()
{
    saat=0;
    dakika=0;
    saniye=0;
    sayac=0;
}
#INT_RB
void saat_kur()
{
    if(input(SART))
    {
        saat++;
        saat=(saat%24);
    }
    if(input(SAZL))
    {
        saat--;
        saat=((24+saat)%24);
    }
    if(input(DART))
    {
        dakika++;
        dakika=(dakika%60);
    }
    if(input(DAZL))
    {
        dakika--;
        dakika=((60+dakika)%60);
    }
}
#INT_AD // ADC çevrimi bitti kesmesi
void ADC_Kesmesi ( )
{
    output_toggle(pin_b6);
}

```



```
/*-----SAAT FONKSİYONLARI-----*/
```

```
#INT_TIMER1 //TIMER1 kesme programı göstergesi
```

```
void kesme_prog() //Kesme programı başlangıcı
```

```
{
```

```
    set_timer1(15536); //Zamanlayıcı değerini 15536'ya kur.
```

```
    if(sayac<24) //"sayac" değeri 10'dan küçükse
```

```
        sayac++; //"sayac" değerini 1 arttır.
```

```
    else //"sayac" değeri 10'dan küçük değilse
```

```
        {
```

```
            i++;
```

```
            sayac = 0; //"sayac" değerini sıfırla./*
```

```
            output_toggle(pin_b7);
```

```
// -----
```

```
    if(saniye < 59) //"saniye" değeri 59'dan küçükse
```

```
        saniye++; //"saniye" değerini 1 arttır.
```

```
    else //"saniye" değeri 59'dan küçük değilse
```

```
        {
```

```
            saniye=0; //"saniye" değerini sıfırla.
```

```
// -----
```

```
    if(dakika < 59) //"dakika" değeri 59'dan küçükse
```

```
        dakika++; //"dakika" değerini 1 arttır
```

```
    else //"dakika" değeri 59'dan küçük değilse
```

```
        {
```

```
            dakika = 0; //"dakika" değerini sıfırla.
```

```
// -----
```

```
    if(saat < 23) //"saat" değeri 23'ten küçükse
```

```
        saat++; //"saat" değerini 1 arttır.
```

```
    else saat=0; //"saat" değeri 23'ten küçük değilse
```

```
        //"saat" değerini sıfırla.
```

```
    }
```

```

/***** ANA PROGRAM FONKSİYONU*****/
void main ( )
{
    setup_psp(PSP_DISABLED);    // PSP birimi devre dışı
    setup_timer_1(T1_INTERNAL | T1_DIV_BY_4);
    setup_timer_2(T2_DISABLED,0,1); // T2 zamanlayıcısı devre dışı
    setup_CCP1(CCP_OFF);        // CCP1 birimi devre dışı
    setup_CCP2(CCP_OFF);        // CCP2 birimi devre dışı
    setup_adc(adc_clock_div_32); // ADC clock frekansı fosc/32
    setup_adc_ports(AN0);
    set_tris_a(0b00000001);
    set_tris_c(0b00000000);
    set_tris_b(0b11110001);
    output_b(0x00);
    set_timer1(15536);
    saat=read_eeprom(0);
    dakika=read_eeprom(1);
    saniye=read_eeprom(2);

    glcd_init(ON); // Grafik LCD hazırlanıyor ve ekran siliniyor
    glcd_rect(8,3,118,13,NO,ON);
    glcd_rect(8,33,118,63,NO,ON);
    glcd_text57(10,5, yazi1, 1, ON); // Kocaeli üniversitesi yazdırılıyor
    glcd_text57(30, 35, yazi2, 1, ON); // GLCD'de yazı yazdırılıyor
    glcd_text57(30, 45, yazi3, 1, ON); // GLCD'de yazı yazdırılıyor
    glcd_text57(30, 55, yazi4, 1, ON);
    glcd_text57(35, 20, yazi5, 1, ON);
    glcd_line(35, 28,87,28,ON);
    delay_ms(2000);
    glcd_init(ON); // döngüye girmeden glcd son kez sıfırlanıyor
    glcd_rect(25,52,75,63,NO,ON); // sıcaklık çerçeve
    glcd_rect(78,52,126,63,NO,ON); // h,m,s çerçeve

    glcd_text57(83, 54,yazi7, 1, ON); // H yazdırılıyor
    glcd_text57(100, 54,yazi8, 1, ON); // M
    glcd_text57(117, 54,yazi9, 1, ON); // s
    glcd_rect(25,0,75,50,NO,ON); // şehir çerçeve
    glcd_text57(27, 2, şehir5, 1, ON); // şehirler yazdırılıyor
    glcd_text57(27, 12, şehir4, 1, ON); // şehirler yazdırılıyor
    glcd_text57(27, 22, şehir3, 1, ON); // şehirler yazdırılıyor
    glcd_text57(27, 32, şehir2, 1, ON); // şehirler yazdırılıyor
    glcd_text57(27, 42, şehir1, 1, ON); // şehirler yazdırılıyor

    glcd_rect(78,0,109,50,NO,ON); // saat dış çerçeve

    glcd_rect(112,40,126,50,NO,ON); // saniye dış çerçeve
    glcd_rect(112,0,126,38,NO,ON); // TIME dış çerçeve

    glcd_rect(1,0,23,63,NO,ON); // sıcaklık çerçeve
    glcd_pixel(12,17,on); // sıcaklık göstergesi
    glcd_line(10,18,14,18,ON);
    glcd_rect(8,19,16,44,NO,ON);
    glcd_circle(12,52,8,ON,ON);

```

```
ext_int_edge(L_TO_H);

enable_interrupts(INT_EXT);
enable_interrupts(INT_RB);
enable_interrupts(INT_TIMER1);
enable_interrupts(INT_AD); // AD çevrimi bitti kesmesi tanıtılıyor
enable_interrupts(GLOBAL);

while(true)
{
/*-----*/
/*-----SICAKLIK ÖLÇÜM KODLARI-----*/

set_adc_channel(0); // RE0/AN5 ucundaki sinyal A/D işlemine tabi tutulacak
delay_us(20); // Kanal seçiminde sonra bu bekleme süresi verilmelidir
bilgi=read_adc();// ADC sonucu okunuyor ve bilgi değişkenine aktarılıyor

sprintf(isi,"%3.1f",(0.004887585533*bilgi*100));

bar=0.004887585533*bilgi*100/4;
glcd_rect(9,20,15,43,ON,OFF); //kapatma
glcd_bar(12,(43-bar),12,43,7,ON);

glcd_rect(27,54,71,62,ON,OFF); // glcd ye alternatif pixel kapatma
glcd_text57(27,54,isi, 1, ON); // isi yazdırılıyor

glcd_pixel(64,54,on); // derece işareti kodları
glcd_pixel(65,54,on); // derece işareti kodları
glcd_pixel(63,55,on); // derece işareti kodları
glcd_pixel(63,56,on); // derece işareti kodları
glcd_pixel(66,55,on); // derece işareti kodları
glcd_pixel(66,56,on); // derece işareti kodları
glcd_pixel(64,57,on); // derece işareti kodları
glcd_pixel(65,57,on); // derece işareti kodları

glcd_text57(69,54,yazi10, 1, ON); // C yazdırılıyor
```

```

if(i==1)
{
    glcd_circle(12,8,5,NO,ON); // :)
    glcd_pixel(10,6,on);    // gözler
    glcd_pixel(14,6,on);
    glcd_pixel(12,8,on); //burun
    glcd_pixel(10,10,on); // ağız
    glcd_pixel(14,10,on);
    glcd_line(11,11,13,11,ON);
    glcd_line(11,9,13,9,ON);

    glcd_rect(93,55,94,56,ON,ON); // her nokta tek tek tanımlanıyor
    glcd_rect(93,59,94,60,ON,ON); // iki nokta birlikte bar olarak tanımlanıp
    glcd_rect(110,55,111,56,ON,ON); // ortasındaki fazla pixeller de silinebilir
    glcd_rect(110,59,111,60,ON,ON);
    glcd_rect(117,2,121,36,ON,OFF); // flash TIME

    glcd_rect(93,3,94,48,ON,OFF); // şehirlerin : flash
}
if(i==2)
{
    glcd_rect(93,3,94,4,ON,ON); // NOW YORK :
    glcd_rect(93,7,94,8,ON,ON); //

    glcd_rect(93,13,94,14,ON,ON); // LONDON :
    glcd_rect(93,17,94,18,ON,ON); //

    glcd_rect(93,23,94,24,ON,ON); // PARIS :
    glcd_rect(93,27,94,28,ON,ON); //

    glcd_rect(93,33,94,34,ON,ON); // ISTANBUL :
    glcd_rect(93,37,94,38,ON,ON); //

    glcd_rect(93,43,94,44,ON,ON); // TOKYO :
    glcd_rect(93,47,94,48,ON,ON); //

    glcd_rect(93,55,94,60,ON,OFF); // flash :
    glcd_rect(110,55,111,60,ON,OFF); // flash :
    glcd_rect(7,3,17,13,ON,OFF); // flash :)
    glcd_line(117,2,121,2,ON); // T harfi
    glcd_line(119,3,119,9,ON); // T harfi
    glcd_line(119,12,119,18,ON); // I harfi
    glcd_line(117,21,117,27,ON); // M harfi
    glcd_line(117,22,119,24,ON); // M harfi
    glcd_line(119,24,121,22,ON); // M harfi
    glcd_line(121,21,121,27,ON); // M harfi
    glcd_line(117,30,117,36,ON); // E harfi
    glcd_line(118,30,121,30,ON); // E harfi
    glcd_line(118,33,121,33,ON); // E harfi
    glcd_line(118,36,121,36,ON); // E harfi
    i=0;
}
}
saat_yazdir();
}
}
void saat_yazdir()
{
    glcd_rect(80,2,90,49,ON,OFF); // saat içersini sıfırlama
    sprintf(hourc,"%d",saat);
    sprintf(hourc1,"%d",((saat+5)%24));
    sprintf(hourc2,"%d",((saat+6)%24));
    sprintf(hourc3,"%d",((saat+7)%24));
    sprintf(hourc4,"%d",((saat+12)%24));

```

```

if(saat<10)
{
    glcd_text57(80,42,yazi11,1, ON);
    glcd_text57(86,42,hourc,1, ON);
}
else glcd_text57(80,42,hourc,1, ON); //
    if(((saat+5)%24)<10)
    {
        glcd_text57(80,32,yazi11,1, ON);
        glcd_text57(86,32,hourc1,1, ON); //
    }
else    glcd_text57(80,32,hourc1,1, ON); //
    if(((saat+6)%24)<10)
    {
        glcd_text57(80,22,yazi11,1, ON);
        glcd_text57(86,22,hourc2,1, ON);
    }
else    glcd_text57(80,22,hourc2,1, ON); //
    if(((saat+7)%24)<10)
    {
        glcd_text57(80,12,yazi11,1, ON);
        glcd_text57(86,12,hourc3,1, ON); //
    }
else    glcd_text57(80,12,hourc3,1, ON); //
    if(((saat+12)%24)<10)
    {
        glcd_text57(80,2,yazi11,1, ON);
        glcd_text57(86,2,hourc4,1, ON); //
    }
else    glcd_text57(80,2,hourc4,1, ON); //
glcd_rect(97,2,107,49,ON,OFF); //dakika içersini sıfırlama
sprintf(dakikac,"%d",dakika);
    if(dakika<10)
    {
        glcd_text57(97,2,yazi11,1, ON);
        glcd_text57(97,12,yazi11,1, ON);
        glcd_text57(97,22,yazi11,1, ON);
        glcd_text57(97,32,yazi11,1, ON);
        glcd_text57(97,42,yazi11,1, ON); //
        glcd_text57(103,2,dakikac,1, ON); //
        glcd_text57(103,12,dakikac,1, ON); //
        glcd_text57(103,22,dakikac,1, ON); //
        glcd_text57(103,32,dakikac,1, ON); //
        glcd_text57(103,42,dakikac,1, ON); //
    }
else
    {
        glcd_text57(97,2,dakikac,1, ON); //
        glcd_text57(97,12,dakikac,1, ON); //
        glcd_text57(97,22,dakikac,1, ON); //
        glcd_text57(97,32,dakikac,1, ON); //
        glcd_text57(97,42,dakikac,1, ON); //
    }
    sprintf(saniyec,"%d",saniye);
glcd_rect(114,42,124,48,ON,OFF);
    if(saniye<10)
    {
        glcd_text57(114,42,yazi11,1, ON);
        glcd_text57(120,42,saniyec,1, ON); //
    }
else
    glcd_text57(114,42,saniyec,1, ON); //
}

```

