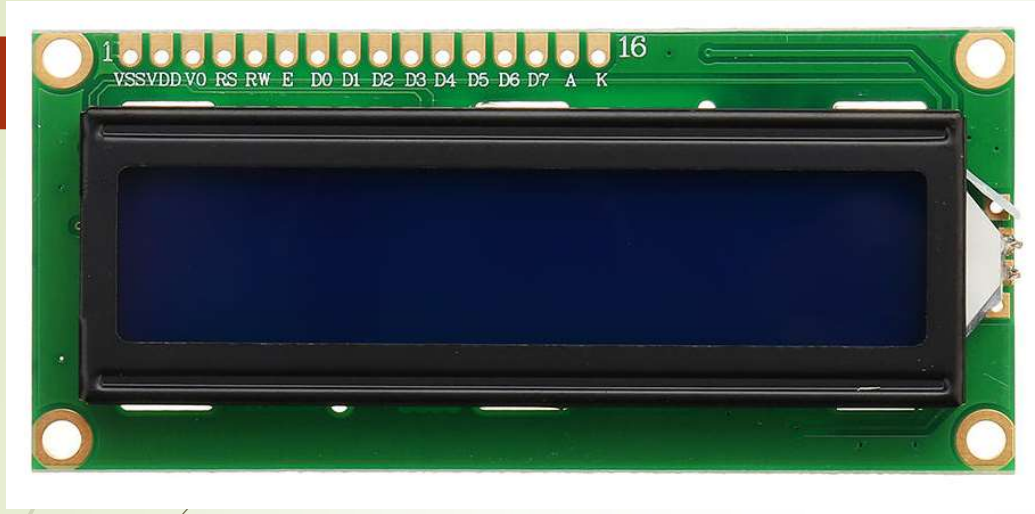
BMS 302 MİKROKONTROLCÜLER

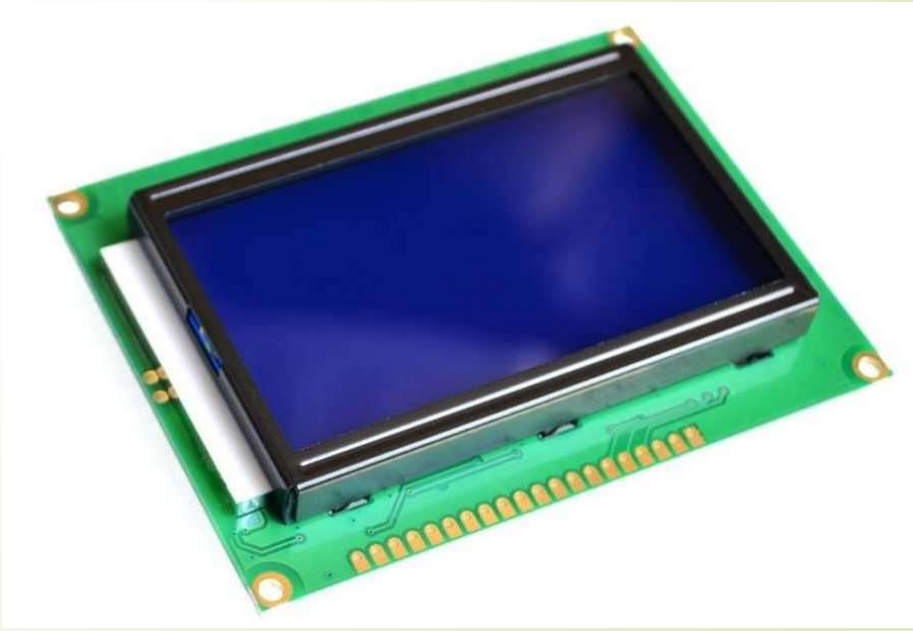
LCD EKRAN UYGULAMALARI

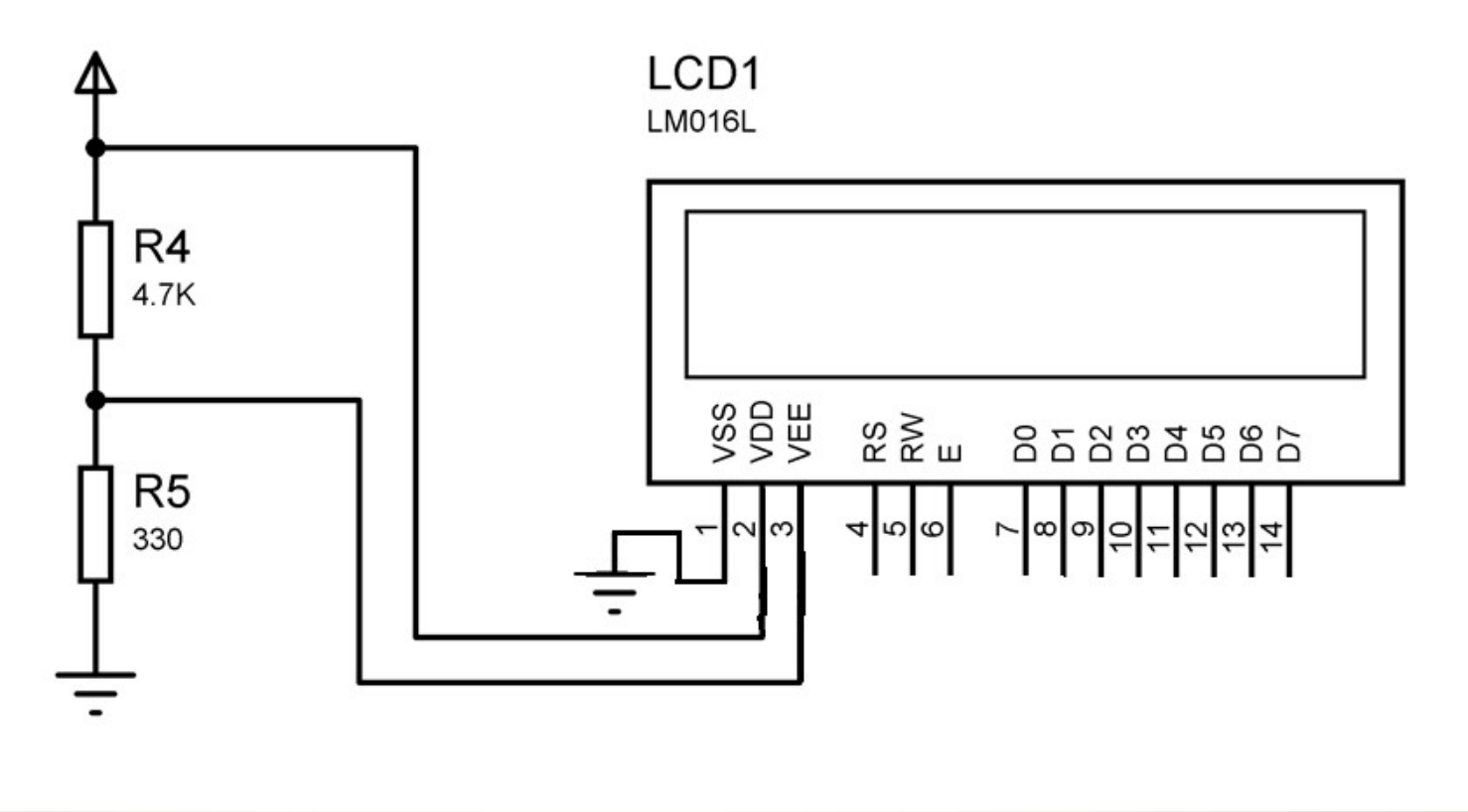
Prof.Dr. Mutlu AVCI

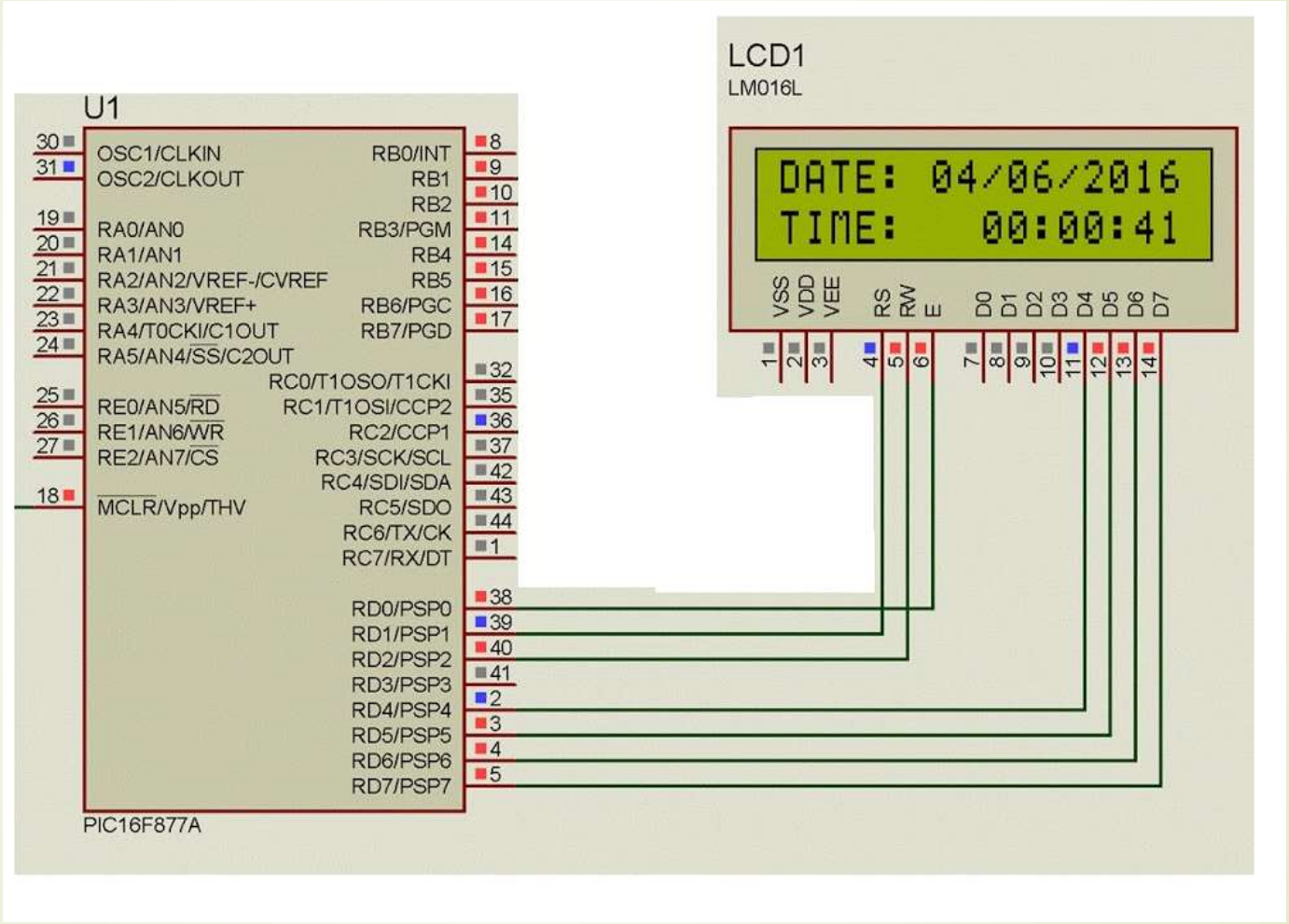
Çukurova Üniversitesi Biyomedikal Mühendisliği Bölümü

Bahar 2020


1
16
VSS VDD VO RS RW E D0 D1 D2 D3 D4 D5 D6 D7 A K




LCD1
LM016L
R4
4.7K
R5
330
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
1
2
3
4
5
6
7
8
9
10
11
12
13
14


U1
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
MCLR/Vpp/THV
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877A
LCD1
LM016L
DATE: 04/06/2016
TIME: 00:00:41
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7

CCS C Karakter Tabanlı Paralel LCD

- CCS C derleyicisi paralel karakter LCD'lerle iletişim için kütüphanesinde "*lcd.c*" dosyasını içerir.
- Bu dosya sayesinde karakter tabanlı LCD'ler ile kolayca iletişim kurulabilmektedir.
- Bu dosyaların programa dahil edilmesi için, programın başında kullanılacak LCD dosyası ***#include <lcd.c>*** komutu ile programa eklenmelidir.

LCD Bağlantılarının Belirtilmesi

- LCD bağlantılarının belirtilmesi için iki yol vardır. ***Bunlardan birincisi***, LCD'nin direkt B yada D portuna bağlanmasıdır. Bunun için LCD bağlantılarının aşağıda belirtildiği gibi ilgili uçlarla yapılması gerekir.

LCD B Portuna Bağlanırsa ;	**LCD D Prtuna Bağlanırsa ;**
B0 > LCD E (Enable) ucuna	*D0 > LCD E (Enable) ucuna*
B1 > LCD RS ucuna	*D1 > LCD RS ucuna*
B2 > LCD R/W ucuna	*D2 > LCD R/W ucuna*
B4 > LCD D4 ucuna	*D4 > LCD D4 ucuna*
B5 > LCD D5 ucuna	*D5 > LCD D5 ucuna*
B6 > LCD D6 ucuna	*D6 > LCD D6 ucuna*
B7 > LCD D7 ucuna	*D7 > LCD D7 ucuna*

- Ayrıca B veya D portu için aşağıdaki tanımlamalar *#include <lcd.c>* komutundan önce yazılmalıdır.

- ***B Portu için ;***
- *#define use_portb_lcd TRUE // veya*
- *#define LCD_DATA_PORT getenv("SFR:PORTB")*

- ***D Portu için ;***
- *#define LCD_DATA_PORT getenv("SFR:PORTD")*

- ***İkinci yöntemde*** ise LCD istenen portun istenen uçlarına bağlanabilir. Bunun için aşağıda verilen örnekte olduğu gibi, LCD'nin her ucunun kontrolcüdeki hangi uçlara bağlanacağının belirtilmesi gerekir.

- #define LCD_ENABLE_PIN PIN_B0
- #define LCD_RS_PIN PIN_E1
- #define LCD_RW_PIN PIN_E2
- #define LCD_DATA4 PIN_B3

- #define LCD_DATA5 PIN_B7
- #define LCD_DATA6 PIN_C2
- #define LCD_DATA7 PIN_A3

- LCD.c dosyasında LCD kontrolü için hazır fonksiyonlar bulunmaktadır. Bu fonksiyonlar ve anlamları aşağıda açıklanmıştır.

lcd_init ( ) = Diğer LCD fonksiyonları çağırılmadan bu fonksiyon mutlaka bir kez çağırılmalıdır.

.

lcd_putc (c) = LCD'ye karakter veya string yazmak için kullanılır.

- Parantez içerisine gönderilecek karakter veya string yazılır. Tek karakter gönderilirken tek tırnak içine alınır 'a'. Eğer string ifadesi göderilecekse çift tırnak içerisine alınır "string". Aynı zamanda parantez içerisine aşağıda belirtilen ***ters eğik çizgi sabitleri*** de getirilebilir

- \a = *imleç pozisyonunu sol üste ayarlar*
- \f = Display'i siler, imleci sol üste ayarlar
- \n = *Satır atlar*
- \b = *bir karakter geri gelir*

lcd_gotoxy(x, y); // İmleci x sütunu y satırına götürür

Lcd_getc(x,y); // imlecin belirtilen konumunda bulunan karakterle geri döner

Lcd_send_byte(byte adres, byte bilgi); // LCD ye istenilen bir komutu gönderir

Lcd_read_byte(); // LCD den bir byte veri okur.

- *printf(lcd_putc," ")* = lcd_putc ( ) komutu ile LCD'ye karakter veya string yazdırmak yerine printf(lcd_putc, " ") komutu kullanılabilir.

- Bu komut kullanılarak LCD'ye değişken sabit veya string belli bir formak ile gönderilebilir.
- Printf komutunda LCD için parantez içine ilk başta kullanılacak LCD komutu yazılır sonra virgül konulur ve çift tırnak açılarak gerekli bilgiler ve sabitler yazılır.
- Çift tırnaktan sonra lcd_putc komutunda olduğu gibi ters eğik çizgi sabitleri kullanılabilir. Komutta % işareti ve ondan sonra gelen kod, değişkenin ekranda hangi formatta yazılacağını belirtir.

- Bu fonksiyonun kullanımı ;

printf(lcd_putc, “\f\n\b ifade %d %f…”,değişken1,değişken2);

printf(lcd_putc,”\f\nSicaklik=%d C”,temp);

- şeklindedir.

- Yukarıdaki örnek ile ilk başta \f komutu ile LCD silinmektedir. Ardından \n komutu ile LCD’ye satır atla komutu verilmiştir ve LCD imleci ikinci satıra geçmiştir.

- LCD'ye 'Sicaklik=' yazısı yazdırılmış, ardından da %d komutunun olduğu yere "temp" demp değişkeninin içeriği tam sayı formatında yazdırılmıştır. sonra 1 karakter boşluk verilmiş ve en son olarak da "C" harfi LCD'ye yazdırılmıştır.

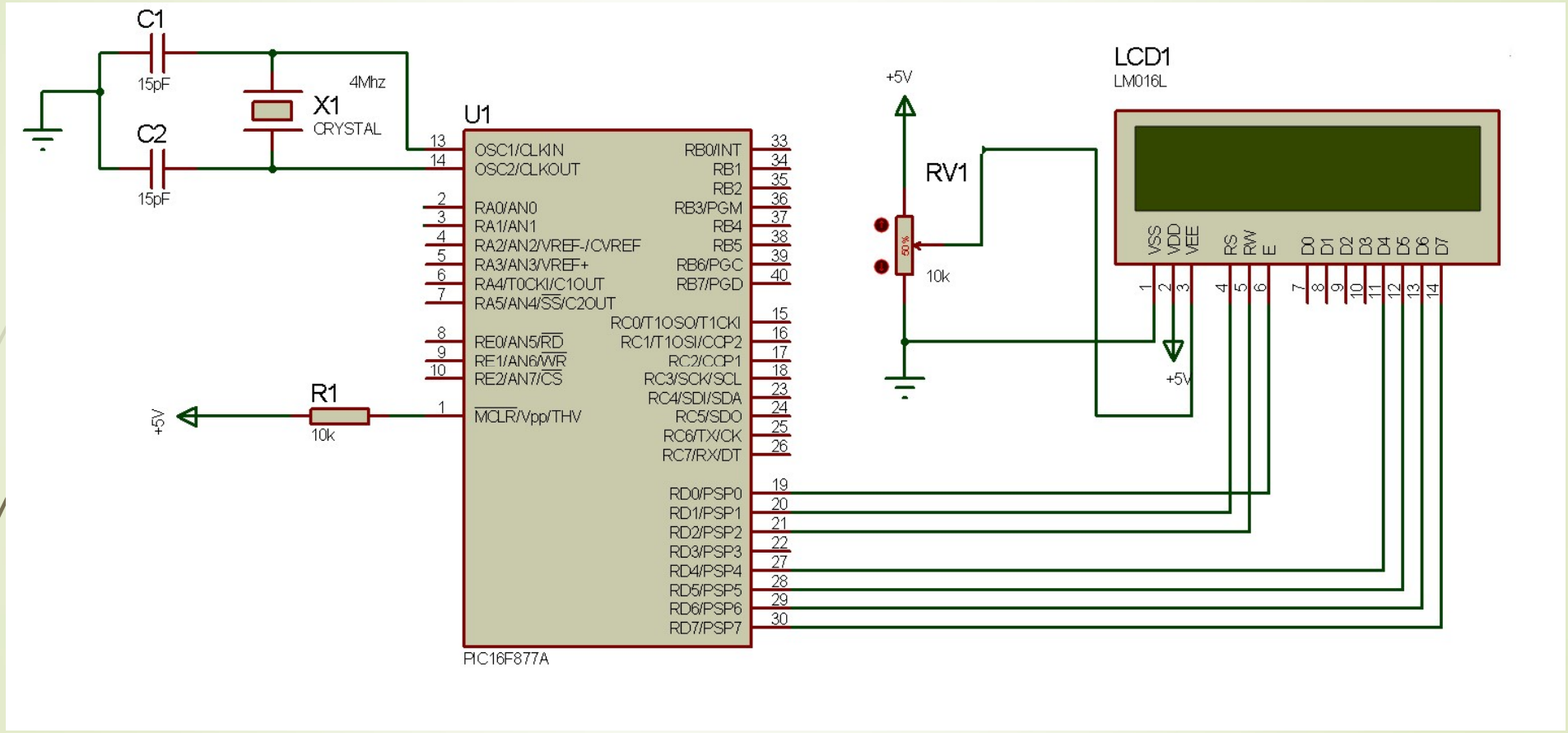
- lcd.c dosyasında gecikme fonksiyonları kullanıldığından programda gecikme fonksiyonlarının kullanılacağı osilatör frekansı belirten #use delay ( ) komutu mutlaka #include <lcd.c> komutundan önce belirtilmelidir.

Örnek: PIC 16F877A mikrokontrolcüsünün D portuna bağlı karakter LCD'ye

a)1 den 10 a kadar olan tam sayıları 1 artarak

b) 1 den 10 a kadar olan ondalıklık sayıları 0.5 artarak

0.5 sn ara ile yazdıran CCS C programını yazın.


C1
15pF
C2
15pF
4Mhz
X1
CRYSTAL
U1
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
MCLR/Vpp/THV
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
PIC16F877A
R1
10k
+5V
RV1
50%
10k
+5V
+5V
LCD1
LM016L
VSS
VDD
VEE
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7

a)

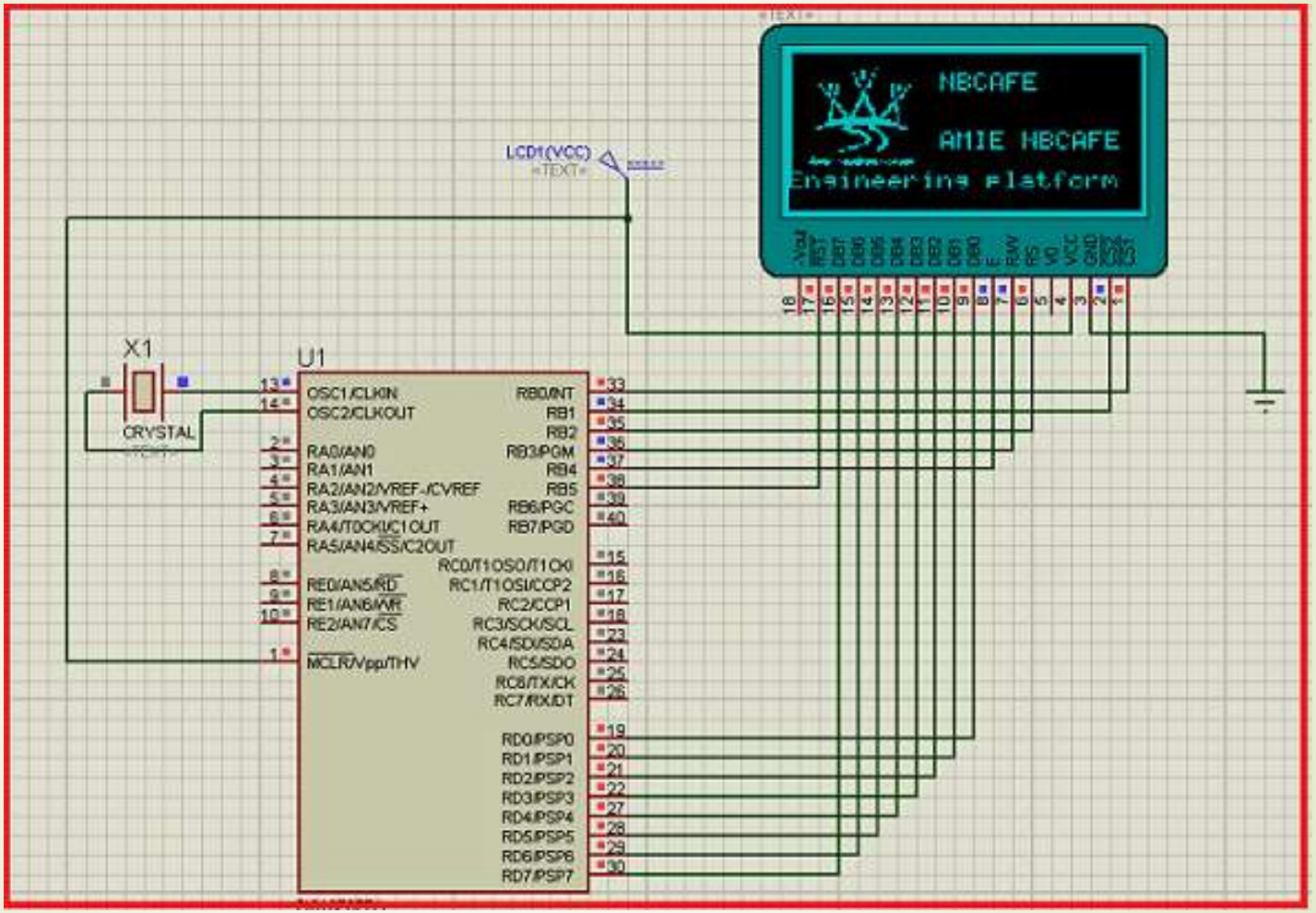
```
#include <prog.h>
#use delay(clock=4000000)
#use fast_io(D)
#define LCD_DATA_PORT getenv(«SFR:PORTD»)
#include <lcd.c>
int i;
void main(){
set_tris_D(0);
Lcd_init();
While(1){
For (i=1;i>=10;i++){
Printf(lcd_putc, «sayi = %d»,i);
Delay_ms(500);}}
```

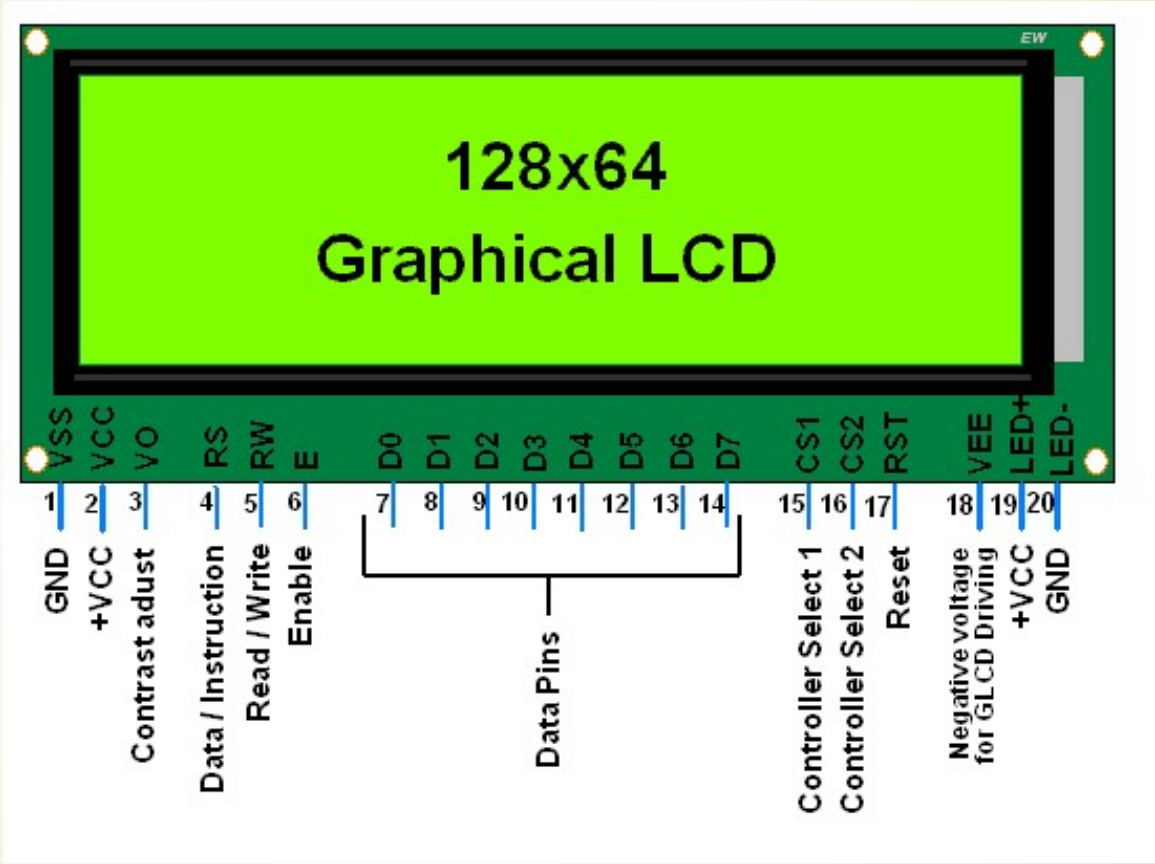
b)

```
#include <prog.h>
#use delay(clock=4000000)
#use fast_io(D)
#define LCD_DATA_PORT getenv(«SFR:PORTD»)
#include <lcd.c>
int i;
float x=0;
void main(){
set_tris_D(0);
Lcd_init();
While(1){
For (i=1;i>=20;i++){
X+=0.5;
Printf(lcd_putc, «sayi = %f»,x);
Delay_ms(500);}}
```

Grafik LCD

- Karakter LCD lerin tüm fonksiyonlarını yerine getirebilirler.
- İlaveten piksel bazlı erişim sayesinde grafikte çizebilirler.
- İçerdiği entegreye göre kütüphane eklenip kullanılırlar.
- Piyasada genellikle K0108 entegresi uyumlu HDM64GS12 çipsetli grafik LCD ler bulunmaktadır.


NBCAFE
AMIE NBCAFE
Engineering Platform
LCD1(VCC)
X1
CRYSTAL
U1
OSC1/CLKIN
OSC2/CLKOUT
RA0/AN0
RA1/AN1
RA2/AN2/VREF-/CVREF
RA3/AN3/VREF+
RA4/T0CKI/C1OUT
RA5/AN4/SS/C2OUT
RE0/AN5/RD
RE1/AN6/WR
RE2/AN7/CS
MCLR/Vpp/THV
RB0/INT
RB1
RB2
RB3/PGM
RB4
RB5
RB6/PGC
RB7/PGD
RC0/T1OSO/T1CKI
RC1/T1OSI/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6/TX/CK
RC7/RX/DT
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7


128x64
Graphical LCD
EW
VSS
VCC
VO
RS
RW
E
D0
D1
D2
D3
D4
D5
D6
D7
CS1
CS2
RST
VEE
LED+
LED-
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
GND
+VCC
Contrast adust
Data / Instruction
Read / Write
Enable
Data Pins
Controller Select 1
Controller Select 2
Reset
Negative voltage for GLCD Driving
+VCC
GND

CCS C ile Grafik LCD sürme

- Kütüphaneler

GLCD.c, HDM64GS12.c, GRAPHICS.c, KS0108.c

GLCD.c ve HDM64GS12.c kütüphanelerinde LCD bağlantısı aşağıdaki şekildedir.

RB0 → CS1	RD0 → DB0	RD6 → DB6
RB1 → CS2	RD1 → DB1	RD7 → DB7
RB2 → DI	RD2 → DB2	
RB4 → R/W	RD3 → DB3	
RB5 → E	RD4 → DB4	
RC0 → RST	RD5 → DB5	

- **GLCD.c** kütüphanesinin sağladığı fonksiyonlar:

Glcd_init(mode) = mode ON veya OFF olabilir. Bu komut LCD yi açar veya kapatır.

Glcd_pixel(x,y,color) = x, y değişkenleri koordinatları belirtir, color ise ON veya OFF değerleri alır.

Glcd_line(x1, y1, x2, y2, color) = belirtilen koordinatlar arası çizgi çeker. Color yine ON veya OFF değeri alır.

Glcd_bar(x1, y1, x2, y2, width, color)

Glcd_circle(x,y,radius,fill,color)

Glcd_text57(x, y, textptr, size, color)

Glcd_fillScreen(color)

HDM64GS12.c dosyasının fonksiyonları

#define FAST_GLCD

Glcd_update()

Glcd_init(mode)

Glcd_pixel(x,y,color)

Glcd_fillScreen(color)

- **graphics.c** dosyası fonksiyonları

GLCD.c dosyasıyla aynı fonksiyonları barındırır.

Genellikle HDM64GS12.c ve graphics.c dosyası birlikte kütüphane olarak kullanılarak bütün işlevler sağlanır.

- Örnek: PIC16F877 A kullanarak bir grafik LCD üzerine yazı yazan, çizgi, dikdörtgen ve daire çizen CCS C programını yazın.

```
#include <prog.h>
#include <HDM64GS12.c>
#include <graphics.c>
Char yazi[]=«ADANA»;
İnt i,x1,x2,y1,y2;
Void main(){
Glcd_init(ON);
Glcd_text57(40,5,yazi,2,ON);
Delay_ms(2000);
Glcd_init(ON);
Glcd_line(5,5,100,5,ON);
Delay_ms(2000);
Glcd_init(ON);
Glcd_rect(5,5,120,65,NO,ON);
Delay_ms(2000);
Glcd_init(ON);
Glcd_circle(60,30, 20,NO,ON);
Delay_ms(2000);
}
```