Modül 3: MSP430 Denetleyiciler ile Karakter LCD Kullanımı

Giriş

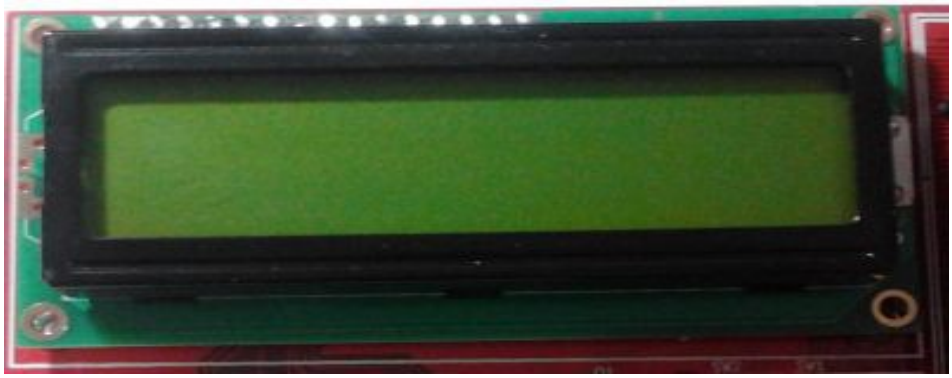
LCD'ler (Liquid Crystal Display/Sıvı Kristal Ekran) adından da anlaşılacağı gibi ekran yani görüntüleme elemanlarıdır. LCD genel bir tanımdır. Bir çok LCD çeşidi olmakla birlikte bir çok alanda kullanılmaktadır. Çeşit olarak, Renkli, Siyah beyaz, Grafik, Karakter gibi bir çok çeşit LCD tipi bulunmaktadır. Kullanım alanları ise çok geniştir. Televizyonlar, Monitörler, Bilgisayarlar, Otomobiller, Elektronik sistemler vb. bir çok sistemde kullanılırlar.

LCD'lerin temel amacı karşısındaki kişiye yada kullanıcıya istenilen görüntüyü sağlamaktır. Karakter LCD'ler ise sadece karakter temelli görüntü sağlarlar. Yani resim yada istenilen tüm şekilleri gösteremezler. Düşük maliyet, küçük boyutları ve düşük güç tüketimleri nedeniyle küçük ölçekli sistemlerde kullanıma uygundur.

Karakter LCDler 2x16, 2x20, 4x16, 4x20 gibi çeşitli boyutlarda tiplerde bulunmaktadır. Bu modül kapsamında biz MPS430 denetleyicler ile geliştirme kartımız üzerinde bulunan 2x16(2 satır 16 sütun) karakter LCDnin kullanımından bahsedeceğiz.

2x16 Karakter LCD

MSP430 geliştirme kartımızın sol üst köşesinde bulunan elemanız 2x16 karakter LCDdir. Aşağıda şekil 1'de resmi görülmektedir.



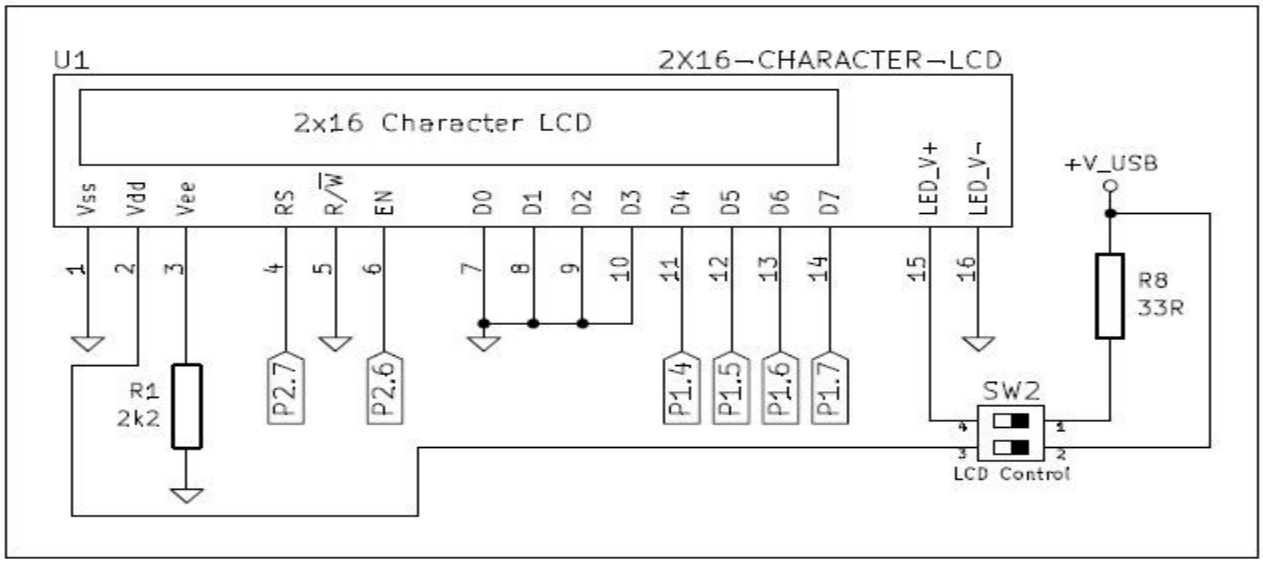
Şekil 1: 2x16 Karakter LCD

Karakter LCDleri kullanmak için 8 bit veri yolu ve kontrol pinleri bulunmaktadır. Bu pinler mikrodenetleyicinin portlarına bağlanarak uygun kodların yazılması ile lcd üzerinde istediğimiz yazıları görüntüleyebiliriz.

Piyasada bulunan coğu karakter LCD üzerinde HD44780 isimli Hitachi firmasına ait karakter LCD kontrolörü bulunmaktadır. Bu kontrolör LCD ile mikrodenetleyici arasında köprü vazifesi görür. Yani biz direk mikrodenetleyici ile LCD üzerindeki pixellere müdahale etmeyiz. Kontrolör üzerinden istediğimiz karakterlerin gösterilmesini sağlarız. HD44780 genel amaçlı bir kontrolör olduğu için çoğu karakter LCD üreticisi ürettiği çeşitli boyutta ve özellikteki LCDlerinde bu kontrolörü kullanır.

HD44780 kontrolörü genel olarak ascii tablsounda bulunan karakterleri göstermekle birlikte birçok işaret ve başka alfabelerde ki harfleride gösterebilmektedir. Detaylı bilgi için kullandığınız LCD ve HD44780 kontrolörün kullanma kılavuzunu okumakta fayda var.

Ayrıca HD44780 kontrolörü 8 adet özel karakter tanımlamanıza da izin vermektedir. Bu sayede özel semboller tanımlayabilir yada ascii tablosunda bulunmayan Türkçe karakterlerimizi tanımlayabiliriz.



Şekil 2: Karakter LCD Pin Bağlantıları ve Geliştirme Kartı Uygulama Devresi

Şekil 2'de LCDmizin pin bağlantıları ve MSP430 denetleyicimize bağlantısı ve uygulama devresi görülmektedir. Görüldüğü üzere 1 numaralı pin VSS yani gnd (şase) pinidir. Aynı şekilde 2 numaralı pin VDD yani besleme pinidir. Karakter LCDmizin besleme gerilmi 5V'tur. Kullandığınız LCDler farklı gerilimlerde çalışabilir. Emin olmak için kullanma kılavuzlarını okumakta fayda var. 3 numaralı pin ise kontrast ayarı için kullanılır. Uygun bir direnç ile gnd hattına bağlanarak lcd kontrast ayarı yapılır.

RS, R/W ve EN pinleri kontrol pinleridir.

RS: Register Select pinidir. Lojik 0 durumunda HD44780'e veri yolundan komut gönderilir. Lojik 1 yapılırsa HDD780'e veri yolundan veri yazılır yada okunur.

HD44780'e ayarlamalar yapmak için komut, karakter verisi göndermek yada almak içinse veri yazmak yada almak gereklidir.

R/W: Okuma yazma pinidir. Lojik 1 durumunda HD44780'den okuma yapılır. Lojik 0 olması durumunda HD44780'e yazma yapılır.

HD44780'e genel olarak yazma yapıldığından uygulamalarda bu pin genellikle direk olarak gnd'ye bağlanır.

EN: Enable yani yetki pinidir. Lojik 1 olması durumunda HD44780'e okuma yazma işlemi yapılır. Lojik 0 durumunda yapılamaz.

D0:D7: HD44780 kontrolörünün veri yoludur 8 bit genişliğindedir okuma, yazma işlemlerinde kullanılır.

LED+, **LED**- : Bu pinler HD44780 ile ilgili olmasalarda LCD'nin ekran aydınlatması için kullanılır. LCD içersinde bulunan aydınlatma amaçlı LED uygun bir direnç ile 5V'a bağlanarak LCD'nin arka plan aydınlatması sağlanır.

Şekil 2'de görüldüğü üzere LCDmizin D0:D3 pinleri gndye bağlanmıştır yani kullanılmamıştır. HD44780 kontrolörünün bir diğer özelliği ise 4 bit veri yolu ile kontrol edilebilmesidir. Yani sadece 4 bit ile veri yolu ile veri yazıp okuyabiliriz. Bu sayede denetleyicimizden 4 pin tasarruf ederiz. Tabi 4 bit veri yolu kullandığımız için 8 bitlik veri yazmak 2 katı zaman alacaktır. MSP430 denetleyicimiz yeterince hızlı olduğu için 2 kat zaman alması sorun değildir. 4 bit veri yolu moduna yazılımla ayarlanarak geçilmektedir.

MSP430 ile Karakter LCD Kullanımı

Karakter LCDlerin yapısını anlattıktan sonra bu LCDleri MSP430 denetleyicimiz ile nasıl kullancağımıza değinelim. Görüldüğü gibi okuma/yazma pinini kullanmayıp(gndye bağlı) 4 bit veri yolunu kullanırsak toplamda LCDyi kullanmak için 6 adet port pinie ihtiyacımız olacaktır. Bunlardan 4 biti veri yolu olduğu için veri yolunun P1.4:P1.7 gibi ardışıl olarak port bitlerine bağlanması yazılımda kolaylık sağlayacaktır. EN ve RS bitleri ise istenilen port bitlerine bağlanarak bit temelli işlemler ile kontrol edilebilir.

Karakter LCDmizi denetleyicimize bağladıktan sonra kullanabilmek için uygun kodları yazmamız hatta modüler olması ve diğer uygulamalarda rahatça kullanabilmek için uygun bir kütüphane hazırlamak gerekir. Bu kütüphanelerde temel olarak ilk ayar rutini, komut/veri okuma/yazma rutinleri ve karakter okuma/yazma rutinleri bulunmaktadır. Kullanacağınız uygulamaya bu kütüphaneleri dahil ederek istediğiniz karakterleri LCD ekranda görüntüleyebilirsiniz.

Karakter LCDler yaygın olarak kullanıldığı için internet üzerinde bir çok denetleyici ile uyumlu, bir çok dilde yazılmış kütüphaneler bulabilirsiniz. Bizde uygulamamızda böyle bir kütüphaneden faydalanacağız. LCD gibi bu tarz harici donanımları kütüphane yazarak kullanmak, bu kütüphaneleri dolayısıla harici donanımlarımızı bir çok uygulamada kolayca kullanmamıza olanak sağlar.

MSP430 Karakter LCD Kütüphanesi

Karakter LCDmizi kullanmak için kütüphanemizi oluştururken yada başka bir kütüphane oluştururken, Kütüphaneler genellikle kütüphane ile ilgili .h uzantılı header(başlık) dosyası ve .c uzantılı C source(kaynak) dosya olmak üzere iki dosya halinde hazırlanır. Başlık dosyasında kütüphane ilgili ön tanımlamalar, ön-işlemci ayarları,varsa port tanımlamaları ve sabit değerler tanımlanır. Ayrıca kaynak dosyasında kullanılan fonksiyonların prototipleride başlık dosyasında bulunur. Bunlar genel kuraldır yapılması zorunlu değildir. Sadece kaynak dosyası bulunan kütüphanelerde oluşturabilir. Kaynak dosyasında ise başlık dosyasında tanımladığımız fonksiyonların içerikleri bulunur. Karakter LCDmize ait LCD.h ve LCD.c kütüphane dosyalarımız aşağıdaki gibidir.

LCD.h dosyası:

```
#ifndef __LCD_H
#define __LCD_H

#ifdef __cplusplus
 extern "C" {
#endif

// LCD bağlantısında kullanılacak pinlerin tanımlamaları
#define LCD_DATA_PORT                   P1OUT
#define LCD_DATA_PORT_DIR               P1DIR
#define LCD_DATA_PORT_SEL               P1SEL
#define LCD_DATA_PORT_SEL2              P1SEL2
#define LCD_CONTROL_PORT                P2OUT
#define LCD_CONTROL_PORT_DIR            P2DIR
#define LCD_CONTROL_PORT_SEL            P2SEL
#define LCD_CONTROL_PORT_SEL2           P2SEL2
#define LCD_RS_BIT                      BIT7
#define LCD_EN_BIT                      BIT6
```

```
// LCD sürme işleminde kullanılan makrolar
#define       EN_AC_KAPA()    LCD_EN(1),LCD_EN(0)
#define       LCD_RS(x)      ( (x) ? (LCD_CONTROL_PORT |= LCD_RS_BIT) :
(LCD_CONTROL_PORT &= ~LCD_RS_BIT) )

#define       LCD_EN(x)      ( (x) ? (LCD_CONTROL_PORT |= LCD_EN_BIT) :
(LCD_CONTROL_PORT &= ~LCD_EN_BIT) )

// LCD kütüphanesinde kullanılan fonksiyon prototipleri
void LCD_Komut_Yaz(unsigned char);        // LCD ye komut göndermeye yarar
void LCD_Temizle(void);                   // LCD ekranı temizler
void LCD_Yazi_Yaz(const char*);           // LCDye string ifade yazar
void LCD_Git_XY(char,char);               // LCDde kursörü konumlandırır
void LCD_Ayarla(void);                    // LCD başlangıçayarları yapılır
void LCD_Karakter_Yaz(char);              // LCDye tek karakter yazmak için kullanılır

#ifdef __cplusplus
}
#endif

#endif /*__LCD_H */
```

LCD.c Dosyası:

```
#include <msp430.h>               // msp430 başlık dosyası ekleniyor
#include "LCD.h"                  // LCD başlık dosyası       ekleniyor
// LCDye karakter yazan fonkisyon
void LCD_Karakter_Yaz(char veri)
{
  unsigned char bSayac;
  LCD_RS(1);
  for(bSayac=0;bSayac<40;bSayac++);
  LCD_DATA_PORT = (LCD_DATA_PORT & 0x0F)|((veri) & 0xF0);
  EN_AC_KAPA();
  LCD_DATA_PORT = (LCD_DATA_PORT & 0x0F)|(veri<<4 & 0xF0);
  EN_AC_KAPA();
}
// LCDye komut yazan fonkiyon
void LCD_Komut_Yaz(unsigned char komut)
{
  unsigned char bSayac;
  LCD_RS(0);
  for(bSayac=0;bSayac<40;bSayac++);
  LCD_DATA_PORT = (LCD_DATA_PORT & 0x0F)|((komut) & 0xF0);
  EN_AC_KAPA();
  LCD_DATA_PORT = (LCD_DATA_PORT & 0x0F)|(komut<<4 & 0xF0);
  EN_AC_KAPA();
}
```

```
// LCD ekranı temizleyip kursor pozisyonunu başlengıça getriren fonksiyon
void LCD_Temizle(void)
{
  unsigned int  bSayac;
  LCD_Komut_Yaz(0x01);
  for(bSayac=0;bSayac<1000;bSayac++);
}
// LCDye yazı yazdıran fonksiyon
void LCD_Yazi_Yaz(const char* yazi)
{
  while(*yazi)
  LCD_Karakter_Yaz(*yazi++);
}
// LCD kursörü istenilen yere götüren fonksiyon
void LCD_Git_XY(char x, char y)
{
  if(x==1)
  LCD_Komut_Yaz(0x80+((y-1)%16));
  else
  LCD_Komut_Yaz(0xC0+((y-1)%16));
}
// LCD başlangıç ayarlarını yapan fonksiyon
void LCD_Ayarla()
{
  unsigned int bSayac1,bSayac2;
  LCD_DATA_PORT_DIR      |=     BIT4 + BIT5 + BIT6 + BIT7;
  LCD_DATA_PORT_SEL      &= ~(BIT4 + BIT5 + BIT6 + BIT7);
  LCD_DATA_PORT_SEL2     &= ~(BIT4 + BIT5 + BIT6 + BIT7);
  LCD_CONTROL_PORT_DIR   |=     LCD_EN_BIT + LCD_RS_BIT;
  LCD_CONTROL_PORT_SEL   &= ~(LCD_EN_BIT + LCD_RS_BIT);
  LCD_CONTROL_PORT_SEL2  &= ~(LCD_EN_BIT + LCD_RS_BIT);
  LCD_DATA_PORT = 0x0F;
  LCD_CONTROL_PORT = 0x00;
  for(bSayac1=0;bSayac1<1000;bSayac1++)
  for(bSayac2=0;bSayac2<500;bSayac2++);
  LCD_RS(0);
  LCD_EN(0);
  for(bSayac1=0;bSayac1<20000;bSayac1++);
  LCD_Komut_Yaz(0x28);  // 4 Bit, çift Satır LCD
  LCD_Komut_Yaz(0x0C);  // imleç Gizleniyor
  LCD_Komut_Yaz(0x06);  // Sağa doğru yazma aktif
  LCD_Komut_Yaz(0x80);  // LCD Birinci Satır Konumunda
  LCD_Komut_Yaz(0x28);  // 4 Bit, çift Satır LCD
  LCD_Temizle();     // Ekran Temizleniyor
}
```

Yukarıda LCDmize ait kütüphane dosyalarının kodları görülmektedir. Görüldüğü gibi kütüphanemiz fonksiyonel biçimdedir. Yani gerekli her işlem için bir fonkisoyn bulunmaktadır. Örneğin LCD'ye bir komut yazmak için LCD_Komut_Yaz fonksiyonunu kullanmak yeterlidir. Ya da LCDmize bir karakter yazmak için LCD_Karakter_Yaz fonksiyonunu kullanmamız yeterlidir. Bu sayede kütüphaneyi uygulamalarımıza ekleyerek rahatlıkla kullanıp istediğimiz karakterleri yazdırabiliriz.

UYGULAMA 3.1 Karakter LCD Kullanımı

Kütüphane kodlarımızı hazırladıktan sonra LCDmizi denemek için yazı yazan bir uygulama gerçekleştirelim. Yeni bir proje açalım ve oluşturduğumuz LCD.h ve LCD.c dosyalarını projemizin bulunduğu klasöre kopyalıyalım. Normalde CCS kopyaladığımız dosyaları otomatik olarak projeye dahil eder. Eğer dahil etmez ise proje ekranında projemizin ismine sağ tıklayıp Add Files butonuna tıklayıp dosyalarımızı elle ekleyebiliriz. Add Files butonuna tıklayıp dosyaları seçtikten sonra çıkan ekranda ok tuşuna basıp dosyalarımızı ekleyebiliriz.

Kütüphane dosyalarımızı ekledikten sonra uygulamamızın ana programına geçelim. Uygulamamızın ana programı aşağıdaki gibidir.

```
#include <msp430.h>                    // MSP430 başlık dosyası
#include "LCD.h"                       // LCD başlık dosyası

void main(void) {
   WDTCTL = WDTPW | WDTHOLD;           // Watchdog timeri durdur.
   LCD_Ayarla();                       // Başlangıçta LCD ayarlarını yap.
   LCD_Git_XY(1,3);                    // Kursörü 1.satır 3.sütuna götür.
   LCD_Yazi_Yaz("Erhan YILMAZ");       // Ekrana Erhan YILMAZ yazdır.
   LCD_Git_XY(2,4);                    // Kursörü 2.satır 4.sütuna götür.
   LCD_Yazi_Yaz("LCD DENEME");         // Ekrana LCD DENEME yazdır.
   __bis_SR_register(LPM4_bits);       // Denetleyiciyi uyku moduna geçir.
}
```

Görüldüğü gibi uygulama ana programımız çok basittir. Kodun başında MSP430 ve LCD için gerekli başlık dosyaları eklenir. Burada msp430.h dosyası MSP430 denetleyiciler için kullanılan genel bir başlık dosyadır. İçerisinde MSP430 denetleyicilere ait değişken tanımlamaları sabit değerler vb. bilgiler bulunmaktadır.

Uygulamayı çalıştırdığımızda başlangıçta denetleyici ve LCD ayarları yapılır. Sonrasında LCD kursörü yani imleci uygun yerlere getirilerek ekrana deneme yazıları yazdırılır.

Şekil 3: Uygulma 3.1'e Ait LCD Ekran Görüntüsü

Şekil 3'te uygulamamıza ait ekran görüntüsü görülmektedir. Kodları kartımıza yüklemeden önce kullanmadığımız birimlerin anahatarlarını off yapmakta fayda var. Ayrıca LCD ve aydınlatmasını anahtarlayan sw2 anahtalarını on konumuna ve sw4, sw15 anahtlarını off konumuna getirip led ve butonları iptal etmeniz gereklidir. Gerekli ayarlamaları yapıp kodlarımızı karta yükledikten sonra yazdırdığımız yazıların istediğimiz şekilde LCD ekran üzerinde oluştuğunu görebilirsiniz.

Piyasada bir çok LCD üreteci firma olduğundan kullanılan LCDlerde kontrast ayarı ile ilgili sorunlar olabilir. Uygulamayı denerken kontrast sorunu yaşarsanız kontrast ayarı için kullanılan R1(2.2 KOhm) direncinin değerini değiştirebilirsiniz. Aynı şekilde bende uygulamayı denerken kontrast sorunu yaşadım ve 1-2 denemeden sonra 1.5 KOhm bir direnç ile sağlıklı bir şekilde görüntüyü elde ettim.

Uygulama 3.2 Özel Karakter Gösterimi

Bu uygulamamızda ise karakter LCDmizde Türkçe karakterleri ve istediğimiz özel sembollleri, işaretleri görüntüleyeceğiz. Bilindiği gibi ascii tablosunda Türkçe karakterler yer almamaktadır. Ama bunun yerine HD44780 entegresinde 8 karakterlik bir özel hafıza bulunmaktadır. Bu hafızaya 5x7 matris şeklinde istediğimiz karakter yada semboller tanımlayarak uygulamalarımızda kullanabilir. Türkçe karakterler Ç, ç, Ğ, ğ İ, ı, Ö, ö, Ş, ş Ü, ü olmak üzere 12 adettir. HD44780, 8 karakter desteği verdiği için bu karakterleri büyük harf yada küçük harf olarak tanımlayıp kullanabiliriz. Uygulamamıza ait kodlar aşağıdaki gibidir.

```
#include <msp430.h>                 // MSP430 başlık dosyası
#include "LCD.h"                    // LCD başlık dosyası
void Ozel_Karakterler();            // Özel karakterler fonkisyon prototipi

void main(void) {
   WDTCTL = WDTPW | WDTHOLD;        // Watchdog timeri durdur.
   LCD_Ayarla();                    // Başlangıçta LCD ayarlarını yap.
   Ozel_Karakterler();              // Özel karakterleri yükle.
   LCD_Git_XY(1,1);                 // Kursörü 1.satır 1.sütuna götür.
   LCD_Yazi_Yaz("T");               // Ekranda T yaz.
   LCD_Karakter_Yaz(5);             // Ekranda Ü yaz.
   LCD_Yazi_Yaz("RK");              // Ekranda RK yaz.
   LCD_Karakter_Yaz(0);             // Ekranda Ç yaz.
   LCD_Yazi_Yaz("E KARAKTER");      // Ekranda E KARAKTER yaz.
   LCD_Git_XY(2,5);                 // Kursörü 2. satır 5. sütuna götür.
   LCD_Karakter_Yaz(0);             // Ekranda Ç yaz.
   LCD_Karakter_Yaz(1);             // Ekranda Ğ yaz.
   LCD_Karakter_Yaz(2);             // Ekranda İ yaz.
   LCD_Karakter_Yaz(3);             // Ekranda Ö yaz.
   LCD_Karakter_Yaz(4);             // Ekranda Ş yaz.
   LCD_Karakter_Yaz(5);             // Ekranda Ü yaz.
   LCD_Karakter_Yaz(6);             // Ekranda ı yaz.
   LCD_Karakter_Yaz(7);             // Ekranda santigrat işeretini yaz.
   LPM4;                            // Denetleyiciyi uyku moduna geçir.
}
```

```
// LCD'ye özel Türkçe karakterleri ve santigrat işaretini tanıtan fonksiyon
void Ozel_Karakterler()
{
// CGRAM 1.Adrese ; 'Ç' Karakteri
LCD_Komut_Yaz(0x40);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x11);
LCD_Karakter_Yaz(0x10);LCD_Karakter_Yaz(0x10);
LCD_Karakter_Yaz(0x11);LCD_Karakter_Yaz(0x0E);
LCD_Karakter_Yaz(0x04);LCD_Karakter_Yaz(0x00);
// CGRAM 2.Adrese ; 'Ğ' Karakteri
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x00);
LCD_Karakter_Yaz(0x0F);LCD_Karakter_Yaz(0x10);
LCD_Karakter_Yaz(0x13);LCD_Karakter_Yaz(0x11);
LCD_Karakter_Yaz(0x0F);LCD_Karakter_Yaz(0x00);
// CGRAM 3.Adrese ; 'I' Karakteri
LCD_Karakter_Yaz(0x04);LCD_Karakter_Yaz(0x00);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x04);
LCD_Karakter_Yaz(0x04);LCD_Karakter_Yaz(0x04);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x00);
// CGRAM 4.Adrese ; 'Ö' Karakteri
LCD_Karakter_Yaz(0x0A);LCD_Karakter_Yaz(0x00);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x11);
LCD_Karakter_Yaz(0x11);LCD_Karakter_Yaz(0x11);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x00);
// CGRAM 5.Adrese ; 'S' Karakteri
LCD_Karakter_Yaz(0x0F);LCD_Karakter_Yaz(0x10);
LCD_Karakter_Yaz(0x10);LCD_Karakter_Yaz(0x0E);
LCD_Karakter_Yaz(0x01);LCD_Karakter_Yaz(0x05);
LCD_Karakter_Yaz(0x1E);LCD_Karakter_Yaz(0x00);
// CGRAM 6.Adrese ; 'Ü' Karakteri
LCD_Karakter_Yaz(0x0A);LCD_Karakter_Yaz(0x00);
LCD_Karakter_Yaz(0x11);LCD_Karakter_Yaz(0x11);
LCD_Karakter_Yaz(0x11);LCD_Karakter_Yaz(0x11);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x00);
// CGRAM 7.Adrese ; 'i' Karakteri
LCD_Karakter_Yaz(0x00);LCD_Karakter_Yaz(0x00);
LCD_Karakter_Yaz(0x0C);LCD_Karakter_Yaz(0x04);
LCD_Karakter_Yaz(0x04);LCD_Karakter_Yaz(0x04);
LCD_Karakter_Yaz(0x0E);LCD_Karakter_Yaz(0x00);
// CGRAM 8.Adrese ; '°' Sembolü
LCD_Karakter_Yaz(0x0C);LCD_Karakter_Yaz(0x12);
LCD_Karakter_Yaz(0x12);LCD_Karakter_Yaz(0x0C);
LCD_Karakter_Yaz(0x00);LCD_Karakter_Yaz(0x00);
LCD_Karakter_Yaz(0x00);LCD_Karakter_Yaz(0x00);
}
```

Uygulama kodlarında görüldüğü gibi başlangıçta özel karakterler isimili fonksiyon ile özel karakterler sırasıyla HD44780nin karakter ram alanına kayıt edilmiştir. Sonrasında uygulamada aynı karakter yazdırma işleminde olduğu gibi sırasına göre istenilen karakter ekrana yazdırılmıştır. Ayrıca bu uygulamada diğerlerinden farklı olarak LPM4 komtusu kullanılmıştır. Bu komut msp430.h dosyasında tanımlıdır ve _bis_SR_register(LPM4_bits) komutu ile aynı işlevi görmektedir. Yani denetleyici LPM4 tasarruf moduna sokar.

Şekil 4: Uygulama 3.2 LCD Ekran Görüntüsü

Uygulama 3.1'de olduğu gibi kart ayarlarını aynı şekilde yapıp kodları işlemcimize yükledikten sonra herhangi bir sorun yok ise şekil 4'te ki gibi LCD ekranda yazdırdıklarımız gelekcektir.

Böylece görüldüğü gibi ascii karakter tablosunda bulunmayan özel karakterleri yada görüntülemek istediğimiz sembollerimizi istediğimiz gibi tanımlayıp kullanabiliriz.

Bu modül kapsamında karakter LCDler hakkında bilgi verilmiş ve MSP430 geliştirme kartı üzerinde uygulaması yapılarak MSP430 denetleyicilerle kullanımı gerçekleştirilmiştir. Böylece diğer modüllerde ki uygulamalarda istediğimiz değerleri görüntülemek için karakter LCD kullanarak uygulamalarımız daha kolay anlaşılır hale getirebiliriz.