**GÖMÜLÜ ve GERÇEK ZAMANLI SİSTEMLER**

**Final Projesi Raporu**

**Konu: Dot Matrix Gösterimi**

Hazırlayanlar:

05-11-44 Onurhan ÇELİK

05-12-240 Onur TEMUROĞLU

Sunulan:

Yrd. Doç. Dr. Mustafa ENGİN

Tarih: 11.01.2016

İÇİNDEKİLER

[1. GİRİŞ 2](#_Toc440450676)

[2. KULLANILAN YAZILIM ORTAMLARI 2](#_Toc440450677)

[3. Stellaris LM4F120 LaunchPad TANITIMI 2](#_Toc440450678)

[4. MAX7219 TANITIMI 3](#_Toc440450679)

[5. GERÇEKLEŞTİRİM 4](#_Toc440450680)

# GİRİŞ

Projemizin konusu 8x8’lik dot matrix’te karakterlerin gösterimi ve klavyeden alınan giriş verilerinin bu dot maxtrix’te gösterimidir. Bu proje için Cortex M4 işlemcisini içeren Texas Instruments’in Stellaris LM4F120 LaunchPad’i ile MAX7219 LED Display Driver’ını kullandık.

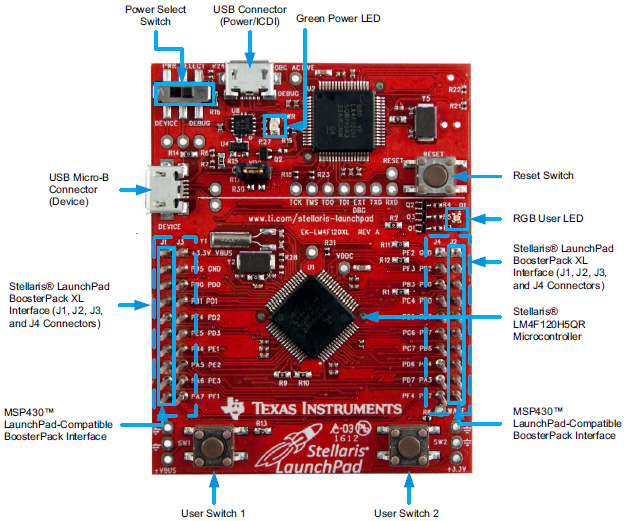
# KULLANILAN YAZILIM ORTAMLARI

LaunchPad için yaptığımız gömülü sistemin yazılımını Energia platformunda yazdık. Klavyeden girdi verilerinin okunmasını ve bu verilerin seri haberleşme yolu ile LaunchPad’e aktarılmasını sağlayan form uygulamasını ise Microsoft Visual Studio 2015 ortamında Visual C# dili ile gerçekleştirdik.

# Stellaris LM4F120 LaunchPad TANITIMI

Stellaris LM4F120 LaunchPad’i, Texas Instruments’in ARM® Cortex™-M4F tabanlı mikrodenetleyicileri için düşük maliyetli bir değerlendirme platformudur. Bizim kullandığımız LaunchPad’in modeli ise EK-LM4F120XL’dir. Bu LaunchPad’in ekstra sunduğu özellikler ise custom uygulamalar için kullanıcı butonları ve RGB LED sunmasıdır.

Şekil 1’de LM4F120’nin genel görünümünü görünmektedir. 2 farklı çalışma modu vardır. Debug ve Device(Cihaz) olmak üzere. Bu modlar arasındaki geçiş şekildeki “Power Select Switch”i ile yapılır. İki farklı Micro USB girişi vardır. Bir tanesi güç kaynağına bağlanır, diğeri ise diğer cihaza bağlanır. Yazılım yüklemek için Debug modunda çalıştırmak gerekir.

  
**Şekil 1** – Stellaris LM4F120 LaunchPad Evolution Board

LaunchPad’i iki ayrı bölüm olarak inceleyebiliriz. Üst kısım birinci bölümdür ve LaunchPad’in Debug işlemleri içindir. Bu bölümde bir de Reset Switch butonu vardır. Programın ilk çalışma anına geri dönmesini sağlar.

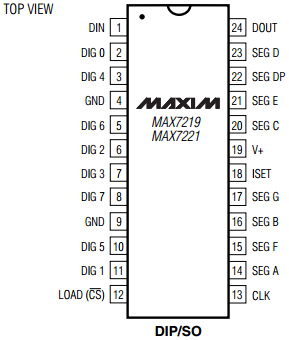
İkinci kısım olan alt kısımda ise RGB LED, iki adet kullanıcı butonları (User Switch 1-2) ve 40 adet bağlantı pinleri(J1,J2,J3,J4 Connectors) bulunmaktadır. Bu pinler genel olarak I2C, Serial UART ve SPI haberleşme protokolleri için kullanılırlar. Birçoğu veri giriş ve çıkışları için kullanılırken, bazıları Clock, CS(Chip Select), VCC ve Ground pinleri olarak kullanılmaktadırlar.

# MAX7219 TANITIMI

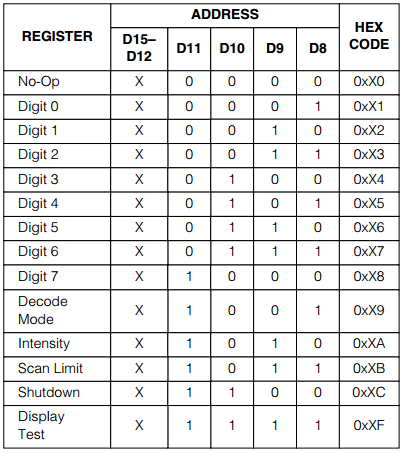
MAX7219 kompakt, seri giriş-çıkışlı, katotların ortak bir pine bağlı olduğu gösterim cihazlarıdır. Mikrodenetleyicilere LED gösterimler üzerinden bir arayüz sağlar. 64 LED’e kadar destekler.

Cihaz farklı modlar içermektedir. Bunlar; 150 µA’lik düşük-güç kapatma modu, analog ve dijital parlaklık ayarları, LED’lerin hangilerinin gösterileceğini ayarlayan scan-limit register modu, ve tüm LED’lerin çalışıp çalışmadığını test eden bir test modu içermektedir.

MAX7219’un pin konfigürasyonu Şekil 2’de bulunmaktadır.

  
**Şekil 2** – MAX7219 Pin Konfigürasyonu

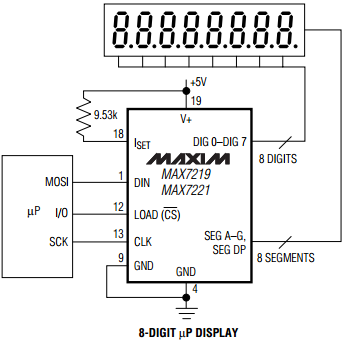
MAX7219’un yukarıda bahsettiğimiz farklı modları vardır. Bu modların ayarlanması kendi içinde tanımlanmış olan register adreslerine göre yapılmaktadır. Bunların opcode’larını Tablo 1’de görebilirsiniz.

**Tablo 1.** Register Adres Map  


Digit 0-Digit 7 arası olan modlar 8x8’lik LED matriste hangi sütuna hangi verilerin yazılacağını ayarlar. Diğer modlar da ilgili özellikleri gerçekleştirir. (parlaklık ayarı, kapatma modu vs.)

# GERÇEKLEŞTİRİM

MAX7219 üzerinde LM4F120 LaunchPad’ine bağlanması için gerekli çıkışlar bulunmaktadır. Bunlar VCC, GND(ground), DIN(Data Input), CS(Chip Select-LOAD) ve CLK(Clock) çıkışlarıdır. İlgili çıkışlara ait devre şeması Şekil 3’te görülmektedir.

  
**Şekil 3 –** MAX7219 Devresi

Verilerin aktarımı Serial Peripheral Interface(SPI) bağlantı standardıyla gerçekleşmektedir. SPI’da CLK, MISO(I/O), MOSI veri yolları bulunmaktadır. Burada MOSI veri yolu DIN çıkışıdır. MISO veri yolu ise CS çıkışıdır. SCK ise CLK pinidir. Bu çıkışlara karşılık gelen LM4F120’deki bağlantı pinleri birden fazladır. Bizim kullandıklarımız ise şu şekildedir;

* DIN çıkışı – **PB\_7**
* CS çıkışı – **PA\_3**
* CLOCK çıkışı – **PB\_4**

Projede istenilen durumlar, sıralı olarak rakamlar ile Türkçe alfabenin büyük ve küçük harfli olarak gösterilmesi ve aynı zamanda klavyeden alınan giriş karakterlerinin direkt olarak 8x8’lik LED matrixte gösterimidir.

Karakterlerin 8x8’lik LED matrixte yazdırılması için öncelikle hangi karakterler için hangi ledlerin yanacağını belirleyen bir çok boyutlu dizi oluşturduk. Bunu oluştururken kolaylık olması amacıyla ASCII numaralarından yararlandık ve ASCII numaraları sırasında diziyi (CH dizisi) oluşturduk. Ancak Türkçe karakterlerin (çÇ,ğĞ,ıİ,öÖ,şŞ,üÜ) ASCII numaraları bu diziye uygun olmadığı için ayrı şekilde Türkçe karakterler için de bir çok boyutlu dizi(TR\_CH dizisi) oluşturduk. Türkçe karakterlerin bazılarının ASCII kodu 255’den fazla olduğu için ve de Byte verisinin 0-255 arasında olmasından dolayı, verileri aldıktan sonra bu 255’i aşan Türkçe karakterlerin fonksiyonlarda yazımı için kolaylık olsun diye ASCII tablosunda kullanmadığımız 0-6 arasındaki numaralar ile bu Türkçe karakterleri yazdırdık.

Bu dizilerde(CH ve TR\_CH) karakterlerin kaç satır ve sütunda gösterileceğini içeren satır ve sütun değerleri ile ilgili ledlerden hangilerinin yanacağını belirten 5 farklı byte verileri bulunmaktadır. Biz karakterleri 8x8’lik matriste en fazla 5x8’lik olacak şekilde düzenledik. Bu yüzden dizide var olan veri byte’ları 5 adettir.

Diziden örnek verecek olursak “a” karakteri için dizideki bölüm şu şekildedir;

**“4, 8, B0100000, B1010100, B1010100, B1111000, B0000000”**

İlk iki değer ile “a” karakterinin 4 sütun ve 8 satırda var olacağını belirtmiş olduk. Sonraki 5 değer ise bu 4 sütunda byte olarak hangi ledlerin yakılacağını göstermektedir.

Dot Matrix’e SPI ile veri gönderimini(hem yazdırma hem ilgili komutla ayar yapmayı) Energia kütüphanesinde varolan shiftOut fonksiyonuyla gerçekleştirdik. “shiftOut” fonksiyonu 1 byte veriyi kaydırma yolu ile bir zamanda 1 bitini gönderir. En sağdan veya en soldan başlayarak gönderme işlemi gerçekleşir. Parametre ile ayarlanır bu gönderim yönü. Her bit data pin’ine gider, clock’un pulse olmasından sonra (high-low). Syntax gösterimi **“shiftOut(dataPin, clockPin, bitOrder, value)”** şeklindedir fonksiyonun. dataPin ve clockPin; MAX7219’u bağladığımız DIN ve CLK pinleridir. bitOrder parametresi ise verinin soldan veya sağdan gideceğini ayarlar, bu değer MSBFIRST veya LSBFIRST olabilir. Biz MSBFISRT kullandık, dizilerimizdeki verileri sondan başlayarak yaptığımız için. Value parametresi ise MAX7219’a gönderilecek olan komut veya değer verisidir. Öncelikle yapılacak işleme göre Opcode listesinden ilgili veri gönderilir shiftOut ile, daha sonra yine shiftOut ile yapılacak işlemin gerektirdiği değer gönderilir. Örneğin 3. Sütunun tüm ledlerini yakacaksak öncelikle ilk shiftOut çağrımında value parametresi “0x03” byte’ı olur ve 3. sütuna yazım olacağını belirtir, hemen ardından tekrar çağrılan shiftOut fonksiyonunda ise value değeri ledlerin hangilerinin yakılacağı verisidir. B11111111” şeklinde byte verisi ile tüm ledler yansın istemiş oluruz. Tabiki bu shiftOut fonksiyonlarının çağrımından önce ve sonra CS pininin HIGH ve LOW şeklinde pulse’lanması gerekir. Bu shiftOut fonksiyonlarıyla MAX7219’a veri gönderimine ait kod parçası örneğini aşağıda görebilirsiniz.

|  |
| --- |
| opcode = 0x03;  value = B11111111;  digitalWrite(cs, LOW);  shiftOut(data, clock, MSBFIRST, opcode);  shiftOut(data, clock, MSBFIRST, value);  digitalWrite(cs, HIGH); |

Hem Energia kaynak kodlarında(DotMatrixGosterimi dosyasındaki DotMatrixGosterimi.ino, MaxMatrix.cpp, MaxMatrix.h) hem de Visual C# kaynak kodlarını(dotMatrix/Form1.cs), kodların anlatımını açıklama satırlarıyla görebilirsiniz. Visual C#’da SerialPort sınıfından oluşturduğumuz nesnenin port numarasını “COM4” olarak girdik. Çünkü bizim bilgisayarımızda LM4F120 cihazımız COM4 portunu kullanmaktadır. Eğer farklı bir bilgisayarda Visual C# programı çalıştırılacaksa kendi port numaranızı COM4 yerine yazınız. Bahsettiğimiz kod parçası aşağıdaki şekildedir.

|  |
| --- |
| SerialPort sp;  sp = new SerialPort("COM4", 9600); *//sp SerialPort nesnesinin hangi porttan, kaçlık bir BaudRate ile //gideceğini belirttik.*  sp.Open(); *//Portu haberleşmeye açtık.* |