**Stellaris Launchpad Kullanılarak**

**LCD Ekrana Yazı Yazdırılması**

**Nedret Gegeoğlu**

**Kocaeli Üniversitesi**

**Bilgisayar Mühendisliği 2016 – 2017**

**1.1 Problem Tanımı:**

Bizden gömülü sistem programlamada LCD ekrana yazı yazdırmamız istenildi. Öncelikle malzemeleri temin edilip sorun parçalar halinde çözümlenmeye başlandı. İlk olarak yapmamız gereken sorunsuz bir şekilde Launchpad ile LCD arasındaki bağlantıları sorunsuz yerine getirmekti. Kurulum için çeşitli kaynaklardan yardım aldıktan sonra kurulumu tamamlayıp LCD ekran denemelerine başlanıldı.

Kurulum konusunda LCD ekran modülüne göre 8-bit yada 4-bit kurulum yapılıp ekran kullanılabilirdi.

* 8-bit kurulum: LCD; D0-D7 arası pinlerin kullanılması.
* 4-Bit kurulum LCD; D4-D7 arası pinlerin kullanılması.

8 bit kurulum yapıp, Kart üzerindeki A ve B portlarını kullandım.

* **A** Portu; Bu porta LCD üzerindeki RS ve Enable pinleri bağlandı.

**PA6: RS( Register Select)**

**PA7: E (Enable)**

* **B** Portu; Bu portta LCD üzerindeki erişim pinleri bağlandı (D0-D7)

**D0: PB0 D3: PB3 D6: PB6**

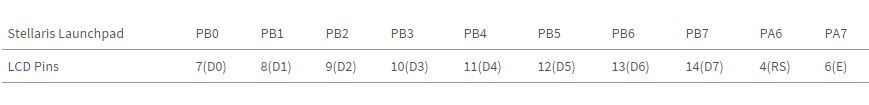
**D1: PB1 D4: PB4 D7: PB7**

**D2: PB2 D5: PB5**

Bunun haricinde LCD üzerinde kalan pinlerin bağlantıları:

**VSS: GND VDD: +5V V0: 10 K Pot**

**RW: GND A: +5V K: GND**



**Resim 1:** Pin bağlantıları

**1.2 Yapılan Araştırmalar:**

Öncelikle gömülü sistem konusunda açıkçası Türkçe kaynak bulmak pek kolay değil. Bu konular üzerinde fazla çalışma yapan, yada çalıştıklarını paylaşan pek Türk internet sitesi olduğunu söyleyemem. İngilizce kaynaklardan bahsedecek olursak yeterince konuya hakim, paylaşımcı forumlar mevcut. Karşılaştığım ilk ve çözmesi en baş ağrıtan problem **LCD initialize** kodlarında ki uyumsuzluktu:

Çoğu örnekte kullanılan **delay**ler farklı, **SysCtl Clock** ayarları değişken olduğu için bir standart tutturmak söz konusu değil. Bunun üzerine yapılan araştırmada kullanılan LCD modülüne göre delay değişken bekleme değerlerinin farklı olabileceğini gördüm. Bende bunun üzerine kendi kullandığım **1602A** LCD modülünün **Datasheet \*(1.1)** kaynağına ulaşıp Lcd üzerindeki pinlerin amaçları ve kullanım şekilleri üzerinde bilgi edindim. Bunun üzerine delay komutlarını güncelleyip artırarak sonunda LCD modülü üzerinde karakter basmayı başarabildim.

LCD modüle karakter basmak bir yana kelime yazdırmak için aynı işlemi üst üste gerçekleştirmek gerekiyordu. Bunun için Kullanılacak kelimelerde ki harflerin bulunduğu **ASCII** dizileri oluşturuldu: kou[11], ned[15]. Bu kelimeleri döngüler halinde ekrana basmayı başarabildim.

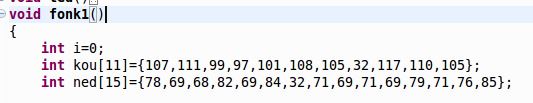
Alt alta 2 satırıda kullanırken, kart üzerinde alt satıra inmek için datasheet üzerinde bulunan bilgilerden yararlanıp **D4** pininin bu işleme yaradığı, kullanırken de **RS** ve **E** pinlerinin ‘**0’** durumunda olması gerekiyordu. Bunuda halledip iki satır halinde ilk iki işlemi yerine getirdim.

Kayan Yazı konusunda LCD modül için hazır fonksiyonlar mevcut olabilir fakat ben bunlara ulaşamadığım için sorunu kendimce çözümledim. İlk olarak **sağdan – sola** kaydırma konusunda **while** döngüsü içerisinde **kontrol** ve **baslangic** değişkenleri oluşturarak döngünün gerektiği kadar devam etmesini sağladım. **While** döngüsü içerisine **For** döngüsü girip **kontrol** değişkeninin olduğu kadar boşluk “ “ karakteri girmesini sağladım. Böylelikle hızlı bir şekilde imleç son sütuna ilerleyip kelimenin ilk harfini yazdırmaya hazır hale gelecekti. Daha sonra **harf** değişkeni oluşturarak 16 sayısından kontrol değişkenini çıkararak kaç harf eklememiz gerektiğini bulup **For** döngüsü içerisinde dizi üzerinden ekleme yapıldı. **Kontrol** değişkeni bir azaldı ve bir sonraki işlemde **harf** değişkeni bir artarak kayan yazı yazdırma işlemi tamamlandı.

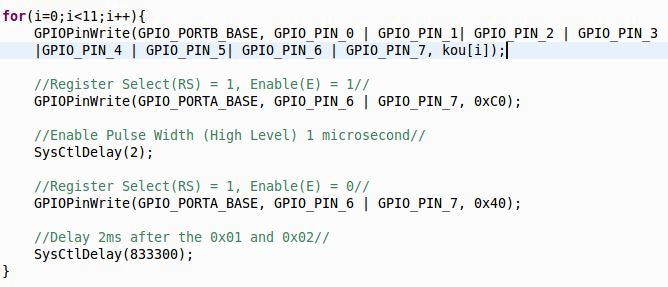
**Soldan – Sağa** kaydırma işlemi ise daha farklı bir mantıkla işliyordu; Burda yazdırılacak kelime karakteri kadar **While** döngüsü içerisinde işlem gerçekleniyor ve **kontrol** değişkeni ile bu hesap gerçekleniyordu. **While** içerisinde kullanılan **For** döngüsünde ise **i** değişkeni **son** değişkeni ile son karakterden başlayıp **hedef** değişkenine kadar azalarak ilerliyordu. Bunun sebebi soldan – sağa kayan yazı üzerinde son karakterden başlayıp son karakter ve önceki karakterler yazılmasıydı. Böylelikle kayan yazı problemide bu şekilde çözümlendi.

**1.3 Tasarım**

Kullanılan fonksiyonların bazı kod çıktıları:



**Resim 2:** Hazırlanan ascii tablosu dizileri



**Resim 3:** koü dizisini LCD modüle bastırma işlemi.

**1.4 Genel Yapı:**

Proje üzerinde bizden istenen dört farklı işlem için dört farklı fonksiyon hazırlandı.

* **Fonk1() :** Bu fonksiyonda LCD modül üzerinde üst satırda **kocaeli üni**, alt satırda **NEDRET GEGEOGLU** yazdırıldı.
* **Fonk2() :** Bu fonksiyonda LCD modül üzerinde üst satırda **NEDRET GEGEOGLU**, alt satırda **kocaeli üni** yazdırıldı.
* **Fonk3() :** Bu fonksiyonda sağdan – sola kaydırma işlemi beraberinde **kocaeli üni** ve **NEDRET GEGEOGLU** yazdırıldı.
* **Fonk4() :** Bu fonksiyonda soldan – sağa kaydırma işlemi beraberinde **kocaeli üni** ve **NEDRET GEGEOGLU** yazdırıldı.
* **Ekran\_sifirla() :** Bu fonksiyonda LCD modül initialize işlemleri gerçeklenildi ve LCD kullanıma hazır hale getirildi.
* **Tazele() :** Bu fonksiyonda **fonk3()** ve **fonk4()** üzerinde gerçeklenen kaydırma işlemleri sırasında ekran yenilenmesi için delay süreleri kısaltılarak ekran güncellemeleri gerçeklenildi.



**Resim 4:** LCD modülü çalışması

**1.5. Referanslar:**

1.1.LCD Datasheet :

[https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf](https://www.openhacks.com/uploadsproductos/eone-1602a1.pdf%20)

1.2.Texas Instruments Ti e2e Community :

[https://e2e.ti.com/support/microcontrollers/tiva\_arm/f/908/t/279206#pi239031350=2](https://e2e.ti.com/support/microcontrollers/tiva_arm/f/908/t/279206%23pi239031350=2)