EN AZ SAYIDA BOZUK PARA VERME PROBLEMİ

Şeyma Nur MUTLU ve Melike OĞUZ

*Bilgisayar Mühendisliği Kocaeli Üniversitesi* {170201004 & 170201028}@kocaeli.edu.tr

***ÖZET***

***“En Az Sayıda Bozuk Para Verme Problemi" adlı programda bir otomat makinesinin minimum sayıda bozuk para üstü vererek çalışması beklenmektedir. Butonlar, LCD ekran ve Seri Port bağlantılarıyla veri giriş çıkışı yaparak çalışır.***

***GİRİŞ***

En Az Sayıda Bozuk Para Verme Problemi projesinde butonlar ile ürün sayıları kontrolleri, atılan para miktarı kontrolleri gibi işlemler yapılırken LCD ekrandan ve seri porttan ise bu işlemlerin yazdırması yapılır.

Program, ürünler katarını parse ederek başlar ve ilgili bilgileri liste adındaki struct’ın elemanlarına yerleştirir bu aşamadan sonra işlemler structlar içerisindeki bilgilerden yapılmaya devam eder. Butonlar üzerinden okunan bilgiler doğrultusunda programda matematiksel işlemler yapılarak bilgiler güncel tutulur. Bu işlemlerin doğruluğu seri port aracılığıyla kontrol edilir. Sonrasında program içerisinde üretilen bir random sayı ile para sıkışma olayı kontrol edilir. Para sıkışmadığı durumda kasadaki para miktarlarına ve verilecek olan para üstüne uygun şekilde bozuk paraları verir. Para sıkıştığı veya kasadaki para miktarının para üstü vermeye yetmediği durumda LCD ekranda durum bilgisini verir.

***2.Temel Bilgiler***

Program, C dilinde geliştirilmiş olup geliştirme ortamı Tiva C TM4C123G elektronik kartıdır. Programın yazılımında debian isletim sisteminin bir dağıtımı olan “debian-kou-embedded-v$” işletim sistemi kullanılmış olup debug ve release işlemleri “Eclipse IDE” ile yapılmıştır.

**Devre kurulumunda;**

Geliştirme kartı olarak Texas Instruments Tiva C Series TM4C123G LaunchPad Evaluation,

LCD olarak “Qapass 1602 LCD”

1 adet kırmızı, 1 adet yeşil LED,

2 adeti 330 Ohm, 10 adeti 10 kOhm olmak üzere 12 adet direnç,

10 adet harici, 1 adet dahili buton kullanılmıştır.

***3. Tasarım***

Proje aşağıdaki başlıklar altında geliştirilimiştir.

**3.1 Yazılım Tasarımı**

**3.1.1 Fonksiyonlar**

**Portların Aktifleştirilmesi**

* initPortE()
* initPortB()
* initPortD()
* initPortF()

**LCD Ekran Kontrolü**

* Lcd\_init()
* Lcd\_Komut
* Lcd\_Putch
* Lcd\_Goto
* Lcd\_Temizle
* Lcd\_Puts

**LED Kontrolü**

* turnOnRed()
* turnOffRed()
* turnOnGreen()
* turnOffGreen()
* turnOnLed()

**Buton Kontrolü**

* moneyButtonCounter
* productsButtonCounter
* productsValue

**Seri Port Ayarı:**

* initUARTstdio()

**Random Sayı Üreteci**

* generateRandom

**3.1.2 Genel Değişkenler**

struct allPrice{

float price;

char currency[6];

};

struct product{

char productID;

char productAd[50];

int productStock;

struct allPrice price;

};

Yapısı her bir satırı ve bilgileri doğru şekilde kullanabilmek ve depolayabilmek adına oluşturulmuş olan yapı.

* #define pin0 0x01
* #define pin1 0x02
* #define pin2 0x04
* #define pin3 0x08
* #define pin4 0x10
* #define pin5 0x20
* #define pin6 0x40
* #define pin7 0x80

DATA ve DIR fonksiyonlarını kullanırken kolaylık sağlaması açısından isimlendirme sistemi yapılmıştır.

volatile unsigned long delay : port aktive ederken ve program içerisindeki döngülerde kullanılacak değişken.

***4.Karşılaşılan Sorunlar ve Çözümleri***

**4.1 Kütüphane Ekleme Sorunu**

Kodu yazmaya başladığımızda ek kütüphanelere ihtiyaç duyduk ve ekleme yapmamız gerekti, aynı işlemleri bir bilgisayarda yapınca ekleniyor fakat başka bir bilgisayarda işe yaramıyordu. Bu sırada başka kodları yazmaya devam ettik ve en sonunda Suhap Hoca’ya sorunca bilgisayar ve işletim sistemi arasında bir uyuşmazlık olduğu için bu hatayı aldığımızı ve çözümünün mümkün olmadığını söyledi.

**4.2 Buton Bağlantıları ve Çalışmaları**

Buton bağlantı ve kodlarını 4 tane buton için yaptığımızda çalışıp 6 tane buton için birebir aynı işlemleri yaptığımızda çalışmaması üzerine 1 hafta boyunca butonları bir çok farklı şekilde bağlayıp tekrar tekrar denedik en sonunda Gömülü Sistemler Laboratuvarına giderek durumu anlattık, beraber butonları tekrar bağlayıp kodu gözden geçirdik, hepsinin doğru olmasına rağmen kodumuz çalışmadı. En sonunda 6.butonu geliştirme kartı üzerindeki PF0 pinine bağlı olan buton olarak kullandık.

**4.3 Geliştirme Kartından Elimizde Bir Tane Olması**

Kartın bir tane olduğu için projeyi yaparken yanyana olmamız gerekiyordu fakat bu durum daha sonraları zaman problemi yarattı ve bilgisayarda sesli olarak Discord’tan konuşurken aynı anda telefondan görüntülü olarak devreye baktık.

**4.4 Xcode IDE’de Çalışan Kodun Sanal Makine Üzerindeki Eclipse IDE’de hata vermesi**

Bu hatayı almamızın sebebi de bilgisayar ve işletim sistemi arasında bir uyuşmazlık olmasıymış, Code Composer kullanmamız tavsiye edildi fakat şimdiye kadar kurduğumuz sistemler de Code Composerda istediğimiz gibi çalışmayınca bu sorunu kodun bazı kısımlarını fonksiyondan main içerisine alarak ve daha az pointer kullanarak çözmeye çalıştık

**4.5 Kart Üzerinde TXT’den Okuma Yapılması**

Bir hafta boyunca araştırmalarım üzerine EEPROM kullanılarak okuma yapılabileceğini fakat istediğimiz bir dosyayı EEPROM içerisine istediğimiz adrese yerleştirmenin kart üzerinde bir dosyalama sistemi olmasından dolayı mümkün olmayacağını öğrendik, proje isterler içeriği de bu yönde değiştirildi.

**4.6 Kart Üzerinde Seri Porttan Okuma Yapılması**

TXT bilgilerini seri port üzerinden okuyarak almamız istenildiğinde bu durumu araştırmaya başladık, okuduklarımız ve gördüklerimiz seri porta yazma durumuydu ve daha sonrasında proje isterlerinin içeriği de bu yönde değiştirildi.

**4.7 Random Sayı Üretilmesi**

Projeyi ilk yazmaya başladığımızda time.h kütüphanesini projeye import edip buradan random sayı üretmeyi düşündük kütüphane ekleme sıkıntısından dolayı koda uyguladığımızda çalışmadı, biz de bir fotodiyot yardımıyla o anda gelen değeri kullanarak rastgele sayı üretmeyi çözmeyi düşündük fakat ilerleyen zamanlarda aynı işlemleri, başka bir sanal makine üzerine uygulayınca çalıştı. Random sayı büyük değerler gelebiliyordu, gelen sayıların modunu alarak istediğimiz aralığı elde ettik.

***5.Kazanımlar***

1-Texas Instruments Tiva C Series TM4C123G LaunchPad Evaluation kartını daha efektif kullanmayı,

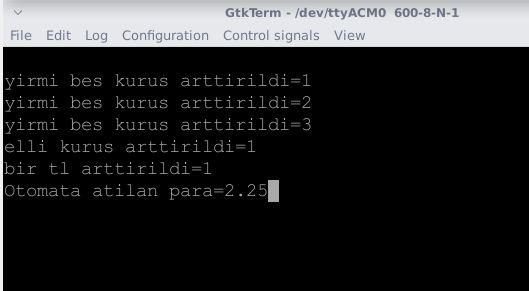
2-Harici bir butondan pull-up direnci yardımı ile veri almayı,

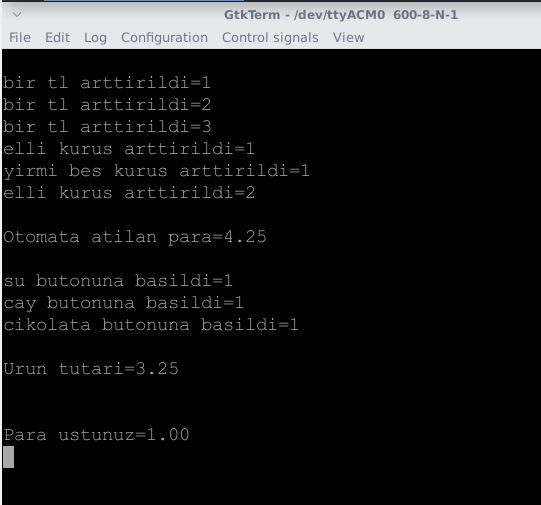
3-LCD ekranı kullanmayı ve bağlamayı,

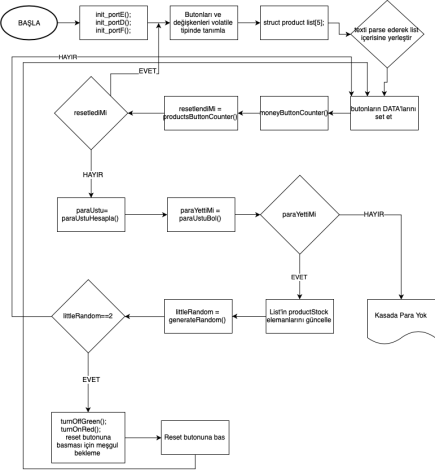
4-Harici bir kartta debug yapmayı.

5-Texas Instruments Tiva C Series TM4C123G LaunchPad Evaluation kartının dosya sistemi olmadığını,

6-Seri port üzerinde işlem yapmayı

**6-Ekran Görüntüleri**

******

**7- Akış Şeması**

***8.Portlar***

Projede B,D,E,F portlarını kullandık

**LCD PIN**

VSS +3.3V

VDD GND

V0 Potansiyometre’nin orta pini

RS B0

RW GND

E B1

D4 B4

D5 B5

D6 B6

D7 B7

A +3.3V

K GND

**Potansiyometre PIN**

Sol GND

Middle V0

Right +3.3V

**LED PIN**

E0 Kırmızı LED

E1 Yeşil LED

**Buton PIN**

E2 yirmiBesKurusButton

E3 elliKurusButton

E4 birTlButton

E5 endButton

PB0 suButton

PB1 cayButton

PB4 kahveButton

PB5 cikolataButton

PB6 biskuviButton

PB7 resetButton

***9.Kaynakça***

[1] <http://www.mcu-turkey.com/?s=lcd>

[2]<https://e2e.ti.com/support/tools/ccs/f/81/t/292491>

[3]<https://e2e.ti.com/support/tools/ccs/f/81/t/272610>

[4]<https://e2e.ti.com/support/processors/f/791/t/650990>

[4]<https://e2e.ti.com/support/tools/ccs/f/81/t/612994?CCS-Adding-external-library>

[5]<https://e2e.ti.com/support/microcontrollers/other/f/908/t/514565>

[6]<https://e2e.ti.com/support/tools/ccs/f/81/t/363319?stdio-h-download>

[7]<http://e2e.ti.com/support/microcontrollers/other/f/908/t/403695>

[8][https://github.com/MaxKox/Simple\_File\_System](https://github.com/Max-Kox/Simple_File_System)

[9]https://www.geeksforgeeks.org/convert-floating-point-number-string/