

KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
TASARIM PROJESİ



ANİMASYONLU DİJİTAL SAAT

AYŞE NUR DURMUŞ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
ANABİLİM DALI

BAHAR DÖNEMİ
2013-2014 YILI

KARADENİZ TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
TASARIM PROJESİ

ANİMASYONLU DİJİTAL SAAT

AYŞE NUR DURMUŞ
244035

DANIŞMAN: DOÇ. DR. MUSTAFA ULUTAŞ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
ANABİLİM DALI

BAHAR DÖNEMİ
2013-2014 YILI

ÖNSÖZ

Proje konusu bu konularla ilgilenenler için bir başlangıç niteliği taşımaktadır. Belli başlı donanımların kullanımı ve Arduino' nun programlanması hakkında sizlere yol gösterebilir.

Projede bana destek veren ve yardımda bulunan değerli hocalarıma teşekkür eder, saygılarımı sunarım.

AYŞE NUR DURMUŞ

TRABZON - 2014

İÇİNDEKİLER

ÖNSÖZ	i
ÖZET	iii
1. GİRİŞ	1
2. ARDUİNO MİMARİSİ	2
2.1. Arduino ATmega328 Bellek Alanları.....	2
3. STANDARTLAR VE KISITLAR	3
4. BENZER ÇALIŞMALAR	4
5. ÖNERİLEN YÖNTEM.....	5
5.1. Arduino LCD Bağlantıları.....	5
5.2. Arduino Kayan Saat Uygulaması.....	7
6. DENEYSEL SONUÇLAR.....	10
7. KAYNAKLAR.....	11

ÖZET

Projede amaç, animasyonlu dijital saat yapımı ve bunun gösterilmesi işlemidir. Programlanacak cihaz olarak Arduino kullanılmıştır. Arduino' nun kendine özel bir programlama dili vardır. Bu dil oldukça basit anlaşıldır. Bu nedenle programlama açısından bir zorluk yaşanmamıştır.

Arduino programlama dili ile gerekli saat, dakika ve saniye hesaplamaları yapılmıştır. Animasyon katması için yazıları kaydıracak gerekli kodlar eklenmiştir. Ardından da kod yüklenmiş ve proje çalışır hale getirilmiştir.

1. GİRİŞ

Projenin konusu, Arduino ile animasyonlu dijital saat yapımıdır. Saat hesaplaması yapıldıktan sonra, saniyelerin geçişleri ve rakam değiştiğinde bunun gösterilmesi gibi basit animasyonlar ile saat LCD ekranda gösterilmektedir.

Bu projenin yapımında LCD ekran yerine segmentler de kullanılabilirdi. Çünkü amaç saatteki değişimleri göstermektir. Ama LCD ekranda gösterim daha düzenli olduğu için bu kullanılmıştır.

Öncelikle saat, dakika ve saniye hesaplamaları yapılmıştır. Bu değerler arasına da “:” işareti getirilerek birbirlerinden ayrılması sağlanmıştır. Sonrasında da yapılan hesaplamalar LCD panelde gösterilmiştir. Bu değerler ekranda gösterilirken animasyon katmak amacıyla, her bir saniye geçtiğinde değerler sağa veya sola kaydırılmıştır. Saat, dakika ve saniye değerleri aynı anda hareket etmektedir. Böylece bir hareketlilik kazandırılmıştır.

Tüm bu işlemler önce Arduino programlayıcı ile kodlanmıştır. Ardından da gerekli donanımlar gerekli şekilde bağlanmış ve proje çalışır hale getirilmiştir. Arduino dışında, Lcd ekran, potansiyometre ve yeterli sayıda direnç kullanılarak donanım oluşturulmuştur.

2. ARDUİNO MİMARİSİ

Arduino üzerinde iki adet chip bulundurulur. Bunlardan biri ATmega16U2, diğeri ATmega328.16U2, USB kontroller olarak kullanılıyor. USB Serial'dan okuduğu değeri ATmega328'in okuyabilmesi için UART'a yazıyor. Arduino'nun programlanması da bu şekilde sağlanmış oluyor.

USB üzerinden programlama yapılırken, aslında SPI kullanılmıyor. Bunun yerine ATmega328'de çalışır olarak gelen bir bootloader firmware, UART'tan okuduğu değerleri flash'a yazıyor. Böylece programlama işlemi gerçekleşiyor.

Hem 16U2, hem de ATmega328 üzerinde bootloader diye adlandırılan bir program var. Bu program Arduino çalışmaya başladığı andan itibaren koşturaya başlar. Arduino'nun sağladığı en güzel faydalardan biri de budur. Böylece, bir programlayıcı kullanmadan da chip'inizi programlayabilirsiniz.

2.1. Arduino ATmega328 Bellek Alanları

Arduino üzerinde 3 farklı depolama alanı bulunur. Aslında bu Arduino üzerinde değil, ATmega328 üzerindedir. Flash, RAM, EEPROM.

Flash programlarınızın saklandığı alandır. Ve programlar aslında direk burada koştur. Yani buradan RAM bölgesine aktarılmaz. Bootloader da bu bölgede bulunur. ATmega328 ayarları yapılarak bootloader için ayrılan alan azaltılabilir, hatta silinebilir. Böylece daha büyük programlar da koşturabilirsiniz. Ancak böyle bir durumda ATmega328'i Arduino üzerindeki USB ile programlayamazsınız, çünkü UART'ı dinleyen bootloader'ı silmiş olursunuz. Böyle bir durumda SPI interface ile chip'inizi programlamanız gerekir.

RAM, koştur sırasında değişkenlerin tutulduğu alandır. Enerji kesilmesi durumunda bu bölgedeki veriler silinecektir.

EEPROM, programcıya kalıcı olarak veri saklaması için sunulmuş alandır. Aslında isteğe göre flash'da da programcı veri saklayabilir. Ama bunun için koştur programın bootloader bölgesinde olması gerekir. Aksi halde güvenlik problem oluşacağından, ATmega328 buna izin vermez.

3. STANDARTLAR VE KISITLAR

Projenin tasarım boyutuna bakıldığında karmaşıklık söz konusu değildir. Gerek kullanılan malzemelerin az olması, gerekse bu malzemelerin temin edilebilir olmasından dolayı kolaylık sağlamıştır.

Projede kullanılan malzemeler anlaşılması ve kullanımı ya da kurulumu zor olmayan malzemelerdir. Ekonomik açıdan değerlendirme yapıldığında maliyet çok da yüksek değildir. Burada bulunması zor olan ve belki de maliyeti bir nebze artıran cihaz Arduino'dur. Bunun yerine PIC kullanılarak, olan maliyet daha da düşürülebilir. Bu proje için PIC programlanarak yapılan birçok örnek de mevcuttur. Kayan yazı ekranları örnek verilebilir.

Aslında Arduino da bir PIC'tir. Üzerinde bulundurduğu mikrodenetleyiciler PIC ile aynı kategoridedir. Bu nedenle PIC ile yapılabilecek birçok uygulama Arduino ile de yapılabilmektedir. Bu da kullanım açısından kolaylık sağlamaktadır.

Proje geliştirilebilirlik açısından değerlendirildiğinde, ileriye götürülebilmesi kullanıcıya bağlıdır. Saate daha farklı animasyonlar katılabilir. Kullanılan donanım artırılabilir. Mesela saat ayarlaması yazılım ile değil de el ile yapılabilecek hale getirilebilir. Bunun için de butonlar veya başka elemanlar kullanılabilir.

Günümüzde bu tür dijital saatler çok kullanılmaktadır. Maliyetin de çok yüksek olmamasından dolayı temin edilmesi de kolay olan bir proje olarak söylenebilir.

4. BENZER ÇALIŞMALAR

Animasyon söz konusu olduğunda donanım ile yapılan birçok proje ya da örnek uygulama bulunabilir. Ve temelde hepsi aynı mantığa dayanmaktadır. PIC kullanarak ya da Arduino kullanarak bu uygulamalar gerçekleştirilebilir. Örnek birkaç uygulama şu şekilde maddeler halinde yazılabilir.

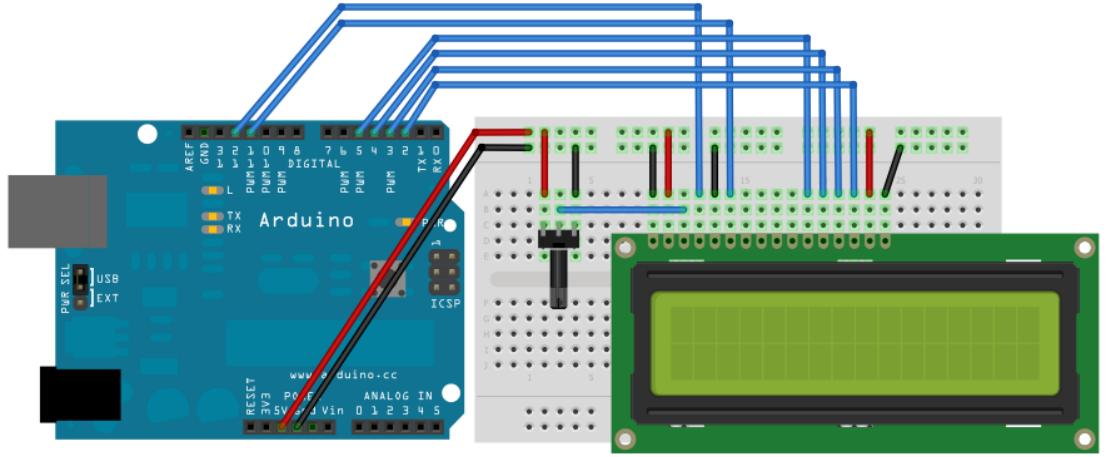
- Asansördeki kat bilgisini gösteren ekranda bu animasyon yöntemi kullanılabilir.
- Reklam tabelalarında kullanılan ledli kayan yazı işlemleri bu mantıkla yapılabilir.
- Görsellik katması amacıyla analog saatin led kullanılarak yapılması.
- Günümüzde çok fonksiyonlu dijital kol saatleri bu yöntemle benzer çalışmalar kullanılarak üretilmektedir.

Bunlara benzer birçok alanda çalışmalar mevcuttur. Bu tarz çalışmalar özellikle günlük hayatta, birçok iş kurumunda, halka açık alanlarda, görselliğin ön planda olduğu birçok yerde kullanılmakta ve geliştirilmektedir.

5. ÖNERİLEN YÖNTEM

5.1. Arduino LCD Bağlantıları

Aşağıda LCD ekran ile Arduino arasındaki bağlantıların nasıl yapılacağını gösteren şema bulunmaktadır.



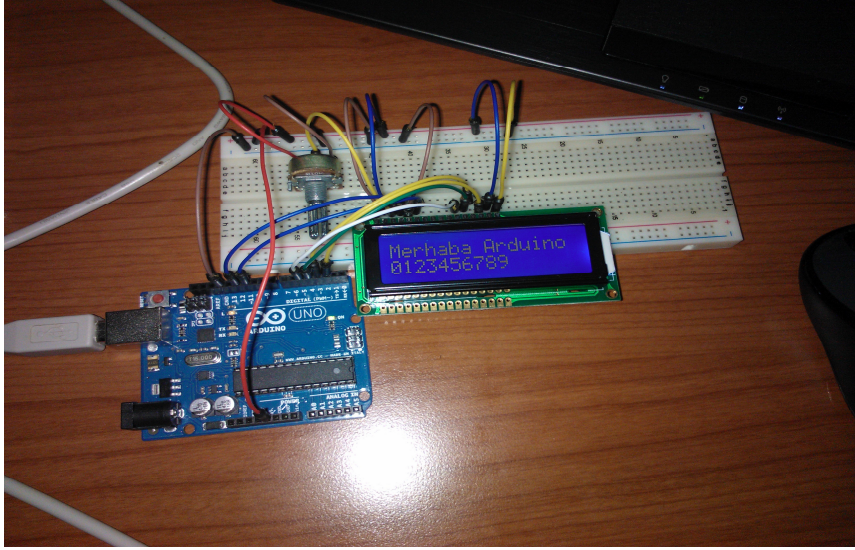
Şekil 5.1.1. Arduino bağlantıları

Burada potansiyometre, LCD ekrandaki yazının parlaklığını ayarlamak için kullanıldı. Maviışıklı LCD ler için bir de arka ışık vardır. Bu ışığı yakmak için LCD nin son iki pinine sırasıyla 5V ve toprak bağlamak gerekiyor.

```
#include<LiquidCrystal.h>           //ilgili kütüphaneyi çağırma
LiquidCrystallcd(12, 11, 5, 4, 3, 2); //LCD yi oluşturma
void setup() {
  lcd.begin(16, 2);                 //LCD nin boyutunu belirleme
}
void loop() {
  lcd.setCursor(0, 0);              //cursor u baştan başlatma
  lcd.print("MerhabaArduino");
  lcd.setCursor(0, 1);              //cursor u alt satıra alma
  lcd.print("0123456789");
  delay(2000);                      //gecikme
  lcd.clear();                      //lcd yi temizleme
  lcd.setCursor(0, 0);
  lcd.print("Arduino");
  lcd.setCursor(0, 1);
  lcd.print("Turkiye");
  delay(2000);                      //gecikme
}
```

Şekil 5.1.2. LCD ekranın nasıl kullanıldığını gösteren örnek kod

Burada ana döngünün içinde 2000 ms (yani 2 sn) aralıklarla iki farklı yazı yazdırılıyor. Koddaki delay fonksiyonları bu sebeple kullanıldı.



Şekil 5.1.3. Bağlantıların orijinal görüntüsü

5.2. Arduino Kayan Saat Uygulaması

```
int s=0;  
int sec=0;  
inthrs=0;  
int minutes=0;  
intinitialHours = 02;  
intinitialMins = 0;  
intinitialSecs = 00;
```

Şekil 5.2.1. Saat için öncelikle ilk değerler atanmıştır.

```
int seconds()  
{  
s = initialHours*3600;  
s = s+(initialMins*60);  
s = s+initialSecs;  
s = s+(millis()/1000);  
return s;  
}
```

Şekil 5.2.2. second() fonksiyonu

Saniye hesabı seconds() fonksiyonuyla yapılıyor. Burada Arduino'nun kendi kütüphanesinde bulunan millis() fonksiyonu kullanılarak gerçek zamanlı ilerleyiş hesaplanmış oluyor. Burada saniyeden kasıt, şuan ki zamanın saniye cinsinden değeridir. Bu kod parçası Şekil 5.2.2 de gösterilmiştir.

```
int secs()
{
  sec = seconds();
  sec = sec%60;
  return sec;
}
```

Şekil 5.2.3 secs() fonksiyonu

Saniye hesabı secs() fonksiyonuyla yapılıyor. İlerleme seconds() fonksiyonu çağrılarak sağlanmış oluyor. Dakika hesabı mins() fonksiyonuyla yapılıyor. İlerleme seconds() fonksiyonu çağrılarak sağlanmış oluyor.

```
int mins()
{
  minutes = seconds();
  minutes = minutes/60;
  minutes = minutes%60;
  return minutes;
}
```

Şekil 5.2.4. mins() fonksiyonu

Saat hesabı hours() fonksiyonuyla yapılıyor. İlerleme seconds() fonksiyonu çağrılarak sağlanmış oluyor. Bu kod parçası da aşağıdaki şekilde verildiği gibidir.

```
int hours()
{
hrs = seconds();
hrs = hrs/3600;
hrs = hrs%24;
returnhrs;
}
```

Şekil 5.2.5. hours() fonksiyonu

digitalClockDisplay() fonksiyonu ile hesaplanmış değerlerin LCD ekranda görüntülenmesi sağlanıyor. Ayrıca görüntünün kaydırılması da burada gerçekleştiriliyor. Buna ait kod parçası da aşağıdaki gibi verilmiştir.

```
void digitalClockDisplay(){
lcd.setCursor(5, 0);
if(state == 0) {
a--;
if(a == 0)
state = 1;
else if(a == 750 || a == 500 || a == 250)
lcd.scrollDisplayLeft();
}
else if(state == 1) {
a++;

if(a == 1000)
state = 0;
else if(a == 750 || a == 500 || a == 250)
lcd.scrollDisplayRight();
}
```

Şekil 5.2.6. Saatin sağa veya sola kaydırılması bu kod ile sağlanmıştır.

6. DENEYSEL SONUÇLAR

Genel olarak animasyonlu saatleri ele aldığımızda; gerekli donanım elemanları ve gerekli yazılımlarla çok çeşitli animasyon eklemek mümkündür. Geliştirilmesi açısından değerlendirildiğinde, kullanıcının isteğine bağlı olarak geliştirilmesi artırılabilir.

Kullanım alanları oldukça geniştir. Günümüzde hemen hemen her yerde bu tarzda cihazlar bulunmaktadır. Burada dezavantaj olarak kullanım alanlarının çeşitliliğinden kaynaklanan enerji problemi oluşabilir. Gerekli elektriği sağlamak için de özellikle açık alanlarda güneş enerjisinden faydalanılabilir.

Sonuç olarak şu söylenebilir. Bu projede kullanılan malzemeler rahatlıkla bulunabilmiştir ve maliyeti de düşüktür. Programlama dili açısından değerlendirme yapıldığında ise Arduino' nun sağlamış olduğu programlama ortamı işlemleri kolaylaştırmış ve hızlandırmıştır. Aynı şekilde donanımın bağlanması konusunda da avantaj sağlamıştır.

7. KAYNAKLAR

1. <http://www.gurayyildirim.com.tr/arduino-ile-ilk-adim-baslangic-902.html>
2. <http://www.trerk.com/teknik/7/potansiyometre/index.html>
3. http://www.robotiksistem.com/lcd_yapisi_calismasi.html
4. http://tr.wikipedia.org/wiki/Diren%C3%A7_%28elektrik%29
5. http://forum.donanimhaber.com/m_32768272/tm.html
6. <http://arduino.cc/en/Tutorial/HomePage>