

BÜLENT ECEVİT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

MATLAB ORTAMINDA ARDUINO PROGRAMLAMA

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

DÖNEM ÖDEVİ

KEREM CAN KATIRCI

Mayıs 2017

BÜLENT ECEVİT ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

MATLAB ORTAMINDA ARDUINO PROGRAMLAMA

ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

DÖNEM ÖDEVİ

Kerem Can KATIRCI

DANIŞMAN : Yrd. Doç. Dr. İbrahim Alışkan

ÖDEV DANIŞMANI : Öğr. Gör. Aytaç ALTAN

ZONGULDAK

Mayıs 2017

“Bu alıřmadaki tm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiđini ve sunulduđunu; ayrıca bu kuralların ve ilkelerin gerektirdiđi řekilde, bu alıřmadan kaynaklanmayan btn atıfları yaptıđımı beyan ederim.”

KEREM CAN KATIRCI

ÖZET

Programlamaya Giriş Dönem Ödevi

MATLAB ORTAMINDA ARDUINO PROGRAMLAMA

Kerem Can KATIRCI

Bülent Ecevit Üniversitesi

Mühendislik Fakültesi

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Ödev Danışmanı: Öğr. Gör. Aytaç ALTAN

Mayıs 2017, 47 sayfa

MATLAB, genellikle pozitif bilim ve mühendislik hesaplamaları için kullanılan bir bilgisayar programıdır.

Arduino ise kolay programlanabile üzerinde Atmel mikro işlemci bulunan ve giriş çıkış pinlerine sahip olan bir mikro denetleyicidir.

Ödevde MatLab, Arduino ve kullanılan malzemeler hakkında genel bir bilgi verilmiştir. Sonrasında ise MatLab'ın kurulumu anlatılmış, Command Window'da basit uygulamalar yapılmış ve son bölümde ise bir arayüz oluşturularak proje geliştirilmiştir.

MatLab'ın bilgisayara kurulumundan başlayarak proje geliştirmeye kadar olan adımların anlatıldığı bir çalışma olmuştur.

Anahtar Kelimeler: MATLAB, Arduino

ABSTRACT

Introducttion To Programming Term Paper

ARDUINO PROGRAMMING IN MATLAB

Kerem Can KATIRCI

Bülent Ecevit University

Engineering Faculty

Department of Electrical Engineering

Advisor: Assist. Prof. Dr. İbrahim Alışkan

May 2017, 47 pages

MATLAB is a computer program that is usually used for positive science and engineering calculations.

Arduino is a microcontroller that is easy Programmable. The microcontroller has an Atmel microprocessor and has input and output pins.

Matlab, Arduino and the materials used are given general information at the assignment. After that, the installation of MatLab is explained, simple applications are made in the Command window and In the last part, an interface was formed and the project was developed.

It was an assignment that explained the steps from MatLab's computer installation to project development.

Keywords: MATLAB, ARDUINO

TEŞEKKÜR

Bu çalışmam boyunca ilminden faydalandığım, disiplinini kendime örnek edindiğim, hocam olmasından onur duyduğum ve ayrıca tecrübelerinden yararlanırken göstermiş olduğu hoşgörü ve sabırdan dolayı değerli hocam sayın Aytaç ALTAN' a,

Bana bu çalışmamda ve yapacağım tüm çalışmalarda, tüm hayatımda ve her anımda en büyük desteği sağlayan Berfin YEMELEK'e teşekkürlerimi sunarım.

Kerem Can KATIRCI

İÇİNDEKİLER

| | <u>Sayfa</u> |
|---|--------------|
| ÖZET..... | vii |
| ABSTRACT..... | ix |
| TEŞEKKÜR..... | xi |
| İÇİNDEKİLER..... | xiii |
| ŞEKİLLER DİZİNİ..... | xv |
| KISALTMALAR DİZİNİ..... | xvii |
| BÖLÜM 1 | 19 |
| MATLAB ORTAMINDA ARDUINO PROGRAMLAMA..... | 19 |
| 1.1 GENEL BİLGİ..... | 19 |
| 1.2 ÇALIŞMADA KULLANILAN MALZEMELER..... | 19 |
| BÖLÜM 2 | 23 |
| MATRIX LABORATORY..... | 23 |
| 2.1 MATLAB NEDİR?..... | 23 |
| 2.2 MATLAB'IN WINDOWS ORTAMINA KURULMASI..... | 24 |
| BÖLÜM 3 | 28 |
| MATLAB ve ARDUINO HABERLEŞMESİ..... | 28 |
| 3.1 MATLAB'A ARDUINO IO EKLENTİSİNİN KURULMASI..... | 28 |
| BÖLÜM 4 | 30 |
| COMMAND WINDOW'DA BASİT UYGULAMALAR..... | 30 |
| 4.1 COMMAND WINDOW'DA LED KONTROLÜ..... | 30 |
| 4.2 COMMAND WINDOW'DA SICAKLIK ÖLÇMEK..... | 33 |
| BÖLÜM 5 | 35 |
| MATLAB İLE GUI OLUŞTURMAK ve PROJE GELİŞTİRMEK..... | 35 |
| 5.1 MATLAB İLE GUI OLUŞTURMAK..... | 35 |
| KAYNAKLAR..... | 45 |
| ÖZGEÇMİŞ | 47 |

ŞEKİLLER DİZİNİ

| <u>No</u> | <u>Sayfa</u> |
|---|--------------|
| Şekil 1.2.1 Arduino UNO..... | 19 |
| Şekil 1.2.2 Led ve Yapısı | 20 |
| Şekil 1.2.3 LM35..... | 20 |
| Şekil 1.2.4 Röle | 21 |
| Şekil 1.2.5 LDR..... | 21 |
| Şekil 1.2.6 Breadboard | 22 |
| Şekil 1.2.7 Jumpe Kablo | 22 |
| Şekil 1.2.8 MATLAB Yüklü Bir Bilgisayar | 22 |
| Şekil 2.1.1 MATLAB Kullanıcı Arayüzü | 23 |
| Şekil 2.1.2 MATLAB ile Çizilen Grafik..... | 24 |
| Şekil 2.2.1-10 Kurulum Adımları | 24-27 |
| Şekil 3.1.1 Math Works İnternet Sayfası | 28 |
| Şekil 3.1.2 MATLAB Çalışma Klasörünün Açılması..... | 28 |
| Şekil 3.1.3 Current Folder | 29 |
| Şekil 4.1.1 Led İçin Kurulan Devre Şematiği | 30 |
| Şekil 4.1.2 Led İçin Kurulan Devre | 31 |
| Şekil 4.1.3 Yanan Led | 32 |
| Şekil 4.2.1 LM35 İçin Kurulan Devre Şematiği | 33 |
| Şekil 4.2.2 Sıcaklık Ölçümü İçin Kurulan Devre..... | 32 |
| Şekil 5.1.1 GUI'yi Açmak..... | 35 |
| Şekil 5.1.2 Tasarım Ekranının Açılması | 35 |
| Şekil 5.1.3 Tasarım Ekranı..... | 36 |
| Şekil 5.1.4 Proje İçin Hazırlanan Tasarım | 36 |
| Şekil 5.1.5 Projenin Devre Şematiği | 40 |
| Şekil 5.1.6 Breadboard'a Kurulan Devre | 41 |
| Şekil 5.1.7 GUI..... | 41 |
| Şekil 5.1.8 Sensöre Gelen Işığı Engelleme | 42 |

| | |
|---|----|
| Şekil 5.1.9 Tüm Ledler Aktif | 42 |
| Şekil 5.1.10 Devreye Genel Bakış | 43 |
| Şekil 5.1.11 Programa Genel Bakış | 43 |

KISALTMALAR DİZİNİ

KISALTMALAR

| | |
|---------------|----------------------------------|
| MCU | : Micro Controler Unit |
| IDE | : Integrated Development Environ |
| LED | : Light Emitting Diode |
| mV | : Milivolt |
| LDR | : Light Dependet Resistance |
| MATLAB | : Matrix Laboratory |
| IO | : Input / Output |
| D0 | : Dijital 0 Pini |
| C | : Santigrat |
| V | : Volt |
| A0 | : Analog 0 Pini |
| GUI | : Graphical User Interface |

BÖLÜM 1

MATLAB ORTAMINDA ARDUINO PROGRAMLAMA

1.1 GENEL BİLGİ

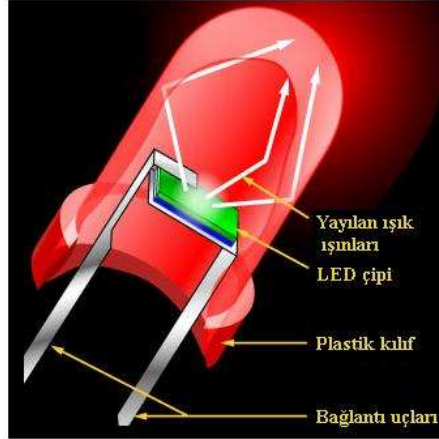
“Algoritma Geliştirme ve Programlamaya Giriş” dersi ve kendimizi geliştirmek, yeni teknolojiler öğrenmek için yapılan bu çalışma toplamda 5 bölümden oluşmaktadır. Temelde MATLAB ile Arduino adı verilen bir küçük çaplı bir geliştirme kartının programlanması üzerinde durulmuştur. Bu bölümde genel bilgi ve kullanılan malzemeler hakkında bilgiler verilmiştir.

1.2 ÇALIŞMADA KULLANILAN MALZEMELER



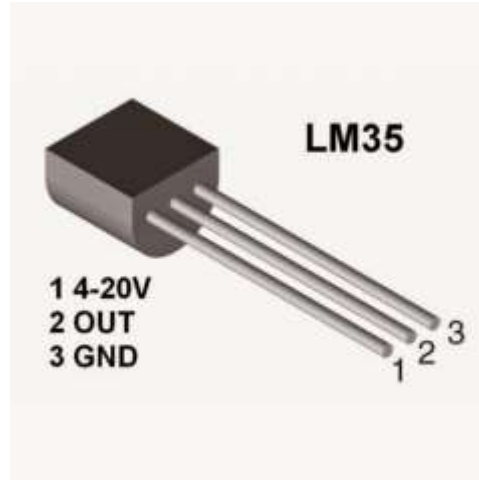
Şekil 1.2.1 Arduino UNO

Arduino, açık kaynak kodlu bir mikrodenetleyici kartıdır. Ana MCU, Atmel tabanlıdır. (MCU’ya önceden bir mini program-bootloader yüklenmiştir). Kart ile robotik ve elektronik uygulamalar kolaylıkla gerçekleştirilir. Açık kaynak kodludur. Kendi IDE’ si sayesinde program geliştirmek kolaydır.



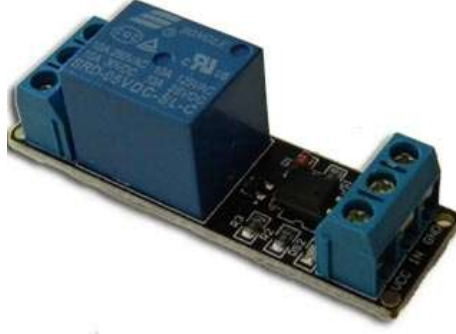
Şekil 1.2.2 Led ve Yapısı

LED, yarı-iletken, diyot temelli, ışık yayan bir elektronik devre elemanıdır. 1920'lerde Rusya'da icat edildi ve 1962 yılında Amerika'da pratik olarak uygulanabilen elektronik bir bileşen haline getirildi.



Şekil 1.2.3 LM35

LM35, analog bir sıcaklık sensörüdür. Analog bir sensörün getirdiği tüm artılara sahiptir. Hassas ölçüm, düşük enerji tüketimi bunların başında gelir. Kılavuzuna göre sensör 1 derece sıcaklık için 10mV gerilim üretir ve biz de bir takım basit matematiksel işlemlerden sonra sıcaklık değerini elde ederiz.



Şekil 1.2.4 R le

Elektromanyetik  alıřan bir devre elemanıdır.  zerinden akım ge tiđi zaman  alıřır. D ř k voltajlarla y ksek voltajları kontrol etmeye yarar.



Şekil 1.2.5 LDR

Foto diren ler,  zerlerine d řen ıřık řiddetiyle ters orantılı olarak diren leri deđiřen elemanlardır. Foto diren ,  zerine d řen ıřık arttık a diren  deđeri lineer olmayan bir řekilde azalır. LDR'nin aydınlıkta direnci minimum, karanlıkta maksimumdur.



Şekil 1.2.6 Breadboard

Breadboard, elektrik ve elektronik için vazgeçilmez bir araçtır. Türkçe'ye deneysel devre tahtası olarak çevirilebilir. Elektronik devreler tasarlanmadan önce devre elemanlarının çalışıp çalışmadığını ya da bağlantıların yanlış olup olmadığını kontrol etmek amacıyla kullanılır.



Şekil 1.2.7 Jumper Kablo

Jumper kablolar pratiklik için üretilen breadboardlar ile uyumlu bağlantı elemanlarıdır.



Şekil 1.2.8 MATLAB Yüklü Bir Bilgisayar

MATLAB yüklü bir bilgisayar, kodlarımızı yazıp Arduino'muza göndermemiz için gerekmektedir.

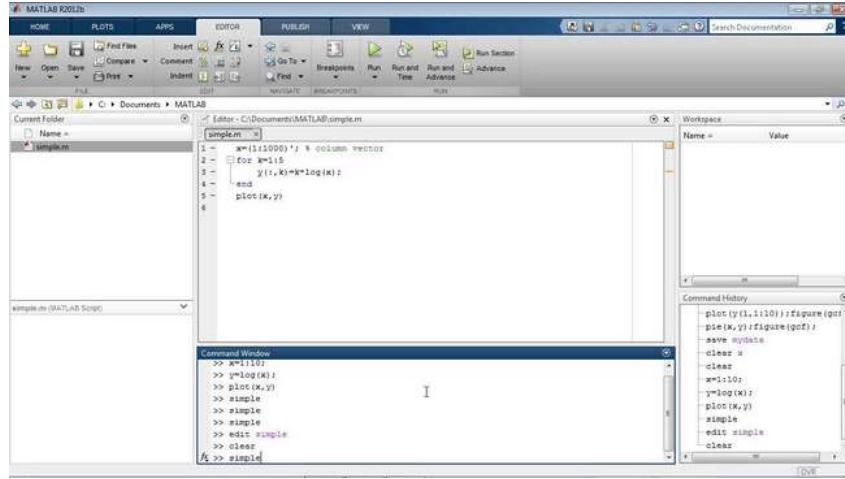
BÖLÜM 2

MATRIX LABORATORY

2.1 MATLAB NEDİR?

MATLAB, genellikle pozitif bilim ve mühendislik hesaplamaları için kullanılan bir bilgisayar programıdır. Amerika Birleşik Devletleri merkezli MathWorks firması tarafından geliştirilen MATLAB, aynı zamanda bir programlama dilidir. İngilizce “Matrix Laboratory” kelimelerinin birleştirilmesi ile oluşmuş olan MATLAB, isminden de anlaşılacağı gibi matris tabanlı bir çalışma sistemine sahiptir.

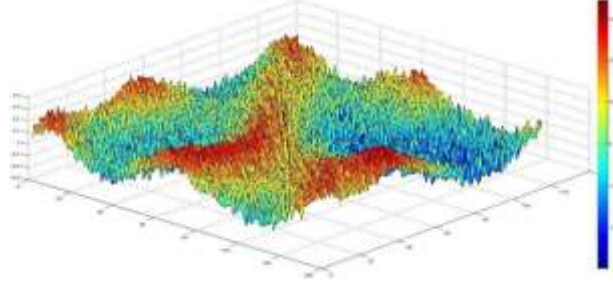
Lineer cebir, istatistik, optimizasyon, nümerik analiz, optimizasyon, fourier analizi gibi pek çok matematiksel hesaplamaların etkili ve hızlı şekilde yapılmasına olanak sağlayan MATLAB programı aynı zamanda 2D ve 3D grafik çizimi için de kullanılır.



Şekil 2.1.1 MATLAB Kullanıcı Arayüzü

MATLAB ile kullanıcılar kendi programlarını hazırlayabilirler. Matrisler ve onların etkileşim içinde olduğu fonksiyonlarla programlama yapılmasına izin veren MATLAB ile çok karmaşık matematik hesaplamaları bile birkaç saniye içinde tamamlanır. Temel

programlama fonksiyonları ile benzer fonksiyonların kullanılabildiği MATLAB ile etkili ve pratik programlar hazırlanabilir.



Şekil 2.1.2 MATLAB ile Çizilen Grafik

MATLAB ile temel matematik fonksiyonlarının iki ve üç boyutlu grafikleri çizilebilir. Polinomlar, parboller, sinüs dalgaları başta olmak üzere her tür iki ve üç boyutlu matematiksel grafik MATLAB ile elde edilebilir.

2.2 MATLAB'IN WINDOWS ORTAMINA KURULMASI

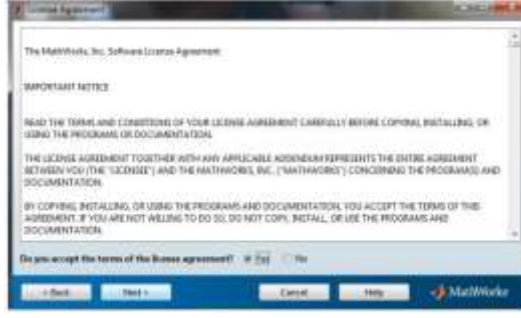
MATLAB'ı bilgisayarımıza “<https://www.mathworks.com/products/matlab.html>” adresinden indirdikten sonra MathWorks Installer'ı çalıştırarak kurulum yardımcımızı açıyoruz.

Hoş geldiniz ekranında Install without using the Internet'i seçerek kurulumla başlıyoruz.



Şekil 2.2.1 Kurulum Birinci Adımı

Daha sonra lisans sözleşmesi ekranında sözleşmeyi kabul edip, next ile devam ediyoruz.



Şekil 2.2.2 Kurulum İkinci Adımı

Dosya sunucusunda yer alan Matlab2013a_Lisans_key.txt dosyası içindeki key'i girdikten sonra next diyerek devam ediyoruz.



Şekil 2.2.3 Kurulum Üçüncü Adımı

Sonraki adımca Typical kurulumu seçiyoruz, dilerseniz kendiniz özelleştirebilirsiniz, ve kurulumla devam ediyoruz.



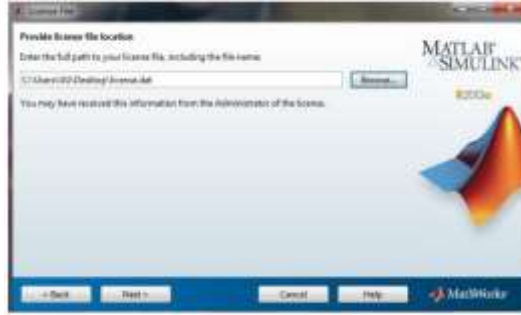
Şekil 2.2.4 Kurulum Dördüncü Adımı

Sonraki adımda MATLAB'ın hangi dizine kurulacağını belirliyoruz ve kurulumla devam ediyoruz.



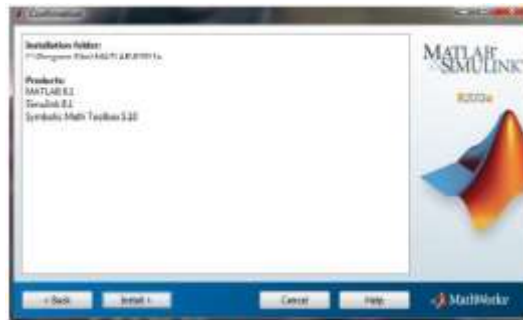
Şekil 2.2.5 Kurulum Beşinci Adımı

Bu adımda ise MATLAB'ı lisanslamak için Browse'u tıklayıp dosya sunucusundan edindiğimiz License.dat dosyasının yolunu gösteriyoruz.



Şekil 2.2.6 Kurulum Altıncı Adımı

Tipik kurulum bileşenleri MATLAB kendi seçtiğinden dolayı Install'a tıklayıp devam ediyoruz.



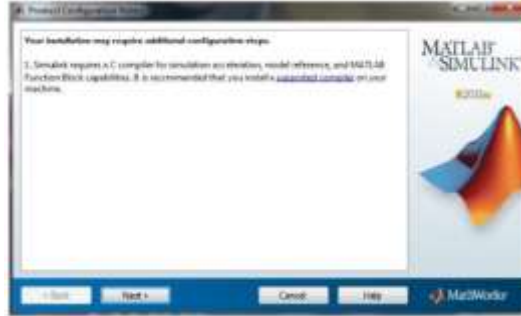
Şekil 2.2.7 Kurulum Yedinci Adımı

MATLAB'ın bilgisayarımıza kurulumunun tamamlanmasını bekliyoruz.



Şekil 2.2.8 Kurulum Sekizinci Adımı

Bu kısımda ise Next'e tıklayıp devam ediyoruz.



Şekil 2.2.9 Kurulum Dokuzuncu Adımı

Son olarak Installation is complete yazısını gördükten sonra Finish'e tıklayıp MATLAB 2013a' ı çalıştırabiliriz.



Şekil 2.2.10 Kurulumun Son Adımı

BÖLÜM 3

MATLAB ve ARDUINO HABERLEŞMESİ

3.1 MATLAB'A ARDUINO IO EKLENTİSİNİN KURULUMU

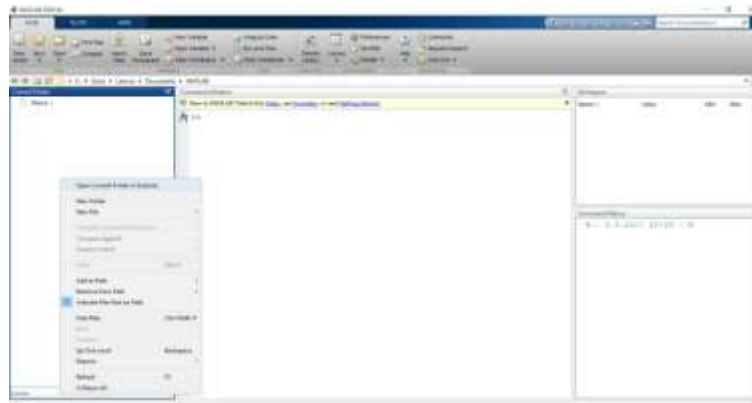
MatLab'ın Arduino ile haberleşebilmesi için MatLab Arduino IO eklentisinin kurulması gerekmektedir. Bunun için ilk olarak aşağıdaki linke tıklayarak MatLab ve Simulink destek paketini bilgisayarımıza indirmemiz gerekmektedir.

“<https://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino>”



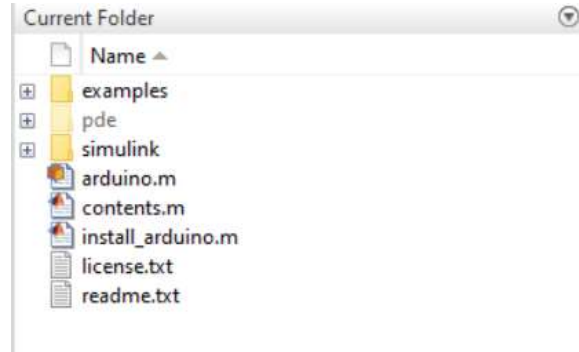
Şekil 3.1.1 Math Works İnternet Sayfası

Daha sonra indirdiğimiz sıkıştırılmış dosyayı arşivden çıkartıyoruz ve MatLab'da çalışma klasörümüze kopyalıyoruz. Eğer çalışma klasörümüzün nerede olduğunu bilmiyorsak MatLab arayüzünde Current Folder sekmesi altında herhangi bir boş alana sağ tıklıyoruz ardından Open Current Folder in Explorer' e tıklıyoruz ve çalışma klasörümüz açılıyor.



Şekil 3.1.2 MatLab Çalışma Klasörünün Açılması

Böylelikle çalışma klasörümüze Arduino IO kütüphanemizi kopyalamış olduk fakat işimiz daha bitmedi.



Şekil 3.1.3 Current Folder

Sonraki adımda Command Window'a "install_arduino" komutunu girerek kütüphanemizi MatLab'a eklemiş oluyoruz. Bu adım gerçekleştirilirken MatLab'ı yönetici olarak çalıştırmamız gerekmektedir, aksi takdirde program hata verecektir.

```
>> install_arduino
```

MatLab'a Arduino IO eklentimizi kurduktan sonra MatLab ile Arduino'muzu kontrol edebiliriz.

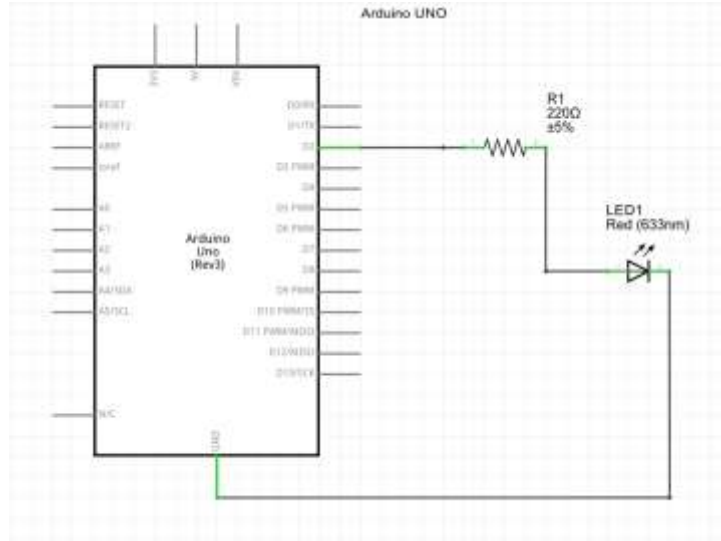
BÖLÜM 4

COMMAND WINDOW'DA BASİT UYGULAMALAR

4.1 COMMAND WINDOW'DA LED KONTROLÜ

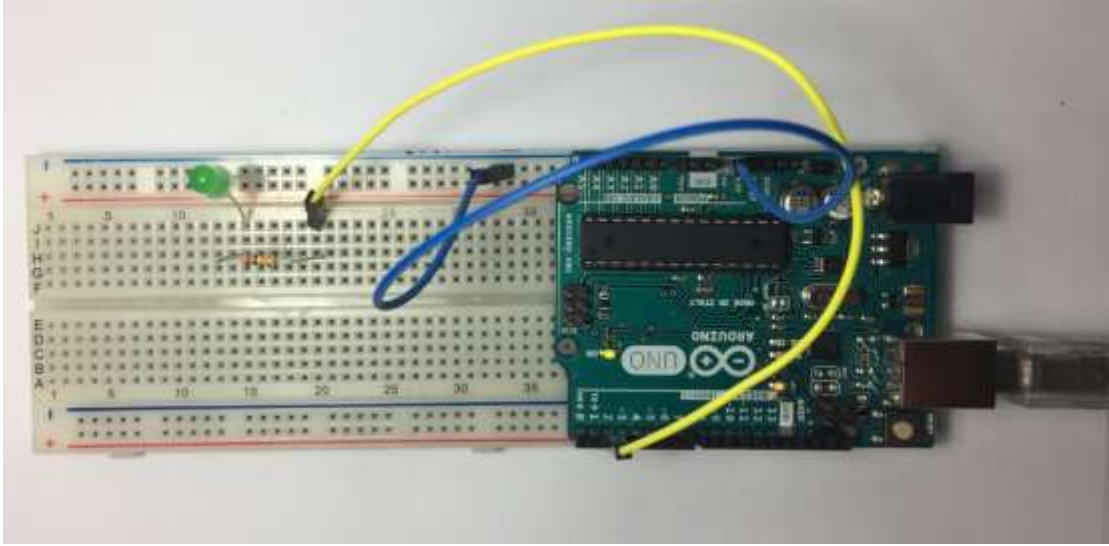
MatLab'da Command Window penceresinde hızlı ve basit bir şekilde programlar çalıştırılır, fonksiyonlar yazılır veya hesaplamalar yapılır. Bu bölümde led yakıp söndürme işlemini yapacağız. İlk olarak devremizi kurmalıyız ardından MatLab ile Arduino arasındaki bağlantıyı sağlamalıyız ve led kontrolü için gereken kodlarımızı yazmalıyız.

Uygulamamız için kuracağımız devremizin tasarımını Eagle, Proteus, Electronic Workbench, Fritzing gibi devre tasarım programları ile yapabiliriz. Bu ödevde detaya inilmediği için Fritzing kullanılmıştır.



Şekil 4.1.1 Led İçin Kurulan Devre Şematığı

Şekil 4.1.1'de tasarladığımız devremizde 1 adet led, 1 adet 220Ω direnç ve Arduino Uno kullanılmıştır. Led Arduino'nun D0 pinine direnç ile bağlanmıştır.



Şekil 4.1.2 Led İçin Kurulan Devre

Devremizi kurduktan sonra MatLab ile Arduino'nun haberleşebilmesi için kodlarımızı yazmamız gerekmektedir.

İlk olarak yazılan kod parçacığında Arduino ile MatLab arasındaki bağlantıyı gerçekleştirmek için bir Arduino objesi oluşturduk ve hangi port üzerinden haberleşeceğimizi belirledik.

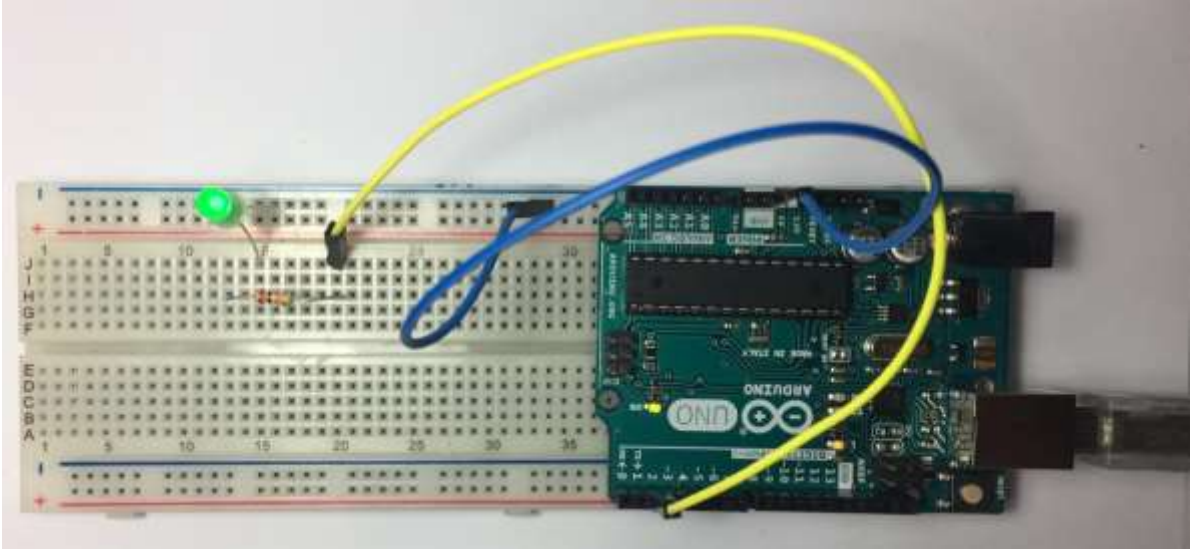
```
>> if exist('a','var') && isa(a,'arduino') && isvalid(a),  
    else  
        a=arduino('COM3');  
    end  
Attempting connection .....  
Basic Analog and Digital I/O (adio.pde) sketch detected !  
Arduino successfully connected !
```

Daha sonra ledi bağladığımız pini tanımladık ve bu pinin çıkış pini olduğunu belirttik.

```
>> a.pinMode(2,'output');
```

Kapalı konumda olan ledimizi açmak için aşağıdaki kodumuzu yazdık ve ledimizi açtık.

```
>> a.digitalWrite(2,1)
```



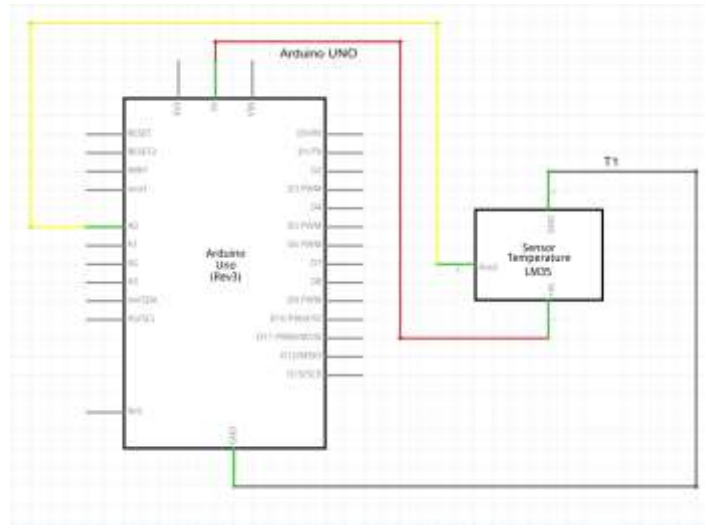
Şekil 4.1.3 Yanan Led

Ledimizi kapatmak içinse aşağıdaki kodu girerek ledimizi söndürdük.

```
>> a.digitalWrite(2,0)
```

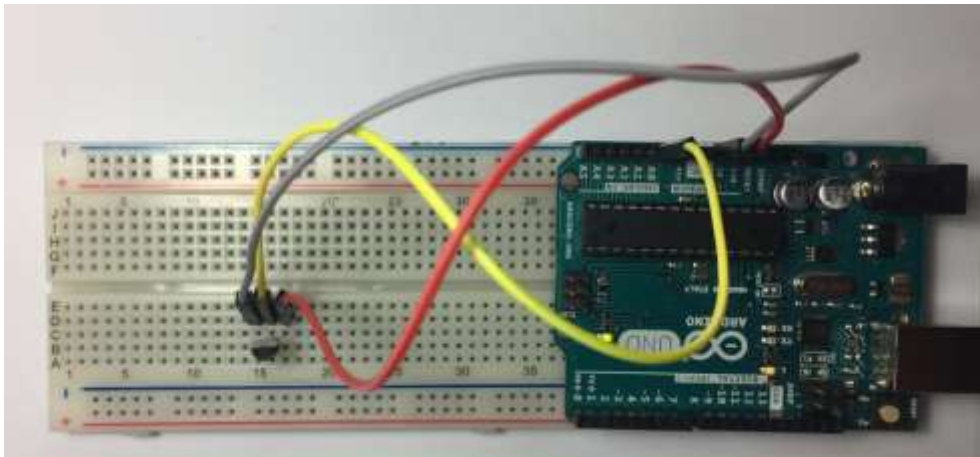

4.2 COMMAND WINDOW'DA SICAKLIK ÖLÇMEK

Bu uygulamamızda ise LM35 sıcaklık sensörü ile ortam sıcaklığını ölçeceğiz. LM35 sıcaklık sensörü, hassas sıcaklık ölçümü yapan analog çıkışlı bir sıcaklık sensörüdür. 25 °C'de 0,5 °C hassasiyete sahip olan bu sensörü, Arduino'muzun analog girişinden bağlayarak sıcaklık ölçümü yapılabilir. Bu sensör çıkış bacağından sıcaklık ile doğru orantılı olacak şekilde 0 ile 1V arasında gerilim vermektedir. Aşağıda kurulacak devrenin şematiği görülmektedir.



Şekil 4.2.1 LM35 için Kurulan Devrenin Şematiği

Şekil 4.2.1'de tasarladığımız devremizde Arduino ve LM35 sıcaklık sensörü kullanılmıştır. Sensörün çıkış bacağı A0 pinine bağlanmıştır.



Şekil 4.2.2 Sıcaklık Ölçümü İçin Kurulan Devre

Devremizi kurduktan sonra MatLab ile Arduino'nun haberleşebilmesi için kodlarımızı yazmamız gerekmektedir. Önceki örnekte seri haberleşme için bağlantı kodlarımız yazmıştık dosyamızı kapatmadığımız için bu ayarlar hala geçerlidir.

Kodlarımızı yazmaya Arduino'muzun analogdan dijitale çeviricisinin 1,1V referans voltajına ayarlama ile başlıyoruz. Bu sayede, analog girişten okuyacağımız 0-1023 arası değerlerin her bir adımını 1,0742 mV gerilime denk düşüyor. LM35 sıcaklık sensörünün çıkış bacağına okuduğumuz her 10 mV, 1 °C sıcaklığa denk geldiğinden; $10 / 1,0742$ bize yaklaşık olarak 9,31 değerini veriyor. Yani bu demek oluyor ki, analog girişten ölçtüğümüz değeri 9,31'e bölersek, elde ettiğimiz sonuç sensörün verdiği çıkış geriliminin santigrat cinsinden karşılığını elde ediyoruz. Aynı zamanda bu çevirme sayesinde %80 daha iyi bir sonuç elde etmiş oluyoruz.

```
>> a.analogReference('internal');
```

Sonraki adımda ise sensörün bize gönderdiği değeri okumak için aşağıdaki kodu giriyoruz ve sensörün bize gönderdiği ham veriyi okuyoruz ve “sicaklik” değişkenine atıyoruz.

```
>> sicaklik = analogRead(a,0)
```

```
sicaklik =
```

```
209
```

Daha sonra bu veriyi 9,31'e bölerek ortam sıcaklığımızı santigrat cinsinden ölçmüş oluyoruz.

```
>> sicaklik/9.31
```

```
ans =
```

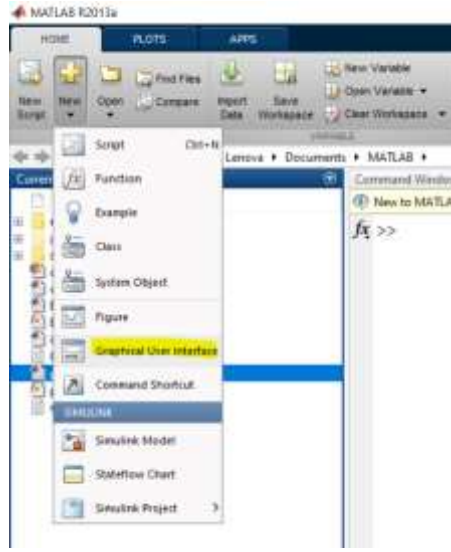
```
22.4490
```

BÖLÜM 5

MATLAB İLE GUI OLUŞTURMAK VE PROJE GELİŞTİRMEK

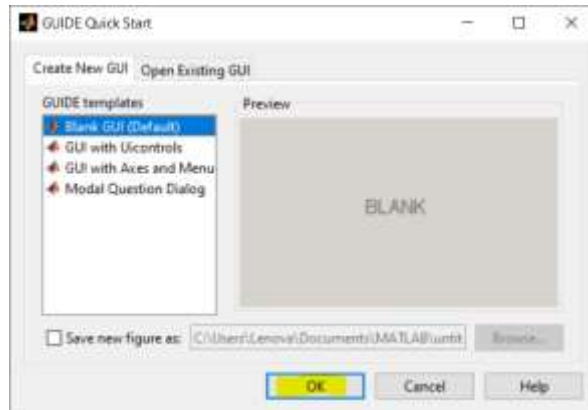
5.1 MATLAB İLE GUI OLUŞTURMAK

MatLab ile GUI oluşturmak için ilk olarak New sekmesi altında Graphical User Interface seçeneğine tıklıyoruz.

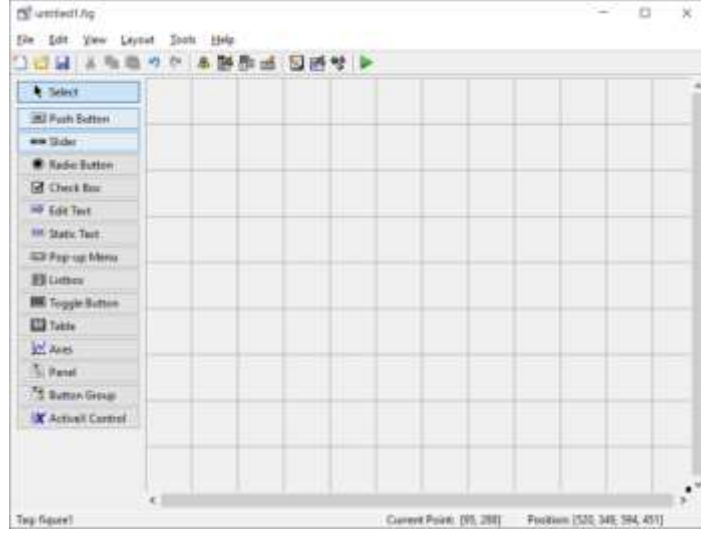


Şekil 5.1.1 GUI'yi Açmak

Daha sonra açılan pencerede Blank GUI (Default) seçip OK'a tıklayarak arayüz tasarımıımızı yapacağımız ekranı açıyoruz.

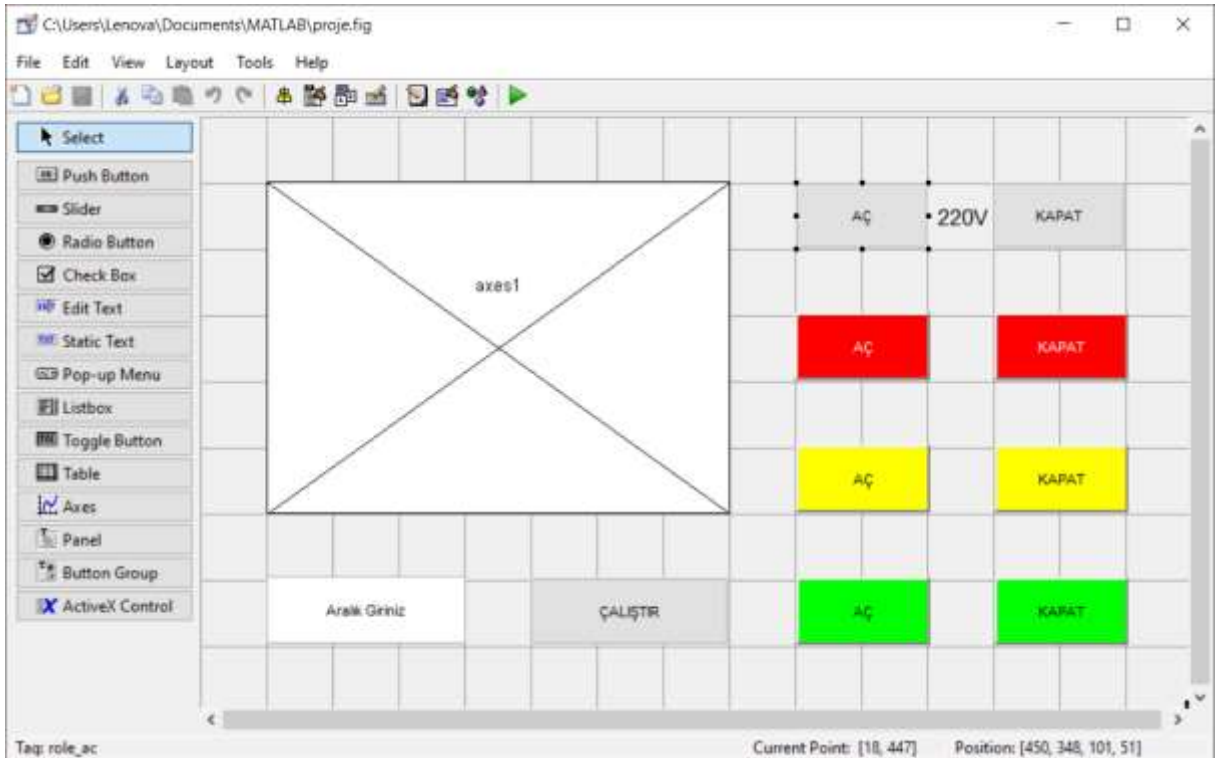


Şekil 5.1.2 Tasarım Ekranının Açılması



Şekil 5.1.3 Tasarım Ekranı

Bu ekranda arayüz tasarımı için kullanabileceğimiz araçlar bulunmaktadır. Buradan projemize uygun araçları seçerek tasarımı yapabiliriz. Bu projede Push Button, Axes, Edit ve Static Text araçları kullanılmıştır.



Şekil 5.1.4 Proje İçin Hazırlanan Tasarım

Bu araçlarımızı kullanarak tasarımımızı yaptıktan sonra sıra kodlarımızı yazmaya gelmiştir. Tasarladığımız GUI’de kullandığımız tasarım araçlarına ait uygun kodlar MatLab tarafından otomatik bir şekilde oluşturulmuştur. Bizim yapmamız gereken sadece bu tasarım araçlarının nasıl kullanılacağını ve programın nasıl çalışması gerektiğini belirlemektir.

Bu projenin amacı hazırlanan arayüz ile led yakıp söndürmek, ortamdaki ışık miktarını ölçerek ışık miktarı azaldığı zaman otomatik bir şekilde sisteme bağlanan rölenin çalışmasını sağlamaktır. Projenin amacı bir ortamın ışıklarını bilgisayar tarafından kontrol etmek ve ışık değeri şiddetini ölçerek, ışık şiddeti azaldığında ortamın ışıklarının otomatik bir şekilde açılmasını sağlamaktır.

Hazırladığımız arayüzü çalışma klasörümüze “proje” adında kaydettikten sonra “proje.fig” ve “proje.m” adında iki dosyamız oluşacaktır. Kodlarımızı yazabilmek için “proje.m” dosyamızı açmalıyız.

Dosyamızda ilk olarak “proje_OpeningFcn(.....)” fonksiyonunun altına Arduino ile iletişimin başlamasını sağlayan kodumuzu ve kullandığımız pinlerimizi tanımlayan kodlarımızı yazıyoruz.

```
function proje_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to proje (see VARARGIN)

% Choose default command line output for proje
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes proje wait for user response (see UIRESUME)
% uiwait(handles.figure1);
clear a;
global a;
a = arduino('COM3');
a.pinMode(8, 'output');
a.pinMode(9, 'output');
a.pinMode(10, 'output');
a.pinMode(11, 'output');
```

Daha sonra tasarladığımız objelerin (tasarım araçları) fonksiyonlarını yazmalıyız. Bu aşamada butonlara basıldığında hangi işlevleri gerçekleştireceğinin kodları yazacağız. Tasarımda objelere verdiğimiz etiketler sayesinde hangi butonun ne için tasarlandığını karıştırmamış oluyoruz. İlk olarak kırmızı ledimizin yanması ve sönmesi için gereken kodlarımızı yazıyoruz ve aynı işlemleri diğer ledlerimiz için de yapıyoruz.

```
function kirmizi_ac_Callback(hObject, eventdata, handles)
% hObject      handle to kirmizi_ac (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(8,1);

% --- Executes on button press in kirmizi_kapat.
function kirmizi_kapat_Callback(hObject, eventdata, handles)
% hObject      handle to kirmizi_kapat (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(8,0);

function sari_ac_Callback(hObject, eventdata, handles)
% hObject      handle to sari_ac (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(9,1);

% --- Executes on button press in sari_kapat.
function sari_kapat_Callback(hObject, eventdata, handles)
% hObject      handle to sari_kapat (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(9,0);

function yesil_ac_Callback(hObject, eventdata, handles)
% hObject      handle to yesil_ac (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(10,1);

% --- Executes on button press in yesil_kapat.
function yesil_kapat_Callback(hObject, eventdata, handles)
% hObject      handle to yesil_kapat (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(10,0);
```


Aynı zamanda rölemizi manuel olarak kontrol etmek için de aşağıdaki kod bloğumuzu yazıyoruz.

```
function role_ac_Callback(hObject, eventdata, handles)
% hObject    handle to role_ac (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(11,0);

% --- Executes on button press in role_kapat.
function role_kapat_Callback(hObject, eventdata, handles)
% hObject    handle to role_kapat (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global a;
a.digitalWrite(11,1);
```

Daha sonra sensörden değer okumaya ve bu değeri grafiğe aktarmak için gereken kod bloğumuzu yazıyoruz. Sensörden kaç birim değer okunacağını bir sonraki kod parçasığında öğreneceğiz.

Değer okumak için yazdığımız döngümüzün içine bir koşul bloğu koyuyoruz, böylelikle ışık şiddeti 50 birimin altına düştüğü zamana otomatik olarak rölemiz devreye giriyor ve çalışıyor.

```
function calistir_Callback(hObject, eventdata, handles)
% hObject    handle to calistir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global k a
x=0;

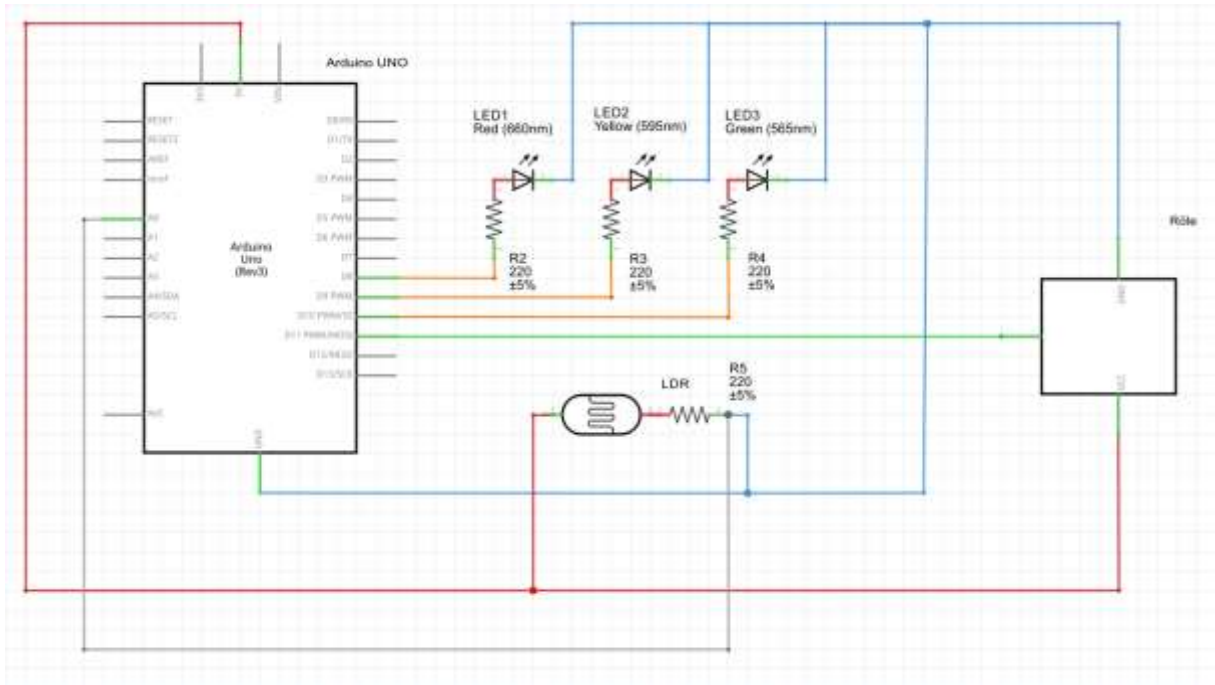
for k=1:1:handles.xSamples
    b=a.analogRead(0);
    if b<50
        a.digitalWrite(11,0);
    else
        a.digitalWrite(11,1);
    end
    x=[x,b];
    plot(x,'LineWidth',2); grid on;
    axis([0 handles.xSamples 0 500]);
    pause(0.01);
end
```

Sensörden kaç birim değer okunacağını belirlemek için kullanıcıdan veri alıyoruz ve bu veriyi bir önceki kısımda yazdığımız döngünün son değeri olarak belirliyoruz. Aynı zamanda bir önceki kısımda grafiğimizi oluşturmak için kodlarımız da yer almaktadır.

```
function aralik_gir_Callback(hObject, eventdata, handles)
% hObject    handle to aralik_gir (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

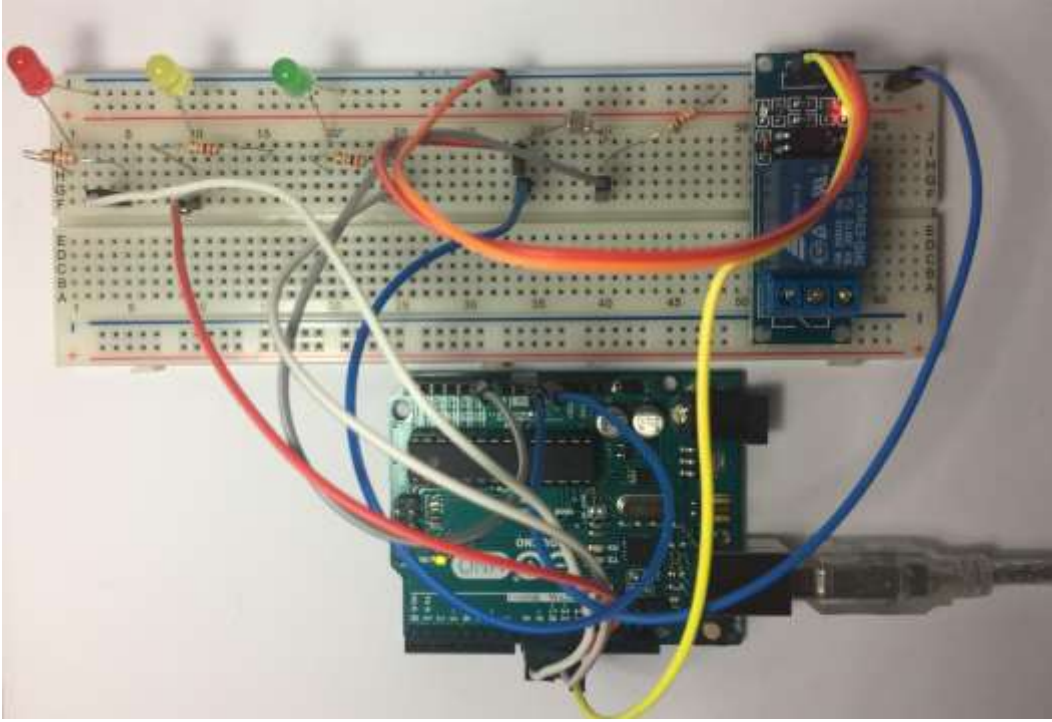
% Hints: get(hObject,'String') returns contents of aralik_gir as text
% str2double(get(hObject,'String')) returns contents of aralik_gir
handles.data1=get(hObject,'String');
handles.xSamples=str2double(handles.data1);
guidata(hObject,handles);
```

Böylelikle kodlarımızı da yazmış oluyoruz. Programımızı çalıştırarak Arduino'muzu kontrol etmeye başlamak için Şekil 5.1.5'deki devremizi kuruyoruz.



Şekil 5.1.5 Projenin Devre Şematiği

Daha sonra devremizi breadboarda kuruyoruz.



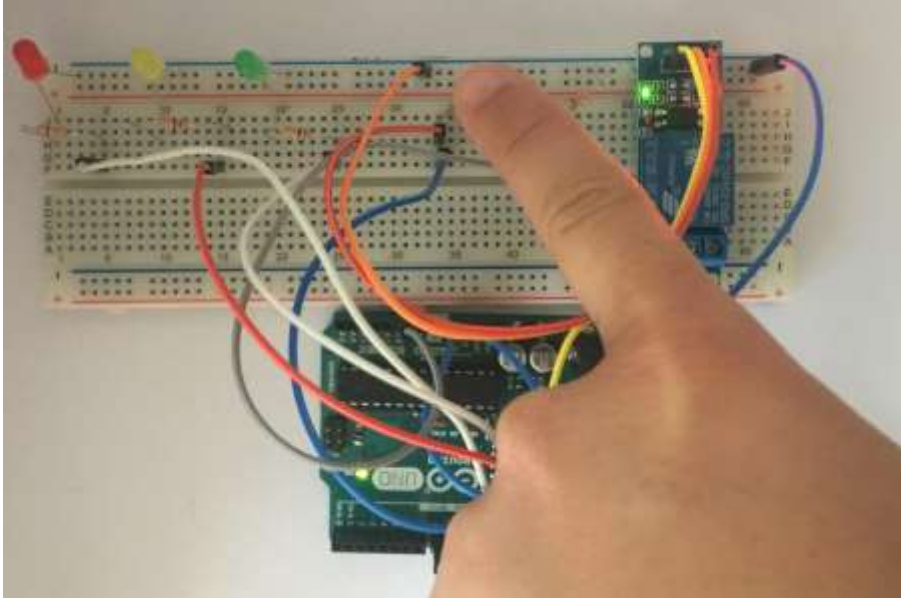
Şekil 5.1.6 Breadboard'a Kurulan Devre

Kaç birim değeri okunacağını girerek çalıştırıyoruz ve sensörden değeri okumaya başlıyoruz.



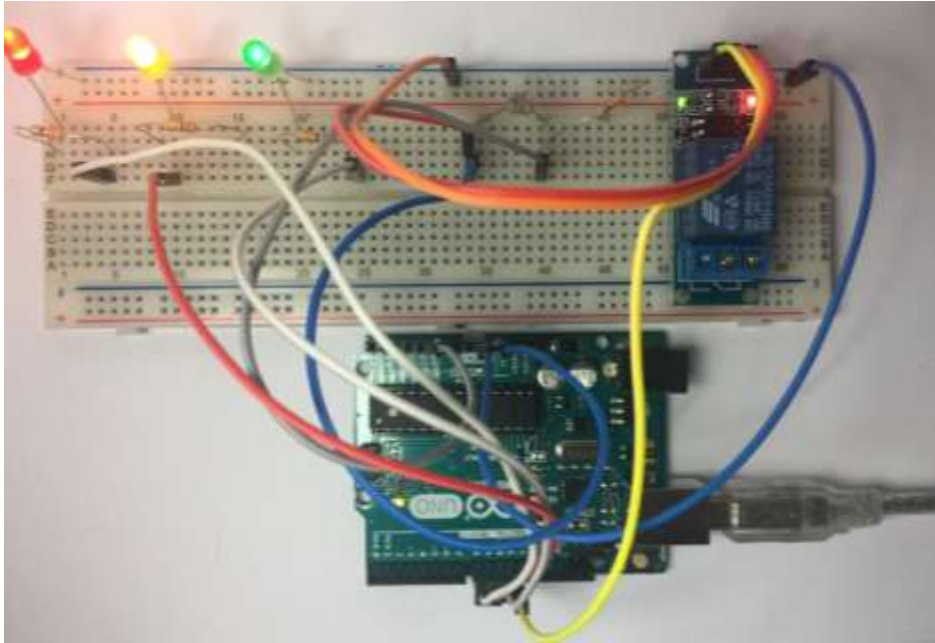
Şekil 5.1.7 GUI

Sensörden deęerler okunurken sensöre gelen ışığı kesiyoruz ve böylelikle rölenin aktif konuma geçtiğini görüyoruz.



Şekil 5.1.8 Sensöre Gelen Işıęı Engelleme

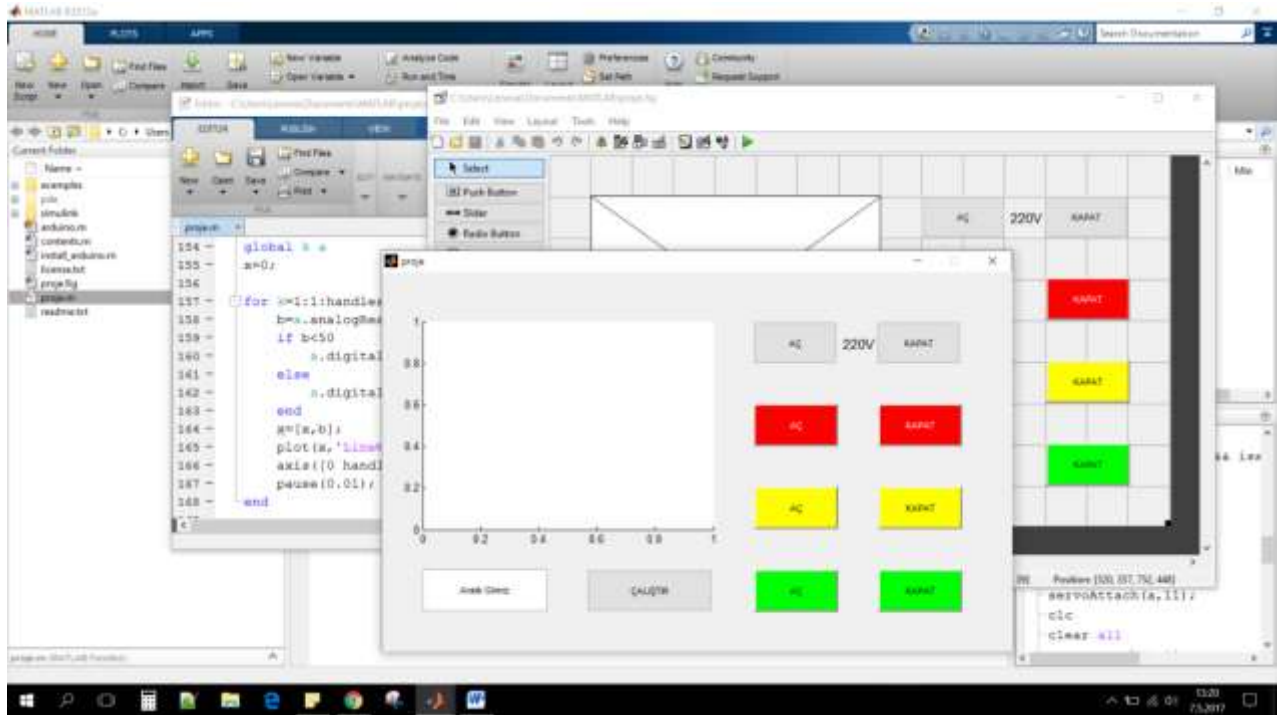
Projenin dięer aşamalarında ise ledlerimizi kontrol ediyoruz ve sistemimizin çalıştığını görüyoruz.



Şekil 5.1.9 Tüm Ledler Aktif



Şekil 5.1.10 Devreye Genel Bakış



Şekil 5.1.11 Programa Genel Bakış

KAYNAKLAR

Legacy MATLAB Support for Arduino. (19.04.2016) *Math Works Destek ve Bilgilendirme*

Adres: <http://www.mathworks.com/matlabcentral/fileexchange/32374-legacy-matlab-and-simulink-support-for-arduino?requestedDomain=www.mathworks.com>

URL-1 <<http://www.mathworks.com>>, Ziyaret tarihi: 07.05.2017

ÖZGEÇMİŞ

Kerem Can KATIRCI 08.07.1996 yılında Zonguldak'ta doğdu. Çaycuma Koleji'ni ve Mehmet Çelikel Lisesi'ni bitirdi. 2014 yılında Bülent Ecevit Üniversitesi Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü'nde lisans eğitimine başladı.

2008 ve 2010 yılları arasında öğrenmeye başladığı HTML ve PHP dillerini öğrenmeye başladı. Bu diller üzerinde kendini geliştirdi. Böylelikle programlamaya merak sardı.

ADRES BİLGİLERİ:

Adres: Bülent Ecevit Üniversitesi Elektrik-Elektronik Mühendisliği

Tel: (+90) 530 101 40 13

E-posta: katrckeremcan@gmail.com

Web: <http://www.keremcankatirci.com>