



**T.C.
KARABÜK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ**

**ARDUİNO İLE PARK SENSÖRÜ BLUETOOTH HABERLEŞME
VE ARDUİNO İLE DİJİTAL KUMPASTAN VERİ OKUMA**

**Hazırlayan
ENVER ALİOĞLU**

2011010225045

Tez Danışmanı

Yrd.Doç.Dr. CİHAN MIZRAK

KARABÜK-2016

**ARDUİNO İLE PARK SENSÖRÜ BLUETOOTH HABERLEŞME
VE ARDUİNO İLE DİJİTAL KUMPASTAN VERİ OKUMA**

ENVER ALİOĞLU

Karabük Üniversitesi

Mühendislik Fakültesi

Mekatronik Mühendisliği Anabilim Dalında

Lisans Tezi

Olarak Hazırlanmıştır

KARABÜK

AĞUSTOS / 2016

KABUL VE ONAY

ENVER ALİOĞLU tarafından hazırlanan " ARDUİNO İLE PARK SENSÖRÜ
BLUETOOTH HABERLEŞME VE ARDUİNO İLE DİJİTAL KUMPASTAN VERİ
OKUMA " başlıklı bu tezin Lisans Bitirme Tezi olarak uygun olduğunu onaylarım.

23/08/2016

Tez Danışmanı
Yrd.Doç.Dr. Cihan Mızrak

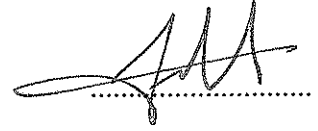


Bu çalışma, jürimiz tarafından oy birliği / oy çokluğu ile Mekatronik Mühendisliği
Anabilim Dalında Lisans Bitirme Tezi olarak kabul edilmiştir.

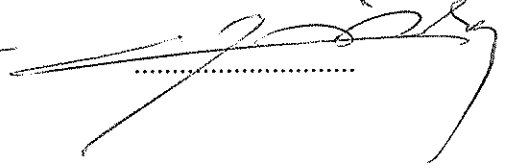
23/08/2016

Tez Jürisi

Başkan: Yrd.Doç.Dr. İbrahim ÇAYIROĞLU



Üye : Yrd.Doç.Dr. Cihan Mızrak

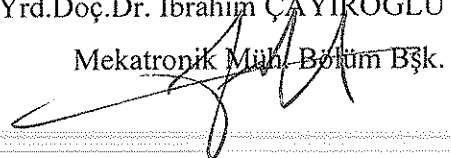


Üye :

.....

KBÜ Mühendislik Fakültesi, Mekatronik Mühendisliği Mezuniyet Komisyonu ve Bölüm
Başkanlığı bu tezi Lisans Bitirme Tezi olarak onamıştır. 23/08/2016

Yrd.Doç.Dr. İbrahim ÇAYIROĞLU
Mekatronik Müh. Bölüm Bşk.



“Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımızı beyan ederim.”

ENVER ALIOĞLU

ÖNSÖZ

Bu çalışma ; Arduino ile dijital kumpas ölçümü yaparak dijital malzemelerin çalışma prensibini ve arduinoda okunması öğrenmek ayrıca bu makalede uzaktan izlene bilir Park Sensörünün genel bir tanıtımı yapıldıktan sonra uygulamalı iki örneği verilmiştir. Park Sensörü otomobillerin arka tamponuna yerleştirilen ultrasonik dedektörler aracılığı ile geri manevra sırasında araçla arkasındaki engel arasındaki mesafeyi ölçer. Bu ölçüm engele yaklaştıkça frekansı yükselen bip sesleri ile sürücü tarafından algılanır. Engel ile araç arasındaki mesafe 30 cm'ye kadar düştüğünde bip sesi sürekli hale gelir. Park sensörüne sahip bir araç kullanırken, park sensörü yokmuş gibi arkayı kontrol etmeniz gerekir. Çünkü bu sensörler küçük veya ince çubuk yada direk şeklindeki engelleri algılamayabilir. Ayrıca geri manevra sırasında aracın arkasına hareketli bir çocuk veya hayvan geçebileceğini de unutmamalıyız. Bu tez çalışmasının planlanmasında, araştırılmasında benden ilgi ve desteğini hiçbir zaman esirgemeyen, bilgi ve tecrübeleri ile çalışmamı bilimsel temeller ışığında şekillendiren danışman hocam Yrd.Doç. Dr. Cihan Mızrak'a sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

KABUL VE ONAY.....	iii
ÖNSÖZ.....	vi
İÇİNDEKİLER.....	vii

BÖLÜM 1

1. ARDUİNO İLE PARK SENSÖRÜ BLUETOOTH HABERLEŞME	
1.1MALZEME LİSTESİ.....	
1.2 ARDUNIO UNO	
1.3 ARDUİNO UNO HABERLEŞME.....	
1.4 HC-SR04 ULTRASONİK SENSÖR.....	
1.5 BUZZER.....	
1.6 ARDUNIO NUN PC YE TANITILMASI.....	
1.7 APP INVENTER YAPIMI.....	
1.8 PROGRAM KODU.....	

BÖLÜM 2.....

2. Arduino Hakkında Genel Bilgiler.....	
2.1 Arduino Nedir.....	
2.1.1 Kartların Yapısı Ve Donanımı.....	
2.1.2 Arduino Kartları Nasıl Geliştirilir.....	
2.1.3 Arduino Kartları İle Neler Yapılabilir.....	
2.1.4 Arduino ile Ne Yapılmaz.....	
2.2 Arduino Çeşitleri.....	
2.2.1 Arduino Uno.....	

2.2.2 Arduino Mega 2560.....	
2.3 Arduino Programlanması.....	
2.3.1 setup().....	
2.3.2 loop().....	
2.3.3 Arduino Programlama Dilinin Temel Özellikleri.....	

BÖLÜM 3.....

3. Seri Haberleşme

3.1 Seri Haberleşme Nedir.....	
3.2 Serial Port Ekranının Aurdino Programındaki Yeri.....	
3.3 Serial Monitör Kullanımı Ve Programlanması.....	

BÖLÜM 4.....

ARDUİNO İLE DİJİTAL KUMPASTAN VERİ OKUMA

4.1 MALZEME LİSTESİ	
4.2 ARDUİNO İLE DİJİTAL KUMPASTAN VERİ OKUMA AMACI	
4.3 Mikrodenetleyicilerin En Önemli Özelliği Interrupt	
4.4 Arduino Harici Kesme	
4.5 Arduino Timer Kullanımı ve Ayarlanması	
4.6 Zamanlayıcı Yazmaçlarının Ayarlanması	
4.7 Overflow Interrupt.....	
4.8 Clear Timer on Compare Match Kesmesi (CTC).....	
4.9 DİJİTAL KUMPAS DEVREMİZ	
4.10 Arduino da İnterput Kullanımı	
4.11 Kartlara Göre Kesme Pinleri	

BÖLÜM 5

KAYNAKLAR.....	
ÖZGEÇMİŞ.....	...

BÖLÜM 1

1.ARDUİNO İLE PARK SENSÖRÜ BLUETOOTH HABERLEŞME

Park Sensörünün Önemi

Park kazalarını çok büyük oranda önler. Sesli uyarılarıyla arabanın arkasındaki araba ve diğer cisimlere çarpmamızı engeller. Dikiz aynasıyla görülemeyen alçak varlıklara (küçük çocuklar, kedi, köpek vs hayvanlar) zarar vermememizi sağlar.

Gece araba kullanırken normalde göreceğimiz çoğu cismi karanlık ya da dikkatsizlik nedeniyle göremeyebiliriz. Burada imdadımıza park sensörleri yetişir. Dar bir sokakta geri geri gitmemiz gerektiğinde yolda ve kaldırımdan yola doğru çıkan tüm cisimlerin varlığından haberdar ederek ona göre davranmamızı sağlar. İki araç arasına park ederken sağımız ve solumuzdaki ya da önümüz ve arkamızdaki araçlara zarar vermemizi engellemiş olur. Park sensörlerinin her arabada bulunması gerekir.

Park sensörü,sürücüler park ederken, bazen otomobillerinin çevresindeki nesneleri görmekte zorlanır. Böyle durumlarda park sensörleri işlerini kolaylaştırır. Park sensörü, otomobille yakın çevresindeki nesneler arasındaki mesafeyi hesaplayarak sürücüye bildiren bir donanımdır. Park sensörlerinin iki çeşidi vardır. Bunlardan biri ses dalgaları gönderip alarak, diğeryse elektromanyetik alan oluşturarak çalışır. Ses dalgalarıyla çalışan park sensörleri daha yaygın olarak kullanılır.

Mesafe ölçerler : Farklı tekniklerle mesafe ölçen sistemler piyasada mevcut. Bu sensörlerin kullanım alanlarını farklı kılan ise hassasiyet, maliyet, gürültü hassasiyeti, ölçebileceği maksimum mesafe, ağırlık ve benzeri etkenlerin tasarım sırasında göz önünde bulundurulması. Bu sistemleri mesafeyi ölçme teknikleri açısından sınıflandıracak olursak;

- Ultrasonik (yüksek frekanslı ses dalgaları)
- Kızılötesi ışık
- Elektromanyetik dalga (RADAR sistemlerinin vaz geçilmezi)

- Lazer

Benim aklıma gelen sistemler bunlar. Bu sistemlerin çalışma prensipleri ise birbirine oldukça yakın. Her biri, sensörden gönderdikleri bir sinyalin dönüşteki durumunu inceleyip bu gelen sinyale göre bir mesafe çıktısı üretmeye çalışıyor. Benim uygulamasını yaptığım ve üzerinde duracağım sistem ise Ultrasonik bir sensör içeriyor.

Ultrasonik sensörlerin kullanım alanlarını sıralayacak olursak;

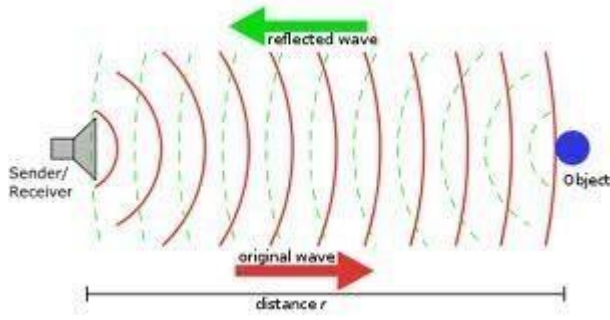
- Robotik uygulamalarında
- Mini insansız hava araçlarında
- Park sensörlerinde
- Görme engelliler için geliştirilen cihazlarda
- Çeşitli kontrol sistemlerinde
- Deniz altılarda (Sonar cihazı)
- Havacılıkta, parçaların hasarsız kontrollerinde(NDI)
- ...

Ultrasonik sensörlerin çalışma prensipleri ise genel olarak şu şekilde;

- 1- 40kHz frekansta bir ses sinyali özel bir hoparlör yardımıyla ortama gönderiliyor.
- 2- Gönderilen siyalin yankısı özel bir mikrofon yardımıyla alınıyor.
- 3- Gönderilen siyal ile gelen siyal arasındaki zaman farkı hesaplanıyor
- 4- Ölçülen zaman farkı sesin o ortamdaki hızı ile çarpıldığında ise mesafe bulunuyor.

İnsan kulağı 20Hz ile 20kHz arası frekanlardaki ses dalgalarını duyabilme yetisine sahip olduğu için bu sensörden çıkan sesleri duyamaz, hatta yaş ilerledikçe bu eşik 20kHz den aşağı düşer bu sebepten dolayı ortamda bizi rahatsız edici bir gürültü

almaz. Aşağıda bu olayı görselleştiren çizimi incellerseniz, olay daha rahat anlaşılacaktır. Aslında bu olayın su yüzeyindeki dalgalardan bir farkı yok.



1.1. MALZEME LİSTESİ

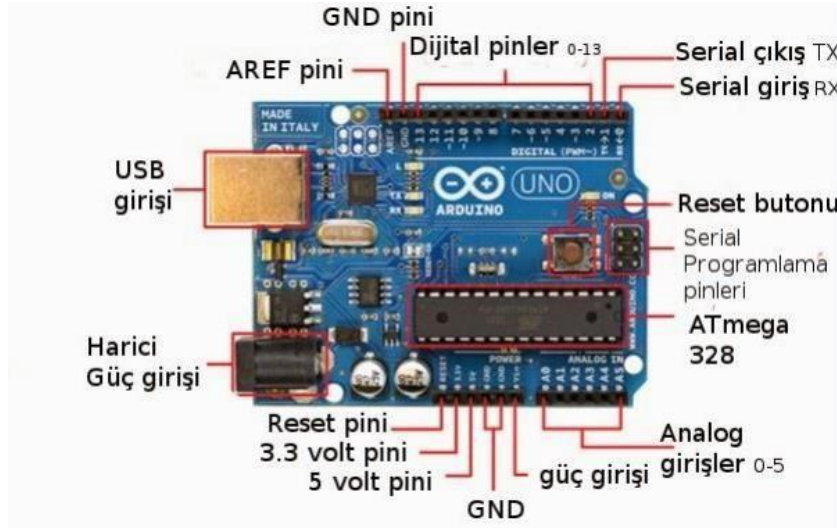
- * Arduino Uno
- * USB kablo
- * Ultrasonik Mesafe Sensörü
- * Breadboard
- * Jumper Kablolar
- * 2 adet 15 Ω direnç
- * Buzzer ve yeşil led * HC06 Arduino Bluetooth Modülü Haberleşme

1.2. ARDUNIO UNO

* Arduino Uno normal USB girişine ve adaptör girişine sahiptir. USB girişi sayesinde bilgisayara bağlayıp kolay bir şekilde kod atabilir yada bilgisayar ile haberleşmesini sağlarız. Adaptör girişi ise adaptör yada pil sayesinde arduino için gereken gücü almamızı sağlar ancak usb ile bilgisayara bağlıyken yada başka kaynaktan güç alırken bu girişi

kullanmak zorunda değiliz. Arduino Uno 3 adet led barındırır bunlardan ikisi RX ve TX ledleridir ve seri haberleşme akışı sırasında çalışırlar diğer led ise 13. pine bağlıdır bu sayede led uyarısı istediğimiz durumlarda harici led bağlamadan bu ledi kullanabiliriz.

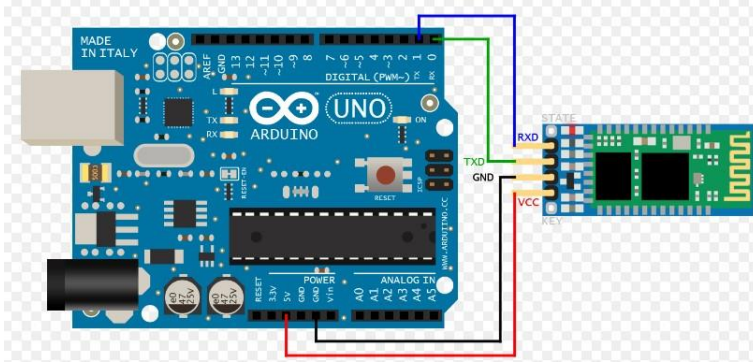
* ArduinoUno barındırdığı ATmega 328 sayesinde 32 Kb lık bir hafızaya sahiptir ancak bunun 0.5 Kb lık kısmı Arduino bootloaderi tarafından kullanılmaktadır. Ayrıca Arduino Uno 2 Kb RAM ve 1 Kb EEPROM barındırır.



1.3. ARDUİNO UNO HABERLEŞME VE BAĞLANTI

Arduino Uno sahip olduğu ATmega 328 sayesinde 0 (RX) ve 1 (TX) pinleri sayesinde seri haberleşme yapabilmektedir, ayrıca sahip olduğu USB bağlantı sayesinde doğrudan USB bağlantısında bulunabilir. Bu bağlantılar Arduino Uno kartımıza kod atmak dışında sensörlerden ölçtüğümüz verileri Arduino IDE içinde bulunan "serial monitor" sayesinde görebilirken yine bu araç sayesinde Arduino Unoya anlık komutlar verebiliriz.

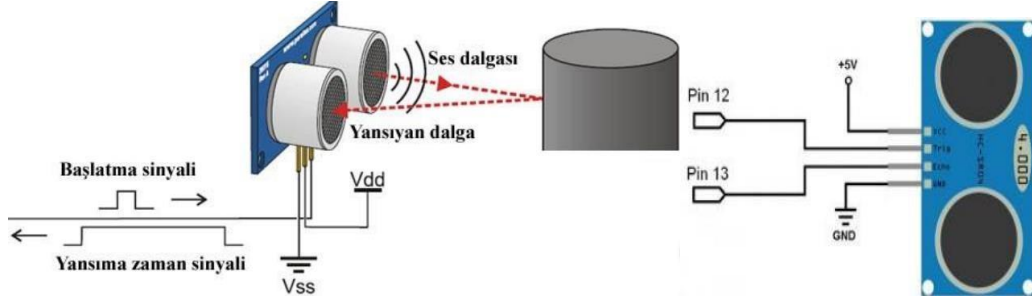
HC06 Arduino Bluetooth Modülü Haberleşme



1.4. HC-SR04 ULTRASONİK SENSÖR

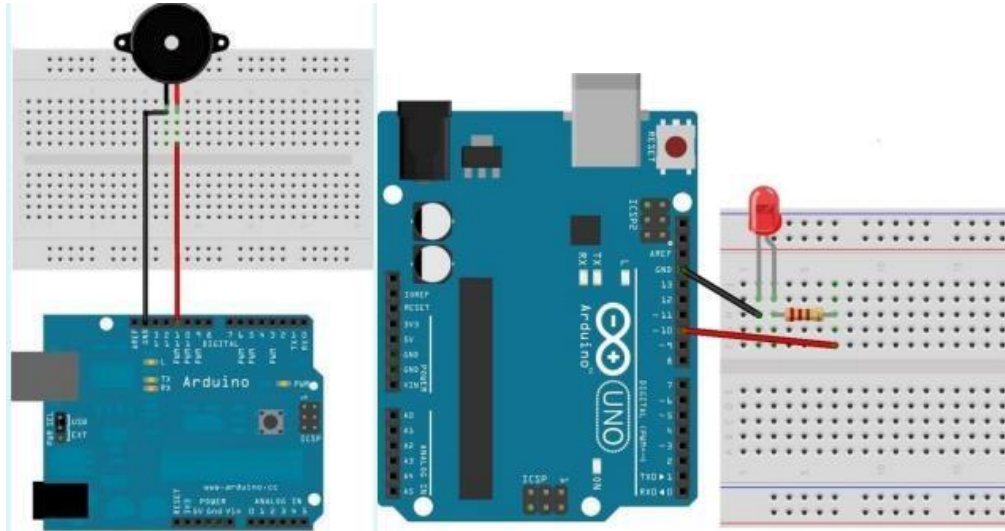
Ses dalgaları sınıflandırılmasında 20Khz- 1Ghz aralığındaki ses sinyalleri ultrasonik ses olarak tanımlanmıştır. Bizim sensörümüz ve bir çok ultrasonik sensör 40Khz frekansında ultrasonik ses üretmektedir. Burada sesin yüksekliğinde belirleyici olan etken frekanstır. Ses yüksekse frekansta yüksektir. Ultrasonik ses sinyallerini insan kulağı algılayamaz. Ultrasonik sensörlerin yapısını incelediğimizde, bir tane özel hoparlör ve bir tane özel mikrofon bulunmaktadır.Özel hoparlör sayesinde 40kHz seviyesinde ses dalgaları üretilmektedir.Özel mikrofon sayesinde de bu ses dalgaları algılanmaktadır

Sensörü kullanmak için trig pininden yaklaşık 10us'lık bir pals gönderiyoruz.Sensör kendi içerisinde 40khz frekansında bir sinyal üretip 8 pals verici transdüserine gönderiyor.Bu ses dalgası havada, deniz seviyesinde ve 15 °C sıcaklıkta 340 m/s bir hızla yol alır.Bir cisme çarpar ve geri sensöre yansır.Cismin sensörden uzaklığı ile doğru orantılı olarak echo pini bir süre lojik- 1 seviyesinde kalır ve tekrar lojik- 0 olur.Bizim bu mesafeyi ölçmek için tek yapmamız gereken echo pininin ne kadar lojik- 1 olduğunun süresini bulmaktır.



1.5. BUZZER

Elektronik cihazlarda üzerine gelen voltajın büyüklüğü ile orantılı “bip” sesi çıkaran minik hoparlörlerdir. Devreli ve devresiz buzzer olmak üzere iki çeşidi var ve devreli olanlardan daha güçlü bir ses çıkıyor. Arabaların gösterge panelinde bulunan, 2 tonlu sese sahip parça buzzerdir. Işıklar açıkken kapı açıldığında ya da bildiğimiz sinyal verildiğinde çıkan ses bu parçadan gelmektedir. Yine multimetre, park sensörleri, doğal gaz kaçak alarm cihazları, hırsız alarm devreleri dahil bir çok elektronik parçada buzzer kullanılmaktadır.



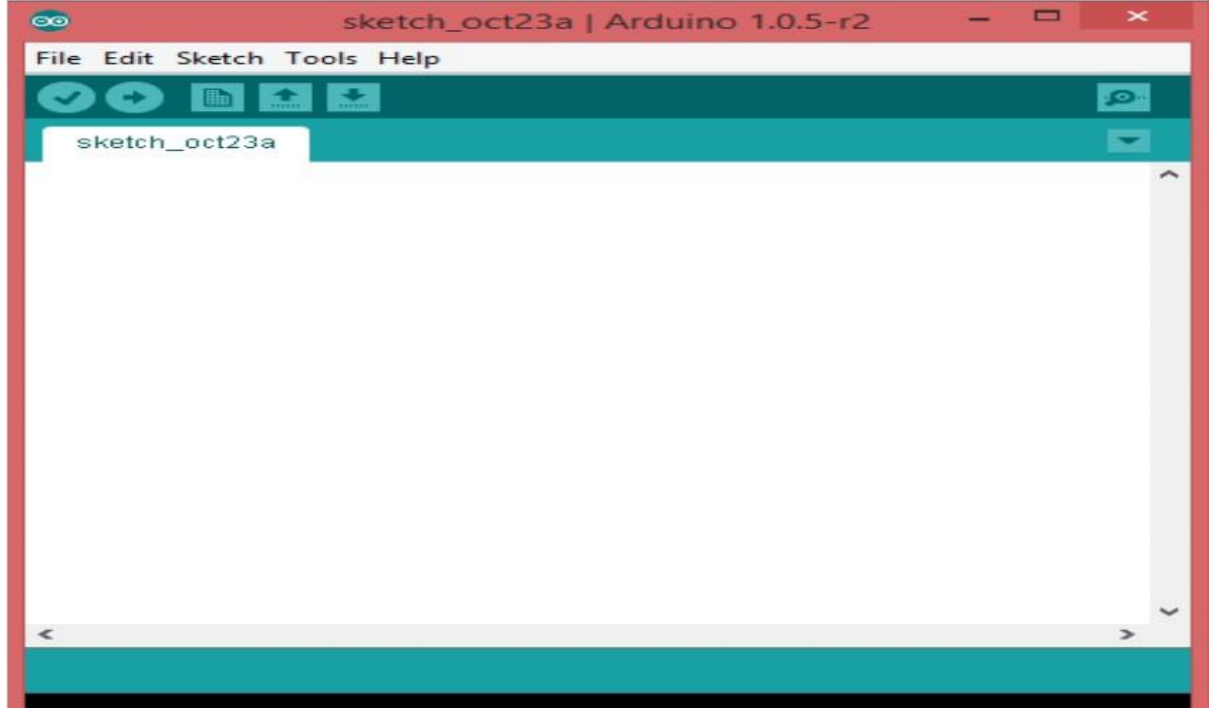
LED BAĞLANTI ŞEMASI

1.6. ARDUNIO NUN PC YE TANITILMASI

- Arduino kartımızı bilgisayarımıza tanıtmak için ilk önce arduino resmi sitesinden Arduino IDE programını indirip bilgisayarımıza kurmamız gerekiyor. Arduino UNO için basit bir yazıcı kablosuyla doğrudan bilgisayarımızın USB portuna bağlarız. Bilgisayarım (Sağ tık) > Özellikler > Aygıt Yöneticisi açılan Listedden Arduino UNO' ya çift tıklarız. Ardından sürücüyü güncelleştire tıklayıp açılan menüden gözata tıklamalıyız. Oradan arduino klasöründen driversı secince artık sürücülerimiz yüklenmiştir.

ARDUİNO PROGRAMI NASIL KULLANILIR? ARDUİNO'YA NASIL PROGRAM ATILIR?

- Arduino' ya program atmak için ilk olarak bilgisayarımıza bağladığımız Arduino' yu ve bağlantı portunu Arduino program menüsünden seçmemiz gerekiyor. Bunun için sırayla şu adımları izleyebiliriz:
- İlk olarak, programı açıp aşağıdaki ekranı görüyoruz.
- Hemen ardından, Tools menüsü altında Serial Port seçeneğine gelip bağladığımız portun yanında tik işareti yoksa tıklayıp işaretliyoruz. İkinci aşama olarak, kullandığımız Arduino modelini tanıtıyoruz.



1.7. APP INVENTER YAPIMI

MITAppInventor 2 Aracılığıyla internet üzerinden yapılabilen basit bir program sayılabilir tüm analog veri değerlerimizi burdan okuyabiliriz.Bu nedenle başka projelerde de kolayca kullanabileceğimiz güzel bir program.Arduino yazılımınızı oluştururken, android bağlantısı için ayrıca bir kod yazmanıza gerek yok. Sadece HC-06 Bluetooth modülünüzü Arduino'ya bağlamanız ve görmek istediğiniz sensör ya da herhangi değeri, Arduino IDE'deki "Serial Monitör" de görmek ister gibi yazmanız yeterli. Bunun için bildiğiniz üzere **"Serial.println();" komutunu** kullanıyoruz.(not:okuma sıklığını yüksek tutarsak android telefonlarda hiçbir sıkıntı çıkarmayacaktır.)


```

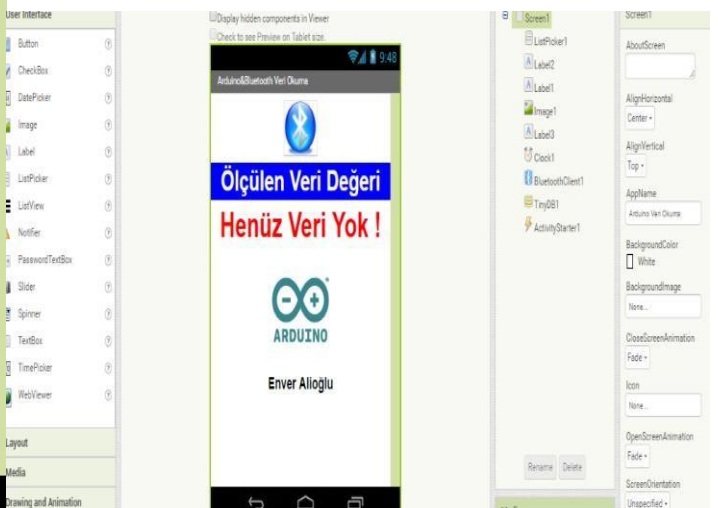
when ListPicker1.BeforePicking
do
  set ListPicker1.Elements to BluetoothClient1.AddressesAndNames

when ListPicker1.AfterPicking
do
  if call BluetoothClient1.Connect
    address ListPicker1.Selection
  then
    set ListPicker1.Text to ListPicker1.Selection
  else
    call TinyDB1.StoreValue
    tag "MAC address"
    valueToStore segment text ListPicker1.Selection
    start "1"
    length "17"
    call TinyDB1.StoreValue
    tag "MAC address and name"
    valueToStore ListPicker1.Selection

when Clock1.Timer
do
  if BluetoothClient1.IsConnected
  then
    if call BluetoothClient1.BytesAvailableToReceive > 0
    then
      set Label1.Text to call BluetoothClient1.ReceiveText
      numberOfBytes call BluetoothClient1.BytesAvailableToR

when Screen1.Initialize
do
  call ActivityStarter1.StartActivity
  if call BluetoothClient1.Connect
    address call TinyDB1.GetValue
    tag "MAC address"
    valueIfTagNotThere "0"
  then
    set ListPicker1.BackgroundColor to
    set ListPicker1.Text to call TinyDB1.GetValue
    tag "MAC address and name"
    valueIfTagNotThere "0"

```



1.8. PROGRAM KODU

```
#define trigPin 13 // dalga oluşturacak pini digital 8 e tanımlıyoruz

#define echoPin 12 // dinleme yapan pini digital 9 a tanımlıyoruz

#define buzzer 7 // buzzer için digital 13 pinini
ayarıyoruz void setup() {
pinMode(buzzer,OUTPUT); pinMode(trigPin,
OUTPUT); pinMode(echoPin, INPUT);

  Serial.begin(9600); // mesafeyi okumak için port açıyoruz
}

void loop()
{ long zaman, mesafe; digitalWrite(trigPin,
LOW); delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10); digitalWrite(trigPin,
LOW); // dalga oluşturuldu zaman =
pulseIn(echoPin, HIGH); // dinleme yapıldı
mesafe = (zaman/2) / 29.1; // zamana göre cm
hesabı yapıldı if (mesafe >= 150 || mesafe <= 0){
// mod ayarlaması yapıldı Serial.println("Engel
Yok"); digitalWrite(buzzer,LOW);
}
if(mesafe > 0 && mesafe < 15){
```

```

    Serial.print(mesafe);
Serial.println(" cm");
digitalWrite(buzzer,HIGH);
}
if(mesafe > 15 && mesafe < 30){
    Serial.print(mesafe);
Serial.println(" cm");
uyari(1);
}
if(mesafe > 30 && mesafe < 50){
    Serial.print(mesafe);
Serial.println(" cm");
uyari(2);
}
if(mesafe > 50 && mesafe < 90){
    Serial.print(mesafe);
Serial.println(" cm");
uyari(3);
}
if(mesafe > 90 && mesafe < 150){
    Serial.print(mesafe);
Serial.println(" cm");
uyari(4);
} } void uyari(int mod){ // uyarı modları için
fonksiyon yazıldı if(mod==1){
digitalWrite(buzzer,HIGH);    delay(50);
digitalWrite(buzzer,LOW);    delay(70);
}

```

```
    if(mod==2){  
digitalWrite(buzzer,HIGH);  
delay(50);  
digitalWrite(buzzer,LOW);  
delay(200);  
    }  
    if(mod==3){  
digitalWrite(buzzer,HIGH);  
delay(50);  
digitalWrite(buzzer,LOW);  
delay(350);  
    }  
    if(mod==4){  
        digitalWrite(buzzer,HIGH);  
delay(50);  
        digitalWrite(buzzer,LOW);  
delay(550);  
    }  
}
```

BÖLÜM 2

ARDUINO HAKKINDA GENEL BİLGİLER

2.1. ARDUİNO NEDİR

Arduino, Processing/Wiring dilini kullanarak çevre elemanları ile temel giriş çıkış uygulamalarını gerçekleştiren açık kaynaklı fiziksel programlama platformudur. Arduino ile bağımsız olarak interaktif uygulamalar gerçekleştirilebilirsiniz. Aynı zamanda Arduinoyu bilgisayar ile Flash, Processing, MaxMSP, C Sharp gibi bir çok yazılım üzerinden yada kendi yazdığınız yazılımlarla haberleştirerek de kullanabilirsiniz[3].

2.1.1. KARTLARIN YAPISI VE DONANIMI

Her kartın üzerinde Atmel AVR mikrodenetleyici 5V regüle entegresi, 16 MHZ kristal osilatör bulunur. Kartların çeşidine göre osilatör değişiklik gösterebilir; bunlar istisnai durumdur. Bu kartların işlevlerine, büyüklüğüne göre çeşitleri vardır, ilki kullanıcılar tarafından en çok tercih edilen arduino uno'dur. İkinci olarak; arduino mega üzerindeki portlar daha fazladır.

Haliyle işlevsel olarak uno'ya göre daha gelişmiştir. Bir diğeri arduına mega adk'dır. Adk'nın farkı Android işletim sistemine sahip akıllı telefonlar ile rahatlıkla haberleşmeyi sağlayan USB host arayüzüne sahip olmasıdır. Ve son olarak lilypad daire şeklinde bir karttır. Nitekim de en büyük özelliği yıkanabilir olmasıdır.

Programın yazılımı düşünülecek olursa; kartların mikro denetleyicisinde, önceden yüklü bir bootloader programı olduğu için ek bir yazılıma ihtiyaç duyulmaz[3].

2.1.2. ARDUİNO KARTLARI NASIL GELİŞTİRİLİR

Kartları geliştirmede c/c++ karışımı bir dil kullanılır. Kütüphanesi çok zengindir. Her türlü işleme algoritmaya açıktır. Kütüphane bilgileri firmanın resmi sitesinde yer almaktadır. Analog ve dijital girişleri sayesinde analog ve dijital verileri işleyebilirsiniz[1].

2.1.3. ARDUİNO KARTLARI İLE NELER YAPILABİLİR

Arduino kütüphaneleri ile kolaylıkla programlama yapabilirsiniz. Analog ve digital sinyalleri alarak işleyebilirsiniz. Sensörlerden gelen sinyalleri kullanarak, çevresiyle etkileşim içerisinde olan robotlar ve sistemler tasarlayabilirsiniz. Tasarladığınız projeye özgü olarak dış dünyaya hareket, ses, ışık gibi tepkiler oluşturabilirsiniz. Arduino 'nun farklı ihtiyaçlara çözüm üretebilmek için tasarlanmış çeşitli kartları ve modülleri mevcuttur. Bu kart ve modülleri kullanarak projelerinizi elıştırabilirsiniz[4].

2.1.4. ARDUINO İLE NE YAPILMAZ

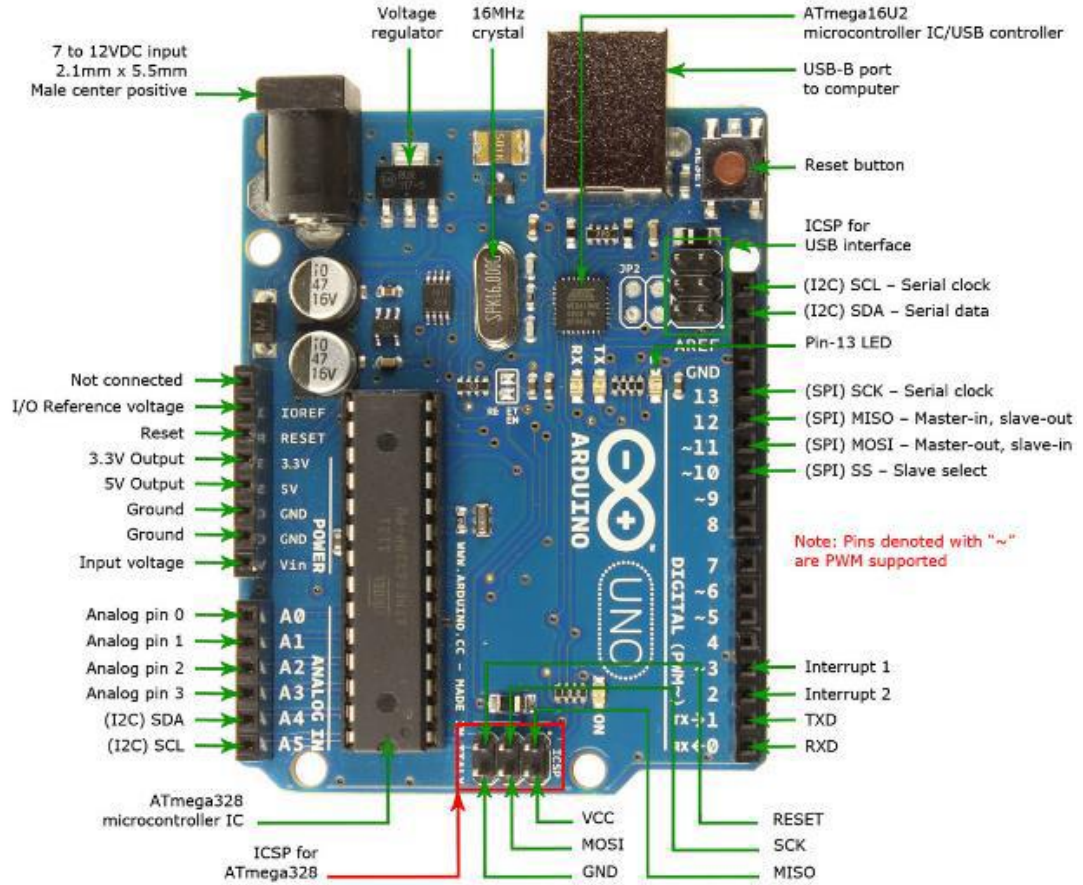
Gerçek zamanlı sinyal işleme, kamera görüntüsü aktarma gibi ağır işleri yapamayız. Fakat firma bunun üzerinde yoğun çalışmalar yapmaktadır ve kısmen başardılar sayılır. Bu kartlar üzerinde linux, android gibi işletim sistemleri çalıştırmak zordur. Bu tür çalışmalar Raspberry Pi, Beagle Bone gibi kartlarla yapılabilir. Ve atmega mikrodenetleyici yüzde yüz kullanamazsınız ki bu sorun da çözülmesi zor olan sorunlardandır[4].

2.2. ARDUINO ÇEŞİTLERİ

Arduino tüm bu avantajlı özelliklerine rağmen, tüm projelerinizi sıfır elektronik ve yazılım bilgisi ile çabucak yapabileceğiniz bir araç değildir. Hazır kütüphaneleri ve örnekleri kullanarak belli bir yerden sonra tıkanmamak için Arduino ile birlikte elektronik ve yazılım da öğrenmeniz gerekir. Genel olarak en çok tercih edilen arduino kartları Arduino UNO ve Arduino Mega 2560'dır.

2.2.1. ARDUINO UNO

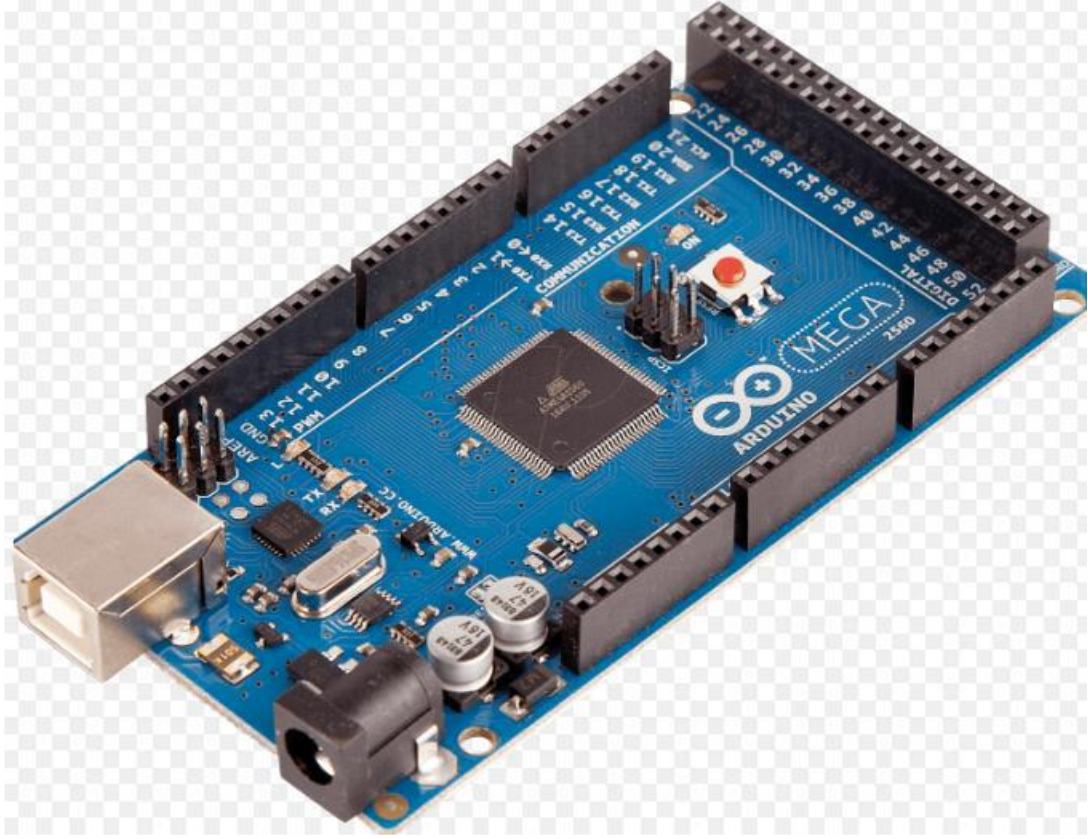
ATmega328 mikrodenetleyici içeren bir Arduino kartıdır. Arduino 'nun en yaygın kullanılan kartı olduğu söylenebilir. Arduino Uno 'nun ilk modelinden sonra Arduino Uno R2, Arduino Uno SMD ve son olarak Arduino Uno R3 çıkmıştır. Arduino 'nun kardeş markası olan Genuino markasını taşıyan Genuino Uno kartı ile tamamen aynı özelliklere sahiptir.



ŞEKİL 2.1 ARDUİNO UNO R3

2.2.2. ARDUINO MEGA 2560

Arduino Mega 2560 'ta 54 tane dijital giriş / çıkış pini vardır. Bunlardan 15 tanesi PWM çıkışı olarak kullanılabilir. Ayrıca 16 adet analog girişi, 4 UART (donanım seri port), bir adet 16 MHz kristal osilatörü, USB bağlantısı, power jakı (2.1mm), ICSP başlığı ve reset butonu bulunmaktadır. Arduino Mega 2560 bir mikrodenetleyiciyi desteklemek için gerekli bileşenlerin hepsini içerir. Arduino Mega 2560 bir bilgisayara bağlanarak, bir adaptör ile ya da pil ile çalıştırılabilir. Arduino Mega, Arduino Duemilanove ya da Diecimila için tasarlanan shield lerin çoğu ile kullanılabilir[5].



ŞEKİL 2.2. ARDUİNO MEGA 2560

2.3. ARDUINO PROGRAMLANMASI

Arduino Programlama dili basitleştirilmiş C++ kullanır. Genel olarak üç bölümden oluşur: Tanımlamalar, kurulum ve ana program bloğu[6]

```
1 //Tanımlamalar
2 void setup() {
3 // Kurulum kodları buraya yazılır. (Bir defa çalışır)
4 }
5 void loop() {
6 // Ana program kodları buraya yazılır. (Sürekli çalışır)
7 }
```

2.3.1. SETUP()

Arduino'ya enerji verildiğinde veya yeniden başlatıldığında setup() fonksiyonu bir defa çalışır. Bu fonksiyon değişkenler, pin modları, seri iletişim, kütüphaneler vb. için kullanılır[6].

2.3.2. LOOP()

Loop Türkçesi döngüdür. Adından anlaşılacağı gibi setup() fonksiyonundan sonra döngü şeklinde sürekli çalışır. Ana program kodları bu fonksiyon içine yazılır[6].

2.3.3. ARDUINO PROGRAMLAMA DİLİNİN TEMEL ÖZELLİKLERİ

Program yazımı belirli kalıpta, bloklar halinde olur.

- ☐ Bloklar, { } parantezleri ile oluşturulur.
- ☐ Komutlar aynı veya alt alta satırlara yazılabilirler.
- ☐ Tüm komutlar, noktalı virgül (;) ile biter. Yalnız blok başlatan ifadelerden sonra noktalı virgül kullanılmaz.
- ☐ Programda kullanılan tüm değişkenler ve bilgi tipleri bildirilir.
- ☐ Programın başında kütüphaneler aktifleştirilir/çağrılır.
- ☐ Açıklamalar “//” ve “/* */” (Birden fazla satır için) ile yazılır.
- ☐ #define ile eşdeğer ifade atanır, #include ile kütüphane çağrılır[6].

BÖLÜM 3

SERİ HABERLEŞME

3.1. SERİ HABERLEŞME NEDİR

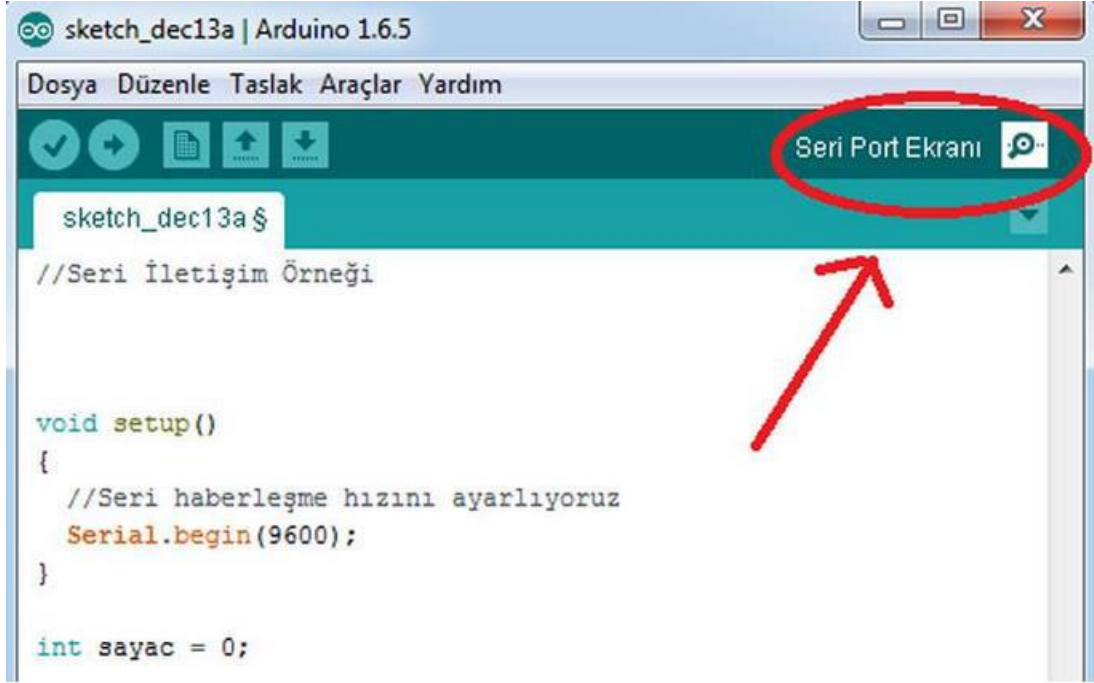
Seri haberleşme ya da seri iletişim Arduino'da ki çıkış pinlerinin verilerini bilgisayar üzerinden kontrol etmektir. Seri iletişimin Arduino üzerinde kullanılmasına ise Serial Monitor denilmektedir. Seri iletişim en basit anlamıyla dijital bilginin yani 1 ve 0'ların arka arkaya iletilmesi anlamına gelmektedir.

Seri iletişimi kullanmak için ayrı bir parçaya gerek yok çünkü Arduino'nun üzerinde seri iletişim pini bulunmakta ve Arduino seri iletişimi kendisi yapmaktadır ve tüm bu verileri Arduino üzerindeki Serial Port Ekranı'ndan görebiliriz[7].

3.2. SERIAL PORT EKRANININ ARDUINO PROGRAMINDAKİ YERİ

Seri Port Ekranından görünüşünde dikkat edilmesi gereken nokta Serial monitor'ün sağ alt köşesindeki değerle programımızdaki değerın aynı olduğuna dikkat edin. Arduino'dan gönderdiğimiz verileri düzgün görüntüleyebilmek için bu değerler aynı olmalıdır.

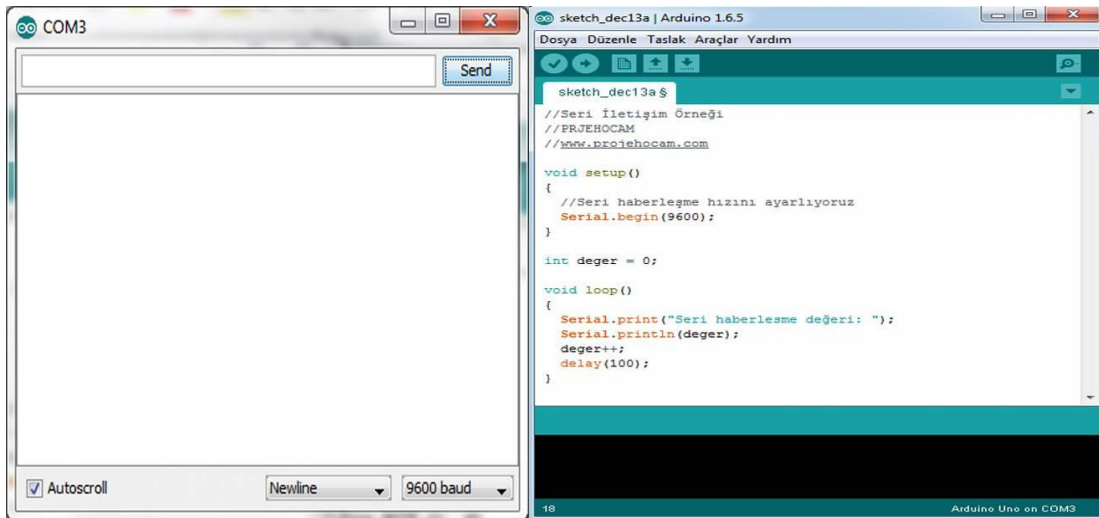
Autoscroll seçili ise ekran otomatik kaydırma ile verileri görüntülemeye devam eder. Newline da yine satırların durumuyla alakalıdır[7].



Şekil 3.1.Seri Haberleşme Port Ekranı

3.3. SERIAL MONİTOR KULLANIMI VE KODLANMASI

Arduino’da seri haberleşme için bir kütüphane bulunmaktadır. Bu kütüphanenin en çok kullanılan fonksiyonlarından print ve println’in kullanımı Şekil 3.2. gösterilmiştir.



Şekil 3.2. Seri Haberleşme Ekranı

Kodumuzda void setup() kısmında belirttiğimiz Serial.begin(9600) değeri sistemin çalışma hızını belirtir ve değeri “baud”cinsinden yazılır.

Serial.print() verilerin sıralanması belirten fonksiyon **Serial.println()** fonksiyonu sıralanması fakat verilerin alt alta sıralanmasını belirten fonksiyondur[7].

BÖLÜM 4

ARDUİNO İLE DİJİTAL KUMPASTAN VERİ OKUMA

4.1 MALZEME LİSTESİ

2N2222 TRANSİSTÖR 2 ADET

10 KOHM DİRENÇ 4 ADET

JUMPER KABLO

Dijital Kumpas Kalınlık ölçer 150mm KARBON

ARDUİNO UNO - MSP430 Launchpad

4.2 ARDUİNO İLE DİJİTAL KUMPASTAN VERİ OKUMA AMACI

Dijital veri okuyan cihazları arduino ile okumak genel olarak darbe saat mantığıyla çalışan bu cihazları arduino ile okuyup çıkış verisini görmektir. Dijital kumpaslar harici kesme(Interrupt) mantığına göre çalışmaktadır.

4.3 Mikrodenetleyicilerin En Önemli Özelliği Interrupt

Mikrodenetleyiciler ile mikroişlemciler arasında ki en büyük fark mikrodenetleyicilerin RAM, ROM, CPU, portlar gibi bütün birimleri bir aradadır. Mikrodenetleyiciler kod sırasını takip ederek tek bir işlemi gerçekleştirirler. Mikrodenetleyicilerin bu dezavantajını ortadan kısmen kaldırmanın tek yolu interrupt yani Türkçe ismi ile kesme kullanılmasıdır.

Interrupt kullanımının basit bir dille izahı şöyledir;

Bir kişi televizyon izlerken kapı çalarsa televizyon izleme işini keser ve kapıyı açıp, tekrar televizyon izlemeye devam eder. Bu da interrupt'ın işleyiş mantığını anlatan basit bir örnektir.

Ancak belirtilmelidir ki mikrodnetleyicilerde ki her bacak interrupt bacağı değıldir ve interrupt bacakları sınırlı sayıdadır. Ayrıca programlarken interruptlar arasında da öncelik sıralaması yapılması gereklidir ki, devreniz ve programınız sağlıklı çalışabilsin.

Neden Interrupt Bu Kadar Kritik Öneme Sahiptir?

Özellikle buton içeren, kritik öneme sahip pin işlevlerinde, uzun yazılıma sahip devrelerde ya da düşük hızda mikrodnetleyicilerde öncelik tanımak yaptığınız devreyi oldukça verimli kılar. Bu nedenle de interrupt kullanmaya özen gösterirseniz daha başarılı devreler oluşturabilirsiniz.

4.4 Arduino Harici Kesme

Dışarıdan müdahale ile bir fonksiyon çalıştırmak istediğimiz zaman kullanabiliriz. Harici kesmeler zamanlama problemleri çözme, encoder okuma, kullanıcı girdilerini izleme gibi konularda kullanışlıdır.

"Arduino Timer Kullanımı Kesmeler – Interrupts

4.5 Arduino Timer Kullanımı ve Ayarlanması

Merhaba arkadaşlar, bu makalemde arduino ve aynı zamanda avr 8 bit mimarisine sahip işlemcilerde timer kullanımından bahsetmek istiyorum. Arduino adı altında toplanan nerdeyse tüm hazır boardlarda kullanılan işlemciler atmega firmasına ait olan avr-8 bit veya 32 bit işlemcileridir. Bu işlemcilerde timer kullanmaktan bahsetmeden önce bu mimariye sahip tüm işlemciler benzer şekilde ayarlamalar yapabileceğinizi belirtmek isterim. Her işlemcide benim örnek vererek anlatacağım timerlar bulunmayabilir. Bunun için gerekli bilgiyi kullandığınız işlemcinin datasheet dosyasından öğrenebilirsiniz. (Örnek : [Atmega328](#))

Timer Nedir ?

Timer belirli zaman aralıklarında yapmak istediğimiz olayların gerçekleşmesinde bize yardımcı olan yapıdır. İşlemciye o zaman dilimi geldiğinde ne yapması gerektiğini hatırlatır. Bu yapılacak işlem bir dizi kendi komutunuz yada timerın desteklediği spesifik bir özellik olabilir. Örneğin klasik bir örnek olan belirli zaman aralıklarında herhangi bir çıkışa bağlı led'i yakmak istediğinizde timer kullanmadığınızı varsayarsak bu işlemi gerçekleştirmek için geçen zamanı sürekli olarak kontrol edebilir yada delay gibi kaçılması gereken komutlarla o an işlemciyi bloke edebilirsiniz. Bunları yapmak yerine işlemcide var olan bu yapılar bize gereken zaman kontrolünü sağlamaktadırlar. Bunu yaparken işlemcide diğer paralel işlemlerin devam etmesi ise bize çoklu işlem yapabilme yeteneği kazandırıyor.

Timer Nasıl Çalışır ?

Timerlar sayaç yazmacı olarak bilinen (counter register) değişkenini periyodik olarak arttırlar. Bu yazmaç belirli bir değere kadar ulaşabilir. Bu değeri belirleyen faktör ise kullandığımız timer'ın kaç bit olduğudur. Genellikle 8 bit veya 16 bit olarak karşımıza çıkarlar. Bu yazmaç maksimum değerine ulaştığı zaman (8 bit ise bu değer 255 çünkü 0'dan başlıyor) sıfıra resetlenir ve taşmanın olduğuna dair bir flag bit'i set edilir. Bu biti bizim takip etmemize gerek kalmadan Interrupt Service Routine (ISR) bu işlemi bizim için gerçekleştirerek bizim kesme olduğundan haberdar olmamızı sağlar. Bu flag bit'i için gereken tüm kontrolleri ISR sağlamaktadır. Peki timer counter değişkenini periyodik olarak artırır dedik. Bu periyodu neye göre referans alıp artırır ? Timer çalışması için bir clock sinyaline ihtiyaç duyar. Bu sinyal her oluştuğunda bu değişkeni bir arttırır. Bu sinyal harici olabileceği gibi dahili olarakta çoğu işlemcide mevcuttur.

Örneğin siz bir timer'a 16MHz clock sinyali sağlarsanız, bu timer için periyot :

$$T = 1 / f \text{ (Frekans Hz olarak)}$$

$$T = 1 / (16 * 10^6 \text{ Hz})$$

$T = 62.5 \text{ ns}$

Eğer 32Mhz olsaydı bu değer 31.25 ns olacaktır. ($1 \text{ sn} = 10^9 \text{ ns}$)

Yukarıda timerların genel çalışma mantığından bahsettik. 8-bit AVR işlemcilerinde birden çok timer mevcut. Her timer varsayılan olarak arduino için farklı işlemlerde kullanılmak üzere ayarlanmıştır. Timerlar üzerinden bilmeyerek yapabileceğimiz değişikliklerle zaman bağımlı fonksiyonlarda beklenmedik sonuçlar ile karşılaşabiliriz. (Motor, delay, millis vb.) **Varsayılan Timer Ayarları**

Timer0 – 8-bit bir zamanlayıcıdır. Counter değişkeni maksimum 255 değerini alır. Arduino için gerekli temel fonksiyonlar(delay() ve millis() gibi) bu timer’a bağlıdır. Bu timer üzerinde ne yaptığınızı bilmediğiniz sürece değişiklik yapmamak en iyisi olacaktır.

Timer1 – 16-bit bir zamanlayıcıdır. Counter değişkeni maksimum 65535 değerini alır. Servo kütüphanesi bu zamanlayıcıyı kullanmaktadır.

Timer2 – 8-bit bir zamanlayıcıdır. Counter değişkeni maksimum 255 değerini alır. tone() fonksiyonu bu zamanlayıcıyı kullanmaktadır.

Kullandığınız işlemcide bunların dışında farklı Timer3, Timer4, Timer5 gibi zamanlayıcılar bulunabilir. Gerekli bilgileri önceden bahsettiğim gibi onların pdfleri üzerinden ulaşabilirsiniz. Genel anlamda yapacağımız ayarlamaların hepsi birbiri için geçerli olacaktır.

4.6 Zamanlayıcı Yazmaçlarının Ayarlanması

Timerlar için gerekli ayarların saklanacağı yazmaçlar vardır. Bunlar üzerinde yukarıda bahsettiğimiz ayarlamaların nasıl yapılacağını birazdan örnekler üzerinde daha iyi anlayacaksınız. Bu yazmaçlardan iki tanesini timer ayarlarını set ederken kullanırız. Bunlar TCCRxA ve TCCRxB’dir. Herbir yazmaç 8 bit veri saklar.Burada

x yerine ayarlamak istediğiniz timer rakamını yazabilirsiniz. Aşağıda örnek olarak Timer1 için bu yazmaçlar verilmiştir.

TCCR1A – Timer/Counter1 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x80)	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TCCR1B – Timer/Counter1 Control Register B

Bit	7	6	5	4	3	2	1	0	
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	TCCR1B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Timer interrupt'ın periyotunu set ederken gerekli olan clock sinyalini 0 1 2 numaralı bitlerin kombinasyonlarıyla sağlarız.

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{IO}/1$ (No prescaling)
0	1	0	$clk_{IO}/8$ (From prescaler)
0	1	1	$clk_{IO}/64$ (From prescaler)
1	0	0	$clk_{IO}/256$ (From prescaler)
1	0	1	$clk_{IO}/1024$ (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Örneğin Timer1'i 2MHz clock hızında çalıştırmak istersek. Yani her count arttırımı için geçen süre 0.5us olacak biçimde ayarlarsak aşağıdaki gibi bir kod bloğu yazmamız gerekir. Ben örnekleri arduino ide'si üzerinden anlatıyorum.

```
#define PIN 13 // Board üzerinde var olan led.
```

```
Arduino Uno R3void setup()
```

```
{
```

```
pinMode(PIN, OUTPUT); // Timer1 Ayarlanması
```

```

cli();    // tüm interruptlar dev dışı

TCCR1A = 0; // tüm TCCR1A yazmacını sıfırlar

TCCR1B = 0; // tüm TCCR1B yazmacını sıfırlar // timer1 için overflow
interruptını açar
TIMSK1 |= (1 << TOIE1); //Kullandığınız işlemcide bu timer yoksa farklı bir timer
ilede aynı ayarları set edebilirsiniz

// CS11 bitini atadık vede timer şuanda clock hızında çalışacaktır.

TCCR1B |= (1 << CS11); // böleni 8 olarak ayarladık. Yani 16/8 = 2 MHZ
sei(); //Tüm interruptları çalıştır.
}

```

4.7 Overflow Interrupt

Burada yaptığımız ayarlama timer'ın ne tür bir kesme kullanacağını belirtmedik. Bunun kontrolünü sağlayan Timer/Counter1 Interrupt Mask Register TIMSK1'dir. Bu yazmaca TOIE1 ataması yaparak timer'ın taşma (overflow) interruptı kullanmasını set ederiz. Biraz daha sonra farklı bir kesme türü olan CTC (Clear Timer on Compare Match) kesmesinden de bahsedeceğim. Şimdi kaldığımız yerden devam edecek olursak overflow olduğu zaman ISR(TIMER1_OVF_vect) çağrılır.Sonuç olarak :

```

#define PIN 13 // Board üzerinde var olan
led. Arduino Uno R3void setup(){
pinMode(PIN, OUTPUT); // Timer1
Ayarlanması
cli();    // tüm interruptlar dev dışı

TCCR1A = 0; // tüm TCCR1A yazmacını sıfırlar

```

TCCR1B = 0; // tüm TCCR1B yazmacını sıfırlar // timer1 için overflow interruptını açar

TIMSK1 |= (1 << TOIE1); //Kullandığınız işlemcide bu timer yoksa farklı bir timer ilede aynı ayarları set edebilirsiniz

// CS11 bitini atadık vede timer şuanda clock hızında çalışacaktır.

TCCR1B |= (1 << CS11); // bölüni 8 olarak ayarladık. Yani 16/8

= 2 MHZ TIMSK1 |= (1 << TOIE1); // Overflow kesmesini

açarız sei();//Tüm interruptları çalıştır.

}

void loop(){}ISR(TIMER1_OVF_vect){ digitalWrite(PIN, !digitalRead(PIN));} Buraya kadar olan bölümde kesme tanımlamasını oluşturduk. Kesme frekansını hepsaplayacak olursak:

Clock sinyalimizi 8 bölüni kullanarak (CS11) set ettik yani 2MHz clock sinyali elde ettik

Bir count değeri arttırmak için geçen süre 0.5us.

Counter değeri 0 – 65535’e çıkarken 65536 değer

arttırmış olur. $T = 65536 * 0.5 = 32.77\text{ms}$ olur $f = 1/T =$

30.51 Hz

Yani bizim ISR fonksiyonumuz saniyede yaklaşık olarak 30.51 kere çalıştırılacak.

Overflow kesmesinin bazı dezavantajlarından birtanesi istediğimiz frekans değerlerinin tam olarak ayarlanabilmesinin zor olması. Çünkü her seferinde counter’ın sona kadar arttırılıp kesmenin oluşması beklenmekte. Bunun yerine daha hassas frekans oynamaları yapabilmek için timer’in bizim belirlediğimiz bir değere ulaşmasından sonra kesme oluşmasını sağlayabileceğimiz mod olan CTC modundan bahsetmek istiyorum.

4.8 Clear Timer on Compare Match Kesmesi (CTC)

Timer'ın counter değerinin overflow yapmasını beklemek yerine, bizim daha önceden set ettiğimiz bir count değerine eşit olunca oluşan bir kesmedir. Burada bu değeri atayabileceğimiz register ACRxA'dır. Burada x seçeceğimiz timer'a göre farklılık göstermektedir. Aynı zamanda atayacağımız maksimum değer bu seçtiğimiz timer'ın kaç bit olduğuna göre değişir. Yine örnek olarak timer1 kullanacak olursak atayacağımız değerler işaretli 65536'dan küçük değerler olmalıdır. Timer0 kullanacak olursak bu değer 256'dan küçüktür. Bu modu kullanmanın diğer bir avantajı timer0 kullanırken default prescale (64 böleni) kullanırsak overflow kesmesinde herhangi bir değişiklik olmadığı için diğer temel fonksiyolar doğru çalışmaya devam edecektir. Yani timer0'ıda bu mod ile kesmelerde kullanabiliriz. Fakat 8 bit olmasından dolayı uyarlayabileceğimiz frekans aralıkları sınırlı olacaktır. Burada size istediğiniz frekansı elde edebileceğiniz bir formül vermek istiyorum. Bu formül ile istediğiniz frekansa ulaşmak için gereken adım sayısını kolayca hesaplayabilirsiniz.

$$\text{Compare Değeri} = \frac{(16 \times 10^6)}{(\text{frekans} * \text{bölen})} - 1$$

Formulde yer alan frekans değeri elde etmek istediğimiz frekanstır. Bölen daha önce bahsettiğimiz işlecinin clock hızını ayarladığımız değerdir. 1 çıkarmamızın nedeni 0'dan başlıyor olmasıdır. Şimdi bir örnek ile 16KHz kesme frekansı oluşturalım.

```
#define PIN 13 // Board üzerinde var
```

```
olan led. Arduino Uno R3void setup(){
```

```
pinMode(PIN, OUTPUT); // Timer1
```

```
Ayarlanması
```

```

cli();      // tüm interruptlar dev dışı

TCCR1A = 0; // tüm TCCR1A yazmacını sıfırlar

TCCR1B = 0; // tüm TCCR1B yazmacını sıfırlar
TCCR1A |= (1 << WGM21); // CTC modunu aktif hale getirir

// CS11 bitini atadık vede timer şuanda clock hızında çalışacaktır.

OCR2A = 124; // ((16*10^6)/(16000*8))-1 Bize 16KHz kesme frekansı sağlar
TCCR1B |= (1 << CS11); // bölüni 8 olarak ayarladık. Yani 16/8 = 2 MHZ clock
sinyali TIMSK1 |= (1 << OCIE1A); // Timer Compare kesmesini aktif hale getir.

sei(); //Tüm interruptları çalıştır.
}

void loop(){ISR(TIMER1_COMPA_vect) // 16 KHz frekansa sahip
.{ digitalWrite(PIN, !digitalRead(PIN)); // Aç kapa 2 kesme tuttuğu için çıkışta
ölçeceğiniz değer 8KHz olacaktır.}

```

Kesmeler ile değişkenleri kullanırken dikkat edilmesi gereken önemli bir nokta ise volatile değişimdir. Bu tanımlama ile değişkenlerde farklı kesmelerinin değişken üzerinde sıralı işlem yapmasını sağlayıp daha stabil güvenilir bir saklama yapmış oluruz. Çok yüksek frekans değerlerinde bu deyim yeterli olmayıp kesmeyi devre dışı bırakıp değişkeni okuyup tekrar açma işlemide uygulanabilir. "

Simpson's 3 8 Method C Code

<http://sinanbir.com/arduino-timer-kullanimi-kesmeler-avr-8-bit/> (makalesinden alıntıdır)

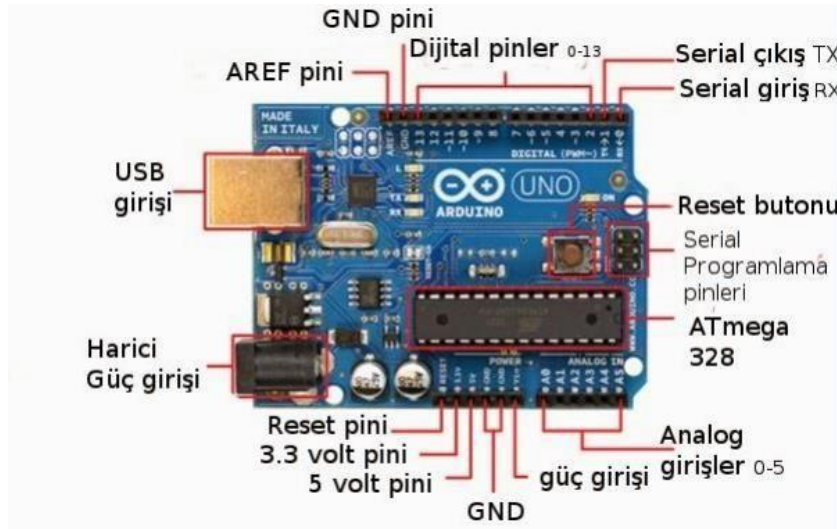
ARDUNIO UNO

* Arduino Uno normal USB girişine ve adaptör girişine sahiptir. USB girişi sayesinde bilgisayara bağlayıp kolay bir şekilde kod atabilir yada bilgisayar ile haberleşmesini sağlarız. Adaptör girişi ise adaptör yada pil sayesinde arduino için

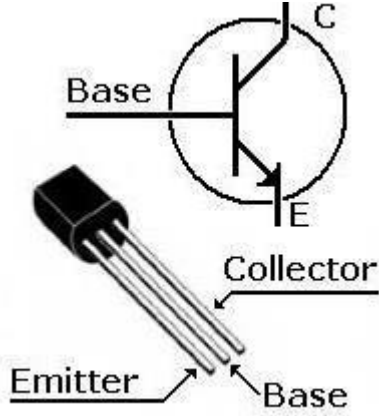
gereken gücü almamızı sağlar ancak usb ile bilgisayara bağlıyken yada başka kaynaktan güç alırken bu girişi

kullanmak zorunda değiliz. Arduino Uno 3 adet led barındırır bunlardan ikisi RX ve TX ledleridir ve seri haberleşme akışı sırasında çalışırlar diğer led ise 13. pine bağlıdır bu sayede led uyarısı istediğimiz durumlarda harici led bağlamadan bu ledi kullanabiliriz.

* ArduinoUno barındırdığı ATmega 328 sayesinde 32 Kb lık bir hafızaya sahiptir ancak bunun 0.5 Kb lık kısmı Arduino bootloaderi tarafından kullanılmaktadır. Ayrıca Arduino Uno 2 Kb RAM ve 1 Kb EEPROM barındırır.



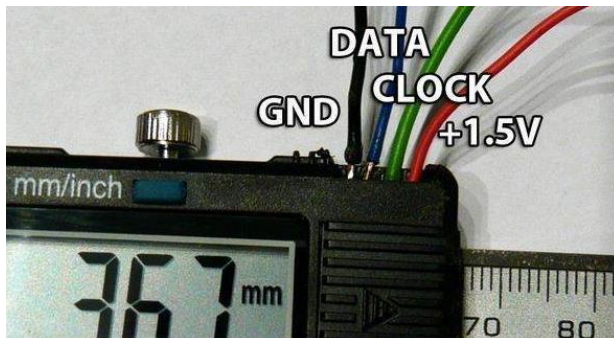
2N2222 TRANSİSTÖR



JUMPER KABLO



DİJİTAL KUMPAS

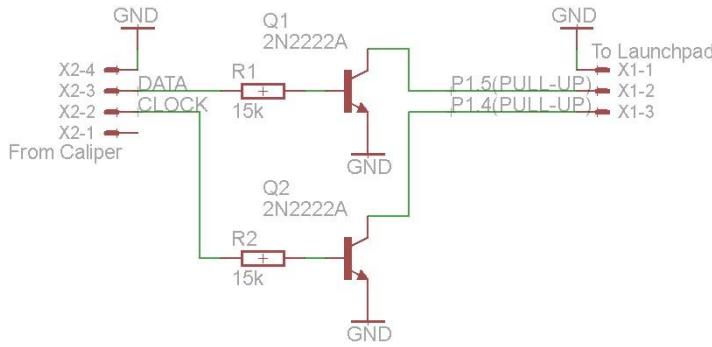


Clock 24 bit darbelerinin serisi ve daha sonra Pulse olarak gönderiliyor. 24 bit patlamaları arasındaki pulse (sabit sinyali) yaklaşık 115 msn uzunluğundadır.

Böylece, saniyede 8 kere pozisyon güncellemesi olur. Bit metrik sistem durumunda mm 100 sayısına karşılık gelen tamsayı oluşturur. inç cinsinden halinde inç ve 2000.. Ekran 3.67mm yazıyorsa Yani tamsayı 367. inç aynı okuma 0,1445 olsun, ve bu bize 289 (* 2000 0,1445) verir. kumpas sivicimiz açıksa ,bu durumda, 289 yerine 367 gösterir.

4.9 DİJİTAL KUMPAŞ DEVREMİZ

Arduinoda 5v ve 3v güç kaynaklarımız vardır. Dijital kumpaslar ise 1.5 v ile çalışmaktadır. Eğer bu farkı kapatmazsak arduino muzı yada kumpasımızı yakabiliriz bu yüzden devre kurmamız gerekir.



4.10 Arduino da İnterput Kullanımı

Arduino kullanıcılarının pek çoğu ne yazık ki interrupt kullanmayı bilmezler hatta interrupt'ın varlığından dahi haberdar değildir. Bu nedenle de programlarını kısıtlı olarak programlamak ve devrelerini ona göre dezavantajlı şekilde oluşturmaktadırlar. İşinde iyi bütün elektronikçiler bilir ki interrupt kullanımı mikrodnetleyiciler için çok kritiktir.

Arduino kartlarında ki interrupt kullanımlarının tamamı aynıdır. Ancak bilinmesi gereklidir ki her Arduino kartın interrupt bacakları farklıdır, bu bilgiye ulaşmak için Arduino kartının üstünde ki mikrodnetleyicinin datasheetine bakabilir, bir dakikadan kısa sürede bu bilgiye ulaşabilirsiniz.

4.12 Kartlara Göre Kesme Pinleri

Kartlara göre kesme kullanabileceğiniz pinlerin tablosu	
Uno, Nano, Mini	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo,	0, 1, 2, 3, 7
Zero	4 hariç bütün pinler
Due	Bütün pinler

BÖLÜM 5

KAYNAKLAR

- [1] **D.Kavak**, 2008. ‘‘İnsansız Kara Araçları Navigasyonunda Genişletilmiş Kalman (GKF) ve Sıkıştırılmış Genişletilmiş Kalman Filtre (SGKF) Tabanlı Yöntemlerinin Geliştirilmesi ve Karşılaştırılması, *Yüksek Lisans Tezi*, Y.T.Ü. Fen Bilimleri Enstitüsü, İstanbul.
- [2] *Mobile robot*. (n.d.), from http://en.wikipedia.org/wiki/Mobile_robot. Retrieved September 7, 2013
- [3] <http://www.bilgiustam.com/arduino-nedir/>
- [4] http://www.robotiksistem.com/arduino_nedir_arduino_ozellikleri.html
- [5] http://www.robotiksistem.com/arduino_mega_2560_ozellikleri.html
- [6] <http://cnrgzg.com/arduino-programlamaya-bakis/>
- [7] <http://www.projehocam.com/arduino-seri-haberlesme-serial-monitor/>
- [8] http://www.robotiksistem.com/arduino_uno_ozellikleri.html [9] <http://www.projehocam.com/l293d-motor-surucu-kullanimi/>
- [10] <http://elektronikhobi.net/l293d-surucu-entegresinin-kullanimi/>
- [11] <http://www.robotistan.com/kablolu-hc06-bluetooth-serial-modul-karti-hc06> <http://www.robotistan.com/kablolu-hc06-bluetooth-serial-modul-karti-hc06-bluetooth-to-serial-port-mbluetooth-to-serial-port-m>
- [12] H. Çakır, B. Okutan, ‘‘Ses Kontrollü Web Tarayıcı’’, Bilişim Teknolojileri Dergisi Cilt:4 Sayı: 1, Ocak 2011.
- [13] İ.Boztepe, ‘‘Sesli Komutlar İle Araç Kontrolü’’, Karadeniz Teknik Üniversitesi, Bilgisayar Mühendisliği Bölümü Ana Bilim Dalı, Trabzon, Haziran 2013

ÖZGEÇMİŞ

Enver Alioğlu 1992'de Trabzon'da doğdu; ilk ve orta öğrenimini İstanbul'da tamamladı; Pendik Lisesi, Fen Bilimleri Bölümü'nden mezun olduktan sonra 2011 yılında Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü'ne girdi. Halen bu bölümde eğitimini sürdürmektedir.

İletişim Bilgileri :

Adres: 100.Yıl Mah. 1052 Nolu Sokak Aygün Sitesi B Blok Kat:2 No:6

Merkez /KARABÜK

E-posta: enveralioglu@hotmail.com

Tel: 0534-685-45-75