

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



İNSANSIZ HAVA ARAÇLARI İÇİN KUMANDA

LİSANS BİTİRME ÇALIŞMASI

KÜBRA TÜRKOĞLU

Bilgisayar Mühendisliği Bölümü

07.07.2022

BURSA TEKNİK ÜNİVERSİTESİ
MÜHENDİSLİK VE DOĞA BİLİMLERİ FAKÜLTESİ



İNSANSIZ HAVA ARAÇLARI İÇİN KUMANDA

LİSANS BİTİRME ÇALIŞMASI

KÜBRA TÜRKOĞLU
(19360859077)

Bilgisayar Mühendisliği Bölümü

Danışman: Doç. Dr. Haydar ÖZKAN

07.07.2022

BTÜ, Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü'nün 19360859077 numaralı öğrencisi KÜBRA TÜRKOĞLU, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "BİTİRME ÇALIŞMASI BAŞLIĞI" başlıklı bitirme çalışmasını aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Danışmanı : **Doç. Dr. Haydar ÖZKAN**
Bursa Teknik Üniversitesi

Jüri Üyeleri : **Dr. Öğr. Üyesi VOLKAN ALTUNTAŞ**
Bursa Teknik Üniversitesi

Dr. Öğr. Üyesi İZZET FATİH ŞENTÜRK
Bursa Teknik Üniversitesi

Arş. Gör. Sena DİKİCİ
Bursa Teknik Üniversitesi

Savunma Tarihi : 07 Temmuz 2022

BM Bölüm Başkanı : Prof. Dr. TURGAY TUGAY BİLGİN
Bursa Teknik Üniversitesi

...../...../.....

İNTİHAL BEYANI

Bu bitirme çalışmasında görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, bitirme çalışması içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri bitirme çalışmasında kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Öğrencinin Adı Soyadı:Kübra TÜRKOĞLU

İmzası :

ÖNSÖZ

Araştırmamdaki her aşamada bana yardımcı olan değerli danışmanım Doç Dr. Haydar ÖZKAN'a, değerli bölüm başkanım Prof. Dr. Turgay Tugay BİLGİN'e ve destekleri için Dr. Öğretim üyesi Seçkin YILMAZ'a teşekkürü borç bilirim.

Temmuz 2022

Kübra TÜRKOĞLU

İÇİNDEKİLER

Sayfa

| | |
|--|------|
| İNTİHAL BEYANI..... | iii |
| ÖNSÖZ | v |
| İÇİNDEKİLER | vi |
| KISALTMALAR | viii |
| SEMBOLLER | ix |
| ŞEKİL LİSTESİ..... | x |
| ÖZET | xi |
| SUMMARY | xiii |
| 1. GİRİŞ | 15 |
| 2. UZAKTAN KUMANDA SİSTEMLERİNİN ARAŞTIRILMASI | 16 |
| 3. SERİ İLETİŞİM PROTOKOLLERİ..... | 17 |
| 3.1 Senkron Haberleşme | 17 |
| 3.2 Asenkron Haberleşme | 18 |
| 3.2.1 Hata kontrolü..... | 18 |
| 3.2.2 Uart Protokolü..... | 19 |
| 3.2.2.1 Haberleşmesi | 20 |
| 3.2.2.2 Nasıl kullanılır..... | 20 |
| 3.2.2.3 Avantajları..... | 21 |
| 3.2.2.4 Dezavantajları | 21 |
| 3.2.3 I2C protokolü | 21 |
| 3.2.3.1 Kullanım amaçları..... | 21 |
| 3.2.3.2 Kullanım Alanları..... | 22 |
| 3.2.3.3 Avantajları..... | 22 |
| 3.2.3.4 Dezavantajları | 22 |
| 3.2.6 SPI potokolü..... | 22 |
| 3.2.6.1 Avantajları..... | 24 |
| 3.2.6.2 Dezavantajları | 24 |
| 4. VERİCİ VE ALICI DEVRELERİN TASARLANMASI..... | 25 |
| 4.1 Malzemelerin Detaylı Açıklanması..... | 25 |
| 4.1.1 Arduino Uno | 25 |
| 4.1.2 NRF24L01 modülü | 25 |
| 4.1.3 Joystick..... | 29 |
| 4.1.4 Potansiyometre..... | 31 |
| 4.2 Devrelerin Tasarımı..... | 33 |
| 4.2.1 Verici Devresinin Tasarımı | 33 |
| 4.2.2 Alıcı Devresinin Tasarımı | 33 |
| 4.3 Kodların Açıklanması | 34 |
| 4.3.1 Verici Kodunun Açıklanması..... | 34 |
| 4.3.2 Alıcı Kodunun Açıklanması..... | 37 |
| SONUÇ VE ÖNERİLER..... | 39 |
| KAYNAKLAR | 40 |
| ÖZGEÇMİŞ..... | 41 |

KISALTMALAR

İHA : İnsansız Hava Aracı
UAV : Unmanned Aerial Vehicle

SEMBOLLER

| | |
|----------|-----------------|
| V | : Volt |
| I | : Amper |
| R | : Direnç Birimi |

ŞEKİL LİSTESİ

| | <u>Sayfa</u> |
|--|---------------------|
| Şekil 1 SPI çıkışları | 23 |
| Şekil 2 Antensiz nrf modülü. | 26 |
| Şekil 3 Antenli nrf modülü. | 26 |
| Şekil 4 Nrf modülü Pinout. | 27 |
| Şekil 5 nrf modül devre çizimi. | 28 |
| Şekil 6 Nrf adaptörü..... | 28 |
| Şekil 7 Joystick modülü eksen değerleri..... | 29 |
| Şekil 8 Joystick modülü pinout..... | 30 |
| Şekil 9 Joystick modülü devre şeması. | 31 |
| Şekil 10 Potansiyometre gösterimi. | 31 |
| Şekil 11 Verici devresi..... | 33 |
| Şekil 12 Alıcı devresi..... | 34 |
| Şekil 13 Struct kodları. | 35 |
| Şekil 14 Fonksiyon oluşturma. | 36 |
| Şekil 15 Void loop kodu. | 37 |
| Şekil 16 Struct oluşturma..... | 37 |
| Şekil 17 Defoult değerler..... | 38 |
| Şekil 18 Fonksiyon oluşturma. | 38 |

İHA İÇİN KONTROL SİSTEMİ

ÖZET

Günümüzde birçok alanda teknolojik ürünlerden faydalanılmasına rağmen üretim açısından yabancı firmalara bağılılığımız devam etmektedir. Bu sebeple teknolojik ürünlerin üretilmesi ve geliştirilmesi açısından yerli firmalara büyük sorumluluk düşmektedir. Bu proje ile teknolojik alanda dışa bağımlılığı azaltmak ve yerli üretimi teşvik etmek amacı güdülmektedir. Proje kapsamında kumanda, uzaktan kontrol sistemleri, incelenmiş olup yerli bir tasarım örneği sunulmuştur. Bu projede kumandanın kablosuz haberleşme ile İHA'mızı hareket ettirmek için kullanımı amaçlanmıştır. Çalışmada, biri verici diğeri alıcı olmak üzere iki modülden oluşan bir haberleşme kumanda cihazı sunulmuştur. Verici ve Alıcı Modülleri birer adet NRF24L01 antenli modüllerden oluşmaktadır. İki adet X-Y eksenli Joystick modülü ve iki adet potansiyometre ile servolarımızın hareketi sağlanmıştır. Bu uzaktan kumanda sistemi özellikle İHA'lar için tasarlanmış olup her türlü hareket kabiliyeti olan araba, robot vs. cihazlarını x ve y eksenlerinde ileri veya geri hareket ettirmek için geliştirilmiştir. Cihazların hareketlerini kontrol edecek gerekli programlar da yazılmış ve sistemde uygulanmıştır. Bu süreçte kumandanın çalışabilirliğini tespit etmek amaçlı örnek projeler denenmiştir. Sonuçta, geliştirilmiş uzaktan kumanda sisteminin geleneksel kablolu sisteme göre maliyetinin daha düşük olduğu, otomasyona daha uyumlu olduğu görülmüştür. Kablolu kablosuz arasındaki fark tespit edildikten sonra NRF24L01 modülünün antenli mi antensiz mi daha iyi olduğunu tespit etmek amaçlı aynı projeler bir de bu iki modül arasında test edilmiştir. Çıkan sonuçlardan antenli olan modülün veri aktarımının daha kuvvetli ve aradaki haberleşme mesafesinin daha fazla olduğu tespit edilmiştir. Yapılan uzaktan kontrol sistemi bir tek model iha için değil gemi araba tank vs. sistemlerinde de denenip uyarlanabilmektedir.

Araştırmalar sonucunda bazı modüllerin bozulabileceği ve deneme sürecinin yarıda kalabileceği tespit edilmiştir. Bu sonuca göre bu tarz projelerin maliyetli olabileceği fark edilmiştir, fakat yerli ve milli ürünler elde edebilmek için bu harcamaların yapılması gerektiği yadsınamaz bir gerçeği oluşturmaktadır. Dolayısıyla finansal bir sponsor ve ekipman desteği ile benzeri projelerin desteklenmesi gerekmektedir.

Arařtırmalarda bu tarz projeleri destekleyen büyük firmalar kuruluşlar olduđu tespit edilmiştir.

Anahtar kelimeler: NRF24L01, Kumanda, İHA, Arduino

CONTROL SYSTEM FOR UAV

SUMMARY

Technological products are used in many areas today, and we are dependence on foreign companies in terms of production. For this reason, domestic companies have a great responsibility in terms of producing and developing technological products. With this project, it is aimed to reduce foreign dependency in the field of technology and to encourage domestic production. Within the scope of the project, control and remote control systems were examined and an example of a domestic design was presented. In this project, it is aimed to use the remote control to move our UAV with wireless communication. In the study, a communication control device consisting of two modules, one of which is the transmitter and the other is the receiver, is presented. Transmitter and Receiver Modules consist of modules with one NRF24L01 antenna. The movement of our servos is provided with two X-Y axis Joystick modules and two potentiometers. This remote control system is especially designed for UAVs and it can be used with all kinds of mobility such as cars, robots, etc. It was developed to move devices forward or backward in the x and y axes. Necessary programs to control the movements of the devices were also written and implemented in the system. In this process, sample projects were tried to determine the operability of the controller. As a result, it has been seen that the improved remote control system is less costly and more compatible with automation compared to the traditional wired system. After determining the difference between wired and wireless, the same projects were also tested between these two modules in order to determine whether the NRF24L01 module is better with or without antenna. From the results, it has been determined that the data transfer of the module with antenna is stronger and the communication distance is longer.

The remote control system made is not for a single model UAV, but for a ship, car, tank, etc. systems can be tested and adapted. As a result of the research, it has been determined that some modules may be broken and the trial process may be interrupted. According to this result, it has been realized that such projects can be costly, but it is an undeniable fact that these expenditures must be made in order to obtain domestic and national products. Therefore, similar projects should be supported with a financial sponsor and equipment support. In the researches, it has been determined that there are large companies and organizations that support such projects.

Keywords: NRF24L01, Commander, UAV, Arduino, locality

1. GİRİŞ

Günümüzde, görsel ve sözel haberleşme ihtiyacının yanı sıra bilgi haberleşmesi de önem kazanan konulardan biridir.

Bilgi haberleşmesinde, elektronik sistemlerin kullanılmasıyla sistem çalışma hızı ve güvenilirliği gibi özelliklerde önemli oranda iyileştirmeler gerçekleştirilmiştir. Bu özelliklerden dolayı da birçok alanda rahatlıkla uygulanabilmektedir. Kullanıldıkları alanlarda kontrol noktalarının sayısının artırılabilmesi, kanal kapasitesinin genişletilebilmesi ve yapılan iletişimin güvenilir bir şekilde gerçekleşip gerçekleşmediği gibi sistem kontrol özelliklerinin de geliştirilmesini sağlamıştır. Bir merkezden, değişik yerlerdeki kontrol noktalarını idare edebilmek için kablolu uzaktan kumanda sistemleri yaygın olarak kullanılmaktaydı. Günümüzde gelişen elektronik teknolojinin getirdiği avantajlarla kablosuz uzaktan kumanda sistemleri yaygın bir şekilde kullanılmaktadır.

Mikrokontrollörlerin, elektronik haberleşme alanındaki değişik uygulamalarıyla kablosuz uzaktan kumanda sistemlerinin fonksiyonel özelliklerinde büyük gelişmeler sağlanmıştır.

Kablolu ve kablosuz uzaktan kumanda sistemleri birbiri ile karşılaştırıldığında, kablosuz kumanda sisteminin, kablolu kumanda sistemine göre,

- Maliyetinin daha düşük oluşu,
- Arazi koşullarından etkilenmemesi,
- Kurulmasının daha kolay oluşu,
- Hasar görme riskinin daha az olması,

gibi avantajlara bağlı olarak daha üstün olduğu görülmüştür.

2. UZAKTAN KUMANDA SİSTEMLERİNİN ARAŞTIRILMASI

Haberleşme alanında, elektroniğin yaygın bir şekilde kullanılmasıyla uzaktan kumanda sistemleri ortaya çıkmıştır. Haberleşmede klasik çözüm, kablo ile yapılan haberleşme iken günümüzde fiber optik kablolar ve gelişen elektronik teknolojisi ile kablosuz haberleşme gibi uygulamalar da kullanılmaktadır. Radyo linkler, telsizler, araç ve cep telefonları, güvenlik ve kontrol sistemleri gibi uzaktan kumanda sistemlerinin kullanımı ile kablo maliyeti, kablonun çekileceği güzergâhın önemi, kablonun dış etkenlerden korunması, kurulmasının pratik olmayışı, zaman, işçilik ve kullanım zorluğu gibi etkenler ortadan kalkmaktadır.

Bu tarz sistemlere günümüzden örnek vermek gerekirse akıllı ev aletleridir. Elimizde olan bütün elektronik ev aletlerini tek bir kumanda veya telefon ile yönetmek mümkün hale gelmiştir.

3. SERİ İLETİŞİM PROTOKOLLERİ

Seri haberleşme için geçerli, standartlaşmış bazı senkron ve asenkron protokoller vardır. Bunlar UART, SPI ve I2C protokolleridir. Bunlar cihazların birbiriyle veya bir ana kontrolcü ile haberleşmesini sağlar.

3.1 Senkron Haberleşme

Senkron seri iletişimde, seri verinin 1 ve 0'larının doğru sıralanıp sıralanmadığını araştırmanın yanı sıra verinin ilk bitini de belirlemeye ihtiyaç vardır. Alıcı ve verici arayüz ünitelerinin başlangıç senkronizasyonu ile bu işlem yapılır. Senkronizasyondan sonra alıcı, n bitlik bir sözcüğü oluşturmak için n tane pals alır. Güvenirliliği devam ettirmek için alıcı ve verici arayüz ünitelerinin, iletim süresi boyunca senkronizasyon içinde olması gerekir. Alıcı clockundaki gürültü ve kaymadan dolayı senkronizasyonun kaybolmasını engellemek için başlangıçtaki senkronizasyon yeterli değildir. Bunun için verici ve alıcı aynı clock sinyali ile çalışırlar. Genellikle bu clock sinyali, gönderici uçtaki clock jeneratöründen alınır. Bu iletişim moduna senkron seri iletişim adı verilir(Sinha 1988).

Senkron seri iletişimin başlangıcında verici bir seri pals gönderir. Bu gönderilen palslerin ilk birkaç biti önceden belirlenmiş olan formattadır. Bunlara match karakter veya sync pattern adı verilir. Bu match karakter, alıcıda belirli bir registerda depolanır. Alınan diğer sinyaller, match karakterle uyuyorsa alıcı match karakteri kendi alıcı registerine yerleştirir ve devamına n tane bit sayar. Seri veri transferinin senkron modunda peş peşe iki sözcüğün arasında bekleme yoktur. Veri genellikle pals dizisi şeklinde veya sözcük sayısı önceden belirlenmiş pals blokları halinde gönderilir. Verinin iletilmediği zaman aralığı genellikle null veya fill karakterlerle doldurulur. Bu karakterler, herhangi bir bilgiyi içermezler. Fakat alıcının zamanlama palslerinin akışını sağlarlar. Senkron haberleşme gerçekleştirilirken, gönderilen veri biti ve alınan veri biti birbiriyle uyum içerisinde olmalıdır. İletişimi gerçekleştirecek olan aygıtlar eş zamanlı olarak çalışmak zorundadır. Yani alıcı ve verici aynı saat (clock) üzerinde olmalıdır.

3.2 Asenkron Haberleşme

Veri iletiminin bu şeklinde, alıcı ve verici arayüz modülleri için iki ayrı clock sinyali kullanılır. Bu iki clock sinyalinin iletiminin yapıldığı süre boyunca birbiri ile çok iyi senkronizasyon içinde olması gerekir (Sinha 1988). Bu metod, alıcı modülüne bazı karmaşıklıklar getirmesine rağmen avantajı sadece iki hatta ihtiyaç duymasıdır. Eğer ortak topraklama mümkün ise tek hat yeterlidir. Çok sayıda iletişim hattı olduğunda bu metod uygundur. Örneğin telefon, telex gibi. Asenkron modunda, veri formatında start biti alıcı için başlangıç zamanlama sinyali olarak çalışır. Bu bit yardımı ile alıcı clockunu senkronize eder. Bu senkronizasyon biti alındıktan sonra önceden belirtilmiş iletilmekte olan verinin karakterleri, veri olarak kaydedilir. Eğer alıcı ve verici clock frekansları tam uyumlu değilse alıcı shift registerına son bit yükleninceye kadar geçen zaman içinde senkronizasyonda küçük bir kayıp olabilir. Buna bağlı olarak hata olasılığını yok etmek için her sözcüğün sonunda stop bitleri bulunur. Bu bitler, doğru okumayı sağlayabilecek ölçüde alıcı clockunun bozulup bozulmadığını anlayabilmek için kontrol amacıyla kullanılırlar. Eğer bozulmuşsa çerçeve hata sinyali üretilir . Asenkron haberleşme yapmak için belirli bir clock'a ihtiyaç duyulmaz. Veri herhangi bir anda iletilebilir. Belirli standartlar kullanılarak gerçekleştirilir ve Senkron haberleşmeye göre daha yavaş bir iletim olur.

3.2.1 Hata kontrolü

Dijital devreler, analog devrelere göre daha düşük gürültü bağışıklılığına sahiptir. Bir iletişim hattındaki tek bir bit hatası veya gürültü etkisi veriyi bozabilir. İletişim esnasında her hangi bir hata olduğunu tanımlamak amacıyla gönderilen, alınan verinin aynı olduğundan emin olmak için kullanılan kontrol ve hata kontrol metodları vardır. CPU sistemlerinde, en çok kullanılan hata kontrol metodu parity check'dir. Parity check metodunda, verinin içindeki 1'ler sayılır ve veri sözcüğünün sonunda extra bir bit üretilir. Bu bite, parity biti adı verilir. Parity check'in iki tipi vardır. Bunlar tek parity veya çift parity parity'dir. Tek parity'nin üretildiği metotta verinin parity biti de dahil olmak üzere bir bayt içindeki 1'lerin sayısının tek olduğunu göstermek için kullanılır. Çift parity'nin üretildiği metotta, parity biti de dahil olmak üzere verinin ABSTRACT içindeki 1'lerin sayısının çift olduğunu belirtmek için kullanılır. Parity biti genellikle tek parity için 0, çift parity için ise 1'dir. Alıcı aldığı verinin parity biti ile kendisinin ürettiği parity bitini karşılaştırır ve ikisi birbirinden farklı ise parity hata

sinyali üretir (Sinha 1988). Parity check kodu birbirini yok eden hataları gözleyemez. Örneğin mekanik bir bağlantıdan dolayı veya çevre gürültüsünden dolayı iki tane 1 sözcüğe eklenirse parity check biti değişmez. Fakat gönderilen veri hatalıdır. Parity check devresi genellikle alıcı ve verici arayüz devresi üzerinden kurulur. 8 bitlik bir sistemde, her veri sözcüğüne bir parity check eklendiğinde veri depolama alanında %12,5'luk bir kayıp olur. Diğer hata kontrol metotları daha az kayıplı depolama alanı içerirler. Bunlar yaygın olarak manyetik disk ve bantlarda kullanılırlar. Bu metod fazlalığa göre kontrol (CRC) olarak adlandırılır. Bu metodun temelinde parity check'deki gibi baytlar içindeki hata değil baytlardan oluşmuş veri blokları içindeki hatayı arama vardır. Bu toplam kontrolün birçok yolu vardır. Bunlardan biri, ayrı ayrı verinin Bit 0, bit 1 ,...,bit 7 pozisyonlarındaki değerlerinin toplanması ve veri bloğunun sonuna 8 bayt şeklinde bu toplamın ilave edilmesidir. Diğer bir yol ise veri bloku içindeki her bir baytın toplanması ve gönderilen veri bloğunun sonuna 2 bayt şeklinde bu toplamın ilave edilmesidir(Sinha 1988). Alınan veride bir hata olduğu belirlendiğinde, alma işleminin sonuna veri transferinin başarısız olduğunu veya hatanın meydana geldiğini bildirmek için geri yönde bir sinyal gönderilir. Böylece hatalı veri bloğunun yerine geçerli olan başka bir veri bloğunun gönderilmesi gerektiği bildirilir. Hata kontrolünün sonunda hatayı düzeltmek için kullanılan kodlar vardır. Bu kodlar alma işlemi sonunda bir koda göre toplanarak, ardışıl bir tanımlama içinde veri sözcüğündeki parity bitleri şeklinde oluşturulur.

3.2.2 Uart Protokolü

UART (Universal Asynchronous Receiver Transmitter), bilgisayar ve mikrokontrolcüler veya mikrokontroller ve çevre birimler arasında haberleşmeyi sağlayan haberleşme protokolüdür. Asenkron olarak çalıştığı için herhangi bir "clock" ihtiyacı duymaz. USART (Universal Synchronous Asynchronous Receiver Transmitter) ise hem senkron hem de asenkron olarak çalışabilir. UART'a göre daha gelişmiş bir protokoldür. Haberleşme mantıklı aynı şekilde çalışır ancak USART aynı zamanda senkron haberleşmeleri de gerçekleştirebilir. Yeni çıkan bir mikroişlemcinin datasheet'ine baktığınız zaman bu birimleri genelde USART birimi olarak görüyoruz çünkü USART aynı zamanda UART'ı da kapsayan bir birim olarak tasarlanmıştır.

USART, 5 ve 9 bit arası data uzunluğuna sahip veriyi taşıma özelliğine sahiptir. Ancak genel olarak 8 veya 9 bitlik kullanımlar tercih edilir.

3.2.2.1 Haberleşmesi

UART haberleşmesini gerçekleştirirken ilk olarak baudrate (veri taşıma hızı) ayarlanması gerekir. Veri taşıma hızı çok çeşitli aralıklarda olabilir ancak piyasada yaygın olarak kullanılan baudrate'ler 4800, 9600, 57600, 115200 ve mikro işlemciler için çok fazla tercih edilmese de 921600. (921600 genelde hızlı işlem gerektiren yerlerde kullanılır.) Baudrate verilerin saniyede ne kadarlık byte'ını taşıyacağını belirlememize yarar. Örneğin veri taşıma hızımızı 115200 seçersek bu bizim için saniyede yaklaşık olarak 11520 byte veri iletimi sağlayacaktır. Veri iletimi için aşağıdaki görseldeki gibi bir yapı kullanılır. Yani haberleşme işlemimiz bir başlangıç bitinden sonra data bitleri, ardından parity biti ve son olarak da bitiş biti gönderilerek sonlandırılır. Bu işlem sırasında data uzunluğu ve parity biti opsiyonel olarak değişkenlik gösterebilir. Bu haberleşme tipini kullanabilmemiz için alıcı ve vericinin veri taşıma hızlarının (baudrate) aynı olması gerekiyor (veya birbirine çok yakın değerler olması gerekiyor.) Bunun sebebi ise aktarım sırasında oluşabilecek hataları minimuma indirmek. Hata payları tolere edilebilir seviyede olması taktirde bir sorun yaratmayacaktır.

3.2.2.2 Nasıl kullanılır

USART haberleşmesi yapabilmek için mikro işlemcimizdeki daha önceden tanımlanmış olan pinleri kullanırız. Bunun için ya USB-TTL dönüştürücü ya da RS232 modülü kullanırız. Her iki modülü de RX-TX pinleri mikro işlemcimizin RX-TX pinleri ile ters olarak bağlanacak şekilde bağlantısını yaptıktan sonra iletişimi başlatabiliriz.

3.2.2.3 Avantajları

- Sadece iki kablo kullanılır.
- Saat sinyaline ihtiyaç duymaz.
- Hata denetimine izin vermek için bir eşlik biti vardır.
- Her iki taraf da buna göre ayarlandığı sürece veri paketinin yapısı değiştirilebilir
- Yaygın olarak kullanılan bir yöntemdir.

3.2.2.4 Dezavantajları

- Veri çerçevesinin boyutu maksimum 9 bit ile sınırlıdır
- Birden çok bağımlı(çevre birimi/slave) veya birden çok ana sistemi(kontrolcü/master) desteklemez.
- Her UART'ın baud hızları, birbirinin %10'u dahilinde olmalıdır.

3.2.3 I2C protokolü

I2C 2 adet pin üzerinden iletişim kurmayı sağlayan bir yazılım protokolüdür. Philips firması tarafından oluşturulmuştur. Bir entegre yada parça eğer içerisinde I2C sistemi var ise bu protokol ile kullanılabilir yani bir RAM entegresinde I2C yok ise bu protokol ile kullanılamaz. I2C protokolünde yönetici MCU lara “master” yönetilen diğer parçalara ise “slave” denir.

3.2.3.1 Kullanım amaçları

I2C protokolü normalde çok fazla pin ayrılması gereken parçaların sadece 2 adet pin kullanılarak sürülmesini amaçlar. I2C protokolünde sadece 2 pin ayrılarak aynı hat üzerine birçok RAM , EEPROM , RTC vb. parça bağlanıp kullanılabilir bu da fazladan pin ihtiyacını ortadan kaldırır. I2C protokolünde hat üzerine başka MCU larda bağlanabilir bunlar gerekli zamanlarda hem master hem slave olarak yerini alabilir.

3.2.3.2 Kullanım Alanları

- I2C, monitörlerde renk dengesi, değişen kontrast ve renk tonu için kullanılır.
- Akıllı hoparlörde ses I2C iletişim yöntemi kullanılarak değiştirilir.
- Bilgisayarlarda fan hızı gibi tanı sensörlerinin okunmasında kullanılır.
- Sistem bileşenlerinde güç kaynağının açılması ve kapatılmasında kullanılır.
- Kullanıcı ayarlarını tutan gerçek zamanlı saatlere ve NVRAM yongalarına erişimde kullanılır.

3.2.3.3 Avantajları

- Çoklu master ve çoklu slave'i desteklediği için çoğu proje uyarlanabilir bir iletişim yöntemidir.
- Yalnızca iki kablo kullanır ve çok yönlüdür, farklı bağımlı cihazların gereksinimlerine kolayca uyum sağlayabilir

3.2.3.4 Dezavantajları

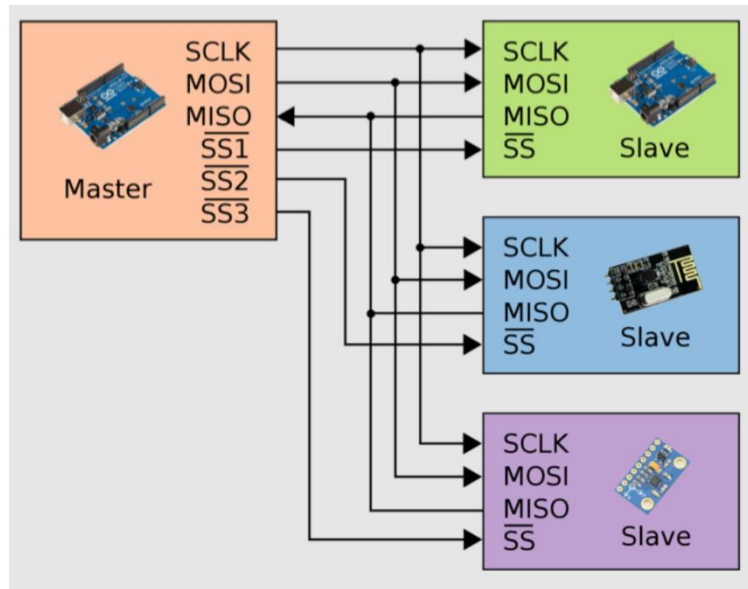
- Uzun mesafeli iletişim için geçerli değildir. Yalnızca iki kablo kullanıldığı için yavaş hıza sahiptir.
- Direnç kullanımı nedeniyle daha fazla alan gerektirir
- Cihaz sayısı arttıkça karmaşıklaşır

3.2.6 SPI potokolü

SPI (Serial Peripheral Interface), Arduino'nun desteklediği senkron seri haberleşme türlerinden biridir. Özellik ve kullanım olarak I2C'ye benzer. Bir Arduino'nun diğer Arduino veya sensörlerle kısa mesafede haberleşmesini sağlar. SPI protokolünde de I2C'de olduğu gibi bir adet Master cihaz bulunur. Bu cihaz hatta bağlı çevresel cihazları kontrol eder. Master ve çevresel cihazlara bağlanan MISO (Master In Slave Out), MOSI (Master Out Slave In) ve SCK (Serial Clock) olmak üzere üç adet SPI hattı bulunur.

- **MISO:** Çevresel cihazlardan (slave) yollanan verilerin master cihaza aktarıldığı hattır.
- **MOSI:** Master cihazdan yollanan verilerin çevresel cihazlara aktarıldığı hattır.
- **SCK:** SPI haberleşmesinde senkronu sağlayan saat sinyalinin bulunduğu hattır. Saat sinyali master cihaz tarafından üretilir.

MISO ve MOSI hatlarından da anlaşıldığı gibi SPI protokolünde I2C'den farklı olarak veri hatları tek yönlüdür. Ayrıca çevresel cihazların (slave) adreslerinin olmasına gerek yoktur. Her çevresel cihazın seçim ayağı bulunur. Bu ayağa, SS (Slave Select) denir. Bu hattın sayısı kullanılan çevresel cihazların sayısı kadardır. Her cihaz için master cihazından ayrı SS hattı çıkar. SS hattı LOW (0 volt) düzeyinde olan çevresel cihaz, master cihaz ile iletişime başlar.



Şekil 1 SPI çıkışları

Örnek bir SPI haberleşme hattı Şekil1’de gösterilmiştir. Şekilde de görüleceği üzere, Master cihazdan çevresel cihaz sayısı kadar SS çıkışı bulunur. Master cihaz iletişime geçmek istediği çevresel cihazın SS pinini LOW (0 Volt) düzeyine çeker.

3.2.6.1 Avantaları

- Asenkron seri haberleşmeden daha hızlıdır
- Veri alma donanımı basit bir shift register olabilir
- Birden fazla çevre birimini(alıcı) destekler

3.2.6.2 Dezavantajları

- Diğer iletişim yöntemlerinden daha fazla sinyal hattı (kablo) gerektirir
- İletişim önceden iyi tanımlanmış olmalıdır (istediğiniz zaman rastgele miktarda veri gönderemezsiniz)
- Denetleyici tüm iletişimi kontrol etmelidir (çevre birimleri doğrudan birbirleriyle konuşamaz)
- Genellikle her bir çevre birimi için ayrı CS hatları gerektirir; bu, çok sayıda çevre birimine ihtiyaç duyulursa sorunlu olabilir.

4. VERİCİ VE ALICI DEVRELERİN TASARLANMASI

Öncelikle Verici devremizde kullanacak malzemeler belirlenir.

- 1 Adet Arduino Uno
- 1 Adet Nrf24L01 Modül
- 6 Adet Servo Motor
- 4 Adet Direnç
- 4 Adet Led

Kumandayı test etmek istediğimiz sistemde kullanılacak malzemeler belirlenir.

- 1 Adet Arduino Uno
- 1 Adet nRF24L01 Modül
- 2 Adet Joystick
- 2 Adet Potansiyometre
- 4 Adet Buton

4.1 Malzemelerin Detaylı Açıklanması

4.1.1 Arduino Uno

Arduino, elektronik projeler oluşturmak için kullanılan, oldukça kolay donanıma ve yazılıma sahip açık kaynaklı bir elektronik platformdur. Tüm Arduino kartlarının ortak bir özelliği vardır, o da mikro denetleyicidir. Bir mikro denetleyici temelde küçük bir bilgisayardır.(5) Fiyatlarının uygun olması bakımından projelerde yaygın olarak kullanılmaktadır.

4.1.2 NRF24L01 modülü

Şekil 2’de gösterilen gösterilen Nordic firması tarafından geliştirilen 2.4GHz frekans bandında hem alıcı hem de verici özelliğinde çalışan düşük güç tüketimine sahip bir dijital radyo frekans kablosuz haberleşme çipidir.



Şekil 2 Antensiz nrf modülü.

Antenli ve antensiz olmak üzere iki tane modeli vardır. Her iki modelin bağlantısı aynı olmakla beraber aralarındaki tek fark kablosuz iletişim menzildir. Antensiz modelinde açık alanda 100 metreye kadar haberleşme sağlanabilirken antenli modelinde menzil 1000 metreye kadar varabilmektedir.

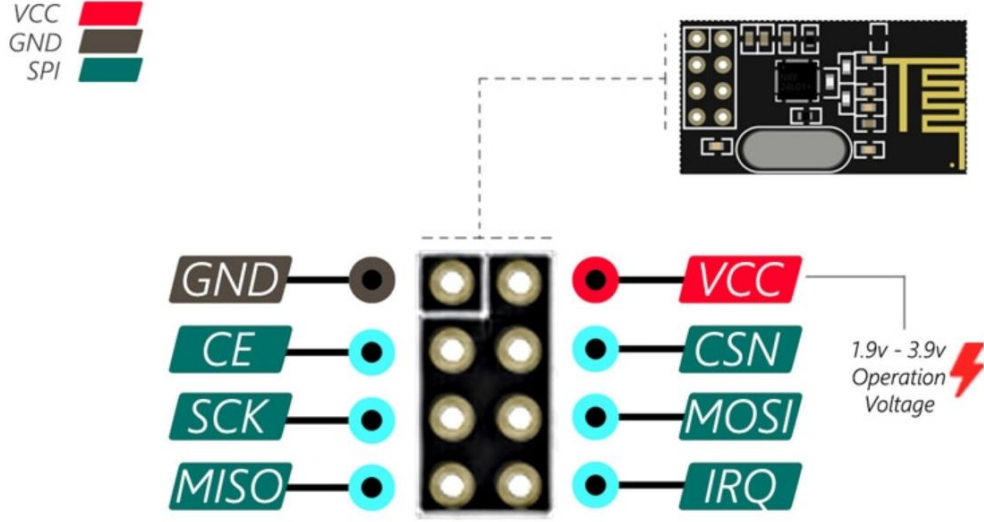


Şekil 3 Antenli nrf modülü.

SPI arabirimini desteklemekte olup, Bu sayede tam altı tane Arduino'nun bir ana Arduino ile haberleşmesi sağlanabilmektedir.

Pinout

Modülün pinlerine ve kullanımına değinmek gerekirse, modülde toplam sekiz tane pin bulunmaktadır. Projede aktif olarak bu pinlerin yedi tanesini haberleşmek için kullanılmaktadır.



Şekil 4 Nrf modülü Pinout.

GND Toprak pinidir.

VCC Güç pinidir. 1,9 ila 3,9 arasında bir voltaj verilebilmektedir. Arduino ile kullanımda 3.3V çıkışa bağlanmaktadır. 5V pinine kesinlikle bağlanılmaması gerekmektedir.

CE aktif bir HIGH pinidir. Seçildiğinde, NRF24L01 hangi moduna bağlı olarak veri gönderir ya da alır.

CSN aktif bir LOW pinidir ve normalde HIGH tutulmaktadır. Bu pin LOW olduğunda NRF24L01, SPI portunu veri alımı moduna geçirmektedir.

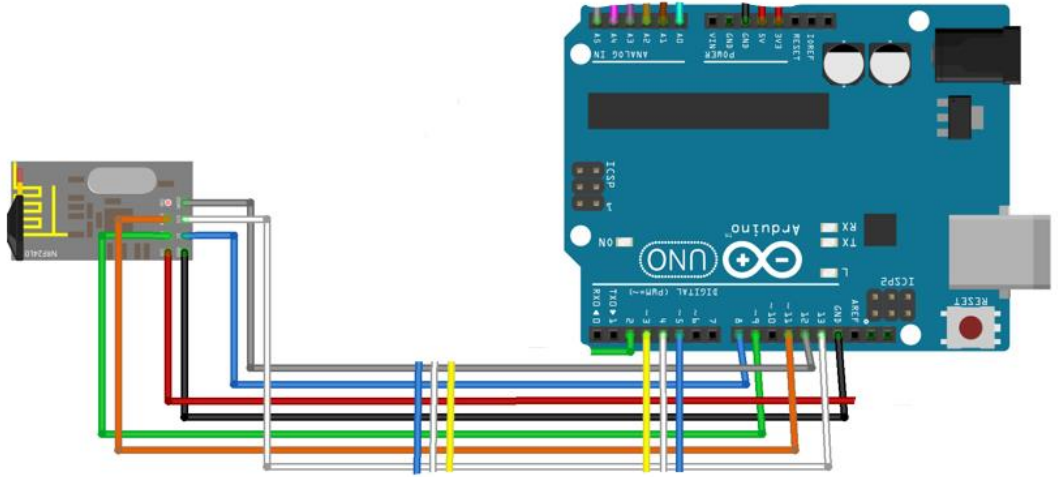
SCK SPI haberleşme için gerekli olan saat hareketlerini kabul etmektedir.

MOSI NRF24L01'in SPI girdisidir.

MISO NRF24L01'den SPI çıkışıdır.

SCK, MISO ve MOSI pinleri SPI haberleşme ile alakalı olduğundan dolayı Arduino çeşidine göre farklılık gösterebilmektedir.

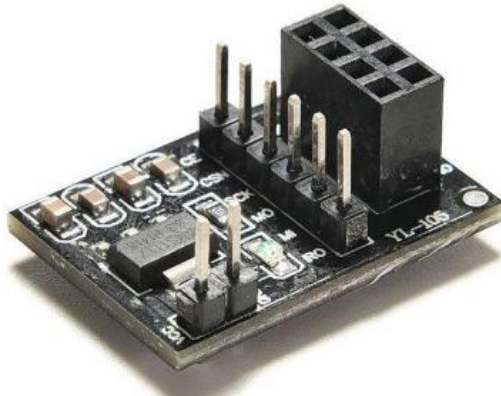
Devre Şeması



Şekil 5 nrf modül devre çizimi.

Ek bilgiler

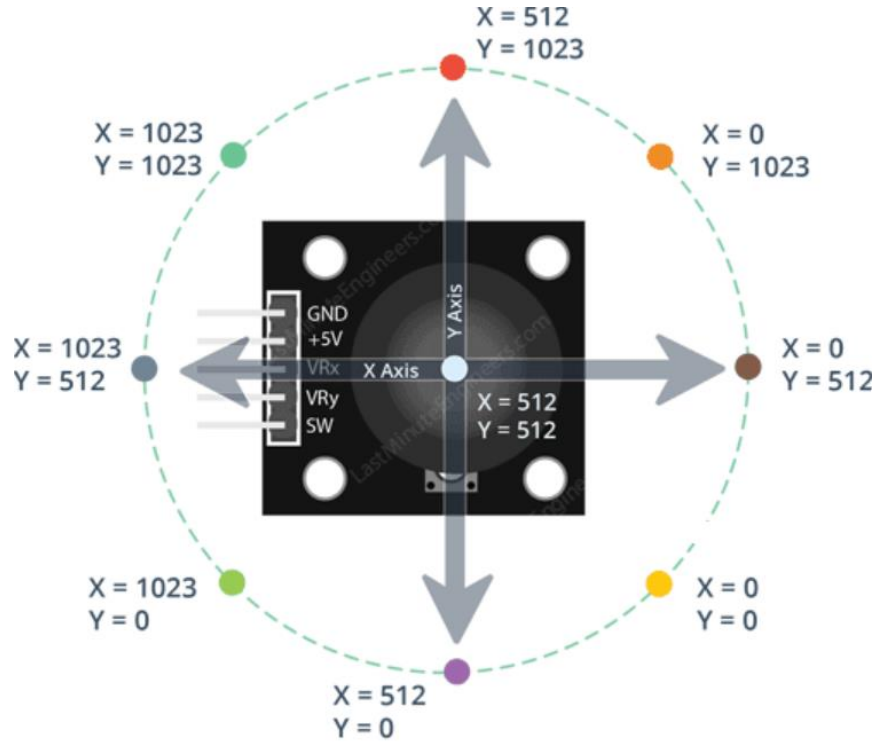
NRF24L01 modülünü genel olarak tek başına kullanmamaktır. Kullandığımız zaman ise aktarmak istediğimiz veriler düzgün bir şekilde aktarılamamaktadır. Bu modülün yanında NRF24 adaptörü bulunmaktadır. 5V-3.3V VCC Adaptör Kartı, Arduino gibi 5V sistemlerde NRF24L01 tipi alıcı-vericileri kullanmanızı sağlamaktadır. 5V girişi 1.9 ~ 3.6V DC'ye kadar düzenler ve güvenilir çalışma için bypass kapasitörlerini içermektedir.



Şekil 1 Nrf adaptörü

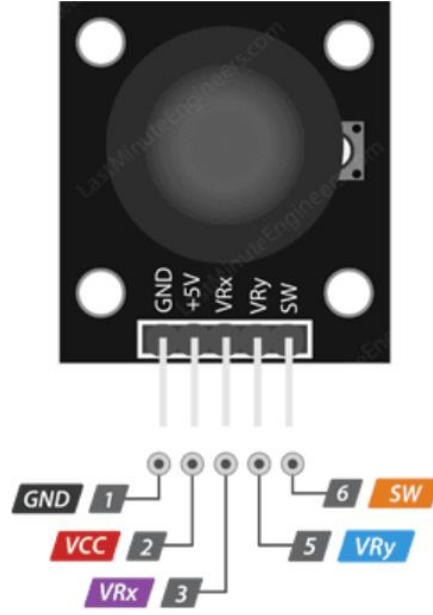
4.1.3 Joystick

Kumanda kolunun fiziksel konumunu okumak için, bir potansiyometrenin direncindeki değişikliği ölçmemiz gerekmektedir. Bu değişiklik ADC kullanılarak bir Arduino analog pini tarafından okunabilir. Arduino kartı 10 bitlik bir ADC çözünürlüğüne sahip olduğundan, her analog kanaldaki (eksen) değerler 0 ile 1023 arasında değişebilmektedir. Bu nedenle, çubuk X ekseninde bir uçtan diğer uca hareket ettirilirse, X değerleri değişecektir. 0 ile 1023 arasında benzer bir değer hareketi de Y eksenini boyunca hareket ettirildiğinde gerçekleşmektedir. Kumanda kolu orta konumda kaldığında değer 512 civarındadır. Şekil 7'deki grafik X ve Y yönlerini gösterir ve ayrıca joystick çeşitli yönlerle itildiğinde çıkışların nasıl tepki vereceğine dair bilgi edinmenizi sağlar. (2)



Şekil 2 Joystick modülü eksen değerleri

Pinout



Şekil 3 Joystick modülü pinout.

GND ; Arduino'ya GND pinini bağladığımız Toprak Pinidir.

VCC ; Modüle güç sağlamaktadır. Arduino'nuzdaki 5V çıkışa bağlanabilmektedir.

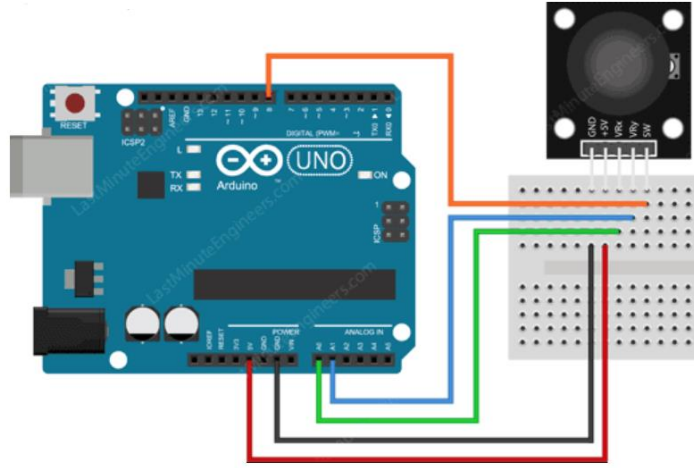
VRx ; Kumanda kolunun yatay yönde (X koordinatı) okunmasını sağlar; yani kumanda kolunun ne kadar sola ve sağa itildiğini ölçmektedir.

VRy ; Joystick'in dikey yönde okunmasını sağlar (Y koordinatı), yani joystick'in ne kadar yukarı ve aşağı itildiğini göstermektedir.

SW ; Basmalı düğmeden gelen çıktıdır. Normalde açıktır, yani SW pininden dijital okuma HIGH olacaktır. Düğmeye basıldığında GND'ye bağlanarak LOW çıktı vermektedir.

Devre Çizimi

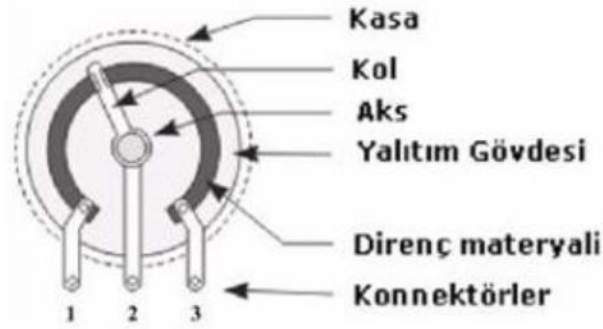
Joystick'in X ve Y koordinatlarını belirlemek için, joystick'in her iki analog çıkışlarını Arduino'daki analog pinlere bağlamamız gerekmektedir. Arduino kartımız için VRx'i Arduino'nun **A0** analog pinine ve VRy'yi Arduino'nun **A1** analog pinine bağlarız. Joystick üzerindeki orta düğmenin basılı olup olmadığını okumak için kumanda kolunun SW pinini Arduino'nun dijital pin **8**'ine bağlarız. Bunun dışında, joystick sadece güce ihtiyaç duymaktadır. VCC pini Arduino'nun 5V terminaline ve GND pini Arduino'nun GND terminaline bağlanır.



Şekil 4 Joystick modülü devre şeması.

4.1.4 Potansiyometre

Potansiyometre, bir diğer adıyla reosta, direnç çeşitlerinden biridir. Potansiyometrenin özelliği kontrol edilebilir direnç olmasıdır. Elektronikğin temel elemanlarından biridir ve kontrol gerektiren devrelerin birçoğunda bulunmaktadır.



Potansiyometrenin İç Yapısı

Şekil 5 Potansiyometre gösterimi.

Potansiyometre 3 bacaklı bir devre elemanıdır bu bacakların ikisi iç yapısında sabit, üçüncü bacak ise iç yapısında hareketli bir yapıya sahiptir. Bu yapı sayesinde sabit olan diğer iki baktan değişken biri voltaj çıkışı sağlanabilmektedir. Ayarlı orta uç, direnç üzerinde hareket edebilir. Bu değerin değişimi sayesinde gerilim bölme yani çıkış gerilimini ayarlamış olunmaktadır.

Avantajları

- Voltaj ayarlamak için düşük maliyetlidir.
- Kullanımı diğerlerine göre daha kolaydır.
- Kullanım amacına göre teknoloji açısından kolay anlaşılabilir durumdadır.

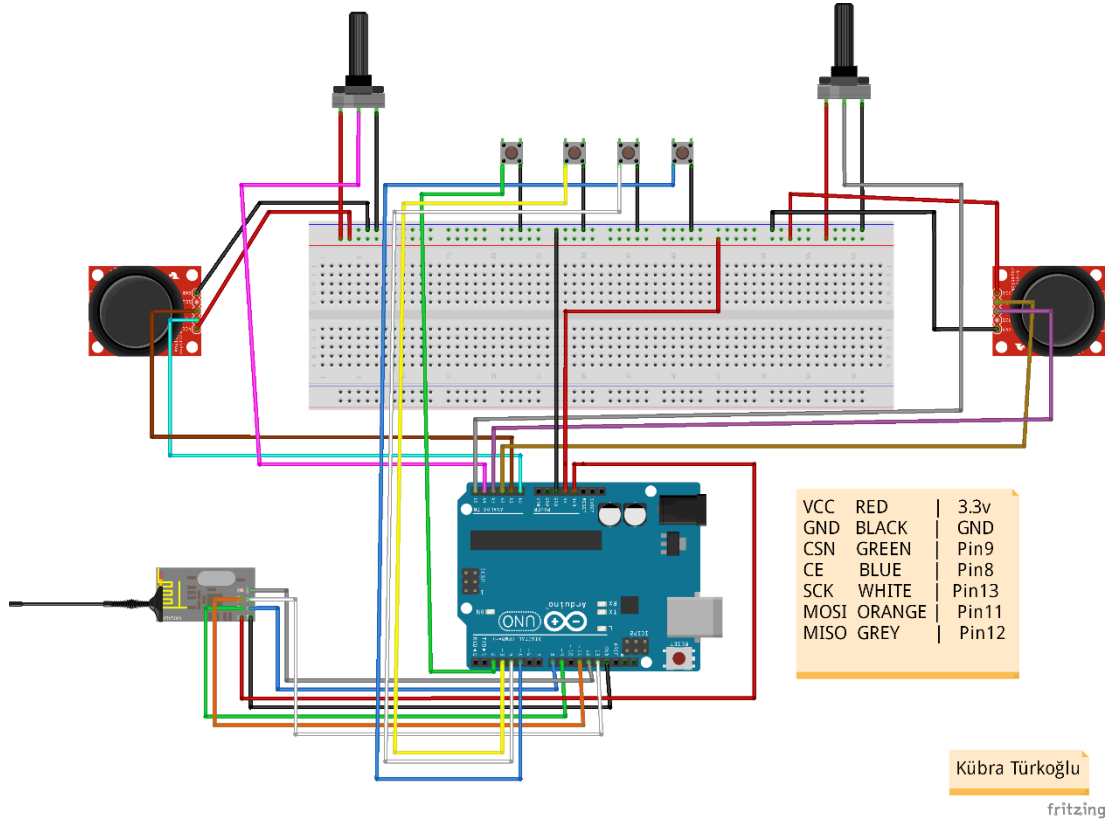
Dezavantajları

- Çok kullanımdan kaynaklı iç parçalarda sürtünmeden kaynaklı aşınma meydana gelmektedir.
- Yenilenme özelliği düşüktür.

4.2 Devrelerin Tasarımı

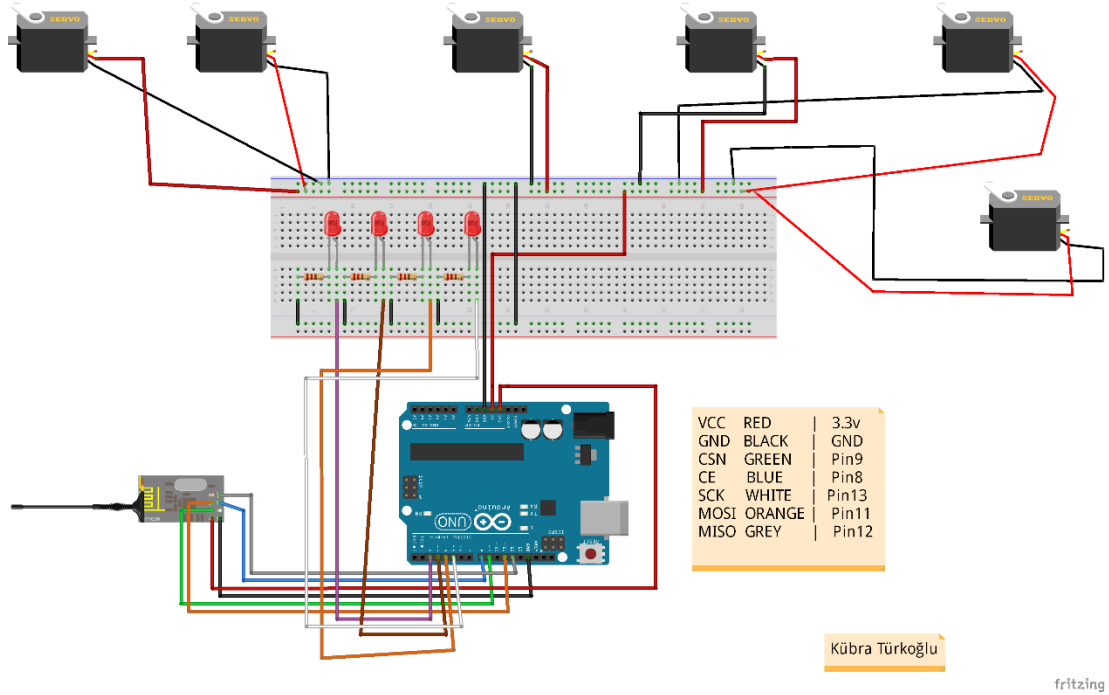
Devrelerin çizimi gerekli kütüphaneler yüklenip Fritzing uygulaması kullanılarak devre şeması elde edilmiştir.

4.2.1 Verici Devresinin Tasarımı



Şekil 11 Verici devresi.

4.2.2 Alıcı Devresinin Tasarımı



Şekil 12 Alıcı devresi.

4.3 Kodların Açıklanması

4.3.1 Verici Kodunun Açıklanması

Öncelikli olarak koda Nrf kütüphaneleri eklenmektedir. Bunlardan RF24.h'ı Arduino'nun kendi kütüphanelerinden indirip yüklerken diğer nrf kütüphanesini internette yükledikten sonra ekliyoruz.

```
#include <SPI.h>
```

```
#include <nRF24L01.h>
```

```
#include <RF24.h>
```

NRF24L01 Modüllerinin birbiri ile haberleşmesi için aralarında belirli bir şifre belirlememiz gerekmektedir. Şifre belirlemeyi birkaç şekilde yapabilmekteyiz örneğin aşağıdaki gibi yapılabilir:

const byte verici_adresi[]={.....}; şeklinde bir değişken tanımlaması yapılır. Burada verici modülümüzün veya alıcı ise alıcı modülümüzün eşleşmesi için gerekli olan 5 adet adres verisi tanımlanır. Verici kodlarında da alıcı kodlarında da bu 5 adet adres

verisi bire bir aynı olmak zorundadır aksi halde iki modül birbiri ile iletişim kuramazlar.

Örneğin Verici için:

```
const byte verici_adresi[]={ 0xFF , 0xFF , 0x00 , 0x11 , 0x01 };
```

Örneğin Alıcı için: `const byte alici_adresi[]={ 0xFF , 0xFF , 0x00 , 0x11 , 0x01 };`

Kodda yapılan ise:

const uint64_t verici_adresi = 0xF9E8F0F0E1LL; //Burada değerimiz alıcıdaki şifre ile aynı olmalı

Bir Struct yapısı oluşturulur.

```
struct kumanda_deger
{
    byte lxAxisValue;
    byte lyAxisValue;
    byte rxAxisValue;
    byte ryAxisValue;
    byte lPotValue;
    byte rPotValue;
    byte switch1Value;
    byte switch2Value;
    byte switch3Value;
    byte switch4Value;
};
```

Şekil 13 Struct kodları.

Struct yapısı neden kullanılır? (5)

Birbirleriyle ilişkili değişkenlerin, bir isim altında toplanmasına struct adı verilir. Structlar, değişik veri tiplerinde elemanlar içerebilirler ve dosya içinde tutulacak kayıtları oluşturmakta kullanılmaktadır.

Vericiden gelen verilerde birbiri ile bağlantılı olduğu için struct yapısı kullanılır.

`kumanda_deger data; //struct yapısından bir değişken oluşturuyoruz.`

Void Setup kısmında:

Haberleşme için gerekli kodları giriyoruz.

```
radio.begin();
radio.setDataRate(RF24_250KBPS);
radio.openWritingPipe(verici_adresi);
radio.stopListening(); //Radio transmitteri başlat
```

Kumandadaki buton değerlerini tanımlıyoruz.

```
pinMode(2,INPUT_PULLUP); //2. pini pullup olarak belirliyoruz.
```

```
pinMode(3,INPUT_PULLUP);
```

```
pinMode(4,INPUT_PULLUP);
```

```
pinMode(5,INPUT_PULLUP);
```

Joystickten alınan değerlerin map'lenmesi ve ölü nokta belirlenmesi için bir fonksiyon oluşturulur.

```
int maple_olunokta_olustur(int value, bool reverse)
{
    if (value >= 530)
    {
        value = map(value, 530, 1023, 127, 254);
    }
    else if (value <= 500)
    {
        value = map(value, 500, 0, 127, 0);
    }
    else
    {
        value = 127;
    }

    if (reverse)
    {
        value = 254 - value;
    }
    return value;
}
```

Şekil 14 Fonksiyon oluşturma.

Bu fonksiyon joystickten aldığı 0-1023 değerleri 0-254 e map'liyor yani çeviriyor. 127 gönderdiğimiz değerlerin orta noktası. Ayrıca joystick 'in hareketsiz olduğu andaki değerini de ayarlar.

Joystick değerleri 0-1023 arasında değişir. Ancak merkez değeri her zaman 511 değildir. Kullanmak istediğimiz alana göre değeri değişmektedir.

Void loop da ise

Kumandadan aldığımız veriler maplenir yani bizim kullanacağımız değer aralığına dönüştürülür. Bu sayede hassaslığı da değiştirmiş oluruz.

```

void loop()
{ // kumanda hareket ettığımız stickleri algılayıp alıcıya aktardığımız kısım
  data.lxAxisValue = maple_olunokta_olustur(analogRead(A0), false); // joystickden aldığımız verileri önce mapleyip sonrasında aktarıyoruz.
  data.lyAxisValue = maple_olunokta_olustur(analogRead(A1), false);
  data.rxAxisValue = maple_olunokta_olustur(analogRead(A2), false);
  data.ryAxisValue = maple_olunokta_olustur(analogRead(A3), false);
  data.lPotValue = map(analogRead(A4), 0, 1023, 0, 254); // potansiyometreden aldığımız değerleri mapleyip aktarıyoruz.
  data.rPotValue = map(analogRead(A5), 0, 1023, 0, 254);
  data.switch1Value = !digitalRead(2); // butonlardan aldığımız verileri aktarıyoruz.
  data.switch2Value = !digitalRead(3);
  data.switch3Value = !digitalRead(4);
  data.switch4Value = !digitalRead(5);
}

```

Şekil 15 Void loop kodu.

Mapleme işlemini yaptığımız değerler alıcı devresine aktarılır.

radio.write(&data, sizeof(kumanda_deger)); // kumandadan okuduğumuz değerleri alıcı devresine aktarıyoruz.

4.3.2 Alıcı Kodunun Açıklanması

-Öncelikli olarak kodumuza Nrf kütüphanelerini ekliyoruz. Bunlardan RF24.h'ı Arduino'nun kendi kütüphanelerinden indirip yüklerken diğer nrf kütüphanesini internette yükledikten sonra ekliyoruz.

```
#include <SPI.h>
```

```
#include <nRF24L01.h>
```

```
#include <RF24.h>
```

```
#include <Servo.h>
```

Verici devresinden farklı olarak burada Servo kütüphanesini eklenir.

```
const uint64_t alici_adresi = 0xF9E8F0F0E1LL; // alıcı şifresi
```

```
RF24 radio(8, 9); // Ce, Csn Pinlerinin tanımlanması
```

```

struct kumanda_deger // verileri tanımladığımız struct yapısı
{
  byte lxAxisValue;
  byte lyAxisValue;
  byte rxAxisValue;
  byte ryAxisValue;
  byte lPotValue;
  byte rPotValue;
  byte switch1Value;
  byte switch2Value;
  byte switch3Value;
  byte switch4Value;
};
kumanda_deger alici_verileri; //struct degiskenini olusturuyoruz

```

Şekil 16 Struct oluşturma.


```

void setInputDefaultValues()    // default değerlerini giriyoruz.
{
    // Joystick'in orta noktası (254/2=127)
    receiverData.lxAxisValue = 127;
    receiverData.lyAxisValue = 127;
    receiverData.rxAxisValue = 127;
    receiverData.ryAxisValue = 127;
    receiverData.lPotValue = 0;
    receiverData.rPotValue = 0;
    receiverData.switch1Value = LOW;
    receiverData.switch2Value = LOW;
    receiverData.switch3Value = LOW;
    receiverData.switch4Value = LOW;
}

```

Şekil 17 Default değerler.

```

void mapAndWriteValues() // aldığımız değeri mapleyip ilgili modülleri harekete geçirme işlemi
{
    servo1.write(map(receiverData.lxAxisValue, 0, 254, 0, 180));
    servo2.write(map(receiverData.lyAxisValue, 0, 254, 0, 180));
    servo3.write(map(receiverData.rxAxisValue, 0, 254, 0, 180));
    servo4.write(map(receiverData.ryAxisValue, 0, 254, 0, 180));
    servo5.write(map(receiverData.lPotValue, 0, 254, 0, 180));
    servo6.write(map(receiverData.rPotValue, 0, 254, 0, 180));

    digitalWrite(led1, receiverData.switch1Value);
    digitalWrite(led2, receiverData.switch2Value);
    digitalWrite(led3, receiverData.switch3Value);
    digitalWrite(led4, receiverData.switch4Value);
}

```

Şekil 18 Fonksiyon oluşturma.

SONUÇ VE ÖNERİLER

Araştırmalarım sonucunda Türkiye'de olan yerli yazılım eksikliğinin giderilmesi için hali hazırda olan ürünlerin ve yazılımların biz gençler tarafından eldeki kaynaklarla tekrardan üretilip yerlileştirilmesi gerekmektedir. Bunun için küçük veya büyük proje demeden her noktaya el atmamız gerekmektedir. Ben de üretmiş olduğum bu kumanda ile Türkiye' nin yerliligine katkı sağlamayı hedefledim.

En çok kullandığımız ürünlerin yerlileştirilmesinden sonra sırada bu ürünlerin geliştirilmesi ve özgünlük kazanması kalıyor.

Bu süreçte geliştirme kartları üzerinde gerekli modüller ile deneme yapmak için yeterli miktarda bütçeye sahip olmak gerekiyor. Benim de karşılaştığım deneme yaparken olan hatalar sonucu nrf modüllerim bozuldu ve tekrar almak durumunda kaldım. Bütçe konusunda araştırmalarım sonucunda Ülkemizin yerli ve milli ürünler üretmesi için TÜBİTAK Teknofest aracılığıyla veya kendi içerisinde projelereler ile üreticilere yardım sağladığını öğrendim.

KAYNAKLAR

- (4)admin. (2015, Şubat 18). *elektronik*. 2022 tarihinde elektrik rehberiniz: <https://www.elektrikrehberiniz.com/elektronik/potansiyometre-nedir-10639/> adresinden alındı
- (3)Bakkal, F. (2022, Nisan 2). *akademi*. 2022 tarihinde Robolinkmarket: <https://akademi.robolinkmarket.com/arduino-ile-nrf24l01-kullanimi/> adresinden alındı
- (2)robocombo. (2021, Temmuz 20). 2022 tarihinde robocombo: <https://www.robocombo.com/blog/icerik/arduino-ile-joystick-kullanimi#:~:text=Bir%20joystick'in%20temel%20%C3%A7al%C4%B1%C5%9Fma,nun%20i%C5%9Fleyebilece%C4%9Fi%20elektronik%20bilgilere%20%C3%A7evirmektir.adresinden%20alındı>
- (5)*Seyir Defteri*. (2013, Eylül 6). 2022 tarihinde İTÜBİDB: [https://bidb.itu.edu.tr/sevir-defteri/blog/2013/09/06/yap%C4%B1-\(struct\)-kavram%C4%B1-ve-yap%C4%B1-ile-s%C4%B1n%C4%B1f-\(class\)-aras%C4%B1ndaki-farklar#:~:text=Yap%C4%B1%20\(Struct\)%20%3A%20Birbirleriyle%20ili%C5%9Fkili,i%C3%A7inde%20tutulacak%20kay%C4%B1tladresinden%20alındı](https://bidb.itu.edu.tr/sevir-defteri/blog/2013/09/06/yap%C4%B1-(struct)-kavram%C4%B1-ve-yap%C4%B1-ile-s%C4%B1n%C4%B1f-(class)-aras%C4%B1ndaki-farklar#:~:text=Yap%C4%B1%20(Struct)%20%3A%20Birbirleriyle%20ili%C5%9Fkili,i%C3%A7inde%20tutulacak%20kay%C4%B1tladresinden%20alındı)
- (1)Turkcell. (2020, 11 26). *Gelecegiyazanlar*. 2022 tarihinde Turkcell: <https://gelecegiyazanlar.turkcell.com.tr/konu/egitim/arduino-401/spi-protokolu> adresinden alındı
- (6)Yüksel, Ö. (2021, Kasım 21). 2022 tarihinde Cenuta: <https://www.cenuta.com/blog/arduino-nedir-ne-ise-yarar-ve-nerelerde-kullanilir-arduino-nasil-kullanilir-ve-cesitleri/> adresinden alındı

ÖZGEÇMİŞ

KÜBRA TÜRKOĞLU

- Bilgisayar Mühendisliği 4.sınıf
- GSM: 0 553 984 17 27
- Mail: **kubratürkoglu16@gmail.com**
- LinkedIn Hesabı: <https://www.linkedin.com/in/kubra-turkoglu/>
- Github : <https://github.com/kubratürkoglu>

KARİYER HEDEFİ

Kendimi geliştirebileceğim, yeniliklere açık, Gömülü sistem Mühendisi veya Veri Bilimci olarak şirketlerde çalışıp belli bir alanda yetkinlik kazanmak istiyorum.

Öğrenim Durumu

- Lisans Öğrencisi
- Bursa Teknik Üniversitesi Bilgisayar Mühendisliği (2019 -2022)
- Trakya Üniversitesi Bilgisayar Mühendisliği (2018-2019)
- Bursa Anadolu Kız Lisesi (2013-2017)

YABANCI DİL VE DÜZEYİ

- İngilizce – İyi Düzeyde
- Almanca – Orta Düzeyde

BİLGİSAYAR BECERİLERİ

- C# - İyi Düzeyde
- Python- İyi Düzeyde
- HTML, Css, PHP, Js -İyi Düzeyde
- C- Orta Düzeyde
- SQL- İyi düzeyde
- Microsoft Office – İyi Düzeyde
- Java -Orta Düzeyde
- Linux - Başlangıç Düzeyde
- Proteus – İyi Düzeyde
- Matlab -Orta Düzeyde

Deneyimler

- Poyraz İHA Takımı Yazılım Görevlisi (2020 - Devam ediyor)
- Borusan-Borçelik Uzun Dönem Staj (2021-2022)
- Nergal Elektronik Yaz Stajı(2021)
- Bursa Teknik Üniversitesi Bilgi İşlem Kısmi Zamanlı Öğrenci (2020-2021)

