



**T.C.  
OSMANIYE KORKUT ATA ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ**

**LİSANS TEZİ**

**Bahtiyar BAYIR**

**ROBOT KOL VE KONTROL ARA YÜZÜ**

**MÜHENDİSLİK FAKÜLTESİ**

**OSMANIYE – 2021**

**T.C.  
OSMANİYE KORKUT ATA ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
LİSANS BİTİRME TEZİ**

**ROBOT KOL VE KONTROL ARA YÜZÜ**

**Bahtiyar BAYIR**

**ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ  
ANABİLİM DALI**

**OSMANİYE  
TEMMUZ – 2021**

## TEZ BİLDİRİM SAYFASI

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, bu çalışma sonucunda elde edilmeyen her türlü bilgi ve ifade için ilgili kaynağa eksiksiz atıf yapıldığını ve bu tezin Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırlandığını bildiririm.

Bahtiyar BAYIR



# ÖZET

## ROBOT KOL VE KONTROL ARA YÜZÜ

Bahtiyar BAYIR

Lisans, Elektrik Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi İbrahim ÖZTÜRK

Temmuz 2021, 20 sayfa

Sanayilerde ve üretim hatlarında insan gücü azalmış, yerini otomasyon sistemlerine bırakmaya başlamıştır. Bu sistemler insan gücüne kıyasla daha verimli çalışmakta ve insanlara özgü olan olumsuz durumları bulunmamaktadır. Bu çalışmada STM32F4-Discovery geliştirme kartı üzerinde geliştirilmiş basit seviyeli bir robot kol yapılmak istenmiştir. UART seri haberleşme protokolü ile bilgisayar ile haberleşmesi sağlanmış olup, durum takibi, görev tanımlaması ve görev listesi üzerinde değişikliklerin yapılabilmesi hedeflenmiştir.

**Anahtar Kelimeler:** Robot kol, Seri haberleşme, Gömülü sistem, Ara yüz programlama

## **TEŞEKKÜR**

Lisan bitirme tezi çalışmalarında bana yol gösteren, değerli bilgi ve birikimlerini esirgmeden sabır ve özveriyle aktaran, öğrenim hayatı sonrasında da iş hayatımda kullanabileceğim önemli bilgiler veren danışman hocam Sayın Dr. Öğr. Üyesi İbrahim ÖZTÜRK'e teşekkürlerimi sunarım.

Aynı zamanda bu süreçte yanımda olan ve desteklerini eksik etmeyen aileme ve arkadaşlarıma teşekkürlerimi sunarım.

## İçindekiler

GİRİŞ.....	1
1. KULLANILAN MODÜLLER.....	1
1.1. STM32F4-Discovery Geliştirme Kartı .....	1
1.2. PL2303 USB-TTL Dönüştürücü.....	2
1.3. MG90S Servo Motor .....	2
2. PROJE BAĞLANTI ŞEMASI.....	3
3. YAZILIM ÇALIŞMALARI.....	4
3.1. Gömülü Sistem Yazılımı.....	4
3.1.1. Çalışmada Kullanılan Kesmeler .....	4
3.1.2. Servo Motorlar İçin PWM Sinyali Oluşturma .....	6
3.1.3. Haberleşme Senaryosu .....	7
3.1.4. Ring Buffer Algoritması .....	9
3.1.5. PID Kontrol Algoritması.....	10
3.2. Kontrol Ara Yüzü Yazılımı .....	11
KAYNAKÇA.....	13

Şekil 1 - Robot kol bağlantı şeması .....	3
Şekil 2 - UART kesmesi blok diyagramı .....	4
Şekil 3 - TIM2 kesmesi blok diyagramı.....	5
Şekil 4 - TIM3 Kesmesi blok diyagramı.....	5
Şekil 5 - Bilgisayardan gelebilecek mesajların listesi.....	7
Şekil 6 - Mesaj parçalama ve işlemenin akış şeması .....	8
Şekil 7 - Ring buffer'a ilk veri geldiğindeki durum.....	9
Şekil 8 - Ring buffer'a 'd' verisi geldiğindeki durum .....	9
Şekil 9 - Ring buffer'dan veri okunduğundaki durum.....	10
Şekil 10 - PID blok şeması[4] .....	10
Şekil 11- Robot kol için tasarlanan kontrol ara yüzü .....	12

## GİRİŞ

Günümüzde ağırlıklı olarak üretim sanayisinde kullanılan robot kollar sayesinde üretim hızı artmış, hata payları da azalmıştır. Bu yapılar önceden programlanarak belirtilen görevi yerine getirmektedirler. İnsanlara kıyasla yorulma, hastalanma, ölüm gibi olumsuz unsurları bulunmadığı için üretimde aksama riskini de minimum seviyeye indirmişlerdir.

Bu çalışmada, içerisinde STM firmasının ürettiği güçlü bir mikrodenetleyici bulunduran STM32F4-Discovery geliştirme kartı kullanılarak prototip seviyesinde robot kol geliştirilmesi hedeflenmiştir.

C yazılım dili kullanılarak programlanan bu geliştirme kartı, robot kolun bütün işlevlerini yerine getirmesi beklenmiştir (bilgisayar ile haberleşme, motor kontrolü ve sürülmesi vb.).

Bilgisayar ortamında ise C# dili ile bir Windows Form Application geliştirilmiş ve seri port kullanılarak robot kol ile iletişim kurulmuş, durum takibi ve görev ataması gibi işlemlerin yerine getirilmesi hedeflenmiştir.

## 1. KULLANILAN MODÜLLER

Çalışmada kullanılan modüllerin listesi;

1. STM32F4-Discovery geliştirme kartı
2. PL2303 USB-TTL dönüştürücü
3. MG90S Servomotor
4. 3D Printer'dan üretilmiş robot kol gövdesi

### 1.1. STM32F4-Discovery Geliştirme Kartı

STM32F4-Discovery geliştirme kartı, STM32F407 yüksek performanslı mikro denetleyicisinin yeteneklerinden yararlanarak kullanıcıların kolayca ses uygulamaları geliştirmesine olanak tanır. Bir ST-LINK/V2-A yerleşik hata ayıklama aracı, bir ST-MEMS dijital ivmeölçer, bir dijital mikrofon, entegre D sınıfı hoparlör sürücüsüne sahip bir ses DAC'si, LED'ler, basmalı düğmeler ve bir USB OTG Micro-AB konektörü içerir. Özel eklenti kartları, uzatma başlık konektörleri aracılığıyla bağlanabilir.[1]

Teknik özellikleri:[1]

- FPU çekirdekli 32-bit Arm® Cortex ®-M4 mikrodenetleyici
- 1Mbyte flash bellek ve 192 Kbyte RAM



- ST MEMS 3 eksenli ivmeölçer
- ST-MEMS ses sensörü çok yönlü dijital mikrofon
- Entegre D sınıfı hoparlör sürücüsüne sahip Audio DAC
- Kullanıcı ve sıfırlama butonları
- Sekiz LED:
- USB iletişimi için LD1 (kırmızı/yeşil)
- 3,3 V güç açma için LD2 (kırmızı)
- Dört kullanıcı LED'i, LD3 (turuncu), LD4 (yeşil), LD5 (kırmızı) ve LD6 (mavi)
- İki USB OTG LED'i, LD7 (yeşil) VBUS ve LD8 (kırmızı) aşırı akım
- Kart konektörleri:
- Mikro-AB USB
- Stereo kulaklık çıkış jakı
- Prototipleme panosuna hızlı bağlantı ve kolay problama için tüm LQFP100 I/O'lar için 2,54 mm pin başlığı
- Esnek güç kaynağı seçenekleri: ST-LINK, USB VBUS veya harici kaynaklar
- Harici uygulama güç kaynağı: 3 V ve 5 V
- Eski standart kitaplıkları kullanmak için çeşitli örnekler, STM32CubeF4 MCU Paketinin bir parçası veya STSW-STM32068 dahil olmak üzere kapsamlı ücretsiz yazılım
- USB yeniden numaralandırma özelliğine sahip yerleşik ST-LINK/V2-A hata ayıklayıcı/programlayıcı: yığın depolama, Sanal COM bağlantı noktası ve hata ayıklama bağlantı noktası
- IAR Embedded Workbench®, MDK-ARM ve STM32CubeIDE dahil olmak üzere çok çeşitli Entegre Geliştirme Ortamları (IDE'ler) desteği

## 1.2. PL2303 USB-TTL Dönüştürücü

UART seri haberleşme protokolünü destekleyen cihazlar ile bilgisayar arasında iletişim sağlanması için kullanılan modüldür. İçerisinde PL2303 chipsetini barındırmaktadır.

Bu projede STM32F4 Discovery geliştirme kartı ile bilgisayar haberleşmesini sağlaması amacıyla kullanılmıştır. Pin çıkışı olarak 5V, 3.3V, RX, TX ve GND pinlerini barındırır. Sadece geliştirme kartı ile kullanılacağı durumlarda RX ve TX bağlantısı yeterlidir. Projede motorlar için harici bir besleme bağlanacağı için 5V ve GND pinleri de bağlanmıştır.

## 1.3. MG90S Servo Motor

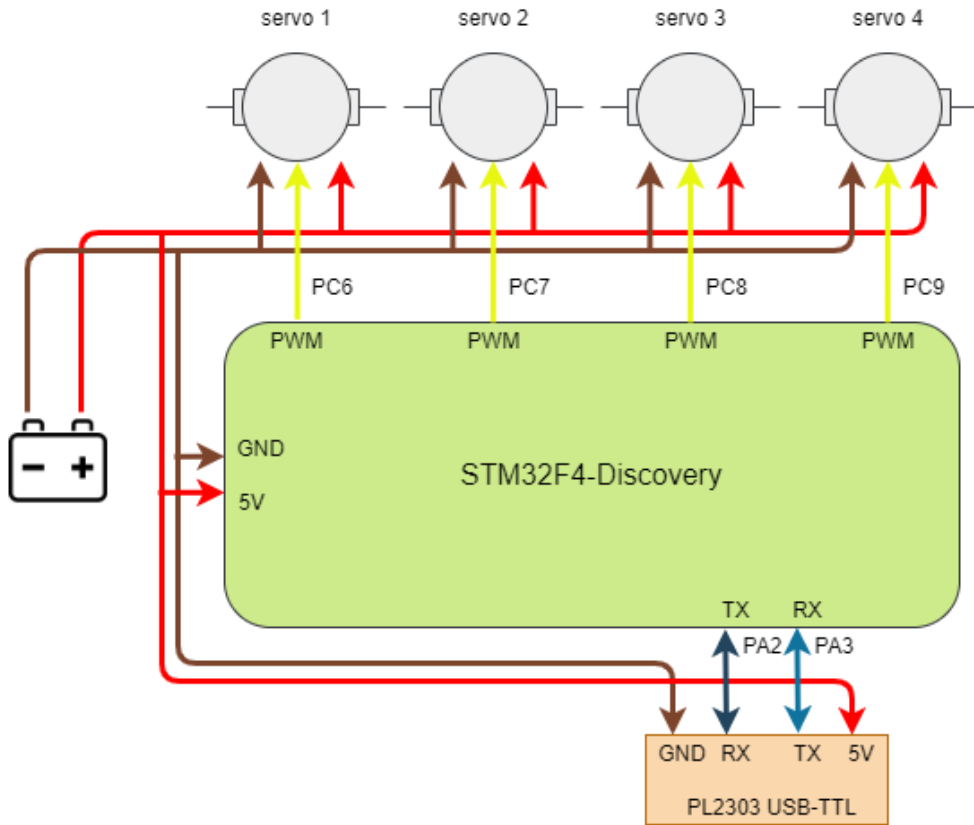
Servo, herhangi bir mekanizmanın işleyişini hatayı algılayarak yan bir geri besleme düzeneğinin yardımıyla denetleyen ve hatayı gideren otomatik aygıttır. Robot teknolojisinde en çok kullanılan motor çeşididir. Bu sistemler mekanik olabileceği gibi elektronik, hidrolik, pnömatik veya başka alanlarda da kullanılabilir.

Servo motorlar; çıkış, mekaniksel konum, hız veya ivme gibi değişkenlerin kontrol edildiği, özetle hareket kontrolü yapılan bir düzeneştir.[2]

Bu projede robot kolun belirtilen konuma hareket ettirilmesi için mg90s model servo motor tercih edilmiştir. İçerisindeki dişli setinin metal olması sayesinde muadil modeli olan sg90s model servo motora kıyasla daha dayanıklıdır. Teknik özellikleri;[3]

- Çalışma voltajı 4.8V – 6V
- Stall tork 1.8kgf.cm(4.8V), 2.2 kgf.cm(6V)
- Çalışma hızı 0.1s/60°
- PWM frekansı 50Hz

## 2. PROJE BAĞLANTI ŞEMASI



Şekil 1- Robot kol bağlantı şeması

Robot kolun bağlantı şeması yukarıda verildiği gibidir. Servo motorların kontrolü için geliştirme kartının beslemesi yetersiz kaldığı için harici bir güç kaynağından besleme alınmıştır.

### 3. YAZILIM ÇALIŞMALARI

Robot kolun geliştirilmesi iki aşamalı olarak gerçekleştirilmiştir. Bu aşamalar;

1. Gömülü sistem yazılımı
2. Kontrol ara yüzü geliştirilmesi

Şeklindedir.

#### 3.1. Gömülü Sistem Yazılımı

Bu alanda çalışmalar C dili ile Keil MDK-ARM V5.27 ortamında geliştirilmiştir. Sistem kesmeler (interrupt) ile oluşturulmuştur. 3 adet kesme ile sistem belirli periyotlarla kontrol edilerek çalışmaktadır.

Bilgisayar ile haberleşmede belleğin daha verimli kullanılabilmesi için ring buffer algoritması kullanılmıştır. Bu algoritma ile bellekte sadece belirli bir bölüm haberleşme için kullanılmıştır.

Servo motorların kontrolünde ise PID kontrol algoritması kullanılmıştır. Motorların belirlenen açıya ulaşmasında daha stabil bir çalışma hedeflenmiştir.

#### 3.1.1. Çalışmada Kullanılan Kesmeler

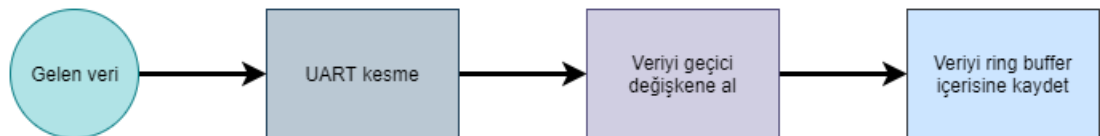
Kullanılan kesmeler;

- Bilgisayar ile haberleşme için UART kesmesi,
- ring buffer içerisindeki gelen verileri almak için TIM2 kesmesi,
- Görevlerin gerçekleştirilmesi için TIM3 kesmesi

Şeklindedir.

##### 3.1.1.1. UART Kesmesi

STM32F4-Discovery geliştirme kartında USART2 RX pinine (PA3) her bir veri geldiği zaman aktif olur. Gelen veri geçici bir değişkene alınır ve ardından ring buffer içerisine yerleştirilir.



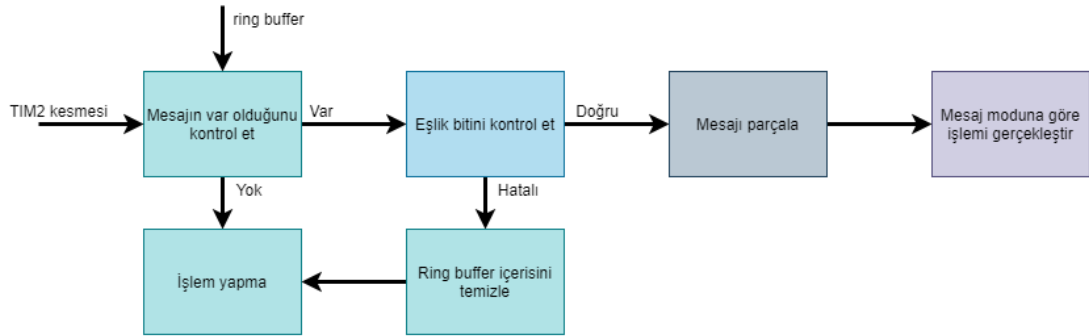
### 3.1.1.2. TIM2 Kesmesi

TIM2'nin ön zamanlayıcı değeri 1000, periyodu ise 160 olarak ayarlanmıştır.

$$(Çalışma\ ferkansı / ön\ ölçekleyici) / periyod$$

Formülünün sonucunda TIM2 10ms olarak ayarlanmıştır.

Kesmenin aktif edilmesiyle birlikte her 5ms'de ring buffer içerişini kontrol eder. Eğer ring buffer içerişinde mesaj var ise bir sonraki aşama olan eşlik biti kontrolüne alır. Bu aşamadan da onay aldıktan sonra mesajı parçalama fonksiyonlarına iletir. Parçalama fonksiyonu ise gelen mesajın içerişindeki mesaj moduna göre işlemini gerçekleştirir.



Şekil 3 - TIM2 kesmesi blok diyagramı

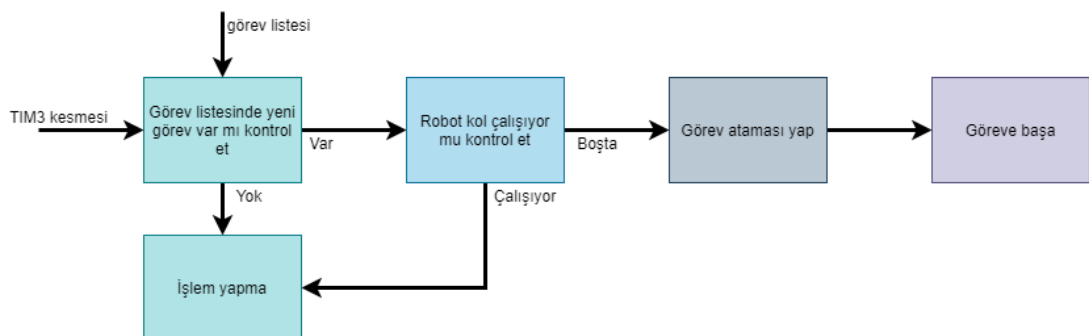
### 3.1.1.3. TIM3 Kesmesi

TIM2'nin ön zamanlayıcı değeri 1000, periyodu ise 8000 olarak ayarlanmıştır.

$$(Çalışma\ ferkansı / ön\ ölçekleyici) / periyod$$

Formülünün sonucunda TIM3 500ms olarak ayarlanmıştır.

Daha öncesinde ayarlanmış süre aralıklarıyla önce görev listesinde görevin olup olmadığını kontrol eder. Eğer görev varsa robot kolun çalışma durumunu kontrol eder. Robot kol boş durumda ise yeni görev ataması gerçekleştirir ve yönetici fonksiyona görevi iletir.



Şekil 4 - TIM3 Kesmesi blok diyagramı

### 3.1.2. Servo Motorlar İçin PWM Sinyali Oluşturma

Servo motorların kontrolü için PWM sinyali kullanılmaktadır. Motorlar, PWM sinyalinin logic 1 olduğu süreye göre açılarını değiştirmektedir. Bu PWM sinyalinin frekansı ise 50Hz olmalıdır. Bu sinyali üretmek için TIM8 kullanılmıştır. 4 motorun kontrolü için TIM8'in 4 kanalı kullanılmaktadır.

Sinyali oluşturulması için öncelikle TIM8'in tanımlamasında öncelikle 50Hz'lik bir zamanlama oluşturulması gerekmektedir. Bunun için;

$$(Çalışma\ frekansı / ön\ ölçekleyici) / periyod$$

Formülünden faydalanılmaktadır. STM32F4-Discovery kartının çalışma frekansı 16MHz olduğu bilindiğine göre uygun ön ölçekleyici ve periyod değeri belirlenmelidir.

Servo motorun kontrolünde daha hassas olması için PWM sinyalinin duty cycle değeri aralığı geniş olmalıdır. Bunun için ön ölçekleyici değeri düşük, periyod değeri yüksek seçilirse bu ister sağlanmış olur. Bu çalışmada ön ölçekleyici değeri 125 olarak belirlenmiştir. Formülden yola çıkarsak;

$$50Hz = (Çalışma\ frekansı / ön\ ölçekleyici) / periyod$$

$$50Hz = (16MHz / 125) / periyod$$

$$Periyod = 2560$$

Olarak bulunur. Duty cycle değeri ise 0 ile periyod değeri arasındadır. Bu değer sayesinde PWM sinyalinin logic 1 seviyesinde kalacağı süre belirlenmiş olur.

Servo motorun 0 – 180 derece arasında hareket için PWM sinyalinin 0,5ms(%2,5) ile 2,5ms(%12,5) arasında logic 1 seviyesinde olmalıdır. Bu durumda duty cycle değeri;

$$minimum\ duty\ cycle = 2560 * 0,025 = 64$$

$$maksimum\ duty\ cycle = 2560 * 0,125 = 320$$

Olarak belirlenir. Bu değer aralığında verilen değere göre servo motor konum almaktadır. (Teorik bilginin yanısıra çalışma esnasında bu değerler farklılık göstermiş olup, minimum duty cycle değeri 60, maksimum duty cycle değeri 305 olarak belirlenmiştir.)

Girilen bir açı değerine göre duty cycle değeri belirlemek için aşağıdaki formül kullanılmıştır. (Formüllerde duty cycle DY olarak kısaltılmıştır.)

$$DY = min.DY + \left( (maks.DY - min.DY) * \left( \frac{istenen\ açı}{180} \right) \right)$$

Örnek vermek gerekirse 0° derece için duty cycle değeri 60 ve 180° derece için duty cycle değeri 305 ise 90 derece için;

$$DY = 60 + \left( (305 - 60) * \left( \frac{90}{180} \right) \right)$$

$$DY = 60 + (245 * 0,5)$$

$$DY = 182,5$$

Olarak bulunmuş olur

### 3.1.3. Haberleşme Senaryosu

Bu alanda bilgisayardan STM32F4 Discovery geliştirme kartına bir mesaj geldiği zaman gerçekleşecek senaryo gösterilmiştir. Aşağıda gelebilecek mesajlar verilmiştir.

KAYIT MESAJI										
1. data set	{	Eslik	@	Mes.M.	^	dk	sn	sl	}	
byte	1	1	1	1	1	1	1	1	1	9
2. data set	{	Eslik	b.x	b.y	b.z	e.x	e.y	e.z	}	
byte	1	1	1	1	1	1	1	1	1	9
toplam										18

GOREV SIL MESAJI										
1. data set	{	Eslik	@	Mes.M.	^	dk	sn	sl	}	
byte	1	1	1	1	1	1	1	1	1	9
2. data set	{	Eslik	ind						}	
byte	1	1	1							4
toplam										13

GOREV LİSTESİ İSTEME MESAJI										
Mesaj	{	Eslik	@	Mes.M.	^	dk	sn	sl	}	
Toplam byte	1	1	1	1	1	1	1	1	1	9

ACILIS MESAJI										
Mesaj	{	Eslik	@	Mes.M.	^	dk	sn	sl	}	
Toplam byte	1	1	1	1	1	1	1	1	1	9

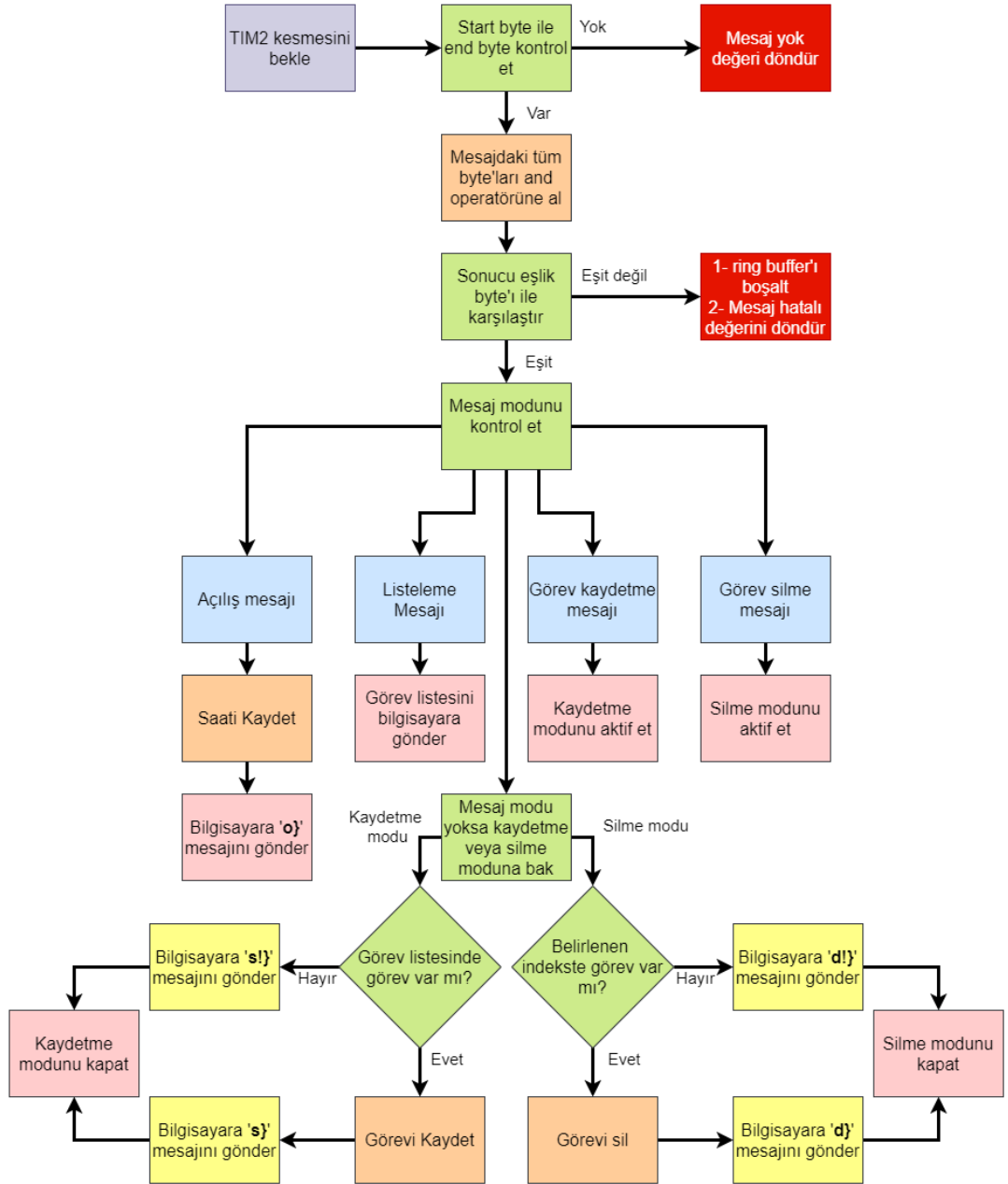
  

Kısaltmaların anlamları										
{	Başlangıç byte'ı									
Eslik	Mesaj içerisinde başlangıç ve bitiş byte'ı dışındaki verilerin hesaplanması ile oluşturulmuş veri									
@	Kendisinden sonra mesaj modunun geldiğini belirten işaret									
Mes. M.	Mesaj modunu belirten veri									
^	Kendisinden sonra zaman verisinin geleceğini belirten işaret									
dk	Gelen mesajın dakikası									
sn	Gelen mesajın saniyesi									
sl	Gelen mesajın salisesi									
b.x	Gelen görevin başlangıç konumunun X koordinatı									
b.y	Gelen görevin başlangıç konumunun Y koordinatı									
b.z	Gelen görevin başlangıç konumunun Z koordinatı									
e.x	Gelen görevin başlangıç konumunun X koordinatı									
e.y	Gelen görevin başlangıç konumunun Y koordinatı									
e.z	Gelen görevin başlangıç konumunun Z koordinatı									
ind	Silinecek görevin indeksi									

Şekil 5 - Bilgisayardan gelebilecek mesajların listesi

Bu tablodan yola çıkarak gelen mesajların işlenmesi alt başlıklarda açıklanmıştır. Sayısal veriler gönderirken gelen karakterin ASCII tablosundaki karşılık geldiği değer gönderilmiştir. Haberleşmede Sayısal veriler 0 – 100 arasında olacağı için başlangıç ve bitiş byte'ları bu aralık dışından seçilmiştir.

Gelen mesajlar öncesinde UART kesmesi ile ring buffer içerisine kaydedilir. Bir yandan da TIM2 kesmesi ring buffer içerisinde mesaj olup olmadığını kontrol eder. Eğer tamamlanmış (başlangıç ve bitiş byte'ları olan) bir mesaj bulursa ring buffer içerisinden mesajı çeker ve işlemeye başlar. Mesajın işlenmesi ve parçalanmasına dair süreçler aşağıdaki akış diyagramında verilmiştir.



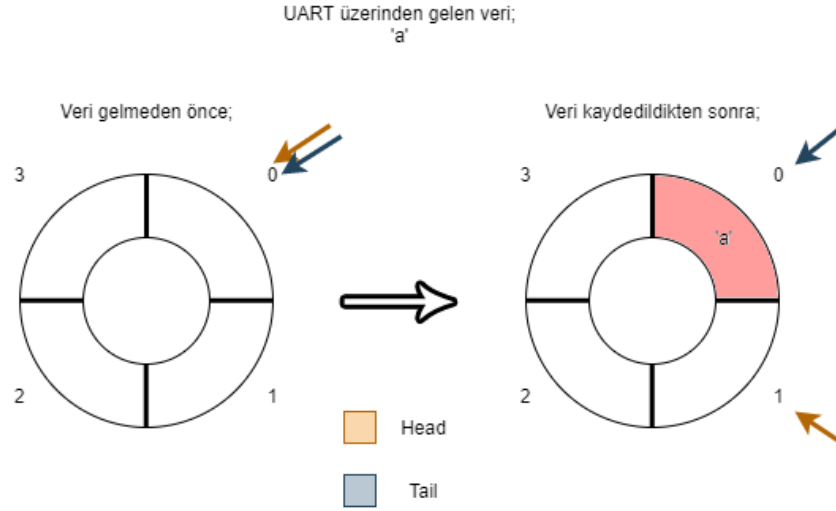
Şekil 6 - Mesaj parçalama ve işlemenin akış şeması

### 3.1.4. Ring Buffer Algoritması

Ring buffer algoritması, lineer bir yapının iki ucunun birleştirilmesi ile elde edilmiş bir halka olarak düşünülebilir. İçerisinde 2 adet işaretçi barındıran yapı, sadece bu halka içerisinde yazma ve okuma işlemi yapar. Bu sayede bellekte sadece belirlenen bir uzunluktaki alan sabit olarak kullanıldığı için verimli bir yöntemdir.

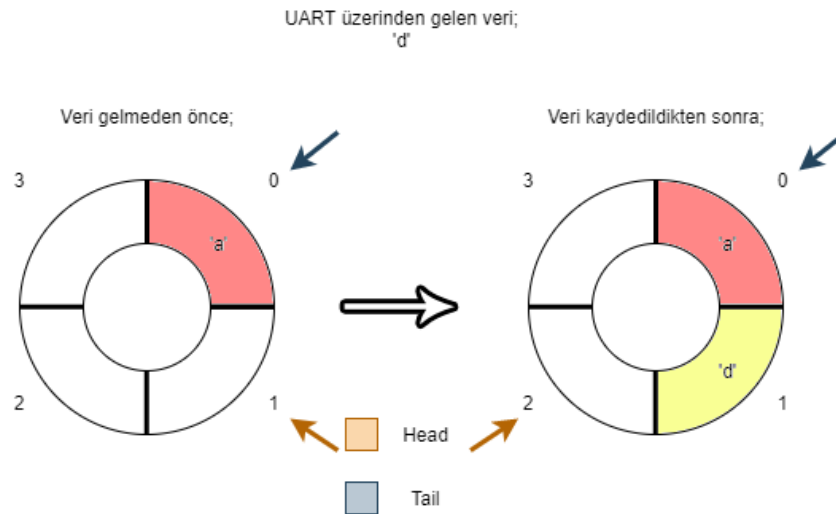
Dezavantaj, yapının boyutunun sabit olmasıdır. Bu sebeple kullanılacak ring buffer'ın uzunluğunu doğru belirlemek gerekir.

Çalışma mantığı, içerisinde barındırdığı head ve tail olmak üzere iki işaretçinin veri yazma ve okuma için hareket ettirilmesidir.



Şekil 7 - Ring buffer'a ilk veri geldiğindeki durum

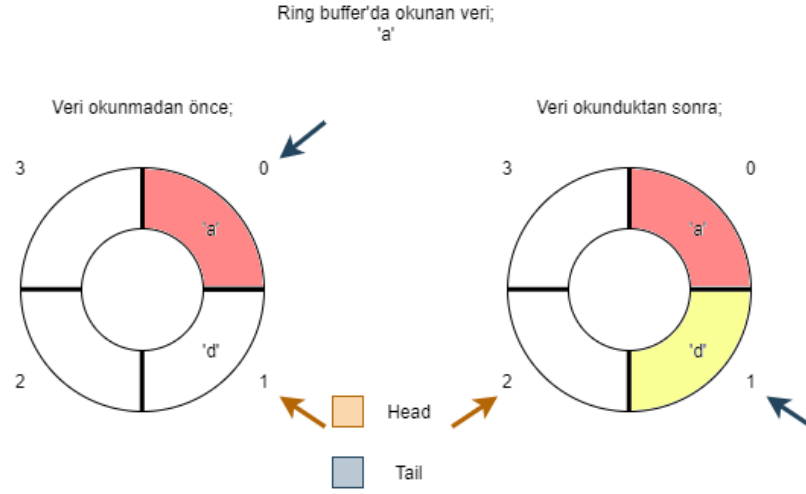
Şekilde belirtildiği gibi ilk çalışma anında head ve tail 0. indeksi göstermektedir. UART üzerinden 'a' verisi geldiği zaman ring buffer'ın 0. indeksine veri kaydedilir ve head bir artırılarak 1. indeksi gösterir. Bu durumda bir sonraki veri geldiğinde head'in gösterdiği 1. indekse veri yazılır.



Şekil 8 - Ring buffer'a 'd' verisi geldiğindeki durum



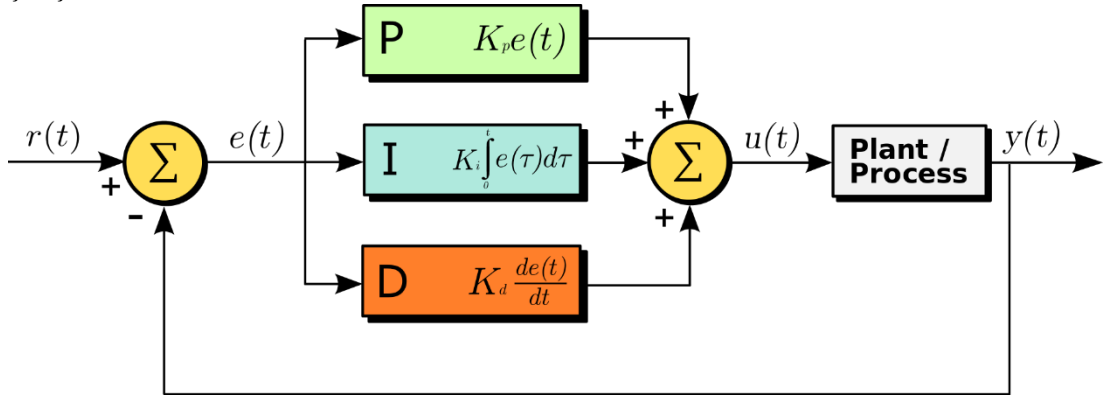
Ring buffer'da veri kaydetme yukarıda anlatıldığı gibi gerçekleşmektedir. Bu işlemin tersi olan veri boşaltma işlemi ise tail yardımı ile gerçekleşmektedir. Her veri okunduğu zaman tail'in gösterdiği adresteki değer alınır ve tail bir sonraki indekse kaydırılır.



Şekil 9 - Ring buffer'dan veri okunduğundaki durum

### 3.1.5. PID Kontrol Algoritması

PID (İngilizce: Proportional Integral Derivative) oransal-integral-türevsel denetleyici kontrol döngüsü yöntemi, endüstriyel kontrol sistemlerinde yaygın olarak kullanılan bir geri besleme denetleyicisi yöntemidir. Bir PID denetleyici sürekli olarak bir hata değerini, yani amaçlanan sistem durumu ile mevcut sistem durumu arasındaki farkı hesaplar. Denetleyici süreç kontrol girdisini ayarlayarak hatayı en aza indirmeye çalışır.[4]



Şekil 10 - PID blok şeması[4]

PID algoritması hata değerini azaltmak için üç ayrı sabit parametreyi kontrol eder, bu sebeple bazen *üç aşamalı kontrol* olarak adlandırılır: oransal,  $P$  ile gösterilir; integral  $I$  ile gösterilir; türev değerleri,  $D$  ile gösterilir. Sezgisel olarak, bu değerler mevcut değişim göz önüne alınarak zaman açısından şu şekilde yorumlanabilir;  $P$  mevcut hataya bağlıdır,  $I$  geçmiş hataların toplamı ve  $D$  gelecekteki hataların bir tahminidir. Bu üç eylemin ağırlıklı toplamı yoluyla kontrol edilen süreç istenilen seviyeye ayarlamak için kullanılır. Böylece örnek olarak bir kontrol valfinin pozisyonu

ya da bir ısıtma elemanı çıkış gücü kontrol edilerek istenilen akış veya sıcaklık seviyesi en düşük hata ile elde edilmeye çalışılır.[4]

Bu projede de servo motorların istenilen açıya gelmesi için PID kontrol algoritması kullanılmıştır.

Açı değerine göre PWM sinyalinin duty cycle değeri hesaplandıktan sonra o andaki PWM sinyalinin duty cycle değeri ile PID kontrol algoritmasına verilir.

İlk olarak P için istenen değeri ile anlık değer hesaplamaya girer. Çıkan sonuç hata olarak kabul edilir ve I için toplam hataya eklenir. D değeri için ise o anki hata değerinden bir önceki hata değeri çıkartılır.

Çıkan sonuçlar ile kendi katsayıları çarpılarak toplanır. Elde edile sonuç ise yeni duty cycle değeri olarak atanır. Son olarak en son hesaplan hata değeri ise önceki hata değeri olarak kaydedilir. Bir sonraki hesaplamada kullanılır.

Bu süreç istenen değere ulaşana kadar devam eder.

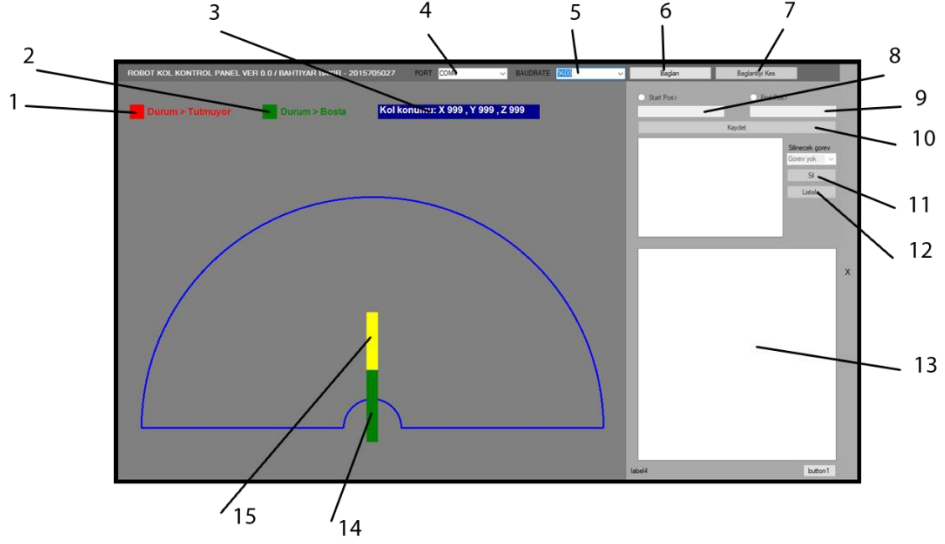
### **3.2. Kontrol Ara Yüzü Yazılımı**

Bu alandaki çalışmalar gömülü sistem yazılım ile paralel olarak yürütülmüştür. Kontrol ara yüzünün geliştirilmesindeki amaç, robot kolu uzaktan erişim sağlayabilmektir. Yeni görev ataması, görev listesinden görev silme, görevleri listeme ve anlık kol durumu takibi gibi işlemler bu ara yüz üzerinden yapılması hedeflemiştir.

Kontrol ara yüzü C# dili kullanılarak Microsoft Visual Studio 2017 ortamında Windows Form Application olarak geliştirilmiştir. .NET Freamwork içerisinde bulunan hazır yapılar sayesinde geliştirme süreci daha da hızlanmıştır.

Ara yüzde anlık kolun durumlarının takibi için özel kontrol (custom controller) tasarlanmıştır. Bu sayede alınan parametreler bu yapıya verilerek görselleştirilmiştir.

Projeye seri port sınıfı eklenerek bilgisayar ve STM32F4-Discovery geliştirme kartının haberleşmesi sağlanmıştır.



Şekil 11- Robot kol için tasarlanan kontrol ara yüzü

Şekil 11’de gösterilen ara yüz sayesinde istenilen işlemler yapılabilmektedir. Numaralandırılmış yapıların görevleri;

1. Robot kolun tutma/bırakma durumunun takibi,
2. Robot kolun çalışma/çalışmama durumunun takibi,
3. Robot kolun tutucu kısmının konumunun takibi,
4. Seri port bağlantısı için port seçme alanı,
5. Seri port için baud rate ayarı,
6. Seri porta bağlanmak için buton,
7. Seri port bağlantısının kapatılması için buton,
8. Yeni görev için başlangıç koordinatlarının girilmesi gereken alan,
9. Yeni görev için bitiş koordinatlarının girilmesi gereken alan,
10. Yeni görevi kaydetmek için buton,
11. Seçilen görev indeksinin silinmesi için buton,
12. Görev listesinin yenilemek için buton,
13. STM32F4-Discovery geliştirme kartı ile haberleşme esnasında log kayıtlarının takip alanı,
14. Robot kolun birinci parçasının temsili görseli,
15. Robot kolun ikinci parçasının temsili görseli

Şeklindedir.

## KAYNAKÇA

1. -. (-, - -). *Discovery kit with STM32F407VG MCU \* New order code STM32F407G-DISC1 (replaces STM32F4DISCOVERY)*. st.com: <https://www.st.com/en/evaluation-tools/stm32f4discovery.html> adresinden alındı
2. -. (2011, 4 12). *Servo motor*. wikipedia.org: [https://tr.wikipedia.org/wiki/Servo\\_motor](https://tr.wikipedia.org/wiki/Servo_motor) adresinden alındı
3. -. (-, - -). *MG90S Datasheet*. alldatasheet.com: <https://html.alldatasheet.com/html-pdf/1132104/ETC2/MG90S/109/1/MG90S.html> adresinden alındı
4. -. (2012, 3 4). *PID*. Wikipedia: <https://tr.wikipedia.org/wiki/PID> adresinden alındı