

**T.C. MALTEPE ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**GSM - GPRS TABANLI CİHAZ**  
**KONTROL YAZILIMI**

**YÜKSEK LİSANS TEZİ**

**Nazır AŞIKKUTLU**

**Tez Danışmanı**

**Yrd. Doç. Dr. Şenol Zafer ERDOĞAN**

**İSTANBUL - 2013**

Bu tez çalışması, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 29/08/2013 tarih ve 2013 /13 sayılı kararıyla oluşturulan jüri tarafından **Bilgisayar Mühendisliği Yüksek Lisans Tezi** olarak kabul edilmiştir.

JÜRİ

Yrd. Doç. Dr. Şenol Zafer ERDOĞAN

Danışman

Yrd. Doç. Dr. Turgay BİLGİN

Üye

Yrd. Doç. Dr. Fatih YÜCALAR

Üye

## ÖZET

Bu tez çalışmasında hücresele ağlar vasıtası ile uzaktan kontrol edilebilen mobil cihaz sistemi tasarlanmış ve gerçekleştirilmiştir. Gerçekleştirilen sistemde mobil cihazın hücresele ağlara bağlanabilmesi için Libellium firması tarafından geliştirilen GPRS/3G elektronik modülü kullanılmıştır. GPRS/3G modülünün seri bağlantı ara birimine bağlanmış olan mobil cihaz, internete bağlı herhangi bir bilgisayar üzerinde kurulu olan kumanda panelinden gönderilen komutlar ile yönetilebilmektedir.

Sistem insansız hava araçlarına uyarlanabilecek şekilde tasarlanmıştır. Bu amaçla insansız hava aracı olarak, Multikopter cihazı kullanılmıştır. Sistem, Multikopter cihazını hücresele ağlar üzerinde uzaktan kontrol edebilmeyi sağlamaktadır.

**Anahtar Kelimeler:** GSM, GPRS, 3G, Multikopter, Arduino, İHA

## **ABSTRACT**

In this study, a mobile machine system that can be controlled remotely over cellular network, is designed and implemented. A GPRS/3G electronic module, developed by Libellium, is used to connect the mobile machine system to the cellular network. The mobile machine system that is integrated with the GPRS/3G electronic module over its serial port, can be controlled by any command that is sent from a computer connected to internet and command panel installed.

The system is designed as customizable on unmanned aerial vehicles. So, a multicopter vehicle is used as an unmanned aerial vehicle. The system provides to control the multicopter remotely over cellular network.

**Anahtar Kelimeler:** GSM, GPRS, 3G, Multicopter, Arduino, UAV

## TEŐEKKÖR

Yüksek lisans çalıřmalarım sırasında, kıymetli tecrübe ve fikirlerini esirgemeyen tez danışmanım Sayın Yrd. Doç. Dr. Őenol Zafer Erdoğan'a, eğitim hayatım boyunca maddi ve manevi desteęini hiçbir zaman esirgemeyen kıymetli aileme, ayrıca bu çalıřma süresince bana olaęanüstü sabırla destek olan sevgili eřim Tülay Ařıkkutlu'ya teőekkürlerimi sunarım.

## İÇİNDEKİLER

1	GİRİŞ .....	1
2	UZAKTAN KONTROL SİSTEM UYGULAMALARI .....	2
2.1	TC65 GSM/GPRS Modülü Kullanılarak Uzaktan Kontrol Sistemi Gerçekleştirilmesi .....	2
2.1.1	Sistemin Gerçekleşmesi.....	2
2.2	İnternet Tabanlı Uzaktan Denetim için Bir Ara yüz Tasarımı .....	6
2.2.1	Sistemin Gerçekleşmesi.....	7
2.3	PLC ile kontrol edilen endüstriyel bir sistemin cep telefonu ile uzaktan kontrolü .....	13
2.3.1	Sistemin Gerçekleşmesi.....	14
3	GMS – GPRS/3G TABANLI CİHAZDAN CİHAZA KONTROL YAZILIMI UYGULAMASI.....	20
3.1	Uzaktan Kontrollü İnsansız Hava Aracı Uygulamaları .....	22
3.2	Sistem Mimarisi .....	24
3.2.1	Donanım.....	24
3.3	Uçuş Kontrol Yazılımı ve Uyarlaması.....	32
3.3.1	Uçuş Kontrol Yazılımının Uçuş Kontrol Kartına Yüklenmesi .....	35
3.3.2	Uçuş Kontrol Yazılımı Uyarlaması.....	41
3.4	GPRS/3G Modül ve Kullanımı .....	51
3.5	Bilgisayar Tabanlı Kumanda Panel Yazılımı.....	56
4	SONUÇ .....	65
5	KAYNAKÇA.....	66

## KISALTMALAR

<b>Kısaltma</b>	<b>İngilizcesi</b>	<b>Türkçesi</b>
GSM	Global System for Mobile Communications	Mobil İletişim İçin Küresel Sistem
GPRS	General Packet Radio Service	Genel Paket Radyo Servisi
SMS	Short Message Service	Kısa Mesaj Servisi
PC	Personal Computer	Kişisel Bilgisayar
GPIO	General Purpose Input Output	Genel Amaçlı Giriş Çıkış
PLC	Programmable Logic Controller	Programlanabilir Mantıksal Denetleyici
RF	Radio Frequency	Radyo Frekansı
DTMF	Dual-Tone Multi-Frequency	Çift-Tonlu Çoklu-Frekans
TCP	Transmission Control Protocol	İletim Kontrol Protokolü
IP	Internet Protocol	İnternet Protokolü
WAP	Wireless Application Protocol	Kablosuz Uygulama Protokolü
FPV	First Person View	İnsan Görüşü
GPS	Global Positionning System	Küresle Konumlama Sistemi
IDE	Integrated Development Environment	Tümleşik Geliştirme Ortamı
USB	Universal Serial Bus	Evrensel Seri Veriyolu

## ŞEKİLLER

Şekil 2-1 Gerçekleştirilen Sistemin Blok Diyagramı[1] .....	3
Şekil 2-2 Gerçekleştirilen Sistemin Blok Simülasyonu[1] .....	4
Şekil 2-3 PC ortamında sistemin çalışma ara yüzü[1] .....	6
Şekil 2-4 Uzaktan denetim ara yüz devresi blok diyagramı[2].....	7
Şekil 2-5 Micromaster 420 sürücü[11] .....	8
Şekil 2-6 Telegram yapısı[2].....	9
Şekil 2-7 Açık veri bloğu[2] .....	9
Şekil 2-8 Kullanıcı ara yüz programı[2] .....	11
Şekil 2-9 Ana akış bloğu[2] .....	12
Şekil 2-10 PLC kontrollü plastik enjeksiyon makinesinin telefon ile kontrolü[3] ....	15
Şekil 2-11 DTMF kod çözücü[3] .....	16
Şekil 2-12 Sistemin çalışmasına ait akış diyagramı[3] .....	18
Şekil 3-1 GPRS protokolü temelinde iletişimin sağlanması .....	21
Şekil 3-2 ANKA insansız hava aracı[12].....	22
Şekil 3-3 FPV Sistemi .....	23
Şekil 3-4 Uçuş kontrol kartı .....	25
Şekil 3-5 Arduino Mega 2650 programlama kartı[15] .....	26
Şekil 3-6 Aeroquad v2.1 donanım kartı .....	27
Şekil 3-7 Multikopter kasası .....	28
Şekil 3-8 A2217/9 Fırçasız motor .....	29
Şekil 3-9 APC 10*4.7 Pervane.....	30
Şekil 3-10 Pentium 30A ESC.....	31
Şekil 3-11 5000A Batarya.....	31
Şekil 3-12 Multikopter .....	32
Şekil 3-13 Arduino programlama ortamı .....	34
Şekil 3-14 Arduino programlama kartı için bağlantı portu seçimi .....	35
Şekil 3-15 Uçuş kontrol yazılımı yükleme ve yapılandırma (konfigürasyon) ara yüzü .....	37
Şekil 3-16 Uçuş kontrol kartının bilgisayara bağlantı portunun tespit edilmesi .....	38



Şekil 3-17 Arduino programlama ara yüz ortamının kurulu olduğu dizinin belirtilmesi .....	39
Şekil 3-18 Uçuş kontrol kartının bilgisayara üzerinde bağlı olduğu portun seçilmesi .....	39
Şekil 3-19 Uçuş kontrol yazılımının yüklenme testi ve yüklenmeye başlaması.....	40
Şekil 3-20 Uçuş kontrol yazılımı yüklemesinin tamamlanması .....	41
Şekil 3-21 “SETUP” prosedürü uyarlaması.....	44
Şekil 3-22 GPRS/3G modülün Arduino programlama kartı ile hazır hale getirilmesi .....	45
Şekil 3-23 GPRS/3G modülünü çalıştırmak için gerilim uygulanması .....	45
Şekil 3-24 “loop” prosedüründe gerçekleştirilen uyarlamaları.....	48
Şekil 3-25 “compareCommandActualValues” fonksiyonu .....	48
Şekil 3-26 “SyncActualandInstantValues” fonksiyonu .....	49
Şekil 3-27 “readRemoteValues” fonksiyonu .....	50
Şekil 3-28 GPRS/3G Modül .....	51
Şekil 3-29 GPRS/3G Modüle entegre edilebilen aygıtlar .....	52
Şekil 3-30 GPRS/3G Modüle üzerindeki pin dağılımı .....	53
Şekil 3-31 GPRS/3G Modüle ve uçuş kontrol kartı arasındaki donanımsal entegrasyon bağlantıları .....	54
Şekil 3-32 Kumanda paneli yazılımı kullanıcı ara yüzü .....	58
Şekil 3-33 Kumanda paneli artım fonksiyonu akış diyagramı.....	60
Şekil 3-34 Kumanda paneli yazılımı.....	64

## TABLULAR

Tablo 2-1 TC65 Terminale SMS ile gönderilen komutlar ve bilgi SMS'leri[1].....	5
Tablo 2-2 Denetim görev kümesi[2].....	10
Tablo 2-3 DTMF frekans çiftleri[3].....	17
Tablo 2-4 DTMF sinyallerine karşı üretilen ikili değerler[3].....	17
Tablo 2-5 PLC cihazında elde edilen komutlar[3].....	19
Tablo 3-2 "loop" prosedüründe çalışan görevler ve çalışma frekansları .....	43
Tablo 3-3 Yer istasyonundan uçuş kontrol kartına gönderilen komut dizisi formatı	47
Tablo 3-4 Gelişmiş AT komut kütüphanesi komut formatları kullanımı .....	55
Tablo 3-5 TCP/IP protokolünün kullanımı için gerekli AT komutları .....	56
Tablo 3-6 Kullanıcı arayüzünde gerçekleştirilebilen işlemler .....	59



# 1 GİRİŞ

Son 20 yılın en önemli özelliği olan mobilite gün geçtikçe insan hayatının çok daha fazla alanında rol almaktadır. 90'lı yıllarda ticari olarak kullanılmaya başlayan GSM (Global System for Mobile Communications – Mobil İletişim için Küresel Sistem) teknolojisi, 2000'li yıllarla birlikte gelişmesini ivmelendirerek sürdürmüştür.

İlk dönemlerinde sadece cep telefonları ile mobil olarak sesli görüşme yapılmasına olanak sağlayan GSM teknolojisi, artık görüntülü olarak görüşme sağlamakta ve kullanıcılarını bilgiye sürekli olarak bağlı tutabilmektedir.

GSM teknolojisinin gelişmesi ile beraber, cep telefonları dışında, bu teknolojiden faydalanan farklı ürünler de ortaya çıkmaktadır. Günümüzde yoğun olarak kullanılan tablet bilgisayarlar kullanıcılarını mobil ortamlarda internete bağlı tutarken, sanayi alanında da GSM teknolojisinden faydalanan makineler kullanılmaktadır.

Bu yüksek lisans çalışmasında, uzaktan yönetilebilen mobil cihazlar üzerinde GSM teknolojisi entegrasyonu incelenmektedir. Mobil cihaz olarak kullanılan dört motorlu bir Multikopter cihazın, normal kontrol için kullanılan alıcı-verici donanım dışında, GSM teknolojisi kullanılarak yönetilmesi amaçlanmaktadır.

Tezin ikinci bölümünde, uzaktan cihaz yönetimi konusunda gerçekleştirilmiş olan diğer çalışmalar ayrıntılı olarak incelenmiş olup, üçüncü bölümde ise gerçekleştirilen bu sistemin diğer sistemlerden farkı, tasarımı ve uyarlaması ayrıntılı olarak anlatılmaktadır.

Tezin son bölümünde ise çalışma sonunda ortaya çıkan yeni iletişim sisteminin sağladığı olanaklar anlatılmaktadır.

## **2 UZAKTAN KONTROL SİSTEM UYGULAMALARI**

Çalışmanın bu bölümünde, uzaktan kontrol sistemleri ile ilgili daha önceden yapılmış olan diğer çalışmalar incelenmektedir. Yapılan bu incelemeler ile, hem literatürde bu konuda ne gibi çalışmalar yapıldığı hakkında bilgi sahibi olunurken, hem de bu çalışmada gerçekleştirilmekte olan uzaktan kontrol sisteminin daha önceden geliştirilen bu sistemlere göre farklı yönleri daha iyi anlaşılmış olacaktır.

### **2.1 TC65 GSM/GPRS Modülü Kullanılarak Uzaktan Kontrol Sistemi Gerçekleştirilmesi**

Bu uygulamada, TC65 GSM/GPRS terminali kullanılarak uzaktan ortam kontrol sistemi tasarlanmış ve gerçekleştirilmiştir. Gerçekleştirilen sistemde, Siemens TC65 terminali kullanılmaktadır. TC65 terminalin çıkışlarına bağlanan röleler, herhangi bir mobil telefondan gönderilen SMS ile kontrol edilmektedir. Isı sensöründen alınan sıcaklık bilgisi mobil telefona SMS olarak iletilmektedir.

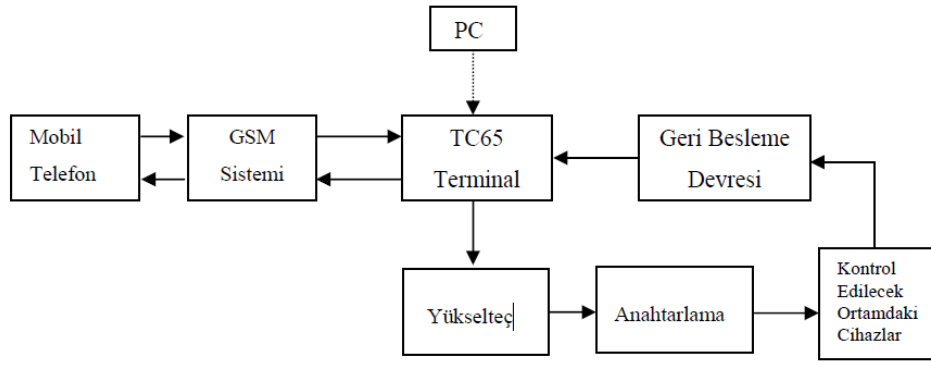
Sistem, akıllı evler, akıllı arabalar ya da endüstriyel otomasyon sistemlerine uyarlanabilecek şekilde tasarlanmıştır. TC65 terminal ve bir iletişim aracı olan mobil telefon kullanılarak gerçekleştirilen sistem; kablosuz olarak uzaktan ortam kontrolü sağlamaktadır [1].

#### **2.1.1 Sistemin Gerçekleşmesi**

Sistemin gerçekleşmesine ait blok diyagram Şekil – 2.1’de yer almaktadır. Sistem yedi ana bölüm olarak tasarlanmıştır. Bu bölümler;

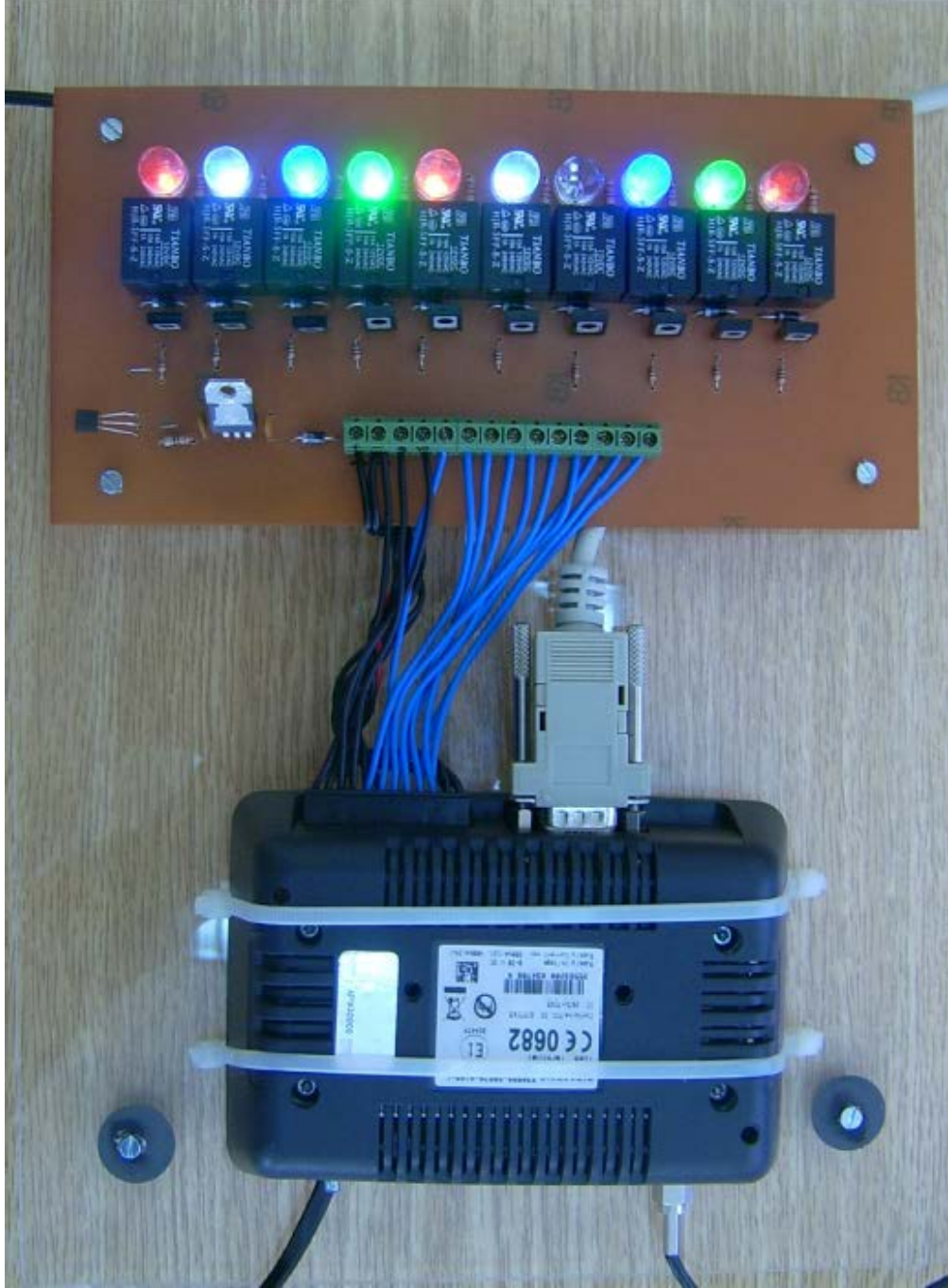
1. Mobil telefon,
2. GSM sistemi,

3. PC,
4. Siemens TC65 Terminal,
5. Geri besleme,
6. Yükselteç ve röle devresi,
7. Kontrol edilecek diğer cihazlar.



Şekil 2-1 Gerçekleştirilen Sistemin Blok Diyagramı[1]

Şekil – 2.1’deki blok diyagrama göre, herhangi bir mobil telefondan gönderilen SMS, GSM sistemi sayesinde TC65 Terminaldeki SIM karta iletilir. Borland Delphi 6 yazılım geliştirme ortamında yazılan program, istenilen aralıkla yeni mesaj kontrolü yapar. Sistem gelen yeni SMS’i kullanıcı tarafından program arayüzünde belirlenen periyodlarla okuyarak, TC65 Terminalin GPIO(General Purpose Input Output - Genel Amaçlı Giriş/Çıkış) çıkışlarını aktif yapar. Aktif olan GPIO pininden 2.93 voltluk gerilim alınır ve bu gerilim yükselttilerek 12 voltluk röle açılır ya da kapatılır. Röleye bağlı olan cihaz ise çalışır ya da durur. Açma ya da kapatma işlemi gerçekleştiği zaman SMS’in gönderildiği mobil telefona geri besleme mekanizması ile bilgi mesajı gönderilir. TC65’in 2 adet analog girişi vardır [1]. TC65, bu girişlerle analog sensörlerden veri okuma yeteneği kazanmaktadır. Sistemin gerçekleştirilmesine ait simülasyon Şekil-2.2’de görülmektedir. Buradaki led bileşenleri, kontrol edilmek istenen cihazı simgelemektedir.



Şekil 2-2 Gerçekleştirilen Sistemin Blok Simülasyonu[1]

Bir kombi cihazının kontrol edildiği ortamda suyun sıcaklığı ısı sensörünce algılanarak TC65 Terminalin analog girişlerine uygulanır ve bu bilgi mobil telefona SMS'le bildirilir. Bu durumda kullanıcı, suyun sıcaklığına bağlı olarak, SMS mesajı

içerisinde göndereceği bir komut sayesinde kombiyi pasif veya aktif konuma getirebilmektedir.

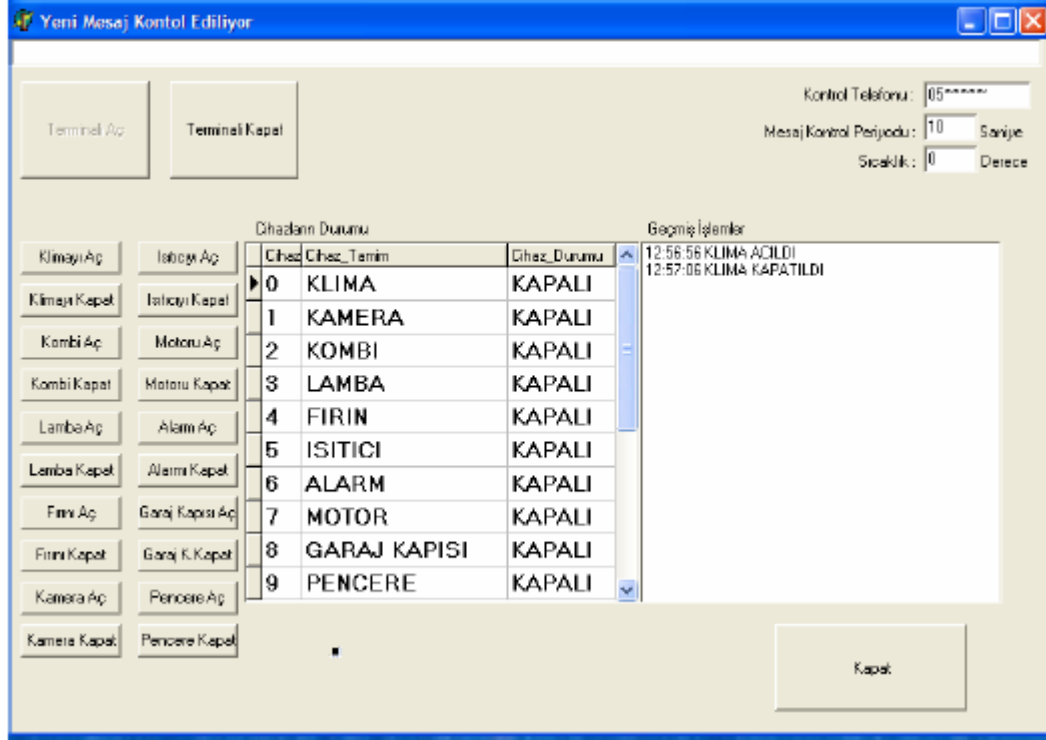
Sistemin kullanımını sırasında kontrol edilecek cihaza gönderilebilen komut kümesi Tablo-2.1’de yer almaktadır. Bu tabloya göre kullanıcı mobil telefonundan SMS mesajı içerisine “CK” komutunu yazılıp, sisteme gönderdiğinde, sistem, kontrol edilen kombi cihazını pasif konuma getirir ve işlemin sonucunu geri besleme mekanizması ile “KOMBİ KAPATILDI” ifadesini içeren bir SMS mesajı olarak kullanıcının mobil cihazına gönderir [1].

**Tablo 2-1 TC65 Terminale SMS ile gönderilen komutlar ve bilgi SMS’leri[1]**

TC65 TERMİNALE GÖNDERİLEN KOMUTLAR VE TC65'DEN GELEN BİLGİ SMS'LERİ			
No	Gönderilen SMS'ler	GPIO PIN durumu	Bilgi SMS'leri
1	KA	GPIO0 giriş	KLİMA ACILDI
2	KA (2.kez aynı SMS gelirse)	GPIO0 değişiklik yok	KLİMA ZATEN AÇIK
3	KK	GPIO0 çıkış	KLİMA KAPATILDI
4	KK (2.kez aynı SMS gelirse)	GPIO0 değişiklik yok	KLİMA ZATEN KAPALI
5	RA	GPIO1 giriş	KAMERA ACILDI
6	RA (2.kez aynı SMS gelirse)	GPIO1 değişiklik yok	KAMERA ZATEN AÇIK
7	RK	GPIO1 çıkış	KAMERA KAPATILDI
8	RK(2.kez aynı SMS gelirse)	GPIO1 değişiklik yok	KAMERA ZATEN KAPALI
9	CA	GPIO2 giriş	KOMBİ ACILDI
10	CA (2.kez aynı SMS gelirse)	GPIO2 değişiklik yok	KOMBİ ZATEN AÇIK
11	CK	GPIO2 çıkış	KOMBİ KAPATILDI
12	CK (2.kez aynı SMS gelirse)	GPIO2 değişiklik yok	KOMBİ ZATEN KAPALI
13	LA	GPIO3 giriş	LAMBA ACILDI
14	LA (2.kez aynı SMS gelirse)	GPIO3 değişiklik yok	LAMBA ZATEN AÇIK
15	LK	GPIO3 çıkış	LAMBA KAPATILDI
16	LK (2.kez aynı SMS gelirse)	GPIO3 değişiklik yok	LAMBA ZATEN KAPALI
17	FA	GPIO4 giriş	FİRİN ACILDI
18	FA (2.kez aynı SMS gelirse)	GPIO4 değişiklik yok	FİRİN ZATEN AÇIK
19	FK	GPIO4 çıkış	FİRİN KAPATILDI
20	FK (2.kez aynı SMS gelirse)	GPIO4 değişiklik yok	FİRİN ZATEN KAPALI
21	IA	GPIO5 giriş	ISITICI ACILDI
22	IA (2.kez aynı SMS gelirse)	GPIO5 değişiklik yok	ISITICI ZATEN AÇIK
23	IK	GPIO5 çıkış	ISITICI KAPATILDI
24	IK (2.kez aynı SMS gelirse)	GPIO5 değişiklik yok	ISITICI ZATEN KAPALI
25	AA	GPIO6 giriş	ALARM ACILDI
26	AA (2.kez aynı SMS gelirse)	GPIO6 değişiklik yok	ALARM ZATEN AÇIK
27	AK	GPIO6 çıkış	ALARM KAPATILDI
28	AK (2.kez aynı SMS gelirse)	GPIO6 değişiklik yok	ALARM ZATEN KAPALI
29	MA	GPIO7 giriş	MOTOR CALIŞIYOR
30	MA (2.kez aynı SMS gelirse)	GPIO7 değişiklik yok	MOTOR ZATEN CALIŞIYOR
31	MK	GPIO7 çıkış	MOTOR DÜRDÜRÜLDÜ
32	MK (2.kez aynı SMS gelirse)	GPIO7 değişiklik yok	MOTOR ZATEN DÜRDÜRÜLDÜ
33	GA	GPIO8 giriş	GARAJ KAPISI ACILDI
34	GA (2.kez aynı SMS gelirse)	GPIO8 değişiklik yok	GARAJ KAPISI ZATEN AÇIK
35	GK	GPIO8 çıkış	GARAJ KAPISI KAPATILDI
36	GK (2.kez aynı SMS gelirse)	GPIO8 değişiklik yok	GARAJ KAPISI ZATEN KAPALI
37	PA	GPIO9 giriş	PENCERE ACILDI
38	PA (2.kez aynı SMS gelirse)	GPIO9 değişiklik yok	PENCERE ZATEN AÇIK
39	PK	GPIO9 çıkış	PENCERE KAPATILDI
40	PK (2.kez aynı SMS gelirse)	GPIO9 değişiklik yok	PENCERE ZATEN KAPALI

Sistemin PC ortamındaki yönetimi ve kontrolü Şekil-2.3’de ara yüzü görülebilen bilgisayar uygulaması vasıtası ile sağlanmaktadır. Bu uygulama ile geri planda belirli periyodlarla SMS ve sıcaklık kontrolü yapılmasının yanında, sistem doğrudan uygulama ekranında bulunan tuşlar vasıtası ile yönetilebilmektedir.





Şekil 2-3 PC ortamında sistemin çalışma ara yüzü[1]

## 2.2 İnternet Tabanlı Uzaktan Denetim için Bir Ara yüz Tasarımı

Bu uygulamada, cihazların uzaktan denetlenebilmesi ve izlenebilmesi için internet tabanlı bir arayüz tasarımı gerçekleştirilmiştir. Sistemin kullanıcı arayüz yazılımı Java Applet teknolojisi kullanılarak tasarlanmıştır. Tasarlanan sistemde, arayüz yazılımı vasıtasıyla kullanıcı tarafından gönderilen istekler, mikro denetleyicili bir arayüz donanımı tarafından alınıp değerlendirildikten sonra cihaz yönlendirilmektedir [2].

Uzaktan denetim ve izleme işlevlerinin herhangi bir uzak bilgisayardan yapılabilmesi ve sunucu tarafta bilgisayar yerine tasarlanacak bir arayüz devresi bulunmasının daha uygun olduğu düşünülmektedir. Tasarlanacak arayüz devresinin duruma göre uyarlanabilmesi için flash program belleğine sahip bir mikrodenetleyici tercih edilmiştir. İnternete bağlı herhangi bir bilgisayar ya da mobil cihaz tarafından

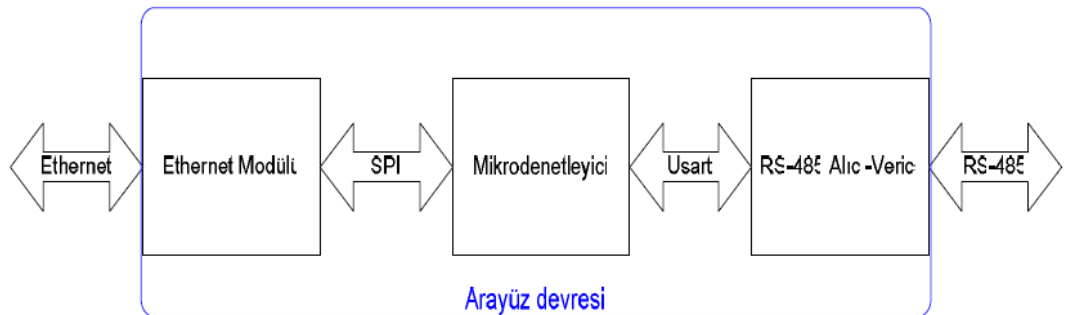
denetim ve izleme yapılabilmesi için özel hazırlanmış Java Applet'i bulunduran Web sayfasının kullanılması uygun görülmüştür [2].

### 2.2.1 Sistemin Gerçekleşmesi

Uzaktan denetim arayüzü, yazılım ve donanım olmak üzere iki ana bileşenden oluşmaktadır. Yazılım bileşeni uygulama ve arayüz yazılımlarından oluşurken, uygulama yazılımını donanım bileşenlerinden Ethernet modülü ve mikro denetleyici arasında iletişimi sağlayan bir köprü vazifesi görecektir şekilde tasarlanmıştır. Arayüz yazılımı ise donanım biriminin kontrol ve denetimini sağlamayı amaçlamıştır.

Şekil – 2.4'te sistem içindeki bir bilgisayar ağına bağlanmış, erişilebilir durumda olan bir devre donanım bileşeni görülmekte olup, uzaktan bir istemci tarafından yönetilmektedir. Bu şekilde görüldüğü gibi devre 3 ana donanımdan oluşmaktadır. Bunlar;

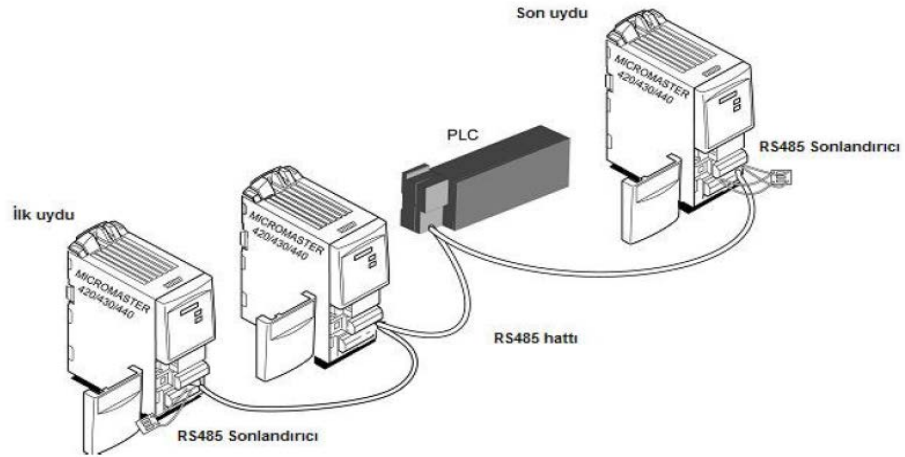
1. Ethernet modülü,
2. Mikro denetleyici,
3. RS-485 alıcı-verici modüldür.



Şekil 2-4 Uzaktan denetim ara yüz devresi blok diyagramı[2]

Bu çalışmanın konusunu oluşturan ve istemci bilgisayardan kontrol amaçlı komutları gönderebilmek amacı ile tasarlanan arayüz yazılımı Borland Jbuilder platformunda geliştirilmiştir [2].

Sistemde, istemci bilgisayardan gönderilen komutlar bağlı bulunan bilgisayar ağı üzerinden donanım arayüzüne gönderilir. Donanım arayüzündeki ethernet modülü vasıtası ile alınan komut verisi SPI arayüzü ile mikro denetleyiciye iletilir. Mikro denetleyici ethernet modülünden aldığı veriyi anlamlı hale getirerek RS-485 modülüne iletir. RS-485 modülünün alıcısına bağlanan bir cihaz ise uzaktan yönetilebilmektedir. Bu çalışmayı gerçekleştiren araştırmacı çalışmasında, bir asenkron motor kontrol etmektedir. Bu motorun kontrolü, sistemde kullanılan motor sürücüsü vasıtası ile, sürücüye gerekli parametrelerin iletilmesi ile gerçekleştirilmektedir. Kullanılan sürücü ise Siemens tarafından üretilen, Şekil – 2.5’de görülen Micromaster 420 model sürücülerdir [11].



Şekil 2-5 Micromaster 420 sürücüsü[11]

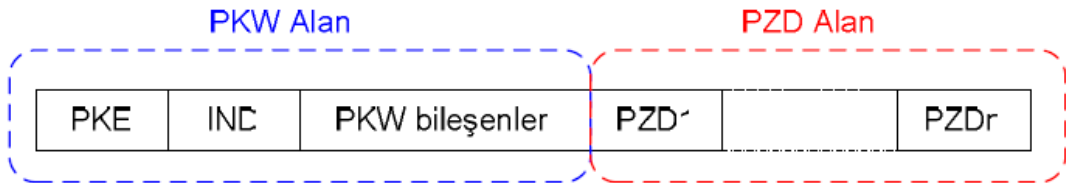
Geliştirilen uygulama ara yüzü, sistemin donanım bileşenine, motor sürücülerinde tanımlı olan parametreler için değerler göndermektedir. Bu gönderim, Şekil – 2.6’da görüldüğü formatta evrensel seri ara yüz protokolü vasıtası ile taşınan veri paketleri ile yapılmaktadır.

STX	LGE	ADF	Açık Veri Karakterler	BCC
-----	-----	-----	-----------------------	-----

Şekil 2-6 Telegram yapısı[2]

Her bir veri paketi, paketin başlangıcını belirten STX karakteri (02h) ile baslar, onu uzunluk belirten karakter (LGE) ve adres baytı (ADR) izler. Açık veri karakterleri bu 3 bayt'tan sonra gelir. Veri paketi blok denetim karakteri (BCC) tarafından sonlandırılır [2].

Açık veri karakterleri için ayrılmış olan alan içerisinde motor sürücüsüne gönderilecek komutlar ve alınan yanıtlar ile motor sürücüsünün okunacak ve değiştirilecek parametreleri tutulmaktadır. Şekil – 2.7’de görüldüğü üzere açık veri karakterleri alanı da kendi içerisinde PKW ve PZD olmak üzere iki alana ayrılmaktadır. PKW alanı parametre değerlerinin okunup yazıldığı alan olarak kullanılırken PZD alanı ise sürücünün denetlenmesi ve izlenmesi için kullanılmaktadır. Genel bir örnekle açıklamak gerekirse, bir veri paketinde kullanılan PZD alanının uzunluğuna PKW alanında ilgili parametrenin (P2012) değeri ile karar verilirken, motor sürücüsünün hangi hareketi yapacağına PZD alanının değeri ile karar verilmektedir.



Şekil 2-7 Açık veri bloğu[2]

Sürücünün denetlenmesi ve izlenmesi için PZD alanına gönderilebilen görev kümesi Tablo – 2.2’de görülmektedir.

Tablo 2-2 Denetim görev kümesi[2]

Bit0	AÇ/KAPAT1(rampalı)	0-Hayır, 1-Evet
Bit1	KAPAT2(elektriği kes)	0-Evet, 1-Hayır
Bit2	KAPAT3(Hızlı dur)	0-Evet, 1-Hayır
Bit3	Darbe yetkisi	0-Hayır, 1-Evet
Bit4	RFG yetkisi	0-Hayır, 1-Evet
Bit5	RFG başla	0-Hayır, 1-Evet
Bit6	Ayar noktası yetkisi	0-Hayır, 1-Evet
Bit7	Hata onayı	0-Hayır, 1-Evet
Bit8	Sağa dürt(JOG)	0-Hayır, 1-Evet
Bit9	Sola dürt	0-Hayır, 1-Evet
Bit10	PLC'den denetim	0-Hayır, 1-Evet
Bit11	Ayar noktasını ters çevir	0-Hayır, 1-Evet
Bit12	Atanmamış	
Bit13	Motorpoti yükselt	0-Hayır, 1-Evet
Bit14	Motorpoti düşür	0-Hayır, 1-Evet
Bit15	Yerel/uzak	0-P0719 ind 0 1-P0719 ind 1

Arayüz yazılımına ait ekran görüntüsü Şekil – 2.8’de gösterilmektedir. Arayüz yazılımını çalıştırmak için java applet çalıştırabilen bir web tarayıcı yeterlidir. Arayüz yazılımı için geliştirilen applet bir web sunucusunda saklanmaktadır ve kullanıcı bir internet tarayıcısı ile sunucudan applet uygulamasını talep ettiği zaman, applet tarayıcıya yüklenir ve kullanıcının bilgisayarında açılır. Kullanıcı sisteme kullanıcı adı ve şifre girmek suretiyle giriş yapar. Applet, donanım bileşeni erişilebilir durumdayken, donanıma otomatik olarak bağlantı kurmaktadır [2].

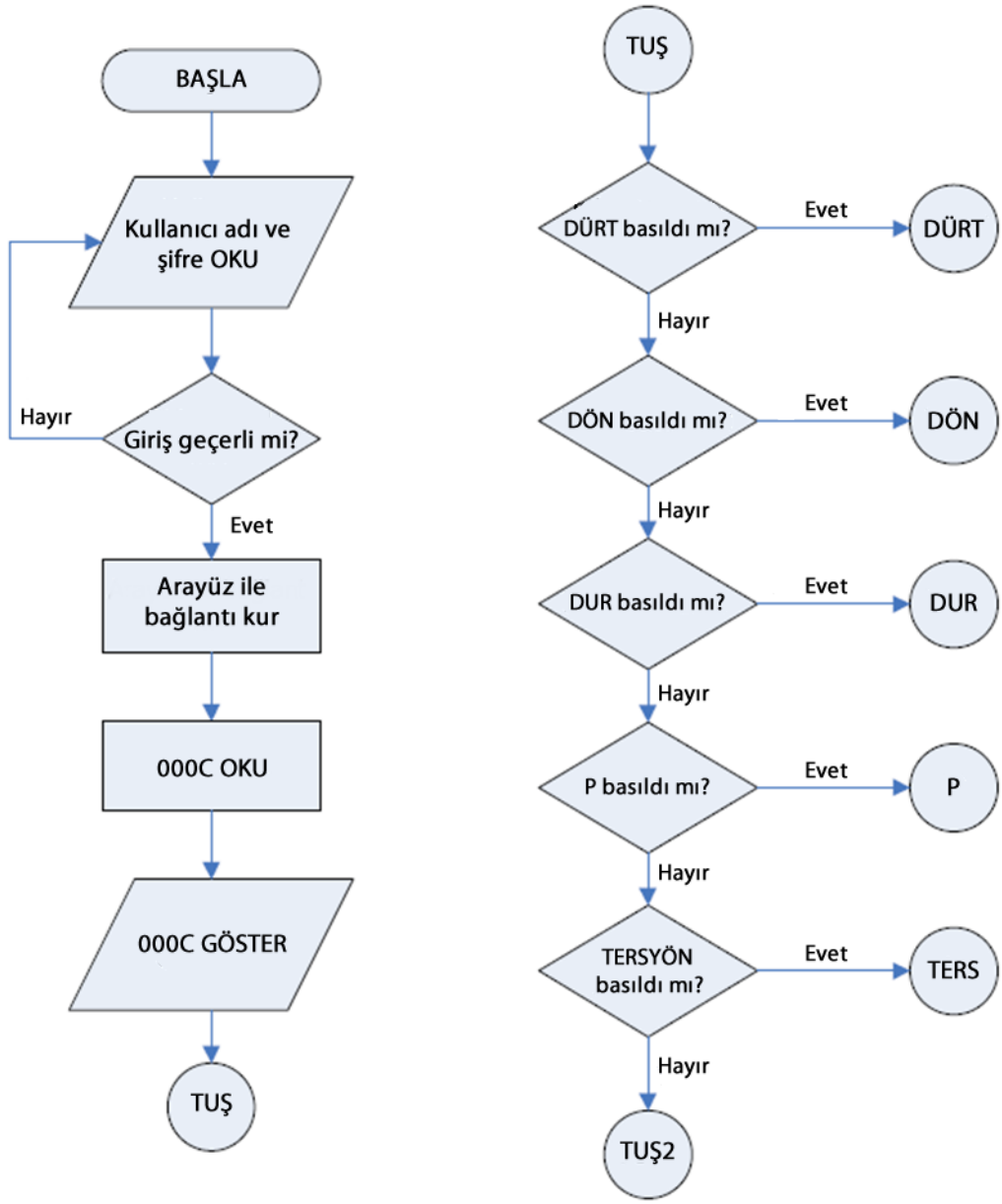
Donanımla bağlantı kurulduktan sonra, arayüz ekranında “r0000” parametresi gözükmemektedir ve kullanıcı arayüzde bulunan tuşlar vasıtası ile sistemin parametre değerlerini değiştirmekte ve sisteme komut göndermektedir [2].



Şekil 2-8 Kullanıcı ara yüz programı[2]

Kullanıcı arayüzünde yer alan “P” tuşu, parametre değerlerini görmek ve değişikliği kaydetmek için kullanılır. “Artır” ve “Azalt” tuşları istenilen parametre numarasına ulaşılmasını ve parametre değerinin ayarlanmasını sağlar. Parametreler arasında gezilirken r0000 parametresine dönmek için “Fn” tuşu kullanılabilir. “Dürt (Jog)” tuşu önceden belirlenmiş düşük frekans devrinde motoru döndürür. Bu tuş basılı kaldığı sürece motor döner, bırakıldığında ise durur. Eğer motor zaten çalışıyorsa bu tuş işe yaramaz. “TersYön” tuşu motorun dönüş yönünü değiştirmek için kullanılabilir. “Dön” tuşu, belirlenen frekansta motoru döndürür. “Dur” tuşuna bir kez basılarak Dur1, iki kez yada uzun basınca Dur2 moduna geçilir. Dur1 modu belirlenen oranda yavaşça motoru durdururken, Dur2 ise motorun gerilimini keserek durdurur [2].

Uygulamaya ait ana akış bloğu Şekil – 2.9’da gösterilmektedir.



Şekil 2-9 Ana akış bloğu[2]

### **2.3 PLC ile kontrol edilen endüstriyel bir sistemin cep telefonu ile uzaktan kontrolü**

Uygulama uzaktan kontrol sistemlerine cep telefonunun yanı sıra internet üzerinden de kontrol edilebilme olanağı tanımaktadır. Ayrıca cep telefonu ile PLC sistemlerini kontrol edebilme özelliği ve PLC cihazının veri aktarımını RF üzerinden gerçekleştirmesi fikri çalışmayı PLC cihazları üzerinde yapılan diğer çalışmalardan ayıran önemli faktörlerdir [3].

Sistemde, PLC cihazına kontrol ve denetleme amaçlı iki adet bağlantı kurulmaktadır. Bunlar komut gönderimi ve uzaktan kontrol amaçlı cep telefonu ve cihazın durumu hakkında bilgi edinmek amaçlı kullanılan internet bağlantısına sahip bir bilgisayardır. PLC cihazının bağlı olduğu bilgisayar sistemine uzak masaüstü programları ile bağlanmak kaydıyla, PLC cihazına internet üzerinden farklı bir bilgisayar ile de veri aktarımı sağlanabileceği için, kullanıcı tercih ederse bu yolla da PLC cihazına komut verebilecek ve programı düzenleyebilecektir. Burada PLC cihazının plastik enjeksiyon makinesine doğrudan bağlı olmayışı, sistemin internet üzerinden kontrol edilebilmesinde önemli bir maliyet avantajı sağlamaktadır. Bu avantaj RS232/PPI data kablosunun metrelerce uzatılarak bilgisayar ve PLC cihazı arasındaki fiziksel bağlantı maliyetini ortadan kaldırmaktadır. Bunun yerine PLC bağlı olduğu bilgisayara yakın mesafede durmakta ve PLC cihazının çıkış devresine bağlanan RF verici ve enjeksiyon makinesine bağlanan RF alıcı ile veri aktarımı uzaktan kablosuz olarak yapılmaktadır [3].

Ayrıca enjeksiyon makinesinin yakınına yerleştirilmiş bir IP (İnternet Protokol) kamera ile sistemin her an izlenilebilmesi mümkün hale gelmektedir. Kamera sayesinde cep telefonu veya ev telefonu ile gerçekleştirilen işlem, internete bağlanabilen bir cep telefonu ve bir bilgisayar yardımıyla da izlenebilmektedir. Böylece sistemin cevap verme durumu görsel olarak da takip edilebilmektedir [3].

Cep telefonları veya ev telefonlarından gelen DTMF (Dual Tone Multi – Frequency – İkili Ton Çoklu Frekans) sinyalleri bir kod çözücü entegresi ve mikroişlemci vasıtasıyla derlenerek, alınan bilgiler PLC cihazına data girişi olarak



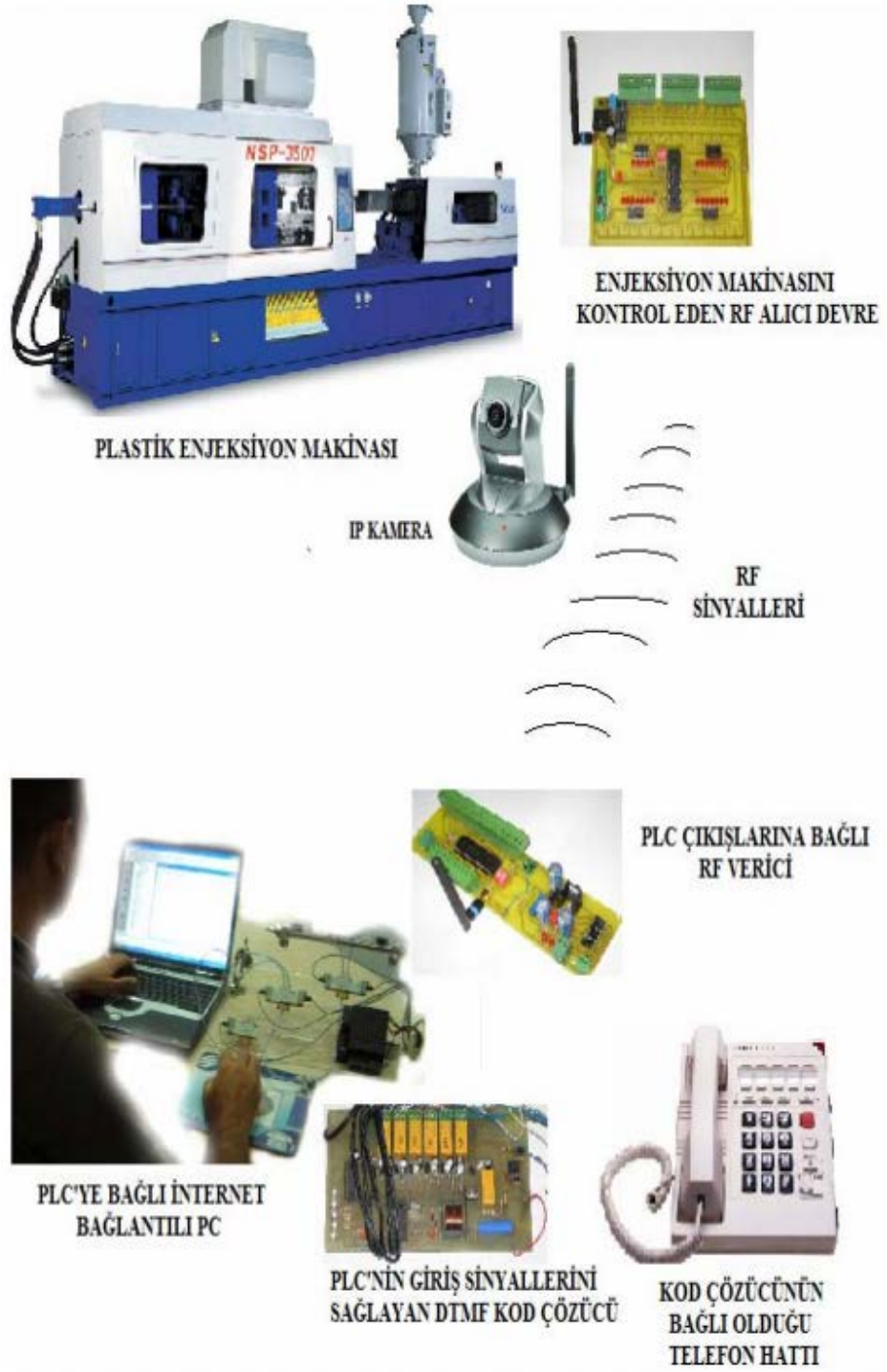
kullanılmaktadır. PLC cihazı alınan bu verileri değerlendirerek, RF temelli bir verici vasıtasıyla Plastik Enjeksiyon Makinesi'ne monte edilen bir alıcı üzerinden yine kablosuz bir şekilde kontrol akışı sağlamaktadır [3].

Sistemin en büyük avantajlarından bir tanesi de PLC cihazının bilgisayar bağlantısı kesilmeden PLC'ye doğrudan müdahale etme şansını getirmiş olmasıdır. Böylece kullanıcı uzakta bulunan makinesinin kullanımını istediği yerden yapabilmekte, bilgisayarıdan makinenin anlık durumlarını kontrol edebilmektedir. Ayrıca kullanıcı internet bağlantısı olan bir bilgisayar yardımıyla, istediği yerden işyerine gitmeksizin makinesini kontrol eden PLC cihazını görebilmektedir [3].

PLC cihazına bağlı bilgisayar ile kullanıcının yönetim için kullandığı bilgisayar arasında sanal özel ağ protokolü kullanılarak bağlantı sağlanmaktadır. Bu protokol üzerinde çalışmakta olan uzak masaüstü bağlantı programlarından "Teamviewer" programı bağlantı kurmak için tercih edilmiştir. Bu program, bağlantı kurulan bilgisayarın uzak bir noktadan, doğrudan kullanılabilmesine olanak sağlamaktadır. Dolayısı ile bu program sayesinde kullanıcı uzaktaki bir bilgisayara müdahale edebilir veya ekran görüntüsüne ulaşabilir aynı zamanda bilgisayarın ekranından PLC yazılımının simülasyonunu görebilir. Böylece ani bir arıza durumundan hemen haberdar olabilen kullanıcı, olası tehlikelere karşı kendini güvende hissedebilmektedir [3].

### **2.3.1 Sistemin Gerçekleşmesi**

Şekil – 2.10'da sistemin kurulmuş hali ve çalışma özeti gösterilmektedir.



Şekil 2-10 PLC kontrollü plastik enjeksiyon makinesinin telefon ile kontrolü[3]

Burada kod çözücünün bağlı olduğu telefon hattı ile bir cep telefonu veya sabit telefon vasıtası ile bağlantı kurulmaktadır. Şekil – 2.11’de kod çözücü görülebilmektedir.



Şekil 2-11 DTMF kod çözücü[3]

Bağlantı, hedef telefona bağlanan bir mikroişlemcinin bu telefonun belirli sayıda çalmasını sayıp telefonu yanıtlaması ile başlatılmaktadır. Kurulan bu bağlantı ortamında, iletişimi başlatan telefondan her tuşa basıldığında, hedef telefona bir ses tonu ulaşmaktadır. Bu ses tonlarına DTMF sinyalleri denmektedir. DTMF kodlama sisteminde temel olarak dört adet iki çift ton kullanılır. Bu iki ton kombinasyonu ile 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, #, \*, A, B, C, D rakam ve sembolleri ifade edilir [3]. Tablo – 2.3’de tuş takımı frekanslarına ait çizelge gösterilmektedir.

Tablo 2-3 DTMF frekans çiftleri[3]

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

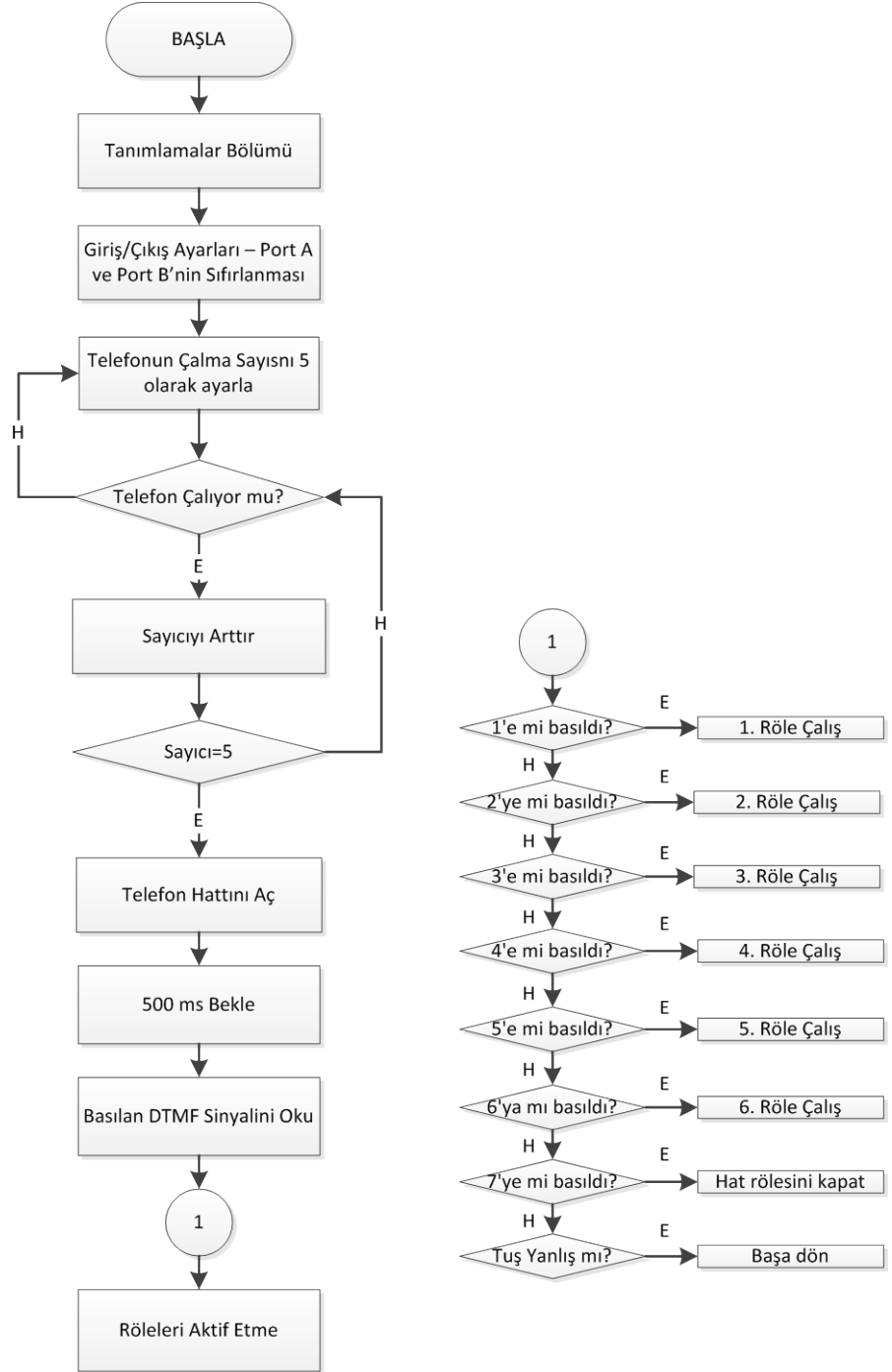
Hedef telefona ulaşan DTMF sinyalleri bir kod çözücü ile 4 bitlik binary değere dönüştürülmektedir [3]. Bu değerler Tablo – 2.4’te gösterilmektedir.

Tablo 2-4 DTMF sinyallerine karşı üretilen ikili değerler[3]

Tuş	f alt	f üst	Q4	Q3	Q2	Q1
1	697	1209	0	0	0	1
2	697	1336	0	0	1	0
3	697	1477	0	0	1	1
4	770	1209	0	1	0	0
5	770	1336	0	1	0	1
6	770	1477	0	1	1	0
7	852	1209	0	1	1	1
8	852	1336	1	0	0	0
9	852	1477	1	0	0	1
0	941	1336	1	0	1	0
*	941	1209	1	0	1	1
#	941	1477	1	1	0	0
A	697	1633	1	1	0	1
B	770	1633	1	1	1	0
C	852	1633	1	1	1	1
D	941	1633	0	0	0	0

Bu veri mikro işlemcinin girişlerine bağlıdır ve mikroişlemci aldığı bu datayı değerlendirerek çıkış portlarındaki rölelere gerekli çıkışı gönderir. Mikro işlemcinin çıkış bacaklarına bağlanan PLC cihazı mikro işlemcinin çıkışından gelen

sinyalleri giriş parametresi olarak alır ve değerlendirir. PLC cihazının girişlerine gelen bu verinin elde edilmesinde izlenen algoritma Şekil – 2.12’de gösterilmektedir.



Şekil 2-12 Sistemin çalışmasına ait akış diyagramı[3]

PLC cihazının giriş devresine gelen veri, PLC cihazının içerisinde yüklü program tarafından yorumlanarak plastik enjeksiyon makinesinin çalıştıracağı komut elde edilmekte ve PLC cihazının çıkış devresine bağlı olan RF verici cihazının girişine göndermektedir. RF verici cihazı ise aldığı bu veriyi, plastik enjeksiyon makinesine bağlı olan RF alıcı cihazına kablosuz olarak göndermektedir. PLC cihazı tarafından elde edilen komutlar Tablo – 2.5’te gösterilmektedir.

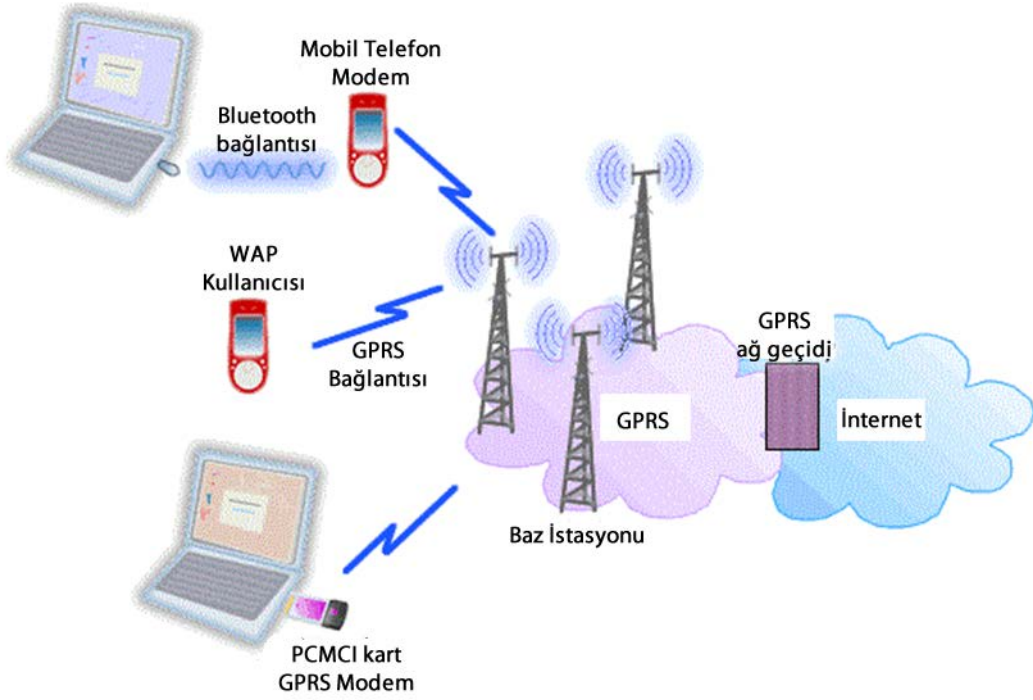
**Tablo 2-5 PLC cihazında elde edilen komutlar[3]**

TUŞ NUMARASI	RF ALICI KARTINDAKİ NUMARASI	GÖREVİ	PLC ÇIKIŞI
1	0	ANA KONTAKTÖRÜ AÇ-KAPA	Q0.1
2	1	MENGENE AÇ-KAPA	Q0.2
3	4	GRUP İLERİ	Q0.5
4	5	GRUP GERİ	Q0.6
5	RESET		
6			
7	6,7 VE 8	100 DERECE	Q0.7(SET4)
8	6,7 VE 8	150 DERECE	Q0.7(SET4)
9	6,7 VE 8	200 DERECE	Q0.7(SET4)

### **3 GMS – GPRS/3G TABANLI CİHAZDAN CİHAZA KONTROL YAZILIMI UYGULAMASI**

GSM (Global System for Mobile Communications), dijital hücreli ağlarda mobil cihazlarla iletişim kurulabilmesini sağlayan, 2. nesil protokol kümesidir. Günlük hayatta 2G olarak da kullanılmaktadır.

Kullanıcılarının hayatına 1990'lı yıllarının ortalarında giren mobil telefon cihazlarının birbirleri ile kurdukları iletişimi yöneten GSM protokolü, cihazdan cihaza ses ve kısa mesaj taşıyarak kullanıcılarına servis vermekteyken, 2000 yılı ile birlikte mobil telefon kullanıcıları, daha geniş bir ağ üzerinde hızlı veri taşıma kapasitesine sahip olan GPRS protokolü ile tanışmıştır. GPRS, verilerin mevcut GSM şebekeleri üzerinden saniyede 28.8 kb'den 115 kb'e kadar varabilen hızlarda iletilebilmesine imkan veren, cep telefonu, dizüstü bilgisayar, PDA (Cep Bilgisayarı) ve diğer mobil cihaz kullanıcılarına kesintisiz internet bağlantısı sunan bir mobil iletişim servsidir [12]. Bu servis de aynı zamanda 2.5G olarak adlandırılmaktadır. Bu protokol data taşıyabilmek için TCP/IP protokolünü kullanmaktaydı ve kullanıcıların günlük iletişim ihtiyaçları için çok önemli bir dönüm noktasını oluşturmaktaydı. Kullanıcılar sabit noktalarda bulunan bilgisayar sistemleri vasıtası ile internet erişimi sağlarken, GPRS teknolojisi bu hizmeti mobil ortama taşımıştır. GPRS ile birlikte sadece kullanıcılar arası bir iletişim değil, aynı zamanda farklı türden cihazlar arası iletişimin de kurulabilmiş ve bu alanda farklı uygulamalar geliştirilmeye başlanmıştır. GPRS protokolü ile iletişim Şekil – 3.1'de görsel olarak yer almaktadır.



Şekil 3-1 GPRS protokolü temelinde iletişimin sağlanması

Bu çalışmada GPRS tabanlı olarak, uzaktaki bir mobil cihazın kontrolü sağlanabilmektedir. Uzak cihaz olarak 4 pervaneli bir helikopter seçilmiştir. Dolayısı ile çalışma sonunda bir insansız hava aracının uzaktan kontrolü ile ilgili bir tasarım ortaya çıkacaktır. Ortaya çıkacak bu sistemi diğer çalışmalardan ayıran en önemli özellik, yer istasyonu ve uzak cihaz arasında kurulan iletişimin hücresel ağlar üzerinde, GPRS (General Packet Radio Servis – Genel Paket Radyo Servisi) protokolü kullanılarak gerçekleştirilmiş olmasıdır. Daha önce incelenen sistem uygulamalarına bakıldığında sistemlerin veri aktarımı kısa mesaj servisi veya DTFM sistemi ile yapılmaktaydı. Dolayısı ile iletişim, kısa mesaj servisi ile oldukça gecikmeli veya DTFM sistemi ile çok kısıtlı komut kümeleri ile yapılmaktadır. GPRS protokolü ise uzak cihaza gönderilen komut akışındaki gecikmeyi minimuma indirilebilmenin yanı sıra, çok daha geniş komut kümeleri ile çalışabilmeyi, çift yönlü olarak sağlamaktadır.



### 3.1 Uzaktan Kontrollü İnsansız Hava Aracı Uygulamaları

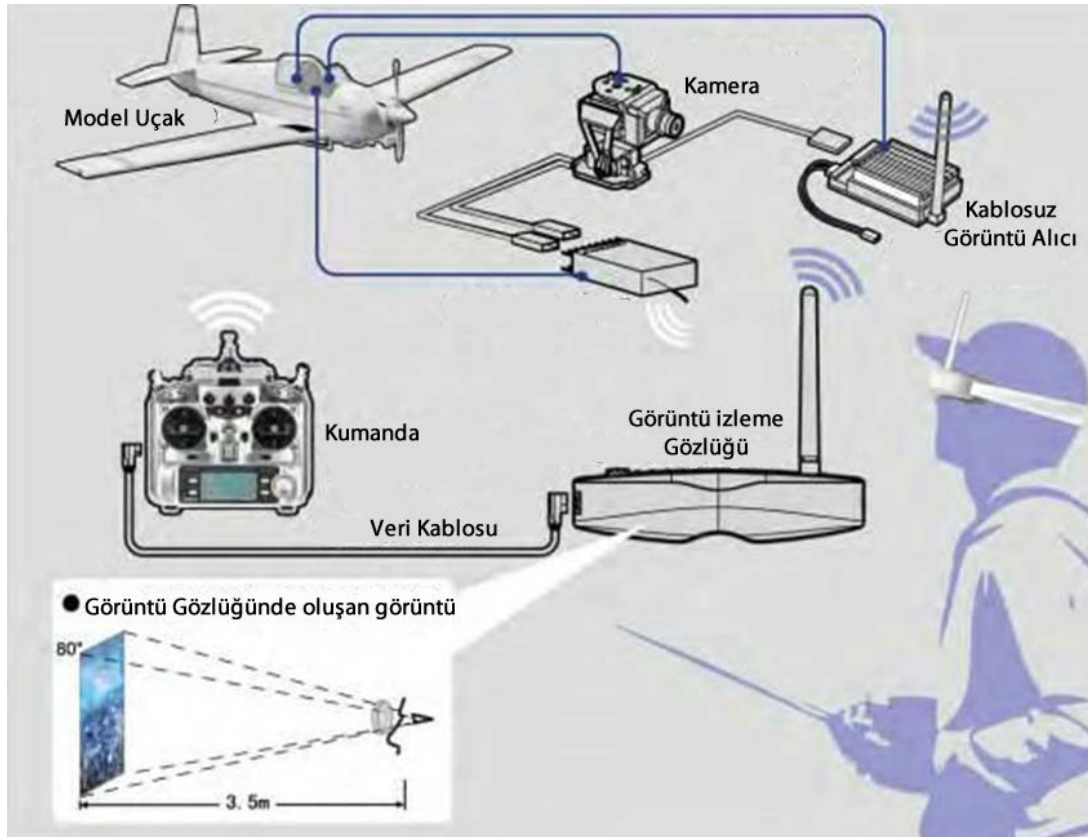
Günümüzde insansız hava araçları, uzaktan kumanda edilebilen hava araçları olarak birçok alanda, farklı amaçlar için kullanılmaktadır.

Günümüzde teknolojinin gelişimine öncülük eden askeri amaçlı projeler, insansız hava araçlarının gelişimi ve kullanımı için de yüksek önem arz etmektedir. Ülkemizde TAI tarafından geliştirilmekte olan ANKA projesi bu projelerden birisidir. ANKA insansız hava aracı, maksimum 30.000 metre yükseklikte, kesintisiz hizmet verebilmektedir. Gece ve gündüz, kötü hava şartları da dahil, keşif, gözetleme, sabit/hareketli hedef tespit, teşhis, tanımlama ve takip amaçlı, gerçek zamanlı görüntü istihbaratı görevlerini yerine getirmeyi amaçlamaktadır. Yer ile cihaz arası iletişim için uydu haberleşme teknolojileri kullanılmaktadır [12]. Şekil – 3.2’de ANKA insansız hava aracı görülmektedir.



Şekil 3-2 ANKA insansız hava aracı[12]

İnsansız hava araçlarının, günümüzde kullanıldığı en önemli alanlardan biri de film sektörüdür. Özellikle çekimi zor ve yüksek maliyet gerektiren alanlarda, insansız hava araçları işi çok önemli ölçüde kolaylaştırmanın yanı sıra, maliyet açısından da önemli avantajlar sağlamaktadır. Bu sistemlerde, hava aracına yerleştirilen bir kamera ve kamera kontrol ünitesi yerleştirilmektedir. Kamera kontrol ünitesi hava aracının alıcına bağlanmakta ve yerdeki kumanda panelinden gönderilen sinyallerle kamerayı hareket ettirmektedir. Kameradan elde edilen görüntü verisi ise, uzak noktalar arası veri gönderimi sağlayan telemetri sistemi alıcı vericileri ile hava aracından yer istasyonundaki panele aktarılmaktadır. Bazı sistemlerde ise görüntü aktarımı olmayıp, kameranın elde ettiği görüntü doğrudan hava aracı üzerindeki hafıza ünitesinde depolanabilmektedir. Bu amaçla kullanılan sistemlere FPV (First Person View – İlk İnsan Görüşü) adı verilmektedir. Şekil – 3.3’de bir FPV sistemine ait genel sistem tasarımı görülebilmektedir.



Şekil 3-3 FPV Sistemi

## 3.2 Sistem Mimarisi

Bu çalışmada gerçekleştirilen sistemin en önemli özelliği, uzak cihaz ve yer istasyonu arasındaki iletişimin dijital hücreli ağ alt yapısı üzerinde GPRS/3G protokolünü kullanmasıdır. Bu şekilde, geleneksel RF kumanda cihazlarına göre çok daha uzun menzile elde edilebilmekte ve iletişim ağı üzerinde çift yönlü olarak daha büyük boyutlarda veri taşınabilmektedir. Bununla beraber, RF kumandalarına göre uzak cihaz ve yer istasyonu arasındaki iletişim hızının daha yavaş olduğu gözlemlenmektedir.

Uzak cihaz olarak seçilmiş olan 4 pervaneli helikopter literatürde incelendiği zaman, iki adetten daha fazla pervanesi olduğu için helikopter değil, multikopter sınıfına girmekte olduğu anlaşılmaktadır. Bunun için çalışmanın bundan sonraki bölümünde “dört pervaneli helikopter” ifadesi yerine sadece “multikopter” ifadesi kullanılacaktır.

Çalışma sırasında kullanılan multikopter sistemi, bir açık kaynaklı multikopter projesi olan Aeroquad projesi baz alınarak geliştirilmiştir [14]. Projenin açık kaynaklı olması, multikopter sistemi üzerinde en ince detayına inceleyebilmeye imkan vermektedir.

Çalışma sırasında kullanılan multikopter başlıca iki ana kısımdan oluşmaktadır. Bunlar aşağıdaki kısımlardır.

1. Donanım
2. Yazılım

### 3.2.1 Donanım

Multikopterin donanımı da aşağıdaki üç ana bölüm olarak ayrılmaktadır. Bunlar;

- Uçuş kontrol Kartı
- Kasa
- Yardımcı Bileşenler

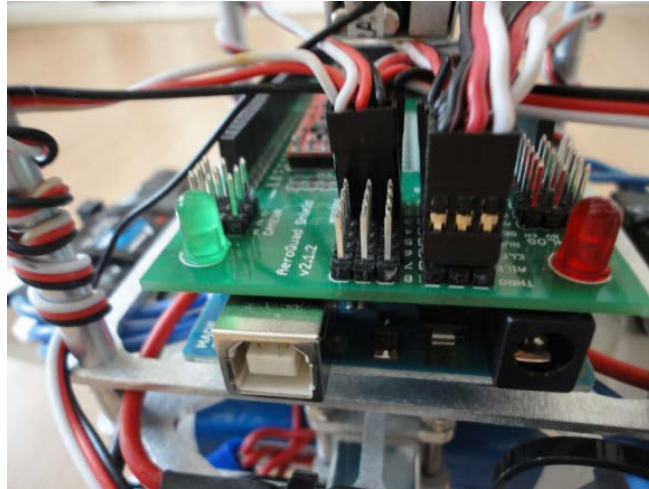
### 3.2.1.1 Uçuş Kontrol Kartı

Uçuş kontrol kartı sistemin beynini oluşturmaktadır. Kullanıcıdan aldığı komutları ve üzerinde taşıdığı sensörlerden aldığı verileri uçuş kontrol yazılımı vasıtası ile değerlendirerek gerekli komutları üretmekte ve bu komutları motorlara göndermektedir.

Uçuş kontrol kartı iki ana bölümden oluşmaktadır. Bunlar;

1. Arduino Mega 2650 programlama kartı
2. Aeroquad v2.1 donanım katmanı

Aeroquad v2.1 donanım kartı, Arduino Mega 2650 programlama kartının üzerine monte edilerek oluşturulan sistem bütün olarak uçuş kontrol kartını oluşturmaktadır [14]. Uçuş kontrol kartı Şekil – 3.4’de görülebilmektedir.



Şekil 3-4 Uçuş kontrol kartı

Arduino Mega 2650, Arduino programlama kartı ailesinin bir üyesidir. Bu kart ile sadece multikopter değil, birçok farklı amaç için uygulama geliştirilebilmektedir. Kartın programlaması hakkında ileriki bölümlerde daha ayrıntılı bilgi verilecektir. Şekil – 3.5’de Arduino Mega 2650 programlama kartı görülebilmektedir.



Şekil 3-5 Arduino Mega 2650 programlama kartı[15]

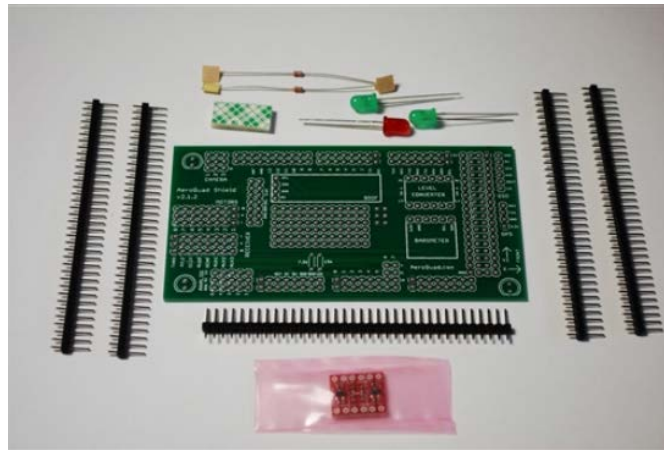
Aeroquad v2.1 donanım kartı ise, Arduino programlama kartı ile sistemdeki sensörler, elektronik hız kontrolcüler ve alıcı arasındaki iletişimi sağlayan bir devre kartıdır. Sistemin çalışma anında, tüm bileşenler Aeroquad v2.1 donanım kartı üzerine monte edilmiş durumda olmaktadır. Bu bileşenler aşağıda görevlerine dair açıklamaları ile listelenmektedir.

1. **Alıcı:** Uzak noktadan gönderilen radyo frekans dalgalarını okuyarak dijital sayı değerlerine döndürmekte ve Arduino programlama kartına iletmektedir.
2. **Barometrik Basınç Sensörü:** Bulunduğu ortamın atmosferik basıncını ölçmektedir. Atmosferik basınç yükseklik arttıkça azalmakta

olduğundan, bu sensör kullanılarak sensörün bulunduğu noktanın yüksekliği elde edilmektedir.

3. **Akselometre:** Üzerine düşen statik veya dinamik ivmeyi ölçmektedir. Bu sayede multikopterin sağa, sola, ileri ya da geri yatması durumunda, ne kadarlık bir hız ile bu hareketi yaptığını uçuş kontrol kartına sağlamaktadır.
4. **Magnetometre:** Manyetik alan şiddetini ve yönünü ölçmektedir. Multikopterin yön bilgisi bu sensör vasıtası ile elde edilmektedir.
5. **Gyro:** Bulduğu ortamın eğimini hesaplamaktadır. Multikopterin de anlık eğim değerini elde edip, uçuş kontrol kartına bu veriyi sağlamaktadır.
6. **Dört adet elektronik hız kontrolcüsü :** Elektronik hız kontrolcüleri, uçuş kontrol kartında hesaplanan motorların dönüş hızlarını girdi olarak alarak, bataryadan aldığı elektrik akımını bağlı olduğu motora iletmektedir.

Aeroquad v2.1 donanım kartının monte edilmemiş durumu aşağıda bulunan Şekil – 3.6’da gösterilmektedir.

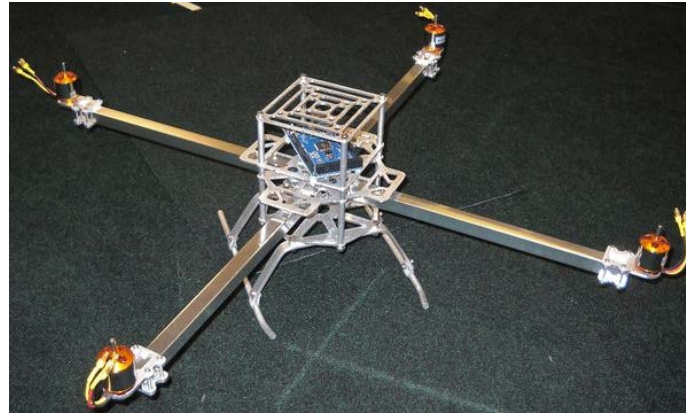


Şekil 3-6 Aeroquad v2.1 donanım kartı

Sistemin alıcı görevi, aynı zamanda bu multikopter sistemini diğerlerinden farklılaştırmakta olan bir GPRS/3G elektronik modülü ile gerçekleştirmektedir. Modül olarak Libelium firması tarafından üretilmekte olan 3G + GPS shield for Arduino komponenti kullanılmaktadır. Çalışmanın kilometre taşlarından birisi, bu komponentin kullanımı ve programlaması olduğundan, “GSM / GPRS Modül ve Kullanımı” başlıklı bölümde ayrıca detaylandırılmaktadır.

### 3.2.1.2 Kasa

Multikopter sisteminin tüm donanım bileşenleri kasa üzerinde monte edilmiştir. Şekil – 3.7’de kasanın monte edildikten sonraki durumu görülebilmektedir [16]. Kasanın motordan motora uzaklığı 550 mm olup, boş ağırlığı 275 gr ağırlığındadır.



Şekil 3-7 Multikopter kasa

### 3.2.1.3 Yardımcı Bileşenler

Multikopterin kasa ve uçuş kontrol kartı dışında, tüm bileşenleri yardımcı bileşenler olarak adlandırılabilir. Bunlar;

- Motorlar
- Elektronik Hız Kontrolcleri
- Batarya
- Pervaneler

Motorlar, pervaneler ve elektronik hız kontrolcleri birlikte multikopterin uçuşunu saęlayan en önemli bileşenlerdir. Uçuş kontrol kartının dış pinlerinin saęladığı komutlar elektronik hız kontrolclerinin giriş pinlerine bağlanmaktadır. Elektronik hız kontrolcleri aldıkları komutları motorlara iletmekle görevlidirler. Motorlara baęlı pervaneler ise, uçuşun gerekleşmesini saęlayan bileşenlerdir.

Sistemde kullanılan motorlar, BP firması tarafından üretilmekte olan fırçasız, dıştan dönümlü A2217-9 model numaralı motorlardır. Şekil – 3.8’ de örnek bir motoru göstermektedir.



Şekil 3-8 A2217/9 Fırçasız motor

BP A2217-9 model numaralı motora ait özellikleri tablo 3.1’deki gibidir.

Table 3-1 A2217-9 model motorun özellikleri[17]

<b>Destekledięi Batarya</b>	2 veya 3 hücreli Li-po Piller
<b>Üretilen Maksimum Güç</b>	200W
<b>Destekledięi Maksimum Akım</b>	5A - 15A
<b>alışmaya Başlama Akımı</b>	10V gerilimde 0,9A



<b>Ağırlık</b>	73,4 gr
<b>Şaft Çapı</b>	4 mm
<b>Maksimum Kaldırabileceği Ağırlık</b>	1000g

Multikopterde kullanılan pervaneler sistemin havalanmasını sağlamakta ve boyutları sistemin kaldırabileceği maksimum yük kapasitesini belirlemede önemli rol oynamaktadır. Multikopter sistemlerinde uçuşun sağlanması için kullanılan pervaneler, çeken ve iten pervanelerdir. Çeken pervaneler saatin tersi yönde dönerken, iten pervaneler saat yönünde dönmektedir. Çalışmada ikişer adet APC marka 10x4.7 cm boyutlarında iten ve çeken pervaneler kullanılmaktadır. Şekil –3.9 örnek bir pervaneyi göstermektedir.



**Şekil 3-9 APC 10\*4.7 Pervane**

Uçuş kontrol kartından aldığı komutları motorlara ileten elektronik hız kontrolcülerini ise HobbyWing tarafından üretilen, maksimum 30A akım sağlayabilen Pentium 30A model kontrolcülerdir. Şekil – 3.10’da elektronik hız kontrolcülerine ait bir örnek gösterilmektedir.



Şekil 3-10 Pentium 30A ESC

Çalışmada sisteme güç sağlaması için ZIPPY marka Flightmax 5000 20c model batarya kullanılmaktadır. Batarya üç hücreli olup, 11.1V gerilim sağlamaktadır. Batarya tam dolu olduğunda 5000A akım depolayabilmektedir. Sisteme getirdiği ağırlık yükü ise 404 gr.'dır. Bu çalışmada geliştirilen multikopter sistemi, tam dolu bir batarya ile yaklaşık on beş dakika aralıksız olarak havada kalabilmektedir. Şekil 3.11'de batarya gösterilmektedir.



Şekil 3-11 5000A Batarya

Bu bölümde hazırlanmış olan multikopter sistemi donanımsal olarak açıklanmıştır. Sistemin tüm donanımlarının monte edildikten sonraki hali Şekil – 3.12’de gösterilmektedir.



Şekil 3-12 Multikopter

### 3.3 Uçuş Kontrol Yazılımı ve Uyarlaması

Uçuş kontrol kartı, sistemin istikrarlı bir şekilde uçuşunu gerçekleştirebilmek için ihtiyaç olunan uçuş kontrol yazılımını içermektedir. Aeroquad projesi kapsamında geliştirilen yazılım, bu tez kapsamında eklenen GSM modülüyle iletişime geçecek ve hücresel ağ üzerinden komut alabilmesini sağlayacak şekilde yeniden düzenlenmiştir.

Uçuş kontrol yazılımı, Aeroquad proje ekibi tarafından ilk olarak 24.04.2009 tarihinde final versiyonu olarak piyasaya sürülmüştür. Açık kaynaklı proje kapsamında zamanla farklı özellikleri de destekleyecek şekilde güncellenen ve geliştirilen yazılım son olarak 3.2 versiyonu ile 28.01.2013 tarihinde yayınlanmıştır [14].

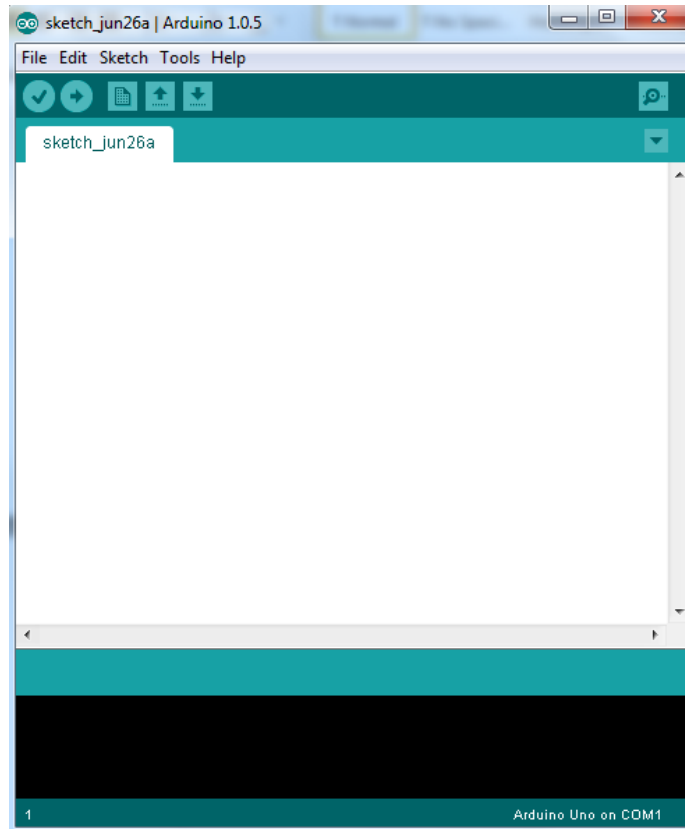
Aeroquad uçuş kontrol yazılımı 3.2 versiyonu itibarı ile multikopter üzerinde aşağıdaki konfigürasyonlara destek verebilir hale getirilmiştir [14].

- **Desteklenen Multikopter Tipleri**
  - Quad X, Quad +, Quad Y4
  - Tri, Hex X, Hex +, Hex Y6
  - Octo X, Octo + and Octo X8
- **Desteklenen Uçuş Opsiyonları**
  - Gyro veya Manyetometre ile Rota Sabitleme
  - Barometre ile yükseklik sabitleme
  - Sonar uzaklık sensörlü ile yükseklik sabitleme
- **Batarya Özellikleri**
  - Pil azaldığında otomatik yavaş iniş
  - Pil hücre sayısını belirleyebilme
  - (OSD) On Screen Display özelliği ile entegrasyon
- **Alıcı Opsiyonları**
  - Maksimum 10 kanallı alıcı desteği
  - PWM alıcılar
  - PPM alıcılar
  - Donanım zamanlayıcısı ile PPM alıcılar
  - Futaba sBus alıcı
- **Telemetri Opsiyonları**
  - Kablosuz telemetri
  - Ses dalgaları ile telemetri
  - Temel MavLink protokolü entegrasyonu
- **Kamera Stabilizasyon Opsiyonları**
  - Aşağı-yukarı, sağ-sol ve kendi etrafında dönüş için özel ayrılmış alıcı kanalları
  - Sisteme entegre edilen servo motorlarla uzaktan manüel ve uçuş kontrol yazılımı ile otomatik kamera stabilizasyonu
- **MAX7456 modülü ile On-Screen-Display Desteği**

- Kullanılacak video standardını seçebilme
- Kurulan görüntü bağlantısına özel isim verebilme
- Yükseklik Gösterimi
- Uzaktan PID ayarı yapabilme

Uçuş kontrol yazılımı, Arduino programlama kartı üzerine, arduino programlama dili kullanılarak geliştirilmiştir. Arduino programlama dili C/C++ baz alınarak geliştirilmiş bir programlama dilidir [15].

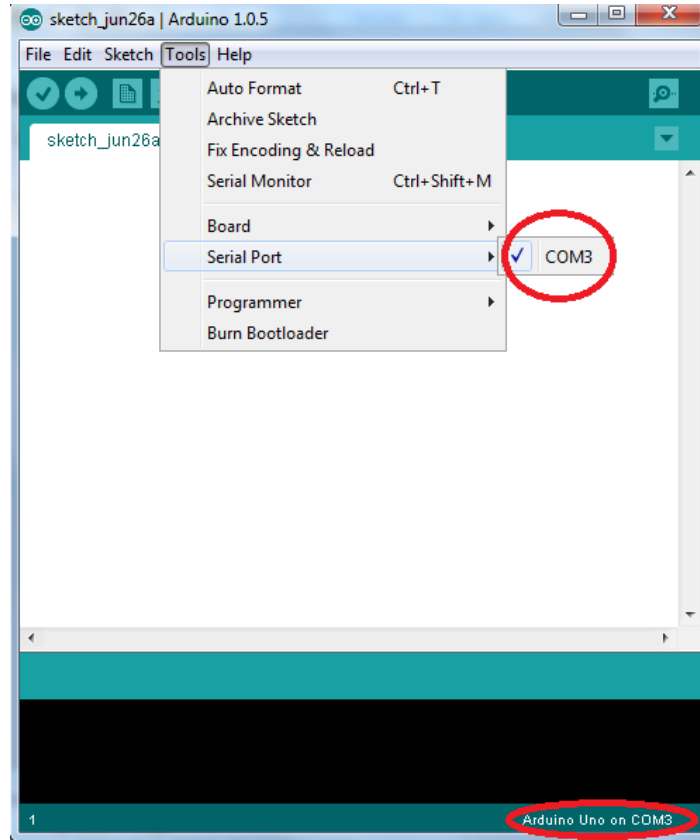
Arduino programlama dili ile geliştirilecek programlar skeç olarak adlandırılır ve Arduino programlama (IDE) ortamı kullanılarak geliştirilmektedir. Arduino programlama ortamı açık kaynak kodlu bir projedir. Dolayısı ile lisans gerektirmeden Arduino projesinin internet sitesinden edinilebilmektedir. Şekil – 3.13’de Arduino programlama ortamı görülebilmektedir.



Şekil 3-13 Arduino programlama ortamı

### 3.3.1 Uçuş Kontrol Yazılımının Uçuş Kontrol Kartına Yüklmesi

Arduino programlama ortamında geliştirilen skeçler tamamlandıktan sonra Arduino programlama kartına yüklenmektedir. Bunun için kullanılan Arduino programlama kartının, geliştirilenin yapıldığı bilgisayara USB arayüzü ile bağlanması gerekmektedir. USB arayüzü ile bağlı olan Arduino programlama kartının, bilgisayara hangi porttan bağlı olduğunun da bilinmesi gerekmektedir. Bu çalışma kapsamında kullanılmış olan Arduino Mega 2650 programlama kartı on yedi numaralı port üzerinden bilgisayara bağlanmaktadır. Bağlantı kurulan portun Arduino programlama ortamında da aşağıdaki Şekil – 3.14’te görülebildiği üzere tanıtılması gerekmektedir.



Şekil 3-14 Arduino programlama kartı için bağlantı portu seçimi

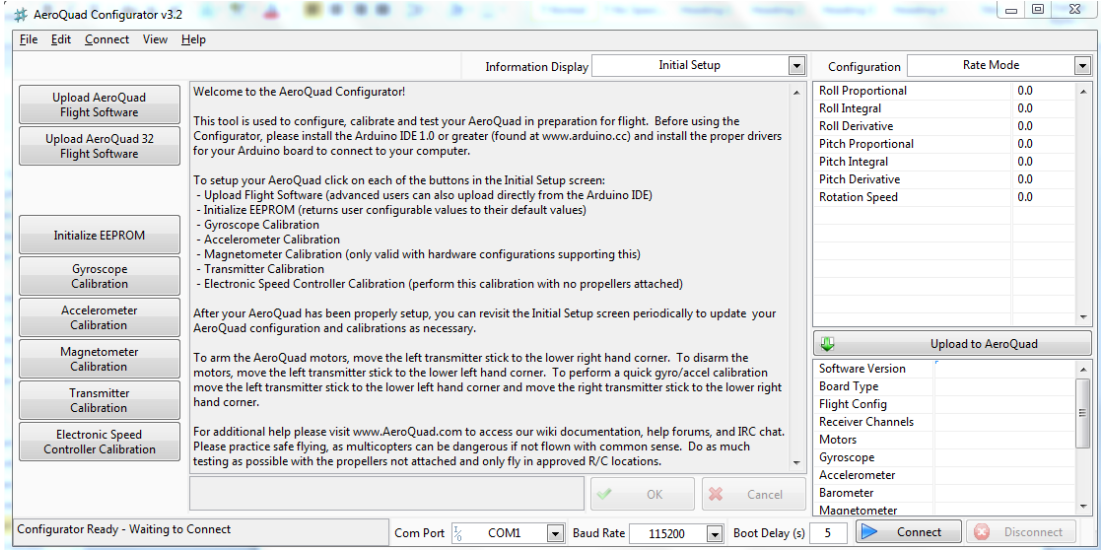
Bu tez çalışmasında, Aeroquad projesi kapsamında geliştirilen Aeroquad Uçuş Kontrol Yazılımı üzerinde yeniden düzenlemeler yapılmakta ve GSM modül ile gerçekleştirilmesi gereken iletişim sağlanmaktadır.

Uçuş kontrol yazılımı Aeroquad proje ekibi tarafından zaman zaman güncellenerek yayınlanmaktadır. Her yeni yayınlanan yazılımın Arduino programlama kartının belleğine yüklenmesi gerekmektedir.

Aeroquad uçuş kontrol yazılımının yeni versiyonunun yayınlanmasının dışında, kullanılan versiyon üzerinde bir konfigürasyon yapılmak istendiğinde de, yazılımın Arduino programlama kartına yüklenmesi gerekmektedir.

Aeroquad uçuş kontrol yazılımının, Arduino programlama kartına yüklenmesi için yine Aeroquad projesi kapsamında geliştirilen “Aeroquad Configurator” yazılımı kullanılmaktadır. Bu yazılım ile Aeroquad uçuş kontrol yazılımının parametrik konfigürasyonlarının daha pratik olarak yapılması amaçlanmıştır. Konfigürasyon yazılımı açıldığında öncelikle Şekil – 3.15’te görülen arayüz açılmaktadır.

Bilgisayara USB arayüzü vasıtası ile bağlı olan Arduino programlama kartı bazlı uçuş kontrol kartının, bilgisayara hangi port üzerinden bağlandığı tespit edilmiştir. Tespit edilen bu port bilgisi Şekil – 3.15’te görülen Aeroquad konfigürasyon arayüzünün alt bölümünden ayarlanmıştır. Dolayısı ile yapılandırma (konfigürasyon) programından uçuş kontrol kartına bağlantı sağlamak için bu port kullanılmış olacaktır. Port bilgisi dışında, yapılandırma arayüzünde bulunan seri port bağlantı hızını da 115200 olarak seçilmiştir. Yapılandırma ara yüzünde yer alan “Connect” etiketli tuş yardımı ile yapılandırma (konfigürasyon) programı ile uçuş kontrol yazılımı arasında bağlantı kurulmuştur.



**Şekil 3-15 Uçuş kontrol yazılımı yükleme ve yapılandırma (konfigürasyon) ara yüzü**

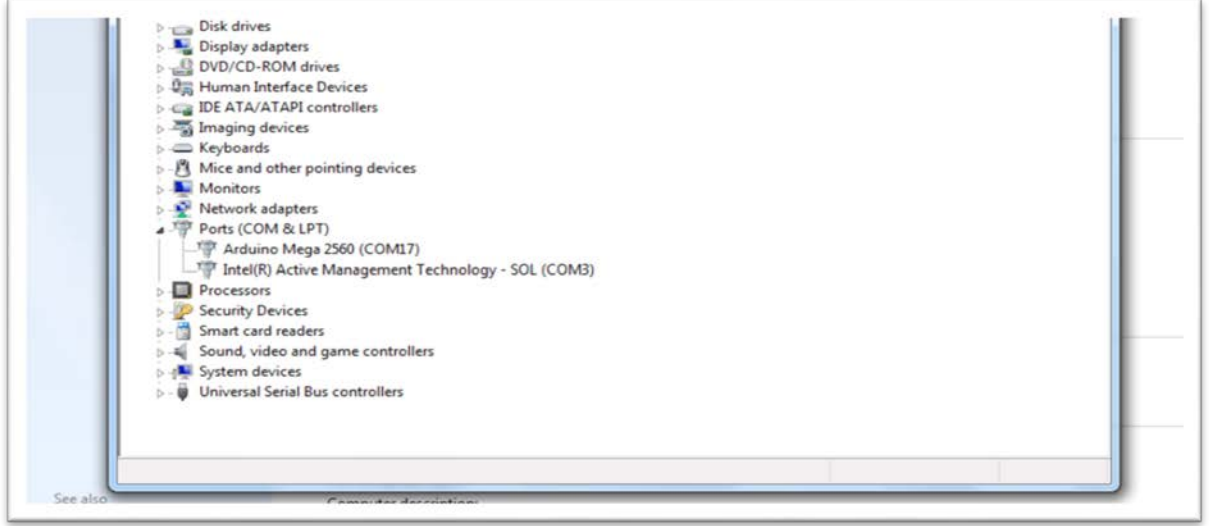
Aeroquad konfigürasyon programı ile uçuş kontrol kartı arasındaki bağlantı sağlandıktan sonra, uçuş kontrol yazılımı uçuş kontrol kartına yüklenebilir duruma getirilmiştir. Bu arayüzde görülen “Initial Setup” – “Başlangıç Ayarları” sekmesinde, Aeroquad uçuş kontrol yazılımının, uçuş kontrol kartına yüklenmesi sağlanabilmiştir. Ayrıca bu ekranda uçuş kontrol kartına yüklenen yazılımın, başlangıç parametrelerinin yüklenmesi, gyro, akselerometre, manyetometre, kumanda ve elektronik hız kontrolcülerini bileşenlerinin kalibrasyonları da yapılmıştır. Bu kalibrasyonların doğru bir şekilde yapılması multikopterin istikrarlı (stabil) bir şekilde uçuş gerçekleştirebilmesi için bir zorunluluktur. Aksi takdirde multikopter düzensiz hareketler sergileyecektir.

Uçuş kontrol yazılımının, uçuş kontrol yazılımına yüklenmesi için aşağıdaki adımlar izlenmektedir.

- Uçuş kontrol kartı, yazılımı yüklemek için kullanılacak bilgisayara bağlanmalıdır. Bağlantı uçuş kontrol kartında ve bilgisayardaki USB ara yüzü vasıtası ile gerçekleştirilir. Sonrasında bilgisayarın donanım ayarları kontrol edilir ve Şekil – 3.16’da görüldüğü gibi uçuş kontrol kartının

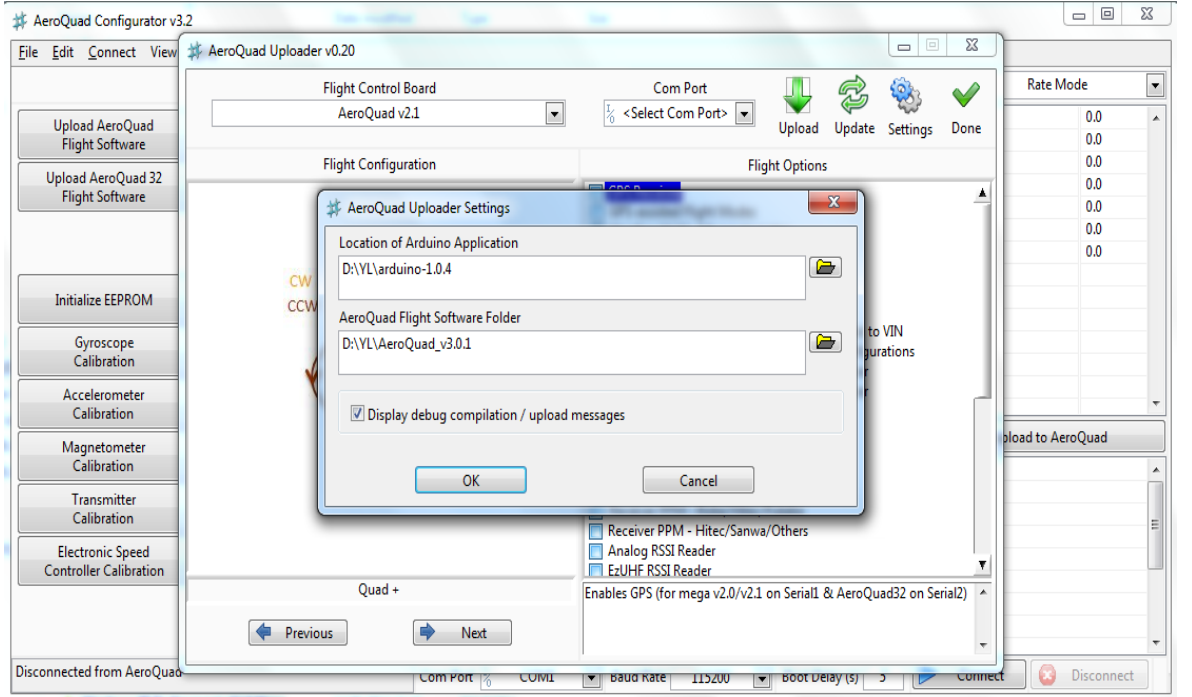


bilgisayara hangi port üzerinden bağlandığı tespit edilmesi gerekir. Bu port bilgisi ilerleyen adımlarda kullanılacaktır.



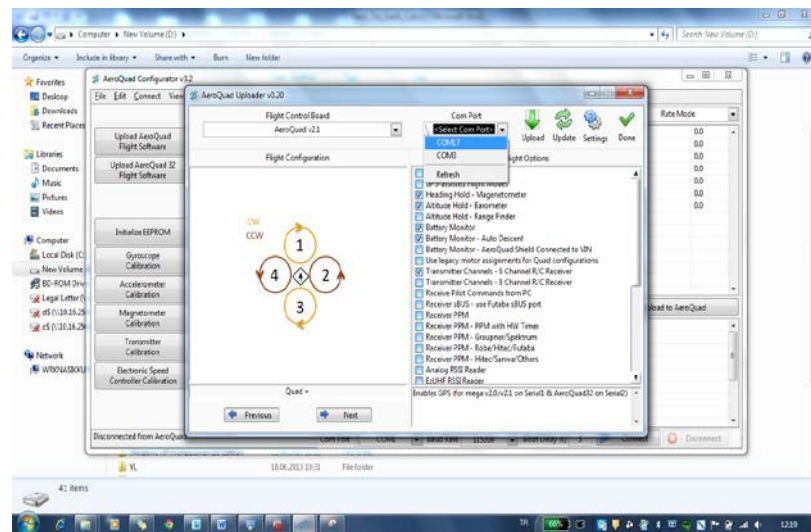
Şekil 3-16 Uçuş kontrol kartının bilgisayara bağlantı portunun tespit edilmesi

- Bilgisayarda kurulu olan Aeroquad Configurator programı açılır ve program ana ekranındaki “Upload Aeroquad Flight Software” tuşu vasıtası ile uçuş kontrol yazılımının yükleneceği ekrana giriş yapılır. Ekranı ilk giriş yapıldığında, Arduino programlama ara yüzünün kurulu olduğu dizin ve yüklenecek uçuş kontrol yazılımının olduğu dizin bilgisi Şekil – 3.17’de olduğu gibi ekrandaki ilgili alanlara girilmelidir.



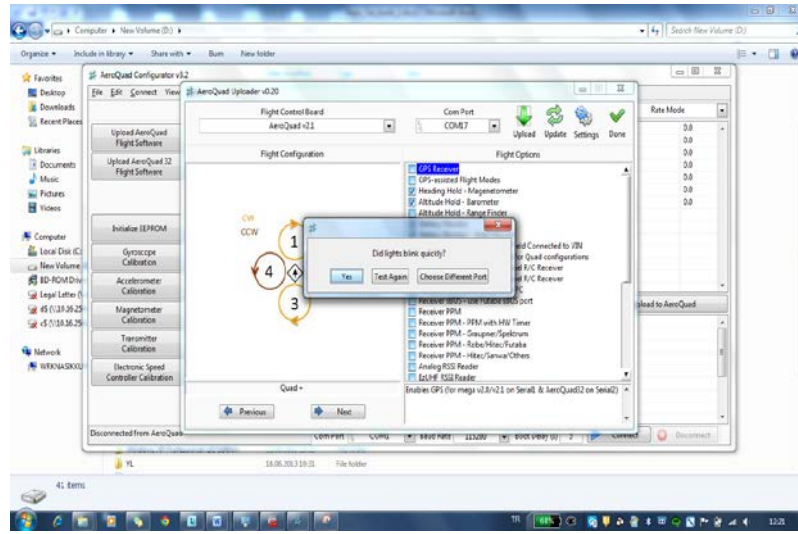
Şekil 3-17 Arduino programlama ara yüz ortamının kurulu olduğu dizinin belirtilmesi

- Uçuş kontrol kartının bilgisayara üzerinde bağlı olduğu port bilgisi şekil 3.18’de olduğu gibi seçilir.



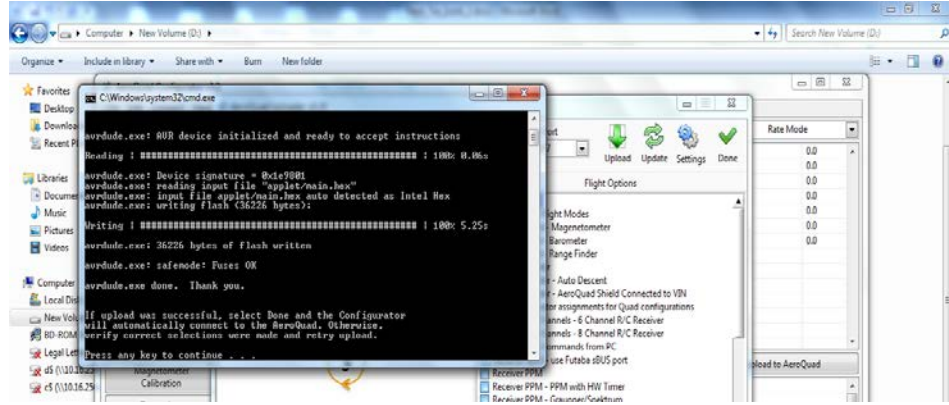
Şekil 3-18 Uçuş kontrol kartının bilgisayara üzerinde bağlı olduğu portun seçilmesi

- Ekranda bulunan “Upload” tuşuna basılır. Bu aşamada sistem uçuş kontrol kartına olan bağlantıyı test etmek için Şekil – 3.19 ‘da da görüleceği üzere bir test uygular. Bu test, uçuş kontrol kartı üzerinde bulunan kontrol ışığının birkaç defa yanıp sönmelerini test eder. Eğer yanıp sönmeleri oluyorsa, test sonucu başarılıdır. Aksi halde bir bağlantı problemi vardır ve test sonucu başarısızdır.



Şekil 3-19 Uçuş kontrol yazılımının yüklenme testi ve yüklenmeye başlaması

- Bir önceki adımda başarılı olan bir test sonucunda, Şekil – 3.20’de görüleceği üzere yükleme işlemi gerçekleştirilir.



Şekil 3-20 Uçuş kontrol yazılımı yüklemesinin tamamlanması

### 3.3.2 Uçuş Kontrol Yazılımı Uyarlaması

Bu çalışmayı, diğer çalışmalardan ayıran en önemli özellik olan, Aeroquad uçuş kontrol yazılımının, multikopterin kendisine monte edilen GPRS / 3G modül ile haberleşebilmesi ve multikopterin hücresel ağlar üzerinden sağlanan bağlantı ile kontrol edilebilmesini imkan veren yeteneğe kavuşması bu tez kapsamında yerine getirilmiştir.

Uçuş kontrol kartı ve GPRS/3G modülün haberleşebilmesi için öncelikle donanımsal olarak birbiri ile haberleşebilmesi sağlanmaktadır.. Bölüm 3.4'te, uçuş kontrol kartı ve GPRS/3G modülün donanımsal olarak bağlantısı ayrıntılı olarak anlatılmaktadır.

Uçuş kontrol kartı ve GPRS/3G kartının birbiri ile donanımsal olarak bağlantısı sağlandıktan sonra, uçuş kontrol yazılımı üzerinde yapılan düzenlemeler ile uçuş kontrol kartı ve GPRS/3G modülü birbirine data alıp/verebilir duruma getirilmiştir.

Uçuş kontrol yazılımını ve yapılan düzenlemeleri anlayabilmek için öncelikle Arduino programlama kartının yazılımsal olarak nasıl çalıştığını anlamak gereklidir. Arduino programlama kartı genel olarak iki prosedür üzerinde çalışmaktadır. Bunlar;

- setup,
- loop

prosedürleridir.

**Setup prosedürü:** Arduino programlama kartının elektrik akımını aldığı anda ilk olarak çağırdığı prosedürdür. Bu prosedür içerisinde, Arduino programlama kartının en çok bir defa gerçekleştireceği ve başlangıç hazırlıkları olarak da ifade edilebilecek işlemleri gerçekleştirmektedir.

**loop prosedürü:** setup prosedürü çalışmasını tamamladıktan sonra çağırılan prosedürdür. Bu prosedür içerisinde yazılan işlemler, Arduino programlama kartı kapalı konuma getirilene kadar sürekli olarak tekrar edecek şekilde yürütülür. Bu prosedür bir anlamda sonsuz döngü olarak da tanımlanabilir.

Aeroquad uçuş kontrol yazılımı içerisinde kullanılan “setup” prosedürü, multikopterin ilk çalışma anında çağırılarak, sistemin seri bağlantı hızı, motor ayarlarının algılanması, alıcı modülün algılanması, sistemdeki sensörlerin algılanmasını ve kalibrasyonu gibi bir çok çalışma başlangıç anı işlemlerini gerçekleştirir.

Aeroquad uçuş kontrol yazılımındaki “loop” prosedürü incelenecek olursa, sistemin çalışması için sürekli tekrar etmesi gereken işlemlerin, kendi aralarında zaman frekanslarına bölündükleri anlaşılmaktadır. Bu yöntem ile uçuş kontrol kartının kaynakları, sistemde tekrar eden işlemlerin öncelikleri göz önüne alınarak, daha verimli kullanılabilir hale getirilmiştir. Tablo – 3.2’de “loop” prosedürü içerisinde çalışan prosedürlerin çalışma frekansları gösterilmektedir. Görüldüğü üzere alıcıdan alınan veri 20000 mikro saniyede bir defa okunurken, barometre sensörü 10000 mikro saniyede bir okunmaktadır. Yani barometre sensörü, alıcıya göre iki kat daha sık kontrol edilmektedir.

Tablo 3-1 “loop” prosedüründe çalışan görevler ve çalışma frekansları

Görev	Açıklama	Mikrosaniye
ReadGyro	Gyro sensör okuması	Sürekli
ReadAccel	Akselerometre sensör okuması	Sürekli
RunDCM	Anlık hareket durumu hesaplaması	( 10000); // 100hz
FlightControls	Uçuş kontrol verisi hesaplama	( 10000); // 100hz
ReadBaro	Barometre sensörü okuması	( 10000); // 100hz
ReadReceiver	Alıcı cihaz sinyal okuması	( 20000); // 50hz
ReadCompass	Yön verisi okuması	( 100000); // 10Hz
ProcessTelem	Uzak telemetri cihaz verisi okuması	( 100000); // 10Hz
ReadBattery	Batarya durumu okuması	( 100000); // 10Hz

Bu çalışma kapsamında, Aeroquad uçuş kontrol yazılımının ana iki prosedürü üzerinde gerçekleştirilen düzenlemeler ile yazılımın GPRS/3G modülü üzerinden yönetilebilmesi sağlanmıştır. Bu amaçla Aeroquad uçuş kontrol yazılımı üzerinde geliştirilen uyarlamalar, “setup” prosedürü kapsamında geliştirilen uyarlamalar ve “loop” prosedürü kapsamında geliştirilen uyarlamalar olarak iki bölümde incelenecektir.

### 3.3.2.1 “Setup” Prosedürü Kapsamında Geliştirilen Uyarlamalar

Her Arduino programlama kartının çalışma prensibinde olduğu gibi, Aeroquad uçuş kontrol kartının çalışma prensibi de, uçuş kontrol yazılımındaki “setup” prosedürü içerisinde, sistemde en fazla bir defa çalışacak, başlangıç işlemlerinin gerçekleştirilmesi şeklindedir. Bu amaçla Aeroquad uçuş kontrol yazılımının, hali hazırda “setup” prosedürü içerisinde bulunan başlangıç işlemlerine ek olarak, GPRS/3G modülü ile iletişim kurulabilmesi için öncelikle GPRS/3G modülünün açık ve iletişime hazır konuma getirilmesi gerekmektedir. Dolayısı ile sistemdeki “setup” prosedürünün içerisindeki alt prosedürlere altı numaralı satırda bulunan “activateModuleNetwork” belirteci ile çağırılan yeni bir prosedür geliştirilmiştir. Sistemdeki “setup” prosedürünün düzenlenmiş sürümü Şekil – 3.21’de görülebilmektedir.

```

1 void setup() {
2   SERIAL_BEGIN(BAUD);
3   pinMode(LED_Green, OUTPUT);
4   digitalWrite(LED_Green, LOW);
5   //GPRS/3G Modülüne güç verilmekte ve çalışmaya hazır hale getirilmektedir.
6   activateModuleNetwork();
7   // Read user values from EEPROM
8   readEEPROM(); // defined in DataStorage.h
9   if (readFloat(SOFTWARE_VERSION_ADR) != SOFTWARE_VERSION) {
10    initializeEEPROM();
11    writeEEPROM();
12  }
13  initPlatform();
14  ...
15

```

Şekil 3-21 “SETUP” prosedürü uyarlaması

“activateModuleNetwork” prosedürü içerisinde GPRS/3G modülünün aktif edilmesinden, iletişime hazır hale getirilmesine kadar gerekli işlemler gerçekleştirilmektedir. Bu işlemler Şekil – 3.22’de görülebilmektedir. İlk olarak altı numaralı satırda “switchModule()” prosedürü ile GPRS/3G modülünün aktif edilmesi sağlanmaktadır. GPRS/3G modülünün GSM ağına bağlanması ve kullanıma hazır hale gelmesi için kırk sekiz byte uzunluğunda datanın GPRS/3G modülünden uçuş kontrol kartına gelmesi beklenmektedir. Bu data modülün hazır hale geldiğini içeren ifadeleri içermektedir. On bir numaralı satırda, kullanıma hazır hale gelen GPRS/3G modülüne internet ağına bağlanabilmesi için kullanılacak erişim parametreleri gönderilmektedir. On üç numaralı satırda ise GPRS/3G modülünün TCP/IP protokolünü kullanarak internet ağına bağlanması sağlanmaktadır. On sekiz numaralı satırda da GPRS/3G modülünün internet ağından gelen bağlantı taleplerini kabul etmesi için sunucu moduna geçirilmesi görülmektedir. On dokuz numaralı satırda GPRS/3G modula gelen datada bulunan başlık verisinin göz ardı edilmesi için gerekli parametre değeri değiştirilmekteyken, bir sonraki satırda yine gelen datanın uzunluk verisinin göz ardı edilmesi için gerekli parametre değiştirilmektedir.

```

1 void activateModuleNetwork()
2 {
3   Serial.println("Starting Serial!"); Serial3.begin(115200); delay(2000);
4   Serial.println("Serial Started!"); pinMode(onModulePin, OUTPUT);
5   Serial.println("3G Module Preparing...");
6   switchModule(); while(Serial3.available() !=48);
7   int count=0; count=Serial3.available(); Serial.println(count);
8   if (count>0) { int i=0; byte data; while(i<count) { Serial3.read(); i++; }}
9   Serial3.println("AT+CMGF=1"); delay(100);
10  while(Serial3.read() !='K');
11  Serial3.println("AT+CGSOCKCONT=1,\"IP\", \"mgbs\""); delay(100); while(Serial3.read() !='K');
12  Serial3.flush();
13  Serial3.println("AT+NETOPEN=\"TCP\",80"); delay(10); Serial3.flush(); delay(90);
14  x=0;
15  do{ while(Serial3.available() ==0); data[x]=Serial3.read(); Serial.print(data[x]); x++;
16  }while(!(data[x-1]=='d' && data[x-2]=='e'));
17  Serial3.flush(); delay(100); Serial.println(); delay(1000);
18  Serial3.println("AT+SERVERSTART"); delay(100); while(Serial3.read() !='K'); Serial3.flush();
19  Serial3.println("AT+CIPHEAD=0"); delay(100); while(Serial3.read() !='K'); Serial3.flush();
20  Serial3.println("AT+CIPSRIP=0"); delay(100); while(Serial3.read() !='K');
21  while(Serial3.available() !=0) { Serial3.read(); } Serial.println("3G Module Ready!");
22 }

```

Şekil 3-22 GPRS/3G modülün Arduino programlama kartı ile hazır hale getirilmesi

“switchModule” prosedürü şekil – 3.23’de görülebilmektedir. Burada uçuş kontrol kartının 2 numaralı pin bacağından, GPRS/3G modülün 4 numaralı pin bacağına iki saniye boyunca gerilim uygulanması için komut verilir. Bu komut kümesi “switchModule” prosedürü içerisinde yer almaktadır.

```

1 void switchModule() {
2   digitalWrite(onModulePin, HIGH);
3   delay(2000);
4   digitalWrite(onModulePin, LOW);
5 }

```

Şekil 3-23 GPRS/3G modülünü çalıştırmak için gerilim uygulanması

### 3.3.2.2 “Loop” Prosedürü Kapsamında Geliştirilen Uyarlamalar

Arduino programlama kartının iki ana prosedüründen ikincisi olan “loop” prosedürü, programlama kartının rutin işlevinin yerine getirildiği ana prosedürdür. Bu prosedür içerisindeki kod bloğu bir sonsuz döngü şeklinde programlama kartı pasif edilene kadar çalışmaktadır.



Aeroquad uçuş kontrol programında yer alan “loop” prosedürü daha önce genel hatları ile anlatılmıştır. Bu çalışma kapsamında yapılan düzenlemeler ise bu kısımda anlatılmaktadır.

Çalışmadan kullanılan multikopterin, GPRS/3G modül vasıtası ile hücresel ağlardan gönderilen komutlarla yönetilebilmesi için uçuş kontrol yazılımı üzerinde geliştirilen düzenleme genel olarak alıcı verisinin okunması için çağırılan “*readPilotCommands*” prosedürü içerisinde gerçekleştirilmektedir. Daha önce uçuş kontrol kartı üzerine monte olan alıcıdan alınan komut verisi, yapılan düzenleme ile GPRS/3G modülünden gelen verilerle elde edilebilir hale getirilmiştir.

“loop” prosedürü içerisinde çağırılan “*readPilotCommands*” prosedürünün, sistemdeki işlevi, yer istasyonundan alınan verileri elde edip uçuş kontrol programının motorlara göndereceği komutları üretmesi için girdi olarak sağlamaktır. Uçuş kontrol yazılımı motorlara göndereceği komutları belirlerken sadece yer istasyonundan gelen verileri değil, ayrıca uçuş kontrol kartı üzerinde yer alan sensörlerden aldığı verileri de girdi olarak değerlendirmektedir.

Çalışma kapsamında, yer istasyonu ile uçuş kontrol kartı arasındaki haberleşme arasında gönderilen komut dizisi formatı Tablo – 3.3’te gösterilmektedir. Uçuş kontrol yazılımı yer istasyonundan maksimum 8 kanal için komut alabilmektedir. Bu çalışmada ise 4 ana kanal üzerinde çalışılmıştır. Bunlar throttle, X boyut, Y boyut ve Z boyut kanallarıdır. Her kanal değeri için uçuş kontrol yazılımı 1000 – 2000 değerleri arasında girdi almaktadır. Throttle kanalı multikopterin yükselip alçalmasını, X boyut kanalı sağa veya sola yatmasını, Y boyut kanalı öne veya arkaya yatmasını ve Z boyut kanalı ise kendi ekseninde sağa veya sola dönmesini sağlamaktadır.

Tablo 3-2 Yer istasyonundan uçuş kontrol kartına gönderilen komut dizisi formatı

İFADE	AÇIKLAMA
@TTTT;XXXX;YYYY;ZZZZ!	Yer İstasyonundan Uçuş Kontrol Kartına Gönderilen Komut Dizisi
@	Komut Başlangıç Karakteri
T	Throttle komutu için 0-9 arasında karakter
X	X boyutu komutu için 0-9 arasında karakter
Y	Y boyutu komutu için 0-9 arasında karakter
Z	Z boyutu komutu için 0-9 arasında karakter
!	Komut Sonlandırma Karakteri

“loop” prosedüründe gerçekleştirilen uyarlamalar genel olarak şekil – 3.24’te görülebilmektedir. Burada bir numaralı satırda, bir önceki komut ile son okunan komut arasındaki aşırı fark olup olmadığı veya daha önceden tespit edilen aşırı farkın normalleştirilip normalleştirilemediği kontrol edilir. Bir önceki komutlar ile şimdiki komutlar arasındaki fark normal ise veya daha önceden tespit edilen aşırı fark senkronize halde iki numaralı satırda GPRS/3G modülden gelen veri okunmaktadır. Üç numaralı satırda okunan son komutun bir önceki komuta göre aradaki farkı hesaplanır. Bu hesap işlevi şekil – 3.25’te ayrıntılı olarak gösterilen “*compareCommandActualValues*” fonksiyonu ile gerçekleştirilmektedir. Aradaki fark normal ise, uçuş kontrol yazılımına gidecek parametreler son okunan komut değerleri ile set edilir.

```

1 if (HugeDifference=="Normal" && Synchronised){
2   readRemoteValues();
3   HugeDifference=compareCommandActualValues();
4   if (HugeDifference=="Normal") {
5     if (int(xaxis_i/1000)>=1) { xaxis_i_old=xaxis_i; }
6     if (int(yaxis_i/1000)>=1) { yaxis_i_old=yaxis_i; }
7     if (int(zaxis_i/1000)>=1) { zaxis_i_old=zaxis_i; }
8     if (int(throttle_i/1000)>=1){ throttle_i_old=throttle_i; }
9   }
10  else{
11    Synchronised=SyncActualandInstantValues(HugeDifference);
12    if (Synchronised) { HugeDifference="Normal"; }
13  readPilotCommands();

```

Şekil 3-24 “loop” prosedüründe gerçekleştirilen uyarlamaları

```

1 String compareCommandActualValues ()
2 {
3   int BigDiffFlag=0;
4   String Result="Normal";
5   if (abs(throttle_i-throttle_i_old)>=100 && BigDiffFlag==0)
6   {
7     BigDiffFlag=1;
8     Result="Throttle";
9   }
10  if (abs(xaxis_i-xaxis_i_old)>=100 && BigDiffFlag==0)
11  {
12    BigDiffFlag=1;
13    Result="Xaxis";
14  }
15  if (abs(yaxis_i-yaxis_i_old)>=100 && BigDiffFlag==0)
16  {
17    BigDiffFlag=1;
18    Result="Yaxis";
19  }
20  if (abs(zaxis_i-zaxis_i_old)>=100 && BigDiffFlag==0)
21  {
22    BigDiffFlag=1;
23    Result="Zaxis";
24  }
25  return Result;
26 }

```

Şekil 3-25 “compareCommandActualValues” fonksiyonu

Bir önce okunan komutlar ile son okunan komutlar arasında aşırı fark olması tespit edilmesi durumunda, bir önce okunan komutlar son okunan komutlara eşitlenecek şekilde sabit bir değer kullanılarak artırım/azaltım yöntemi ile şekil – 3.26’da ayrıntılı olarak gösterilen “*SyncActualandInstantValues*” fonksiyonu içerisinde senkronize edilerek uçuş kontrol yazılımına iletilmektedir. Aksi takdirde, bir önce okunan komutlar ile son okunan komut arasında aşırı fark olması durumunda multikopter ani hareketlenmelere sebep olabilmektedir. Dolayısı ile son okunan değerler mümkün olduğunda yumuşatılarak uçuş kontrol kartına gönderilmektedir.

```

1 boolean SyncActualandInstantValues(String HugeDiff)
2 {
3     delay(5); boolean Result=false;
4     if (HugeDiff=="Throttle") {
5         if (throttle_i<1000 || throttle_i>2000) {
6             if (throttle_i_old>=1000 && throttle_i<=2000) {
7                 throttle_i=throttle_i_old; }
8             else { throttle_i=1300; }
9         if (throttle_i>throttle_i_old){ throttle_i_old=throttle_i_old+20; }
10        if (throttle_i<throttle_i_old) { throttle_i_old=throttle_i_old-20; }
11        if (throttle_i==throttle_i_old) { Result=true; }
12        //Aşırı fark X boyutu için gönderilen komuttan kaynaklanıyorsa
13        if (HugeDiff=="Xaxis")
14            //Aşırı fark Y boyutu için gönderilen komuttan kaynaklanıyorsa
15            if (HugeDiff=="Yaxis")
16                //Aşırı fark Z boyutu için gönderilen komuttan kaynaklanıyorsa
17                if (HugeDiff=="Zaxis")
18                    //Senkronizasyon anında elde edilen kanal değerleri bilgi amaçlı gösterilmektedir.
19                    Serial.println("*****Synchronizing(old to remote)*****");
20                    Serial.println(String(xaxis_i_old)+" to "+String(xaxis_i));
21                    Serial.println(String(yaxis_i_old)+" to "+String(yaxis_i));
22                    Serial.println(String(zaxis_i_old)+" to "+String(zaxis_i));
23                    Serial.println(String(throttle_i_old)+" to "+String(throttle_i));
24        return Result;
25 }

```

Şekil 3-26 “SyncActualandInstantValues” fonksiyonu

Ayrıca yer istasyonundan gönderilen komut verilerinin elde edilmesi işlevi “*readPilotCommands*” prosedüründen çıkarılmıştır ve bu prosedür sadece elde

edilen yer istasyonu komut verilerinin uçuş kontrol yazılımına girdi olarak sağlanması işlevini yerine getirir hale getirilmiştir.

Yer istasyonundan gelen verilerin okunması, Şekil – 3.27’de da görüldüğü gibi, “*readRemoteValues*” prosedürü içerisinde çalışan işlevler ile elde edilmektedir. Prosedür her çağırıldığında GPRS/3G modülünden uçuş kontrol kartına gönderilmiş veri olup olmadığı test edilir. İki numaralı satırda bulunan “*Serial.available()*” işlevi uçuş kontrol kartının GPRS/3G modülü ile bağlantılı olduğu seri porta gelen verinin byte cinsinden büyüklüğünü geri döndürmektedir.

Yer istasyonundan gönderilen her komut satırı, tüm kanallar için komut içermektedir. Her gelen komut satırı “*readRemoteValues*” prosedürü içerisinde ayrıştırılarak, 4 kanal için ayrı ayrı değerler elde edilmektedir.

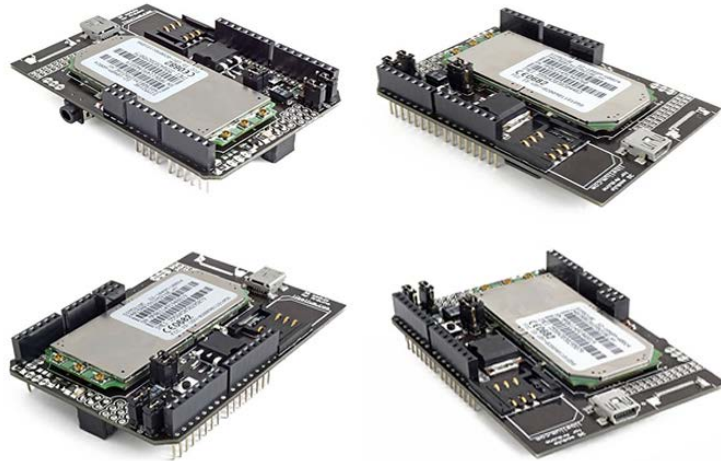
```
1 void readRemoteValues() { .....
2 int sa=0; sa=Serial.available(); int commandFlag=0; .....
3 while(Serial.available()>0 && commandFlag==0) .....
4 if (sa>0) { commandFlag=1; char str='a'; .....
5 char str1[1024]; int cx=0; Serial.println(sa); .....
6 do { str=char(Serial.read()); str1[cx]=str; cx++; .....
7 }while (str!='!' && cx<sa); .....
8 str1[cx]='\0'; String str2=str1; String tmp; .....
9 tmp=str2.substring(str2.indexOf('@')); .....
10 String th=tmp.substring(tmp.indexOf('@',0)+1,tmp.indexOf(';',0)); .....
11 tmp=tmp.substring(tmp.indexOf(';',0)+1); .....
12 String xa=tmp.substring(0,tmp.indexOf(';',0)); .....
13 tmp=tmp.substring(tmp.indexOf(';',0)+1); .....
14 String ya=tmp.substring(0,tmp.indexOf(';',0)); .....
15 tmp=tmp.substring(tmp.indexOf(';',0)+1); .....
16 String za=tmp.substring(0,tmp.indexOf('!',0)); .....
17 Serial.println(); .....
18 xa.trim(); ya.trim(); za.trim(); th.trim(); .....
19 if (atoi(str2char(xa))>999 && atoi(str2char(xa))<2001 && xa.length()==4) .....
20 xaxis_i = atoi(str2char(xa)); .....
21 if (atoi(str2char(ya))>999 && atoi(str2char(ya))<2001 && ya.length()==4) .....
22 yaxis_i = atoi(str2char(ya)); .....
23 if (atoi(str2char(za))>999 && atoi(str2char(za))<2001 && za.length()==4) .....
24 zaxis_i = atoi(str2char(za)); .....
25 if (atoi(str2char(th))>999 && atoi(str2char(th))<2001 && th.length()==4) .....
26 throttle_i = atoi(str2char(th)); .....
27 Serial.println("*****Remote Values*****"); .....
28 Serial.println(String(xaxis_i)); Serial.println(String(yaxis_i)); .....
29 Serial.println(String(zaxis_i)); Serial.println(String(throttle_i)); .....
30 if (nzz_counter>300) { mode_i=1900; aux1_i=1900; .....
31 Serial.println("Stability Mode Opened"); } }
```

Şekil 3-27 “*readRemoteValues*” fonksiyonu

### 3.4 GPRS/3G Modül ve Kullanımı

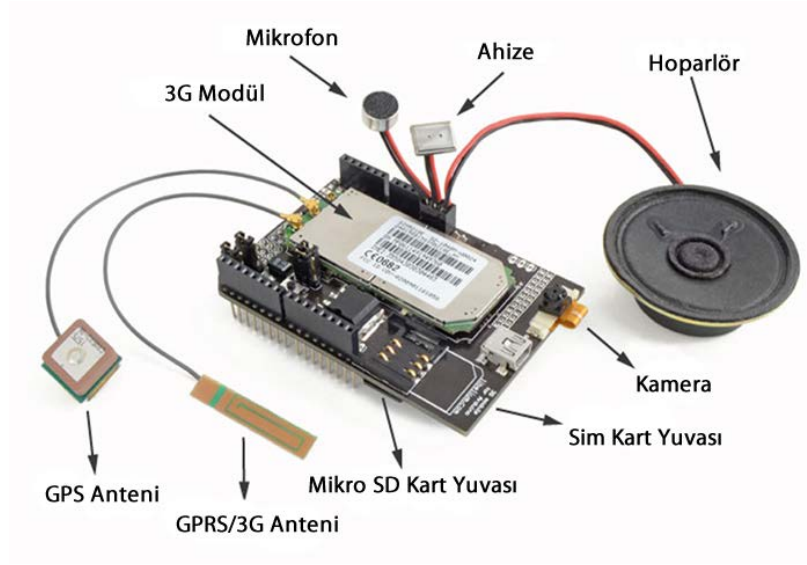
Günümüzde kullanılmakta olan mobil cihazlarda, cihazların bir birleri ile veri alış verişinde bulunmasına olanak sağlayan ağ sistemi hücresel ağlardır. Hücresel ağlar, GSM teknolojilerinin insanların günlük hayatlarına girmesi ile daha da gelişme imkanı bulmuştur. GSM teknolojileri ifadesi ile ilk olarak akla kullanılan cep telefonları gelmektedir. Başlarda hücresel ağlar üzerinde sadece ses ve yazı verileri taşınırken, zamanla hücresel ağlar daha büyük boyutlarda veri taşıyabilir hale gelmiştir. Hücresel ağların gelişmesi ile, bu ağlar üzerinde sadece cep telefonları değil, farklı amaçlarla kullanılmakta olan mobil cihazlar da ortaya çıkmıştır. Araç takip sistemleri, mobil el terminalleri, video konferans sistemleri gibi cihazlara bakıldığında, bu cihazlar da hücresel ağlar vasıtası ile veri alış verişi yapabilmektedirler [9].

Bu çalışma kapsamında, yer istasyonu ile multikopter arasındaki veri alış verişi de GPRS/3G modül olarak adlandırılan, hücresel ağlara bağlanabilen ve bu ağlar üzerinde veri alış verişi yapabilen mobil bir cihaz ile gerçekleştirilmektedir. GPRS/3G modülü Şekil – 3.28’de görülebilmektedir.



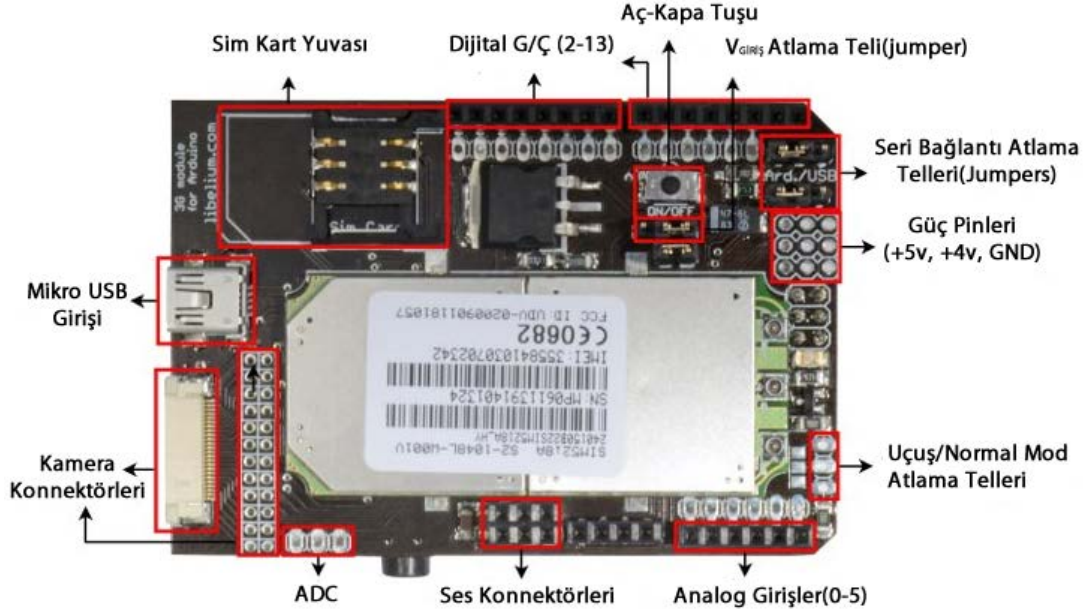
Şekil 3-28 GPRS/3G Modül

GPRS/3G modül üzerinde bir çok fonksiyon için kullanılmak üzere gerekli aygıtların bağlantısını sağlamak üzere giriş çıkış pinleri bulunmaktadır. Şekil – 3.29’ye bakıldığında GPRS/3G modül üzerine monte edilebilecek aygıtlar ayrıntılı olarak görülebilmektedir [18].



Şekil 3-29 GPRS/3G Modüle entegre edilebilen aygıtlar

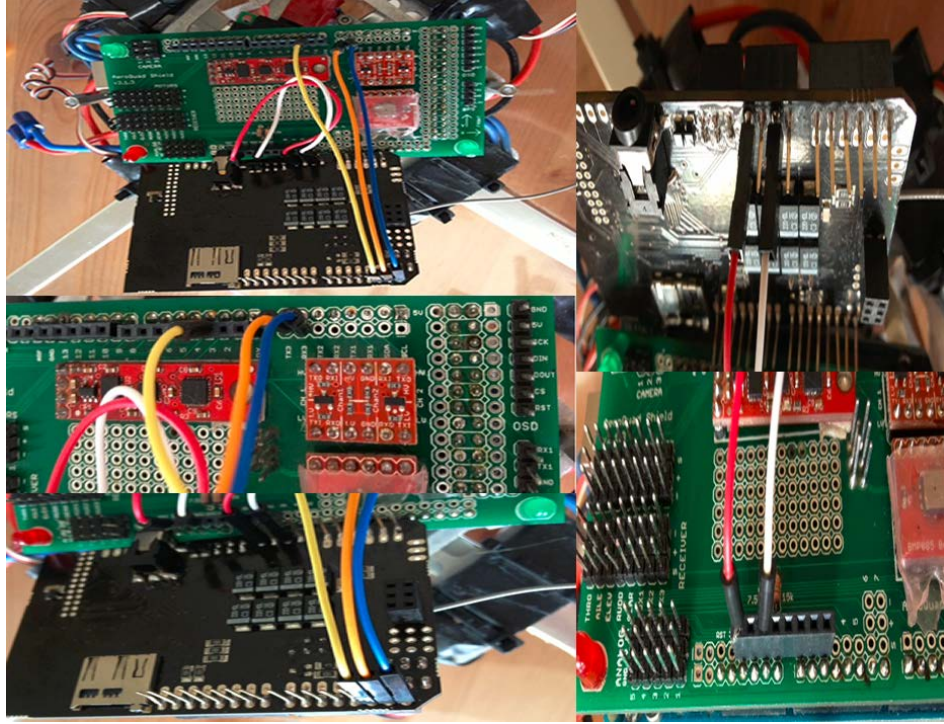
GPRS/3G modül ile üretici firma tarafından örnek olarak verilen uygulamaların dışında daha özel bir uygulama yapılmak istendiğinde, GPRS/3G modülünün daha ayrıntılı olarak incelenmesi gerekmektedir. Modül üzerindeki pin başları ve konnektörlerin hangi işlevler için kullanılabileceğinin öğrenilmesi gerekmektedir. Bu amaçla Şekil – 3.30’a bakıldığında genel hatları ile GPRS/3G modül üzerindeki pin ve konnektörlerin işlevleri anlaşılmaktadır [18].



Şekil 3-30 GPRS/3G Modüle üzerindeki pin dağılımı

Bu çalışma kapsamında GPRS/3G modül, uçuş kontrol kartı ile entegre edilmekte ve modülün yönetimi uçuş kontrol kartı vasıtası ile yapılmaktadır. Entegrasyon donanımsal olarak, uçuş kontrol kartı ve modüldeki SPI olarak bilinen seri bağlantı ara yüzleri, uçuş kontrol kartı ile modülün otomatik olarak açılma kapanmasını sağlayan dijital giriş/çıkış pinleri ve GPRS/3G modüle uçuş kontrol kartından güç sağlayan güç pinleri vasıtası ile sağlanmıştır. Donanımsal entegrasyon Şekil – 3.31’te görülebilmektedir. Burada kırmızı ve beyaz kablolar uçuş kontrol kartından GPRS/3G modüle güç sağlamaktadır. Mavi ve turuncu kablolar ile uçuş kontrol kartı ve GPRS/3G modül arasında seri bağlantı ara yüzü vasıtası ile iletişim kanalı kurulmuştur. Uçuş kontrol kartının seri gönderici pini turuncu kablo ile GPRS/3G modülün alıcı pinine bağlanmıştır. Mavi kablo ise uçuş kontrol kartının alıcı pini ile GPRS/3G modülün gönderici pini arasında köprü kurmaktadır. Sarı kablo ile, uçuş kontrol kartının 13 numaralı pini, GPRS/3G modülün 4 numaralı pinine bağlanmıştır. Bu bağlantı kanalı ile, uçuş kontrol kartından iki saniye boyunca sağlanan +5V gerilim ile GPRS/3G modülü otomatik olarak açılabilen ve kapanabilmektedir.





Şekil 3-31 GPRS/3G Modüle ve uçuş kontrol kartı arasındaki donanımsal entegrasyon bağlantıları

GPRS/3G modülü, AT komut seti olarak adlandırılan kütüphanede tanımlı işlevler kullanılarak yönetilmektedir. USB ara yüzü ile bir bilgisayara doğrudan seri olarak bağlanıp yönetilebildiği gibi, bu çalışmada olduğu gibi arduino programlama kartı üzerinden de yönetilebilmektedir. Bilgisayarla doğrudan bağlantılı olarak modem modunda yönetilmektedir. Arduino programlama kartı üzerinden yapılan bağlantı ise ağ geçidi modu olarak adlandırılmaktadır.

AT komut setinde tanımlı komutlar, GPRS/3G modüle seri bağlantı ara yüzü ile gönderilir ve komut sonucu oluşan yanıtlar yine aynı ara yüz ile elde edilir. AT komut setleri üç ana format ile kullanılmaktadır. Bunlar;

- Temel(Basic)
- S Parametre
- Gelişmiş(Extended)

Formatlarıdır [19].

Temel komut kümesi daha çok anlık olarak GPRS/3G modülde bulunan parametrik verilerin okunması için kullanılmaktadır. Kullanım formatı “**AT<x><n>**” şeklindedir. “x” komutun adını, “n” ise opsiyonel olarak komutun parametresini içermektedir. “n” kullanılmadığı zaman komut üzerinde varsayılan bir parametre varsa o kullanılmaktadır [19].

S parametreleri ise, sistemin kayıt tutucu (register) denilen belleğinde bulunan sabit parametrelere erişim için kullanılmaktadır. Bu parametre değerleri modül kapansa dahi bellekte sabit olarak kalmaktadır. Bu komut setinin kullanım formatı ise “**ATS<n>=<m>**” şeklindedir. “n” değeri set edilecek olan kayıt tutucunun indis değerini, “m” ise bu kayıt tutucuya atanacak değeri ifade etmektedir. “m” değeri opsiyonel olup, belirtilmediği durumlarda kayıt tutucuya varsayılan değer atanacaktır. Bu durumda kullanım formatı ise “**ATS<n>**” şeklinde olacaktır [19].

Gelişmiş komut kümesi ise modül üzerindeki ana işlevlerin yerine getirildiği komutları içermektedir. Bu komut kümesinde kullanılan komutların formatları, komut üzerinde gerçekleştirilen işleme göre aşağıdaki şekilde değişmektedir. Tablo – 3.4’te komut işlevlerine göre formatlarının nasıl değiştiği gösterilmektedir [19].

**Tablo 3-3 Gelişmiş AT komut kütüphanesi komut formatları kullanımı**

İşlev	Format	Açıklama
Test Komutu	AT+<NAME>=?	Böyle bir komutun varlığı test edilir. Varsa parametreleri hakkında bilgi alınır.
Okuma Komutu	AT+<NAME>?	Parametrenin anlık olarak değeri elde edilir.
Yazma Komutu	AT+<NAME>=<...>	Parametrenin değeri set edilir.
Yürütme Komutu	AT+<NAME>	Komut ile modülde bir işlev yürütülür.

Bu çalışma kapsamında GPRS/3G modülünün sahip olduğu AT kütüphanesinin TCP/IP protokolü ile ilgili işlevleri içeren komutları kullanılmıştır. Bu komutlar Tablo – 3.5’te işlev açıklamaları ile birlikte gösterilmektedir.

**Tablo 3-4 TCP/IP protokolünün kullanımı için gerekli AT komutları**

Komut	Açıklama
AT+CMGF=1	SMS mesaj formatı "text" mode olarak atanmaktadır.
AT+CGSOCKCONT=1,"IP","mgbs"	Modül mobil internet parametreleri girilmektedir.
AT+NETOPEN="TCP",80	Modülün internet bağlantı soketi açılmaktadır.
AT+SERVERSTART	Modül sunucu moduna alınmaktadır.
AT+CIPHEAD=0	Alınan verinin başlık datası çıkarılmaktadır.
AT+CIPSRIP=0	Alınan verinin büyüklük datası çıkarılmaktadır.
AT+IPADDR	Modülün internet ağındaki IP adresi öğrenilmektedir.

### 3.5 Bilgisayar Tabanlı Kumanda Panel Yazılımı

Multikopter sistemleri incelendiğinde, genel olarak sistemin uzaktan kumanda edilebilmesini sağlayan radyo frekans teknolojisi ve kablosuz ağ teknolojisinin olduğu görülmektedir. Radyo frekans teknolojisi, aslında bu tür sistemlerde geleneksel olarak kullanılmakta olan, halk arasında uzaktan kumanda olarak da kullanılan teknolojidir. Bu yöntem ile uzaktaki bir aygıtı kontrol edebilmek için, aygıtın üzerinde bulunan kontrol kartına bağlı bir radyo frekans sinyal alıcı ve uzaktan sinyal göndererek aygıtın kontrolünün sağlanacağı bir radyo frekans verici olması gerekmektedir.

Bu çalışma kapsamında, çalışma amacına uygun olacak şekilde düzenlenmiş olan multikopter sistemi, varsayılan olarak geleneksel yöntem olan radyo frekans

teknolojisi tabanlı alıcı-verici sistemi ile çalışmaktadır. Fakat çalışma kapsamında gerçekleştirilen düzenleme ile, multikopter hücresel ağ tabanlı alıcı-verici sistemi ile yönetilmesi sağlanmaktadır.

Multikopter'in uçuş kontrol kartına entegre edilmiş olan GPRS/ 3G modül multikoter sisteminin alıcı ünitesi olarak görev yapmaktadır.

Multikopter sisteminin, uzaktan yönetimini sağlamak için kullanılan verici, bilgisayar ortamında çalışan, internet tabanlı yönetim paneli uygulamasıdır. İnternet tabanlı yönetim paneli uygulaması, bilgisayar ortamında sanal bir verici olarak geliştirilmiştir. Bundan sonraki ifadelerde verici uygulaması yerine kumanda paneli kullanılmaktadır.

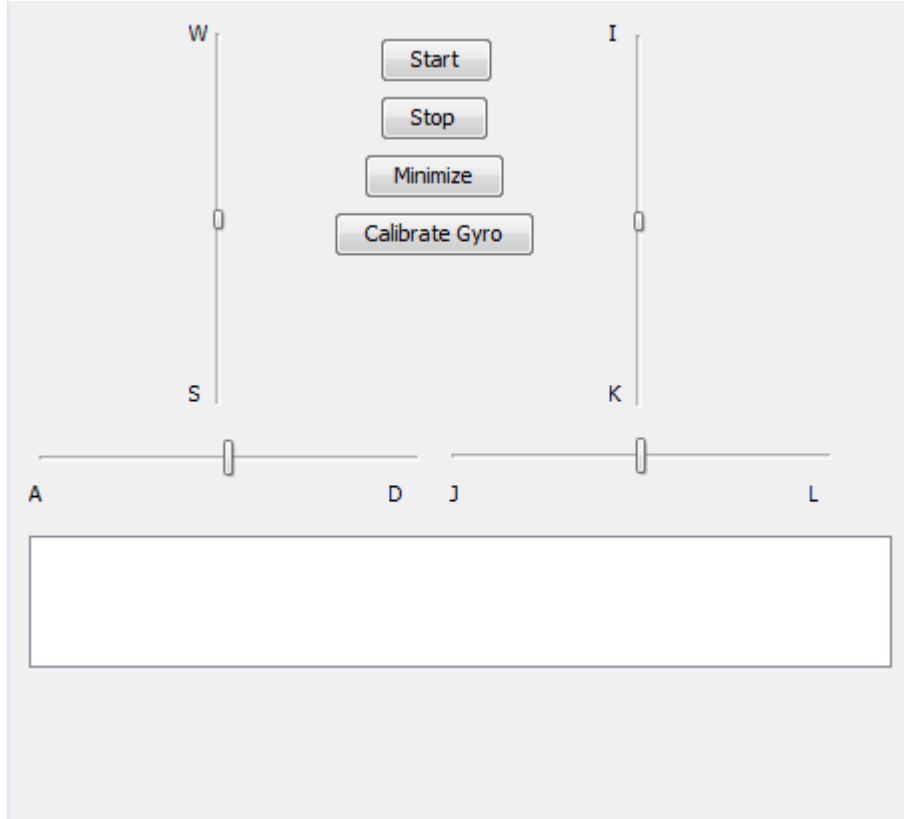
Kumanda paneli ve mikrokoopter üzerindeki GPRS/3G modül üzerlerinde bulunan ağ soketleri vasıtası ile iletişim bağlantı sağlamaktadırlar. Soketler internet iletişiminin uç noktalarıdır. İstemciler, istemci soketleri oluşturur ve onları sunucu soketlerine bağlar. Bu bağlantı üzerinden veri alıs-verisi gerçekleştirilir [7].

Kumanda paneli yazılımı Netbeans geliştirme ortamı kullanılarak JAVA programlama dili ile geliştirilmiştir. Şekil – 3.32'da kumanda paneli yazılımının, kullanıcı ara yüzü gösterilmektedir. Kumanda panelinin kullanılabilmesi için, öncelikle, uçuş kontrol kartına entegre edilen GPRS/3G modülünün aktif edilmesi ve bağlı olduğu hücresel ağ üzerinde gelen bağlantı taleplerini dinler konuma getirilmesi gerekmektedir. Aksi takdirde kumanda paneli yazılımı, başlangıç aşamasında GPRS/3G modül ile arasında sağlamak istediği bağlantıyı sağlayamamaktadır ve kullanıcı ara yüzü açılmadan yazılım bağlantı sağlanamadığına dair hata verip sona ermektedir. Bağlantı sağlandığında ise kullanıcı ara yüzü açılacak ve multikopter sistemini yönetecek kullanıcı, açılan bu ara yüzdeki işlevleri kullanarak multikopter sistemini yönetebilmektedir.

Kullanıcı ara yüzü bir JAVA applet yardımı ile internet ortamında da kullanılabilir hale getirilebilecek şekilde yeniden geliştirilebilir [10]. Appletler pek çok güvenlik sınırlamalarına sahiptir. Örneğin bir applet yerel dosya sistemine herhangi bir yoldan erişemez, bir yerel dosyayı okuyamaz, dosyaya yazamaz,

dosyalar hakkında herhangi bir bilgiye ulaşamaz, istemcideki diğer programları çalıştıramaz, kullanıcının makinesi hakkında bilgiye ulaşamaz, vs. Sadece indirildikleri sunucuya bir socket bağlantısı açabilirler.[5] Bazı özel durumlarda tüm bu güvenlik sınırlamaları bir Java Arşiv (JAR, Java Archive) dosyasına bir sayısal imza ilştirilerek kaldırılabilir [6].

Java.applet paketi Java Uygulama Programlama Arayüz'leri (API, Application Programming Interface) içindeki en küçük pakettir. Tüm appletler Java.applet paketi içinde yer alan applet sınıfının bir alt sınıfı olarak tanımlandıkları için java.applet paketini içe aktarmak (import) zorundadırlar [7].



Şekil 3-32 Kumanda paneli yazılımı kullanıcı ara yüzü

Kullanıcı ara yüzü vasıtası ile gerçekleştirilebilen işlemler Tablo – 3.6'da gösterilmektedir.

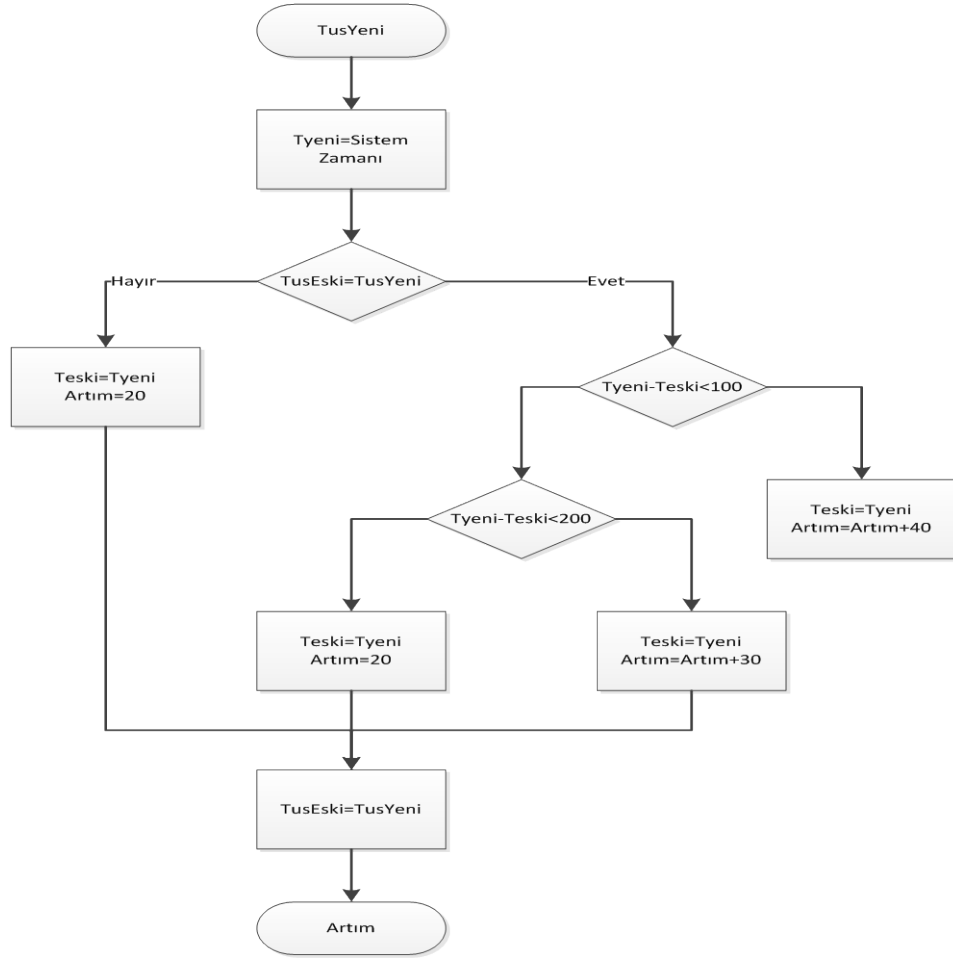
Tablo 3-5 Kullanıcı arayüzünde gerçekleştirilebilen işlemler

Giriş Aygıtı	Olay	Tuş	Açıklama
Klavye	Basıldığında	W	Sistemin throttle kanalına değeri arttırılır
Klavye	Basıldığında	S	Sistemin throttle kanalına değeri azaltılır
Klavye	Basıldığında	D	Sistemin Z boyutundaki kanal değeri arttırılır
Klavye	Basıldığında	A	Sistemin Z boyutundaki kanal değeri azaltılır
Klavye	Basıldığında	I	Sistemin Y boyutundaki kanal değeri arttırılır
Klavye	Basıldığında	K	Sistemin Y boyutundaki kanal değeri azaltılır
Klavye	Basıldığında	L	Sistemin X boyutundaki kanal değeri arttırılır
Klavye	Basıldığında	J	Sistemin X boyutundaki kanal değeri azaltılır
Fare	Sol Tuş ile Tıklandığında	Start	Motorlar minimum güç ile çalışır konuma getirir
Fare	Sol Tuş ile Tıklandığında	Stop	Motorlar kapatılır
Fare	Sol Tuş ile Tıklandığında	Minimize	Motorlara verilen güç, motorlar çalışır konumda ise minimum değere getirilir
Fare	Sol Tuş ile Tıklandığında	Calibrate Gyro	Gyro sensörünün kalibrasyonu sağlanır

Kumanda paneli yazılımı geliştirilirken göz önüne alınması gereken en önemli kriterlerden bir tanesi, geliştirilen kumanda panelinin hareket kabiliyeti ile yönetilecek aygıtın hareket kabiliyetidir. Multikopter gibi oldukça ani ve geniş hareket kabiliyeti olan bir sistem için, sistemin bu kabiliyetine maksimum derece yakın hareket kabiliyeti olan bir kumanda yazılması gerekmektedir. Aksi halde kumanda paneli ile multikopter sistemine gönderilen komutlar ya çok gecikmeli üretilmiş olacak, ya da üretilen komut anlık olarak multikopter sisteminin hareketine etki etmeyecektir.

Bu çalışmada geliştirilen kumanda paneli, multikopter sisteminin ani ve geniş hareket kabiliyetine maksimum seviyede uyum sağlayabilmesi için, kullanıcının paneli kullanım şekline bağlı olarak veri üretebilecek şekilde tasarlanmış ve geliştirilmiştir. Dolayısı ile kullanıcı, kumanda paneli ile multikopter sisteminin motor gücünü ani olarak arttırmak ya da azaltmak istediği takdirde, kumanda paneli bu durumu kullanıcının klavyedeki “W/S” tuşuna basma frekansına bağlı olarak tespit edip, bu doğrultuda veri üretmektedir.

Kumanda panelinin, komut üretmek için kullandığı artım fonksiyonuna ait algoritma Şekil – 3.33’de gösterilmektedir. Buna göre kullanıcının aynı tuşa birden fazla bastığı durumlarda, iki basma hareketi arasında geçen süre hesaplanmakta ve bu süre 100 milisaniyeden küçük ise, artım için kullanılan değişkenin değeri 40 birim arttırılmaktadır. Aynı tuşa birden fazla basım arasında geçen süre 100 milisaniyeden büyük veya eşit, 200 milisaniyeden küçük ise artım değişkeni bir önceki değerine göre 30 birim arttırılmaktadır. Eğer aynı tuşa birden fazla basım arasında geçen süre 200 milisaniyeden büyük veya eşit ise artım değişkenine sabit olarak 20 birim değeri verilmektedir. Komut değerinin azaltılması gerektiği durumda ise, artım olarak hesaplanan değer, komut değerinin son değeri ile negatif olarak toplanmaktadır.



Şekil 3-33 Kumanda paneli artım fonksiyonu akış diyagramı

Çalışma kapsamında geliştirilen kumanda panel yazılımının işlevlerine ait program kodları şekil - 3.34'de gösterilmektedir. Yazılım çalıştırıldığı anda ilk olarak altı numaralı satırda yer alan KumandaPaneliYazilimi() prosedürünü yürütmektedir. Bu prosedür içerisinde dokuz ile on yedinci satırlar arasında kumanda paneli yazılımının çalıştırıldığı bilgisayardan multikoptere bağlı GPRS/3G modülüne doğru bağlantı kurulması sağlanmaktadır. Bağlantı kurulamaması durumunda gerekli mesaj verilerek program kapatılmaktadır. Bağlantı kurulduğu anda ise şekil 3-32'de görülen kumanda paneli ara yüzü açılmaktadır. Bu andan itibaren kumanda paneli yazılımı, kullanıcının klavyeden tablo 3-6'da görülebilen tuşlardan birisine basmasını beklemektedir. Yirmi altında satırda başlamakta olan jPanel1KeyPressed() prosedürü kullanıcının klavyeden girmiş olduğu tuşun hangisi olduğuna karar vermekte ve girilmiş olan bu tuşa bağlı olarak GPRS/3G modülüne gönderilecek olan komut değerlerini elde etmektedir. Yüz on numaralı satırda başlayan sendCommand() prosedürü ise GPRS/3G modülüne gönderilmek üzere elde edilen komut değerlerini kullanarak, modüle gönderilecek katar dizisini oluşturmakta ve GPRS/3G modüle iletmektedir.



```

1 public class KumandaPaneliYazilimi extends javax.swing.JFrame {
2     private TransmitterCommand Comm;
3     private Socket socket = null; private PrintWriter out = null;
4     private String command; long old_time=System.currentTimeMillis();
5     private int artim=10; char key_old; BlankArea blankArea; JTextArea textArea;
6     public KumandaPaneliYazilimi () {
7         initComponents();
8         Comm=new TransmitterCommand(); jPanel1.setFocusable(true);
9         try {
10            socket = new Socket("5.11.128.194", 80);
11            out = new PrintWriter(socket.getOutputStream(), true); }
12            catch (UnknownHostException e) {
13                System.err.println("Don't know about host");
14                System.exit(1); }
15            catch (IOException e) {
16                System.err.println(e.getMessage());
17                System.exit(1); }
18            command="";
19        }
20        //Her tuşa basıldığında komuta etki edecek artım değeri hesabı
21        private int ArtimHesapla(char key_now)
22        {
23            //Klavyeden basılan tuş dinlenmekte ve kullanıcı ara yüzünde
24            //belirtilen tuşlardan birisine basıldığında yakalanmakta ve
25            //o tuşa atanan komut üzerinde gerekli işlem yapılmaktadır.
26            private void jPanel1KeyPressed(java.awt.event.KeyEvent evt) {
27                char key=Character.toUpperCase(evt.getKeyChar());
28                TransmitterCommand.CommandType ct = null;
29                //Klavyeden basılan tuş W ise;
30                if (key=='W') {
31                    jPanel1Slider1.setValue(jPanel1Slider1.getValue()+ArtimHesapla(key));

```

```

32 ..... ct=TransmitterCommand.CommandType.THROTTLE;
33 ..... Comm.ReceiveCommand(ct, jSlider1.getValue());
34 ..... try { sendCommand(); }
35 ..... catch (IOException ex) {
36 .....     Logger.getLogger(KumandaPaneliYazilimi.class.getName(
37 .....     )).log(Level.SEVERE, null, ex); } }
38 ..... //Klavyeden basýlan tuþ S ise;
39 ..... if (key=='S')
40 .....     //Klavyeden basýlan tuþ A ise;
41 .....     if (key=='A')
42 .....         //Klavyeden basýlan tuþ D ise;
43 .....         if (key=='D')
44 .....             //Klavyeden basýlan tuþ K ise;
45 .....             if (key=='K')
46 .....                 //Klavyeden basýlan tuþ I ise;
47 .....                 if (key=='I')
48 .....                     //Klavyeden basýlan tuþ L ise;
49 .....                     if (key=='L')
50 .....                         //Klavyeden basýlan tuþ J ise;
51 .....                         if (key=='J')
52 .....                             }
53 .....
54 ..... //Motorlarýn çalyþtýrylmasý için manuel komut verilmektedir.
55 ..... private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) { .....
56 .....     jSlider1.setValue(1010); jSlider2.setValue(1990);
57 .....     Comm._Throttle=1010; Comm._Xaxis=1500; Comm._Yaxis=1500; Comm._Zaxis=1990;
58 .....     jCommandArea.setText(Integer.toString(Comm._Throttle)+" "+
59 .....     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
60 .....     » Integer.toString(Comm._Zaxis));
61 .....     command="@"+Integer.toString(Comm._Throttle)+" "+
62 .....     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
63 .....     » Integer.toString(Comm._Zaxis)+"!";
64 .....     for(int i=0;i<5;i++) { out.println(command); }
65 .....     jPanell.requestFocusInWindow(); jSlider2.setValue(1500); Comm._Zaxis=1500; } .....
66 .....
67 ..... //Motorlarýn durdurulmasý için manuel komut verilmektedir. ....
68 ..... private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) { .....
69 .....     jSlider1.setValue(1010); jSlider2.setValue(1010);
70 .....     Comm._Throttle=1010; Comm._Xaxis=1500; Comm._Yaxis=1500; Comm._Zaxis=1010;
71 .....     jCommandArea.setText(Integer.toString(Comm._Throttle)+" "+
72 .....     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
73 .....     » Integer.toString(Comm._Zaxis));
74 .....     command="@"+Integer.toString(Comm._Throttle)+" "+
75 .....     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
76 .....     » Integer.toString(Comm._Zaxis)+"!";
77 .....     for(int i=0;i<5;i++) { out.println(command); }
78 .....     jPanell.requestFocusInWindow(); jSlider1.setValue(1500); jSlider2.setValue(1500);
79 .....     Comm._Zaxis=1500; Comm._Throttle=1500; } .....
80 .....
81 ..... //Motorlarýn minimum güçle çalyþmasý için manuel komut verilmektedir.
82 ..... private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) { .....
83 .....     jSlider1.setValue(1200); jSlider2.setValue(1500); jSlider3.setValue(1500);
84 .....     jSlider4.setValue(1500); Comm._Throttle=1200;
85 .....     Comm._Xaxis=1500; Comm._Yaxis=1500; Comm._Zaxis=1500;
86 .....     jCommandArea.setText(Integer.toString(Comm._Throttle)+" "+
87 .....     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
88 .....     » Integer.toString(Comm._Zaxis));
89 .....     command="@"+Integer.toString(Comm._Throttle)+" "+
90 .....     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
91 .....     » Integer.toString(Comm._Zaxis)+"!";
92 .....     for(int i=0;i<5;i++) { out.println(command); }
93 .....     jPanell.requestFocusInWindow(); } .....

```

```

94
95 --- //Gyro kalibrasyonu için manuel komut verilmektedir. ---
96 --- private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) { ---
97 ---     jSlider2.setValue(1010); jSlider4.setValue(1990); Comm._Throttle=1500;
98 ---     Comm._Xaxis=1990; Comm._Yaxis=1500; Comm._Zaxis=1010;
99 ---     jCommandArea.setText(Integer.toString(Comm._Throttle)+" "+
100 ---     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
101 ---     » Integer.toString(Comm._Zaxis));
102 ---     command="@"+Integer.toString(Comm._Throttle)+" "+
103 ---     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
104 ---     » Integer.toString(Comm._Zaxis)+"!";
105 ---     for(int i=0;i<15;i++) { out.println(command); }
106 ---     jPanel1.requestFocusInWindow(); jSlider2.setValue(1500);
107 ---     jSlider4.setValue(1500); Comm._Zaxis=1500; Comm._Xaxis=1500; } ---
108
109 --- //Üretilen komutun GPRS/3G modüle gönderilmesi tetiklenmektedir. ---
110 --- private void sendCommand() throws IOException {
111 ---     jCommandArea.setText(Integer.toString(Comm._Throttle)+" "+
112 ---     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
113 ---     » Integer.toString(Comm._Zaxis));
114 ---     command="@"+Integer.toString(Comm._Throttle)+" "+
115 ---     » Integer.toString(Comm._Xaxis)+" "+Integer.toString(Comm._Yaxis)+" "+
116 ---     » Integer.toString(Comm._Zaxis)+"!";
117 ---     out.println(command);
118 }

```

Şekil 3-34 Kumanda paneli yazılımı

#### 4 SONUÇ

Bu tez çalışması kapsamında yürütülmüş olan araştırmalar ve uygulamalar sayesinde, hücresel ağlar vasıtası bir mobil cihaz ile iletişim kurulup, bu mobil cihaza komut gönderilebilmiş ve bu gönderilen komutlarla mobil cihaz uzaktan kontrol etme imkanı sağlanmıştır.

Mobil cihaz ile uzaktan kurulan iletişimin hücresel ağ üzerinden gerçekleştirilmesi, radyo frekansı ile kurulan iletişime göre mobil cihazın çok daha geniş bir alanda yönetilebilmesini sağlamaktadır. Çalışma esnasında mobil cihaz olarak kullanılan multikopter sistemi, düzenleme öncesinde kullanılan radyo frekans temelli uzaktan yönetim sistemi ile maksimum 1500 metre yarı çapında bir alan içerisinde hareket edebilme yeteneğine sahipken, hücresel ağ temelinde gerçekleştirilen uzaktan yönetim sistemi ile hücresel ağlara erişim sağlanabildiği sürece hareket edebilme yeteneği kazandırılmıştır.

Bu çalışma sonrasında, sadece komut gönderilebilen değil, aynı zamanda verilen bir GPS noktasına kendi kendine gidebilen, geçtiği noktalarda sıcaklık, nem, görüntü vb. bilgileri yer istasyonuna iletebilen birçok mobil cihaz uygulamalarının geliştirilebilmesi için de temel oluşturulmuş olmaktadır. Bu çalışmada hücresel ağlar üzerinden iletilen komut verileri multikopterin kontrolü için önemli olmakla birlikte, aslında sembolik nitelik taşımaktadır. Burada hücresel ağ üzerinden mobil bir aygıt ile veri alış verişi yapılmak üzere iletişim kurulabileceği gösterilmektedir.

## 5 KAYNAKÇA

1. Sema Y., TC65 GSM/GPSR Modülü Kullanılarak Uzaktan Kontrol Sistemi Gerçekleştirilmesi, Selçuk Üniversitesi, 2008
2. Muammer Kadir K., İnternet Tabanlı Uzaktan Denetim için Bir Ara yüz Tasarımı, Muğla Üniversitesi, 2008
3. Kudret D., PLC ile Kontrol Edilen Endüstriyel Bir Sistemin Cep Telefonu ile Kontrolü, Gazi Üniversitesi 2010
4. Javorski, J., Java 1.1 Developer's Guide Second Edition, Sams.net Publishing, Indianapolis 1997
5. David R., Java Network Programming FAQ, 2000
6. Flanagan, D., Java in a Nutshell a Desktop Quick Reference Second Edition, O'REILLY Associates Inc., Sebastopol 1997
7. Schildt, H., Herkes için Java 2, Alfa Basım Yayım Dağıtım Ltd. Sti., İstanbul 2003
8. Stuckmann, P.; Ehlers, N.; Wouters, B., GPRS traffic performance measurements, Vehicular Technology Conference, Aachen 2002.
9. Ünlü, B., İnternet Üzerinden Mobil Bir Robotun Kontrolü, Lisans Tezi, Yıldız Teknik Üniversitesi Elektrik-Elektronik Fakültesi, İstanbul 2007.
10. Sun Microsystems, Java Tutorials, 2007
11. Siemens Industry Automation and Drive Technologies Service&Support, Micromaster 420 0,12 kW - 11 kW
12. <https://www.tai.com.tr/tr/proje/anka> (06.07.2013).
13. <http://www.nokia.com.tr/id4170.html> (06.07.2013)
14. <http://aeroquad.com/showwiki.php> (06.07.2013)
15. <http://www.arduino.cc/> (06.07.2013)
16. <http://aeroquad.com/showthread.php?1563-AeroFPV-Frame/page12>  
(13.05.2013)
17. <http://www.bphobbies.com/view.asp?id=V450327&pid=B2632605>  
(06.07.2013)

18. <http://www.cooking-hacks.com/index.php/documentation/tutorials/arduino-3g-gprs-gsm-gps> (06.07.2013)
19. [http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/SIM5218\\_AT\\_command\\_manual.pdf](http://www.cooking-hacks.com/skin/frontend/default/cooking/pdf/SIM5218_AT_command_manual.pdf) (06.07.2013)

## ÖZGEÇMİŞ

Nazır AŞIKKUTLU, 1985 yılında Bursa’da doğdu. Öğrenimlerini sırasıyla Emirbuhari İlkokulu ve Kartal Anadolu İmam Hatip Lisesi’nde tamamladı. 2003 yılında Maltepe Üniversitesi Mühendislik – Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünü kazandı ve 2007 yılında bölüm birincisi olarak mezun oldu. Sırası ile, Bizpro Sistem’de Mobil Uygulama Geliştirme Uzmanı, Maltepe Üniversitesi’nde Uygulama Geliştirme Uzmanı, Aegon Emeklilik ve Hayat’ta Analist Yazılımcı, Avea İletişim Hizmetleri’nde Uygulama Operasyon Uzmanı olarak görev aldı. Şubat 2010’da, Maltepe Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans programında yüksek lisans öğrenimi yapmaya hak kazandı. Halen İnnova Bilişim Çözümleri’nde SAP Danışmanı olarak görev yapmaktadır.