

İçindekiler

1. ÖZET	1
2. GİRİŞ	2
3. QUADCOPTER PARÇALARI	4
3.1. LİTYUM - POLİMER PİLLER	4
3.2. FIRÇASIZ MOTORLAR.....	5
3.2.1. İNLINE FIRÇASIZ MOTORLAR.....	5
3.2.2. OUTLINE FIRÇASIZ MOTORLAR.....	6
3.3. ELEKTRONİK HIZ KONTROL ÜNİTESİ (ESC).....	7
3.4. BEAGLEBONE BLACK.....	9
3.5. MUTLAK ORYANTASYON SENSÖRÜ.....	11
3.6. WIRELESS ADAPTÖRÜ	12
3.7. ŞASE.....	12
3.8. PERVANELER	13
4. PROJENİN GERÇEKLEŞTİRİLMESİ.....	13
4.1. YAZILIM DİLİNİN BELİRLENMESİ.....	13
4.2. KONTROL ALGORİTMASININ BELİRLENMESİ.....	14
4.3. BAĞLANTI PROTOKOLÜ.....	15
4.4. BASKI DEVRE HAZIRLANMASI	16
4.5. KUMANDA YAZILIMI	17
4.6. KONUMLAMA SİSTEMİ	19
6. KAYNAKLAR	25

1. ÖZET

İlk dönem bitirme projesinde, quadcopter gömülü yazılımı hazırlamıştık. Hazırlamış olduğumuz yazılım açık kaynak kodlu, Multiwii isimli yazılım ile arduino platformu üzerinde hazırlanmıştı. Açık kaynak kodlu yazılım üzerinde ek geliştirmeler yapılmıştı.

Bu dönem yapmış olduğumuz projede, arduino geliştirme kartını değiştirip, daha güçlü işlemciye sahip bir platform üzerinde quadcopterin uçuşunu gerçekleştirmek amaçlanmıştır. Bu amaçla Beaglebone Black isimli Single Board Computer tercih edilmiştir. Platformun değişmesiyle birlikte, quadcopter için uçuş yazılımının yeniden tasarlanması ve baştan, yeni tasarıma göre kodlanması yapılmıştır. Yeni tasarlanan sisteme göre, Android platformunda çalışan kumanda uygulaması geliştirilmiştir.

Quadcopter ile kumanda arasındaki iletişim için internet ağı üzerinde ve TCP/IP Socket protokolü kullanılmıştır.

Quadcopter yazılımının hazır olmasıyla birlikte, otopilot sistemlerinde kullanılabilmesi için konumlama sistemi geliştirilmiştir. Bu sistem iki adet kameradan oluşmaktadır. Her kameradan sadece iki boyutlu konum verisi elde edilebilmektedir. İki kameradan elde edilen iki boyutlu konum verisi birleştirilerek üç boyutlu konum verisi elde edilmektedir.

2. GİRİŞ



ŞEKİL 2.1: Quadcopter

İnsansız hava araçları önemini giderek arttırmakta ve daha çok alanda adı geçmektedir. Üzerlerine düşen ve yapması beklenen görevlerin başarısı en iyi performansı sunmasına bağlıdır. Bu bağlamda hava aracı tasarımında insansız hava araçlarının yeri giderek artmaktadır [4].

Döner kanat dikey iniş kalkış yapabilen, havada asılı kalabilen, manevra kabiliyeti yüksek, kontrol sistemi karışık olmasına rağmen yapısal olarak basit 4 rotorlu, motorların ürettiği tahrik kuvvetinden yararlanarak pervaneleri vasıtasıyla taşıma kuvveti oluşturan döner kanatlı insansız hava aracıdır [8].

İnsansız Hava Araçları (İHA) üzerine yapılan çalışmalar gün geçtikçe artmaktadır. Bu çalışmalara günümüzde yaşanan teknolojik gelişmelerin katkısı ve İHA'ların yapımında kullanılan malzemelerin temin edilmesindeki kolaylıklar gibi durumların katkısı büyüktür [1].

İHA, özellikle askeri uygulamalar başta olmak üzere doğal afetler, çeşitli spor faaliyetlerinin izlenmesi, trafik denetimi, kaçak yapılaşma vb. birçok durumun çözüm veya analizinin hızlı ve güvenli bir şekilde gerçekleştirilmesine olanak sağlamaktadır [5].

İnsansız hava aracı konulu çalışmalarda, yeni tasarım yaklaşımları, gelişen analitik ve sayısal yöntemler, gelişen işlemci ve elektronik altyapı imkânları, artık kabul görmüş kompozit malzeme kullanımı çokça işlenmektedir. Bunlara ek olarak ve belki de en önemlisi tasarımda insana yönelik kısıtlamaların mevcut bulunmaması bu tasarımların çok yırtıcı bir şekilde ilerlemesine ve yakın zamanda hava araçları içerisinde azımsanmayacak sayılarda yer bulunmasına zemin hazırlamaktadır [4].

Yakın geçmişe bakılacak olursa İHA'ların genel olarak askeri amaçlar haricinde çok fazla kullanılmadığı görülmektedir. Teknolojinin son 20 yıldaki hızlı gelişimi ile birlikte farklı özellik ve tipte İHA'ların popülerliği ve kullanım alanları artmıştır [6].

İnsansız hava araçları kullanım alanlarına göre farklı biçimlerde üretilmektedir. İHA'ların en çok tercih edilen uygulamalarından biri, quadrotor olarak bilinen dört rotorlu insansız hava aracıdır. Quadrotor, isminden de anlaşılacağı üzere birbirinden bağımsız dört adet rotora sahip, çok hızlı manevra yapabilen insansız hava araçlarına verilmiş genel bir isimdir. Quadrotor'ların rotor sayısı arttırılarak elde edilmiş farklı formları mevcuttur. Altı motorlu hexakopter ve sekiz motorlu oktokopter bu İHA'lara örnek olarak verilebilir [5].

Kullanım alanları ve amaçlarının artmasıyla birlikte, özellikle insan faktörünün sınırlarını zorlayacak esneklikte ve riskli görevlerde kullanılabilmesi için, oto pilot sistemleri geliştirme çabaları da artmaktadır. Özellikle dört rotorlu insansız hava araçları (quadrotors) diğer insansız hava araçlarına göre sahip oldukları avantajları nedeniyle; (agresif manevra, bulunduğu yerden dikey olarak iniş ve kalkış yapabilme, havada asılı kalma, 4 rotor sayesinde fazla yük taşıyabilme, vb.) sıkça tercih edilmektedirler [7].

İHA'lar RF (Radyo Frekansı) haberleşmesine sahip bir kumanda yardımıyla manuel olarak kontrol edilebildiği gibi üzerlerinde taşıdıkları mikrodenetleyici, sensörler ve gömülü yazılımlar sayesinde görev tanımlı uçuş yeteneğine sahip olarak da kullanılabilir [3]. İHA'ların görev tanımlı uçuş yeteneği kazanması ve üzerinde bulunan malzemelerin maliyet değerinin azalması ile birlikte bu tip araçlar, olumsuz hava şartlarında arama ve kurtarma çalışmalarının yapılması, sinema, televizyon ve fotoğrafçılık sektörü, zirai ilaçlama çalışmaları, kaçak yapılaşma, sınır gözetleme gibi pek çok farklı alanda kullanılmaya başlanmıştır [6].

3. QUADCOPTER PARÇALARI

3.1. Lityum - Polimer Piller

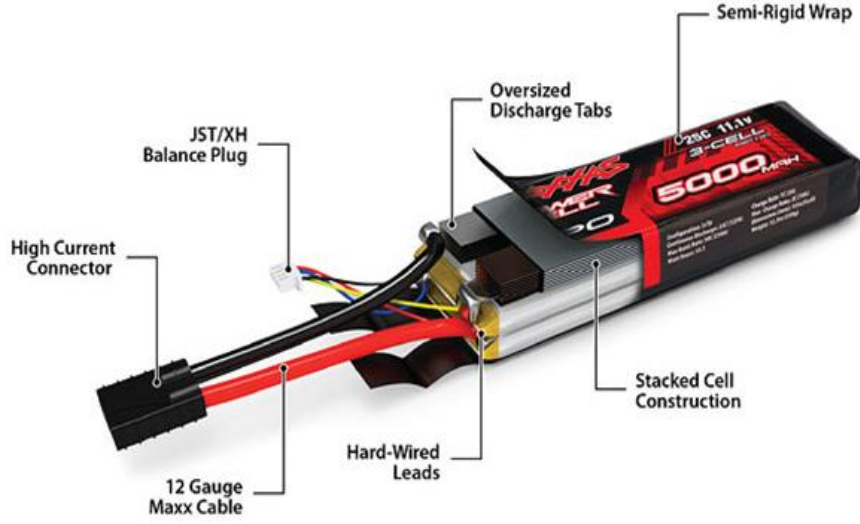
Lipo piller (lityum-polimer) helikopter ve uçaklarda kullanılan şarj edilebilir pillerdir. Lipo pillerin önemli özellikleri:

- Lipo bataryalar yüksek deşarj hızına sahiptirler ve günümüzde enerjiyi çok harcayan elektrik motorlarını besleyebilirler.
- Lipo bataryalar hafiftir ve istenen her şekilde boyutta üretilebilirler.
- Lipo bataryalar yüksek kapasiteye sahiptir. (NiCad ve NiMH bataryalara göre)

Quadrotorlar da kullanılan fırçasız motorların devir hızları 10000 rpm civarlarında olmaktadır. Yüksek rpm sebebiyle fırçasız motorlar yüksek akım veren ve yüksek kapasiteye sahip pillere ihtiyaç duymaktadırlar. Lipo piller yüksek akım ve yüksek kapasite ihtiyacını en yüksek verimde karşılayan besleme kaynaklarıdır. Lipo piller küçük hücrelerden oluşmaktadırlar. Her bir hücrenin nominal gerilim değeri 3,7V olmakta olup, amper seviyesinde yüksek akım ihtiyacını karşılayabilmektedirler. Bu özellikleri ile hiçbir pil ile karşılaştırılamayacak performansı gösterebilmektedirler.

Projemizde pil değerimizin 11.1v olacağını belirlemiştik. Burada pilin üzerinde yazan mAh değerine de dikkat edilmelidir. Bizim pilimiz 2800 mAh ve 30c deşarj kapasiteli olarak seçildi. Yani pilimiz anlık olarak $2800 \times 30 = 84000 \text{mA}$ akım sağlayabilir. Bizim motorlarımızın çektikleri akım hesaplanmıştı ve maximum 48A olarak bulunmuştu. Pilimizin bu değerden düşük olmaması gerekir. Aksi takdirde motorlar tam verimle çalışamaz. Pilin anlık vereceği amper değeri ne kadar yüksek ise aracın havada kalma süresi artmış olacaktır.

Pilimizi şarj edebilmek için özel dengeleyici şarj aleti kullanmak gerekir. Pilimiz 3 hücreli olduğu için şarj aletini de 3 hücreli şarj özelliği olan ImaxRC B3 Compact Charger kullanılıyor. Bu alet 3 hücrenin de dengeli bir şekilde dolumunu sağlıyor. Aksi halde pil ömrü kısılır.



ŞEKİL 3.1: LiPo Batarya

3.2. Fırçasız Motorlar

Fırçasız motorlar gelişen teknolojiyle beraber birçok alanda kullanılmakta olup, robotik alanda da kullanılmaya başlamıştır. Bu motorlar radyo kontrollü projelerde (Multikopterler, Helikopterler, RC Arabalar vb.) sıklıkla kullanılmaktadırlar. Bu alanda kullanılmalarının ana sebeplerinden biri diğer motorlara göre yüksek performans sergilemeleridir. Fırçasız motorların diğer motorlara göre daha sessiz çalışma, elektriksel gürültü oluşturmama, daha kolay bakım, daha uzun ömür, daha hızlı çalışma ve daha güçlü torklara sahip olma gibi avantajlarının yanı sıra; daha pahalı sistemler, kompleks kontrol ve yüksek performanstan dolayı oluşan ısınma gibi dezavantajları da vardır.

Fırçasız motorlar iç yapılarına göre inline fırçasız motorlar ve outline fırçasız motorlar olmak üzere ikiye ayrılırlar:

3.2.1. Inline Fırçasız Motorlar

Inline fırçasız motorun rotoru iç kısımdadır ve stator kısmı sabittir. Görünüş olarak standart fırçalı motorlara benzemekle beraber hızları daha yüksek ama voltaj başına torkları daha azdır.



ŞEKİL 3.2: Inline Fırçasız Motor

3.2.2. Outline Fırçasız Motorlar

Outline fırçasız motorların rotor kısmı dış tarafta, stator kısmı iç taraftadır. Stator kısmı sabit, rotor kısmı ise hareketlidir. Outline motorların özellikleri Inline motorların özelliklerinin tam tersidir. Hızları düşük fakat voltaj başına torkları daha yüksektir. Bu sebeple hava araçlarında bu motorlar kullanılmaktadır.

Projede de voltaj başına yüksek tork oluşturan bu outline fırçasız motorlardan kullanılmıştır. Çünkü quadrotorlar gibi küçük çaplı insansız hava araçları için en uygun motor cinsi outline fırçasız motorlarıdır.



ŞEKİL 3.3: Outline Fırçasız Motor

Bu noktada seçeceğimiz motorların kaldırmak istediğimiz ağırlığa ve pile göre seçmemiz gerekir. Kabaca anlatmak gerekirse motorumuz 1000kv değerindedir yani 1 volt değerinde voltaj verildiğinde dakikada 1000 devir dönmesi anlamına gelmektedir.

Kullandığımız pil 11.1 volt değerindedir. O halde motorumuz yüksüz durumda $11.1 \times 1000 = 11100$ devirde dönmektedir. Motorumuz maksimum çekeceği akım 12A/60s

olarak üretici tarafından belirlenmiştir. O halde 1 motorun harcadığı maksimum güç; $11.1\text{V} \cdot 12\text{A} = 133.2\text{ W}$ dır. Buda motorun çekme ağırlığının (thrust) 750 değeri ve maksimum 8400 devir yapacağı elde edilmiş olur.

Ancak bu hesap çok doğru değildir çünkü motor maksimum 12 amper harcar ama pratik olarak 11 amper civarındır.

Bu değerlerin hesaplanmasında pervane (propeller) boyutu da önemlidir. Bu değerleri tablo yaparak daha anlaşılır vaziyete getirirsek:

Motor sayısı	Pervane(inç)	Volt(V)	Amper(A)	Çekme ağırlığı(G)	Rpm/min Devir sayısı	Güç(w)
Tek motor	1047	11,1	11(max.)	730	9100	122,1
4 motor	1047	11,1	44(max.)	2920	9100	488,4

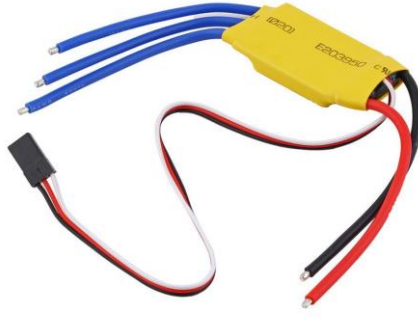
TABLO 3.1: Motor Parametreleri

3.3. Elektronik Hız Kontrol Ünitesi (ESC)

Tüm elektrikli RC modellerde kullanmak zorunda olduğumuz, fırçalı ve fırçasız motorların hızını ayarlayan kontrol ünitesidir. ESC'ler pilden aldığı enerjiyi, alıcının gaz kanalından aldığı sinyal ile sürerek motora hareket verir.

Fırçalı motorlar, DC motorlar olup iki uç ile ESC'den enerji alırlar. Bu sebeple fırçalı motorda kullanılan ESC'ler genel olarak daha basit yapıdadırlar. Pilden aldıkları iki girişleri ve motora gönderdikleri iki DC akım çıkışları mevcuttur. Motor uçları ters bağlanırsa, motor ters döner.

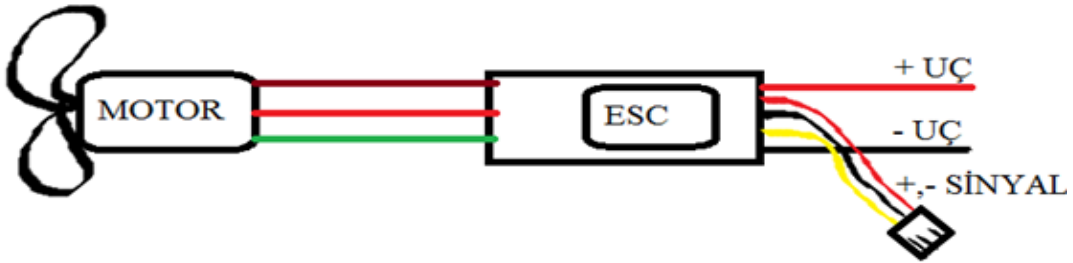
Multicopter sistemlerinde ise fırçasız motorlar kullanılır. “Tri Faze” olarak adlandırılan 3 fazlı motorlardır. Bu motorlarda kullanılan ESC'ler daha karmaşık yapıda olup, pilden aldığı DC gerilimi 3 kablo üzerinden AC akım olarak motora gönderirler.



ŞEKİL 3.4: Elektronik Hız Kontrolcüsü – ESC

Fırçasız ESC'ler birer frekans konvertörü gibi çalışarak değişik frekanslarda değişik devir elde etmemize yarar. Yani girişlerine uygulanan PWM (Darbe Genişlik Modülasyonlu İşaretler) sinyali ile fırçasız motorun hızını kontrol eder.

Kumandadan gelen dijital sinyalin darbe genişliği değiştirilerek, motora giden gerilim değeri dolaylı yoldan frekans kontrolü ile yapılır. Darbe genişliğini değiştirme işlemi ESC'nin çıkış akımını sürekli açıp kapatmasıyla gerçekleşir. Motora giden çıkış akımının ne kadar süre iletimde, ne kadar süre kesimde olduğu ile ilgilidir.



ŞEKİL 3.5: Fırçasız Motorlarda Kullanılan ESC

ESC'ler kontrol işlemini üzerlerinde bulunan mikrodenetleyici ile gerçekleştirirler. Bunun yanında ESC'ler, fırçasız motorun çekebileceği ani yüksek akımlara dayanmak zorundadır

Projemizde seçmiş olduğumuz ESC'ler 30A değerindedir. Kullandığımız motorlar ise maksimum 12 amper çekmektedir. Motorun akımı, ESC'nin akımını aşmadığı için ESC'ler daha az ısınır. Böylece ESC'nin ve motorun yanmaması sağlanmış olur.

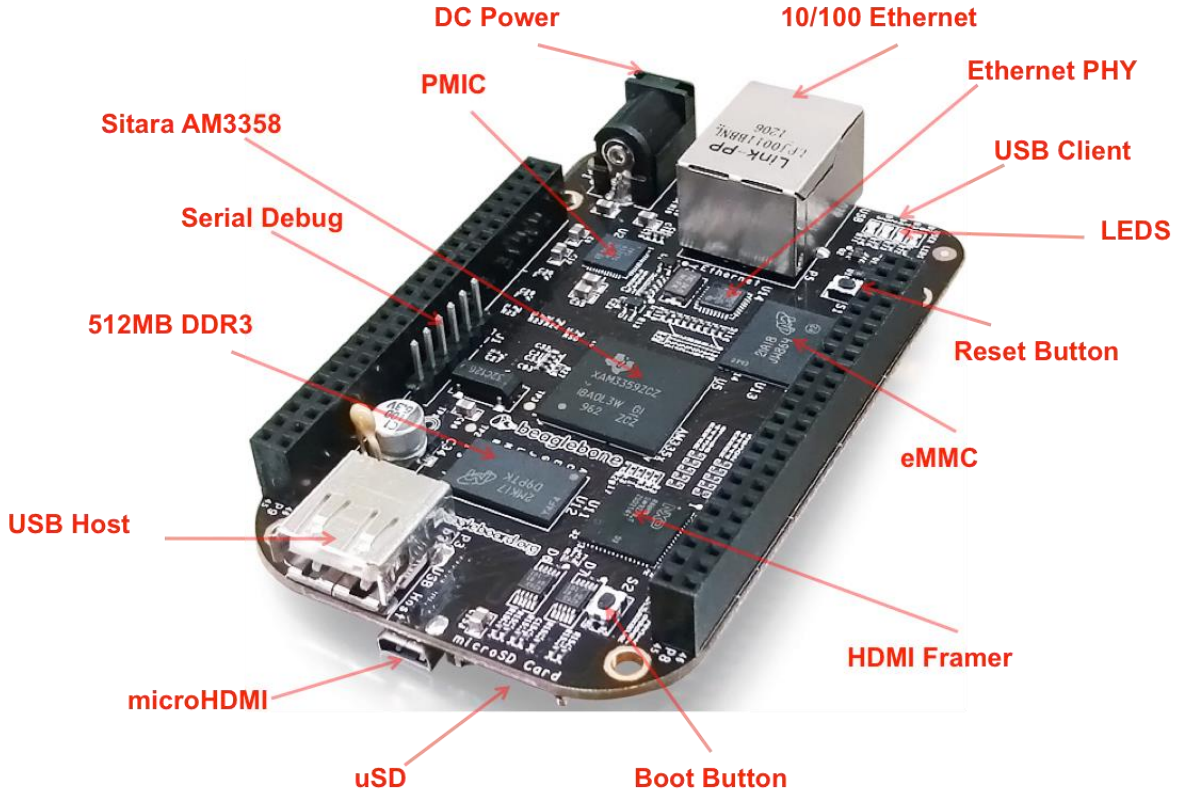
Akım	30 A
BEC	5V / 3A
BEC Modu	Doğrusal
Lipo Hücreleri	2 / 4
Ni-Mh	5 – 12
Ağırlık	27 gr

TABLO 3.2: Kullanılan ESC'nin Özellikleri

3.4. BeagleBone Black

BeagleBone Black , üstün özellikli bir mini-bilgisayar kartıdır. Ubuntu ve Android platformları üzerinde çalışabilir yapıdadır. Düşük fiyat ile yüksek performans sürebilen kart üzerinde Texas Ins. tarafından geliştirilen Sitara™ AM3358 ARM® Cortex™-A8 işlemcisi bulunmaktadır. Yüksek performansa sahip donanımlar ile çalışmak isteyen herkes için ideal bir karttır.

Kart üzerinde 1 GHz işlemci, 512 MB DDR3 dahili RAM, Linux Debian ön yüklemeli dahili 4 GB hafıza ve ses ve görüntü çıkışı için dahili µHDMI bağlantısı mevcuttur. DC 5V adaptör girişi, GPIO pinleri, microSD hafıza kartı girişi ve USB çıkışları ise BeagleBone'un kullanım kolaylığını arttıran özelliklerdendir.



ŞEKİL 3.6: BeagleBone Black

- İşlemci: Sitara AM3359AZCZ100 1GHz, 2000 MIPS
- Grafik Motoru: SGX530 3D
- SDRAM Bellek: 512MB DDR3L 606MHZ
- Dahili Flaş Hafıza: 4GB, 8bit Gömülü eMMC
- PMIC: TPS65217C PMIC regülatörü ve bir ek LDO.
- Opsiyonel Tümleşik 20-pin CTI JTAG, Seri Başlık: Destek hata ayıklama
- Güç Kaynağı: miniUSB veya DC Jack 5VDC Adaptör Girişi
- 88.98mm x 54.63mm x 18.84mm
- Göstergeler 1-Güç, 2-Ethernet, 4-Kullanıcı Kontrol LED
- Yüksek Hızlı USB 2.0
- 6 pin 3.3V TTL
- Ethernet 10/100, RJ45

- SD / MMC Bağlantı microSD, 3.3V
- Kullanıcı Girişi: Reset Düğmesi, Boot Düğmesi, Güç Düğmesi
- Video Çıkışı: 16b HDMI, 1280x1024 (MAX), 1024x768, 1280x720, 1440x900 w / EDID Destek
- HDMI arabirimi, Stereo Via Ses
- Tüm sinyalleri 3.3V I / O
- McASP0, SPI1, I2C, GPIO (65), LCD, GPMC, MMC1, MMC2, 7 AIN (1.8V MAX), 4 Zamanlayıcılar, 3 Seri Port, CAN0, EHRPWM (0,2), XDMA Kesme, Güç düğmesi, Genişleme
- Ağırlık : 39.80 gram

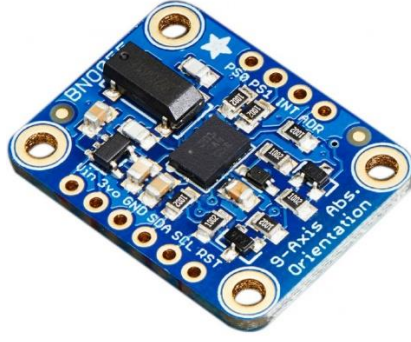
3.5. Mutlak Oryantasyon Sensörü

Mutlak oryantasyon bilgisi için BNO055 sensörü kullanılmıştır.

Bu 9 eksenli sensör sayesinde, objelerin üç boyutlu uzayda hangi konumda olduğu ekstra hesaplama yapılmadan öğrenilebilmektedir. Üzerinde bulunan ARM Cortex-M0 mikroişlemci sayesinde normal IMU kartı şeklinde kullanımın yanı sıra, cisimlerin mutlak konumlarını dördey veya Euler vektörü şeklinde elde edebilmeniz mümkündür. I²C ara yüzünü kullanır. 3.3V ve 5V lojik seviyeleri ile kullanılabilir.

Bu sensörü kullanarak elde edebileceğiniz bilgiler:

- Mutlak konum (100 Hz, Euler Vektörü yada dördey)
- Açısal hız vektörü (100 Hz, rad/sn)
- İvme vektörü (100 Hz, üç eksenli ivme, yerçekimi + doğrusal hareket, m/s²)
- Manyetik alan şiddeti vektörü (20 Hz, üç eksenli manyetik alan ölçümü, mikro Tesla uT)
- Doğrusal ivme vektörü (100 Hz, üç eksenli ivme vektörü, m/s²)
- Yerçekimi vektörü (100 Hz, üç eksenli yerçekimi vektörü, m/s²)
- Sıcaklık (1 Hz, °C)



ŞEKİL 3.7: BNO055 Sensörü

3.6. Wireless Adaptörü

Wireless adaptörü olarak TP-LINK markalı TL-WN822N modeli kullanılmıştır.

Quadcopterin kumanda ile komuta işlemleri web socket ile yapılacaktır. Bu işlemin yapılabilmesi için quadcopterin bir ağa bağlı olması gerekmektedir. Ağ bağlantısının yapılabilmesi için ilgili wireless adaptörü kullanılmıştır.

3.7. Şase

Şase, multicopter'in iskeletidir, genellikle karbon fiber ve alüminyum malzemeler tercih edilerek üretilmektedir. Şase tercihinde dikkat edilmesi gereken konular, devre koruması mevcut olup olmaması, iniş takımının yerden yüksekliği ve yeri kavraması, kolların titreşim oluşturmayacak şekilde üretilmiş olması, eksenlerde eğrilik ve kayıklık olmaması, batarya sabitlenecek kısmın belirli ve bataryanın kaymayacağı şekilde sabit olmasıdır.



ŞEKİL 3.8: Şase

3.8. Pervaneler

Motorlara önerilen pervane seçimi veriminizi yükseltecektir. Pervaneler plastik, tahta ve karbon fiber olarak üç ana sınıfta toplanabilir. Tahta ve karbon fiber pervaneler profesyonel sistemler için uygundur. Plastik pervanelerin sistemlerde kullanılması önerilmez.

4. PROJENİN GERÇEKLEŞTİRİLMESİ

Birinci dönem aynı proje üzerinde geliştirme yapılmıştır. Arduino Uno R3 geliştirme kartı ve açık kaynak kodlu Multiwii yazılımı ile quadcopter kontrol işlemi gerçekleştirilmiştir. Var olan açık kaynak kodlu yazılım üzerinde geliştirmeler yapılmış, ek özellikler eklenmiştir. Bu dönem projenin kapsamını geliştirerek yeni bir geliştirme kartı üzerinde tamamını kendimizin hazırlamış olduğu yerli gömülü quadcopter uçuş kontrol yazılımı geliştirilmesi planlanmıştır.

Bu kapsamda Arduino Uno'dan daha güçlü özelliklere sahip kartlar araştırılmıştır. Geliştiricilerin, çoğunlukla Raspberry Pi ve Beaglebone Black kartlarını kullandıkları görülmüştür. Bu kartlar özelleştirilmiş tek kart mini bilgisayarlar (Single Board Computer)'dır. Özellikleri kıyaslandığında Beaglebone Black kartının Raspberry Pi'den daha güçlü olduğu görülmüştür. Bununla birlikte Beaglebone Black kartının fiyatı Raspberry Pi'den daha yüksektir. Bu kart'ın tercih edilme sebebi projemizde ileride görüntü işleme ya da işlemci gerektirecek kısımların eklenme olasılığının olması ve bunlar eklendiğinde sistemde sorun oluşturmaması içindir. Verilen karar sonucunda Beaglebone Black kartı temin edilmiştir.

4.1. Yazılım Dilinin Belirlenmesi

Yazılım en baştan hazırlanacağı için doğru yazılım dilinin, protokollerin, algoritmaların ve kütüphanelerin belirlenmesi gerekmektedir. Yapılan araştırmalar ve deneyimler sonucunda pinlerin ve sensörlerin yönetiminin kolay olması açısından Python dili tercih edilmiştir. Kullanılan sensörler I²C protokolünü kullanmaktadırlar. Python dili ile bu protokolün yönetimi için hazır kütüphaneler bulunmaktadır. Kumanda iletişimi için TCP Web Socket protokolü kullanılması uygun görülmüştür. Hazırlanan yazılım içerisinde bir adet Socket Server oluşturulmuştur ve kumanda yazılımının bu server'a bağlanması sağlanmıştır.

4.2. Kontrol Algoritmasının Belirlenmesi

Quadcopter, sensörlerden aldığı durum bilgilerini kullanarak, içerisinde çalışan kontrol algoritması ile uçuş ve manevra kabiliyetlerine sahip olmaktadır. Bu noktada kontrol algoritmasının iyi seçilmesi ve gerçekleşmesi gerekmektedir.

Literatüre bakıldığında farklı amaçlar için kullanılmış birçok kontrol algoritması bulunduğu görülmektedir. Bu algoritmalarından en klasik olanı PID kontrol algoritmasıdır. PID kontrol algoritmasının birçok alanda kullanılmasının yanı sıra, bizim için önemli olan quadcopter sistemlerinde de yaygın olarak kullanıldığı görülmektedir. Bu sebeple proje kapsamında quadcopter içerisinde kullanılacak kontrol algoritmasının PID kontrol algoritması olması kararlaştırılmıştır.

PID kontrol sistemi, oransal, integral ve türev terimlerinin bir araya gelmesiyle oluşmuştur. PID kontrol basit yapısından dolayı endüstride sıkça kullanılan klasik bir kontrol yöntemidir. PID kontrol sisteminde ki hatayı; hatanın miktarı, hatadaki değişim hızı ve yapılan toplam hatayı hesaplayarak giderme yöntemidir. Bu değerlerin etkisini kontrol etmek için katsayılar kullanılmaktadır. P katsayısı sistemin hataya karşı vereceği tepkiyi belirler. Büyük olursa hatayı hızlı kapatır ama denge noktasını geçer. Bu etkiyi frenlemek için hatadaki değişim hesaplanır. Hatadaki değişimin etkisini de D katsayısı belirler. Toplam hata, sistemin geçmişte yaptığı hatayı telafi etmek için hesaplanır ve etkisi I katsayısıyla belirlenir.

PID kontrol sisteminde her eksen için P, I ve D olarak 3 adet parametrenin sisteme girilmesi gerekmektedir. Sistem bu değerleri kullanarak gerekli hesaplamaları yapar ve kendisini dengede tutar. Bu değerlerin hesaplanması sistemden sisteme değişmekte ve çoğunlukla parametre taraması ile bulunmaktadır.

Quadcopter üzerinde bir adet 9 eksenli sensör bulunmaktadır. Bu sensör sayesinde quadcopterin uçuş için ihtiyacı olan tüm bilgileri elde edilebilmektedir. Uçuş işleminin gerçekleşebilmesi için sensörden iki tür veri alınmaktadır. Bu verilerden biri quadcopterin üç boyutlu uzaydaki pozisyonunun bilinebilmesi için gereken euler yani roll, pitch ve yaw açılarıdır. Bu açılardan roll ve pitch açıları quadcopterin yer düzlemi ile yaptığı açığı vermektedir. Yaw açısı ise quadcopterin yönelimini yani manyetik kuzey ile yaptığı açığı vermektedir. Bu bilgiler, sensör üzerinden direkt olarak alınabilmekte, ek hesaplama yapmaya gerek kalmamaktadır. Sensörden alınan ikinci veri, gyro verisidir. Bu veri quadcopterin hareketindeki ivmelenmeleri vermektedir. Bu bilgi, quadcopterin yalpalama hareketlerini

yapmasını önlemek ve dışarıdan gelen ani etkilere tepki verebilmesi ve mevcut konumunu korumaya çalışması için kullanılmaktadır.

Sensörlerden her eksen için alınan iki farklı veri ayrı ayrı PID kontrol algoritması ile işleme sokulmakta ve sonrasında birleştirilerek tek bir sonuç elde edilmektedir. Bu sayede quadcopterin açı bozulmalarına tepki verebilmesinin yanında, ani durum değişikliklerine yani ivmelenmelere de tepki verebilmesi sağlanmış olmaktadır.

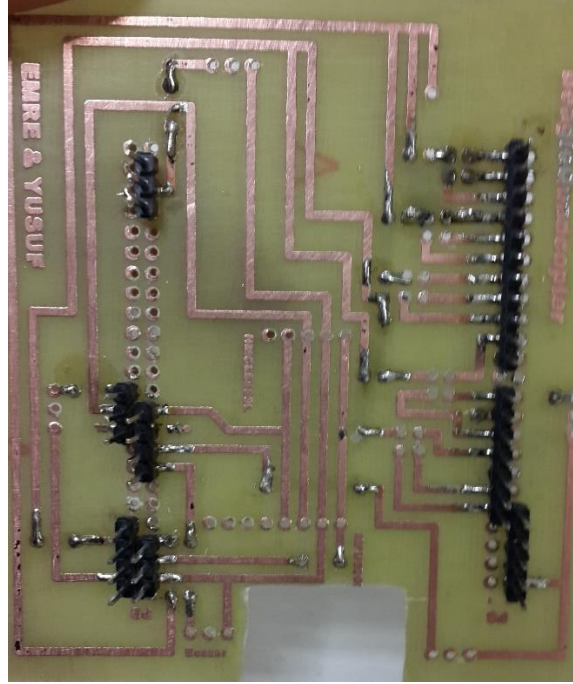
4.3. Bağlantı Protokolü

Quadcopterin, kumanda yazılımı ile iletişimi TCP Web Socket kullanılarak yapılması kararlaştırılmıştır. Bağlantının başlatılabilmesi için quadcopter içerisinde Socket Server başlatılmakta ve bağlantı beklemektedir. Kumanda yazılımı, bu Socket'e bağlanmakta ve kontrol işlemlerini bu bağlantı üzerinden gerçekleştirmektedir. Ayar değiştirme bağlantılarında herhangi bir problem olmamıştır. RC kontrol işleminin, yani quadcopteri uçarken komuta edebilmek için gereken bağlantıda iletişimin sürekli ve hızlı olması gerekmektedir. Kumanda üzerinde yapılan hareketlerin anında quadcoptere ulaşması gerekmektedir. Bu işlemin sağlanması için ilk denenen sistem, kumandanın sabit aralıklarla komuta verilerini quadcoptere yollaması olmuştur. Belirtilen bu aralık yaklaşık olarak 5 milisaniye olarak ayarlanmıştır. Sistem hazırlanıp test edildiğinde yapılan eylemlere tepki saniyelerce hatta dakikalarca sonra olmuştur. Bu gecikmenin nedeni socket üzerindeki yığılmadan kaynaklanmaktadır. Bandwidth'in düşük olması, quadcopterin veri alma hızının düşük olması gibi etkenler yüzünden sabit hızla gönderilen verilerde yığılma olmaktadır. Bunun sonucu olarak da quadcopter kumanda hareketlerine çok fazla gecikmeli tepki vermektedir.

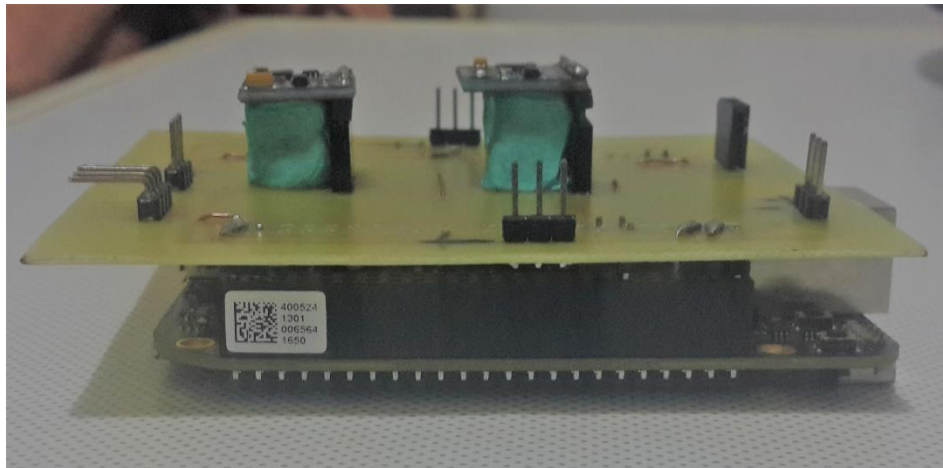
Bu soruna çözüm olarak yeni bir iletişim sistemi uygulanmıştır. Yeni sistemde kumanda yazılımı uçuş komuta işlemini başlattığı anda, quadcopter ve kumanda arasında karşılıklı bir döngü oluşturulmaktadır. Kumanda quadcoptere veriyi gönderdikten sonra quadcopterden yanıt beklemektedir. Quadcopterin yanıtı, gelen verilerin işlenmesi sonucu ortaya çıkan yeni kumanda verileridir. Kumanda yazılımı veriyi gönderdikten sonra bir sonraki veriyi yanıt geldikten sonra yapmaktadır. Bu şekilde yığılmanın önüne geçilmiş ve olabilecek en hızlı iletişim sağlanmış olmaktadır. Quadcopter aktif şekilde çalışırken bu iletişim döngüsünün süresi sürekli değişmekle birlikte yaklaşık olarak 20-30 milisaniye kadar olmaktadır.

4.4. Baskı Devre Hazırlanması

Sistem geliştirme süreci devam ederken, sensörleri sabitlemek, kablo kalabalığını engellemek, daha az yer kaplamak ve sistemi stabil hale getirmek için baskı devre yapılması düşünülmüştür. Proteus programı vasıtasıyla devre yolları çizimi tamamlanmış ve baskı devre birkaç hatalı deneme sonucunda başarıyla bakır levha üzerine aktarılmıştır.



ŞEKİL 4.1: Yapılan Baskı Devre



ŞEKİL 4.2: Baskı Devre ile Beaglebone Black'in Birleştirilip Sensörlerin Takılmış Hali

4.5. Kumanda Yazılımı

Quadcopter'in uzaktan kontrol edilebilmesi, kumanda edilebilmesi için Android işletim sistemli telefonlarda çalışacak bir uygulama geliştirilmiştir. Uygulama içerisinde şuan azami gereklilikteki ekranlar hazırlanmıştır. Bu bölümler;

- IP ve port'un yazılarak bağlantının sağlandığı login ekranı
- PID parametrelerinin ayarlanabildiği ayar ekranı
- Kalibrasyon işlemlerinin yapılabildiği ekran
- RC kontrol işlemlerinin yapılabildiği ekrandır

4.5.1. Login Ekranı

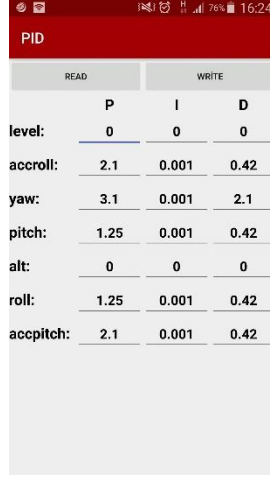
Android cihaz üzerindeki kontrolcü yazılımın quadcoptere bağlanabilmesi için IP bilgisinin girilebildiği, ilk açılış ekranıdır. IP bilgisinin kullanıcı tarafından girmesi istenilir. Bağlan butonuna dokunulduğunda, bağlantı işlemi sağlanır.



ŞEKİL 4.3: Android Login ve Menü Ekranı

4.5.2. PID Parametre Ekranı

Quadcopterin kontrolcüsü olan PID kontrolör için gereken parametrelerin görüntülendiği ve değiştirilebildiği ekrandır. Quadcopter içerisinde tanımlanması gereken tüm parametreler burada bulunmaktadır.

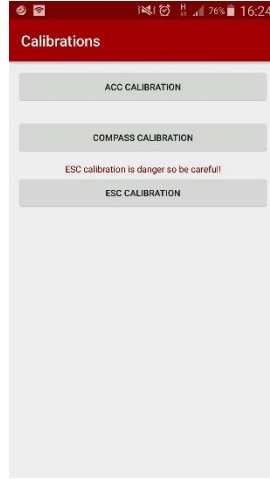


	READ	WRITE	
	P	I	D
level:	0	0	0
accroll:	2.1	0.001	0.42
yaw:	3.1	0.001	2.1
pitch:	1.25	0.001	0.42
alt:	0	0	0
roll:	1.25	0.001	0.42
accpitch:	2.1	0.001	0.42

ŞEKİL 4.4: PID Kontrolcü Parametre Ayar Ekranı

4.5.3. Kalibrasyon Ekranı

Yatay düzlem, pusula ve motor kalibrasyonunun yapılabilmesi için butonların bulunduğu ekrandır. Azami gereklilikte olduğu için sadece yatay düzlem kalibrasyonu çalışır haldedir. Yatay düzlem kalibrasyonu yapılabilmesi için, quadcopterin düz bir düzleme konulması gerekmektedir.

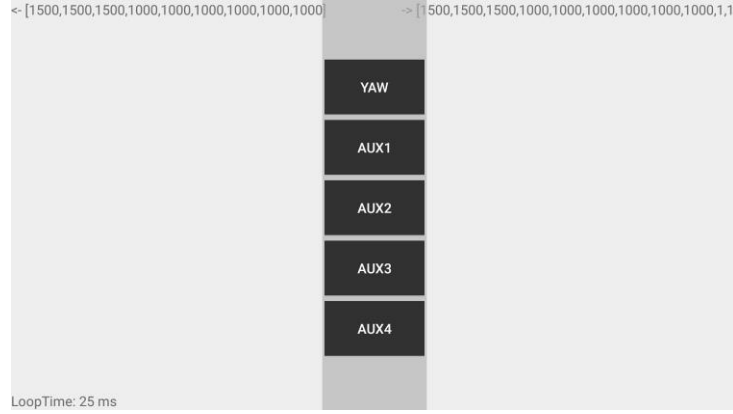


ŞEKİL 4.5: PID Kontrolcü Parametre Ayar Ekranı

4.5.4. RC Kontrol Ekranı

Uygulamanın kumanda işlevini görebilmesi için hazırlanmış ekrandır. Ekran iki parçadan oluşmaktadır. Bu parçalardan bir tanesi throttle ve yaw hareketini yönetmekte, diğer parça ise

roll ve pitch açılarını yönetmektedir. Orta bölümde bulunan butonlar AUX ayarlarının aktif hale getirilmesi veya kapatılması için kullanılmaktadır.



ŞEKİL 4.6: RC Kontrol Ekran

4.6. Konumlama Sistemi

Quadcopter'in uçuş yazılımı hazırlanıp kontrol işlemi gerçekleştirildikten sonra otopilot yazılımı geliştirilmesi planlanmaktadır. Otopilot yazılımının çalışması için quadcopterin konum verisine ihtiyacı vardır. Quadcopter küçük boyutlarda olduğu için bu konum verisi cm derecesinde hassasiyete sahip olmalıdır ve sapma değeri olmamalıdır.

Konum verisinin nasıl elde edilebileceğine dair araştırmalar gerçekleştirilmiştir. Standart GPS sistemleri cm hassasiyetinde değil metre hassasiyetindedir ve sapmalar gerçekleştirmektedir. Çok hassas sistemlerin maliyeti ise çok yüksektir.

Bulunan yöntemler içerisinde maliyet açısından en düşük, hassasiyet açısından en yüksek olan yöntem görüntü işleme ile konum tespiti olmuştur. Quadcopter üzerine, dışı belirlenen bir renkle kaplanmış küp yerleştirilecektir. Bu rengin tanınması ile konum tespit edilecektir.

Konumlama sisteminin çalışmasında iki adet kamera kullanılmaktadır. Bu kameralar yerden belirli bir yükseklikte, belirlenmiş bir alanı bir birlerine 90 derecelik açı yapacak şekilde izlemektedirler. Bir kamera kendi görüş açısına göre sadece X ve Y koordinatlarını verebilmektedir. Her iki kameradan alınan farklı düzlemlerdeki X ve Y koordinatları alınarak üç boyutlu X, Y, Z koordinatları hesaplanacaktır.

Kameraların, belirlenen rengi tanıma işleminin yapılabilmesi için OpenCV kütüphanesinin kullanılması kararlaştırılmıştır. Bu kütüphane birçok platformda ve birçok dilde çalışacak şekilde hazırlanmış ve uyarlanmıştır.

OpenCV (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. 1999 yılında İntel tarafından geliştirilmeye başlanmış daha sonra Itseez, Willow, Nvidia, AMD, Google gibi şirket ve toplulukların desteği ile gelişim süreci devam etmektedir. İlk sürüm olan OpenCV alfa 2000 yılında piyasaya çıkmıştır. İlk etapta C programlama dili ile geliştirilmeye başlanmış ve daha sonra birçok algoritması C++ dili ile geliştirilmiştir. Open source yani açık kaynak kodlu bir kütüphanedir ve BSD lisansı ile altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelmektedir. OpenCV platform bağımsız bir kütüphanedir, bu sayede Windows, Linux, FreeBSD, Android, Mac OS ve iOS platformlarında çalışabilmektedir. C++, C, Python, Java, Matlab, EmguCV kütüphanesi aracılığıyla da Visual Basic.Net, C# ve Visual C++ dilleri ile topluluklar tarafından geliştirilen farklı wrapperlar aracılığıyla Perl ve Ruby programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilir.

OpenCV kütüphanesi içerisinde görüntü işlemeye (image processing) ve makine öğrenmesine (machine learning) yönelik 2500'den fazla algoritma bulunmaktadır. Bu algoritmalar ile yüz tanıma, nesneleri ayırt etme, insan hareketlerini tespit edebilme, nesne sınıflandırma, plaka tanıma, üç boyutlu görüntü üzerinde işlem yapabilme, görüntü karşılaştırma, optik karakter tanımlama OCR (Optical Character Recognition) gibi işlemler rahatlıkla yapılabilmektedir.

OpenCV kütüphanesini bileşenleri;

- **Core:** OpenCV'nin temel fonksiyonları ve matris, point, size gibi veri yapılarını bulundurur. Ayrıca görüntü üzerine çizim yapabilmek için kullanılabilecek metotları ve XML işlemleri için gerekli bileşenleri barındırır.
- **HighGui:** Resim görüntüleme, pencereleri yönetme ve grafiksel kullanıcı arabirimleri için gerekli olabilecek metotları barındırır. 3.0 öncesi sürümlerde dosya sistemi üzerinden resim dosyası okuma ve yazma işlemlerini yerine getiren metotları barındırmaktaydı.
- **Imgproc:** Filtreleme operatörleri, kenar bulma, nesne belirleme, renk uzayı yönetimi, renk yönetimi ve eşikleme gibi neredeyse tüm fonksiyonları içine alan bir pakettir. 3 ve sonra sürümlerde bazı fonksiyonlar değişmiş olsada 2 ve 3 sürümünde de bir çok fonksiyon aynıdır.
- **Imgcodecs:** Dosya sistemi üzerinden resim ve video okuma/yazma işlemlerini yerine getiren metotları barındırmaktadır.

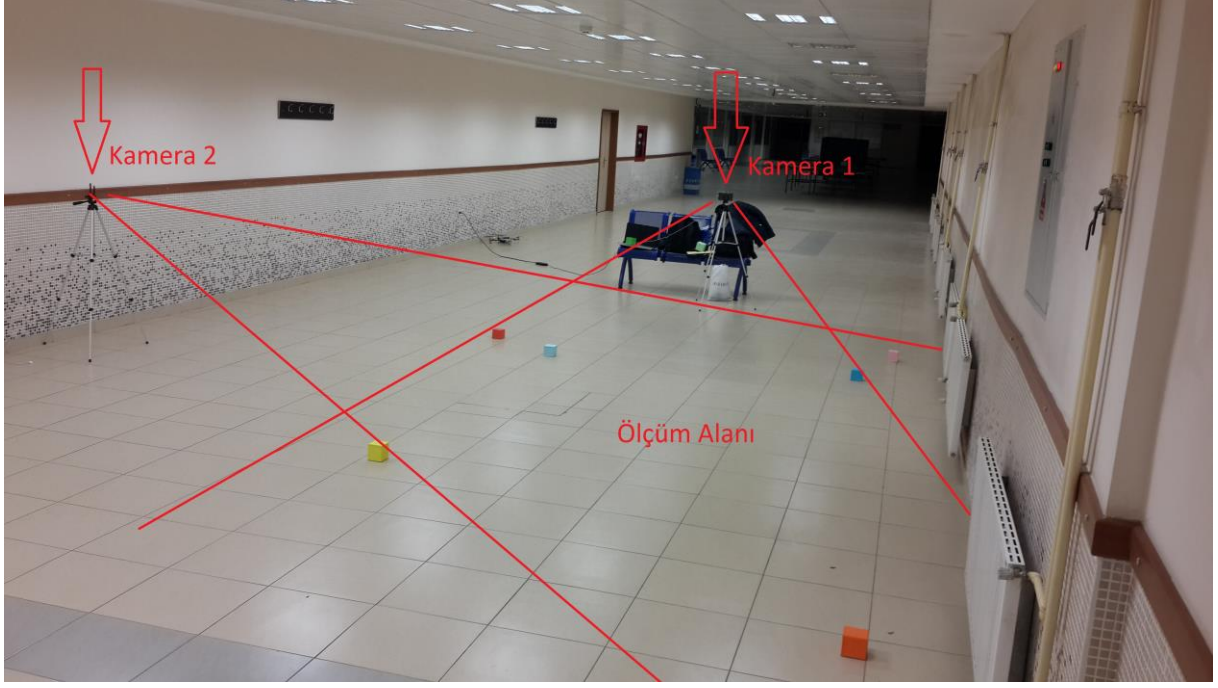
- **Videoio:** Kameralara ve video cihazlarına erişmek ve görüntü almak ve görüntü yazmak için gerekli metotları barındırır. OpenCV 3 sürümü öncesinde bu paketteki birçok metot video paketi içerisindeydi [3].

OpenCV kütüphanesinin çalışıp, belirlenen rengin tanınacağı kamera olarak android işletim sistemine sahip iki adet Samsung Note 3 telefon kullanılmıştır. Quadcopterin kontrolü için tasarımı yapılan uygulamanın içerisine eklenen bir buton ile görüntü tanıma işlemi başlatılacak, ayrıca ek bir uygulama hazırlanmayacaktır. Görüntüde tanınacak rengin belirlenmesi, kullanıcıyı ekrandaki görüntü üzerinden renge dokunması ile yapılmaktadır. Kullanıcı ekrana dokunduğu anda dokunduğu yerdeki renk tanınacak renk olarak belirlenmektedir.

İki kamera ŞEKİL 2.4’de görüldüğü gibi yerleştirilmiştir ve kameraların içinden, kameraya dik şekilde geçtiği varsayılan doğrular, birbirlerine dik olacak şekilde kesiştirilmiştir. Yerleşim şekli bu şekilde oluşturulduğunda, her iki kameranın da görüntü alabildiği ve konumlama işleminin yapılabildiği bölge ŞEKİL 2.5’de görüldüğü kadar bir bölge olmaktadır. Bu bölge bizim çalışmayı gerçekleştirdiğimiz yerde çok geniş olmamakla birlikte, duvarların olmadığı, açık bir alanda çok daha geniş bir bölge olabilmektedir.



4.7: Kameraların Yerleşimi



ŞEKİL 4.8: Kameraların yerleşimi ve Ölçüm Yapılan Alan

Renk belirlendikten sonra kameradan OpenCV'nin kenar bulma fonksiyonu ile tanınan renk kümelerinin kenarlar pikselleri array olarak elde edilmektedir. Daha sonra bu kenarların içinde kalan orta nokta hesaplanmaktadır. Sonuç olarak cismin merkezinin X ve Y koordinatı piksel olarak elde edilmektedir. Konum hesaplanması için gereken veri cm türünde olmalıdır. Piksel bilgisinin cm'ye çevrilmesi için kameranın kaç pikselinin kaç cm'ye eşit olduğunun hesaplanması gerekiyor. Bunun için kameradan bir metre uzaklıkta, kameranın görebildiği en sağ ve en sol nokta belirlenip bu iki nokta arasındaki mesafe ölçülmüştür.

$$GK = TPOK \times \frac{KMGM}{KPS}$$

GK = Gerçek konum

TPOK = Telefondan Piksel Olarak Konum

KMGM = Kamerada Max Görülen Mesafe

KPS = Kamera Piksel Sayısı

Formülü ile hesaplanmaktadır. Kameralar, konumlama işlemi yapılacak alana yerleştirilirken orijin olarak belirlenen noktaya olan uzaklıkları, yerden yükseklikleri ve yerleşime göre hangi kamera olduğu bilgileri girilmesi gerekmektedir. Bu bilgiler kullanılarak aşağıdaki ve formüller ile X, Y ve Z koordinatları hesaplanmaktadır.

$$x = \frac{(z_2 - z_1) * x_1 * x_2}{x_1 * z_2 - x_2 * z_1}$$

$$z = \frac{(x_2 - x_1) * z_1 * z_2}{x_2 * z_1 - x_1 * z_2}$$

$x_1 = 0$ ise;

$$y = y_2$$

$z_2 = 0$ ise;

$$y = y_1$$

$x_1 \neq 0$ ve $z_2 \neq 0$ ise;

$$y = \frac{(x_2 - x) * y_2}{x_2} = \frac{z_1 * y_1}{z_1 + z} = \frac{y_2 * z}{z_2} = \frac{x * y_1}{x_1}$$

x_1 = Kamera 1'den gelen x konumu

y_1 = Kamera 1'den gelen y konumu

z_1 = Kamera 1'in origine olan uzaklığı

x_2 = Kamera 2'nin origine olan uzaklığı

y_2 = Kamera 2'den gelen y konumu

z_2 = Kamera 2'den gelen x konumu

x, y, z = Gerçek koordinatlar

Her kamera belirlenen rengi, görüntü üzerinde tanıyıp, merkez noktasını bulduktan ve x, y konumlarını hesapladıktan sonra quadcoptere bu konum verilerini göndermektedir. Kameralardan gelen x, y konum bilgileri ile quadcopter, verilen formülleri çalıştırmakta ve üç boyutlu ekseninde x, y, z konumunu hesaplamaktadır.

5. SONUÇ

İlk dönem yapmış olduğumuz, açık kaynak kodlu yazılım ile uçuş işlemini gerçekleştiren, quadcopter gömülü yazılımı projesinde, edindiğimiz tecrübeler ile bu dönem tüm yazılımı kendimiz hazırlamaya karar verdik. Bu bağlamda, ilk olarak yazılımı çalıştırdığımız geliştirme kartını değiştirip, daha güçlü özelliklere sahip, Debian işletim sistemi ile çalışan bir Single Board Computer edindik. Gerekli araştırmaları ve literatür taramalarını gerçekleştirdik. Bunun sonucunda tamamen kendi tasarımı ve kodlamamız olan quadcopter gömülü yazılımını hazırladık. Quadcopterin kontrol edilebilmesi için Android telefonlarda çalışan kumanda uygulaması geliştirdik.

Quadcopterin kumanda ile kontrolünün yanı sıra otopilot yazılımı ile birlikte otomatik uçuş yazılımı geliştirmeyi planladık. Bu sistemin çalışabilmesi için quadcopterin konum bilgisine sahip olması ve bu bilginin yeteri seviyede stabil olması gerekiyordu. Konum verisinin elde edilebilmesi için kullanılabilecek yöntemleri taradık. Kullanılabilecek birçok yöntem olmasıyla birlikte, maliyet ve stabilite açısından en iyi sistemin kameralar ile konum algılama sistemleri olduğuna karar verdik. Bunun sonucunda iki adet Samsung Note 3 telefonun kamerasını kullanarak konum algılayabilen bir sistem geliştirdik.

6. KAYNAKLAR

1. Korkmaz, H., *Sabit Kanatlı Bir İnsansız Hava Aracı İçin Takip Ve Stabilizasyon Amaçlı Otopilot Sistemi Geliştirilmesi*, Yüksek Lisans Tezi, TOBB Ekonomi Ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 2013.
2. Ökten, İ., Üçgün, H., Yüzgeç, U., & Kesler, M. *Döner Kanat İnsansız Hava Aracının Dinamik Modellenmesi ve Benzetimi*.
3. OpenCV Nedir? URL: <http://mesutpiskin.com/blog/opencv-nedir.html>, Son Erişim Tarihi: 12.04.2017
4. Koyuncu, G. (2015). *İnsansız Hava Aracı Disiplinlerarası Tasarım Optimizasyonu* (Doctoral dissertation, Fen Bilimleri Enstitüsü).
5. Merç, Y., & Bayılmış, C. (2011, May). Dört Rotorlu İnsansız Hava Aracı (Quadrotor) Uygulaması. In *6th International Advanced Technologies Symposium (IATS'11)* (pp. 18-20).
6. Üçgün, H., Ökten, İ., Yüzgeç, U., & Kesler, M. *Dört Rotorlu Döner Kanat İnsansız Hava Aracı Test Düzeneği Geliştirilmesi*.
7. Elbir, Ö., Yıldız, H., & Kasnakoğlu, C. *Dört Rotorlu İha İçin H_∞ Loop Shape Yöntemi İle Dengeleyici Otopilot Tasarımı*.
8. Kıyak, E., & Göl, G. *Arduino Geliştirme Kartı ile Döner Kanatın Kontrolü ve Kontrol Yazılımlarının Geliştirilmesi*.