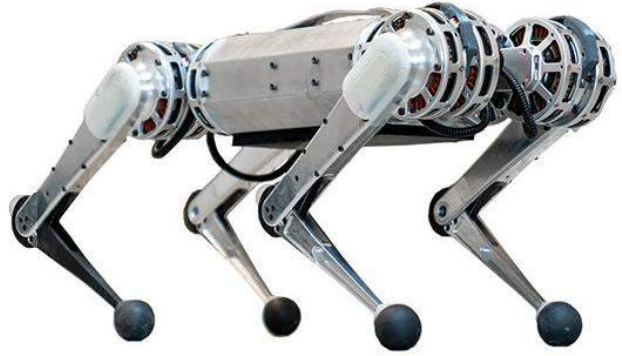


Ek 1 Dört Ayaklı Robot Tasarımında Öne Çıkan Çalışmalar



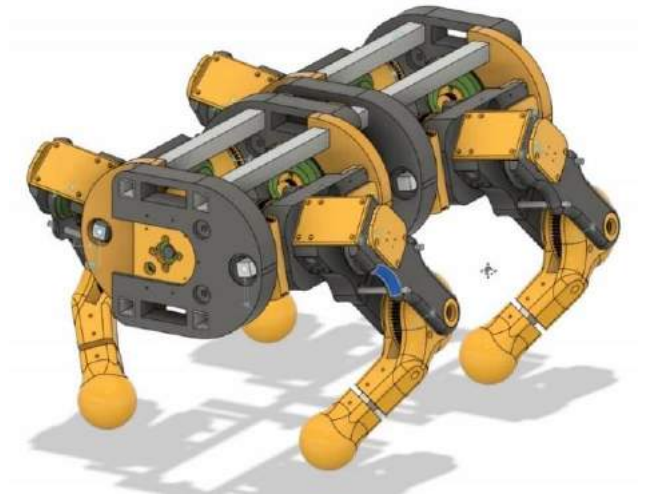
Boston Dynamics'in geliştirdiği SPOT



Massachusetts Institute of Technology'nin (MIT) geliştirdiği CHEETAH

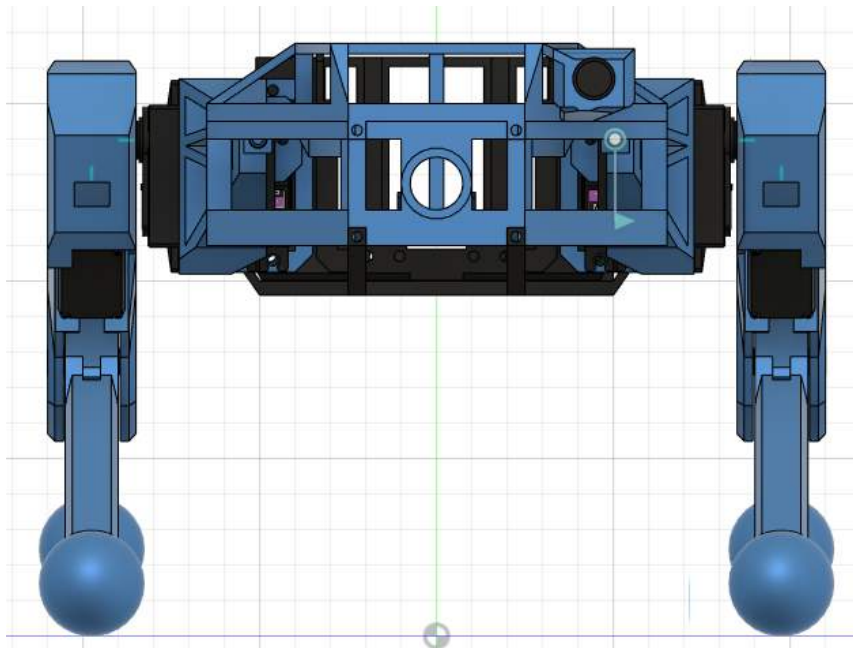
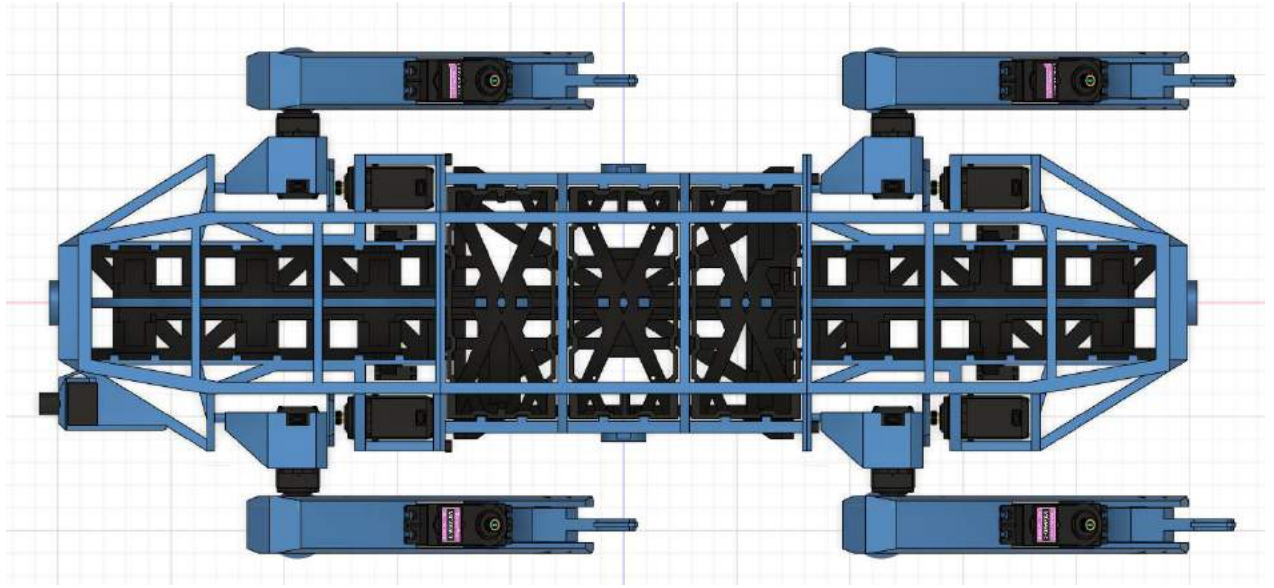
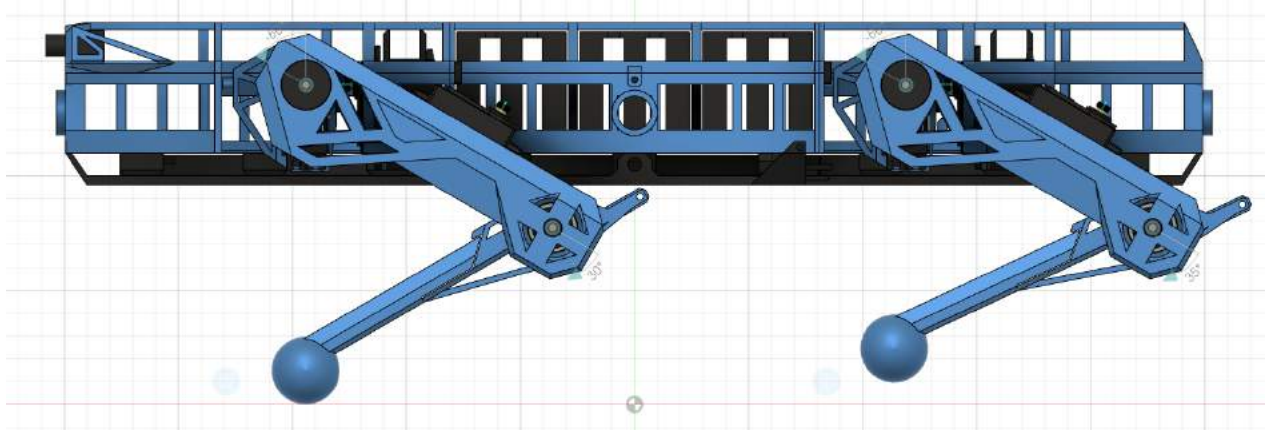


Akin Robotics'in geliştirdiği ARAT



James Bruton'un geliştirdiği openDog V2

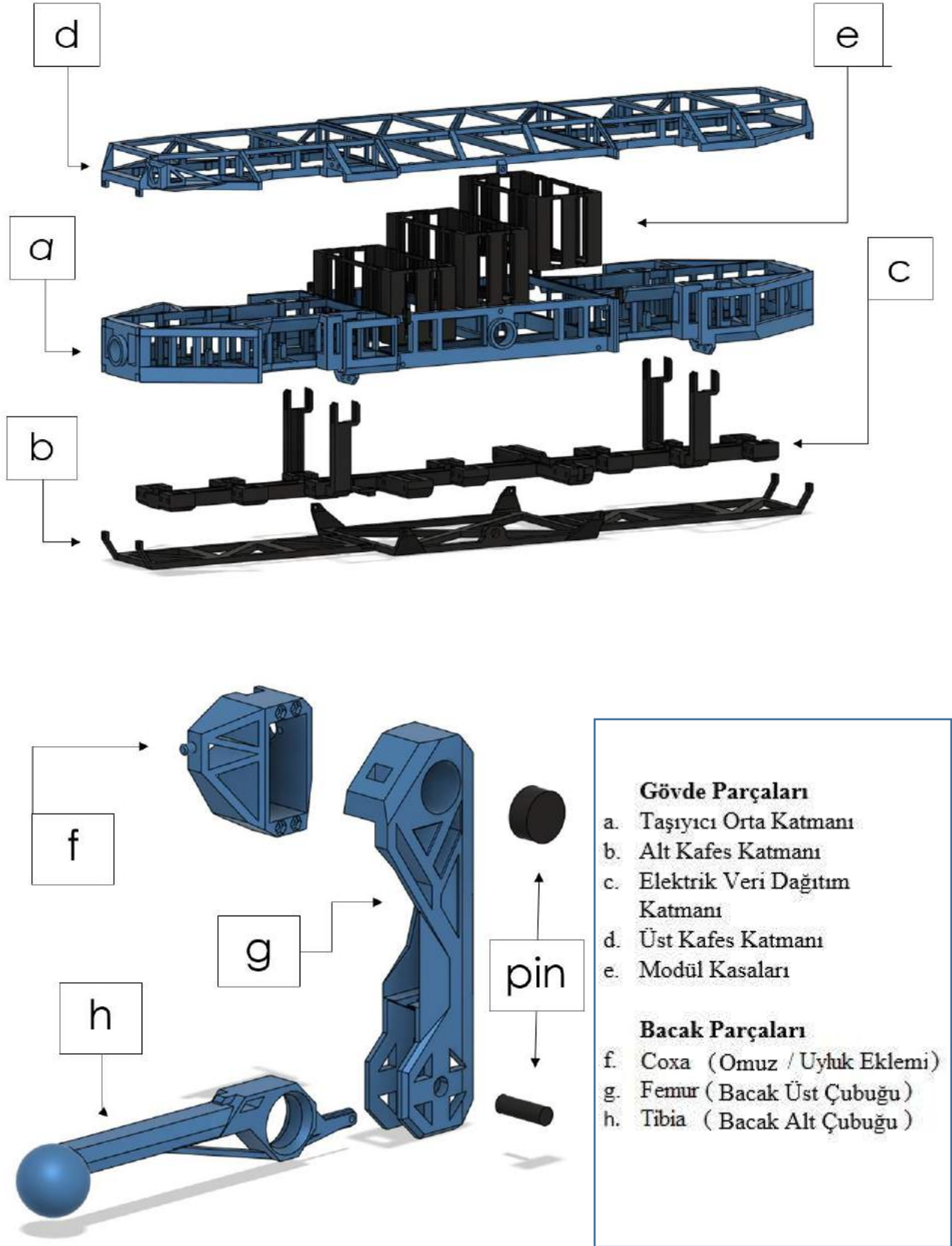
Ek 2 Şase Genel Görünüm



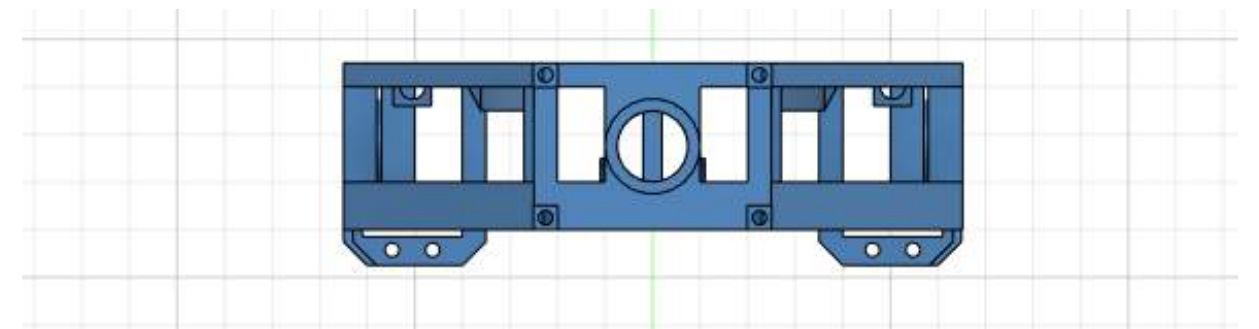
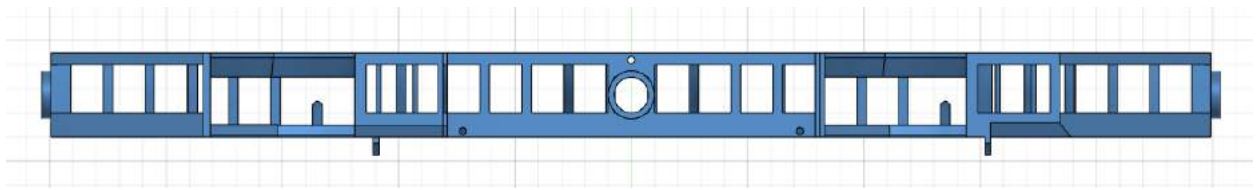
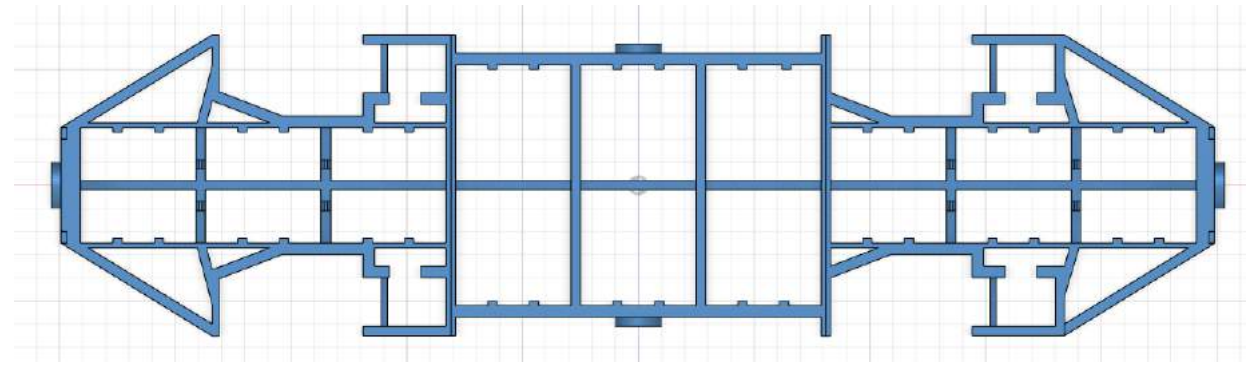
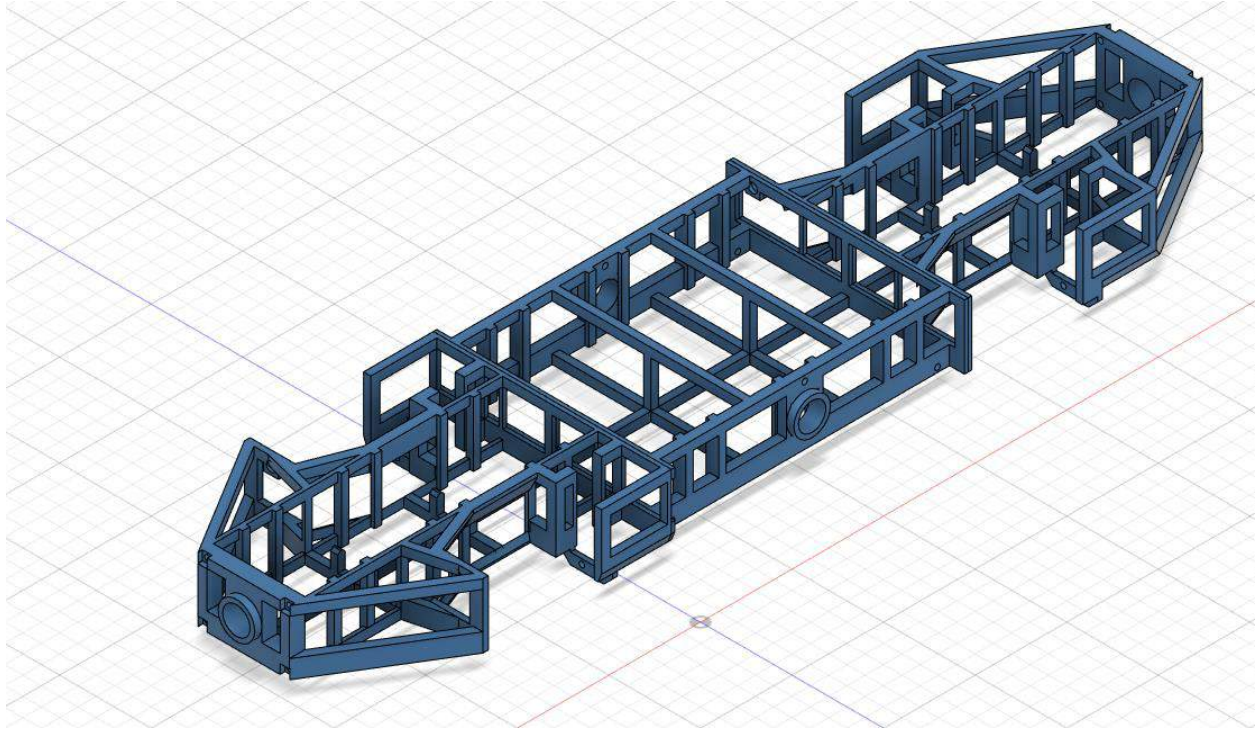
Ek 3 Gövde ve Bacaklar Malzeme Listesi

GÖVDE VE BACAKLAR MALZEME LİSTESİ			
		Malzeme	Adet
1		Filament 1.75 mm Turkuaz PLA - ABG	1
2		Filament 1.75 mm Siyah PLA - ABG	1
3		20x30 Delikli Plaket	1
4		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
5		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
6		Dişi Pin Header 2.54mm	12
7		12mm Erkek Header	17
8		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	92
9		M3 Somun	52
10		MG958 X-Large Digital Servo	12
11		25T Servo Motor Kolu – Siyah	4
12		25T Servo Motor Metal Başlığı 14 mm Aliminyum	8
13		608ZZ Rulman - 3D Yazıcı Rulmanı	4
14		Eachine TX01 NTSC Super Mini AIO 5.8G 40CH 25MW VTX 600TVL 1/4 Cmos FPV Camera	1
15		90C Dişi Header	12
16		32mm Erkek Header	21
17		Yarıklı Ağırlık Takımı	2
18		RP-L-170 İnce Film Basınç Sensörü	2
19		1k Ohm Direnç	4

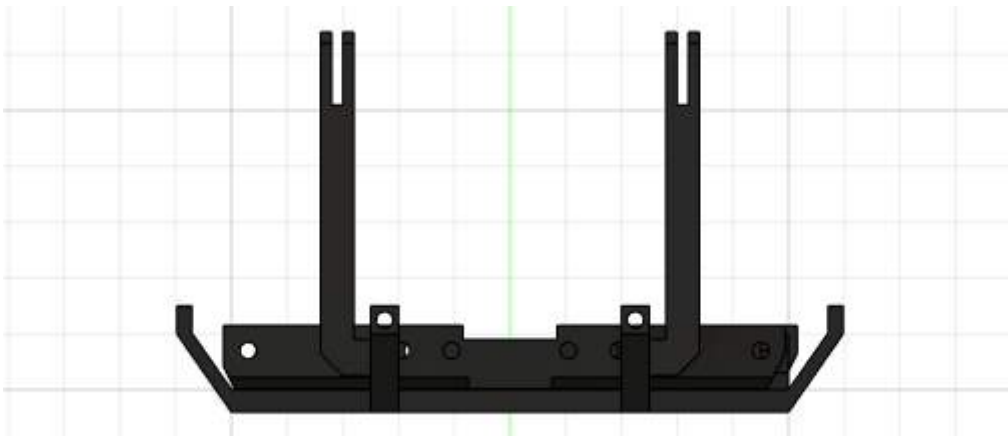
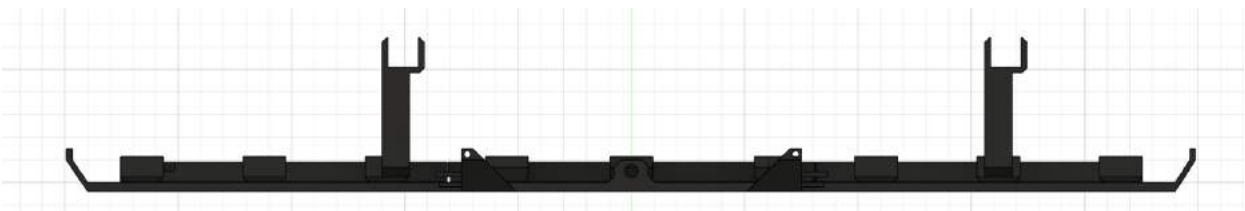
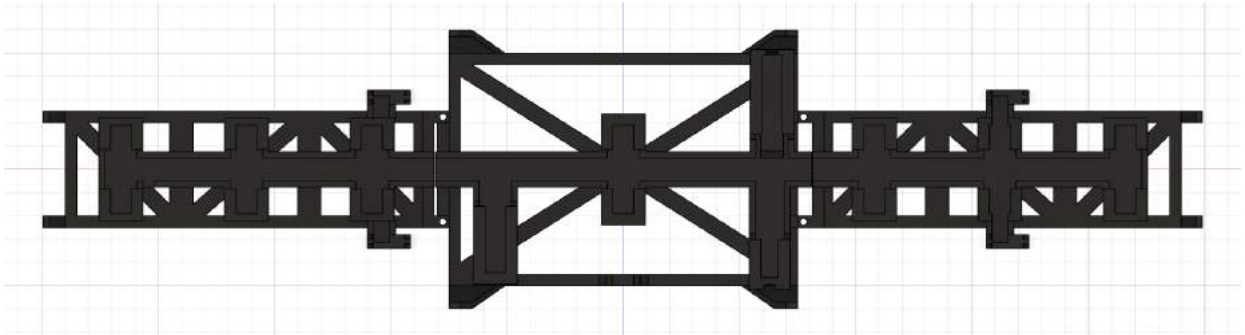
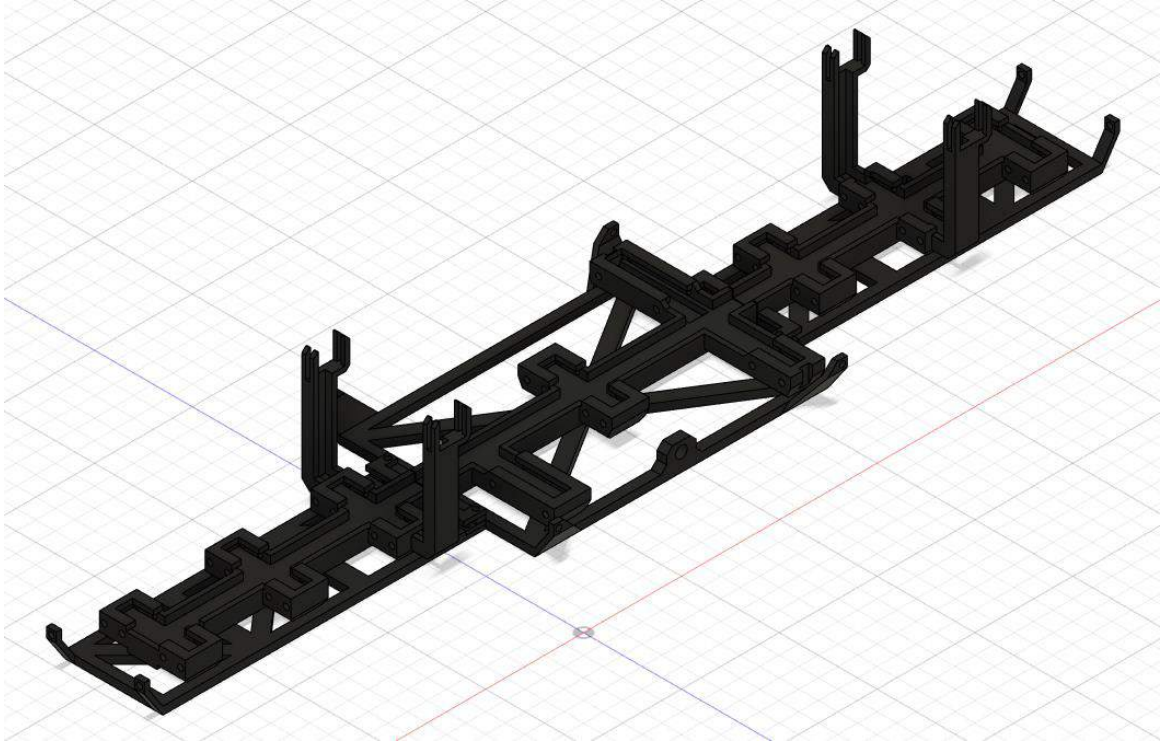
Ek 4 Gövde ve Bacak Parçaları



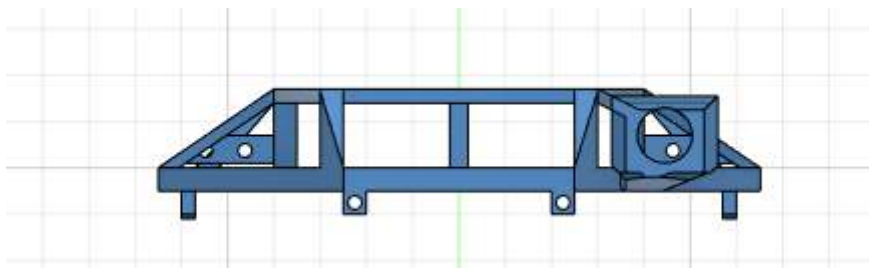
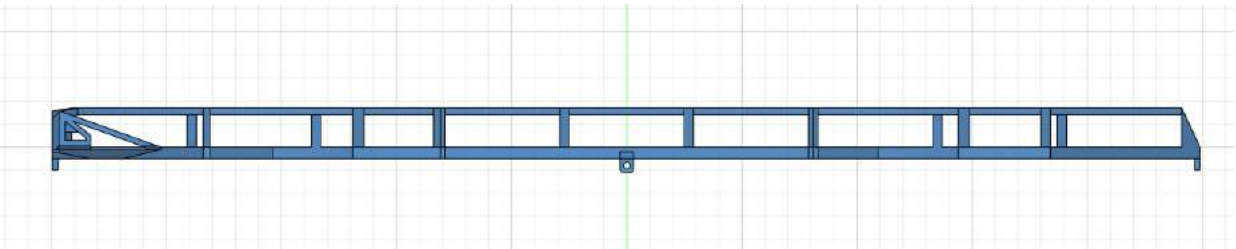
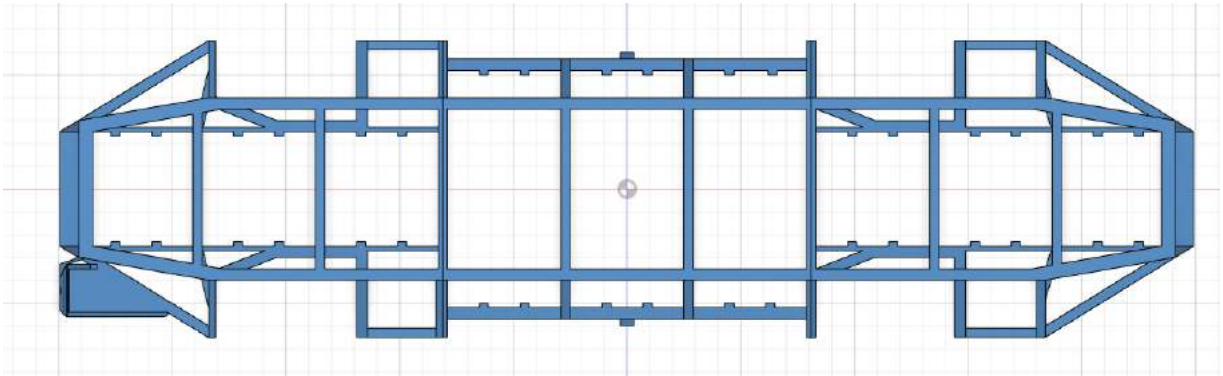
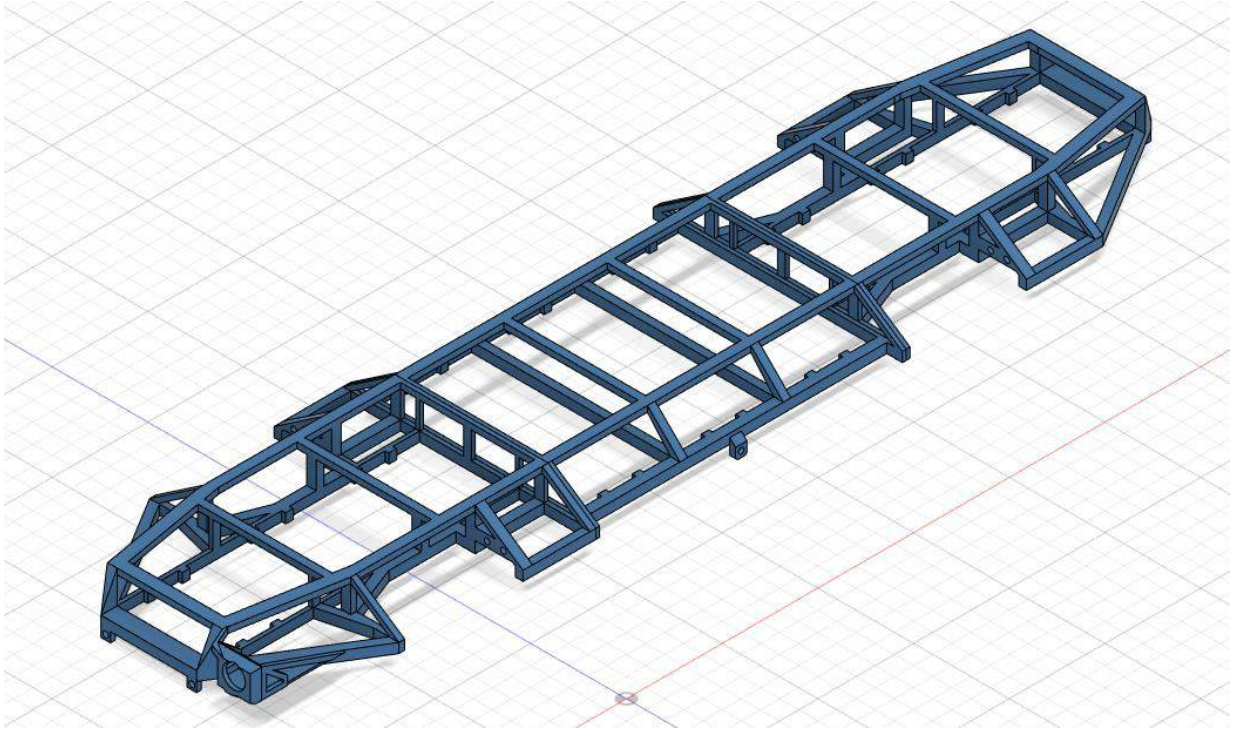
Ek 5 Taşıyıcı Orta Katman



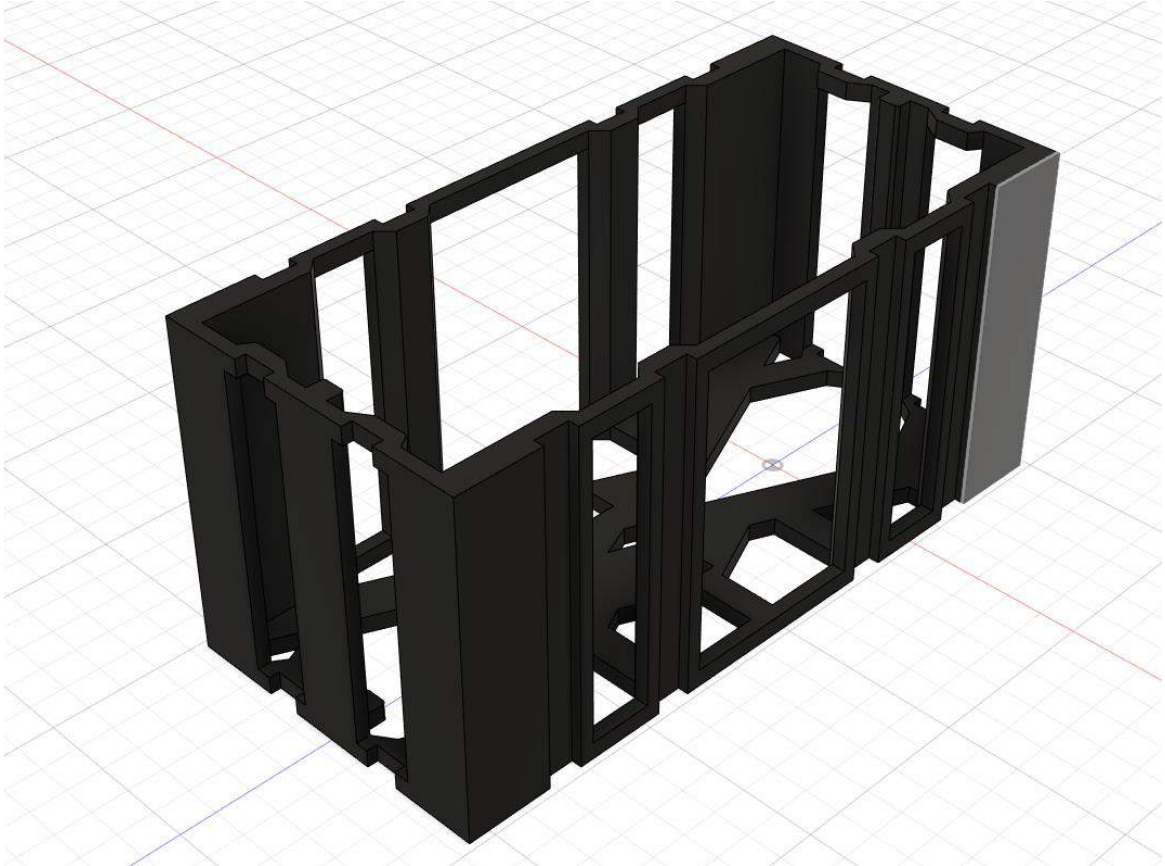
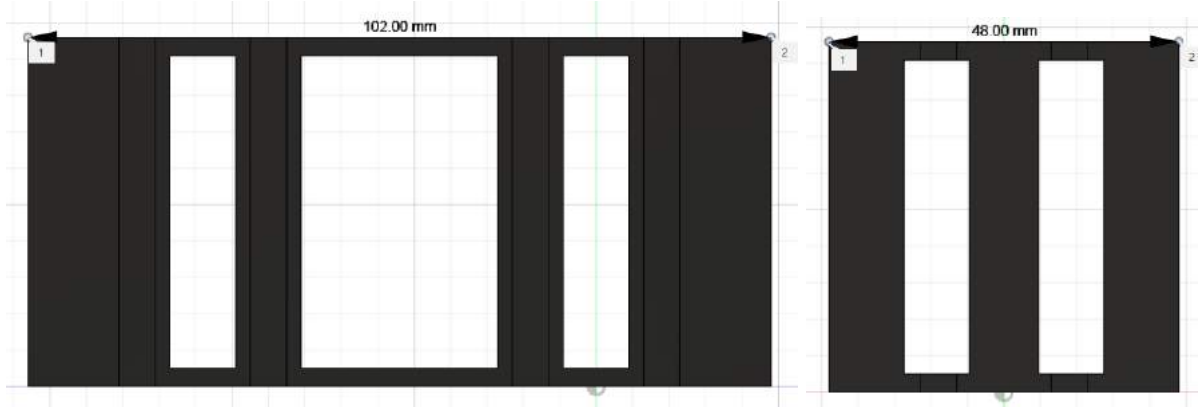
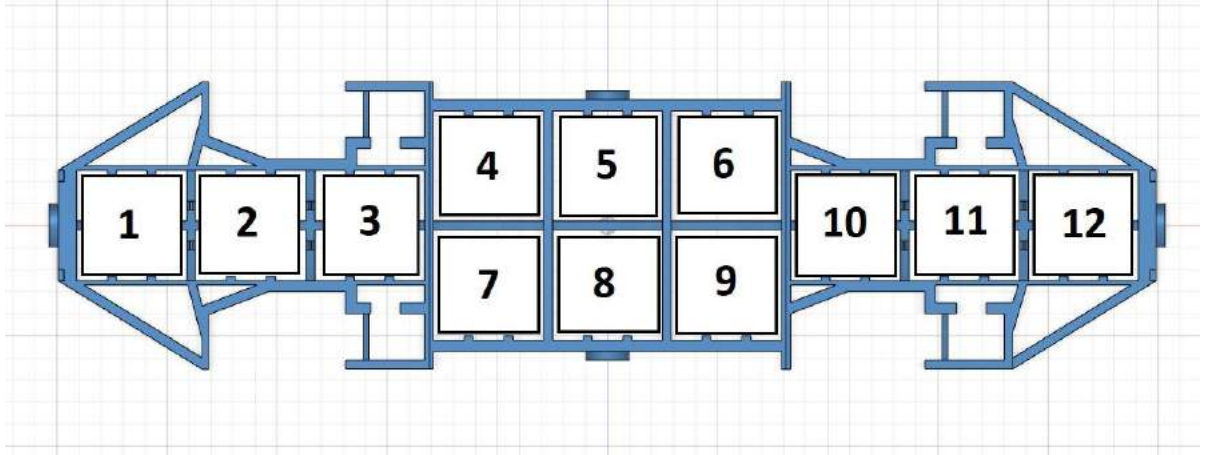
Ek 6 Alt Kafes Katmanı, Elektrik ve Veri Dağıtım Katmanı



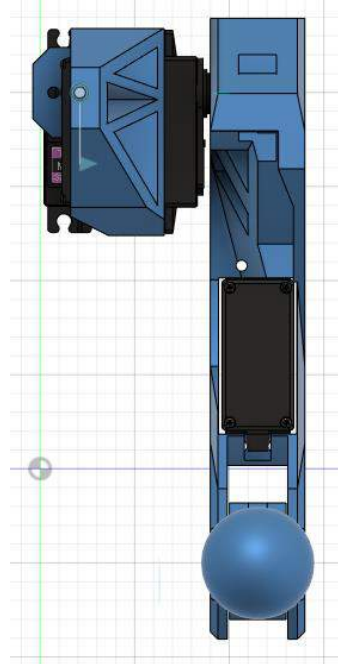
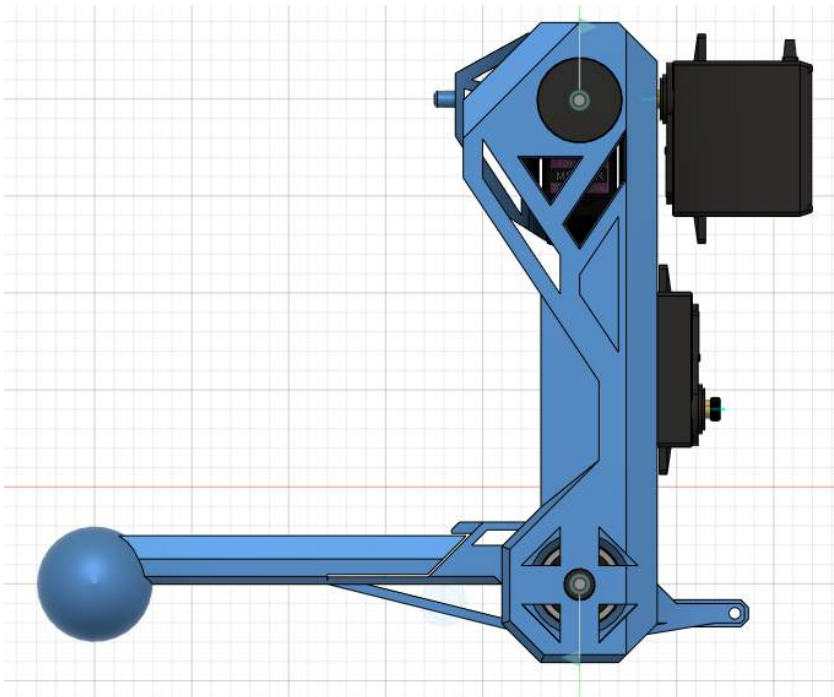
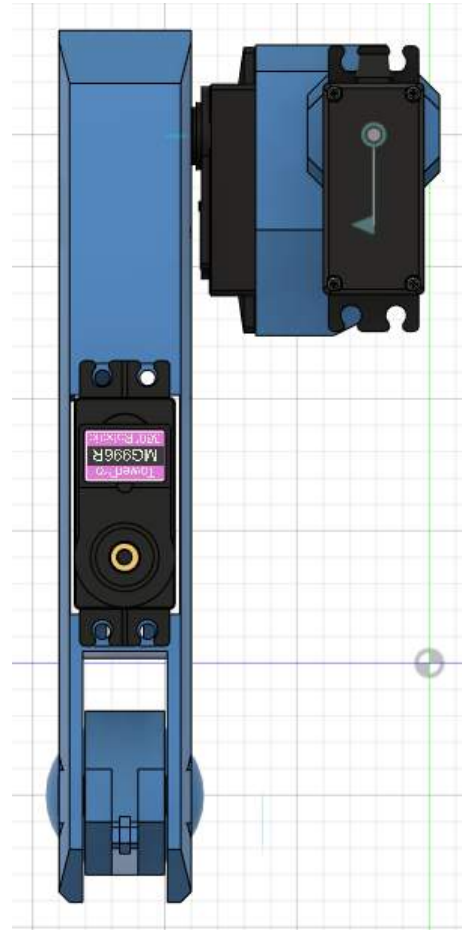
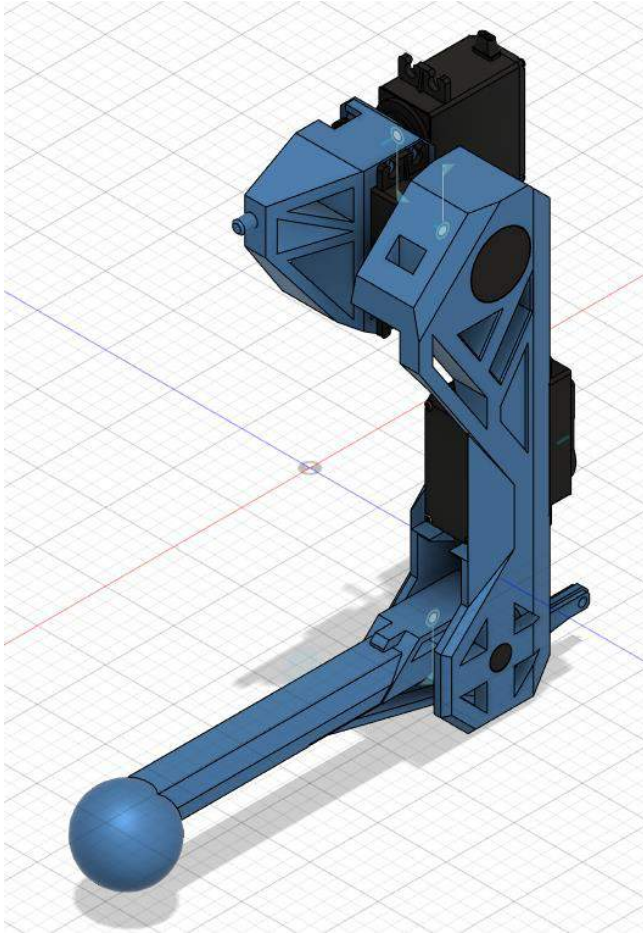
Ek 7 Üst Kafes Katmanı



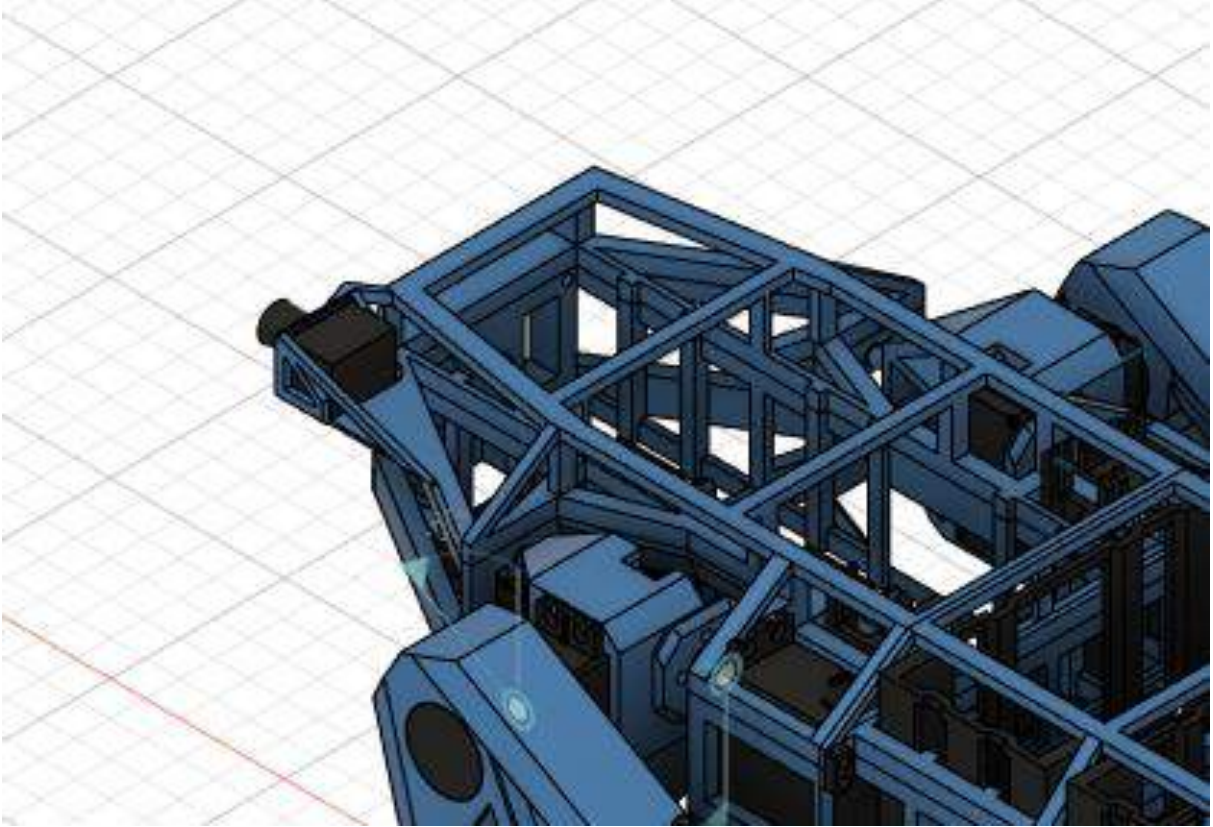
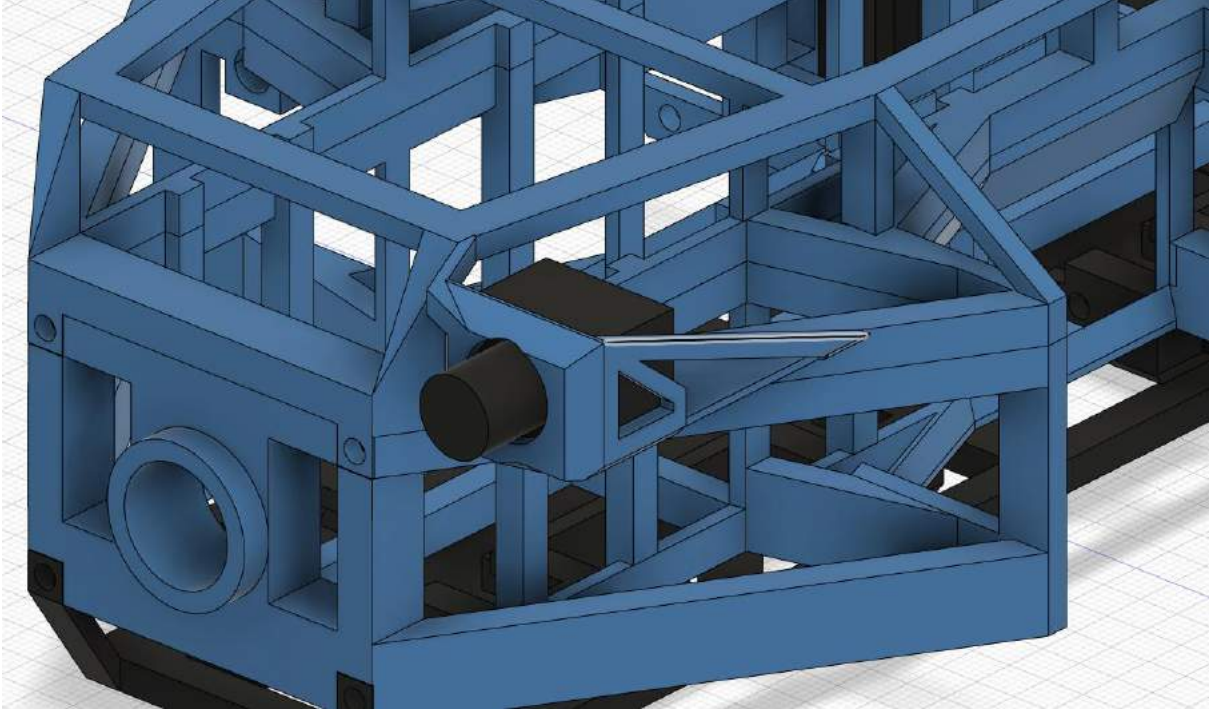
Ek 8 Modül Kasaları



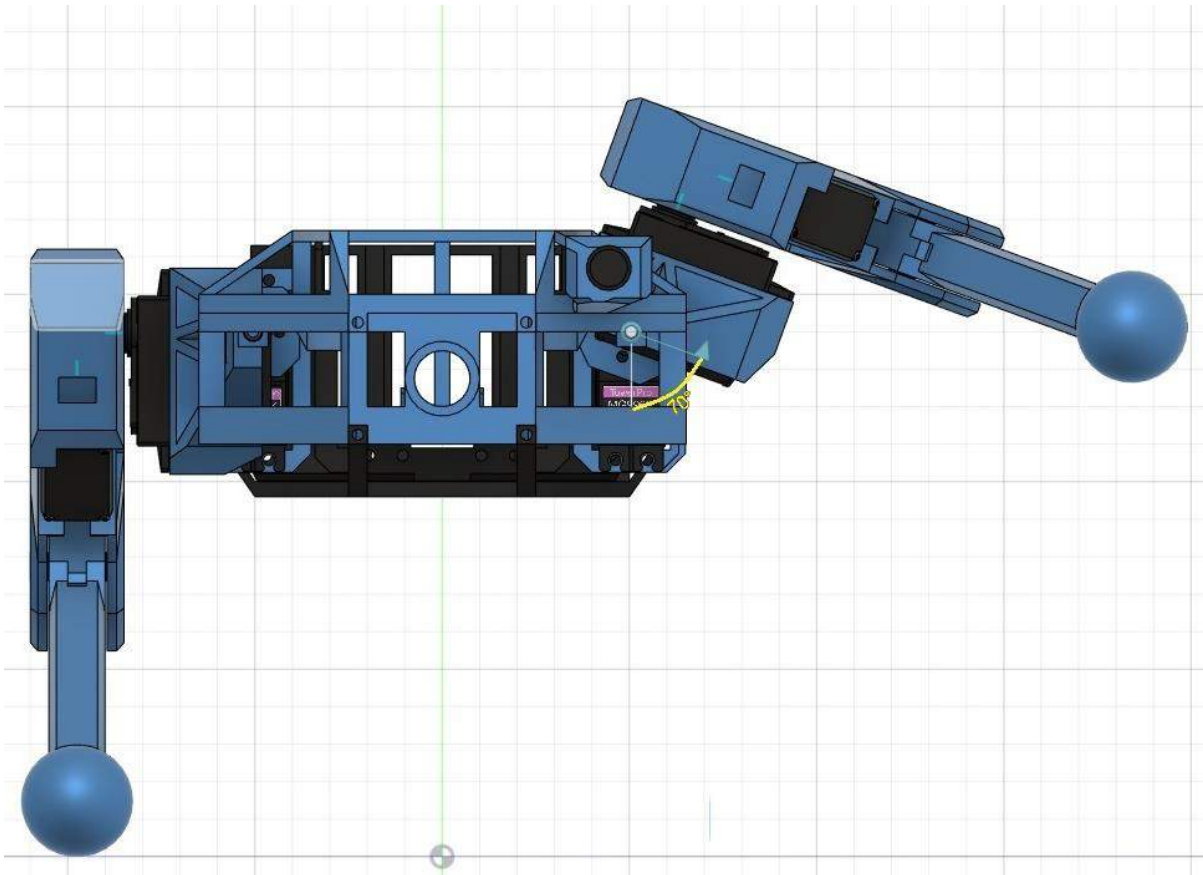
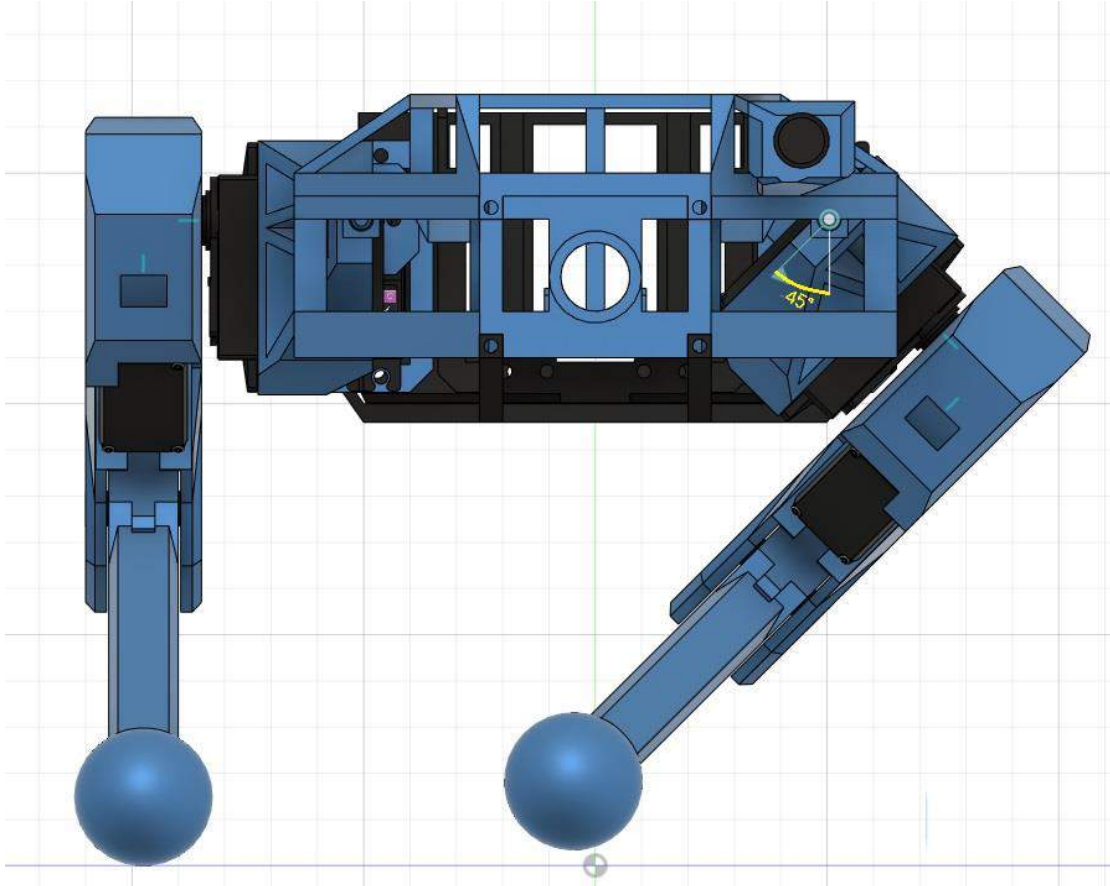
Ek 9 Bacaklar



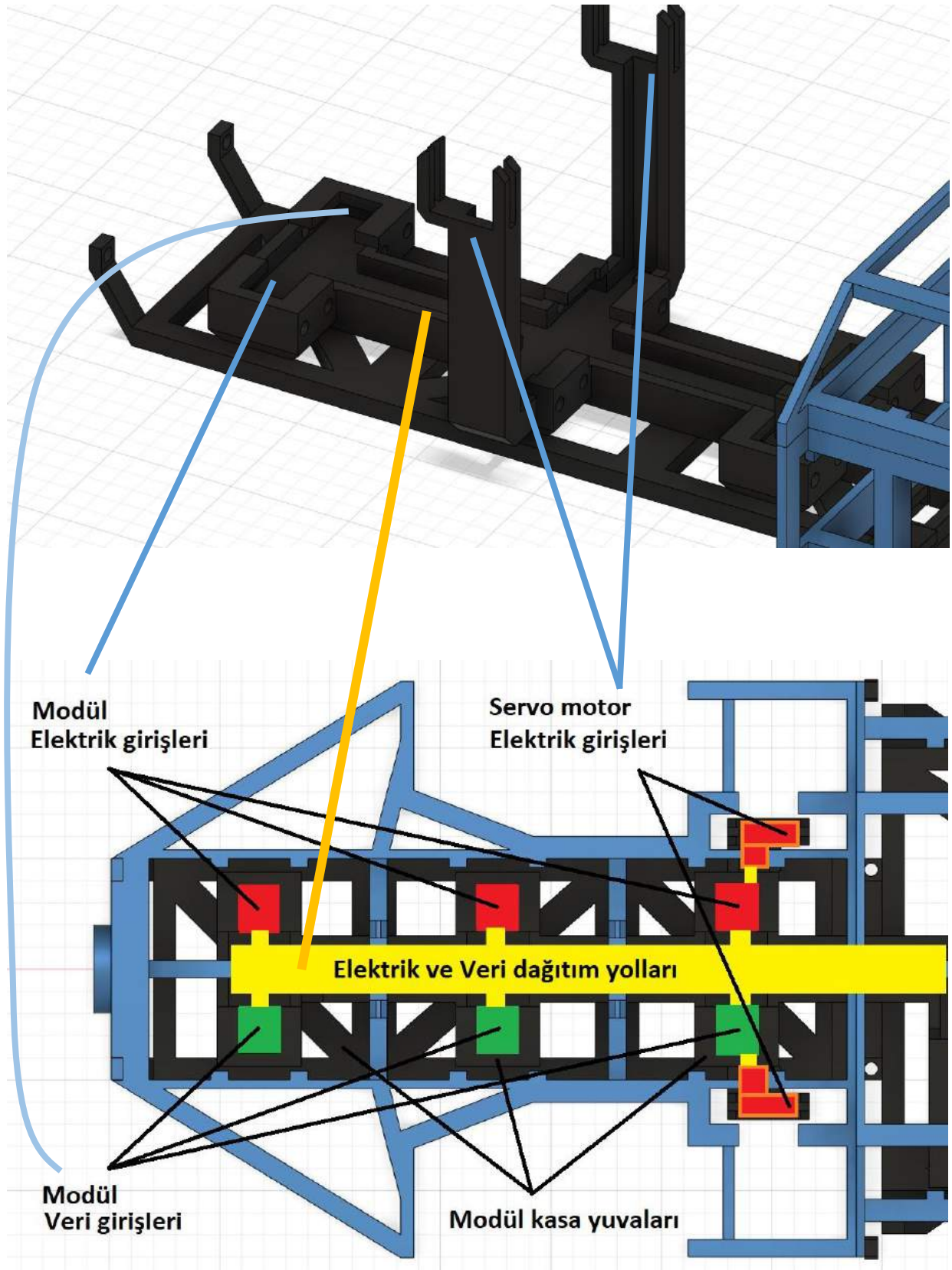
Ek 10 Üst Kafeste Kamera Ekipmanı için tasarlanmış alan



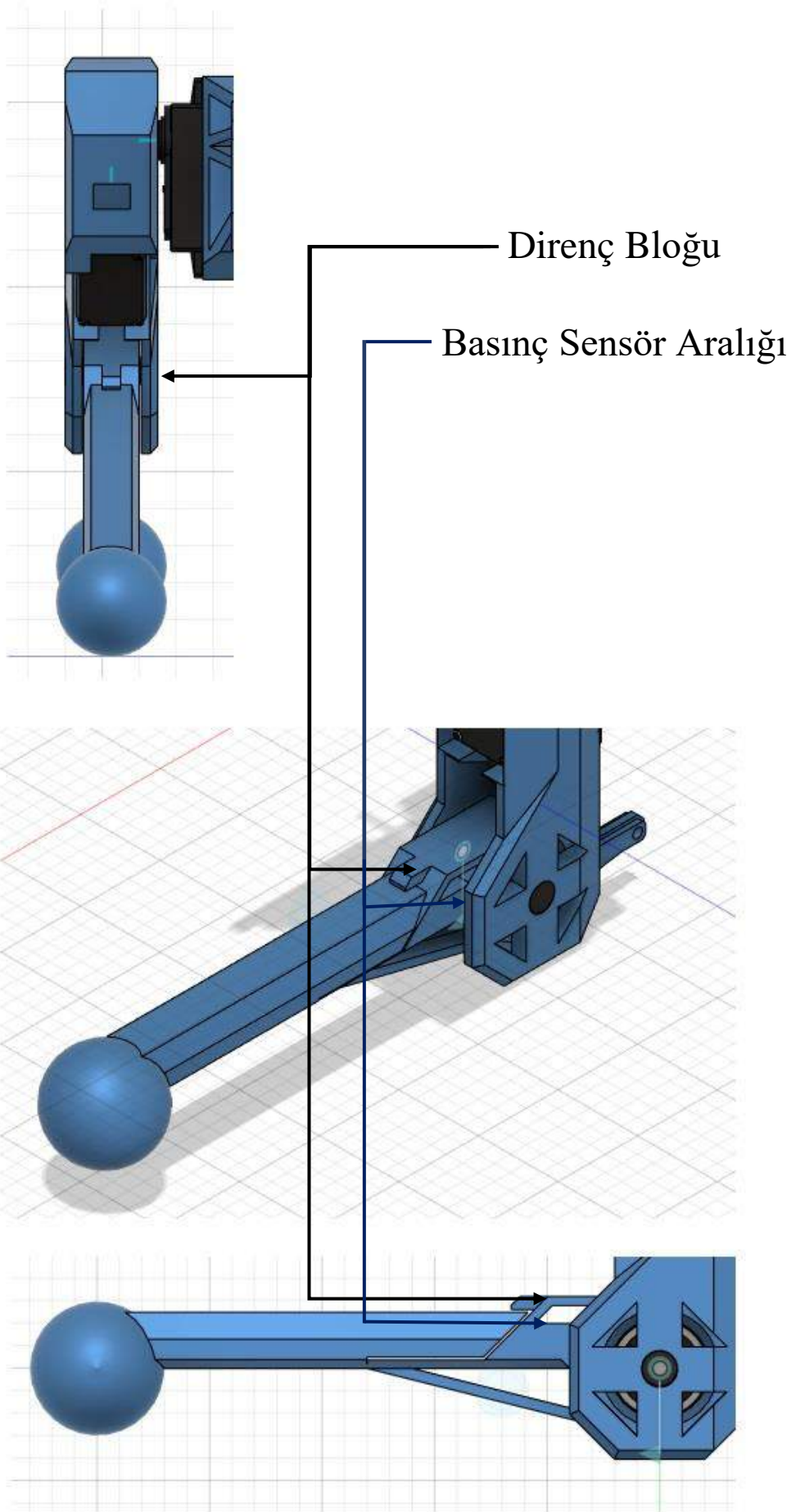
Ek 11 Şasede omuz hareketi için yapılan eğimlerin tasarımı



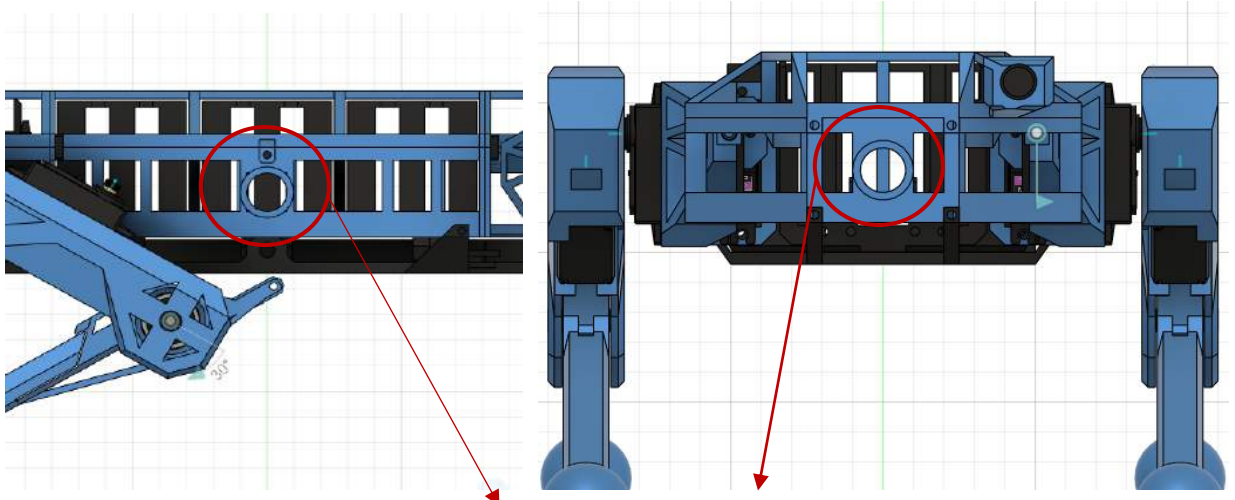
Ek 12 Elektrik Veri dağıtım ve girişleri tasarımı



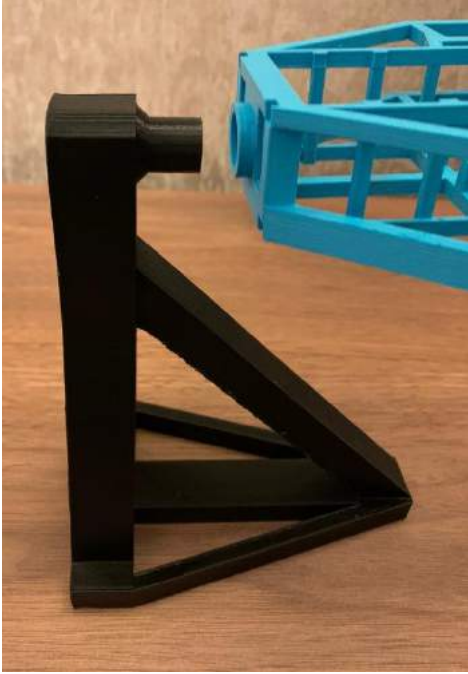
Ek 13 Basınç Sensörünün Bacak Alt Çubuğu üzerindeki yerleşim tasarımı



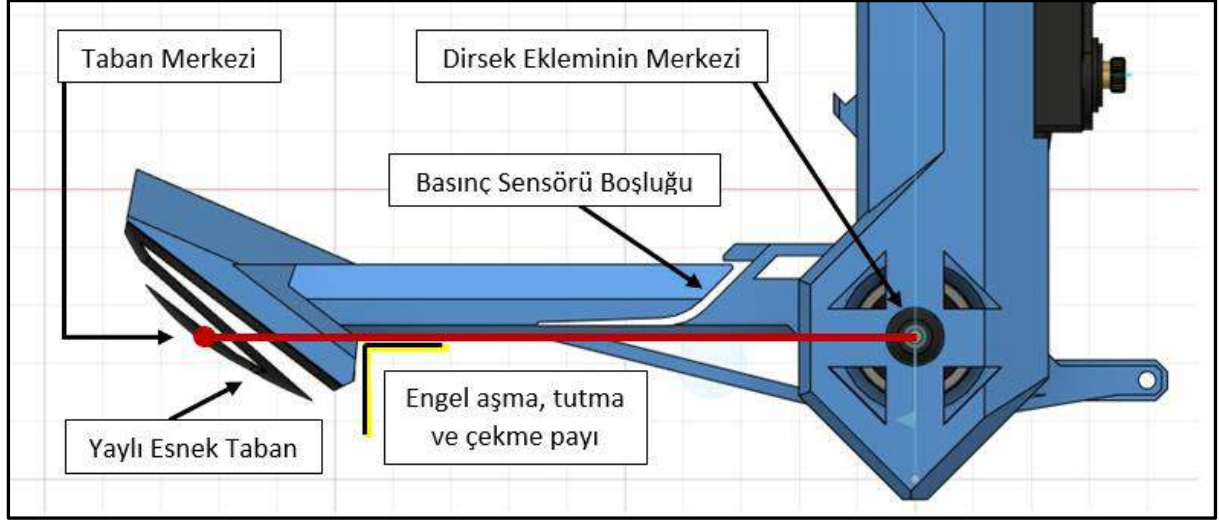
Ek 14 Denge Merkezleri



Askı Stant tutma boşlukları

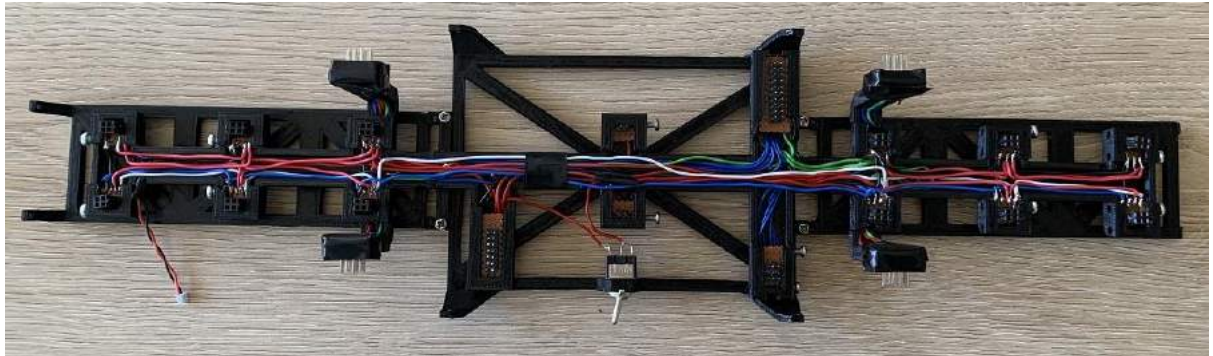


A. Ayak ve Bacak Alt Çubuğu Bağlantısı

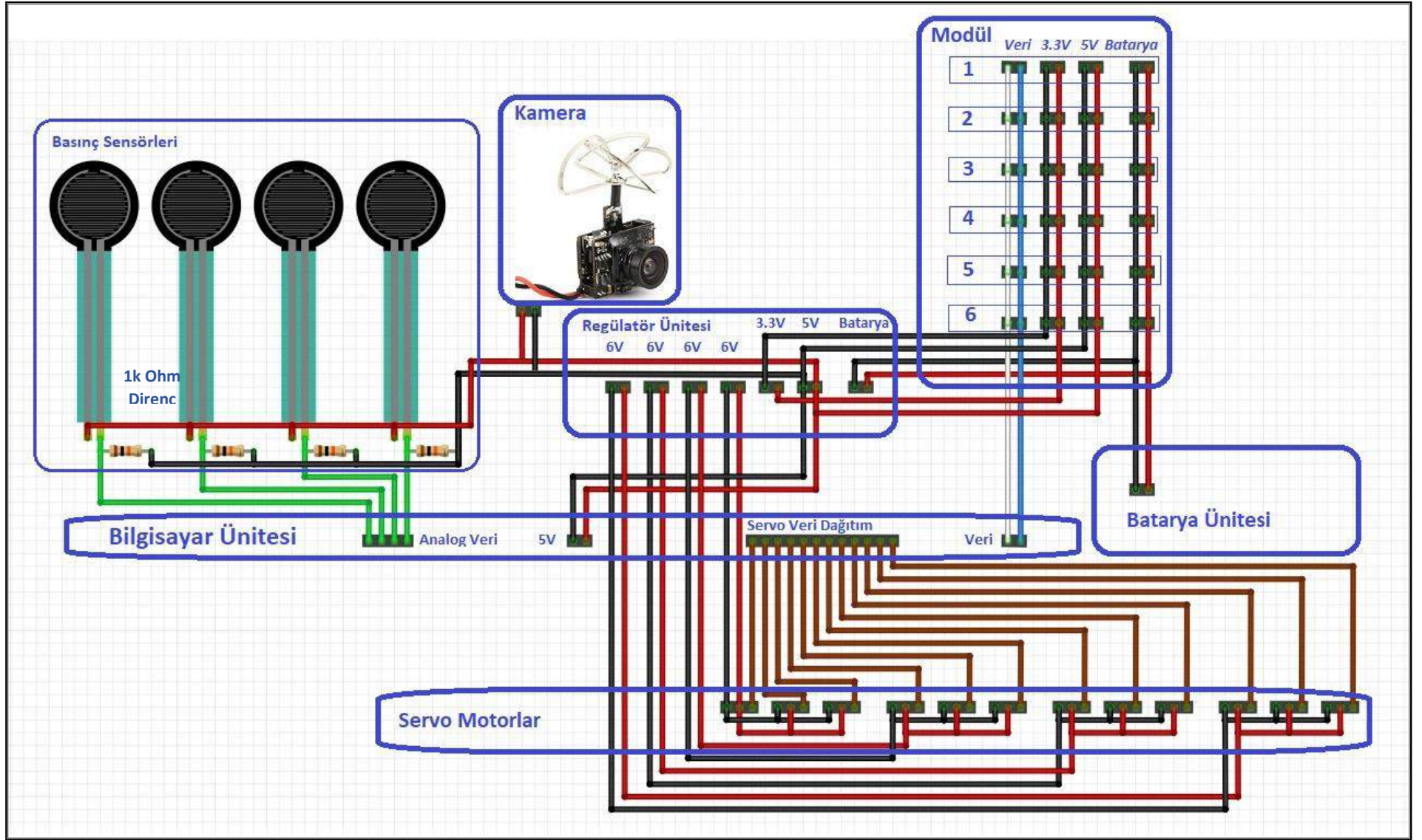


Matematiksel modellerin doğruluğunu temin için bacak alt çubuğunun dirsek eklemine bağlantısı da eşit oranda yukarı kaydırılmış ve alttan bir destek ile dengelenmiştir. Bu sayede bacak alt çubuğu ile bacak üst çubuğunun eşit uzunluklarını korunmuştur. Bacak hareketinde, ayağın yere basışının yumuşatılması için monte edilebilir bir yaylı esnek taban tasarlanmıştır. Bu sayede yere basma esnasında oluşan darbe etkisi azaltılmış ve daha yumuşak bir hareket elde edilmiştir. Ayrıca ayağın bacak ile birleştiği topuk bölgesinde, engel aşma, tutma, çekme ve geriye itme hareketlerini gerçekleştirebilmeyi sağlayan kullanışlı bir alan oluşturulmuştur.

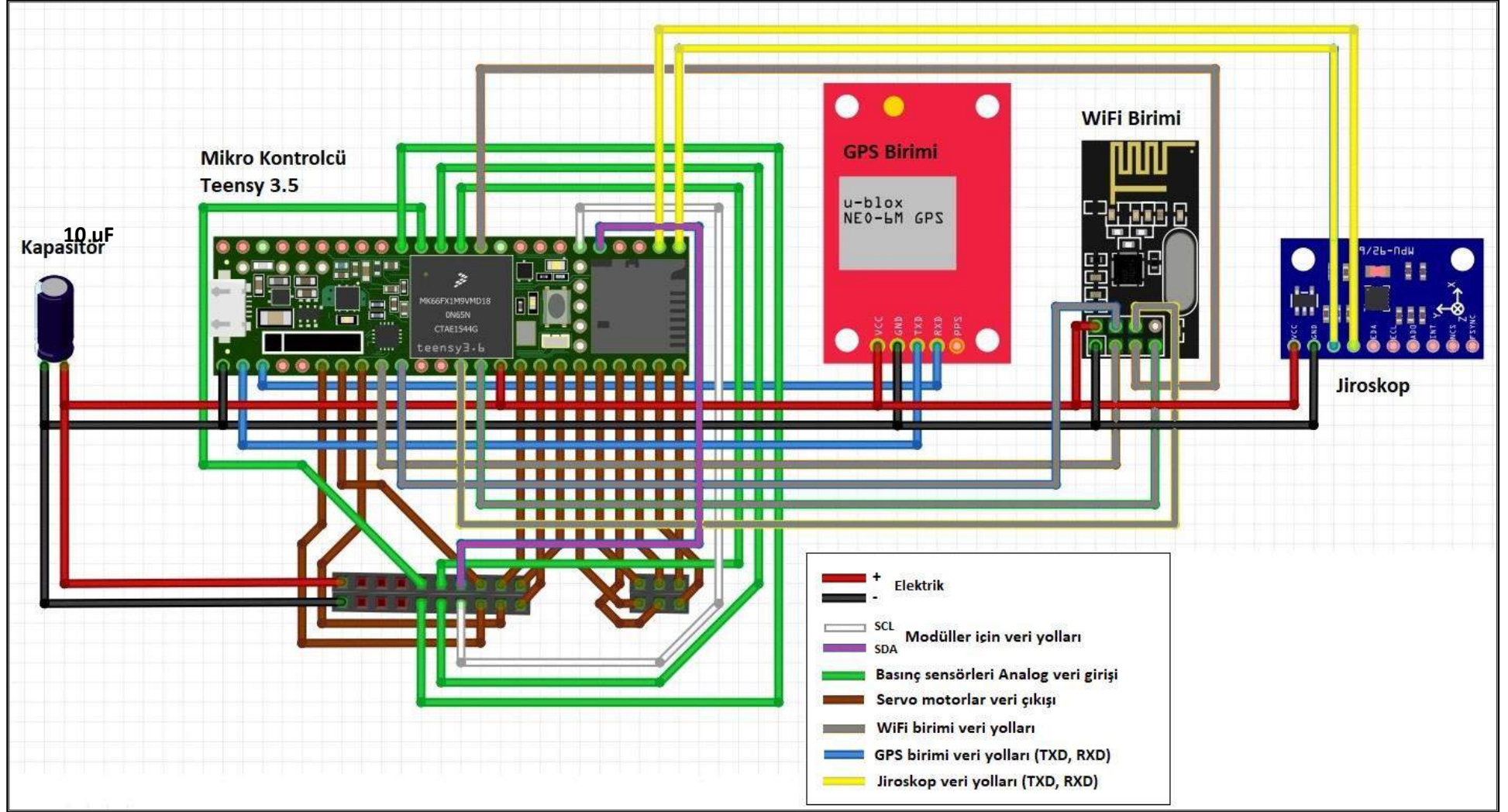
B. Elektrik ve Veri Dağıtım Katmanı kablolama



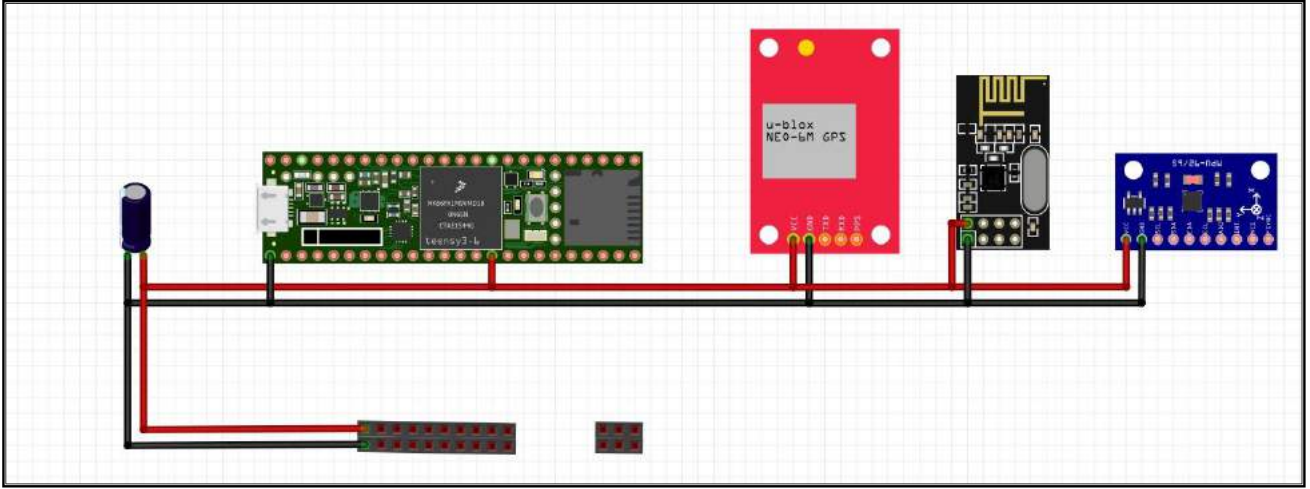
Ek 16 Elektrik ve Veri Dağıtım Genel Şeması



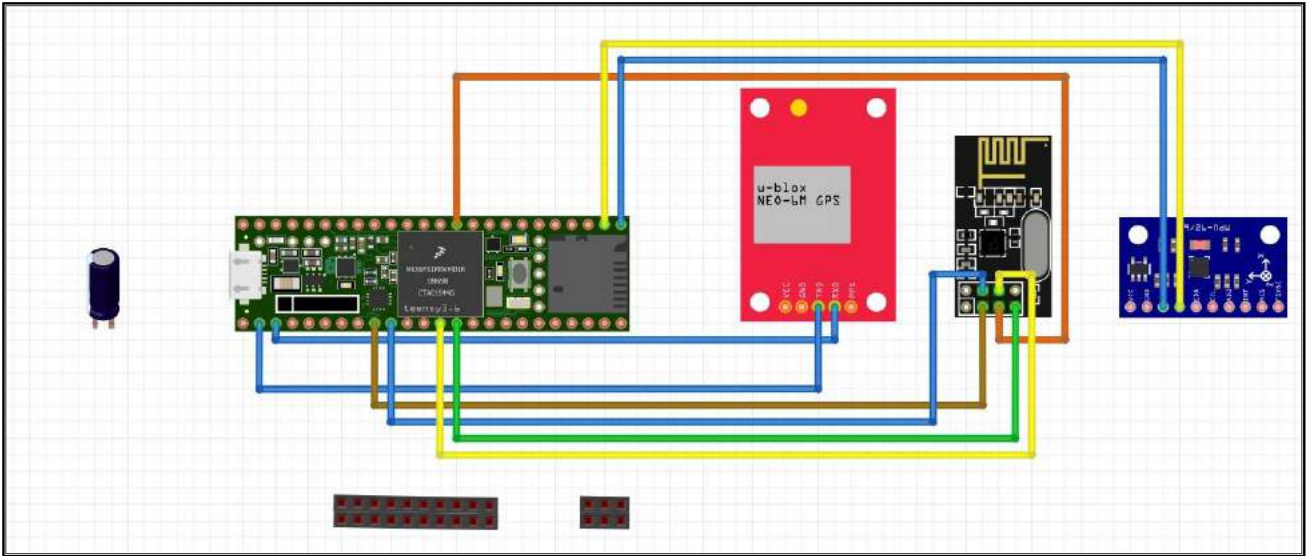
Ek 17 Bilgisayar Ünitesi Elektrik ve Veri Dağıtım Şeması



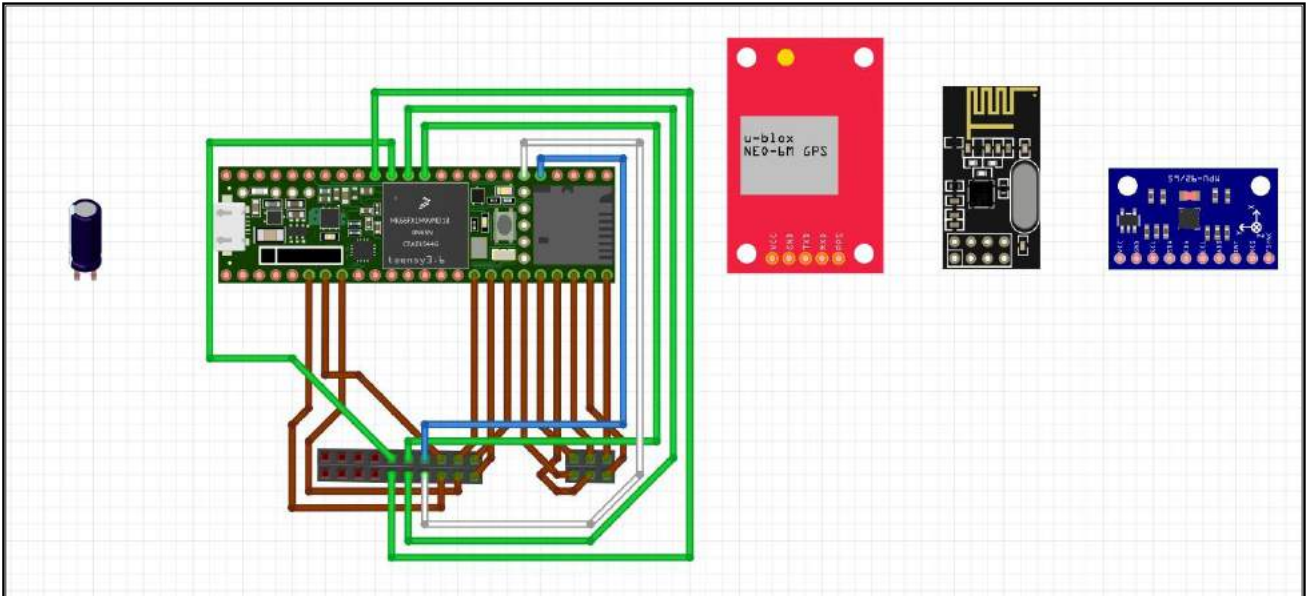
Bilgisayar Ünitesi Elektrik Dağıtım Şeması



Bilgisayar Ünitesi Birimler Veri Dağıtım Şeması



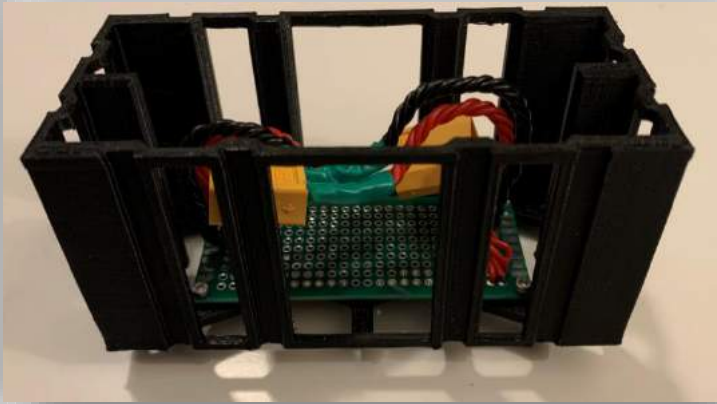
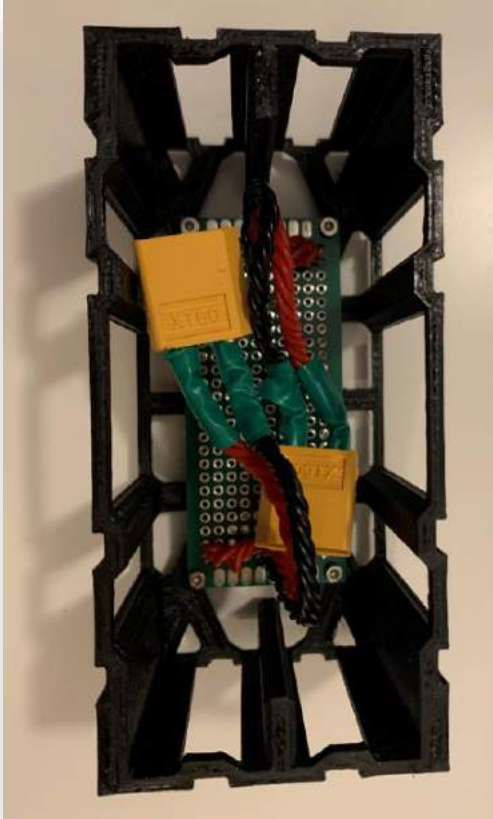
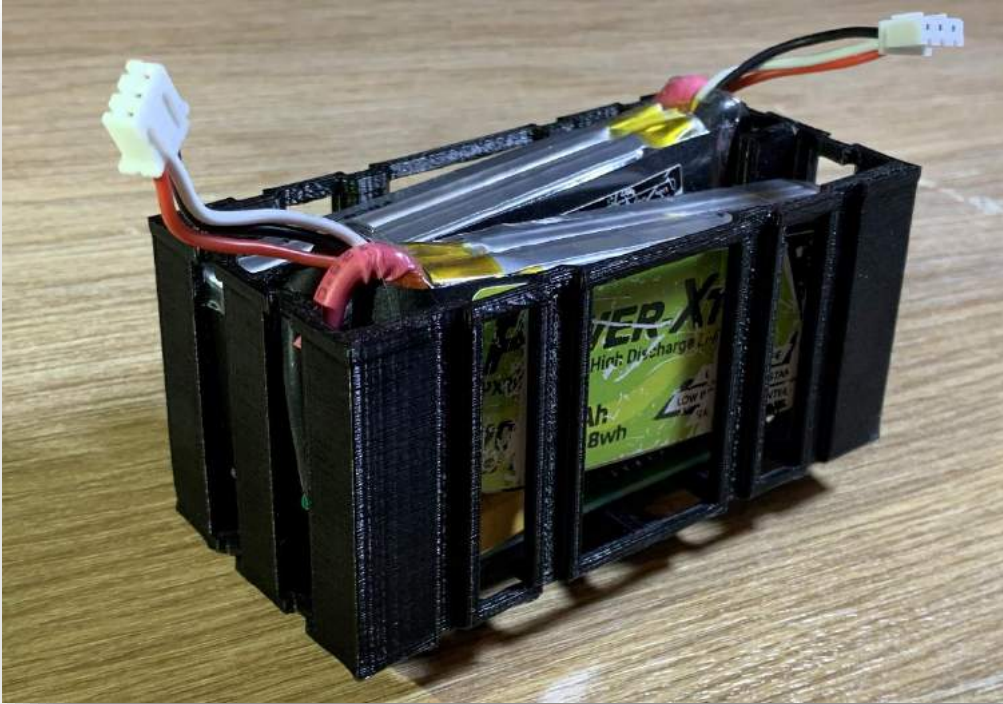
Bilgisayar Ünitesi Konnektör Veri Yolları Şeması



Ek 19 Batarya Ünitesi Malzeme Listesi

BATARYA ÜNİTESİ MALZEME LİSTESİ			
		Malzeme	Adet
1		7.4V 2S1P 2000 mAh (30C) Li-Polymer Pil - Soketli	2
2		XT60 Plug 60A Li-Po Konnektör Takım	2
3		20x30 Delikli Plaket	1
4		Filament 1.75 mm Siyah PLA - ABG	1
5		Dişi Pin Header 2.54mm	2
6		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
7		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
8		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	4

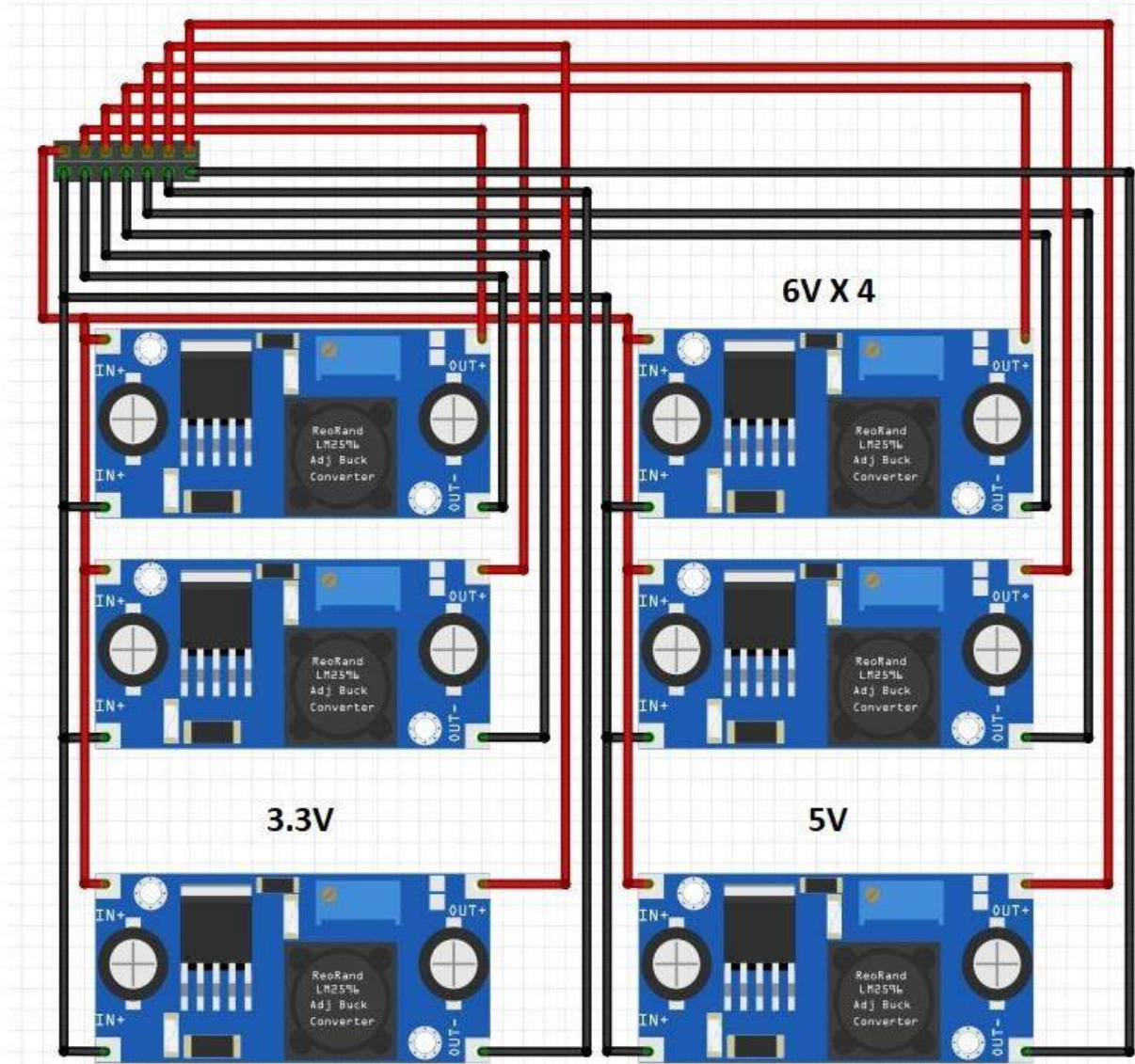
Ek 20 Batarya Ünitesi



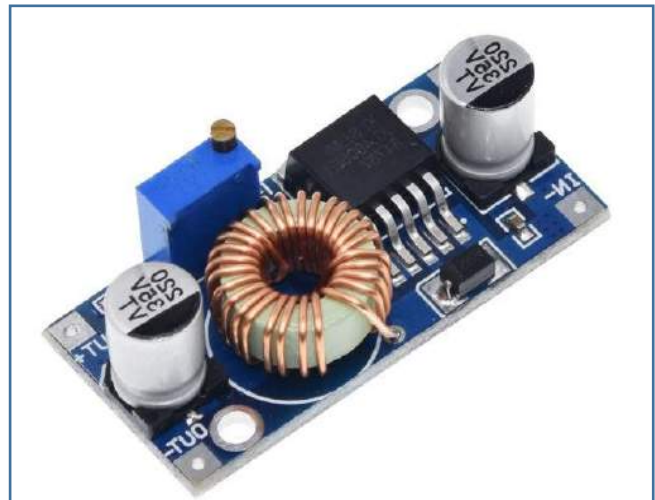
Ek 21 Regülatör Ünitesi Parça Listesi

REGÜLATÖR ÜNİTESİ MALZEME LİSTESİ			
		Malzeme	Adet
1		DC/DC - AC/DC Converter	4
2		LM2596 Ayarlanabilir Voltaj Düşürücü Güç Modülü (4-35V Giriş - 1-30V Çıkış)	2
3		2.5 V - 30V Mini Dijital Kırmızı Voltmetre	1
4		20x30 Delikli Plaket	1
5		Filament 1.75 mm Siyah PLA - ABG	1
6		Dişi Pin Header 2.54mm	1
7		12mm Erkek Header	1
8		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
9		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1

Ek 22 Regülatör Ünitesi Elektrik Şeması

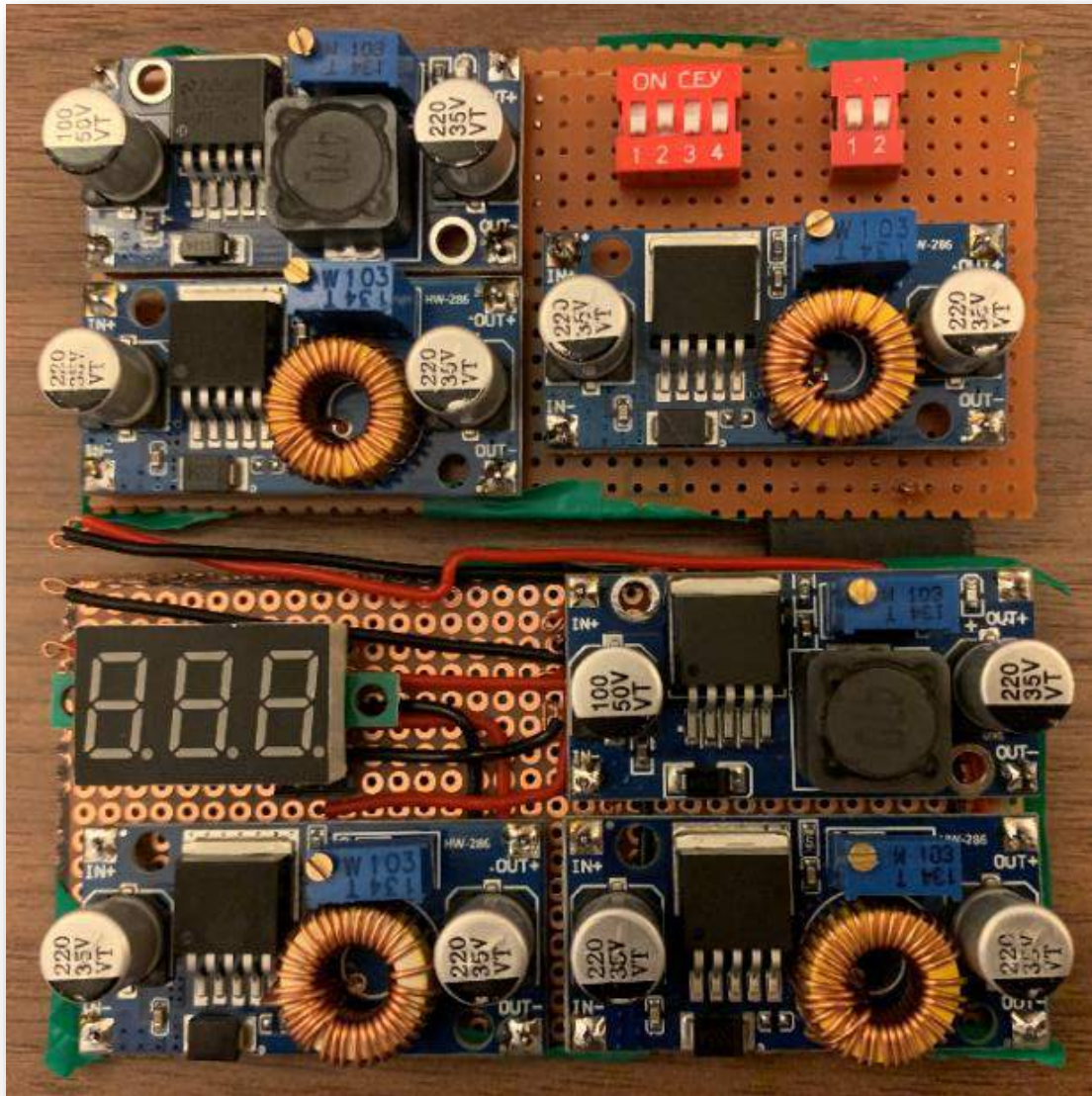
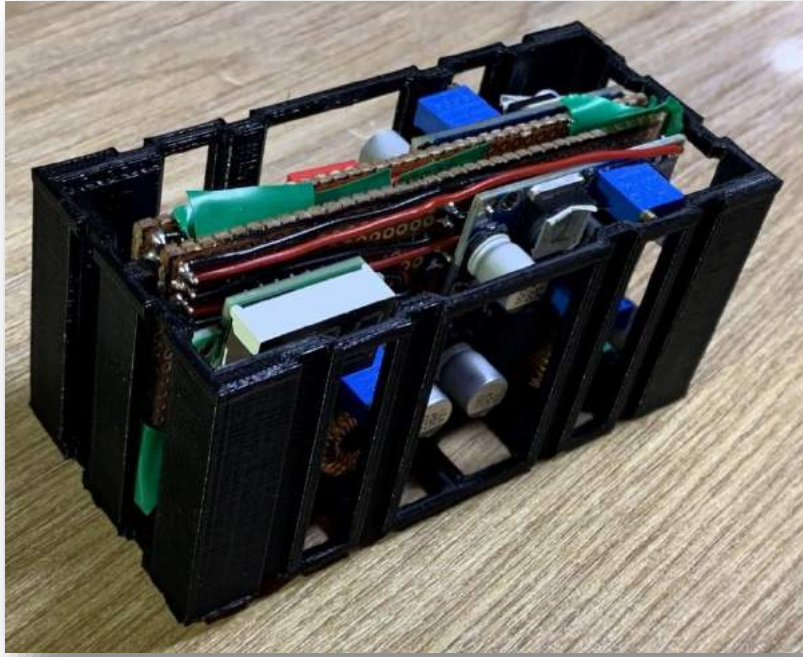


LM2596 DC-DC Çevirici (4-35V/ 1-30V)
(2 Adet)



XL4005 DC/DC Çevirici (5-32V / 1.25-32V)
(4 Adet)

Ek 23 Regülatör Ünitesi



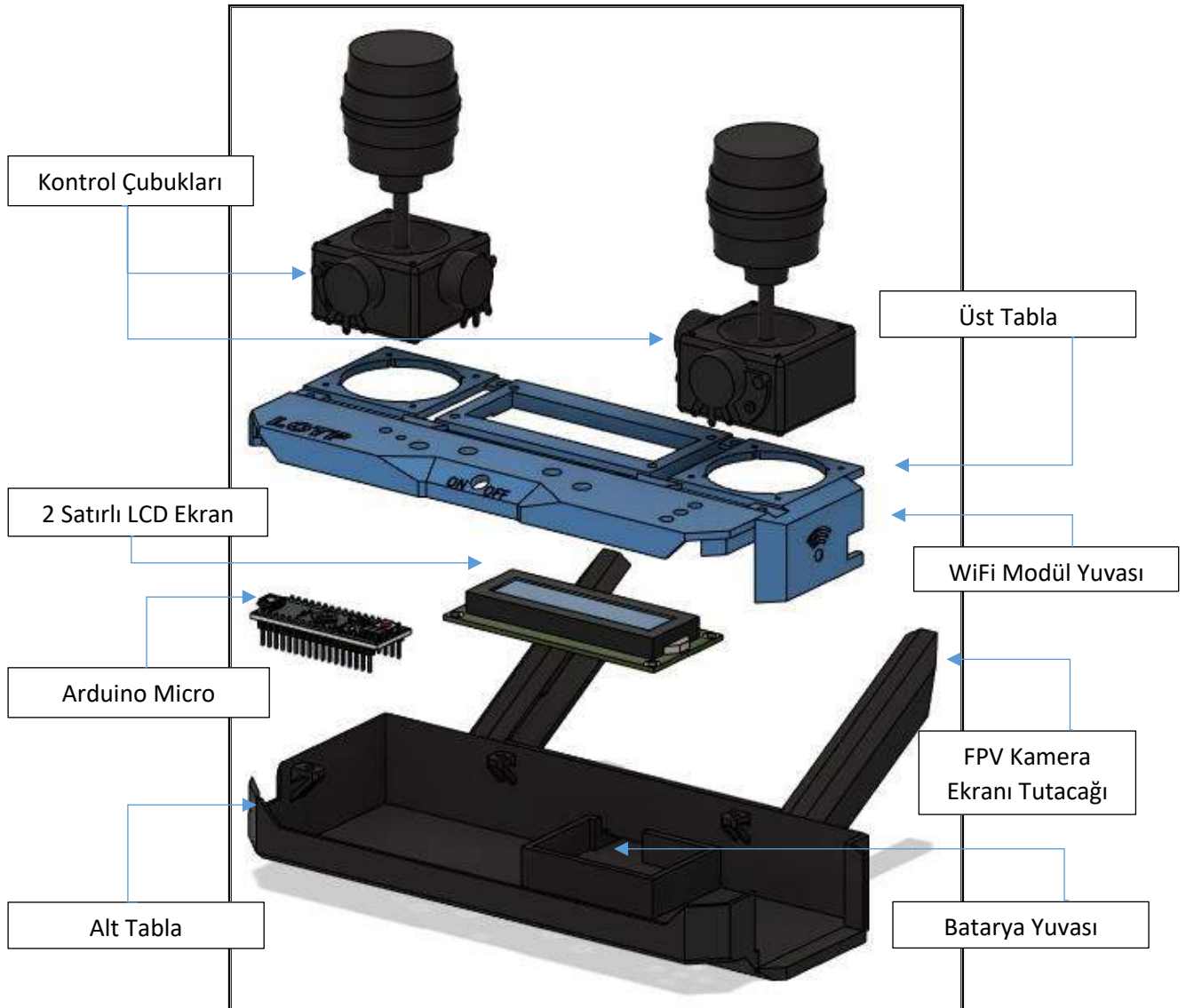
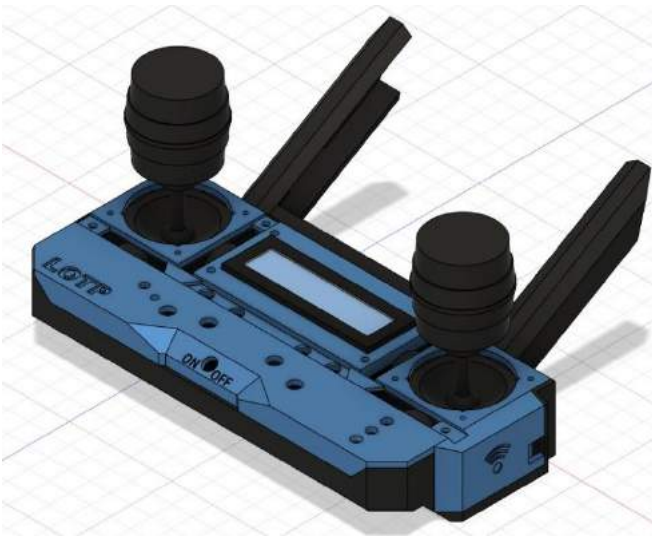
Ek 24 Bilgisayar Ünitesi Malzeme Listesi

BİLGİSAYAR ÜNİTESİ MALZEME LİSTESİ			
		Malzeme	Adet
1		Teensy 3.5	1
2		Neo-7m Arduino Shield Mini Gps Modülü	1
3		Seramik Kablolulu 28dB Yüksek Kazançlı Aktif GPS Anteni	1
4		NRF24L01+PA+LNA SMA 2.4G Kablosuz Wifi Modül	1
5		MPU6050 6 Eksen İvme ve Gyro Sensörü gy-521	1
6		Logic Level Converter (3.3 V-5 V)	1
7		100uF 63v Kondansatör	1
8		20x30 Delikli Plaket	1
9		Filament 1.75 mm Siyah PLA - ABG	1
10		Dişi Pin Header 2.54mm	7
11		12mm Erkek Header	5
12		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
13		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
14		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	2
15		10 uF Kapasitör	1

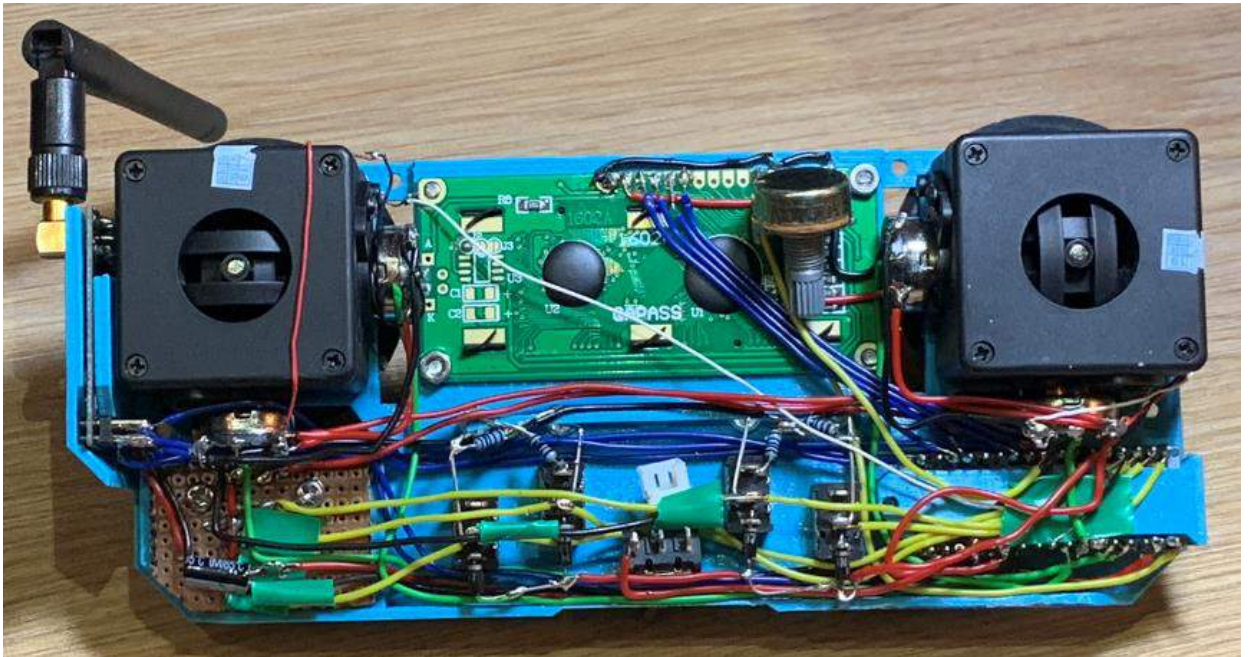
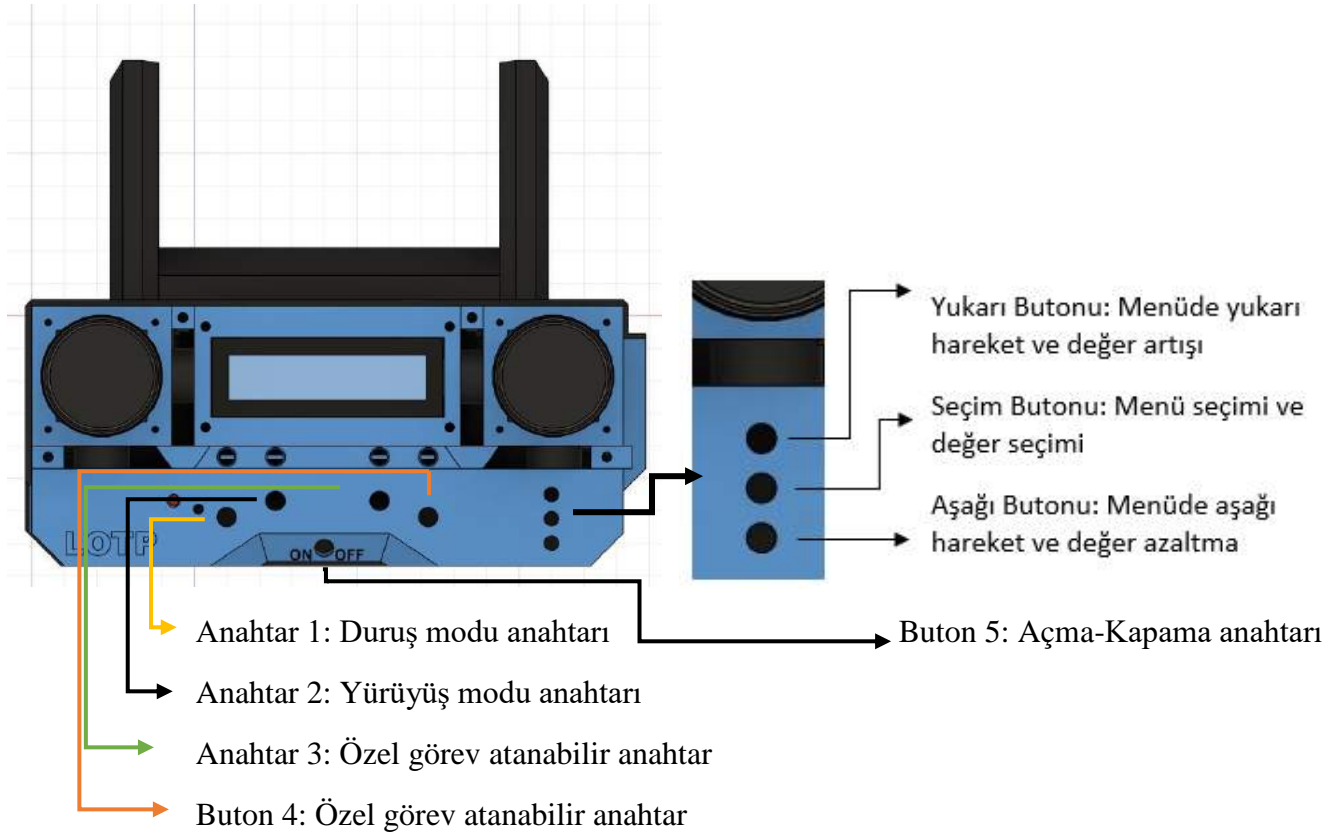
Ek 25 Mikro Kontrolcü Bilgisayar Ünitesi



Ek 26 Kontrol Kumandası Tasarımı



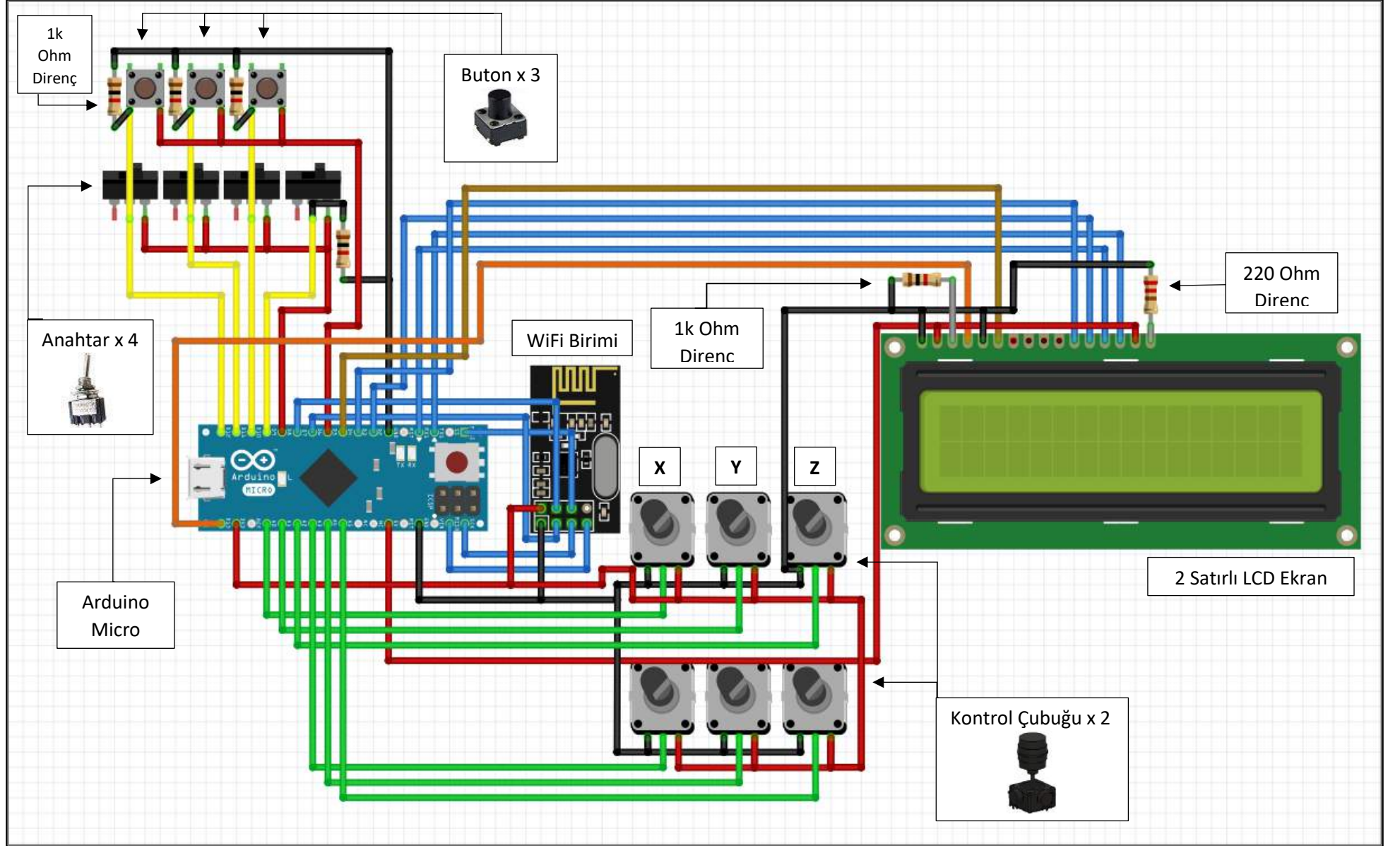
Ek 27 Kontrol Kumandası Tuşlar ve Elektrik Veri Dağıtım Yapısı



Ek 28 Kontrol Kumandası Malzeme Listesi

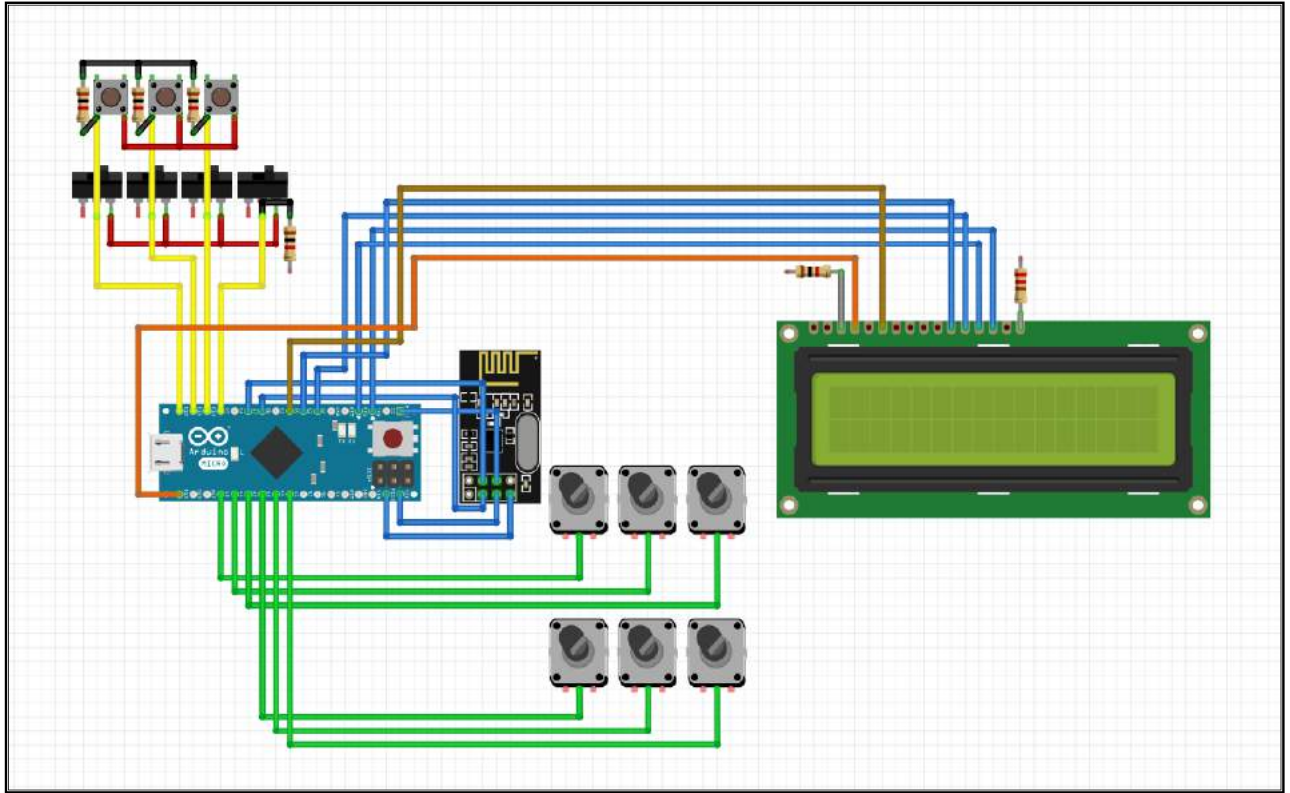
KONTROL KUMANDASI MALZEME LİSTESİ			
		Malzeme	Adet
1		Pro 3 Eksen Joystick (Butonsuz)	2
2		TP4056 3.7 V 1S Lipo Şarj Aleti - 5 V 1 A Lithium Battery Charger	1
3		KTS102 On / Off 3 Ayak Toggle Anahtar	5
4		6x6 12mm Tach Buton (4 Bacak)	3
5		NRF24L01+PA+LNA SMA 2.4G Kablosuz Wifi Modül 1 km Uzun Mesafe	1
6		5.8G 48CH 4.3 Inch LCD 480x272 16:9 NTSC/PAL FPV Monitor	1
7		2x16 LCD Ekran Sol Üst Mavi - Qapass	1
8		Arduino Micro (Klon)	1
9		3.7V 590 mAh Li-Polymer Pil	1
10		5V Voltaj Yükseltici Regülatörü 0.5/5.5V Giriş	1
11		5mm Şeffaf Beyaz Led - 4000-6000 mcd	4
12		Filament 1.75 mm Turkuaz PLA - ABG	1
13		Filament 1.75 mm Siyah PLA - ABG	1
14		Tek Damarlı Montaj Kablosu - 22 AWG - 15 Metre Mavi	1
15		Tek Damarlı Montaj Kablosu - 22 AWG - 15 Metre Kırmızı	1
16		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	7
17		M3 Somun	4
18		1k Ohm Direnç	5
19		200 Ohm Direnç	1

Ek 29 Kontrol Kumandası Elektrik ve Veri Dağıtım Şeması

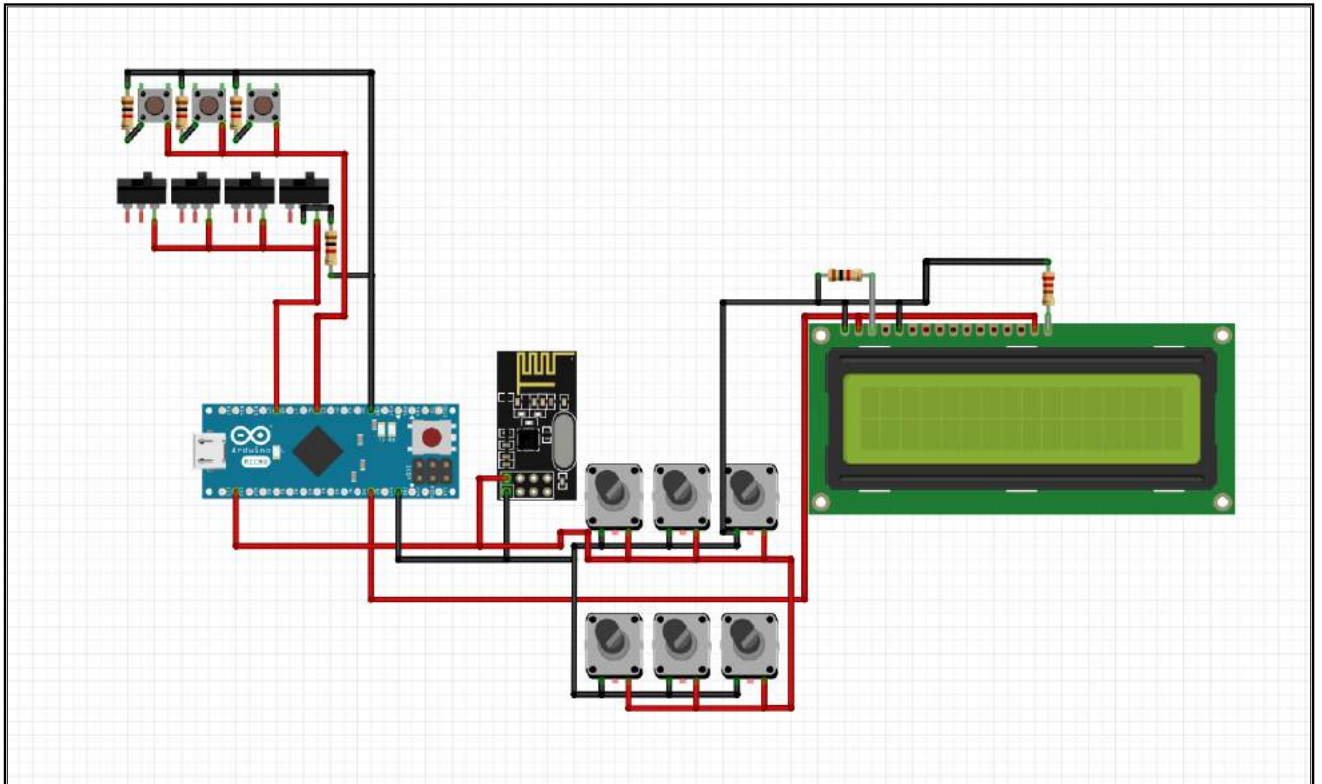


Ek 30

Kontrol Kumandası Veri Dağıtım Şeması

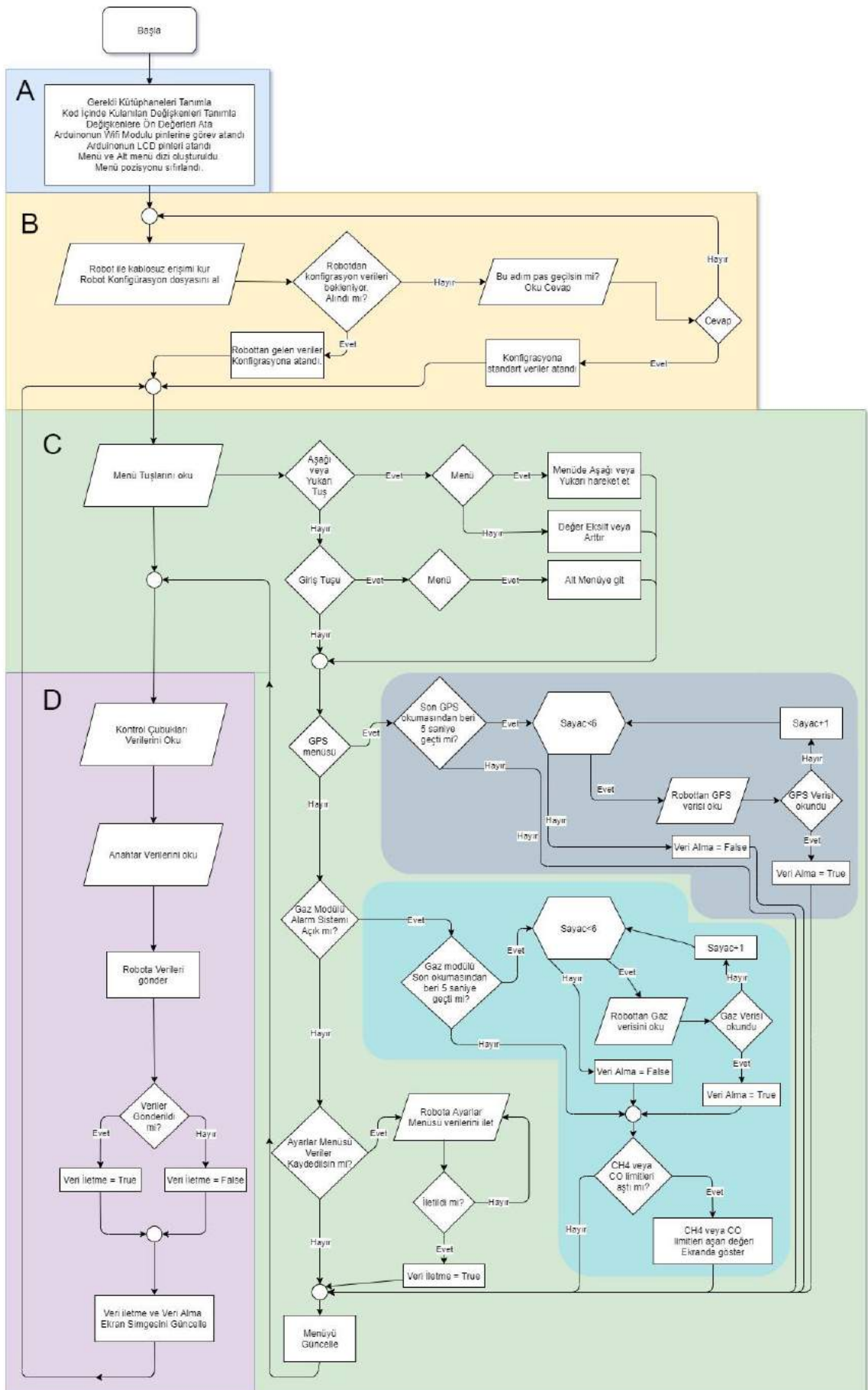


Kontrol Kumandası Elektrik Dağıtım Şeması



Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(1)



(2)

```

1  #include <LiquidCrystal.h>
2  #include <SPI.h>
3  #include <nRF24L01.h>
4  #include <RF24.h>
5
6  RF24 radio(6, 7); // CE, CSN
7  LiquidCrystal lcd(13, 4, 3, 2, 0, 1);
8
9  int joystickZero[6] = {10,10,10,10,10,10};
10
11 char joypack[24];
12 int menuCursor = 0;
13 int BinMode = 5;
14 int SinMode = 8;
15 int Switches[4];
16 int oldSwitches[4];
17 int switchPins[4] = {12,11,10,9};
18 bool Bnew[3] = {1,1,1};
19 bool bEnter = 0;
20 int menuLoc[3] = {0,0,0};
21 int buttonPins[3] = {11,10,9};
22 int Cstat = 2;
23 int CurserX = 9;
24 int menuLength = 4;
25 bool tStatus = 0;
26 bool rStatus = 0;
27 bool recive = 0;
28 bool doOnes[2] = {0,0};
29
30 char configPackage[80];
31 int packageCharSize = 5;
32
33 String menu[4] = {"-GPS      ", "-Modules", "-Settings", "-About  "};
34 String menuGPS[4] = {"< Back", "-Location", "-Time  ", "      "};
35 String menuSettings[8] = {"< Back      ", "Restore^  ", "J Smoth   ", "Path      ",
36   "Step H    ", "gAssist   ", "fAssist   ", "Walk Mode "};
37 String menuModules[4] = {"< Back", "Lidar", "Drone ", "DGM  "};
38 String menuLidar[4] = {"< Back      ", "Turn Off", "Switch None", "      "};
39 String menuDrone[4] = {"< Back      ", "Turn Off", "Switch None", "      "};
40 String menuGas[4] = {"< Back      ", "Alert Off", "Gas Variables", "      "};
41
42 String menuAbout[6] = {"< Back  ", "LOTP RoboDog ", "Prototype", "V2      ",
43   "Made By :", "Halid Yildirim"};
44
45 const uint64_t tAddress = 0xE8E8F0F0E11L;
46 const uint64_t rAddress = 0xE8E8F0F0E0LL;
47 const uint64_t tDrone = 0xE8E8F0F1E11L;
48
49 unsigned long pTime = millis();
50 unsigned long cTime = millis();
51 unsigned long dTime = 1000;
52
53 byte conChar[8] = {
54   B00000,
55   B01110,
56   B10001,
57   B00100,
58   B01010,
59   B00000,
60   B00100,
61 };
62
63 byte disconChar[8] = {
64   B00000,
65   B00000,
66   B00000,
67   B00000,
68   B00000,
69   B00000,
70   B00100,

```


Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(3)

```
71 };
72
73 byte blank[8] = {
74     B00000,
75     B00000,
76     B00000,
77     B00000,
78     B00000,
79     B00000,
80     B00000,
81 };
82
83 struct Package{
84     int16_t joystick[6];
85     bool gpsRequest;
86     bool gasRequest;
87
88     bool standMode;
89     bool walkMode;
90
91     bool Gps;
92     bool Lidar;
93     bool Drone;
94     bool Gas;
95
96     bool restoreConfig;
97
98     byte joystickSmoothness;
99     byte path;
100    byte stepH;
101    bool gyroAssist;
102    bool forceAssist;
103    bool WalkMode;
104
105    byte lidarSwitch;
106    byte droneSwitch;
107 };
108
109
110 struct GPSPackage{
111     byte staller;
112
113     byte hour;
114     byte minute;
115
116     int16_t year;
117     byte month;
118     byte day;
119
120     float longti;
121     float lati;
122 };
123
124 struct GasPackage{
125     float H2;
126     float LPG;
127     float CH4;
128     float CO;
129     float Alchol;
130 };
131
132 struct Config{
133     byte joystickSmoothness;
134     byte path;
135     byte stepH;
136     bool gyroAssist;
137     bool forceAssist;
```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(4)

```
138     byte lidarSwitch;
139     byte droneSwitch;
140 };
141
142 Package package;
143
144 GPSPackage gpsPackage;
145
146 GasPackage gasPackage;
147
148 Config config;
149
150 void setup(){
151     Serial.begin(9600);
152
153     pinMode(SinMode, OUTPUT);
154     pinMode(BinMode, OUTPUT);
155
156     for(int i = 0; i < 4; i++){
157         pinMode(switchPins[i], INPUT);
158     }
159
160     lcd.begin(16, 2);
161     lcd.createChar(1,disconChar);
162     lcd.createChar(2,conChar);
163     lcd.createChar(0,blank);
164
165     radio.begin();
166     radio.openWritingPipe(tAddress);
167     radio.openReadingPipe(1,rAddress);
168     radio.setPALevel(RF24_PA_MIN);
169
170     lcd.setCursor(0, 0);
171     lcd.print("LOTP RoboDog");
172     lcd.setCursor(1, 1);
173     lcd.print("V2");
174
175     delay(1500);
176     lcd.clear();
177
178     lcd.setCursor(0, 0);
179     lcd.print("Searching for");
180     lcd.setCursor(0, 1);
181     lcd.print("config   Skip <");
182
183     delay(5);
184     digitalWrite(BinMode, 1);
```

(5)

```
185 radio.startListening();
186 while(!radio.available() and !digitalRead(buttonPins[1])){}
187 for(int i = 0; i < 5; i++){
188     if(radio.available()){
189         lcd.clear();
190         radio.read(&config, sizeof(config));
191         lcd.setCursor(0, 0);
192         lcd.print("Config Recived");
193         setConfig();
194         delay(750);
195         i = 5;
196     }
197     else{
198         setConfigDefault();
199         Bnew[1] = 0;
200     }
201 }
202 if(package.gyroAssist){
203     menuSettings[5] = "gAssist On ";
204 }
205 else{
206     menuSettings[5] = "gAssist Off ";
207 }
208 if(package.forceAssist){
209     menuSettings[6] = "fAssist On ";
210 }
211 else{
212     menuSettings[6] = "fAssist Off ";
213 }
214 if(package.WalkMode){
215     menuSettings[7] = "Walk Mode 4s";
216 }
217 else{
218     menuSettings[7] = "Walk Mode 2s";
219 }
220
221 radio.stopListening();
222 digitalWrite(BinMode, 0);
223 delay(50);
224
225
226 lcd.clear();
227 }
```


(6)

```
229 void loop(){
230     cTime = millis();
231     Serial.println(package.WalkMode);
232
233     package.restoreConfig = 0;
234
235     lcd.setCursor(CurserX, menuCursor%2);
236     lcd.write(byte(0));
237
238     digitalWrite(BinMode, 1);
239     //Up Button
240     if(digitalRead(buttonPins[0]) and menuCursor > 0 and Bnew[0]){
241         menuCursor--;
242         Bnew[0] = 0;
243     }
244     else if(!digitalRead(buttonPins[0])){
245         Bnew[0] = 1;
246     }
247
248     //Down Button
249     if(digitalRead(buttonPins[2]) and menuCursor < menuLength-1 and Bnew[2]){
250         menuCursor++;
251         Bnew[2] = 0;
252     }
253     else if(!digitalRead(buttonPins[2])){
254         Bnew[2] = 1;
255     }
256
257     //Enter Button
258     if(digitalRead(buttonPins[1]) and Bnew[1]){
259         Bnew[1] = 0;
260         bEnter = 1;
261     }
262     else if(!digitalRead(buttonPins[1]) and !Bnew[1]){
263         lcd.clear();
264         updateConnectionStat();
265         Bnew[1] = 1;
266         bEnter = 0;
267     }
268     digitalWrite(BinMode, 0);
269
270     if(menuLoc[0] == 0){
271         CurserX = 9;
272         mainMenu();
273         menuLength = 4;
274     }
275
276     if(menuLoc[0] == 1){
277         package.Gps = 1;
278         recive = 1;
279         if(menuLoc[1] == 0){
280             CurserX = 9;
281             menuLength = 3;
282             GPSTMenu();
283         }
284
285         if(menuLoc[1] == 1){
286             CurserX = 13;
287             LocMenu();
288         }
289         if(menuLoc[1] == 2){
290             CurserX = 13;
291             TimeMenu();
292         }
293
294         if(cTime - pTime > dTime){
295             if(cTime - pTime < dTime + 10){
296                 rStatus = 0 ;
297             }

```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(7)

```
298     package.gpsRequest = 1;
299     if(cTime - pTime > dTime + 100){
300         pTime = cTime;
301     }
302 }
303 else{
304     package.gpsRequest = 0;
305 }
306 }
307 else{
308     rStatus = 0;
309     package.Gps = 0;
310 }
311
312 if(menuLoc[0] == 2){
313     if(menuLoc[1] == 0){
314         CurserX = 10;
315         menuLength = 4;
316         modulesMenu();
317     }
318
319     if(menuLoc[1] == 1){
320         CurserX = 11;
321         menuLength = 3;
322         lidarMenu();
323     }
324     if(menuLoc[1] == 2){
325         CurserX = 11;
326         menuLength = 3;
327         droneMenu();
328     }
329     if(menuLoc[1] == 3){
330         CurserX = 13;
331         menuLength = 3;
332         if(menuLoc[2] == 0){
333             gasMenu();
334         }
335         if(menuLoc[2] == 1){
336             gasVarMenu();
337         }
338     }
339 }
340
341 }
342
343 if(package.Gas){
344     gasAlert();
345     if(cTime - pTime > dTime){
346         if(cTime - pTime < dTime + 10){
347             rStatus = 0 ;
348         }
349         package.gasRequest = 1;
350         if(cTime - pTime > dTime + 100){
351             pTime = cTime;
352         }
353     }
354     else{
355         package.gasRequest = 0;
356     }
357 }
358
359 if(menuLoc[0] == 3){
360
361     if(menuLoc[1] == 0){
362         CurserX = 12;
363         menuLength = 8;
364
365         settingsMenu();
366     }
367     if(menuLoc[1] == 1){
```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(8)

```
368 RestoreConfig();
369 menuLoc[1] = 0;
370 menuCursor = 1;
371 bEnter = 0;
372 }
373
374 if(menuLoc[1] == 2){
375     menuLength = 512;
376     if(doOnes[0] == 0){
377         doOnes[0] = 1;
378         menuCursor = -package.joystickSmoothness + 256;
379     }
380     lcd.setCursor(7, 0);
381     lcd.print(" ");
382     lcd.setCursor(7, 0);
383     lcd.print(-menuCursor + 256);
384     if(bEnter == 1){
385         lcd.setCursor(7, 0);
386         lcd.print(" ");
387         package.joystickSmoothness = -menuCursor + 256;
388         menuLoc[1] = 0;
389         menuCursor = 2;
390         bEnter = 0;
391         doOnes[0] = 0;
392     }
393 }
394 if(menuLoc[1] == 3){
395     menuLength = 512;
396     if(doOnes[0] == 0){
397         doOnes[0] = 1;
398         menuCursor = -package.path + 256;
399     }
400     lcd.setCursor(7, 0);
401     lcd.print(" ");
402     lcd.setCursor(7, 0);
403     lcd.print(-menuCursor + 256);
404     if(bEnter == 1){
405         lcd.setCursor(7, 0);
406         lcd.print(" ");
407         package.path = -menuCursor + 256;
408         menuLoc[1] = 0;
409         menuCursor = 3;
410         bEnter = 0;
411         doOnes[0] = 0;
412     }
413 }
414 if(menuLoc[1] == 4){
415     menuLength = 512;
416     if(doOnes[0] == 0){
417         doOnes[0] = 1;
418         menuCursor = -package.stepH + 256;
419     }
420     lcd.setCursor(7, 0);
421     lcd.print(" ");
422     lcd.setCursor(7, 0);
423     lcd.print(-menuCursor + 256);
424     if(bEnter == 1){
425         lcd.setCursor(7, 0);
426         lcd.print(" ");
427         package.stepH = -menuCursor + 256;
428         menuLoc[1] = 0;
429         menuCursor = 4;
430         bEnter = 0;
431         doOnes[0] = 0;
432     }
433 }
434 if(menuLoc[1] == 5){
435     menuLength = 1;
```


Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(9)

```
436     if(package.gyroAssist){
437         package.gyroAssist = 0;
438         menuSettings[5] = "gAssist Off ";
439     }
440     else{
441         package.gyroAssist = 1;
442         menuSettings[5] = "gAssist On  ";
443     }
444     menuLoc[1] = 0;
445     menuCursor = 5;
446     bEnter = 0;
447     doOnes[0] = 0;
448 }
449 if(menuLoc[1] == 6){
450     menuLength = 1;
451     if(package.forceAssist){
452         package.forceAssist = 0;
453         menuSettings[6] = "fAssist Off ";
454     }
455     else{
456         package.forceAssist = 1;
457         menuSettings[6] = "fAssist On  ";
458     }
459     menuLoc[1] = 0;
460     menuCursor = 6;
461     bEnter = 0;
462     doOnes[0] = 0;
463 }
464 if(menuLoc[1] == 7){
465     menuLength = 1;
466     if(package.WalkMode){
467         package.WalkMode = 0;
468         menuSettings[7] = "Walk Mode 2s";
469     }
470     else{
471         package.WalkMode = 1;
472         menuSettings[7] = "Walk Mode 4s";
473     }
474     menuLoc[1] = 0;
475     menuCursor = 7;
476     bEnter = 0;
477     doOnes[0] = 0;
478 }
479
480
481 }
482
483 if(menuLoc[0] == 4){
484     CurserX = 12;
485     menuLength = 5;
486
487     aboutMenu();
488 }
489
```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(10)

D

```
490 //Get joystick and add in to Package
491 zipJoystick();
492
493 //Get Switch mode and add in to Package
494 getSwitch();
495
496 package.standMode = Switches[0];
497 package.walkMode = Switches[1];
498
499 if(package.lidarSwitch == 3 and oldSwitches[2] != Switches[2]){
500     package.Lidar = Switches[2];
501 }
502 if(package.lidarSwitch == 4 and oldSwitches[3] != Switches[3]){
503     package.Lidar = Switches[3];
504 }
505
506 if(package.droneSwitch == 3 and oldSwitches[2] != Switches[2]){
507     package.Drone = Switches[2];
508 }
509 if(package.droneSwitch == 4 and oldSwitches[3] != Switches[3]){
510     package.Drone = Switches[3];
511 }
512
513 for(int i = 0; i < 4; i++){
514     oldSwitches[i] = Switches[i];
515 }
516
517 //Send Package and check for connection
518 radio.stopListening();
519 tStatus = radio.write(&package, sizeof(package));
520 if(package.gpsRequest){
521     getGpsPackage();
522 }
523 if(package.gasRequest){
524     getGasPackage();
525 }
526 updateConnectionStat();
527 }
528
```

(11)

```

529 void zipJoystick(){
530     int joystick[6] = {analogRead(A0),analogRead(A1),analogRead(A2),
531                       analogRead(A3),analogRead(A4),analogRead(A5)};
532
533     for(int i = 0; i < 6; i++){
534         if(abs(joystick[i] - 512) < joystickZero[i]){
535             joystick[i] = 512;
536         }
537         package.joystick[i] = joystick[i];
538     }
539 }
540
541 void getSwitch(){
542     digitalWrite(SinMode, 1);
543     for(int i = 0; i < 4; i++){
544         Switches[i] = digitalRead(switchPins[i]);
545     }
546     digitalWrite(SinMode, 0);
547 }
548
549 void updateConnectionStat(){
550     lcd.setCursor(14,0);
551     lcd.write("T");
552     lcd.setCursor(15,0);
553     if(tStatus){
554         lcd.write(byte(2));
555     }
556     else{
557         lcd.write(byte(1));
558     }
559     lcd.setCursor(14,1);
560     lcd.write("R");
561     lcd.setCursor(15,1);
562     if(rStatus){
563         lcd.write(byte(2));
564     }
565     else{
566         lcd.write(byte(1));
567     }
568 }
569
570 void mainMenu(){
571     lcd.setCursor(CurserX, menuCursor%2);
572     lcd.write(char(60));
573     lcd.setCursor(0, 0);
574     lcd.print(menu[int(menuCursor/2)*2]);
575     lcd.setCursor(0, 1);
576     lcd.print(menu[int(menuCursor/2)*2+1]);
577
578     if(bEnter == 1){
579         menuLoc[0] = menuCursor + 1;
580         menuCursor = 0;
581         bEnter = 0;
582     }
583 }
584
585 void GPSMenu(){
586     lcd.setCursor(CurserX, menuCursor%2);
587     lcd.write(char(60));
588     lcd.setCursor(0, 0);
589     lcd.print(menuGPS[int(menuCursor/2)*2]);
590     lcd.setCursor(0, 1);
591     lcd.print(menuGPS[int(menuCursor/2)*2+1]);
592
593     if(bEnter == 1 and menuCursor == 0){
594         menuLoc[0] = 0;
595         menuLoc[1] = 0;
596         menuLoc[2] = 0;
597         menuCursor = 0;
598         bEnter = 0;
599     }

```


Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(12)

```
600     else if(bEnter == 1){
601         menuLoc[1] = menuCursor;
602         menuCursor = 0;
603         bEnter = 0;
604     }
605 }
606
607 void LocMenu(){
608     lcd.setCursor(0, 0);
609     lcd.print("Lati");
610     for(int i = 0; i < 8; i++){
611         lcd.setCursor(i + 5, 0);
612         lcd.print(gpsPackage.lati);
613     }
614
615     lcd.setCursor(0, 1);
616     lcd.print("Long");
617     for(int i = 0; i < 8; i++){
618         lcd.setCursor(i + 5, 1);
619         lcd.print(gpsPackage.longti);
620     }
621
622     if(bEnter == 1){
623         menuLoc[1] = 0;
624         menuCursor = 1;
625         bEnter = 0;
626     }
627 }
628
629 void TimeMenu(){
630     lcd.setCursor(0, 0);
631     lcd.print("Time");
632
633     lcd.setCursor(7, 0);
634     lcd.print(gpsPackage.hour);
635
636     lcd.setCursor(9, 0);
637     lcd.print(":");
638
639     lcd.setCursor(10, 0);
640     lcd.print(gpsPackage.minute);
641
642
643     lcd.setCursor(2, 1);
644     lcd.print(gpsPackage.day);
645
646     lcd.setCursor(4, 1);
647     lcd.print(".");
648
649     lcd.setCursor(5, 1);
650     lcd.print(gpsPackage.month);
651
652     lcd.setCursor(7, 1);
653     lcd.print(".");
654
655     lcd.setCursor(8, 1);
656     lcd.print(gpsPackage.year);
657
658
659     if(bEnter == 1){
660         menuLoc[1] = 0;
661         menuCursor = 2;
662         bEnter = 0;
663     }
664 }
665
666 void aboutMenu(){
667     lcd.setCursor(CurserX, menuCursor%2);
668     lcd.write(char(60));
669     lcd.setCursor(0, 0);
```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması
(13)

```
670     lcd.print(menuAbout[int(menuCursor/2)*2]);
671     lcd.setCursor(0, 1);
672     lcd.print(menuAbout[int(menuCursor/2)*2+1]);
673
674     if(bEnter == 1 and menuCursor == 0){
675         menuLoc[0] = 0;
676         menuLoc[1] = 0;
677         menuLoc[2] = 0;
678         menuCursor = 3;
679         bEnter = 0;
680     }
681 }
682
683 void getGpsPackage(){
684     delay(1);
685
686     radio.startListening();
687     for(int i = 0; i < 5; i++){
688         if(radio.available()){
689             i = 5;
690             rStatus = 1;
691             radio.read(&gpsPackage, sizeof(gpsPackage));
692         }
693     }
694     delay(1);
695 }
696
697 void getGasPackage(){
698     delay(1);
699
700     radio.startListening();
701     for(int i = 0; i < 5; i++){
702         if(radio.available()){
703             i = 5;
704             rStatus = 1;
705             radio.read(&gasPackage, sizeof(gasPackage));
706         }
707     }
708     delay(1);
709 }
710
711 void settingsMenu(){
712     lcd.setCursor(CurserX, menuCursor%2);
713     lcd.write(char(60));
714     lcd.setCursor(0, 0);
715     lcd.print(menuSettings[int(menuCursor/2)*2]);
716     lcd.setCursor(0, 1);
717     lcd.print(menuSettings[int(menuCursor/2)*2+1]);
718
719     if(bEnter == 1 and menuCursor == 0){
720         menuLoc[0] = 0;
721         menuLoc[1] = 0;
722         menuLoc[2] = 0;
723         menuCursor = 2;
724         bEnter = 0;
725     }
726     else if(bEnter == 1){
727         menuLoc[1] = menuCursor;
728         menuCursor = 0;
729         bEnter = 0;
730     }
731 }
732
733 void restoreConfig(){
734     package.restoreConfig = 1;
735 }
736
737 void modulesMenu(){
738     lcd.setCursor(CurserX, menuCursor%2);
739     lcd.write(char(60));
```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(14)

```
740 lcd.setCursor(0, 0);
741 lcd.print(menuModules[int(menuCursor/2)*2]);
742 lcd.setCursor(0, 1);
743 lcd.print(menuModules[int(menuCursor/2)*2+1]);
744
745 if(bEnter == 1 and menuCursor == 0){
746     menuLoc[0] = 0;
747     menuLoc[1] = 0;
748     menuLoc[2] = 0;
749     menuCursor = 1;
750     bEnter = 0;
751 }
752 else if(bEnter == 1){
753     menuLoc[1] = menuCursor;
754     menuCursor = 0;
755     bEnter = 0;
756 }
757 }
758
759 void lidarMenu(){
760     lcd.setCursor(CurserX, menuCursor%2);
761     lcd.write(char(60));
762     lcd.setCursor(0, 0);
763     lcd.print(menuLidar[int(menuCursor/2)*2]);
764     lcd.setCursor(0, 1);
765     lcd.print(menuLidar[int(menuCursor/2)*2+1]);
766
767     if(bEnter == 1 and menuCursor == 0){
768         menuLoc[1] = 0;
769         menuLoc[2] = 0;
770         menuCursor = 1;
771         bEnter = 0;
772     }
773     else if(bEnter and menuCursor == 1 and !doOnes[1]){
774         doOnes[1] = 1;
775         if(!package.Lidar){
776             menuLidar[1] = "Turn On ";
777             package.Lidar = 1;
778         }
779         else{
780             menuLidar[1] = "Turn Off";
781             package.Lidar = 0;
782         }
783     }
784
785     else if(bEnter and menuCursor == 2 and !doOnes[1]){
786         doOnes[1] = 1;
787         if(package.lidarSwitch == 0){
788             menuLidar[2] = "Switch 3 ";
789             package.lidarSwitch = 3;
790         }
791         else if(package.lidarSwitch == 3){
792             menuLidar[2] = "Switch 4 ";
793             package.lidarSwitch = 4;
794         }
795         else{
796             menuLidar[2] = "Switch None";
797             package.lidarSwitch = 0;
798         }
799     }
800     else if(!bEnter and doOnes[1]){
801         doOnes[1] = 0;
802     }
803 }
804
805 void droneMenu(){
806     lcd.setCursor(CurserX, menuCursor%2);
807     lcd.write(char(60));
808     lcd.setCursor(0, 0);
809     lcd.print(menuDrone[int(menuCursor/2)*2]);
810     lcd.setCursor(0, 1);
```


Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(15)

```
811     lcd.print(menuDrone[int(menuCursor/2)*2+1]);
812
813     if(bEnter == 1 and menuCursor == 0){
814         menuLoc[1] = 0;
815         menuLoc[2] = 0;
816         menuCursor = 2;
817         bEnter = 0;
818     }
819     else if(bEnter and menuCursor == 1 and !doOnes[1]){
820         doOnes[1] = 1;
821         if(!package.Drone){
822             menuDrone[1] = "Turn On ";
823             package.Drone = 1;
824         }
825         else{
826             menuDrone[1] = "Turn Off";
827             package.Drone = 0;
828         }
829     }
830
831     else if(bEnter and menuCursor == 2 and !doOnes[1]){
832         doOnes[1] = 1;
833         if(package.droneSwitch == 0){
834             menuDrone[2] = "Switch 3 ";
835             package.droneSwitch = 3;
836         }
837         else if(package.droneSwitch == 3){
838             menuDrone[2] = "Switch 4 ";
839             package.droneSwitch = 4;
840         }
841         else{
842             menuDrone[2] = "Switch None";
843             package.droneSwitch = 0;
844         }
845     }
846     else if(!bEnter and doOnes[1]){
847         doOnes[1] = 0;
848     }
849 }
850
851 void gasMenu(){
852     lcd.setCursor(CurserX, menuCursor%2);
853     lcd.write(char(60));
854     lcd.setCursor(0, 0);
855     lcd.print(menuGas[int(menuCursor/2)*2]);
856     lcd.setCursor(0, 1);
857     lcd.print(menuGas[int(menuCursor/2)*2+1]);
858
859     if(bEnter == 1 and menuCursor == 0){
860         menuLoc[1] = 0;
861         menuLoc[2] = 0;
862         menuCursor = 3;
863         bEnter = 0;
864     }
865     else if(bEnter and menuCursor == 1 and !doOnes[1]){
866         doOnes[1] = 1;
867         if(!package.Gas){
868             menuGas[1] = "Alert On ";
869             package.Gas = 1;
870         }
871         else{
872             menuGas[1] = "Alert Off";
873             package.Gas = 0;
874         }
875     }
876     else if(bEnter and menuCursor == 2 and !doOnes[1]){
877         doOnes[1] = 1;
878         menuLoc[2] = 1;
879         menuCursor = 0;
880         bEnter = 0;
881     }
}
```

(16)

```
882     else if(!bEnter and doOnes[1]){
883         doOnes[1] = 0;
884     }
885 }
886
887 void gasVarMenu(){
888     delay(1);
889     if(menuCursor == 0){
890         lcd.setCursor(7,0);
891         lcd.write(" ");
892         lcd.setCursor(0,0);
893         lcd.write("H2 ");
894         lcd.print(gasPackage.H2);
895         lcd.setCursor(7,1);
896         lcd.write(" ");
897         lcd.setCursor(0,1);
898         lcd.write("LPG ");
899         lcd.print(gasPackage.LPG);
900     }
901     if(menuCursor == 1){
902         lcd.setCursor(7,0);
903         lcd.write(" ");
904         lcd.setCursor(0,0);
905         lcd.write("CH4 ");
906         lcd.print(gasPackage.CH4);
907         lcd.setCursor(7,1);
908         lcd.write(" ");
909         lcd.setCursor(0,1);
910         lcd.write("CO ");
911         lcd.print(gasPackage.CO);
912     }
913     if(menuCursor == 2){
914         lcd.setCursor(7,0);
915         lcd.write(" ");
916         lcd.setCursor(0,0);
917         lcd.write("Alchol ");
918         lcd.setCursor(0,1);
919         lcd.write(" ");
920         lcd.setCursor(0,1);
921         lcd.print(gasPackage.Alchol);
922     }
923
924     if(bEnter == 1){
925         menuLoc[2] = 0;
926         menuCursor = 2;
927         bEnter = 0;
928     }
929 }
930
931 void RestoreConfig(){
932     package.restoreConfig = 1;
933 }
```

Ek 31 Kontrol Kumandası Yazılımı ve Akış Şeması

(17)

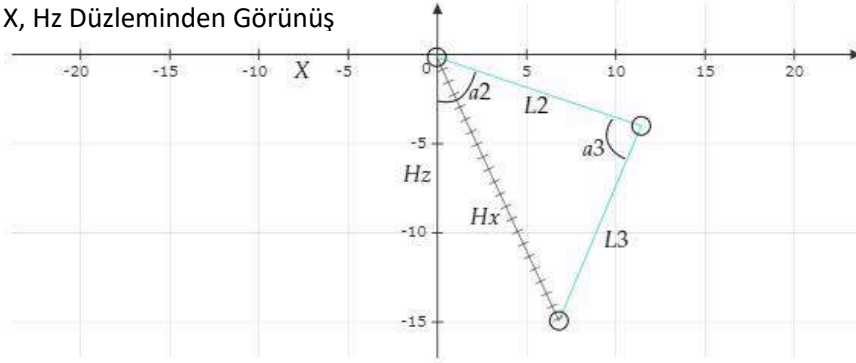
```
934
935 void setConfig(){
936     package.joystickSmoothness = config.joystickSmoothness;
937     package.path = config.path;
938     package.stepH = config.stepH;
939     package.gyroAssist = config.gyroAssist;
940     package.forceAssist = config.forceAssist;
941     package.lidarSwitch = config.lidarSwitch;
942     package.droneSwitch = config.droneSwitch;
943 }
944
945 void gasAlert(){
946     bool BCH4 = gasPackage.CH4 > 10000;
947     bool BCO = gasPackage.CO > 50;
948     if(BCO or BCH4){
949         lcd.clear();
950         delay(1);
951         if(BCO){
952             lcd.setCursor(0,0);
953             lcd.print("CO Danger!!");
954         }
955         else if(BCH4){
956             lcd.setCursor(0,0);
957             lcd.print("CH4 Danger!!");
958         }
959     }
960 }
961
962 void setConfigDefault(){
963     package.joystickSmoothness = 4;
964     package.path = 45;
965     package.stepH = 75;
966     package.gyroAssist = 1;
967     package.forceAssist = 1;
968     package.lidarSwitch = 0;
969     package.droneSwitch = 0;
970 }
971 }
```


Ek 32 Kontrol Kumandası Menü Ağacı

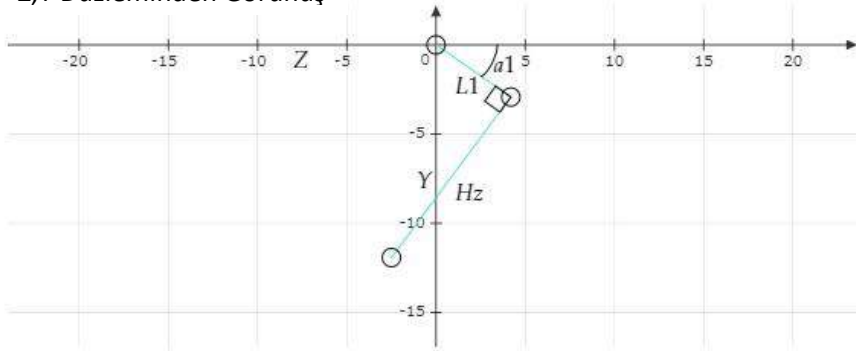
- a. GPS
 - a.1. Konum
 - Enlem, Boylam
 - a.2. Zaman
 - Saat, Dakika, Tarih (Gün/Ay/Yıl)
- b. Modüller
 - b.1. Lidar
 - Açma / Kapama
 - 3. veya 4. Anahtarlara Atama
 - b.2. Drone
 - Açma / Kapama
 - 3. veya 4. Anahtarlara Atama
 - b.3. Tehlikeli Gaz Modülü
 - İkaz Açma / Kapama
 - Gaz Değerleri CO, CH4, H2, LPG, Alkol
- c. Ayarlar
 - c.1. Başlangıç Değerlerine Dön
 - c.2. Kontrol Çubuğu Hassasiyeti
 - c.3. Bacak Hareket Aralığı Sıklığı (Path)
 - c.4. Adım Yüksekliği
 - c.5. Jiroskop Aktivasyon
 - c.6. Basınç Sensörü Aktivasyon
- d. Hakkında
 - LOTP Robot Köpek
 - Prototip Versiyon 2
 - Halid YILDIRIM Tarafından Yapılmıştır.

Ek 33 Kinematik Model Şeması ve Formülleri

X, Hz Düzleminde Görünüş



Z, Y Düzleminde Görünüş



$$X, Y, Z \Rightarrow a1, a2, a3$$

$$Hz = \sqrt{(Z + L1)^2 + Y^2} \quad (1)$$

$$Hx = \sqrt{X^2 + Hz^2} \quad (2)$$

$$a3 = \arccos\left(\frac{L2^2 + L3^2 - Hx^2}{2 * L2 * L3}\right) \quad (3)$$

$$a2 = \arccos\left(\frac{Hx^2 + L2^2 - L3^2}{2 * Hx * L2}\right) - \arcsin\left(\frac{X}{Hx}\right)$$

$$a1 = \arctan\left(\frac{Z + L1}{Y}\right) - \arctan\left(\frac{L1}{Y}\right)$$

L1: Omuz/Uyluk Ekleminin (Coxa) uzunluğunu temsil eder.

L2: Bacak Üst Çubuğunun (Femur) uzunluğunu temsil eder.

L3: Bacak Alt Çubuğunun (Tibia) uzunluğunu temsil eder.

Hx: Omuz ekleminde ayak ucuna kadar olan mesafe (X, Hz Düzleminde)

Hz: Omuz ekleminde ayak ucuna kadar olan mesafe (Z, Y Düzleminde)

a1: L1 ile Z düzlemlerinin açısı

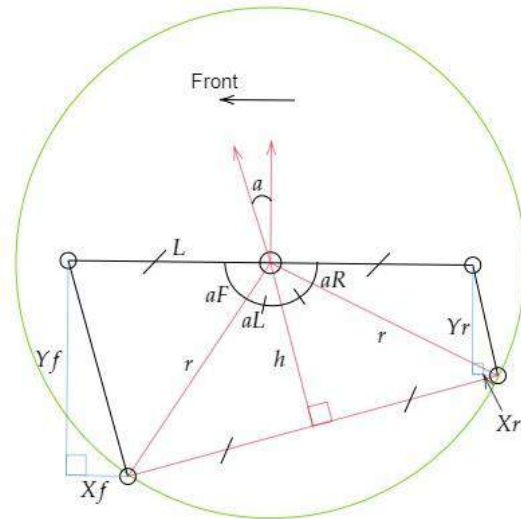
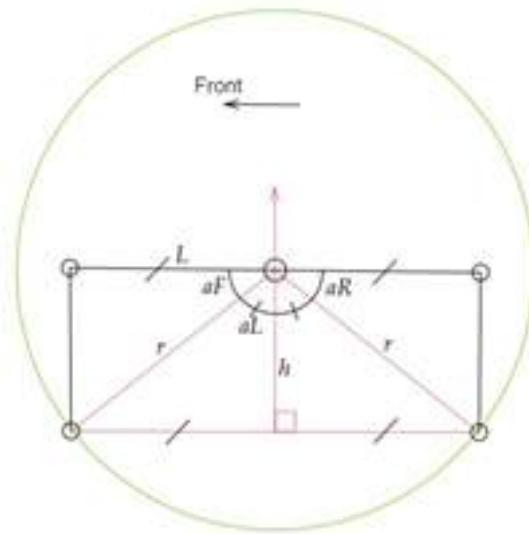
a2: L2 ve Hz'nin açısı

a3: L2 ve L3'ün açısı

Ek 34 Kinematik Model Fonksiyonu

```
682 void moveLegsPositions(){
683     double Hz = sqrt(sq(FLLegPos[2] + Leg1) + sq(FLLegPos[1]));
684     double Hx = sqrt(sq(FLLegPos[0]) + sq(Hz));
685
686     double a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
687     double a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2) ) * 57296 /
688     1000 - asin(FLLegPos[0] / Hx) * 57296 / 1000;
689     double a1 = atan((FLLegPos[2] + Leg1) / FLLegPos[1]) * 57296 / 1000 - atan(Leg1 /
690     FLLegPos[1]) * 57296 / 1000;
691
692     FLLeg[2].write(map(a3, 35, 135, 180, 0) + FLLegAdd[2]);
693     FLLeg[1].write(a2 + FLLegAdd[1]);
694     FLLeg[0].write(180 - (90 - a1) + FLLegAdd[0]);
695
696     Hz = sqrt(sq(FRLegPos[2] + Leg1) + sq(FRLegPos[1]));
697     Hx = sqrt(sq(FRLegPos[0]) + sq(Hz));
698
699     a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
700     a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2) ) * 57296 / 1000 -
701     asin(FRLegPos[0] / Hx) * 57296 / 1000;
702     a1 = atan((FRLegPos[2] + Leg1) / FRLegPos[1]) * 57296 / 1000 - atan(Leg1 /
703     FRLegPos[1]) * 57296 / 1000;
704
705     FRLeg[2].write(map(a3, 35, 135, 0, 180) + FRLegAdd[2]);
706     FRLeg[1].write(180 - a2 + FRLegAdd[1]);
707     FRLeg[0].write(180 - (90 + a1) + FRLegAdd[0]);
708
709     Hz = sqrt(sq(RLLegPos[2] + Leg1) + sq(RLLegPos[1]));
710     Hx = sqrt(sq(RLLegPos[0]) + sq(Hz));
711
712     a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
713     a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2) ) * 57296 / 1000 -
714     asin(RLLegPos[0] / Hx) * 57296 / 1000;
715     a1 = atan((RLLegPos[2] + Leg1) / RLLegPos[1]) * 57296 / 1000 - atan(Leg1 /
716     RLLegPos[1]) * 57296 / 1000;
717
718     RLLeg[2].write(map(a3, 35, 135, 180, 0) + RLLegAdd[2]);
719     RLLeg[1].write(a2 + RLLegAdd[1]);
720     RLLeg[0].write((90 - a1) + RLLegAdd[0]);
721
722     Hz = sqrt(sq(RRLegPos[2] + Leg1) + sq(RRLegPos[1]));
723     Hx = sqrt(sq(RRLegPos[0]) + sq(Hz));
724
725     a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
726     a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2) ) * 57296 / 1000 -
727     asin(RRLegPos[0] / Hx) * 57296 / 1000;
728     a1 = atan((RRLegPos[2] + Leg1) / RRLegPos[1]) * 57296 / 1000 - atan(Leg1 /
729     RRLegPos[1]) * 57296 / 1000;
730
731     RRLeg[2].write(map(a3, 35, 135, 0, 180) + RRLegAdd[2]);
732     RRLeg[1].write(180 - a2 + RRLegAdd[1]);
733     RRLeg[0].write((90 + a1) + RRLegAdd[0]);
734 }
```


Ek 35 Ters Kinematik Şeması ve Formülleri



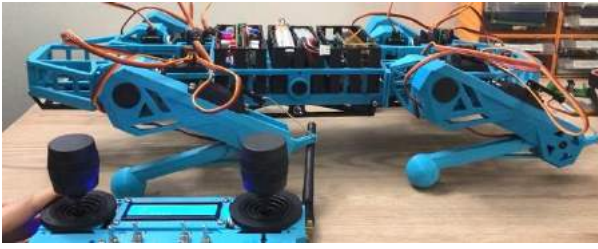
$r = \sqrt{L^2 + h^2}$	(1)
$aL = \arccos\left(\frac{r^2 + h^2 - L^2}{2 * r * h}\right)$	(2)
$aF = 270 - a - aL$ $aF = 270 + a - aL$	(3)
$Yr = r * \sin(aR)$ $Yf = r * \sin(aF)$	(4)
$Xr = L + r * \cos(aR)$ $Xf = L + r * \cos(aF)$	(5)

$$a, h \Rightarrow Xf, Yf, Xr, Yr$$

Ek 36 Ters Kinematik Model Fonksiyonu

```
1 void standingMove(){
2
3 // X,Y Bakış açısı
4
5 double r = sqrt(sq(LegtoOrg[0]) + sq(Yin));
6 double A1 = degrees(acos((sq(r) + sq(Yin) - sq(LegtoOrg[0])) / (2 * r * Yin)));
7
8 double Ar = 270 - Ay - A1;
9 double Af = 270 + Ay - A1;
10
11 double Yr = r * sin(radians(Ar)) + sin(radians(Ax)) * Xin + sin(radians(Az)) * Zin;
12 double Yf = r * sin(radians(Af)) + sin(radians(Ax)) * Xin + sin(radians(Az)) * Zin;
13
14 double Xf = LegtoOrg[0] + r * cos(radians(Af));
15 double Xr = LegtoOrg[0] + r * cos(radians(Ar));
16
17 double Xz = cos(radians(32.34 + Az)) * sqrt(sq(LegtoOrg[1]) + sq(LegtoOrg[0])) - 150;
18 double Zz = sin(radians(32.34 + Az)) * sqrt(sq(LegtoOrg[1]) + sq(LegtoOrg[0])) - 95;
19
20 //-----
21
22 // Y,Z Bakış açısı (Ön Bacaklar)
23
24 r = sqrt(sq(LegtoOrg[1]) + sq(Yf));
25 A1 = degrees(acos((sq(r) + sq(Yf) - sq(LegtoOrg[1])) / (2 * r * Yf)));
26
27 double Ari = 270 - Ax - A1;
28 double Ale = 270 + Ax - A1;
29
30 double Yri = r * sin(radians(Ari));
31 double Yle = r * sin(radians(Ale));
32
33 double Zri = LegtoOrg[1] + r * cos(radians(Ari));
34 double Zle = LegtoOrg[1] + r * cos(radians(Ale));
35
36 FLLegPos[0] = -Xf + cos(radians(Ax)) * Xin + Xz;
37 FLLegPos[1] = Yle;
38 FLLegPos[2] = -Zle - cos(radians(Az)) * Zin + Zz;
39
40 FRLegPos[0] = -Xf + cos(radians(Ax)) * Xin - Xz;
41 FRLegPos[1] = Yri;
42 FRLegPos[2] = -Zri + cos(radians(Az)) * Zin - Zz;
43
44 // Y,Z Bakış açısı (Arka Bacaklar)
45
46 r = sqrt(sq(LegtoOrg[1]) + sq(Yr));
47 A1 = degrees(acos((sq(r) + sq(Yr) - sq(LegtoOrg[1])) / (2 * r * Yr)));
48
49 Ari = 270 - Ax - A1;
50 Ale = 270 + Ax - A1;
51
52 Yri = r * sin(radians(Ari));
53 Yle = r * sin(radians(Ale));
54
55 Zri = LegtoOrg[1] + r * cos(radians(Ari));
56 Zle = LegtoOrg[1] + r * cos(radians(Ale));
57
58 RLLegPos[0] = -Xf + cos(radians(Ax)) * Xin + Xz;
59 RLLegPos[1] = Yle;
60 RLLegPos[2] = -Zle - cos(radians(Az)) * Zin - Zz;
61
62 RRLegPos[0] = -Xf + cos(radians(Ax)) * Xin - Xz;
63 RRLegPos[1] = Yri;
64 RRLegPos[2] = -Zri + cos(radians(Az)) * Zin + Zz;
65 }
```

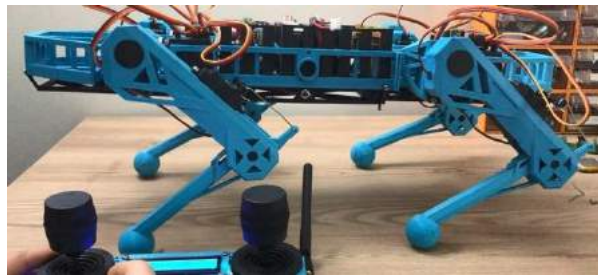
Ek 37 Kinematik model ve Ters Kinematik model Yerinde Hareket Uygulamaları -1



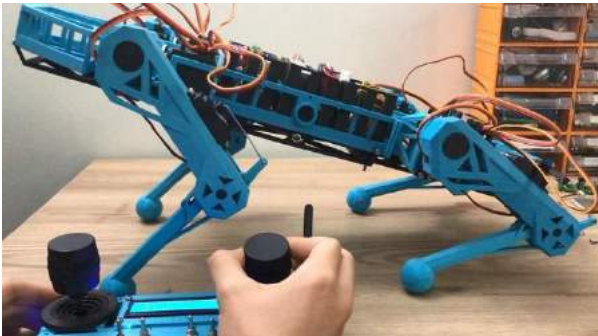
Açık



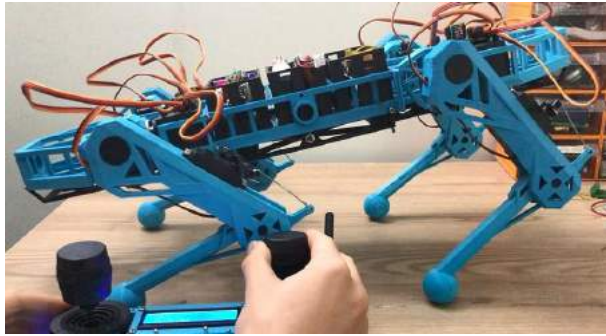
Kapalı



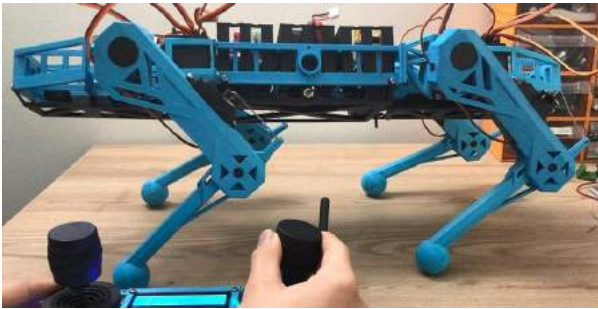
Normal Duruş



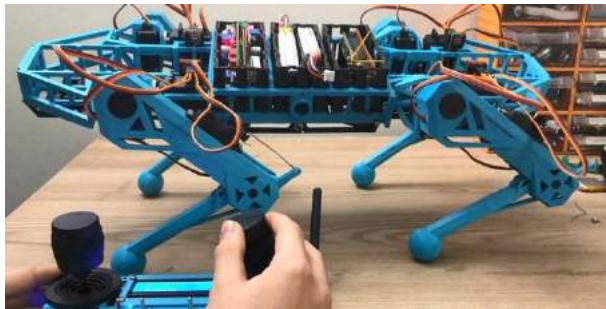
1. Arkaya Eğilme



2. Öne Eğilme

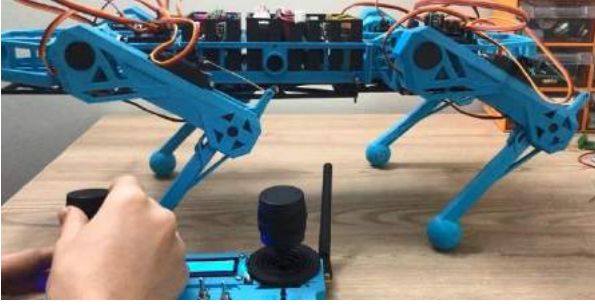


3. Sağa Eğilme

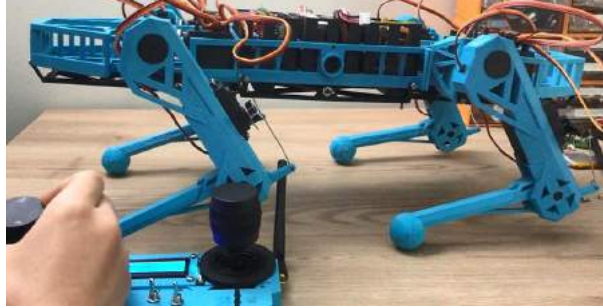


4. Sola Eğilme

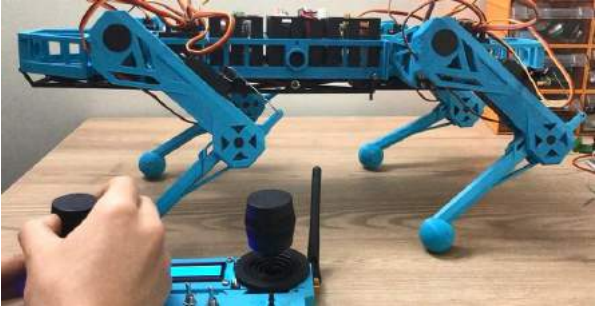
Ek 38 Kinematik model ve Ters Kinematik model Yerinde Hareket Uygulamaları -2



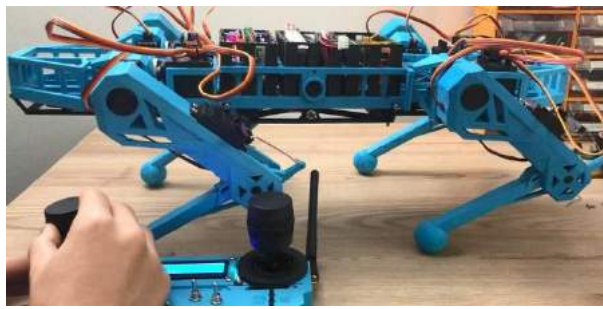
5. Öne Yatma



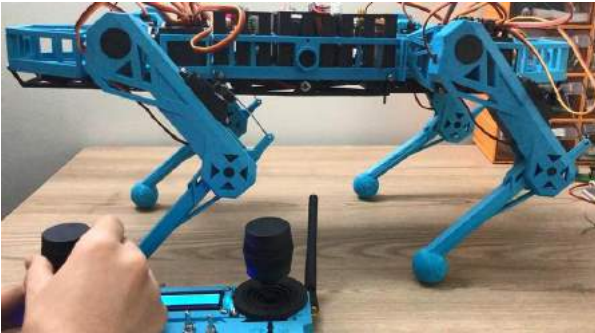
6. Arkaya Yatma



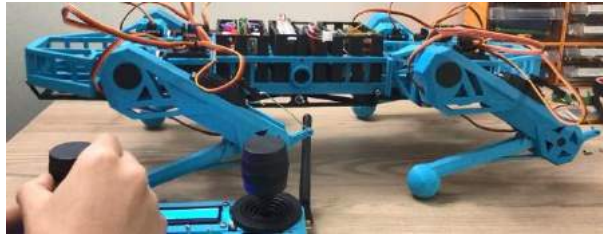
7. Sağa Yatma



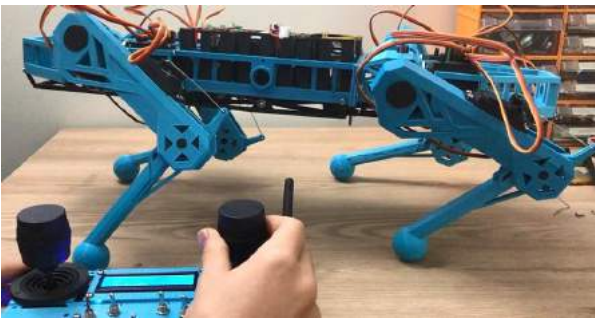
8. Sola Yatma



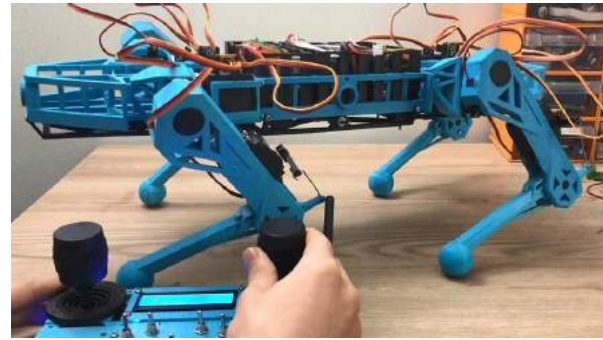
9. Yukarı Yükselme



10. Aşağı Alçalma

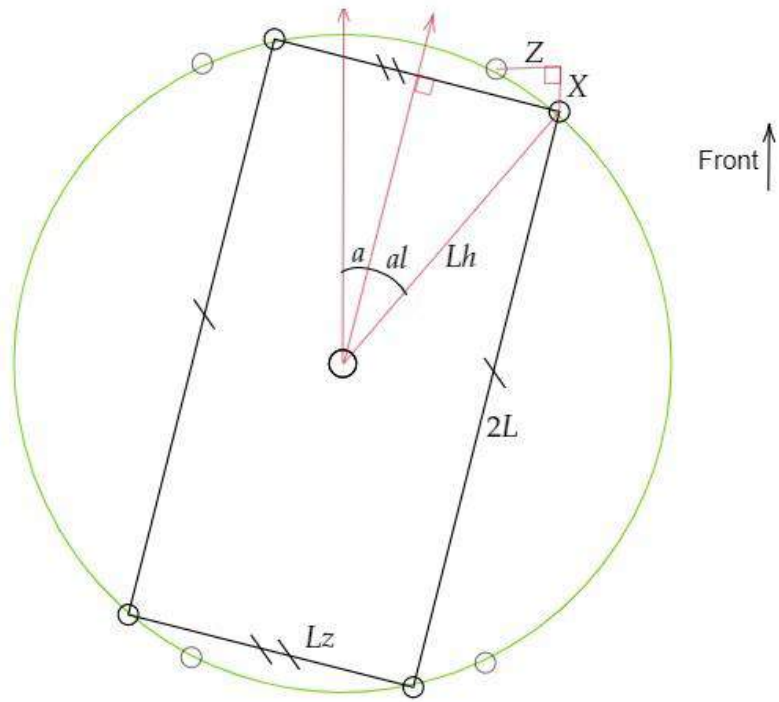


11. Sağa Yönelme



12. Sola Yönelme

Ek 39 X,Z Düzleminin Ters Kinematik Model Şeması ve Formülleri



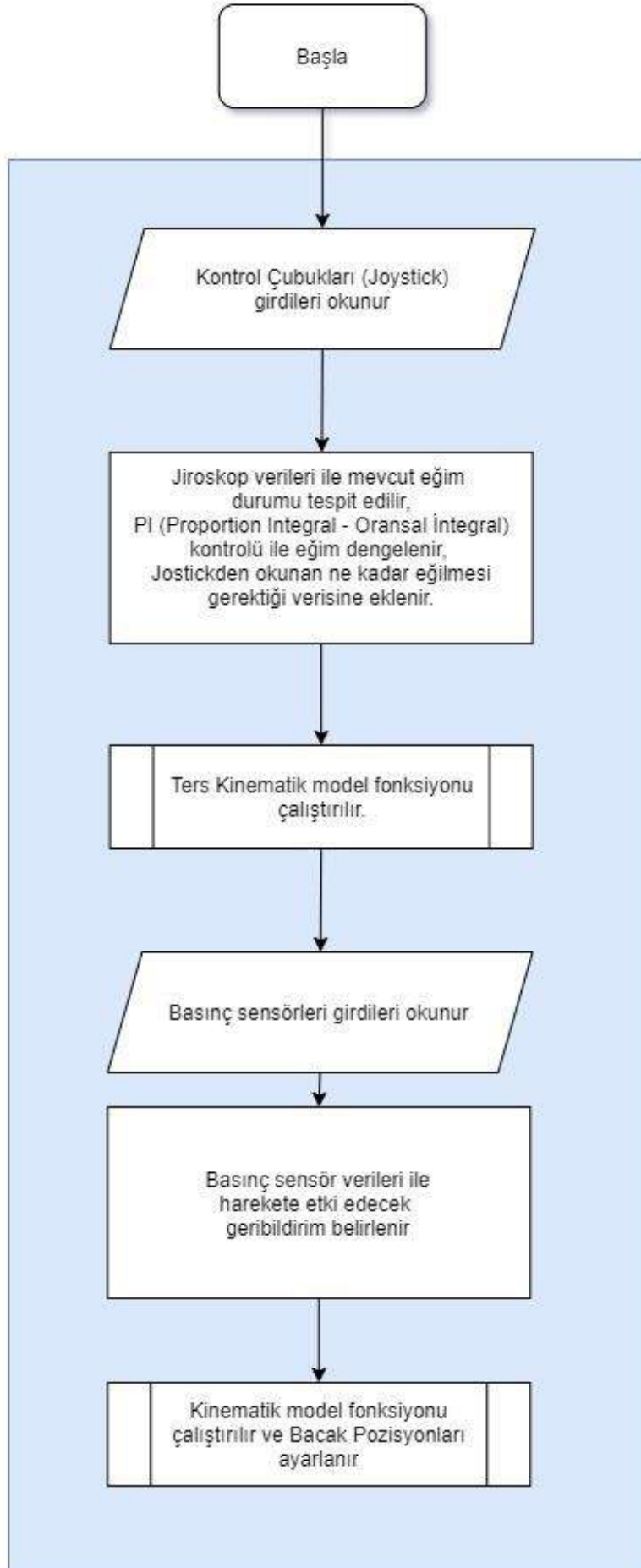
$a \Rightarrow X, Z$

$$Lh = \sqrt{\left(\frac{Lz}{2}\right)^2 + L^2} \quad (1)$$

$$al = \text{atan}\left(\frac{Lz}{2L}\right) \quad (2)$$

$$\begin{aligned} X &= \cos(a + al) + \cos(l) \\ Z &= \sin(a + al) + \sin(l) \end{aligned} \quad (3)$$

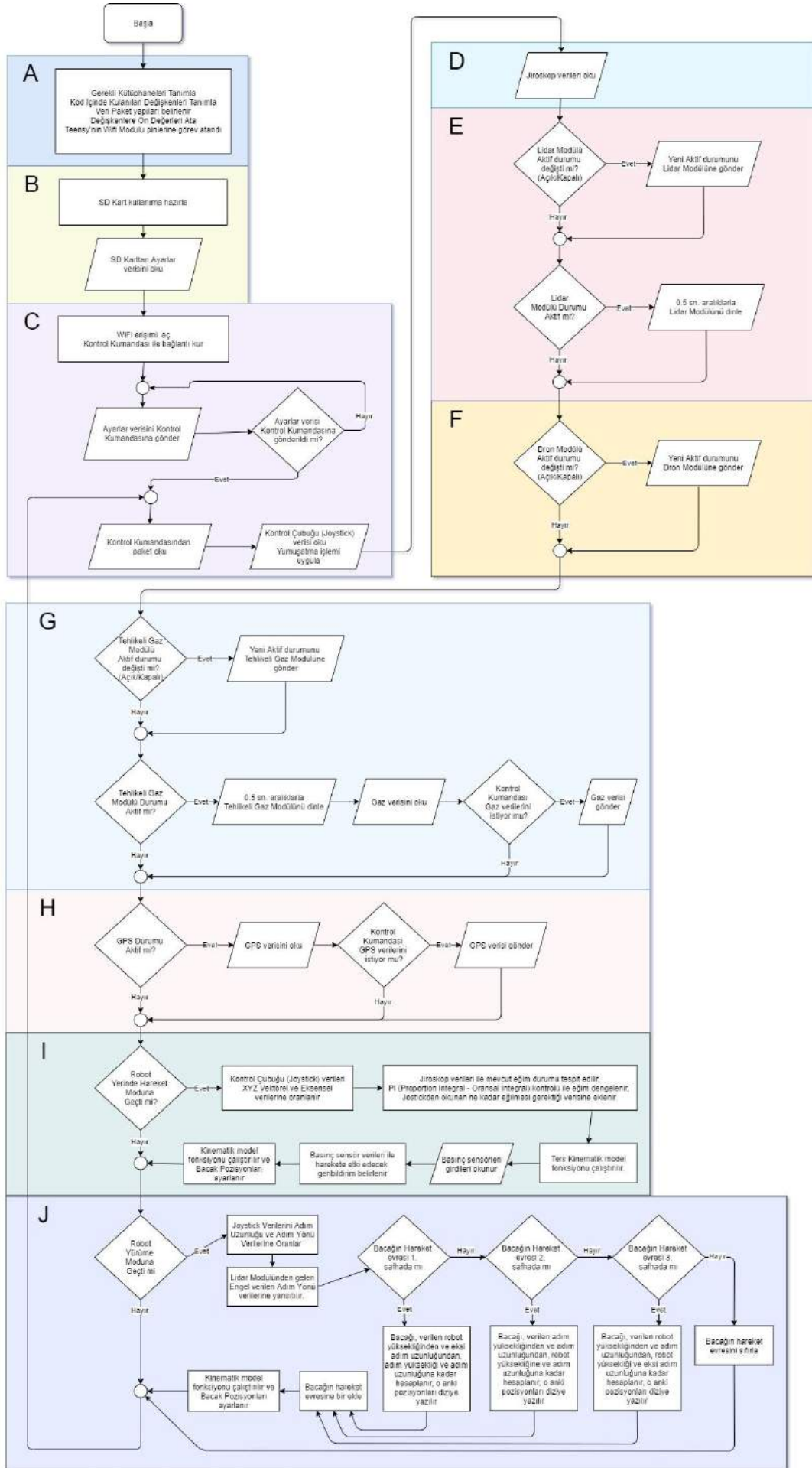
Ek 40 Yerinde Hareket Yazılımı ve Akış Şeması (1)



Ek 40 Yerinde Hareket Yazılımı ve Akış Şeması (2)

```
1
2 // Joystick girdileri okunur
3
4 Yin = 200 - map(joystick[5], 0, 1023, 50, 0);
5 Xin = map(joystick[4], 0, 1023, -50, 50);
6 Zin = map(joystick[3], 0, 1023, -45, 45);
7
8 Ax = map(joystick[0], 0, 1023, -25, 25);
9 Ay = map(joystick[1], 0, 1023, -15, 15);
10 Az = map(joystick[2], 0, 1023, -15, 15);
11
12 // Jiroskop verileri okunur, mevcut eğim durumu tespit edilir,
13 // PI kontrol ile eğim dengelenir,
14 // Jostickden okunan ne kadar eğilmesi gerektiği verisine eklenir.
15
16 gyro[0] += Ax;
17 gyro[1] += Ay;
18 int p;
19 if(millis() - gyroLast and millis() - gyroLast < 200){
20   p = (millis() - gyroLast) * 50;
21 }
22 else{
23   p = 100;
24 }
25 for(int i = 0; i < 2 and config.gyroAssist; i++){
26
27   if(abs(gyro[i]) > 2 and abs(Ag[i] + gyro[i] / p) <= 15) {Ag[i] += gyro[i] / p; gyroLast = millis();}
28
29 }
30
31 Ax += Ag[0];
32 Ay += Ag[1];
33 Zin += sin(radians(Ag[0])) * Yin;
34 Xin += sin(radians(Ag[1])) * Yin;
35
36 // Ters Kinematik model uygulanır
37
38 standingMove();
39
40 // Basınç verileri okunur ve harekete etki edecek geribildirim belirlenir
41
42 if(config.forceAssist){
43   if(analogRead(A1) > maxForce[0] and -FLLegPress < 50){
44     FLLegPress -= analogRead(A1) / maxForce[0];
45   }
46   else if(abs(FLLegPress) > 1){
47     FLLegPress += 1;
48   }
49   FLLegPos[1] += FLLegPress;
50
51   if(analogRead(A0) > maxForce[1] and -FRLegPress < 50){
52     FRLegPress -= analogRead(A0) / maxForce[1];
53   }
54   else if(abs(FRLegPress) > 1){
55     FRLegPress += 1;
56   }
57   FRLegPos[1] += FRLegPress;
58
59   if(analogRead(A3) > maxForce[2] and -RLLegPress < 50){
60     RLLegPress -= analogRead(A3) / maxForce[2];
61   }
62   else if(abs(RLLegPress) > 1){
63     RLLegPress += 1;
64   }
65   RLLegPos[1] += RLLegPress;
66
67   if(analogRead(A2) > maxForce[3] and -RRLegPress < 50){
68     RRLegPress -= analogRead(A2) / maxForce[3];
69   }
70   else if(abs(RRLegPress) > 1){
71     RRLegPress += 1;
72   }
73   RRLegPos[1] += RRLegPress;
74 }
75
76 // Kinematik model uygulanır ve Bacak Pozisyonları ayarlanır.
77
78 moveLegsPositions();
79
```


Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması (1)



Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması (2)

A

Gerekli kütüphaneler tanımlanır.

Kod içinde kullanılan değişkenleri tanımlanır.

Veri paket yapıları belirlenir.

Değişkenlere ön değerleri atanır.

Teensy'nin Wifi modülü pinlerine görev atanır.

```
1 #include <SoftwareSerial.h>
2 #include <MPU6050_light.h>
3 #include <ArduinoJson.h>
4 #include <TinyGPS++.h>
5 #include <nRF24L01.h>
6 #include <Servo.h>
7 #include <RF24.h>
8 #include <math.h>
9 #include <Wire.h>
10 #include <SPI.h>
11 #include <SD.h>
12
13 int Leg1 = 50;
14 int Leg2 = 125;
15 int Leg3 = 125;
16
17 SoftwareSerial ss(0, 1); //rx, tx
18 RF24 radio(7, 8); // CE, CSN
19 MPU6050 mpu(Wire);
20 TinyGPSPlus gps;
21 Servo FLleg[3];
22 Servo FRleg[3];
23 Servo RLleg[3];
24 Servo RRleg[3];
25
26 const uint64_t tAddress = 0xE8E8F0F0E0LL;
27 const uint64_t rAddress = 0xE8E8F0F0E1LL;
28
29 int joystick[6] = {512, 512, 512, 512, 512, 512};
30 int oldJoystick[6] = {512, 512, 512, 512, 512, 512};
31 float gyro[3] = {0, 0, 0};
32 int maxForce[4] = {300, 450, 300, 500};
33 float FLlegPress;
34 float FRlegPress;
35 float RLlegPress;
36 float RRlegPress;
37 double gyroLast;
38 double limitMove[7];
39 float Ag[3] = {0, 0, 0};
40 int curser = 6;
41 int moveSideA = 0;
42 double Sz;
43 double disLimit[2] = [5000, 5000]
44
45 int Yin;
46 int Xin;
47 int Zin;
48
49 double Ax;
50 double Ay;
51 double Az;
52
53 char oldStand;
54
55 int LegtoOrg[3] = {150,95,0};
56
57 int FLlegPins[3] = {6,25,26};
58 int FRlegPins[3] = {5,24,4};
59 int RLlegPins[3] = {31,29,32};
60 int RRlegPins[3] = {30,28,27};
61
62 float FLlegPos[3] = {0,100,0};
63 float FRlegPos[3] = {0,100,0};
64 float RLlegPos[3] = {0,100,0};
65 float RRlegPos[3] = {0,100,0};
66
67 float FLlegAdd[3] = {0,10,0};
68 float FRlegAdd[3] = {5,0,0};
69 float RLlegAdd[3] = {5,15,0};
70 float RRlegAdd[3] = {0,0,10};
71
72 union{
73     byte array[4];
74     float bigNum;
75 } H2;
76
```

A

Gerekli kütüphaneler tanımlanır.

Kod içinde kullanılan değişkenleri tanımlanır.

Veri paket yapıları belirlenir.

Değişkenlere ön değerleri atanır.

Teensy'nin Wifi modülü pinlerine görev atanır.

```
77 union{
78   byte array[4];
79   float bigNum;
80 } LPG;
81
82 union{
83   byte array[4];
84   float bigNum;
85 } CH4;
86
87 union{
88   byte array[4];
89   float bigNum;
90 } CO;
91
92 union{
93   byte array[4];
94   float bigNum;
95 } Alchol;
96
97 union{
98   byte array[4];
99   float bigNum;
100 } xForce;
101
102 union{
103   byte array[4];
104   float bigNum;
105 } yForce;
106
107 struct Package{
108   int16_t joystick[6];
109   bool gpsRequest;
110   bool gasRequest;
111
112   bool standMode;
113   bool walkMode;
114
115   bool Gps;
116   bool Lidar;
117   bool Drone;
118   bool Gas;
119
120   bool restoreConfig;
121
122   byte joystickSmoothness;
123   byte path;
124   byte stepH;
125   bool gyroAssist;
126   bool forceAssist;
127   bool WalkMode;
128
129   byte lidarSwitch;
130   byte droneSwitch;
131 };
132
133 struct GPSPackage{
134   byte staller;
135
136   byte hour;
137   byte minute;
138
139   int16_t year;
140   byte month;
141   byte day;
142
143   float longti;
144   float lati;
145 };
146
147 struct GasPackage{
148   float H2;
149   float LPG;
150   float CH4;
151   float CO;
152   float Alchol;
153 };
```

Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması (4)

A

Gerekli kütüphaneler tanımlanır.

Kod içinde kullanılan değişkenleri tanımlanır.

Veri paket yapıları belirlenir.

Değişkenlere ön değerleri atanır.

Teensy'nin Wifi modülü pinlerine görev atanır.

B

SD Kart kullanıma hazırlanır

SD Karttan Ayarlar verileri okunur

```
154
155 ▼ struct Config {
156     byte joystickSmoothness;
157     byte path;
158     byte stepH;
159     byte gyroAssist;
160     byte forceAssist;
161     byte lidarSwitch;
162     byte droneSwitch;
163     bool WalkMode;
164 };
165
166
167 const char *filename = "/config.txt";
168 Config config;
169
170 Package package;
171 Package oldPackage;
172
173 GPSPackage gpsPackage;
174 GasPackage gasPackage;
175
176 int DroneAddress = 3;
177 int GasAddress = 5;
178 int LidarAddress = 9;
179 byte LidarMap[50];
180 byte gasVar[5];
181 double lidarTime = 0;
182 double gasTime = 0;
183
184 float hoverTime = 2;
185 float stepDelay = 1;
186 float pushTime = hoverTime + stepDelay;
187
188 int maxStepX= 150;
189 int maxStepZ = 100;
190 int stepOffset = 0;
191 int tilt = 10;
192
193 float sTime[4];
194
195 float W2s[4] = {0, pushTime, pushTime, 0};
196 float W4s[4] = {0, pushTime, 2 * pushTime, 3 * pushTime};
197
198 ▼ void setup() {
199     Serial.begin(115200);
200     ss.begin(9600);
201     Wire.begin();
202     Wire1.begin();
203
204     while (!SD.begin(BUILTIN_SDCARD)) {
205         Serial.println(F("Failed to initialize SD library"));
206         delay(1000);
207     }
208
209     mpu.begin();
210     Serial.println("MPU6050 Found!");
211
212     loadConfiguration(filename, config);
213
214     for(int i = 0; i < 4; i++){
215         sTime[i] = W2s[i];
216     }
217
218     for(int i = 0; i < 4; i++){
219         sTime[i] *= config.path;
220     }
221
222     for(int i = 0; i < 3; i++){
223         FLLeg[i].attach(FLLegPins[i]);
224         FLLeg[i].write(90 + FLLegAdd[i]);
```


Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması

(5)

B

C

WiFi erişimi açılır. Kontrol Kumandası ile bağlantı kurulur. Ayarlar verisi gönderilir ve kontrol edilir.

Kontrol Çubuğu (Joystick) verisi okunur

D

Jiroskop verisi okunur

E

Lidar Modülü Aktif durumu kontrol edilir. Yeni Aktif durumu Lidar Modülüne gönderilir. 05 sn. aralıklarla Lidar Modülü dinlenir

F

Dron İstasyonu Modülü Aktif durumu kontrol edilir. Yeni Aktif durumu gönderilir.

G

Tehlikeli Gaz Modülü Aktif durumu kontrol edilir.

```
276 if(package.Lidar and millis() - lidarTime > 500){
277     lidarTime = millis();
278     getLidar();
279 }
280
281 if(oldDrone != package.Drone){
282     if(package.Drone){
283         Wire1.beginTransmission(DroneAddress);
284         Wire1.write(1);
285         Wire1.endTransmission();
286     }
287     else{
288         Wire1.beginTransmission(DroneAddress);
289         Wire1.write(0);
290         Wire1.endTransmission();
291     }
292 }
293
294 if(oldGas != package.Gas){
295     if(package.Gas){
296         Wire1.beginTransmission(GasAddress);
297         Wire1.write(1);
298         Wire1.endTransmission();
299     }
300     else{
301         Wire1.beginTransmission(GasAddress);
302         Wire1.write(0);
303     }
304 }
305
306 for(int i = 0; i < 3; i++){
307     FRLeg[i].attach(FRLegPins[i]);
308     FRLeg[i].write(90 + FRLegAdd[i]);
309 }
310
311 for(int i = 0; i < 3; i++){
312     RLleg[i].attach(RLlegPins[i]);
313     RLleg[i].write(90 + RLlegAdd[i]);
314 }
315
316 for(int i = 0; i < 3; i++){
317     RRleg[i].attach(RRlegPins[i]);
318     RRleg[i].write(90 + RRlegAdd[i]);
319 }
320
321 radio.begin();
322 radio.openWritingPipe(tAddress);
323 radio.openReadingPipe(1,rAddress);
324
325 moveLegsPositions();
326
327 delay(2500);
328 mpu.calcOffsets();
329
330 radio.startListening();
331 delay(1);
332 }
333
334 void loop() {
335
336     int oldDrone = package.Drone;
337     int oldLidar = package.Lidar;
338     int oldGas = package.Gas;
339     getPackage();
340
341     getJoystick();
342
343     mpu.update();
344     getGyro();
345
346     if(oldLidar != package.Lidar){
347         if(package.Lidar){
348             Wire1.beginTransmission(LidarAddress);
349             Wire1.write(1);
350             Wire1.endTransmission();
351         }
352         else{
353             Wire1.beginTransmission(LidarAddress);
354             Wire1.write(0);
355             Wire1.endTransmission();
356         }
357     }
358 }
```

Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması (6)

G

Yeni durumu
Tehlikeli Gaz
Modülüne
gönderilir.
05 sn.
aralıklarla
Tehlikeli Gaz
Modülü
dinlenir

H

GPS Durumu
aktif mi
kontrol edilir.

GPS verisi
okunur

Kontrol
Kumandası
GPS verilerini
istiyor mu
kontrol edilir.

İstiyorsa GPS
verisi
gönderilir.

I

Robot
Yerinde
Hareket
Modunda
ise

Kontrol
Çubuğu
(Joystick)
verileri XYZ
Vektörel ve
Eksensel
verilerine
oranlanır

```
303     Wire1.endTransmission();
304     }
305 }
306 if(package.Gas and millis() - gasTime > 500){
307     gasTime = millis();
308     getGas();
309 }
310 if(package.gasRequest){
311     delay(1);
312     radio.stopListening();
313     radio.write(&gasPackage, sizeof(gasPackage));
314     radio.startListening();
315     delay(1);
316 }
317
318 if(package.Gps){
319     while(ss.available() > 0){
320         if(gps.encode(ss.read())){
321             if(gps.location.isValid()){
322                 gpsPackage.lati = gps.location.lat();
323                 gpsPackage.longti = gps.location.lng();
324             }
325
326             if(gps.time.isValid()){
327                 gpsPackage.hour = gps.time.hour();
328                 gpsPackage.minute = gps.time.minute();
329             }
330
331             if(gps.date.isValid()){
332                 gpsPackage.year = gps.date.year();
333                 gpsPackage.month = gps.date.month();
334                 gpsPackage.day = gps.date.day();
335             }
336
337             if(gps.satellites.isValid()){
338                 gpsPackage.staller = gps.satellites.value();
339             }
340
341             if(package.gpsRequest){
342                 delay(1);
343                 radio.stopListening();
344                 radio.write(&gpsPackage, sizeof(gpsPackage));
345                 radio.startListening();
346                 delay(1);
347             }
348         }
349     }
350 }
351
352 if(package.standMode and !package.walkMode){
353     if(oldStand != package.standMode){
354         int Bpos[12];
355         for(int i = 0; i < 3; i++){
356             Bpos[i] = FLLegPos[i];
357         }
358         for(int i = 0; i < 3; i++){
359             Bpos[i + 3] = FRLegPos[i];
360         }
361         for(int i = 0; i < 3; i++){
362             Bpos[i + 6] = RLLegPos[i];
363         }
364         for(int i = 0; i < 3; i++){
365             Bpos[i + 9] = RRLegPos[i];
366         }
367
368         int sitPoint[3] = {0,175,0};
369         int line[3];
370         for(int i = 0; i < 100; i++){
371             delay(5);
372             for(int v = 0; v < 3; v++){
373                 line[v] = Bpos[v] - sitPoint[v];
374                 FLLegPos[v] = Bpos[v] - line[v] * i / 100 ;
375             }
376         }
377     }
```

Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması

(7)

<p> </p>	<p>Kontrol Çubuğu (Joystick) verileri XYZ Vektörel ve Eksensel verilerine oranlanır</p>	<pre> 378 379 for(int v = 0; v < 3; v++){ 380 line[v] = Bpos[v + 3] - sitPoint[v]; 381 382 FRLegPos[v] = Bpos[v + 3] - line[v] * i / 100 ; 383 } 384 385 for(int v = 0; v < 3; v++){ 386 line[v] = Bpos[v + 6] - sitPoint[v]; 387 388 RLLegPos[v] = Bpos[v + 6] - line[v] * i / 100 ; 389 } 390 391 for(int v = 0; v < 3; v++){ 392 line[v] = Bpos[v + 9] - sitPoint[v]; 393 394 RRLegPos[v] = Bpos[v + 9] - line[v] * i / 100 ; 395 } 396 397 moveLegsPositions(); 398 } 399 oldStand = package.standMode; 400 } 401 Yin = 200 - map(joystick[5], 0, 1023, 50, 0); 402 Xin = map(joystick[4], 0, 1023, -50, 50); 403 Zin = map(joystick[3], 0, 1023, -45, 45); 404 405 Ax = map(joystick[0], 0, 1023, -25, 25); 406 Ay = map(joystick[1], 0, 1023, -15, 15); 407 Az = map(joystick[2], 0, 1023, -15, 15); 408 409 gyro[0] += Ax; 410 gyro[1] += Ay; 411 int p; 412 if(millis() - gyroLast and millis() - gyroLast < 200){ 413 p = (millis() - gyroLast) * 50; 414 } 415 else{ 416 p = 100; 417 } 418 for(int i = 0; i < 2 and config.gyroAssist; i++){ 419 420 if(abs(gyro[i]) > 2 and abs(Ag[i] + gyro[i] / p) <= 15) {Ag[i] += gyro[i] / p; 421 gyroLast = millis();} 422 } 423 424 Ax += Ag[0]; 425 Ay += Ag[1]; 426 Zin += sin(radians(Ag[0])) * Yin; 427 Xin += sin(radians(Ag[1])) * Yin; 428 429 standingMove(); 430 431 if(config.forceAssist){ 432 if(analogRead(A1) > maxForce[0] and -FLLegPress < 50){ 433 FLLegPress -= analogRead(A1) / maxForce[0]; 434 } 435 else if(abs(FLLegPress) > 1){ 436 FLLegPress += 1; 437 } 438 FLLegPos[1] += FLLegPress; 439 440 if(analogRead(A0) > maxForce[1] and -FRLegPress < 50){ 441 FRLegPress -= analogRead(A0) / maxForce[1]; 442 } 443 else if(abs(FRLegPress) > 1){ 444 FRLegPress += 1; 445 } 446 FRLegPos[1] += FRLegPress; 447 448 if(analogRead(A3) > maxForce[2] and -RLLegPress < 50){ 449 RLLegPress -= analogRead(A3) / maxForce[2]; 450 } 451 else if(abs(RLLegPress) > 1){ </pre>
<p>Jiroskop verileri ile mevcut eğim durumu tespit edilir.</p>	<p>PI (Proportion Integral Oransal İntegral) kontrolü ile eğim dengelenir.</p>	<p>Kontrol Çubuğundan (Joystick) okunan veriler ne kadar eğilmesi gerektiği verisine eklenir.</p>
<p>Ters Kinematik model fonksiyonu çalıştırılır.</p>	<p>Basınç sensörleri girdileri okunur.</p>	

Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması

(8)

I
Basınç sensör verileri ile harekete etki edecek geribildirim belirlenir.

J
Kinematik model fonksiyonu çalıştırılır ve Bacak Pozisyonları ayarlanır.

J
Robot Yürüme Modunda ise

```

452     RLLegPress += 1;
453     }
454     RLLegPos[1] += RLLegPress;
455
456     if(analogRead(A2) > maxForce[3] and -RRLegPress < 50){
457         RRLegPress -= analogRead(A2) / maxForce[3];
458     }
459     else if(abs(RRLegPress) > 1){
460         RRLegPress += 1;
461     }
462     RRLegPos[1] += RRLegPress;
463 }
464
465 moveLegsPositions();
466
467 }
468
469 if(!package.standMode and oldStand != package.standMode){
470
471     int Bpos[12];
472     for(int i = 0; i < 3; i++){
473         Bpos[i] = FLLegPos[i];
474     }
475     for(int i = 0; i < 3; i++){
476         Bpos[i + 3] = FRLegPos[i];
477     }
478     for(int i = 0; i < 3; i++){
479         Bpos[i + 6] = RLLegPos[i];
480     }
481     for(int i = 0; i < 3; i++){
482         Bpos[i + 9] = RRLegPos[i];
483     }
484
485     int sitPoint[3] = {0,100,0};
486     int line[3];
487     for(int i = 0; i < 100; i++){
488         delay(5);
489         for(int v = 0; v < 3; v++){
490             line[v] = Bpos[v] - sitPoint[v];
491
492             FLLegPos[v] = Bpos[v] - line[v] * i / 100 ;
493         }
494
495         for(int v = 0; v < 3; v++){
496             line[v] = Bpos[v + 3] - sitPoint[v];
497
498             FRLegPos[v] = Bpos[v + 3] - line[v] * i / 100 ;
499         }
500
501         for(int v = 0; v < 3; v++){
502             line[v] = Bpos[v + 6] - sitPoint[v];
503
504             RLLegPos[v] = Bpos[v + 6] - line[v] * i / 100 ;
505         }
506
507         for(int v = 0; v < 3; v++){
508             line[v] = Bpos[v + 9] - sitPoint[v];
509
510             RRLegPos[v] = Bpos[v + 9] - line[v] * i / 100 ;
511         }
512
513         moveLegsPositions();
514     }
515 }
516
517 double Sy = 200 - map(joystick[5], 0, 1023, 50, 0);
518 if(package.standMode and package.walkMode){
519     double Wx = map(joystick[1], 0, 1023, maxStepX, -maxStepX);
520     double Wz = map(joystick[0], 0, 1023, -maxStepZ, maxStepZ);
521
522     double Fz = map(joystick[3], 0, 1023, -maxStepZ, maxStepZ) + Wz;
523     double Rz = -Fz + 2*Wz;
524
525     if(package.Lidar){
526         Serial.println(xForce.bigNum);
527         Serial.println(yForce.bigNum);
528         Serial.println(" ");
529
530         if(abs(xForce.bigNum) < disLimit[0]){

```


J

Joystick verilerini, adım uzunluğu ve adım yönü verilerine oranlar

Lidar Modülünden gelen Engel verileri Adım Yönü verilerine yansıtılır.

Bacağın Hareket evresi 1. safhada ise

Bacağın, verilen robot yüksekliğinden ve eksi adım uzunluğundan, adım yüksekliği ve adım uzunluğuna kadar hesaplanır, o anki pozisyonları diziye yazılır.

Bacağın Hareket evresi 2. safhada ise

Bacağın, verilen adım yüksekliğinden ve adım uzunluğundan, robot yüksekliğine ve adım uzunluğuna kadar hesaplanır, o anki pozisyonları diziye yazılır

```

530     Wx += 15 * (abs(xForce.bigNum) + (xForce.bigNum - 1))
531   }
532   if(abs(yForce.bigNum) < disLimit[1]){
533     Wz += 15 * (abs(xForce.bigNum) + (xForce.bigNum - 1))
534   }
535 }
536
537 if(sqrt(sq(Wx) + sq(Wz)) > 10 or abs(Fz) > 10){
538   FLlegPos[0] -= stepOffset;
539   FRlegPos[0] -= stepOffset;
540   RLlegPos[0] -= stepOffset;
541   RRlegPos[0] -= stepOffset;
542 }
543
544 FLlegPos[1] -= tilt;
545 RLlegPos[1] -= tilt;
546 if(0 <= sTime[0] / config.path and sTime[0] / config.path < hoverTime / 2){
547   FLlegPos[0] = FLlegPos[0] + ((Wx - FLlegPos[0]) * (1 / ((config.path * (hoverTime /
548     2)) - sTime[0])));
549   FLlegPos[1] = tilt + FLlegPos[1] + ((Sy - config.stepH - FLlegPos[1]) * (1 /
550     ((config.path * (hoverTime / 2)) - sTime[0])));
551   FLlegPos[2] = FLlegPos[2] + ((Fz - FLlegPos[2]) * (1 / ((config.path * (hoverTime /
552     2)) - sTime[0])));
553 }
554 if(hoverTime / 2 <= sTime[0] / config.path and sTime[0] / config.path < hoverTime){
555   FLlegPos[0] = Wx;
556   FLlegPos[1] = tilt + FLlegPos[1] + ((Sy - FLlegPos[1]) * (1 / ((config.path *
557     hoverTime / 2) - (sTime[0] - (hoverTime / 2) * config.path))));
558   FLlegPos[2] = Fz;
559 }
560 if(hoverTime <= sTime[0] / config.path and sTime[0] / config.path < 2 * pushTime){
561   FLlegPos[0] = FLlegPos[0] + ((-Wx - FLlegPos[0]) * (1 / ((config.path * (pushTime
562     + stepDelay)) - (sTime[0] - hoverTime * config.path))));
563   FLlegPos[1] = tilt + FLlegPos[1] + ((Sy - FLlegPos[1]) * (1 / ((config.path *
564     (pushTime + stepDelay)) - (sTime[0] - hoverTime * config.path))));
565   FLlegPos[2] = FLlegPos[2] + ((-Fz - FLlegPos[2]) * (1 / ((config.path * (pushTime
566     + stepDelay)) - (sTime[0] - hoverTime * config.path))));
567 }
568
569 if(0 <= sTime[1] / config.path and sTime[1] / config.path < hoverTime / 2){
570   FRlegPos[0] = FRlegPos[0] + ((Wx - FRlegPos[0]) * (1 / ((config.path * (hoverTime
571     / 2)) - sTime[1])));
572   FRlegPos[1] = FRlegPos[1] + ((Sy - config.stepH - FRlegPos[1]) * (1 / ((config.path
573     * (hoverTime / 2)) - sTime[1])));
574   FRlegPos[2] = FRlegPos[2] + ((-Fz - FRlegPos[2]) * (1 / ((config.path * (hoverTime
575     / 2)) - sTime[1])));
576 }
577 if(hoverTime / 2 <= sTime[1] / config.path and sTime[1] / config.path < hoverTime){
578   FRlegPos[0] = Wx;
579   FRlegPos[1] = FRlegPos[1] + ((Sy - FRlegPos[1]) * (1 / ((config.path * hoverTime
580     / 2) - (sTime[1] - (hoverTime / 2) * config.path))));
581   FRlegPos[2] = -Fz;
582 }
583 if(hoverTime <= sTime[1] / config.path and sTime[1] / config.path < 2 * pushTime){
584   FRlegPos[0] = FRlegPos[0] + ((-Wx - FRlegPos[0]) * (1 / ((config.path * (pushTime
585     + stepDelay)) - (sTime[1] - hoverTime * config.path))));
586   FRlegPos[1] = FRlegPos[1] + ((Sy - FRlegPos[1]) * (1 / ((config.path * (pushTime
587     + stepDelay)) - (sTime[1] - hoverTime * config.path))));
588   FRlegPos[2] = FRlegPos[2] + ((Fz - FRlegPos[2]) * (1 / ((config.path * (pushTime
589     + stepDelay)) - (sTime[1] - hoverTime * config.path))));
590 }
591 }
592
593 if(0 <= sTime[2] / config.path and sTime[2] / config.path < hoverTime / 2){
594   RLlegPos[0] = RLlegPos[0] + ((Wx - RLlegPos[0]) * (1 / ((config.path * (hoverTime
595     / 2)) - sTime[2])));
596   RLlegPos[1] = tilt + RLlegPos[1] + ((Sy - config.stepH - RLlegPos[1]) * (1 /
597     ((config.path * (hoverTime / 2)) - sTime[2])));
598   RLlegPos[2] = RLlegPos[2] + ((Rz - RLlegPos[2]) * (1 / ((config.path * (hoverTime
599     / 2)) - sTime[2])));
600 }
601 if(hoverTime / 2 <= sTime[2] / config.path and sTime[2] / config.path < hoverTime){
602   RLlegPos[0] = Wx;
603   RLlegPos[1] = tilt + RLlegPos[1] + ((Sy - RLlegPos[1]) * (1 / ((config.path *
604     hoverTime / 2) - (sTime[2] - (hoverTime / 2) * config.path))));

```

Ek 41 Bilgisayar Ünitesi Yazılımı ve Akış Şeması

(10)

J

Bacağın Hareket evresi 3. Sahnada ise

Bacağı, verilen robot yüksekliğinden ve adım uzunluğundan, robot yüksekliği ve eksi adım uzunluğuna kadar hesaplanır, o anki pozisyonları diziye yazılır.

Bacağın hareket evresini sıfırla.

Bacağın hareket evresine bir ekle.

Kinematik model fonksiyonu çalıştırılır ve Bacak Pozisyonları ayarlanır.

Fonksiyonlar

```

605     RLLegPos[2] = Rz;
606 }
607 if(hoverTime <= sTime[2] / config.path and sTime[2] / config.path < 2 * pushTime){
608     RLLegPos[0] = RLLegPos[0] + ((-Wx - RLLegPos[0]) * (1 / ((config.path * (pushTime
609     + stepDelay)) - (sTime[2] - hoverTime * config.path))));
610     RLLegPos[1] = tilt + RLLegPos[1] + ((Sy - RLLegPos[1]) * (1 / ((config.path *
611     (pushTime + stepDelay)) - (sTime[2] - hoverTime * config.path))));
612     RLLegPos[2] = RLLegPos[2] + ((-Rz - RLLegPos[2]) * (1 / ((config.path * (pushTime
613     + stepDelay)) - (sTime[2] - hoverTime * config.path))));
614 }
615
616
617 if(0 <= sTime[3] / config.path and sTime[3] / config.path < hoverTime / 2){
618     RRLegPos[0] = RRLegPos[0] + ((Wx - RRLegPos[0]) * (1 / ((config.path * (hoverTime
619     / 2)) - sTime[3])));
620     RRLegPos[1] = RRLegPos[1] + ((Sy - config.stepH - RRLegPos[1]) * (1 /
621     ((config.path * (hoverTime / 2)) - sTime[3])));
622     RRLegPos[2] = RRLegPos[2] + ((-Wz - RRLegPos[2]) * (1 / ((config.path * (hoverTime
623     / 2)) - sTime[3])));
624 }
625 if(hoverTime / 2 <= sTime[3] / config.path and sTime[3] / config.path < hoverTime){
626     RRLegPos[0] = Wx;
627     RRLegPos[1] = RRLegPos[1] + ((Sy - RRLegPos[1]) * (1 / ((config.path * hoverTime
628     / 2) - (sTime[3] - (hoverTime / 2) * config.path))));
629     RRLegPos[2] = -Wz;
630 }
631 if(hoverTime <= sTime[3] / config.path and sTime[3] / config.path < 2 * pushTime){
632     RRLegPos[0] = RRLegPos[0] + ((-Wx - RRLegPos[0]) * (1 / ((config.path * (pushTime
633     + stepDelay)) - (sTime[3] - hoverTime * config.path))));
634     RRLegPos[1] = RRLegPos[1] + ((Sy - RRLegPos[1]) * (1 / ((config.path * (pushTime
635     + stepDelay)) - (sTime[3] - hoverTime * config.path))));
636     RRLegPos[2] = RRLegPos[2] + ((Wz - RRLegPos[2]) * (1 / ((config.path * (pushTime
637     + stepDelay)) - (sTime[3] - hoverTime * config.path))));
638 }
639
640 FLLegPos[0] += stepOffset;
641 FRLegPos[0] += stepOffset;
642 RLLegPos[0] += stepOffset;
643 RRLegPos[0] += stepOffset;
644
645 for(int i = 0; i < 4; i++){
646     Serial.print(i);
647     Serial.print(" ");
648     Serial.println(sTime[i]);
649     if((sTime[i] / config.path >= 2 * pushTime and !package.WalkMode) or (sTime[i]
650     / config.path >= 5 * pushTime and package.WalkMode)){
651         sTime[i] = 0;
652     }
653     else{
654         sTime[i] ++;
655     }
656 }
657 }
658 else{
659     FLLegPos[1] = Sy + tilt;
660     FRLegPos[1] = Sy;
661     RLLegPos[1] = Sy + tilt;
662     RRLegPos[1] = Sy;
663     if(package.WalkMode){
664         for(int i = 0; i < 4; i++){
665             sTime[i] = W4s[i] * config.path;
666         }
667     }
668     else{
669         for(int i = 0; i < 4; i++){
670             sTime[i] = W2s[i] * config.path;
671         }
672     }
673 }
674
675 moveLegsPositions();
676 }
677
678 oldStand = package.standMode;
679 }
680
681 void moveLegsPositions(){
682     double Hz = sqrt(sq(FLLegPos[2] + Leg1) + sq(FLLegPos[1]));
683     double Hx = sqrt(sq(FLLegPos[0]) + sq(Hz));

```

FONKSİYONLAR

```

684
685 double a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
686 double a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2)) * 57296 /
687 1000 - asin(FLLegPos[0] / Hx) * 57296 / 1000;
688 double a1 = atan((FLLegPos[2] + Leg1) / FLLegPos[1]) * 57296 / 1000 - atan(Leg1 /
689 FLLegPos[1]) * 57296 / 1000;
690
691 FLLeg[2].write(map(a3, 35, 135, 180, 0) + FLLegAdd[2]);
692 FLLeg[1].write(a2 + FLLegAdd[1]);
693 FLLeg[0].write(180 - (90 - a1) + FLLegAdd[0]);
694
695 Hz = sqrt(sq(FRLegPos[2] + Leg1) + sq(FRLegPos[1]));
696 Hx = sqrt(sq(FRLegPos[0]) + sq(Hz));
697
698 a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
699 a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2)) * 57296 / 1000 -
700 asin(FRLegPos[0] / Hx) * 57296 / 1000;
701 a1 = atan((FRLegPos[2] + Leg1) / FRLegPos[1]) * 57296 / 1000 - atan(Leg1 /
702 FRLegPos[1]) * 57296 / 1000;
703
704 FRLeg[2].write(map(a3, 35, 135, 0, 180) + FRLegAdd[2]);
705 FRLeg[1].write(180 - a2 + FRLegAdd[1]);
706 FRLeg[0].write(180 - (90 + a1) + FRLegAdd[0]);
707
708 Hz = sqrt(sq(RLLegPos[2] + Leg1) + sq(RLLegPos[1]));
709 Hx = sqrt(sq(RLLegPos[0]) + sq(Hz));
710
711 a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
712 a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2)) * 57296 / 1000 -
713 asin(RLLegPos[0] / Hx) * 57296 / 1000;
714 a1 = atan((RLLegPos[2] + Leg1) / RLLegPos[1]) * 57296 / 1000 - atan(Leg1 /
715 RLLegPos[1]) * 57296 / 1000;
716
717 RLLeg[2].write(map(a3, 35, 135, 180, 0) + RLLegAdd[2]);
718 RLLeg[1].write(a2 + RLLegAdd[1]);
719 RLLeg[0].write((90 - a1) + RLLegAdd[0]);
720
721 Hz = sqrt(sq(RRLegPos[2] + Leg1) + sq(RRLegPos[1]));
722 Hx = sqrt(sq(RRLegPos[0]) + sq(Hz));
723
724 a3 = acos( (sq(Leg2) + sq(Leg3) - sq(Hx)) / (2 * Leg2 * Leg3) ) * 57296 / 1000;
725 a2 = 90 + acos( (sq(Hx) + sq(Leg2) - sq(Leg3)) / (2 * Hx * Leg2)) * 57296 / 1000 -
726 asin(RRLegPos[0] / Hx) * 57296 / 1000;
727 a1 = atan((RRLegPos[2] + Leg1) / RRLegPos[1]) * 57296 / 1000 - atan(Leg1 /
728 RRLegPos[1]) * 57296 / 1000;
729
730 RRLeg[2].write(map(a3, 35, 135, 0, 180) + RRLegAdd[2]);
731 RRLeg[1].write(180 - a2 + RRLegAdd[1]);
732 RRLeg[0].write((90 + a1) + RRLegAdd[0]);
733 }
734
735 void getPackage(){
736 for(int i = 0; i < 5; i++){
737 if ( radio.available() ) {
738 i = 5;
739 radio.read(&package, sizeof(package));
740 }
741 }
742 if(package.restoreConfig){
743 setConfig();
744 }
745 }
746
747 void getJoystick(){
748
749 for(int ax = 0; ax < 6; ax++){
750 joystick[ax] = package.joystick[ax];
751 }
752
753 for(int j = 0; j < 6; j++){
754 if(joystick[j] - oldJoystick[j] > config.joystickSmoothness){
755 delay(config.joystickSmoothness/10);
756 joystick[j] = oldJoystick[j] +config.joystickSmoothness;
757 }
758 else if(joystick[j] - oldJoystick[j] < -config.joystickSmoothness){
759 delay(config.joystickSmoothness/10);
760 joystick[j] = oldJoystick[j] -config.joystickSmoothness;
761 }

```

FONKSİYONLAR

```
762     oldJoystick[j] = joystick[j];
763     }
764 }
765
766 void moveTwoLeg(){
767     FLleg[0].write(map(joystick[2],0,1023,45,180));
768     FLleg[1].write(map(joystick[0],0,1023,0,180));
769     FLleg[2].write(map(joystick[1],0,1023,0,180));
770     FRleg[0].write(180-map(joystick[5],0,1023,45,180));
771     FRleg[1].write(map(joystick[3],0,1023,0,180));
772     FRleg[2].write(map(joystick[4],0,1023,0,180));
773 }
774
775 void addToServos(){
776     FLleg[0].write(FLleg[0].read() + 0);
777     FLleg[1].write(FLleg[1].read() + 0);
778     FLleg[2].write(FLleg[2].read() + 0);
779
780     FRleg[0].write(FRleg[0].read() + 0);
781     FRleg[1].write(FRleg[1].read() + 0);
782     FRleg[2].write(FRleg[2].read() + 0);
783
784     RLleg[0].write(RLleg[0].read() + 0);
785     RLleg[1].write(RLleg[1].read() + 0);
786     RLleg[2].write(RLleg[2].read() + 0);
787
788     RRleg[0].write(RRleg[0].read() + 0);
789     RRleg[1].write(RRleg[1].read() + 0);
790     RRleg[2].write(RRleg[2].read() + 0);
791 }
792
793 void setLegstoFL(){
794     FRlegPos[1] = FLlegPos[1];
795     FRlegPos[0] = FLlegPos[0];
796     FRlegPos[2] = FLlegPos[2];
797
798     RLlegPos[1] = FLlegPos[1];
799     RLlegPos[0] = FLlegPos[0];
800     RLlegPos[2] = FLlegPos[2];
801
802     RRlegPos[1] = FLlegPos[1];
803     RRlegPos[0] = FLlegPos[0];
804     RRlegPos[2] = FLlegPos[2];
805 }
806
807 void loadConfiguration(const char *filename, Config &config) {
808     File file = SD.open(filename);
809     StaticJsonDocument<512> doc;
810     DeserializationError error = deserializeJson(doc, file);
811     if (error)
812         Serial.println(F("Failed to read file, using default configuration"));
813
814     config.joystickSmoothness = doc["joystickSmoothness"];
815     config.path = doc["path"];
816     config.stepH = doc["stepH"];
817     config.lidarSwitch = doc["lidarSwith"];
818     config.gyroAssist = doc["gyroAssist"];
819     config.droneSwitch = doc["droneSwith"];
820     config.forceAssist = doc["forceAssist"];
821
822     file.close();
823 }
824
825 void restoreConfig(){
826     while(!radio.write(&config, sizeof(config))){}
827 }
828
829 void setConfig(){
830     config.joystickSmoothness = package.joystickSmoothness;
831
832     for(int i = 0; i < 2; i++){
833         sTime[i] /= config.path;
834     }
835     config.path = package.path;
836     if(package.WalkMode){
837         for(int i = 0; i < 4; i++){
838             sTime[i] = W4s[i];
839         }
840     }
841 }
```


FONKSİYONLAR

```
840     }
841     else{
842         for(int i = 0; i < 4; i++){
843             sTime[i] = W2s[i];
844         }
845     }
846     for(int i = 0; i < 2; i++){
847         sTime[i] *= config.path;
848     }
849     config.stepH = package.stepH;
850     config.lidarSwitch = package.lidarSwitch;
851     config.gyroAssist = package.gyroAssist;
852     config.forceAssist = package.forceAssist;
853     config.droneSwitch = package.droneSwitch;
854     config.WalkMode = package.WalkMode;
855
856     saveConfiguration(filename, config);
857 }
858
859 void saveConfiguration(const char *filename, const Config &config) {
860     SD.remove(filename);
861
862     File file = SD.open(filename, FILE_WRITE);
863     if (!file) {
864         Serial.println(F("Failed to create file"));
865         return;
866     }
867     StaticJsonDocument<256> doc;
868
869     doc["joystickSmoothness"] = config.joystickSmoothness;
870     doc["path"] = config.path;
871     doc["stepH"] = config.stepH;
872     doc["lidarSwitH"] = config.lidarSwitch;
873     doc["gyroAssist"] = config.gyroAssist;
874     doc["droneSwitH"] = config.droneSwitch;
875     doc["forceAssist"] = config.forceAssist;
876
877
878     if (serializeJson(doc, file) == 0) {
879         Serial.println(F("Failed to write to file"));
880     }
881     file.close();
882 }
883
884
885 void getGyro(){
886     gyro[0] = -mpu.getAngleX();
887     gyro[1] = -mpu.getAngleY();
888     gyro[2] = -mpu.getAngleZ();
889 }
890
891 void getTemperature(){
892     Serial.println(mpu.getTemp());
893 }
894
895 void getLidar(){
896     Wire1.requestFrom(LidarAdress, 4);
897     byte gasVar[5];
898     int i = 0;
899     int d = 5;
900     while(Wire1.available()){
901         xForce.array[i] = Wire1.read();
902         i++;
903     }delay(d);
904
905     Wire1.requestFrom(LidarAdress, 4);
906     i = 0;
907     while(Wire1.available()){
908         yForce.array[i] = Wire1.read();
909         i++;
910     }delay(d);
911 }
912 }
913
914 void getGas(){
915     Wire1.requestFrom(GasAdress, 4);
916     Serial.println(" ");
917     byte gasVar[5];
918     int i = 0;
```

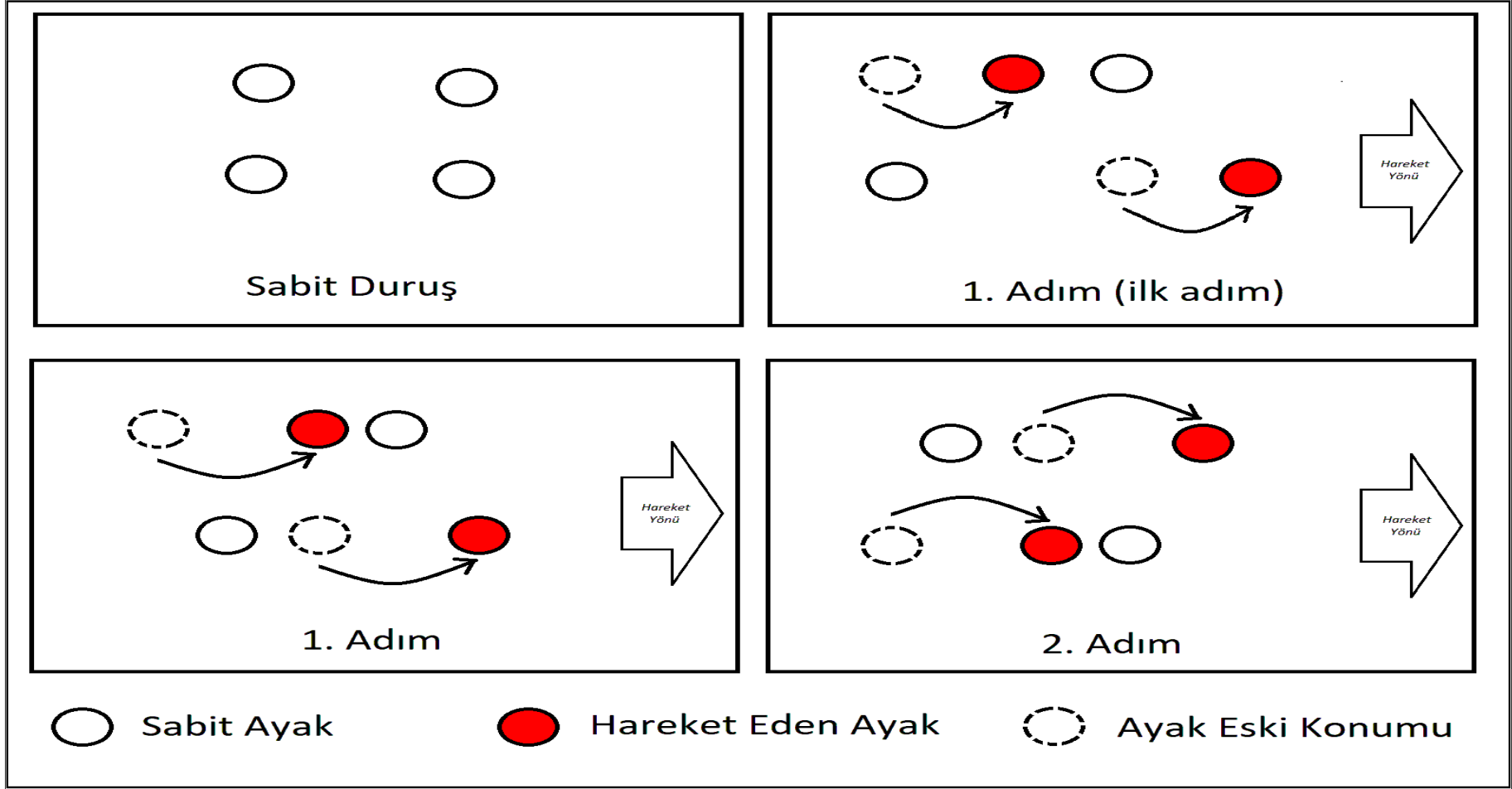
FONKSİYONLAR

```
919     int d = 5;
920     while(Wire1.available()){
921         H2.array[i] = Wire1.read();
922         i++;
923     }delay(d);
924     Serial.println(H2.bigNum);
925
926     Wire1.requestFrom(GasAdress, 4);
927     i = 0;
928     while(Wire1.available()){
929         LPG.array[i] = Wire1.read();
930         i++;
931     }delay(d);
932     Serial.println(LPG.bigNum);
933
934     Wire1.requestFrom(GasAdress, 4);
935     i = 0;
936     while(Wire1.available()){
937         CH4.array[i] = Wire1.read();
938         i++;
939     }delay(d);
940     Serial.println(CH4.bigNum);
941
942     Wire1.requestFrom(GasAdress, 4);
943     i = 0;
944     while(Wire1.available()){
945         CO.array[i] = Wire1.read();
946         i++;
947     }delay(d);
948     Serial.println(CO.bigNum);
949
950     Wire1.requestFrom(GasAdress, 4);
951     i = 0;
952     while(Wire1.available()){
953         Alchol.array[i] = Wire1.read();
954         i++;
955     }delay(d);
956     Serial.println(Alchol.bigNum);
957
958     gasPackage.H2 = H2.bigNum;
959     gasPackage.LPG = LPG.bigNum;
960     gasPackage.CH4 = CH4.bigNum;
961     gasPackage.CO = CO.bigNum;
962     gasPackage.Alchol = Alchol.bigNum;
963 }
964
965 void standingMove(){
966     double r = sqrt(sq(LegtoOrg[0]) + sq(Yin));
967     double A1 = degrees(acos((sq(r) + sq(Yin) - sq(LegtoOrg[0])) / (2 * r * Yin)));
968
969     double Ar = 270 - Ay - A1;
970     double Af = 270 + Ay - A1;
971
972     double Yr = r * sin(radians(Ar)) + sin(radians(Ax)) * Xin + sin(radians(Az)) * Zin;
973     double Yf = r * sin(radians(Af)) + sin(radians(Ax)) * Xin + sin(radians(Az)) * Zin;
974
975     double Xf = LegtoOrg[0] + r * cos(radians(Af));
976     double Xr = LegtoOrg[0] + r * cos(radians(Ar));
977
978     double Xz = cos(radians(32.34 + Az)) * sqrt(sq(LegtoOrg[1]) + sq(LegtoOrg[0])) - 150;
979     double Zz = sin(radians(32.34 + Az)) * sqrt(sq(LegtoOrg[1]) + sq(LegtoOrg[0])) - 95;
980
981     r = sqrt(sq(LegtoOrg[1]) + sq(Yf));
982     A1 = degrees(acos((sq(r) + sq(Yf) - sq(LegtoOrg[1])) / (2 * r * Yf)));
983
984     double Ari = 270 - Ax - A1;
985     double Ale = 270 + Ax - A1;
986
987     double Yri = r * sin(radians(Ari));
988     double Yle = r * sin(radians(Ale));
989
990     double Zri = LegtoOrg[1] + r * cos(radians(Ari));
991     double Zle = LegtoOrg[1] + r * cos(radians(Ale));
```

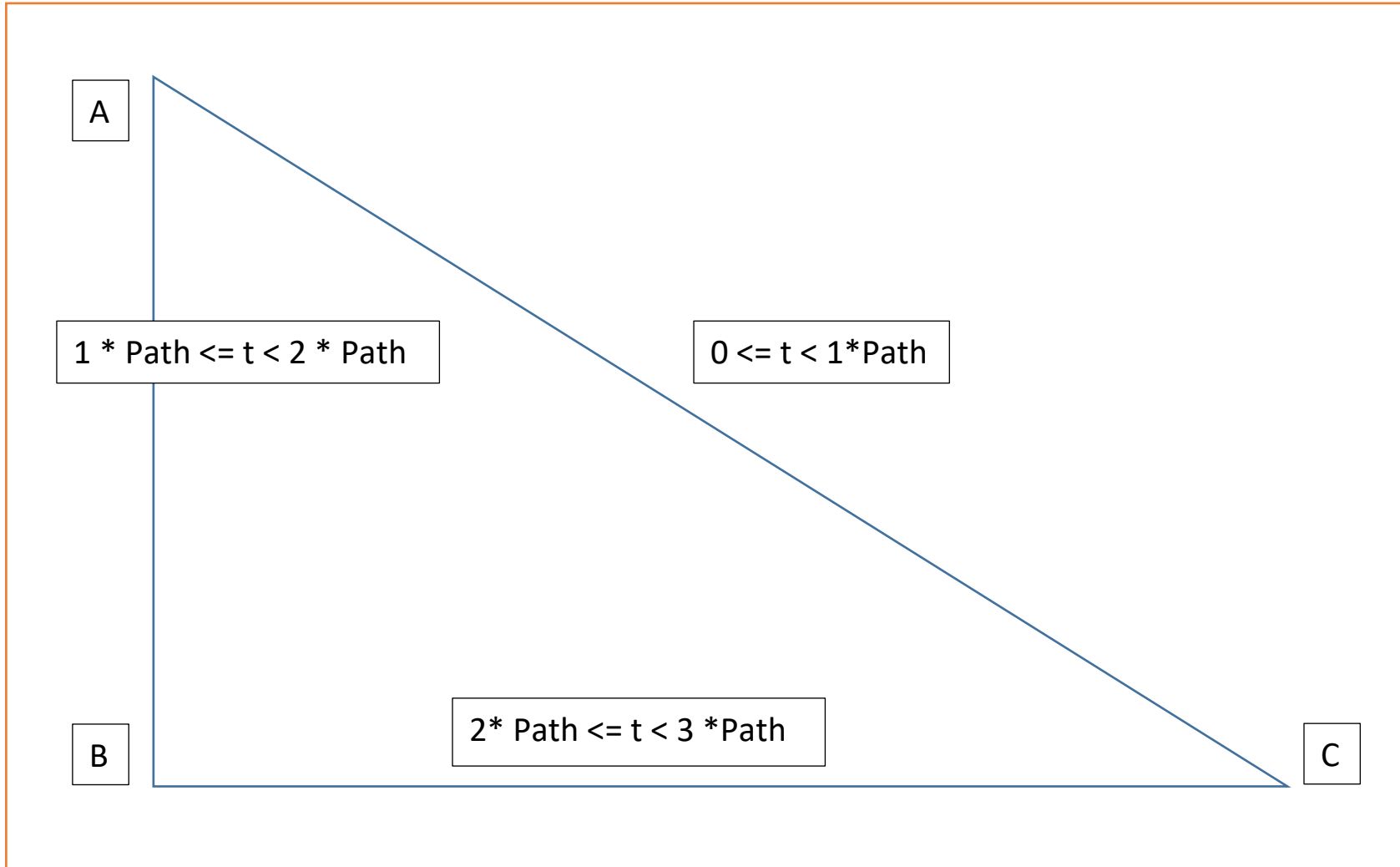
FONKSİYONLAR

```
992
993   FLLegPos[0] = -Xf + cos(radians(Ax)) * Xin + Xz;
994   FLLegPos[1] = Yle;
995   FLLegPos[2] = -Zle - cos(radians(Az)) * Zin + Zz;
996
997   FRLegPos[0] = -Xf + cos(radians(Ax)) * Xin - Xz;
998   FRLegPos[1] = Yri;
999   FRLegPos[2] = -Zri + cos(radians(Az)) * Zin - Zz;
1000
1001   r = sqrt(sq(LegtoOrg[1]) + sq(Yr));
1002   Al = degrees(acos((sq(r) + sq(Yr) - sq(LegtoOrg[1])) / (2 * r * Yr)));
1003
1004   Ari = 270 - Ax - Al;
1005   Ale = 270 + Ax - Al;
1006
1007   Yri = r * sin(radians(Ari));
1008   Yle = r * sin(radians(Ale));
1009
1010   Zri = LegtoOrg[1] + r * cos(radians(Ari));
1011   Zle = LegtoOrg[1] + r * cos(radians(Ale));
1012
1013   RLLegPos[0] = -Xf + cos(radians(Ax)) * Xin + Xz;
1014   RLLegPos[1] = Yle;
1015   RLLegPos[2] = -Zle - cos(radians(Az)) * Zin - Zz;
1016
1017   RRLegPos[0] = -Xf + cos(radians(Ax)) * Xin - Xz;
1018   RRLegPos[1] = Yri;
1019   RRLegPos[2] = -Zri + cos(radians(Az)) * Zin + Zz;
1020 }
1021
```

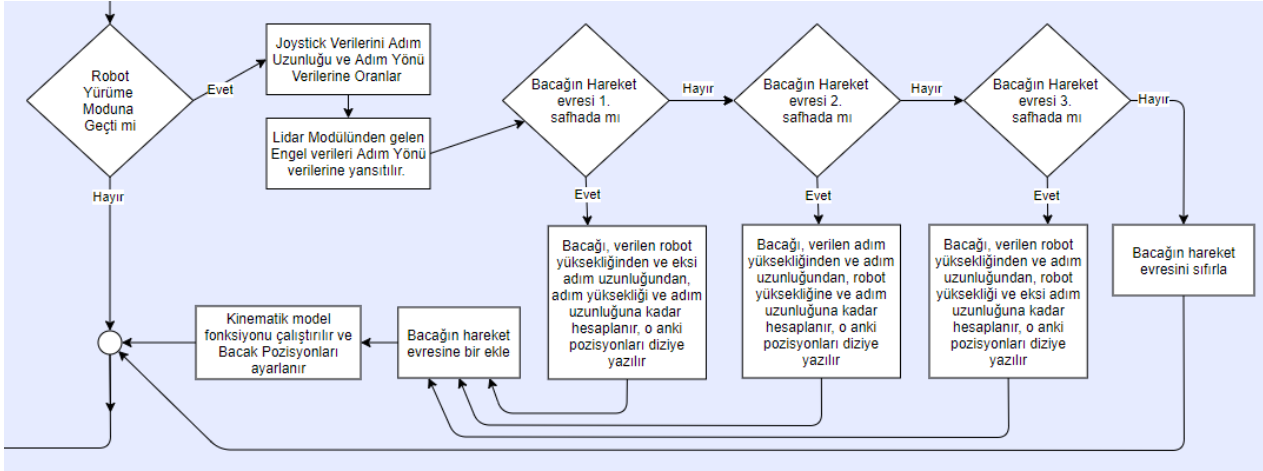
Ek 42 Tırıs Yürüme Modeli



Ek 43 Adımda Temel Bacak Hareketleri



Ek 44 Yürüme Yazılımı ve Akış Şeması (1)



```

518  if(package.standMode and package.walkMode){
519      double Wx = map(joystick[1], 0, 1023, maxStepX, -maxStepX);
520      double Wz = map(joystick[0], 0, 1023, -maxStepZ, maxStepZ);
521
522      double Fz = map(joystick[3], 0, 1023, -maxStepZ, maxStepZ) + Wz;
523      double Rz = -Fz + 2*Wz;
524
525      if(package.Lidar){
526          Serial.println(xForce.bigNum);
527          Serial.println(yForce.bigNum);
528          Serial.println(" ");
529
530          if(abs(xForce.bigNum) < disLimit[0]){
531              Wx += 15 * (abs(xForce.bigNum) + (xForce.bigNum - 1))
532          }
533          if(abs(yForce.bigNum) < disLimit[1]){
534              Wz += 15 * (abs(xForce.bigNum) + (xForce.bigNum - 1))
535          }
536      }
537
538      if(sqrt(sq(Wx) + sq(Wz)) > 10 or abs(Fz) > 10){
539          FLlegPos[0] -= stepOffset;
540          FRlegPos[0] -= stepOffset;
541          RLlegPos[0] -= stepOffset;
542          RRlegPos[0] -= stepOffset;
543
544          FLlegPos[1] -= tilt;
545          RLlegPos[1] -= tilt;
546          if(0 <= sTime[0] / config.path and sTime[0] / config.path < hoverTime / 2){
547              FLlegPos[0] = FLlegPos[0] + ((Wx - FLlegPos[0]) * (1 / ((config.path * (hoverTime / 2) - sTime[0]))));
548              FLlegPos[1] = tilt + FLlegPos[1] + ((Sy - config.stepH - FLlegPos[1]) * (1 / ((config.path * (hoverTime / 2) - sTime[0]))));
549              FLlegPos[2] = FLlegPos[2] + ((Fz - FLlegPos[2]) * (1 / ((config.path * (hoverTime / 2) - sTime[0]))));
550          }
551          if(hoverTime / 2 <= sTime[0] / config.path and sTime[0] / config.path < hoverTime){
552              FLlegPos[0] = Wx;
553              FLlegPos[1] = tilt + FLlegPos[1] + ((Sy - FLlegPos[1]) * (1 / ((config.path * hoverTime / 2) - (sTime[0] - (hoverTime / 2) * config.path))));
554              FLlegPos[2] = Fz;
555          }
556          if(hoverTime <= sTime[0] / config.path and sTime[0] / config.path < 2 * pushTime){
557              FLlegPos[0] = FLlegPos[0] + ((-Wx - FLlegPos[0]) * (1 / ((config.path * (pushTime + stepDelay)) - (sTime[0] - hoverTime * config.path))));
558              FLlegPos[1] = tilt + FLlegPos[1] + ((Sy - FLlegPos[1]) * (1 / ((config.path * (pushTime + stepDelay)) - (sTime[0] - hoverTime * config.path))));
559              FLlegPos[2] = FLlegPos[2] + ((-Fz - FLlegPos[2]) * (1 / ((config.path * (pushTime + stepDelay)) - (sTime[0] - hoverTime * config.path))));
560          }
561      }
562  }
563  }
564  }
565  }
566  }
567  }
568  }

```

Ek 44 Yürüme Yazılımı ve Akış Şeması (2)

```

569
570 ▼ if(0 <= sTime[1] / config.path and sTime[1] / config.path < hoverTime / 2){
571     FRLegPos[0] = FRLegPos[0] + ((Wx - FRLegPos[0]) * (1 / ((config.path * (hoverTime
572     / 2)) - sTime[1]]));
573     FRLegPos[1] = FRLegPos[1] + ((Sy - config.stepH - FRLegPos[1]) * (1 / ((config.path
574     * (hoverTime / 2)) - sTime[1]]));
575     FRLegPos[2] = FRLegPos[2] + ((-Fz - FRLegPos[2]) * (1 / ((config.path * (hoverTime
576     / 2)) - sTime[1]]));
577 }
578 ▼ if(hoverTime / 2 <= sTime[1] / config.path and sTime[1] / config.path < hoverTime){
579     FRLegPos[0] = Wx;
580     FRLegPos[1] = FRLegPos[1] + ((Sy - FRLegPos[1]) * (1 / ((config.path * hoverTime
581     / 2) - (sTime[1] - (hoverTime / 2) * config.path))));
582     FRLegPos[2] = -Fz;
583 }
584 ▼ if(hoverTime <= sTime[1] / config.path and sTime[1] / config.path < 2 * pushTime){
585     FRLegPos[0] = FRLegPos[0] + ((-Wx - FRLegPos[0]) * (1 / ((config.path * (pushTime
586     + stepDelay)) - (sTime[1] - hoverTime * config.path))));
587     FRLegPos[1] = FRLegPos[1] + ((Sy - FRLegPos[1]) * (1 / ((config.path * (pushTime
588     + stepDelay)) - (sTime[1] - hoverTime * config.path))));
589     FRLegPos[2] = FRLegPos[2] + ((Fz - FRLegPos[2]) * (1 / ((config.path * (pushTime
590     + stepDelay)) - (sTime[1] - hoverTime * config.path))));
591 }
592
593
594 ▼ if(0 <= sTime[2] / config.path and sTime[2] / config.path < hoverTime / 2){
595     RLLegPos[0] = RLLegPos[0] + ((Wx - RLLegPos[0]) * (1 / ((config.path * (hoverTime
596     / 2)) - sTime[2]]));
597     RLLegPos[1] = tilt + RLLegPos[1] + ((Sy - config.stepH - RLLegPos[1]) * (1 /
598     ((config.path * (hoverTime / 2)) - sTime[2]]));
599     RLLegPos[2] = RLLegPos[2] + ((Rz - RLLegPos[2]) * (1 / ((config.path * (hoverTime
600     / 2)) - sTime[2]]));
601 }
602 ▼ if(hoverTime / 2 <= sTime[2] / config.path and sTime[2] / config.path < hoverTime){
603     RLLegPos[0] = Wx;
604     RLLegPos[1] = tilt + RLLegPos[1] + ((Sy - RLLegPos[1]) * (1 / ((config.path *
605     hoverTime / 2) - (sTime[2] - (hoverTime / 2) * config.path))));
606     RLLegPos[2] = Rz;
607 }
608 ▼ if(hoverTime <= sTime[2] / config.path and sTime[2] / config.path < 2 * pushTime){
609     RLLegPos[0] = RLLegPos[0] + ((-Wx - RLLegPos[0]) * (1 / ((config.path * (pushTime
610     + stepDelay)) - (sTime[2] - hoverTime * config.path))));
611     RLLegPos[1] = tilt + RLLegPos[1] + ((Sy - RLLegPos[1]) * (1 / ((config.path *
612     (pushTime + stepDelay)) - (sTime[2] - hoverTime * config.path))));
613     RLLegPos[2] = RLLegPos[2] + ((-Rz - RLLegPos[2]) * (1 / ((config.path * (pushTime
614     + stepDelay)) - (sTime[2] - hoverTime * config.path))));
615 }
616
617
618 ▼ if(0 <= sTime[3] / config.path and sTime[3] / config.path < hoverTime / 2){
619     RRLegPos[0] = RRLegPos[0] + ((Wx - RRLegPos[0]) * (1 / ((config.path * (hoverTime
620     / 2)) - sTime[3]]));
621     RRLegPos[1] = RRLegPos[1] + ((Sy - config.stepH - RRLegPos[1]) * (1 /
622     ((config.path * (hoverTime / 2)) - sTime[3]]));
623     RRLegPos[2] = RRLegPos[2] + ((-Wz - RRLegPos[2]) * (1 / ((config.path * (hoverTime
624     / 2)) - sTime[3]]));
625 }
626 ▼ if(hoverTime / 2 <= sTime[3] / config.path and sTime[3] / config.path < hoverTime){
627     RRLegPos[0] = Wx;
628     RRLegPos[1] = RRLegPos[1] + ((Sy - RRLegPos[1]) * (1 / ((config.path * hoverTime
629     / 2) - (sTime[3] - (hoverTime / 2) * config.path))));
630     RRLegPos[2] = -Wz;
631 }
632 ▼ if(hoverTime <= sTime[3] / config.path and sTime[3] / config.path < 2 * pushTime){
633     RRLegPos[0] = RRLegPos[0] + ((-Wx - RRLegPos[0]) * (1 / ((config.path * (pushTime
634     + stepDelay)) - (sTime[3] - hoverTime * config.path))));
635     RRLegPos[1] = RRLegPos[1] + ((Sy - RRLegPos[1]) * (1 / ((config.path * (pushTime
636     + stepDelay)) - (sTime[3] - hoverTime * config.path))));
637     RRLegPos[2] = RRLegPos[2] + ((Wz - RRLegPos[2]) * (1 / ((config.path * (pushTime
638     + stepDelay)) - (sTime[3] - hoverTime * config.path))));
639 }
640
641     FLLegPos[0] += stepOffset;
642     FRLegPos[0] += stepOffset;

```

Ek 44 Yürüme Yazılımı ve Akış Şeması (3)

```
643     RLlegPos[0] += stepOffset;
644     RRlegPos[0] += stepOffset;
645
646     for(int i = 0; i < 4; i++){
647         Serial.print(i);
648         Serial.print(" ");
649         Serial.println(sTime[i]);
650         if((sTime[i] / config.path >= 2 * pushTime and !package.WalkMode) or (sTime[i]
651             / config.path >= 5 * pushTime and package.WalkMode)){
652             sTime[i] = 0;
653         }
654         else{
655             sTime[i] ++;
656         }
657     }
658 }
659 else{
660     FLlegPos[1] = Sy + tilt;
661     FRlegPos[1] = Sy;
662     RLlegPos[1] = Sy+ tilt;
663     RRlegPos[1] = Sy;
664     if(package.WalkMode){
665         for(int i = 0; i < 4; i++){
666             sTime[i] = W4s[i] * config.path;
667         }
668     }
669     else{
670         for(int i = 0; i < 4; i++){
671             sTime[i] = W2s[i] * config.path;
672         }
673     }
674 }
675
676 moveLegsPositions();
677 }
678
679 oldStand = package.standMode;
680 }
681 }
```


Ek 45 Kamera ve Ekran Kurulumu

Kontrol Kumandası

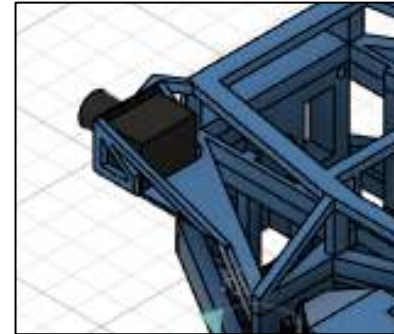
Model: FPV Monitör LCD Ekran
Kanal sayısı: 48
Çözünürlük: 480x272
Video formatı: NTSC/PAL
Güç tüketimi: 500mA
Çalışma süresi: 2 saat

Frekans aralığı: 5645MHz~5945MHz
Ekran boyutları: 4.3"
En boy oranı: 16:9
Çalışma voltajı: 5V
Dahili batarya: 600mAh
Frekans: 6 bant 48 kanal



Robot Köpek

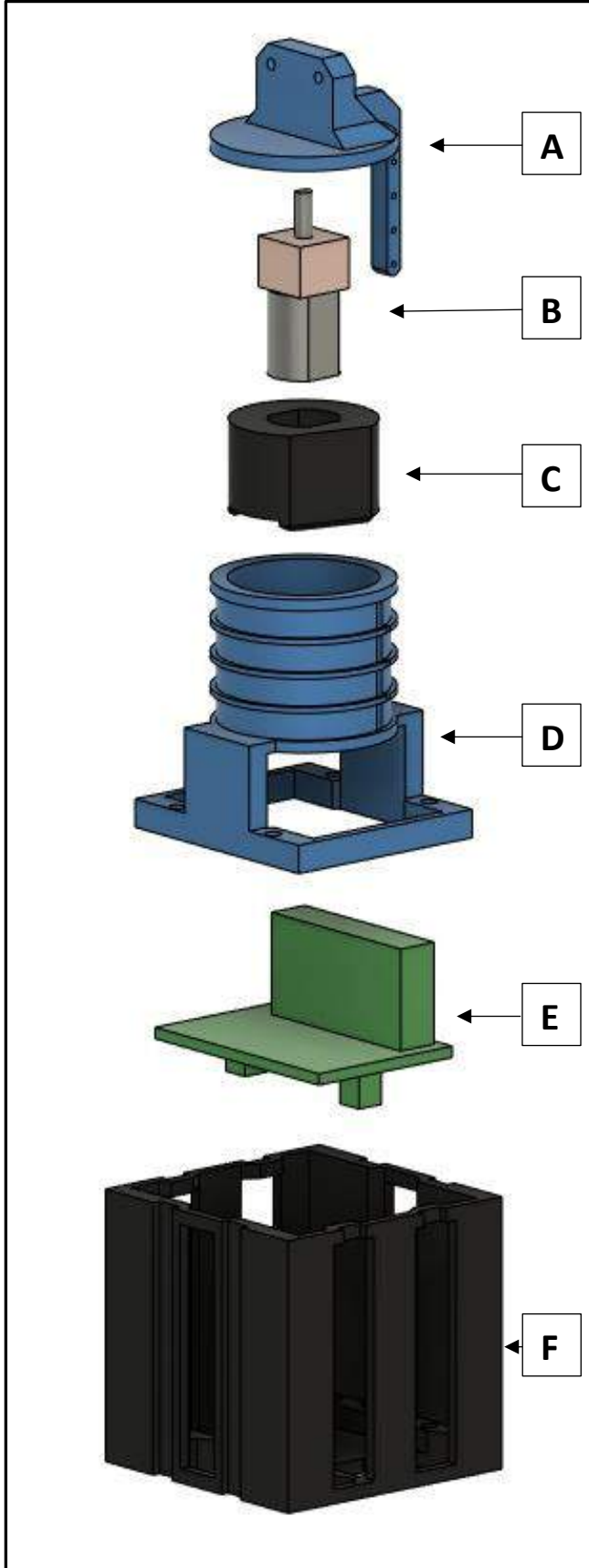
Model: TX01 AIO FPV VTX Kamera
Elektrik çıkış: 25mW
Elektrik giriş: 3.3-5V
Devre elektrik gereksinimi:
3.3-5V, 200mA
Kamera çözünürlüğü: 600TVL
Video formatı: NTSC
Aydınlatma: 1 lux
Görüş açısı: 120 derece
Anten boyutları: Nikel 22x29mm
Frekans: 5.8GHz 5 bant 40 kanal
Bant göstergesi LED Ekran
Ebatlar: 20x13x6mm
Ağırlık: 4.48g



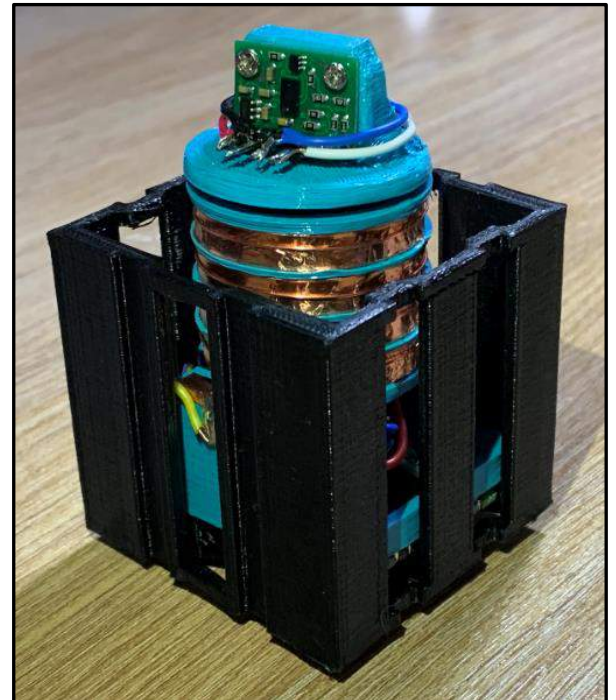
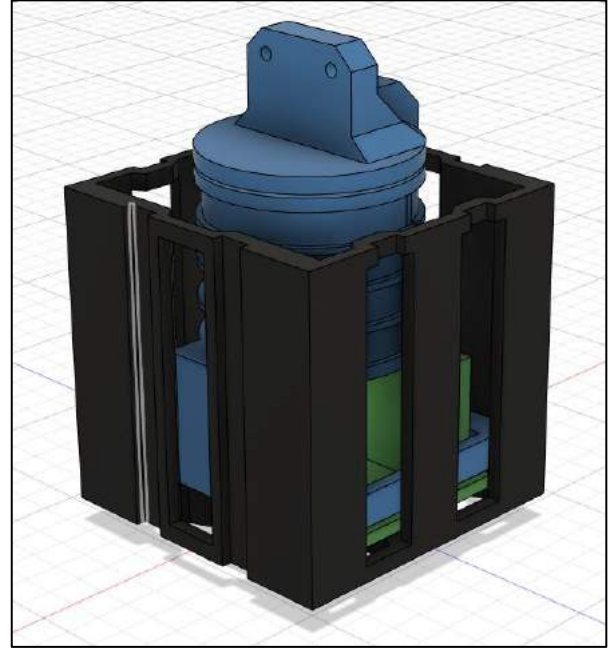
Ek 46 Lidar Modülü Malzeme Listesi

LİDAR MODÜLÜ MALZEME LİSTESİ			
		Malzeme	Adet
1		Arduino Pro Mini 328 - 3.3 V / 8 MHz (Header'lı)	1
2		VL53L0X Time-of-Flight Regülatörlü Mesafe Sensörü - 200 cm - PL2490	1
3		12V 12mm 60 RPM Redüktörlü Mikro DC Motor	1
4		20x30 Delikli Plaket	1
5		Filament 1.75 mm Turkuaz PLA - ABG	1
6		Filament 1.75 mm Siyah PLA - ABG	1
7		12mm Erkek Header	2
8		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
9		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
10		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	6
11		5mm Bakır Bant 30mt	1
12		1k Ohm Direnç	2
13		BC237 Transistör	1
14		1N4007 Diyot	1
15		100 nF Kondansatör	1

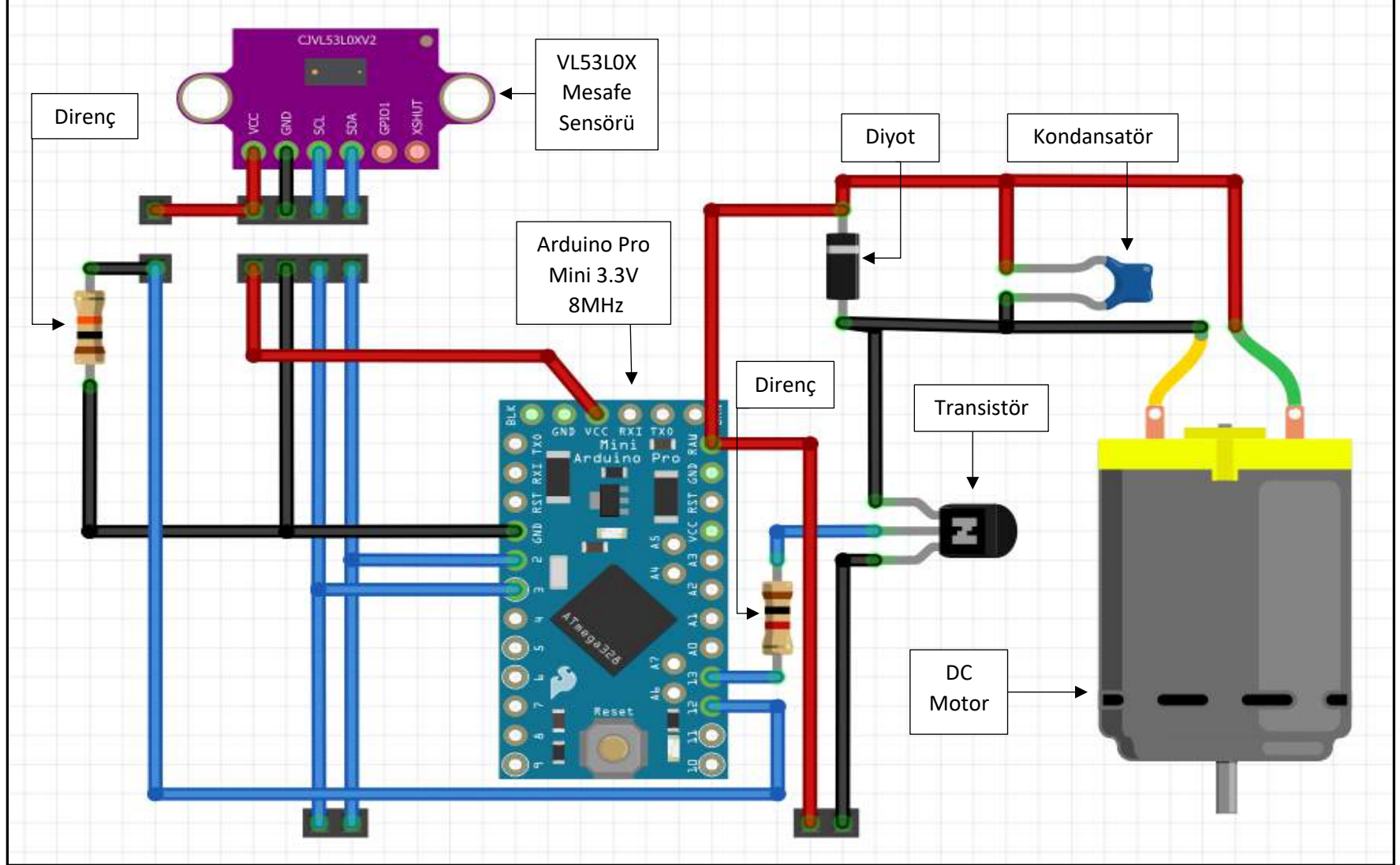
Ek 47 Lidar Modülü Tasarımı



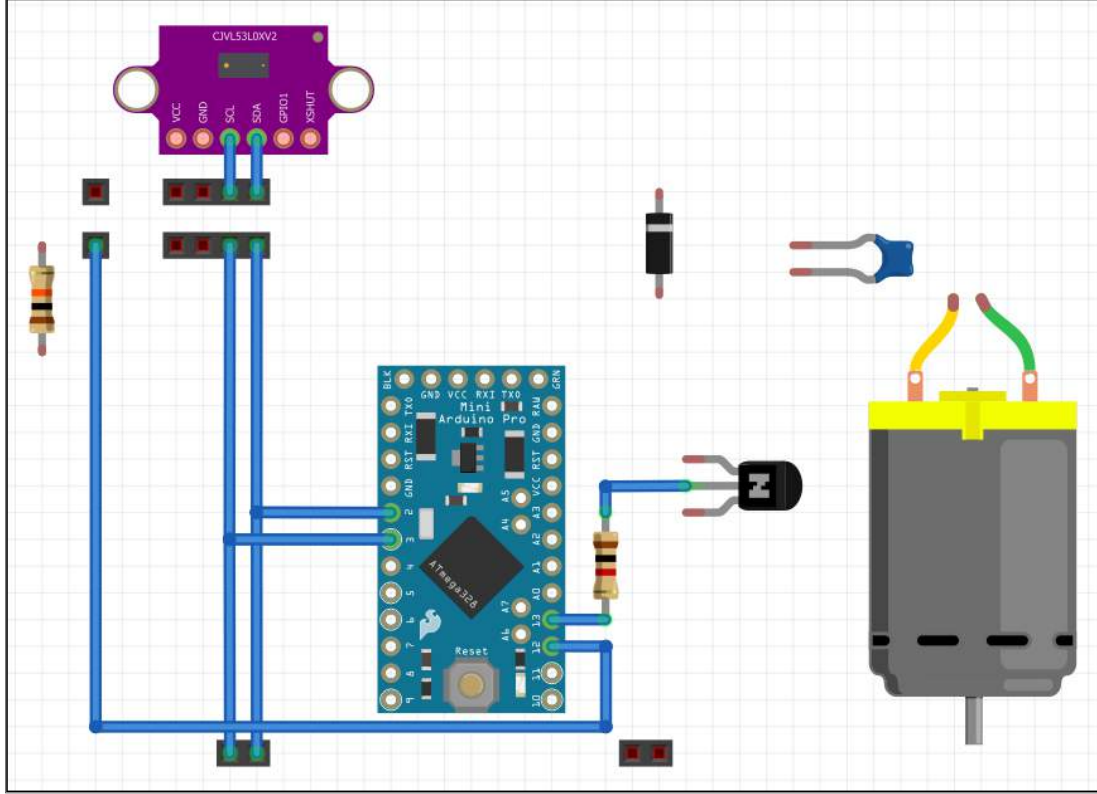
- A. Lidar Gözü Döner Tablası
- B. DC Motor
- C. DC Motor Sabitleyici Yuva
- D. Bakır Şeritli Taşıyıcı Blok
- E. Elektronik Devre
- F. Modül Kasası



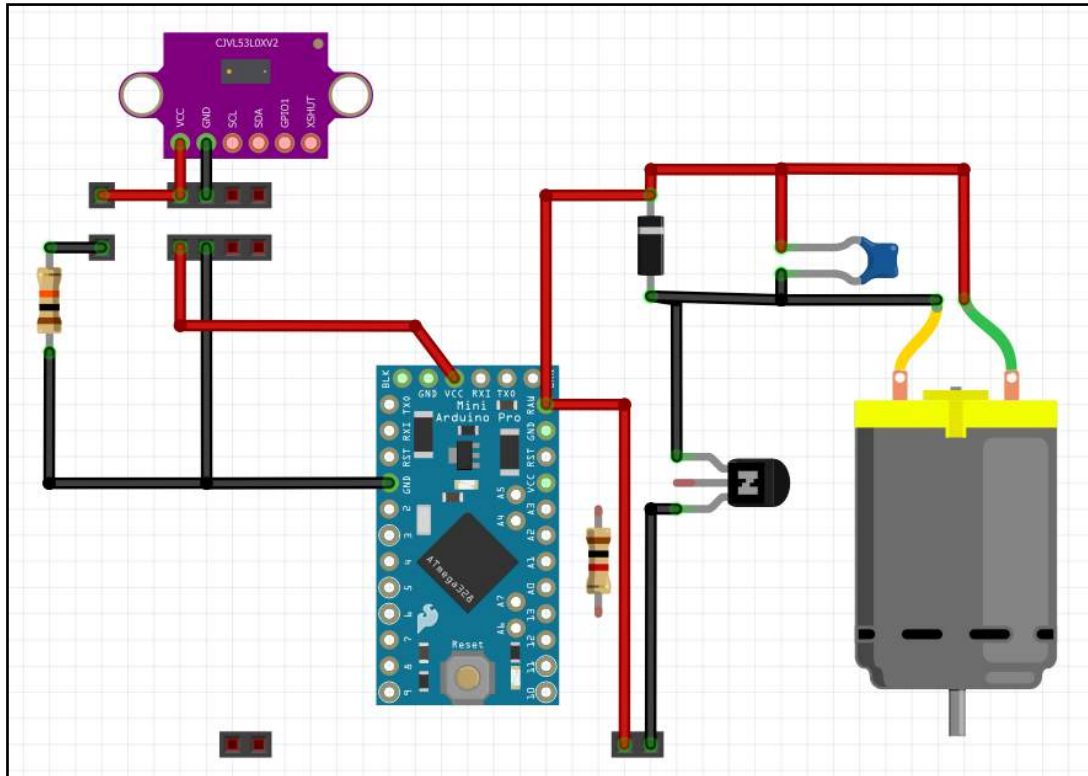
Ek 48 Lidar Modülü Elektrik ve Veri Dağıtım Şeması



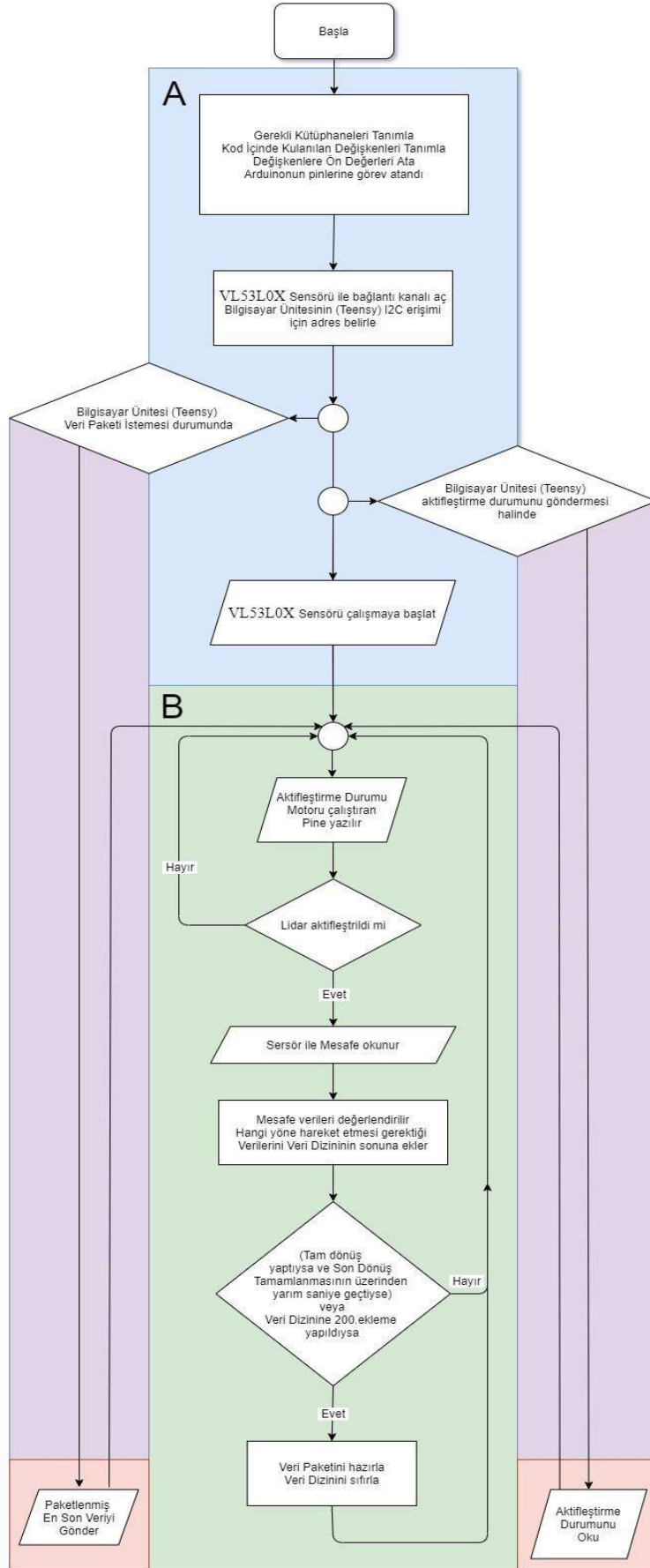
Lidar Modülü Veri Dağıtım Şeması



Lidar Modülü Elektrik Dağıtım Şeması



Ek 50 Lidar Modülü Yazılımı ve Akış Şeması (1)



Ek 50 Lidar Modülü Yazılımı ve Akış Şeması (2)

A

```
1  #include <VL53L0X.h>
2  #include <Wire.h>
3
4  double pastTime = 0;
5  bool R;
6  float Map[200];
7  float Sides[8];
8  float Force[2];
9  int req = 0;
10 double lastReq = 0;
11 int d = 0;
12 bool turn = 1;
13
14 union {
15     byte array[4];
16     float bigNum;
17 } xPackage;
18
19 union {
20     byte array[4];
21     float bigNum;
22 } yPackage;
23
24 int Adress = 9;
25
26 VL53L0X sensor;
27
28 void setup() {
29     Serial.begin(9600);
30     Wire.begin(Adress);
31
32     sensor.init();
33
34     pinMode(13, OUTPUT);
35     pinMode(12, INPUT);
36     Wire.onReceive(receiveEvent);
37     Wire.onRequest(requestEvent);
38
39     sensor.startContinuous();
40 }
```

B

```
42 void loop() {
43     R = digitalRead(12);
44     digitalWrite(13, turn);
45
46     if(millis() - lastReq > 200){
47         req = 0;
48     }
49
50     if(turn){
51         Map[d] = sensor.readRangeContinuousMillimeters();
52         d++;
53
54         if((R and millis() - pastTime > 500) or d == 200){
55             pastTime = millis();
56
57             for(int i = 0; i < 8; i++){
58                 for(int p = 0; p < d/8; p++){
59                     Sides[i] += Map[p + (d/8) * i];
60                 }
61                 Sides[i] /= d/8;
62             }
63             Force[1] = 0;
64             Force[0] = 0;
65
66             Force[1] += Sides[3];
67             Force[1] += Sides[4];
68             Force[1] -= Sides[0];
69             Force[1] -= Sides[7];
70
71             Force[0] += Sides[5];
72             Force[0] += Sides[6];
73             Force[0] -= Sides[1];
74             Force[0] -= Sides[2];
75
76             xPackage.bigNum = Force[0];
77             yPackage.bigNum = Force[1];
78             d = 0;
79         }
80     }
81 }
```

Ek 50 Lidar Modülü Yazılımı ve Akış Şeması (3)

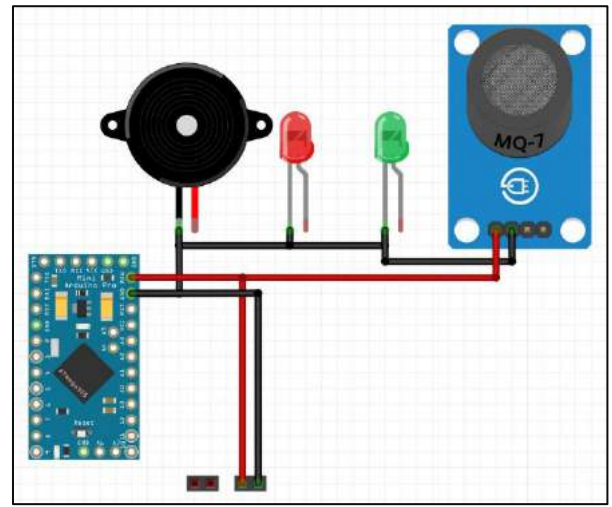
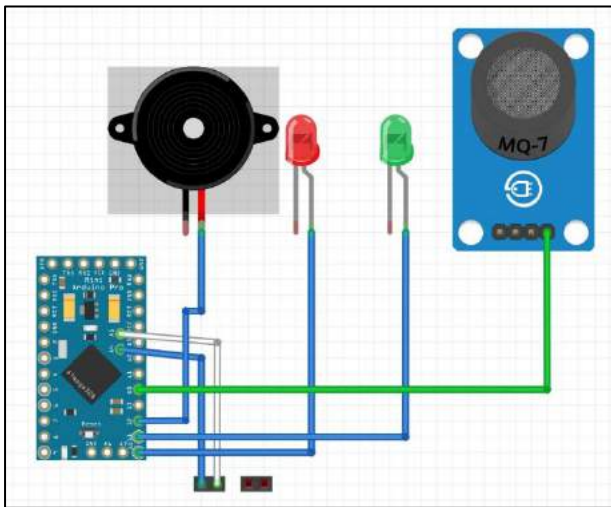
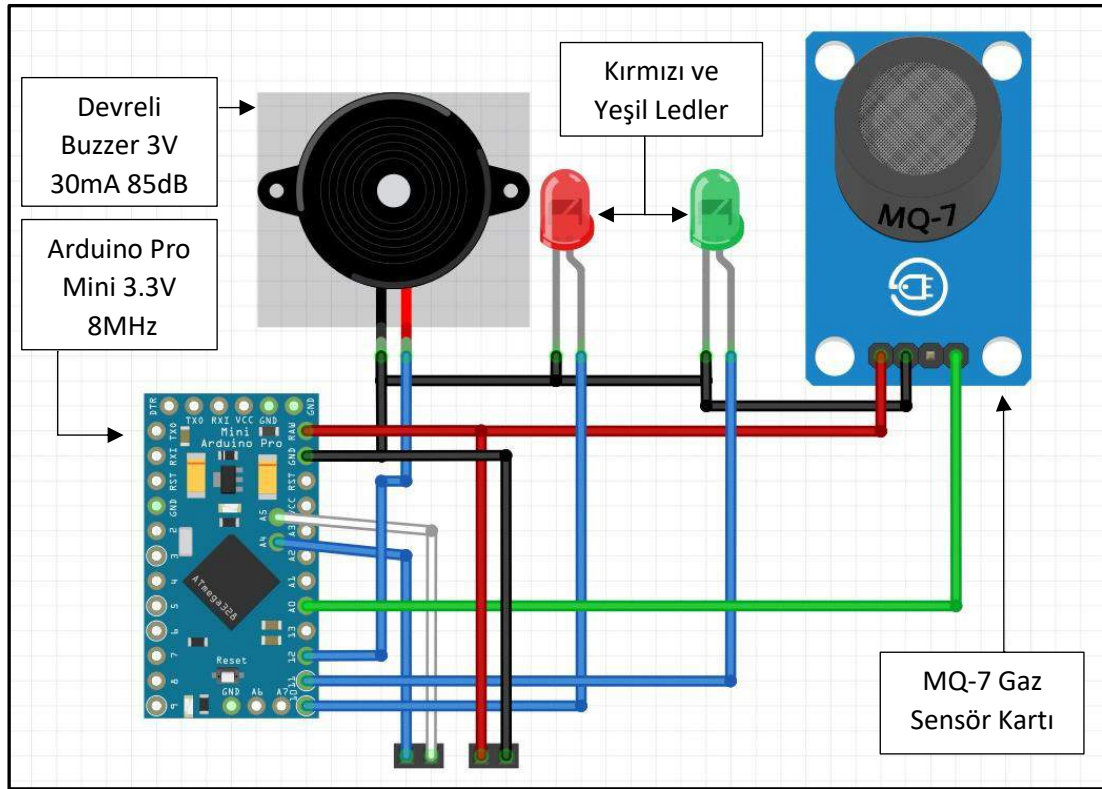
Fonksiyonlar

```
82
83 void receiveEvent() {
84     while(Wire.available()) {
85         turn = Wire.read();
86     }
87     Serial.println("Rec");
88 }
89
90 void requestEvent() {
91     lastReq = millis();
92     if(req == 0){
93         for(int i = 0; i < 4; i++){
94             Wire.write(xPackage.array[i]);
95         }
96         req = 1;
97     }
98     else if(req == 1){
99         for(int i = 0; i < 4; i++){
100             Wire.write(yPackage.array[i]);
101         }
102         req = 0;
103     }
104 }
105
```

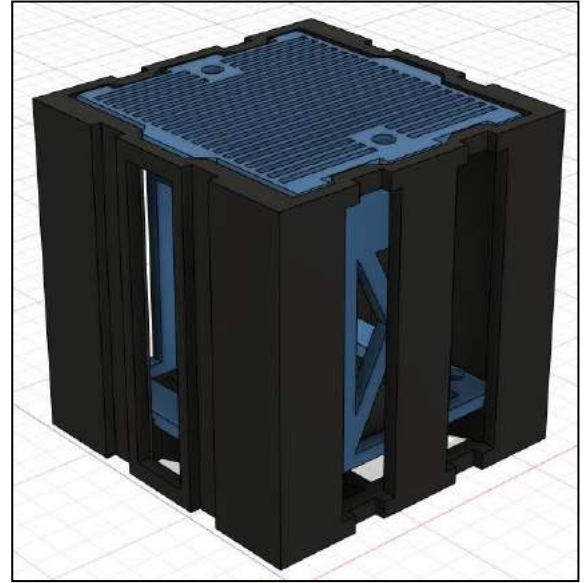
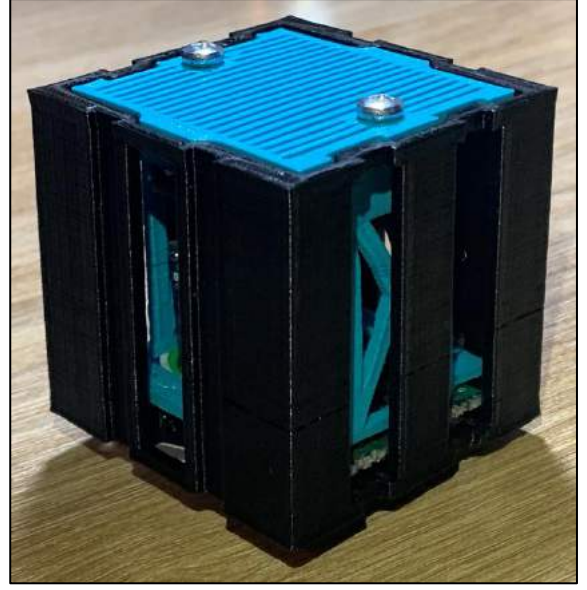
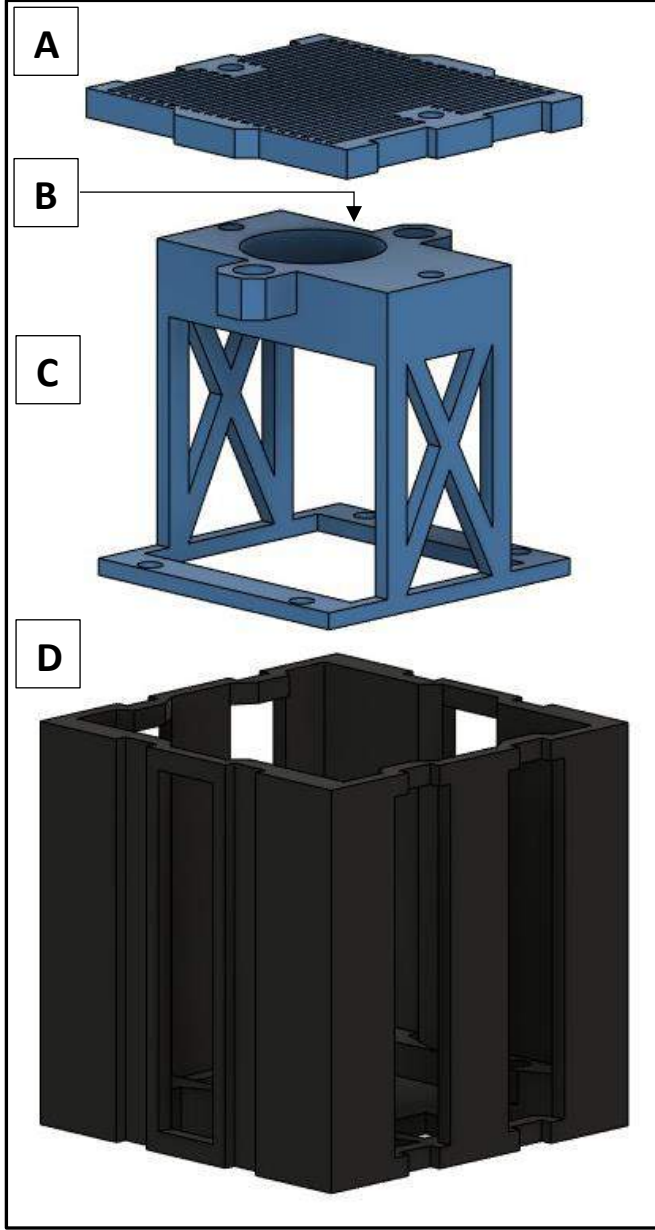

Ek 51 Tehlikeli Gaz Tespit Modülü Malzeme Listesi

TEHLİKELİ GAZ TESPİT MODÜLÜ MALZEME LİSTESİ			
		Malzeme	Adet
1		Arduino Pro Mini 328 - 3.3 V / 8 MHz (Header'lı)	1
2		Karbonmonoksit Gaz Sensör Kartı - MQ-7	1
3		20x30 Delikli Plaket	1
4		Filament 1.75 mm Turkuaz PLA - ABG	1
5		Filament 1.75 mm Siyah PLA - ABG	1
6		Dişi Pin Header 2.54mm	1
7		12mm Erkek Header	3
8		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
9		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
10		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	6
11		Devreli Buzzer 12mm - 3V 30mA - 85dB	1
12		5mm Led Kırmızı - 400-500 mcd	1
13		5mm Led Yeşil - 400-500 mcd	1

Ek 52 Tehlikeli Gaz Tespit Modülü Elektrik ve Veri Dağıtım Şeması

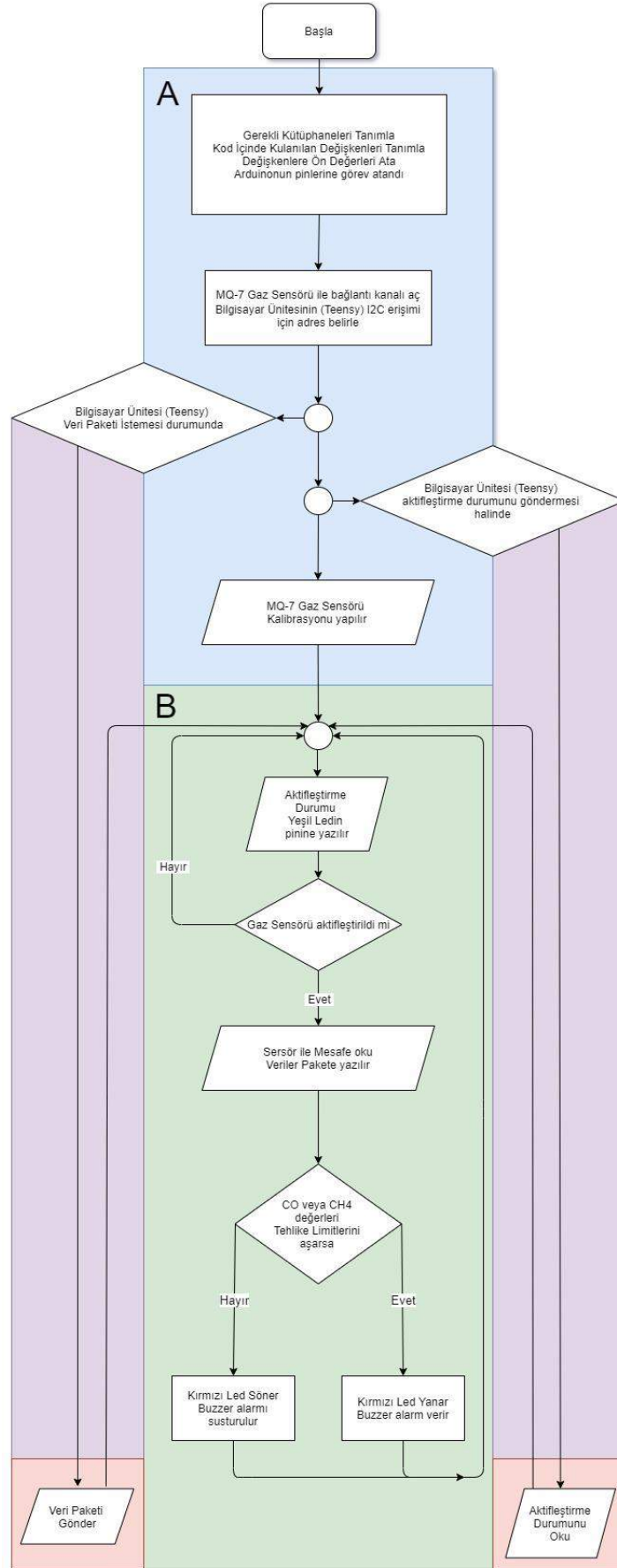


Ek 53 Tehlikeli Gaz Tespit Modülü Tasarımı



- A. Koruyucu Izgara
- B. Gaz Sensörü Yuvası
- C. Sensör ve Devre Taşıyıcı Blok
- D. Modül Kasası
- E. MQ-7 Gaz Sensörü

Ek 54 Tehlikeli Gaz Tespit Modülü Yazılımı ve Akış Şeması (1)



```
1  #include <MQUnifiedsensor.h>
2  #include <Wire.h>
3
4  byte Package[5];
5  int Address = 5;
6  bool turn = 0;
7  int req = 0;
8  double lastReq = 0;
9  double lastBuzz = 0;
10 int Buzz = 12;
11 int Red = 10;
12 int Green = 11;
13 int danger = 0;
14
15 union {
16     byte array[4];
17     float bigNum;
18 } H2;
19
20 union {
21     byte array[4];
22     float bigNum;
23 } LPG;
24
25 union {
26     byte array[4];
27     float bigNum;
28 } CH4;
29
30 union {
31     byte array[4];
32     float bigNum;
33 } CO;
34
35 union {
36     byte array[4];
37     float bigNum;
38 } Alcohol;
39
40 MQUnifiedsensor MQ7("Arduino Pro Mini", 5, 10, A0, "MQ-7");
41
42 void setup() {
43     Serial.begin(9600);
44     Wire.begin(Address);
45
46     Wire.onReceive(receiveEvent);
47     Wire.onRequest(requestEvent);
48
49     MQ7.setRegressionMethod(1);
50     MQ7.init();
51
52     for(int i = 0; i < 3; i++){
53         pinMode(i+10, OUTPUT);
54     }
55
56     Serial.print("Calibrating please wait.");
57     float calcR0 = 0;
58     for(int i = 1; i<=10; i ++)
59     {
60         MQ7.update(); // Update data, the arduino will be read the voltage on the analog pin
61         calcR0 += MQ7.calibrate(27.5);
62         Serial.print(".");
63     }
64     MQ7.setR0(calcR0/10);
65     Serial.println(" done!");
66
67     if(isinf(calcR0)) {Serial.println("Warning: Conection issue founded, R0 is infite
68         (Open circuit detected) please check your wiring and supply"); while(1);}
69     if(calcR0 == 0){Serial.println("Warning: Conection issue founded, R0 is zero
70         (Analog pin with short circuit to ground) please check your wiring and supply"); while(1);}
71 }
72
```



```

71 void loop() {
72   danger = 0;
73   MQ7.update();
74   digitalWrite(Green, turn);
75
76   if(millis() - lastReq > 100){
77     req = 0;
78   }
79
80   if(turn){
81
82     MQ7.setA(6000000000000); MQ7.setB(-10.54);
83     CH4.bigNum = MQ7.readSensor(); //CH4
84     if(CH4.bigNum > 9999){danger ++;}
85
86     MQ7.setA(99.042); MQ7.setB(-1.518);
87     CO.bigNum = MQ7.readSensor(); //CO
88     if(CO.bigNum > 50){danger ++;}
89
90     MQ7.setA(69.014); MQ7.setB(-1.374);
91     H2.bigNum = MQ7.readSensor(); //H2
92
93     MQ7.setA(700000000); MQ7.setB(-7.703);
94     LPG.bigNum = MQ7.readSensor(); //LPG
95
96     MQ7.setA(4000000000000000); MQ7.setB(-12.35);
97     Alcohol.bigNum = MQ7.readSensor(); //Alcohol
98
99   }
100   if(danger > 0){
101     if(millis() - lastBuzz < 250){
102       digitalWrite(Buzz, 1);
103     }
104     else{
105       digitalWrite(Buzz, 0);
106     }
107     if(millis() - lastBuzz > 500){
108       lastBuzz = millis();
109     }
110     digitalWrite(Red, 1);
111   }
112   else{
113     digitalWrite(Buzz, 0);
114     digitalWrite(Red, 0);
115   }
116
117 }

```







Fonksiyonlar

```

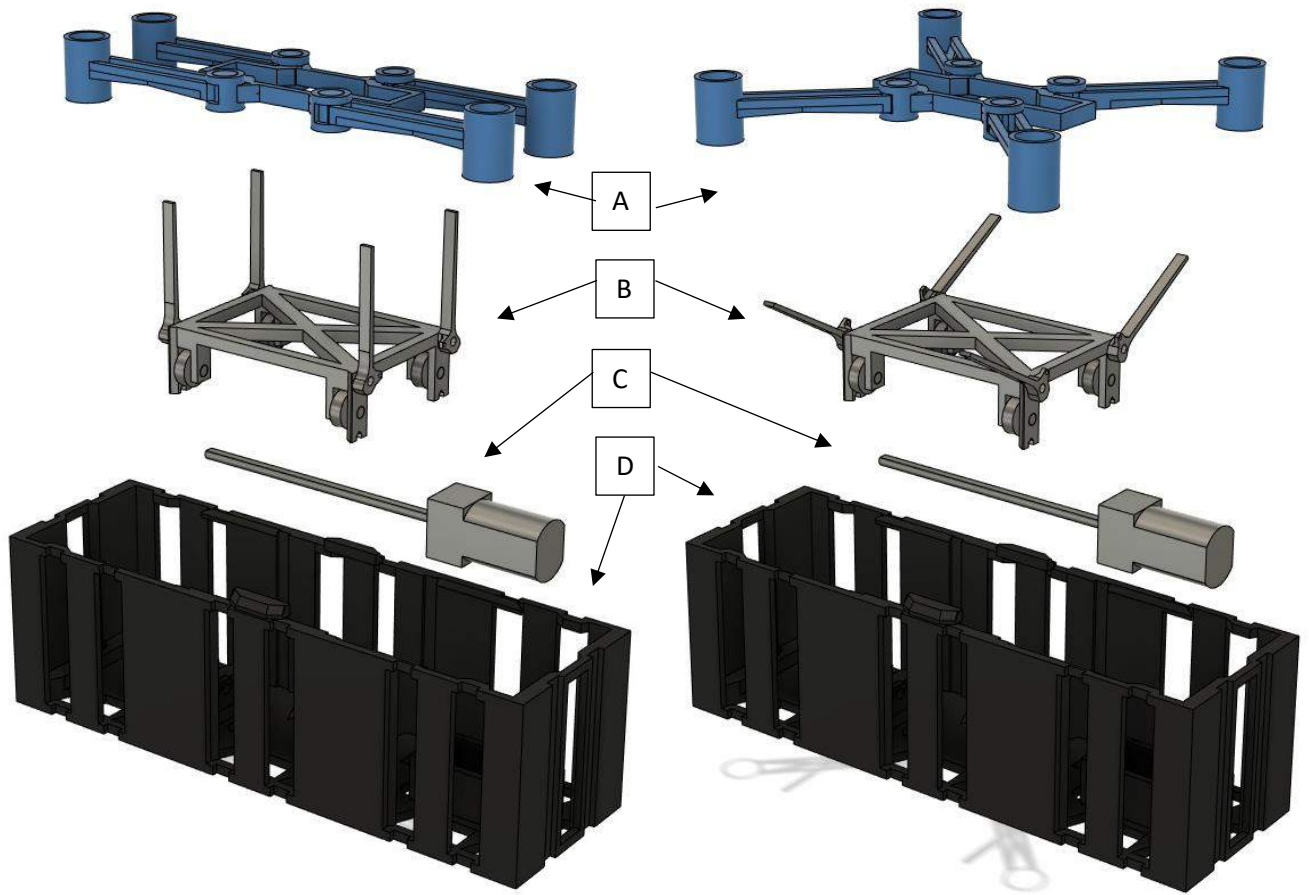
122 void receiveEvent() {
123   while(Wire.available()) {
124     turn = Wire.read();
125   }
126   Serial.println("Rec");
127 }
128
129 void requestEvent() {
130   lastReq = millis();
131   if(req == 2){
132     for(int i = 0; i < 4; i++){
133       Wire.write(CH4.array[i]);
134     }
135     req = 3;
136   }
137   else if(req == 3){
138     for(int i = 0; i < 4; i++){
139       Wire.write(CO.array[i]);
140     }
141     req = 4;
142   }
143 }

```

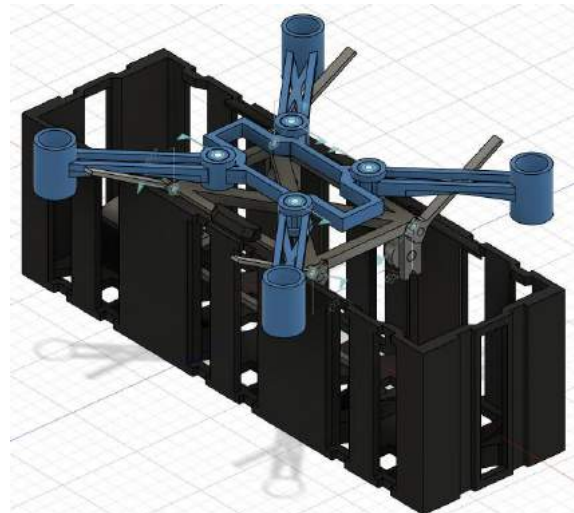
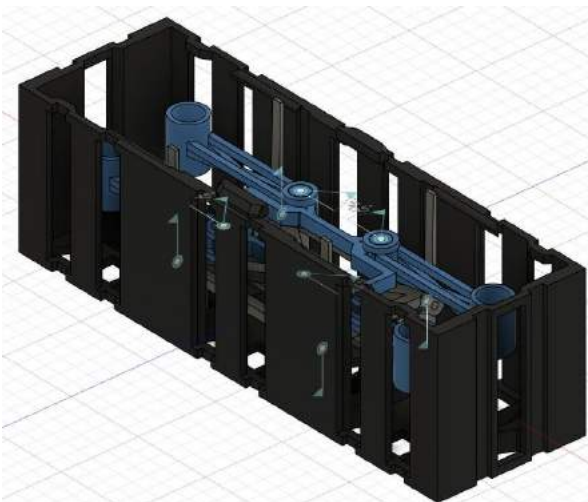
Ek 55 Dron İstasyonu Modülü Malzeme Listesi

DRON MODÜLÜ MALZEME LİSTESİ			
		Malzeme	Adet
1		6V 30Rpm 15mm Redüktörlü Dc Motor	1
2		Arduino Pro Mini 328 - 3.3 V / 8 MHz	1
3		Filament 1.75 mm Turkuaz PLA - ABG	1
4		Filament 1.75 mm Siyah PLA - ABG	1
5		12mm Erkek Header	2
6		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
7		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
8		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	2
9		20x30 Delikli Plaket	1
10		1k Ohm Direnç	8
11		10k Ohm Direnç	4
12		BC237 Transistör	8
13		Mikro Anahtar	1

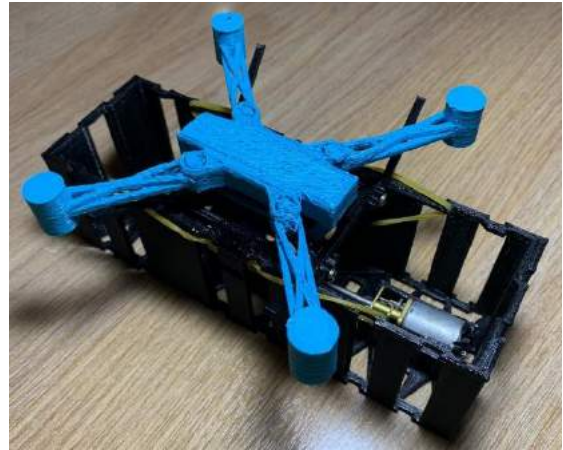
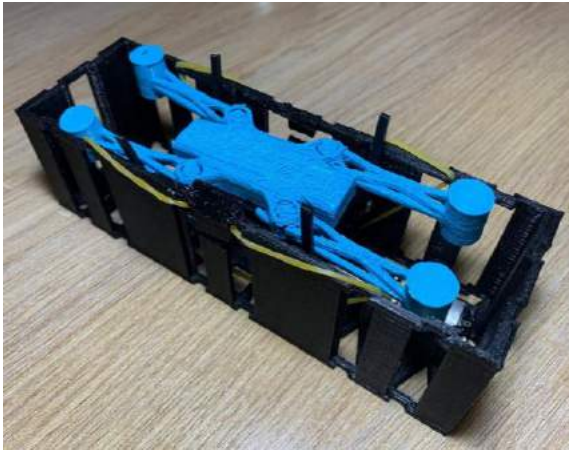
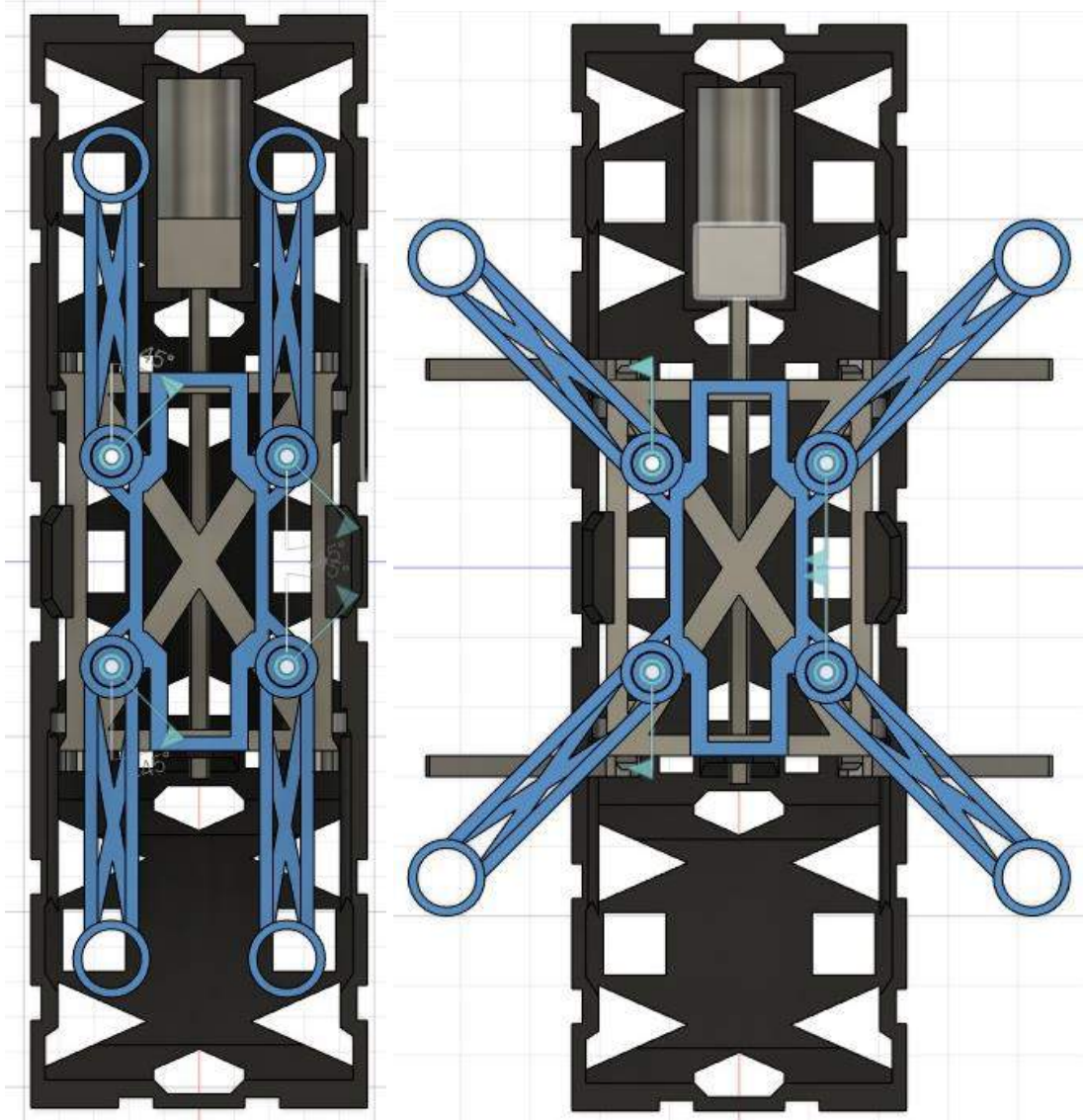
Ek 56 Dron İstasyonu Modülü Tasarım (1)



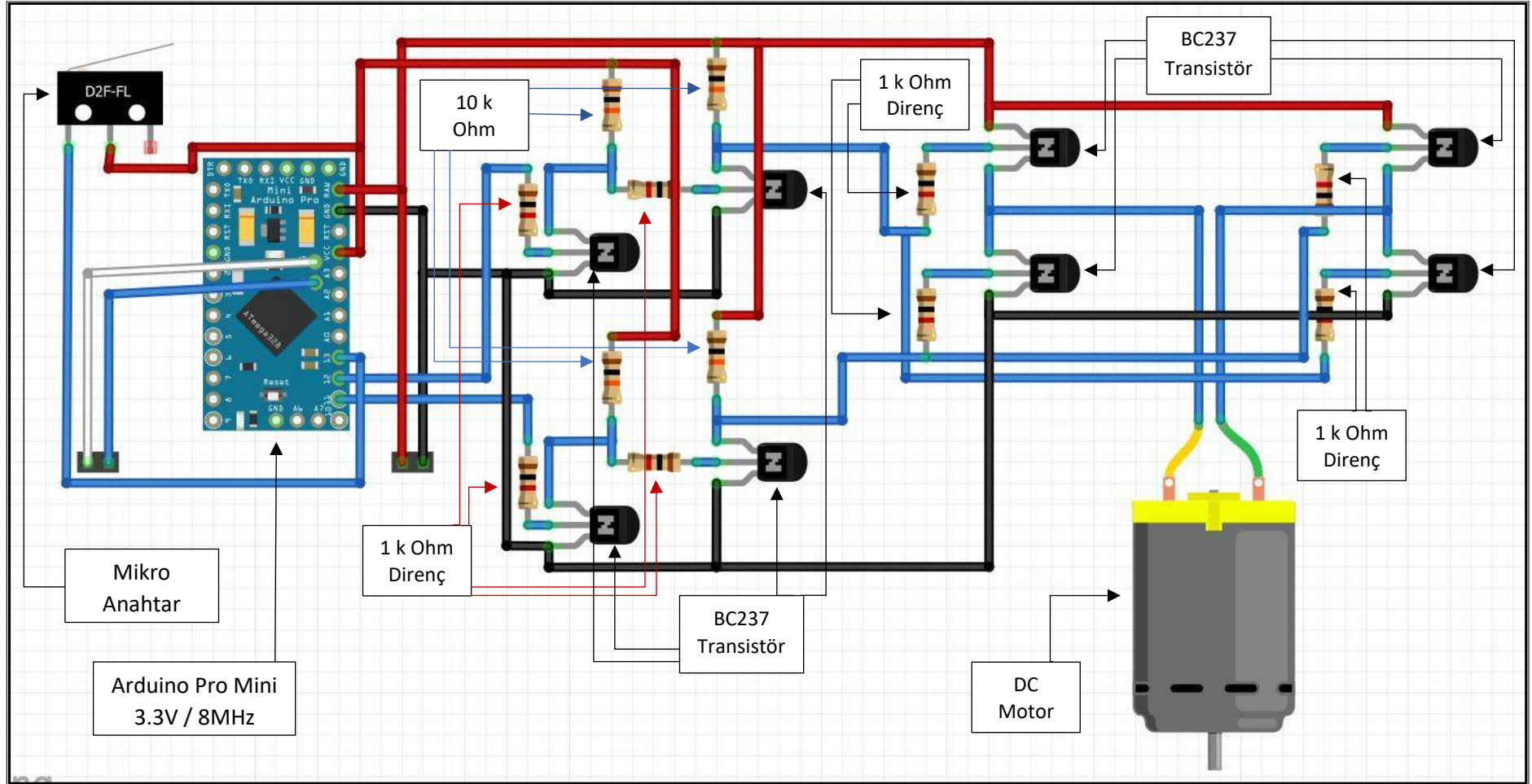
- A. Örnek Katlanabilir Dron Şasesi (Bu Proje kapsamında örnek bir Dron Şasesi hazırlanmıştır)
- B. Dron Platformu. (Üzerindeki Dronu istasyon içine çekmek üzere katlanır ve İstasyon içine iner)
- C. Dc Motor (Dron Platformunu ve Dronu İstasyon içine çeker)
- D. Koruyucu Kasa (156x48x48 mm ölçülerinde modül kasası kullanılmıştır.)



Ek 56 Dron İstasyonu Modülü Tasarım (2)

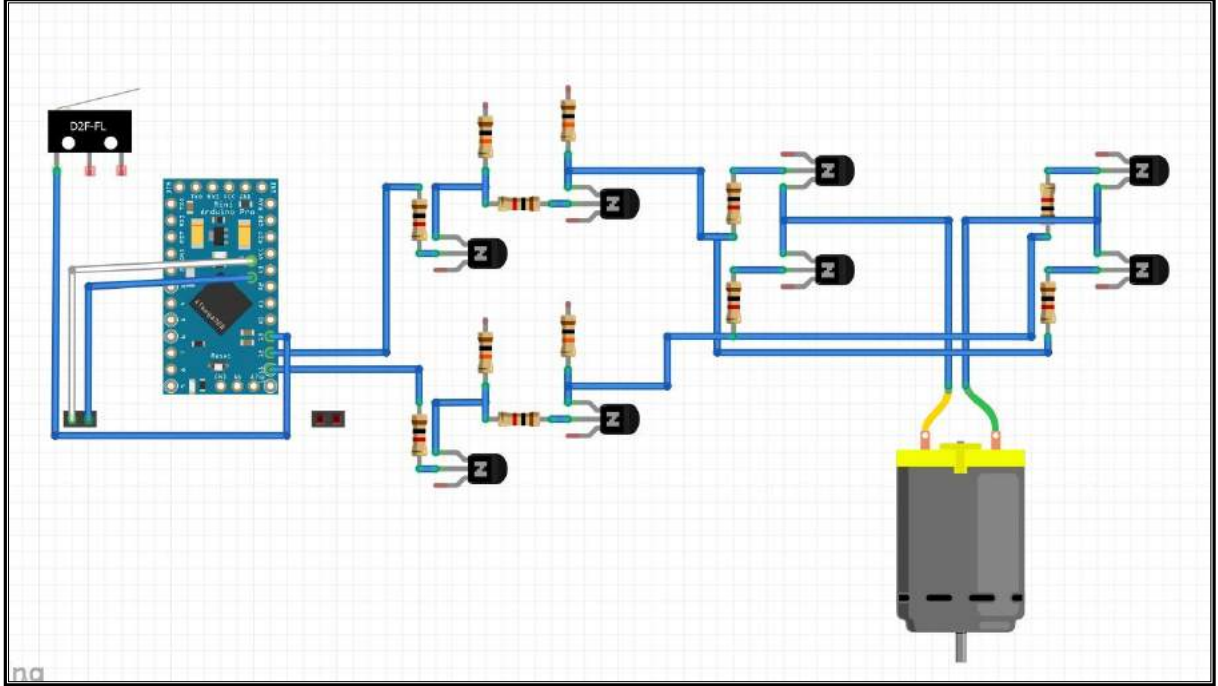


Ek 57 Dron İstasyonu Modülü Elektrik ve Veri Dağıtım Şeması (1)

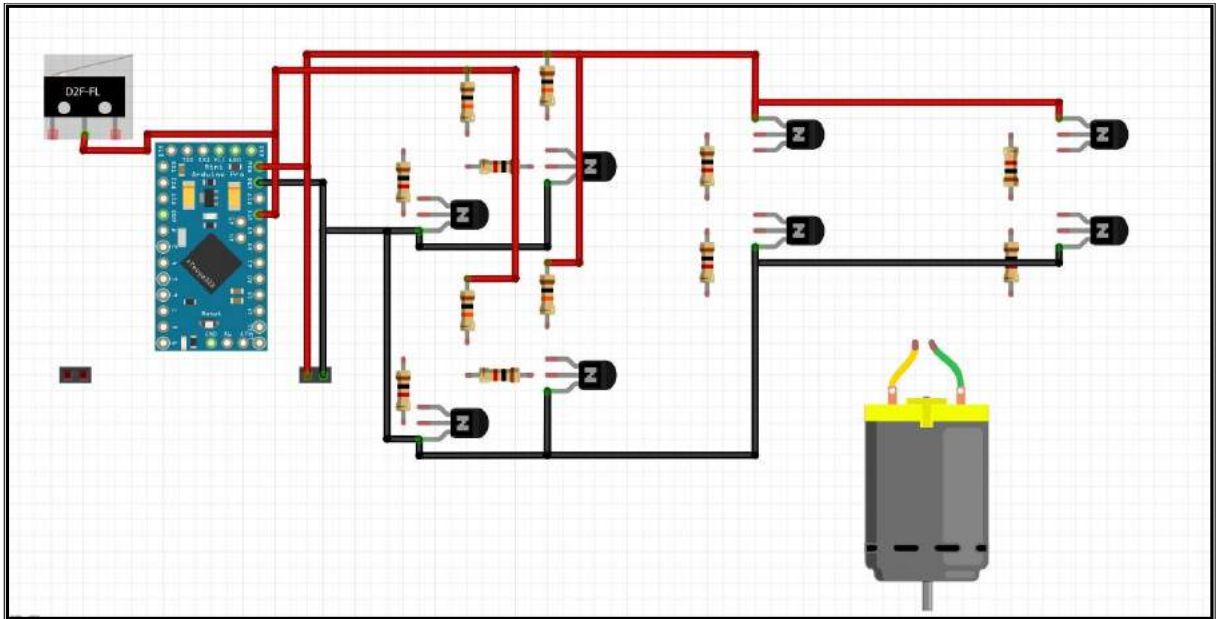


Ek 57 Dron İstasyonu Modülü Elektrik ve Veri Dağıtım Şeması (2)

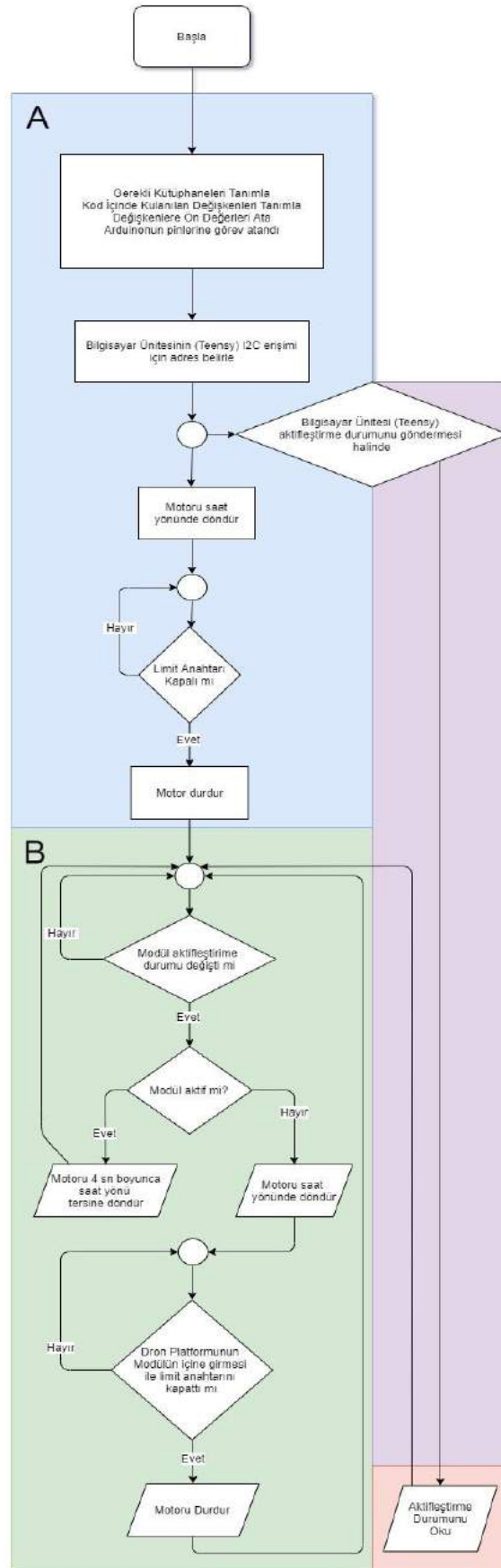
Dron İstasyonu Modülü Veri Dağıtım Şeması



Dron İstasyonu Modülü Elektrik Dağıtım Şeması



Ek 58 Dron İstasyonu Modülü Yazılımı ve Akış Şeması (1)



Ek 58 Dron İstasyonu Modülü Yazılımı ve Akış Şeması (2)

A

```
1  #include <Wire.h>
2
3  int Adress = 3;
4  bool turn = 0;
5  bool oldTurn = 0;
6
7  void setup() {
8      Serial.begin(9600);
9      Wire.begin(Adress);
10
11     Wire.onReceive(receiveEvent);
12
13     for(int i = 0; i < 2; i++){
14         pinMode(i+11, OUTPUT);
15     }
16     pinMode(13, INPUT);
17 }
18
```

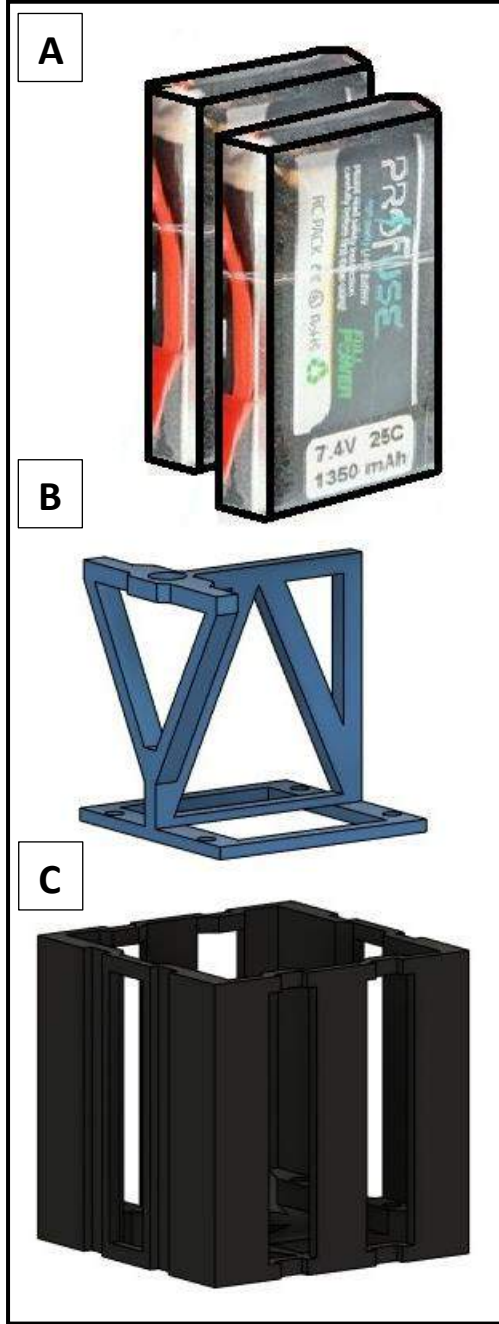
B

```
19 void loop() {
20     if(oldTurn != turn){
21         if(turn){
22             digitalWrite(11,1);
23             delay(4000);
24         }
25         else{
26             digitalWrite(12,1);
27             while(!digitalRead(13)){
28                 digitalWrite(12,0);
29             }
30         }
31     }
32 }
33
```

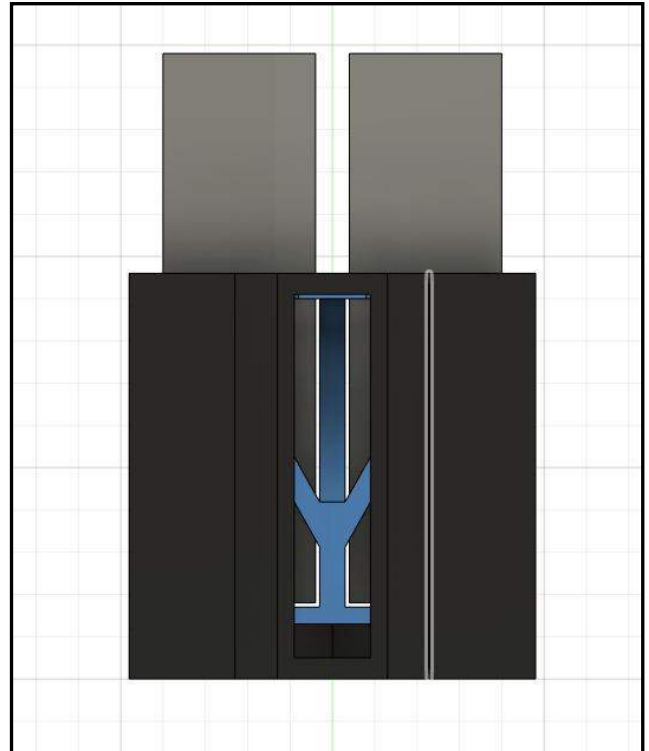
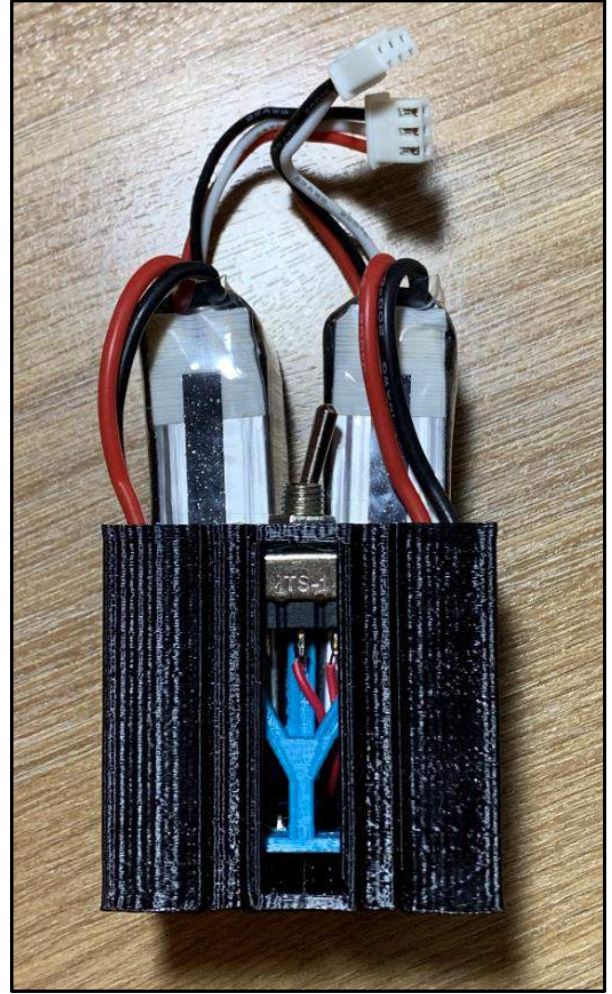
Fonksiyonlar

```
34 void receiveEvent() {
35     while(Wire.available()) {
36         turn = Wire.read();
37     }
38     Serial.println("Rec");
39 }
```

Ek 59 Yedek Batarya Ünitesi Tasarımı



- A. 2 X 7.4 V 2S Lipo
Batarya 1350 mAh
- B. Batarya Tutucu Blok
- C. Modül Kasası



Ek 60 Yedek Batarya Ünitesi Malzeme Listesi

YEDEK BATARYA ÜNİTESİ MALZEME LİSTESİ			
		Malzeme	Adet
1		7.4 V 2S Lipo Batarya 1350 mAh 25C	2
2		KTS102 On/Off 3 Ayak Toggle Anahtar	1
3		20x30 Delikli Plaket	1
4		Filament 1.75 mm Siyah PLA - ABG	1
5		Filament 1.75 mm Turkuaz PLA - ABG	1
6		12mm Erkek Header	4
7		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Mavi	1
8		Tek Damarlı Montaj Kablosu 22 AWG - 15 Metre Kırmızı	1
9		M3 8 mm YSB Yıldız Silindirik Baş Metrik Vida	2

Ek 61 Performans Değerleri (1)

ROBOT	Bulgu	Açıklama	Kaynak
Ağırlık - Tam yüklü (gr)	2600 gr		Ölçüm
Ağırlık - Modülsüz (gr)	2200 gr		Ölçüm
Batarya ömrü - Kesintisiz çalışma (dakika)	15.75 dakika		Hesaplama
Çalışma Sıcaklığı - Maksimum (°C)	+55°C	Servo Motorlar	Teknik Bilgi Formu
Çalışma Sıcaklığı - Minimum (°C)	0 °C	Servo Motorlar	Teknik Bilgi Formu
Azami Hız (metre/saat)	720 m/s		Ölçüm
Taşıyabileceği Yük (gr)	6000 gr		Ölçüm
Yükseklik - Normal Duruş (cm)	20 cm		Ölçüm
Yükseklik - Maksimum (cm)	22 cm		Ölçüm
Yükseklik - Minimum (cm)	12 cm		Ölçüm
Uzunluk (cm)	51 cm		Ölçüm
Genişlik (cm)	22.5 cm		Ölçüm
Adım Yüksekliği - normal yürüyüş (cm)	5 cm		Ölçüm
Adım Uzunluğu - normal yürüyüş (cm)	20 cm		Ölçüm
Kablosuz kontrol mesafesi (m)	1000 m		Teknik Bilgi Formu
Kamera görüş açısı	120°		Teknik Bilgi Formu
Kamera Sistemi	NTSC		Teknik Bilgi Formu
Kamera Görüntüsü Maksimum Aktarım Mesafesi (m)	1500 m		Teknik Bilgi Formu
GPS Lokasyon Hassasiyeti (m)	2.5 m		Teknik Bilgi Formu
Basınç Sensörü Duyarlılığı - Minimum (gr)	20 gr		Teknik Bilgi Formu
Basınç Sensörü Duyarlılığı - Maksimum (gr)	10000 gr		Teknik Bilgi Formu
Jiroskop X, Y, Z Eksenleri Maksimum Açılabilir Ölçüm Aralığı (°/s)	±2000 °/s		Teknik Bilgi Formu
LİDAR MODÜLÜ			
Yükseklik (cm)	7 cm		Ölçüm
Uzunluk (cm)	4.8 cm		Ölçüm
Genişlik (cm)	4.8 cm		Ölçüm
Ağırlık (gr)	62 gr		Ölçüm
Kızılötesi Sensörü Minimum Mesafe Ölçümü (cm)	3 cm		Teknik Bilgi Formu
Kızılötesi Sensörü Maksimum Mesafe Ölçümü (cm)	200 cm		Teknik Bilgi Formu
Rotasyon Hızı (rpm)	60 Rpm		Teknik Bilgi Formu
TEHLİKELİ GAZ TESPİT MODÜLÜ			
Yükseklik (cm)	4.8 cm		Ölçüm
Uzunluk (cm)	4.8 cm		Ölçüm
Genişlik (cm)	4.8 cm		Ölçüm
Ağırlık (gr)	48 gr		Ölçüm
Karbonmonoksit (CO) Gazı Sensörü Minimum Ölçüm Değeri (ppm)	20 ppm		Teknik Bilgi Formu
Karbonmonoksit (CO) Gazı Sensörü Maksimum Ölçüm Değeri (ppm)	2000 ppm		Teknik Bilgi Formu
Metan (CH4) Gazı Sensörü Minimum Ölçüm Değeri (ppm)	50 ppm		Teknik Bilgi Formu
Metan (CH4) Gazı Sensörü Maksimum Ölçüm Değeri (ppm)	4000 ppm		Teknik Bilgi Formu
YEDEK BATARYA ÜNİTESİ			
Yükseklik (cm)	8 cm		Ölçüm
Uzunluk (cm)	4.8 cm		Ölçüm
Genişlik (cm)	4.8 cm		Ölçüm
Ağırlık (gr)	170 gr		Ölçüm
Batarya ömrü - Kesintisiz çalışma (dakika)	10.62 Dakika		Hesaplama

Ek 61 Performans Değerleri (2)

DRON İSTASYONU MODÜLÜ			
Yükseklik (cm)	4.8 cm		Ölçüm
Uzunluk (cm)	15.6 cm		Ölçüm
Genişlik (cm)	4.8 cm		Ölçüm
Ağırlık (gr)	120 gr		Ölçüm
KONTROL KUMANDASI			
Yükseklik (cm)	7 cm		Ölçüm
Uzunluk (cm)	20 cm		Ölçüm
Genişlik (cm)	16 cm		Ölçüm
Ağırlık (gr)	480 gr		Ölçüm
Batarya ömrü - Kesintisiz çalışma (saat)	8 saat		Ölçüm
Kablosuz kontrol mesafesi (m)	1000 m		Teknik Bilgi Formu
Çalışma Sıcaklığı - Maksimum (°C)	+58 °C	Filament (+60°C Batarya)	Teknik Bilgi Formu
Çalışma Sıcaklığı - Minimum (°C)	-10 °C	Kontrol Çubuğu (Joystick)	Teknik Bilgi Formu
Ekran Sistemi	NTSC/PAL		Teknik Bilgi Formu
Ekran Çözünürlüğü	480 x 272		Teknik Bilgi Formu
Ekran Batarya Ömrü (dakika)	120 dk		Teknik Bilgi Formu

Ek 62 Kullanılan Ekipmanın Performans ve Dayanıklılık Değerleri (1)

1.1.Şase

Servo Motorlar - MG958: (Ek 63-1)

Taşıma kapasitesi (Durma Torku) 6.0v: 20.0kg.cm

Çalışma Sıcaklığı Aralığı: 0 ile +55 C° arası

Çalışma Hızı (6.0v): 0.15sn / 60 derece

Şase Materyali: Filament PLA (Ek 63-2)

PLA (Polilaktik Asit) maddesi, bio plastiktir, mısır nişastası bazlıdır.

Erime Sıcaklığı: 220 °C

Kırılma Sıcaklığı (Yumuşama) Sıcaklığı: +58 °C

Kopma esnemesi : %4

Gerilme Direnci (Megapascal): 57.86 MPa (590 x 0.0980665), 590 kg/cm²

1.2. Bilgisayar Modülü ve Sensörleri

Mikrokontrolcü Teensy 3.5 (Ek 63-3)

32-bit 120MHz ARM Cortex-M4 işlemci (MK64FX512VMD12)

Mikroişlemci Çalışma Sıcaklığı : -40 ile 105°C arası

512Kb flaş, 192Kb RAM, 4Kb EEPROM bellek

Basınç Sensörü RP-L-170 (Ek 63-4)

Basınç Ölçüm Aralığı: 20g ~ 10kg veya daha büyük

Çalışma Sıcaklığı: -40 °C ~ + 85 °C

Tepki Süresi: <10ms

GPS Modülü Neo-7m (Ek 63-5)

GPS Lokasyon Hassasiyeti 2.5 m

İlk veri yakalama 29 s

Veri alma tekrarlama 1 s

Çalışma Sıcaklığı -40° C ile 85° C arası

Seramik Kablolulu 28dB Yüksek Kazançlı Aktif GPS Anteni, 30 MHz bant genişliği (Ek 63-6)

Wifi Modülü NRF24L01+PA+LNA SMA 2.4G Kablosuz (Ek 63-7)

Frekans: 2.4GHz~2.5GHz

250 Kbps, 1 Mbps ve 2 Mbps kablosuz data aktarım hızı

IPX anteni ile 1000 metre mesafeye kadar haberleşme

6 Data kanalı

Çalışma Sıcaklığı: -40 °C ile + 85 °C arası

Ek 62 Kullanılan Ekipmanın Performans ve Dayanıklılık Değerleri (2)

Jiroskop ve Eksen İvme Sensörü, MPU6050 Gyro ve 6 Eksen İvme Sensörü
Jiroskop X, Y, Z eksenlerinin açısal sensör ölçüm aralığı (programlanabilir)
 $\pm 250, \pm 500, \pm 1000, \pm 2000$ °/saniye (dps)
İvme Ölçer açısal ölçüm aralığı (programlanabilir) $\pm 2g \pm 4g \pm 8g \pm 16g$
Sıcaklık sensörü -40°C ile $+85^{\circ}\text{C}$ arası
(Ek 63-8)

1.3. Batarya ve Regülatör ünitesi

2 adet 7.4V 2S1P 2000 mAh (30C) Li-Polymer Pil kullanılmıştır.

Her pil 7.4V ve 2000 mAh kapasitededir.

(Ek 63-9)

2 adet LM2596 DC-DC Çevirici (4-35V/ 1-30V)

Robotun 3.3 Volt ve 5 Volta ihtiyaç duyan birim ve modüllerini beslemek üzere Regülatör ünitesine eklenmiştir (Ek 14 - 17).

(Ek 63-10)

4 adet XL4005 DC/DC Çevirici (5-32V / 1.25-32V)

Robotun 6 Volta ihtiyaç duyan servo motorları beslemek üzere Regülatör ünitesine eklenmiştir (Ek 14 - 17). Her biri 3 adet servo motoru besler.

(Ek 63-11)

1.4.Kontrol Kumandası

Mikrokontrolcü Arduino Micro 5V (Ek 63-12)

8 bit Atmega32u4 Mikrokontrolcü temelli bir mikro denetleyici karttır.

32Kb flaş, 2.5Kb SRAM, 1Kb EEPROM bellek, 16 mhz saat hızı

Mikroişlemci Çalışma Sıcaklığı : -40 ile 85°C arası

2 Adet Kontrol çubuğu, Pro 3 Eksen Joystick kullanılmıştır. X ve Y ekseninde 50° lik eğim, bükme topuzu ile Z eksenini hareketi sağlanmaktadır.

Çalışma Sıcaklığı:: -10°C ile $+80^{\circ}\text{C}$

(Ek 63-13)

1 adet 16 karakter x 2 satır kapasiteli LCD ekran bulunur.

(Ek 63-14)

Wifi Modülü NRF24L01+PA+LNA SMA 2.4G Kablosuz

(Ek 63-15)

Frekans: 2.4GHz~2.5GHz

250 Kbps, 1 Mbps ve 2 Mbps kablosuz data aktarım hızı

IPX anteni ile 1000 metre mesafeye kadar haberleşme

6 Data kanalı

Çalışma Sıcaklığı: -40°C ile $+85^{\circ}\text{C}$ arası

Ek 62 Kullanılan Ekipmanın Performans ve Dayanıklılık Değerleri (3)

Batarya 3.7V 590 mAh Li-Polymer Pil, 590 mAh değerine sahip Lityum Polimer pil kullanılmaktadır. (Ek 63-16)

Çalışma Sıcaklığı: -20 °C ile +60 °C

Bataryanın elektriğini 3.7V'dan 5V yükseltmek amacıyla 1 adet 5 V Yükselteçli Voltaj Regülatörü U1V10F5 - PL-2564 kullanılmıştır. (Ek 63-17)

1.5. Kamera ve FPV Monitör

1 adet kamera kullanılmıştır.

Eachine TX01 NTSC Super Mini AIO 5.8G 40CH 25MW VTX 600TVL 1/4 Cmos (Ek 63-18)

1 adet robot köpeğe edilmiştir.

Sistem ve çözünürlüğü NTSC - 600TVL

120 derece görüş açısı

Frekans: 5.8GHz 5 bant 40 kanal

Ağırlık: 4.48 g

1 adet FPV LCD monitör, kameradan gelen görüntüleri izlemek amacıyla Kontrol Ünitesine dahil edilmiştir. (Ek 63-19)

5.8G 48CH 4.3 Inch LCD 480x272 16:9 NTSC/PAL FPV Monitor

Auto Search With OSD Build-in Battery

Frekans aralığı : 5645MHz~5945MHz

6 Bant ve 48 Kanal

Çözünürlük : 480x272, Çerçeve oranı: 16:9, Video Formatı: NTSC/PAL

Operating Voltage: 5V, Elektrik tüketimi: 500mA, Dahili batarya: 600mAh

Çalışma süresi: 2 saat

Ekran boyutları 4.3 inch LCD

1.6. Tehlikeli Gaz Tespit Modülü

Mikrokontrolcü Arduino Pro Mini 3.3V 8Mhz

(Ek 63-20)

8 bit Atmega328 Mikrokontrolcü temelli bir mikro denetleyici karttır.

32Kb flaş, 2Kb SRAM, 1Kb EEPROM bellek, 16Mhz

Mikroişlemci Çalışma Sıcaklığı : -40 ile 105°C arası

Karbonmonoksit gazı sensörü MQ7 (Ek 63-21)

CO (Karbonmonoksit) algılama yoğunluğu: 20 – 2.000 ppm (parts per million)

CH4 (Metan) algılama yoğunluğu: 200 ppm - 10.000 ppm (parts per million)

Ek 62 Kullanılan Ekipmanın Performans ve Dayanıklılık Değerleri (4)

1.7. Lidar Modülü

Mikrokontrolcü Arduino Pro Mini 3.3V 8Mhz

(Ek 63-20)

8 bit Atmega328 Mikrokontrolcü temelli bir mikro denetleyici karttır.

32Kb flaş, 2Kb SRAM, 1Kb EEPROM bellek, 16Mhz

Mikroişlemci Çalışma Sıcaklığı : -40 ile 105°C arası

VL53L0X Time-of-Flight Regülatörlü Mesafe Sensörü

VL53L0X kızılötesi sensörü mesafe ölçümü ~3 cm ile 200 cm arasındır.

Çalışma Voltajı 2.6 ile 5.5V arası, Çalışma akımı 10mA, maksimum 40mA

(Ek 63-22)

Çalışma Sıcaklığı -20°C ile +70°C arası

12V 12mm 60 RPM Redüktörlü Mikro DC Motor, kızılötesi sensörün dönüşü için kullanılmıştır. (Ek 63-23)

Çalışma Voltajı: 12V

Devir: 60 Rpm/Dk

Boşta Çektiği Akım: 80mA ± 40mA Zorlanma Akımı: 0.45A

Tork: 15 kg/cm Ağırlık: 9.5gr

5mm Bakır Bant, Malzeme: % 99.98 CU (Bakır) iletken yapışkanlı (akrilik yapışkanın ısıyla indüksiyonu) Sıcaklık dayanımı -10 °C ile 120 °C arası

(Ek 63-24)

1.8. Drone Modülü

6V 30Rpm 15mm Redüktörlü Dc Motor, platformun ve dronun istasyondan çıkarılması ve hazır hale getirilmesi için kullanılır.

Çalışma Voltajı: 6V

Hızı: 30 Rpm

Boşta Çektiği Akım: 300 mA, Zorlanma Akımı: 2.3A

(Ek 63-25)

1.9. Yedek Batarya Ünitesi

2 adet 7.4 V 2S Lipo Batarya 1350 mAh 25C Li-Polymer PİL kullanılmıştır.

Her pil 7.4V ve 1350 mAh kapasitededir.

(Ek 63-26)

Ek 63 Kullanılan Ekipmanın Teknik Özellikleri

1. MG958 TDS

<http://www.towerpro.com.tw/product/mg958/>

[Home](#) / [Servos & Parts](#) / [X-Large Servo 50g+](#) / MG958



MG958

MG958

MG958 is best choice if you need higher torque than 15KG.

Excellent performance for RC-cars from 10 to 6-th scale truggy and monster and 30cc 60cc airplane.

Specification:

Weight: 65g

Dimensions: 40.2 x 20.1 x 36.8mm

Operating Voltage: 4.8v-6.6v

Stall Torque (4.8v): 18.0kg.cm

Stall Torque (6.6v): 20.0kg.cm

Servo case: Alloy case in the middle

Gear Type: Metal gear

Bearing: Double ball bearing

Temperature range: 0- 55deg

Operating Temperature Range: 0 ~ +55 Degree C

Operating Speed (4.8v): 0.18sec/60 degree

Operating Speed (6.0v): 0.15sec/60 degree

servo wire length: 32cm

Current draw at idle 10MA

No load operating current draw 170MA

Stall current draw 1600MA

Servo Plug: JR (Fits JR and Futaba)

servo arms & screws included and fit with Futaba servo arm

CE & RoHS approved

We have upgraded our servo gear set and shaft to aluminum 6061-T6.

It is stronger and lighter than copper.

Categories: [Servos & Parts](#), [X-Large Servo 50g+](#)

2. ABG PLA Filament

ABG Enterprise Girişim Teknoloji Danışmanlık Sanayi ve Ticaret LTD ŞTİ



ABG FILAMENT PLA TDS

PHYSICAL, MECHANICAL AND THERMAL PROPERTIES

Properties	Test Condition	Test Method	Unit	Typical Value
Specific Gravity		ASTM D792		1,24
Molding Shrinkage		ASTM D955	%	0,2-0,4
Melt Flow Rate	220 °C/10 Kg	ASTM D1238	g/10 min.	27
Tensile Strength	@3,2mm Yield 50mm/dak.	ASTM D638	kg/cm ²	590
Tensile Elongation	@3,2mm Break 50mm/min.	ASTM D638	%	4
Tensile Modulus	@3,2 mm 1mm/min.	ASTM D638	kg/cm ²	33,000
Flexural Strength	@3,2mm 15mm/min.	ASTM D790	kg/cm ²	550
Flexural Modulus	@3,2mm 15mm/min.	ASTM D790	kg/cm ²	23,000
Izod Impact Strength 6,4mm Notched	23°C -30°C	ASTM D256	kg-cm/cm kg-cm/cm	16 10
Izod Impact Strength 3,2mm Notched	23°C -30°C	ASTM D256	kg-cm/cm kg-cm/cm	16 10
Rockwell Hardness	R-Scale	ASTM D785		75
Vicat Softening Temperature	5kg, 50°C/h	ASTM D1525	°C	58

<http://www.abgfilament.com/assets/images/PLAEN.pdf>

Temin Edildiği adres:

www.direnc.net/filament-175-mm-turkuazz-pla-abg

www.direnc.net/filament-175-mm-siyah-pla-abg

3. Teensy 3.5 (1)

<https://www.robotistan.com/teensy-6328>



Teensy 3.5, küçük boylu ve breadboard uyumlu bir geliştirme kartıdır. 32-bit ARM Cortex-M4 platformuna sahiptir. Arduino IDE'nin modifiye edilmiş bir versiyonu (Teensyduino) ile kolay bir şekilde programlama imkanı sunar. Teensy 3.5, 3.2 sürümünün güncellenmiş ve daha fazla işlem gücü sunan sürümüdür.

3.5 sürümü, kayan nokta ünitesi (floating point unit) içeren 32-bit 120MHz ARM Cortex-M4 işlemciye sahiptir. Tüm dijital ve analog girişler 5V uyumludur.

Özellikler:

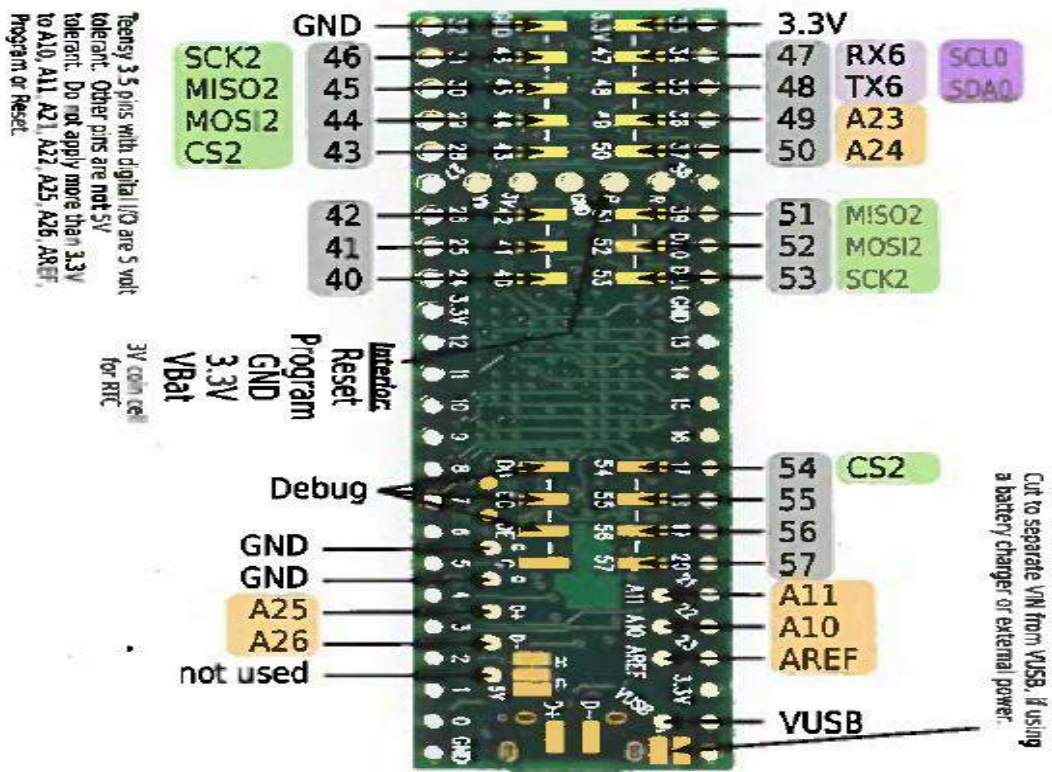
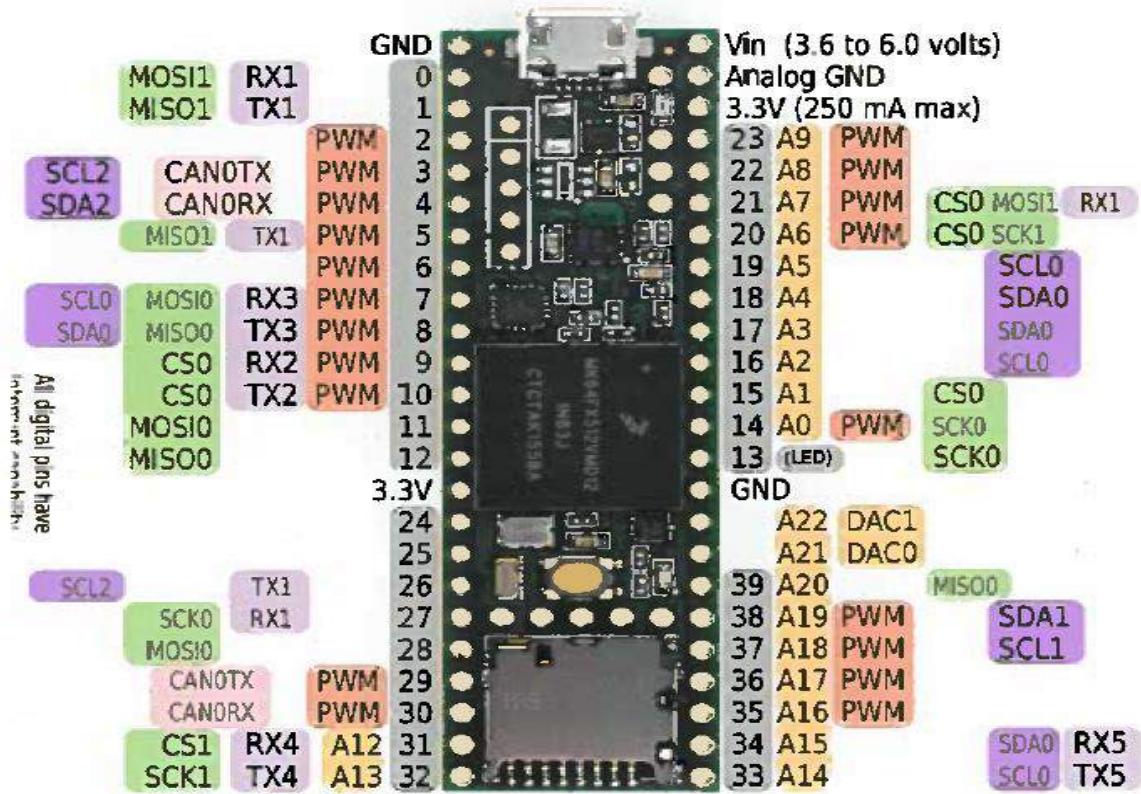
- 120MHz ARM Cortex-M4 işlemci, kayan nokta hesabı destekli
- 512K flaş, 192K RAM, 4K EEPROM bellek
- MK64FX512VMD12 mikrokontrolcü ([datasheet](#))
- 1 adet CANBUS portu
- 16 adet genel kullanım için DMA kanalı
- 5V toleranslı giriş/çıkış pinleri

Arduino IDE benzeri bir program aracılığı ile programlanabilse de, Teensy resmi bir Arduino ürünü değildir. Arduino ile daha önce yapmış olduğunuz uygulamaların bir çoğu Teensy ile uyumlu şekilde çalışsa da, çoğu kütüphanede sorun yaşamanız olasıdır.

Teknik Özellikler:

- 62 adet giriş/çıkış pini (42 adedi breadboard ile kullanılabilir)
- 25 adet analog giriş, 2 adet 13-bit çözünürlüklü ADC
- 2 adet analog çıkış, 12-bit çözünürlüklü DAC
- 20 adet PWM çıkış
- Full speed (12 Mbit/sn) destekli USB portu
- 100 Mbit/sn destekli Ethernet bağlantı desteği
- Native (4-bit SDIO) mikro SD kart desteği
- I2S ses çıkışı, 4 kanal dijital ses giriş/çıkış desteği
- 14 adet donanımsal timer
- Kriptografik hızlandırıcı ünitesi
- Rastgele sayı üretici
- CRC hesaplama ünitesi
- 6 adet seri port (2 adet FIFO ve hızlı baud rate desteği sunar)
- 3 adet SPI portu (1 adedi FIFO desteği sunar)
- 3 adet I2C portu
- Gerçek zaman saati (real time clock)

3. Teensy 3.5 (2)



4. RP-L-170 Basınç Sensörü (1)



RP-L-170 Thin Film Pressure Sensor

SKU:SEN0293

INTRODUCTION

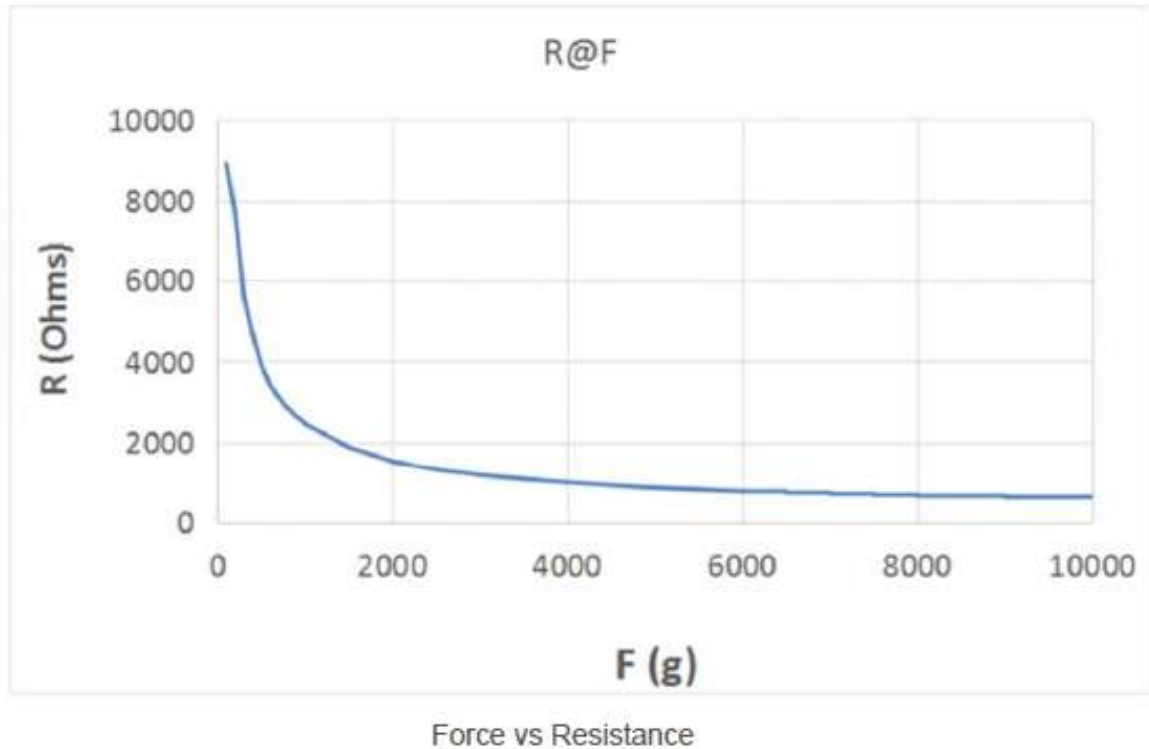
This is a long flexible thin film pressure sensor with a length of 170mm, which can be used to realize highly sensitive detection of pressure. The sensor is durable and designed to sense static and dynamic pressure in a high respond speed. Its advantages of recording the intensity and frequency of force make it widely used in all kinds of applications, such as, pressure switch, bed monitoring system, intelligent sneaker and medical device system. These sensors are also very easy to use.

RP-L flexible pressure sensor is made of ultra-thin film of excellent mechanical property, excellent conductive materials and nanometre pressure sensitive layers. There are thin film and pressure sensitive layer on the upper layer of the sensor, and thin film and conductive circuit on the lower layer. These two layers are glued together by double sided tape. When outside pressure applies to the active area, the disconnected circuit of the lower layer will be connected through the pressure sensitive layer of the upper layer, by which to convert pressure into resistance. The output resistance decreases as pressure increases.



RP-L-170 Dimension Diagram

4. RP-L-170 Basınç Sensörü (2)

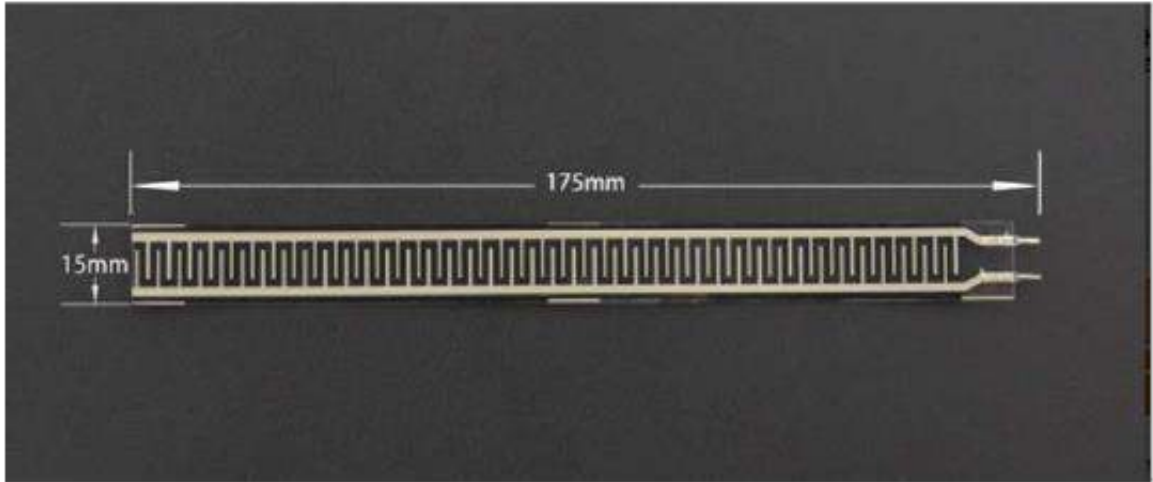


SPECIFICATION

- Thickness: 0.35mm
- Trigger Force: <20g, triggered (default resistance<200k Ω)
- Pressure Measuring Range: 20g~10kg or larger
- Static Pressure & Dynamic Pressure Measurement (within the frequency of 10Hz)
- Initial Resistance: >10M Ω
- Activation Time: <0.01S
- Operating Temperature: -40°C~+85°C
- Lifespan: >1million times
- Hysteresis: +10%, (RF+ -RF-)/FR+, 1000g Force
- Response Time: <10ms
- EMI: Not generate
- EDS: Not generate
- Drift: <5%, 2.5Kg Force , Static load 24H

Ek 63 Kullanılan Ekipmanın Teknik Özellikleri

4. RP-L-170 Basınç Sensörü (3)



<https://www.dfrobot.com/product-1843.html?search=SEN0293/5-9-19>

5. Gps Modülü Neo-7m Arduino Shield Mini



Neo-7m Mini Gps Modülü Arduino ile uyumlu çalışmaktadır. Bilindiği üzere GPS (Global Positioning System) bir Küresel Konumlama Sistemi'dir. Dünya üzerinde herhangi bir görüş hattında, 4 veya daha fazla uydusu ile her türlü hava koşullarında yer ve zaman bilgilerini sağlayan uzay tabanlı uydu navigasyon sistemidir.

Arduino Shield Mini Gps Modülü Teknik Özellikleri

Üretici Firma	China
Ürün Kategorisi	Arduino Shield
Çalışma Gerilimi	3.3V/5 V
Ölçü Değerleri	17*23*33cm
Çıkış Frekansı	1Hz
Çıkış Baud Hızı	9600
Çıkış Protokolü	nmea-0183

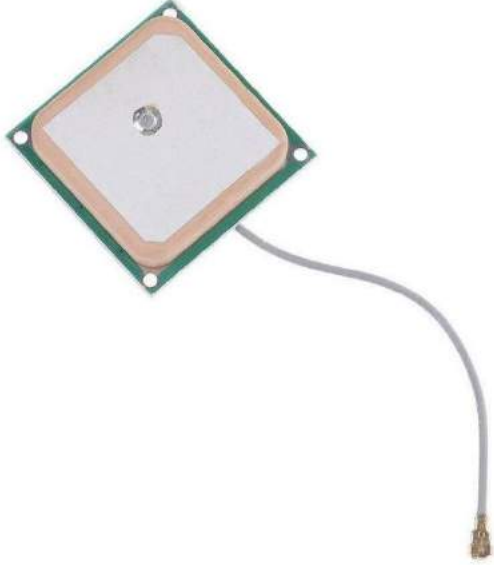
https://www.u-blox.com/sites/default/files/products/documents/MAX7-NEO7_HardwareIntegrationManual_%28UBX-13003704%29.pdf

https://www.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf

https://www.u-blox.com/sites/default/files/products/documents/NEO-7_ProductSummary_%28UBX-13003342%29.pdf

<https://www.direnc.net/neo-7m-arduino-gps-shield>

6. GPS Anteni Seramik Kablolı 28dB Aktif



Seramik Kablolı 28dB Yüksek Kazançlı Aktif GPS Anteni

İki kademeli düşük gürültülü amplifikatör ve filtre tasarımı. Bu GPS aktif anten, 28dB yüksek kazanç, GPS aktif anten SMA fişi, merkezi frekans 1575.42MHz'dir. GPS uydu sinyalleri alınabilir. DC akım 10mA, düşük güç tüketimi.

Teknik Özellikler

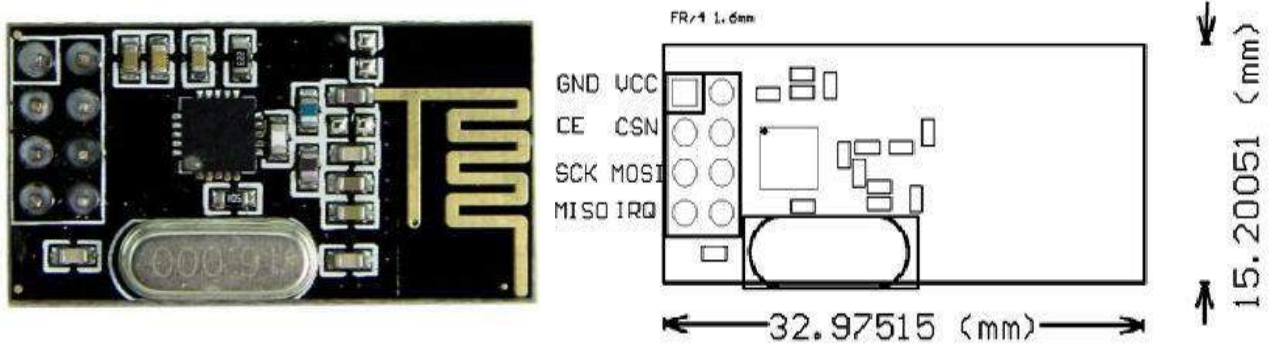
- Merkezi frekans: 1575MHz
- VSWR: 2: 1
- Bant genişliği: ± 30 MHz
- Empedans: 50 Ohm
- maksimum kazanç:> 3dBic 7x7cm'ye göre
- Anten polarizasyonu: RHCP
- Amplifikatör / filtre
- Amplifikatör kazancı: 32dB
- Gürültü faktörü: 1.5dB
- Güç tüketimi: 10mA
- DC gerilimi: 3V - 5V
- Güç tüketimi: 10mA
- Konektör türü: IPEX
- Boyut: 28 * 28 * 8mm
- Kablo uzunluğu: 5 cm
-

<http://rfcoax.com.tw/image/ANTENNA/GPS/GPS-KR-01.pdf>

<https://www.motorobit.com/urun/seramik-kablolu-28db-yuksek-kazancli-aktif-gps-anteni>

7. NRF24L01 2.4 GHz Kablosuz Alıcı Verici Modül

Nordic firmasının geliştirilen NRF24L01 kablosuz modül, 2.4GHz frekansında kablosuz haberleşme yapmanıza imkan sağlayan düşük güç tüketimine sahip modüldür. Çeşitli hobi, robotik ve endüstriyel projelerde sıklıkla kullanılacak 2MBps haberleşme hızına sahip olup, SPI arabirimini destekler.



Özellikleri:

- 2.4GHz bandında yayın yapabilir.
- 250KBps, 1MBps ve 2MBps gibi hızlarda haberleşme hızı seçilebilir.
- Gelişmiş ShockBurst™ hızlandırma protokolünü desteklemektedir.
- Ultra düşük güç tüketimi
- Çalışma Voltajı: 1.9-3.6V
- IO Portları Çalışma Voltajı:0-3.3V/5V
- Verici Sinyal Gücü: +7 dB
- Alıcı Hassasiyeti ≤ 90 dB
- Haberleşme Mesafesi: Açık Alanda 250m
- Boyutları: 15x29mm

Dokümanlar:

- [Kart şematiği için tıklayınız.](#)
- [NRF24L01 datasheeti için tıklayınız.](#)
- [Arduino Playground-NRF24L01 sayfasına gitmek için tıklayınız.](#)
- [Örnek Arduino programı için tıklayınız.](#)

https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Plus Preliminary Product Specification v1_0.pdf

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Wireless/Nordic/nRF24L01-Chip-v12.pdf>

<https://www.robotistan.com/wireless-nrf24l01-24ghz-transceiver-modul-24ghz-alici-verici-modul-1>

8. MPU6050 6 Eksen İvme ve Gyro Sensörü gy-521



MPU-6050 çeşitli hobi, multicopter ve robotik projelerinde sıklıkla kullanılan üzerinde 3 eksenli bir gyro ve 3 eksenli bir açısal ivme ölçer bulunduran 6 eksenli bir IMU sensör kartıdır.

Kart üzerinde voltaj regülatörü bulunduğundan 3 ile 5 V arası bir besleme voltajı ile çalıştırılabilir. İvme ölçer ve gyro çıkışlarının her ikisi de ayrı kanallardan I²C çıkışı vermektedir. Her eksenle 16 bitlik bir çözünürlükle çıkış verebilmektedir.

Pinler arası boşluk standart olarak ayarlandığı için breadboard veya farklı devre kartlarında rahatlıkla kullanılabilir.

Özellikleri:

- Çalışma gerilimi: 3-5V
- Gyro ölçüm aralığı: + 250 500 1000 2000 ° / s
- Açısal ivme ölçer ölçüm aralığı: ± 2 ± 4 ± 8 ± 16 g
- İletişim: Standart I²C

Faydalı Bilgiler:

[MPU-6050 Sensör Datasheeti İçin Tıklayınız](#)

<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Components/General%20IC/PS-MPU-6000A.pdf>

<https://www.f1depo.com/MPU6050-6-Eksen-Ivme-ve-Gyro-Sensoru-gy-521,PR-293.html>

9. Li-Polymer Pil 7.4V 2S1P 2000 mAh (30C) - Soketli



7.4V 2S1P 2000 mAh (30C) Li-Polymer Pil - Soketli

7.4V 2S1P 2000 mAh (30C) Li-Polymer Pil, 2000mah değerine sahip bu model li-polimer modeldir.

7.4V 2S1P 2000 mAh (30C) Li-Polymer Pil - Teknik Özellikler

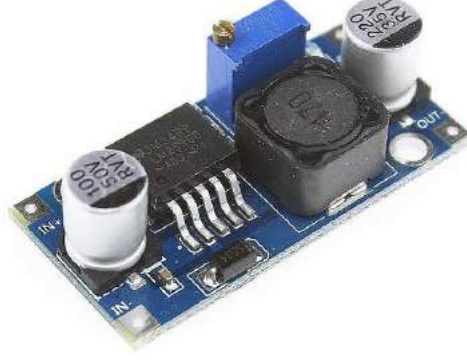
- Kimya: Li-Po, Lityum Polimer, Lithium Polymer
- Hücre Modeli: PX2000HP
- Voltaj: 7.4V
- Kapasite: 2000 mAh
- Ağırlık: 112 gr
- Boyut: 18 mm x 35 mm x 90 mm

<https://www.direnc.net/74v-2s1p-2000-mah-30c-li-polymer-pil-soketli>

10. LM2596 Ayarlanabilir Voltaj Düşürücü Güç Modülü

(4-35V Giriş - 1-30V Çıkış)

Arduino Uyumlu Voltaj Regülatörü - [China](#)



LM2596 Ayarlanabilir DC-DC Çevirici ile 4V DC - 35V DC arası giriş gerilimlerini 1.23V DC - 30 VDC aralığında regüle edebilirsiniz. Üzerinde bulunan trimpot aracılığı ile çıkış gerilimini istediğinize göre ayarlayabilirsiniz. Küçük boyutları ile oldukça kullanışlı olan bu modül üzerinde LM2596 regüle entegresi bulundurulmuştur. Devre 3A DC' ye kadar olan akımı taşıyabilir. 30 mV'luk ufak bir sapma payı ise diğer DC-DC konvertörlere göre büyük bir avantaj sağlar.

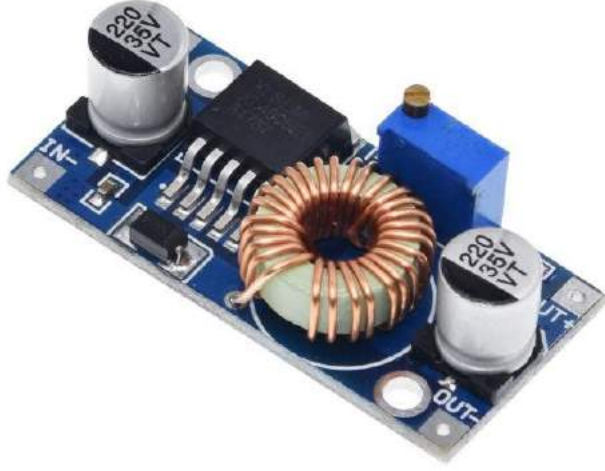
DİKKAT: Ürün voltaj düşürücü olduğundan çıkış gerilimi giriş geriliminden fazla olamaz. 30V giriş bağlarsanız 0-30V aralığında istediğiniz gerilime ayarlayabilirsiniz. Ancak girişe 12V bağlarsanız çıkışı 0-12V aralığında ayarlayabilirsiniz.

LM2596 Voltaj Düşürücü Ayarlanabilir Güç Modülü Teknik Özellikleri

- Modül özelliği: Yalıtımsız buck
- Düzeltme modu: Senkronize olmayan düzeltme
- Boyutu: 48 * 23 * 14mm (Uzunluk * Genişlik * Yükseklik)
- Tip: LM2596 Ayarlanabilir Güç Kaynağı Modülü
- Giriş Voltajı: DC 4V-35V
- Çıkış Voltajı: DC 1.23V-30V
- Çıkış Akımı: 3A (Maksimum)
- Dönüşüm Verimliliği: % 92 (En Yüksek)
- Çıkış Dalgalanma: 30mv (Maksimum)
- Anahtarlama Frekansı: 150KHz
- Yük regülasyonu: % 0,5
- Voltaj regülasyonu: % 2,5
- Çalışma sıcaklığı: -40 degress +85 degress

<https://www.direnc.net/lm2596-power-supply-modul>

11. XL4005 5-32V / 1.25-32V DC/DC Konvertör



XL4005 DC/DC Konvertör dilediğiniz değerdeki DC gerilimi elde etmenizi sağlar. Bu DC/DC dönüştürücü ile 1.25V-32V aralığında gerilim değeri elde edilebilir.

XL4005 5-32V DC/DC Konvertör Teknik Özellikleri

Üretici Firma	China
Ürün Kategorisi	Elektronik Kartlar ve Modüller
Giriş Gerilimi	5V-32V
Çıkış Gerilimi	1.25V-32V
Çıkış Akımı	0-5A
Çıkış Gücü	75W
Çalışma Sıcaklık Aralığı	-40°C ~ +85°C
Çalışma Frekansı	180KHz
Ölçü Bilgileri(L*W*H)	43*21*14mm

- **XL4005 5-32V DC/DC Konvertör Datasheet**
- <https://pdf.direnc.net/upload/xl4005-5-32v-125-32v-dc-dc-konvertor-datasheet.pdf>
- <https://www.direnc.net/xl4005-5-32v-125-32v-dc-dc-konvertor>

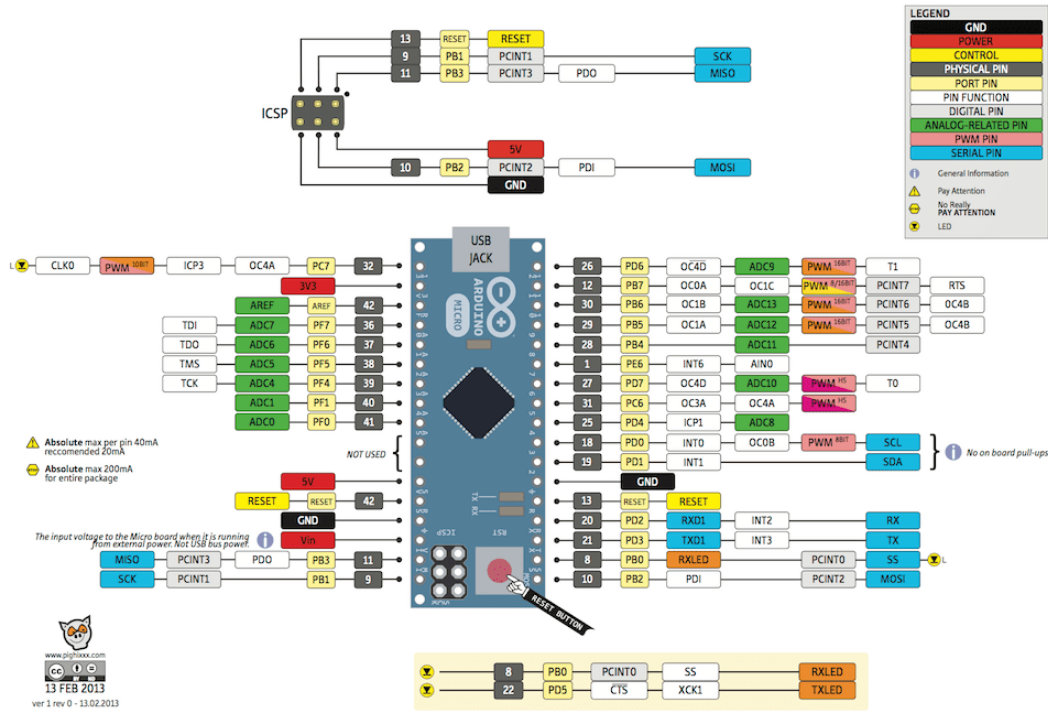
12. Arduino Micro (1)



Arduino Micro'nun son versiyonunun klonudur. Ürün içeriğinde ayrıca bilgisayar bağlantısı için Usb kablo bulunmaktadır.

Ürün İçeriği:

- Arduino Micro (Klon)
- Mikro Usb Kablo



Temin Edildiği adres :

<https://www.robotistan.com/arduino-micro-klon>

12. Arduino Micro (2)

Arduino Micro; Atmega32u4 temelli bir mikrodenetleyici kartıdır. Üzerinde 20 adet dijital giriş/çıkış pini (7 tanesi PWM çıkışı, 12 tanesi analog giriş olarak kullanılabilir), 16Mhz kristal, mikro usb soketi, ICSP konektörü ve reset tuşu bulundurmaktadır.

Micro'yu diğer bir çok arduino modelinden ayıran en büyük özellik; Atmega32u4 üzerinde dahili usb haberleşme özelliğinin olmasıdır. Bu şekilde 16u2 gibi ikinci bir usb-seri dönüştürücü işlemcisine gerek yoktur. Bu sayede sanal com portun (CDC) dışında micro bilgisayara mouse, klavye gibi bağlanarak kullanılabilir.

Teknik Özellikler:

- Mikrodenetleyici ATmega32u4
- Çalışma Gerilimi 5V
- Giriş Gerilimi (önerilen) 7-12V
- Giriş Gerilimi (limit) 6-20V
- Dijital I/O Pinleri 20 (7 tanesi PWM çıkışı, 12 tanesi analog giriş)
- Analog Giriş Pinleri 12
- Her I/O için Akım 40 mA
- 3.3V Çıkış için Akım 50 mA
- Flash Hafıza 32 KB (ATmega32u4) 4 KB kadarı bootloader tarafından kullanılmaktadır
- SRAM 2.5 KB (ATmega32u4)
- EEPROM 1 KB (ATmega32u4)
- Saat Hızı 16 MHz
- Uzunluk 48 mm
- Genişlik 18 mm
- Ağırlık 13 g

Güç:

Arduino Micro gücünü usb üzerinden veya harici güç kaynağından alabilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör ve batarya kart üzerindeki GND ve Vin pinleri üzerinden bağlanabilir.

Kartın çalışması için sürekli olarak usb'nin bağlı olması şart değildir. Kart sadece adaptör veya batarya ile çalıştırılabilir. Bu sayede kart bilgisayardan bağımsız olarak çalıştırılabilir.

Harici güç kaynağı olarak 6-20V arası kullanılabilir. Ancak bu değerler limit değerleridir. Kart için önerilen harici besleme 7-12V arasındadır. Çünkü kart üzerinde bulunan regülatör 7V altındaki değerlerde stabil çalışmayabilir. 12V üstündeki değerlerde de aşırı ısınabilir.

Micro kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur. Vin pini veya güç soketi üzerinden verilen 7-12V arası gerilim kart üzerinde bulunan voltaj regülatörü ile 5V'a düşürülerek karta dağılır.

Güç pinleri aşağıdaki gibidir:

- **VI:** Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.
- **5V:** Bu pin regülatörden çıkan 5V çıkışı verir. Eğer kart sadece usb (5V) üzerinden çalışıyor ise usb üzerinden gelen 5V doğrudan bu pin üzerinden çıkış olarak verilir. Eğer karta güç Vin (7-12V) veya güç soketi (7-12V) üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
- **3V3:** Kart üzerinde bulunan 3.3V regülatörü çıkış pini. Maks. 50mA çıkış verebilir.
- **GND:** Toprak pinleridir.

Hafıza:

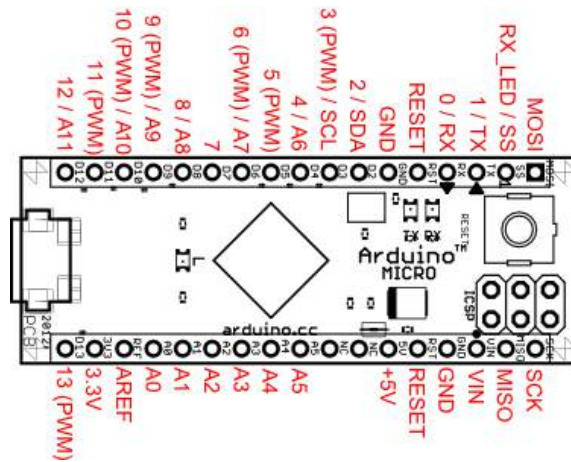
Atmega32u4 32 KB'lık flash belleğe sahiptir (4 KB kadarı bootloader tarafından kullanılmaktadır). 2.5 KB SRAM ve 1 KB EEPROM'u bulunmaktadır.

12. Arduino Micro (3)

Giriş ve Çıkış:

Micro üzerindeki 20 adet dijital pinin hepsi giriş veya çıkış olarak kullanılabilir. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maks. 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir.

- **Seri Haberleşme, 0 (RX) ve 1 (TX):** TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Dikkat edilmesi gereken nokta, Serial sınıfı micro üzerindeki USB (CDC) haberleşmesinde kullanılır. Pin 0 ve 1 pinleri üzerindeki seri haberleşme için Serial1 sınıfı kullanılmalıdır.
- **Harici Kesme, 3 (interrupt 0), 2 (interrupt 1), 0 (interrupt 2), 1 (interrupt 3), 7 (interrupt 4):** Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. Ayrıntılı bilgi için [attachInterrupt\(\)](#) fonksiyon sayfasını inceleyebilirsiniz.
- **PWM, 3,5,6,9,10,11 ve 13:** 8-bit çözünürlükte PWM çıkış pinleri olarak kullanılabilir.
- **SPI, ICSP Header'i üzerinde:** Bu pinler SPI haberleşmesi için kullanılır. Dikkat edilmesi gereken nokta ICSP header'ı üzerindeki SPI pinleri kart üzerindeki başka hiçbir pine bağlı değildir. Yani uno'daki gibi bu pinler 10,11,12 ve 13. pinlere bağlı değildir. Eğer SPI haberleşmesi yapan bir shield kullanacaksanız, bu shield üzerinde 3x2 pinlik ICSP header'ı olması gerekiyor. Yoksa bu shield micro ile kullanılamaz.
- **LED, 13:** Micro üzerinden 13. pine bağlı olan dahili bir led bulunmaktadır. Pin HIGH yapıldığında led yanacak, LOW yapıldığında led sönecektir.
- **Analog, A0-A5 ve A6-A11(aynı zamanda 4,6,8,9,10 ve 12. dijital pinler):** Micro 12 tane 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinlerden A0-A5 arasında olanlar uno'da olduğu gibi analog pin kısmındadır. A6-A11 arasındakiler ise dijital pin kısmındadır ve sırasıyla dijital 4,6,8,9,10 ve 12. pinlere bağlıdır. Bu pinler kartın altında belirtilmektedir. Tüm analog pinler dijital giriş ve çıkış içinde kullanılabilir. Pinlerin ölçüm aralığı 0-5V'dur. AREF pini ve analogReference() foksionu kullanılarak alt limit yükseltilip, üst limit düşürülebilir.
- **I2C, 2 veya SDA pini ve 3 veya SCL pini:** Bu pinler I2C haberleşmesi için kullanılır. Bu pinler uno üzerinde A4 ve A5 pinlerine bağlıdır. Micro da ise pin 2 ve 3'e bağlıdır.
- **AREF:** Analog giriş için referans pini.
- **Reset:** Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan Reset Butonu ile de yapılabilir.



Arduino micro ile Atmega32u4 arasındaki [pin haritalaması](#) sayfasını inceleyebilirsiniz.

12. Arduino Micro (4)

Haberleşme:

Arduino Micro'nun bilgisayarla, başka bir arduino veya mikrodenetleyici ile haberleşmesi için birkaç farklı seçenek vardır. Atmega32u4, 0 (RX) ve 1 (TX) pinleri üzerinden UART TTL (5V) seri haberleşme imkanı sunar. 32u4 usb üzerinden bilgisayara bağlanarak sanal bir com port açar ve serial (CDC) haberleşme yapmaya imkan sağlar. Arduino bilgisayar programı içerisinde barındırdığı seri monitör ile arduino ile bilgisayar arasında text temelli bilgilerin gönderilip alınmasını sağlar. Micro ile bilgisayar arasında usb üzerinden haberleşme olduğu zaman kart üzerinde bulunan RX ve TX ledleri yanacaktır.

Micro üzerinde donanımsal olarak bir adet seri port bulunmaktadır. Ancak **SoftwareSerial kütüphanesi** ile bu sayı yazılımsal olarak artırılabilir.

Atmega32u4 aynı şekilde I2C ve SPI portları da sağlamaktadır. Arduino bilgisayar programı ile gelen **Wire kütüphanesi** I2C kullanımını, **SPI kütüphanesi** de SPI haberleşmesini sağlamak için kullanılır.

Micro bilgisayara mouse, klavye gibi tanıtılabilir ve **Keyboard and Mouse** sınıfları ile kullanılabilir.

Programlama:

Arduino Micro kartı **Arduino bilgisayar programı** (Arduino IDE) ile programlanır. Programda Tools > Board sekmesi altında Arduino Micro'yu seçip programlamaya başlayabilirsiniz. Ayrıntılı bilgi için **referans** ve **temel fonksiyonlar** sayfasını inceleyebilirsiniz. Arduino Micro üzerindeki Atmega32u4 üzerine **bootloader** denilen özel bir yazılım yüklü gelir. Bu sayede kartı programlarken ekstra bir programlayıcı kullanmanıza gerek yoktur. Haberleşme orjinal AVR109 protokolü ile sağlanır.

Bootloader yazılımı bypass edilerek kart doğrudan mikrodenetleyicinin ICSP header'i üzerinden **ISP programlayıcı** ile programlanabilir (**Referans**).

USB Aşırı Akım Koruması:

Arduino Micro üzerinde bulunan resetlenebilir sigorta bilgisayarın usb portunu kısa devrelerden veya aşırı akım tüketimi durumlarından korumaktadır. Kart bilgisayar usb portu üzerinden 500mA'den fazla akım çektiğinde kart otomatik olarak usb'den aldığı gücü koruma amacıyla kesmektedir. Fazla akım durumu veya kısa devre ortadan kaldırıldığında sigorta normal konuma döner ve tekrar bağlantı kurulur.

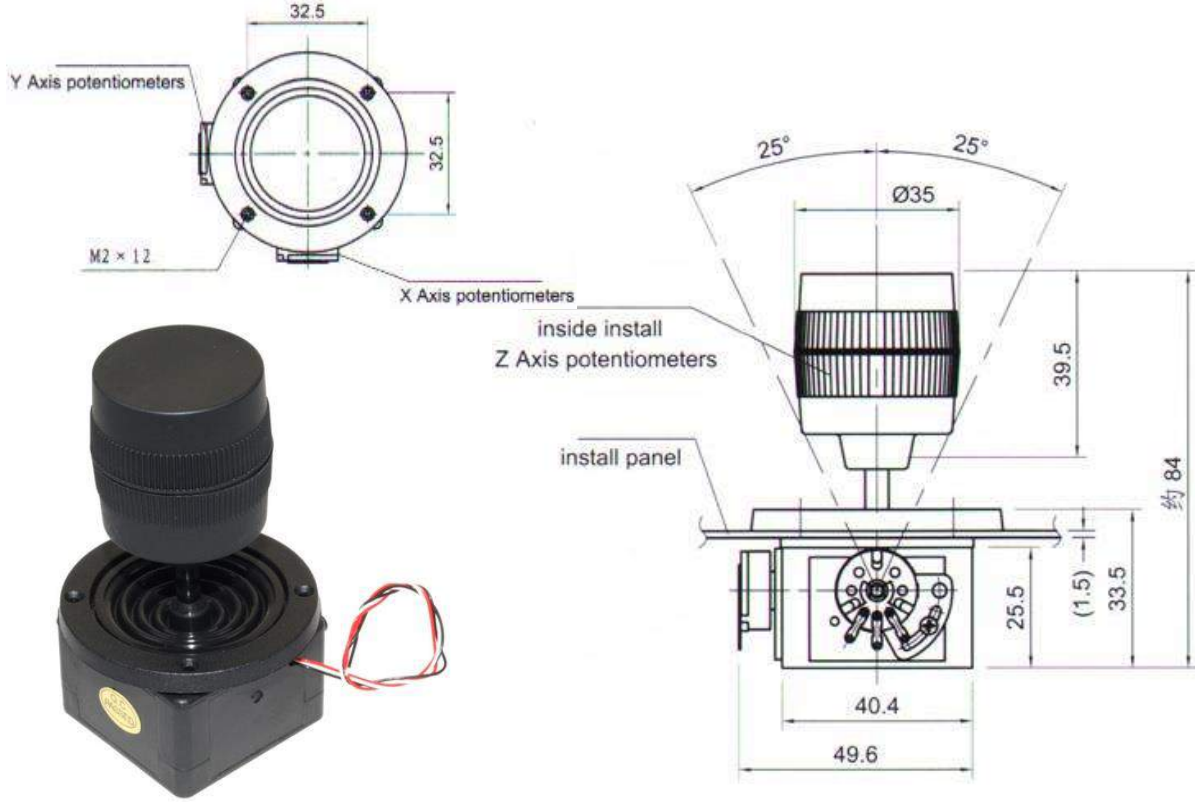
Dökümanlar:

- **Atmega32u4 Datasheet**
- **Sürücü Yükleme Dökümanı** (Windows)
- **Kart Şematiği**
- **Eagle PCB Çizim Dosyaları**
- **Arduino Yazılımı** (Arduino IDE)
- **Arduino Sayfası**

Mikroişlemci Çalışma Sıcaklığı : -40 ile 85°C arası

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf

13. Pro 3 Eksen Joystick (Butonsuz)



X ve Y ekseninde 50°'lik eğim imkanı sunan iki eksen ve twist topuzu ile üçüncü eksen hareketini sağlayabileceğiniz bu joystick çeşitli robotik ve endüstriyel projelerde ve kumanda sistemlerinde kullanılabilen bir üründür. Joystick üzerindeki potansiyometreler 5K'lık olup, serbest bırakıldığı zaman orta noktaya gelecek şekildedir.

<https://module-center.com/administrator/files/UploadFile/JH-D300X-R4.pdf>

<https://www.motorobit.com/urun/pro-3-eksen-joystick-butonsuz>

<https://www.superdroidrobots.com/shop/item.aspx/analog-3-axis-joystick/1263/>

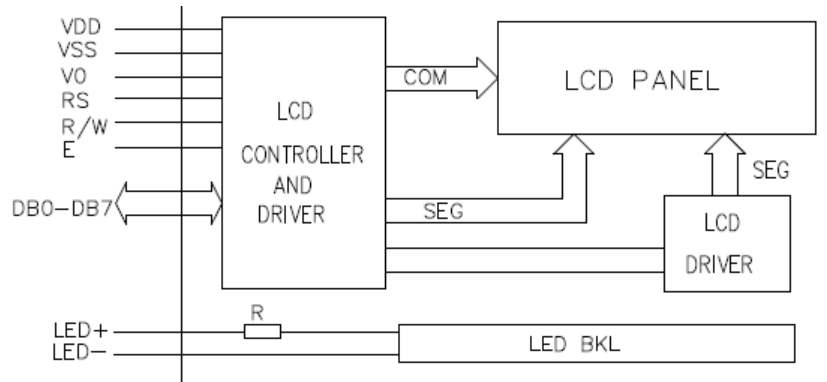
Features and Specifications:

- Resistance: 5k ohms
- Resistance Tolerance: 5%
- Independent Linearity Tolerance: 1%
- Resolution: Infinity
- X and Y Axis Rotational Angle: 50 degrees
- Operating Torque: 440gf
- Rotational life expectancy: 500,000 times
- Operating Temperature: -10 deg C to 80 deg C

14. LCD Ekran 2x16 Sol Üst Mavi - Qapass



- 16 karakter görüntüleyebilen 2 satır
- Karakter büyüklüğü 5×8 piksel
- 8-bit ve 4-bit modda çalışabilir
- Sonradan tanımlı karakterleri de gösterebilir
- ST7066 kontrolcü bulunur.



<https://components101.com/16x2-lcd-pinout-datasheet>

<https://www.sparkfun.com/datasheets/LCD/ADM1602K-NSW-FBS-3.3v.pdf>

<file:///E:/T%C3%BCbitak%202020/Proje%20D%C3%B6k%C3%BCman%C4%B1/Destek%20Dokumanlar/ST7066.PDF>

<https://www.direnc.net/2x16-lcd-display-sol-ust-mavi-qapass>

Ek 63 Kullanılan Ekipmanın Teknik Özellikleri

15. Kablosuz Wifi Modül

NRF24L01+PA+LNA SMA 2.4G

1 km Uzun Mesafe



NRF24L01+PA+LNA SMA 2.4G Kablosuz Modül

nRF24L01+ modülü ile dahili güç amfisi ve IPX anteni ile açık alanda 1000 metre mesafeye kadar haberleşme yapmanız mümkündür.

MCU/ARM/PIC/AVR/STM32 sistemlerinle birlikte kullanabilirsiniz.

Özellikler:

Frekans: 2.4GHz~2.5GHz

Çalışma voltajı : 3 ~ 3.6V Max

Akım : 115mA

Çoklu Frekans: 125 frequency

Data alımı için 6 kanala kadar destek.

<https://www.robotshop.com/media/files/pdf/datasheet-wir020.pdf>

<https://www.cakirelektronik.com/urun/nrf24l01-pa-lna-sma-2-4g-kablosuz-modul>

https://www.sparkfun.com/datasheets/Components/nRF24L01_prelim_prod_spec_1_2.pdf

<https://www.roboshop.com.tr/NRF24L01-PA-LNA-SMA-kablosuz-wifi-modul-uzun-mesafe>

16. 3.7V 590 mAh Li-Polymer Pil



3.7V 590 mAh Li-Polymer Pil

3.7V 590 mAh **Li-Polymer Pil**, 590 mAh değerine sahip **Lityum Polimer pil** çeşitlerindedir.

3.7V 590 mAh Li-Polymer Pil - Teknik Özellikleri

Üretici Firma	Power-Xtra
Ürün Kategorisi	Li-po Pil
Kimya	Li-polimer, lityum polimer, lithium polymer
Voltaj	3.7V
Kapasite	590 mAh
Boyut	5.2mm x 30.0mm x 40.0mm

https://www.birikimpilleri.net/Resim/900869503040_1.pdf

<https://www.direnc.net/37v-590-mah-li-polymer-pil>

17. Yükselteçli Voltaj Regülatörü 5V U1V10F5 - PL-2564

Bu küçük yükseltici regülatör kartı 0.5V'a kadar olan düşük voltajları 5V'a yükseltmek için kullanılır. Bir alkalin pil veya çeşitli düşük voltaj çıkışlı güç kaynaklarını 5V'a yükselterek çeşitli devre ve projelerde kullanılabilir. 1.2A'e kadar akım verebilir.

Marka / Menşei:	Pololu
Çalışma Voltajı :	0.5V-5V
	%4 doğrulukla 5V çıkışı
Boyutları :	9 x 11.5 x 2.5 mm
Ağırlığı :	0.6 g

<https://www.robotistan.com/5v-yukseltecli-voltaj-regulatoru-ncp1402-5v-boost-regulator-ncp1402-pl>

Pololu 5V Step-Up Voltage Regulator U1V10F5

<https://www.pololu.com/product/2564/specs>

This tiny (0.35"×0.45") U1V10F5 switching step-up (or boost) voltage regulator efficiently generates **5 V** from input voltages as low as 0.5 V. Unlike most boost regulators, the U1V10F5 automatically switches to a linear down-regulation mode when the input voltage exceeds the output. The pins have a 0.1" spacing, making this board compatible with standard solderless breadboards and perfboards.

Dimensions

Size: 0.35" × 0.45" × 0.1"¹

Weight: 0.4 g¹

General specifications

Minimum operating voltage: 0.5 V

Maximum operating voltage: 5.5 V

Maximum input current: 1.2 A²

Output voltage: 5 V

Reverse voltage protection?: N

Maximum quiescent current: 1 mA³



Ek 63 Kullanılan Ekipmanın Teknik Özellikleri

18. Kamera FPV Transmitter Eachine TX01AIO 5.8G 40CH 25MW VTX

(1)



https://www.eachine.com/Eachine-TX01-Super-Mini-AIO-5_8G-40CH-25MW-VTX-600TVL-1-or-4-Cmos-FPV-Transmitter-p-418.html



Eachine TX01 Super Mini AIO 5.8G 40CH 25MW VTX 600TVL 1/4 Cmos FPV Transmitter

Specification:

Item name: TX01 AIO FPV VTX Camera
Output power: 25mW
Input power: 3.3-5V
Current consumption: 3.3-5V, 200mA (type)
Camera resolution: 600TVL
Video system: NTSC
Mini illumination: 1 lux
Field of view: 120 degree viewing angle
Antenna dimensions: 22x29mm diameter
Antenna: 4 lobe right hand circular polarized
Frequency: 5.8GHz 5 bands 40 channel
Size: 20x13x6mm
Weight: 4.48g

18. Kamera FPV Transmitter Eachine TX01AIO 5.8G 40CH 25MW VTX

(2)

Feature:

Super mini size and light weight

Power failure memory, remember the frequency channel after powering off

Nickel-plated 4 Leaf Antenna

Five bands 40 channels with Raceband, show on LED Digital Display

Five bands 40 channels with Raceband,
Show on LED Digital Display

FR	CH	CH							
		CH 1	CH 2	CH 3	CH 4	CH 5	CH 6	CH 7	CH 8
FR	FR1(A)	5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
	FR2(b)	5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
	FR3(E)	5705M	5685M	5665M	5645M	5885M	5905M	5925M	5945M
	FR4(F)	5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M
	FR5 (r)	5658M	5695M	5732M	5769M	5806M	5843M	5880M	5917M

https://tr.banggood.com/Eachine-TX01-NTSC-Super-Mini-AIO-5_8G-40CH-25MW-VTX-600TVL-1-or-4-Cmos-FPV-Camera-p-1088374.html?rmmids=myorder&cur_warehouse=CN

Marka adı: Eachine

Item name: TX01 AIO FPV VTX Camera

Çıkış gücü: 25mW

Input power: 3.3-5V

Akım tüketimi: 3.3-5V, 200mA (tip)

Camera resolution: 600TVL

Video sistemi: NTSC

Mini illumination: 1 lux

Görüş alanı: 120 derece görüş açısı

Antenna dimensions: 22x29mm diameter

Anten: 4 lob sağdan dairesel polarize

Frequency: 5.8GHz 5 bands 40 channel

Boyut: 20x13x6mm

Ağırlık: 4.48 g

Özellik:

Süper mini Numara ve hafif

Power failure memory, remember the frequency channel after powering off

Nikel kaplama 4 Yaprak Anten

Five bands 40 channels with Raceband, show on LED Digital Display

18. Kamera FPV Transmitter Eachine TX01AIO 5.8G 40CH 25MW VTX

(3)

PRODUCT SPECIFICATION

Frequency and channel frequency table:

FR	CH	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
FR1/Band A		5865M	5845M	5825M	5805M	5785M	5765M	5745M	5725M
FR2/Band b		5733M	5752M	5771M	5790M	5809M	5828M	5847M	5866M
FR3/Band E		5705M	5685M	5665M	5645M	5625M	5605M	5585M	5565M
FR4/Band F		5740M	5760M	5780M	5800M	5820M	5840M	5860M	5880M
FR5/Band r		5658M	5695M	5732M	5769M	5806M	5843M	5880M	5917M



www.eachine.com



EACHINE TX01

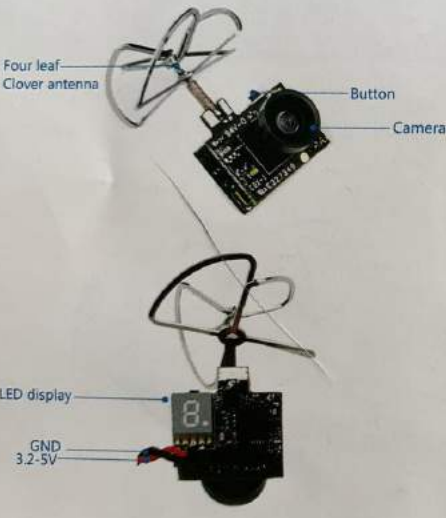
Super Mini 5.8GHz 40CH FPV Transmitter + 600TVL FPV Camera + Four leaf antenna



25mW

Product Instruction Manual

PRODUCT SPECIFICATION



Four leaf Clover antenna

Button

Camera

LED display

GND

3.2-5V

25mW Super Mini 5.8GHz 40CH FPV Transmitter + 600TVL FPV Camera + Four leaf antenna

PRODUCT SPECIFICATION

Operation instruction:

Output power: 25mW
 Input power: 3.2-5V
 Current consumption: 3.2V-5V, 250mA (type)
 Camera resolution: 600TVL
 Video system: PAL
 Mini illumination: 1 lux
 Field of view: 120° viewing angle
 Antenna dimensions: 22mm X 29mm diameter
 Antenna: 4 lobe right hand circular polarized
 Frequency: 5.8GHz 5 bands 40 channels, with Raceband: 5658-5917MHz
 Size: 18.7*13mm (L*W)
 Weight: 4.3g

Specification and Parameters:

1. Three adjustable gears, on/off switch, CH, FR, automatically cycle display three gears on LED after power on.
2. Short press button to adjust "CH", 1-8 adjustable.
3. Press and hold button 2s, transfer to FR, A, b, E, F, r adjustable, then short press button to adjust FR.
4. Press and hold button 5s to adjust on/off, "0" and "=" adjustable, short press button to adjust, "0" stands for "off", and "=" stands for "on".

19. FPV Monitor

5.8G 48CH 4.3 Inch LCD 480x272 16:9 NTSC/PAL FPV Monitor Auto Search With OSD Build-in Battery



Description:

Item Name: FPV Monitor

Frequency Range: 5645MHz~5945MHz

Channel: 48

Screen Size: 4.3"

Resolution: 480x272

Aspect Ratio: 16:9

Brightness: 500cd/m²

Video Format: NTSC/PAL

Operating Voltage: 5V

Power Consumption: 500mA

Built-in Battery: 600mAh (if you receive a 210mAh battery, please contact us, we will resend you a 600mAh battery)

Working Time: 2 Hours

Sensitivity: -90dBm

Working Status OSD Display: Battery indicator, Channel not working

One Key Search Frequency: Yes, Button control

Support Language: Chinese, English, Russian, Spanish, Germany

Features:

Small, light weight, durable.

4.3 inch LCD wireless FPV monitor.

6 frequency bands and total 48 channels.

One key to automatic signal search, simple operation.

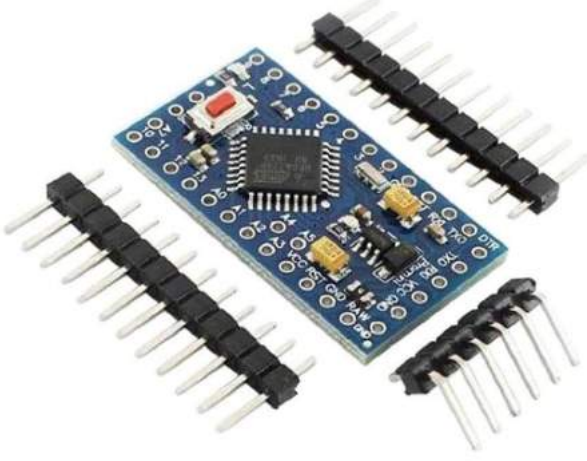
Built-in battery, easy to carry and outdoor use.

With battery and channel work status OSD display.

High definition and high receiving sensitivity.

https://www.banggood.com/5_8G-48CH-4_3-Inch-LCD-480x272-16-9-NTSC-or-PAL-FPV-Monitor-Auto-Search-With-OSD-Build-in-Battery-p-1364626.html

20. Arduino Pro Mini 328 - 3.3 V / 8 MHz (Header'lı) (1)



Arduino Pro Mini kartı yanında erkek header ile beraber lehimsiz gönderilir. Pro Mini modeli özellik olarak Mini modeline çok benzerdir. Ancak pin sırası olarak Mini kartından farklı bir yapıdadır, birbirlerinin yerine kullanılamaz.

Ürün İçeriği:

- Arduino Pro Mini (3.3V/8 MHz)
- Erkek Header

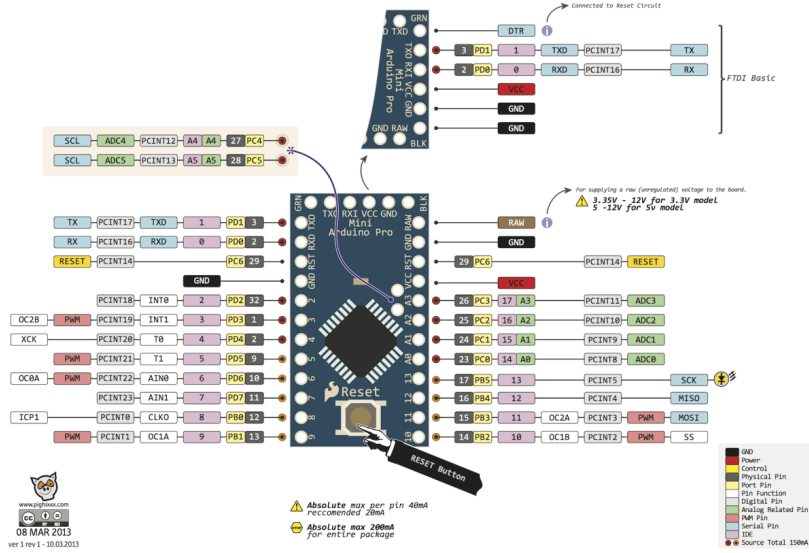
Arduino Pro Mini; Atmega328 temelli bir mikrodenetleyici kartıdır. Üzerinde 14 adet dijital giriş/çıkış pini (6 tanesi PWM çıkışı olarak kullanılabilir), 8 analog giriş, 8Mhz kristal ve reset tuşu bulundurmaktadır. Breadboard üzerinde kullanmak için ve boyutun önemli olduğu yerlerde kullanılmak üzere yapılmıştır. Pro Mini üzerinde usb soketi ve programlayıcı bulunmamaktadır. Kartı programlamak için [USB-Serial Dönüştürücü](#) veya başka usb-seri dönüştürücüler kullanılabilir.

Pro Mini'nin 2 versiyonu bulunmaktadır. 3.3V/8 MHz ve 5V/16 MHz.

Teknik Özellikler:

- Mikrodenetleyici ATmega328
- Çalışma Gerilimi 3.3V veya 5V (Modele bağlıdır)
- Giriş Gerilimi (önerilen) 5V-12V (3.3V modeli) veya 7-12V (5V modeli)
- Dijital I/O Pinleri 14 (6 tanesi PWM çıkışı)
- Analog Giriş Pinleri 8
- Her I/O için Akım 40 mA
- Flash Hafıza 32 KB (ATmega328) 0.5 KB kadarı bootloader tarafından kullanılmaktadır
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Saat Hızı 8 MHz (3.3V modeli) veya 16 MHz (5V modeli)

20. Arduino Pro Mini 328 - 3.3 V / 8 MHz (Header'lı) (2)



Güç:

Arduino Pro Mini gücünü 5V pini üzerinden veya harici güç kaynağından alabilir. 5V VCC pini üzerinden verilir. Harici güç kaynağı AC-DC adaptör olabileceği gibi bataryada olabilir. Adaptör veya batarya kart üzerindeki GND ve RAW pinleri üzerinden bağlanabilir.

Harici güç kaynağı olarak 7-12V arası kullanılabilir. 12V üzerinde gerilim uygulamak karta zara verebilir.

Pro Mini kartının üzerindeki mikrodenetleyicinin çalışma gerilimi 5V'dur. RAW pini üzerinden verilen 7-12V arası gerilim kart üzerinde bulunan voltaj regülatörü ile 5V'a düşürülerek karta dağılır.

Güç pinleri aşağıdaki gibidir:

- **RAW:** Harici güç kaynağı kullanılırken 7-12V arası gerilim giriş pini.
- **VCC:** Bu pin regülatörden çıkan 5V çıkışı verir. Eğer kart 5V ile beslenecekse yine bu pin 5V giriş pini olarak kullanılabilir. Eğer karta güç RAW üzerinden veriliyorsa regülatörden çıkan 5V doğrudan bu pin üzerinden çıkış olarak verilir.
- **GND:** Toprak pinleridir.

Not: Pro Mini üzerinde 3.3V regülatör bulunmamaktadır.

Hafıza:

Atmega328 32 KB'lık flash belleğe sahiptir (0.5 KB kadarı bootloader tarafından kullanılmaktadır). 2 KB SRAM ve 1 KB EEPROM'u bulunmaktadır.

Giriş ve Çıkış:

Pro Mini üzerindeki 14 adet dijital pinin hepsi giriş veya çıkış olarak kullanılabilir. 8 tane analog giriş pini de bulunmaktadır. Yine Atmega328 kullanılmasına rağmen Uno üzerinde 6 tane analog giriş pini bulunmaktaydı. Pro Mini üzerinde kullanılan Atmega328 mikrodenetleyicisinin kılıfında 8 tane analog giriş pini bulunmaktadır. Bu analog giriş pinleri de aynı şekilde dijital giriş ve çıkış olarak kullanılabilir. Yani kart üzerinde toplam 22 tane dijital giriş çıkış pini vardır. Bu pinlerin tamamının lojik seviyesi 5V'dur. Her pin maks. 40mA giriş ve çıkış akımı ile çalışır. Ek olarak, bazı pinlerin farklı özellikleri bulunmaktadır. Özel pinler aşağıda belirtildiği gibidir.

20. Arduino Pro Mini 328 - 3.3 V / 8 MHz (Header'lı) (3)

- **Seri Haberleşme, 0 (RX) ve 1 (TX):** TTL Seri veri alıp (RX), vermek (TX) için kullanılır. Bu pinler üzerinden usb-seri dönüştürücüsüne bağlanarak kart programlanabilir. Yani bilgisayardan karta kod yüklerken veya bilgisayar-pro mini arasında karşılıklı haberleşme yapılırken bu pinler kullanılır. O yüzden karta kod yüklerken veya haberleşme yapılırken hata olmaması için mecbur kalınmadıkça bu pinlerin kullanılmamasında fayda vardır.
 - **Harici Kesme, 2 (interrupt 0) ve 3 (interrupt 1):** Bu pinler yükselen kenar, düşen kenar veya değişiklik kesmesi pinleri olarak kullanılabilir. Ayrıntılı bilgi için [attachInterrupt\(\)](#) fonksiyon sayfasını inceleyebilirsiniz.
 - **PWM, 3,5,6,9,10 ve 11:** 8-bit çözünürlükte PWM çıkış pinleri olarak kullanılabilir.
 - **SPI, 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK):** Bu pinler SPI haberleşmesi için kullanılır.
 - **Analog, A0-A7:** Pro Mini 8 tane 10-bit çözünürlüğünde analog giriş pinine sahiptir. Bu pinler dijital giriş ve çıkış için de kullanılabilir. Pinlerin ölçüm aralığı 0-5V'dur.
 - **I2C, A4 (SDA), A5 (SCL):** Bu pinler I2C haberleşmesi için kullanılır.
 - **Reset:** Mikrodenetleyici resetlenmek istendiğinde bu pin LOW yapılır. Reset işlemi kart üzerinde bulunan Reset Butonu ile de yapılabilir.
- Arduino pro mini ile Atmega328 arasındaki [pin haritalaması](#) sayfasını inceleyebilirsiniz.

Haberleşme:

Arduino Pro Mini'nin bilgisayarla, başka bir arduino veya mikrodenetleyici ile haberleşmesi için bir kaç farklı seçenek vardır. Atmega328, 0 (RX) ve 1 (TX) pinleri üzerinden UART TTL (5V) seri haberleşme imkanı sunar.

Pro Mini üzerinde donanımsal olarak bir adet seri port bulunmaktadır. Ancak [SoftwareSerial kütüphanesi](#) ile bu sayı yazılımsal olarak artırılabilir.

Atmega328 aynı şekilde I2C ve SPI portları da sağlamaktadır. Arduino bilgisayar programı ile gelen [Wire kütüphanesi](#) I2C kullanımını, [SPI kütüphanesi](#) de SPI haberleşmesini sağlamak için kullanılır.

Dökümanlar:

- [Atmega328 Datasheet](#)
- [Sürücü Yükleme Dökümanı](#) (Windows)
- [Kart Şematiği](#)
- [Eagle PCB Çizim Dosyaları](#)
- [Arduino Yazılımı](#) (Arduino IDE)
- [Arduino Sayfası](#)

https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

Mikroişlemci Çalışma Sıcaklığı : -40 ile 105°C arası

Temin Edildiği Adres :

<https://www.robotistan.com/arduino-pro-mini-328-3v8mhz-headerli>

21. Gaz Sensörü MQ7 (1)

HANWEI ELECTRONICS CO., LTD

MQ-7

<http://www.hwsensor.com>

TECHNICAL DATA MQ-7 GAS SENSOR

FEATURES

- * High sensitivity to carbon monoxide
- * Stable and long life

APPLICATION

They are used in gas detecting equipment for carbon monoxide(CO) in family and industry or car.

SPECIFICATIONS

A. Standard work condition

Symbol	Parameter name	Technical condition	Remark
Vc	circuit voltage	5V±0.1	Ac or Dc
VH(H)	Heating voltage (high)	5V±0.1	Ac or Dc
VH(L)	Heating voltage (low)	1.4V±0.1	Ac or Dc
RL	Load resistance	Can adjust	
RH	Heating resistance	33Ω±5%	Room temperature
TH(H)	Heating time (high)	60±1 seconds	
TH(L)	Heating time (low)	90±1 seconds	
PH	Heating consumption	About 350mW	

b. Environment conditions

Symbol	Parameters	Technical conditions	Remark
Tao	Using temperature	-20℃-50℃	
Tas	Storage temperature	-20℃-50℃	Advice using scope
RH	Relative humidity	Less than 95%RH	
O ₂	Oxygen concentration	21%(stand condition) the oxygen concentration can affect the sensitivity characteristic	Minimum value is over 2%

c. Sensitivity characteristic

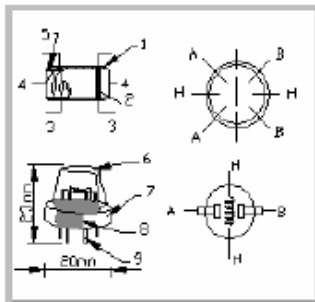
symbol	Parameters	Technical parameters	Remark
Rs	Surface resistance Of sensitive body	2-20k	In 100ppm Carbon Monoxide
a (300/100ppm)	Concentration slope rate	Less than 0.5	Rs (300ppm)/Rs(100ppm)
Standard working condition	Temperature -20℃±2℃ Vc:5V±0.1V VH:5V±0.1V VH:1.4V±0.1V	relative humidity 65%±5%	RL:10KΩ±5%
Preheat time	No less than 48 hours	Detecting range: 20ppm-2000ppm carbon monoxide	

D. Structure and configuration, basic measuring circuit

Structure and configuration of MQ-7 gas sensor is shown as Fig. 1 (Configuration A or B), sensor composed by micro AL₂O₃ ceramic tube, Tin Dioxide (SnO₂) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components. The enveloped MQ-7 have

21. Gaz Sensörü MQ7 (2)

6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current.



Parts	Materials
1 Gas sensing layer	SnO ₂
2 Electrode	Au
3 Electrode line	Pt
4 Heater coil	Ni-Cr alloy
5 Tubular ceramic	Al ₂ O ₃
6 Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7 Clamp ring	Copper plating Ni
8 Resin base	Bakelite
9 Tube Pin	Copper plating Ni

Fig.1

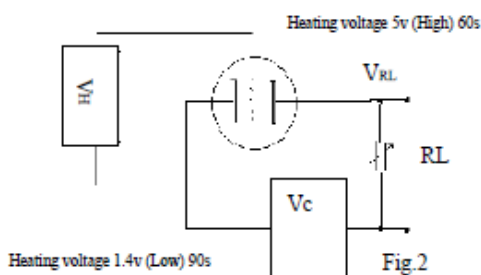
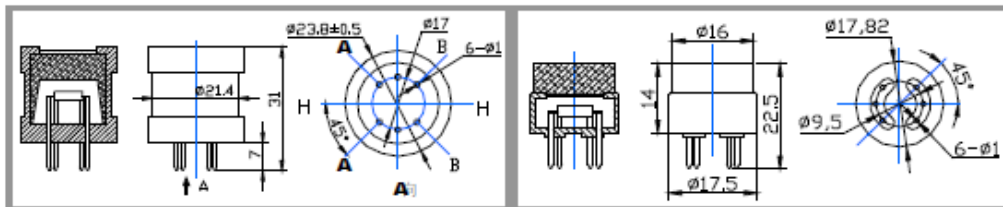


Fig.2

Standard circuit:

As shown in Fig 2, standard measuring circuit of MQ-7 sensitive components consists of 2 parts. one is heating circuit having time control function (the high voltage and the low voltage work circularly). The second is the signal output circuit, it can accurately respond changes of surface resistance of the sensor.

Electric parameter measurement circuit is shown as Fig.2

E. Sensitivity characteristic curve

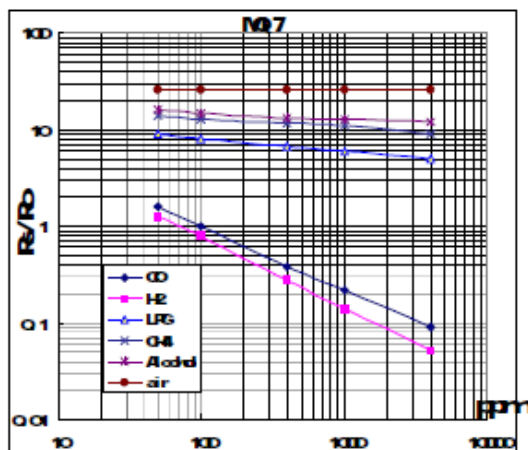


Fig.3 sensitivity characteristics of the MQ-7

Fig.3 is shows the typical sensitivity characteristics of the MQ-7 for several gases.

in their: Temp: 20°C,

Humidity: 65%.

O₂ concentration 21%

RL=10k Ω

R₀: sensor resistance at 100ppm

CO in the clean air.

R_s: sensor resistance at various

concentrations of gases.

21. Gaz Sensörü MQ7 (3)

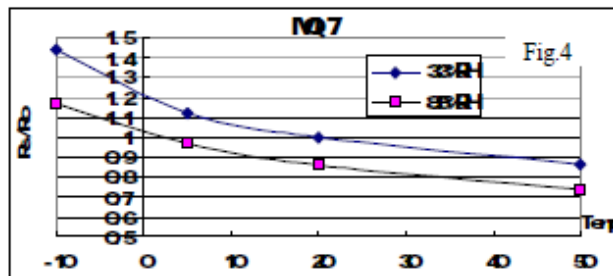


Fig.4 is shows the typical dependence of the MQ-7 on temperature and humidity.

Ro: sensor resistance at 100ppm CO in air at 33%RH and 20degree.

Rs: sensor resistance at 100ppm CO at different temperatures and humidities.

OPERATION PRINCIPLE

The surface resistance of the sensor R_s is obtained through effected voltage signal output of the load resistance R_L which series-wound. The relationship between them is described:

$$R_s/R_L = (V_c - V_{RL}) / V_{RL}$$

Fig. 5 shows alterable situation of R_L signal output measured by using Fig. 2 circuit output

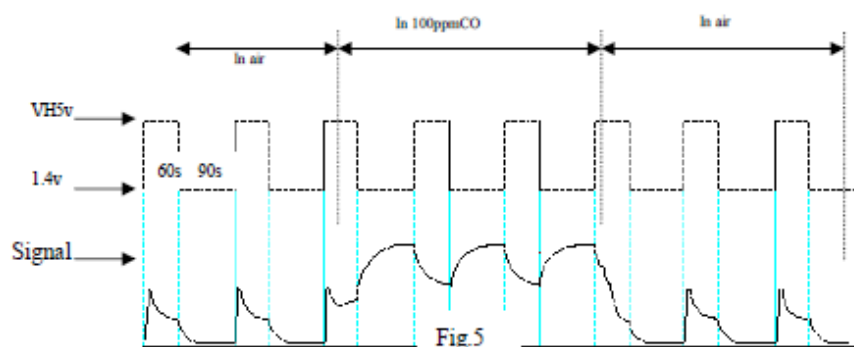


Fig 5

signal when the sensor is shifted from clean air to carbon monoxide (CO), output signal measurement is made within one or two complete heating period (2.5 minute from high voltage to low voltage).

Sensitive layer of MQ-7 gas sensitive components is made of SnO_2 with stability, So, it has excellent long term stability. Its service life can reach 5 years under using condition.

SENSITIVITY ADJUSTMENT

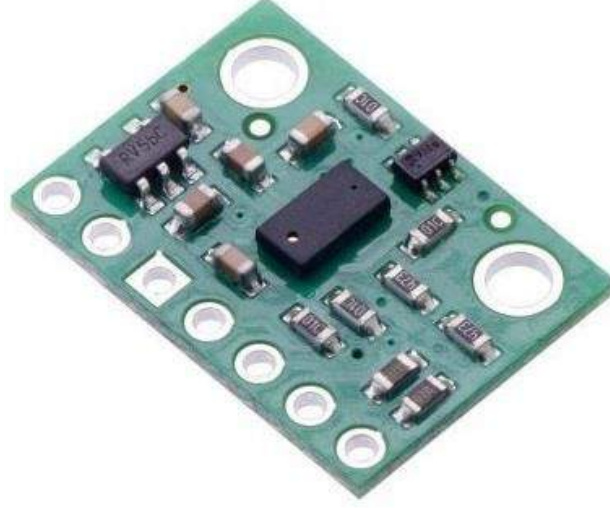
Resistance value of MQ-7 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary. we recommend that you calibrate the detector for 200ppm CO in air and use value of Load resistance that(R_L) about 10 K Ω (5K Ω to 47 K Ω).

When accurately measuring, the proper alarm point for the gas detector should be determined after considering the temperature and humidity influence. The sensitivity adjusting program:

- Connect the sensor to the application circuit.
- Turn on the power, keep preheating through electricity over 48 hours.
- Adjust the load resistance R_L until you get a signal value which is respond to a certain carbon monoxide concentration at the end point of 90 seconds.
- Adjust the another load resistance R_L until you get a signal value which is respond to a CO concentration at the end point of 60 seconds.

[Supplying special IC solutions, More detailed technical information, please contact us.](#)

22. VL53L0X Time-of-Flight Regülatörlü Mesafe Sensörü



Time-of-flight mesafe ölçümü, bir cisme gönderilen ışığın cisme ulaşip yansiyarak sensöre geri geldiği sürenin hesaplanması prensibiyle çalışır.

Bu kartta bulunan VL53L0X sensörü, kızılötesi dalgaboyunda gönderdiği pulsler sayesinde 2 metreye kadar olan mesafeleri ölçebilme yeteneğine sahiptir. Kart üzerinde bulunan 2.8V regülatör sayesinde 2.6-5.5V arasındaki gerilimler ile sorunsuzca kullanılabilir. I2C haberleşme arayüzünü kullanır. 0.1" header bağlantıları sayesinde standart breadboardlar, delikli kartlar ve baskılı devrelerde rahatlıkla kullanılabilir.

Özellikler:

- Boyutlar: 13 mm x 18 mm x 2 mm
- Ağırlık (headerlar hariç): 0.5gr
- Çalışma gerilimi: 2.6-5.5V
- Çalışma akımı: 10mA (tipik ortalama değer, maksimum 40mA'e kadar çıkabilmektedir)
- Çıkış: I2C üzerinden 16-bit milimetre cinsinden mesafe değeri
- Ölçüm aralığı: 2 metreye kadar (efektif menzil konfigürasyon, obje ve ortam koşullarına göre değişiklik gösterebilir)
- Minimum ölçüm mesafesi: ~3cm

Dökümanlar:

- [VL53L0X datasheet](https://www.pololu.com/file/0J1187/VL53L0X.pdf)
- <https://www.pololu.com/file/0J1187/VL53L0X.pdf>
- [Arduino kütüphanesi](#)
- [Raspberry Pi API'si](#)
- [Devre şeması](#)
- <https://www.pololu.com/file/0J1188/vl53l0x-time-of-flight-distance-sensor-carrier-schematic.pdf>
- <https://www.robotistan.com/vl6180x-time-of-flight-mesafe-sensoru-1>

23. Redüktörlü Mikro DC Motor 12V 12mm 60 RPM



12V 12mm 60 RPM Redüktörlü Mikro DC Motor

Motor, redüktörlü yapısı sayesinde yüksek torklu çıkış sağlamakta çok büyük bir kolaylık sağlamaktadır. Redüktörlü yapısının yanı sıra dış kasasının malzemesinin de metal yapılması sayesinde uzun ömürlü, dayanıklı ve stabil bir sonuç elde etmenize olanak verir. Bu parça iki telli bir bağlantı kullanır ve yön kontrolünün yanı sıra PWM hız kontrolünü de destekler.

Özellikler:

- Çalışma Voltajı: 12V
- Devir: 60 Rpm/Dk
- Motor Çapı: 12mm
- Boşta Çektiği Akım: 80mA \pm 40mA
- Zorlanma Akımı: 0.45A
- Redüktör Çapı: 12mm
- Mil: 3mm D Şaft
- Mil Uzunluğu: 10mm
- Tork: 15 kg/cm
- Uzunluk (Mil Hariç): 24mm
- Ağırlık: 9.5gr

Temin edildiği Web adresi

<https://www.robotistan.com/12v-12mm-60-rpm-reduktorlu-dc-motor>

24. Bakır Bant 30mt 5mm



5mm Bakır Bant 30mt - Akrilik Yapışkanlı

BAKIR BANT, akrilik yapışkanlık özelliğine sahiptir. Bakır bantların elektrik ileten bakır folyo taşıyıcıları bulunmaktadır. İnce saf bakırdan üretilmiş esnek bir yapıya sahiptir. Bakır Bant yapıştığı yüzeyin şeklini almaktadır. üzerine lehim yapılabilen Bakır Bant bir kablo kadar akım taşıyabilmektedir. Yapışkanının altında koruyucu silikonlu kağıt bulunmaktadır. üzerine lehim yapılabilmesi kullanım alanlarını artırmakta olup Elektrik ve elektronik sektöründe, sanayi sektöründe, inşaat sektöründe, sıhhi tesisat sektöründe, sanayi sektöründe, otomobil sektöründe, savunma sanayin de birçok kullanım alanına sahiptir.

Bakır Bant Özellikleri

- Boyut: 5mm x 30m
- Malzeme:% 99.98 CU
- Kolloidal bileşenler: iletken yapışkan (akrilik yapışkanın ısıyla indüksiyonu)
- Yapışma: 1.5 ~ 1.3kg / 25mm
- Sıcaklık dayanımı -10 °C --- 120 °C
- 4.5 ~ 4.8 kg / mm yoğunluğu
- % 7-7 ila% 3-4 uzama

<https://www.direnc.net/5mm-termal-bant-30mt>

Ek 63 Kullanılan Ekipmanın Teknik Özellikleri

25. Redüktörlü Dc Motor 6V 30Rpm 15mm

Kategori
15mm Serisi

Marka
Keskinler



6V 30Rpm 15mm Redüktörlü Dc Motor

DC 6V'ta 30 Rpm hıza sahip redüktörlü motordur. Engelden kaçan, çizgi izleyen gibi çeşitli robot projelerinizde kullanabilirsiniz. Uzun mil gerektiren projeleriniz için idealdir.

Teknik Özellikleri;

Çalışma Voltajı: 6V

Hızı: 30 Rpm

Boşta Çektiği Akım: 300 mA

Zorlanma Akımı: 2.3A

Mil Uzunluğu : 7 cm

Temin edildiği adres:

<https://www.motorobit.com/urun/6v-30rpm-15mm-reduktorlu-dc-motor>

26. Lipo Batarya 7.4 V 2S 1350 mAh 25C



Lipo piller şarj edilebilir olduğundan dolayı sizlere yüksek akım sağlar.

Özellikler:

- Hücre Sayısı: 2s
- Voltaj: 7.4V
- Akım Kapasitesi: 1350mAh
- Ağırlık: 85gr
- Boyutları: 63x32x14mm
- JST çıkış soketi ve balancer şarj soketi mevcuttur.
- 25C (50C anlık, maksimum 10sn)

Temin edilen adres:

<https://www.robotzade.com/urun/7-4-v-2s-lipo-batarya-1350-mah-25c>