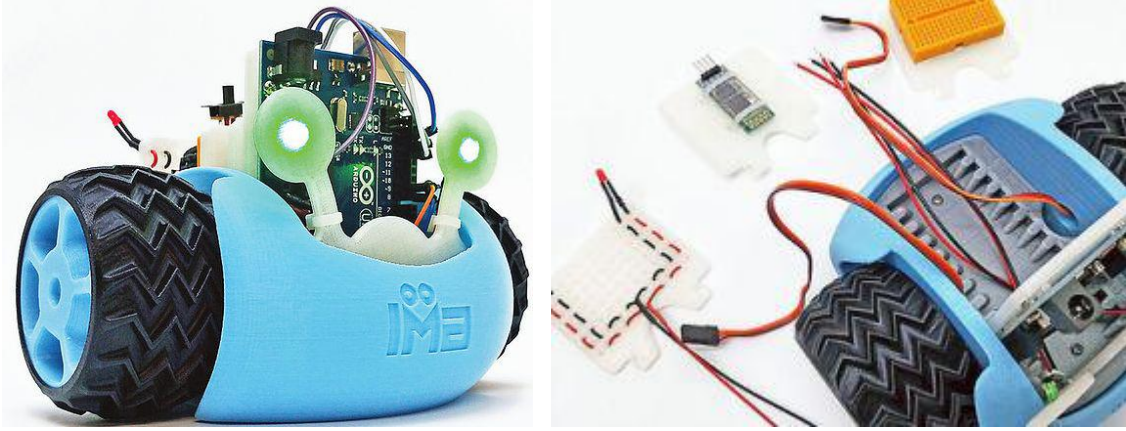


IMA Juno'yu Kıpırdatmak

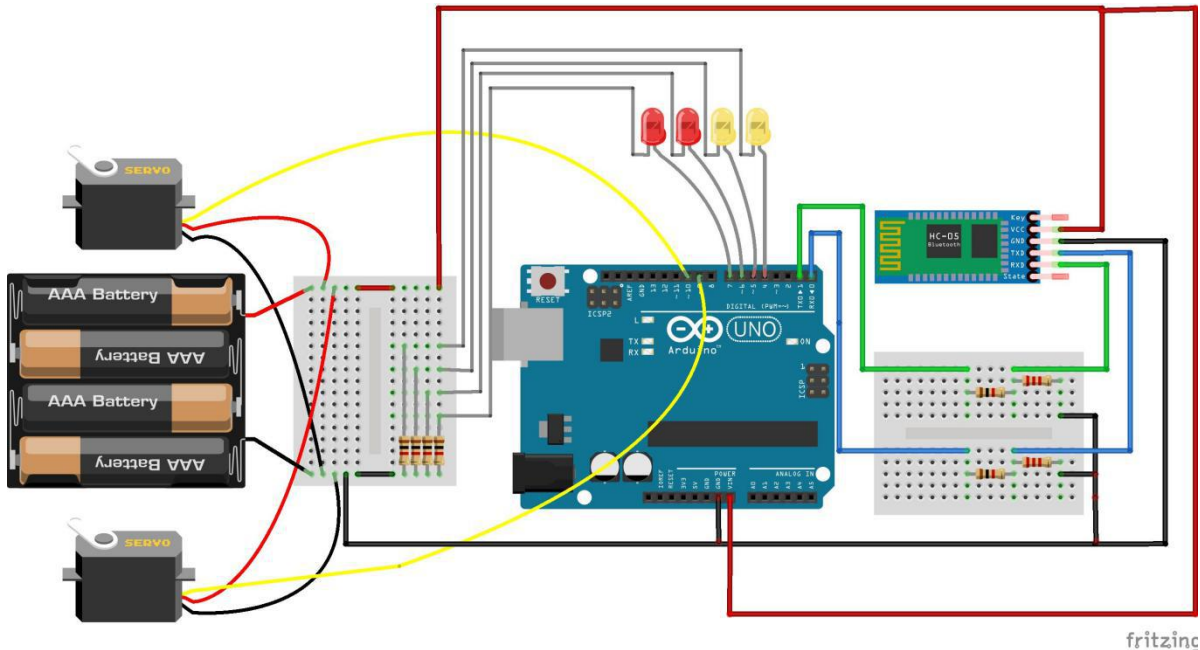
IMA Juno denilen ve bluetooth üzerinden kontrol edilebilen aşağıdaki 2 tekerli şirin robotun [STL dosyası](#) ve [Arduino kodu kendi sitesinde](#) sunulmaktadır. Tekerleri ve şasisi dahil plastik aksamının tamamı 3 boyutlu yazıcıdan alınan bu robotun Google Play Store'daki kendi Android uygulaması bluetooth eşleşmesi yapılmasına rağmen Juno'yu algılamadı ve dolayısıyla çalıştırmadı.



Bu çalışmada Juno'yu hareket ettirebilmek için [“Arduino ile Bluetooth Kontrollü Araba Uygulaması”](#) Juno'ya göre revize edildi. Bluetooth kontrollü araba ile Juno arasındaki donanımsal en büyük fark Juno'da DC motor yerine 360 derecelik servo motor kullanılmasıydı. Bunun için Arduino kodunun servo motorlara göre tekrardan düzenlenmesi gerekti. Servo motorların kullanımı ile ilgili ufak bir araştırma ile bilinmesi gereken temel noktalardan bazıları şöyle sıralanabilir:

- Servo kütüphanesi write() fonksiyonu değeri; standart servoda milin dönme açısını belirlerken 360 derecelik servoda milin dönme hızını belirler.
- write() fonksiyonu değeri 90 ise 360'lık servo hareket etmez.
- write() fonksiyonu değeri 0-90 arası ise mil bir yöne, 90-180 arası ise diğer yöne hareket eder. 0 ve 180 servonun en hızlı döndüğü uç değerlerdir.
- Servo.h kütüphanesi koda eklendiğinde Arduino Uno'da 9. ve 10. pinlerin PWM özelliği servo motorun bu pinlere bağlanıp bağlanmadığına bakılmaksızın pasif hale getirilir.
- FS90R'nin (360'lık servo) yüksüz 4.8V'de çektiği akım 100mA iken 6V'da 120mA'dir.
- Servolar fazla akım çekebileceğinden bir ve ikiden fazla sürülmesi gerektiğinde ayrı bir kaynaktan (Arduino'nun +5V pininden değil) beslenmesi gerekmektedir.

IMA Juno için devre tasarımı aşağıdaki gibi değiştirildi:



Revize edilmiş Arduino kodumuz ise şöyledir:

```
1  /** IMA Juno Alternative **/  
2  #include <Servo.h>  
3  
4  Servo solServo, sagServo;  
5  
6  /*Servo veri pinleri*/  
7  int solServoPin = 9; //Sol servo motor veri pini  
8  int sagServoPin = 10; //Sağ servo motor veri pini  
9  
10 String satir = ""; //Seri porttan okunan satir  
11 int yon = 0; //Gelen yön verisi  
12 int hiz = 0; //Gelen hiz verisi  
13 int semiColonPos = 0; //Noktalı virgül pozisyonu  
14  
15 int solServoHizi = 0; //Sol servo motor hizi  
16 int sagServoHizi = 0; //Sağ servo motor hizi  
17  
18 /*Ledler*/  
19 int solLed = 4;  
20 int sagLed = 5;  
21 int arkaSolLed = 6;  
22 int arkaSagLed = 7;  
23  
24  
25 void setup() {  
26  
27     solServo.attach(solServoPin);  
28     sagServo.attach(sagServoPin);  
29  
30     pinMode(solLed, OUTPUT);  
31     pinMode(sagLed, OUTPUT);  
32     pinMode(arkaSolLed, OUTPUT);  
33     pinMode(arkaSagLed, OUTPUT);  
34  
35     digitalWrite(solLed, HIGH);  
36     digitalWrite(sagLed, HIGH);  
37     digitalWrite(arkaSolLed, HIGH);  
38     digitalWrite(arkaSagLed, HIGH);  
39  
40     Serial.begin(9600);  
41     //Zaman aşımı süresi appinventor timer süresi (15ms)  
42     //ile uyumlu hale getiriliyor  
43     Serial.setTimeout(15);  
44 }  
45  
46 void loop() {  
47     if (Serial.available()>0)  
48     {  
49         satir = Serial.readString(); //15ms zaman aşımı  
50         /*  
51          * Gönderilen veri Yön;Hız formatındadır.  
52          * Yön için: 1-> Sağa, 2->Sola, 3->İleri, 4->Geri, 0->Dur  
53          * Hız için değer aralığı 0..90'dır.  
54          */  
55         semiColonPos = satir.indexOf(';');  
56         yon = satir.substring(0,semiColonPos).toInt();  
57         hiz = satir.substring(semiColonPos+1).toInt();  
58  
59         solServoHizi = yon == 4 ? 90 - hiz : 90 + hiz;  
60         sagServoHizi = yon == 4 ? 90 + hiz : 90 - hiz;  
61  
62         switch (yon)  
63         {  
64             case 1: //Sağa dön  
65                 solServo.write(solServoHizi);  
66                 sagServo.write(90);  
67                 Serial.print("Sağa dönüyorum");  
68  
69                 break;  
70             case 2: //Sola dön  
71                 solServo.write(90);  
72                 sagServo.write(sagServoHizi);  
73                 Serial.print("Sola dönüyorum");  
74  
75                 break;
```

```

76     case 3: //İleri
77         solServo.write(solServoHizi);
78         sagServo.write(sagServoHizi);
79         Serial.print("İleri gidiyorum");
80
81         break;
82     case 4: //Geri
83         solServo.write(solServoHizi);
84         sagServo.write(sagServoHizi);
85         Serial.print("Geri gidiyorum");
86
87         break;
88     case 0: //Dur
89         solServo.write(90);
90         sagServo.write(90);
91         Serial.print("Durdum");
92
93         break;
94     }
95
96     ledYak();
97 }
98
99 }
100
101
102 void ledYak()
103 {
104     switch (yon)
105     {
106         case 1: //Sağa dön
107             digitalWrite(solLed, LOW);
108             digitalWrite(sagLed, HIGH);
109             digitalWrite(arkaSolLed, LOW);
110             digitalWrite(arkaSagLed, LOW);
111
112             break;
113         case 2: //Sola dön
114             digitalWrite(solLed, HIGH);
115             digitalWrite(sagLed, LOW);
116             digitalWrite(arkaSolLed, LOW);
117             digitalWrite(arkaSagLed, LOW);
118
119             break;
120         case 3: //İleri
121             digitalWrite(solLed, LOW);
122             digitalWrite(sagLed, LOW);
123             digitalWrite(arkaSolLed, LOW);
124             digitalWrite(arkaSagLed, LOW);
125
126             break;
127         case 4: //Geri
128             digitalWrite(solLed, LOW);
129             digitalWrite(sagLed, LOW);
130             digitalWrite(arkaSolLed, LOW);
131             digitalWrite(arkaSagLed, LOW);
132
133             break;
134         case 0: //Dur
135             digitalWrite(solLed, LOW);
136             digitalWrite(sagLed, LOW);
137             digitalWrite(arkaSolLed, HIGH);
138             digitalWrite(arkaSagLed, HIGH);
139
140             break;
141     }
142 }

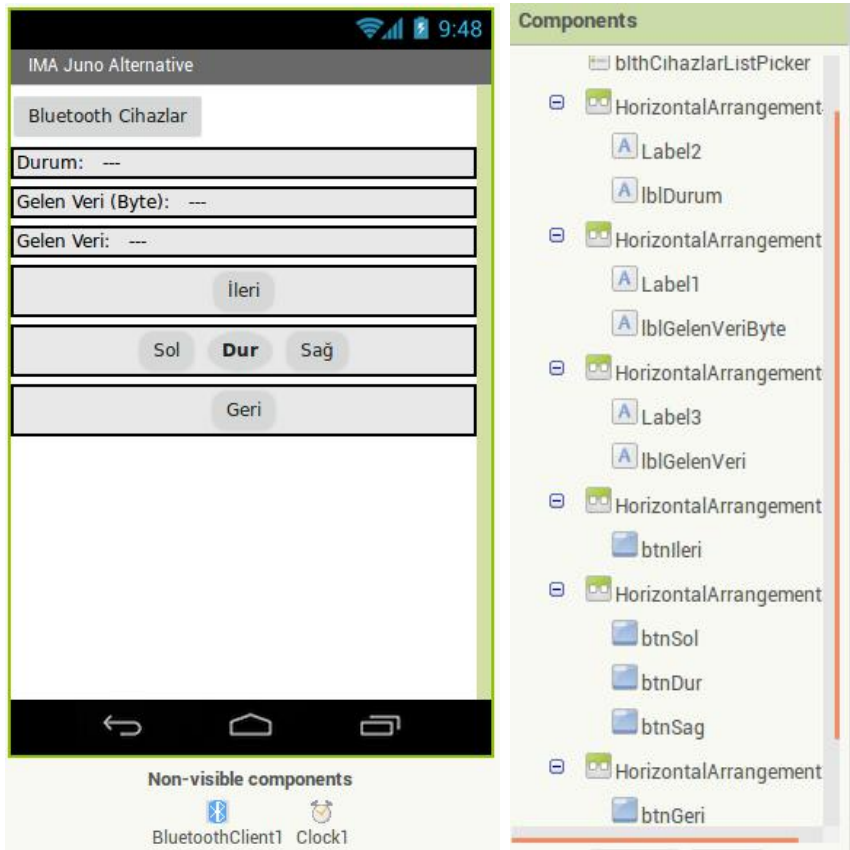
```

Juno'nun tekerlerinin ikisinin de aynı yönde dönmesi için sol ve sağ servoların hız değerlerinin birbirinin tersi yönde olacak şekilde hesaplanmış olduğuna dikkat edelim.

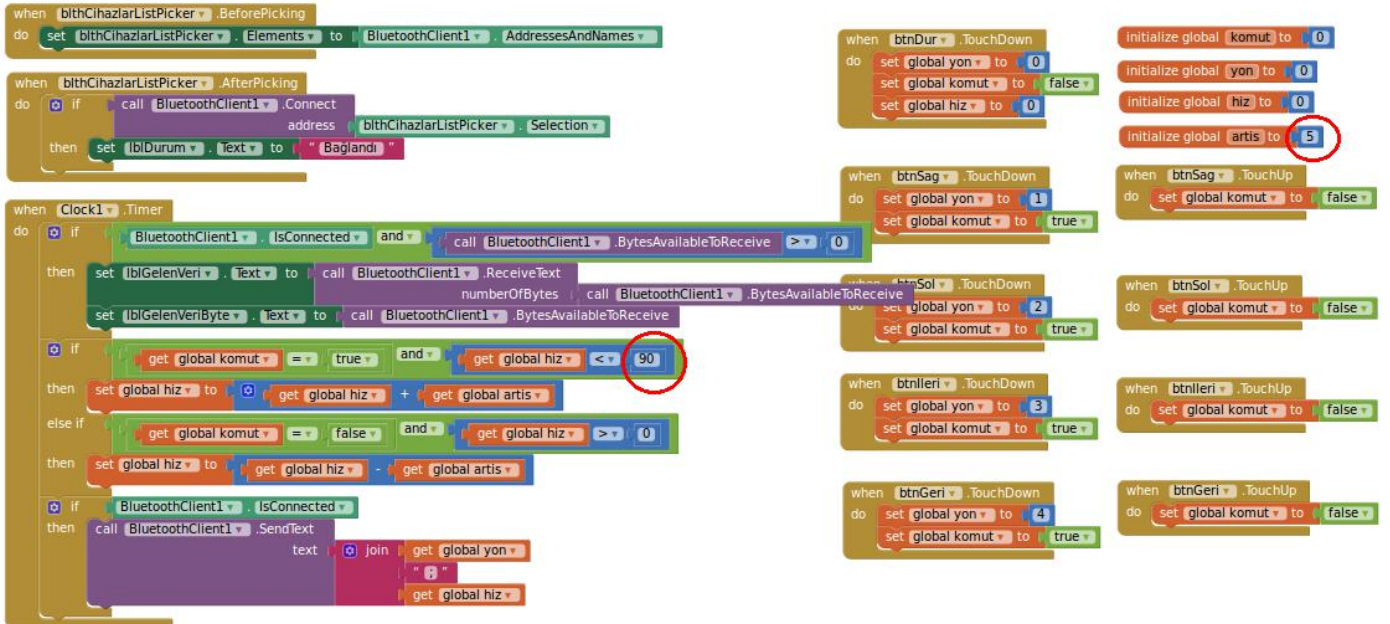
Android tarafında ise Appinventor kodunda çok ufak 2 değer değişikliği yapılmıştır. Bluetooth kontrollü arabada analogWrite() için 0-255 arası hız değeri gönderilirken Juno'da 0-90 arası hız değeri gönderilmektedir. Ayrıca hız artışı 15'ten 5'e düşürülmüştür. Bunun haricinde Appinventor Android

uygulama bileşenleri ve Android uygulamasının çalışması [“Bluetooth Kontrollü Araba Uygulaması – 2. Kısım”](#) ile tamamen aynıdır. İlgili yazıdan Android uygulamasının çalışması incelenebilir.

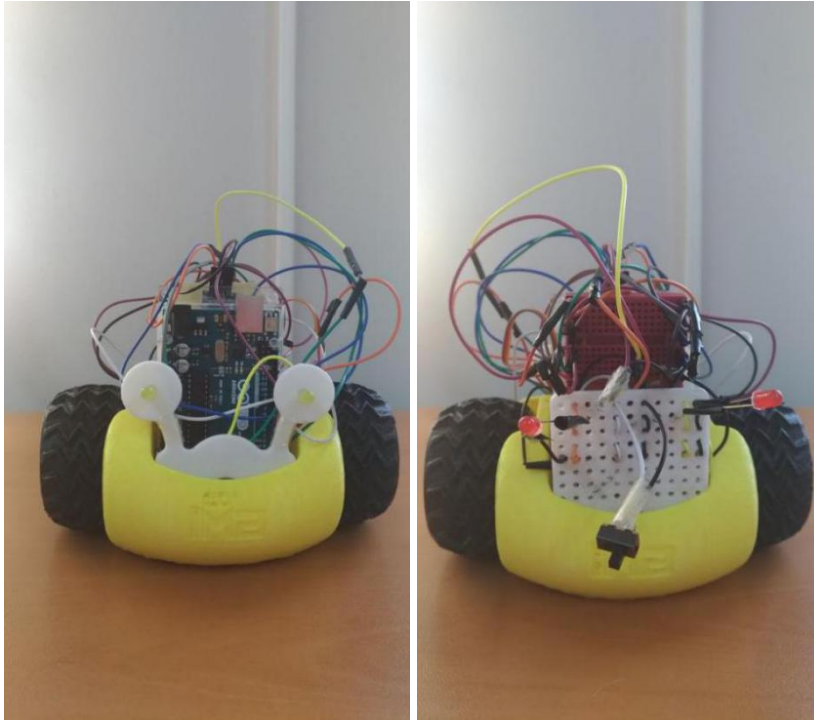
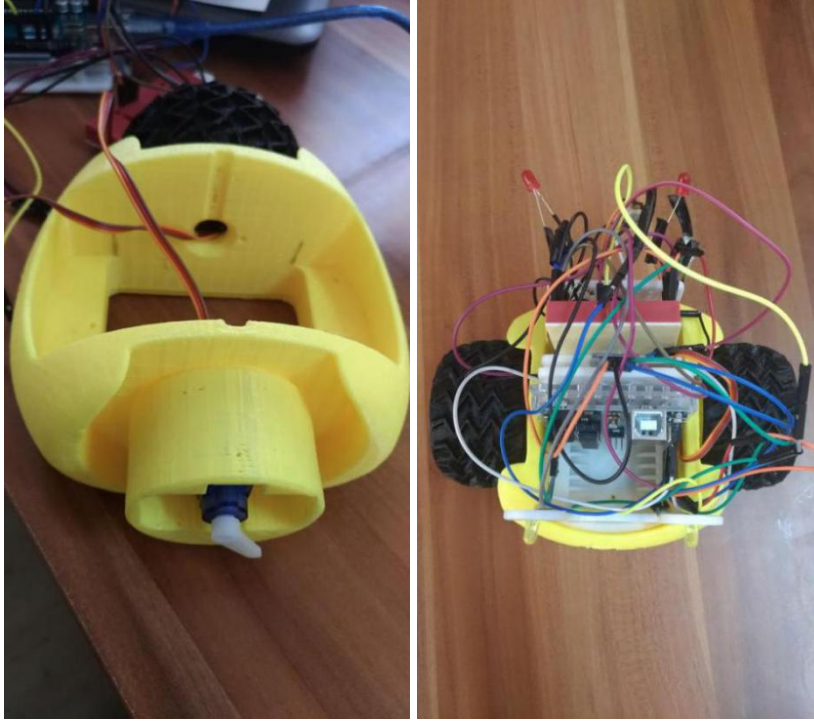
Uygulama arayüzü ve bileşenleri şöyledir:



Appinventor kod blokları aşağıdaki gibidir (değişiklikler kırmızı daire ile gösterilmiştir):



Uygulama görüntüleri:



Uygulama videosu: [Uygulama videosu için buraya tıklayınız.](#)

APK dosyası linki: [Dosyayı indirmek için buraya tıklayınız.](#)

Kaynakça

- <https://www.exploremaking.com>
- <https://www.thingiverse.com/thing:1720394>
- <https://github.com/exploremaking/Juno/blob/master/Juno.ino>
- <https://www.arduino.cc/en/Reference/Servo>
- <https://www.arduino.cc/en/Reference/ServoWrite>
- <http://ctc-dev.verkstad.cc/en/course-literature/continuous-rotation-servo/>
- <https://core-electronics.com.au/continuous-rotation-micro-servo-fs90r.html>
- https://media.digikey.com/pdf/Data%20Sheets/Adafruit%20PDFs/2442_Web.pdf