

Arduino ile LDR kontrolü RGB led.

28. 1. led (red) // 2. led (green) // 3. led (blue)

4. led (yellow) // 5. led (cyan) // 6. led (magenta)

7. led (white) // 8. led (black) // 9. led (grey)

10. led (brown) // 11. led (pink) // 12. led (purple)

13. led (orange) // 14. led (olive) // 15. led (teal)

16. led (light blue) // 17. led (light green) // 18. led (light yellow)

19. led (light pink) // 20. led (light purple) // 21. led (light brown)

22. led (light orange) // 23. led (light olive) // 24. led (light teal)

25. led (light light blue) // 26. led (light light green) // 27. led (light light yellow)

28. led (light light pink) // 29. led (light light purple) // 30. led (light light brown)

31. led (light light orange) // 32. led (light light olive) // 33. led (light light teal)

34. led (light light light blue) // 35. led (light light light green) // 36. led (light light light yellow)

37. led (light light light pink) // 38. led (light light light purple) // 39. led (light light light brown)

40. led (light light light orange) // 41. led (light light light olive) // 42. led (light light light teal)

43. led (light light light light blue) // 44. led (light light light light green) // 45. led (light light light light yellow)

46. led (light light light light pink) // 47. led (light light light light purple) // 48. led (light light light light brown)

49. led (light light light light orange) // 50. led (light light light light olive) // 51. led (light light light light teal)

52. led (light light light light light blue) // 53. led (light light light light light green) // 54. led (light light light light light yellow)

55. led (light light light light light pink) // 56. led (light light light light light purple) // 57. led (light light light light light brown)

58. led (light light light light light orange) // 59. led (light light light light light olive) // 60. led (light light light light light teal)

61. led (light light light light light light blue) // 62. led (light light light light light light green) // 63. led (light light light light light light yellow)

64. led (light light light light light light pink) // 65. led (light light light light light light purple) // 66. led (light light light light light light brown)

67. led (light light light light light light orange) // 68. led (light light light light light light olive) // 69. led (light light light light light light teal)

70. led (light light light light light light light blue) // 71. led (light light light light light light light green) // 72. led (light light light light light light light yellow)

73. led (light light light light light light light pink) // 74. led (light light light light light light light purple) // 75. led (light light light light light light light brown)

76. led (light light light light light light light orange) // 77. led (light light light light light light light olive) // 78. led (light light light light light light light teal)

79. led (light light light light light light light light blue) // 80. led (light light light light light light light light green) // 81. led (light light light light light light light light yellow)

82. led (light light light light light light light light pink) // 83. led (light light light light light light light light purple) // 84. led (light light light light light light light light brown)

85. led (light light light light light light light light orange) // 86. led (light light light light light light light light olive) // 87. led (light light light light light light light light teal)

88. led (light light light light light light light light light blue) // 89. led (light light light light light light light light light green) // 90. led (light light light light light light light light light yellow)

91. led (light light light light light light light light light pink) // 92. led (light light light light light light light light light purple) // 93. led (light light light light light light light light light brown)

94. led (light light light light light light light light light orange) // 95. led (light light light light light light light light light olive) // 96. led (light light light light light light light light light teal)

97. led (light light light light light light light light light light blue) // 98. led (light light light light light light light light light light green) // 99. led (light light light light light light light light light light yellow)

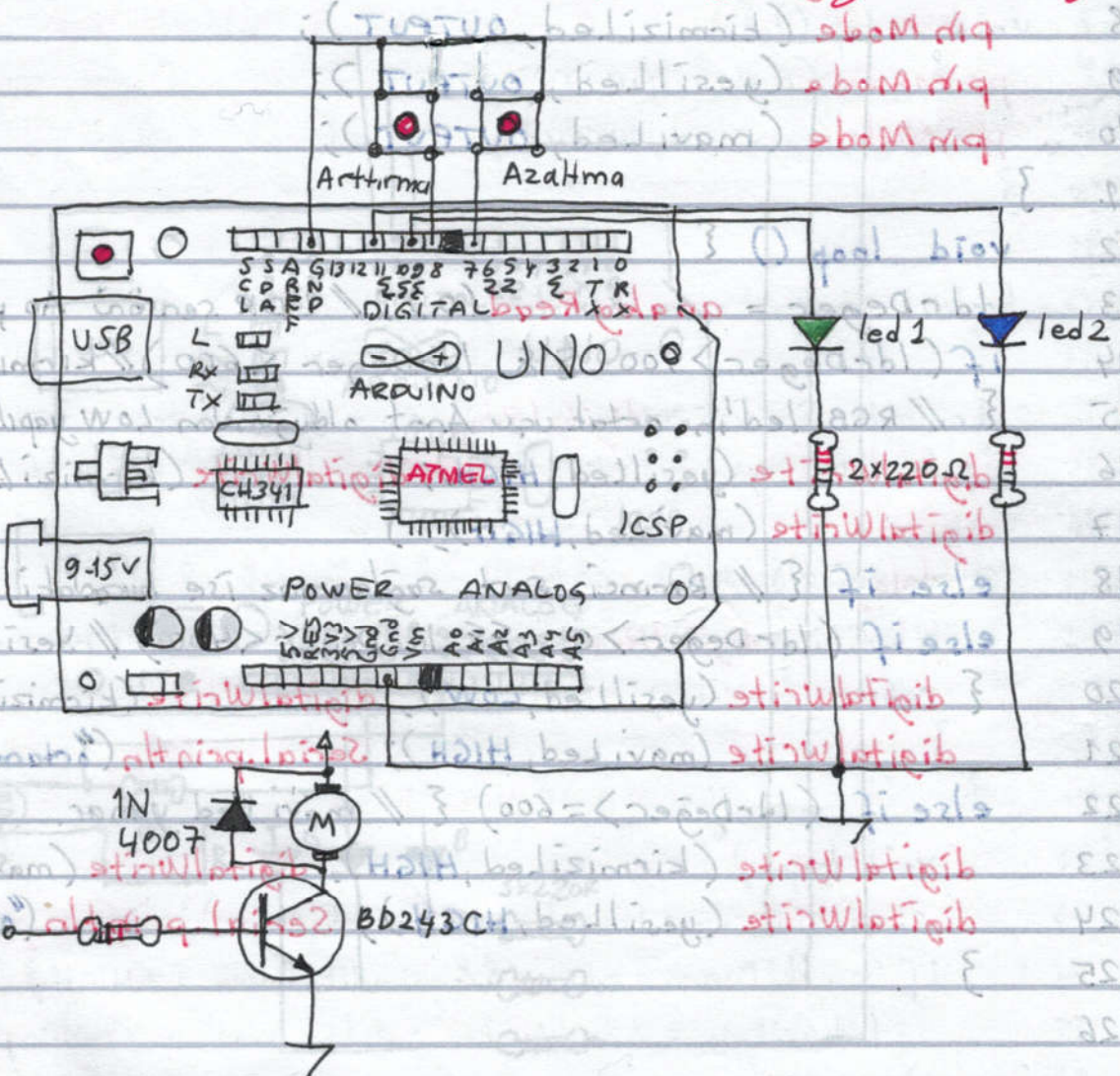
100. led (light light light light light light light light light light pink) // 101. led (light light light light light light light light light light purple) // 102. led (light light light light light light light light light light brown)

103. led (light light light light light light light light light light orange) // 104. led (light light light light light light light light light light olive) // 105. led (light light light light light light light light light light teal)

106. led (light light light light light light light light light light light blue) // 107. led (light light light light light light light light light light light green) // 108. led (light light light light light light light light light light light yellow)

109. led (light light light light light light light light light light light pink) // 110. led (light light light light light light light light light light light purple) // 111. led (light light light light light light light light light light light brown)

Arduino Butonlar ile Led parlaklığını ayarlama bağlantısı.



Arduino Butonlar ile Led parlaklığı ayarlama

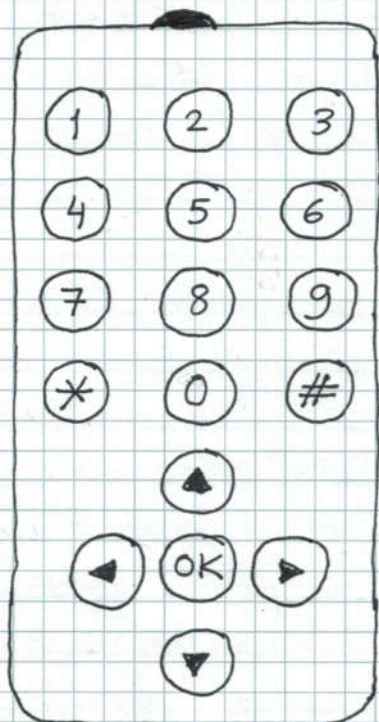
BUTON-LEDDIMMER

Bu programda brightness adlı integer değer (32768 farklı değer) alabilen değerlerle 7. pmine bağlı butona basılınca azalacak ve 8. pmine bağlı butona basılınca arttırılacaktır. Bu değer aralığı ise if komutu yardımıyla 0 ile 255 arasında tutularak analogWrite komutu ile 11. ve 9. pinlere (pwm n) bağlı olan ledlerin parlaklığı değiştirilmiştir.

```
1. int brightness = 127; //brightness parlaklık değeri 127 olur
2.
3. void setup() {
4.   Serial.begin(9600); // Serial iletişimi 9600 hızında başlat.
5.   pinMode(7, INPUT_PULLUP); // azaltma butonu
6.   pinMode(8, INPUT_PULLUP); // arttırma butonu
7.   pinMode(9, OUTPUT); // led1 9. pwm pinine bağlı
8.   pinMode(11, OUTPUT); // led2 11. pwm pinine bağlı
9. }
10.
11. void loop() {
12.   Serial.print("led parlaklığı = ");
13.   Serial.println(brightness); // parlaklık değerini yazdırır.
14.   if(digitalRead(7) == LOW) brightness--;
15.   if(digitalRead(8) == LOW) brightness++;
16.   if(brightness < 0) brightness = 0;
17.   if(brightness > 255) brightness = 255;
18.
19.   analogWrite(9, brightness); // led1'in parlaklığını ayarlar.
20.   analogWrite(11, brightness); // led2'nin parlaklığını ayarlar.
21.   delay(10); // 10 ms bekleme yap
22. }
```

Aynı bağlantıda örneğin Led2 yerine D11 pinine bir transistör (NPN) üzerinden transistörün anahtarlama elemanı olarak kullanılmasına ait bağlantı üzerinden DC bir motor bağlanırsa ilk anda motor yarı hızında dönecektir. Azalt butonuna basılınca motorun pwm hız değeri 127'den 0'a doğru azalarak motor yavaşlayıp duracaktır. Azalt yerine Arttır ile hızı artacaktır.

NEC - 32 bits



0 - 0xFF9867 - Far aç

1 - 0xFFA25D - Far kapa

2 - 0x FF629D

3 - 0x FFE21D

4- 0x FF22DD

5 - 0x FF02FD

6 - 0x FFC23D

7 - 0 x FFE01F

8 - 0 x FFA857

9 - 0x FF906F

UP-0xFF18E7 $\div$ ileri komutu

OK.DVR-0xFF38C7 - dvr //

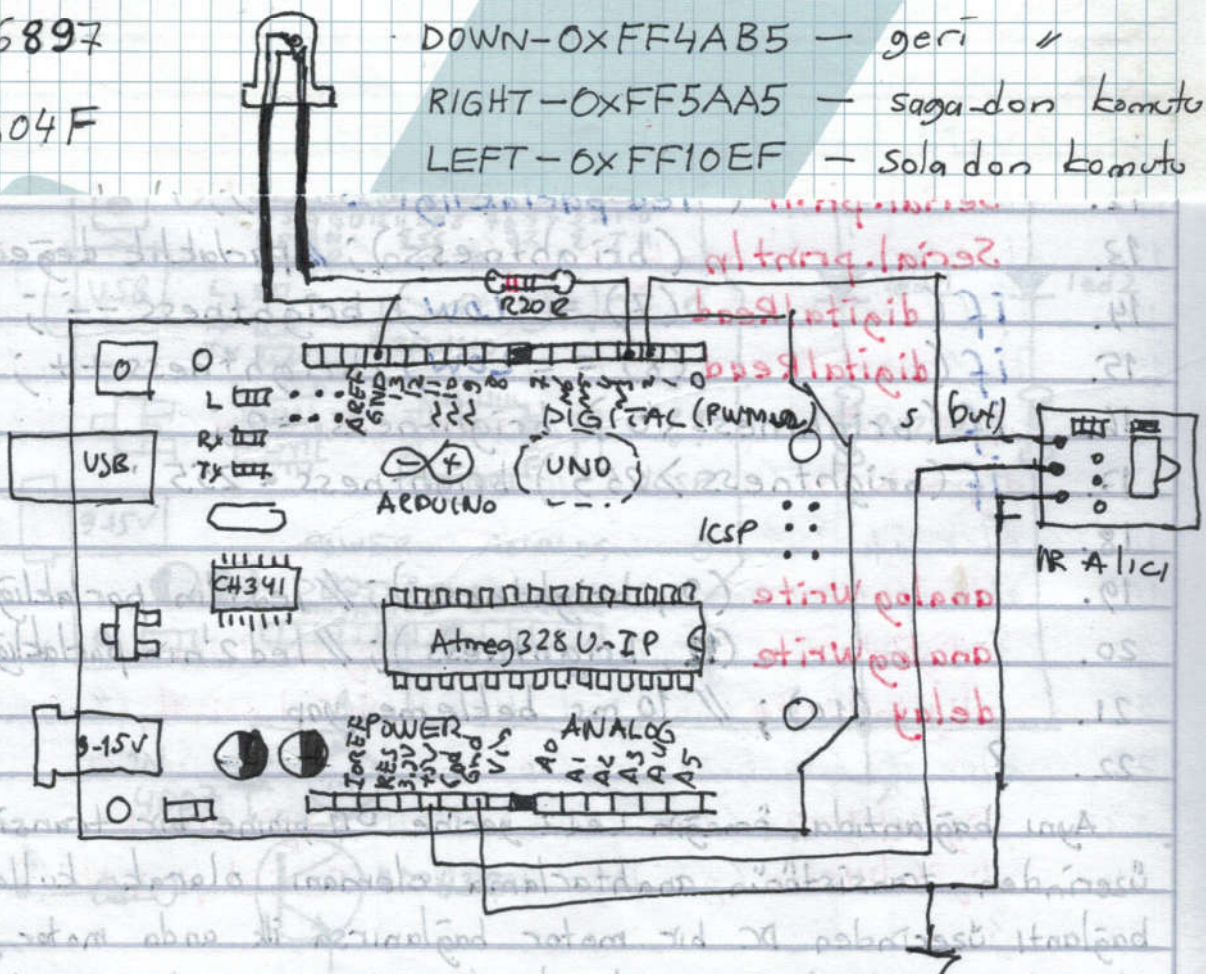
DOWN-0xFF4AB5 - geri //

RIGHT-0xFF5AA5 - saga-don kemutu

LEFT-0xFF10EF - soldan komutu

* -0xFF 6897

- 0xFFB04F



Arduino IR alıcı (TSOP1838) kullanımı.

IR_LED.ino

Arduino IDE içinde bulunan IRremote kütüphanesi kullanılarak Infrared (Kızılötesi) kumanda sinyalleri kullanılarak istenilen bir görev komutlar yardımıyla yaptırılabilir. Örneğin alıcı üzerine gelen bir sinyal olduğunda Arduino çıkışlarından birisine bağlanmış olan Role devresi çalıştırılarak istenilen bir lamba kumanda ile çalışabilir. Bu örnekte IR alıcıya gelen sinyal ile bir LED diyot kumanda edilmiştir.

```
1 #define BUTON0 0xFF9867 // 0. Tuş
2 #define BUTON1 0xFFA25D // 1. Tuş
3 #include <IRremote.h> // İnc. dosyasını ekliyoruz.
4 int RECV_PIN = 2; // IR alıcı 2. pine bağlı
5 int ledim = 3; // Led 3. pine bağlı
6 IRrecv irrecv(RECV_PIN); // IR dönüştürme pinini RECV_PIN
7 decode_results results;
8
9 void setup() {
10   Serial.begin(9600); // Seri iletişimi 9600 Baud hızında başlat.
11   irrecv.enableIRIn(); // IR ledi çalıştır.
12   pinMode(RECV_PIN, INPUT); // RECV_PIN girişi olarak ayarlanır.
13   pinMode(ledim, OUTPUT); // İlk başta Led sönmek
14   digitalWrite(ledim, LOW);
15 }
16
17 void loop() {
18   if (irrecv.decode(&results)) {
19     Serial.println(results.value, HEX); // Gelen bilgiyi HEX'e dönüştür.
20     if (results.value == BUTON1) {
21       Serial.println("1. tusa basıldı.");
22       if (digitalRead(ledim) == LOW) (digitalWrite(ledim, HIGH)); }
23
24     if (results.value == BUTON0) {
25       Serial.println("0. tusa basıldı.");
26       if (digitalRead(ledim) == HIGH) (digitalWrite(ledim, LOW)); }
27     irrecv.resume(); }
28 }
```


Arduino IRremote kütüphanesi alıcı dışında verici olarak kullanmak içinde programa eklenebilir. Aşağıdaki program iki butonu giriş ve bir IR-led diyodu (Infrared diyot) ile birlikte Arduino'yu bir kumanda vericisi gibi çalıştırır. Programdaki for komutu aynı kumanda kodunu en az 3 defa göndererek alıcının algılaması ihtimalini artırır.

```

1 #include <IRremote.h>
2 IRsend irsend; // irsend vericisi tanımlandı.
3 int ac = 5; // ac butonu 5. pine bağlı.
4 int kapa = 4; // kapa butonu 4. pine bağlı.
5 int IRled = 3; // IR verici 3. pine bağlı.
6 void setup() {
7   Serial.begin(9600); // Serial iletişimi 9600 Baud hızında başlat.
8   pinMode(ac, INPUT_PULLUP); // giriş yapıldı. (ac butonu)
9   pinMode(kapa, INPUT_PULLUP); // giriş yapıldı. (kapa butonu)
10  pinMode(IRled, OUTPUT); // IR ledi çıkış yapıldı.
11  void loop() {
12    if (digitalRead(ac) == LOW) // Eğer ac butonuna basılırsa
13    {
14      for (int i=0; i<3; i++) { // 3 defa arka arkaya
15        irsend.sendNEC(0xFFA25D, 32); // NEC kodu gönderir.
16        Serial.println("Led yandı");
17        delay(40);
18      }
19      if (digitalRead(kapa) == LOW) // Eğer kapa butonuna basılırsa
20      {
21        for (int i=0; i<3; i++) { // 3 defa arka arkaya
22          irsend.sendNEC(0xFF9867, 32); // NEC kodu gönderir.
23          Serial.println("Led söndü");
24          delay(40);
25        }
26        delay(300); // Buton sıçraması için bekleme yap.
27      }

```

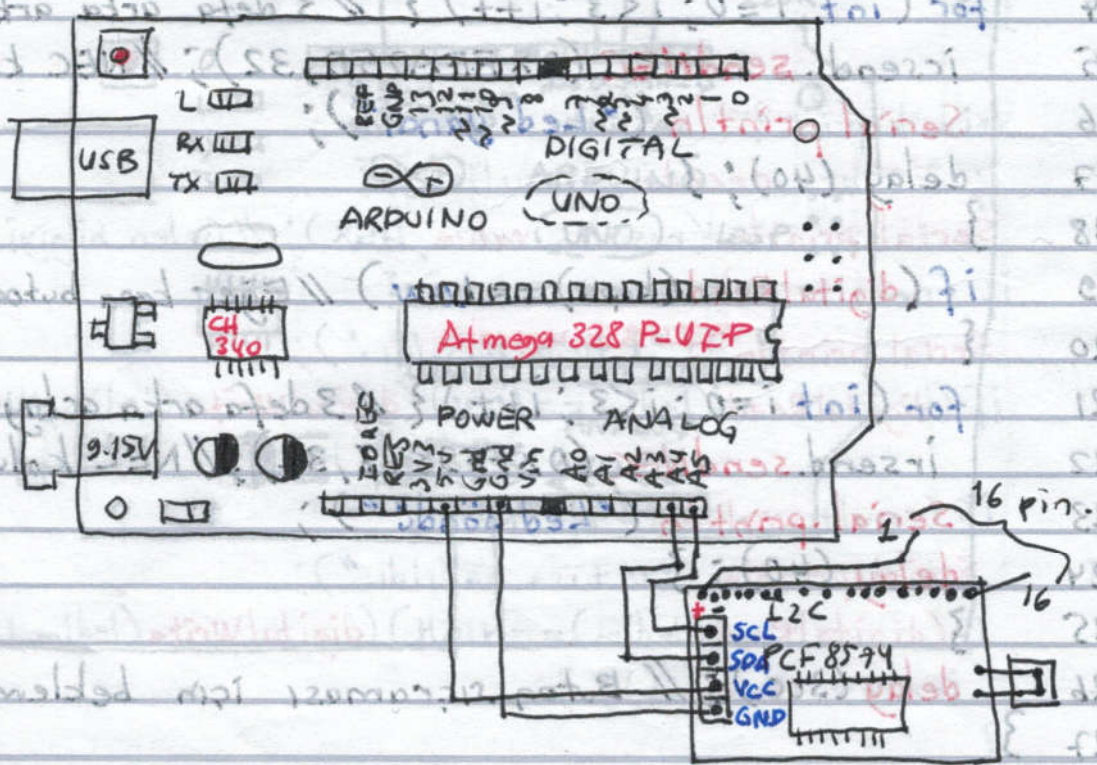
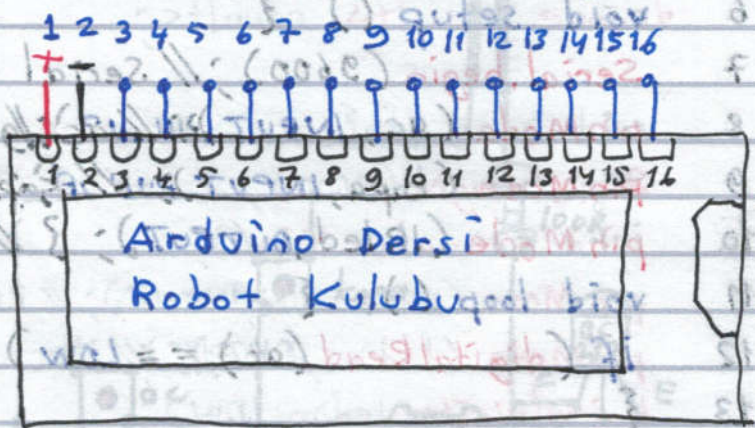
Arduino IR-Verstärker ist ein Projekt, das die Funktion eines IR-Verstärkers nachbaut. Es besteht aus einem Arduino Uno, einem L298N-Motorcontroller und einem IR-Verstärker-Modul. Das Projekt ist für die Steuerung von Motoren und anderen elektrischen Geräten geeignet. Die Schaltung ist einfach zu realisieren und kann mit verschiedenen Sensoren und Aktoren erweitert werden.

Arduino ile 2x16 LCD ekran kullanımı başlangıcı

İzci PCF8574 pinleri

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



Arduino ile LCD ekran kullanımı (I2C) LCD-Serial.ino

Arduino ile I2C (Inter-Integrated Circuit) Seri haberleşme metodu kullanımını öğrenerek sadece 4 tane kablo ile LCD ekrana yazı yazdırılabilir. Bu pinlerden ikisi Vcc ve Gnd diğer ikisi SCL (Serial Clock) ve SDA (Serial Data) pinleridir. PCF8574 entegresi I2C dönüştürücü entegresi iki hattan haberleşmeyi mümkün kılar. Aksi durumda LCD'ye yazı yazdırmak için en az 8 pinin Arduino'ya bağlanması gerekir. "LiquidCrystal_I2C" kütüphanesinin indirilmesi ve yüklenmesi gerekir. Bu örnekte 2x16 karakter LCD'ye yazı yazmak için SDA pininin A4 pinine ve SCL pininin de A5 pinine bağlı olması gerekir.

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd (0x27, 16, 2); // 2 satır 16
4 // karakter LCD için 0x27 adresi belirlendi. 0x3F'de olabilir.
5 void setup() {
6   Serial.begin(); // Seri iletişimi 9600 Baud hızında başlat.
7   lcd.begin(); // LCD başlangıç ayarlarını yapar.
8   lcd.backlight(); // LCD arka plan ışığını yak.
9   lcd.setCursor(0, 0); // kursoru 0. satır 0. sütuna götür.
10  lcd.print("Arduino Dersi"); // LCD'ye Arduino Dersi yazar.
11  lcd.setCursor(0, 1); // kursoru 0. sütun ve 1. satıra götür.
12  lcd.print("Robot Kulubu"); // LCD'ye Robot Kulubu yazar.
13  void loop() {
14    if (Serial.available()) { // Seri ekrandan gelen veri varsa
15      delay(100); // 100 msn bekle
16      lcd.clear(); // Ekranı temizle.
17      while (Serial.available() > 0) { // Gelen seri veri varsa
18        lcd.write(Serial.read()); // Seri bilgiyi lcd'ye yaz.
19      }
20    }
21  }
```

NOT: Ekranda yazı görülmese adres değiştirilmelidir. 0x27 yerine 0x3F yazılarak yeniden yüklenmelidir. Yazının kontrast ayarı I2C dönüştürücü üstündeki POT ayarlanarak değiştirilebilir.

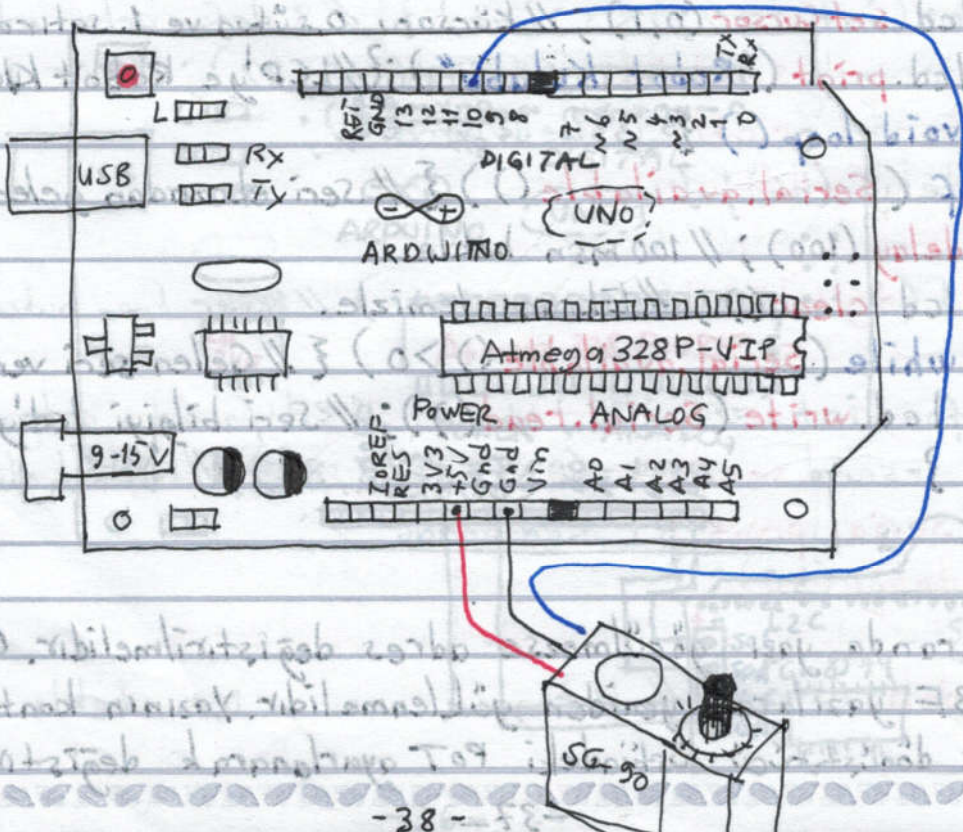
Arduino ile LCD ekran kullanımı (I2C)

Arduino ile I2C (inter integrated circuit) serisi haberleşme metodu kullanılarak öğretilen bir yöntemdir. Bu yöntemde bir VCC ve GND diğer iki pin (Serial Clock) ve SDA (Serial Data) pinleri. PC8234 entegresi I2C dönüştürücü entegresi ile haberleşme yapar. Bu entegre Arduino ile haberleşme için en az 8 pinin Arduino'ya bağlanması gerekir. Bu entegre 5V ile çalışır. SDA pinine 5V ve SCL pinine GND bağlanmalıdır. Ayrıca 5V ve GND pinlerine de 5V ve GND bağlanmalıdır.

```
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3 LiquidCrystal_I2C lcd(0x27, 16, 5); // 5 satır 16
4 // karakter LCD için 0x27 adresi belirtilmelidir. 0x27 de olabilir.
```

Arduino ile servo motor kontrolü bağlantısı

```
5 #include <Servo.h>
6 Servo servo;
7 // Servo motorun 5V ve GND pinlerine 5V ve GND bağlanmalıdır.
8 // Servo motorun SCL pinine 5V ve SDA pinine GND bağlanmalıdır.
9 // Servo motorun 0.1s ile çalıştırılmalıdır.
10 // Servo motorun 0.1s ile çalıştırılmalıdır.
11 // Servo motorun 0.1s ile çalıştırılmalıdır.
```



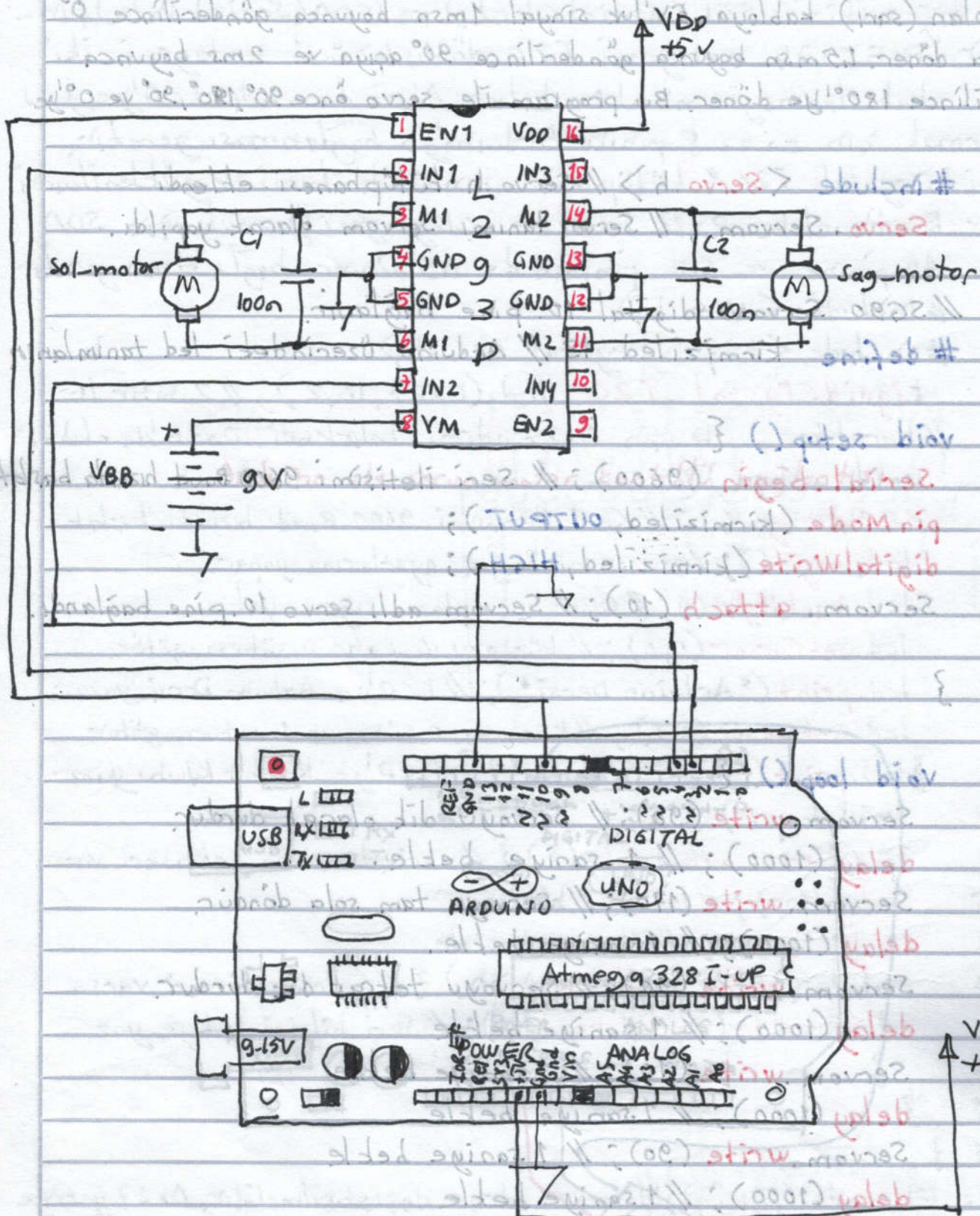
Arduino ile Servo Motor Kontrolü.

Servo_donus.ino

Servo motorlar verilen komutlar yardımıyla ve hazır olarak Arduino IDE si ile verilen Servo.h kütüphanesiyle çalışan açısal olarak (0° , 90° , 180°) olarak dönebilen motorlardır. Örneğin pals pini olan (sarı) kabloya 5V'lık sinyal 1msn boyunca gönderilince 0° açısına döner. 1.5msn boyunca gönderilince 90° açığa ve 2msn boyunca gönderilince 180° 'ye döner. Bu program ile servo önce 90° , 180° , 90° ve 0° 'ye döner.

```
1  #include <Servo.h> // Servo.h kütüphanesi eklendi
2  Servo Servom; // Servo tanımı Servom olarak yapıldı.
3
4  // SG90 Servosu digital 10. pine bağlanır.
5  #define kırmızıled 13 // Arduino üzerindeki led tanımlanır
6
7  void setup() {
8    Serial.begin(9600); // Seri iletişimi 9600 Baud hızında başlat.
9    pinMode(kırmızıled, OUTPUT); // Sol motor EN çıkışı
10   digitalWrite(kırmızıled, HIGH);
11   Servom.attach(10); // Servom adlı servo 10. pine bağlandı.
12   // digitalWrite(sol motor EN 12); // Sol motor 1/2 g'ne ileri
13 }
14
15 void loop() {
16   Servom.write(90); // Servoyu dik olarak durdur.
17   delay(1000); // 1 saniye bekle.
18   Servom.write(170); // Servoyu tam sola döndür.
19   delay(1000); // 1 saniye bekle.
20   Servom.write(90); // Servoyu tekrar dik durdur.
21   delay(1000); // 1 saniye bekle
22   Servom.write(10); // 1 saniye bekle
23   delay(1000); // 1 saniye bekle
24   Servom.write(90); // 1 saniye bekle
25   delay(1000); // 1 saniye bekle
26   Serial.println("Servo 90 derece. Galışıyor.");
27 }
```

Arduino Uno ile L293D Motor sürücü bağlantısı.

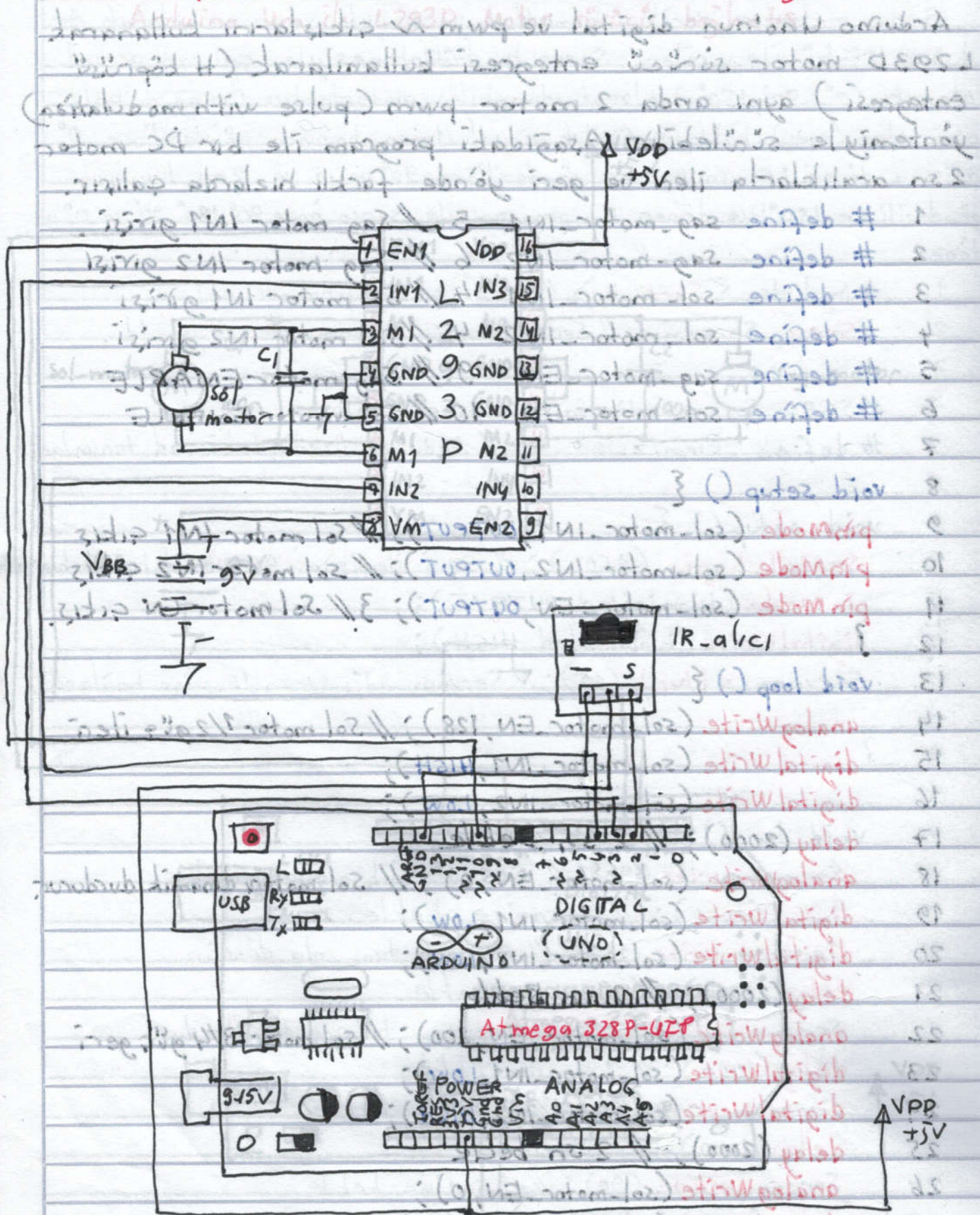


Arduıno Uno ile L293D Motor sürücü L293D-TEST1

Arduıno Uno'nun dıgital ve pwm v çıkışlarını kullanarak L293D motor sürücü entegresi kullanılarak (H köprüsü entegresi) aynı anda 2 motor pwm (pulse with modulation) yöntemıle sürülebilir. Aşağıdaki program ıle bir DC motor 2 sn aralıklarla ılerı ve gerı yönde farklı hızlarda ęalırır.

```
1 #define sag-motor-IN1 5 // Sag motor IN1 gırıřı
2 #define sag-motor-IN2 6 // Sag motor IN2 gırıřı
3 #define sol-motor-IN1 7 // Sol motor IN1 gırıřı
4 #define sol-motor-IN2 4 // Sol motor IN2 gırıřı
5 #define sag-motor-EN 9 // Sag motor ENABLE
6 #define sol-motor-EN 10 // Sol motor ENABLE
7
8 void setup() {
9   pinMode(sol-motor-IN1, OUTPUT); // Sol motor IN1 çıkış
10  pinMode(sol-motor-IN2, OUTPUT); // Sol motor IN2 çıkış
11  pinMode(sol-motor-EN, OUTPUT); // Sol motor EN çıkış
12 }
13 void loop() {
14   analogWrite(sol-motor-EN, 128); // Sol motor 1/2 güę ılerı
15   digitalWrite(sol-motor-IN1, HIGH);
16   digitalWrite(sol-motor-IN2, LOW);
17   delay(2000); // 2 sn bekle
18   analogWrite(sol-motor-EN, 0); // Sol motor dınamik durdurur
19   digitalWrite(sol-motor-IN1, LOW);
20   digitalWrite(sol-motor-IN2, LOW);
21   delay(2000); // 2 sn bekle
22   analogWrite(sol-motor-EN, 200); // Sol motor 3/4 güę gerı
23   digitalWrite(sol-motor-IN1, LOW);
24   digitalWrite(sol-motor-IN2, HIGH);
25   delay(2000); // 2 sn bekle
26   analogWrite(sol-motor-EN, 0);
27   digitalWrite(sol-motor-IN1, LOW);
28   digitalWrite(sol-motor-IN2, LOW);
29   delay(2000); // 2 sn bekle
```

Arduino Uno ile IR kodlu Motor kontrolü bağlantısı.



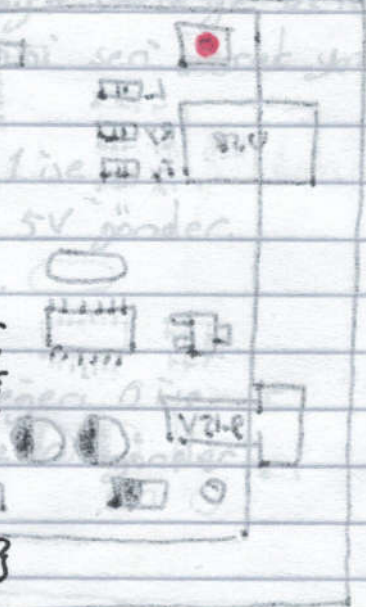
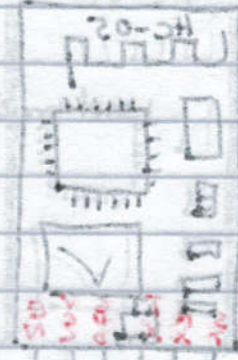
Arduino Uno ile IR kodlu Motor Kontrolü IR_L293_TEST.mo

Arduino Uno R3 kullanılarak IR alıcı modül yardımıyla alınan IR kodu denetlenerek, bu kodların değişimine göre L293D motor sürücüsü kontrol edilebilir. Böylece istenilen motor çalıştırılıp durdurulabilir veya yönü ve hızı kontrol edilebilir. Bu örnekte bir motor (DC) çalıştırılıp durdurulmuştur.

```

1 #include <IRremote.h>
2 #define sol-motor-IN1 3
3 #define sol-motor-IN2 4
4 #define sol-motor-EN 10
5 #define kirmiziled 13
6 #define BUTON0 0xFF9867
7 #define BUTON1 0xFFA25D
8 int RECV_PIN=2;
9 IRrecv irrecv(RECV_PIN);
10 decode_results results;
11 void setup() { Serial.begin(9600);
12   irrecv.enableIRIn();
13   pinMode(sol-motor-IN1, OUTPUT);
14   pinMode(sol-motor-IN2, OUTPUT);
15   pinMode(sol-motor-EN, OUTPUT);
16   pinMode(kirmiziled, OUTPUT);
17   digitalWrite(kirmiziled, HIGH);
18   void loop() {
19     if(irrecv.decode(&results)) {
20       Serial.println(results.value, HEX);
21       if (results.value == BUTON1) {
22         Serial.println("1.TUS MOTOR ILER");
23         if (digitalRead(kirmiziled) == HIGH) {
24           analogWrite(sol-motor-EN, 128);
25           digitalWrite(sol-motor-IN1, HIGH);
26           digitalWrite(sol-motor-IN2, LOW);
27         }
28         if (irrecv.decode(&results)) {
29           Serial.println(results.value, HEX);

```

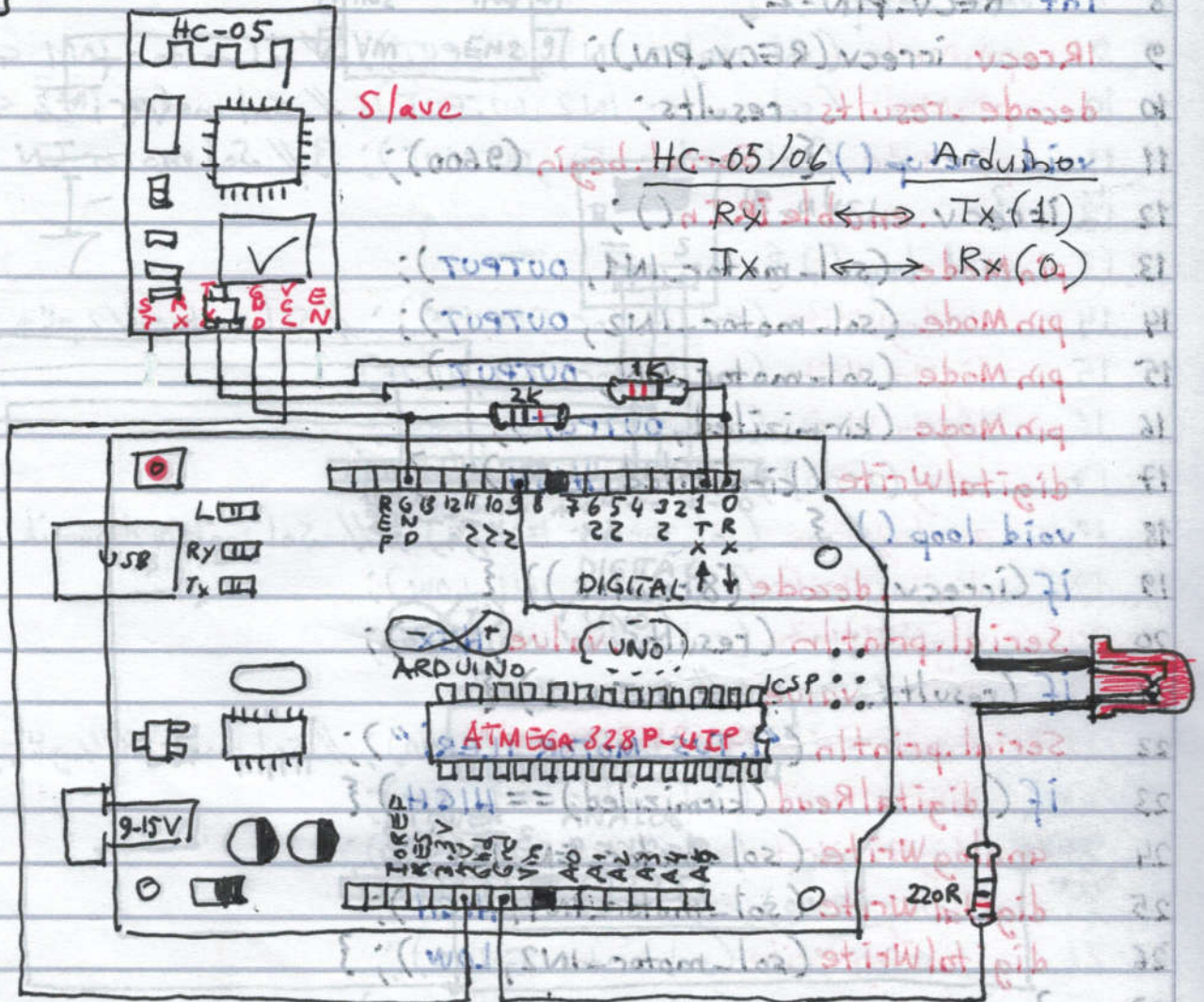


Arduino ve HC-05 Bluetooth modül led kontrolü bağlantısı.

```

30. if (results.value == BUTONO) {
31.   Serial.println("0.TUS MOTOR DURDU");
32.   if (digitalRead(kirmiziled) == HIGH) {
33.     analogWrite(sol-motor-EN,0);
34.     digitalWrite(sol-motor-IN1,LOW);
35.     digitalWrite(sol-motor-IN2,LOW); }
36. }
37. irrecv.resume();
38. }
39. }
40. }

```



Arduino ve HC-05 Bluetooth modül led kontrolü digitalReceiver.ino

Bluetooth, farklı cihazların birbirleriyle haberleşmesinde kullanılan kablosuz 2.4 GHz hızında iletişim sağlayan bir yöntemdir. Bluetooth, radyo frekans standartlarını kullanarak iletişim sağlar. Arduino ile HC-05 Bluetooth-seri port dönüştürücü modülü kullanarak hem master hem de slave aygıt olarak çalışan haberleşme sistemi yapılabilir. HC-06 ise sadece slave aygıt olarak kullanılabilir. Aşağıdaki kod ile Arduino'nun 9. pinine bağlı led bluetooth seri dönüştürücü ile kontrol edildi.

Arduino ve HC-05 Bluetooth modülü 5V ve GND'ye bağlanmalıdır.

```
1. #define ledPin 9 // Led pin numarası
2. int state = 0; // led'in durumu
3.
4. void setup() {
5.   pinMode(ledPin, OUTPUT);
6.   digitalWrite(ledPin, HIGH);
7.   Serial.begin(38400); // Varsayılan iletişim hızı
8.   delay(1000);
9. }
10.
11. void loop() {
12.   if (Serial.available() > 0) { // Eğer seri porttan data geliyorsa
13.     state = Serial.read(); // gelen datayı state'e aktar.
14.     Serial.write(state); // state değerini seri olarak yazdır.
15.
16.     if (state == '1') { // state değeri 1 ise
17.       digitalWrite(ledPin, HIGH); // ledPin'e 5V gönder.
18.       state = 0; // state değerini 0 yap.
19.     }
20.     if (state == '0') { // Eğer state değeri 0 ise
21.       digitalWrite(ledPin, LOW); // ledPin'e 0V gönder.
22.       state = 1; // state değerini 1 yap.
23.     }
24.   }
25. }
```

Arb. Nr. 12 H.C-02 Blutgefäßmodell bei Kontroll- digital/Analoge

Arduino ve HC-05 Bluetooth modülü ile kontrolü bağlantısı:

[illegible]

33. Arduino ve HC-05'in Master olarak kullanımı bağlantısı.

34. digitalWrite (sol_motor - INT, LOW); // digital output # 1

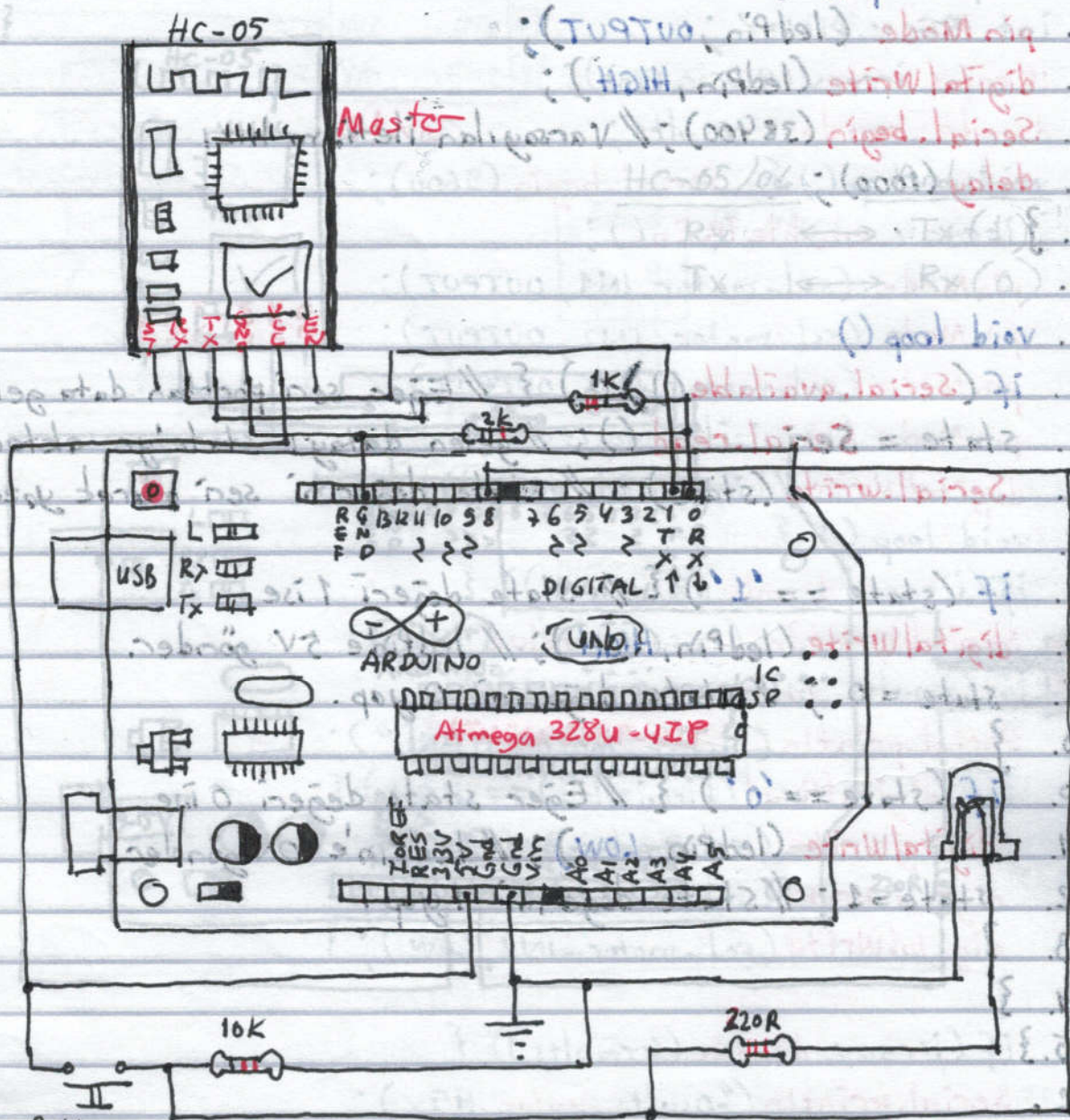
35. digital/Weite (cool-motoren) $\frac{1}{10}$: $a = \text{statz} \cdot \text{trf} \cdot s$

```
37 irrecv.resume();
```

HC-05

39.  (Hoch, Tief) (Hoch, Tief)

Master



ARDUINO ve HC-05'in Master olarak kullanımı digital-TX.ino

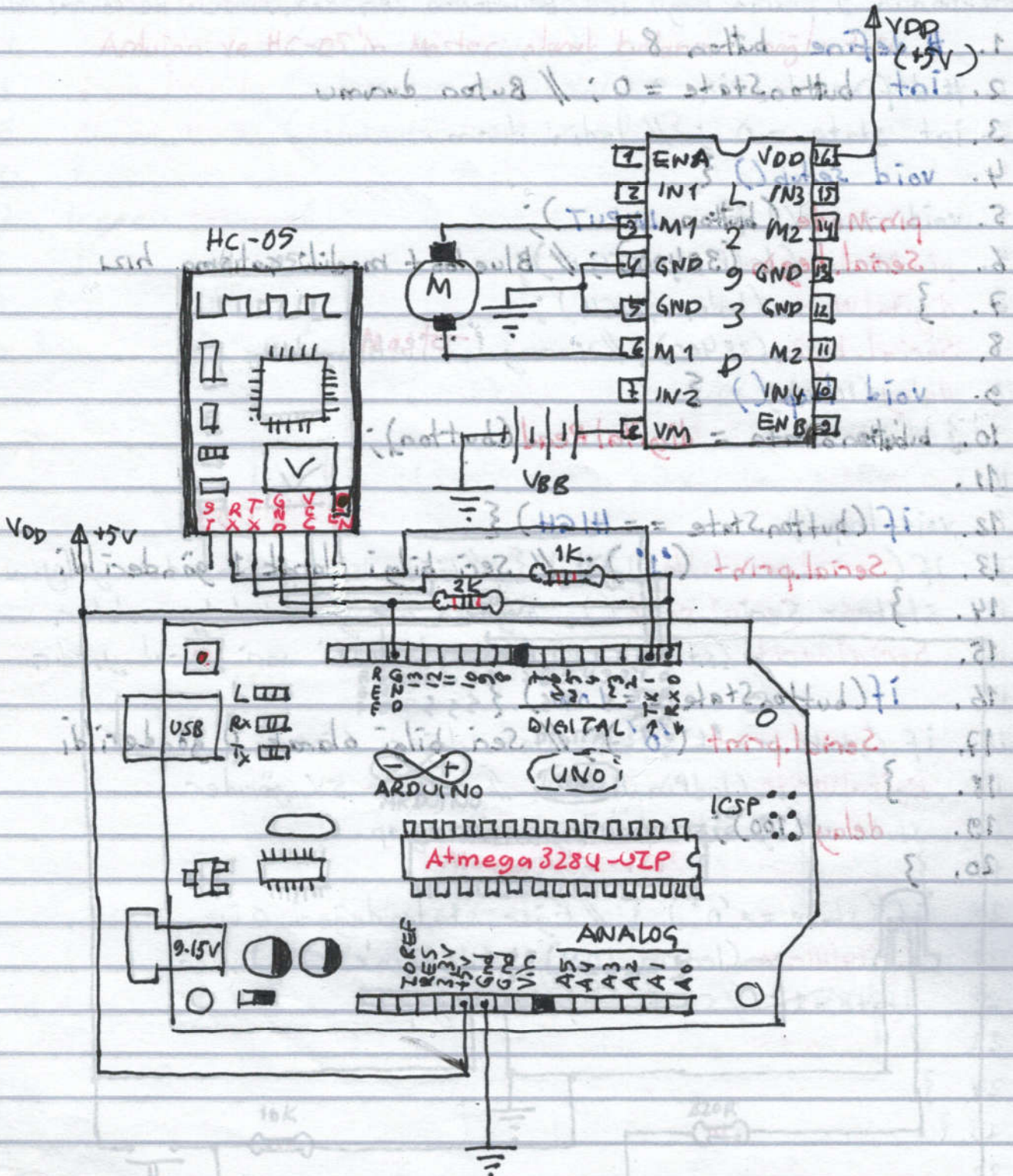
Arduino ve HC05 Bluetooth modül'ün bu sefer verici olarak kullanılması sağlanmıştır. Bunu sağlamak için önce HC05 AT komutları kullanılarak ismi değiştirilmiştir. Verici için Master tabiri kullanılmıştır. Böylece cep telefonu veya Bilgisayar ile Bluetooth üzerinden bağlantı yapılırsa ismi Master olarak seçilebilir. Karışıklığın önüne geçilir.

```
1. #define button 8
2. int buttonState = 0; // Buton durumu
3.
4. void setup() {
5.   pinMode(button, INPUT);
6.   Serial.begin(38400); // Bluetooth modülü çalışma hızı
7. }
8.
9. void loop() {
10.  buttonState = digitalRead(button);
11.
12.  if (buttonState == HIGH) {
13.    Serial.print('1'); // Seri bilgi olarak 1 gönderildi
14.  }
15.
16.  if (buttonState == LOW) {
17.    Serial.print('0'); // Seri bilgi olarak 0 gönderildi
18.  }
19.  delay(100);
20. }
```

on?XT-İtigiib

Arduino ve HC-05'in Slave kullanımı - L293 bağlantısı.

Arduino ve HC-05 Bluetooth modülü'nü bu referansla olarak kullanılmak üzere yapılmıştır. Bu bağlantıya önce HC-05 AT komutları kullanılarak işlemi gerçekleştirilmiştir. Aşağıda ise devre tabiri kullanılmıştır. Böylece cep telefon veya bilgisayar ile Bluetooth üzerinden bağlantı yapılarak işlemi slave olarak gerçekleştirilebilir. Karşılıklı olarak yapılır.



Arduino ile HC-05'in slave kullanımı. L293D HC-05 Slave.ino

Arduino Uno ile HC-05 38400 Baud rate hızında iletilme geçerek robotx ismi verilen HC05'e, cep telefonundan AppStore ile yüklenen Arduino Bluetooth uygulaması içindeki robotx seçilip bu örnekte Terminal mode seçilerek motorun sağa veya sola dönüp durdurulması sağlanmıştır.

```
1 #define IN1 2 // L293D Entegresinin 1. pinleri
2 #define IN2 3 // L293D Entegresinin 2. pinleri
3 #define ENA 5 // L293D Entegresinin 3. pinleri
4 #define led1Pin 10
5 #define led2Pin 11
6 #define led3Pin 12
7 int state = 0; // state değişkeni tanımlanıyor
8
9 void setup() {
10   pinMode(led1Pin, OUTPUT);
11   pinMode(led2Pin, OUTPUT);
12   pinMode(led3Pin, OUTPUT);
13   digitalWrite(led2Pin, HIGH);
14   pinMode(IN1, OUTPUT);
15   pinMode(IN2, OUTPUT);
16   pinMode(ENA, OUTPUT);
17   Serial.begin(9600); // HC-05 modül seri iletişim hızı.
18   delay(100);
19 }
20
21 void loop() {
22   if (Serial.available() > 0) {
23     state = Serial.read(); // state değişkeni seri olarak okunuyor
24     Serial.write(state); // state değişkenini kaydet. Yaz.
25     if (state == 'L') { // state = L ise sola dön
26       digitalWrite(ENA, HIGH);
27       digitalWrite(IN1, HIGH);
28       digitalWrite(IN2, LOW);
29       state = 0;
30     }
31   }
32 }
```