

**KASTAMONU ÜNİVERSİTESİ**

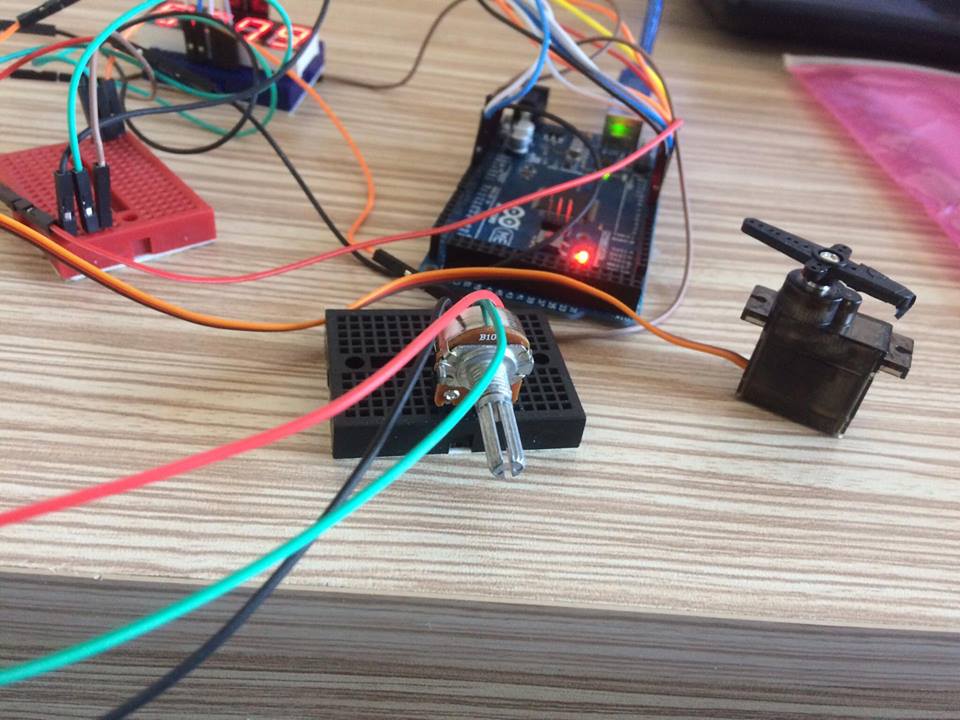
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

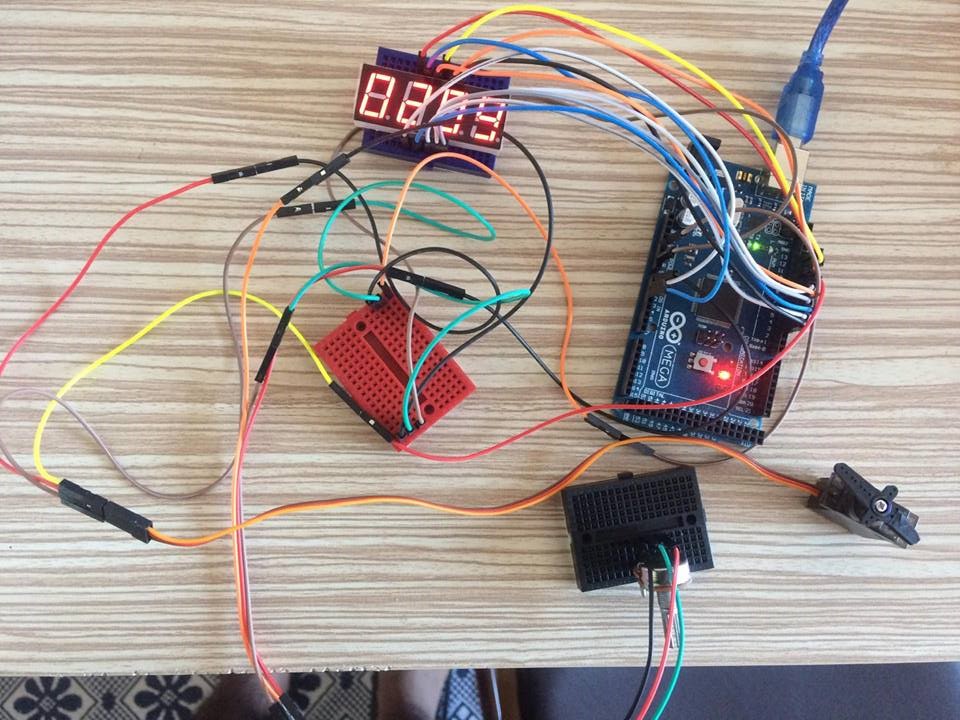
**ROBOTİK TEKNOLOJİLERİ**

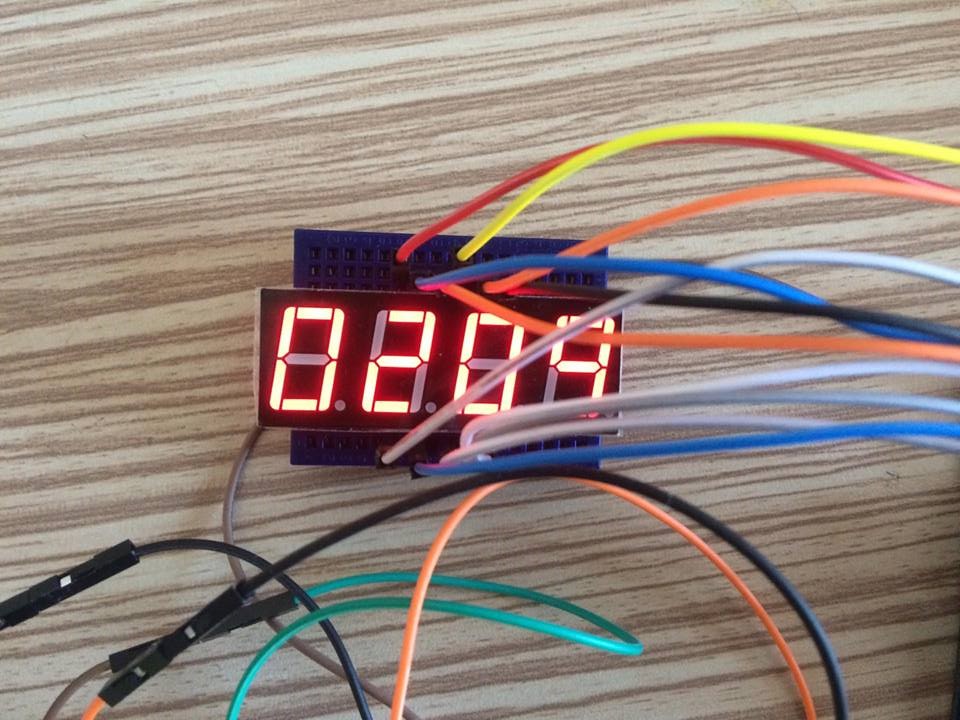
**SERVO MOTOR KONTROLÜ**

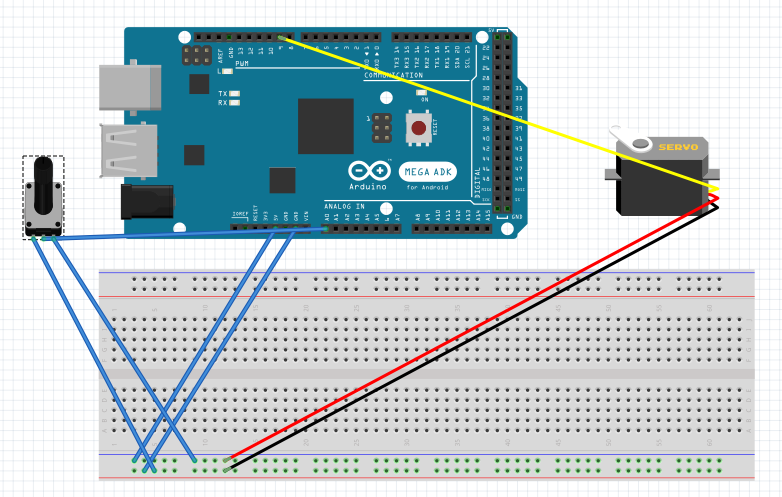
**Raporu Yazan Öğrenciler:**

RAMAZAN ŞAHİN(134410008)

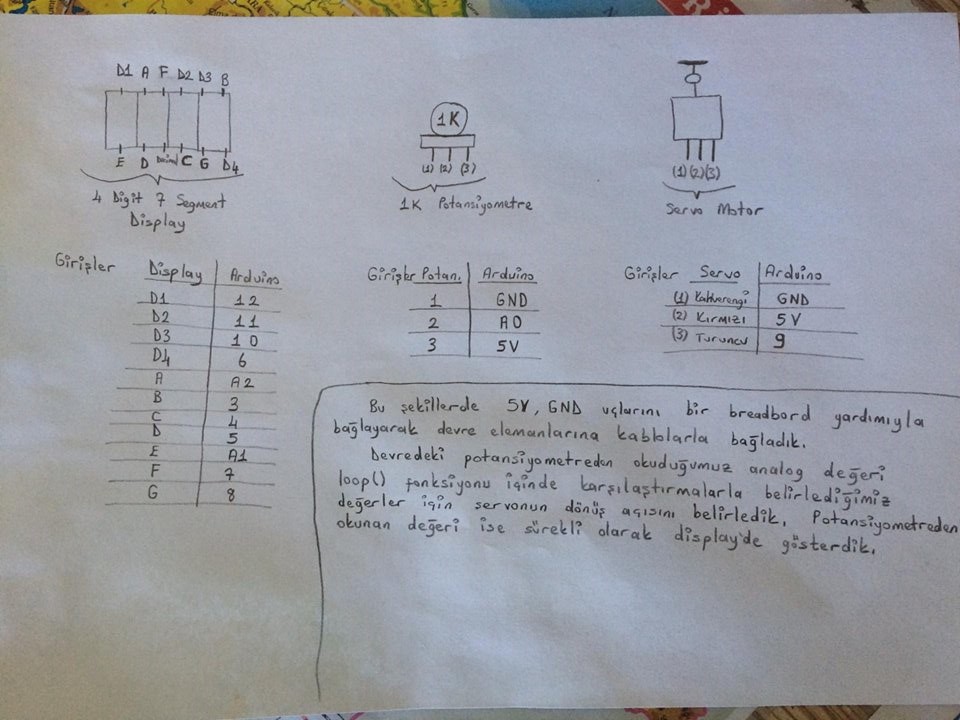








Bu devrede display hariç devre şeması. Fritzing kullanılarak yapılmıştır.



#include <LiquidCrystal.h>

LiquidCrystal lcd(38,39,36,37,34,35);//

#include <Servo.h> /\* Servo kutuphanesi projeye dahil edildi \*/

Servo servoNesnesi; /\* servo motor nesnesi yaratildi \*/

//Display Girisler

int digit1 = 12;

int digit2 = 11;

int digit3 = 10;

int digit4 = 6;

int segA = A2;

int segB = 3;

int segC = 4;

int segD = 5;

int segE = A1;

int segF = 7;

int segG = 8;

**void setup()**

{

//Display Giris Tanımları

pinMode(segA, OUTPUT);

pinMode(segB, OUTPUT);

pinMode(segC, OUTPUT);

pinMode(segD, OUTPUT);

pinMode(segE, OUTPUT);

pinMode(segF, OUTPUT);

pinMode(segG, OUTPUT);

pinMode(digit1, OUTPUT);

pinMode(digit2, OUTPUT);

pinMode(digit3, OUTPUT);

pinMode(digit4, OUTPUT);

pinMode(A0,INPUT);

Serial.begin(9600);

servoNesnesi.attach(9); /\* Servo motor 9 numarali pine baglandi \*/

}

**void loop()**

{

int smoke=analogRead(A0);//Potansiyometreden okuduğumuz değeri smoke değişkenine aktarıyoruz

Serial.println(smoke);//okuduğumuz değeri serial ekrana yazdırıyoruz

if(smoke>=0 && smoke<150)

{

servoNesnesi.write(0); /\* Motorun mili 100. dereceye donuyor \*/

}

else if(smoke>100 && smoke<350)

{

servoNesnesi.write(50); /\* Motorun mili 100. dereceye donuyor \*/

}

else if(smoke>350 && smoke<600)

{

servoNesnesi.write(100); /\* Motorun mili 100. dereceye donuyor \*/

}

else if(smoke>600 && smoke<800)

{

servoNesnesi.write(150); /\* Motorun mili 100. dereceye donuyor \*/

}

else

{

servoNesnesi.write(180); /\* Motorun mili 100. dereceye donuyor \*/

}

displayNumber(smoke);//okuduğumuz değeri display'de gösteriyoruz

}

//Display'de gösterilen değerlerin yazdırılma komutları

**void displayNumber(int toDisplay)** {

#define DISPLAY\_BRIGHTNESS 500

#define DIGIT\_ON HIGH

#define DIGIT\_OFF LOW

long beginTime = millis();

for(int digit = 4 ; digit > 0 ; digit--) {

//Turn on a digit for a short amount of time

switch(digit) {

case 1:

digitalWrite(digit1, DIGIT\_ON);

break;

case 2:

digitalWrite(digit2, DIGIT\_ON);

break;

case 3:

digitalWrite(digit3, DIGIT\_ON);

break;

case 4:

digitalWrite(digit4, DIGIT\_ON);

break;

}

//Turn on the right segments for this digit

lightNumber(toDisplay % 10);

toDisplay /= 10;

delayMicroseconds(DISPLAY\_BRIGHTNESS);

//Display digit for fraction of a second (1us to 5000us, 500 is pretty good)

//Turn off all segments

lightNumber(10);

//Turn off all digits

digitalWrite(digit1, DIGIT\_OFF);

digitalWrite(digit2, DIGIT\_OFF);

digitalWrite(digit3, DIGIT\_OFF);

digitalWrite(digit4, DIGIT\_OFF);

}

while( (millis() - beginTime) < 10) ;

//Wait for 20ms to pass before we paint the display again

}

//Given a number, turns on those segments

//If number == 10, then turn off number

**void lightNumber(int numberToDisplay)** {

#define SEGMENT\_ON LOW

#define SEGMENT\_OFF HIGH

switch (numberToDisplay){

case 0:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 1:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 2:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_OFF);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_ON);

break;

case 3:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_ON);

break;

case 4:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 5:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 6:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 7:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

case 8:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_ON);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 9:

digitalWrite(segA, SEGMENT\_ON);

digitalWrite(segB, SEGMENT\_ON);

digitalWrite(segC, SEGMENT\_ON);

digitalWrite(segD, SEGMENT\_ON);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_ON);

digitalWrite(segG, SEGMENT\_ON);

break;

case 10:

digitalWrite(segA, SEGMENT\_OFF);

digitalWrite(segB, SEGMENT\_OFF);

digitalWrite(segC, SEGMENT\_OFF);

digitalWrite(segD, SEGMENT\_OFF);

digitalWrite(segE, SEGMENT\_OFF);

digitalWrite(segF, SEGMENT\_OFF);

digitalWrite(segG, SEGMENT\_OFF);

break;

}

}