

```

//DENEEME

#include <DHT11.h>
#include <LiquidCrystal_I2C.h>
#include <AFMotor.h>
#include <SoftwareSerial.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
#define echoPin 7
#define trigPin 8
DHT11 dht11(2);
SoftwareSerial bt_iletisim(0, 1);

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

int dgr;
int hiz = 255;
bool btMod = true;

void setup() {
  Serial.begin(9600);
  bt_iletisim.begin(9600);
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
  dur();
  lcd.init();
}

void loop() {

  if (btMod) {
    bt_kontrol();
  } else {
    engelden_kacan();
  }
}

void ileri() {
  motor1.setSpeed(hiz); //Define maximum velocity
  motor1.run(FORWARD); //rotate the motor clockwise
  motor2.setSpeed(hiz); //Define maximum velocity
  motor2.run(FORWARD); //rotate the motor clockwise
  motor3.setSpeed(hiz); //Define maximum velocity
  motor3.run(FORWARD); //rotate the motor clockwise
  motor4.setSpeed(hiz); //Define maximum velocity
  motor4.run(FORWARD); //rotate the motor clockwise
}

void geri() {
  motor1.setSpeed(hiz); //Define maximum velocity
  motor1.run(BACKWARD); //rotate the motor anti-clockwise

```

```

    motor2.setSpeed(hiz); //Define maximum velocity
    motor2.run(BACKWARD); //rotate the motor anti-clockwise
    motor3.setSpeed(hiz); //Define maximum velocity
    motor3.run(BACKWARD); //rotate the motor anti-clockwise
    motor4.setSpeed(hiz); //Define maximum velocity
    motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void geri2() {
    motor1.setSpeed(hiz / 2.4); //Define maximum velocity
    motor1.run(BACKWARD); //rotate the motor anti-clockwise
    motor2.setSpeed(hiz / 2.4); //Define maximum velocity
    motor2.run(BACKWARD); //rotate the motor anti-clockwise
    motor3.setSpeed(hiz); //Define maximum velocity
    motor3.run(BACKWARD); //rotate the motor anti-clockwise
    motor4.setSpeed(hiz); //Define maximum velocity
    motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void sol() {
    motor1.setSpeed(hiz); //Define maximum velocity
    motor1.run(FORWARD); //rotate the motor clockwise
    motor2.setSpeed(hiz); //Define maximum velocity
    motor2.run(FORWARD); //rotate the motor clockwise
    motor3.setSpeed(hiz); //Define maximum velocity
    motor3.run(BACKWARD); //rotate the motor anti-clockwise
    motor4.setSpeed(hiz); //Define maximum velocity
    motor4.run(BACKWARD); //rotate the motor anti-clockwise
}

void sag() {
    motor1.setSpeed(hiz); //Define maximum velocity
    motor1.run(BACKWARD); //rotate the motor anti-clockwise
    motor2.setSpeed(hiz); //Define maximum velocity
    motor2.run(BACKWARD); //rotate the motor anti-clockwise
    motor3.setSpeed(hiz); //Define maximum velocity
    motor3.run(FORWARD); //rotate the motor clockwise
    motor4.setSpeed(hiz); //Define maximum velocity
    motor4.run(FORWARD); //rotate the motor clockwise
}

void dur() {
    motor1.setSpeed(0); //Define minimum velocity
    motor1.run(RELEASE); //stop the motor when release the button
    motor2.setSpeed(0); //Define minimum velocity
    motor2.run(RELEASE); //rotate the motor clockwise
    motor3.setSpeed(0); //Define minimum velocity
    motor3.run(RELEASE); //stop the motor when release the button
    motor4.setSpeed(0); //Define minimum velocity
    motor4.run(RELEASE); //stop the motor when release the button
}

void bt_kontrol() {
    if (bt_iletisim.available()) {

```

```

char dgr = bt_iletisim.read();
Serial.println(dgr);
dur();

if (dgr == 'F') {
    ileri();
} else if (dgr == 'B') {
    geri();
} else if (dgr == 'L') {
    sol();
} else if (dgr == 'R') {
    sag();
} else if (dgr == 'x' || dgr == 'X') {
    btMod = false;
} else if (dgr == 'v' || dgr == 'V') {
    btMod = true;
} else {
    dur();
}
}
}

```

```

void engelden_kacan() {
    if (bt_iletisim.available()) {
        char dgr = bt_iletisim.read();
        if (dgr == 'x' || dgr == 'X') {
            btMod = false;
        } else if (dgr == 'v' || dgr == 'V') {
            btMod = true;
        }
    }
}

```

```

digitalWrite(trigPin, LOW);
delayMicroseconds(5);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
int sure = pulseIn(echoPin, HIGH);
int uzaklik = sure / 29.1 / 2;
lcd.backlight();
lcd.setCursor(0, 0);
lcd.print("KAYU BILGISAYAR");
lcd.setCursor(0, 1);
lcd.print("PROGRAMCILIGI");
Serial.print(uzaklik);
Serial.println(" cm \n");
if (uzaklik <= 15) {
    hiz = 150;
    ileri();
    delay(200);
    sol();
    delay(100);
} else {
    hiz = 200;
}

```

```

        geri2();
    }
}

/*void lcd_bilgi() {

    if (bt_iletisim.available()) {
        char dgr = bt_iletisim.read();
        if (dgr == 'x' || dgr == 'X') {
            btMod = false;
        } else if (dgr == 'v' || dgr == 'V') {
            btMod = true;
        }

        int nem = dht11.readHumidity();
        int sicaklik = dht11.readTemperature();

        lcd.backlight();
        lcd.clear();

        lcd.setCursor(0, 0);
        lcd.print("KAYU BILG. PROG.");

        lcd.setCursor(7, 1);
        lcd.print(sicaklik);
        lcd.print("C");

        lcd.setCursor(13, 1);
        lcd.print(nem);
        lcd.print("%");
    }
}*/

```