Number: 2017-2
FEN VE TEKNOLOJİ BİLGİ PAYLAŞIMI
SCIENCE AND TECHNOLOGY INFORMATION SHARING
Article Web Page: www.ibrahimcayiroglu.com

Uzaktan Kumandalı Manyetik Küresel Dengelemeli Robot

Radio Controlled Magnetic Spherical Balancing Robot

Mustafa Attila US* , İbrahim ÇAYIROĞLU**

*Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği, 78050, Karabük, mustafattila@outlook.com
**Karabük Üniversitesi, Mühendislik Fakültesi, Mekatronik Mühendisliği, 78050, Karabük, icayiroglu@yahoo.com

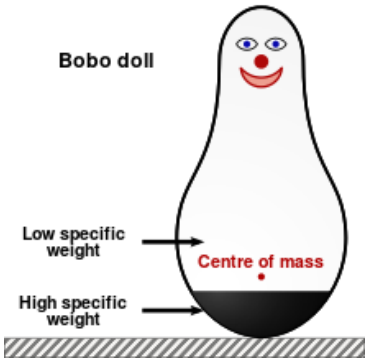
Anahtar Kelimeler:
Bobo Doll Robotu

Özet: Bu makalede Bobo Doll'un genel bir tanıtımı yapıldıktan sonra aynı prensiple çalışan bir robot örneği verilmiştir. Bobo Doll ağırlık merkezinin iki farklı kütle yardımıyla desteklenen devrilmeyen bir oyuncaktır. Verilen örnek direk uygulanıp denenebilir. Kavramların daha iyi anlaşılması için hem Türkçe alternatif anlatımları hem de İngilizce karşılıkları verilmiştir.

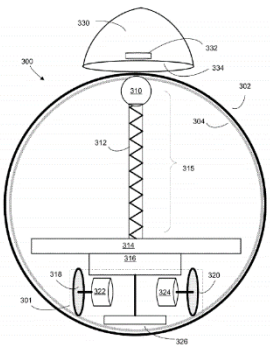
Keywords:
Bobo Doll Robot

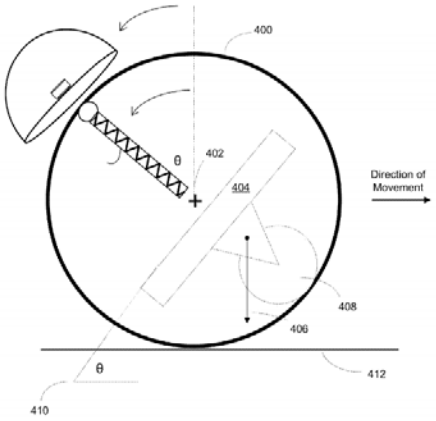
Abstract: In this article, given an robotic example of Bobo Doll after a general introduction. Bobo Doll is an doll which is impossible to overthrow due to location of its center of gravity. The examples given are applied directly. Alternative explanations for better understanding of concepts in both Turkish and English equivalents are given.

1. Giriş



Bobo Doll olarak bahsedilen Türkiye'de hacıyatmaz olarak bilinen oyuncağın çalışma prensibi şekilde gözüktüğü gibi ağırlığın tabanda oldukça yüksek ve geri kalan kısımlarda oldukça düşük olması ve buna bağlı olarak ağırlık merkezinin oyuncağın tabanına yakın olmasıyla devrilmemesi üzerinedir. Bu prensibin robotikte uygulanması da mümkündür.





Şekilde görüldüğü üzere seçilen hareket yönünde hareket edilebilmesi için harici ağırlıkların robotun iç tasarımında kullanılması gerekmektedir ve bu hareketi yaparken tekerlek kullanmaktadır. Benzetmek gerekirse hamster çarkı gibi düşünülebilir. Gövde kürenin içinde yere paralel uzanan düzlem sisteminin bitimlerindeki 4 adet tekerlek vasıtasıyla hamster türü farelerin yaptığı hareketi sağlayarak düzlemin yere daima paralel bir şekilde durması sağlanmalıdır. Bunu sağlamak için düzlemin alt kısmına -üst kısmında Arduino Uno R3 Board bulunacağından- kütle merkezini yere yaklaştıracak şekilde ağırlık asılması gerekmektedir.

2. Malzemelerin Bir Araya Getirilmesi

2.1 *Batarya*

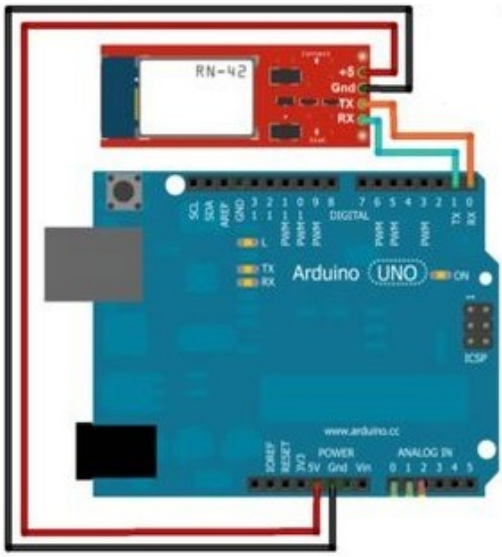
Malzeme listesinde belirtildiği gibi kullanılacak olan 18650 Lityum-İyon şarj edilebilir 3.7V 3000mAh pillerin 2 paralel set olarak birbirine bağlanmasıyla 14.4V 6000mAh gücünde bir batarya sistemi elde edilecektir.

2.2 *Bluetooth Modülün Kurulumu*

Arduino Uno R3 uyumlu Bluetooth modülü HC-05 Şekil 4.'te görüldüğü üzere

1. Arduino TX -> HC-05 RX'e
2. Arduino RX -> HC-05 TX'e
3. 5V -> VCC Güç bağlantısı
4. Topraklamaların ortaklanması

Olmak üzere 4 adımda bağlantısı yapılmaktadır.



Şekil 4. Arduino Uno R3 & HC-05 Bağlantıları

3.2 *Arduino için Pololu Motor Shield Library Programlanması*

VNH5019MotorShield.h DOSYASI:

```
#ifndef VNH5019MotorShield_h
#define VNH5019MotorShield_h

#include <Arduino.h>

class VNH5019MotorShield
{
  public:
        VNH5019MotorShield(); //
Varsayılan pin seçimi.
    VNH5019MotorShield(unsigned char INA1,
unsigned char INB1, unsigned char
EN1DIAG1, unsigned char CS1,
                       unsigned char
INA2, unsigned char INB2, unsigned char
EN2DIAG2, unsigned char CS2);


    void init(); // TIMER 1'i kur, PWM
değerini 20kHZ ayarla.
    void setM1Speed(int speed); // Hızı M1
Ayarla.
    void setM2Speed(int speed); // Hızı M2
Ayarla.
    void setSpeeds(int m1Speed, int
m2Speed); // Hızı hem M1 hem M2 için
ayarla.
    void setM1Brake(int brake); // M1
Frenlemesi.
    void setM2Brake(int brake); // M2
Frenlemesi.
    void setBrakes(int m1Brake, int
m2Brake); // Hem M1 hem M2 frenlenmesi.
    unsigned int getM1CurrentMilliamps();
// Akım değerini M1'den al.
    unsigned int getM2CurrentMilliamps();
// Akım değerini M2'den al.
    unsigned char getM1Fault(); // Hata
değerini M1'den al..
    unsigned char getM2Fault(); // Hata
değerini M2'den al..

  private:
    unsigned char _INA1;
    unsigned char _INB1;
    static const unsigned char _PWM1 = 9;
    unsigned char _EN1DIAG1;
    unsigned char _CS1;
    unsigned char _INA2;
    unsigned char _INB2;
    static const unsigned char _PWM2 = 10;
    unsigned char _EN2DIAG2;
    unsigned char _CS2;

};

#endif
```

VNH5019MotorShield.cpp DOSYASI

```
#include "VNH5019MotorShield.h"


VNH5019MotorShield::VNH5019MotorShield()
{
  //Pin haritası
  _INA1 = 2;
  _INB1 = 4;
  _EN1DIAG1 = 6;
  _CS1 = A0;
  _INA2 = 7;
  _INB2 = 8;
  _EN2DIAG2 = 12;
  _CS2 = A1;
}
```

```
VNH5019MotorShield::VNH5019MotorShield(uns
igned char INA1, unsigned char INB1,
unsigned char EN1DIAG1, unsigned char CS1,

unsigned char INA2, unsigned char INB2,
unsigned char EN2DIAG2, unsigned char CS2)
{
  //Pin haritası
  //PWM1 ve PWM2 kütüphanede PWM timer1
olduğu için yeniden haritalanamıyor
  _INA1 = INA1;
  _INB1 = INB1;
  _EN1DIAG1 = EN1DIAG1;
  _CS1 = CS1;
  _INA2 = INA2;
  _INB2 = INB2;
  _EN2DIAG2 = EN2DIAG2;
  _CS2 = CS2;
}
void VNH5019MotorShield::init()
{
// pinler için pinMode tanımla ve timer1
için frekans gir.

  pinMode(_INA1,OUTPUT);
  pinMode(_INB1,OUTPUT);
  pinMode(_PWM1,OUTPUT);
  pinMode(_EN1DIAG1,INPUT);
  pinMode(_CS1,INPUT);
  pinMode(_INA2,OUTPUT);
  pinMode(_INB2,OUTPUT);
  pinMode(_PWM2,OUTPUT);
  pinMode(_EN2DIAG2,INPUT);
  pinMode(_CS2,INPUT);
  #if defined(__AVR_ATmega168__)||
defined(__AVR_ATmega328P__) ||
defined(__AVR_ATmega32U4__)
  // Timer 1 konfigürasyonu
  // prescaler: clockI/O / 1
  // outputs enabled
  // phase-correct PWM
  // top of 400
  //
  // PWM frekans hesaplanması
  // 16MHz / 1 (prescaler) / 2 (phase-
correct) / 400 (top) = 20kHz
  TCCR1A = 0b10100000;
  TCCR1B = 0b00010001;
  ICR1 = 400;
  #endif
}
// motor 1 için hız ayarla, hız değeri -
400 ile 400 arasında olmalı
void VNH5019MotorShield::setM1Speed(int
speed)
{
  unsigned char reverse = 0;

  if (speed < 0)
  {
    speed = -speed;  // hızı pozitif
değere çevir
    reverse = 1;  // yön seç
  }
  if (speed > 400)  // Maks PWM dutycycle
    speed = 400;
  #if defined(__AVR_ATmega168__)||
defined(__AVR_ATmega328P__) ||
defined(__AVR_ATmega32U4__)
  OCR1A = speed;
  #else
  analogWrite(_PWM1,speed * 51 / 80); //
analogWrite kullan, 400den 255e
  #endif
  if (speed == 0)
  {
    digitalWrite(_INA1,LOW);   // motor
kızağını sıfır al
    digitalWrite(_INB1,LOW);   // hangi
yöne döndüğüne göre
  }
  else if (reverse)
  {
    digitalWrite(_INA1,LOW);
    digitalWrite(_INB1,HIGH);
  }
  else
  {
    digitalWrite(_INA1,HIGH);
    digitalWrite(_INB1,LOW);
  }
}

//  motor 2 için hız ayarla, hız değeri -
400 ile 400 arasında olmalı
void VNH5019MotorShield::setM2Speed(int
speed)
{
  unsigned char reverse = 0;

  if (speed < 0)
  {
    speed = -speed;  // hızı pozitife
çevir
    reverse = 1;  // yön seç
  }
  if (speed > 400)  // maksimum
    speed = 400;
  #if defined(__AVR_ATmega168__)||
defined(__AVR_ATmega328P__) ||
defined(__AVR_ATmega32U4__)
  OCR1B = speed;
  #else
  analogWrite(_PWM2,speed * 51 / 80); //
analogWrite kullan, 400den 255e
  #endif
  if (speed == 0)
  {
    digitalWrite(_INA2,LOW);   // motor
kızağını sıfır al
    digitalWrite(_INB2,LOW);   // hangi
yöne döndüğüne göre
  }
  else if (reverse)
  {
    digitalWrite(_INA2,LOW);
    digitalWrite(_INB2,HIGH);
  }
  else
  {
    digitalWrite(_INA2,HIGH);
    digitalWrite(_INB2,LOW);
```

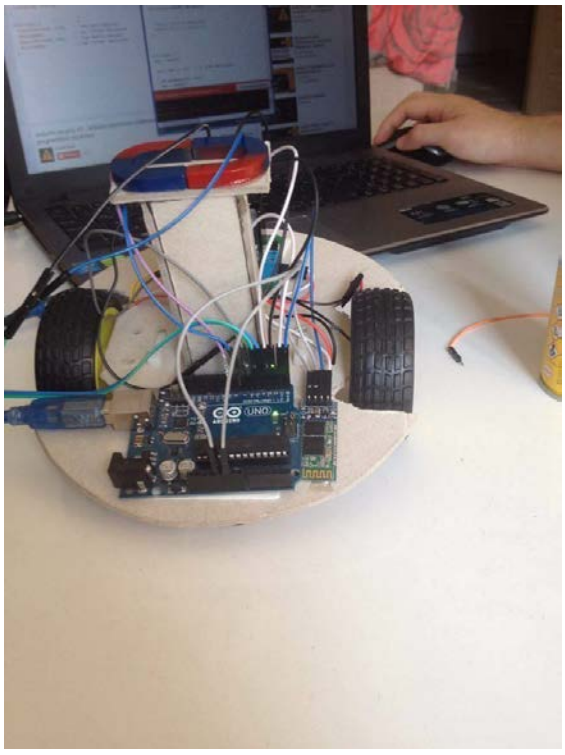
```
  }
}

// motor 1 ve 2 için hız ayarı
void VNH5019MotorShield::setSpeeds(int
m1Speed, int m2Speed)
{
  setM1Speed(m1Speed);
  setM2Speed(m2Speed);
}

// motor 1 frenle, frenleme 0 ile 400
arasında olmalı
void VNH5019MotorShield::setM1Brake(int
brake)
{
  // normalize fren
  if (brake < 0)
  {
    brake = -brake;
  }
  if (brake > 400)  // maks fren
    brake = 400;
  digitalWrite(_INA1, LOW);
  digitalWrite(_INB1, LOW);
  #if defined(__AVR_ATmega168__)||
defined(__AVR_ATmega328P__) ||
defined(__AVR_ATmega32U4__)
  OCR1A = brake;
  #else
  analogWrite(_PWM1,brake * 51 / 80); //
analogWrite kullan, 400den 255e
  #endif
}

// motor 2yi frenle, frenleme değeri 0 ile
400 arası
void VNH5019MotorShield::setM2Brake(int
brake)
{
  // normalize fren
  if (brake < 0)
  {
    brake = -brake;
  }
  if (brake > 400)  // Maks brake
    brake = 400;
  digitalWrite(_INA2, LOW);
  digitalWrite(_INB2, LOW);
  #if defined(__AVR_ATmega168__)||
defined(__AVR_ATmega328P__) ||
defined(__AVR_ATmega32U4__)
  OCR1B = brake;
  #else
  analogWrite(_PWM2,brake * 51 / 80); //
analogWrite kullan, 400den 255e
  #endif
}

// motor 1 and 2 frenle, frenleme değeri 0
ile 400 arası
void VNH5019MotorShield::setBrakes(int
m1Brake, int m2Brake)
{
  setM1Brake(m1Brake);
  setM2Brake(m2Brake);
}

// motor 1 akım değerini miliamper olarak
göster.
unsigned int
VNH5019MotorShield::getM1CurrentMilliamps(
)
{
  // 5V / 1024 ADC counts / 144 mV per A =
34 mA per count
  return analogRead(_CS1) * 34;
}

// motor 2 akım değerini miliamper olarak
göster.
unsigned int
VNH5019MotorShield::getM2CurrentMilliamps(
)
{
  // 5V / 1024 ADC counts / 144 mV per A =
34 mA per count
  return analogRead(_CS2) * 34;
}

//  motor 1  hatası
unsigned char
VNH5019MotorShield::getM1Fault()
{
  return !digitalRead(_EN1DIAG1);
}

// motor 2 hatası
unsigned char
VNH5019MotorShield::getM2Fault()
{
  return !digitalRead(_EN2DIAG2);
}
```

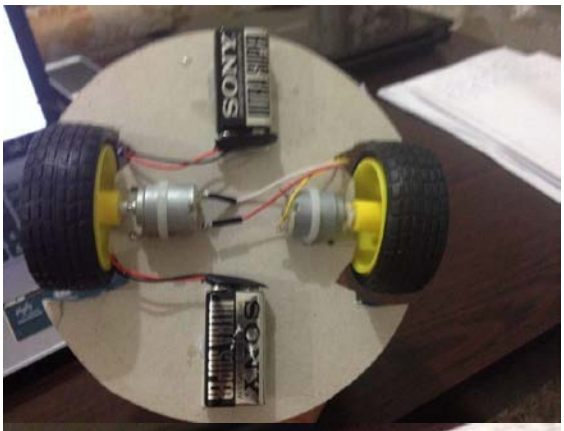
3.3 Arduino Programlanması

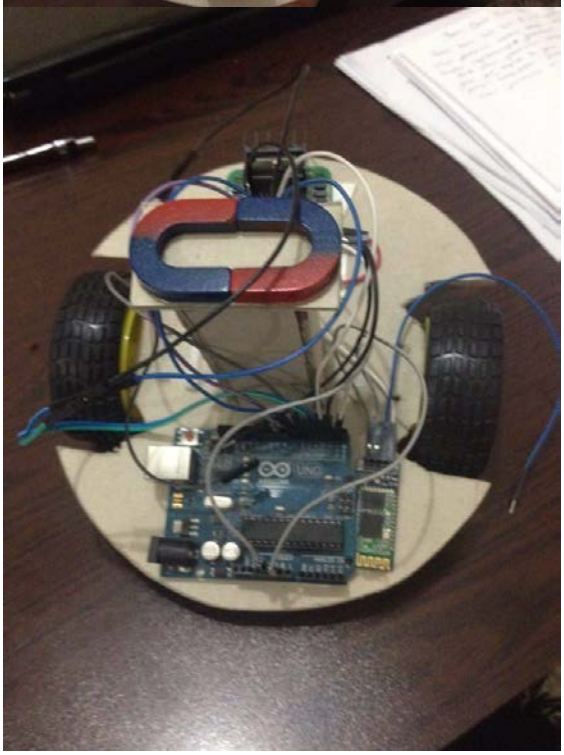
```
#include "VNH5019MotorShield.h"
DualVNH5019MotorShield md;
/*
#include <Servo.h>
Servo myservo;
*/

char dataIn='S';
char determinant;
char det;
int vel = 200; //Bluetooth Ayarı

int overdrive = 13; //Swiç #1 e bas, pin13
LED yanar.

void setup(){
Serial.begin(9600);md.init();

/*
myservo.attach(6);delay(100);
myservo.write(90);delay(100);
*/

}
```

```
void loop(){ det = check();

      while (det == 'F')   // F, ileri yön
hareket
       {md.setSpeeds(vel,vel);det =
check();}

      while (det == 'B')   // B, geri
hareket
       {md.setSpeeds(-vel,-vel);det =
check();}

      while (det == 'L')   // L, tekerleri
sola döndür
       {md.setSpeeds(-vel,vel);det =
check();}

      while (det == 'R')   // R, tekerleri
sağa döndür
       {md.setSpeeds(vel,-vel);det =
check();}

      while (det == 'I')   // I, sağ ve
ileri dön
       {md.setSpeeds(vel,vel/2);det =
check();}

      while (det == 'J')   // J, sağa ve
geri dön
       {md.setSpeeds(-vel,-vel/2);det =
check();}

      while (det == 'G')   // G, sol ve
ileri dön
       {md.setSpeeds(vel/2,vel);det =
check();}

      while (det == 'H')   // H, sol ve
geri dön
       {md.setSpeeds(-vel/2,-vel);det =
check();}

      while (det == 'S')   // S, dur
       {md.setSpeeds(0,0);det = check();}

      //--------------------swiç
klodları-----------------//
      /*while (det ==
'W'){myservo.write(180);delay(100);det =
check();}
      while (det ==
'w'){myservo.write(90);delay(100);det =
check();}

      while (det ==
'U'){myservo.write(0);delay(100);det =
check();}
      while (det ==
'u'){myservo.write(90);delay(100);det =
check();}
      */

}

int check()
{if (Serial.available() > 0) {dataIn =
Serial.read();
        if (dataIn == 'F'){determinant =
'F';}
        else if (dataIn ==
'B'){determinant = 'B';}else if (dataIn ==
'L'){determinant = 'L';}
        else if (dataIn ==
'R'){determinant = 'R';}else if (dataIn ==
'I'){determinant = 'I';}
        else if (dataIn ==
'J'){determinant = 'J';}else if (dataIn ==
'G'){determinant = 'G';}
        else if (dataIn ==
'H'){determinant = 'H';}else if (dataIn ==
'S'){determinant = 'S';}
        else if (dataIn == '0'){vel =
400;}else if (dataIn == '1'){vel = 380;}
        else if (dataIn == '2'){vel =
340;}else if (dataIn == '3'){vel = 320;}
        else if (dataIn == '4'){vel =
280;}else if (dataIn == '5'){vel = 240;}
        else if (dataIn == '6'){vel =
200;}else if (dataIn == '7'){vel = 160;}
        else if (dataIn == '8'){vel =
120;}else if (dataIn == '9'){vel = 80;}
        else if (dataIn == 'q'){vel = 40;}
        else if (dataIn ==
'U'){determinant = 'U';}else if (dataIn ==
'u'){determinant = 'u';}
        else if (dataIn ==
'W'){determinant = 'W';}else if (dataIn ==
'w'){determinant = 'w';}

        }return determinant;}
```

4. Projeye Ait Bazı Resimler







_____ The Author _____

Ibrahim Cayiroglu *is an instructor in Mechatronic Engineering at Karabük University, Turkey. He received his B.Sc. in Mechanical Engineering from Istanbul Technical University in 1991. He received his M.Sc. and Ph.D. in Computer Aided Design and Manufacturing from Kirikkale University, in 1996 and 2002, respectively. His research interests include CAD-CAM, Software and Mechatronic Systems.*

Mustafa Attila US *is a student in Mechatronic Engineering at Karabük University, Turkey. He was born in Nürnberg/DEUTSCHLAND. He studied and practiced AutoCAD, Ansys, Visual Studio, Matlab, Plc, Arduino, KUKA, C#, SolidCAM & SolidWORKS programs with such projects like tower crane design & stress analysis, excavator design & rigid dynamics analysis, question answering engine, aircraft design & fluent analysis, automated student record system since his first year in university(2013).*