



Obstacle avoiding robot

Документация

Изготвили :

Семир Балджиев

Бешир Моллов

Семир Караахмедов

Самина Демирова

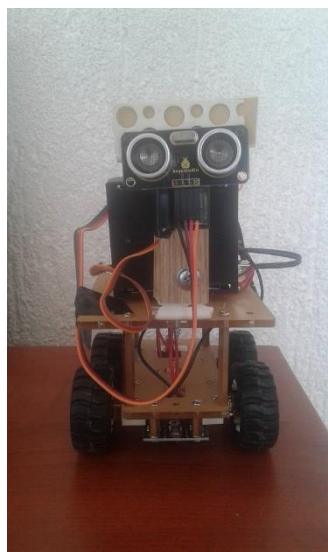
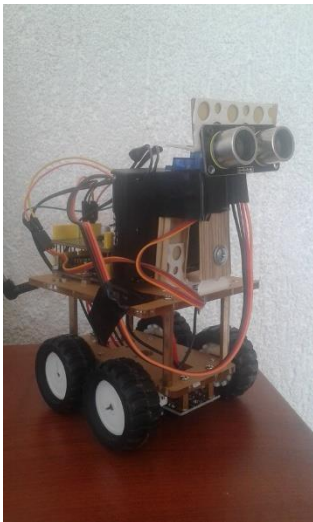
Ивета Говедарова

Съдържание

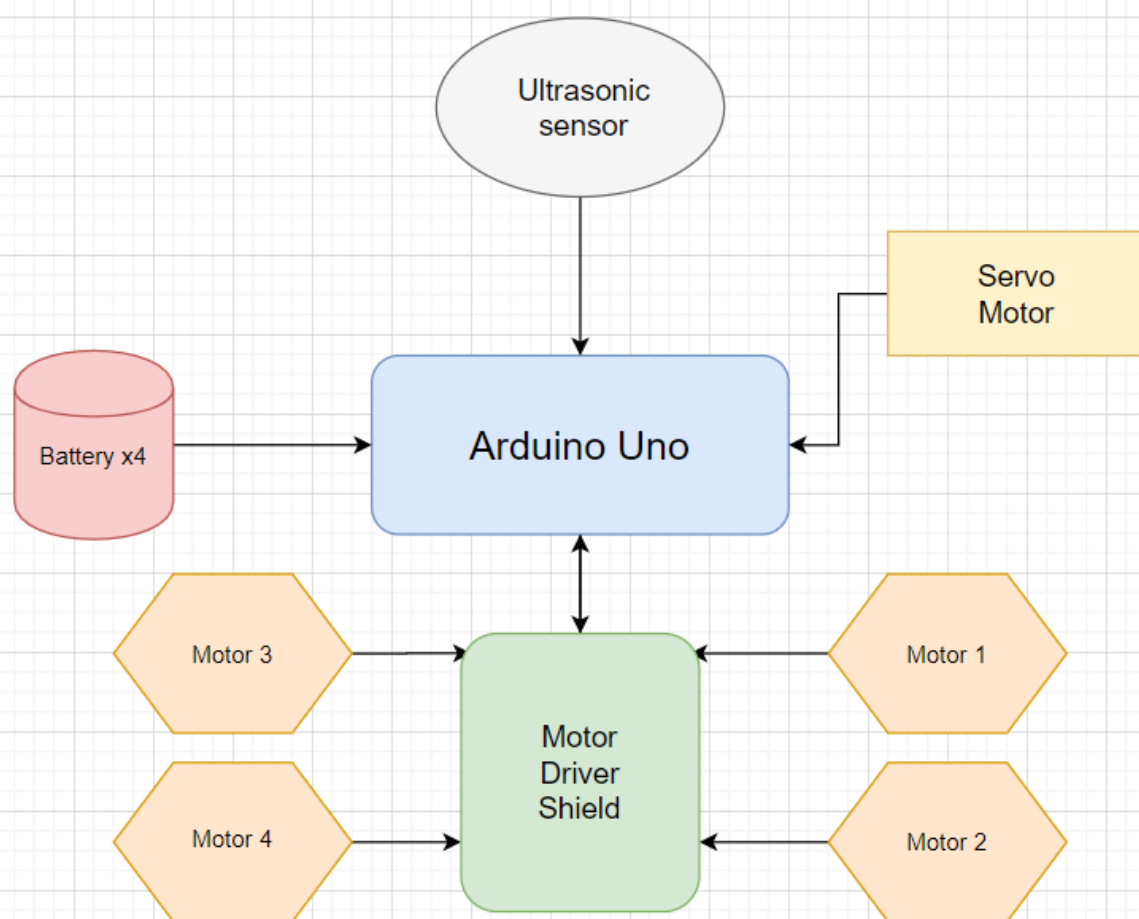
Описание.....	3
Блокова схема.....	4
Електрическа схема.....	5
Списък съставни части.....	6
Сорс код – описание на функционалността.....	7
Заключение.....	11

Описание

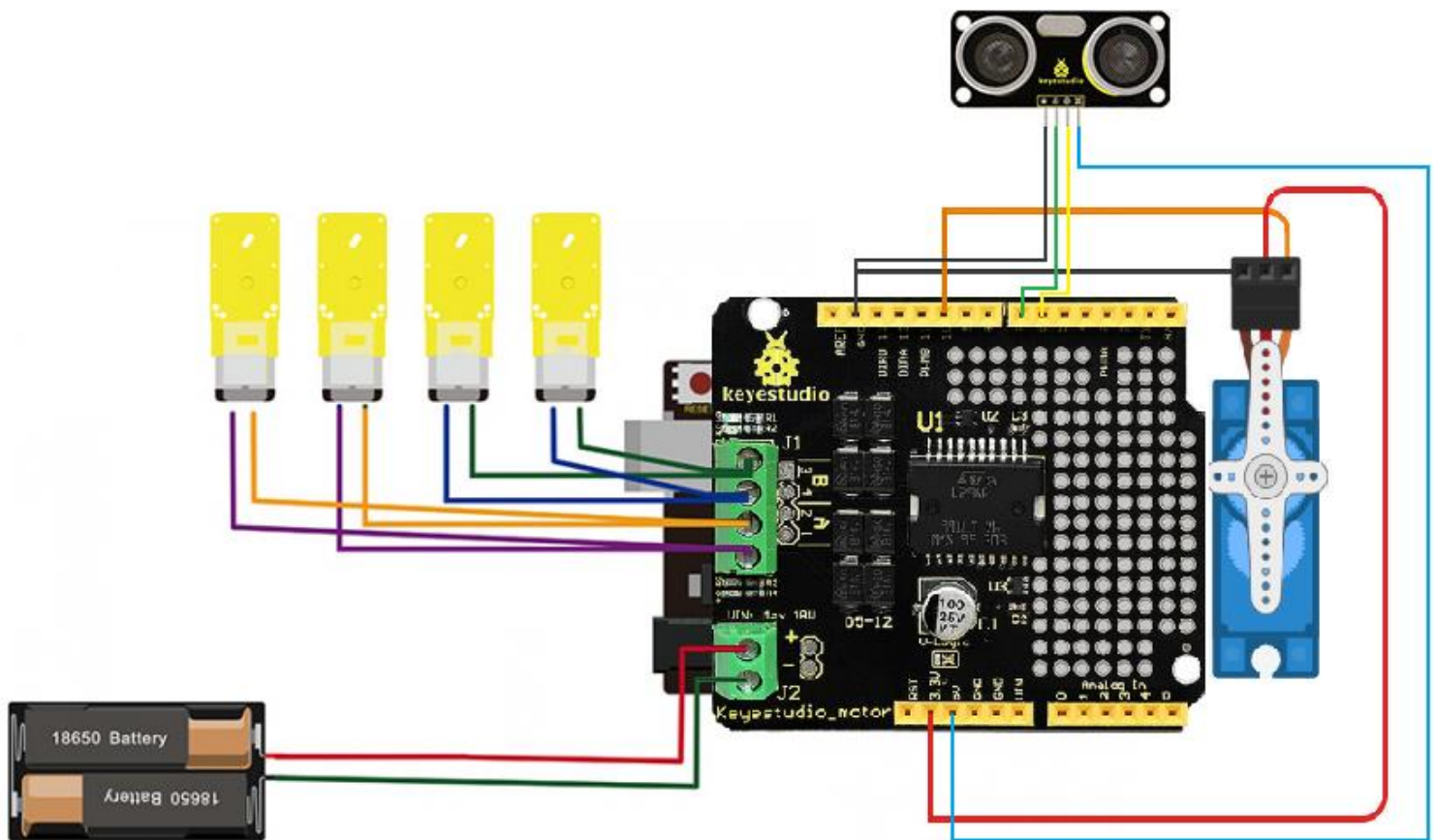
Нашият проект се казва „**Obstacle avoiding robot**“ и представлява робот който сам се ориантира в зависимост от преградите които засича напред, наляво, надясно и избира посоката в която ще тръгне. Намирането на преградите се осъществява от **Ultrasonic sensor**, който отчита на какво разстояние се намира робота от преградата. Основните посоки в които може да се движи са напред, надясно, наляво и назад. Нашият робот може да измерва разстояния в три посоки, за целта сме закрепили сензора за **Servo motor** който се върти на зададени градуси като завърта сензора в определената посока. За проекта сме използвали **Arduino Uno** платка.



Блокова схема

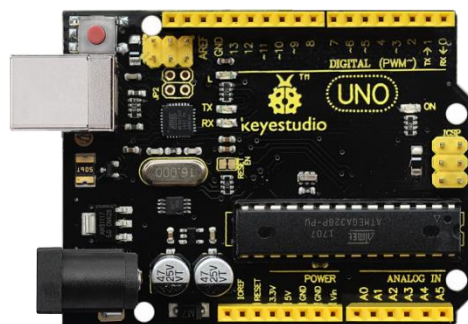


Електрическа схема

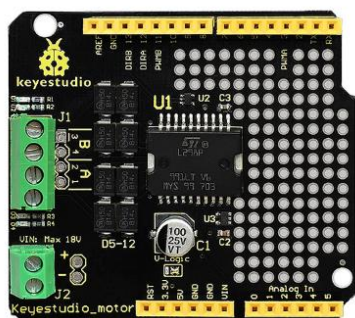


Списък съставни части

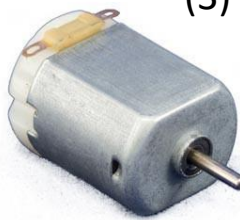
1. Arduino Uno
2. Motor Driver Shield L298P
3. DC Motor (x4)
4. Servo Motor
5. Ultrasonic Sensor
6. Wheels (x4)
7. Battery Holder
8. Male and Female Jumper
9. Battery (x4)



(1)



(2)



(3)



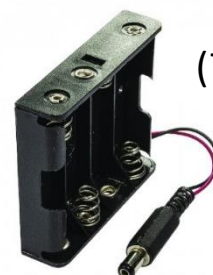
(4)



(5)



(6)



(7)



(8)



(9)

Сорс код – описание на функционалността

Тук ще опишем съдържанието на кода за Arduino Uno платката използван за работа. Програмата се разделя на няколко части като в началото дефинираме променливите и библиотеките които използваме.

```
#include <NewPing.h>
#include <Servo.h>

#define E1 3
#define E2 11
#define M1 12
#define M2 13

#define TRIG_PIN 7
#define ECHO_PIN 6
#define MAX_DISTANCE 200

NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
Servo myservo;

int distance;
int distanceR;
int distanceL;
```

Библиотеките които използваме са **NewPing** и **Servo**. **NewPing** библиотеката служи за настройване на **Ultrasonic Sensor**-а при подадени параметри. А **Servo** ни дава достъп до функциите за използване на серво мотора.

Пиновете **E1, E2, M1, M2** служат за задействане и контролиране на моторите.

Следващата част от кода са методите **Setup** и **Loop**.

Метода Setup се изпълнява първоначално при стартиране на работа, който изглежда така:

```

void setup() {
    // put your setup code here, to run once:
    pinMode(M1, OUTPUT);
    pinMode(M2, OUTPUT);
    pinMode(E1, OUTPUT);
    pinMode(E2, OUTPUT);
    myservo.attach(10);
    myservo.write(90);
    delay(300);
    Serial.begin(9600);
}

```

В този метод се настройват изходните пинове и се задава пин за серво мотора.

В **Loop** метода който изглежда така:

```

void loop() {
    // put your main code here, to run repeatedly:
    myservo.write(90);
    distance = readPing();

    if(distance<=22)
    {
        moveStop();
        delay(300);
        distanceR = lookRight();
        delay(300);
        distanceL = lookLeft();
        delay(300);
        if((distanceL<=25)&&(distanceR<=25)){
            moveBackward(200);
            delay(3000);
        }
        else
        {
            if(distanceR>=distanceL)
            {
                turnRight();
                delay(1000);
            }
            else
            {
                turnLeft();
                delay(1000);
            }
        }
    }
    else
    {
        moveForward();
        delay(1000);
    }
}

```

Се изпълнява цялата логика за действията на робота.

Логиката на робота е следната:

Преди тръгване измерва разстоянието пред него чрез метода **readPing** който изглежда така:

```
int readPing() {
    delay(70);
    int cm = sonar.ping_cm();
    if(cm==0)
    {
        cm = 250;
    }
    return cm;
}
```

Този метод връща разстоянието до преградата.

Ако се появи преграда която е на разстояние 22см или по-малко той ще спре и ще се огледа надясно или наляво което се осъществява чрез следните методи:

```
void moveStop()          // stop
{
    digitalWrite(M1,LOW);
    digitalWrite(M2, LOW);
    analogWrite(E1, 0);
    analogWrite(E2, 0);  // both left and right wheels stop
}
```

```
int lookRight()
{
    myservo.write(10);
    delay(650);
    int distance = readPing();
    delay(100);
    myservo.write(90);
    return distance;
}
```

```
int lookLeft()
{
    myservo.write(170);
    delay(650);
    int distance = readPing();
    delay(100);
    myservo.write(90);
    return distance;
}
```

Методът **lookRight** завърта серво мотора надясно и връща разстоянието до преградата ако има такава.

Методът **lookLeft** е аналогичен на предишния, но единствената разлика е, че серво мотора се завърта наляво.

Ако разстоянието и от двете страни е 25см или по-малко то робота ще се движи назад.

Методът изглежда по следния начин:

```
void moveBackward(int speedBack)          // move backward
{

    digitalWrite(M1,HIGH);  // both left and right wheels move backward
    digitalWrite(M2, HIGH);
    analogWrite(E1,speedBack);
    analogWrite(E2, speedBack);
}
```

Като тук скоростта се дава като параметър в зависимост от положението.

Но, ако разстоянието е по-голямо то сравнява дистанциите за наляво и надясно и в зависимост от това коя е по-голяма, тръгва в избраната посока.

```
void turnRight()          // turn right
{

    digitalWrite(M1,HIGH);
    digitalWrite(M2,LOW);
    analogWrite(E1,220);
    analogWrite(E2, 220);
}
void turnLeft()          // turn left
{

    digitalWrite(M1,LOW);
    digitalWrite(M2, HIGH);
    analogWrite(E1,220);
    analogWrite(E2, 220);
}
```

Ако има достатъчно разстояние пред него той ще се движи напред, а метода е следния:

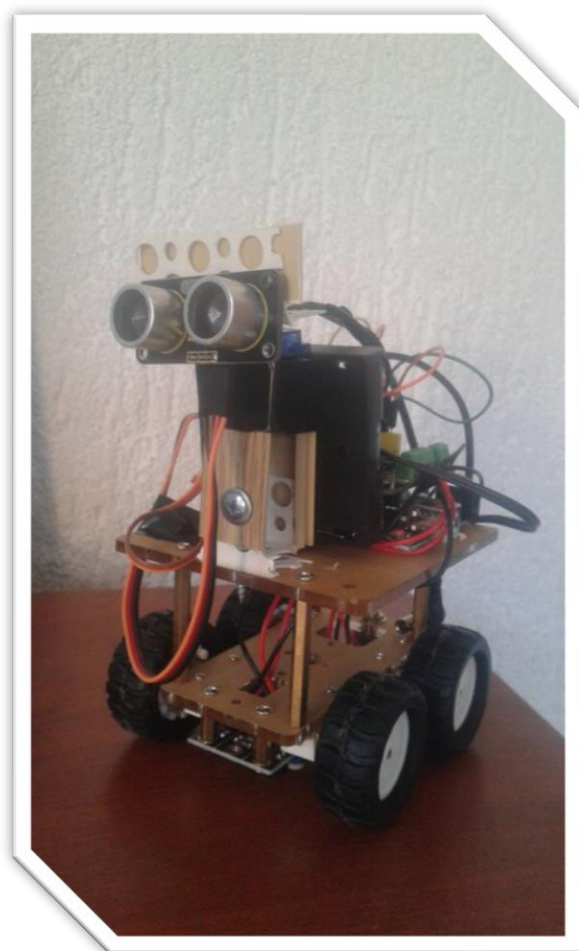
```
void moveForward()        // move forward
{

    digitalWrite(M1,LOW);
    digitalWrite(M2,LOW);
    analogWrite(E1,120);
    analogWrite(E2,120);
}
```

Библиотеките които са използвани за създаване на програмата са инсталирани от Manage Libraries на Arduino IDE.

Заклучение

Роботът се придвижва с четири колела и не се блъска в преградите. Той е тестван и работи.



GitHub: <https://github.com/SemirBaldzhiev/ArduinoProject>