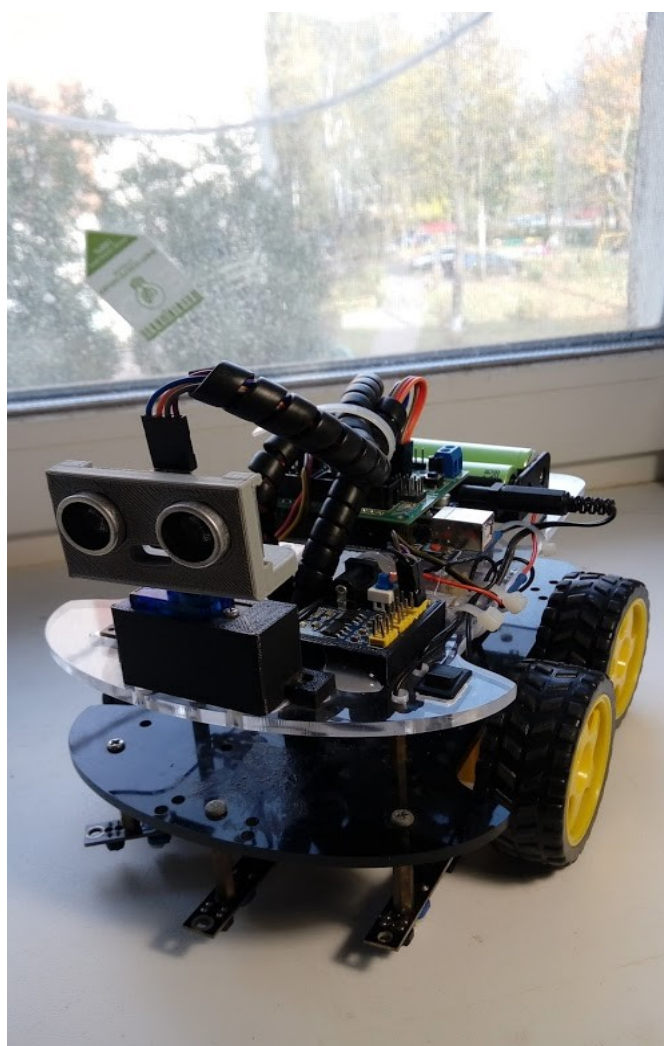


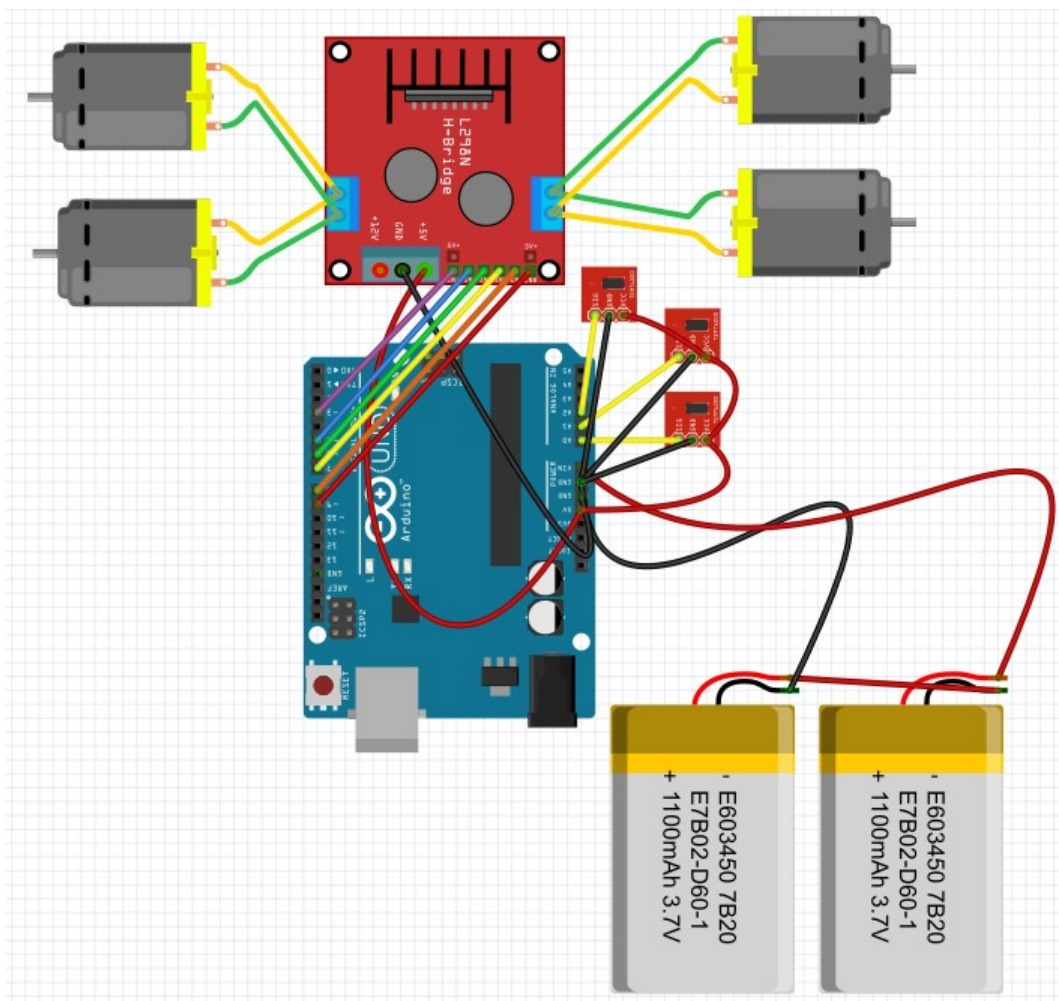
Робот «Следопыт»

Следопыт предназначен для движения по линии определённого цвета. Он может быть полезен на складах, где требуется непрерывное курсирование по нужному маршруту, с целью развоза небольших грузов определённого назначения. Это освободит людей на автокарах от монотонной работы.

Робот так же может использоваться для обучения программированию в среде Arduino. Конструкция робота позволяет проводить его модернизацию за счёт расширения номенклатуры подключаемых модулей.

Следопыт построен из недорогих модулей на базе микроконтроллера Arduino. Питание робота осуществляется от двух аккумуляторов типа 18650 с суммарным напряжением 7,4 вольта. Имеет 4 ведущих колеса с моторами постоянного тока, что позволяет роботу обрести курсовую устойчивость на гладких и пересечённых поверхностях. Для слежения по чёрной линии используются инфракрасные датчики отражения в количестве 3-х штук. Корпус робота изготовлен из акрила и пластика.





Паспорт изделия

- Название работы: Мобильный Универсальный Робот «Следопыт»
- Фамилия, имя, отчество возраст участника: Дорофеев Егор Дмитриевич, 11 лет.
- Номинация: «Робототехника»
- Ф.И.О. педагога Изотов Александр Александрович
- Полное название образовательного учреждения: Центр детского (юношеского) технического творчества "Юность"
- Территория (город или район): Сергиев Посад
- Размер работы (длина, ширина, высота): 29 мм./ 19 мм./ 11мм.

Программный код

```
// ----- //
// Black line movement for China car, biathlon. Egor D.
// V 1.0
// ----- //
#include <Arduino.h>

const int ENA = 3;
const int IN1 = 5;
const int IN2 = 6;
const int IN3 = 7;
const int IN4 = 8;
const int ENB = 9;
const int SENS_LEFT = 14;
const int SENS_CENTER = 15;
const int SENS_RIGHT = 16;

const int SPEED = 150;

void turnGo(int speed, int times);
void turnBack(int speed, int times);
void turnStop(int times);
void turnLeft(int speed, int times);
void turnRight(int speed, int times);
void sensTest(int times);

void setup() {
  Serial.begin(9600);
  pinMode(ENA, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT);
  pinMode(IN4, OUTPUT);
  pinMode(ENB, OUTPUT);
}

void loop() {
  do {
    bool sLeft = digitalRead(SENS_LEFT);
    bool sCenter = digitalRead(SENS_CENTER);
    bool sRight = digitalRead(SENS_RIGHT);
    if ((sLeft == 0) && (sCenter == 1) && (sRight == 0)) { // 010
```

```

    Serial.println("Go!");
    turnGo(SPEED, 10);
}
else if ((sLeft == 0) && (sCenter == 0) && (sRight == 0)) { // 000
    Serial.println("Go!");
    turnGo(SPEED, 10);
}
else if ((sLeft == 1) && (sCenter == 0) && (sRight == 0)) { // 100
    Serial.println("Left!");
    turnLeft(SPEED, 10);
}
else if ((sLeft == 1) && (sCenter == 1) && (sRight == 0)) { // 110
    Serial.println("Left!");
    turnLeft(SPEED, 10);
}
else if ((sLeft == 0) && (sCenter == 0) && (sRight == 1)) { // 001
    Serial.println("Right!");
    turnRight(SPEED, 10);
}
else if ((sLeft == 0) && (sCenter == 1) && (sRight == 1)) { // 011
    Serial.println("Right!");
    turnRight(SPEED, 10);
}
else {
    Serial.println("Left!");
    turnGo(SPEED, 70);
    turnLeft(SPEED, 100);
}
}
while (true);

//sensTest(100); // Для диагностики датчиков отражения
/*
turnGo(SPEED, 2000);
turnStop(2000);
turnBack(SPEED, 2000);
turnStop(2000);
turnLeft(SPEED, 2000);
turnStop(2000);
turnRight(SPEED, 2000);
turnStop(2000);
*/
}

```

```
void turnGo(int speed, int times) {  
    analogWrite(ENA, speed);  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    analogWrite(ENB, speed);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
    delay(times);  
}
```

```
void turnBack(int speed, int times) {  
    analogWrite(ENA, speed);  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    analogWrite(ENB, speed);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    delay(times);  
}
```

```
void turnStop(int times) {  
    analogWrite(ENA, 0);  
    analogWrite(ENB, 0);  
    delay(times);  
}
```

```
void turnLeft(int speed, int times) {  
    analogWrite(ENA, speed);  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    analogWrite(ENB, speed);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
    delay(times);  
}
```

```
void turnRight(int speed, int times) {  
    analogWrite(ENA, speed);  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    analogWrite(ENB, speed);  
    digitalWrite(IN3, LOW);  
}
```

```
digitalWrite(IN4, HIGH);  
delay(times);  
}
```

```
void sensTest(int times) {  
  bool sLeft = 0; bool sCenter = 0; bool sRight = 0;  
  Serial.print("Left: ");  
  Serial.print(sLeft = digitalRead(SENS_LEFT));  
  Serial.print(" ");  
  Serial.print("Center: ");  
  Serial.print(sCenter = digitalRead(SENS_CENTER));  
  Serial.print(" ");  
  Serial.print("Right: ");  
  Serial.println(sRight = digitalRead(SENS_RIGHT));  
  delay(times);  
}  
// ===== //  
// END FILE  
// ===== //
```