

2019

Робот-автомобиль Yahboom на базе Arduino

Руководство по программированию в Arduino IDE



Программирование

Содержание

1- Подготовка	2
2- Движение автомобиля	7
3- Эксперимент «причудливые движения»	12
4- Эксперимент с кнопкой и звуковым сигналом	19
5- Движение по треку	27
6- Управление по Bluetooth	36

1- Подготовка

Цель эксперимента:

Соберите робот-автомобиль в соответствии с инструкцией, установив два литиевых аккумулятора типа 14500. Выполните подключение в соответствии со схемой, после чего загрузите программу advance.ino. Включите питание на задней панели. После паузы длительностью 0,5 секунд устройство запустится.

Необходимые компоненты:

Робот-автомобиль на базе Arduino — 1 шт.

USB-кабель — 1 шт.



Код для эксперимента:

```
int Left_motor_back=9;    //ВХОД 1
int Left_motor_go=5;      //ВХОД 2

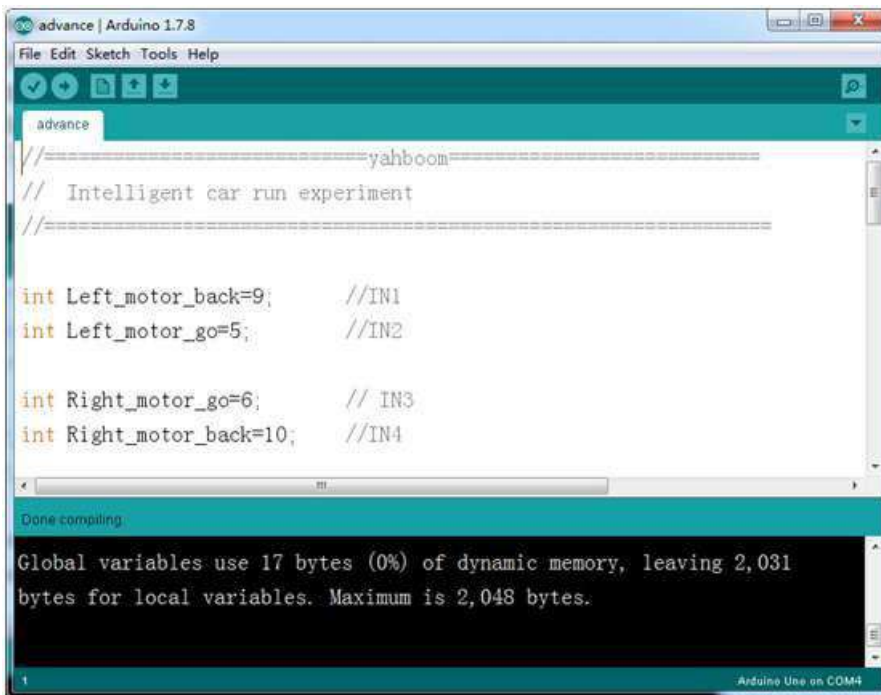
int Right_motor_go=6;     //ВХОД 3
int Right_motor_back=10;  //ВХОД 4
void setup()
{
    //инициализация выхода драйвера электродвигателей
    pinMode(Left_motor_go,OUTPUT); //КОНТ. 5 (ШИМ)
    pinMode(Left_motor_back,OUTPUT); //КОНТ. 9 (ШИМ)
    pinMode(Right_motor_go,OUTPUT); //КОНТ. 6 (ШИМ)
    pinMode(Right_motor_back,OUTPUT); //КОНТ. 10 (ШИМ)
}
void run(int time)
{
    digitalWrite(Right_motor_go,HIGH);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                     //небольшое увеличение/уменьшение разницы между правым //и
                                     левым колесом
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                     //небольшое увеличение/уменьшение разницы между левым //и
                                     правым колесом
    analogWrite(Left_motor_back,0);
    delay(time * 100); //время выполнения, регулируемое
}
```



```
void loop()
{
  delay(500);
  run(10);
}
```

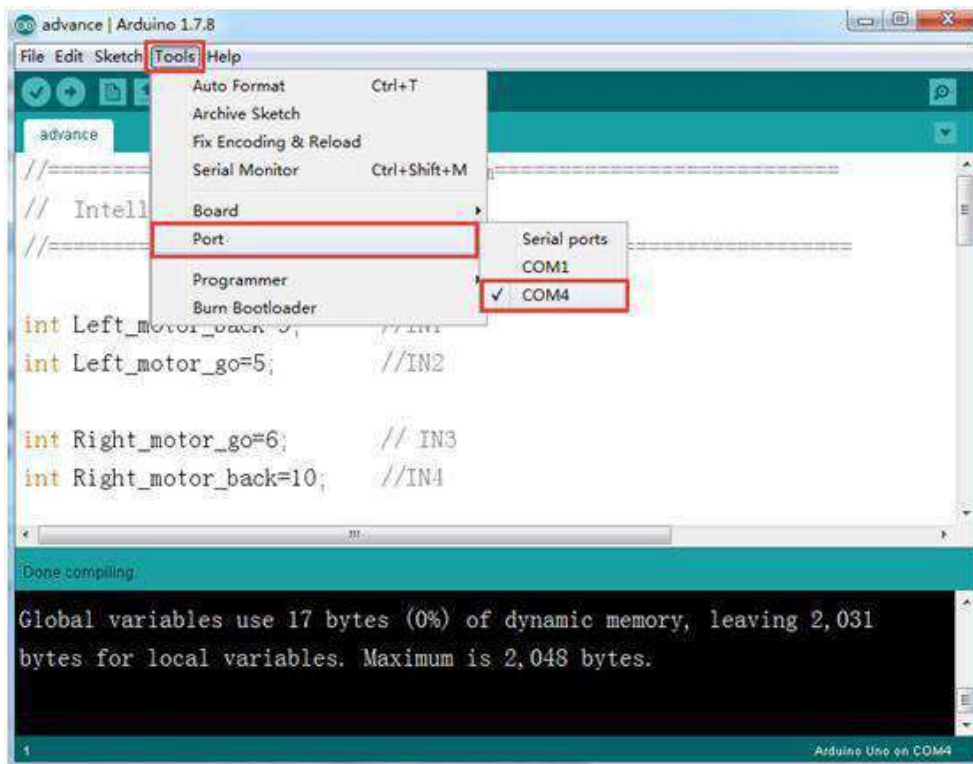
Этапы эксперимента:

1. Откройте файл с кодом эксперимента: **advance.ino**, нажмите “√” на панели меню для компиляции кода, дождитесь сообщения **"Done compiling"** (компиляция завершена) в нижнем правом углу (см. рисунок для примера).

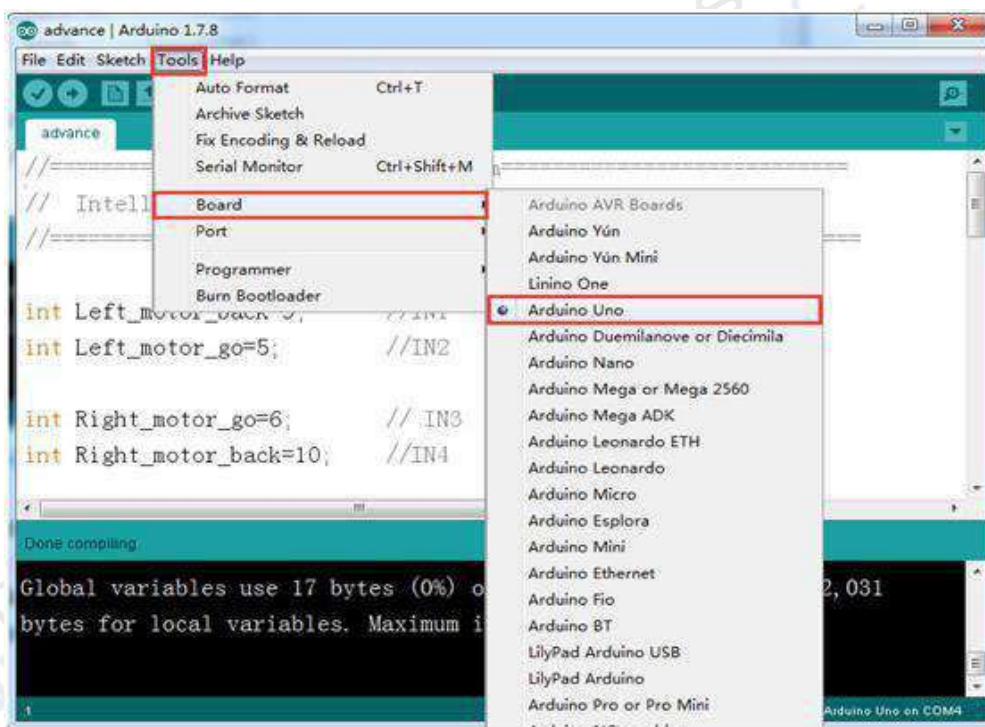


2. В меню программы Arduino IDE выберите пункт Tools】 --- 【Port】 ---, после чего выберите порт, соответствующий серийному номеру в Диспетчере устройств (пример на рисунке).

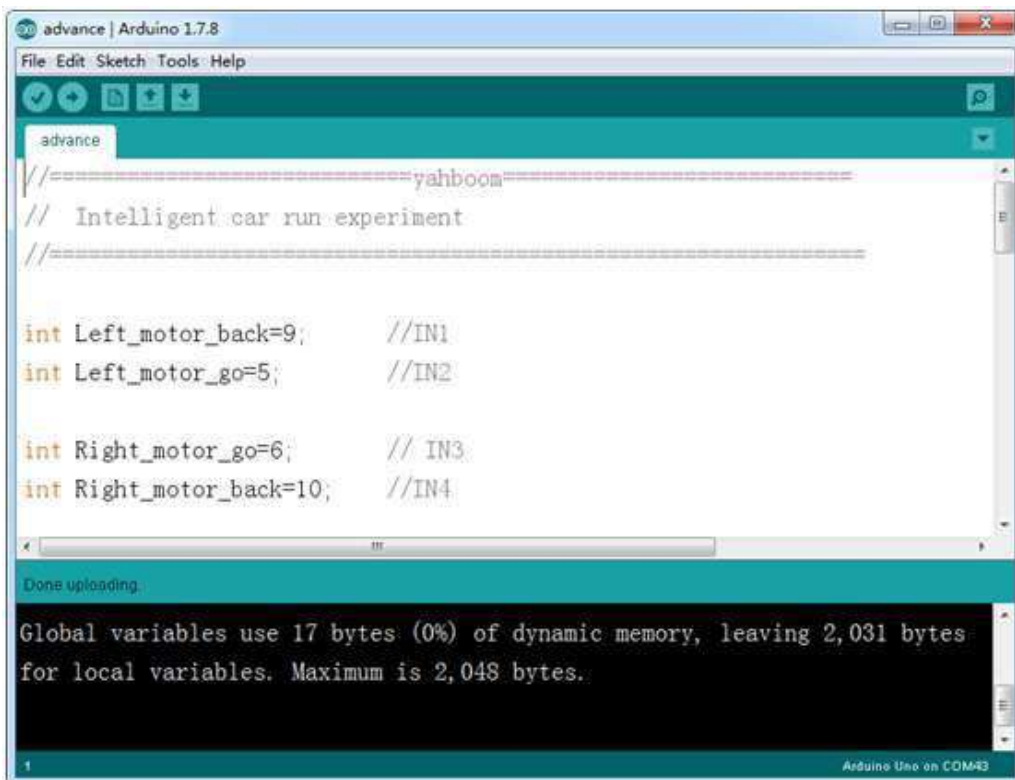




3. Нажмите [Tools]---[Board]---, после чего выберите пункт Arduino Uno.



4. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. В результате успешной загрузки кода в плату Arduino UNO в нижнем левом углу отображается сообщение “Done uploading” (загрузка завершена, см. рисунок).



```
advance | Arduino 1.7.8
File Edit Sketch Tools Help
advance
//=====yahboom=====
// Intelligent car run experiment
//=====

int Left_motor_back=9;      //IN1
int Left_motor_go=5;        //IN2

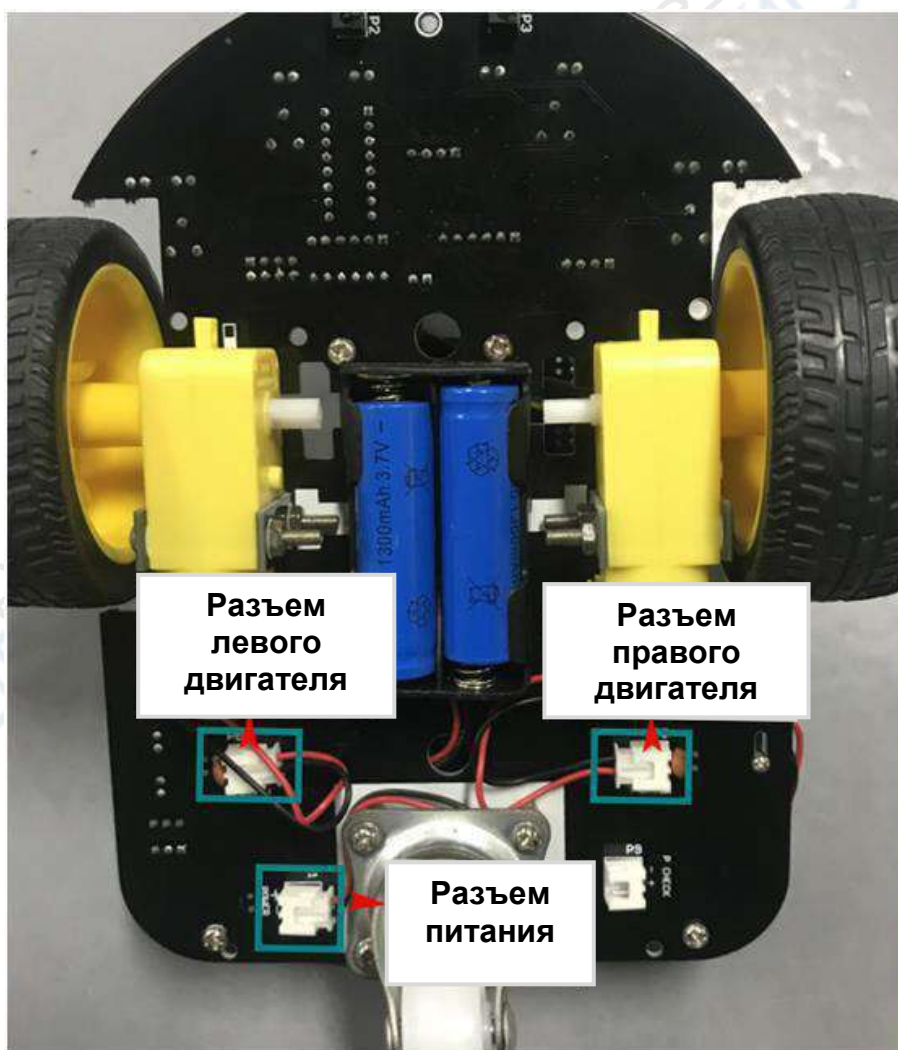
int Right_motor_go=6;       // IN3
int Right_motor_back=10;    //IN4

Done uploading.

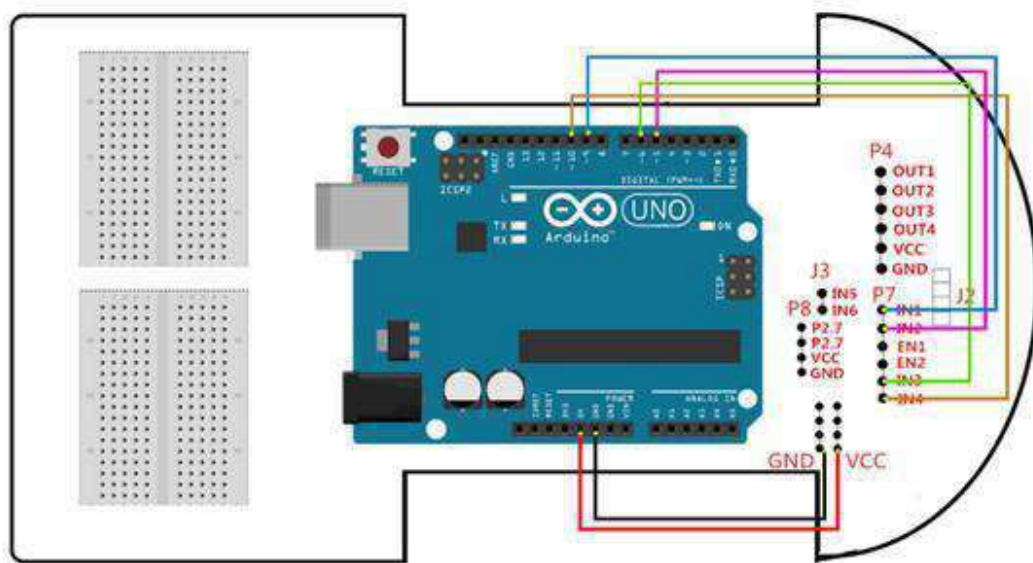
Global variables use 17 bytes (0%) of dynamic memory, leaving 2,031 bytes
for local variables. Maximum is 2,048 bytes.

Arduino Uno on COM43
```

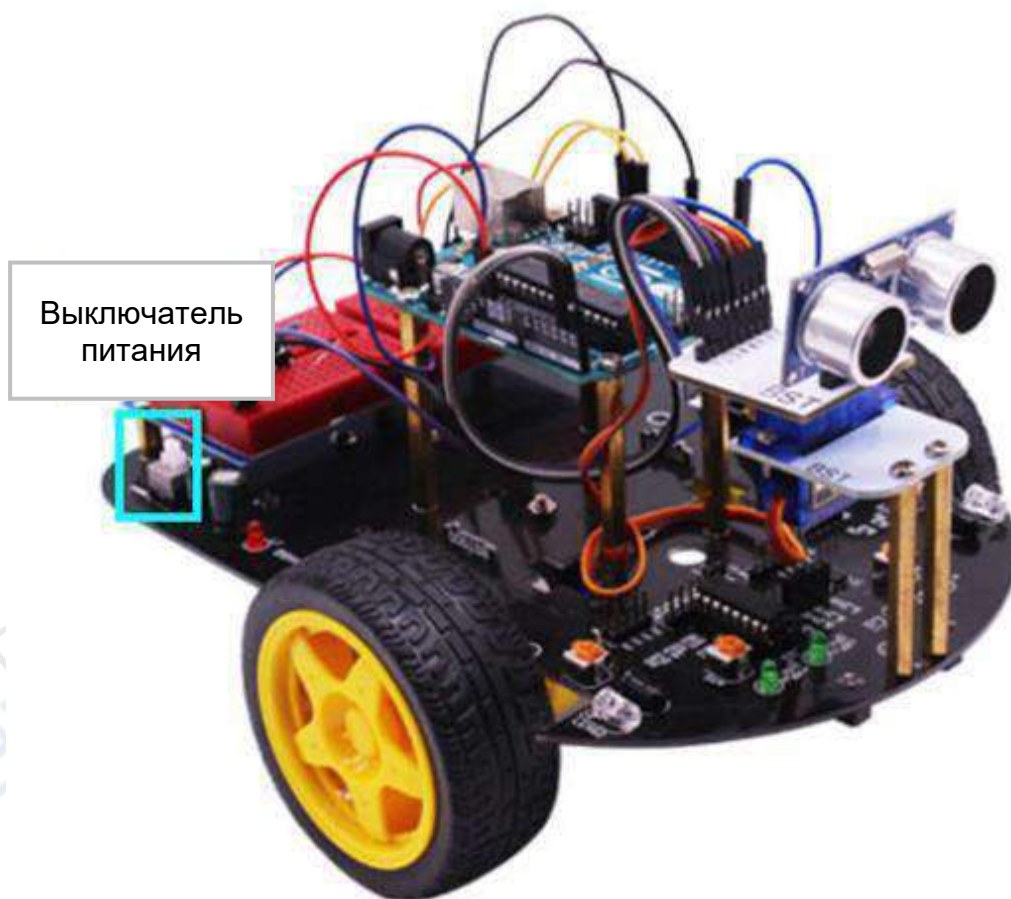
5. Выполните подключение в соответствии со следующим рисунком.



Электрическая схема подключения драйвера электродвигателей



6. После подключения проводов и загрузки программы включите питание на задней панели устройства. Автомобиль запустится через 0,5 секунды.



2- Движение автомобиля

Цель эксперимента:

После загрузки программы включите автомобиль с помощью выключателя, расположенного на задней панели. Через 2 секунды робот-автомобиль начнет двигаться назад, вперед, влево, вправо, развернется вправо и влево.

Необходимые компоненты:

Робот-автомобиль на базе Arduino — 1 шт.

USB-кабель — 1 шт.



Код для эксперимента:

```
int Left_motor_back=9;      //(ВХОД 1)
int Left_motor_go=5;        //(ВХОД 2)

int Right_motor_go=6;       //(ВХОД 3)
int Right_motor_back=10;    //(ВХОД 4)
void setup()
{
    //инициализация выхода драйвера электродвигателей
    pinMode(Left_motor_go,OUTPUT); //КОНТ. 5 (ШИМ)
    pinMode(Left_motor_back,OUTPUT); //КОНТ. 9 (ШИМ)
    pinMode(Right_motor_go,OUTPUT); //КОНТ. 6 (ШИМ)
    pinMode(Right_motor_back,OUTPUT); //КОНТ. 10 (ШИМ)
}
void run(int time)
{
    digitalWrite(Right_motor_go,HIGH);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                     //небольшое увеличение/уменьшение разницы между левым //и
                                     //правым колесом
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                     //небольшое увеличение/уменьшение разницы между левым
                                     //и правым колесом
    analogWrite(Left_motor_back,0);
    delay(time * 100); //время выполнения, регулируемое
}
void brake(int time)
```



```

{
digitalWrite(Right_motor_go,LOW);

digitalWrite(Right_motor_back,LOW);
digitalWrite(Left_motor_go,LOW);
digitalWrite(Left_motor_back,LOW);
delay(time * 100); //время выполнения, регулируемое
}
void left(int time) //поворот налево (левое колесо останавливается, правое //продолжает
                    вращаться)
{
digitalWrite(Right_motor_go,HIGH); //правый двигатель работает
digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,200);
analogWrite(Right_motor_back,0); //ШИМ-регулирование оборотов: 0~255,
digitalWrite(Left_motor_go,LOW); //левый двигатель останавливается
digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,0); // ШИМ-регулирование оборотов: 0~255
delay(time * 100); //время выполнения, регулируемое
}
void spin_left(int time) //левый разворот (левое колесо вращается в обратном
                        //направлении, правое колесо — в прямом)
{
digitalWrite(Right_motor_go,HIGH); //правый двигатель вращается в прямом
                                //направлении
digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,200);
analogWrite(Right_motor_back,0); //ШИМ-регулирование оборотов: 0~255
digitalWrite(Left_motor_go,LOW); //левый двигатель вращается в обратном
                                //направлении
digitalWrite(Left_motor_back,HIGH);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,200); //ШИМ-регулирование оборотов: 0~255
delay(time * 100); //время выполнения, регулируемое
}
void right(int time) //поворот направо (правое колесо останавливается, левое
                    //продолжает вращаться)
{
digitalWrite(Right_motor_go,LOW); //правый двигатель останавливается
digitalWrite(Right_motor_back,LOW);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,0); //ШИМ-регулирование оборотов: 0~255
digitalWrite(Left_motor_go,HIGH); //левый двигатель работает
digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,200);
analogWrite(Left_motor_back,0); //ШИМ-регулирование оборотов: 0~255
delay(time * 100); //время выполнения, регулируемое
}
void spin_right(int time) //правый разворот (правое колесо вращается в обратном
                        //направлении, левое — в прямом)
{

```

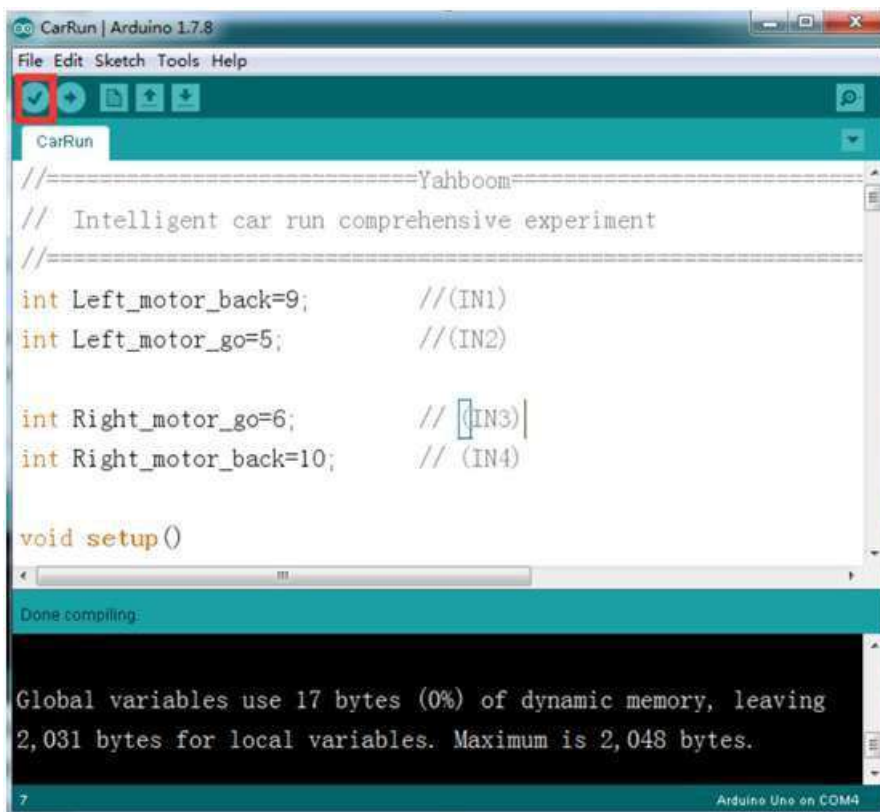
```

digitalWrite(Right_motor_go,LOW); //правый двигатель вращается в обратном
                                //направлении
digitalWrite(Right_motor_back,HIGH);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,200); //ШИМ-регулирование оборотов: 0~255
digitalWrite(Left_motor_go,HIGH); //левый двигатель работает
digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,200);
analogWrite(Left_motor_back,0);    //ШИМ-регулирование оборотов: 0~255
delay(time * 100);    //время выполнения, регулируемое}
void back(int time)    //движение назад
{
digitalWrite(Right_motor_go,LOW); //правый двигатель вращается в обратном
                                //направлении digitalWrite(Right_motor_back,HIGH);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,150); //ШИМ-регулирование оборотов: 0~255
digitalWrite(Left_motor_go,LOW); //левый двигатель вращается в обратном
                                //направлении
digitalWrite(Left_motor_back,HIGH);
analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,150); // ШИМ-регулирование оборотов: 0~255
delay(time * 100);    //время выполнения, регулируемое
}
void loop()
{
delay(2000); //запуск после паузы 2 сек
back(10); //движение назад 1 сек
brake(5); //остановка 0,5 сек
run(10); //движение вперед 1 сек
brake(5); //остановка 0,5 сек
left(10); //поворот налево 1 сек
right(10); //поворот направо 1 сек
spin_right(20); //разворот направо 2 сек
spin_left(20); //разворот налево 2 сек
brake(5); //стоп
}

```

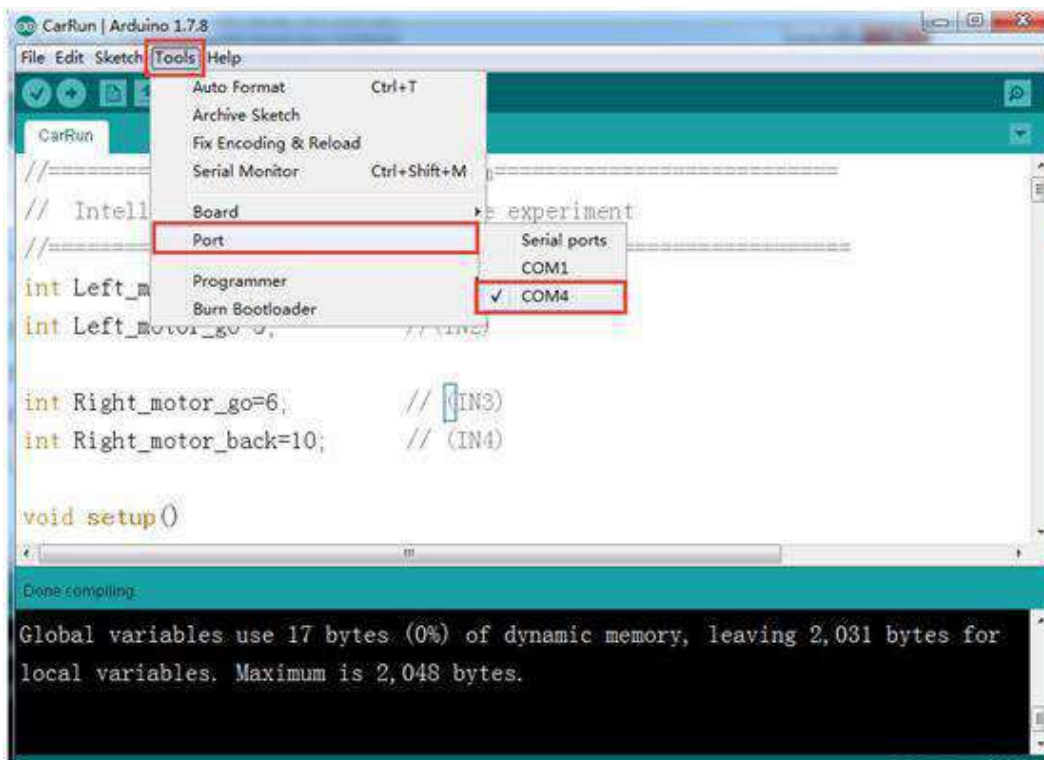
Этапы эксперимента:

1. Откройте файл с кодом эксперимента: **CarRun.ino**, нажмите кнопку“√”на панели меню для компиляции кода, дождитесь сообщения **"Done compiling"** (компиляция завершена) в нижнем правом углу (см.рисунок для примера).

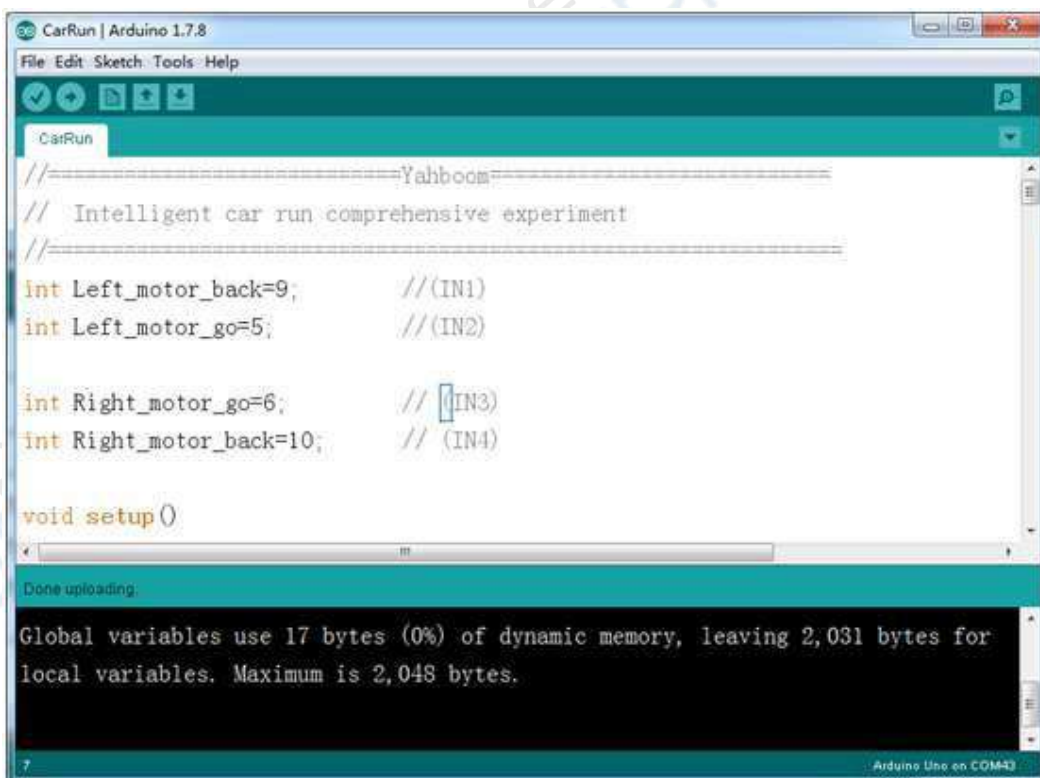


2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).



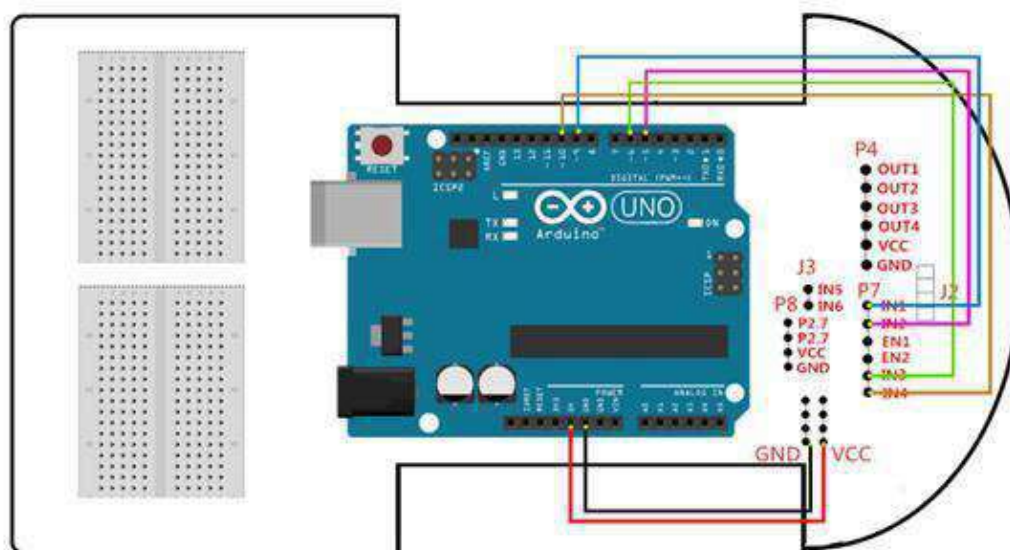


3. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. После успешной загрузки кода в плату в нижнем левом углу отобразится сообщение “**Done uploading**” (загрузка завершена, см. рисунок).



4. Выполните подключение в соответствии со следующим рисунком.

Электрическая схема подключения драйвера электродвигателей



5. Отключите USB-кабель, поставьте робот-автомобиль на просторную площадку и включите питание. После двухсекундной паузы автомобиль поедет назад, затем вперед, повернет налево, направо и т.д.



3- Эксперимент «причудливые движения»

Цель эксперимента:

После загрузки программы и включения робота-автомобиля он осуществляет запрограммированное движение по истечении двухсекундной паузы.

Необходимые компоненты:

Робот-автомобиль на базе Arduino — 1 шт.

USB-кабель — 1 шт.



Код для эксперимента:

```
//=====yahboom=====
// Эксперимент с набором причудливых движений робота-автомобиля
//=====
int Left_motor_back=9;    //(ВХОД 1)
int Left_motor_go=5;      //(ВХОД 2)
int Right_motor_go=6;     //(ВХОД 3)
int Right_motor_back=10;  //(ВХОД 4)
void setup()
{
    //инициализация выхода драйвера электродвигателей
    pinMode(Left_motor_go,OUTPUT); //КОНТ. 5 (ШИМ)
    pinMode(Left_motor_back,OUTPUT); //КОНТ. 9 (ШИМ)
    pinMode(Right_motor_go,OUTPUT); //КОНТ. 6 (ШИМ)
    pinMode(Right_motor_back,OUTPUT); //КОНТ. 10 (ШИМ)
}
void run(int time)    //автомобиль движется
{
    digitalWrite(Right_motor_go,HIGH);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                     //небольшое увеличение/уменьшение разницы между левым //и
                                     правым колесом
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200); // ШИМ-регулирование оборотов: 0~255,
                                     //небольшое увеличение/уменьшение разницы между левым //и
                                     правым колесом
    analogWrite(Left_motor_back,0);
    delay(time * 100); //время выполнения, регулируемое
```



```

}

void brake(int time)      //остановка автомобиля
{
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,LOW);
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,LOW);
    delay(time * 100);// время выполнения, регулируемое
}

void left(int time)        //поворот налево (левое колесо
                           //останавливается, правое продолжает вращаться)
{
    digitalWrite(Right_motor_go,HIGH);    //правый электродвигатель вращается
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);      //левое колесо вращается в обратном
                                           //направлением
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    (time * 100);      //время выполнения, регулируемое
}

void spin_left(int time)    //разворот налево (левое колесо вращается в
                           //обратном направлении, правое — в прямом)
{
    digitalWrite(Right_motor_go,HIGH);    //правый электродвигатель вращается
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);      //левое колесо вращается в обратном
                                           //направлении
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,200);    //ШИМ-регулирование оборотов: 0~255
    delay(time * 100);      //время выполнения, регулируемое
}

void right(int time)        //поворот направо(правое колесо останавливается,
                           //левое продолжает движение)
{
    digitalWrite(Right_motor_go,LOW);      //правый двигатель вращается в обратном
                                           //направлении
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,HIGH);      //левый двигатель вращается в прямом
                                           //направлении
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);
    analogWrite(Left_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
}

```

```

    delay(time * 100);          //время выполнения, регулируемое
}

void spin_right(int time)      //правый разворот (правое колесо вращается в //обратном
                               направлении, левое колесо — в прямом)
{
    digitalWrite(Right_motor_go,LOW);    //правый двигатель вращается в
                                           //обратном направлении

    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,200); //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,HIGH);    //левый двигатель вращается в
                                           //обратном направлении

    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);
    analogWrite(Left_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    delay(time * 100);                  //время выполнения, регулируемое
}

void back(int time)           //задний ход
{
    digitalWrite(Right_motor_go,LOW);    //правое колесо вращается в
                                           //обратном направлении

    digitalWrite(Right_motor_back,HIGH);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,150); //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);     //правое колесо вращается в
                                           //обратном направлении

    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,150); //ШИМ-регулирование оборотов: 0~255
    delay(time * 100);                  //время выполнения, регулируемое
}

void loop()
{
    int i;
    delay(2000); //запуск после паузы 2 секунды
    run(10);
    back(10);
    brake(5);
    for(i=0;i<5;i++)
    {
        run(10); //пятикратное прерывистое движение автомобиля вперед
        brake(1);
    }
    for(i=0;i<5;i++)
    {
        back(10); //пятикратное прерывистое движение автомобиля назад
        brake(1);
    }
    for(i=0;i<5;i++)
    {
        left(10); //большая серия непрерывного вращения
    }
}

```

```

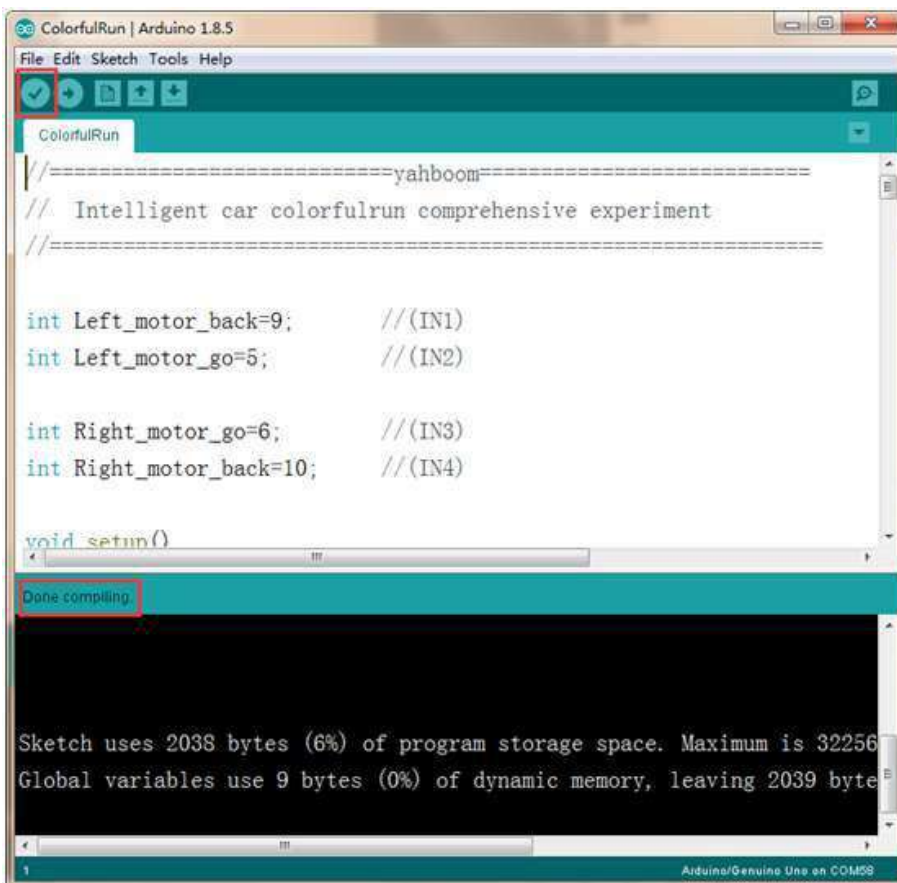
        //влево
    spin_left(5);
}
for(i=0;i<5;i++)
{
    right(10); //большая серия непрерывного вращения
               //вправо

    spin_right(5);
}
for(i=0;i<10;i++)
{
    right(1); //прерывистый поворот вправо
    brake(1);
}
for(i=0;i<10;i++)
{
    left(1); //прерывистый поворот влево
    brake(1);
}
for(i=0;i<10;i++)
{
    left(3); //движение вперед змейкой
    right(3);
}
for(i=0;i<10;i++)
{
    spin_left(3); //дискретное вращение влево
    brake(3);
}
for(i=0;i<10;i++)
{
    spin_right(3); //дискретное вращение вправо
    brake(3);
}
}
}

```

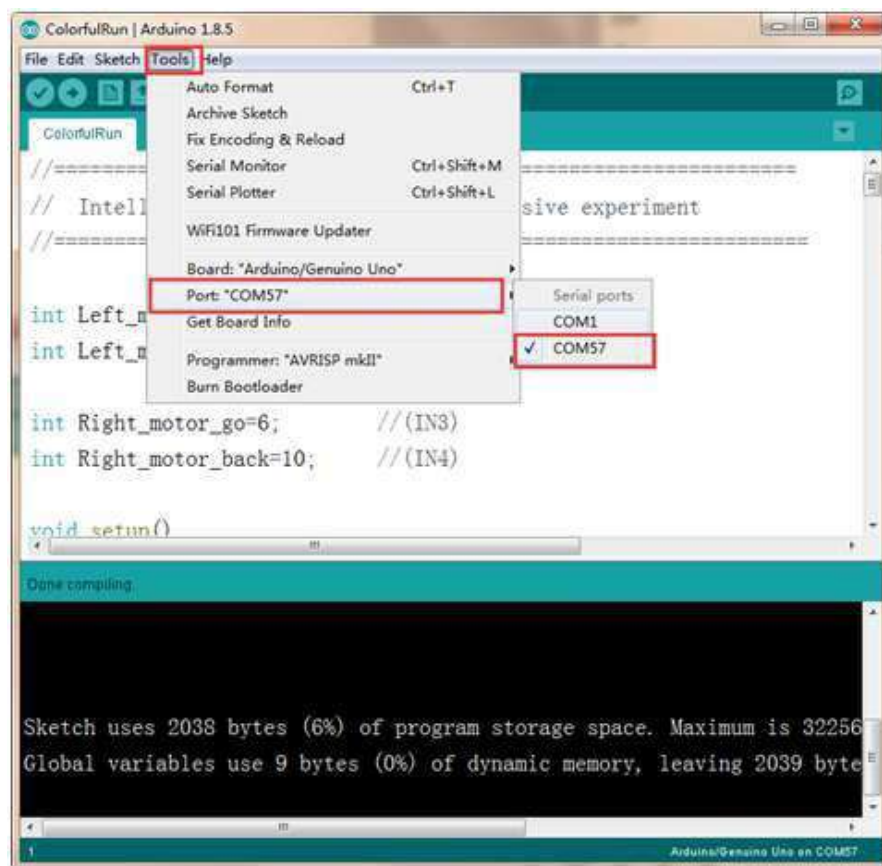
Этапы эксперимента:

1. Откройте файл с кодом эксперимента: **ColorfulRun.ino**, нажмите кнопку “√” под панелью меню для компиляции кода, дождитесь сообщения **"Done compiling "** (компиляция завершена) в нижнем правом углу (см. рисунок для примера).

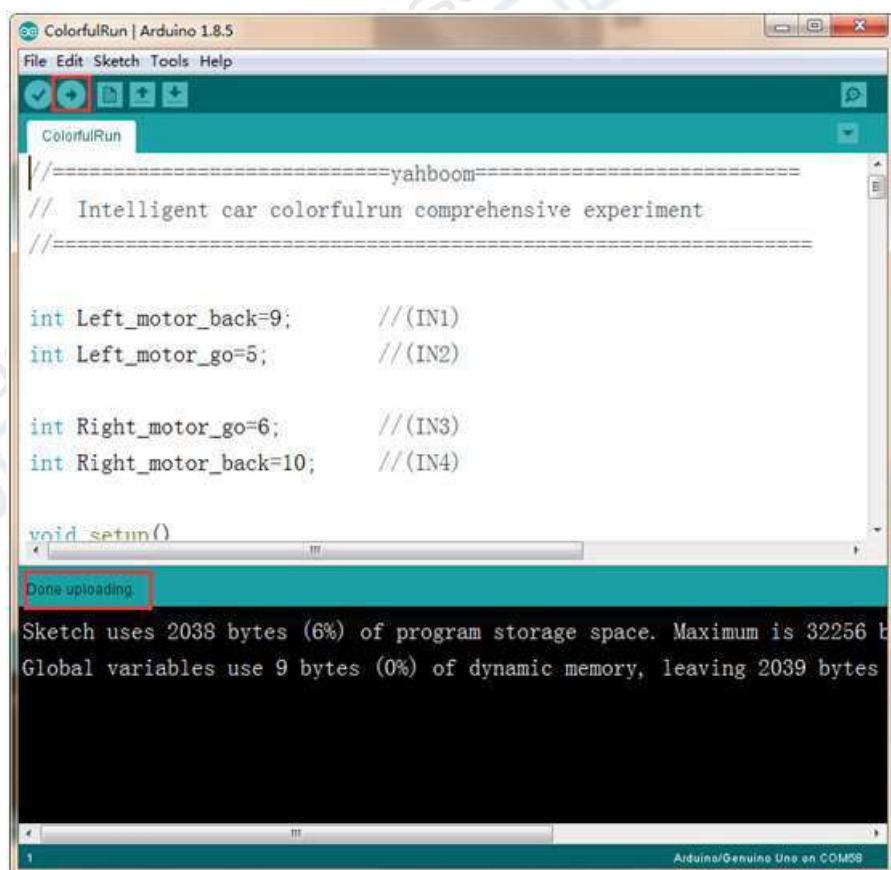


2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).



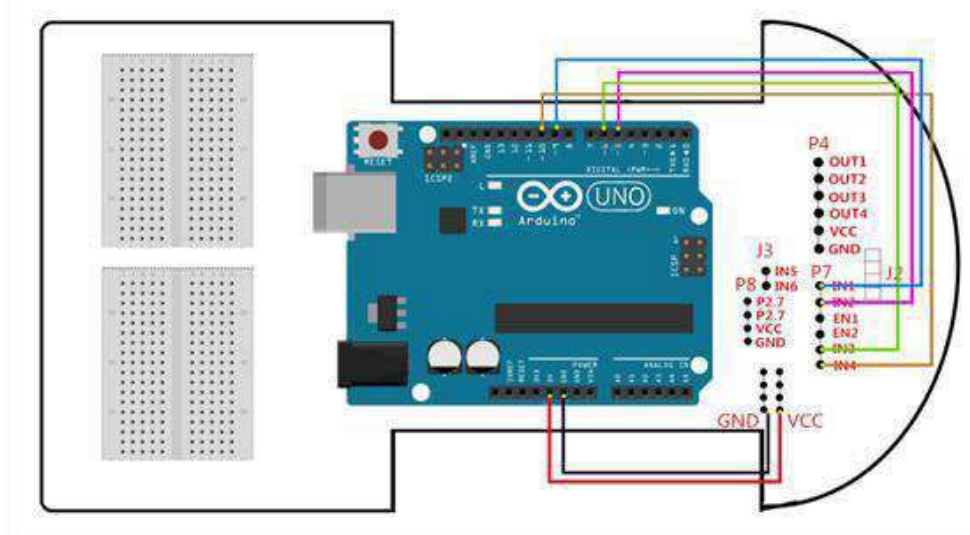


3. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. После успешной загрузки кода в плату Arduino UNO в нижнем левом углу отображается сообщение “**Done uploading**” (загрузка завершена, см. рисунок).

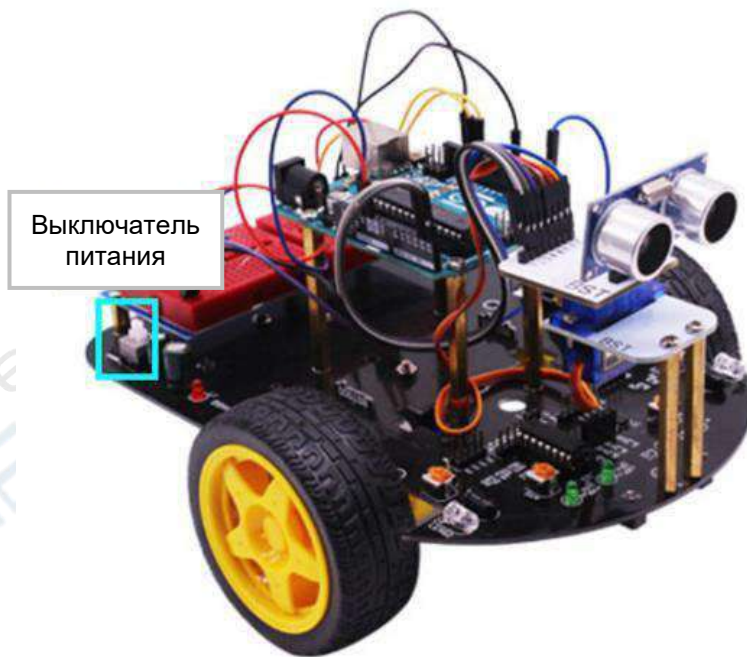


4. Выполните подключение в соответствии со следующим рисунком.

Электрическая схема подключения драйвера электродвигателей



5. Отключите USB-кабель, поставьте умный автомобиль на просторную площадку и включите питание. После двухсекундной паузы автомобиль выполнит запрограммированные движения.



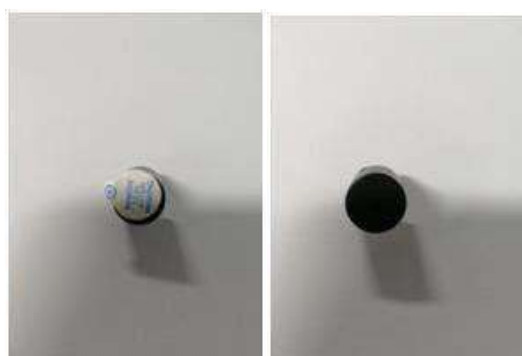
4- Эксперимент с кнопкой и звуковым сигналом

Цель эксперимента:

После загрузки программы keyssacnStartCa.ino разместите автомобиль на просторной площадке и включите его питание. Автомобиль остается неподвижным. После нажатия на кнопку запуска автомобиль начинает выполнять запрограммированные движения с коротким звуковым сигналом.

Предостережения:

1. А — активное сигнальное устройство, В — пассивное сигнальное устройство. В эксперименте используется активное сигнальное устройство.



А

В

Необходимые компоненты:

Робот-автомобиль на базе Arduino — 1 шт.

USB-кабель — 1 шт.

Активное сигнальное устройство — 1 шт.

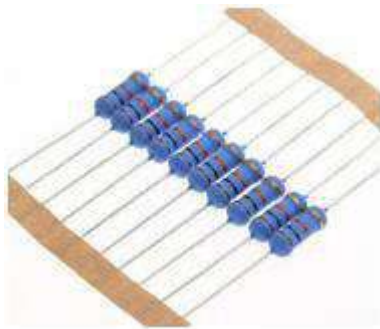
Провод DuPont — 1 шт.

Монтажная плата — 1 шт.

Кнопка — 1 шт.

Резистор 10 кОм — 1 шт.





Код для эксперимента:

```
//=====yahboom=====
// Эксперимент с кнопкой и звуковым сигналом
//=====
int Left_motor_back=9;    //(ВХОД 1)
int Left_motor_go=5;     //(ВХОД 2)
int Right_motor_go=6;    //(ВХОД 3)
int Right_motor_back=10; //(ВХОД 4)
int key=A0; //определение интерфейса кнопки A0
int beep=A1; //определение интерфейса сигнального устройства A1
void setup()
{
    //инициализация выхода драйвера электродвигателей
    pinMode(Left_motor_go,OUTPUT); //КОНТ. 5 (ШИМ)
    pinMode(Left_motor_back,OUTPUT); //КОНТ. 9 (ШИМ)
    pinMode(Right_motor_go,OUTPUT); //КОНТ. 6 (ШИМ)
    pinMode(Right_motor_back,OUTPUT); //КОНТ. 10 (ШИМ)
    pinMode(key,INPUT); //определение интерфейса кнопки для
                        //получения входного сигнала
    pinMode(beep,OUTPUT);
}
void run(int time) //движение вперед
{
    digitalWrite(Right_motor_go,HIGH);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                    //небольшое увеличение/уменьшение разницы между левым //и
                                    правым колесом
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200); //ШИМ-регулирование оборотов: 0~255,
                                    //небольшое увеличение/уменьшение разницы между левым //и
                                    правым колесом
    analogWrite(Left_motor_back,0);
    delay(time * 100); //время выполнения, регулируемое
}

void brake(int time) //остановка автомобиля
{

```

```

digitalWrite(Right_motor_go,LOW);
digitalWrite(Right_motor_back,LOW);
digitalWrite(Left_motor_go,LOW);
digitalWrite(Left_motor_back,LOW);

delay(time * 100); //время выполнения, регулируемое
}
void left(int time)      //поворот налево (левое колесо останавливается,правое
                        //колесо продолжает движение)
{
    digitalWrite(Right_motor_go,HIGH);    //правый двигатель вращается в прямом
                                           //направлении

    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);      //левое колесо вращается в обратном
                                           //направлении

    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    delay(time * 100); //время выполнения, регулируемое
}
void spin_left(int time) //разворот влево (левое колесо вращается в обратном
                        //направлении, правое колесо — в прямом)
{
    digitalWrite(Right_motor_go,HIGH);    //правый двигатель вперед
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);      //левое колесо назад
    digitalWrite(Left_motor_back,HIGH);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,200);     //ШИМ-регулирование оборотов: 0~255
    delay(time * 100); //время выполнения, регулируемое
}
void right(int time)     //поворот направо (правое колесо останавливается, левое
                        //продолжает движение)
{
    digitalWrite(Right_motor_go,LOW);     //правый двигатель вращается в прямом
                                           //направлении

    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,HIGH);     //левый двигатель вращается в прямом
                                           //направлении

    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,200);
    analogWrite(Left_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
    delay(time * 100); //время выполнения, регулируемое
}
void spin_right(int time) //правый разворот (правое колесо вращается в обратном

```

```

        //направлении, левое колесо — в прямом)
    {
        digitalWrite(Right_motor_go,LOW);    //правый двигатель вращается в обратном
                                              //направлении
        digitalWrite(Right_motor_back,HIGH);
        analogWrite(Right_motor_go,0);
        analogWrite(Right_motor_back,200); //ШИМ-регулирование оборотов: 0~255
        digitalWrite(Left_motor_go,HIGH);    //левый двигатель вращается в прямом
                                              //направлении
        digitalWrite(Left_motor_back,LOW);
        analogWrite(Left_motor_go,200);
        analogWrite(Left_motor_back,0);      //ШИМ-регулирование оборотов: 0~255
        delay(time * 100);                  //время выполнения, регулируемое
    }
    void back(int time)                    //движение назад
    {
        digitalWrite(Right_motor_go,LOW);    //правое колесо вращается в обратном
                                              //направлении
        digitalWrite(Right_motor_back,HIGH);
        analogWrite(Right_motor_go,0);
        analogWrite(Right_motor_back,150); //ШИМ-регулирование оборотов: 0~255
        digitalWrite(Left_motor_go,LOW); //левое колесо вращается в обратном
                                              //направлении
        digitalWrite(Left_motor_back,HIGH);
        analogWrite(Left_motor_go,0);
        analogWrite(Left_motor_back,150); //ШИМ-регулирование оборотов: 0~255
        delay(time * 100);                  //время выполнения, регулируемое
    }
    void keysacn()
    {
        int val;
        val=digitalRead(key); //считывание значения уровня порта 7 в val
        while(!digitalRead(key)) //когда кнопка не нажата, цикл запущен
        {
            val=digitalRead(key); //при пропуске данной строчки — выход из цикла
        }
        while(digitalRead(key)) //если кнопка нажата
        {
            delay(10);
            val=digitalRead(key); //считывание значения уровня порта 7 в val
            if(val==HIGH)        //определение повторного нажатия кнопки
            {
                digitalWrite(beep,HIGH);    //звуковой сигнал
                while(!digitalRead(key))    //определение отпускания кнопки
                {
                    digitalWrite(beep,LOW); //отключение сигнала
                }
            }
            else
                digitalWrite(beep,LOW);     //отключение сигнала
        }
    }

```



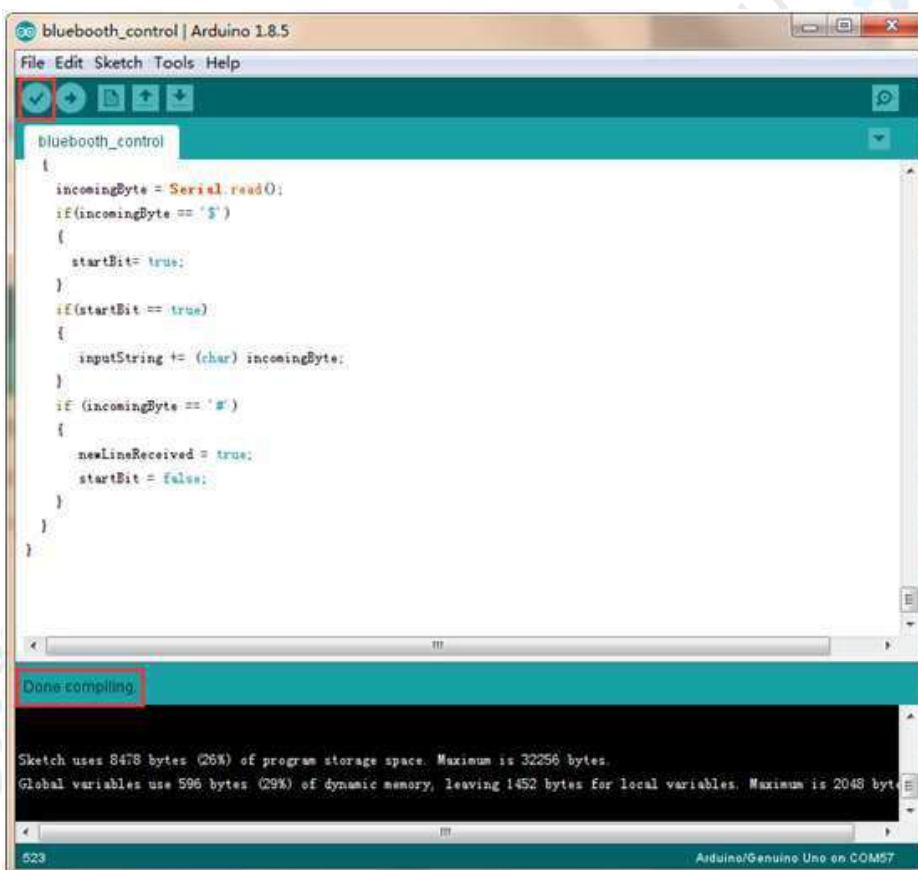
```

}
void loop()
{
  keysacn();    //вызов функции определения состояния кнопки
  back(10);     //назад 1 сек
  brake(5);     //стоп 0,5 сек
  run(10);      //вперед 1 сек
  brake(5);     //стоп 0,5 сек
  left(10);     //поворот налево 1 сек
  right(10);    // поворот направо 1 сек
  spin_left(20); //разворот влево 2 сек
  spin_right(20); // разворот вправо 2 сек
  brake(5);     //стоп
}

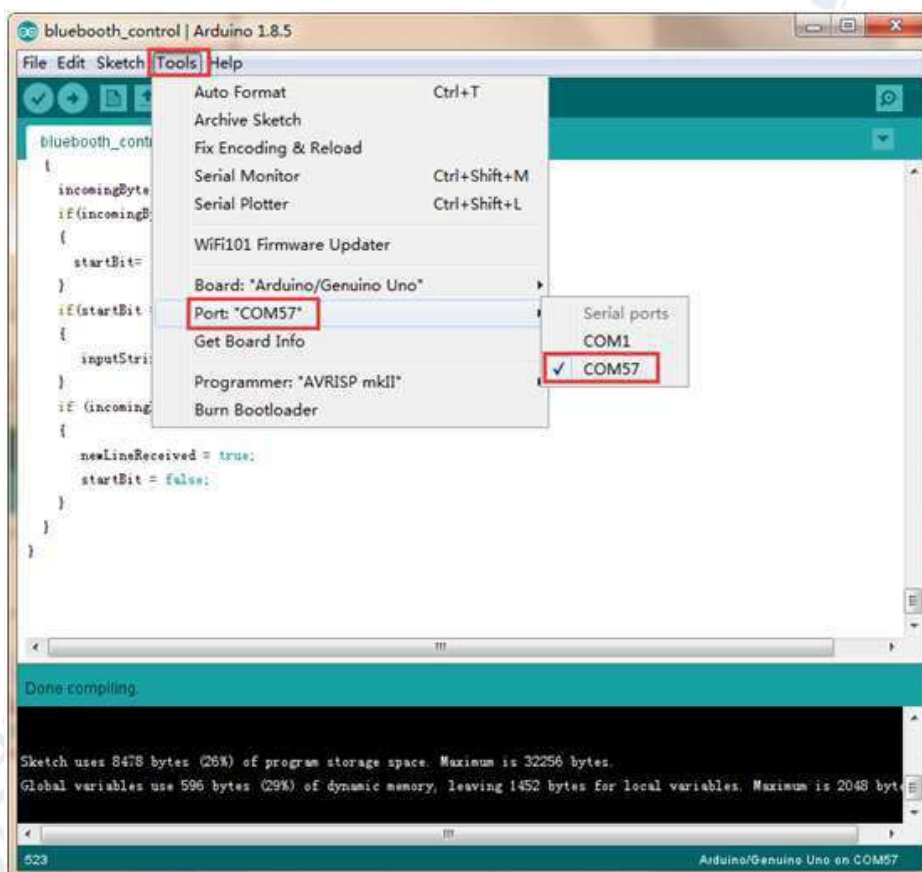
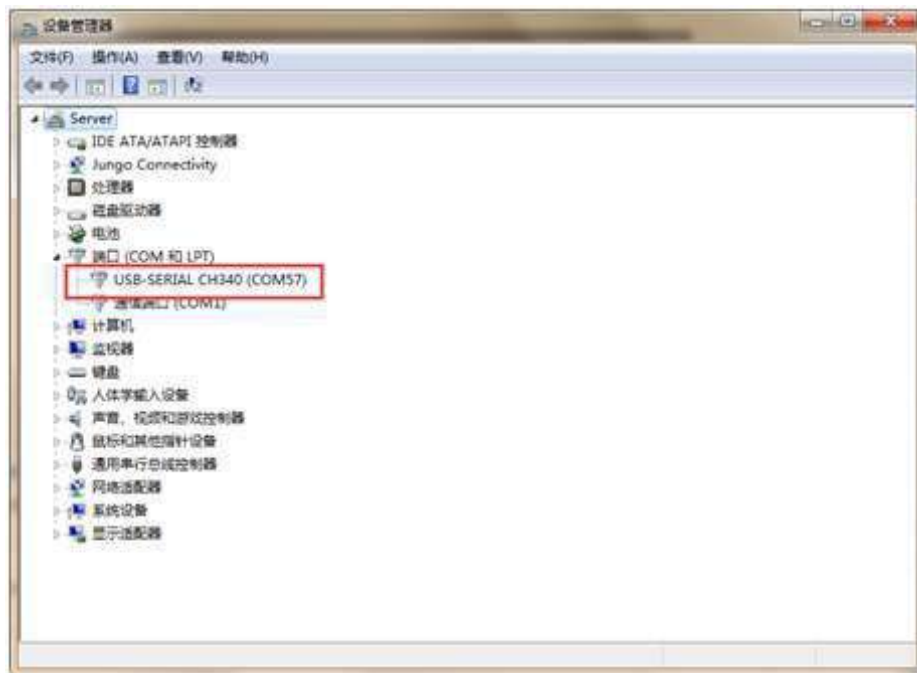
```

Этапы эксперимента:

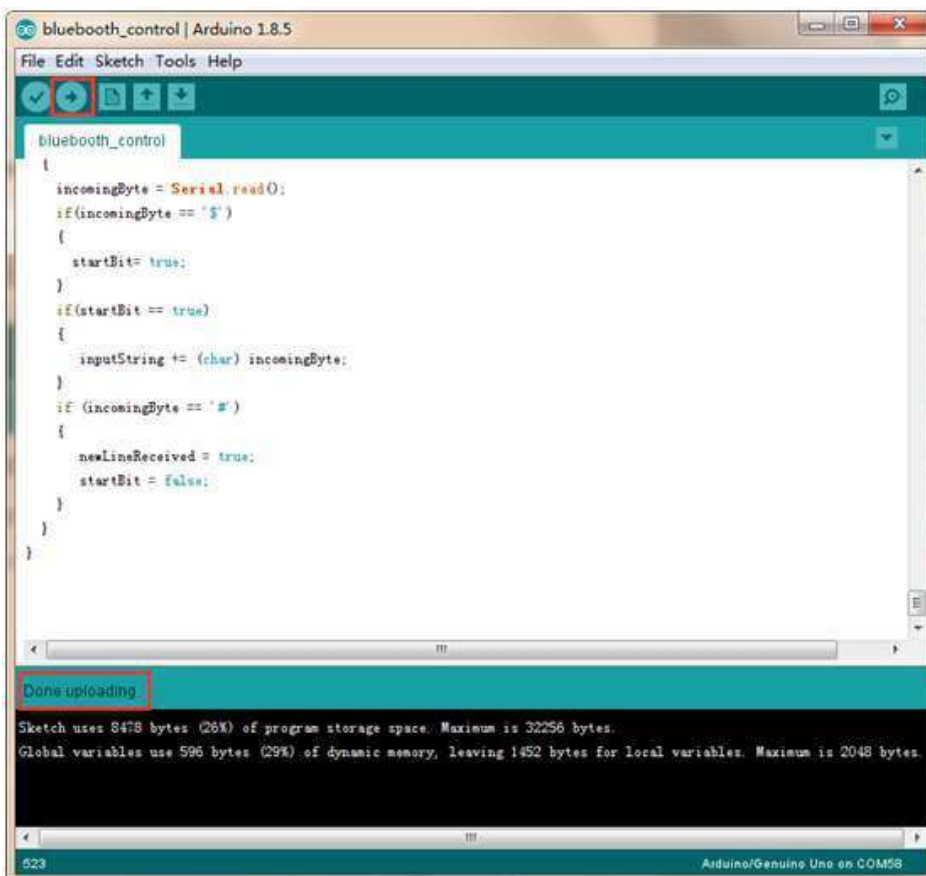
1. Откройте файл с кодом эксперимента: **keysacnStartCar.ino**, нажмите кнопку “✓” под панелью меню для компиляции кода, дождитесь сообщения **"Done compiling"** (компиляция завершена) в нижнем правом углу (см. рисунок для примера).



2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).



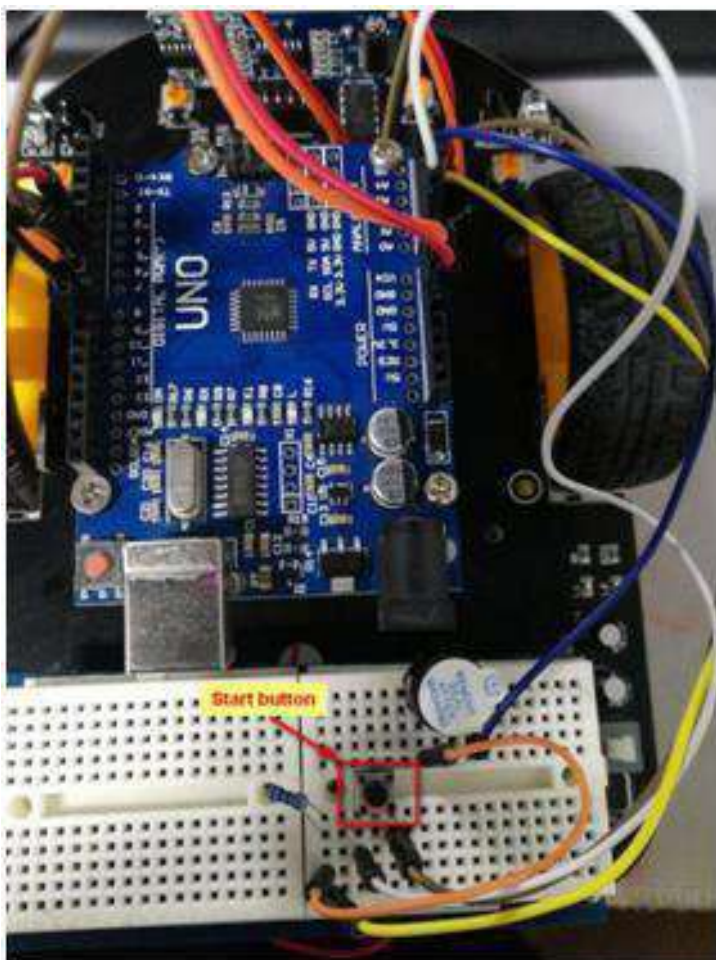
3. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. После успешной загрузки кода в плату Arduino UNO в нижнем левом углу отобразится сообщение “**Done uploading**” (загрузка завершена, см. рисунок).



4. Выполните подключение в соответствии со следующим рисунком.



5. Отключите USB-кабель, поставьте робот-автомобиль на просторную площадку и включите питание. После нажатия на кнопку запуска автомобиль выполнит запрограммированные движения с коротким звуковым сигналом.



5- Движение по треку

Цель эксперимента:

Разместите автомобиль на треке и нажмите кнопку запуска. Автомобиль издаст короткий звуковой сигнал и начнет двигаться по треку.

Внимание:

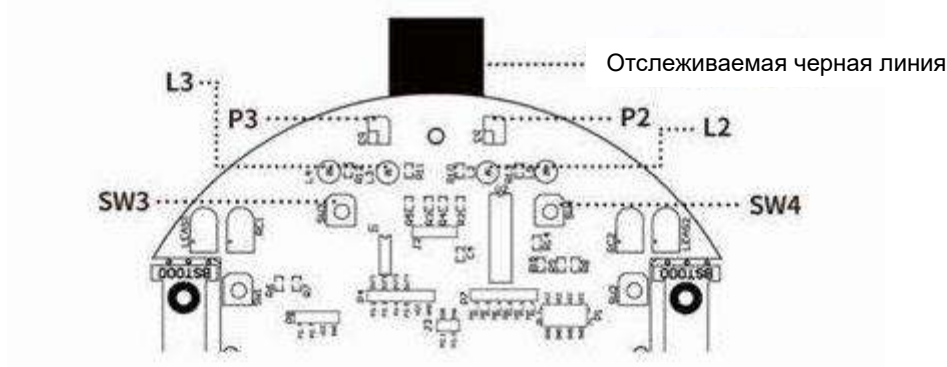
1. Перед выполнением эксперимента необходимо подстроить резисторы sw3 и sw4 в передней части автомобиля в соответствии с нижеприведенной инструкцией.

Режим ИК-сканирования

А) Отрегулируйте потенциометр [SW3] таким образом, чтобы излучение светодиода [L3] регистрировалось при нахождении фотодатчика [P3] над белой поверхностью и не регистрировалось при нахождении [P3] над черной поверхностью.

Б) Отрегулируйте потенциометр [SW4] таким образом, чтобы излучение светодиода [L2] регистрировалось при нахождении фотодатчика [P2] над белой поверхностью и не регистрировалось при нахождении [P2] над черной поверхностью.

Внимание: при регулировке необходимо плавно перемещать ручку потенциометра, регулировка осуществляется в пределах приблизительно 30°.



2. Эксперимент необходимо проводить в месте без естественного освещения, или занавесить окна, чтобы предотвратить проникновение света с улицы.

Необходимые компоненты:

Робот-автомобиль на базе Arduino — 1 шт.

USB-кабель — 1 шт.

Активное сигнальное устройство — 1 шт.

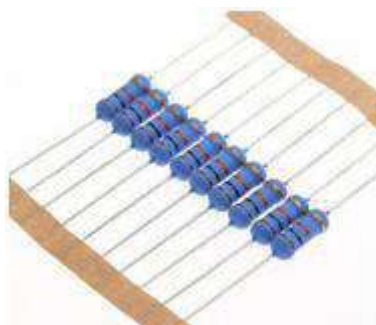
Провод DuPont — 13 шт.

Монтажная плата — 1 шт.

Кнопка — 1 шт.

Изолента шириной около 1,6 см — 1 шт.

Резистор 10 кОм — 1 шт.



Код для эксперимента:

```
//=====yahboom=====
// Эксперимент с движением по треку
//=====
int Left_motor_back=9;    //(ВХОД 1)
int Left_motor_go=5;     //(ВХОД 2)
int Right_motor_go=6;    //(ВХОД 3)
int Right_motor_back=10; //(ВХОД 4)
int key=A0;//определение интерфейса кнопки A0
int beer=A1;//определение интерфейса сигнального устройства A1
```

```

const int SensorRight = A2;      //правый ИК-датчик (P3.2 OUT1)
const int SensorLeft = A3;      //левый ИК-датчик(P3.3 OUT2)
int SL;    //состояние левого ИК-датчика
int SR;    //состояние правого ИК-датчика void
setup()
{
    //инициализация выхода драйвера электродвигателей
    pinMode(Left_motor_go,OUTPUT);    //КОНТ. 5 (ШИМ)

    pinMode(Left_motor_back,OUTPUT); //КОНТ. 9 (ШИМ)
    pinMode(Right_motor_go,OUTPUT); //КОНТ. 6 (ШИМ)
    pinMode(Right_motor_back,OUTPUT); //КОНТ. 10 (ШИМ)
    pinMode(key,INPUT);    //определение интерфейса кнопки для
                          //подачи входного сигнала

    pinMode(beep,OUTPUT);
    pinMode(SensorRight, INPUT); //определение правого ИК-датчика для
                          //подачи входного сигнала

    pinMode(SensorLeft, INPUT); //определение левого ИК-датчика для
                          //подачи входного сигнала
}
//=====Основное действие автомобиля=====
//void run(int time)
void run()
{
    digitalWrite(Right_motor_go,HIGH); //правый двигатель вращается в обратном
                          //направлении

    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,150);// //ШИМ-регулирование оборотов: 0~255,
                          //небольшое увеличение/уменьшение разницы между правым //и
                          левым колесом
    analogWrite(Right_motor_back,0);
    digitalWrite(Left_motor_go,HIGH);    //левый двигатель вращается в прямом
                          //направлении
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,150); // ШИМ-регулирование оборотов: 0~255,
                          //небольшое увеличение/уменьшение разницы между левым //и
                          правым колесом
    analogWrite(Left_motor_back,0);
    delay(time * 100);    //время выполнения, регулируемое
}
//void brake(int time)
void brake()
{
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,LOW);
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,LOW);
    delay(time * 100);    //время выполнения, регулируемое
}
//void left(int time)
void left()    //поворот налево (левое колесо останавливается, правое продолжает

```

```

движение в прямом направлении)
{
    digitalWrite(Right_motor_go,HIGH);    //правый двигатель вращается в прямом
                                           //направлении

    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,150);
    analogWrite(Right_motor_back,0);    //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,LOW);
    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,0);    //ШИМ-регулирование оборотов: 0~255
    delay(time * 100);    //время выполнения, регулируемое
}

void spin_left(int time)    //разворот влево (левое колесо вращается в обратном
                           //направлении, правое — в прямом)
{
    digitalWrite(Right_motor_go,HIGH);    //правый двигатель вращается в прямом
                                           //направлении

    digitalWrite(Right_motor_back,LOW);

    analogWrite(Right_motor_go,200);
    analogWrite(Right_motor_back,0);    //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,LOW);
    digitalWrite(Left_motor_back,HIGH);    //левый двигатель вращается в обратном
                                           //направлении

    analogWrite(Left_motor_go,0);
    analogWrite(Left_motor_back,200);    // ШИМ-регулирование оборотов: 0~255
    delay(time * 100);    //время выполнения, регулируемое
}

//void right(int time)
void right()    //поворот направо (правое колесо останавливается, левое
               //продолжает движение)
{
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,LOW);
    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,0);    //ШИМ-регулирование оборотов: 0~255
    digitalWrite(Left_motor_go,HIGH);    //левый двигатель вращается в прямом
                                           //направлении
    digitalWrite(Left_motor_back,LOW);    analogWrite(Left_motor_go,150);
    analogWrite(Left_motor_back,0);    //ШИМ-регулирование оборотов: 0~255
    delay(time * 100);    //время выполнения, регулируемое
}

void spin_right(int time)    //правый разворот (правое колесо вращается в обратном
                            //направлении, левое колесо — в прямом)
{
    digitalWrite(Right_motor_go,LOW);
    digitalWrite(Right_motor_back,HIGH);    //правый двигатель вращается в обратном
                                           //направлении

    analogWrite(Right_motor_go,0);
    analogWrite(Right_motor_back,200);    //ШИМ-регулирование оборотов: 0~255

```



```

digitalWrite(Left_motor_go,HIGH);          //левый двигатель вращается в прямом
                                           //направлении

digitalWrite(Left_motor_back,LOW);
analogWrite(Left_motor_go,200);
analogWrite(Left_motor_back,0);           //ШИМ-регулирование оборотов: 0~255
delay(time * 100);    //время выполнения, регулируемое
}
//void back(int time)
void back(int time)
{
digitalWrite(Right_motor_go,LOW); //правый двигатель вращается в обратном
                                //направлении

digitalWrite(Right_motor_back,HIGH);
analogWrite(Right_motor_go,0);
analogWrite(Right_motor_back,150); //ШИМ-регулирование оборотов: 0~255
digitalWrite(Left_motor_go,LOW);
digitalWrite(Left_motor_back,HIGH); //левый двигатель вращается в обратном
                                //направлении

analogWrite(Left_motor_go,0);
analogWrite(Left_motor_back,150); //ШИМ-регулирование оборотов: 0~255
delay(time * 100);    //время выполнения, регулируемое
}
//=====
void keysacn()
{
int val;
val=digitalRead(key); //считывание значения уровня порта 7 в val
while(!digitalRead(key)) //когда кнопка не нажата, цикл запущен
{
val=digitalRead(key); //при пропуске данной строчки — выход из цикла
}
while(digitalRead(key))//когда кнопка нажата
{
delay(10);
val=digitalRead(key);//считывание значения уровня порта 7 в val
if(val==HIGH) //определение повторного нажатия кнопки
{
digitalWrite(beep,HIGH);          //звуковой сигнал
while(!digitalRead(key))          //определение отпускания кнопки
digitalWrite(beep,LOW);           //звуковой сигнал выключен
}
else
digitalWrite(beep,LOW);           //звуковой сигнал выключен
}
}
void loop()
{
keysacn();    //вызов функции определения состояния кнопки
while(1)
{
//Сигнал — LOW, нет сигнала — HIGH

```

```

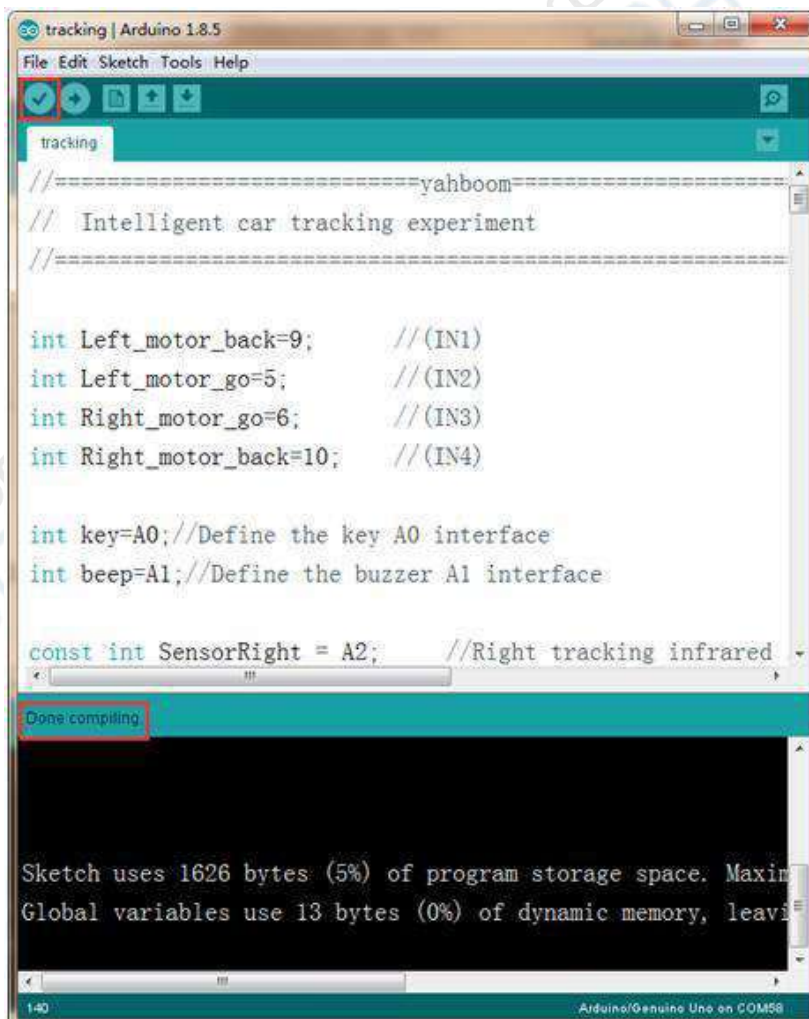
SR = digitalRead(SensorRight);
//наличие сигнала, если излучение L3 отражается от белой поверхности;
//отсутствие сигнала, если излучение L3 поглощается черной поверхностью;

SL = digitalRead(SensorLeft);
//наличие сигнала, если излучение L2 отражается от белой поверхности;
//отсутствие сигнала, если излучение L2 поглощается черной поверхностью;
if (SL == LOW && SR == LOW)
    run(); //вызов функции движения
else if (SL == HIGH & SR == LOW)
    //срабатывание левого ИК-датчика, автомобиль отклонился от трека, поворот //налево
    left();
else if (SR == HIGH & SL == LOW)
    //срабатывание правого ИК-датчика, автомобиль отклонился от трека, поворот //направо
    right();
else //везде белая поверхность, стоп
    brake();
}
}

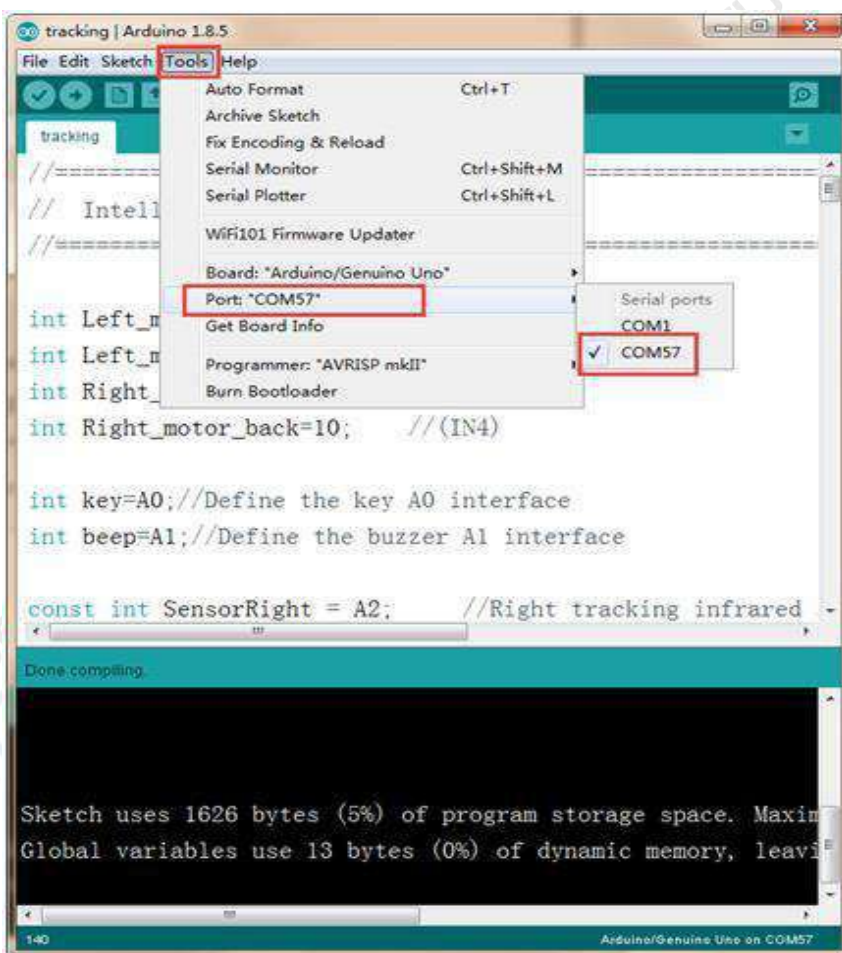
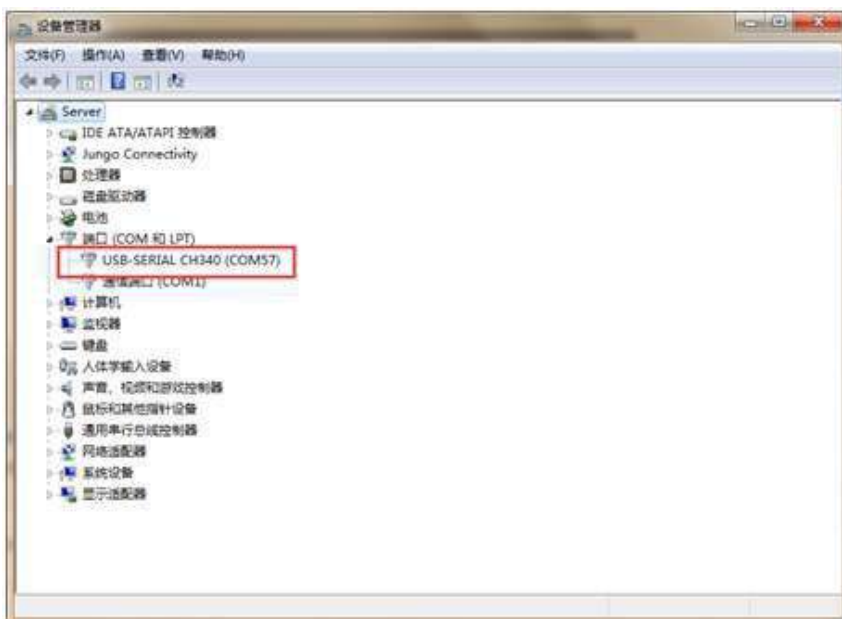
```

Этапы эксперимента:

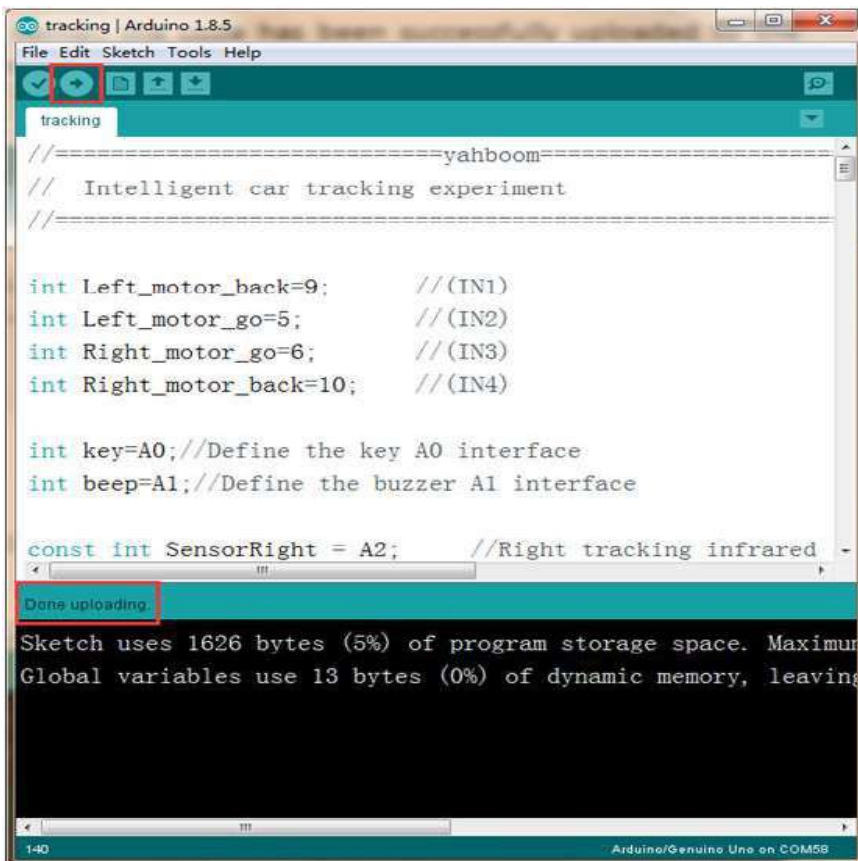
1. Откройте файл с кодом эксперимента: **tracking.ino**, нажмите кнопку “✓” под панелью меню для компиляции кода, дождитесь сообщения **"Done compiling"** (компиляция завершена) в нижнем правом углу (см. рисунок для примера).



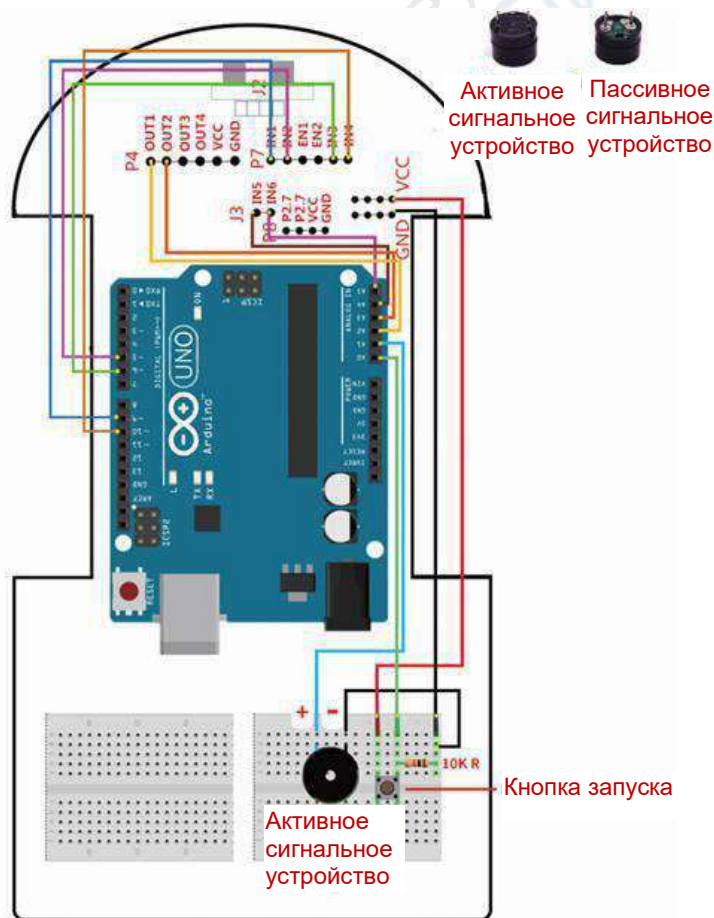
2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).



3. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. После успешной загрузки кода в плату Arduino UNO в нижнем левом углу отобразится сообщение **“Done uploading”** (загрузка завершена, см. рисунок).



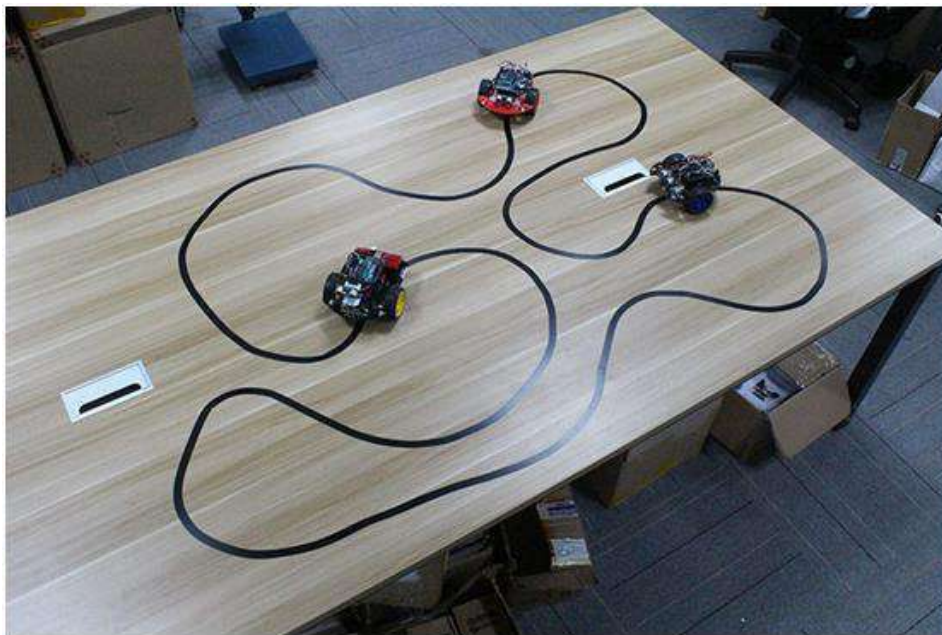
4. Выполните подключение в соответствии с рисунком.



Примечание: вставьте ультразвуковой датчик в слот J2, как показано на рисунке

Эксперимент является комплексным. Автомобиль может обнаруживать препятствия при движении по треку. При обнаружении препятствия автомобиль останавливается. После удаления препятствия автомобиль продолжает движение.

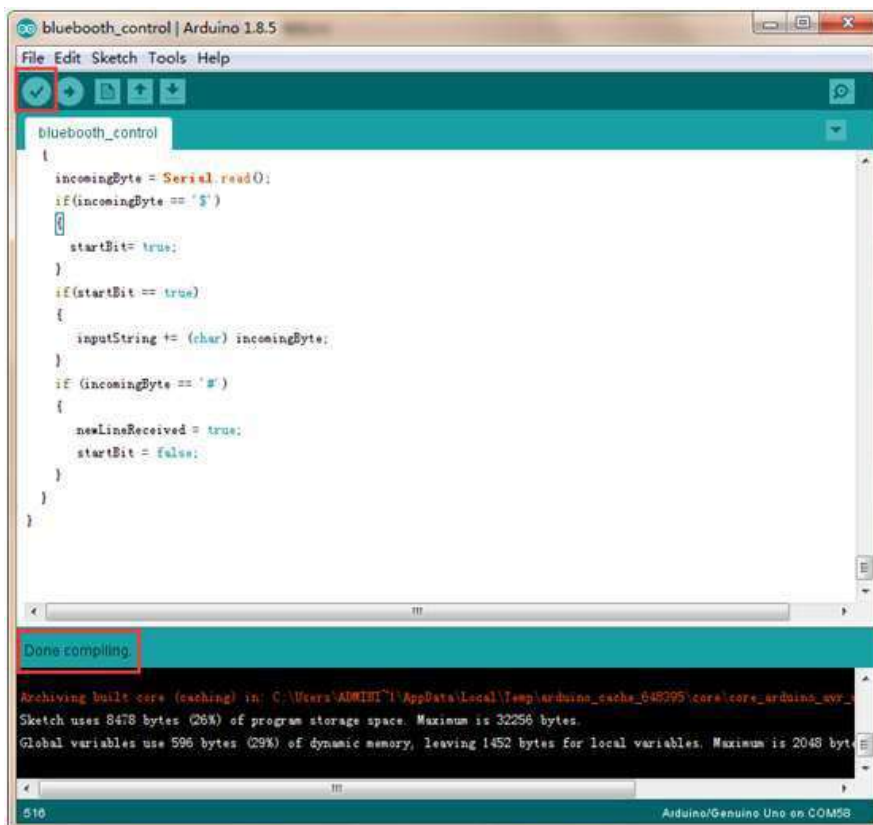
5. Для формирования трека на светлой поверхности, например, на столе, используйте черную изоленту шириной 1,6 см. Установите запрограммированный автомобиль на трек и нажмите кнопку запуска. Автомобиль начнет движение по черной линии.



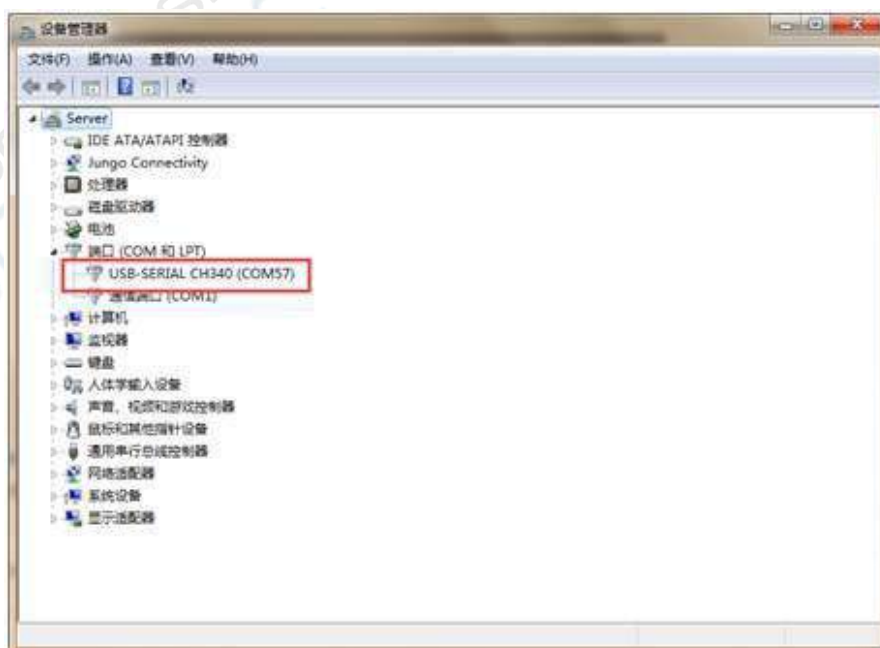
6- Управление по Bluetooth

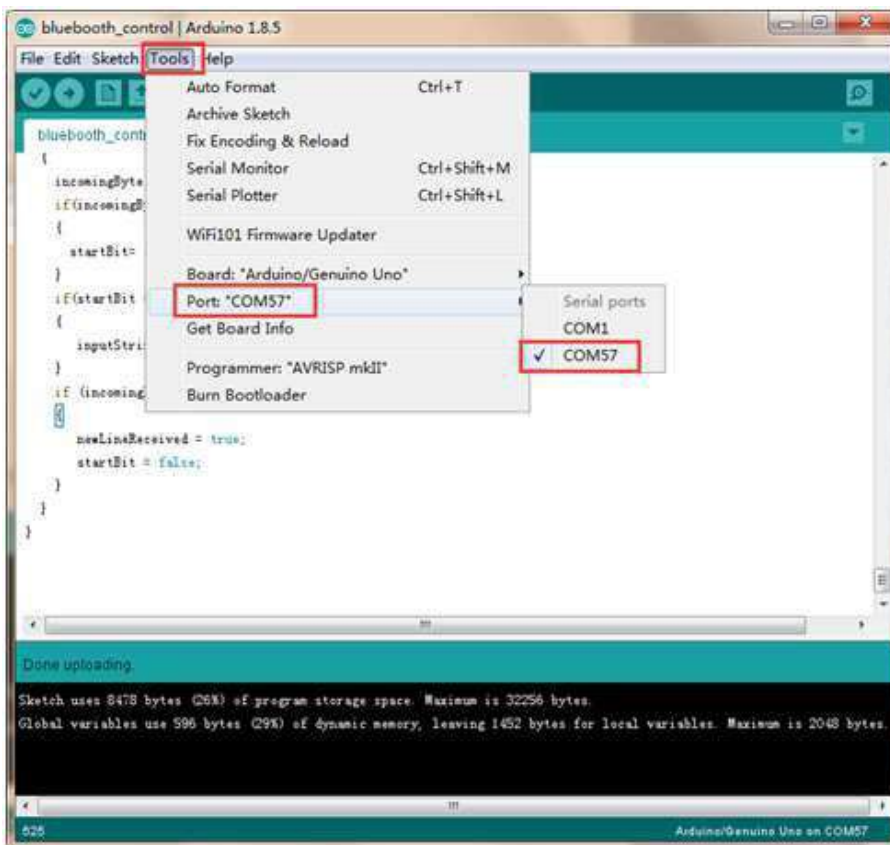
Этапы эксперимента:

1. Откройте файл с кодом эксперимента: **bluebooth_control.ino**, нажмите кнопку “✓” под панелью меню для компиляции кода, дождитесь сообщения **"Done compiling"** (компиляция завершена) в нижнем правом углу (см. рисунок для примера).

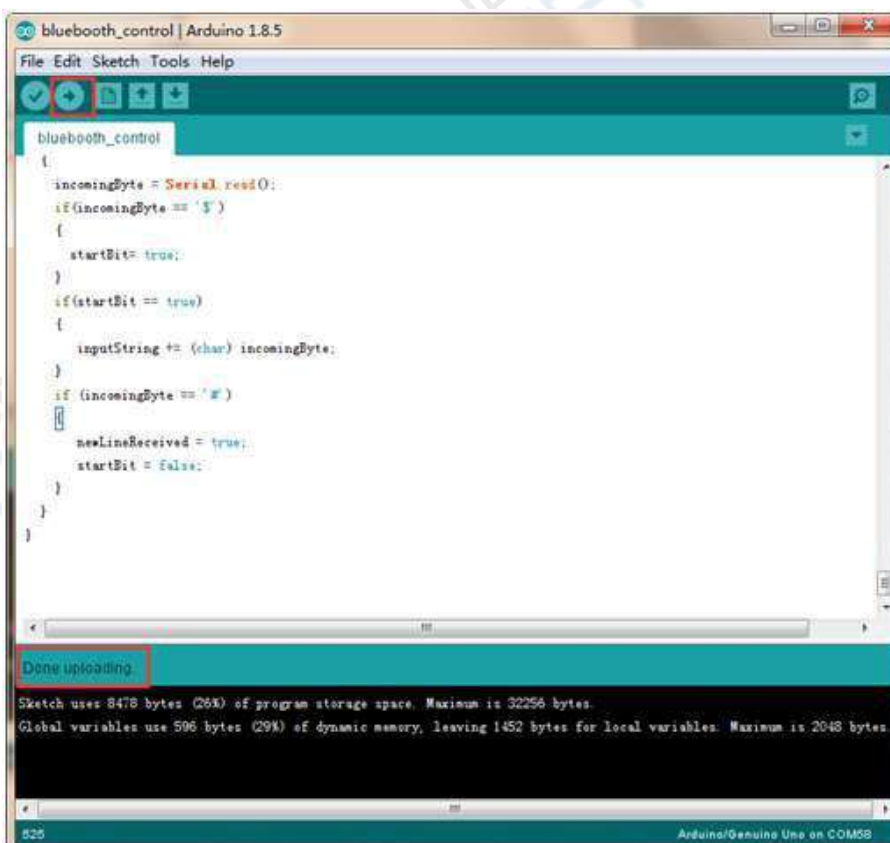


2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).





3. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. После успешной загрузки кода в плату Arduino UNO в нижнем левом углу отобразится сообщение “**Done uploading**” (загрузка завершена, см. рисунок).



4. Выполните подключение в соответствии со следующим рисунком.

Схема подключения для удаленного управления роботом-автомобилем по Bluetooth



5. Поскольку модуль Bluetooth и серийный порт программатора совместно используют порты ввода/вывода 0 и 1, при прошивке возможно возникновение ошибки. Отключите питание модуля Bluetooth перед прошивкой и подключите его снова после завершения прошивки.

Робот-автомобиль на базе Arduino (с Bluetooth)

1. Комплект поставки

Стандартный робот автомобиль на базе Arduino + модуль Bluetooth

2. Введение

Версия автомобиля с Bluetooth отличается от стандартной версии только наличием Bluetooth-модуля. Пользователи могут использовать предоставляемое нами приложение для удаленного управления автомобилем по Bluetooth. С помощью приложения можно управлять направлением движения (вперед, назад, направо, налево), скоростью и поворотами. Приложение является удобным и простым в использовании, что необходимо для желающих удаленно управлять роботом-автомобилем. Приложение доступно только для устройств на базе Android.

3. Порядок настройки Bluetooth-соединения

А) Выполните подключение в соответствии со схемой P15 раздела по удаленному управлению через Bluetooth.

Б) Включите питание автомобиля и проверьте наличие питания на Bluetooth-модуле (светодиод на модуле должен мигать).

В) Откройте настройки Bluetooth на смартфоне, установите связь с Bluetooth-модулем автомобиля и введите пароль 1234.

Г) Откройте на смартфоне приложение для удаленного управления. Интерфейс приложения приведен ниже. Если на смартфоне Bluetooth отключен, кнопка рядом с переключателем Bluetooth подсвечивается серым фоном. Для активации соединения нажмите на переключатель.

Д) Для выбора Bluetooth-модуля автомобиля нажмите кнопку SEARCH.

Е) Нажмите кнопку CONNECT. Программа сообщит об успешном подключении. Если подключиться не удастся, повторите процедуру.



ПРИМЕЧАНИЕ: Перед загрузкой кода необходимо отключить Bluetooth-модуль.