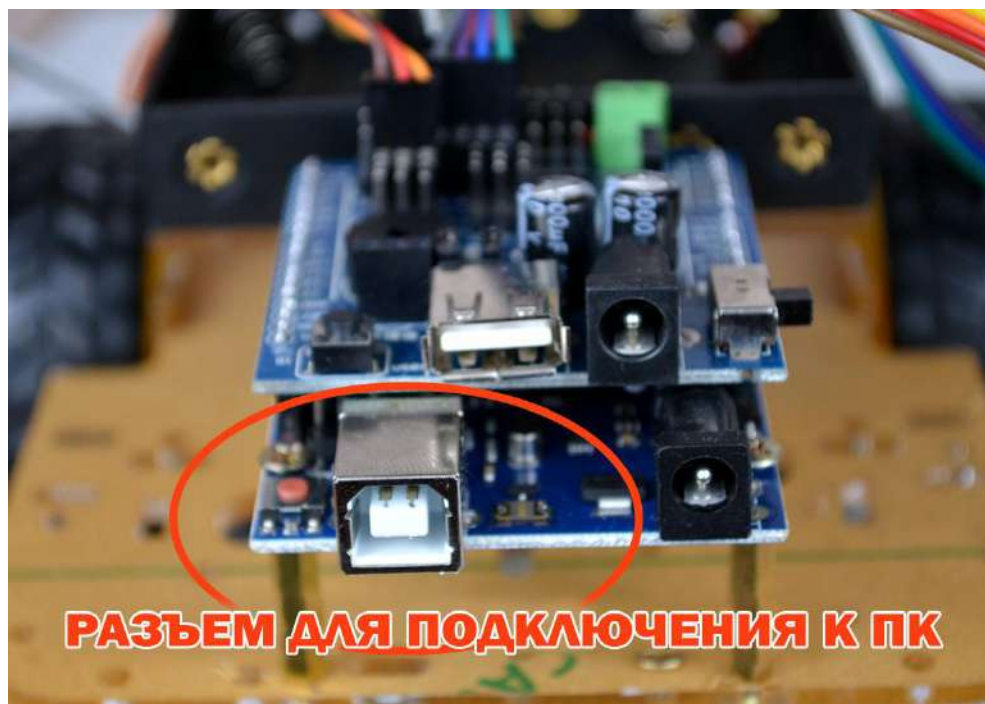


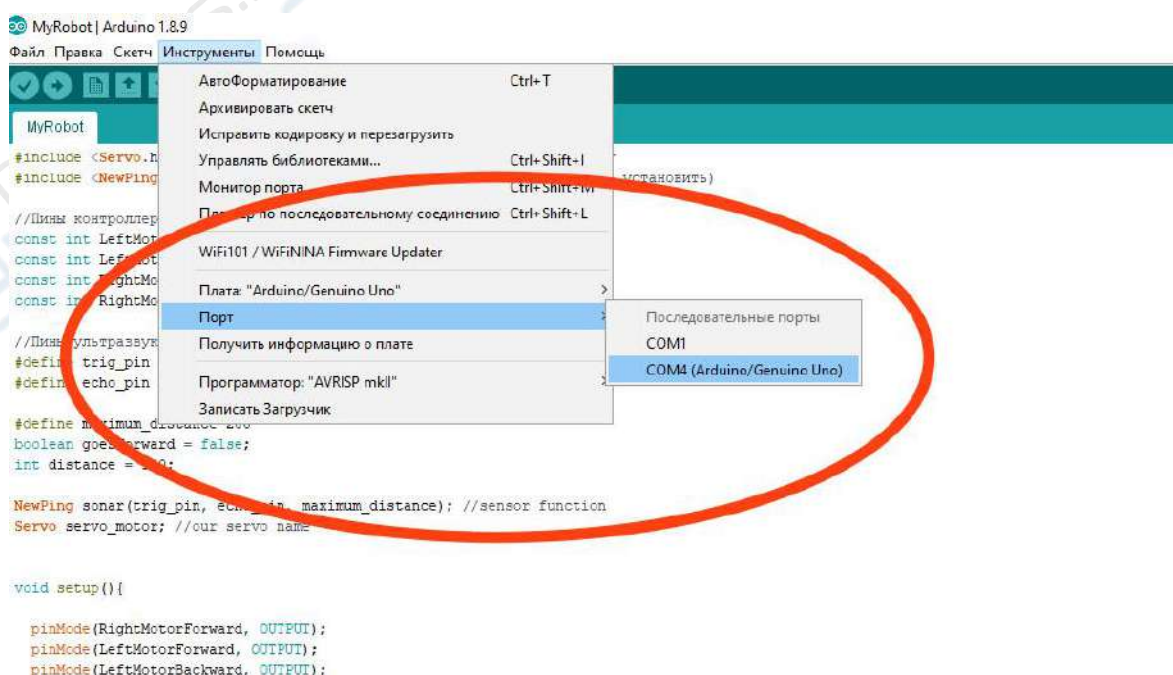
ЗАГРУЗКА СКЕТЧА В ARDUINO

Перед началом работы установите среду разработки Arduino IDE. Скачать ее можно [по ссылке](#).

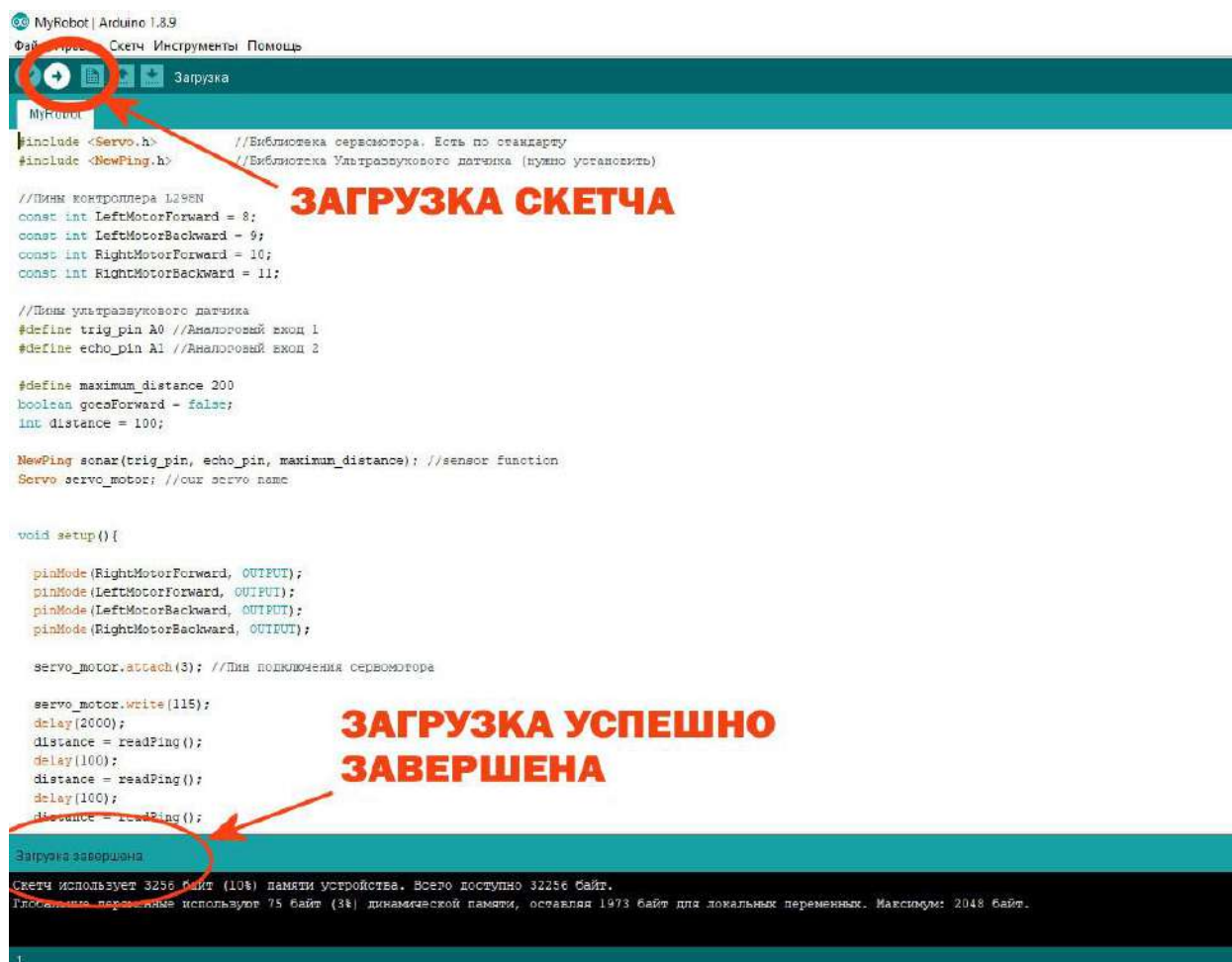
Для загрузки скетча необходимо подключить плату Arduino с помощью USB кабеля к компьютеру.



После установки и запуска среды разработки и подключения arduino к ПК необходимо зайти в меню инструменты -> порт и выбрать com-порт, на который подключился Arduino.



Загрузите [ИСХОДНЫЙ КОД](#) или вставьте копированием и через кнопку «загрузить» прошейте Arduino.



Отключите USB кабель от платы, включите кнопку питания. Робот готов.

КОД СКЕТЧА

```
#include <Servo.h>      //Библиотека сервомотора. Есть по стандарту
#include <NewPing.h>     //Библиотека Ультразвукового датчика (нужно установить)

//Пины контроллера L298N
const int LeftMotorForward = 8;
const int LeftMotorBackward = 9;
const int RightMotorForward = 10;
const int RightMotorBackward = 11;

//Пины ультразвукового датчика
#define trig_pin A0 //Аналоговый вход 1
#define echo_pin A1 //Аналоговый вход 2

#define maximum_distance 200
boolean goesForward = false;
int distance = 100;

NewPing sonar(trig_pin, echo_pin, maximum_distance); //sensor function
Servo servo_motor; //our servo name

void setup(){

  pinMode(RightMotorForward, OUTPUT);
  pinMode(LeftMotorForward, OUTPUT);
  pinMode(LeftMotorBackward, OUTPUT);
  pinMode(RightMotorBackward, OUTPUT);

  servo_motor.attach(3); //Пин подключения сервомотора

  servo_motor.write(115);
  delay(2000);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
  distance = readPing();
  delay(100);
}

void loop(){

  int distanceRight = 0;
  int distanceLeft = 0;
  delay(50);

  if (distance <= 20){
```

```
moveStop();
delay(300);
moveBackward();
delay(400);
moveStop();
delay(300);
distanceRight = lookRight();
delay(300);
distanceLeft = lookLeft();
delay(300);
```

```
if (distance >= distanceLeft){
    turnRight();
    moveStop();
}
else{
    turnLeft();
    moveStop();
}
}
else{
    moveForward();
}
distance = readPing();
}
```

```
int lookRight(){
    servo_motor.write(50);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
}
```

```
int lookLeft(){
    servo_motor.write(170);
    delay(500);
    int distance = readPing();
    delay(100);
    servo_motor.write(115);
    return distance;
    delay(100);
}
```

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

```
return cm;  
}
```

```
void moveStop(){
```

```
    digitalWrite(RightMotorForward, LOW);  
    digitalWrite(LeftMotorForward, LOW);  
    digitalWrite(RightMotorBackward, LOW);  
    digitalWrite(LeftMotorBackward, LOW);  
}
```

```
void moveForward(){
```

```
    if(!goesForward){  
  
        goesForward=true;  
  
        digitalWrite(LeftMotorForward, HIGH);  
        digitalWrite(RightMotorForward, HIGH);  
  
        digitalWrite(LeftMotorBackward, LOW);  
        digitalWrite(RightMotorBackward, LOW);  
    }  
}
```

```
void moveBackward(){
```

```
    goesForward=false;  
  
    digitalWrite(LeftMotorBackward, HIGH);  
    digitalWrite(RightMotorBackward, HIGH);  
  
    digitalWrite(LeftMotorForward, LOW);  
    digitalWrite(RightMotorForward, LOW);  
  
}
```

```
void turnRight(){
```

```
    digitalWrite(LeftMotorForward, HIGH);  
    digitalWrite(RightMotorBackward, HIGH);  
  
    digitalWrite(LeftMotorBackward, LOW);  
    digitalWrite(RightMotorForward, LOW);
```

```
    delay(500);
```

```
    digitalWrite(LeftMotorForward, HIGH);  
    digitalWrite(RightMotorForward, HIGH);
```

```
    digitalWrite(LeftMotorBackward, LOW);
```

```
digitalWrite(RightMotorBackward, LOW);
```

```
}
```

```
void turnLeft(){
```

```
    digitalWrite(LeftMotorBackward, HIGH);
```

```
    digitalWrite(RightMotorForward, HIGH);
```

```
    digitalWrite(LeftMotorForward, LOW);
```

```
    digitalWrite(RightMotorBackward, LOW);
```

```
    delay(500);
```

```
    digitalWrite(LeftMotorForward, HIGH);
```

```
    digitalWrite(RightMotorForward, HIGH);
```

```
    digitalWrite(LeftMotorBackward, LOW);
```

```
    digitalWrite(RightMotorBackward, LOW);
```

```
}
```