

2019

Мобильный робот Yahboom Arduino 4WD Robot Kit

5 уроков по программированию робота в Arduino IDE



Программирование

Оглавление

1- Разноцветная индикация.....	2
2- Движение вперед	5
3- Движение по алгоритму.....	9
4- Сервоуправление цветом.....	13
5- ИК-обнаружение препятствий	17

1- Разноцветная индикация

1) Подготовка



1-1 Плата Arduino UNO



1-2 Модуль RGB

2) Цель эксперимента

После загрузки кода и двухсекундной паузы циклически переключаются семь различных цветов.

3) Суть эксперимента

На световом модуле RGB размещены три светодиода (красный, зеленый, синий). Получение различных цветов (256x256x256) достигается путем управления яркостью свечения указанных светодиодов.

В соответствии с электрической схемой, лампа RGB является светодиодом с общим катодом, один контакт которой подключен к GND (заземление), а остальные три контакта подключены соответственно к контактам 11, 10, 9 платы Arduino UNO. Каждый светодиод должен быть подключен последовательно через резистор 220 Ом, который ограничивает ток. Мы можем управлять светодиодом, подавая напряжение на соответствующий контакт платы Arduino UNO.

4) Этапы эксперимента

4-1 Электрическая схема



4-1 Схема подключения платы Arduino UNO



4-2 Подключение модуля RGB

4-2 В соответствии с электрической схемой:

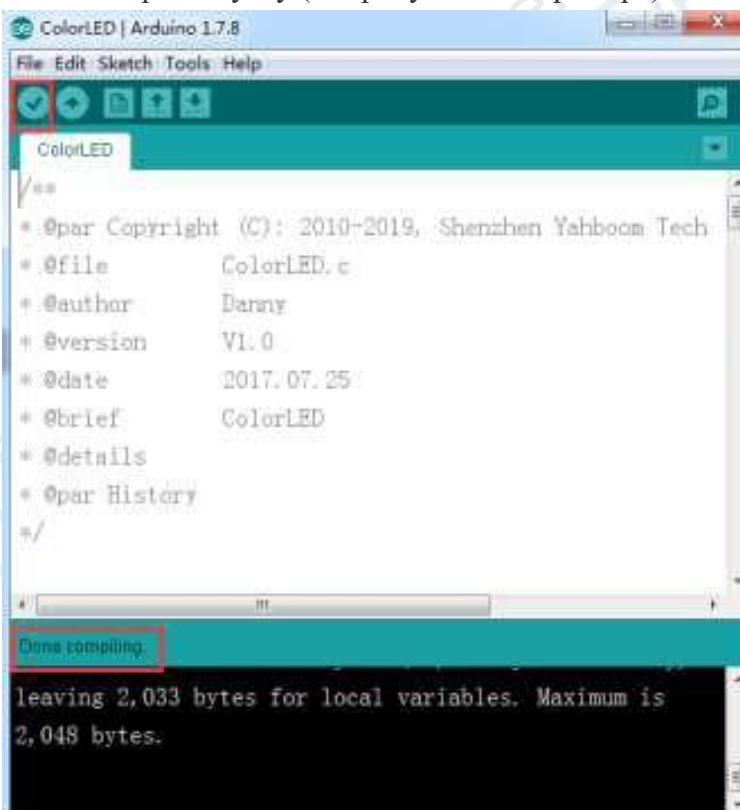
LED_R-----11 (Arduino UNO)

LED_G-----10 (Arduino UNO)

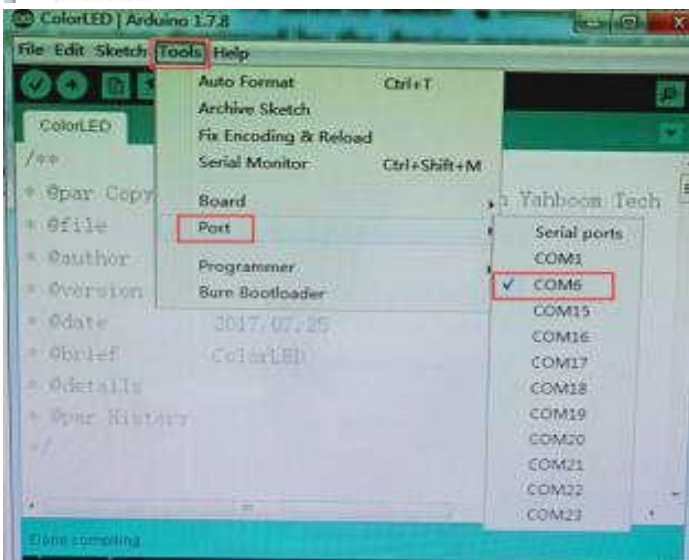
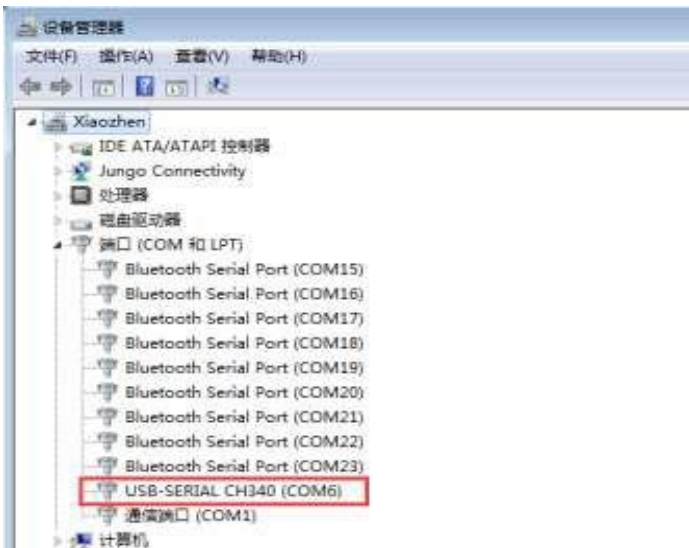
LED_B-----9 (Arduino UNO) 4-

5 Программный код

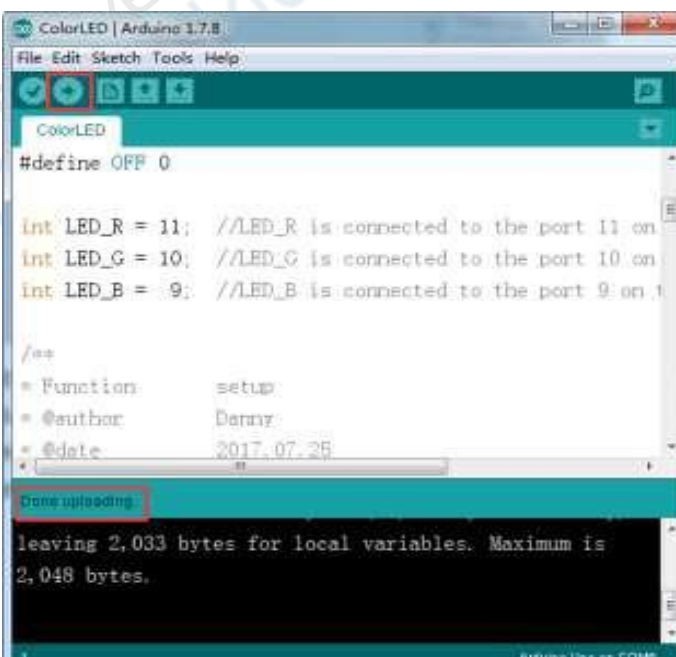
1. Откройте файл с кодом эксперимента: **ColorLED.ino** (ссылка: <https://yadi.sk/d/iafbCLQ8XXXTvQ>), нажмите кнопку “√” под панелью меню для компиляции кода, дождитесь сообщения "**Done compiling**" (компиляция завершена) в нижнем правом углу (см. рисунок для примера).



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).



2- Движение вперед

1) Подготовка



1-1 Плата Arduino UNO



1-2 Четыре электродвигателя

2) Цель эксперимента

После загрузки кода и двухсекундной задержки осуществляется движение вперед в течение 1 сек.

3) Суть эксперимента

В эксперименте используется драйвер TB6612FNG, предназначенный для управления электродвигателями. Запуск двигателей в прямом и обратном направлении, а также их остановка осуществляются путем управления уровнями сигналов, подаваемых на контакты драйвера AIN1, AIN2, BIN1, BIN2, PWMA и PWMB.

В данном эксперименте движение автомобиля осуществляется благодаря подаче сигнала высокого уровня на контакт AIN1, сигнала низкого уровня на AIN2, сигнала высокого уровня на BIN1, сигнала низкого уровня на BIN2. Скорость робота-автомобиля управляется путем ШИМ-регулирования (контакты PWMA, PWMB (0-255)). При этом используется одноканальный ШИМ-регулятор частоты вращения электродвигателей.

4) Этапы эксперимента

4-1 Электрическая схема

Схема подключения платы Arduino



4-1 Схема подключения платы Arduino UNO



Схема подключения драйвера

4-2 Драйвер электродвигателей --- TB6612FNG

4-2 В соответствии с электрической схемой:

AIN1-----7 (Arduino UNO)

AIN2-----8 (Arduino UNO)

PWMA-----6 (Arduino UNO)

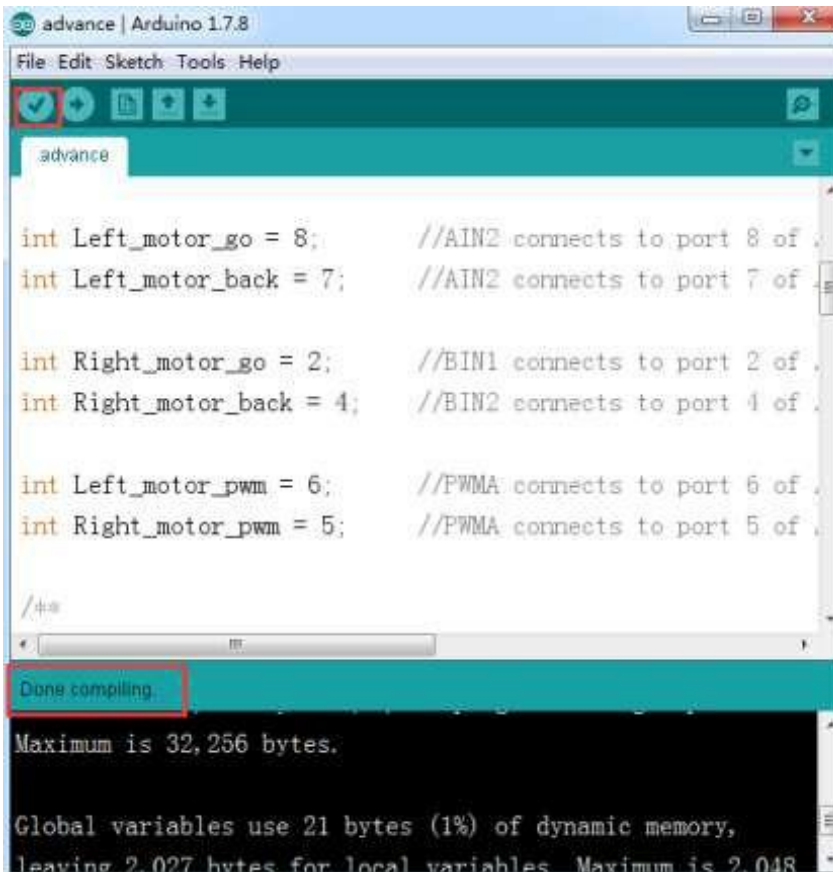
BIN1-----4 (Arduino UNO)

BIN2-----2 (Arduino UNO)

PWMB-----5 (Arduino UNO)

4-3 Программный код

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



```
advance | Arduino 1.7.8
File Edit Sketch Tools Help
advance

int Left_motor_go = 8; //AIN2 connects to port 8 of
int Left_motor_back = 7; //AIN2 connects to port 7 of

int Right_motor_go = 2; //BIN1 connects to port 2 of
int Right_motor_back = 4; //BIN2 connects to port 4 of

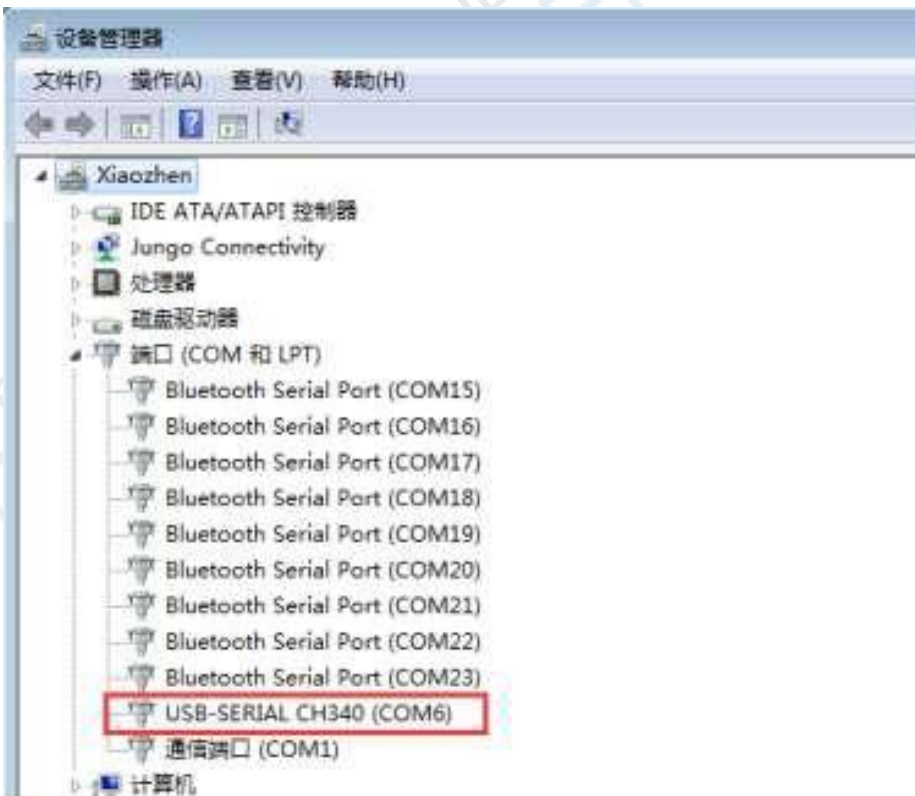
int Left_motor_pwm = 6; //PWMA connects to port 6 of
int Right_motor_pwm = 5; //PWMA connects to port 5 of

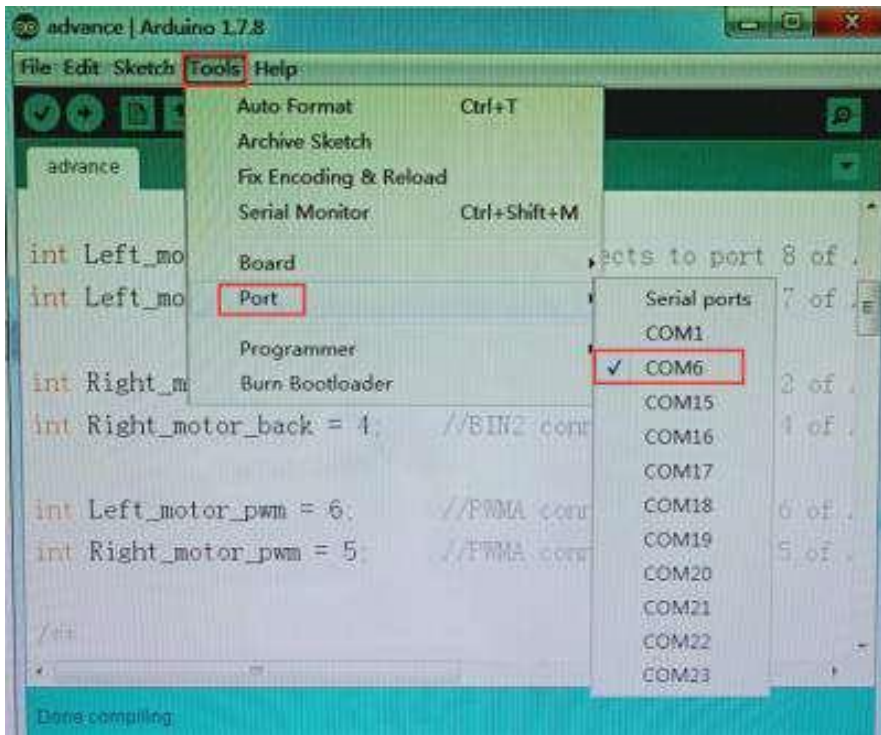
/*

Done compiling.
Maximum is 32,256 bytes.

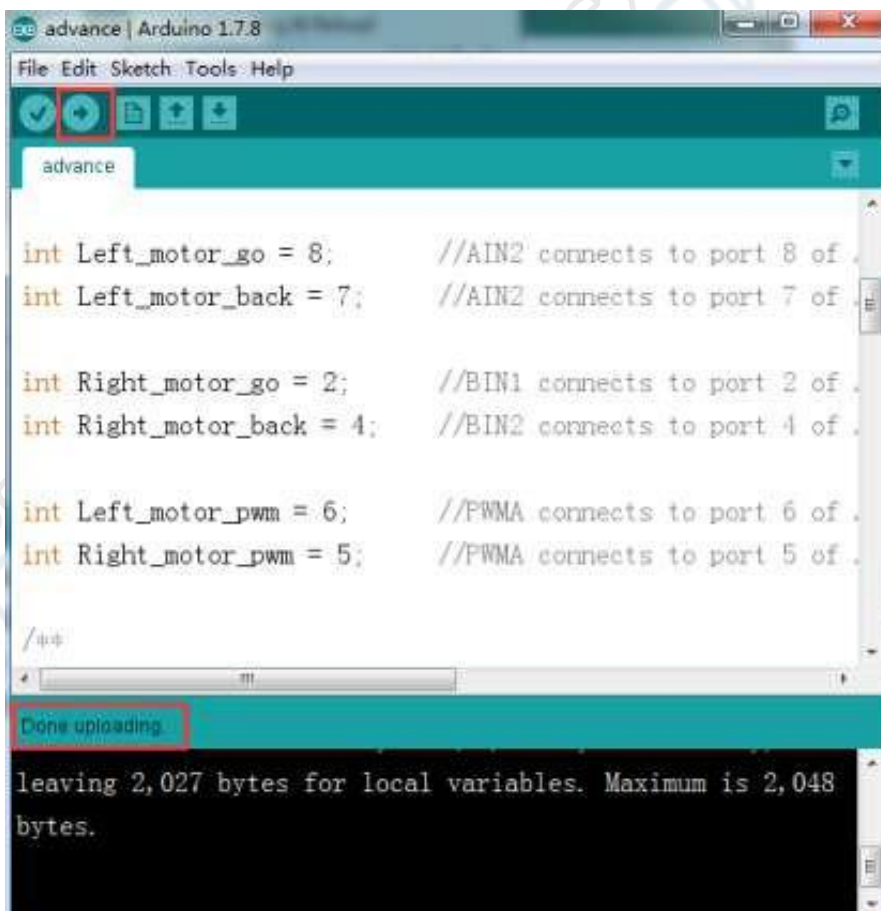
Global variables use 21 bytes (1%) of dynamic memory,
leaving 2,027 bytes for local variables. Maximum is 2,048
```

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).



3- Движение по алгоритму

1) Подготовка



1-1 Плата Arduino UNO



1-2 Четыре электродвигателя

2) Цель эксперимента

После загрузки кода и двухсекундной задержки робот-автомобиль движется вперед в течение 1 сек., назад в течение 1 сек., поворачивает налево в течение 2 сек., поворачивает направо в течение 2 сек., разворачивается на месте через левую сторону в течение 3 сек., разворачивается на месте через правую сторону в течение 3 сек., останавливается на 0,5 секунд. Вышеописанный цикл движений повторяется снова и снова.

3) Суть эксперимента

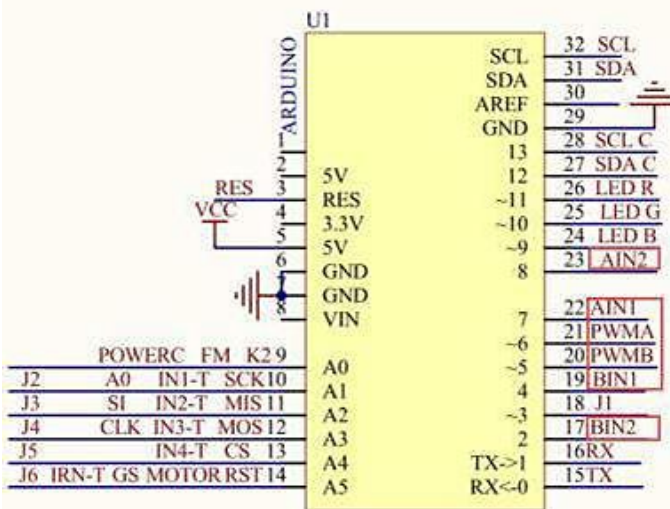
В данном эксперименте также используется драйвер TB6612FNG. Запуск двигателей в прямом и обратном направлении, а также их остановка осуществляются путем управления уровнями сигналов, подаваемых на контакты драйвера AIN1, AIN2, BIN1, BIN2, PWMA и PWMB.

Движение автомобиля осуществляется благодаря подаче сигнала высокого уровня на контакт AIN1, сигнала низкого уровня на AIN2, сигнала высокого уровня на BIN1, сигнала низкого уровня на BIN2. Скорость автомобиля управляется путем ШИМ-регулирования (контакты PWMA, PWMB (0-255)). При этом используется одноканальный ШИМ-регулятор частоты вращения электродвигателей.

4) Этапы эксперимента

4-1 Электрическая схема

Схема подключения платы Arduino



4-1 Схема подключения платы Arduino UNO



Схема подключения драйвера

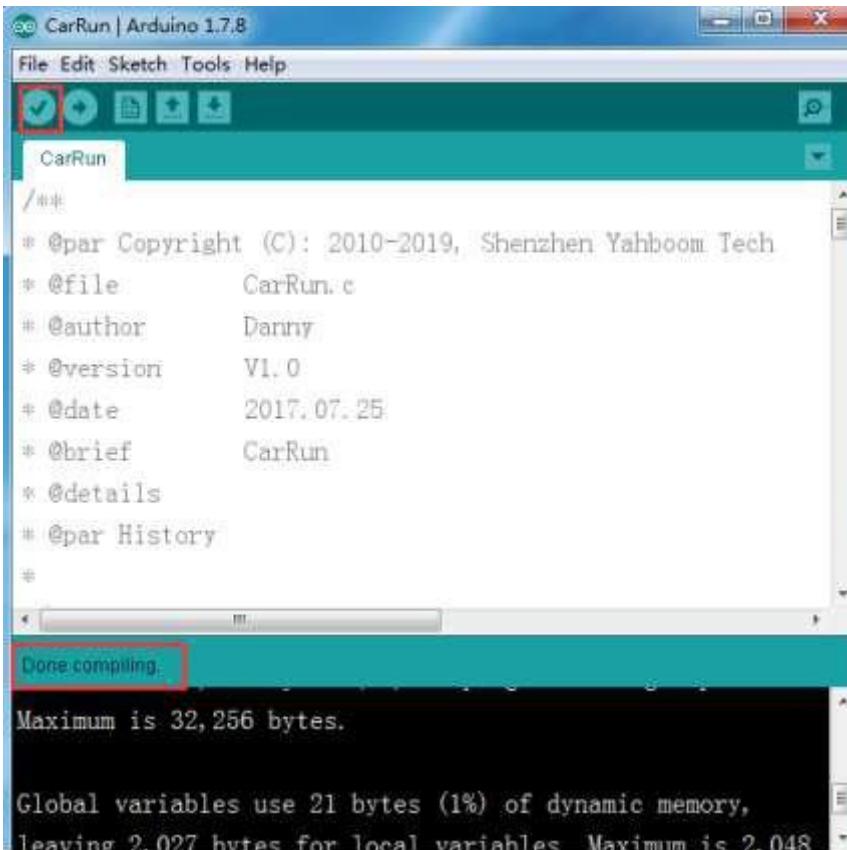
4-2 Драйвер электродвигателей ---TB6612FNG

4-2 В соответствии с электрической схемой :

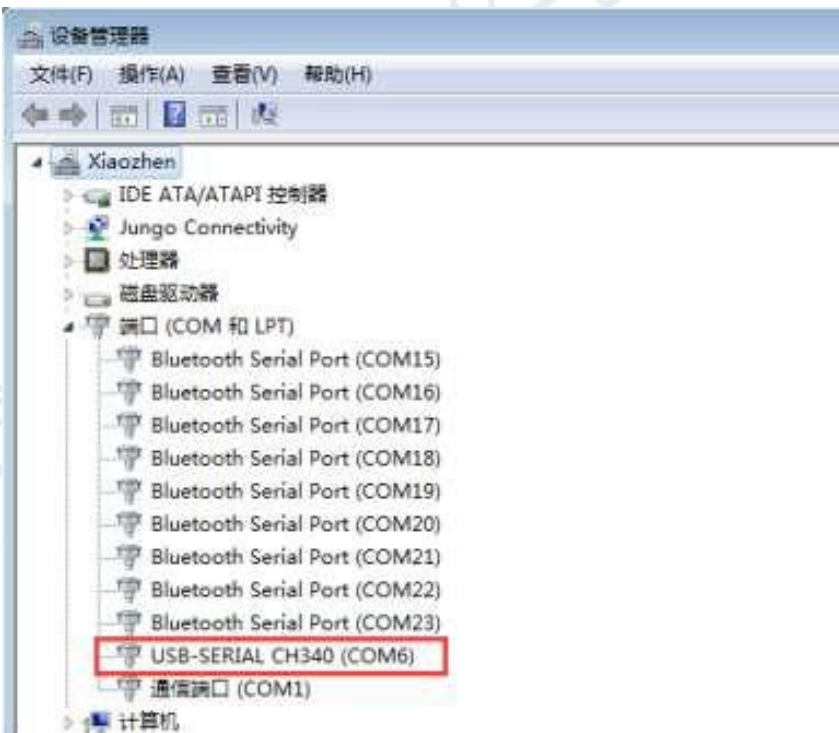
- AIN1-----7 (Arduino UNO)
- AIN2-----8 (Arduino UNO)
- PWMA-----6 (Arduino UNO)
- BIN1-----4 (Arduino UNO)
- BIN2-----2 (Arduino UNO)
- PWMB-----5 (Arduino UNO)

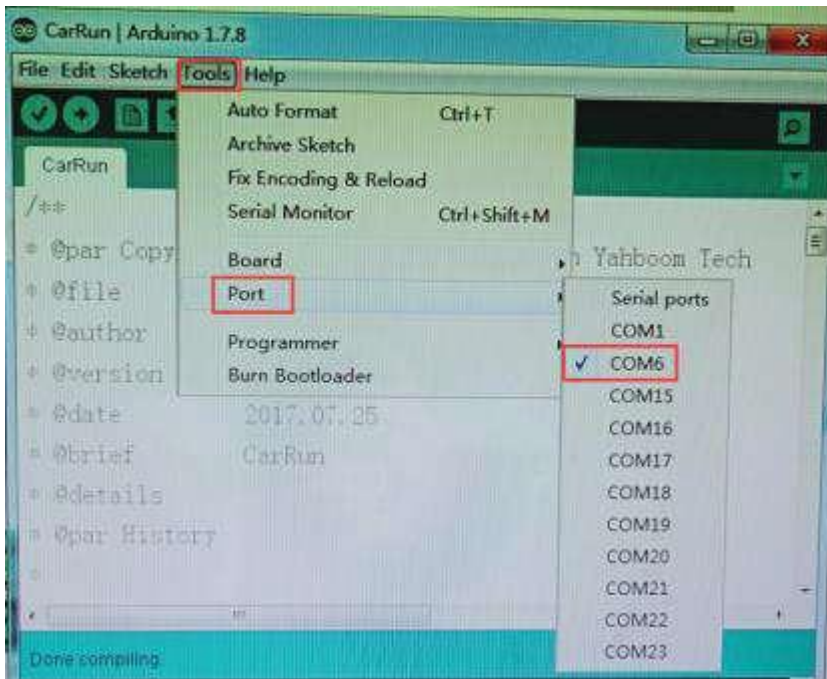
4-3 Программный код

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

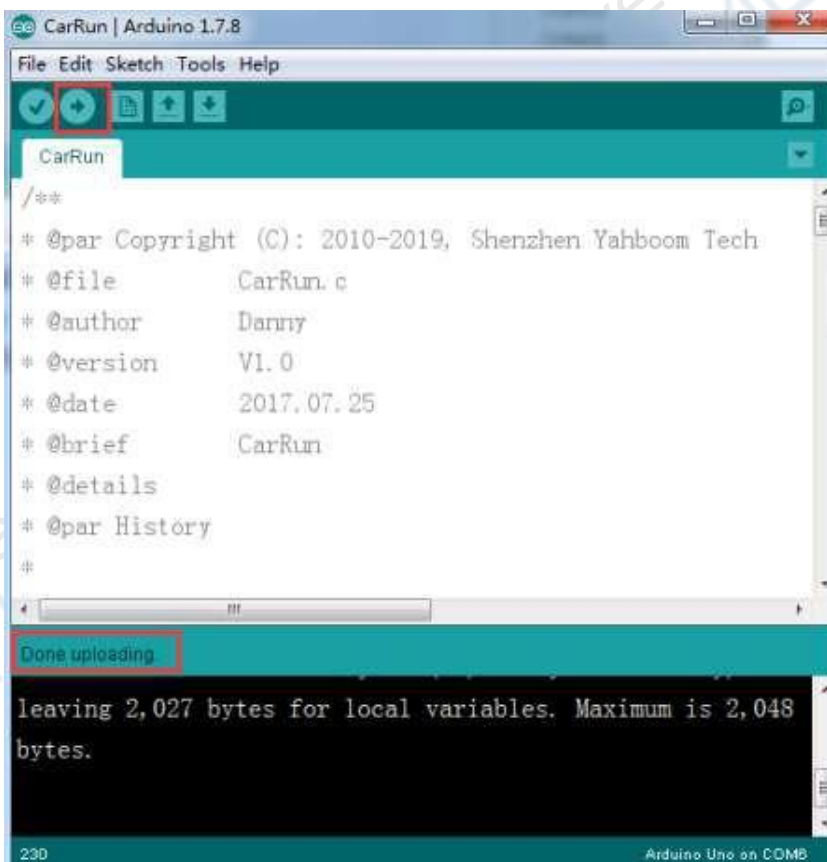


2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).





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- Сервоуправление цветом

1) Подготовка



1-1 Плата Arduino UNO



1-2 Серводвигатель SG90



1-3 Модуль RGB

2) Цель эксперимента

После загрузки кода автомобиль остается неподвижным в течение 0,5 сек. Серводвигатель начинает движение, при повороте на определенные углы включается освещение различных цветов.

3) Суть эксперимента

Принцип работы серводвигателя: управляющий сигнал поступает на модулирующую микросхему с канала приемника для генерации опорного напряжения смещения (постоянный ток). Управляющая схема генерирует эталонное напряжение с периодом 20 мс длительностью 1,5 мс. Схема сравнивает опорное напряжение с напряжением на потенциометре для определения разницы этих напряжений и генерации выходного сигнала. Положительная или отрицательная разница напряжений подается на микросхему привода, которая активирует двигатель в прямом или обратном направлении.

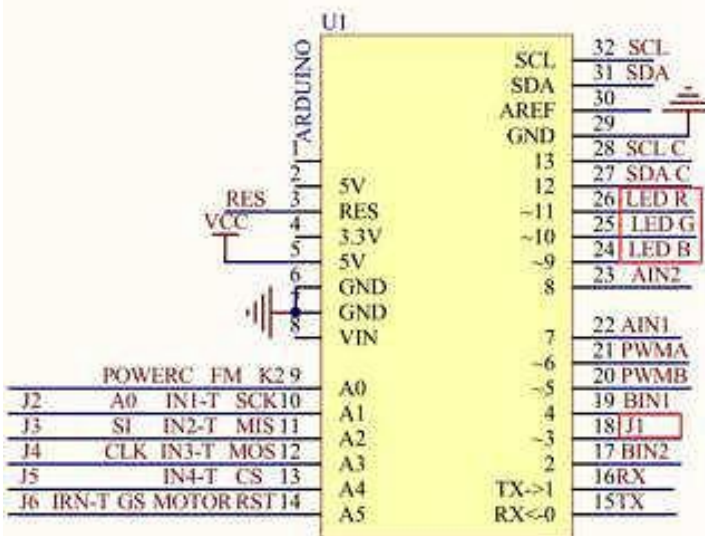
Угол поворота серводвигателя регулируется коэффициентом заполнения ШИМ-сигнала (широтно-импульсная модуляция). Стандартный ШИМ-сигнал имеет фиксированный период, равный 20 мс (50 Гц). Теоретически, распределение длительности импульса должно составлять от 1 мс до 2 мс, но фактически ширина импульса может составлять от 0,5 мс до 2,5 мс. Ширина импульса и угол поворота серводвигателя 0~180° соответствуют нижеприведенным значениям.

0,5мс-----	0°
1,0мс-----	45°
1,5мс-----	90°
2,0мс-----	135°
2,5мс-----	180°

4) Этапы эксперимента

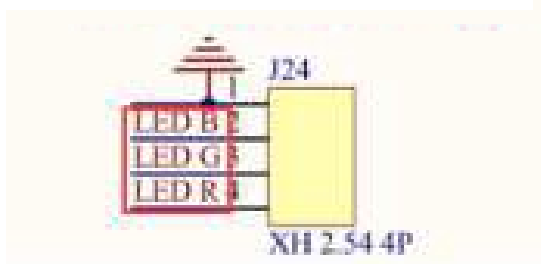
4-1 Электрическая схема

Схема подключения платы Arduino

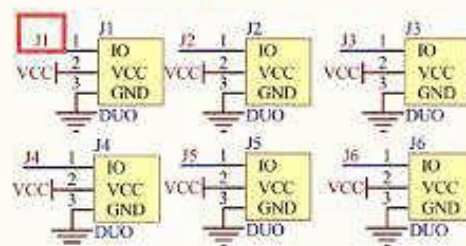


4-1 Схема подключения платы Arduino UNO

Светодиодный модуль



Серводвигатель



4-2 В соответствии с электрической схемой:

LED_R-----11 (Arduino UNO)

LED_G-----10 (Arduino UNO)

LED_B-----9 (Arduino UNO)

J1---3 (Arduino UNO)

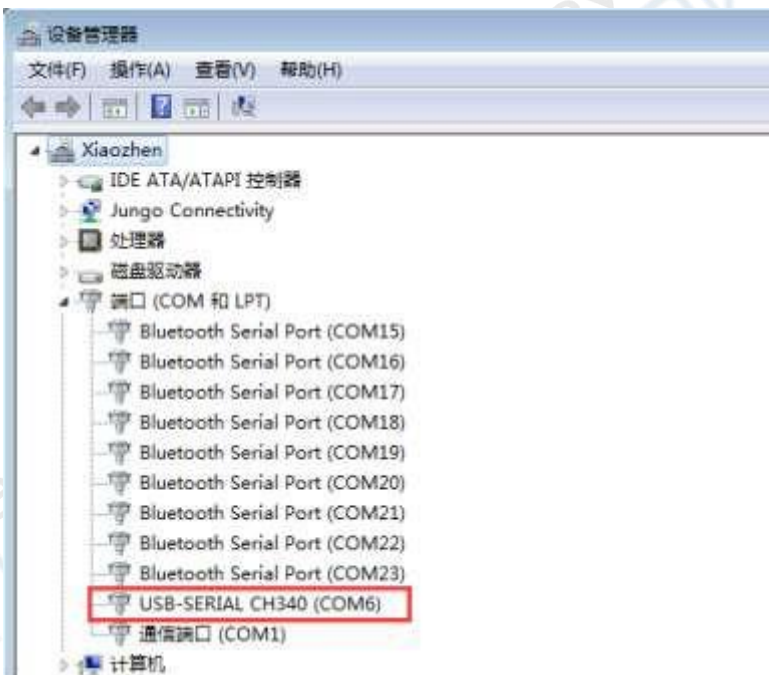
(Примечание: серводвигатель SG90 подключается к интерфейсу рулевого механизма J1).

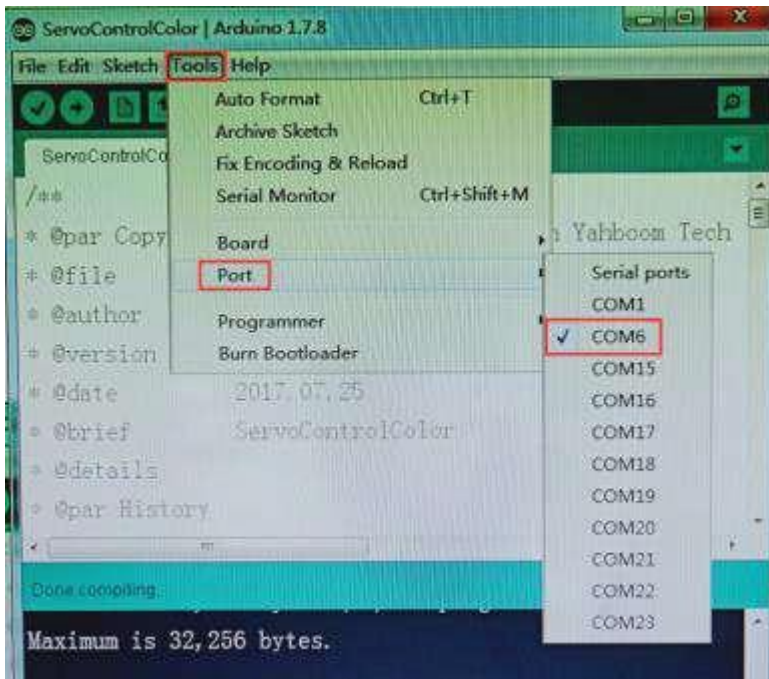
4-3 Программный код

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

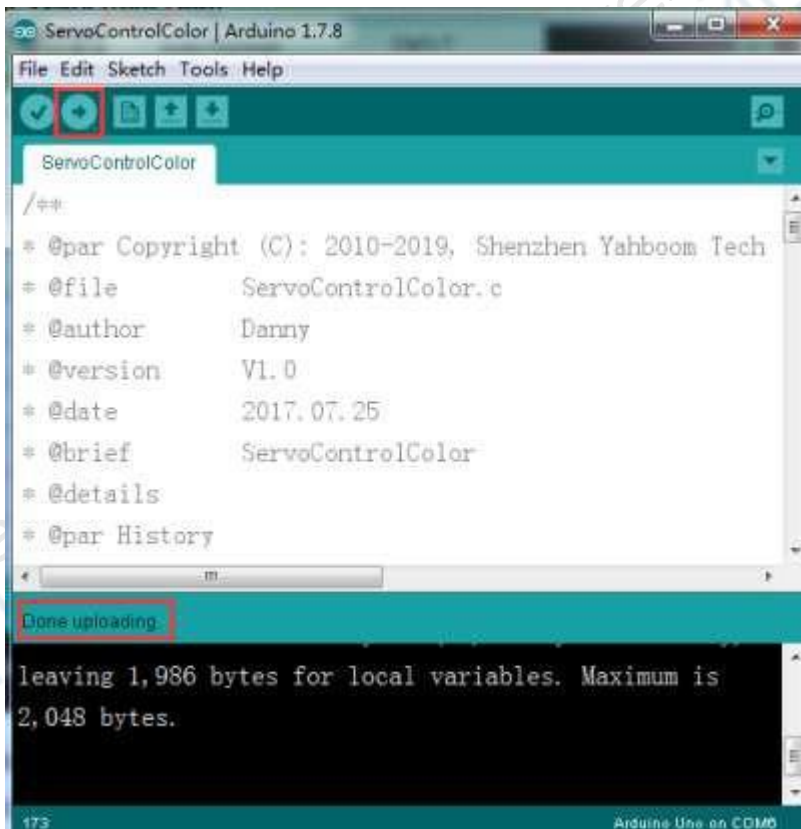


2. В меню программы Arduino IDE выберите пункт **【Tools】** --- **【Port】** ---, после чего выберите порт, соответствующий серийному номеру порта в Диспетчере устройств (пример на рисунке).



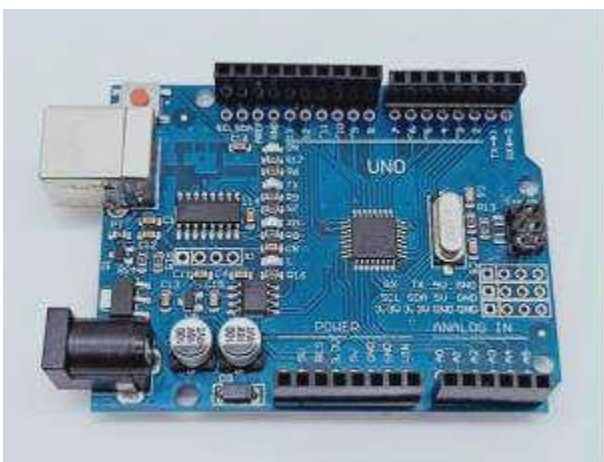


3. После выбора нужного пункта следует нажать кнопку “→”, расположенную под панелью меню, для загрузки кода в плату Arduino UNO. После успешной загрузки кода в плату Arduino UNO в нижнем левом углу отображается сообщение “**Done uploading**” (загрузка завершена, см. рисунок).



5- ИК-обнаружение препятствий

1) Подготовка



1-1 Плата Arduino UNO



1-2 Модуль ИК-обнаружения препятствий

2) Цель эксперимента

После компиляции кода необходимо нажать кнопку К2 для запуска автомобиля и активации инфракрасного обнаружения препятствий. При обнаружении препятствия на пути автомобиль автоматически его объедет.

3) Суть эксперимента

Принцип работы ИК-датчика заключается в использовании отражающих свойств предметов. При наличии в определенном диапазоне препятствия ИК-лучи отражаются от этого препятствия и попадают в приемник. В эксперименте используются два ИК-датчика, подключенные к плате Raspberry Pi, предназначенные для обнаружения препятствий с последующим выполнением соответствующих действий по предотвращению столкновений.

4) Этапы эксперимента

4-1 Электрическая схема

Схема подключения платы Arduino

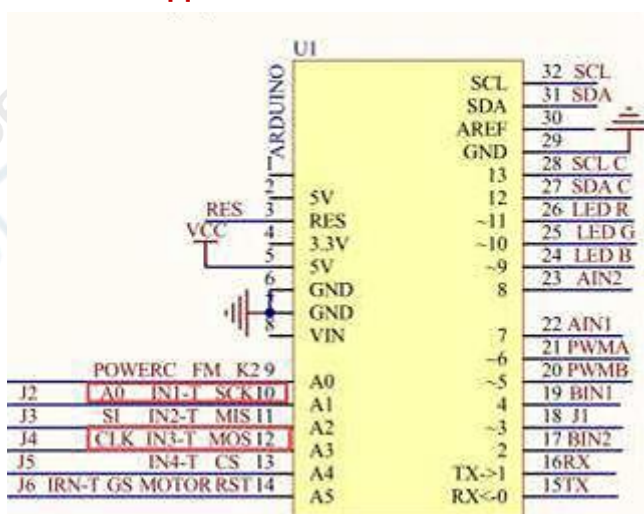
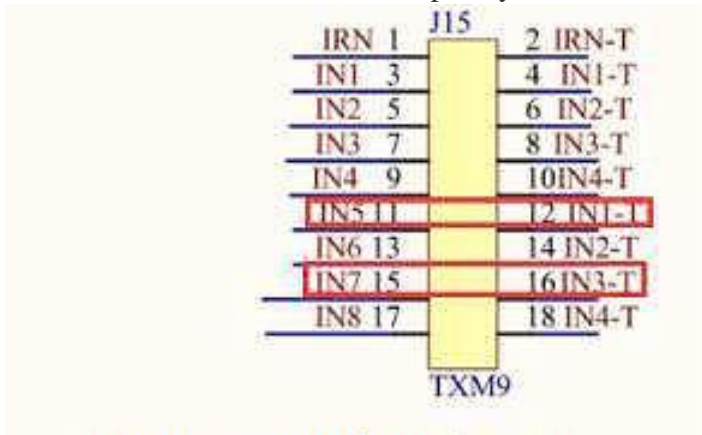


Схема подключения платы Raspberry Pi



Переключатель для выбора функции Arduino

4-1 Переключатель для выбора функции Arduino



4-2 Подключение правого и левого ИК-датчиков

4-2 В соответствии с электрической схемой:

Левый ИК-датчик----- А3

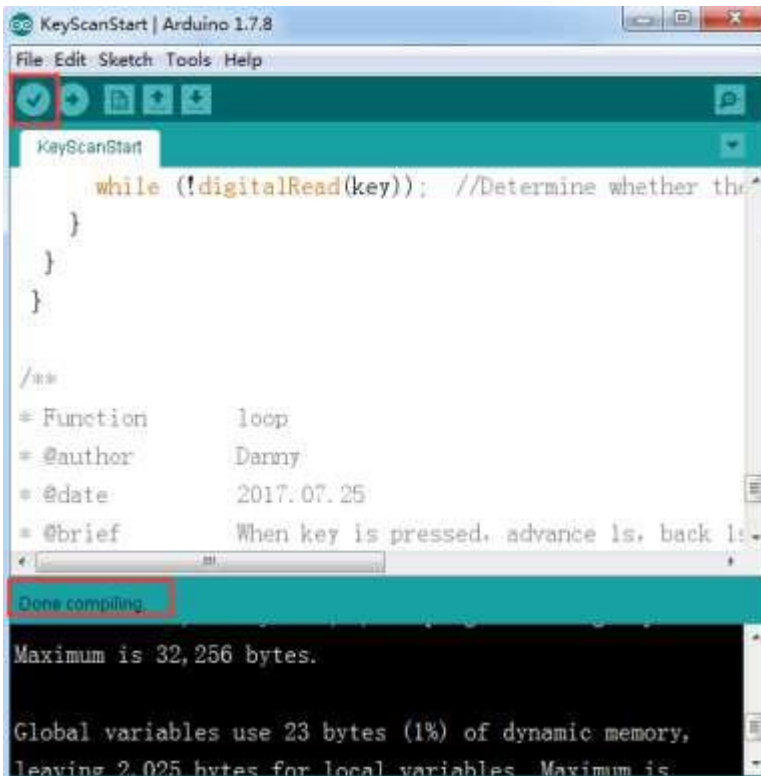
Правый ИК-датчик-----А1

(Примечание: для записи кода используется библиотека PI).

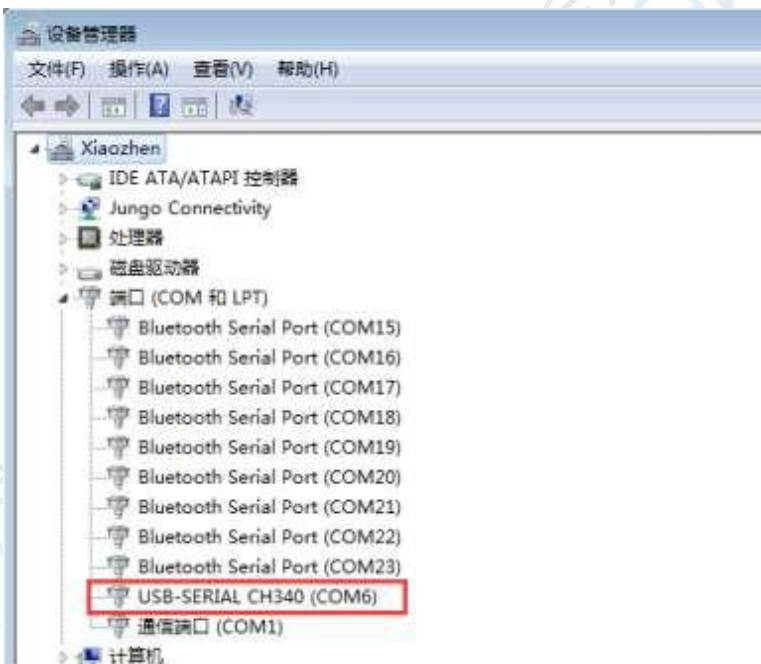
(Примечание: для получения оптимальных результатов чувствительность ИК-датчика может быть настроена потенциометром на ИК-модуле).

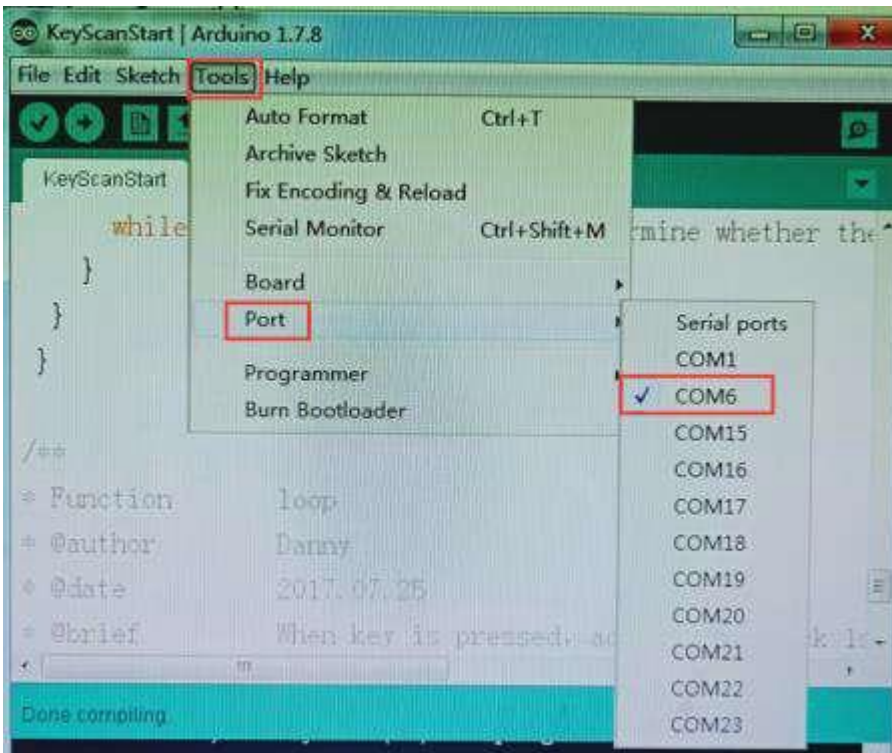
4-3 Программный код

1. Откройте файл с кодом эксперимента: **infrared_avoid.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).

