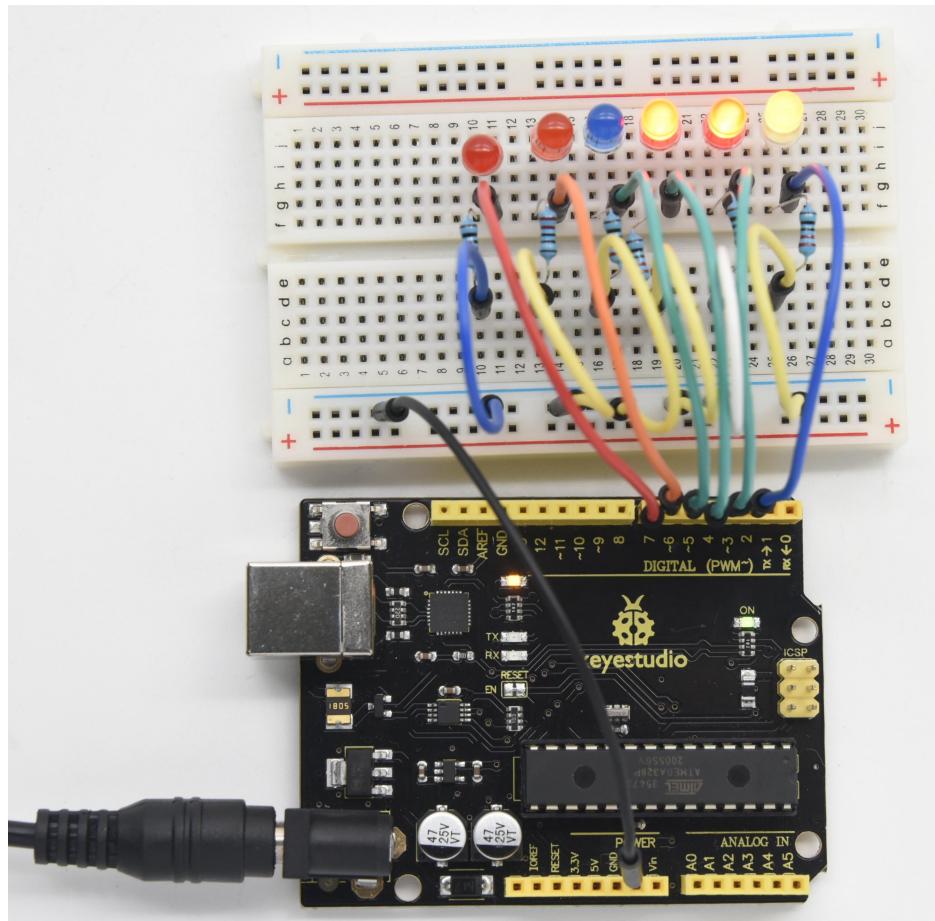


||||||||||||||||||||||||||

5. результат теста

Вы можете видеть, как светодиоды мигают последовательно.



Проект 6: светодиодный индикатор с кнопочным управлением



1. Введение

Порт ввода / вывода означает интерфейс для ВХОДА и ВЫХОДА. До сих пор мы

использовали только функцию ВЫВОД.

В этом эксперименте мы попытаемся использовать функцию INPUT, которая должна

считывать выходное значение устройства, подключенного к нему. Мы используем 1

кнопку и 1 светодиод для ввода и вывода, чтобы вы могли лучше понять функцию

ввода / вывода.

Кнопочный переключатель, знакомый большинству из нас, представляет собой компонент значения

переключателя (цифрового значения). При нажатии цепь находится в замкнутом (проводящем)

состоянии.

2. Требуется оборудование

- Плата V4.0 или плата MEGA 2650 * 1 кнопочный
- переключатель * 1
- Красный светодиод M5 * 1
- 220 Ом Резистор * 1
- Резистор 10 кОм * 1
- Макетная плата * 1
- Перемычка макетной платы * 6
- USB-кабель * 1

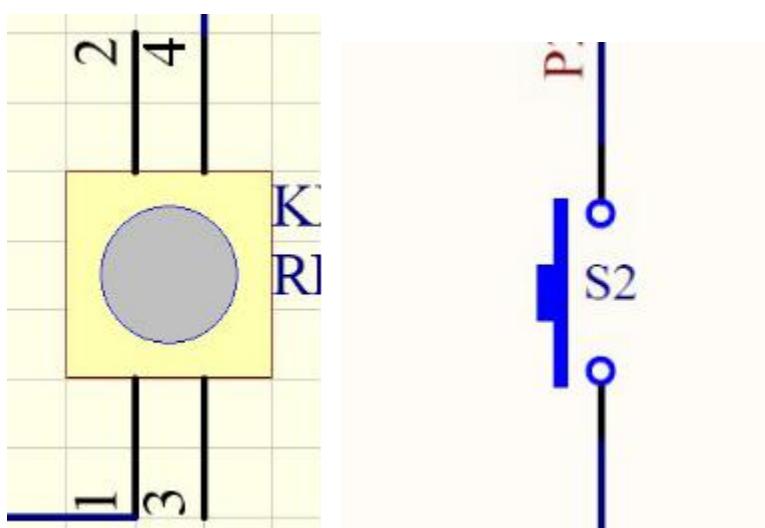
3. Мало знаний

Я считаю, что кнопочный переключатель обычен и популярен среди людей. Он принадлежит к компоненту количества переключателя (цифрового количества). Состоит из нормально разомкнутого и нормально замкнутого контактов, принцип его работы аналогичен обычному переключателю.

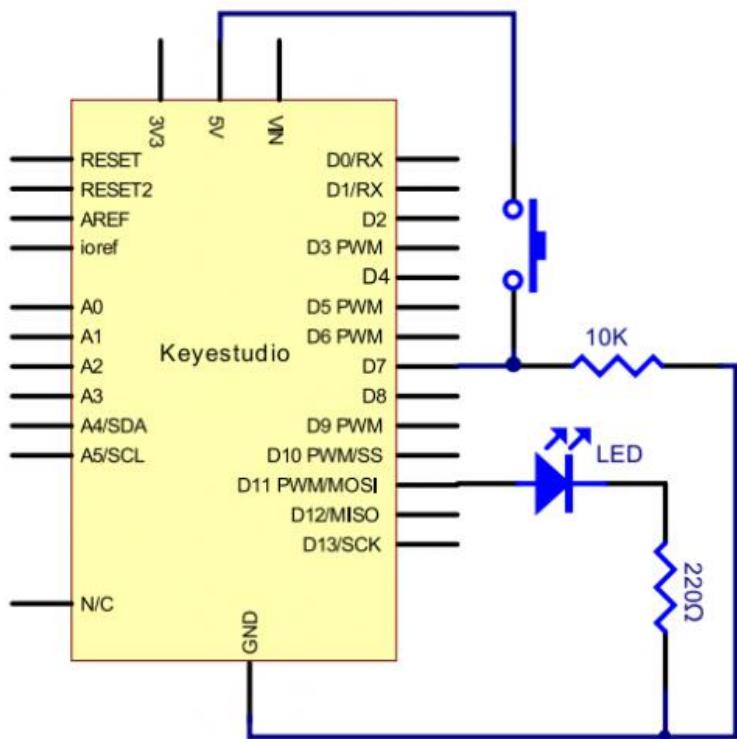
Когда нормально открытый контакт выдерживает давление, цепь находится в рабочем состоянии; однако, когда это давление исчезает, нормально открытый контакт возвращается в исходное состояние, то есть в выключенное состояние. Давление - это действие, которое мы переключаем на кнопку.



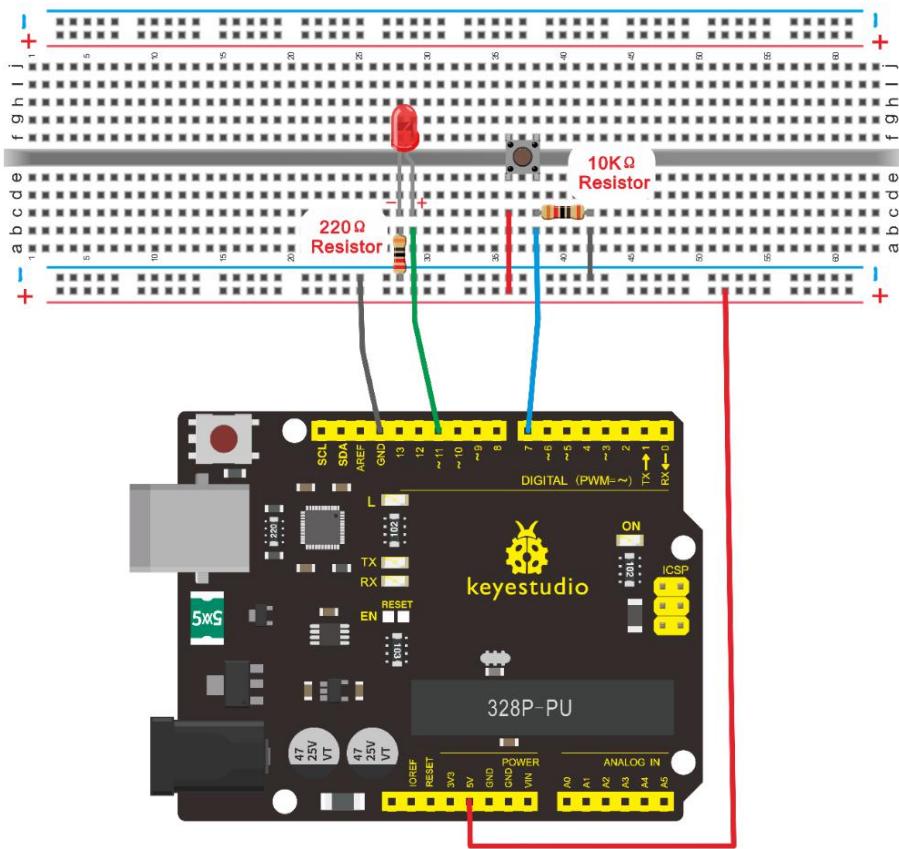
Принципиальные схемы:



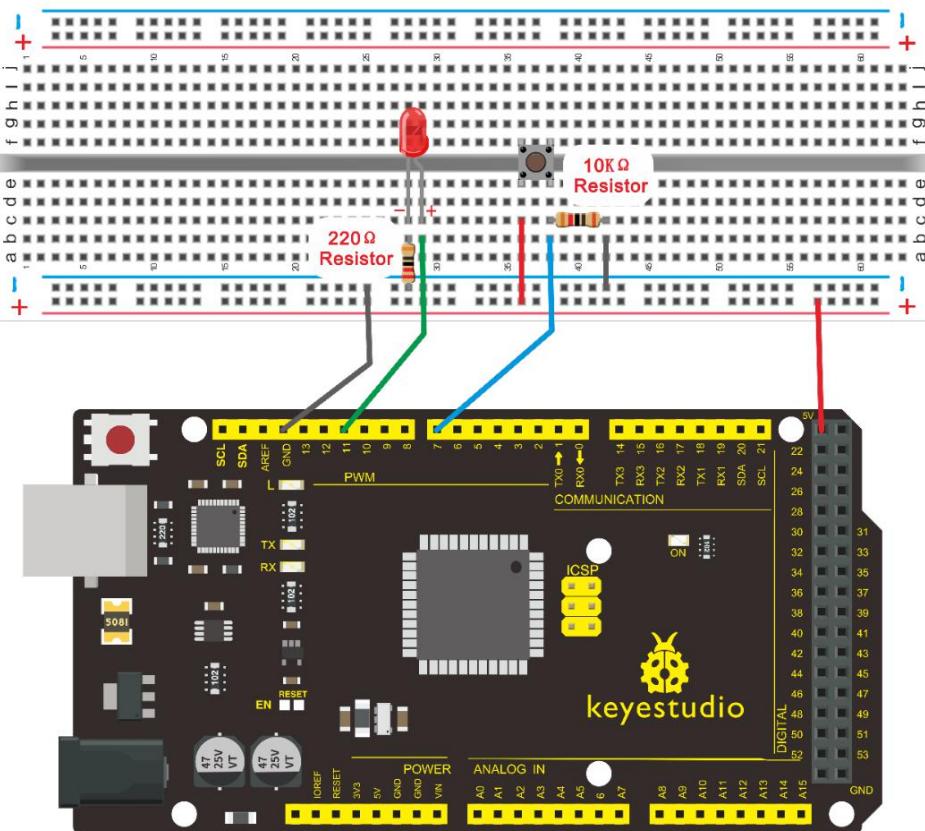
4. Схема подключения



Подключение для V4.0



Подключение для 2560



5. образец кода

Теперь приступим к компиляции. Когда кнопка нажата, загорится светодиод. Судя по предыдущему исследованию, кодирование должно быть для вас простым.

В этой программе мы добавляем изложение суждения. Здесь мы используем **если ()** заявление.

IDE Arduino основана на языке C, поэтому операторы языка C, такие как while, switch и т. Д., Безусловно, могут быть использованы для программы Arduino.

Когда мы нажимаем кнопку, вывод 7 будет выводить высокий уровень. Мы можем запрограммировать вывод 11 на вывод высокого уровня и включить светодиод. Когда вывод 7 выводит низкий уровень, вывод 11 также выводит низкий уровень, и светодиод остается выключенным.

```
/*
супер обучающий комплект keyestudio Project
6
Кнопка
http://www.keyestudio.com
*/
int ledpin = 11; // инициализировать вывод 11 int inpin =
7; // инициализировать вывод 7
```

```
int val; // определяем val

void setup ()
{
pinMode (ledpin, OUTPUT); // установить вывод светодиода как «выход» pinMode
(inpin, INPUT); // установить вывод кнопки как «вход»}

пустой цикл ()

{
val = digitalRead (inpin); // считываем значение уровня контакта 7 и назначаем if на val if (val ==
LOW) // проверяем, нажата ли кнопка, если да, включаем светодиод {digitalWrite (ledpin, LOW);}

еще

{digitalWrite (ledpin, HIGH);}

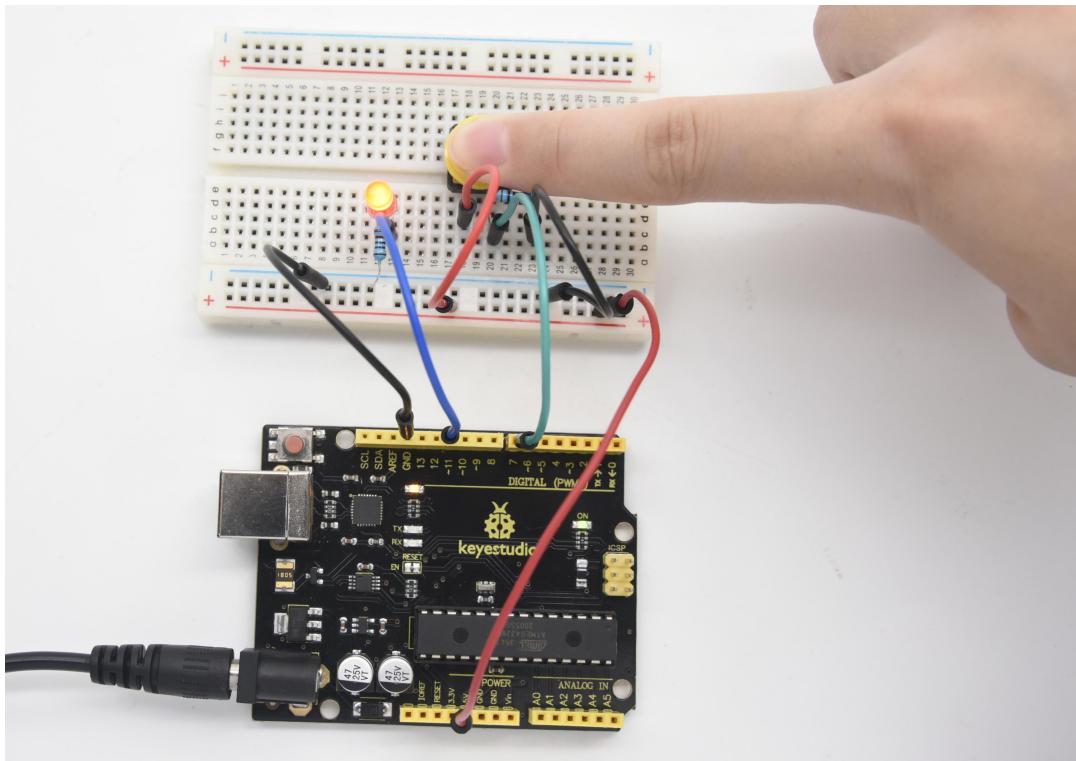
}

||||||||||||||||||||||||||||||||||||||||
```

6. результат теста

Когда кнопка нажата, светодиод горит, в противном случае светодиод не горит.

На этом эксперимент с кнопочным светодиодом завершен.



Простой принцип этого эксперимента широко используется в различных схемах

и электроприборах. Вы легко можете встретить это в повседневной жизни.

Типичный пример: когда вы нажимаете определенную клавишу на телефоне,

включается подсветка.

Проект 7: Активный зуммер



1. Введение

Активный зуммер широко используется в качестве звукового элемента на компьютере, принтере, будильнике, электронной игрушке, телефоне, таймере и многом другом. Имеет внутренний источник вибрации. Просто подключите его к источнику питания 5 В, он может гудеть постоянно.

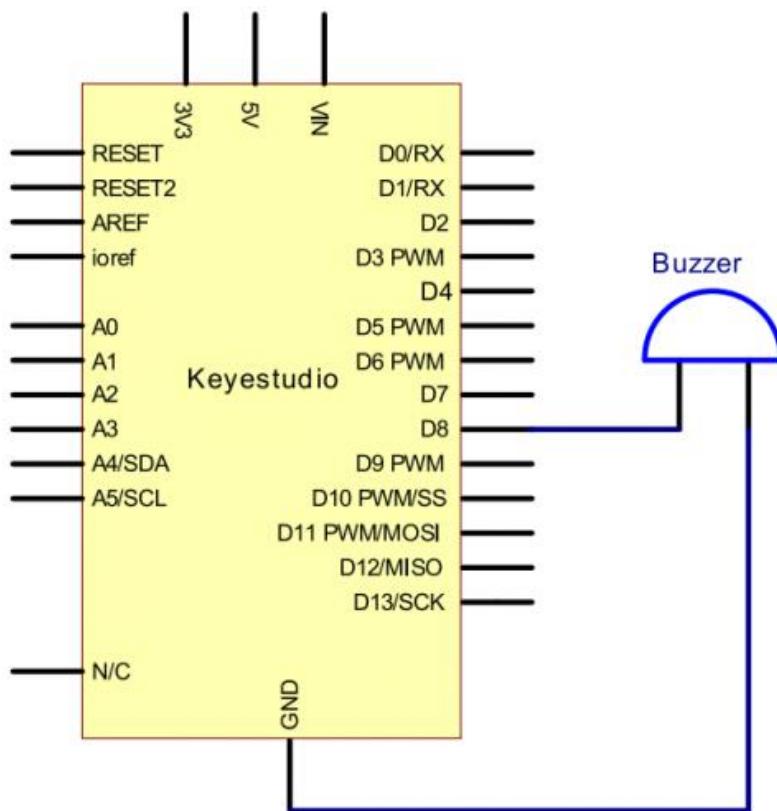
2. Требуется оборудование

- Плата V4.0 или плата MEGA 2650 * 1
- Зуммер * 1
- Макетная плата * 1
- Перемычка макетной платы * 2
- USB-кабель * 1

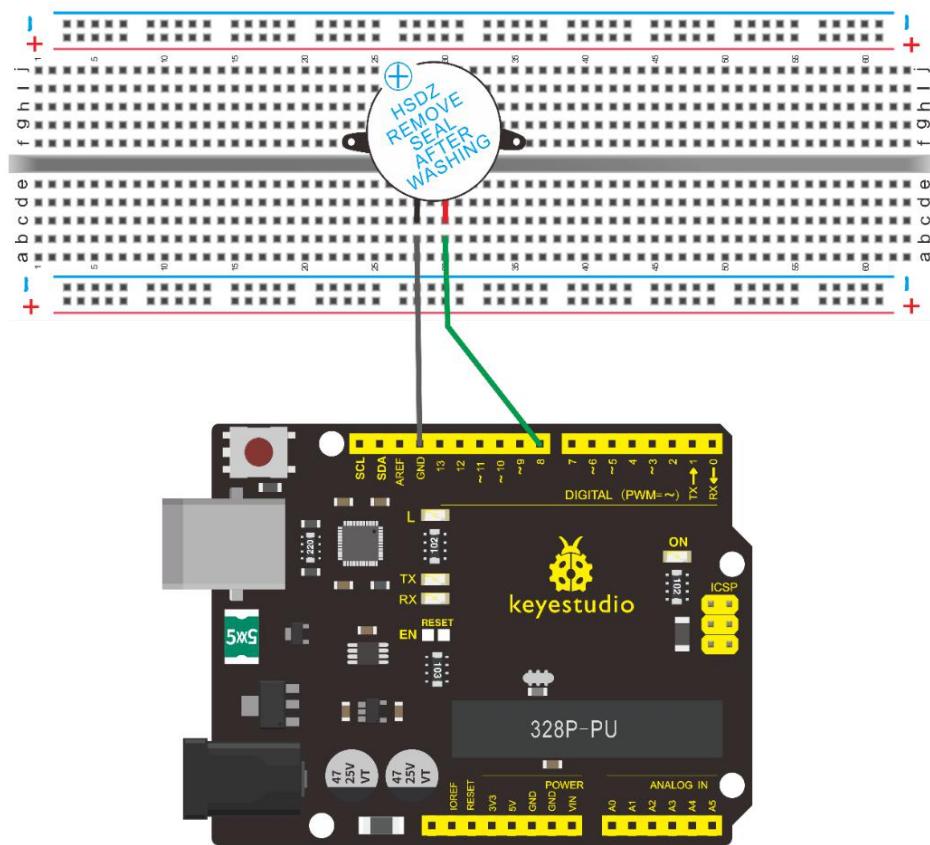
3. Мало знаний

Существует два типа зуммера: активный зуммер и пассивный зуммер. В этом уроке мы будем использовать Micro:bit для включения активного зуммера. Активный зуммер внутри имеет простую схему генератора, которая может преобразовывать постоянный постоянный ток в импульсный сигнал определенной частоты. Когда активный зуммер получает высокий уровень, он издает звуковой сигнал.

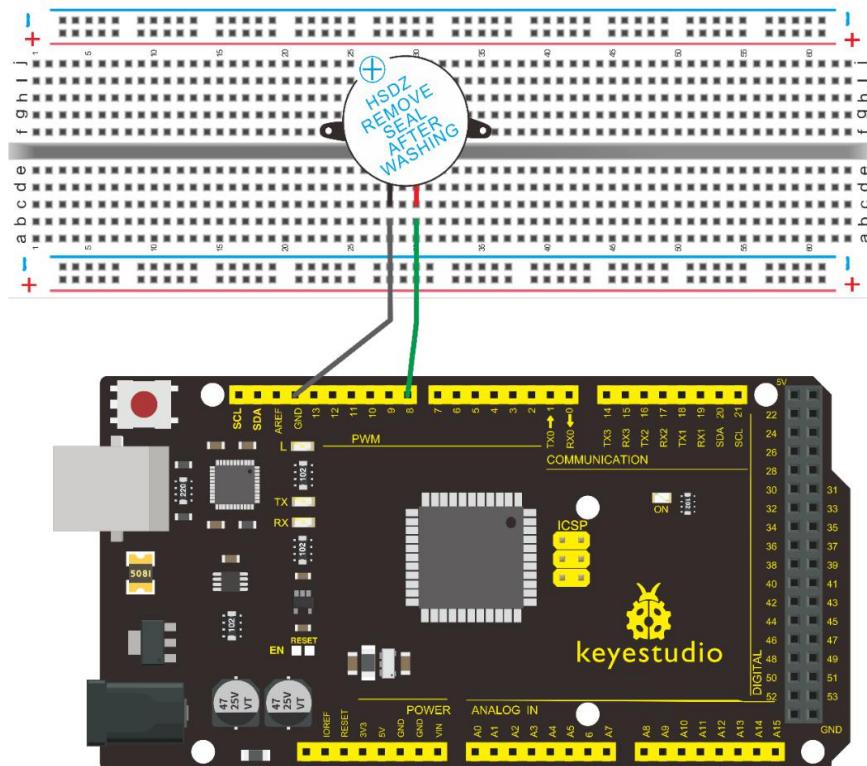
4. Схема подключения



Подключение для V4.0



Подключение для 2560



При подключении цепи обратите внимание на положительный и отрицательный полюса зуммера. На фото вы видите красные и черные линии.
Когда схема закончена, вы можете начинать программирование.

5. образец кода

Программа простая. Вы управляете зуммером, выводя высокий / низкий уровень.

```
/*
```

супер обучающий комплект keyestudio Project

7

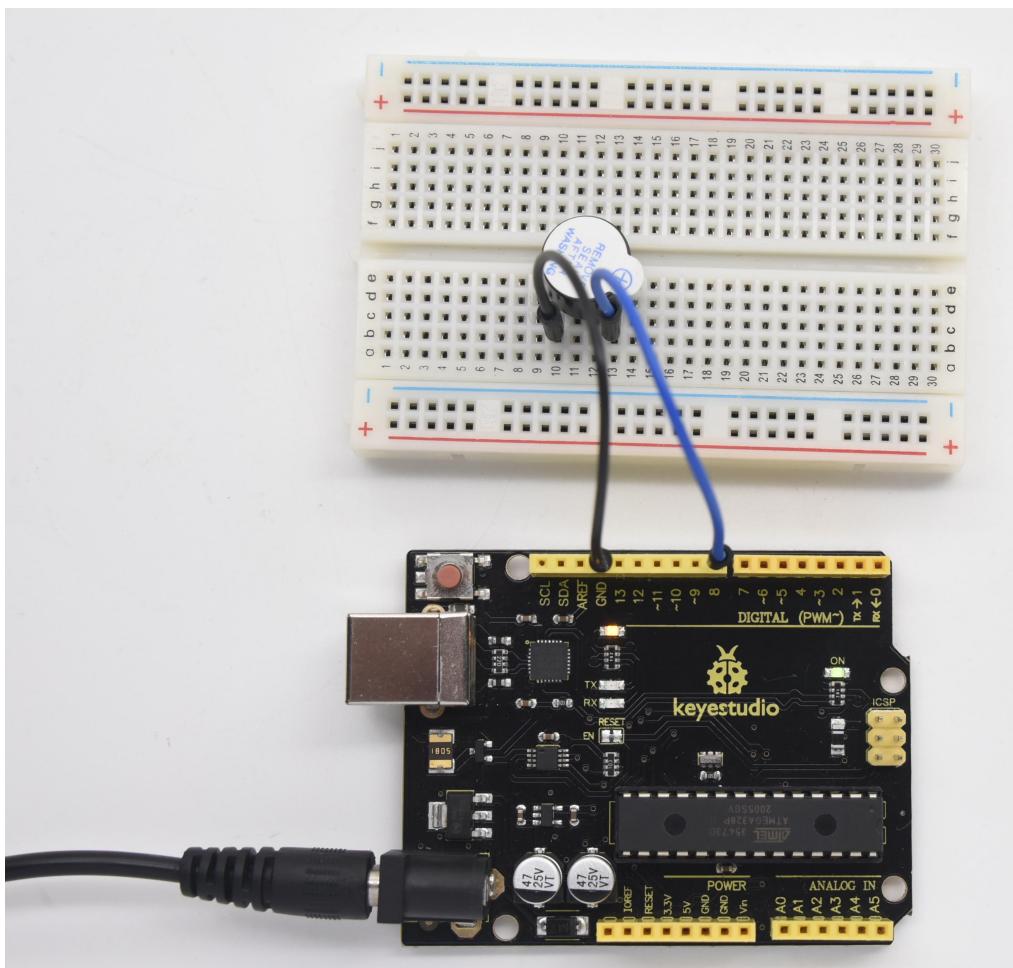
Активный зуммер

<http://www.keyestudio.com>

```
* /  
  
int buzzer = 8; // инициализируем цифровой вывод ввода-вывода, который управляет зуммером void setup  
{  
()  
  
    pinMode (buzzer, OUTPUT); // устанавливаем режим вывода как «output»  
  
}  
  
пустой цикл ()  
  
{  
  
    digitalWrite (зуммер, ВЫСОКИЙ); // воспроизводим звук}  
  
||||||||||||||||||||||||||||||||||||||||
```

6. результат теста

После загрузки программы эксперимент с зуммером завершен. Вы видите, что звонит зуммер.



Проект 8: Пассивный зуммер



1. Введение

Мы можем использовать Arduino для выполнения множества интерактивных работ.

Чаще всего используется акустооптический дисплей. Все

предыдущий эксперимент как-то связан со светодиодами. Однако схема в этом эксперименте может воспроизводить звук. Обычно эксперимент проводится с зуммером, но не с динамиком, в то время как зуммер проще и проще в использовании.

Зуммер, который мы здесь представили, является пассивным зуммером. Он не может быть активирован сам по себе, но с помощью внешних частотных импульсов. Разная частота дает разный звук. Мы можем использовать Arduino для кодирования мелодии песни, что довольно весело и просто.

2. Требуется оборудование

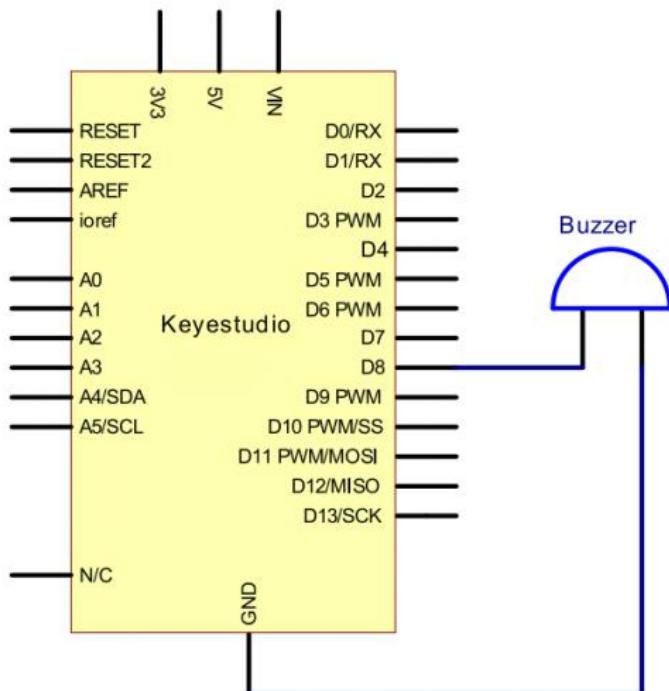
- Плата V4.0 или плата MEGA 2650 * 1 Пассивный
- зуммер * 1
- Макетная плата * 1
- Перемычка макетной платы * 2
- USB-кабель * 1

3. Мало знаний

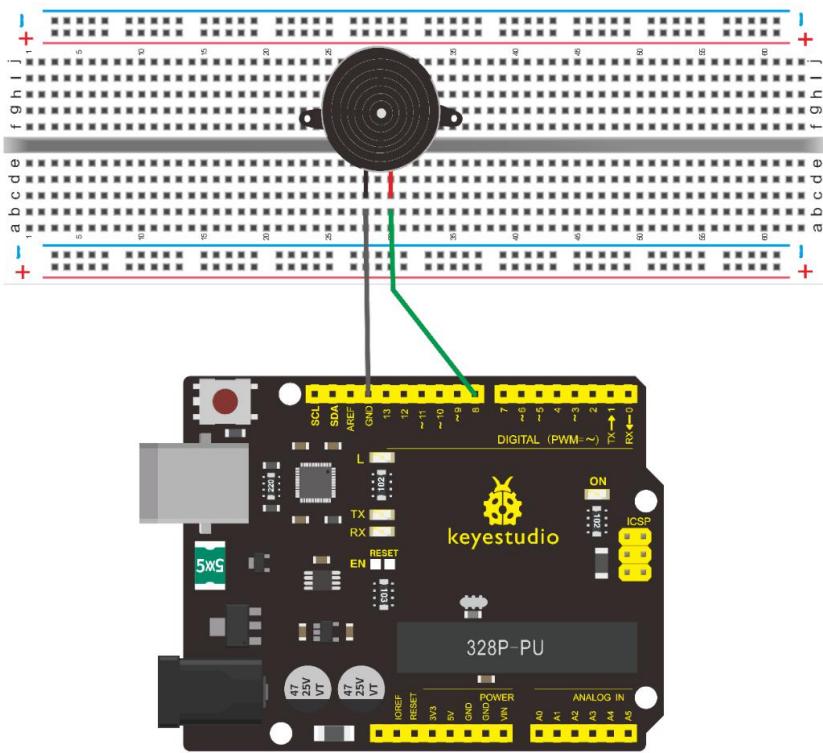
Пассивный зуммер представляет собой встроенный электронный зуммер без источника вибрации внутри. Он должен управляться прямоугольной волной 2К-5К вместо сигналов постоянного тока. Между двумя зуммерами разница небольшая, но когда штыри двух зуммеров подняты, пассивный зуммер становится зеленым.

печатная плата, а та, что заклеена винилом, представляет собой активный зуммер.

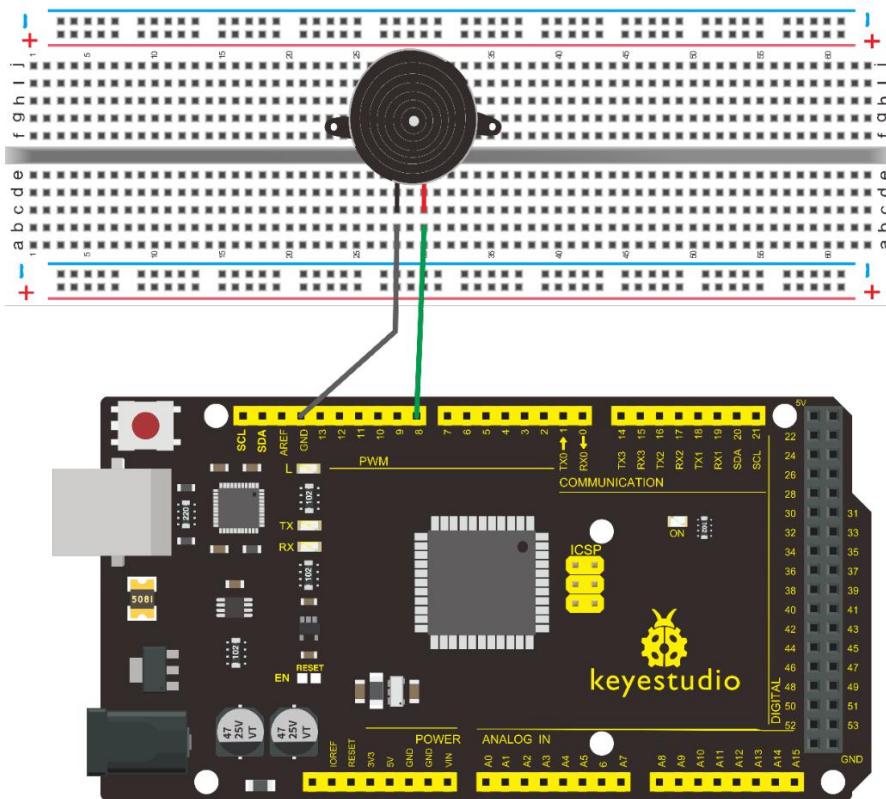
4. схема подключения



Подключение для V4.0



Подключение для 2560



5. образец кода

```
/*  
  
супер обучающий комплект keyestudio Project  
  
8  
  
Пассивный зуммер  
  
http://www.keyestudio.com  
  
*/  
  
int buzzer = 8; // выбираем цифровой вывод ввода-вывода для зуммера void setup ()  
  
{  
  
pinMode (buzzer, OUTPUT); // устанавливаем шаблон вывода цифрового ввода-вывода, OUTPUT для вывода  
  
}  
  
пустой цикл ()  
  
{unsigned char i, j; // определяем переменную while (1)  
  
{for (i = 0; i <80; i++) // вывод частотного звука {digitalWrite  
(buzzer, HIGH); // звук  
  
delay (1); // задержка 1 мс  
  
digitalWrite (buzzer, LOW); // не звучит  
  
delay (1); // задержка в мс  
  
}  
  
for (i = 0; i <100; i++) // вывод частотного звука
```

```

{digitalWrite (buzzer, HIGH); // звук

delay (2); // задержка 2 мс

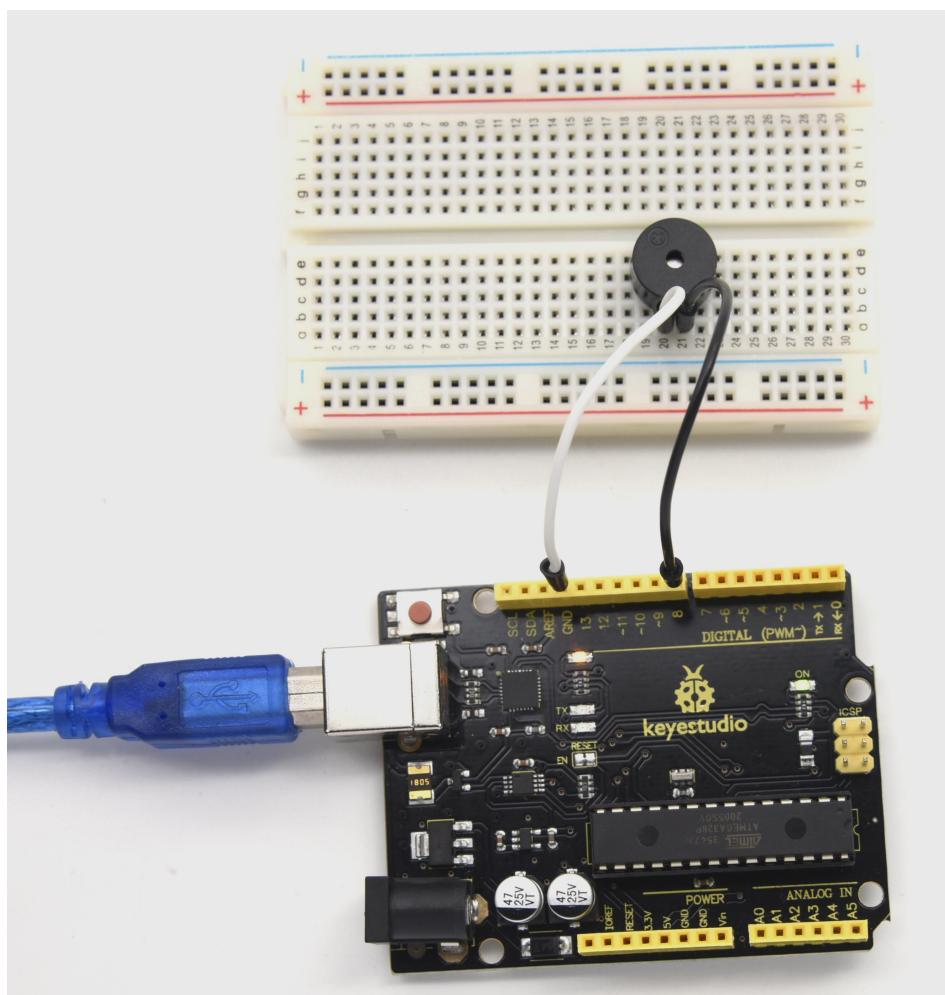
digitalWrite (buzzer, LOW); // не звучит

delay (2); // задержка 2 мс

}}
```

6. результат теста

После загрузки программы эксперимент с зуммером завершен, вы можете услышать звук зуммера.



Проект 9: светодиод RGB



1. Введение

Цветовой режим RGB является стандартом цвета в отрасли. Он получает различные цвета путем изменения трех цветовых каналов: красного (R), зеленого (G) и синего (B) и их объединения. RGB обозначает три цвета: красный, зеленый и синий.

В этом проекте мы используем Arduino для смешивания этих трех цветов в равных количествах для получения белого света.

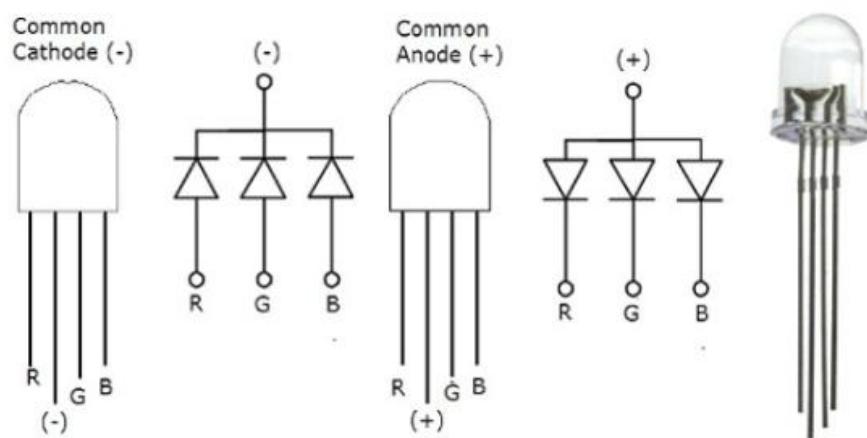
2. Требуется оборудование

- Плата V4.0 или плата MEGA 2650 * 1
- USB-кабель * 1
- RGB светодиод * 1
- Резистор * 3
- Проволочная перемычка макета * 5

3. Мало знаний

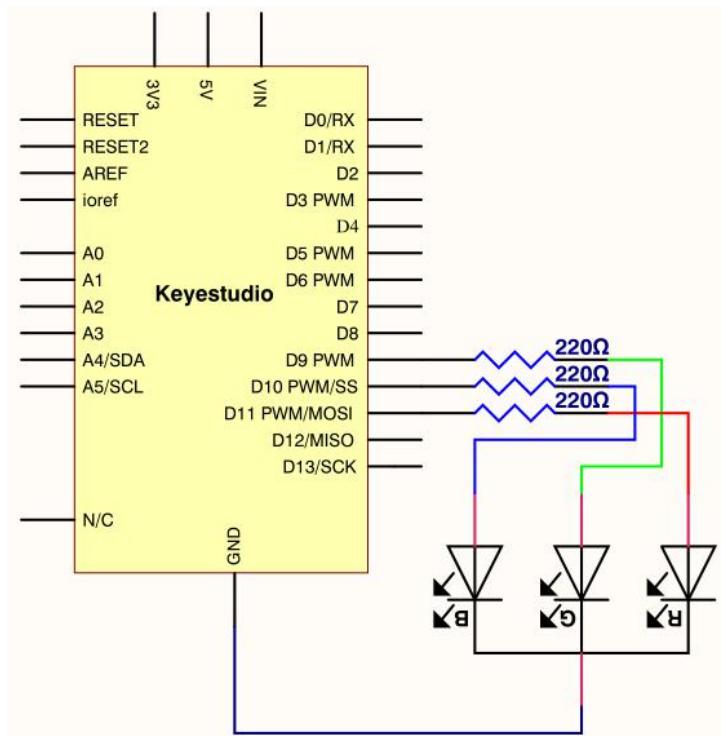
Мониторы в основном используют цветовой стандарт RGB, и все цвета на экране компьютера состоят из трех цветов: красного, зеленого и синего, смешанных в разных пропорциях.

RGB включает общий катодный RGB и общий анодный RGB.

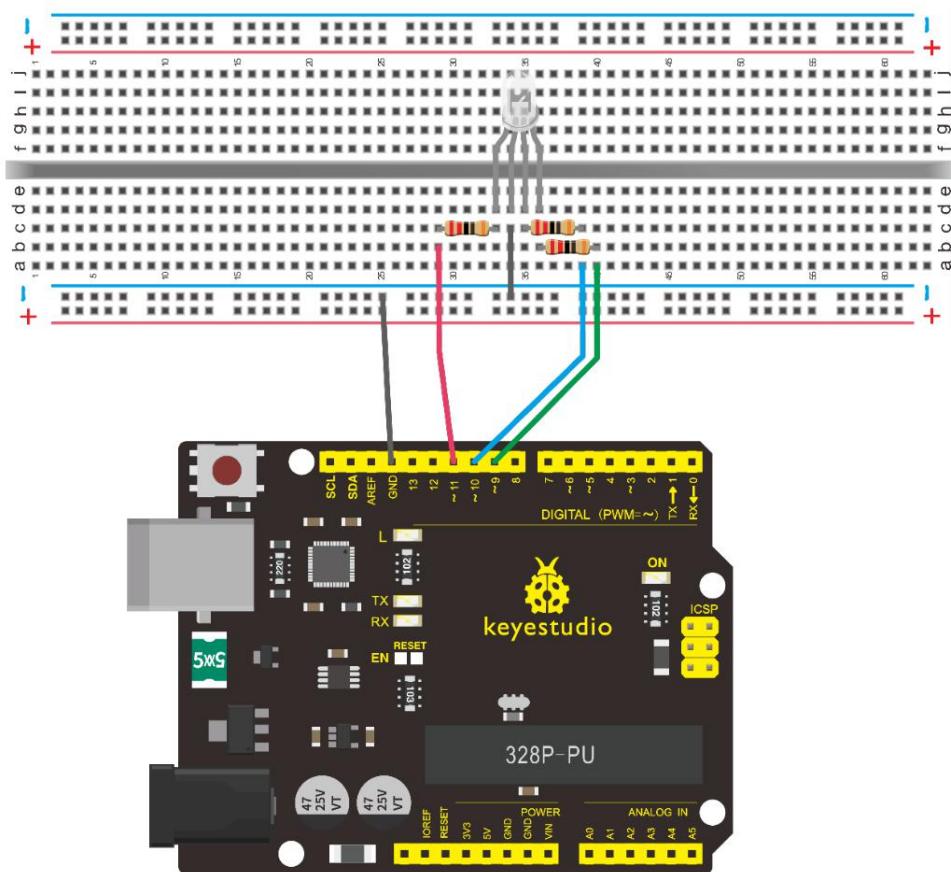


И мы могли регулировать яркость светодиода с помощью ШИМ

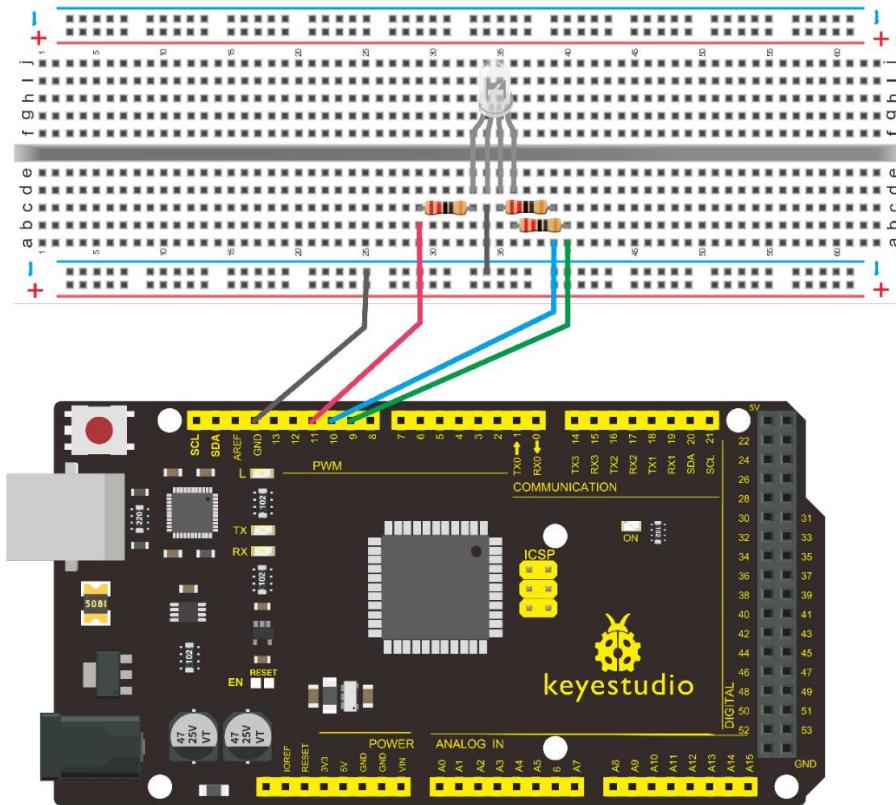
4. схема подключения



Подключение для V4.0



Подключение для 2560



5. образец кода

```
/*
```

```
супер обучающий комплект keyestudio Project
```

```
9
```

```
RGB
```

```
http://www.keyestudio.com
```

```
* /
```

```
int redpin = 11; // выбираем вывод красного светодиода int bluepin = 10; //
```

```
выбираем контакт для синего светодиода
```

```
int greenpin = 9; // выбираем вывод для зеленого светодиода int val;
```

```
void setup () {
```

```
    pinMode (редпин, ВЫХОД);
```

```
    pinMode (bluepin, ВЫХОД);
```

```
    pinMode (зеленый контакт, ВЫХОД);
```

```
    Serial.begin (9600);
```

```
}
```

```
пустой цикл ()
```

```
{
```

```
для (val = 255; val> 0; val--)
```

```
{
```

```
    analogWrite (11, val);
```

```
    analogWrite (10, 255-val);
```

```
    analogWrite (9, 128-val);
```

```
    задержка (1);
```

```
}
```

```
для (значение = 0; значение <255; значение ++)
```

```
{
```

```
    analogWrite (11, val);
```

```
    analogWrite (10, 255-val);
```

```
    analogWrite (9, 128-val);
```

задержка (1);

}

Serial.println (val, DEC);

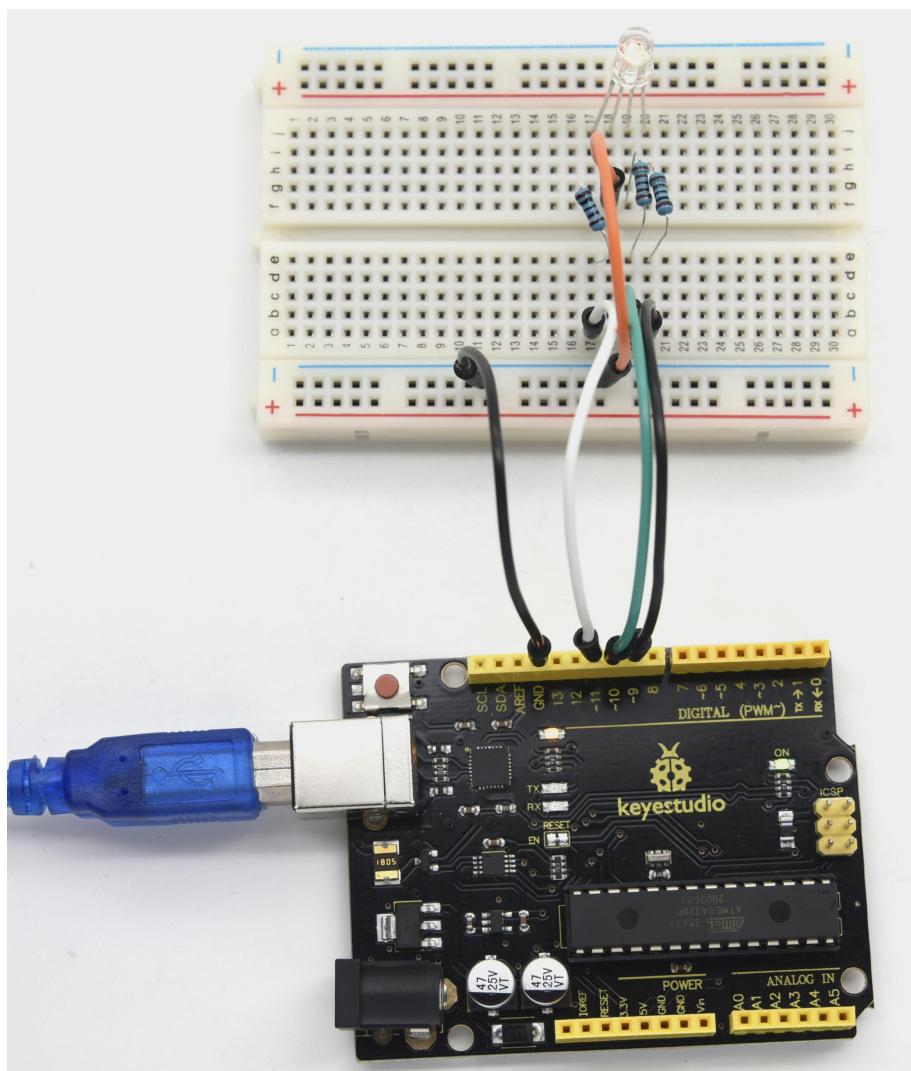
}

||||||||||||||||||||| /||||||| /||||||| /

6. результат теста

Непосредственно скопируйте приведенный выше код в IDE arduino, нажмите «Загрузить»,

подождите несколько секунд, вы увидите полноцветный светодиод.



Проект 10: Фоторезистор



1. Введение

После завершения всех предыдущих экспериментов вы можете получить некоторые базовые представления и знания о приложении Arduino. Мы представили цифровой вход и выход, аналоговый вход и ШИМ.

Теперь приступим к изучению сенсорных приложений.

Фоторезистор (Photovaristor) - это резистор, сопротивление которого зависит от силы падающего света. Он основан на фотоэлектрическом эффекте полупроводника. Если падающий свет интенсивен, его сопротивление уменьшается; если падающий свет слабый, сопротивление увеличивается.

2. Требуется оборудование

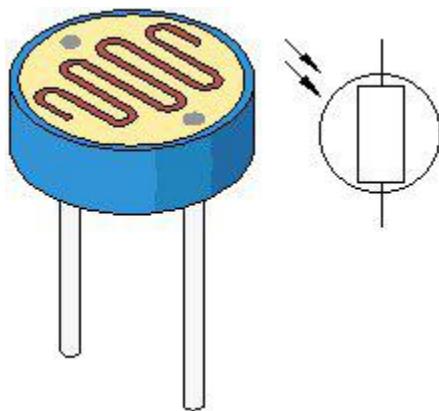
- Плата V4.0 или плата MEGA 2650 * 1
- Фоторезистор * 1
- Красный светодиод M5 * 1

- Резистор 10 кОм * 1
- Резистор 220 Ом * 1
- Макетная плата * 1
- Перемычка макетной платы * 5
- USB-кабель * 1

3. Мало знаний

Фотоваристор обычно применяется для измерения света, управления освещением и фотоэлектрического преобразования (преобразования изменения света в изменение электричества).

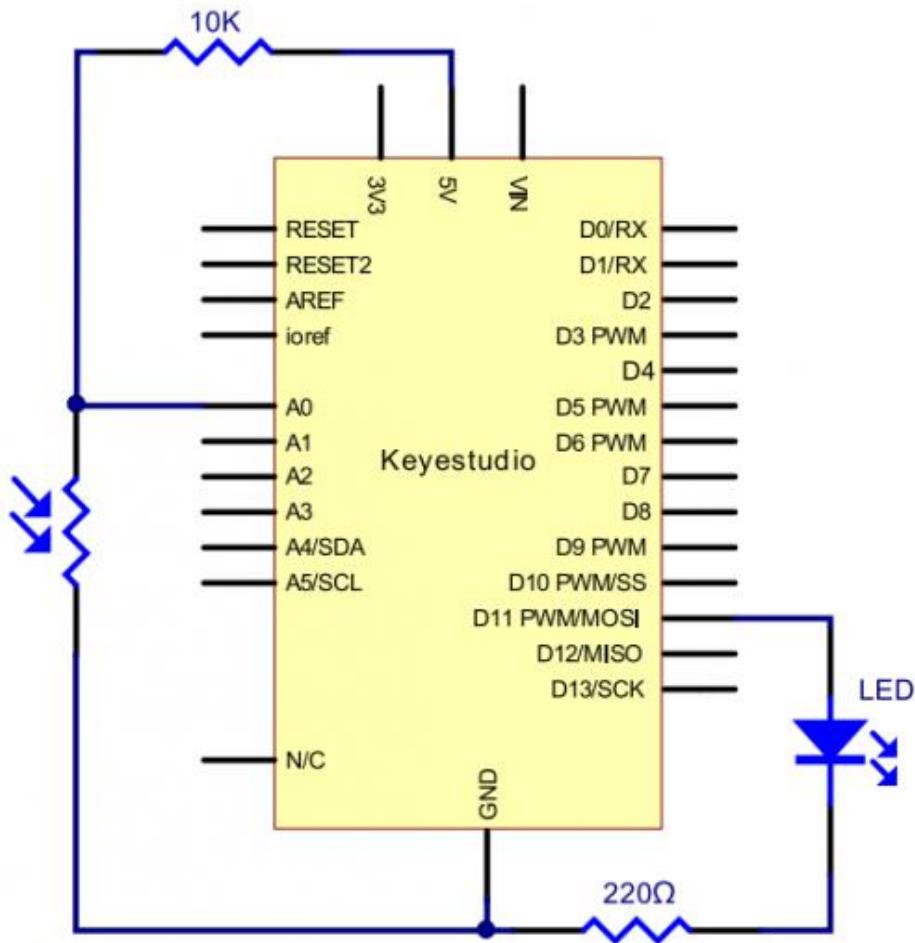
Фоторезистор также широко применяется в различных схемах управления освещением, таких как управление и регулировка света, оптические переключатели и т. д.



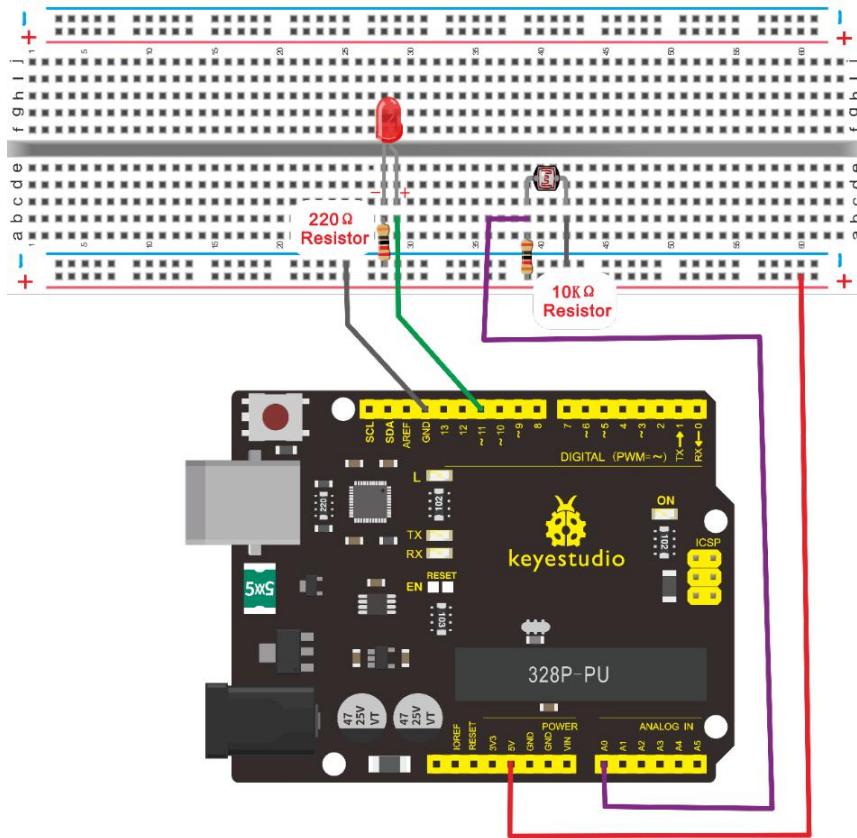
Мы начнем с относительно простого эксперимента по применению фотоваристора.

Фоторезистор - это элемент, который может изменять свое сопротивление при изменении силы света. Поэтому нужно прочитать аналоговое значение. Можно обратиться к эксперименту с ШИМ, заменив потенциометр на фоторезистор. Когда происходит изменение силы света, на светодиоде происходит соответствующее изменение.

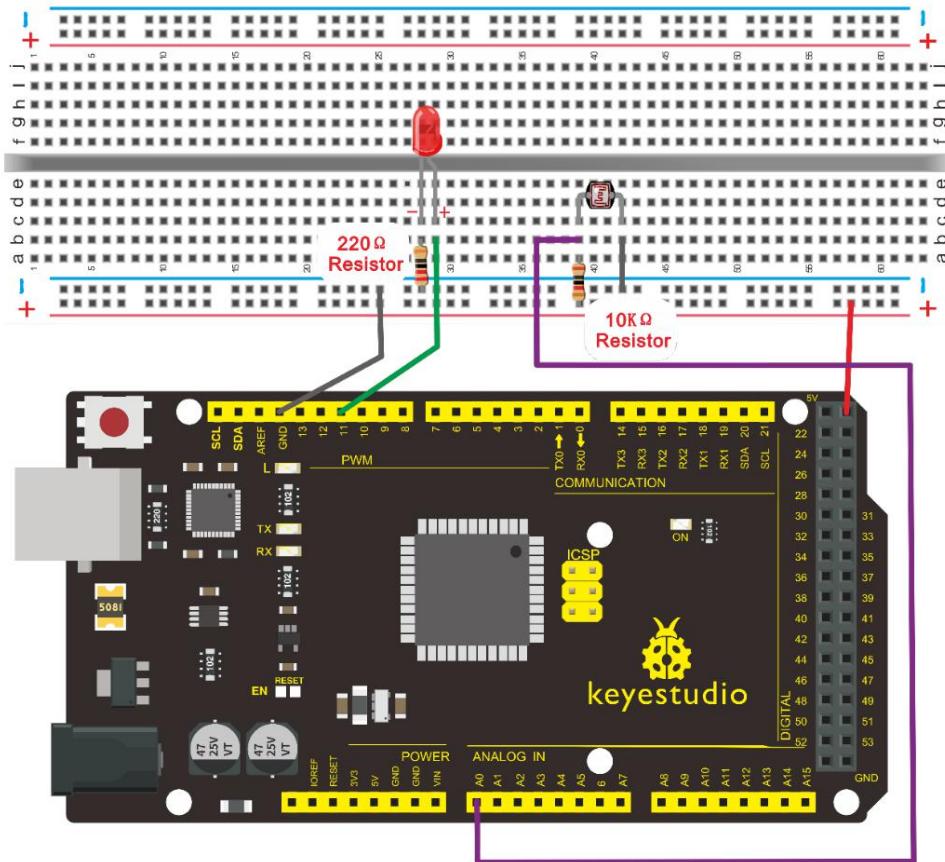
4. схема подключения



Подключение для V4.0



Подключение для 2560



5. образец кода

После подключения приступим к компиляции программы. Программа похожа на ШИМ.

Подробнее об изменениях см. В образце кода ниже.

```
/*  
keyestudio супер обучающий комплект  
Project 10  
Фото резистор  
http://www.keyestudio.com  
*/  
  
int potpin = 0; // инициализируем аналоговый вывод 0, связанный с фотоваристором  
  
int ledpin = 11; // инициализируем цифровой вывод 11, int val = 0; //  
инициализируем переменную val void setup ()  
  
{  
pinMode (ledpin, OUTPUT); // установить цифровой вывод 11 как «выход» Serial.begin (9600); //  
установить скорость передачи на «9600»  
}  
пустой цикл ()  
{  
val = analogRead (potpin); // считываем значение датчика и
```

назначить его val

```
Serial.println (val); // выводим значение val
```

```
analogWrite (ledpin, val / 4); // устанавливаем яркость (максимальное значение 255)
```

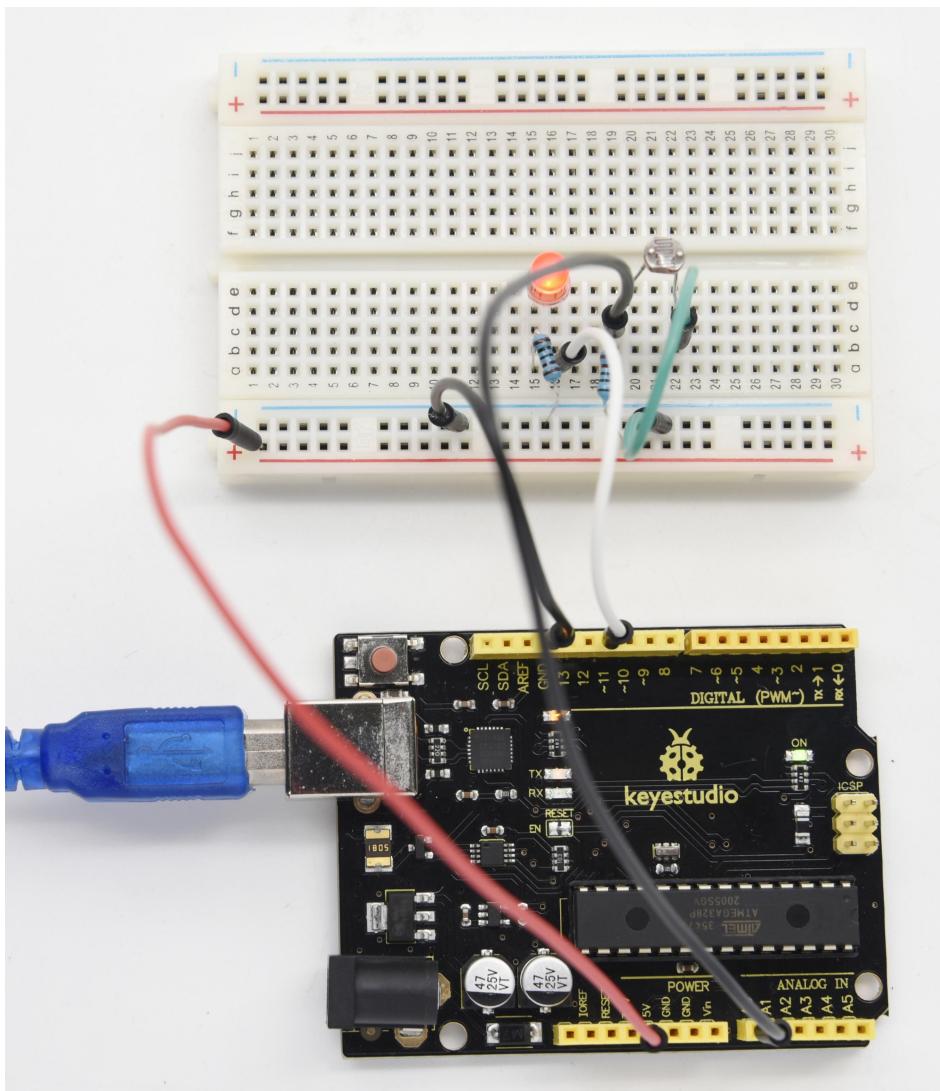
```
delay (10); // ждем 0,01}
```

```
||||||||||||||||||||||||||||||||||||||||||
```

6. результат теста

После загрузки программы вы можете изменить силу света вокруг фотодиода и увидеть соответствующее изменение яркости светодиода.

В повседневной жизни фотодиоды находят самое разнообразное применение. На его основе можно делать другие интересные интерактивные проекты.



Проект 11: Датчик пламени



1. Введение

Датчик пламени (инфракрасный приемный триод) специально используется роботами для поиска источника огня. Этот датчик имеет высокую чувствительность к пламени.

2. Требуется оборудование

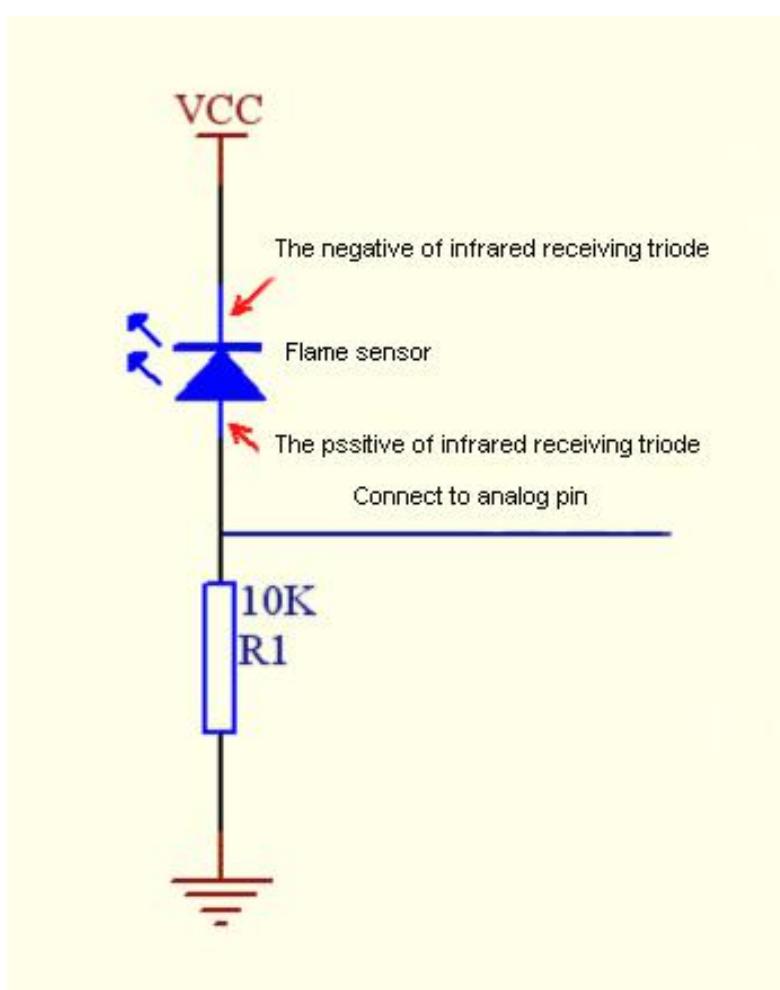
- Плата V4.0 или плата MEGA 2650 * 1
- Датчик пламени * 1
- Зуммер * 1
- Резистор 10 кОм * 1
- Перемычка макетной платы * 6
- USB-кабель * 1

3. Мало знаний

Датчик пламени основан на том принципе, что инфракрасные лучи очень чувствительны к пламени. Он имеет инфракрасную приемную трубку, специально разработанную для обнаружения пожара, а затем для преобразования яркости пламени в сигнал колеблющегося уровня. Затем сигналы поступают в центральный процессор и обрабатываются соответствующим образом.

Более короткий вывод приемного триода предназначен для отрицательного вывода, а другой - для положительного. Подключите отрицательный к выводу 5V, положительный к резистору; подключите другой конец резистора к GND,

подключите один конец перемычки к зажиму, который электрически подключен к плюсу датчика, а другой конец - к аналоговому выводу. Как показано ниже



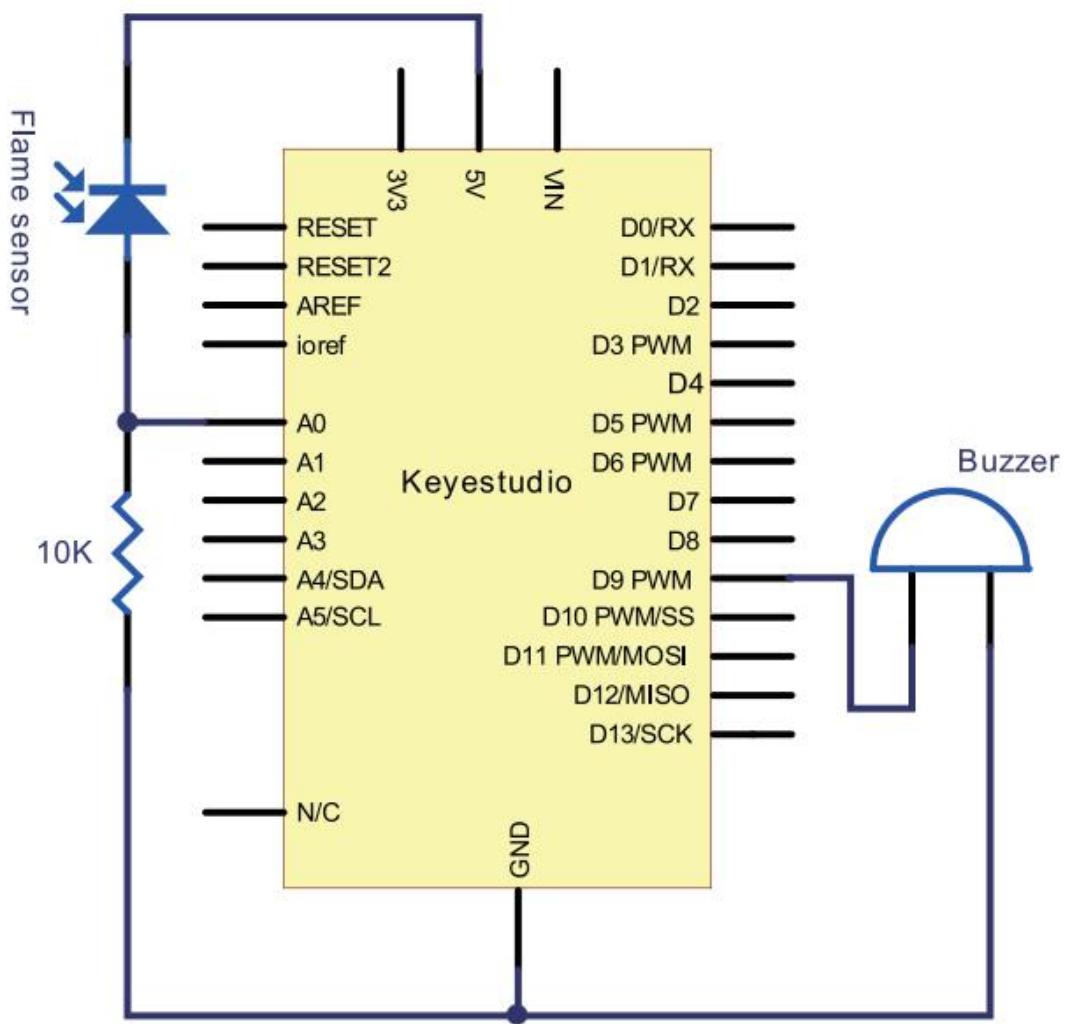
4. Принцип эксперимента.

Когда приближается пожар, значение напряжения, считывающееся с аналогового порта, будет другим. Если вы используете мультиметр, вы можете увидеть, что когда нет приближающегося огня, напряжение, которое он показывает, равно

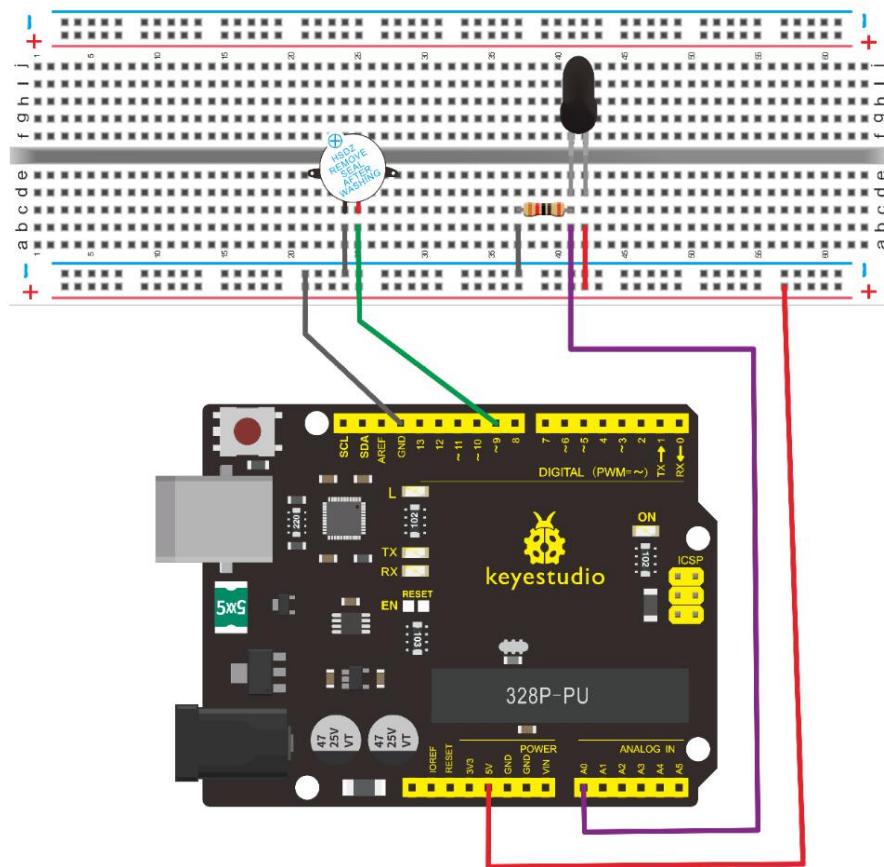
около 0,3 В; когда приближается пожар, считываемое напряжение составляет около 1,0 В. Чем ближе огонь, тем выше напряжение.

Итак, в начале программы вы можете инициализировать значение напряжения **я** (нет пожарной ценности); Затем непрерывно считывайте значение аналогового напряжения. **ј** и получить значение разницы **k = ji**; сравните **k** с 0,6 В (123 в двоичном формате), чтобы определить, приближается пожар или нет; если да, зуммер будет гудеть.

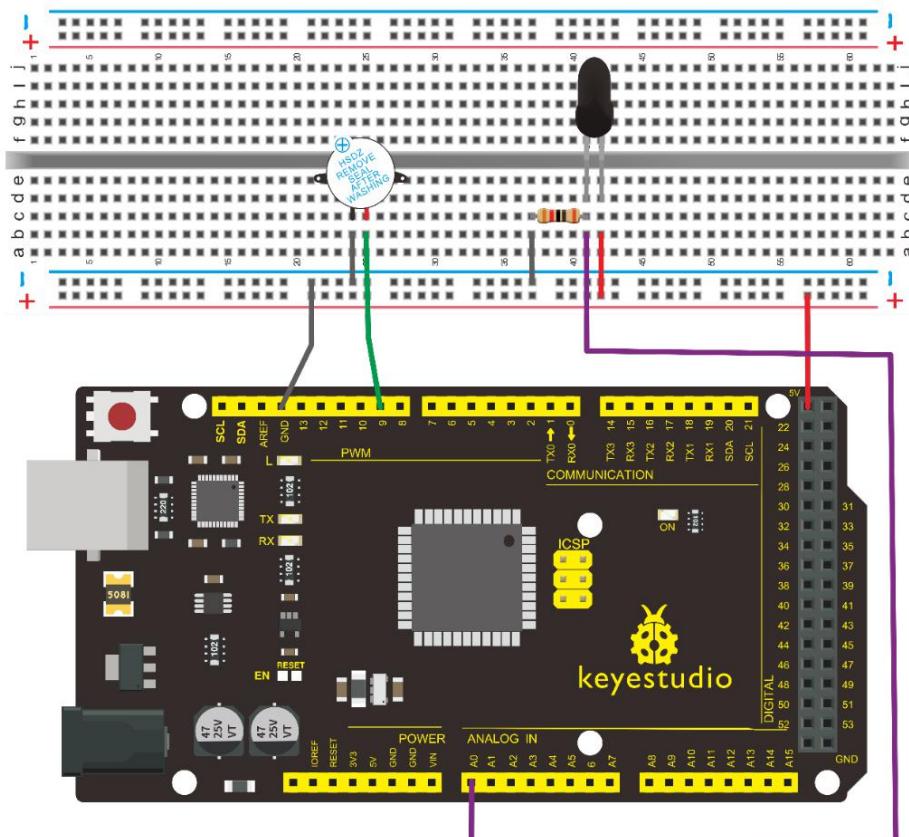
5. схема подключения



Подключение для V4.0



Подключение для MEGA 2560



6. образец кода

```
/*
супер обучающий комплект keyestudio
Project 11
Пламя
http // www.keyestudio.com
*/
int flame = 0; // выбираем аналоговый вывод 0 для датчика int Beep = 9; //
выбираем цифровой вывод 9 для зуммера int val = 0; // инициализируем
переменную
установка void ()
{
pinMode (Beep, OUTPUT); // устанавливаем вывод светодиода как «выход» pinMode
(flame, INPUT); // устанавливаем вывод зуммера как «ввод» Serial.begin (9600); //
устанавливаем скорость передачи на «9600»}

пустой цикл ()
{
val = analogRead (flame); // считываем аналоговое значение датчика Serial.println (val); //
выводим и отображаем аналоговое значение if (val> = 600) // когда аналоговое значение
больше 600, зуммер будет гудеть
```

```

{

digitalWrite (Beep, HIGH);

} еще

{

digitalWrite (Beep, LOW);

}

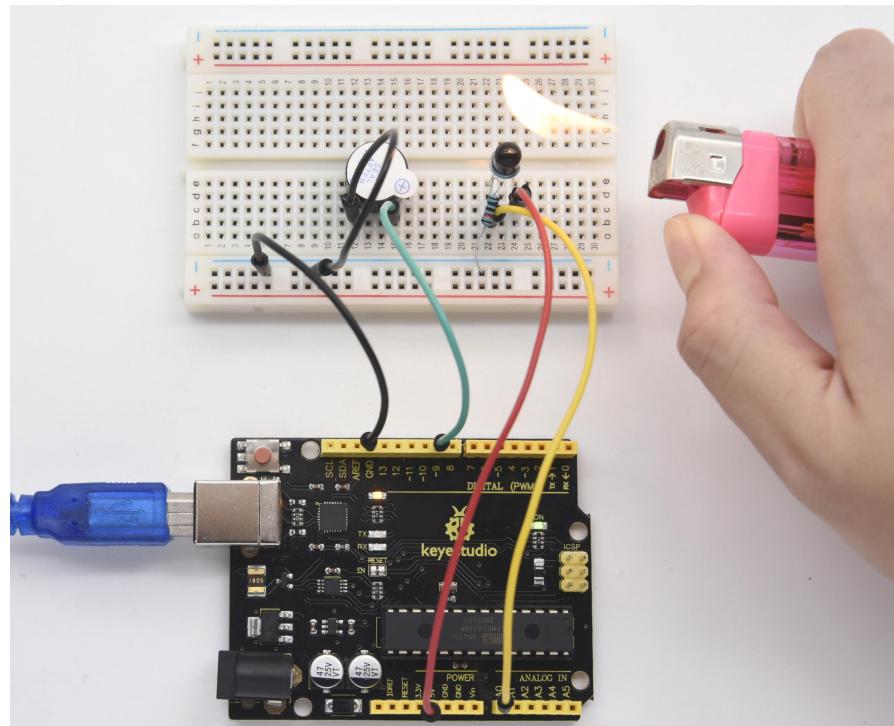
задержка (500);

}

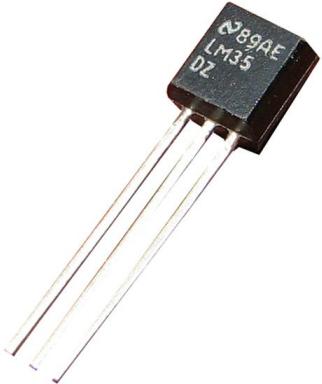
|||||||||||||||||||||||||||||||||||
```

7. результат теста

Эта программа может имитировать сигнал тревоги при пожаре. Когда нет огня, все нормально; при возгорании немедленно срабатывает сигнализация.



Проект 12: Датчик температуры LM35

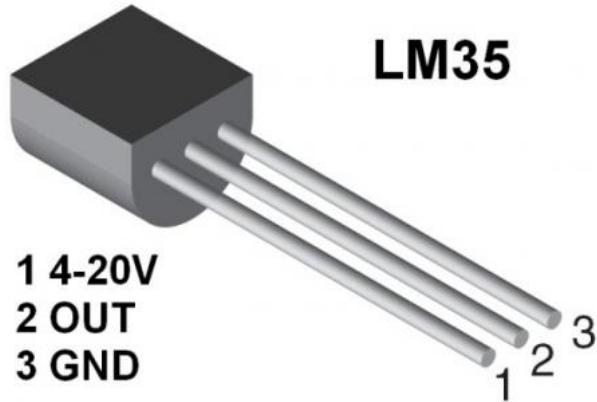


1. Введение

LM35 - обычный и простой в использовании датчик температуры. Не требует другого оборудования. Вам просто нужен аналоговый порт, чтобы он работал. Сложность заключается в компиляции кода для преобразования считываемого аналогового значения в температуру по Цельсию. В этом проекте мы расскажем, как использовать датчик температуры LM35.

2. Принцип работы

LM35 - широко используемый датчик температуры с множеством различных типов корпусов. При комнатной температуре он может достичь точности $\pm 1/4$ ° С без дополнительной калибровки.



Датчик температуры LM35 может выдавать разное напряжение при разной температуре

Когда температура равна 0 °C, он выдает 0 В; при увеличении на 1 °C выходное напряжение увеличится на 10 мВ.

Выходная температура составляет 0 °C 100 °C, формула преобразования выглядит следующим образом:

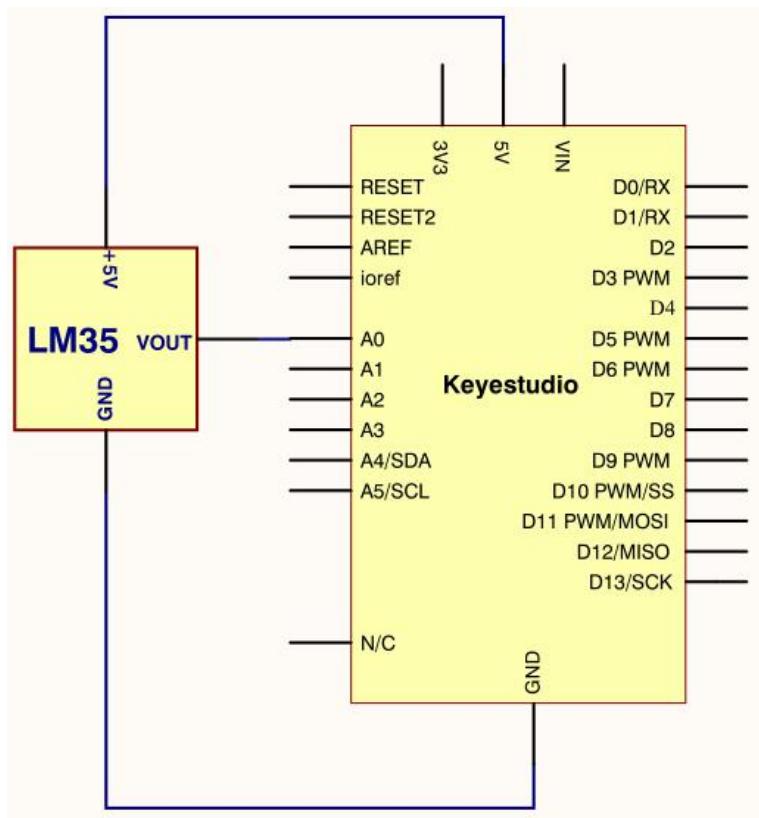
$$V_{\text{out_LM35}}(T) = 10 \text{ mV}/\text{°C} \times T^{\circ}\text{C}$$

3. требуется оборудование

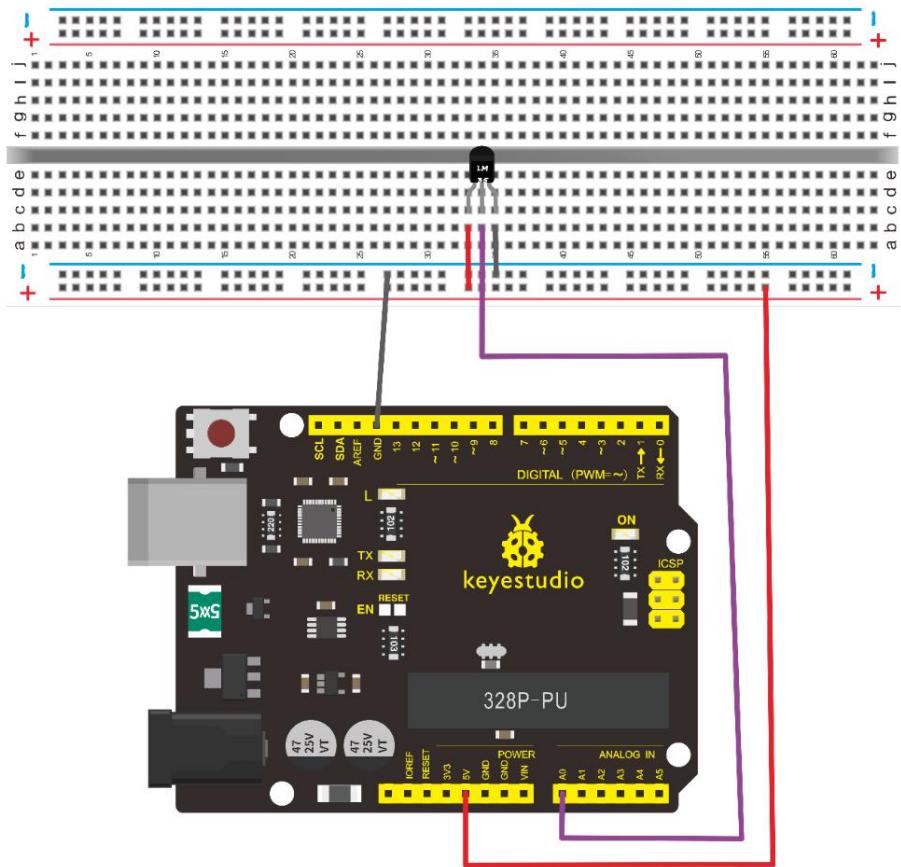
- Плата V4.0 или плата MEGA 2650 * 1
- LM35 * 1
- Макетная плата * 1
- Перемычка макетной платы * 5

- USB-кабель*

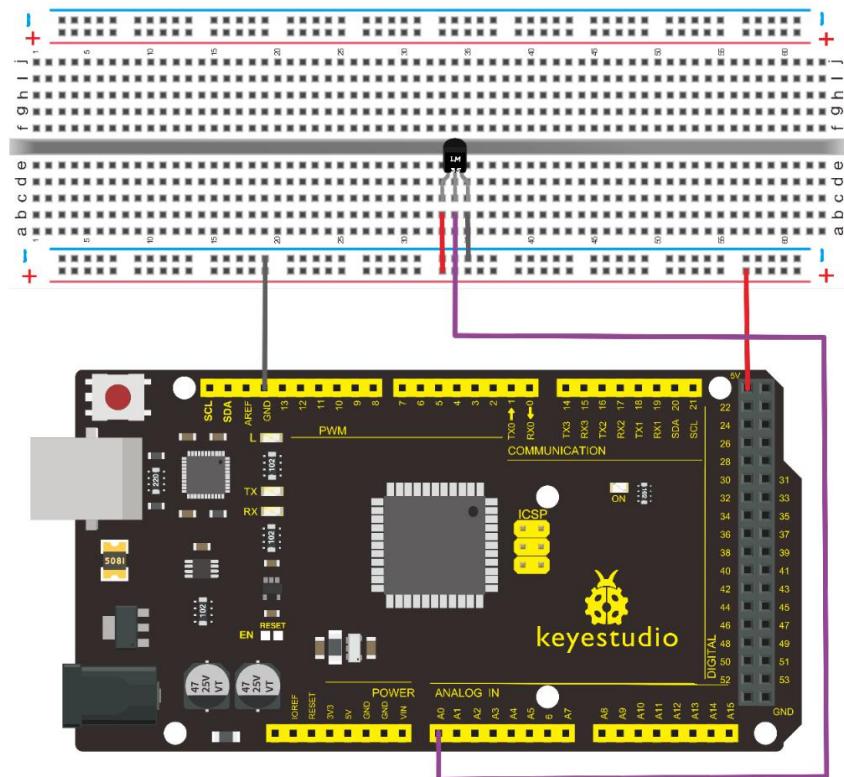
4. схема подключения



Подключение для V4.0



Подключение для 2560



5. образец кода

```
/*
```

```
супер обучающий комплект keyestudio
```

```
Project 12
```

```
LM35
```

```
http://www.keyestudio.com
```

```
*/
```

```
int potPin = 0; // инициализируем аналоговый вывод 0 для датчика температуры LM35
```

```
установка void ()
```

```
{
```

```
Serial.begin (9600); // устанавливаем скорость передачи "9600"}
```

```
пустой цикл ()
```

```
{
```

```
int val; // определение переменной int dat; //
```

```
определение переменной
```

```
val = analogRead (0); // считываем аналоговое значение датчика и присваиваем его
```

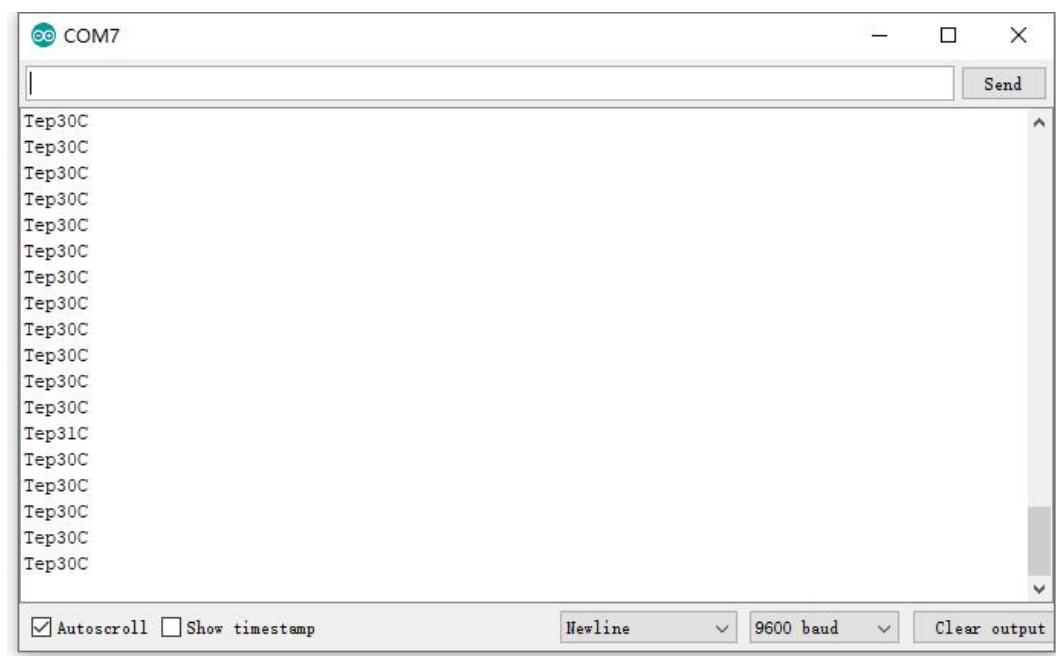
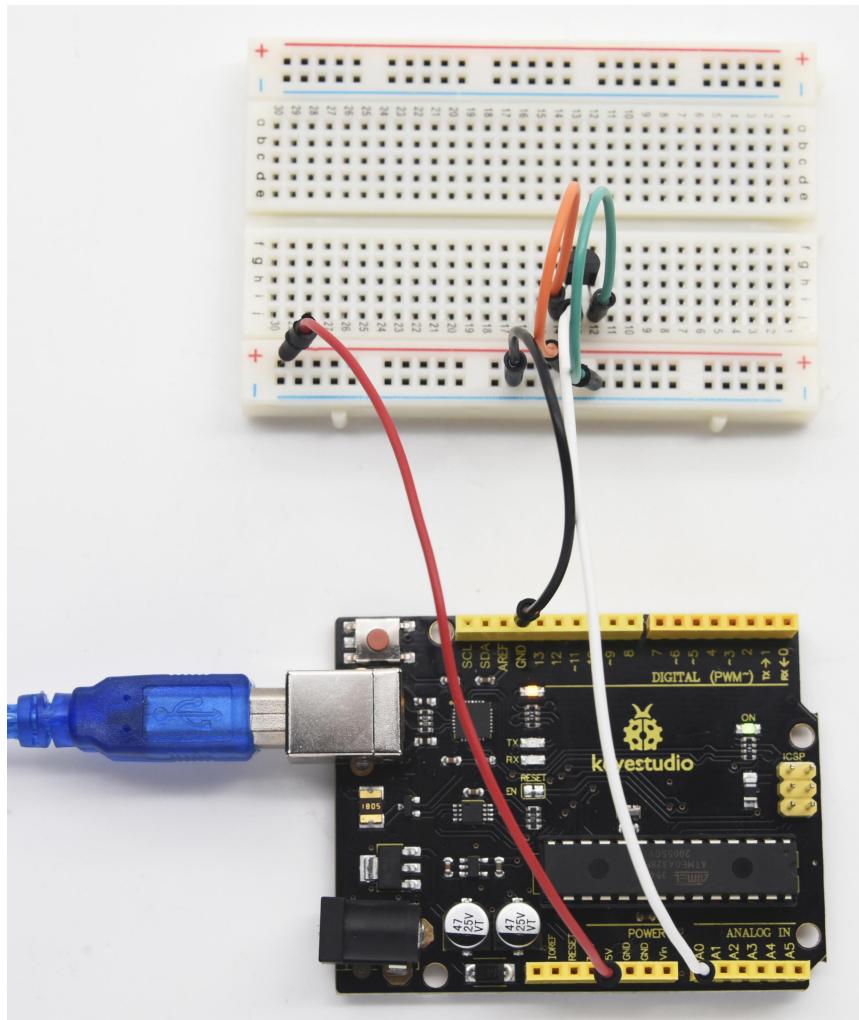
```
значению val
```

```
dat = (125 * val) >> 8; // формула расчета температуры
```

```
Serial.print ("Ter"); // выводим и отображаем символы,  
начинающиеся с Тер  
Serial.print (dat); // вывод и отображение значения dat Serial.println  
("C"); // отображение символов "C" delay (500); // ждем 0,5  
секунды  
}  
||||||||||||||||||||||||||||||||||||||||
```

6. результат теста

После загрузки программы вы можете открыть окно мониторинга, чтобы увидеть текущую температуру.



Проект 13: переключатель наклона



1. Введение

Это эксперимент с шаровым переключателем. Шариковый переключатель также называют стальным шаровым переключателем. Он управляет цепью, соединяя направляющий штифт с катящимся шариком. В этом проекте мы управляем светодиодной подсветкой, считывая состояние шарового переключателя.

2. принцип работы

Когда один конец переключателя находится ниже горизонтального положения, переключатель включен.

Напряжение аналогового порта составляет около 5 В (1023 в двоичном формате). Светодиод загорится.

Когда другой конец переключателя находится ниже горизонтального положения, переключатель выключен. Напряжение аналогового порта составляет около 0 В (0 в двоичном формате). Светодиод погаснет.

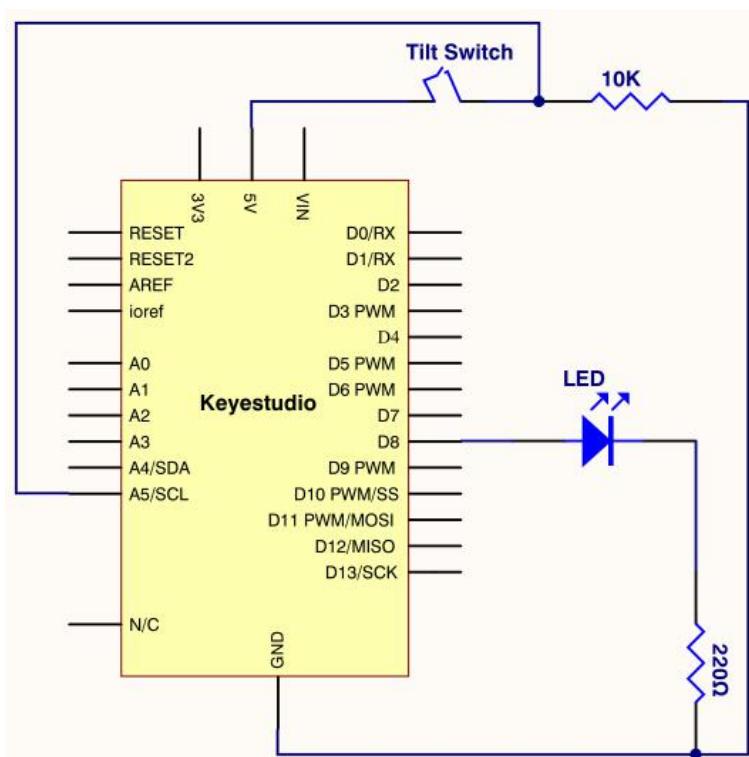
В программе определяем включен переключатель или выключен

в зависимости от значения напряжения аналогового порта, выше 2,5 В (512 в двоичном формате) или нет.

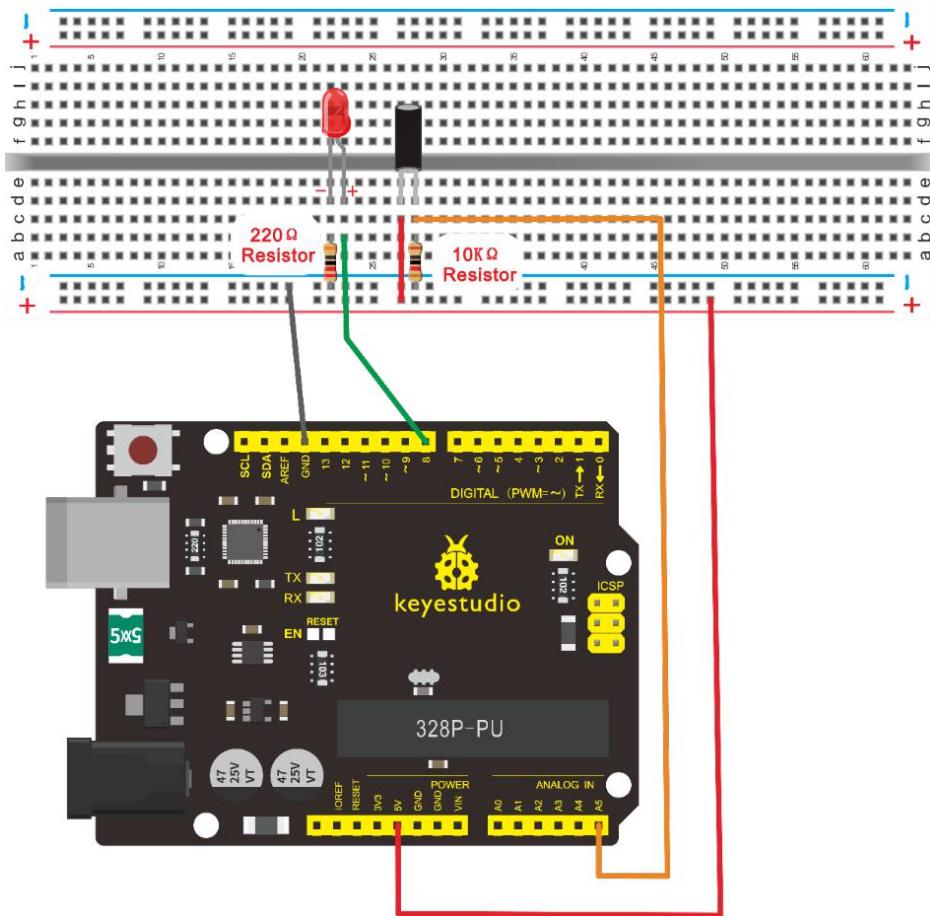
3. требуется оборудование

- Плата V4.0 или плата MEGA 2650 * 1 Шаровой
- переключатель * 1
- Светодиод * 1
- Резистор 220 Ом * 1
- Резистор 10 кОм * 1
- Перемычка макетной платы * 5
- USB-кабель * 1

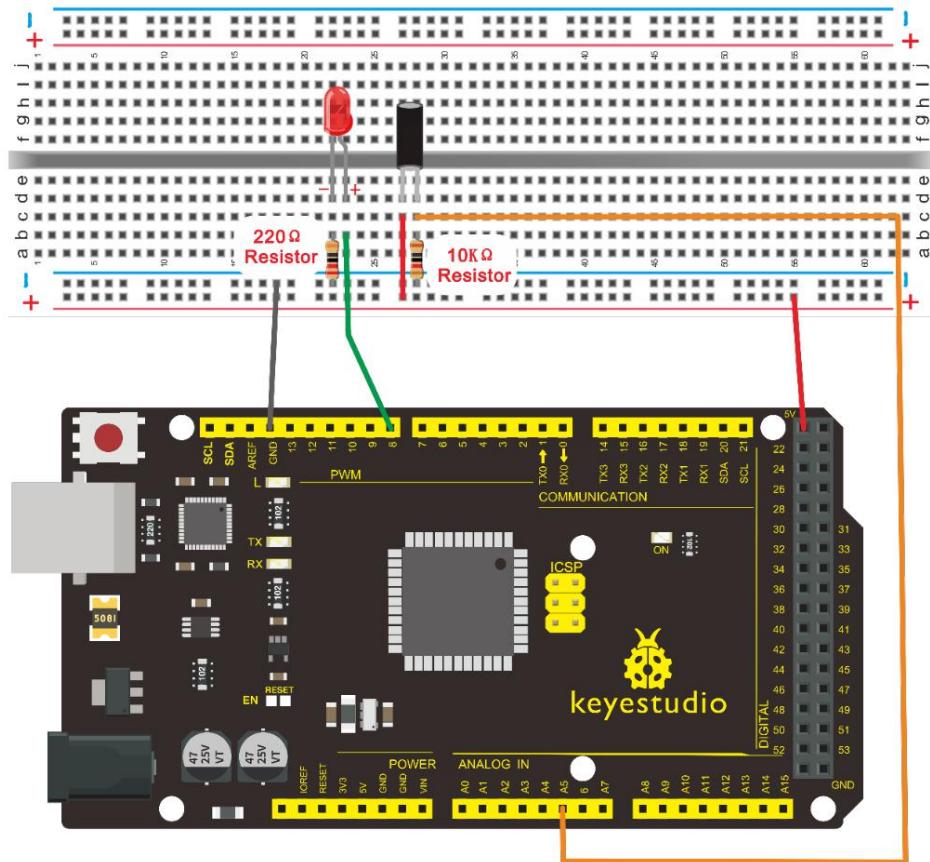
4. схема подключения



Подключение для V4.0



Подключение для 2560



5. образец кода

```
/*
```

супер обучающий комплект keyestudio

Project 13

Переключатель наклона

<http://www.keyestudio.com>

```
*/
```

установка void ()

```
{
```

```
pinMode (8, OUTPUT); // устанавливаем цифровой контакт 8 как «выход»
```

```
}

пустой цикл ()

{

int i; // определяем переменную i while

(1)

{

i = analogRead (5); // считываем значение напряжения аналогового вывода 5 if (i> 512) //

если больше 512 ( 2,5 В )

{

digitalWrite (8, LOW); // включаем светодиод

}

else // иначе

{

digitalWrite (8, HIGH); // выключаем светодиод}}}
```

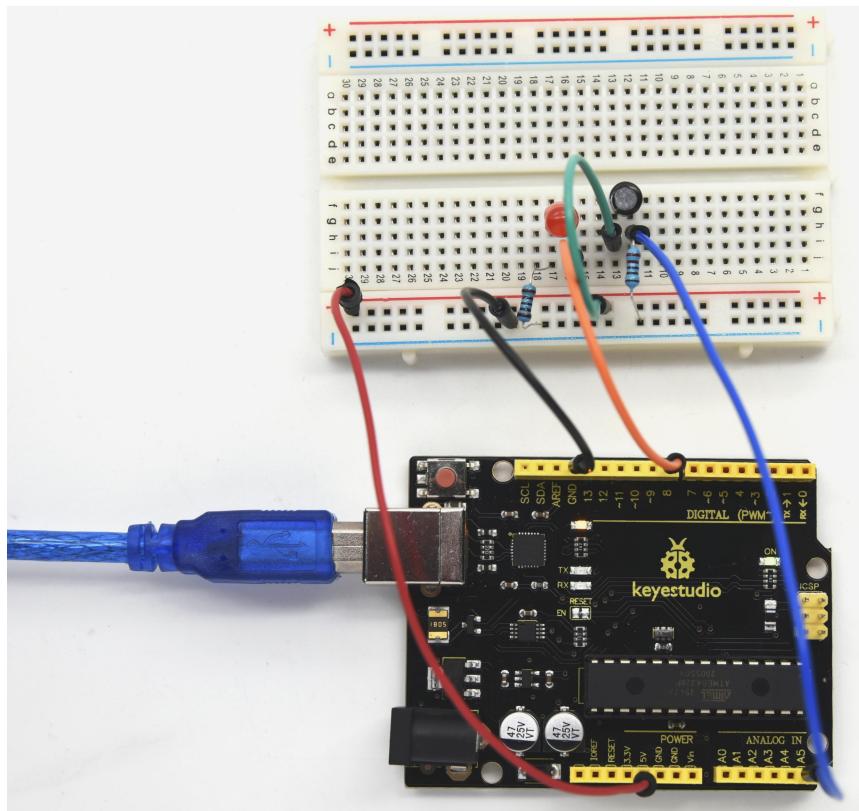
||||||||||||||||||||||||||||||||||||

6. результат теста

Держите макет рукой. Наклоните ее до некоторой степени, светодиод загорится.

Если наклона нет, светодиод не горит. Принцип этого эксперимента можно применить и к релейному управлению.

Эксперимент завершен. Спасибо!



Проект 14: ИК-пульт дистанционного управления



1. Введение

Что такое инфракрасный приемник?

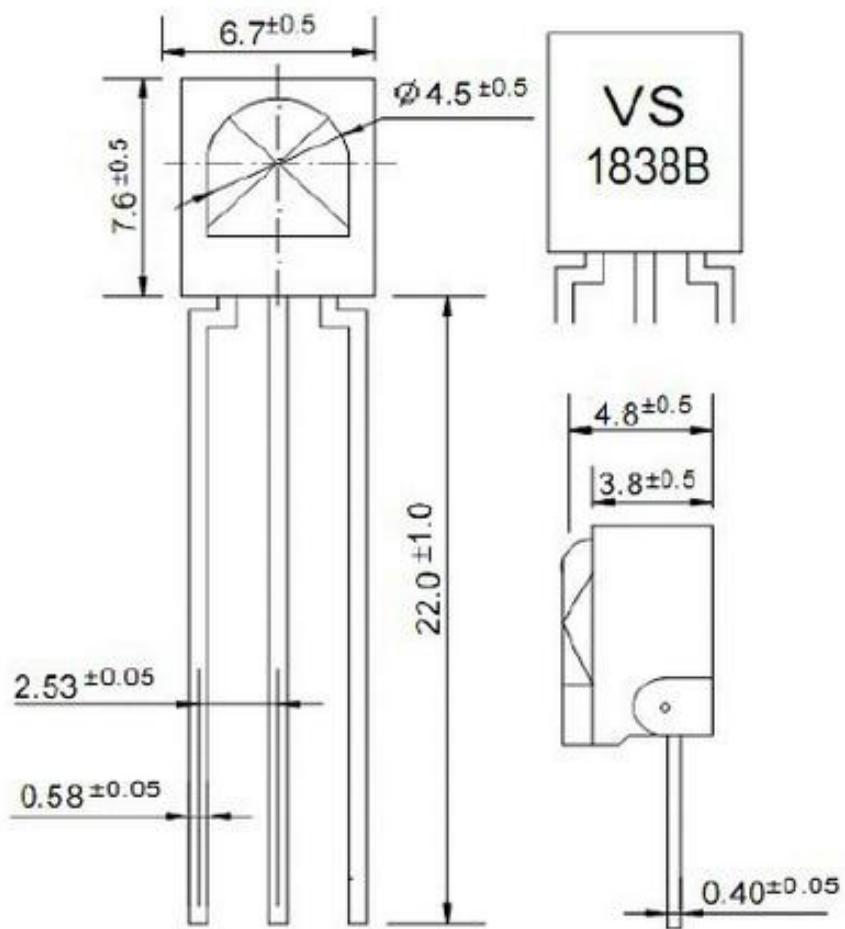
Сигнал от инфракрасного пульта дистанционного управления представляет собой последовательность двоичных импульсных кодов. Чтобы избежать других помех инфракрасного сигнала во время беспроводной передачи, сигнал предварительно модулируется на определенной несущей частоте, а затем отправляется с помощью диода инфракрасного излучения.

Приемное устройство инфракрасного излучения должно отфильтровывать другие волны и принимать сигналы на этой конкретной частоте и модулировать их обратно в двоичный импульсный код, известный как демодуляция.

2. принцип работы

Встроенный приемник преобразует световой сигнал, полученный от отправителя, в слабый электрический сигнал. Сигнал будет усилен усилителем IC. После автоматической регулировки усиления, полосовой фильтрации, демодуляции, формирования волны он возвращается к исходному коду. Затем код вводится в схему идентификации кода с помощью выходного вывода сигнала приемника.

3. контакт и проводка для инфракрасного приемника



Инфракрасный приемник имеет 3 контакта. Когда вы его используете, подключите VOUT к аналоговому выводу, GND к GND, VCC к + 5V.



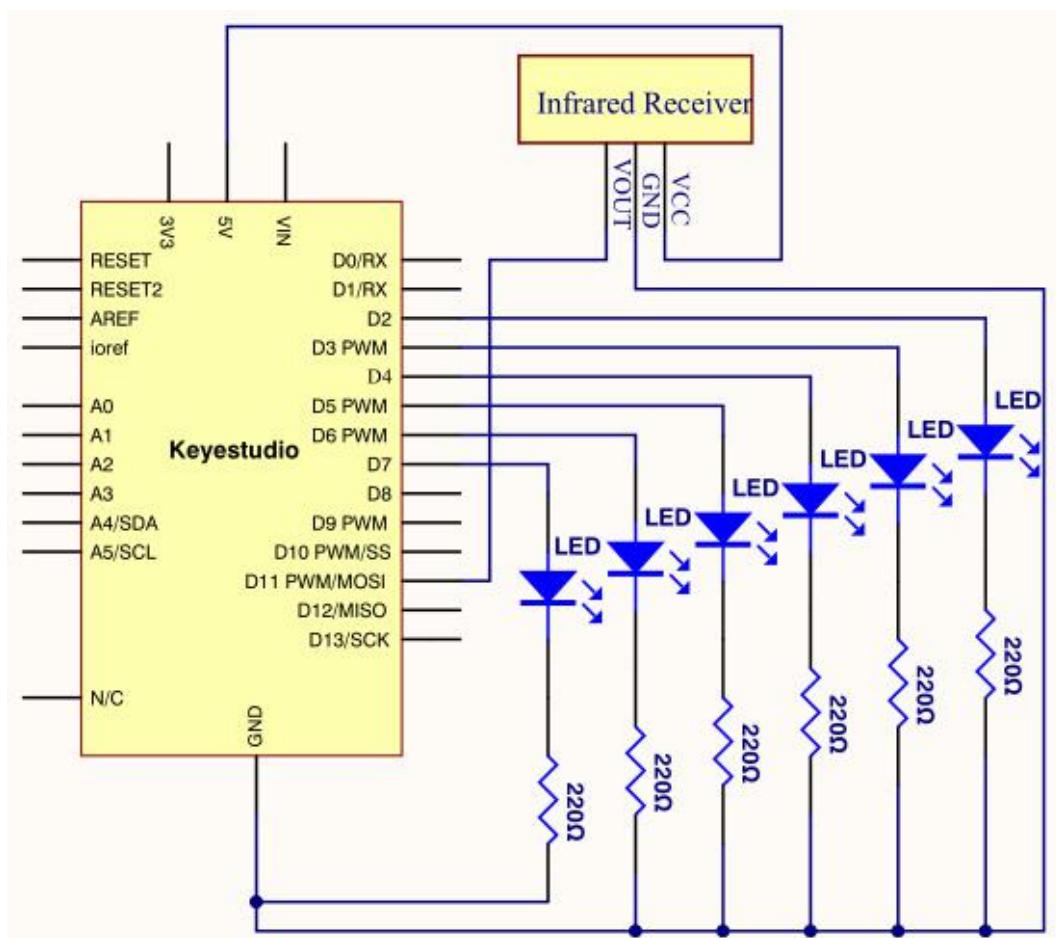
4. требуется оборудование

- Плата V4.0 или плата MEGA 2650 * 1

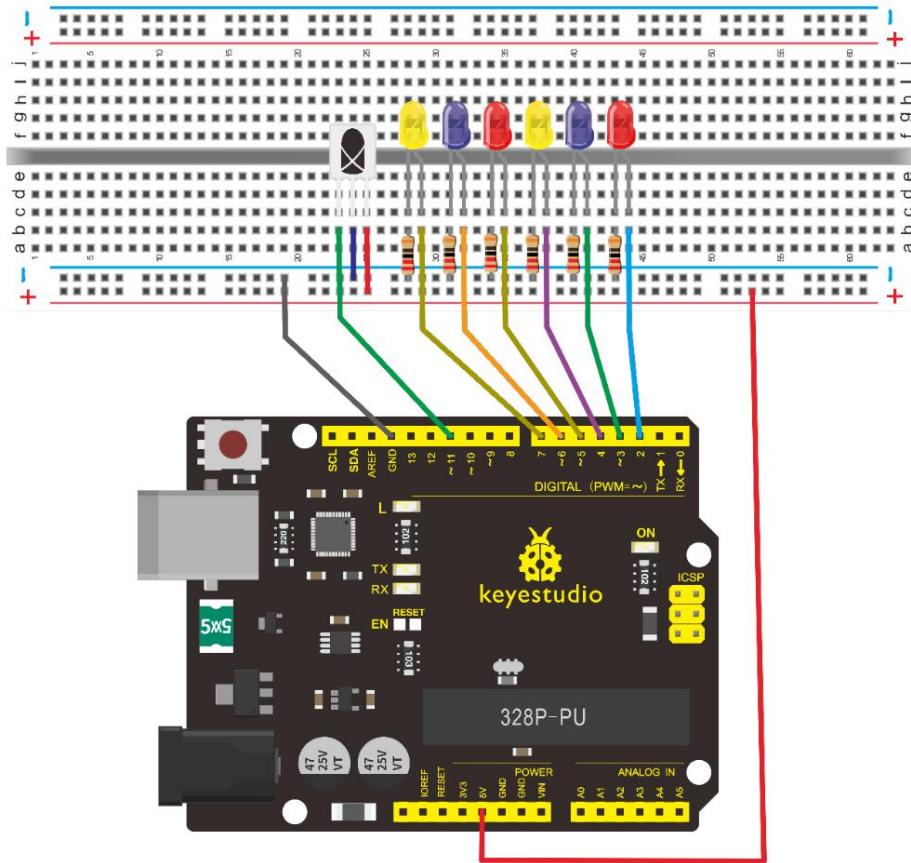
- Инфракрасный пульт дистанционного управления * 1
- Инфракрасный приемник * 1
- Светодиод * 6
- 220 Ом Резистор * 6
- Провод макетной платы * 11
- USB-кабель * 1

5. схема подключения

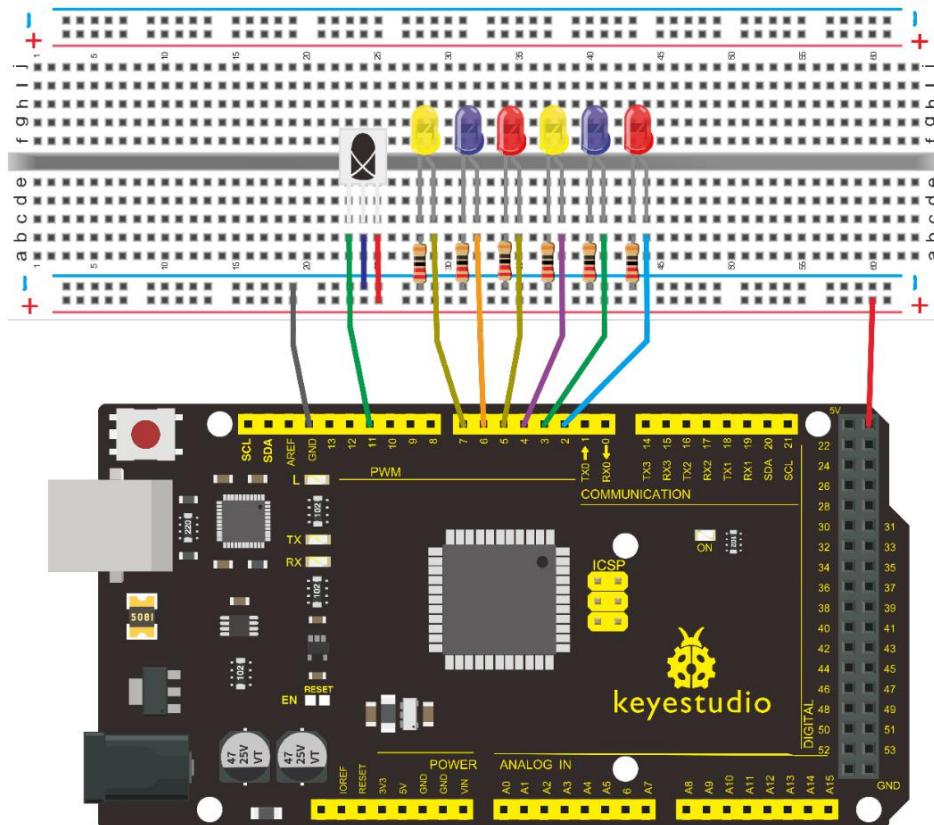
Сначала подключите плату контроллера; затем подключите инфракрасный приемник, как указано выше, подключите VOUT к цифровому контакту 11, соедините светодиоды с резисторами и подключите резисторы к контактам 2,3,4,5,6,7.



Подключение для V4.0



Подключение для 2560



6. Принцип эксперимента.

Если вы хотите расшифровать код с пульта дистанционного управления, вы должны сначала узнать, как он закодирован. Используемый здесь метод кодирования - это протокол NEC.

Ниже приводится краткое введение.

- Протокол NEC

7. особенности

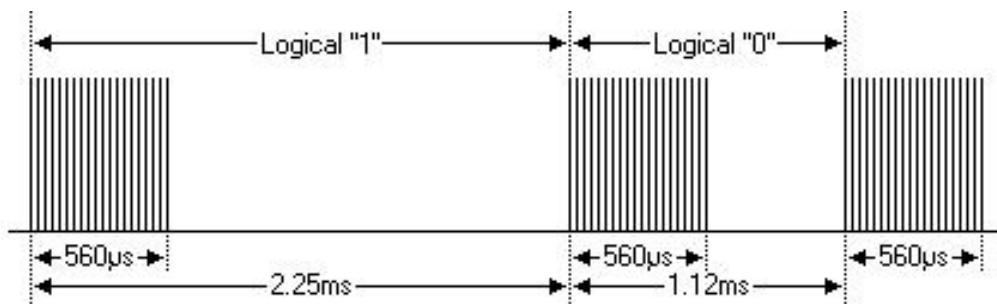
(1) 8-битный адрес и 8-битная длина команды

(2) адрес и команда передаются дважды для надежности (3) импульсная
дистанционная модуляция

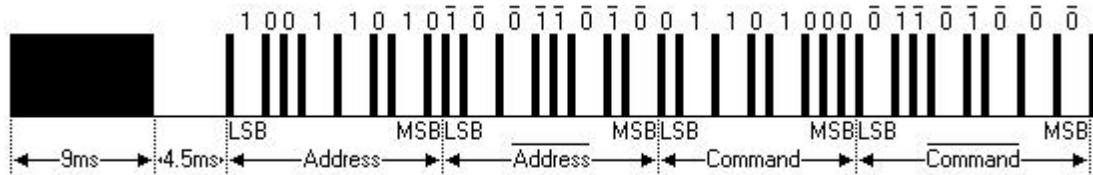
(4) несущая частота 38 кГц (5) время передачи
битов 1,125 мс или 2,25 мс

8. протокол, как показано ниже

- Логические 0 и 1 определены ниже.

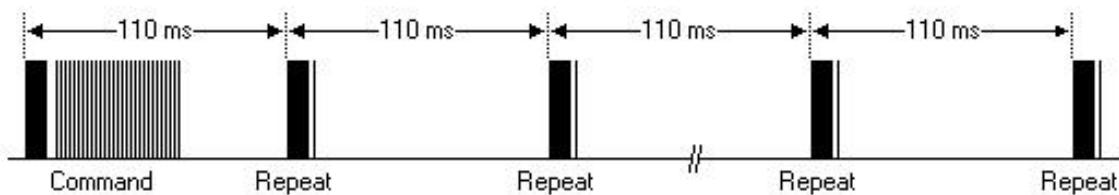


- Импульс передается при нажатии кнопки и сразу же
отпускается



На рисунке выше показана типичная последовательность импульсов протокола NEC. В этом протоколе сначала передается LSB. В этом случае передаются Адрес \$ 59 и Команда \$ 16. Сообщение начинается с пакета AGC длительностью 9 мс, который использовался для установки усиления более ранних ИК-приемников. После этого пакета AGC следует интервал 4,5 мс, за которым следуют адрес и команда. Адрес и команда передаются дважды. Во второй раз все биты инвертируются и могут использоваться для проверки полученного сообщения. Общее время передачи постоянно, потому что каждый бит повторяется с инвертированной длиной. Если вас не интересует такая надежность, вы можете игнорировать инвертированные значения или увеличить адрес и команду до 16 бит каждое!

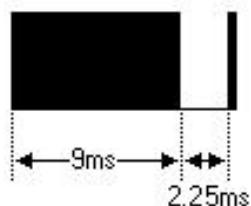
- Импульс передается при нажатии кнопки и отпускается через некоторое время.



Команда передается только один раз, даже если кнопка на пульте

элемент управления остается нажатым. Каждые 110 мс передается код повтора, пока клавиша остается нажатой. Этот код повторения представляет собой просто импульс АРУ длительностью 9 мс, за которым следует интервал 2,25 мс и пакет 560 мкс.

- Повторный импульс



Запись когда импульс попадает в интегрированный приемник, происходит декодирование, усиление сигнала и формирование волны. Поэтому вам нужно убедиться, что уровень выходного сигнала прямо противоположен уровню на стороне отправки сигнала.

То есть, когда нет инфракрасного сигнала, выходной конец находится на высоком уровне; когда есть инфракрасный сигнал, выходной конец находится на низком уровне. На осциллографе можно увидеть импульс на приемном конце. Постарайтесь лучше понять программу на основе увиденного.

9. образец кода

```
/*
супер обучающий комплект keyestudio
Project 14
Удаленный
http // www.keyestudio.com
*/
#include <IRremote.h>

int RECV_PIN = 11; int
LED1 = 2;
int LED2 = 3;
int LED3 = 4;
int LED4 = 5;
int LED5 = 6;
int LED6 = 7;
длинный on1 = 0x00FF6897; long
off1 = 0x00FF9867; длинный on2 =
0x00FFB04F; long off2 =
0x00FF30CF; длинный on3 =
0x00FF18E7; long off3 =
0x00FF7A85; длинный on4 =
0x00FF10EF;
```

```
long off4 = 0x00FF38C7; длинный  
on5 = 0x00FF5AA5; long off5 =  
0x00FF42BD; длинный оп6 =  
0x00FF4AB5; long off6 =  
0x00FF52AD; IRrecv irrecv  
(RECV_PIN);  
  
decode_results результаты;  
  
// Выгружает структуру decode_results. // Вызов после  
IRrecv :: decode ()  
  
// void * для решения проблемы компилятора // void  
  
dump (void * v) {  
  
// decode_results * results = (decode_results *) v void dump  
  
(decode_results * results) {  
  
int count = результаты-> rawlen;  
  
if (results-> decode_type == UNKNOWN) {  
  
Serial.println («Не удалось расшифровать сообщение»);  
}  
  
еще  
{  
  
if (results-> decode_type == NEC) {
```

```
    Serial.print ("Декодированный NEC:");

}

else if (results-> decode_type == SONY) {

    Serial.print ("Декодированный SONY:");

}

иначе if (results-> decode_type == RC5) {

    Serial.print ("Декодированный RC5:");

}

иначе if (results-> decode_type == RC6) {

    Serial.print ("Декодированный RC6:");

}

Serial.print (результаты-> значение, HEX);

Serial.print ("(");

Serial.print (результаты-> биты, DEC);

Serial.println ("биты)");

}

Serial.print ("Необработанный ());

Serial.print (количество, DEC);

Serial.print ("(":');
```

```
для (int i = 0; i < count; i ++)

{

    if ((i% 2) == 1) {

        Serial.print (результаты-> rawbuf [i] * USECPERTICK, DEC);

    }

еще

{

    Serial.print (- (int) results-> rawbuf [i] * USECPERTICK, DEC);

}

Serial.print ("");

}

Serial.println ("");

}

установка void ()

{

pinMode (RECV_PIN, INPUT);

pinMode (LED1, ВЫХОД);

pinMode (LED2, ВЫХОД);

pinMode (LED3, ВЫХОД);

pinMode (LED4, ВЫХОД);

pinMode (LED5, ВЫХОД);

pinMode (LED6, ВЫХОД);
```

```
pinMode (13, ВЫХОД);

Serial.begin (9600);

irrecv.enableIRIn (); // Запускаем приемник

}

int on = 0;

беззнаковый long last = millis (); пустой

цикл ()

{

если (irrecv.decode (& результаты))

{

// Если с момента последнего // приема ИК-сигнала прошло не менее 1/4

секунды, переключить реле

if (millis () - last > 250 {

on =! on;

// digitalWrite (8, на? ВЫСОКИЙ: НИЗКИЙ); digitalWrite

(13, на? ВЫСОКИЙ: НИЗКИЙ); дамп (& результаты);

}

если (results.value == on1)

digitalWrite (LED1, HIGH);

если (results.value == off1)
```

```
digitalWrite (LED1, LOW);

если (results.value == on2)

    digitalWrite (LED2, ВЫСОКИЙ);

если (results.value == off2)

    digitalWrite (LED2, LOW);

если (results.value == on3)

    digitalWrite (LED3, HIGH);

если (results.value == off3)

    digitalWrite (LED3, LOW);

если (results.value == on4)

    digitalWrite (LED4, HIGH);

если (results.value == off4)

    digitalWrite (LED4, LOW);

если (results.value == on5)

    digitalWrite (LED5, HIGH);

если (results.value == off5)

    digitalWrite (LED5, LOW);

если (results.value == on6)

    digitalWrite (LED6, HIGH);

если (results.value == off6)

    digitalWrite (LED6, LOW);

последний = миллис ();
```

```
irrecv.resume(); // Получение следующего значения}
```

```
}
```

```
||||||||||||||||||||||||||||||||||||||
```

Запись добавьте папку **IRremote** в директорию установки \ Arduino \ compiler

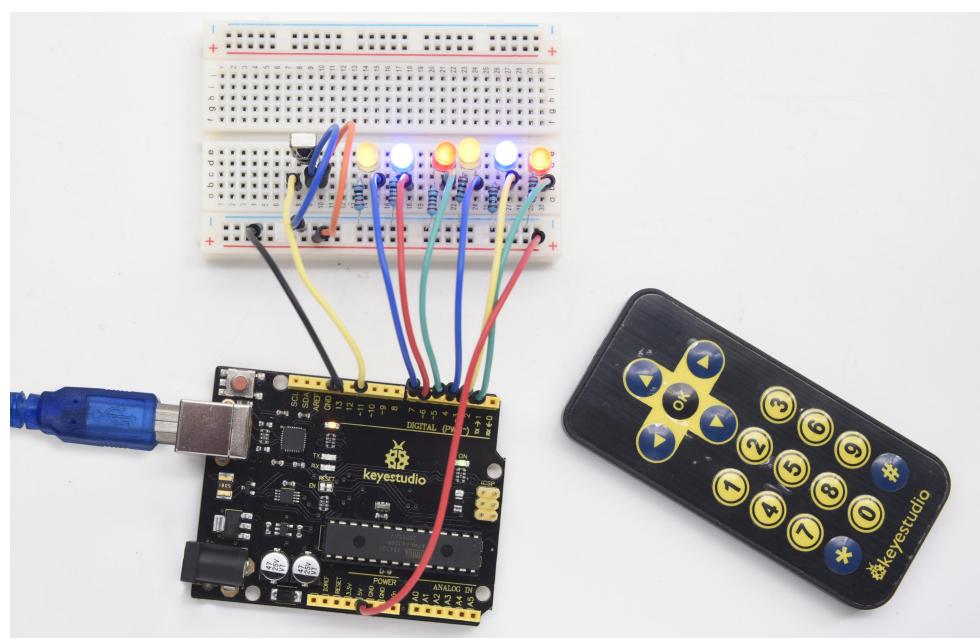
libraries, иначе вы не сможете ее скомпилировать.

Инфракрасная удаленная библиотека

<https://github.com/shirriff/Arduino-IRremote>

10. результат теста

Декодируйте кодированный импульсный сигнал, излучаемый пультом дистанционного управления, а затем выполните соответствующее действие в соответствии с результатами декодирования. Таким образом, вы сможете управлять своим устройством с помощью пульта дистанционного управления.



После загрузки откройте монитор последовательного порта, вы увидите результат, как показано ниже.

```
Decoded NEC: FF6897 (32 bits)
Raw (68): -9614 9050 -4550 550 -550 550 -600 550 -600 550 -550 550 -600 550 -550 600 -550 550
Decoded NEC: FF9867 (32 bits)
Raw (68): -18792 9000 -4550 550 -600 550 -550 550 -600 550 -550 550 -600 550 -550 550 -600 550
Decoded NEC: FFB04F (32 bits)
Raw (68): -30414 9050 -4500 550 -600 550 -550 550 -600 550 -600 550 -550 550 -600 550 -600 550
Decoded NEC: FF30CF (32 bits)
Raw (68): 22076 9000 -4550 550 -600 550 -550 550 -600 550 -600 550 -550 550 -600 550 -550 550
Decoded NEC: FF18E7 (32 bits)
Raw (68): 26764 9050 -4500 550 -600 550 -550 600 -550 600 -500 600 -550 600 -550 550 -550 600
Decoded NEC: FF7A85 (32 bits)
Raw (68): -28020 9100 -4450 650 -500 650 -500 600 -500 650 -500 600 -550 600 -500 600 -550 600
Decoded NEC: FF10EF (32 bits)
Raw (68): 11350 9100 -4500 600 -500 600 -550 600 -500 600 -550 600 -550 600 -500 600 -550 550
Decoded NEC: FF38C7 (32 bits)
Raw (68): 6750 9050 -4500 550 -600 550 -550 550 -600 550 -600 550 -550 550 -600 550 -600 550 -
Decoded NEC: FF5AA5 (32 bits)
Raw (68): -4672 9000 -4550 550 -550 550 -600 550 -600 550 -550 550 -600 550 -600 550 -550 550
Decoded NEC: FF42BD (32 bits)
Raw (68): -9092 9100 -4450 650 -500 650 -500 600 -500 600 -550 600 -550 600 -500 600 -550 600
Decoded NEC: FF4AB5 (32 bits)
Raw (68): -12136 9100 -4500 600 -500 650 -500 650 -450 650 -500 650 -500 600 -500 650 -500 650
Decoded NEC: FF52AD (32 bits)
Raw (68): 25480 9050 -4500 550 -600 550 -550 550 -600 550 -600 550 -550 550 -600 550 -600 550
< >
 Autoscroll  Show timestamp   
```

Проект 15: Считывание аналогового значения



1. Введение

В этом эксперименте мы начнем изучение аналоговых интерфейсов ввода-вывода.

На Arduino есть 6 аналоговых интерфейсов, пронумерованных от A0 до A5. Далее приступим к нашему проекту. Используемый здесь потенциометр представляет собой типичный выходной компонент аналогового значения, который нам знаком.

2. Требуется оборудование

- Плата V4.0 или плата MEGA 2650 * 1
- Потенциометр * 1
- Макетная плата * 1
- Перемычка макетной платы * 3
- USB-кабель * 1

3. характеристики потенциометра

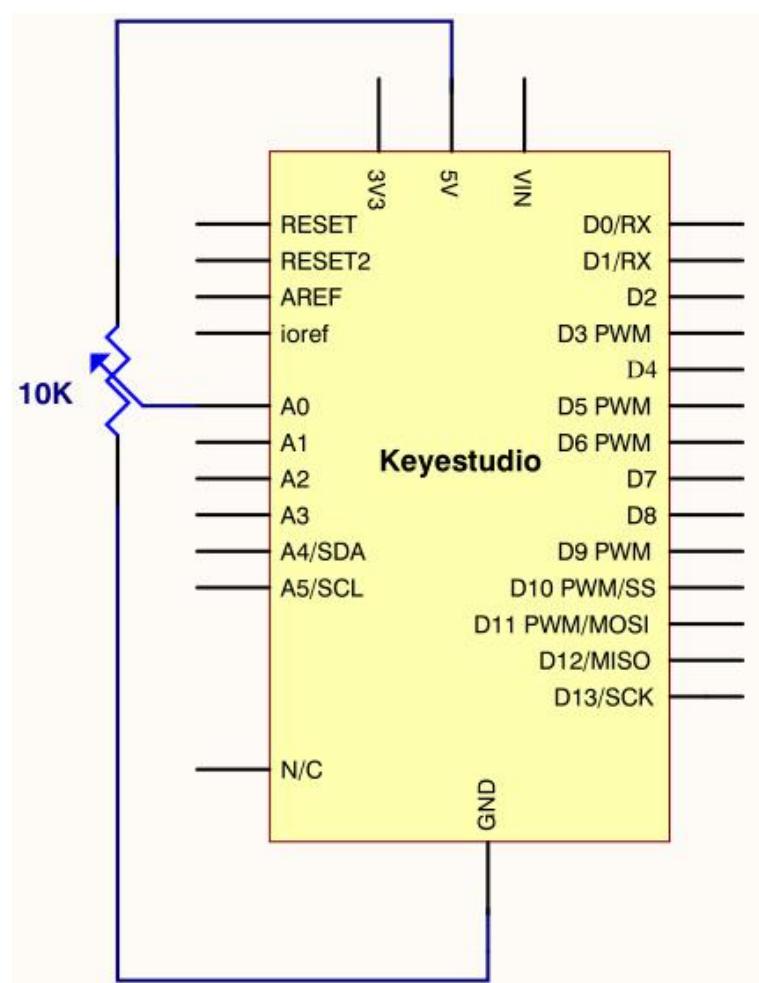
Регулируемый потенциометр - это просто своего рода резистор. При вращении потенциометра изменяется сопротивление, напряжение, скорость, яркость и температура. Это аналоговый электронный компонент, который имеет два состояния: 0 и 1 (высокий уровень и низкий уровень). Аналоговое количество другое. Его данные

Состояние представляет собой линейное состояние, например от 1 до 1000.

4. схема подключения

В этом эксперименте мы преобразуем значение сопротивления потенциометра в аналоговое и отобразим его на экране.

Это приложение, которое вам нужно хорошо освоить для наших будущих экспериментов.



Подключение для V4.0