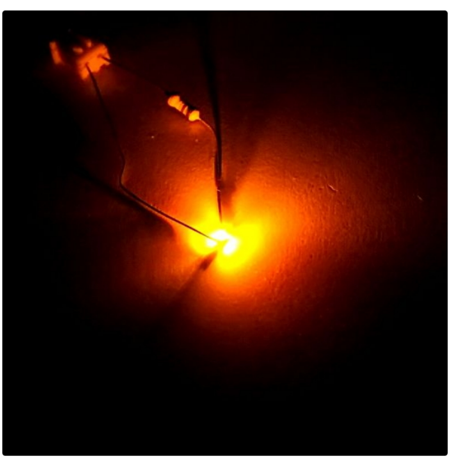
FLICKER UP TO 6 LEDS WITH ARDUINO

By gotcha99 in Technology > Arduino 21,507 41 8

Published Nov. 30, 2010

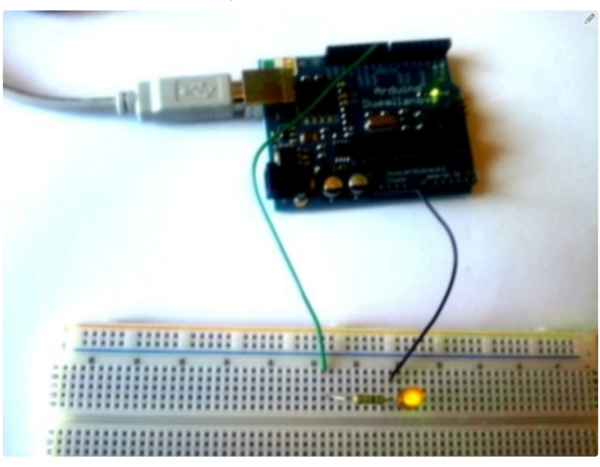


By gotcha99
Follow

Hi there.
Christmas is coming closer, and I was wondering if I could put some candles in my room to get in the "christmas mood".
But after a friends house burned down of christmas tree candles, I don't want to burn real candles into my room.
So why don't let the Arduino light for you?!

This Instructable will show you, how to flicker up to 6 LEDs using an Arduino Microcontroller.
For more information about Arduino visit http://www.arduino.cc/

Step 1: What You Need...



Here are the things you will need:

-Arduino Microcontroller (or similar)
-USB cable
-LEDs (up to 6 LEDs are possible)
-Resistors (equal to the number of used LEDs)
LED resistor calculator

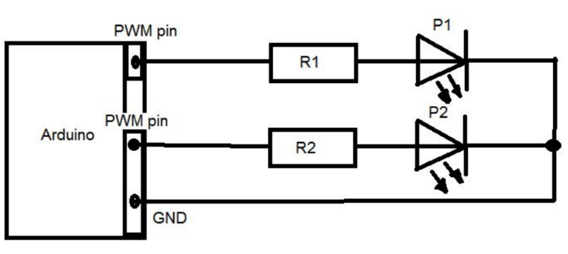
if you don't want to solder the circuit:
-Breadboad
-Jumper wires

Optional:
-DC power source

Downloads:
-Arduino IDE
Get it here!

-TrueRandom Library
Get it here! (Instructions included)
Unzip it to your "Libraries" folder.

Step 2: Wire It Up!

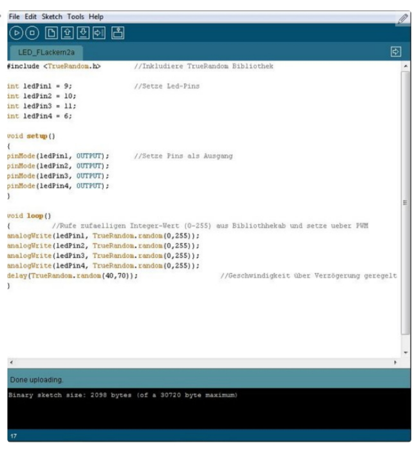


Connect the **resistor to the PWM pins** (e.g. 6, 9...) on the Arduino board.
Now connect the **second "leg" of the resistor to the anode of the LED (long "leg")**
Then connect the **cathode (short "leg") of the LED to ground (GND)** on the Arduino board.

Repeat this step for each LED you want to use.

When the wiring is done, continue to the next step!

Step 3: Code It!



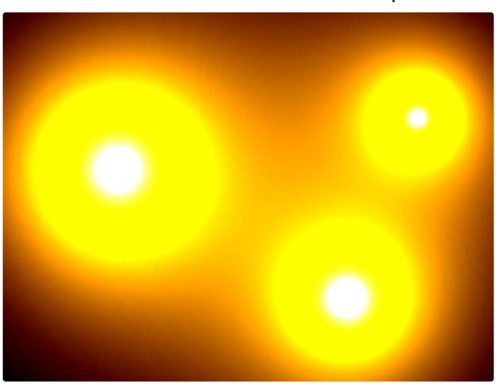
```
#include <TrueRandom.h>        //Inkludiere TrueRandom Bibliothek

int ledPin1 = 9;               //Setze Led-Pins
int ledPin2 = 10;
int ledPin3 = 11;
int ledPin4 = 6;

void setup()
{
pinMode(ledPin1, OUTPUT);      //Setze Pins als Ausgang
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(ledPin4, OUTPUT);
}

void loop()
{          //Rufe zufaelligen Integer-Wert (0-255) aus Bibliothhekab und setze ueber PWM
analogWrite(ledPin1, TrueRandom.random(0,255));
analogWrite(ledPin2, TrueRandom.random(0,255));
analogWrite(ledPin3, TrueRandom.random(0,255));
analogWrite(ledPin4, TrueRandom.random(0,255));
delay(TrueRandom.random(40,70));           //Geschwindigkeit über Verzögerung geregelt
}
```

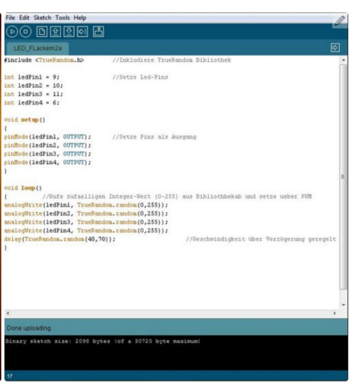
```
Done uploading.
Binary sketch size: 2098 bytes (of a 30720 byte maximum)
17
```

Copy this code to the Arduino IDE:

```
//Multiple LED flicker by gotcha (2010)
//
//
#include  <TrueRandom.h>  //Include the TrueRandom Library

int ledPin1 = 9; //Set the number of LEDs you use.
int ledPin2 = 10;
int ledPin3 = 11;
int ledPin4 = 6;
//You can use up to 6 LED pins

void setup()
{
pinMode(ledPin1, OUTPUT); //Sets the LED pins as output pins.
pinMode(ledPin2, OUTPUT); //Use the same pins used above
pinMode(ledPin3, OUTPUT);
pinMode(ledPin4, OUTPUT);
}

void loop()
{ //Sets a random value of LED voltage in the range of 0V - 5V with PWM
analogWrite(ledPin1, TrueRandom.random(0,255)); //Use the same pins used above
analogWrite(ledPin2, TrueRandom.random(0,255));
analogWrite(ledPin3, TrueRandom.random(0,255));
analogWrite(ledPin4, TrueRandom.random(0,255));
delay(TrueRandom.random(40,70)); //Limits the speed.
}
```

Step 4: Test It!





```
#include <TrueRandom.h>        //Inkludiere TrueRandom Bibliothek

int ledPin1 = 9;               //Setze Led-Pins
int ledPin2 = 10;
int ledPin3 = 11;
int ledPin4 = 6;

void setup()
{
pinMode(ledPin1, OUTPUT);      //Setze Pins als Ausgang
pinMode(ledPin2, OUTPUT);
pinMode(ledPin3, OUTPUT);
pinMode(ledPin4, OUTPUT);
}

void loop()
{          //Rufe zufaelligen Integer-Wert (0-255) aus Bibliothhekab und setze ueber PWM
analogWrite(ledPin1, TrueRandom.random(0,255));
analogWrite(ledPin2, TrueRandom.random(0,255));
analogWrite(ledPin3, TrueRandom.random(0,255));
analogWrite(ledPin4, TrueRandom.random(0,255));
delay(TrueRandom.random(40,70));           //Geschwindigkeit über Verzögerung geregelt
}
```

```
Done uploading.
Binary sketch size: 2098 bytes (of a 30720 byte maximum)
17
```

Now **connect the Arduino to the PC (or MAC) with the USB cable.**
Left click on the compile button at the top of the coding field.
Now **wait for the compiling to be done**.

Now **click on "Upload" and wait for the upload.**

A few moments later, the Arduino should start flickering the LEDs!

I hope you liked my Instructable.
Don't mind grammar and speech, I'm from Germany :)
Merry Christmas!

Add Tip Ask Question Post Comment

We have a **be nice** policy.
Please be positive and constructive.

Add Images Post

8 Comments

goldenshuttle 3 years ago Reply

The question is how to add a potentiometer to Arduino, making the 6 leds dimmable. That they flicker at high or low intensity, just like when you dim an old oil lantern and leave it all night flicker at very low intensity ?? it is easy to hook a 5K pot to arduino input, but can someone suggest how to modify the sketch please ?

Build_it_Bob 5 years ago Reply

I like very much ! Thank you and Happy Holidays !
Build_it_Bob

hds0405 5 years ago Reply

Great project, simple to implement and just perfect for the application I needed (inside a carved Halloween pumpkin display for my son). Thank you!!!

phatguppy 6 years ago Reply

This was a great starter tutorial and introduction to Arduino. I'd like to slow down the randomness in the blinking, to make it look more like fire flies. Any suggestions on how to tweak the TrueRandom library to do this?

1 reply

ledfreak01 6 years ago Reply

this i awesome keep it up

gotcha99 (author) 7 years ago Reply

Hey,

thanks for the infromtation,but somehow i can't add the needed lines in the instructable.

For those wanting to use the code: put " "
after "#include"

CODIY 7 years ago Reply

Hi Gotcha,

Great little script. I am planning to use it to drive lights on a wreath. I noticed that you left out the library name in your include statement.

For those looking to duplicate this code, note that you need to download and install the TrueRandom library into your Arduino folders, then complete the include statement in the script as follows:

#include

the library can be found here:
http://code.google.com/p/tinkerit/wiki/TrueRandom

Post Comment

Introduction

TrueRandom generates true random numbers on Arduino. They are different every time you start your program, and are truly unpredictable unlike the default Arduino random() function.

Compatibility

TrueRandom currently functions on the Arduino Diecimila, Duemilanove, 168 and 328 based Arduinos. It does not yet function on the Arduino Mega. TrueRandom uses Analog 0. Do not connect anything to this pin. These restrictions may be removed in future versions of this library.

Download

Download TrueRandom library. Extract the zip file, and copy the directory to your Arduino libraries folder.

What happens when you use the Arduino random() function?

The Arduino default random() function generates what appear to be random numbers. They are actually calculated from a formula. On reset, the formula is reset at a start point, then progresses through a long sequence of random looking numbers. However, Arduino starts at the same point in the sequence every reset. You can move to a different part of the sequence using srandom(), but how do you get a random start point from in the first place?

What happens when you use TrueRandom.random() function?

You get a random number. Really random. Different every time you restart.

Example time

```

## include

void setup() { Serial.begin(9600);

Serial.print("I threw a random die and got "); Serial.print(random(1,7));

Serial.print(". Then I threw a TrueRandom die and got "); Serial.println(TrueRandom.random(1,7));

}

void loop() { ; // Do nothing } ```

Upload that code to an Arduino Duemilanove and watch it on the Serial Monitor at 9600 baud. Hit the reset button, and see what it does. The random() function returns the same value every time, but the TrueRandom version is always different.

## TrueRandom basic functions

The existing random functions of Arduino are replicated in TrueRandom.

### TrueRandom.random()

Like the Arduino library and ANSI C, this generates a random number between 0 and the highest signed long integer 2,147,483,647.

### TrueRandom.random(n)

This generates a random number between 0 and (n-1). So random(6) will generate numbers between 0 and 5.

### TrueRandom.random(a,b)

This generates a random number between a and (b-1). So random(1,7) will generate numbers between 1 and 6.

## TrueRandom advanced functions

### TrueRandom.randomBit()

Generating true random numbers takes time, so it can be useful to only generate as many random bits as you need. randomBit() generates a 0 or a 1 with 50% probability. This is the core function from which the other TrueRandom libraries are built.

### TrueRandom.randomByte()

Generates a random byte between 0 and 255. Equivalent to random(256).

### TrueRandom.rand()

Like the ANSI C rand() command, this generates a random number between 0 and the highest signed integer 32767.

### TrueRandom.memfill(address, length)

Fills a block of bytes with random numbers. (length) bytes are filled in total, starting at the given (address).

## TrueRandom specialist functions

### TrueRandom.mac(address)

When operating devices on an Ethernet network, each device must have a unique MAC address. Officially, MAC addresses should be assigned formally via the IEEE Registration Authority. However, for practical purposes, MAC addresses can be randomly assigned without problems. This function writes a 6 byte MAC address to a given address. Randomly generated MAC addresses are great for projects or workshops involving large numbers of Arduino Ethernet shields, as each shield has a different MAC address, even though they are running identical code. See the MacAddress example which shows this in use.

### TrueRandom.uuid(address)

UUIDs are unique identifiers. They are 16 bytes (128 bits) long, which means that generating them randomly This generates a random UUID, and writes it to an array. UUIDs are globally unique numbers that are often used in web services and production electronics. TrueRandom can produce any one of 5,316,911,983,139,663,491,615,228,241,121,378,304 different numbers. You're more likely to win top prize in the national lottery 3 times in a row than get two matching UUIDs.

## How TrueRandom works

It is hard to get a truly random number from Arduino. TrueRandom does it by setting up a noisy voltage on Analog pin 0, measuring it, and then discarding all but the least significant bit of the measured value. However, that isn't noisy enough, so a von Neumann whitening algorithm gathers enough entropy from multiple readings to ensure a fair distribution of 1s and 0s.

The other functions within TrueRandom construct the requested values by gathering just enough random bits to produce the required numbers. Generating a random bit takes time, so a significant part of the code works to ensure the random bits are used as efficiently as possible.

## Projects using TrueRandom

Generative Music from Gijs
```

True Random.zip

Arduino FlickerCode

Attachments

Sheet: 1/1

```
//Copy this code to the Arduino IDE:

//Multiple LED flicker by gotcha (2010)
//
//
#include   <TrueRandom.h>   //Include the TrueRandom Library

int ledPin1 = 9; //Set the number of LEDs you use.
int ledPin2 = 10;
int ledPin3 = 11;
int ledPin4 = 6;
//You can use up to 6 LED pins

void setup()
{
pinMode(ledPin1, OUTPUT); //Sets the LED pins as output pins.
pinMode(ledPin2, OUTPUT); //Use the same pins used above
pinMode(ledPin3, OUTPUT);
pinMode(ledPin4, OUTPUT);
}

void loop()
{ //Sets a random value of LED voltage in the range of 0V - 5V with PWM
analogWrite(ledPin1, TrueRandom.random(0,255)); //Use the same pins used above
analogWrite(ledPin2, TrueRandom.random(0,255));
analogWrite(ledPin3, TrueRandom.random(0,255));
analogWrite(ledPin4, TrueRandom.random(0,255));
delay(TrueRandom.random(40,70)); //Limits the speed.
}
```