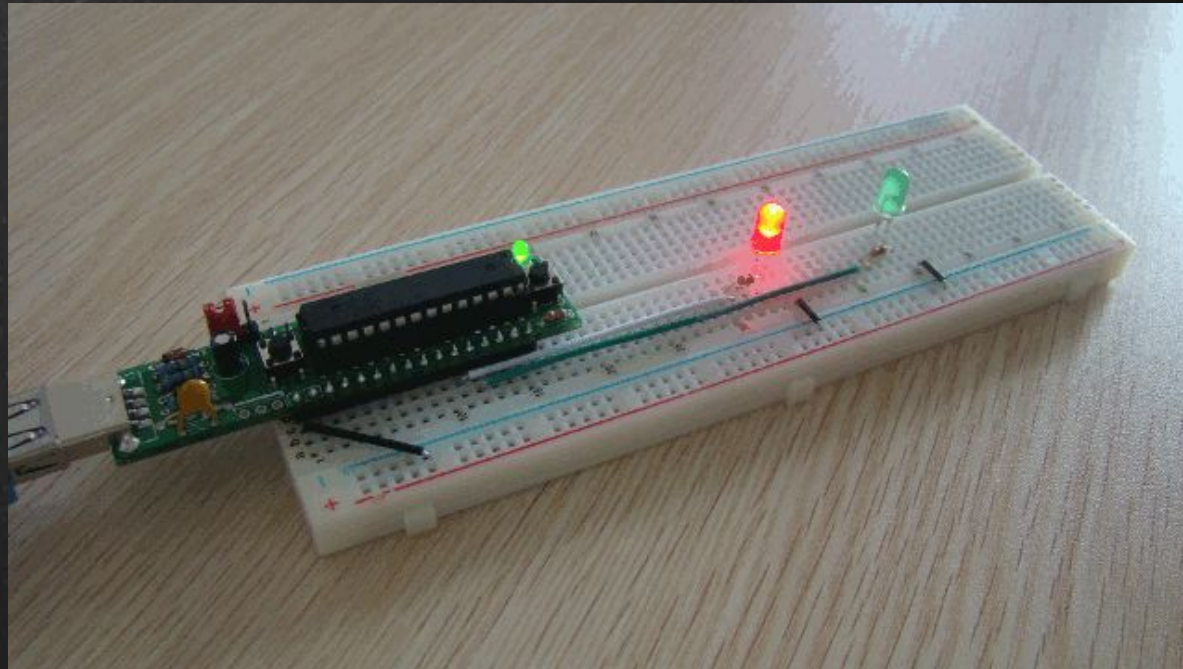


# IDD001 Lecture 2: Let there be blink



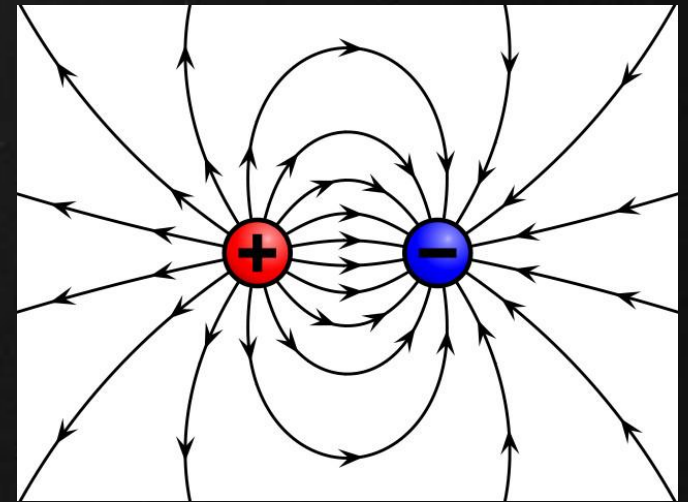
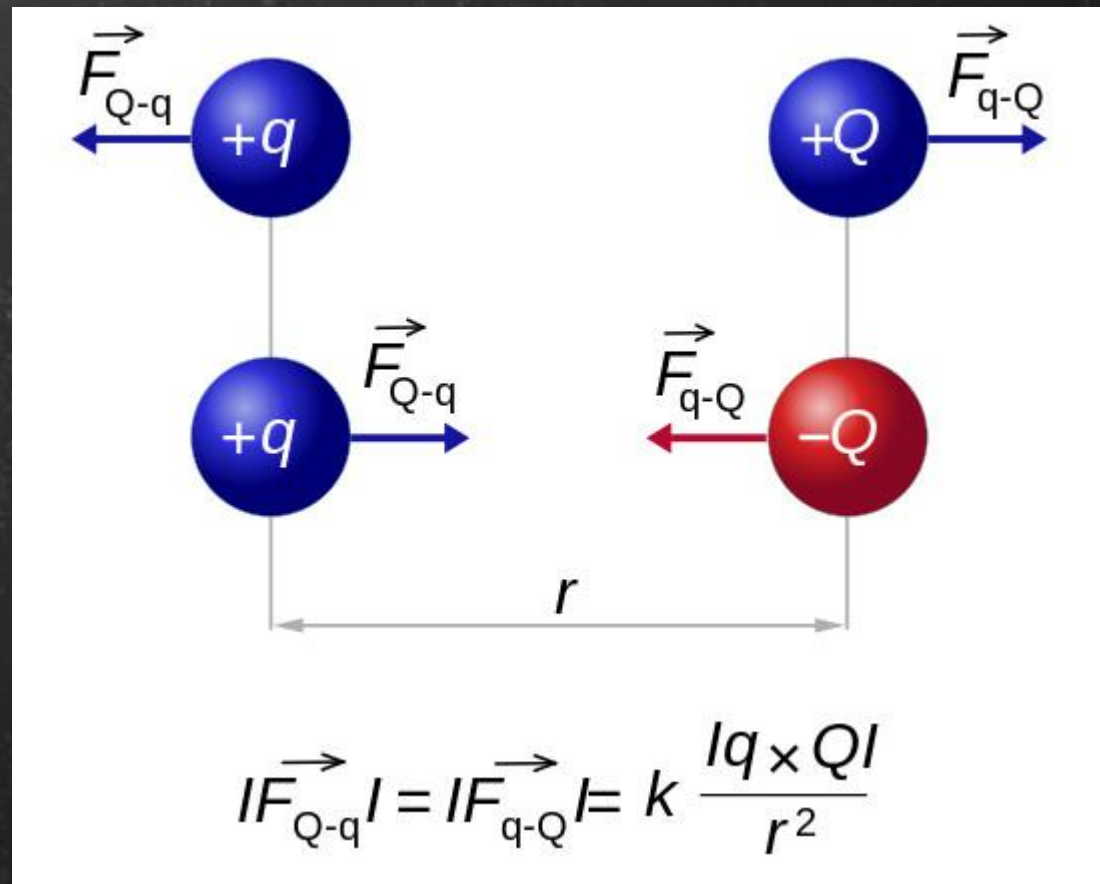
Atommann <[atommann@gmail.com](mailto:atommann@gmail.com)>  
2017 Fall  
I call this course “Play with Arduinos”

All men<sup>[1]</sup> by nature desire to know. -- Aristotle

在技术书籍的阅读中，我偏爱爱因斯坦阐释的方法——“在所阅读的书中，找出可以把自己引向深入的东西，把其他的一切统统抛掉。”这就是抛掉使大脑负担过重和把自己诱离要点的一切。-- Someone

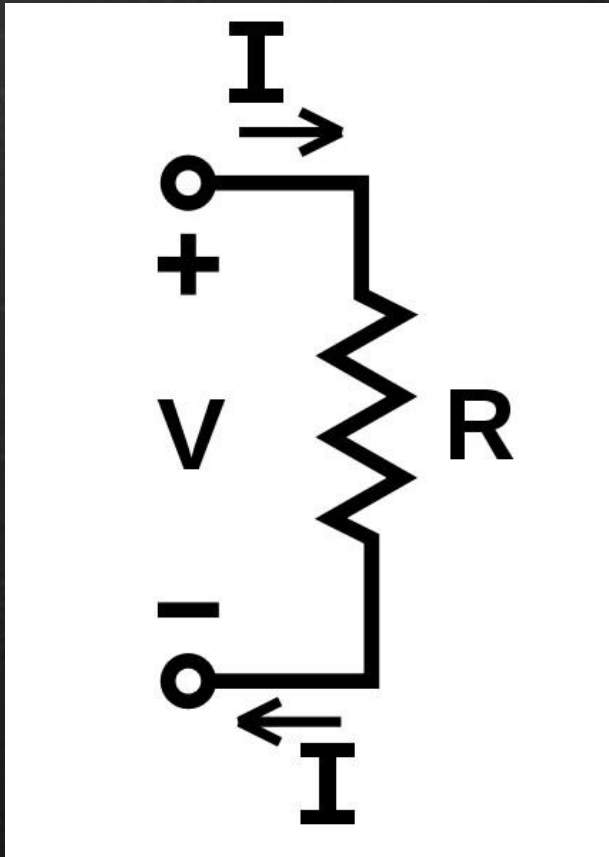
[1] 所有人，不分性别。

# 异种电荷相互吸引



# Ohm's Law

When you read you begin with ABC, when you elec you begin with Ohm's Law :)



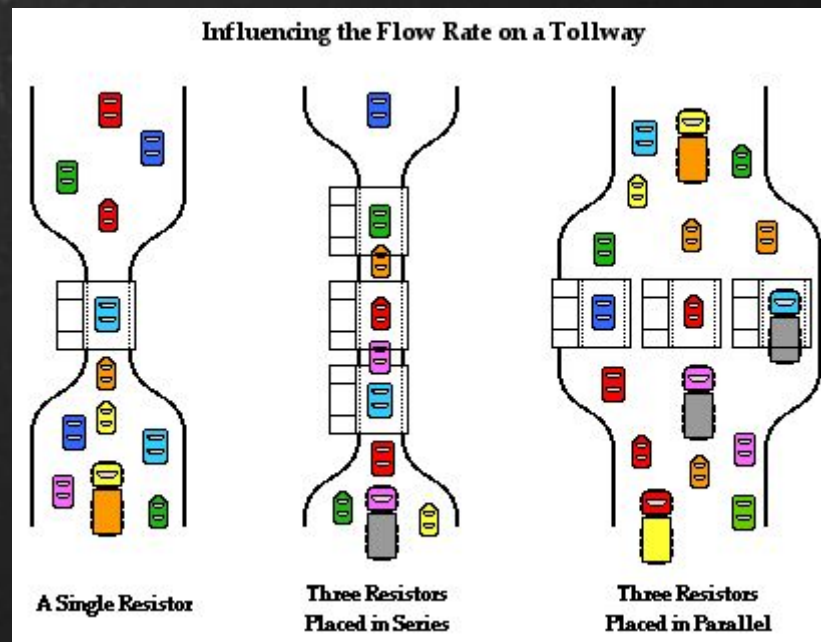
$$I = \frac{V}{R}$$

$$R = \frac{V}{I}$$

$$V = I \times R$$



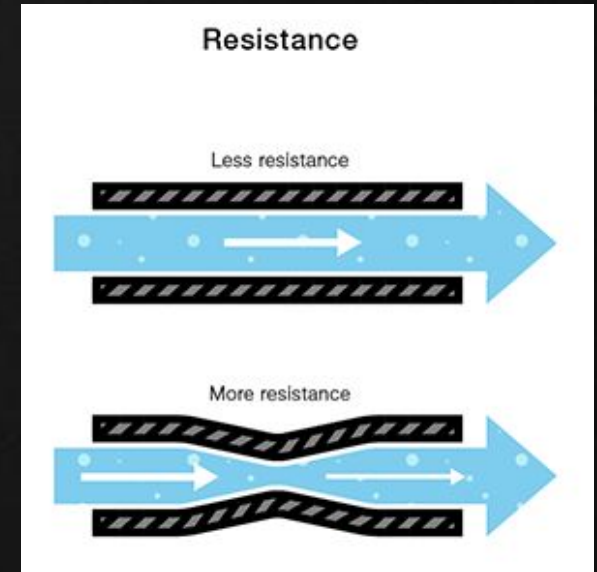
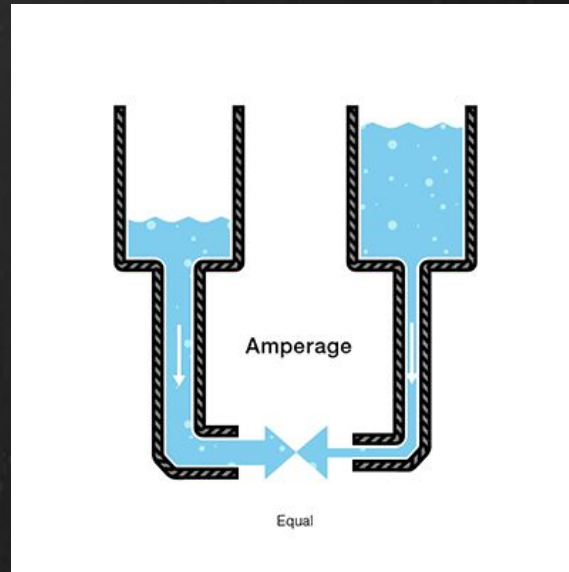
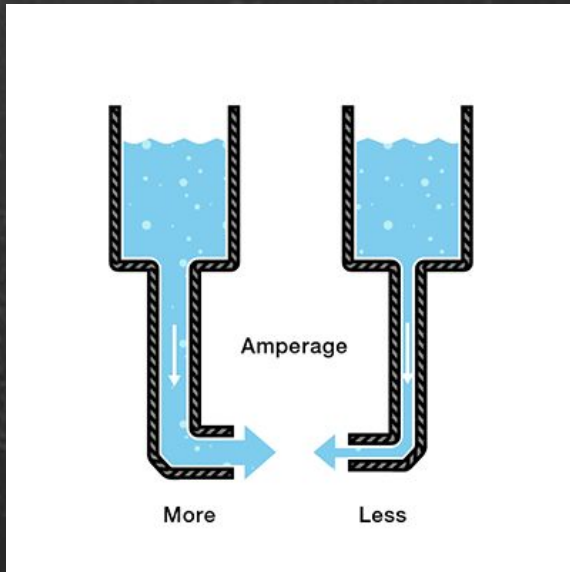
# Ohm's Law



车流类比

图片取自 <http://www.petervaldivia.com/resistance-and-ohm-law/>

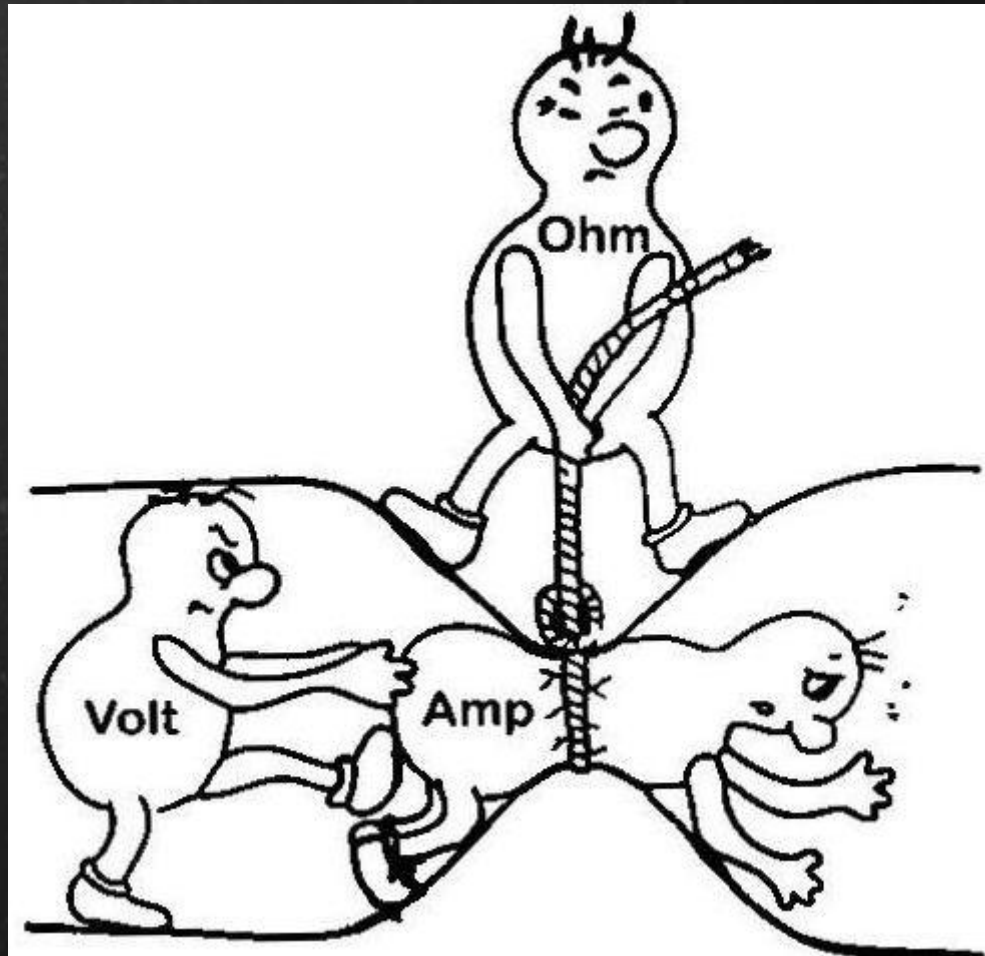
# Ohm's Law



## 水流类比

推荐网页 <https://learn.sparkfun.com/tutorials/voltage-current-resistance-and-ohms-law>

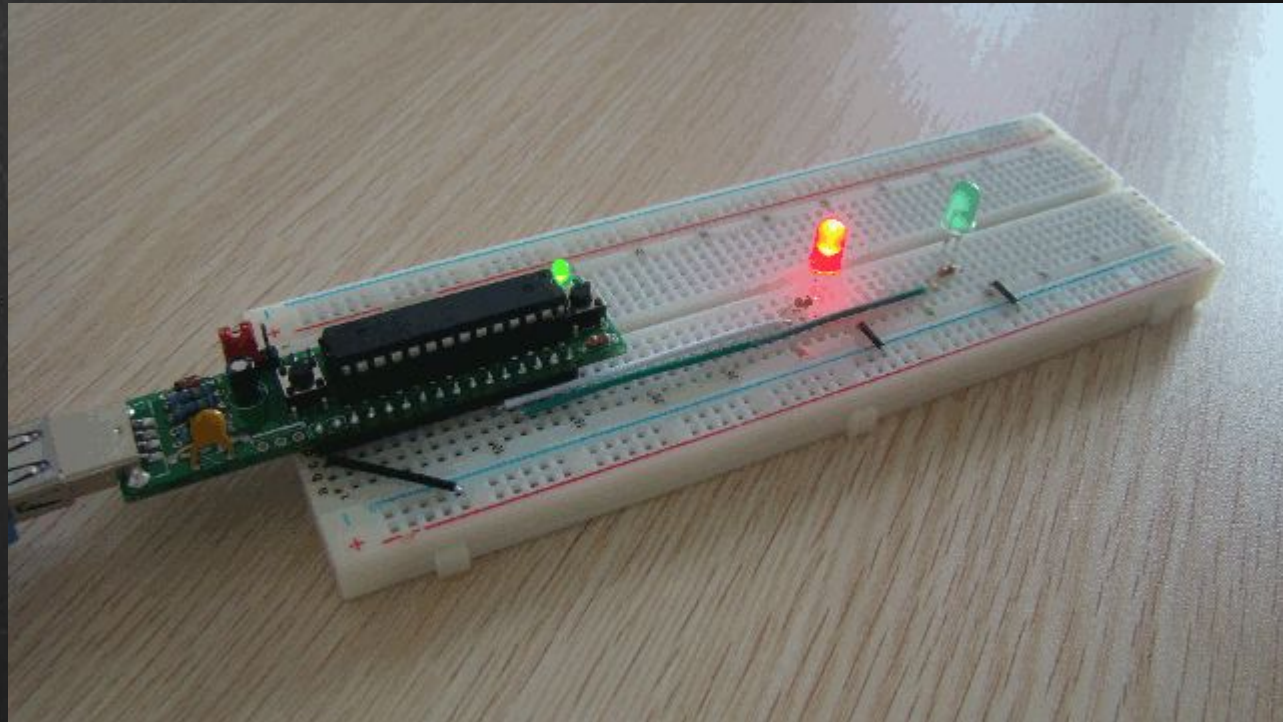
# Ohm's Law



Note: 电子学知识会在后续课程中逐步介绍

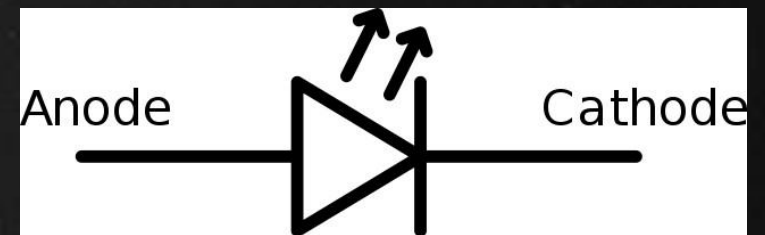
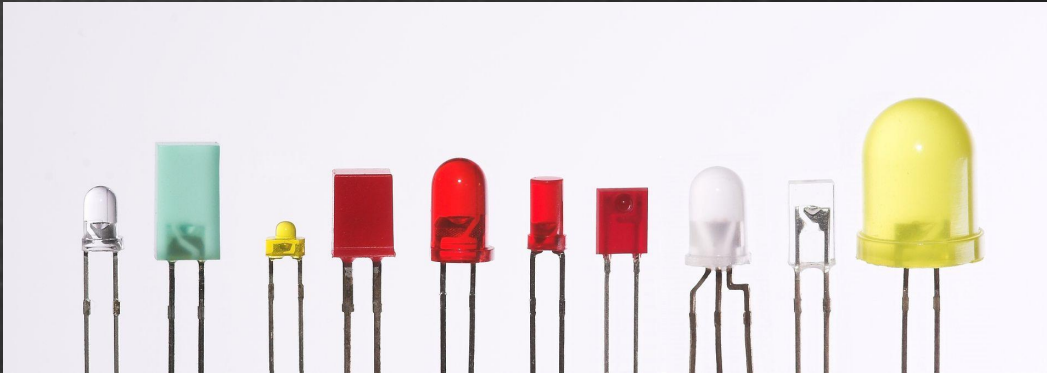


# Let There Be Blink

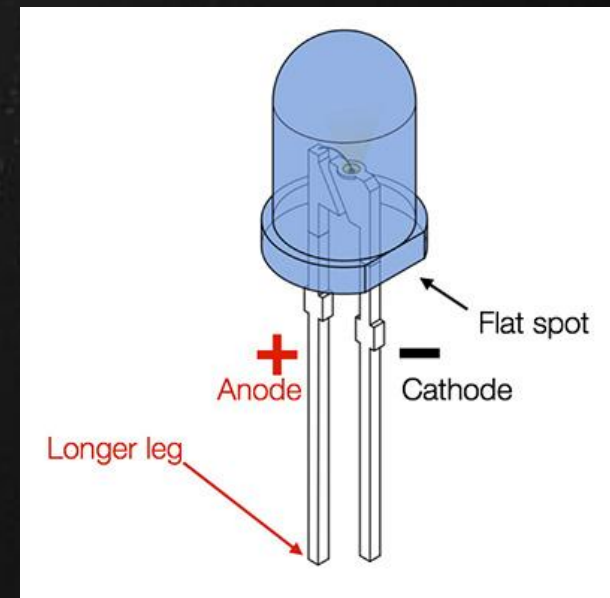




# LED: Light-Emitting Diode

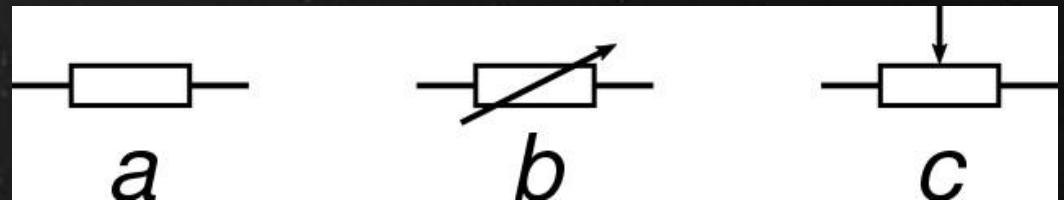
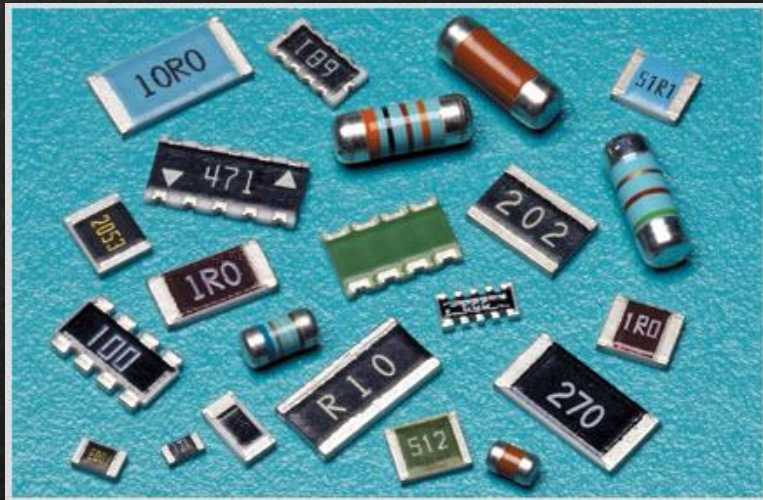
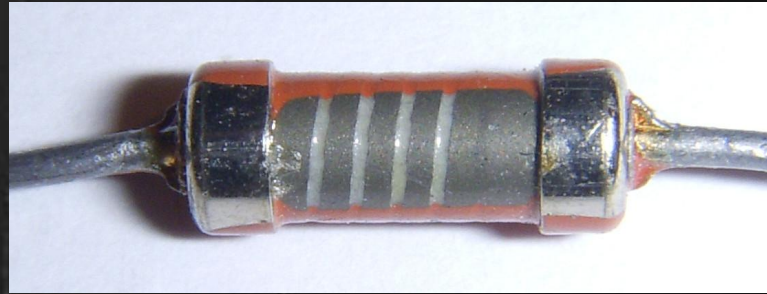


LED 在电路中的符号

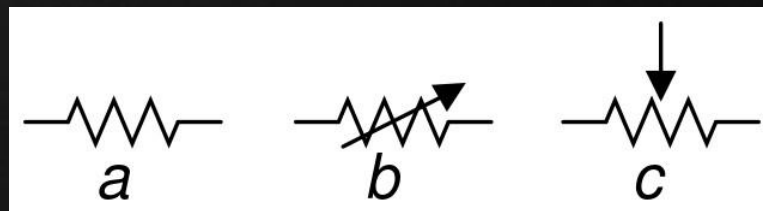


LED 的极性, 长正短负

# 电阻器 Resistors

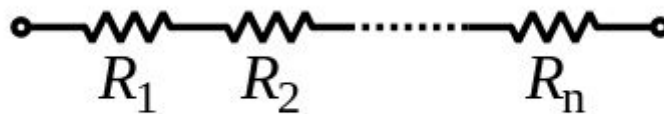


电阻的欧式符号

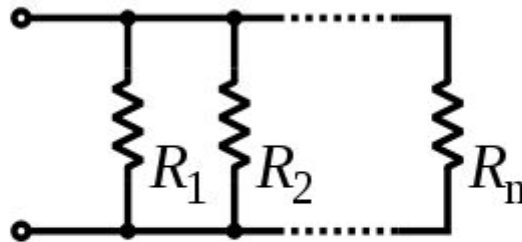


电阻的美式符号

# 电阻器 - 串联和并联



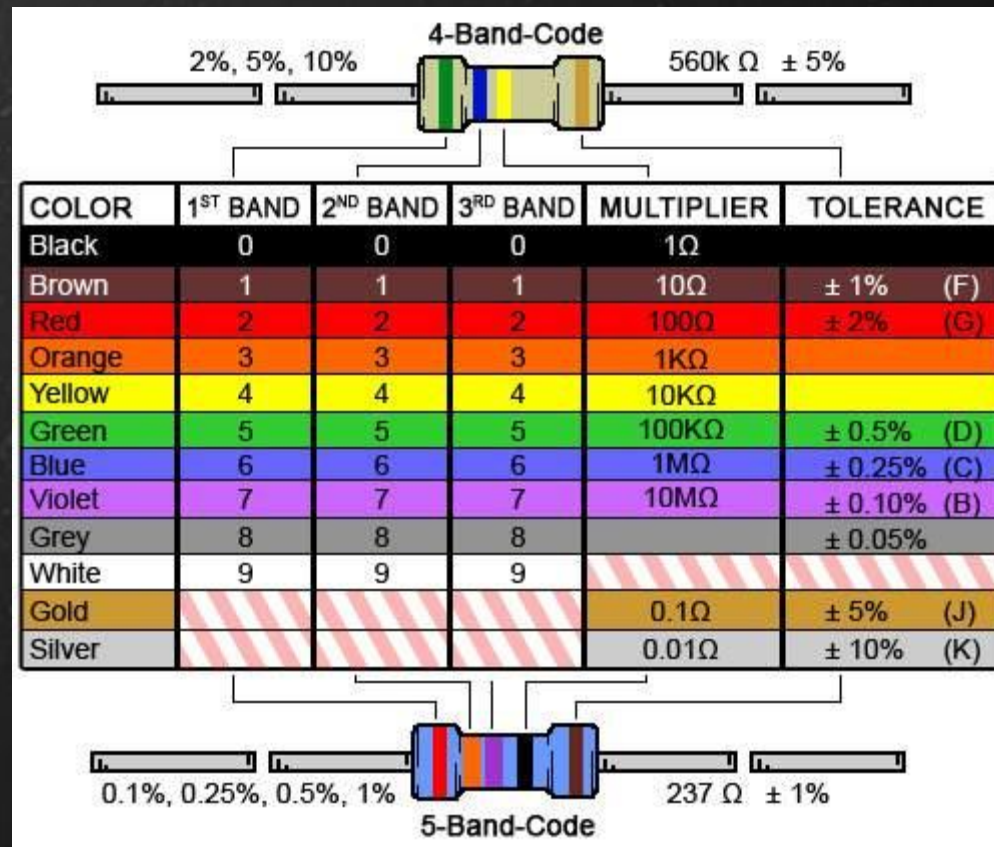
$$R_{\text{eq}} = R_1 + R_2 + \cdots + R_n.$$



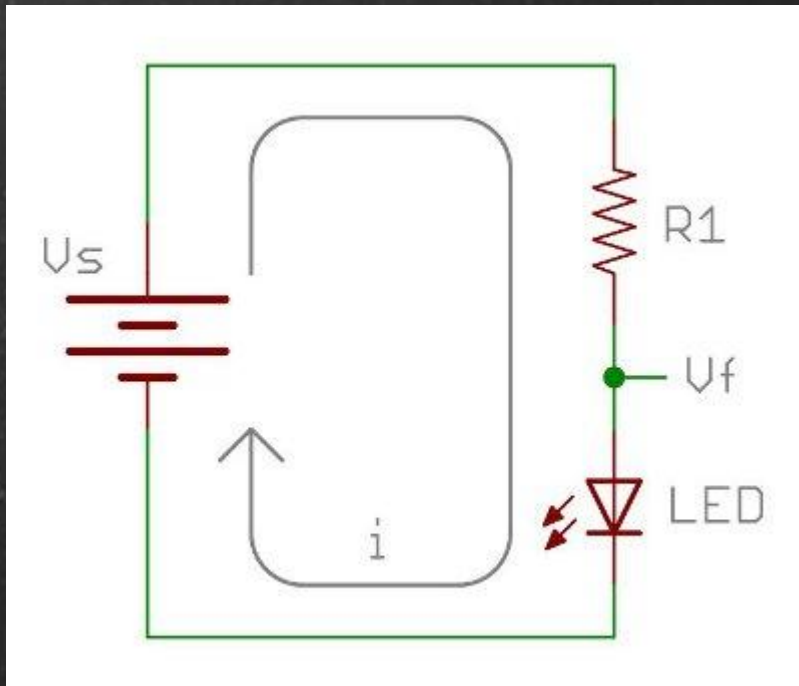
$$\frac{1}{R_{\text{eq}}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}.$$



# How to Read a Resistor



# Let's do some Math



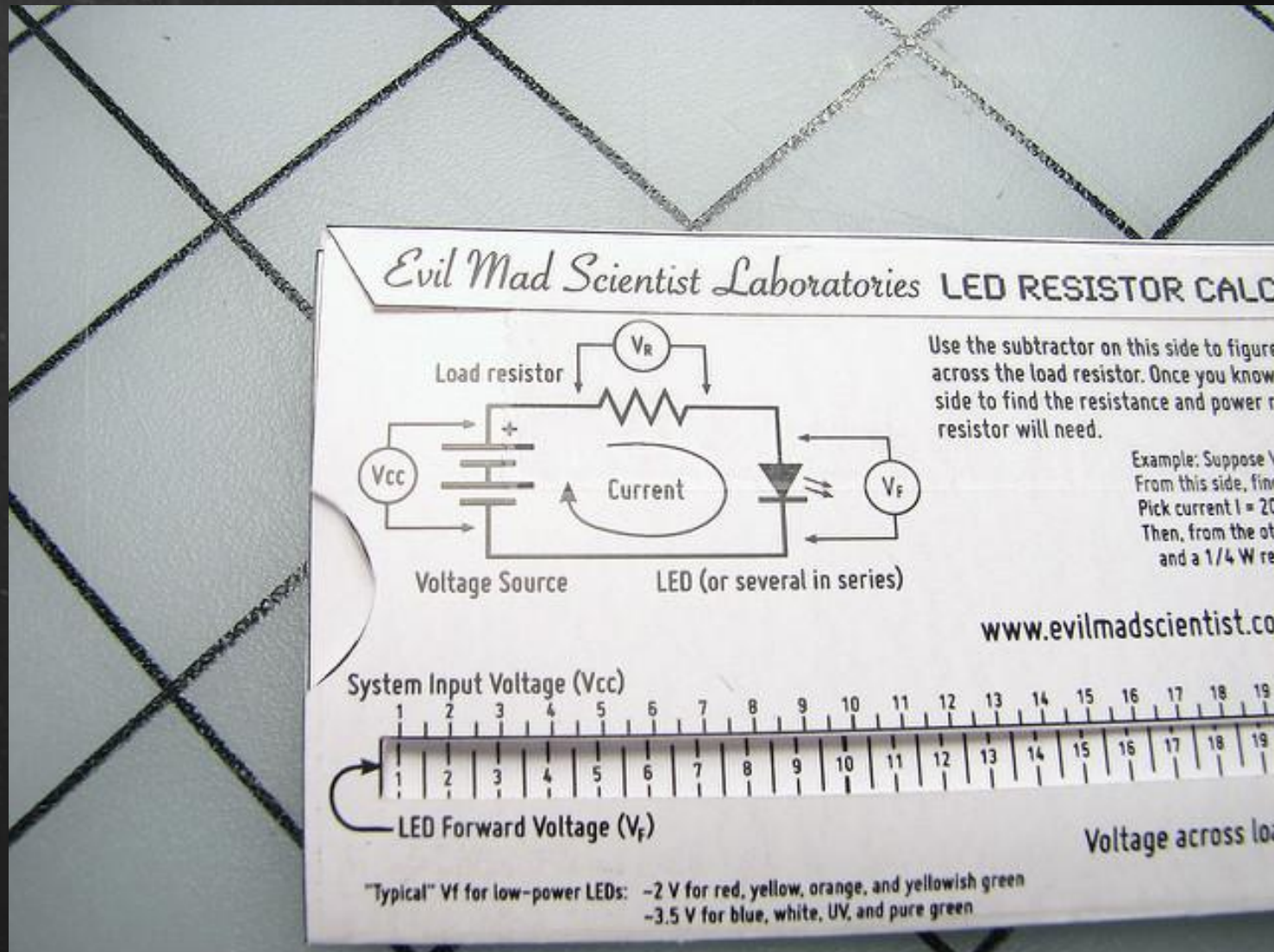
$$I = \frac{V_s - V_f}{R_1}$$

$$\frac{5V - 1.7V}{10mA} = 330 \text{ Ohm}$$

Try it:

- 用万用表测量 LED 两边的压降
- 更换  $R_1$  的值观察 LED 的亮度
- 不同颜色的 LED 有不同的  $V_f$ , find it out.

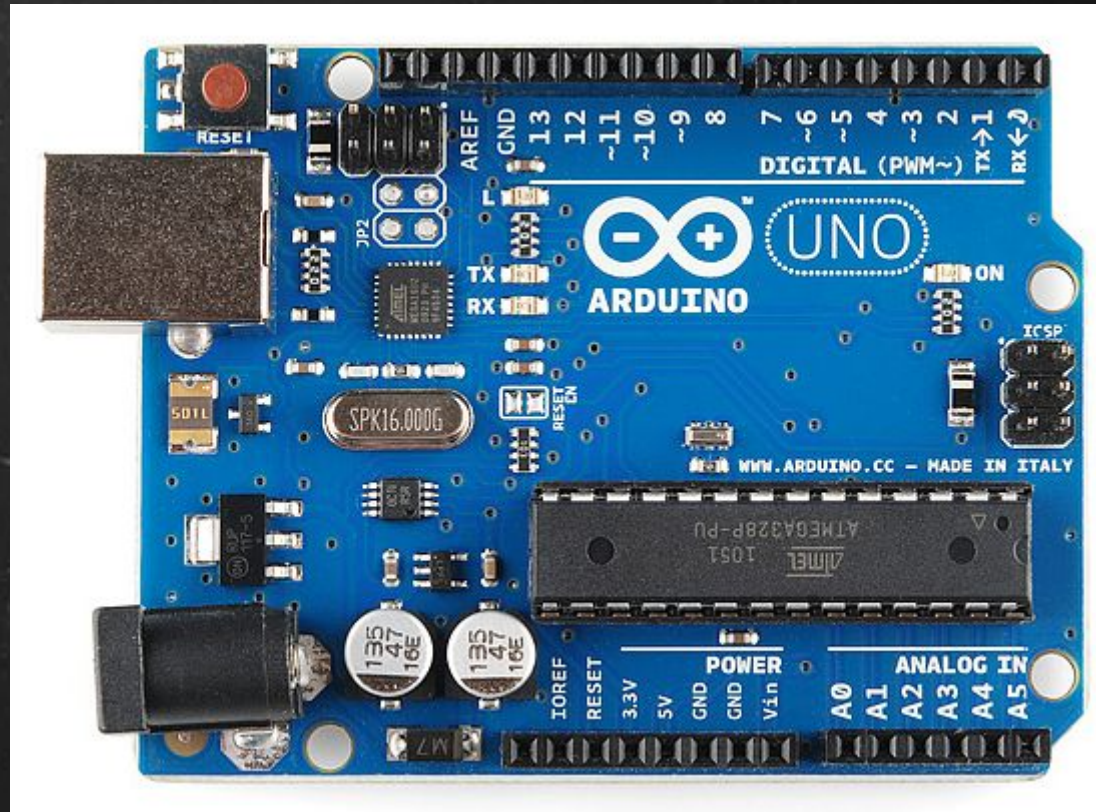
# Let's do some Math



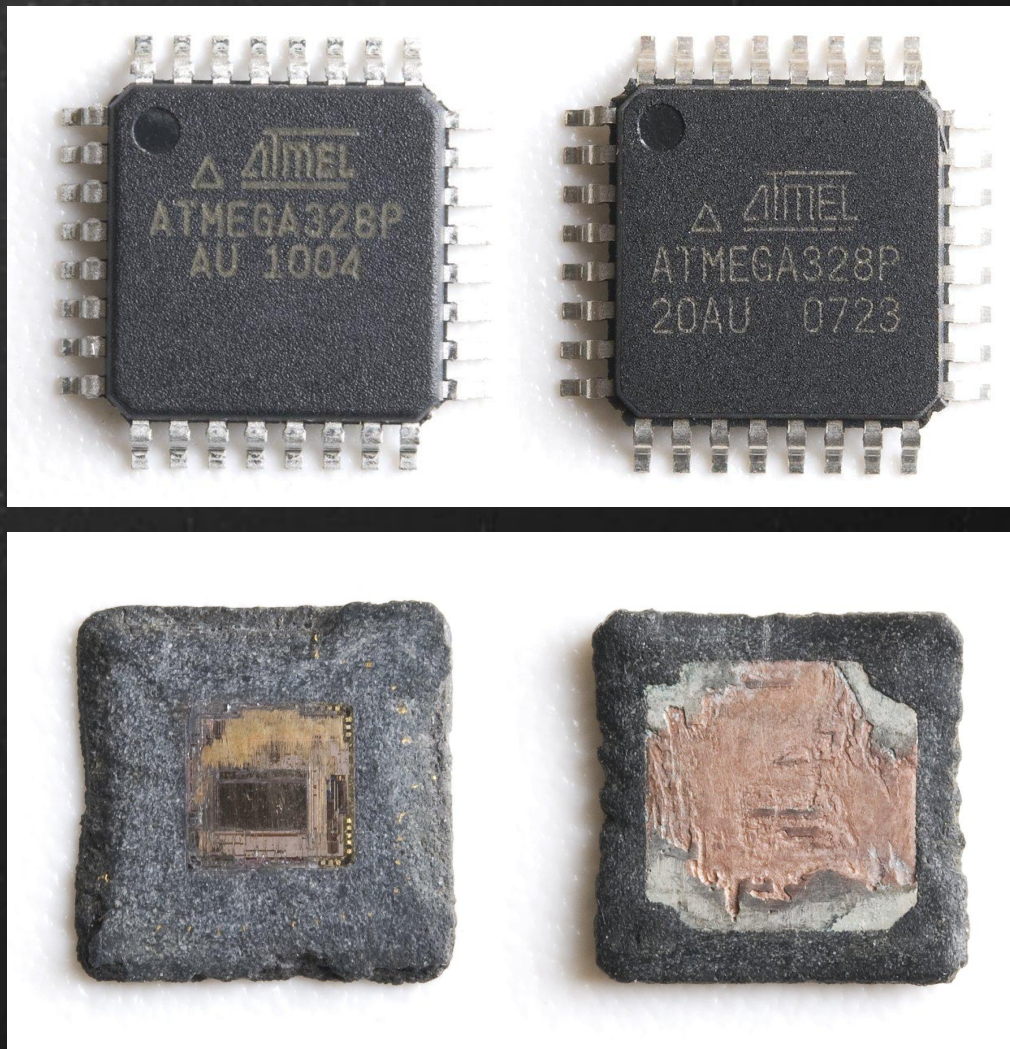
来源 Wallet-size LED Resistance Calculator (Evil Mad Scientist)



# Arduino: A mini computer



# 微控制器(1)



照片取自 [Revisiting the Counterfeit ATmega328s \(Sparkfun\)](#)



# 微控制器 - Off Topic

## [The Visual 6502](#)

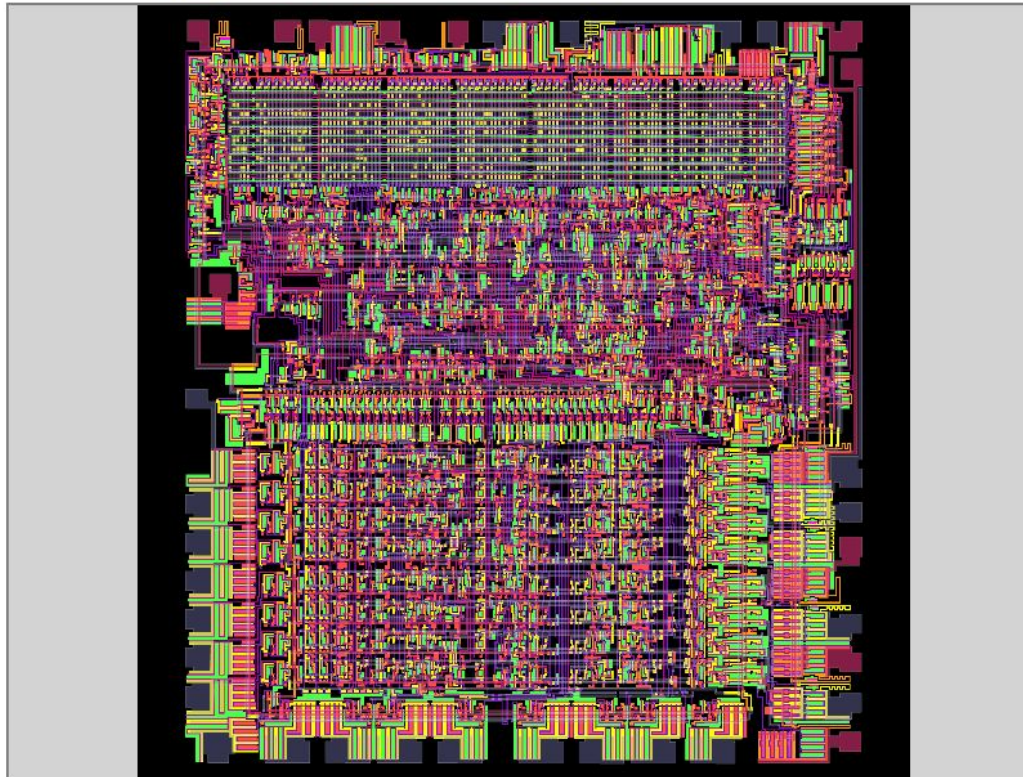
[FAQ](#) [Blog](#) [Links](#)

This simulator uses HTML5 features only found on the latest versions of browsers and needs lots of RAM. If you have trouble, please [check compatibility](#).

Keyboard controls: 'z' to zoom in, 'x' to zoom out, 'n' to step the simulation.

Mouse controls: Left-click and drag to scroll around (when you're zoomed in.)

More information in the [User Guide](#).



▶◀◻▶ ... or try [Advanced](#)

halfcyc:0 phi0:0 AB:0000 D:a9 RnW:1  
PC:0000 A:aa X:00 Y:00 SP:fd nv-BdIZc  
Hz: 1.0

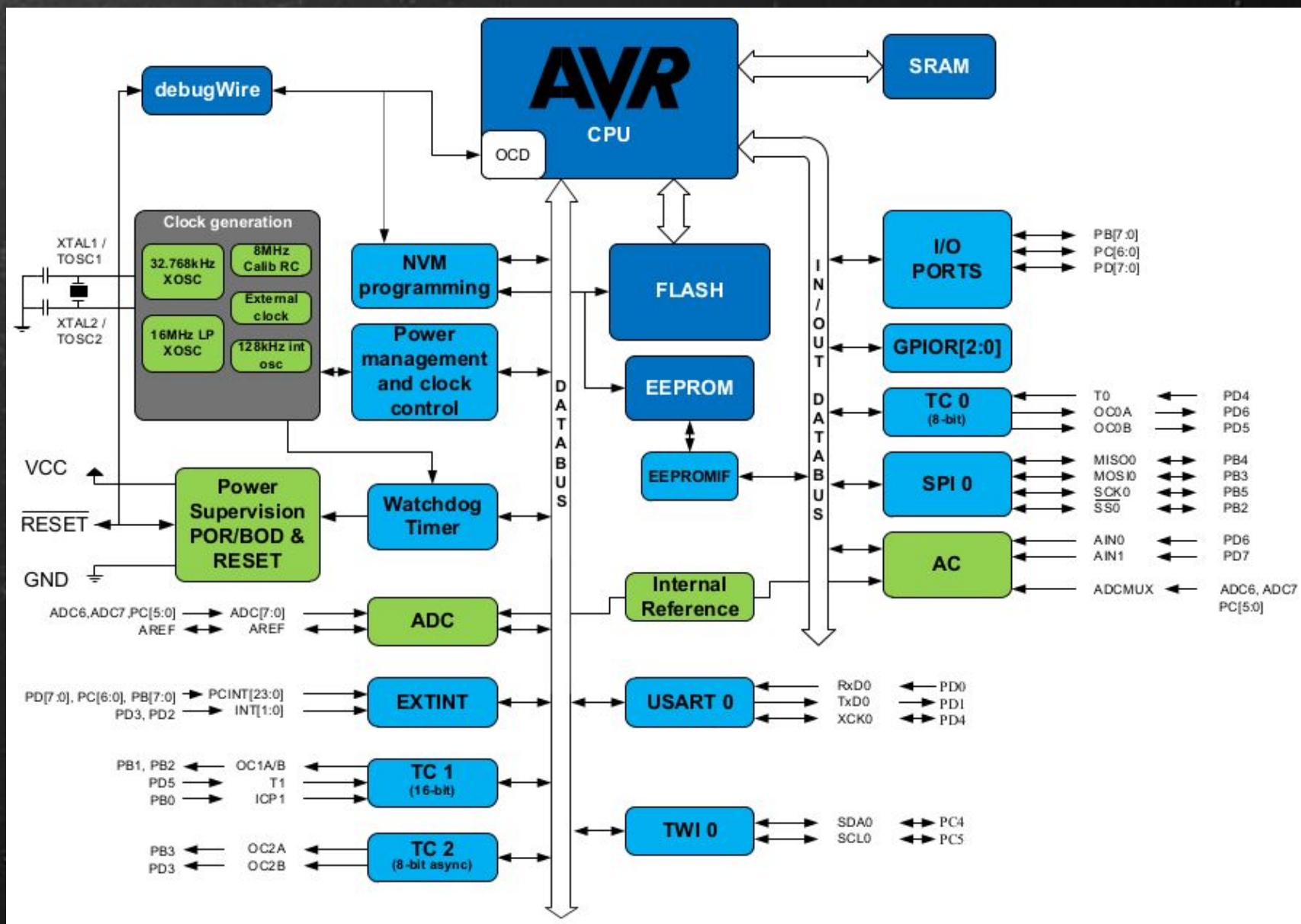
```
0000: a9 00 20 10 00 4c 02 00 00 00 00 00 00 00 00 40
0010: e8 88 e6 0f 38 69 02 60 00 00 00 00 00 00 00 00
0020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0070: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0180: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01f0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Super cool project

Visual6502 <http://www.visual6502.org/JSSim/index.html>

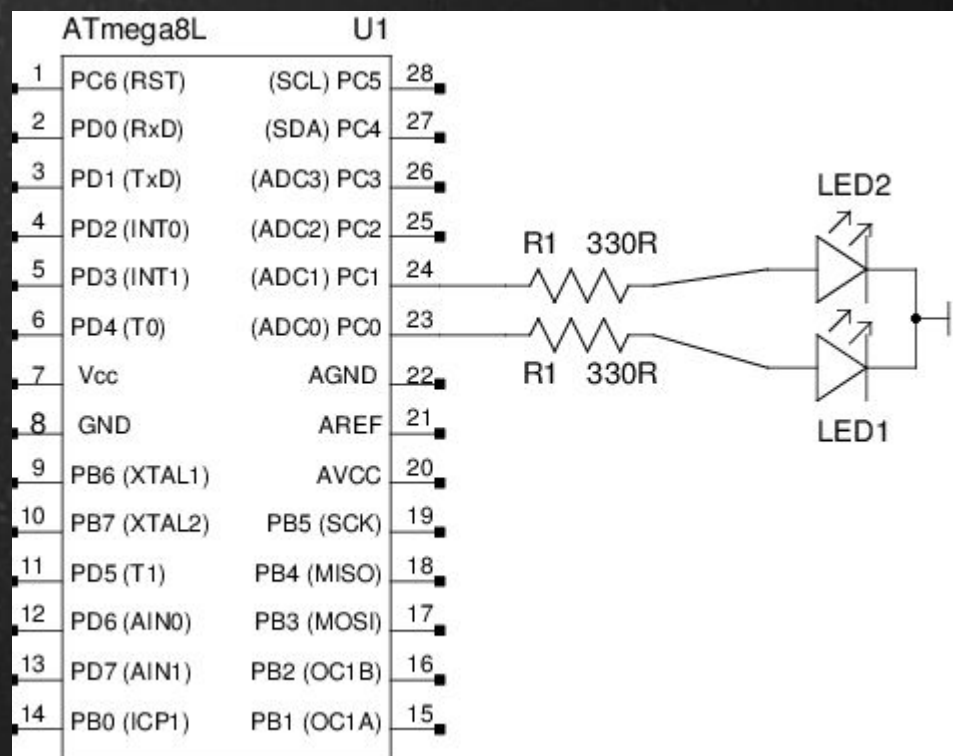


# 微控制器-内部框图



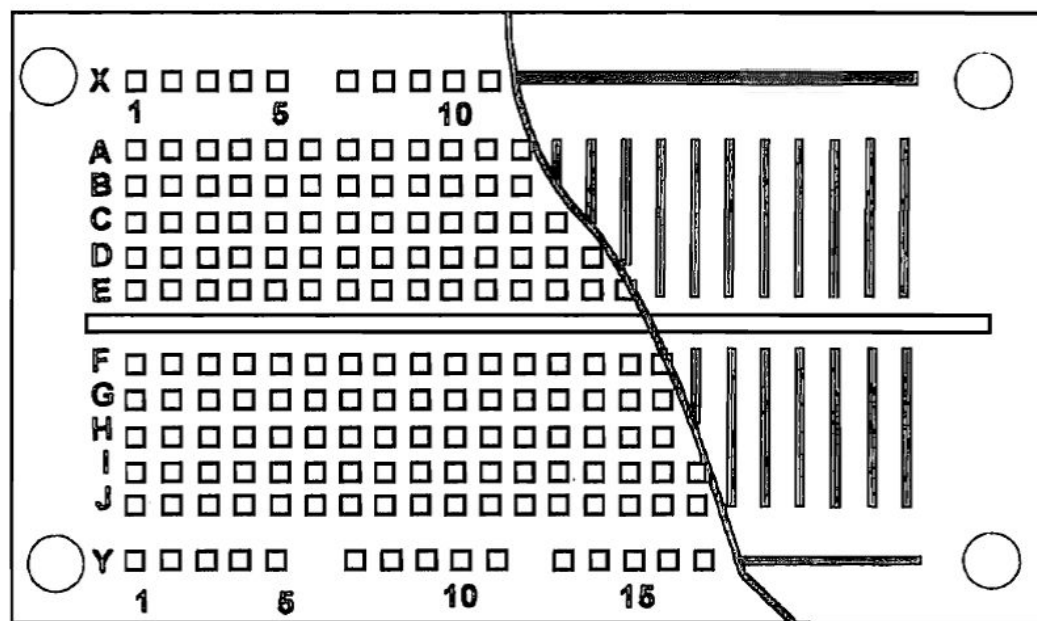
更多细节请见 [Atmega328 的 datasheet](#)

# 接线





# 面包板



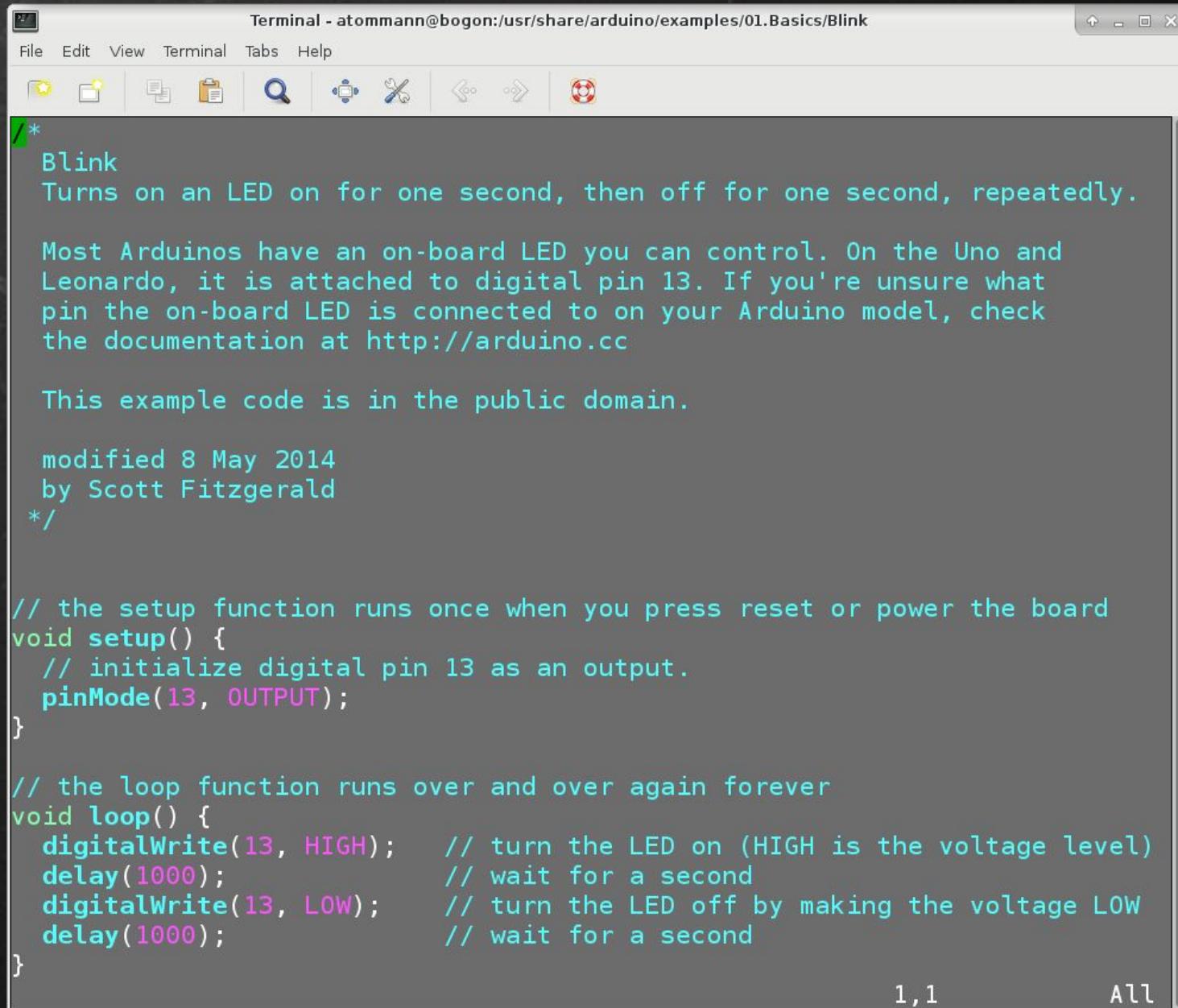
面包板的内部结构



我们现在回到 1900 年左右(那时还没有晶体管,也没有现在面包店里卖的切片面包,切片面包在 1928 年首次出现),人们做原型电路的时候就是用切面包用的砧板来搭电路的,因为这种木板很容易弄到而且也不贵,这个词一直沿用到今天。



# 用软件控制硬件 - The Arduino way



The image shows a screenshot of the Arduino IDE's terminal window. The title bar reads "Terminal - atommann@bogon:/usr/share/arduino/examples/01.Basics/Blink". The menu bar includes "File", "Edit", "View", "Terminal", "Tabs", and "Help". The toolbar contains icons for opening files, saving, undo, redo, and other standard editing functions. The terminal area displays the following text:

```
/*  
Blink  
Turns on an LED on for one second, then off for one second, repeatedly.  
  
Most Arduinos have an on-board LED you can control. On the Uno and  
Leonardo, it is attached to digital pin 13. If you're unsure what  
pin the on-board LED is connected to on your Arduino model, check  
the documentation at http://arduino.cc  
  
This example code is in the public domain.  
  
modified 8 May 2014  
by Scott Fitzgerald  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin 13 as an output.  
  pinMode(13, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(13, HIGH);  // turn the LED on (HIGH is the voltage level)  
  delay(1000);             // wait for a second  
  digitalWrite(13, LOW);   // turn the LED off by making the voltage LOW  
  delay(1000);             // wait for a second  
}
```

At the bottom right of the terminal window, the text "1,1" and "All" are visible, indicating the current line and column position.

# Arduino 的优势 - 软件库

```
Terminal - atommann@bogon:/usr/share/arduino/libraries
File Edit View Terminal Tabs Help

[atommann@bogon libraries]$ pwd
/usr/share/arduino/libraries
[atommann@bogon libraries]$ ll
total 68
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Bridge
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Esplora
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Ethernet
drwxr-xr-x 8 root root 4096 Oct 23 2016 FastLED
drwxr-xr-x 5 root root 4096 Apr 25 12:50 Firmata
drwxr-xr-x 5 root root 4096 Apr 25 12:50 GSM
drwxr-xr-x 4 root root 4096 Apr 25 12:50 LiquidCrystal
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Robot_Control
drwxr-xr-x 4 root root 4096 Apr 25 12:50 RobotIRremote
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Robot_Motor
drwxr-xr-x 4 root root 4096 Apr 25 12:50 SD
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Servo
drwxr-xr-x 3 root root 4096 Apr 25 12:50 SpacebrewYun
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Stepper
drwxr-xr-x 4 root root 4096 Apr 25 12:50 Temboo
drwxr-xr-x 5 root root 4096 Apr 25 12:50 TFT
drwxr-xr-x 4 root root 4096 Apr 25 12:50 WiFi
[atommann@bogon libraries]$
```



# 用软件控制硬件 - 中级语言 C

```
Terminal - atommann@bogon:~/arduino/sustech/lect02/src-c
File Edit View Terminal Tabs Help

#include "avr/io.h"           // (1) 包含头文件
#define F_CPU 16000000UL      // 设置 CPU 时钟频率
#include "util/delay.h"       // (2) 我们需要的 delay 函数

int main(void)               // (3)
{
    DDRB = 0b00100000;       // (4) 设置为输出
    PORTB = 0x00;            // (5) 上电时的输出

    while (1)                // (6) loop forever
    {
        PORTB = 0b00100000;   // (7) High
        _delay_ms(1000);      // (8) do nothing
        PORTB = 0b00000000;   // (9) Low
        _delay_ms(1000);      // (10) do nothing
    }
}
```

4,0-1 All



# 中级语言 C - C 标准库

**avr-libc** 2.0.0

Standard C library for AVR-GCC

**AVR Libc**  
**Home**  
**Page**



**Main Page**

**User  
Manual**

**Library  
Reference**

**FAQ**

**Example  
Projects**

## Modules

Here is a list of all modules:

<alloca.h>: Allocate space in the stack	
<assert.h>: Diagnostics	
<ctype.h>: Character Operations	
<errno.h>: System Errors	
<inttypes.h>: Integer Type conversions	
<math.h>: Mathematics	
<setjmp.h>: Non-local goto	
<stdint.h>: Standard Integer Types	
<stdio.h>: Standard IO facilities	
<stdlib.h>: General utilities	
<string.h>: Strings	
<time.h>: Time	
<avr/boot.h>: Bootloader Support Utilities	
<avr/cpufunc.h>: Special AVR CPU functions	
<avr/eeprom.h>: EEPROM handling	
<avr/fuse.h>: Fuse Support	
<avr/interrupt.h>: Interrupts	
<avr/io.h>: AVR device-specific IO definitions	

网址 <http://www.nongnu.org/avr-libc/>

# 用软件控制硬件 - 汇编语言

```
Terminal - atommann@bogon:~/arduino/sustech/lect02/src-assembler
File Edit View Terminal Tabs Help

.set PINB,      0x03
.set DDRB,      0x04
.set TCCR0B,    0x25
.set TCNT0,     0x26
.set LED_MASK, 0b00100000
.set PS_1024,   0b00000101

setup:
    ldi r16, PS_1024      ; Set r16 with prescaler 1024 value
    out TCCR0B, r16       ; Set the TCCR0B to 1024
    ldi r16, LED_MASK     ; Set r16 to the LED bit
    out DDRB, r16         ; Set LED pin to output
    clr r18               ; Clear the saved timer

loop:
    ldi r20, 61           ; Initialize our software counter

check_timer:
    in r17, TCNT0         ; Read the timer
    cp r17, r18           ; Compare with previous value
    mov r18, r17          ; Save current value
    brsh check_timer      ; unless the timer has decreased, repeat

decrement:
    dec r20               ; decrement the software counter
    brne check_timer      ; if not zero, go back to checking the timer

toggle:
    out PINB, r16         ; toggle the LED
    rjmp loop

7,0-1 All
```



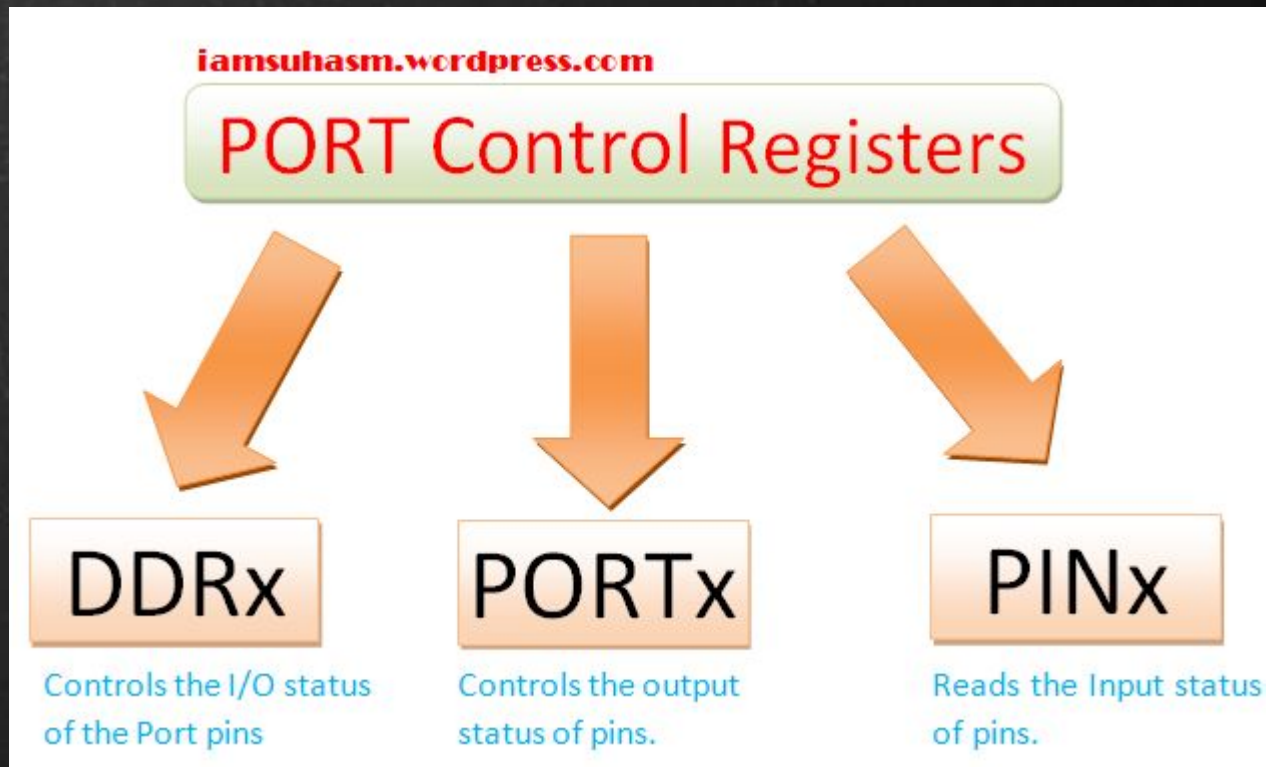
# 用软件控制硬件 - 机器码

Address	Opcode	Mnemonic	Description
0000	7A	REQ	Reset 'Q' = 0 (turns 'Q' LED off)
0001	F8	LDI	Load data (10) into 'D' register
0002	10	(data)	<i>data for preceding instruction</i>
0003	B1	PHI	Set high order byte of register R1 = 'D' (10)
0004	21	DEC	Decrement contents of register R1
0005	91	GHI	Set 'D' equal to contents of high order byte of R1
0006	3A	BNZ	If 'D' $\neq$ 0, short branch to address 00 <b>04</b>
0007	04	(data)	<i>data (low order address) for preceding instruction</i>
0008	31	BQ	If 'Q' = 1, short branch to address 00 <b>00</b>
0009	00	(data)	<i>data (low order address) for preceding instruction</i>
000A	7B	SEQ	Set 'Q' = 1 (turns 'Q' LED on)
000B	30	BR	Unconditional short branch to address 00 <b>01</b>
000C	01	(data)	<i>data (low order address) for preceding instruction</i>

This example code runs on COSMAC ELF, it blinks an LED.



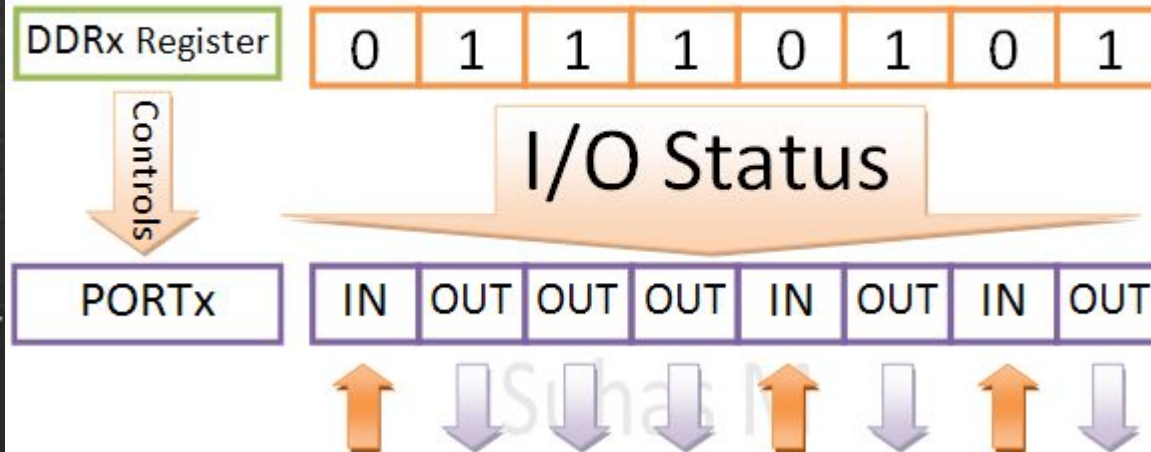
# 到底发生了什么？ - Registers



# 到底发生了什么？控制输入输出

[iamsuhasm.wordpress.com](http://iamsuhasm.wordpress.com)

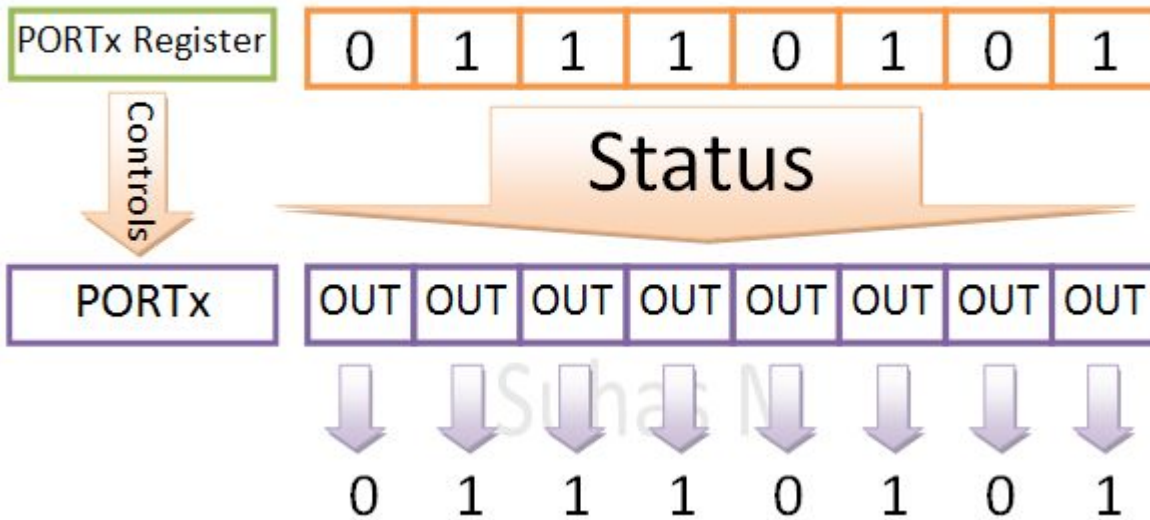
The DDRx register controls if a pin is in Input mode or Output mode



# 到底发生了什么？ - 端口数据 Register

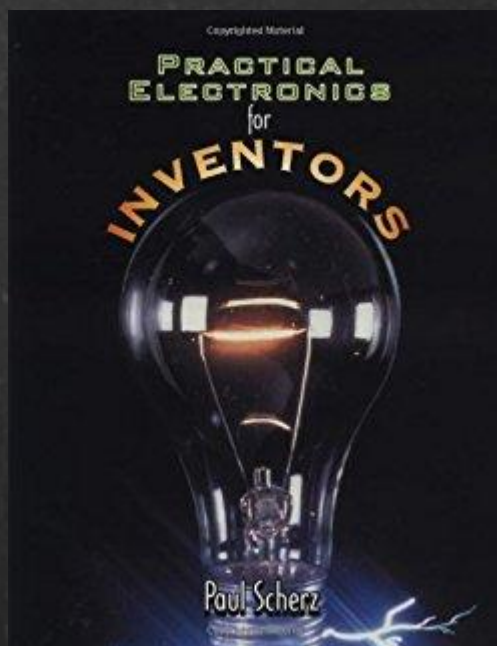
[iamsuhasm.wordpress.com](http://iamsuhasm.wordpress.com)

If all the pins in PORTx are configured as output:

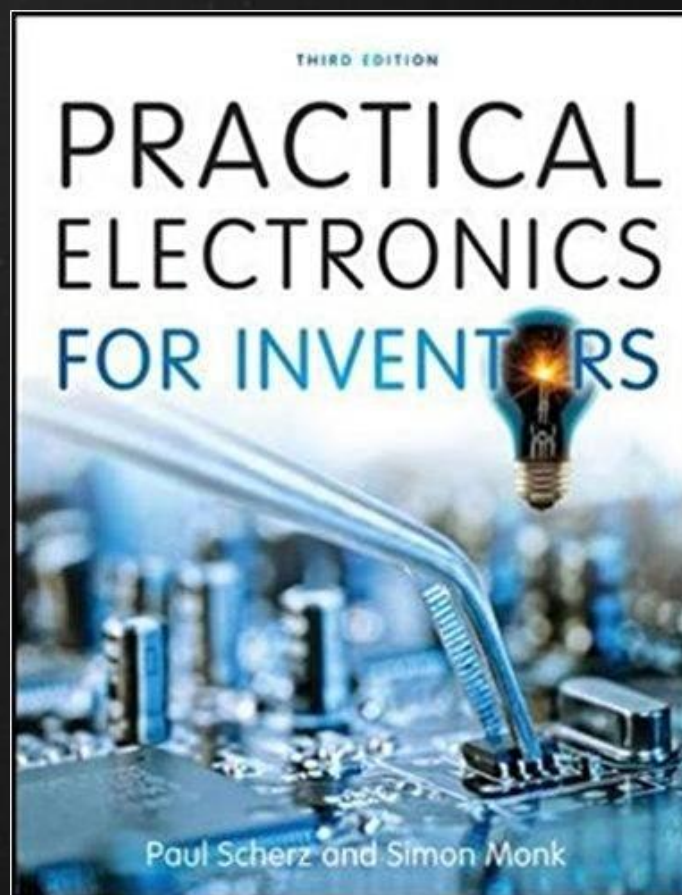




# 推荐的电子学图书(1)



《发明者电子设计宝典》第二版  
《实用电子元器件与电路基础》(第3版)

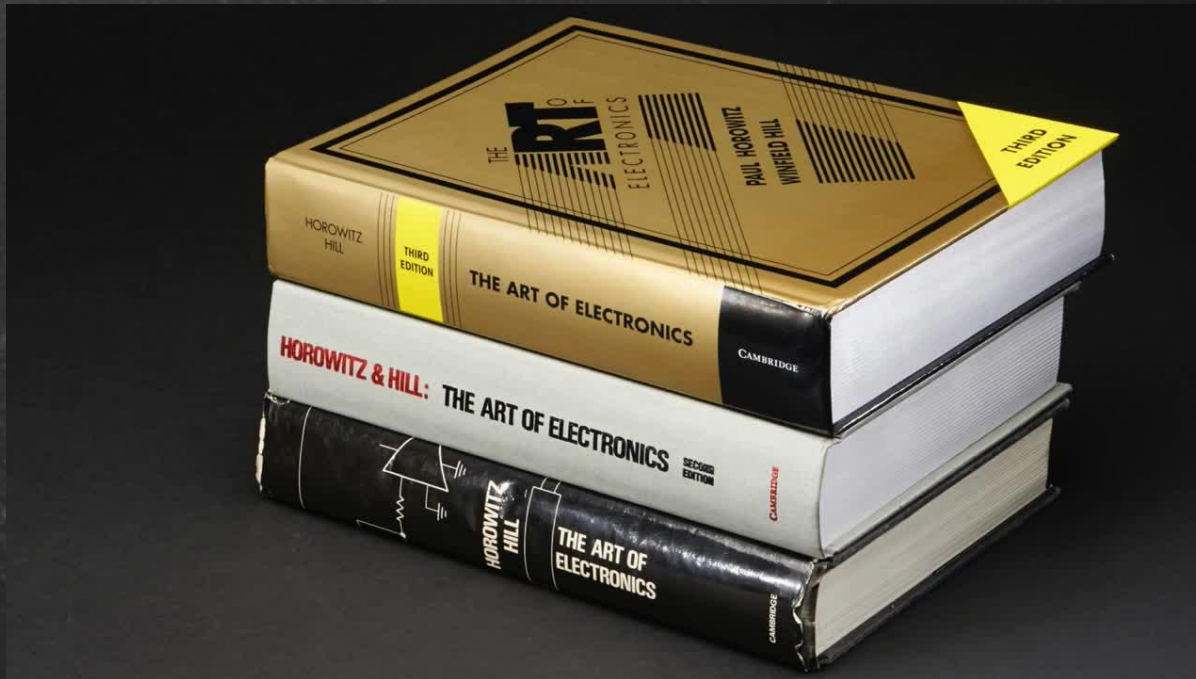


Practical Electronics for Inventors

最新:英文第 4 版

特点:手绘电路, 直观讲解, 电路有参数, 水类比, 易懂

# 推荐的电子学图书(2)



The Art of Electronics《电子学》

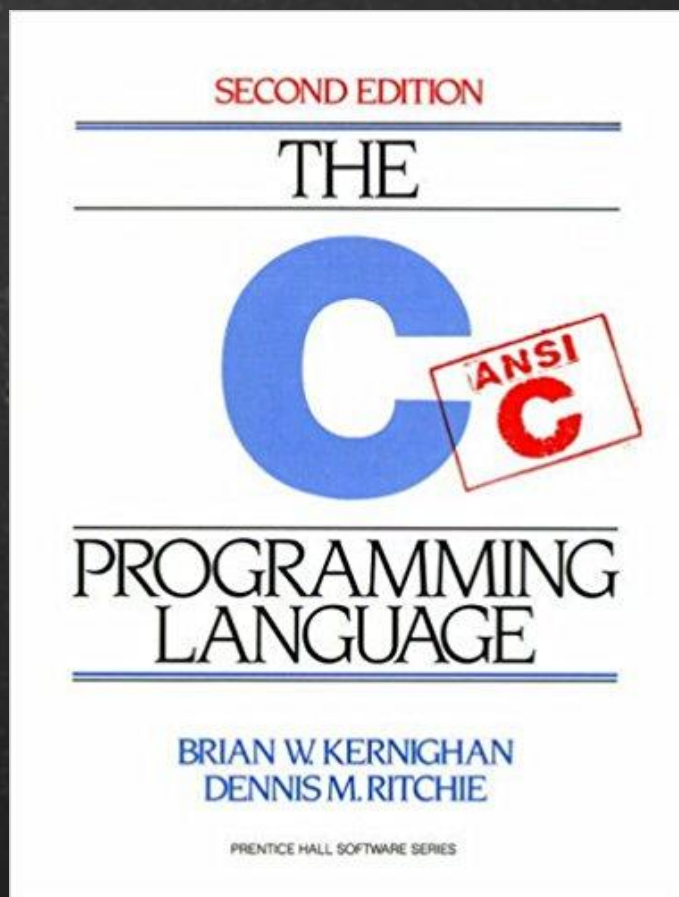
最新:第 3 版

特点:内容全面, The Bible of Electronics

更多 EE book [https://blog.adafruit.com/category/ee\\_bookshelf/](https://blog.adafruit.com/category/ee_bookshelf/)

但是, 别忘记:请多多实践。

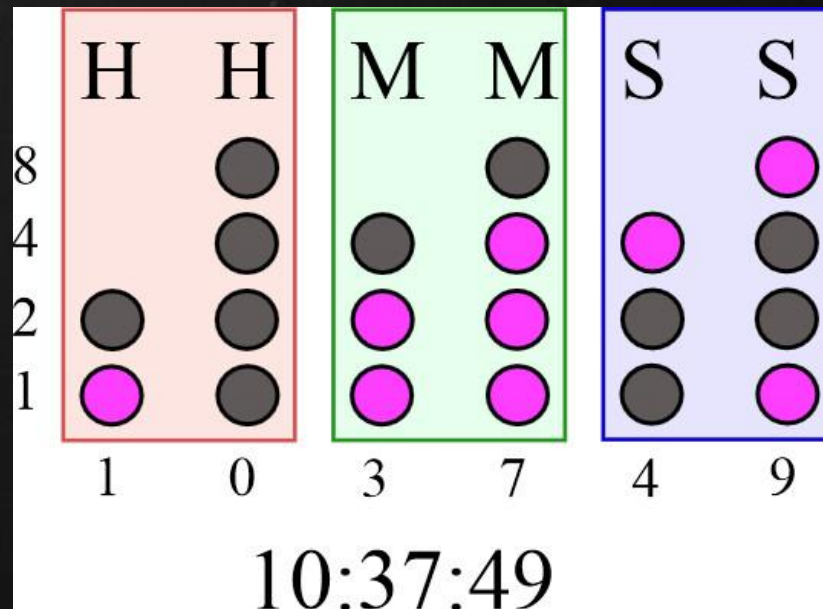
# 推荐的 C 语言书





# 作业

- 安装 Fritzing 并探索软件的使用
- 实现一个 Binary Time Counter
- 用 LED 做 Persistence of Vision 实验, 显示一个字符或图案
- 看看 Cornell EE4760 的项目
  - <http://people.ece.cornell.edu/land/courses/ece4760/>
- 逛华强北电子市场
  - 全世界最大的电子市场, 创客的天堂
  - Optional
- 探索 Arduino 自带例子



# 版权说明

Lot of diagrams/pictures in this slide are from Wikipedia and Google search, I appreciate the author's cool job, if there are some licence issues please email me.

本 slide 的很多图片来自 Wikipedia 和 Google 搜索, 我感谢原创作者的杰出贡献, 如有任何违反版权的事宜, 请联系我。