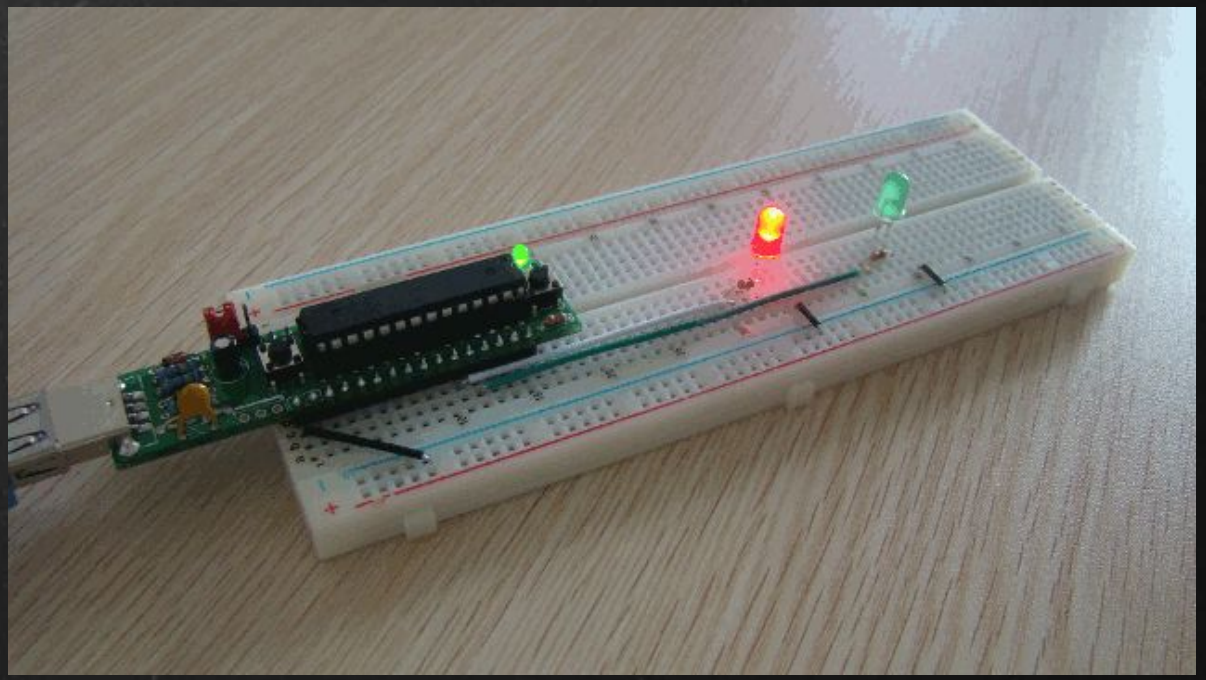
# IDD001 Lecture 2: Let there be blink



Atommann <atommann@gmail.com>

2017 Fall

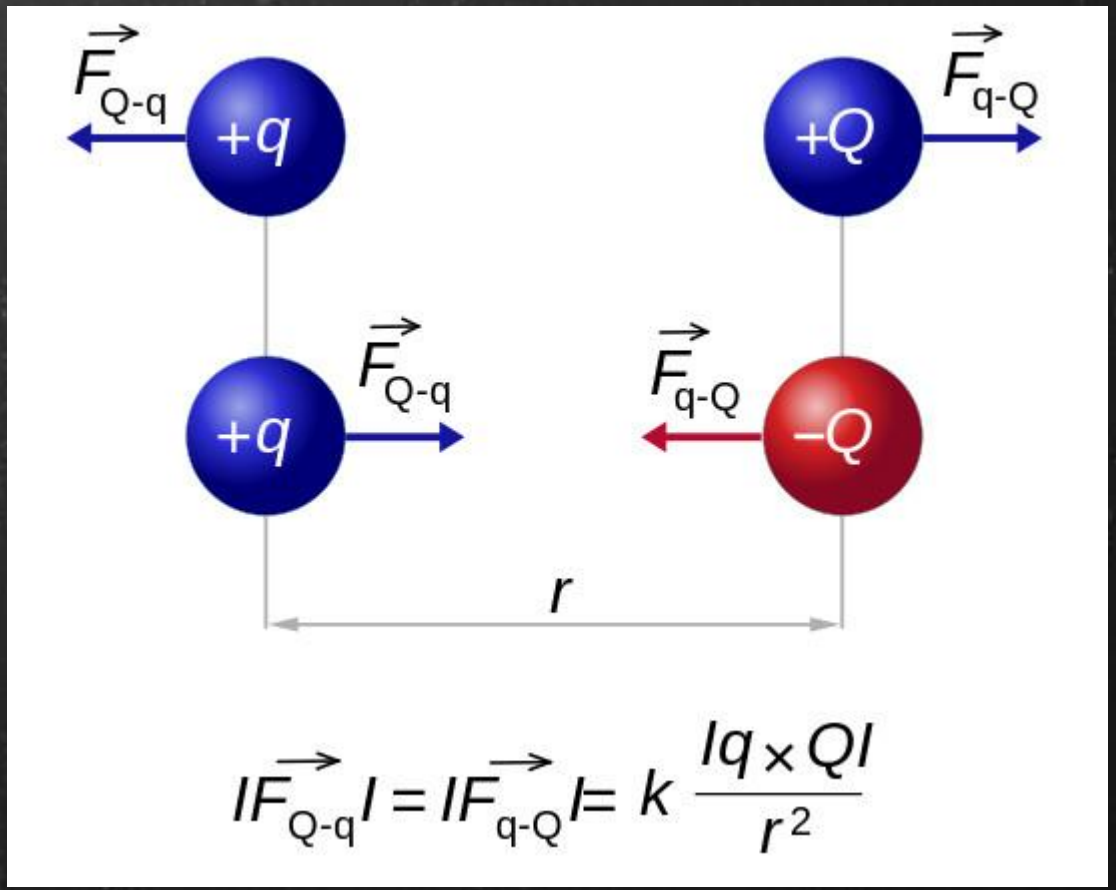
I call this course “Play with Arduinos”

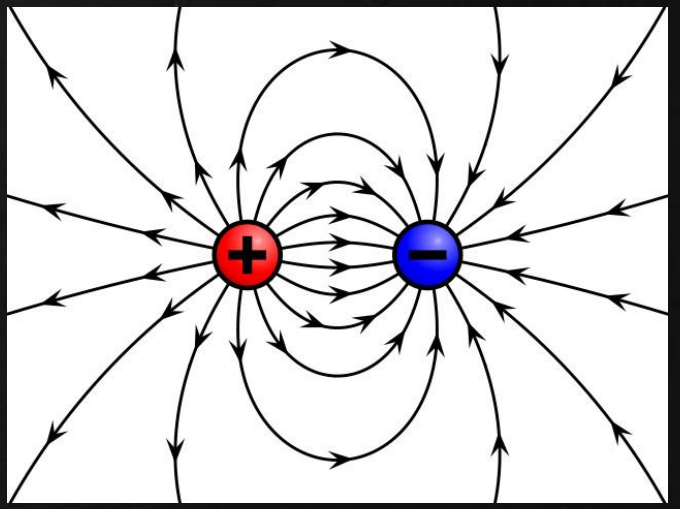
All men[1] by nature desire to know. -- Aristotle

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

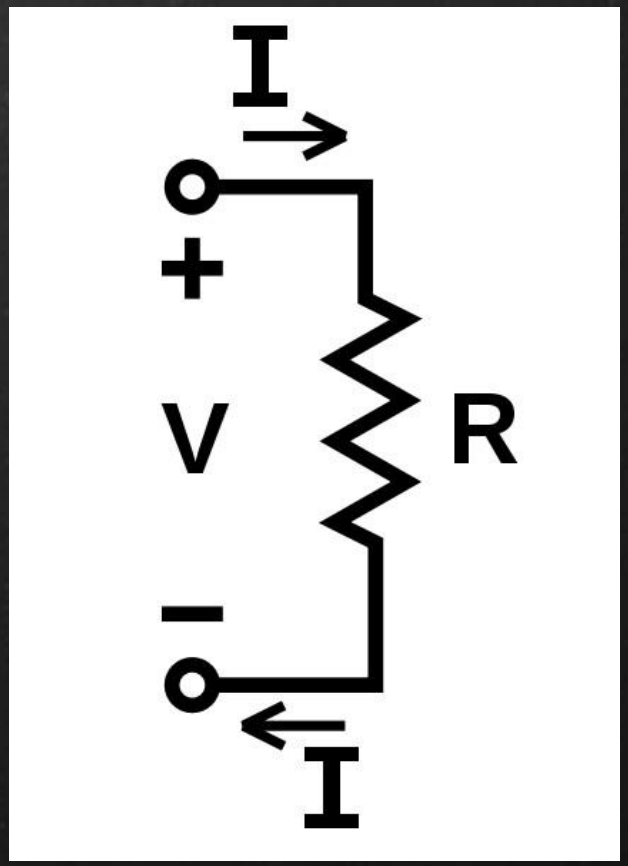



Coulomb's law - Wikipedia

# Ohm's Law

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

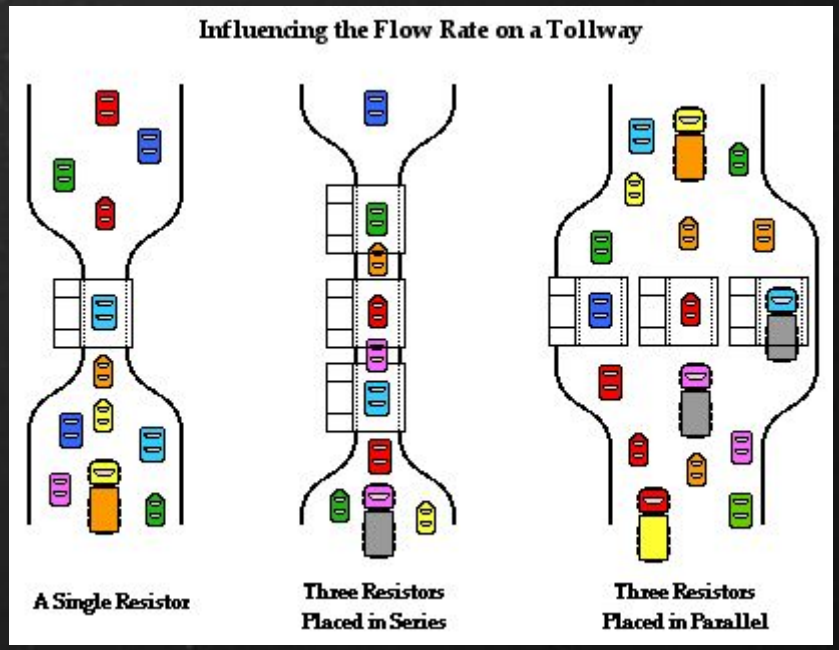



$$I = \frac{V}{R}$$

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

$$V = I \times R$$

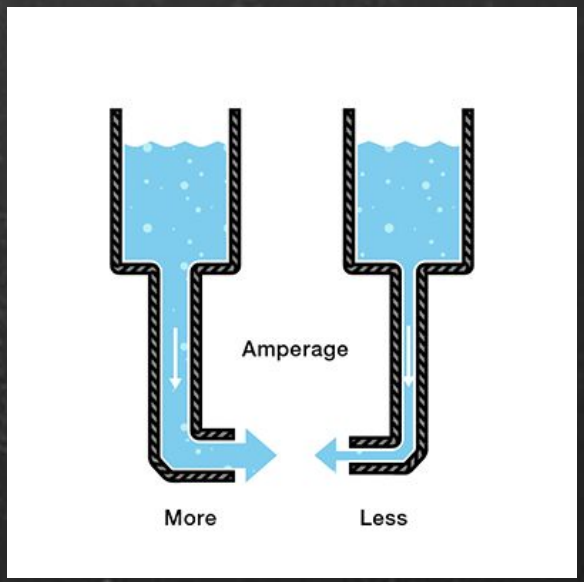
# Ohm's Law

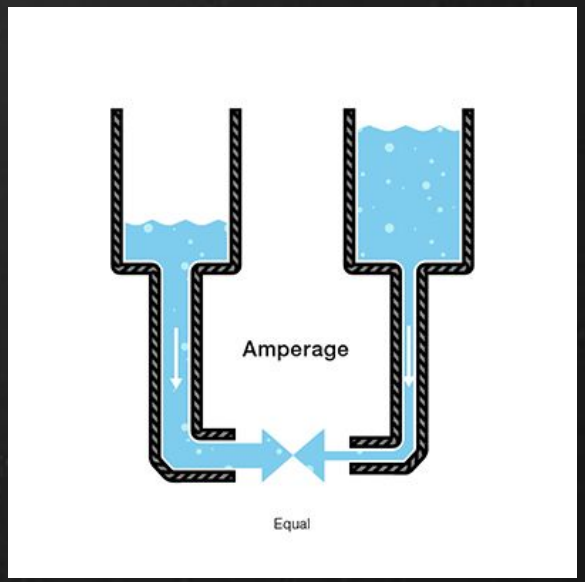


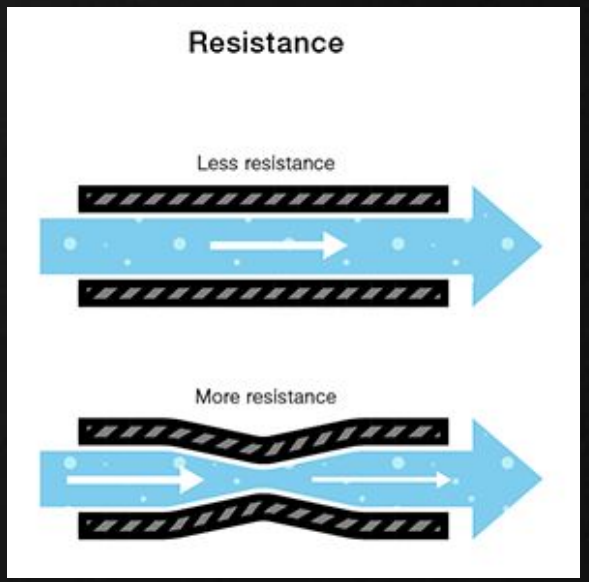

车流类比

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

# Ohm’s Law



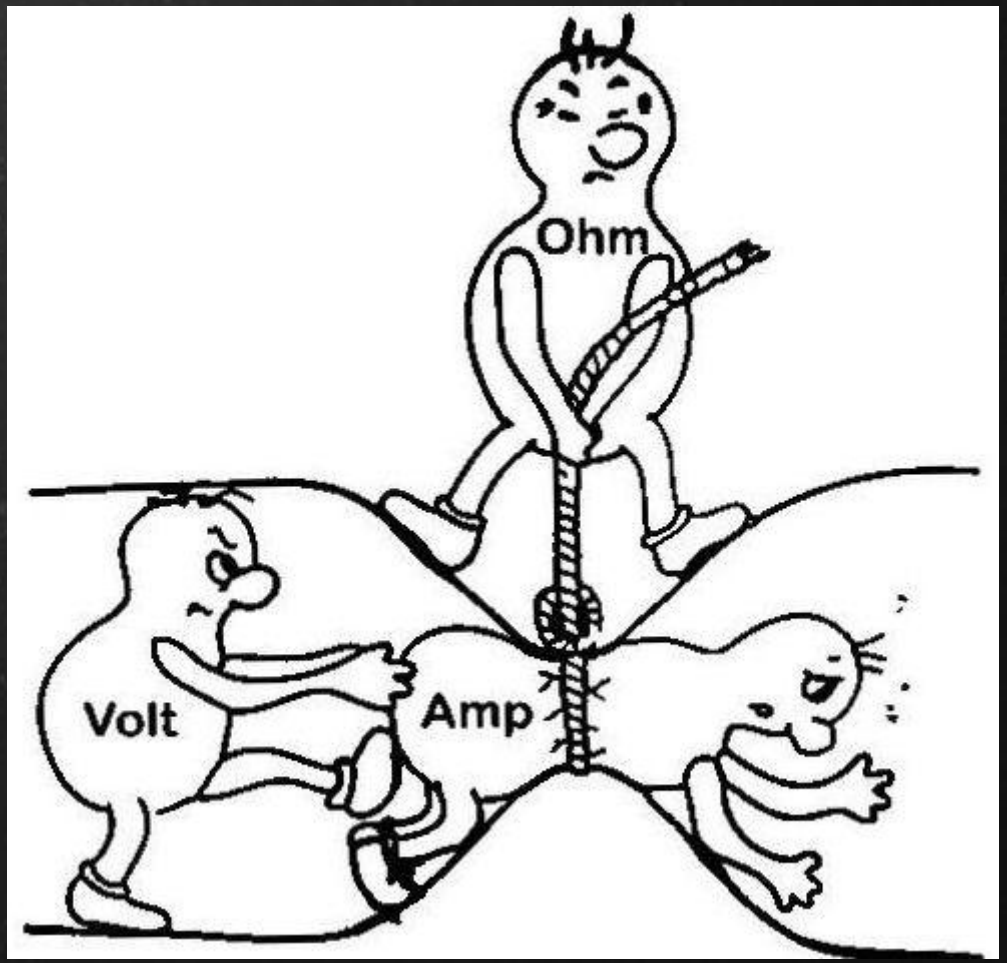




水流类比

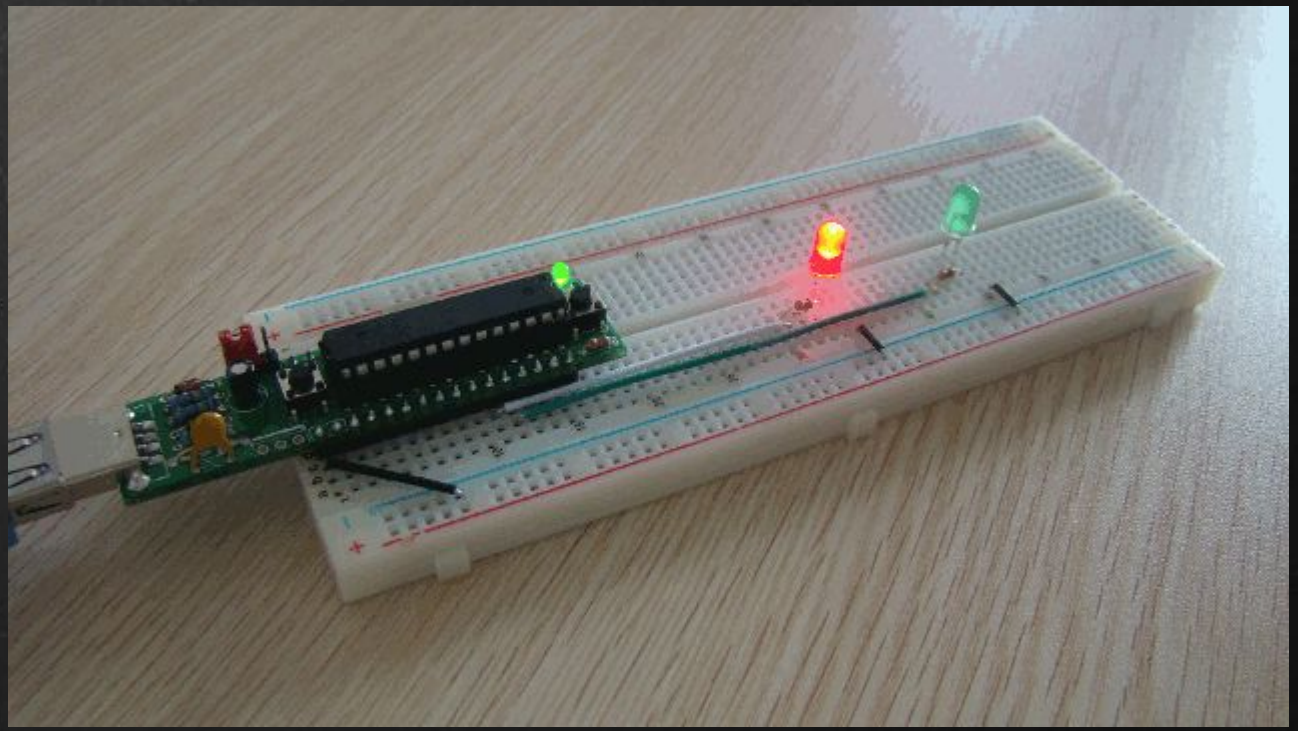
推荐网页 https://learn.sparkfun.com/tutorials/voltage-current-resistance-and-ohms-law

# Ohm’s Law

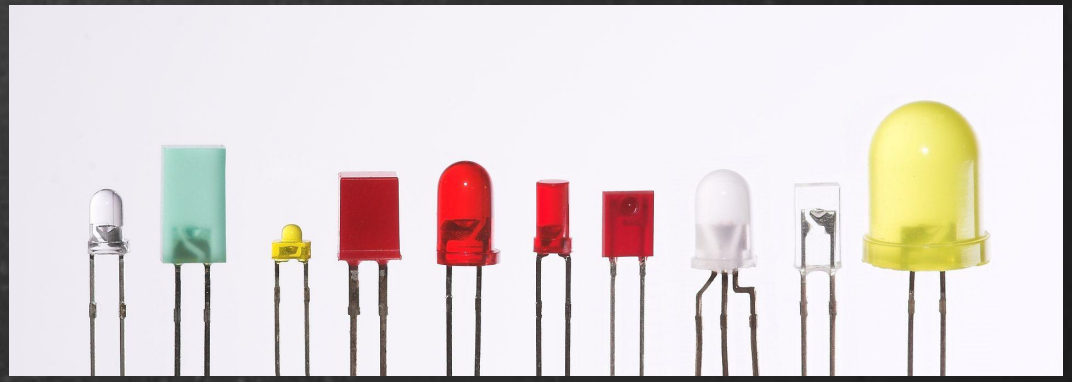


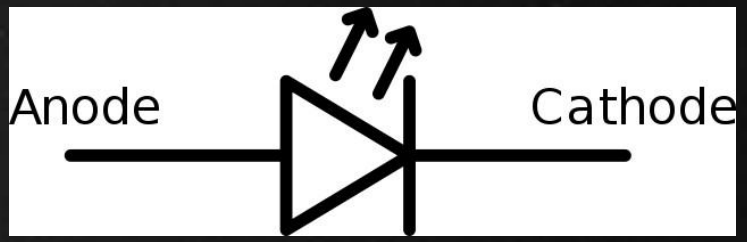

Note: 电子学知识会在后续课程中逐步介绍

# Let There Be Blink



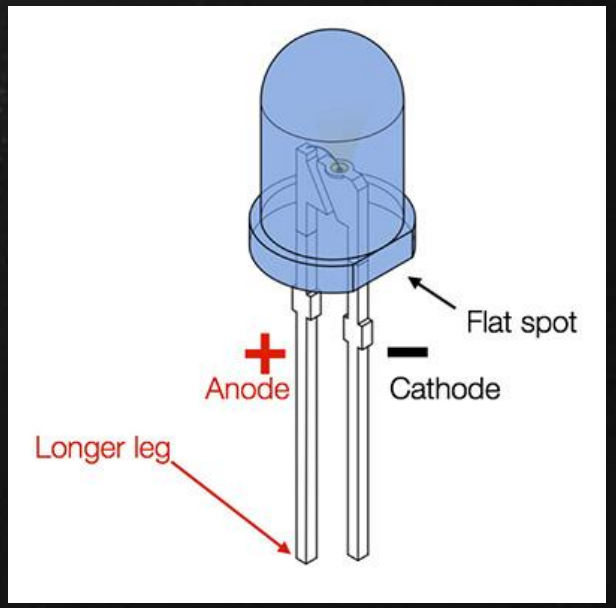
# LED: Light-Emitting Diode





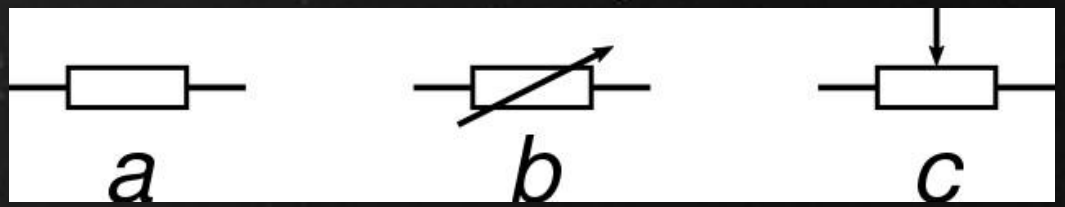

LED 在电路中的符号

Light-emitting diode - Wikipedia

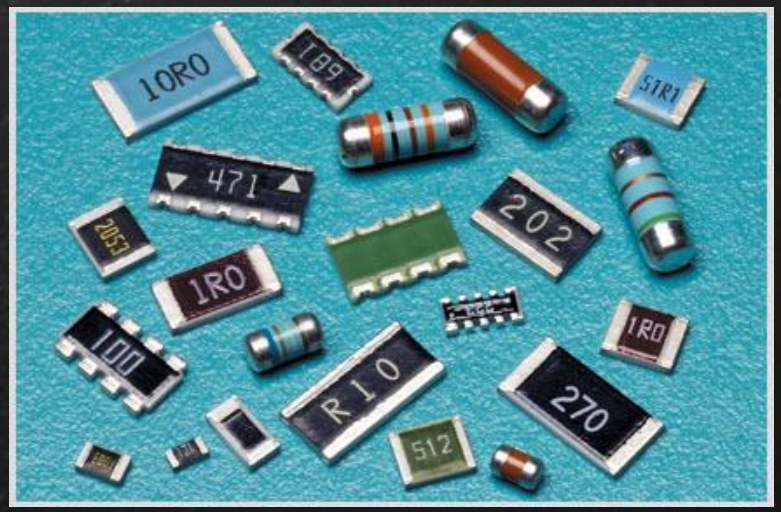


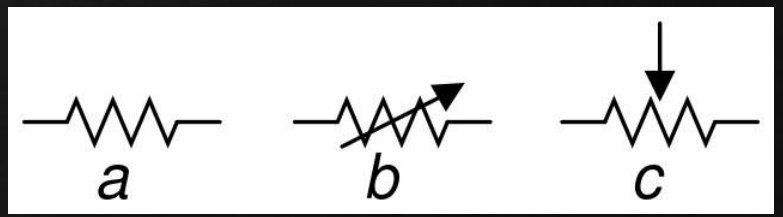

LED 的极性，长正短负

# 电阻器 Resistors



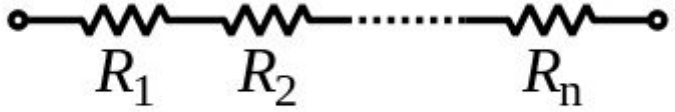

电阻的欧式符号



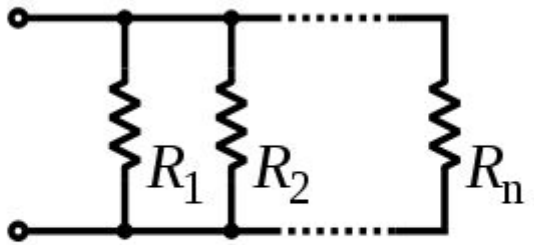




电阻的美式符号

# 电阻器 - 串联和并联

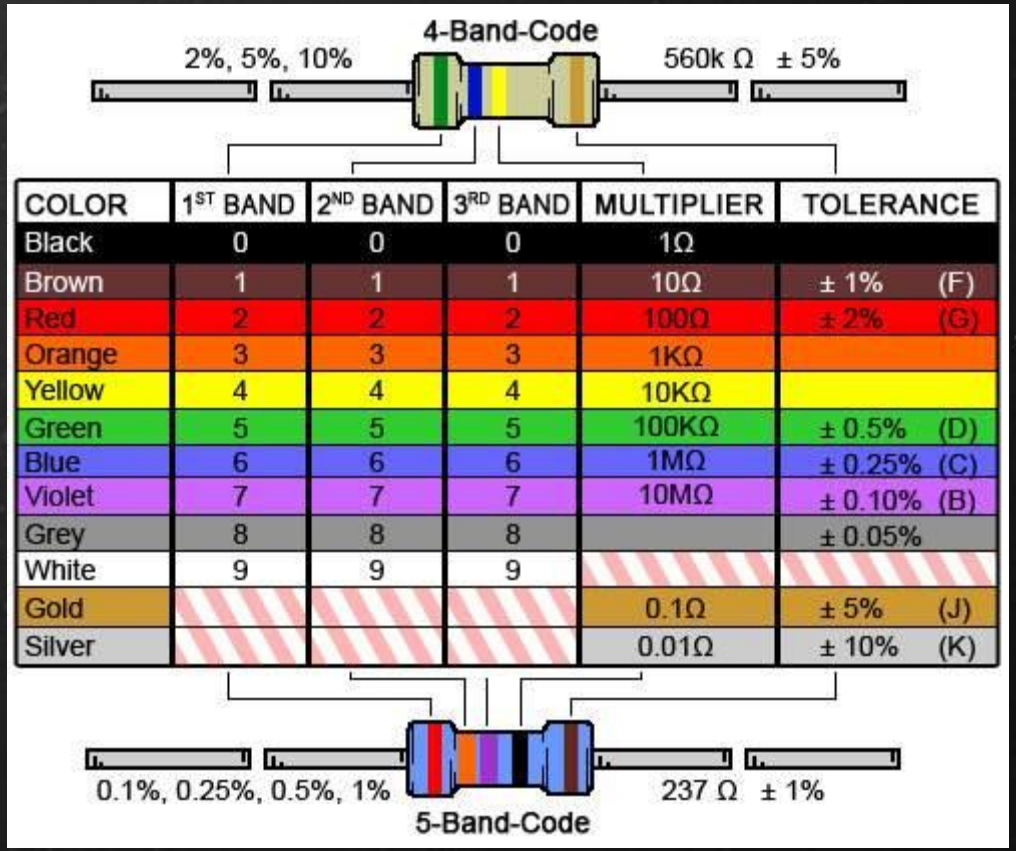



$$R_{eq} = R_1 + R_2 + \cdots + R_n.$$



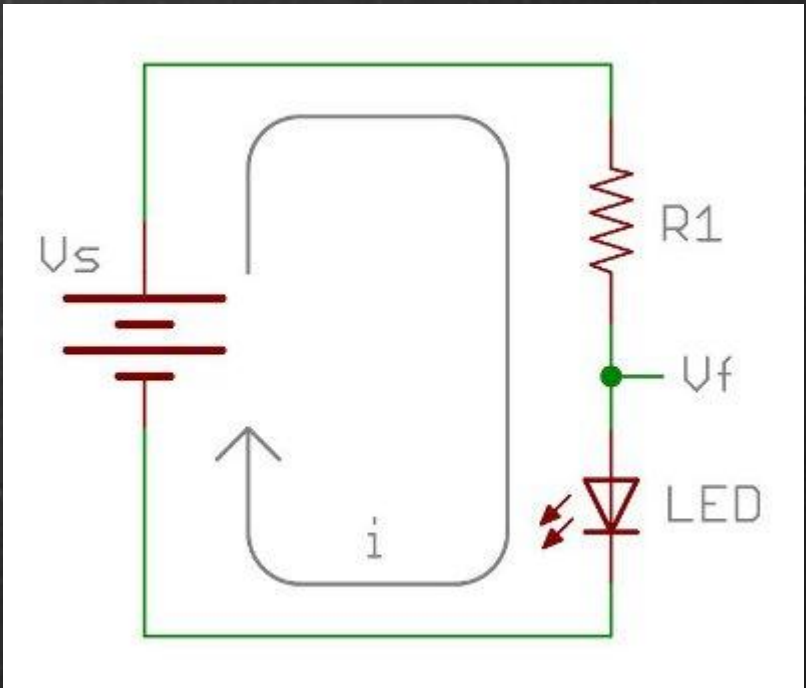

$$\frac{1}{R_{eq}} = \frac{1}{R_1} + \frac{1}{R_2} + \cdots + \frac{1}{R_n}.$$

# How to Read a Resistor




| COLOR | 1ST BAND | 2ND BAND | 3RD BAND | MULTIPLIER | TOLERANCE |
|---|---|---|---|---|---|
| Black | 0 | 0 | 0 | 1Ω | |
| Brown | 1 | 1 | 1 | 10Ω | ± 1% (F) |
| Red | 2 | 2 | 2 | 100Ω | ± 2% (G) |
| Orange | 3 | 3 | 3 | 1KΩ | |
| Yellow | 4 | 4 | 4 | 10KΩ | |
| Green | 5 | 5 | 5 | 100KΩ | ± 0.5% (D) |
| Blue | 6 | 6 | 6 | 1MΩ | ± 0.25% (C) |
| Violet | 7 | 7 | 7 | 10MΩ | ± 0.10% (B) |
| Grey | 8 | 8 | 8 | | ± 0.05% |
| White | 9 | 9 | 9 | | |
| Gold | | | | 0.1Ω | ± 5% (J) |
| Silver | | | | 0.01Ω | ± 10% (K) |

# Let’s do some Math



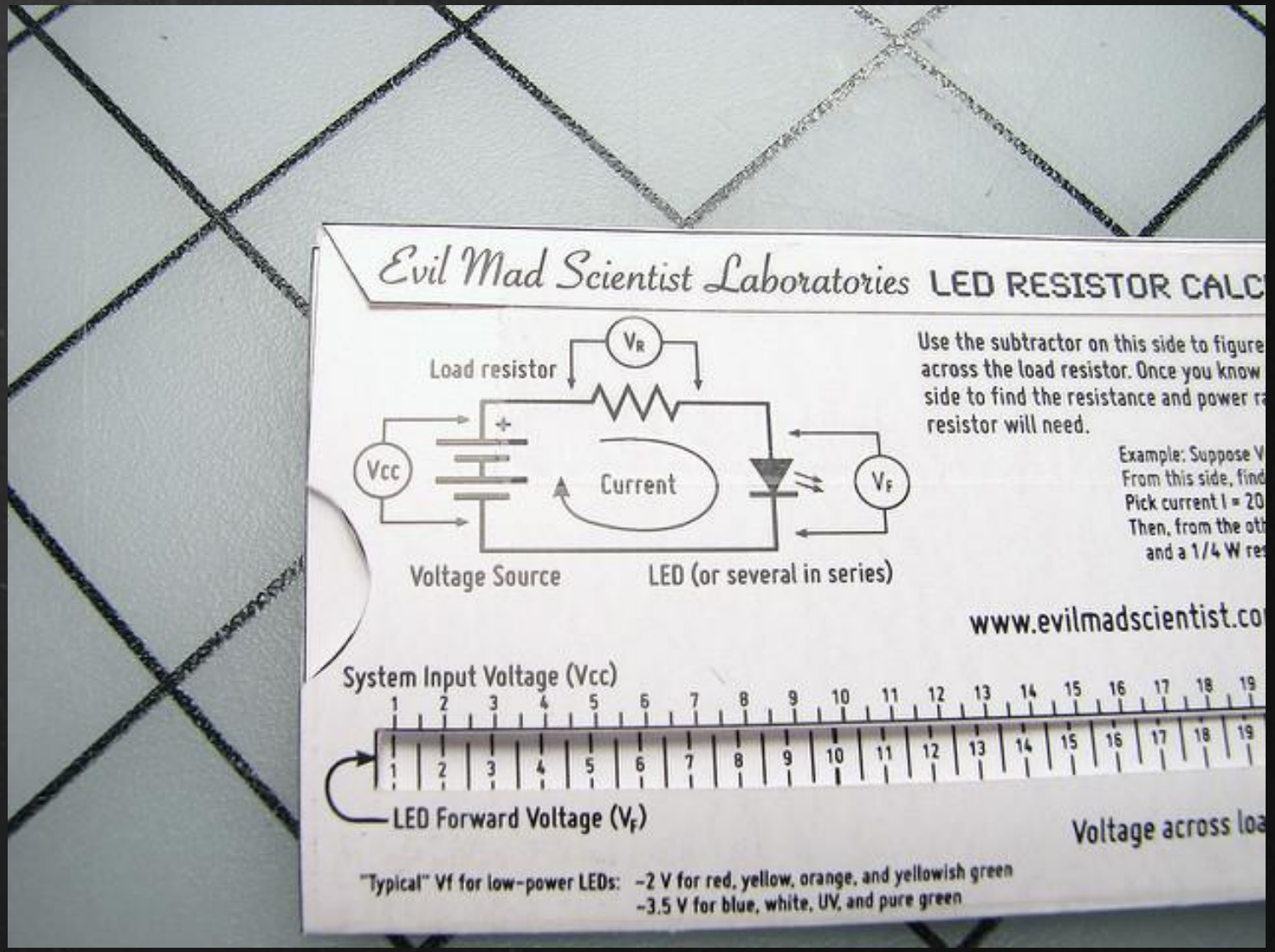

$$I = \frac{Vs - Vf}{R1}$$

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

Try it:

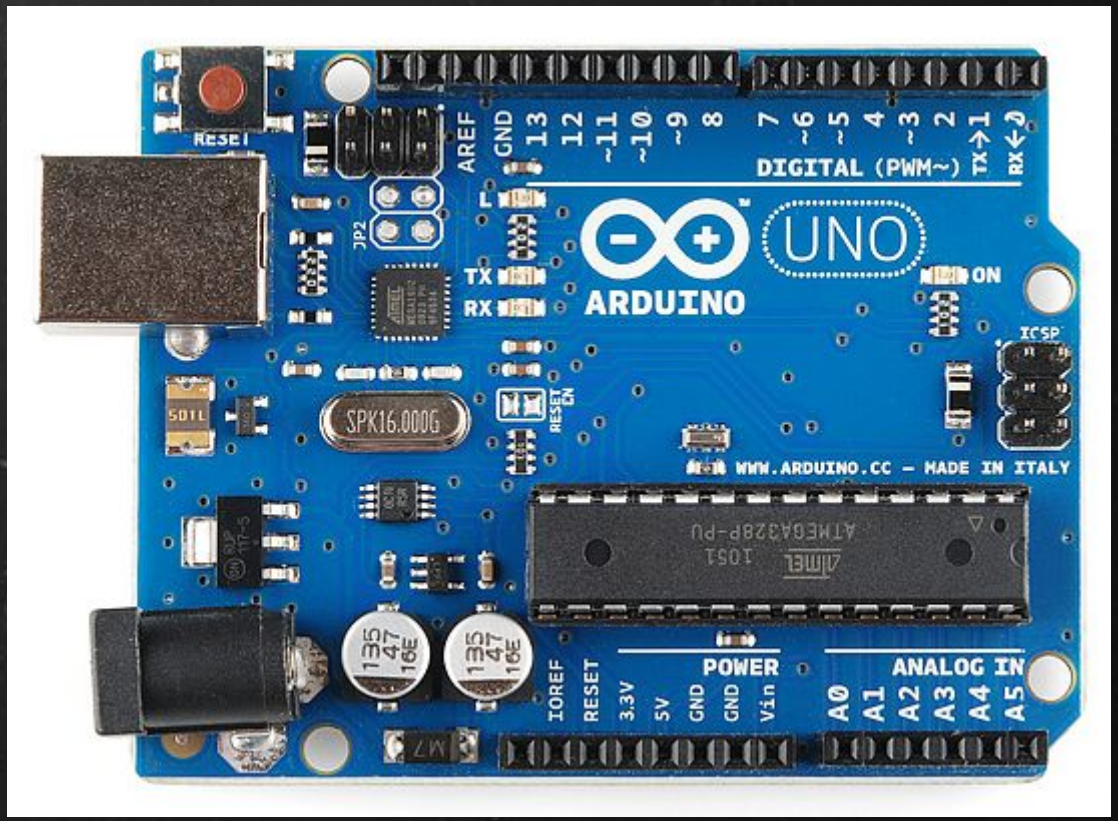
- 用万用表测量 LED 两边的压降
- 更换 R1 的值观察 LED 的亮度
- 不同颜色的 LED 有不同的 Vf, find it out.

# Let’s do some Math

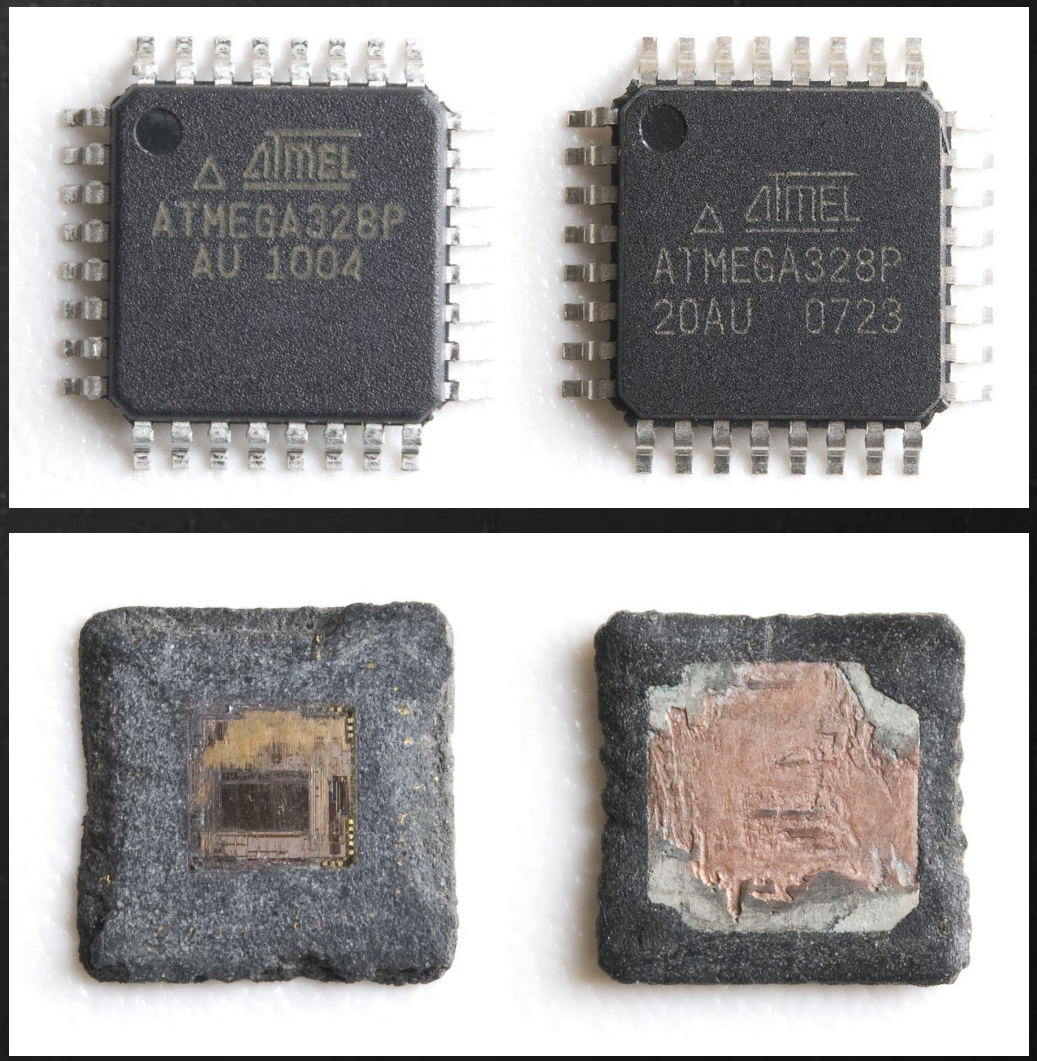



来源 Wallet-size LED Resistance Calculator (Evil Mad Scientist)

# Arduino: A mini computer

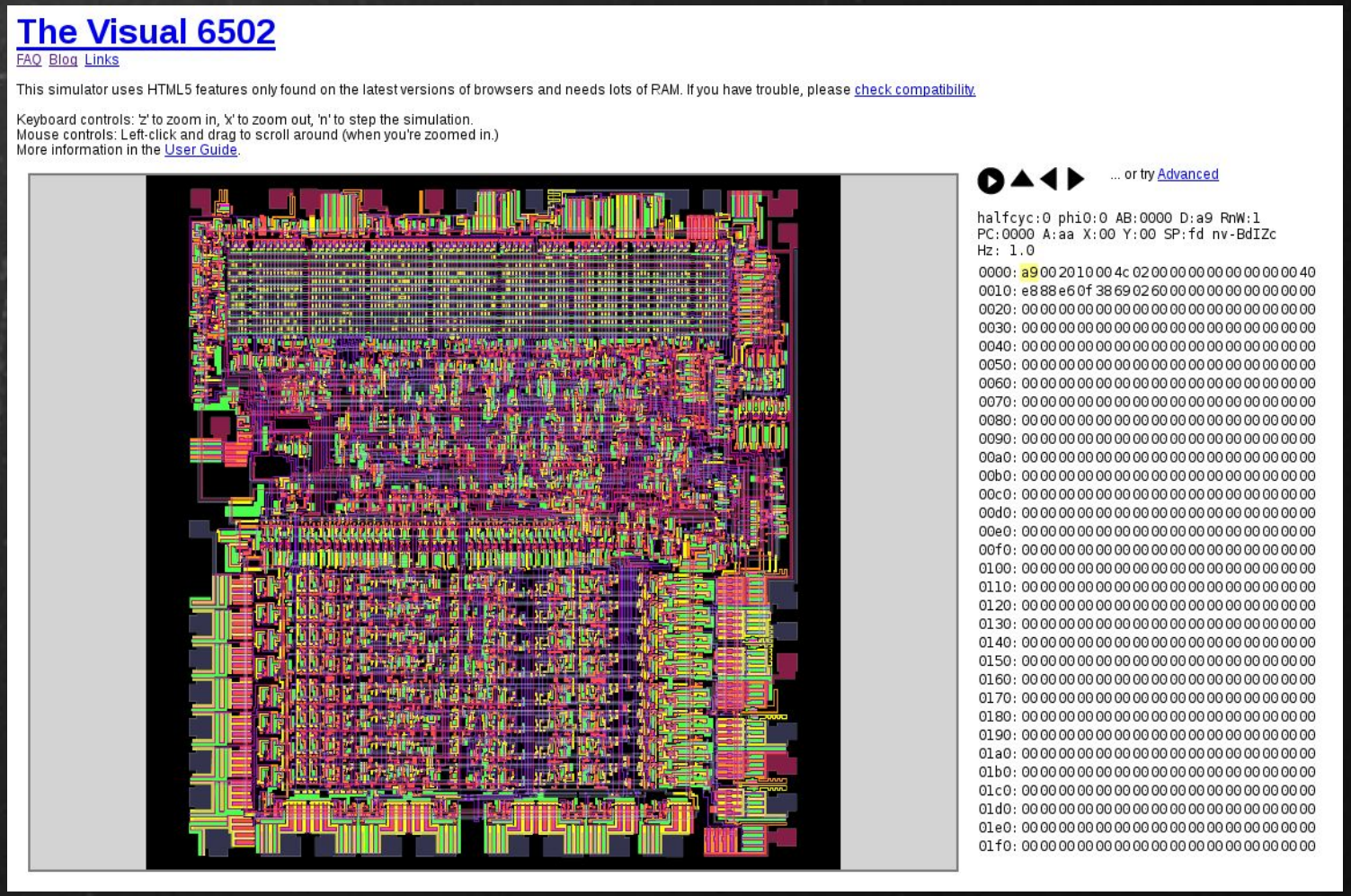


# 微控制器(1)




照片取自 Revisiting the Counterfeit ATMega328s (Sparkfun)

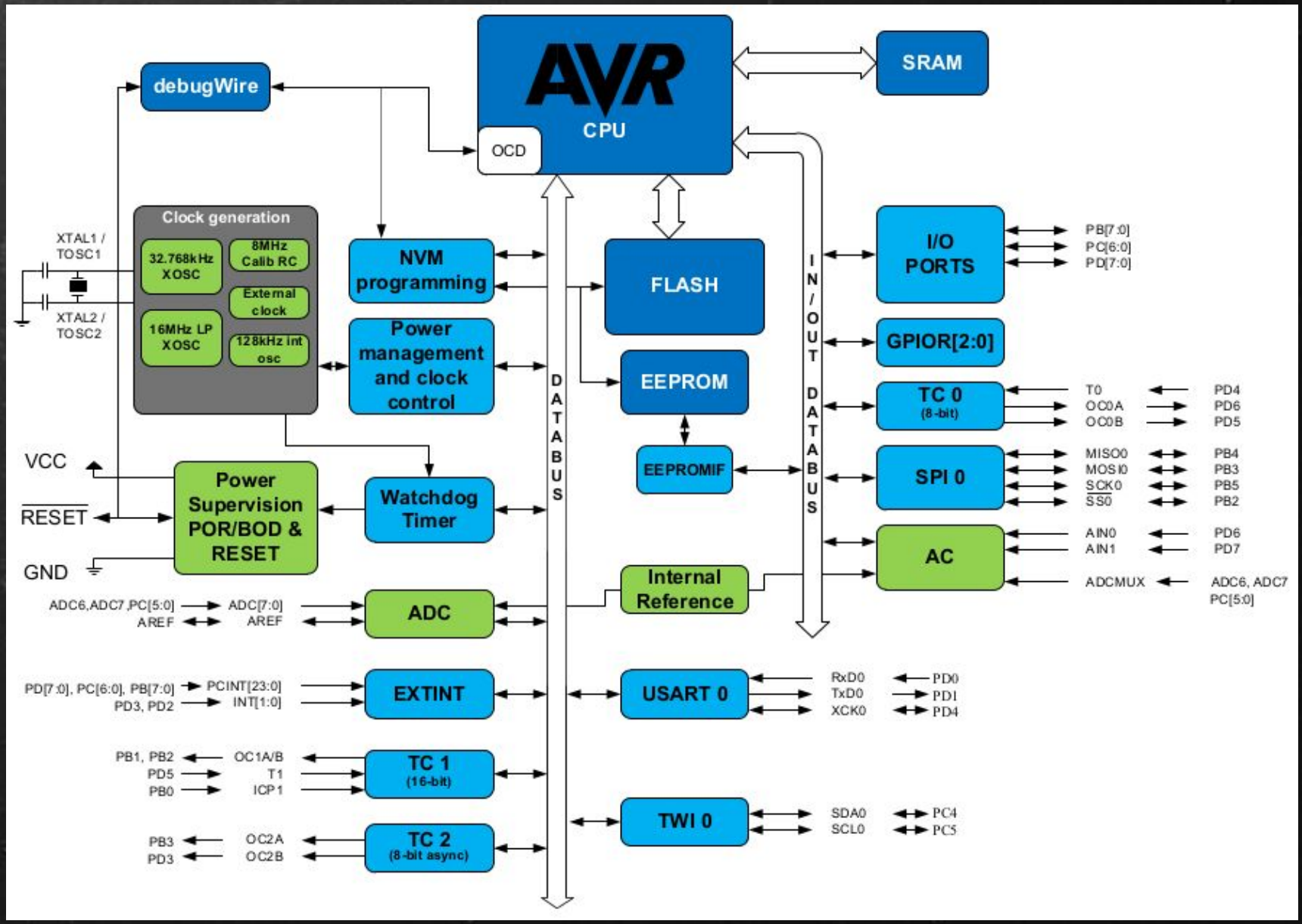
# 微控制器 - Off Topic




Super cool project

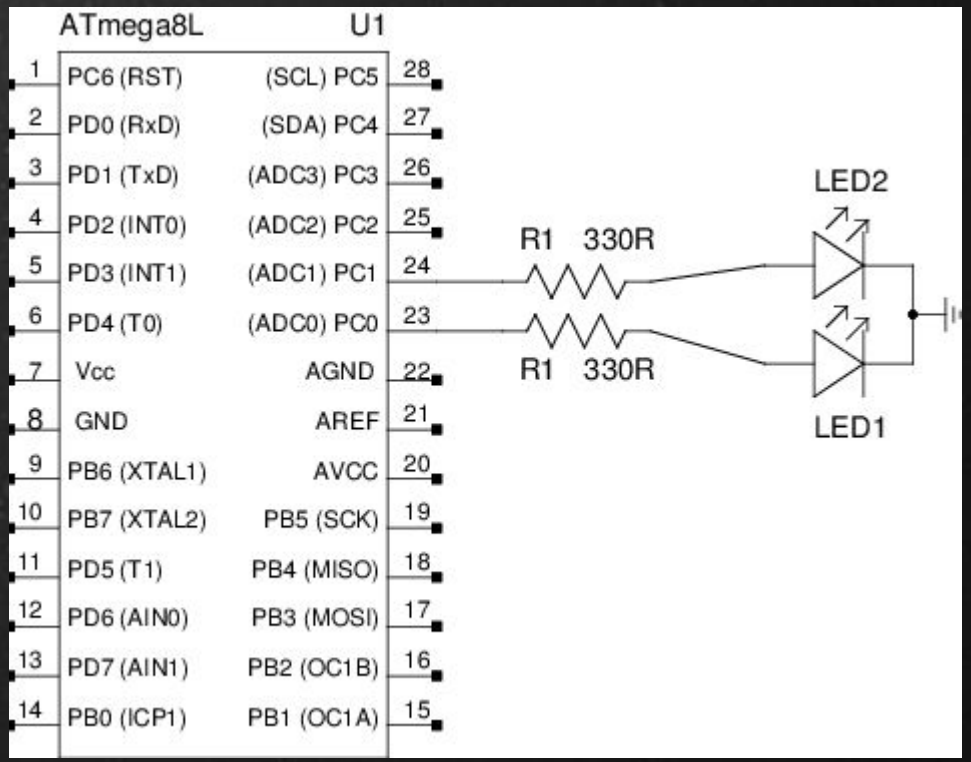
Visual6502 http://www.visual6502.org/JSSim/index.html

# 微控制器-内部框图

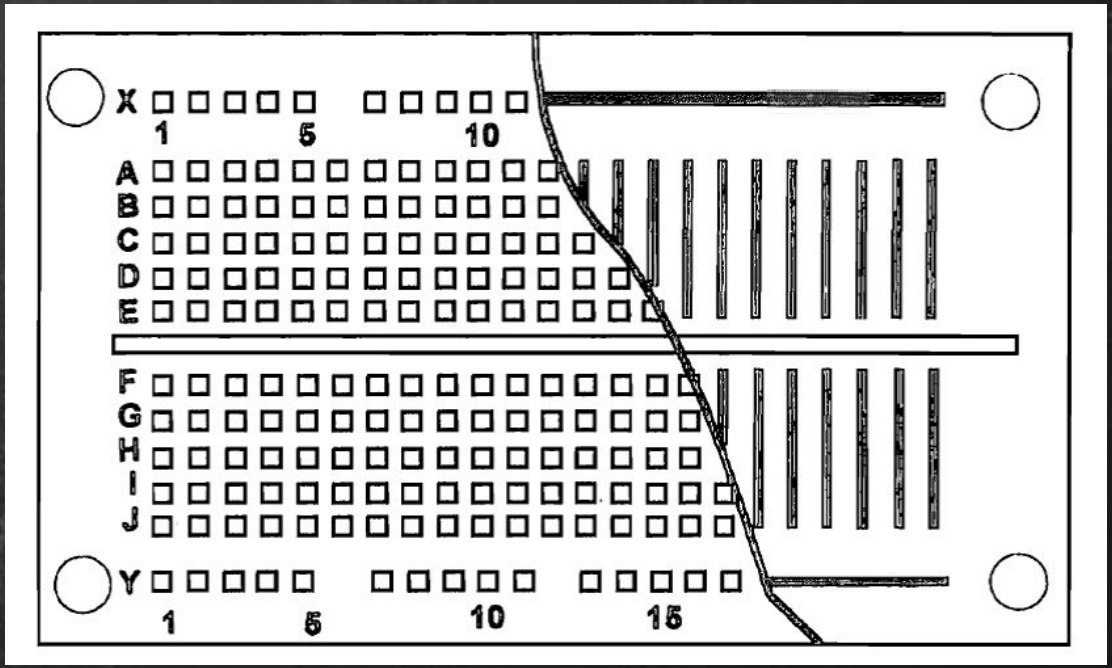



更多细节请见 Atmega328 的 datasheet

# 接线


ATmega8L
U1
1 PC6 (RST)
2 PD0 (RxD)
3 PD1 (TxD)
4 PD2 (INT0)
5 PD3 (INT1)
6 PD4 (T0)
7 Vcc
8 GND
9 PB6 (XTAL1)
10 PB7 (XTAL2)
11 PD5 (T1)
12 PD6 (AIN0)
13 PD7 (AIN1)
14 PB0 (ICP1)
28 (SCL) PC5
27 (SDA) PC4
26 (ADC3) PC3
25 (ADC2) PC2
24 (ADC1) PC1
23 (ADC0) PC0
22 AGND
21 AREF
20 AVCC
19 PB5 (SCK)
18 PB4 (MISO)
17 PB3 (MOSI)
16 PB2 (OC1B)
15 PB1 (OC1A)
R1 330R
R1 330R
LED2
LED1

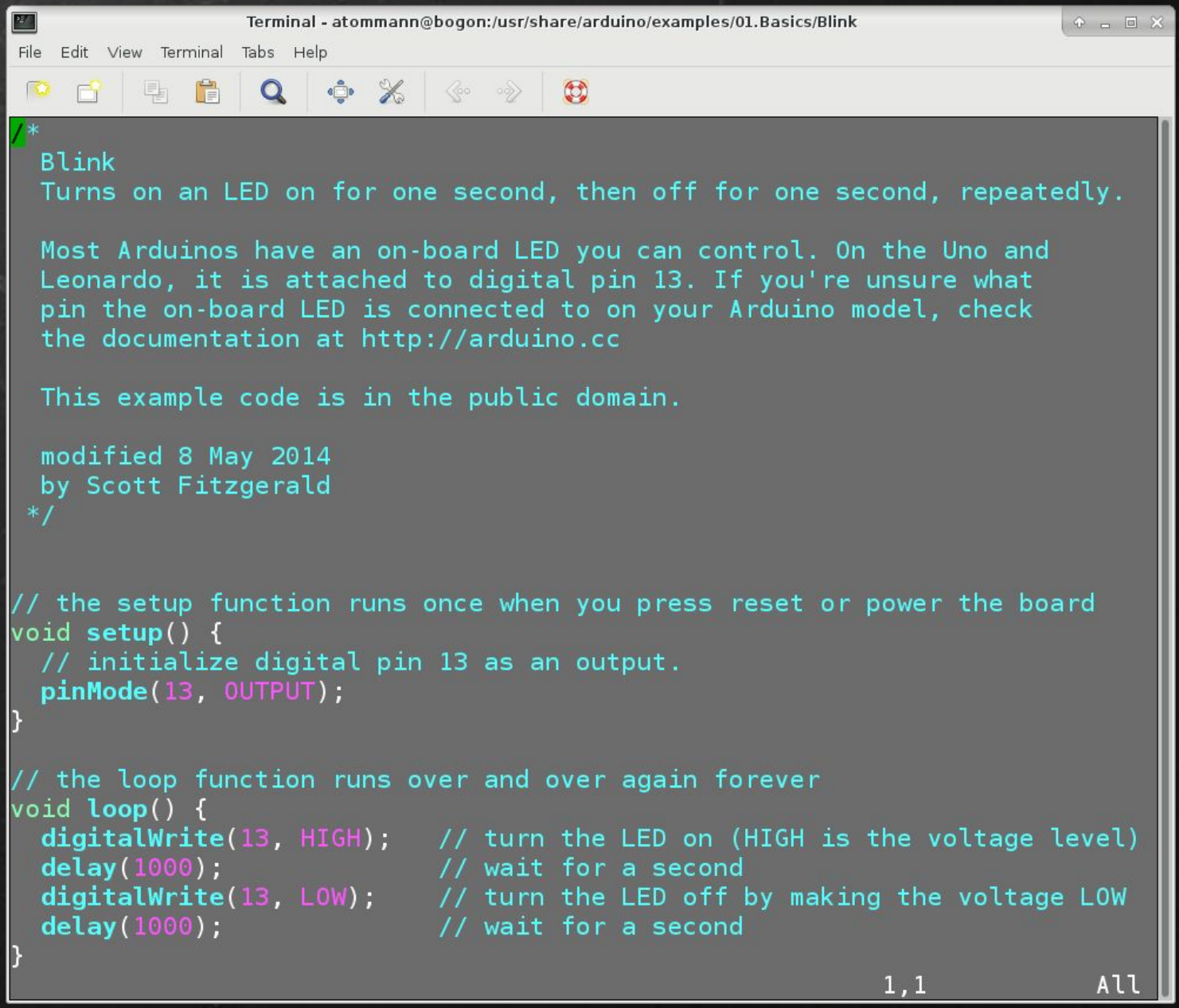
# 面包板



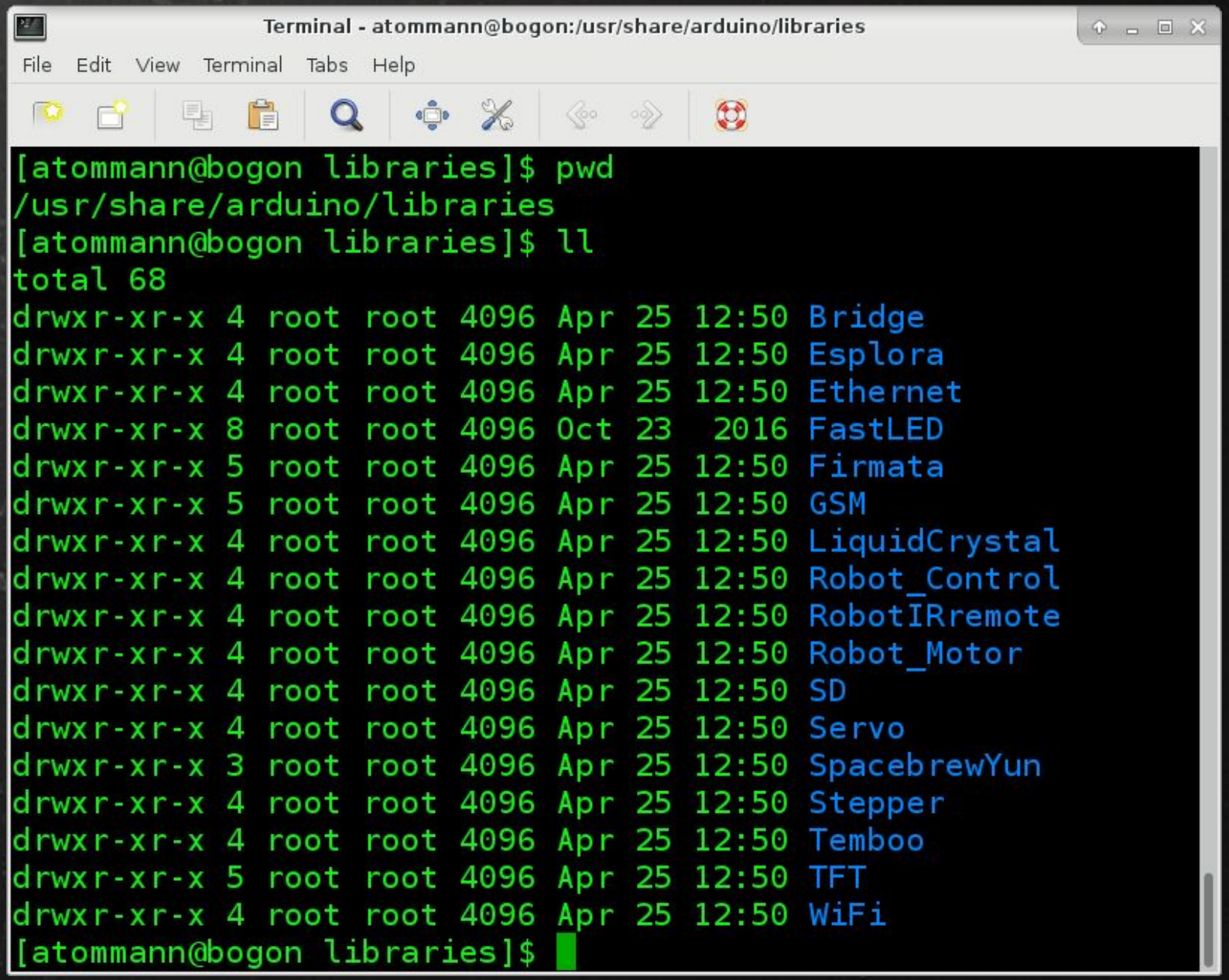

面包板的内部结构

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。

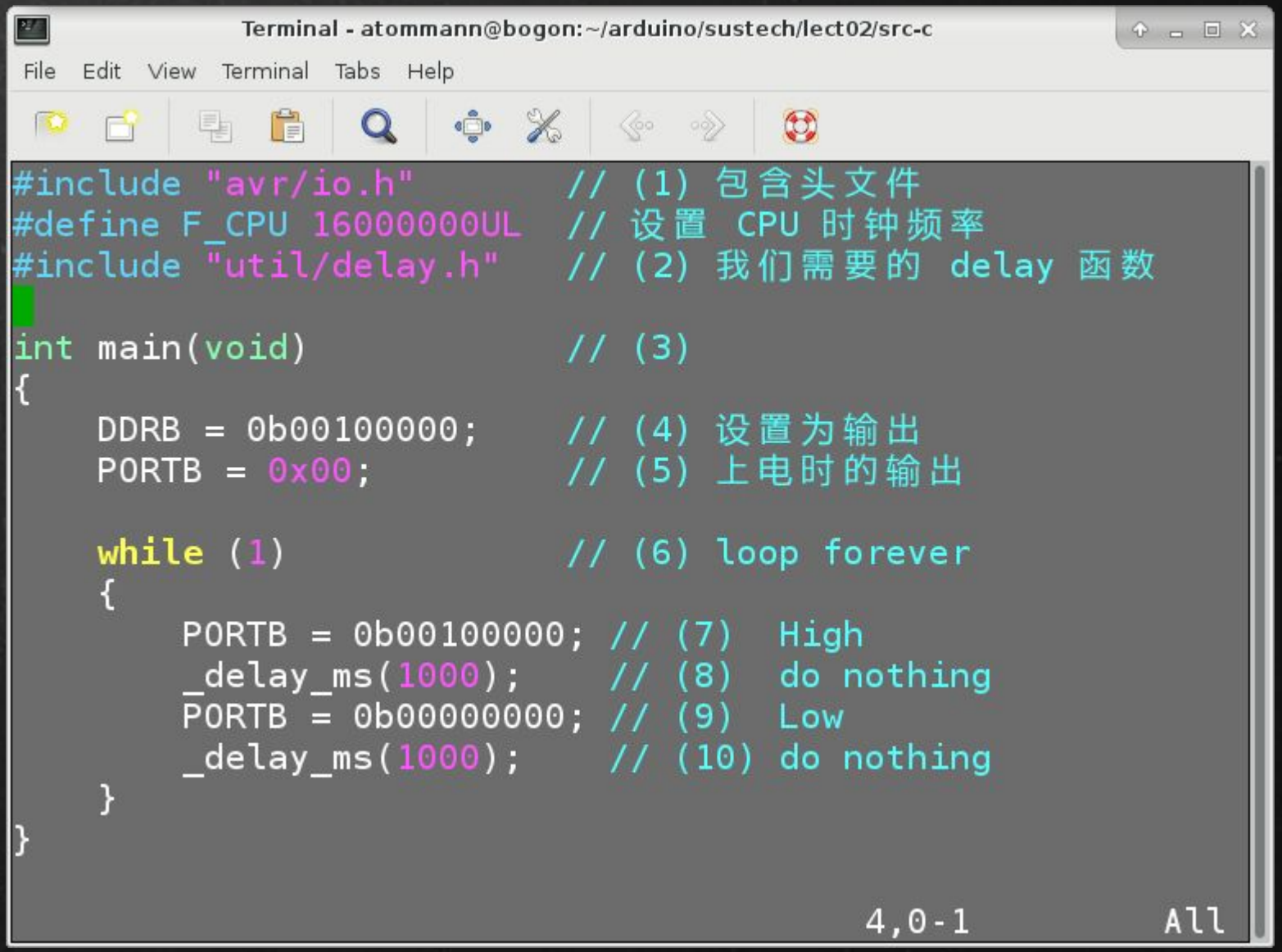
# 用软件控制硬件 - The Arduino way



# Arduino 的优势 - 软件库



# 用软件控制硬件 - 中级语言 C



# 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

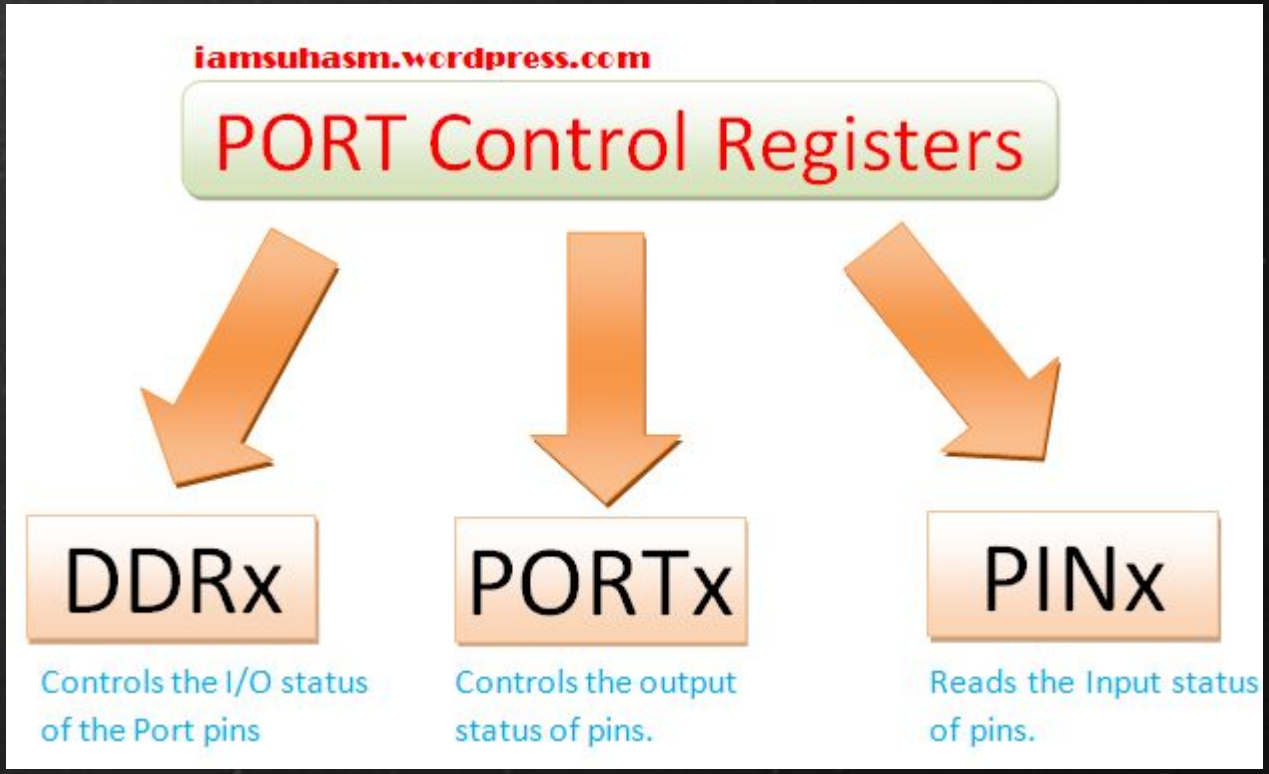
```
.set PINB,      0x03
.set DDRB,      0x04
.set TCCR0B,    0x25
.set TCNT0,     0x26
.set LED_MASK, 0b00100000
.set PS_1024,  0b00000101

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

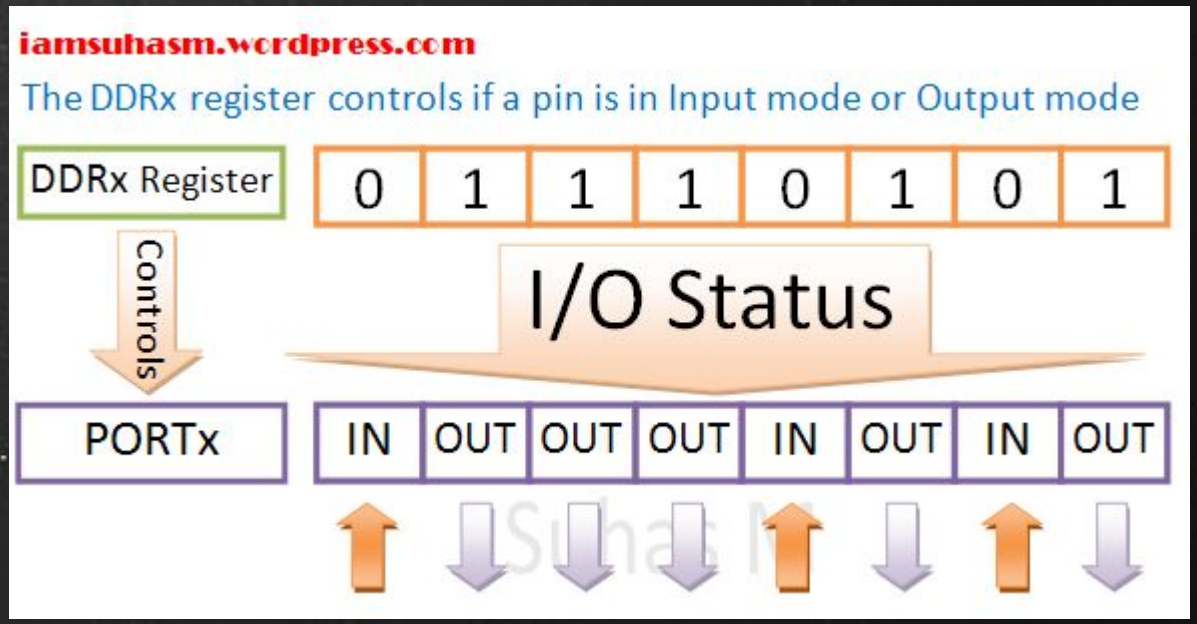
| Address | Opcode | Mnemonic | Description |
|---|---|---|---|
| 0000 | 7A | REQ | Reset ‘Q’ = 0 (turns ‘Q’ LED off) |
| 0001 | F8 | LDI | Load data (10) into ‘D’ register |
| 0002 | 10 | (data) | *data for preceding instruction* |
| 0003 | B1 | PHI | Set high order byte of register R1 = ‘D’ (10) |
| 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’ ≠ 0, short branch to address 00***04*** |
| 0007 | 04 | (data) | *data (low order address) for preceding instruction* |
| 0008 | 31 | BQ | If ‘Q’ = 1, short branch to address 00***00*** |
| 0009 | 00 | (data) | *data (low order address) for preceding instruction* |
| 000A | 7B | SEQ | Set ‘Q’ = 1 (turns ‘Q’ LED on) |
| 000B | 30 | BR | Unconditional short branch to address 00***01*** |
| 000C | 01 | (data) | *data (low order address) for preceding instruction* |

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

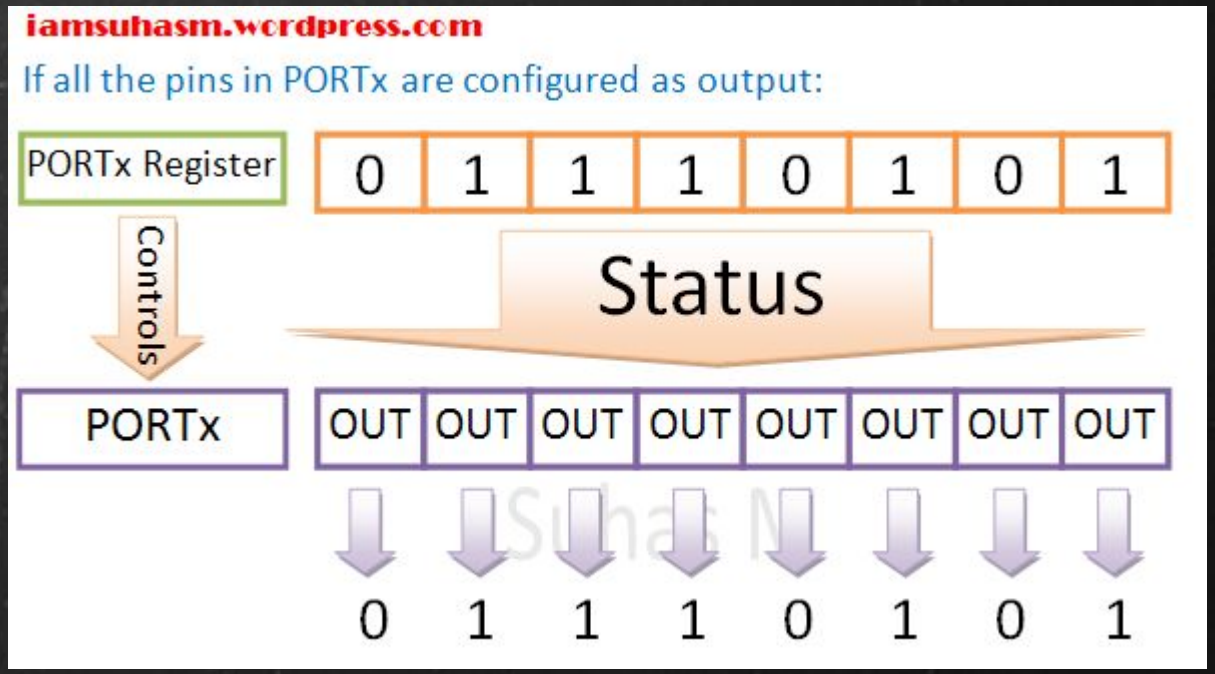
# 到底发生了什么？- Registers



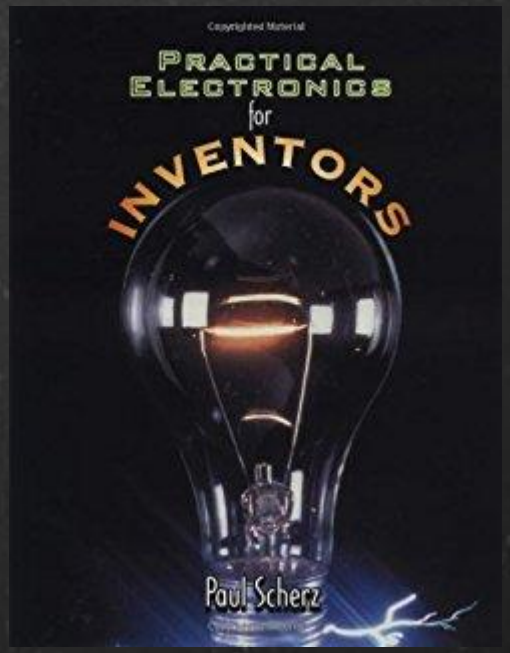
# 到底发生了什么？控制输入输出



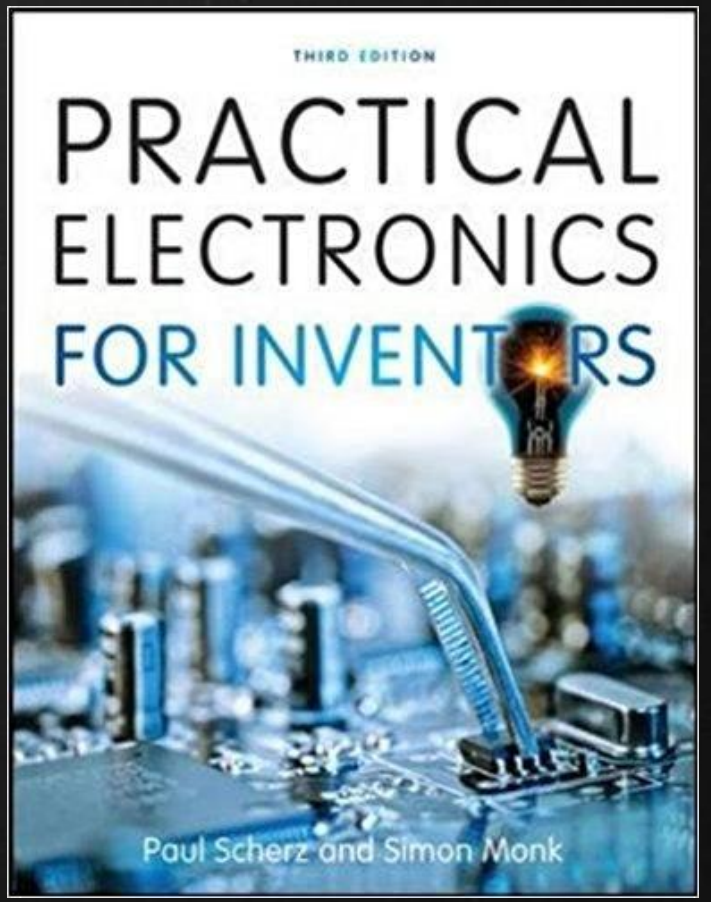
# 到底发生了什么？- 端口数据 Register



# 推荐的电子学图书(1)

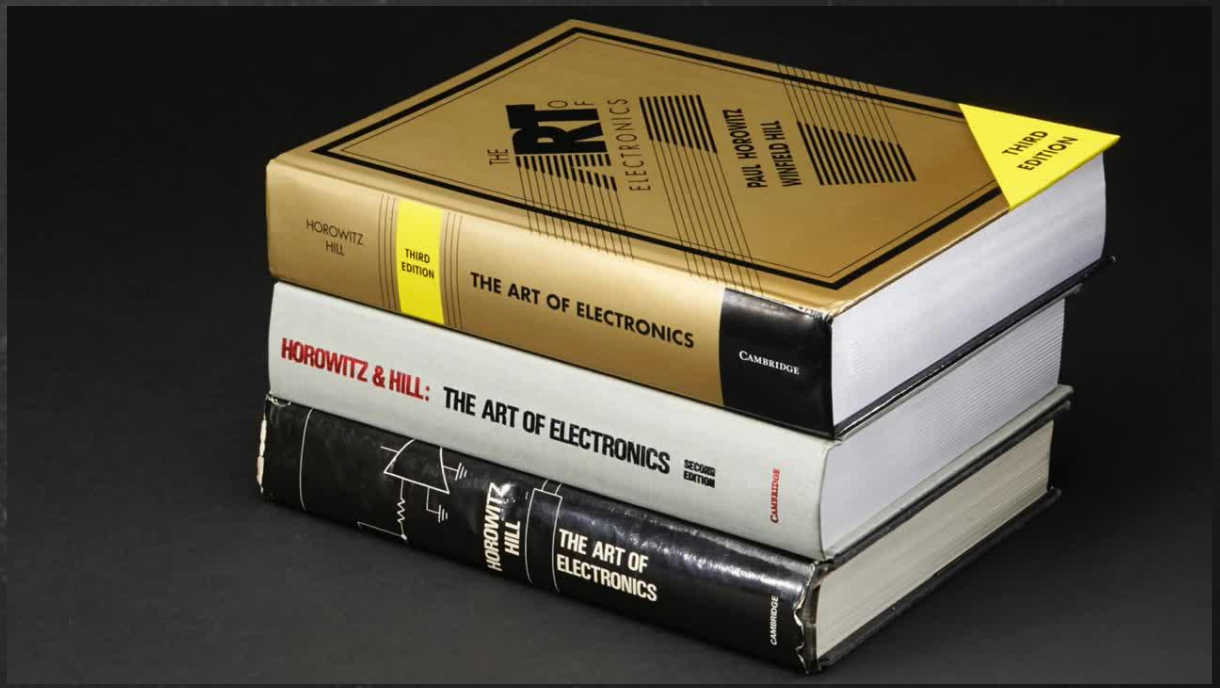



《发明者电子设计宝典》第二版
《实用电子元器件与电路基础》(第3版)



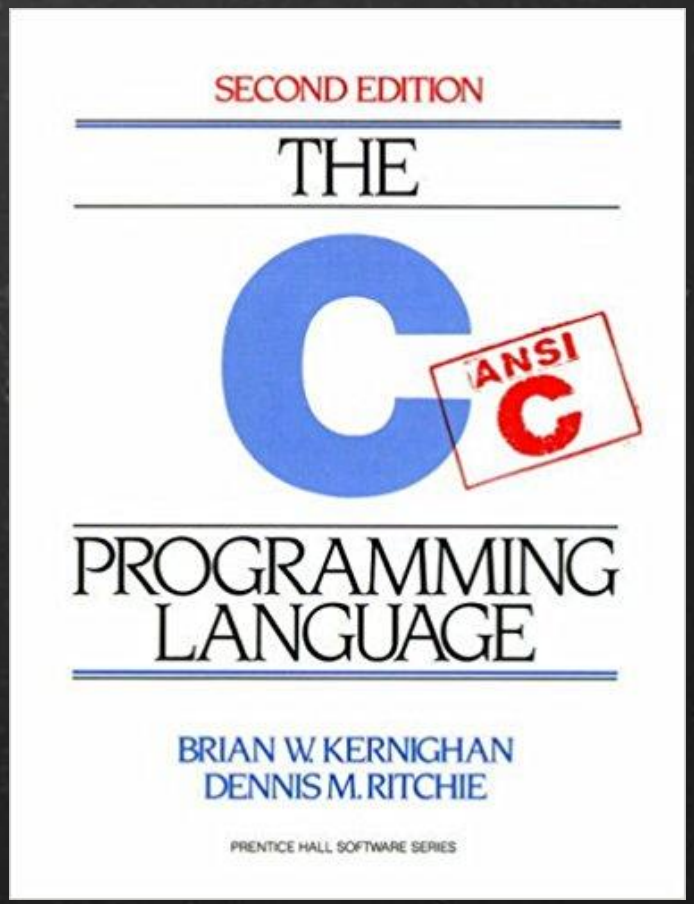

Practical Electronics for Inventors
最新:英文第 4 版
特点:手绘电路，直观讲解，电路有参数，水类比，易懂

# 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 Bibble of Electronics
更多 EE book https://blog.adafruit.com/category/ee_bookshelf/
但是，别忘记：请多多实践。

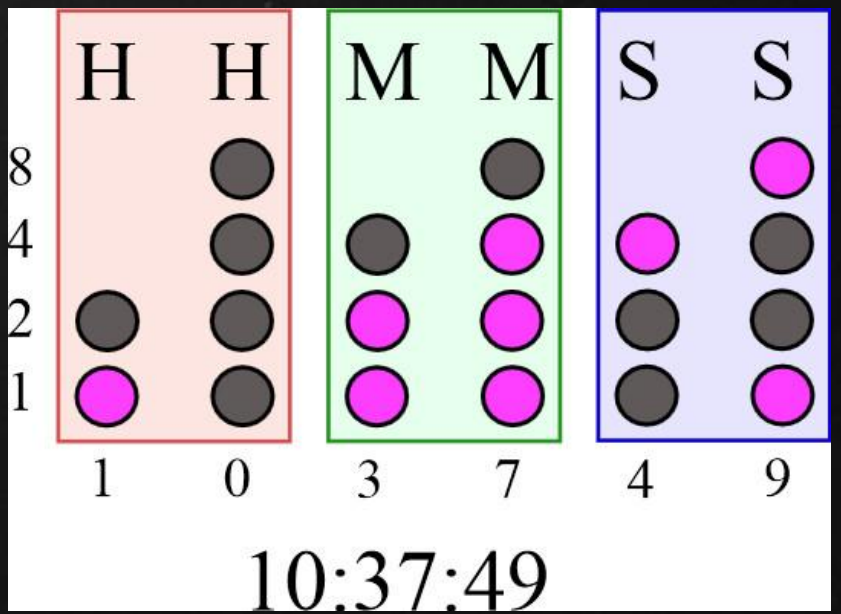
# 推荐的 C 语言书


SECOND EDITION
THE
C
ANSI
C
PROGRAMMING
LANGUAGE
BRIAN W. KERNIGHAN
DENNIS M. RITCHIE
PRENTICE HALL SOFTWARE SERIES


近10年来最好的一部C语言著作
C Programming:
A Modern Approach, Second Edition
C语言程序设计
现代方法
第2版
[美] K. N. King 著
人民邮电出版社

# 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 appriciate 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。