



4th January 4, 2020

本次课程为复习课，学生们复习了 Arduino 的基础知识和之前学过的几个案例代码。并且学了新的语法 map。代码如下，代码范例已经分享在微信群，可以直接打开。

```
int ledPin = 9;    // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }

  // fade out from max to min in increments of 5 points:
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
  }
}
```

作业:

复习已经学过的各个项目，尝试 IDE 内置的 sample.

使用 LED 和任意一个其它的传感器，练习自己连接 Arduino 并且运行 map 功能。

Arduino 基础知识复习

什么是 Arduino

Arduino 是一张微控制器板子 (microcontroller board)，约莫一个手掌大：

用户可以在 Arduino 板子上接上各种电子装置，例如 LED 灯、喇叭、马达、开关、温湿度传感器、红外线发射与接收器、LCD 显示设备，以及 Ethernet, WiFi, XBee, Bluetooth, RFID, GPS 等各种通讯模块。若再配合撰写一些自动控制的程序，就能利用 Arduino 做出各式各样的自动控制应用，例如利用温度传感器控制风扇的运转、使用可变电阻控制灯光的明暗、控制马达的转速、利用红外线遥控家电 / 利用伺服机 (Servo) 控制机械手臂或机器人，以及制作自走车、飞行器等等。

Arduino 的特色

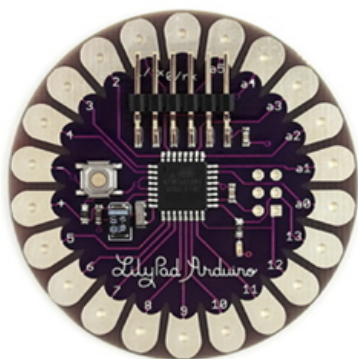
简单来说，Arduino 有下列这些特色：

1. 开放源码。不仅软件是开放源码，连硬件也是开放的。开发软件用的 IDE 可免费下载，Arduino 的电路设计图也可以从网络上下载。
2. 简单好用资源多。传统上，要开发微控制器的程序，开发者需要具备电子电机相关科系的背景，一般人不容易进入这个世界。Arduino 进入门槛低，即便没有电子电机相关科系的背景，也可以很容易学会使用 Arduino。再者，由于 Arduino 开放的精神，很多人都乐于分享他们的作品，所以网络上有非常多的资源。很多时候，我们只要参考网友的作品，配合自己的需求调整一下设计，就可以在短时间内完成自己的作品。
3. 物美价廉。一张微控制器板子动辄上千元，相较于这类微控制板子，Arduino 控制板只要 30 美元左右，可以说是非常廉价。

Arduino 的应用

Arduino 的应用非常多，不胜枚举，底下只举几个我所知的应用：

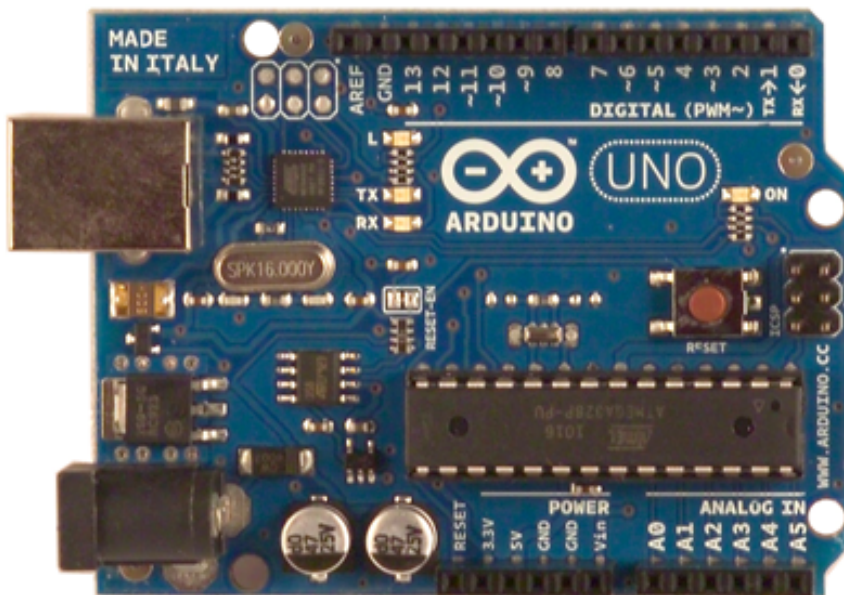
[LilyPad](#) - 可穿戴在身上的 Arduino，这东西应该可以拿来作钢铁人的发光手套。





Arduino 硬件规格

底下说明 2010 年最新的板子 Arduino UNO 的硬件规格。 Arduino UNO 的外观如下：



硬件规格：

微控制器	ATmega328
工作电压	5V
输入电压(建议)	7-12V

输入电压(限制)	6-20V
• I/O 引脚	14 支(其中有 6 支脚位可提供 PWM 输出)
类入 Input Pins	6 支
I/O 引脚直流电流	40mA
3.3V 引脚直流电流	50 马
Flash 内存	32KB, 其中 0.5KB 拿去给 bootloader 使用
Sram	2KB
Eeprom	1KB
频率	16MHz

• I/O 引脚:

14 支数字 I/O Pins 可以当作 input 使用, 也可以当作 output 使用, 使用方法是透过 [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) 这几个函式。这 14 支数字 I/O Pins, 其中几支脚有特殊的功能:

Serial 通讯	0(RX) 和 1 (TX) 这两支脚。用来接收(RX)与传输(TX) TTL 讯号的序列数据。这两支脚也连接到 USB Converter 芯片中。
外部中断	2 和 3 这两支脚。这两支脚可以利用外部事件触发中断。详细内容请参考 attachInterrupt() 函式。
Pwm	3, 5, 6, 9, 10 和 11 共六支脚。透过 analogWrite() 函式可以提供 8-bit 的 PWM 输出。
Spi	10 (SS), 11 (MOSI), 12 (MISO) 和 13 (SCK) 这四支脚。这四支脚搭配 SPI Library 可提供 SPI 序列通讯。
导致	13。内建一颗 LED, 当 pin 脚为 HIGH 时, LED 打开, 当 pin 脚为 LOW 时, LED 关闭。

模拟输入 Pins:

Arduino Uno 有 6 支模拟输入脚, 标记为 A0 到 A5, 每支脚都可提供 10 位的解析 (即 1024 种不同的数值)。这些脚位所用的参考电压默认为 0 到 5V, 不过参考电压也是可以更改的, 方法是透过 AREF 脚和 [analogReference\(\)](#) 函式。

另外, 有几支脚也有特殊功能:

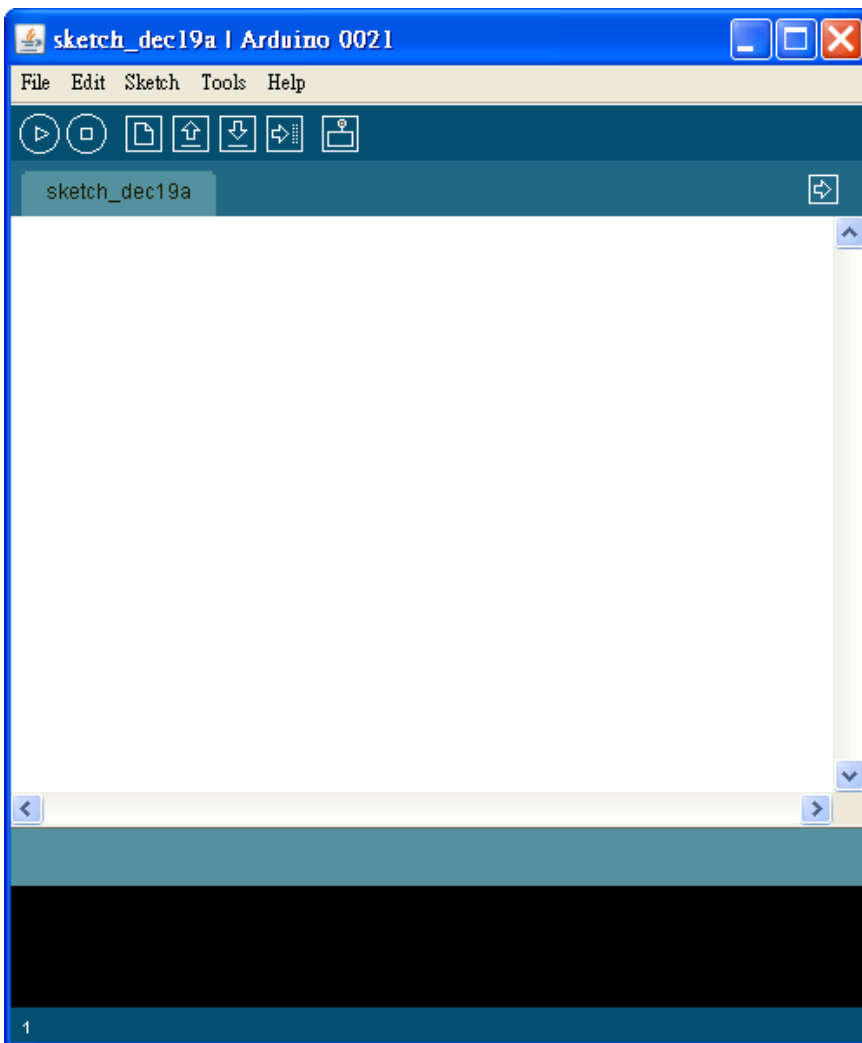
I2c	4 (SDA) 和 5 (SCL) 这两支脚。透过 Wire library 可以提供 I2C 通讯。
-----	---

其它:

AREF	模拟输入的参考电压，搭配 analogReference() 函式一起使用。
重置	当 Reset 脚为 LOW 时，微控制器会重置。

Arduino 软件开发环境

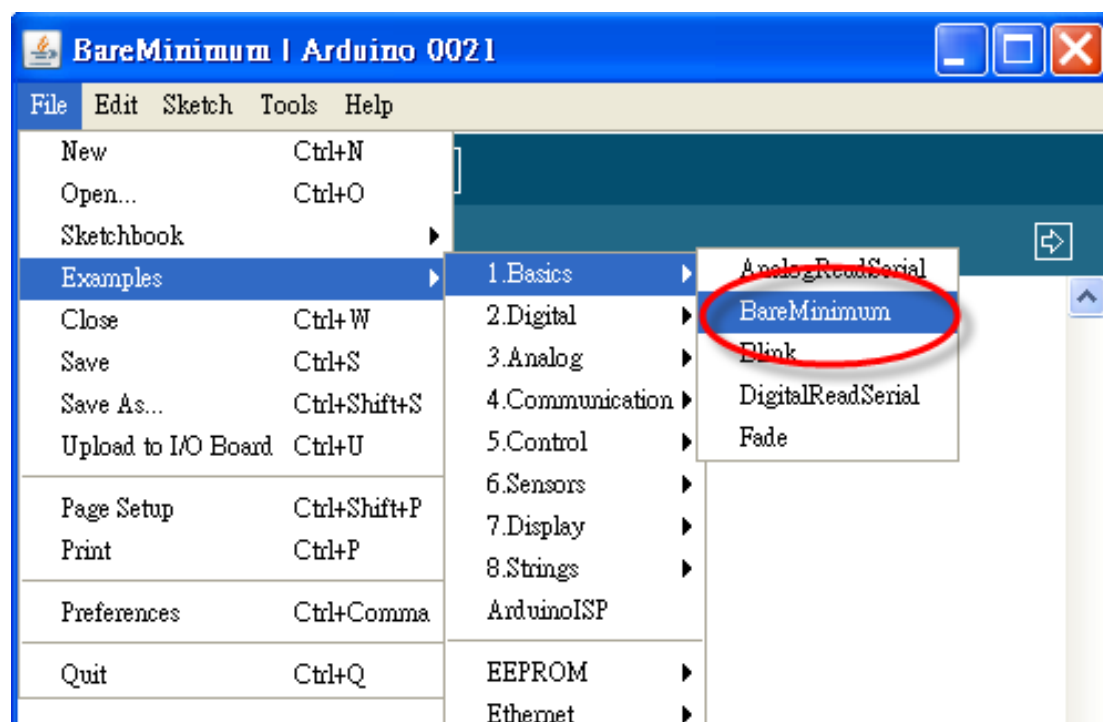
Arduino 的软件开发环境是开放源码的 IDE (Open-source IDE)，可以在它的[官网](#)免费下载，它所用的程序语言语法类似于 C/C++，而且 Arduino IDE 是跨平台的，有 Windows, Macintosh OSX 和 Linux 的版本。Arduino IDE 的软件接口如下：



Arduino 的程序叫作 Sketch，Sketch 意为脚本、素描、速写或小品，因为 Arduino 程序都小小一个，不是很大，之所以取名为 Sketch，猜想大概有小品的意思。Arduino 程序主要由 setup() 和 loop() 这两个函数组成：

```
1 void setup() {  
2   // setup 函数只会跑一次  
3 }  
4  
5 void loop() {  
6   // loop 函数会不断的执行  
7 }
```

因为 `setup()` 和 `loop()` 是每支 Arduino 程序都会用到的两个函数，为了方便，Arduino IDE 已经帮大家准备好程序骨架了，我们在写 Arduino 程序的时候，可以直接点选 **File > Examples > 1.Basics > BareMinimum** 这个模板档，另存新档后，然后再填写 `setup()` 和 `loop()` 两个函数式的内容即可。



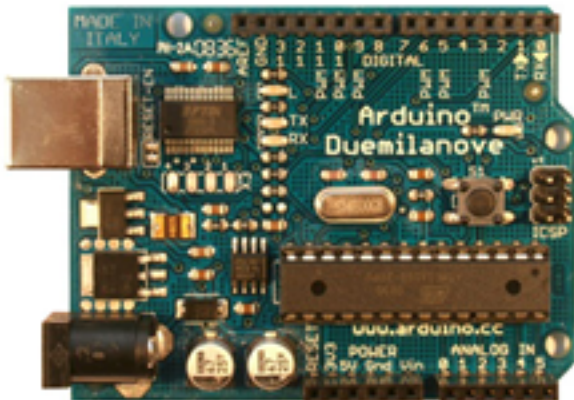
Arduino 硬件版本

Arduino 控制板 (I/O Boards) 有很多种版本，我们在使用的时候，可以依据自己的需求(例如用途、尺寸、容量、I/O 脚数量等)挑选适当的板子。底下条列几款 Arduino 控制板，更多的信息请上 [Arduino 官网](http://arduino.cc) 查询。

Arduino Uno



Arduino Duemilanove



Arduino Mega



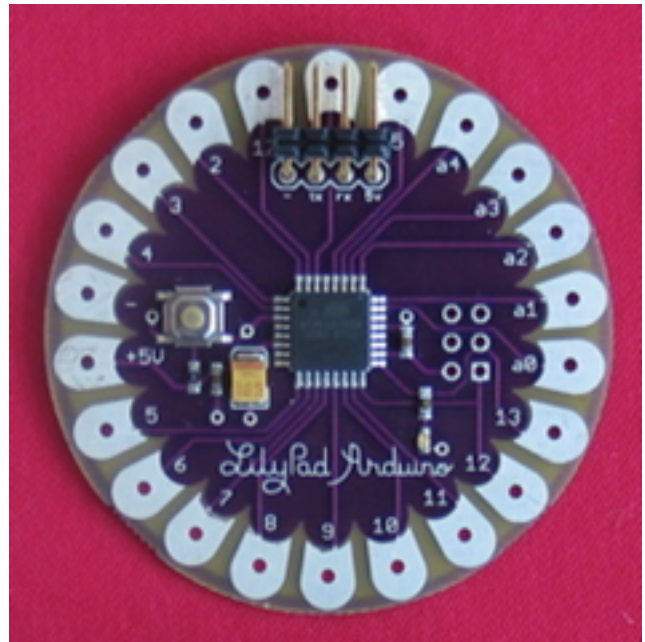
Arduino Fio



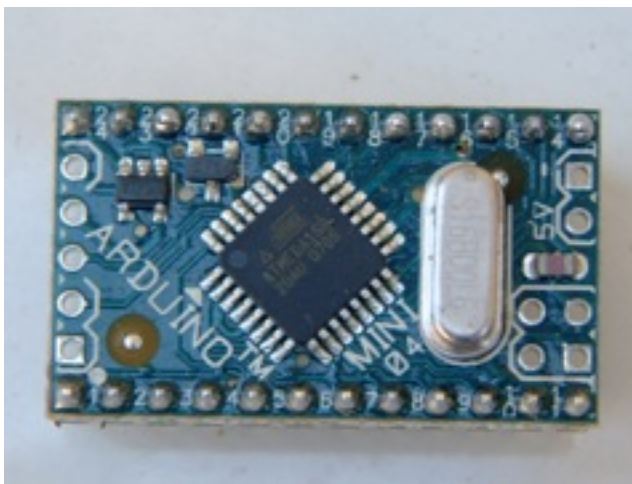
Arduino Nano 3.0



LilyPad Arduino 02



Arduino Mini



Arduino BT (BlueTooth)



MP3 Shield

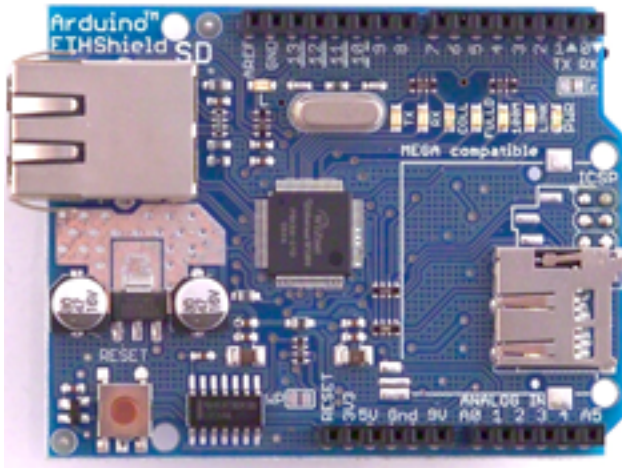
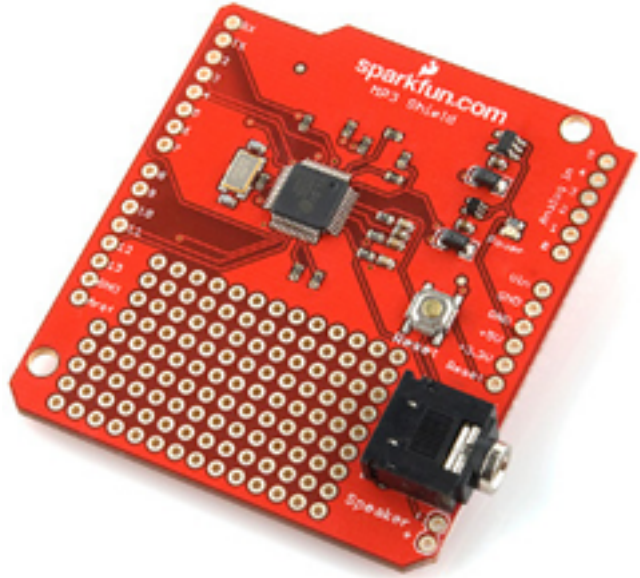
提供 MP3 音樂解碼能力

Arduino shield 扩充板

Shields 是扩充板，就像好像堆积木一样，扩充板可以直接迭在 Arduino 控制板上，让 Arduino 增加更加的能力，例如控制游戏杆、通讯、记忆卡、LCD 显示、MP3 音乐播放等等。底下条列几款扩充板，更多的信息请上 [Arduino Shield List](#) 网站查询。

Ethernet Shield

提供 Ethernet 上網能力



Input Shield

提供 Joystick 搖桿、按鍵及震動馬達能力

TouchShield

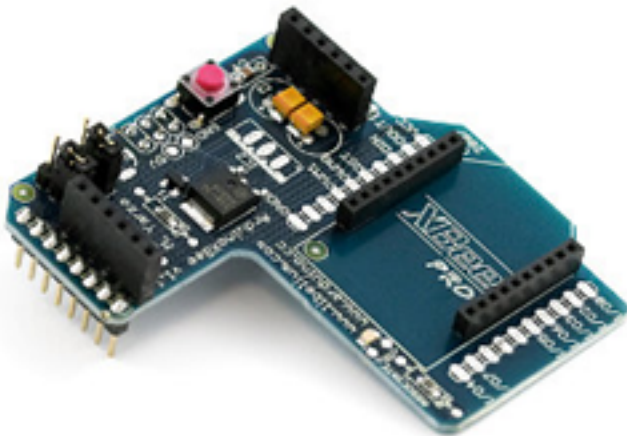
提供 OLED 觸控螢幕顯示能力



XBee Shield
提供 ZigBee 无线通信能力



GPS Shield
提供 GPS 定位能力



WiShield
提供 Wi-Fi 上网能力



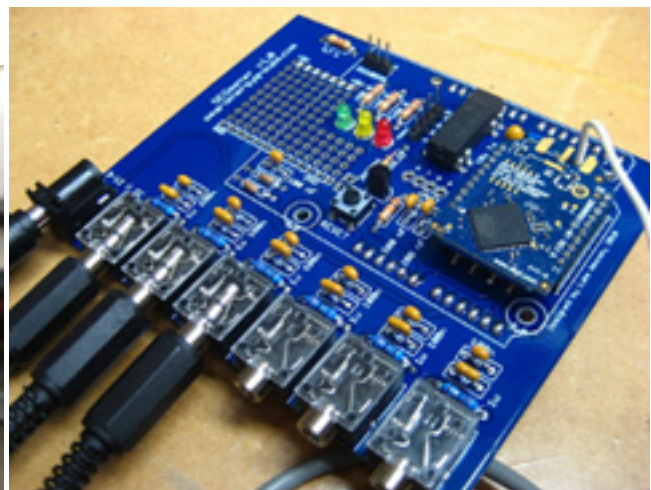
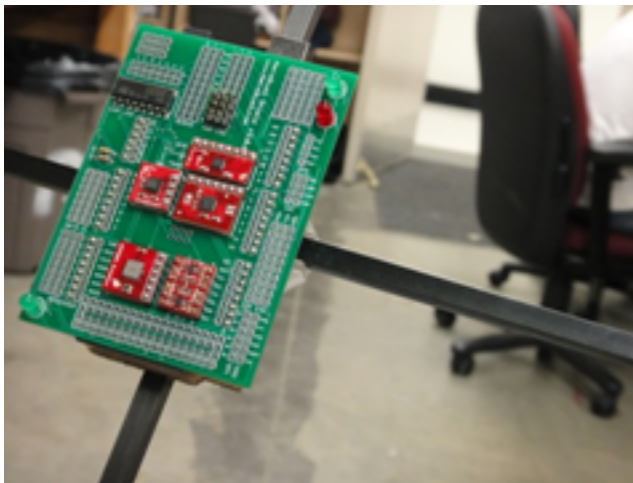
microSD Shield
提供 microSD 储存媒体能力



AeroQuad Shield
提供三轴陀螺仪与加速计的控制



Smart Energy Groups SEGMeter
提供家庭与工业用电测量能力



项目一：LED 闪烁

项目目的

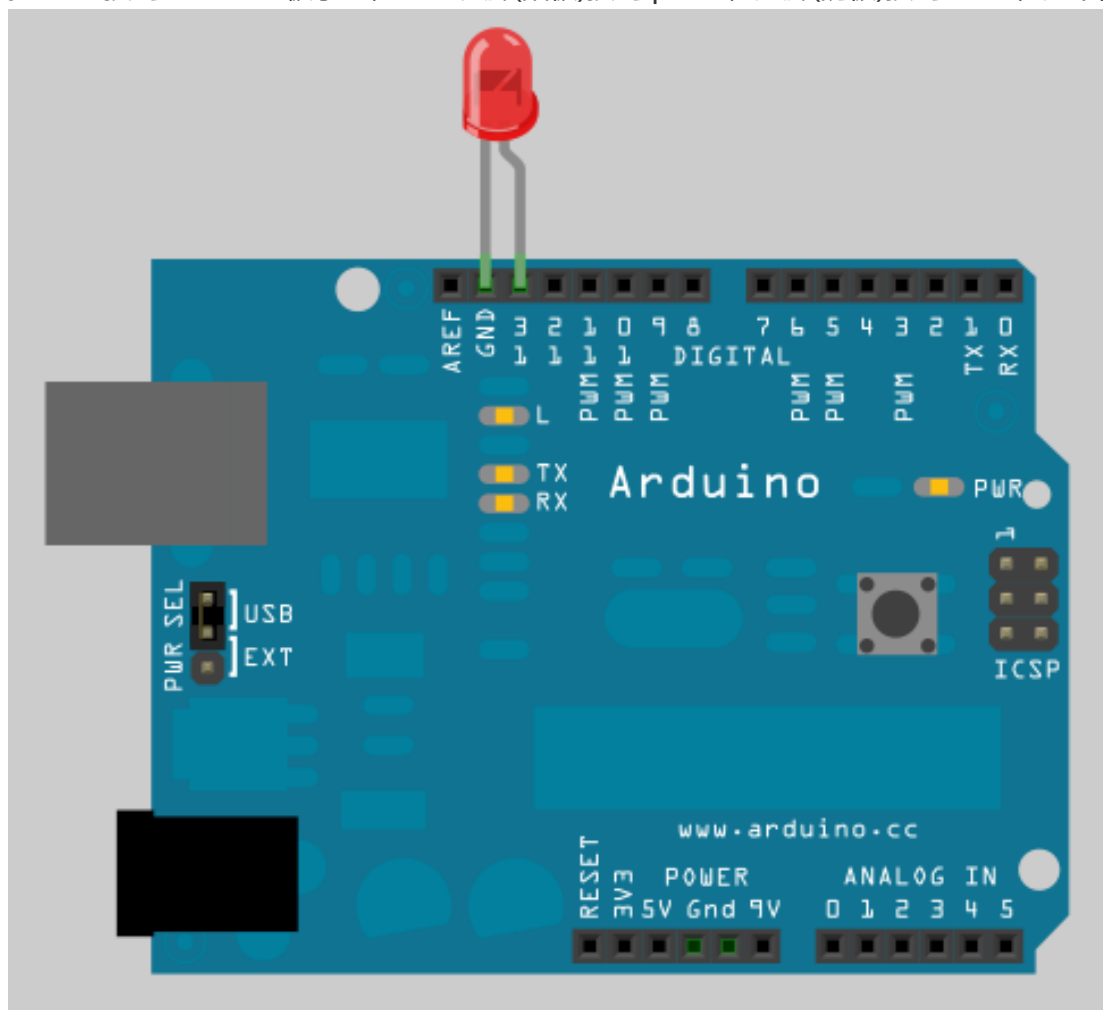
让一颗灯号闪烁，每隔一秒切换一次灯号。

材料

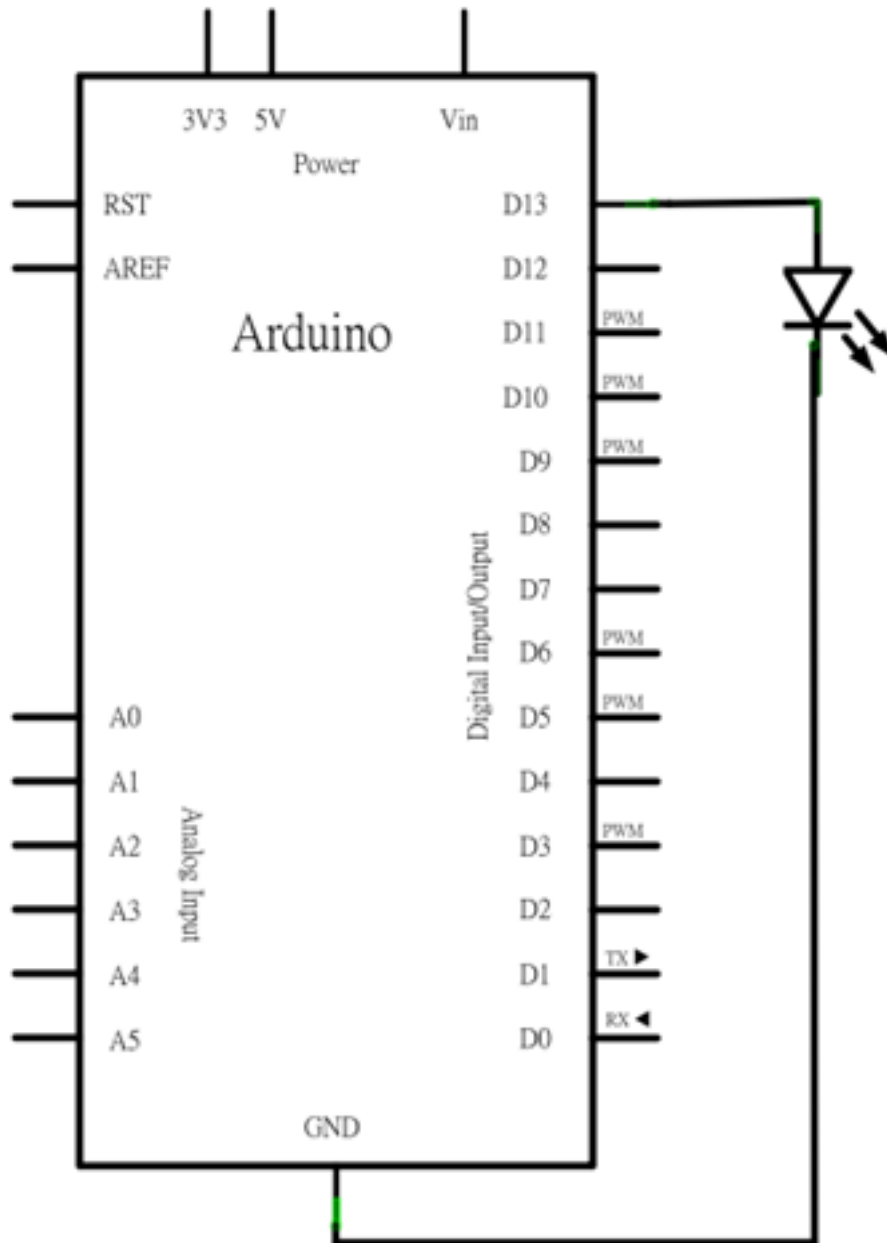
1. Arduino 主板 x 1
2. LED x 1

接线

把 LED 接到 Arduino 板子上，LED 长脚(阳极)接到 pin13，短脚(阴极)接到 GND，如下图：



电路图



代码



```
01  /*
02  Blink
03  Turns on an LED on for one second, then off for one second,
    repeatedly.
04
05  This example code is in the public domain.
06  */
07
08  void setup() {
09    // initialize the digital pin as an output.
10    // Pin 13 has an LED connected on most Arduino boards:
11    pinMode(13, OUTPUT);
12  }
13
14  void loop() {
15    digitalWrite(13, HIGH);    // set the LED on
16    delay(1000);               // wait for a second
17    digitalWrite(13, LOW);     // set the LED off
18    delay(1000);               // wait for a second
19  }
```

说明:

1. L11: pinMode(13, OUTPUT) 这行把 pin13 设置成 output pin
2. L15: digitalWrite(13, HIGH) 这行供应 5V 电压到 pin13, 藉此打开 LED 灯号
3. L16: delay(1000) 让 CPU 闲置一秒钟, 让 LED 灯号亮着一秒钟
4. L17: digitalWrite(13, LOW) 关闭灯号
5. L18: delay(1000) 让 CPU 闲置一秒钟, 让 LED 灯号关闭一秒钟

注: 这支是 Arduino 内建的范例程序, 点选 **File > Examples > 1.Basics > Blink** 就可以找到。

项目二 使用按键控制 LED

目的

使用按键 (PushButton)控制 LED 灯号的开关, 当按键被按下时打开 LED 灯号, 按键放开时关闭 LED 灯号。

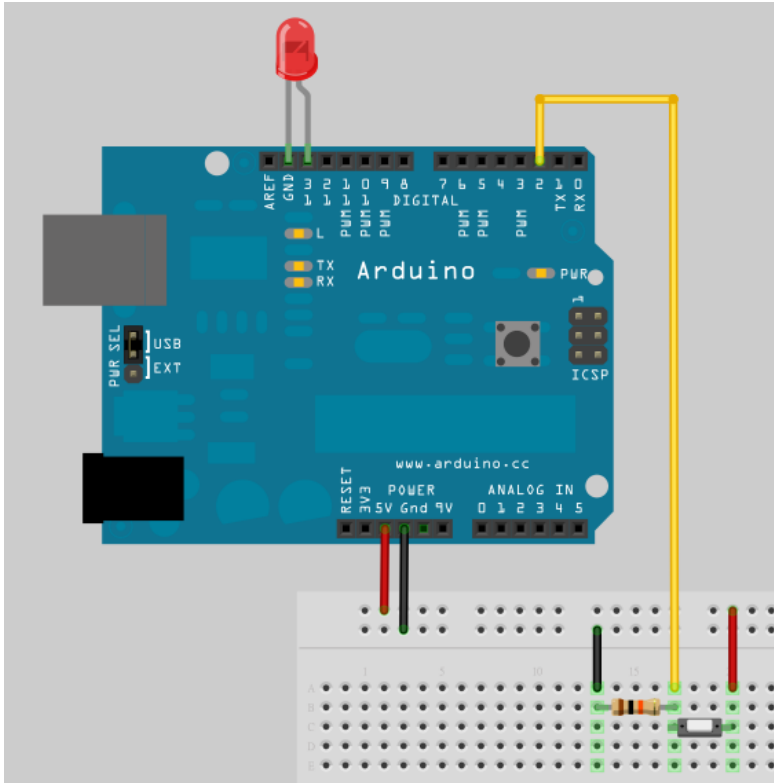
材料

1. Arduino 主板 x 1
2. LED x 1
3. PushButton 或 Switch 开关 x 1
4. 10K 电阻 x 1
5. 麵包板 x 1

6. 单心线 x N

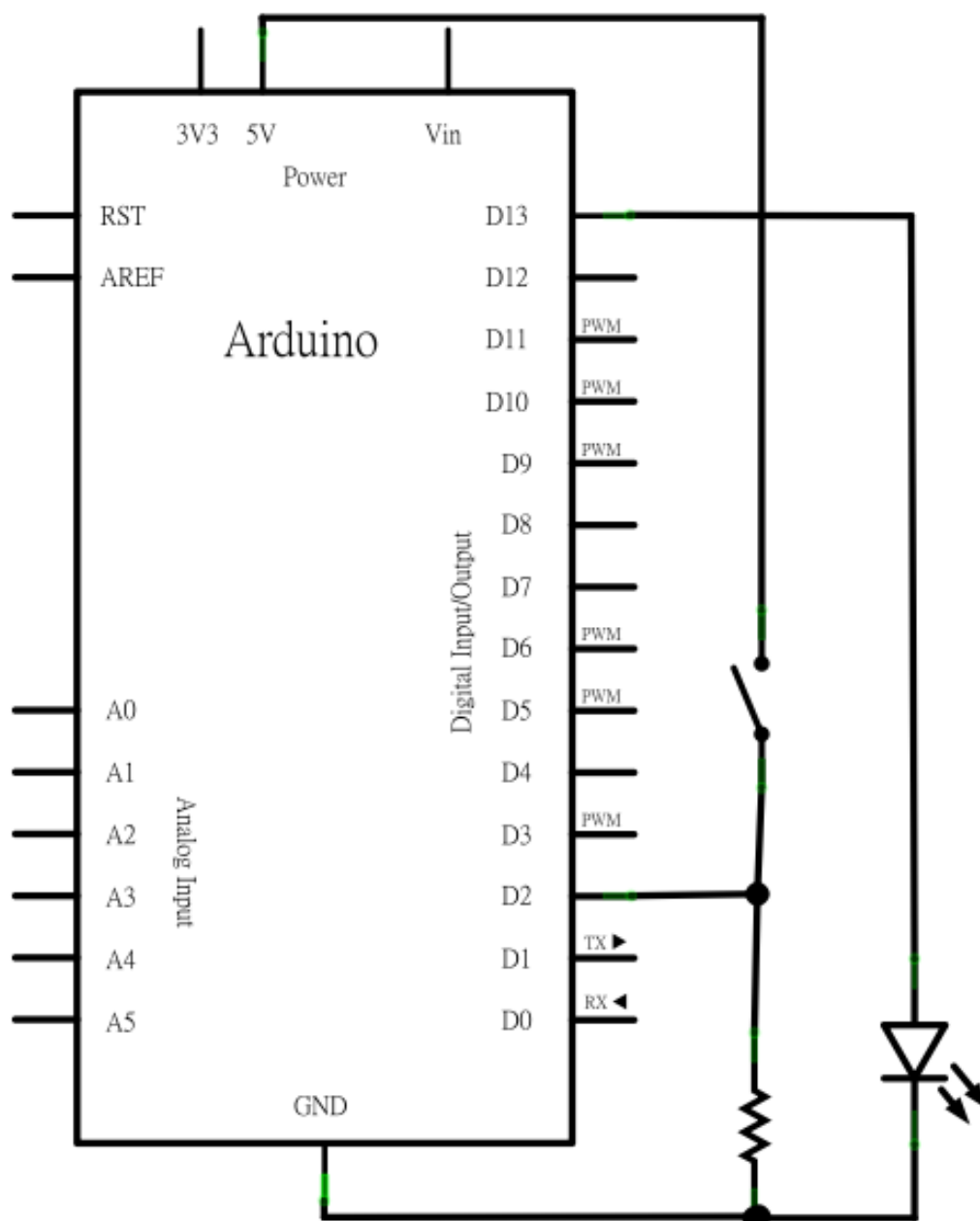
接线

1. 把 LED 接到 pin13, 长脚(阳极)接到 pin13, 短脚(阴极)接到 GND
2. PushButton 一支脚接到 +5V
3. pin2 接到 Pushbutton 的另一支脚, 同一支脚位接一个 10K 电阻连到 GND



4.

电路图



代码

```
01 const int buttonPin = 2;    // the number of the pushbutton pin
02 const int ledPin = 13;     // the number of the LED pin
03
04 // variables will change:
05 int buttonState = 0;        // variable for reading the
    pushbutton status
06
07 void setup() {
08   // initialize the LED pin as an output:
09   pinMode(ledPin, OUTPUT);
10   // initialize the pushbutton pin as an input:
11   pinMode(buttonPin, INPUT);
12 }
13
14 void loop() {
15   // read the state of the pushbutton value:
16   buttonState = digitalRead(buttonPin);
17
18   // check if the pushbutton is pressed.
19   // if it is, the buttonState is HIGH:
20   if (buttonState == HIGH) {
21     // turn LED on:
22     digitalWrite(ledPin, HIGH);
23   }
24   else {
25     // turn LED off:
26     digitalWrite(ledPin, LOW);
27   }
28 }
```

说明:

1. L01~L02: 定义按键与 LED 的脚位, 按键接在 pin2, 而 LED 接在 pin13
2. L16: 读取按键的状态并保存到 buttonState 变量里
3. L20~L26: 这支程序的目的是按下按键时要打开 LED 灯号, 放开按键时要关闭 LED 灯号, 因此, 假如 buttonState 为 HIGH, 代表按键状态是按下(pressed)的, 此时要打开 LED, 反之, 假如 buttonState 为 LOW, 代表按键状态是放开的, 此时要关闭 LED

注: 这支是 Arduino 内建的范例程序, 点选 **File > Examples > 2.Digital > Button** 就可以找到。

挑战:

在不修改程序代码的条件下, 让 LED 变成在正常情况下是亮的, 而当按下按键时让 LED 灯号关掉。提示: 条件是**不能**修改程序代码, 要改变这支程序的行为, 你只能够动接线。

项目三 控制 LED 灯光亮度

挑战目的:

利用 PWM (Pulse Width Modulation, 脉冲宽度调变) 控制 LED 灯光亮度。

材料

1. Arduino 主板 x 1
2. LED x 1
3. 220 ohm 电阻 x 1
4. 单心线 x N

接线

1. LED 接到 pin9 和 GND，长脚(阳极)串接一颗 220 ohm 电阻到 pin9，短脚(阴极)直接接到 GND

实验目的

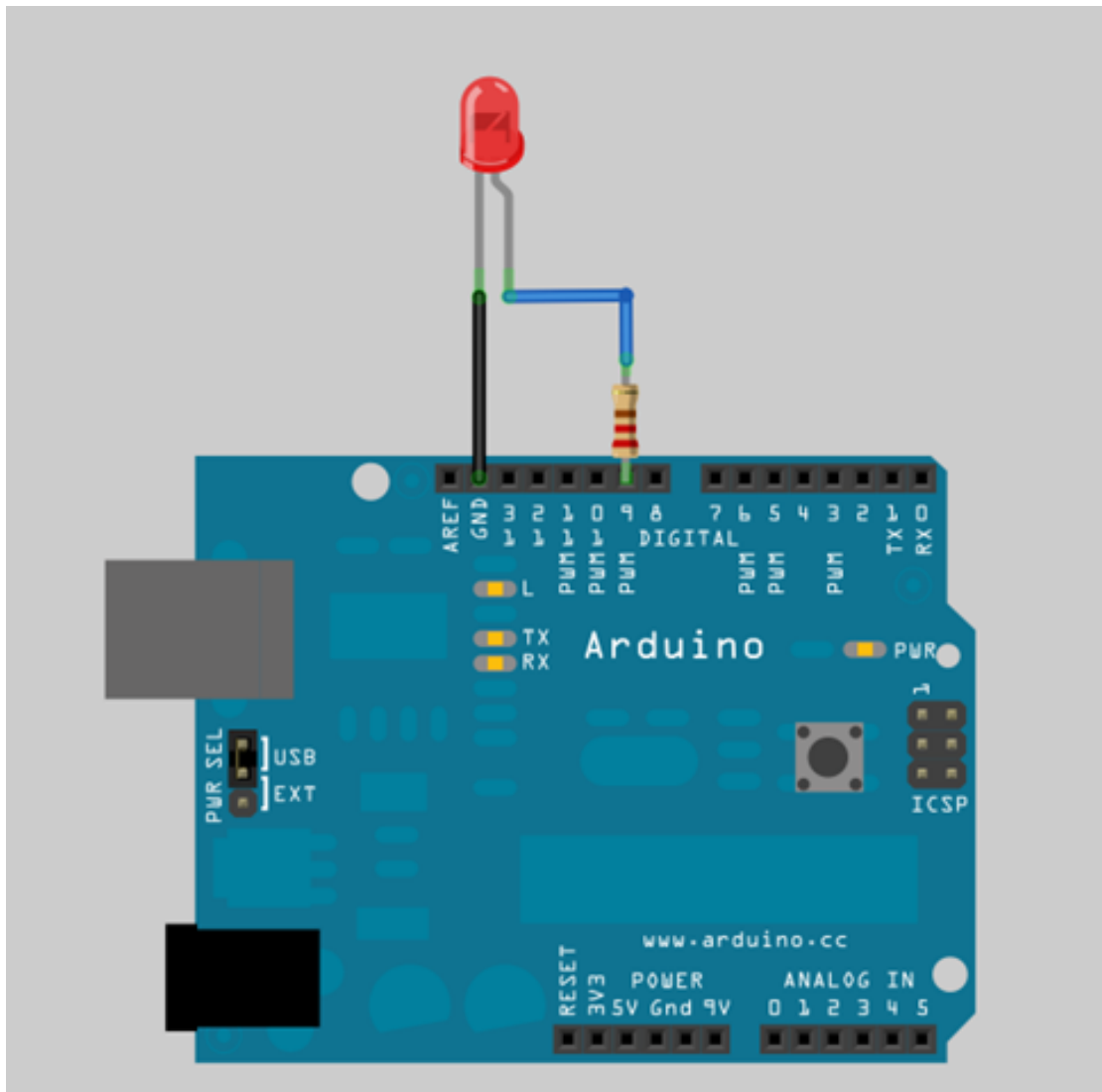
利用 PWM (Pulse Width Modulation, 脉冲宽度调变) 控制 LED 灯光亮度。

材料

1. Arduino 主板 x 1
2. LED x 1
3. 220 ohm 电阻 x 1
4. 单心线 x N

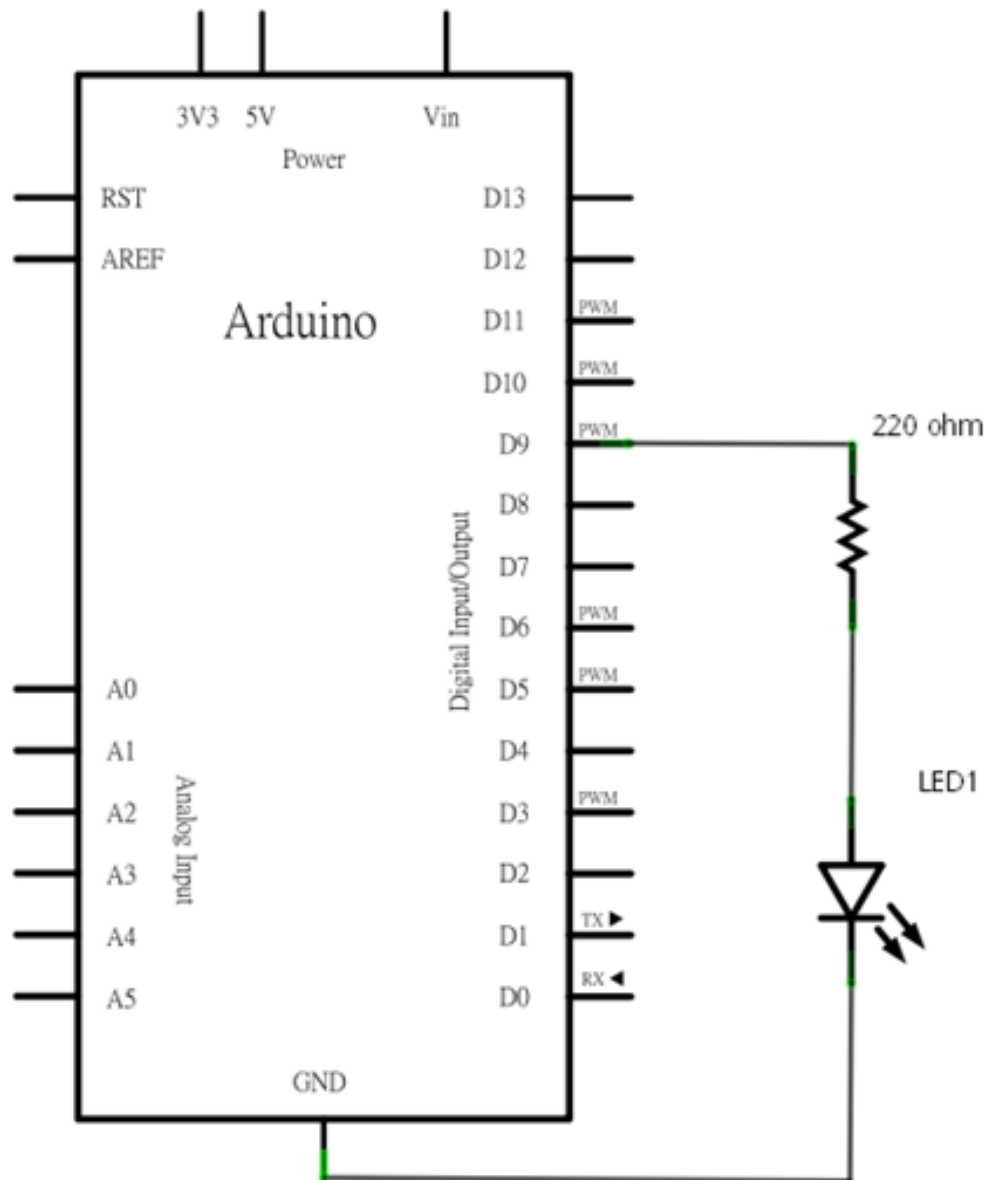
接线

1. LED 接到 pin9 和 GND，长脚(阳极)串接一颗 220 ohm 电阻到 pin9，短脚(阴极)直接接到 GND



2.

电路图



代码：

```
01 int brightness = 0;    // how bright the LED is
02 int fadeAmount = 5;    // how many points to fade the LED by
03
04 void setup() {
05     // declare pin 9 to be an output:
06     pinMode(9, OUTPUT);
07 }
08
09 void loop() {
10     // set the brightness of pin 9:
11     analogWrite(9, brightness);
12
13     // change the brightness for next time through the loop:
14     brightness = brightness + fadeAmount;
15
16     // reverse the direction of the fading at the ends of the
    fade:
17     if (brightness == 0 || brightness == 255) {
18         fadeAmount = -fadeAmount ;
19     }
20     // wait for 30 milliseconds to see the dimming effect
21     delay(30);
22 }
```

编译这支程序，然后上传到 Arduino 板子上，过数秒后，就会看到 LED 灯光不断地改变亮度，一下子渐亮，一下渐暗。

说明：

1. L01: brightness 变量用来保存目前的灯光亮度
2. L02: fadeAmount 变量用来设定每一次灯光亮度的调整值
3. L06: 宣告 pin9 为 output pin, LED 接在 pin9 上
4. L11: 使用 analogWrite(9, brightness) 设定 pin9 上的 LED 灯光亮度
5. L14: 调整下一次的灯光亮度
6. L17~L19: 改变 fadeAmount 灯光亮度调整值，假如 brightness 已达到最亮(255)，就将 fadeAmount 改成 -5，让灯光下一次的变化改成渐渐变暗，假如 brightness 已达到最暗(0)，就将 fadeAmount 改回 +5，让灯光下一次的变化改成渐渐变亮。
7. L21: 延迟 30ms，这样肉眼才能看得到 LED 调光的效果

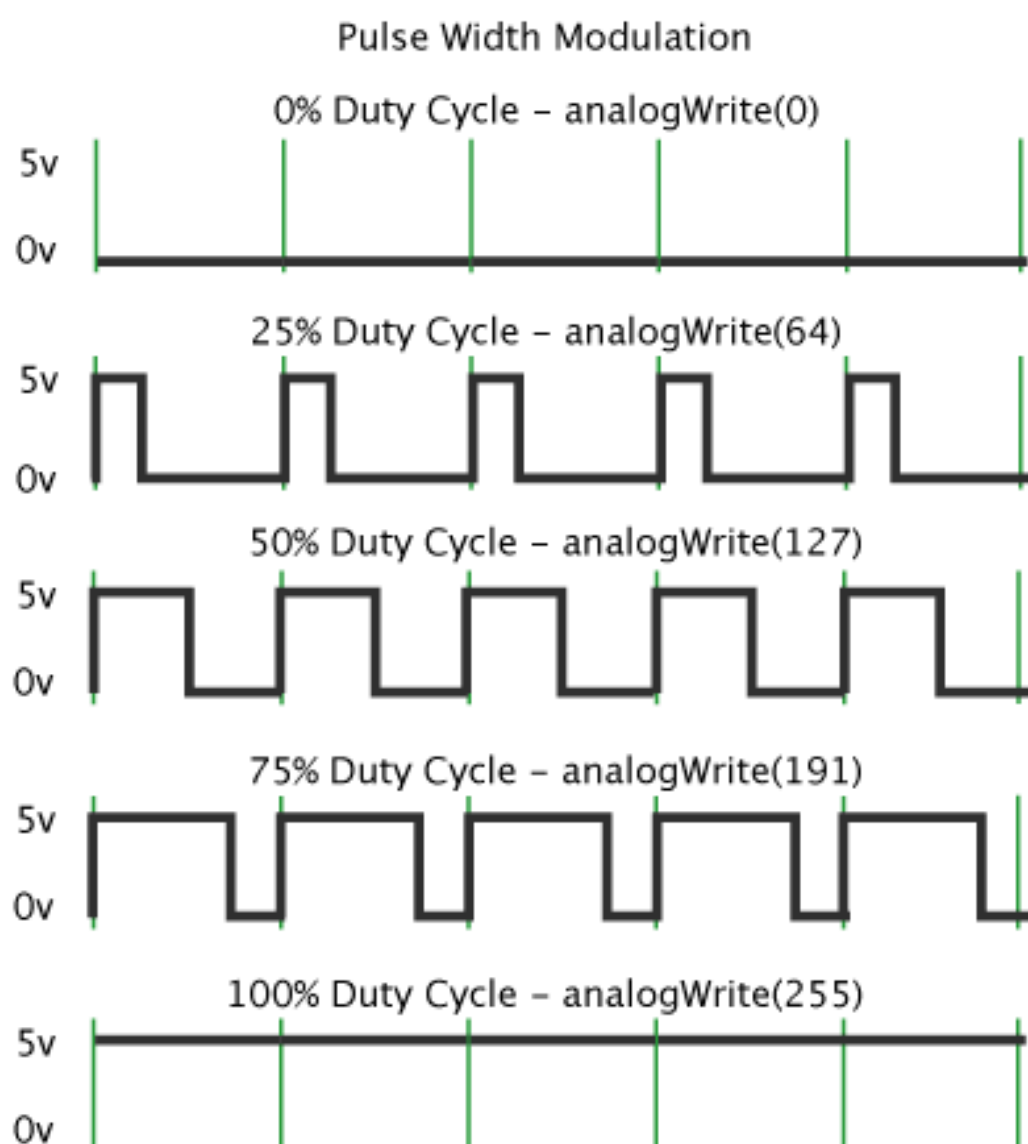
PWM 原理简介

数字输出可以控制讯号的开跟关，开意味着通电，关意味着断电，如果我们能够进一步通电的时间比例，就能让模拟输出产生变化，例如 LED 灯光通电时间为 50%，就可以控制 LED 让它只有 50% 的亮度，如果把通电时间比例改为 25%，就可以控制 LED 让它只有 25% 的亮度。这个方法称为 PWM

(Pulse Width Modulation)脉冲宽度调变，PWM 是一个利用数字讯号来控制模拟输出的技术，常用于蜂鸣器、电热器、马达或风扇转速、灯光亮度等的控制。

下图中，垂直的绿线代表一个规律的时间周期，在 Arduino 中，每个周期是 2ms (PWM 频率为 500Hz)。analogWrite() 的参数值范围介于 0 到 255 之间，analogWrite(255) 代表产生 100% duty cycle 的输出(一直通电)，而 analogWrite(127) 则是产生 50% duty cycle 的输出(一半的时间通电，一半的时间断电)。

Duty Cycle 为每一个周期通电(on) 的时间比例。



挑战

1. 如何加快或放慢 LED 亮度变化的速度？（提示：delay() 函式）
2. 改用 pin9 以外的其它 PWM 脚位
3. 多接几颗 LED，让每颗 LED 使用不同的速率改变亮度。

项目四 使用可变电阻调光

挑战目的

使用可变电阻 (potentiometer) 控制 LED 的灯光亮度，达到调光的目的。

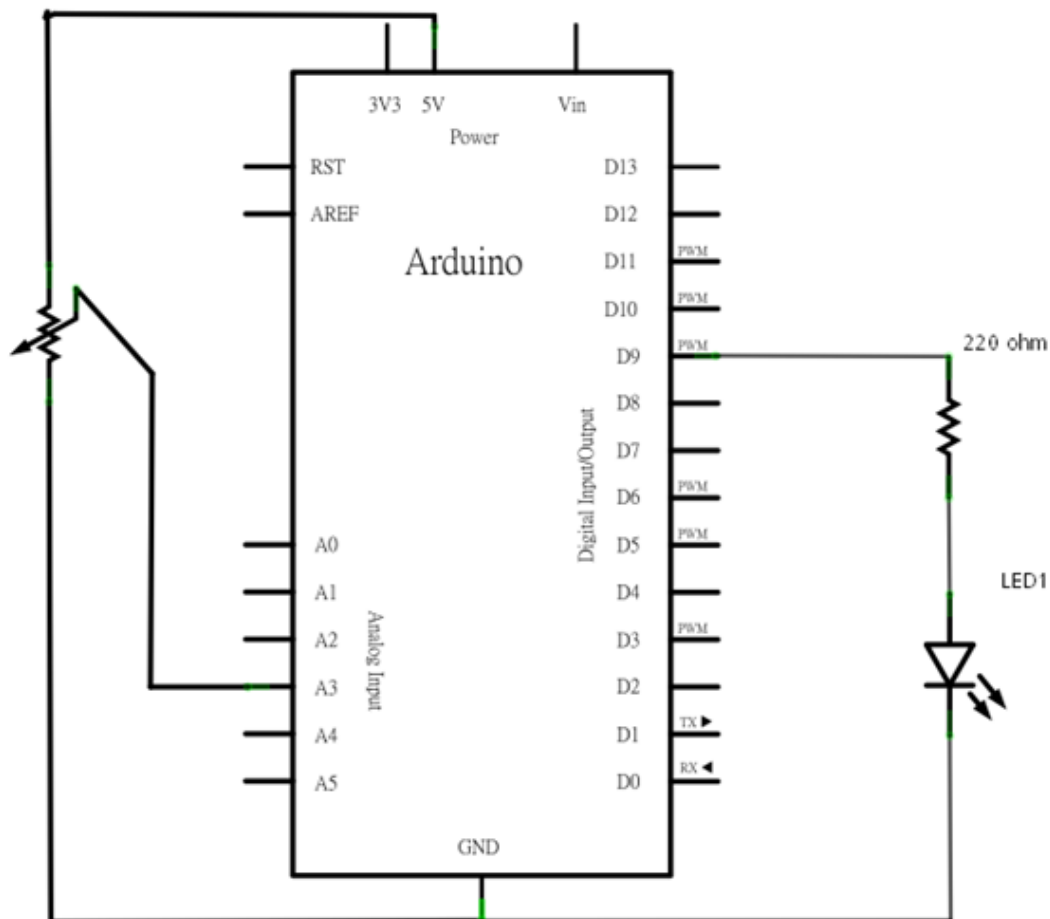
材料

1. 麵包板 x 1
2. Arduino 主板 x 1
3. LED x 1
4. 220 ohm 电阻 x 1
5. 可变电阻 x 1
6. 单心线 x N

接线

1. LED 接到 pin9 和 GND，长脚(阳极)串接一颗 220 ohm 电阻到 pin9，短脚(阴极)直接接到 GND
2. 可变电阻中间脚位接到模拟输入(Analog Input) pin3，剩下的两支脚位，一支接到 5V，另外一支接到 GND

电路图



代码范例

```

01 int potPin = 3; // select the input pin for the potentiometer
02 int ledPin = 9; // select the pin for the LED
03
04 void setup() {
05   Serial.begin(9600);
06 }
07
08 void loop() {
09   int sensorValue = analogRead(potPin);
10   Serial.println(sensorValue, DEC);
11
12   sensorValue = sensorValue/4; // convert from 0-1024 to 0-255
13   analogWrite(ledPin, sensorValue);
14   delay(150);
15 }
  
```

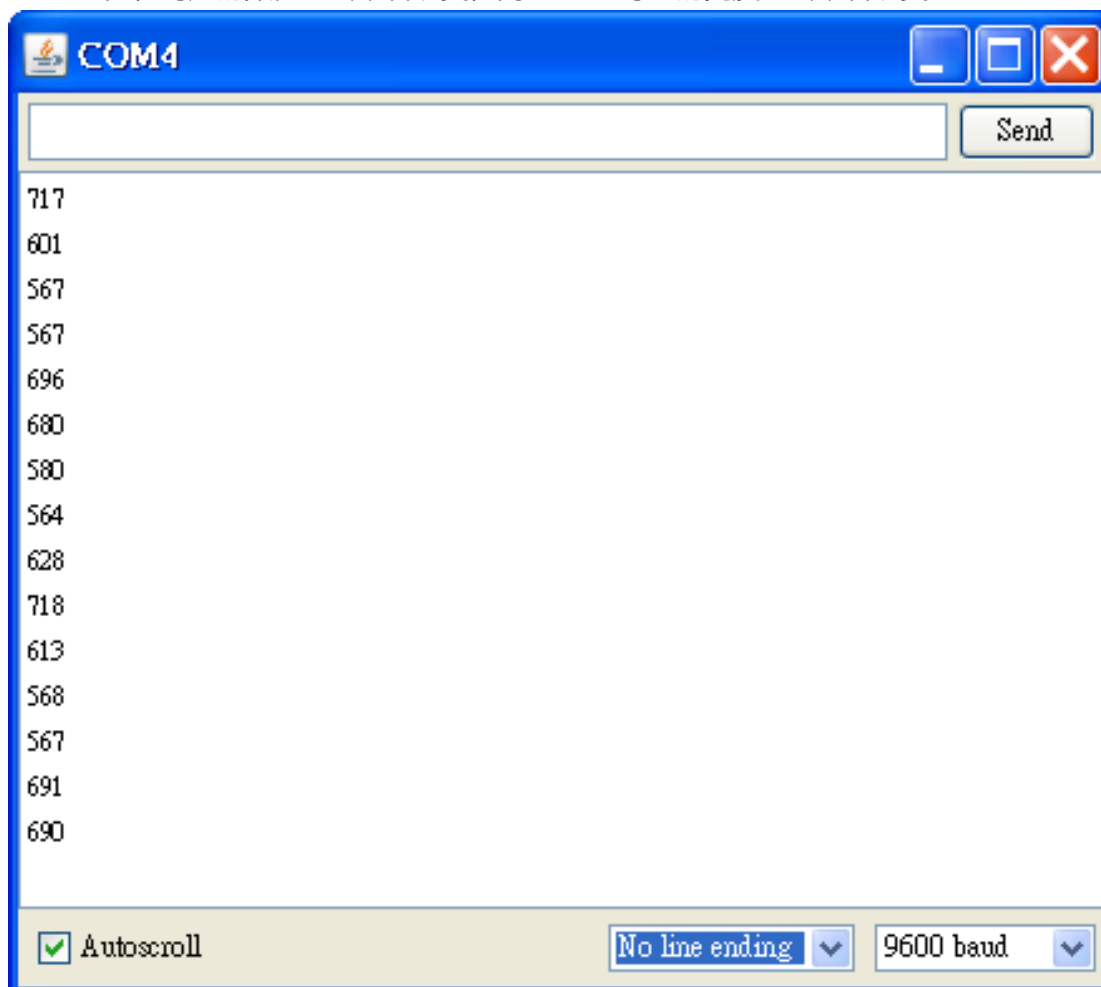
说明:

1. L01: 宣告可变电阻所用的模拟输入脚位
2. L02: 宣告 LED 灯号脚位

3. L05: 设定 SerialPort 的传输速率, 速率为 9600 bps (bit per second)
4. L09: 读取可变电阻读值并且放到 sensorValue 变量里
5. L10: 使用 Serial.println(sensorValue, DEC) 把电阻值打印到 SerialPort。DEC 代表以十进制显示数字
6. L12: analogRead() 读进来的是一个 10 位的数值, 值域为 0 到 1023, 由于 analogWrite() 的参数只能接受 0 到 255 的数值, 所以得将 sensorValue 除以 4, 让 sensorValue 的数值从 0-1023 等比例缩小到 0-255 的范围。



编译好程序, 上传到 Arduino 后, 按下 Serial Monitor 这个按钮, 就会看到 COM Port 不断收到一串范围介于 0 到 1023 的数值, 这些数值与可变电阻是相关联的。如果旋转可变电阻, 对应的数值也会跟着改变, 而且 LED 灯号的亮度也会跟着改变。





同济大学设计创意学院
COLLEGE OF DESIGN AND INNOVATION TONGJI UNIVERSITY



Fablab O
中国“数制”工坊