

Week 2

Howard Ke

Code

```
void setup() {  
    pinMode(13, OUTPUT);    //設定腳位13為輸出  
}  
  
void loop() {  
    digitalWrite(13, HIGH); //腳位13輸出5V, LED亮  
    delay(1000);            //等1000毫秒  
  
    digitalWrite(13, LOW);  //腳位13輸出0V, LED暗  
    delay(1000);            //等1000毫秒  
}
```

Update

```
int led_pin = 13;

void setup() {
    pinMode(led_pin, OUTPUT);    //設定腳位13為輸出
}

void loop() {
    digitalWrite(led_pin, HIGH); //腳位13輸出5V, LED亮
    delay(1000);                //等1000毫秒

    digitalWrite(led_pin, LOW);  //腳位13輸出0V, LED暗
    delay(1000);                //等1000毫秒
}
```

Update

```
int led_pin1 = 12;
int led_pin2 = 13;

void setup() {
    pinMode(led_pin1, OUTPUT);
    pinMode(led_pin2, OUTPUT);
}

void loop() {
    digitalWrite(led_pin1, HIGH);
    delay(1000);
    digitalWrite(led_pin1, LOW);
    delay(1000);
    digitalWrite(led_pin2, HIGH);
    delay(1000);
    digitalWrite(led_pin2, LOW);
    delay(1000);
}
```

變數

```
int a = 10;  
int b = 9;
```

```
a = 20;           // a: 20, b: 9  
a = 20 - 1;       // a: 19, b: 9  
a = a - 1;        // a: 18, b: 9  
a += 100;         // a: 118, b: 9  
a -= 80;          // a: 38, b: 9  
a--;              // a: 37, b: 9  
a = b;            // a: 9, b: 9  
a = b - 1;        // a: 8, b: 9
```

運算子

符號	說明	範例
=	賦值	<code>a = 10</code>
+	加	<code>a = 10 + 1</code>
-	減	<code>a = 10 - 1</code>
*	乘	<code>a = 10 * 2</code>
/	除	<code>a = 10 / 2</code>
%	取餘數	<code>a = 10 % 2</code>
<op>=	算術指派	<code>a *= 10</code>
--	減一	<code>a--</code>
++	加一	<code>a++</code>

變數命名

- 組成：大小寫英文字母、數字、底線

`i_10ve_c0de`

- 第一個字不能是數字

`1stnum`

- 不要重複宣告

- 大小寫不一樣

`LOVE != love`

變數型態

型態	說明	範例
int	整數	<code>int a = 1;</code>
float	單精度浮點數	<code>float b = 9.4;</code>
double	雙精度浮點數	<code>double c = 8.7</code>
char	字元	<code>char d = 'u';</code>

陣列

```
int a1 = 10;  
int a2 = 20;  
int a3 = 30;  
int a4 = 40;  
int a5 = 50;  
int a6 = 10;  
int a7 = 20;  
int a8 = 30;  
int a9 = 40;  
int a10 = 50;
```

陣列

```
int a[10] = {10, 20, 30, 40, 50, 10, 20, 30, 40, 50};
```

陣列

```
int a[10] = {10, 20, 30, 40, 50, 10, 20, 30, 40, 50};
```

	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]
a	10	20	30	40	50	10	20	30	40	50

各種陣列

```
int a[3] = {1, 2, 3};
```

```
float f[2] = {9.4, 8.7};
```

```
char s[7] = "arduino";
```

陣列賦值

```
int arr[3];
```

```
arr[0] = 1;
```

```
arr[1] = 2;
```

```
arr[2] = 3;
```

陣列賦值

```
int arr[3];
```

```
arr[0] = 1;
```

```
arr[1] = 2;
```

```
arr[2] = 3;
```

```
arr[3] = {1, 2, 3};
```

陣列取值

```
int a[3] = {1, 2, 3};
```

```
arr[1];           // 2
```

```
arr[2];           // 3
```

```
int it = 1;
```

```
arr[it]           // 2
```

```
it += 1;
```

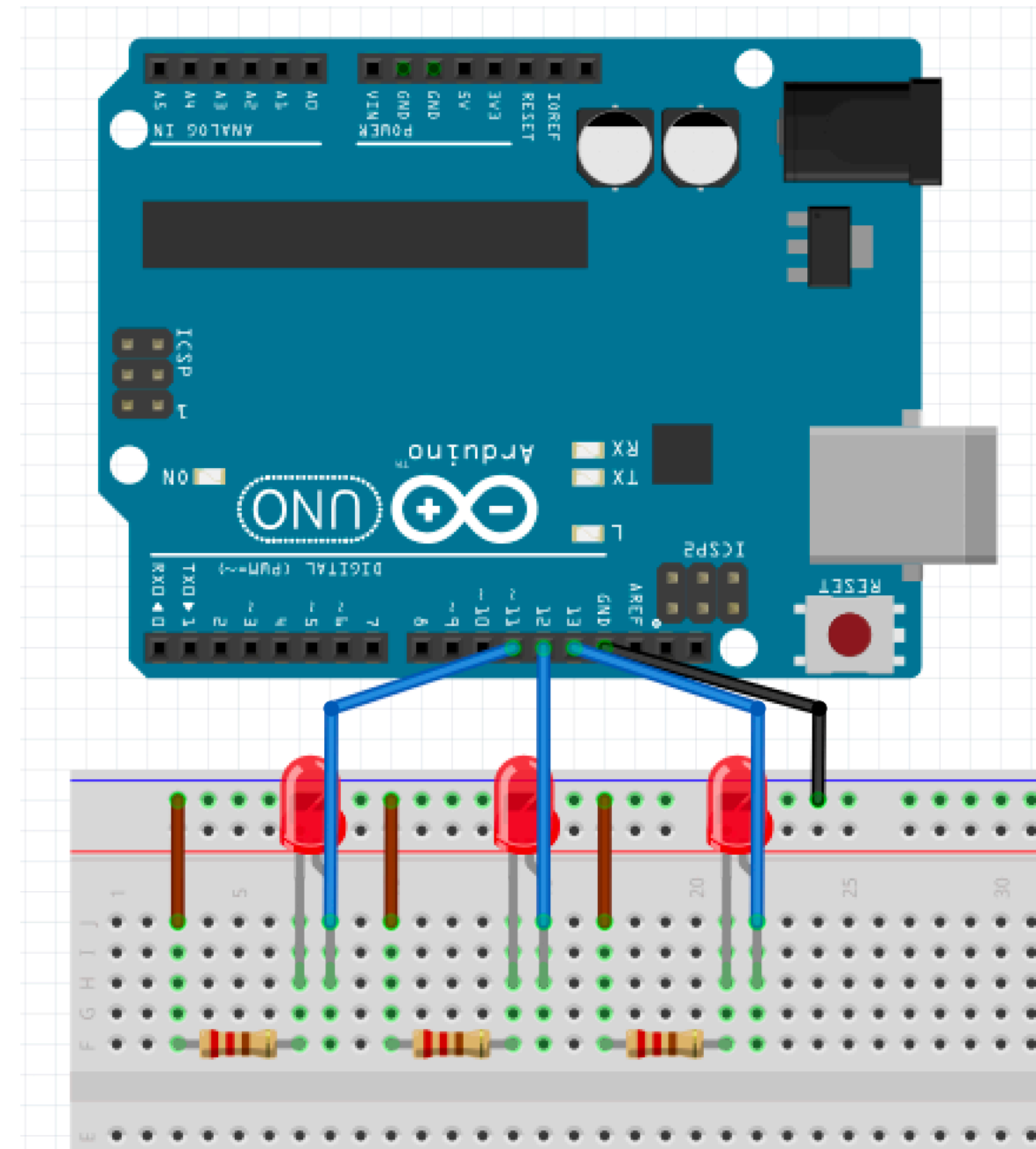
```
arr[it]           // 3
```

Lab 01

Goal: 霹靂燈

1. 用變數+陣列寫出霹靂燈

- 單向
- 3+ LEDs



條件

```
if (條件) {  
    // 如果條件成立執行我  
}
```

條件

```
if (a == 2) {  
    // 如果 a 等於 2 執行我  
}
```

```
if (a < b) {  
    // 如果 a 小於 b 執行我  
}
```

比較

比較	說明
<code>a == b</code>	a 等於 b
<code>a != b</code>	a 不等於 b
<code>a > b</code>	a 大於 b
<code>a < b</code>	a 小於 b
<code>a >= b</code>	a 大於等於 b
<code>a <= b</code>	a 小於等於 b

邏輯

邏輯	說明
&&	而且
	或者
!	不

各種條件

```
if(a == 2 && b == 3){  
    // 如果 a 等於 2 而且 b 等於 3 執行  
}
```

```
if(a < b || a >= 2){  
    // 如果 a 小於 b 而且 a 大於等於 2 執行  
}
```

```
if( !(a < b) ){  
    // 如果 不是 a 小於 b 執行  
}
```

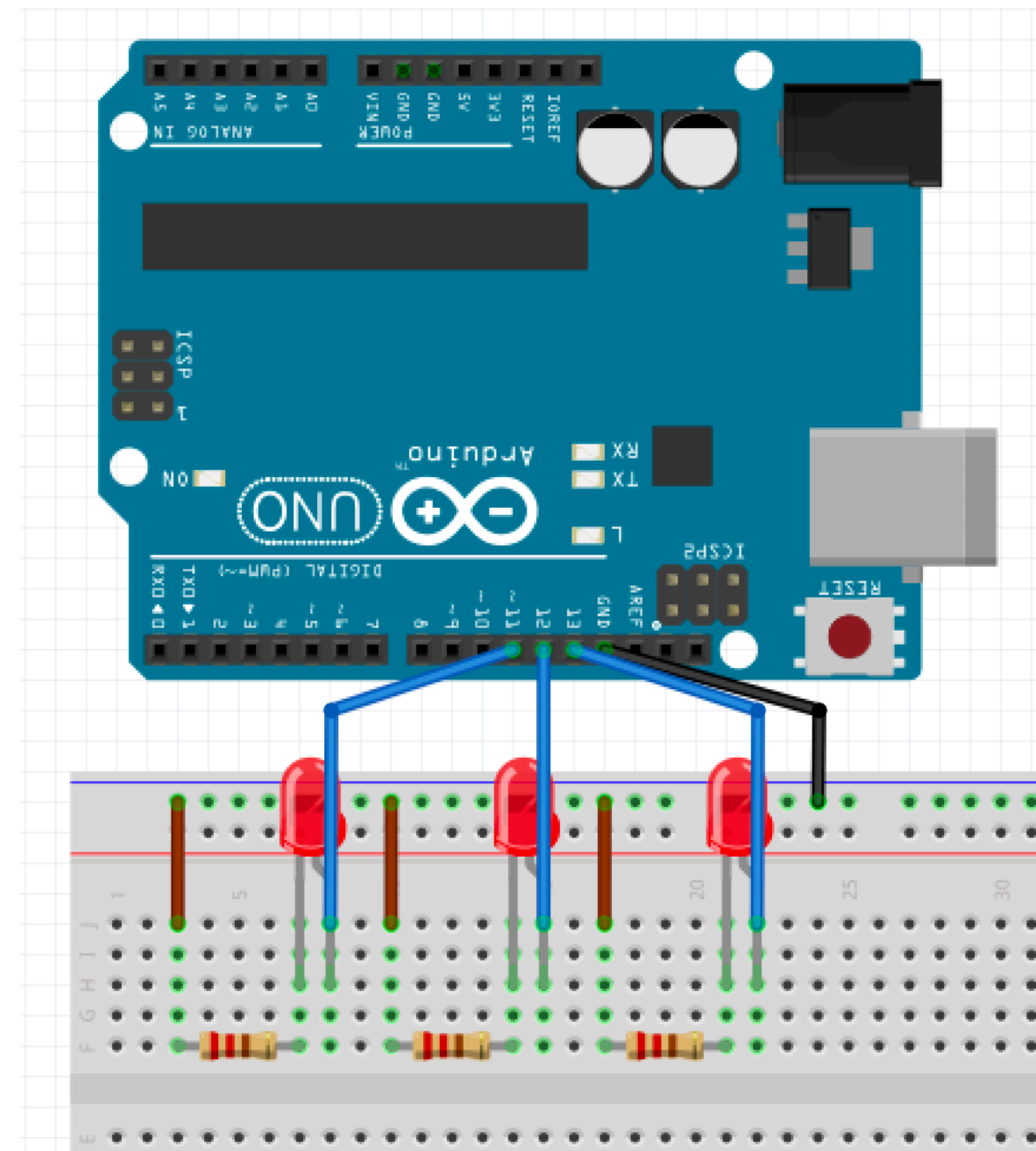
多種條件

```
if( 條件1 ) {  
    // 如果 條件1成立 執行  
}  
else if ( 條件2 ) {  
    // 不然如果 條件2成立 執行  
}  
else {  
    // 條件1, 條件2 都不成立的話執行  
}
```

Lab 02

Goal: 霹靂燈

1. 用條件判斷精簡程式
 - 只用兩次digitalWrite
 - 提示：想想loop() 特性？
2. 改成雙向
 - 提示：增加一個變數？



迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```


迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

```
i: 0  
a: 20
```

迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

```
i: 1  
a: 0
```

迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

```
i: 2  
a: -20
```

迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

```
i: 3  
a: -40
```

迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

```
i: 4  
a: -60
```

迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

```
i: 5  
a: -80
```

迴圈 - while

```
int i = 0, a = 20;
```

```
while(i < 5){  
    a = a - 20;  
    i = i + 1;  
}
```

迴圈 - for

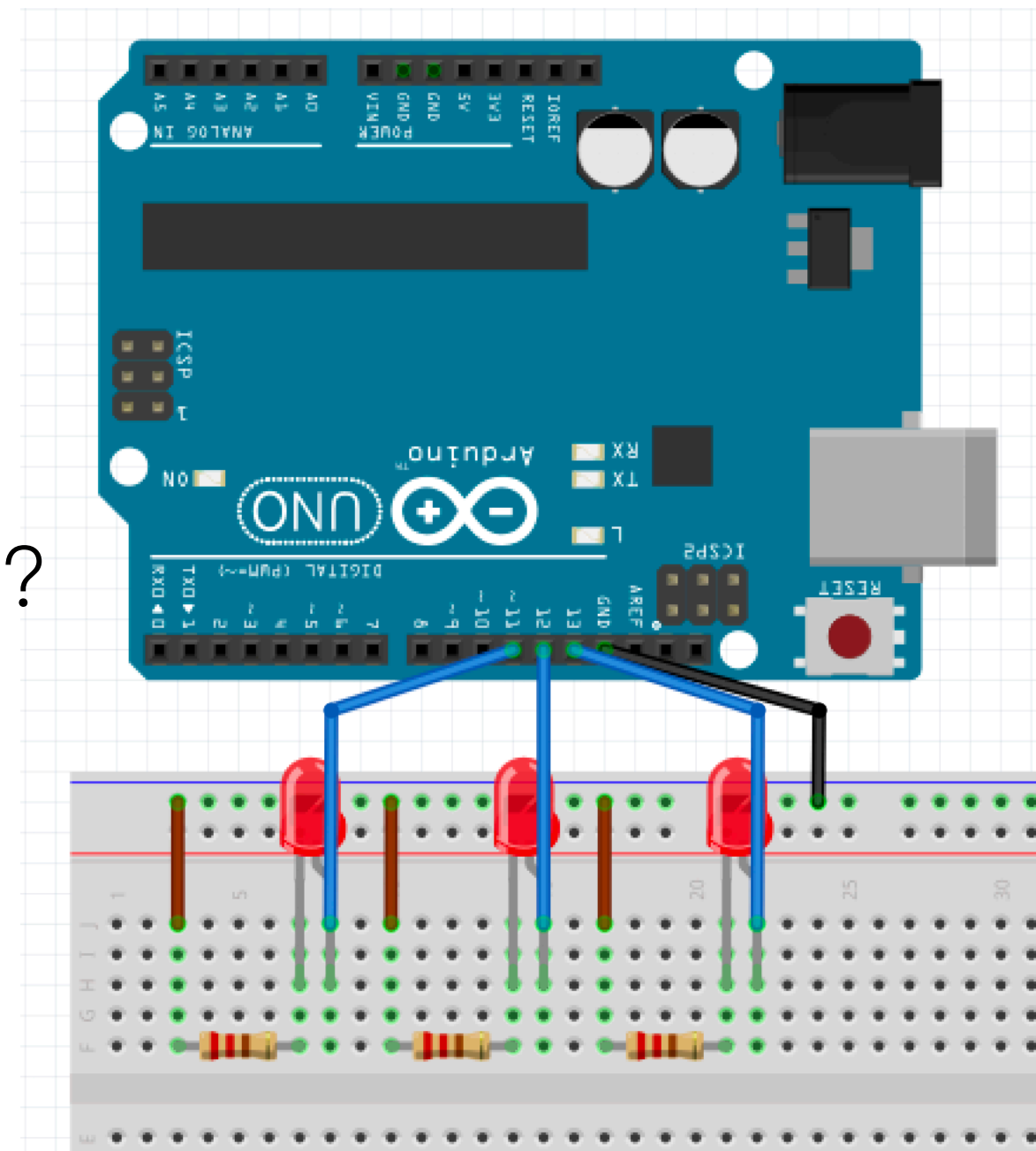
```
int a = 20;
```

```
for(int i = 0; i < 5; i++){  
    a -= 20;  
}
```


Lab 03

Goal: 霹靂燈

1. 使用 for loop 完成雙向
2. —Challenge—
 - 只出現一次 digitalWrite
 - 提示：HIGH, LOW 怎麼辦？

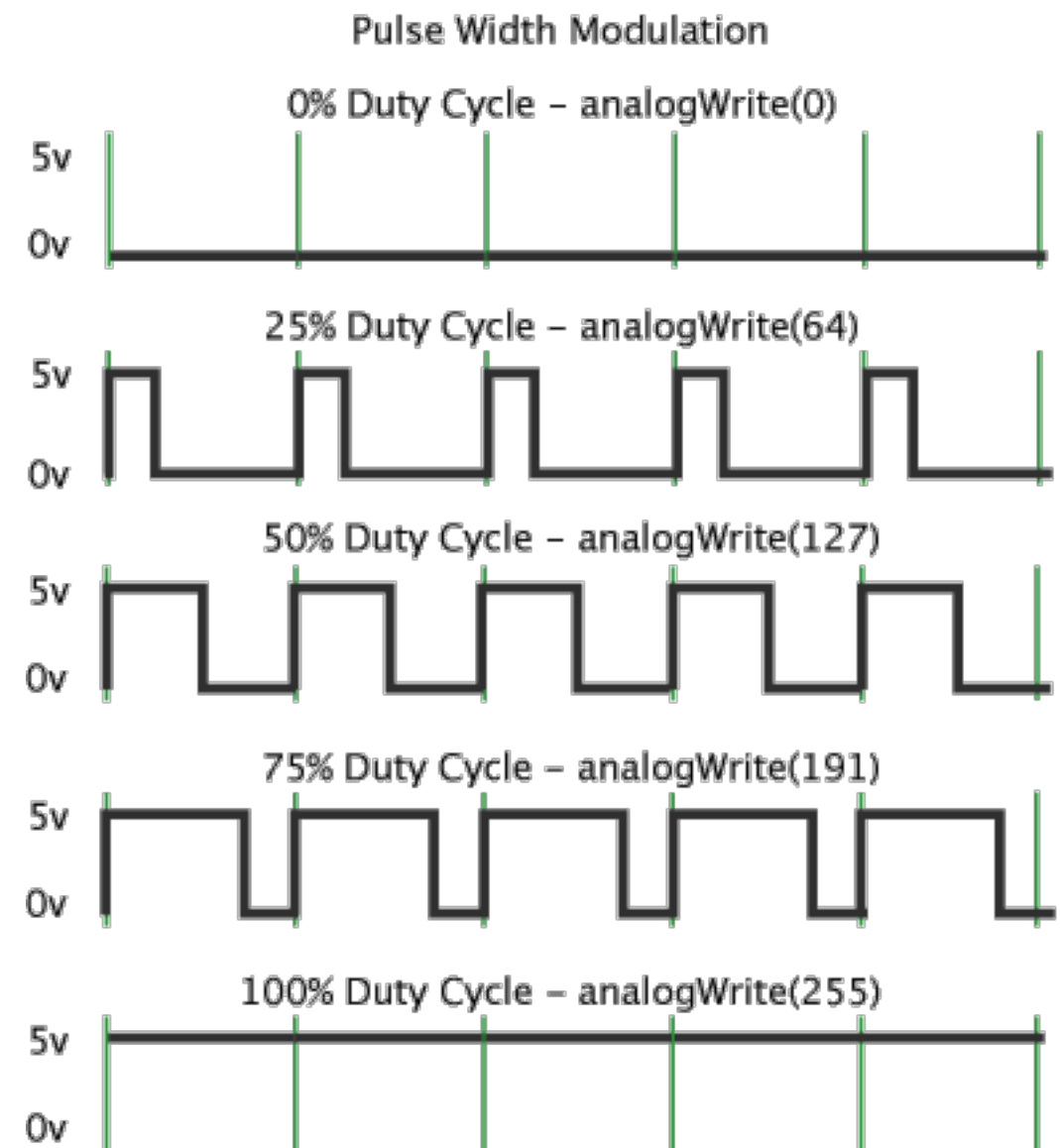


PWM

我要怎麼控制亮度？

PWM (Pulse Width Modulation)

- 用數位輸出模擬類比輸出
- 迅速的開關，用通電的時間比例控制輸出能量
- Arduino - `analogWrite`



PWM (Pulse Width Modulation)

- 腳位
 - 旁邊有 ~ 符號
 - Arduino UNO: 3, 5, 6, 9, 10, 11
- `analogWrite(pin, value)`
 - 輸入範圍 0(全暗) ~ 255(全亮)
 - Duty cycle: $\text{<input>} / 256 * 100\%$

PWM

```
int led_pin = 9;

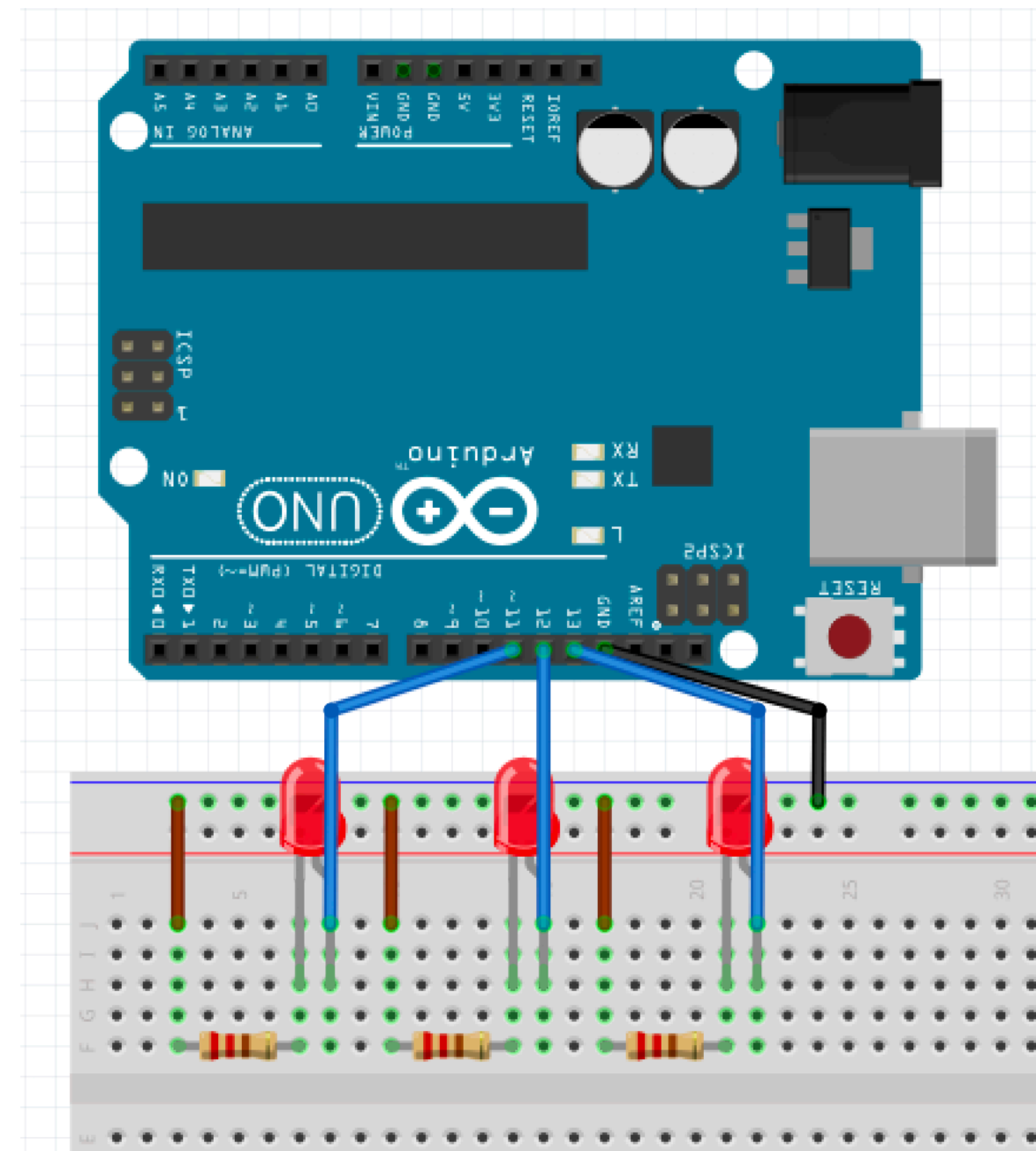
void setup() {
    pinMode(led_pin, OUTPUT);
}

void loop() {
    analogWrite(led_pin, 64);
    delay(1000);
    analogWrite(led_pin, 255);
    delay(1000);
}
```

Lab 04

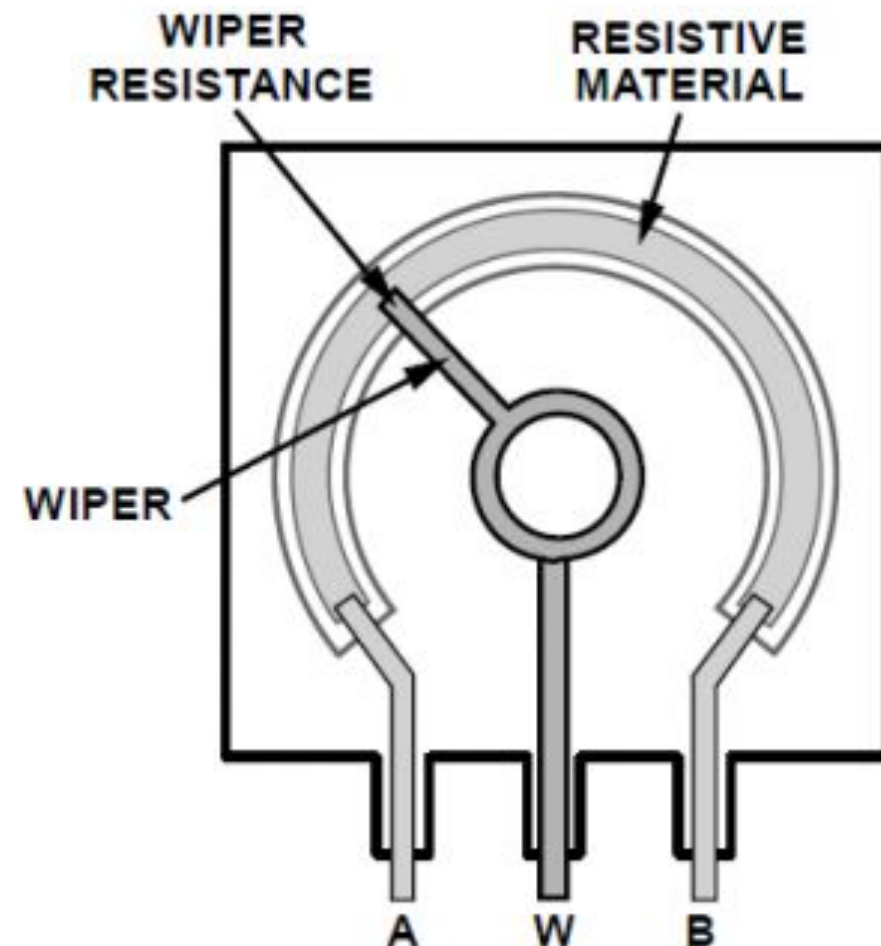
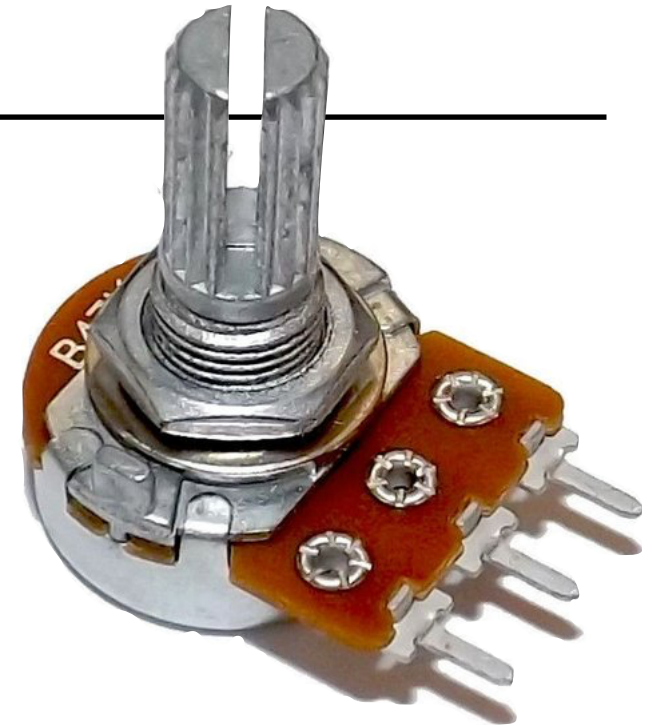
Goal: 呼吸霹靂燈

1. 用程式碼控制LED亮暗
2. 讓LED從暗漸漸變亮
3. 曳尾霹靂燈



可變電阻

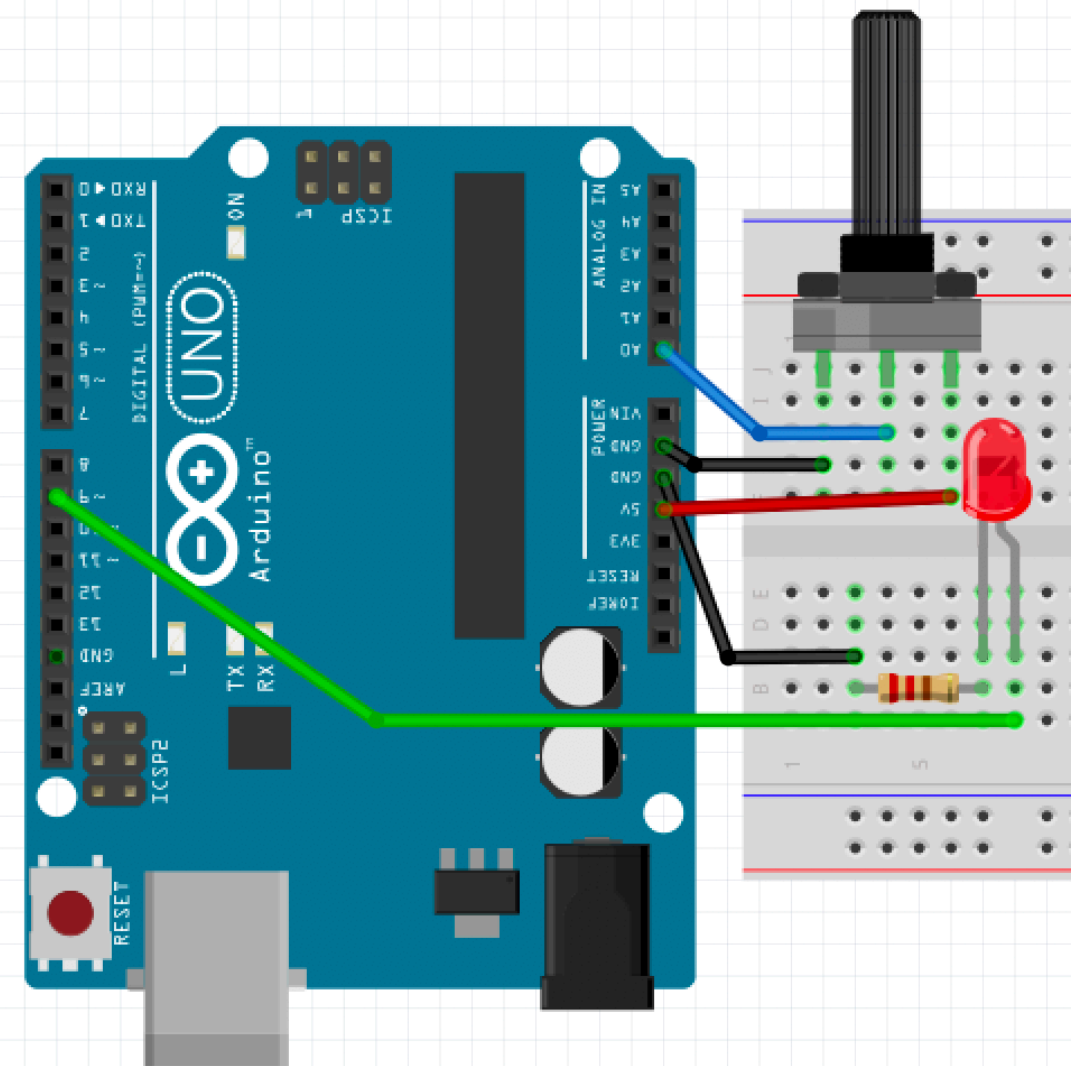
- 兩個固定接點+一個滑動接點
- 透過改變滑動端和固定端間距離改變電阻值
- 用途：
 - 音量控制
 - 位置、距離傳感器



Lab 05

Goal: 可變電阻控制LED

1. 用Arduino讀取可變電阻電壓
2. 輸出對應的PWM訊號控制LED燈亮度
3. bonus: 反向



Lab 05 - code

```
int led_pin = 9;
void setup() {
    pinMode(led_pin, OUTPUT);
}

void loop() {
    int sensorValue = analogRead(A0);

    analogWrite(led_pin, sensorValue);
    delay(150);
}
```

Lab 05 - bonus

Goal: 可變電阻控制霹靂燈速度

