

Arduino 教學手冊

醫工二戴偉璿

May 6, 2025

Contents

1	Arduino 介紹	3
2	Appendix- C++ 語法介紹	4
2.1	Arduino 的輸出	4
2.2	變數的種類與介紹	4
2.3	if-else 語法	5
2.4	for loop	5
2.5	while loop	6

1 Arduino 介紹

平時你在寫程式時，會使用電腦的 CPU 來執行程式碼，而 Arduino 的核心就是一顆微處理器，這顆微處理器可以執行你寫的程式碼，並且控制外部的電子元件，例如 LED 燈、馬達、感測器等等。

Arduino 的開發環境是 IDE，這是一個可以讓你寫程式、編譯程式、上傳程式到 Arduino 開發板的軟體，我們已經幫你們安裝好了，但如果你在家裡想要嘗試，可以去 <https://www.arduino.cc/en/software> 下載最新版本的 IDE。

2 Appendix- C++ 語法介紹

2.1 Arduino 的輸出

平時使用電腦編譯 C++ 程式時，會使用 `cout` 來輸出結果，但在使用 Arduino 時，執行程式的核心不是電腦的 CPU，而是你手上那一塊 Arduino 開發板，所以我們不能使用 `cout` 來輸出結果，因為 Arduino 開發板並沒有螢幕可以顯示結果。而是使用 `Serial` 來輸出結果，這樣 Arduino 開發板就可以透過 USB 線將結果傳送到電腦上，然後電腦就可以使用 `Serial Monitor` 來顯示結果。

這樣的作法等同於僅借用點腦的螢幕來顯示結果，電腦本身不參與運算。

```

1 使用Serial輸出結果
2 Serial.begin(9600); //初始化Serial，9600是傳輸速率
3 Serial.println("Hello World!"); //輸出Hello World!並且換行
4 Serial.print("Hello World!"); //輸出Hello World!不換行
5 Serial.print(123); //輸出123
6 Serial.print(a); //輸出變數a的值
7 如果要串接變數與字串，可以這樣寫：
8 Serial.print("a的值是" + String(a)); //輸出a的值

```

2.2 變數的種類與介紹

類型	大小 (位元組)	範圍	用途
int	4	-2,147,483,648 to 2,147,483,647	整數運算
double	8	$\pm 2.3\text{E}-308$ to $\pm 1.7\text{E}+308$	浮點數運算
char	1	-128 to 127 或 0 to 255	儲存單一字元

Table 1: C++ 常用的三種變數

雖然 C++ 有以上三種常用變數，但在 Arduino 中，我們比較常使用的是 `int` 與 `double`。要注意的是，由於 Arduino 的運算速度較慢，使用 `double` 會影響運算速度。

以下是操縱變數的範例程式碼：

```

1 宣告變數
2 int a=1, b=2;
3 double pi=3.14;
4
5 變數運算
6 int sum=a+b; //計算a+b
7 int mod=a%b; //計算a除以b的餘數
8
9 更改變數的值
10 a=3; //將變數a的值改為3
11 b+=a; //將變數b的值加上變數a的值

```

```

12
13 以下的寫法都是把變數a的值加1
14 a=a+1;
15 a+=1;
16 a++;
17 ++a;

```

2.3 if-else 語法

我們難免會遇到一些情況，必須根據不同的條件來執行不同的程式碼。這時候就需要用到 if-else 語法。

```

1 if-else 範例程式碼
2 if(a>b){
3     //當a大於b時執行的程式碼
4 }else{
5     //當a不大於b時執行的程式碼
6 }

```

在判別時，邏輯運算子也會派上用場，以下是常見的邏輯運算子：

運算子	描述	範例
&&(AND)	當兩個條件都為真時，結果為真	(a > b) && (b > c)
(OR)	當至少一個條件為真時，結果為真	(a > b) (b > c)

Table 2: C++ 邏輯運算子

```

1 邏輯運算子範例程式碼
2 if(a>b && b>c){
3     //當a大於b且b大於c時執行的程式碼
4 }else if(a>b || b>c){
5     //當a大於b或b大於c時執行的程式碼
6 }else{
7     //當a不大於b且b不大於c時執行的程式碼
8 }

```

2.4 for loop

如果要重複很多次執行同樣的程式碼，使用 for loop 會比重複寫一樣的程式碼來得簡單許多。for loop 的語法如下：

```

1 for loop 語法
2 for(初始值; 條件; 更新){
3     //要執行的程式碼
4 }

```

5 舉例而言，假設我們要輸出1到10的數字，我們可以這樣寫：

```
6  
7 for(int i=1;i<=10;i++){  
8     Serial.println(i);  
9 }
```

10 其中，i是初始值，i<=10是條件，i++是更新的方式。

2.5 while loop

在明確知曉執行次數的情況下，使用 for loop 會比較簡單，但在不明確知曉執行次數的情況下，使用 while loop 會比較直觀。while loop 的語法如下：

```
1 while loop 語法  
2 while(條件){  
3     //要執行的程式碼  
4 }
```

5 舉例而言，假設我們要輸出1到10的數字，我們可以這樣寫：

```
6 int i=1;  
7 while(i<=10){  
8     Serial.println(i);  
9     i++;  
10 }
```

11 在這個例子中，我們先宣告一個變數i，然後使用while loop來判斷i是否小於等於10，
12 如果是，就輸出i的值，然後將i的值加1，直到i大於10為止。

以上這是明確知曉執行次數的情況下，使用 while loop 的範例程式碼，明顯會比 for loop 來得繁瑣，但在不明確知曉執行次數的情況下，使用 while loop 會比較直觀。

```
1 String password="arduino123";  
2 String input="";  
3  
4 while(input!=password) {  
5     Serial.println("請輸入密碼：");  
6     while(Serial.available()==0){  
7         //等待使用者輸入  
8     }  
9     input=Serial.readStringUntil('\n');  
10    input.trim();//去除換行符  
11 }  
12 Serial.println("密碼正確，歡迎進入系統！");  
13 在這個例子中，我們使用while loop來判斷使用者輸入的密碼是否正確，  
14 如果不正確，就一直要求使用者輸入，直到輸入正確為止。
```