第2單元

電腦運算思維與資料數位化

電腦的運算思維

電腦與人類有點不同,人類處理 3+2,可說非常直覺,馬上回答 5。但電腦就有點不同了,電腦要進行資料處理,都要先將資料以 變數儲存,然後放到電腦記憶體,接著將記憶體內容放到中央處 理運算單元(簡稱 CPU)運算,CPU 運算結果還是要先以變數為 名稱丢到記憶體,然後再將這些記憶體輸出到螢幕,此即為電腦 的運算思維。本單元則要談電腦運算思維的第一步『資料的數位 化』。也就是資料要數位化,資料才能放到主記憶體,這也是電腦 運算思維的第一步。

其次,電腦和人腦的運算思維有點不同,人腦判斷能力強,運算 能力較差,電腦則是執行判斷較費時,計算能力超強且速度快。 學習程式設計,當然要先學習電腦這些運算思維,這樣電腦才能 精準且快速的完成指定工作。以下將陸續介紹一些電腦運算思 維,這樣寫程式才能事半功倍。

電腦的最小記憶單元為位元(bit),每個位元僅能儲存0或1,8個 位元稱為一個位元組(Byte),那資料如何儲存於電腦呢?答案就 是要將資料數位化。

資料數位化方法

程式設計最常見的資料為數值、字元、字串,將以上資料轉為二 進位的方式,才能儲存於電腦,以上資料的數位化方法,分別說 明如下:

■正整數

所有整數都要先指定資料的長度,本例先假設是 8 位元,其餘 16 位元或 32 位元也是相同原理。其次,還要假設僅儲存正數,還是 同時儲存正負整數。首先,以儲存正數爲例,那其二進位編碼如 下,共可表示 0 到 255 的正整數。

正數	二進位編碼
0	0000000
1	00000001
2	0000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
254	11111110
255	11111111

範例 2a

示範正整數的數位化。請鍵入以下程式,並觀察8個LED的亮 滅,請問與上表的正數是否相符。

🗑 輸出結果

請留意8個LED代表8位元,燈號『亮』就是1,燈號『滅』就是0。

☑ 操作步驟

1. 請準備 LED 電路如下圖。



 若使用麵包板,則實體接線如下圖:(電阻 330Ω 顏色是橙橙 棕,LED 長腳為正,若腳被剪斷過,則往 LED 裡面看,接 點小的為正端,接點大的為負端,下圖 LED 長腳請接電阻, 短腳接地)



2. 若使用本書以下實驗板,只要將 13,12,11,10,50,51,52,53 的 腳位使用杜邦線連接到限流電阻 (J16),再從限流電阻另一端 (J17)連接到 LED (J18)即可,Gnd 內部已自動連接。在電 源指示燈正常情況下,即可進行實驗。其次,因為微控板有 過載保護,若微控板電源指示燈熄了,實驗板指示燈也會熄 減,表示您的電路有問題,有導線短路了,微控板自動切掉 電源,此時請迅速拔掉電源連接線,檢查電路,直到指示燈 正常為止。

3. 杜邦線的顏色有意義,請依 電阻的色碼使用,『棕紅澄 黄綠藍紫灰』分別代表1 到8,『白黑』請用來接正 負電源。



衍程式列印

1. 鍵入以下程式,並觀察輸出結果。

```
void setup() {
 DDRB=B1111111; //指派PORTB功能爲輸出
 Serial.begin(9600);//啓動序列埠
}
unsigned char i=12; //宣告i為8位元正整數,這樣i可儲存0~255的整數
void loop() {
 PORTB=i;
 Serial.println((int)i);//轉為整數
}
```

2. 若使用本書中學生實驗版,此實驗板並未準備以上LED 電 路,而是拿 8*8 點陣 LED 來當作 8 位元 LED。此實驗板內 部接線如下:



(<u>)</u>26

第二單元 電腦運算思維與資料數位化 27

請在以上程式加入以下底線字程式。本書往後使用任何獨 立 LED 的程式也都這樣,直接使用軟體設定的方式,設定 C1 ~ C7 皆為高電位,C8 為低電位,那此 8*8 點陣 LED, C8 Column 就可當作 8 個 LED 來使用。

```
void setup() {
    DDRB=B1111111;//指派PORTB全為輸出
    DDRC=0xFF;PORTC=0xFE;//0xFE is B1111110 將8*8點陣LED當作8個LED使用
    Serial.begin(9600);
}
unsigned char i=12; //宣告i爲8位元正負整數,這樣i可儲存0~255的整數
void loop() {
    PORTB=i;
    Serial.println((int)i);//轉為整數
}
```

 請鍵入以下程式,並觀察序列埠視窗與LED的燈號,驗證0 到255的數位化。

```
void setup() {
    DDRB=B1111111;
    Serial.begin(9600);
    DDRC=0xFF;PORTC=0xFE;//0xFE is B1111110 將點陣LED當作8個LED使用
    unsigned char i=0; //宣告i為8位元正整數,這樣i可儲存0~255的整數
    void loop() {
        PORTB=i; //使用LED顧示記憶體內容
        Serial.println((int)i);//轉 為 整 數 ,使用序列埠輸出記憶體內容
        delay(1000);
        i=(i+1)%256;//保障在0到255循環
    }
```

🗑 自我練習

請自行將112轉為二進位,並觀察結果是否相符。

■正負整數

若要考慮正負整數,那可用最高位元則用來表示正或負數,最高 位元為0時表示正數,所以可表示的範圍是0到127如下表;

二進位編碼	數字
00000000	0
00000001	1
00000010	2
01111110	126
01111111	127

最高位元為1時表示負數,那負數編碼如下:

二進位編碼	數字
10000000	-0
10000001	-1
10000010	-1
11111110	-126
11111111	-127

這樣會有0與-0的問題,0重複出現,造成不是一對一函式, 這樣函數與反函數互轉比較麻煩。所以計算機前輩就繼續腦力激 盪,想出二補數的編碼,且沿用到今天。二補數的編碼如下:正 數就直接編碼,同上;負數就取其2補數。以-3為例,先將3轉 為二進位

00000011

取1補數,1補數是0變1,1變0,所以是

11111100

加1,所以是

11111101

第二單元 電腦運算思維與資料數位化 29

也就是2補數是1補數加1。以上即為-3的2補數編碼,等一下 我用燈號證明給您看,您會看到『亮,亮,亮,亮,亮,亮,滅, 亮』。同理,看到燈號為

11111101

最高位元為1,表示負數,那到底負多少?就取2補數,2補數是 先取1補數(0變1,1變0稱為1補數)再加1。以上

11111101

先取2補數

0000010

加1

00000011

那就是-3。所以1000000代表-128,10000001,代表-127,… 1111111代表-1,以上即為2補數的編碼,如下表所示,這樣就 沒有正負0的問題了。其次,2補數的編碼竟然也同時解決減法 問題,眞是一石二鳥。因為

8-3

可以看成

8 + (-3)

也就是直接將3取2補數,再相加就可以。因為計算機內部組織 只有加法器與比較器,繼續研讀本書,您會發現乘法也是可以用 加法完成,除法也是用加減法完成。

二進位編碼	數字
00000000	0
00000001	1
00000010	2
00000011	3
00000100	4
00000101	5
00000110	6
00000111	7
01111110	126
01111111	127 最大正數
1000000	-128 最小負數
1000001	-127
11111110	-2
11111111	-1

範例 2b

示範包含正負整數的數位化。請鍵入以下程式,並觀察8個 LED的亮滅,請問與上表的正負數是否相符?(電路同範例 2a)

🖓 程式列印

- 以-3 為例,先將3轉為二進位 00000011 取1補數 11111100
 - 加1,所以是
 - 11111101

() 30

第二單元 電腦運算思維與資料數位化 31

 看到11111101,最高位元為1,表示此為負數,到底負多 少,那也是取2補數,先取1補數如下:
 00000010
 再加1,結果是
 00000011
 所以是-3。

3. 請鍵入以下程式,並觀察執行結果。

```
void setup() {
    DDRB=B1111111;
    DDRC=0xFF;PORTC=0xFE;//0xFE is B1111110 將8*8點陣LED當作8個LED使用
    Serial.begin(9600);
}
char i=-3; //宣告i爲8位元正負整數,這樣i可表示-128~127
void loop() {
    PORTB=i;
    Serial.println((int)i);//轉爲整數
}
```

🗑 補充說明

 本例 Serial.println((int)i);(int) 為將字元型態轉為整數,請修 改為 Serial.println(i),並觀察執行結果。

☆ 自我練習

請自行將 -112 轉為二補數,並觀察結果是否正確。

■字元

鍵盤能用的字元有大小寫的 a,b,c、數字 0 ~ 9、還有一些控制字 元,例如,跳列、歸位(回到該列最左邊)、跳頁、return等, 這些字元總共沒有超過 127 個,因此使用 1 個 byte 儲存就綽綽 有餘。所以這些字元的編碼,在 1960 年就已經編碼完成,稱為 ASCII 碼 (American Standard Code for Information Interchange), (合) 32 Arduino程式設計

如下圖所示:也就是我們將所有字元編號,此編號與我們的座號 相同,當叫到65號,大家都能體認此爲字元『A』,當叫到33 號,就表示此爲字元『!』。

DEC	Character	DEC	Character	DEC	Character	DEC	Character
Value		Value		Value		Value	
0	null	32	space	64	0	96	•
1		33	İ	65	A	97	а
2		34		66	В	98	b
3		35	#	67	С	99	С
4		36	\$	68	D	100	d
5		37	%	69	E	101	e
6		38	&	70	F	102	f
7		39	1	71	G	103	g
8		40	(72	Н	104	h
9	tab	41)	73	I	105	i
10	line feed	42	*	74	J	106	j
11		43	+	75	К	107	k
12		44	,	76	L	108	1
13	carriage return	45	-	77	М	109	m
14		46		78	N	110	n
15		47	1	79	0	111	0
16		48	0	80	Р	112	р
17		49	1	81	Q	113	q
18		50	2	82	R	114	r
19		51	3	83	S	115	s
20		52	4	84	Т	116	t
21		53	5	85	U	117	u
22		54	6	86	V	118	V

由上圖可知共使用7位元,0到127,前面32(0~31)為控制字元,48~57是0~9的數字,65~90是大寫英文字元, 97~122是小寫英文字元。

範例 2c

示範英文小寫字元的表示。請自行鍵入以下程式,並觀察燈號 與輸出結果。(電路同範例 2a)

🗑 輸出結果

 請留意字元 'a' 的表示,且內部是以 (97)10=(61)16 表示,燈 號的明滅就是其二進位,本例會是『滅,亮,亮,滅,滅, 滅,滅,亮』,代表 01100001。

```
void setup() {
    DDRB=B1111111;
    Serial.begin(9600);
    DDRC=0xFF;PORTC=0xFE;//0xFE is B1111110 將8*8點陣LED當作8個LED使用
}
char i ='a';//字元用單引號 ;
void loop() {
    Serial.print((int)i);Serial.print(":");Serial.println(i);
    PORTB=i;//97,0x61
    delay(1000);
}
```

₩ 自我練習

- 1. 請自行於 Arduino 線上手冊搜尋『ASCII』。
- 2. 請將 a 改為大寫 A, 並觀察記憶體內容, 塡入下表:

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

3、請將 char i='a' 改為 char i=1,並觀察記憶體內容,填入下表。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

4、請將 char i='a' 改為 char i='1',並觀察記憶體內容,填入下 表,請問是否與 ACCII 的符號 1 相同。

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0

■字串

連續字元的集合稱為字串,例如,

"ABC"

(二) 34 Arduino程式設計

請留意前面字元用單引號,本例字串則用雙引號『"』。C語言使 用字元陣列表示字串。例如,以下陣列a即可儲存以上字串。

char a[]="ABC";//字串用雙引號

變數經過以上宣告,往後就可以使用索引存取內部字元。請鍵入 以下程式,並開啓序列埠視窗,觀察執行結果。

```
void setup() {
   Serial.begin(9600);
   char a[]="ABC";
   Serial.println(a);//ABC
   Serial.println(a[0]); //A
   Serial.println(a[1]); //B
   Serial.println(a[2]);//C
   a[0]='D';
   Serial.println(a);//DBC
}void loop() {}
```

C++ 則新增字串型態 string,例如,

```
string a="ABC";//字串用雙引號
```

變數經過以上宣告,往後就可以使用索引存取內部字元。請鍵入 以下程式,並開啓序列埠視窗,觀察執行結果,就會明瞭。

```
void setup() {
  Serial.begin(9600);
  String a="ABC";
  Serial.println(a);//ABC
  Serial.println(a[0]); //A
  Serial.println(a[1]); //B
  Serial.println(a[2]);//C
  a[0]='D';
  Serial.println(a);//DBC
}void loop() {}
```

以上兩種字串表示, Arduino 都可以接受, 如以上程式。

範例 2d

示範字串的表示。請自行鍵入以下程式,並觀察燈號與輸出結果。(電路同範例 2a)

🗑 輸出結果

- 請留意 a[0] 是字元'A',編號是 65,內部記憶體是 01000001(=0x41), 燈號會是 01000001。
- 2. a[1] 是字元 'B',編號是 66, a[2] 是字元 'C',編號是 67。

65:A		
66:B		
67:C		

🗑 程式列印

```
void setup() {
 DDRB=B11111111;
 DDRC=0xFF; PORTC=0xFE; //0xFE is B11111110 將8*8點陣LED當作8個LED使用
 Serial.begin(9600);
}
void loop() {
 //char a[]="ABC";//C語言的字串宣告方式
 String a="ABC";//C++語言的字串宣告方式,Arduino两者都可以接受
 Serial.print((int)a[0]);Serial.print(":");
 Serial.println(a[0]);
 PORTB=a[0];//65,0x41
 delay(1000);
 Serial.print((int)a[1]);Serial.print(":");
 Serial.println(a[1]);
 PORTB=a[1];//66,0x42
 delay(1000);
 Serial.print((int)a[2]);Serial.print(":");
 Serial.println(a[2]);
 PORTB=a[2];//67,0x43
 delay(12000);
```