

4

CHAPTER

通用輸入/輸出控制實習

本章單元

- GPIO 控制
- GPIO 基礎實習-LED、三色 LED、按鍵、蜂鳴器
- GPIO 應用實習-紅黃綠燈、繼電器、步進馬達、七段顯示器、LCM 顯示器

MG32x02z 系列的通用輸入/輸出(GPIO: General Purpose Input Output)接腳，同時兼具各項特殊功能。以 LQFP64 包裝的編號 MG32F02U128AD64 為例共有 59 支 GPIO 腳(PA0~15、PB0~15、PC0~14、PD0~11)，其中在 TH244A 開發板可提供使用者的接腳有 47 支(D0~46)，如下圖及下表所示：

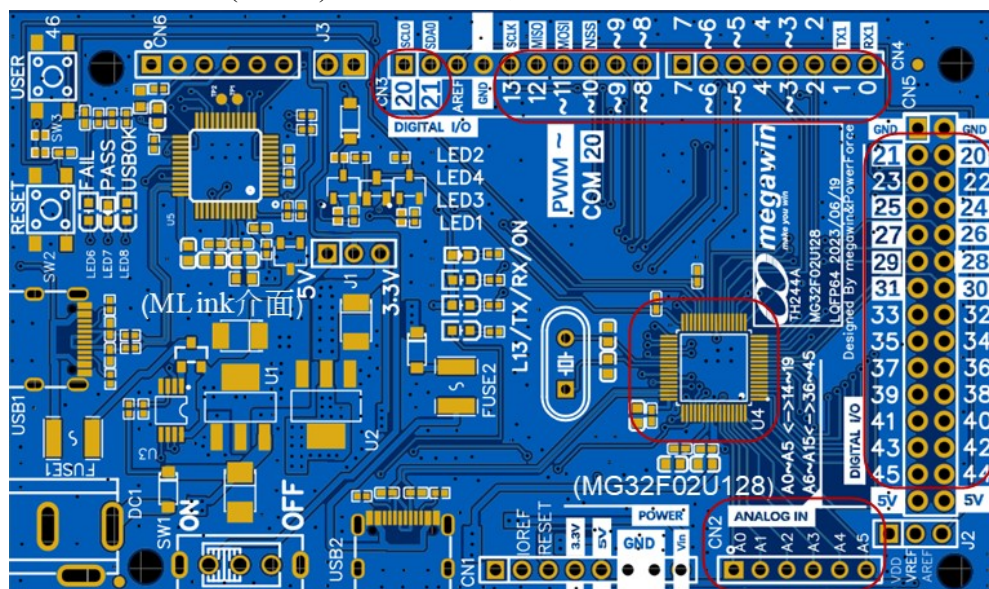


圖 4-1 TH244A 開發板 GPIO 接腳

表 4-1 MG32F02U128AD64 與 TH244A 開發板接腳

編號	MCU 接腳編號	MCU 接腳名稱	開發板接腳名稱
1	35	PC9/RXD1	D0/0
2	34	PC8/TXD1	D1/1
3	14	SD3/PB4	D2/2
4	20	PB10	D3/3
5	33	PC7	D4/4
6	21	PB11	D5/5
7	26	PC0/ICKO	D6/6
8	15	PB5/SD2	D7/7
9	38	PC12	D8/8
10	42	PD0	D9/9
11	51	PD9/NSS	D10/10

12	44	PD2/MOSI	D11/11
13	49	PD7/MISO	D12/12
14	50	PD8/SPI0_CLK	D13/13
15	58	PA0/ADC0	D14/A0/14
16	59	PA1/ADC1	D15/A1/15
17	60	PA2/ADC2	D16/A2/16
18	61	PA3/ADC3	D17/A3/17
19	62	PA4/ADC4	D18/A4/18
20	63	PA5/ADC5	D19/A5/19
21	13	PB3/MOSI/SCL0/RX7	D20/20
22	12	PB2/SCK/SDA0/TX7	D21/DAC/21
23	37	PC11/SDA1	D22/22
24	36	PC10/SCL1	D23/23
25	17	PB7/RX2	D24/24
26	16	PB6/TX2	D25/25
28	23	PB13/RX4	D26/26
27	24	PB14/TX4	D27/27
29	25	PB15/RX5	D28/28
30	22	PB12/TX5	D29/29
32	11	PB1/MISO/RX6	D30/30
31	10	PB0/NSS/TX6	D31/31
33	43	PD1	D32/32
34	45	PD3	D33/33
35	52	PD10	D34/34
36	53	PD11	D35/35
37	64	PA6/ADC6	D36/A6/36
38	1	PA7/ADC7	D37/A7/37
39	2	PA8/ADC8	D38/A8/38
40	3	PA9/ADC9	D39/A9/39
41	4	PA10/ADC10	D40/A10/40
42	5	PA11/ADC11	D41/A11/41
43	6	PA12/ADC12	D42/A12/42
44	7	PA13/ADC13	D43/A13/43
45	8	PA14/ADC14	D44/A14/44
46	9	PA15/ADC15	D45/A15/45
47	27	PC1	D46/USER KEY/46

開發板硬體特性如下表所示：

表 4-2 TH244A 開發的硬體特性

工作電壓	工作頻率	數位 IO
5V /3.3V 可切換	IHRCO 12MHz MCU 主頻 36MHz	使用 47 腳(D0~D46)

TH244A 開發板的數位 I/O 接腳有 47 支(D0~D46)，可設定的數位接腳名稱有 0~13、14~19(A0~5)、20~35、36~45(A6~15)、46。其中接腳 13 連接 L13(LED) 及接腳 46 連接按鍵 SW3(USER)，方便測試之用。

GPIO 接腳可工作於 5V 或 3.3V，接腳輸出電流約 40mA(5V)/13mA(3.3V)，其各項電壓及電流的工作範圍，如下表(a)(b)所示，詳細內容請看資料手冊。

表 4-3(a) GPIO 電氣特性($V_{DD}=5V$)

名稱	說明($V_{DD}=5V$ ， $V_{SS}=0V$ ， $TA=25^{\circ}C$ 時)	最小	一般	最大	單位
V_{IH}	輸入 1 時的電壓(RSTN,XIN 及 XOUT 腳除外)	3			V
V_{IL}	輸入 0 時的電壓(RSTN,XIN 及 XOUT 腳除外)			0.75	V
I_{IH}	輸入 1 電流, $V_{PIN}=V_{DD}$ 時		0.05	0.1	uA
I_{IL}	輸入 0 電流(準雙向,輸入模式,含提升電阻)		0.01	0.1	uA
I_{H2L}	輸入 1→0 電流(準雙向,輸入模式,含提升電阻)		320	500	uA
I_{OH1}	輸出 1 電流, $V_{PIN}=2.4V$ (推挽式,全準位)		38.5		mA
I_{OH2}	輸出 1 時電流, $V_{PIN}=2.4V$ (推挽式,1/2 準位)		19.8		mA
I_{OH3}	輸出 1 時電流, $V_{PIN}=2.4V$ (推挽式,1/4 準位)		10.1		mA
I_{OH4}	輸出 1 時電流, $V_{PIN}=2.4V$ (推挽式,1/8 準位)		5.2		mA
I_{OL1}	輸出 0 時電流, $V_{PIN}=0.4V$ (全準位)		30.4		mA
I_{OL2}	輸出 0 時電流, $V_{PIN}=0.4V$ (1/2 準位)		15.7		mA
I_{OL3}	輸出 0 時電流, $V_{PIN}=0.4V$ (1/4 準位)		8.0		mA
I_{OL4}	輸出 0 時電流, $V_{PIN}=0.4V$ (1/8 準位)		4.0		mA

表 4-3(b) GPIO 電氣特性($V_{DD}=3.3V$)

名稱	說明($V_{DD}=3.3V$, $V_{SS}=0V$, $T_A=25\text{ }^{\circ}C$ 時)	最小	一般	最大	單位
V_{IH}	輸入 1 時的電壓(除了 RSTN, XIN 及 XOUT 腳)	1.98			V
V_{IL}	輸入 0 時的電壓(除了 RSTN, XIN 及 XOUT 腳)			0.5	V
I_{IH}	輸入 1 電流, $V_{PIN}=V_{DD}$		0.02	0.1	μA
I_{IL}	輸入 0 電流(準雙向, 輸入模式, 含提升電阻)		0.01	0.1	μA
I_{H2L}	輸入 1 \rightarrow 0 電流(準雙向, 輸入模式, 含提升電阻)		115	150	μA
I_{OH1}	輸出 1 電流, $V_{PIN}=2.4V$ (推挽式, 全準位)		13		mA
I_{OH2}	輸出 1 時電流, $V_{PIN}=2.4V$ (推挽式, 1/2 準位)		6.5		mA
I_{OH3}	輸出 1 時電流, $V_{PIN}=2.4V$ (推挽式, 1/4 準位)		3.5		mA
I_{OH4}	輸出 1 時電流, $V_{PIN}=2.4V$ (推挽式, 1/8 準位)		1.7		mA
I_{OL1}	輸出 0 時電流, $V_{PIN}=0.4V$ (全準位)		22		mA
I_{OL2}	輸出 0 時電流, $V_{PIN}=0.4V$ (1/2 準位)		11.3		mA
I_{OL3}	輸出 0 時電流, $V_{PIN}=0.4V$ (1/4 準位)		5.6		mA
I_{OL4}	輸出 0 時電流, $V_{PIN}=0.4V$ (1/8 準位)		2.8		mA

4-1 GPIO 控制與基礎實習

本章包括 GPIO 控制暫存器(初學者可省略)及基礎實習，其中控制暫存器的介紹初學者可省略。

4-1.1 GPIO 控制暫存器(初學者可省略)

在 Arduino IDE 沒有並列輸出入指令，但可使用底層的控制暫存器來彌補，MG32x02z 系列中的每個 GPIO 埠均為 32-bit 控制暫存器，存取方式有：字元(word)W=32-bit、半字元(half word)H[0~1]=16-bit、位元組(byte)B[0~3]=8-bit 及位元 PX(0~15)=1-bit。

以 MG32F02U128AD64 為例，有 GPIO 腳 PA~PD(GPIOA~D)及四組 GPIO 控制暫存器。所有 GPIO 控制暫存器指令(x=A~D)，如下：

GPIOx->OUT	GPIOx->IN	GPIOx->SC	GPIOx->SCR0~3	GPIOx->CR0~15
------------	-----------	-----------	---------------	---------------

1. 位元輸出入指令：

```
PD8=1;    //令 LED(D13/PD8)=1，LED 亮

PD8=0;    //令 LED(D13/PD8)=0，LED 暗

if(PC1==0) ..... // 檢查 USER KEY(D46/PC1)是否有按鍵?
```

2. Px 輸出資料暫存器(Px_OUT)：Reset= 0x0000_FFFF，Px=GPIOA~E。

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Px_OUT15	Px_OUT14	Px_OUT13	Px_OUT12	Px_OUT11	Px_OUT10	Px_OUT9	Px_OUT8
7	6	5	4	3	2	1	0
Px_OUT7	Px_OUT6	Px_OUT5	Px_OUT4	Px_OUT3	Px_OUT2	Px_OUT1	Px_OUT0

半字元設定，例如：GPIOA->OUT.H[0]=0x1234，令 PA0~15=0x1234。

位元組設定，例如：GPIOA->OUT.B[0]=0x34，令 PA0~7=0x34。

位元組設定，例如：GPIOA->OUT.B[1]=0x12，令 PA8~15=0x12。

3. Px 輸入資料暫存器(Px_IN)：Reset= 0x0000_0000，Px=GPIOA~E。

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Px_IN15	Px_IN14	Px_IN13	Px_IN12	Px_IN11	Px_IN10	Px_IN9	Px_IN8
7	6	5	4	3	2	1	0
Px_IN7	Px_IN6	Px_IN5	Px_IN4	Px_IN3	Px_IN2	Px_IN1	Px_IN0

半字元輸入，例如：temp=GPIOA->IN.H[0]，讀取 PA0~15 資料存入變數 temp。

位元組輸入，例如：temp=GPIOA->IN.B[1]，讀取 PA8~15 資料存入 temp。

4. Px 設定/清除資料暫存器(Px_SC)：Reset= 0x0000_0000，Px=GPIOA~E。

31	30	29	28	27	26	25	24
Px_CLR15	Px_CLR14	Px_CLR13	Px_CLR12	Px_CLR11	Px_CLR10	Px_CLR9	Px_CLR8
23	22	21	20	19	18	17	16
Px_CLR7	Px_CLR6	Px_CLR5	Px_CLR4	Px_CLR3	Px_CLR2	Px_CLR1	Px_CLR0
15	14	13	12	11	10	9	8
Px_SET15	Px_SET14	Px_SET13	Px_SET12	Px_SET11	Px_SET10	Px_SET9	Px_SET8
7	6	5	4	3	2	1	0
Px_SET7	Px_SET6	Px_SET5	Px_SET4	Px_SET3	Px_SET2	Px_SET1	Px_SET0

例如：GPIOD->SC.W=1<<8，令接腳 PD8=1，其餘不變。

GPIOD->SC.W=1<<8<<16，令接腳 PD8=0，其餘不變。

5. Px 埠設定/清除資料暫存器 0(Px_SCR0)：位元控制，Reset=0x0000_0000。

31	30	29	28	27	26	25	24
Reserved							Px_SC3
23	22	21	20	19	18	17	16
Reserved							Px_SC2
15	14	13	12	11	10	9	8
Reserved							Px_SC1
7	6	5	4	3	2	1	0
Reserved							Px_SC0

例如：GPIOA->SCR0.B[0]=1 令 PA0=1 及 GPIOA->SCR0.B[0]=0 令 PA0=0。

flag = GPIOA->SCR0.B[3]讀取接腳 PA3 的資料存入變數 flag。

6. Px 埠設定/清除資料暫存器 1(Px_SCR1)：位元控制，Reset= 0x0000_0000。

31	30	29	28	27	26	25	24
Reserved							Px_SC7
23	22	21	20	19	18	17	16
Reserved							Px_SC6
15	14	13	12	11	10	9	8
Reserved							Px_SC5
7	6	5	4	3	2	1	0
Reserved							Px_SC4

例如：GPIOA->SCR1.B[0]=1 令 PA4=1 及 GPIOA->SCR1.B[0]=0 令 PA4=0。

flag = GPIOA->SCR1.B[3]讀取接腳 PA7 的資料存入變數 flag。

7. Px 埠設定/清除資料暫存器 2(Px_SCR2)：位元控制，Reset=0x0000_0000。

31	30	29	28	27	26	25	24
Reserved							Px_SC11
23	22	21	20	19	18	17	16
Reserved							Px_SC10
15	14	13	12	11	10	9	8
Reserved							Px_SC9
7	6	5	4	3	2	1	0
Reserved							Px_SC8

例如：GPIOD->SCR2.B[0]=1 令 PD8=1 及 GPIOD->SCR2.B[0]=0 令 PD8=0。
flag = GPIOA->SCR2.B[3]讀取接腳 PA11 的資料存入變數 flag。

8. Px 埠設定及清除暫存器 3(Px_SCR3)：位元控制，Reset=0x0000_0000。

31	30	29	28	27	26	25	24
Reserved							Px_SC15
23	22	21	20	19	18	17	16
Reserved							Px_SC14
15	14	13	12	11	10	9	8
Reserved							Px_SC13
7	6	5	4	3	2	1	0
Reserved							Px_SC12

例如：GPIOA->SCR3.B[0]=1 令 PA12=1 及 GPIOA->SCR3.B[0]=0 令 PA12=0。
flag = GPIOA->SCR3.B[3]讀取接腳 PA15 的資料存入變數 flag。
輸出入操作模式控制，如下表所示：

表 4-4 輸出入操作模式控制

IO Mode	Pull High	High Speed	IO Configuration	
Px_IOMn	Px_PUn	Px_HSn		
0x0	-	-	Analog IO	
0x1	0	0	Open Drain Output	
	0	1		High Speed
	1	0		Pull High
	1	1		Pull High + High Speed
0x2	0	0	Push Pull Output	
	0	1		High Speed
	1	0		Pull High
	1	1		Pull High + High Speed
0x3	0	-	Digital Input	
	1	-		Pull High
0x4	0	0	Quasi-Bidirectional Output (drive high one clock)	
	0	1		High Speed
	1	0		Pull High
	1	1		Pull High + High Speed

Note : "Px" = {PA,PB,PC,PD,PE}, "n" = Pin index number, "-" = Don't care

4-1.2 GPIO 基礎實習

GPIO 基礎實習包括 LED、按鍵、三色 LED 及蜂鳴器，如下圖(a)~(c)所示：

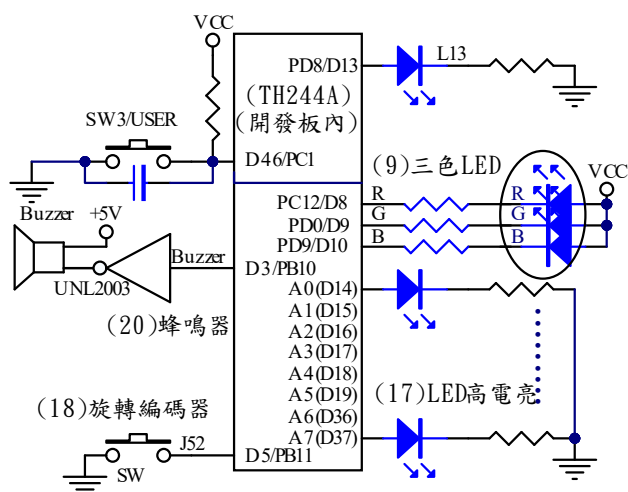


圖 4-2(a) TH244A 基礎實習電路

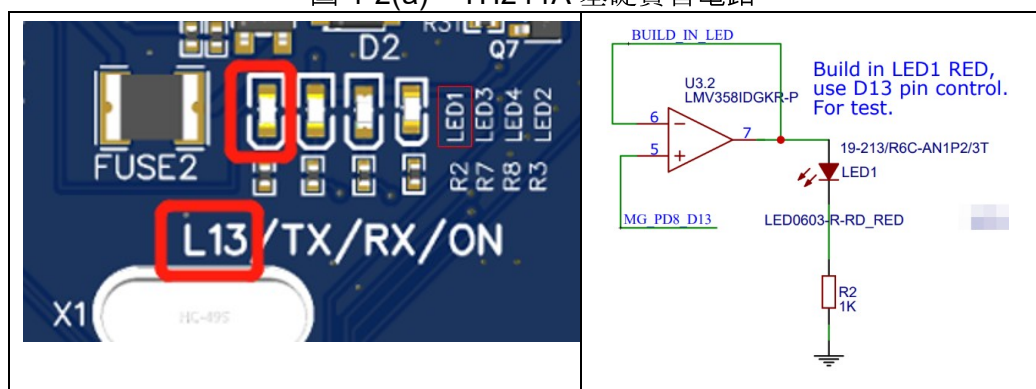


圖 4-2(b) TH244A 的按鍵開關外型及電路

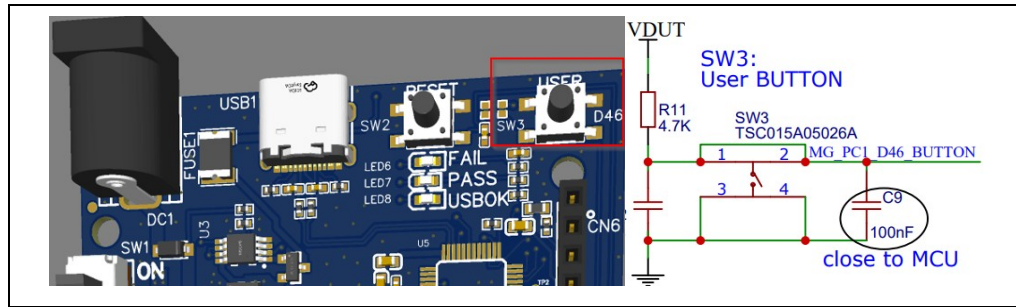


圖 4-2(c) USER 按鍵開關外型與電路

實習板電源可由 USB(M13)提供 5V，再將 J34(+5V, VCC)短路，此時兩片电路板的 GND 相連接即可。若沒有使用馬達，也可以由開發板 TH244A 提供(5V、GND)。如下圖所示：

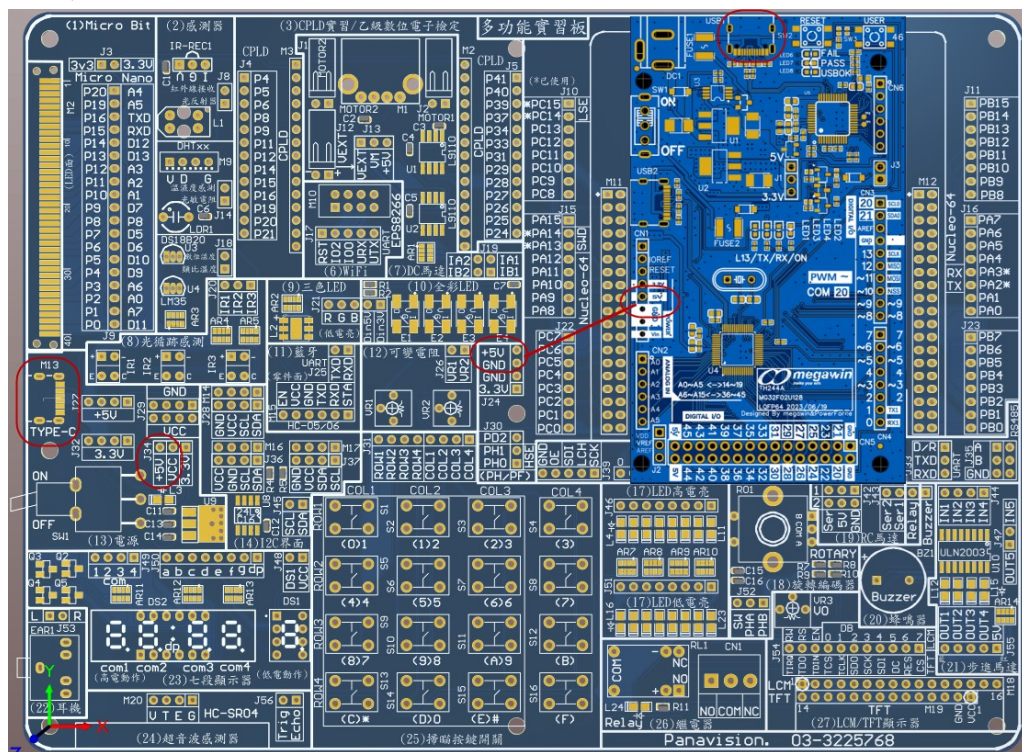


圖 4-3 實習板的電源提供

實習板的 LED、蜂鳴器、旋轉編碼器按鍵與三色 LED 實習。如下圖所示：

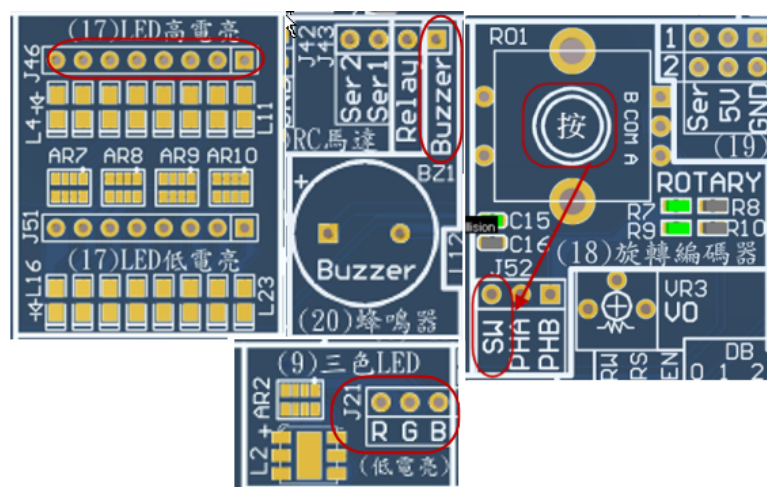


圖 4-4 LED、蜂鳴器、旋轉編碼器按鍵與三色 LED 外型

在 Arduino 輸出入控制實習可使用函數式，如下表(a)~(c)所示：

表 4-5(a) 輸出入(I/O)函數式

數位輸出入	語法	常數
pinMode() digitalWrite() digitalRead()	pinMode(pin, mode) digitalWrite(pin, value) digitalRead(pin)	pin: 0~13、14~19(A0~5)、20~46、36~45(A6~15)、46 mode: OUTPUT、INPUT、INPUT_PULLUP、 INPUT_PULLDOWN value: 1、0、HIGH、LOW

表 4-5(b) 常用計時(Time)函數式

延時函數	語法	常數
delay()	delay(ms)	延時 ms = 1~4,294,967,295
delayMicroseconds()	delayMicroseconds(us)	延時 us = 1~4,294,967,295

表 4-5(c) 音頻(Tone)函數式

音頻函數	語法	常數
tone() noTone()	tone(pin, frequency, duration) noTone(pin)	pin: 3、6、5、8、9、10、11 frequency=1~65535Hz, duration=持續 ms

1. LED 輸出控制實習：

- (1) 範例 LED1：令 LED(D13/PD8)閃爍。
- (2) 範例 LED1A：使用底層暫存器控制，令 LED(D13/PD8)閃爍。
- (3) 範例 LED1B：使用底層暫存器控制，令 LED(D13/PD8)閃爍。
- (4) 範例 LED1C：使用底層暫存器控制，令 LED(D13/PD8)閃爍。
- (5) 範例 LED2：令 LED 埠跑馬燈輸出。
- (6) 範例 LED2 A：使用暫存器令 LED 埠跑馬燈輸出。
- (7) 範例 LED3：令 LED 埠跑馬燈輸出。
- (8) 範例 LED4：令 LED 埠跑馬燈輸出。
- (9) 範例 LED5：使用控制暫存器令 LED 遞加輸出。
- (10) 範例 LED6：使用組態設定令 LED 遞加輸出。

2. 按鍵控制實習：按鍵控制具防止機械彈跳及接腳設定，如下圖所示：

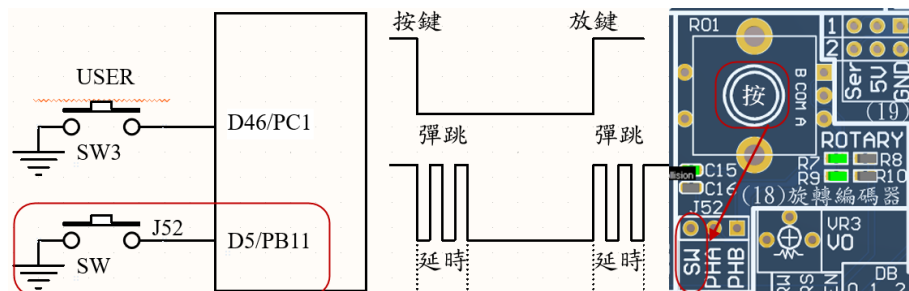


圖 4-5 按鍵機械彈跳與按鍵開關(SW)外型

- (1) 範例 KEY1：令 LED 遞加輸出，按鍵 USER(D46/PC1)則暫停。
- (2) 範例 KEY2：由 USER(D46/PC1)按鍵開關輸入，令 LED 遞加輸出，有機械彈跳若加上延時 delay(2)則會改善。

- (3) 範例 KEY3：由按鍵開關 SW 令 LED 遞加輸出，有機械彈跳，若加上延時 delay(2)則會改善。
- (4) 範例 KEY4：由按鍵開關 SW 令 LED 遞加輸出，可防止機械彈跳。
- (5) 範例 KEY5：每按一次按鍵 USER，控制 LED 遞加輸出或停止。
- (6) 範例 KEY6：每按一次按鍵 USER，超過 1 秒以上才令 LED 遞加。

3. 三色 LED 輸出範例及接線如下：

接線：D8~10(PC12、PD0、PD9)-->J21(R、G、B)

- (1) 範例 RGB1：令三色 LED 輸出不同彩色。
- (2) 範例 RGB2：使用控制暫存器，令三色 LED 輸出不同彩色。

4. 蜂鳴器輸出範例及接線如下：

接線：D3-->J43(Buzzer)

- (1) 範例 Buzzer1：按鍵令 Buzzer 輸出嗶嗶聲。
- (2) 範例 Buzzer2：使用 tone 函數令蜂鳴器輸出嗶嗶聲。
- (3) 範例 Buzzer3：蜂鳴器輸出嗶嗶聲，按下 USER 鍵時，蜂鳴器停止。
- (4) 範例 Buzzer4：由按鍵控制蜂鳴器音頻，按 KEY1 增加音頻，按 KEY2 減少音頻。

4-2 GPIO應用實習

本章應用實習包括：紅黃綠燈、步進馬達、七段顯示器、LCM 顯示器，如下：

4-2.1 紅黃綠燈輸出控制實習

由列表取出資料及時間值，輸出紅黃綠燈的變化，如下圖及下表所示。

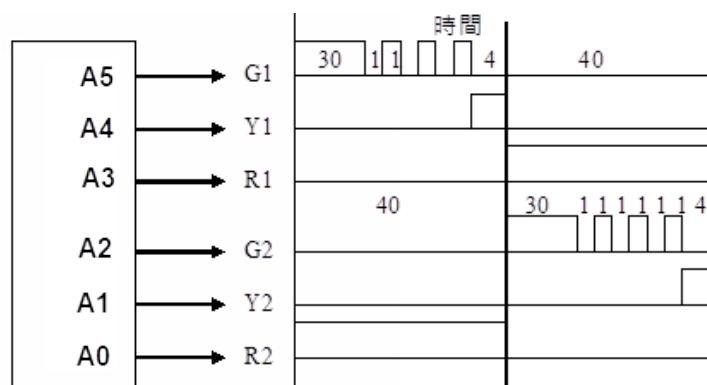


圖 4-6 紅黃綠燈變化時序圖

表 4-6 紅黃綠燈的輸出資料

順序	LED	7 6 5 4 3 2 1 0	時間	順序	LED	7 6 5 4 3 2 1 0	時間
	資料	綠黃紅綠黃紅			資料	綠黃紅綠黃紅	
0	0x21	0 0 1 0 0 0 0 1	30	8	0x0C	0 0 0 0 1 1 0 0	30
1	0x01	0 0 0 0 0 0 0 1	1	9	0x08	0 0 0 0 1 0 0 0	1
2	0x21	0 0 1 0 0 0 0 1	1	10	0x0C	0 0 0 0 1 1 0 0	1
3	0x01	0 0 0 0 0 0 0 1	1	11	0x08	0 0 0 0 1 0 0 0	1
4	0x21	0 0 1 0 0 0 0 1	1	12	0x0C	0 0 0 0 1 1 0 0	1
5	0x01	0 0 0 0 0 0 0 1	1	13	0x08	0 0 0 0 1 0 0 0	1
6	0x21	0 0 1 0 0 0 0 1	1	14	0x0C	0 0 0 0 1 1 0 0	1
7	0x11	0 0 0 1 0 0 0 1	4	15	0x0A	0 0 0 0 1 0 1 0	4

紅黃綠燈 LED 範例及接線，如下：

操作：A0~5(PA0~5)-->J46(LED 高電亮)，按 USER(D46)鍵

1. 範例 RYG1：令 LED 輸出紅黃綠燈的變化。
2. 範例 RYG2：令 LED 輸出紅黃綠燈的變化。
3. 範例 RYG3：令 LED 輸出紅黃綠燈變化，按 USER 鍵時黃燈互閃。

5-2.2 繼電器控制實習

繼電器(Relay)常用於大電力控制，可以將 MCU 輸出的電氣信號轉換為機械開關動作，在 J43(Relay)輸入高電位，會令 LED(L24)亮及繼電器動作，令共接點/常開點(COM/NO)導通及共接點/常閉點(COM/NC)斷路。實習電路如下圖(a)(b)：

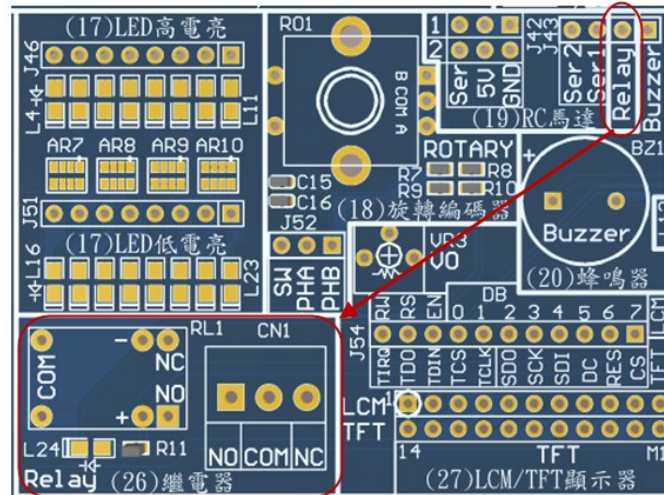


圖 4-7(a) 繼電器實習外型

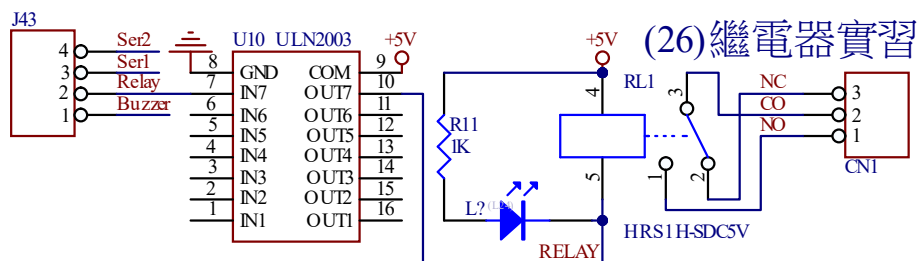


圖 4-7(b) 繼電器實習電路圖

1. 範例 RELAY1：令繼電器(RELAY)動作。
2. 範例 RELAY2：使用按鍵開關來控制繼電器 ON/OFF。

4-2.3 步進馬達控制實習

步進馬達本身是以寸動(每次走一步)的方式來工作，具有無慣性作用、高精密度的旋轉角度及容易控制等優點，尤其適用於微處理機的控制，雖然速度稍嫌緩慢及力量較小外，不失為一良好控制元件。步進馬達模組控制電路，如下圖(a)(b)所示：

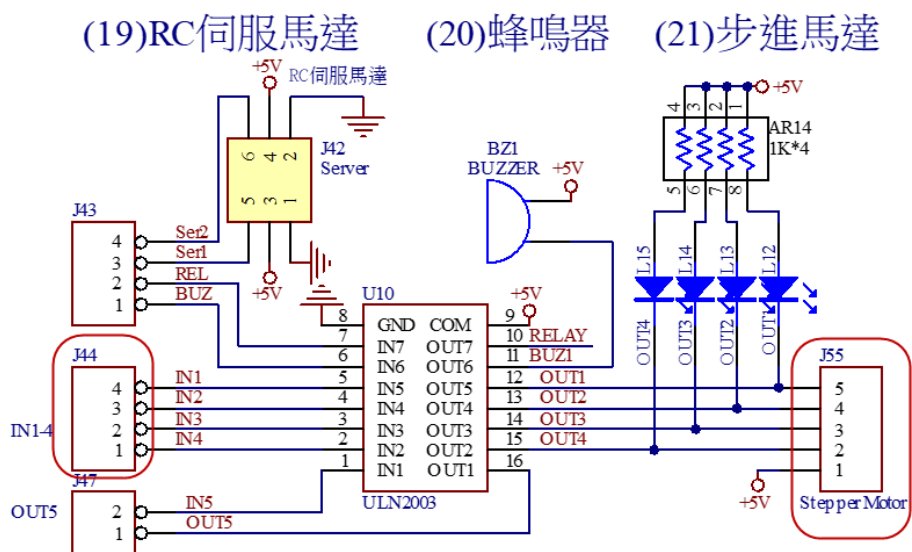


圖 4-8(a) 步進馬控制電路

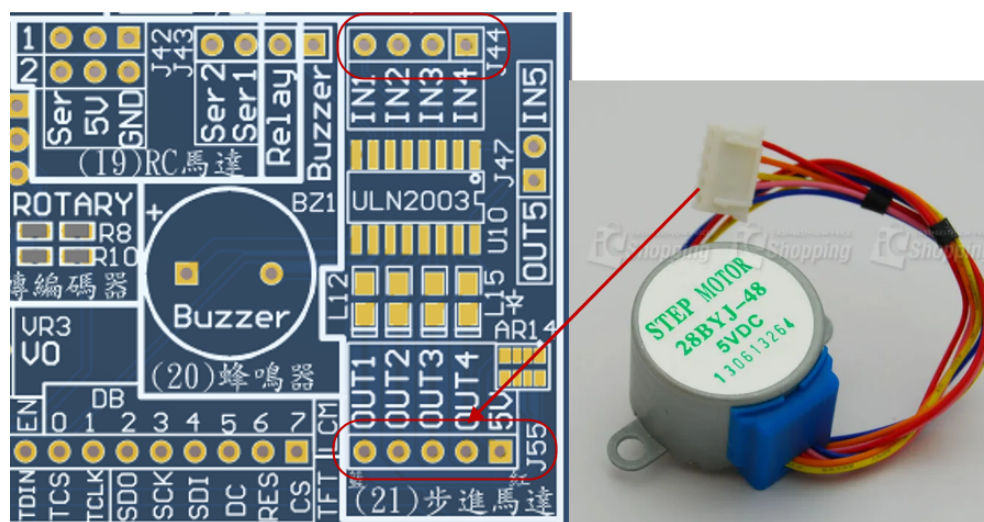


圖 4-8(b) 步進馬達外型

使用時必須在 J55 插入四相五線式步進馬達，開發板於 J44(IN1~4)輸入驅動數碼，經 UNL2003 來驅動步進馬達運轉及令 LED(L12~15)顯示驅動數碼。

四相五線式的步進馬達每圈為 2048 步，其驅動方式一般分為三種：單相全步運轉、雙相全步運轉(扭力較大)及單雙相半步運轉(精度加倍)，只要依照順序去驅動步進馬達的每一組線圈，就可以改變它的正反轉及速度，驅動數碼如下表所示：

表 4-7 步進馬達驅動數碼

正轉	反轉	單相全步運轉					雙相全步運轉					單雙相半步運轉				
		步	A	B	/A	/B	步	A	B	/A	/B	步	A	B	/A	/B
	↑	0	0	0	0	1	0	0	0	1	1	0	1	0	0	1
		1	0	0	1	0	1	0	1	1	0	1	0	0	0	1
		2	0	1	0	0	2	1	1	0	0	2	0	0	1	1
		3	1	0	0	0	3	1	0	0	1	3	0	0	1	0
		0	0	0	0	1	0	0	0	1	1	4	0	1	1	0
		1	0	0	1	0	1	0	1	1	0	5	0	1	0	0
		2	0	1	0	0	2	1	1	0	0	6	1	1	0	0
	↓	3	1	0	0	0	3	1	0	0	1	7	1	0	0	0

步進馬達實習範例及接線如下：

接線：A0~A3-->J44(IN1~IN4)，步進馬達-->J55(OUT1~4, 5V)

1. 範例 STEP1：將驅動數碼存在陣列資料，由 STEP 輸出驅動步進馬達運轉。
2. 範例 STEP2：將驅動數碼存在陣列資料內，以手動方式控制步進馬達運轉。
3. 範例 STEP3：將各種動作存在陣列資料內，以自動方式控制步進馬達運轉。

4-2.4 七段顯示器應用實習

七段顯示器是由 8 個 LED 所組成的，分為共陽極及共陰極，如下圖(a)(b)：

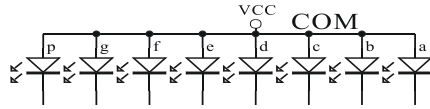


圖 4-9(a) 共陽極七段顯示器

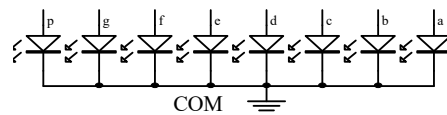
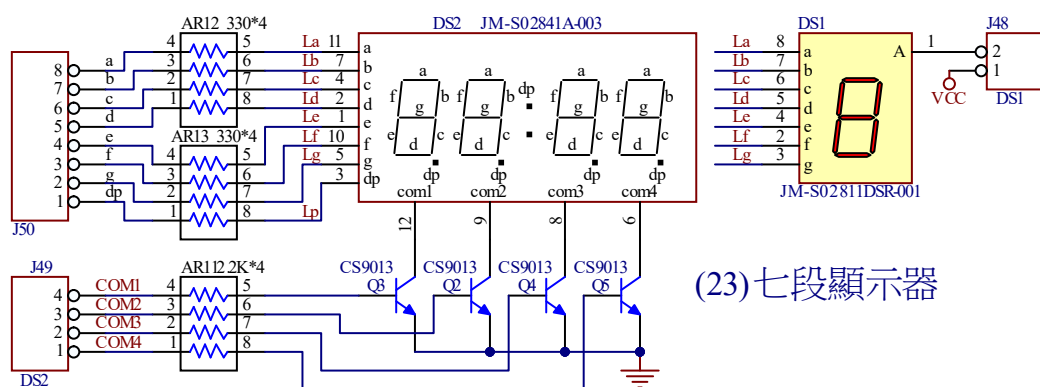


圖 4-9(b) 共陰極七段顯示器

本章使用四位數共陰極及一位數共陽極七段顯示器電路，如下圖(a)(b)所示：



(23) 七段顯示器

圖 4-10(a) 七段顯示器電路

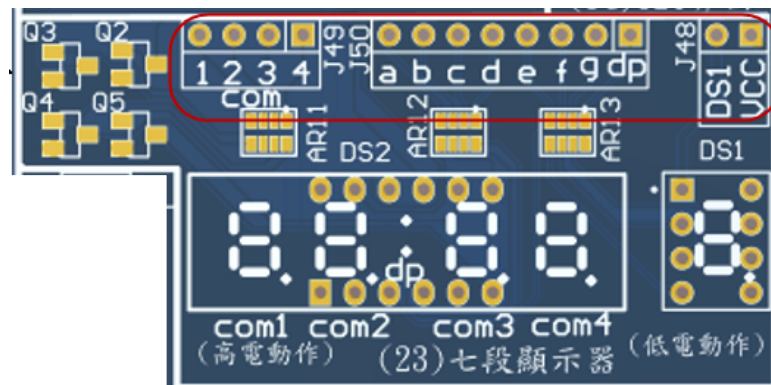


圖 4-10(b) 七段顯示器外型

若要令一位數共陽極七段顯示器(DS1)工作，先令 J48(DS1、VCC)短路提供 DS1 電源，同時在 J50(a~g、dp)輸入低電位數碼資料，即可令七段顯示器(DS1)亮。

若要令四位數共陰極七段顯示器(DS2)工作，由 J50(a~g、dp)輸入高電位數碼資料及 J49(com1~4)輸入高電位驅動 NPN 電晶體導通，分別選擇(com1~4)共陰七段顯示器(DS2)顯示數字。

七段顯示器數碼資料表以共陰極為例，如下表所示：

表 4-8 共陰極七段顯示器數碼表

	數字	pgfedcba	數碼	數字	pgfedcba	數碼
	0	00111111	0x3F	8	01111111	0x7F
	1	00000110	0x06	9	01101111	0x6F
	2	01011011	0x5B	A	01110111	0x77
	3	01001111	0x4F	B	01111100	0x7C
	4	01100110	0x66	C	00111001	0x39
	5	01101101	0x6D	D	01011110	0x5E
	6	01111101	0x7D	E	01111001	0x79
	7	00000111	0x07	F	01110001	0x71

七段顯示器實習範例及接線如下：

接線：D8~11-->J49(com1~4)，A0~A7-->J50(a~g、dp)

1. 範例 SEG1：送出數碼資料到一位數共陽極七段顯示器顯示 0~F。
2. 範例 SEG2：四位數共陰極掃描計數器，以十六進制顯示 000~FFFF。
3. 範例 SEG3：四位數共陰極掃描計數器，以十進制顯示 0000~9999。
4. 範例 SEG4：四位數共陰極掃描計數器，以十進制顯示-1999~1999。
5. 範例 SEG5：四位數共陰極七段顯示器顯示電子鐘的時、分及秒為閃爍，並具有調時及調分功能。

4-2.5 文字型 LCD 應用實習

液晶顯示器稱為 LCD 或 LCM，在市面上的消費性產品應用得非常多。且控制方式大致雷同分成控制器、驅動電路及顯示器三大部門。因受限於 MCU 接腳較少，本書使用以 4-bit 方式控制 LCM 模組。在文字型液晶顯示器插入 M18(LCM)，由 J54(LCM)連接主控板，同時在 VR3(VO)調整字幕顯示灰階，如下圖(a)(b)所示：

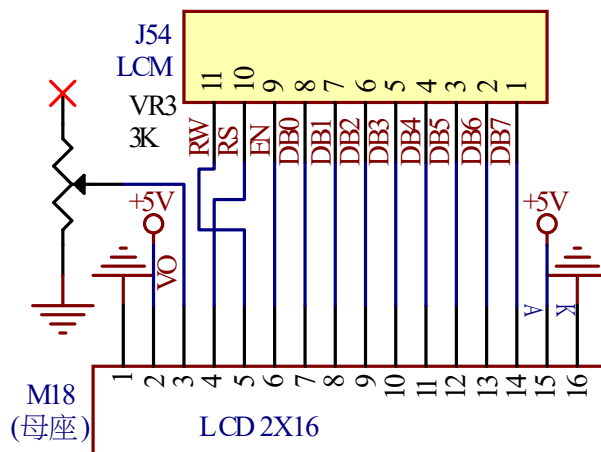


圖 4-11(a) 文字型 LCM 顯示器實習電路

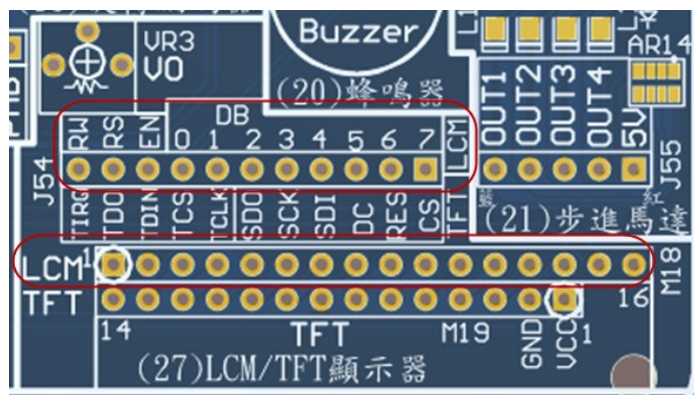


圖 4-11(b) 文字型 LCM 顯示器外型

圖中內含控制線(RS、EN、R/W)及資料線(DB0-7)。如下表(a)所示。

表 4-9(a) LCM 接腳定義

腳位	名稱	功 能 描 述	動 作
1	GND	電源接地	0V
2	VCC	電源正端	+5V 也可以接+3.3V
3	Vo	亮度調整	電壓愈低螢幕愈亮，使用可變電阻來調整
4	RS	暫存器 選擇信號	選擇 DB0~7 傳送為資料或命令 1=送到資料暫存器，0=送到命令暫存器
5	R/W	Read/Write 信號線	選擇 DB0~7 對 LCM 為讀取或寫入 1=Read 讀取，0=Write 寫入(固定接地)
6	E	致能信號	1=致能 LCM，0=禁能 LCM
7-14	DB0~7	匯流排	可用 8-bit 或 4-bit(DB4~7)輸入資料、命令及位址

LCM 是由 E、R/W 及 RS 三支腳共同配合來控制資料匯流排的流向，其時序是要先送暫存器選擇信號(RS)及讀寫信號(R/W)，再等 140ns 後，才能再送出致能信號(E)，如此才能完成資料或命令的寫入動作，如下表(b)所示。

表 4-9(b) LCM 的控制接腳

E	R/W	RS	DB0-7
0	X	X	無法存取 LCM
1	0	0	寫入命令到指令暫存器(IR)
1	1	0	讀取忙碌旗標(BF)到 DB7
1	0	1	寫入資料到資料暫存器(DR)
1	1	1	讀取資料暫存器(DR)的資料

LCM 指令碼如下表所示：

表 4-10 LCM 指令碼控制表

*：無作用

COMMAND	COMMAND CODE										COMMAND CODE	E-CYCLE f _{osc} =250KHz
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
SCREEN CLEAR	0	0	0	0	0	0	0	0	0	1	Screen Clear, Set AC to 0 Cursor Reposition	1.64ms
CURSOR RETURN	0	0	0	0	0	0	0	0	1	*	DDRAM AD=0, Return, Content Changeless	1.64ms
INPUT SET	0	0	0	0	0	0	0	1	I/D	S	Set moving direction of cursor, Appoint if move	40us
DISPLAY SWITCH	0	0	0	0	0	0	1	D	C	B	Set display on/off,cursor on/off, blink on/off	40us
SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	Remove cursor and whole display,DDRAM changeless	40us
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	Set DL,display line,font	40us
CGRAM AD SET	0	0	0	1	ACG						Set CGRAM AD, send receive data	40us
DDRAM AD SET	0	0	1	ADD						Set DDRAM AD, send receive data	40us	
BUSY/AD READ CT	0	1	BF	AC						Executing internal function, reading AD of CT	40us	
CGRAM/ DDRAM DATA WRITE	1	0	DATA WRITE						Write data from CGRAM or DDRAM	40us		
CGRAM/ DDRAM DATA READ	1	1	DATA READ						Read data from CGRAM or DDRAM	40us		
	I/D=1: Increment Mode; I/D=0: Decrement Mode S=1: Shift S/C=1: Display Shift; S/C=0: Cursor Shift R/L=1: Right Shift; R/L=0: Left Shift DL=1: 8D DL=0: 4D N=1: 2R N=0: 1R F=1: 5x10 Style; F=0: 5x7 Style BF=1: Execute Internal Function; BF=0: Command Received										DDRAM: Display data RAM CGRAM: Character Generator RAM ACG: CGRAM AD ADD: DDRAM AD & Cursor AD AC: Address counter for DDRAM & CGRAM	E-cycle changing with main frequency. Example: If fcp or f _{osc} =270KHz 40us x 250/270 =37us

LCM 內部字型(CG ROM)輸入 ASCII(b7~b0)碼即可顯示字型，如下表所示：

表 4-11 CG ROM 字型表

b7- b3 -b0		0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)		0	a	P	`	P		-	9	3		α	p
0001	(2)	!	1	A	Q	a	q	.	7	7	4		ä	q
0010	(3)	"	2	B	R	b	r	「	イ	ツ	×		ß	θ
0011	(4)	#	3	C	S	c	s	」	ウ	テ	エ		ε	ω
0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト		μ	Ω
0101	(6)	%	5	E	U	e	u	=	オ	ナ	ユ		σ	0
0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ		ρ	Σ
0111	CG RAM (8)	'	7	G	W	g	w	ア	キ	ヌ	ラ		g	π
1000	CG RAM (1)	(8	H	X	h	x	ィ	ク	ホ	リ		5	×
1001	(2))	9	I	Y	i	y	ッ	ケ	ル	ル		ˆ	y
1010	(3)	*	:	J	Z	j	z	エ	コ	ン	レ		j	7
1011	(4)	+	:	K	[k	[オ	サ	ヒ	ロ		*	π
1100	(5)	,	<	L	¥	l	l	ヤ	シ	フ	ワ		φ	π
1101	(6)	-	=	M]	m]	ユ	ズ	ハ	ン		ε	÷
1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	°		ñ	
1111	CG RAM (8)	/	?	O	_	o	+	ッ	リ	マ	°		ö	■

LCM 程式庫的定義檔(LiquidCrystal.h)常用函數式指令如下表所示：

表 4-12 LCM 函數式指令

函數	語法	常數
LiquidCrystal()接腳	lcd.LiquidCrystal(pin)	pin= RS,RW,EN,D0,D1,D2,D3,D4,D5,D6,D7
begin()開啓	lcd.begin(cols,rows)	設定 LCM 顯示格式 cols=16 , rows=2
print()畫面顯示	lcd.print(val) lcd.print(val, format)	val=數值、'字元'或"字串" format= BIN、OCT、DEC、HEX
write()寫入資料	lcd.write(val) lcd.write(str)	val =寫入字元資料或 ASCII 碼 str =寫入字串資料
createChar()造字	createChar(val, array[])	val =0~7 字 , array=字型資料陣列

Arduino 程式庫(LiquidCrystal)可設定 LCM 為 8-bit 或 4-bit 控制接腳，如下：

```
LiquidCrystal (RS,RW,EN,D0,D1,D2,D3,D4,D5,D6,D7); //設定 8-bit 接腳
LiquidCrystal (RS,EN,D0,D1,D2,D3,D4,D5,D6,D7);    //設定 8-bit 接腳
LiquidCrystal (RS,RW,EN,D4,D5,D6,D7); //設定 4-bit 控制接腳
LiquidCrystal (RS,EN,D4,D5,D6,D7); //設定 4-bit 控制接腳(不用的接腳須接地)
```

LCM(LCD)實習範例及接線以 4-bit 控制為例，如下：

接線：D2~D3-->J54(RS,EN),D4~7-->J54(DB4~7),GND-->J54(DB0~3,RW)

1. 範例 LCD1：以 4 bits 控制 LCD，顯示兩行文字，令其閃爍或移位。
2. 範例 LCD2：在 LCD 顯示"COUNT="，再重覆顯示-9999~+9999。
3. 範例 LCD3：在文字型 LCD 顯示上下午 12 小時電子鐘的變化。
4. 範例 LCD4：在文字型 LCD 顯示造字的變化。