

HC-05 藍芽模組

目錄

(按下 CTRL 鍵並按一下滑鼠以追蹤連結)

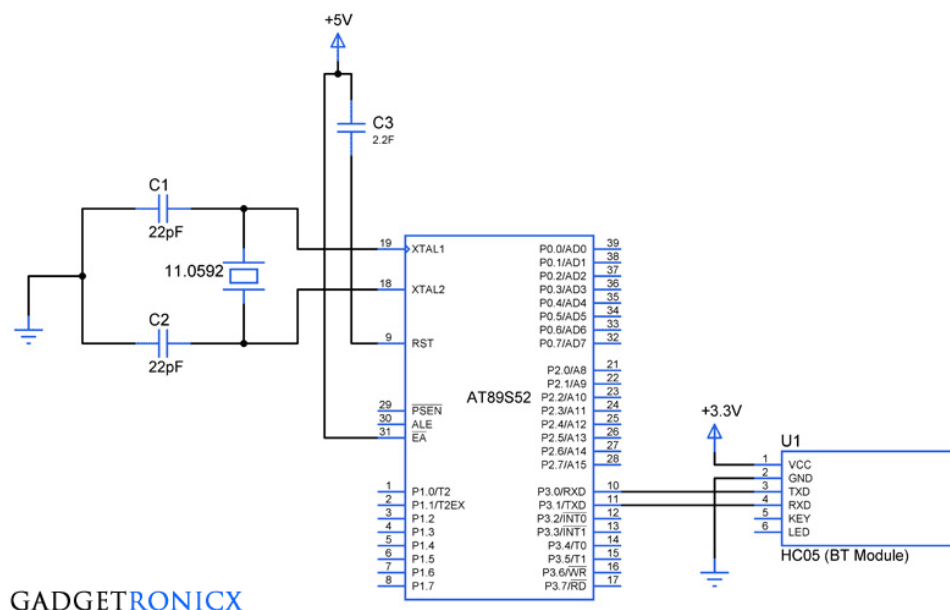
一、	模組資料	3
二、	8051 控制電路圖	3
三、	程式流程圖	4
四、	完整程式	5
五、	作品照片	14

一、模組資料

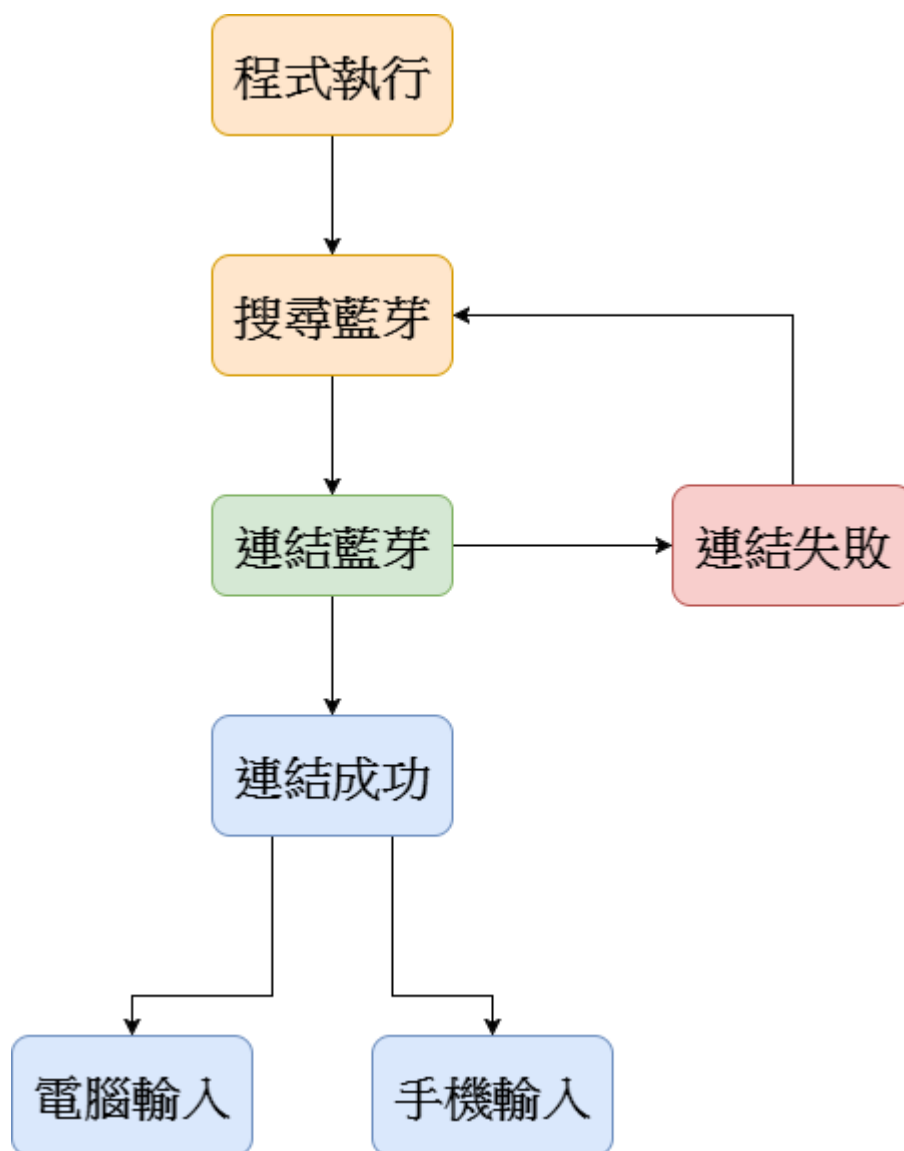
HC-05 採用英國劍橋的 CSR (Cambridge Silicon Radio) 公司的 BC417143 晶片，支援藍牙 2.1+EDR 規範。

主 / 從 (master/slave) 一體成型，出廠預設通常是「從端」模式，但是能自行透過 AT 命令修改。

二、8051 控制電路圖



三、程式流程圖



四、完整程式

```
#include <REG_MPC82G516.h> //引入 MPC82G516 SFR 定義檔
#include <intrins.h>

#define RL(x) _crol_(x,1); //重新定義 RL 左旋函式
void delay(int t);

sbit LCD_RS=P1^0; //LCM 接腳定義
sbit LCD_RW=P1^1;
sbit LCD_E=P1^2;
#define LCD_Data P2

void WriteDataLCD(char WDLCD);
void WriteCommandLCD(char WCLCD);
void LCDInit(void);

void delay_us(int t)
{
    while(t--);
}

unsigned char bbf ;
#define KB P0
#define RL(x) _crol_(x,1); //重新定義 RL 左旋函式
void key(char y);

void main(void)
{

    unsigned char    buffer, kb_buf, x, tmp;
    IE = 0x90;        //設定允許串列傳輸中斷 P6-4
```

```

SCON = 0x50;          //設定串列傳輸工作於模式 1, 可接收狀態
TMOD = 0x20;         //TIMER1 工作於 mode2(自動載入模式)
AUXR2 = 0x40;        //T1X12= 1 =>TIMER1 之計時時脈不除以 12
TH1 = 217;           //自動載入值為 217(鮑率為 9600bps)
TR1 = 1;              //啟動 TIMER1
buffer = 0;

delay_us(20000);     //若 LCM 反應較慢, 建議電源啟動後需先等一段時間再規劃 LCM, LCM
才可正常動作
LCDInit();
//==work=====
WriteCommandLCD(0x80);
while(1)
{

kb_buf = 0xfe;
for (x=0;x<4;x++)
{
KB = kb_buf;
tmp = KB;
while((tmp & 0xf0) != 0xf0)          //將 P0.3~P0.0 清為 0, 判斷 P0.7~P0.4 是
否皆為 1, 如果不是, 代表有按鍵被壓下
{
delay(3);          //延遲, 判斷是否為彈跳訊號
tmp = KB;
if ((tmp & 0xf0) != 0xf0)          //如果 P0.7~P0.4 仍然有 0 的情況, 則
表示非彈跳訊號
key(KB);          //呼叫按鍵副程式, 將按鍵值送到 LCD 顯示
do{
tmp = KB;          //讀回按鍵值
}while((tmp & 0xf0) != 0xf0);    //直到 P0.7~P0.4 皆為 1 為止(代表
按鍵已放開)
}

kb_buf = RL(kb_buf);    //掃描資料左旋, 準備掃描下一行
}
}

```

```
//=====
=====
```

```
switch(bbf) //依照 rs232 接收到的內容，執行相對應的敘述
{
```

```
    case '0': //a 字
    {
        LCDInit();
        delay(100); //延時 0.1 秒
        bbf=0; //清除接收暫存器
        break;
    }
```

```
    case '1': //a 字
    {
        WriteDataLCD(0x31);
        delay(100); //延時 0.1 秒
        bbf=0; //清除接收暫存器
        break;
    }
```

```
    case '2': //b 字
    {
        WriteDataLCD(0x32);
        delay(100); //延時 0.1 秒
        bbf=0; //清除接收暫存器
        break;
    }
```

```
    case '3': //c 字
    {
        WriteDataLCD(0x33);
        delay(100); //延時 0.1 秒
        bbf=0; //清除接收暫存器
        break;
    }
```

```
    case '4': //d 字
    {
        WriteDataLCD(0x34);
        delay(100); //延時 0.1 秒
        bbf=0; //清除接收暫存器
    }
```

```

        break;
    }
    case '5':    //d 字
    {
        WriteDataLCD(0x35);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;
    }
    case '6':    //d 字
    {
        WriteDataLCD(0x36);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;
    }
    case '7':    //d 字
    {
        WriteDataLCD(0x37);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;
    }
    case '8':    //d 字
    {
        WriteDataLCD(0x38);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;
    }
    case '9':    //d 字
    {
        WriteDataLCD(0x39);

```



```

        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;

    }

    case 'A':          //d 字
    {
        WriteDataLCD(0x41);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;

    }

    case 'B':          //d 字
    {
        WriteDataLCD(0x42);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;

    }

    case 'C':          //d 字
    {
        WriteDataLCD(0x43);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;

    }

    case 'D':          //d 字
    {
        WriteDataLCD(0x44);
        delay(100);    //延時 0.1 秒
        bbf=0;        //清除接收暫存器
        break;

    }

    case 'E':          //d 字

```

```

        {
            WriteDataLCD(0x45);
            delay(100);    //延時 0.1 秒
            bbf=0;        //清除接收暫存器
            break;
        }
        case 'F':        //d 字
        {
            WriteDataLCD(0x46);
            delay(100);    //延時 0.1 秒
            bbf=0;        //清除接收暫存器
            break;
        }
    }
}
}
}
//==副程式=====

void delay(int t)        //延遲函數開始
{
    int i, j;            //宣告整數變數 i, j
    for (i=0;i<t;i++)    //計數 t 次, 延遲 t x 1ms
        for (j=0;j<800;j++);    //計數 800 次, 延遲 1ms
}
//=====串列中斷副程式=====
void uart_isr(void) interrupt 4 //UART 的中斷函數, 使用中斷向量 4
{
    if (TI ==1)
        TI = 0;
    else
    {
        RI = 0;
        bbf = SBUF;    //將 buffer 內容送至 P2
    }
}
}

```

```

//寫資料到 LCD 副程式=====
void WriteDataLCD(char WDLCD)
{
    LCD_Data = WDLCD;
    LCD_RS = 1;
    LCD_RW = 0;
    LCD_E = 1; //產生負緣
    delay_us(100); //延遲等訊號穩定
    LCD_E = 0;
    delay_us(100); //延遲等訊號處理完畢
}

//寫指令到 LCD 副程式
void WriteCommandLCD(char WCLCD) //BuysC 為 0 時忽略忙檢測
{
    LCD_Data = WCLCD;
    LCD_RS = 0;
    LCD_RW = 0;
    LCD_E = 1;
    delay_us(100); //延遲等訊號穩定
    LCD_E = 0;
    delay_us(100); //延遲等訊號處理完畢
}

void LCDInit(void) //LCM 初始化
{
    WriteCommandLCD(0x38); //設定 LCD 介面 8 位元，2 行，5*7 字型
    WriteCommandLCD(0x06); //顯示游標移動設置
    WriteCommandLCD(0x0C); //顯示開及游標設置
    WriteCommandLCD(0x01); //顯示清除
    delay_us(5000);
}

//=====
void key(char y) //依照按鍵顯示對應鍵值副程式

```

```

{
switch(y)
{
case 0xee:
    SBUF = 0x30;
    break;
case 0xed:
    SBUF = 0x31;
    break;
case 0xeb:
    SBUF = 0x32;
    break;
case 0xe7:
    SBUF = 0x33;
    break;
case 0xde:
    SBUF = 0x34;
    break;
case 0xdd:
    SBUF = 0x35;
    break;
case 0xdb:
    SBUF = 0x36;
    break;
case 0xd7:
    WriteDataLCD(' 7 ');
    break;
case 0xbe:
    WriteDataLCD(' 8 ');
    break;
case 0xbd:
    WriteDataLCD(' 9 ');
    break;
case 0xbb:
    WriteDataLCD(' A ');
    break;
case 0xb7:
    WriteDataLCD(' B ');

```

```
        break;
    case 0x7e:
        WriteDataLCD(' C ');
        break;
    case 0x7d:
        WriteDataLCD(' D ');
        break;
    case 0x7b:
        WriteDataLCD(' E ');
        break;
    case 0x77:
        WriteDataLCD(' F ');
        break;
}

}
```

五、作品照片

