Forum: Microchip官方研討會，課程活動相關
Topic: 使用APP001開發版遇到的怪問題
Subject: 使用APP001開發版遇到的怪問題
作者: firststop0
2017年06月20日 16:52:08

device:PIC18F4520
IDE:X v3.60
compiler XC81.41

程式碼:

```
/*
 * File:      newmain.c
 * Author: user
 *
 * Created on 2017年5月17日, 下午 12:56
 */


#include <xc.h>
// PIC18F4520 Configuration Bit Settings
// 'C' source line config statements
// CONFIG1H
#pragma config OSC = INTIO67        // Oscillator Selection bits (Internal oscillator block,
port function on RA6 and RA7)
#pragma config FCMEN = OFF            // Fail-Safe Clock Monitor Enable bit (Fail-Safe Clock
Monitor disabled)
#pragma config IESO = OFF             // Internal/External Oscillator Switchover bit
(Oscillator Switchover mode disabled)

// CONFIG2L
#pragma config PWRT = ON               // Power-up Timer Enable bit (PWRT enabled)
#pragma config BOREN = ON             // Brown-out Reset Enable bits (Brown-out Reset enabled
and controlled by software (SBOREN is enabled))
#pragma config BORV = 3                 // Brown Out Reset Voltage bits (Minimum setting)

// CONFIG2H
#pragma config WDT = OFF               // Watchdog Timer Enable bit (WDT disabled (control is
placed on the SWDTEN bit))
#pragma config WDTPS = 32768        // Watchdog Timer Postscale Select bits (1:32768)

// CONFIG3H
#pragma config CCP2MX = PORTC      // CCP2 MUX bit (CCP2 input/output is multiplexed with
RC1)
#pragma config PBADEN = OFF          // PORTB A/D Enable bit (PORTB<4:0> pins are configured
as digital I/O on Reset)
#pragma config LPT1OSC = OFF        // Low-Power Timer1 Oscillator Enable bit (Timer1
```

```
configured for higher power operation)
#pragma config MCLRE = ON              // MCLR Pin Enable bit (MCLR pin enabled; RE3 input pin
disabled)

// CONFIG4L
#pragma config STVREN = ON             // Stack Full/Underflow Reset Enable bit (Stack
full/underflow will cause Reset)
#pragma config LVP = OFF               // Single-Supply ICSP Enable bit (Single-Supply ICSP
disabled)
#pragma config XINST = OFF             // Extended Instruction Set Enable bit (Instruction set
extension and Indexed Addressing mode disabled (Legacy mode))

// CONFIG5L
#pragma config CP0 = OFF               // Code Protection bit (Block 0 (000800-001FFFh) not
code-protected)
#pragma config CP1 = OFF               // Code Protection bit (Block 1 (002000-003FFFh) not
code-protected)
#pragma config CP2 = OFF               // Code Protection bit (Block 2 (004000-005FFFh) not
code-protected)
#pragma config CP3 = OFF               // Code Protection bit (Block 3 (006000-007FFFh) not
code-protected)

// CONFIG5H
#pragma config CPB = OFF               // Boot Block Code Protection bit (Boot block
(000000-0007FFh) not code-protected)
#pragma config CPD = OFF               // Data EEPROM Code Protection bit (Data EEPROM not
code-protected)

// CONFIG6L
#pragma config WRT0 = OFF              // Write Protection bit (Block 0 (000800-001FFFh) not
write-protected)
#pragma config WRT1 = OFF              // Write Protection bit (Block 1 (002000-003FFFh) not
write-protected)
#pragma config WRT2 = OFF              // Write Protection bit (Block 2 (004000-005FFFh) not
write-protected)
#pragma config WRT3 = OFF              // Write Protection bit (Block 3 (006000-007FFFh) not
write-protected)

// CONFIG6H
#pragma config WRTC = OFF              // Configuration Register Write Protection bit
(Configuration registers (300000-3000FFh) not write-protected)
#pragma config WRTB = OFF              // Boot Block Write Protection bit (Boot block
(000000-0007FFh) not write-protected)
#pragma config WRTD = OFF              // Data EEPROM Write Protection bit (Data EEPROM not
write-protected)

// CONFIG7L
#pragma config EBTR0 = OFF             // Table Read Protection bit (Block 0 (000800-001FFFh)
not protected from table reads executed in other blocks)
```

```
#pragma config EBTR1 = OFF             // Table Read Protection bit (Block 1 (002000-003FFFh)
not protected from table reads executed in other blocks)
#pragma config EBTR2 = OFF             // Table Read Protection bit (Block 2 (004000-005FFFh)
not protected from table reads executed in other blocks)
#pragma config EBTR3 = OFF             // Table Read Protection bit (Block 3 (006000-007FFFh)
not protected from table reads executed in other blocks)

// CONFIG7H
#pragma config EBTRB = OFF             // Boot Block Table Read Protection bit (Boot block
(000000-0007FFh) not protected from table reads executed in other blocks)

// #pragma config statements should precede project file includes.
// Use project enums instead of #define for ON and OFF.

#include <xc.h>


void interrupt HighISR(void);

void main(void){


    OSCCONbits.IRCF = 0b110;       // Internal Oscillator Frequency Select, Fosc = 4MHz

    TRISD = 0;
    TRISAbits.TRISA4 = 1;
    LATD = 0x00;

    T1CONbits.TMR1ON = 1;     // 1 = Enables Timer1
    T1CONbits.TMR1CS = 1;     // 1 = External clock from pin RC0/T1OSO/T13CKI (on the
rising edge)
    T1CONbits.T1SYNC = 0;     // 1 = Do not synchronize external clock input
    T1CONbits.T1OSCEN = 1;   // 1 = Timer1 oscillator is enabled
    T1CONbits.T1CKPS = 0;     // 00 = 1:1 Prescale value
    T1CONbits.T1RUN = 0;     // 0 = Device clock is derived from another source
    T1CONbits.RD16 = 1;       // 1 = Enables register read/write of TImer1 in one 16-bit
operation


    PIE1bits.TMR1IE = 1;     // 1 = Enables the TMR1 overflow interrupt
    RCONbits.IPEN = 1;       // 1 = Enable priority levels on interrupts
    INTCONbits.PEIE = 1;     // 1 = Enables all low-priority peripheral interrupts
    INTCONbits.GIE = 1;       // 1 = Enables all high-priority interrupts
    OSCCONbits.IDLEN = 1;     // 1 = Device enters an Idle mode on SLEEP instruction
    IPR1bits.TMR1IP = 1;     // 1 = High priority

    while(1);
```

```
}
void interrupt HighISR(void)
{
    PIR1bits.TMR1IF = 0;
    TMR1H = 0xF0;          // 65536-(32768/2) = 16384
    TMR1L = 0x00;
    LATD++;
}
```

我發現燒進去PIC後電路板的LED沒動作, 但我只要輕壓電路板中間的DIP SW 就會照著程式跑了 是電路板壞掉了嗎?