```
/********************************************************************
* FileName:        template_PIC18.c
* Dependencies:    See INCLUDES section below
* Processor:
* Compiler:        MPLAB C18 v.3.06
* Company:         Microchip Technology, Inc.
*
* This file is a basic template for creating C code for a PIC18F
* device.  Copy this file into your project directory and modify or
* add to it as needed.
*
* Add the suitable linker script to your project
*   - \mcc18\lkr\18f4520i.lkr if debugging with MPLAB ICD2
*   - \mcc18\lkr\18f4520.lkr if only programming the device
*   - \mcc18\lkr\18f4520_e.lkr if programming the device in extended mode
*   - \mcc18\lkr\18f4520i_e.lkr if debugging extended mode using ICD2
*
* If interrupts are not used, all code presented for that interrupt
* can be removed or commented out with C-style comment declarations.
*
* Software License Agreement
*
* The software supplied herewith by Microchip Technology Incorporated
* (the ìCompanyî) for its PICmicroÆ Microcontroller is intended and
* supplied to you, the Companyís customer, for use solely and
* exclusively on Microchip PICmicro Microcontroller products. The
* software is owned by the Company and/or its supplier, and is
* protected under applicable copyright laws. All rights are reserved.
* Any use in violation of the foregoing restrictions may subject the
* user to criminal sanctions under applicable laws, as well as to
* civil liability for the breach of the terms and conditions of this
* license.
*
* THIS SOFTWARE IS PROVIDED IN AN ìAS ISî CONDITION. NO WARRANTIES,
* WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT LIMITED
* TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL NOT,
* IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR
* CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.
*
*
* Author               Date        Comment
*~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~*/
// Dennis Cecic   11/05/2007    First release
// Stu Chandler   07/26/2007    Fine Tuned for clarity

/** Processor Header Files *****************************************
*
* Include the appropriate header (.h) file, depending on device used
* Example (for PIC18f4520): #include <p18F4520.h>
*
* Alternatively, the header file may be inserted from the Project
* window in the MPLAB IDE
*********************************************************************/

#include <p18f4520.h>
#include "LED7S.h"

/** Define Constants Here ******************************************/
```

```
#define CONSTANT1 10
#define CONSTANT2 20

#define TIEMPO 25000
#define TIEMPOMPX 250
#define S2 PORTAbits.RA4

/** Local Function Prototypes ***********************************/

void low_isr(void);
void high_isr(void);

void delay(void);
void delaympx (void);

void roll (void);

void Valida_Tiempo(void);
void Set_Time(char,char,char);
void Trata_Tiempo(void);

// ============================================================
// Configuration Bits
// For details on PIC18F configuration bit settings, see
// PIC18 Configuration Settings in MPLAB-IDE Help
// ============================================================

#pragma config OSC = EC
#pragma config WDT = OFF
#pragma config LVP = OFF
#pragma config BOREN = OFF

// ============================================================



/** Declare Interrupt Vector Sections ****************************/

#pragma code high_vector=0x08
void interrupt_at_high_vector(void)
{
   _asm goto high_isr _endasm
}


#pragma code low_vector=0x18
void interrupt_at_low_vector(void)
{
   _asm goto low_isr _endasm
}


/** Declarations *************************************************/
char segundos=0,minutos=0,horas=0,centesimas=0;
char Mode=0,MODEF=0;
int Contador = 4;
long tiempo = TIEMPO;
int i;
```

```
/*********************************************************************
* Function:        void main(void)
*
*********************************************************************/
#pragma code

void main (void)
{
    TRISA|=0x10;    //Activates RA4 as an input

    init_LED7S();
    Start_Display();
    tiempo /=4;
    Set_Time(20,00,0);
    Update2(minutos,segundos);

    while(1){
            //roll();
            //delay();
            if(MODEF){
                Trata_Tiempo();
                MODEF=0;
            }

            if(!S2){
                    while(!S2);
                    Mode++;
                    Mode%=3;
            }


    }



}


/*********************************************************************
* Function:        void high_isr(void)
* PreCondition:    None
* Input:
* Output:
* Side Effects:
* Overview:
*********************************************************************/
#pragma interrupt high_isr          // declare function as high priority
    isr
void high_isr(void)
{
}


/*********************************************************************
* Function:        void low_isr(void)
* PreCondition:    None
* Input:
* Output:
* Side Effects:
```

```
* Overview:
********************************************************************/
#pragma interruptlow low_isr        // declare function as low priority
    isr
void low_isr(void)
{
    if(_TMR0IF){
        Refresh();
        if(!Contador--){
            MODEF=1;
            Contador=4;
        }
    }
}




void delay (void){
    // Delay so human eye can see change
    int j;
    for (j = 0; j < tiempo; j++);
        Nop();
}

void delaympx (void){
    //Delay so human eye can see change
    int j;
    for (j = 0; j < TIEMPO; j++);
        Nop();
}


void roll (void){
    static int i=0,j=0;
    const char mensaje[10]={0x76,0x3F,0x38,0x77,0x00,0x1F,0x3F,0x6D,0xF9,
        0x00};

    for(j=3;j>=0;j--)
        Update_L(mensaje[(i++)%10],j);
    i-=3;
}

void Valida_Tiempo(void){
    if(centesimas>=100){
        centesimas=0;
        segundos++;
        if(segundos>=60){
            segundos=0;
            minutos++;
            if(minutos>=60){
                minutos=0;
                horas++;
                if(horas>=25){
                    horas=1;
                }
            }
        }
    }
```

```
}

void Set_Time (char h,char m,char s){
    if(h>=0&&h<=24)horas=h;
    if(m>=0&&m<=60)minutos=m;
    if(s>=0&&s<=60)segundos=s;
    //if(c>=0&&c<=99)centesimas=c;
}


void Trata_Tiempo(void){
    centesimas++;
    Valida_Tiempo();
    switch(Mode){
        case 0:
            roll();
            Erase_HourPoints();
            delay();
            break;
        case 1:
            Update2(horas,minutos);
            break;
        case 2:
            Update2(segundos,centesimas);
            break;
    }
    if((centesimas==0||centesimas==50)&&Mode!=0)Toggle_HourPoints();
}
```