

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "api.h"

//=====
// Notes on GDB Stub
//=====
// This program runs under Interrupt Control Mode = 2.
// During running the GDB STUB support routines,
// the interrupt level shown in EXR(CPU) is set to 6,
// and it uses USB/UART interrupt with level 7.
// To prevent from running user's interrupt handlers in STUB,
// USER MUST USE YOUR INTERRUPT WITH LEVEL FROM 1 TO 6, ONLY.
//=====

//=====
// Globals
//=====
extern sDateTime gDateTime; // Current Clock Database

//=====
// Main Routine
//=====
int main(void)
{
    //-----
    // Initialize System Hardware
    //-----
    Init_USB();
    Init_CPG();
    Init_GPIO();
    Init_INTC();
    Init_BSC();
    Init_TMR67();
    //
    Init_SPI();
    Init_RTC();
    Init_UART();
    Init_LCD1602();

    //-----
    // Set RTC
    //-----
    {
        gDateTime.bcd_years = 0x10;
        gDateTime.bcd_months = 0x04;
        gDateTime.bcd_days = 0x02;
        gDateTime.weekdays = FRI;
    }
}

```

```

    gDateTime.bcd_hours   = 0x20;
    gDateTime.bcd_minutes = 0x04;
    gDateTime.bcd_seconds = 0x00;
    //
//   RTC_Set (&gDateTime);
}
//-----
// Clock Function
//-----
{
    UI08 *WEEK[] = {(UI08*)"SUN", (UI08*)"MON", (UI08*)"TUE", (UI08
*)"WED", (UI08*)"THU", (UI08*)"FRI", (UI08*)"SAT"};
    UI08 str0[64];
    UI08 str1[64];
    UI08 sec_prev = 255;
    UI08 key;
    UI08 str2[32];
    UI08 str3[32];
    UI08 str4[3];
    UI08 str5[3];
    UI08 str6[3];
//   int i=0;
    while(1)
    {
        if (RTC_Check_Power ())
        {
            RTC_Get (&gDateTime);
            key = KEY_Get_Static_State ();
            if (key== 0x80)
            {
                sprintf((char*) str0, "SETTING %02x:%02x:%02x",
gDateTime.bcd_hours, gDateTime.bcd_minutes, gDateTime.bcd_seconds);
//               sprintf((char*) str0, "%02x", gDateTime.bcd_seconds);
                str4[0]=str0[8];
                str4[1]=str0[9];
                str5[0]=str0[11];
                str5[1]=str0[12];
                str6[0]=str0[14];
                str6[1]=str0[15];
                LCD1602_Print_String(0, str0);
                LCD1602_Print_String(1, " ");
                int SetPos[3]={8, 11, 14};
                int pos_i=2;
                LCD1602_Set_Pos(SetPos[pos_i], 1);
                LCD1602_Set_Char(0x5e);
                key = KEY_Get_Static_State ();
                while(0x00!=KEY_Get_Static_State());
                while(1) {
                    //-----

```

```

// Static Key State
//-----
key = KEY_Get_Static_State();
while(0x00!=KEY_Get_Static_State());

if(key==0x02) {
    LCD1602_Set_Pos(SetPos[pos_i], 1);
    LCD1602_Set_Char(' ');
    if(pos_i>0) pos_i--;
    else pos_i=2;
    LCD1602_Set_Pos(SetPos[pos_i], 1);
    LCD1602_Set_Char(0x5e);
}
if(key==0x01 && pos_i==2) {
    if(str0[15]<'9') {
        str0[15]=str0[15]+1;
    }
    else{
        str0[15]='0';
        str0[14]=str0[14]+1;
        if(str0[14]>'5') {
            str0[14]='0';
        }
    }
}
if(key==0x01 && pos_i==1) {
    if(str0[12]<'9') {
        str0[12]=str0[12]+1;
    }
    else{
        str0[12]='0';
        str0[11]=str0[11]+1;
        if(str0[11]>'5') {
            str0[11]='0';
        }
    }
}
if(key==0x01 && pos_i==0) {
    if(str0[8]<'2')
    {
        if( str0[9]<'9') {
            str0[9]=str0[9]+1;
        }
        else{
            str0[9]='0';
            str0[8]=str0[8]+1;
        }
    }
}
else

```

```

        {
            if(str0[9]<'3')
            {
                str0[9]=str0[9]+1;
            }
            else
            {
                str0[8]='0';
                str0[9]='0';
            }
        }
    }
    LCD1602_Print_String(0, str0);
    //-----
    // Wait for Tick Change
    //-----
    if(key==0x80)
    {

        UI08 i_sec, i_min, i_hour;
        i_sec=(str0[14]-0x30)*0x10+(str0[15]-0x30);
        i_min=(str0[11]-0x30)*0x10+(str0[12]-0x30);
        i_hour=(str0[8]-0x30)*0x10+(str0[9]-0x30);

        //                                     gDateTime.bcd_years   =
0x10;
        //                                     gDateTime.bcd_months  = 0x04;
        //                                     gDateTime.bcd_days    = 0x02;
        //                                     gDateTime.weekdays   = FRI;
        gDateTime.bcd_hours   = i_hour;
        gDateTime.bcd_minutes = i_min;
        gDateTime.bcd_seconds = i_sec;
        //
        RTC_Set(&gDateTime);

        break;
    }
    TMR_Wait_One_Tick();
}
if (gDateTime.bcd_seconds != sec_prev)
{
    sprintf((char*) str0, "SETTUS %02x:%02x:%02x",
gDateTime.bcd_hours, gDateTime.bcd_minutes, gDateTime.bcd_seconds);
    sprintf((char*) str1, "20%02x/%02x/%02x (%s)",
gDateTime.bcd_years, gDateTime.bcd_months, gDateTime.bcd_days, WEEK
[gDateTime.weekdays - 1]);
}

```

```
        LCD1602_Print_String(0, str0);
        LCD1602_Print_String(1, str1);
        sec_prev = gDateTime.bcd_seconds;
    }
}
else
{
    LCD1602_Print_String(0, (UI08*) "[H8SX] ERROR !");
    LCD1602_Print_String(1, (UI08*) "RTC Power Fail.");
    while(1);
}
}
}

//-----
// End of main()
//-----
return 0;
}
```