Real-Time Clock (RTC) example project

1.0

Features

- Alarm configuration and usage example
- Multiple overflow cases
- Daylight Saving Time (DST) configuration and usage example

General Description

This example project demonstrates Real Time Clock component operation with the configured fixed Daylight Saving Time (DST) mode, active alarm issuing. The PM/AM and leap year status, and overflow between years are demonstrated.

Development kit configuration

1. This project is written for 2X16 display as the one available on CY8CKIT-001.
2. The PSoC3/5 DVK CY8CKIT-001 board besides default configuration should have LCD power jumper (J12).
3. Build the project and program the hex file on to the target device.
4. Power cycle the device and observe the results on the LCD.

Project configuration

The example project consists of the Character LCD and RTC components. The top design schematic is shown on the Figure 1. The Character LCD component is used for date and time displaying.

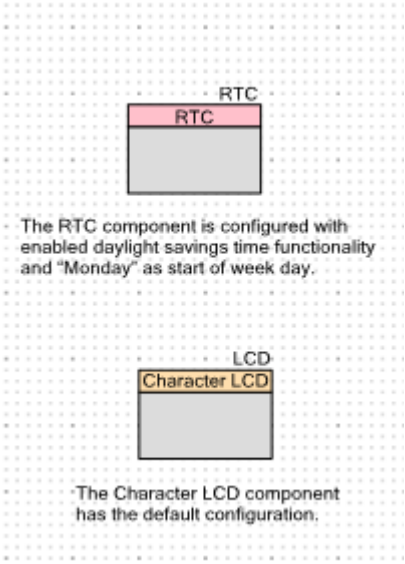

Figure 1. Top design schematic.

The Character LCD component has the default configuration. The Keil Reentrancy File is added to the project to make LCD_Position, LCD_WriteControl, LCD_IsReady, LCD_WriteData, LCD_WrDatNib, LCD_PrintString functions reentrant as they are called from the both: main.c and RTC ISR.

The RTC component is configured with enabled daylight savings time functionality and “Monday” as start of week day.

The clock system configuration is shown on the Figure 2. The XTAL 32 kHz should be enabled for proper RTC component operation.

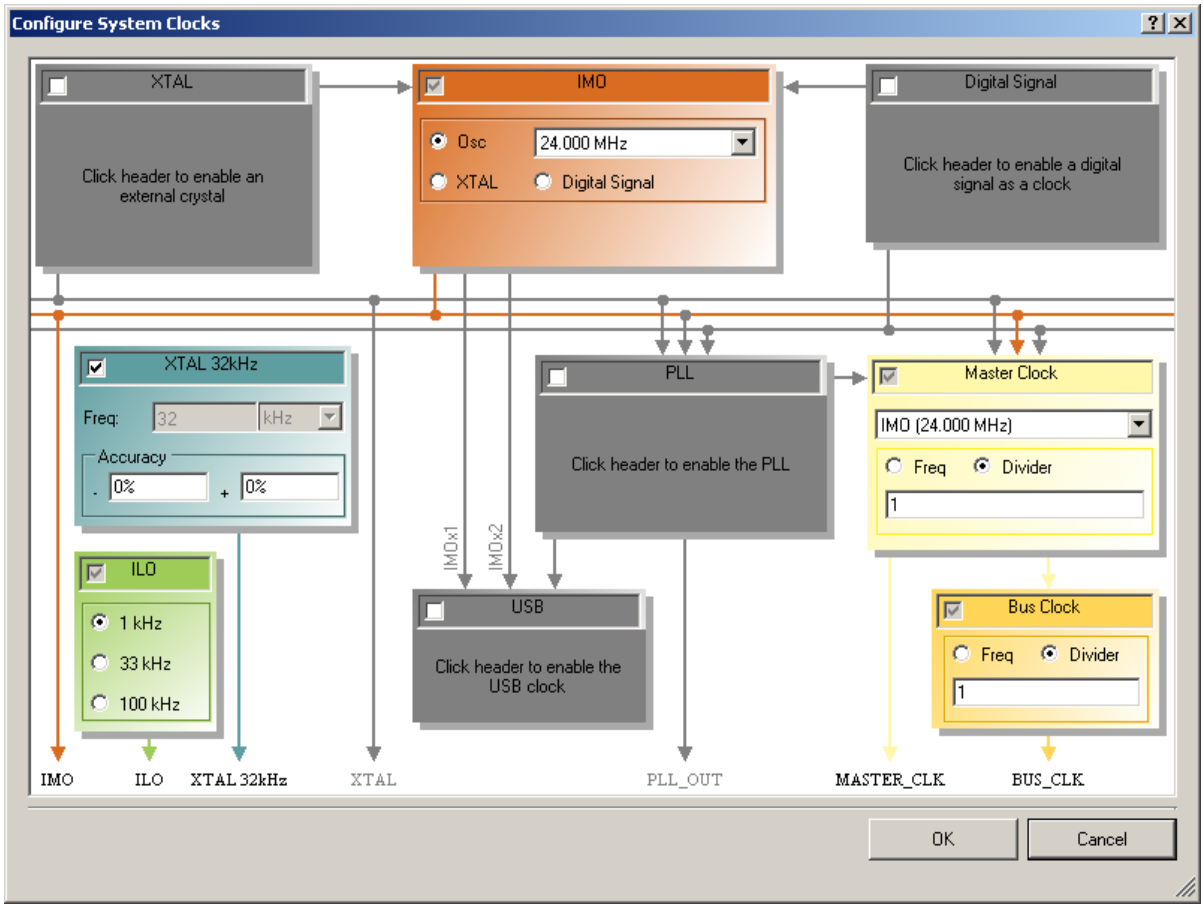

Figure 2. The “Configure System Clocks” window.

Project description

The initial time in the project set to 22:59:55 and date set to 2007-12-31, by filling up the declared variable of RTC_TIME_DATE type and passing it as a parameter to WriteTime(RTC_TIME_DATE *timeDate):

```
RTC_TIME_DATE Start;

/* Fill struct with date and time */
Start.Sec = 55u;
Start.Min = 59u;
Start.Hour = 22u;
Start.DayOfMonth = 31u;
Start.Month = 12u;
Start.Year = 2007u;

/* Set date and time */
RTC_WriteTime(&Start);
```

Active alarm configuration: set alarm time to 01:03:05 and date to 2008-01-01, set alarm mask

```
/* Set alarm date and time */
RTC_WriteAlarmSecond(5u);
RTC_WriteAlarmMinute(3u);
RTC_WriteAlarmHour(1u);
RTC_WriteAlarmDayOfMonth(1u);
RTC_WriteAlarmMonth(1u);
RTC_WriteAlarmYear(2008u);

/* Set alarm mask */
RTC_WriteAlarmMask(RTC_ALARM_SEC_MASK | RTC_ALARM_MIN_MASK |
                   RTC_ALARM_HOUR_MASK | RTC_ALARM_DAYOFMONTH_MASK |
                   RTC_ALARM_MONTH_MASK | RTC_ALARM_YEAR_MASK);
```

Configuring interval mask – add handling of the corresponding interrupt stubs of the RTC component:

```
/* Set interval mask - handling of interrupt stubs of the RTC component*/
RTC_WriteIntervalMask(RTC_INTERVAL_SEC_MASK | RTC_INTERVAL_MIN_MASK |
                      RTC_INTERVAL_HOUR_MASK | RTC_INTERVAL_DAY_MASK |
                      RTC_INTERVAL_WEEK_MASK | RTC_INTERVAL_MONTH_MASK |
                      RTC_INTERVAL_YEAR_MASK);
```

Configuring fixed DST mode with start and stop time. Set DST start hour to 23, day of month to 31 and month to 12 and DST stop hour to 23, day of month to 31 and month to 12. The DST offset is set to 123 minutes.

```
/* DST start configuration */
RTC_WriteDSTMode(RTC_DST_ENABLE | RTC_DST_FIXDATE);
RTC_WriteDSTStartHour(23u);
RTC_WriteDSTStartDayOfMonth(31u);
RTC_WriteDSTStartMonth(12u);

/* DST stop configuration */
RTC_WriteDSTStopHour(2u);
RTC_WriteDSTStopDayOfMonth(1u);
RTC_WriteDSTStopMonth(1u);
RTC_WriteDSTOffset(123u);
```

The code below saves current RTC status to the tmpVar variable:

```
/* Get status */
tmpVar = RTC_ReadStatus();
```

The code below reads current daytime (AM/PM) and prints it on the display at a specific position:

```
/* Get and print daytime AM/PM */
if(RTC_STATUS_AM_PM & tmpVar)
{
    myLCD_Position(0u, 14u);
    myLCD_PrintString("PM");
}
else
```

```
{
    myLCD_Position(0u, 14u);
    myLCD_PrintString("AM");
}
```

The code below reads current DST status and prints “D” if DST is active and “ ” if current time and date is out of DST period:

```
/* Get and print DST status */
if (RTC_STATUS_DST & tmpVar)
{
    myLCD_Position(1u, 13u);
    myLCD_PutChar('D');
}
else
{
    myLCD_Position(1u, 13u);
    myLCD_PutChar(' ');
}
```

The code below reads current DST status and prints “D” if DST is active and nothing if current time and date is out of DST period:

```
  /* Get and print alarm status */
  if (RTC_STATUS_AA & tmpVar)
  {
        myLCD_Position(1u, 15u);
        myLCD_PutChar('A');
  }
  else
  {
        myLCD_Position(1u, 15u);
        myLCD_PutChar(' ');
  }
```

The PrintDecNumber() function is implemented in utils.c file. It displays decimal numbers (first parameter) on a specific row (second parameter) and column (the third parameter) on the Character LCD component, named LCD. If the Character LCD component's name will be changed it also should be updated in this function.

There are interrupt stubs in the RTC ISR. The every second interrupt for this example project is listed below. The active alarm, DST, AM/PM and leap year statuses renew code is removed from the listing below, as it is the same as in the main() function and are described above. The only second value update code a shown below. Refer to the RTC_INT.c file for the full code listing.

```
void RTC_EverySecondHandler(void)
{
    /*  Place your every second handler code here. */
    /* `#START EVERY_SECOND_HANDLER_CODE` */

    /* Get and print current seconds */
    temp = RTC_ReadSecond();
    PrintDecNumber(temp, 0, 11);

    /* For the rest of the code, refer to the RTC_INT.c file */

    /* `#END` */
}
```

The date, time and related values are updated in related interrupt stubs. For example, the minutes value are updated in a RTC_EveryHourHandler() as shown below:

```
void RTC_EveryHourHandler(void)
{
    /*  Place your every hour handler code here. */
    /* `#START EVERY_HOUR_HANDLER_CODE` */

    /* Get and print current hours */
    temp = RTC_ReadHour();
    PrintDecNumber(temp, 0, 5);

    /* `#END` */

}
```

Expected results

Display legend is described below:

First string: “RTC HH:MM:SS AP”
RTC – design name, HH – hours, MM – minutes, SS – seconds, AP - AM/PM.

Second string: “YY:MM:DD DW L D”
YY - two last digits of the year, MM – month, DD - day of month, DW - day of week, L - present if year is leap year, D - DST is active.

There are three state of the RTC described below:

The design name is RTC. Current time is 22:59:58 PM. The date is 31-12-07. There is first day of week. As per component configuration the first day of week is Monday.

```
RTC 22:59:58 PM
31-12-07 1
```

When time reaches 23:00:00 on December 31 of any year (31-12-xx) as per DST start time configured above, the time will be shifted for 123 minutes forward. The year value changes to 08, a leap year (“L”). The DST active status is displayed by “D”:

```
RTC 01:03:02 AM
01-01-08 2 L D
```

When time reaches 01:03:05 the “A” will be displayed to show active alarm status:

```
RTC 01:03:05 AM
01-01-08 2 L D A
```