

RTC general-purpose library module

1. Introduction.....	2
2. Module Features	2
3. List of Component Modules	3
4. Using the Library Module in a Project.....	3
1. Use the MpAM to configure the code as required.	3
5. List of Shared Parameters.....	4
<i>Shared Data Bytes</i>	<i>4</i>
<i>Shared Functions</i>	<i>4</i>
<i>Shared Macros</i>	<i>4</i>
6. Functions	5
7. Macros	6
8. Error and Status Flags.....	9

1. Introduction

The primary objective of the RTC (Real Time Clock) general-purpose library module is to speed up programmer's job. With this, programmer's user interface is shifted from one of implementation specifics (such as setting control bits and testing status bits) to one of logical commands (such as read, write and execute). This is very much like using a high level language. This allows the user to focus on the requirements of their applications, instead of small details. The RTC module provides interrupt-based operation which happens every second to update the time. Module code is linkable and relocatable, which provides user facility to use it without modifications.

2. Module Features

It has the following features.

- A regular 24-hour clock (hours, minutes, and seconds).
- Selectable calendar (months, days of the month, years-2digit).
- Selectable day of the week feature for calculating the day of the week for 100 years (January 1st, 1995 to December 31st, 2094).
- Selectable century.
- Selectable day of the week feature for calculating the day of the week for 12700 years (March 1st, 0000 to December 31st 12700). This applies to the Gregorian Calendar. From Webster's New Universal Unabridged Dictionary, 1983 edition:

Gregorian Calendar: A corrected form of the Julian calendar. It was introduced by Pope Gregory XIII in 1582 and now used in most countries of the world. It provides for an ordinary year of 365 days and a leap year of 366 days every fourth even year, exclusive of century years, which are leap years only if exactly divisible by 400.

3. List of Component Modules

Library module zip file contains the following files:

16RTCInttest.ex.txt	This is the main test file developed to demonstrate use of the library functions.
16rtcint.asm	This is the RTC code implementation file. <u>One needs to include this file in their project.</u>
16rtcint.inc	This file contains definition of shared parameters for use in Assembly language. One needs to include this file in the Assembly file where they are calling library routines. This file is taking care of definition of all Extern Global parameter so one can directly call library routines in their program.
P16xxx.inc	General purpose processor definition file for PIC16xxx family

4. Using the Library Module in a Project

Please follow below steps to use this library module in your project.

1. Use the Application Maestro to configure the code as required.
2. At the Generate Files step, save the output to the directory where your projects code resides.
3. In MPLAB open your project workspace.
4. Verify that the Microchip language tool suite is selected (*Project>Select Language Toolsuite*).
5. In the Workspace view, right-click on the "Source Files" node. Select the "Add Files" option. Select 16rtcint.asm and click **OK**.
6. Now right-click on the "Linker Scripts" node and select "Add Files". Add the appropriate linker file (.lkr) for the project's target microcontroller. Click on OK to close Edit Project window. Now save your project.
7. Add any other files that the project may require. Save and close the project.
8. In your main source (assembler) file, include 16rtcint.int. By doing so, all files required to make the generated code work in your project will be included by reference when you build the project.
9. To use the module in your application, invoke the functions or macros as needed.

Call the required function to achieve application specific task. For details of all functions, please refer to section 6.

5. List of Shared Parameters

Shared Data Bytes

VRTCStatus	It contains the flag bits which indicate a change in RTC parameters (refer to Section 8 for more details).
VSec	This is the 8-bit register that holds the value for the seconds (0-59).
VMin	This is the 8-bit register that holds the value for the minutes (0-59).
VHour	This is the 8-bit register that holds the value for the hours (0-23).
VDay	This is the 8-bit register that holds the value for the days (1-31).
VMonth	This is the 8-bit register that holds the value for the months (1-12).
VYear	This is the 8-bit register that holds the value for the years (0-99).
VCentury	This is the 8-bit register that holds the value for the centuries (0-127).
VDOW	This is the 8-bit register that holds the value of the days of the week (0-6). Sunday=0, Monday=1, ..., Saturday=6.

Shared Functions

RTCIntInit	This function is used for RTC Initialization. It initializes TIMER1 and the time and date according to compile time selections.
RTCIntISR	This is an Interrupt service routine for the RTC, which happens every time TIMER1 overflows (i.e. every second). It handles incrementing the time and date on interrupt. Call it from Interrupt service routine after context savings.
RTCIntCheckLeapYear	This routine is used after setting the date to check if current year is a leap year or not. If it is, then it sets the lyf flag, otherwise it clears it.

Shared Macros

mClearTIMER1	Clear TIMER1 (clears TMR1H and TMR1L).
mLoadTIMER1	Load TIMER1 with 0x8000 so it overflows every second.
mReLoadTIMER1	Re-loads TIMER1 with 0x8000. This macro adds 0x8000 to TIMER1.
mRTCIntClearSeconds	This is used to clear the seconds and reset them to zero.
mRTCIntSetSeconds	This is used to set the seconds.
mRTCIntSetMinutes	This is to set the minutes.
mRTCIntSetHours	This is to set the hours.
mRTCIntSetTime	Sets the time Hours, Minutes, and Seconds.
mRTCIntSetDay	This is to set the day of the month.
mRTCIntSetDOW	This is to set the day of the week manually.
mRTCIntSetMonth	This is to set the month.
mRTCIntSetYear	This is to set the year.
mRTCIntSetCentury	This is to set the century.
mRTCIntSetDate	Sets the date Month, Day, Century, and Year.

6. Functions

Function	RTCIntInit
Preconditions	None
Overview	This routine is used to initialize the RTC. It initializes the TIMER1 module for external watch-crystal oscillator. It also clears the time to 00:00:00 and sets the date to "Saturday, January 1 st , 2000" depending on runtime options.
Input	None
Output	None
Side Effects	Databank, W and STATUS changed
Stack Requirement	1 level deep
Function	RTCIntISR
Preconditions	TIMER1 overflow which causes an interrupt (every second)
Overview	This is an Interrupt service routine for TIMER1 interrupt. It handles updating the clock and calendar on interrupt. Call it from Interrupt service routine.
Input	None
Output	Updates the Real time clock (sec, min, hour , day, dow, month, year, century, and vRTCStatus which contains the flags that are set when one of the variables in RTC changes)
Side Effects	Databank, PCLATH, Wreg, STATUS reg changed
Stack Requirement	2 level deep
Function	RTCIntCheckLeapYear
Preconditions	The date has to be set
Overview	This routine is used after setting the time to check if current year is a leap year or not. If it is, the routine sets the lyf flag, otherwise it clears it.
Input	None
Output	Sets or clears vRTCStatus, lyf (leap year flag) bit.
Side Effects	Databank, Wreg, STATUS reg changed
Stack Requirement	1 level deep

7. Macros

Macro	mClearTIMER1
Overview	Clears Timer1 (clears TMR1H and TMR1L).
Input	None
Output	None
Side Effects	Data bank changed
Stack Requirement	None
Macro	mLoadTIMER1
Overview	Loads TIMER1 with 0x8000 so that it overflows every second.
Input	None
Output	None
Side Effects	Databank, W changed
Stack Requirement	None
Macro	mReloadTIMER1
Overview	Adds 0x80 to TMR1H.
Input	None
Output	None
Side Effects	Databank, W changed
Stack Requirement	None
Macro	mRTCIntClearSeconds
Overview	This macro is used to reset the seconds in the RTC.
Input	None
Output	None
Side Effects	Databank changed
Stack Requirement	None
Macro	mRTCIntSetSeconds
Overview	This macro is used to set the seconds in the RTC.
Input	Seconds
Output	None
Side Effects	Databank, W changed
Stack Requirement	None
Macro	MRTCIntSetMinutes
Overview	This macro is used to set the minutes in the RTC.
Input	Minutes
Output	None
Side Effects	Databank, W changed
Stack Requirement	None

Macro	mRTCIntSetHours
Overview	This macro is used to set the hours in the RTC.
Input	Hours
Output	None
Side Effects	Databank, W changed
Stack Requirement	None
Macro	mRTCIntSetTime
Overview	This macro Sets the time.
Input	Hours, Minutes, and Seconds
Output	None
Side Effects	Databank, W changed
Stack Requirement	None
Macro	mRTCIntSetDay
Overview	This macro is used to set the day of the month in the RTC.
Input	Day of the Month (Sunday=0...Saturday=6)
Output	None
Side Effects	Databank, W, STATUS and PCLATH changed
Stack Requirement	up to 2 levels deep
Macro	mRTCIntSetDOW
Overview	This macro is used to set the day of the week in the RTC.
Input	Day of the Week
Output	None
Side Effects	Databank, W changed
Stack Requirement	None
Macro	mRTCIntSetMonth
Overview	This macro is used to set the months in the RTC.
Input	None
Output	None
Side Effects	Databank, W, STATUS and PCLATH changed
Stack Requirement	up to 2 levels deep
Macro	mRTCIntSetYear
Overview	This macro is used to set the years in the RTC.
Input	Year.
Output	None
Side Effects	Databank, W, STATUS and PCLATH changed
Stack Requirement	up to 2 levels deep
Macro	mRTCIntSetCentury
Overview	This macro is used to set the centuries in the RTC.
Input	Century.
Output	None
Side Effects	Databank, W, STATUS and PCLATH changed
Stack Requirement	up to 2 levels deep

Macro	mRTCintSetDate
Overview	Sets the Date in the RTC
Input	Month, Day, Century, Year
Output	None
Side Effects	Databank, W, STATUS and PCLATH changed
Stack Requirement	up to 2 levels deep

8. Error and Status Flags

All status bits are set as a bit flag in memory location named `vRTCStatus`. Individual bit flag indicates different changes in the RTC. Please refer below list for the information.

Secf	Indicates a change in seconds (this bit is set every second).
Minf	Indicates a change in minutes (this is set every minute).
Hourf	Indicates a change in hours (this is set every hour).
Dayf	Indicates a change in days (this is set once a day).
DOWf	Indicates a change in Day_of_Week (this is set once a day).
Monthf	Indicates a change in months (this is set once a month).
Yearf	Indicates a change in years (this is set once a year).
lyf	Indicates a leap year.

All these flags need to be reset in software by the user except for “lyf “ which is set and reset automatically.