



The following document contains information on Cypress products. Although the document is marked with the name "Spansion" and "Fujitsu", the company that originally developed the specification, Cypress will continue to offer these products to new and existing customers.

Continuity of Specifications

There is no change to this document as a result of offering the device as a Cypress product. Any changes that have been made are the result of normal document improvements and are noted in the document history page, where supported. Future revisions will occur when appropriate, and changes will be noted in a document history page.

Continuity of Ordering Part Numbers

Cypress continues to support existing part numbers. To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information

Please contact your local sales office for additional information about Cypress products and solutions.

About Cypress

Cypress (NASDAQ: CY) delivers high-performance, high-quality solutions at the heart of today's most advanced embedded systems, from automotive, industrial and networking platforms to highly interactive consumer and mobile devices. With a broad, differentiated product portfolio that includes NOR flash memories, F-RAM™ and SRAM, Traveo™ microcontrollers, the industry's only PSoC® programmable system-on-chip solutions, analog and PMIC Power Management ICs, CapSense® capacitive touch-sensing controllers, and Wireless BLE Bluetooth® Low-Energy and USB connectivity solutions, Cypress is committed to providing its customers worldwide with consistent innovation, best-in-class support and exceptional system value.

Colophon

The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for any use that includes fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for any use where chance of failure is intolerable (i.e., submersible repeater and artificial satellite). Please note that Spancion will not be liable to you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the US Export Administration Regulations or the applicable laws of any other country, the prior authorization by the respective government entity will be required for export of those products.

Trademarks and Notice

The contents of this document are subject to change without notice. This document may contain information on a Spancion product under development by Spancion. Spancion reserves the right to change or discontinue work on any product without notice. The information in this document is provided as is without warranty or guarantee of any kind as to its accuracy, completeness, operability, fitness for particular purpose, merchantability, non-infringement of third-party rights, or any other warranty, express, implied, or statutory. Spancion assumes no liability for any damages of any kind arising out of the use of the information in this document.

Copyright © 2013 Spancion Inc. All rights reserved. Spancion®, the Spancion logo, MirrorBit®, MirrorBit® Eclipse™, ORNAND™ and combinations thereof, are trademarks and registered trademarks of Spancion LLC in the United States and other countries. Other names used are for informational purposes only and may be trademarks of their respective owners.

Application Note

Software Real Time Clock

for MB89630R and similar Microcontrollers

© Fujitsu Microelectronics Europe GmbH

Vers. 5.1 by F. T.

1) Overview

This Application Note features a Real Time Clock from a Watch Timer Interrupt (thus usable with a 32kHz Sub-Clock). It has been developed for the MB89630R series (and tested on the corresponding Starter-kit) but can be easily adapted for the other 8-bit series (MB89120/A, MB89130, ..., MB89990).

It also shows how it is possible to build a project composed of different modules, some written in Assembler and others in C, and to handle variables defined in independent sections.

2) Components of the project

The project is named “Calendar.Prj” and is composed of the following modules:

- **Init.Asm** processes global initialization (Stack definition, RAZ memory) and a jump to the main module (Calendar.C).
- **Mode_Ext.Asm** defines the Reset Vector.
- **Calendar.C** contains the Clock speed definition, and the initialization of the RTC variables (CALLing RazClock.Asm). Manages the Watch Timer interrupt (CALLing WTIT.Asm), toggling LED0 each second occurs, and can check the date calculated (CALLing CHECK_DATE within WTIT.Asm).
- **RazClock.Asm** initializes the Date parameters.
- **WTIT.Asm** manages the Watch Timer interrupt and updates the RTC variables.
- **Int89630.Asm** defines the interrupt used by the Processor (!! to change for the appropriate Fujitsu’s module if another Microcontroller is used !!).
- **MB89630.Asm** defines the I/O registers of the Processor (!! to change for the appropriate Fujitsu’s module if another Microcontroller is used !!).

The variables which can be found within the project are :

CLF	:	Contains number of Watch Timer interrupts (32 ones make 1 sec)
SEC	:	Contains current second (00 to 59)
MIN	:	Contains current minute (00 to 59)
HOURL	:	Contains current hour (00 to 23)
DAY	:	Contains current day (1 to 7)
MONTH	:	Contains current month (01 to 12)
DATE	:	Contains current date (01 to 31)
YEAR	:	Contains current year (00 to 99)
DAY2	:	Shadow current day (1 to 7) !! not used !!
MONTH2	:	Shadow current month (01 to 12)
DATE2	:	Shadow current date (01 to 31)
YEAR2	:	Shadow current year (00 to 99)
LEAP	:	Boolean variable : leap year or not? LEAP:0 is used for current year YEAR
_switch	:	Debug switch to toggle LED1 !! not used !!
Check	:	Switch to decide whether running Date checking or not

3) Possible improvements of the project

In this current version, the program only calculates the date (toggling the LED0 after each second), updating the internal variables, from the second to the year (taking into account the leap ones). One restriction however is that only the years between 1996 and 2100 are considered.

The project has been written with the purpose of showing a practical handling of the Fujitsu's Software environment. Many parts can then be optimized: the C-module, for instance, is not necessary, and the algorithms used in the project can certainly be re-written in a more efficient way.

It belongs thus to the user to adapt this example to the demands of his own application in terms of accuracy, size of code, speed of execution, and consumption. Regarding this latter consideration, one possible improvement of this example could be to use the WATCH mode, waking up the Microcontroller only to update the date (!! this, of course, is only possible for Microcontrollers with an integrated Sub-Clock !!).