AN206255

F²MC/FR Family SOFTUNE Workbench How to Use Procedure File

Associated Part Family: F²MC/FR Family

This document provides examples of how to use the procedure file included with SOFTUNE Workbench. Normally, commands are executed one at a time using the "Command" window from the debug menu. However, multiple commands can be executed sequentially by using a text editor to create a file with a "*.prc" extension that contains the multiple commands and selecting "Menu" – "Open". Furthermore, the commands can be executed using a single button by registering with the "Customize Bar".

Contents

1 Introduction....1
2 System Settings....1
2.1 Customize Bar Installation....1
2.2 Procedures for Registering in Customize Bar....2
3 Procedure File Creation Method....3
3.1 Method for Creating Procedure File....3
Document History....3
Worldwide Sales and Design Support....3

1 Introduction

This document provides examples of how to use the procedure file included with SOFTUNE Workbench. Normally, commands are executed one at a time using the "Command" window from the debug menu. However, multiple commands can be executed sequentially by using a text editor to create a file with a "*.prc" extension that contains the multiple commands and selecting "Menu" – "Open". Furthermore, the commands can be executed using a single button by registering with the "Customize Bar".

This document shows how procedures such as logging the system debugger can be performed with minimal mouse operations using this function.

2 System Settings

2.1 Customize Bar Installation

For the SOFTUNE Workbench installation, open the checkbox screen for selecting the software to install from the installation screen. At this time, the Customize Bar can be registered in the SOFTUNE Workbench icons by selecting "Customize Bar".

You can register up to 10 *.prc files (hereinafter referred as to procedure files and used synonymously with procedure files in this document unless otherwise specuified).

Figure 1. Customize Bar (when not active)



2.2 Procedures for Registering in Customize Bar

Procedure files can be registered in button numbers 1 to 10 in the Customize Bar by selecting “View”, “Customize Bar”, and then “Settings” during emulation execution (This section describes the registration procedure assuming that the procedure file *.prc already exists. See the “SOFTUNE Workbench Command Reference” in the SOFTUNE help file for the procedure file creation procedure).

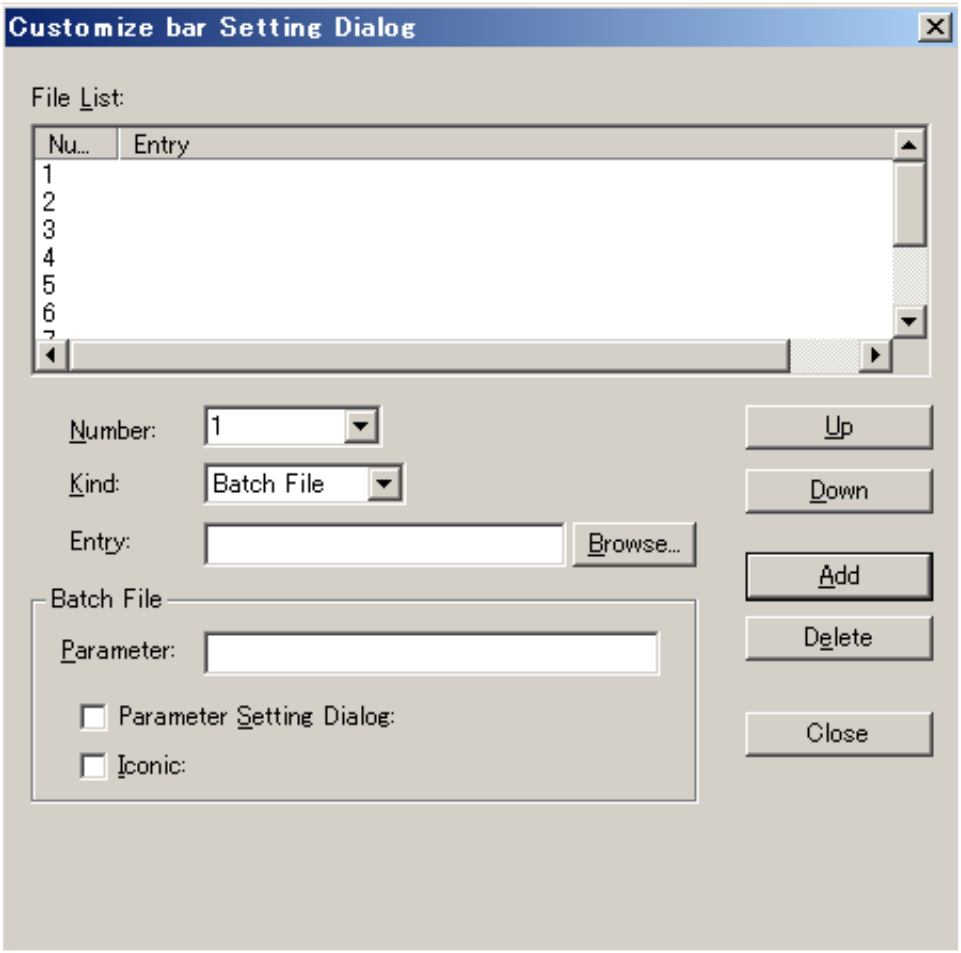
Set the “Number” to the number of the icon to be registered between 1 and 10.

Leave “Type” set to procedure file.

Specify the procedure file in “Entry”. Select the procedure file from the folder containing procedure files by using the “Browse” button.

After implementing the above operation, click the “Add” button on the right side to register the procedure file with the icon. To delete a procedure file that you have already registered, select the registration number on the same screen and click the “Delete” button to delete the registration of the procedure file from the icon (The procedure file itself is not deleted).

Figure 2. Customize Bar settings screen



When a procedure file is registered in the Customize Bar, the icon for the button changes in color and the button becomes enabled. Click on the button with the mouse to execute the procedure file.

Figure 3. Customize Bar (icon display)



3 Procedure File Creation Method

3.1 Method for Creating Procedure File

A regular text editor is required to create a procedure file.

In this section, the procedure file is created using the editor built in SOFTUNE Workbench.

First, create the text file by selecting “File”, “New”, and then “Text” from the menu.

Once you have created the new text file, select “File”, and then “Save As” from the menu, then select the directory to save the file, and attach the “*.prc” extension.

Next, write the commands in the file.

The commands that can be executed in a procedure file are detailed in the “Command Reference” in the SOFTUNE Workbench help files. For details, see this file. The following explanation is given using the SOFTUNE Workbench for the 32-bit FR family as an example. A sample procedure file is shown below.

Figure 4. Example procedure file

```
SAMPLE PRC FILE
#
RESET
SET BREAK main$23
SET BREAK main$31
GO
CLEAR TIMER
GO
PRINTF "Logging start!\n"
PRINTF "\n"
SET LOGGING/UNEXPANSION log.log
SHOW TIMER
SHOW MEMORY/BYTE 0x0000
SHOW MEMORY/HALFWORD 0x0004
SHOW MEMORY/WORD 0x0008
DUMP 0x40000..0x4001F
CANCEL LOGGING
PRINTF "\n"
PRINTF "Logging end!\n"
# PRC END
```

The following gives simple explanations of the commands in the order they are written.

The first “#” indicates that the following are treated as commands.

“RESET” is a command to issue a reset from the tool to the microcontroller.

“SET BREAK main$31” is a command to set a breakpoint.

The “main” in “main$31” refers to main.c The “$(number)” part refers to the line number within the program. That is, this command means “set a breakpoint on line 31 of main.c”.

“CLEAR TIMER” is a function to clear the timer counter of the time measurement function.

“GO” is the command to perform continuous execution, as the name suggests.

“PRINTF “Logging start!\n”” is used to display text in the command window.

Variable values can also be substituted and displayed. New lines are created by \n or \n.

“SET LOGGING/UNEXPANSION log.log” is a command to save the results displayed in the command window as a log file.

/UNEXPANSION indicates that the command display be excluded and only the results be logged.

Log.log is the name of the log file to save the results. This is normally saved in the Debug folder in the workspace.

“SHOW TIMER” is a command that displays the time measurement results.

“SHOW MEMORY” is a command that shows the value of the specified address.

/BYTE indicates description in units of bytes, /HALFWORD indicates description in units of half words, and /WORD indicates description in units of words.

“DUMP” is a function that dumps and displays the data in the specified range of addresses. The address range is specified using the notation of (starting address) to (ending address).

“CANCEL LOGGING” is a command that ends logging and saves the log file.

The procedure file shown in the example performs continuous execution of main.c from line 23 to line 31, and saves to log.log the execution time of that program, and the data at address 0x0000 (byte), 0x0004 (halfword), 0x0008 (word), and the values from address 0x40000 to 0x4001F.

The following shows the results displayed in the command window in SOFTUNE Workbench and the contents of the log file.

Figure 5. Command window display

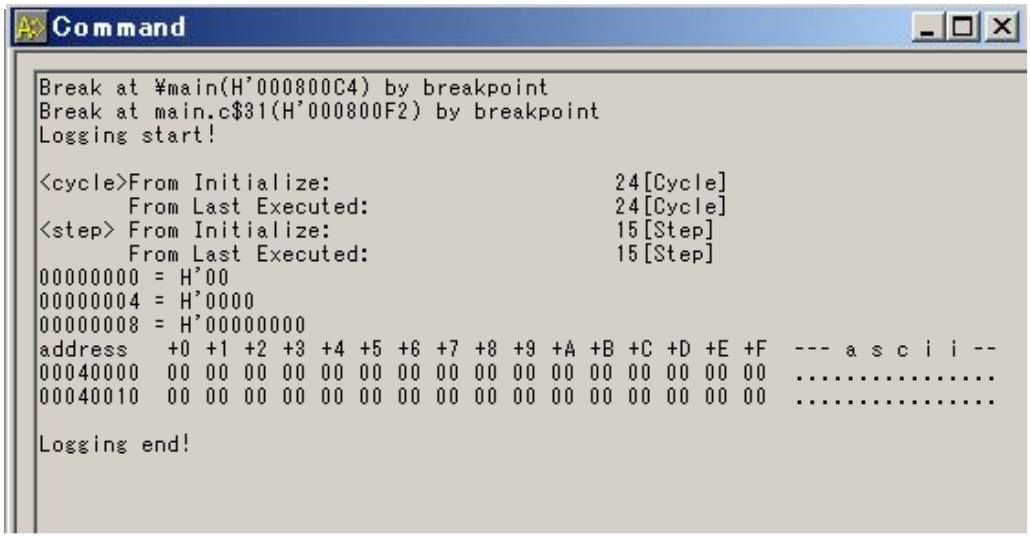


Figure 6. Contents of log.log

```
<cycle>From Initialize:                    24[Cycle]
        From Last Executed:                   24[Cycle]
<step> From Initialize:                    15[Step]
        From Last Executed:                   15[Step]
00000000 = H'00
00000004 = H'0000
00000008 = H'00000000
address    +0 +1 +2 +3+4 +5 +6 +7 +8 +9 +A +B +C +D +E +F   --- a s c i i --
00040000   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00040010   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

Procedure files can be used in this way to obtain result data for program evaluation, initialize program registers, and perform other tasks.

Document History

Document Title: AN206255 - F^{2}MC/FR Family SOFTUNE Workbench How to Use Procedure File

Document Number: 002-06255

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	NNAK	09/20/2007	Initial release. This document is Cypress 002-06255 Rev. ** from Spansion application note AN07-00105-1E. At this revision, there is no change.
*A	5879355	NNAK	09/11/2017	Migrated from Spansion format to Cypress format.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

ARM® Cortex® Microcontrollers	cypress.com/arm
Automotive	cypress.com/automotive
Clocks & Buffers	cypress.com/clocks
Interface	cypress.com/interface
Internet of Things	cypress.com/iot
Memory	cypress.com/memory
Microcontrollers	cypress.com/mcu
PSoC	cypress.com/psoc
Power Management ICs	cypress.com/pmic
Touch Sensing	cypress.com/touch
USB Controllers	cypress.com/usb
Wireless Connectivity	cypress.com/wireless

PSoC® Solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP | PSoC 6

Cypress Developer Community

Forums | WICED IOT Forums | Projects | Videos | Blogs | Training | Components

Technical Support

cypress.com/support

All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709

© Cypress Semiconductor Corporation, 2007-2017. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.