The following document contains information on Cypress products. The document has the series name, product name, and ordering part numbering with the prefix “MB”. However, Cypress will offer these products to new and existing customers with the series name, product name, and ordering part number with the prefix “CY”.

How to Check the Ordering Part Number

1. Go to www.cypress.com/pcn.
2. Enter the keyword (for example, ordering part number) in the **SEARCH PCNS** field and click **Apply**.
3. Click the corresponding title from the search results.
4. Download the Affected Parts List file, which has details of all changes

For More Information

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

About Cypress

Cypress is the leader in advanced embedded system solutions for the world's most innovative automotive, industrial, smart home appliances, consumer electronics and medical products. Cypress' microcontrollers, analog ICs, wireless and USB-based connectivity solutions and reliable, high-performance memories help engineers design differentiated products and get them to market first. Cypress is committed to providing customers with the best support and development resources on the planet enabling them to disrupt markets by creating new product categories in record time. To learn more, go to www.cypress.com.

AN205381

F²MC-8L/16LX/FR, MB9xxxx, Softune Workbench Automatic Debugger Start

This application note describes the steps necessary to setup the Softune Workbench Development Environment to automatically start the selected debugger at start-up of the Softune Workbench and to load a specified target .abs-file.

Contents

1 Introduction....1
2 Preparation of the SWB....2
2.1 Creating a new Softune Workbench Project and Workspace and adjust Workspace settings....2
3 Calling the Softune Workbench....6
3.1 Calling the Softune Workbench from the DOS command line....7
4 How to Change Debugger Environment (e.g. changing automatic testing)....8
4.1 Debug Environment and Debug Support Information File....8
Document History....10

1 Introduction

This application note describes the steps necessary to setup the Softune Workbench Development Environment to automatically start the selected debugger at start-up of the Softune Workbench and to load a specified target .abs-file.

The Softune Workbench Development Environment (from here on only called Softune Workbench of SWB) is to be used to control a debugger. Debugging can be done using

- Code Simulator
- Monitor Debugger
- Emulator Debugger using ICE

Automatic start of debugging at start of the SWB might be used for example in the field of automatic testing of complete applications or when one wants to use a software development environment other than Softune Workbench but still wants to be able to debug the software for example using emulator debugging via ICE (In Circuit Emulator).

2 Preparation of the SWB

The steps shown in this chapter show how to setup a dummy Softune Workbench project together with a corresponding Softune Workbench workspace containing the aforementioned project. This SWB project/workspace can be used for debugging when for example the tool chain for building the application software is different from the Softune Workbench.

Of course, the steps after creating a new Softune Workbench project/workspace can surely be applied to an existing Softune Workbench project/workspace if one wants to have the current Softune Workbench project/workspace to be re-opened automatically at start of the Softune Workbench (even without automatic start of the debugger).

2.1 Creating a new Softune Workbench Project and Workspace and adjust Workspace settings

In the following it is assumed that the Softune Workbench (together with necessary drivers if needed) is already installed.

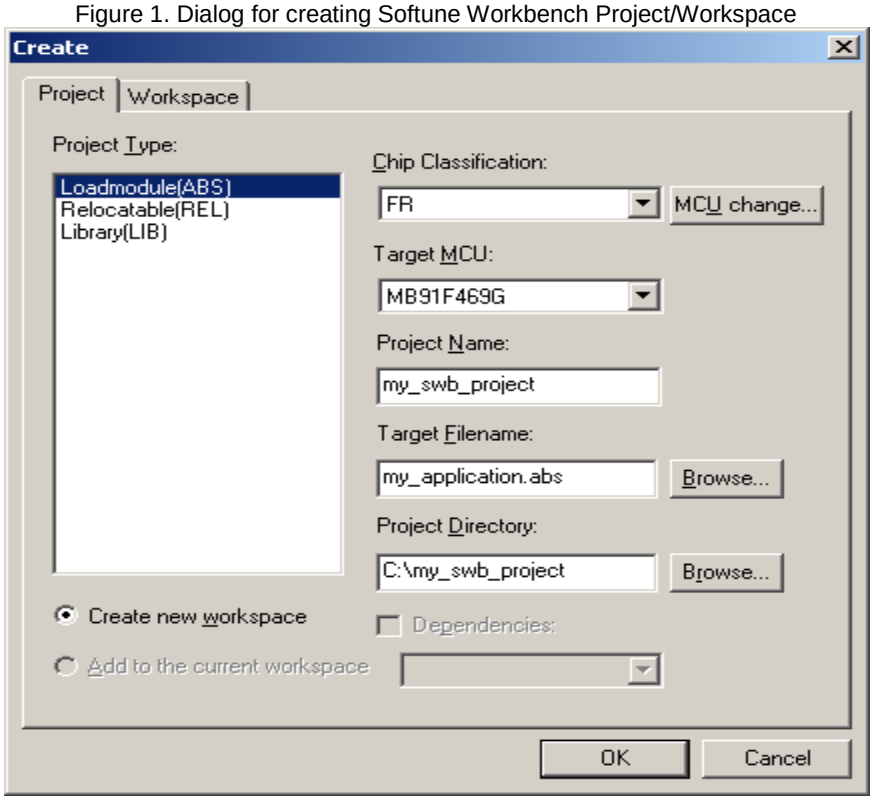
The directory of the target .abs-file (generated from within a different Softune Workbench project or generated by another tool chain using the Softune Workbench language tools) is `C:\my_application_project\ABS` (choose this path corresponding to your needs). The name of your application software target .abs file is assumed to be `my_application.abs`.

2.1.1 Create SWB Project/Workspace and select .abs-file to be used

To create a new SWB project/workspace from scratch use File -> New -> Workspace/Project file after start-up of the Softune Workbench IDE. This will display the dialog windows as shown in

. Fill in the needed information (selection of Chip, Target MCU, Project Name, Target File Name and Project Directory).

Figure 1. Dialog for creating Softune Workbench Project/Workspace



Please choose Project Name and Project Directory corresponding to your needs. Please choose Target Filename corresponding to the name of the .asb-File to be used with this SWB Project/Workspace.

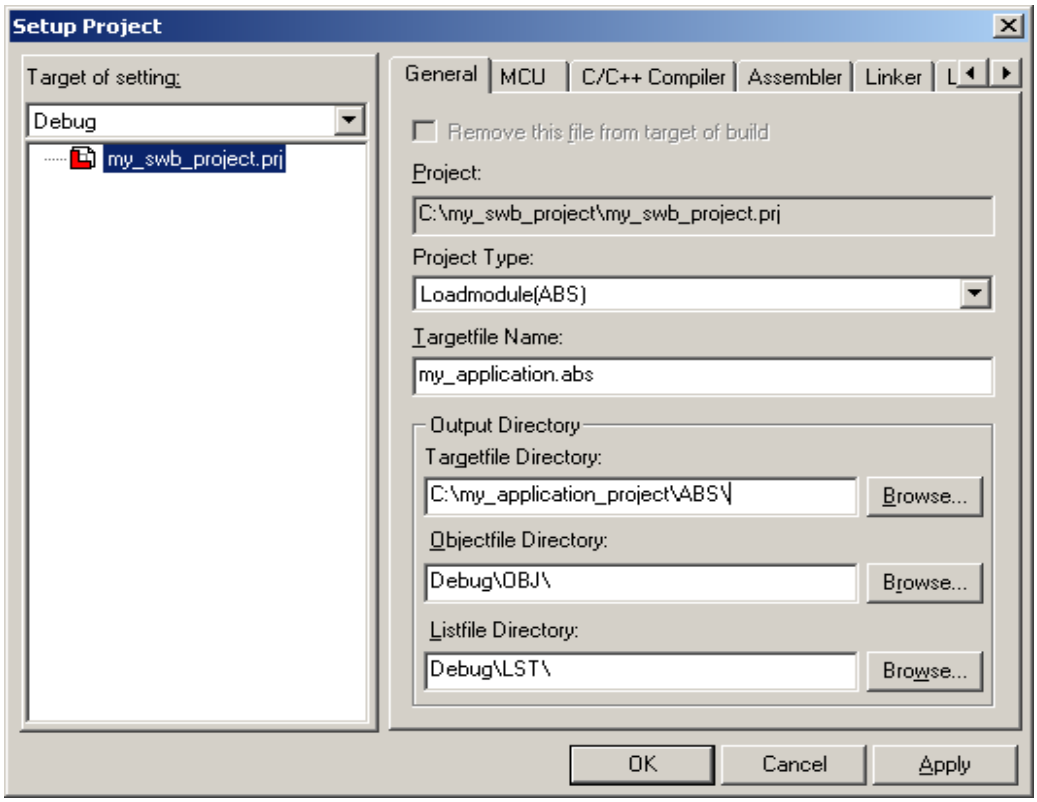
This will create (if not already existing) a directory/file structure like:

```
C:\my_swb_project -|
                   |_ Debug -|
                   |         |_ my_swb_project.sup
                   |         |_ ABS
                   |         |_ LST
                   |         |_ OBJ
                   |         |_ OPT
                   |
                   |_ my_swb_project.dat
                   |_ my_swb_project.prj
                   |_ my_swb_project.wsp
```

The .prj-file contains the settings of the project and the .wsp-file contains the settings of the workspace. These two files are normal text files which might also be generated by using a script tool and which of course can also be edited/changed by using a normal text editor.

Since the current directory of the target .abs-file is set by default to C:\my_swb_project\debug\abs\my_application.abs this has to be adapted to the real path to the .abs-file (`C:\my_application_project\ABS\my_application.abs` in this example). Please use the dialog shown by selecting Project -> Setup Project … -> General from the Softune Workbench menu within the my_swb_project project (as shown inFigure 2) to set the path information accordingly.

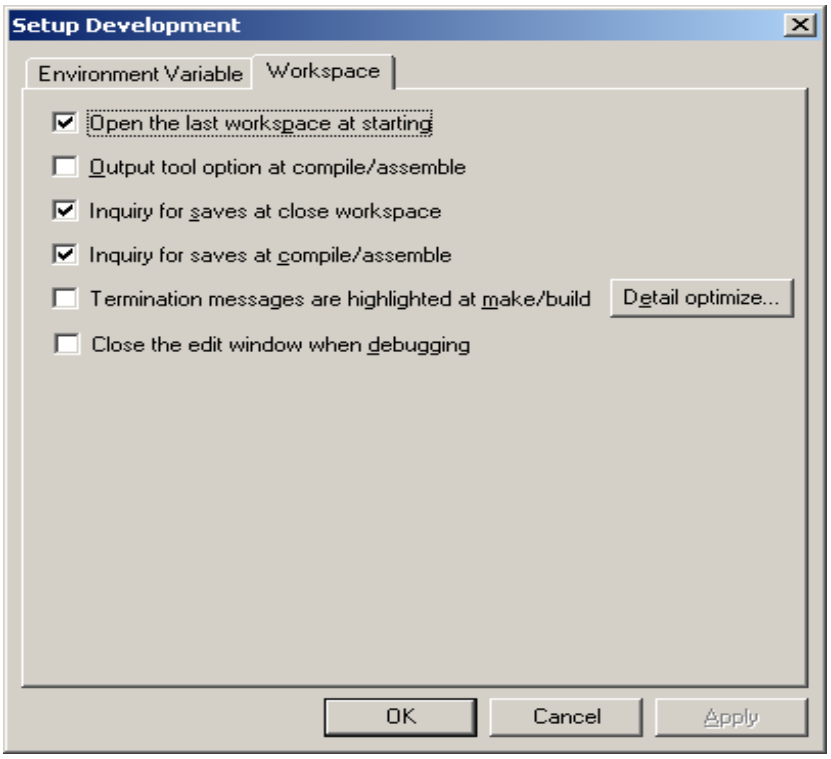
Figure 2. Dialog to change the Path to the application .abs-file to be used within my_swb_project



2.1.2 Adjust SWB Project/Workspace settings for re-opening at SWB start

The Softune Workbench can be configured to re-open automatically at start-up the workspace that was used when the Softune Workbench was closed. This feature is enabled by checking the option “Open the last workspace at starting” in the dialog shown when selecting Setup -> Development… -> Workspace from the Softune Workbench Menu.

Figure 3: Dialog opened via Setup -> Development... (selecting Workspace-Tab).



2.1.3 Adjust SWB Project/Workspace settings for automatic debugger start at SWB start

A Softune Workbench workspace can be configured to start the debugger using the last selected .sup-file when the workspace was opened by the Softune Workbench. To select this configuration one has to open the according configuration dialog via the Softune Workbench menu Project -> Setup Workspace and check the option “Start Debugger” under “Debug when workspace is open” (please refer to

Figure 4).

2.1.4 Setup Debug Environment

When creating the new Softune Workbench Project/Workspace “my_swb_project” a default debug environment my_swb_project.sup was created (please refer to the “debug”-folder in the project view of the Softune Workbench). The settings of this .sup-file have to be adjusted to the corresponding debugger tool (simulator or emulator debugger) and the behaviour of the debugger tool at debugger start has to be setup following the instructions in dialog opened when right-klicking the my_swb_project.sup and selecting the menu option “Change…” (please refer to Figure 5).

Figure 4: Workspace configuration dialog having "Start Debugger" at Workspace opening option selected.

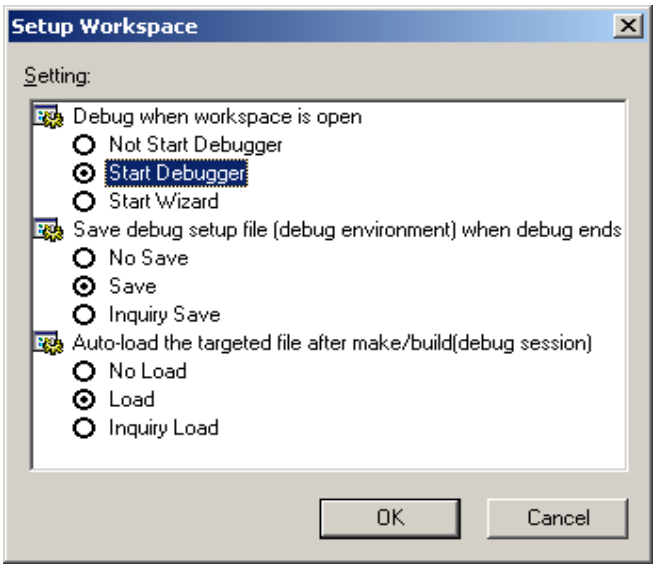
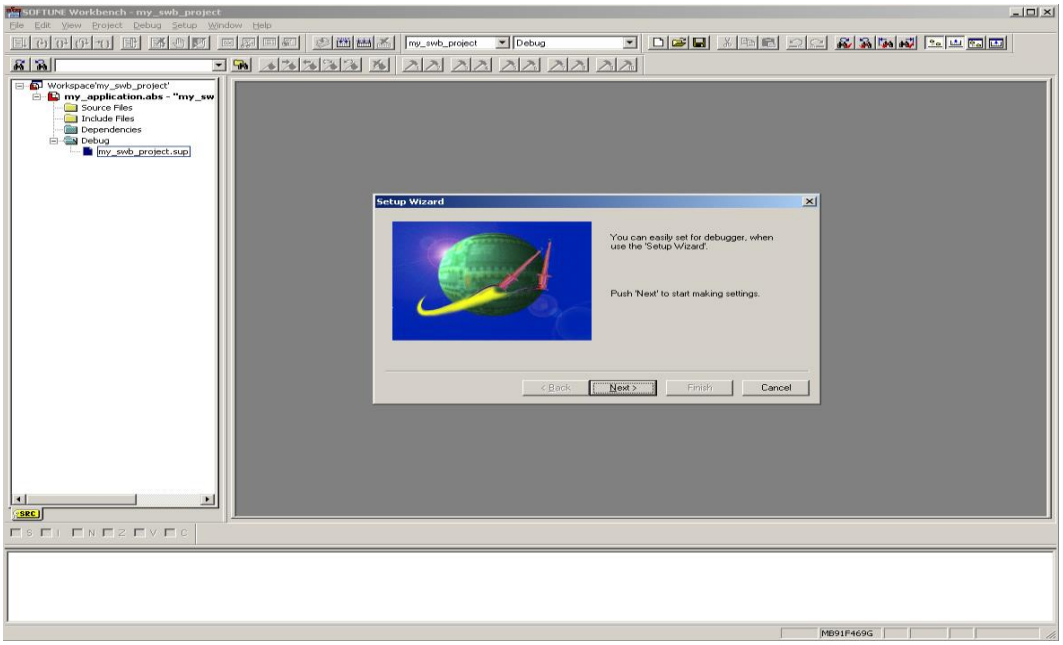


Figure 5: Start Dialog for selecting and setting up the debugger tool for my_swb_project.sup (please follow the instructions given in the corresponding dialogs).



After finishing the necessary configuration steps the Softune Workbench has to be closed without closing the current Softune Workbench Workspace (`my_swb_project` in this example). One has to agree to the corresponding dialog to save the done changes to the current Softune Workbench Workspace.

3 Calling the Softune Workbench

In the following it is shown how to call the Softune Workbench (automatically starting the debugger and loading the target .abs-file.

3.1 Calling the Softune Workbench from the DOS command line

The Softune Workbench can be called from the command line using the following command (assuming for this example the FR Softune Workbench V6 was installed to the directory C:\Softune6 on the PC).

```
C:\Softune\fs911s.exe
```

When the preparation steps in chapter 2 have been followed appropriately the Softune Workbench will start and automatically load the last used Workspace (my_swb_project in this example) and finally proceed with starting the debugger tool as selected in `my_swb_project.sup`.

Application Notes related to Softune Workbench and/or Debugging/Emulation for further reading:

- AN205222 - FR Family MB2198-01 Emulator System Getting Started Guide
- AN204828 - F2MC-16FX Family, Emulating and Debugging with Softune and MB2198-01
- AN205205 - FR Family MB91460, SWB Monitor Debugger
- AN205224 - How to use Customize Build for Softune WB

4 How to Change Debugger Environment (e.g. changing automatic testing)

To run different test procedures for example in automatic application test one might want to change the procedure files called. The following shows which part of the .sup-file defines the procedure files called by the debugger.

4.1 Debug Environment and Debug Support Information File

The Softune Workbench basically has two files containing information about the debugging environment. First which debugger environment to be called (information stored in the .prj-project file) and second (when the debugger environment is selected) the settings for a given debugger environment (debug support information .sup-file).

These files are text files which can be manipulated by general text file editors. If one needs to have different debugger environments or if one wants to change test procedures (e.g. different tests for automatic application test using debugger) these files can be changed prior to the start of the Softune Workbench for example by running a script manipulating the corresponding files.

4.1.1 Selecting the Debug Environment

The .prj-project file contains the information about the list of available .sup-files [DebState-xyz] (xyz is the used SWB project configuration) and the selected actual debug environment (.sup-file) (Current) which will be used when the debugger is started:

…

[DebState-STANDALONE]

SupCount=6

Supfile-0=STANDALONE\Simulator.sup

Supfile-1=STANDALONE\MB2198-01-USB.sup

Supfile-2=STANDALONE\MB2198-01-COM1.sup

Supfile-3=STANDALONE\MB2198-01-LAN.sup

Supfile-4=STANDALONE\Mondeb_57K6_com1.sup

Supfile-5=STANDALONE\Mondeb_57K6_com2.sup

Current=MB2198-01-USB.sup

AliasFile=

CPURunMode=

…

4.1.2 Changing the Command Procedure Files

Besides other information the Debug Support Information .sup-file contains information about the command batch files to be executed

- At debugger start
- Before loading of .abs-file
- After loading of .abs-file

The following shows a part of an .sup-file executing no command batch file once at debugger start and `/PRC/set_ROMs.prc` before downloading .abs-file and in addition `/PRC/set_RESET.prc` after download of .abs-file (the path is relative to the position of the .prj-file of the corresponding Softune Workbench project):

```
[Target]
  …
Before Batch File=PRC\set_ROMS.prc
After Batch File=PRC\set_RESET.prc
  …
[Start]
Batch File=
  …



-- END --
```

Document History

Document Title: AN205381 - F²MC-8L/16LX/FR, MB9xxxx, Softune Workbench Automatic Debugger start

Document Number:002-05381

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	-	MKEA	06/19/2008	V1.0 RSchum Initial Version
			10/20/2008	V1.1 RSchum Error corrected and info about .sup-file added
*A	5068030	MKEA	01/06/2016	Converted Spansion Application Note “MCU-AN-300097-E-V11” to Cypress format
*B	6070279	NOFL	02/13/2018	Sunset Review Migrated to new template Updated links

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 MCU

Cypress Developer Community

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

Technical Support

cypress.com/go/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, 2008-2018. 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. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress does not assume any liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. 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.