

Static Code For Init Components

Processor Expert

Processor Expert software allows fast creation of drivers customized to application requirements without the need to have knowledge of silicon. With the help of Eclipse based graphical interface where you can configure your driver and generate corresponding source code. However, in some cases fully generated source code is not acceptable, especially in applications with an emphasis on safety, applications where code certification is a requirement or if a software developer wants to have a full control over the application code.

Contents

1	About this note	2
2	Static initialization code.....	2
3	Summary.....	5

1 About this paper

This paper introduces new Processor Expert features:

- Static initialization code
- Local and shared static code repository
- New embedded initialization components

The importance of software in today's world of embedded solutions brings more demand on software tools side and variety of requirements they should meet. Let us look at two examples of embedded application developers and their demands:

- Developers who do not want to bother with all the hardware complexity or who are interested in specific functionality. For example, developers who want to initialize silicon to certain state and run the application on it or to use Processor Expert embedded components just to create communication interface between microcontroller and external devices.
- Developers who know the silicon that they use and want to have a full control over the code. For example, developers of motor control applications that require the safety standards.

Processor Expert usability for users from the second group was limited however, new Processor Expert features offers solutions for applications where generated code is not acceptable and full control over the code is necessary. Now, we can offer static initialization code with more transparent coverage of MCU peripherals using Processor Expert embedded initialization components.

2 Static initialization code

For supported microcontrollers, Processor Expert embedded initialization components now use static peripheral initialization driver. Each instance of MCU peripheral has its own initialization driver placed in static (not generated) module. This guarantees easier certification of application developed with Processor Expert while preserving initialization components graphical interface to which users are used to. The only difference is that initialization components added to project no longer generate full source code. Instead, only configuration header file included in static initialization driver is generated.

Also, hardware startup (controlled by Processor Expert processor (CPU) component) is part of the new static initialization drivers along with processor component runtime methods. As processor memory map and Peripheral Device Drivers (PDD) are already part of Processor Expert static library, developers are now able to create applications not containing any generated source modules. From now, Processor Expert can be used to create certification-aware software applications.

Another advantage of the static initialization drivers is that their source code is placed in common `.c/.h` modules so software developer is able to modify Processor Expert peripheral initialization driver and customize it according to application requirements.

2.1 Local and shared static drivers' repository

The new feature introduced with improved Processor Expert static code support is possibility to choose where the static drivers should be stored. For each project there are two options, store complete set of static drivers directly in the project directory or link the project with common static drivers repository stored in Processor Expert installation directory.

Standalone projects with static drivers stored directly in the project directory are used when it's necessary to have the projects independent of each other and from development environment. Changes made in drivers in one project don't affect rest of the projects. Also, standalone project allows easier application distribution as all the project source modules are in one place.

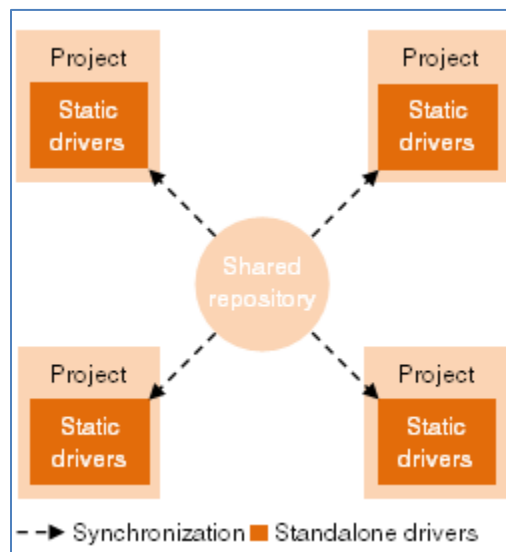


Figure 1 Example of standalone project

Although in this case static library is independent from currently used development environment developer still has option to synchronize project's local repository with the common one, for example in case that the development environment is updated with a newer version of Processor Expert static drivers. Processor Expert detects changes in the shared repository, informs user about them and offers project local repository update. It is possible to review all the changes in the code and decide to accept or deny update for each static source file separately. If the update is denied it is still possible to synchronize repository later.

Shared static drivers repository is advantageous when more projects should share the same version of static drivers. In this case, static drivers are not physically placed in the project directory but each project is virtually linked with shared, common repository. This way the management of the projects' drivers can be done in one place and any changes made in the shared repository is automatically distributed across all of the linked projects, for example in case of bug fixing or library update and also backup or archiving of the static drivers versions is very simple.

It is of course possible to combine both approaches when some of the projects use shared repository and others local repositories.

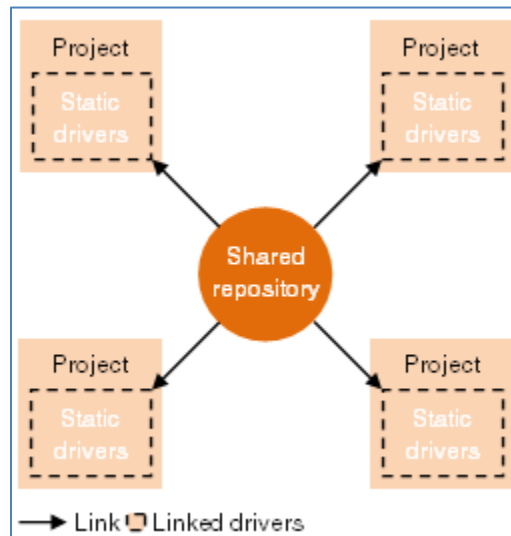


Figure 2 Example of linked project

2.2 New embedded initialization components

Previously, some of the MCU peripherals (typically system peripherals like System Integration Module or Interrupt Controller) were not covered by distinct Processor Expert initialization component, their configuration was controlled by processor (CPU) component or distributed among multiple components. Such fragmentation is not always clear and is not suitable for developers familiar with hardware they use and those who orient in chip configuration using reference manuals. To help developers in creating their application faster and smoother, Processor Expert now allows to configure each peripheral using its own initialization component (with exception for system and bus clock generator peripherals, whose control remain with processor component, as they are part of Processor Expert timing model and their initialization is done during the startup). However, it is still possible to configure settings like pin routing or interrupts where they are needed. Both possibilities are explained in the following example.

Let us have an application that uses A/D converter with measurement synchronized with external device connected using pin on chip package. Routing of such pin can be a complex job, for example, switch pin functionality from GPIO function to peripheral function (controlled by GPIO peripheral), if pin has multiple peripheral functions choose the right one (controlled by System Integration peripheral) and connect the pin with right peripheral (controlled by Crossbar Switch peripheral) and connect the pin with right peripheral (controlled by Crossbar Switch peripheral).

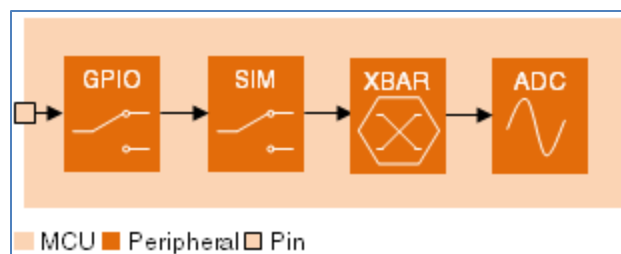


Figure 3 Example of ADC synchronization pin routing

The developer whose application uses pin (interrupt, peripheral's clock gate enable or any similar resource) and do not want to bother with its routing settings can make the settings specific for peripheral that uses the resource (for example, the synchronization pin can be selected in the ADC initialization component). In such case, Processor Expert takes care of initialization code of functionalities external to configured peripheral and generates it (in example, the code initializing particular GPIO, SIM, and XBAR registers is generated).

If developer requires full control over the code it is possible to add to the project initialization components for all peripherals involved in routing of some resource and make the settings manually. Then Processor Expert will not generate any code and application contains only static source code.

3 Summary

Processor Expert now offers solution for applications where generated code is not acceptable. Initialization of processor is possible using static, optimized and open drivers preserving all benefits of rapid development tool. New features include:

- Static driver used in all initialization drivers
- Projects with local or shared static drivers repositories
- New initialization components

Processor Expert with these new features supports different development approaches, from small demonstration applications to complex embedded solutions with safety critical aspects or applications requiring software certification.

Full coverage of these new features is available for selected families.

How to Reach Us:**Home Page:**

www.freescale.com

Web Support:

<http://www.freescale.com/support>

USA/Europe or Locations Not Listed:

Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:

Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:

Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:

Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 010 5879 8000
support.asia@freescale.com

For Literature Requests Only:

Freescale Semiconductor Literature Distribution
Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals", must be validated for each customer application by customer's technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standards and terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, CodeWarrior, and Processor Expert are trademarks of Freescale Semiconductor, Inc. Reg. U.S. Pat. & Tm. Off. All other product or service names are the property of their respective owners.

© Freescale Semiconductor, Inc. 2013. All rights reserved.