

Version

1.0

PROCESSOR EXPERT FOR MOTOROLA HC08 FAMILY

UNIS

Processor Expert User Guide

**PROCESSOR EXPERT FOR MOTOROLA HC08 FAMILY
SOFTWARE DEVELOPMENT TOOL FOR HC08 DESIGNERS**

Processor Expert User Guide

© 2002 UNIS, spol. s r.o. ALL RIGHTS RESERVED
Processor Expert is registered trademark of UNIS, spol. s r.o.
All other brands or trademarks are property of their respective holders.

Table of Contents

CHAPTER 1	3
About the Product	3
Software Product Description	3
Benefits	3
CHAPTER 2	4
Installation	4
System Requirements	4
Installation Step-by-Step	4
CHAPTER 3	5
Registration	5
About the Registration	5
Registration Step-by-Step	5
CHAPTER 4	6
Getting Started	6
About the Tool	6
Typical Development Steps	6
Supported Beans	9
Important Information for Beginners	12
Information for Advanced Users	12
CHAPTER 5	13
Support	13
Metrowerks	13
WWW	13
Emails	13
CHAPTER 6	14
Contacts	14
CHAPTER 7	15
Revisions	15

Software Product Description and Benefits

About the Product

SOFTWARE PRODUCT DESCRIPTION

This release package contains software tools to support development of applications for Motorola HC08 series of microcontrollers.

PROCESSOR EXPERT STAND-ALONE (DEMO VERSION): is an application builder for HC08 microcontrollers, including Embedded Beans, reference projects, examples, tutorials and electronic documentation. This version of Processor Expert supports several derivatives of Motorola HC08 Family and compiler Metrowerks CodeWarrior for HC08 (please refer to README.TXT for details).

PROCESSOR EXPERT PLUG-IN: is the Processor Expert™ environment integrated into the Metrowerks CodeWarrior IDE application.

BEAN WIZARD (OPTIONALLY): is a tool for Beans creation including Beans Creator, examples and electronic documentation.

BENEFITS

- Integrated development environment increases productivity and saves time and cost.
- Usage of the tool minimizes time to learn the microcontroller.
- Processor Expert allows rapid prototyping of entire application.
- Processor Expert provides modular, reusable and fully tested functions.
- It's easy to modify and port the existing implementations using Processor Expert.
- Processor Expert supports all on-chip peripherals and provides design-time settings verifications.

System Requirements, Installation Process Step-by-Step

Installation

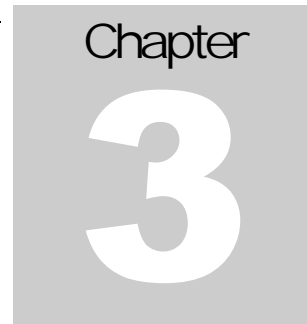
SYSTEM REQUIREMENTS

- PC Compatible Machine with Pentium processor or higher
- Microsoft Windows 98/NT 4.0/2000
- Service Pack 2 if you have Windows 2000
- 64MB RAM
- 150MB free space on HDD
- Color graphic display with resolution 800x600, 1024x768 strongly recommended
- Mouse or other compatible pointing device
- Internet Explorer 4.01 or higher installed
- Metrowerks CodeWarrior IDE 4.2.6 with HC08 V2.0
- Acrobat Reader 5.0
- xpdf package installed in subdirectory xpdf\
- Administrator rights in Windows NT and Windows 2000

Please refer to README.TXT for details.

INSTALLATION STEP-BY-STEP

Follow the CodeWarrior installation process (see CodeWarrior Users Guide).



Descriptions of the Registration Process – Step-by-Step

Registration

ABOUT THE REGISTRATION

Customers, who register CodeWarrior with Processor Expert plug-in for HC08 are automatically registered for using the Processor Expert Plug-in for HC08. Registering CodeWarrior with Processor Expert plug-in for HC08 customers agree that their registration data shall be transferred to UNIS, spol. s r.o.

REGISTRATION STEP-BY-STEP

Follow the CodeWarrior registration process (see CodeWarrior Users Guide).

First-time Usage of the Processor Expert

Getting Started

ABOUT THE TOOL

PROCESSOR EXPERT for MOTOROLA HC08 FAMILY (**PE**) is an Integrated Development Environment (IDE) for design, implementation, verification, optimization of the HC08 based embedded microcontroller applications. This high productivity development platform allows efficient use of the HC08 microcontrollers and its peripherals, allows building of portable solutions, and saves development time and cost.

Processor Expert has built-in knowledge (internal definitions) of the entire HC08 microcontroller with all its integrated peripherals. Functional elements are encapsulated into concepts of configurable components (called Beans), for which method and event subroutines are provided. An intuitive and powerful graphical UI allows the user to define the system's desired behavior using Bean component functional settings, and drag & drop selections. The IDE includes utilization of the generated methods and event subroutines, along with the combining of generated code and user written code as appropriate.

TYPICAL DEVELOPMENT STEPS

This chapter describes typical development process (step-by-step) and contains recommendations/tips for your first application:

1. **Create new project (use project template), and configure CPU Bean (properties, methods and events)**
 - Start Processor Expert from Window's Start menu or start CodeWarrior with PE Plug-in.
 - Use command `main menu | File | New Project` to create new application.

- Processor Expert stand-alone: Select project template. Each project template usually contains two configurations of the MCU: *Debug* and *Release*. The only difference is in setting of property *Cpu mode selection*. Processor Expert Plug-in: Select HC08 Stationery and desired derivative in the ProcessorExpert folder.
 - The target CPU Bean is displayed in the Project Panel. Double-click on the Bean icon to select it and open Bean Inspector, which allows customizing the Bean settings. The default setting of the CPU Bean is suitable for most applications.
2. **Find the closest Bean for supporting desired function – by clicking on the Bean Palette tabs (or Bean Selector in the Plug-in) and reviewing available Beans and selecting / adding the matching Bean into the project**
 - Processor Expert project consists of building block called **Embedded Beans (Bean)**. Each Bean represents encapsulation of functionality of any on-chip peripheral, external peripheral, virtual peripheral or some software algorithm. See chapter "Supported Beans " for more information about supported Beans and their functionality.
 - Click on the Bean name on the Beans Palette/Bean Selector and the Bean will be added into your Project Panel.
 3. **In the Bean Inspector, configure the behavior by setting Bean properties, and deciding about methods and events to be included into the project during code design/generation**
 - Bean Inspector contains Properties, Methods and Events of the selected Bean.
 - **Properties** define the Bean connection to the target CPU, settings of allocated peripherals, initialization of the Bean and other Bean run-time options.
 - **Methods** are the functions, which are generated from the Processor Expert and are prepared for the user utilization. Methods are located in the Bean driver, which should not be changed by the user.
 - **Events** are the functions templates, which are called from the Beans driver like an interrupt service routines. Events are generated into the event module and should be implemented by the user.
 4. **Repeating steps 2 and 3 realize all needed functions by utilizing the support of the predefined Beans**
 5. **In case the tool displays exclamation marks, correct necessary Bean settings to remove those**
 - Processor Expert provides powerful design-time verifications, which protects users from the run-time problems and bugs. Error messages are displayed in the third column of the Bean Inspector or in the Error Window.

- **Peripherals settings:** Processor Expert verifies if the selected peripheral meet the Bean requirements.
 - **Peripheral allocation:** Processor Expert also provides a protection against usage of one peripheral by several Beans. If the peripheral is used by one Bean, its settings cannot be modified by any other Bean.
 - **Timing:** The settings of all timed devices are checked. If no error is detected, it means that you can work with the given settings without problem in run-time. Generated drivers are provided in such a way that they cannot influence each another - it means that, even if several Beans share one physical device, changes in the Bean settings using the values given (and checked) by the Processor Expert is possible in run-time without affecting any other Bean.
6. **Initiate a code design and check the status window. If the code design was successful (no errors), proceed with the next step**
- Use command `main menu | Code Design | Code Design` in the stand-alone version OR `main menu | Processor Expert | Code Design` in the plug-in version. All the Bean drivers are generated during "code design" process, and the event and main modules are updated (or created if does not exist).
 - The code design can be finished successfully only if the project is set correctly.
7. **Utilize Bean methods by implementing their calls from the main function (in module "project".c), from any event functions ("event".c) or from the modules implemented by the user from scratch. "List of methods" is available in CPU's pop-up menu in the Project Panel window. User written code can be also added into the main, event and user written modules**
- Use built-in editor to write your own source into the generated events and main functions. The main function is located in the module with the same name as the project and is called immediately after initialization of the CPU and Beans. Event handlers are located in the Events module. Do not change Bean drivers as they are regenerated each time you select `Code Design`.
 - In the stand-alone version you can use drag & drop of methods from the Project Panel into the source code. The invocation of the method will be inserted into the source text.
8. **Build the application and correct compiler errors**
- CodeWarrior integrated development environment allows you to build your application. Use command `main menu | Project | Make` to compile your application.
 - Processor Expert stand-alone allows you to invoke command-line version of CodeWarrior C Compiler and Linker and display the error messages in the Error Window. Use command `main menu | Tools | Build` to compile your application. Settings for the compiler can be reviewed in the CPU Bean Inspector.

9. If you need to add more functionality to your project, go back and continue with step 2

- Bean Inspector can hide the advanced settings of the Bean using the predefined "view levels": basic, advanced and expert (see `Bean Inspector | View`). Basic level is selected by default and you can get more options for the Bean by selecting a higher level.
- You can add also external modules to your project. Use commands `Project Panel | Programs (popup menu) | Add external program` Or `Project Panel | Programs (popup menu) | New external program` (pop-up menu is opened on right mouse button click on the appropriate object).

SUPPORTED BEANS

Embodied Bean is an component which can be used in the Processor Expert. Embedded Beans encapsulate functionality of basic elements of embedded systems (like CPU core, CPU on-chip peripherals, standalone peripherals, virtual devices, and pure software algorithms) and provide the control of these devices via properties, methods, and events (like objects in Object Oriented Programming). Processor Expert generates drivers for Metrowerks C Compiler from each Bean. The generated driver contains initialization of the Bean and peripheral and optimized code of methods, according to user settings of the Bean.

All the Beans are located on the **Beans Palette**, in the Processor Expert's stand-alone Main Panel or in the **Bean Selector** (in both versions). The Bean can be added to the project by clicking on the button with the Bean name. There are the following pages on the Beans Palette, which can be selected by clicking on the appropriate tab:

CPUs

The CPUs page of the Beans Palette contains beans of all supported CPU Beans. This version of Processor Expert supports many derivatives from Motorola HC08 Family, for complete list of supported derivatives please refer to the file README.TXT.

Input/Output

The Input/Output page contains beans, which encapsulate Input/Output peripherals, e.g. I/O pins, I/O ports and external interrupts.

- **BitIO** – 1-bit input/output using one I/O pin.
- **BitsIO** – multi-bit input/output using 1-8 contiguous I/O pins.
- **ByteIO** – byte input/output using one 8-bit I/O port.
- **Byte2IO** – 16-bit input/output that uses two 8-bit I/O ports.
- **Byte3IO** – 24-bit input/output that uses three 8-bit I/O ports.
- **Byte4IO** – 32-bit input/output that uses four 8-bit I/O ports.
- **ExtInt** – external interrupt bean generates event caused by signal edge (or level) of the pin (based on the hardware interrupt).

- **KBI** – keyboard interrupt module.

Timers

The Timers page contains all beans regarding timer functions, e.g. generation of periodic interrupt, support for time and date, generation of output signal and programmable pulses, counting of external events, capture functionality and watchdog.

- **TimerInt** – periodic interrupt generator. User's subroutine (event) is called at the end on each period. Interrupt period is independent on the selected speed mode.
- **TimeDate** – implements real time clock, with support for date and alarm.
- **TimerOut** – periodic signal generator with fixed duty-cycle ratio 50 %.
- **PWM** – programmable pulse generation with variable duty.
- **PPG** – programmable pulse generation with variable duty and period.
- **EventCnt8** – counts external events specified edge in 8-bit counter.
- **EventCnt16** – counts external events specified edge in 16-bit counter.
- **EventCnt32** – counts external events specified edge in 32-bit counter.
- **Capture** – encapsulates capture peripheral. Counter value is caught in the capture register on the external signal and optional event is generated.
- **RTIshared** – implements virtual periodic interrupt generator, which can be used (shared) by other Beans as a timer source.
- **Watchdog** – encapsulates WatchDog and Clock Monitor.
- **FreeCnt8** – provides time measurement in 8-bit counter with selectable precision.
- **FreeCnt16** – provides time measurement in 16-bit counter with selectable precision.
- **FreeCnt32** – provides time measurement in 32-bit counter with selectable precision.

Communication

Communication page contains the following communication beans: asynchronous and synchronous serial communication via SCI and SPI, and support of CAN.

- **AsynchroSerial** – asynchronous serial communication on SCI. The bean supports interrupt or polling mode, attention mode, RTS and CTS handshake, break signal and more.
- **AsynchroMaster** – master side of asynchronous serial communication with address bit and wake-up function. Simple protocol included.
- **AsynchroSlave** – slave side of asynchronous serial communication with address bit and wake-up function. Simple protocol included.
- **SynchroMaster** – master of the synchronous serial communication via SPI.
- **SynchroSlave** – slave of the synchronous serial communication via SPI

- **MotorolaCAN** – encapsulation of the CAN protocol.
- **Term** – bean for ANSI terminal emulation. This bean inherits `AsynchroSerial` bean, which implements serial asynchronous communication.

Converters

Page “Converters” contains beans encapsulating the A/D converter peripheral. The page contains three beans, which operates in different modes.

- **ADfast** – encapsulates A/D converter. The bean generates an optimized code, because it does not allow sharing of A/D converter. The bean may measure all channels in continuous mode.
- **ADC** – encapsulates A/D converter. The bean generates an optimized code, because it does not allow sharing of A/D converter. The bean is optimized for measurement of single channel.
- **ADconverter** – encapsulates A/D converter. The bean allows sharing of A/D channels between several `ADconverter` beans.

Programming Support

Programming Support page contains beans for internal EEPROM memory, service of any interrupt vector handler and software beans data conversions and terminal.

- **IntEEPROM** – implements access and programming of the internal MCU EEPROM memory.
- **ExternalFile** – allows including external binary file (table, database, sound, etc.) to the project. The content of the file is transformed to the format required by the compiler.
- **StringList** – allows including text to the project and accessing it by strings.
- **BWImage** – converts the two-color black and white picture into the simple internal data structure, usable by other beans (graphical displays).
- **COLImage** – converts color picture into internal data structure.
- **InterruptVector** – user allocation and service of any interrupt vector in the MCU vector table.

IMPORTANT INFORMATION FOR BEGINNERS

You can find much important information in the documentation provided with the tool in the Processor Expert stand-alone main menu or CodeWarrior main menu item Processor Expert

- **Tutorial Courses** – step-by-step procedure how to build the first application in the Processor Expert.

Help | Tutorial

- **Quick Start** – recommendations how to learn the Processor Expert quickly.

Help | Quick start

- **Terms and Definitions** – explanation of used terminology.

Help | Quick start

- **List of Supported Beans** – list of supported Beans with detailed description of the Bean features, properties, methods and events.

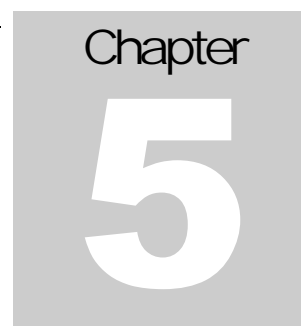
Help | Embedded Beans

View | Installed Beans Overview

If you need to know more, please refer the included electronic documentation or contact our support department.

INFORMATION FOR ADVANCED USERS

- **Test projects** - test projects are designed for user who is starting with Processor Expert. Using these projects user can easily test functionality of the Embedded beans and internal peripherals of HC08 family. Each bean has one test project located in `Projects\Test projects` subdirectory.
- **PESL** - Processor Expert System Library is a useful supplement of the Embedded bean's concept. Using this library user can access to all functionality of internal peripherals. See also example in subdirectory `Projects\PESL`.
- **Version specific information for HC08** – each implementation of the Processor Expert is customized according to the features of the supported microcontrollers and compilers. "*Version specific information*" contains detail information about the generated code, directives, tool support, etc.
- **Bean Wizard** is a tool for a Bean design. The usage of the tool requires a good skill of work with Processor Expert.



Support Services

Support

METROWERKS

To obtain assistance with Processor Expert for Motorola HC08 family, please contact the Metrowerks Technical Support Groups.

WWW

www.metrowerks.com/support

EMAILS

support@metrowerks.com
support_europe@metrowerks.com

Contacts

UNIS

UNIS, spol. s r.o.

Jundrovská 33

624 00 BRNO

Czech Republic

Tel. +420-5-41 515 111

Fax +420-5-41 210 360

e-mail: info@processorexpert.com, sales@processorexpert.com

web: <http://www.unis.cz/EN>, <http://processorexpert.com/hc08>

