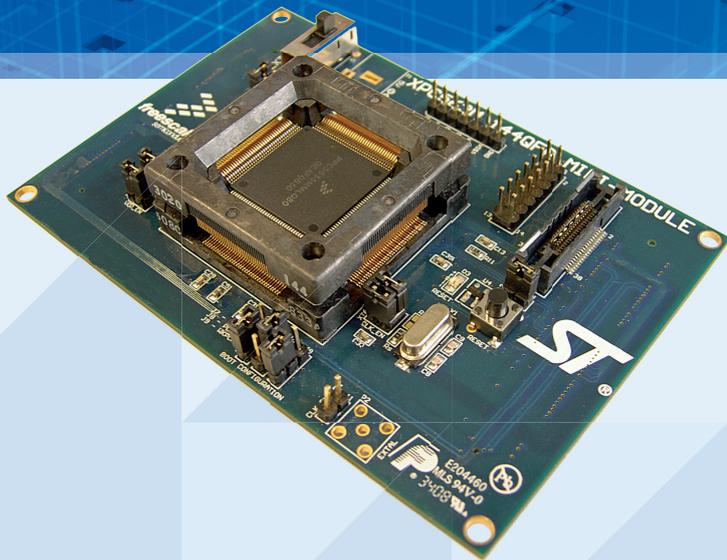


# Power Architecture Development Platform



The Development Platform consists of HighTec's

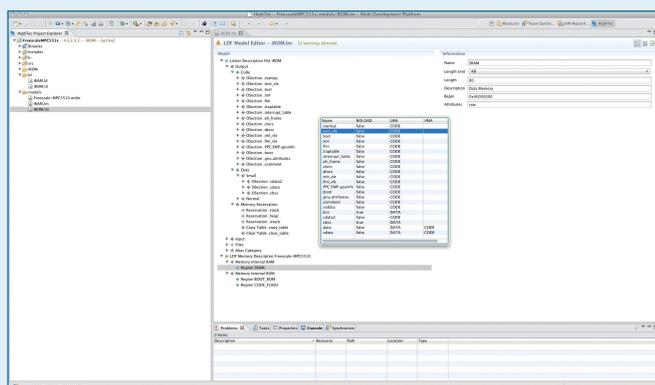
- **Power Architecture™ C/C++ multi-core compiler suite with leading optimization technology**
- **Eclipse™ integrated development environment**

## Features

The Development Platform includes powerful wizards and supports the latest Power Architecture™ families. It manages the project settings and entire build process for compiler, assembler and linker, and, moreover, the initialization settings of the hardware.

- **Project management**
- **Setup wizards**
- **Version control**
- **Model-driven graphical memory layout configuration**

The Eclipse™-based Power Architecture™ Development Platform allows simple definition of projects. After having selected a particular microcontroller derivative, the generation of a project with a correct startup code, the necessary hardware initialization, a valid and appropriate memory layout, and header files that describe the actual hardware configuration, e.g. special function registers, can be prompted virtually at the push of a button. The project contains a simple main function, which allows the implementation to be started immediately.



Memory layout can easily be adapted by means of the model-driven GUI in Eclipse™. This configuration will be verified against the derivative memory layout and will assist customers in placing the code and data in the appropriate regions of the memory.

## C/C++ compiler suite

- **Robust, compact and fast executing code**
- **AUTOSAR MCAL driver support**
- **ANSI/ISO conformance**
- **VLE and SPE support**
- **Multi-core support (ISO- and EABI-compliant)**
- **Long-term support**
- **Commercial standard and math libraries (no open source)**
- **Different addressing modes**
- **Code coverage support**

The Power Architecture™ GNU-based C/C++ compiler suite is one of the fastest build systems on the market. Furthermore, it can be started several times simultaneously for speeding up the build process by parallel compilation processes.

To take full advantage of the powerful 32-bit architecture, HighTec's compiler suite combines leading optimization technologies with flexibility and the required hardware control. The compiler includes various optimization strategies such as loop optimization, function inlining and global optimization strategies, resulting in excellent runtime behaviour and application codes of high code density. The compiler can generate both standard PowerPC instructions as well as instructions of variable length encoding (VLE) and thus achieve high code density. Generation of VLE instructions can be configured for individual modules or for the overall project.

The Power Architecture™ offers various addressing modes for optimizing code size and runtime behavior. These addressing modes are very efficient, because address calculation can be omitted and addresses are encoded in the assembler instructions at compile time. The HighTec compiler allows the use of different addressing modes via "pragma" statements in the source code to control the spreading and allocation of code and data in the appropriate memory regions.

In conjunction with AUTOSAR operating systems, some vendors need to reserve certain cpu registers for exclusive usage. For this purpose, the compiler provides an option for specifying so-called fixed registers, which the compiler is prevented from using. The HighTec C/C++ compiler is fully EABI-compliant and conforms to all relevant languages and ISO standards. The HighTec compiler is supported by leading AUTOSAR operating systems vendors such as ETAS, Elektrobit and Vector.

Automotive software requires portability to multiple platforms. The HighTec C/C++ compiler is able to generate position-independent code (PIC) and position independent data (PID). This functionality is useful for programming automotive ECUs during their production and testing process.

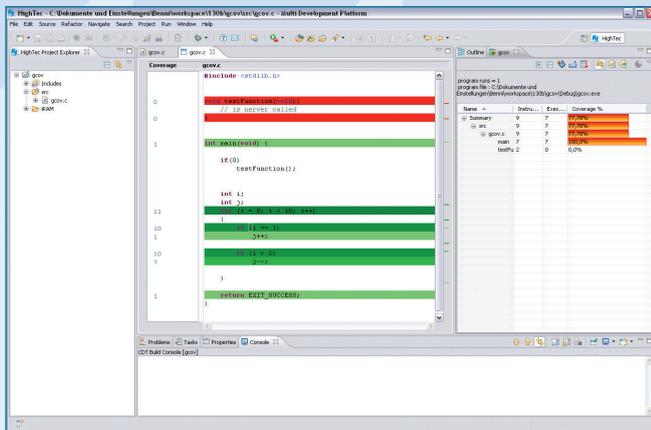
## ISO-26262

According to ISO-26262, application vendors in the automotive sector may be required to reproduce, after a random period of time, a software status generated by a particular tool version. HighTec provide a long term maintenance service for frozen tool versions, diversified, if at all, only by well-considered bug fixes, based on said versions.

For handling error detection and prevention measures, HighTec provide a sophisticated so-called bug-finding-compiler, which is able to consider known silicon bugs and even known compiler bugs. It is able to inspect the customer's application code and provide information on whether the application is affected by the bug, and if so, which of the modules are affected. This means, the user only needs to rebuild affected modules, thereby saving time and costs for building and testing the application.

Regarding confidence in use: The GCC-based compiler suite has been developed and used for over 20 years in general applications. In the automotive field, for instance, the HighTec GCC-based compiler has been established in leading Tier-1 and OEM projects for over ten years. Longterm and widespread worldwide use and development of the compiler suite in combination with well-established development and review processes guarantee high software quality. Moreover, commercial as well as open source test environments are used for validating the correctness of the toolchain.

For analyzing the code coverage of safety critical applications, the HighTec C/C++ compiler supports the generation of instrumented code. At run-time, collected code coverage information is written to standard gcov format. This information is available in plain text format, and the code coverage information of the corresponding source code line is visualized in the Eclipse editor.



Traceability of the software build process is relevant for the quality assurance of the software. The HighTec compiler suite supports traceability by recording certain information during the build process, such as the compiler version and build options. This information is stored in a dedicated section of the executable, which is not loaded into the target system and therefore does not occupy any additional memory. In addition, the HighTec tools are capable of assigning 'alias' names to variables and thus ensure compliance with customer's naming conventions.

## Multi-core support

HighTec's Power Architecture™ compiler suite supports homogeneous as well as heterogeneous multi-core architectures.

HighTec's solution integrates the multi core support at linker level, making source code adaption unnecessary. This ISO- and EABI-compliant multi-core support makes it easy to port existing single-core based source code to a multi-core system. The user can easily assign executable code or data to the appropriate core at linker level. Data exchange between different cores is handled by the hide-and-visibility concept, which specifies user-defined interfaces between the cores at linker level. Data and code have to be explicitly specified in order to be visible for any other cores than the one they belong to. Any data or code that is shared between cores, is listed in a separate output file and can be used to analyse the cross-reference between the cores. The hide-and-visibility feature implies a reduction of complexity in distributed systems, resulting in a reduced debugging effort and improved maintainability of the software system as a whole.

Apart from this, the execution of independent applications on each core is supported, since separation of code and data is assured, unless otherwise specified by the user.

The HighTec linker automatically handles the remapping of memories even for heterogeneous multi-cores. The linker is able to interlink object files for different core architectures and to generate one resulting output elf file.

The assignment of data and code to a core is stored as a flag in the CORE-ID property. These flags are checked by the linker to prevent that code or data, which are intended for usage in a single core, are assigned to a different core. The assigned CORE-ID of code and data is integrated in the symbol information and can easily be used for analysing purposes.

Efficient addressing modes, such as small addressing mode, can be used separately for each core. The corresponding address registers of each core must be initialized in the start-up code. The linker will automatically initialize the necessary base register for the small addressable areas of each core.

The compiler suite supports the definition of global or core local interrupt tables and trap tables.

## Example

A variable from one core can be accessed as an extern variable in another core, if it is explicitly exported beforehand by the first core.

This multi-core functionality is supported by leading debugger vendors such as Lauterbach, pls and isystem.

## Summary

HighTec's GNU-based Development Platform

- is easy to use in combination with the widespread and popular Eclipse™ IDE and a graphical memory management tool
- has short compilation time
- produces excellent code quality, as regards code size as well as code runtime behavior
- is able to handle multi-core architectures, even with heterogeneous cores, particularly for porting a heritage single core software to a multicore system
- provides easy portability between different hardware architectures

'Eclipse' and 'Built on Eclipse' are trademarks of Eclipse Foundation, Inc.

Distributor: **NeoMore** 23 rue des Poiriers F-78370 PLAISIR FRANCE +33 1 30 64 15 81 [www.neomore.com](http://www.neomore.com)



DEVELOPMENT PLATFORM

Compiler - Realtime operating system

<http://www.hightec-rt.com>