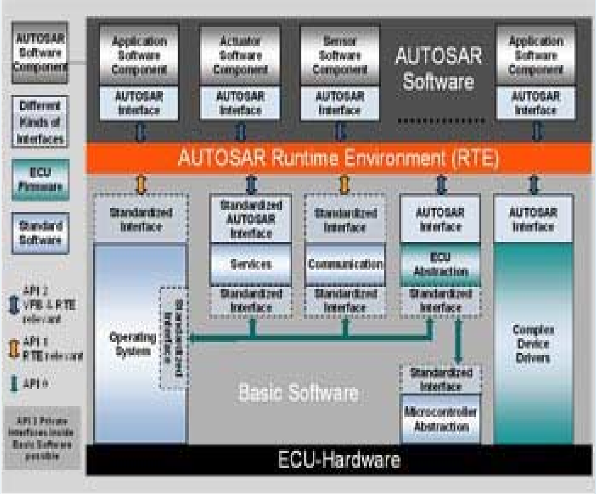
RTA-RTE2.1

AUTOSAR Release 2.1 RTE Generator



Features at a Glance

- Mature RTE Generation Tool
- AUTOSAR Release 2.1 compliant
- Easy integration into a variety of build environments
- Validation of Input Data (XML)
- Optimizes RTE resourse usage for either memory or CPU usage.
- Generates OS configuration to help with integration of RTE and OS
- Compiler and target independent RTE
- Easy debugging using VFB tracing

ETAS Group
Automotive LifeCycle
Solutions

Background

The AUTOSAR Runtime Environment (RTE) is the 'core' of the AUTOSAR software architecture, providing all communication mechanisms between the application Software Components (SWCs). The RTE presents an AUTOSAR interface to the SWCs, allowing them to be developed independently of the platform hardware and software on which they will eventually run. This allows AUTOSAR SWCs to be configured to run on multiple ECUs across a network or to be re-usable in multiple vehicle models, helping to reduce the overall cost of automotive software development.

ETAS has been involved for many years in the production of operating systems (OS) for the automotive industry. These OSs are currently used in over 250 million ECUs on the road worldwide. Members of the same engineering team that developed these successful OSs have been involved within the AUTOSAR RTE group since it was first formed, and they were active in RTE development even before then. Since 2006, the ETAS RTE generator has received positive feedback from those customers who have worked with the pre-release versions of the product. This considerable experience in automotive embedded software and RTE development has been used to develop a mature, robust production quality RTE Generator that builds on the results of the previous RTE development projects. ETAS's commitment to AUTOSAR continues as Premium members in AUTOSAR Phase II which runs from 2007 – 2009.

Introduction to RTA-RTE2.1

RTA-RTE2.1 is an RTE generator that is compliant with the AUTOSAR Release 2.1 RTE specification. It is a PC-based, command-line tool that can be easily integrated into a variety of build environments to generate an RTE that can be optimised for either memory or CPU usage. RTA-RTE2.1 helps users to solve many of the problems involved in the RTE Generation process by, for example, providing validation of the input XML, generating operating system (OS) configuration information and by providing easy debugging capabilities.

AUTOSAR divides the software development process into 2 main phases. In the first phase, the main SW architecture is developed independently of the hardware on which it will eventually run. This is achieved by developing the SWCs against a Virtual Function Bus that provides all of the communication mechanisms required by the SWCs. The second phase of the AUTOSAR development process is system integration, in which SWCs are mapped and integrated into the final production ECUs. RTA-RTE2.1 provides support for both of these phases of the AUTOSAR development process.

The Contract Phase

During the development of the SWCs, RTA-RTE2.1 allows the interface to the RTE to be generated before the allocation of SWCs to ECUs is known. Using an AUTOSAR SWC Description XML as input, RTA-RTE2.1 has sufficient information to generate the interface definition files necessary to allow engineers to start developing the SWCs. In this situation, the interface defines the contract between the RTE and the SWC; what that SWC must provide if future integration work is to happen easily. This is known as the Contract Phase.

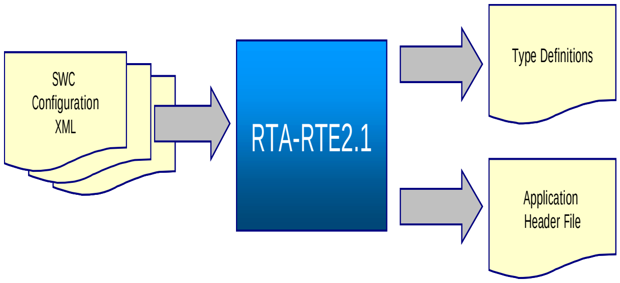


Figure1. RTA-RTE2.1 in the Contract Phase

Figure 1 shows that during the Contract Phase RTA-RTE2.1 uses the SWC configuration XML files to generate the Type Definition and Application Header files. These files contain the definition of the interface that the RTE will present to the SWC, including such information as declarations of API functions, SWC data structures and Runnable Entity prototypes for example. These header files allow the developer to start implementation of the SWC without having specific knowledge of the underlying basic software and hardware.

The RTE Generation Phase

Once the mapping of all SWCs to ECUs is known, and the ECU has been fully configured, the RTE itself can then be generated. At this stage, a much more complete set of RTE configuration information is available, including how many instances of a SWC exist, where runnables are executing, which communication is local to an ECU and which must be routed across the network etc. This information is contained in the 'ECU Configuration Description' XML files defined in the AUTOSAR methodology. These XML files are used as input to RTA-RTE2.1 to re-generate the interface definition files to include optimisations based on this additional context. This is known as RTE Generation Phase.

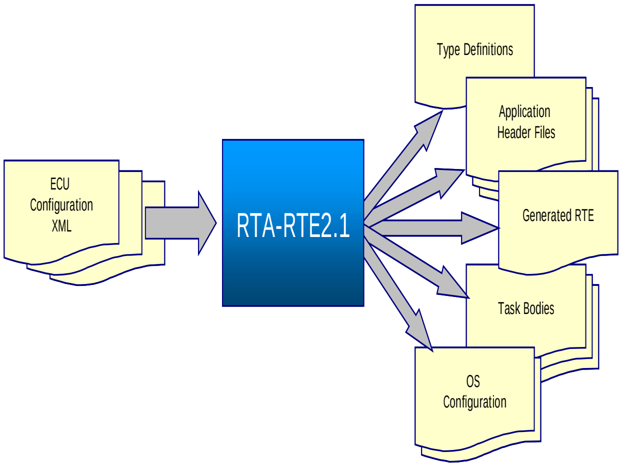


Figure2. RTA-RTE2.1 in the RTE Generation Phase

Figure 2 demonstrates the operation of RTA-RTE2.1 in the RTE Generation Phase. During this phase, in addition to producing optimized SWC header files, RTA-RTE2.1 also generates the actual RTE code for the specified ECU, along with build information indicating the files that need to be compiled and any dependencies. The generated RTE is output in C, and is fully MISRA C compliant. RTA-RTE2.1 also creates task bodies based on the mapping of the runnable entities in each SWC to OS tasks specified in the input ECU Configuration Description XML files. The final set of output during the RTE Generation phase is the OS configuration information that is required by the RTE.

RTE Optimisation

An embedded system is a resource constrained environment where typically the resource limit is either memory space or available CPU cycles. Since it is not possible to pre-determine which constraint is most pressing for all systems, it is desirable to be able to choose the optimisation strategy based on the resources available for a specific ECU. An important feature of RTA-RTE2.1 is that it will optimize the resource usage of the generated RTE whenever possible, and also gives the user the flexibility to specify whether the optimization should be performed for either code size (memory) or for run-time (CPU usage). In either case RAM is usually the most scarce resource and therefore solutions that favour ROM usage over RAM are generally preferred by RTA-RTE2.1.

The implementation of the RTE API presented to SWCs by the generated RTE is optimised by RTA-RTE2.1 specifically for each ECU based on the mapping of SWCs to ECUs and the abstract communication matrix. The optimizations are applied when generating in the RTE phase, and provide for highly efficient API access. Some examples of possible optimizations include direct read/write for sender-reciver communication, direct function calls for client-server communication and direct access to per-instance memories. When applied, these optimizations are implemented purely as macros to eliminate all function call overheads for the relevant RTE APIs. Many of these optimisations are additional to the AUTOSAR standard, allowing users of RTE-RTE2.1 to generate an RTE that provides market leading resource efficiency.

Integration into existing build environments

An important requirement for any RTE generation tool is that it must integrate easily into existing toolchains. RTA-RTE2.1 is controlled by a combination of command-line options and an INI file. The INI file configures which DLLs are available for use by the application. The common feature of these control mechanisms is that they both allow for a degree of automation, whether using a DOS batch file, a makefile or some other scripting language. In this way, RTA-RTE2.1 facilitates easy integration into a variety of existing build environments.

Cooperation with OS

Figure 3 shows the AUTOSAR software architecture, and illustrates the relationship between the RTE and the underlying basic software, including the operating system (OS). The AUTOSAR methodology includes tools for the generation of the RTE and for the configuration of the OS. Within RTA-RTE2.1, aspects of both of these process steps are integrated into a single tool. RTA-RTE2.1 uses OS plugins to create OS configurations that greatly ease the integration of the RTE into a system by eliminating the significant task of keeping the OS and RTE configurations consistent.

The OS configuration fragment is created in the OSEK Implementation Language (OIL), and defines all OS objects needed to implement the generated RTE. This includes tasks, alarms, counters and, where appropriate, schedule tables.

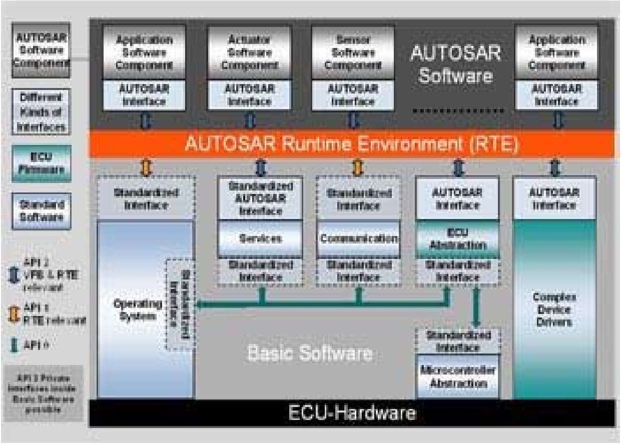


Figure 3. The AUTOSAR software architecture.

The product is provided with multiple OS plugins that create the OS configuration output for various OS types. These include support for OSEK 2.2.3 (OIL), AUTOSAR SC1 support (OIL, with RTA-OSEK extensions) and ERCOSEK v4.3. The creation of the OS configuration information is not a formal requirement on the RTE in the AUTOSAR R2.1 specification. Inclusion of this functionality in RTA-RTE2.1 therefore provides users with a significant enhancement to the standard AUTOSAR functionality in this important area. RTA-RTE2.1 has been developed and tested to work seamlessly with RTA-OSEK v5.0 (OSEK 2.2.3 compliant) and is therefore the ideal choice for existing RTA-OSEK users.

The OS configuration files generated by RTA-RTE2.1 contain partial configuration information that is relevant only to the RTE. These configuration files must therefore be combined with other, non-RTA-RTE2.1 generated, configuration files before being passed to your OS configuration tools (not provided with RTA-RTE2.1). The scope of the configuration data for the OS generated by RTA-RTE2.1 are only those data necessary for the RTE to work. If additional OS configuration information is provided as input, then this will be ignored by RTA-RTE2.1.

Input XML Validation

Another important function that is performed by RTA-RTE2.1 is the rigorous checking of the input XML configuration data for correctness and completeness against the AUTOSAR Release 2.1 specification. A large number of checks are performed to ensure that the input data are consistent (e.g. that references within the XML point to a valid object of the right type) and correct. If a problem is detected in the input data RTA-RTE2.1 will raise an error message to inform the user of the nature of the problem and of the action that has been taken.

Source Code or Object Code SWCs

The generated RTE supports SWCs written in C or C++ that are provided as either source code or as object code. The application header files generated in Contract Phase contain sufficient detail to allow a SWC to be developed, compiled and shipped as object files to the ECU integrator, allowing the SWC developer to protect their valuable Intellectual Property.

Target and Compiler Independence

RTA-RTE2.1 makes use of the AUTOSAR compiler and memory mapping abstractions to ensure that the generated RTE is compiler and target independent, allowing its use with a wide range of ECU platforms.

Easy Debugging

RTA-RTE2.1 facilitates easy debugging by allowing the user to perform Virtual Function Bus (VFB) tracing. This optional capability allows for the monitoring of AUTOSAR signals as they are sent and received across the VFB by the use of a series of configurable hook functions.

Product Roadmap

RTA-RTE2.1 is the second in a roadmap of production quality RTE generator releases from ETAS, following on from the release of RTA-RTE2.0 in 2007. The numerals in the product name of each RTA-RTE release indicate compliance with a particular release of the AUTOSAR specification. Hence RTA-RTE2.0 is compliant with the AUTOSAR Release 2.0 specification and similarly, RTA-RTE2.1 is compliant with the AUTOSAR Release 2.1 specification.

ETAS will continue to support our customers' evolving requirements by releasing a series of RTA-RTE products to follow on from RTA-RTE2.0 and RTA-RTE2.1. In 2008, this process will continue with the release of RTA-RTE3.0 that will be compliant with AUTOSAR Release 3.0. ETAS will fulfill customer-specific requirements by offering an engineering service to develop new functionality that can be added to the core RTA-RTE products.

RTA-RTE2.1 Summary

- Mature RTE generation tool
- AUTOSAR Release 2.1 compliant
- Easy integration into a variety of build environments
- Validation of Input Data (XML) for correctness and consistency
- Support for both Contract Phase and RTE Generation Phase
- Optimizes RTE resourse usage for either memory or speed
- Output of OS configuration eases integration of the RTE and OS
- Generated RTE is fully MISRA C compliant
- Compiler and target independent RTE can be used with a wide range of ECU platforms
- Supports SWCs written in C or C++ provided as either source code or object code
- Easy debugging using Virtual Function Bus (VFB) tracing
- Generated RTE is compatible with OSEK 2.2.3, AUTOSAR SC1 and ERCOSEK v4.3 Operating Systems
- Seamless cooperation with RTA-OSEKv5.0

Technical Data

RTA-RTE2.1

Host System Requirements	
Hardware	1 GHz Pentium PC, 512 MB RAM
Operating System	Windows 2000, Windows XP
Memory	> 50Mb free space

Ordering Information

RTA-RTE2.1

Order Name	Short Name	Order Number
Machine-Named license for RTA-RTE2.1	LD_RTA-RTE2.1_LIC-MP	F-00K-106-050
User-Named license for RTA-RTE2.1	LD_RTA-RTE2.1_LIC-UP	F-00K-106-051
Concurrent license for RTA-RTE2.1	LD_RTA-RTE2.1_LIC-CP	F-00K-106-052
Media for RTA-RTE2.1	LD_RTA-RTE2.1_PROD	F-00K-106-053
Service Contract for a Machine-Named license for RTA-RTE2.1	LD_RTA-RTE2.1_SRV-ME52	F-00K-106-062
Service Contract for a User-Named license for RTA-RTE2.1	LD_RTA-RTE2.1_SRV-UE52	F-00K-106-063
Service Contract for a Concurrent license for RTA-RTE2.1	LD_RTA-RTE2.1_SRV-CE52	F-00K-106-064

RTA-RTE2.1 Production License

To commercially deploy an RTE that has been generated using RTA-RTE2.1 in ECU applications, a valid production license is required. Please contact your local ETAS sales office for details of production licenses for RTA-RTE2.1.

Contact addresses

ETAS GmbH
70469 Stuttgart, Germany
Phone +49 711 89661-0
Fax +49 711 89661-106
sales@etas.de

ETAS S.A.S.
94588 Rungis Cedex, France
Phone +33 1 56 70 00 50
Fax +33 1 56 70 00 51
sales@etas.fr

ETAS Ltd.
Burton-upon-Trent
Staffordshire DE14 2WQ
Great Britain
Phone +44 1283 54 65 12
Fax +44 1283 54 87 67
sales@etas-uk.net

ETAS Inc.
Ann Arbor, MI 48103, USA
Phone +1 888 ETAS INC
Fax +1 734 997-9449
sales@etas.us

ETAS K.K.
Yokohama 220-6217, Japan
Phone +81 45 222-0900
Fax +81 45 222-0956
sales@etas.co.jp

ETAS Korea Co., Ltd.
Seoul 137-889, Korea
Phone +82 2 57 47-016
Fax +82 2 57 47-120
sales@etas.co.kr

ETAS (Shanghai) Co., Ltd.
Shanghai 200120, P.R. China
Phone +86 21 5037 2220
Fax +86 21 5037 2221
sales.cn@etasgroup.com

ETAS Automotive India Pvt. Ltd.
Bangalore 560 068, India
Phone +91 80 4191 2588
Fax +91 80 4191 2586
sales.in@etasgroup.com

www.etasgroup.com

Subject to change (07/2007)