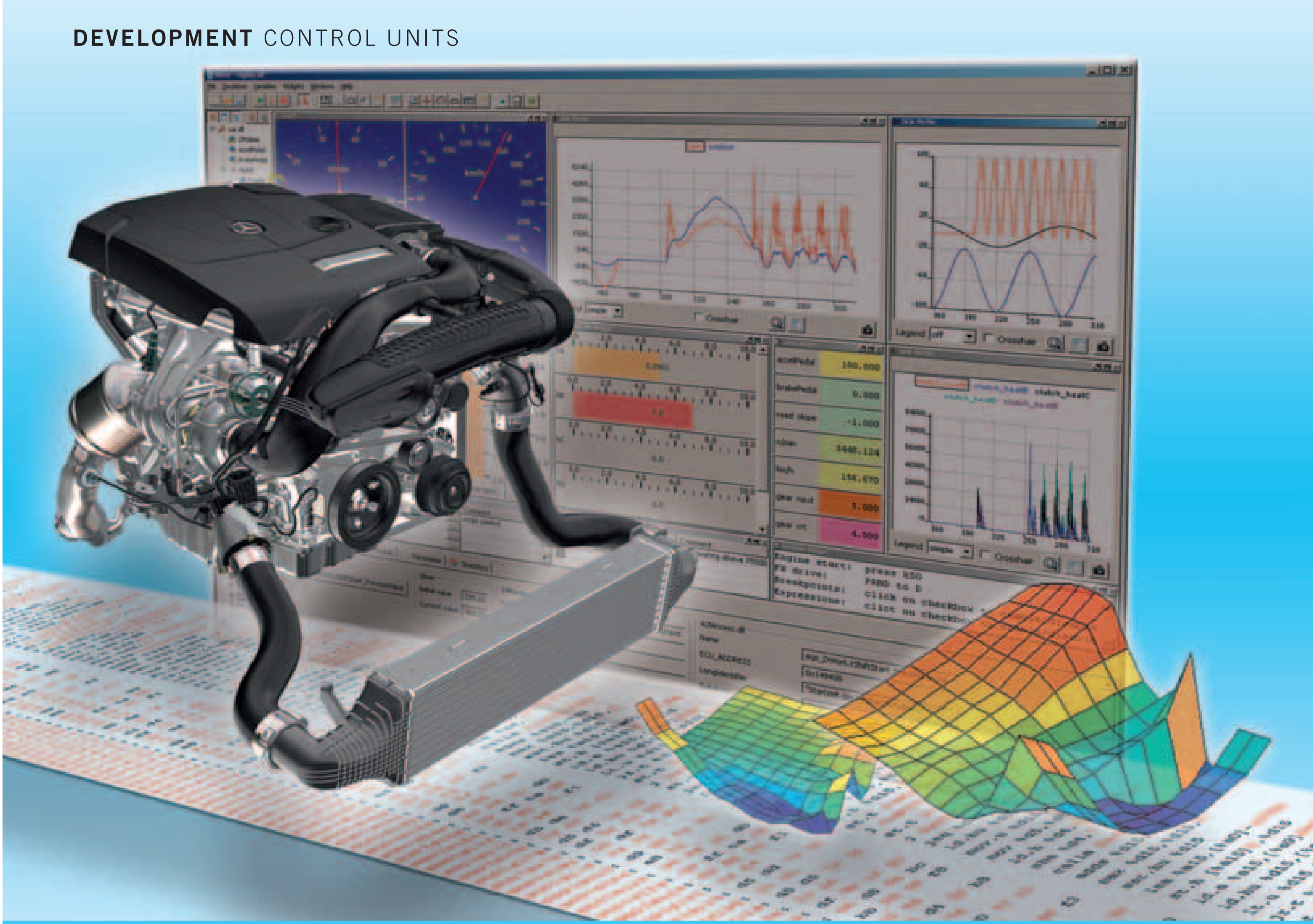


CHIP SIMULATION OF AUTOMOTIVE ECUS ON PC

Modern ECUs contain tens of thousands of engine parameters that need to be tuned. Calibration of these parameters is time consuming and complex. Simulation and optimization on a PC could help to automate and speed up the calibration process, in particular if simulation runs much faster (e.g. 20 times) than real-time. QTronic and Daimler describe how the ECU Simulation is integrated in an engine development process by the OEM.

AUTHORS

DR. JAKOB MAUSS
is Managing Director of QTronic GmbH in Berlin (Germany).

MATTHIAS SIMONS
is an Engineer at Daimler AG in Stuttgart (Germany).

AVOID MISTAKES

Engine calibration is typically performed by an OEM, while the ECU code is owned by the supplier of the ECUTherefore, the OEM is typically unable to set up a ECU simulation based on the original C code of the ECU. Instead, to set up a simulation, time consuming and error prone reverse engineering is needed to develop an "equivalent model" of the ECU function of interest. To improve this situation, we have integrated a chip simulator into the virtual ECU tool Silver. This is used to simulate hex files compiled for TriCore targets directly on PC. Simulation requires:

- a hex file that contains programme code and parameters of the simulated functions
- start addresses of the functions to be simulated
- an ASAP2/a2l file that defines the conversion rules for the involved inputs, outputs, and characteristics, as well as corresponding addresses; the start addresses of functions can e.g. be extracted from a map file generated together with the hex file.

Silver uses the a2l file to automatically convert scaled integer values to physical values and vice versa during simulation. A TriCore simulation can also be exported as SFunction (mexw32 file) for use in Matlab/Simulink. On a standard PC, hex simulation runs with about 40 MIPS. If only simulating selected functions of an ECU, this is fast enough to run a simulation much faster than real-time. In this article, we also report how such simulations are used today to support the development of gasoline engines at Daimler.

VIRTUAL ECUS IN THE DEVELOPMENT PROCESS

Simulation has great potential to improve the development process for ECUs. Simulation helps to move development tasks to PC, where they often can be performed faster, cheaper or better in some respect. Examples that illustrate this point:

- On a PC, an engineer can easily "freeze time", i. e. stop simulation and inspect the call stack and all variables of a virtual (i.e. simulated) ECU without band width limitation and repeat a simulation deterministically as often as needed. In contrast, real ECU as used in HiL settings or test rigs must run in real time. Stopping and stepping is impossible or requires considerable extra effort, e.g. based on the JTAG debug interface. Exact reproduction of experiments is difficult or impossible on a HiL or test rig, due to non-deterministic effects.
- On a PC, a calibration tool like INCA (ETAS) or CANape (Vector) can be connected to a virtual ECU via XCP to measure into a running simulation and to tune characteristics online. This way, many parameters of a ECU can be tuned using a relatively cheap and highly available PC platform, without blocking rare and more expensive resources like real prototypes and test rigs.
- A virtual ECU might run on PC 20 times faster than real time. When used in combination with test automation, a simple PC gives then 20 time higher test throughput than a much more expensive HiL test system.
- On a PC, a development engineer can rebuild the entire ECU within five minutes after modification of a module, thanks to incremental build, and test drive the result in a simulated environment. This helps to detect problems early on the developer's PC, and decreases the number of problems that show up late, when integrating all modules. As experience shows, such early checks speed up development.

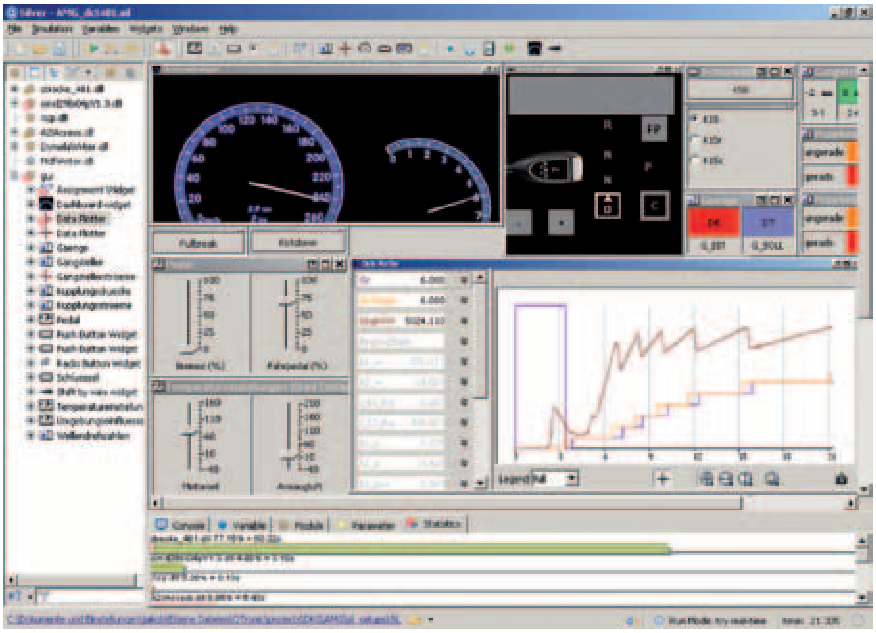
To exploit these and other benefits when developing an ECU, the ECU must first be ported to PC. This is typically done based on the C code of the ECU, which is either hand coded, or generated by tools such as Ascet (ETAS), TargetLink (dSpace) or Embedded Coder (MathWorks). For example, QTronic's virtual ECU tool Silver [1] provides a framework to:

- compile given ECU tasks for Windows PC
- emulate the underlying RTOS and other services (CAN, XCP)
- run the resulting virtual ECU closed-loop with a simulated vehicle.

Typical applications are [2, 5], where a virtual ECU is used to develop the controller for an automatic transmission. For closed-loop simulation ❶, vehicle models can be imported from many simulation tools into Silver, including Matlab/Simulink, Dymola, SimulationX and MapleSim, e.g. through the FMI format for model exchange [4]. However, sometimes C code is not available for implementing a virtual ECU. There are two main sources for such a situation:

: Protection of intellectual property – all or major parts of the ECU have been developed by a supplier and the OEM interested in building a virtual ECU (e.g. to support calibration, a task typically performed by an OEM) has therefore no access to the C code.
: Target-specific C code – C code is available but the C code uses inline assembler or other target or compiler specific constructs, which prevents compilation for other targets, such as the Windows x86 platform.

To deal with such situations, we have recently integrated a chip simulator into the virtual ECU tool Silver. This way, a virtual ECU can be build based on a hex file compiled for the target processor of the ECU. No access to C code is needed in this case. Instead of compiling C code for the Windows x86 platform, the chip simulator takes the binary compiled for the target processor and simulates the execution of the instructions by the target processor on Windows PC.



❶ Transmission control unit running in Silver (taken from [5])

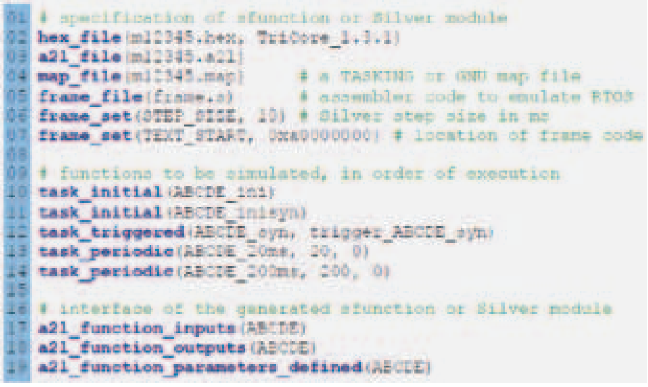
CHIP SIMULATION FOR TRICORE TARGETS

Many automotive controllers are based on processors of Infineon's TriCore family, in particular in the powertrain domain. Examples are engine controllers of the MED and EDC family by Bosch and transmission controllers by Continental.

Silver 2.5 uses a specification file that lists the tasks of a hex file to simulate. Silver automatically turns such a spec file into an executable Silver module (dll) or SFunction. ❷ shows a typical spec file. The spec file first lists the required file s ② (line 2 to 5). The map file is optional. If given, the spec file may use symbolic names for functions (such as ABCDE_20ms), instead of addresses (such as 0x80081cde). File frame.s (line5) contains startup code to set up stacks, registers, timer and other resources. ② (lines 10 to 14) lists the functions to run, and specifies when and in which order to run these functions. Silver uses this to generate the application-specific part of the RTOS emulation. For event triggered tasks, Silver offers two alternative event models, ② (line 12). Finally, ② (lines 17 to 19) define the inputs, outputs and parameters of the generated module or SFunction. In this case, we just reuse the interface of a "Function element" of the a2l file, for a function called ABCDE. It is also possible, to list individual variables here by name, as long as their properties (such as address, conversion rule, data type) are described in the a2l file. In addition, the spec file offers means to specify:

: properties of the XCP emulation, if any, to support online calibration and measurement using tools such as INCA and CANape
: data sections to be included into the generated Silver module or SFunction. This way, initial loading of the hex file into simulated memory can be avoided, to speed up simulation
: memory areas to be copied to other (faster) memory by the start-up code
: functions to be replaced by other functions. This way, a getter or setter function to access a sensor or actuator can be replaced by a function that directly accesses a plant model or measured values instead, bypassing low-level device dependent code
: logging options, e.g. to track memory access during simulation.

The Silver module or SFunction generated this way performs exactly the same computations on PC, as on the real target,



```
01 # specification of sfunction or Silver module
02 hex_file(m12345.hex, TriCore_1.3.1)
03 a2l_file(m12345.a2l)
04 map_file(m12345.map)      # a TASKING or GNU map file
05 frame_file(frame.s)       # assembler code to emulate RTOS
06 frame_set(STEP_SIZE, 10) # Silver step size in ms
07 frame_set(TEXT_START, 0xa0000000) # location of frame code
08
09 # functions to be simulated, in order of execution
10 task_initial(ABCDE_ini)
11 task_initial(ABCDE_inisyn)
12 task_triggered(ABCDE_syn, trigger_ABCDE_syn)
13 task_periodic(ABCDE_20ms, 20, 0)
14 task_periodic(ABCDE_200ms, 200, 0)
15
16 # interface of the generated sfunction or Silver module
17 a2l_function_inputs(ABCDE)
18 a2l_function_outputs(ABCDE)
19 a2l_function_parameters_defined(ABCDE)
```

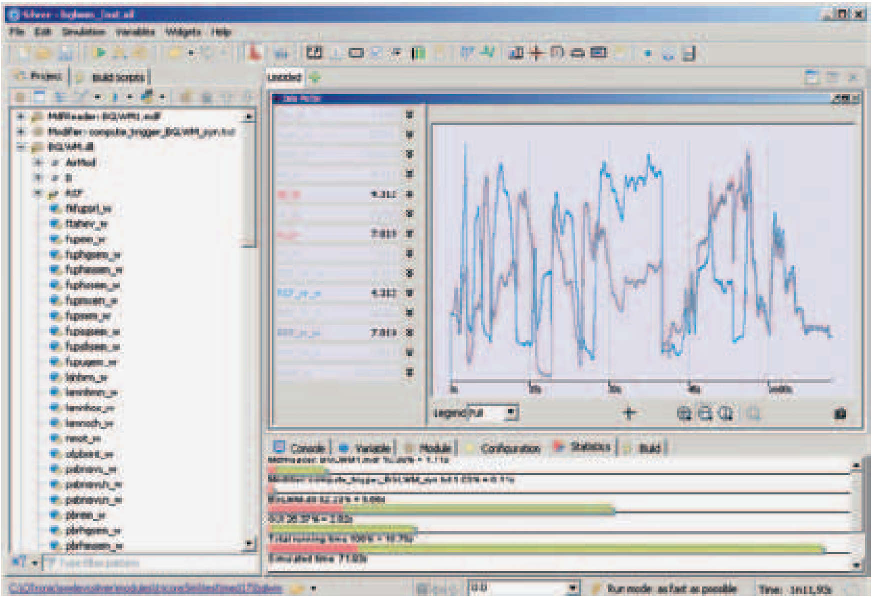
❷ Specification file for chip simulation since the effect of every machine instruction on memory and chip registers is exactly simulated on PC. However:

- Simulation is just instruction accurate, not cycle accurate. This means, the simulation on PC cannot be used to exactly predict execution time on the real target. For example, pipeline effects of different access times to memory (e.g. fast on-chip RAM versus external RAM) are not modelled.
- Conceptually, simulated tasks execute infinitely fast: The emulated RTOS never interrupts a task and related effects cannot be analysed using the generated model.
- Silicon bugs are not simulated. If a compiler for the real target does not work around a silicon bug correctly, this is likely to be invisible in the simulation: simulated and target behaviour might differ in such cases.
- On chip devices such as the PCP co-processor or CAN controllers are not simulated.

The spec file used to port selected parts of a hex file to PC might contain bugs. To locate these, Silver integrates a debugger based on the instruction set simulator tsim, developed by Infineon.

In order to measure the execution speed of chip simulation, we have ported a complex ECU function (MED17 with TC1797) implemented by 5 different C functions that run initially, every 10 and 200 ms, and synchronous to the crankshaft. The spec file is very similar to the one shown in ②. The function has 114 scalar inputs, 102 scalar outputs and 108 parameters (characteristics), many of them axes and maps. We have then measured all inputs and outputs of the function on an engine test rig for a scenario of 3.5 min and used the resulting measurement (mdf/dat) file to drive simulations in Silver, using either tsim or a generated Silver module. Each simulation executed 380.205256 million instructions (counted by tsim) and has been repeated 5 times on a Windows PC with Intel i5 processor at 2.4 GHz and 2.92 GB RAM. Average execution times found this way are shown in ❹.

Silver can also turn a spec file into a SFunction, i.e. a mexw32 file that runs in Simulink. This is particularly interesting when using chip simulation to support automated optimisation of parameters, because many optimisation tools are implemented on top of Matlab/Simulink. The generated SFunction accepts all characteristics listed in the spec file as SFunction parameters. This makes it easy to connect the generated SFunction with an optimisation procedure. For example, the SFunction can be called with workspace variables that are then automatically varied by the optimisation procedure between SFunction calls. The performance of a generated SFunction is again about 40 MIPS.



❸ The BGLWM function running in Silver, driven by measurement file

APPLICATIONS OF CHIP SIMULATION

During development of an engine controller, a developer might want to replace a certain function of the ECU by its own version of that function, bypassing the original function. For real ECUs, this can be done with tools such as EHooks (ETAS) or No-Hooks (ATI). These tools manipulate the original hex file, such that the bypassed function is not executed any more, but just calls the new function instead. The placing of bypass hooks by direct manipulation of the hex file is a mighty but error-prone tool. Sometimes a hooked function is not called at all or only some variables are overwritten and some not. Normally, such errors are only detected after the manipulated hex-file was flashed on the ECU and then run on the test bench or in a car. With the possibility of instruction accurate simulation of the patched hex file, we can detect these errors much faster and without any risk to car or engine.

A second application of chip simulation is the numerical optimisation of engine parameters [3]. We have combined chip simulation as described above with a procedure for numerical optimisation to compute optimal values for certain engine parameters.

These computations require an accurate and fast model of the engine function of interest. In the past, we have used hand-coded models of ECU functions, developed with Matlab/Simulink. This has been time consuming and error prone.

TARGET	EXECUTION TIME	MIPS
SILVER IN DEBUG MODE	919.15 sec	0.41
GENERATED SILVER MODULE OR MATLAB/SIMULINK SFUNCTION	9.30 sec	40.80
MED17 WITH TC1797, 180 MHZ	210.00 sec	270

❹ Performance of chip simulation for the BGLWM example

We have now partially replaced these hand-coded models with SFunctions generated automatically by Silver from a given hex file. The generated SFunctions proofed to run as fast as their hand coded counterparts.

CONCLUSIONS

As demonstrated above, an ECU hex file compiled for some target processor can be executed by the virtual ECU tool Silver on Windows PC, either open-loop driven by measurements or in closed-loop with a vehicle model. Depending on the application, selected ECU functions are simulated, or nearly the entire ECU.

This kind of simulation opens new possibilities to move development tasks from road, test rig or HiL to PCs, where they can be processed faster, cheaper or better in some respect, without requiring access to the underlying C code. Daimler currently uses this innovative simulation approach to support controls development for gasoline engines. Other applications, such as online calibration on PC via XCP seem to be doable as well.

REFERENCES

[1] Junghanns, A.; Serway, R.; Liebezeit, T.; Bonin, M: Building Virtual ECUs Quickly and Economically, ATZelektronik 03/2012, June 2012

[2] Brückmann, H.; Strenkert, J; Keller, U; Wiesner, B.; Junghanns A.: Model-based Development of a Dual-Clutch Transmission using Rapid Prototyping and SiL. International VDI Congress Transmissions in Vehicles 2009, Friedrichshafen, Germany, 30.06.-01-07.2009. http://qtronic.de/doc/DCT_2009.pdf

[3] Röpke , R.(ed.): Design of Experiments (DoE) in Engine Development – Innovative Development Methods for Vehicle Engines. Expert Verlag, 2011

[4] Blochwitz, T.; Otter M. et. al.: Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. 9th International Modelica Conference, Munich, 2012

[5] Tatar, M.; Schaich, R.; Breitinger, T.: Automated test of the AMG Speedshift DCT control software. 9th International CTI Symposium Innovative Automotive Trans-missions, Berlin, 2010. http://qtronic.de/doc/TestWeaver_CTI_2010_paper.pdf

IMPRINT

ATZ elektronik

WORLDWIDE

www.ATZonline.com

06 I 2012 _ December 2012 _ Volume 7
Springer Vieweg I Springer Fachmedien Wiesbaden GmbH
P. O. Box 15 46 · 65173 Wiesbaden · Germany I Abraham-Lincoln-Straße 46 · 65189 Wiesbaden · Germany
Amtsgericht Wiesbaden, HRB 9754, USt-IdNr. DE811148419

Managing Directors Dr. Ralf Birkelbach (Chairman), Armin Gross, Albrecht Schirmacher I **Director Corporate Publishing, Corporate Content & Ad Sales** Armin Gross
Director Marketing & Direct Sales Rolf-Günther Hobbeling I **Director Production** Christian Staral

EDITOR IN CHARGE
Dr. Johannes Liebl, Wolfgang Siebenpfeiffer

EDITOR IN CHIEF
Dr. Alexander Heintzel
phone +49 611 7878-342 · fax +49 611 7878-462
alexander.heintzel@springer.com

VICE-EDITOR IN CHIEF
Markus Schöttle
phone +49 611 7878-257 · fax +49 611 7878-462
markus.schoettle@springer.com

MANAGING EDITOR
Kirsten Beckmann M. A.
phone +49 611 7878-343 · fax +49 611 7878-462
kirsten.beckmann@springer.com

EDITORIAL STAFF
Dipl.-Ing. (FH) Richard Backhaus
phone +49 611 5045-982 · fax +49 611 5045-983
richard.backhaus@rb-communications.de
Dipl.-Ing. (FH) Andreas Fuchs
phone +49 6146 837-056 · fax +49 6146 837-058
fuchs@fachjournalist-fuchs.de
Dipl.-Ing. Ulrich Knorra
phone +49 611 7878-314 · fax +49 611 7878-462
ulrich.knorra@springer.com
Dipl.-Ing. Michael Reichenbach
phone +49 611 7878-341 · fax +49 611 7878-462
michael.reichenbach@springer.com
Roland Schedel
phone +49 6128 853-758 · fax +49 6128 853-759
rschedel@text-com.de
Stefan Schlott
phone +49 8726 9675972
redaktion_schlott@gmx.net
Martina Schraad
phone +49 611 7878-276 · fax +49 611 7878-462
martina.schraad@springer.com

PERMANENT CONTRIBUTORS
Andreas Burkert, Prof. Dr.-Ing. Stefan Breuer,
Hartmut Hammer, Prof. Dr.-Ing. Fred Schäfer

ENGLISH LANGUAGE CONSULTANT
Paul Willin

ONLINE I ELECTRONIC MEDIA
Portal Manager Automotive
Christiane Brünglinghaus
phone +49 611 7878-136 · fax +49 611 7878-462
christiane.bruenglinghaus@springer.com
Editorial Staff
Katrin Pudenz M. A.
phone +49 6172 301-288 · fax +49 6172 301-299
redaktion@kpz-publishing.com

SPECIAL PROJECTS
Managing Editorial Journalist
Markus Bereszewski (mb)
phone +49 611 7878-122 · fax +49 611 7878-462
markus.bereszewski@springer.com

ASSISTANCE
Christiane Imhof
phone +49 611 7878-154 · fax +49 611 7878-462
christiane.imhof@springer.com

Marlena Strugala
phone +49 611 7878-180 · fax +49 611 7878-462
marlena.strugala@springer.com

ADDRESS
Abraham-Lincoln-Straße 46 · 65189 Wiesbaden
P. O. Box 1546 · 65173 Wiesbaden, Germany
redaktion@ATZonline.de

ADVERTISING

HEAD OF KEY ACCOUNT MANAGEMENT
Tanja Pfisterer

HEAD OF SALES MANAGEMENT
Britta Dolch

SALES MANAGEMENT
Volker Hesedenz
phone +49 611 7878-269 · fax +49 611 7878-78269
volker.hesedenz@best-ad-media.de

MEDIA SALES
Ahmadou Ndiaye
phone +49 611 7878-260 · fax +49 611 7878-78260
ahmadou.ndiaye@best-ad-media.de

KEY ACCOUNT MANAGEMENT
Rouwen Bastian
phone +49 611 7878-399 · fax +49 611 7878-78399
rouwen.bastian@best-ad-media.de

DISPLAY AD MANAGER
Susanne Bretschneider
phone +49 611 7878-153 · fax +49 611 7878-78153
susanne.bretschneider@best-ad-media.de

AD PRICES
Advertising ratecard from October 2012

MARKETING I OFFPRINTS

PRODUCT MANAGEMENT AUTOMOTIVE MEDIA
Sabrina Brokopp
phone +49 611 7878-192 · fax +49 611 7878-407
sabrina.brokopp@springer.com

OFFPRINTS
Martin Leopold
phone +49 2642 907-596 · fax +49 2642 907-597
leopold@medien-kontor.de

PRODUCTION I LAYOUT
Iris Conradi
phone +49 611 7878-263 · fax +49 611 7878-78263
iris.conradi@springer.com

SUBSCRIPTIONS
Springer Customer Service Center GmbH
Haberstraße 7 · 69126 Heidelberg · Germany
phone +49 6221 3454-303 · fax +49 6221 3454-229
Monday to Friday, 8 a.m. to 6 p.m.
springervieweg-service@springer.com

SUBSCRIPTION CONDITIONS

The eMagazine appears six times a year at an annual subscription rate 99 € for private persons and 149 € for companies. Special rate for students on proof of status in the form of current registration certificate 59 €. Special rate for VDI/ÖVK members on proof of status in the form of current member certificate 79 €. Special rate for studying VDI members on proof of status in the form of current registration and member certificate 45 €. Annual subscription rate for combination ATZelektronik worldwide (eMagazine) and ATZelektronik (print) 198 €. All prices include VAT at 7 %. Every subscription comes with access to the ATZonline archive. However, access is only available for the individual subscription holder. The subscription can be cancelled in written form at any time with effect from the next available issue.

YOUR HOTLINE TO ATZelektronik

Editorial Staff
☎ +49 611 7878-257
Customer Service
☎ +49 6221 3454-303
Advertising
☎ +49 611 7878-260

HINTS FOR AUTHORS

All manuscripts should be sent directly to the editorial journalists. By submitting photographs and drawings the sender releases the publishers from claims by third parties. Only works not yet published in Germany or abroad can generally be accepted for publication. The manuscripts must not be offered for publication to other journals simultaneously. In accepting the manuscript the publisher acquires the right to produce royalty-free offprints. The journal and all articles and figures are protected by copyright. Any utilisation beyond the strict limits of the copyright law without permission of the publisher is illegal. This applies particularly to duplications, translations, microfilming and storage and processing in electronic systems.