

# High-level Tool Targeted for AVR Controllers

By Henrik Leerberg, IAR Systems

*Developing embedded applications often require significant and specific knowledge from the involved engineers. Typically, issues like real-time properties, timing bottlenecks, memory constraints, realization of hardware device drivers etc. all demands previous experience within embedded applications development in order to successfully finalize the application or project. The same actually applies for tools. The times where every embedded project could deploy any general tool on the market have passed. Today, tools for embedded projects need to be specific and customized for the problem domain and hardware devices selected. RTOS vendors have already adapted towards this trend, and now also the graphical design and development tools that automate several processes in building and testing embedded software are becoming target-specific.*

## The tool vendor's challenge

There are many good reasons for using tools and other 3rd party solutions when building a new embedded device. The fact that code generation technology has matured significantly during the recent years as well as emerging technologies for automatically being able to test the embedded applications, have all added to the number of reasons why corporations should seriously consider (re)investing in tools for new projects.

General purpose tools that cover everything from different analysis and design approaches to several flavors of code generation and languages, (here we can call them "case-tools"), as complex to learn, manage and use as a nuclear power plant, and will not meet the future demands of the embedded developers. The embedded developer requires tools developed for him and for the problems he is facing.

The embedded developer has up until now always been the last in line to get new tools to help building products. So far, many tool vendors have focused their energy on tools for mainframe-, PC-, administrative-type application development, and this is the world from where many of the existing tool vendors originate. However, some of them have been trying, and still try, to re-engineer their tools into serving the embedded developer's needs, but this "conversion" will be tough for many – if not impossible. These vendors' starting-point is already off-track since their tools once were built for a completely other purpose and target audience. In that world, nobody had to worry about real-time issues, memory constraints and a few different operating systems to support. Thus, it was extremely difficult to try to squeeze all this "freedom" into a well-working tool solution for embedded projects deploying small scale microcontrollers.

## UML with reliable, compact and targeted source code

The Unified Modeling Language (UML) with its real-time extensions is becoming a suitable approach for designing embedded real-time applications. However, it is not enough to view the nail from many different angles, if what you really want, is to hit the nail on the head in the best and fastest possible way. In this case, you need your design to be turned into an implementation through a fully automated process.

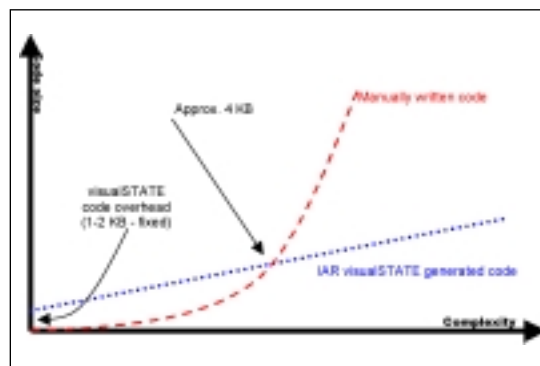
For this purpose, automatic code generation is the key, but automatic code generation from a UML design is not as trivial and straightforward as it might sound. Lots of details have to be consistent and precise in order to successfully convert a graphical representation to source code.

One of the hardest jobs is to generate fully functional and executable source code, which must not subsequently be modified by the software developer in order to get the embedded application running. This requirement must be combined with the constraints of small microcontrollers that are often used in embedded projects. Of course many embedded projects use 32-bit processors or higher, but just as many (if not more) realize the application using 8-bit or 16-bit chips. Small chips are typically easier to program and they cost less.

Another important factor is memory. Memory prices are continuously fluctuating so a company producing e.g. 1,000,000 units can save considerable amounts by trying to stay below a 32 Kbytes limit instead of selecting a 64 Kbytes module.

The automatic code generation facility of visualSTATE is built around a principle of having as much production-ready code available as possible to the developer at all times. By the term *production-ready* code we mean code generated by the tool that is directly capable of fitting into even the smallest controllers on the market, and at the same time having real-time properties. This also means that the code is not made for simulation purposes for example, although it handles this purpose very well too.

The code is very compact, but just as important is that the code size increases linearly compared to manually written code which has a tendency to increase exponentially the more complex it gets (see picture). Of course this also means that applying the automatically generated code for very small and non-complex applications (below 4 Kbytes of code) normally does not offer efficiency gains because the overhead and relative code size will exceed the overhead and size of manually written code.



The automatically generated code produced by visualSTATE carries a digital signature, which can be embedded in the source code. This means that maintaining and bug fixing already released versions becomes easier because it is always possible to track and trace the exact model from which the code was generated.

## Embedded tools for embedded engineers

Embedded engineers should be able to purchase and deploy tools that are specified and verified by embedded engineers, and which only require a learning curve compared with learning a general spreadsheet program.

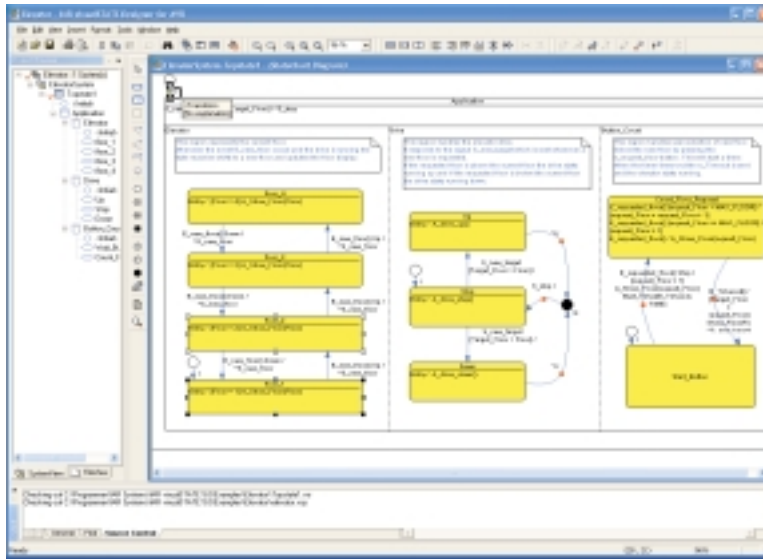
In today's tool market there exist some, but few, tools that originate from the embedded world. One of these tools is IAR Systems' visualSTATE for Atmel AVR controllers.

This product focuses on providing an embedded tool solution for embedded engineers, who wish to deploy a visual programming approach, while still being able to use even the smallest micro controllers available in their projects.

IAR visualSTATE for AVR offers a full UML compliant statechart design environment, where control related problems can be described and maintained.

Included with visualSTATE for AVR is also a comprehensive test suite that includes formal, manual and on-target debugging and testing capabilities. Furthermore, visualSTATE provides a unique code generation engine, that is built for embedded devices and supports both C and C++ implementations. Full Microsoft Word or HTML formatted reports can be automatically generated, which again ensures that code documentation is always up-to-date and easy to create and maintain. All-in-all IAR visualSTATE supports the embedded project in its entire life-cycle, from birth to death.

IAR visualSTATE for AVR is developed for corporations, which are project oriented rather than department oriented. This basically means that IAR Systems is able to deliver a low-cost turn-key solution for most projects, eliminating or at least reducing the rather large investments in both time and money, which are normally required when searching for productivity- and quality improving tools for the next embedded project.



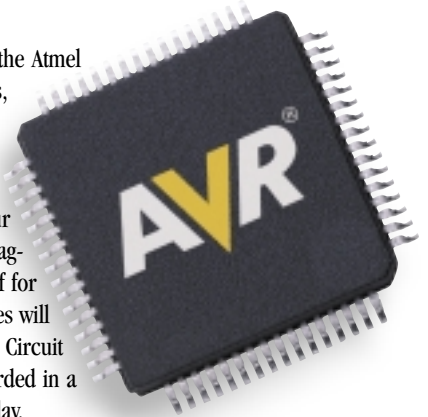
Picture: IAR visualSTATE for Atmel AVR

# New Atmel Design Contest!

## AVR<sup>®</sup> 2004 DESIGN CONTEST

When you choose an Atmel AVR microcontroller to jumpstart your entry to AVR Design 2004, your finished design becomes a gateway to incredible rewards. Not only will you be a "winner" for choosing one of the fastest, smartest, and the most readily supported microcontrollers in the world, your design will be in the starting lineup to be reviewed by leading industry experts. Our judges will be giving their full attention to your entry as they decide which designs promote the truly innovative spirit of the AVR microcontrollers.

Much like the capabilities of the Atmel AVR family of microcontrollers, your contest entry options are nearly unlimited. Simply present us with a working design that demonstrates your mastery of microcontroller imagination. Then prepare yourself for one wild ride – winning entries will be posted or published in Circuit Cellar, and prizes will be awarded in a variety of categories. Enter today.



Start your journey today towards a successful design at: [www.atmel.com](http://www.atmel.com)

Coming in March

# CIRCUIT CELLAR<sup>®</sup>

THE MAGAZINE FOR COMPUTER APPLICATIONS



© 2004 Atmel Corporation. Atmel and the Atmel logo are registered trademarks of Atmel Corporation.