Optimized run time behavior

Profiling in software technology means that information is processed, which helps analyze the specific execution time of application POUs. For this purpose, the Codesys Profiler is provided as an add-on component.

After being installed in the Codesys Development System, the Codesys Profiler automatically measures single tasks of IEC 61131-3 automation projects. The measuring results allow users to optimize the runtime behavior and thus the quality of their control applications. The Codesys Profiler is part of the Codesys Professional Developer Edition since May 2015.

These days, manufacturers in the automation industry supply hardware of different performance levels. Users of these hardware devices work with tools like the Codesys Development System to develop the necessary control programs in the languages of IEC 61131-3. The processing power of the hardware has become more and more exhausted in this process. If users realize that their application code is reaching the performance limits of their controller, they can optimize the application code before taking the software into operation. Thanks to methodical support, the data necessary for the optimization process can be determined without additional hardware or application software. For example, the runtime of the CANopen Safety Stack can be measured and taken into consideration.

The task configuration of the Codesys Development System already provides information on the (maximum) cycle time of the used tasks. The Codesys Profiler is an optional add-on tool and supplies application developers with detailed runtime data of the control program on POU level. In order to identify these data, the activated add-on generates additional machine code wherever there is a function input or output when compiling the IEC 61131-3 application code. This machine code is transferred to the target system along with the application code and then executed. In contrast to a manual procedure, a measurement of all POUs is automatically ensured, thus minimizing the time needed for runtime measurement. Looking at the data gathered, application developers can easily identify which POUs are most relevant to the total running time of the application and consequently make modifications to optimize time behavior.

During code runtime, the measurement results can be demanded, loaded, and displayed directly in the development environment at the user's discretion. As illustrated in the picture, different views allow the user to choose different perspectives to analyze the situation. POUs passing the pre-defined threshold value, limiting their share of the total running time of the application, are marked in a different color. Thus, users can identify POUs with a particularly long running time. Most helpful information is provided by the display of the call frequency of POUs, as these data allow for the user to determine the relevance of a certain POU for the total running time.

After finishing the runtime analysis, the Codesys Profiler can be deactivated with a mouse click. The optimized application code is then loaded onto the target system. Whereas manual measurement leaves some residue in the final code, this is not the case when using the Codesys Profiler. When comparing historical with current values, all results should be stored in the runtime analysis. By regularly evaluating these results, the efficiency behavior of the application is displayed over the whole development period. In this way, problems are revealed at an early stage, thus putting the user in a position to estimate whether the projected task execution times can be accomplished.

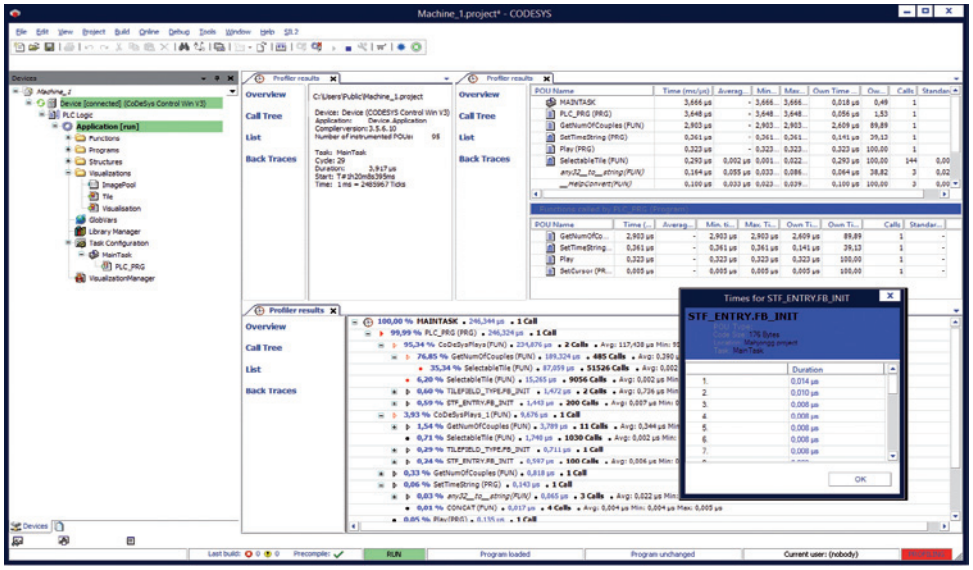

Figure 1: Integrated profiling results with helpful information (Photo: Codesys)

Using tools which provide methodical development support improves the quality of IEC 61131-3 applications in automation technology and at the same time reduces the necessary effort. The Codesys Profiler for runtime analysis helps users identify and prevent potential

problems caused by controller overload. It can be purchased in the Codesys Store and can be directly installed in the Codesys Development System. With the available trial version, users can test the product before use.

Author

Michael Schwarz
3S-Smart Software Solutions GmbH
www.codesys.com

Access to CANopen via IEC 61131-3 devices

The CiA SIG (special interest group) Application Layer is currently finalizing the CiA 314 specification. Usage of this document allows standardized access to CANopen services from devices programmable in IEC 61131-3 languages. Such devices may be PLCs (programmable logic controllers), PC-based controllers, HMIs (human machine interfaces), etc.

The document specifies function blocks to produce or consume CANopen communication services and to provide local CANopen functions. These include, among others, SDO (service data object) read/write access, NMT (network management) control, emergency message handling, as well as the creation of object dictionary entries. The available CANopen functions are based on CiA 301 v. 4.2, which has no relation to CAN FD (CAN with flexible data rate).

CiA 314 derives from the withdrawn CiA 405 document, which provided a different addressing scheme. The new addressing is as follows: All CANopen communication stacks in a system have a unique number (kernel-ID). A kernel operates on one CANopen interface port (interface-ID). Specific CANopen devices in the network are still addressed via their node-ID. In the system, a mapping table exists, which assigns kernels to interfaces. Thus, the kernel-ID implicitly specifies on which physical CANopen interface port the communication is running and the interface-ID is not used for addressing. The benefit of this addressing is that a kernel may later be moved to a different interface port without changing the available application code. The kernel-ID stays the same. This makes the hardware addressing transparent to the user.

To simplify the specification use in different systems, some platform-dependent parameters, e.g. time stamp and pointers to data, are left for the user to define. The timeout function (maximum allowed execution time) is also included into the function blocks.

The CiA 314 specification substitutes a part of the CiA 405 document, whose content was also partly moved to CiA 302-4 (network variables and process image) and CiA 306-3 (network variable handling and tool integration). The application note CiA 809, which is currently under development, provides an implementation and a user guideline for IEC 61131-3 devices.

Olga Fischer (CAN in Automation)