eXtended Automation Engineering

TwinCAT 3: C++ drives automation

With TwinCAT 3, the latest software generation from Beckhoff, PLC programmers can now also use C/C++ for more convenient programming, in addition to the traditional IEC 61131-3 languages. The application of C++ as SFC in Beckhoff controllers is not new: Even the DOS-based S2000 PC controller, developed by Beckhoff in 1993, was able to integrate C as SFC.

In 1993 Beckhoff introduced the S2000 controller, a Microsoft-DOS-based control system that was real-time capable through pure software extensions and enabled processing of PLC, Motion and visualization on a single CPU. However, as a single-task operating system DOS can only start one program at a time. If another program is to be executed, the one currently being processed must be terminated first. TSR (Terminate Stay Resident), which was activated through interrupts, was used to enable background functions and to retain small auxiliary programs in the memory after the end of the main program. The S2000 software distinguished between three task levels: the NC task with the highest priority, followed by the PLC task and the foreground task and the user interface, which was executed in the remaining available computing time of the PC. A scheduler was used for switching between the task levels. The S2000 foreground task consisted of cooperative multitasking, which made user inputs in parallel to background functions possible, thus removing the single-task limitation of the DOS operating system. The visual display of the user interface was implemented by textual graphics with 80 x 25 characters and an ASCII character set. As an example of a background task, C++ could be used to establish communication with a database in order to obtain new machine orders or acknowledge completed orders.

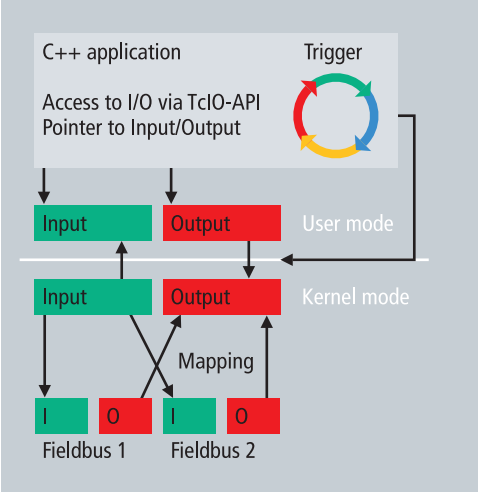
Limited hardware resources: one of the biggest challenges

One of the biggest engineering challenges for C++ developers was a platform with very limited hardware resources: DOS only provided 640 kB of RAM, which had to cover the application with Motion, PLC, HMI and its own C++ code. Auxiliary tools such as QEMM provided moments of bliss, since the Quarterdeck Expanded Memory Manager enabled the available memory to be slightly extended beyond the 640 kB limit. Nevertheless, it was rarely possible to load the whole application of the newly created C++ code in the memory for debugging. Finding errors turned into a kind of sport, so to speak, since only a small number of sub modules and associated debugging information could run in the limited memory. Today's TwinCAT C++ programmers smile about resource and memory limitations, although many developers of non-Beckhoff embedded solutions are still struggling with this issue on a daily basis. Lack of online monitoring, diagnostic or debugging options are addressed by outputting test signals on a serial port: "Welcome to history!"

TwinCAT CE offers fieldbus infrastructure for C++

In Windows CE, Microsoft has developed an operating system which is compatible to desktop Windows and can be finely scaled in terms of size and function. Originally planned for use mainly in mobile devices, Windows CE has been real-time-capable since version 3.0 and is used increasingly in the industrial environment. Due to the relationship of Windows CE with the "larger" Windows, it was possible to adapt "TwinCAT CE" to "TwinCAT XP" with compatible source code. Since TwinCAT has mapped all real-time functions on Windows CE, TwinCAT real-time applications such as software PLC and Motion Control can coexist with real-time applications of a different origin. In addition to PLC and motion functionality, TwinCAT CE offers the option of running a deterministic cycle that is synchronized down to the fieldbus from its own native C code. The TwinCAT system is sufficiently flexible to enable the physical inputs and outputs of different fieldbus systems to be mapped to logical I/O tasks initially. The pointers to the logical input and output process images are available in the C application, as is the TwinCAT real-time callback. From this deterministic and cyclic callback, reading of the physical inputs can initially be triggered with a certain method before the logical outputs are distributed to the physical outputs after the actual logical calculation.

C programmers value the fact that the TwinCAT CE solution uses an open automation platform that enables C code development for other operating systems to continue to run as cyclic logical code with minimum adaptation: Neither the connections to different fieldbus systems nor the connectivity with the HMI and ERP worlds have to be implemented separately, since they are already included in the infrastructure of the TwinCAT CE system.



TwinCAT 2 CE devices enable processing of deterministic C code right down to the various fieldbus systems.

Stefan Hoppe, TwinCAT Product Manager, Beckhoff Automation

TwinCAT 3 C++: extended automation platform

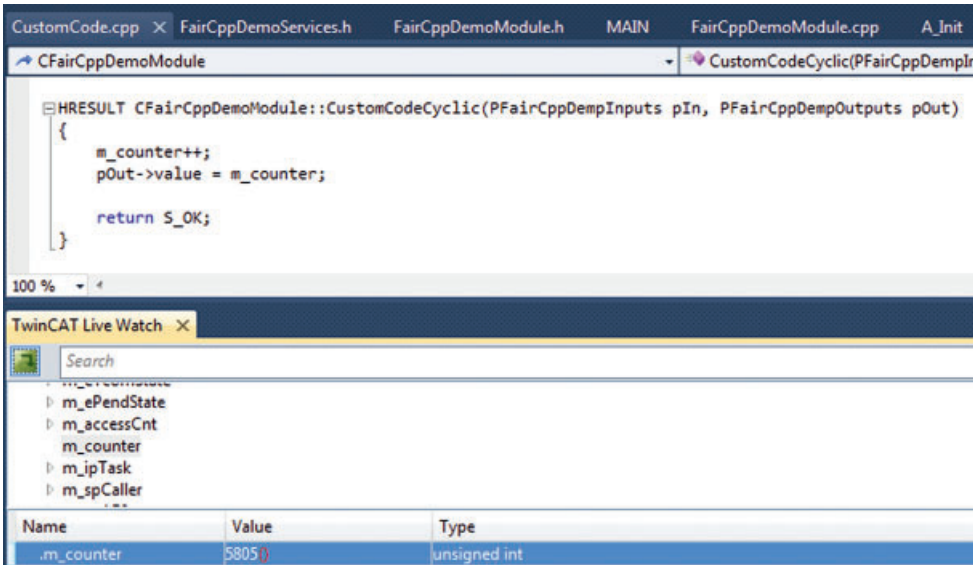
C++ with TwinCAT 3 offers developers a high-performance, cutting-edge editor in the form of Microsoft Visual Studio® 2010: Source-Safe or Team-Foundation integration for code backup and versioning is only one of several advantages offered by the new engineering environment, which greatly facilitates teamwork. Faster and more effective engineering of C code sounds like pure marketing speak, but the TwinCAT 3 automation platform does in fact offer C programmers previously unknown opportunities and freedoms: C code can run cyclically for accessing the physical I/O level, or it can only provide mathematical functions that are called up on demand by other C modules or the PLC programmer. Based on multi-core CPUs and simple configuration, several instances of the C module can be easily distributed to different CPU cores. Here too all the basic functionality is available: If actions are to be started in the C code that require a longer processing time than the cycle time allows, then these tasks must be decoupled. Beckhoff makes a "Software Development Kit" (SDK) available for this in order to start actions from the deterministic real-time process and to monitor the processing status. The reading and/or writing of files, the starting of threads, the allocation of memory and communication with databases takes place via functions from the SDK and corresponds to the mechanism of using PLC libraries familiar to the PLC programmer. The C compiler contained in Microsoft Visual Studio® 2010 is used for the generation of C code. After the compilation, the object code is loaded into the real-time runtime system in the form of dynamically loadable libraries (DLL). The user is thus able to extend the TwinCAT system using program modules. Diagnostic options, such as the monitoring and changing of process variables during a running PLC cycle, are matters of course for PLC programmers.

A C developer initially sets a break point in order to stop the processing of code so that the status of the symbols can be modified in watch windows. Here too, Beckhoff has extended the Visual Studio® options and created ideal diagnostic options for the use of C/C++ in automation technology: A custom debug transport channel was created with the aid of the "Custom Debugger Interface." The online data are subsequently available in a "TwinCAT watch" window for monitoring and changing, without having to stop the machine sequence with a break point. This kind of real-time application diagnostics is available both locally and via the network.

Welcome to the future!

The application of C++ in automation technology is not new. However, TwinCAT 3 provides a new, high-performance modular concept, which, in combination with the scalable multi-core CPU options and Beckhoff's extended Visual Studio® C++ monitoring, offers much more flexible, fast engineering: "Welcome to the future!"

www.beckhoff.com/TwinCAT3



Under TwinCAT 3, C++ variables can be monitored with or without break points.