

Why make automation complicated when it can be so easy?

Developments in automation generally have an eye on time and cost savings. Reducing these is a declared objective.

With the new automation system PSS4000 Pilz offers a solution for all machine control and safety technology functions, including motion control, diagnostics and visualisation. The objective was to simplify the decentralisation of control functionalities and reduce the engineering work. The groundbreaking software platform PAS4000 plays a key role. It enables users to retain a centralised perspective, even of decentralised, distributed control structures.



All of the software platform's tools for programming and parameter setting functions are fully compatible, with standard data and user interfaces. PAS4000 is based on proven standards such as all of the IEC61131-3 programming languages. These have become established not only on the market but with users too, which is much more important.

What's more, Pilz was able to base PAS4000 on a proven tool and develop it further: the easy to use PNOZmulti Configurator. With this graphics editor Pilz has virtually created a standard for safety-related programming. Thanks to the expansions on the PAS4000, this editor – known as the PAS multi – is now even more flexible to use, serving as a central tool for networking the available automation components. And not just for safety-related applications, as in the past, but also for automation functions in a non-safety-related environment.

What this means for engineering

Key differences in the way the system is handled are already evident at the start of an automation project, when, after the initial project approval the actual work begins. With the automation system PSS4000, the hardware and software design can begin immediately, for the software can now be developed independently of the hardware. Ultimately the required hardware can also be developed faster, for the automation system PSS4000 enables the system and hardware development to run in parallel.

Certain aspects need to be considered when selecting control components for the hardware. The following are important: what cycle time does a machine need to achieve; what is the operating and monitoring structure to look like, i.e. how many HMI devices and where; which devices are to represent which aspects? Another point that needs to be considered is the structure of the automation devices and programmable logic controller (PLC). For example, is each individual machining station to have a PLC or will several stations be operated from a common control system? All these are matters which are usually cleared up prior to design, to avoid causing the considerable expense of subsequent modifications.

The automation system PSS4000 opens up a whole range of possibilities in this regard. Due to its structure and properties, decisions such as the selection of a PLC to suit the design can be postponed to a much later point in the

design phase. With PSS4000 therefore, it's possible to concentrate on the design of the components required for the process; issues of performance and structure can take a back seat.

Components ease the burden of programming

As well as new hardware the PSS4000 also offers a new programming tool: PAS4000 – the Pilz Automation Suite. The key aspect here is component-oriented programming. Previously it was only possible to select ready-made safety blocks from libraries, but now it's also possible to create your own blocks, in this case components, and manage these in libraries.

The selection of ready-made components from libraries is itself nothing new. However, PAS4000 also provides the option to create your own components and to manage these within libraries. A special feature in this case is that components can be assigned "properties". These allow the parameters for the required functions to be set simply via drag and drop, as with the PNOZmulti Configurator. This is a particular benefit in terms of the standardisation of functions: previously the user had to ask himself how many functions he could bundle into a block without adversely affecting clarity. In practice the result was either a vast number of blocks, or too few blocks with a vast number of parameters.

With PAS4000 it's different. Here the user "writes" his optimum block – also called component – and creates a user level for its application, through which he selects the various functions. As a result, only the selected function is visible as a parameter; the block itself hides any complexity and remains easy to manage for the user.

Parallel engineering

The use of components has an additional advantage: Programming and hardware design can be split and can be run in parallel, not in sequence as before. It's common practice for the software development only to start once the machine specification has been established. With PSS4000, these processes can be split and run in parallel. As soon as the machine functions are standardised by creating components within libraries, ultimately the overall structure of the machine you wish to build can be designed with just some initial information. The detailed function of a component can be established later via the individual properties.

The components that the machine will require can be roughly deduced from the initial project approval of all the functions. Detailed functions can be defined retrospectively via properties. To make it even simpler, "aspects" have been defined as structure elements. Filter functions are the issue here: as well as its actual functions, each component has additional tasks such as diagnostics or visualisation. It's possible, therefore, to concentrate on one task. Only this one task will be displayed in the editor. With such an approach it's possible to add a mechatronic structure to the task structure.

As well as component libraries, PSS4000 also offers an additional automation model: a set of rules for component interconnection. The automation model also considers issues such as operating mode, interlock, diagnostics or visualisation. As a result, the user has a tool which he can use to generate a complete automation solution in the simplest way possible – particularly in view of the stock of components.

Simplifying communication

Normally considerable amount of work is involved in establishing and maintaining communication between subscribers on a modern machine network.

With PSS4000, the communication setup is no longer explicit, but implicit. All information is known and accessible equally to all network subscribers. Communication functions go beyond the norm, extending to data exchange between control systems and distribution of the application program across multiple control systems

The system platform PAS4000 also offers additional supporting auxiliary functions. The compiler function for example, which runs in parallel to the software generation and ensures that invalid entries or structural errors are displayed immediately. Download times are significantly reduced. PAS4000 can also run functions such as ToolTip, entry completion or suggestions for whole sections – similar to an “IF then Else structure”.

With an expanded editor, the components, the automation model, intelligent communication support and integrated tool support, these innovations ultimately help not only to reduce the engineering effort but also to reduce engineering costs.

Contact: Pilz Safe Automation
Phone: 03 9544 6300
Email: safety@pilz.com.au
Web: www.pilz.com.au