

Robot-based Automation: 3 Tips for a Time and Cost-efficient Implementation

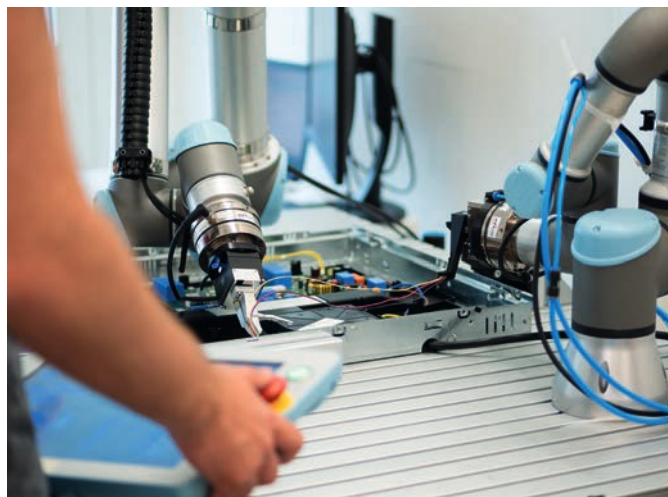
Why no-code & low-code tools have become indispensable in robotics

Using robots is almost always worthwhile for companies. They can reduce labor costs, relieve employees, and make production more flexible - because the required batch sizes are becoming smaller and production processes more individual. In addition, a robot or cobot offers another major advantage: it can work without breaks and fatigue, thereby increasing product quality and reducing scrap.

With modern robot systems, almost all processes can now be automated. Usually, robots take over simple, dirty, monotonous, physically demanding, or even dangerous tasks. However, with the right hardware and, above all, software, very complex or particularly demanding tasks, so-called "Advanced Robotics" applications, can also be solved. Examples of this include the assembly of flexible and bendable components such as cables, wires, or hoses or force-controlled surface processing.

Following the no-code/low-code trend, there are various software solutions on the market that enable graphical and therefore simplified and faster programming. The advantage is that no special programming skills are required. The portfolio ranges from manufacturer-specific solutions to independent offerings that can be used to program robots from different manufacturers with one single software. Especially in the latter case, experts recommend using tools that automatically generate native robot code for the particular robot controller instead of controlling the robot arm via a separate IPC.

In the first case, users remain flexible when it comes to adjustments or optimizations during operation and avoid a lock-in effect, as they can con-



Advanced robotics applications such as the assembly of flexible cables pose a major challenge for classical robot programming

tinue to program the robot in the traditional way using line code even without using the software.

Whether with external engineering tools or line code, there are factors on the path to robot-based automation that users often underestimate and therefore do not pay enough attention to. How to avoid three of the most important stumbling blocks will be briefly explained below.

Factor 1: Programming effort

The time required for programming an application is usually not underestimated, but there are other pitfalls in this phase: for example, process tolerances and variances that have not been taken into account or the increased complexity when incorporating sensors or establishing a communication between the robot and a PLC. In addition, programming a system is often a tailor-made and complex solution that is difficult to adapt. Moreover, programmers often have their own style, which can make the resulting code or program difficult to understand and modify for other programmers. At this point, the market supports the user with the aforementioned no-code/low-code solutions. For example, with pre-defined function blocks, programs can be constructed and structured in a clear and understandable manner for others. Process tolerances and variances can



No-code/low-code tools simplify and speed up robot programming thanks to their template-based approach





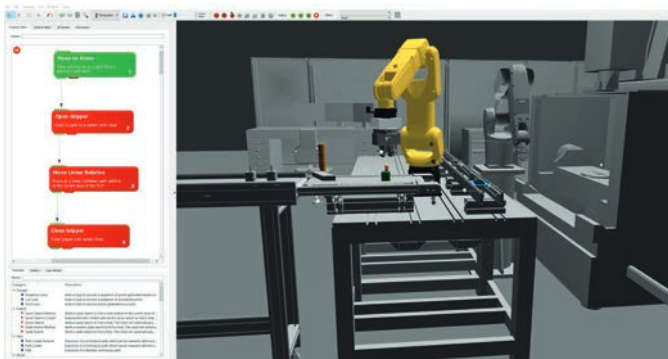
also be automatically compensated, analyzed, and optimized by using the right software. If the corresponding interfaces are already integrated, the effort required to connect sensors or set up a PLC communication is also immensely reduced.

Factor 2: Effort During Commissioning

A rule of thumb states that the cost of the robot itself is only about one-third of the initial cost of the cell, and experience shows that around 45 percent of the typical costs are incurred during ramp-up. This is because users often underestimate the time required for commissioning. Although the system is programmed offline and simulated throughout in advance, differences between theory and practice often become apparent during commissioning.

Thus, despite good preparation, the process can take significantly longer than planned, and adjustments and changes can quickly become expensive. In addition, factors that were not apparent during digital preparation now need to be addressed on-site during commissioning. This makes this phase difficult to calculate. With consistent tools, the ramp-up can be implemented in a controlled manner and without great loss of time, so that this phase does not become a cost driver. It is important to combine simulation, programming, sensors, and data analysis in one single software package. This way, the engineering chain becomes consistent without the user having to compromise on functionality.

This makes changes and adjustments faster, more flexible and easier. With ideally automatically generated robot code and the ability to transfer teach points back from the real robot into the software, such a solution integrates seamlessly and optimally into existing commissioning and



Programs can be structured and made traceable using pre-defined function blocks. Before commissioning, the process can be simulated as realistically as possible in a 3D simulation environment and tested

When teach points can be transferred from the real robot back to the programming software as well, such a solution seamlessly integrates into the commissioning process and saves unexpected effort

maintenance processes. This also offers the greatest flexibility in terms of online and offline programming, allowing the user to choose the best option and easiest way for the respective task.

Factor 3: Changes during the lifecycle of the cell

Even when the robot is running, there is still a danger that is often not considered: Over the system runtime, numerous changes in general conditions can occur that require adaptation of the programming.

These can be vibrations and shocks, for example caused from forklifts or other machines, wear and tear of tools, replacement parts that react differently than the previous components, changes in workpiece batches, and the space available in the hall or a change of the operating personnel. Changed lighting conditions and temperature conditions or the difference between a cold-started and warmed-up robot can also have an impact.

Software that standardizes and simplifies programming allows the user to react simply, quickly, and flexibly to these and many other changes and to make necessary adjustments in the program itself. If the worker uses a tool that also makes changes or the resulting consequences visible early on in terms of forces, cycle times, or defects and quality, it becomes easier to analyze and derive possible optimizations, which makes the user best prepared.

Silke Glasstetter, Artiminds

►► 63699 at www.ien.eu

Model GY series / **Magnetostrictive Displacement Transducers**



Model GY series are absolute type, high accuracy and non-contact linear sensors employing magnetostrictive phenomena. Compact, mini series, GYcRS series which enables alarm, velocity output, Electronic all-in-one GYSE series, $\phi 25 \times 22.5$ mm profile GYKM series, GYDC-05 controller presenting 1 μ m resolution, all our line-up can answer to your requirements in variety.

■ Main Specifications		■ Typical Applications	
Stroke	max 7500mm	<ul style="list-style-type: none"> • as positioning sensor for advanced industrial machinery. • as displacement transducer that can be installed inside hydraulic/pneumatic cylinders. • as a high linearity liquid level detector for a wide range of applications. 	
Linearity	less than 0.025%FS		
Resolution	less than 0.005%FS		
Output	Analog/digital		
Allowable pressure	max 350bar(probe)		

STC It makes Technological Sense
SANTEST CO., LTD.
www.santest.co.jp

Head Office 2-51, Shimaya 4-Chome, Konohanaku, Osaka 554-8691, Japan
Phone. 06 (6465) 5561 FAX. 06 (6465) 5921
Tokyo Office Phone. 03 (3432) 1417 FAX. 03 (3432) 1337

►► 63137 at www.ien.eu



FREE DIGITAL SUBSCRIPTION