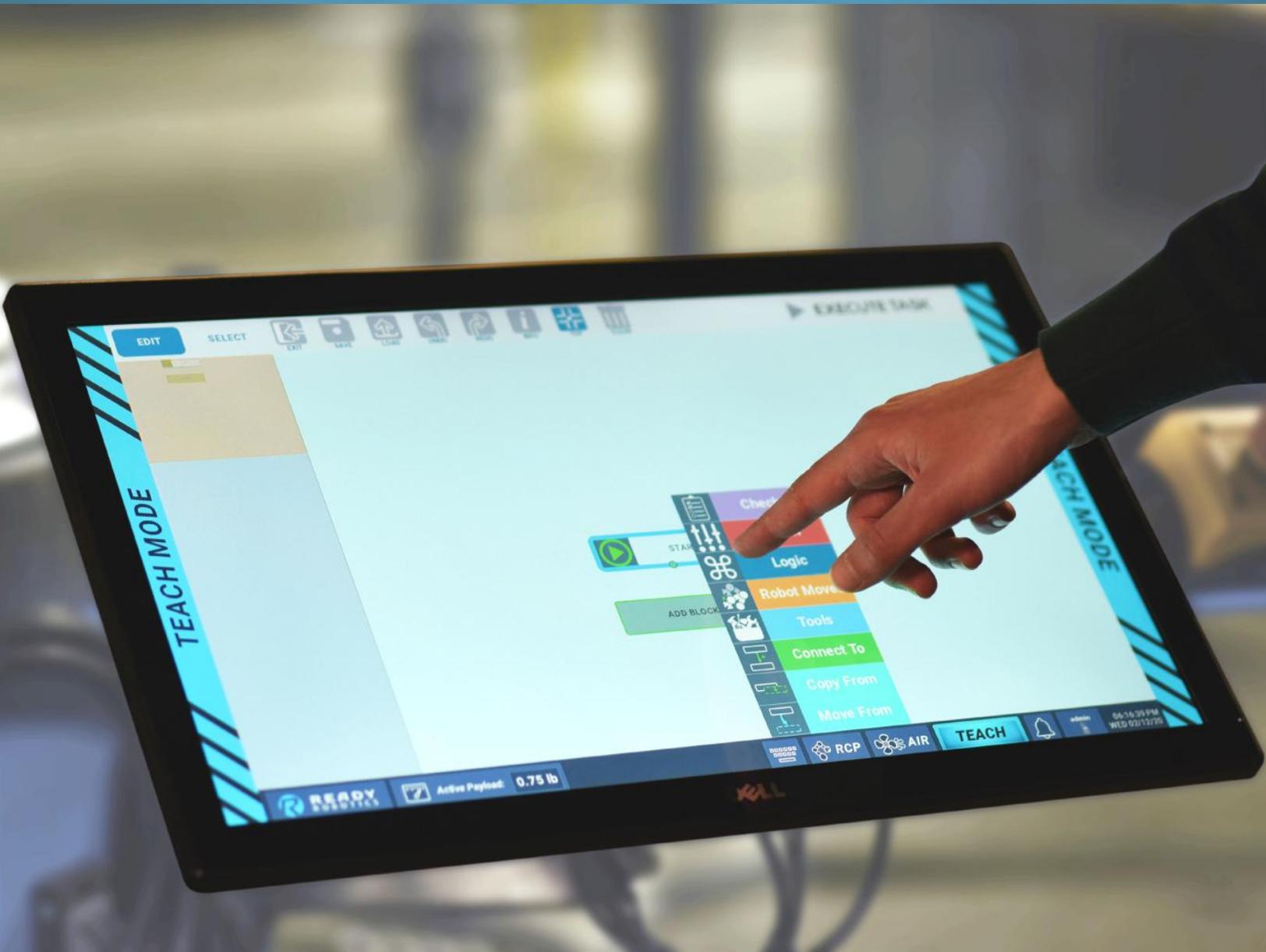


The No Code Revolution

How No Code robot programming democratizes manufacturing automation



INTRODUCTION

There has been a revolution with No Code solutions enabling more users to develop business applications without having to write a single line of code. In everything from website creation to mobile app production, No Code solutions have fundamentally reduced - if not eliminated - the barriers to entry by delivering high-skill level capabilities to the masses. Some call these users "citizen developers" since the No Code tools democratize the development of applications. No longer is development restricted to those who can write code in programming languages or the proprietary scripting languages common to many tools. It's a revolution because there are many, many more users than skilled resources who can write programs.

But wait a minute, what does the No Code movement have to do with robots? As we discussed extensively in our [5 Reasons Programming Robots is Hard](#) whitepaper, robots have very difficult to use programming environments. Even the vendors who have made robot programming a little easier still rely on scripting languages and procedural languages requiring extensive training for users to be productive. The revolution in robots is that there is now a No Code environment available from READY.



THE BENEFITS OF NO CODE ROBOT PROGRAMMING

In our articles and white papers, we talk extensively about being agile in the development of automation. Being agile in manufacturing automation means that you can do the proper level of planning and then rapidly implement the automation solution with a focus on immediate business value. The benefits of No Code robot programming compared to programming in an archaic robot programming language created in the 80s are:

- **By developing the application yourself, you know exactly how it operates.** No longer are your teams reliant on the integrator, or the single person from another factory, that developed the program in the robot's programming language. There is no part of the application your team won't understand since they developed it all. Team members are often CNC programmers, maintenance technicians, or machine operators who work with the machines being automated. By incorporating those closest to the processes undergoing the automation transformation, your likelihood of developing a successful automated process out-of-the-gate grows exponentially.
- **You are self sufficient.** Need a change? That's no longer a problem. Your team can do it. Updating robot reference points, changing a tool, changeover to an entirely new SKU. It doesn't matter. Your team knows how to do it. Your team can also rapidly troubleshoot any issues that arise.
- **Your team gets to continuously improve and upgrade their skill set.** With each task they implement, your team gets better. With a small amount of training, operators can program their first task. But when they see how easy it is, they'll want to program more, and do more automation. This excitement is how automation builds, and the teams will be able to move faster, with less iteration, on each new task they automate. This upskilling can also have the benefit of improving job satisfaction and reducing turnover.

- **Community.** Our goal at READY is to democratize industrial automation. This is not a landscape to be owned by one supplier, or be fragmented by dozens of OEMs. A universal industrial operating system with a No Code environment benefits everyone. No Code is fun because more workers are empowered. Fun means that workers will be creative, motivated, inspired and willing to support one another.

EXPERT KNOWLEDGE TRANSFER (THE OLD WAY)

One of the most valuable aspects of a No Code environment is how it can empower an expert to transfer their knowledge of a task to an automation program without a middleman. Traditionally, expert knowledge transfer doesn't always go smoothly because of how integration projects are executed. A typical project to develop robot automation for an existing workcell goes like this:

1. Manufacturer contacts an integrator to automate their workcell
2. Integrator visits the factory, observes, and writes down requirements from the customer and their expert in the process
3. Integrator goes back to their shop where they design the solution
4. The solution is then reviewed with the customer and signed off
5. The integrator develops the solution in their shop, and oftentimes through trial and error, develops the automation program that controls the robot in the robot's native programming language
6. When the solution is developed, a runoff is done in the integrator's shop after which the customer approves it. A runoff is done according to parameters specified during the design, and includes factors such as number of parts and the testing criteria to confirm the parts is processed correctly.
7. The robot solution is then installed in the workcell, a runoff is done on-site, and the project is considered complete



There are many challenges from this approach, which in enterprise software projects is called a Waterfall Approach. It is called a waterfall because as each phase is completed, the next one starts, and there is a single path where success is expected at the end. Success is oftentimes not accomplished because it's nearly impossible to capture all the requirements, especially with complex projects such as polishing a part, at the beginning of a project. With a waterfall approach, it is also very difficult to accommodate any changes during the development.

KNOWLEDGE TRANSFER IN A NO CODE ENVIRONMENT

When a No Code environment is used, the experts themselves can be directly involved in the development of automation. In a No Code environment, a project can more easily proceed in an agile way, as discussed in the READY article on [How to Launch Agile Automation](#). The main difference being that whether an integrator is used, or the integration is done on site, the development of the automation application is not completely done by the integrator. An integrator is still helpful if machine modifications or fabricated items are required. For example, a custom pedestal may be required along with modification of a cabinet in order to accommodate the robot. However, the key difference with a No Code environment is that the expert can be involved directly in the program development so there is no handoff of the code through a middleman.



PAIR PROGRAMMING: BORROWING BEST PRACTICES FROM THE SOFTWARE INDUSTRY

To involve the process expert during development we recommend a practice that the software industry calls pair programming. Since READY's Task Canvas is a No Code environment, the programming is done with visual building blocks, but we still call the activity of developing the automation "programming." Pair programming is when two or more people work together, simultaneously, on the same device when developing an automation program. This technique is extremely useful and has the following benefits:

- Two heads are better than one. If a problem is encountered, the process expert and automation expert can fix it in a cross-functional way in real-time.
- More efficient. Some would think that having two people work together slows down the development. However, studies have shown that two people working together are more effective since there is no lost time with communication. Also, the finished product is more likely to be correct.
- Fewer mistakes. Because there are two people, with different skills, it results in better implementation. In fact, studies show this practice results in 15% fewer issues than when developed by a single person.
- Training. Training of the process expert in the No Code environment is a natural outcome of pair programming. There is no better way to train than working on your own tasks in a hands on way.

When it comes to applications such as painting, polishing or sandblasting, or wet blasting it is difficult to capture all of the fine requirements. Therefore, if the operator can be directly involved, rapid iteration can be done in an agile way, allowing for fine-tuning the final aspects of the physical setup as well as the application.

CONCLUSION

No Code robotic programming is a revolutionary change in how robotic automation can be implemented. Not only can programs be written 2 to 10 times faster than existing robot programming languages, but they can be owned and maintained by the workers managing the workcell. In addition:

- Your team developed and owns the code.
- It is easy to make changes, updates and support new SKUs whenever needed.
- The team gets something working in hours, there is little training for anyone to get started.
- No Code democratizes industrial automation. Robot programming is now accessible to nearly anyone.
- Downtime is reduced since more people can manage the workcell

READY

At READY we are solving the problem that robots are too hard to program with our revolutionary visual, flowchart style, No Code robot programming application. Our Task Canvas application, running on Forge/OS, the first enterprise grade cross platform operating system for automation, enables anyone to program a robot intuitively with little to no training. We're also enabling an ecosystem of OEMs and software developers to create a platform where we can make industrial automation easy to design, develop and maintain by everyone in the factory, not just those with advanced degrees and advanced training.



Contact

Chris Harbert, VP of Sales chris.harbert@ready-robotics.com

Authors

Luke Tuttle, Chief Operating Officer luke.tuttle@ready-robotics.com

Ben Gibbs, CEO and Co-Founder ben.gibbs@ready-robotics.com

Thought Leadership Team

Kel Guerin, PhD in Robotics, CTO and Co-Founder

Josh Davis, PhD in Robotics, VP of Engineering

Jake Huckaby, PhD in Robotics, VP of Strategic Partnerships

Production Team

Erik Bjornard, VP of Marketing

Kirk Higgins, Product Marketing Manager

For more resources and thought leading content and case studies on automation:

www.ready-robotics.com/resources

For press inquiries, contact

press@ready-robotics.com

[LinkedIn](#)

[Instagram](#)

[Facebook](#)

[Twitter](#)



1080 STEELWOOD RD.
COLUMBUS, OH 43212
(833) 732-3967