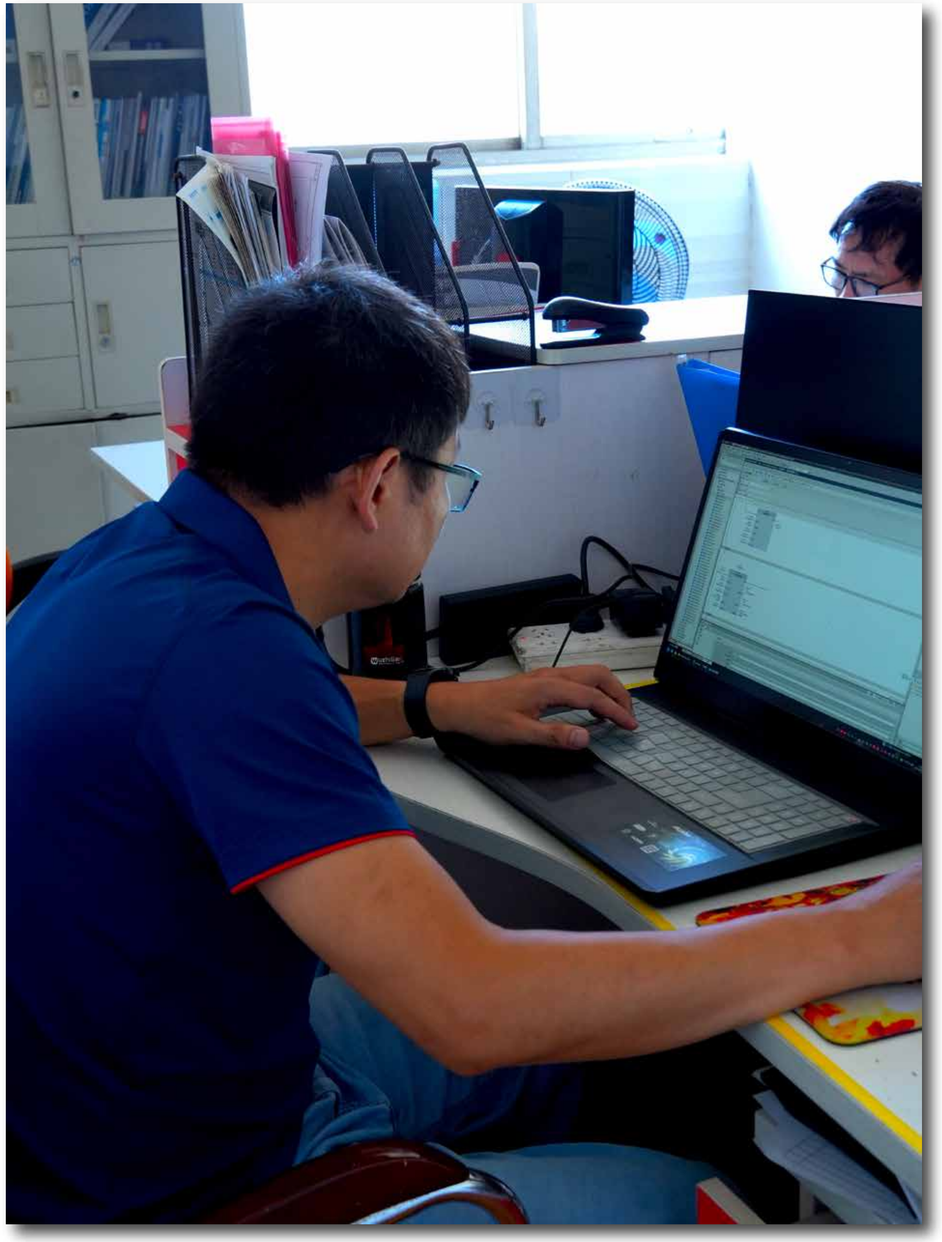
Q&A

with **Ouyang Ning**

PLC Programmer

Every signal matters,
every detail counts.



Introduction

My name is Ouyang Ning, and I work as a PLC Programmer at Jota Machinery. My role is to give our machines their “logic”—the invisible code that makes every motion precise and reliable. From ladder diagrams to structured text, I design and optimize control programs that connect mechanical parts with electrical signals, ensuring smooth communication between operators and machines.

What motivates me most is seeing lines of code transform into real production. When a program runs flawlessly—stabilizing tension, aligning blades, or responding instantly to operator input—it feels like bringing a machine to life. For me, programming is not just writing logic, it’s about responsibility: making sure our clients can run their machines with confidence, efficiency, and safety.

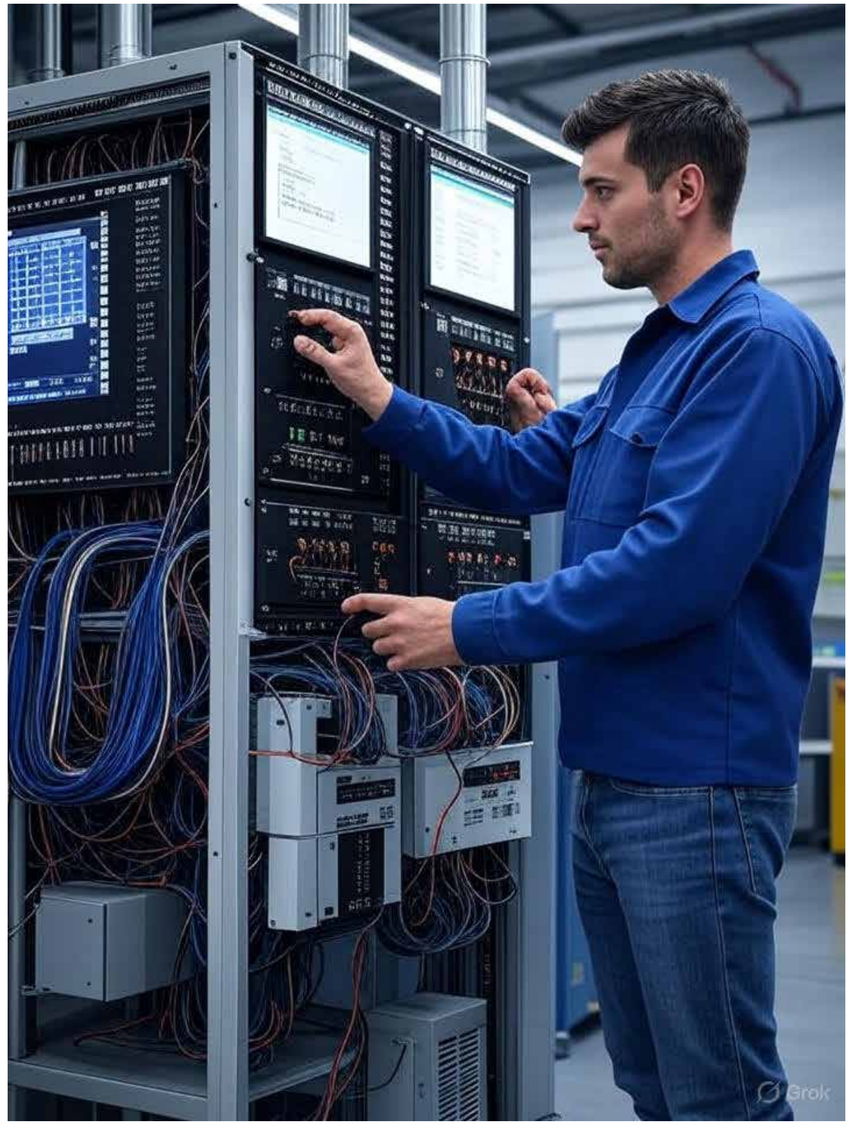
01 What inspired you to specialize in PLC programming?

I’ve always enjoyed problem-solving, especially when it comes to connecting hardware with logic. PLC programming is where mechanical motion, electrical signals, and digital logic meet. The idea that a few lines of code can bring an entire machine to life fascinated me early in my career.

02 What types of PLC systems do you work with most often?

At Jota Machinery, we mainly use Siemens and Mitsubishi PLCs, but I’ve also worked with Omron and Allen-Bradley depending on customer requests. Each brand has its own strengths. Siemens is very versatile for structured automation, Mitsubishi is compact and cost-efficient, while Allen-Bradley integrates well with large-scale SCADA systems.

03 When debugging a machine, what's your step-by-step process for locating faults?



My process is always structured:

Check power and wiring – make sure the hardware is correct.

Look at I/O signals – verify sensors and actuators are responding.

Trace logic flow – follow the ladder logic or structured text step by step.

Run simulation if possible – isolate the problem virtually.

Test incrementally – fix one issue at a time.

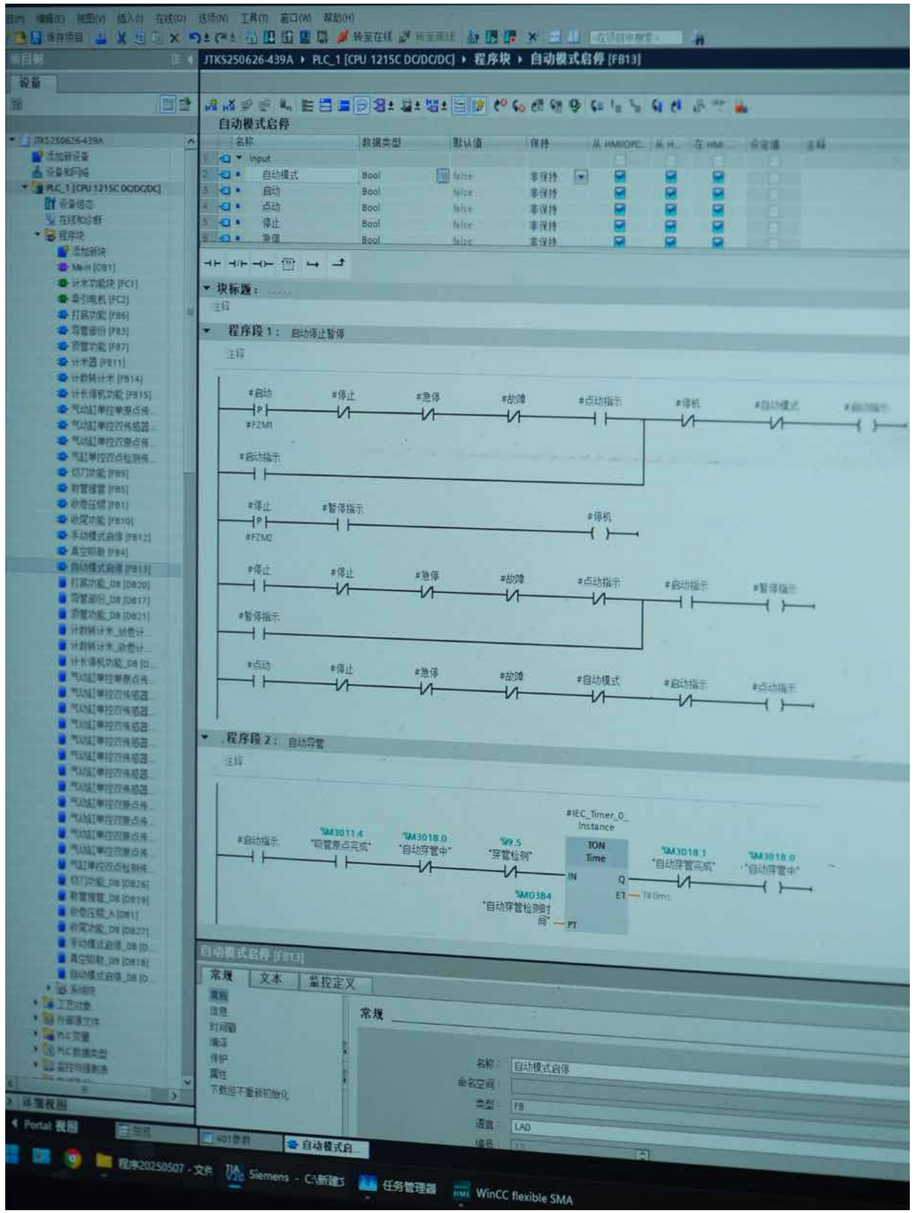
This systematic approach prevents wasted time chasing symptoms instead of root causes.

04 How do you approach writing clean and reliable ladder logic code?

My rule is clarity first. Every rung must be easy to read for the next person. I always use consistent tag naming (e.g., “MOTOR_START” instead of “M1”) and structure programs into logical modules: start/stop sequences, safety interlocks, alarms, and production cycles. I also add comments for critical logic, so future technicians won’t waste time guessing.

05 What’s the most challenging automation problem you’ve solved in the field?

Once, during a machine installation abroad, the rewinder tension kept fluctuating at high speed. The issue turned out to be not hardware but sampling time in the PID loop inside the PLC. After adjusting the loop timing and filtering the sensor input, the tension stabilized. It reminded me how software can solve what looks like a mechanical fault.



06 How do you ensure that a PLC program is user-friendly for operators, not just engi-neers?

I design from the operator’s point of view. That means clear HMI screens, logical button layouts, alarm messages written in plain language, and as few manual steps as possible. For example, instead of “Error Code 203,” the screen should say “Check left-side photo-electric sensor.” The machine should guide the operator, not confuse them.

07 How is debugging different in developed vs. developing countries?

Safety is built into the code as much as the hardware. I always add redundant checks for emergency stops, interlocks for guard doors, and fail-safes for critical motion. PLCs also make it possible to log safety events, which is important for CE and ISO compliance. My philosophy: a safe program is a reliable program.

08 How do you balance standardization versus customization when writing control logic?

I rely on standard modules for core functions—like motor start/stop, fault handling, and alarm logic. But every project needs customization, especially when clients use unique raw materials or production methods. The balance is to keep the backbone standardized, but allow flexible parameters for operators to adjust.

09 What’s your method for testing and validating PLC programs

We use a mix of offline simulation and real machine dry runs. First, I simulate the sequence in the development software to check logical flow. Then, once the machine is assembled, I test each module separately—unwinder, slitter, rewinder—before running full production cycles. Only after stable runs at rated speed do I sign off the program.

10 How do you keep up with the latest automation technologies?

I attend automation expos, study vendor updates (Siemens, Mitsubishi, etc.), and keep learning about Industry 4.0, IoT, and SCADA systems. At Jota, we’re exploring remote monitoring and predictive maintenance, so I also follow developments in sensor integration and cloud-based control.

What we build together, moves the world forward.

2025 EMPLOYEE VOICE

Pubslished date :

August 2025

Pubslisher:

Jota Machinery Industrial (Kunshan) Co., Ltd.

Address:

No. 68, Shaojing Road, Economic Development Zone, Kunshan City, Jiangsu Province, China

Email:

jotamachinery@gmail.com

Linkedin

TikTok

Website

Youtube

Instagram

X