Research Article
Open Access

Integration of a vision-guided robotic cell using PLC as process master

Abstract

This project presents the integration of a robotic cell with a vision system, composed of a KUKA industrial robot, a PLC designated as the process master, and an industrial camera. The focus is on the development of the integration along with the programming logic involved, highlighting how Ladder language in the PLC and KRL in the robot can be structured to ensure synchronization and efficiency. The project demonstrates the organization of the logical flow in the PLC, which is responsible for coordinating the cell, and the mutual communication between the devices. The types of exchanged signals, communication protocols used, and how commands are transmitted, processed, and executed are explored. In this way, the study aims to demonstrate how the PLC acts as the control center, commanding the interaction among the cell's devices. The results discussed show that using the PLC as the master provides greater flexibility in the process and facilitates logic adjustments without the need to reprogram the robot for every variation. This logical integration, combined with the vision system, enhances the robustness and scalability of applications for Industry 4.0.

Keywords: PLC, ladder programming, KUKA robot, KRL, industrial communication

Volume 11 Issue 3 - 2025

Murillo Henrique Souza da Cruz, Paulo Bueno Santos, Marcelo Campos Rebouças, Julio Cesar de Almeida Freitas, Rogério Adas Pereira Vitalli

Department of Robotics Engineering, Robotics Advanced Institute (IAR), Brazil

Correspondence: Murillo Henrique Souza da Cruz, Department of Robotics Engineering, Robotics Advanced Institute (IAR), Brazil, Tel 55(13)997180446

Received: October 1, 2025 | **Published:** October 20, 2025

Abbreviations: PLC, programmable logic controller; KRL, KUKA robot language; IO, input/output; ProfiNet, Industrial communication network protocol; Ethernet/EtherCAT, Industrial network protocol for real-time automation; DeviceNet, Industrial network protocol for device communication; Profibus-DP, industrial automation communication protocol; KCP, KUKA control panel; CMOS, complementary metal-oxide semiconductor; VGA, video graphics array; IP, internet protocol; PoE, power over ethernet; TIA Portal, Totally integrated automation; KOP, KUKA Option Package; SCL, structured control language

Introduction

The integration of robotic cells in Industry 4.0 goes beyond merely connecting equipment. The real challenge lies in organizing the programming logic that links the PLC, the robot, and the vision systems.[1] Structuring this flow requires clearly defining signal exchange of signals, the commands for each device, and the central role of the programmable logic controller as the process master. Challenges also extend to the variety of programming languages that professionals must master.[2]

In this work, we highlight the construction of a communication flow where the PLC is responsible for coordinating the robotic cell. The process was structured in stages, ranging from the initial configuration of the communication network to the management of data and execution commands. This logical sequence ensures that each device operates at the right moment with the correct information, always respecting safety requirements.[3]

The developed steps included configuring communication between the PLC, the KUKA robot, and the COGNEX camera; the initial exchange of test signals; the execution of logic in KRL; the integration of the camera signal; and finally, the consolidation of the entire flow under the PLC's command. In this way, we demonstrate that the effectiveness of the project lies in the control logic, which is capable of commanding the devices involved and ensuring cell synchronization.[1,3]

1. Communication
2. Integration
3. Result
4. Discussion
5. Conclusion

Materials and methods

The industrial robot used is the KUKA KR6 R900 sixx model shown in Figure 1, equipped with six axes, a maximum reach of 901 mm, a payload capacity of up to 6 kg, and the KR C4 compact control system.[4,5] Its peripherals include the KCP (KUKA Control Panel), a portable SmartPAD control panel with a 7-meter cable for manual operation and programming; an I/O board with 16 digital inputs (24 VDC) and 16 digital outputs (24 VDC); a power supply with a maximum output of 24 VDC/4 A for I/O board signals; and a communication system compatible with ProfiNet (master/slave), EtherCAT, Ethernet, and optionally DeviceNet or Profibus-DP, configurable via software. In addition, a Siemens controller model S7-1200, code 1212C DC/DC/DC (6ES7 212-1AE40-0XB0), with ProfiBus communication card CM 1243-5 (6GK7 243-5DX30-0XE0), was integrated into the system.

The industrial camera used is the Cognex In-Sight Micro 1400 ISM model shown in Figure 2, a compact and monochrome Micro series camera, designed for real-time inspections in industrial environments, especially in pick-and-place applications with coordinate detection (X, Y, and angle).[6] It features a high-speed 1/1.8-inch CMOS sensor (8.7 mm diagonal), with 5.3 x 5.3 µm square pixels and a maximum VGA resolution of 640 x 480 pixels (0.3 MP). The frame rate can reach up to 213 full fps at this resolution, with 8-bit depth (256 grayscale levels) and an electronic shutter adjustable between 4 µs and 500 ms. Triggering is performed through an opto-isolated input or via Ethernet commands, with an RS-232C option through an I/O module. Communication is carried out through a 10/100 BaseT Ethernet port with auto MDI/MDIX, compatible with DHCP, static IP, and link-local. Power is supplied via PoE Class 2 (Type A/B), with a maximum consumption of 6.49 W (48 V DC nominal).

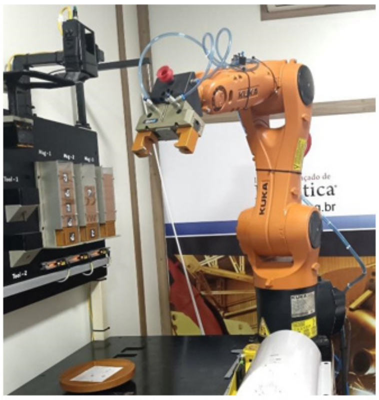

Figure 1 KUKA robot in HOME position.

Figure 2 COGNEX camera positioned for inspection.

The Siemens SIMATIC S7-1200 CPU 1212C DC/DC/DC shown in Figure 3 is a compact programmable logic controller designed for machine automation in mid-to-low performance ranges, such as coordinating vision and robotics systems in pick-and-place applications. It integrates onboard I/Os, supports ProfiNet as master, and is programmed via TIA Portal, focusing on efficiency and scalability. The control panel shown in Figure 4 contains the Siemens S7-1200 PLC, circuit breakers for electrical protection, ProfiNet connectors for communication between the PLC and robot/camera, and I/O terminals for control signals. It includes 24V DC power supply and outlets for peripherals, ensuring robustness and organization for continuous industrial operations.

Communication

To carry out the project, the methodology of flowcharts was adopted as an analysis tool to define the best integration strategy among the components. The starting point was establishing communication between the involved hardware and software, and after succeeding in this step, the development of operating logic began. The defined network topology positioned the PLC as the master of the Profinet network, responsible for receiving coordinates (X, Y, and angle) from the camera, processing them, and sending them to the robot, which executes the movements and confirms each stage via bidirectional communication. This architecture was implemented in TIA Portal as shown in Figure 5, requiring precise configuration of data types and addressing used in the logic. Variables were created in the PLC to enable both receiving coordinates and sending activation signals to the camera.

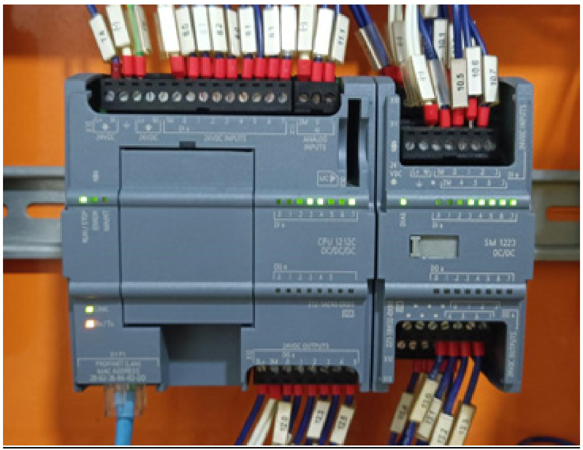

Figure 3 PLC with I/Os connected and its expansion.

Figure 4 Electrical panel composed of the PLC, camera power supply, safety relay, and drives, integrated with the Profinet communication switch.

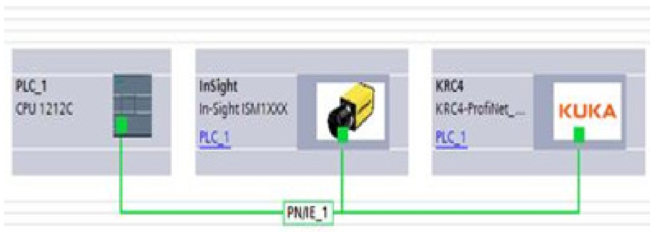

Figure 5 Profinet network where the PLC acts as master, interconnecting the camera and controller configured as slaves in the communication.

Citation: Cruz MHS, Santos PB, Rebouças MC, et al. Integration of a vision-guided robotic cell using PLC as process master. *Int Rob Auto J.* 2025;11(3):88‒92. DOI: 10.15406/iratj.2025.11.00305

Communication between the devices takes place via Profinet.[6] For sending coordinates, addressing had to be configured in the PLC, defining the creation of variables within TIA Portal that cover all necessary data exchanges, both for receiving information from the camera and for activation signals This step was fundamental to ensure that the PLC could correctly interpret the received data and command the robot with precision. The camera sends position and orientation data, and the PLC transforms them into movement commands for the robot, maintaining synchronization among the devices.

To integrate the robot into the system, the WorkVisual software was used, following specific procedures to configure the robot controller and enable variable exchange with the PLC. The integration relies on the KOP (KUKA Option Package) file, a configuration package containing libraries, drivers, and parameters required to enable Profinet communication between the robot and the PLC.[7] This file is already available in the KUKA controller and must be installed in WorkVisual so that the robot can be recognized as a device in the network. With this configuration, the complete system was able to establish efficient communication among camera, PLC, and robot, ensuring the integrated operation of the automated cell.

Integration

The programming logic developed for the integration between the Programmable Logic Controller (PLC), the industrial robot, and the computer vision system is represented in Figure 6. It was structured based on sequential and functional cycles.[8] After completing a task, the robot verifies whether there are other pending functions or if the cycle should be repeated. If new instructions exist, the robot returns to the waiting state and restarts the verification cycle. If there are no further tasks to execute, the robot ends the process and sends the PLC a confirmation signal of task completion. This signal allows the system to recognize the end of the operation and be ready to start a new sequence, promoting efficient and synchronized automation.

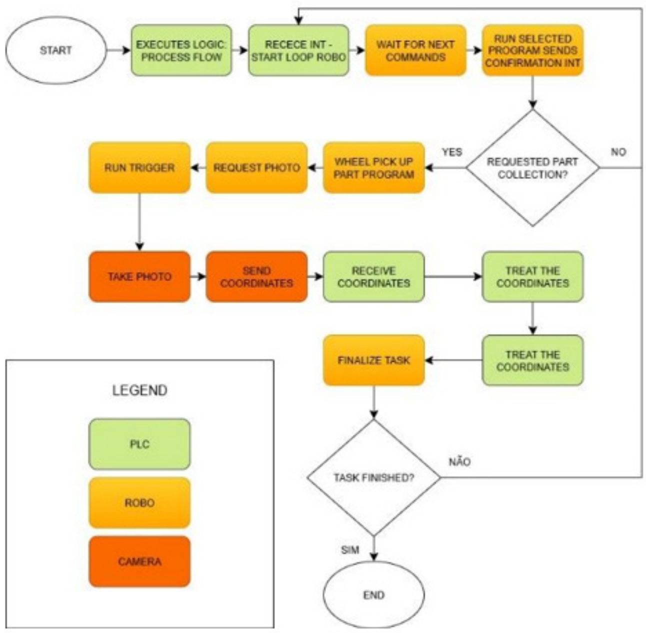

Figure 6 Flowchart of the integration logic, starting from the PLC and ending with task completion verification.

The PLC control logic was developed based on a modular structure composed of five blocks: Main, Camera_Config, Angulo_Cam, Camera_to_Robo, and Fluxo_Processo. Each of these modules performs a specific and interdependent function, forming a cohesive system for integration between camera, PLC, and industrial robot. The Camera_Config responsible for communication between the PLC and the Cognex Insight Explorer camera software. It manages control and monitoring signals, such as trigger commands, operational status verification, and fault handling. This stage ensures that the camera operates in synchronization with the rest of the system, allowing the captured data to be reliable and correctly used in subsequent steps. The Angulo_Cam represented in Figure 7, this is the core of the data processing logic. Its main function is to normalize the angles provided by the camera, adjusting them to the robot's operational range and avoiding singularity situations that compromise movement capability.[6]

```
#Atemp := "Angulo.Camera" - #offsetA;

WHILE #OK = 0 DO
    // Statement section WHILE
    //
    IF #Atemp > 180 THEN
        // Statement section IF

        #Atemp := #Atemp - 360;
        ;


    ELSIF #Atemp < -180 THEN
        // Statement section ELSIF
        #Atemp := #Atemp + 360;
        ;
    ELSE
        // Statement section ELSE
        IF #Atemp > 0 AND #Atemp < 165 THEN
            // Statement section IF
            #Atemp := #Atemp - 180;

            ;
        END_IF;

        #Arobot := #Atemp ;
        #OK := 1;
        ;
    END_IF;
```

Figure 7 SCL logic ensuring that the angle is suitable for safe part pickup without singularity.

The angle processing logic captured by the camera was developed to ensure that the robot, when performing the pick-up movement, does not enter a singularity condition. Singularities are geometric configurations where the robot loses degrees of freedom, compromising accuracy, stability, and, in some cases, preventing certain movements. To avoid this scenario, the system applies a series of corrections to the received angular data.

The process begins with subtracting an offset value from the raw angle provided by the camera. The result is stored in a temporary variable and passes through a repetition loop that ensures the angle remains within the operational range of ±180 degrees. Whenever the value exceeds +180 degrees, 360 is subtracted; if it falls below -180 degrees, 360 is added. This routine is repeated until the angle is normalized within the safe range.

Next, an additional verification is performed. If the angle is between 0 and 165 degrees, a correction of -180 degrees is applied. This step corresponds to a specific business rule designed to prevent the robot, when moving to pick up the part, from rotating in critical regions associated with the parameter known as start-turn. The start-turn represents the initial rotation point of one of the robot's axes, usually the tool orientation axis.[2] When rotational movement begins from an unfavorable angle, the risk of singularity increases, which may cause instability or lockups.

Thus, the logic implemented in the PLC works preventively, adjusting the angles so that the robot always starts its movements from safe positions. This way, the system avoids risk zones and ensures that the robot maintains its full motion capability during part pick-up operations. The Camera_to_Robo handles the conversion of position and orientation data to ensure compatibility between the involved systems. This step is critical due to differences in byte ordering between the Siemens PLC, which uses the big-endian standard, and the KUKA controller, which operates in little-endian. To prevent misinterpretations of data, the SwapDword function[6] is applied, which reorganizes the bytes of each two-byte word. This operation is performed both in transmission and reception, ensuring that values are correctly understood by both systems. This logic is essential for communication integrity and for the precise functioning of integration between camera and robot. Fluxo_Processo: Acts as the system's master sequencer, coordinating the robot's task execution based on the variable #passosPlc, which represents the current process state. Transitions between states depend on the robot's return, indicated by the variable #passoRobo. The logic begins with the conversion of input and output data byte order. Then, it monitors the robot's status and updates #passosPlc as steps progress. The work cycle starts when the system is in standby mode and the Start button is pressed, assigning the value ten to the control variable and initiating the sequence of commands.

The state machine is composed of transitions conditioned on the robot's confirmation of the previous step. For example, moving to step twenty (part pick-up operation) only occurs if #passosPlc is at ten and #passoRobo indicates completion of step ten. This interlock logic ensures that the stages of the production cycle are executed in an orderly and safe manner.

The system also includes recovery mechanisms, such as the reset_Robot flag, which allows forced transitions to specific steps in case of failures. When both the PLC and robot reach the final step fifty, the system returns to the initial zero state, preparing for a new cycle.[4]

In addition, the logic implements a startup reset mechanism via a short-pulse timer triggered by the Start button. This feature ensures that the value zero is quickly written into #passosPlc, forcing the state machine to start from the standby state and preventing incorrect sequences after PLC power-up. Based on the received coordinates, the robot performs the necessary movements to accurately locate and manipulate the part. This step is essential to guarantee process accuracy, especially in industrial applications requiring high repeatability and operational reliability.

Results

In the results, the most difficult part of the integration was dealing with multiple different programming languages. For the integration to work, it was essential to create a detailed process flowchart, separating the steps of each system and later uniting them into a general flow.[9] The hierarchical organization ensures an efficient, synchronized, and conflict-free Figure 8 communication flow, minimizing errors and optimizing execution time. The reduction in cycle time from 2:54 to 2:28 minutes equivalent to approximately 14.94% highlights the practical benefits of this approach. For future work, integrating cloud-based PLC control and machine learning techniques could further enhance scalability in Industry 4.0 environments, potentially reducing cycle times even more through predictive optimization.

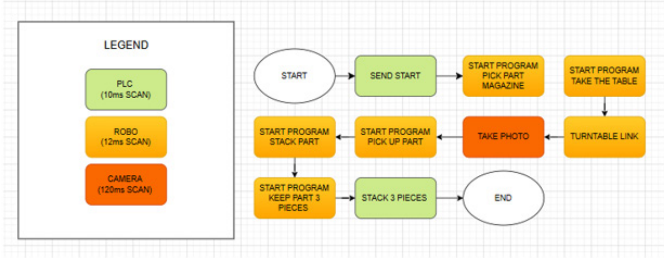

Figure 8 Flowchart illustrating the integration work cycle, showing the device scan times and totaling the integration cycle time at 2 minutes and 28 seconds.

The development of the flowchart was fundamental to the success of the integration. It allowed the requirements, necessary steps, and the best way to perform each task to be clearly identified. By incorporating the flowchart into the process, it was possible to understand how the PLC and the robot—each with its own priorities and communication rates—would interact efficiently.

The following table illustrates the level of difficulty in each stage:

Integration aspect	Difficulty level	Justification
Communication between PLC, robot, and camera	High	Use of the Profinet protocol with the PLC as master, ensuring synchronization and efficient communication.
Integration of multiple languages	High	Relevant challenge in handling different languages (Ladder, KRL, and vision system), requiring adaptations.
Structuring the logical flow	Medium	Detailed flowcharts allowed organization of steps and anticipation of conflicts, promoting smooth integration.
Organization and planning	Medium	Thorough planning and precise variable mapping were essential for project success.
Overall integration difficulty	High	High complexity due to technological diversity, overcome with a structured and effective methodology.

Discussion

The successful integration of industrial systems relies heavily on precise variable mapping, beginning with the PLC (Programmable Logic Controller), which serves as the master and must contain all necessary identifiers to coordinate the other devices. Once this foundation is established, it's crucial to define the priority among subordinate components, such as the robot and the camera, according to the specific process requirements. This hierarchical setup ensures that each element operates within its designated role, promoting clarity and control. Such an organized structure enables efficient, synchronized communication and minimizes conflicts, which in turn reduces failures and optimizes execution time. By clearly assigning roles and priorities, the system avoids command overlaps and data inconsistencies critical factors in automated industrial environments.

This also simplifies maintenance and enhances scalability, making the system more robust and adaptable.[10]

Moreover, the developed integration architecture lays a strong groundwork for implementing emerging technologies like cloud-based PLC control and machine learning algorithms for predictive maintenance and adaptive optimization. These innovations align with the evolving demands of Industry 4.0, empowering industrial systems to become smarter, more autonomous, and resilient in the face of modern challenges.

Conclusion

Despite the challenges encountered, the project proved to be extremely effective, demonstrating the importance of careful planning and organization. The integration of different programming languages and logic, which at first seemed like an obstacle, taught us to value the clarity and sequencing of each process step. Through this experience, it became evident that the key to overcoming complexity is not only technical expertise but also the ability to structure and organize variables in a logical and sequential manner.

The guidance process and the creation of the process flowchart were crucial to the project's success. This method allowed us to visualize and understand the interaction among all components, from the PLC to the robot. By mapping each step and each communication, we were able not only to identify critical points but also to ensure that different priorities and communication rates were respected. This organized approach transformed what could have been a chaotic system into a cohesive and predictable workflow.

In summary, this project not only achieved its technical objectives but also provided valuable learning about managing complex systems. The effectiveness demonstrated in the final result is proof that, even in the face of technical challenges and multiple variables, a structured approach and a deep understanding of communication logic are the most powerful tools for problem solving.

Acknowledgements

The authors would like to thank the Advanced Institute of Robotics – IAR for the support provided during the development of this work. They also extend their gratitude to their colleagues who assisted in the tests and practical validation of the robotic cell.

Conflicts of interest

The authors declare that there are no conflicts of interest related to the publication of this research.

References

1. RAP Vitalli. Digital twin and virtual commissioning of FANUC robotic cells based on the industry 4.0 Context. *Advanced Journal of AI & Robotics*. 2021;2(2):04–10.
2. Ciuccio RL, Silva VJ, Vitalli RAP. Comparison of modified D-H Notation with standard D-H for direct kinematics of industrial robotic manipulators. *Engineering and Technology Journal*. 2022;07(07):1419–1421.
3. Vitalli RAP, Rosário JM. Digital twin of ABB IRB-120 robotic cell based on the industry 4.0. *International Robotics & Automation Journal.* 2022;8(3):89–92.
4. Vitalli RAP. Virtual commissioning and digital twin of yaskawa motoman robotic cells based on the industry 4.0 context. *International Robotics & Automation Journal.* 2021;7(2):35–38.
5. Ricardo José Marques da Silva, Jeferson Nicacio de Mendonça, Vitor Araujo Francisco, et al. Determination of the inverse kinematic of the industrial robot yaskawa-motoman-Gp7 through analytical and recursive calculation. *Advances in Robotics & Mechanical Engineering*. 2022;3(5).
6. Vitalli RAP, Arnaldo Ortiz Clemente, Rosário João Mauricio. Industrial robots integration based in object location and component inspection in industry 4.0 using vision guided robot (VGR). *International Robotics & Automation Journal*. 2022;8(1):1–7.
7. Frederico Garavati Fioretti, Paulo Bueno Santos, Marcelo Campos Rebouças, et al., A comparative study of time-related performance: COBOT with industrial robots using digital twin robotic cells. *International Robotics & Automation Journal.* 2024;10(3):97–101.
8. Vitalli RAP. Digital Twin and virtual commissioning of robotic cells based on the industry 4.0 context. *International Journal of Science Academic Research.* 2021;02(02):1149–1151.
9. Vitalli RAP, Moreira JML. Mathematical modeling of digital twins and industry 4.0 technologies for measuring pipe wall thickness at Angra 1 nuclear power plant. *Journal of Artificial Intelligence, Machine Learning and Data Science*. 2023;1(2):72–77.
10. Igor Manoel Garbim Cruz, Kennedy Robert Sotana Bertoncelli, Tiago de Aguiar Souza, et al. Symbolic and numerical mathematical modeling of the Jacobian of the Yaskawa Motoman - GP7 robot using Matlab©. *Journal of Robotics and Automation Research.* 2022;3(2):227–234.

Citation: Cruz MHS, Santos PB, Rebouças MC, et al. Integration of a vision-guided robotic cell using PLC as process master. *Int Rob Auto J.* 2025;11(3):88–92. DOI: 10.15406/iratj.2025.11.00305