AUTOMATIC PRESSURE CONTROL SYSTEM

Schifîrneț Petre-Iustin

Petroleum-Gas University of Ploiești
Faculty of Mechanical and Electrical Engineering – Applied Informatics and Automation

Introduction

This project represents the development of a pressure regulation system using Rockwell Automation software and hardware. Unlike previous simulations performed entirely in software, this version included direct interaction with a physical PLC device through Ethernet/IP and RSLinks. It was a crucial step closer to real industrial implementation, involving live data from a sensor and integration with an HMI interface. Although the system did not drive a real actuator, the control logic, communication, and real-time behavior mirrored that of an operational industrial process.

The aim of the project was both functional and educational: to design a logic system capable of monitoring pressure from a real input module and responding appropriately according to a reference value. The actual values were acquired from an analog input module and transformed into engineering units for interpretation. Throughout the process, the experience of downloading code to the PLC, establishing IP communication, and linking controller tags to an HMI interface helped cement the practical knowledge required in real-world automation.

Project Description

The core logic of the project was designed using ladder diagram in Studio 5000. The system is activated based on the combination of START and STOP inputs. Once activated, a tag called Sistem_Reglare_ON controls the operational state. A periodic timer ensures that sensor data is read at regular intervals. A MOV instruction transfers raw analog data into a DINT variable, which is then processed by a CPT block that converts it into real engineering units. The value obtained is compared with a reference setpoint, and if the pressure is below the reference, a tag Compressor_ON is triggered.

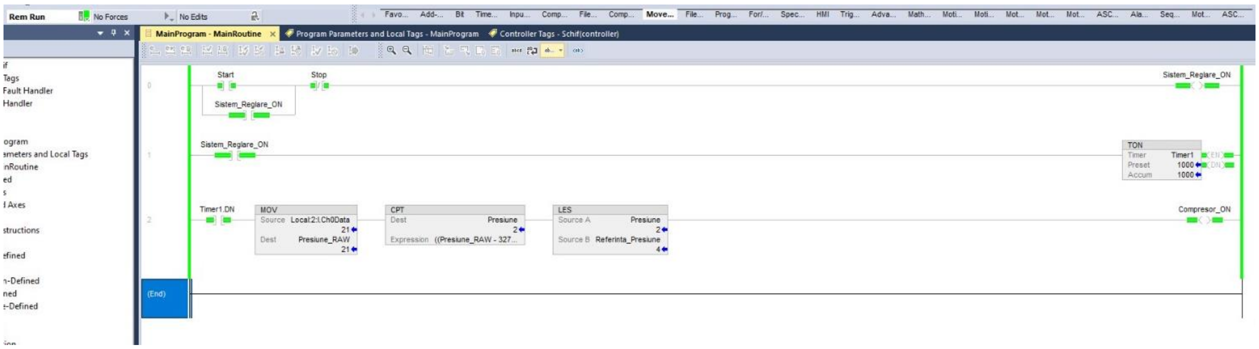


Figure 1 – Ladder logic (when pressure is below reference)

Although the compressor is not physically implemented, the logic that drives its activation is realistic and scalable. It simulates how a control system would operate in practice, using real-time sensor feedback. The comparison is done with a LES instruction, maintaining a clean and modular structure throughout the logic.

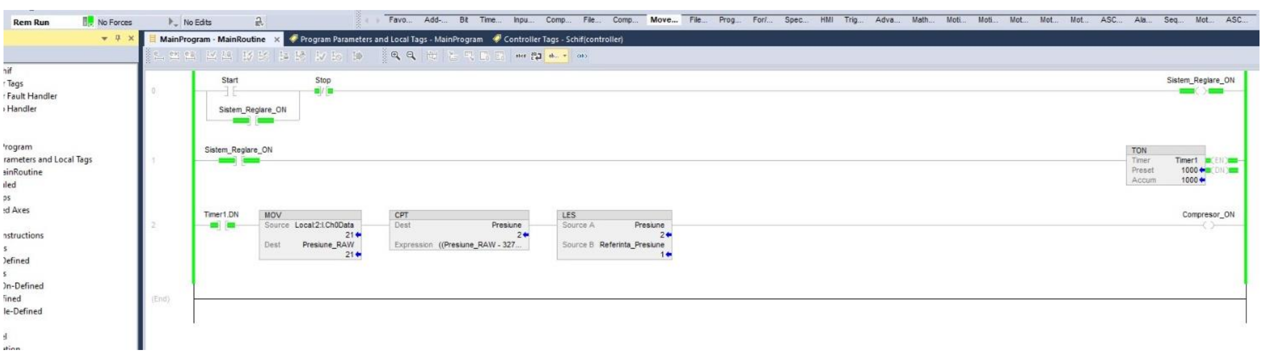
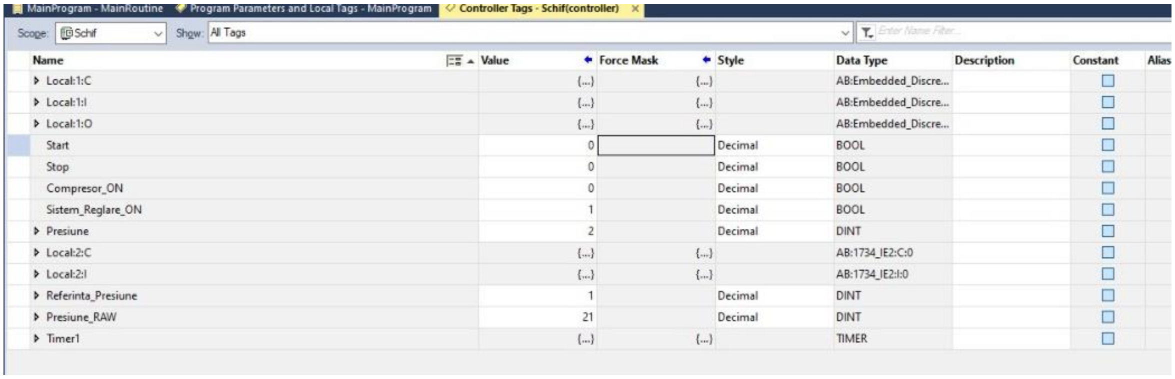


Figure 2 – Ladder logic (when pressure is above reference)

The tags used in this project were structured in a straightforward manner. Boolean types (BOOL) were used for system commands such as START, STOP, and ON/OFF indicators. Integer types (DINT) handled raw sensor input, while the actual pressure in engineering units was stored in a REAL. A TIMER structure was used for periodic execution, emulating a basic polling mechanism.



MainProgram - MainRoutine | Program Parameters and Local Tags - MainProgram | Controller Tags - Schif(controller)

Scope: Schif   Show: All Tags

Name	Value	Force Mask	Style	Data Type	Description	Constant	Alias
Local:1:C	{...}	{...}		AB:Embedded_Discre...			
Local:1:I	{...}	{...}		AB:Embedded_Discre...			
Local:1:O	{...}	{...}		AB:Embedded_Discre...			
Start	0		Decimal	BOOL			
Stop	0		Decimal	BOOL			
Compresor_ON	0		Decimal	BOOL			
Sistem_Reglare_ON	1		Decimal	BOOL			
Presiune	2		Decimal	DINT			
Local:2:C	{...}	{...}		AB:1734_IE2:C:0			
Local:2:I	{...}	{...}		AB:1734_IE2:I:0			
Referinta_Presiune	1		Decimal	DINT			
Presiune_RAW	21		Decimal	DINT			
Timer1	{...}	{...}		TIMER			

Figure 3 – The tags used

HMI Integration

The FactoryTalk View Studio HMI was configured to reflect the live status of the PLC. It consisted of a numeric display showing the current pressure, START and STOP buttons with clear visual feedback, and a button labeled "MODIFY PRESSURE" to allow input of a reference setpoint. Although simple, this interface mimicked an industrial panel and allowed for interaction with the control logic.

The tags used in the PLC were connected through RSLinks Enterprise to the HMI objects. This integration meant that changes in logic or sensor values were reflected in real time on the screen. It also introduced a level of realism in testing and validation, as interaction with the system mimicked that of a production environment.

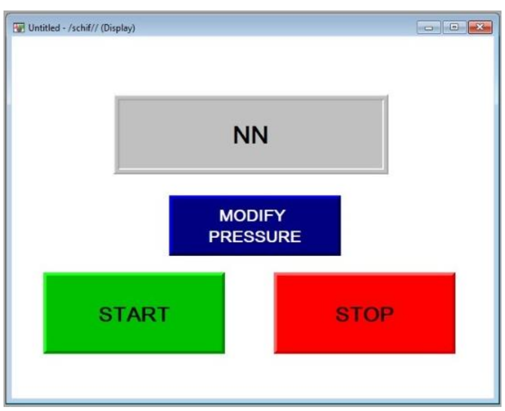


Figure 4 – HMI Interface

Reflections and Experience Gained

Working on this project pushed my understanding of automation from a software-only context to one grounded in real hardware communication and sensor feedback. The process of downloading the ladder logic to the controller, assigning IP addresses, troubleshooting tag values, and synchronizing with the HMI interface added layers of technical and practical insight.

Even though the final system was not connected to an actuator, I was able to simulate the entire data flow and observe how logic responds to changing sensor inputs. I developed a better grasp on the handling of analog signals, specifically the transformation from raw digital readings to usable engineering values. I also understood how precise tag definition, data typing, and structured logic influence system reliability and clarity.

This project marked a turning point in approaching industrial automation—not just from a theoretical standpoint, but from a field-ready perspective. The ability to confidently test, monitor, and interact with both hardware and HMI prepares me for more advanced systems that include actuators, feedback loops, safety layers, and full deployment.