International Journal
Information Technology Applications (ITA)
Volume 14, Number 1, July 2025

LOCAL STORAGE OF DATA FROM THE PRODUCTION PROCESS

Filip Žemla, Ján Cigánek

Abstract:

This paper presents the design and implementation of a local data storage system for process data acquired from a model of a manufacturing line. The line is controlled by a PLC, for which a custom control program was developed. Process data are collected and visualized using the Promotic SCADA system, which also handles data logging. The application enables both monitoring and basic control of the production line in real time. Collected data are stored in structured text files, organized into folders by time or batch, allowing for easy access and analysis. This work provides an overview of key industrial automation technologies, including PLC programming languages, SCADA systems, and data handling methods. Special focus is given to the Promotic environment and its capabilities in educational and small-scale industrial scenarios. The system was tested on a school-scale production model and proved reliable and suitable for pedagogical use. Challenges related to communication and data integrity were addressed and resolved. The final solution is cost-effective, simple to maintain, and serves as a foundation for further development or integration with external systems.

Keywords:

PLC, SCADA, HMI, Local storage, Process data.

Introduction

In modern industrial environments, the role of process data has evolved from being a mere diagnostic tool to a critical asset for decision-making, traceability, and continuous optimization. As the manufacturing sector embraces Industry 4.0, the integration of intelligent control systems, real-time monitoring, and secure data storage becomes a strategic imperative. One of the key technological enablers of this transformation is the Programmable Logic Controller (PLC)—a robust automation element designed for precise control under harsh industrial conditions. Coupled with SCADA (Supervisory Control and Data Acquisition) systems and HMI (Human-Machine Interface) solutions, PLCs enable the real-time interaction between operators and machines while capturing and distributing valuable process data [1].

Despite the increasing trend toward cloud-based analytics and remote data access, many industrial facilities, especially those in safety-critical or isolated environments, still require localized data storage solutions. These are especially relevant for manufacturing lines that operate autonomously or within secure zones where internet access is limited or deliberately restricted. In such cases, the ability to store, organize, and retrieve process data locally ensures reliability, auditability, and long-term stability without the cybersecurity risks associated with cloud connectivity [1, 6, 8].

This paper presents the design and implementation of a local data storage system for a model production line, developed as part of an academic research project. The solution is based on a laboratory-scale automated assembly line controlled by a Siemens S7-1200 PLC. A Promotic-based SCADA application was developed to visualize and manage operations, and simultaneously log critical process events (e.g., product movements, alarms, user logins) into a structured local file system. The proposed architecture emphasizes modularity, role-based access control, and timestamped data logging, offering a practical template for small-scale industrial deployments and educational purposes. The study aims to demonstrate how localized data storage, when properly designed, can meet both operational and security requirements in modern production systems [8, 9].

1 State of the Art

The efficient management of industrial data requires a synergistic integration of multiple technological layers, including hardware controllers, programming standards, visualization tools, and storage systems. In the context of modern manufacturing, data plays a key role not only in diagnostics and control but also in long-term planning, traceability, and system optimization [1, 2].

Modern production lines are commonly classified based on their degree of automation, which directly impacts complexity, scalability, and labor demands. The three main types are (Fig.1), [2]:

- **Manual assembly lines** - operated entirely by human workers. Materials are handled manually, and each step is performed at a workstation. Typical for low-volume, customized production.
- **Semi-automated lines** - combine human labor with robotic or automated subsystems. Some actions (e.g. transport, machining) are automated, while operators still perform key tasks. Offers a good balance between cost and flexibility.
- **Fully automated lines** - operate with minimal or no human intervention. PLCs or control systems govern all actions. These lines are further divided into:
  – Hard automation: used for fixed-cycle mass production.
  – Flexible automation: adaptable for different products and configurations.

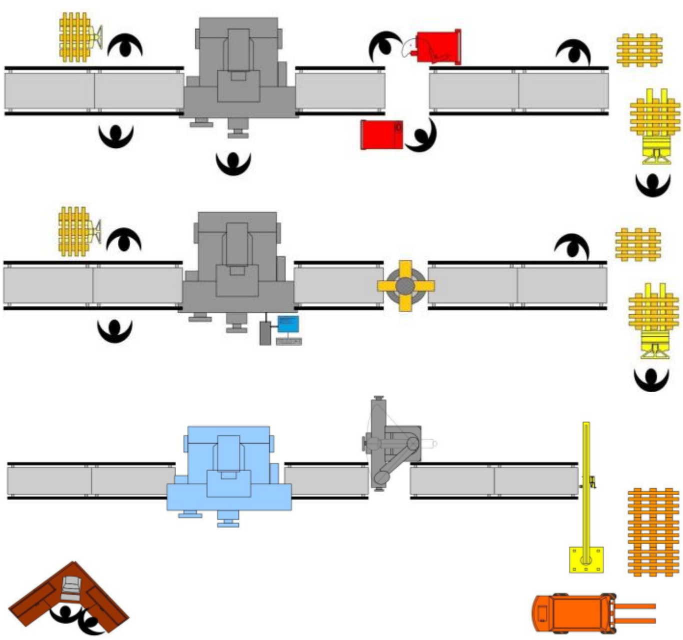

Fig.1. Examples of (up to down): manual, semi-automated, and fully automated assembly lines [2].

At the heart of any automated line lies the **Programmable Logic Controller (PLC)**. PLCs are designed to function reliably in industrial environments, replacing hardwired logic with programmable, reusable control logic. The first PLC, Modicon 084, introduced in 1968, revolutionized industrial automation by enabling digital control of processes [1].

PLCs come in various configurations depending on size and architecture, such as [1]:

- Nano PLCs: < 16 I/O, for simple logic tasks.
- Micro/Mini PLCs: 16–500 I/O, for small production lines.
- Modular or rack-based PLCs: 500+ I/O, used in large-scale industrial systems.

They are also classified by design [1]:

- Compact: fixed number of I/O, integrated in one unit.
- Modular: expandable via I/O and communication modules.
- Rack-mounted: highly scalable, often used in networked systems.

To ensure interoperability and standardized logic, PLC programming is defined by IEC 61131-3, which specifies five primary languages [1]:

- LD (Ladder Diagram) – based on relay logic,
- FBD (Function Block Diagram) – block-oriented,
- IL (Instruction List) – low-level, now deprecated,
- ST (Structured Text) – high-level, similar to Pascal/C,
- SFC (Sequential Function Chart) – process-step driven.

Each one is suited to different control needs. Examples of their syntax and structure are shown in (Fig.2).

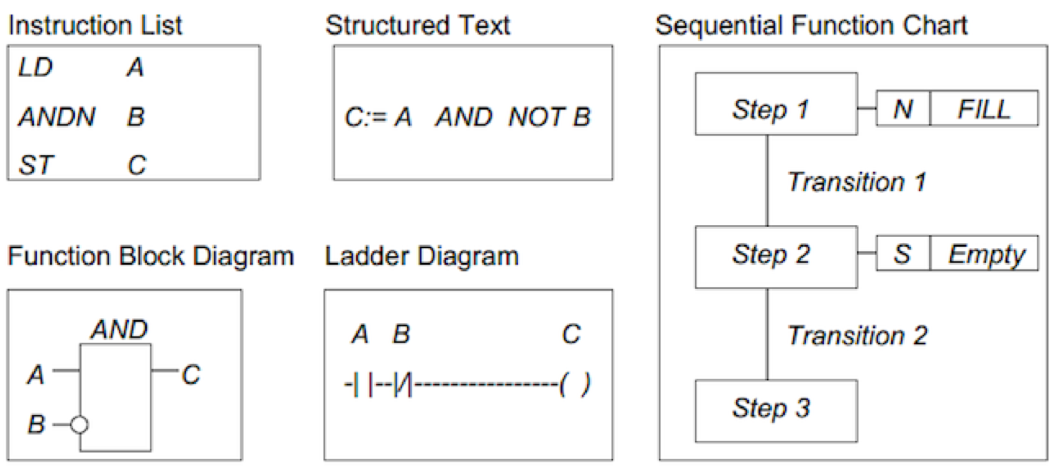

Fig.2. Overview of IEC 61131-3 PLC programming languages [12].

Closely integrated with PLCs are SCADA systems (Supervisory Control and Data Acquisition), which allow real-time visualization and control of automated processes. A SCADA system typically includes [1]:

- Human-Machine Interfaces (HMI).
- Databases or file-based storage.
- Communication interfaces (e.g., OPC-UA, Modbus).

Among commonly used SCADA tools are SIMATIC WinCC, Reliance, and Promotic. In this project, Promotic was selected for its simplicity, licensing flexibility, support for VBScript/JavaScript, and its ability to create and manage local file storage [5].

The Promotic environment transitioned from VBScript to JavaScript in its newer versions. While both are supported, JavaScript is recommended due to broader compatibility. A syntax comparison is provided in (Fig. 3) [5].

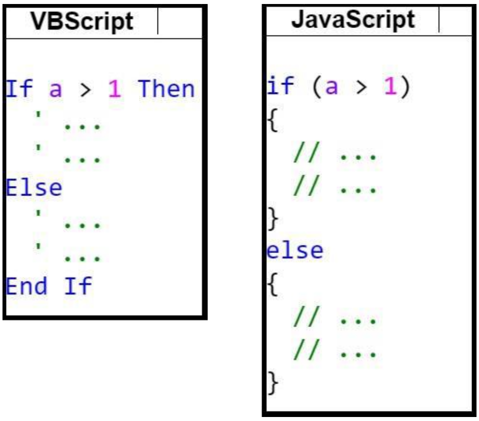

Fig.3. Comparison of IF statements in VBScript (left) and JavaScript (right) [5].

From a design perspective, ergonomic principles are essential. The IEC 73 standard defines best practices for SCADA color schemes and operator response. For instance, red indicates critical alarms, yellow means warnings, and green confirms normal conditions. These are summarized in (Tab. 1) [1].

Table 1. IEC 73 color coding for operator interfaces [1].

Color	**State**	**Operator action**
Red	Alarm	Immediate reaction required
Yellow	Warning	Monitor closely
Green-yellow	Caution	Attention required
Green	Normal	No action needed

Finally, data storage in SCADA environments is handled either locally or externally. Local storage offers security, autonomy, and stability — especially in air-gapped or isolated systems. External (cloud-based) storage enables remote access and scalability but introduces cybersecurity risks. This work focuses on a local file-based storage architecture, in which process data is written as structured text files into a date-based directory tree. This method ensures transparency, offline functionality and long-term maintainability for small-scale industrial or educational setups [1, 5].

2 Design of Research

The practical implementation was carried out using a laboratory-scale production model designed to emulate the workflow of a real manufacturing line. The aim was to develop a fully functional control, visualization, and data logging system using standard industrial components and software, while ensuring modularity and maintainability.

a) Control Algorithm for PLC

The core of the system is a U-shaped assembly line model by Fischertechnik, comprising four conveyor segments, two processing machines, and five photoelectric sensors for position detection. Transport between line segments is handled by two automated shuttles equipped with motion limit switches. All actuators and sensors are connected through a custom driver interface to a Siemens SIMATIC S7-1200 PLC (CPU 1215C DC/DC/DC), which provides 14 digital inputs and 10 digital outputs, sufficient for the requirements of the experiment (Fig. 4).

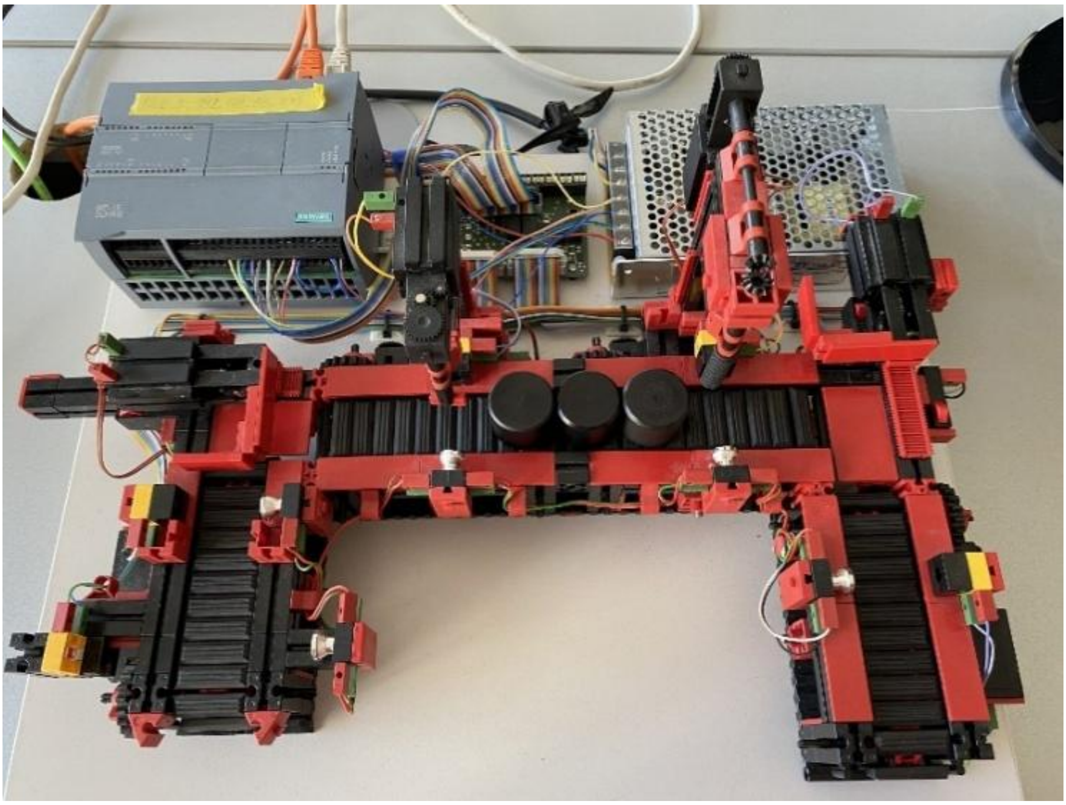

Fig.4. Laboratory U-shaped production line
connected to Siemens S7-1200 PLC via driver interface.

The PLC was programmed using **TIA Portal v17**, utilizing the **FBD (Function Block Diagram)** language due to its intuitive graphical structure and support for real-time logic.

The logic allows two control modes:

- **Automatic mode**, where the PLC executes the entire control logic based on sensor states and time triggers.
- **Manual mode**, which allows authorized users to override machine control via the SCADA interface.

The control logic handles:

- Detection of part positions on the line,
- Operation timing for machine cycles (5s for machine 1, 7s for machine 2),
- Shuttle movement between line sections,
- Counting of finished products and parts currently on the line,
- Triggering and acknowledgment of alarms.

Control mode switching and functional flow of part movement are defined within a hierarchical logic tree (Fig. 5).

b) Human Machine Interface

On top of the PLC logic, a Promotic 8.3.32 SCADA system was developed to visualize and interact with the line. A free license (supporting up to 100 variables and one-hour runtime) was used for development. The SCADA application includes:

- Login panel with user authentication,
- Role-based access levels (monitoring-only, operator, admin),
- Dynamic visualization of line status, material flow, and alarms,
- Manual override panel (admin only),
- Real-time counters and system reset features.

Depending on user level, the main visualization screen (MainPanel) provides different functionalities. The screen layout includes component status (e.g., motor/semaphore color), part counters, alarm state, and manual control buttons (Fig. 6).

Login data, permissions, and authentication logic are handled through an embedded Microsoft Access database. All login attempts, status changes, and alarm confirmations are logged with timestamps and user identifiers. Data generated during the operation of the line are stored in a structured folder hierarchy on the local disk. Each day, a new folder is created and organized by year, month, and date. Within this folder, logs are grouped into two categories:

- Aggregated logs (e.g., system-wide events)
- Product-specific logs (each manufactured part has its own file)

The folder for a given date contains:

- Alarms.txt: a list of all alarm events with timestamps, status, and responsible user.
- Login.txt: a log of all user login, logout, and session change events.
- Material.txt: a summary of finished parts with timestamps of completion.
- ProcessData/: a subfolder where each finished unit receives an individual log file named Material-<ID>.txt.

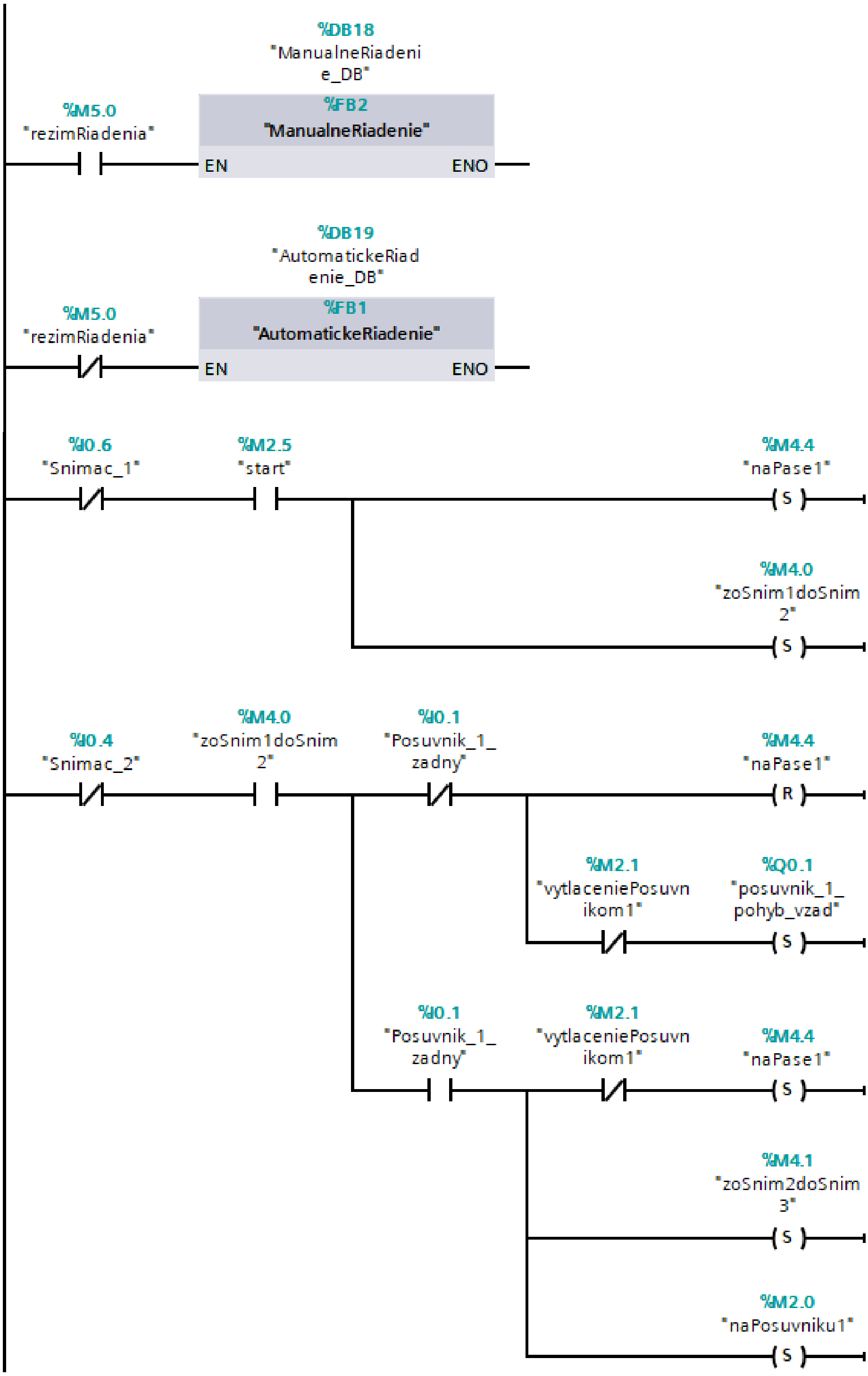

Fig.5. Simplified control logic: mode selection and part handling logic in automatic mode.

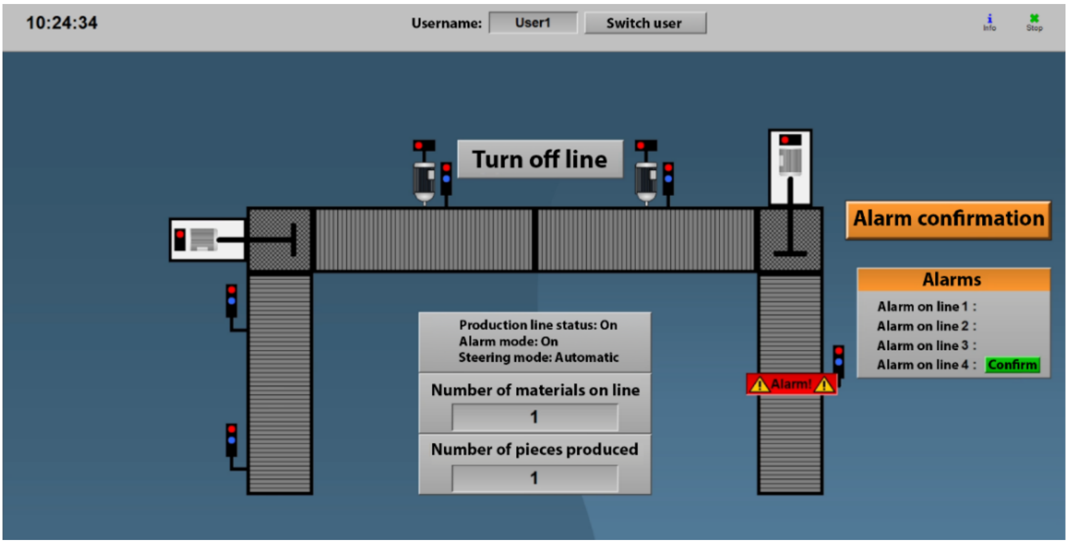

Fig.6. SCADA MainPanel for user level 2 (admin): full system visibility and manual control access.

A sample folder structure is shown in (Fig.7), demonstrating how daily operational data is clearly separated and archived.

```
DatabaseProductionLine
  └── 2024
        └── 5
              └── 2024-05-09
                    ├── Alarms.txt
                    ├── Login.txt
                    ├── Material.txt
                    └── ProcessData
```

Fig.7. Example of folder structure for a specific production day.

The file *Login.txt* captures every authentication-related action, including successful logins, incorrect credentials, user switching, and system shutdowns. All entries are timestamped and labeled by user level (Fig. 8).

Time	- Status	- Name	- Level
05/09/2024 17:01:57	- Login	- User2	- 0
05/09/2024 17:02:03	- Logout		
05/09/2024 17:02:09	- Login	- User1	- 1
05/09/2024 17:02:19	- Logout		
05/09/2024 17:02:26	- Login	- User2	- 0
05/09/2024 17:02:44	- Logout		
05/09/2024 17:02:53	- Login	- User1	- 1
05/09/2024 17:05:59	- Logout		
05/09/2024 17:06:06	- Login	- Admin	- 2
05/09/2024 17:10:24	- Application shutdown and logout		
05/09/2024 17:19:37	- Login	- Admin	- 2

Fig.8. Sample output from *Login.txt* showing login events and user activity.

Alarm handling is also logged continuously. The *Alarms.txt* file records activation, confirmation, and deactivation of each alarm. These logs include alarm name, severity level, status, and the operator who responded. Additionally, *Materials.txt* stores time-marked entries for each completed product (Fig. 9).

Each individual product log contains a full timeline of the part's journey across the production line. This includes:

- Entry and exit timestamps at each sensor.
- Machine start/end cycles.
- Processing durations.
- Final exit from the system.

An example of such a product log is shown in (Fig. 10), illustrating complete traceability from start to finish.

This logging structure provides not only transparency and reproducibility but also makes it possible to perform offline diagnostics and long-term operational analysis without relying on an external database or internet connection.

Time	- Alarm	- Status	- Level	- User
05/09/2024 17:02:38	- Alarm Zone 4	- Active		
05/09/2024 17:02:57	- Alarm Zone 2	- Acknowledged	- 1	- User1
05/09/2024 17:05:22	- Alarm Zone 2	- Active		
05/09/2024 17:05:26	- Alarm Zone 2	- Acknowledged	- 1	- User1
05/09/2024 17:05:36	- Alarm Zone 3	- Active		
05/09/2024 17:05:39	- Alarm Zone 3	- Acknowledged	- 1	- User1
05/09/2024 17:06:16	- Alarms Off			
05/09/2024 17:06:32	- Alarms On			
05/09/2024 17:06:42	- Alarms Off			
05/09/2024 17:06:56	- Alarms On			
05/09/2024 17:07:00	- Alarm Zone 4	- Active		
05/09/2024 17:07:06	- Alarm Zone 4	- Acknowledged	- 2	- Admin

Time	- Number of completed materials
05/09/2024 17:02:57	- Material - 1
05/09/2024 17:05:11	- Material - 2
05/09/2024 17:05:52	- Material - 3
05/09/2024 17:07:06	- Material - 4
05/09/2024 17:07:40	- Material - 5
05/09/2024 17:07:49	- Material - 6
05/09/2024 17:07:59	- Material - 7
05/09/2024 17:08:08	- Material - 8
05/09/2024 17:08:17	- Material - 9
05/09/2024 17:09:00	- Material - 10
05/09/2024 17:20:07	- Material - 11
05/09/2024 17:20:16	- Material - 12
05/09/2024 17:20:25	- Material - 13
05/09/2024 17:20:34	- Material - 14
05/09/2024 17:20:43	- Material - 15

Fig.9. Sample outputs from *Alarms.txt* (top) and *Materials.txt* (bottom).

```
05/09/2024 17:07:17 - Material entered the production process
05/09/2024 17:07:19 - Material arrived at sensor 2
05/09/2024 17:07:29 - Material arrived at sensor 3
05/09/2024 17:07:34 - Machine 1 started
05/09/2024 17:07:37 - Material arrived at sensor 4
05/09/2024 17:07:44 - Machine 2 started
05/09/2024 17:07:49 - Material left the production process
```

Fig.10. Log file for a specific product (*Material-007.txt*) with event history.

3 Discussion of Results and Experiences

The implemented system successfully demonstrated that a combination of industrial-grade control hardware and lightweight SCADA software can be used to design a robust local data storage solution tailored for small production environments or educational settings. Throughout the development and testing process, several key insights were gained regarding both technical functionality and user experience.

From a control standpoint, the use of a Siemens S7-1200 PLC provided sufficient flexibility and reliability for managing a multi-segment production model. The real-time control of conveyors, shuttles, and processing machines worked smoothly in both manual and automatic modes. The system correctly handled concurrent part movements, accurately counted completed units, and responded to unexpected states by triggering alarms. The use of FBD programming proved intuitive and well-suited for this application, enabling efficient troubleshooting and future extensibility.

The Promotic SCADA application played a crucial role in visualization and user interaction. Thanks to its built-in support for scripting, UI design, and data handling, it was possible to develop a fully featured interface without requiring an external database or third-party software. The login system with three user roles ensured appropriate access control and allowed different levels of operational authority, from basic monitoring to full manual override and reset functionality. User feedback indicated that the layout was clear, and the use of color-coded indicators (e.g., semaphores, alarm states) improved usability and situational awareness.

One of the most significant outcomes was the implementation of local file-based data storage, which proved to be both transparent and reliable. The system created structured folders and logs automatically without operator intervention. This approach enabled:

- Traceability of each product (with individual logs for each unit).
- Auditability of system events (e.g., alarms, user logins).
- Offline diagnostics without the need for network access.

Although the use of plain text files may not be suitable for large-scale data analytics or integration with cloud-based ERP systems, it offers several advantages for controlled environments, including air-gapped facilities and student laboratories. Logs could be easily reviewed using basic tools (e.g., text editors or Excel), and the folder structure remained human-readable and well-organized.

During testing, the most challenging aspect was the need to track multiple products simultaneously and assign the correct data to the corresponding product log. This was solved by incrementing unique product IDs based on sensor edge detection and completion counters. The solution worked reliably even in cases where multiple parts were processed in parallel.

Performance-wise, the application was responsive, and the logging mechanism had no observable impact on real-time control, even under frequent data writing intervals (0.1–0.5 s).

The system also responded well to simulated faults and edge cases (e.g., blocked sensors, alarm conditions), confirming the robustness of the control logic and the flexibility of the visualization layer.

In summary, the project confirmed that low-cost SCADA and PLC integration with structured local storage can be used to develop a scalable, secure, and maintainable solution for data collection in automated manufacturing systems. It also offers a useful reference architecture for further expansion into hybrid systems combining local and remote storage or more complex process analytics.

Conclusion

This work presented the design, implementation, and evaluation of a local data storage system for an automated laboratory-scale production line. The project combined industry-relevant technologies—namely: Siemens PLCs, Promotic SCADA software and a custom file-based logging architecture—to create a fully autonomous and offline-capable system for control, monitoring and data archiving.

The solution was built to reflect real-world industrial workflows while maintaining modularity and transparency. The control logic allowed seamless switching between automatic and manual operation, supported alarm handling, and enabled real-time tracking of material flow across the line. The visualization layer, developed in Promotic, provided a clear, role-based interface with differentiated access for operators and administrators.

One of the most important contributions of this project is the implementation of structured local storage that requires no internet connection, external database, or third-party services. All relevant process data—product movements, system alarms, and user actions—are logged in human-readable files, organized by date and type. This not only ensures long-term maintainability but also addresses key cybersecurity concerns by keeping the system fully offline and self-contained.

The solution was tested extensively and proved to be reliable, scalable within its scope, and user-friendly. It offers a viable reference model for:

- Educational labs simulating real production environments.
- Small-scale industrial operations requiring traceability without cloud dependencies.
- Isolated or secure facilities where external data transmission is undesirable.

All project objectives were met, and the system behaved as expected under both normal and fault conditions. Furthermore, the experience gained during development suggests several promising directions for future research, including:

- Integration with external databases or web-based dashboards.
- Migration from flat-file storage to structured formats (e.g., JSON, SQLite).
- Implementation of remote diagnostics and predictive analytics.

In conclusion, the project demonstrates that it is feasible to build a compact and secure data acquisition platform using readily available automation tools and simple design principles. Such systems can contribute to increased transparency, process understanding, and educational value in the field of modern industrial automation.

Acknowledgement

This research was funded by the Slovak Agency for Research and Development - grant No. APVV-21-0125, by the Cultural and Educational Grant Agency of the Ministry of Education, Research, Development and Youth of the Slovak Republic KEGA 021STU-4/2024, and by the Scientific Grant Agency of the Ministry of Education, Research and Sport of the Slovak Republic No. 1/0637/23 and 1/0045/25.

References

[1] Cigánek, J., Žemla, F.: PLC systémy v mechatronike. (*PLC systems in mechatronics*) 2023. ISBN 978-80-8208-097-4.

[2] Kuždák, V.: Optimalizácia výrobnej linky ako súčasť výrobného procesu. (*Production line optimization as part of the production process*) [online]. https://www.sjf.tuke.sk/umpadi/taipvpp/2008/index.files/Priemyselne_inzinierstvo/kuzdak.pdf

[3] Duchoň, F. Babinec, A. Dekanová, M. Mudrák, M. Korec, S.: Metodológia na testovanie nábehu výrobných liniek (1). (*Methodology for testing the start-up of production lines*) [online]. https://www.atpjournal.sk/novetrendy/metodologia-na-testovanie-nabehu-vyrobnych-liniek-1.html?page_id=27597

[4] BPS - Bratislavský parkovací systém. Parkovanie v Bratislave. (*Bratislava parking system. Parking in Bratislava*) [online], January 15, 2025, https://paas.sk

[5] Promotic. Promotic [online]. https://www.promotic.eu/cz/

[6] Hegedűs, T., Varga, I. and Abonyi, J.: Development of a SCADA system using open-source technologies. Periodica Polytechnica Electrical Engineering and Computer Science, vol. 63, no. 1, pp. 14–21, 2019. DOI: 10.3311/PPee.13279

[7] Leitão, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T. and Colombo, A.W.: Smart agents in industrial cyber–physical systems. Proceedings of the IEEE, vol. 104, no. 5, pp. 1086–1101, 2016. DOI: 10.1109/JPROC.2016.2521931

[8] Wollschlaeger, M., Sauter, T. and Jasperneite, J.: The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. IEEE Industrial Electronics Magazine, vol. 11, no. 1, pp. 17–27, 2017. DOI: 10.1109/MIE.2017.2649104

[9] El-Halwagi, M.E.: Design of a SCADA System for Real-Time Industrial Process Monitoring and Control. International Journal of Computer Applications, vol. 182, no. 4, pp. 21–26, 2018.

[10] Swapan, B.: Plant intelligent automation and digital transformation. 2022. ISBN 978-0-323-90246-5.

[11] Gérer, A. et al.: Montážna linka sa spája s menom Ford. (*The assembly line is associated with the Ford name*) In: AT&P journal. ISSN 1336-233X, 2003, vol. 9, p. 12-13.

[12] Motion Control Tips a Design World Resource What Are IEC 61131-3 and PLCopen [online]. https://www.motioncontroltips.com/iec-61131-3-plcopen/

Authors

Ing. Filip Žemla, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava, Slovakia
filip.zemla@stuba.sk
Researcher and full-stack developer. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in 2023. His current work focuses on the virtualization and optimization of modern manufacturing processes, with a strong background in SCADA systems, database technologies, back-end development and front-end development. In addition to his academic background, he actively works as a full-stack developer, combining industrial automation with advanced software engineering skills.

Ing. Ján Cigánek, PhD.
Faculty of Electrical Engineering and Information Technology,
Slovak University of Technology in Bratislava,Slovakia
jan.ciganek@stuba.sk
He was born in 1981 in Malacky, Slovakia. He received the diploma and PhD. degree in Automatic Control from the Faculty of Electrical Engineering and Information Technology, Slovak University of Technology (FEI STU) in Bratislava, in 2005 and 2010, respectively. He is now Assistant Professor at Institute of Automotive Mechatronics FEI STU in Bratislava. His research interests include optimization, robust control design, computational tools, SCADA systems, big data, and hybrid systems.