

Control Technology ACMA 6001

PLC Control – Developments & Functional Specifications

BEng Manufacturing Systems

Chris Beddoe (2114310)

Date: 16th September 2024

ACMA6001 Control Technology

PLC Control – Developments & Functional Specifications

Chris Beddoe

University of Wales Trinity Saint David

16th September 2024

Executive Summary

This report examines the development and impact of Programmable Logic Controllers (PLCs) in industrial automation and the creation of Functional Design Specifications (FDS) for PLC control systems. It starts with by outlining the origins of PLCs in the late 1960s, highlighting their robustness and flexibility compared to traditional relay systems. Is explores how PLCs improved operational efficiency across various sectors, including automotive manufacturing, benefitting from advancements in technology that enable them to handle complex tasks and enhance communication protocols.

The report also outlines essential stages in developing FDS documentation, including system assessment, requirement specification, hardware selection, software design, integration, and testing. It discusses various models, such as the V-Model and Waterfall Model, along with critical factors influencing the effectiveness of FDS, including clearly defined requirements and safety considerations.

Real-world examples demonstrate best practices and challenges in FDS development, underscoring the importance of engaging stakeholders and implementing effective change management to prevent scope creep and address integration issues with legacy systems. In conclusion, a well-structured FDS is crucial for successfully implementing PLC control systems, ensuring efficient, reliable, and scalable automation solutions that meet operational needs.

Table of Contents

Executive Summary	1
1. Introduction	1
2. Task 1	1
2.1 Introduction to PLCs	1
2.2 Historical Context.....	2
2.3 Technological Advancements	2
2.4 Industrial Impact of PLCs	3
3. Task 2	4
3.1 Introduction to FDS	4
3.2 Commonly used models.....	4
3.2.1 V- Model	4
3.2.2 Waterfall Model.....	5
3.2.3 Iterative Model	5
4. Essential Stages in Developing a Functional Design Specification	6
4.1 Initial System Assessment	6
4.2 Requirement Specification	6
4.3 Hardware Selection.....	7
4.4 Software Design and Programming	7
4.5 Integration with Supervisory Systems	7
4.6 Testing and Validation.....	8
4.7 Documentation and Handover	8
5. Critical Factors in FDS Documentation	8
5.1 System Requirements.....	8
5.2 Hardware and Network Topology.....	9
5.3 Software Design and Interfaces	9
5.4 Testing and Commissioning Plans.....	9
5.5 Safety Considerations	9
5.6 Maintenance and Support.....	10
6. Real World Examples and Critical Analysis.....	10
7. Best Practices and Common Challenges	11
8. Conclusion	11
7.0 Bibliography	13
8.0 Appendix A.....	15

1. Introduction

Programmable Logic Controllers (PLCs) have been present in industrial automation for many years, offering reliable and scalable solutions to complex processes. Their evolution has been pushed by the growing demands of industries seeking more efficient, precise, and adaptable control systems. The following assignment explores two key aspects of PLC technology. The first task will provide a literature review on the historical evolution of PLCs, examining their origins, major technological advancements, and their lasting impact on industrial practices. The second task will delve into the development of Functional Design Specifications (FDS) for PLC control systems, analysing the critical stages involved and the key factors that contribute to a robust and effective system design. Through case studies and real-world examples, this analysis aims to uncover best practices and challenges in PLC system development.

2. Task 1

2.1 Introduction to PLCs

PLCs are specialised digital computers designed for the control and automation of industrial processes, such as machinery operation, assembly lines, and robotic devices. Unlike general-purpose computers, PLCs are built to withstand harsh industrial environments, featuring rugged construction, resistance to extreme temperatures, and immunity to electrical noise. PLCs work by continuously monitoring inputs and making decisions based on programmed instructions to control outputs, thus ensuring smooth and precise automation of processes (Bolton, 2015). The primary advantage of PLCs lies in their flexibility, enabling engineers to modify control systems through software changes without the need for extensive rewiring. Over the years, PLCs have become an integral part of industrial automation, replacing traditional relay-based systems due to their ability to handle complex tasks with greater reliability and efficiency (Regh & Sartori, 2008). Their widespread use across industries such as manufacturing, automotive, and energy reflects their essential role in modern production systems.

2.2 Historical Context

The development of PLCs can be traced back to the late 1960s. During this time industrial automation faced significant challenges due to the limitations of traditional relay-based control systems. These systems were time consuming, difficult to modify, and prone to frequent breakdowns. The automotive industry, particularly in the United States, was at the forefront of seeking a solution to these inefficiencies. General Motors (GM), a major player in the automotive sector, needed a more reliable and flexible control system to manage its assembly lines, which were growing increasingly complex. This demand led to the invention of the first PLC by Dick Morley and his team at Bedford Associates in 1968. It was known as the MODICON 084 and was the first commercially available PLC (Bolton, 2015). The original specifications that were required for the PLC included 16 inputs, 16 outputs, and 1 kilobyte of memory space (Peterson, 2022).

The introduction of PLCs revolutionised industrial automation by replacing hard-wired relay logic with programmable digital systems, offering flexibility, scalability, and reduced downtime during reconfiguration (Regh & Sartori, 2008). As industries became more sophisticated in the 1970s and 1980s, the adoption of PLCs expanded across various sectors, including manufacturing, food processing, and chemical production. The growing demand for efficient control systems, coupled with advancements in microprocessor technology, further propelled the development of more powerful and versatile PLCs, solidifying their role as an essential tool in industrial automation (Petruszella, 2010).

2.3 Technological Advancements

Since their introduction, PLCs have undergone significant technological advancements, both in hardware and software, which have greatly enhanced their capabilities and efficiency in industrial control systems. Early PLCs were limited by rudimentary processing power and basic functionality, primarily focusing on binary input/output operations (Bolton, 2015). However, advancements in microprocessor technology during the 1980s and 1990s marked a turning point. These innovations

allowed PLCs to handle more complex tasks such as analog signal processing, enhanced memory capacity, and faster response times, enabling their integration into more complex industrial processes (Regh & Sartori, 2008).

In terms of hardware, modern PLCs now feature powerful processors, expanded input/output modules, and built-in networking capabilities, which support the integration of multiple PLCs into larger control systems (Petruzella, 2010). These developments have been complemented by improvements in communication protocols, such as Ethernet/IP and Profinet which enable real-time data exchange and greater integration between PLCs and other industrial devices.

On the software front, advancements in programming environments have made PLCs more accessible and versatile. The introduction of standardised programming languages, particularly those defined by the IEC 61131-3 standard, such as ladder logic, function block diagrams, and structured text, has allowed engineers to program more sophisticated control algorithms (Bolton, 2015). Additionally, modern PLC software supports simulation and debugging tools, which have further enhanced the efficiency and reliability of industrial automation systems.

2.4 Industrial Impact of PLCs

The advent of PLCs has had a transformative impact on industrial operations, fundamentally altering manufacturing processes, productivity, and industry standards. One of the most significant changes brought about by PLC technology is the shift from manual and relay-based systems to automated, programmable systems. This transition has allowed manufacturers to streamline operations, reduce human error, and improve overall efficiency. The flexibility of PLCs enables rapid reconfiguration of production lines, which is particularly beneficial in industries like automotive and electronics where product variations are frequent (Rehg & Sartori, 2009).

PLCs have also played a critical role in enhancing productivity by reducing downtime and maintenance needs. The use of diagnostic tools embedded within PLC systems allows for real-time monitoring and predictive maintenance, minimising unplanned

stoppages and increasing equipment lifespan (Bolton, 2015). Their ability to handle both discrete and continuous processes has made PLCs versatile across sectors, including food and beverage, pharmaceuticals, and chemical processing.

In terms of industry standards, the widespread adoption of PLCs has driven the development of global automation standards such as the IEC 61131-3, which governs the programming languages used in industrial control systems. These standards have facilitated greater interoperability between different systems and vendors, reducing costs and improving scalability (Petruszella, 2010). As industrial practices evolve towards smart manufacturing and Industry 4.0, PLCs continue to be integral in enabling communication and data exchange between machines, enhancing operational transparency and efficiency.

3. Task 2

3.1 Introduction to FDS

A Functional Design Specification (FDS) is a development process used to define the operational requirements and functional aspects of software or a control system. An FDS effectively serves as a blueprint for system development, programming and testing. There are some existing models that FDS can follow including the V Model, Waterfall Model and others (Rahmany, 2018). Depending on the model used, FDS will involve various stages and considerations. All stages and considerations must be planned to ensure efficient, reliable and scalable system result.

3.2 Commonly used models

3.2.1 V- Model

The V-model is a systems development methodology where the process is divided into two main phases, development and testing. These phases form a "V" shape, with each development stage corresponding to a matching test stage. It emphasises validation and verification at each step.

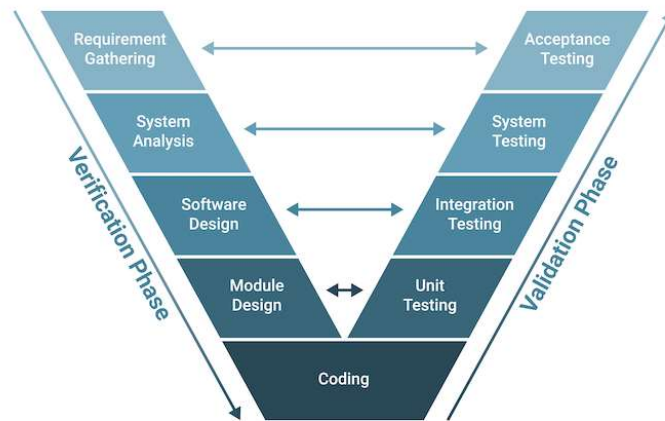


Figure 1 – V-Model methodology (Oppermann, 2023)

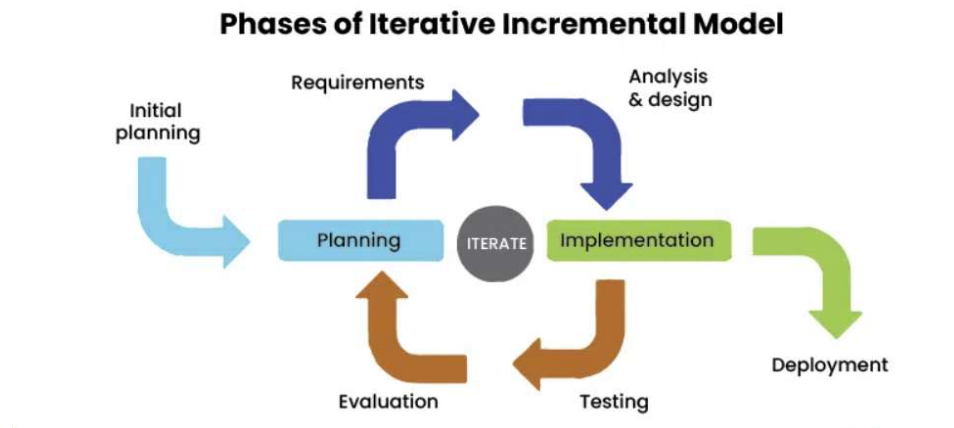
In this model, the left side represents design, and the right side focuses on validation/testing, ensuring that requirements are met throughout. (Sommerville, 2015)

3.2.2 Waterfall Model

The Waterfall model is a linear and sequential approach to FDS development. The five stages are Definition of requirements, Design, Implementation, Verification and Maintenance (Hoory & Bottorff, 2023). Each stage must be completed before the next begins, and there is little to no overlap. This model is simple but lacks flexibility, which can be a challenge when changes arise during development.

3.2.3 Iterative Model

In the Iterative model, FDS development is done cyclically, using the stages of Planning, Design, Implementation, Evaluation and Refinement until a desired system is in place. Each iteration builds upon the previous one, incorporating feedback and making improvements at each cycle.



requirements would also be documented here, which serve as the baseline for system design (Dunn, 2018). For example, specifying that a conveyor system should automatically sort products based on weight and colour would be defined at this stage.

4.3 Hardware Selection

Selecting the appropriate hardware for the PLC system is a critical step. This involves choosing PLC modules, sensors, actuators, and other devices based on system requirements and environmental factors such as temperature, humidity, and potential hazards. Considerations include input and output capacity, processor speed, and communication protocols (Petruszella, 2010). For example, selecting a PLC with robust communication protocols like Ethernet/IP for real-time data monitoring may be selected for use in an automotive manufacturing plant. If there are paint ovens that require certain temperatures, appropriate temperature sensors should be considered that can withstand the temperature within the ovens and give accurate feedback within the desired temperature range.

4.4 Software Design and Programming

At this stage, the control logic and software design are developed. This involves programming the PLC using standard languages (IEC 61131-3) such as Ladder Logic or Structured Text to meet the system requirements. It's essential to map out how inputs will be processed into outputs to control machinery, and this is where the logic of the system is encoded.

4.5 Integration with Supervisory Systems

PLC systems often need to be integrated with Supervisory Control and Data Acquisition (SCADA) systems or other higher-level enterprise systems. The FDS

must include how these systems will interact, ensuring seamless data exchange and control (Boyer, 2009). For example, integrating a PLC system with a SCADA platform for remote monitoring of an oil refinery or chemical production plant.

4.6 Testing and Validation

Testing is a critical stage in ensuring that the PLC control system functions as intended. This includes unit testing of individual components, integration testing of the system as a whole, and user acceptance testing (UAT). The FDS must detail how the system will be validated, including testing scenarios, acceptance criteria, and compliance with safety standards (Regh & Sartori, 2008). An example of testing and validation would be simulating emergency stop conditions in a factory to validate the PLC's safety systems are operating as desired or running some test parts through the system to ensure final product is as desired.

4.7 Documentation and Handover

Comprehensive documentation must be provided to guide operators and maintenance teams on how to use and troubleshoot the system. This documentation typically includes operation manuals, wiring diagrams and PLC logic diagrams. Additionally, the FDS should outline training plans for system users (Dunn, 2018).

5. Critical Factors in FDS Documentation

5.1 System Requirements

Clearly defined system requirements are essential to designing and developing a successful PLC control system. Without a comprehensive outline of both functional and non-functional requirements, the system may not meet operational demands or may fail to comply with the safety and performance standards required (Bolton, 2015).

5.2 Hardware and Network Topology

Hardware selection and network topology are crucial for ensuring the control system's scalability and compatibility with existing infrastructure. Proper planning in this area reduces downtime and facilitates easier integration with other systems (Regh & Sartori, 2008).

5.3 Software Design and Interfaces

The design of control logic and the integration of software interfaces must be thoroughly documented to ensure that the system performs as required. The design of interfaces with SCADA systems or other higher-level systems is particularly important to guarantee efficient data communication and operational control (Boyer, 2009). An example of the PLC interfaces required to be outlined would be type of human-machine interface (HMI) to be used, this could be in the form of a touchscreen or PC workstation. Other interfaces to consider would be input or output devices to be utilised within the system. See Appendix A for an example of a HMI used within industry.

5.4 Testing and Commissioning Plans

Comprehensive testing procedures should be documented in the FDS to ensure that all system functions are validated before handover. The inclusion of Factory Acceptance Testing (FAT) and Site Acceptance Testing (SAT) ensures that the system will operate as intended in real-world conditions (Petruszella, 2010).

5.5 Safety Considerations

Safety features such as emergency stops and interlocks must be documented in the FDS, and it must be ensured that they comply with safety regulations. Standards such as IEC 62061 and ISO 13849-1 offer guidelines on how safety should be integrated into the system design (Bolton, 2015).

5.6 Maintenance and Support

The FDS should include guidelines for long-term maintenance, system updates, and support procedures, which are critical for ensuring the sustainability of the control system (Rehg & Sartori, 2009). Examples of maintenance and support that should be outlined within the FDS include the following: -

- Planned and preventative maintenance
- Corrective maintenance
- Spare parts and inventory management
- Continual software updates and packages
- User training and documentation
- Remote monitoring and support
- End of life planning

All the above are essential to maintaining the reliability, efficiency, and safety of the system. Additionally, the inclusion of remote monitoring capabilities and end of life planning will help organisations respond quickly to issues and improve the system's lifespan.

6. Real World Examples and Critical Analysis

In real world PLC control system projects, best practices in developing FDS documentation can vary depending on the industry. For example, in the chemical industry, safety is paramount, and an FDS in this industry would place a heavy emphasis on interlock systems, emergency stops, and environmental hazard mitigation (Bolton, 2015). In contrast, a food and beverage plant FDS may prioritise the system's sanitation capabilities and frequent line changeover processes.

A hypothetical example that would highlight the challenges of FDS development would be the automation of an older large-scale wastewater treatment facility. One of the main challenges here could be integrating the new PLC system with older, dated systems. For this, the FDS would need to include detailed specifications on how data would flow between the systems and define robust communication protocols to prevent data loss or system downtime.

Another example, this time using the automotive sector summarises the importance of flexibility in FDS documentation. In this instance, an automotive manufacturer may specify a highly adaptable PLC system for its assembly line, which is required to accommodate various models and configurations. The FDS in this case should focus on modular hardware design and software reusability, allowing the company to reconfigure the production line quickly and efficiently as new models are introduced.

7. Best Practices and Common Challenges

Best practices in developing FDS documentation for PLC control systems include involving stakeholders early in the design process, defining clear acceptance criteria, and maintaining comprehensive version control (Petruszella, 2010). One common challenge is scope creep, where additional features are requested after the FDS has been finalised, this would usually lead to delays and budget overruns (Regh & Sartori, 2008). To mitigate this, change management processes should be included in the FDS, detailing how requests for changes will be handled and approved.

Another challenge is the integration of modern PLC systems with legacy equipment. In many industries, existing equipment may not support advanced communication protocols, creating difficulties in achieving seamless data exchange (Boyer, 2009). In such cases, the FDS should include contingency plans, such as using protocol converters or phased migration strategies (Regh & Sartori, 2008).

8. Conclusion

The development of a Functional Design Specification (FDS) is a crucial step in the successful implementation of PLC control systems. It provides a clear roadmap for system design, programming, testing, and commissioning, ensuring that all stakeholder requirements are met. By following best practices such as early stakeholder engagement, detailed requirement documentation, and robust testing protocols, the challenges commonly faced in PLC system development can be mitigated. Additionally, real-world case studies show that flexibility, safety, and

system integration are key considerations that should be addressed in every FDS to improve the overall design and implementation process.

7.0 Bibliography

Bolton, W., 2015. *Programmable Logic Controllers*. 6th ed. s.l.:Newnes.

Boyer, S. A., 2009. *SCADA: Supervisory Control and Data Acquisition*. 4th ed. s.l.:Instrument Society of America.

Dunn, W. C., 2018. *Fundamentals of Industrial Instrumentation and Process Control*. 2nd ed. s.l.:McGraw Hill.

Geeks For Geeks, 2024. *Iterative Incremental Model in Designing System*. [Online] Available at: <https://www.geeksforgeeks.org/iterative-incremental-model-in-designing-system/> [Accessed 20th September 2024].

Hoory, L. & Bottorff, C., 2023. *What Is Waterfall Methodology? And Can It Help Your Project Management Strategy?*. [Online] Available at: <https://www.forbes.com/uk/advisor/business/what-is-waterfall-methodology/#:~:text=the%20Waterfall%20methodology%3F-,There%20are%20five%20phases%20of%20the%20Waterfall%20methodology%3A%20Requirements%2C%20Design,on%20to%20the%20next%20phase.> [Accessed 20 September 2024].

Oppermann, A., 2023. *What Is the V-Model in Software Development?*. [Online] Available at: <https://builtin.com/software-engineering-perspectives/v-model> [Accessed 20 September 2024].

Peterson, D., 2022. *The Origin Story of the PLC*. [Online] Available at: <https://control.com/technical-articles/the-origin-story-of-the-plc/> [Accessed 18 September 2024].

Petruzella, F. D., 2010. *Programmable Logic Controllers*. 4th ed. s.l.:McGraw-Hill Education.

Rahmany, N., 2018. *Agile, V Model, Waterfall, Spiral – a comprehensive guide to the development models*. [Online] Available at: <https://productivehut.com/v-model-agile-waterfall-spiral/> [Accessed 20 September 2024].

Regh, J. A. & Sartori, G. J., 2008. *Programmable Logic Controllers*. 2nd ed. s.l.:Pearson.

RS, 2024. *RS Online*. [Online] Available at: https://uk.rs-online.com/web/p/hmi-displays/8643967?cm_mmc=UK-PLA-DS3A--google--CSS_UK_EN_ePMax_Prio3--8643967&matchtype=&&qad_source=1&qclid=Cj0KCQjwpP63BhDYARIsAOQkATYjDDfWZxH5p4wnmrVKGqNbMaaOyNnyYBi6nkVrYmClnWfcTFhiGcgaAtE4EALw_wcB&qclsrc=aw.ds [Accessed 4 October 2024].

Sommerville, I., 2015. *Software Engineering*. 10th ed. s.l.:Pearson.

8.0 Appendix A

Siemens SIMATIC Series KTP400 Basic HMI Panel - 4.3 in, TFT Display, 480 x 272pixels

RS Stock No.: 864-3967 | Mfr. Part No.: 6AV2123-2DB03-0AX0 | Brand: [Siemens](#)







Price Each

£422.00

(exc. VAT)

£506.40

(inc. VAT)

Units

Select or type quantity

1

+

Add to basket

27 In stock - FREE next working day delivery available*

18 available from Europe for delivery within 2 working day(s).

9 available from Europe for delivery within 2 working day(s).

* Delivery dates may change based on your chosen quantity and delivery address.

FREE delivery for orders over £50.00

View all HMI Displays

Example of HMI used within industry (RS, 2024)