Computing Project (BSc)

Local Control Function Interpreter

Author:	Samuel A. Marti
Place:	Prague College
Module:	Computing project (BSc)
Module Code:	CPJT-600-1901
Project Supervisor:	Bohus Ziskal Ph. D., Alex Moucha Ph. D.
Date:	05.04.2019
Semester Code:	1804 - 1901
Word Count:	10756

Abstract

This paper is mainly concerned with implementation and research of the Industry 4.0 principle "decentralize decision taking" coupled with the clientship of the industrial company Eaton. It ultimately describes the implementation and design of a stack machine based interpreter on a microcontroller which is used for realizing edge computing on an industrial IO module.

Together with the implementation, also research into device-to-device communication and time deterministic behaviour was conducted along with an assessment of the standards IEC61131 and IEC61499, always in respect to the implemented interpreter and Eaton technology used.

The paper presents also the used methods and validates the produced artefacts compared to their requirements.

Acknowledgement

Independence does not exist! Just as independence might be an unattainable dream of mere earth dwellers this work is the humble result of human effort accumulated since the dawn of time. The technology, knowledge and environment shaped by humans, nature and physics are all obvious requisites for the reader to perceive this paper. Thus, it is hoped that this paper might result in a humble enrichment of the human knowledge, even if it is just enjoyed by a few.

The persons indirectly involved in this paper number over the hundreds: Parents, tutors, siblings, friends, enemies, colleagues, loved ones and the scientific community all play a role and each single word placed can be attributed to one person or another.

I would like to give acknowledgement to some of the hundreds for their direct influence be it for knowledge, ability or motivation. Firstly, these studies would have never been conducted if it were not for my past managers: Petr Matousek, Engelbert Hetzmannseder and Andrea Kirschner, you supported and encouraged my ambition by funding and providing resources necessary.

Also, many colleagues contributed directly to the pool of knowledge necessary for conducting the studies, therefore a special thanks to Michal Horn and Pavel Dedourek which helped me to improve my programming knowledge in embedded devices.

Self-evidently thanks to Prague College and its staff which were the main source of knowledge throughout the studies. Also, a special thanks to my project supervisor Bohus Ziskal for the close guidance and confidence for this rather unusual computing project.

Most importantly I want to thank Natalia Estefania Juardo Espin for being the motivational pillar of my ambitions and for her versed technical knowledge used for helping whenever possible.

Table of Content

1 Introduction

This project report documents the implementation of the Local Control Function Interpreter (LCF Interpreter) project as the BSc-Computing Project of Samuel A. Marti, student of Prague College and Teesside University. It is concerned with the development of an interpreter which targets a specific microcontroller of Eaton, allowing for the execution of logical control algorithms within industrial devices.

The project was implemented based on the Local Control Function Interpreter Computing Project Proposal, see (Appendix A).

1.1 Glossary & Abbreviations

This glossary defines distinct terms used within this document.

Term	Definition
EEIC	Eaton European Innovation Center, a branch of Eaton Corporation, located in Roztoky, Czechia. (Eaton Corporation, 2018)
LCF Project	Local Control Functions Project, an Eaton project executed within EEIC.
LCF System	The system developed as result of the LCF Project.
LCF Interpreter	The subject of this computing project which is an interpreter and part of the LCF System.
SWD	SmartWire DT, a proprietary Fieldbus System by Eaton. (Eaton Corporation, 2018)
SWD Device	A slave device/component able to use SWD as Communication fieldbus.
SWD Coordinator	A master device/component which is controlling the SWD fieldbus.
ASIC/ASIC2	Proprietary microcontroller of Eaton used within SWD Devices.
PLC	Programmable Logic Controller. (UNITRONICS, 2018)
MOEM	A market evolved around Micro-Opto-Electro-Mechanical systems. (Eaton Corporation, 2016)
HMI	Human Machine Interface. (Eaton Corporation, 2018)
LED	Light Emitting Diode. (Cambridge, 2018)
IIoT	Industrial Internet of Things. (General Electrics, 2019)
RAM	Random Access Memory.
IOMUX	Input and output multiplexer, used for selecting pin functionality.
SWD Assist	An Eaton Software tool used by customers for configuring the SWD bus.
LCF Instructions	Instructions generated for LCF Interpreter by the LCF Compiler.

Table 1 Glossary & Abbreviations

2 Background

In the end of 2017 the LCF Project was initialized by EEIC as part of the "Factory of Future" initiative. The project targets the development of a distributed automation system which would help Eaton to identify new product opportunities and to assess technological possibilities of the existing automation system of Eaton. In the beginning 2018 the LCF interpreter was defined as a component and subproject of the overall LCF Project and in the end of 2018 the implementation of it was chosen as the subject of this computing project.

2.1 Clientship

The LCF Interpreter project was proposed by EEIC and is executed with their clientship. The clientship is ensured by the employment of the project assignee within EEIC and the LCF Project where the assignee holds a project leading and engineering role. (Andrea & Samuel, 2018)

EEIC is a development centre of Eaton located in the Czech Republic employing over hundred Engineers which are working directly for all Eaton businesses as it was established with the intention of creating a cross business innovation hub. (Eaton Corporation, 2018)

Eaton is a multinational corporation founded in the United States of America with their headquarters located in Ireland. Eaton identifies itself as being a power management company with having businesses in Aerospace, Hydraulics, Filtration, Golfing, Vehicle, Electrical & Industrial.

Note: The LCF Project contributes to the Electrical business unit of Eaton.

In 2018 Eaton employed worldwide around 99'000 people and had a revenue of 21.6 Billion US dollars. (Eaton Corporation, 2019)

2.2 Local Control Function Project

The LCF Project is concerned with the development of a distributed automation system (LCF System), which is utilizing Eaton technology such as SWD and ASIC2. The resulting LCF System will consist of components which can be utilized in industrial devices allowing for decentralized decision taking.

[Figure 1 LCF Components] visualizes all components of the LCF System, with their intended location within the LCF System and describes the LCF Interpreter as the main subject of this computing project. Further description of the LCF System can be found in (Appendix_F).

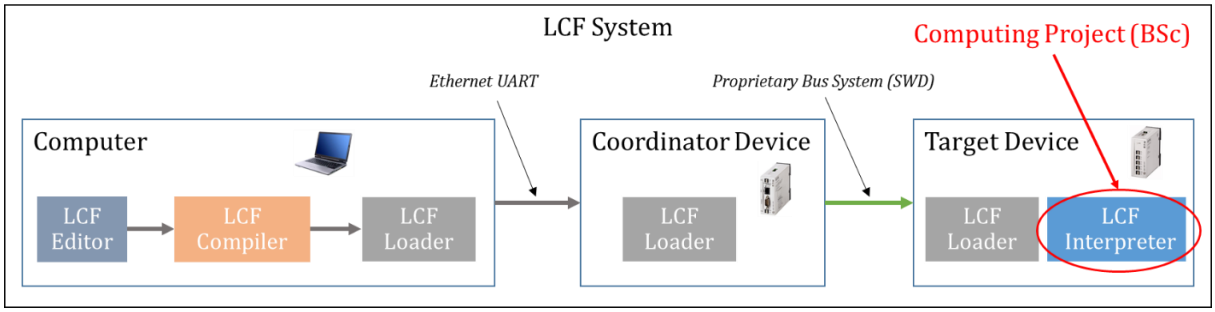


Figure 1 LCF Components

3 Problem Analysis

The problem analysis relates to the whole LCF Project [2.2], from which problems for the LCF Interpreter can be cascaded. The analysis was conducted by studying the current automation system of Eaton (SWD) (Eaton Corporation, 2017), conducting interviews with product manager (Heribert Einwag, 2018) and observing trends in the market. It states the current state of the art within Eaton and compares it to trends in the industry.

Note: The underlying driver of the LCF Project is to keep the Eaton automation system competitive.

3.1 Current State

Eaton's industrial automation system for machines and control cabinets is based on SmartWire DT (SWD). This proprietary field bus system and its components targets MOEM customers with the value proposition of a simple and smart wiring solution, which ultimately reduces the cost of ownership by minimizing installation, debugging and maintenance cost of machinery and control cabinets. (Eaton Corporation, 2017)

[Figure 2 SmartWire DT Example] visualise the SWD bus connected with a gateway, two motor starters with motor-protective circuit-breaker and three command & control devices.

Figure 2 SmartWire DT Example

SWD is used for controlling a variety of industrial components such as: Motor Starters, Circuit Breakers, IO Modules, Frequency Drives, Push Buttons, LED's, HMI and PLC's. [Figure 3 SmartWire DT Control Cabinet] visualizes a control cabinet containing SWD devices in an exemplary set up

Figure 3 SmartWire DT Control Cabinet

3.1.1 SWD Technology

The SWD fieldbus system functions fundamentally in a master-slave scheme where a master device is named SWD Coordinator and a slave device is named SWD Device. A SWD Coordinator initializes the fieldbus and assigns the addresses to all SWD Devices based on their physical position within the fieldbus system. The coordinator will also request and write data from and to the devices in cyclic and time deterministic manner.

3.1.1.1 SWD Coordinator

The SWD Coordinator controls the fieldbus and the SWD Devices; thus, the coordinator is usually implemented within control devices such as: PLC's, HMI's or gateways.

3.1.1.2 SWD Device

Each SWD Device embeds a ASIC/ASIC2 microcontroller which contains the implementation of the SWD protocol. A SWD Devices can be either a periphery, actuator or sensor.

(Eaton Corporation, 2018), (Appendix_B)

3.2 Identified Limitation of SWD

The comparably high reaction time of SWD was identified is a major limitation of the system based on an interview with (Magdolinic, 2018). In a best-case scenario, the current SWD system would only achieve a reaction time of 10-20ms, referring to [3.2.1]. This limitation does not allow SWD to penetrate markets which require a low reaction time of its control systems. (Eaton Corporation, 2017)

Example: The controlling of a variable frequency drive used in an injection molding machine requires a reaction time of less than ~1ms according to (Magdolinic, 2018). *[1]

3.2.1 SWD Reaction Time Calculation

The reaction time of the SWD system is mainly influenced by the cycle time of the SWD communication and the cycle time used for computation by a PLC.

The following calculation will assume a XC-152 as SWD Coordinator which defines 4ms as the shortest computational cycle time, whereas the shortest cycle time for the SWD communication is defined as 3ms. (Eaton Corporation, 2017), (Eaton Corporation, 2015)

Given

CycleTimeSWD = 3ms	CycleTimePLC = 4ms
SWDCyclesWorst = 4	PLCCyclesWorst = 2
SWDCyclesBest = 2	PLCCycleBest =1

$$ReactionTime_{Worst} = \text{SWDCycleWorst} * \text{CycleTimeSWD} + \text{PLCCycleWorst} * \text{CycleTimePLC}$$
$$20\text{ms} = 4 * 3\text{ms} + 2 * 4\text{ms}$$

$$ReactionTime_{Best} = \text{SWDCycleBest} * \text{CycleTimeSWD} + \text{PLCCycleBest} * \text{CycleTimePLC}$$

*[1] *In this context reaction time is defined as the time needed for a signal of a SWD Device being communicated to the SWD Coordinator until the response of the SWD Coordinator is propagated back to any SWD Device.*

3.3 Trend

The term Industry 4.0 is a widely used for describing the recent and future transformation of manufacturing and machine building. This transformation is driven by technologies, increasing hardware performance and the aim of a more efficient, flexible, predictable and autonomous production systems. Thus, systems embracing Industry 4.0 will incorporate topics such as: Predictive Maintenance, IoT, Edge Computing, Intelligent Devices, Big Data etc. of which all follow the underlying Industry 4.0 design principles of interconnection, information transparency, technical assistance and decentralized decision taking for cyber physical systems. (Klingenberg & Antunes, 2017), (Hermann, et al., 2016)

3.4 Conclusion

Needed features can be identified by comparing the present SWD system with the trend of the industry [3.3].

[Table 2 Industry 4.0 Assessment] assesses the Industry 4.0 design principles with the current SWD system and states collected business needs. It also suggests steps which can be taken for approximating the system to the design principles.

Principle	Assessment	Approximation Step	Business Need
Interconnection	SWD already implements several gateway devices used for the interconnection to other systems or IoT. (Eaton Corporation, 2018)	Implementing more IIoT protocols such as: OPC UA, MQTT natively into SWD. (Automation.com, 2015)	Most customers do not express an urgent need for more IIoT capable protocols, but that might change in the future. (Heribert Einwag, 2018)
Information Transparency	SWD can collect and propagates a huge amount of SWD device data which is used for controlling the devices.	A model for each SWD Device can be creating, allowing customers creating live digital twins. (Uhlemanna, et al., 2017)	Since information transparency is partially based on interconnection of the system most customers are not looking into this yet.
Technical Assistance	SWD provides user with a diagnostic and debugging tool for troubleshooting. (Eaton Corporation, 2018)	New ways of troubleshooting the SWD bus can be explored such as virtual reality.	The current tools for assisting personnel are adequate.
Decentralized Decisions	SWD Devices do not take decentralized decisions and are controlled over the SWD Coordinator	Enable SWD devices to be programable and implement a device-to-device communication concept.	In some situations, customers request edge computing to gain a faster reaction time or save hardware costs on a PLC. (Markus, 2018)

Table 2 Industry 4.0 Assessment

Concluding on [Table 2 Industry 4.0 Assessment] only the Industry 4.0 principle “Decentralized Decisions” can be identified as currently most pressuring and relevant to SWD. Thus, the LCF Project aligns to the trend and solves the identified limitation stated in [3.2].

3.5 Problem Statement

For Eaton to keep the SWD system competitive in the coming years it will need to be more approximated to the trends of the market [3.3].

Based on the LCF Project [2.2], identified SWD limitation [3.2], the conclusion of the problem analysis [3.4] and the project alignment meeting (Andrea & Samuel, 2018) following problem statements can be drafted:

- SWD Devices cannot be programmed to take their own decisions.
- SWD Devices cannot communicate with each other directly.
- The SWD system does not provide models for creating digital twins.
- The SWD system does yet not support IIoT protocols for the interconnection directly *[2].
- Decision taking of the SWD system has a too long reaction time for some applications.

3.6 Proposed Solution

The LCF System would allow SWD Devices to be programmed using boolean algebra and so it tackles some of the problem statements in [3.5]. However, the problem statements relating to the digital twin and IIoT protocols are not in the scope of the LCF System and can be disregarded.

The LCF Interpreter which is implemented as part of this computing project will complete the LCF System and is so with inherently part of the solution for solving the relevant problem statements.

3.7 Assumptions

The assumptions are based on interviews with experts of the SWD technology and ASIC2. (Arguinariz & Dedourek, 2018)

For the execution for the LCF project it is assumed that the current SWD technology will allow for a distributed automation technology to be developed without the need of introducing changes to the current SWD protocol or system, thus keeping SWD Devices using the new technology compatible with current SWD Devices. (Appendix_C)

It is also assumed that the ASCI2 embeds enough computational power and memory for the solution [3.6] to be applied together with the device and SWD specific firmware. (Eaton Industries GmbH, 2018)

*[2] *Currently support can be achieved by using third party tools such as the Codesys environment and SWD touch panels (Codesys, 2018).*

4 Project Specification

The topic of this computing project is the implementation of the LCF Interpreter along with the research of relating topics to the LCF Project [2.2]. This chapter will specify the scope and goals of the LCF Interpreter including it deliverables.

4.1 Project Aim

The project aims to develop the LCF Interpreter as embedded firmware of a target SWD Device, allowing it to interpret given instructions by the LCF System. Additionally, it aims to research defined topics relevant to the SWD Device with the aim of defining the next steps necessary for the commercialization of such technology. Ultimately, the LCF Interpreter will complete the first version of the SWD System. *[3]

4.2 Project Objectives

Based on the project aim [4.1] and the initial project alignment meeting (Andrea & Samuel, 2018) the objectives were defined in [Table 3 Project Objectives].

ID	Objective
OBJ01	Implementation of the LCF Interpreter on a target SWD Device embedded on ASIC2. Allowing the target device to be programmed using digital logic.
OBJ02	Gain a LCF Interpreter prototype for showcasing the LCF System.
OBJ03	Assessment and research of time deterministic behaviour and its potential implementation within the LCF Interpreter, so that future versions of LCF can implement such a behaviour.
OBJ04	Definition and research of device-to-device communication with its potential implementation in LCF and SWD, so that an implementation of a device-to-device communication with SWD and LCF can be started.
OBJ05	Assessment of PLC standards IEC61131 and IEC61499 in regards of LCF by identifying discrepancies and suggesting actions to take based on the results.
OBJ06	Suggestion of an automated testing system for the produced prototype, so that a testing system can be implemented and used for the target SWD Devices.

Table 3 Project Objectives

Note: The objectives changed compared to the project proposal, see [12.1].

4.3 Target SWD Device

By conducting meetings with business partners, engineering and product management,

It was decided that the prototype implementation of the LCF Interpreter [OBJ01] will be a module SWD with four pairs of digital IO named [EU5E-SWD-4D4D]. The decision was taken because, hardware embedding the ASIC2 is available, the value proposition is clear, the device could be used as standalone controller and it is suitable for demonstrations.

Additionally, an IO module implementing the LCF Interpreter will be distinct from other IO modules due to its ability of decentralized decision taking which ultimately reduces the reaction time and load of the automation system. (Heribert Einwag, 2018), (Appendix_E)

*[3] *Currently the LCF Project is a R&D activity which results in porotypes used for further decisions & development.*

4.4 Used Technology and Hardware

The used technology and hardware is given and defined by the project client [2.1] and must be used for the project implementation. The following table lists defined system elements, stating their purpose and gives a rationale for their inclusion based on the coordination with product managers and engineers at Eaton.

ID & Name	Description	Purpose	Visualization	Rational
ASIC2	New generation of microcontroller for SWD Devices	Host and execute the LCF interpreter.	 Figure 4 ASIC2	The new ASCI2 will allow for more processing power and memory compared to the ASIC and is planned to be implemented in all SWD Devices. (Eaton Industries GmbH, 2018)
EU5E-SWD-4D4D	Four digital Input and four digital Output Module.	Target SWD Device.	 Figure 5 EU5E-SWD-4D4D	Will allow the implementation of the LCF Interpreter and so, completes the first LCF Prototype. (Appendix_E)
IAR Embedded Workbench	Software tool used for programming hardware in C.	Programming and debugging of ASIC2.	Figure 6 IAR Embedded Workbench	The IAR Embedded workbench is the standard within Eaton and thus also the only IDE to implement a ASIC2 compiler.
ASIC2 Library	Software library in C for ASCI2.	Implements functions for manipulating the ASIC2 hardware.	<C>	The library eases the development of the firmware.
ASIC2 Development Board	Development board containing ASIC2 with peripheries.	Allows for easier development of ASIC2 based products	 Figure 7 ASCI2 Development Board	The development board will allow to use its peripheries for an easier debugging and development of the LCF Interpreter.
SWD Programmer	A device used for loading the firmware into the ASIC2 memory.	Allows to program the ASIC2 in production environment.	 Figure 8 SWD PROG	Needed for loading new firmware and updating the LCF Instructions.

Table 4 Given Technology & Hardware

4.5 Project Deliverables

Based on the project aim [4.1], objectives [4.2] and the project alignment meeting (Andrea & Samuel, 2018) the deliverables of the technology and research can be defined.

[Table 5 Deliverables] defines the project deliverables while matching them with their relevant objectives.

ID	Deliverable	Objective
D01	C source code, implementing the LCF Interpreter using the [ASIC2] and [ASIC2 Library].	OBJ01, OBJ02
D02	A [EU5E-SWD-4D4D] prototype, embedding the LCF Interpreter.	OBJ02
D03	A description stating the behaviour of time deterministic control systems (PLC) and how such a behaviour can be implemented within the LCF Interpreter.	OBJ03
D04	A description of a potential device-to-device communication concept for the SWD Device based on technological restrains.	OBJ04
D05	A table assessing the standards IEC61131 and IEC61499 in regards of LCF and a conclusion suggesting next steps to take.	OBJ05
D06	Topological and conceptual definition of an automated Test System for the prototype IO module [4.4] utilizing the LCF Interpreter.	OBJ06

Table 5 Deliverables

4.6 Project Exclusion

- The project does not include any hardware development necessary for the [EU5E-SWD-4D4D] prototype referring to [TD02].

- The implemented interpreter shall be validated functionally, thus the validation of performance or compliance of IEC61131 are out of scope.

- The project is mainly concerned with the implementation of the LCF Interpreter. That does not include other LCF subprojects such as the editor, compiler and loader.

5 Requirements

For each deliverable a table of requirements was defined and agreed upon based on objectives [4.2] and aim [4.1]. (Andrea & Samuel, 2018).

Clear requirements are drafted by defining the object, the target to achieve and the reason for doing so, as defined in [Figure 9 Requirement Template].

The <OBJECT>, must <FEATURE/PROBLEM/GOAL>, so that <REASON>.

Figure 9 Requirement Template

5.1 D01 & D02 – Implementation of LCF Interpreter

ID	Requirement
R01	The LCF Interpreter, must parse LCF instructions from the memory of the ASIC2, so that it can be interpreted.
R02	The LCF Interpreter, can interpret following list of instructions as defined in (Appendix_H), so that designed programs with the LCF System can be executed. PUSH, PUSH_P, POP, POP_P, MAX, MIN, SUB, COMPARE_NEQ.
R03	The LCF Interpreter, starts interpretation as soon as the [EU5E-SWD-4D4D] device is powered, so that the device is functional from the beginning.
R04	The LCF Interpreter interprets all instructions in a cyclical manner, so that there is a clear order of the interpretation process.
R05	The LCF Interpreter, does not stop interpreting, so that device will always react to changes on its IO's.
R06	The LCF Interpreter can read digital signals from the [EU5E-SWD-4D4D] inputs, so that their values can be used by the interpreter. (Appendix_E).
R07	[EU5E-SWD-4D4D] IO inputs, must be read in the beginning of each cycle, so that later changes will not influence the computation of the current cycle.
R08	The LCF Interpreter can write digital signals to the [EU5E-SWD-4D4D] outputs, so that computed results take effect. (Appendix_E).
R09	[EU5E-SWD-4D4D] IO outputs, must be written in the end of each cycle, so that induced changes will not influence the computation of the current cycle.

Table 6 LCF Interpreter Requirements

5.2 D03 – Time Deterministic Behaviour

ID	Requirement
R31	The section, will describe time deterministic behaviour in regards of PLC, so that an understanding is ensured.
R32	The section, will conclude on how time deterministic behaviour could be implemented on the LCF Interpreter.

Table 7 Time Deterministic Behaviour Research Requirements

5.3 D04 – SWD Device-to-Device Communication

ID	Requirement
R41	The section, will describe how SWD communication is currently conducted.
R43	The section, will conclude on how device-to-device communication could be implemented in the LCF System using SWD.

Table 8 Device-to-Device Communication Research Requirements

5.4 D05 – Assessing of Standards

ID	Requirement
R51	The section, will describe how well the LCF Systems complies with the IEC61131 standard, so that the information can be used for concluding on appropriate actions.
R52	The section, will describe how well the LCF Systems complies with the IEC61499 standard, so that the information can be used for concluding on appropriate actions.
R53	The section, will suggested potential approximation actions for achieving compliance with a standard, so that product management can take an informed decision.

Table 9 Standards Assessment Requirements

5.5 D06 – Automated Test System

ID	Requirement
R61	The section, will describe a system for automatic testing of the LCF Interpreter on the target SWD Device [4.3], so that a test system can be designed.
R62	The section, defines the system topology and technology of the automatic test system for the LCF Interpreter on the target SWD Device [4.3], so a test system can be implemented.

Table 10 Standards Assessment Requirements

6 Methodologies and Practices

This chapter defines used methodologies, practices and processes used within the computing project.

6.1 Project Methodology

This computing project demands clearly defined requirements, scope, deliverables and deadline with extensive documentation. Based on that information the waterfall model was chosen as project methodology since it thrives in rigid and documentation heavy environments with a clear deadline unlike agile, scrum and KANBAN which excel in flexible long-term projects where the environment, scope and requirements are under constant change.

6.2 Software Development Principles

The used software development principles aim to develop a more maintainable and readable code and are based on the "Clean Code" book. (Martin, 2008)

ID	Name	Description
SP01	Conventions	The project follows consistent conventions aligned with Eaton.
SP02	Simple	Implemented solutions aim to be as simple as feasible.
SP03	Boy Scout	The code base is left always cleaner than it was before.
SP04	Descriptive Naming	The naming of functions, variables etc shall be unambiguously.
SP05	Smaller is Better	Many small functions with few parameters is superior opposed to a big function implementing the same.
SP06	One Thing	A function only implements a single behaviour at once at has no side effects.
SP07	One Argument	The best amount of arguments for a function is one.
SP08	Documentation	The source code is documented by doxygen comments for generating software documentation.

Table 11 Software Development Principles

7 Research

This chapter presents the process, results and conclusions for the executed research within the computing project. The research mainly contributes to the deliverables [D03], [D04] and [D05] as defined in [4.5].

7.1 Time Deterministic behaviour in LCF

This section will conclude on the potential implementation of time deterministic behaviour within the LCF Interpreter by researching and defining the implementation of time deterministic behaviour in existing controls systems.

7.1.1 Definition

A system with time deterministic behaviour has fixed time constraints for reading, computing and propagating data. *[4] (Petters, 2007)

7.1.2 Approach

The subject was researched by studying literature such as IEC61131 standard, relevant articles and analysing the behaviour of existing PLC systems from different vendors such as Eaton and Siemens. (International Electrotechnical Commission, 2003), (BUDIMIR, 2017), (Berger, 2014), (3S-Smart Software Solutions GmbH, 2019)

The research concludes on a definition of how time deterministic behaviour shall behave within industrial control systems so that a potential implementation can be proposed regarding the LCF Interpreter.

*[4] *Within this document time deterministic behaviour is used as a synonym of "real-time" which is a widely used term within the literature.*

7.1.3 Result

[Table 12 Time Deterministic Behaviour Observations] summarizes the finding of time deterministic behaviour for each studied item.

Item	Behaviour
IEC61131	The application may consist of programs which may be executed periodically.
Eaton PLC	The application consists of several programs where a fixed time constraint can be defined for each program. The error behaviour in case of a program being out of time can be defined.
Siemens PLC	Same behaviour as Eaton PLC, no significant differences were found.

Table 12 Time Deterministic Behaviour Observations

Based on the research [Figure 10 PLC Execution Cycle] can be drawn. It describes the execution cycle of one program in a PLC and the implementation of time deterministic behaviour.

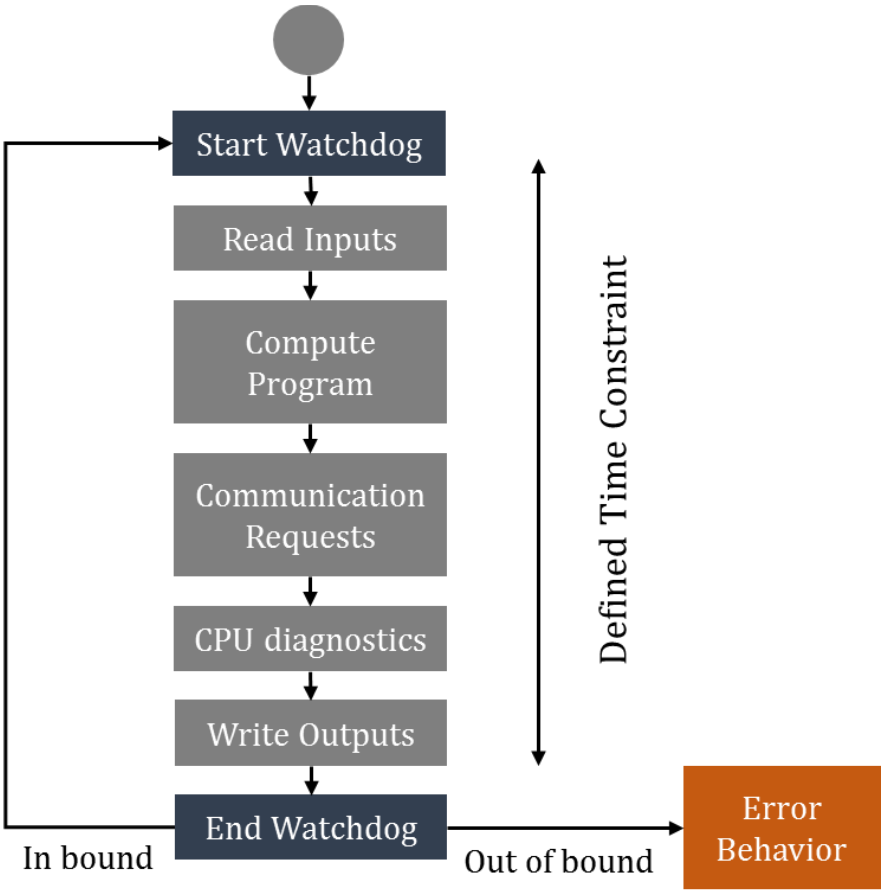


Figure 10 PLC Execution Cycle

In PLC's a single entity (watchdog) is responsible to ensure that defined time constrains are met. In case of an out-of-bound the watchdog will cease the execution of the program and enter a defined error behaviour. The specific implementation of time determinist behaviour studied within the LCF Interpreter is described in [8.2].

7.2 Standards Assessment

To successfully engineer, produce and commercialize an industrial product utilizing LCF System or any of its sub-components, compliance of LCF with relevant standards must be assessed and analysed, allowing for an informed roadmap planning of new LCF versions and an educated adaptation of the LCF System for products to be.

The assessment should point out deviations of the current LCF Project in regards of relevant standards and propose mitigation strategies for future LCF versions or products to be.

7.2.1 Approach

The research was conducted by identifying relevant standards [7.2.2] used by existing PLC's and assessing each subject of the standards in terms of current and potential compliance of the LCF system. Based on that assessment conclusions regarding potential mitigation and approximation strategies for the LCF System can be drawn.

7.2.2 Relevant Standards

Based on conducted research the standards IEC61131 and IEC61499 were identified as relevant towards the LCF System.

IEC61131 is a widely adapted standard of the industry and concerned with the classical PLC, whereas IEC61499 is a newer PLC standard targeting distributed control systems and has yet not found acceptance within the industry, despite being promoted by academia (Rikin, 2015), (Thramboulidis, 2013).

The assessment with IEC61131 will disclose how much LCF differs to a classical PLC, while the assessment with IEC61499 on the other hand will determine if LCF could be one of the first adaptations of that standard.

Standards about electromagnetic compatibility/disturbance, environmental testing, and electronic equipment such as: IEC EN61000-4, IEC60068-2-6, IEC60068-2-27, EN55011 and EN50178 are not relevant for the LCF System, since they are concerned with the hardware of the products to be and as defined in [4.6] LCF System does not influence the hardware design in a direct manner.

7.2.3 Result

The results were formalized by creating a table of the assessed parts and sections with the matching LCF components and identified mitigation strategies. The resulting tables can be found in the appendix under (Appendix_N) and (Appendix_O).

7.2.3.1 IEC61131

The IEC61131 is split into nine parts, the relevance assessment of these parts has shown that only the parts IEC61131-1 and IEC61131-2 are relevant towards LCF.

7.2.3.1.1 IEC61131-1

IEC61131-1 describes mainly the characteristics of a PLC and the assessment leads to the conclusion that the current implementation of the LCF System will not comply with the most sections of this part. A future version of LCF can comply with more sections. However, a full compliance with the standard will not be feasible mainly due the limited hardware capacity of the ASIC2 which is hosting the LCF Interpreter behaviour.

7.2.3.1.2 IEC61131-2

IEC61131-2 defines required equipment and requirement tests of a PLC. The assessment shows that current implemented version of the LCF System cannot be validated according the standard. However, a version of the LCF System to be commercialized will be able to be validated according to the standard except for remote IO ([6.8]Appendix_N) and memory power back up ([6.3]Appendix_N)

7.2.3.1.3 Conclusion

The reaction to this research results could be the plain acceptance of the non-compliance, the approximation of all feasible sections to the standard or the creation of a new standard especially accommodating for physically distributed automation system using limited low-end hardware.

However, according to the standards the current LCF version cannot be categorized or validated as PLC since the most characteristics defined in the standard are not met. However, the possibility is given that future version of LCF might comply more with the standard but given its weaker hardware a full characterization as PLCs would be probably never possible.

Thus, the most accommodating would be to define a new standard for control system in a distributed environment based on the weaker hardware, where the current IEC61131 can be used as basis. (International Electrotechnical Commission, 2017-2018), (International Electrotechnical Commission, 2003)

7.2.3.2 IEC61499

From the four parts of the IEC61499 only the IEC61499-1 and IEC61499-4 have been deemed as relevant for the LCF System.

7.2.3.2.1 IEC61499-1

The IEC61499-1 describes the architecture of distributed and event driven execution model in automation systems, unlike the IEC61131 which focuses on the cyclic execution model. The implementation of such an execution model will require adaptations especially in the software tools used by the user of the system such as the programming environment, simulator and compiler. The current version of LCF does not comply with any subject defined by the standard mainly due the programming model not being based on function blocks (wisdomjobs, 2018).

7.2.3.2.2 IEC61449-4

IEC61449-4 describes how interoperability, portability and configurability between different vendors tool adhering the standard can be achieved. The current LCF system will not comply with any of the defined subjects especially since the LCF system is not meant to be used together with third party vendors. Future versions of the LCF System might comply with the standard however that would require SWD to comply as well since SWD manages the communication between devices and tools.

7.2.3.2.3 Conclusion

The benefits of complying with the IEC61499 standard are doubtful since adaptations within the industry are rare and the required resources of adapting the whole system outweigh the benefit. Especially the required adaptability of the system with third party tools is disputable since it would require SWD to adapt as well. Nevertheless, the adaptation of the device-to-device communication concept and event driven behaviour of function blocks would be beneficial to the functionality of the system. (European Committee for Electrotechnical Standardization, 2012), (European Committee for Electrotechnical Standardization, 2013)

7.3 Device-to-Device Communication

In order to embrace the industry 4.0 trend of enabling decentralized decision taking [3.3] for the LCF System a device-to-device communication concept based on the SWD technology must be conceptualized and defined. This section outlines the current state of the SWD communication and proposes an implementable device-to-device communication concept for SWD which is also backward combability to older SWD version.

7.3.1 SWD Communication

Based on [3.1.1] and (Appendix_C) this chapter summarizes the current behaviour of the SWD communication, so that a device-to-device communication concept can be drafted based on it.

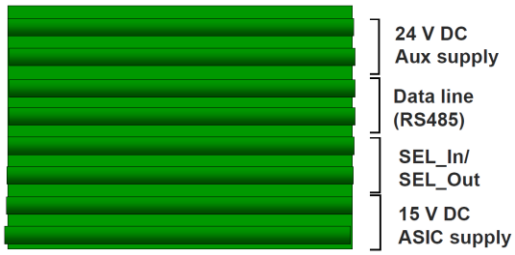


Figure 11 SWD Cable Layout

[Figure 11 SWD Cable Layout] describes the pin layout of the SWD cable used for the SWD communication. The 24V/GND pair is used by some SWD Devices [3.1.1.2] as power source for their hardware functionalities. The data line is used for the data communication between the SWD Coordinator [3.1.1.1] and SWD Devices. The SEL_In/Out are used by the SWD protocol to address each device on the bus and the 15V/GND pair is used as power supply for the ASIC2 in the SWD Device.

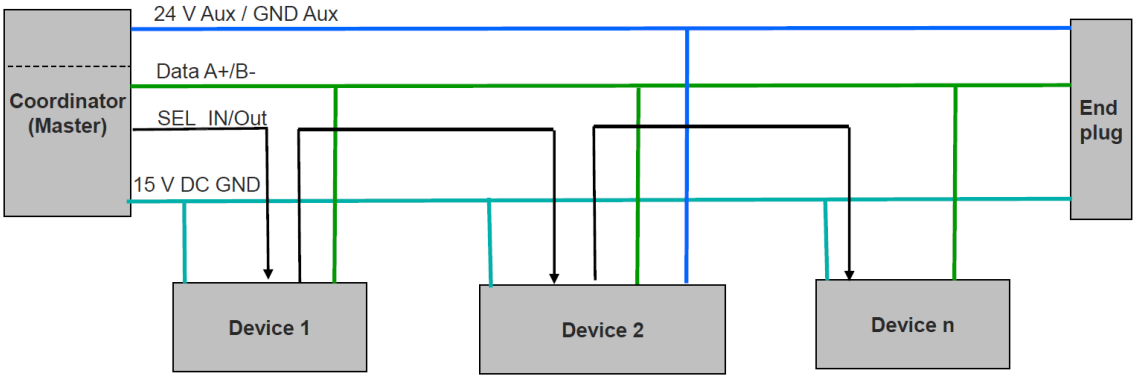


Figure 12 SWD Topology

[Figure 12 SWD Topology] visualizes an exemplary topology utilizing SWD with a set of SWD Devices. Before any data communication can occur the SWD protocol will execute a start-up procedure in which each device is assigned its address based on the physical position within the bus. In that start-up procedure the SWD Coordinator will use the SEL_In/Out line in order to address each device and negotiate their data profiles. After a successful start-up procedure SWD will start its communication cycle.

The SWD Coordinator initiates all communication cycles periodically. In every communication cycle, every device will broadcast its data onto the bus within its dedicated time frame. (Appendix_C)

7.3.2 Identified Solution

A device-to-device communication using SWD can be achieved by utilizing the fact that each SWD Device is broadcasting its data to the whole bus as defined in [7.3.1]. Ergo, while SWD Device XX is broadcasting data on the bus, all other SWD Devices can receive and use that data. However, there is no SWD Device which is doing so, mainly due to missing firmware and addressing of the data.

From a SWD Device point of view the data broadcasted by other SWD Devices are meaningless since the SWD Device is not aware of its surrounding devices and only listens to the data of the SWD Coordinator.

7.3.3 Problem

The major problem of the proposed solution [7.3.2] is that current SWD Devices are not aware of other SWD Devices on the bus, except the SWD Coordinator. Thus, for a successful data exchange between devices the address of the target data will have to be determined, which is not a trivial task since the data addresses are based on the physical bus positions and data models of their host devices.

7.3.4 Suggested Implementation

The problem [7.3.3] can be solved by loading the exact addresses of data to be retrieved together with the LCF Instructions onto the target SWD Device, however that will require the LCF Editor [2.2] to know the exact SWD bus layout in order compute the correct addresses. Therefore, the current SWD configuration tool (SWD Assist) may be used, since it is used for configuring and monitoring the SWD bus and requires the user to define the SWD bus layout. Ergo, the same layout can be used by the LCF Editor to compute the target addresses.

Concluding the table [Table 13 Device -to-Device Communication Implementation Steps] suggests the implementation steps required for achieving a device-to-device communication with SWD.

ID	Implementation Step	Target	Reason
01	The SWD Assist must allow the SWD layout to be exported.	SWD Assist	So, that the layout can be used by the LCF Editor
02	The LCF Editor should compute the addresses for variables communication between devices based on the SWD layout.	LCF Editor	So that the data can be loaded by the LCF loader.
03	The LCF Loader should load instructions and addresses given by the LCF Editor and LCF Compiler to the target SWD Device	LCF Editor, LCF Loader, LCF Compiler	So that the SWD Device knows the addresses required.
04	The LCF Interpreter uses the given addresses for reading values form the SWD Bus.		So that the interpreter can use the values for the computation.

Table 13 Device -to-Device Communication Implementation Steps

8 Project Implementation

The project was executed according to the project methodology [6.1] and followed the project plan specified in ([1.10] Appendix_A). This chapter will present the results of the executed computing project.

8.1 LCF Interpreter

The chapter describes the implemented LCF Interpreter on the target SWD Device and relates to the deliverable [TD01] and [TD02].

8.1.1 Process

The LCF Interpreter was implemented by analysing first related requirements [5.1], to draft component and state machine diagrams [8.1.2.1] [8.1.2.2], so that an overview and understanding of the firmware to be developed and can be gained. Additionally, a flowchart was created for a deeper understanding of the interpreting cycle [8.1.2.3].

Based on the drafted diagrams the specific features of each component were defined [8.1.2.1.1-8.1.2.1.12] and implemented according to the software development principles [6.2], starting with the components which depend on already existing components. Additionally, each component was validated manually for its functionality after its completion. For some components unit tests were created given a testability without the necessity of using mock objects.

The ASIC2 Development Board [4.4] was used for development and debugging of the firmware and the EU5E-SWD-4D4D [4.4] was only utilized for the final implementation and deployment.

8.1.2 Software Architecture

By analysing the requirements [5.1] and given hardware restrictions (Eaton Industries GmbH, 2018) regarding the software architecture, flowing conclusions concerning the software architecture were drawn:

- The architecture should allow to change, add and remove LCF instructions with minimal effort and intrusion of the source code.
- The LCF Interpreter is not depended on the memory sources and changes can be introduces with minimal effort.
- The IO's used by the LCF Interpreter can be adapted with minimal effort and intrusion of the source code.

Summarizing, the architecture aims to be extendable so that the adding and modification of instructions, memory and peripheries can be achieved by minimal effort and intrusion of the core algorithm. This can be achieved by encapsulating dependencies within designated components which provide an internal API for the interpreter.

8.1.2.1 Component Diagram

The first step of designing the architecture leads to a component diagram which helps to understand the overall component structure and describes the interconnections but is limited in the expression of the intended functionality.

[Figure 13 LCF Interpreter Component Diagram] visualizes the component diagram of the LCF interpreter, it describes each component and its dependencies.

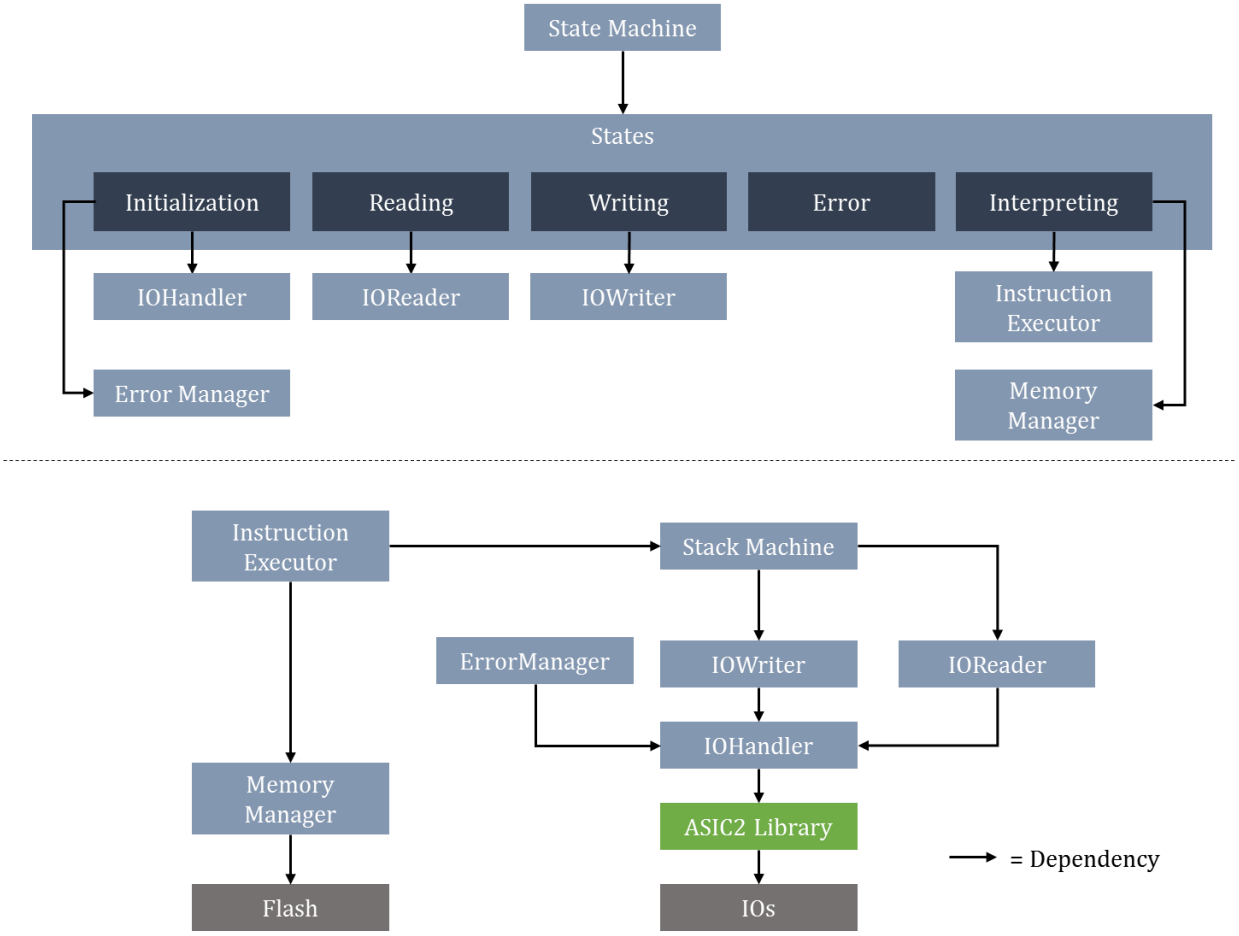


Figure 13 LCF Interpreter Component Diagram

Following descriptions relates to [Figure 13 LCF Interpreter Component Diagram].

8.1.2.1.1 ASIC2 Hardware

The LCF Interpreter will access memory and peripheries of the ASIC2. However, the usage of peripheries is limited since the hardware of the target SWD Device [4.3] uses only the IMOUX pads 1-4 for the digital inputs and 18-21 for the digital outputs of the ASIC2 [Appendix_E] as defined in the schematics [Appendix_J].

8.1.2.1.2 ASIC2 Library

The ASIC2 library is provided by the project client [2.1] and implements wrapper and access functions in C for configuring and manipulating the ASIC2. The LCF interpreter mainly uses it for configuring the used IO's by the ASIC2.

8.1.2.1.3 IO Handler

The IO Handler implements the handling and initialization of the physical peripheries of the ASIC2 and provides an abstracted API for the IO Writer and IO Reader. Changes relating to the physical peripheries will be implemented within this component.

8.1.2.1.4 IO Reader

The IO Reader implements functions for accessing the input register and functions for reading relevant inputs of the target SWD Device [4.3] into the input register, by using the IO Handler

8.1.2.1.5 IO Writer

The IO Writer implements functions for writing into the output register and functions for the propagation of the output register to the physical outputs of the target SWD Device [4.3] by using the IO Handler.

8.1.2.1.6 Memory Manager

The Memory Manager manages the flash memory of the ASIC2 containing the LCF Instructions for the interpreter. It implements functions for moving within the memory and obtaining information such as checksum, size and version of the LCF Instructions.

8.1.2.1.7 Stack Machine

The Stack Machines implements the core algorithm for computing results based upon the LCF Instructions. It uses the registers of the IO Reader and IO Writer for retrieving and storing computation results.

8.1.2.1.8 Instruction Executor

The Instruction Executor identifies LCF Instruction in the flash memory of the ASIC2 by using the Memory Manager and initializes the computation of them by the Stack Machine.

8.1.2.1.9 States

The states component defines several states the LCF Interpreter can be in and is used by the State Machine. Each state orchestrates the relevant components for executing their functionality.

8.1.2.1.10 State Machines

The State Machine organizes and defines the transitions of the LCF Interpreter states defined in the States component and so with embodies the highest level of abstraction within the firmware.

8.1.2.1.11 Error Manager

The Error Manager is used by the IO Handler and Stack Machine for registering errors. Additionally, the State Machine checks the Error Manager for any registered errors and enters the error state if given.

8.1.2.1.12 Conclusion

The Memory Manager and IO Handler encapsulate outside influences such as the memory and physical peripheries from the core algorithm of the interpreter, by doing so the conclusions of [8.1.2] are accounted for. The states run by the state machine organize the interpretation of the LCF Instructions by using the Instruction Executor, Stack Machine, IO Reader and IO Writer.

8.1.2.2 State Machine

The [Figure 14 LCF Interpreter State Machine] visualizes all the states implemented within the target SWD Device [4.3].

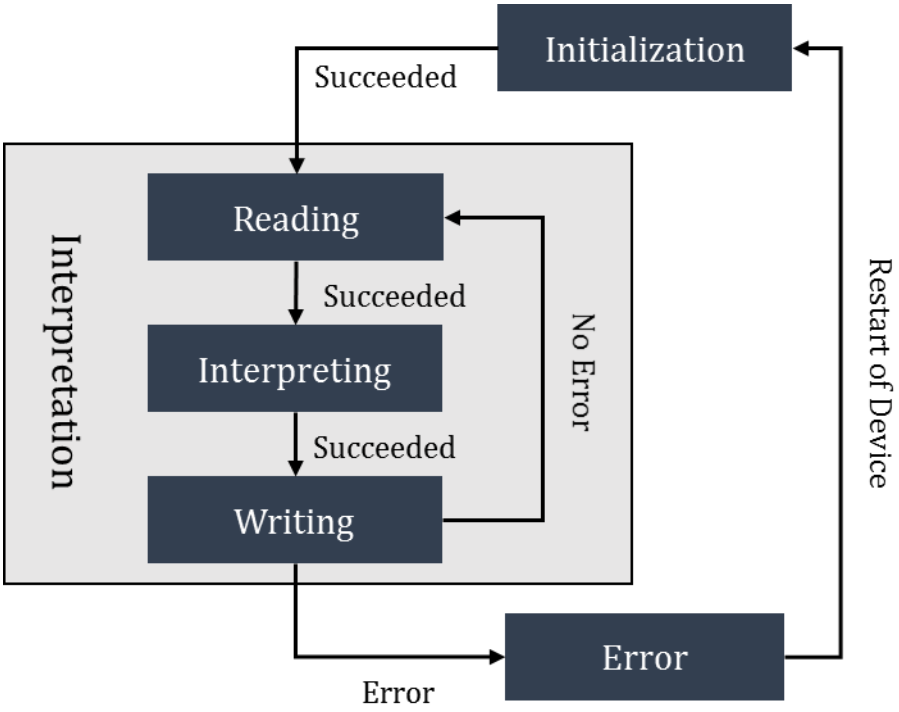


Figure 14 LCF Interpreter State Machine

Following is referring to the components stated in [8.1.2.1].

Directly after powering the target SWD Device the initialization state is entered, in which the IO Handler and Memory Manager are initialized.

After a successful initialization, the device will enter the interpretation modus in which it will loop through the states of reading, interpreting and writing. Where the IO Reader is used for reading physical inputs, the Stack Machine, Instruction Executor and Memory Manager for interpreting LCF Instructions and the IO Writer for propagating the computed results to the physical outputs.

The error state is entered in case of an error within the interpretation or initialization. The device can only recover from an error by a manual restart, since there are no physical peripheries available for that purpose (Appendix_E).

8.1.2.3 Interpreting Cycle

[Figure 15 LCF Interpreting Cycle] relates to the interpreting mode described in [Figure 14 LCF Interpreter State Machine]. It describes the interpreting procedure and makes references to the used components [8.1.2.1] in the process.

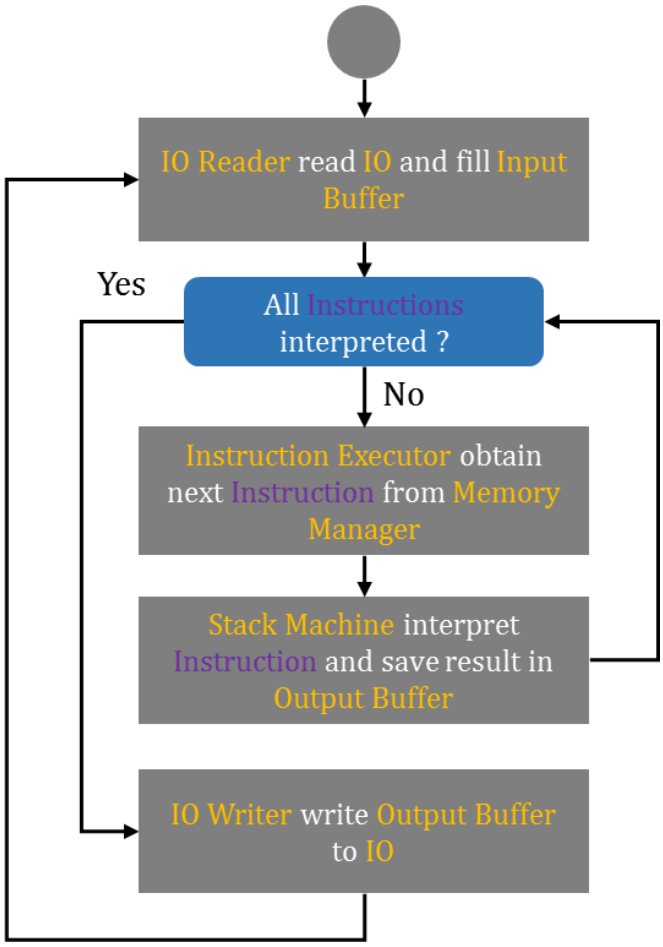


Figure 15 LCF Interpreting Cycle

Describing [Figure 15 LCF Interpreting Cycle], the IO Reader will read all inputs of the target SWD Device by using the IO Handler into its Input Buffer. Then the Instruction Executor passes successive instructions to the Stack machine until all instructions are computed and the loop exits. In the end the IO Writer will write the Output Buffer to the IO's of the target SWD.

8.2 Potential Implementation of Time Deterministic Behaviour

The implementation of a time deterministic behaviour as described in [7.1.3] can be achieved by implementing a watchdog in the LCF Interpreter and wrapping it around the interpretation cycle defined in [8.1.2.3]. However, it must be ensured that each loop within the interpretation cycle implements a call to the watchdog, in order to avoid unintentional infinite loops and ensuring a prompt termination of the interpretation in case of a time constraint violation.

[Figure 16 Interpretation Cycle with Watchdog] visualizes the implementation of a watchdog within LCF Interpreter using the [Figure 15 LCF Interpreting Cycle] as basis.

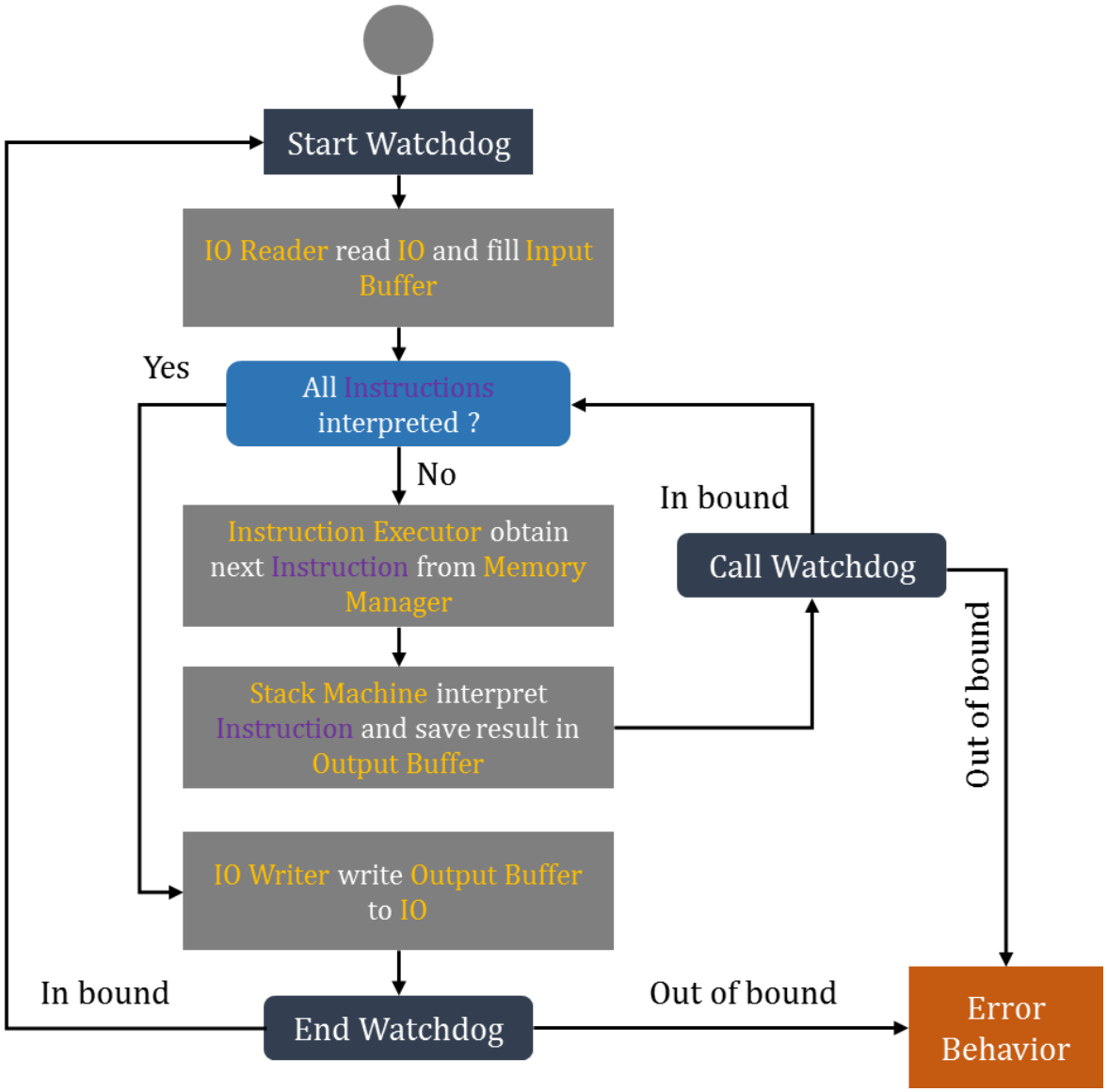


Figure 16 Interpretation Cycle with Watchdog

8.3 Prototype Deployment

The firmware is deployed onto the prototype using a ASIC2 proprietary compiler over IAR [4.4]. The LCF Instructions can be loaded and transferred by using a SWD Programmer which is able to access the flash memory of the ASIC2 over the SWD bus.

[Figure 17 Prototype Demonstrator] is photograph of the LCF Interpreter demonstrator stand which implements the LCF Interpreter on the [EUE5-SWD-4D-4D] prototype along with four LEDs and four switches used for visualization purposes.

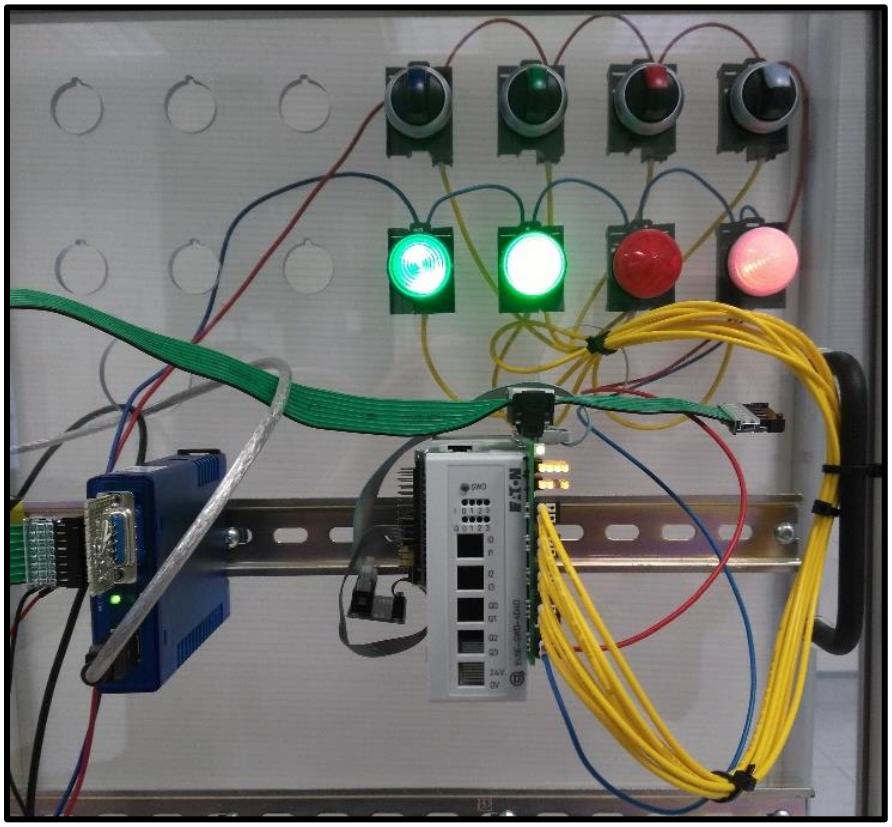


Figure 17 Prototype Demonstrator

The LCF Interpreter source code be found in the accompanying artefacts under the directory “source/lcf” together with the Doxygen documentation and all appendixes.

8.4 Automated Test System

This chapter relates to the deliverable [D06] and will analyse and describe the system architecture of an automated test system designed for undertaking automatic tests of the LCF Interpreter embedded in the target SWD Device.

8.4.1 Analysis

Validation results can be achieved by designing an automated test system which is able to execute black box testing by utilizing the same peripheries as the systems under test would utilize within its production environment. (ISTQB, 2011)

The [Table 14 Test System Abilities] describes abilities needed by the test system and identifies solutions for enabling the abilities.

ID	Ability	Solution
A01	The test system is able to load new LCF Instruction onto the target SWD Device.	The loading of new LCF Instructions is done using the LCF Loader which is located on the computer, thus an application able to utilize the LCF Loader is necessary.
A02	The test system is able to write & read onto the inputs of the target SWD Device.	The target SWD Device [4.3] is specified as IO module with four digital inputs and four digital outputs. The reading and writing of such can be achieved by a second IO module (Test IO) controlled by a PLC (Test PLC).
A03	The test system is able to validate the target SWD Device based on a test scripts.	The validation of the target SWD Device will require an application able to control the test system and while interpreting the test script.

Table 14 Test System Abilities

Based on [Table 14 Test System Abilities] following components were identified in [Table 15 Test System Components].

Component	Responsibility
Test Master	Application located on the computer which manages the test procedure of initiating, executing and validating
Test Script	The script defining the parameters of a single test.
Test PLC	A PLC connected by the computer and controlled by the Test Master.
Test IO	IO module matching the target SWD Device under test and controlled by Test PLC.
Test Result	A log file which records the test results.
LCF Loader	Will load given instruction to the target SWD Device.

Table 15 Test System Components

8.4.2 Topology

Based on the analysis [8.4.1] the topology in [Figure 18 Test System Topology] can be drawn.

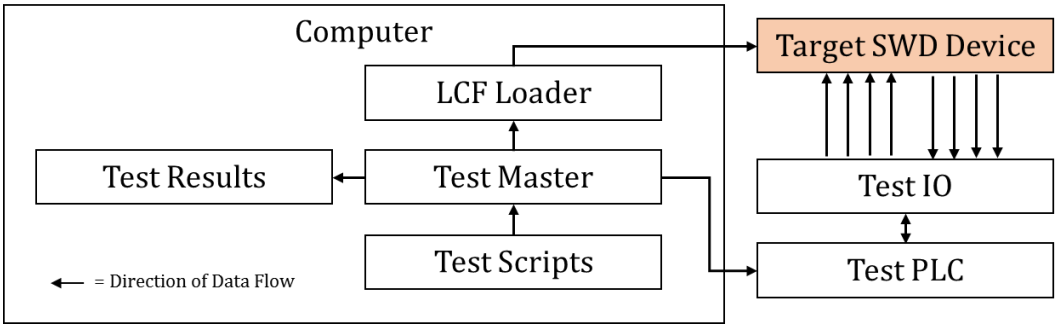


Figure 18 Test System Topology

8.4.3 Test Procedure

Relating to the topology [8.4.2] the procedure of a single test would be executing in following fashion:

1. The Test Master parses the Test Script
2. The Test Master will load the defined LCF Instructions onto the Target SWD Device by using the LCF Loader
3. The Test Master will execute the test by accessing the Test PLC and manipulating the peripheries of the Test IO.
4. The Test Master will retrieve the received values of the Test IO.
5. The Test Master will validate the retrieved values with the test expectations defined in the Test Script.
6. The Test Master will log the test results in the Test Results.

The steps from 1-6 might be executed repeatedly while incrementing over a set of Test Scripts.

8.4.4 Technology

This chapter will specifies the technology necessary for implementing the test system based on the test procedures [8.4.3] and components [8.4.2] by utilizing Eaton products as much as possible.

The [Table 16 Test System Technology] list the components while matching them with the necessary technology used for implementing them.

ID	Technology	Rationale
Test IO	EU4E-SWD-4D4D	Using the same type of IO module which matches the target SWD Device is simple and convenient.
Test PLC	XC152, Codesys Runtime	XC152 is Eaton PLC with the lowest cost able to control the Test IO. XC152 uses the Codesys Runtime to function.
LCF Loader	Python, SWD PROG2	The LCF Loader already exists.
Test Master	Codesys IDE, IronPython	Codesys IDE is needed for programming and using the Test PLC. IronPython can access and control the Codesys IDE programmatically.
Test Result	Text File	Simple text files are easy to parse and draft.
Test Script	Text File	Simple text files are easy to parse and draft.

Table 16 Test System Technology

9 LCF Interpreter Functional Testing

This sections summaries the functional testing approach and results of the implemented LCF Interpreter [8.1]. The functional testing is mainly concerned with the interpretation of LCF Instructions and the proper use of IOs as defined in the requirements [5.1].

9.1 Approach

Due to the nonexistence of an automated test system [8.4] the functional testing was executed manually by defining and executing a certain amount of test cases with the aim of proofing the implementation of being functional according to the requirements [5.1]. However, exhaustive testing is not feasible in a manual fashion due to the high amount of possible combinations as input, therefore an automated test system as defined in [8.4] would be required for reaching a higher testing coverage.

To approximate the testing procedure as close as possible to a real production environment the black box testing approach was applied.

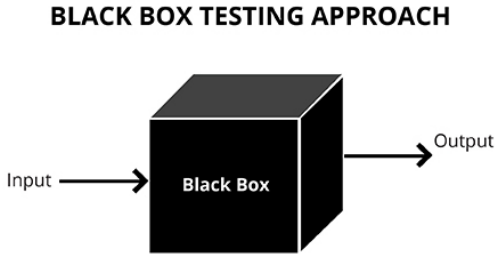


Figure 19 Black Box Testing

Black box testing is an approach where the system under test and its implementation is unknown to the tester and testing is conducted by feeding a set of inputs to the system while expecting a defined output as result. (ISTQB, 2011)

In case of the LCF Interpreter, LCF Instructions were considered as inputs and the resulting behaviour of the IOs as outputs.

9.2 Test Case Definition

In order to accommodate the chosen testing approach [9.1] each test case will define the input and expected output of the system and the requirement it covers. [Table 17 Test Case Exampl] is an exemplary test case visualizing how test cases are defined stating the LCF Instructions as input and a truth table as output.

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T01</td><td>T01</td><td>Tests if Q0 can be set by AND.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX1 AND %IX0;</td><td>0x07 0x06 0x01 0x00 0x01 0x01 0x05 0x02 0x00</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

Table 17 Test Case Example

9.3 Test Execution

For each test execution the used tools and environment must be noted to ensure reproducibility of the test results. That includes things such as used hardware, software version, environmental temperature, date, place etc.

The execution of a test case uses following procedure

1. Load LCF Instruction defined by test case to powered target SWD Device [4.3] using the SWD Programmer [4.4].
2. Manipulate the inputs of the device and assess resulting output with the expectation of the tests case.
3. Note the results of the assessment and in case of failure define the reproduction steps, so that the failure can be reproduced.

9.4 Test Results

The defined test cases of the LCF Interpreter were successfully executed and all test cases passed the output criteria and so with full fill the LCF Interpreter requirements [5.1]. The defined test cases and test results can be found in the (Appendix_D).

10 Validation

The validation of the project implementation is split into several subcategories, so that a higher assessment coverage can be achieved. The validation will assess the requirements, deliverables, objectives and finally the whole aim of the project.

10.1 Requirement Assessment

Following tables assess the defined requirements in [5] regarding of their implementation, by stating the status and placing the references for each requirement.

Req.	Status	Assessment	Reference
R01	Done	LCF Instruction are parsed.	[9]
R02	Done	LCF Interpreter can interpret instructions as defined.	[9]
R03	Done	LCF Interpreter is functional from the start up.	[9], [8.1.2.2]
R04	Done	LCF Interpreter is interpreting cyclical.	[9], [8.1.2.3]
R05	Done	LCF Interpreter works in a loop.	[9], [8.1.2.3]
R06	Done	LCF Interpreter can use target SWD Device inputs.	[9], [8.1.2.1]
R07	Done	LCF Interpreter inputs are read in the beginning of each cycle.	[9], [8.1.2.3]
R08	Done	LCF Interpreter can use target SWD Device outputs.	[9], [8.1.2.1]
R09	Done	LCF Interpreter outputs are written in the end of each cycle.	[9], [8.1.2.3]

Table 18 D01 & D02 Requirement Assessment

Req.	Status	Assessment	Reference
R31	Done	Time deterministic behaviour was described.	[7.1.3]
R32	Done	Implementation proposal of time deterministic behaviour given.	[8.2]

Table 19 D03 Requirement Assessment

Req.	Status	Assessment	Reference
R41	Done	Communication of SWD described.	[7.3.1]
R42	Done	Implementation proposal of device-to-device communication given.	[7.3.4]

Table 20 D04 Device-to-Device Communication

Req.	Status	Assessment	Reference
R51	Done	IEC61131 compliance assessed.	[7.2.3.1]
R52	Done	IEC61499 compliance assessed.	[7.2.3.2]
R53	Done	Conclusion deviates from approximation, since full compliance is doubtful.	[7.2.3.2.3]

Table 21 D05 Assessing of Standards

Req.	Status	Assessment	Reference
R61	Done	Automatic testing system described.	[8.4]
R62	Done	Topology and technology described.	[8.4.2], [8.4.4]

Table 22 D06 Automated Test System

10.2 Deliverables Assessment

[Table 23 Deliverables Assessment] lists the deliverables [4.5] and describes how they are achieved.

Dev.	Status	Assessment	Reference
D01	Done	Source code was implemented.	(Appendix_L)
D02	Done	Source code applied on target SWD Device [4.3].	[8.3]
D03	Done	Section delivered within research and project implementation.	[7.1], [8.2]
D04	Done	Section delivered.	[7.3]
D05	Done	Conclusion and table assessment delivered.	[7.2], (Appendix_N), (Appendix_O)
D06	Done	Topology and concept delivered.	[8.4]

Table 23 Deliverables Assessment

10.3 Objectives Assessment

[Table 24 Objective Assessment] lists the objectives [4.2] and assess their achievement.

Obj.	Status	Assessment	Reference
OBJ01	Done	The source code was implemented on the ASIC2 platform. Application to the target SWD Device is possible.	(Appendix_L) [8.3]
OBJ02	Done	LCF Interpreter prototype is done and the LCF System can be showcased.	[8.3]
OBJ03	Done	The suggested implementation for the time deterministic behaviour allows for a direct execution.	[8.2]
OBJ04	Done	The identified device-to-device communication defines and guides the next steps for the implementation.	[7.3]
OBJ05	Done	The conclusion of the standard assessment allows the product management to act upon.	[7.2]
OBJ06	Done	The delivered topology and concept define the implementation of an automatic testing system.	[8.4]

Table 24 Objective Assessment

10.4 Aim Assessment

Referring to [4.1], the implemented LCF Interpreter is completing the first version of the LCF Interpreter which allows it to be demonstrated. Additionally, the executed research defines the next steps to take for the LCF development and greatly contribute to future version of the LCF System.

10.5 Client Validation

The produced artefacts were presented to the client [2.1] on the 02.04.2019 in EEIC (Bořivojova 2380, 252 63 Roztoky, Czechia) to validate the user acceptance and gather feedback. In general the LCF Interpreter was perceived positively and several suggestions for further enhancements were gathered [11].

11 Further Enhancements

Based on the executed research the LCF system can be enhanced by implementing a time deterministic behaviour [7.1] and device-to-device communication [7.3]. Additionally, with the development of the automatic test system the development cycle can be shortened due to faster validation and release processes of the target SWD Device [4.3].

The development of the LCF System only begun and for a feasible commercialization it must implement a variety of features. Including utilities such as debugging, the simulation of LCF Code and the implementation of new functionalities such as timers, loops, triggers, variables, int, float, analog to digital converters etc. Thus, the next step for the project would be to create a prioritized feature roadmap which defines a minimal viable product as target.

12 Deviations from Project Proposal

Following are deviations of the computing project proposal submitted on the 16.11.2018 compared with the actual execution of the project. (Appendix_A)

12.1 Definition of LCF Code

In ([1.6.1]Appendix_A) of the project proposal, it is stated that the definition of the LCF instructions (Code) for the LCF Interpreter will be part of the executed research, however the definition of the LCF instruction was concluded with the finalization of the LCF Compiler which happened in the midst of the execution of this computing project.

12.2 Automated Test System

The design of an automated tests system [8.4] was not part of the project proposal. It was included within the computing project due it's apparent need and for compensating the missing effort introduced by not defining the LCF Code [12.1].

13 Research Ethics

Refereeing to the computing project proposal ([1.8] Appendix_A) and research ethics guidelines of the Teesside University (Teesside University, 2014 - 2015) this chapter will state how the identified problems were met.

1. The produced report was checked by Natalia Estefania Juardo Espin 05.04.2019 for any subtle company advertisement.
2. Due to the agreement with Eaton the disclosed work will not be disputed, since protected technology is not revealed.

14 Conclusion

Based on the validation [10] and executed functional test cases [9] the project can be deemed as success comparing to the stated aim and objectives [4]. Additionally, the research [7] and feedback gathered form the clients [10.5] allowed for a divers gathering of possible enhancements [11] guiding the next steps of the LCF Project.

14.1 Critical Evaluation

- The assessment of the standards [7.2] should viewed with caution due the inexperience of the author regarding the commercialization of products complying with the standards. Thus, before taking actions based on the conclusions a person with such experience should be consulted.

- Most of used references originate from the client [2.1]. Thus, the information presented in them can be biased and exposes a positive tendency towards the client and its technology and products.

- The tests were defined and executed by the author which allows for developer biased, hence the quality of the tests could be compromised. Developer bias can only be avoided by having sperate people for developing and testing the system in order to reach independent testing. (Try QA, 2019)

14.2 Project Realization

The project plan defined in the project proposal ([1.10]Appendix_A) was mostly followed as stated only deviating regards the research of time deterministic behaviour and device-to-device communication which were in reality executed after the development of the LCF Interpreter was completed. This change to the project plan was introduced because it became apparent that knowledge of the LCF Interpreter implementation and SWD Devices will greatly influence the quality of the research outcomes.

There were also no major problems within the project realization since stakeholders and experts were always within physical reach in case of needed clarification and questions.

The defined project methodology [6.1] proved to be the correct choice due the smooth execution of the project and production of required documentation. However, it is to note that changing requirements would have introduced some troubles. Thus, the clear definition of the requirements together with the client in the beginning of the project can be accounted for the smooth project execution.

14.3 Contribution

This computing project greatly contributes to the initial steps towards a distributed automation system for Eaton. Knowledge produced here might be applied in several hundred thousands of devices for machine automation and may disrupt the PLC market which relies on a central automation concept.

14.4 Personal

In this computing project I have deepened my knowledge regarding industrial automation machine building as well refined my knowledge of C programming and embedded controllers. Especially, the research of the current Industry 4.0 trends and studied PLC standards elevated my understanding surrounding the machine industry.

These personal advancements put me in a central role for the LCF project and relating activities, hence my value towards the company and industry increased tremendously by this computing project.

Not to forget the joy, motivation and fulfilment experienced by this work which fuels the energy required for the next steps ahead.

15 References

3S-Smart Software Solutions GmbH, 2019. 3.3 Application Handling. In: 18, ed. *CODESYS Control V3 Manual.* s.l.:CODESYS, pp. 30-40.

Andrea, K. & Samuel, M., 2018. *BSc Project Alignment Meeting.* Roztoky: Eaton.

Arguinariz, H. & Dedourek, P., 2018. *The size of LCF Interpreter & Memory usage of EU5E-SWD-4D4D* [Interview] (16 November 2018).

Automation.com, 2015. *IIoT Protocols to Watch.* [Online]
Available at: https://www.automation.com/library/white-papers/iiot-protocols-to-watch
[Accessed 06 March 2019].

Berger, H., 2014. *Automating with SIMATIC S7-300.* 2nd ed. Erlangen: Publics Publishing.
BUDIMIR, M., 2017. *What do PLC watchdog timers do?.* [Online]
Available at: https://www.motioncontroltips.com/plc-watchdog-timers/
[Accessed 16 June 2019].

Cambridge, 2018. *Definition of "led" - English Dictionary.* [Online]
Available at: https://dictionary.cambridge.org/us/dictionary/english/led
[Accessed 15 November 2018].

Codesys, 2018. *Codesys OPC UA.* [Online]
Available at: https://www.codesys.com/products/codesys-runtime/opc-ua-server.html
[Accessed 07 March 2019].

Eaton Corporation, 2015. 2.3.5.2 Acyclic data transfer. In: H. Einwag, ed. *SmartWire-DT The System.* 53105 Bonn: Eaton Industries GmbH, p. 55.

Eaton Corporation, 2016. *MOEM.* [Online]
Available at:
http://www.eaton.com.cn/EN/EatonCNES/ProductsSolutions/Electrical/ProductsandServices/MarketSolutions/MOEM/index.htm
[Accessed 15 November 2018].
Eaton Corporation, 2017. 9.2 System. In: A. Panten-Nonnen, ed. *XC-152 Compat Controller.* 53105 Bonn: Eaton Industries GmbH, p. 72.

Eaton Corporation, 2017. *SmartWire-DT The System.* [Online]
Available at:
http://www.eaton.com/ecm/groups/public/@pub/@electrical/documents/content/mn05006002z_en.pdf
[Accessed 15 November 2018].

Eaton Corporation, 2018. *About the EEIC.* [Online]
Available at: http://www.eaton.com/Eaton/EEIC/AboutEEIC/index.htm
[Accessed 9 November 2018].

Eaton Corporation, 2018. *HMI Series.* [Online]
Available at:
http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/Automation/LegacyProducts/OperatorInterface/HMi/index.htm
[Accessed 15 November 2018].

Eaton Corporation, 2018. *Product Guide & Catalog.* [Online]
Available at: http://es.eaton.com/SmartWire-DT/catalog.php
[Accessed 06 March 2019].

Eaton Corporation, 2018. *SmartWire-DT Software: SWD-Assist.* [Online]
Available at: http://www.eaton.eu/Europe/Electrical/ApplicationSolutions/SmartWireDT/SWD-Assist/index.htm
[Accessed 06 March 2019].

Eaton Corporation, 2018. *SmartWire-DT Wiring Solutions.* [Online]
Available at:
http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/Connectivity/index.htm?wtredirect=www.eaton.com/smartwire-dt
[Accessed 12 November 2018].

Eaton Corporation, 2019. *Homepage.* [Online]
Available at: https://www.eaton.com/us/en-us.html
[Accessed 01 March 2019].

Eaton Industries GmbH, 2018. *ASIC2 Datasheet.* 1st ed. Bonn: Eaton Industries GmbH ICPD-E-S&S-PD.

European Committee for Electrotechnical Standardization, 2012. *IEC 61499-1.* 2.0 ed. Brussels: IEC.

European Committee for Electrotechnical Standardization, 2013. *IEC 61499-4.* 2.0 ed. Brussels: IEC.

General Electrics, 2019. *Everything you need to know about the Industrial Internet of Things.* [Online]
Available at: https://www.ge.com/digital/blog/everything-you-need-know-about-industrial-internet-things
[Accessed 06 March 2019].

Heribert Einwag, -. S. P. M., 2018. *The Future of SWD* [Interview] (2 November 2018).

Hermann, M., Pentek, T. & Otto, B., 2016. *Design Principles for Industrie 4.0 Scenarios,* 49th Hawaii International Conference on System Sciences: IEEE.
International Electrotechnical Commission, 2003. *IEC 61131-1.* 2.0 ed. Brussels: IEC.

International Electrotechnical Commission, 2017-2018. *IEC 61131-2.* 4.0 ed. Geneva: IEC.

ISTQB, 2011. *Foundation Level Syllabus.* ISTQB 2011 ed. s.l.:ISTQB.

Klingenberg, C. O. & Antunes, J. A. d. V., 2017. *Industry 4.0: what makes it a revolution? ,* São Leopoldo: Polytechnic School, Universidade do Vale do Rio dos Sinos.
Koopman, P., 1989. *A GENERIC STACK MACHINE.* [Online]
Available at: https://users.ece.cmu.edu/~koopman/stack_computers/sec3_2.html
[Accessed 04 April 2019].

Magdolinic, R., 2018. *Limitations of SWD* [Interview] (15 November 2018).

Markus, S., 2018. *Distributed Automation Use Case* [Interview] (01 March 2018).

Martin, R. C., 2008. *Clean Code.* 1 ed. s.l.:Pearson Education.

Petters, S. M., 2007. *Real-Time Systems.* 2007/2008 ed. s.l.:NICAT.

Rikin, B., 2015. *IEC STANDARDS FOR PLC,* Ahmedabad, India: Nirma University.

Sestoft, P. & Tinelli, C., 2016. *A stack machine for micro-C compiling micro-C to stack machine code,* s.l.: University of Iowa.

Teesside University, 2014 - 2015. *Policy, Procedures and Guidelines for Resreach Ethics.* 2014-15 ed. Middlesbrough: Teesside University.

Thramboulidis, K., 2013. *IEC 61499 vs. 61131: A Comparison Based on Misperceptions,* Pratas, Greece: University of Patras.

Try QA, 2019. *What is independent testing? It's benefits and risks.* [Online]
Available at: http://tryqa.com/what-is-independent-testing-its-benefits-and-risks/
[Accessed 05 April 2019].

Uhlemanna, T. H.-J., Lehmann, C. & Steinhilper, R., 2017. *The Digital Twin: Realizing the Cyber-Physical Production System for,* Bayreuth: ScienceDirect.

UNITRONICS, 2018. *What is the definition of "PLC"?.* [Online]
Available at: https://unitronicsplc.com/what-is-plc-programmable-logic-controller/
[Accessed 12 November 2018].

wisdomjobs, 2018. *FUNCTION BLOCKS PROGRAMMABLE LOGIC CONTROLLERS.* [Online]
Available at: https://www.wisdomjobs.com/e-university/programmable-logic-controllers-tutorial-523/function-blocks-14686.html
[Accessed 28 March 2019].

16 Figures & Tables

17 Appendixes

Appendix	Name	Location
Appendix_A	Project Proposal (Local Control Function Interpreter)	Included in report.
Appendix_B	System Presentation of Smart Wire	Included in the artefacts.
Appendix_C	Basics of SWD	Included in the artefacts.
Appendix_D	Functional Tests and Test Results (LCF Cheesecake 0.35 – Test Case Definition)	Included in report.
Appendix_E	EU5E-SWD-4D4D IO Module	Included in the artefacts.
Appendix_F	LCF Project Presentation	Included in the artefacts.
Appendix_H	LCF Instruction Definition (LCF Compiler Commands & Instruction Set)	Included in report.
Appendix_J	EU5E-SWD-4D4D Schematic	Included in the artefacts.
Appendix_N	IEC61131 Assessment (LCF IEC61131)	Included in report.
Appendix_O	IEC61499 Assessment (LCF IEC61499)	Included in report.

Computing Project (BSc)

Project Proposal

Author:	Samuel A. Marti
Place:	Prague College
Module:	Computing project (BSc)
Module Code:	CPJT-600-1804
Module Leader:	Bohus Ziskal, Julian Warren
Date:	16.11.2018
Semester Code:	1804 - 1901
Word Count:	2188

Proposing:

1 Local Control Function Interpreter

Contents

1.1 Glossary & Abbreviations

This glossary defines distinct terms used within this document.

Term	Definition
EEIC	Eaton European Innovation Center, a branch of Eaton Corporation. ([0] Eaton Corporation, 2018)
LCF	Local Control Functions, a Eaton project executed within EEIC.
SWD	SmartWire DT, a proprietary Fieldbus System by Eaton. ([8] Eaton, 2018)
SWD Device	A device/component able to use SWD as Communication Fieldbus.
ASIC/ASIC2	Proprietary microcontroller of Eaton used within SWD Devices.
PLC	Programmable Logic Controller. ([5] UNITRONICS, 2018)
MOEM	A market evolved around Micro-Opto-Electro-Mechanical systems. ([9] Eaton, 2016)
HMI	Human Machine Interface. ([10] Eaton, 2018)
LED	Light Emitting Diode. ([12] Cambridge, 2018)

Table 1 Glossary & Abbreviations

1.2 Introduction

Following document describes the project proposal *"Local Control Function (LCF) Interpreter"* as the BSc-Computing Project of Samuel A. Marti, student at Prague College and Teesside University.

The proposed project would be executed with the clientship of the *"Eaton European Innovation Center (EEIC)"* which is located in Roztoky, Czechia. That clientship is ensured due the general employment of the assignee by EEIC, thus required resources and guidance of the proposed project are warranted. ([1] Andrea & Samuel, 2018)

1.3 Project Type

The proposed project would require the handling and programming of an embedded experimentation board which is hosting a proprietary microcontroller named *"ASIC2"* which can be programmed using the "C" language. ([3] Eaton Industries GmbH, 2018)

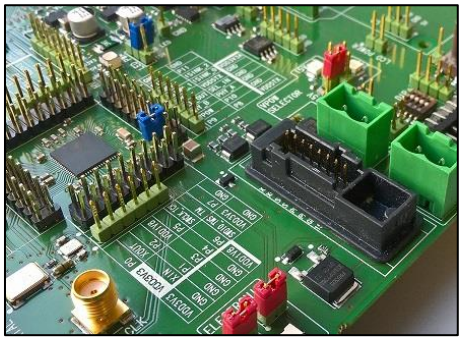


Figure 1 ASIC2 Evalboard

1.4 Project Background

Eaton owns a proprietary fieldbus named *"Smart Wire DT (SWD)"*. This fieldbus allows *"Micro-Opto-Electro-Mechanical (MOEM)"* customers of Eaton to easily to interconnect and control varies types of industrial devices, such as: Circuit Breakers, Motor Starters, I/O Modules, Frequency Drives, Bush Buttons, Switches, LED's, HMI, PLC etc. All SWD Devices implement a ASIC/ASIC2 in order to communicate with the SWD fieldbus system. ([11] Eaton, 2017)

1.4.1 Problem Statement

For some applications SWD does not provide a fast enough "reaction time", that is caused by the time needed for computing and replying to a request, that time is based on the cycle time of the SWD bus and the PLC.

The fastest cycle time of SWD is 3ms with 20byte input and 20byte output data ([13] Eaton, 2015). The fastest cycle time of an SWD capable PLC such as XC-152 is 4ms for 1k byte of instructions ([15] Eaton, 2017). Following formula calculates the total time needed for a request of the target device to reach the PLC and back.

$$ReactionTime_{Worst} = 4 * \text{CycleTimeSWD} + 2 * \text{CycleTimePLC} = 4 * 3\text{ms} + 2 * 4\text{ms} = 20\text{ms}$$
$$ReactionTime_{Best} = 2 * \text{CycleTimeSWD} + \text{CycleTimePLC} = 2 * 2\text{ms} + 2\text{ms} = 10\text{ms}$$

E.G the control of a variable frequency drive would require a reaction time of less than 10ms, according to a Controls Engineer at Eaton ([16] Magdolinic, 2018). This "speed" limitation of SWD leads to the situation where certain markets cannot be entered by the system.

1.4.2 Solution

EEIC is developing a new distributed automation system for industrial components with the project name *"Local Control Function (LCF)"*. The project will allow users to program control functionalities directly onto the target device and so with bypass the PLC and SWD, that will ultimately reduce the reaction time of the system by cutting the cycle times of SWD and the PLC for those control functionalities. Thus LCF tackles the problem stated in [1.4.1].

1.4.3 Additional Feature

LCF not only tackles the problem stated in [1.4.1], but will also enable new possibilities in regards to distributed automation and thus is a major initiative in the eyes of the SWD Product Management ([14] Heribert Einwag, 2018).

1.4.4 LCF Project Organization

The LCF Project is aggregated by four sub-projects: LCF Editor, LCF Compiler, LCF Loader, LCF Interpreter. The proposed Computing Project BCs would be solely concerned with the LCF Interpreter and is thus clearly distinguishable from other parts of the whole LCF Project.

Following [Figure 2] describes the whole LCF System with its subprojects and their relationship on an abstract level.

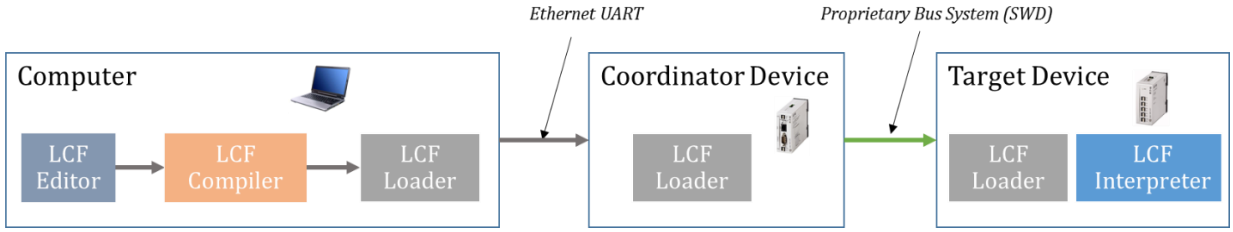


Figure 2 LCF Overview

Referring to [Figure 2] the user of the whole LCF System will be able to design a logical control function in the LCF Editor and generate so called *"LCF Code"*. The compiler will then compile the LCF Code into *"LCF Binary Code"* which is then transported using the LCF Loader onto the target device on which the LCF Binary Code will be interpreted by the LCF Interpreter.

1.5 First Use Case & Target Device

By coordinating with business partners, engineering managers and SWD product management, it was agreed that the first SWD Device implementing the LCF System would be a 4 digital input & 4 digital output module, due following reasons:

- ASIC2 based hardware is available.
- Could be used as standalone controller.
- Clear requirements and objectives.
- Comprehensible value proposition, which can be used for showcases.

An industrial input/output module which is implementing the LCF System will gain a clear advantage in terms of speed and reaction time over comparable products which need to be controlled by an external controller (PLC), due the localization of the control functions from the controller towards the target device.

1.6 Project Scope

The herewith proposed Computing Project BCs is concerned with the implementation, research and evaluation of the embedded interpreter (LCF Interpreter) located on the target device as described in [1.4.4] and [1.5].

1.6.1 Research

The proposed project will be concerned with research into the following topics:

- Definition of LCF Code for accommodating a digital input/output module.
- Time-deterministic behaviour and its potential implementation into the LCF interpreter.
- Device to device communication concepts using SWD as base.
- PLC and micro PLC standards such as IEC61131, IEC61499 and their potential implementation.

1.6.2 Development

The proposed project will include the development of the LCF Interpreter which is accommodating the defined target device [1.5] on the ASIC2 Evalboard [Figure 1]. The developed LCF Interpreter shall be packaged as reusable development library, so that it can be easily applied on other SWD Devices. Additionally the architecture of the LCF Interpreter shall be designed in such a way that it can accommodate further improvements based on the research results defined in [1.6.1].

1.6.3 Deliverables

1. Research Results as defined in [1.6.1].
2. Embedded LCF Interpreter Library as defined in [1.6.2].

1.6.4 Exclusions

- The proposed project does not include any other subproject as mentioned in [Figure 2] except the LCF Interpreter.
- The implementation of the LCF Interpreter shall work functionally but must not satisfy all PLC standard's, such as IEC 61131.

1.6.5 Project Contribution

The deliverables will contribute to following topics:

- Complement the LCF System with the LCF Interpreter.
- Will allow Eaton to apply the developed LCF Interpreter on any type of target device utilizing the ASIC2. E.G: Push Buttons, LED, Switch, Circuit Breakers, Motor Starters, Frequency Drives etc.
- The research results will provide a potential implementation assessment of:
    - Device to device communication.
    - Time-deterministic behaviour.
    - Relevant Standards.
- The LCF System will be able to be demonstrated at trade fairs such as the SPS IPC Drives in Nürnberg and to business partners of EEIC/Eaton.

1.7 Methodology

Since the proposed project and its scope is clear, the requirements can be drafted easily and changes are unlikely to happen. As such this project is going to be executed in the traditional waterfall model. Waterfall provides a lightweight framework made for projects where time and scope are defined and documentation is important, unlike agile, scrum or KANBAN which excel in environments with changing scope and long running projects. ([6] Lotz, 2013)

1.8 Research Ethics

Referring to the Teesside University research governance document ([7] Teesside University, 2014 - 2015), following describes relevant issues identified based on the research ethics principles of Teesside University.

1. Due to the collaboration of the assignee and EEIC; the produced work is under threat of being used for biased company advertisement, thus any resulting work shall be checked for its objectivity and integrity by a third party (To be defined).
2. There is a potential conflict of interest in regards to publication due to the collaboration with EEIC. Thus an agreement with EEIC was made in which the publications of the resulting research shall be undisputed as long it does not disclose technical details of protected technologies of Eaton or produced source code.

1.9 Project Feasibility

In order to ensure the feasibility of the proposed Computing Project BCs, it was necessary to research if the given memory size of the ASIC2 would be sufficient for the project.

The ASIC2 provides 2 flash drives with 40kB + 88kB ([3] Eaton Industries GmbH, 2018). The first flash drive is used for storing the SWD communication stack and the second for storing the application code, the LCF Interpreter and the LCF Binary Code. According to ([17] Arguinariz, 2018) the current application code for an I/O module such as described in [1.5] will account for maximal 50kB. Thus the LCF Interpreter and LCF Binary Code will need to fit into 38kB.

Since the architecture of LCF Interpreter is yet undefined a useful estimation of its potential size is not possible to be conducted, however according to SWD firmware developers at Eaton ([18] Arguinariz & Dedourek, 2018) “38kB are more than sufficient”.

1.10 Project Plan

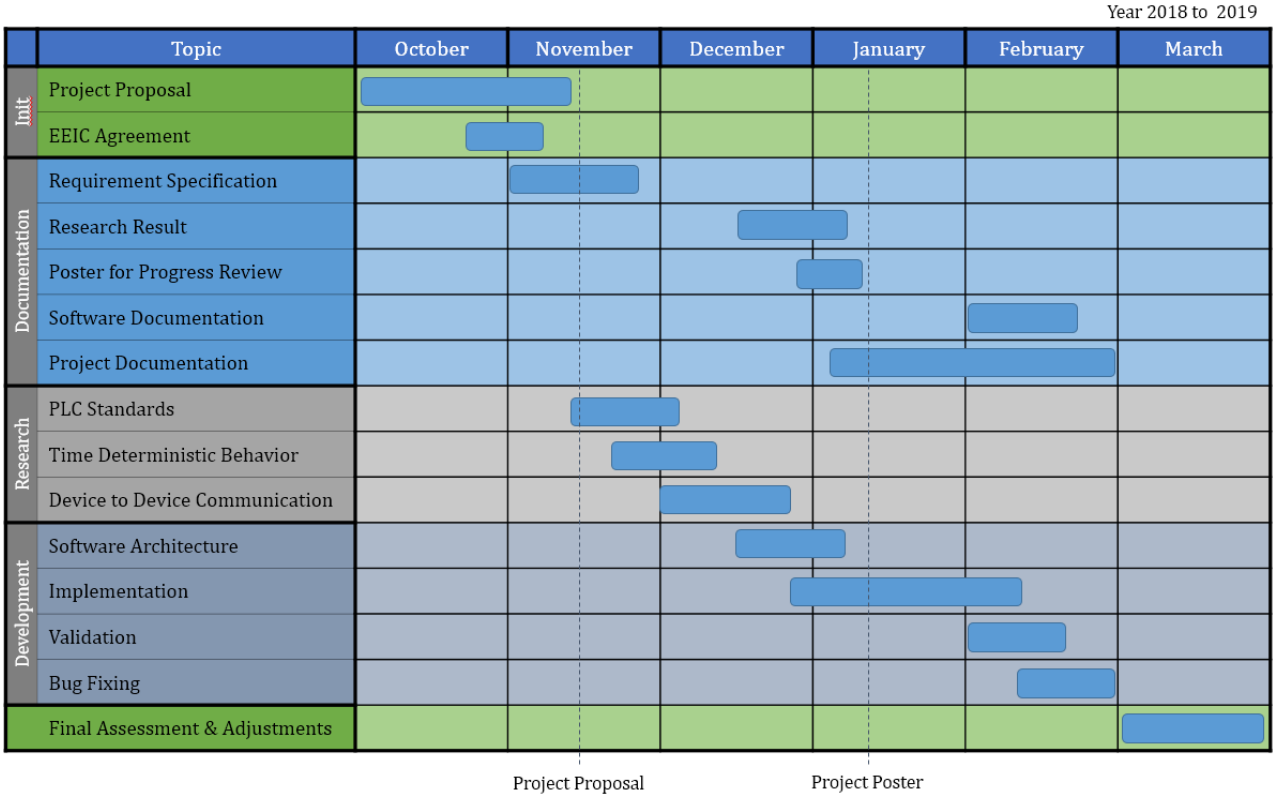


Figure 3 Project Plan

As described in [Figure 3] after the project proposal submission the main focus lays on the research of the described topics since they might influence the software architecture design. After that the implementation follows in a traditional waterfall like manner, meanwhile the documentation is executed in parallel for each corresponding item.

The month of March is allocated as buffer for refinements of artefacts and documents.

The weekly effort estimated for executing the proposed project plan in [Figure 3] is estimated between 18 – 24 working hours, which could add up to 432-576 working hours until the end of March.

1.11 Personal Objective

I have a personal passion for industrial automation and machine building, more so I am intrigued by being able to control the whole system of a machine. Thus by developing the LCF Interpreter I will learn architectures and the standards needed for creating a true “pico” PLC with the potential for setting the conceptual cornerstone for one of the first fully distributed automation systems.

The gained knowledge and skills will not only elevate me as a valuable member of the industrial automation community but will also make me essential for further developments of LCF within EEIC and so will contribute directly to my career advancement as an engineer and further my goal of being a technology leader within the industry.

2 References

[0] Eaton Corporation, 2018. *About the EEIC.* [Online]
Available at: http://www.eaton.com/Eaton/EEIC/AboutEEIC/index.htm
[Accessed 9 November 2018].
[1] Andrea, K. & Samuel, M., 2018. *BSc Project Alignment Meeting.* Roztoky: Eaton.
[10] Eaton, 2018. *HMi Series.* [Online]
Available at:
http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/Automation/LegacyProducts/OperatorInterface/HMi/index.htm
[Accessed 15 November 2018].
[11] Eaton, 2017. *SmartWire-DT The System.* [Online]
Available at: http://www.eaton.com/ecm/groups/public/@pub/@electrical/documents/content/mn05006002z_en.pdf
[Accessed 15 November 2018].
[12] Cambridge, 2018. *Definition of "led" - English Dictionary.* [Online]
Available at: https://dictionary.cambridge.org/us/dictionary/english/led
[Accessed 15 November 2018].
[13] Eaton, 2015. 2.3.5.2 Acyclic data transfer. In: H. Einwag, ed. *SmartWire-DT The System.* 53105 Bonn: Eaton Industries GmbH, p. 55.
[14] Heribert Einwag, -. S. P. M., 2018. *The Future of SWD* [Interview] (2 November 2018).
[15] Eaton, 2017. 9.2 System. In: A. Panten-Nonnen, ed. *XC-152 Compat Controller.* 53105 Bonn: Eaton Industries GmbH, p. 72.
[16] Magdolinic, R., 2018. *Limitations of SWD* [Interview] (15 November 2018).
[17] Arguinariz, H., 2018. *Memory Usage of SWD I/O Module* [Interview] (16 November 2018).
[18] Arguinariz, H. & Dedourek, P., 2018. *The size of LCF Interpreter* [Interview] (16 November 2018).
[18] Wikipedia, 2018. *Stack machine.* [Online]
Available at: https://en.wikipedia.org/wiki/Stack_machine
[Accessed 16 November 2018].
[3] Eaton Industries GmbH, 2018. *ASIC2 Datasheet.* 1st ed. Bonn: Eaton Industries GmbH ICPD-E-S&S-PD.
[4] Marti, S. A., 2018. *Nutshell of Local Control Functions,* Roztoky: Eaton European Innovation Center.
[5] UNITRONICS, 2018. *What is the definition of "PLC"?.* [Online]
Available at: https://unitronicsplc.com/what-is-plc-programmable-logic-controller/
[Accessed 12 November 2018].
[6] Lotz, M., 2013. *Waterfall vs. Agile: Which is the Right Development Methodology for Your Project?.* [Online]
Available at: https://www.seguetech.com/waterfall-vs-agile-methodology/
[Accessed 12 November 2018].
[7] Teesside University, 2014 - 2015. *Policy, Procedures and Guidelines for Resreach Ethics.* 2014-15 ed. Middlesbrough: Teesside University.
[8] Eaton, 2018. *SmartWire-DT Wiring Solutions.* [Online]
Available at:
http://www.eaton.com/Eaton/ProductsServices/Electrical/ProductsandServices/AutomationandControl/Connectivity/index.htm?wtredirect=www.eaton.com/smartwire-dt
[Accessed 12 November 2018].
[9] Eaton, 2016. *MOEM.* [Online]
Available at:
http://www.eaton.com.cn/EN/EatonCNES/ProductsSolutions/Electrical/ProductsandServices/MarketSolutions/MOEM/index.htm
[Accessed 15 November 2018].

3 Figures & Tables

LCF Cheesecake 0.35 - Test Case Definition

EEIC Electronics Group

Exported on 04/04/2019

Table of Contents

1 Document Specification

Target release	LCF 0.35 - Cheescake[1]
Document status	**SUGGESTED**
Document owner	Marti, Samuel[2]
Reviewer	Horn, Michal[3], Kovbasjukova, Oxana[4]
Document Purpose	Defines the test cases for the target release.

1 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
2 https://confluence-prod.tcc.etn.com/display/~E9955465
3 https://confluence-prod.tcc.etn.com/display/~E9990291
4 https://confluence-prod.tcc.etn.com/display/~C9930424

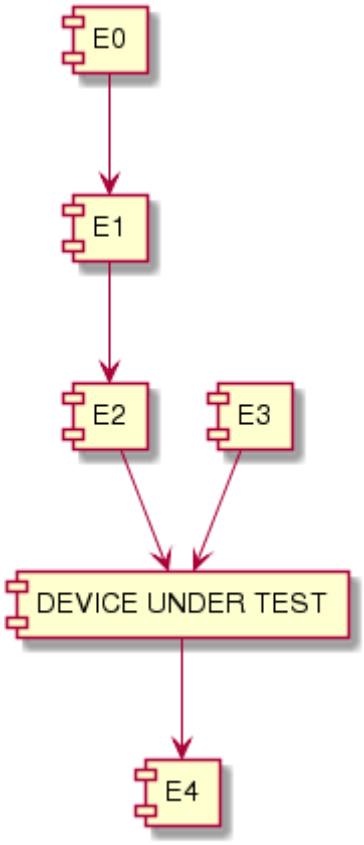
2 Test Equipment

Following equipment is required for executing tests.

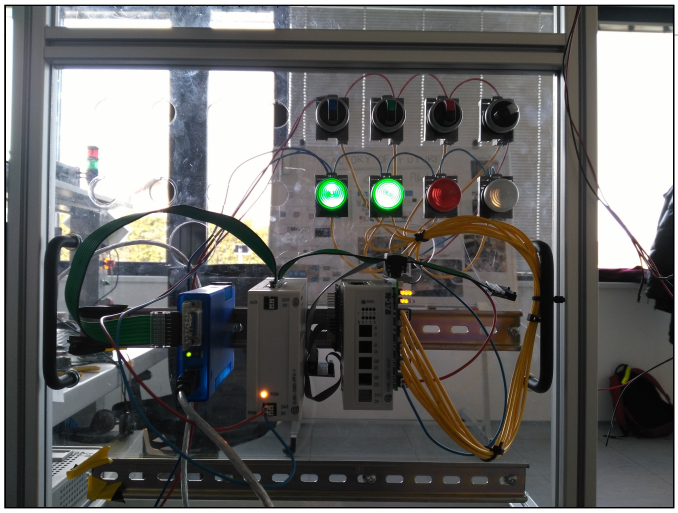
ID	Equipment	Purpose	Reference
E0	SWD PROG	Software used for loading LCF Instructions to the target Device.	http://pxgit.raleigh.software.ch.etn.com/ASIC2_Doc/doc/html/howto_swd_prog2.html
E1	Interface USB <> RS232/RS422/RS485 Industry	D[5]eviec used by SWD PROG and SmartWire DT	https://www.wut.de/e-38211-ww-daus-000.php
E2	EU5C-SWD-PF2-1	Used as power relay for powering the SWD PROG and the device under test.	https://datasheet.eaton.com/datasheet.php?model=116380&locale=en_GB
E3	4x M22-K01 Switches	For manipulating the inputs of the device under test.	https://datasheet.eaton.com/datasheet.php?model=216378&locale=en_GB
E4	4x M22-LED	For indicating the output status of the device under test.	https://datasheet.eaton.com/datasheet.php?model=216557&locale=en_GB

5 https://www.wut.de/e-38211-ww-daus-000.php

3 Test Setup



E0 will use E1 and E2 in order to load LCF Instruction into the memory of the device under test. E3 is used for giving the inputs and E4 for validating the behaviour of the device under test.



4 Test Cases

Legend: I refers to one of the E3 devices, Q refers to one of the E4 devices. X = Has no influence.

4.1 AND Test Cases

<table>
<tr><th>ID</th><th>Req ID.</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T01</td><td>R01,
R02,
R06,
R08,</td><td>Test Q0 can be set with AND gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX1 AND %IX0;</td><td>0x07 0x06 0x01 0x00 0x01 0x01 0x05 0x02 0x00</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T02</td><td>R01,
R02,
R06,
R08,</td><td>Test Q1 can be set with AND gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX1 := %IX1 AND %IX0;</td><td>0x07 0x06 0x01 0x00 0x01 0x01 0x05 0x02 0x01</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ID</th><th>Req ID.</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T03</td><td>R01,
R02,
R06,
R08,</td><td>Test Q2 can be set with AND gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX2 := %IX1 AND %IX0;</td><td>0x07 0x06 0x01 0x00 0x01 0x01 0x05 0x02 0x02</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T04</td><td>R01,
R02,
R06,
R08,</td><td>Test Q3 can be set with AND gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX2 := %IX1 AND %IX0;</td><td>0x07 0x06 0x01 0x00 0x01 0x01 0x05 0x02 0x02</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ID</th><th>Req ID.</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr>
<td>T05</td>
<td>R01,
R02,
R06,
R08,</td>
<td>Test if all Inputs matter.</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 AND %IX1 AND %IX2 AND %IX3;</td><td>0x0d 0x06 0x01 0x03 0x01 0x02 0x05 0x01 0x01 0x05 0x01 0x00 0x05 0x02 0x00</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td>
</tr>
<tr>
<td>T06</td>
<td>R01,
R02,
R06,
R08,</td>
<td>Test if all outptus matter</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 AND %IX1;
%QX1 := %IX0 AND %IX1;
%QX2 := %IX2 AND %IX3;
%QX3 := %IX2 AND %IX3;</td><td>0x1c 0x06 0x01 0x01 0x01 0x00 0x05 0x02 0x00 0x01 0x01 0x01 0x00 0x05 0x02 0x01 0x01 0x03 0x01 0x02 0x05 0x02 0x02 0x01 0x03 0x01 0x02 0x05 0x02 0x03</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td>
</tr>
</table>

4.2 Or Test Cases

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T11</td><td>R01,
R02,
R06,
R08,</td><td>Test Q0 can be set with OR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 OR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x04 0x02 0x00</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T12</td><td>R01,
R02,
R06,
R08,</td><td>Test Q1 can be set with OR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX1 := %IX0 OR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x04 0x02 0x01</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T13</td><td>R01,
R02,
R06,
R08,</td><td>Test Q2 can be set with OR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX2 := %IX0 OR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x04 0x02 0x02</td></tr>
</table>
</td><td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr>
<td>T14</td>
<td>R01,
R02,
R06,
R08,</td>
<td>Test Q3 can be set with OR gate.</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX3 := %IX0 OR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x04 0x02 0x03</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td>
</tr>
<tr>
<td>T15</td>
<td>R01,
R02,
R06,
R08,</td>
<td>Test if all Inputs matter.</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 OR %IX1 OR %IX2 OR %IX3;</td><td>0x0d 0x06 0x01 0x03 0x01 0x02 0x04 0x01 0x01 0x04 0x01 0x00 0x04 0x02 0x00</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T16</td><td>R01,
R02,
R06,
R08,</td><td>Test if all outptus matter</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 OR %IX1;
%QX1 := %IX0 OR %IX1;
%QX2 := %IX2 OR %IX3;
%QX3 := %IX2 OR %IX3;</td><td>0x1c 0x06 0x01 0x01 0x01 0x01 0x00 0x04 0x02 0x00 0x01 0x01 0x01 0x00 0x04 0x02 0x01 0x01 0x03 0x01 0x02 0x04 0x02 0x02 0x01 0x03 0x01 0x02 0x04 0x02 0x03</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

4.3 Not Test Cases

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T21</td><td>R01,
R02,
R06,
R08,</td><td>Test Q0 can be set with NOT gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := NOT %IX0;</td><td>0x07 0x06 0x01 0x00 0x00 0x01 0x06 0x02 0x00</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T22</td><td>R01, R02, R06, R08,</td><td>Test Q1 can be set with NOT gate.</td><td><table><tr><th>LCF Code</th><th>LCF Instructions</th></tr><tr><td>%QX1 := NOT %IX0;</td><td>0x07 0x06 0x01 0x00 0x00 0x01 0x06 0x02 0x01</td></tr></table></td><td><table><tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr><tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>T23</td><td>R01, R02, R06, R08,</td><td>Test Q2 can be set with NOT gate.</td><td><table><tr><th>LCF Code</th><th>LCF Instructions</th></tr><tr><td>%QX2 := NOT %IX0;</td><td>0x07 0x06 0x01 0x00 0x00 0x01 0x06 0x02 0x02</td></tr></table></td><td><table><tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr><tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>T24</td><td>R01, R02, R06, R08,</td><td>Test Q3 can be set with NOT gate.</td><td><table><tr><th>LCF Code</th><th>LCF Instructions</th></tr><tr><td>%QX3 := NOT %IX0;</td><td>0x07 0x06 0x01 0x00 0x00 0x01 0x06 0x02 0x03</td></tr></table></td><td><table><tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr><tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table></td></tr>
</table>

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T25</td><td>R01, R02, R06, R08,</td><td>Test if all Inputs matter.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := NOT %IX0;
%QX0 := NOT %IX1;
%QX0 := NOT %IX2;
%QX0 := NOT %IX3;</td><td>0x1c 0x06 0x01 0x00 0x00 0x01 0x06 0x02 0x00 0x01 0x01 0x00 0x01 0x06 0x02 0x00 0x01 0x02 0x00 0x01 0x06 0x02 0x00 0x01 0x03 0x00 0x01 0x06 0x02 0x00</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>T26</td><td>R01, R02, R06, R08,</td><td>Test if all outptus matter</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := NOT %IX0;
%QX1 := NOT %IX0;
%QX2 := NOT %IX0;
%QX3 := NOT %IX0;</td><td>0x1c 0x06 0x01 0x00 0x00 0x01 0x06 0x02 0x00 0x01 0x00 0x00 0x01 0x06 0x02 0x01 0x01 0x00 0x00 0x01 0x06 0x02 0x02 0x01 0x00 0x00 0x01 0x06 0x02 0x03</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
</table>

4.4 XOR Test Cases

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T31</td><td>R01,
R02,
R06,
R08,</td><td>Test Q0 can be set with XOR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 XOR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x07 0x02 0x00</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T32</td><td>R01,
R02,
R06,
R08,</td><td>Test Q1 can be set with XOR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX1 := %IX0 XOR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x07 0x02 0x01</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td><td>1</td><td>0</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T33</td><td>R01,
R02,
R06,
R08,</td><td>Test Q2 can be set with XOR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX2 := %IX0 XOR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x07 0x02 0x02</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>T34</td><td>R01,
R02,
R06,
R08,</td><td>Test Q3 can be set with XOR gate.</td><td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX3 := %IX0 XOR %IX1;</td><td>0x07 0x06 0x01 0x01 0x01 0x00 0x07 0x02 0x03</td></tr>
</table>
</td><td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr>
<td>T35</td>
<td>R01,
R02,
R06,
R08,</td>
<td>Test if all Inputs matter.</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 XOR %IX1; %QX0 := %IX2 XOR %IX3;</td><td>0x0e 0x06
0x01 0x01
0x01 0x00
0x07 0x02
0x00 0x01
0x03 0x01
0x02 0x07
0x02 0x00</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
<tr>
<td>T36</td>
<td>R01,
R02,
R06,
R08,</td>
<td>Test if all outptus matter</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := %IX0 XOR %IX1; %QX1 := %IX2 XOR %IX3; %QX2 := %IX2 XOR %IX3; %QX3 := %IX2 XOR %IX3;</td><td>0x1c 0x06
0x01 0x01
0x01 0x00
0x07 0x02
0x00 0x01
0x03 0x01
0x02 0x07
0x02 0x01
0x01 0x03
0x01 0x02
0x07 0x02
0x02 0x01
0x03 0x01
0x02 0x07
0x02 0x03</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>X</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>X</td><td>X</td><td>X</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

4.5 Combination Test

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr>
<td>T41</td>
<td>R01,
R02,
R06,
R08,</td>
<td>All Gates and All Inputs</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr><td>%QX0 := (%IX0 XOR %IX1) AND (%IX2 OR (NOT %IX3));</td><td>0x10 0x06 0x01 0x03 0x00 0x01 0x06 0x01 0x02 0x04 0x01 0x01 0x01 0x00 0x07 0x05 0x02 0x00</td></tr>
</table>
</td>
<td>
<table>
<tr><th>I0</th><th>I1</th><th>I2</th><th>I3</th><th>Q0</th><th>Q1</th><th>Q2</th><th>Q3</th></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>X</td><td>X</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr>
<td>T42</td>
<td>R01,
R02,
R06,
R08,</td>
<td>All Outputs All Gates All Inputs</td>
<td>
<table>
<tr><th>LCF Code</th><th>LCF Instructions</th></tr>
<tr>
<td>%QX0 := (%IX0 XOR %IX1) AND (%IX2 OR (NOT %IX3));
%QX1 := (%IX0 AND %IX3) OR (%IX1 AND (NOT %IX2));
%QX2 := (%IX0 XOR (NOT %IX1)) AND (%IX2 OR %IX3);
%QX3 := (%IX3 XOR %IX1) XOR (%IX2 OR (NOT %IX0));</td>
<td>0x40 0x06
0x01 0x03
0x00 0x01
0x06 0x01
0x02 0x04
0x01 0x01
0x01 0x00
0x07 0x05
0x02 0x00
0x01 0x02
0x00 0x01
0x06 0x01
0x01 0x05
0x01 0x03
0x01 0x00
0x05 0x04
0x02 0x01
0x01 0x03
0x01 0x02
0x04 0x01
0x01 0x00
0x01 0x06
0x01 0x00
0x07 0x05
0x02 0x02
0x01 0x00
0x00 0x01
0x06 0x01
0x02 0x04
0x01 0x01
0x01 0x03
0x07 0x07
0x02 0x03</td>
</tr>
</table>
</td>
<td>
<table>
<tr><th>I 0</th><th>I 1</th><th>I 2</th><th>I 3</th><th>Q 0</th><th>Q 1</th><th>Q 2</th><th>Q 3</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

5 MISC

<table>
<tr><th>ID</th><th>Req. ID</th><th>Purpose</th><th>Input</th><th>Expected Output</th></tr>
<tr><td>T51</td><td>R03</td><td>Test if device under test starts interpretation after device startup.</td><td>1. Supply test setup with 24V
2. Execute Test T01
3. Restart Device
4. Validate Test T01</td><td>The output of T01 still passes.</td></tr>
<tr><td>T52</td><td>R04</td><td>Test if interpretation is cyclical.</td><td><table><tr><th>LCF Code</th><th>LCF Instructions</th></tr><tr><td>%QX0 := NOT %IX0;
%QX0 := %IX0;</td><td>0x07 0x06
0x01 0x00
0x00 0x01
0x06 0x02
0x00 0x01
0x00 0x02
0x00</td></tr></table></td><td>Q0 will always equal I0.</td></tr>
<tr><td>T53</td><td>R05</td><td>Test if interpretation does not stop</td><td>1. Run test T51
2. Let the device under test run for 24 hours
3. Validate if T51 still passes</td><td>The output of T51 still passes.</td></tr>
</table>

6 LCF Cheesecake 0.35 Test Run 01.04.2019

6.1 Document Specification

Target release	LCF 0.35 - Cheescake[6]
Test Results	**PASSED**
Date	**01.04.2019**
Start Time - End Time	19:40 - 21:20 (GMT+2)
Tester	Marti, Samuel[7]
Document Purpose	Notes the test results of the test execution.
Place	EEIC - Eaton Bořivojova 2380, 252 63 Roztoky, Czechia
Environmental Temperature	23.4 C°

6.2 AND

Test Case ID	Test Result	Bug Report		
T01	**PASSED**	-		
T02	**PASSED**	-		
T03	**PASSED**	-		
T04	**PASSED**	-		
T05	**PASSED**	-		

6 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
7 https://confluence-prod.tcc.etn.com/display/~E9955465

Test Case ID	Test Result	Bug Report
T06	PASSED	-

6.3 OR

Test Case ID	Test Result	Bug Report
T11	PASSED	-
T12	PASSED	-
T13	PASSED	-
T14	PASSED	-
T15	PASSED	-
T16	PASSED	-

6.4 NOT

Test Case ID	Test Result	Bug Report
T21	PASSED	-
T22	PASSED	-
T23	PASSED	-
T24	PASSED	-
T25	PASSED	-
T26	PASSED	-

6.5 XOR

Test Case ID	Test Result	Bug Report
T31	PASSED	-
T32	PASSED	-
T33	PASSED	-
T34	PASSED	-
T35	PASSED	-
T36	PASSED	-

6.6 Combination

Test Case ID	Test Result	Bug Report
T41	PASSED	-
T42	PASSED	-

6.7 MISC

Test Case ID	Test Result	Bug Report
T51	PASSED	-
T52	PASSED	-
T53	PASSED	-

6.8 Summary

All test were passed successfully the version LCF 0.35 - Cheescake[8] can be released.

No bugs were found and reported.

8 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819

LCF Compiler Commands & Instruction Set

EEIC Electronics Group

Exported on 03/25/2019

Table of Contents

1 Introduction

This page presents the instruction set used by the interpreter which are as well the output of the compiler.

The used terms refere to the instruction set architecture defined herer: wikipedia[1]

---

1 https://en.wikipedia.org/wiki/Instruction_set_architecture

2 Instruction set

Following commands were defined

Instruction Syntax:

Operation Code	Address	Immediate Value
0000 0001	1001 0000	0000 0001

The Address and Immediate value are optional, if they are used is defined by the Operation Code.

Version	Operation Name	Operation Code Hex	Address	Immediate Value	Description
LCF 0.35 - Cheescake[2]	PUSH	0x00	NONE	Binary Value (see page 7)	Put value onto the top of a stack.
LCF 0.35 - Cheescake[3]	PUSH_P	0x01	Address Table (see page 6)	NONE	Put value onto the top of a stack from physical input.
LCF 0.35 - Cheescake[4]	POP_P	0x02	Address Table (see page 6)	NONE	Remove value from the top of stack and return it to physical output.
LCF 0.35 - Ch	POP	0x03	NONE	NONE	Return and remove value from the top of a stack.

2 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
3 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
4 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819

Version	Operation Name	Operation Code Hex	Address	Immediate Value	Description
ees cake[5]					
LCF 0.35 - Chees cake[6]	MAX	0x04	NONE	NONE	Find maximal value from top two values from stack. Remove them and put on the top of a stack result.
LCF 0.35 - Chees cake[7]	MIN	0x05	NONE	NONE	Find minimal value from top two values from stack. Remove them and put the result on top of the stack.
LCF 0.35 - Chees cake[8]	SUB	0x06	NONE	NONE	Subtract value on second position of the stack from value on first position on the stack. Remove them and return result on the top of a stack.
LCF 0.35 - Chees cake[9]	COMPARE_NEQ	0x07	NONE	NONE	Compare whether are top two values equal or not. When values not equal, put value 1(true) on the top of a stack if not put 0, original values are removed.

Bytecodes have values in range from 0x00 up to 0xFF. To send the smallest amount of data possible, bytecode 0 is assigned to most used command (push).

5 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
6 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
7 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
8 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819
9 http://loutcsjira01:8080/browse/EEICEG/fixforversion/16819

3 Address Table

The following table defines the possible addresses and their address code as defined in the IO Control of the ASIC2 Platfom. io_crtl.h[10]

Address Input	Code Hex
I0	0x00
I1	0x01
I2	0x02
I3	0x03

Address Output	Code Hex
O0	0x00
O1	0x01
O2	0x02
O3	0x03

10 https://bitbucket-prod.tcc.etn.com/projects/ICPD_ICP/repos/asic2/browse/io_ctrl.h

4 Immediate Values

Following tables define immediate values.

4.1 Binary Value

Value	Code Hex
FALSE	0x00
TRUE	0x01

5 Command Example

5.1 Model Langauge

With Following model langauge

```
%QX0.0 := %IX0.2 AND %IX0.0 OR %IX0.3;
%QX0.1 := NOT %IX0.2;
```

5.2 Instructions

Follwing commands will be genrated

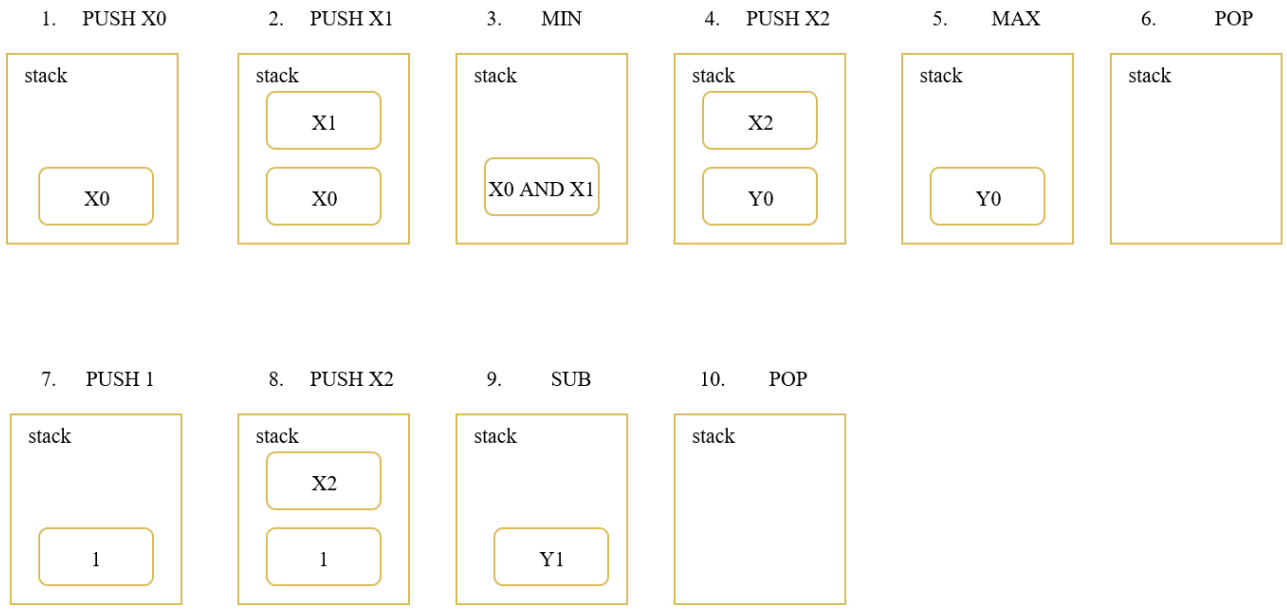
```
PUSH_P %IX0.2
PUSH_P %IX0.4
MIN
PUSH_P %IX0.3
MAX
POP_P %QX0.0
PUSH_P %IX0.2
PUSH True
SUB
POP_P %QX0.1
```

5.3 Instructions in HEX

Follwing are the instruction in hex. The inputs start with 0 and outputs with 1.

```
0x01 0x02
0x01 0x04
0x05
0x01 0x03
0x04
0x02 0x00
0x01 0x02
0x00 0x01
0x06
0x02 0x01
```

5.4 Example Visualized



6 Syntax of commands

Possible order of tokens in command is written in following tables. First table presents allowed type of Token in front of the Token. Second row presents allowed type of Token after the Token. First row presents all possible types of tokens and first collumn presents Token according to which is table completed.

F. e. first row means, that before token of type output is allowed only type termination (output token has to be only on the beginning of new command).

PRE	**output**	**input**	**and**	**or**	**xor**	**not**	**:=**	**(**	**)**	**;**
output										yes
input			yes	yes	yes	yes	yes	yes		
and		yes							yes	
or		yes							yes	
xor		yes							yes	
not			yes	yes	yes	yes	yes	yes		
:=	yes									
(			yes	yes	yes	yes	yes	yes		
)		yes							yes	
;		yes							yes	
POST	**output**	**input**	**and**	**or**	**xor**	**not**	**:=**	**(**	**)**	**;**
output							yes			
input			yes	yes	yes				yes	yes
and		yes				yes		yes		
or		yes				yes		yes		
xor		yes				yes		yes		
not		yes				yes		yes		
:=		yes				yes		yes		
(		yes				yes		yes		

POST	output	input	and	or	xor	not	:=	(	)	;
)			yes	yes	yes				yes	yes
;	yes									

6.1 Examples

```
y := x1 AND x2;
y := x1 OR x2;
y := NOT x1;
y := (x1 AND x2);
y := (x1 OR x2) AND (x3 OR x4);
y := (x1 OR x2) OR (x3 OR x4);
y := NOT NOT x1;
y := (NOT x1);
y := NOT (x1 AND x2);
y := ((x1 AND x2) OR (x3 AND x4));
y := (x1 AND NOT x2);
y := (x1 OR NOT x2);
```

LCF IEC 61131

EEIC Electronics Group

Exported on 03/27/2019

Table of Contents

IEC 61131 *is an* IEC[1] *standard for* programmable controllers[2]*. It was known as IEC 1131 before the change in numbering system by IEC. The parts of the IEC 61131 standard are prepared and maintained working group 7, programmable control systems, of subcommittee SC 65B of* Technical Committee[3] *TC65 of the* IEC[4]*.*

Wikipedia[5]

---

1 https://en.wikipedia.org/wiki/International_Electrotechnical_Commission
2 https://en.wikipedia.org/wiki/Programmable_logic_controller
3 https://en.wikipedia.org/wiki/List_of_IEC_technical_committees
4 https://en.wikipedia.org/wiki/International_Electrotechnical_Commission
5 https://en.wikipedia.org/wiki/IEC_61131

1 Relevance Assessment

Following table was created after studying the overview of the IEC 61131 standard and its subjects, it is assessing what parts of the IEC 61131 are relevant for LCF.

Standard	Topic	Relevance	R
IEC 61131-1	General information.	The introduction into the basics of the standards and PLC's is critical.	✅
IEC 61131-2	Equipment requirements and tests.	Defines the requirements for validating PLC's with its associated peripherals.	✅
IEC 61131-3	Programming languages.	Not relevant since LCF is concerned with the interpreter, and the prototype will use Blockly.	❌
IEC 61131-4	User guidelines.	Not relevant since user of the prototype are not yet intended.	❌
IEC 61131-5	Communications.	Not relevant right since it describes the implementation on the application layer (OSI), ergo user point of view. Also it does not describe the dirstibuted automation asspect. Thus the IEC61499 could suit better.	❌
IEC 61131-6	Functional safety.	Not necessary but could be interesting to know if it would be possible.	⚠
IEC 61131-7	Fuzzy control[6] programming.	Fuzzy logic is not in the scope of the FoF project and thus will not be considered. Also, it is related to programming languages.	❌
IEC 61131-8	Guidelines for the application and implementation of programming languages.	Guidelines for the programming language could be interesting for later steps also in regards if Blockly would be sufficient, but it is not necessary.	⚠
IEC 61131-9	Single-drop digital communication interface for small sensors and actuators (SDCI, marketed as IO-Link[7])	IO-Link are not in the FoF or LCF scope.	❌

6 https://en.wikipedia.org/wiki/Fuzzy_control
7 https://en.wikipedia.org/wiki/IO-Link

2 Resources

Following additional resources can be found.

Resource	Describtion	Link
PLCopen	Describes the standard with some overviews, with all its categories. (1 to 9)	http://www.plcopen.org/pages/tc1_standards/
Wikipedia	Describtive entry.	https://en.wikipedia.org/wiki/IEC_61131

3 LCF IEC 61131-1

Following is an assessment of the defined content of the standard and its relevance to the LCF Interpreter[8], referring to LCF IEC 61131 (see page 3).

The IEC 61131-1 mainly refers to the characteristics of an PLC. This page refers to the **IEC61131-1:2003** version.

3.1 Relevance Assessment

Following table assesses the subjects of the specific standard and its relevance towards the LCF Interpreter[9].

Subjects not mentioned in the table do not have any relevance.

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4.1	8	Basic functional structure of programmable controller system	Interpreter	✗	?	There will be always deviation since LCF will not be a fully independent PLC and implement all functions.
4.2.1	11-12	Summary	Interpreter	✗	?	Will currently not implement all functions defined. But in the future might.
4.2.2	12	Operating System	Interpreter	✗	?	The operating system startup behaviour will not be implemented as defined.
4.2.3	13	Memory Application Data Storage	Interpreter	✓	✓	Application storage is implemented as described.
4.2.4	13	Execution of tasks	Interpreter	✗	?	The current system will implement only on task, if more tasks will be possible hast to be determined.
4.3	13-14	Characteristics of the interface function to sensors	Interpreter	✓	✓	The inputs used for the IO module fall into the described characteristics.

8 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter
9 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4.4	14	Characteristics of the communication function	Interpreter	✗	?	The currently used communication goes over SWD but does not implement alarm reporting.
4.5	14	Characteristics of the human-machine interface (HMI) function	Interpreter	✗	?	No HMI included within the interpreter. The definition of HMI will need to be changed so that the whole automation system can have an HMI. The HMI is realized over other devices on the same bus system.
4.6.2	14-15	Language	Editor	✗	✓	The current editor will not comply with any of the defined languages in the standard. However, it must be adapted later.
4.6.3	15	Writing the application programme	Editor	✓	✓	The editor complies fully with the characteristics.
4.6.4	15-16	Automated System Start-up	Interpreter	✗	?	The 0.6 version will not indicate the automated system status and allow for live changes of the controller.
4.6.5	16	Documentation	All	✗	✓	Since the 0.6 version is a prototype the documentation does not.
4.6.6	16	Application Program Archiving	Editor, Compiler	✓	✓	Will be given. The editor will archive programmes in the model language and the compiler in instruction sets. Refering to: LCF Compiler Commands & Instruction Set[10]
4.7	16	Power Supply Function	Interpreter	✓	✓	Will be given over the SWD bus system.
5	16-17	Avalabiliity and Reliability	Interpreter	✗	✗	The hardware limitation introduced by the PD0014 SWD ASIC2[11] chip will not allow for the implementation of diagnostics and self testing.

✗ = Not applicable, implemented or correct.

10 https://confluence-prod.tcc.etn.com/pages/viewpage.action?pageId=62685861
11 https://confluence-prod.tcc.etn.com/display/ICPDPD/PD0014+SWD+ASIC2

= Applicable, implemented or correct.

= Possible to be applicable, implemented or correct.

LCF 0.6 refers to LCF 0.6 Gulab - Specification[12] and the LCF X.X refers to any feasible future without altering the basic hardware configuration of the LCF system according to architecture in LCF Development & Documentation[13].

3.2 Conclusion

The LCF 0.6 cannot be categorized as PLC according to the **IEC61131-1:2003** since the most characteristics defined in the standard are not met.

The possibility is given that LCF will comply more with the standard, but given it's weaker hardware a full characterisation as PLCs would be probably never possible.

Thus, the most accommodating would be to define a new standard for micro PLCs in a distributed system.

12 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+0.6+Gulab+-+Specification
13 https://confluence-prod.tcc.etn.com/pages/viewpage.action?pageId=61316392

4 LCF IEC 61131-2

Following is an assessment of the defined content of the standard and its relevance to the LCF Interpreter[14], referring to LCF IEC 61131 (see page 3).

The IEC 61131-2 mainly refers to equipment and requirement tests of a PLC. This page refers to the **IEC61131-2:2017** version.

4.1 Relevance Assessment

Following table assesses the subjects of the specific standard and its relevance towards the LCF Interpreter[15].

Subjects not mentioned in the table do not have any relevance.

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4.2.2	20	Equipment to be tested (equipment under test/EUT)	ALL	✓	✓	It is planned to execute all relevant and expected tests.
4.2.4	23	Withstand test conditions	ALL	✓	✓	Modules will be tested alone.
4.2.5	23	Climatic tests	Interpreter	✗	?	The climate test depends on the product using the interpreter.
4.2.6	23-27	Functionality verification with temperature	Interpreter	✗	?	Depends on the device using the interpreter. The version 0.6 will not validate temperature.
4.2.7	27-28	Verification Procedure	ALL	✓	✓	The defined procedure will be followed.
4.2.8	28	Requirements for test programmes and proper functioning verification procedures (PFVPs) to be provided by the manufacturer	ALL	✓	✓	Will be ensured by the validation.
4.2.9	28-29	EMC Performance criteria	Interpreter	✗	?	Will depend on implementing product. The 0.6 version will not implement it.

14 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter
15 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4.2.10	29	General facility/ laboratory conditions for tests	Interpreter	✅	✅	The hardware will be validate under specified conditions.
4.3	29-30	Test Report	ALL	❓	❓	Creating a test report which is referring to the standard and assess its implementation is possible, but not yet defined.
5.1	30	General	ALL	❌	✅	The 0.6 prototype is not a final product and thus not fully intended for industrial environment.
5.2.1	34	Ambient temperature and relative humidity	Interpreter	❌	❓	The 0.6 version will not validate that. If the version XX will do it depends on the device implemented.
5.2.2	34-35	Altitude	Interpreter	❌	❓	The 0.6 version will not validate that. If the version XX will do it depends on the device implemented.
5.3	35-37	Mechanical operating conditions and requirements	Interpreter	❌	❓	The 0.6 version will not validate that. If the version XX will do it depends on the device implemented.
5.4	37-39	Transport and storage conditions and requirements	Interpreter	❌	❓	The 0.6 version will not validate that. If the version XX will do it depends on the device implemented.
6.2	41-46	Power Input Ports	Interpreter	❌	❓	The 0.6 version will not validate that. If the version XX will do it depends on the device implemented.
6.3	46	Memory Power Back-up	Interpreter	❌	❌	Devices probably never implement this.
6.5	48-60	Digital I/Os	Interpreter	❌	❓	The prototype will have digital I/O as defined, but will not be verified as described especially regarding the temperature.
6.5	61-63	Analog I/Os	Interpreter	❌	❓	Currently no analog IOs are requested.

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
6.6	64	Communication Interface Requirement	Interpreter	✗	✓	The 0.6 version will be used over a development/debug port. The released version should verify the communication interfaces accordingly.
6.7	64-65	Main processing unit(s) and memory(ies) requirements	Interpreter	?	?	If defined standards are met by the chip supplier is not ensured yet.
6.8	65	Remote input/output station (RIOS) requirements	Interpreter	✗	✗	The current implementation does not have any remote IO. Also, the future implementation will never have it since the SWD bus is used and remote IO's would be own systems.
6.9	66-67	Peripherals (PADTs, TEs, HMIs) requirements	Interpreter	✗	?	The current module will not have any external peripherals. The future version might have some. Also, the SWD bus itself with the other components could be seen as one.
6.10	67	Self-tests and diagnostics requirements	Interpreter	✗	?	The 0.6 version will not include any diagnostics for the user or debugging tools. However the X.X version could.
7	68-78	Electormagnetic compatibility (EMC) requirements	Interpreter	✓	✓	The used hardware for the 0.6 or X.X is usually already exsiting within the ICPD product portfolio and thus tested for EMC anyways.
8	78-88	Marking requirements for information in EMC installation	Interpreter	✗	✓	The 0.6 will be a protoype thus marking requirments will not be fullfilled. However the commerically avaliable version X.X will fullfill the marking requirements.

✗ = Not applicable, implemented or correct.

✓ = Applicable, implemented or correct.

? = Possible to be applicable, implemented or correct.

LCF 0.6 refers to LCF 0.6 Gulab - Specification[16] and the LCF X.X refers to any feasible future without altering the basic hardware configuration of the LCF system according to architecture in LCF Development & Documentation[17].

16 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+0.6+Gulab+-+Specification
17 https://confluence-prod.tcc.etn.com/pages/viewpage.action?pageId=61316392

4.2 Conclusion

The LCF 0.6 will not be validated according to the **IEC61131-2:2017** since most defined requirements are unescessary for a protoype/Case Study.

However a commercial product implementing X.X will be able to be validate according to the definition, expect in regards of external or remote IOs and the memory power back up, since LCF will be implement in low cost devices.

LCF IEC 61499

EEIC Electronics Group

Exported on 03/27/2019

Table of Contents

IEC 61499 defines a generic model for distributed control systems and is based on the IEC 61131[1] *standard.* Wikipedia[2]

---

1 https://en.wikipedia.org/wiki/IEC_61131
2 https://en.wikipedia.org/wiki/IEC_61499

1 Relevance Assessment

Following table was created after studying the overview of the IEC 61131 standard and its subjects, it is assessing what parts of the IEC 61131 are relevant for LCF.

Standard	Topic	Relevance	Result
IEC 61499-1	Architecture	The introduction architecture of the whole standard. Thus, necessary	✅
IEC 61499-2	Software tool requirements	Requriements for software tools such as the design studio. Thus, not necessary for LCF.	❌
IEC 61499-3	Tutorial Information (**WITHDRAWN**)	Was withdrawn.	❌
IEC 61499-4	Rules for Compliance Profiles	Rules which must be met in order to be compliant with the standard.	✅

2 Resources

Following additional resources can be found.

Resource	Describtion	Link
Wikipedia	Describtive entry.	https://en.wikipedia.org/wiki/IEC_61499

3 LCF IEC 61499-1

Following is an assessment of the defined content of the standard and its relevance to the LCF Interpreter[3], referring to LCF IEC 61499 (see page 3).

The IEC 61499-1 Defines generic architecture for distributed function block design. This page refers to the **IEC61499-1:2012** version.

3.1 Relevance Assessment

Following table assesses the subjects of the specific standard and its relevance towards the LCF Interpreter[4].

Subjects not mentioned in the table do not have any relevance.

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4.1	18	System Model	ALL	✗	✓	0.6 will be standalone device with no means of device to device communication. X.X will allow should allow for that referring to: LCF Device to Device Communication Concepts[5].
4.2	19	Device Model	Interpreter, Device	✗	✓	0.6 will not contain a communication interface
4.3	19-20	Resource Model	Interpreter, Device	✗	✓	0.6 will not implement a resources model which is able to communicate. X.X will allow for this as defined in the standard.
4.4	21	Application Model	Editor, Interpreter	✗	?	0.6 will not implement such a concept of application. X.X could to that but it is not decided yet if so.
4.5	21-25	Function Block Model	Editor, Interpreter	✗	?	0.6 does not implement the function block model as described. X.X might implement a model as described.
4.6	25	Distribution Model	Interpreter	✗	✓	0.6 will not implement the model as described since it will not support a device to device communication. X.X will support it since it will implement a device to device communication.

3 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter
4 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter
5 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Device+to+Device+Communication+Concepts

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4.7	25-26	Management Model	Editor, Interpreter	✗	?	0.6 will not implement. X.X might implement it if needed (SWD assist).
4.8	27	Operational State Model	Interpreter	✗	?	0.6 will not implement it. X.X might implement it if an operational state model is needed, but it is unlikely since the applications on SWD are supposed to work more independent from each other than defined. (PUB/SUB style instead of direct connections)
5.2	28-33	Basic Function Blocks	Editor, Interpreter	✗	?	X.X might implement the basic function blocks as specified, with the instance behaviours etc.
5.3	33-36	Composite Function Blocks	Editor, Interpreter	✗	?	X.X might implement the composite function blocks as specified based on the basic function blocks.
5.4	36-38	Subapplications	Editor, Interpreter	✗	?	X.X might implement sub applications as defined, based on the function blocks.
5.5	38-41	Adapter Interface	Editor, Interpreter	✗	?	X.X might implement adapter interfaces for interfacing with different function blocks.
5.6	41	Expection and Fault Handling	Editor, Interpreter	✗	?	X.X might implement fault handling and exceptions.
6	41-52	Service Interface Function Blocks	Editor, Interpreter	✗	?	X.X might implement management or communication function blocks.
7	52-55	Configuration of functional units and systems	Editor, Interpreter	✗	?	X.X might implement configuration possibilities defined, the 0.6 version does not allow for any configuration or communication.

✗ = Not applicable, implemented or correct.

✓ = Applicable, implemented or correct.

? = Possible to be applicable, implemented or correct.

LCF 0.6 refers to LCF 0.6 Gulab - Specification[6] and the LCF X.X refers to any feasible future without altering the basic hardware configuration of the LCF system according to architecture in LCF Development & Documentation[7].

6 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+0.6+Gulab+-+Specification
7 https://confluence-prod.tcc.etn.com/pages/viewpage.action?pageId=61316392

3.2 Conclusion

The LCF 0.6 version does not implement any of the in **IEC61499-1:2012** specified standards, mainly due the editor and interpreter not using function blocks at all. (Just Logic Gates) and having no communication between devices.

The X.X version can be mostly implement the standard since it will have a device to device communication. However, it is doubtable that a full implementation of **IEC61499-1:2012** is possible since it aims for adaptability between different vendors, which is not given by using SWD and proprietary technologies. However, if the design and application should be model after the defined function blocks is questionable since the devices using X.X are aimed for simple settings.

Additionally, the standard is mainly created by universities and has yet not found a full adaptation within the industry. Referincing to:TheFutureOfIndustrialAutomationIEC61499.pdf[8] and https://arxiv.org/ftp/arxiv/papers/1303/1303.4761.pdf

8 https://confluence-prod.tcc.etn.com/download/attachments/80155150/TheFutureOfIndustrialAutomationIEC61499.pdf?api=v2&modificationDate=1549012305697&version=1

4 LCF IEC 61499-4

Following is an assessment of the defined content of the standard and its relevance to the LCF Interpreter[9], referring to LCF IEC 61499 (see page 3).

The IEC 61499-4 Defines the features in order to gain interoperability, portability and configurability between vendors and tools. This page refers to the **IEC61499-4:2013** version.

4.1 Relevance Assessment

Following table assesses the subjects of the specific standard and its relevance towards the LCF Interpreter[10].

Subjects not mentioned in the table do not have any relevance.

Nr.	Page	Subject	Component	LCF 0.6	LCF X.X	Comment
4	7-8	Contents of Complience Profile	ALL	✗	?	0.6 will not produce any complience profile. However the version X.X might do so, but a complete interoperability between SWD devices and non SWD devices will probably not be implemented.

✗ = Not applicable, implemented or correct.

✓ = Applicable, implemented or correct.

? = Possible to be applicable, implemented or correct.

LCF 0.6 refers to LCF 0.6 Gulab - Specification[11] and the LCF X.X refers to any feasible future without altering the basic hardware configuration of the LCF system according to architecture in LCF Development & Documentation[12].

4.2 Conclusion

The implementation of the standard would result in a document describing how the IEC61499-4 standard implements the portability configurability and interoperability of the standard. However, the version X.X may never achieve the requested interoperability between different vendor devices if SWD is not used.

9 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter
10 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+Interpreter
11 https://confluence-prod.tcc.etn.com/display/EEICEG/LCF+0.6+Gulab+-+Specification
12 https://confluence-prod.tcc.etn.com/pages/viewpage.action?pageId=61316392