Mathias Maijanen

PRODUCTIZATION OF PCB HANDLER SOFTWARE

PRODUCTIZATION OF PCB HANDLER SOFTWARE

Mathias Maijanen
Opinnäytetyö
Syksy 2019
Sähkö- ja automaatekniikan tutkinto-ohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Sähkö- ja automaatiotekniikan tutkinto-ohjelma, automaatiotekniikka

---

Tekijä: Mathias Maijanen
Opinnäytetyön nimi suomeksi: Piirilevykäsittelijän ohjelmiston tuotteistus
Opinnäytetyön nimi englanniksi: Productization of PCB handler software
Työn ohjaajat: Satu Vähänikkilä (OAMK), Mauri Savolainen (JOT Automation Oy)
Työn valmistumislukukausi ja -vuosi: syksy 2019
Sivumäärä: 63 + 2 liitettä

Opinnäytetyönä tehdyn projektin tavoitteena oli piirilevykäsittelijän näyttö- ja logiikkasovellusten päivitykset sekä laitteen dokumentaation tarkastus ja päivitys.

Näyttöpäivityksen (HMI) tarkoituksena oli tuottaa laitteelle näyttösovellus, joka on valmiina tulevaisuuden integrointia varten. Näyttösovelluspäivitys toteutettiin käyttäen pohjana ja referenssinä saman käsittelijätuoteperheen pidemmälle kehitettyä laitetta ja sen näyttösovellusta. Näyttöpäivitys toteutettiin Beijer iX Developer kehitysympäristössä.

Logiikkasovelluspäivitys (PLC) toteutettiin pohjautuen pidemmälle kehitetyn laitteen ohjelmistoon, aikaisempiin kehityshuomioihin sekä sovellustestauksen ja -testauksen aikana esiin nousseisiin seikkoihin. Sovelluskehitys toteutettiin Omron CX-Programmer -ohjelmointiympäristössä. Tavoitteena oli kehittää uusia vaadittuja toimintoja vastaava ohjelmisto.

Laitteen vuokaaviot sekä operaattoritason käyttöohje päivitettiin vastaamaan nykyisiä toiminnallisuuksia.

Opinnäytetyöraportissa esitellään työn määritelmä, työssä hyödynnetyt työkalut ja teoria, työsuunnitelmat, työn vaiheet, toteutus sekä lopputulokset. Työ esitellään luonteensa vuoksi projektidokumentin muodossa.

Työn lopputuloksena piirilevykäsittelijän käyttämät HMI- ja PLC-ohjelmistot päivitettiin vastaamaan vaadittuja ja suunniteltuja toiminnallisuuksia, laitteen käyttöohjeen viimeistelyä varten koottiin vaadittavat muutokset sisällään pitävä dokumentti ja laitteen vuokaaviot tarkastettiin sekä korjattiin vastaamaan nykyistä toiminnallisuutta.

Asiasanat: Ohjelmoitava logiikka, graafinen käyttöliittymä, ohjelmointi, käyttöönotto, testiautomaatio.

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Electrical and Automation Engineering, Automation Engineering

---

Author: Mathias Maijanen
Title of thesis: Productization of PCB handler software
Supervisors: Satu Vähänikkilä (OAMK), Mauri Savolainen (JOT Automation Ltd)
Term and year when the thesis was submitted: autumn 2019
Pages: 63 + 2 appendices

This project was done as the author's bachelor's thesis. The objectives set at the beginning of the project included the update of a PCB handler's HMI and PLC programs along with the review and revision of associated documentation.

The purpose of the HMI update was to enhance the machine's GUI usability, further develop its HMI software based on development done on a more advanced M-series machine's HMI and enable the future integration of a HMI with a larger screen size to the PCB handler. The updates were carried out in the Beijer iX Developer development environment.

The aim of the PLC program update was to further develop the software based on work done on a more advanced M-series PCB handler, previously made development notes and things noted during testing and development of the program. Software development was done using Omron CX-Programmer.

The machine's flowcharts and the operator level user manual were updated to match current machine functionality.

The bachelor's thesis report contains details on the project's definition, planning, execution and results alongside the theory, equipment and tools used to carry it out.

As a result of the software productization project the machine's PLC and HMI software were updated to match the latest functionalities according to defined references, the machine's flowcharts now match its functionality and the user manual is ready to be updated in the future by the JOT documentation department based on the compilation document.

Keywords: Programmable logic, graphical user interface, programming, commissioning, test handling.

INDEX

GLOSSARY

BCR: Bar code reader.

C#: C Sharp. Programming language.

CPU: Central Processing Unit.

DUT: Device under Test.

FBD: Function Block Diagram programming. An IEC-61131 compliant programming language.

FINS: Factory Interface Network Service.

GUI: Graphical User Interface. Visible on the HMI.

HMI: Human-Machine Interface. A part of the machine's UI.

IDE: Integrated Development Environment.

IEC: International Electrotechnical Commission. An international standardization organization.

JOT: JOT Automation Ltd. Referred to as 'JOT', 'company' or 'commissioner'.

LD: Ladder Diagram programming. An IEC-61131 compliant programming language.

M1: A JOT Automation test handler. Referred to as 'handler', 'machine' or 'M1'.

M5: A JOT Automation test handler. Referred to as 'M5'.

PCB: Printed Circuit Board.

PID: Proportional-integral-derivative. A control type used in process control.

PLC: Programmable Logic Controller.

SCADA: Supervisory Control and Data Acquisition. A PC based process control and monitoring system used as an interface for industrial controllers.

SMEMA: IPC SMEMA-9851:2007. An equipment interface standard.

ST: Structured Text programming. An IEC-61131 compliant programming language.

UDP: User Datagram Protocol. Network communication protocol.

UI: User Interface. Includes the HMI and all mechanical parts of the interface.

UINT: 32-bit unsigned integer value. Up to 2^{32} (4 294 967 296, decimal value).

1 INTRODUCTION

This document details the execution of a PCB handler software productization project commissioned by JOT Automation Ltd (JOT) done as the author's bachelor's thesis during autumn 2019. The primary objectives of the project are the further development of the JOT M1 test handler's PLC and HMI software and the review and revision of its documentation according to updated specifications and features.

The project consists of four parts: the review and updating of the M1 test handler functionality specifications (flowcharts), the updating of the Human-Machine Interface (HMI) software, the further development of the machine's Programmable Logic Controller (PLC) software and the updating of the machine's user manual. The updates are based on earlier solutions.

This document goes through the definition, theory, planning, execution and results of the project. The execution of the project itself is divided into five somewhat overlapping phases: starting, definition, planning, execution and closing phases. The project is executed using the modified waterfall model that enables the flexible planning and execution of phases.

JOT Automation Ltd is a company specializing in the design and manufacture of automated electronics handling, testing and assembly solutions. Its products and solutions serve clients globally in a wide range of industries. The company has been headquartered in Oulu area since its inception in 1988 and has offices in 7 different countries. Since June 2018 it has been a part of Suzhou Victory Precision Manufacture Co. Ltd. (1.)

2 PROJECT DEFINITION

This chapter defines the project objectives. The project was commissioned by JOT Automation Ltd in August 2019 and the execution begun immediately. The baseline goals and milestones for the further development of the M1 test handler were defined during a starting meeting held on 15.8.2019. The set milestones are part of the larger goal of productization of the handler's software.

At the heart of the project was the need to bring the M1 up to par with the M5 test handler in terms of software functionality (where applicable, clarification in chapter 2.3) and usability.

2.1 Machine Specifications Update

The review and update of the machine's existing flowcharts is needed due to changed handler specifications. The existing released flowcharts are made for an older version of the machine that among other things makes use of physical switches instead of an HMI for manual machine control. A version of the machine equipped with an HMI was first created in July 2017. The latest flowchart version is from May 2017.

The accuracy of the flowcharts is to be determined by comparing them to the current and upcoming handler features and functionality.

In case a need for changes emerges, the current version of the flowchart is to be revised to match the updated machine functionality. The machine's software updates are to be executed based on the updated flowchart.

The flowchart will be reviewed again upon completion of the HMI and PLC software's in case changes have happened.

2.2 HMI Software Update

The Human-Machine Interface (HMI) software is to be updated to match the set general look and user experience of the M-series product family. This update includes a total facelift to match the visuals of the M5, a password protection overhaul and a change to a larger HMI screen size.

The current Beijer X2 Base 5 HMI is set to be replaced by the Beijer X2 Base 7. This change increases the HMI's display size from 5 inches to 7 inches while the resolution stays the same. It will further enhance the machine's user experience as the larger screen is noticeably easier to use. The Beijer X2 Base 7 screen is also used in other JOT Automation machines and thus the generifying of the screens in use will affect maintenance, upkeep and spare part availability positively.

The changes are made to unify the user interface design between the JOT M1 and M5 test handlers. The HMI is currently only implemented into the PLC software of the machine‘s demo-version and as such needs to be implemented into the actual product's software.

Upon completion of the HMI and PLC software updates they both need to be tested to verify functionality.

2.3 PLC Software Update

The M1 test handler's PLC software is to be updated to match the current needs and specifications.

At the beginning of the project the machine's software was not in a condition where it could easily be implemented into a newly manufactured machine due to changed features and HMI integration. The PLC software update's objective is to make a ready-to-use software capable of being applied to new M1 test handlers equipped with an HMI.

The goal is to review, update and enhance the existing M1 test handler software based on the M5 test handler software where applicable. During the process the program will also be cleaned up of unnecessary I/O's, code and program networks. Internal machine specific features will also be combed through.

One major difference between the M1 and the M5 comes from the PLC's used: the M1 uses an Omron CP1L-EM PLC and the M5 uses an Omron NX1P2 PLC. This difference means that the two test handler's software solutions will never be exactly the same as the tools used to program the two PLC's are different: the M1 PLC is programmed with Omron CX-Programmer and the M5 PLC is programmed with Omron Sysmac Studio. This, however, opens up further development possibilities.

Upon completion of the HMI and PLC software updates they both need to be tested to verify functionality.

2.4 User Manual Update

The M1 test handler's user manual contents are to be updated to match the current operator level functionality. This mainly concerns the use of the updated HMI software and the changes it causes to the usage of the machine.

The entire user manual is reviewed to see whether any other changes need to be made.

The user manual update will be carried out by the document department of JOT Automation Ltd. The contents of the update are to be compiled by the thesis author.

3 THEORY

This chapter goes through the equipment, tools and standards used during the planning and execution of the project. The chapter takes a look at the thesis' subjects in a more general fashion. Additionally this chapter includes a section detailing some of the core functionalities and differences between the JOT M1 and the M5 test handlers.

3.1 JOT M1 Test Handler

The JOT M1 test handler (picture 1) is a modular, low-volume test handler capable of being up-scaled by connecting it to a production line with other M-Test System machines. (2.) The M1 test handler's modularity comes from being able to be used with any single 2-7 U (Unit, a space unit) JOT M10 Series Test Box according to the user's needs.

The M1 can process DUT's (Device under test) ranging from 80 x 50 mm (Length x Width) to 300 x 200 mm (L x W). (3.) It is IPC SMEMA 9851:2007 (SMEMA) compliant. The handler features a single board transfer system with a segmented conveyor built-in. It supports all SMEMA compliant electrical interface signal lines (board available, machine ready, pass/fail board coming/going).

The handler's primary function is to drive DUT's to the connected test box and then drive them out accompanied by a test result signal (pass/fail).

The PLC controlling the M1 test handler is a CP1L-EM40. The CP1L-EM40 has built-in Ethernet capability which is used in the M1 to communicate with the test box, the test PC and the HMI.

The demo version of the M1 test handler is equipped with a Beijer Electronics X2 Base 5 HMI and features a Microscan Vision Mini code reader.

PICTURE 1. JOT M1 test handler. (2.)

3.1.1 IPC SMEMA-9851:2007

The IPC SMEMA-9851:2007 is an equipment interface specification standard for board transfer manufacturing systems of surface-mounted Printed Circuit Boards (PCB). The standard specifies the minimum mandatory requirements that a conveyor-to-conveyor transport system has to meet along with non-mandatory specifications. By doing so it also sets some guidelines for machine mechanical design. The standard is not a complete specification of a system's interface. The 2007 version of the standard supersedes the SMEMA 1.2 standard. (4.)

IPC-SMEMA-9851 February 2007

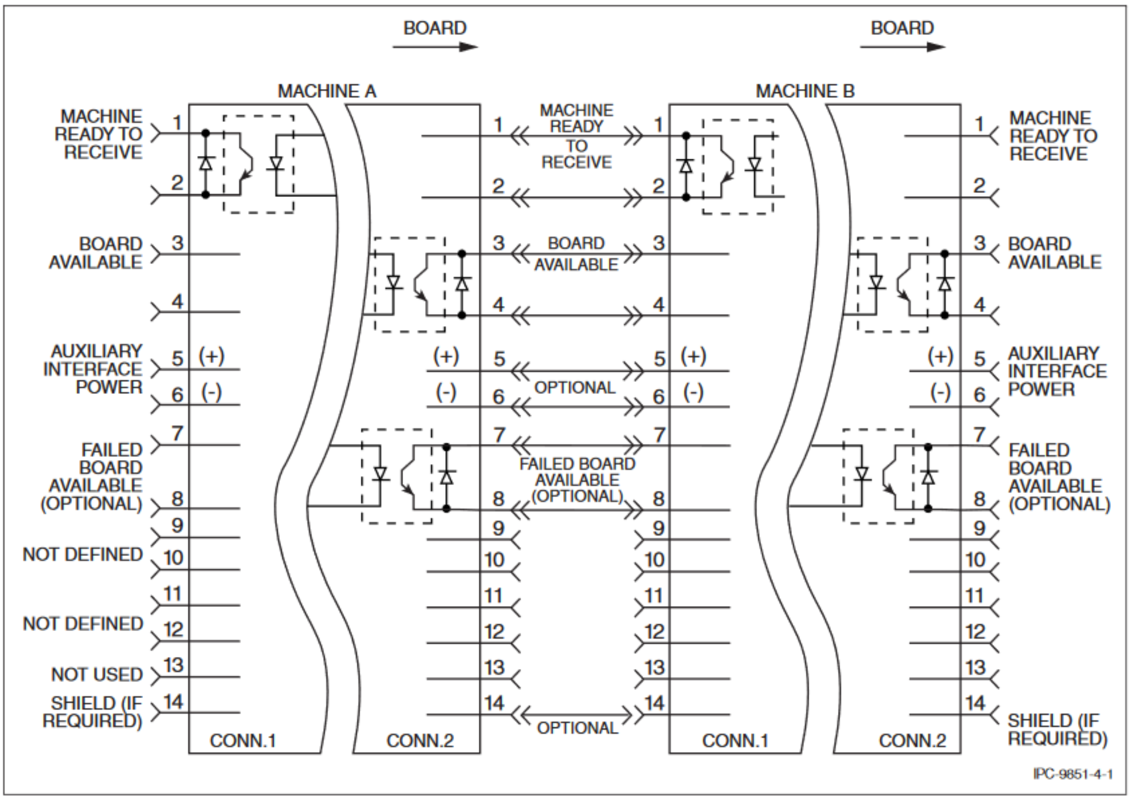


Figure 4-1 Electrical Interface Schematic

PICTURE 2. SMEMA Electrical Interface Schematic (4.)

The SMEMA standard electrical interface (picture 2) includes two mandatory and a single optional signal line for communication between production line machines: the “Board Available” (BA) and “Machine Ready” (NB, not busy) signals are mandatory for SMEMA systems and the “Board Pass/Fail” signal is optional. Board Pass/Fail signal was added in the 2007 version of the standard.

More details including functional descriptions and timing diagrams available in appendix 2.

3.2 JOT M5 Test Handler

The JOT M5 (picture 3) is a modular test handler suitable for high-mix and high-volume production. It is a part of the M-Test System machine lineup and supports the same M10 Series Test Boxes as the M1. It features space for up to five 4 U test boxes (20 U total). (14.)

The M5's primary purpose, much like the M1's, is to drive incoming DUT's to a connected, correct test box. Where it differs from the M1 is that is supports the testing and handling of multiple different products (high-mix production).

PICTURE 3. JOT M5 Test handler. (15.)

The M5 features automatic width adjustment functionality. Width to be used can be set manually or fetched from a product database. The M5 can process DUT's with dimensions ranging from 80 x 50 mm (Length x Width) to 300 x 200 mm (L x W). The maximum DUT top and bottom side heights are 50 mm and 25 mm respectively. The M5 supports all SMEMA compliant electrical interface signals.

The M5 supports a product database of up to 50 saved product configurations. The product configuration parameters include component height, board thickness, length and width.

The M5 is controlled by an Omron NX1P2 PLC and is programmed with Omron Sysmac Studio. A Beijer X2 Base 7 HMI is used for monitoring, operation and service purposes. The NX1P2 features built-in Ethernet and EtherCAT ports for communication with other parts and peripherals of the test system such as the lift's servo motors.

3.2.1 M1 compared to M5

The JOT M1 and M5 test handlers both feature the same core idea: the automation of PCB testing. Both machines achieve this goal using a patented conveyor system, a lift and an integrated test box control interface. Both machines can make use of the same test boxes with the key difference being the amount supported: the M5 supports up to 5 test boxes while the M1 only supports a single box.

The M1 features manual width adjustment and a three segmented conveyor. The conveyor's 1st and 3rd segments are normal conveyors and the 2nd, middle segment, is a lift conveyor. The middle segment's lift is driven down and back up in order to drive an incoming PCB to a test box. Manual width adjustment means that in order to change the type of tested PCB the user has to unlock the non-fixed side of the conveyor and reposition it correctly.

The M5 features a lift conveyor: both the lift and the conveyor are built into a single entity controlled with servo motors. The lift conveyor has automatic width adjustment meaning that it can adjust itself according to incoming product specifications without the user having to reposition it.

The supported amount of test boxes and the type of width adjustment in the machine are key factors in what kind of production the machine supports. In the case of M1 and M5, the M5 supports mixed production and high-volume testing while the M1 supports lower test volumes with more manual labor required if the PCB width changes.

Both machines support code reading for acquiring incoming PCB specifications as an optional feature. The read code, in most cases the product number, is sent to a connected test box for product identification. The M5 also supports an internal (built-in and saved on the controller) product database from which the current product information can be set and fetched.

The M1 software runs on a CP1L-EM PLC and is programmed using CX-Programmer. The M5 software runs on an NX1P2 PLC and is programmed using Sysmac Studio. Both machines feature Beijer X2 Base HMI's programmed with iX Developer.

3.3 Human-Machine Interface (HMI)

A human-machine interface (HMI) is a screen or a monitor used in industrial applications as a part of a system's user interface (UI). The system's graphical user interface (GUI) is visible on the HMI. HMI's can be used for monitoring, controlling and visualizing processes and associated data. They are typically used for monitoring and communication between the process operator and the process controller such as a programmable logic controller or a PID controller. (20, 21.)

3.3.1 Beijer X2 Base 7 HMI

The Beijer Electronics X2 Base 7 (picture 4) features a 7", 800x480 pixel resolution TFT-LCD touch panel. The panel's hardware specifications include a 400 MHz ARM9 CPU, 128 MB RAM and a 256 MB flash memory of which 200 MB is reserved for user applications. It features a single Ethernet port, 4 serial communication ports (RS232/422/485) and a single USB port by default.

The X2 Base line of panels feature a corrosion-proof IP65 rated plastic casing meaning that the panel is water jet resistant and wholly dust tight according to the IEC 60529 standard. The Finnish SFS-EN 60529 standard's contents are the same and it is used as a reference for the purposes of this thesis due to availability. (6.)

The HMI uses Windows Embedded CE 6.0 as its runtime's operating system. (11.) It is programmed using the Beijer iX Developer development environment.

PICTURE 4. Beijer X2 Base 7 HMI (11.)

3.3.2 Beijer iX Developer

The Beijer iX Developer is the development environment used for programming the X2-series Beijer HMI's.

iX Developer features .NET Framework 4.7.1 compatibility and supports C Sharp (C#) programming and Xaml scripting in the IDE. This enables versatile programming of the HMI: the user can choose to use both basic and advanced functions according to their project's needs.

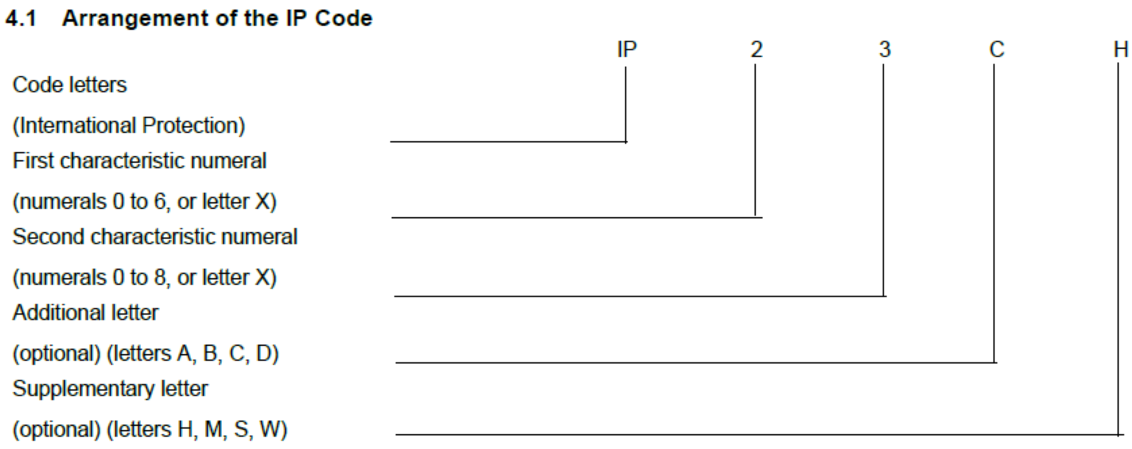
Basic functions include the premade drag-and-drop style functionalities available in iX Developer. Advanced functions consist mainly of user created scripts. These scripts can be made in either C# or Xaml languages.

The Beijer iX Developer IDE supports configuring communications for numerous manufacturers' PLC's with support for multiple different communication protocols including Ethernet/IP, Omron FINS and serial communications.

3.3.3 IEC 60529/SFS-EN 60529

The standard consists of the classification of degrees of protection for electrical equipment provided by their enclosures. It is applicable for equipment with a rated voltage of up to 72.5 kV. The original IEC standard was released in 1989 and has had 2 amendments added since: the first in 1999 and the second in 2013. The Finnish version of the standard is the SFS-EN 60529. It was released in 2000 and translated in 2011. (12, p.1.)

The standard uses a code created with specific, pre-designated elements to indicate the protection class of equipment. It is called the IP Code. It is not to be confused with IP address. The code is formed as shown in picture 5.



PICTURE 5. Arrangement of the IP Code (18, p. 25.)

The four parts of the code concern the following:

- First character after IP: Protection of equipment against ingress of solid foreign objects OR protection of persons against access to hazardous parts with…
- Second character after IP: Protection of equipment against ingress of water
- Third character after IP: Protection of persons against access to hazardous parts with…
- Fourth character after IP: Supplementary information for a specific type of test/equipment/condition.

The chart consisting of the elements and details of the IP Code is included in this document as appendix 1.

3.4 Programmable Logic Controller (PLC)

A programmable logic controller (PLC), or a programmable logic, is used in industrial applications to control processes and systems. A PLC is a central processing unit (CPU) based controller used to execute software stored in its programmable memory based on input and output sensor data and the instructions given.

They are designed to enable easy monitoring and programming without requiring a lot of programming or development experience, in some cases by maintenance personnel (22, ch.1.1.3.) PLC’s employ IEC 61131 standard compliant programming languages (8.) See chapter 3.4.3 for more information on programming languages.

PLC’s are designed to withstand and operate in harsh industrial environments without failure. This is enabled by resistance to vibrations, noise and air humidity and a large scale of operation temperatures.

PLC’s typically feature all core hardware such as the CPU, the power supply unit (PSU), input and output interfaces, communications interfaces and programmable memory built-in. (22, ch.1.2.)

3.4.1 Omron CP1L-EM40 PLC

The CP1L-EM40 PLC (picture 6) is manufactured by Omron Corporation. It is classified by Omron as a compact PLC with modular features. It features built-in Ethernet communications with socket services support, two expansion slots for serial communications and analog I/O option boards and support for up to 3 CP1W-series expansion units in any combination.

The CP1L-EM40 used in the M1 features 40 built-in digital I/O points. The 40 points are divided into 24 inputs and 16 sourcing transistor outputs. It also features 2 analog inputs. The PLC supports up to 160 I/O points when equipped with three 40 point CP1W expansion units. (5, 19.)

PICTURE 6. Omron CP1L-EM40 PLC (17.)

Omron PLC programming and setup are done using Omron CX-Programmer. CX-Programmer is a part the CX-One package which contains tools related to PLC configuration and functionality.

The CP1L-EM40 PLC supports IEC 61131-3:2013 standard compliant programming with Function Blocks (FB), Structured Text (ST) and Ladder (LD). (13, 19.)

3.4.2 Omron CX-Programmer

CX-Programmer is a logic programming environment developed by Omron. It is used to program Omron PLC's released before the newer N-series PLC's. It is the predecessor of Omron Sysmac Studio and the successor of Omron SYSWIN. CX-Programmer is a part of the CX-One automation software package and as such to take full advantage of it the user needs access to the rest of the tools, too. (7.)

CX-Programmer can be used to program PLC's with Ladder (LD), Structured Text (ST) or Sequential Function Chart (SFC) programming languages depending on hardware specifications. The most commonly used language is LD. A Statement List (STL) language editor is accessible as an option in the ladder program editor.

The user can create or import IEC 61131-3 compliant Function Blocks in either LD or ST languages. These Function Blocks enable the modularization of simple, relatively unchanging processes. Not all PLC's support execution of Function Blocks.

When using CX-Programmer the user must assign I/O's to specific addresses. These addresses can be in either physical (in- or outputs) or work range. The physical I/O ranges vary depending on the PLC hardware configuration. The user can save memory data to the PLC for use with settings and parameters. This memory data can be accessed, read and written to in the user program.

3.4.3 IEC 61131-3

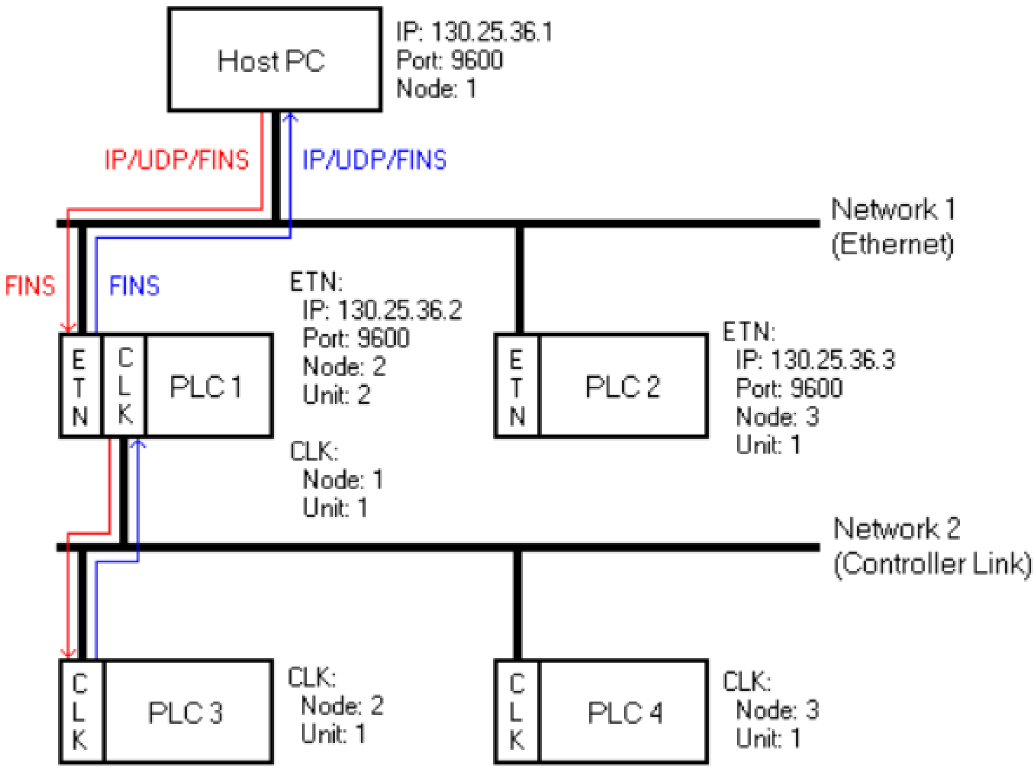
The IEC 61131-3:2013 is the 2013 version of the 3rd part of the IEC standard for programmable controllers. It concerns PLC programming languages. It sets guidelines for languages used in PLC programming and systems. The IEC 61131 standard group consists of 10 parts concerning PLC's, their programming, additional equipment and general guidelines. The standard is developed by the International Electrotechnical Commission (IEC). (8.)

The 10th part (IEC 61131-10) was released in April 2019. It concerns an XML-based exchange format for importing and exporting of IEC 61131-3 compliant projects in XML format. (9.)

IEC 61131-3:2013 compliant languages include two textual and two graphical languages. The supported textual languages are Instruction List (IL, same as Statement List) and Structured Text (ST). The supported graphic languages are Ladder (LD) and Function Block Diagram (FBD). The standard also includes Sequential Function Chart (SFC) for the purpose of defining the structure of PLC programs' and function blocks internal organization. (10.)

3.4.4 Omron FINS

Omron FINS is a network protocol created to enable the communication of Omron PLC's over different physical networks. It is primarily used for device communication over Ethernet or HostLink (RS232C, RS422A or RS485). Ethernet communications can be done with either TCP- or UDP/IP protocols with UDP/IP communication being the more transport protocol. The default FINS/UDP port is 9600 but it can be modified in the PLC settings. (16.)



PICTURE 7. Omron FINS network layers (16.)

In picture 7 the PLC 1 works as a gateway between the Ethernet and Controller Link networks. The picture depicts the sending of a command from the Host PC to PLC 3 via PLC 1 using two different network protocols. FINS supports up to three network layers: in a system such as the one in the above picture a third level can be established using either PLC 3 or 4 as a gateway.

4 PLANNING

This chapter includes information about the planning phase of the project. It will define the plan used to meet the goals set in the definition phase individually for each part of the project. It will also define the tools to be used during execution, requirements of each part and any required pre-execution setup or configuration modifications.

The project in its entirety was planned to take place between the starting date and the end of October. A comprehensive plan of execution for the project was formed at the start of the project with room for possible future changes. This plan detailed the dates, timespans and projected completion goals for each phase along with meetings, checks and other project milestones.

The plan of action is formed according to the commissioner’s internal practices and prevalent industry standards.

4.1 Machine Specifications Update

There are two existing flowchart versions for the M1: an original from February 2017 and an updated one is from May 2017. Both are to be reviewed and taken into account when making updates.

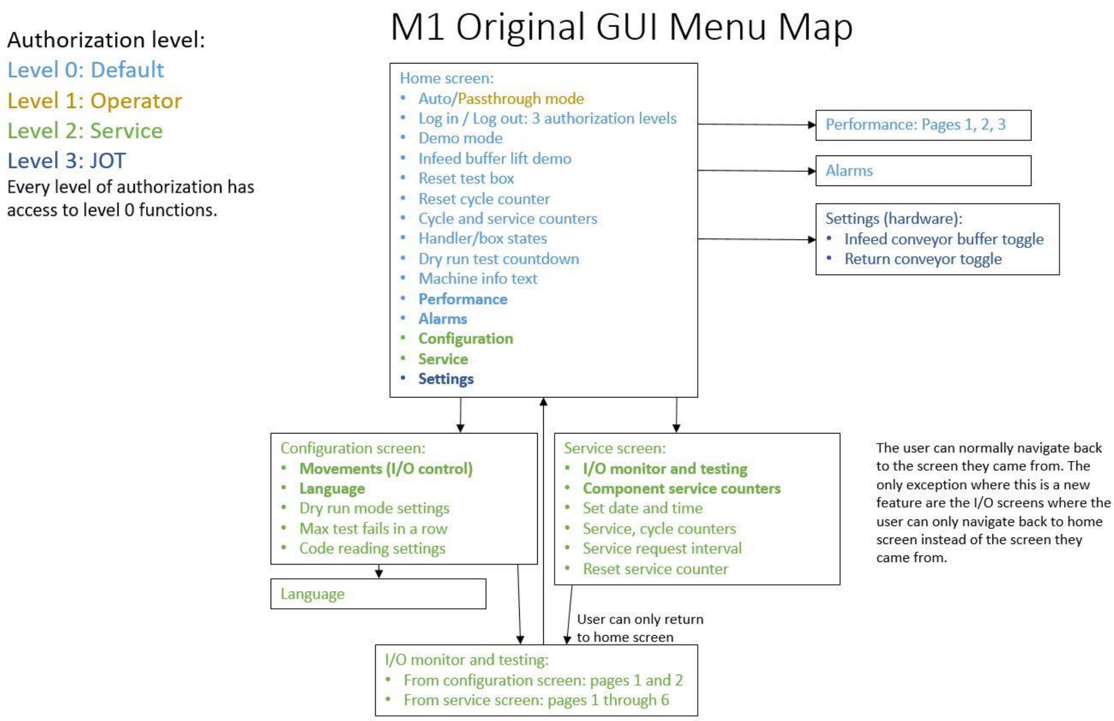
The review and updates will be done to the May 2017 version. They are to be carried out with Microsoft Visio. The main goal is to find out what is missing, what needs to be changed and which parts are already up-to-date in the original flowchart. The accuracy will be determined by comparing the flowcharts with the machine software functionality and contents.

Among the pre-determined changes to be reviewed is the integration of an HMI to the machine and if it has affected machine functionality as described in the flowcharts. This integration mainly concerns the service, hardware settings and configuration of the M1 software.

The HMI and PLC software functionality updates will be done based on the revised handler flowchart.

4.2 HMI Software Update

The HMI software update was planned using the JOT M5's HMI software and the revised flowchart as references for functionality. At the end of the project's execution phase the user experience of both machines' HMI's should be virtually identical save for the differences in machine configurations and features. GUI menu maps (pictures 8 and 9) were created to help visualize the program structure and plan the changes to be made. The revised GUI menu map was designed to be as close to the M5 navigation scheme as possible.



PICTURE 8. M1 original GUI navigation map.

The HMI update is to be carried out using Beijer iX Developer. Some of the updates are to be made alongside the PLC software updates to ensure compatibility. An HMI update is required as a part of the PCB handler software productization efforts as the majority of operation is to be done using the HMI and the currently available HMI software and its PLC integration are lacking.

The HMI features under focus are the user interface and graphics, the password protection system, test box IP address modification functionality, test box control and the change to a larger screen size.

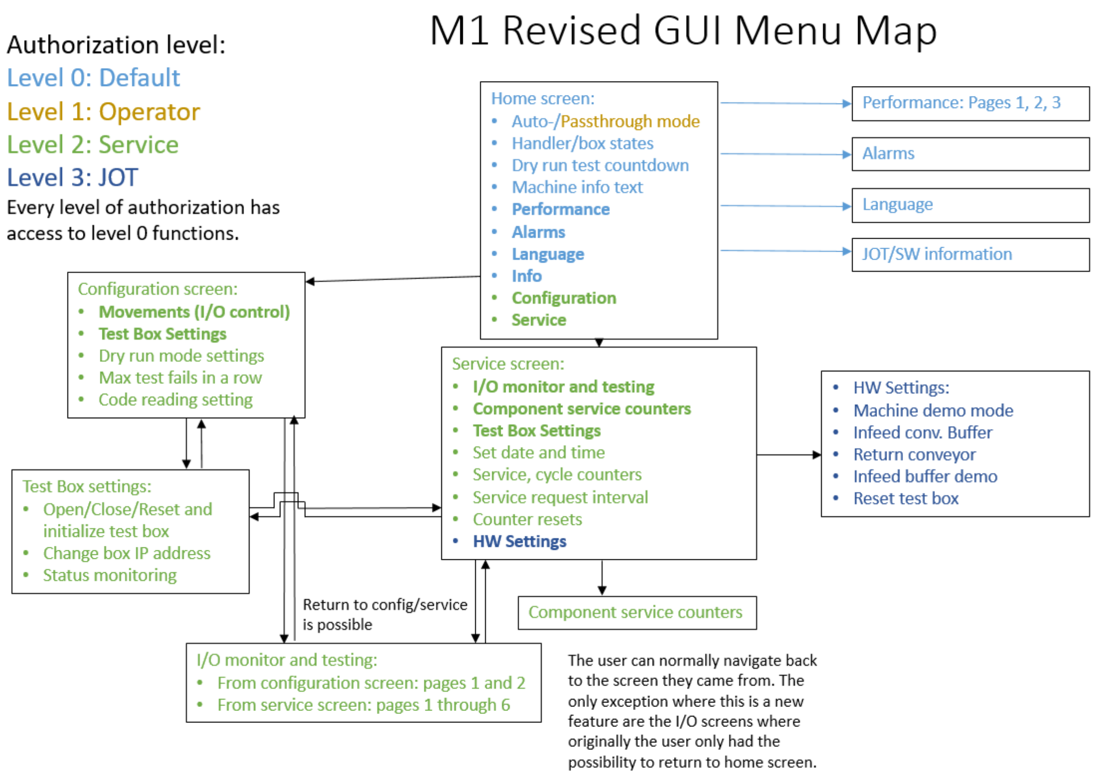
The HMI update is to be carried out in the following order:

1. User interface (UI) graphics overhaul including change to a larger screen size,
2. UI element placement,
3. Password protection modification,
4. Addition of new functionalities,
5. Testing of completed modifications.

The parts 1 through 3 of the above list can be executed entirely in iX Developer without concerns about PLC software compatibility. Parts 1 and 2 are entirely graphical and they consist mainly of modification, addition and placement of UI elements with some changes to the actions linked to the elements. These graphics changes will make navigation in the GUI user-friendlier.

The password protection system will be changed from the existing Log In – Log Out implementation to a system prompting the user to input a password when they try to access a certain GUI page: e.g. Pass through mode requires a level 1 (Operator) password, configuration mode requires a level 2 (Service) password and hardware settings require a level 3 (JOT) password. The password protection system will be programmed with C#.

The 4th and 5th parts of the above list will be executed at the same time as the corresponding PLC software updates. Features to be added or updated include machine operation mode control, UI navigational changes and test box control. Test box control includes commands for initializing, opening, closing and resetting a test box along with a function for test box IP address modification via HMI. The test box IP address modification will be programmed with C#. It will be implemented in the PLC software using ladder programming.



PICTURE 9. M1 GUI navigation after update.

Complete functionality testing will be carried out once the PLC and HMI software are up-to-date. The HMI software will be tested on a X2 Base 7 HMI.

4.3 PLC Software Update

The M1 PLC software update was planned based on the programming solutions used in the M5, the revised M1 flowchart and previously made software development plans and the update needs noted by the thesis' author. Features to be added include things that are already implemented in the M5's software that can also be used in the M1 to enhance functionality. The end goal of the PLC and HMI updates is the productization of the PCB handler's software.

The update is carried out using CX-Programmer and will partially overlap the HMI software update. The programming will be done using LD (ladder) programming. The update is to be done based on the M1 demo software version.

The update will consist of a program clean up, reviewing and updating of GUI states and text controls, machine operation mode control, test box control enhancements, test box IP address modification changes and general bug fixes.

The M1 software is not entirely finished and thus contains remnants of previous development changes in the form of bypassed program parts and redundant functions. The entire software along with its I/O list will be reviewed and unnecessary parts will be cleaned up. The machine's functionality will be tested along the way to ensure nothing breaks.

The GUI state and text controls in the PLC are lacking. These are to be reviewed and updated according to updated specifications and functionalities.

The machine's controls are not entirely functional with the HMI integrated. Parts of the PLC software controlled with the HMI are to be updated to function correctly. This includes additions and revisions to operation modes and ensuring proper machine mode resetting. Operation modes will be controlled entirely based on the HMI state.

Additional test box controls will be added. A test box IP address setting modification program will be created to enable the changing of currently used test box easily via the GUI. Modification at the start of the project requires the user to have access to the PLC software as the IP is stored in the PLC's memory and can only be changed manually. This makes usage of a test box with an IP differing from a preset one difficult.

A test box reset and initialization command from GUI will be added to the software. This makes the setup of a new test box and the solving of possible connection issues easier as a test box in an error state can be reset without cycling its power or using the HTML interface.

As with the HMI software updates the setup detailed in chapter 5.2 must be done before beginning the PLC software updates. Testing of both the HMI and PLC software will be done concurrently.

4.4 User Manual Update

The user manual is to be updated to match the latest specifications. The writing of the update is to be done by JOT Automation documentation department to ensure company policies and standards are met. The contents of the update are to be gathered by the thesis' author.

The update will be planned using a printed copy of the latest version of the handler manual. Changes and additions will be marked on the printed document and then gathered to a single digital file and indicated so that future implementation into the user manual can be easily done.

The parts of the user manual to be updated shall be compiled into a single, comprehensive document which is then forward to the document maker. The focus of the update is on the parts concerning HMI and machine control, but the manual in its entirety is to be reviewed for possible mistakes or inconsistencies.

5 EXECUTION

This chapter goes through the work done during the project. All of the execution phase work is based on the plans made during the planning phase of the project with the tools and methods detailed in the earlier chapters.

Along with the end results of the project work this chapter will detail the steps required to enable the successful execution of the project. These steps include the review of machine specifications, mechanical setup of the handler and establishing communication between different parts of the system.

The project was executed largely in the same order as detailed in this chapter.

5.1 Machine Specifications Update

The M1's existing flowcharts were reviewed to see whether they were still accurate and up to date. The flowcharts' accuracy was tested by running and monitoring the M1 test handler in Auto and Pass through modes and keeping an eye on the functionalities and tasks executed by the program. Inspection was done by monitoring the machine's visible functionalities by eye and using CX-Programmer to monitor the processes running in the background.

The review found very few things to be lacking. Contrary to the original plan, the integration of an HMI had not had much of an effect on the machine's functions covered in the flowcharts and thus did not require major changes. HMI integration mainly concerns service, setup and configuration functionalities.

Some needs for minor modifications were found. Changes and additions included naming changes, description changes, additional conditions and minor functionalities.

References to "Manual mode" were changed to "Service mode" to fit the current naming scheme. The code reader's maximum re-read decision was changed from a fixed "Failed re-reads over 5" to "Over maximum fail limit". Maximum fail limit is set in the GUI. Anchor points missing from some connectors were added.

Missing conditions were added as necessary. Flowcharts and modifications needed for demo mode functionality were made.

Demo mode flowchart consists of starting either Auto or Pass through mode subprocesses normally and running them with a modified DUT take out process. This process stops the DUT at the end of the outfeed conveyor, returns it to the start of the infeed conveyor and then drives it back to the infeed conveyor stopper.

The result of the flowchart and machine specifications review is that currently no major changes are needed as functionality remains largely unchanged. Additions and corrections were made as necessary. The flowchart is ready for use and describes the machine's functionality well. Changes to information regarding service, setup and configuration modes and their functionality are updated in the user manual.

5.2 Execution Pre-Requirements

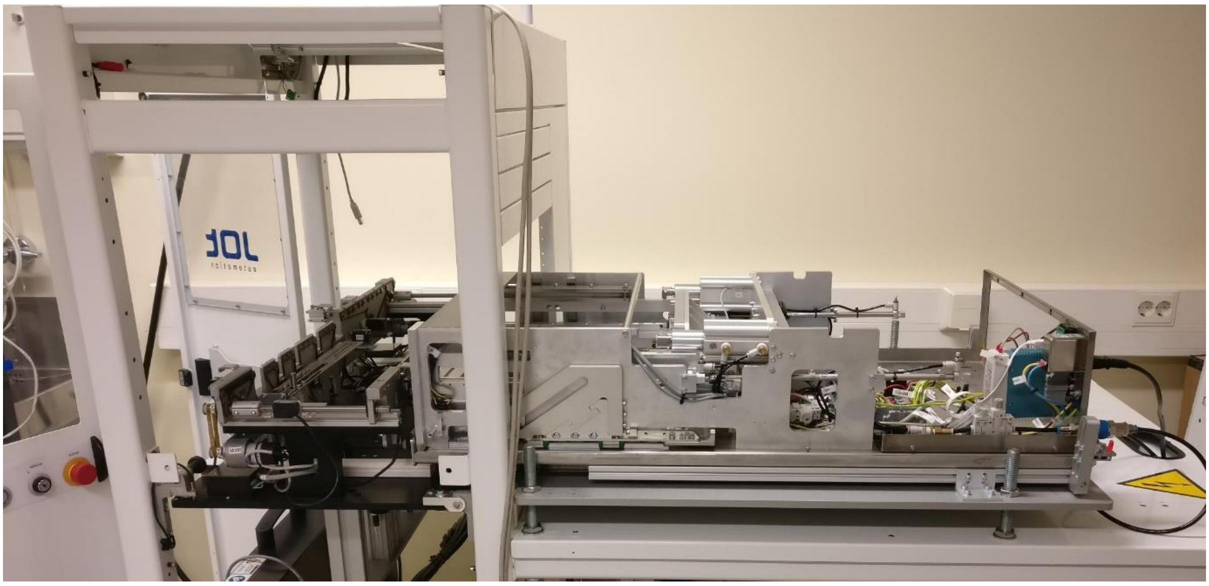
This chapter goes through the pre-requirements of the software updates including the machine's mechanical setup, test box setup and establishing communications. The different setups were done in the in order of appearance. Each part is vital for the correct functioning of the machine and must be tested after completion to prevent problems moving onwards.

This chapter assumes that the machine comes with working software pre-downloaded. Setup requires the user to have access to the necessary tools and software.

5.2.1 Machine Setup

Machine setup, or commissioning, is the first pre-requirement for the successful execution of any further updates. It is done before the communications setup and the test box setup. Setup includes inspection, connecting and testing of the machine. Some functions can only be tested after the communications are established.

A physical inspection of the machine was carried out at the start of the setup. This included making sure nothing looks broken or out of place, ensuring that every part is where it should be or that they were included in the shipment and that the general outlook of the machine is as specified. Inspection is made easier by removing the M1's cover panels as seen in picture 10. After the inspection the next step of the setup is to start making the necessary cable connections.



PICTURE 10. The M1 test handler profile with covers off and test box in place.

Connections required for full functionality include machine power feed, pneumatics feed and Ethernet connections within the system. The power cables inside the system are connected first. This includes the peripherals panel, test box connection and PLC/HMI power connections.

The machine has a single power feed that is connected to a regular 230 V output as the last part of the power connections setup. The main power switch is not turned on before all other connections are made to prevent possible malfunctioning.

After the power cable connections are done the pneumatic tubes can be connected. The pneumatics are fed through a single connector that splits into three tubes. Only the test box's pneumatics need to be connected separately, other cables come pre-connected.

After ensuring that the pressurized air feed is compliant with the machine specifications and the pneumatic cables are connected, the main air feed valve can be turned on. Power must not be turned on before ensuring that the pneumatics are correctly installed and no leaks appear anywhere. In case a test box is not used the third connector split can be blocked.

After all connections are made the power can be turned on. The machine power is turned on with the main switch in the front panel and the test box is powered on with its own power switch on the back of the box. After powering up all instruments and system parts the user must check that everything is functional and powered. This includes any peripherals, the HMI panel, indicator lights, the beacon and the PLC panel.

After turning on the machine see if the HMI is communicating with the PLC. If a message stating "*Communication Error: Station 0*" appears, communication settings are wrong and will be setup later. This will most likely happen on a fresh install. Fresh installations require all software to be downloaded manually.

Before testing any functionalities the user must ensure that the machine's conveyor width adjustment is correct for the PCB in use and that if there's a box in place the conveyor, PCB and box inlet have matching widths. Width is adjusted manually by unlocking the hex nuts on both ends of the non-stationary part of the conveyor. Correct width can be verified by running a PCB through the machine manually and seeing that it moves without problems.

Testing machine functionality can be started after all other parts of commissioning are done. Testing of a standalone machine without loop conveyors or other machines in the line can be done using the built-in *Demo*- and *Dry Run mode* functionalities that return the DUT to the infeed conveyor after testing. These modes are used with *Auto* mode.

5.2.2 Communications Setup

The PLC, test box and HMI communications settings have to be setup in such a way that the machines can successfully communicate with each other and the other parts of the system. The setup has to be done to ensure successful execution of the project's other parts and the correct functionality of the handler.

For communication between a CP1L-EM PLC and an X2 Base 7 HMI the user is required to setup Omron FINS (Factory Interface Network Service) communication. This requires access to CX-Programmer, iX Developer and the HMI internal service menu. The computer with which these settings are made must also be set to the correct (the same) IP address range as the other machines.

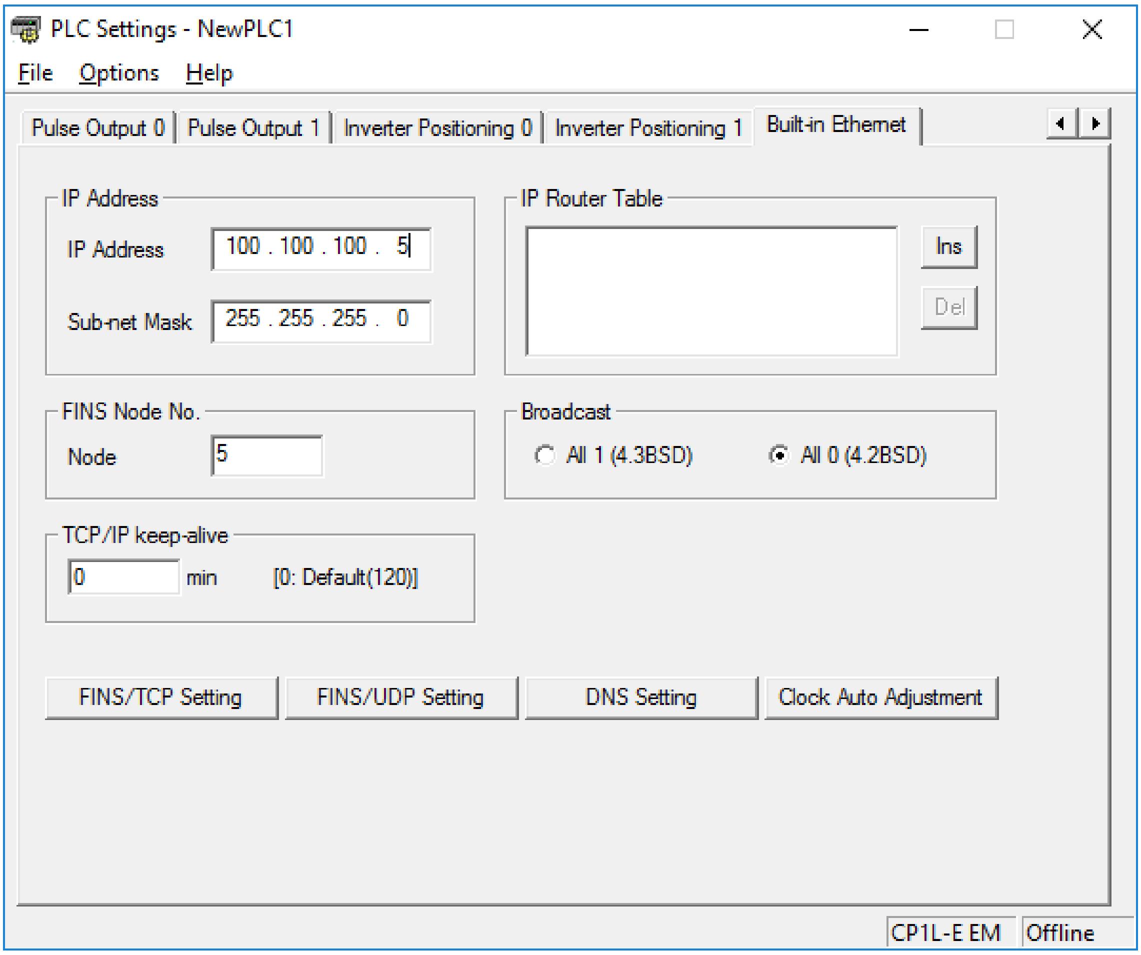
For the purposes of this project the IP address range is set as 100.100.100.nnn along with a 255.255.255.0 subnet mask. The PLC used in the M1 uses the IP 100.100.100.5, the HMI uses 100.100.100.55 and the PC used for setup is assigned to 100.100.100.20. The M10 test boxes use IP addresses that change depending on the box number, typically they are set up as 100.100.100.10*n* where *n* is the box number. On a fresh install the IP addresses for HMI, PLC, PC and test boxes will most likely be set as default (192.168.250.1 or similar) and must be changed to the correct ones.

The test box used during this project uses the IP address 100.100.100.104. The test box IP address that the PLC connects to is set manually in the PLC software memory manager. The memory words D905 through D908 store the current box IP address. Before the HMI update the IP was set here one octet at a time in hex number format. After the HMI and PLC software updates this change can be done using the HMI panel.

The test box's setup, configuration and control can be done in its own HTML interface that can be accessed with an internet browser using the IP address assigned to the box. This is detailed in the corresponding chapter.

The HMI's IP address is changed in the service menu. It can be accessed upon booting the HMI and before a project is loaded. The IP address and subnet mask are set to the correct range and the power is cycled to enable the settings. The user can also set date and time settings from the service menu.

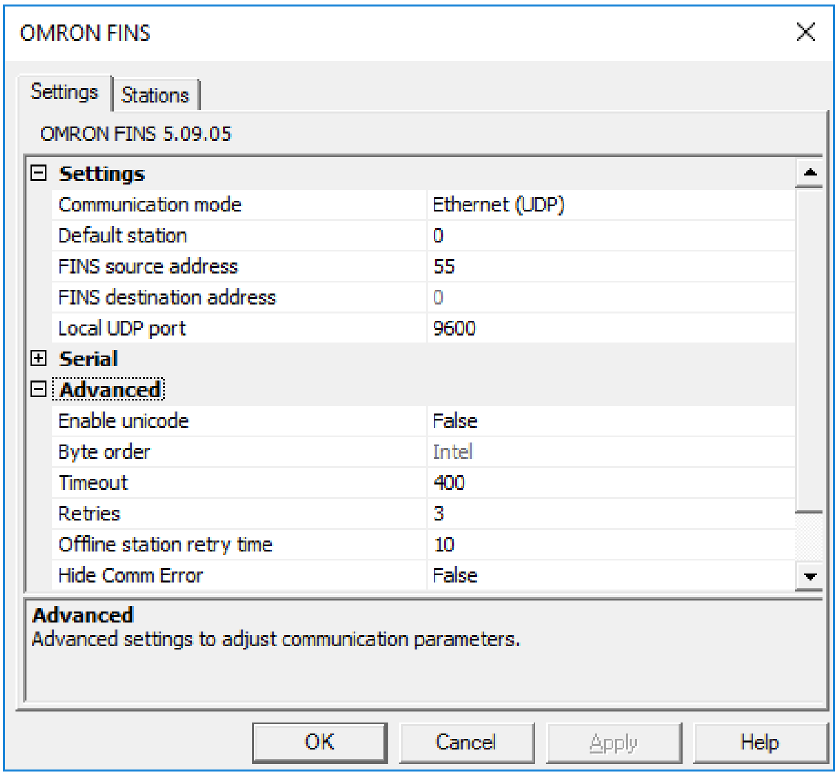
PLC FINS communication can be setup in the CX-Programmer Ethernet/IP settings (picture 11). The user must assign the PLC a FINS node number and check that the FINS/UDP settings are correct. FINS node number is the same as the IP address' last octet's value. The PLC's IP address changes after downloading the new settings. This will result in a connection error and the user must connect to the PLC using the new IP address going forward.



PICTURE 11. PLC Built-In Ethernet settings with FINS setup in CX-Programmer.

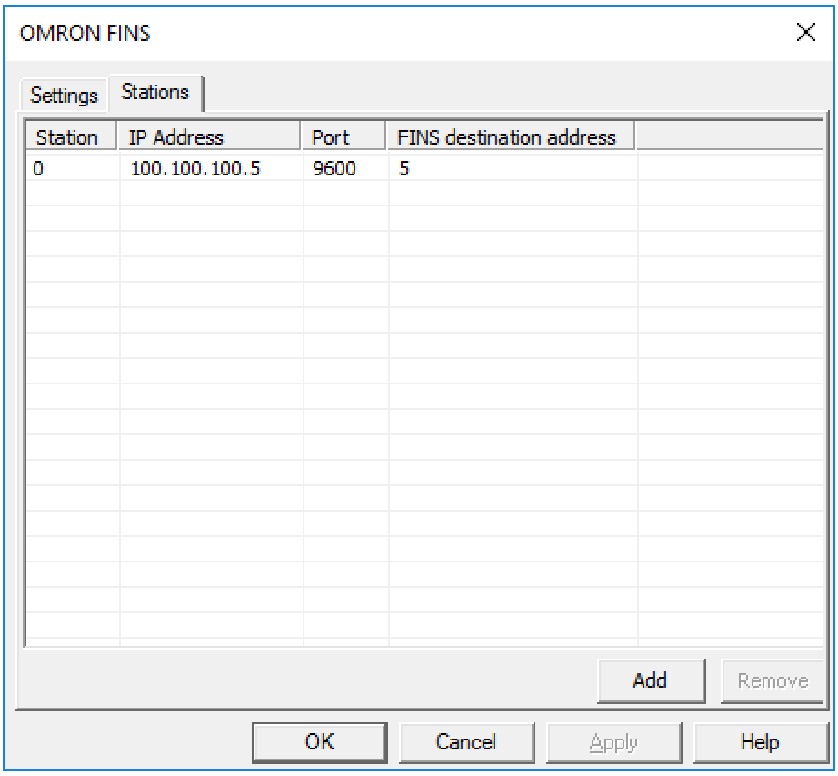
After successfully testing the new PLC communication settings the same settings can be assigned to the controller used in the iX Developer project. The controller's communication settings can be accessed in the Function menu under *Tags -> Controllers -> Controller…* and *Settings…*

The desired controller manufacturer and communication protocol to use is chosen in the *Controller…* menu. For communications within the M1 Omron FINS driver is chosen. FINS network settings are done under *Settings…* As shown in pictures 12 and 13.



PICTURE 12. FINS driver connection settings in iX Developer.

Communication mode is set as Ethernet (UDP) with the default port (9600) in use. This port is the same as that set up in the PLC settings. FINS source address is the same as the HMI's IP address' last octet, in this case 55 (100.100.100.55). Default station is 0 by default. Advanced settings can be modified as necessary.



PICTURE 13. Controller/station settings in iX Developer.

The controller/station to connect to is set in the *Stations* tab. The formerly set PLC IP settings are copied and accepted here. After making the necessary changes, download the project to the HMI.

If every part of the communications setup is correct, no communication errors should appear and the box state should be visible in the HMI's main menu.

5.2.3 Test Box Setup

The test box setup for the M1 is done at the end of the other setups as it needs pre-established communications to be properly done. Test box setup includes establishing connections, box positioning and configuring box settings. Physical connections are detailed in the Machine Setup chapter. The test box rests on the M1 box tray and is secured with two clamps on both sides at the back of the box. The box itself is stationary and the tray is moved to position the box correctly. This is done manually.

After the physical pneumatic, network and power connections are made the test box's functionality can be tested by trying to access its HTML interface. The interface can be accessed with a web browser by connecting to the current test box IP address. The assigned IP address should be visible on the back of the test box near the power switch.

The HTML interface can be used for settings, debugging, monitoring and testing of test box functions. Before moving on with the test box positioning, ensure that the box status is correct, the box IP, type and model are correct and that it responds to commands.

If the test box IP and other communication settings are correctly set in the PLC memory, the user can test issuing commands with the HMI from the I/O control screens. After any changes to test box configuration the power should be cycled to enable them.

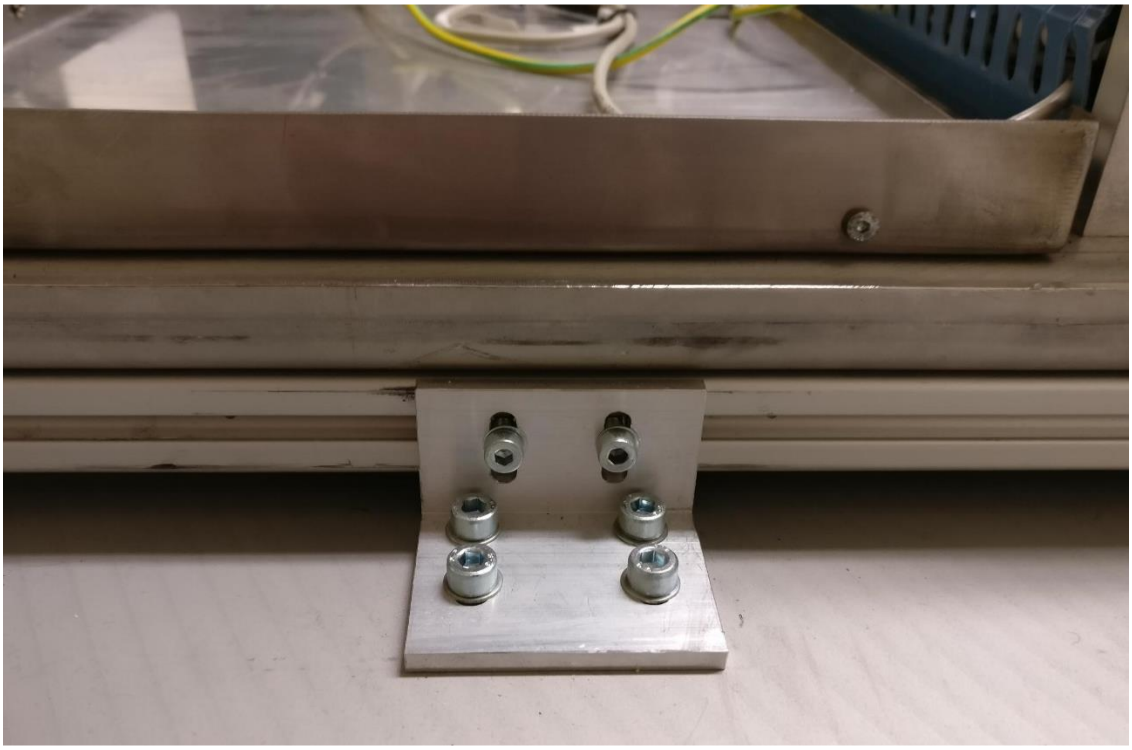
After the communications, connections and configuration setups are done and tested the box can be positioned. The tools needed for this include a spanner, a hex key and the HMI.

The first thing is to check that the *Test box in place* –sensor is activated. If not, ensure that the box's guide cylinder hits the guide hole on the box tray and that the sensor's signal strength and position are correct. After the sensor is configured the test box can be secured in place with the clamps at the back.

After the box is positioned correctly and secured on the tray and communications are established between all parts of the machine the tray itself can be positioned.

The user must first drive the M1 middle conveyor lift down from the HMI I/O control screen, issue a command from the same HMI screen or the HTML interface to open the box slide and then unlock the test box tray lock from the same HMI screen to allow horizontal positioning.

After the lift is down and the box slide and the tray lock are open, the tray's horizontal repositioning can be done by unlocking the screws holding the tray in place with a hex key (picture 14). The box must be positioned so that the lift fits the box inlet in use and does not hit the slide when moving up under any circumstances. The user must first have successfully done the width adjustment detailed in chapter 5.2.1 to prevent collisions and ensure correct functionality.



PICTURE 14. Screws preventing horizontal movement of the tray.

After the horizontal positioning is done, the user must fasten the screws to prevent the tray from moving during vertical positioning or machine operation. After ensuring the position is correct the user can test it by driving the middle lift up and seeing if it fits through the inlet without making contact.

If the middle conveyor’s stopper is in the wrong position, it needs to be moved to a position where it fits through the inlet. This can be done by untightening the screws holding it and the sensor in place. Stopper position can be verified at the same time as other functions are tested by driving the lift down, box open, lift up and stopper up in the I/O screen.

If the box is too low or too high for correct operation, the tray height can be modified by unlocking the nuts holding the entire rack in place and tightening or loosening the guide nuts until the tray is at correct height. The user must always ensure that the box is perfectly level to avoid malfunctions. After the test box is correctly positioned and the other parts of machine commissioning are successful, the testing of machine functionality can be done.

Testing of a standalone machine without return conveyors, lifts or other machines in the line such as the one used in this project can be done using the built-in *Demo*- and *Dry Run* mode functionalities that return the DUT to the infeed conveyor after testing. These functions are used with *Auto* mode to simulate normal machine functionality in a production line.

5.3 HMI Software Update

The update of the M1's HMI software was done with the goal of further improving the M1's user experience. The entire graphical outlook and functionality of the GUI was updated based on the M5. Some parts of the update were executed alongside the PLC software update to ensure compatibility.

The update was carried out using iX Developer as the primary development environment with some changes being done to the PLC software with CX-Programmer when need arose. The update included graphical changes, unifying of graphical elements, enhancing usability, clarifying navigation and creating new functionalities with both iX Developer's toolkit and C# scripting.

Before starting the actual update process the network and communications setups detailed in the previous chapters were done and the HMI used in the project was changed from the old X2 Base 5 to the new X2 Base 7. The old screen's cables were unplugged from the machine and replaced with those of the new screen.

The first part of the GUI update was the moving of elements to appropriate pages according to the revised GUI menu map (picture 9). New screens were created and elements moved as planned. The element's appearances were updated at the same time as they were moved to new positions. The appearances were modified to match the attributes of elements used in the M5 GUI. The background image was moved to its own screen and designated as a parent screen to make further development efforts less likely to move background elements by accident.

Movement of elements to new pages warranted changes to the existing navigation implementation. Among other things a ladder program enabling returning to the previous page based on the state of the PLC program was created for configuration, service and handler and test box control screens. The navigation scheme was unified with the M5 according to the GUI menu map as planned.

After the GUI element replacements were done, the entire old password protection system was removed and remade based on that used in the M5. The old password protection system was implemented in the iX Developer project and featured a simple log in / log out authentication and user authorization system. It featured three authorization levels: operator, service and JOT. Screen features were enabled according to the authorization level of the currently logged in user. The old password protection system’s problem was the lack of automatic log out and the changing of current user. The new password protection system removes these problems entirely.

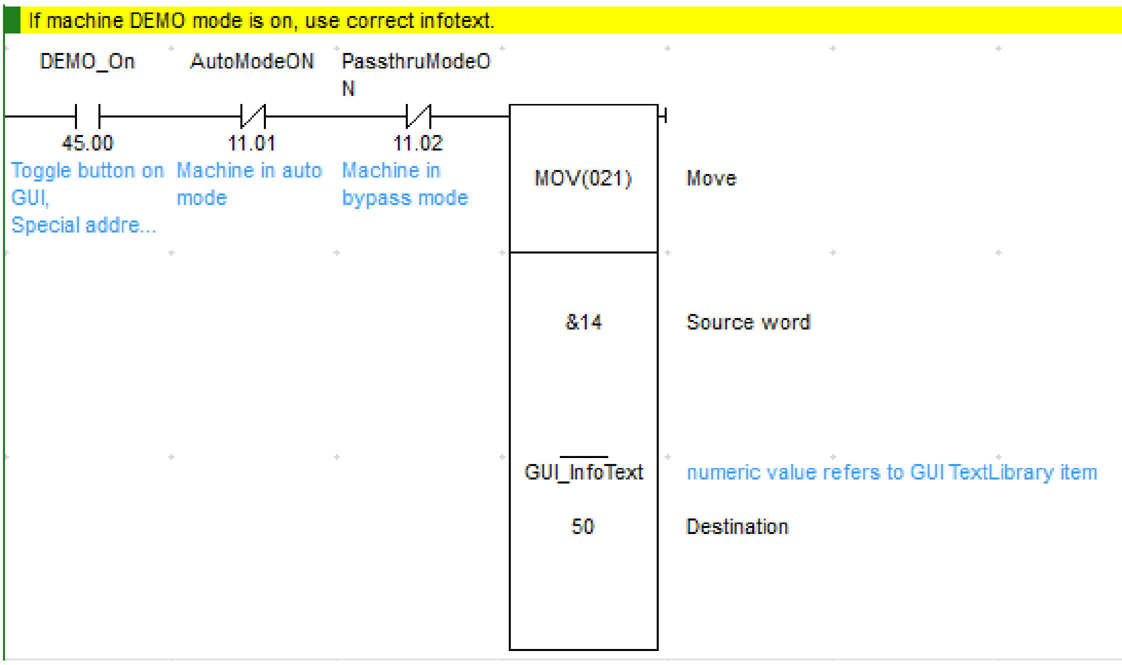
The new password protection system features the same three levels of authorization as the old one but access is gained in a different way. Instead of enabling and disabling features entirely, all features can be accessed by the user when the machine is in idle mode. Upon trying to access a feature requiring authorization, e.g. the service menu, the user is prompt for a password associated with the level of protection assigned to the feature. For example, accessing the service menu requires the user to have authorization to use level 2 (service) features (picture 15).

PICTURE 15.The new password prompt.

The password level prompt is unique to each feature. As the password protection is a feature on its own pop-up screen, a value is assigned to a variable upon access request to identify which function the user tried to access last and where to forward the user if the password is successfully entered. If the user is unsuccessful in entering the password, they are simply prompt to enter it again or exit the password screen. The new password protection system is scripted primarily in C#.

Features added and modified in parallel with the PLC software update included state, alarm and information texts, test box IP address modification, box control changes and PLC software version information.

States, alarms and information shown on the HMI screen are controlled by the state of the PLC software for most part. The control of these messages was updated as was necessary. Each message is assigned to a specific value in a text library in the HMI. When the state of the PLC software changes, a value corresponding with an appropriate state message in the text library is moved to a variable (picture 16) that is then read on the HMI and the correct message is shown (picture 17.)



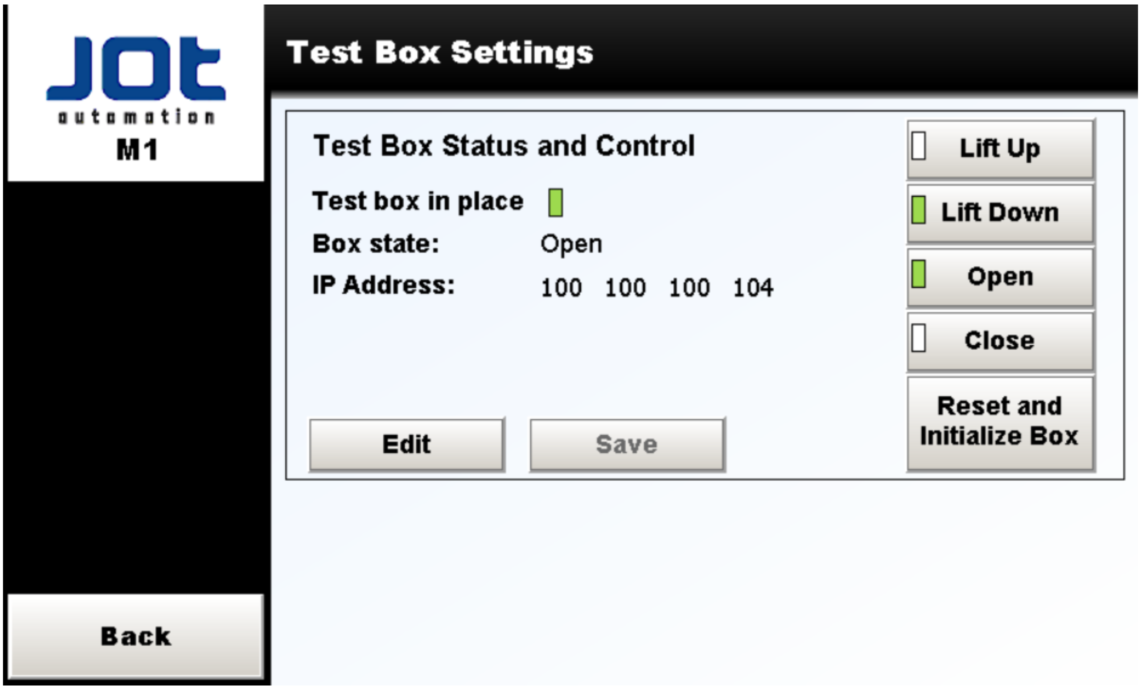
PICTURE 16. UINT value 14 is moved to the GUI_InfoText variable when the conditions are true.

Machine in DEMO mode, waiting to START	14

PICTURE 17. Message assigned to the value 14 in the HMI text library.

For example, picture 16 shows a line of ladder code that controls the display of a message assigned to the value 14. Picture 17 shows the message assigned to the value 14 in the HMI text library. When the assigned conditions are true the UINT value 14 is moved to the GUI_InfoText variable and the associated message is shown on the HMI. In the example pictures the conditions require the machine to be in demo mode and idling: neither auto nor pass through modes have started.

A test box IP monitoring and control screen (picture 18) was created. The user is able to modify the PLC's current test box connection settings from this page without having to have access to the PLC program itself. This enables changing of test boxes by people without access to the tools required. Previously access to either CX-Programmer or the test box HTML interface was needed to enable communication.



PICTURE 18. Test box settings window.

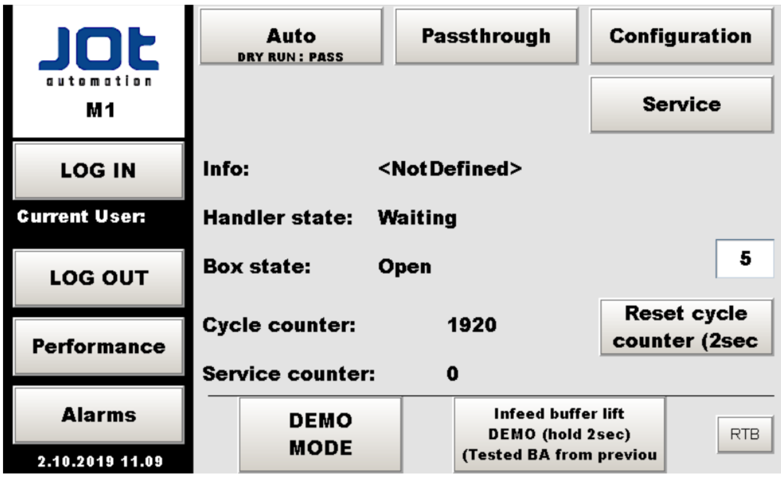
This screen can be accessed from both configuration and service screens. Code was created to enable returning to the screen the user came from. This page can also be used to troubleshoot, setup and test the functionality of a newly implemented or a just connected test box.

The user can now keep track of the current PLC and HMI program versions by viewing it on the GUI information screen shown in picture 19. This feature helps with troubleshooting as the user can quickly identify whether the machine is using the correct software versions or not. The PLC version information is fetched from the PLC software and the HMI version is fetched from the current HMI project filename using a system tag in iX Developer.

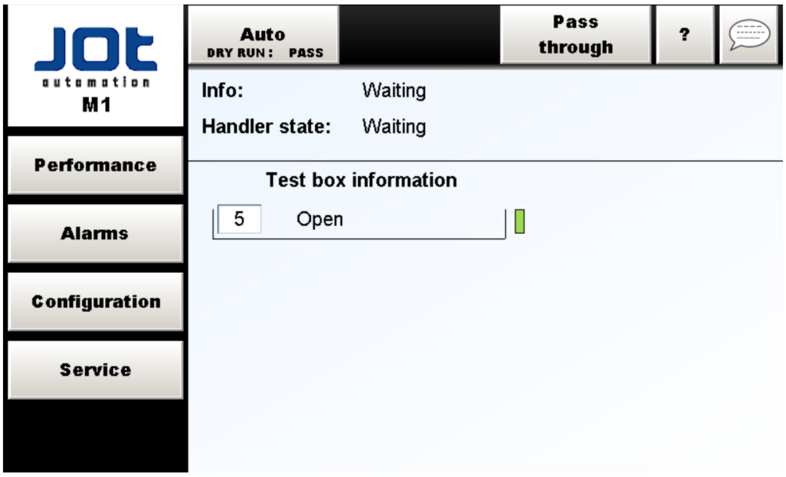
PICTURE 19. GUI information screen.

The HMI software updates were tested alongside the PLC software updates to ensure correct functionality. The results of the test confirmed the success of the HMI software update. Tests were done for configuration, service and normal operation with the screen. Additions made to the HMI work as expected.

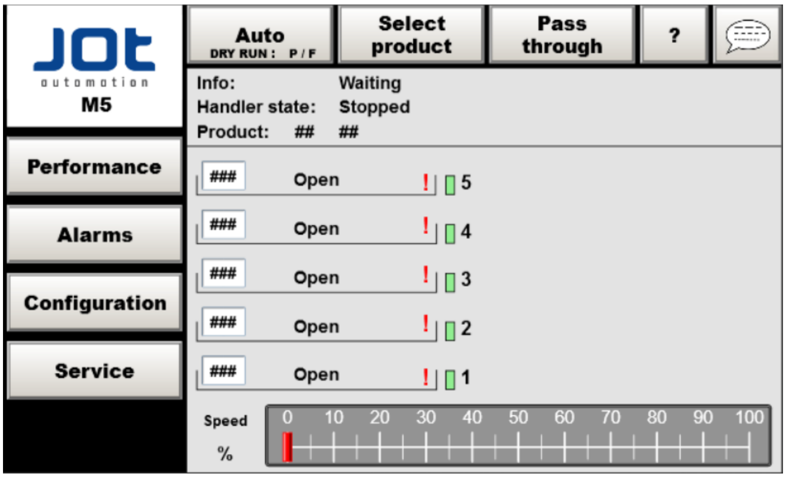
The update resulted in a HMI software which meets the criteria defined by the commissioner at the beginning of the project. The screens graphical interface resembles that of the M5 where applicable and the program is ready to use. The following page features images of the original M1 home screen (picture 20), the home screen after the HMI update (picture 21) and a picture of the M5 home screen (picture 22) for reference.



PICTURE 20. Old M1 home screen.



PICTURE 21. New M1 home screen.



PICTURE 22. M5 home screen.

5.4 PLC Software Update

The M1 PLC software update's goal was to further develop the latest version of the M1 software according to released software made for a client project, the M5's solutions, and a previous to-do list. Along with these sources of modifications, the program was changed based on things found lacking during the development and testing processes.

The updates were done using a demo version M1 software as the baseline. They were executed using CX-Programmer and partially overlapped the HMI software update. Improvements were made up until the time the machine left based on feedback given. The updates were tested and confirmed to be successful at the start of October before the machine left for the exhibition.

The changes done during the project were documented in an external file and internally in a program section reserved for version information (picture 26). The documentation can be used to track the changes made to the program if further development is done.

The software update resulted in multiple new features, bug fixes, quality of life enhancements, further integration of the HMI and a cleanup of the program.

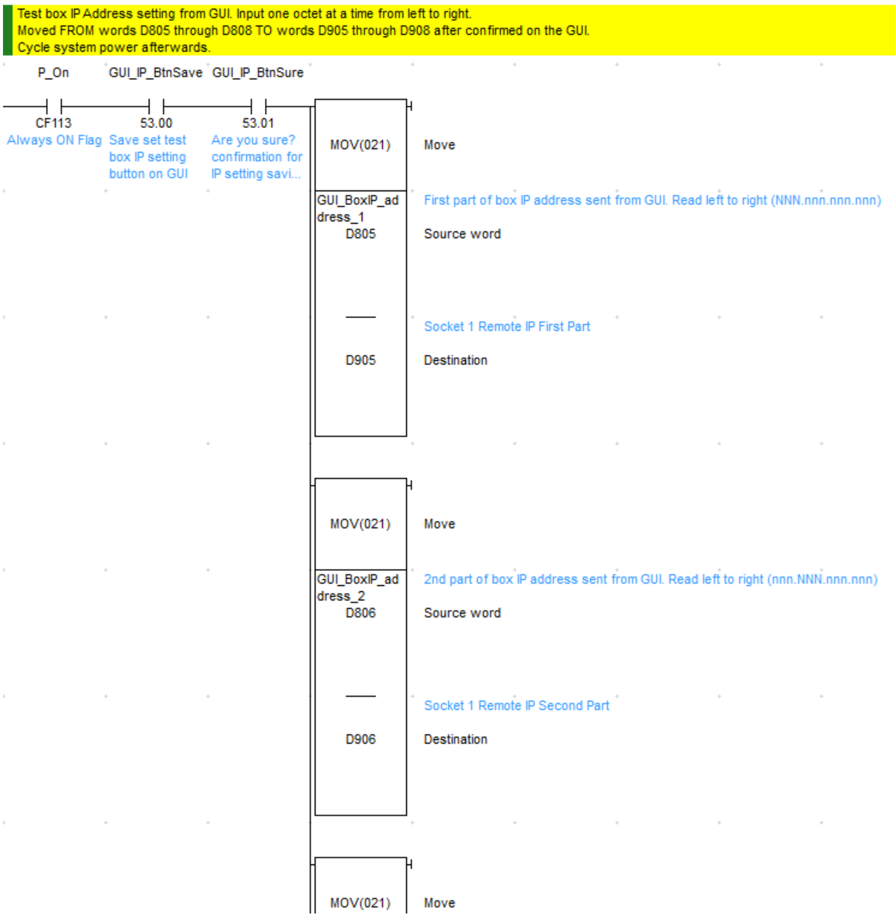
The cleanup was the first part of the update process. The machine's I/O list was reviewed, unnecessary symbols were removed and the program was screened for redundant code.

The I/O list cleanup was done manually by cross-referencing and examining every symbol on the list individually. The removal of unneeded program code was done much the same way. All program sections were screened for code that had been previously bypassed or otherwise deemed unnecessary and these parts were removed. The machine's functionality was tested after and during the cleanup and before proceeding with the next updates to ensure nothing vital was deleted. Backups were taken during the process to ensure reverting changes would be possible if need arose.

The modifications done on the control of handler and test box states and machine information are among the quality of life updates. The updated texts more accurately resemble the current state of the system. These are controlled in the PLC program and shown on the HMI screen as demonstrated in pictures 16 and 17 in chapter 5.3. Updates to states include resetting them if an alarm triggers, preventing incorrect information from showing and making sure that an undefined message is never shown on the HMI.

The machine's demo mode was developed further. The demo mode can be used to test the M1's functionality by returning the current DUT from the outfeed to the infeed conveyor after testing or otherwise passing through. The modifications and additions resulted in multiple bugs being removed and enable the testing of entire machine functionality.

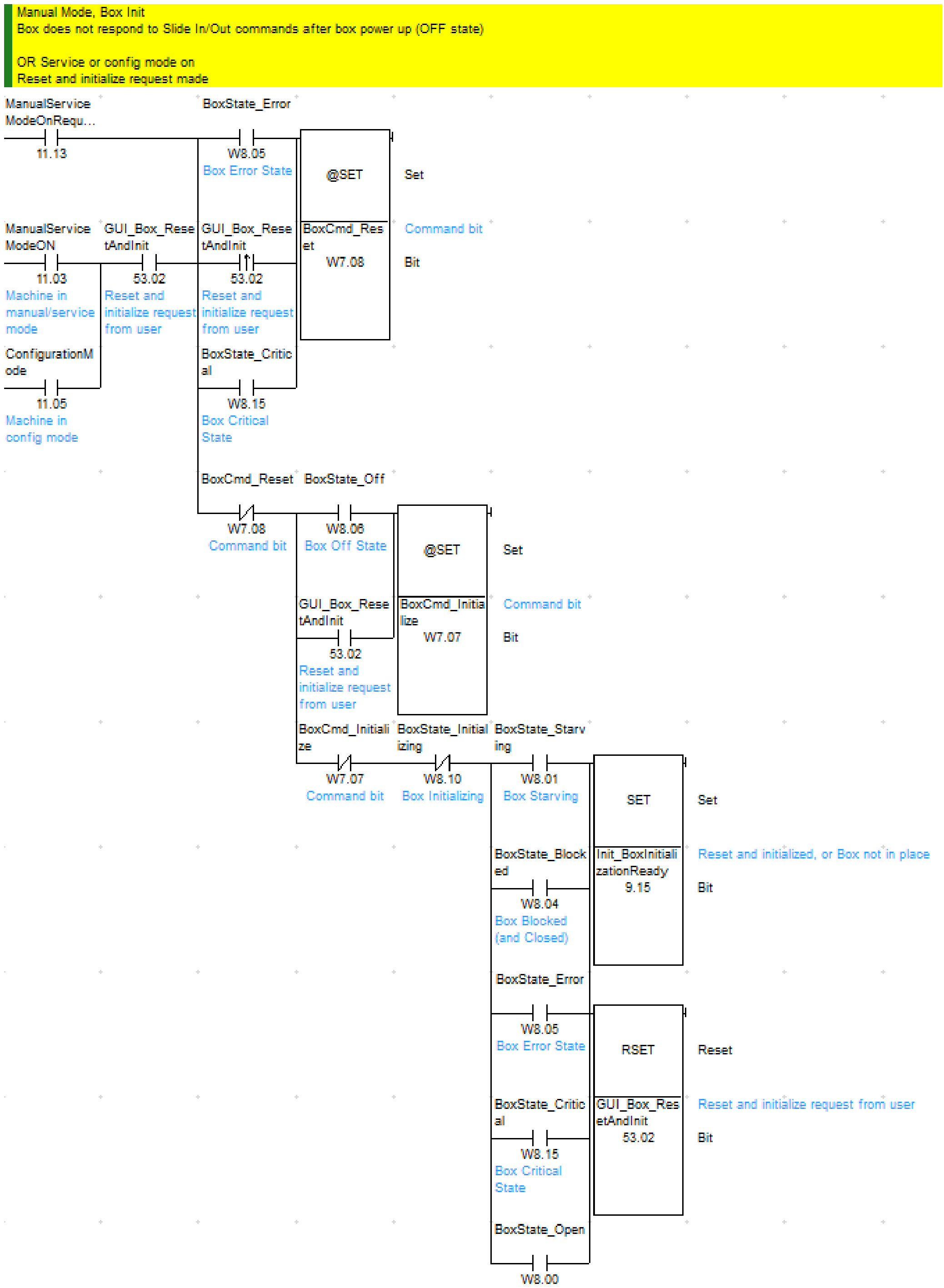
Problems solved in regards to demo mode functionality included modifications to machine stopping procedures upon exiting demo mode and when using pass through demo mode, added clarity for the user that the machine is in demo mode with state texts and resetting demo mode when the machine is stopped. Running the machine in pass through demo mode is now possible and no longer causes any unexpected behavior.



PICTURE 23. Test box IP address modification.

The test box IP address modification (picture 23) is done by moving IP address information input on the HMI from the input words, D805 through D808, to the memory areas the IP address is stored in on the PLC, D905 through D908. The IP address is stored in four 16-bit data areas (words). It is input on the HMI screen (picture 18) in string format, split into four octets with dots removed, translated into hex values and stored one octet at a time. The memory data is retained to prevent loss of power from clearing the machine's settings. Changing the box IP address was previously done by directly accessing the PLC memory and modifying it there.

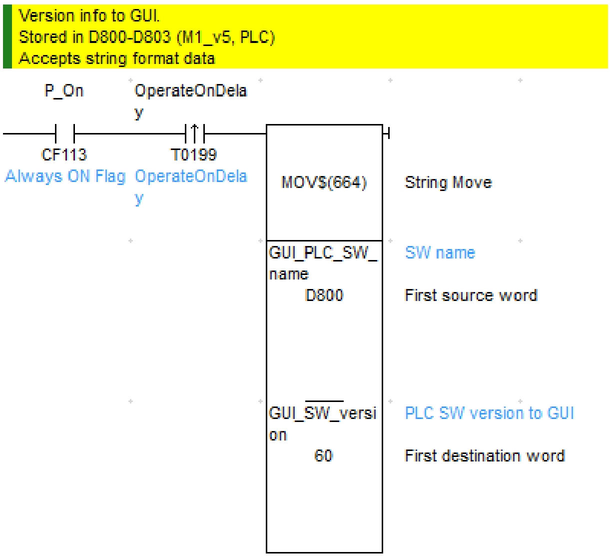
A test box reset and initialize command was added to the screen shown in picture 18. At the press of a button a prompt is given to the PLC to command the test box to reset and re-initialize itself if the required conditions are met. (Picture 24.)



PICTURE 24. Test box reset and initialize prompt.

The reset and initialize command can be used as a part of machine commissioning, troubleshooting and changing of test boxes along with the IP address modification function.

The current PLC software version info is stored in the PLC memory and moved to a word in string format to be read on the HMI (picture 19). The information is updated on the OperateOnDelay bit's rising edge pulse (picture 25.)



PICTURE 25. Software version information to HMI.

The character support for BCR code length was updated from 40 to 120 characters according to the work done in a previous client project. This was done by modifying the code reader's programming in the PLC software and altering the associated memory data areas accordingly. The change enables processing of DUTs with long product codes if the optional code reader is enabled.

The pass through mode update enables the starting of the machine in said mode if a test box is in place and if the box's slide is open. Previously the user had to manually close the test box's slide prior to initialization in case it was in any other position. Changes done to pass through mode prevent the machine from becoming stuck during the pass through initialization if a test box is in place.

The machine now initializes itself in pass through mode as it would in auto mode: drives the middle lift down, closes the test box slide, drives the middle lift up, looks for unknown DUTs on the conveyor and drives them out as fail DUTs if any are found. After this normal pass through functionality is carried out. The mode was also changed so that it no longer stops the DUT at the infeed conveyor stopper unnecessarily.

Numerous bug fixes were done during the update. The changes were documented internally (picture 26) in the software itself and in an external changelog document.

```
12.09.19
M1_v4 maijmat
-Added initialization when starting machine in Passthru mode and Test Box is open.
 -Added waiting for test box init ready (9.15) to initialization rung 8
 -Bypassed Reset of Passthrough Mode if testbox is in place and not closed.(RaM passthru mode OFF rung)
 -Added starting of test box init in passthru mode if test box in place. Works the same as in auto mode.
 -Allows flexible change of conveyor to passthru w/o removing testbox in production line.
-Fixed a bug where test box wouldn't engage in auto mode dry run.
-Fixed (bypassed) bug where after an alarm GUI_BootReady goes to 0 and machine/GUI stop responding before power is cycled.
 -IF gui screen <> 0, then GUI has booted and operateON delay timer can run.
-Fixed bug where demo mode won't stop if STOP is pressed while DUT is returning to infeed.
-Passthru in Demo mode should work properly now.
 -Doesn't stop program cycle after first return to the infeed conveyor stopper; doesn't activate stopper at all in passthru mode.
 -DUT coming from Previous needs BA from previous signal to start conveyor even in demo mode: this is to prevent conveyor timeout upon starting machine in passthru demo mode.
```

PICTURE 26. Example of documentation in software version information.

The corrections included the following and more:

- Making sure that the machine's operation mode resets correctly when the machine is stopped.
- Ensuring that when warnings or alarms are triggered during operation, configuration or service they do not leave the operator and the machine stuck due to modes not resetting. Alarms and warnings change the screen visible on the HMI to the alarm viewer screen. If the current operation mode is not reset from the background, it will stay on and incorrectly prevent further operation until complete reset of the machine is done.
- Guaranteeing that DUT's do not get stuck in between actions if the machine is stopped or the current operation mode is changed. Ongoing movements are now finished and operation is stopped when no DUTs are moving and no tests are in progress.
- The flag signaling that GUI boot process is ready is bypassed if the HMI screen value is other than 0. This prevents the Operate On -permission from turning false due to updating the PLC software and not cycling the HMI's power afterwards. GUI boot ready flag is not retained after loss of power and is only set to true upon leaving the GUI boot screen.

- A fix for a bug where the infeed conveyor running backwards would not stop at the press of stop button when returning DUT from outfeed to infeed conveyor in pass through demo mode was implemented. The machine now stops operation when the DUT reaches the infeed conveyor sensor as it should.

The PLC software's primary tests were done using the demo mode functionality and the dry run mode functionality. The tests were executed with the machine in automatic mode.

What this means is that the DUTs in the machine were driven as normal to the test box (auto mode), the commands used for engaging a DUT for testing in the connected test box were issued as normal without actual testing happening (dry run mode) and after the test was done the DUT was driven out to the outfeed conveyor and then returned to the infeed conveyor for another testing sequence to take place (demo mode).

Along with the auto mode tests, the machine's new pass through mode functionality was tested similarly by making use of the demo mode for DUT returning.

Tests done on the software ensured that no problems should occur during normal operation. The final testing done at the end of the project made certain that the program updates were a success. The machine now works as intended, the goals set were met, the operation control and monitoring using the machine's HMI is now clearer and the software is ready for deployment and further development.

5.5 User Manual Update

The user manual update for the M1 was carried out as the last part of the project. The manual update was done based on the latest version of the user manual including the HMI from January 2018. A copy of this version was printed out and changes to be made were plotted on paper upon review.

The review found multiple parts lacking or containing old information. The errors were corrected and the manual was brought up to date to match the current functionalities of the M1. After the needed updates were made on the paper copy they were translated into a digital form categorized by page, chapter and paragraph to provide clarity.

The revision of the manual consisted of changing, removing and making additions to the base manual. Changes made included correcting typographical errors, changing word forms, adding functional descriptions and updating pictures associated with the chapter to match the current form. Updated pictures consist of various modified HMI screens. Modifications to functional descriptions revolve around the changes done to the HMI software and the references to old versions' manual mode buttons and switches. Changed parts of text were marked with italicized font for clarity.

Parts of the manual referencing out of date software functionality were removed. This includes things such as having to manually close a potentially open test box slide to start machine in pass through mode. Removed parts of the text were marked with bold letters.

An additional chapter regarding machine configuration was made for the added test box settings –functionality. This chapter includes a brief on how to configure and control the test box with the HMI. Configuration mainly concerns changing the IP address of the connected test box that the PLC tries to connect to.

After the writing of the compilation document was done, it was reviewed for mistakes and sent to the document makers for future implementation along with a file containing all of the pictures taken and the original paper copy on which the changes were planned.

6 CONCLUSION

The project and the thesis were finished in time and largely according to plan. The goals defined in the beginning were met and all of the updates were successful. No large problems arose during the course of the project, although some parts were relatively challenging. The machine testing was executed without a return conveyor/lift setup in the built-in demo mode.

At the start of the project it became evident that the work done on the M1 would face a hard time limit as the machine was to be shipped to an exhibition for demo purposes around a month after beginning the majority of updates. The limited timespan, however, was not much of an issue and all of the planned updates and testing were carried out on time. The limited time and real deadline could even be seen as a good thing as it positively affected the project's pace and created a great incentive to finish the productization project effectively as testing of software would become much harder without the actual machine at hand.

As for meeting the goals of the project most things were successful. The end results of the machine specifications/flowchart updates and the user manual update differ from what was defined and/or expected at the beginning. The flowchart update differs as upon review the flowcharts were found to match the machine's functionality well for most part. The user manual update was not implemented in the actual document and the parts to-be-updated were instead compiled on an external document.

The biggest difference between the definition and the end result of the specification update was that the integration of an HMI had not largely affected the core functionality of the machine, only the operation. Operation along with service and configuration is not necessary to be featured on the flowcharts. Still, the goal of reviewing and updating the machine specifications if necessary was met. Minor additions were made in the form of naming changes and demo mode additions such as returning DUT from out to infeed conveyor.

The user manual review and update resulted in a comprehensive document detailing the changes needed to match the current machine functionality. The changes could possibly have been done to the original work document but were done instead in a separate document to prevent any mistakes in implementation caused by a writer not familiar with the company documentation guidelines.

Both the PLC and the HMI updates were successful and the defined goals were met. The resulting software is ready-to-use in a production line setup. However, it should first be validated using actual DUTs and a return conveyor/lift setup. The tests done on the machine to date show no problems with functionality which indicates that no problems would surface in an actual test environment either.

Upon review of both the HMI and the PLC software some further minor changes were made according to the feedback from the supervisor of the bachelor’s thesis and the software were confirmed to be in working order.

The M1’s new HMI GUI now looks and functions largely the same as the M5 GUI where applicable, as was defined. The elements are the same in terms of sizing, coloring and placement where possible. Some additional graphics and quality of life type updates were done in the process, too. Learning to program with C# was among the key competences needed for creating and finishing the updated HMI.

Navigation in the HMI is now more intuitive and should feel familiar to a user accustomed to the M5 and vice versa. In a production environment this translates to a lowered machine update threshold as the user will already be somewhat familiar with how the UI and setup works before ever touching the other machine.

The PLC update resulted in a software that is up to par with other released versions and includes multiple new, enhanced and modified features. The software is now in working condition and can be used in further development endeavors or as the machine’s primary software. However, it is not compatible with a version of the M1 that does not employ a HMI.

Due to lack of time to do in-depth testing of the new software solutions, larger HMI not being mechanically integrated yet, and the fact that the users operating the demo machine at the exhibition it was sent to were already accustomed to the old software versions, at the end of the bachelor's thesis project the old software solutions were returned to the machine from backups.

The wide range of different changes done to the PLC software not only resulted in a better-than-previous software solution but also in a great learning experience for myself. This applies equally to the work done on the HMI. The PLC software's functionality was updated to be comparable to that of the M5 where applicable, although many further development paths and future possibilities are still open.

6.1 Further development possibilities

Further development possibilities are various and constantly change. As the M1 is the "little brother" of the M5 in terms of functionality, things to implement will most likely be largely based on the progress done in the M5's development. This includes upgrades to both the HMI and the PLC software and any possible hardware modifications yet to be planned.

Development ideas and discussion included things such as PLC hardware updates, feature updates, problems possibly emerging from updates, problems related to updates and whether or not some things should or should not be taken into account.

The machine's PLC could be updated and upgraded from the CP1L-EM to the NX1P2 to further enhance machine compatibility and simplify the software development between the M1 and the M5.

Changing the PLC used would enable development and debugging in the same IDE making exporting and importing new features between machines much easier. This could also lower license costs and future-proof the machine. A question arose as to whether the rework would be worth the cost and ould the additional modernization work on the machine's software ever pay itself back? Would the benefits outweigh the negative aspects?

Updating the PLC as aforementioned would mean recreating the entire software in Sysmac Studio. After the software work is done, any future updates could be directly implemented between machines employing the N-series PLC's when necessary. Further software development could be simplified.

Service and commissioning could be made easier by a more modern solution. The solution would be future-proof in the sense that the newer PLC and IDE are almost guaranteed to have a longer product lifetime and product support than the older solutions. This translates to greater maintenance and replacement security in case problems arise.

The PLC hardware upgrade would engage designers specialized in different sections of machine design to rework a machine meant to be "entry-level". Would the time required for mechanical, software and automation translate into an increase in machine price? Would that in turn change the way the machine is perceived? Would it decrease interest?

As the machine is meant to be a low threshold update to a low volume production line the importance of price as a suitability factor increases. An entry-level machine is not expected to include as many features as the higher-end products in the M-series line up. Price optimization is a must.

Mechanical integration of the larger HMI screen is pending at the time of writing of this document. It'll require the top part of the machine's chassis to be increased in size and the schematics to be updated as there currently is not enough space to fit the bigger screen. The possible integration of the larger screen will enhance the user experience drastically (own opinion). The software to enable the use of the larger screen is already ready for use and proven to work.

SOURCES

1. JOT Automation Ltd. About Us. Available at: https://www.jotautomation.com/about.html. Access date: 16.8.2019.

2. JOT Automation Ltd. M1 Test Handler. Available at: https://www.jotautomation.com/products/jot-m1-test-handler.html. Access date: 19.8.2019.

3. JOT Automation Ltd. M1 Test Handler Datasheet. Available at: https://www.jotautomation.com/media/datasheets/jot_m1-test-handler_datasheet.pdf. Access date: 19.8.2019.

4. IPC SMEMA-9851. 2007. Mechanical Equipment Interface Standard. Available at: https://ocmmanufacturing.com/wp-content/uploads/2016/01/IPC-SMEMA-9851.pdf. Access date: 19.8.2019.

5. Omron Corporation. CP1W Expansion Units. Available at: https://industrial.omron.fi/fi/products/cp1w-expansion-units. Access date: 19.8.2019.

6. Beijer Electronics. X2 Base 7 HMI product page. Available at: https://www05.beijerelectronics.com/en/Products/Operator___panels/X2___base/X2___base___7. Access date: 19.8.2019.

7. Omron Corporation. CX-Programmer Product Page. Available at: https://industrial.omron.fi/fi/products/cx-programmer. Access date: 20.8.2019.

8. PLCopen. IEC 61131-10 preview. 2019. Available at: https://www.plcopen.org/sites/default/files/downloads/iec_61131-10_preview.pdf. Access date: 21.8.2019

9. IEC. About the IEC. Available at: https://www.iec.ch/about/. Access date: 21.8.2019

10. PLCopen. IEC 61131-3: Programming Languages. 2013. Available at: https://www.plcopen.org/iec-61131-3. Access date: 21.8.2019.

11. Beijer Electronics. X2 Panel Image Manual MAEN205B. Available at: https://www.beijerelectronics.tw/en/Products/Operator___panels/X2___base/X2___base___7#downloads. Access date: 22.8.2019.

12. SFS-EN 60529 + A1. 2011. Sähkölaitteiden kotelointiluokat (IP-koodi). Degrees of protection provided by enclosures (IP code). Page 1. Helsinki: Suomen Standardisoimisliitto SFS.

13. Kontiokoski, Joona – Maijanen, Mathias. 2019. OMRON PLC Tool Comparison. JOT Automation Ltd internal document.

14. JOT Automation Ltd. M5 Test Handler Datasheet. Available at: https://www.jotautomation.com/media/datasheets/m5_test_handler_fact_sheet_web_01.pdf. Access date: 27.8.2019.

15. JOT Automation Ltd. M5 Test Handler Product page. Available at: https://www.jotautomation.com/products/jot-m5-test-handler.html. Access date: 2.9.2019.

16. PTC Inc. Omron FINS Ethernet Driver manual. Available at: https://www.kepware.com/getattachment/7551f988-97df-426b-a00d-e8731a5b7607/omron-fins-ethernet-manual.pdf. Access date: 11.9.2019.

17. RS Components Ltd. Picture of Omron CP1L-EM40. Available at: https://uk.rs-online.com/web/p/plc-cpus/8211775/. Access date: 18.9.2019.

18. SFS-EN 60529 + A1. 2011. Sähkölaitteiden kotelointiluokat (IP-koodi). Elements of the IP Code and their meanings (p. 25). Helsinki: Suomen Standardisoimisliitto SFS.

19. Omron Corporation. CP1L-E Datasheet. http://industrial.omron.com.br/uploads/arquivos/CP1L_E_datasheet-EN_201212.pdf. Access date: 2.10.2019.

20. Inductive Automation. What is HMI? https://www.inductiveautomation.com/resources/article/what-is-hmi. Access date: 3.10.2019.

21. Aveva. What is HMI? Human Machine Interface. https://sw.aveva.com/monitor-and-control/what-is-hmi. Access date: 3.10.2019.

22. Bolton, William. 2015. Programmable Logic Controller (6th Edition). Newnes. E-book available at: https://learning.oreilly.com/library/view/programmable-logic-controllers/9780128029299/?ar. Access date: 3.10.2019.

4.2 Elements of the IP Code and their meanings

A brief description of the IP Code elements is given in the following chart. Full details are specified in the clauses indicated in the last column.

Element	Numerals or letters	Meaning for the protection of *equipment*	Meaning for the protection of *persons*	Ref.
Code letters	IP	–	–	–
First characteristic numeral		Against ingress of solid foreign objects	Against access to hazardous parts with	Cl. 5
	0	(non-protected)	(non-protected)	
	1	≥ 50 mm diameter	back of hand	
	2	≥ 12,5 mm diameter	finger	
	3	≥ 2,5 mm diameter	tool	
	4	≥ 1,0 mm diameter	wire	
	5	dust-protected	wire	
	6	dust-tight	wire	
Second characteristic numeral		Against ingress of water with harmful effects	–	Cl. 6
	0	(non-protected)		
	1	vertically dripping		
	2	dripping (15° tilted)		
	3	spraying		
	4	splashing		
	5	jetting		
	6	powerful jetting		
	7	temporary immersion		
	8	continuous immersion		
Additional letter (optional)		–	Against access to hazardous parts with:	Cl. 7
	A		back of hand	
	B		finger	
	C		tool	
	D		wire	
Supplementary letter (optional)		Supplementary information specific to:	–	Cl. 8
	H	High voltage apparatus		
	M	Motion during water test		
	S	Stationary during water test		
	W	Weather conditions		

Source of the picture: see reference 18.

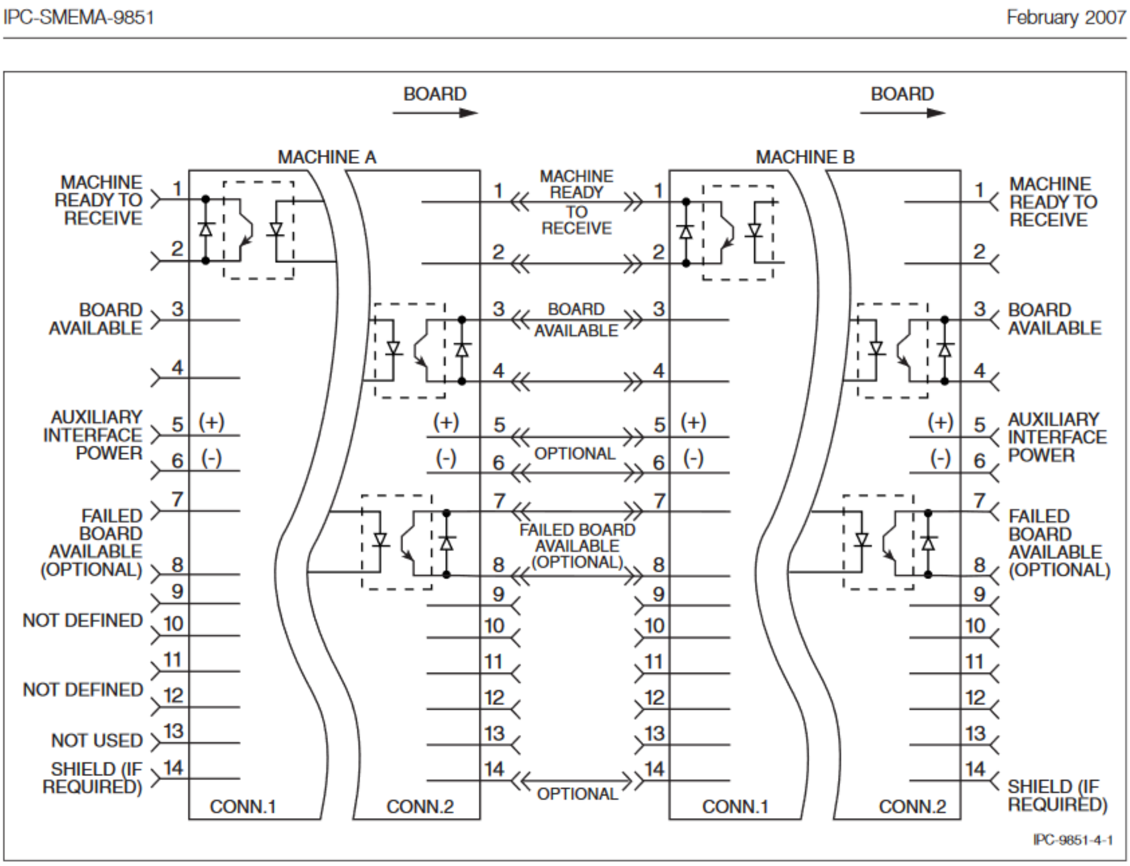


Figure 4-1 Electrical Interface Schematic

SMEMA electrical interface schematic.

Source of picture: see reference 4.

Table 4-1 Electrical Interface Connector/Cable Functional Description

Connector/Cable	Function	Condition	Description
Pair 1-2 (Note 1)	Machine Ready to Receive	Contacts Closed (Notes 2,3)	Machine is ready to receive next board.
Pair 3-4 (Note 1)	Board Available	Contacts Closed (Notes 2,3)	Machine has a good board ready to send. All boards are considered to be "good" if the Board Fail option is not being used.
Pair 5-6	Auxiliary Interface Power (optional)		Available; user to document purpose and operating parameters.
Pair 7-8 (Note 1)	Failed Board Available (Optional)	Contacts Closed (Notes 2,3)	Default (no connection or, if used, contacts are open) is that the incoming board is good and suitable for use. Optional use is to provide closed contacts when it has been determined that the board should stop transfer or be diverted. In such cases, these contacts **shall** be closed in lieu of (and not in addition to) the normal Good Board Available contacts.
Pair 9-10	Not defined		Available; user to document purpose and operating parameters.
Pair 11-12	Not defined		Available; user to document purpose and operating parameters.
13	Not used	Not used	Not used.
14	Cable shield (Optional)		Cable shield attachment if required; follow good engineering practices (connect at only one end).

Note 1: Minimum requirements are to switch 30 Vdc, 10 mA.
Note 2: At 10 mA, the output "LOW" or contact closure **shall not** exceed 0.8 Vdc.
Note 3: Assure proper polarity if using optional optical isolator.
Note 4: Existing equipment not built to this standard may require modified pin-out on the connector and/or the interface cable.

SMEMA electrical interface functional description.

Source of picture: see reference 4.

Table 4-2 Signal Logic

Board transfer occurs when Machine A has a BOARD AVAILABLE (contact closed), and Machine B is MACHINE READY TO RECEIVE (contact closed).
The signals can occur at any time, but board transfer does not occur until both contacts are closed.
The BOARD AVAILABLE signal from Machine A will remain closed until the board has left Machine A.
The MACHINE READY TO RECEIVE signal will remain closed until Machine B has positive control of the board.
Board transfer cannot occur again until each signal opens for at least 50 ms.
Optional: Once both Machine A and Machine B signals are closed, and the board has neither left A nor arrived at B, an error message will be generated (to be defined by users).

Table 4-3 Timing

Time	Action/Condition
T_0	Up-line **board not available**; down-line **machine not ready to receive**.
T_1	Up-line machine has a **board available to send**; down-line **machine not ready to receive**.
T_2	Up-line machine has a **board available to send**; down-line **machine ready to receive, transfer** starts.
T_3 (variable)	Board has completely left control of up-line machine; still moving into down-line machine.
T_4	Transfer complete; down-line machine is completely in control of board. **Board not available**, down-line **machine not ready to receive**.
T_5	Up-line **board not available**, down-line **machine ready to receive**.
T_6	Up-line machine has a **board available to send**, down-line **machine ready to receive, transfer** starts.
T_7 (variable)	Board has completely left control of up-line machine; still moving into down-line machine.
T_8	Transfer complete; down-line machine is completely in control of board. **Board not available**, down-line **machine not ready to receive**.
T_9	Up-line **board not available**, down-line **machine ready to receive**.
T_{10}	Up-line machine has a **board available to send**, down-line **ready to receive, transfer** starts.

SMEMA signal logic and timing tables.

Source of picture: see reference 4.

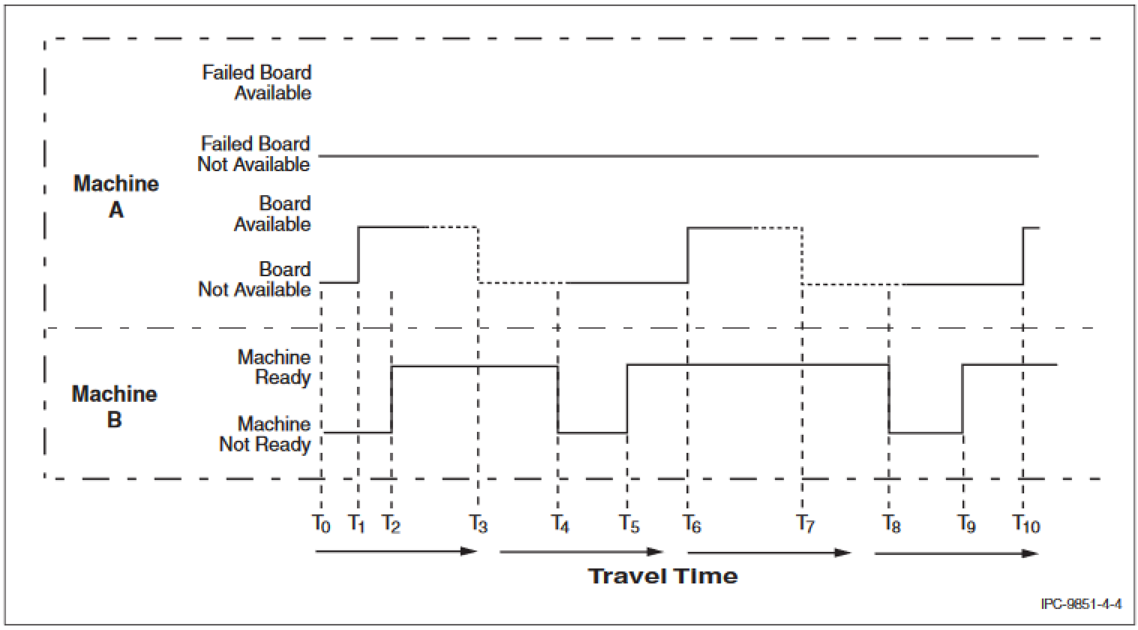


Figure 4-4 Timing Logic Diagram for Normal Transfer

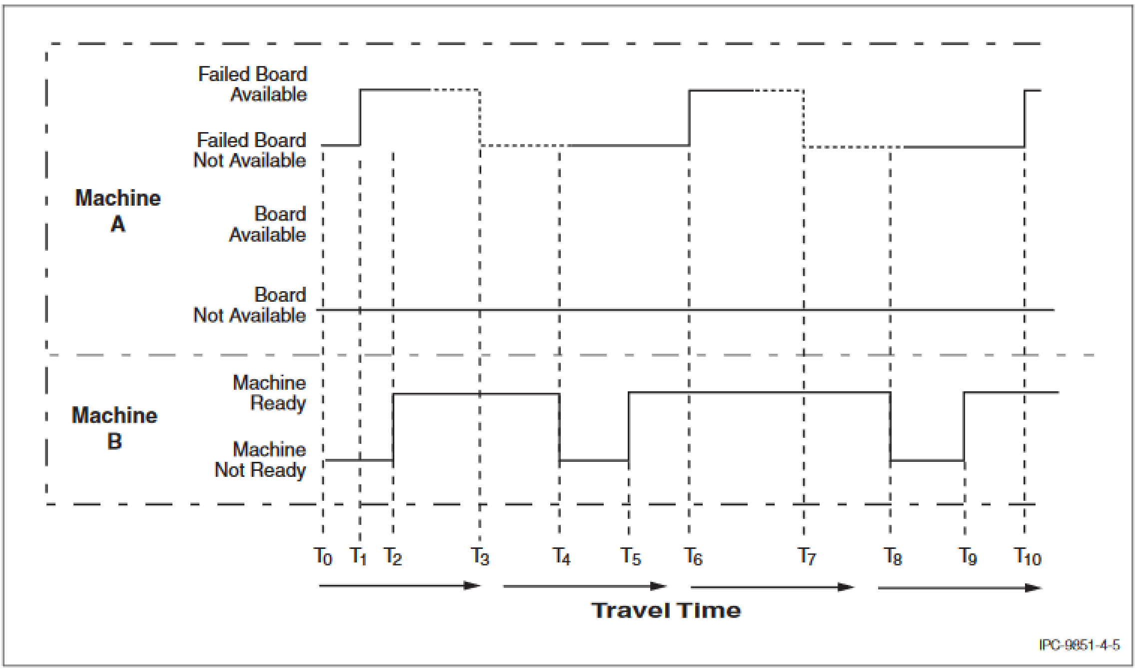


Figure 4-5 Timing Logic Diagram for Failed Board Option

SMEMA timing diagrams.

Source of picture: see reference 4.