MECHATRONICS DESIGN FROM ZERO TO ONE (STRUCTURED LOGIC DESIGN TO PROGRAM LOGIC OF LADDER DIAGRAM FOR PLC)

NOR SYAHMI BIN ZAINAL ABIDIN

ELECTRICAL & ELECTRONICS ENGINEERING
UNIVERSITI TEKNOLOGI PETRONAS
JUNE 2009

MECHATRONICS DESIGN FROM ZERO TO ONE (STRUCTURED LOGIC DESIGN TO PROGRAM LOGIC OF LADDER DIAGRAM FOR PLC)

By

NOR SYAHMI BIN ZAINAL ABIDIN

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

MECHATRONICS DESIGN FROM ZERO TO ONE (STRUCTURED LOGIC DESIGN TO PROGRAM LOGIC OF LADDER LOGIC FOR PLC)

By

Nor Syahmi Bin Zainal Abidin

FINAL PROJECT REPORT

Submitted to the Electrical & Electronics Engineering Programme
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

____
AP Dr. Nordin Saad
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

12/6/09

Nor Syahmi Bin Zainal Abidin

ABSTRACT

The purpose of this work is to study and analyze the methods use to design logic of ladder diagram for PLC-based controller in automated manufacturing systems. Previous method employed to design the logic of ladder diagram does not show clearly on how it is done step by step, widely based on the programmer's experience and their intuition. The methods proposed namely method A and method G hopefully can help the programmer especially the new programmer to design the ladder logic systematically and efficiently while at the same time reduce the time consume to program it. This systematic logic design can help the programmer to trace back their program for debug purpose. A step by step instruction is provided in this paper for both method A and method G. Few basic sequence are tested Finally, a case study on packaging process is provided to illustrate the design procedure of the proposed methods. In the same time, author will also explore the capability of the Automation Studio software.

ACKNOWLEDGEMENT

The author wishes to express his deepest gratitude to all the bodies that contributed to the success of the research. Following are the person or group contributed to the project.

1. Supervisor: AP Dr. Nordin Saad
2. Mr. Azhar Zainal Abidin, Instrumentation Lab Technician, Mr Arofik, Graduate Assistant
3. Electrical and Electronics Engineering Department of University Technology PETRONAS
4. Family and friends

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

The arrival of Programmable Logic Controller (PLC) in the industry in 1969 is considered as one of the major achievement especially in industrial automation and since that "it has become the most common choice for manufacturing control" [11]. According to (Lauzon et. al. 1997 p. 91) "PLC is the most widely employed industrial process control technology today". "The National Electrical Manufacturers Association (NEMA) defines PLC as a digitally operating electronic apparatus which uses a programmable memory for the internal storage of instructions by implementing specific functions, such as logic, sequencing, timing, counting, and arithmetic to control through digital or analog I/O (Input/Output) modules various types of machines or processes."[16] .A general block diagram of a PLC is illustrated below in Figure 1.

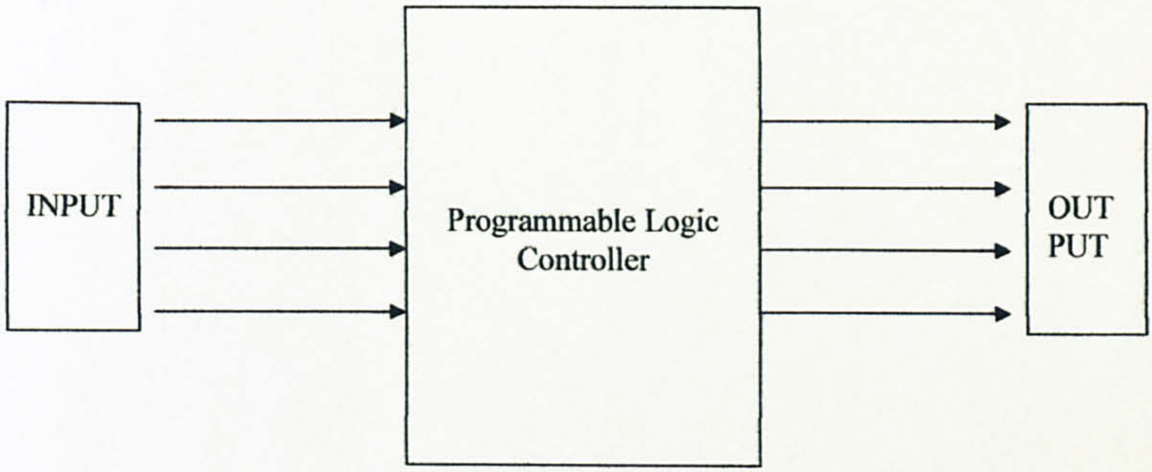


Figure 1: PLC Block Diagram

"As the PLC receives input conditions (i.e. voltage or no voltage), it examines them against the programmed code within the PLC and then executes the proper outputs associated or specified within the programmed code" (Wright, 1998, p.1). Using PLC as one of the control element for industrial automation has a lots of advantages;

- **Low Cost** :The system itself required a small amount of electrical design which can lead to reduce in design cost
- **Flexibility**: It has a set of input and output devices that is capable to be used with many industrial pilot device and controls [16].
- **Simplicity**: Reduce the complexity of a system as opposed to the old system (hard-wired relays, stepping switches and drum programmers) [16].
- **Discrete control**: It is a good controller choice for automation process (material stamping, material handling, sequencing, etc.) which has many discrete devices such as limit switches, motor starters, etc.
- **Error Correction** : By comparing to a hard-wired relay system, PLC is far more easier if the user need to make any correction or amendment to the system as simple as reprogramming the logic [16].
- **Simulation**: A PLC programming can be simulated and evaluated in a lab environment before it is applied in the field hence saving the design cost.

1.1.1 IEC 1131-3

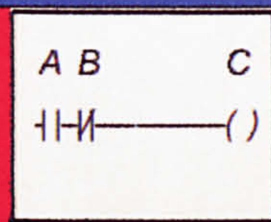
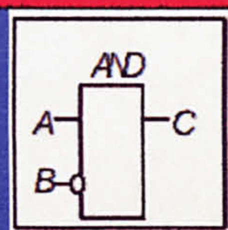
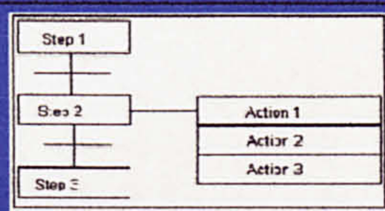
IEC 1131 is the international standard for programmable logic controllers (PLCs) set by International Electrotechnical Commission (IEC).Now it is known as IEC 61131 after the change in numbering system by IEC. According to [19], "IEC 1131-3 is the third part of the IEC 1131 family" and consists of:

- IEC 61131-1 Overview
- IEC 61131-2 Requirements and Test Procedures
- IEC 61131-3 Data Types and Programming Languages
- IEC 61131-4 User Guidelines
- IEC 61131-5 Communication
- IEC 61131-7 Fuzzy Logic

**Notes: According to PLCopen – an organization active in Industrial Control; there will be a new standard with the code name IEC 61131-8 (Guidelines for the application and implementation of programming languages) which is now under the Working Draft 3 stage.*

Any PLC that is IEC 61131-3 compliant means that it allows multiple languages to be used on the same PLC. This is one the primary benefit of this standard. It allows the programmer to choose which languages best suit their needs. Five programming languages are defined in this standard [11]. This is visualized in the table 1 below.

Table 1: PLC Programming Language

Programming Language	Type of language		Example
Instruction List, IL	Textual	Low level language (mnemonic programming)	LD A ANDN B ST C
Structured Text, ST	Textual	High level language (A BASIC like language)	C=A AND NOTB
Ladder Diagram, LD	Graphical	Low level language (based on relay logic – On and Off)	
Function Block Diagram, FBD	Graphical	High level language (graphical dataflow of a program)	
Sequential Function Chart, SFC	Graphical	High level language (similar to Petri-Nets)	

A bulk number of sequential control applications such as the packaging of foods in the factory or chemical plant currently employed the programmable logic controller as the control element of the said applications [12]. From the five programming language listed in the IEC 61131-3 standard, ladder logic is the most employed language for programming a PLCs to be used in sequence system [4], [5], [6], [13], [15], [17], [20]. In the earlier days of electrical control, they use relay as based for controlling any system. As the technology evolved ladder logic has replaced the relay logic. Ladder logic is designed in such a way that it mimics the relay logic. As explained by Hugh Jack in [11], "the earlier PLCs were programmed with a technique that was based on relay logic". This is due to the fact that it is very easy to learn, to use and to design. Simply put, if we use ladder logic as the main programming language, the number of training needed for engineers or technicians will be scaled down [11].

In the earlier days of ladder diagram, heuristic and intuitive method is used to design the logic for ladder diagram. The problem in designing is centered on in expressing the desired sequence/motion of operation in ladder logic notation. There are several methods widely used for designing the logic of ladder diagram for PLC-based controller design in automated manufacturing system namely the CASCADE method, Shift Register method, Karnaugh-Veitch map (the program writer must have the knowledge in Boolean logic function) and Huffman method [1], [2], [9], [11]. Here in this project the author would like to acknowledge two new methods which are widely used in the industry mainly in Europe namely Method A and Method G. These two methods have not been documented precisely in any journal as far as the author knows. Author's supervisor acknowledges him that both Method A and Method G is a refined-version of the approached used by the engineers in the industries but not clearly documented.

1.2 Problem Statement

There is a need for a systematic method and less time consuming approach to create logic for ladder diagram in PLC-based controller design in automated manufacturing systems.

1.3 Objective

The main objective of this project is to analyze the method use to create logic for ladder diagram: namely method A and method G. The other objective is the evaluation of Automation Studio software for used in designing and verifying the ladder diagram developed.

1.4 Scope of Study

The area of study would be on how method A and method G are use to design the logic of ladder diagram for PLC-based controller in manufacturing system.

Other approach used to developed pneumatic system and electropneumatic system is also being covered in this project..

CHAPTER 2

LITERATURE REVIEW

2.1 Formal methods in PLC programming

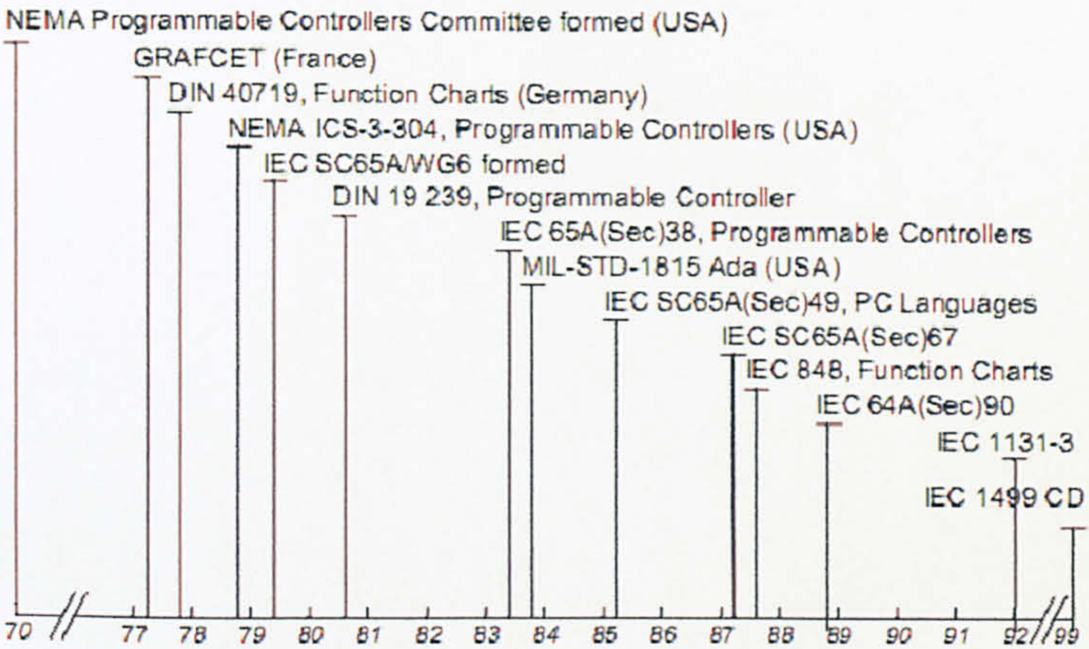


Figure 2 : Standardization in PLC Programming

Frey G. and Litz L. has discussed in their work about the needs for a formal method in PLC programming because of several reasons; "The growing complexity of the control problem, the demand for reduced development time, and the possible reuse of existing software modules and the demand for high quality solutions and especially the application of PLC in safety-critical processes need verification and validation procedures, i.e. formal methods to prove specific static and dynamic properties of the programs as for example liveness, unambiguity or response times." [8].In Figure 2 (which is taken from [21]) shows the international standards in PLC programming while in Figure 3 below is the generic model of the logic control design process as discussed in [23]. Figure 4 shows the generic model of the logic control design without the formalization [8].

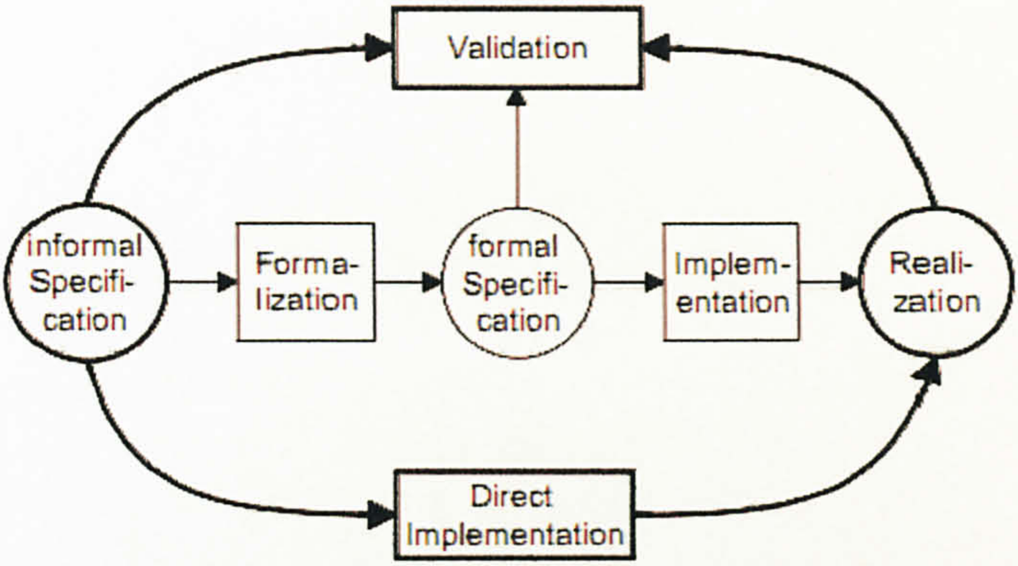


Figure 3 : Design process for logic control systems [22]

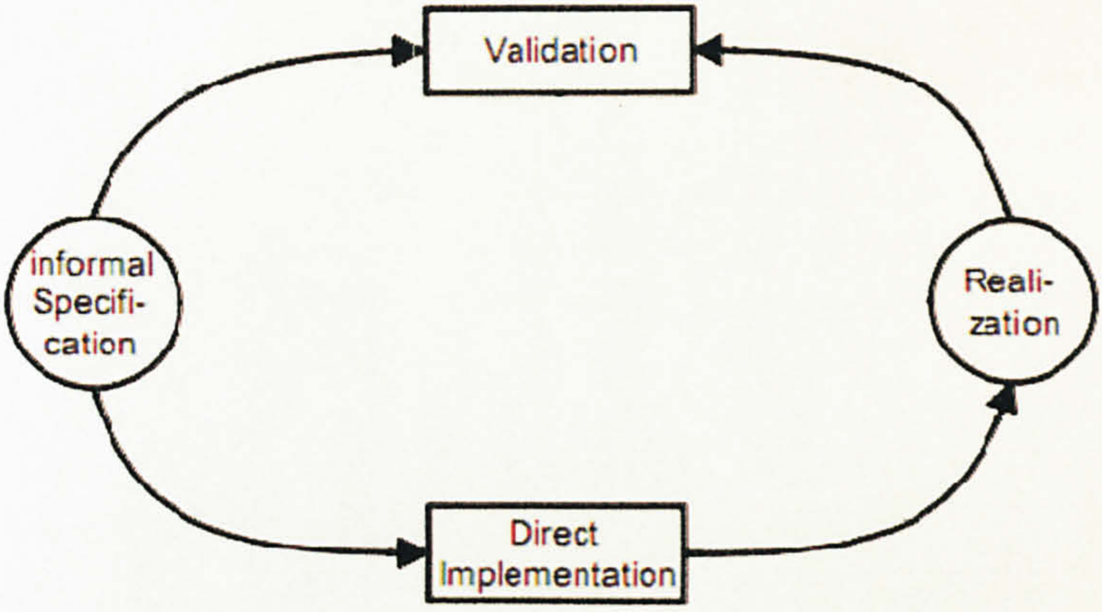


Figure 4 : Design process without the use of formal methods [8]

2.2 A systematic approach for the sequence controller design in manufacturing system

Lee J.S. and Hsu P.L in their works have proposed for a "systematic approach for the design and implementation of the sequence controller in manufacturing systems, by introducing the sensor state into the Petri nets to form a simplified Petri nets controller (SPNC) and use the token passing logic (TPL) to obtain a more compact LLD structure" [5].

2.3 An improved evaluation of ladder logic diagrams and Petri nets for the sequence controller design in manufacturing system

In this article, Lee J.S. and Hsu P.L. have proposed for a "new approach towards evaluating the ladder logic diagram and Petri nets methods via the IF-THEN transformation. This IF-THEN format can result in unified comparison between ladder logic diagram and Petri nets which is the sum of (1) the number of IF-THEN rules and (2) the number of logical operators for both LLD and PN" [4].

2.4 Petri nets based design of ladder logic diagrams

According to Chirn J.L and McFarlane D. C. in their journal, "Petri net (PN) based modelling is to be used as a systematic approach for designing a ladder logic diagram for a PLC. This approach is applied to the programming of sequential logic in order to improve design efficiency and to reduce the test and maintenance effort required for complicated systems" [13]. Figure 5 below shows the relationships among Petri net, LLD and real world.

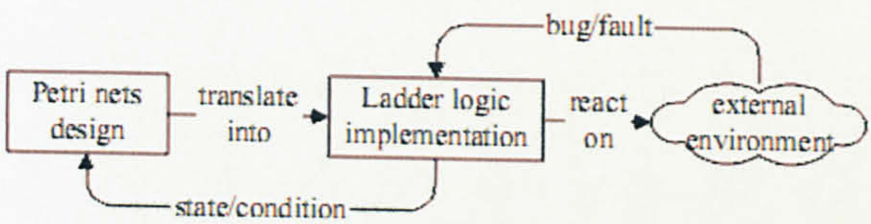


Figure 5: Relationships among PN, LLD and real world [13]

CHAPTER 3

METHODOLOGY

3.1 Procedure Identification

The project flow below show the steps that will be taken to carry out this Final Year Project. This is to ensure that the project flow is smooth and accomplish in the given period.

Figure 6: Project Flow

This project will go through to 5 major phases:

Phase 1

The literature review is done to find the other methods available on programming the PLC-based controller especially in automated manufacturing system.

Phase 2

Study on both methods, method A and method G

Phase 3

A case study simulation based on a packaging process using both methods

Phase 4

Hardware implementation of the case study

Phase 5

Evaluation of all the outcome of this project is to be done in this stage.

3.1.1 Motion Diagram/Sequence Chart

A lot of control systems employ pneumatic actuators as the actuating element and this needs a sequence of extensions and retractions of the actuators to occur. This basic feature of the pneumatic system needs a motion diagram as a tool to show the sequence in a clear manner. This motion diagram is also known as sequence diagram/chart. The example of a sequence and its motion diagram is shown below:

Sequence : A+ B+ B- A-

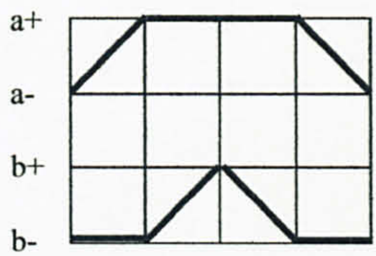


Figure 7: Motion Diagram/

Note: Notation used in the sequence is using reference letter, A and B (A for actuator A and B for actuator B). The symbol '+' means the actuator is extended, while the symbol '-' means the actuator is retracted to its original position. As an example the sequence A+B+B-A- means;

1. *Start of cycle.*
2. *Then cylinder A extends (A+).*
3. *Next cylinder B extends (B+).*
4. *Then cylinder B retracts (B-).*
5. *Cylinder A retracts (A-).*
6. *End of cycle.*

This actuator sequence can be classified into three types:

1. Event-based sequence

   When an operation has completed, it will cause the next operation to start
2. Time-based sequence

It has events that occur at pre-set time intervals and are usually controlled by either mechanical or electronic programmers.

3. Event-based sequence with time delay

   A sequence that combining both event-based sequence and time-based sequence.

3.1.2 General Rule to Create Ladder Diagram

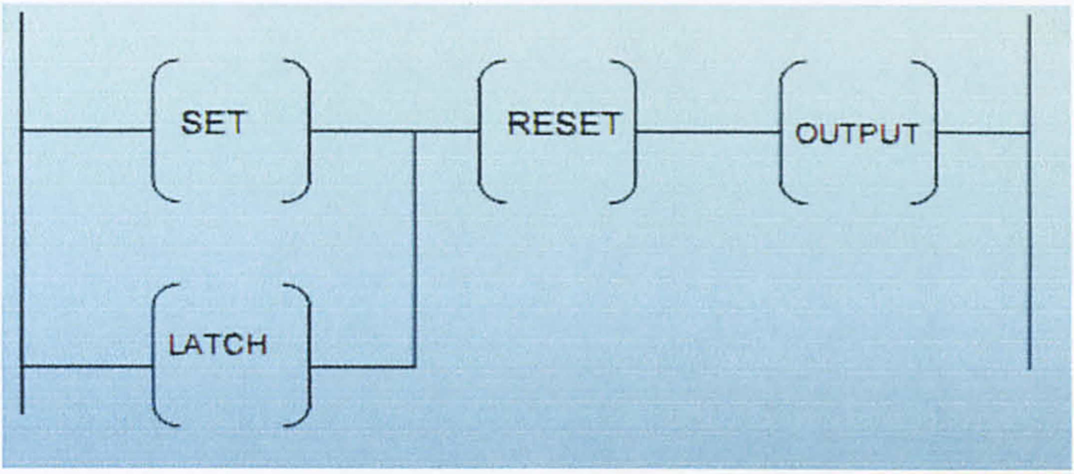


Figure 8: Basic layout of a ladder diagram

Diagram above shows the basic layout that every ladder logic programmer has to follow to ensure that any ladder diagram developed is standardized and easy to understand. This could be helpful especially for debugging purpose. The vertical line in the left represents the 24V power line while the vertical line on the right represents the 0V power line. The first rung and the second rung is where the input and outputs are situated and the number of rung is also depending on the complexity of the system to be designed. Set and Reset is usually the inputs associated with the outputs of a rung; it could be the pushbutton, limit switch or position sensor. The output is usually the coil or solenoid.

3.1.3 Case Study

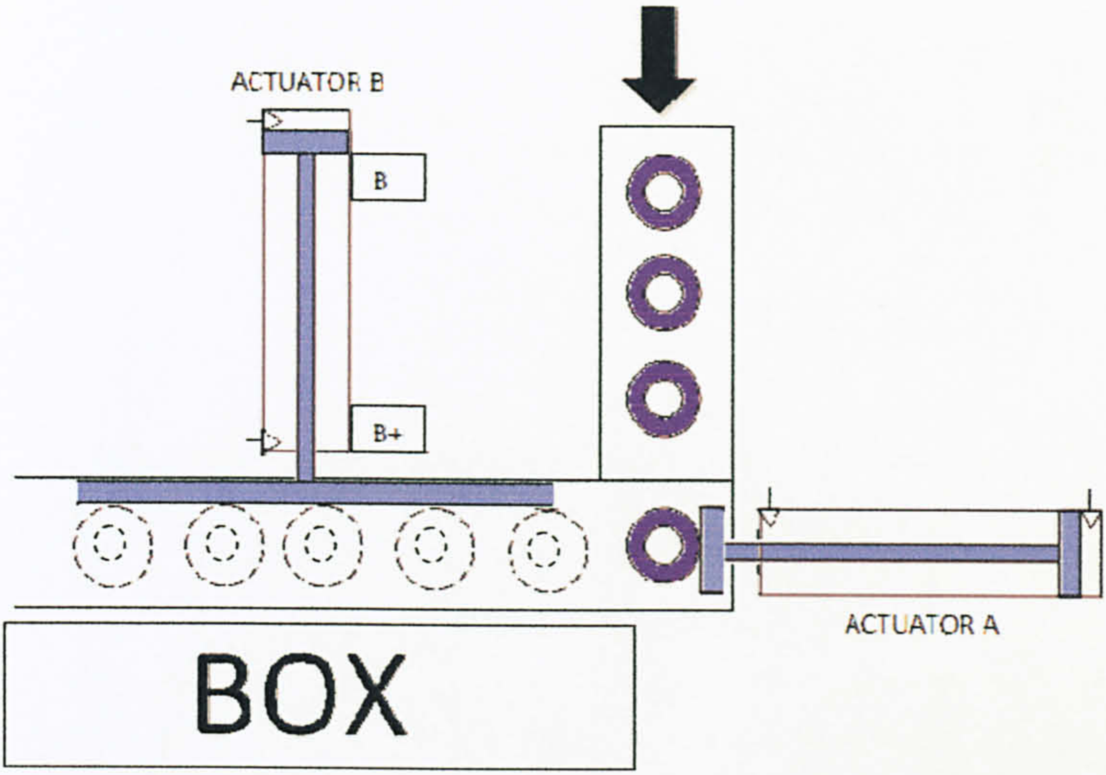


Figure 9: The layout of the system assessed in the Case Study

System description

The system shown above is used for packing tins of paint into cardboard boxes. Every operation of actuator A will insert a tin of paint in the loading magazine. Actuator B then places a row of five tins into the box. The box is full after three rows of tins is pushed into the box. The sequence is initiated by a START Push Button and the required cycle is **{[(A+A-) x5](B+B-)}x3.**

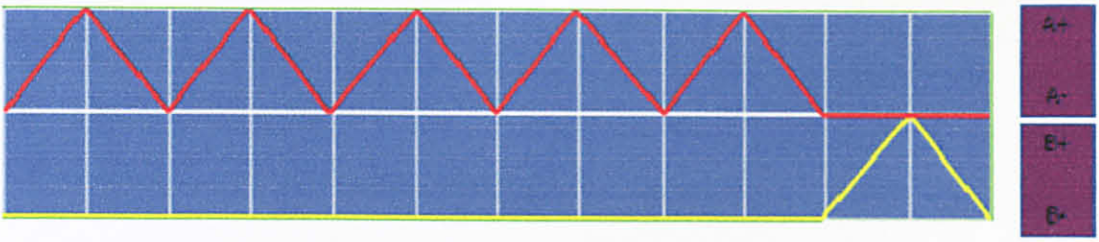


Figure 10: Required motion diagram of the case study

3.1.3 Method A

This method is almost the same concept as the CASCADE method as explained in [1].This method is more suitable if the program writer want to write a ladder diagram/logic for an electro-pneumatic circuit. For a fully-pneumatic circuit it is better to use the CASCADE method which is not covered in this project. It is time consuming if we use method A to design a fully pneumatic circuit. This method can assist the program writer to develop the logic of the ladder diagram for a particular sequence (e.g. A+A-).

General Instructions for Method A

1. List down all the cylinder action sequences (depends on the needs of the system). Example of cylinder action sequence : A+B+A-B-
2. Put all the sequences into several groups. (Rule: No letter can be repeated in a same group). Each group will be assigned with a number (i.e. 1, 2, 3… n). This group number will be known as the Secondary Variable which is to be assigned to the holding relay of the PLC.
3. If the last group has no letter in common with the first, it can be merged into the first group. The aims is to minimize the number of the groups involved (hence reducing the number of holding relay used)

4. The next stage is to create 2 (TWO) SET/RESET tables. Why two? Because one table is for the actuator to be used in the sequence (i.e. Actuator A, Actuator B). Another set of table is for the Secondary Variable.

Table 2: An Example of SET/RESET Table

Actuator	SET	RESET
A		
B		

5. For Actuator (the number of the boxes depends on the number of actuator too. We will then fill in the blanks with the appropriate SET/RESET logic equations under the SET and RESET action by referring to the motion diagram.
6. For Secondary Variable (the number of the boxes depends on the number of groups created – the smaller the better is the ladder diagram. Reduce the number of relay used)
7. The desired ladder diagram will be programmed using the steps mentioned above.

Note: For SET/RESET of Actuator;

- *SET – condition(s) for the actuator to change into a new position from its normal/original position.*
- *RESET – condition(s) for the actuator to go back into its normal/original position.*
- *This concept also applies to other output devices such as timer and counter.*

We shall look into the case study to see how method A is employed. There are two actuator, Actuator A and Actuator B, use in the case study. The sequence is **{[(A+A-) x5](B+B-)}x3**. This sequence is a bit tricky actually if we would like to compare with other sequence such as A+B+C-B-A-C+, A+B+A-B- or B-C+B+C- because it is impossible to group the sequence due to the cycle repetition each actuator has to make. So we need to break this **{[(A+A-) x5](B+B-)}x3** sequence into two simple sequence i.e. A+A- and B+B- .

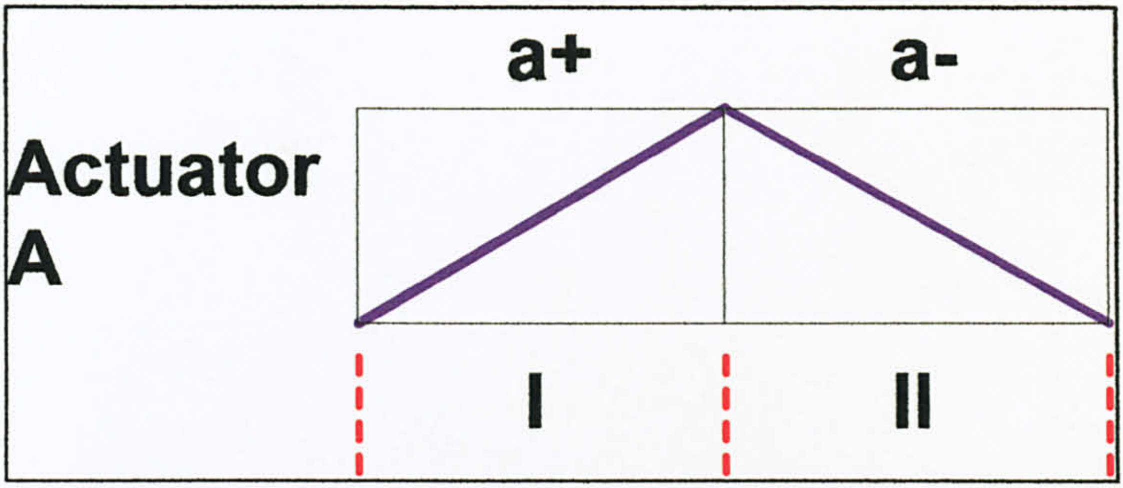


Figure 11: Motion Diagram for sequence A+A-

Table 3: Set/Reset Table for Actuator A (Secondary Variable)

SECONDARY VARIABLE	SET	RESET
I	Start. a^-	a^+
II	a^+. I	a^-

Table 4: Set/Reset Table for Actuator A (Actuator)

ACTUATOR	SET	RESET
A	I	II

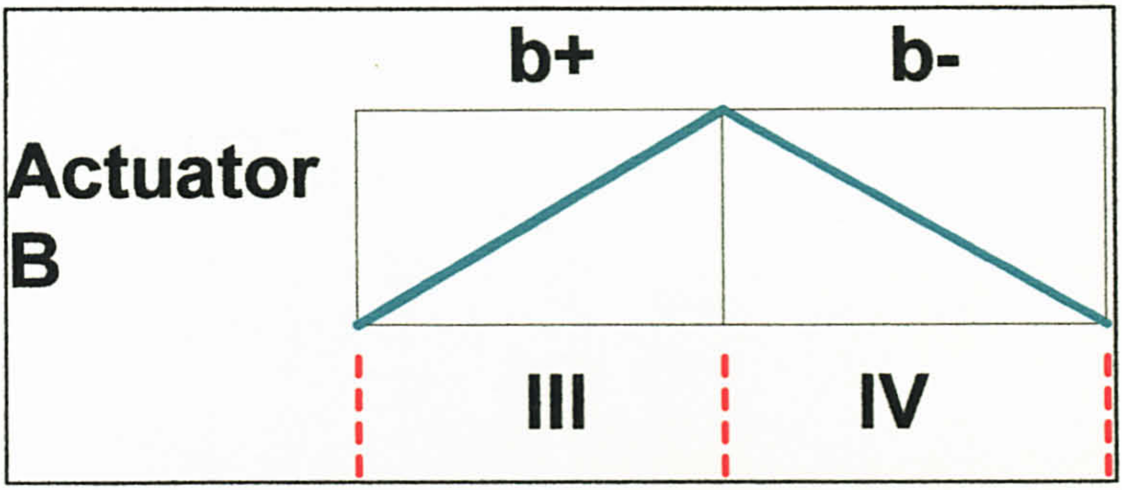


Figure 12: Motion Diagram for sequence B+B-

Table 5: Set/Reset Table for Actuator B (Secondary Variable)

SECONDARY VARIABLE	SET	RESET
III	Start, b^-	b^+
IV	b^+. I	b^-

Table 6: Set/Reset Table for Actuator B (Actuator)

ACTUATOR	SET	RESET
A	I	II

3.1.4 Method G

This method is slightly different from method A but it still employs the SET/RESET concept. It is a graphical method. Method G still can be used to develop the logic of ladder diagram for any system that is sequential in nature and also for system that is combinational in nature (i.e. batch production). Example of batch production would include; bakeries, car factory, and pharmaceutical industry.

General Instructions for Method G

1. For the first step, we need to determine the type of system :
   - System with sequential nature (i.e. a simple system such as A+B+C+A-C-B-)

     The programmer needs to know the sequence of the system that is to be programmed and the sequence chart of the system
   - System with a combinational nature (i.e. a more complex system such as a batch processing)

     The programmer needs to know the required sequence of events for the system to be designed and its sequence chart
   - Based on parameters above we then will complete all the expected sequence on the sequence chart which will include the input such as start button, sensors associated and timers.
   - The sequence chart has three parts – Parameter, Input and Outputs). Below is an example of the sequence chart defined above.

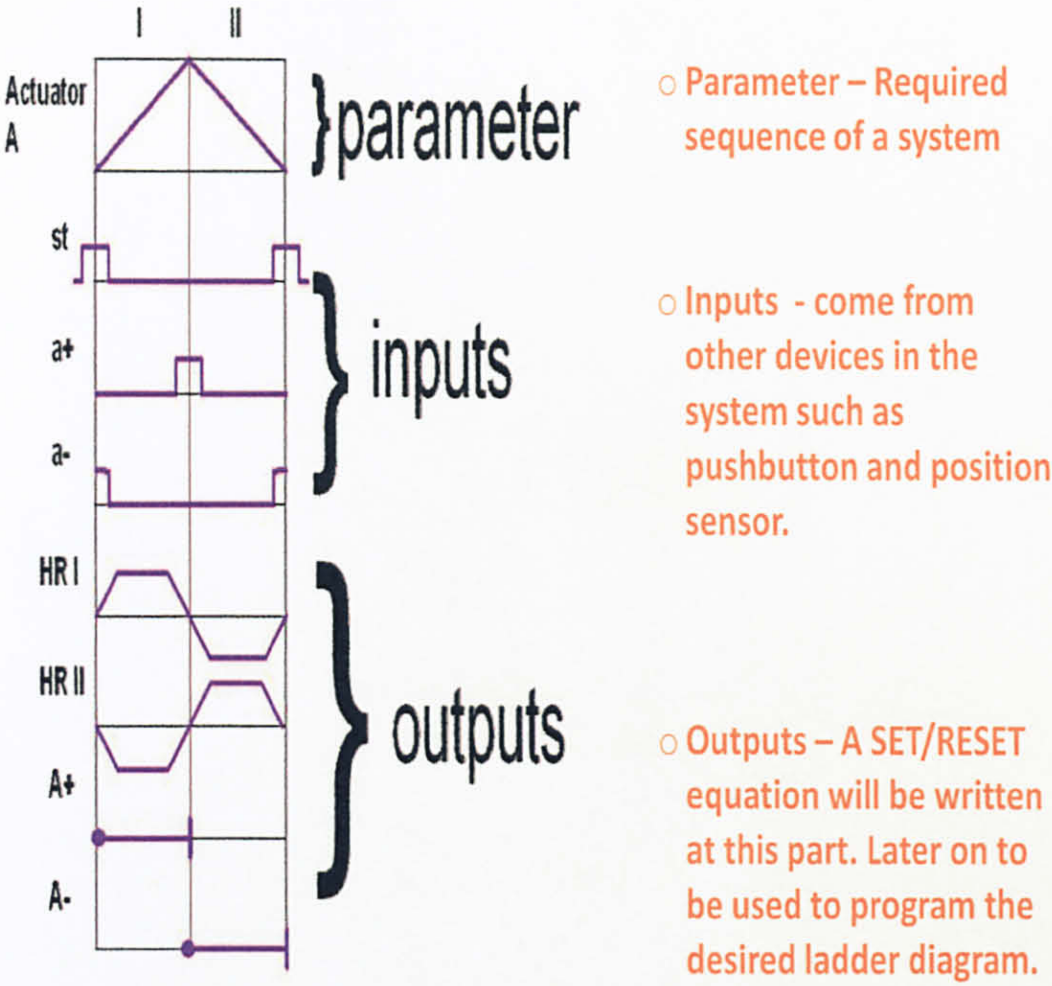


Figure 13: Example of a Sequence Chart for Method G and its explanation.

2. The output of the system will be assigned to one holding relay and we then draw its associated sequence in the same sequence chart.
3. The programmer than will write the SET/RESET logics equations of the outputs of the system.
4. The desired ladder diagram will be programmed using steps 1-3.

We shall look into the case study how method G is employed. As explained in Method A, we will also separate the sequence into two simple sequence.

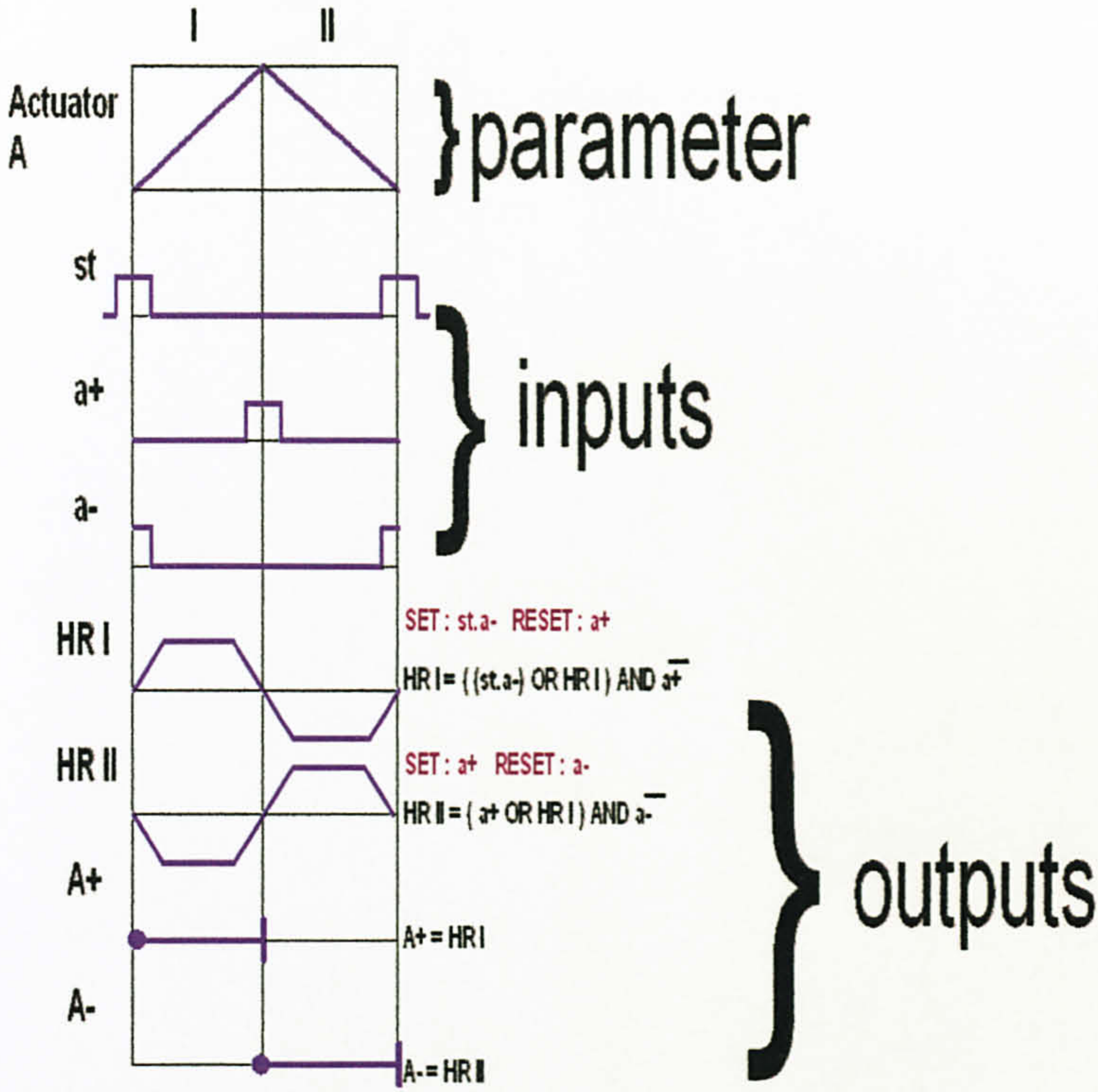


Figure 14: Sequence Chart for sequence A+A-

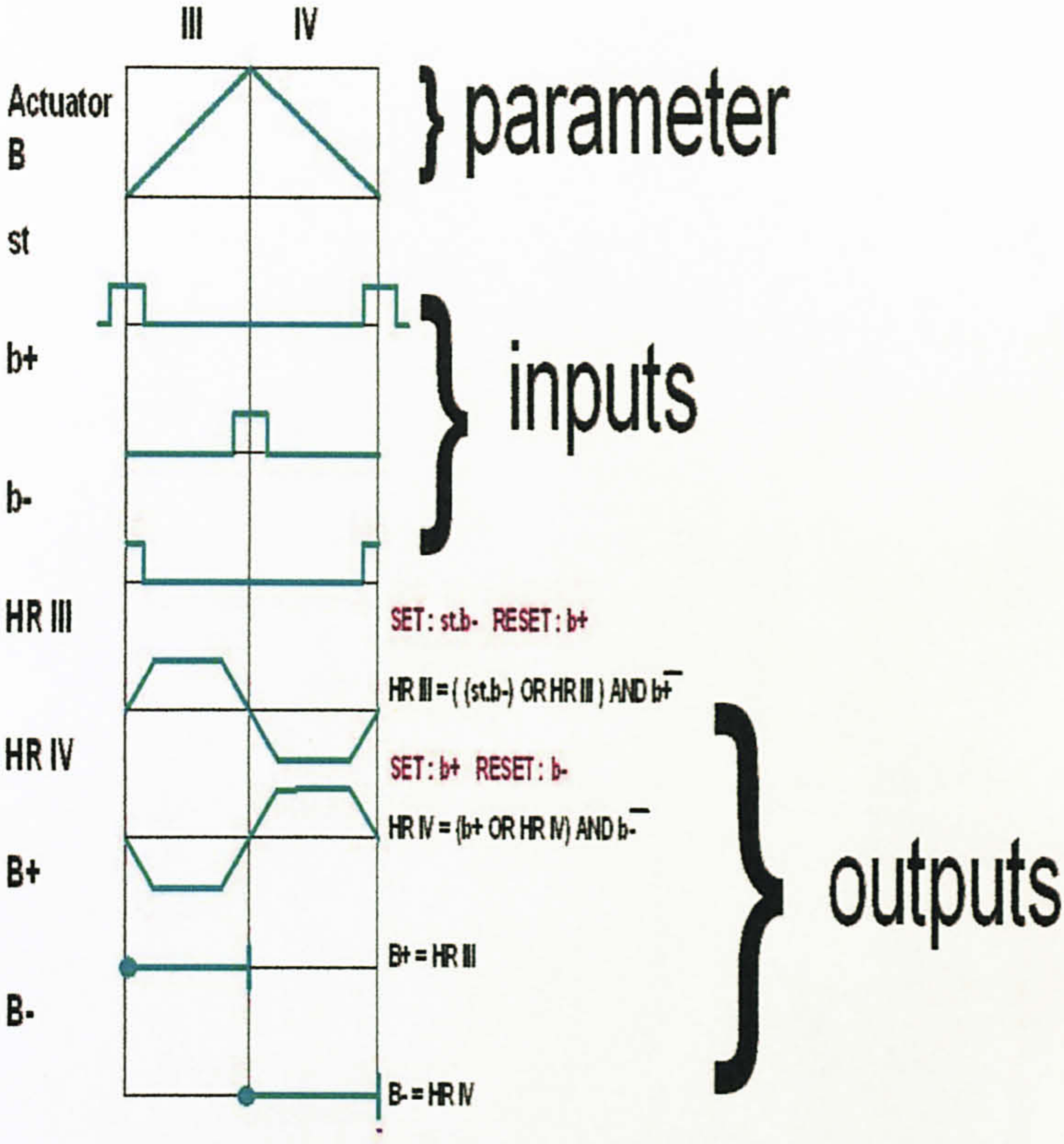


Figure 15: Sequence Chart for sequence B+B-

3.2 Tools Required

Listed below are the identified tools that will be used for the whole period of this project.

1. Automation Studios

   This is an innovative software solution for system design, simulation, engineering, prototyping, diagnostics, and training. The software allows for advanced engineering from system design through training to service. It has a comprehensive library on pneumatic elements, Electrical Control (JIC/IEC Standard) components which is a good choice for this project. It has an enhanced animation module to create realistic machine and component animations (including X-sections) for improved training and communications. This project will use the Automation Studio Version 5.0.

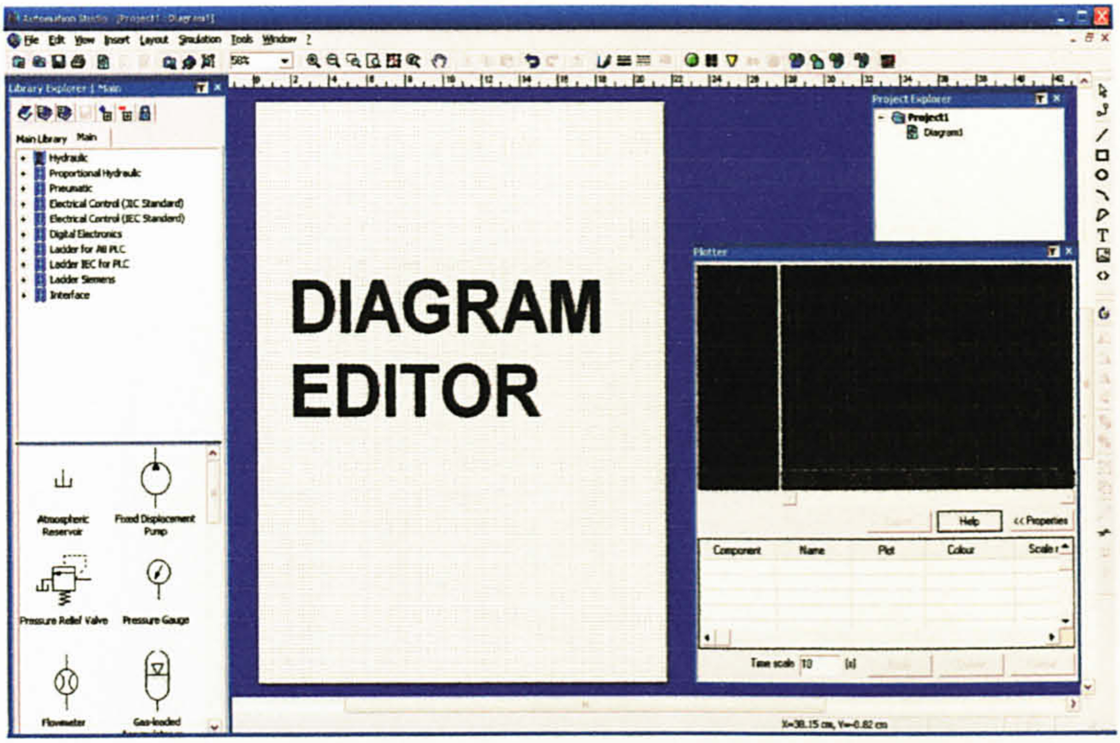


Figure 16: Screenshot of Automation Studio V5.0 software

2. OMRON Programmable Controller Training unit available in-house at Universiti Teknologi Petronas. This Kit has a unit of Omron Programmable Logic Controller, several digital input/output module, and analog module.

Figure 17: Omron Programmable Logic Controller

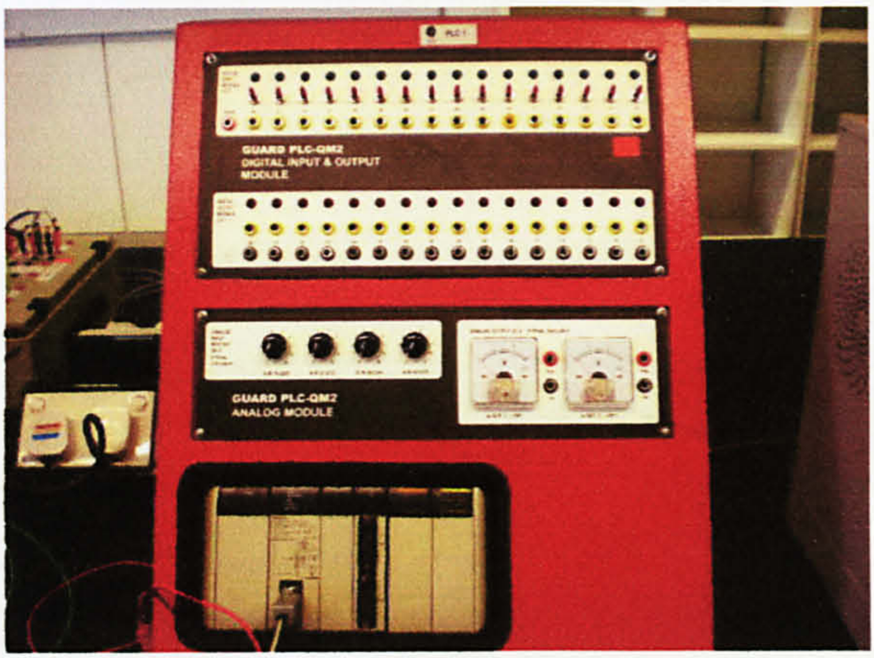


Figure 18: Omron Programmable Controller Training Kit

3. Omron CX- Programmer that provides a platform for developing programming for the Programmable Logic Controller used in this project.
4. Electro-pneumatic actuator equipment kit for hardware implementation.

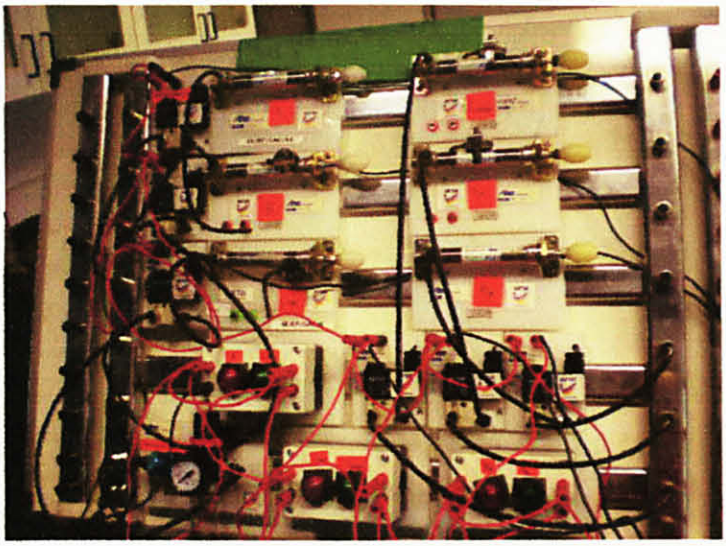


Figure 19: The Electro-pneumatic component

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Results

To demonstrate and simulate the viability of the method A and G in developing logic for ladder diagram, two simple sequences and the case study is presented.

.

I. Sequence: A+ B+ A- B-

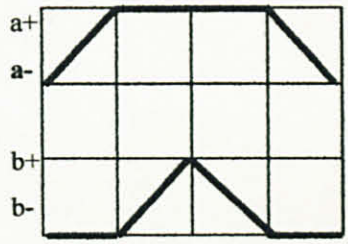


Figure 20: Motion diagram for sequence A+ B+ A- B-

* Method A

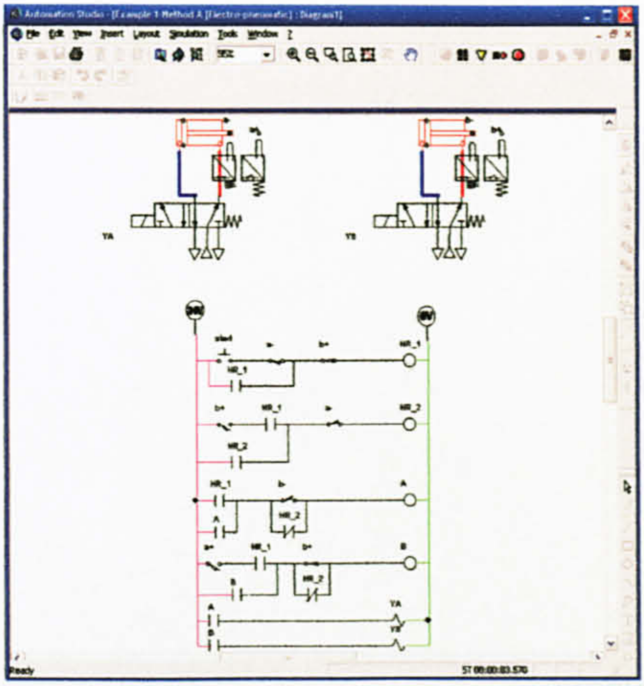


Figure 21: Ladder diagram of sequence A+ B+ A- B-

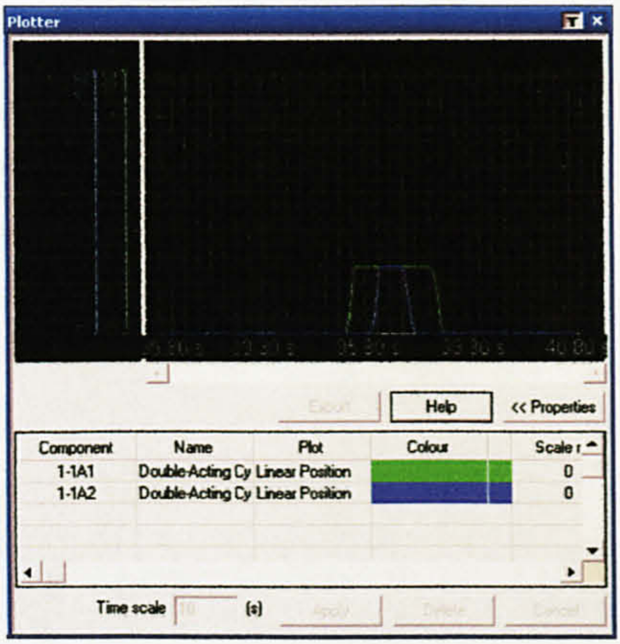


Figure 22: Linear position of the actuator (A & B) for sequence A+ B+ A- B-

- Method G

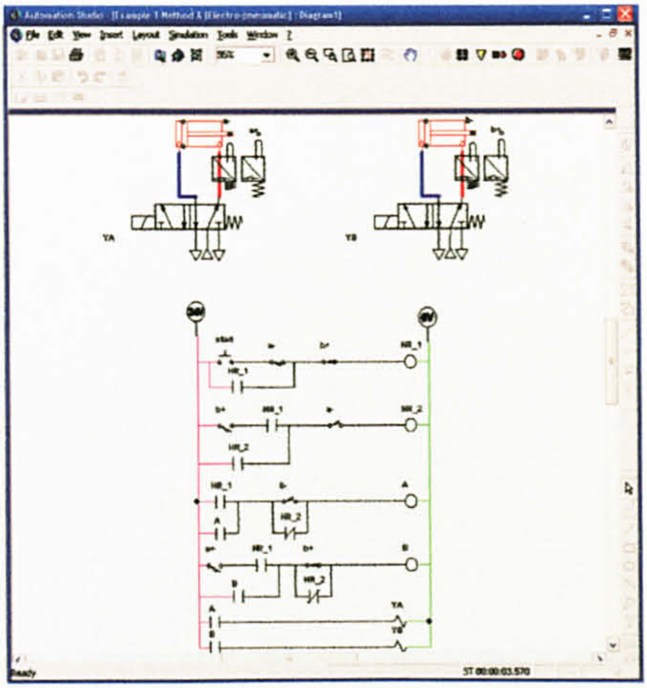


Figure 23: Ladder diagram of sequence A+ B+ A- B-

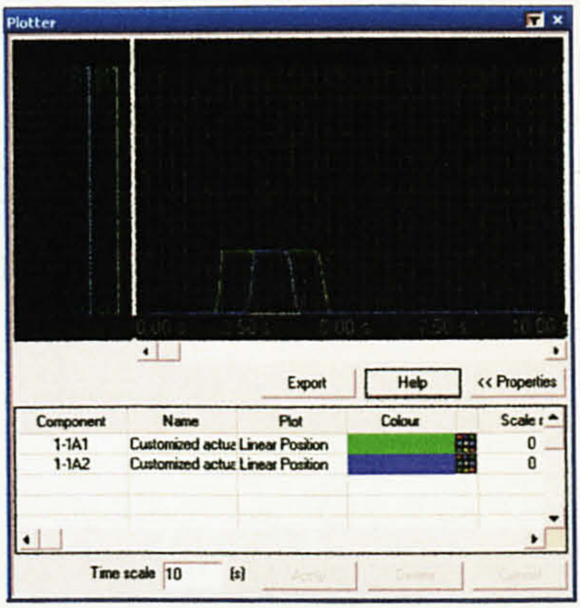


Figure 24: Linear position of the actuator (A & B) for sequence A+ B+ A- B-

II. Sequence : A+ B+B-A-C+C-

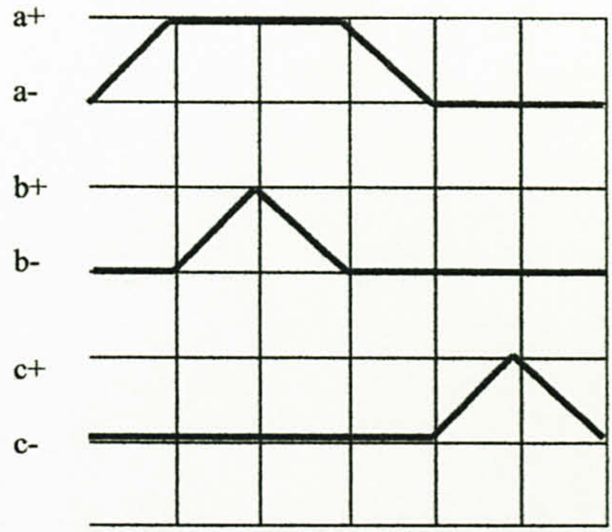


Figure 25: Motion Diagram for the sequence A+ B+B-A-C+C-

- Method A

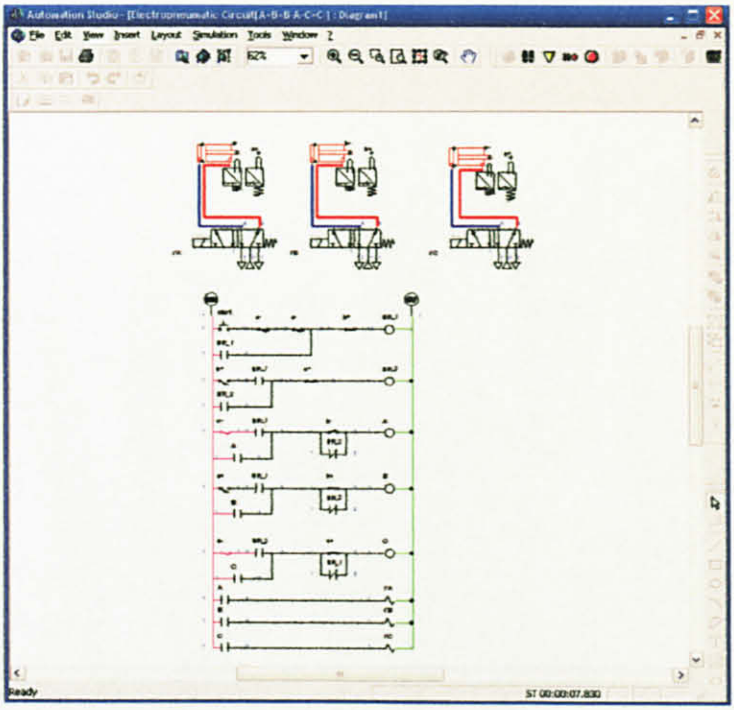


Figure 26: Ladder diagram of sequence A+ B+B-A-C+C-

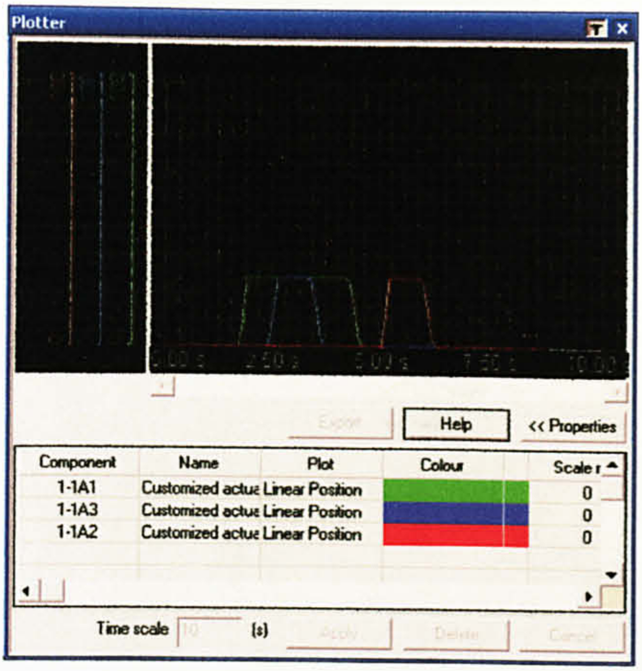


Figure 27: Linear position of the actuator (A, B & C) for sequence A+ B+B-A-C+C-

- Method G

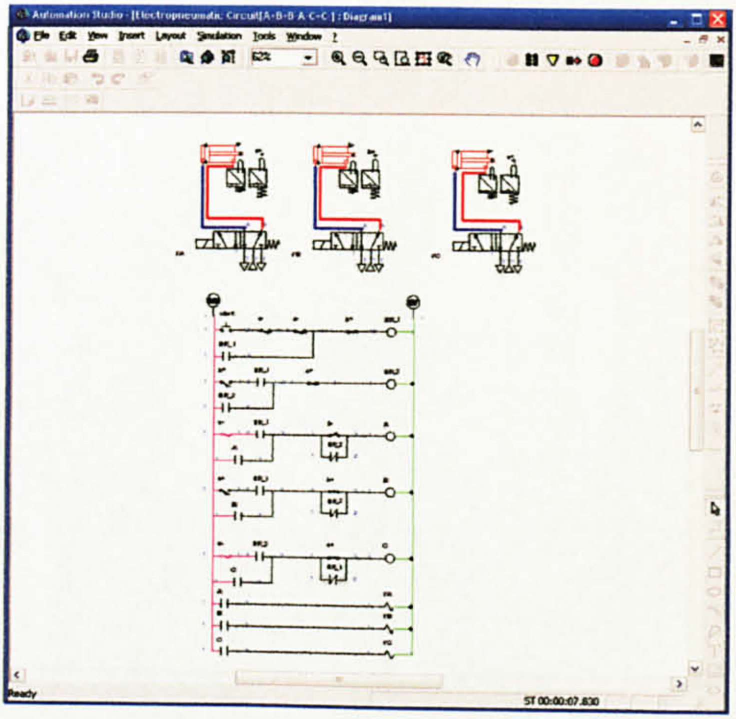


Figure 28: Ladder diagram of sequence A+B+B-A-C+C-

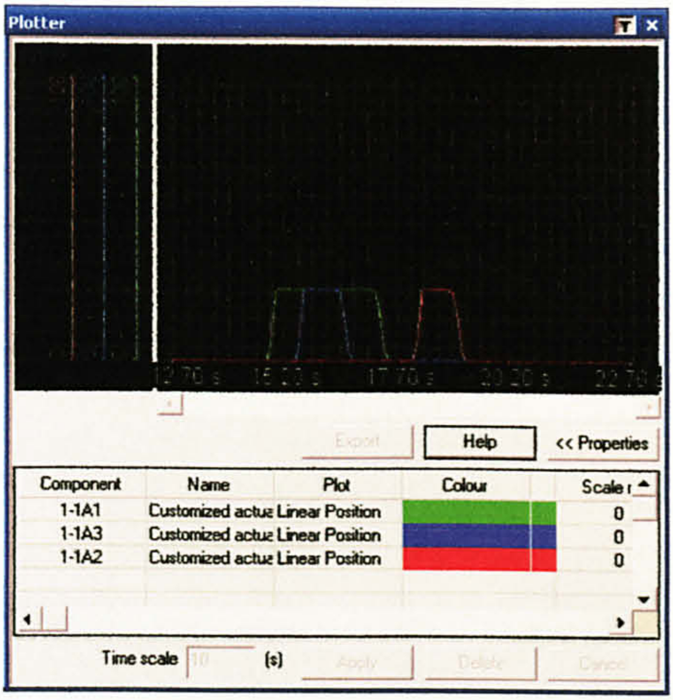


Figure 29: Linear position of the actuator (A, B & C) for sequence A+ B+B-A-C+C-

4.1.1 Implementation of the case study using Method A

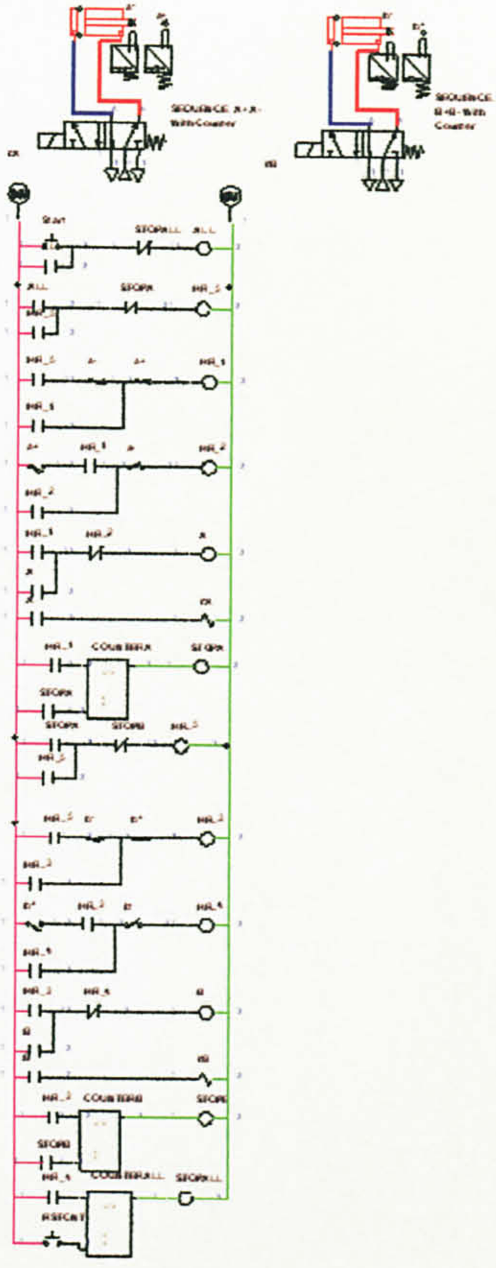


Figure 30: Ladder Diagram of the case study using Method A

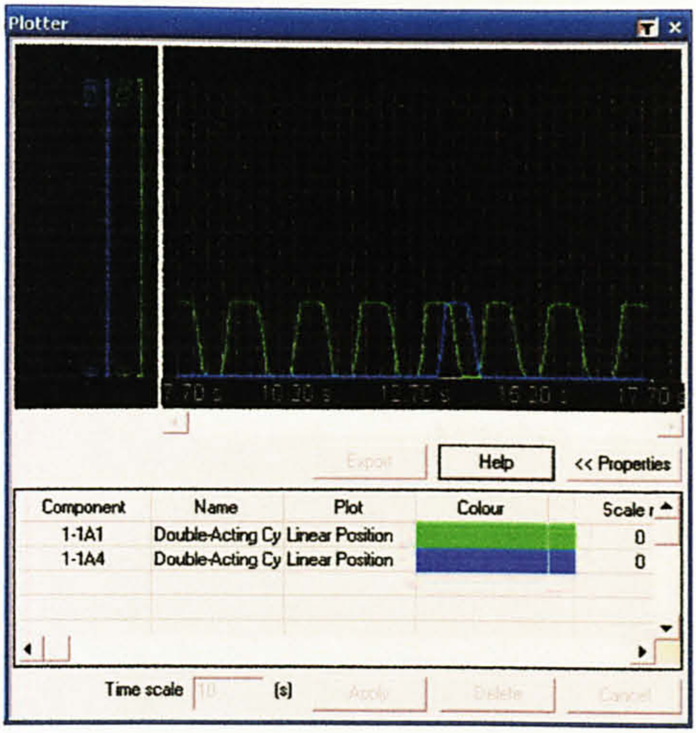


Figure 31: Linear position of the actuator of the case study using method A

4.1.2 Implementation of the case study using Method G

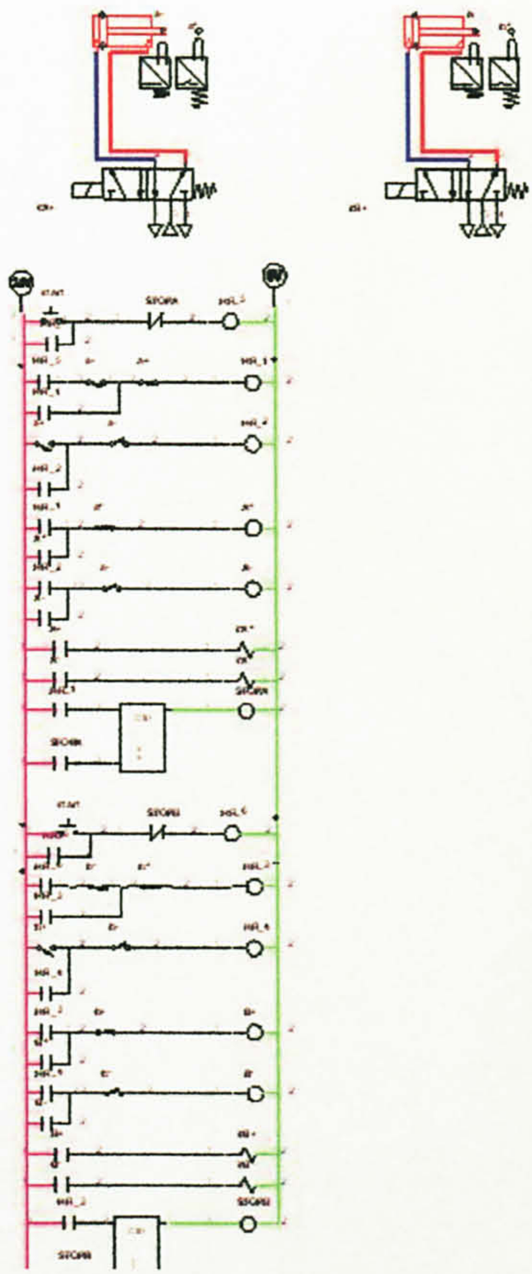


Figure 32: Ladder Diagram of the case study using Method G

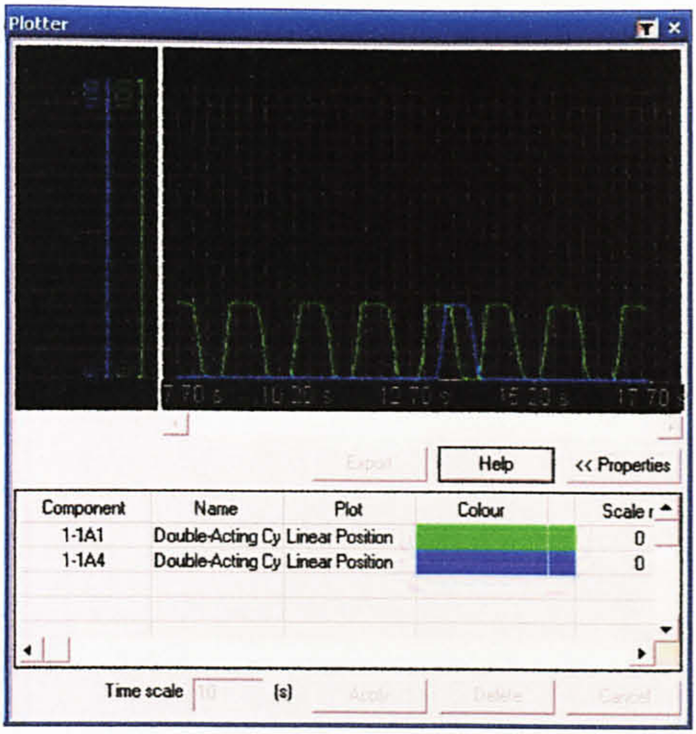


Figure 33: Linear position of the actuator of the case study using method G

4.2 Discussion

It is a good practice for anyone using either method A or method G to start designing a system from a simple sequence before moving on to a more complex system. Even though method A and method G do provide an-easy-to-learn-approach to design/program the logic of ladder diagram, a vast experience and a familiarity with system to be designed and the behaviour of its input and output are needed to produce a hardware implementation with less testing, less debugging and maintenance effort.

It has also been found there are several bug in this version of Automation Studio such as when we do a small correction or editing in the circuit (especially the fully-pneumatic circuit), the movement of the actuator will be interrupted and the output will be not same as before the correction.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

Method A and method G can be use to develop the logic of ladder diagram for PLC-based sequence controller design. It is also suggested that method A is suitable for system that is sequential in nature and method G is suitable for both sequential and combinational system. On the Automation Studio software, it is a good and user friendly software to design and verify the ladder diagram programming. It is recommended as a learning tool for new programmers. On the other side, this software is prone to crash and has a few programming bugs.

5.2 Recommendation

In the future it is recommended to compare the efficiency of method A and method G with other methods of designing the logic of ladder diagram for Programmable Logic Controllers. Examples of the available methods that can be compared include Karnaugh Map, CASCADE Method, Shift Register or Huffman method. Second it is recommended also to evaluate the other simulation software such as Festo FluidSIM and compare it with Automation Studio to determine the reliable simulation software for education, simulating and designing and verifying purpose.

REFERENCES

[1] Chris Stacey (1998), *Practical Pneumatics,* First Edition, Arnold

[2] Michael J. Pinches, Brian J. Callear (1997), *Power Pneumatics,* First Edition, Prentice Hall Europe

[3] Z.J Lansky, Lawrence F. Schrader, Jr. (1986), *Industrial Pneumatic Control,* First Edition, Parker Hannifin Corporation

[4] Lee JS, Hsu PL (2004) *An Improve evaluation of ladder logic diagrams and Petri nets for the sequence controller design in manufacturing systems, Volume 24, Numbers 3-4*, pages 279-pp. Springer-Verlag, August 2004.

[5] Lee JS, Hsu PL (2004) *A systematic approach for the sequence controller design in manufacturing systems, Volume 25,Numbers X-X,* pages 754-760-pp, Springer-Verlag

[6] Tilbury D, Khargonekar P (2001) *Challenges and Opportunities in Logic Control for Manufacturing Systems,* 105-108 pp, IEEE Control Syst Mag 21 (1)

[7] Muthukkaruppan M, Manoj K (2007) *Low Cost Automation Using Electro Pneumatic System - An Online Case Study in Multistation Part Transfer, Drilling and Tapping Machine,* In: Proceedings of 24th International Symposium on Automation & Robotics in Construction (ISARC 2007) pp 551-556 , 2007

[8] Frey G, Litz L (2000) *Formal Methods in PLC Programming,* In : Proceedings of the IEEE Conference on Systems Man and Cybernetics SMC 2000, Nashville , 2000

[9] Jaafar H (2004) *Advance in Logic Controllers Design,* In: First International Symposium on Control Communication and Signal Processing, pp 21-24

[10] Mikulczynski T., Samsonowicz Z., Wieclawek R. (1998) *Using the net transformation method to program programmable logic controllers* ; CONTROL ENGINEERING PRACTICE; VOL 6; NUMBER 8; pp. 989-996; 1998

[11] Hugh Jack (2007) *Automating Manufacturing Systems with PLCs,* Version 4.9 January 2007, GNU Free Documentation License.

[12] Abdelhameed M. M., Tolbah F. A. (2002) *A recurrent neural network-based sequential controller for manufacturing automated systems,* International Journal of Mechatronics (12); pp. 617-633

[13] Chirn J-L, McFarlane D.C. (2000) *Petri Net Based Design of Ladder Logic Diagram,* Submitted to: Control 2000 Cambridge. UK.

[14] Wright J.R. (1999) *The Debate Over Which PLC Programming Language is the State-of-the-Art,* In: Journal of INDUSTRIAL TECHNOLOGY, Volume 15, Number 4 – August 1999 to October 1999.

[15] Lauzon S.C., Mills J.K, Benhabib B (1997) *An implementation methodology for the supervisory control of flexible manufacturing workcells,* Journal of Manufacturing Systems 16(2), pp 91-101

[16] Hee R. B. (1995) *Knowing the Basics of PLCs –Part 1,* Electrical Construction and Maintenance 94(10), pp 20-28

[17] Wright J. R. (1998) *Natural language and mixed modality task presentations in the human-computer interaction using programmable logic controllers,* Ph.D. Thesis, Iowa State University.

[18] Baresi L., Carmeli S., Monti A., Pezzè M. (1998) *PLC Programming Languages: A Formal Approach,* Automation 98, ANIPLA, November 1998.

[19] IEC. Part 3: Programming Languages, IEC 1131-3. Technical Report, International Electrotechnical Commission – Geneva (1993)

[20] Aiken A., Fähndrich M., Su Z. (1998), *Detecting Races in Relay Ladder Logic Programs,* EECS Department, University of California, Berkeley ETATS-UNIS

[21] Christensen J. H. (1998) *PLCopen Standard Presentation V1.0,* Figure : International Language Standardization

[22] Baresi L., Carmeli S., Monti A., Pezzè M. (2000) *PLCTOOLS: Design, formal validation and code generation for programmable controllers* Proc. Of the IEEE SMC. 2000

[23] Frey G, Litz L (1998) *Verification and Validation of Control Algorithms by Coupling of Interpreted Petri Nets,* In : Proceedings of the IEEE Conference on Systems Man and Cybernetics SMC 1998, San Diego, Vol. 1, pp. 7-12

[24] Introduction into IEC 61131-8 Guidelines for the application and implementation of programming languages. http://www.plcopen.org/pages/tc1_standards/iec_6131_8/