CIM CONTROL DESIGN USING GRAFCET

SITI HUZAIMAH BINTI HUSIN

A thesis submitted in partial fulfillment of the requirements for the award of the Degree of Master of Engineering (Electrical)

Kolej Universiti Teknologi Tun Hussein Onn

MAC 2005

For you, my mom and dad

For your truly support and undivided love

For making me the person

Who I am today.....

ACKNOWLEDGEMENT

I would like to take this great opportunity to extend my deepest gratitude to all these who have assisted me in making this project a success. A special note of thanks goes to my respected supervisor, Assoc. Prof. Dr Zainal Alam Haron for his patience and guidance throughout the course of this project.

I would also like to take this opportunity to thanks all technicians who have played a role in accomplish this project.

Not forgetting also to my lovely family and all my dearest friends who have never failed to lend a helping hand when I needed them. Thank you.

ABSTRACT

This project is concerned with learning and exploring industrial automation technologies, such as control systems, Computer Integrated Manufacturing system and the relationship between these major components. The CIM70A System will be used in this project and located at Automation Lab, Kuithho. The project is divided into two major parts. In the first part, the author familiarized herself with the operation and all the functions of the related industrial components of the CIM70A System, such as the sensors, actuators, and the valves. In the second part, the author aimed to study GRAFCET and attempts to relate between the graphical representation and mathematical model using PLC programs. With this formal approach, the programming will be written more neatly, well arranged and standardized. This project successfully explores the possibility of designing the control program using GRAFCET.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER I

INTRODUCTION

1.1 Background

In early 1970's, the programmable logic controller (PLC) were becoming more powerful and more extensively used, the need to describe increasingly logic controller becoming evident. Thus, an efficient approach which can represent the control process by using the graphical representation perhaps a good solution to helps the designer in order to programming the PLCs. There has been a growing interest in programming languages for PLCs. In particular, the sequential function chart (SFC), an international standard based on the GRAFCET language was introduced in France in 1977.

The GRAFCET language has been used as one of the most important means for designing, programming and describing logic sequential control systems. This powerful graphical language dedicated to the specification of the behavior of sequential logical systems. It is standardized by CEI and its semantic is defined for this type of applications. The [1] and [2], concluded that the GRAFCET is a very good tool for logic controller specification, and the graphical nature of the language makes GRAFCET easy to learn and use.

With a tremendous number of inputs and outputs, the model of the control system or application has typically over fifty steps and transitions and therefore, it is difficult for a human operator to understand it; the GRAFCET formalism loses one of its major qualities namely to provide a graphical representation model of the sequential model [3].

GRAFCET also has contributions in design recovery for relay ladder logic. The objective in design recovery is to analyze existing source code and construct from it a structured representation of the program logic. In [4], it has been proved that an existing RLL programs can be translate into GRAFCET that clearly represents the sequential control logic relative to the specified process control application.

For better understanding, a definition and theory of the GRAFCET, PLC, and CIM system is presented. Methods for using GRAFCET as a design tool are shown through the use of examples.

1.2 Problem Statement

Dealing with the computer integrated manufacturing system, we will involve with a great number of programmable logic controller as its workhorse. Many existing programs for the PLCs are written in relay ladder logic (RLL), which in its most primitive form is a graphical representation of Boolean switching functions based on an analogy to physical relay systems. Although RLL is widely used and understood by industry technical person, the RLL is often difficult to debug and modify because its graphical representation of switching logic makes ordinary person difficult to understand the sequential in the program design.

Furthermore, there is no formality in order to design the control system. There is one approach that commonly used in order to develop the control program using PLC. Namely, try and error approach or heuristics approach, which will be differed from one designer to other. Besides, this unstructured approach will not guarantee the safety, the working, the readability and understanding of the operation of the system.

How to represent the logic controller systems in a formal way?

In order to solve the stated problem, this project proposed a formal approach namely GRAFCET as a design tool in order to build up more structured control program.

1.3 Objectives of Project

This project is consisted of two parts. In the Part I, the objectives of this work is to familiarize with the system itself, the function and the controlling elements. Since the PLC as the controller, the CX Programmer as the programming tool will also be learned in this part.

For Part II, the objective is to write the programming using the GRAFCET as a graphical tool in order to develop the working control program that in lined with the existing system. In further, the objective is to provide a set of laboratory manual how to start and use GRACET as a design tool as a references materials for the students or lecturers for future work. As mentioned above, the main objective of this paper is, to design the CIM70A system control program using the GRAFCET that clearly represents the sequential control logic relative to the specified process control application. Also to show that GRAFCET can be used as a design tool that can helps the designers to come out with the structured, safety, working and readability control program.

In this project, we mainly concentrate on the modular of Pick and Place station of CIM70A system as the test bed. We tried to design and develop the control program of this modular and intent to build the communication path between the Master of the PLC (which controlling the conveyor line) and the Slave PLC that controlling the Pick and Place station as the final target.

CHAPTER II

LITERATURE REVIEW

2.1 Introduction

The cost of developing software for the logic controller in the manufacturing world become increasing by a day. Furthermore, in designing the representation of the logic controller such as PLC, many companies using a different languages. In the desire to improve the capabilities of this logic controller, these languages become more complex. This is in contradiction with the main goals of the PLCs, which is that the PLC is a very simple system because it is very important that PLC must be reliable. There obviously needs to be a consistent philosophy for developing reusable code for manufacturing controllers systems.

In today's economic context, the design of these control applications is of a great impact in terms of productivity and production costs. Because of these costs, of the complexity of the control systems and the multiple hardware/software combinations, the designer has to take the safety of these systems into account. In this context, it is necessary to provide the designer with verification methods that ensure the safety and liveness of the control system. In deep, the verification methods will help the designer to troubleshoot the programs while the error occurs in the production.

One way is to ensure the safety of the PLC programs is by using "framework" or standard while programming process began.

As concluded in [1], the usage of Grafcet as a very good tool for logic controller specification has been approved. It allows modeling of concurrency and synchronization. Above all, the input-output behavior is specified without doubt. When some parts of the logic controller can be described separately, one can use macrosteps to simplify the model. Also, the comparison was made between State Table, Petri Net, RLL and Grafcet, and it make an evident that the Grafcet applications was the easiest among all.

The strength of the Grafcet also had shown in [2]. The graphical nature of the language makes Grafcet easy to learn and use. The ability to test different ideas quickly has been very useful in determining the final design. The Grafcet helps the designer determine: modularization of the code, functions that can be performed in parallel, communication between parallel processes, and problems in control flow. The further extension of the Grafcet usage has been proved in [3], which touched on how this tool can be applied to the reduction of a model in a specific context such as for the model that has typically over fifty steps and transitions. The Grafcet also has been successful in [4] in order to converting existing RLL programs into the form that considerably easier to understand and it also helps to modify the programs. Some of Grafcet contributions were touched in [5, 6, 7]. Furthermore, the Grafcet can be implemented to avoid the damages or system failures during the plant operation, due to interactions between human operators and plant. This statement was fully supported in [8, 9], that also shown that the Grafcet language has a particular characteristics that support supervision of external actions over a process. The results demonstrated in [8, 9] also shown that Grafctet implementation over a

PLC can avoid human errors and can indicate on a set of outputs, the parts and variables of the plant that do not satisfy the interaction demand conditions. In order to make the Grafcet success, [10] introduced two different techniques to make proofs on the properties of this language.

2.2 Historical Development

In 1975, the working group called "Logical Systems" from AFCET (Association Française de Cybernétique Economique et Technique) create the standardisation of a requirement representation for a logical automated system. This group trying to define a simple formalism, accepted by everyone and well-adapted for the representation of the sequential evolutions of a system understandable by designers as well as by users and providing potentially easiness for the implementation with hardware and/or software solutions. In December 1977, the Grafcet which means Functional Graph of a Step-Transition Command was derived as a tool of state graphs. After Grafcet had been introduced in higher and technical education, it was supported by the arrival of the first programming languages allowing implementation of Grafcet specification models on industrial logic controllers; it became an AFNOR standard in 1982 known under the reference NF C03190.

Aware of the necessity to precise how to implement a Grafcet specification with hardware and/or software, a synthesis document on recommended interpretations for the transformation from the Grafcet specification model to a specific realization. This thought led the group to propose in 1987 in an AFCET synthesis document, a certain number of Grafcet extensions to meet users' expectations. So as to strengthen the diffusion of Grafcet, and thanks to the efficient action of Paul BRARD (La Télémécanique Electrique), Grafcet became an international standard in 1988 under the name : "Sequential Function Chart" (SFC).

In 1989-1996, the study was carried out of the extensions and moved towards a strengthening of the Grafcet theoretical basis. As a matter of fact, the growing demand of Grafcet users regarding a good understanding of the complexity and the dynamic behaviour of the models, regarding the proof of characteristics for the development of safe operational systems, or regarding the prospective use of the Grafcet model for further modelling tools, didn't match the Grafcet definitions that were only textual.

In 1997 the group organized a presentation stand at EDUCATEC'97 from December 2nd to December 5th, 1997 (CNIT-Paris La Défense) to mark the twentieth birthday of Grafcet. This event gave a complete review of the actions led in Education, Industry, Standardization, and Research and to show the progress made during the last twenty years. In June 1998 an invited session at IFAC INCOM'98 drew attention to foreign laboratories which carry on with researches on Grafcet. The theoretical works on the Grafcet model are increased by taking into account the characteristics of Grafcet in relation to the characteristics of the systems which Grafcet describes the behaviour.

In this sense, the current orientations aim at better defining the properties of the frontier separating Grafcet from its environment (interactions by inputs-outputs, typology of these interactions exploiting the definition of both temporal scales, receptivities, actions, events, the taking into account of the semantics of the modelized system as a guide for the elaboration of a Grafcet. In the future, the Grafcet group will certainly imagine other actions of the same kind in order to enable Grafcet to remain a model with a rich past and a promising future.

CHAPTER III

METHODOLOGY

Referring to Figure 3.1, there are few stages that will be involved in order to work out the aimed objectives in this project. The figure was intended to feature the entire project process. Initially, the first stage was highlighting the preparation and familiarization level. The familiarization the CIM70A system was carried out by identified the controlling components such as sensors, valves and its working principle. Since the PLC is the workhorse of the system, the interrelated between the Input Devices and Output Devices was also get into account.

The second stage is regarding the study and familiarization of the GRAFCET as the design tool to represent the logic controller. Since the available references using higher technical standard jargon, the author tried to produce a guideline that can helps the lecturers and the students in future usage, aimed to represent the GRAFCET in a simple approach. Some experimental workings were developed in order to get more understanding how the GRAFCET itself.

The importance part in this project is to develop the programming for controlling the Assembly Station (Pick and Place) in the CIM70A system using the GRAFCET as a tool design. The CX-Programmer, the programming tool for OMRON PLC is applied to create the programs for use with the controller of CIM70A, OMRON SYSMAC CJ1M PLC. The CX-Programmer provides facilities for the support of PLC devices and address information and for communications with OMRON PLCs and their supported network types.

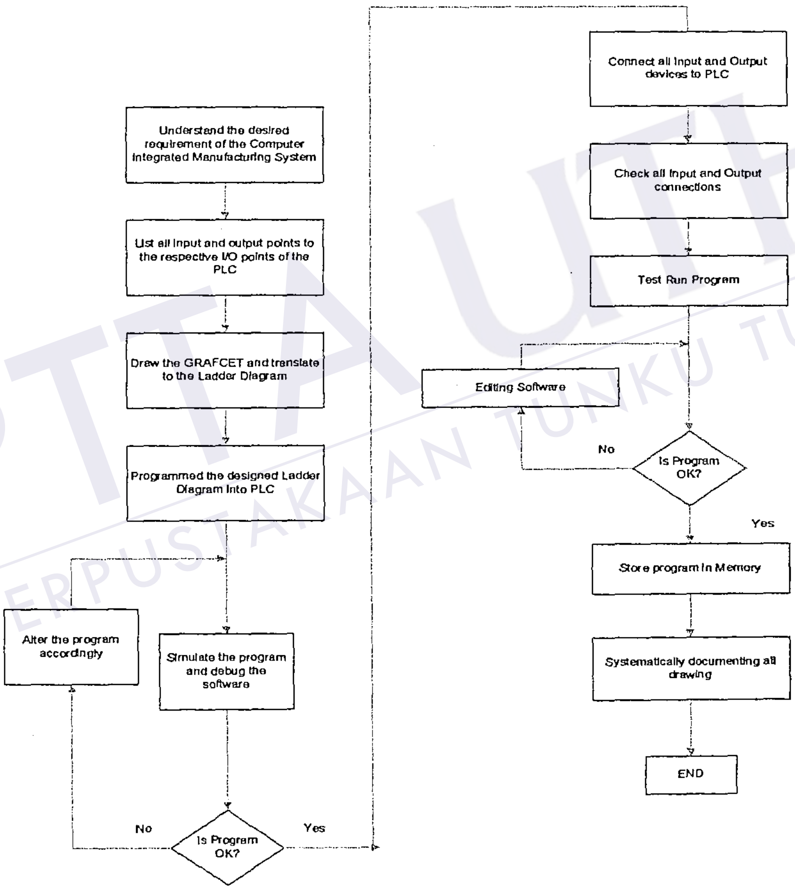

Figure 3.1: The project process flow

As we can see from Figure 3.1, the first two steps, that; understand the desired requirement of CIM system and identifications the I/O address assignment of the PLC is the important part before proceeding to draw the sequence in the graphical manner. From the understanding of the required sequence of the system, we can plan how to control the movement of each components of the related system.

The sequence of the system and the movement of each component will later become the elements of the GRAFCET structure that we intend to design and develop. The state of the GRAFCET will be representing by the sequence and the step of the GRAFCET will be characterized by the movement of the component.

The I/O address assignment is being used when we start to draw the ladder diagram for programming the PLC as usually done.

At this point, there will be a good start to emphasis that the GRAFCET are not created to replace the relay ladder diagram but to provide a step of formality and standardization of the programming language for the PLCs.

CHAPTER IV

PROGRAMMABLE LOGIC CONTROLLER (PLC)

4.1 Introduction

The National Electrical Manufacturing Association (NEMA) defines a programmable controller as follows:

A programmable controller is a digital electronic apparatus with a programmable memory for storing instructions to implement specific functions, such as logic, sequencing, timing, counting and arithmetic to control machines and processes.

Before the first introduction of the first programmable logic controller (PLC) in 1969 by Richard (Dick) E. Morley, who was the founder of the Modicon Corporation [11], logic control was implemented by hard-wired logic. Electromagnetic relays, pneumatic relays, and electronic gates and flip-flops were used. With these technologies, complex problems are difficult to handle, and a logic controller is difficult to modify.

REFERENCES

1. D. Rene, "Grafcet: A Powerful Tool for Specification of Logic Controller", IEEE Trans. On Control System Tech, Vol. 3, Sept (1995), 253-268

2. T. H. Bruce & M. Charles, "Using Grafcet to Design Generic Controllers", National Bureau of Standards, Gaithersburg, (1997), 110-119

3. S. Thierry, "Concept of descent of a Grafcet model Application to the reduction of the model in a specific context", ISMCM, France, (1995), 36-40

4. F. Albert., H.K Bruce, "Design Recovery for Relay Ladder Logic", Carnegie Mellon University, Pittsburgh, PA, (1992), 648-653

5. P. Hervé., L. Pascal., P. Jean-Francois., B. Evelyne, "Contibution of the Grafcet model to Synchrony in Discrete Events Systems Modelling", Nancy Research Center for Automatic Control, University of Nancy, France, (1994), 1527-1532

6. T. Kouthon., M.A. Peraldi., J.D. Decotignie, "Distributing PLC Control", Swiss Federal Institute of Technology, Lausanne, Switzerland, (1995), 1614-1619

7. L. Guillemaud., H. Guéguen., "Extending Grafcet for the specification of control hybrid systems", Cesson-Sévigné cedex, France, (1999), 171-175

8. G. F. Ramon, "Fuzzy Rule-Based Human Fault Avoidance Method Under Grafcet", Universidad de La Coruna, France, (2000), 381-386

9. H. Ralf., L. Ben., B. Nanette., "A Model-Checking Approach to a Safe SFCs", Multiconference on Computational Eng. In Systems Applications, Lille, France, (2003), 1-7

10. P. Le Parc., B. Queguineur., L. Marce., "Two Proof Methods for The Grafcet Language", Universiti de Brelagne Occidenlale, France, (2000), 257-261

11. S. Brian Morris, "Programmable Logic Controllers ", Prentice Hall, Ontario, 2000

12. Gary Dunning, "Introduction to Programmable Logic Controllers", 2nd Edition, Delmar, 2003

13. "A Beginners Guide To PLC" Omron Publisher, 2002

14. J. B. Waldner, "CIM Principles of Computer Integrated Manufacturing", John Wiley & Sons, Inc, 1994

15. S. Brian Morris, "Automated Manufacturing System – Actuators, Controls, Sensors, and Robotics", Mc-Graw Hill Inter. Editions, 1998

16. A. R. James, "Introduction To Robotics in CIM System", 4th Edition, Prentice Hall, 2002

17. R. Verma, "Visual Basic for SCADA and PLCs", Industrial Computing, Pg 35-38, Oct 1995

18. "Visual Programming CPT Upgrades Curriculum", Purdue Technology Today Newsletter, No 14, 1996

19. CIM70A Manual (2004)