Process Description for PLC Program Design

A.Azmi[1,*], M.A.A.Shukor[2]
[1, 2] Faculty of Electrical and Electronics Engineering, Universiti Tun Hussein Onn Malaysia

*Corresponding email: anita@uthm.edu.my

Abstract

Process description is essential and vital for PLC Program development. Various types of process descriptions method had been introduced but Flowchart, Narratives, Grafcet and Heuristic are amongst the favourites due to the direct translation and smooth flow of process sequence. In the study undertaken it is found that the process description can be further improved by combining the Heuristics method with the Grafcet process description method. This improvisation will make the PLC program development become more efficient, reliable and effective.

Keywords: Flowchart, Grafcet , Heuristic, Narratives.

1. INTRODUCTION

Process descriptions are descriptive steps and flow of any process sequences. The PLC programs are developed using these descriptions.

Various types of process descriptions had been used in PLC program development such as Narrative, Boolean, Binary, State diagram, Flow Chart, Grafcet and Heuristic method. Each type has its own strength and weaknesses. The usual practice is that the programmer uses only one type of description as a guide during PLC program development. It is shown that the program built in this form is inefficient with unnecessary usage of extra rung and steps and thus making it not so reliable.

Here, studies and experiments were carried out to improve the efficiency and effectiveness of the PLC program development. Several samples and projects were being tested using various types of process descriptions and the result obtained shows that the combination of the Heuristic method with Grafcet was the best process description in PLC program development.

Further by combining the Initialization Phase of the Narrative method with the Heuristic and Grafcet method a complete process description that is capable of checking and troubleshooting the hardware aspect of the system process and the program development can be achieved.

2. GRAFCET

Grafcet is also known as IEC 848 or sequential function charts (SFC). It is a graphical technique for writing concurrent control programs. Grafcet portrayed the steps and transitions clearly and able to guide the programmer to develop ladder programming precisely. Example of Grafcet is shown in Fig. 1.

From Fig. 1 the important data are as in Table 1.

Table 1 : Extracted Datas From Grafcet

1.	Input devices	PB, LS1, TIM2 contact, LS2, CNT 1 contact, Reset Button
2.	Output devices	Y1, CNT1, TIM2, Lamp
3.	Number of steps / initial rungs	5 basic steps
4.	Transitions or actions	Y1 +, CNT1 on, TIM2 on, TIM2 on, Y1-, Lamp on, Lamp off.

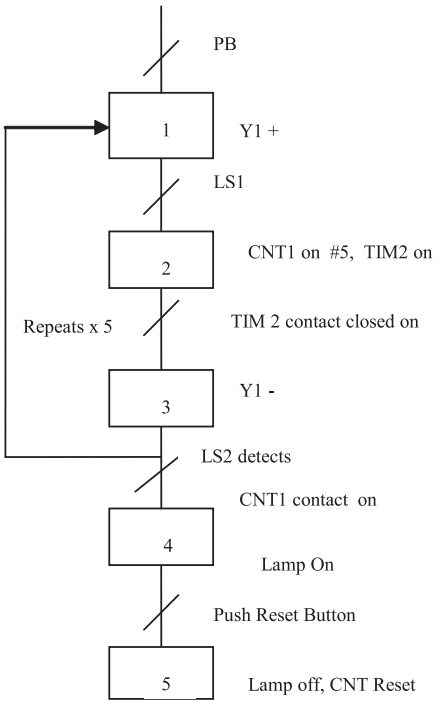

Fig. 1 Example of Grafcet

in Table 1 will guide and aid the PLC program development and hardware selection. It shows that the extraction of data is easy and straight forward and this is the strength of Grafcet if compared to other process descriptions. Further enhancement of the performance of the Grafcet can be achieved by combining it with Heuristic Method.

3. COMBINATION OF GRAFCET AND HEURISTIC

A Heuristic method may include running tests and getting results by trial and error. As more sample data is tested, it becomes easier to create an efficient algorithm to process similar types of data. These algorithms are not always perfect, but work well most of the time. The goal of heuristics is to develop a simple process that generates accurate results in an acceptable amount of time.

Adding Heuristic method into Grafcet results in the improvement of the PLC program performance and the effective steps and number of rungs used. For example from Table 1, in step 1 and step 3 the output is the same, that is Y1. But by combining step 1 and step 3 together only one step is used to control the same output as shown in Table 2. This reduces the redundancy of the unnecessary steps.

4. RESULT

After the first implementation of the Heuristic method to Fig. 1 , step 1 and step 3 can be reduced to one step only to control Y1. The same thing goes to step 4 and step 5 with the lamp as the output. Between step 3 and step 4 there is a repetition loop. The repetition loop needs input LS2 to restart the extraction of Y1 without pressing the PB. In this case, the programmer uses Heuristic method to add LS2 as the input control device that is parallel with PB so that the process will run as planned

Table 2 : First Improvement of Table 1

1.	Input devices	PB, LS1, TIM2 contact, LS2, CNT 1 contact, Reset Button
2.	Output devices	Y1, CNT1, TIM2, Lamp
3.	Number of steps / initial rungs	3 steps for the output control and 1 extra parallel input for repetition
4.	Transitions or actions	Y1 +, CNT1 on, TIM2 on, Y1-, Lamp on, Lamp off.

Further improvement can be made by adding extra steps in Fig. 1. This is to create a control section and the output section. The control section usually uses internal relays. The usage of internal relay as control section will provide isolation and protection for the hardware especially if the outputs are solenoid and motor. Table 3 shows the final improvement combining both Grafcet and Heuristic Method.

Table 3 : Further Improvement of Table 1

1.	Input devices	PB, LS1, TIM2 contact, LS2, CNT 1 contact, Reset Button
2.	Output devices	Y1, CNT1, TIM2, Lamp
3.	Number of steps / initial rungs	5 control steps with IR 1 repetition – LS2 3 output steps for 3 outputs
4.	Transitions or actions	Y1 +, CNT1 on, TIM2 on, Y1-, Lamp on, Lamp off.

From Table 3 the effectiveness, reliability and efficiency of the program and process descriptions have been improved even though the program is longer and indirect.

As a conclusion , by combining other process description with Heuristic method, improvement is being done in the data extraction part and in the PLC program development.

ACKNOWLEDGMENT

The authors would like to thank UTHM for the short term grant that is being used for this research and for giving the chance to the authors to produce this paper for the purposes of knowledge sharing and advance research work.

[1] Dr. John R. Wright, J., *The Debate Over Which PLC Programming Language is the State-of-the-Art*, .Volume 15, Number 4 - August 1999 to October 1999

[2] Finn Haugen, *Introduction to State Diagrams*, April 27, 2005.

[3] Mattias Nilsson, Kristoffer Persson, *Sequential Function Chart Interfacing,* Master Thesis in Industrial Automation at Lund Institute of Technology, Department of Industrial Electrical Engineering and Automation, Lund University, Lund, Sweden 2001.

[4] M.Kudret Yurtseven, *New Technologies and Industrial Automation,* Eastern Mediterranian University, G.Magusa, North Cyprus via Mersin 10 Turkey

[5] Charlotta Johnsson and Karl-Erik Arzen, *Grafchart and Grafcet : A Comparison Between Two Graphical Languages Aimed For Sequential Control Applications*, Lund Institute of Tchnology.

[5] *A concept for the integrated process description, PLC programming and simulation using Petri nets: application in a production process Systems, Man and Cybernetics,* 2000 IEEE International Conference

[7] Gregor Kandare, Giovanni Godena and Stanko Strmcnik, *New approach to PLC software design, 17 January 2007*

[8] Johan Plomp, Pertti Huuskonen,Esko Malm and Jari Paanasalo, *An Object-Oriented Approch to PLC Explaination,* VTT - Electronics, Finland, Proceedings of STeP'96, Jarmo Alander, Timo Honkela and Matti Jakonsson (eds.), Publictions of the Finnish Artificial Intelligence Society.