THE ANNALS "DUNĂREA DE JOS" OF GALAŢI
FASCICLE V, TECHOLOLOGIES IN MECANICAL ENGINEERING, ISSN 1221-4566
2003

Graphic Programming of Programmable Logic Controllers Case Study for a Punching Machine

Vasile MARINESCU, Ionuţ CONSTANTIN
University "Dunarea de Jos" of Galati

ABSTRACT

The Programmable Logic Controller (PLC) plays a vital role in automation and process control. Grafcet is used for representing the control logic, and traditional programming languages are used for describing the pure algorithms. The paper presents an example of implementation for a punching machine for sheet and plates.
***Keywords**: PLC, Grafcet, Punching.*

1. Introduction

Grafcet has its roots the in the Petri nets formalism. Petri nets are a graphical and mathematical model aimed for visualization and simulation of discrete systems. Petri nets are often used in the context of formal methods for analysis of discrete event systems. The advantages of graphical programming languages are simplicity and efficiency.

Although Grafcet uses in essence the same elements as a flowchart. some particularities of the representation offer facilities for both designer and user. This fact leads to spreading of the method.

The basic elements of a Grafcet diagram are:

· Initial steps -symbolize the initial active steps at the beginning of the cycle after initialization or cold restart.

· Simple step- represents a state of operation. A state often has an associated action. The associated actions are performed when the step is active.

· Macro step - a collection of steps.

· Transitions-allow the transfer from one step to another. A transition condition associated with this condition is used to define the logic conditions necessary to cross this transition.

It is possible to simultaneous activate-deactivate several steps (AND divergence-convergence). Also it is possible to select a step sequence in the case of the processes with alternative execution (OR divergence-convergence).

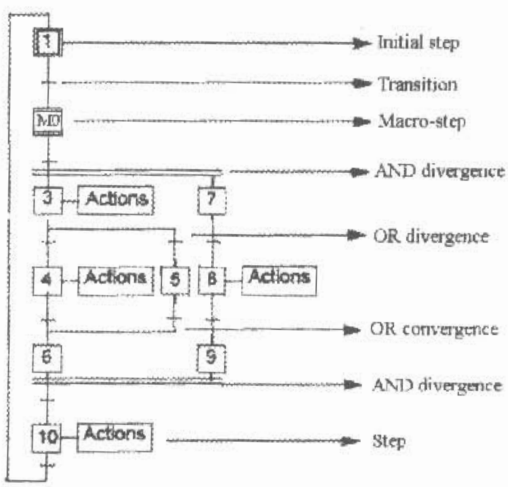

Figure 1. An example of a Grafcet flowchart

2. Case study: Development of a Grafcet chart for a punching machine

We shall present on the elaboration of a command program for a punching machine for plates with two axes for positioning with brushless servo motors with encoder and tachometer and a tool holder turret designed to hold five punches and one marking tool. The command system of the machine consists in a PLC and an industrial computer. The computer is used to assure the human-machine interface. The program that runs on the computer sends to the Programmable Logic Controller (P.L.C.) information about the mode to be selected or the operations to be made and reads from the

P.L.C. data to be displayed to the operator and other information about the state of the machine.
The machine can function in one of these modes: homing, manual and automatic. The homing mode allows finding of the measurement origin point via a specific procedure. If the homing procedure was not completed the movement of the axes cannot be made, so the manual or automatic mode can not be selected.
The information regarding the operation with are to be made in the automatic cycle, introduced by the operator, are taken over and kept in a database by the program that runs into the computer. The data referring to the operations which are executed in the automatic mode are organized in the P.L.C. in two memory areas: one area in which there is information (tool, position of the hole, the fact that the operation received is the last to be executed etc.) referring to the current operation which is executed and a buffer area where is information about the operation to be made. This configuration allows data transfer regarding the operation to be made in the same time with the execution of the current operation.
The program that runs on the computer sends data referring to the operations that are to be executed in the memory buffer area.
The Grafcet section is organized in two pages and contains three independent charts. The left chart in the page 0 shown in figure 2 is used for homing of the axes and right one is for the sequential processing associated to the automatic mode.

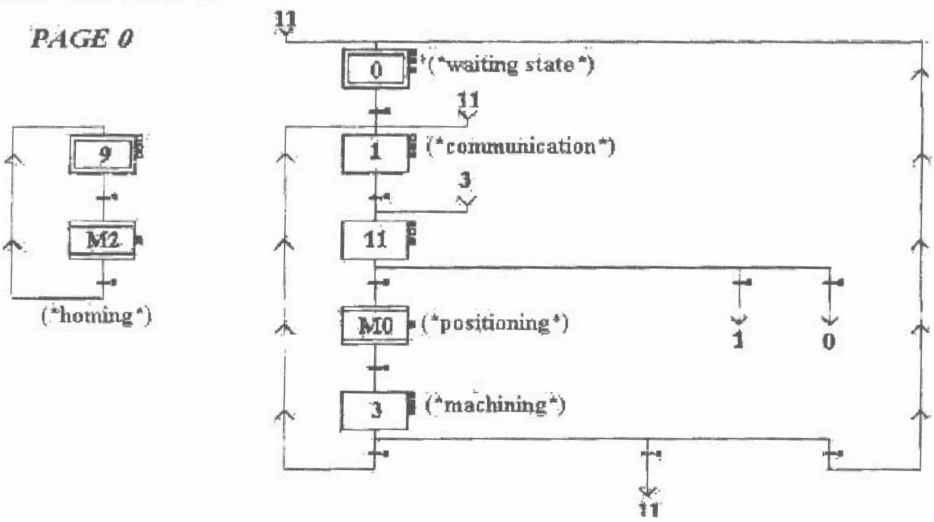

Figure 2. Page 0 of the Grafcet

For each mode there are steps that are active when that mode is inactive. At a certain moment in time at most one mode can be active. The homing of the axes is accomplished in the macro-step M2. This macro-step is active only when the operator requires this procedure and the conditions of the machine allow this operation. After the homing the macro-step is deactivated ad step 9 is activated.
In step 0 (initial step) the activation of the automatic mode is waited. The transition in step 1 is accomplished when this mode is activated. When the step 0 is deactivated, the interface program is notified that it may send the information corresponding to the first operation. In step 1 the data received in the buffer area is transferred in the memory area meant for the current operation. When the transfer is made the transition is accomplished at step 11. Step 1 is deactivated and if the first operation is not the last one, the interface program is notified that it may send the information corresponding to the second operation.
In step 11 the information received is analyzed. If the information received is suitable then the operation is executed (M0-positioning of the turrets and of the axes, step 3 machining). If the information received is inappropriate step 1 will become active if the operation received is not the last one or step 0 will become active when this operation received is the last one.

After the execution of an operation if this operation is not the last one, the transition condition from step 3 to step 1 will be accomplished. The information from the buffer will be transferred in the area referring to the current operation and the PLC will set a bit that informs the interface program that it may write in the buffer area.
After the execution of last operation step 0 will become active and the PLC will be waiting for a

new program execution. If an error appears during the machining or positioning, the safety of the machine and of operator will be assured using a special procedure and the step 11 will become active until the error disappear. The automatic execution is resumed when there are no errors and the operator acknowledges the error.

The first-level Grafcet chart describing the chain of sequences provides greater clarification of the structuring of the control part. Some sequence are associated with a specific symbol for the step: the macro-step.
This idea of "macro-representation" enables the analysis to be organized into a hierarchy. Each level can be completed or changed without affecting the other levels.

For instance we saw that the homing procedure is done in M2. The structure for this macro-step is shown in figure 3.

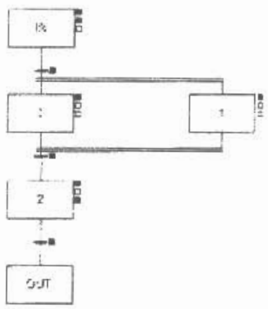

Figure 3. Macro-step M2

The input step of M2 is used for enabling the speed drives used for each axis. After that axes are calibrated using the actions defined in step 0 and in step 1. In step 2 some bits are set so the interface program that runs on the PC is informed that the homing procedure was done.
The Grafcet associated with the macro-step M2 is shown in figure 4.

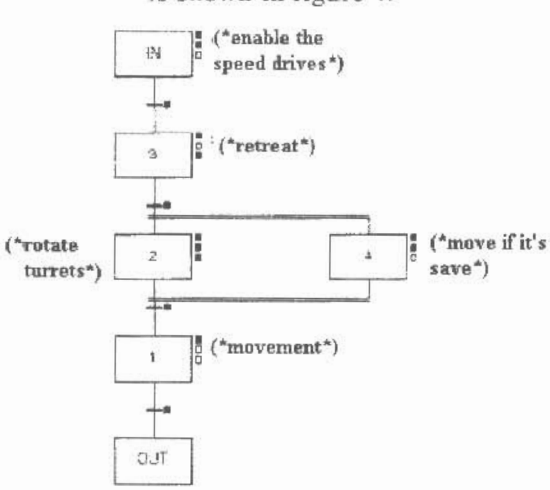

Figure 4. Macro-step M2

The first page of the Grafcet is intended for the sequential processing associated to the manual mode.

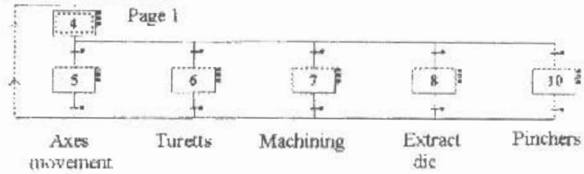

Figure 5. Page1 of the Grafcet.

Step 4 is active when the manual mode is not selected or when this mode is selected and no operation is performed or when an error is detected.
In step 5 the movement of the axes is carried out. In step 6 the tool holder turrets are rotating to the position imposed by the interface program.
In step 7 one of these operations is performed: punching, marking, centre punching. In step 8 the die is extracted and in step 8 the pinchers are opened or closed. The transitions from step 4 to steps 5, 6, 7,8,10 are exclusive so only one of these steps can be active.

3. Conclusions

The increasing complexity of the automatic systems impose the use of high level programming languages such as Grafcet. The use of Grafcet facilitates implementing of the sequential command.

Bibliography

1. **SCHNEIDER ELECTRIC**, *PL7 Micro /Junior /Pro Detailed description of Instructions and Functions;* 2001
2. **V. Marinescu, V. Tabacaru**, *Conducerea sistemelor flexibile de prelucrare;* Ed. Fundației Universitare „Dunărea de Jos", Galați, 2000
3. **M. Ivănescu**, *Roboți industriali-Algoritmi și sisteme de conducere;* Ed. „Universitaria", Craiova, 1994

PROGRAMAREA GRAFICĂ A CONTROLERELOR LOGICE PROGRAMABILE. STUDIU DE CAZ PENTRU COMANDA UNEI MAŞINI DE POANSONAT

Rezumat

Limbajul Grafcet derivă din metodele teoretice de modelare în care sistemele de fabricaţie sunt privite ca maşini cu algoritmi de stare sau ca sisteme cu evenimente discrete Petri Net. În comparaţie cu celelalte metode de programare grafică se distinge prin claritate şi simplitate. Deşi un Grafcet redă în esenţă aceleaşi elemente ca şi o organigramă de stări, anumite particularităţi de reprezentare oferă facilităţi atât proiectantului cât şi utilizatorului, ceea ce a dus la lărgirea ariei de utilizare şi folosirea metodei într-un mare număr de aplicaţii.

Pentru un caz particular se remarcă faptul că programarea controlerului logic programabil este facilitată de folosirea limbajului Grafcet. Gestionarea modurilor de lucru este implementată folosind grafice separate. Depanarea unui program este facilă deoarece se poate vizualiza grafic starea controlerului izolând astfel problemele apărute. Din acest motiv depanarea se face pe module. Realizarea aceleaşi aplicaţii printr-o metodă clasică ar fi ridicat probleme deosebite programatorului.