World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:2, No:10, 2008

Graphical Programming of Programmable Logic Controllers -Case Study for a Punching Machine-

Vasile Marinescu, Ionut Clementin Constantin, Alexandru Epureanu, and Virgil Teodor

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. Grafcet is used for dividing the process to be automated in elementary sequences that can be easily implemented. Each sequence represent a step that has associated actions programmed using textual or graphical languages after case. The programming task is simplified by using a set of subroutines that are used in several steps. The paper presents an example of implementation for a punching machine for sheets and plates. The use the graphical languages the programming of a complex sequential process is a necessary solution. The state of Grafcet can be used for debugging and malfunction determination. The use of the method combined with a set of knowledge acquisition for process application reduces the downtime of the machine and improve the productivity.

Keywords**—**Grafcet, Petrinet, PLC, punching.

I. Introduction

GRAFCET (Sequential Function Charts-SFC) has its roots 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 [1], [8]. The advantages of graphical programming languages are simplicity and efficiency. Grafcet like all the other PLC languages is defined in [2] as Sequential Function Chart.

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.

Manuscript received September 1, 2006. This work was supported in part by the MEC Romania under CEEX 23/2005.

V.M. Marinescu is with the Manufacturing Engineering Department, University Dunărea de Jos, Galati, 47 Domneasca, Romania (phone: +4074093272 , e-mail: vasile.marinescu@ugal.ro).

I. Constantin is with the Manufacturing Engineering Department, University Dunărea de Jos, Galati, 47 Domneasca, Romania (phone: +40745771872, e-mail: ionut.constantin@ugal.ro).

Al. Epureanu is with the Manufacturing Engineering Department, University Dunărea de Jos, Galati, 47 Domneasca, Romania (phone: +40722362606, e-mail: alexandru.epureanu@ugal.ro).

V. Teodor is with the Manufacturing Engineering Department, University Dunărea de Jos, Galati, 47 Domneasca, Romania (e-mail: virgil.teodor@ugal.ro).

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. The associated actions can be programmed with one of the programming languages defined in IEC 1131-3. Each step has an associated Boolean object that is true when the step is active. Many PLC s allow the use of three types of actions:

- actions for activation: actions are executed once when the corresponding step is passing from inactive to active state,
- actions for deactivation: actions are executed once when the corresponding step is passing from active to inactive state,
- continuous actions : these actions are carried out for as long as the step with which they are associated is active.

• Macro step-a collection of steps. The macro steps are used for hierarchical development of the process. A macro step expansion is characterized by two specific steps: an input (IN) step that obeys the same rules as a regular step and an output (OUT) step which cannot have any associated actions. A macro step can contain one or more macro steps. The hierarchy depends of the PLC specification.

• Transitions-allow the transfer from one step to another. A condition associated with this transition is used to define the logic conditions necessary to cross the transition. A transition is enabled if all the steps preceding the transition are active. Some PLC's allow the time dependent programming of the transition conditions. All the active transitions are crossed simultaneously.

The normal Grafcet evolution can be changed using a set of system objects. It is possible to:

- Preset the chart activating or deactivating several steps independent of the transitions state.
- Initialize the chart by deactivating all the regular steps and activating the initial steps,
- Reseting the chart by deactivating all the active steps,
- Freezing the chart by stopping the normal evolution of the chart. No steps are activated irrespective of the state of the active transition condition.

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). A formal model and the operational semantics are described in [3].

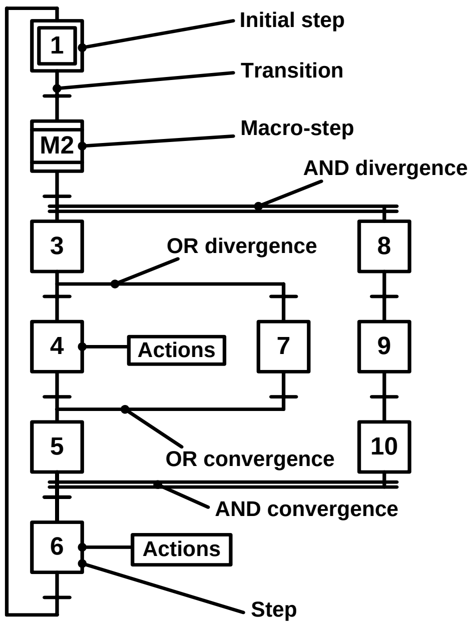

Fig. 1 An example of a Grafcet flowchart

The proposed method presents the facility of dividing a sequential complex process into elementary sequences which can be hierarchical structured. Although some authors recommends more elaborate methods based on Petri Nets [8], using Ladder implementation, Grafcet remains the most used language for programming sequential processes. Each step and transition is programmed using a graphical or a textual language. The choice is made from case to case. Graphical languages are often use for programming Boolean expressions and textual languages are used for numerical processing. For the graphical languages an advantage is represented by the fact that the state of the program can be displayed on a terminal allowing elimination of installation malfunctions. Some similar actions can be programmed as subroutines that can be accessed from several Grafcet steps. This approach reduces the programming code and the time necessary for implementation. Some processes are evolving from an initial state so the Grafcet and the subroutines use knowledge acquisition for process application algorithms.

The programming technique exposed in this paper offers the following advantages:

- the debugging part is simplified,
- the programming part is simplified due the use of the subroutines. Those subroutines have a general character and can be used for other programs.
- the program can adjust its parameters so it can function properly despite the evolution of the system.

Further in this paper we present a case study referring to the proposed procedures for a punching machine for plates.

II. Case Study: Development of a Grafcet Chart for a Punching Machine

The machine has with two movement axes for positioning. The work piece is held by a set of hydraulic pinchers. Each axis is powered by a brushless servo motor equipped with encoder and tachometer. Each motor is controlled using a speed drive. The machine is equipped with a tool holder turret designed to hold five punches and one marking tool arranged at equal intervals of 60°. Similar with the tool holder turret there is a die holding turret. For each punch mounted in the tool holder turret, a matching die must be mounted in the die turret. The machining consists in the hydraulic punching.

The command system of the machine consists in a PLC produced by Schneider Electric. The particularities of the PLC are described in [4] and [6]. The sequential processing was programmed using Grafcet language. The PLC structure is modular. For the positioning task, the PLC is equipped with a dedicated movement module that controls the speed drives and acquires the signal from each incremental encoder.

The human machine interface of the machine is represented by an industrial PC and a set of conventional elements (push buttons, lamps and controllers). The industrial PC is used for running a program developed under the Visual C++ framework. Such architecture is detailed in [9]. The computer program uses an OPC data server to communicate with the PLC using a dedicated protocol. The program that runs on the computer sends to the PLC information about the working mode to be selected or the operations to be made and reads from the PLC data to be displayed to the operator and other information about the state of the machine. The terminal program is also used for displaying the state of the Grafcet chart. The service personal can determine the cause of a malfunction by visualizing the chart.

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 can not be made, so the manual or automatic mode can not be selected.

The information regarding the operation which 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 PLC. 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. Using this buffer the execution is not interrupted because of the low transfer speed.

The program that runs on the computer sends data referring to the operations which are to be executed in the memory buffer area.

The terminal program is also used for displaying and controlling some of the machine parameters. The operator can adjust those parameters so the machine can function properly.

The PLC program is organized into sections. There is a dedicated section for determination of the malfunctions of the machine, a section for managing the working modes, a section the safety locks of the outputs and a section dedicated for the sequential processing- Grafcet. The PLC used for this application allows the graphical programming of the Grafcet. Some PLC's do not have this facility, the Grafcet section is programmed with specific instructions. Due to its flexibility Grafcet is widely used for programming PLC's commanding robots, manipulators, flexible manufacturing systems [5].

The Grafcet section is organized in two pages and contains three independent charts. The left chart in the page 0 shown in Fig. 2 is used for the homing procedures and right one is for the sequential processing associated to the automatic mode.

The sequential process was decomposed into elementary sequences. A hierarchical approach was used for some elements for allowing the decomposing of some sequences into more little entities.

For each work mode there are steps which are active when that mode is inactive. At a certain moment in time at most one mode can be active.

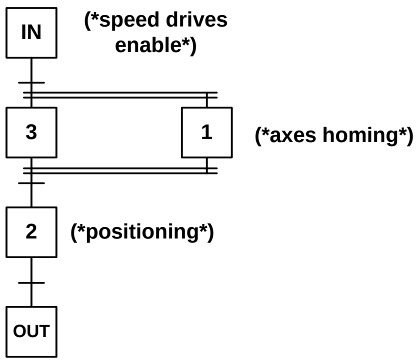

Fig. 3 M2 Macrostep

The homing of the axes is accomplished in the macro-step M2 (Fig. 3). This macro-step is active only when the operator requires this procedure and the conditions of the machine allow this operation. This macro step consists in the following steps:

- IN- step used for enable the speed drives. The condition transition to the following steps is done when the PLC receives the confirmation signals from the speed drives.
- 3 and 1 – in these steps the homing procedure is enabled for each axis. The dedicated module of the PLC commands the movement of each axis to a extreme position where the homing switches are activated by the corresponding cam. The homing procedure is completed when for each axis the Z signal of the incremental encoders is detected.
- 2- this step is used for the movement of the axis in the loading zone. The operator can use the pinchers for holding a piece.
- OUT – the output step, no actions.

After the homing the macro-step is deactivated ad step 9 is activated.

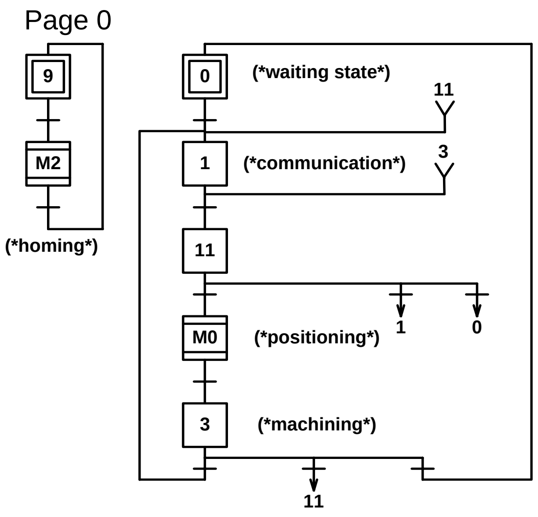

Fig. 2 Page 0 of the Grafcet

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. During the execution of an operation the next operation is transferred into the buffer area.

In step 11 the information received is analyzed. If the information received is suitable then the operation is executed else if the data received from the PC are not correct the execution of the program or the current operation is aborted. The transitions to steps 1 and 0 are used for these tasks

M0 is a macro step, a collection of steps, used for implementing of axes and the turrets positioning. In the input step (IN) the speed drives are activated if the movement of the axes is required. In event that the moving part is too close to the turrets and the turrets must be rotated, in step 3 the moving part is retracted to a safety position. If the moving part is not in the collision zone and the trajectory of the pinchers does not intersect this zone, the rotation of the turrets can be executed simultaneously with the moving of the axes. This task is executed in 3 and 1 steps. The movement of the axes is inhibited if a collision might happen. In this case the positioning task is executed in step 2. This step is also used for deactivating the speed drives if in the current program no more movements are required.

After the positioning of the axes and of the turrets, the machining will be executed in step 3 of the diagram represented in Fig. 2. In this step the following operations can be made:

- Punching – the selected punch pierce through piece until a limit switch is actuated.
- Marking or pointing- the marking or the pointing punch is pressed against the work piece for a fixed time interval. The operator fixes this interval in function of the piece hardness. The PLC reads the hydraulic pressure using a pressure switch. If the pressure is too high the operation is completed regardless of the time interval value.

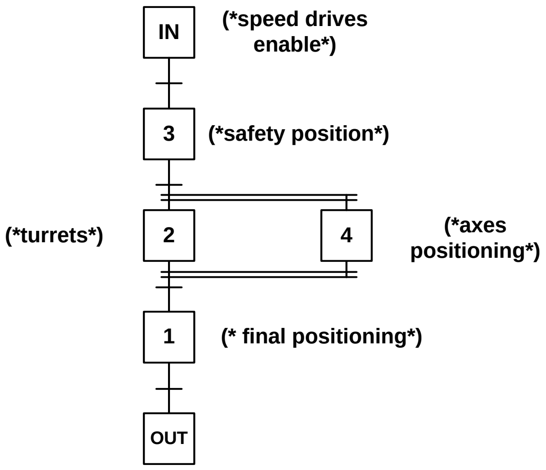

Fig. 4 M0 macrostep

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 assured using a special procedure and the step 11 will become active until the error disappears. The automatic execution is resumed when there are no errors and the operator acknowledges the error.

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

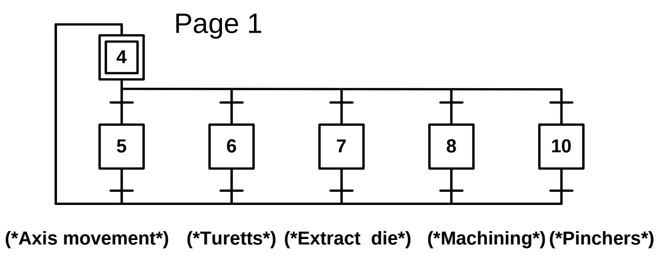

Fig. 5 Page 1 of the Grafcet

The initial step 4 is active when the manual mode is not selected, when the manual 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. The operator commands the movement of the axes with a controller.

In step 6 the tool holder turrets are rotating to the position imposed by the interface program. The PLC checks if the two turrets have the same working support selected the punch and the corresponding die.

In step 8 one of these operations is performed: punching, marking and pointing. The operations are executed using the same procedure as in automatic mode. In fact the actions associated with machining and the positioning steps are programmed using the same programming code. The difference is made by a set of specific parameters. The procedure for the positioning steps is implemented using knowledge acquisition for process. The operator introduces corrections for each position of each axis in case that the positioning error is to high. Those corrections are kept in a database in the terminal. The PLC analyzes those correction elements and act properly reducing the deviation.

In step 8 the selected die is extracted. This procedure is used for changing a tool from the turrets.

In step 10 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.

The programming task of the sequential process was simplified using Grafcet. At the HMI level the operator can visualize the Grafcet diagram and in event of a malfunction of the machine the cause can be easily determined. The actions and the transitions were also programmed using a graphical language: Ladder. For each rang the programmer can see the state of any Boolean object, and the value of the numerical objects. The program is similar with a wiring diagram and is very effective for logical expressions.

This program can be implemented using state diagrams as well but the programming and the debugging task would be very difficult because the whole sequential process would have to be implemented using a specific logic. The debugging of such program would be very difficult. The method proposed is also based on using some general subroutines that would be used in several sequences. Developing separate subroutines for each action is often a cause of the programming bugs. The program uses knowledge acquisition for process application algorithms that allow the operator to input corrections without the program modification

III. Conclusion

The increasing complexity of the automatic systems imposes the use of high level programming languages such as Grafcet. The use of Grafcet facilitates implementing of the sequential command. The case study presented in the paper is used to demonstrate that the programming task is simplified because the sequential processing was implemented using Grafcet. In the case presented in this paper each working mode was treated in separate graphs. The debugging is greatly simplified because the state of the chart can be visualized isolating the problems. The debugging part can be treated as modular task. The same program would be more complex if the sequential task would be described using a "classical" method such as state diagrams.

REFERENCES

[1] R. David and H. Alla. Petri Nets & Grafcet. Prentice Hall, 1992
[2] International Electrotechnical Commission, Technical Committee No. 65. Programmable Controllers – Programming Languages, IEC 61131-3, second edition, November 1998. Committee draft.
[3] N. Bauer, R. Huuck, and B. Lukoschus. A parameterized semantics for sequential function charts. Institute of Computer Science and Applied Mathematics, University of Kiel, 2001.
[4] SCHNEIDER ELECTRIC, PL7 Micro/Junior/Pro Detailed description of Instructions and Functions. 2001.
[5] V. Marinescu, Conducerea sistemelor flexibile de prelucrare, Ed. Fundației Universitare „ Dunărea de Jos", Galați,
[6] V. Marinescu - Sisteme şi Echipamente de Comandă Numerică - Volumul 1 -Controlere Logice Programabile, Editura Cartea Universitară, Bucureşti, ISBN 973-7956-14-1, 2004
[7] V. Marinescu - Sisteme şi Echipamente de Comandă Numerică - Volumul 2 -Comanda maşinilor de prelucrat prin ştanţare, Editura Cartea Universitară, Bucureşti, ISBN 973-7956-15-x, 2004
[8] M. Uzam , A. H. Jones. Discrete Event Control System Design Using Automation Petri Nets and their Ladder Diagram Implementation Advanced Manufacturing Technology, Springer-Verlag London Limited, 1998
[9] A. Ramirez Serrano, S.C. ZHU, S.K.H. Chan A hybrid PC/PLC architecture for manufacturing system control- theory and implementation, Journal of Intelligent Manufacturing,13, 261-281, Kluwer Academic Publisher, 2002

.