IAENG International Journal of Computer Science, 45:1, IJCS_45_1_03

A Study on the Control of Carrying Robots Using SFC Programs

Heon-Tag Kong, Jeong-Bong You

***Abstract*—Those robots that feed LCDs in LCD manufacturing processes are called carrying robots. In general, programmable logic controllers (PLCs) are used as control components in semiconductor manufacturing equipment. PLCs not only have higher processing speeds compared to other control systems and complex functions and can easily cope with various forms of functional changes but also are very conveniently maintained. Ladder diagrams (LDs) are the most frequently used as control programs to drive PLCs. However, a sequential function chart (SFC) is used to complement shortcomings of LD programs. Advantages of these SFC programs are easily understandable control flows and easy maintenance. In the present study, a method of controlling carrying robots in LCD manufacturing processes using SFC language was developed and the validity of the method was checked.**

***Index Terms*— PLC, ladder diagram, SFC, Carrying Robot, LCD**

I. Introduction

LCDs are displays that have been used the most widely recently. Thus far, quite a few studies have been conducted to enhance the resolution of LCDs and many of those studies have been commercialized. The replacement of LDC back lights by LED with a view to enhancing the resolution of LCDs led to the birth of LED TVs resulting in the realization of high resolution. To enhance the resolution of existing LCD displays without replacing their back lights with LEDs, photoregist (Photo Regist) processing processes are very important. The equipment that processes photoregist in LCD equipment is called in-line coater. This in-line coater is divided into five processes; input unit, spin coater unit, dry unit, edge rinse unit, and output unit. The input unit plays the role of receiving glasses to be coated from the glass loader in the upstream, aligning the glasses so that they can be coated in the spin coater unit, and handing over the glasses to the spin coater unit. The spin coater unit spreads photoregist thin on the glasses received from the input unit through a slit nozzle and spins the glasses to evenly spread the photoregist. The dry unit had not been included in medium sized or smaller coaters and it decompresses and dries the glasses in order to prevent traces from being formed on the back of the glasses due to temperature differences between the vacuum chuck of the spin coater unit and the vacuum chuck of the edge rinse unit. The edge rinse unit rinses the edges of the glasses with the rinse nozzle to remove the photoregist deposited on the cross sections of glasses while the photoregist was spread by the spin coater and remains on the edges until the glasses were handed over to the edge rinse unit through the dry unit. Finally, the output unit aligns the glasses received from the edge rinse unit and hands over the aligned glasses to the downstream device[1].

LCD glasses coated with photoregist are manufactured through these processes and these processes play an important role for the quality of LCDs. In addition, the process to feed LCD glasses also plays an important role that affects the quality and carrying robots feed the LCDs. The present study is intended to use PLCs to control individual processes of the in-line coater and carrying robots, use SFC programs as PLC control programs for more convenient control, and check the validity of the foregoing.

Manuscript received June 15, 2017, revised August 20, 2017

H.T. Kong is with Dept. Computer Science Eng., Kongju National University, Korea (e-mail: Htkong@kongju.ac.kr).

J. B. You is with Dept. of Electrical, Electronics and Control Eng., Kongju national University, Korea

(corresponding author e-mail: jbyou@kongju.ac.kr).

II. Definition of SFC language

A. Definition

SFCs are graphs that have two forms of nodes; step and transition and each SFC has at least one step and one transition condition[2]. In addition, directed arcs are lines that connect between steps and transition conditions and between transition conditions and steps.

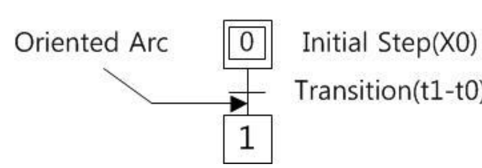

Fig. 1. Structure of SFC

Steps are expressed as squares as shown in Fig. 1 and have two different states; one is an activated state expressed by tokens and the other is an inactive state. The step that is activated when the system starts is called initial step and expressed by a double square. Steps are related to actions in the system and correspond to outputs in SFCs[3] Transition conditions are represented by transition t1-t0 in Fig. 1 and indicate conditions necessary for transition from a step to the next step. Vertical arcs run downward. Transition conditions

(Advance online publication: 10 February 2018)

are represented by bars including double bars. In this case, double bars are used as AND or OR conditions to combine two or more transition conditions and sometimes wait for multiple inputs that must be activated before transition conditions are fired. In addition, transitions are also used to separate two or more transition conditions. Directed links should be connected from a step to a transition or from a transition to a step.

B. *Firing rule*

SFC firing rules are as follows[4]

① Two steps cannot be connected directly with ea ch other. Two steps should be always separated by one transition.

② Two transitions cannot be connected directly wit h each other. Two transitions should be always s eparated by one step.

③ If one step is connected to two or more steps, se quences for such steps should be initiated simul taneously and simultaneous sequences progress independently.

④ If more than one transition conditions are simult aneously true, all activations and deactivations a ssociated with steps will occur simultaneously.

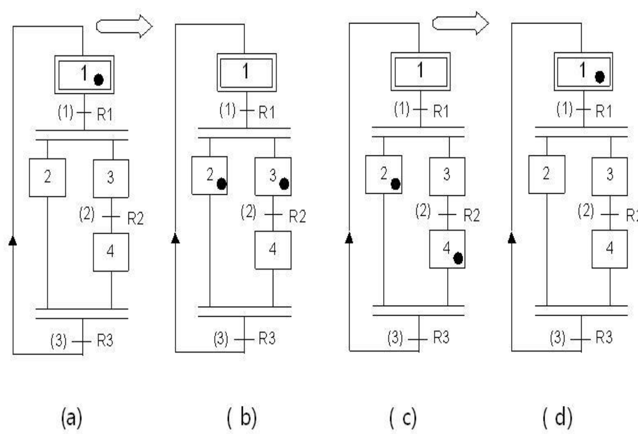

Fig. 2. Evolution of Situations

Fig. 2 shows the evolutions of SFC situations with four steps and three transition conditions. Situations are expressed as sets of active steps and the initial situation in Fig. 2(a) is step 1. In this situation, only transition condition (1) can be fired and will be fired when R1 becomes true. When a transition has been fired, all input steps of the transition are deactivated and the output steps of the transition are activated. These actions cannot be separated and are performed simultaneously. When transition condition (1) has been fired, situation step 1 leads to situation steps 2 and 3. As shown in Fig. 2(b), steps 2 and 3 were activated simultaneously. Now, the firing of transition condition (2) leads to situation steps 2 and 4 and as shown in Fig. 2(c), the firing of transition condition (3) leads to returning to the initial state. As shown in Fig. 2(c), steps 2 and 4 were deactivated and step 1, which is the initial step, was activated.

C. *Actions and action qualifiers*

Actions are represented by squares attached to steps and described by an IEC language, that is, one of ST, LD, and IL[5]. Fig. 3 shows a form of SFC actions. All actions have names unique in the current program constituting units. One of IEC languages can be used to describe actions and each action has a qualifier[4]. The 'N' qualifier in step 0 in Fig. 3 indicates that the action is carried out only when the related step is active and the 'P' qualifier in step 1 means pulses, that is, either rising edge pulses or falling edge pulses. The kinds of these qualifiers are as shown in Table 1[6,7]. As shown in Table 1, qualifiers mean prompting specific actions of individual associated steps.

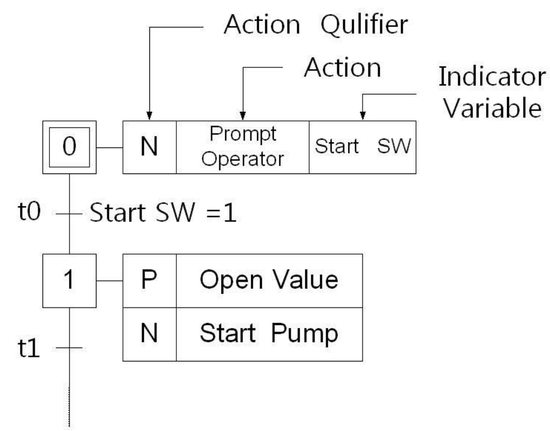

Fig. 3. Main Feature of SFC Action

TABLE I.
SFC Action Qualifiers

Qualifier	Abbreviations	Description
N	Non Stored	Carried out while related steps are active
R	Reset	Reset stored actions
S	Set	Set active actions
L	Limited	Terminate after the given time
D	Time Delayed	Start after the given time
P	Pulse	Pulse action carried out once when the step is activated or deactivated
SD	Stored & Time Delayed	The action is activated after the given time following the step setting
DS	Time Delayed & Stored	The action is activated and set after the given time
SL	Stored and Time Limited	The action is started and carried out for the given time

III. SYSTEM DESIGN

A. *Overview of carrying robot systems*

Carrying robot controlling methods are divided into linear methods and indexing method. Linear feeding methods are implemented by shuttles to reduce the spatial size of equipment[8]. In the present study, carrying robots are controlled using linear feeding methods using shuttles.

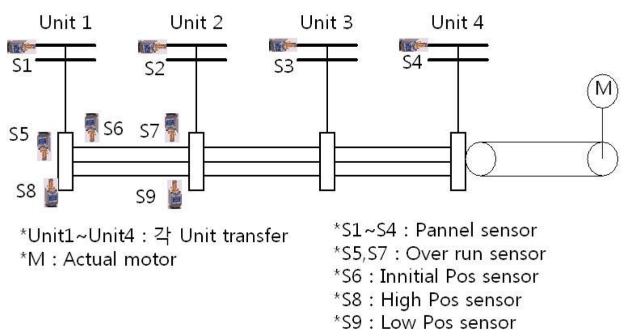

Fig. 4. Configuration of carrying robots

B. *Algorithm design*

The carrying robot algorithm to be applied to the present study was designed[9]. The carrying robot is driven through five different processes implemented by five units; an input unit, a spin coater unit, a dry unit, an edge rinse unit, and an output unit. When the individual units have completed their own actions, the carrying robot moves the glasses simultaneously. Carrying robots' many actions include an action to return to the origin point which is the initial position, manual operation, that is, movements of the robot by the operator without any automatic movement of the robot, a function to enable the robot to move automatically, and an alarm function. The carrying robot is returned to the origin point before starting working so that working begins at the initial position. Step 9 shows the unique feeding works for individual units and step 12 shows the unique works of individual units. In addition, step 19 is works to prepare unique feeding to individual units in which the carrying robot is preparing for feeding to the next units. The alarm rings when any problem has occurred during any of unique works of individual units or when any error has occurred. When any alarm has rung, all works are stopped so that the cause of the alarm is found and related problems are solved before restarting the works. When a series of works has been completed, the sequence goes back to the initial step and the works begins from the initial step. The control algorithm was prepared using SFC language so that all works could be seen at a glance and problems could be easily solved when any error has occurred. In addition, individual steps and transitions in the SFC were designed to be programmable using LD language. The algorithm for the carrying robot is as shown in Fig. 5.

Important matters in the algorithm shown in Fig. 5 are that steps 9, 12, and 19 are processed as parallel sequences and these parallel sequences are processed as synchronous steps to process individual works and the works are completed through re-synchronous steps. However, unique works completed first should wait in the wait step until other works are completed no matter what the unique works are. This is the core of parallel sequences. However, these parallel sequences can be also processed simply by processing them with a macro action. However, although this macro action is efficient because all works are processed simultaneously, the processing becomes uncertain because it is not known to which step the sequence should jump when the macro action has been completed.

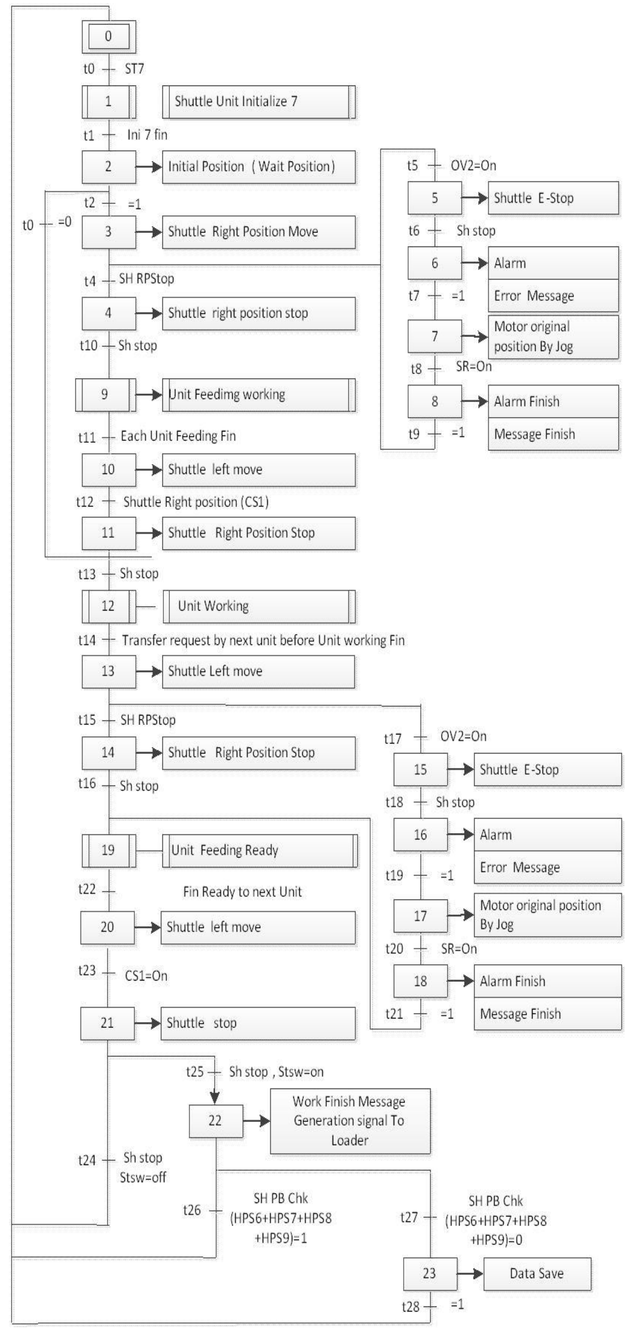

Fig. 5. Algorithm of Carrying Robot

IV. SIMULATION

The carrying robot algorithm set forth in Fig. 6 was applied to a GLOFA PLC and the PLC used was a GM4 CPU. GM-Win was used as an edit program to drive this PLC. The characteristics of the GM-Win program include the fact that it enables editing and debugging of many programs simultaneously and that it provides diverse languages that satisfy IEC 1131-3 such as LD, SFC, and IL so that a language easily applicable to the system can be selected and used. In addition, since GM-Win programs are written for individual projects in one PLC system, many programs can be included and this makes writing and testing programs much easier. In addition, programs written using GM-Win can be verified by performing a function that is like operating the PLC from the computer without actual connection with the PLC.

Using this GM-Win, the entire algorithm was written in SFC language and individual steps and transition conditions of the SFC were programmed in LD language. However, in the case of control units in which a PLC is used, when SFC language is used, programs cannot be partially written in LD language and partially in SFC language. Fig. 6 shows the entire results of algorithm simulations.

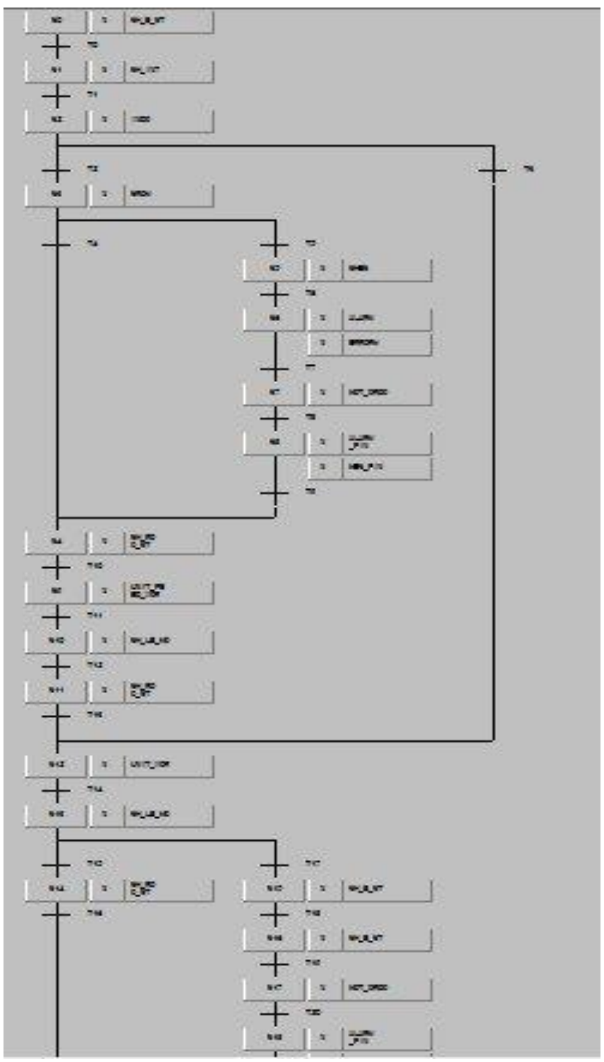

Fig. 6. The Result of Simulation

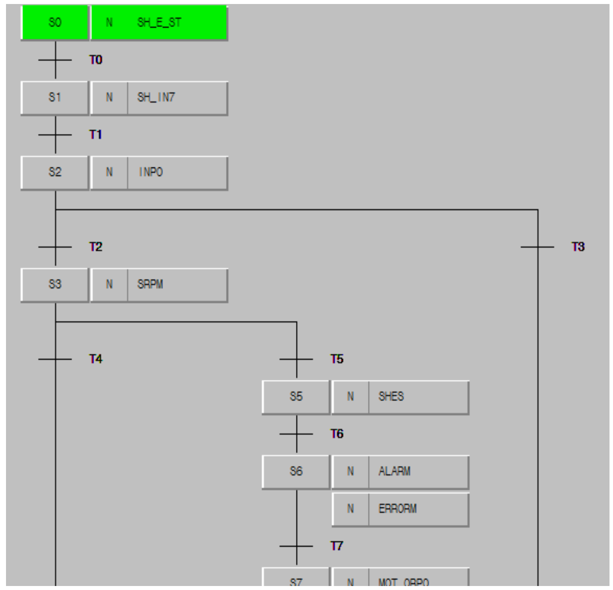

Fig. 7. State where the initial step was activated

When the power has been turned on in the state shown in Fig. 6, step S0 is activated and progressing to the next step is prepared as shown in Fig. 7. Fig. 8 shows a state where step 3 was activated because t2 condition had been satisfied in step 2, which is a routine selected at the selection branch. Fig. 9 shows a routine selected as t5 condition was satisfied in a state where step 3 had been activated.

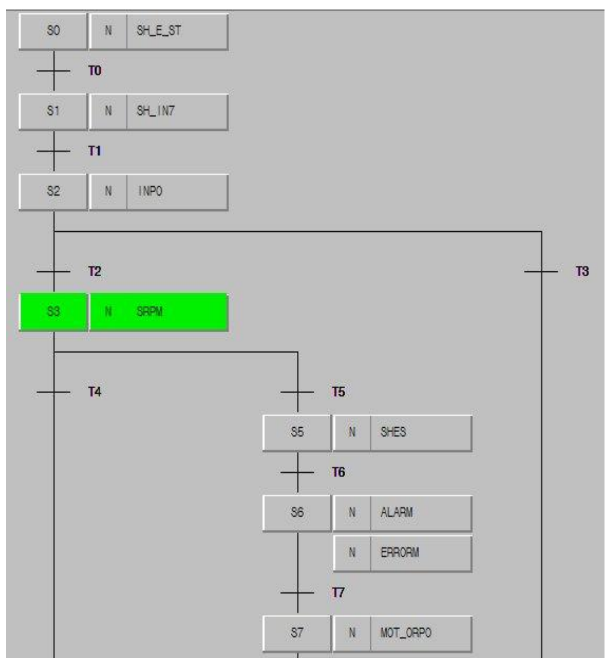

Fig. 8. A state where step S3 has been activated

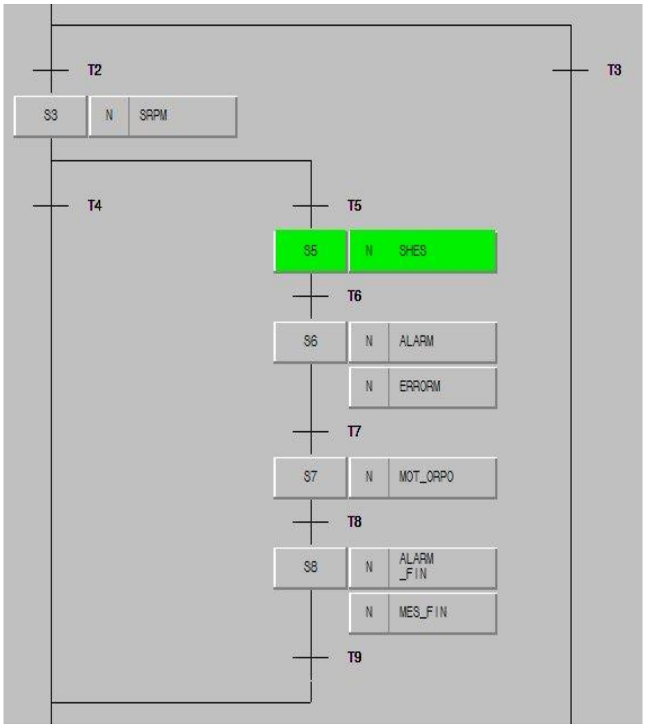

Fig. 9. A state where step s5 has been selected

V. Conclusion

As can be seen from the results of simulations, even when selection branches have been set, those selection branches are activated depending on individual transition conditions. Carrying robots can use indexing feeding methods or linear feeding method. In the present study, a linear feeding method was adopted to conduct actual simulations. The fact that linear feeding methods can reduce the spatial size of equipment and can remarkably reduce memory sizes could be identified through the present study. In addition, by writing programs using SFC language, it could be identified that SFC language is a graphic language suitable for indicating Sequential Control Logics. In addition, it could also be seen that SFC language enables easy maintenance and show excellent diagnostic ability for mechanical disorders.

In process controls being implemented today, the frequency of use of SFC language is becoming higher compared to LD language and efforts are being made to complement the shortcomings of SFC language. Studies for control of carrying robots using SFC language are expected to greatly contribute to other studies.

References

[1] Shih-Lun Chen; Hsin-Ju Tsai, A Vovel Adaptive Local Dimming Backlight Control Chip Design Based on Gaussian Distribution for Liquid Crystal Displays. IEEE Journal of Display Technology, 2016, 12(12), pp.1494-1505.
[2] IAN G. Warnock. Programmable Controllers Operation and Application. Prentice Hall. 1992
[3] Devinder Thapa, Chang Mok Park, Snag C Park and Gi Nam Wang. Auto-Ganeration of IEC Standard PLC Code Using t-MPSG. International Jounal of Control. Automation and Systems. 2009, 7(2), pp. 165-174.
[4] A. Selwin Mich Priyadharson, S. Vinson Joshua, C. ThilipKumar. PLC-HMI and Ethernet based Monitoring and Control of Mimo System in a Petrochemical Industry. Indian Journal of Science & Technology. 2015, 8(27).
[5] Jeong-Bong You. Improvement Implementation of Interlock Using Management Step Described by SFC. The Korean Institute of Illuminating and Electrical Installation Engineers. 2005 May, 19(3).
[6] GLOFA GM4C Manual. 2003.
[7] Mitsubishi PLC Programming Manual. Mitsubishi. QnA series, 2010.
[8] Babak Rooholahi, P. Lokender Reddy. Concept and Application of PID Control and Implementation of Continuous PID Controller in Siemens PLCs. Indian Journal of Science & Technology, 2015, 8(35).
[9] Gajanan Kolmare; Prasad Bhase; Amruta Deshpande, Development of control system for an acid solution formation plant using DCS, 2016 IEEE international Conference on Computing, communication and Automation, pp.977-981.