

PLC Functional Instruction in Programming of the Application

Jiayan Song

Nanjing Institute of Railway Technology, Nanjing, Jiangsu, 210031, China

Abstract—In this paper, the function characteristics of instruction are introduced, and the focus is on contact compare instruction and the characteristics of the transfer instructions. With the traffic lights control as an example, this paper proposes the application of contact compare instruction and transfer instruction to program design method, making the program structure compact and statement concise. The control requirements are easily completed.

Keywords—function instruction; programming; traffic lights; contact compare instruction; transfer.

I. INTRODUCTION

Modern PLC can process logical switch quantity and data. Basic logic instruction is mainly used for processing logical volumes, and function instruction is for processing digital quantity, including data transmission, transformation, operation, and the program flow control. In addition, function instruction is mainly used for processing PLC and controlling the data of peripheral equipment. Mitsubishi FX series PLC provides the rich resources about function instruction, and also there are more and more types of PLC with increasingly powerful functions, which can be more easily applied.

Contact compare instruction and move instruction (MOV) are the most common in instruction functions, which can be the basics in instruction functions. With the application of these two instructions to programming, the advantages such as compact structure, concise statement, easy programming, and flexible modification are available. In this paper, the application of transmission instruction and compare instruction to programming is introduced by focusing on traffic lights control program.

II. THE FEATURES OF INSTRUCTION

In PLC, in fact, most instruction functions are some subroutines to accomplish a specific task. In the application, a corresponding operand is filled according to the requirements of function instructions, and then it (actually a corresponding subroutine) is driven in the program to complete the function operation of represented by the function instruction.

Contact compare instruction and move instruction (MOV) are the most common in instruction functions, and their main function is to read/write and reset soft components and compare/exchange sub-component, etc. These instructions are the foundations of PLC for a variety of data processing and numerical calculation, and their application can make the control of some logical

operation controls simplified and optimized.

A. Contact compare instruction

Contact compare instruction is essentially a normally open contact, and its actuation is affected by the comparison result of S1 and S2 in the instruction, but not bit unit input (X) or word unit coil output (Y, M, S). The contact will be actuated if the comparison condition is established, and otherwise it will fail to be actuated. In the process of use, it is placed on the horizontal line of ladder diagram like ordinary contacts, so it is called as online compare order.

B. MOV

Move instruction (MOV) is the basic function instruction that is the mostly frequently used. It is essentially an instruction to set bit unit and read/write word unit. With the application of combination bit unit to resetting and setting operations, more outputs can be implemented, but also the programs can be significantly simplified; it is unnecessary to modify the programs one by one as long as the numerical value of the output word is modified if there is any change to the output.

III. ANALYSIS ON TRAFFIC LIGHTS CONTROL SYSTEM

In the north, south, east and west directions of city crossroad, red, green and yellow traffic lights are furnished. For traffic safety, red, green and yellow traffic lights will go on necessarily according to a specific sequence.

A. The requirement of the control system

The green light in the north and south directions goes on when the red light in the east and west directions goes on; the green light in the north and south directions will flash 3 times after 25s, and then the yellow light in the north and south directions goes on for 2s. The green light in the east and west directions and the red light in the north and south directions go on when the yellow light goes out in the north and south directions; the green light in the east and west directions flashes 3 times (1s each time) after 25s, and subsequently the yellow light in the east and west directions goes on for 2s (meaning the red light will go on). Traffic lights move in the cycles repeatedly. The control model of traffic lights is shown in Fig.1.

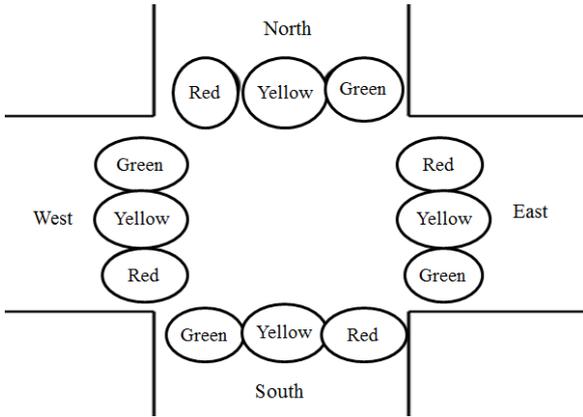


Figure 1. The model of traffic lights

B. Determining system I/O

12 traffic lights are set up generally, but the lights in the east-west direction and the lights in the north-south direction will work together as long as they belong to the same group. To save output ports, the same colored lights in the same directions are connected together. Finally, the half of PLC output ports is saved. Thus, 6 output relays are only necessary. The I/O wiring diagram of traffic lights is drawn, as shown in Fig.2.

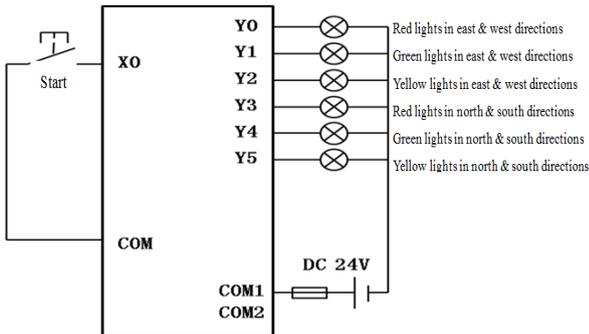


Figure 2. The I/O wiring diagram of traffic lights

According to the design requirements, the I/O address distribution of the system can be determined, as shown in Tab.1.

TABLE I. THE I/O ADDRESS DISTRIBUTION OF THE SYSTEM

Input		Output	
Start button (SB)	X0	The red lights in the east and west directions	Y0
		The green lights in the east and west directions	Y1
		The yellow lights in the east and west directions	Y2
		The red lights in the north and south directions	Y3
		The green lights in the north and south directions	Y4

		and south directions	
		The yellow lights in the north and south directions	Y5

C. System timing sequence diagram

According to the design requirements of traffic lights and combining with the system I/O, the traffic lights timing sequence diagram is drawn using number axis, as shown in Fig.3. The working process of the system is represented using timing sequence diagram, so the complex control requirements can be simplified and also programming ideas can be found.

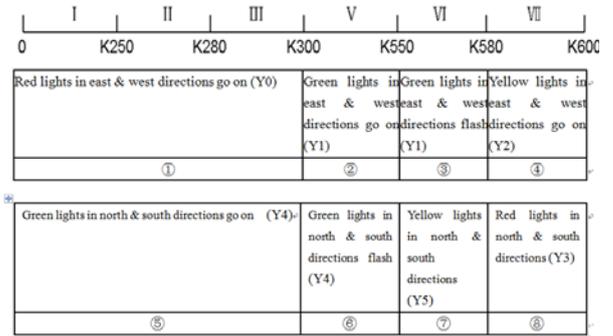


Figure 3. The timing sequence of traffic lights

Through mapping timing sequence diagram, the system can be divided into 6 periods (I~VII) or 8 states (①~⑧). Each period and each state have a corresponding out, as shown in Tab.2.

TABLE II. OUTPUT, STATE, STATE NO. AND PERIOD

Output	State	State No.	Period
Y0	The red lights in the east and west directions goes on for 30S	①	I II III
Y1	The green lights in the east and west directions goes on for 25S	②	V
	The green lights in the east and west directions flash for 3S	③	VI
Y2	The yellow lights in the east and west directions flash for 2S	④	VII
Y4	The green lights in the north and south directions go on for 25S	⑤	I
	The green lights in the north and south directions flash for 3S	⑥	II
Y5	The yellow lights in the north and south directions goes on for 2S	⑦	III
Y3	The red lights in the north and south directions go on for 30S	⑧	V VI VII

IV. PROGRAMMING METHOD

In traffic lights control system programming, function instructions such as step-by-step instruction, displacement

instruction, and setting/reset instruction are used mostly, but there are so many intermediate auxiliary components that a lot of inconvenience is caused for the programming even if it is much more simple and easier to understand than the general basic instruction programming. According to the characteristics of contact compare instruction and MOV and combining with the requirement of traffic lights control, programming is implemented using MOV and compare instruction and features easy to use, clear logic, easy to understand, and easy to control, etc.

A. Implement traffic lights control using contact compare instruction

Contact compare instruction can easily express the meaning of each comparison and use time as the comparing condition of word unit, so as to implement the control of traffic lights. In Fig.3, the eight working states (①~⑧) of traffic lights in different periods are listed using number axis. The periods corresponding to states (①~⑧) are 30S, 25S, 3S, 2S, 30S, 25S, 3S and 2S, respectively. These 8 periods are compared as the conditions if whether traffic lights are in the working state. As long as the conditions are fulfilled, normally open contact will be closed, that is, the lights will be powered on once coil is output. The control program is as shown in Fig.4.

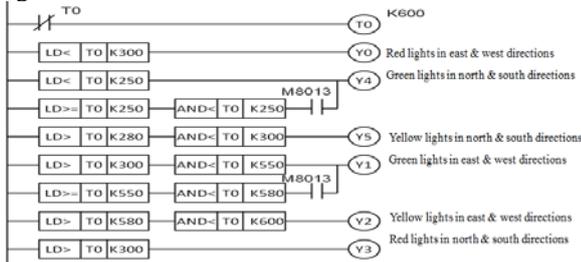


Figure 4. The traffic lights controlled by contact compare instruction

From Fig.4, it is known that the coil of each output relay is the only, making the double coil fault prevented. ">K250, <K280" needs two contact compare instructions: LD>= and AND<, so the current state of traffic lights is "green lights in the north and south directions flash". From this example, it can be seen that contact compare instruction is normally open and its position is equivalent to a closed contact allowing current flow from it as long as its own judgment is right.

B. Implement traffic lights control using move instruction

MOV is a function of transmitting data, which can implement traffic lights control using bit operation with the same requirement. Using the computer binary working principle, four-bit binary number constituted one-bit hexadecimal number. Therefore, the system output can be seen as two-bit hexadecimal number; in computer arithmetics, "1" represents component is in working state, and "0" means component is not in working state. If output changes, modifying the numerical value of output word is only necessary, but modifying the programs one

by one is unnecessary, from which great convenience and operability are embodied.

From Fig.3, it is known that periods I ~VII are 20S, 3S, 2S, 20S, 3S and 2S respectively. The working state of each light in each period is as shown in table 3.

In Tab.3, different numbers can be given to coil group K2Y0 (Y0-Y7), in order to let green light flash. Before programming, all the meanings represented by Y in each period in table are necessarily understood. For example, the current state of the traffic lights is "the red lights in the east and west directions and the green lights in the north and south directions go on" if K2Y0=00010001 and Y4/Y1=1.

TABLE.III. 0, 1 OF TRAFFIC LIGHTS IN EACH PERIOD

Time	Period	Converted to hexadecimal number
		K2Y0(Y7Y6Y5Y4Y3Y2Y1Y0)
0~K250	I	(0001 0001) ₂ =H11
K250~K280	II	(0001 0001) ₂ or (0000 0001) ₂ =H11 or H01
K280~K300	III	(0010 0001) ₂ =H21
K300~K550	V	(0000 1010) ₂ =H0A
K550~K580	VI	(0000 1010) ₂ or (0000 1000) ₂ =H0A or H08
K580~600	VII	(0000 1100) ₂ =H0C

Tab.3 is used for designing the ladder diagram of traffic lights control program, as shown in Fig.5.

In Fig.5, double coil problem exists, but there are a lot of instructions loading numerical value into K2Y0 in program line, and all of them operate coil Y0-Y7. If multiple contact compare instructions in Fig.5 need to satisfy, the function of MOV to load numerical value into K2Y0 will be executed, but the program line executed in the bottom is saved finally by circular scanning. Thus, the "double coil" here won't cause the logic errors in traffic lights control.

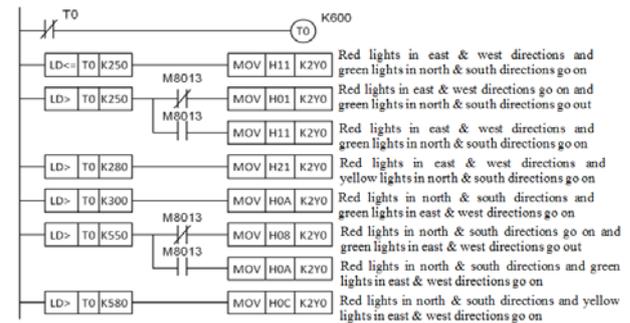


Figure 5. Implementing traffic lights control using MOV

V. CONCLUSION

Functional instruction itself provides programmers with a great space of imagination. In the process of programming, only the basic logic instruction of programmable controller is not enough. If some functional instructions can be applied, the programming problems are likely to be solved smoothly, and

programming will be handy and flexible so that an unexpected result may be achieved.

So far, the control of traffic lights has been implemented using many methods such as basic instruction, step-by-step instruction, displacement instruction, contact comparison, and MOVE, but the control tasks for all these methods are identical. Therefore, it is known that program execution can be slow or fast, and the algorithm has advantages and disadvantages, and the same control effect can be achieved using different methods even if the starting point is different. Programmers can build up their own program thinking as long as they can accumulate much experience.

REFERENCES

- [1] Wei QIN. CAM Sequence Control Program Design and Practice [J]. Journal of Mechanical Engineering and Automation, 2012 (03): 136~138.
- [2] PLC Project-oriented Instruction, Competition and Engineering Practice [M]. Beijing: Mechanical Industry Press, 2012.4.
- [3] Changchu LIAO, Lin ZHOU. The Functional Instruction and Application of Programmable Controller (I): An Introduction to FX series PLC functional Instruction [J]. Electrical Technology, 1999 (7): 16~19.
- [4] Min LV. The Application of Functional Instructions to PLC program [J]. Journal of Wuxi Vocational College, 2006 (04). 38~40.
- [5] Zhemin MI, Jianming YAN. The Application of Functional Instruction to Timing Control Circuit [J]. Journal of Programmable Controller and Factory Automation, 2010 (12): 110~113.