Development of simulation platform for Automatic Storage production line

Qinghua Wang¹, Yanhua Hu²*, Renxuan Fu¹, Junyong He¹, Xinzhou Ma¹ and Jun Wang¹

¹College of mechanical and electrical engineering, Guangdong College of Industry and Commerce, Guangzhou, Guangdong, 510510, China
²School of economics and management, Guangdong Polytechnic of Engineering, Guangzhou, Guangdong, 510000, China
*Corresponding author’s e-mail: wqh@guangdgm.onaliyun.com

Abstract. This paper develops a simulation platform of automatic storage production line based on 89C51 single chip microcomputer, which provides students with an interactive classroom learning and laboratory practical activities to assist the practical learning of industrial robot and programmable controller. The test shows that the transformation of the simulation code into the corresponding actual equipment is real, feasible, safe and reliable.

1. Introduction
According to the characteristics of applied talents education in robotics engineering, to realize the teaching concept of "learning in doing" and "doing in learning", we need to strengthen the learning of industrial robot simulation environment such as irai[1-2], robot studio[3-4], robot master[5], robcad[6], workspace[7] and robot art[8] through practical application cases, and the learning of commercial programmable controller software such as solpica[9], Estevez et al.[10], whitmori CDK[11], GX works, TIA portal. However, many of these two kinds of software are independent and can not well meet the requirements of integrated learning of automatic storage production line.

In [12], a simulation platform of intelligent production line based on single chip microcomputer is proposed, and different solutions are proposed for different programming languages and methods of industrial robot and programmable controller. Based on the above methods, this paper realizes the simulation system of automatic storage production line integrating an industrial robot and a Siemens programmable controller s7-1200, and verifies the feasibility of code conversion of this scheme, as shown in Figure 1.

2. BASIC PRINCIPLES
According to [12], in terms of system hardware structure, this paper uses one 89C51 single chip microcomputer to simulate the industrial robot in automatic storage, another one to simulate Siemens PLC, led, motor, button and switch to simulate the peripheral input and output equipment of Siemens PLC. In the system software, the two single-chip microcomputers are programmed with different programming methods of C language to cooperate to complete a certain process flow. In the code conversion, the corresponding code is translated according to the code conversion table, and the feasibility of the translated code on the real equipment is tested.
3. CONCRETE REALIZATION

3.1. Composition and Process Requirements of Storage Unit
The automatic storage unit is used for temporary storage of parts. It is a three-dimensional warehouse structure. As shown in Figure 2, the warehouse is directly controlled by Siemens PLC, and the industrial robot can achieve different results of warehouse launching or resetting through Siemens PLC.

As shown in Figure 1, the process flow is simply described as follows: ABB robot sends the command of “pick up the max number of warehouses”, then warehouse reclaim, take out the parts with larger bin number in the three-dimensional warehouse and record the bin number.

3.2. PLC Peripheral Interface
In the peripheral equipment of Siemens PLC, the rotation and stop of the motor are used to simulate the action and recovery of the bin cylinder. The switch state of the switch is used to simulate the moving point of the bin cylinder and whether the product is in the bin. The peripheral interface of automatic storage is divided into input interface and output interface. See Table 1 and table 2 respectively.

Table 1. Input ports of PLC (including communication interface with ABB)

| Interface Name                        | Interface Definition |
|---------------------------------------|----------------------|
| Product inspection of warehouse 1     | sbit CPJC1=P1^0;     |
| Product inspection of warehouse 2     | sbit CPJC2=P1^1;     |
| Warehouse 1 launched inspection      | sbit TCJC1=P1^6;     |
| Warehouse 2 launched inspection      | sbit TCJC2=P1^7;     |
| ABB to PLC: Get command               | sbit MingLin0=P3^3;  |

Table 2. Output ports of PLC (including communication interface with ABB)

| Interface Name                  | Interface Definition |
|---------------------------------|----------------------|
| Warehouse 1 red light           | sbit HD1=P0^0;       |
| Warehouse 1 green light         | sbit LD1=P0^1;       |
| Warehouse 1 launched            | sbit M1=P2^4;        |
| PLC to ABB: Allow status        | sbit Status0=P3^2;   |
3.3. ABB Robot Peripheral Interface of Simulation System
When the industrial robot reaches the designated position, it can send a command to Siemens PLC for picking up. We can use an output IO (minglin0, see Table 4) as the picking up command notification interface; Siemens PLC calculates the required bin and sends the "allow picking" signal. We can use an input IO (status0, see Table 3) as the signal interface of "allow picking". Of course, we can also add some motors to simulate the action of the 1-6 axes of the industrial robot. The nixie tube displays the stage of the process flow to which the industrial robot is currently going. The start button is used to simulate the start switch on the teaching pendant of the industrial robot, so that the communication interface of the whole industrial robot and the relevant interfaces of peripherals are completed, as shown in Table 3 and Table 4.

Table 3. Input ports of ABB (including communication interface with PLC)
| Interface Name | Interface Definition       |
|---------------|---------------------------|
| Start Button  | sbit StartBT=P1^0;        |
| PLC to ABB:   | sbit Status0=P3^7;        |
| Allow status  |                           |

Table 4. Output ports of ABB (including communication interface with PLC)
| Interface Name | Interface Definition         |
|----------------|------------------------------|
| Sucker         | sbit GETHUB=P0^0;            |
| Servo Motor    | sbit Motor1=P0^1;            |
| (1-6) Axis     | sbit Motor6=P0^6;            |
| ABB to PLC:    | sbit MingLin0=P3^3;          |

3.4. ABB Robot Function of rGetHub
For the convenience of explanation, a program is attached, as shown in Table 5 for code Notes.

```c
#define WaitDI(status) {while(status);}  
#define WaitTime(delay) {rDelay(delay);}  
void rGetHub(){
    Set(MingLin0); // Get command
    WaitDI(Status0); // Waiting for Allow status
    // Pick up code
    MoveJ(Pos1,2); // Safety point of joint movement
    MoveL(Pos2,1); // Linear mobile positioning
    Set(GETHUB); // Draw the workpiece
    MoveL(Pos1,1); // Straight line back to safety point
    MovJ(Pos_Zero,2); // The joint moves back to zero
    Reset(MingLin0); // Pick up end
    WaitTime(delay); //delay time
}
```
Table 5. rGetHub Program notes.

| FUNCTION | Analysis | ABB FUNCTION |
|----------|----------|--------------|
| Set(MingLin0) | sbit MingLin0=P3^3; ON: Get command | Set, |
| Reset(MingLin0) | OFF: Get completed | Reset |
| Set(GETHUB) | Control sucker | Set |
| WaitDI(Status0) | sbit Status0=P3^7; PLC feedback “allow” | WaitDI |
| MoveJ, MoveL | Dynamic simulation of 1-6 axis servo | MoveJ, MoveL |
| WaitTime(delay) | delay | WaitTime |

3.5. PLC Function of p3GetHub

This is the module of the largest workpiece pick-up in automatic warehousing. For the convenience of explanation, a program is attached.

```c
void p3GetHub(){
    static unsigned char stat=0; //status
    static unsigned int delay=0; //delay
    static unsigned char num=0; //warehouse number
    if(stat==0){
        ... (1)
        M1=0;M2=0;M3=0;M4=0;M5=0;M6=0;
        delay=0;Status0=1;num=0;
    }
    if(stat==0 && (MingLin0==0)) stat=1;
    if(stat==1) {
        ... (2)
        if(!CPJC6) num=6;
        if(!CPJC5 && CPJC6) num=5;
        if(!CPJC4 && CPJC6 && CPJC5) num=4;
        if(!CPJC3 && CPJC6 && CPJC5 && CPJC4) num=3;
        if(!CPJC2 && CPJC6 && CPJC5 && CPJC4 && CPJC3) num=2;
        if(!CPJC1 && CPJC6 && CPJC5 && CPJC4 && CPJC3 && CPJC2) num=1;
        if(CPJC6 && CPJC5 && CPJC4 && CPJC3 && CPJC2 && CPJC1) num=0;
    }
    if(stat==1 && num>0) stat=2;
    if(stat==2) {
        ... (3)
        switch(num) {
            case 1: M1=1;break;
            case 2: M2=1;break;
            case 3: M3=1;break;
            case 4: M4=1;break;
            case 5: M5=1;break;
            default: M6=1;break;
        }
    }
}
```
if(stat==2) {
    if(!(TCJC1) && num==1) stat=3;
    if(!(TCJC2) && num==2) stat=3;
    if(!(TCJC3) && num==3) stat=3;
    if(!(TCJC4) && num==4) stat=3;
    if(!(TCJC5) && num==5) stat=3;
    if(!(TCJC6) && num==6) stat=3;
}
if(stat==3) Status0=0;     ……(4)
if(stat==3 && MingLin0==1) stat=4;
if(stat==4) {
    switch(num) {
        case 1: M1=0;break;
        case 2: M2=0;break;
        case 3: M3=0;break;
        case 4: M4=0;break;
        case 5: M5=0;break;
        default: M6=0;break;
    }
    if(delay<DELAY_TIME) delay++;}
if(stat==4 && delay==DELAY_TIME) stat=0;
}

This module adopts the idea of GRAFCET sequence control program:
Step 0: the initial step stat = = 0. For example, in the Program (1), first explain what events need to be done in step 0, and then how to switch to the next step when the situation occurs.
Step 1: first, compare and select the product with the largest num in the warehouse in Program (2).
Step 2: launch the product in Program (3).
Step 3: Send a pick-up notice to ABB in Program (4).
Step 4: when ABB's pick-up is completed, the empty warehouse will take back and wait for 5S in Program (5).

3.6. Simulation of PLC Code Conversion
We can transform the 89C51 simulation code of PLC into TIA portal ladder diagram through the thinking logic of programming. First of all, we can also create a new FB with background data block (Figure 3): cmd0 is the pick-up command from ABB; cp is Product in warehouse detection sensor; tj is the moving point of the pushing out cylinder; out is the action of pushing out cylinder; ret is the answer of PLC. The interface definition is consistent with p3Gethub, which is called circularly in the main program. Then, we can use the same idea of GRAFCET sequence control program to transform the ladder program of PLC (Figure 4): #cp is Product in warehouse detection sensor; #num is the warehouse number with the largest product serial number in the warehouse; #stat is the status or step.
As shown in Table 6, the basic instruction list of Siemens PLC can basically be explained in C language, that is, the C language simulation code realized by PLC logic can also be transformed into ladder logic of Siemens PLC.

Table 6. Simens PLC Instruction table and corresponding C language.

| PLC          | C language                          | Analysis                      |
|--------------|-------------------------------------|-------------------------------|
| Ld I0.0      | Q0.0=I0.0                           | Assign input to output        |
| = Q0.0       |                                     |                               |
| Ld I0.0      | Q0.0=I0.0 & I0.1;                    | AND Operation                 |
| A I0.1       | Q0.0                                |                               |
| = Q0.0       |                                     |                               |
| Ld I0.0      | Q0.0=I0.0 | I0.1;                          | OR Operation                  |
| O I0.1       | Q0.0                                |                               |
| = Q0.0       |                                     |                               |
| Ld I0.0      | if(I0.0) Q0.0=1;                     | SET Operation                 |
| S Q0.0, 1    |                                     |                               |
| Ld I0.0      | if(I0.0) Q0.0=0;                     | RESET Operation               |
| R Q0.0, 1    |                                     |                               |

3.7. Simulation of ABB Code Conversion

As shown in Table 5, we directly define the simulation function by the name and function of the robot's internal function, so that these can be easily transformed into real ABB Robot equipment through simulation code and used by real equipment.

4. SYSTEM EXPERIMENT TEST

By redefining the peripheral interface, we can download the above ABB conversion code into the teaching pendant of real industrial robot and the above PLC conversion code into the real programmable controller. After many tests of different workpiece quantities and different workpiece positions (see Table 7), the accuracy rate reaches 100%, the correct process flow is obtained, and the reclaiming efficiency of any quantity and any bin is high. The practical results show that the learning of automatic storage products through the combination of virtual simulation and practical verification is real, feasible, safe and reliable.
Table 7. maximum workpiece reclaiming test results of automatic warehousing.

| No. | WH In | WH Out | ACC (%) | T (s) |
|-----|-------|--------|---------|-------|
| 1   | 0     | 0      | 100     | 15    |
| 2   | 1     | 1      | 100     | 15    |
| 3   | 1, 2  | 2      | 100     | 15    |
| 4   | 2, 4  | 4      | 100     | 15    |
| 5   | 1, 2, 3 | 3   | 100     | 15    |
| 6   | 1, 2, 3, 5 | 5 | 100     | 16    |
| 7   | 1-6   | 6      | 100     | 16    |

WH: Warehouse; ACC: Accuracy; T: Time

5. CONCLUSIONS
In this paper, a piece of 51 single chip microcomputer is used to simulate the industrial robot in automatic storage, another one is used to simulate Siemens PLC, led, motor, button and switch are used to simulate the peripheral input and output equipment of Siemens PLC. In the system software, the two single-chip microcomputers are programmed with different programming methods of C language to cooperate to complete a certain process flow. In the code conversion, the corresponding code is translated according to the code conversion table, and the feasibility of the translated code on the real equipment is tested. The experiment shows that it is safe and reliable.

In order to better serve the students, the next focus of the author's team is to automatically convert the verified PLC simulation code into Mitsubishi PLC or Siemens PLC instruction list or ladder diagram through the third-party program.

Acknowledgments
This work was supported in part by the Program of Guangzhou Science and Technology Bureau project under Grant No.201904010121, in part by Department of education of Guangdong Province project under Grant No.2017GKTSCX011, in part by innovation team project of Guangdong Universities under Grant No.2020KCXTD048, in part by Guangdong science and technology innovation strategy special fund under Grant No. pdjh2020b0927.

References
[1] Jiguang Liu, Hao Yuan, Ze Wang, Yue Wang, “Design and implementation of virtual control platform for beer production line based on IRAI”. Experimental technology and management, 2005.6(32), pp.127-129.
[2] Hanyue Gong, Qingsheng Luo, BaoLing Han, “Research and design of virtual control platform for Tracked Robot” . Computer measurement and control, 2012.9(3), pp.694-696.
[3] Guozhu Tian, Jie Ding, “Simulation design of multi industrial robot chain production line based on robotstudio”. Modular Machine Tool & Automatic Manufacturing Technique, 2018, 12, pp.123-124.
[4] Tengfei Niu, Baoguo Yao, “Design and task simulation of automatic machining workstation for industrial robot”, Manufacturing Technology & Machine Tool, 2019(10): 44–48.
[5] Shujun Chen, Jun Zhao, Jun Xiao, “Additive manufacturing system of arc deposition robot for aluminum alloy”. Welding, 2016(4), pp.9-12.
[6] Juguang Lin, Peng Chui, Jiangqiang Wang, “Research on process planning technology of BIW automatic welding line based on robcad”. Journal of Hefei University of Technology. 2009, Vol.32 No.9, pp.1365-1368.
[7] Workspace co, Robot offline programming and simulation software, http://www.workspacelt.com//index.htm.
[8] Yawen Ou. Research on C language programming method based on industrial robot application. Light industry technology. 2019(35), pp. 96-107.
[9] V. Pinto et al. PLC Controlled Industrial Process on-line Simulator, IEEE International Symposium on Industrial Electronics 2007, ISIE 2007, 4-7 June 2007, pp 2954 - 2957.
[10] E. Estevez et al, Graphical Modelling of PLC-based Industrial Control Applications, Proceedings of the 2007 American Control Conference, New York, 11-13 July 2007, pp 220 -225.
[11] S. Shin et al, Whimori CDK: A Control Program Development Kit, International Conference on Computing, Engineering and Information, ICC’09, 2-4 April 2009, pp 115 - 118.
[12] Qinghua Wang, Yanhua Hu, Renxuan Fu, Xinzhou Ma. “Research on simulation platform of intelligent production lines”, Journal of Physics: Conference Series, 2021.