Int. j. adv. multidisc. res. stud. 2024; 4(3):163-166

Received: 18-03-2024
Accepted: 28-04-2024

International Journal of Advanced Multidisciplinary Research and Studies

ISSN: 2583-049X

Position Control for the Machine Using HMI Macro Code

[1] Le Thi Thuy Ngan, [2] Nguyen Thi Mai Huong
[1, 2] Faculty of Mechanical, Electrical and Electronic Technology, Thai Nguyen University of Technology, Vietnam

Corresponding Author: **Le Thi Thuy Ngan**

Abstract
Today, with the strong and outstanding development of science and technology, monitoring electrical and pneumatic systems is an important field in factories. It includes functions of monitoring electrical and compressed air parameters, alarming, and reporting when there is an impending or immediate incident. The integrated monitoring and control system gives you a complete overview of various tasks through an intuitive user interface. Along with the replacement of industrial equipment, smart HMI display systems have also been introduced into the stamping industry. Aiming towards an advanced electrical control method, the author has designed and implemented the function of a monitoring system based on macro code in the HMI. Combined with position control systems, specifically Servo motor control through communication methods. A position control model is designed, using WEINTEK touch screen as the display interface. The controller only uses HMI through interface design and Macro code programming to process data and make output decisions.

Keywords: Monitor System, Electrical Control, Position Control Systems, Display Interface

1. Introduction

MACRO can be understood as a program segment that supports a control command structure and it can act like a simple computer program when acted upon. The advantage of Macros is that they can easily represent how objects operate, such as: Changing data, operating conditions, and operating sequences using scripts familiar to any programming engineer. Modern industrial machines today all support popular industrial communication protocols such as MODBUS, ETHENET... to control and monitor machine status and parameters. In this article the HMI is connected to the communication pins of the controllers.

2. Structures and Features of System Control

Hardware includes: HMI WEINTEK MT8071iP, connection ports COM1, COM2, communication signal types RS232, RS485 (2W) and 1 Ethernet port. Servo Ezi Plus-R with RS485(2W) communication port. Story model from rotation to translation. There is also a 24Vdc supply for the HMI, Servo and the object's base and limit sensors. Connections from the HMI to the devices use shielded twisted copper cables.

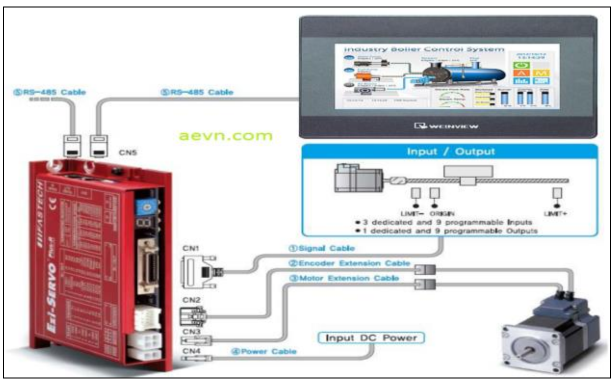


Fig 1: Block diagram of the control systems

3. Designing Communication for the Model

In this design, the communication mode for the device is as follows: Interface: Free Protocol, Data Bits 8, Stop bits 1, Baudrate 115200bps, Parity Bits None, HMI operates in Master mode, Servo(Drive) is Slave with address 1.

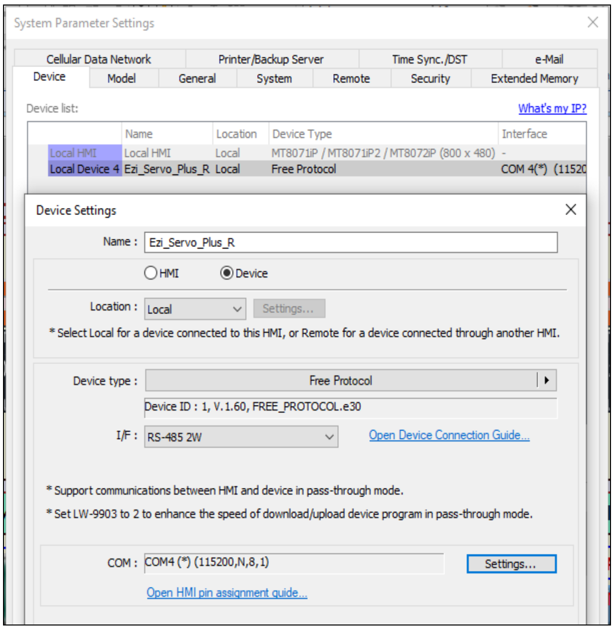


Fig 2: HMI communication settings

4. Main Hmi Design

Using EsyBuilder Pro programming software, we can develop several application functions such as status prompt, numeric setting display, real-time pop-up error alarm display, information storage, recording information, etc., to design clear and orderly operating procedures, display numerical settings, real-time centralized monitoring of parameters, and timely display of warnings and error storage [7-8]. When compiling the interface, it is necessary to fully understand the principles and functions of several related functional components, and create a series of screens with display and transition of functional components to switch between screens. this shape, as shown in Fig 3.

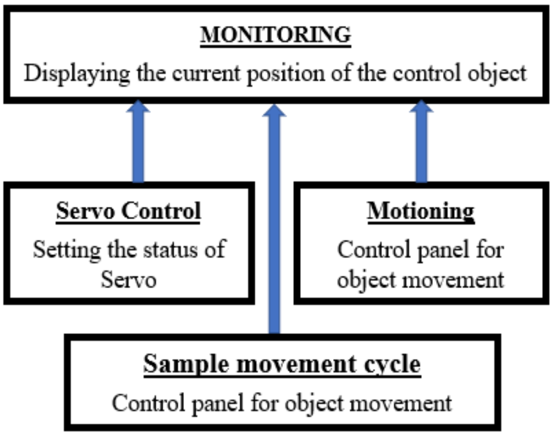


Fig 3: Relationship between screens in the HMI interface

4.1 Principles of flow chart design

All calculations and comparisons are done through Macro Code, in this design we use Background Macro mode because we have to continuously monitor parameters. This mode works like the flow chart below.

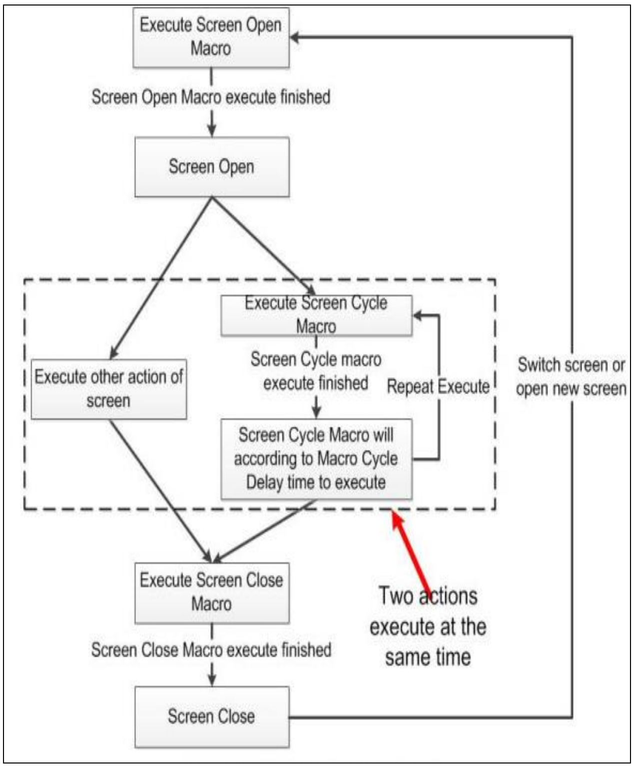


Fig 4: Screen cycle macro flow chart

4.2 Design flowchart for design

The parameters are processed as follows: HMI plays the role of Master sending transmission frames as shown below:

Header	Frame Data	Tail
0xAA 0xCC	4~252 bytes	0xAA 0xEE

1, 0xAA: Delimited byte.
2, 0xAA 0xCC: Displays that the Frame locates in header.
3, 0xAA 0xEE: Displays that the Frame locates in tail.
4, If any of the Frame data is ‘0xAA’, ‘0xAA’ should be added right after it. (add the ‘byte’ to frame data section for distinguish ‘Header’ and ‘Tail’ frame data.)
5, If any data following ‘0xAA’ is not ‘0xAA’, ‘0xCC’ or ‘0xEE’, it displays that an error has occurred.
Detailed Frame Data is configured as follows:

Slave ID	Frame type	Data	CRC	
1 byte	1 byte	0 ~ 248 bytes.	2 bytes	
			Low byte	High byte

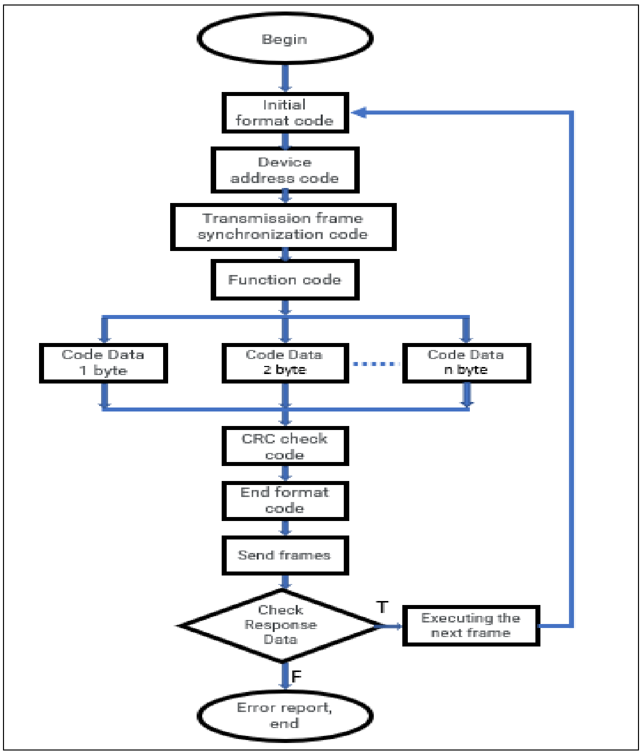
1, Slave ID: Drive module number (0~15) connected to the PC communication port.
2, Frame type: To designate command type of relevant frames. For the command type, refer to 「Frame Type and Data Configuration」 section.
3, Data: Data structure and length is set according to Frame type. For more information, refer to 「Frame Type and Data Configuration」 section.
4, CRC: To check that an error occurs during communication, ‘0xA001’ of a polynomial factor in CRC16(Cyclic Redundancy Check) is used. Or ‘X16+X15+X2+1’ of a polynomial factor in CRC-16-IBM(Cyclic Redundancy Check) is used. CRC calculation is

performed for all items (Slave ID, Frame type, Data) prior to CRC item.

The response frame from the device looks like this:

Slave ID	Frame type	Data		CRC	
1 byte	1 byte	1 byte.	0 ~ 247bytes.	2 bytes	
		Communication status	Response data	Low byte	High byte

1, Slave ID: Same to sending Frame.
(When this is not same to sending data, it should be recognized as the error status.)
2, Frame type: Same to sending Frame.
(When this is not same to sending data, it should be recognized as the error status.)
3, Data: When simple executive instructions are sent, this data cannot be read. However, in case of response, 1 byte is added to display the communication status (error / normal).
The program in Macro is shown according to the flow chart below:



4.3 Designed on HMI EsyBuilder Pro software

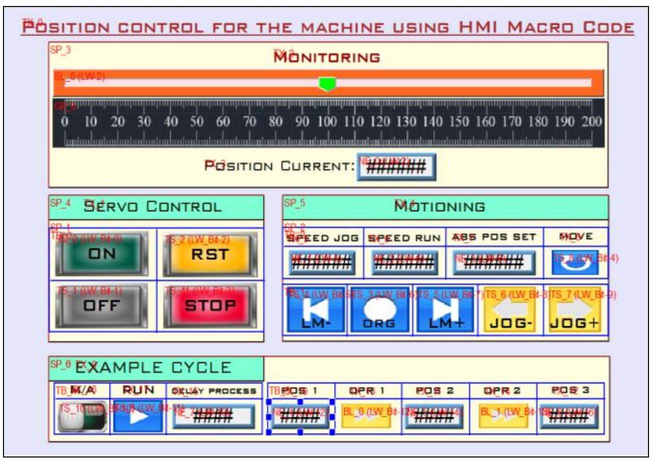


Fig 5: Design interface

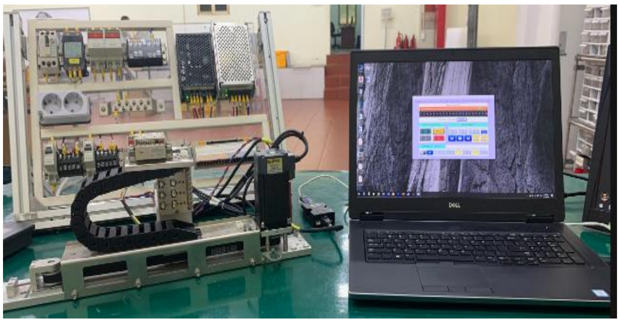
Easy Builder Pro supports drag-and-drop design and very flexible parameter settings. You will set object properties that match the engine parameters.

```
GetData(bit_JOG_A, "Local HMI", LW_Bit, 9, 1)
GetData(bit_MODE, "Local HMI", LW_Bit, 10, 1)
GetData(bit_RUN, "Local HMI", LW_Bit, 11,1)
if((bit_ON==1)or(bit_OFF==1)or(bit_RST==1)or(bit_MOVE==1)
bit_INPUT=1
//bit_INPUT_OLD=0
else
bit_INPUT=0
bit_INPUT_OLD=1
if(JOG_MODE==1)then
Syn_No=Syn_No+1
if (Syn_No==255) then
Syn_No=1
end if
Frame_CRC_3byte[0]=0x01
Frame_CRC_3byte[1]=Syn_No
Frame_CRC_3byte[2]=0x31
CRC(Frame_CRC_3byte[0],CRC_check,3)
HIBYTE(CRC_check,CRC_H)
LOBYTE(CRC_check,CRC_L)

Frame_SV_3byte[0]=0xaa
Frame_SV_3byte[1]=0xcc
Frame_SV_3byte[2]=0x01
Frame_SV_3byte[3]=Syn_No
Frame_SV_3byte[4]=0x31
Frame_SV_3byte[5]=CRC_L
Frame_SV_3byte[6]=CRC_H
Frame_SV_3byte[7]=0xaa
Frame_SV_3byte[8]=0xee
OUTPORT(Frame_SV_3byte[0], "Ezi_Servo_Plus_R", 9)
JOG_MODE=0
end if
end if

if((bit_INPUT==1)and(bit_INPUT_OLD==1))then
```

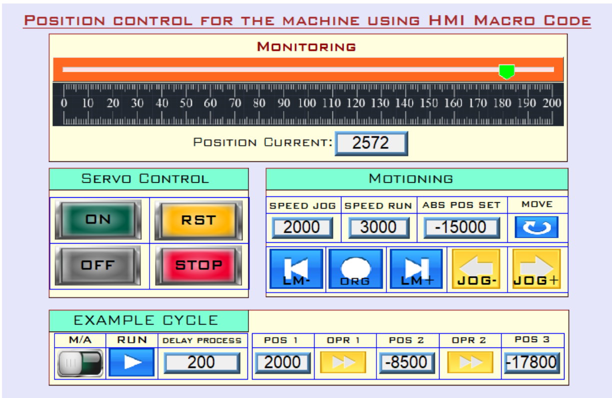
Fig 6: Macro Code writing interface

5. Achieved Results

The accuracy of the screen program is verified by the actual device installed on the model, the installation position and actual position match, the buttons function properly.



After creating the interface and writing the Macro Code, we proceed to simulate and fix errors right on the computer.



Loading the program into the HMI to directly control the position control Servo system

Fig 7: Actual operation of the model

6. Conclusion

With Macro Code, we can carry out transmission projects that are not too complicated without using another intermediate controller, bringing more economic efficiency. The above is just the Macro mode running in the background, we can fully exploit Macro Code with many other modes, more flexible in program processing.

7. Acknowledgement

This work is supported by Thai Nguyen University of Technology, Vietnam

8. References

1. Hu Haiquan. Industrial design application ergonomics [M]. 2013.
2. Wei Wei, Gong Xiaodong. Development trend of human computer interface based on user experience [J]. Journal of Beijing University of Aeronautics and Astronautics, 2011, 37 (7).
3. Liu Wei, Zhuang Damin, Liu Zhongqi. Human machine interface design [M]. 2011
4. Shi Xiaohua. Application of Fieldbus Technology in automation system [J]. Digital technology and application, 2011, 000 (005): p.130-130.
5. Chen Shiqing. Design and implementation of press electrical control display system [D]. 2016.
6. Amit Kumar Jain, Bhupesh Kumar Lad. A novel integrated tool condition monitoring system [J]. Journal of Intelligent Manufacturing, 2019, 30.
7. Sun yening. Design of automatic assembly line training system based on Siemens PLC [D]. 2018.
8. Liao Changchu. S7-300/400 PLC application technology [M]. 2012.
9. Wu Mingliang, fan Minglong. Automatic production line technology [M]. 2011.
10. Lin Jing. Development of fault diagnosis system for mechanical press [D]. Guangdong University of technology, 2013.
11. Jiang Ligang, Zhang Chengxiang. Modern equipment management, fault diagnosis and maintenance technology [M]. Harbin Engineering University Press, 2010.