

# **FESTO Project 2**

## **Mechatronics 282 778**

Marc Alexander Sferrazza \*<sup>†</sup>

March 30, 2017

# **FESTO**

### **Abstract**

A brief report on Festo PLC programming and the process for four module stations. Each station was tested and debugged to ensure a robust result with exception of station 3 which has been coded in theory but not tested as the station is disabled.

---

\*This work was not supported by any organization

<sup>†</sup>Faculty of Mechatronics Engineering, Massey University, Albany, Auckland, New Zealand Progress of project:  
<https://github.com/alex1v1a/Mechatronics/tree/master/Project%202>

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>2</b>	<b>METHOD</b>	<b>3</b>
2.1	Programming the PLC . . . . .	3
2.2	Algorithm . . . . .	3
2.3	Station One . . . . .	3
2.4	Station two . . . . .	3
2.5	Station three . . . . .	3
2.6	Station four . . . . .	3
<b>3</b>	<b>RESULTS</b>	<b>3</b>
3.1	Debugging . . . . .	3
3.2	Testing . . . . .	4
3.3	Finalising . . . . .	4
<b>4</b>	<b>CONCLUSIONS</b>	<b>4</b>

# 1 INTRODUCTION

To design the manufacturing based task of the Festo MPS (modular production system) stations using PLC (programmable logic controller) programming. When considering control of the modular stations a series of sensors and actuators of mechanical, pneumatic and electrical are revised. There are 4 stations of which each contain specific sorting tasks that utilise the range of many interconnected relays to perform sequential logical instructions in loops stored locally on the device. Utilising these gates instructions will be given to move an array of pucks from point A to point B while simultaneously sorting them in some cases.

## 2 METHOD

Template files given for the allocation list, default Stop and Emergency programs are used with the project to replicate a given instruction produced in the provided videos for each module station.

### 2.1 Programming the PLC

Using the statement method an instruction has been written to control the PLC system. There is also another method using a graphical map of ladder logic; however due to the simplicity of the STEP function the statement method has been chosen for easy to follow logical steps.

### 2.2 Algorithm

For every station a set of initialisation commands are given for when the user selects input, start, stop or reset while manual switch is on. The start operation sets the station to initial position then performs the sequence tasks as instructed by steps. The stop operation immediately seizes all functions and brings the system to a halt, while the reset button will return the all functions to initial positions and reset the entire system at any time.

### 2.3 Station One

### 2.4 Station two

### 2.5 Station three

### 2.6 Station four

## 3 RESULTS

Completing tasks for programming each of the stations and linking them together in a series to provide an efficient step by step system as displayed per each demonstration video. Each task was completed and checked for errors in runtime with physical testing for a robust program outside of the videos demonstration. Stages on what to do after the tasks have been completed, before the task starts, what to do in result of a malfunction and linking the stations together have all been completed in the steps below. The entire programming task for all stations took 6-8 solid hours to learn and write, which felt significantly less due to the deep focus at the time. This was not that demanding in time but did require some thought effort to realise what stages things can work, but after the initial logic was realised, the rest of the program was simple.

### 3.1 Debugging

While the process is straight forward and consists of step by step operations, the procedure in which to get there is not. While in a program a system may appear sound, there are many underlying issues, that of which if are not addressed will cause serious malfunction down the line.

It is imperative that all processes are enabled and disabled in a suitable fashion such that the system is reliant on the next steps status. This means that while it may be simple to just use a timer and guess the time intervals between stages, after the 10th or 100th or 1000th step the time interval may have adjusted out of sync with the rest of the sequence and thus cause a jam, or other malfunction in which is catastrophic for the system. To avoid this simple stages based on sensors are used in steps, and timers may be used in these steps as long as such use is consistent with the stages in the loop and will not potentially become out of phase with the system.

## 3.2 Testing

It is always important to test the system for any glitches while debugging. There is always room for improvement in making the system more efficient and effective from start to finish. This includes ensuring that the system knows what to do in the event that the task at hand is complete, or even after this stage when more tasks or jobs are added to the system.

The Festo program also offers live debugging with current status of steps with the online function. While I did not find this particularly useful as I did not need to use it; I did however explore this procedure which displays live feedback for the status of sensors throughout the steps of the system.

## 3.3 Finalising

Making the system stable and concise with as few steps, and as robust as possible builds for a good design. The method of testing all variables that might effect the system is required to make this achievable.

# 4 CONCLUSIONS

The procedure of breaking a system down into steps and stages to build up a process chain of stable instructions is relatively straight forward, and easy enough of language to understand for the next person to come along and recognise what is happening. It is very well structured and being able to debug a complex set of sequences and tweak with very little difficulty. The immediate advantages of the direct response to sensors and actuation makes for a great way to preform any set of tasks by instigating a signals high or low outputs to preform a task in correct order with very low room for fault rate.

PLC is an easy to learn language with great potential and versatility in a range of manufacturing lines, and is used in many industries today. While expensive to prototype, once in place the system is robust and efficient saving costs in the long term if maintained and set up correctly.

## References

- [1] J. Engel, J. Chen, and C. Liu, “Development of polyimide flexible tactile sensor skin,” *Journal of Micromechanics and Microengineering*, vol. 13, no. 3, p. 359, 2003.
- [2] J. Lötters, W. Olthuis, P. Veltink, and P. Bergveld, “The mechanical properties of the rubber elastic polymer polydimethylsiloxane for sensor applications,” *Journal of Micromechanics and Microengineering*, vol. 7, no. 3, p. 145, 1997.
- [3] S. Omata and Y. Terunuma, “New tactile sensor like the human hand and its applications,” *Sensors and Actuators A: Physical*, vol. 35, no. 1, pp. 9–15, 1992.
- [4] C. Larson, B. Peele, S. Li, S. Robinson, M. Totaro, L. Beccai, B. Mazzolai, and R. Shepherd, “Highly stretchable electroluminescent skin for optical signaling and tactile sensing,” *Science*, vol. 351, no. 6277, pp. 1071–1074, 2016.
- [5] J. Engel, J. Chen, Z. Fan, and C. Liu, “Polymer micromachined multimodal tactile sensors,” *Sensors and Actuators A: physical*, vol. 117, no. 1, pp. 50–61, 2005.
- [6] U. Paschen, M. Leineweber, J. Amelung, M. Schmidt, and G. Zimmer, “A novel tactile sensor system for heavy-load applications based on an integrated capacitive pressure sensor,” *Sensors and Actuators A: Physical*, vol. 68, no. 1, pp. 294–298, 1998.
- [7] Y. Hasegawa, M. Shikida, D. Ogura, Y. Suzuki, and K. Sato, “Fabrication of a wearable fabric tactile sensor produced by artificial hollow fiber,” *Journal of micromechanics and microengineering*, vol. 18, no. 8, p. 085014, 2008.
- [8] K. Horch, S. Meek, T. G. Taylor, and D. T. Hutchinson, “Object discrimination with an artificial hand using electrical stimulation of peripheral tactile and proprioceptive pathways with intrafascicular electrodes,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 5, pp. 483–489, 2011.
- [9] L. R. Huang H, Kuiken TA, “A strategy for identifying locomotion modes using surface electromyography,” *IEEE Trans Biomed Eng*, vol. 1, no. 56, pp. 65–73, 2009.
- [10] S. P. Sober SJ, “Flexible strategies for sensory integration during motor planning,” *Nature Neuroscience*, vol. 4, no. 8, pp. 490–497, 2005.
- [11] Z. W. J. J. R. M. et a, “Effects of carpal tunnel syndrome on adaptation of multi-digit forces to object weight for whole-hand manipulation,” *PloS One*, vol. 6, no. 11, 2011.
- [12] J. R. Jenmalm P, “Visual and somatosensory information about object shape control manipulative fingertip forces,” *Journal of Neuroscience*, vol. 17, no. 11, pp. 4486–4499, 1997.
- [13] ainburg RL Ghilardi MF Poizner H Ghez C, “Control of limb dynamics in normal subjects and patients without proprioception,” *Journal of Neurophysiology*, vol. 2, no. 73, pp. 820–835, 1995.
- [14] F. H. M. W. K. A. et al, “A neural substrate for nonpainful phantom limb phenomena,” *Neuroreport*, vol. 7, no. 11, pp. 1407–1411, 2000.
- [15] H. J. O. J. M. M. D. M. M. P. K. TA and C. KM, “Novel targeted sensory reinnervation technique to restore functional hand sensation after transhumeral amputation,” *IEEE Trans Neural Syst Rehabil Eng*, vol. 4, no. 22, pp. 765–773, 2014.

APPENDIX

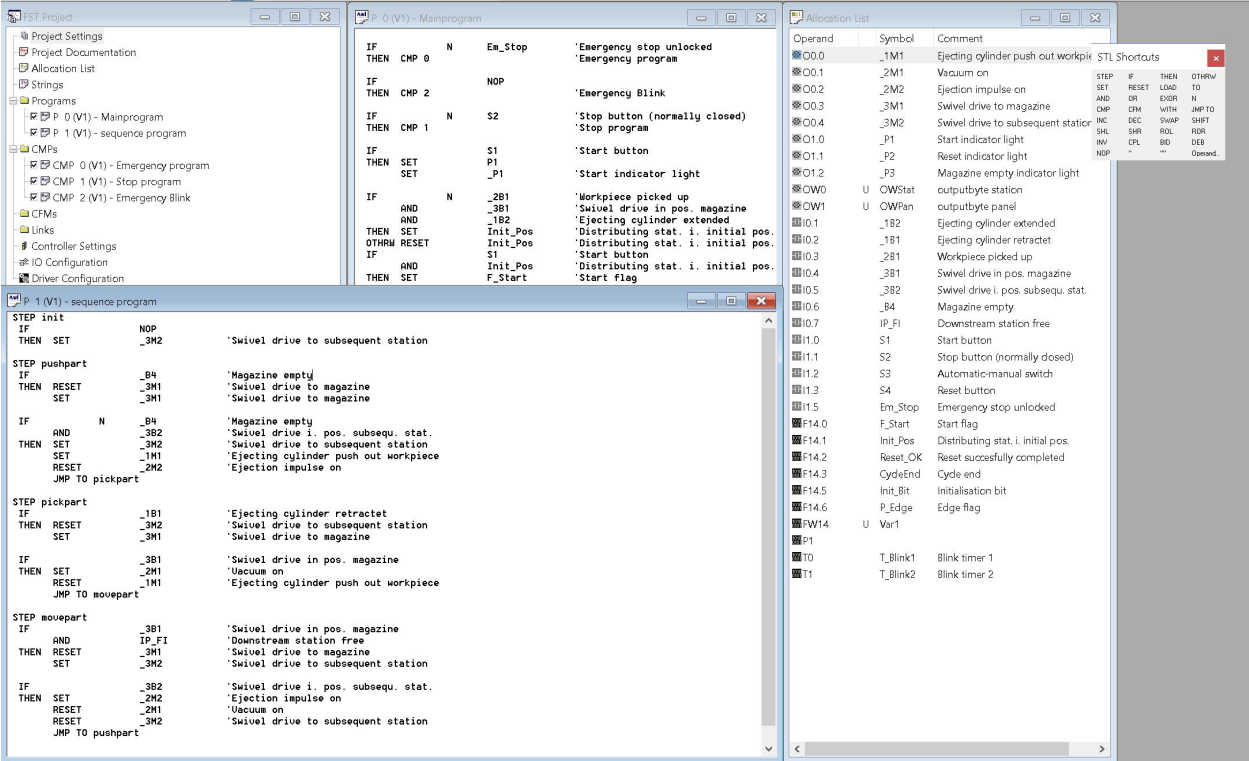


Figure 1: Station 1

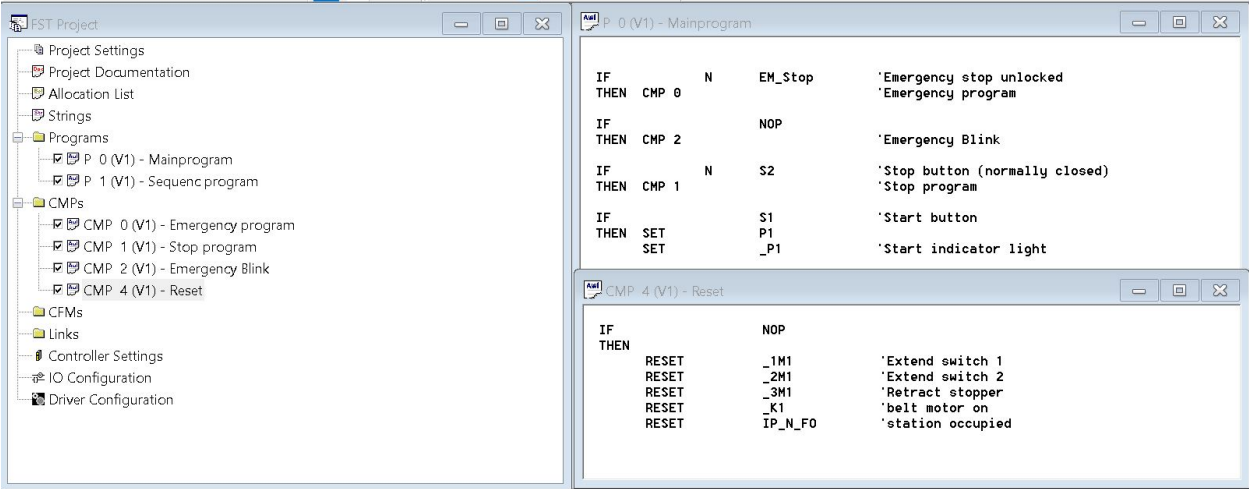


Figure 2: Station 2

P 1 (V1) - Sequenc program

```
STEP init
IF
THEN SET      NOP
      Init_Pos 'Sorting station in initial position
      JMP TO detect

STEP detect
IF
THEN SET      Part_AV 'Part available
      SET      _K1      'belt motor on
      SET      WITH     Timer1
                        1s

STEP wait
IF
THEN RESET    N      Timer1|
      JMP TO  _K1      'belt motor on
      color

STEP color
IF
AND N B3      'Workpiece not black
THEN SET      B2      'Metallic workpiece
      SET      _3M1     'Retract stopper
      SET      _K1      'belt motor on
      SET      _1M1     'Extend switch 1
      RESET    Timer1
      SET      WITH     Timer1
                        0.4s
      JMP TO allowpass

IF
AND N B3      'Workpiece not black
AND N B2      'Metallic workpiece
THEN SET      _3M1     'Retract stopper
      SET      _K1      'belt motor on
      RESET    Timer1
      SET      WITH     Timer1
                        0.4s
      JMP TO allowpass

IF
THEN SET      B2      'Metallic workpiece
      SET      _3M1     'Retract stopper
      SET      _K1      'belt motor on
      SET      _2M1     'Extend switch 2
      RESET    Timer1
      SET      WITH     Timer1
                        0.4s
      JMP TO allowpass

STEP allowpass
IF
THEN RESET    N      Timer1
      RESET    _3M1    'Retract stopper
      SET      WITH     Timer1
                        3s

STEP retract
IF
THEN RESET    N      Timer1
      RESET    _1M1    'Extend switch 1
      RESET    _2M1    'Extend switch 2
      RESET    _K1      'belt motor on
      JMP TO check

STEP check
IF
THEN RESET    B4      'Slide full
      _K1      'belt motor on
OTHRW
      JMP TO detect
```

Figure 3: Station 2 Sequence

Project Settings

Project Documentation

Allocation List

Strings

Programs

P 0 (V1) - Mainprogram

P 1 (V1) - Sequence Program

CMPIs

CMP 0 (V1) - Emergency program

CMP 1 (V1) - Stop program

CMP 2 (V1) - Emergency Blink

CEMs

P 0 (V1) - Mainprogram

```
IF THEN CMP 0 N Em_Stop 'Emergency stop unlocked
      NOP 'Emergency program
IF THEN CMP 2 N S2 'Stop button (normally closed)
      'Stop program
IF THEN CMP 1 N S1 'Start button
      SET P1
      SET _P1 'Start indicator light
      RESET _1M1 'Handling to upstream station
      RESET _1M2 'Handling to downstream station
      RESET _2M1 'Extend gripper
      RESET _3M1 'Open gripper
```

P 1 (V1) - Sequence Program

```
STEP init
IF THEN SET NOP
      Init_Pos 'Handling to upstream station
      'PickAlfa station in initial position
      JMP TO goto

STEP goto
IF THEN RESET _1B1 'Handling at upstream station
      SET _1M1 'Handling to upstream station
      SET _3M1 'Open gripper
      SET _2M1 'Extend gripper
      JMP TO pickup

STEP pickup
IF THEN RESET _2B1 'Gripper extended
      SET _3M1 'Open gripper
      SET _2M1 'Extend gripper
      JMP TO move

STEP move
IF THEN SET _2B2 'Gripper retracted
      SET _1M2 'Handling to downstream station
      JMP TO color

STEP color
IF AND N _1B3 'Handling at sorting position
      _3B1 'Workpiece is not black
THEN RESET _1M2 'Handling to downstream station
      SET _2M1 'Extend gripper
      SET _2B1 'Gripper extended
      SET _3M1 'Open gripper
      SET _2M1 'Extend gripper
      OTHRW JMP TO start

STEP red
IF THEN RESET _1B2 'Handling at downstream station
      SET _1M2 'Handling to downstream station
      SET _2M1 'Extend gripper
      SET _2B1 'Gripper extended
      SET _3M1 'Open gripper
      SET _2M1 'Extend gripper
      JMP TO start

STEP start
IF THEN SET _2B2 'Gripper retracted
      _1M1 'Handling to upstream station
      JMP TO goto
```

Allocation List

Operand	Symbol	Comment
00.0	_1M1	Handling to upstream station
00.1	_1M2	Handling to downstream station
00.2	_2M1	Extend gripper
00.3	_3M1	Open gripper
00.7	_3M1	station occupied
01.0	_P1	Start indicator light
01.1	_P2	Reset indicator light
0W0	OWStat	outputbyte station
0W1	OWPan	outputbyte panel
10.0	Part_AV	Workpiece available
10.1	_1B1	Handling at upstream station
10.2	_1B2	Handling at downstream station
10.3	_1B3	Handling at sorting position
10.4	_2B1	Gripper extended
10.5	_2B2	Gripper retracted
10.6	_3B1	Workpiece is not black
10.7	IP_FI	Downstream station free
11.0	S1	Start button
11.1	S2	Stop button (normally closed)
11.2	S3	Automatic-manual switch
11.3	S4	Reset button
11.5	Em_Stop	Emergency stop unlocked
F44.0	F_Start	Start flag
F44.1	Init_Pos	PickAlfa station in initial position
F44.2	Reset_OK	Reset successfully completed
F44.3	CycleEnd	Cycle end
F44.5	Init_Bit	Initialisation bit
F44.6	delay1	Flag delay time 1 expired
F44.7	F_Mat	part is not black
F44.8	P_Edge	Edge flag
FW44	U Var1	
P1		
F1		
T0		
T1	T_Blink1	Blink timer 1
T2	T_Blink2	Blink timer 2

STL Shortcuts

STEP	IF	THEN	OTHRW
SET	RESET	LOAD	TO
AND	OR	EXOR	N
CMP	CM	WITH	ABSTO
INC	DEC	SWAP	SHIFT
SHL	SHR	ROL	ROR
INV	CPL	B0	DEB
NOP	-	-	Overand.

Figure 4: Station 3

EST Project

Project Settings

Project Documentation

Allocation List

Strings

Programs

CMPS

CFMs

Links

Controller Settings

IO Configuration

P 0 (V1) - Mainprogram

```
IF THEN CMP 0 N Em_Stop 'Emergency stop unlocked
'Emergency program

IF THEN CMP 2 NOP 'Emergency Blink

IF THEN CMP 1 N S2 'Stop button (normally closed)
'Stop program

IF THEN CMP 3 S4 'Reset button
'Reset

IF THEN SET S1 'Start button
SET P1
SET _P1 'Start indicator light
```

P 1 (V1) - Sequence program

```
STEP init
IF THEN SET NOP Init_Pos 'Testing station in initial position

STEP part
IF THEN SET Part_AV 'Workpiece available
RESET IP_N_FO 'station occupied
SET _1M1 'Lower lifting cylinder
WITH Timer1 3s

STEP raise
IF THEN SET N Timer1 'Raise lifting cylinder
SET _1M2

STEP push
IF THEN SET _1B1 'Lifting cylinder raised
SET _3M1 'air slide on
SET _2M1 'Extend ejecting cylinder
WITH Timer1 2s

STEP wait
IF THEN RESET N Timer1 'Extend ejecting cylinder
RESET _2M1 'air slide on
RESET _3M1

STEP lower
IF THEN RESET _2B1 'Ejecting cylinder retracted
SET _1M2 'Raise lifting cylinder
RESET _1M1 'Lower lifting cylinder
RESET IP_N_FO 'station occupied
JMP TO part
```

Allocation List

Operand	Symbol	Comment
Q0.0	_1M1	Lower lifting cylinder
Q0.1	_1M2	Raise lifting cylinder
Q0.2	_2M1	Extend ejecting cylinder
Q0.3	_3M1	air slide on
Q0.7	IP_N_FO	station occupied
O1.0	_P1	Start indicator light
O1.1	_P2	Reset indicator light
O1.3	_P3	Ind. light mat. 1: 0=bik 1=rd 1=si
OW0	U	OWStat
OW1	U	OWPan
I0.0	Part_AV	Workpiece available
I0.1	B2	not black workpiece
I0.2	B4	Safety light barrier
I0.3	B5	Workpiece height correct
I0.4	_1B1	Lifting cylinder raised
I0.5	_1B2	Lifting cylinder lowered
I0.6	_2B1	Ejecting cylinder retracted
I0.7	IP_FI	Downstream station free
I1.0	S1	Start button
I1.1	S2	Stop button (normally dosed)
I1.2	S3	Automatic-manual switch
I1.3	S4	Reset button
I1.5	Em_Stop	Emergency stop unlocked
F24.0	F_Start	Start flag
F24.1	Init_Pos	Testing station in initial position
F24.2	Reset_OK	Reset finished
F24.3	CycleEnd	Cycle end
F24.5	Init_Bit	Initialisation bit
F24.6	delay1	Flag delay time=1 second
F24.7	P_Edge	STL Shortcuts
FW24	U	Vsr1
P1		
T0	Timer1	
T1	T_Blink1	Blink time
T2	T_Blink2	Blink time

CMP 3 (V1) - Reset

```
IF THEN NOP

RESET _1M2 'Raise lifting cylinder
RESET _1M1 'Lower lifting cylinder
SET _1M1 'Lower lifting cylinder

RESET _3M1 'air slide on
RESET _2M1 'Extend ejecting cylinder
RESET IP_N_FO 'station occupied
```

Figure 5: Station 4