FESTO Project 2
Mechatronics 282 778

Marc Alexander Sferrazza
12164165 *†

April 12, 2017

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

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

Contents

1 INTRODUCTION

To design the manufacturing based tasked 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 preform 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. A full break down of the process has been attached and can be found in the code in the appendix.

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 preforms 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

Station one's task is to detect the puck in the magazine, then shift it with the double ejecting cylinder and reload. After the puck has been shifted along the magazine tray, the swivel arm is instructed to then pick up the puck. Once the swivel arm has reached placement of the puck in the magazine position the swivel arm must wait for the station two to give the signal that the puck is ready to be received. When the signal is given from station two that the drop point is clear and ready to receive the next puck, the swivel arm will then move the puck over.

After this process is complete, it will repeat indefinitely until there are no pucks left, where in that case the swivel arm will rest in the magazine position after the last drop to avoid collision with station two. If after the magazine has been emptied and the swivel arm is at rest another puck is placed in the magazine, the entire process will restart from the initial position thus indefinitely continuing the process.

2.4 Station two

Station two will receive the puck from station one via being handed over by the swivel arm. The station uses a timer to wait for the swivel arm to react after dropping the puck and clear. The puck is then raised and deployed via an ejection cylinder onto an air slider platform where it continues to the next phase of the process.

The carrier is then lowered and signal sent to station one that the cartridge is clear and ready to receive the next puck. In the video it is not shown wether sorting occurs at this stage however there is a light sensor to detect the puck colour either black or red so there is an option of sorting at his point before the puck is raised.

2.5 Station three

Station three will receive a signal that the puck is ready to be picked up where a sliding arm will be shifted to the puck and lowered then clamped and raised. After picking the puck up the slider will then move along with the puck past a light colour sensor which also detects if the part is black or red. The part will stop and lower to track one if detected black, or continue and lower to be deployed to track two if the puck is red.

After completion the process is reset and awaiting next signal from station two. The first step of the program when start button pressed is to reset all to initial positions and be ready for the continuous loop.

2.6 Station four

Station four in theory will receive the pucks from station three, in any given colour where a light sensor and metallic sensor sort the pucks. The conveyer belt is activated and a moving arm placed out to guide the pucks to their designated areas, while each puck release from an automatic stopper arm to sort each puck individually.

Once the magazine is full for any particular puck cartridge the system will disarm until the cartridge has been emptied and the start button pressed.

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 cain 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] Festo, “Automating with fst,” *Federal Republic of Germany*, vol. 682 300, no. en 0402NH, 2004.

APPENDIX

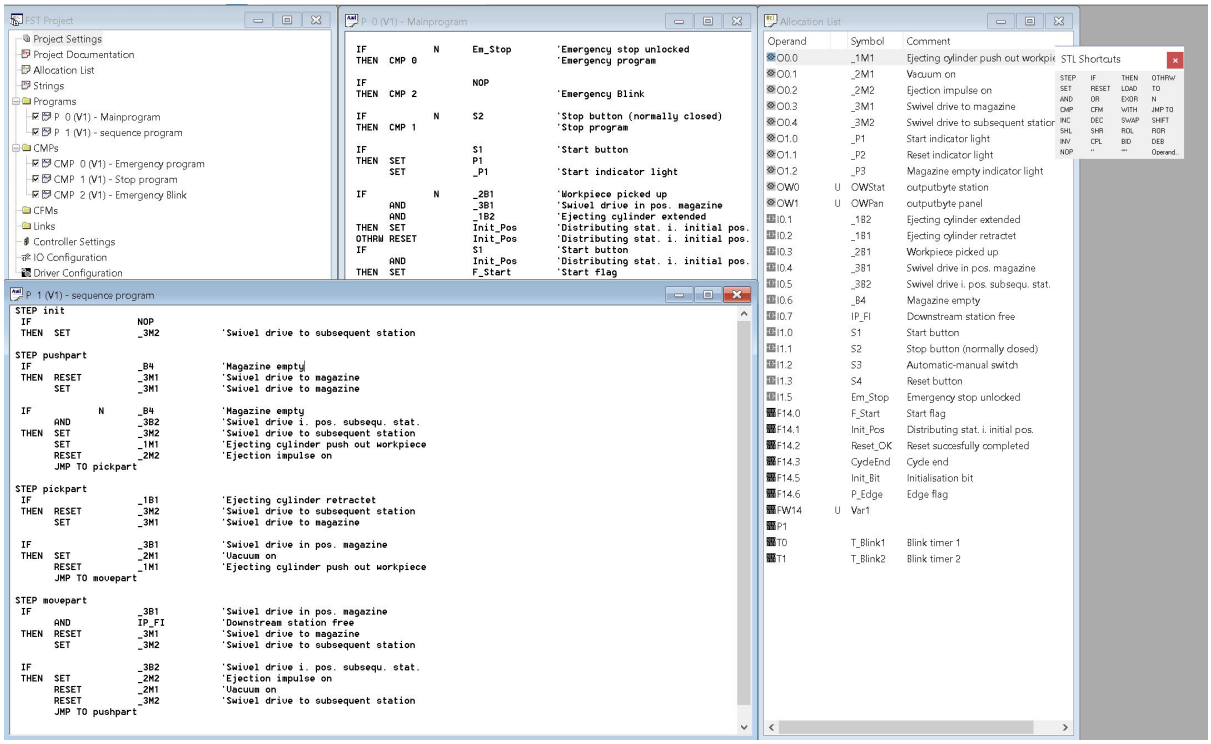

Figure 1: Station 1

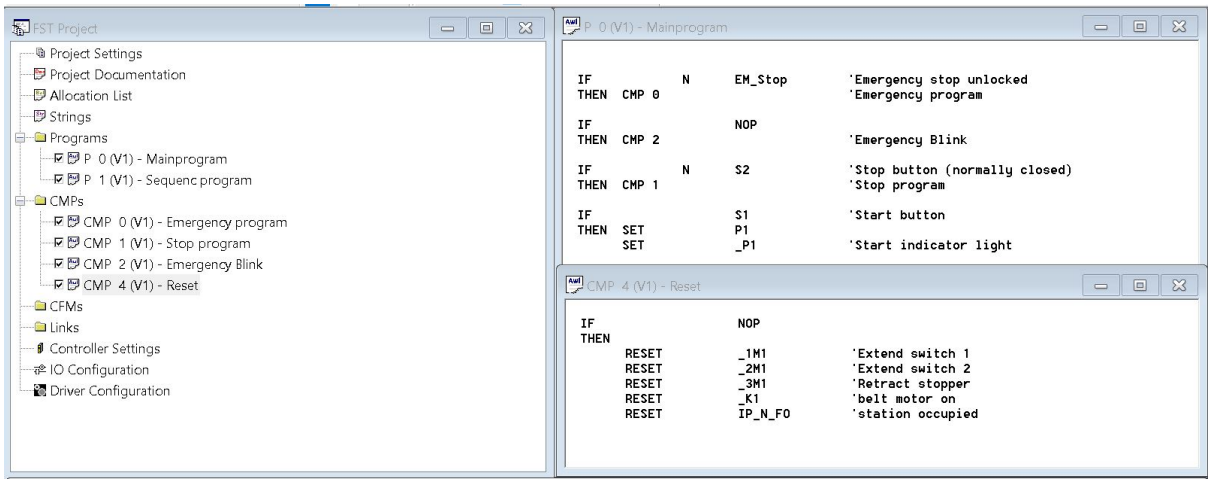

Figure 2: Station 2

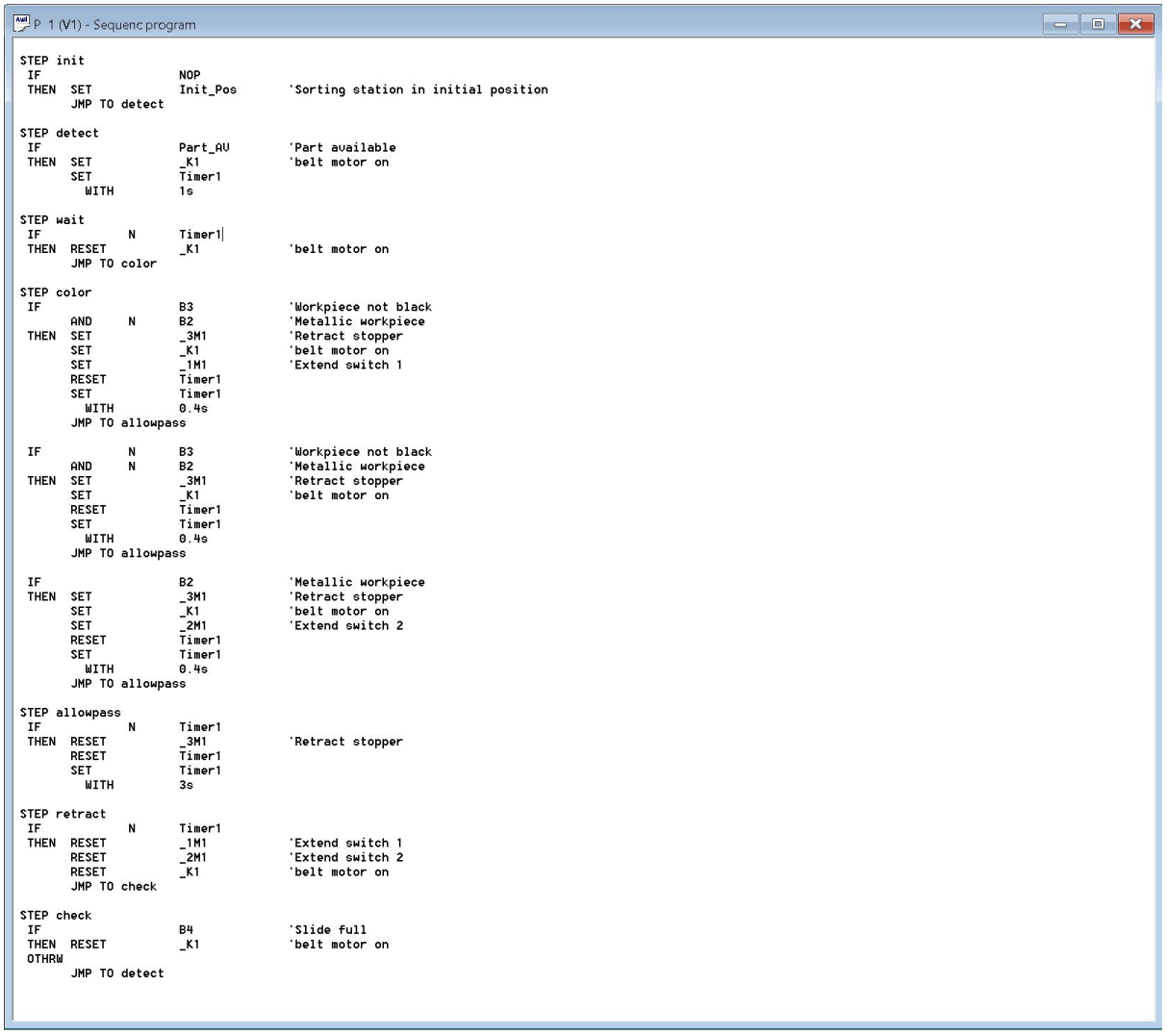

Figure 3: Station 2 Sequence

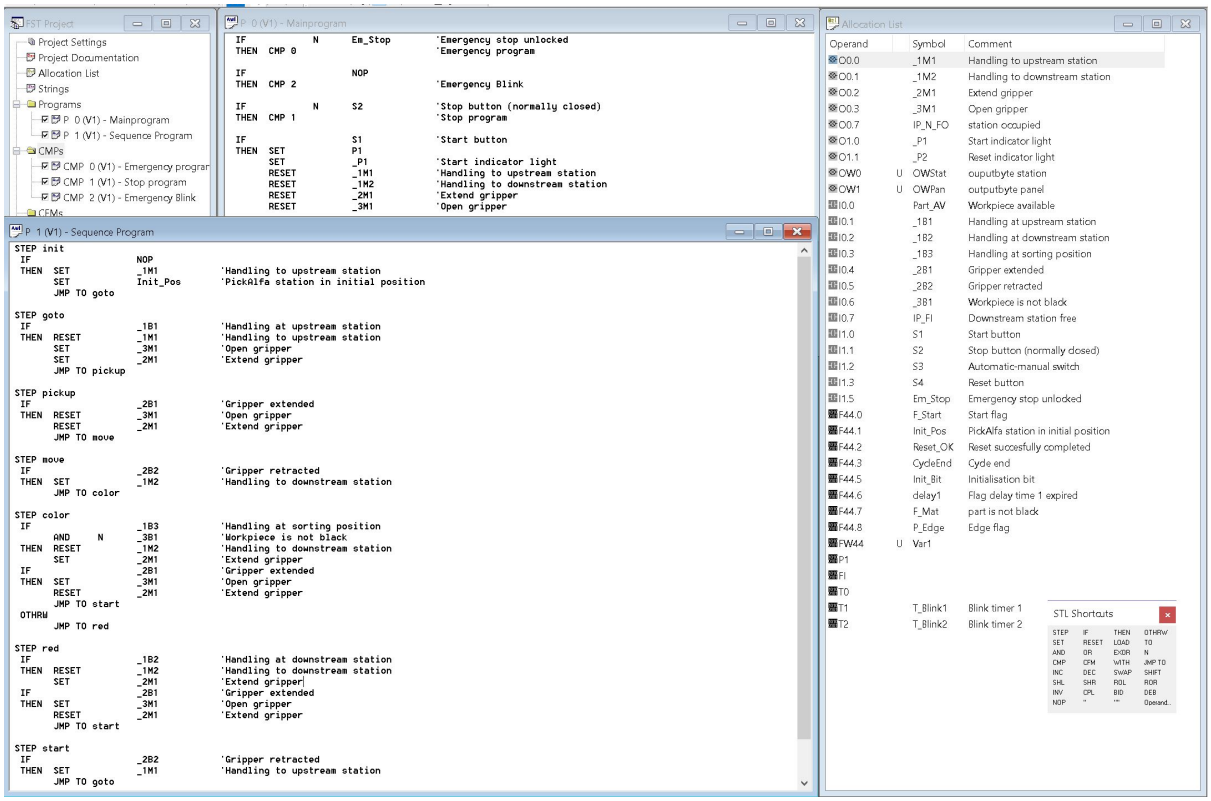

Figure 4: Station 3

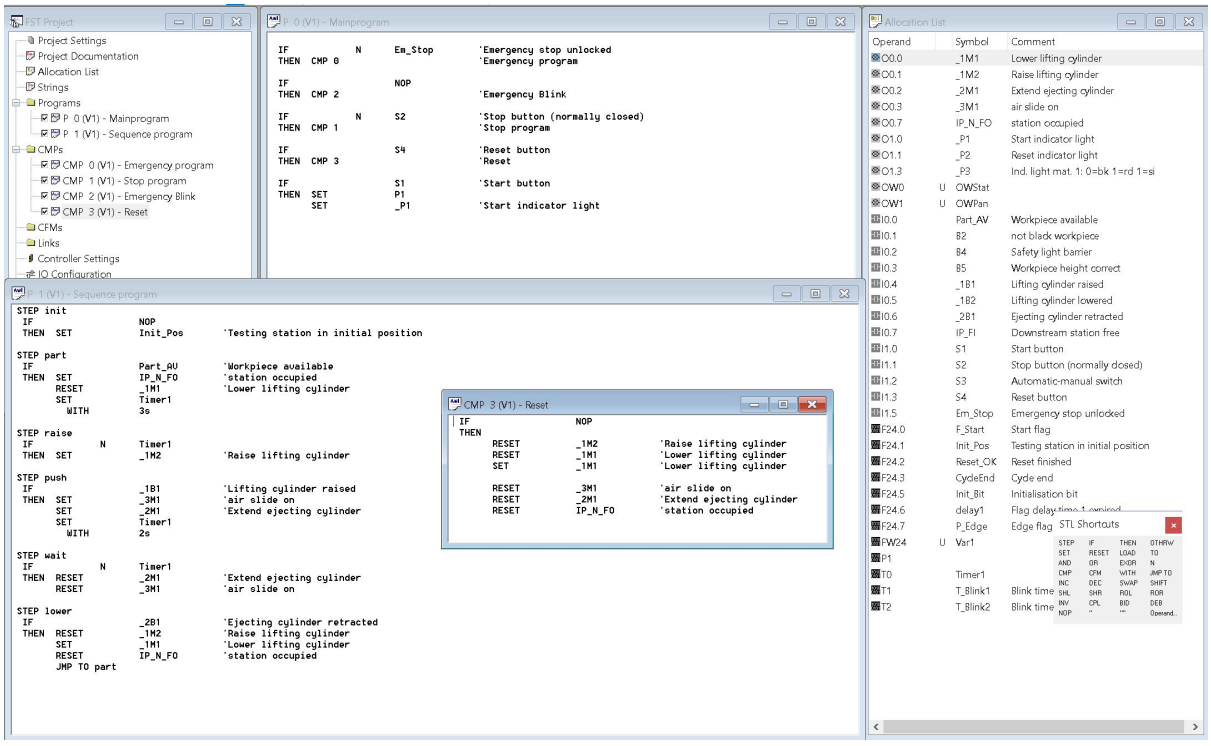

Figure 5: Station 4