MOTION CONTROL SYSTEM WITH PRACTICE-BASED IMPLEMENTATION

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor’s thesis

Valkeakoski
Electrical and Automation Engineering

Spring 2021

Quoc Tien Thang Nguyen

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Electrical and Automation Engineering
Valkeakoski

Author Quoc Tien Thang Nguyen **Year** 2019

Subject Motion control system with practice-based implementation

Supervisor(s) Juha Sarkula

ABSTRACT

The purpose of this thesis was to demonstrate motion control system both generally and in detail with a model as an illustration. The thesis was commissioned by Autotech Machinery JSC. The main goal of the thesis was to gain a deep understanding about motion control theoretically and to apply that knowledge into electrical designing and programming for an actual machine as a practical project.

Motion control is used with motors which include a servo motor, a stepper motor and a linear motor, to provide mechanism movements in a controlled manner. In order to make that happen, drivers and controllers are needed. All of the above are mentioned in the theory part of the thesis. At the same time, a machine called PF Appearance Checking Unit, commissioned by Autotech Machinery JSC, which has a stepper and a servo motor, a cylinder and many other electrical components is the field project to illustrate what I had learnt theoretically and how I applied those to work on a project. The servo motor moves the camera to a predetermined position, then the stepper motor rotates the cylinder-stabled products for the vision system to check the quality. When the quality checking is done, the cylinder moves backward for the products to be removed. It is the result of a project the company did for a customer. I was in charge of selecting electrical components, electrical drawing, wiring, and programming code. The PF Appearance Checking Unit's purpose is to represent the main unit which is a quality checking unit of a fully automated machine which will be built in the future where there will be a supply unit, a quality checking unit and a finishing unit.

The goal of the thesis was to extend the knowledge of motors and motion control both theoretically and practically. Web-based research, literature and research papers were used for the theoretical part of the thesis. For the practical part, the documents on the programming code of the machine which are attached as an appendix indicate the method, the process and the result of my field work.

Keywords Motor, motion control system, stator, rotor, electromagnetism

Pages 78 pages including appendices 41 pages

CONTENTS

1 INTRODUCTION

There is no denial that motion control plays a vital role in every kind of industry. It has become the most essential aspect of modern industrial machinery design. Motion control system presents itself clearly in every factory as an indispensable tool to control mechanism movements of conveyors, robots, etc. The basic idea behind motion control is the actuators cause acceleration to reach the desired destination but there is no guarantee that the joint of the robot or the conveyor belt stops at the destination without any incidents or worst go beyond that. In order to make the machine operate properly and precisely to achieve specific requirements, motion control need to become motion control system. It means that feedback devices are needed to complete the combination of controller, driver and motor. In addition, motion applications require various engineering skills to entirely understand the mechanical and control aspect which resulting in the significant of the background knowledge about motion control systems and their mechanical specifications.

The practical part contains the project, the machine, that I designed for my company. The machine contains a servo and a stepper motor, a PLC, an HMI, a cylinder and some other electrical components. The project was a good opportunity to practice choosing and designing components for an electric box, being familiar with motors and lastly programming code.

The research question was to obtain basic knowledge about the motion control system and its components, alongside with designing the quality checking machine electrically and programming that machine. Through the theoretical sections of the thesis, one is able to obtain more knowledge about motion control system's components and the system as a whole. The thesis would enhance the experience to not only that person's first approach to the topic but also the better preparation when one starts working in factories or companies. The practical application consists of brief background information about the concept and operation of the machine, the software and the hardware used, the process of designing the machine and to conclude, the programming stage. The pictures of the PLC code and HMI screens are showed as Appendix 1.

2 MOTOR

2.1 Introduction to motors

An electric motor is a device which performs converting electrical energy into mechanical energy. The principle behind an electric motor is electromagnetism which is the process where a magnetic field is created by introducing the current in the conductor. Applying that to an electric motor, it generates magnetic field by providing electric current to a coil, the magnetic field causes a force with a magnet then produces movement. (Ducksters, n.d.)

There are two kinds of motors, DC motors and AC motors. Even though they serve the same purpose of a motor, it is obvious that they are different not only running by different power source but also by their design. However, for familiarity purposes, manufacturers classify their products according to the power source as well as the functionalities of the motors. Both classifications are explained in detail in chapters 2.3, 2.4 and 2.5.

2.2 History

Even though being presented by Joseph Henry and Michael Faraday as a motion device using electromagnetic fields, in 1834, the first real electric motor shown in Figure 1 was developed by Thomas Davenport. “Real” emphasizes that the motor was capable of doing an actual task. Despite the fact that these devices could not help mankind at that time anything in particular, they set a great foundation for future generations to develop them into such essential devices. It was Davenport’s motor which turned out to be the first important application of electric power, not light bulb. (Edison Tech Center, n.d.)

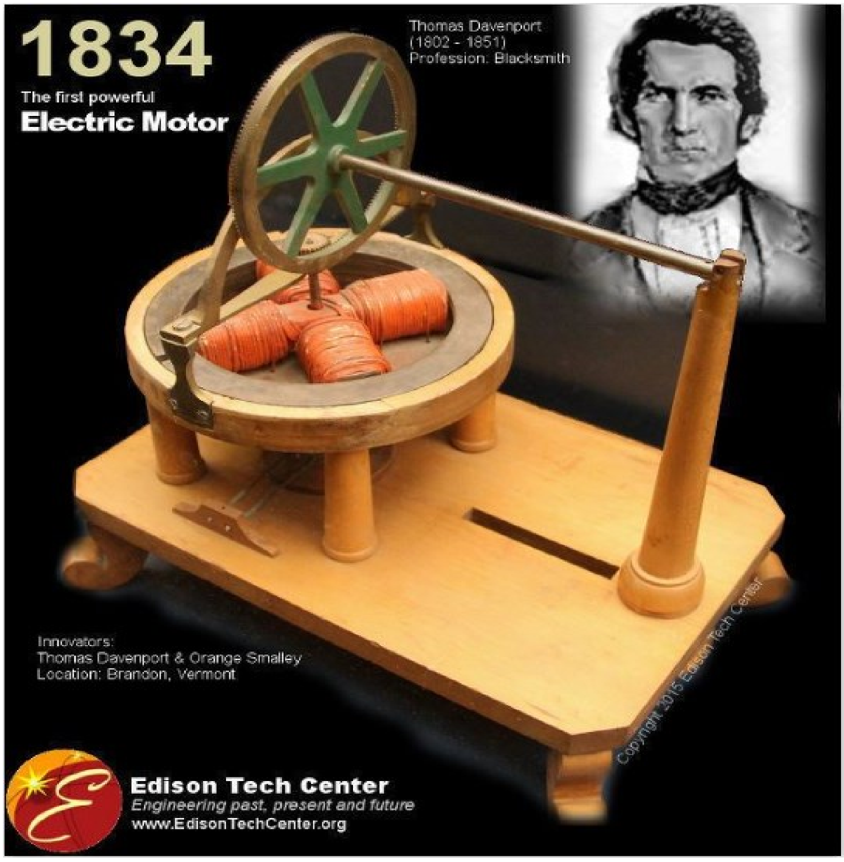


Figure 1. Thomas Denver’s electric motor, the first “real” motor (Edison Tech Center, n.d.)

Until 1873, the electric motor eventually achieved commercial success where Zenobe Gramme demonstrated the first DC electric motor, as illustrated in Figure 2, in exhibitions in Vienna and Philadelphia. 14 years later, Frank Sprague showcased the world the first practical application of electric motor with the electric trolley system. (eNotes Editorial, 2018)

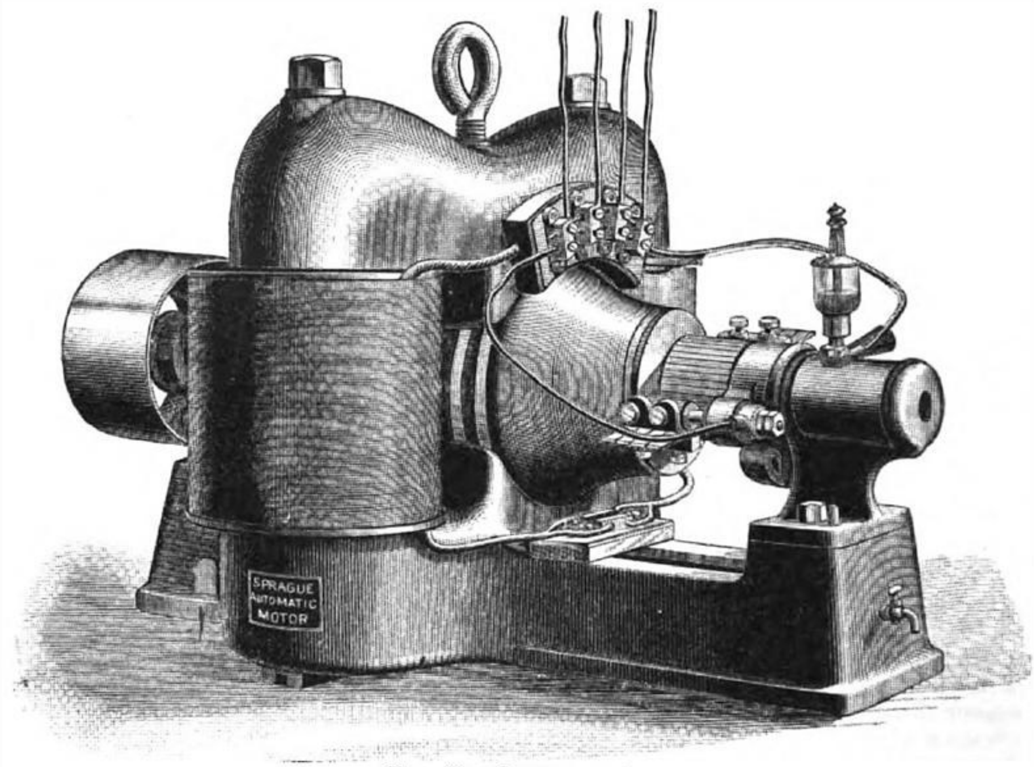


FIG. 29.—Sprague motor.

Figure 2. Frank Sprague's DC electric motor (Vintage Machinery, 2011)

In the case of induction motor (AC motor), the first induction motor (figure 9) was introduced by Nikola Tesla in 1888 resulted in the Second Industrial Revolution as it made efficient generation and long-distance distribution of electric energy by using alternating transmission system, possible. In the 20th century, an enormous expansion of electrical power distribution happened all around the globe. A 25 000-kW capacity was considered a large generating unit for the first decade of the century, but 20 years later, a 208 000 kW was the largest unit in the United States. Due to the huge amount of electricity, the world's technology has been thriving perpetually since then. (BBC, n.d.) (Puiu, 2016)

2.3 **DC motor**

DC means direct current which is the power source for DC motors. DC motors are used wherever variable speed is required. However, due to the limitation of battery-operated motors and the development of variable frequency drives, the popularity of the DC motor has fallen off significantly.

A DC motor consists of three main components:

- Stator: the stationary part which is the permanent magnet.
- Rotor: copper wire wound into a coil mounted on an axle. The magnetic field which is the result of electricity flows through the coil, interacts with the field created by stator and creates rotational mechanical energy by spinning the axle.
- Commutator: commutator is a split copper-made ring, with each segment of the ring attached to the end of the coil. Commutator is to ensure the coil do not move in opposite direction and the axle spins correctly, commutator is needed. (Beckett, 2018)

DC motors are classified into a brushed DC motor and a brushless DC motor. Firstly, a motor brush or carbon brush is the small part of the motor that conducts the electrical current between stator and rotor. Now it is the time for explanation and classification of those two motors.

Brushed DC motor

As described before, the stator is the permanent magnet with north pole and south pole as seen in Figure 3, which generates a magnetic field surrounds the rotor. The rotor or armature of the motor consists of the windings. The winding often is copper wire would into a coil mounted on an axle. When electricity flows through the coil, the magnetic field is made, it interacts with the field made by stator and creates rotational mechanical energy by spinning the axle. The distinctive component of this motor is brushes and commutator. The commutator is a split copper-made ring, with each segment of the ring attached to the end of the coil. The commutator is to ensure the coil do not move in opposite direction and the axle spins correctly, commutator is needed. The brushes are connected directly to the power source. When the motor is running, the brushes contact with different segment of the commutator rings as it rotates. That process ensures the precisely rotation of the commutator and axle. The basic diagram of a brushed DC motor is shown in figure 3.

Brushed DC motor can be connected directly to the power source and controlled simply by a switch; therefore, it saves initial investment. However, due the constant physical contact, brushes and commutators wear out which results in maintenance and replacement fee. The noise comes from the switching action of the commutators is also one of the disadvantages of brushed DC motor. In addition, the efficiency of brushed DC motor is not high (75-80%) compared to brushless DC motor (85-90%). The efficiency of a motor is how much that motor transform the total power usage into rotational force instead of heat dissipation. (Quantum Devices, 2014)

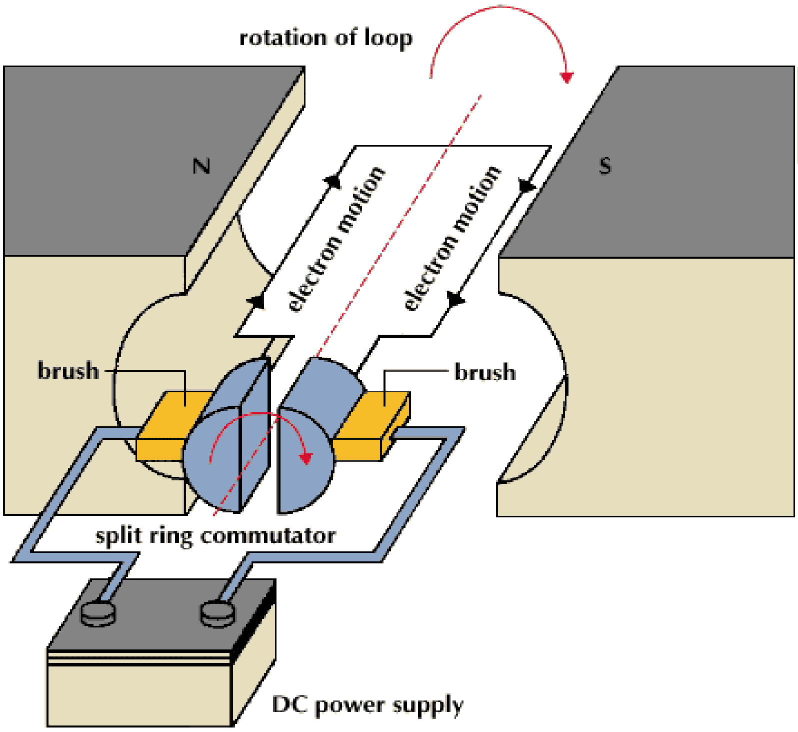


Figure 3. Basic brushed DC motor diagram (Kundu, 2018)

How the stationary magnetic field is produced in the stator differentiates the brushed DC motor into four groups: a permanent magnet, a shunt wound, a series wound and a compound wound motor.

- Permanent magnet

A permanent magnet brushed DC motors (PMDC) displayed in Figure 4 uses the permanent magnet to generate stator magnetic field. The persistent of the magnetic field makes these motors react instantaneous with voltage changes. PMDC motors are in favour where applications involving fractional horsepower due to the cost effective of permanent magnets. The disadvantage is the permanent magnetic wears out the magnetic properties over time. (Condit, 2004)

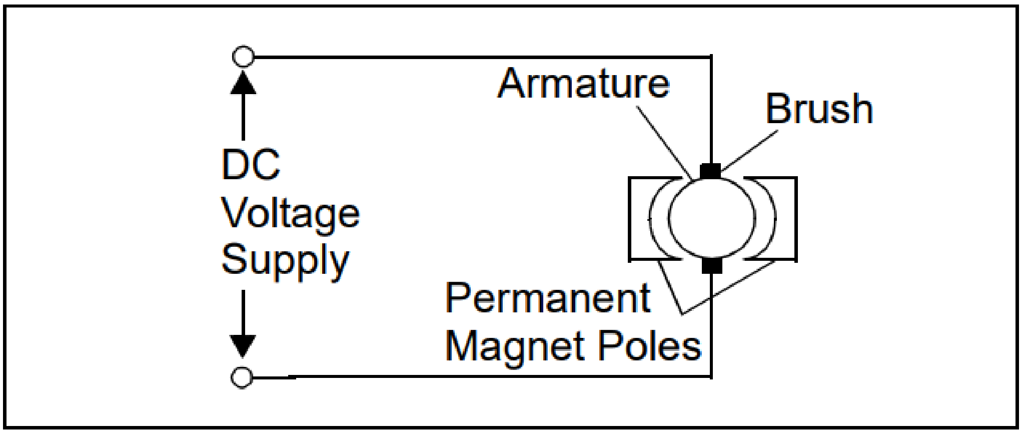


Figure 4. Permanent magnet DC motor (Condit, 2004)

– Shunt wound

A shunt wound brushed DC motor is one that has the field coil connected parallel with the armature, therefore, the current in the field coil is independent from the current in the armature as shown in Figure 5. Five or more horsepowers is needed is where shunt wound brushed DC motor comes in handy. (Condit, 2004)

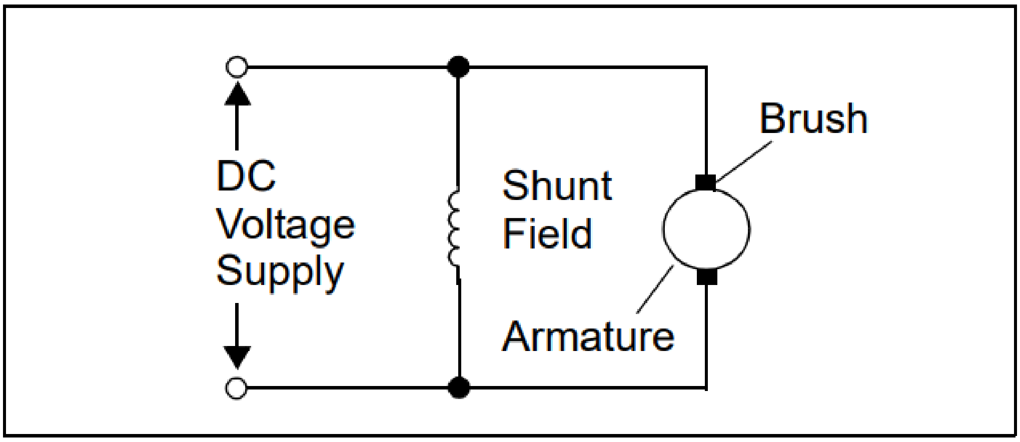


Figure 5. Shunt wound brushed DC motor (Condit, 2004)

– Series wound

A series wound brushed DC motor as seen in Figure 6, has the field coil connected in series with the armature. This motor is an ideal option for high torque applications because when the motor carries a load, the current in stator and rotor both increases. However, the precision in speed control is poor comparing with permanent magnet and shunt wound motor. (Condit, 2004)

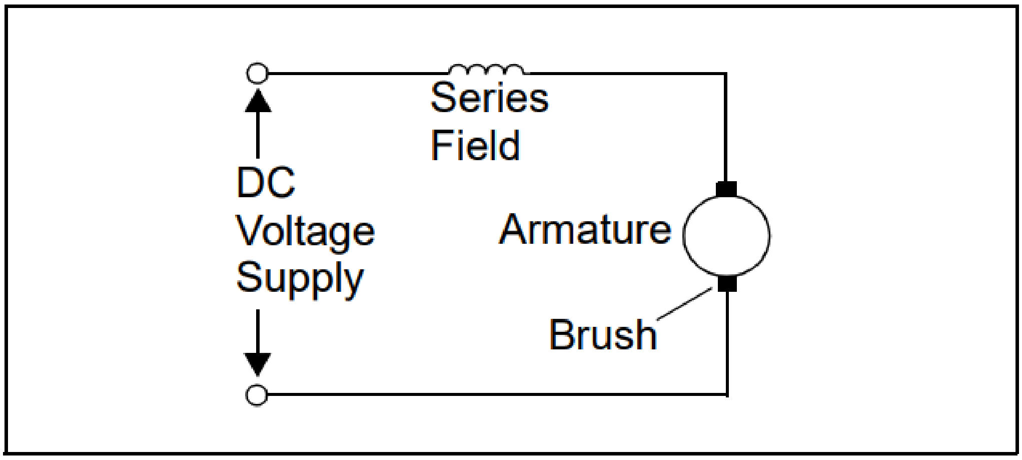


Figure 6. Series wound brushed DC motor (Condit, 2004)

– Compound wound

The combination of series wound and shunt wound makes one a compound wound motor shown in Figure 7. Both series and shunt field exist inside the motor. The compound wound motor have higher torque than shunt wound motor yet displaying a better controlling than shunt wound motor. (Condit, 2004)

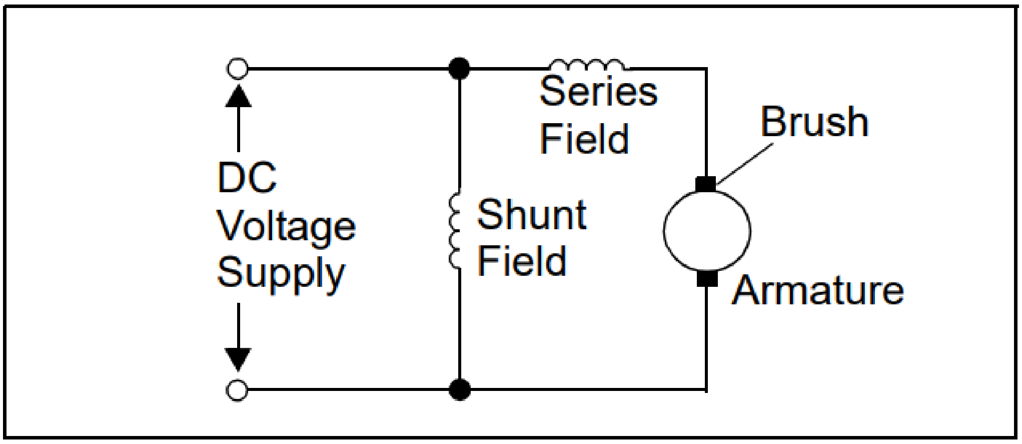


Figure 7. Compound wound DC motor (Condit, 2004)

Brushless DC Motor

The stator and the rotor of each motor is what differentiates those two. The stator of brushed motor is the permanent magnet where the stator of brushless one is the group of coils providing electromagnet. The rotor of the brushed motor is the group of coils where the rotor of brushless one is the permanent magnet. This results to two questions: which coil to be energized by the electricity and when to energize it. A hall effect sensor is used to ensure the rotating field by acknowledging the angular position of the rotor (figure 8). Hall effect sensors which are mounted either 60° or 120° apart from each other, are magnetic field sensors whose job is to monitor the rotor’s position. Three hall effect sensors are mounted to either the rotor or the stator and use six-steps commutation. Most motors have a small set of magnets mounted at the end of the rotor shaft. These magnets are aligned in the same way as the main magnets of the rotor itself. Every time one of these small magnets passes by a sensor, the sensor gives a low or high signal indicating the S or N pole of magnet, which results in rotor’s position detection. As it can be seen in figure 8, one of the hall sensors changes its state after every 60° electrical rotation of the rotor magnetic field. the three-digit code “010’ or “100” represents the position of the rotor field as well as the state of the sensors. Each digit shows the high signal (1) when the sensor is triggered by the North Pole of the rotor and low signal (0). (Learn Engineering, 2014) (Gürocak, 2016)

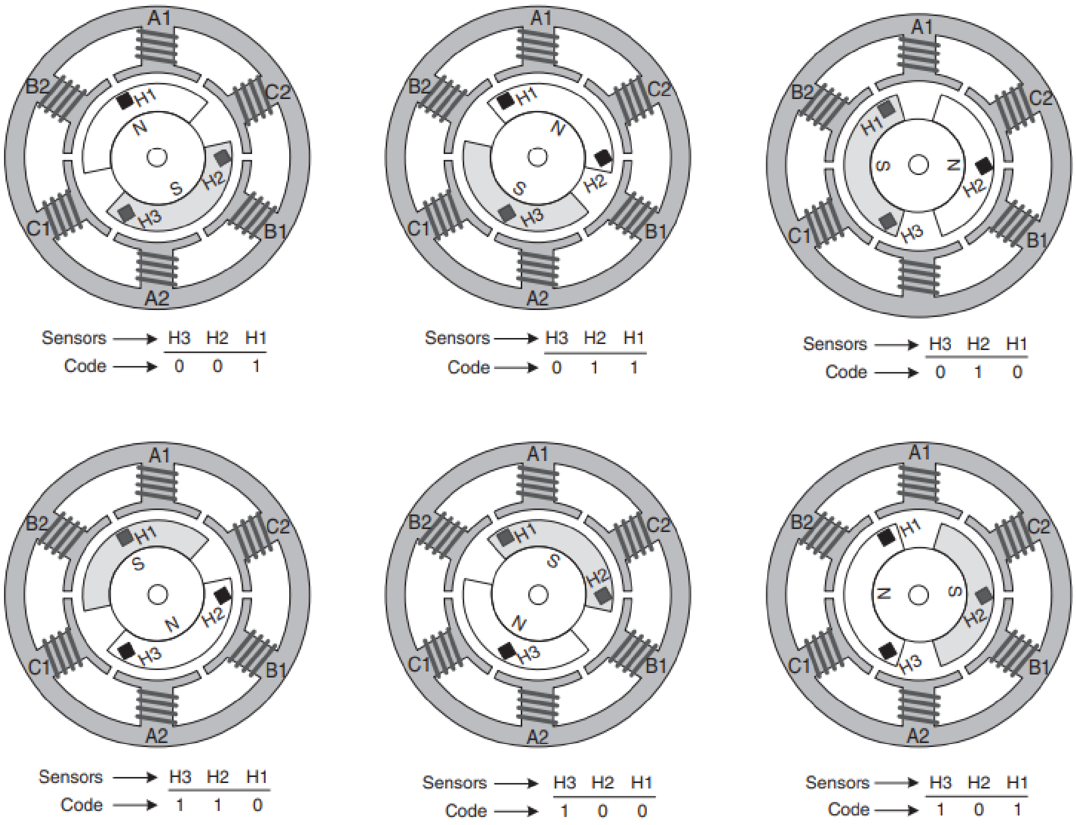


Figure 8. Three hall sensors create six segments for the course measurement of the absolute position of the rotor field. Each segment is identified by three-digit code (Gürocak, 2016)

The phases of a brushless motor must be switch on/off electronically in a certain sequence to keep the two fields in proper orientation to produce torque. The electronic switching is called commutation and the six-step commutation is one of the most common commutation algorithms.

Since there are no brushes to wear out, the maintenance and replacement fee are low. But the initial investment of brushless DC motor is high due to the need of encoder or drive or controller (Quantum Devices, 2014).

2.4 **AC motor**

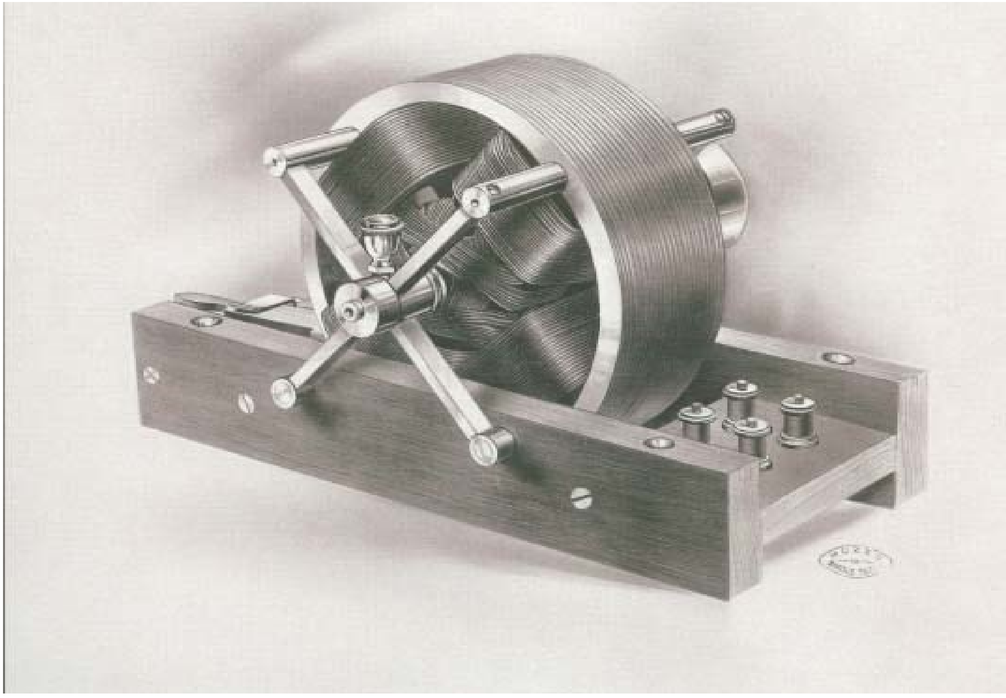


Figure 9. The world's first induction motor invented by Nikola Tesla (Puiu, 2016)

Same as the DC motor, the AC motor has the same design as a typical motor which consists of two main parts: a stator and a rotor. In the case of AC motor, the stator is a stationary part which has windings that transform electricity from power source into rotation magnetic field. The winding places into slots of the stator which are thin highly permeable steel laminations stacks together. The rotor is the rotation part of the motor which is made of aluminium or copper bars that run lengthwise. Those bars are induced by the rotation magnetic field of the stator and rotates the connected shaft continually. Due to the reduction in speed called slip, the induction motor runs slower than a synchronous motor. Three to five percent slip at full load is a normal rate of an induction motor. (Learn Engineering, 2017)

The AC motor is classified into two groups based on how to power it: phase (figure 10). The phase in electricity is the current or the voltage among the existing wire as well as a neutral cable. (El-Pro-Cus, n.d.)

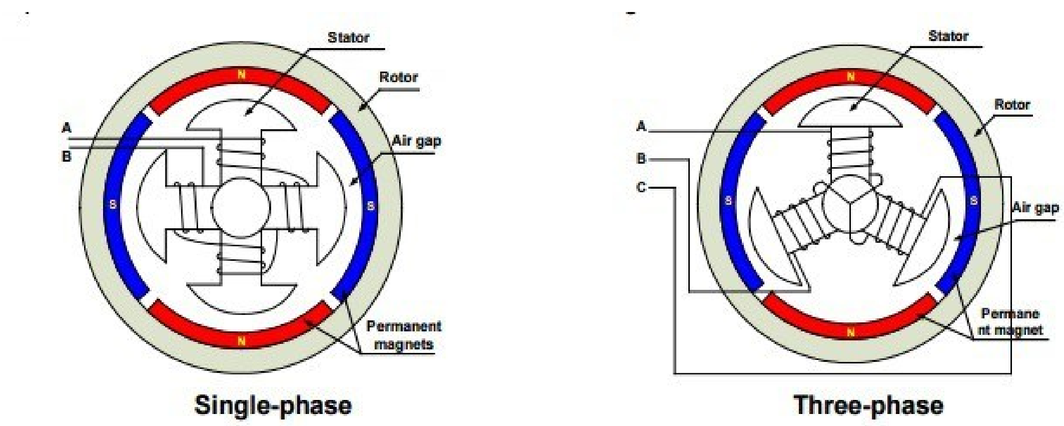


Figure 10. Comparison between single phase and three phase AC motor (LA Construction Heating & Air Conditioning)

- Single phase induction motor:

The motor runs by a single-phase power supply which means the power is provided by a single conductor (wire) with 230V. House appliances use this type of motor the most due to the small amount of power supply, simple design and economical initial investment. (El-Pro-Cus, n.d.)

The stator of the single-phase induction motor has main winding and auxiliary winding. The auxiliary winding is connected to a capacitor and placed perpendicular with the main winding. When electricity flows in the main winding, a fluctuating magnetic field is introduced. This magnetic field keeps a running rotor runs continually which means single phase induction motor is not a self-starting motor. A fluctuating magnetic field is equivalent to the sum of two opposite rotating magnetic field. Since those are opposite magnetic fields, the torques produced by them are equal and opposite results in the net force equal to zero. That is the reason why single-phase induction motor is not self-started motor, it runs properly if the rotor has an initial rotation. The auxiliary winding's purpose is to produce that initial rotation. The ingenious idea was Nikola Tesla's. The auxiliary winding cancels one of those rotating magnetic fields causing the rotor to rotate. (Learn Engineering, 2017)

- Polyphase (three-phase) induction motor:

Three phase power supply is provided for the motor. With the voltage of 415V, three conductors (wires) are connected to the motor, provide electricity and make it operate properly. Its applications are mostly in huge industries' factory where heavy loads are needed to be moved. The reasons why polyphase induction motor's popularity is in industry are the ability to operate at different loads and different speeds and its robustness when operates in hazard environment. (El-Pro-Cus, n.d.)

The stator is basically three coils winding provided electricity by a three phase AC power source. When the current flows through the windings, a rotating magnetic field is produced. The rotor's bars are induced with the current, it starts rotating. In compare with single phase induction motor, three phase motor is a self-started motor. The speed of the magnetic field is N_s, one of the rotor is N_{rotor} and the difference between them is call motor slip which is illustrated in this formula:

$$Slip = \frac{\text{Ns} - \text{Nrotor}}{\text{Ns}}$$

In addition, by changing the frequency of the power input, the speed of the motor can be easily controlled. (Learn Engineering, 2017)

2.5 **Familiar classification**

2.5.1 Stepper motor

In general, stepper motor is a brushless DC motor which rotates in steps. Stepper motor rotates precisely without closed feedback loop which means it is an open loop system. To explain how a motor is an open loop system, the structure of the stepper motor needs to be mentioned. Image a motor with 6 stator teeth which are energized by three DC power sources and 4-tooth rotor (figure 11). With this structure, only a pair of rotor teeth are aligned with the stator ones. A group of opposite stator teeth is energized, a pair of rotor teeth are aligned with those, when the next group of opposite stator teeth is energized while the first group is not, the rotor rotates 30° to be aligned with that group; the motion continues accordingly. In order to reduce the step size, we either increase both rotor and stator's teeth or energise two groups of opposite stator teeth and turn off one of them while energize the next group. (Learn Engineering, 2016)

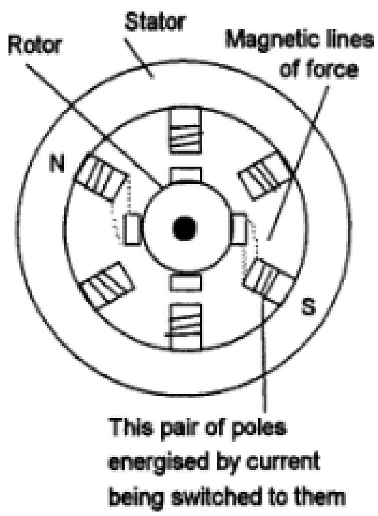


Figure 11. Stepper motor diagram (Bolton, 2004)

The motor can be driven in various ways depending on the step modes which is how and how many times the rotor rotates until a 360° rotation is achieved. These motors are described in the following.

The first mode is full step mode in which a 360° rotation is achieved with minimum steps (number of turns). The disadvantage of the full step mode is less inertia and less smooth of the rotation. Two approaches are used in full step mode: one phase-on stepping means only one group of opposite stator teeth is energized at given time and two phase-on stepping means two groups of opposite coils are energized which leads to better torque and speed but consume more power compare to one phase-one stepping. (Raj, 2018)

The next mode is half step mode which was described earlier on how stepper motor works generally. The combination on one phase-one and two phase-one stepping takes out all the disadvantages of those mentioned above. (Raj, 2018)

The last mode is micro step mode. Despite the fact that this is the most complicated mode, but it brings precision with good torque and smooth operation. The coil is energized with two sine waves which are 90° apart which results in better control the direction and amplitude of the current flowing through the coil. Micro step mode offers the huge number of steps that the motor can make to achieve the full rotation. (Raj, 2018)

2.5.2 Servo motor

Servo motors are commonly used in a closed loop system which consists of control circuit, amplifier, motor and feedback device. The motor is assigned with a specific position, velocity, acceleration, etc commands. The feedback device provides the information of how the motor operates in reality. Then the control unit shows the comparison between the desired motion and reality for the users to make changes or calibrations. (RealPars, 2018)

Servo motor is the most versatile motor and it can be various types. Servo motor can be either AC or DC, brushed or brushless and synchronous or asynchronous motor. The AC servo motor is the most common one which is robust, capable of withstand high current and frequently used for industrial applications. (RealPars, 2018)

2.5.3 Linear motor

Linear motor is the one which converts rotational movement into linear movement. Basically, linear motor needs guide rails to ensure the movement and the magnetic field; and especially the position feedback device. The most common applications of linear motor are sliding doors, conveyors, elevators, etc.

3 MOTION CONTROL SYSTEM

3.1 Definition

In general, motion control is about controlling motors to make mechanism movements in a high precision manner. The motor brings the acceleration and helps one division of the mechanical components reach the desired destination. The motor and the division make and axis. Considering a robot arm which consists of joints and arms, the arm and the motor helps it moving are one axis of the whole arm (Gürocak, 2016). There are two kinds of a motion control system: Open loop system which does not use feedback devices to compare desired outputs and reality ones, closed loop system which uses feedback devices.

Motion control is acknowledged as the heart of automated machinery and process. It can be encountered in every factory in which there are plenty of motors. In order to operate all those motors in a synchronized manner, a motion control system is needed. The basic components for a motion control system are controller, drive and motor. Position, velocity, acceleration and torque of an axis are governed by the system. However, to achieve the precision needed and adaptation for different load, feedback devices are necessary. To achieve the perfect outcome from a motion control system, the engineer needs to possess a numerous of engineering skills to understand the machine and the system thoroughly.

3.2 **History**

Machinery was brought to the world reducing human effort, improving productivity and making life more convenient after the first and second industrial revolutions. However, at that moment, controlling all those machines were a huge challenge. Not until about 70 years ago, a milestone was established in the process of motion control system revolution. Electronic controls were first designed as John Parson pioneered production to manufacture of Sikorsky helicopter blades. He optimized precise and accurate cutting of aerodynamic shapes of blades by exploiting the early computer technology. His action led to the standardize the concept of Computer Numerically Controlled (CNC). By the time of 1950s, control system became a standard of manufacturing for companies. CNC was a gigantic improvement comparing with manual machining of parts. Then Dick Morely's team created the programable controller in 1968 which opened up the opportunities and led the way to the third industrial revolution. Distributed control systems (DCSs) with special control requirements for PID algorithms and sequential control. Nowadays, any processors are capable of performing CNC, DCS, PLC function. Those are all control based with the differences in programming and tasking. (Meyer, 2016)

Perfect coordination is what a machine needs to achieve, and it was accomplished by mechanical means before the era of programmable controllers. The source of the motion came from a large electric motor or and engine which ran at a constant speed. A line shaft was connected to the motion source and was used to govern all the axes. Each axis was conjugated to the line shaft with different connecting methods such as pulleys, belts, linkages, etc as seen in Figure 12. Since only one motor or engine drives many axes, clutches and brakes were used to start or stop specific axis. The speed of one axis was determined by the gear ratios between the line shaft and that axis itself. The biggest difficulty was the system was made for only predetermined purposes. When changes were introduced, it required components adjustments or replacements, not to mention calibrations. (Gürocak, 2016)

Coordination in multi-axes machines is easily achieved through electronic gearing in software. In modern multi-axes machine, each axis has its motor and electric drive, the line shaft is replaced by shorted shafts which are connected directly from motor to the division of mechanical component of the machine illustrated in Figure 12. The typical motion system has a motion controller which interprets commands from a program then operate motors to reach desired intentions of users. With nowadays technology, eight axes coordination motion controllers are familiar. (Gürocak, 2016)

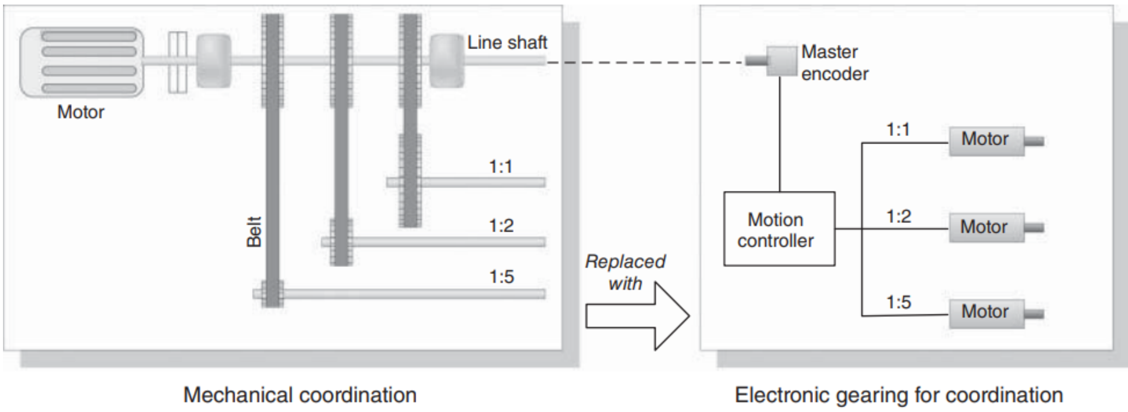


Figure 12. Multi-axes coordination (Gürocak, 2016)

3.3 Applications of motion control in modern world

In this chapter, three main applications of motion control in modern industry are described in the following.

3.3.1 Indexing

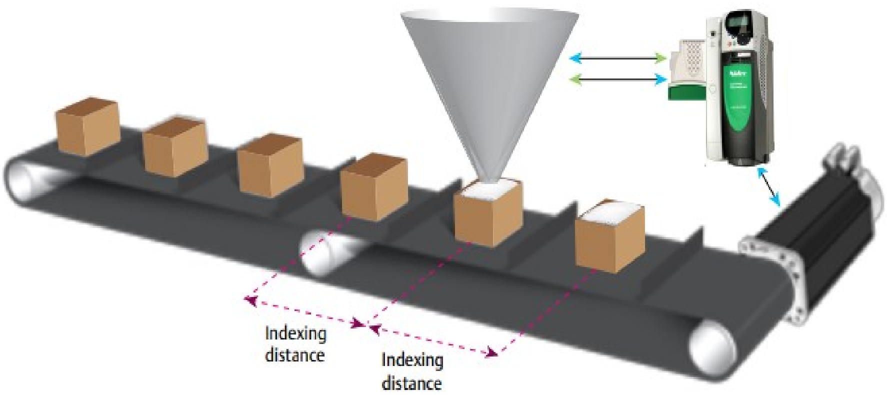


Figure 13. Conveyor indexing (Nidec)

When a motor runs, it brings the object to a predetermined position for the production line to continue its sequence. As Figure 13 illustrates, the conveyor brings the boxes to position which below a feeding tube. The boxes stay at the position for one minute while goods fill up then the boxes continue their way to be concealed and loaded up. It is essential for the motor to runs and moves other objects at the exact predetermined distance in a given time. Following acceleration and deceleration profile makes the movement effortless and more precise. This type of application is known as single axis motion (discrete motion with time). (Nidec)

3.3.2 Flying shear

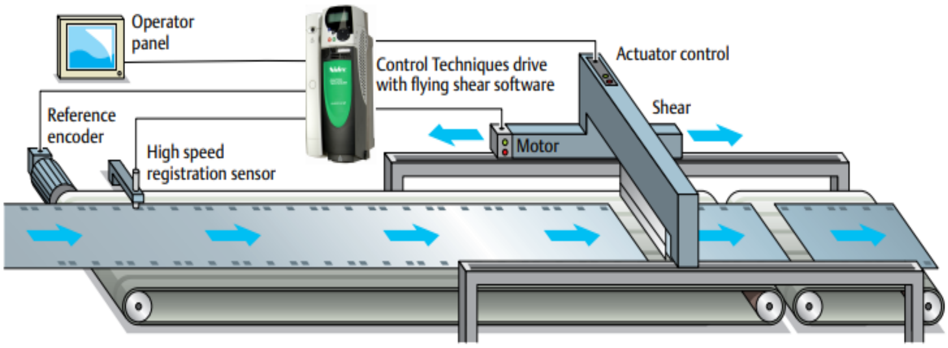


Figure 14. Flying shear saw application (Nidec)

A flying shear shown in Figure 14 is a reference/ follower system. It is used in cutting a continuously moving product to a specific length, the cutting mechanism (follower) must be synchronised with the speed that the product is moving (reference). To achieve the synchronisation, the speed and position of the production must be measured by reference encoder, then it tells the motion controller to adjust the speed of the follower. After the action of cutting is executed, the shear reverses and returns to its initial position. Flying shear is known as 1.5 axis interpolated motion (synchronised discrete motion) where the accurate synchronisation is the most crucial element of the whole system. (Nidec)

3.3.3 Pick and place

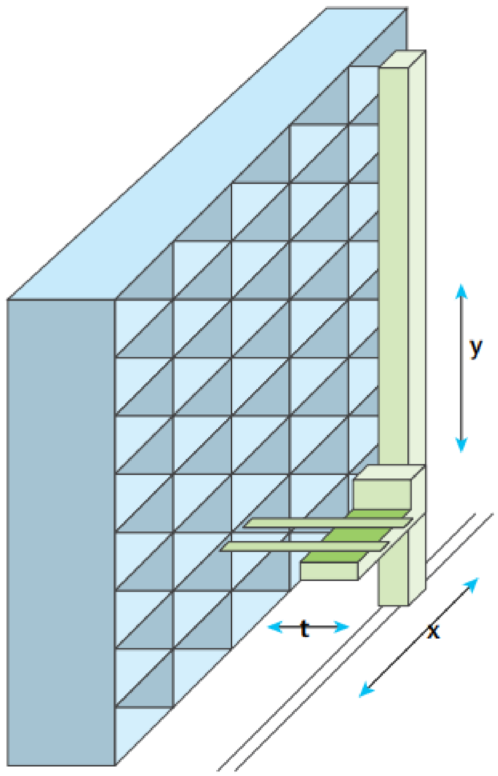


Figure 15. Pick and place application (Nidec)

As the name says for itself, pick and place machines as seen in Figure 15 are used for handling processes where an item needs to be picked up and placed in a different location. An advanced pick and place machine can operate in all dimension at the same time for a higher efficiency. For example, the manipulator (the device that picks up the item and places it) after picking up the object will move diagonally to the destination. To achieve the compatibility of multi-axes movement, common time synchronised clocks are needed. They help defining uniform movements. Since the machines are able to operate in two or three dimensions, they are called X-Y or X-Y-Y tables. (Nidec)

3.4 **Components**

In general, the combination of the motion control software and controllers works like a human brain in comparison with human body. While motor and actuator represent for muscle and joint, feedback devices or sensor operate as human senses in the whole motion control system. (Kerns, 2018). A complete motion control system needs devices as figure 16 illustrated.

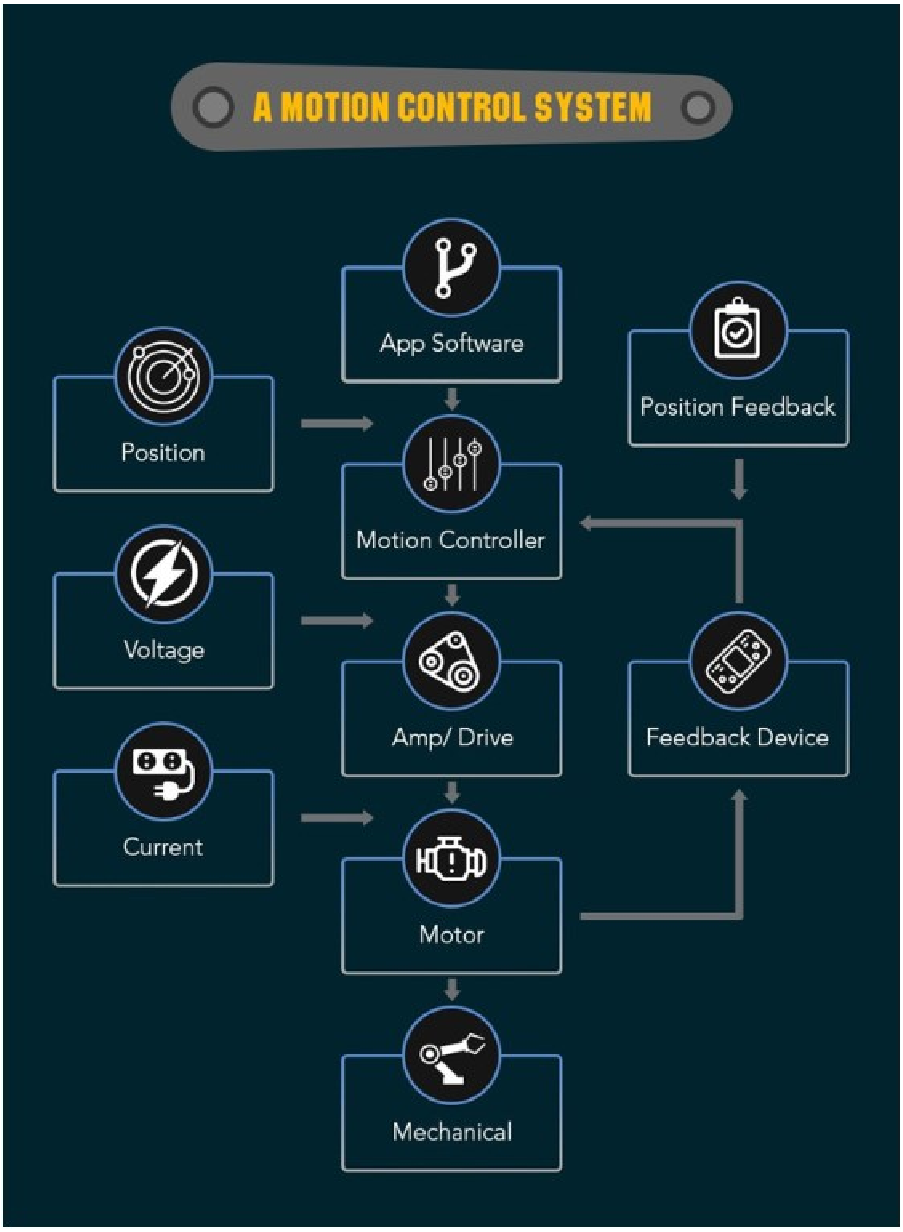


Figure 16. A motion control system (Kerns, 2018)

3.4.1 Motion control software

First off, the motion control software is where an engineer establishes his power and control the whole system, as well as writes programs for coordination and drives synchronisation. However, since late 1990s early 2000s, the developments of motion control enable engineers to focus more on the system design instead of spending hours programming. Not only 40 motion axes capability system but also system with drive tuning is available. In the drive tuning software, a few parameters are all that needed to be provided, the software itself does the rest. In addition, some software support basic element of motion control which are electronic gearing, relative and absolute moves, registration and programmable limit switches. (Langnau, 2000)

3.4.2 Motion controller

Secondly, motion controller is what receives commands from motion control software and tells the motor to do its job. Also, it does the comparison between the input command and the feedback signal for the purpose of immediate action to carry out the desire output. Motion controllers can be divided into three group: stand-alone, PC based and individual microcontrollers. Stand-alone controller is an entire system which consists of power supply, electronics and external connections all concealed in a physical box. Stand-alone controllers are built in a machine that dedicated to one motion control application which brings limitations for the machine itself and users also. PC based controller is mounted onto the motherboard of a basic PC or industrial PC. The PC based controller is processing board that generate and execute motion profiles. This type of controller brings convenient for costumers with the users-friendly interface. The last one is individual controller (IC) which is mounted onto a circuit board along with feedback inputs and outputs. IC is inexpensive and gives users the chip-level access, however, users are required relatively good programming skills. (Motion Control Tips Editor, 2011)

3.4.3 Drives

The next device in the list is drives. It is essential to distinguish motors and drives. Motor is an electrical device that generates rotational or linear force. In other words, motor is a device that physically do the work or carry load. In the other hand, drive is the device that controls the electrical energy sent to the motor. Users choose frequency that being sent to the motor which means users are controlling the speed and torque of the motor. (Mraz, 2015)

3.4.4 Feedback devices

Besides motors which had been mentioned in chapter 2, feedback devices is the last in the list. Depending on the design of the machine and possibilities of improvement that users decide which feedback devices are needed for the machine. However, there are some feedback devices that are must-have. Quadrature encoder which gives momentary position in comparison with the start position. Tachometers which provide velocity feedback. Resolvers serve the purpose of converting mechanical motion into an electric analog signal to find absolute position. Many more to list. (Kerns, 2018)

3.4.5 Mechanical components

The whole motion control system must serve a specific purpose and that is through mechanical components to achieve desired motion. Two of the most popular mechanical components are linkages and actuators. Linkage is an assembly of bodies connected to manage forces and movement. An actuator is a device that create motion in straight line. (Kerns, 2018)

3.5 **Motion profile**

When an object is moved from its initial position to its destination, the object follows a trajectory which is the path was created for the object to reach its destination. In motion control, a trajectory is called motion profile. The motion profile is constructed to ensure the accuracy and safety when an object moves from its initial position to the desired position. Moreover, the motion profile illustrates the velocity, acceleration, deceleration and position commands. The object accelerates from the starting position to reach a constant predetermined velocity moving toward the destination. Until it goes pass a certain point on the way, the object starts to decelerate and stop when it reaches its destination. (Gürocak, 2016)

Before getting into details, some of the basic terminology will be explained. Velocity profile is the graph illustrates the changes of the speed when motor accelerates from 0 to when it decelerates back to 0. Position means the distance from starting position after completing the motion. Jerk indicates the rate of changes of acceleration with respect to time. To understand further about motion profile, two familiar profiles will be demonstrated in detail.

3.5.1 Trapezoidal velocity profile

First one is trapezoidal velocity profile. The trapezoidal velocity profile is familiar due to its simplicity. As it can be seen in Figure 17, the profile itself alongside with the acceleration, position profiles and jerk (also known as jolt, which is a derivative of acceleration). The parameters are usually known as velocity v_m, acceleration a, distance travelled by the load s.

The acceleration and deceleration might not be equal but the total motion always equal to the sum of time of acceleration, deceleration and constant speed:

$$t_{total} = t_a + t_m + t_d$$

The total distance that the load travelled is calculated as follow:

$$S = \frac{ta * vm}{2} + tm * vm + \frac{td * vm}{2}$$

In the case the time of acceleration and deceleration are equal, the total distance and the move time are:

$$S = v_m(t_a + t_m)$$

$$T_m = \frac{L}{vm} - ta$$

(Gürocak, 2016)

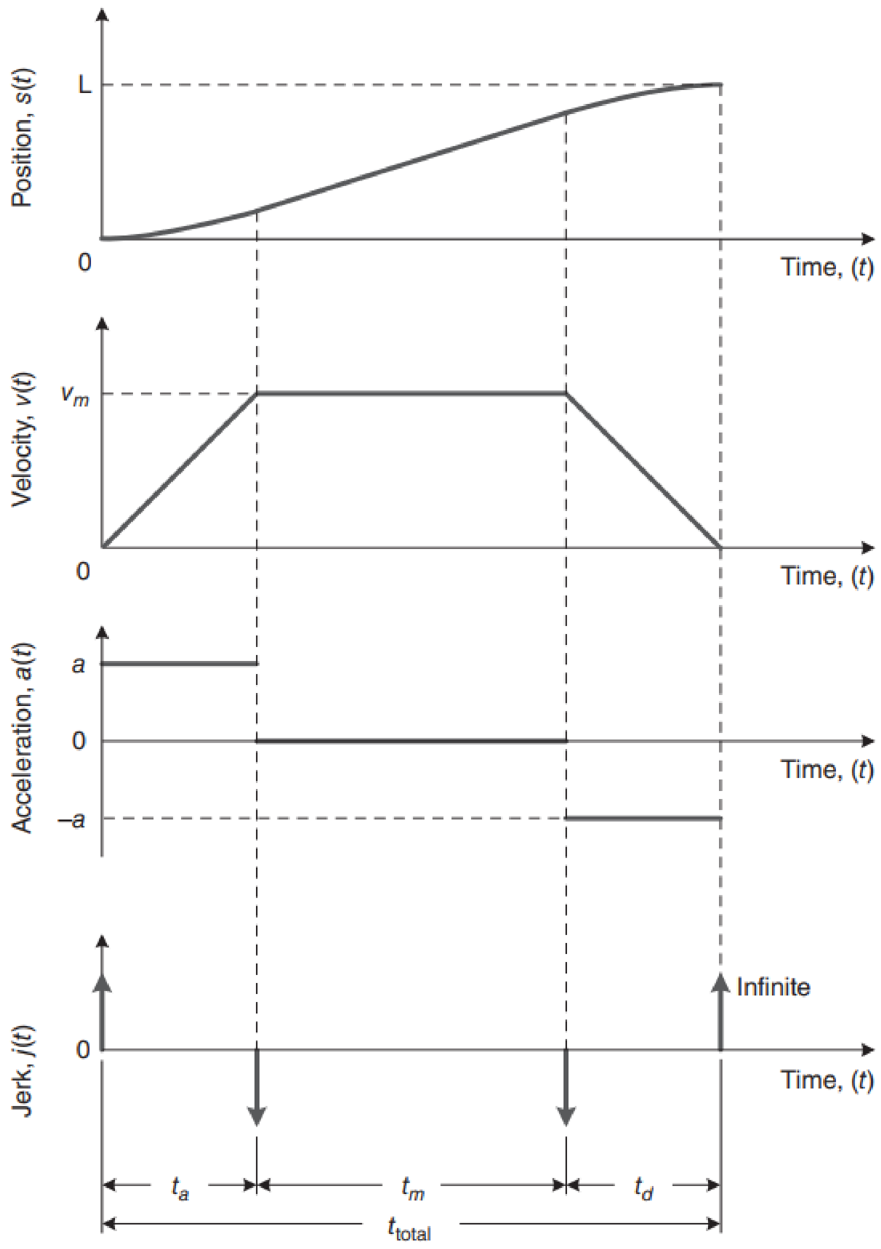


Figure 17. Trapezoidal velocity profile and position, acceleration and jerk profiles (Gürocak, 2016)

3.5.2 S-curve velocity profile

The other familiar profile is S-curve velocity profile (figure 18). This kind of profile makes the whole motion system smoother. The sharp corners result in discontinuities in acceleration and deceleration of the trapezoidal profile which lead to infinite jolts in the system. In the other hand, those corners are rounded to create S-curve velocity profile. Unlike the trapezoidal profile, the acceleration is not constant. As the velocity changes, the transition between positive, zero and negative acceleration phases happens genuinely. The jerk in S-curve velocity profile is finite and as long as it stays that way, the cycle operation does not be disturbed by sudden shock loads. As shown in Figure 18, seven distinct phases are being illustrated. The four curved segments are implemented using quadratic equations. The remaining three are straight lines with positive, zero and negative slope. (Gürocak, 2016)

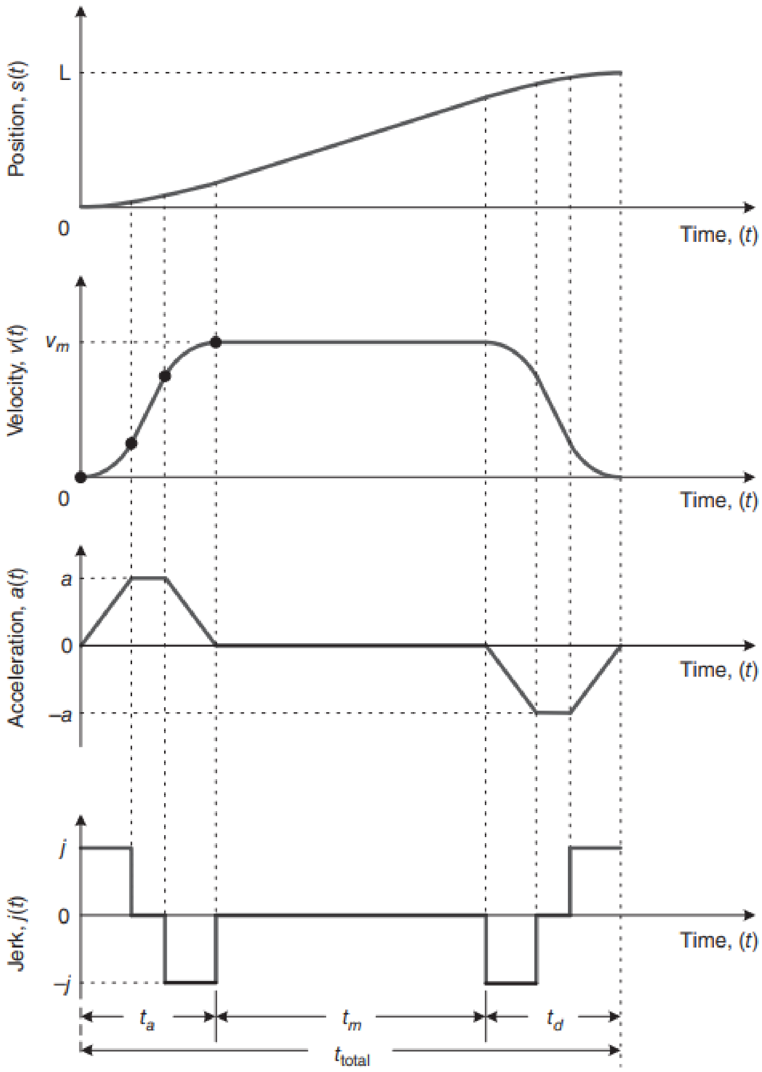


Figure 18. S-curve velocity profile and position, acceleration and jerk profiles (Gürocak, 2016)

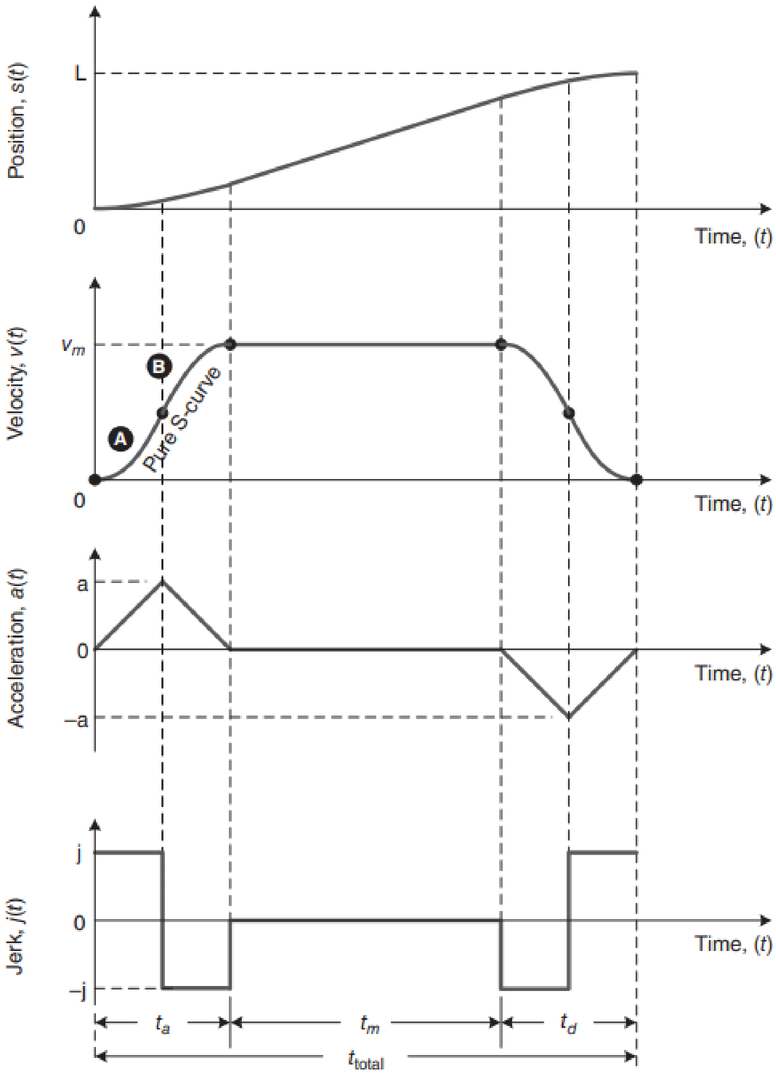


Figure 19. Pure S-curve velocity profile (Gürocak, 2016)

In the case of linear segments elimination, the S-curve velocity profile becomes pure S-curve velocity profile as figure 19. The equations to calculate all elements for S-curve velocity profile are shown in table 1 where C_1, C_2, C_3 are coefficients to be determined using boundary conditions (Gürocak, 2016)

S-curve Velocity Profile (Figure 2.9)

$$t_a = \frac{2v_m}{a}$$

$$C_1 = \frac{a^2}{2v_m}$$

Curve A

$0 \le t \le \frac{t_a}{2}$

$$s_A(t) = C_1 \frac{t^3}{3}$$

$$v_A(t) = C_1 t^2$$

$$a_A(t) = 2C_1 t$$

Curve B

$\frac{t_a}{2} < t \le t_a$

$$s_B(t) = C_1 \frac{t_a^3}{24} + v_m \left(t - \frac{t_a}{2}\right) - C_1 \cdot \left\{ t_a^2 \left(t - \frac{t_a}{2}\right) - t_a \left(t^2 - \left(\frac{t_a}{2}\right)^2\right) + \frac{1}{3}\left(t^3 - \left(\frac{t_a}{2}\right)^3\right) \right\}$$

$$v_B(t) = v_m - C_1 (t_a - t)^2$$

$$a_B(t) = 2C_1 (t_a - t)$$

Table 1. S-curve velocity profile's equations (Gürocak, 2016)

3.6 **Move mode**

Each move consists of moving type and constraints, which are move position/distance, maximum velocity and acceleration/deceleration. Based on these elements, the trajectory generator creates a velocity profile. Following the velocity profile, the axis starts moving from the starting point, accelerates to the predetermined speed the decelerates and stops when reaching the destination. (Gürocak, 2016)

3.6.1 Linear move

The most basic move mode that operates on a straight line between two points. There are two ways to specify the target positions: absolute and incremental. In absolute mode, each move is made with the respect to the origin location. For incremental mode, the subsequent move is made from the current position. (Gürocak, 2016)

3.6.2 Circular move

The tool tip (cutting tool) follows the circular arc which is generated as a path. This move mode requires multiple axes to be coordinated by the controller automatically. In addition, the starting point, final point and center point of the circular arc must be specified, also the direction of the move since clockwise and counter-clockwise arc (figure 20) are available. (Gürocak, 2016)

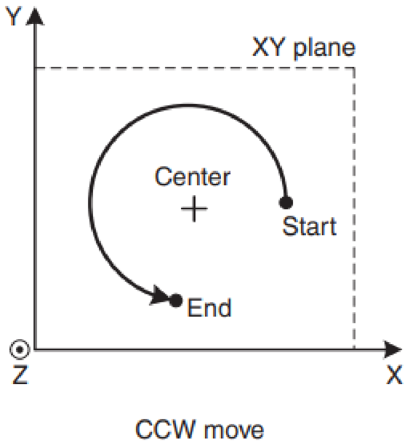


Figure 20. Clockwise and counter-clockwise arc (Gürocak, 2016)

3.6.3 Contour move

The trajectory is made up of many segments in some applications. The contour moves are required when trajectory cannot be constructed by using lines or arc segments. A sequence of positions is specified beforehand, then the controller connect those using splines and form the trajectory. Normally, position profile between two adjacent points is generated using cubic splines which prevent sudden changes in velocity or acceleration. (Gürocak, 2016)

3.7 **Velocity blending**

Blending the velocity of the first move to the second move is what makes the movement smoothly. The blending starts at the point of deceleration in first movement as figure 21 illustrates. When blending is activated, the end position of each segment may not be reached. (Gürocak, 2016)

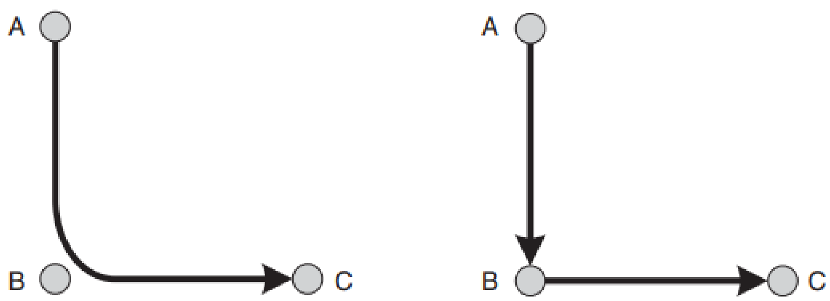


Figure 21. Two linear moves with and without blending. (Gürocak, 2016)

3.8 **Single-axis motion**

Single-axis motion is one axis moving at a certain time without any coordination with other axes. Two main types of single-axis moves are jogging and homing (Gürocak, 2016). These moves are explained briefly below.

Jogging is simply moving an axis, whose parameters are determined in advance to designate its velocity and acceleration. Jogging commands capable of making the axis move positive or negative direction to either specific position or continuously without a predefined stop. (Gürocak, 2016)

In the case of homing moves, the goal is to establish a reference position called home position. All subsequent moves are defined with respect to the home position once it is found. The most familiar approach if to use a move-until-trigger scheme to find the home position registered by a sensor. (Gürocak, 2016)

3.9 **Multi-axis motion**

Multi-axis machines require the perfect coordination between axis to fulfil the desired outcome. In motors' perspective, there are three basic movements of axes: move one axis at a time, move all axes at the same time (slew motion) and adjust the motion of axes so that they all start and finish at the same time (interpolated motion). In motion control system's perspective, coordinated multi-axis motion might involve multiple motors driving one axis, coordinated motion of two or more axes and following using master/follower synchronization. (Gürocak, 2016)

Slew motion represents for the concept of axes start moving at the same time with the same speed but each axis finished its motion at different time. However, interpolated motion represents for the motion where axes are coordinated by the controller. In the interpolated motion, when an axis that complete its task faster than others, controller slows it down to finish the whole system's task. There are two ways to tackle this problem which are whether slow down while keeping the acceleration time or slow down while keeping the acceleration. (Gürocak, 2016)

3.9.1 Multiple motors driving one axis

The common examples for this are gantry machines and advanced pick-and-place systems. Taking a gantry machine as an illustration, two motors drive the linear motion of the base axis in figure 22. These motors must be synchronized to prevent skewing the base axis and they are assigned to the same coordinated system that defines the axis. (Gürocak, 2016)

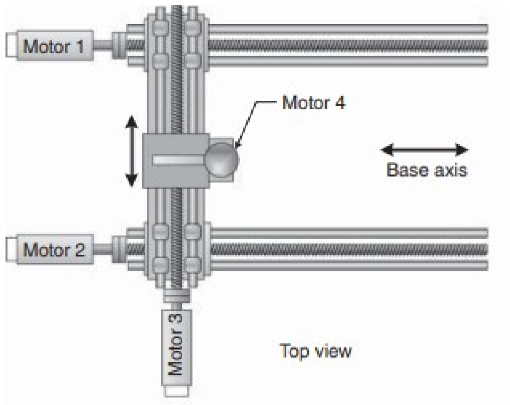


Figure 22. Gantry machine with two motors driving the base axis. (Gürocak, 2016)

3.9.2 Coordinated motion of two or more axes

Complex motion can be created by coordination of multiple axes controlled by motion controller. To be specific, each motor drives a single axis, but all axes move with coordination. Through circular and contour moves, the coordination is achieved successfully. By assigning the axes to the same coordinate system, the coordinated moves are accomplished, alongside with commanding those moves together at once.

3.9.3 Following (master/follower synchronization)

There are many motion control applications where some axes are not under influence of the motion controller. For example, in the case flying shear application, the feeding conveyor does not need the control of controller. However, the cutting mechanism and the cutting tool must be synchronized with the speed of the feeding conveyor to achieve desired outcome. As a result, the master/follower synchronization is needed. The eternal axis is the master (feeding conveyor), the one that follows the master is follower (cutting tool). The follower axis operates under the influence from the master encoder which is external encoder instead of internal trajectory generator. Nevertheless, coordinated motion is more recommended rather than master/follower synchronization due to the smooth trajectories. (Gürocak, 2016)

3.10 **Motion control programming**

As it was mentioned before, motion controller is considered the “brain” of the whole motion control system. Motion control not only monitors I/O but also generates and manages complicated motion profiles. Each manufacturer has their choice of programming language and environment for their own controllers. I will use my practical project to be example for this section.

3.10.1 Motion programs

The desired motion can be programmed using mathematical, logical, machine I/O operations and commands. Controller parameters can be set by the program. One motion program can call other as subroutines. All of the programming language used by controller have constructs to control the logic flow in a program. The basic constructs are WHILE loop, FOR loop, IF … THEN command or mathematical commands such as SIN, COS, SQRT, etc. (Gürocak, 2016).

The motor which generates the rotation as motion for the machine needs a driver, motion control module or inverter to control it as it working to achieve the desire purposes. Basically, all needs to be done are selecting the suitable settings. For the inverter, the parameters required to be set are frequency, speed, deceleration and acceleration time. Those parameters guarantee the motor works properly for simple purposes without controlling with programming code. Because of that, the inverter does not provide the positioning control as driver or motion control module does, that is the reason why a machine or a unit that required a precise position has motor with driver or motion control module, this is where a motion control system is used. According to my experience, a PLC with positioning control sends out commands for the driver or motion control module through communication protocol or output signals. These types of motor controller process the commands and make the motor rotate to complete the motion. The parameter required adjustment for those are the total of pulse input and output and rotation speed. Commands in Mitsubishi PLC provide option for types of positioning control, output pulse and rotation speed so that user do not have to change settings of motion controllers. In addition, motor operates in an industrial environment usually has gearbox. This should be considered and calculated with caution since it affect the speed of the motor alongside load. In particular, one specific motor rotates 3000 revolutions per minute as standard speed, with a 1/10 gear box, 3000 becomes 300 revolutions per minute, load would also lower the RPM.

3.10.2 PLC functionality

PLC is one type of controller that commonly used in automation since 1960s. A PLC program is written mostly in ladder logic format; however, structure text is also a well-known format. A PLC program runs continuously in a scan-mode which means the program runs in an infinite loop one the program is triggered. In each scan cycle, the PLC reads the state of all I/O devices and the whole program from start to bottom to determine the new state of output devices with the given state of input devices. Typically, PLC programs carry out monitoring inputs, update outputs, configuring hardware, sending commands and communication with the human-machine interface. Motion controllers mimic the scan cycle of the PLC as a software that runs on the motion controller itself. It runs PLC programs in the background while executing the motion programs. (Gürocak, 2016).

The best example is my practical work. I chose Mitsubishi components such as: FX3G PLC, GS2107-WTBD HMI; to be the brain which controls and monitors the whole system. The PLC and HMI communicates using RS485 protocol, a RS485 cable to be specific, due to the fact that both of them are capable of using this protocol. For them to be linked, the setting of the HMI must be adjusted to be compatible with the PLC, the setting is showed in figure 23. With these settings and using the same variable for the same purpose, the communication is formed and running.

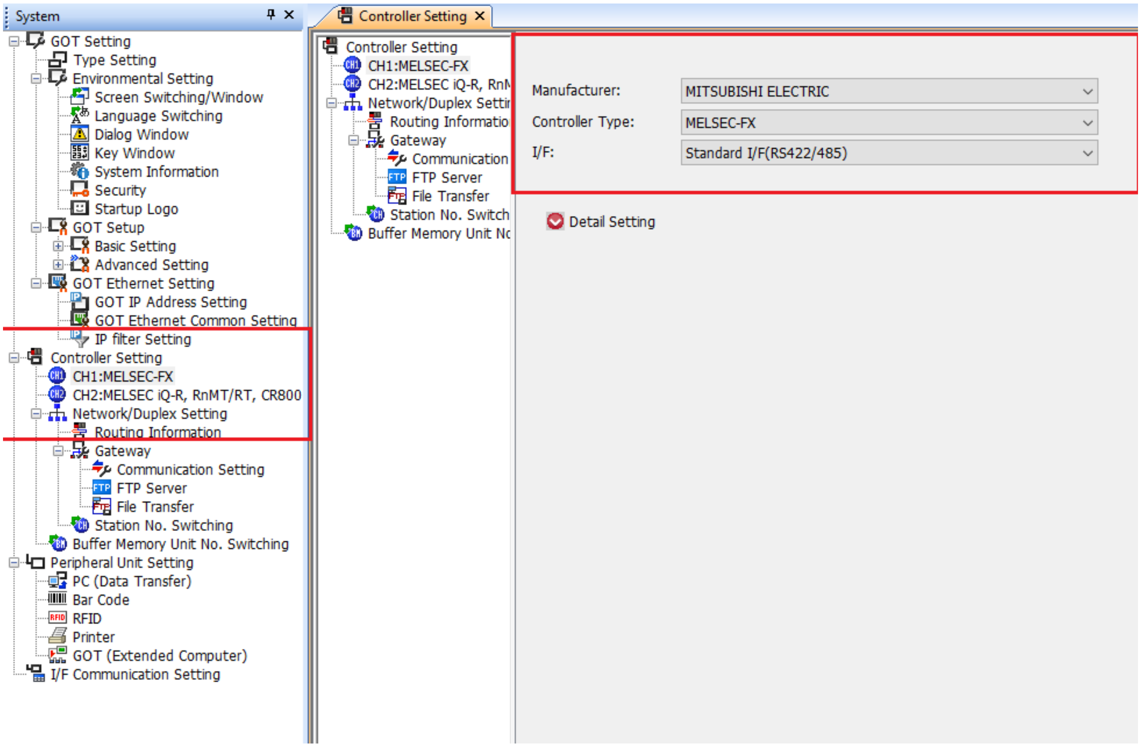


Figure 23. PLC-HMI communication setting

As I mentioned in section 3.10.1, Mitsubishi PLC has commands that optimize the positioning control experience for user. Figure 24 indicates with one simple command, user can drive the motor to the position one wishes. DDRVI means drive to increment which executes one-speed positioning by incremental drive. To simplify it, user provide the total output pulse, the pulse frequency can also be called speed, the device number from which pulses are to be output and the device number to which rotation direction signal is output. The total output pulse can be positive and negative according to users' wishes. Both of those device number are output from PLC; in this case, Mitsubishi FX3G PLC has 4 output pulse which are in pair: Y0 is pulse output and Y4 is rotation direction, the same for Y1 and Y5 respectively.

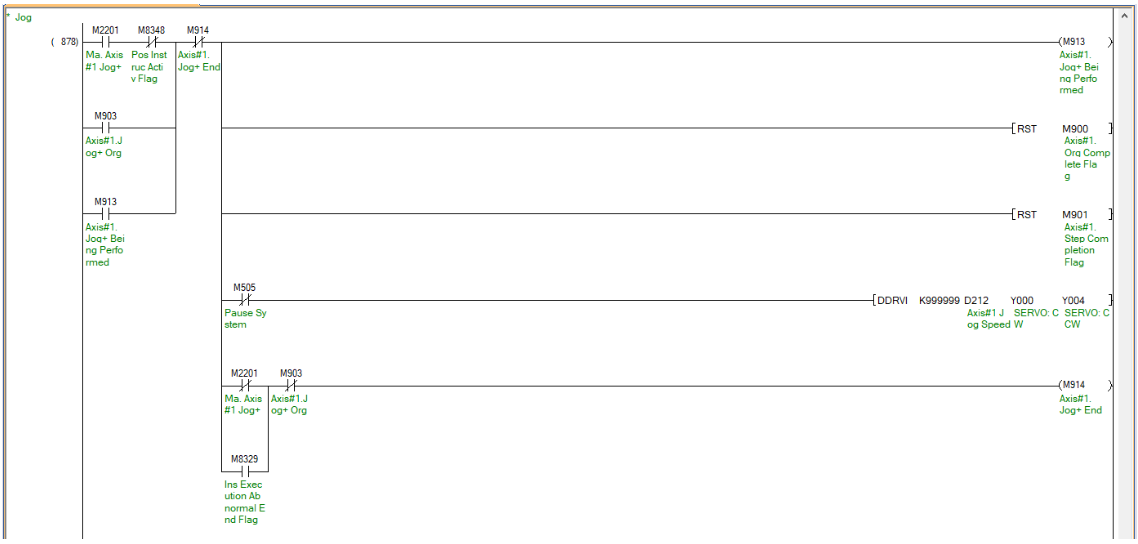


Figure 24. Mitsubishi PLC – Positioning control command

4 PF APPEARANCE CHECKING UNIT

The machine was the result of researching web-based sources, literature and research papers to understand through beyond basic information about the motion control system and motors theoretically; and field experience. Then I used all the above to select equipment, to design an electric box, to wire all the equipment and to program the machine to work as expected. The machine is described in detail in the following.

4.1 Basic concept and operations

The PF Appearance Checking Unit is a machine that uses a camera and a vision system to evaluate products for frauds in the rubber bands or scratches at both ends of the axes. The machine is an experiment one to test the process and the mechanism in order to design a fully automated system.

In the following there are some pictures of the machine and three units of its operation. First one is the vision unit which contains camera which is provided by the customer and testing stage takes place at their factory; and a servo motor as seen in Figure 25. Delta components were chosen. The servo motor model was ECM-B3M-C20604RS1 and the driver is ASD-B3-0421-L. The movement of the mounting bracket is due to a ball screw, a lead screw shaft and a linear movement guide. The servo journey is limited and controlled by three OMRON photomicro sensors of type EE-SX674A.

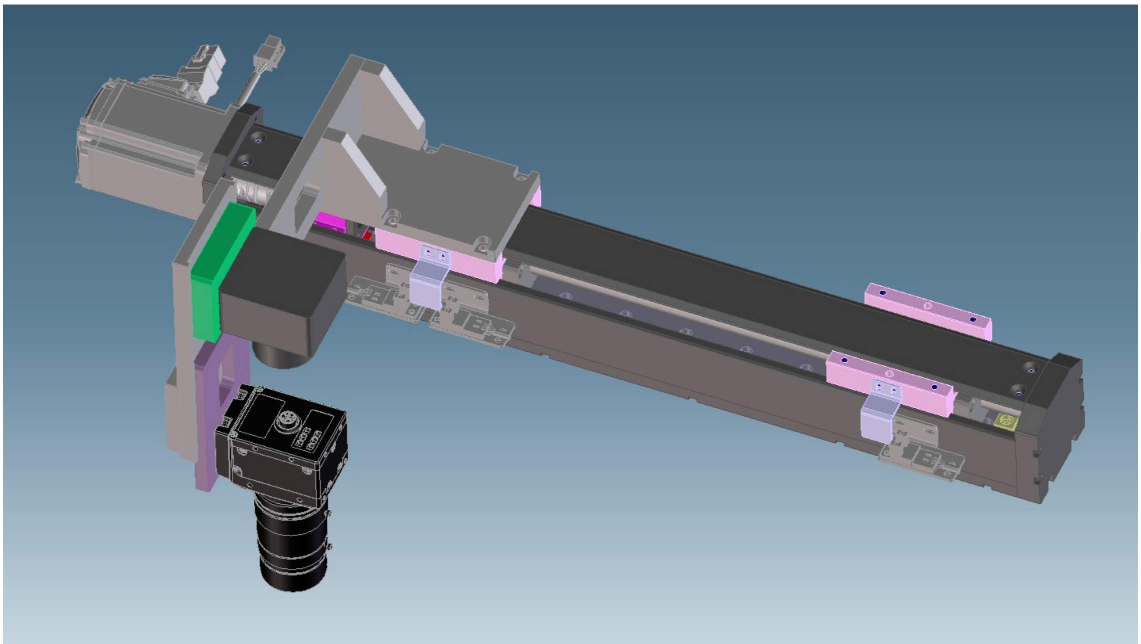


Figure 25. Vision Unit

Figure 26 is the rotary unit, where the stepper motor rotates so that four circular axis rotate which leads to the rotation of products. Autonics components were selected for this project, A16K-G268 was the model of the stepper motor with the MD2U-MD20 driver. Figure 26 displays how the stepper motor rotates the products. As the motor is working, it rotates the belt which leads to the movement of the bearings.

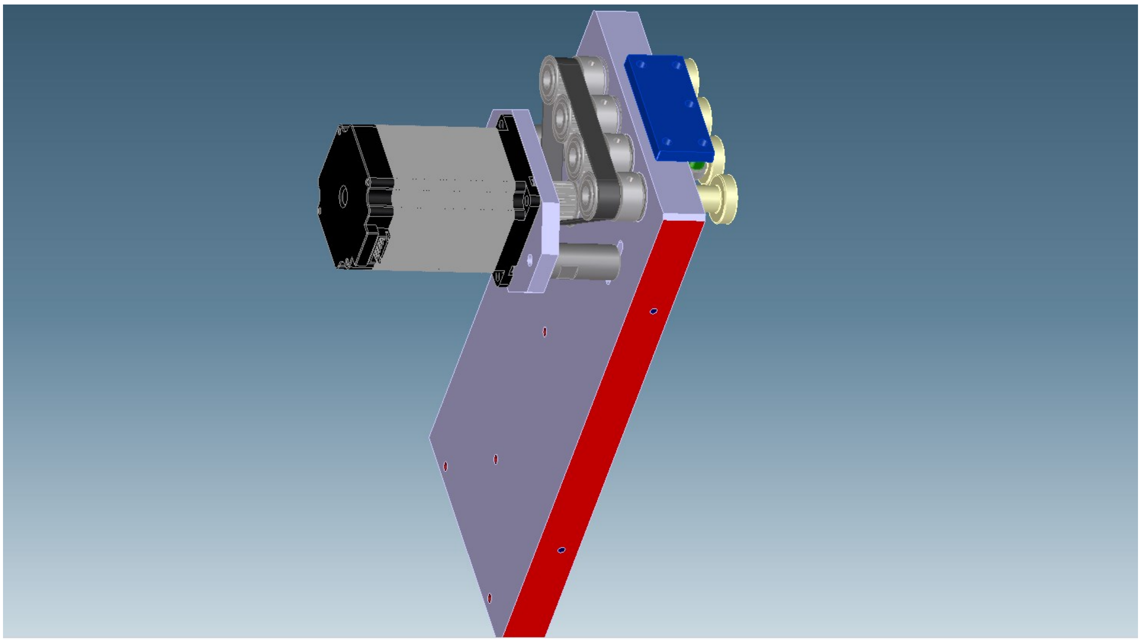


Figure 26. Rotary Unit

Figure 27 shows the cylinder whose main purposes are supporting the rotation and stabilizing the products. The shaft and the hand twist are for adjusting the position of the whole unit. Two SMC limit switches D-A93L are used to indicate the current position of the cylinder.

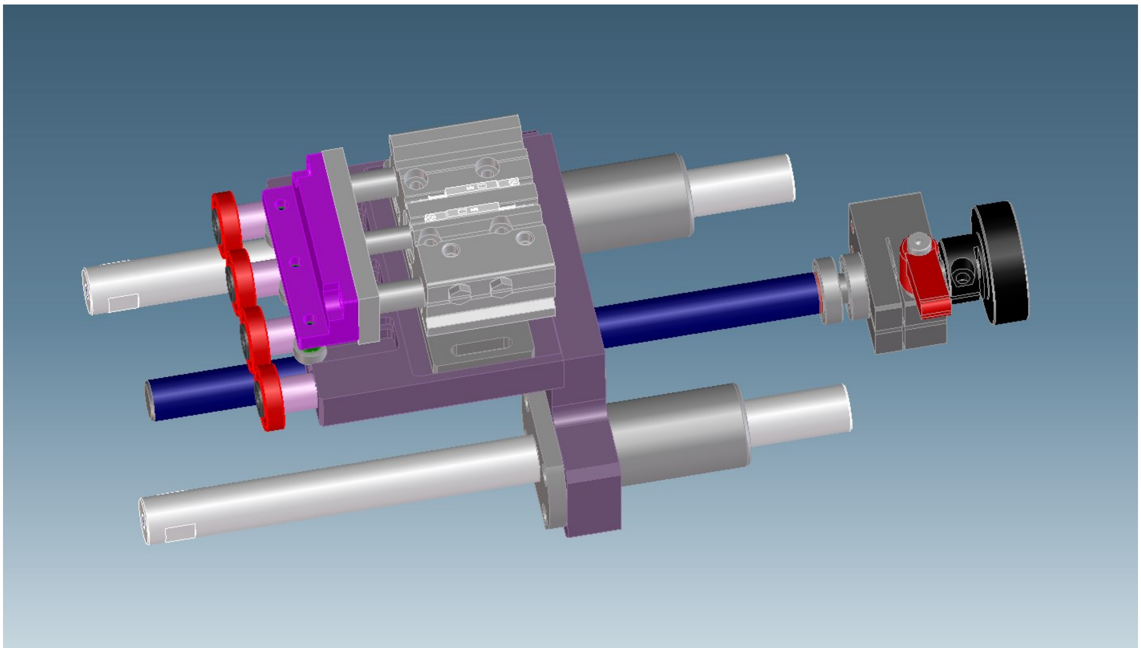


Figure 27. Cylinder

4.2 **Electrical equipment**

4.2.1 Motor

The servo motor chosen was ECM-B3M-C20604RS1 and the driver is ASD-B3-0421-L, both were from Delta and both needed 220VAC power supply. The 400W motor was a three-phase motor which could operate at 3000 revolutions per minute as a standard speed. The driver assisted me to do positioning control and monitor the operation of the motor precisely. The servo rotates in a linear mode and moves the camera into the position to do the quality checking.

The stepper motor was A16K-G268 and the driver was MD2U-MD20 which were from Autotonic and worked properly with the 24VDC power supply. The motor’s resolution was depended on how the microstep was chosen. In this case, I chose a full-step setting which meant per one pulse the motor rotates 1.8°. With this parameter, I could easily calculate how many pulses the motor needed to reach the desired position or angle. The stepper motor rotates the product so that the camera can take pictures and evaluate the quality of the product.

4.2.2 Cylinder

As a cylinder, SMC MGPM16-10Z whose initial position was backward was chosen. This is the air cylinder which operates forward when air flows through the solenoid and back to the initial position when air is blocked at the solenoid. The cylinder helps stabilize the products when they are being rotated by the stepper motor and checked by the vision system. The two D-A93L sensors indicate the position of the cylinder. They were connected to the PLC so they could be monitor as necessary elements for the programming code.

4.2.3 Equipment in electric box

For a 24VDC power supply, Omron S8FS-C15024 was chosen. When provided 220VAC as its power supply, the output was 24VDC. 24VDC was the power source for the HMI, the sensors, the pushbuttons, the relays, and also be the common source for the PLC input and output.

Two types of relays were used: 24VDC and 220VAC. Two 24VDC relays used PLC outputs as input signals, with PLC outputs on, the relays received the signal and changed the polarity so that there was a current to the equipment behind them. The 220VAC was used to supply current to the electric box lamp and the fan. If the door is closed, the lamp is off and the fan is on, when the door is open, vice versa.

FX3G-40MT was the PLC used. Its power source was 220VAC, however, the two common sources for input and output were P24 and N24 respectively in order to receive and send out signals to the equipment.

4.3 **Process of designing the machine**

Since the machine was a company project, I did not do everything on my own. First, the mechanical team designed the machine and sent the 3d model to the electrical team. In the meantime, I consulted the mechanical team and my supervisor when choosing the electrical components for the electric box. In the case of the two motors, work load was calculated with caution and precision to ensure those chosen motors reach the requirements. As soon as I received the 3d model, I started to design and layout the electric box as seen in Figure 28 which included an aptomat, relays, a PLC, a servo driver and a stepper driver. The purpose of the high degree of auxiliary components’ space was for experimenting. As it was said before, the machine served experimental causes so auxiliary space was for when customer wanted to assemble more equipment to the machine.

Figure 28. Electric Box

4.4 **Brief introduction to hardware and software**

4.4.1 Hardware

Firstly, a Mitsubishi MCB BH-D6-2P-10A was used for the 220VAC power supply. The total power (P) was calculated carefully before choosing the MCB. The next component was the noise filter WOONYOUNG WYFS06T1M which was used to reduce noise to the lowest level when electricity travels through the line. The 24VDC power supply operated onto a 220VAC power source to provide a 24VDC power source for the relays, the HMI, the and IO from PLC. The PLC used for this machine was an FX3G-40MT which included 24 inputs and 16 outputs. The PLC was linked to the HMI GS2107-WTBD via a RS485 cable.

4.4.2 Software

Since the main controlling unit was Mitsubishi's, the software for the PLC and HMI were GX Works2 and GT Designer3 respectively.

Ladder is the best method for a huge programming code and I had been working with this method for a while before I started working and have been doing it ever since. That is the reason for choosing ladder as I was working on this project. Nearly 3000 steps were conducted for the PF Appearance Checking Unit and it took over 1000 steps for the servo controlling code alone.

Syncing between the PLC and the HMI was not required as the auxiliary relays, timers, counters and data registers were used in the HMI to serve the same purpose as they did in the PLC.

4.5 **Programming stage**

I divided the whole program into 13 small and specific sections: “Main” which included auto/manual/origin mode selections, “power on” definition and password for the setting on the HMI; “Input”, “Define” where the home position of servo and the whole machine were specified, “Manual” – for the manual mode with a command from a virtual button on the HMI for the servo motor, the stepper motor and the cylinder fix; “ORG” – was the same as “Manual” where the command for the whole machine went back to the initial position, “Auto” - where the process was planned out and executed, “Error” where errors were determined, “HMI” which transferred virtual button on the HMI to command in the PLC, “Servo” and “Step” contained pulse-to-mm conversion, speed predetermination and commands for positioning control of both motors, “Data” contained the cycle time of the process and the quality counter by hours and by days, “Output” where all commands became actual operations.

The most difficult problem that I had to overcome was to make use of the built-in pulse output for the positioning control. Fortunately, there were functions and a good amount of special auxiliary relays and data registers to help me solve the problem. During the assembling period, I was doing the programming in advance, however, I was struggling with the positioning control using built-in pulse output. The special auxiliary relays and device registers are easy to understand but this is not the case with commands. Even though I was done with most of the program, it still took me a day to experiment with the code and the actual motor.

Figure 29 shows how I did the wiring for the IO cable of the servo motor. Y00 and Y04 are built-in pulse outputs which are used for positioning control, the PLC also give “servo on”-Y10 and “servo reset alarm”-Y11 signal for the servo driver. It takes in the signal for “servo ready” and “servo alarm” through input X20 and X21. The signal “servo EMG” is triggered when the servo is faulty in any way or the button EMG is pushed since the button is connected to the relay KA-Y16.

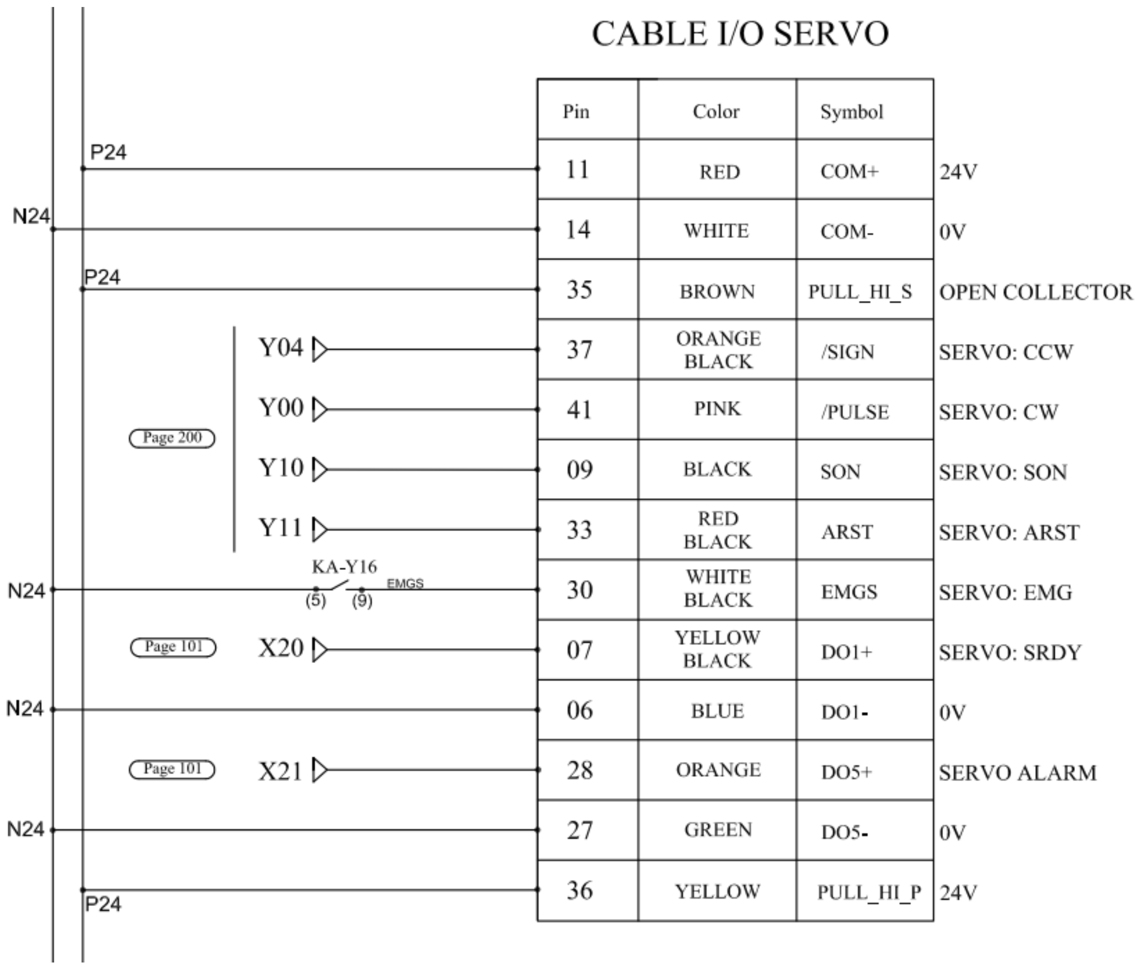


Figure 29. Wiring for servo IO cable

The same applied when controlling the stepper motor. The manual recommended resistor when using 24VDC for external driving pulse as seen in Figure 30. The customer required adjustable rotation quantity, for example, when the servo motor moved the camera to the desired position, the products rotated 12, 15 or 36 times. That rotation quantity represents how much the products rotate in degree; 12 times equals 30°, 36 times equals 106 times equals 10°. The conversion from pulse to degree was complicated, quite a few trials were done until I got it right.

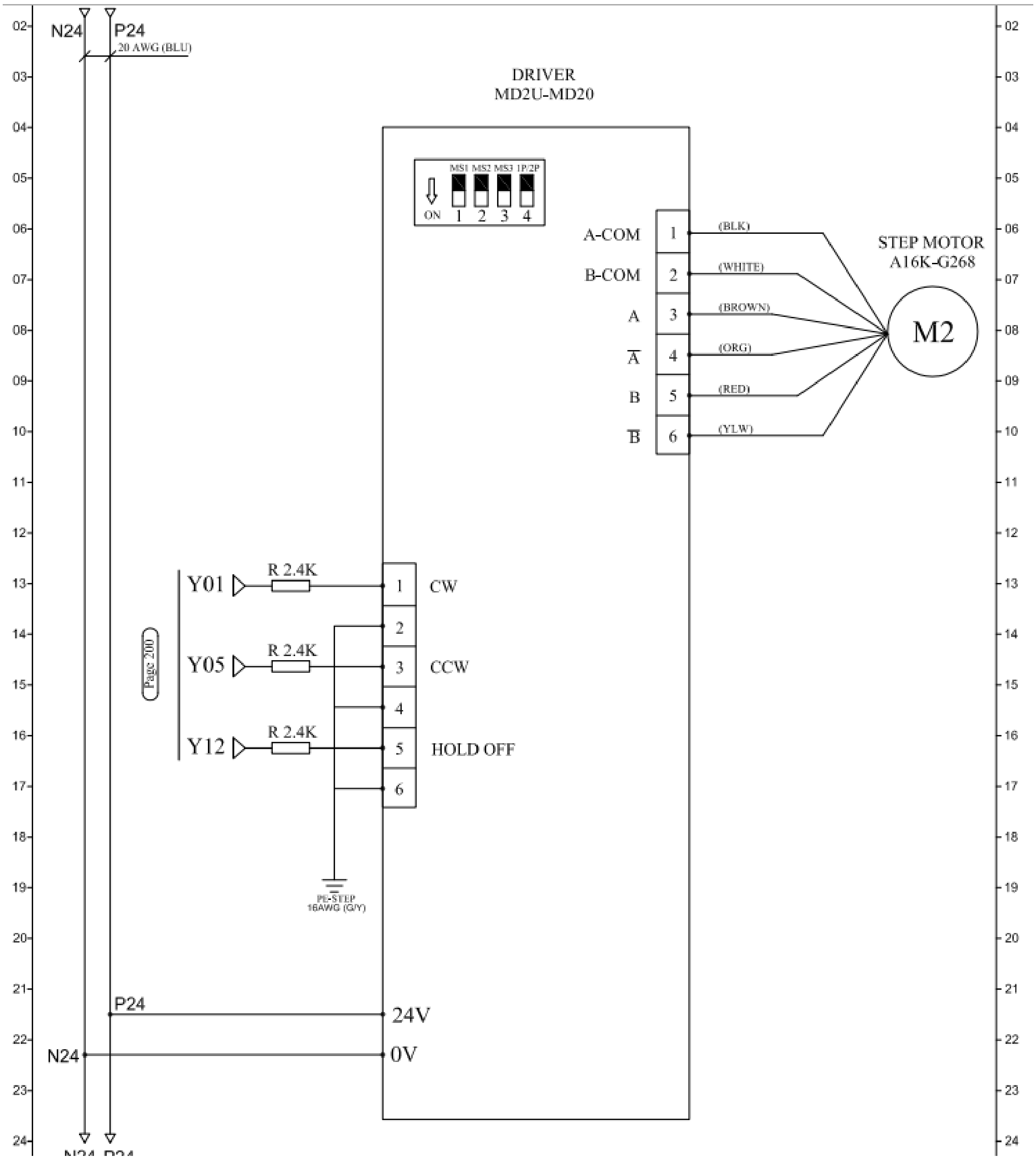


Figure 30. Wiring diagram for stepper motor

The entire program and HMI are attached in the appendix.

4.6 Testing and results

The process of testing the machine when it was done assembling was as follow: firstly, it was made sure that the power sources were correct and the wire was connected the same as in the electric drawing; next stage is called "IO check" where I made sure all the equipment worked properly, for example: the button was pressed, the light was on, the PLC received the signal, or the PLC and the HMI were on, etc; after that was the "manual" stage, at this stage, I downloaded the programming code and the HMI into the devices and controled manually all the equipment; lastly, the "auto" stage was where I let the machine run with the process that I coded and tried to adjust and improved the process until it met the requirements.

The design phase was done after a week and the whole machine was working properly after a month since the project first started. At first, the program was ready simply for just manual and the “IO check” and “manual” took a little amount of time. I started positioning control and the code for the process, this was where I struggled the most. To control both motors manually was quite doable, nonetheless, the pulse-speed-position conversion was the problem for positioning control. I tried a good amount of formulas to calculate the pulse, and after a day of doing them, I finally managed to complete it. As a result, the servo moved to the correct position as the pulse conversion. The same was done for the stepper motor with pulse conversion, however, in this case, I actually did a lot more calculation due to the step angle and microstep setting. Yet, I also finished it at the end of the same day I finished the control for the servo motor. After spending one day to solve the positioning control problem, on the day following, I finished the code for the auto process. However, that process was straightforward which the stepper motor rotated the during the auto process instead of having rotation quantity. The customer’s representative came to have a brief evaluation as soon as the QC department checked all the boxes which meant the machine was ready to be delivered. He asked for a light and a limit switch for the electric box, some adjustment for the the HMI: Comments were in both English and Vietnamese since their operators were from Vietnam, additional functions and processes as a few camera and vision systems will be used experimentally. I spent another day to complete all of these additional requirements.

The machine was delivered to the customer’s factory. They have been experimenting with it and combined it perfectly with two kinds of cameras and vision systems. Me personally have visited there twice to support the communication between the PLC and the vision system.

5 CONCLUSION

The theory part illustrates the basic concept of motors and the motion control system. Every categorized motor is described. Motion control system is analysed both generally and extensively. Basic applications of motion control system, the components which form the system, motion profiles, move modes and single-axis and multi-axis motion are depicted adequately. Additional information about the PLC and programming are also brought up in this thesis.

The thesis continues with the practical project commissioned by Autotech Machinery JSC. The project showcases the structure of how an automated machine operates in a production line, the components included and the programming code. The project is suitable for new employees which contribute to their knowledge and field experience enormously in the long run. The project was to design a machine electrically with the elements of motion control system: the PLC, two types of motor: a servo and a stepper one and drivers. The system is capable of controlling motion and monitoring it. The machine successfully met all the requirements of the customer both as to the electronics and mechanics.

In conclusion, the goal of the thesis was accomplished beyond expectation. An overall knowledge of motors and the motion control system are described in the theory part. The practical project was where I applied my knowledge into the electrical equipment selection, the electric drawing, designing the electric box, wiring and the programming code. The machine operates as a quality checking unit where the servo motor moves the camera to a predetermined position, the stepper motor rotates the products so that the vision system takes pictures to evaluate the quality and to check for frauds. The cylinder guarantees the stabilization of the products when rotated. The project displays a successful designing a machine and programming.

REFERENCES

BBC. (n.d.). *AC Motor*. Retrieved November 2019, from BBC: http://www.bbc.co.uk/ahistoryoftheworld/objects/66hoQYhWRJaUzW7qEycT3Q

Beckett, G. (2018, April 17). *Parts of a motor.* Retrieved November 2019, from Sciencing: https://sciencing.com/parts-motor-5426656.html

Beckhoff. (n.d.). Retrieved November 2019, from https://www.beckhoff.com/

Bolton. (2004). *Instrumentation and Control Systems.* Oxford: Newnes.

Condit, R. (2004). *Brushed DC Motor Fundamentals.* Microchip Technology Incorporated. Retrieved November 2019

Ducksters. (n.d.). *Physics for kids*. Retrieved November 2019, from Ducksters: https://www.ducksters.com/science/physics/electromagnetism_and_electric_motors.php

Edison Tech Center. (n.d.). *The Electric Motor*. Retrieved November 2019, from Edison Tech Center: http://edisontechcenter.org/electricmotors.html

El-Pro-Cus. (n.d.). *Difference between single phase and three phase AC power supply*. Retrieved November 2019, from Elprocus: https://www.elprocus.com/difference-between-single-phase-and-three-phase-ac-power-supply/

eNotes Editorial. (2018, July 1). *Who invented the Electric Motor in 1873?* Retrieved December 6, 2019, from eNotes: https://www.enotes.com/homework-help/who-invented-electric-motor-1873-115383

Gürocak, H. (2016). *Industrial Motion Control.* Chichester: John Wiley & Sons, Ltd.

Kerns, J. (2018, March 17). *Fundamentals of motion control*. Retrieved November 2019, from MachineDesign: https://www.machinedesign.com/motion-control/fundamentals-motion-control

Kundu, A. (2018, September 16). *What are the functions of various parts of an electric motor.* Retrieved November 2019, from Quora: https://www.quora.com/What-are-the-functions-of-the-various-parts-of-an-electric-motor

LA Construction Heating & Air Conditioning . (n.d.). *3 phase vs single phase AC motor: What you need to know.* Retrieved November 2019, from LA Construction Heating & Air Conditioning : https://laheatingairconditioning.com/3-phase-vs-single-phase-ac-motors

Langnau, L. (2000, May 1). *Motion control software: No programming required*. Retrieved November 2019, from MachineDesign: https://www.machinedesign.com/sensors/motion-control-software-no-programming-required

Learn Engineering. (2014, October 13). Brushless DC motor, how it works? Retrieved November 2019, from https://www.youtube.com/watch?v=bCEiOnuODac&t=74s

Learn Engineering. (2016, Octorber 19). How does a stepper motor work? Retrieved December 2019, from https://www.youtube.com/watch?v=eyqwLiowZiU

Learn Engineering. (2017, August 31). How does an induction motor work. Retrieved November 2019, from https://www.youtube.com/watch?v=AQqyGNOP_3o

Meyer, S. (2016, March 10). Steve Meyer on the history of motion control and PC-based control. (L. Elitel, Interviewer) Motion Control Tips. Retrieved November 2019, from https://www.motioncontroltips.com/steve-meyer-on-the-history-of-motion-control-and-pc-based-control/

Motion Control Tips Editor. (2011, Octorber 6). *What is motion controller? Technical summary for motion engineers*. Retrieved November 2019, from Motion Control Tips: https://www.motioncontroltips.com/motion-controller/

Mraz, S. (2015, June 25). *What's the difference between a motor and a drive?* Retrieved November 2019, from MachineDesign: https://www.machinedesign.com/motorsdrives/what-s-difference-between-motor-and-drive

Nidec. (n.d.). *A guide to motion control technology - system & programming.* Retrieved November 2019, from https://www.nidec-netherlands.nl/media/2925-a-guide-to-motion-control-technology-systems-programming-iss2x-0704-0007-02x.pdf

Puiu, T. (2016, May 27). *The history of the induction motor.* Retrieved November 2019, from ZME Science: https://www.zmescience.com/science/history-science/history-induction-motor/

Quantum Devices. (2014, August 27). *Brushless motor vs Brush motors, What's the difference?* Retrieved November 2019, from Quantum Devices: https://www.quantumdev.com/brushless-motors-vs-brush-motors-whats-the-difference/

Raj, A. (2018, Octorber 1). *What is stepper motor and how it works*. Retrieved December 2019, from CircuitDigest: https://circuitdigest.com/tutorial/what-is-stepper-motor-and-how-it-works

RealPars. (2018, August 27). What is servo motor and how it works? Retrieved December 2019, from https://www.youtube.com/watch?v=ditS0a28Sko&t=231s

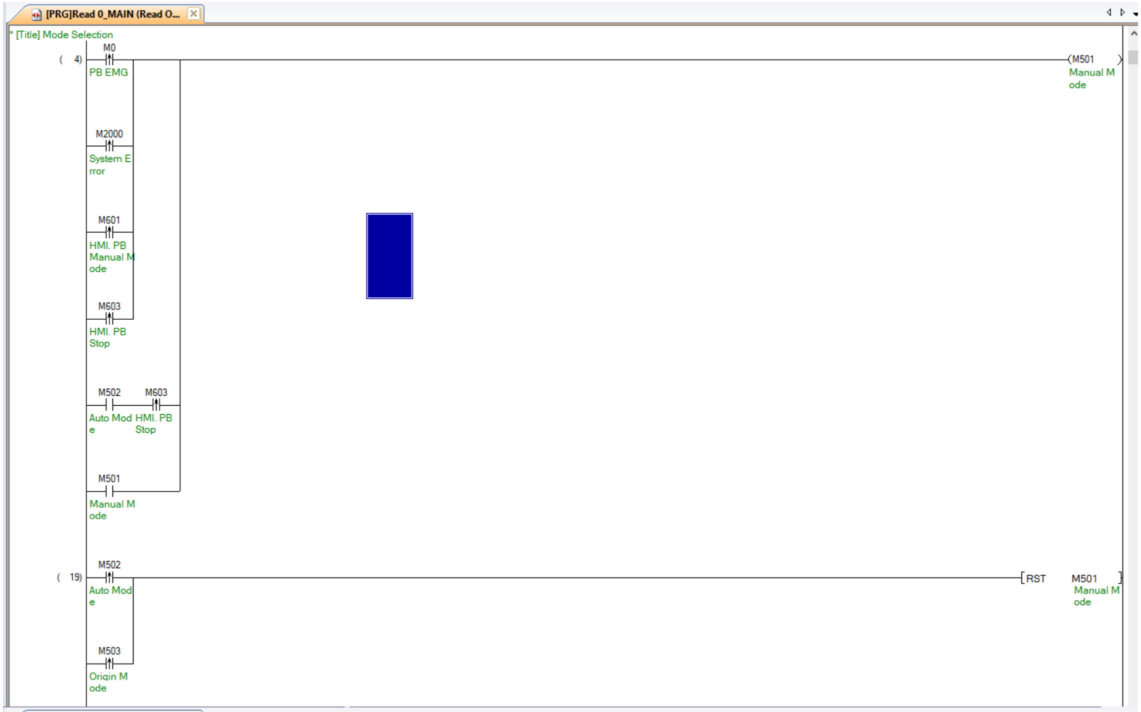
Vintage Machinery. (2011, September 14). *Sprague electric railway & motor Co.* Retrieved November 2019, from Vintage Machinery: http://vintagemachinery.org/mfgindex/imagedetail.aspx?id=4107

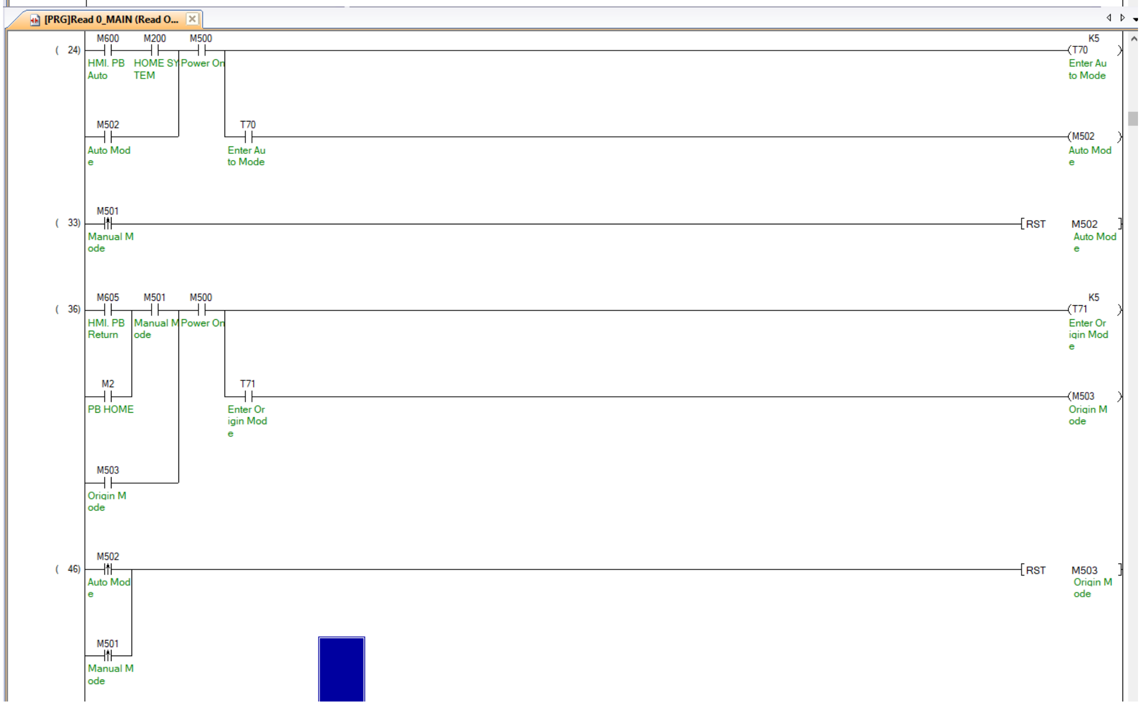
Appendix 1

Programming code

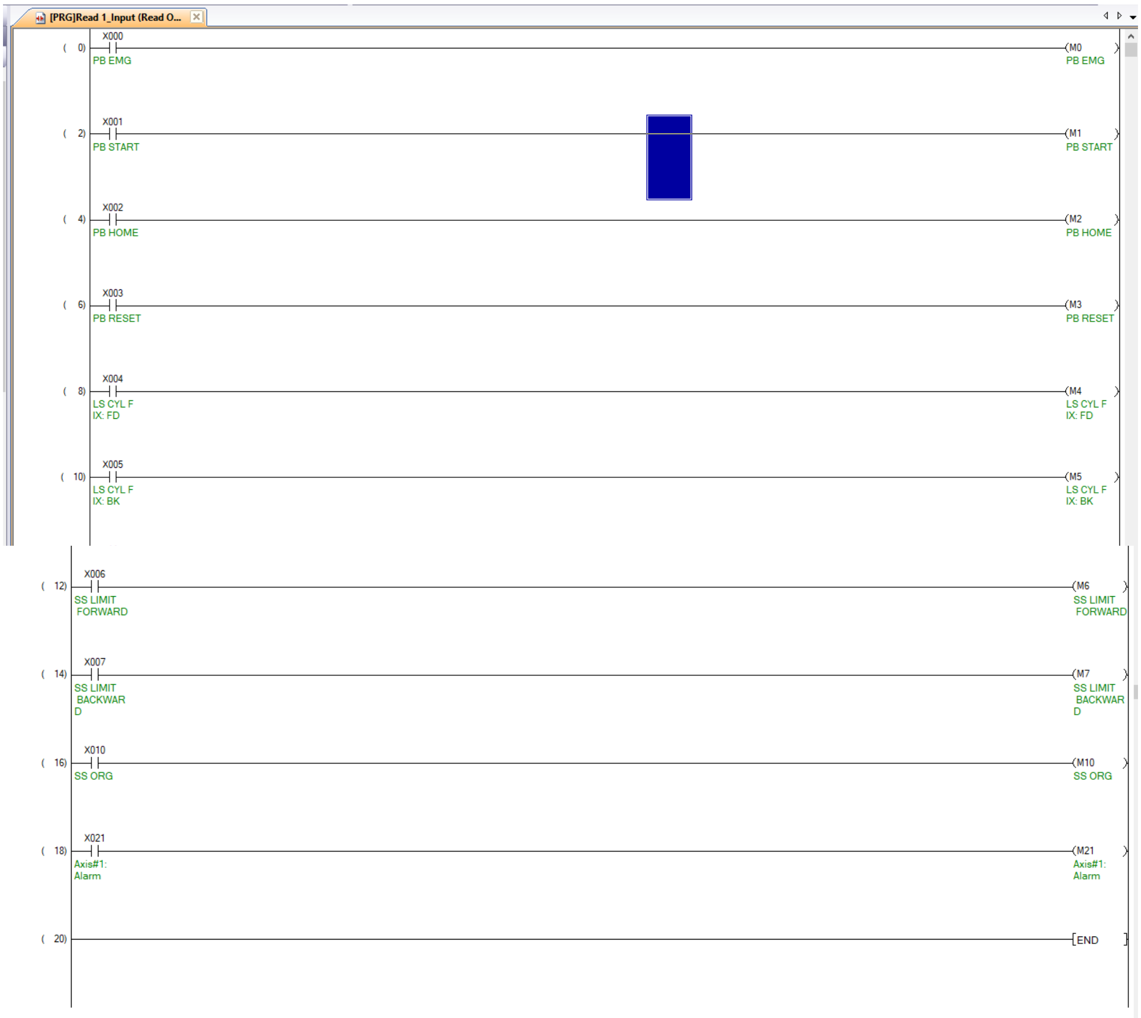
I. PLC

- Main

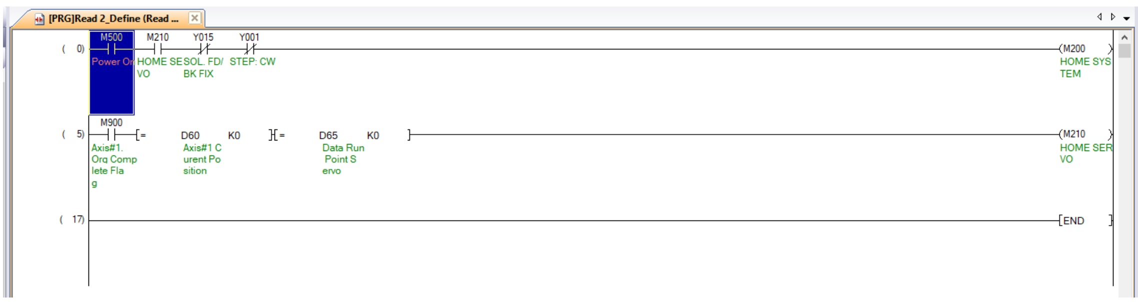




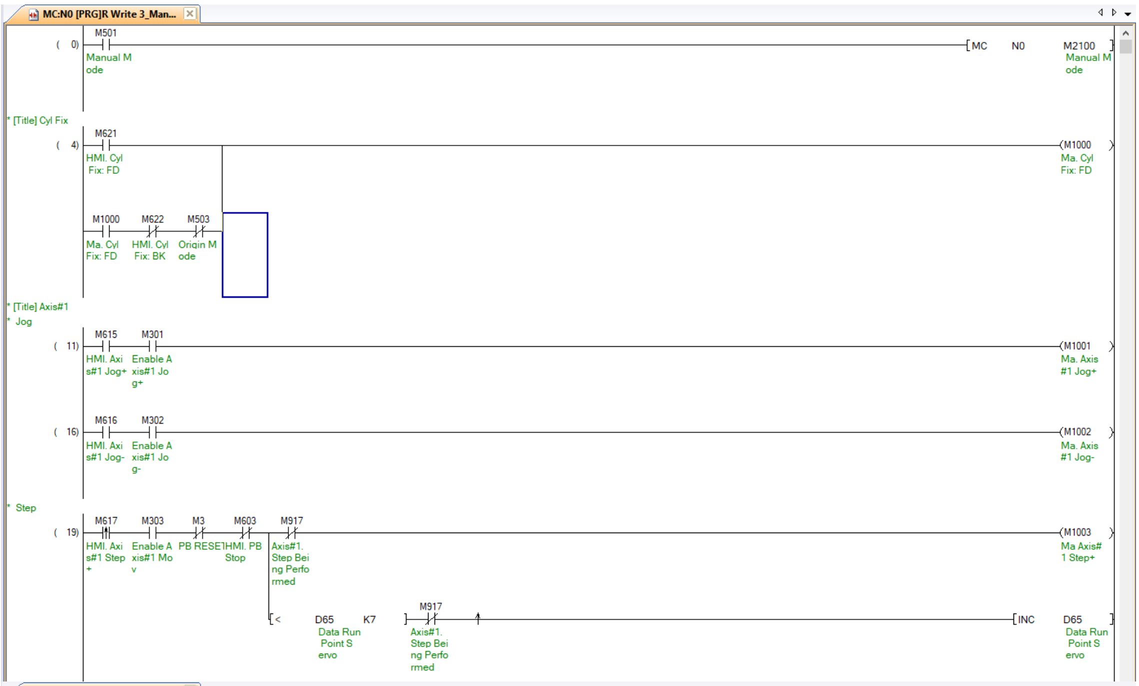
- Input

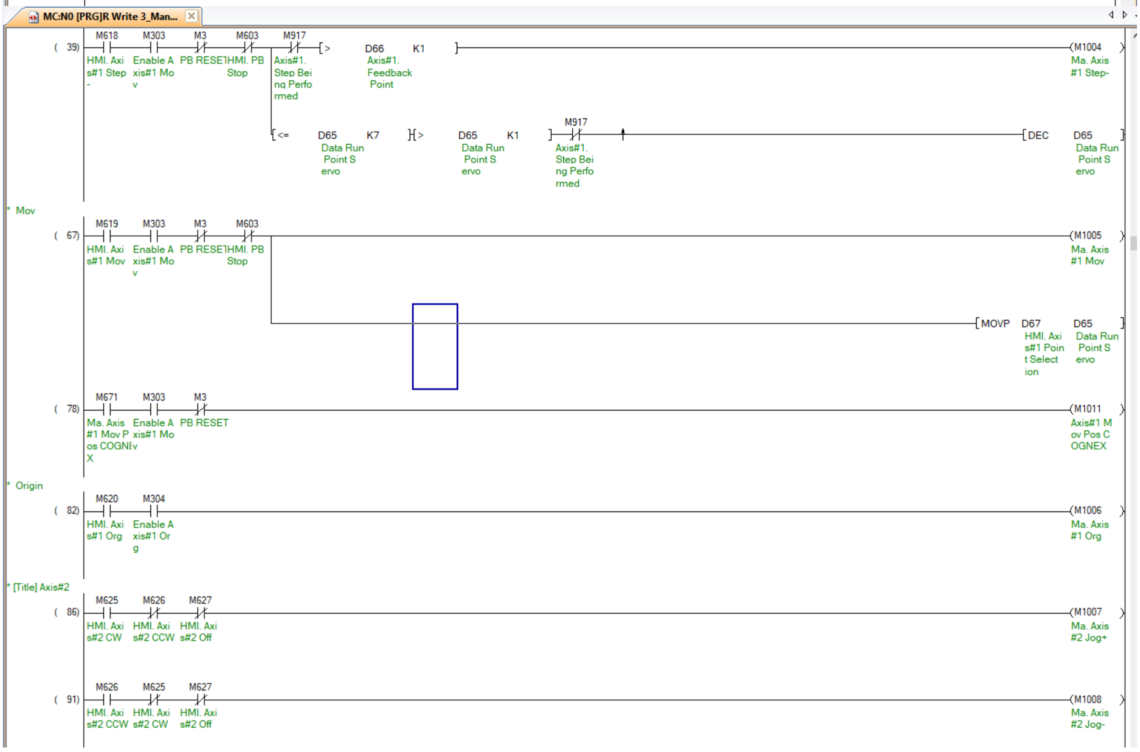


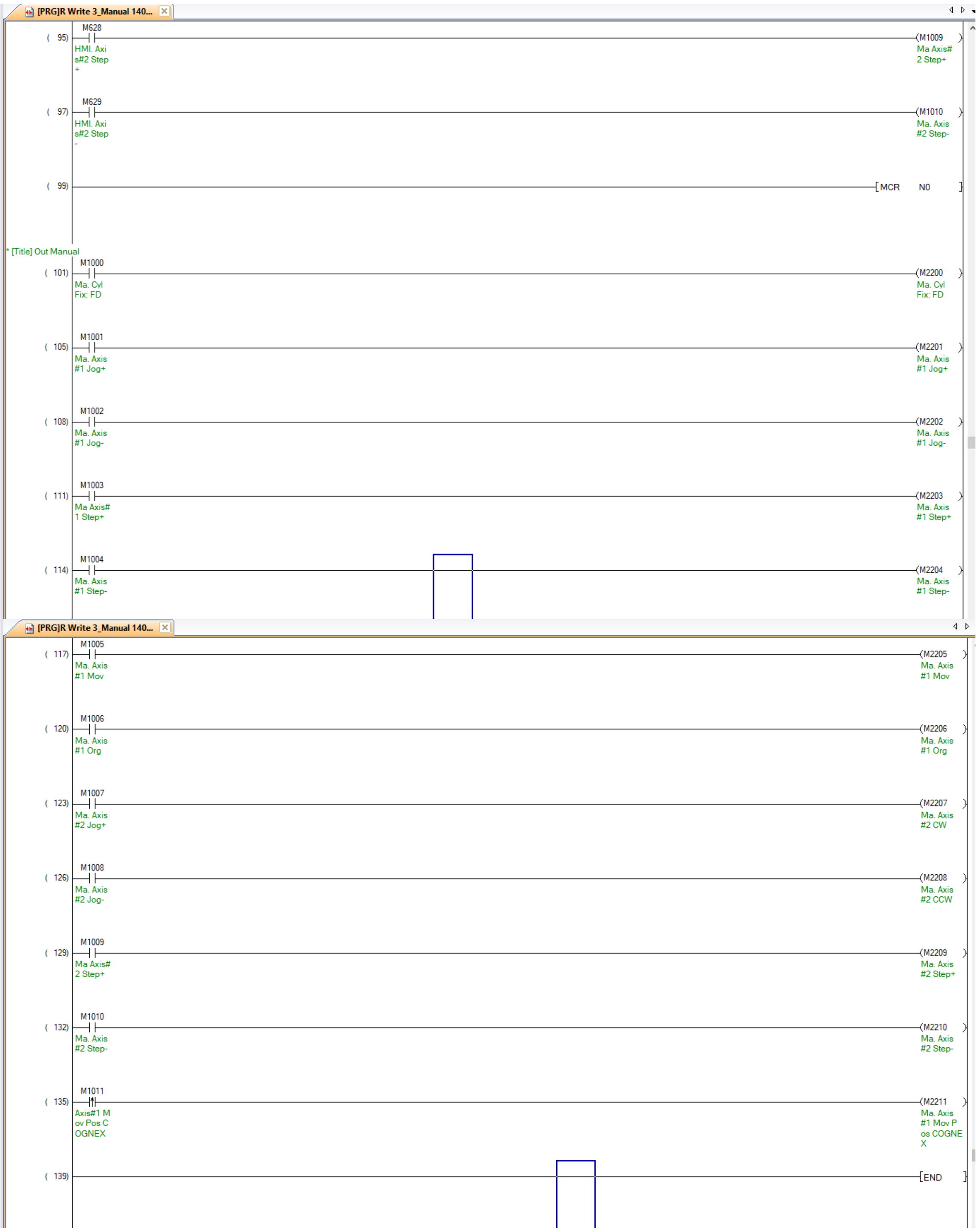
- Define



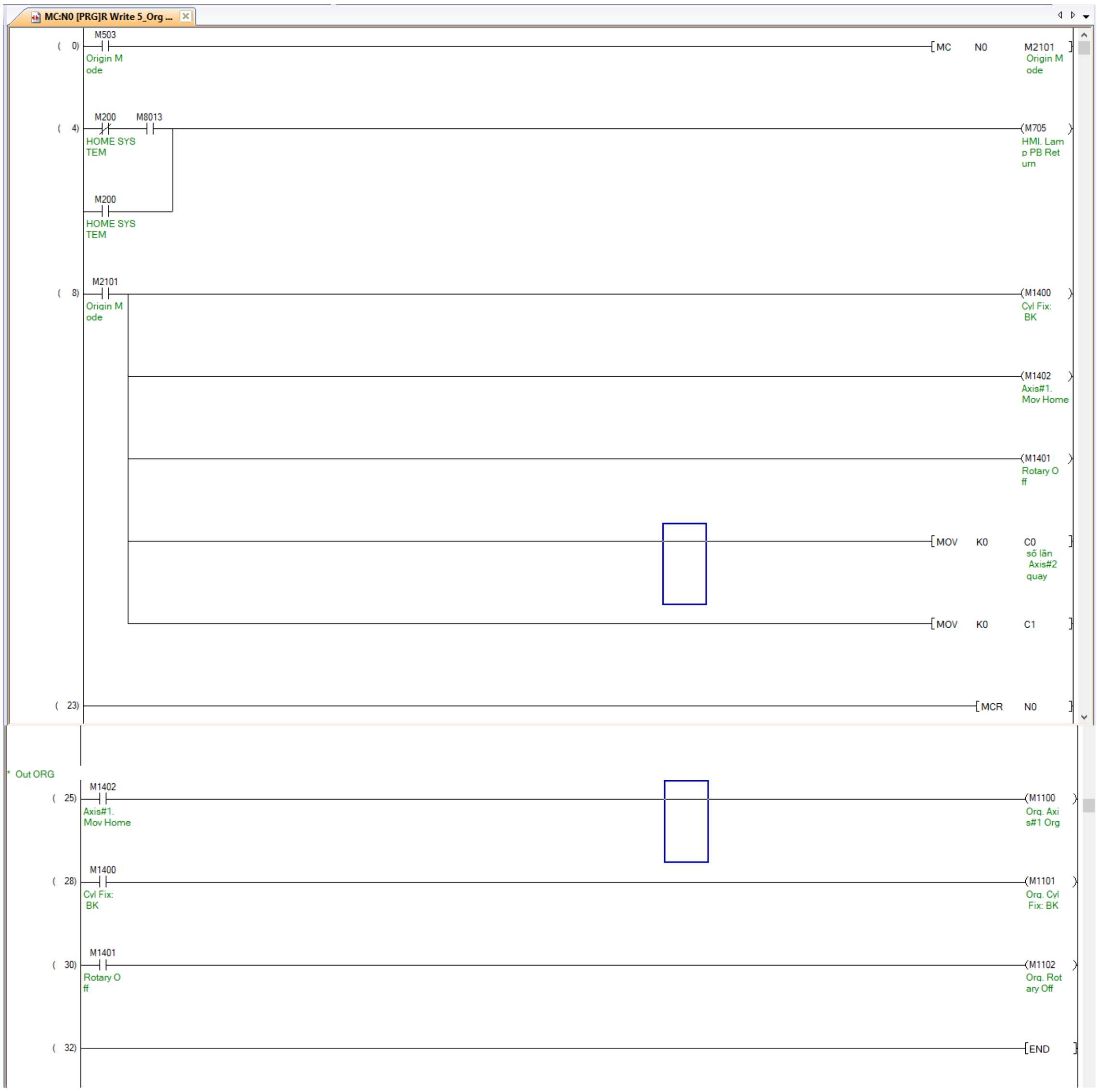
- Manual


MC:N0 [PRG]R Write 3_Man...
M501 Manual Mode
MC N0 M2100 Manual Mode
[Title] Cyl Fix
M621 HMI. Cyl Fix: FD
M1000 Ma. Cyl Fix: FD
M1000 M622 M503
Ma. Cyl Fix: FD HMI. Cyl Fix: BK Origin Mode
[Title] Axis#1
Jog
M615 M301 HMI. Axis#1 Jog+ Enable Axis#1 Jog+
M1001 Ma. Axis #1 Jog+
M616 M302 HMI. Axis#1 Jog- Enable Axis#1 Jog-
M1002 Ma. Axis #1 Jog-
Step
M617 M303 M3 M603 M917
HMI. Axis#1 Step+ Enable Axis#1 Mov PB RESET HMI. PB Stop Axis#1. Step Being Performed
M1003 Ma Axis#1 Step+
< D65 K7 Data Run Point Servo
M917 Axis#1. Step Being Performed
INC D65 Data Run Point Servo

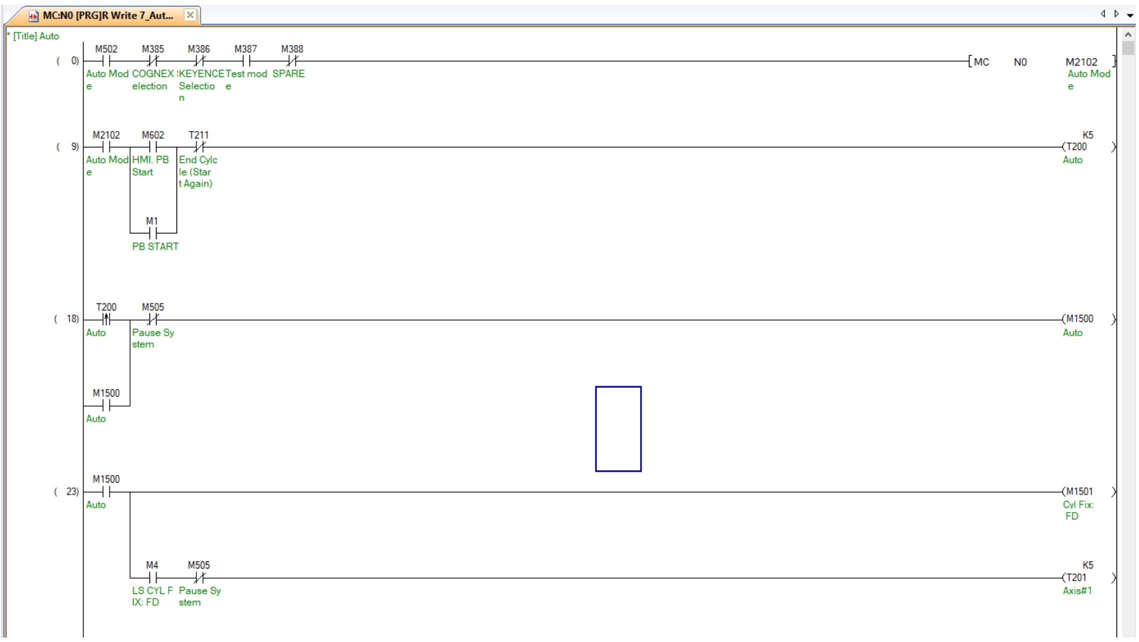

MC:N0 [PRG]R Write 3_Man...
M618 M303 M3 M603 M917
HMI. Axis#1 Step- Enable Axis#1 Mov PB RESET HMI. PB Stop Axis#1. Step Being Performed
> D66 K1 Axis#1. Feedback Point
M1004 Ma. Axis #1 Step-
<= D65 K7 Data Run Point Servo
> D65 K1 Data Run Point Servo
M917 Axis#1. Step Being Performed
DEC D65 Data Run Point Servo
Mov
M619 M303 M3 M603
HMI. Axis#1 Mov Enable Axis#1 Mov PB RESET HMI. PB Stop
M1005 Ma. Axis #1 Mov
MOVP D67 HMI. Axis#1 Point Selection D65 Data Run Point Servo
M671 M303 M3
Ma. Axis #1 Mov Pos COGNIX Enable Axis#1 Mov PB RESET
M1011 Axis#1 Mov Pos COGNEX
Origin
M620 M304 HMI. Axis#1 Org Enable Axis#1 Org
M1006 Ma. Axis #1 Org
[Title] Axis#2
M625 M626 M627 HMI. Axis#2 CW HMI. Axis#2 CCW HMI. Axis#2 Off
M1007 Ma. Axis #2 Jog+
M626 M625 M627 HMI. Axis#2 CCW HMI. Axis#2 CW HMI. Axis#2 Off
M1008 Ma. Axis #2 Jog-

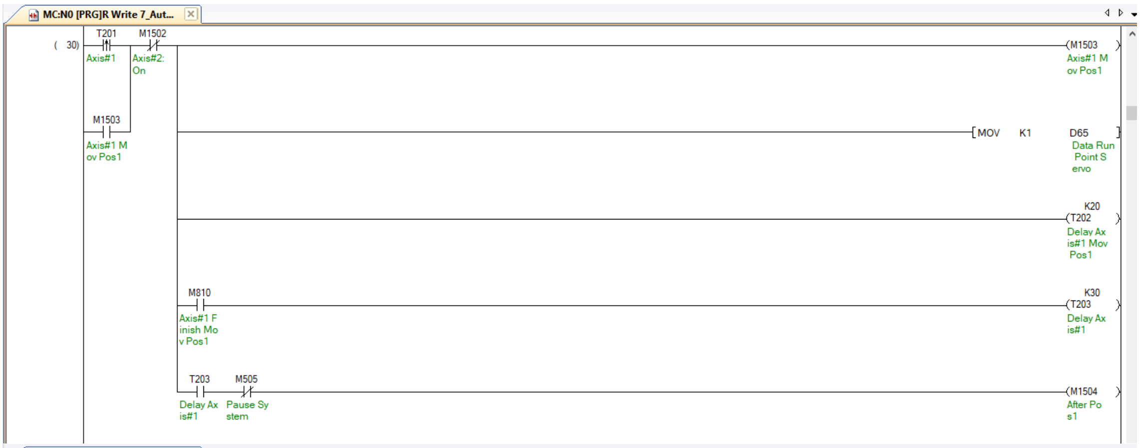

[PRG]R Write 3_Manual 140...
M628
HMI. Axis#2 Step+
M1009
Ma Axis#2 Step+
M629
HMI. Axis#2 Step-
M1010
Ma. Axis#2 Step-
MCR N0
[Title] Out Manual
M1000
Ma. Cyl Fix: FD
M2200
Ma. Cyl Fix: FD
M1001
Ma. Axis#1 Jog+
M2201
Ma. Axis#1 Jog+
M1002
Ma. Axis#1 Jog-
M2202
Ma. Axis#1 Jog-
M1003
Ma Axis#1 Step+
M2203
Ma. Axis#1 Step+
M1004
Ma. Axis#1 Step-
M2204
Ma. Axis#1 Step-
[PRG]R Write 3_Manual 140...
M1005
Ma. Axis#1 Mov
M2205
Ma. Axis#1 Mov
M1006
Ma. Axis#1 Org
M2206
Ma. Axis#1 Org
M1007
Ma. Axis#2 Jog+
M2207
Ma. Axis#2 CW
M1008
Ma. Axis#2 Jog-
M2208
Ma. Axis#2 CCW
M1009
Ma Axis#2 Step+
M2209
Ma. Axis#2 Step+
M1010
Ma. Axis#2 Step-
M2210
Ma. Axis#2 Step-
M1011
Axis#1 Mov Pos COGNEX
M2211
Ma. Axis#1 Mov Pos COGNEX
END

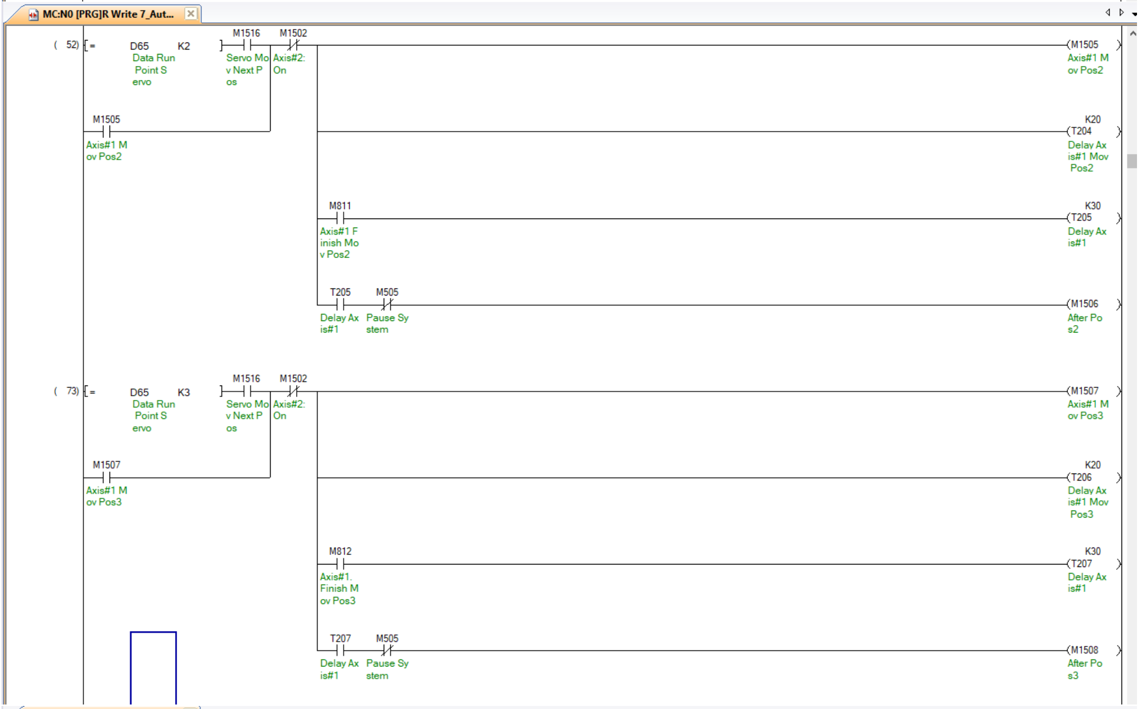
- Org

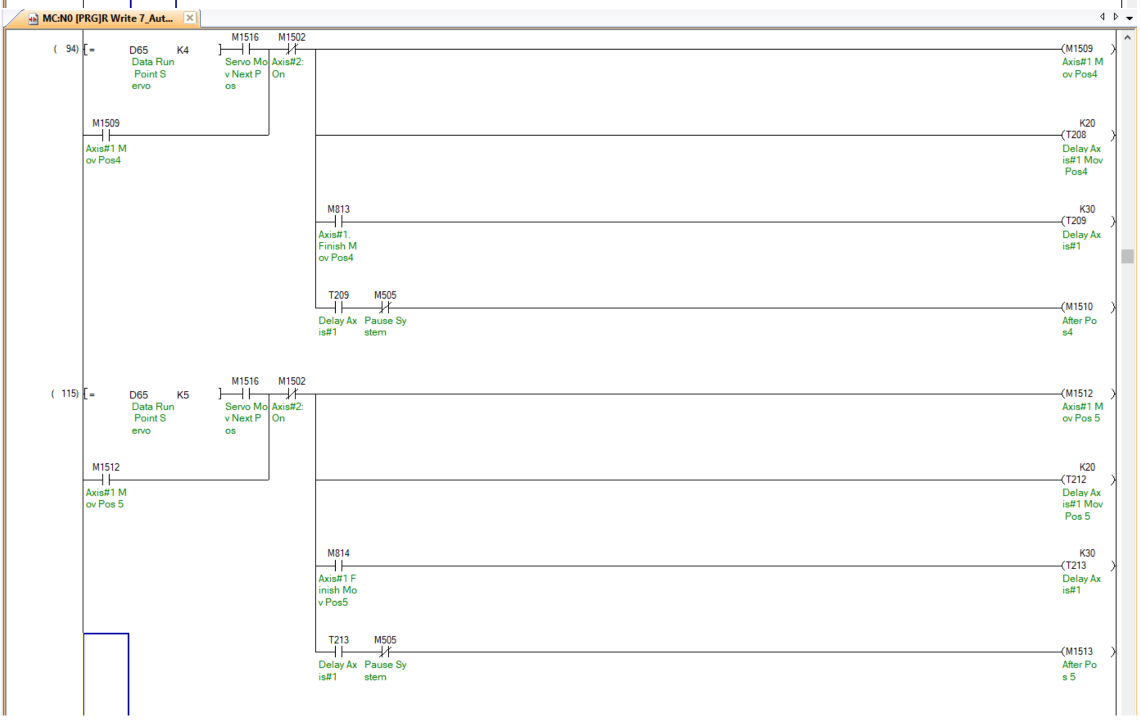


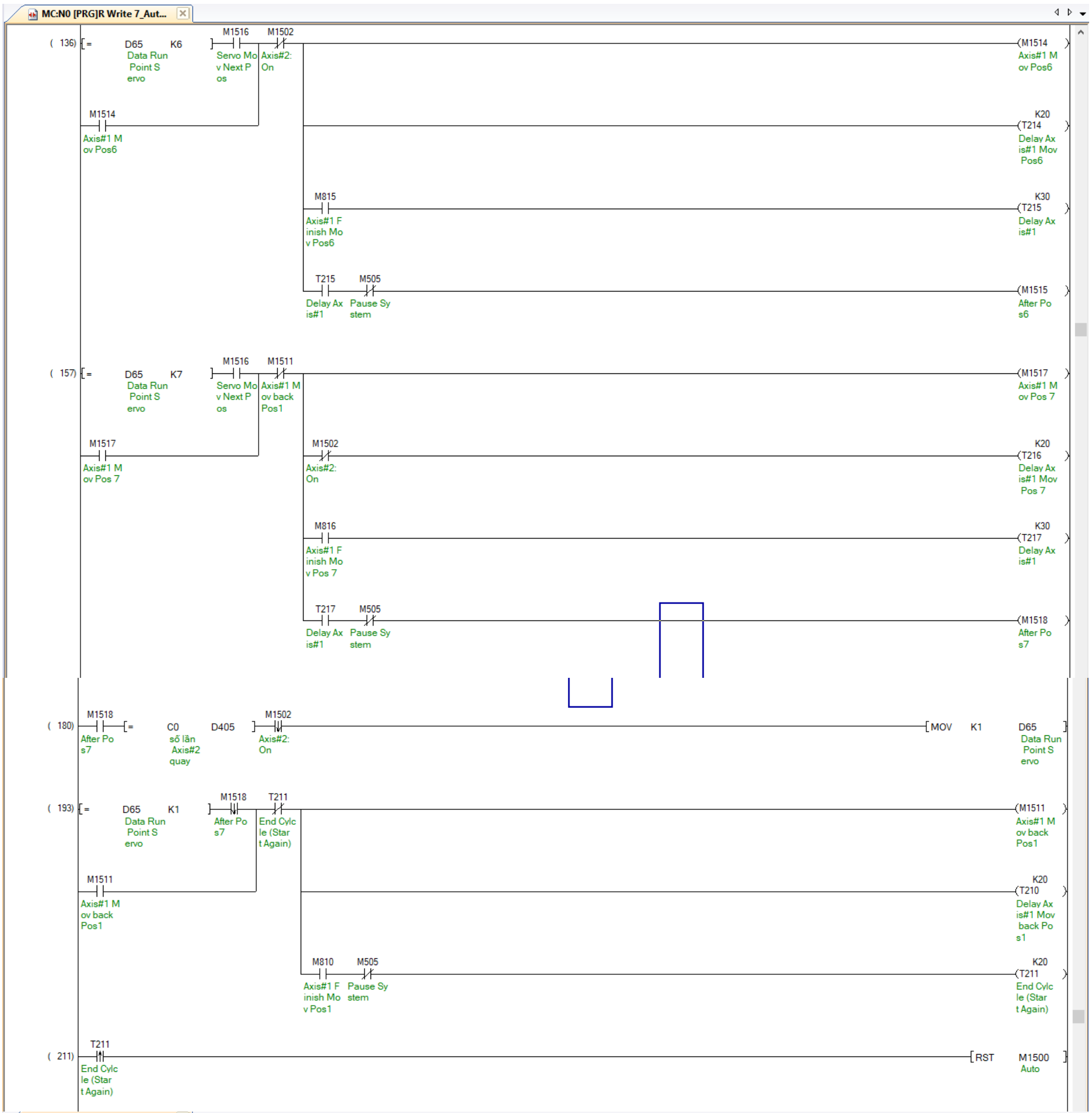
- Auto

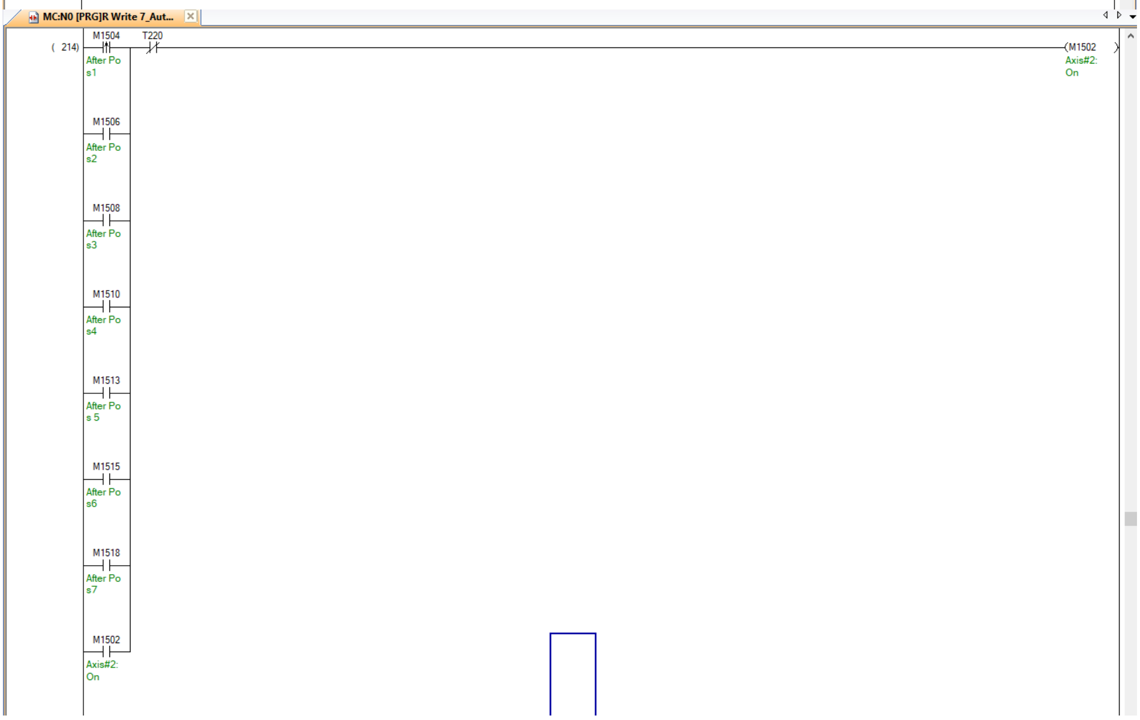


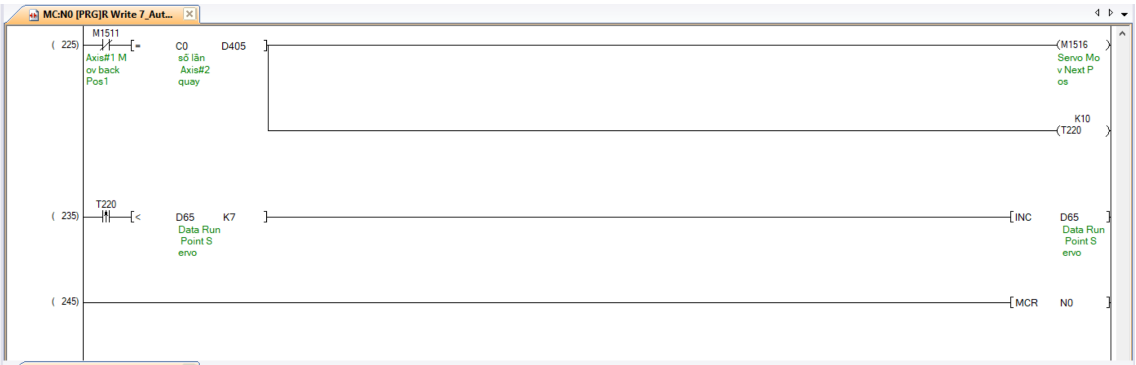

MC:N0 [PRG]R Write 7_Aut...
(30)
T201 Axis#1
M1502 Axis#2: On
M1503 Axis#1 M ov Pos1
MOV K1 D65 Data Run Point S ervo
K20 T202 Delay Ax is#1 Mov Pos1
M810 Axis#1 F inish Mo v Pos1
K30 T203 Delay Ax is#1
T203 Delay Ax is#1
M505 Pause Sy stem
M1504 After Po s1


MC:N0 [PRG]R Write 7_Aut...
(52) =
D65 Data Run Point S ervo
K2
M1516 Servo Mo v Next P os
M1502 Axis#2: On
M1505 Axis#1 M ov Pos2
K20 T204 Delay Ax is#1 Mov Pos2
M811 Axis#1 F inish Mo v Pos2
K30 T205 Delay Ax is#1
T205 Delay Ax is#1
M505 Pause Sy stem
M1506 After Po s2
(73) =
D65 Data Run Point S ervo
K3
M1516 Servo Mo v Next P os
M1502 Axis#2: On
M1507 Axis#1 M ov Pos3
K20 T206 Delay Ax is#1 Mov Pos3
M812 Axis#1. Finish M ov Pos3
K30 T207 Delay Ax is#1
T207 Delay Ax is#1
M505 Pause Sy stem
M1508 After Po s3

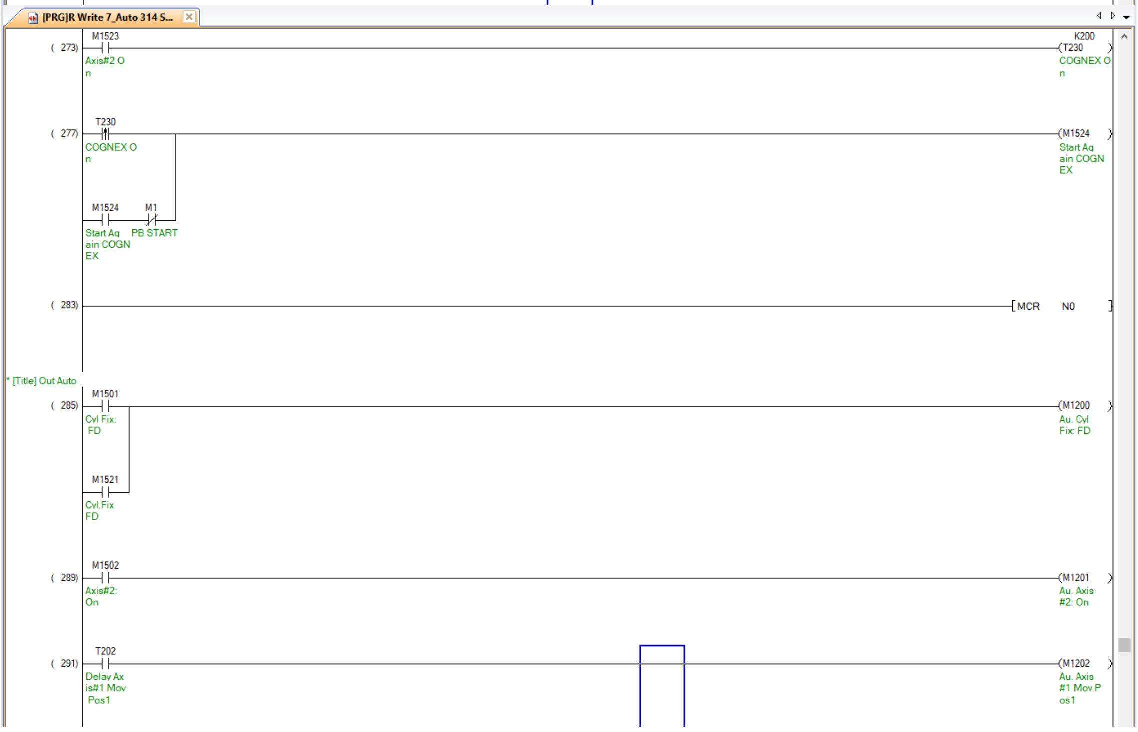

MC:N0 [PRG]R Write 7_Aut...
(94) =
D65 Data Run Point S ervo
K4
M1516 Servo Mo v Next P os
M1502 Axis#2: On
M1509 Axis#1 M ov Pos4
K20 T208 Delay Ax is#1 Mov Pos4
M813 Axis#1. Finish M ov Pos4
K30 T209 Delay Ax is#1
T209 Delay Ax is#1
M505 Pause Sy stem
M1510 After Po s4
(115) =
D65 Data Run Point S ervo
K5
M1516 Servo Mo v Next P os
M1502 Axis#2: On
M1512 Axis#1 M ov Pos 5
K20 T212 Delay Ax is#1 Mov Pos 5
M814 Axis#1 F inish Mo v Pos5
K30 T213 Delay Ax is#1
T213 Delay Ax is#1
M505 Pause Sy stem
M1513 After Po s 5

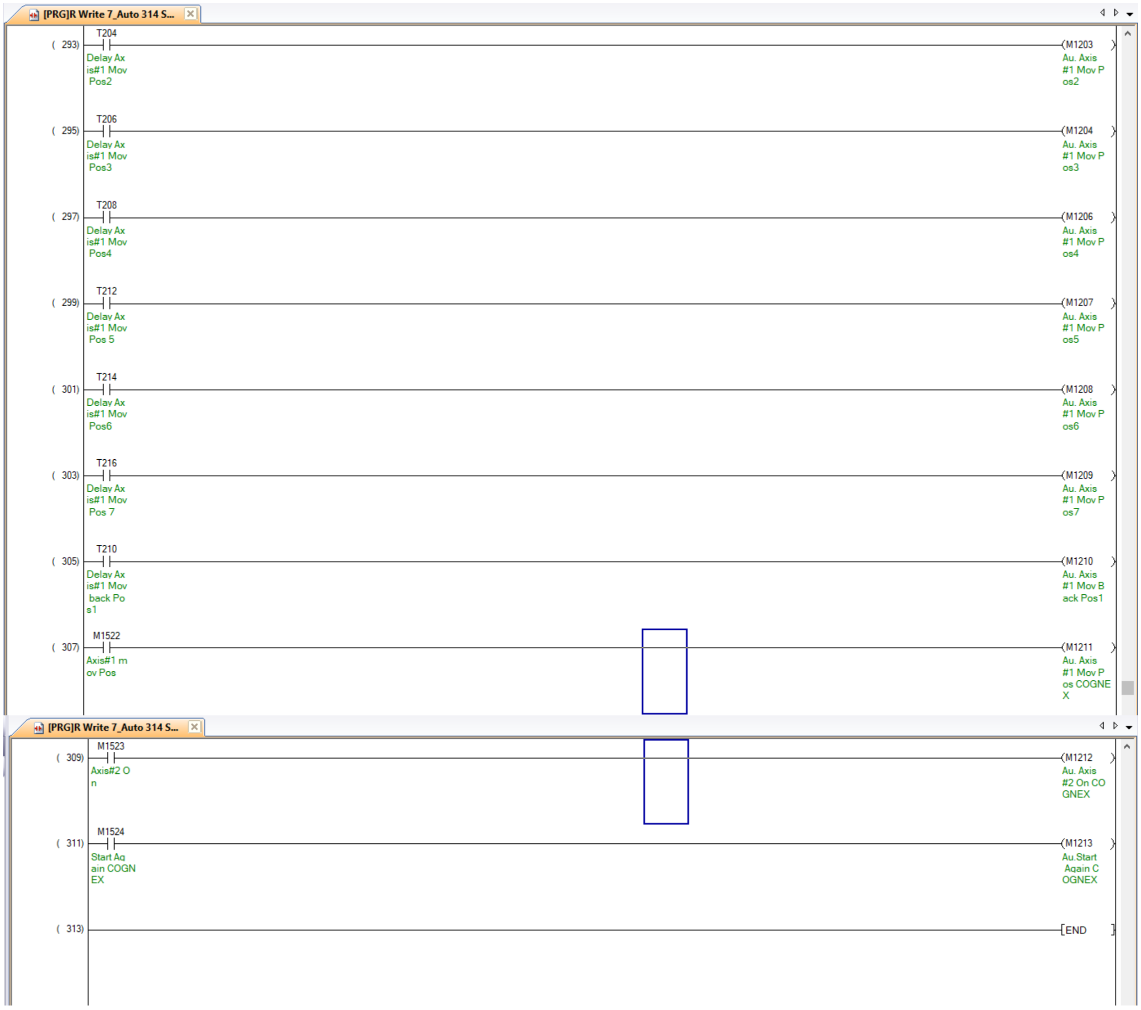

MC:N0 [PRG]R Write 7_Aut...
(136) D65 Data Run Point Servo K6 M1516 Servo Mov Next Pos M1502 Axis#2: On M1514 Axis#1 Mov Pos6
M1514 Axis#1 Mov Pos6 K20 T214 Delay Axis#1 Mov Pos6
M815 Axis#1 Finish Mov Pos6 K30 T215 Delay Axis#1
T215 Delay Axis#1 M505 Pause System M1515 After Pos6
(157) D65 Data Run Point Servo K7 M1516 Servo Mov Next Pos M1511 Axis#1 Mov back Pos1 M1517 Axis#1 Mov Pos 7
M1517 Axis#1 Mov Pos 7 M1502 Axis#2: On K20 T216 Delay Axis#1 Mov Pos 7
M816 Axis#1 Finish Mov Pos 7 K30 T217 Delay Axis#1
T217 Delay Axis#1 M505 Pause System M1518 After Pos7
(180) M1518 After Pos7 C0 số lần Axis#2 quay D405 M1502 Axis#2: On MOV K1 D65 Data Run Point Servo
(193) D65 Data Run Point Servo K1 M1518 After Pos7 T211 End Cycle (Start Again) M1511 Axis#1 Mov back Pos1
M1511 Axis#1 Mov back Pos1 K20 T210 Delay Axis#1 Mov back Pos1
M810 Axis#1 Finish Mov Pos1 M505 Pause System K20 T211 End Cycle (Start Again)
(211) T211 End Cycle (Start Again) RST M1500 Auto


MC:N0 [PRG]R Write 7_Aut...
(214) M1504 After Pos1 T220 M1502 Axis#2: On
M1506 After Pos2
M1508 After Pos3
M1510 After Pos4
M1513 After Pos 5
M1515 After Pos6
M1518 After Pos7
M1502 Axis#2: On

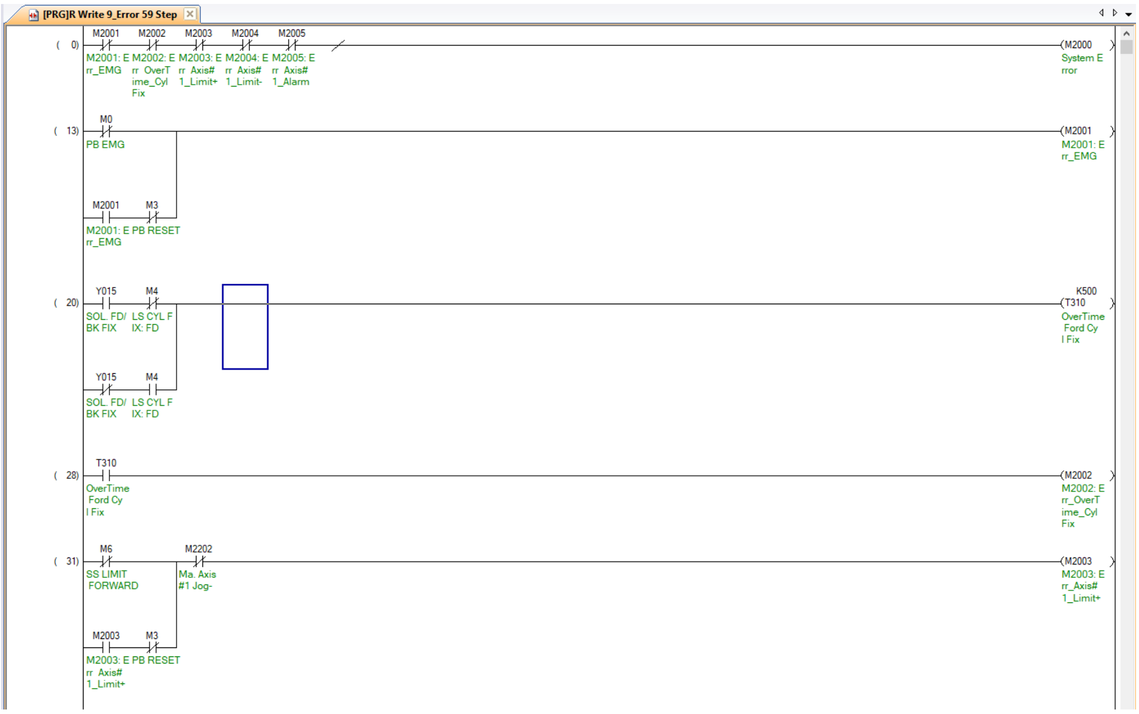

MC:N0 [PRG]R Write 7_Aut...
(225)
M1511
Axis#1 M ov back Pos1
=
C0
số lần Axis#2 quay
D405
M1516
Servo Mo v Next P os
K10
T220
(235)
T220
<
D65
Data Run Point S ervo
K7
INC
D65
Data Run Point S ervo
(245)
MCR
N0

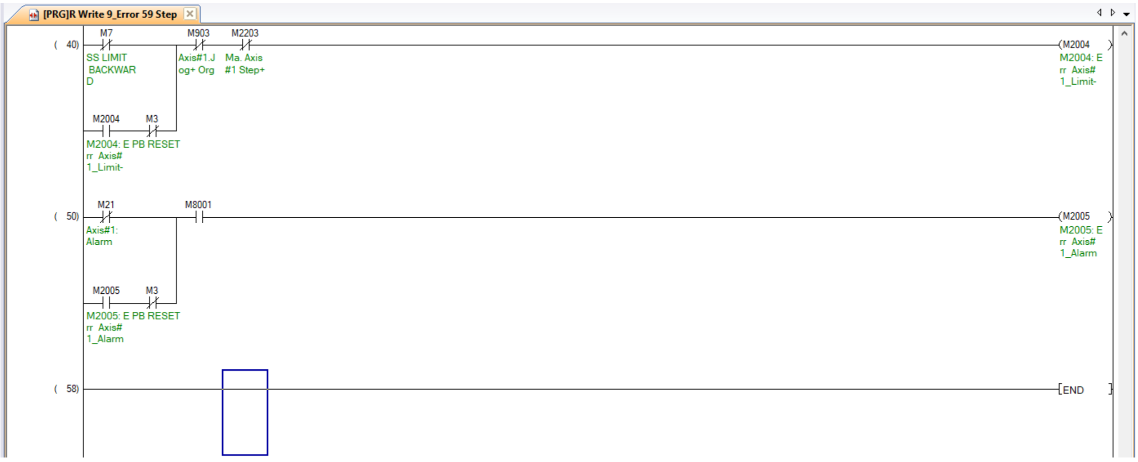

MC:N0 [PRG]R Write 7_Aut...
* [Title] Camera COGNEX
(247)
M502
Auto Mod e
M765
Lamp Sel ect COGN EX
M386
KEYENCE Selectio n
M387
Test mod e
M388
SPARE
MC
N0
M2103
COGNEX
(256)
M1
PB START
M2103
COGNEX
T230
COGNEX O n
M1520
COGNEX A uto
M602
HMI. PB Start
M1520
COGNEX A uto
(263)
M1520
COGNEX uto
T230
COGNEX O n
M1521
Cyl.Fix FD
M1521
Cyl.Fix FD
(267)
M4
LS CYL F IX: FD
M1521
Cyl.Fix FD
M1522
Axis#1 m ov Pos
(270)
M820
Axis#1 F inish Mo v Pos CO GNEX
M4
LS CYL F IX: FD
M1523
Axis#2 O n


[PRG]R Write 7_Auto 314 S...
(273)
M1523
Axis#2 O n
K200
T230
COGNEX O n
(277)
T230
COGNEX O n
M1524
Start Aq ain COGN EX
M1524
Start Aq ain COGN EX
M1
PB START
(283)
MCR
N0
* [Title] Out Auto
(285)
M1501
Cyl Fix: FD
M1200
Au. Cyl Fix: FD
M1521
Cyl.Fix FD
(289)
M1502
Axis#2: On
M1201
Au. Axis #2: On
(291)
T202
Delay Ax is#1 Mov Pos1
M1202
Au. Axis #1 Mov P os1



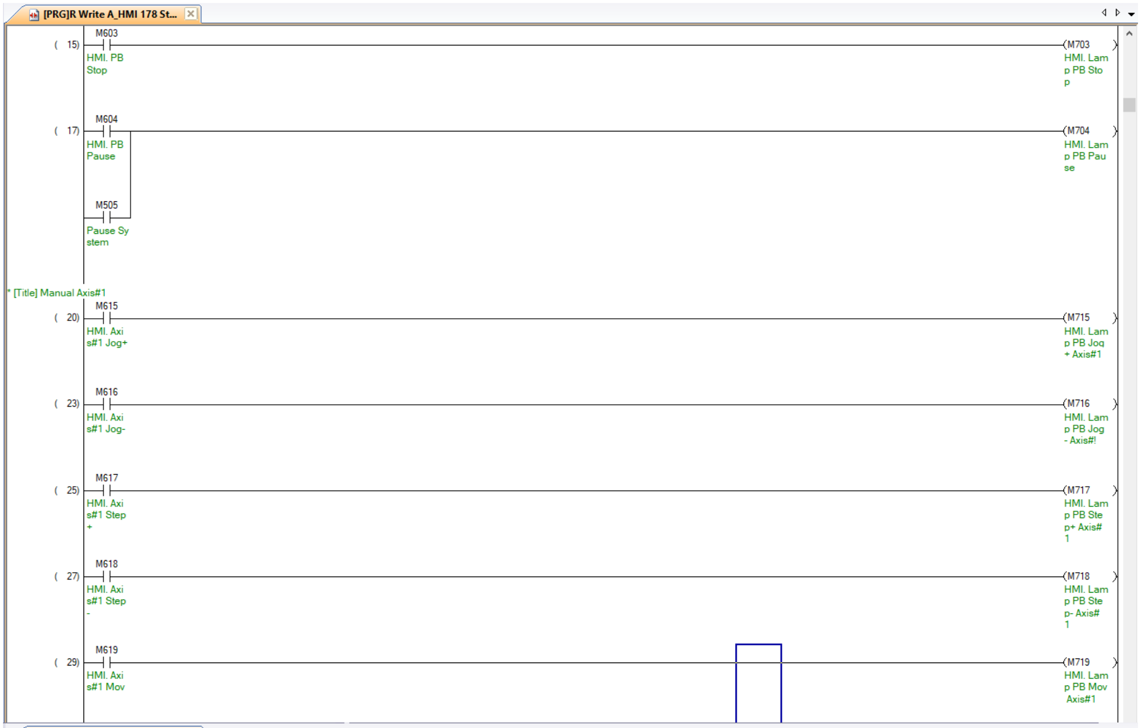
- Error

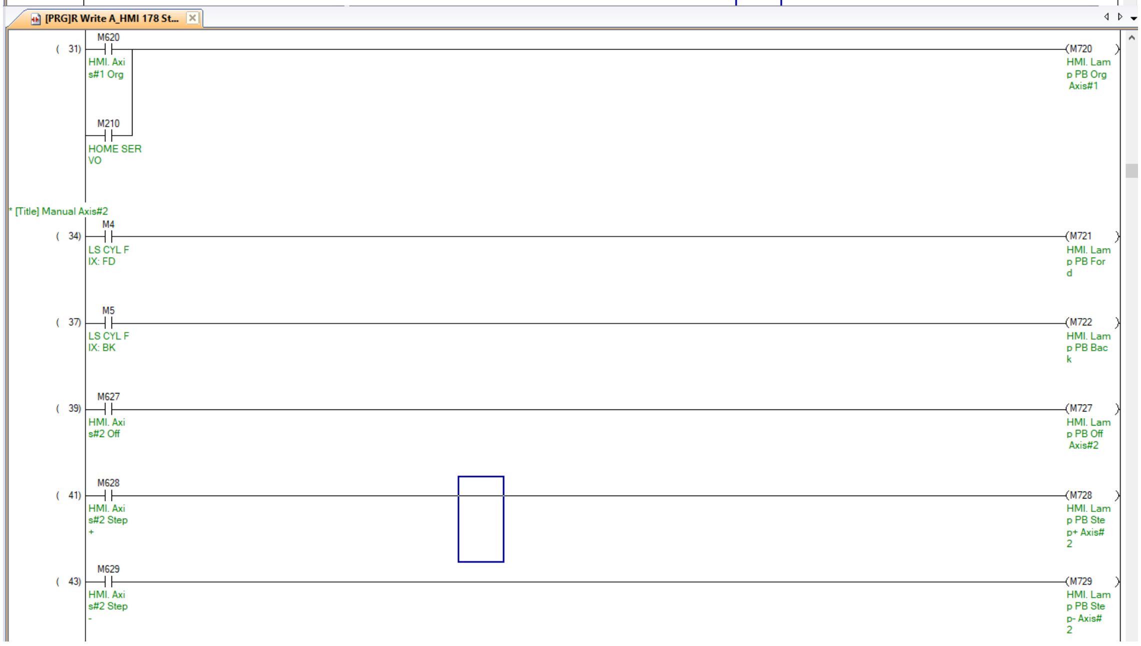



[PRG]R Write 9_Error 59 Step
M7
SS LIMIT BACKWARD
M903
Axis#1.Jog+ Org
M2203
Ma. Axis #1 Step+
M2004
M2004: Err Axis#1_Limit-
M2004
M2004: Err Axis#1_Limit-
M3
PB RESET
M21
Axis#1: Alarm
M8001
M2005
M2005: Err Axis#1_Alarm
M2005
M2005: Err Axis#1_Alarm
M3
PB RESET
END

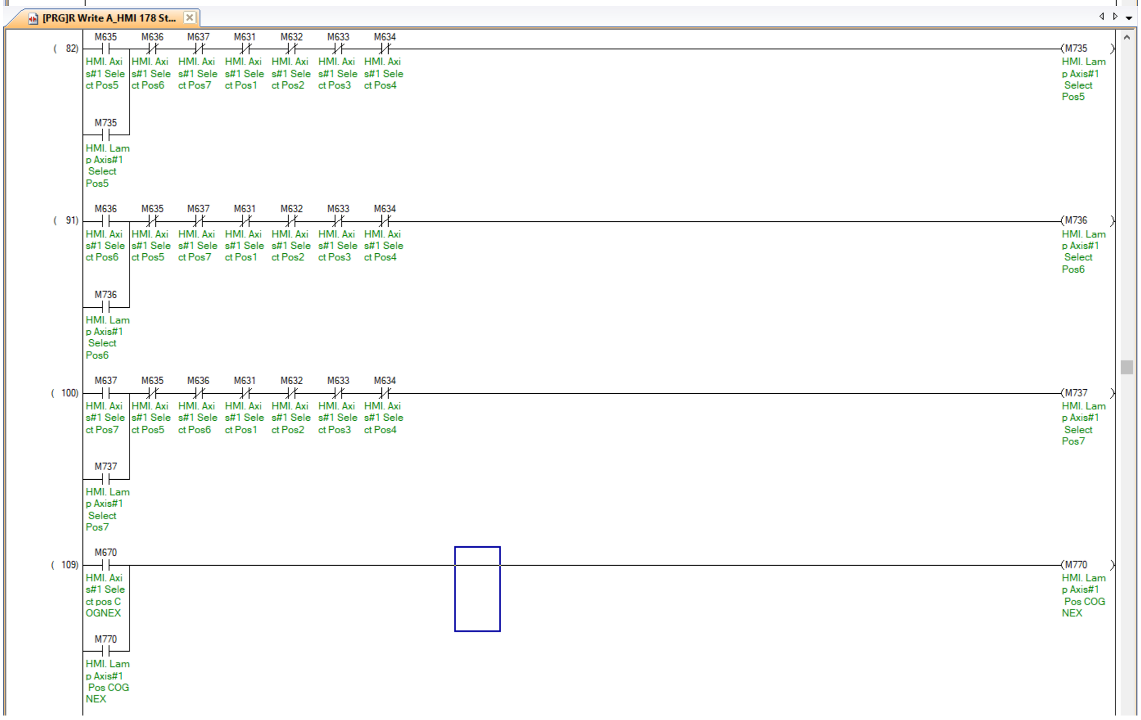
- HMI

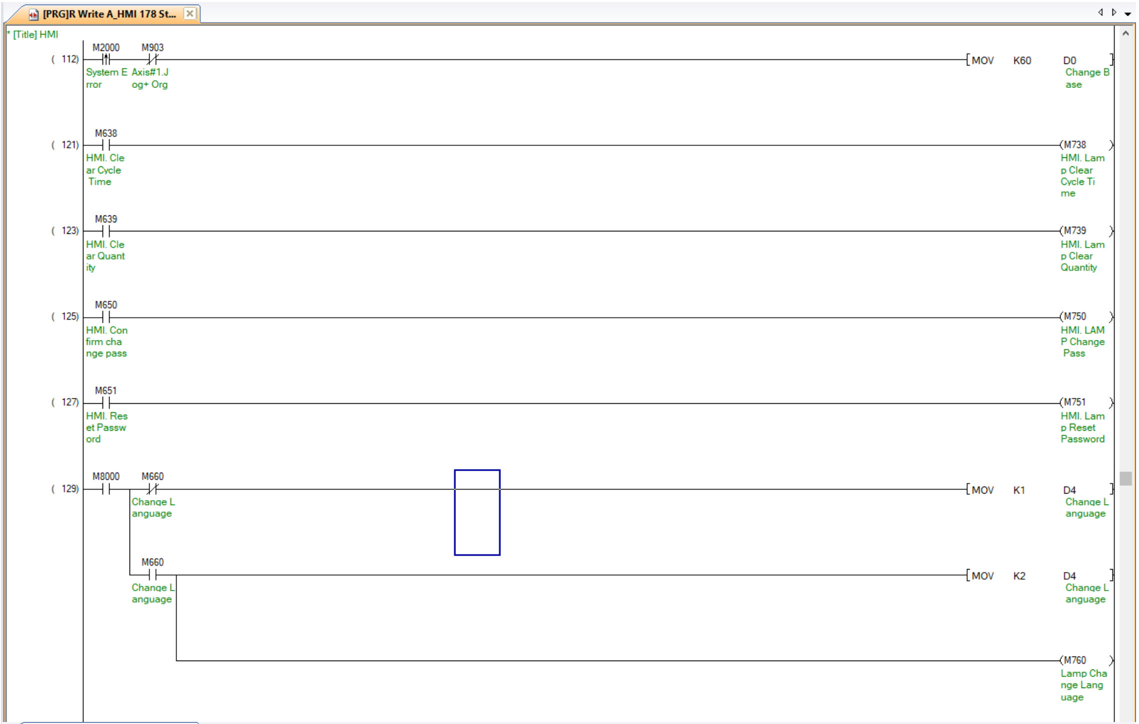

[PRG]R Write A_HMI 178 St...
[Title] Virtual PB
M600
HMI. PB Auto
M502
Auto Mode
M700
HMI Lamp PB Auto
M601
HMI. PB Manual Mode
M501
Manual Mode
M701
HMI Lamp PB Manual
M602
HMI. PB Start
M502
Auto Mode
M8013
M4
LS CYL FIX: FD
M502
Auto Mode
M702
HMI Lamp PB Start


[PRG]R Write A_HMI 178 St...
(15) M603 HMI. PB Stop
M703 HMI. Lamp PB Stop
(17) M604 HMI. PB Pause
M505 Pause System
M704 HMI. Lamp PB Pause
* [Title] Manual Axis#1
(20) M615 HMI. Axis#1 Jog+
M715 HMI. Lamp PB Jog + Axis#1
(23) M616 HMI. Axis#1 Jog-
M716 HMI. Lamp PB Jog - Axis#1
(25) M617 HMI. Axis#1 Step +
M717 HMI. Lamp PB Step+ Axis# 1
(27) M618 HMI. Axis#1 Step -
M718 HMI. Lamp PB Step- Axis# 1
(29) M619 HMI. Axis#1 Mov
M719 HMI. Lamp PB Mov Axis#1

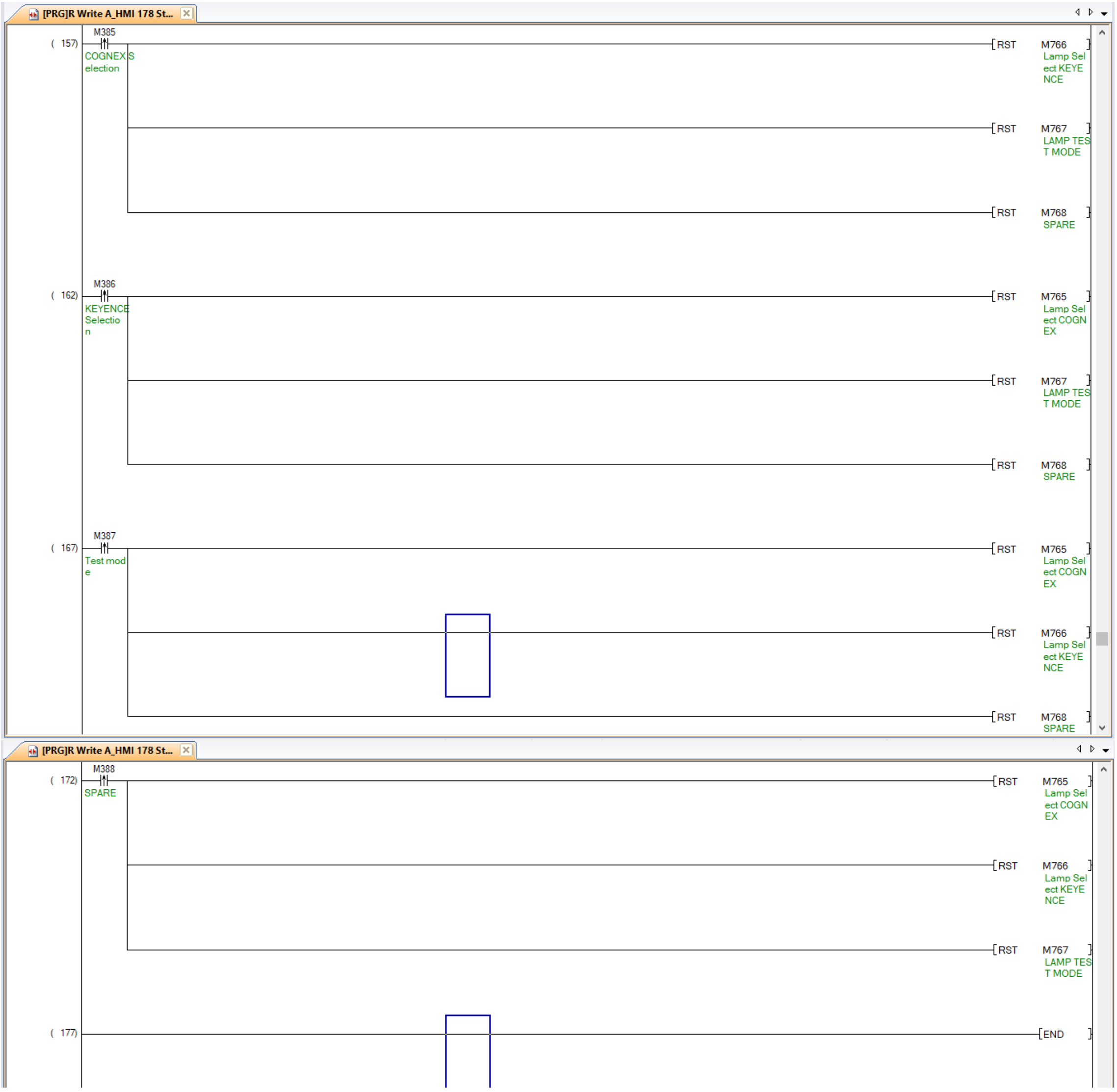

[PRG]R Write A_HMI 178 St...
(31) M620 HMI. Axis#1 Org
M210 HOME SERVO
M720 HMI. Lamp PB Org Axis#1
* [Title] Manual Axis#2
(34) M4 LS CYL FIX: FD
M721 HMI. Lamp PB Ford
(37) M5 LS CYL FIX: BK
M722 HMI. Lamp PB Back
(39) M627 HMI. Axis#2 Off
M727 HMI. Lamp PB Off Axis#2
(41) M628 HMI. Axis#2 Step +
M728 HMI. Lamp PB Step+ Axis# 2
(43) M629 HMI. Axis#2 Step -
M729 HMI. Lamp PB Step- Axis# 2


[PRG]R Write A_HMI 178 St...
[Title] Axis#1 Select Pos
(45)
M631 HMI. Axis#1 Select Pos1
M632 HMI. Axis#1 Select Pos2
M633 HMI. Axis#1 Select Pos3
M634 HMI. Axis#1 Select Pos4
M635 HMI. Axis#1 Select Pos5
M636 HMI. Axis#1 Select Pos6
M637 HMI. Axis#1 Select Pos7
M731 HMI. Lamp Axis#1 Select Pos1
(55)
M632 M631 M633 M634 M635 M636 M637
M732 HMI. Lamp Axis#1 Select Pos2
(64)
M633 M631 M632 M634 M635 M636 M637
M733 HMI. Lamp Axis#1 Select Pos3
(73)
M634 M631 M632 M633 M635 M636 M637
M734 HMI. Lamp Axis#1 Select Pos4


[PRG]R Write A_HMI 178 St...
(82)
M635 M636 M637 M631 M632 M633 M634
M735 HMI. Lamp Axis#1 Select Pos5
(91)
M636 M635 M637 M631 M632 M633 M634
M736 HMI. Lamp Axis#1 Select Pos6
(100)
M637 M635 M636 M631 M632 M633 M634
M737 HMI. Lamp Axis#1 Select Pos7
(109)
M670 HMI. Axis#1 Select pos COGNEX
M770 HMI. Lamp Axis#1 Pos COGNEX

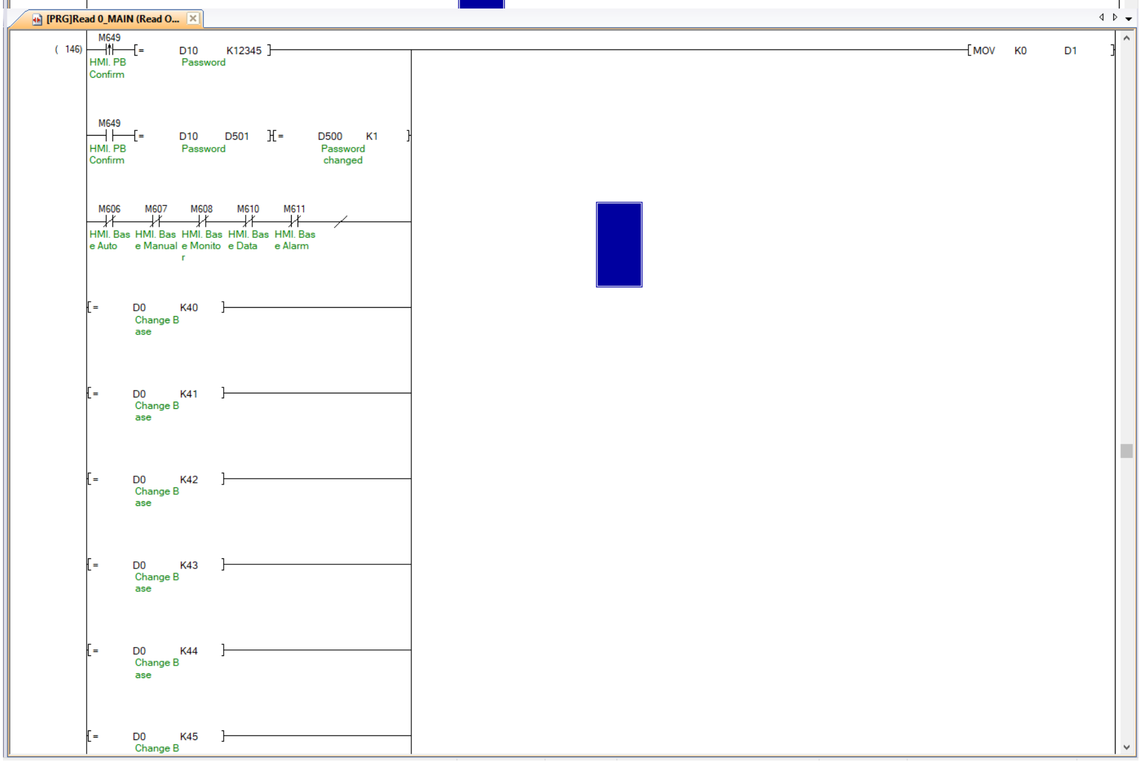

[PRG]R Write A_HMI 178 St...
* [Title] HMI
(112) M2000 System Error M903 Axis#1.Jog+ Org [MOV K60 D0 Change Base]
(121) M638 HMI. Clear Cycle Time (M738 HMI. Lamp Clear Cycle Time)
(123) M639 HMI. Clear Quantity (M739 HMI. Lamp Clear Quantity)
(125) M650 HMI. Confirm change pass (M750 HMI. LAMP Change Pass)
(127) M651 HMI. Reset Password (M751 HMI. Lamp Reset Password)
(129) M8000 M660 Change Language [MOV K1 D4 Change Language]
M660 Change Language [MOV K2 D4 Change Language]
(M760 Lamp Change Language)

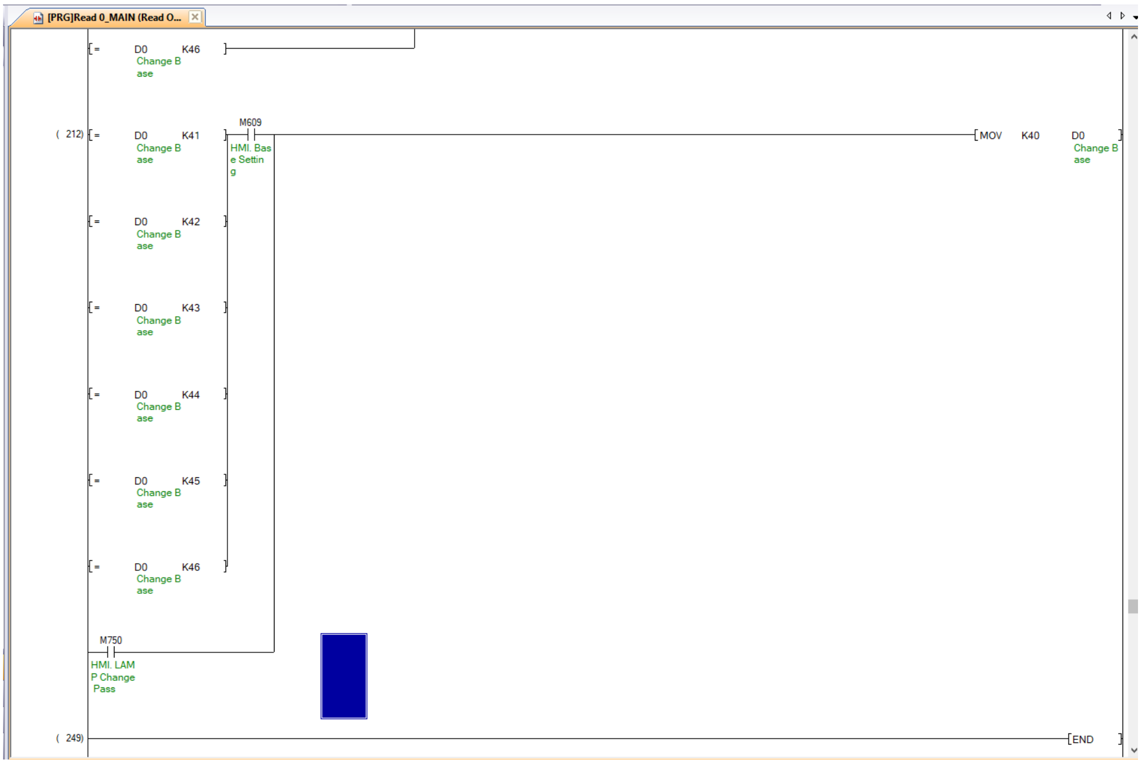

[PRG]R Write A_HMI 178 St...
(145) M385 COGNEX Selection (M765 Lamp Select COGNEX)
M765 Lamp Select COGNEX
(148) M386 KEYENCE Selection (M766 Lamp Select KEYENCE)
M766 Lamp Select KEYENCE
(151) M387 Test mode (M767 LAMP TEST MODE)
M767 LAMP TEST MODE
(154) M388 SPARE (M768 SPARE)
M768 SPARE


[PRG]R Write A_HMI 178 St...
(157)
M385
COGNEX Selection
RST M766 Lamp Select KEYENCE
RST M767 LAMP TEST MODE
RST M768 SPARE
(162)
M386
KEYENCE Selection
RST M765 Lamp Select COGNEX
RST M767 LAMP TEST MODE
RST M768 SPARE
(167)
M387
Test mode
RST M765 Lamp Select COGNEX
RST M766 Lamp Select KEYENCE
RST M768 SPARE
[PRG]R Write A_HMI 178 St...
(172)
M388
SPARE
RST M765 Lamp Select COGNEX
RST M766 Lamp Select KEYENCE
RST M767 LAMP TEST MODE
(177)
END

- Password

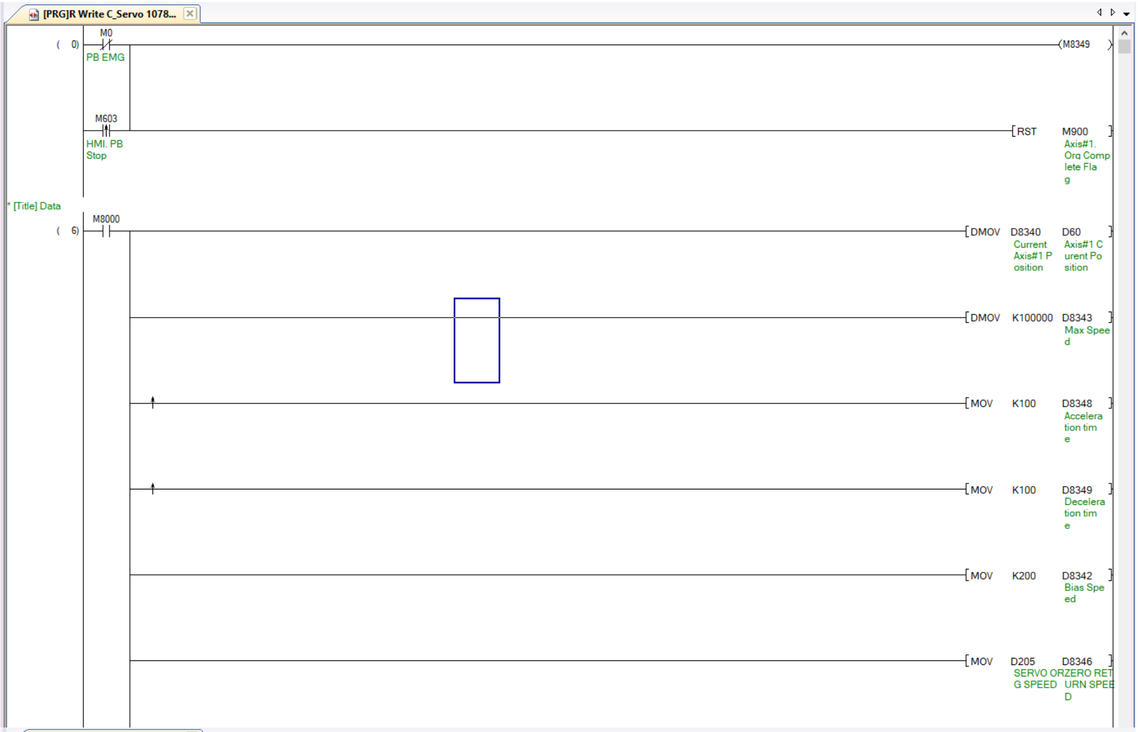

[PRG]Read 0_MAIN (Read O...
[Title] Password
(58)
M649
HMI. PB Confirm
[= D10 K12345][<> D500 K1]
Password
Password changed
[MOV K40 D0]
Change Base
[= D500 K1][= D10 D501]
Password changed
Password
(88)
M650
HMI. Confirm change pass
[<> D12 K0]
[MOVP K1 D500]
Password changed
[MOV D12 D501]
(105)
M651
HMI. Reset Password
[MOVP K0 D500]
Password changed
(111)
[= D0 K40]
Change Base
[MOV K0 D10]
Password
M649
HMI. PB Confirm
[<> D10 K12345][<> D500 K1]
Password
Password changed
M649
HMI. PB Confirm
[<> D10 D501][= D500 K1]
Password
Password changed

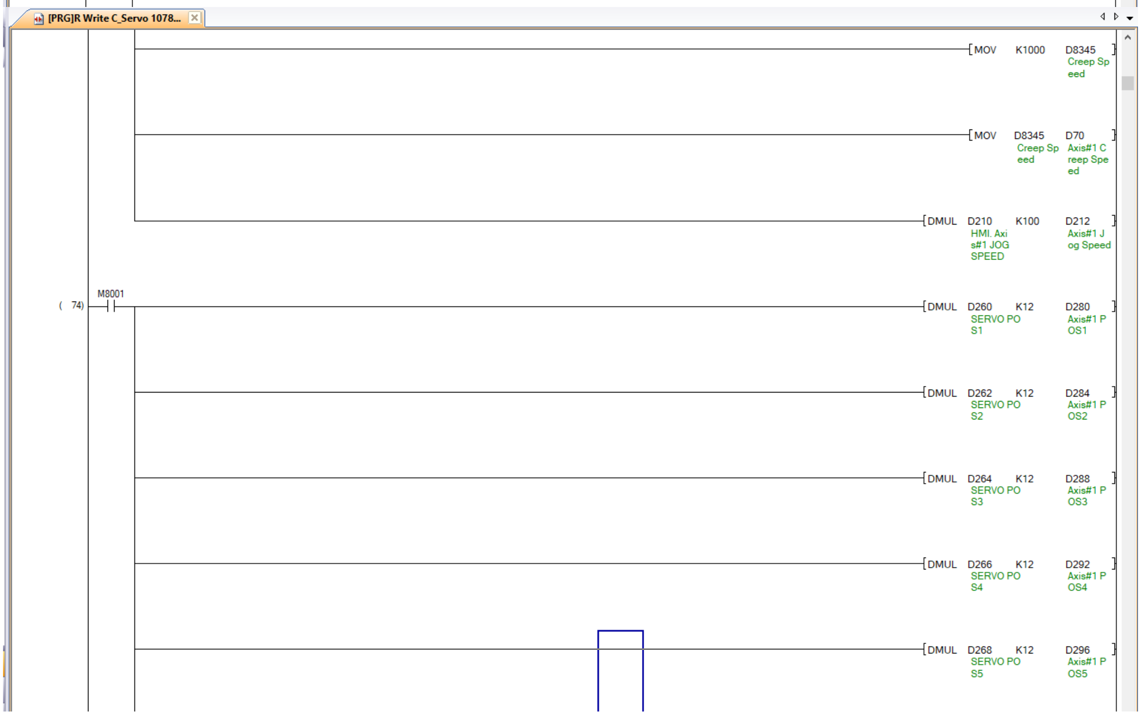

[PRG]Read 0_MAIN (Read O...
(146)
M649
HMI. PB Confirm
[= D10 K12345]
Password
[MOV K0 D1]
M649
HMI. PB Confirm
[= D10 D501][= D500 K1]
Password
Password changed
M606 M607 M608 M610 M611
HMI. Base Auto
HMI. Base Manual
HMI. Base Monitor
HMI. Base Data
HMI. Base Alarm
[= D0 K40]
[= D0 K41]
[= D0 K42]
[= D0 K43]
[= D0 K44]
[= D0 K45]
Change Base



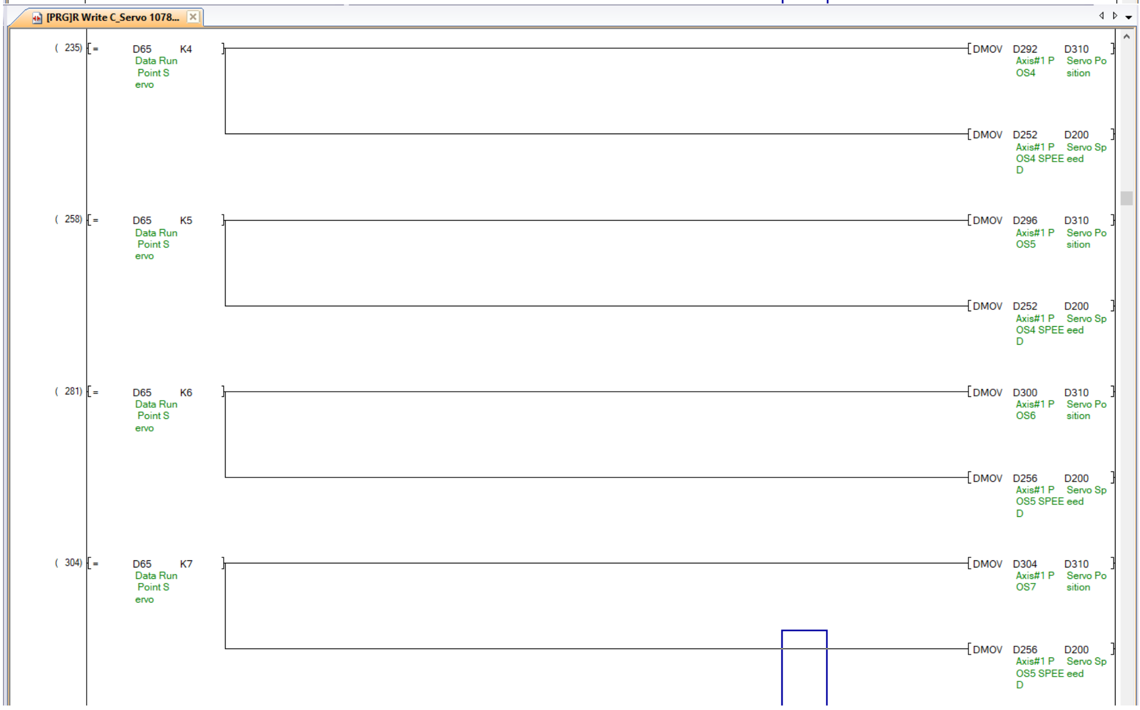
- Enable

- Servo

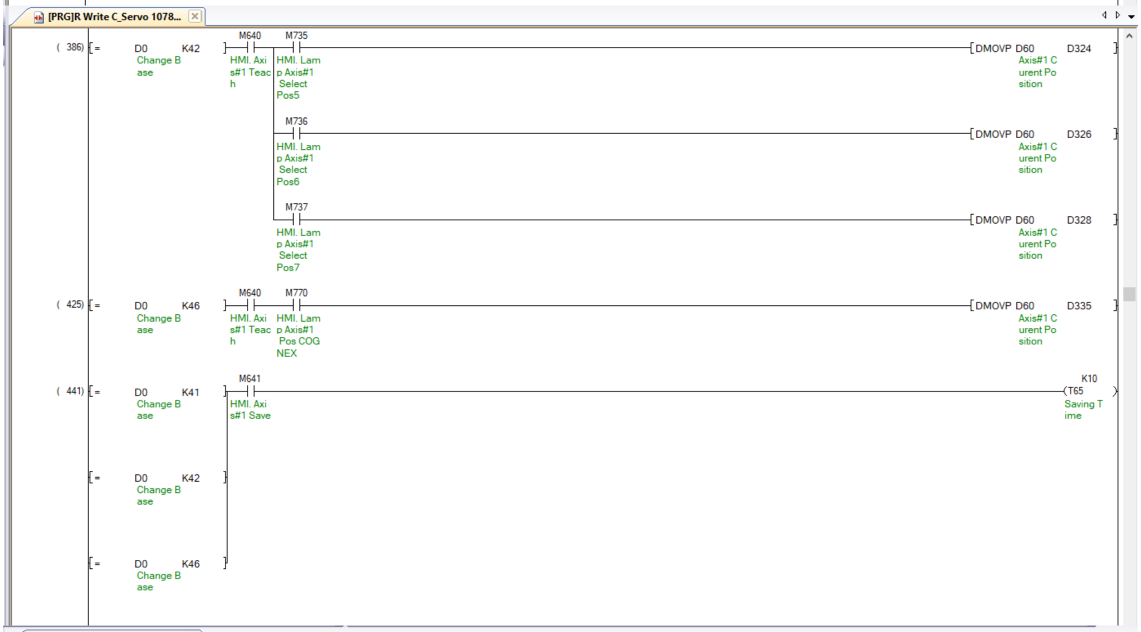

[PRG]R Write C_Servo 1078...
(0)
M0
PB EMG
M8349
M603
HMI. PB Stop
RST M900 Axis#1. Org Complete Flag
* [Title] Data
(6)
M8000
DMOV D8340 Current Axis#1 Position D60 Axis#1 Current Position
DMOV K100000 D8343 Max Speed
MOV K100 D8348 Acceleration time
MOV K100 D8349 Deceleration time
MOV K200 D8342 Bias Speed
MOV D205 SERVO ORG SPEED D8346 ZERO RETURN SPEED

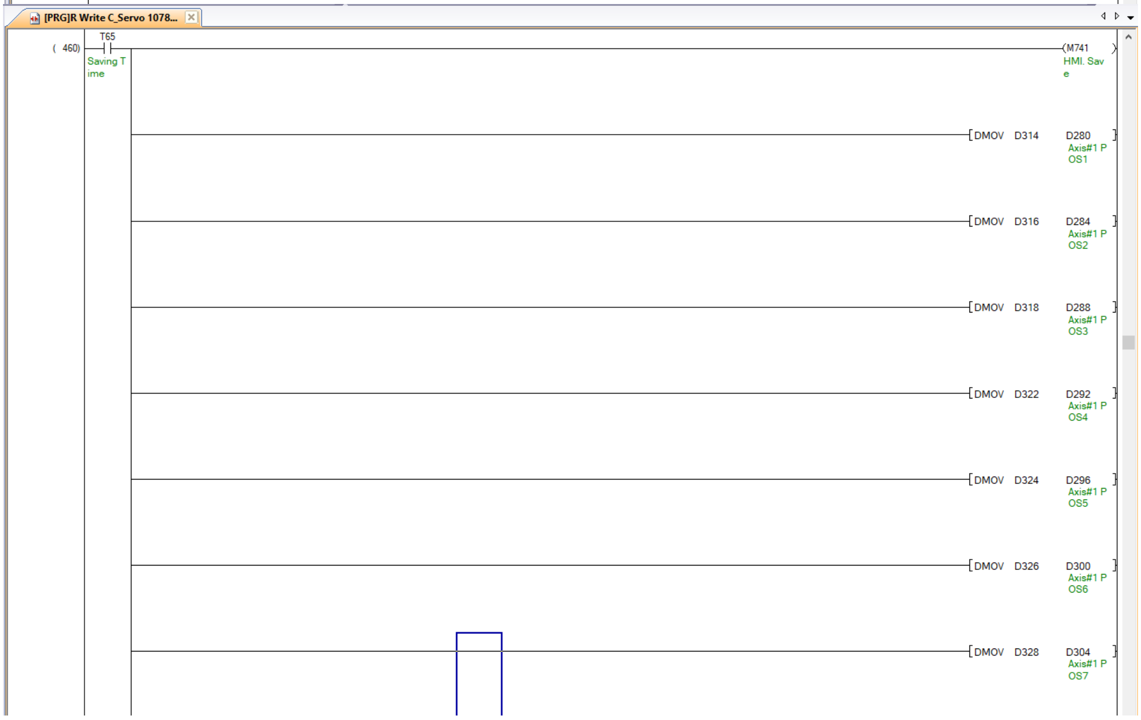

[PRG]R Write C_Servo 1078...
MOV K1000 D8345 Creep Speed
MOV D8345 Creep Speed D70 Axis#1 Creep Speed
DMUL D210 HMI. Axis#1 JOG SPEED K100 D212 Axis#1 Jog Speed
(74)
M8001
DMUL D260 SERVO POS1 K12 D280 Axis#1 POS1
DMUL D262 SERVO POS2 K12 D284 Axis#1 POS2
DMUL D264 SERVO POS3 K12 D288 Axis#1 POS3
DMUL D266 SERVO POS4 K12 D292 Axis#1 POS4
DMUL D268 SERVO POS5 K12 D296 Axis#1 POS5

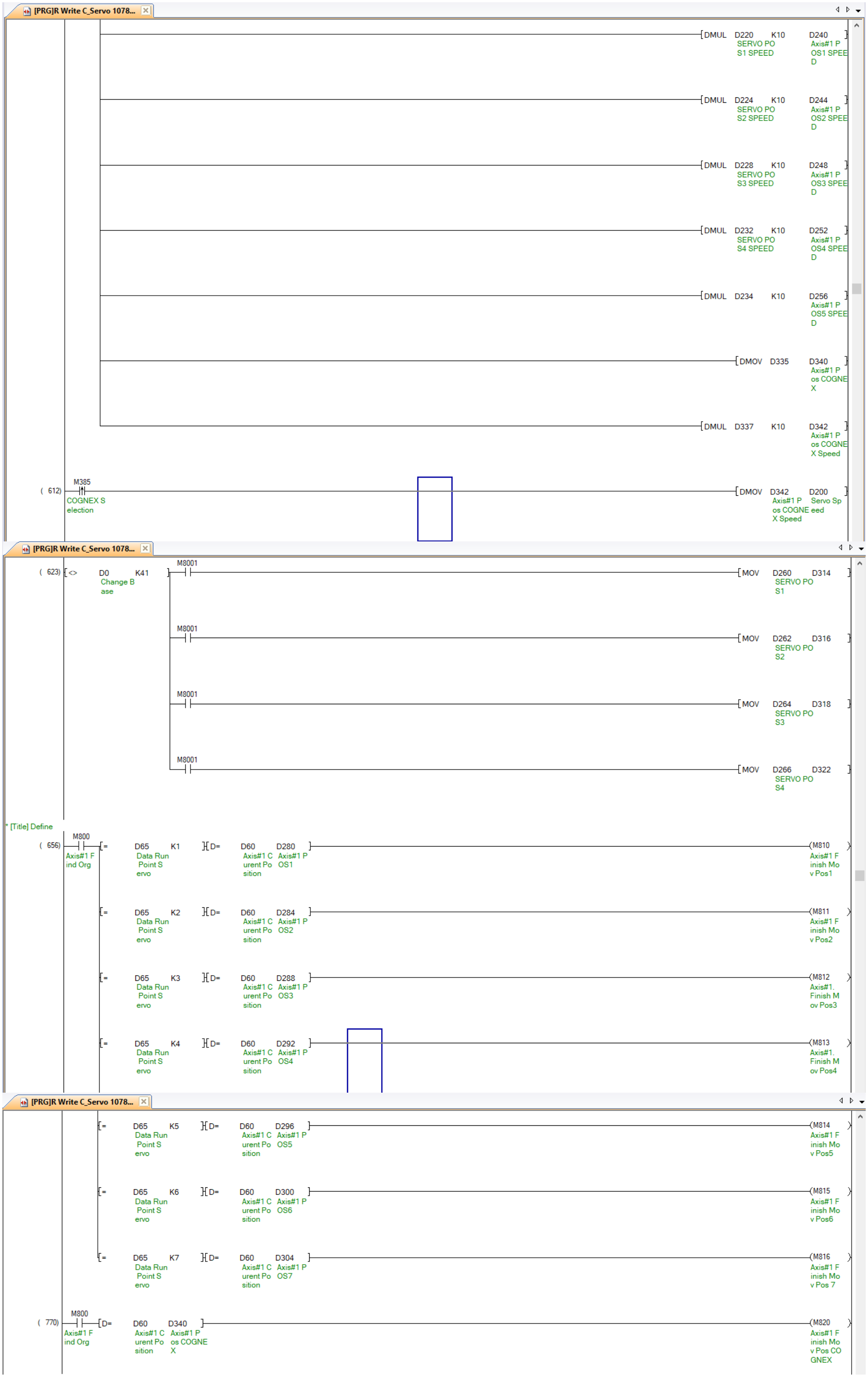

[PRG]R Write C_Servo 1078...
DMUL D270 K12 D300
SERVO PO S6
Axis#1 P OS6
DMUL D272 K12 D304
SERVO PO S7
Axis#1 P OS7
(166) = D65 K1
Data Run Point S ervo
DMOV D280 D310
Axis#1 P OS1
Servo Po sition
DMOV D240 D200
Axis#1 P OS1 SPEE D
Servo Sp eed
(189) = D65 K2
Data Run Point S ervo
DMOV D284 D310
Axis#1 P OS2
Servo Po sition
DMOV D244 D200
Axis#1 P OS2 SPEE D
Servo Sp eed
(212) = D65 K3
Data Run Point S ervo
DMOV D288 D310
Axis#1 P OS3
Servo Po sition
DMOV D248 D200
Axis#1 P OS3 SPEE D
Servo Sp eed

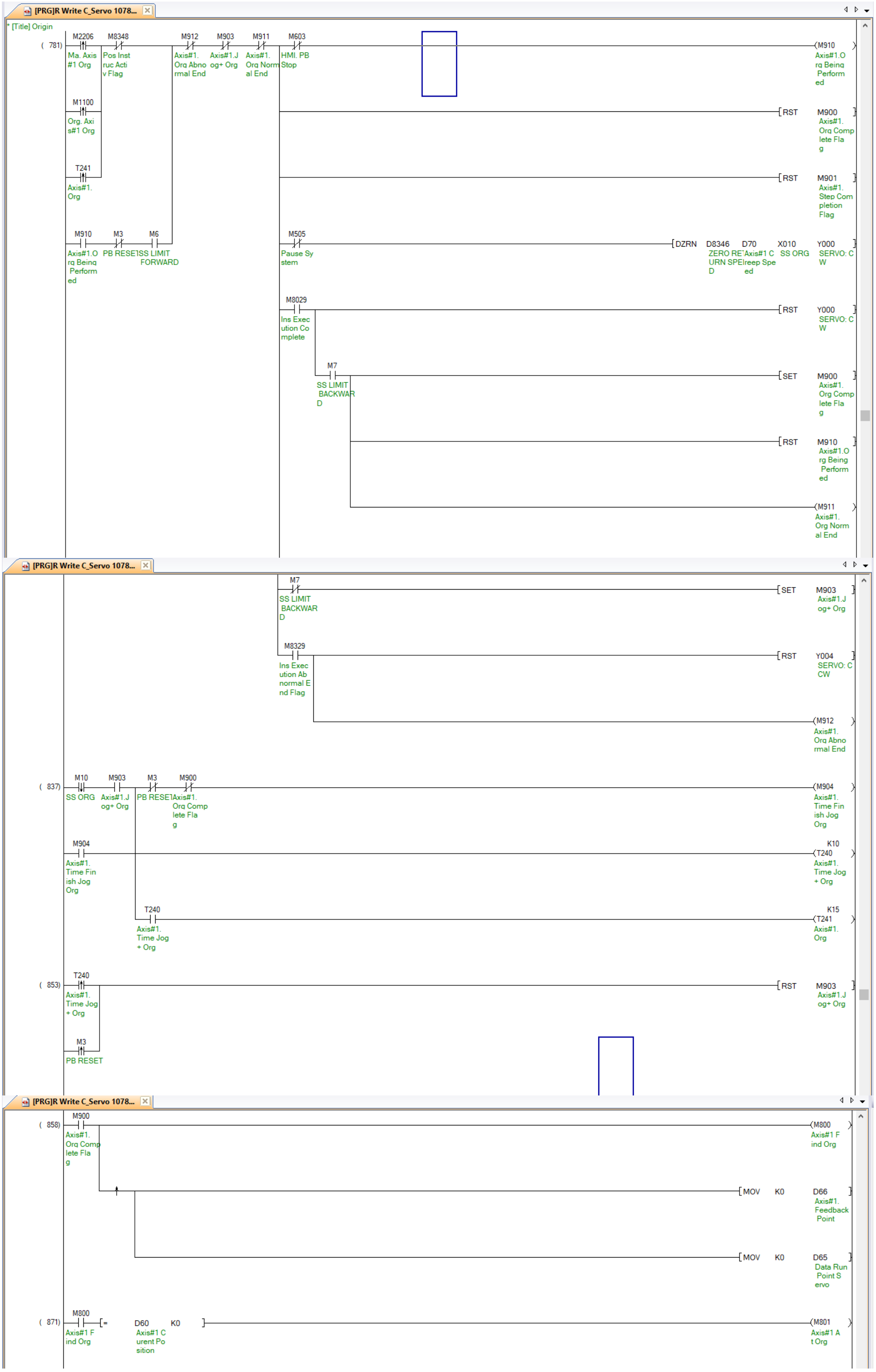

[PRG]R Write C_Servo 1078...
(235) = D65 K4
Data Run Point S ervo
DMOV D292 D310
Axis#1 P OS4
Servo Po sition
DMOV D252 D200
Axis#1 P OS4 SPEE D
Servo Sp eed
(258) = D65 K5
Data Run Point S ervo
DMOV D296 D310
Axis#1 P OS5
Servo Po sition
DMOV D252 D200
Axis#1 P OS4 SPEE D
Servo Sp eed
(281) = D65 K6
Data Run Point S ervo
DMOV D300 D310
Axis#1 P OS6
Servo Po sition
DMOV D256 D200
Axis#1 P OS5 SPEE D
Servo Sp eed
(304) = D65 K7
Data Run Point S ervo
DMOV D304 D310
Axis#1 P OS7
Servo Po sition
DMOV D256 D200
Axis#1 P OS5 SPEE D
Servo Sp eed


[PRG]R Write C_Servo 1078...
(327)
M6
M2202
SS LIMIT FORWAR
Ma. Axis #1 Jog-
M8343 Limit+
(332)
M7
M913
SS LIMIT BACKWAR D
Axis#1. Jog+ Being Performed
M8344 Limit-
(336)
= D0 K41
Change Base
M640
HMI. Axis#1 Teach
M731
HMI. Lamp Axis#1 Select Pos1
DMOV D60 D314
Axis#1 Curent Position
M732
HMI. Lamp Axis#1 Select Pos2
DMOV D60 D316
M733
HMI. Lamp Axis#1 Select Pos3
DMOV D60 D318
M734
HMI. Lamp Axis#1 Select Pos4
DMOVP D60 D322


[PRG]R Write C_Servo 1078...
(386)
= D0 K42
Change Base
M640
HMI. Axis#1 Teach
M735
HMI. Lamp Axis#1 Select Pos5
DMOVP D60 D324
Axis#1 Curent Position
M736
HMI. Lamp Axis#1 Select Pos6
DMOVP D60 D326
M737
HMI. Lamp Axis#1 Select Pos7
DMOVP D60 D328
(425)
= D0 K46
M640
M770
HMI. Lamp Axis#1 Pos COGNEX
DMOVP D60 D335
(441)
= D0 K41
M641
HMI. Axis#1 Save
K10
T65 Saving Time
= D0 K42
= D0 K46

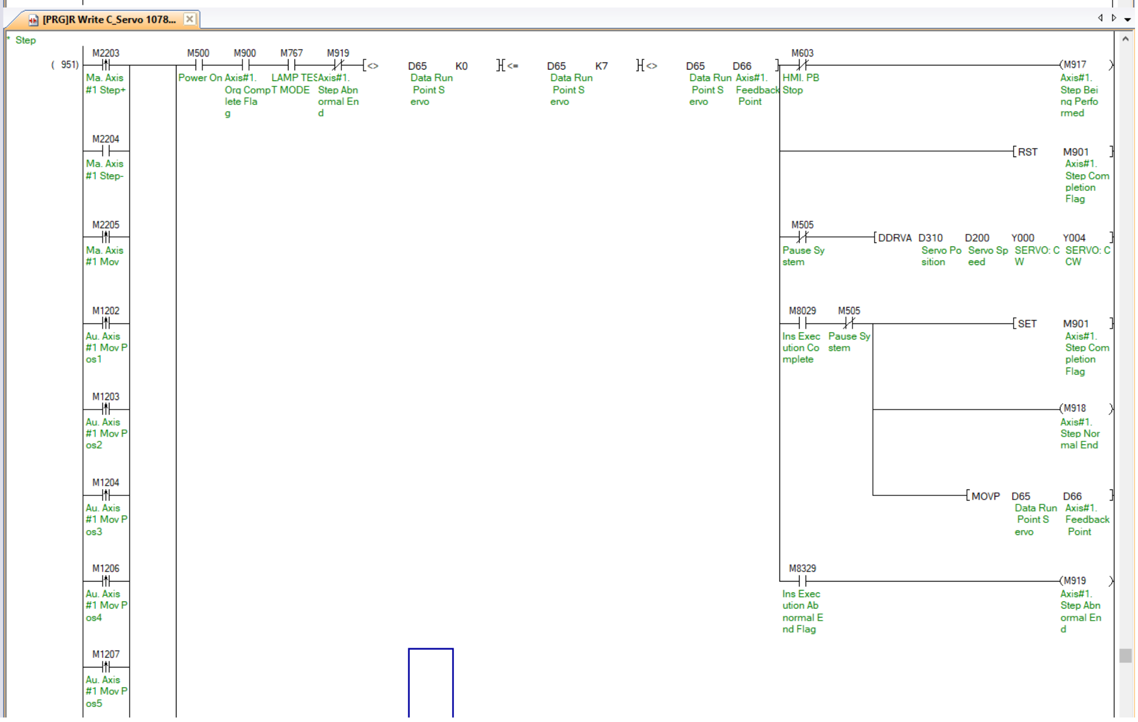

[PRG]R Write C_Servo 1078...
(460)
T65
Saving Time
M741 HMI. Save
DMOV D314 D280 Axis#1 POS1
DMOV D316 D284 Axis#1 POS2
DMOV D318 D288 Axis#1 POS3
DMOV D322 D292 Axis#1 POS4
DMOV D324 D296 Axis#1 POS5
DMOV D326 D300 Axis#1 POS6
DMOV D328 D304 Axis#1 POS7

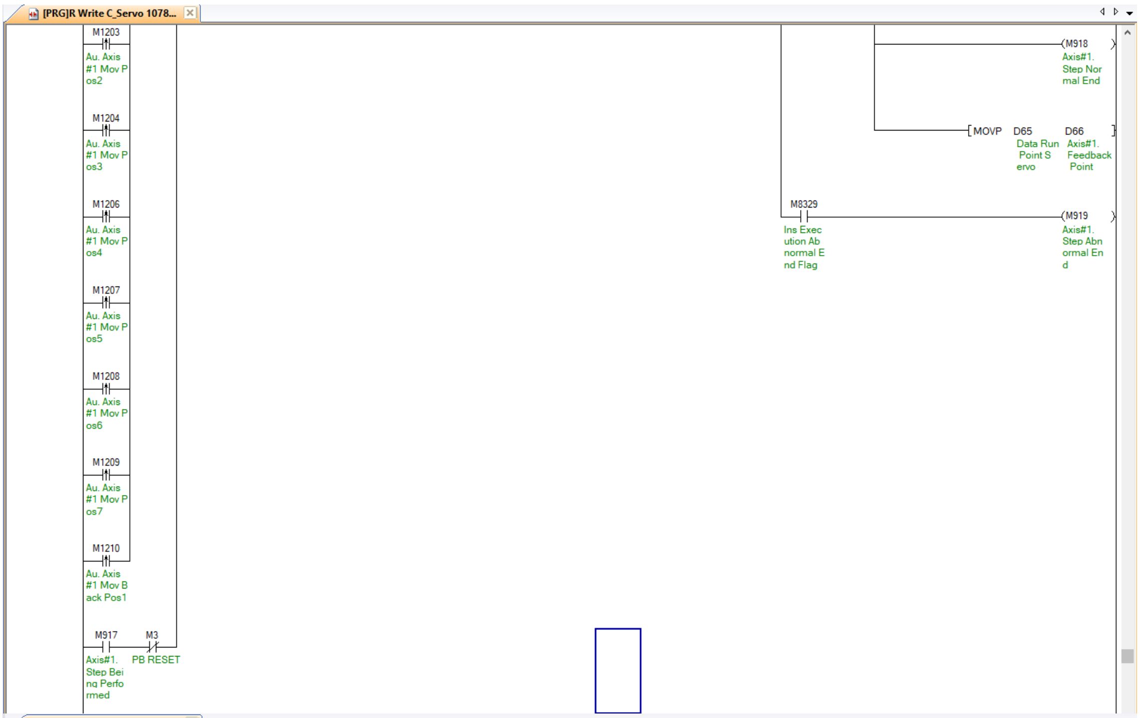

[PRG]R Write C_Servo 1078...
DMUL D220 K10 D240
SERVO POS1 SPEED
Axis#1 POS1 SPEED
DMUL D224 K10 D244
SERVO POS2 SPEED
Axis#1 POS2 SPEED
DMUL D228 K10 D248
SERVO POS3 SPEED
Axis#1 POS3 SPEED
DMUL D232 K10 D252
SERVO POS4 SPEED
Axis#1 POS4 SPEED
DMUL D234 K10 D256
Axis#1 POS5 SPEED
DMOV D335 D340
Axis#1 Pos COGNEX
DMUL D337 K10 D342
Axis#1 Pos COGNEX Speed
(612) M385
COGNEX Selection
DMOV D342 D200
Axis#1 Pos COGNEX Speed
Servo Speed
[PRG]R Write C_Servo 1078...
(623) <> D0 K41
Change Base
M8001
MOV D260 D314
SERVO POS1
M8001
MOV D262 D316
SERVO POS2
M8001
MOV D264 D318
SERVO POS3
M8001
MOV D266 D322
SERVO POS4
* [Title] Define
(656) M800
Axis#1 Find Org
= D65 K1 D= D60 D280
Data Run Point Servo
Axis#1 Curent Position
Axis#1 POS1
M810
Axis#1 Finish Mov Pos1
= D65 K2 D= D60 D284
Axis#1 POS2
M811
Axis#1 Finish Mov Pos2
= D65 K3 D= D60 D288
Axis#1 POS3
M812
Axis#1. Finish Mov Pos3
= D65 K4 D= D60 D292
Axis#1 POS4
M813
Axis#1. Finish Mov Pos4
[PRG]R Write C_Servo 1078...
= D65 K5 D= D60 D296
Axis#1 POS5
M814
Axis#1 Finish Mov Pos5
= D65 K6 D= D60 D300
Axis#1 POS6
M815
Axis#1 Finish Mov Pos6
= D65 K7 D= D60 D304
Axis#1 POS7
M816
Axis#1 Finish Mov Pos 7
(770) M800
Axis#1 Find Org
D= D60 D340
Axis#1 Curent Position
Axis#1 Pos COGNEX
M820
Axis#1 Finish Mov Pos COGNEX

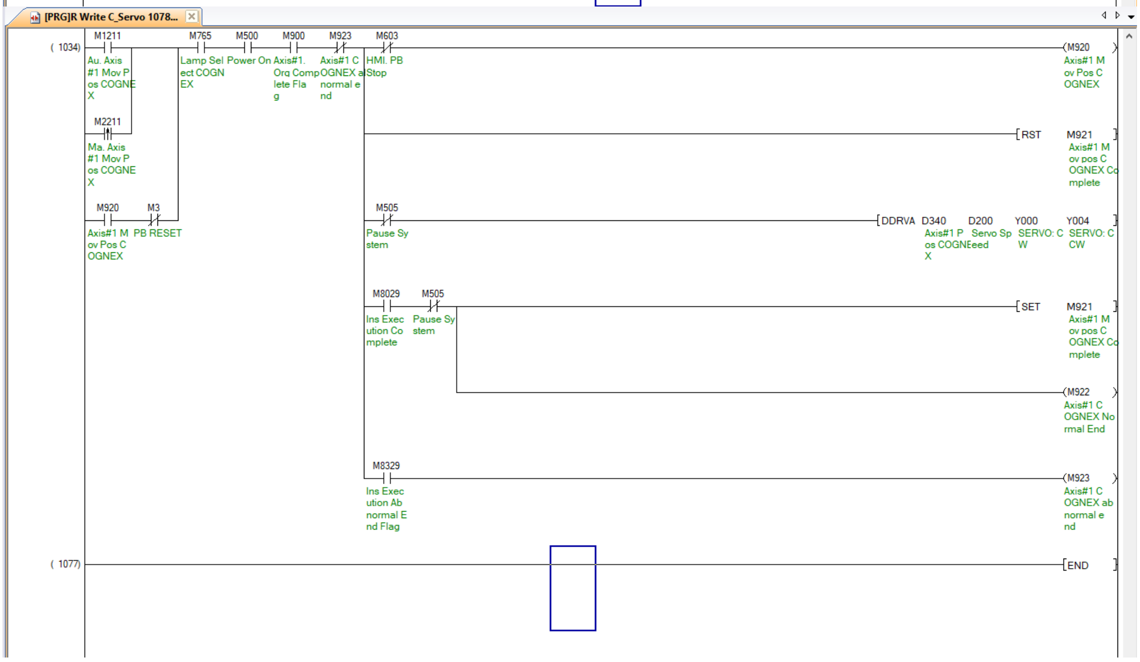

[PRG]R Write C_Servo 1078...
[Title] Origin
(781)
M2206 Ma. Axis #1 Org
M8348 Pos Instruc Activ Flag
M912 Axis#1. Org Abnormal End
M903 Axis#1.Jog+ Org
M911 Axis#1. Org Normal End
M603 HMI. PB Stop
M910 Axis#1.Org Being Performed
M1100 Org. Axis#1 Org
T241 Axis#1. Org
RST M900 Axis#1. Org Complete Flag
RST M901 Axis#1. Step Completion Flag
M910 Axis#1.Org Being Performed
M3 PB RESET
M6 SS LIMIT FORWARD
M505 Pause System
DZRN D8346 ZERO RETURN SPEED D70 Axis#1 Creep Speed X010 SS ORG Y000 SERVO: CW
M8029 Ins Execution Complete
RST Y000 SERVO: CW
M7 SS LIMIT BACKWARD
SET M900 Axis#1. Org Complete Flag
RST M910 Axis#1.Org Being Performed
M911 Axis#1. Org Normal End
[PRG]R Write C_Servo 1078...
M7 SS LIMIT BACKWARD
SET M903 Axis#1.Jog+ Org
M8329 Ins Execution Abnormal End Flag
RST Y004 SERVO: CCW
M912 Axis#1. Org Abnormal End
(837)
M10 SS ORG
M903 Axis#1.Jog+ Org
M3 PB RESET
M900 Axis#1. Org Complete Flag
M904 Axis#1. Time Finish Jog Org
M904 Axis#1. Time Finish Jog Org
K10 T240 Axis#1. Time Jog + Org
T240 Axis#1. Time Jog + Org
K15 T241 Axis#1. Org
(853)
T240 Axis#1. Time Jog + Org
RST M903 Axis#1.Jog+ Org
M3 PB RESET
[PRG]R Write C_Servo 1078...
(858)
M900 Axis#1. Org Complete Flag
M800 Axis#1 Find Org
MOV K0 D66 Axis#1. Feedback Point
MOV K0 D65 Data Run Point Servo
(871)
M800 Axis#1 Find Org
= D60 Axis#1 Curent Position K0
M801 Axis#1 At Org


[PRG]R Write C_Servo 1078...
Jog
(878)
M2201
Ma. Axis #1 Jog+
M8348
Pos Inst ruc Acti v Flag
M914
Axis#1. Jog+ End
M913
Axis#1. Jog+ Bei ng Perfo rmed
M903
Axis#1.J og+ Org
RST M900
Axis#1. Org Comp lete Fla g
RST M901
Axis#1. Step Com pletion Flag
M505
Pause Sy stem
DDRVI K999999 D212 Y000 Y004
Axis#1 J og Speed
SERVO: C W
SERVO: C CW
M2201 M903
M914
M8329
Ins Exec ution Ab normal E nd Flag

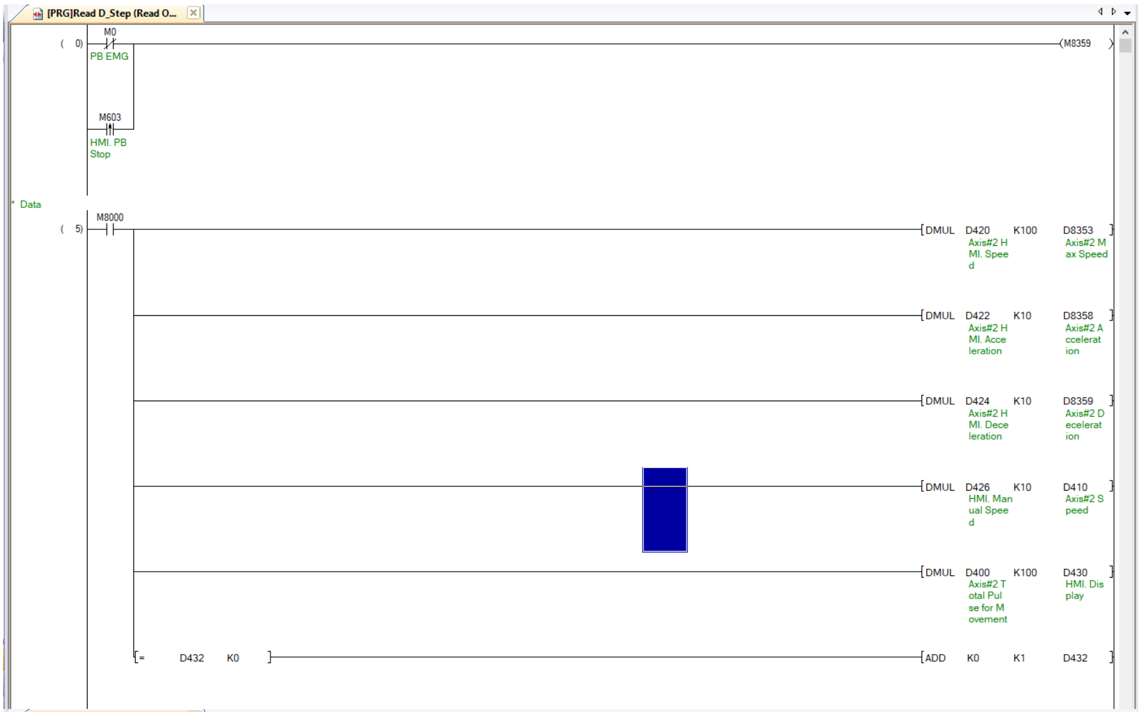

[PRG]R Write C_Servo 1078...
(916)
M2202
Ma. Axis #1 Jog-
M8348
Pos Inst ruc Acti v Flag
M916
Axis#1. Jog- End
M915
Axis#1. Jog- Bei ng Perfo rmed
RST M900
Axis#1. Org Comp lete Fla g
RST M901
Axis#1. Step Com pletion Flag
M505
Pause Sy stem
DDRVI K-999999 D212 Y000 Y004
Axis#1 J og Speed
SERVO: C W
SERVO: C CW
M2202
M916
Axis#1. Jog- End
M8329
Ins Exec ution Ab normal E nd Flag

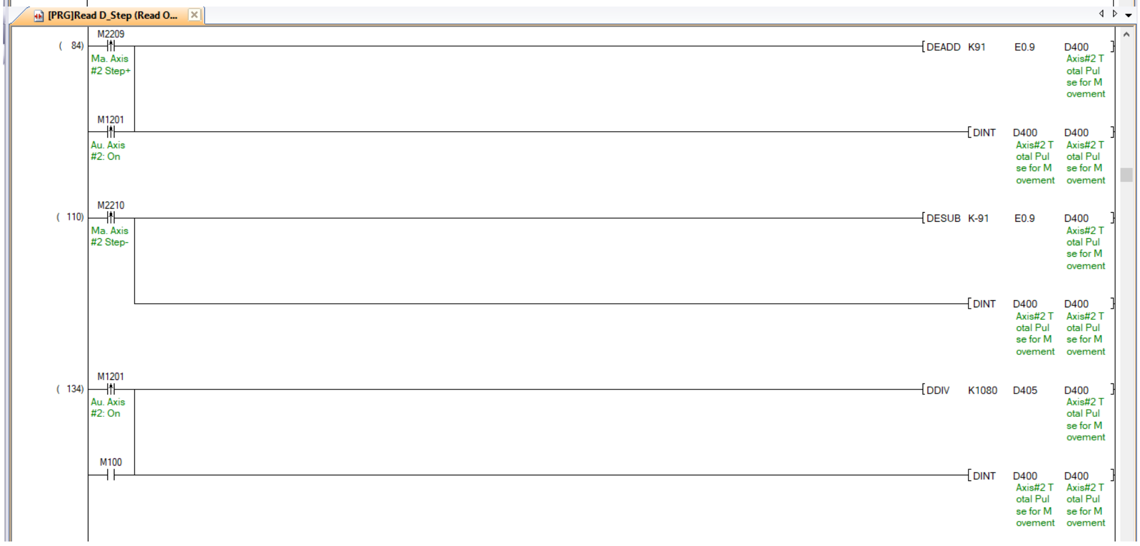

[PRG]R Write C_Servo 1078...
Step
(951)
M2203
Ma. Axis #1 Step+
M500
Power On
M900
Axis#1. Org Comp lete Fla g
M767
LAMP TEST MODE
M919
Axis#1. Step Abn ormal En d
<> D65 K0
Data Run Point S ervo
<= D65 K7
<> D65 D66
Axis#1. Feedback Point
M603
HMI. PB Stop
M917
Axis#1. Step Bei ng Perfo rmed
M2204
Ma. Axis #1 Step-
RST M901
Axis#1. Step Com pletion Flag
M2205
Ma. Axis #1 Mov
M505
Pause Sy stem
DDRVA D310 D200 Y000 Y004
Servo Po sition
Servo Sp eed
SERVO: C W
SERVO: C CW
M1202
Au. Axis #1 Mov P os1
M8029
Ins Exec ution Co mplete
M505
Pause Sy stem
SET M901
M1203
Au. Axis #1 Mov P os2
M918
Axis#1. Step Nor mal End
M1204
Au. Axis #1 Mov P os3
MOVP D65 D66
Data Run Point S ervo
Axis#1. Feedback Point
M1206
Au. Axis #1 Mov P os4
M8329
Ins Exec ution Ab normal E nd Flag
M919
Axis#1. Step Abn ormal En d
M1207
Au. Axis #1 Mov P os5

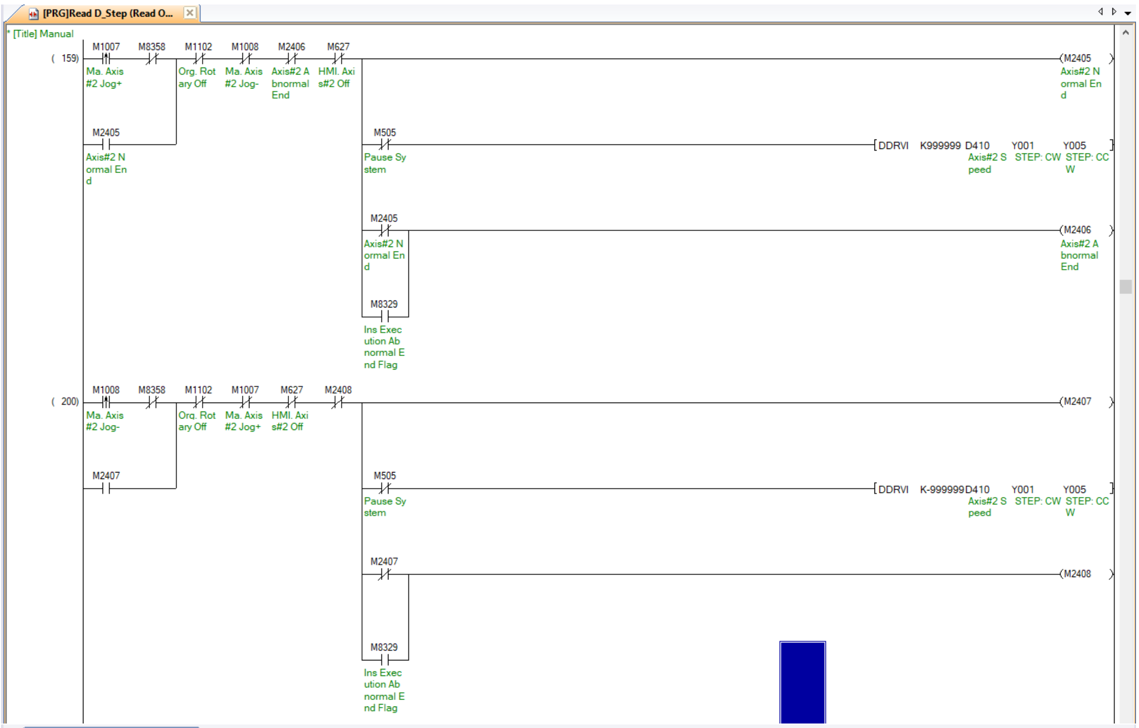

[PRG]R Write C_Servo 1078...
M1203
Au. Axis #1 Mov Pos2
M1204
Au. Axis #1 Mov Pos3
M1206
Au. Axis #1 Mov Pos4
M1207
Au. Axis #1 Mov Pos5
M1208
Au. Axis #1 Mov Pos6
M1209
Au. Axis #1 Mov Pos7
M1210
Au. Axis #1 Mov Back Pos1
M917
Axis#1. Step Being Performed
M3
PB RESET
M918
Axis#1. Step Normal End
MOVP D65 D66
Data Run Point Servo
Axis#1. Feedback Point
M8329
Ins Execution Abnormal End Flag
M919
Axis#1. Step Abnormal End

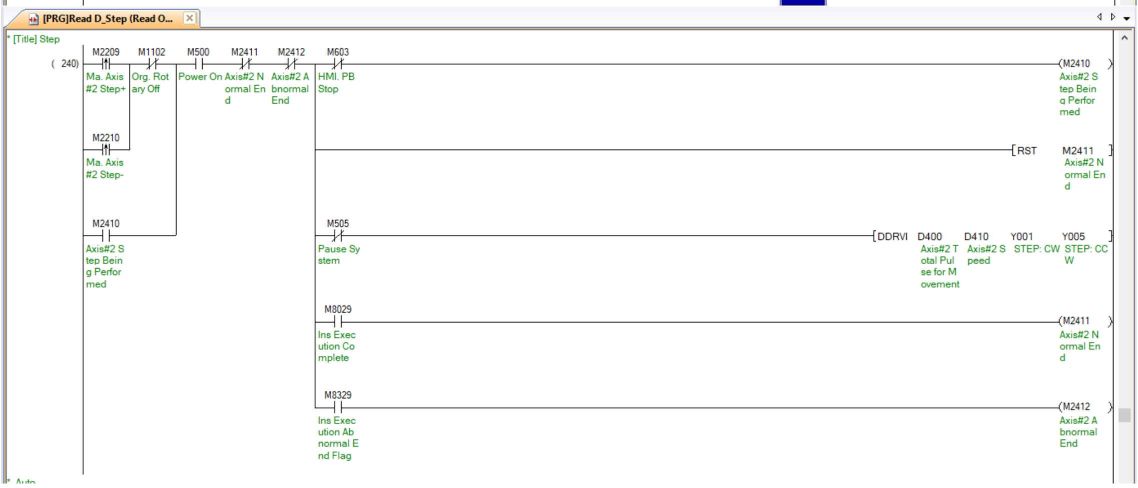

[PRG]R Write C_Servo 1078...
(1034)
M1211
Au. Axis #1 Mov Pos COGNEX
M2211
Ma. Axis #1 Mov Pos COGNEX
M920
Axis#1 Mov Pos COGNEX
M3
PB RESET
M765
Lamp Select COGNEX
M500
Power On
M900
Axis#1. Org Complete Flag
M923
Axis#1 COGNEX abnormal end
M603
HMI. PB Stop
M920
Axis#1 Mov Pos COGNEX
RST M921
Axis#1 Mov pos COGNEX Complete
M505
Pause System
DDRVA D340 D200 Y000 Y004
Axis#1 Pos COGNEX
Servo Speed
SERVO: CW
SERVO: CCW
M8029
Ins Execution Complete
M505
Pause System
SET M921
Axis#1 Mov pos COGNEX Complete
M922
Axis#1 COGNEX Normal End
M8329
Ins Execution Abnormal End Flag
M923
Axis#1 COGNEX abnormal end
(1077)
END

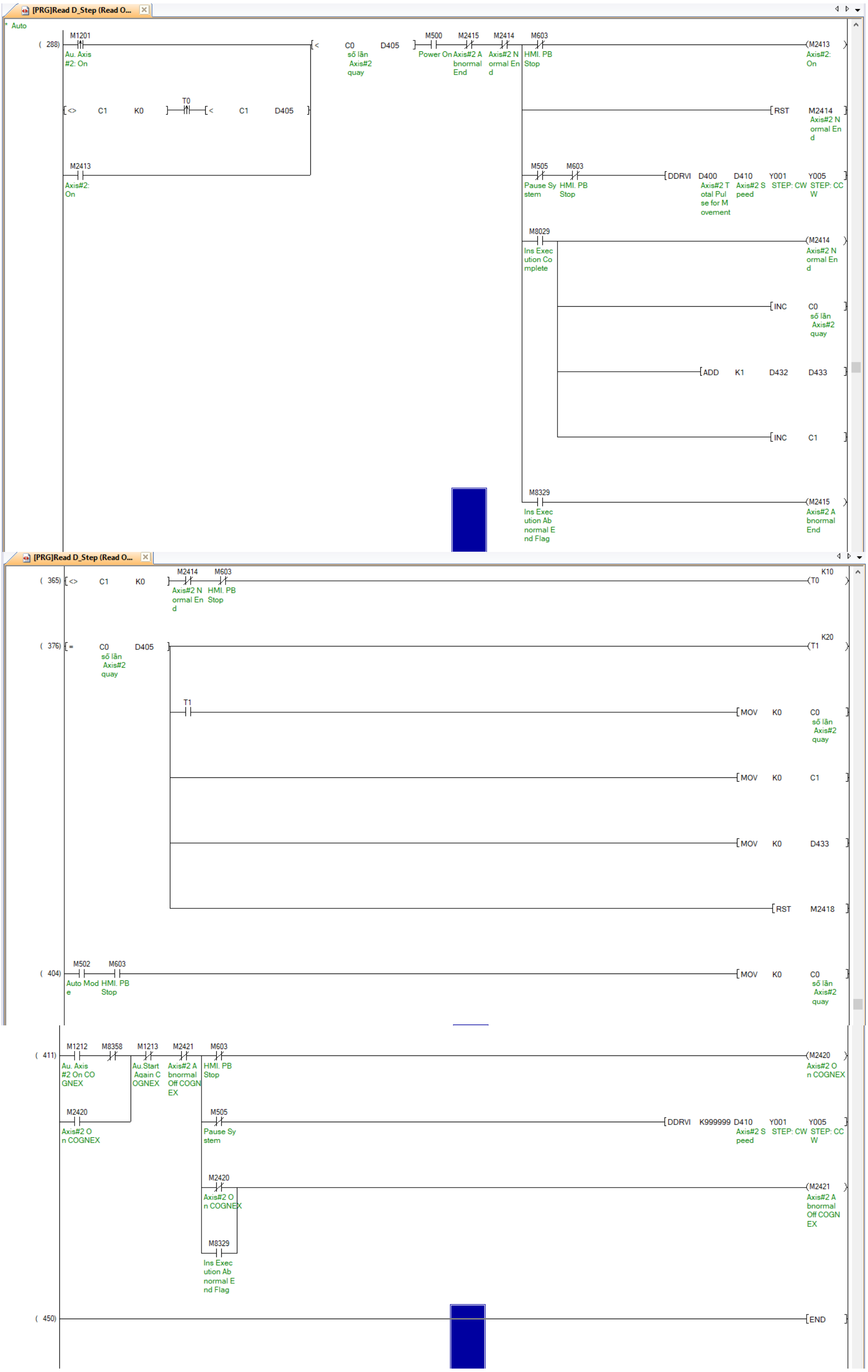
- Step


[PRG]Read D_Step (Read O...
(0)
M0
PB EMG
M8359
M603
HMI. PB Stop
* Data
(5)
M8000
DMUL D420 K100 D8353
Axis#2 HMI. Speed
Axis#2 Max Speed
DMUL D422 K10 D8358
Axis#2 HMI. Acceleration
Axis#2 Acceleration
DMUL D424 K10 D8359
Axis#2 HMI. Deceleration
Axis#2 Deceleration
DMUL D426 K10 D410
HMI. Manual Speed
Axis#2 Speed
DMUL D400 K100 D430
Axis#2 Total Pulse for Movement
HMI. Display
= D432 K0
ADD K0 K1 D432

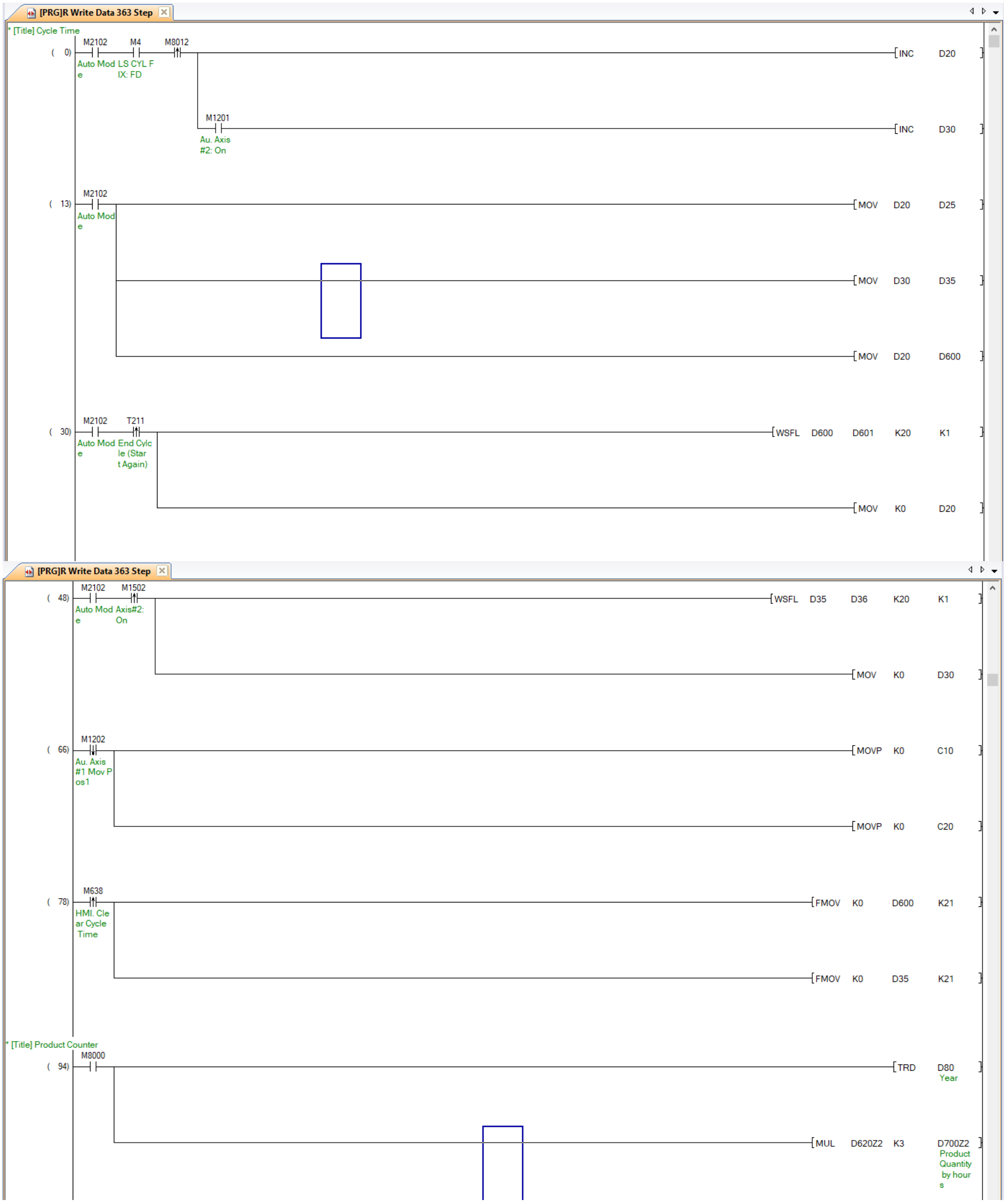

[PRG]Read D_Step (Read O...
(84)
M2209
Ma. Axis #2 Step+
DEADD K91 E0.9 D400
Axis#2 Total Pulse for Movement
M1201
Au. Axis #2: On
DINT D400 D400
Axis#2 Total Pulse for Movement
(110)
M2210
Ma. Axis #2 Step-
DESUB K-91 E0.9 D400
Axis#2 Total Pulse for Movement
DINT D400 D400
Axis#2 Total Pulse for Movement
(134)
M1201
Au. Axis #2: On
DDIV K1080 D405 D400
Axis#2 Total Pulse for Movement
M100
DINT D400 D400
Axis#2 Total Pulse for Movement

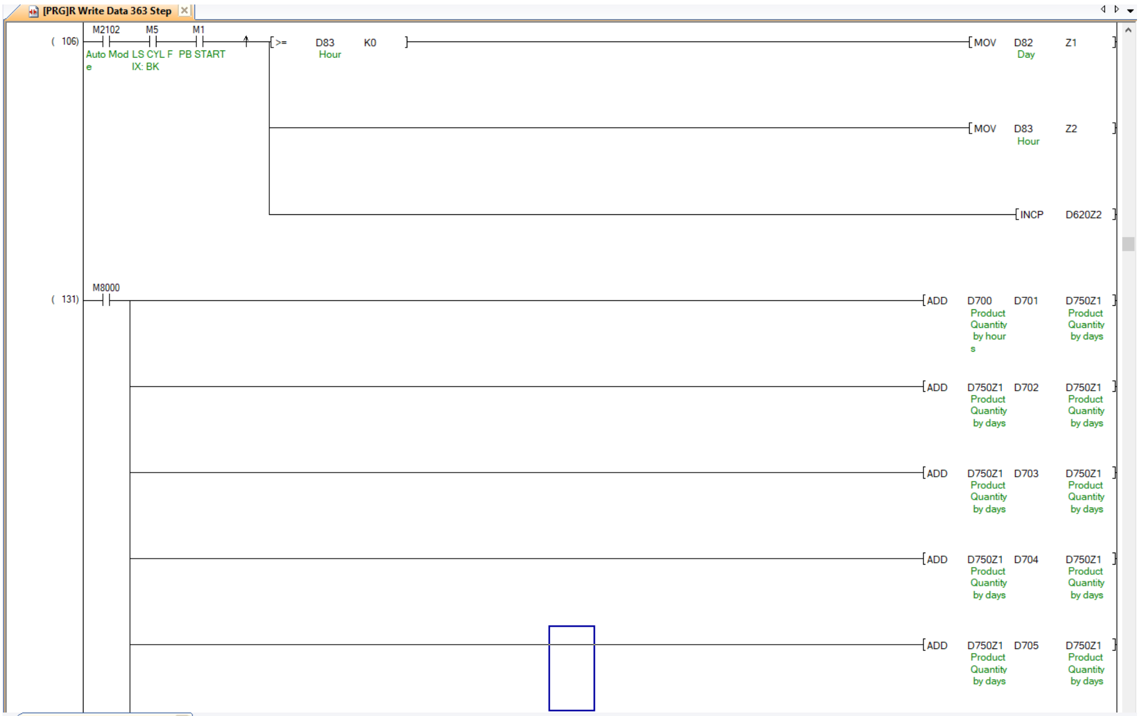

[PRG]Read D_Step (Read O...
[Title] Manual
(159)
M1007 Ma. Axis #2 Jog+
M8358
M1102 Org. Rotary Off
M1008 Ma. Axis #2 Jog-
M2406 Axis#2 Abnormal End
M627 HMI. Axis#2 Off
M2405 Axis#2 Normal End
M2405 Axis#2 Normal End
M505 Pause System
DDRVI K999999 D410 Y001 Y005
Axis#2 Speed STEP: CW STEP: CCW
M2405 Axis#2 Normal End
M2406 Axis#2 Abnormal End
M8329 Ins Execution Abnormal End Flag
(200)
M1008 Ma. Axis #2 Jog-
M8358
M1102 Org. Rotary Off
M1007 Ma. Axis #2 Jog+
M627 HMI. Axis#2 Off
M2408
M2407
M2407
M505 Pause System
DDRVI K-999999 D410 Y001 Y005
Axis#2 Speed STEP: CW STEP: CCW
M2407
M2408
M8329 Ins Execution Abnormal End Flag

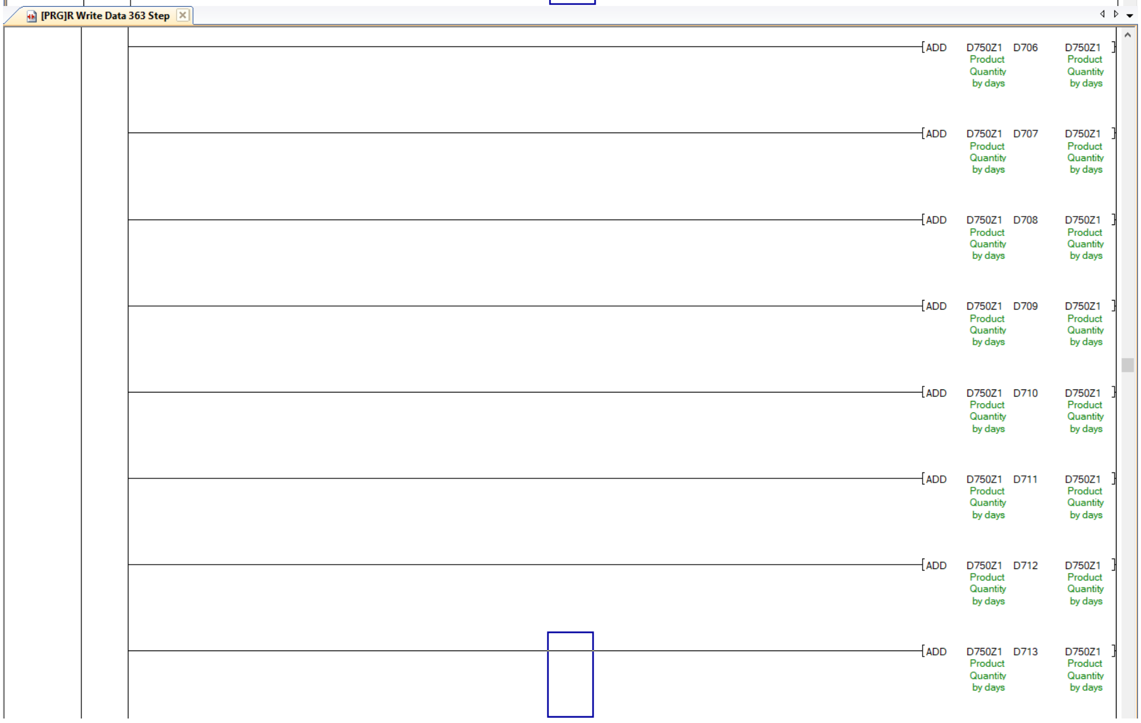

[PRG]Read D_Step (Read O...
[Title] Step
(240)
M2209 Ma. Axis #2 Step+
M1102 Org. Rotary Off
M500 Power On
M2411 Axis#2 Normal End
M2412 Axis#2 Abnormal End
M603 HMI. PB Stop
M2410 Axis#2 Step Being Performed
M2210 Ma. Axis #2 Step-
RST M2411 Axis#2 Normal End
M2410 Axis#2 Step Being Performed
M505 Pause System
DDRVI D400 D410 Y001 Y005
Axis#2 Total Pulse for Movement Axis#2 Speed STEP: CW STEP: CCW
M8029 Ins Execution Complete
M2411 Axis#2 Normal End
M8329 Ins Execution Abnormal End Flag
M2412 Axis#2 Abnormal End

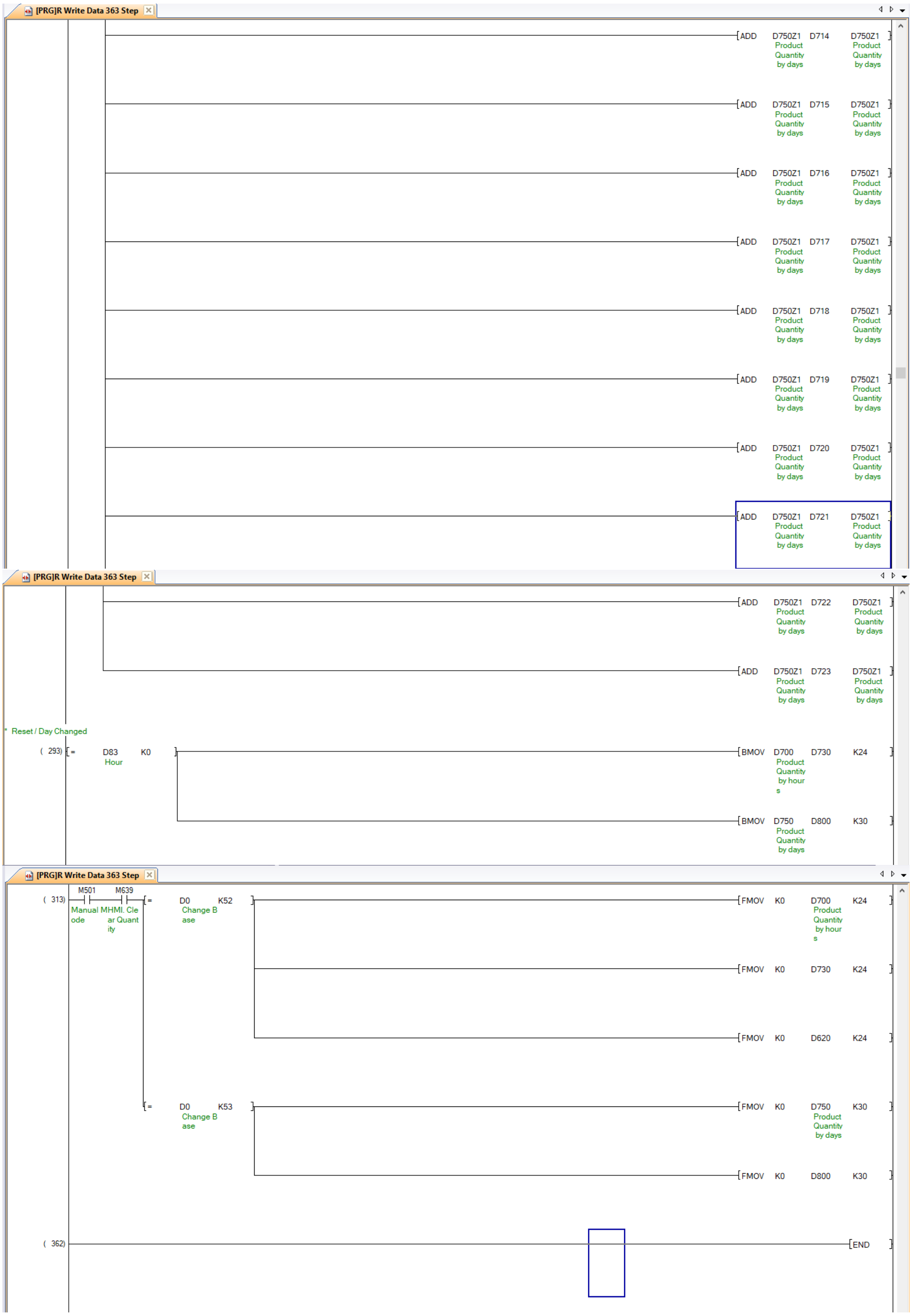

[PRG]Read D_Step (Read O...
Auto
(288) M1201 Au. Axis #2: On
C0 số lần Axis#2 quay D405
M500 Power On
M2415 Axis#2 Abnormal End
M2414 Axis#2 Normal End
M603 HMI. PB Stop
M2413 Axis#2: On
C1 K0 T0 C1 D405
RST M2414 Axis#2 Normal End
M2413 Axis#2: On
M505 Pause System
M603 HMI. PB Stop
DDRVI D400 Axis#2 Total Pulse for Movement D410 Axis#2 Speed Y001 STEP: CW Y005 STEP: CCW
M8029 Ins Execution Complete
M2414 Axis#2 Normal End
INC C0 số lần Axis#2 quay
ADD K1 D432 D433
INC C1
M8329 Ins Execution Abnormal End Flag
M2415 Axis#2 Abnormal End
[PRG]Read D_Step (Read O...
(365) C1 K0 M2414 Axis#2 Normal End M603 HMI. PB Stop K10 T0
(376) C0 số lần Axis#2 quay D405 K20 T1
T1 MOV K0 C0 số lần Axis#2 quay
MOV K0 C1
MOV K0 D433
RST M2418
(404) M502 Auto Mode M603 HMI. PB Stop MOV K0 C0 số lần Axis#2 quay
(411) M1212 Au. Axis #2 On COGNEX M8358 M1213 Au.Start Again COGNEX M2421 Axis#2 Abnormal Off COGNEX M603 HMI. PB Stop M2420 Axis#2 On COGNEX
M2420 Axis#2 On COGNEX
M505 Pause System DDRVI K999999 D410 Axis#2 Speed Y001 STEP: CW Y005 STEP: CCW
M2420 Axis#2 On COGNEX M2421 Axis#2 Abnormal Off COGNEX
M8329 Ins Execution Abnormal End Flag
(450) END

- Data

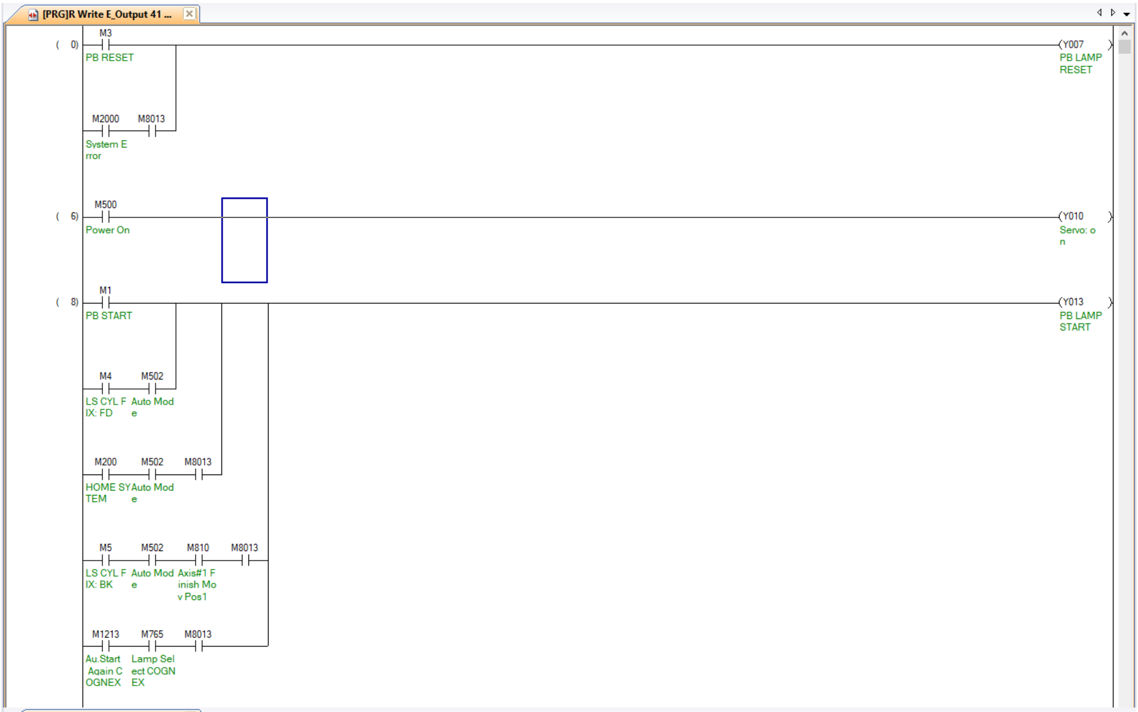

[PRG]R Write Data 363 Step
* [Title] Cycle Time
M2102 M4 M8012
Auto Mode LS CYL FIX: FD
INC D20
M1201
Au. Axis #2: On
INC D30
M2102
Auto Mode
MOV D20 D25
MOV D30 D35
MOV D20 D600
M2102 T211
Auto Mode End Cycle (Start Again)
WSFL D600 D601 K20 K1
MOV K0 D20
[PRG]R Write Data 363 Step
M2102 M1502
Auto Mode Axis#2: On
WSFL D35 D36 K20 K1
MOV K0 D30
M1202
Au. Axis #1 Mov Pos1
MOVP K0 C10
MOVP K0 C20
M638
HMI. Clear Cycle Time
FMOV K0 D600 K21
FMOV K0 D35 K21
* [Title] Product Counter
M8000
TRD D80 Year
MUL D620Z2 K3 D700Z2 Product Quantity by hours


[PRG]R Write Data 363 Step
(106)
M2102
M5
M1
Auto Mode
LS CYL FIX: BK
PB START
>= D83 K0
Hour
MOV D82 Z1
Day
MOV D83 Z2
Hour
INCP D620Z2
(131)
M8000
ADD D700 D701 D750Z1
Product Quantity by hours
Product Quantity by days
ADD D750Z1 D702 D750Z1
ADD D750Z1 D703 D750Z1
ADD D750Z1 D704 D750Z1
ADD D750Z1 D705 D750Z1


[PRG]R Write Data 363 Step
ADD D750Z1 D706 D750Z1
Product Quantity by days
ADD D750Z1 D707 D750Z1
ADD D750Z1 D708 D750Z1
ADD D750Z1 D709 D750Z1
ADD D750Z1 D710 D750Z1
ADD D750Z1 D711 D750Z1
ADD D750Z1 D712 D750Z1
ADD D750Z1 D713 D750Z1


[PRG]R Write Data 363 Step
ADD D750Z1 Product Quantity by days D714 D750Z1 Product Quantity by days
ADD D750Z1 D715 D750Z1
ADD D750Z1 D716 D750Z1
ADD D750Z1 D717 D750Z1
ADD D750Z1 D718 D750Z1
ADD D750Z1 D719 D750Z1
ADD D750Z1 D720 D750Z1
ADD D750Z1 D721 D750Z1
ADD D750Z1 D722 D750Z1
ADD D750Z1 D723 D750Z1
Reset / Day Changed
(293) = D83 Hour K0
BMOV D700 Product Quantity by hours D730 K24
BMOV D750 Product Quantity by days D800 K30
(313) M501 Manual ode M639 MHMI. Clear Quantity
= D0 Change Base K52
FMOV K0 D700 Product Quantity by hours K24
FMOV K0 D730 K24
FMOV K0 D620 K24
= D0 Change Base K53
FMOV K0 D750 Product Quantity by days K30
FMOV K0 D800 K30
(362) END

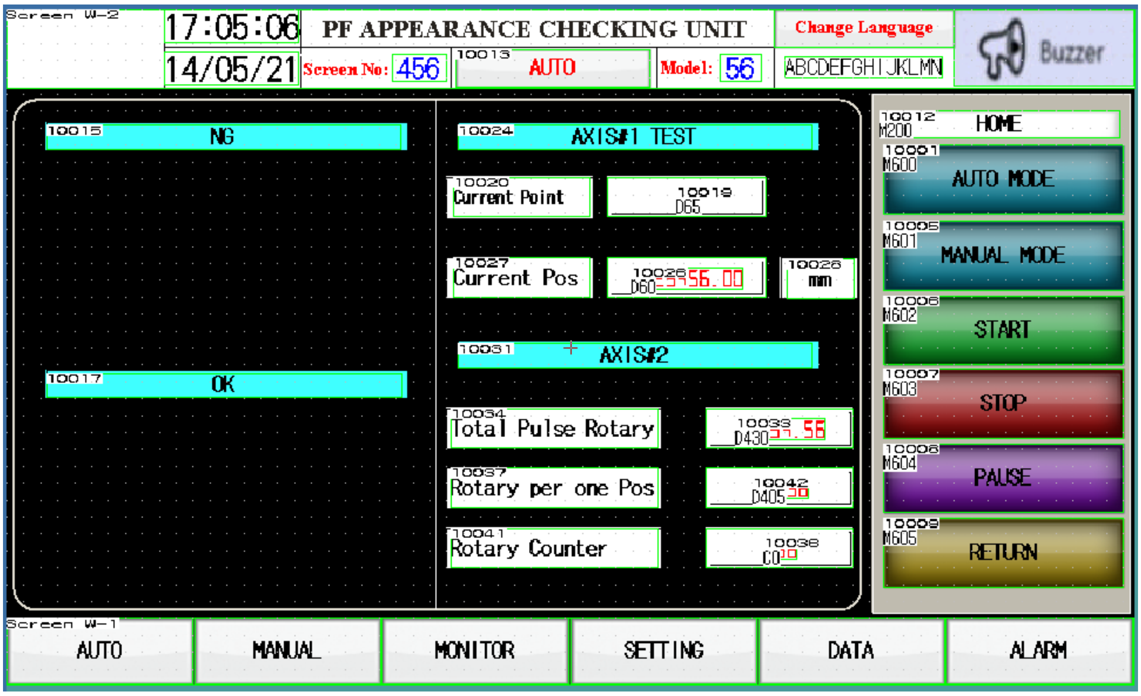
- Output


[PRG]R Write E_Output 41 ...
(0)
M3
PB RESET
M2000
M8013
System E
rror
Y007
PB LAMP
RESET
(6)
M500
Power On
Y010
Servo: o
n
(8)
M1
PB START
Y013
PB LAMP
START
M4
M502
LS CYL F
IX: FD
Auto Mod
e
M200
M502
M8013
HOME SY
TEM
Auto Mod
e
M5
M502
M810
M8013
LS CYL F
IX: BK
Auto Mod
e
Axis#1 F
inish Mo
v Pos1
M1213
M765
M8013
Au.Start
Again C
OGNEX
Lamp Sel
ect COGN
EX

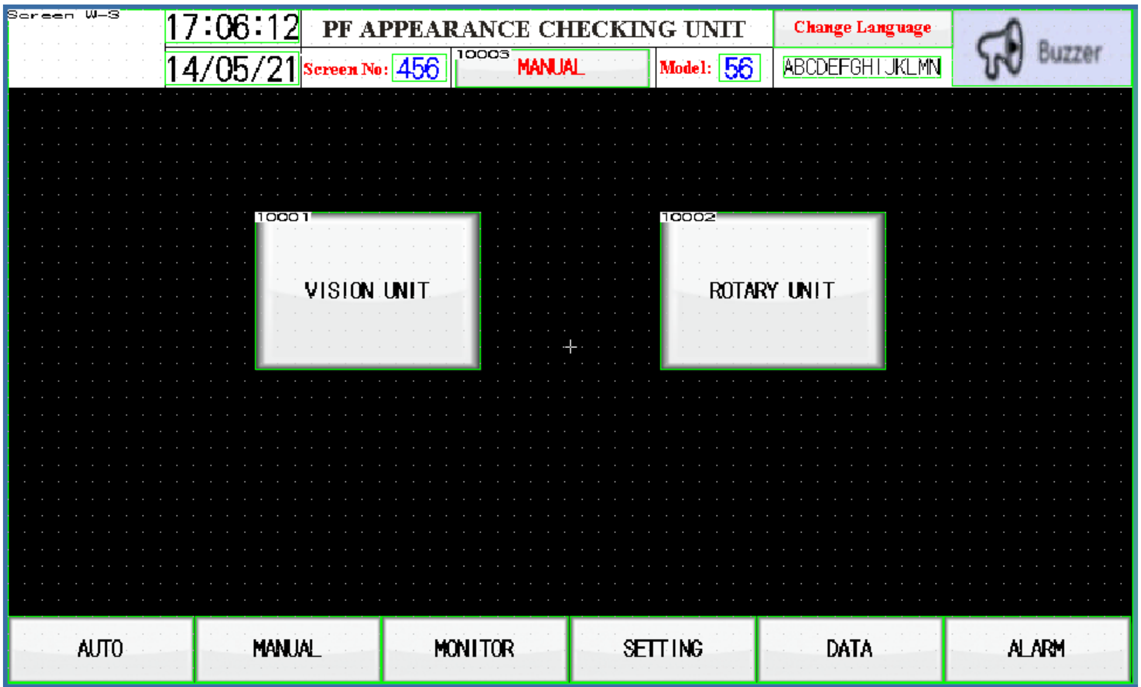

[PRG]R Write E_Output 41 ...
(26)
M2
PB HOME
Y014
PB LAMP
HOME
M200
HOME SYS
TEM
M200
M503
M8013
HOME SY
TEM
Origin M
ode
(33)
M2200
M1101
Ma. Cyl
Fix: FD
Org. Cyl
Fix: BK
Y015
SOL. FD/
BK FIX
M1200
Au. Cyl
Fix: FD
(38)
M3
PB RESET
Y011
Servo: A
RST
(40)
END

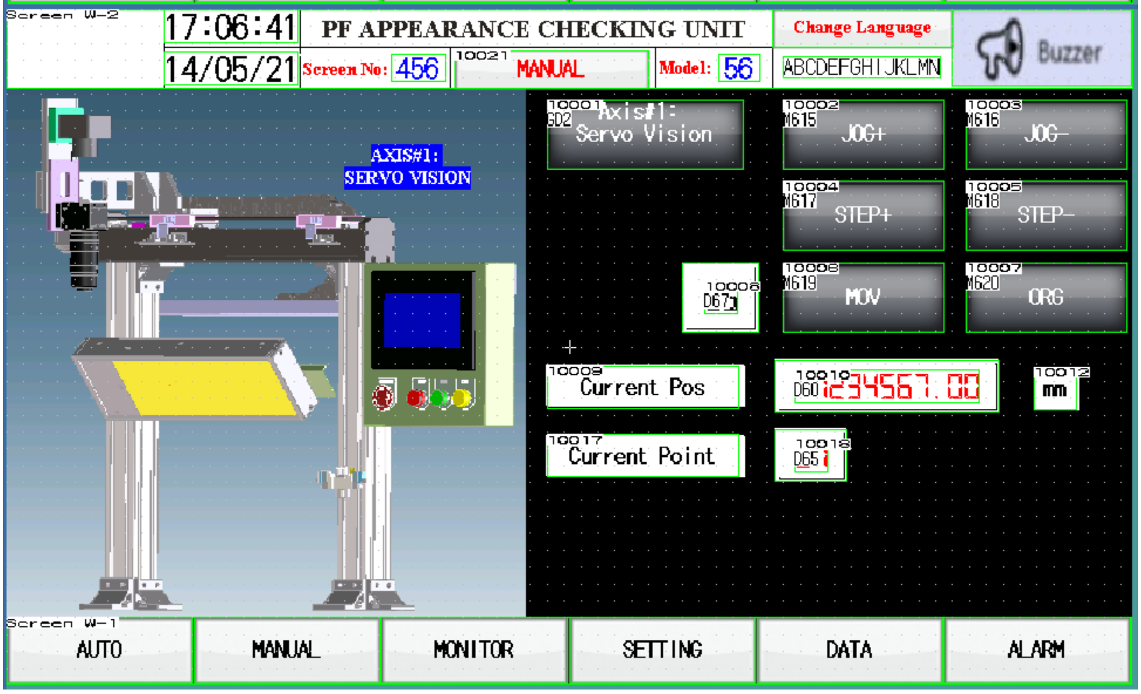
HMI

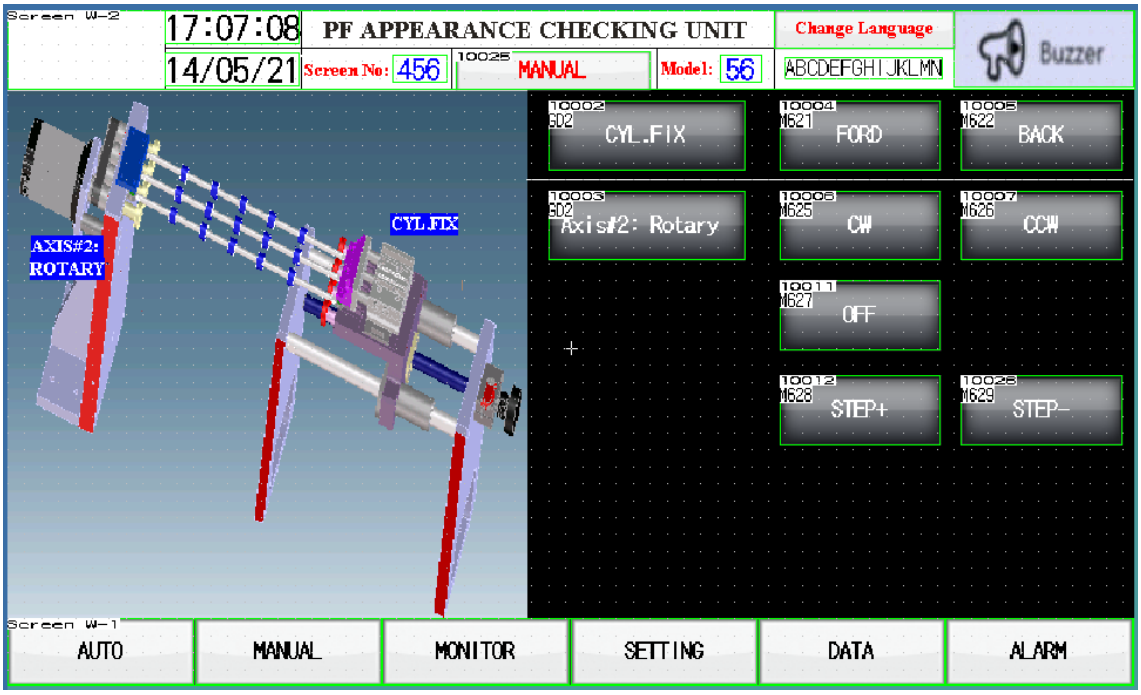
- Initial


17:05:06
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
AUTO
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
NG
OK
AXIS#1 TEST
Current Point
Current Pos
mm
AXIS#2
Total Pulse Rotary
Rotary per one Pos
Rotary Counter
HOME
AUTO MODE
MANUAL MODE
START
STOP
PAUSE
RETURN
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

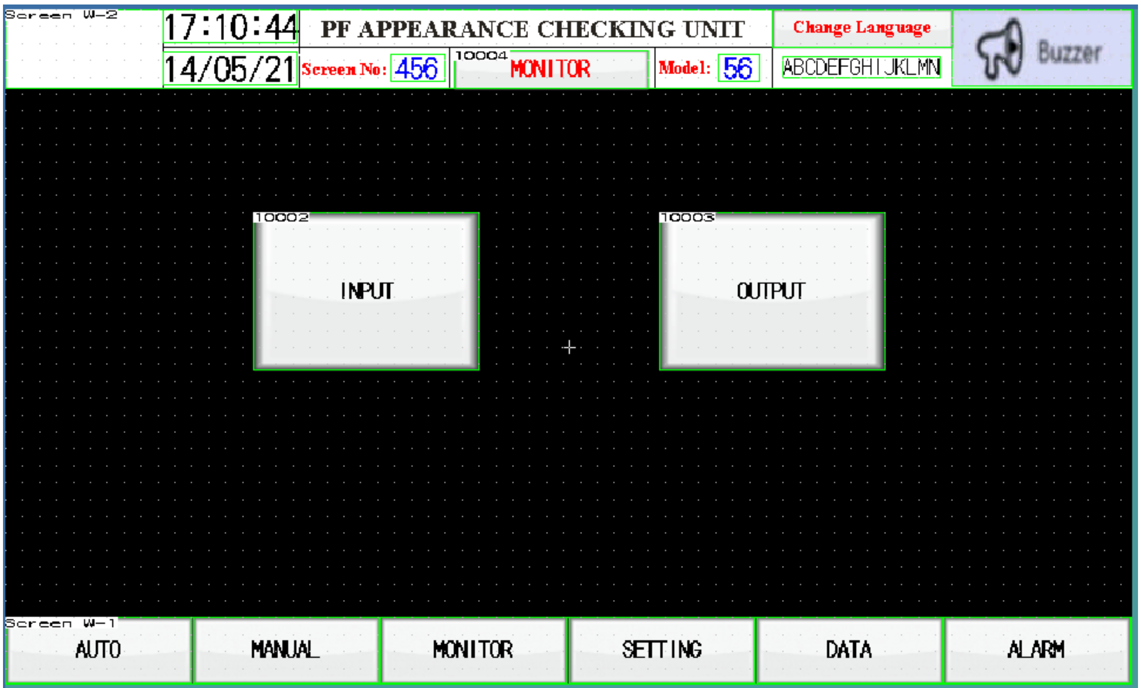
- Manual

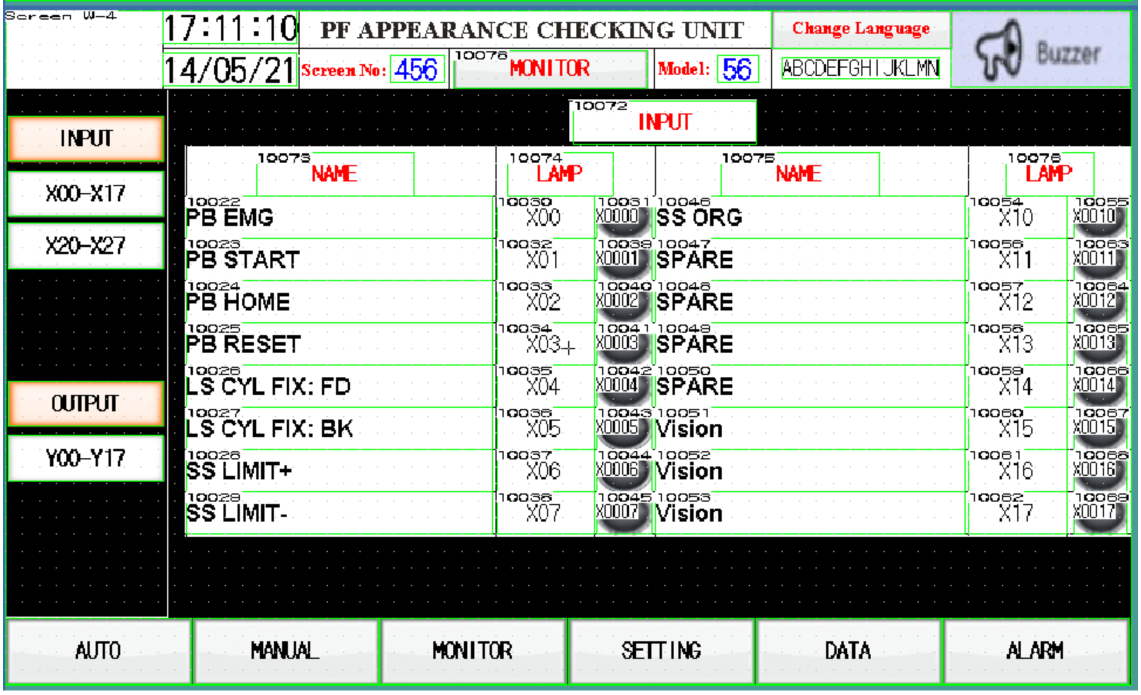

Screen W-3
17:06:12
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
MANUAL
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
VISION UNIT
ROTARY UNIT
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

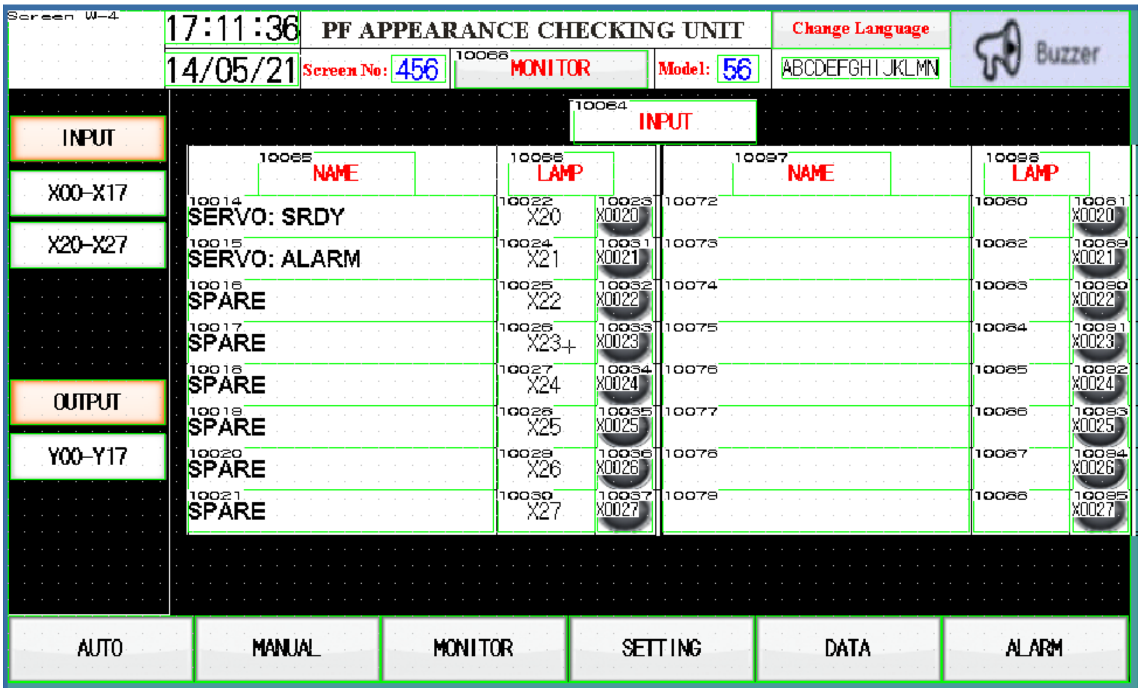

Screen W-2
17:06:41
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
MANUAL
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
AXIS#1:
SERVO VISION
Axis#1:
Servo Vision
JOG+
JOG-
STEP+
STEP-
MOV
ORG
Current Pos
mm
Current Point
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

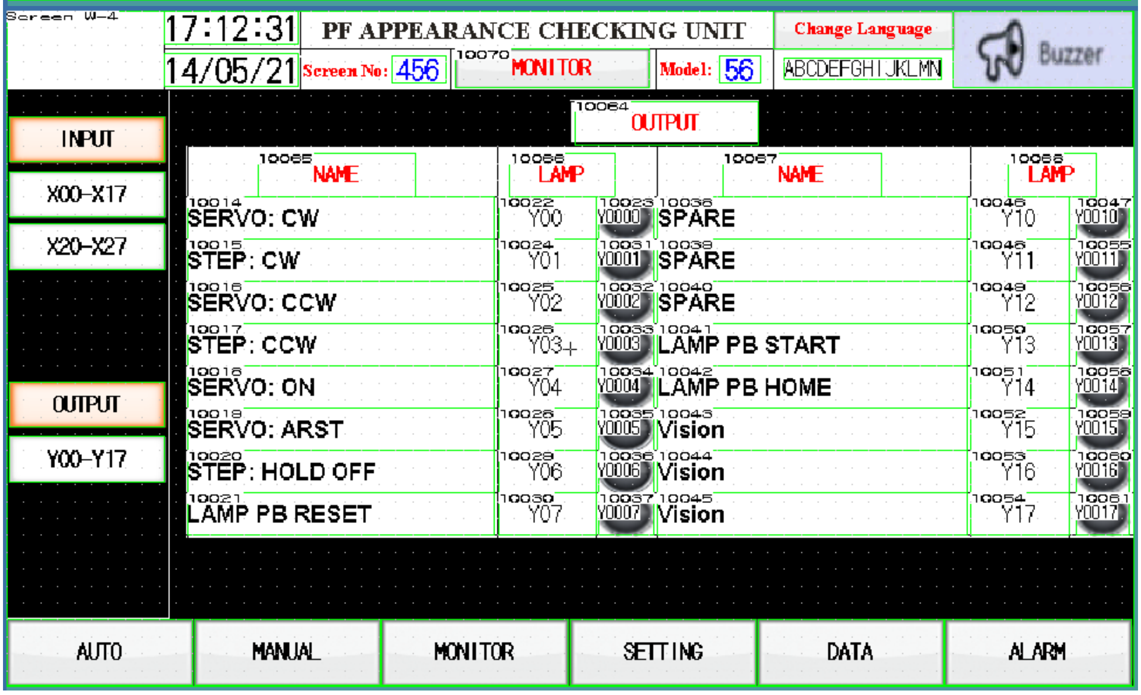

Screen W-2
17:07:08
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
ABCDEFGHIJKLMN
Buzzer
Screen No: 456
10025
MANUAL
Model: 56
AXIS#2: ROTARY
CYL.FIX
10002 GD2 CYL.FIX
10004 M621 FORD
10005 M622 BACK
10003 GD2 Axis#2: Rotary
10006 M625 CW
10007 M626 CCW
10011 M627 OFF
10012 M628 STEP+
10028 M629 STEP-
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

- Monitor:

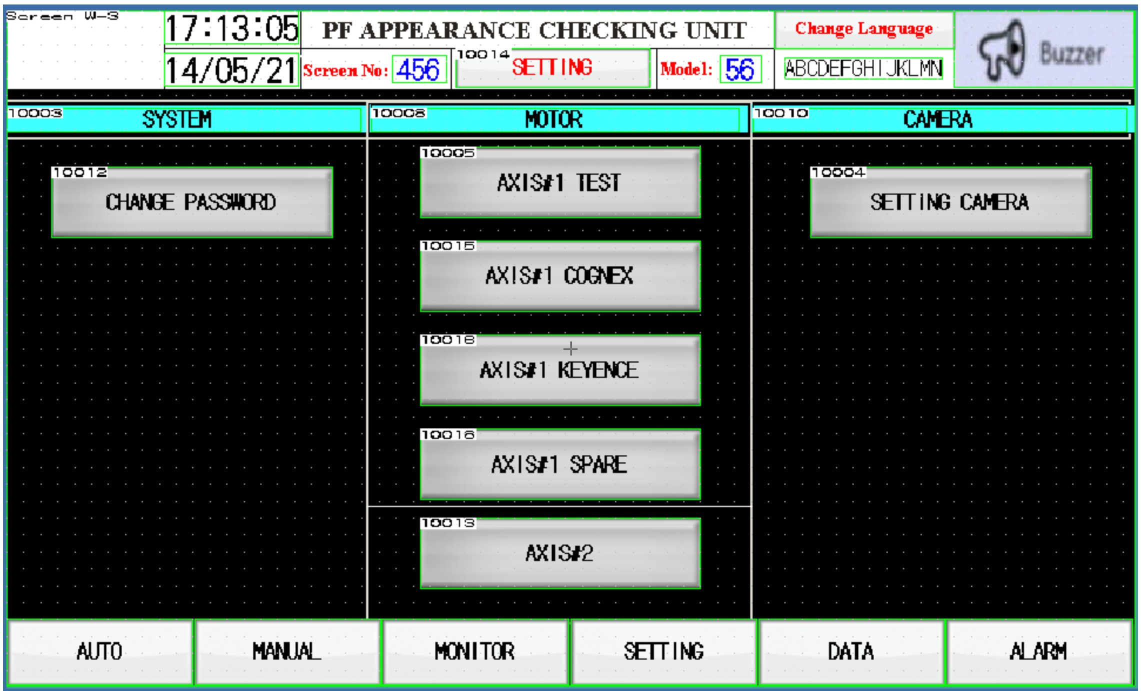

Screen W-2
17:10:44
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
MONITOR
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
INPUT
OUTPUT
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

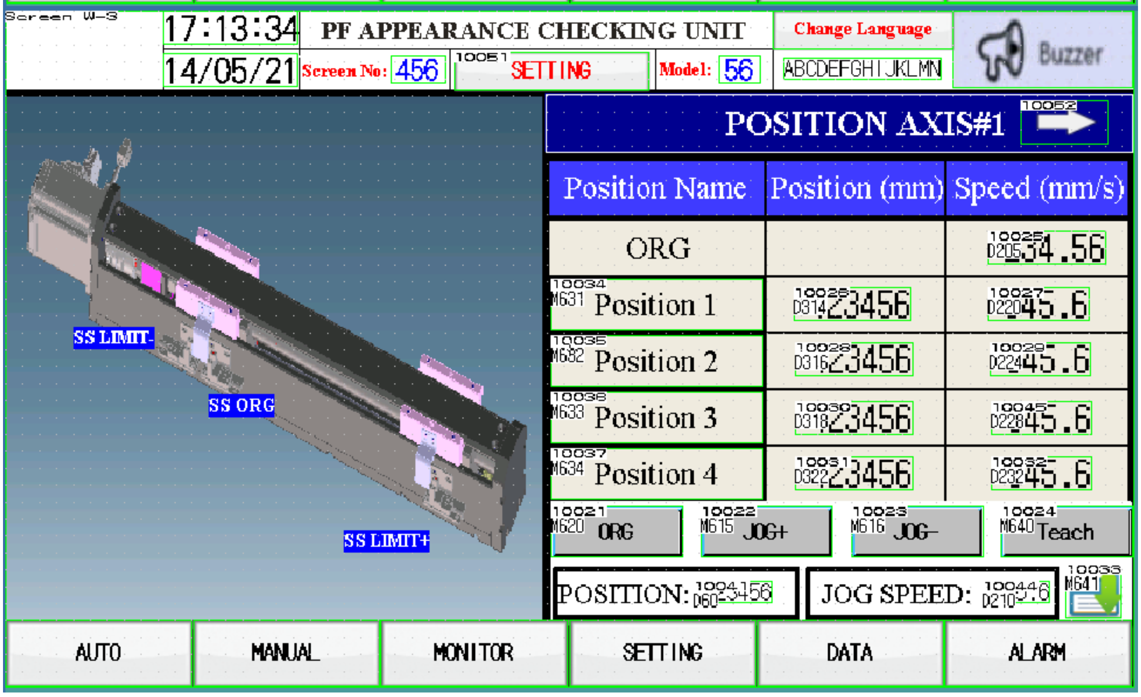

Screen W-4
17:11:10
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
MONITOR
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
INPUT
X00-X17
X20-X27
OUTPUT
Y00-Y17
NAME
LAMP
PB EMG
PB START
PB HOME
PB RESET
LS CYL FIX: FD
LS CYL FIX: BK
SS LIMIT+
SS LIMIT-
X00
X01
X02
X03
X04
X05
X06
X07
SS ORG
SPARE
SPARE
SPARE
SPARE
Vision
Vision
Vision
X10
X11
X12
X13
X14
X15
X16
X17
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

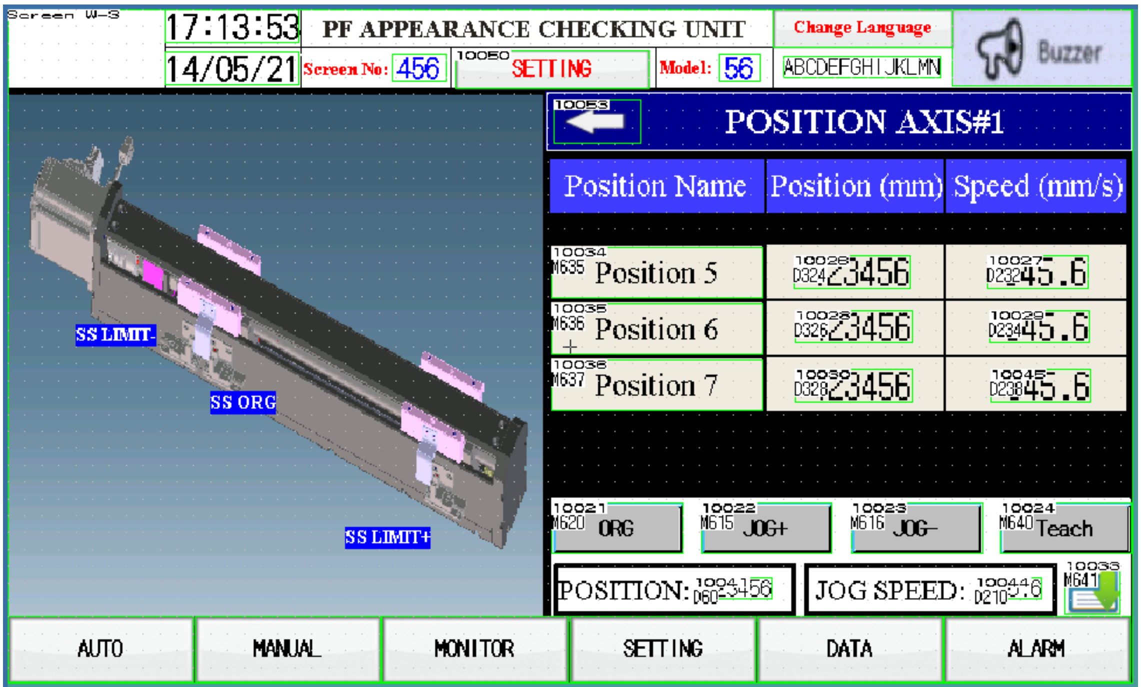

Screen W-4
17:11:36
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
MONITOR
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
INPUT
X00-X17
X20-X27
OUTPUT
Y00-Y17
INPUT
NAME
LAMP
NAME
LAMP
SERVO: SRDY
SERVO: ALARM
SPARE
SPARE
SPARE
SPARE
SPARE
SPARE
X20
X21
X22
X23
X24
X25
X26
X27
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM


Screen W-4
17:12:31
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
MONITOR
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
INPUT
X00-X17
X20-X27
OUTPUT
Y00-Y17
OUTPUT
NAME
LAMP
NAME
LAMP
SERVO: CW
STEP: CW
SERVO: CCW
STEP: CCW
SERVO: ON
SERVO: ARST
STEP: HOLD OFF
LAMP PB RESET
Y00
Y01
Y02
Y03
Y04
Y05
Y06
Y07
SPARE
SPARE
SPARE
LAMP PB START
LAMP PB HOME
Vision
Vision
Vision
Y10
Y11
Y12
Y13
Y14
Y15
Y16
Y17
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

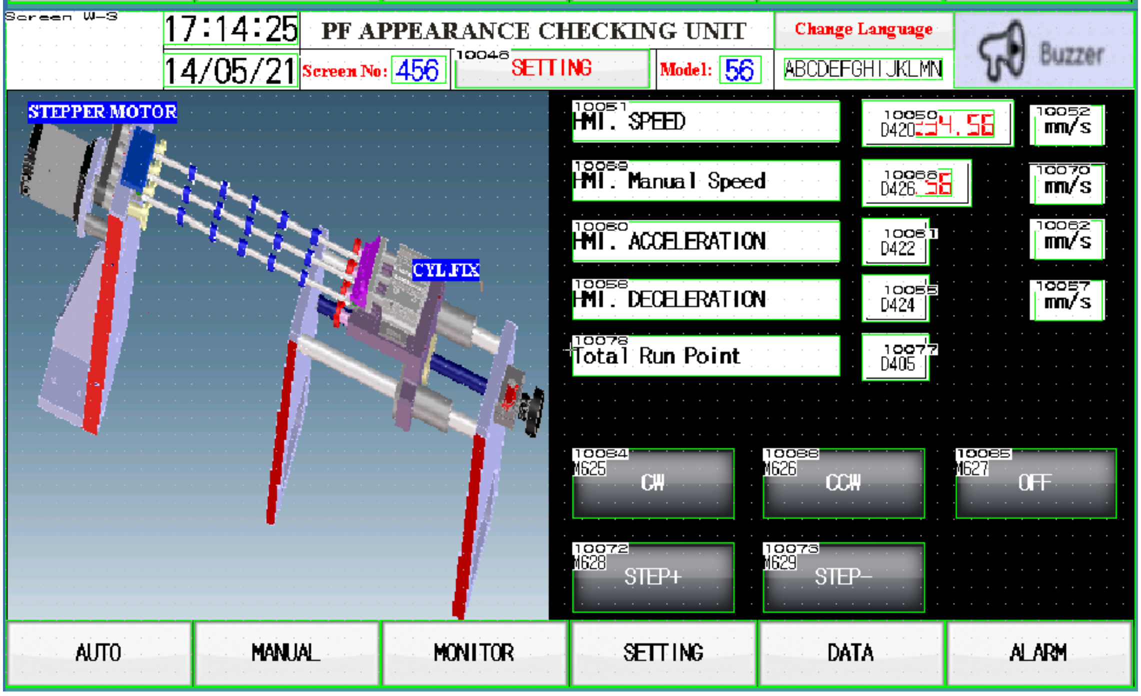
- Setting :


Screen W-3
17:13:05
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
SETTING
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
SYSTEM
MOTOR
CAMERA
CHANGE PASSWORD
AXIS#1 TEST
AXIS#1 COGNEX
AXIS#1 KEYENCE
AXIS#1 SPARE
AXIS#2
SETTING CAMERA
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM


Screen W-3
17:13:34
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
SETTING
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
SS LIMIT-
SS ORG
SS LIMIT+
POSITION AXIS#1
Position Name
Position (mm)
Speed (mm/s)
ORG
34.56
Position 1
123456
45.6
Position 2
123456
45.6
Position 3
123456
45.6
Position 4
123456
45.6
ORG
JOG+
JOG-
Teach
POSITION:
JOG SPEED:
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

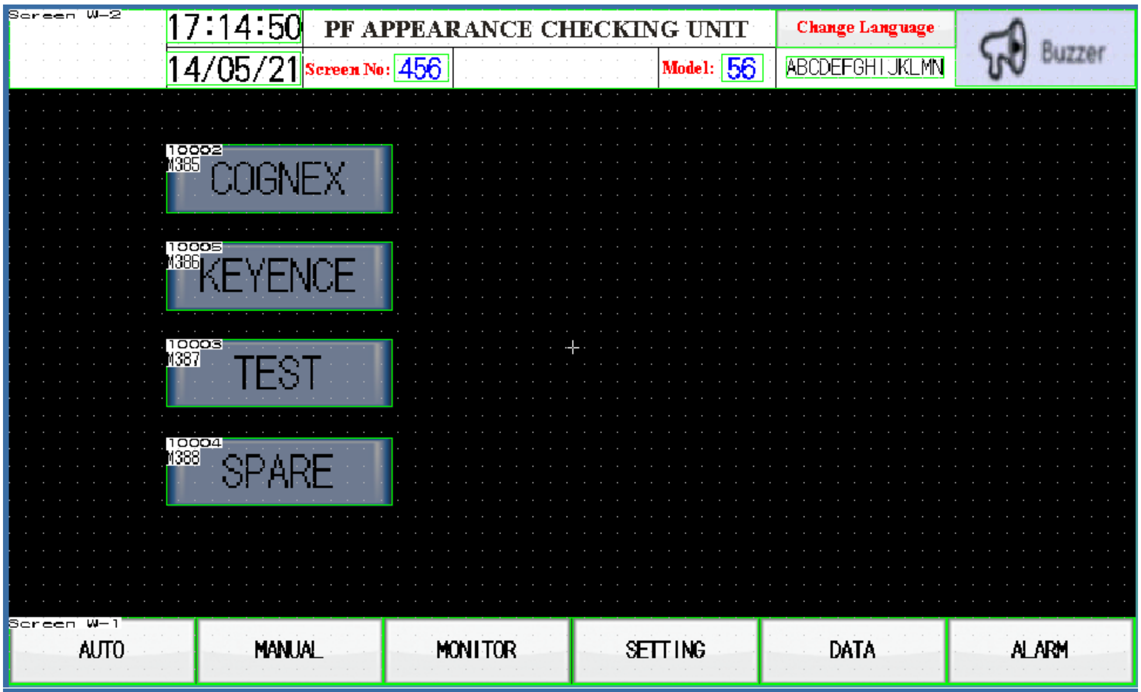


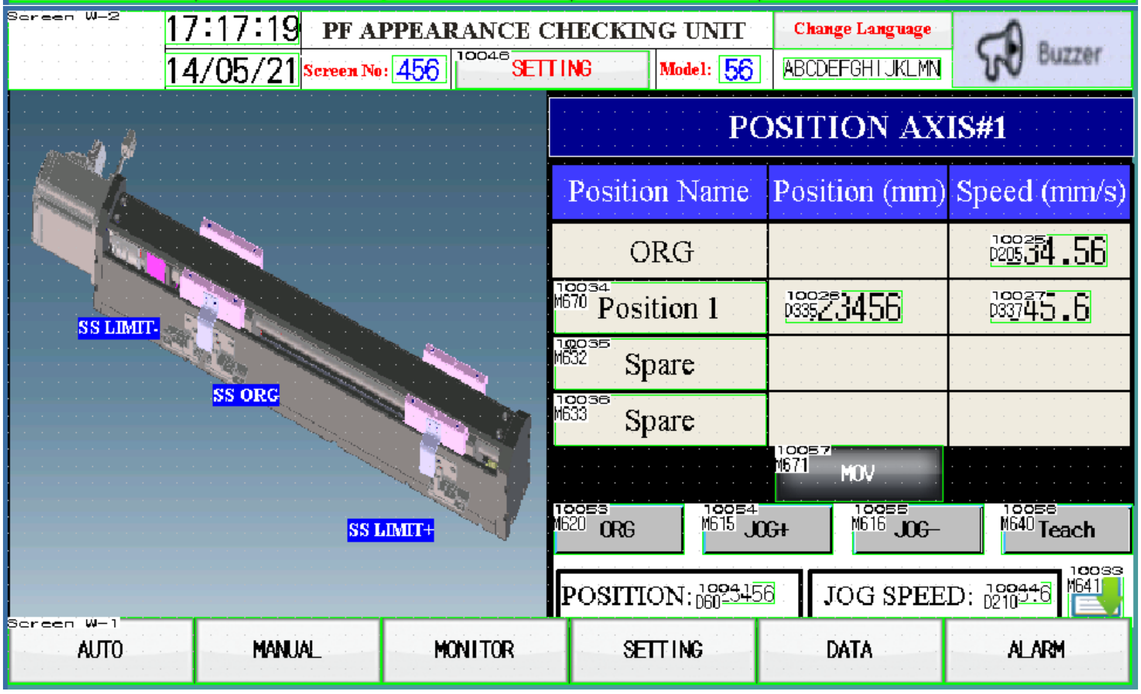
Screen W-3
17:13:53
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
Buzzer
Screen No: 456
SETTING
Model: 56
ABCDEFGHIJKLMN
POSITION AXIS#1
Position Name
Position (mm)
Speed (mm/s)
Position 5
Position 6
Position 7
SS LIMIT-
SS ORG
SS LIMIT+
ORG
JOG+
JOG-
Teach
POSITION:
JOG SPEED:
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM



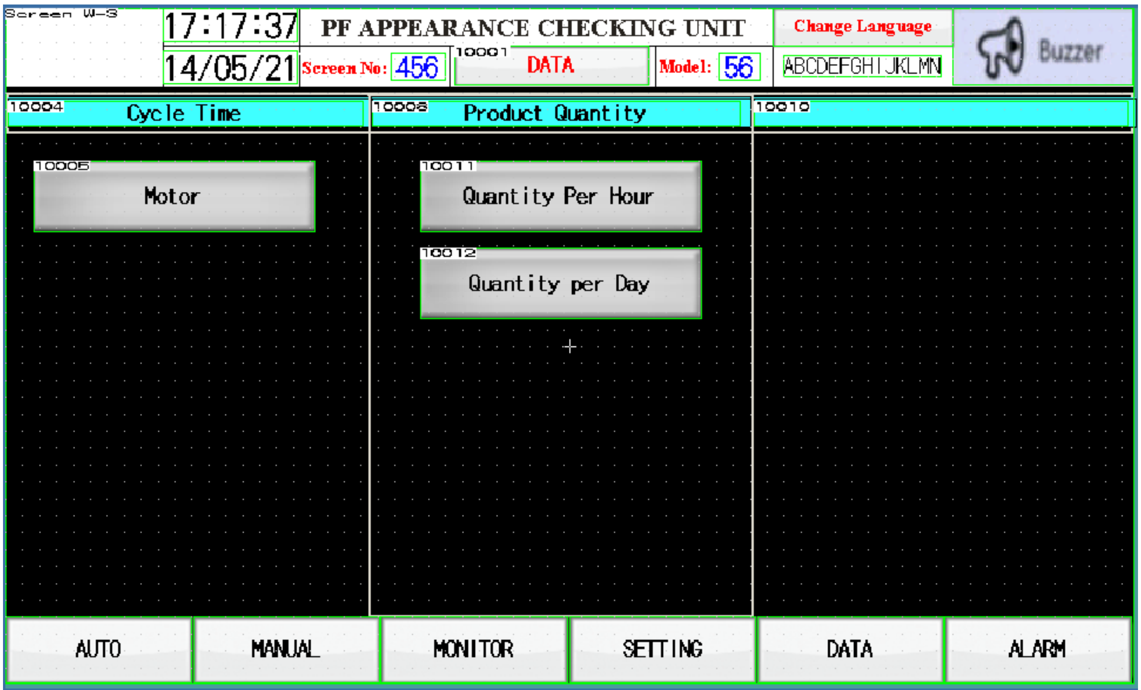
Screen W-3
17:14:25
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
Buzzer
Screen No: 456
SETTING
Model: 56
ABCDEFGHIJKLMN
STEPPER MOTOR
CYL FIX
HMI. SPEED
HMI. Manual Speed
HMI. ACCELERATION
HMI. DECELERATION
Total Run Point
mm/s
mm/s
mm/s
mm/s
CW
CCW
OFF
STEP+
STEP-
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

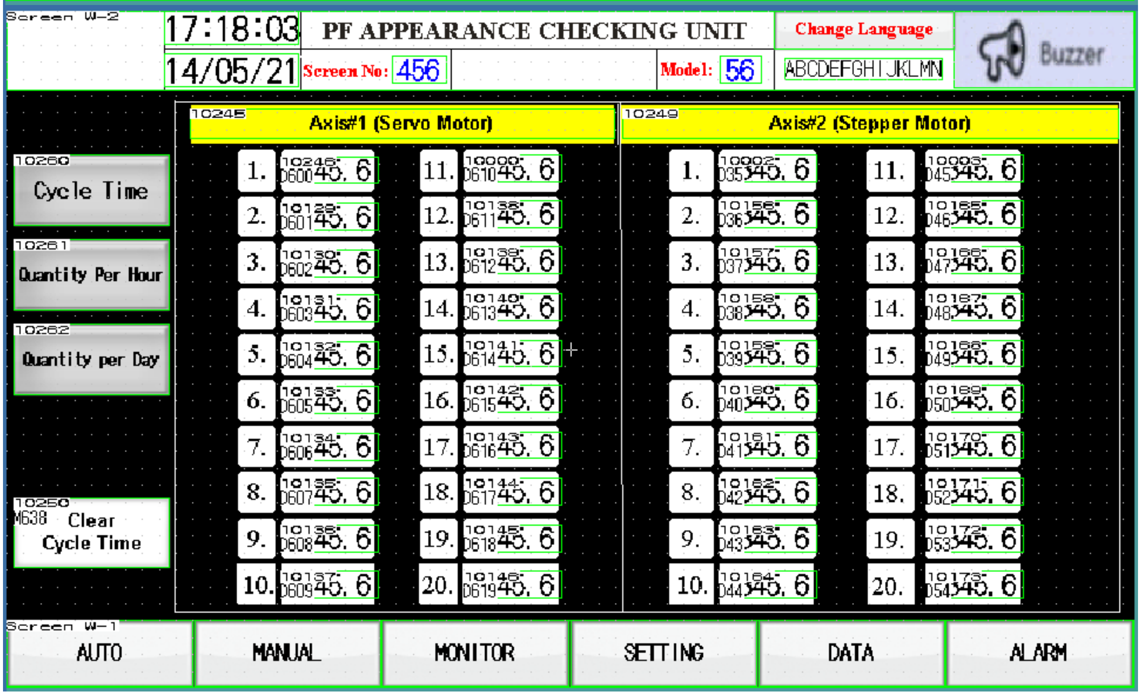
- Setting :

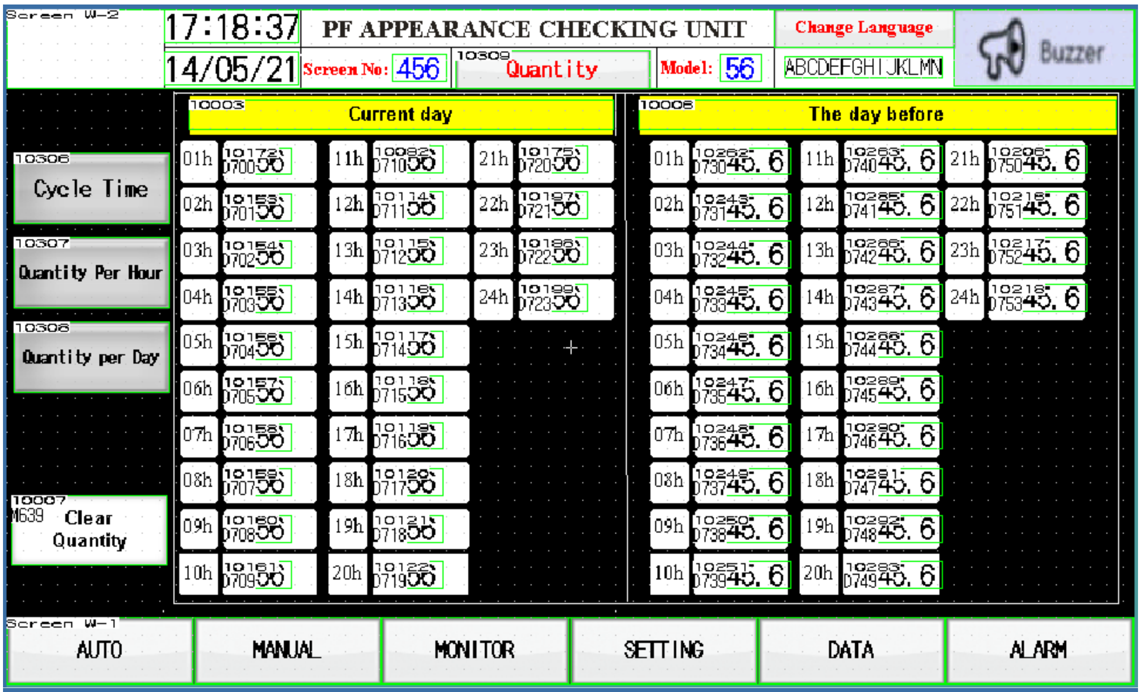

Screen W-2
17:14:50
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
Buzzer
Screen No: 456
Model: 56
ABCDEFGHIJKLMN
COGNEX
KEYENCE
TEST
SPARE
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

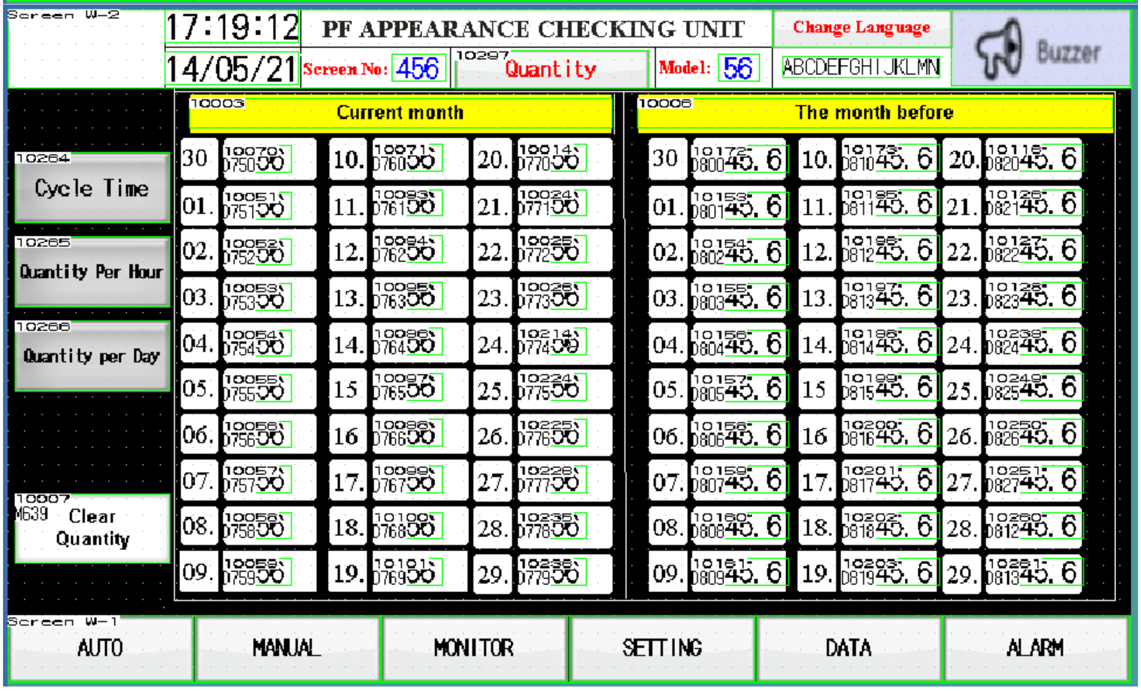

Screen W-2
17:17:19
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
Buzzer
Screen No: 456
SETTING
Model: 56
ABCDEFGHIJKLMN
POSITION AXIS#1
Position Name
Position (mm)
Speed (mm/s)
ORG
34.56
Position 1
23456
45.6
Spare
Spare
MOV
SS LIMIT-
SS ORG
SS LIMIT+
ORG
JOG+
JOG-
Teach
POSITION:
JOG SPEED:
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

- Data :

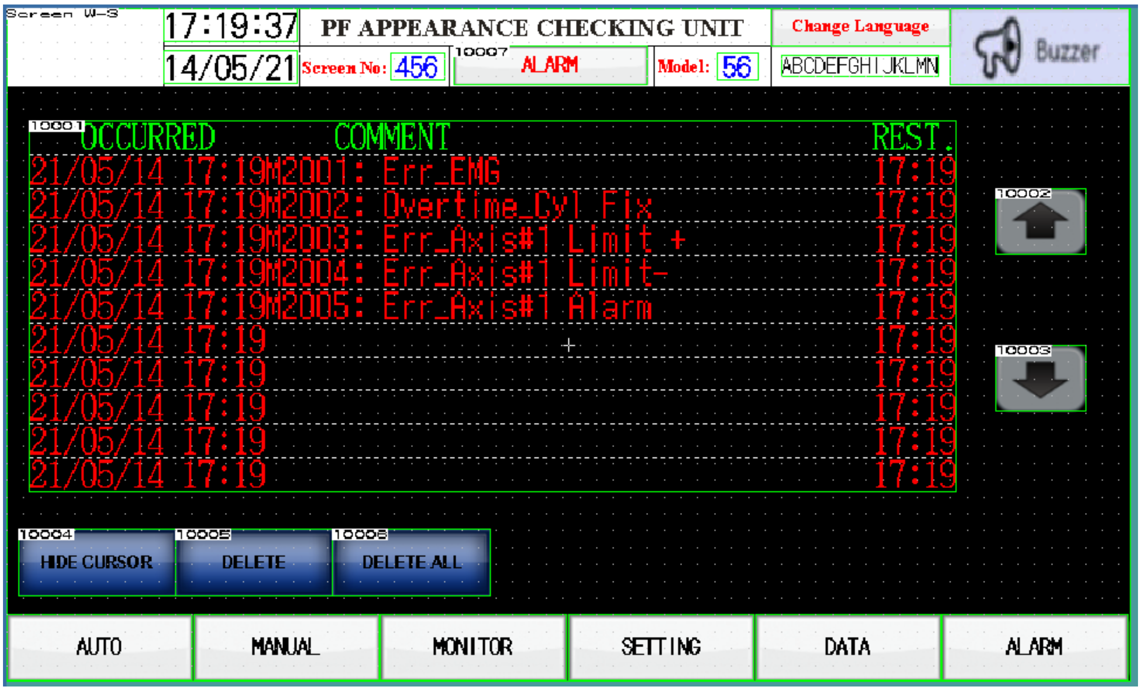

Screen W-3
17:17:37
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
DATA
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
Cycle Time
Product Quantity
Motor
Quantity Per Hour
Quantity per Day
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM


Screen W-2
17:18:03
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
Axis#1 (Servo Motor)
Axis#2 (Stepper Motor)
Cycle Time
Quantity Per Hour
Quantity per Day
Clear
Cycle Time
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

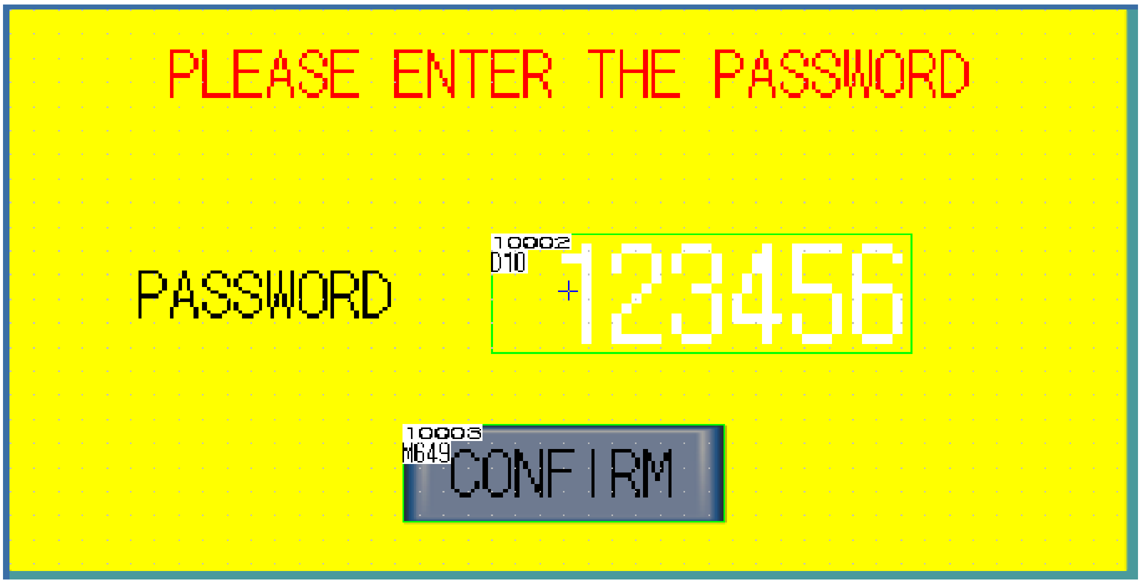

Screen W-2
17:18:37
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
Buzzer
Screen No: 456
Quantity
Model: 56
ABCDEFGHIJKLMN
Current day
The day before
Cycle Time
Quantity Per Hour
Quantity per Day
Clear Quantity
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM


Screen W-2
17:19:12
14/05/21
PF APPEARANCE CHECKING UNIT
Change Language
Buzzer
Screen No: 456
Quantity
Model: 56
ABCDEFGHIJKLMN
Current month
The month before
Cycle Time
Quantity Per Hour
Quantity per Day
Clear Quantity
Screen W-1
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

- Error :


Screen W-3
17:19:37
14/05/21
PF APPEARANCE CHECKING UNIT
Screen No: 456
ALARM
Model: 56
Change Language
ABCDEFGHIJKLMN
Buzzer
OCCURRED
COMMENT
REST.
21/05/14 17:19 M2001: Err_EMG 17:19
21/05/14 17:19 M2002: Overtime_Cyl Fix 17:19
21/05/14 17:19 M2003: Err_Axis#1 Limit + 17:19
21/05/14 17:19 M2004: Err_Axis#1 Limit- 17:19
21/05/14 17:19 M2005: Err_Axis#1 Alarm 17:19
21/05/14 17:19 17:19
21/05/14 17:19 17:19
21/05/14 17:19 17:19
21/05/14 17:19 17:19
21/05/14 17:19 17:19
HIDE CURSOR
DELETE
DELETE ALL
AUTO
MANUAL
MONITOR
SETTING
DATA
ALARM

- Password:


PLEASE ENTER THE PASSWORD
PASSWORD
D10
123456
M649
CONFIRM