PRODUCT FAMILY: Sure Servo **Number:** AN-SERV-004

Subject: Sureservo multiple position with MODBUS **Date issued:** July-30-2010

Revision: 4rd Edition

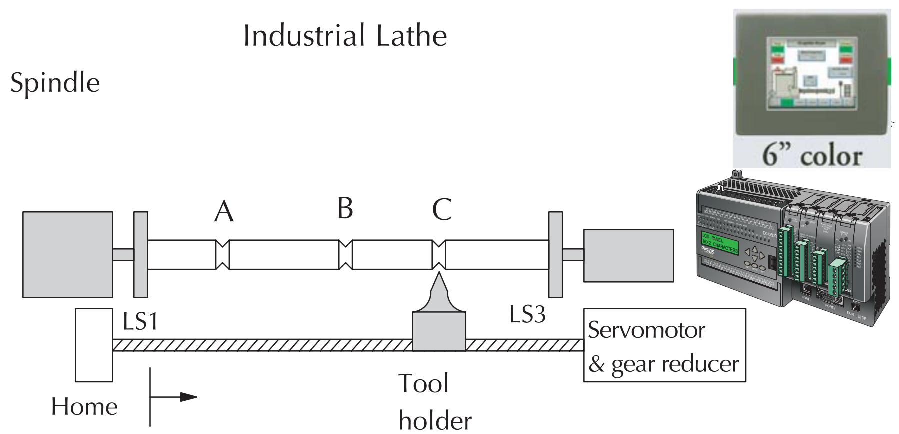
(4th edition updates the cables between PLC and servo and another minor typos)

On this application note we will control an industrial lathe to execute 3 grooves on the material of the spindle by means of programmable position targets on the *Sure*servo drive, with variable distance defined with a Touch panel C-more of the type EA7-S6C, as well we demonstrate the use of MODBUS communication.

This document assumes that you already have a working knowledge with programming the PLC DL06 and using *Direct*SOFT, as well as the C-more touch screen.

For this example we have determined the kinematics of the movement, the sizing of the servo motor and design considerations on the application note AN-SERV-001. This application note is an upgrade of the referenced document. We will wire the servo drive to the PLC DL06. We will show the wiring of the drive, the program of the PLC and the touch screen panel and the programming of the servo drive.

See the following diagram to explain the concept.



LS1 and LS3 are the overtravel limit switches

The normal position of the tool holder is the home position. It may happen that the system gets powered off while the tool holder is in an intermediate position. The PLC detects this fact and on start, the tool holder will move to Home.

The part installed on the spindle of the lathe will receive 3 grooves that will be located on positions A, B and C, freely defined in distance from a Home position.

To recall basic data from the application motor AN-SERV-001, the pitch of the lead screw is 5 revolutions per inch (or a lead of 0.2 inch/revolution). The lead screw cannot run more the 500 rpm, per manufacturer limitations. Gear reducer ratio is 7:1.

The lead screw that displaces the tool is 108 inches long and one of the sides is coupled to the servo motor, thru a gear-reducer.

The operation is the following:

At the start of the job, the servo motor brake is applied; when the **Start** button on the touch panel is touched, the brake is released and the the lead screw will position the tool holder at Home, if not in **Home** position. After that, the system will move the tool holder to position A; the spindle will begin to rotate; then the tool will advance while the spindle rotates until the groove is done. The spindle will continue to rotate; the tool will return to retracted position, at home.

Position sensors in the tool holder will detect the position of the tool and will not allow the servo to move unless the tooling is retracted. The servo's brake is applied during this time. Using the same procedure, the tool holder will repeat the sequence at position B and Position C.

At the end of the cycle, the tool holder will return to the home position, to wait for next cycle. The spindle will stop rotating.The positions A, B and C are defined as distance in inches from the home position,with 3 decimals of precision.

There is a proximity sensor to indicate the Home position to the drive.

Servomotor position concept

The most important concept here is that the servo parameter P1-01 is set as 101, which defines internal register position with forward rotation defined as clockwise rotation. The home position is set with the parameter P1-47 and will do the following:

1 2 2 3

- The move to home will be move reverse to dedicated home sensor
- It will position with the home sensor, instead of z index mark
- Home trigger source is done by input terminal (DI4)
- Stop position as 1 means that after detecting home position, the servo motor will decelerate and stop in the forward position

The positioning with internal register may have absolute or incremental mode.

We have elected to use incremental mode, because sometimes the servo power up may define a different zero position. The use of a home routine will help to start always from the same point.

Math for defining the positions of the grooves.

Since one revolution of the servo motor shaft corresponds to 10000 counts on the encoder, and 1 inch is equal to 35 revolutions, then:

- The maximum allowable position is 108[inch] and 0 counts.
- The position A should be less than the position B and also the position B should be less than the position C.

The data entry will be accomplished on the C-more panel by creating a numeric entry for each one of the positions. See page 7 for the screens in the C-more panel.

The data to be written to the servo drive thru MODBUS are the positions A, B and C, that in fact will be written from the registers V3200 to V3206 to the parameters P1-15 to P1-20, corresponding to the preset revolutions and preset pulses.

These 6 mentioned parameters have the MODBUS addresses 40272 until 40277 decimal on the slave.

a) **Position A**: The data will be written on the C-more panel as a 32 bit BCD number. Let us say, just for doing an example, that the first position is 30 inches and 345 thousands of an inch.

This will be stored on V3000 as a BCD double word. Then V3000 will contain the number 3450 and the register V3001 will contain the number 030 inches. This data will be scaled on the touch screen panel to show the real servo shaft revolutions and counts on the double word V3200.

b) **Position B**: The data will be written on the C-more panel as a 32 bit BCD number. This will be stored on V3002 as a BCD double word. This data will be scaled on the touch screen panel to show the real servo shaft revolutions and counts on the double word V3202.

c) **Position C**: The data will be written on the C-more panel as a 32 bit BCD number. This will be stored on V3004 as a BCD double word. This data will be scaled on the touch screen panel to show the real servo shaft revolutions and counts on the double word V3204.

The operation to translate the unit inches into revolutions and counts is done with the following concept:

108 inches will correspond to 108x5 revolutions of the lead screw = 540 revolutions. One revolution of the lead screw corresponds to 7 revolutions of the servomotor shaft. Then the data of the touch screen, in inches, should be multiplied by 35 to get the proper displacement, in revolutions. This calculation is done scaling this data in the touch screen panel).

We should assure that there is always an incremental distance between positions of at least 1 inch and not a zero as well as the increments do not add more than 108 inches.

The touch screen panel will have 2 screens; the first screen will have the operator main interface, for starting and stopping the sequence, as well as showing the status of the operation. There is a button to go to the second screen.

The second screen will have the interface to enter the preset positions and for that it have created 3 numeric entry objects , that allow to define the distance directly in inches. There is a button to go to the main screen.

The C-more panel is connected to the PLC with serial communication by using the K-sequence protocol to the port 1 and using the cable EA-2CBL

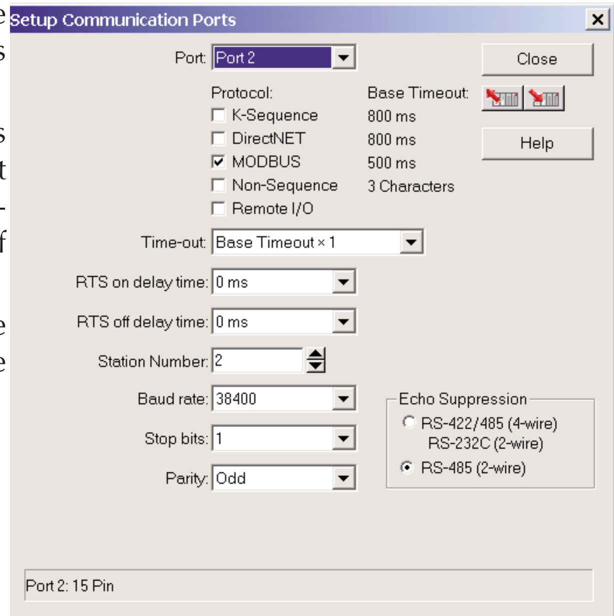
The trigger to transfer the data is the button C3 on the touch panel. It is not necessary to transfer data continuously. Notice that the reading is interlocked with the writing with the contact C3.

See more details on page 7, regarding the programming of the C-more panel.

Concepts of MODBUS communications

We will define the servo as slave 2 and we will establish a baud rate of 38.4 kBaud, linked thru the port 2 of the PLC, using the MODBUS RTU protocol with the RS-485 mode.

The comm. port 2 should be configured according to the dialog box shown on the adjacent figure. Notice that the maximum baud rate on the PLC is 38.4 kBaud.



If more communication speed is necessary, for other applications, it is possible to use the module D0-DCM, that allows a transfer rate of 115.2 kBaud.

On the Servo drive, we **must** set the following parameters to match the settings on the PLC:

P3-00= 2 (station o slave address)

P3-01=3 (baud rate)
corresponding to 19200 kbps

P3-02=8 (data,parity,stop bits)

P3-05 = 2 for RS-485

The rest of the group 3 parameters can be left as default.

P0-09 = 021E to allow the parameter P2-30 to be overwritten

P0-10 = 010F to allow the parameter P1-15 to be overwritten

P0-11 = 0110 to allow the parameter P1-16 to be overwritten

P0-12 = 0111 to allow the parameter P1-17 to be overwritten

P0-13 = 0112 to allow the parameter P1-18 to be overwritten

P0-14 = 0113 to allow the parameter P1-19 to be overwritten

P0-15 = 0114 to allow the parameter P1-20 to be overwritten

It is important to define the parameter P2-30 as 5. The reason is that if not 5, the servo will write to the EEPROM memory, and this could be damaged. Selecting a 5 there allows to write to RAM memoria. That is why we created a routine to check, at the beginning of the operation, that the parameter P2-30 is 5. It if is not 5, the PLC will force a value of **5** into it. The MODBUS address will be 021E, hexadecimal or 40543 decimal.

The other parameters P0-10 to P0-15 are necessary to transfer the preset positions to the proper register into the servo drive. If not, the PLC code will not work properly.

A link to be used on port 2 could be done using the cable SVC-485HD15-CBL-2; this cable has a VGA connector in one side, for the PLC side, port 2 and a proper connector on the servo side, that connects to the port CN3 of the servo drive.

The data for preset revolutions and preset pulses to be written to the servo drive thru MODBUS RTU are the positions A, B and C, that in fact will be written from the registers V3200 to V3206 to the parameters P0-10 to P0-15, corresponding to the block transfer parameters.

The trigger to transfer the data is the button C3 on the touch panel. It is not necessary to transfer data continuously. Notice that the reading is interlocked with the writing with the contact C3. That is,ot can only happen one reading or one writing at any given moment.

The current information will be read **continuously** from the status monitor parameters P0-04 to P0-07, defined as follows:

P0-04= 00; motor feedback pulses;

P0-05= 01; motor feedback revolutions;

P0-06= 06; motor feedback rpm;

P0-07= 11; motor feedback torque %;

These 4 parameters have the MODBUS addresses 40005 until 40008 decimal.

On the program, the read data will be stored on the registers V3300 to V3303.

Then - V3300 will have the current pulses position.

- V3301 will have the current revolution position.
- V3302 will have the current rpm.
- V3303 will have the current torque.

Data will be shown on the numeric display objects of the touch panel on the screen 2.

PLC I/O definition

The next step will be to define the PLC command control.

The concept is described in detail at the right side of the ladder diagram.

The touch screen panel has the screen 2 to define the distance for every one of the positions, directly in inches. The operator should enter the desired positions A, B and C before the start signal is given. When the button "press to transfer data" is touched the data is moved from PLC to the servo drive. See page 7 for more details.

The lathe operator has available 2 pushbuttons on the screen 1 of the panel to give the command to start the cycle and the other, stop, to return to Home, in case there is a power shutdown during the operation or just for stopping the cycle.

On the PLC we will select the following functions and the corresponding outputs:

C1 is the start command to initiate the movement and is given thru the touch panel with the object button START on the screen 1. C2 is the STOP.command and is also given thru the touch panel with the button object STOP on the screen 1.

X3 is the **Tool in retracted position,**wired directly from a sensor located in the tool holder. X22 is **Home completed,** generated by the output D02 coming from the servo drive outputs and X23 is the **At position,** generated by the output D01.

Y10 will have the command **Servo Enable** and go to the servo digital input DI1.

Y11 will have the signal **Move to Home** and go to the servo digital input DI4**.**

Y2 will have the command **Trigger to Start** and go to the servo digital input DI2**.**

Y3 is linked to DI8 and Y4 linked to DI7 and will have the signals to define the target position.

The **FWD overtravel and** the **REVERSE overtravel** limit switches are wired directly to the drive in the digital inputs DI3 and DI5, as well as the **home sensor** in DI6.

Y5 is the servomotor brake coil and the output.

Y6 is the command run to the lathe spindle.

Y7 is the tool start (To begin the groove) and is connected to other part of the system, not shown here.

We show below a summary of the inputs and outputs on the PLC.

I/O	Definition	Comments
X3	Tool retracted	Proximity sensor located on the tool holder
X16	Servo ready	Signal coming from the servo output D04
X21	Home sensor	Proximity sensor located on the lathe
X22	Home complete	Signal coming from the servo output D03
X23	Position completed	Signal coming from the servo output D02
Y2	Trigger to start	Signal goes to servo input DI2
Y3	Position bit 0	Signal goes to servo input DI8
Y4	Position bit 1	Signal goes to servo input DI7
Y5	Servo motor brake	Energizes to release the servo brake
Y6	Run spindle	The lathe spindle will turn when activated
Y7	Start tool	The tool holder will move into the material
Y10	Servo Enable	Signal goes to servo input DI1
Y11	Search home	Signal goes to servo input DI4

The following internal bits for touch screen operation are considered:

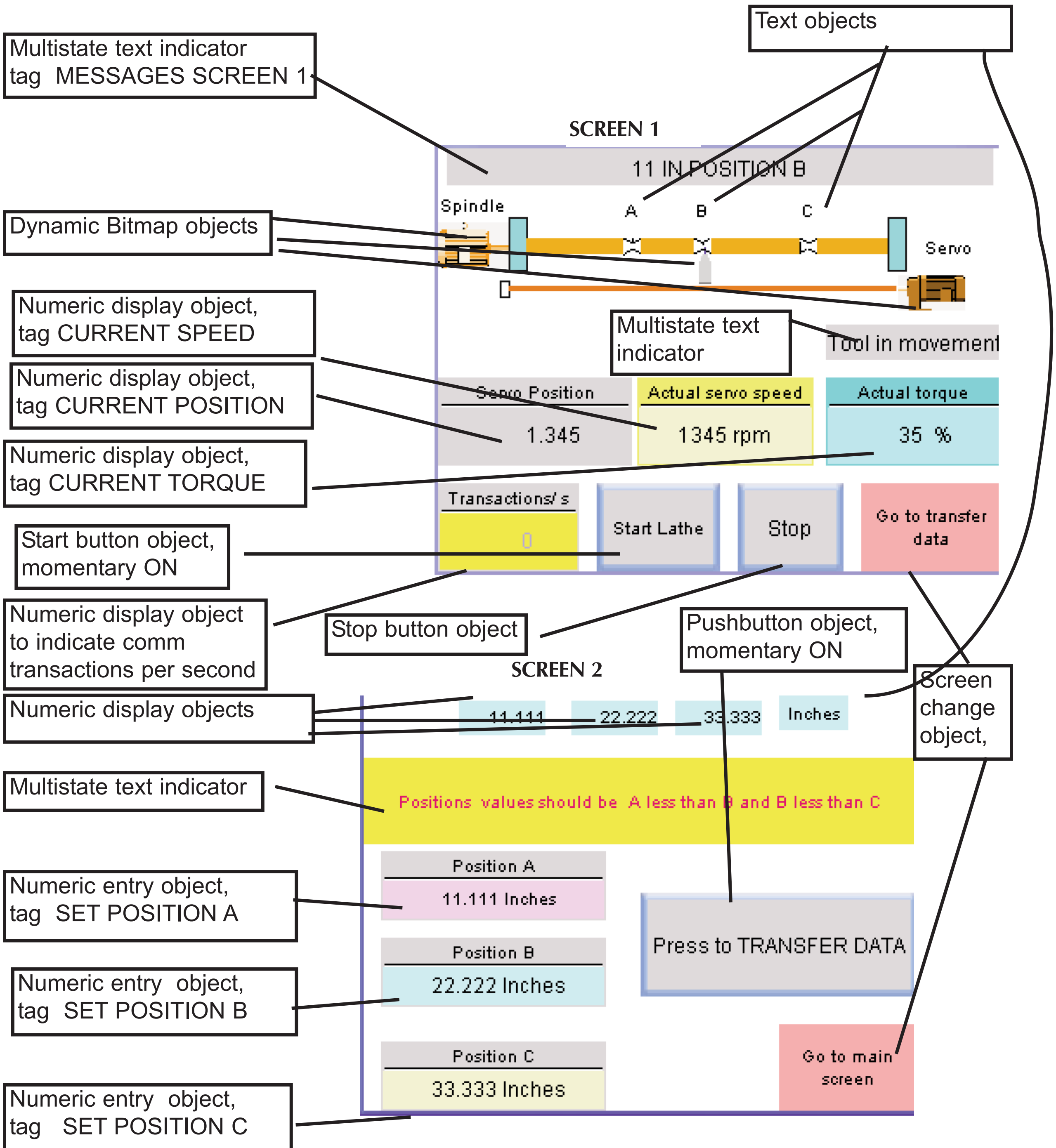
C1	Start command	related to an object on the touch screen
C2	Stop command	related to an object on the touch screen
C3	Data transfer command	related to an object on the touch screen

Touch screen programming

The touch screen panel will have 2 screens, one to start and stop the lathe operation and other to configure the predefined positions A, B and C.

The communication implemented on this example is serial with K-sequence protocol, and the panel is connected to the port 1 of the PLC.

See below for a summary of the objects created; on next pages there is some other details of the objects created on the touch screen,as well as the expected operation.



The touch panel has "objects" to execute different functions.

The objects have associated one tag to execute each one of the operations. The definition of the position is done with the screen 2. This screen can be reached from the screen 1 with a button to CHANGE SCREEN.

The operator will press the pushbutton START and the PLC will command the servo to move the tool holder to Home, if still not in Home.

When the tool holder is at home, the PLC will select the position A, by setting the digital inputs DI8 (Preset position 0 bit -PLC output Y3) and DI7 (Preset position 1 bit - PLC output Y4) to stay OFF.

When this condition is reached, the PLC gives automatically the command trigger to move to position A. The servo brake is released. At certain point after the servo shaft rotates, when the position A is to be reached, the servo shaft begins to stop before that point; then the servo will stop and will give a confirmation "At position"with the digital output D02, that corresponds to PLC input X23.

Next the tool will move to execute the groove on the piece of the lathe. The spindle rotates. When the tool retracts, a sensor tells the PLC that the groove is done (with the input X3) and the next movement could be triggered. In the same way, the position B and C will be reached. At this time the tool holder will do the same operation. Of course, for position B, the digital input Y3 should turn ON. For position C, the digital input Y4 should turn ON.

At the end of the cycle, the tool holder will move back to the Home position and will turn off the cycle.

These are the the tags used on the C-more panel on this project

Tag No	Tag Name	Tag Data Type	PLC Address
1	ACTUAL TORQUE	Signed int 16	V3404
2	Move to A	Discrete	C101
3	In position A	Discrete	C200
4	In position B	Discrete	C201
5	In position C	Discrete	C202
6	CURRENT POSITION	BCD int 32	V3400
7	CURRENT SPEED	Signed int 16	V3402
8	**DATA TRANSFER**	Discrete	C3
9	HOME SENSOR	Discrete	X22
10	INVALID POSITIONS	BCD int 16	V3701
11	MESSAGES SCREEN 1	BCD int 16	V3710
12	POSITION B	BCD int 32	V3002
13	POSITION C	BCD int 32	V3004
14	PRESET POSITION A	BCD int 32	V3000
15	SERVO WORKING	Discrete	Y5
16	SPINDLE	Discrete	Y6
17	**START**	Discrete	C1
18	**STOP**	Discrete	C2
19	TRANSACTIONS/S	BCD int 16	V3600
20	V3700	Signed int 16	V3700

The panel screen have the following functions through the objects

Screen 1 (Main screen)

- Starting and stopping the operation (pushbuttons Start and Stop),
- Display the status of the servo and the spindle showing it as a graphical interface, by indicating when the tool holder is in position and when the tool is doing the groove. The piece is green when rotating and orange when stopped. A dynamic bit map will show the tool holder on the proper position when stopped and penetrating the material.
- The panel displays servo position, speed and torque at any time,
- Show the communication transactions per second (to verify that there is communication in good condition),
- The indication of which step of the operation it is, in plain English (using V3710) by using a MULTISTATE TEXT INDICATOR object . The messages are:

1	1 Ready to start; checking P2-30
2	2 Ready to set positions; EEPROM not being written
3	3 Let us set new positions
4	4 New positions defined
5	5 Moving to Home
6	6 Defining position A
7	7 Moving to position A
8	8 IN POSITION A
9	9 Defining Position B
10	10 Moving to position B
11	11 IN POSITION B
12	12 Defining position C
13	13 Moving to position C
14	14 IN POSITION C
15	15 END=---- go to home
16	16 Home and stop

- Indicates if the tool holder is retracted or in movement.
- A button to CHANGE SCREEN to be able to define the new desired positions.

Screen 2 (New position screen)

- 3 NUMERIC ENTRY objects to change the positions, in inches, with 3 decimals of precision
- PUSHBUTTON to enter the 3 new positions when touched.
- Display the status of the new positions.
- A MULTISTATE TEXT INDICATOR to generate an alarm if the positions are not separated for at least one inch and that the addition of the displacements are not over 108 inches
- A button to CHANGE SCREEN to be able to go to the main screen.

Wiring between PLC and servo drive

On the PLC we have selected the following functions and the corresponding output:

C0 is the start command to initiate the movement. C1 is the trigger to home. C2 is STOP. X3 is the **Tool in retracted position**. X22 is **At Position** and X23 is the **Home completed**.

Y11 will have the signal **Move to Home.** Y10 will have the command **Servo Enable**

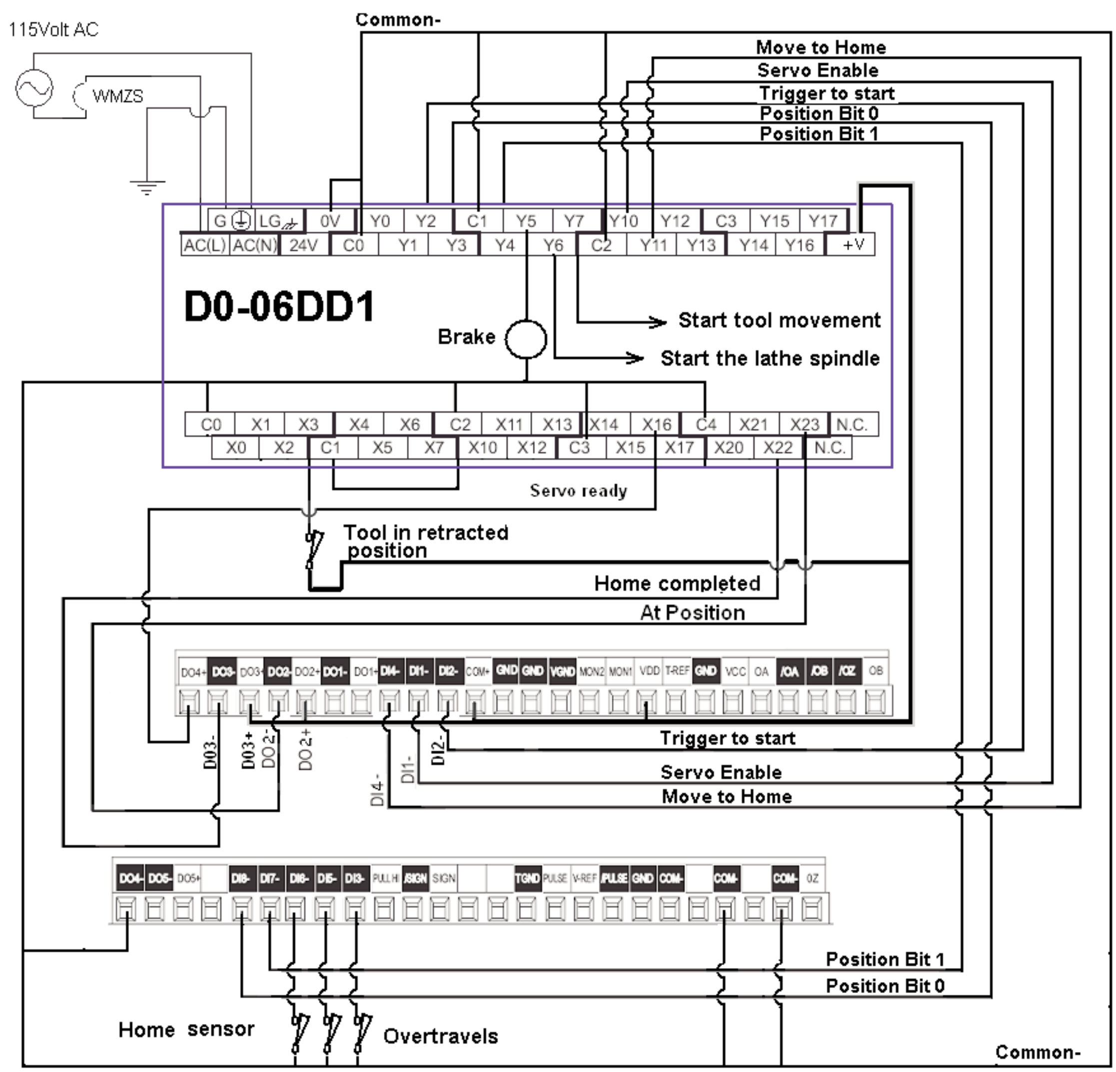
Y2 will have the command **Trigger to Start.**

Y3,and Y4 will have the signals to define the target position, as shown on the table:

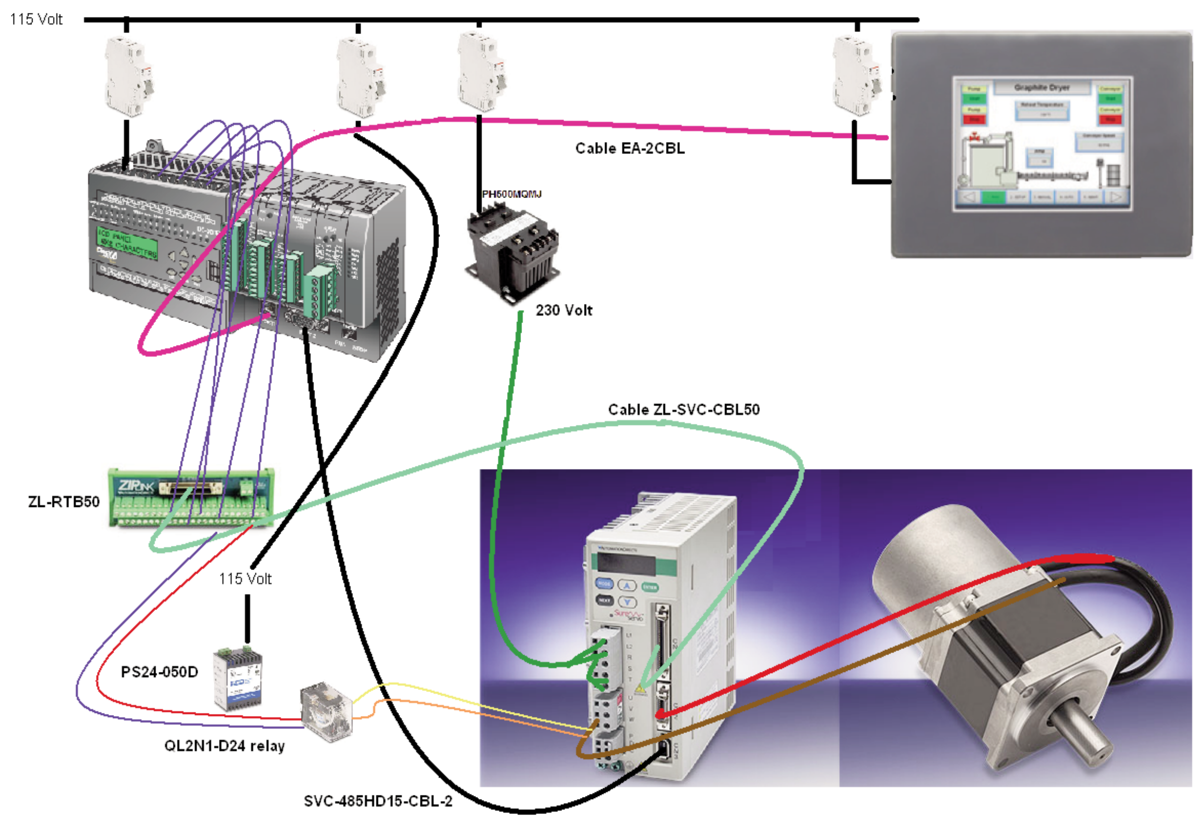
Position	POS0 Bit (Y3)	POS1 Bit (Y4)
0 rev	OFF	OFF
Target A	OFF	ON
Target B	ON	OFF
Target C	ON	ON

The **FWD overtravel and** the **REVERSE overtravel** limit switches are wired directly to the drive, as well as the home sensor. Y5 is the brake and the output Y6 is the command run to the lathe spindle. Y7 is the tool start (To begin the groove).

In the diagram below are shown the control connections necessary to make the system work as required.



The general wiring among all the pieces is shown on the diagram below:



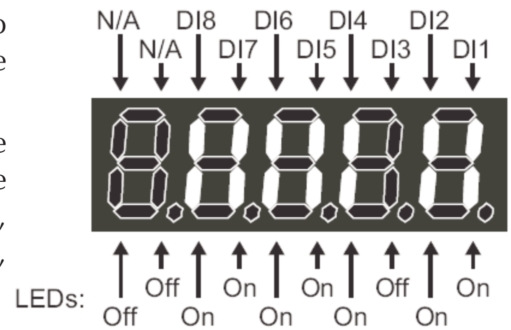
The hardware as implemented on the prototype for this application note consisted of the following materials; 230 Volt AC where obtained with a 500 VA control transformer:

- 1 supplementary protector WMZS2D10 (for general disconnect; be aware that you need a branch protection like fuses or circuit breaker)
- 1 supplementary protector WMZS2D2 (for disconnecting the PLC).
- 1 PLC D0-06DD1 fed with 115 Volt Volt (it could also be fed with 230 Volt).
- 1 C-more panel EA7-S6C.
- 1 servodrive SVA-2040.
- 1 servomotor 200 Watt SVL-202B.
- 1 power cable SVC-PFL-010.
- 1 encoder cable SVC-EFL-010.
- 1 breakout board ZL-RTB50
- 1 Zip link cable ZL-SVC-CBL50
- 1 communication cable SVC-485HD15-CBL-2.
- 1 cable EA-2CBL.
- the required cables for control and power.
- 1 power supply PS24-050D.
- 24 Volt coil relay QL2N1-D24 for brake control including socket

PLC- Servo drive wiring tests

The wiring between the C-more and the PLC is tested using Data View on the *Direct*SOFT program. The data has to be listed on the Data View window and when the PLC and touch panel are energized, the programmer has to check that the panel is writing data on the PLC and also by writing data on the PLC by means of the Data View Dialog box, the touch panel should display the same data on the corresponding screen.

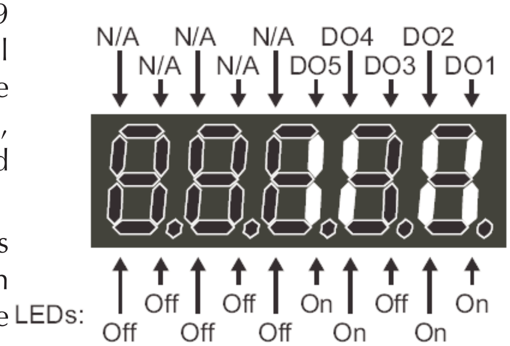
The wiring between the PLC and the servo can can be tested using the override function on the DL06, using *Direct*SOFT.



When the PLC outputs are wired to the servo, the parameter P4-07 allows to test the corresponding digital inputs. For that test, select the parameter P4-07 with the keypad, and then ENTER.

Every time that one digital input is activated, the corresponding LEDs will turn ON on the servo drive display, per the adjacent diagram.

On the same way, the parameter P4-09 shows the status of the servo drive digital outputs and then allows to test the corresponding digital inputs. For that action, select the parameter P4-09 with the keypad and then ENTER.

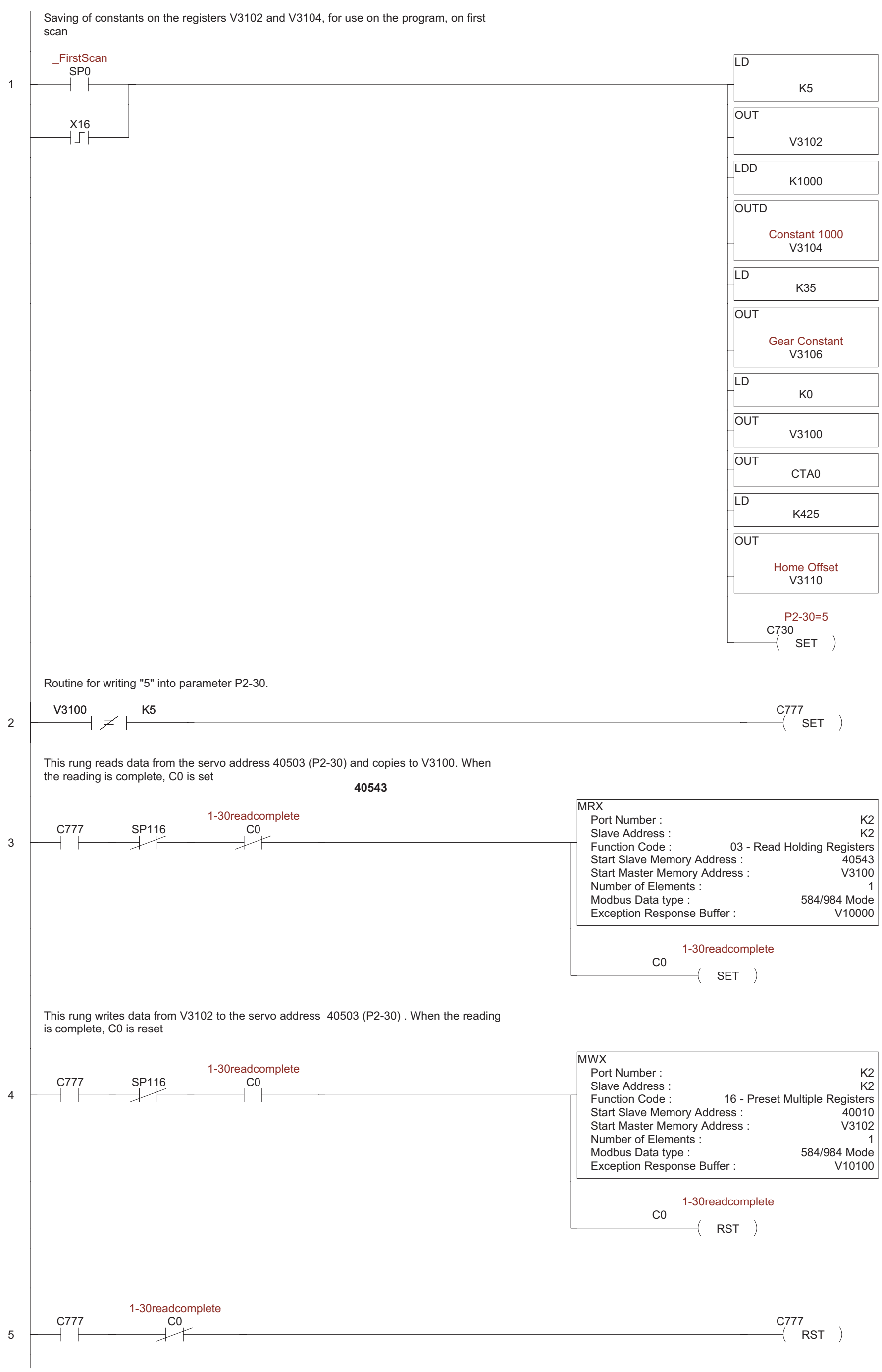


Every time that one digital output is activated, the corresponding LEDS will turn ON on the servo drive display, per the adjacent diagram.

At this time it would help to create a table with the I/0 assignments:

Servo	**PLC**	**Parameter**	**Value**	**Description**
DI1	Y10	P2-10	101	Servo enable
DI2	Y2	P2-11	108	Command trigger
DI3		P2-12	022	Reverse overtravel (directly connected)
DI4	Y11	P2-13	127	Search home trigger
DI5		P2-14	023	Forward overtravel (directly connected)
DI6		P2-15	124	Home sensor (directly connected))
DI7	Y4	P2-16	112	Position select bit 1
DI8	Y3	P2-17	111	Position select bit 0
D01	X22	P2-18	109	Home complete
D02	X23	P2-19	105	At position

The program of the PLC is shown below: explanations are given on the comments of the ladder diagram and the right side of each column.



This rung defines some constant to be used on the program

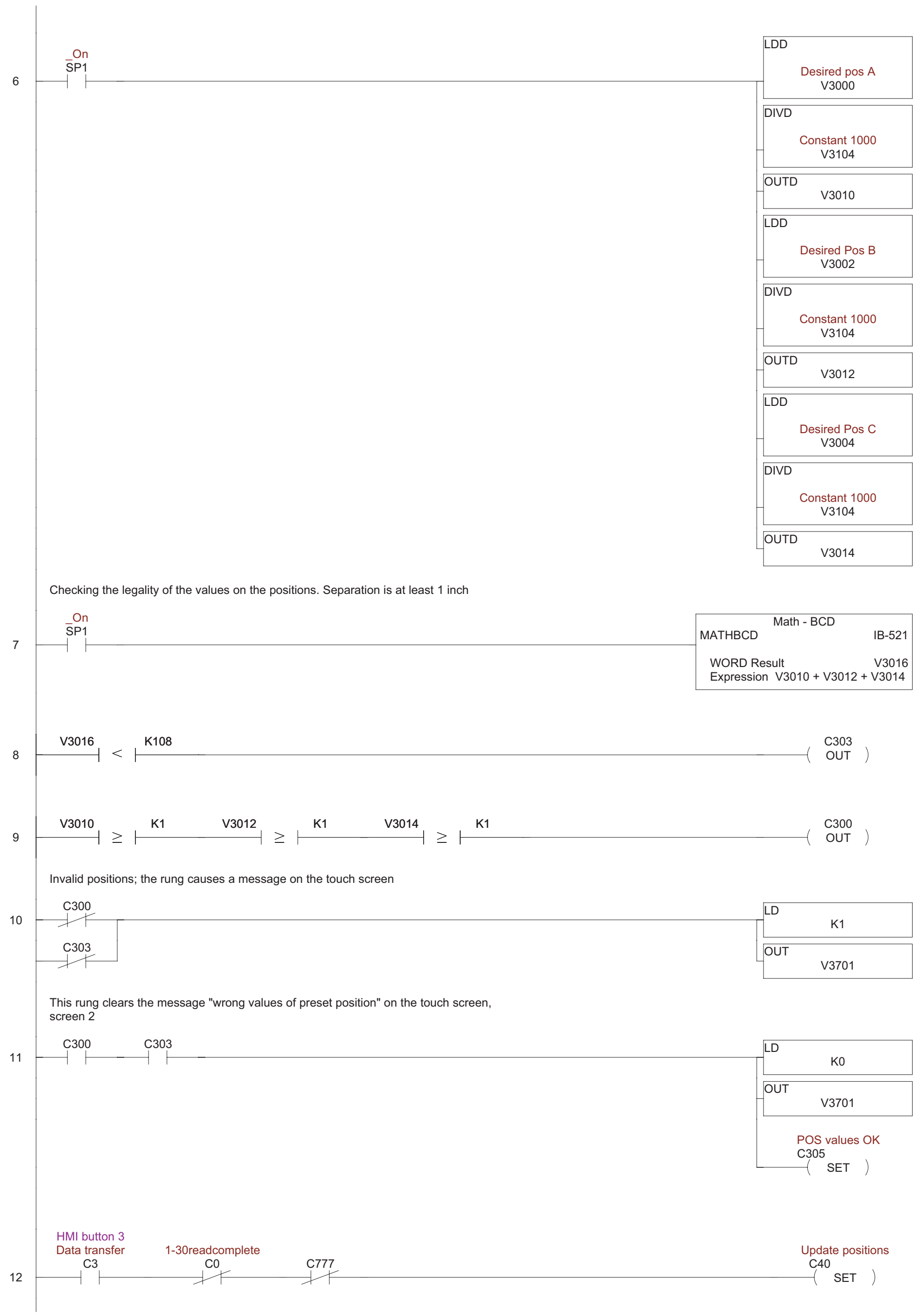
X16 is the servo ready signal.

Note that the value is V3100 is zeroed on one of the conditions fdirst scan SPO or leading edge of X16 to provoke a compare instruction on rung 2, that will be false on the first scan. This will set the coil C777 for further actions

When C777 is active, the read instruction MRX reads the value of V3100, which should be zero at the first scan; this action, when completed, set the bit C0, which activates the write instruction MWX on the rung 4 to copy the value of V3102 to the parameter P2-30, a “5”, to allow continuous writing without damaging the EEPROM on the servo drive.

C777 will stay active as long as the read and write transaction complete the cycle; that is, when C0 resets the rung 4 completes the writing transaction.

C777 is reset when C0 is reset



The value of V3000 and V3001 is the desired position A. We divide by 1000 to have a unit of 1/10th inch, only for comparison and validation that the desired positions are in the accepted range.

The value of V3002 and V3003 is the desired position B. We divide by 1000 to have a unit of 1/10th inch, only for comparison and validation that the desired positions are in the accepted range

The value of V3004 and V3005 is the desired position C. We divide by 1000 to have a unit of 1/10th inch, only for comparison and validation that the desired positions are in the accepted range

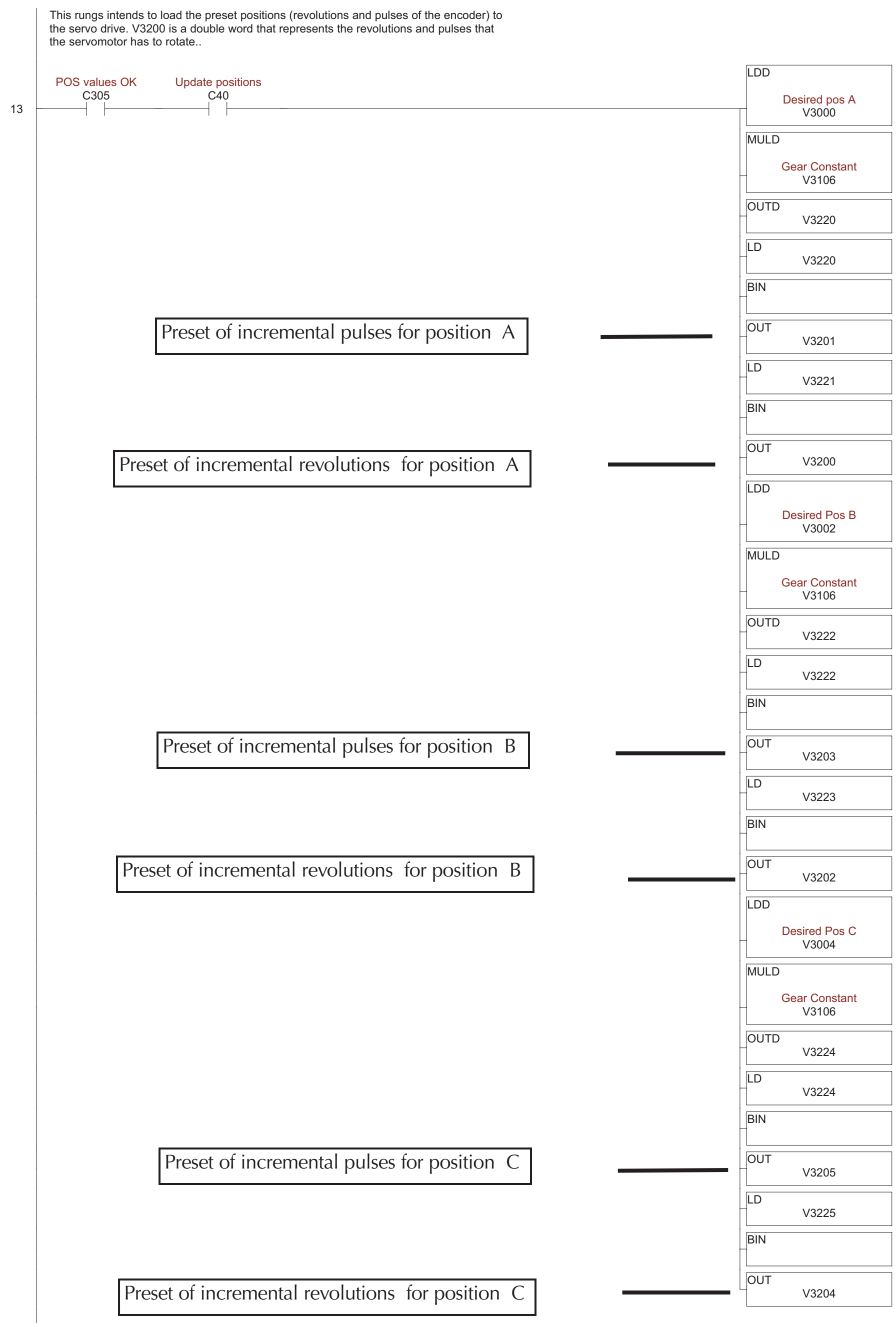
This rung adds the desired positions for check that the addition is not above the proper range (less than 108 inches)

This rung sets the flag C300 to indicate the desired values are separated for more than 1 inch

This rung sets a 1 in V3701 to generate on the screen 2 a message on the multistate text indicator of wrong desired position data.

This rung just resets V3701 to zero. to clear the text on the multistate text indicator message on the screen 2 Also, C305 is a flag to indicate accepted values

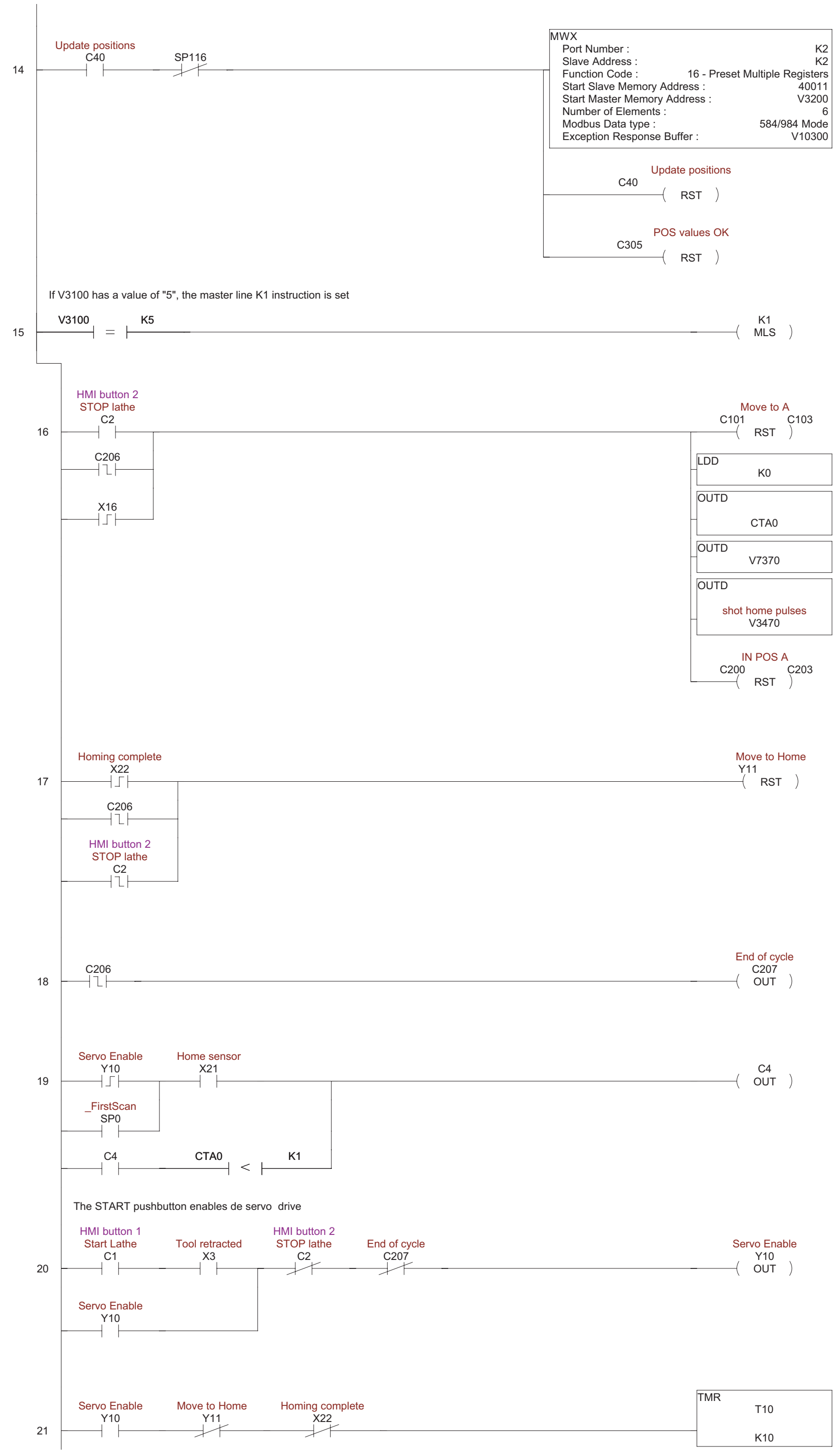
This rung sets C40, to allow a communication transaction to write the desired values into the servo preset position registers.



The rung turns true when the bit C40 is set, to transfer the desired positions to the servo drive. Of course, the values should be valid, and the condition is such that that C305 is set.

The concept here is that, since the data what entered in BCD, the most significant word is really the revolutions and the less significant the pulses.
These rungs process these data and also uses the gear constant (which is 35 in this case) to transform inches into revolutions and pulses.
The position A is multiplied in BCD by the gear constant and the result loaded into V3220 as double word.
The word register V3321 in BCD is formatted as decimal in V3200, and correspond to the incremental revolutions to be displaced by the servomotor.
The word register V3320 in BCD is formatted as decimal in V3201, and correspond to the incremental pulses to be displaced by the servomotor.

Same concept is valid for positions B and C



This rung just writes, in one operation, the desired positions A, B and C, turning them into the preset positions saved in the RAM memory of the servo drive. C40 is the bit to interlock the operation with the continuous reading of the servo current position, speed and torque.

This rung turns ON the master line K1 only if there is a 5 in the memory P2-230 of the servo.

When the STOP button on the C-more panel is touched , the flags of the movements to A, B and C are reset, as well as the counter CT0.

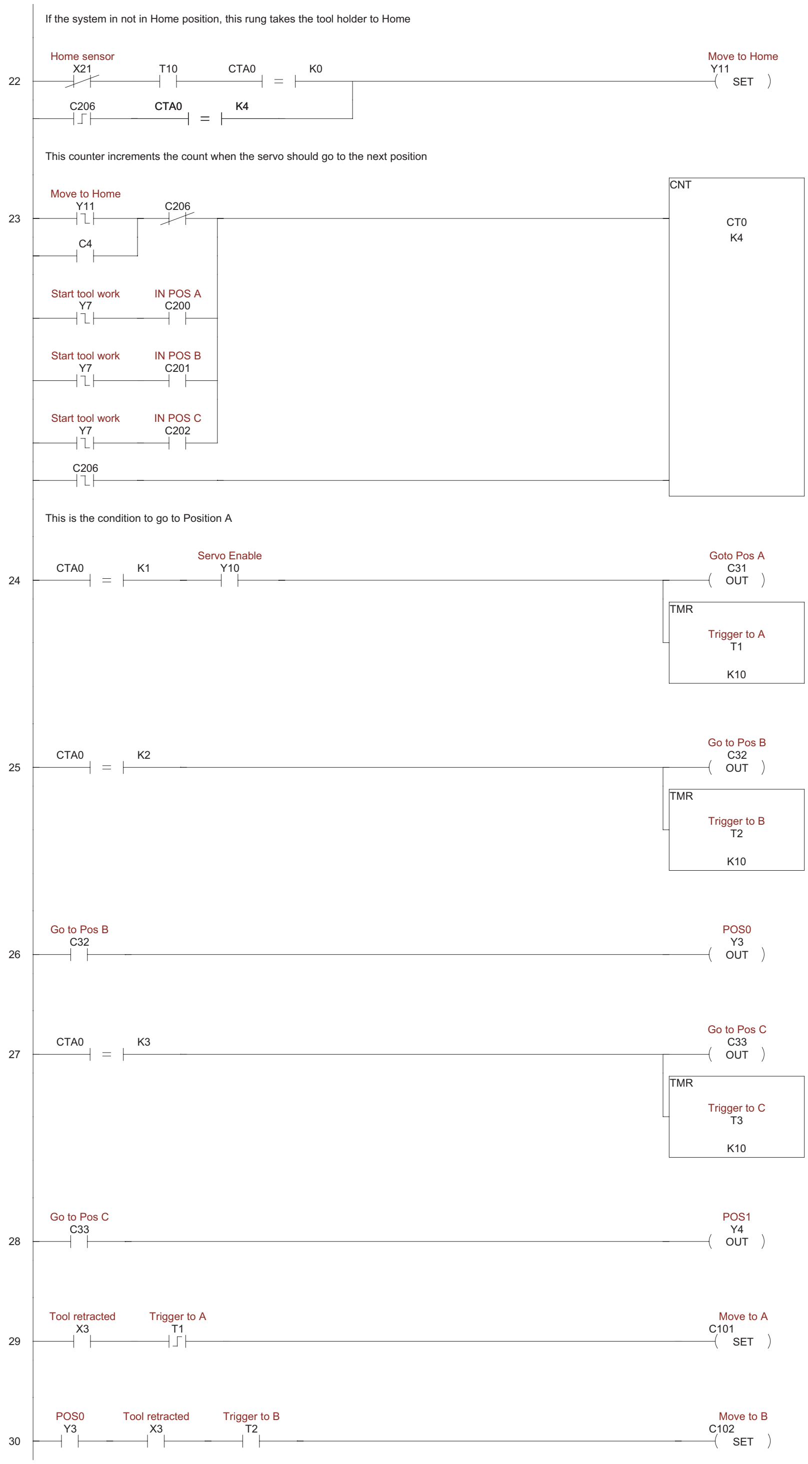
The same effect is done at the end of the cycle, with C206 and X16

This rung resets the output Y11, which is the output to command the servo to go to Home position.

C4 is a bit that is ON when it is detected that the tool holder is in Home position at the moment the servo is energized. Otherwise C4 is NOT set.

The START command enables the servo. The tool holder should be in retracted position.

The code here intends to create a delay on the homing trigger command.



Here the output Y11 is activated to produce the trigger for home search.

This counter is the main instruction to control the sequencing of steps to move to position, A, then B and then C.

Note that the counter increments a count when the servo is in position and the tool holder system has completed a cycle (Y7 is turning OFF).

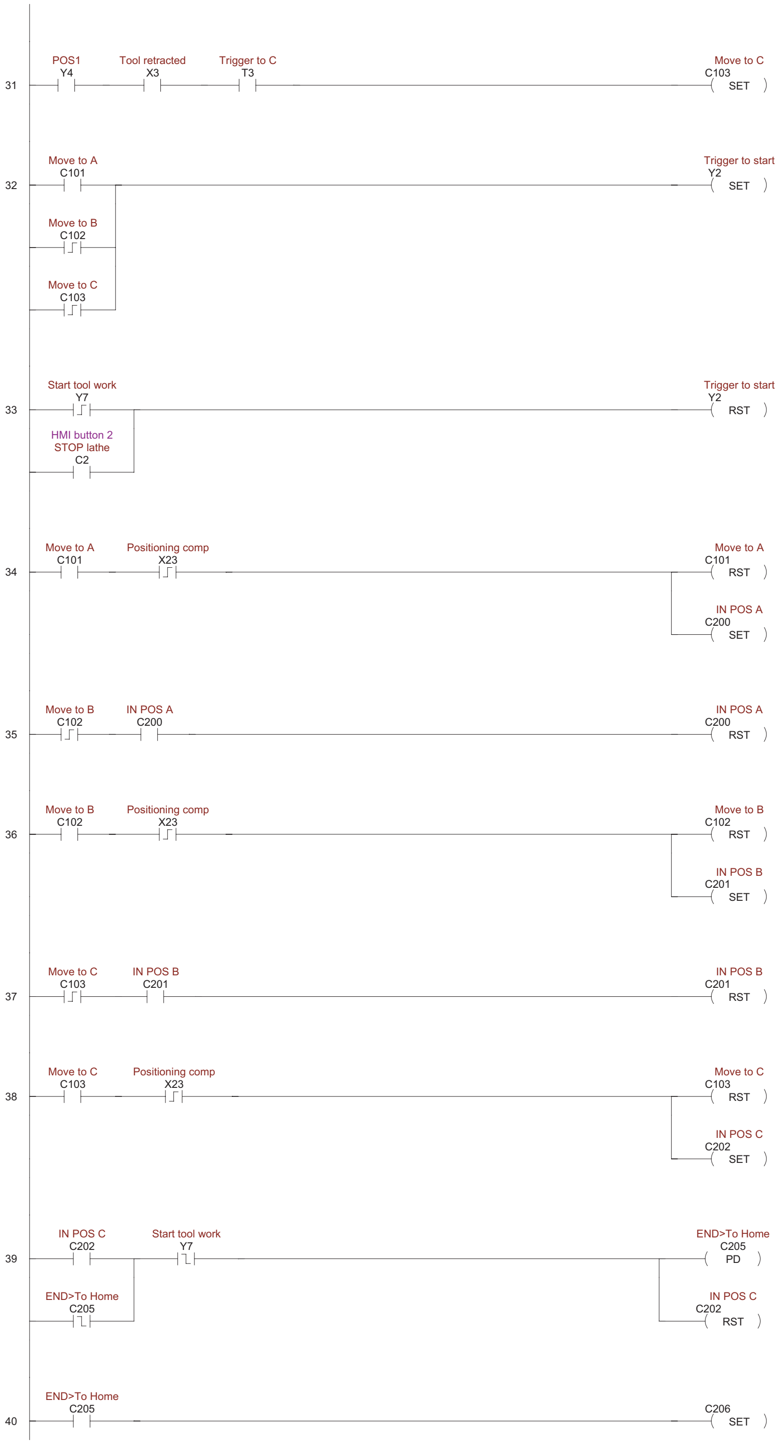
When the counter CT0 is in 1, the PLC is preparing to generate a command on the servo to go to position A.
Y3 and Y4 are OFF.
The timer T1 counts 1 second to generate the trigger to move to A.

When the counter CT0 is in 2, the PLC is preparing to generate a to command the servo to go to position B.
Y3 should be turned ON and Y4 is OFF.
The timer T2 counts 1 second to generate the trigger to move to B.

When the counter CT0 is in 3, the PLC is preparing to generate a command to the servo to go to position C.
Y3 is turned OFF and Y4 should turn ON.
The timer T3 counts 1 second to generate the trigger to move to C.

The timer T1 generates the trigger to move to A.

The timer T2 generates the trigger to move to B.



The timer T3 generates the trigger to move to C.

The leading edge of the pulse in C101, C102 or C103 activates the trigger signal Y2 to move to the proper position.

The trigger signal Y2 is turned OFF when the tool holder begins a movement
The same happens when the object button STOP on the screen 12 is touched.

The flag C101, Move to A, is reset when the servo confirms that the position has been completed, as well the bit C200 is set to indicate that the tool holder is in position A.

The bit C200, In position A, is reset when there is other command to move to B, exactly on the leading edge of the signal C102.

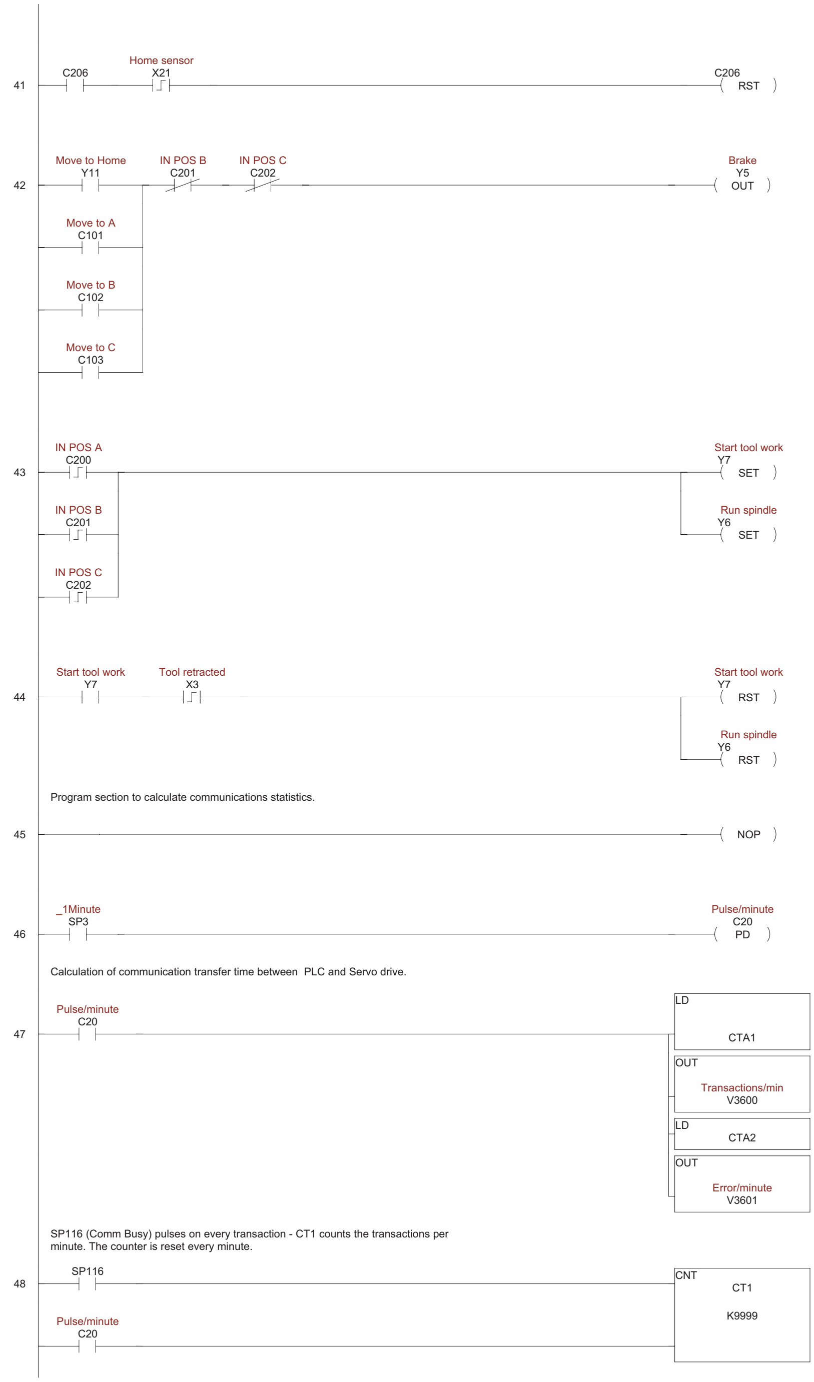
The flag C102, Move to B, is reset when the servo confirms that the position has been completed, as well the bit C201 is set to indicate that the tool holder is in position B.

The bit C201, In position B, is reset when there is other command to move to C, exactly on the leading edge of the signal C103.

The flag C103, Move to C, is reset when the servo confirms that the position has been completed, as well the bit C202 is set to indicate that the tool holder is in position C.

The bit C202, In position C, is reset when there the tool holder retracts and the start tool work signal is turning from ON to OFF, as well as the bit C205 is On on a One shot pulse to flag with C206 that the tool holder should return to Home.

C206 is set with a “One shot” pulse of C205.



C206 is reset when the tool holder reaches the Home position.

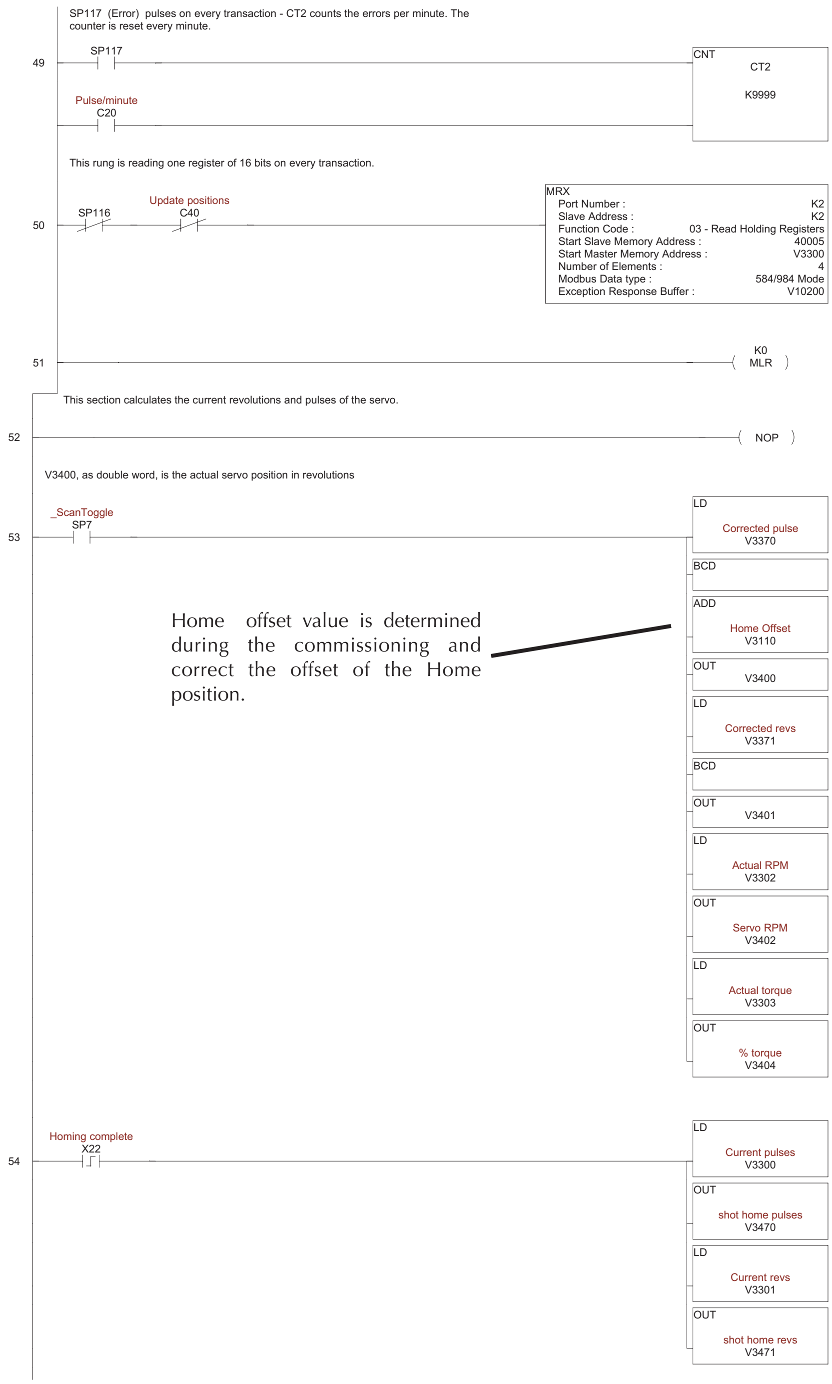
Brake is released with Y5 turned ON Note that the servo brake has to be released when:
- The servo moves to A
- The servo moves to B
- The servo moves to C
- The servo moves to Home.

The tool on the tool holder penetrates the material on the spindle when the command Y7 is activated and at the same time the spindle runs.
The spindle runs when Y6 is ON.

When the tool holder is retracted, Y7 is reset, as well as the spindle signal Y6 for the spindle motor.

This rung generates a pulse of 1 scan duration every minute. It would be also possible to use 1 second pulse with SP4.

Every minute we save the content of the counter CT1 and CT2, before resetting them. This defines the transactions and errors occurring on every minute.
We decided to show only the transactions per second on the C-more panel, where the value in the memory V3600 is divided by 60 to have a rate per second.
V3601 is not shown on the C-more panel, but it could be a good troubleshooting tool with Data View, to control the errors on the communication



Home offset value is determined during the commissioning and correct the offset of the Home position.

This rung continuously reads servo status when the master relay K1 is active, except when C40 is ON.
See more explanations below the rung 54.

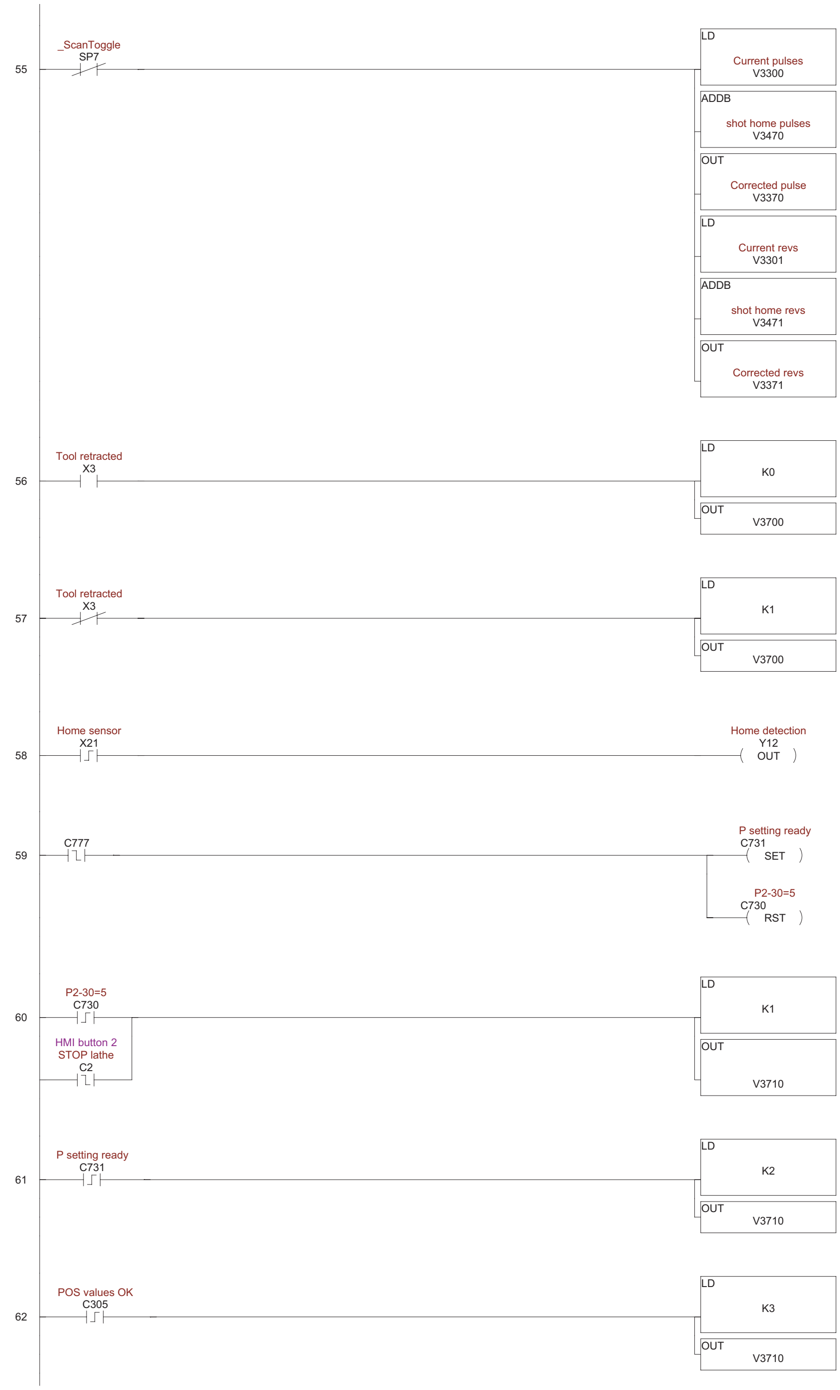
This rung works together with the rungs 54 and 55.
The corrected pulses V memory is the value of pulses that are between he home position and the current position and they are translated to BCD to make an 8 digit number on the double word V3400, together with the corrected revolutions.

The actual RPMs is the value read from the servo and displayed on the C-more display

The actual torque is the value in % read from the servo and displayed on the C-more display

Every time the Home position is reached, it is taken a “snapshot” of the servo position value.
These values will be used to correct the offset that may exist if the servo is energized out of Home position.
The values in V3470 and V3471 are always negative values.

Explanation of the data exchange on rung 50:
V3300 gets the content of V4005 on the servo drive; which is parameter P0-04, configured as 0=> current pulses
V3301 gets the content of V4006 on the servo drive; which is parameter P0-05, configured as 0=> current revolutions
V3302 gets the content of V4007 on the servo drive; which is parameter P0-06, configured as 0=> current speed in rpm
V3303 gets the content of V4008 on the servo drive; which is parameter P0-07, configured as 0=> current % torque.



The current values of servo position are corrected to reflect only the displacement from the Home position.
We use an ADD instruction since the "shot home" value is negative

This rung is used to load a 0 into V3700, to indicate the tool position (retracted or in movement) with a multistate text indicator on screen 1.
- "0" indicates retracted position
- "1" indicates "tool in movement".

These following rungs defines a number into V3710 to show the corresponding message number of the multistate text indicator on the C-more panel.
There are 16 messages.
Some of them cannot be seen because the action is too fast.

The message here is
1 Ready to start; checking P2-30

The message here is
2 Ready to set positions

The message here is
3 Let us to set positions

The message here is
4 New positions defined

The message here is
5 Moving to Home

The message here is:
6 Defining position A

The message here is:
7 Moving to position A

The message here is:
8 IN POSITION A

The message here is:
9 Defining position B

The message here is:
10 Moving to position B

The message here is:
11 IN POSITION B

The message here is:
12 Defining position C

The message here is:
13 Moving to position C

The message here is:
14 IN POSITION C

The message here is:
15 END=......Goto Home

The message here is:
16 Home and STOP

We do not claim that this is the best program for the application. The program only work for this specific example and there are alternatively many other codes to execute the same functions.

The program has a scan time of 7 ms on the PLC DL06. This program uses 395 words on the ladder program.

As described before, the baud rate selected was 38.4 kbps, and with this speed the normal transitions per second are on the order of 38, that is, typically the update of the data reading will take about 26 milliseconds.

The DL06 was communicating with *Direct*SOFT thru an Ethernet module H0-ECOM100, to access the PLC while the port 1 is used for the C-more panel; and the port 2 to communicate to the servo drive. The C-more panel communicates also with Ethernet. In this way, the program can be developed and monitored without unplugging the serial cables connected to the PLC.

The home sensor used on the prototype is a photocell. Proximity sensors can be used alternatively.

We have attached compressed files (file AN-SERV-004.zip) to this pdf format file, to allow the client to install the project on his system (PLC, Servo and C-more panel project files)).

Automation Direct SureServo PRO Drives Configuration Report
Config Name: App note 4 servo.ssc
Motor Code: 11
Rev: 2.001
Notice that with version 2.105 we can do digital inputs though MODBUS.. See application note AN-SERV-007.

Parameter	Value	
P0.00 - Software Version	2001	
P0.01 - Drive Fault Code	0	
P0.02 - Drive Status (Front panel display)	3	
P0.03 - Analog Monitor Outputs	1	
P0.04 - Status Monitor 1	0	
P0.05 - Status Monitor 2	1	
P0.06 - Status Monitor 3	6	
P0.07 - Status Monitor 4	11	
P0.08 - Status Monitor 5	0	
P0.09 - Block transfer parameter 1	*21E*	*Entered by keypad (P2-30)*
P0.10 - Block transfer parameter 2	*10F*	*Entered by keypad (P1-15)*
P0.11 - Block transfer parameter 3	*110*	*Entered by keypad (P1-16)*
P0.12 - Block transfer parameter 4	*111*	*Entered by keypad (P1-17)*
P0.13 - Block transfer parameter 5	*112*	*Entered by keypad (P1-18)*
P0.14 - Block transfer parameter 6	*113*	*Entered by keypad (P1-19)*
P0.15 - Block transfer parameter 7	*114*	*Entered by keypad (P1-20)*
P1.00 - External Pulse Input Type	2	
P1.01 - Control Mode and Output Direction	101	Position with internal registers
P1.02 - Speed and Torque Limit	1	
P1.03 - Output Polarity Setting	0	
P1.04 - Analog Monitor Output Scaling 1 (CH1)	100	
P1.05 - Analog Monitor Output Scaling 2 (CH2)	100	
P1.06 - Analog Velocity Command Low-pass Filter	0	
P1.07 - Analog Torque Command Low-pass Filter	0	
P1.08 - Position Command Low-pass Filter	0	
P1.09 - Preset Velocity Command / Limit 1	100	
P1.10 - Preset Velocity Command / Limit 2	200	
P1.11 - Preset Velocity Command / Limit 3	300	
P1.12 - Preset Torque Command / Limit 1	100	
P1.13 - Preset Torque Command / Limit 2	100	
P1.14 - Preset Torque Command / Limit 3	100	
P1.15 - Position 1 Command (Revolutions)	35	Variable through MODBUS
P1.16 - Position 1 Command (Counts)	5000	Variable through MODBUS
P1.17 - Position 2 Command (Revolutions)	70	Variable through MODBUS
P1.18 - Position 2 Command (Counts)	2500	Variable through MODBUS
P1.19 - Position 3 Command (Revolutions)	350	Variable through MODBUS
P1.20 - Position 3 Command (Counts)	1000	Variable through MODBUS
P1.21 - Position 4 Command (Revolutions)	0	
P1.22 - Position 4 Command (Counts)	0	

P1.23 - Position 5 Command (Revolutions)	0	
P1.24 - Position 5 Command (Counts)	0	
P1.25 - Position 6 Command (Revolutions)	0	
P1.26 - Position 6 Command (Counts)	0	
P1.27 - Position 7 Command (Revolutions)	0	
P1.28 - Position 7 Command (Counts)	0	
P1.29 - Position 8 Command (Revolutions)	0	
P1.30 - Position 8 Command (Counts)	0	
P1.31 - Motor Code	11	200 watt for this prototype
P1.32 - Motor Stop Mode Selection	0	
P1.33 - Position Control Mode (Internal Indexer)	1	
P1.34 - Acceleration Time (Internal Indexer)	1000	
P1.35 - Deceleration Time (Internal Indexer)	1000	
P1.36 - Accel / Decel S-Curve	20	
P1.37 - Inertia Mismatch Ratio	6	
P1.38 - Zero Speed Output Threshold	10	
P1.39 - Target Speed Output Threshold	3000	
P1.40 - Max Analog Velocity Cmd or Velocity Limit	3000	
P1.41 - Max Analog Torque Cmd or Torque Limit	100	
P1.42 - On Delay Time of Electromagnetic Brake	20	
P1.43 - Off Delay Time of Electromagnetic Brake	20	
P1.44 - Electronic Gear Numerator 1	1	
P1.45 - Electronic Gear Denominator	1	
P1.46 - Encoder Output Scaling Factor	1	
P1.47 - Homing Mode	1223	
P1.48 - Homing Speed 1 Fast Search Speed	60	
P1.49 - Homing Speed 2 Creep Speed	12	
P1.50 - Home Position Offset (Revolutions)	0	
P1.51 - Home Position Offset (Counts)	0	
P1.52 - Regenerative Resistor Value	40	
P1.53 - Regenerative Resistor Capacity	60	
P1.54 - In Position Window	100	
P1.55 - Maximum Speed Limit	3500	

P2.00 - Proportional Position Loop Gain (KPP)	35	
P2.01 - Position Loop Gain Boost	100	
P2.02 - Position Feed Forward Gain (KFF)	5000	
P2.03 - Smoothing Constant of Position Feed Forward Gain	5	
P2.04 - Velocity Loop Proportional Gain (KVP)	500	
P2.05 - Velocity Loop Gain Boost	100	
P2.06 - Velocity Loop Integral Compensation (KVI)	100	
P2.07 - Velocity Feed Forward Gain (KVF)	0	
P2.08 - Factory Defaults and Security	0	
P2.09 - Bounce Filter	2	
P2.10 - Digital Input Terminal 1 (DI1)	101	Servo enable
P2.11 - Digital Input Terminal 2 (DI2)	108	Command trigger
P2.12 - Digital Input Terminal 3 (DI3)	122	Reverse inhibit

P2.13 - Digital Input Terminal 4 (DI4)	127	Home trigger
P2.14 - Digital Input Terminal 5 (DI5)	123	Forward Inhibit
P2.15 - Digital Input Terminal 6 (DI6)	124	Home sensor
P2.16 - Digital Input Terminal 7 (DI7)	112	Position 1 select
P2.17 - Digital Input Terminal 8 (DI8)	111	Position 0 select
P2.18 - Digital Output Terminal 1 (DO1)	0	
P2.19 - Digital Output Terminal 2 (DO2)	105	At position
P2.20 - Digital Output Terminal 3 (DO3)	109	Homing completed
P2.21 - Digital Output Terminal 4 (DO4)	0	Servo ready
P2.22 - Digital Output Terminal 5 (DO5)	0	
P2.23 - Notch Filter (Resonance Suppression)	1000	
P2.24 - Notch Filter Attenuation	0	
P2.25 - Low-pass Filter (Resonance Suppression)	2	
P2.26 - External Anti-Interference Gain	0	
P2.27 - Gain Boost Control	0	
P2.28 - Gain Boost Switching Time	10	
P2.29 - Gain Boost Switching Condition	10000	
P2.30 - Auxiliary Function	0	
P2.31 - Auto and Easy Tuning Mode Response Level	68	
P2.32 - Tuning Mode	4	
P2.33 - Reserved	0	
P2.34 - Overspeed Fault Threshold	5000	
P2.35 - Position Deviation Fault Window	30000	
P2.36 - Position 1 Velocity	3500	
P2.37 - Position 2 Velocity	240	possible to go to 3500 rpm
P2.38 - Position 3 Velocity	240	possible to go to 3500 rpm
P2.39 - Position 4 Velocity	240	possible to go to 3500 rpm
P2.40 - Position 5 Velocity	1000	
P2.41 - Position 6 Velocity	1000	
P2.42 - Position 7 Velocity	1000	
P2.43 - Position 8 Velocity	1000	
P2.44 - Digital Output Mode	0	
P2.45 - Index Mode Output Signal Delay Time	1	
P2.46 - Index Mode Stations	6	
P2.47 - Position Deviation Clear Delay Time	0	
P2.48 - Backlash Compensation (Index Mode)	0	
P2.49 - Jitter Suppression	0	
P2.50 - Clear Position Mode	0	
P2.51 - Servo Enable Command	0	
P2.52 - Dwell Time 1 - Auto Index Mode	0	
P2.53 - Dwell Time 2 - Auto Index Mode	0	
P2.54 - Dwell Time 3 - Auto Index Mode	0	
P2.55 - Dwell Time 4 - Auto Index Mode	0	
P2.56 - Dwell Time 5 - Auto Index Mode	0	
P2.57 - Dwell Time 6 - Auto Index Mode	0	
P2.58 - Dwell Time 7 - Auto Index Mode	0	
P2.59 - Dwell Time 8 - Auto Index Mode	0	

P2.60 - Electronic Gear Numerator 2	1	
P2.61 - Electronic Gear Numerator 3	1	
P2.62 - Electronic Gear Numerator 4	1	
P2.63 - Velocity and Position Deviation Scaling Factor	0	
P3.00 - Communication Address	2	Slave 2
P3.01 - Transmission Speed	2	19.2 kbps
P3.02 - Communication Protocol	8	MODBUS RTU
P3.03 - Communication Fault Action	0	
P3.04 - Communication Watchdog Time Out	0	
P3.05 - Communication Selection	2	
P3.06 - Reserved	0	
P3.07 - Communication Response Delay Time	0	
P4.00 - Fault Record - Most recent (N)	9	
P4.01 - Fault Record (N-1)	9	
P4.02 - Fault Record (N-2)	9	
P4.03 - Fault Record (N-3)	9	
P4.04 - Fault Record (N-4)	9	
P4.05 - JOG Function	2000	
P4.06 - Force Outputs Command	0	
P4.07 - Input Status	0	
P4.08 - Reserved	0	
P4.09 - Output Status	2	
P4.22 - Analog Velocity Input Offset	0	
P4.23 - Analog Torque Input Offset	0	