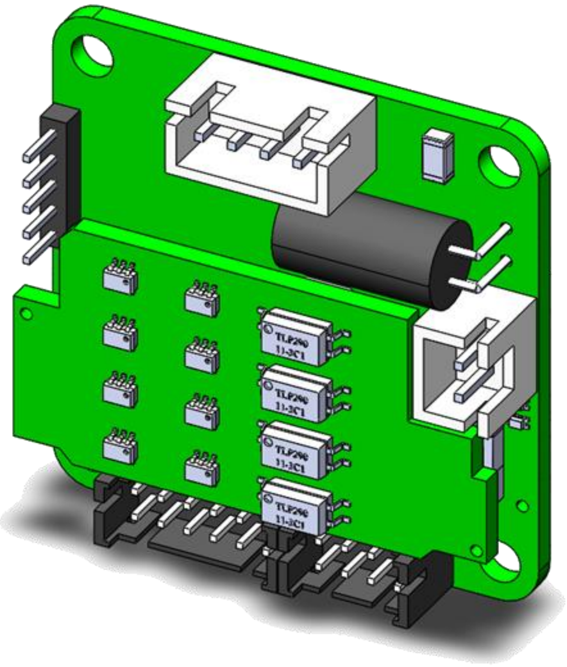
DOC25010601712

Firmware Simple tuner user manual

Order command : EZD-1136-CANOPEN



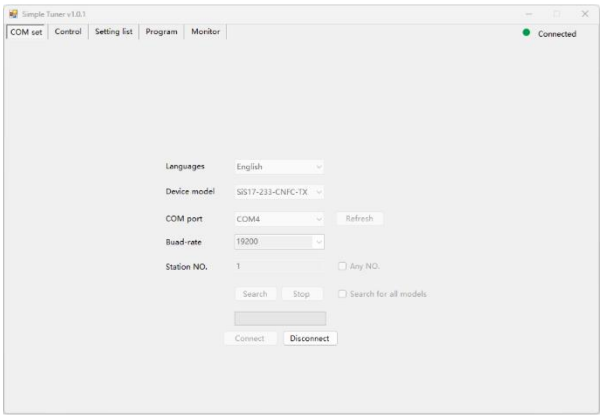
Revisions

Version	Modifications	Writer	Checker	Date
1.2	Initial version	MLE	SRU	12/11/2024
1.3	Update interface description	MLE	SRU	22/11/2024
1.4	Add Programming section	MLE	SRU	09/12/2024
1.5	Complete programming functions	MLE	SRU	17/12/2024
1.6	Add parameters list	MLE	SRU	06/01/2025

Table of Contents

1. Global View

The simple tuner software have 5 different window :



Languages: English

Device model: SiS17-233-CNFC-TX

COM port: COM4

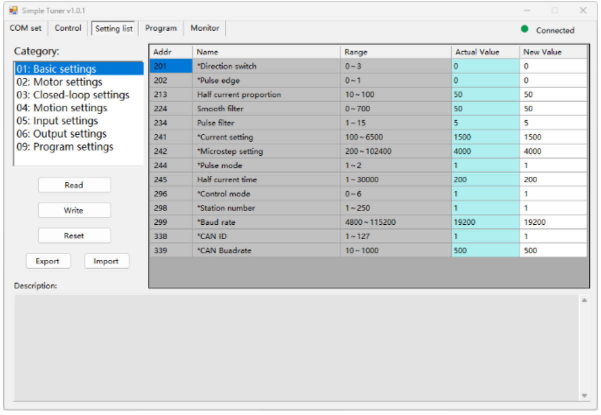
Buad-rate: 19200

Station NO.: 1

□ Any NO.

□ Search for all models

Figure 1 : Communication window

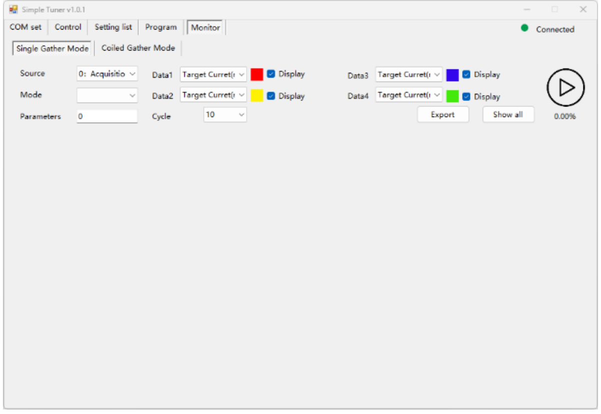


Category:

- 01: Basic settings
- 02: Motor settings
- 03: Closed-loop settings
- 04: Motion settings
- 05: Input settings
- 06: Output settings
- 09: Program settings

Addr	Name	Range	Actual Value	New Value
201	*Direction switch	0~3	0	0
202	*Pulse edge	0~1	0	0
213	Half current proportion	10~100	50	50
224	Smooth filter	0~700	50	50
234	Pulse filter	1~15	5	5
241	*Current setting	100~6500	1500	1500
242	*Microstep setting	200~102400	4000	4000
244	*Pulse mode	1~2	1	1
245	Half current time	1~30000	200	200
296	*Control mode	0~6	1	1
298	*Station number	1~250	1	1
299	*Baud rate	4800~115200	19200	19200
338	*CAN ID	1~127	1	1
339	*CAN Buadrate	10~1000	500	500

Figure 2 : Parameter list window



Source: 0: Acquisitio

Mode:

Parameters: 0

Data1: Target Curret(ı 🟥 ☑ Display

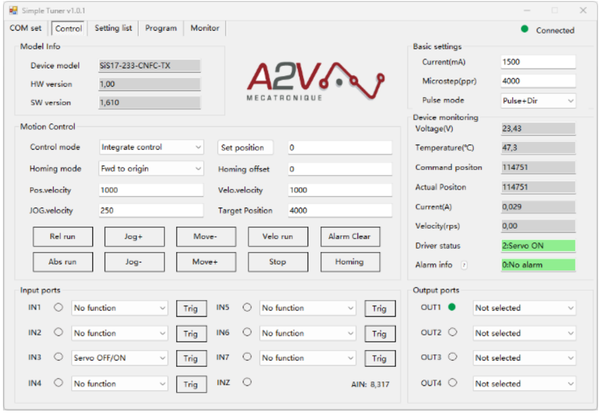
Data2: Target Curret(ı 🟨 ☑ Display

Cycle: 10

Data3: Target Curret(ı 🟦 ☑ Display

Data4: Target Curret(ı 🟩 ☑ Display

Figure 3 : Monitor window



Model Info

Device model: SiS17-233-CNFC-TX

HW version: 1,00

SW version: 1,610

Motion Control

Control mode: Integrate control

Set position: 0

Homing mode: Fwd to origin

Homing offset: 0

Pos.velocity: 1000

Velo.velocity: 1000

JOG.velocity: 250

Target Position: 4000

Basic settings

Current(mA): 1500

Microstep(ppr): 4000

Pulse mode: Pulse+Dir

Device monitoring

Voltage(V): 23,43

Temperature(℃): 47,3

Command positon: 114751

Actual Positon: 114751

Current(A): 0,029

Velocity(rps): 0,00

Driver status: 2:Servo ON

Alarm info: 0:No alarm

Input ports

IN1 ○ No function

IN2 ○ No function

IN3 ○ Servo OFF/ON

IN4 ○ No function

IN5 ○ No function

IN6 ○ No function

IN7 ○ No function

INZ ○

AIN: 8,317

Output ports

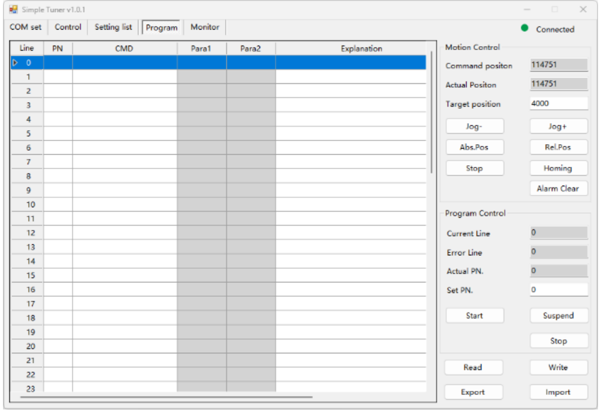
OUT1 ● Not selected

OUT2 ○ Not selected

OUT3 ○ Not selected

OUT4 ○ Not selected

Figure 4 : Control window



Motion Control

Command positon: 114751

Actual Positon: 114751

Target position: 4000

Program Control

Current Line: 0

Error Line: 0

Actual PN.: 0

Set PN.: 0

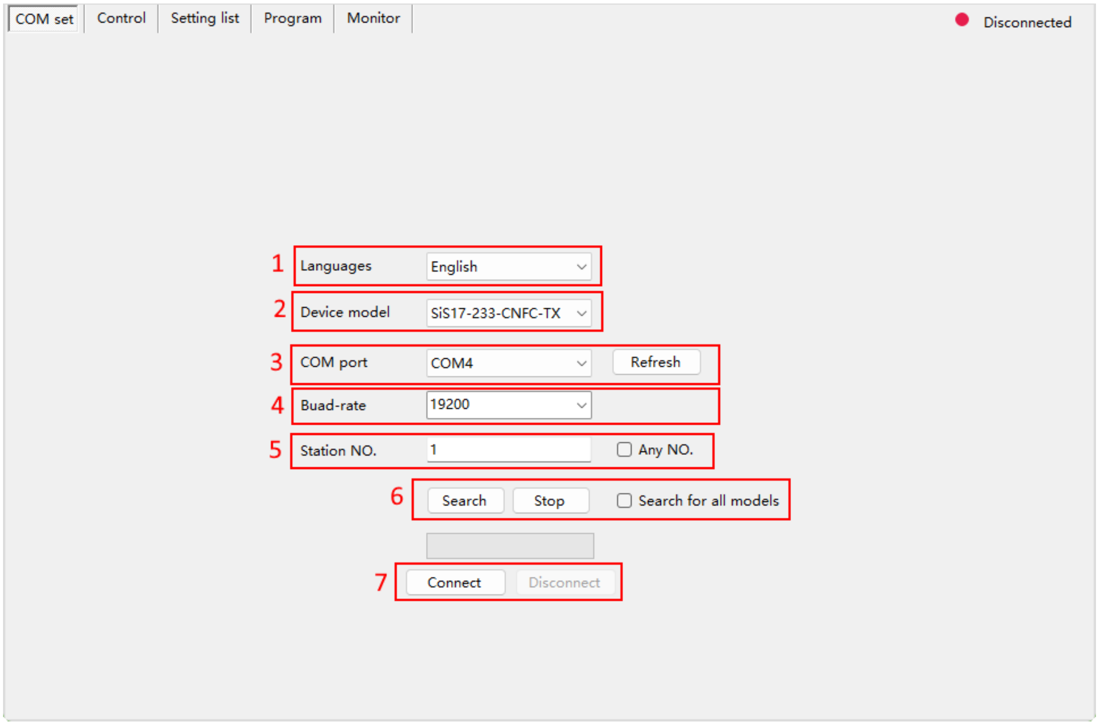
Figure 5 : Program window

2. Software interface

• Initialize communication

Select the device model, the device model needs to be consistent with the connected driver model, and then select the serial port, that is, the device access computer out of the serial port, if you first open the software in the access serial module, you need to press the refresh key, re-read the serial port, and then select the correct serial port. Then select the serial Baud Rate, the default Baud Rate is 19200, just select. Finally, the site number, generally if no modification default is 1, confirm it.

Once you've confirmed it's all done, tap Connect, and the software jumps to the device monitoring interface



Languages English

Device model SiS17-233-CNFC-TX

COM port COM4

Buad-rate 19200

Station NO. 1 □ Any NO.

□ Search for all models

Figure 6 : Communication interface

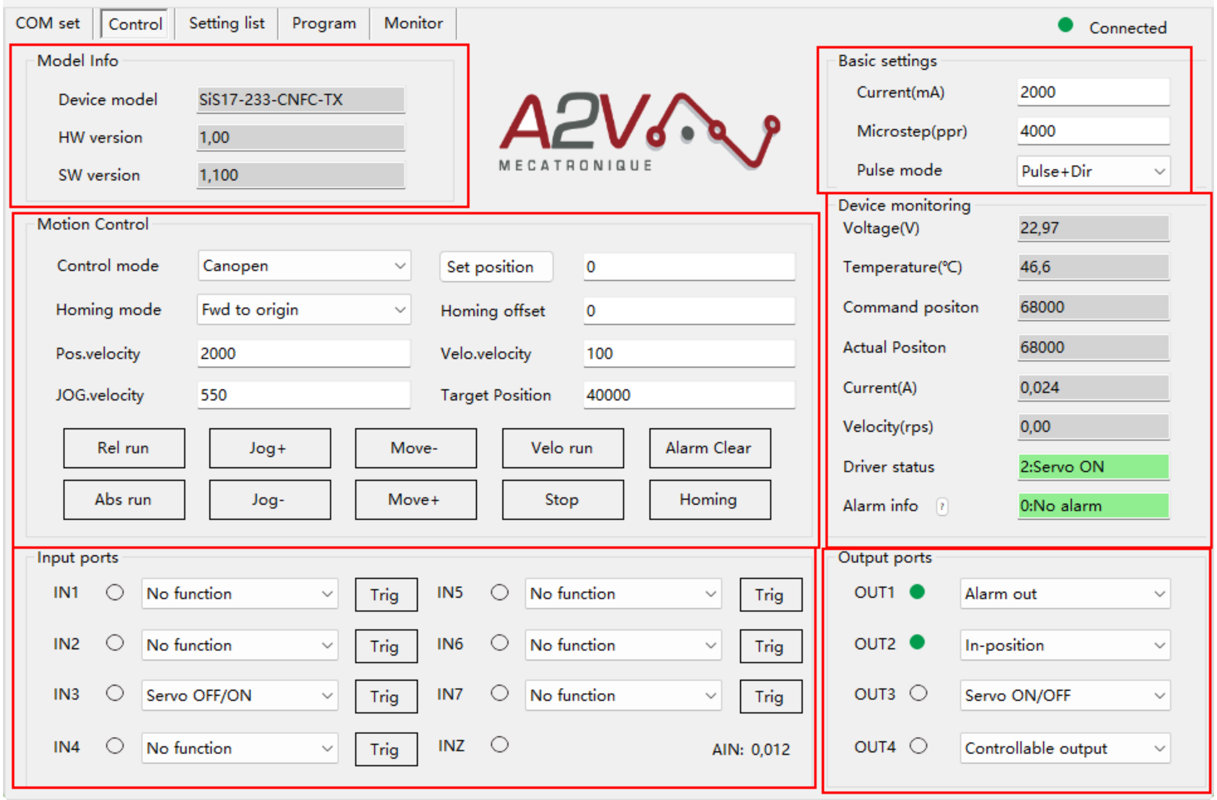
Id	Function
1	Select the language
2	Select the device model, which is the connected driver model
3	Select the serial port and the drive needs to be installed when connecting to the motor using the module
4	Select serial bit rate, default 19200
5	Select site number, default 1
6	Site search, which can be searched when there is only one driver connected and the site is not confirmed
7	Connect and disconnect

Table 1 : Communication Interface description

• Control interface

The device monitoring interface is divided into 6 parts: Model Information, Motion Control, Input ports, Basic setting, Device monitoring and Output ports.

Both parts are display sections where product information is automatically read when connected, and device detection is constantly updated to implement the real-time parameters of the device, including the real-time status of motors such as bus voltage driver temperature.



● Connected

Model Info

Device model	SiS17-233-CNFC-TX
HW version	1,00
SW version	1,100

Basic settings

Current(mA)	2000
Microstep(ppr)	4000
Pulse mode	Pulse+Dir

Device monitoring

Voltage(V)	22,97
Temperature(℃)	46,6
Command positon	68000
Actual Positon	68000
Current(A)	0,024
Velocity(rps)	0,00
Driver status	2:Servo ON
Alarm info	0:No alarm

Motion Control

Control mode	Canopen	Set position	0
Homing mode	Fwd to origin	Homing offset	0
Pos.velocity	2000	Velo.velocity	100
JOG.velocity	550	Target Position	40000

Input ports

IN1 ○	No function	Trig	IN5 ○	No function	Trig
IN2 ○	No function	Trig	IN6 ○	No function	Trig
IN3 ○	Servo OFF/ON	Trig	IN7 ○	No function	Trig
IN4 ○	No function	Trig	INZ ○	AIN: 0,012	

Output ports

OUT1 ●	Alarm out
OUT2 ●	In-position
OUT3 ○	Servo ON/OFF
OUT4 ○	Controllable output

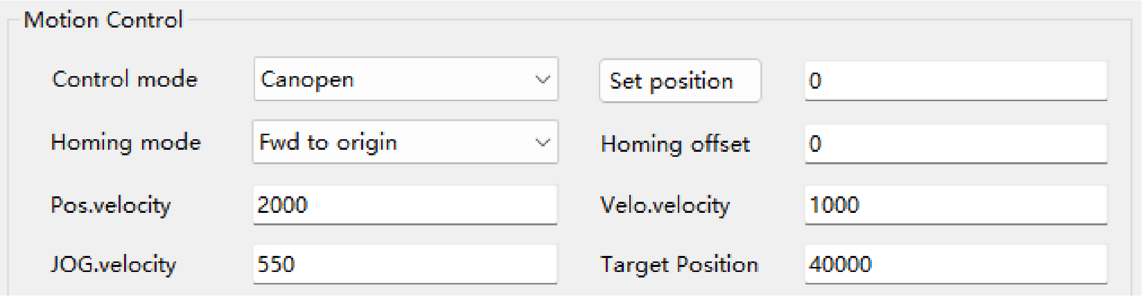
Figure 7 : Control interface

Name Function	Function
Model info	Displays the driver basic information
Motion control	Set motion parameters and perform basic control and alarm clearing
Input ports	Set the input ports configuration and software trigger
Basic setting	Set the run's microstep and current
Device monitoring	Displays the current status of the device
Output ports	Set the output ports configuration

Table 2 : Control interface description

- **Basic setting and motion control**

The first is to set the current setting and segmentation setting, the current setting value should be consistent with the motor current, the segmentation setting determines the number of pulses per revolution of the motor, generally after the first set-up will not change. Then there are the parameters and the corresponding functions.



Motion Control

Control mode: Canopen

Homing mode: Fwd to origin

Pos.velocity: 2000

JOG.velocity: 550

Set position: 0

Homing offset: 0

Velo.velocity: 1000

Target Position: 40000

Figure 8 : Motion control interface

Name	Affected button	Function
Control mode		Change the control mode of the drive, integrate control mode for motion with the software
Homing mode	Homing	Select the homing method
Pos.velocity	Rel run, Abs run, Move+, Move-	Setting the velocity in position mode unit (0.01rps) is independent of microstep
JOG.velocity	Jog+, Jog-	Setting the velocity in Jog mode unit (0.01rps) is independent of microstep
Set position		When set position is pressed, replace the actual position with the value in the box
Homing offset	Homing	Set origin offset, pulses entered in the extra motion box after the return-to-origin motion ends
Velo.velocity	Velo run	Setting the velocity in velocity mode unit (0.01rps) is independent of microstep
Target Position	Rel run, Abs run	Set the target position for the position mode

Table 3 : Motion control interface description

The main concern is the speed of several movements, the units are 0.01rps, do not calculate the subdivision, directly calculate the number of laps per second.

The specific position of operation needs to refer to the instruction position and the actual position, the motor without the encoder cannot display the actual position, always show 0, the motor with the encoder will show the actual position. The operation of all buttons modifies the command position, which is always equal to the actual position in closed-loop mode. Open loop mode may appear deviation, after power-up the default command position is 0, that is, the default origin, the relationship between the instruction position and segmentation is: segmentation set a pulse to run a circle, such as segmentation set 1000, when the command position is 1000, the motor is rotating forward, the command position is -1000, the motor is running a circle in reverse.

Then there is the control mode, the internal pulse control is directly through the drive motor control, can be directly used in the software interface of the ten buttons to achieve motor control. The button functions as follows:

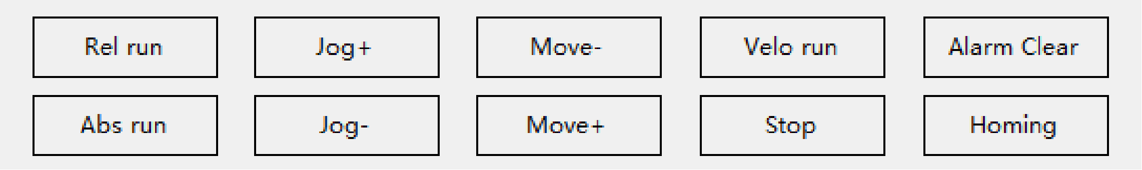


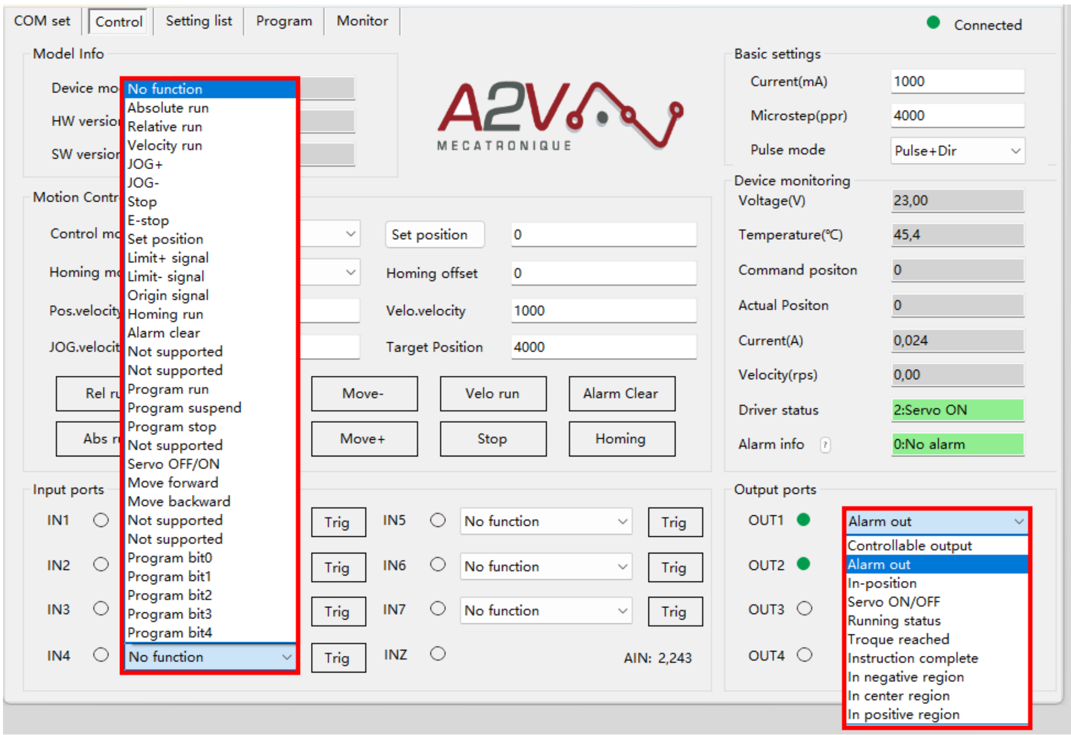
Figure 9 : Motion buttons

Button name	Action
Rel run	Instruction position increases with in function of the target position Unit correspond to pulse number (according to microstep)
Abs run	Instruction position set to the target position Unit correspond to pulse number (according to microstep)
Jog+	Move forward at Jog velocity while holding
Jog -	Move backward at Jog velocity while holding
Move -	Instruction position decrease with in function of the target position. Unit correspond to pulse number (according to microstep)
Move +	Same as Rel run button
Velo mode velo	Run at velocity mode, and the command position automatically increases by microstep
Stop	Stop running while running
Alarm Clear	Clear the alarm when there is an alarm
Homing	Make Homing mode

Table 4 : Motion buttons description

Input ports and Output ports

The input and output port configuration allows for simple control through external triggering. The action can be configure in the control window with drop-down list, or directly by parameters in setting list.



Basic settings
Current(mA) 1000
Microstep(ppr) 4000
Pulse mode Pulse+Dir

Device monitoring
Voltage(V) 23,00
Temperature(℃) 45,4
Command positon 0
Actual Positon 0
Current(A) 0,024
Velocity(rps) 0,00
Driver status 2:Servo ON
Alarm info 0:No alarm

Set position 0
Homing offset 0
Velo.velocity 1000
Target Position 4000

IN1 ○
IN2 ○
IN3 ○
IN4 ○ No function
IN5 ○ No function
IN6 ○ No function
IN7 ○ No function
INZ ○
AIN: 2,243

OUT1 ● Alarm out
OUT2 ●
OUT3 ○
OUT4 ○

Figure 10 : Input and output port configuration

Function Id	Name	Function	Note
0	No function	Make the port invalid, no action	
1	Absolute run	Instruction position set to the target position	
2	Relative run	Instruction position increases in function of the target position	
3	Velocity run	Run at velocity mode	
4	JOG +	Move forward at Jog velocity while holding	
5	JOG-	Move backward at Jog velocity while holding	
6	STOP	Stop running while running in function of deceleration	
7	E-stop	Let the port stop quickly, faster than slow down	
8	Set position	Replace the actual position with the value in the “set position” box	
9	Limit+ signal	Triggers a limit+ signal	Used at homing
10	Limit- signal	Triggers a limit- signal	
11	Origin signal	Triggers a origin signal	
12	Homing run	Strat homing method	
13	Alarm clear	Clear the alarm when there is an alarm	
16	Program run	Start running multiple segments	

17	Program suspend	Pause running multiple segments	
18	Program stop	Stop running multiple segments	
20	Servo OFF/ON	The motor is offline when triggered and resumes when it is not triggered	
21	Move forward	Instruction position increases in function of the target position	
22	Move backward	Instruction position decrease in function of the target position	
25	Program bit0	Input program bit0	Used for multi segmented programming
26	Program bit1	Input program bit1	
27	Program bit2	Input program bit2	
28	Program bit3	Input program bit3	
29	Program bit4	Input program bit4	

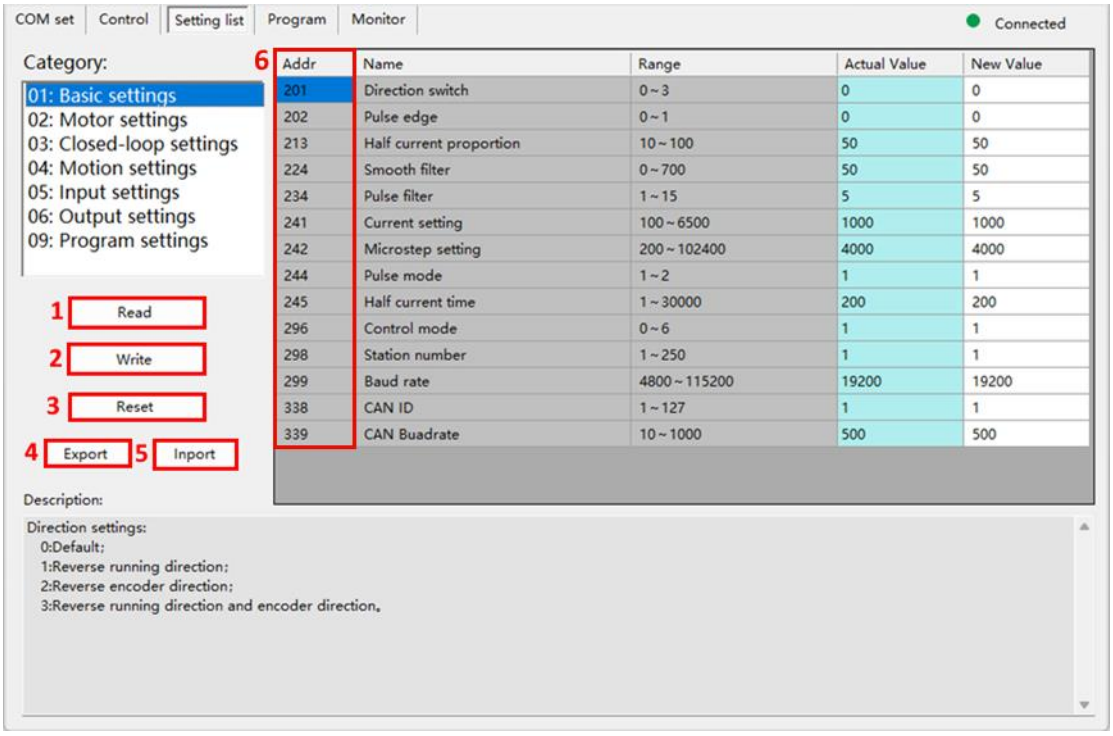
Table 5 : Input port function

Function Id	Name	Function
100	Controllable output	Universal output, which can be triggered by modifying parameters
101	Alarm out	Triggered when no alarm
102	In-position	Triggered when positioning are completed
103	Servo ON/OFF	Triggered when the servo is OFF
104	Running status	Triggered when the motor is running
105	Torque reached	Triggered when torque is reached
106	Instruction complete	Triggered when instruction is completed
107	In negative region	Triggered when position is in negative region
108	In center region	Triggered when position is 0
109	In positive region	Triggered when position is in positive region

Table 6 : Output port function

- **Parameter settings**

Setting list window permit to change the driver parameter, there are not dynamically update. You need to read or write parameter with appropriated button.



Addr	Name	Range	Actual Value	New Value
201	Direction switch	0~3	0	0
202	Pulse edge	0~1	0	0
213	Half current proportion	10~100	50	50
224	Smooth filter	0~700	50	50
234	Pulse filter	1~15	5	5
241	Current setting	100~6500	1000	1000
242	Microstep setting	200~102400	4000	4000
244	Pulse mode	1~2	1	1
245	Half current time	1~30000	200	200
296	Control mode	0~6	1	1
298	Station number	1~250	1	1
299	Baud rate	4800~115200	19200	19200
338	CAN ID	1~127	1	1
339	CAN Buadrate	10~1000	500	500

Direction settings:
0:Default;
1:Reverse running direction;
2:Reverse encoder direction;
3:Reverse running direction and encoder direction.

Figure 11 : Parameter list interface

Id	Function
1	Read the parameters on the driver
2	Write the new parameters value on the driver
3	Reset driver parameters
4	Export driver parameter to external file (.par)
5	Import driver parameter from external file
6	Parameter address, refer to the next section for more details

Table 7 : Parameter list description

Parameter list

Except for the parameters with yellow background which do not save EEPROM automatically, all other parameters save EEPROM automatically, and there is a limitation on the number of times EEPROM can be written, the maximum is 1 million times.

a) Controller basic status (Category 01)

<table>
<tr><th>Adr</th><th>Word</th><th>Items</th><th>Details</th><th>Range/ Unit</th></tr>
<tr><td>0100</td><td>1</td><td>Motor Current</td><td>Real-time motor current value</td><td>0.1%A</td></tr>
<tr><td>0101</td><td>1</td><td>Input Voltage</td><td>Current Input Voltage</td><td>1%V</td></tr>
<tr><td>0104</td><td>2</td><td>The subdivision value</td><td>The subdivision value</td><td>ppr</td></tr>
<tr><td>0106</td><td>1</td><td>Pulse mode</td><td>1 for pulse+ direction mode, 2 for double pulse mode</td><td>1-2</td></tr>
<tr><td>0108</td><td>1</td><td>Error Code</td><td>Alarm code, see 5.2 for content, display "0" for no faults</td><td>-</td></tr>
<tr><td>0109</td><td>1</td><td>operational state</td><td>Driver board operation status, see 5.1 for contents</td><td>-</td></tr>
<tr><td>0110</td><td>1</td><td>hardware version</td><td>Driver Board Hardware Version</td><td>-</td></tr>
<tr><td>0111</td><td>1</td><td>software version</td><td>Driver Board Software Version</td><td>-</td></tr>
<tr><td>0117</td><td>2</td><td>Target Position</td><td>Target Position</td><td>pulse</td></tr>
<tr><td>0119</td><td>1</td><td>Actual velocity display</td><td>-</td><td>0.01rps</td></tr>
<tr><td>0126</td><td>2</td><td>Actual position</td><td>Actual position</td><td>pulse</td></tr>
<tr><td>0174</td><td>1</td><td>IO selects program running paragraphs</td><td>-</td><td>-</td></tr>
<tr><td>0176</td><td>1</td><td>Program error No</td><td>-</td><td>-</td></tr>
<tr><td>0178</td><td>1</td><td>Program Running No</td><td>-</td><td>-</td></tr>
<tr><td>0135</td><td>1</td><td>Input Port Status</td><td><table><tr><td>Data Bit</td><td>Bit6</td><td>Bit5</td><td>Bit4</td><td>Bit3</td><td>Bit2</td><td>Bit1</td><td>Bit0</td></tr><tr><td>Input Port</td><td>X7</td><td>X6</td><td>X5</td><td>X4</td><td>X3</td><td>X2</td><td>X1</td></tr></table></td><td></td></tr>
<tr><td>0136</td><td>1</td><td>Output Port Status</td><td><table><tr><td>data bit</td><td>Bit1</td><td>Bit0</td></tr><tr><td>output port</td><td>Y2</td><td>Y1</td></tr></table></td><td></td></tr>
<tr><td>0144</td><td>1</td><td>Analog Input Voltage</td><td></td><td>0.001V</td></tr>
</table>

Table 8: Controller basic status list

b) Basic Parameter Setting (Category 02)

Adr	Word	Items	Details	Range/Unit
0201	1	*Motor direction switching	Selection of motor running direction (valid for re-start)	0 1～
0206	1	user command	Setting when the motor is stopped 1: User parameter reset 2: Clear alarms (except some hardware failure alarms) 3: Driver board reboot	0~5
0213	1	Idle current ratio	Idle current ratio (valid in open loop mode)	10%~120%
0217	1	*Motor control mode	0: open loop 1: Closed Loop Default: 1	0 1～
0224	1	angular filtering	The smaller the value, the smoother the motor runs, but also the higher the delay	1~700
0234	1	digital filtering	Filter coefficient of the input pulse, the larger the value, the lower the input frequency response	1~15
0241	1	*Input current	Setting current	400~6500 0.4A~-6.5A
0242	2	*Set breakdown	Number of pulses per revolution	200~102400 ppr
0244	1	*Pulse mode	1: Pulse+ Direction Mode 2: Double pulse mode	1~2
0245	1	half-hour	Delay time to enter half-current state after motor stops running (valid in open-loop mode)	1~32767 ms

0296	1	*Operation mode selection	Control mode setting:
			0: Pulse control;
			1: Internal control;
			2: Invalid;
			3: Invalid;
			4: Invalid;
			5: Thrust control;
			6: Position downward pressure control;
			Default: 0
				0~1

0298	1	*MODBUS communication address	Default: 1	1~255
0299	2	*MODBUS baud rate for communication	Default: 19200	4800~115200 bps
0338	1	*CAN conmonication address	Default: 1	1~127
0339	1	*CAN baud rate for communication	Default: 500	10~1000 kbps

Table 9 : Basic parameter setting list

c) Motor Parameters (Classification 03)

Adr	Word	Items	Details	Range/Unit
0200	1	Current loop Kp	Current loop Kp Default: 800	50~20000
0215	1	Current loop Ki	Current loop Ki Default: 800	50~20000
0222	1	Current loop Kp max.	Current loop Kp max. Default: 800	50~20000
0225	1	Current boost	Current automatically adjusts the proportional maximum value. Open loop effective Default: 1000	1000~2000
0228	1	Current loop Ki max.	Current loop Ki max. Default: 800	50~20000
0277	1	Anti-resonance factor	Anti-resonance factor. Open loop effective. Default: 100	0~500
0295	1	Auto motor parameter detection	Auto-detect motor parameter setting: (valid for restart) 0: Set the motor parameters manually; 1: Automatic detection of motor parameters. Default: 1	0~1

Table 10: Motor parameters list

d) Closed loop Parameterization (Category 04)

Adr	Word	Items	Details	Range/Unit
0217	1	*Motor mode	Motor mode: (valid at restart) 0: Open loop; 1: Closing the loop.	0~1
0246	1	*Encoder resolution	Resolution (ppr) = number of encoder lines (cpr) x 4. (Valid at restart). Default: 4000	200~65535 Encoder Resolution
0247	1	In position range	Reach the target position proximity distance, output in place signal Default: 0	1~1000 Encoder Resolution
0248	1	Flux weakening limit	Weak magnetic limitations. Closed loop mode only.	0~1000
0249	1	Flux weakening coefficient 1	Weak magnetic coefficient 1. Closed loop mode only.	0~1000
0250	1	Flux weakening coefficient 2	Weak magnetic coefficient 2. Closed loop mode only.	0~1000
0251	1	Speed loop kp	Velocity loop Kp. Closed loop mode only.	0～ 30000
0252	1	Speed loop Ki	Velocity loop Ki. Closed loop mode only.	0～ 30000
0253	1	Speed loop Kd	Velocity loop Kd. Closed loop mode only.	0～ 30000
0255	1	Position loop Kp	Position loop Kp. Closed loop mode only.	0～ 30000
0256	1	Position loop Ki	Position loop Ki. Closed loop mode only.	0～ 30000
0257	1	Position loop Kd	Position loop Kd. Closed loop mode only.	0～ 30000
0258	1	Position error threshold	Position overshoot threshold, value is the encoder resolution. Closed loop mode only. Default: 1000	0～ 30000 encoder resolution
0264	1	Current loop Kd filter	KD filter coefficients. Closed loop mode only. Default: 50	0～ 1000
0265	1	Kd filter gain	KD Gain. Closed loop mode only. Default: 50	0～ 1000
0266	1	Current filter	Current Coefficient. Closed loop mode only.	0～ 1000

Table 11: Closed loop parameterization list

e) Control parameters (Category 05)

Adr	Word	Items	Details	Range/Unit
0301	1	Start velocity	Default: 100	1~2000 0.01~20rps
0302	1	Stop velocity	Default: 100	1~2000 0.01~20rps
0303	1	Accelerations	Default: 100	5~10000 rps^2
0304	1	Deceleration	Default: 100	5~10000 rps^2
0305	1	Homing Mode	Return to home mode. 0: Clockwise back to the home position 1: Counterclockwise back to the origin 2: Clockwise back to the limit 3: Counterclockwise back to the limit 8: Clockwise z-pulse back to home position 9: Counterclockwise z-pulse back to home position	0~30
0306	1	Velocity for position mode	Default: 1000	1~5000 0.01~50rps
0307	1	Velocity for velocity mode	When in speed mode, the running direction is the same as the speed direction Default: 1000	-5000~5000 -50~50rps
0308	1	Velocity for JOG mode	Default: 100	1~5000 0.01~50rps
0309	1	Homing approach velocity	Default: 200	1~5000 0.01~50rps
0310	1	Homing creep velocity	Running speed after hitting the origin Default: 100	1~5000 0.01~50rps
0311	2	Return to Home Offset	Default: 0	-2000000000~ 2000000000 pulse
0313	2	Output pulse	travel distance Absolute Position Mode: Run to a specified position Relative Position Mode: Runs set offset stroke Default: 0	-2000000000~ 2000000000 pulse
0317	2	Positive soft limit	Default: 2000000000 Note: Invalid during return to home position	-2000000000~ 2000000000 pulse

0319	2	Negative soft limit	Default: -2000000000 Note: Invalid during return to home position	-2000000000~ 2000000000 pulse
0321	2	Setting the current position	Default: 0	-2000000000~ 2000000000 pulse
0323	1	Control Word	0: Empty 1、Absolute operation, running to the set distance, the running direction is determined by the distance positive and negative, the speed positive and negative values are invalid, in the process of running to modify the target position effective 2、Relative operation, run at the set distance and running speed, the running direction is determined by the distance positive and negative, the speed positive and negative values are invalid, in the running process to modify the movement distance is invalid 3. Speed mode 4、Positive point movement 5、Reverse point movement 6. Deceleration stop 7. Emergency stop 8、Set the current position only when the motor stops 12. Back to the origin 13. Alarm Clearance 14: Program data validation 15: Program data retention 16: Program data start 17: Program data suspension 18: End of program data Default: 0	0~29

0324	1	*Memory switch	Memory control switch, pulse control mode is invalid; bit0: Enable the forward soft limit function; bit1: Enable reverse soft limit function; bit3-4: 00: The communication command selects multiple segments of data at address 328; 01: Multi-paragraph IO selects the appropriate paragraph and executes it; 02: Multi-paragraph IO selects the corresponding paragraph by binary combination, and the execution is triggered by the multi-paragraph start IO. bit5: Enable power-on automatic Program running function (Program segment selection is set via address 490);	0-65535
0327	1	Number of paragraphs	Default: 1	1~32
0328	1	Program selection	Default: 0 Note: If the IO port is configured with the program selection function, the IO configuration with program selection takes precedence.	0~31

Table 12 : Control parameters list

NB: The speed parameter range represents only the range of data acceptable to the controller, and does not mean that the speed (high speed) can actually be achieved, as limited by the following factors:

- Maximum motor response speed
- Maximum control frequency of the controller (automatic limitation when out of range)

Therefore, if high speed operation is required, commissioning should start from the minimum subdivision. If ultra-low speed operation is required, commissioning should start from the maximum subdivision.

f) Input Block Designation (Category 06)

Adr	Word	Items	Details	Range/Unit
0400	1	IN1 (X1) function selection	0: Empty 1: Absolute operation 2: Relative operation 3: Speed mode 4: Positive point movement 5: Reverse point movement 6: Deceleration stop 7: Emergency stop 8: Set the current position only when the motor stops 9: Positive Limit 10: Negative Limit 11: Home signal 12: Back to the origin 13: Alarm Clearance 14: Program data validation 15: Program data retention 16: Program data 17: Program data suspension 18: End of program data 20. Enable 25: IO Port Configuration Program Selection Bit0 26: IO Port Configuration Program Selection Bit1 27: IO Port Configuration Program Selection Bit2 28: IO Port Configuration Program Selection Bit3 29: IO Port Configuration Program Selection Bit4 Default: 0	0~30
0401	1	IN2 (X2) function selection	Setting content is the same as IN1 (X1) (default value: 0).	0~30
0402	1	IN3 (X3) Function Selection	Setting content is the same as IN1 (X1) (default value: 0).	0~30
0403	1	IN4 (X4) function selection	Setting content is the same as IN1 (X1) (default value: 0).	0~30
0405	1	IN5 (X5) function selection (PUL port)	Setting content is the same as IN1 (X1) (default value: 0). (Port function disabled when external pulse is applied)	0~30
0406	1	IN6 (X6) function selection (DIR port)	Setting content is the same as IN1 (X1) (default value: 0). (Port function disabled when external pulse is applied)	0~30
0429	1	Universal Digital Input Logic		

0410	1	Pseudo communication setting IN1 (X1)	0: OFF (initial value 0) 1: ON (triggers action configured in IN1 (X1))	0~1
0411	1	Pseudo-communication setting IN2 (X2)	0: OFF (initial value 0) 1: ON (triggers action configured in IN2 (X2))	0~1
0412	1	Pseudo-communication setting IN3 (X3)	0: OFF (initial value 0) 1: ON (triggers action configured in IN3 (X3))	0~1
0413	1	Pseudo-communication setting IN4 (X4)	0: OFF (initial value 0), 1: ON (triggers action configured in IN4 (X4))	0~1
0415	1	Pseudo-communication setting IN5 (X5)	0: OFF (initial value 0), 1: ON (triggers action configured in IN5 (X5))	0~1
0416	1	Pseudo-communication setting IN6 (X6)	0: OFF (initial value 0) 1: ON (triggers action configured in IN6 (X6))	0~1

Table 13 : Input block designation list

g) Output Block Designation (Category 07)

<table>
<tr><th>Adr</th><th>Word</th><th>Items</th><th>Details</th><th>Range/Unit</th></tr>
<tr><td>0420</td><td>1</td><td>OUT1 (Y1) function selection</td><td>100: Universal port
101: Alarm output function
There is an output signal when there is no alarm, and no output signal when there is an alarm.
102: Signal in place
103: Enable control output:
There is an output signal when it is offline and no output signal when it is enabled.
(Default value: 101)</td><td>100~104</td></tr>
<tr><td>0421</td><td>1</td><td>OUT2 (Y2) function selection</td><td>Setting content is the same as OUT 1 (Y2) (default value: 100).</td><td>100~104</td></tr>
<tr><td>0428</td><td>1</td><td>Universal digital output control</td><td>Output Port Output Settings<table><tr><td>data bit</td><td>Bit1</td><td>Bit0</td></tr><tr><td>output port</td><td>OUT2 (Y2)</td><td>OUT1 (Y1)</td></tr></table></td><td></td></tr>
<tr><td>0430</td><td>1</td><td>Digital output logic</td><td>Corresponding output port logic<table><tr><td>data bit</td><td>Bit1</td><td>Bit0</td></tr><tr><td>output port</td><td>OUT2 (Y2)</td><td>OUT1 (Y1)</td></tr></table></td><td></td></tr>
</table>

Table 14 : Output block designation list

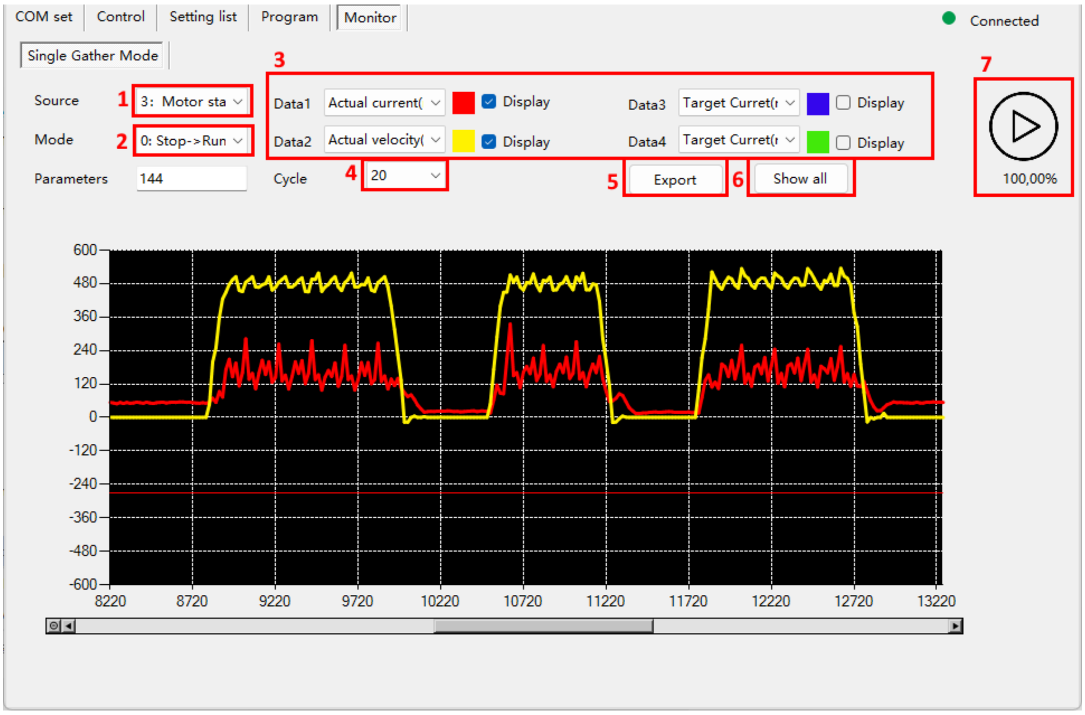
h) Multi-segment programming Instruction Designation (Category 09)

Adr	Word	Items	Details	Range/Unit
0480	2	Global variable 1	Multi-segment use: global variable 1	0~65535
0481	2	Global variable 2	Multi-segment use: global variable 2	0~65535
0482	2	Global variable 3	Multi-segment use: global variable 3	0~65535
0483	2	Global variable 4	Multi-segment use: global variable 4	0~65535
0484	2	Global variable 5	Multi-segment use: global variable 5	0~65535
0485	1	Multi-segment self-starting segment selection	Multi-segment self-starting segment selection	0~65535

Table 15 : Multi-segment programming instruction list

- **Monitor window**

This window permit to record specific value of the driver, data can be read on a chart or can be export to csv.



Single Gather Mode

Source 3: Motor sta

Mode 0: Stop->Run

Parameters 144

Data1 Actual current(🟥 ☑ Display

Data2 Actual velocity(🟨 ☑ Display

Data3 Target Curret(r 🟦 ☐ Display

Data4 Target Curret(r 🟩 ☐ Display

Cycle 20

Figure 12 : Display interface

Id	Function
1	Triger source to start capture
2	Mode of trigger
3	Select data signals
4	Choose number of cycle recording
5	Export the data into csv file
6	Unzoom to see all signals
7	Launch recording

Table 16 : Display interface description

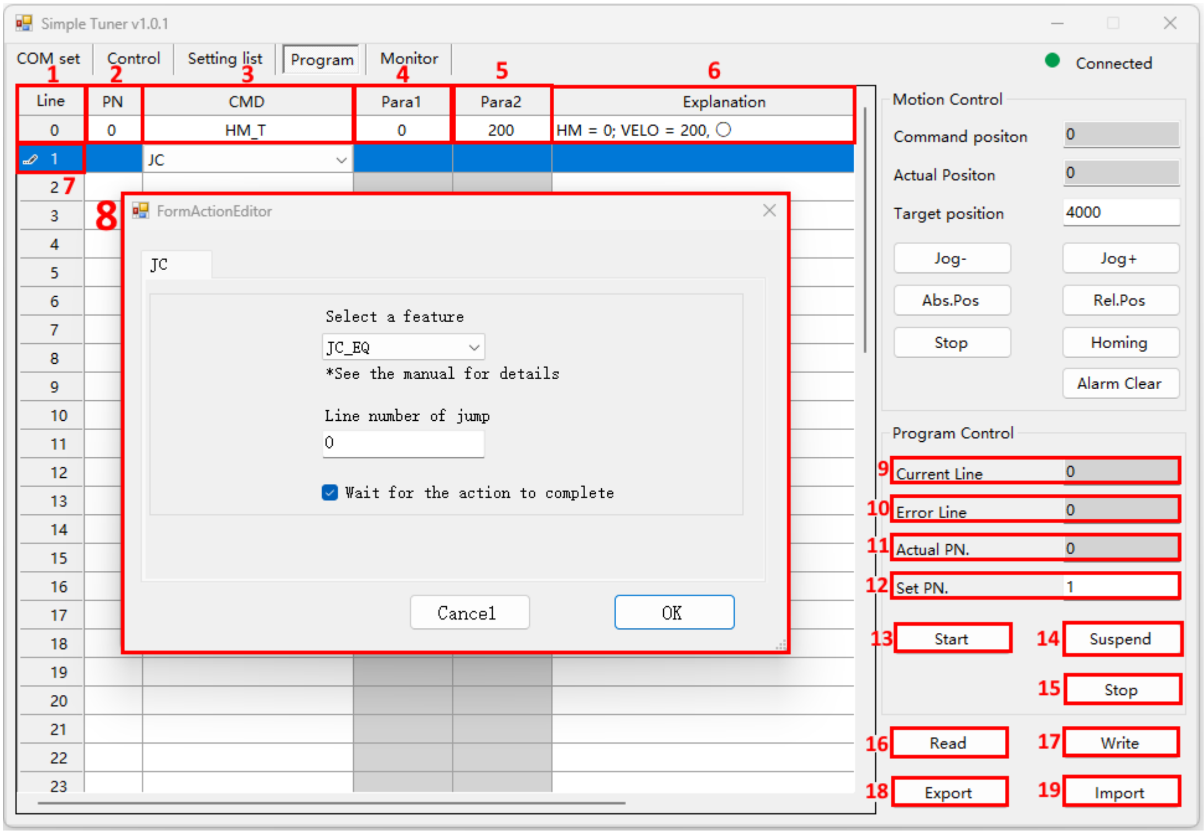
- ## Programming interface

▪ Programming Feature Introduction

Multi-segment position mode is a way to combine multiple position segments in a certain order, trigger motion through an external IO signal, and complete a series of position segment actions. This function can be regarded as a multi-segment combination of position mode, the user can several segments of the description parameters such as deceleration, pulse number, etc. are stored in the EEPROM in advance, need to enable these position segments when only need to provide a trigger signal to complete the work.

▪ Programming Writing

Multi-segment functions can write multiple paragraphs, each of which can set its own motion, supporting up to 16 paragraphs, each of which must end with a paragraph to trigger properly. Paragraph content can be randomly arranged, first in the command bar drop-down, select the need for instructions, most instructions only parameter one can enter the corresponding parameters. Column explain, is automatically generated, allows to have a simple representation of the command line.



Line	PN	CMD	Para1	Para2	Explanation
0	0	HM_T	0	200	HM = 0; VELO = 200, ○
1		JC			
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					

Select a feature: JC_EQ

*See the manual for details

Line number of jump: 0

☑ Wait for the action to complete

Command positon: 0
Actual Positon: 0
Target position: 4000

Current Line: 0
Error Line: 0
Actual PN.: 0
Set PN.: 1

Figure 13 : Program interface

ID	Function
1	Line numbers displayed while editing multiple programs, automatically generated and cannot be edited
2	Program number, automatically generated during program editing, with the next program number automatically generated after the end command of the current program. The user can specify the beginning program number at ⑩, or specify the program number at the input port. For the specific method of program selection, refer to the parameter address "324: Memory control switch" bits 3~4 setting.
3	Command, see the current manual "3.Programming features"
4	Command parameters, automatically generated according to the command edit box ⑧, can also be manually edited.
5	Command parameters, automatically generated according to the command edit box ⑧, can also be manually edited.
6	Command description, a simple description of the current command function based on the command settings.
7	Command edit trigger: Click to pop up the "Command edit box ⑨".
8	Command edit: Set the command based on "3.Programming features".
9	Running line number, displays the current line number being executed by the program.
10	Error line number, the line number where an error occurred during program execution. If an error occurs in command operation, refer to this line number to check if the current line's execution result is valid.
11	Selected program, the current running program.
12	Specify program, effective when the memory switch is set to "0: The communication command selects multiple segments of data at address 328". Otherwise, programs are selected by other means.
13	Start program, trigger button for start program.
14	Pause program, pause the command operation.
15	Stop program, trigger button for stop the program.
16	Read, read program data from the driver and display it in the software
17	Write, write the current program data from the software to the driver.
18	Export, export the program data from the software to the PC and save it as a ".dat" file.
19	Import, import the ".dat" file previously saved on the PC into the software.

Table 17 : Program interface description

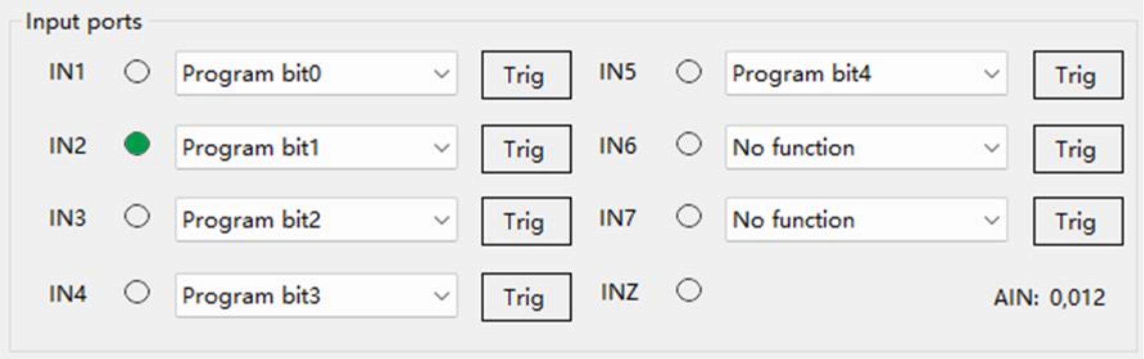
Programming Segment Selection

The choice of multi-segment paragraph number is based on multi-segment bit, multi- segment bit needs to be entered through the input port, the multi-segment bit has a total of four digits, the default is all 0, you can set the multi-segment bit at the input port, and then triggered by input, below is the relationship between multi-segment bit and multi-segment selection.

Bit0	Bit1	Bit2	Bit3	Bit4	Steps
0	0	0	0	0	0
1	0	0	0	0	1
0	1	0	0	0	2
1	1	0	0	0	3
…	…	…	…	…	…
1	1	1	1	1	31

Table 18 : Segment selection in function of program bits

For example, when bit1 is at high level, the number of segments is selected as 2, and when starting multiple segments is enabled, a multi-segment with segment number 2 runs



Input ports

IN1	○	Program bit0	Trig	IN5	○	Program bit4	Trig
IN2	●	Program bit1	Trig	IN6	○	No function	Trig
IN3	○	Program bit2	Trig	IN7	○	No function	Trig
IN4	○	Program bit3	Trig	INZ	○	AIN: 0,012	

Figure 14 : Program bit configuration

3. Programming features

Overview

The programming section includes 5 accumulators (ACC), 5 global variables, and no save function by multi-segment write. The 5 global variable (GV) can be saved to the EEPROM via the software in settings list.

Functions

For each line in program you can choose one function, a window appear to change the parameter of this functions. The list of function available below:

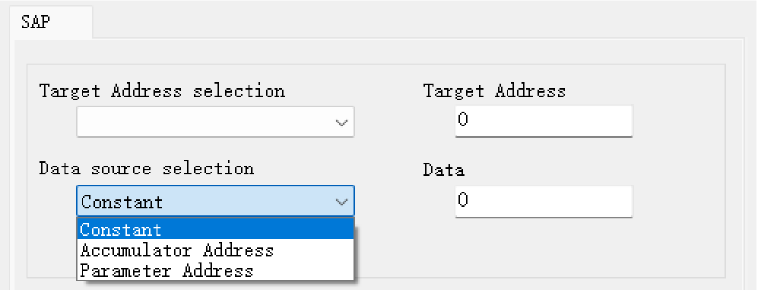
Function		Description
STOP	STOP	Deceleration stops
SAP	Set parameter	Read and write parameters
COMP	COMPARE	Conditional selection
JC	JUMP	Jump to subroutine
CALC	CALCULATE	compute
PP_T	Place Position Target	Positioning runs
PV_T	Place Velocity Target	Speed runs
HM_T	Homing	Homing
SETXY_T	Set Axis Location	Set the current location
WAIT_T	WAIT	Wait
Program_end	END	End of the paragraph (also stop the program)

Table 19 : Program function

Data selection

There are three types of data selection: constant, accumulator, and parameter (including general parameters), which can be selected according to different instruction requirements.

Data selection



SAP

Target Address selection:

Target Address: 0

Data source selection: Constant

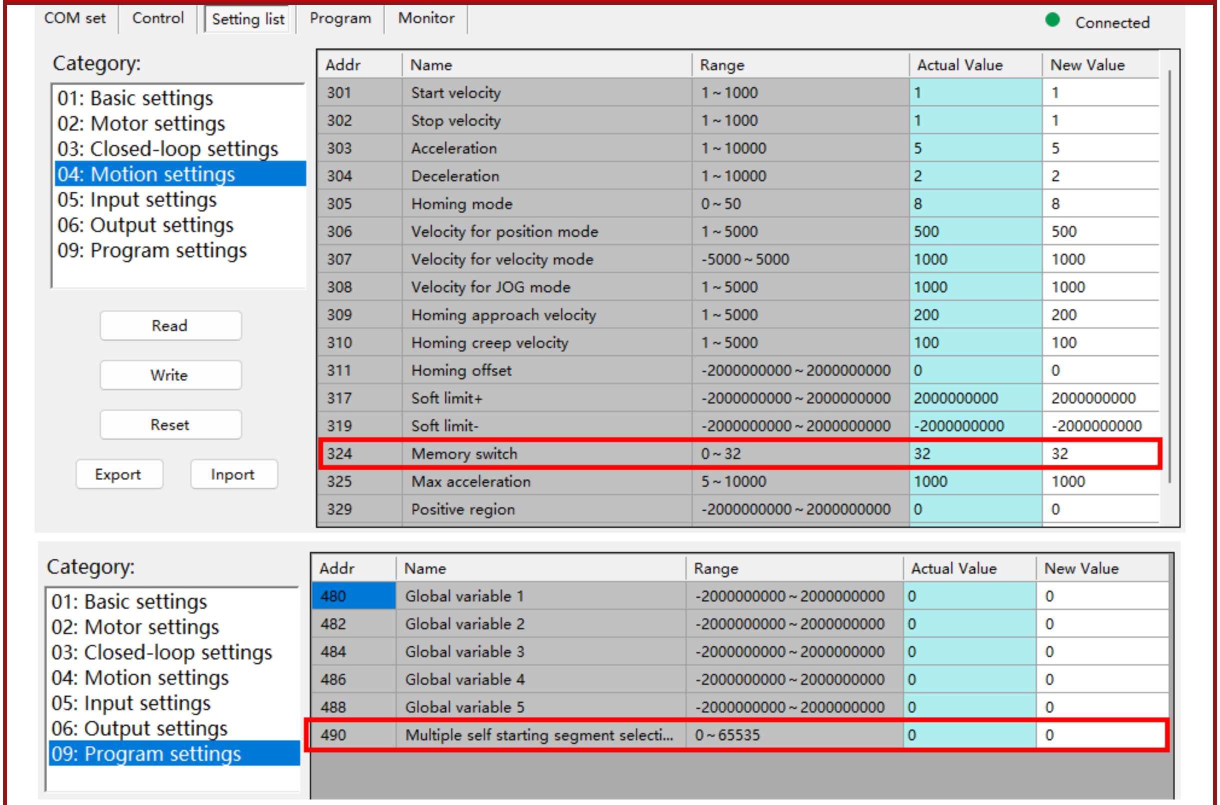
- Constant
- Accumulator Address
- Parameter Address

Data: 0

Auto execute program

According to the actual needs, you can choose whether you need to executing the program at the power ON.

Activate auto execution of program PN 0



Category:

- 01: Basic settings
- 02: Motor settings
- 03: Closed-loop settings
- 04: Motion settings
- 05: Input settings
- 06: Output settings
- 09: Program settings

Addr	Name	Range	Actual Value	New Value
301	Start velocity	1 ~ 1000	1	1
302	Stop velocity	1 ~ 1000	1	1
303	Acceleration	1 ~ 10000	5	5
304	Deceleration	1 ~ 10000	2	2
305	Homing mode	0 ~ 50	8	8
306	Velocity for position mode	1 ~ 5000	500	500
307	Velocity for velocity mode	-5000 ~ 5000	1000	1000
308	Velocity for JOG mode	1 ~ 5000	1000	1000
309	Homing approach velocity	1 ~ 5000	200	200
310	Homing creep velocity	1 ~ 5000	100	100
311	Homing offset	-2000000000 ~ 2000000000	0	0
317	Soft limit+	-2000000000 ~ 2000000000	2000000000	2000000000
319	Soft limit-	-2000000000 ~ 2000000000	-2000000000	-2000000000
324	Memory switch	0 ~ 32	32	32
325	Max acceleration	5 ~ 10000	1000	1000
329	Positive region	-2000000000 ~ 2000000000	0	0

Category:

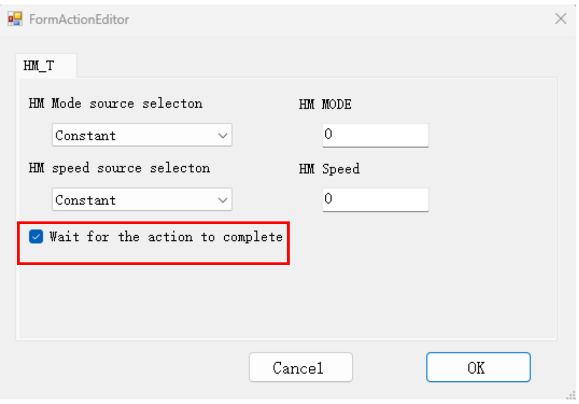
- 01: Basic settings
- 02: Motor settings
- 03: Closed-loop settings
- 04: Motion settings
- 05: Input settings
- 06: Output settings
- 09: Program settings

Addr	Name	Range	Actual Value	New Value
480	Global variable 1	-2000000000 ~ 2000000000	0	0
482	Global variable 2	-2000000000 ~ 2000000000	0	0
484	Global variable 3	-2000000000 ~ 2000000000	0	0
486	Global variable 4	-2000000000 ~ 2000000000	0	0
488	Global variable 5	-2000000000 ~ 2000000000	0	0
490	Multiple self starting segment selecti...	0 ~ 65535	0	0

- **Command waiting for commend completed**

Based on actual needs, you can decide whether to wait for the command to complete before executing the next action.

Whether to wait for the action to complete the selection



FormActionEditor

HM_T

HM Mode source selecton: Constant

HM MODE: 0

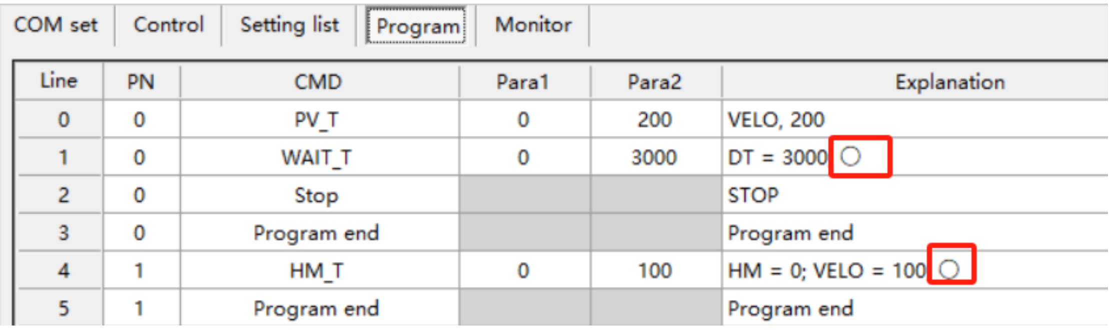
HM speed source selecton: Constant

HM Speed: 0

☑ Wait for the action to complete

After selecting, when the action is complete, a “○” indicator will appear in the explanation section.

Wait for action to complete



Line	PN	CMD	Para1	Para2	Explanation
0	0	PV_T	0	200	VELO, 200
1	0	WAIT_T	0	3000	DT = 3000 ○
2	0	Stop			STOP
3	0	Program end			Program end
4	1	HM_T	0	100	HM = 0; VELO = 100 ○
5	1	Program end			Program end

- **Programming command**

➢ SAP : Read/write parameters

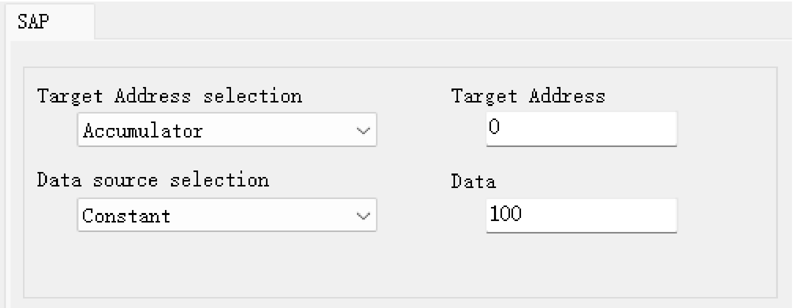
Set the data in parameters or accumulators, for parameter address please refer to the section “Parameter list”

a) Write an accumulator

There are **5 accumulators** in the system that provide the storage of temporary data during the programming process. The programming process sets the address range of the accumulator **0~4**, and setting more than the range will cause an error and the program cannot run.

a1) Constants are written to the accumulator

Set the constant 100 to the accumulator 0



Target Address selection: Accumulator

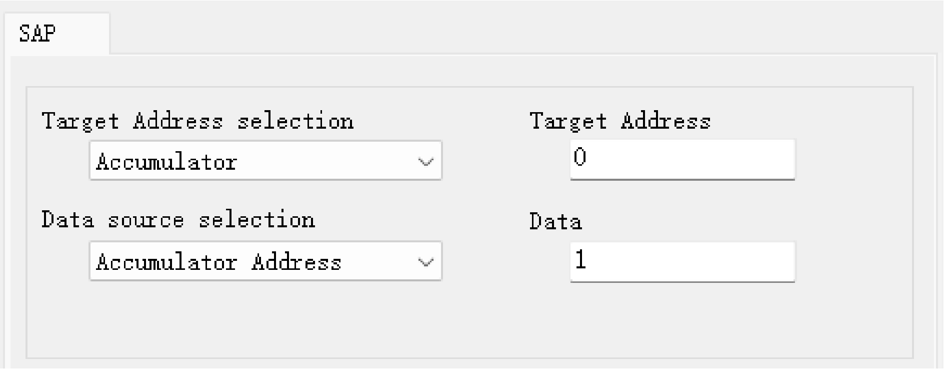
Target Address: 0

Data source selection: Constant

Data: 100

a2) Data in the accumulator is written to the accumulator

Write the data in accumulator 1 to accumulator 0



Target Address selection: Accumulator

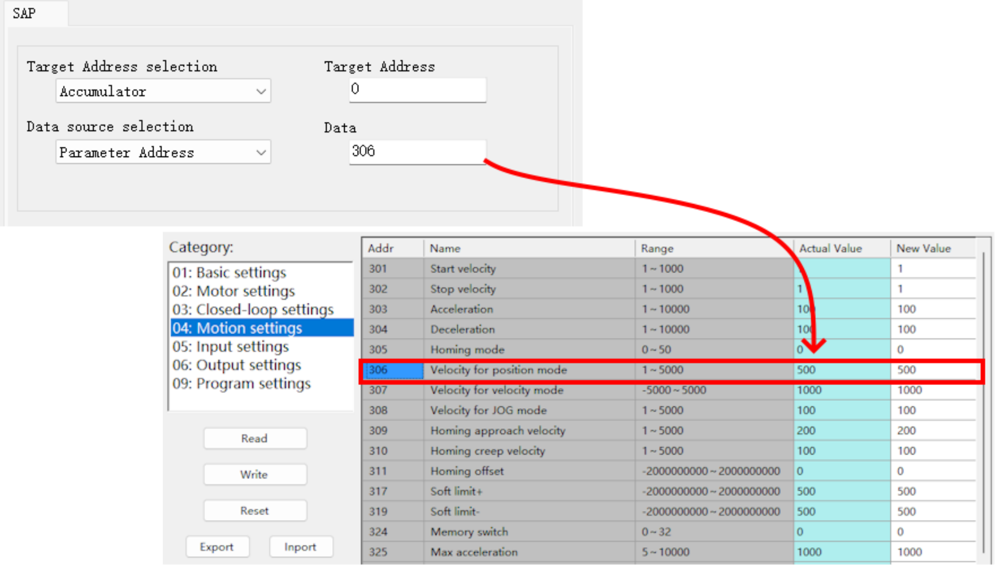
Target Address: 0

Data source selection: Accumulator Address

Data: 1

a3) Write the data in the parameter address to the accumulator

Write the data in parameter address 306 to accumulator 0



SAP

Target Address selection: Accumulator

Target Address: 0

Data source selection: Parameter Address

Data: 306

Category:
- 01: Basic settings
- 02: Motor settings
- 03: Closed-loop settings
- 04: Motion settings
- 05: Input settings
- 06: Output settings
- 09: Program settings

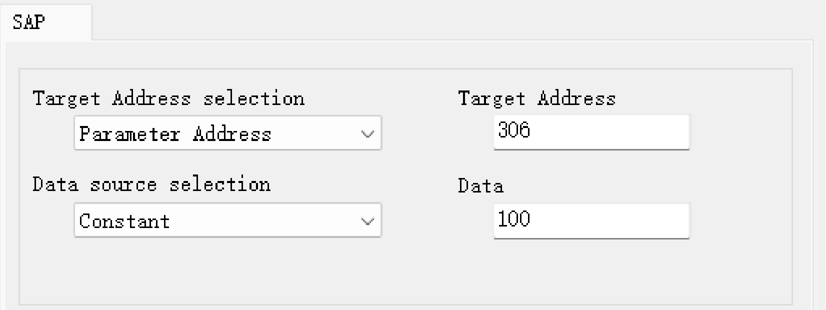
Addr	Name	Range	Actual Value	New Value
301	Start velocity	1～1000	[illegible]	1
302	Stop velocity	1～1000	1	1
303	Acceleration	1～10000	10	100
304	Deceleration	1～10000	10	100
305	Homing mode	0～50	0	0
306	Velocity for position mode	1～5000	500	500
307	Velocity for velocity mode	-5000～5000	1000	1000
308	Velocity for JOG mode	1～5000	100	100
309	Homing approach velocity	1～5000	200	200
310	Homing creep velocity	1～5000	100	100
311	Homing offset	-2000000000～2000000000	0	0
317	Soft limit+	-2000000000～2000000000	500	500
319	Soft limit-	-2000000000～2000000000	500	500
324	Memory switch	0～32	0	0
325	Max acceleration	5～10000	1000	1000

b) Write the parameter address

The parameter address 306 corresponds to the velocity for position mode, all there method permit to change this parameter.

b1) Constant are written to the parameter address

Set the constant 100 to the parameter address 306



SAP

Target Address selection: Parameter Address

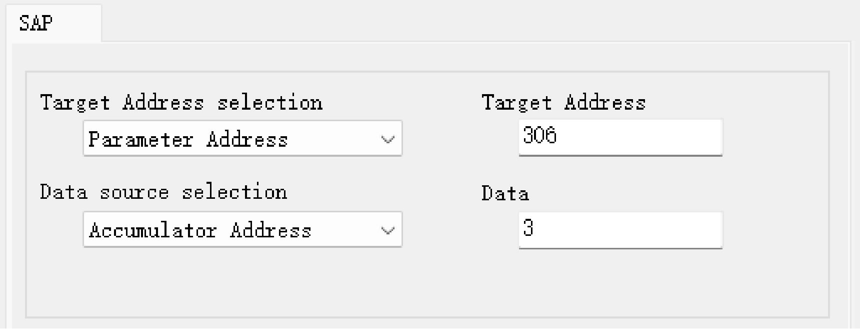
Target Address: 306

Data source selection: Constant

Data: 100

b2) Write the data in accumulator 3 to parameter address

Set the data in the accumulator 3 to the parameter address 306


SAP

Target Address selection: Parameter Address

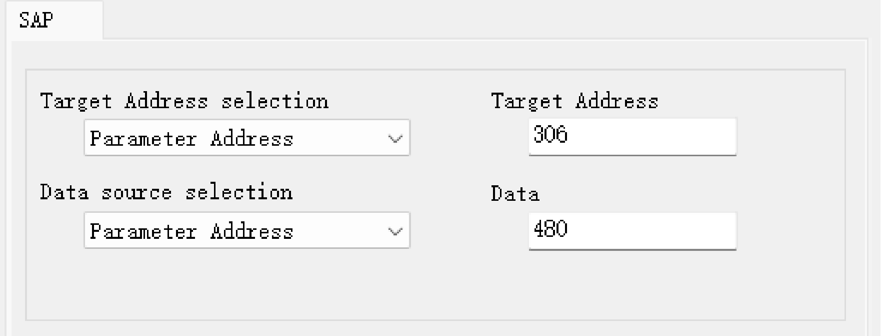
Target Address: 306

Data source selection: Accumulator Address

Data: 3

b3) Write global variable to the parameter address

Set the data in parameter address 480 (Global variable 1) to parameter address 306


SAP

Target Address selection: Parameter Address

Target Address: 306

Data source selection: Parameter Address

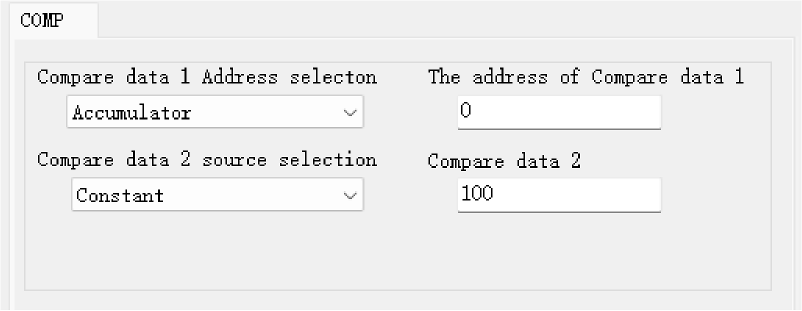
Data: 480

COMP : Conditional Selection

Select the data source you want to compare, must be use with **JC** function.

a) Comparison of accumulator with constant

The data in accumulator 0 is compared with the constant 100



Compare data 1 Address selecton: Accumulator

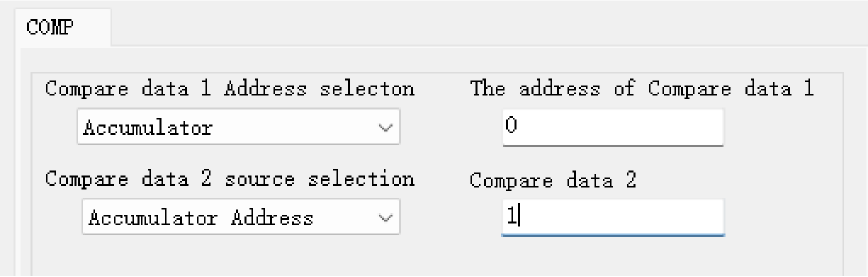
The address of Compare data 1: 0

Compare data 2 source selection: Constant

Compare data 2: 100

b) Comparison of accumulator with accumulator

Compare the data in accumulator 0 with the data in accumulator 1



Compare data 1 Address selecton: Accumulator

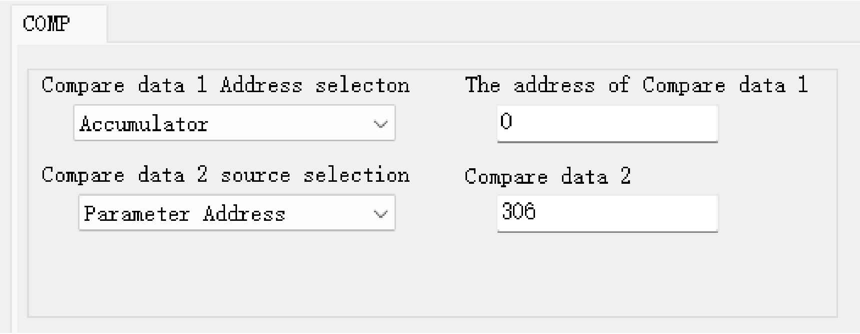
The address of Compare data 1: 0

Compare data 2 source selection: Accumulator Address

Compare data 2: 1

c) Comparison of accumulator with parameter

The data in accumulator 0 were compared with the data in parameter 306



Compare data 1 Address selecton: Accumulator

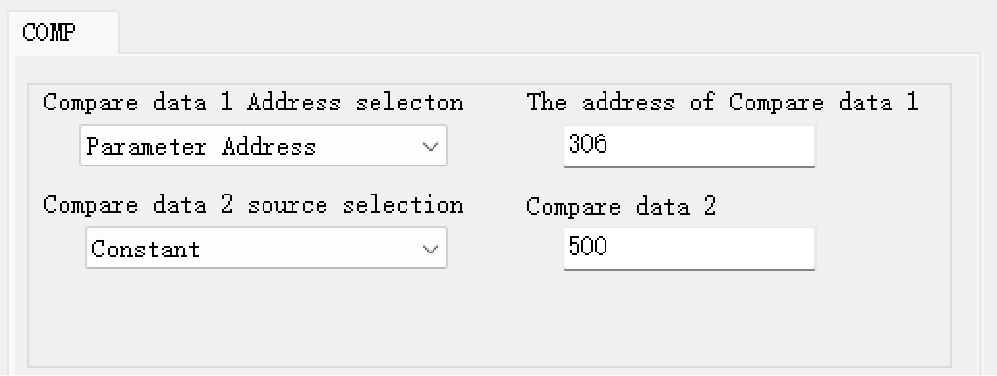
The address of Compare data 1: 0

Compare data 2 source selection: Parameter Address

Compare data 2: 306

d) Comparison of parameter with constant

The data in parameter 306 were compared with the constant 500


COMP

Compare data 1 Address selecton: Parameter Address

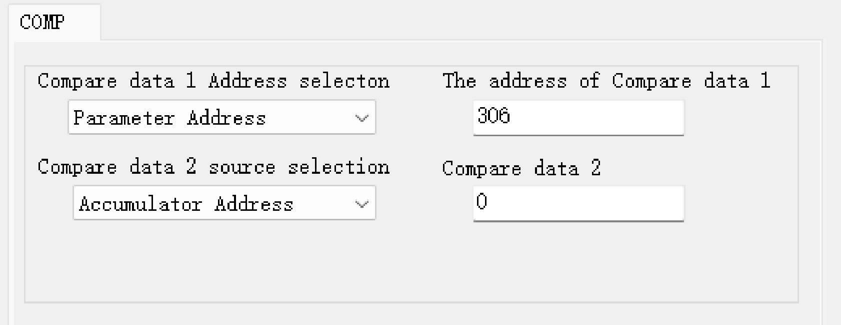
The address of Compare data 1: 306

Compare data 2 source selection: Constant

Compare data 2: 500

e) Comparison parameters with accumulator

The data in parameter 306 were compared with the data in accumulator 0


COMP

Compare data 1 Address selecton: Parameter Address

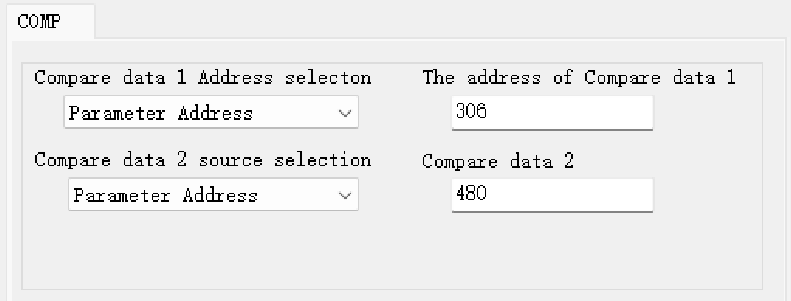
The address of Compare data 1: 306

Compare data 2 source selection: Accumulator Address

Compare data 2: 0

f) Comparison of parameter with parameter

The data in parameter 306 were compared with the data in parameter 480


COMP

Compare data 1 Address selecton: Parameter Address

The address of Compare data 1: 306

Compare data 2 source selection: Parameter Address

Compare data 2: 480

➢ JC : Jump

Select the data according to the COMP, judge the data according to the conditions, and jump to the set line number when the condition is established, otherwise the next row will be automatically executed.

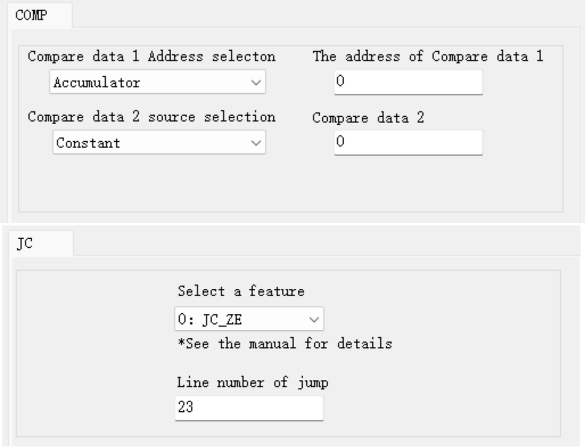
It can be used with COMP (0-7), under certain conditions (8-11), or directly (12).

#	Name	Condition to jump to the set line
0	JC_ZE	Is equal to 0
1	JC_NZ	Is not equal to 0
2	JC_EQ	Equal
3	JC_NE	Not equal
4	JC_GT	Greater than
5	JC_GE	Greater than or equal to
6	JC_LT	Less than
7	JC_LE	Less than or equal to
8	JC_ETO	Timed out
9	JC_INPOSITION	Position reach (PP motion)
10	JC_MAXSPEED	Speed reach (PV motion)
11	JC_HOMEOK	Homing finish (HM motion)
12	JC_NONE	No condition (always jump)

Table 20 : Jump function feature

a) JC_ZE : Equal to 0

If the data in the accumulator is equal to 0, jump to line 23



COMP

Compare data 1 Address selecton: Accumulator

The address of Compare data 1: 0

Compare data 2 source selection: Constant

Compare data 2: 0

JC

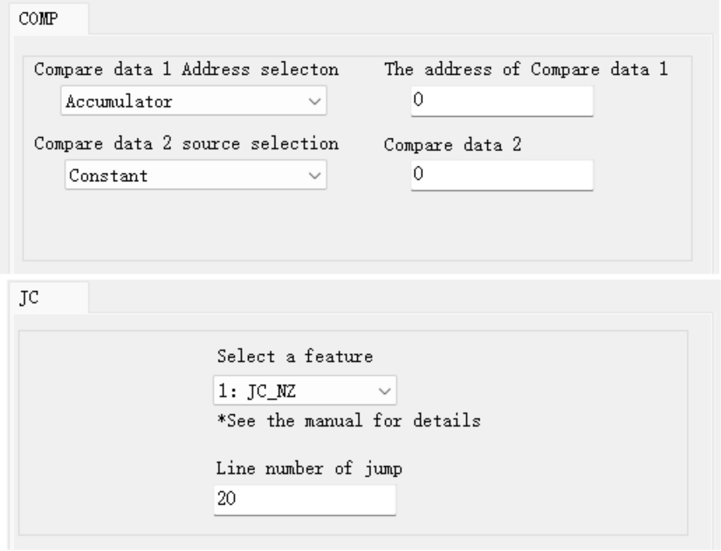
Select a feature: 0: JC_ZE

*See the manual for details

Line number of jump: 23

b) JC_NZ : It is not equal to 0

If the data in the accumulator is not equal to 0, jump to line 20



COMP

Compare data 1 Address selecton: Accumulator

The address of Compare data 1: 0

Compare data 2 source selection: Constant

Compare data 2: 0

JC

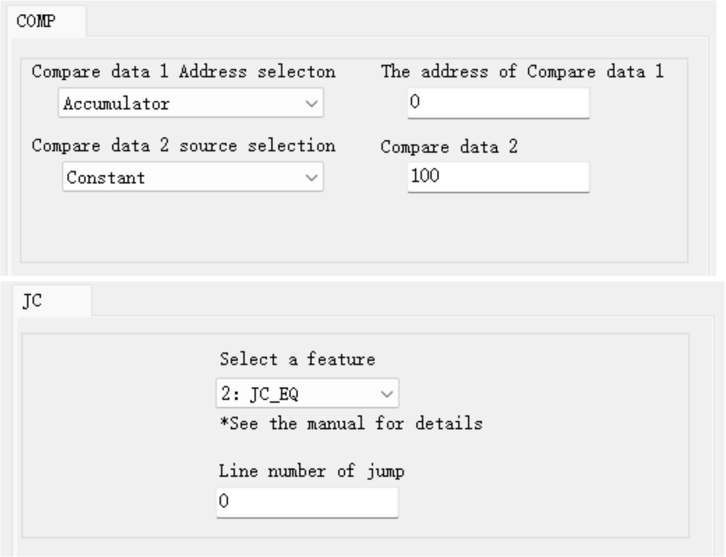
Select a feature: 1: JC_NZ

*See the manual for details

Line number of jump: 20

c) JC_EQ : equal

If the data in the accumulator 0 is equal to 100, jump to line 0



COMP

Compare data 1 Address selecton: Accumulator

The address of Compare data 1: 0

Compare data 2 source selection: Constant

Compare data 2: 100

JC

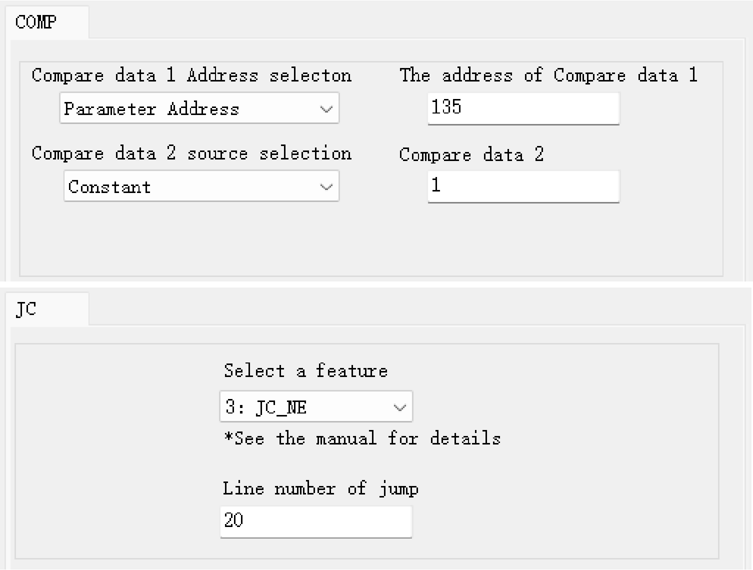
Select a feature: 2: JC_EQ

*See the manual for details

Line number of jump: 0

d) JC_NE : Not equal

If the data of the parameter address 135 is not equal to 1, jump to line 20



COMP

Compare data 1 Address selecton: Parameter Address

The address of Compare data 1: 135

Compare data 2 source selection: Constant

Compare data 2: 1

JC

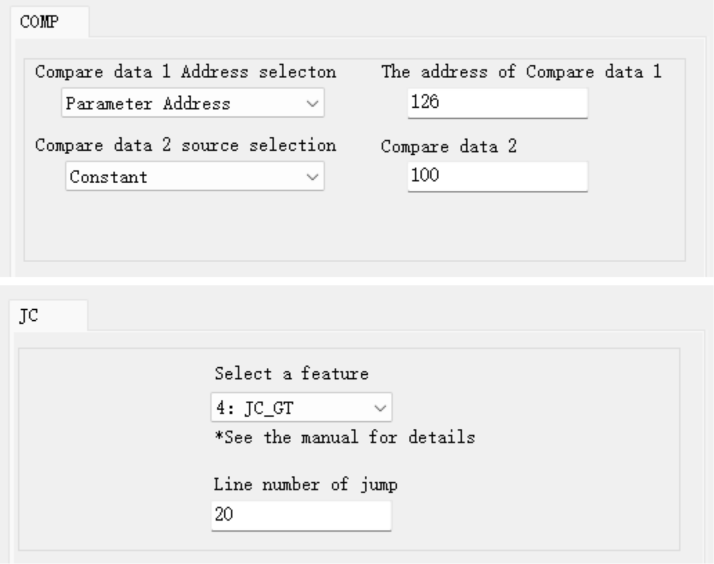
Select a feature: 3: JC_NE

*See the manual for details

Line number of jump: 20

e) JC_GT : Greater than

If the data of the parameter address 126 is greater than 100, jump to line 20



COMP

Compare data 1 Address selecton: Parameter Address

The address of Compare data 1: 126

Compare data 2 source selection: Constant

Compare data 2: 100

JC

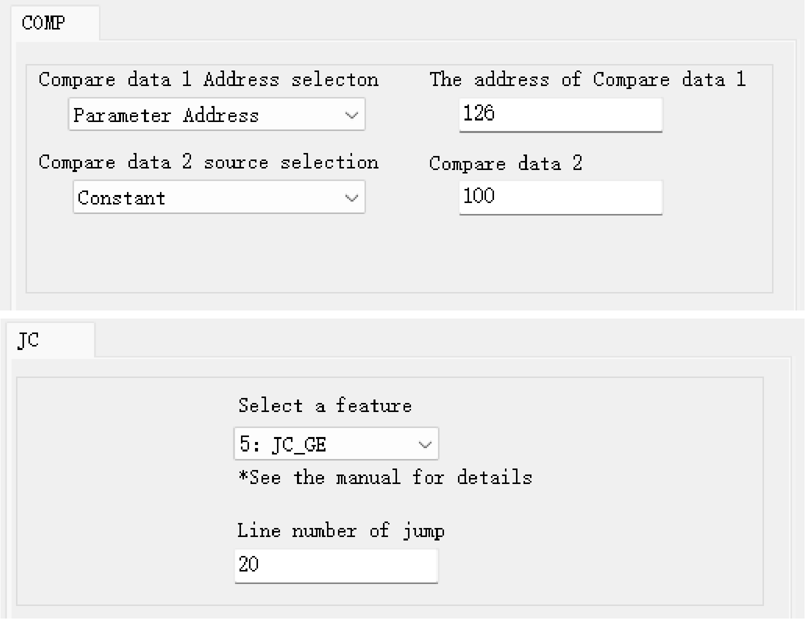
Select a feature: 4: JC_GT

*See the manual for details

Line number of jump: 20

f) JC_GE : Greater than or equal to

If the data of the parameter address 126 is greater than or equal to 100, jump to line 20



COMP

Compare data 1 Address selecton: Parameter Address

The address of Compare data 1: 126

Compare data 2 source selection: Constant

Compare data 2: 100

JC

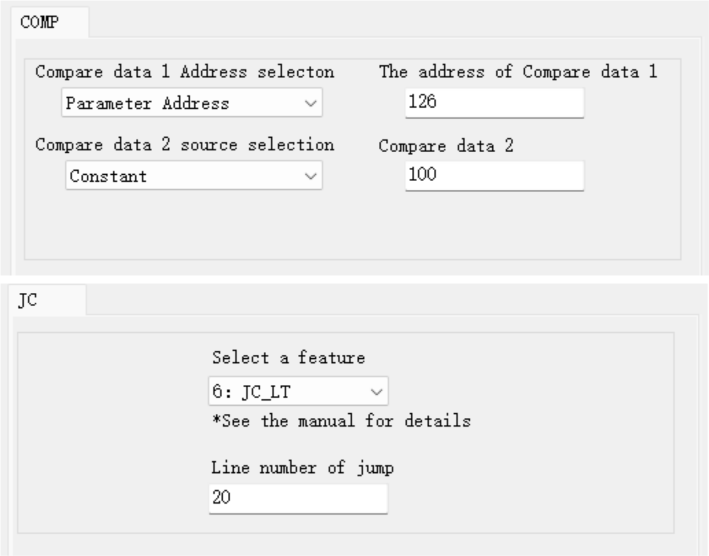
Select a feature: 5: JC_GE

*See the manual for details

Line number of jump: 20

g) JC_LT : Less than

If the data of the parameter address 126 is less than 100, jump to line 20



COMP

Compare data 1 Address selecton: Parameter Address

The address of Compare data 1: 126

Compare data 2 source selection: Constant

Compare data 2: 100

JC

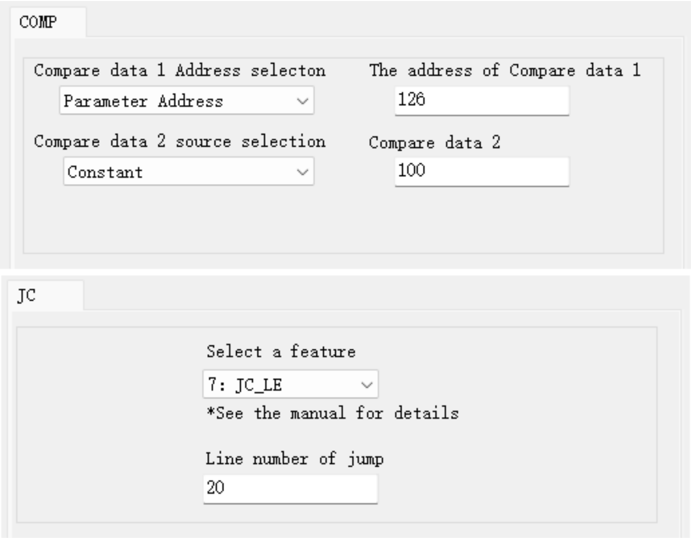
Select a feature: 6: JC_LT

*See the manual for details

Line number of jump: 20

h) JC_LE : Less than or equal to

If the data of the parameter address 126 is less than or equal to 100, jump to line 20



COMP

Compare data 1 Address selecton: Parameter Address

The address of Compare data 1: 126

Compare data 2 source selection: Constant

Compare data 2: 100

JC

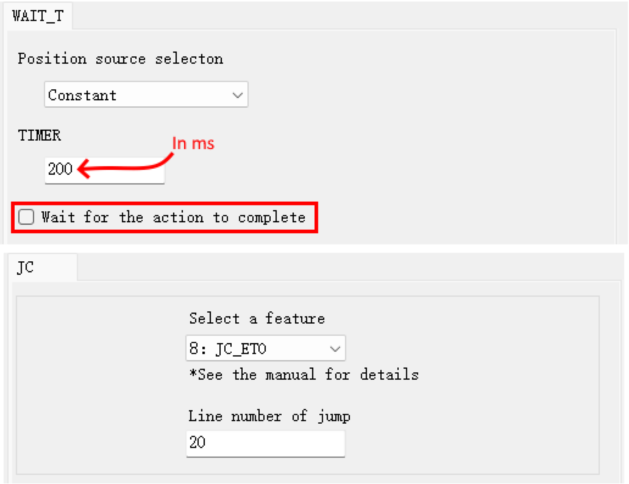
Select a feature: 7: JC_LE

*See the manual for details

Line number of jump: 20

i) JC_ETO : Timeout

If timeout (200ms) go to line 20



WAIT_T

Position source selecton

Constant

TIMER

200

In ms

☐ Wait for the action to complete

JC

Select a feature

8: JC_ETO

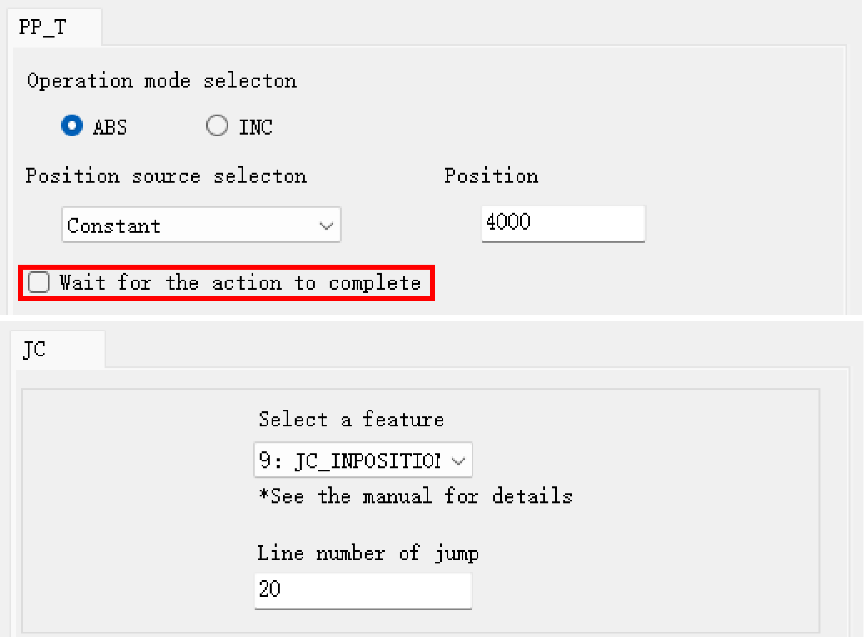
*See the manual for details

Line number of jump

20

j) JC_INPOSITION : Wait motor reach position

If the motor reach the position, jump to line 20



PP_T

Operation mode selecton

● ABS ○ INC

Position source selecton

Constant

Position

4000

☐ Wait for the action to complete

JC

Select a feature

9: JC_INPOSITIOI

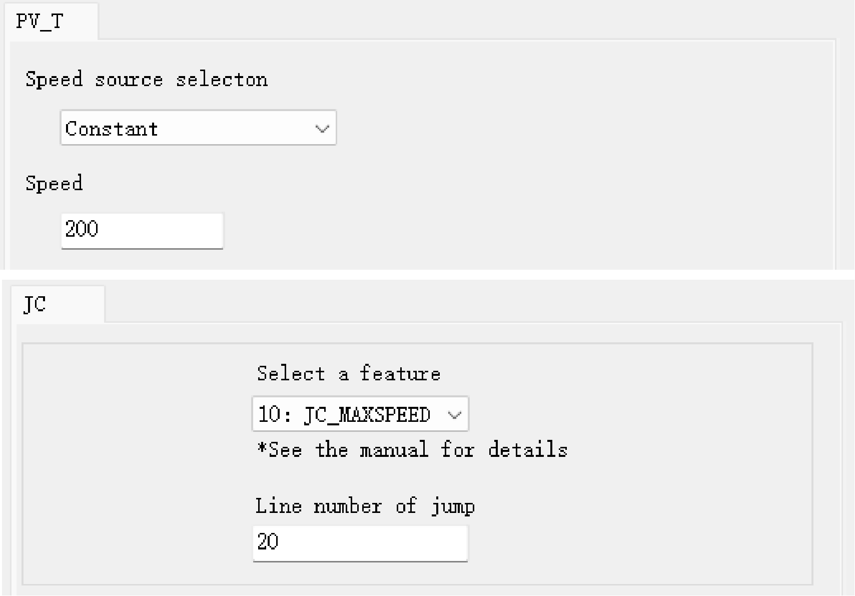
*See the manual for details

Line number of jump

20

k) JC_MAXSPEED : Wait for the motor reach his speed

If the motor reach his target speed, jump to line 20



PV_T

Speed source selecton

Constant

Speed

200

JC

Select a feature

10: JC_MAXSPEED

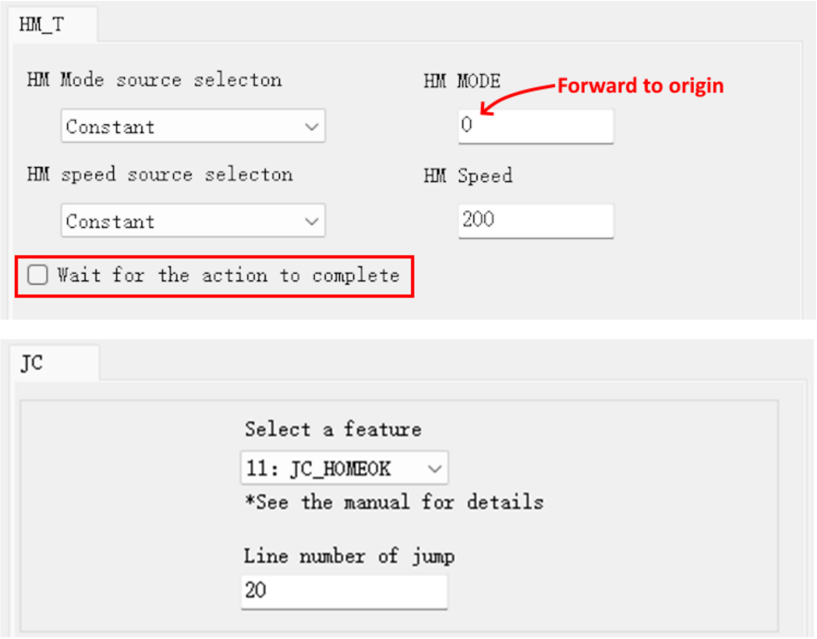
*See the manual for details

Line number of jump

20

l) JC_HOMEOK : The homing action is successful

If the homing action is successful, jump to line 20



HM_T

HM Mode source selecton

Constant

HM MODE

0

Forward to origin

HM speed source selecton

Constant

HM Speed

200

☐ Wait for the action to complete

JC

Select a feature

11: JC_HOMEOK

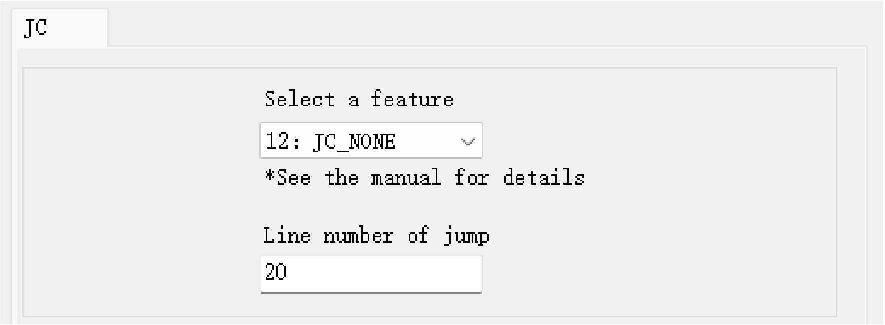
*See the manual for details

Line number of jump

20

m) JC_NONE : Unconditional jump

Jump to line 20



Select a feature

12: JC_NONE

*See the manual for details

Line number of jump

20

➢ CALC : Calculations

The data in the accumulator is calculated and processed according to the rules, and the running results are put back into the accumulator

Liste of the operations available below:

#	Name	Operation
0	CALC_ADD	Add
1	CALC_SUB	Minus
2	CALC_MUL	Multiply
3	CALC_DIV	Divide
4	CALC_MOD	Modulo
5	CALC_AND	AND
6	CALC_OR	OR
7	CALC_XOR	XOR
8	CALC_NOT	bitwise NOT

Table 21 : CALC function features

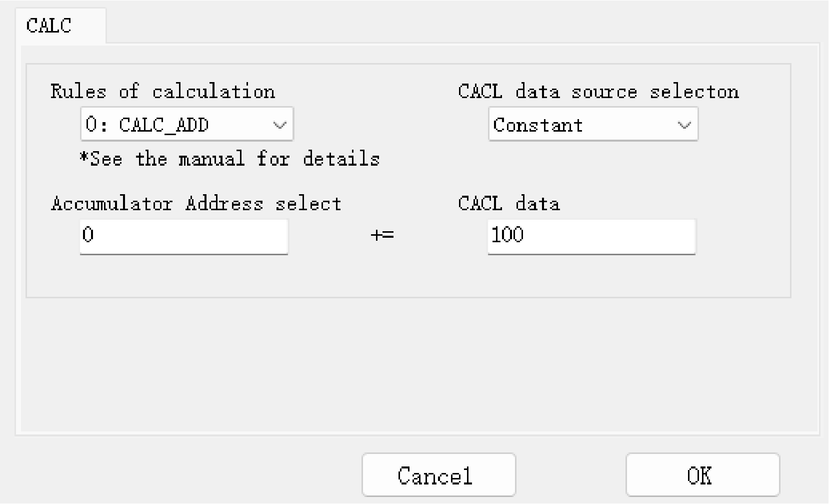
The calculation data source processes the data in the address of the accumulator according to the operation rules, and the calculation result is put into the accumulator

Accumulator = Accumulator < calculation rule> computing data source

The computing data source can be : constant, accumulator or parameter address

a) Data addition operations in the accumulator

Add 100 to the data in the accumulator 0



CALC

Rules of calculation: 0: CALC_ADD

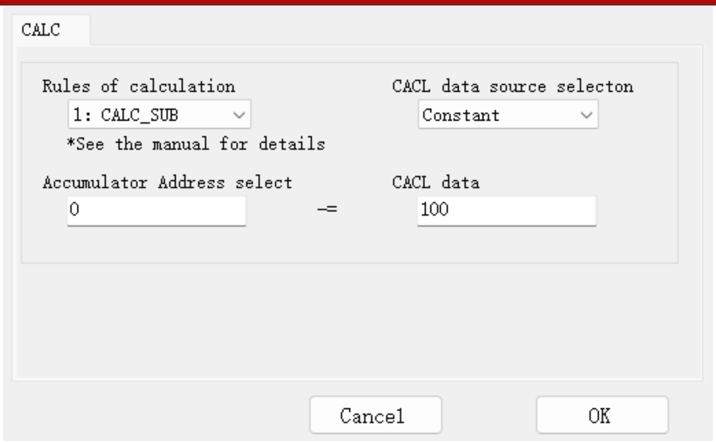
*See the manual for details

CACL data source selecton: Constant

Accumulator Address select: 0 += CACL data: 100

b) Subtract the data in the accumulator

The data in the accumulator 0 is minus 100



CALC

Rules of calculation: 1: CALC_SUB

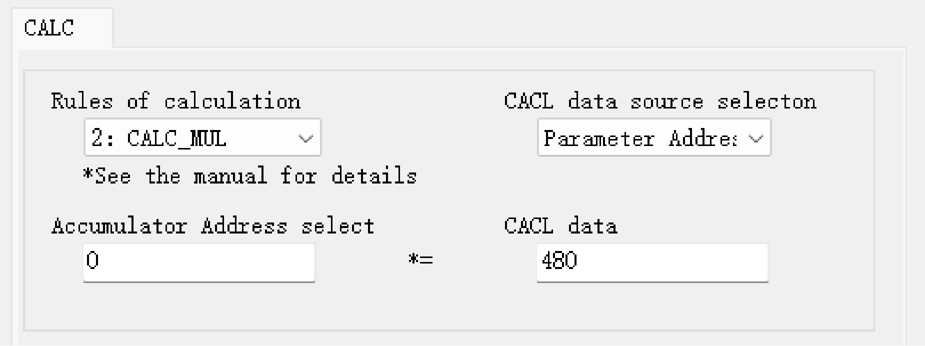
*See the manual for details

CACL data source selecton: Constant

Accumulator Address select: 0 -= CACL data: 100

c) Multiplication of data within an accumulator by parameter address

Multiply the data in the accumulator 0 by the parameter address 480



CALC

Rules of calculation
2: CALC_MUL
*See the manual for details

CACL data source selecton
Parameter Addre:

Accumulator Address select
0

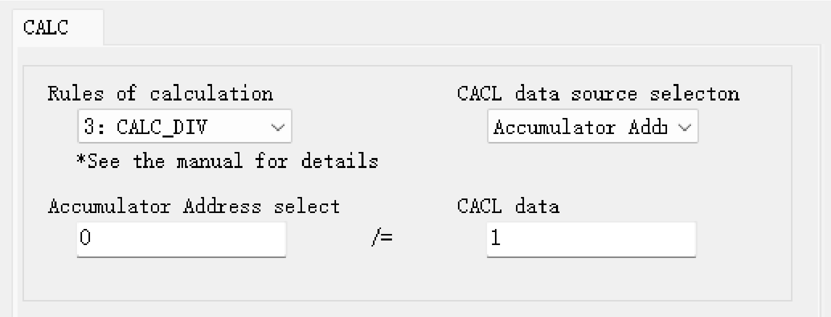
*=

CACL data
480

Example : If ACC[0]=10 and GV1=5 (parameter address 480)
The result of the CALC function is : ACC[0] = 50 and GV1=5

d) Data division operations within the accumulator by another accumulator

The data in accumulator 0 is divided by the data in accumulator 1



CALC

Rules of calculation
3: CALC_DIV
*See the manual for details

CACL data source selecton
Accumulator Addı

Accumulator Address select
0

/=

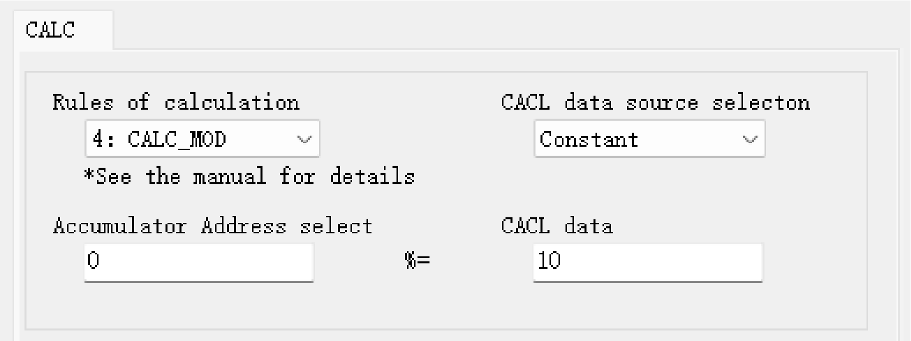
CACL data
1

Example : If ACC[0] = 10 and ACC[1]=6

The result of the CALC function is : ACC[0]=1 and ACC[1]=6

e) Data modulo operation in the accumulator

The data in the accumulator 0 is modulo by 10



CALC

Rules of calculation: 4: CALC_MOD

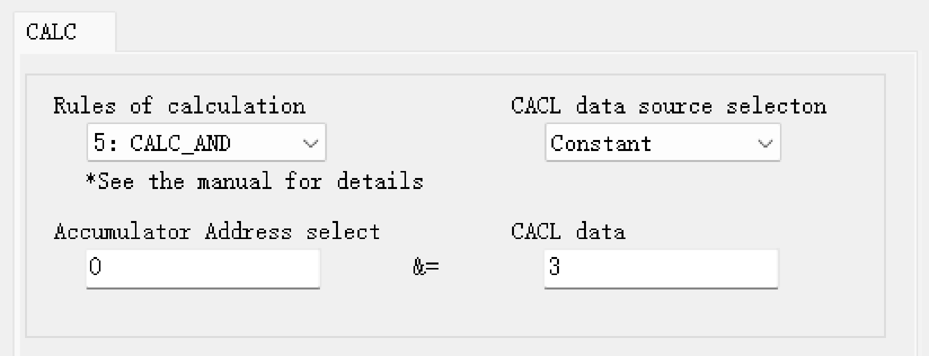
*See the manual for details

CACL data source selecton: Constant

Accumulator Address select: 0 %= CACL data: 10

f) AND operations within an accumulator

The data in accumulator 0 is AND 3



CALC

Rules of calculation: 5: CALC_AND

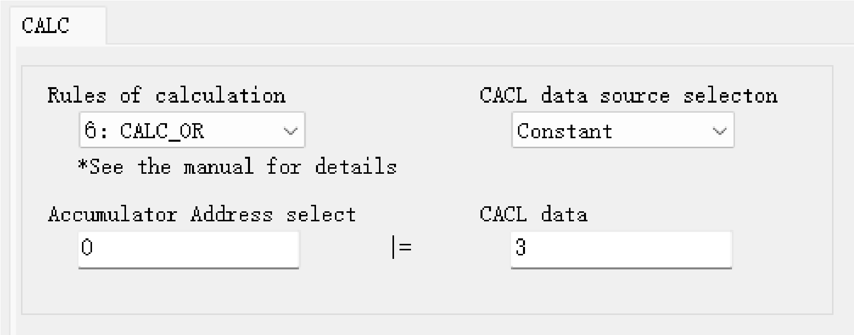
*See the manual for details

CACL data source selecton: Constant

Accumulator Address select: 0 &= CACL data: 3

g) OR operations within an accumulator

The data in the accumulator 0 is OR 3



CALC

Rules of calculation: 6: CALC_OR

*See the manual for details

CACL data source selecton: Constant

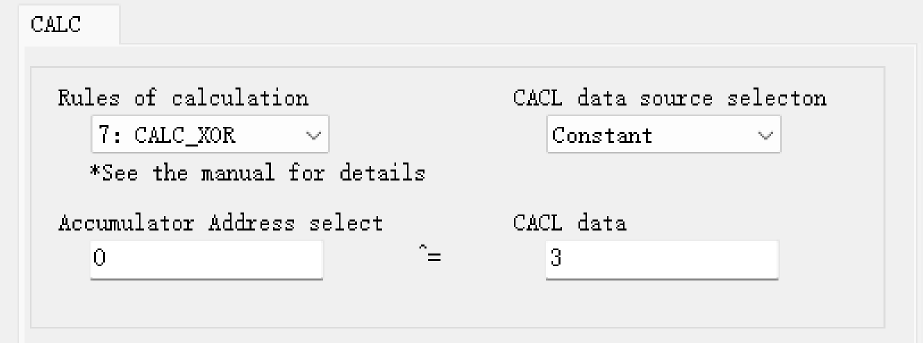
Accumulator Address select: 0

|=

CACL data: 3

h) XOR operations within an accumulator

The data in the accumulator 0 is XOR 3



CALC

Rules of calculation: 7: CALC_XOR

*See the manual for details

CACL data source selecton: Constant

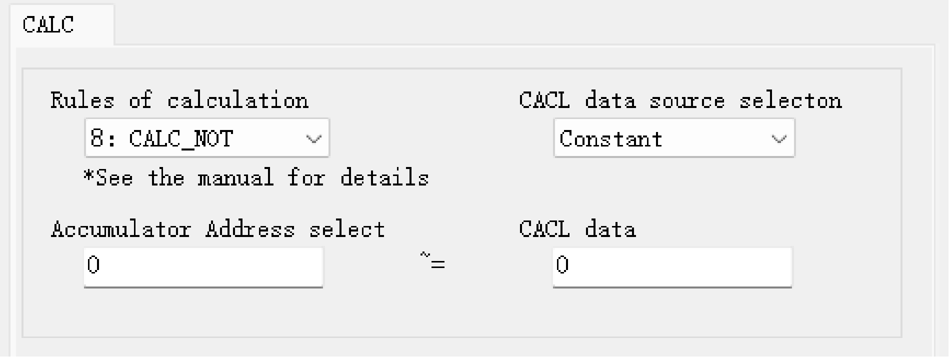
Accumulator Address select: 0

^=

CACL data: 3

i) The data in the accumulator is bitwise NOT

The data in accumulator 0 is bitwise NOT



Rules of calculation: 8: CALC_NOT

*See the manual for details

CACL data source selecton: Constant

Accumulator Address select: 0

~=

CACL data: 0

Example: If ACC[0] = 10 (00001010)b ,

The result of the CALC function is : ACC[0]=-11 (11110101)b

PP_T : Set position target

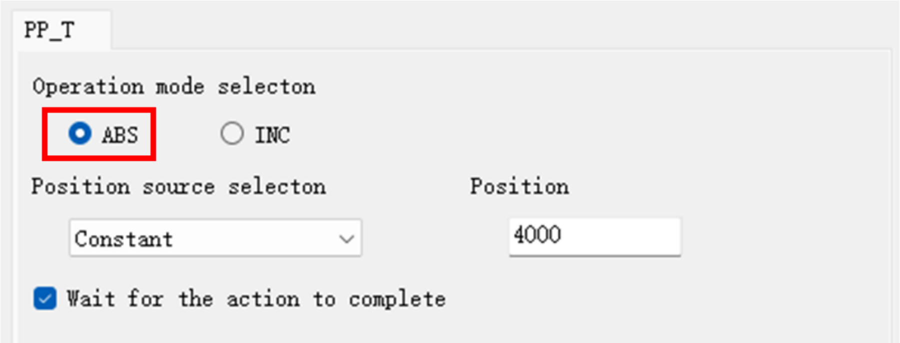
Run to the target location according to the setting.

The position can be set by : constant, accumulator or parameter address.

The case “Wait for the action to complete” permit to wait the motor reach the position before go to the next line, else use JC function.

a) Absolute operation

Move to the absolute position 4000



PP_T

Operation mode selecton

● ABS ○ INC

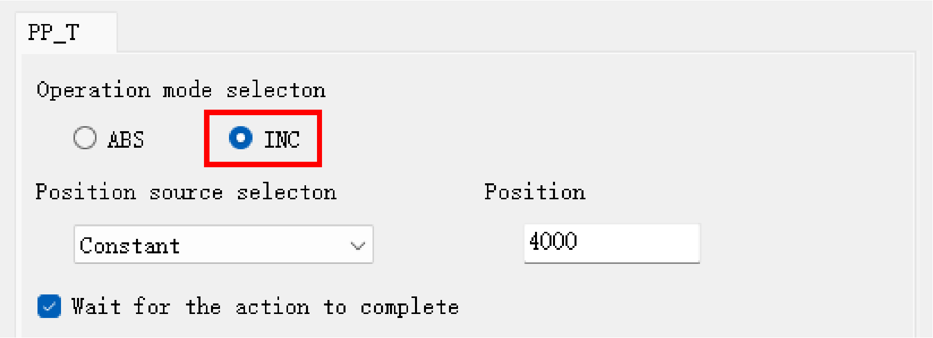
Position source selecton: Constant

Position: 4000

☑ Wait for the action to complete

b) Relative operation

Move to the relative position 4000



PP_T

Operation mode selecton

○ ABS ● INC

Position source selecton: Constant

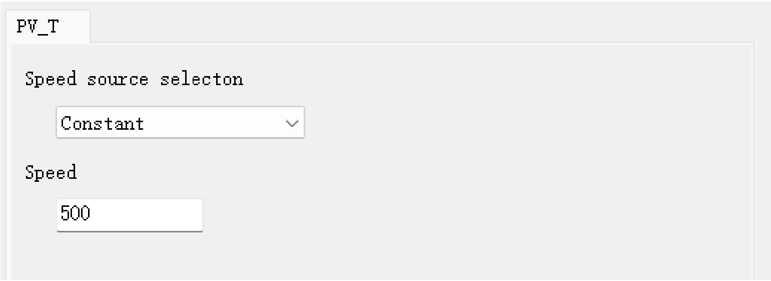
Position: 4000

☑ Wait for the action to complete

➢ PV_T : Set speed target

Run according to the set speed. After the speed command is executed, the next line will be automatically run without waiting to reach the target speed. And if you want to end the action, you need to use the function stop.

Move at a fixed speed of 5rps



PV_T

Speed source selecton

Constant

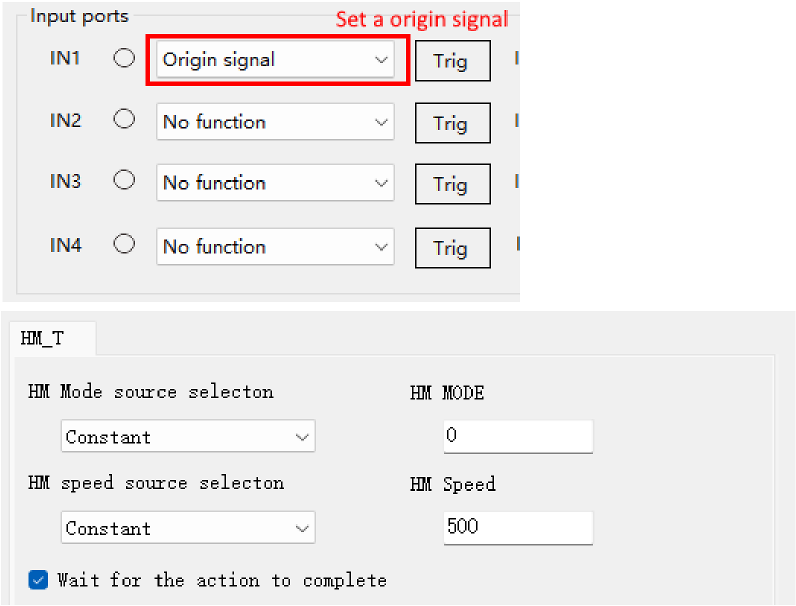
Speed

500

➢ HM_T : Homing method

Run according to the homing mode and speed. homing method and speed can be set by constant, accumulator or parameter address.

Forward rotation to the origin (homing method 0), 5rps speed



Input ports

Set a origin signal

IN1	○	Origin signal	Trig
IN2	○	No function	Trig
IN3	○	No function	Trig
IN4	○	No function	Trig

HM_T

HM Mode source selecton: Constant

HM MODE: 0

HM speed source selecton: Constant

HM Speed: 500

☑ Wait for the action to complete

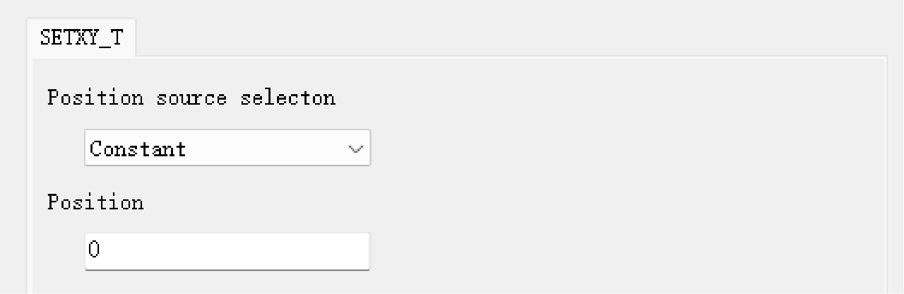
➢ Stop

The stop function will stop the movement according to the deceleration parameter, jump to next line when the motor are completely stop.

➢ SETXY_T : Set current location

Set the current location. The position can be set by constant, accumulator or parameter address.

Set the current location to 0



SETXY_T

Position source selecton

Constant

Position

0

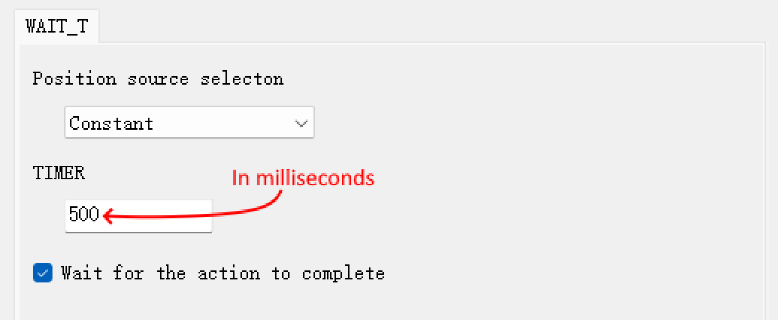
WAIT_T : Wait

If you want to wait, you can select the waiting method. The time can be set by constant, accumulator or parameter address.

The time is in milliseconds.

The case “Wait for the action to complete” permit to wait the time before go to the next line, else use JC function.

Wait 500ms and go to next line



Position source selecton

Constant

TIMER

500

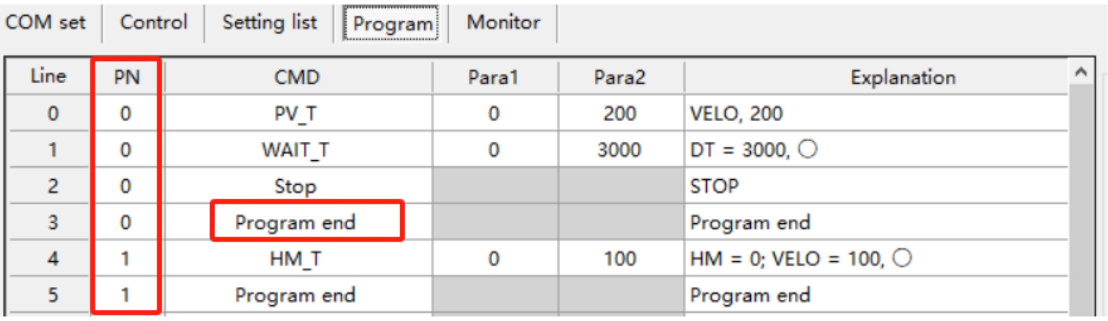
In milliseconds

☑ Wait for the action to complete

Program_end

A paragraph end flag must be set at the end of each programming section. After setting the paragraph end flag, the paragraph number will automatically increment by 1 when programming the next line.

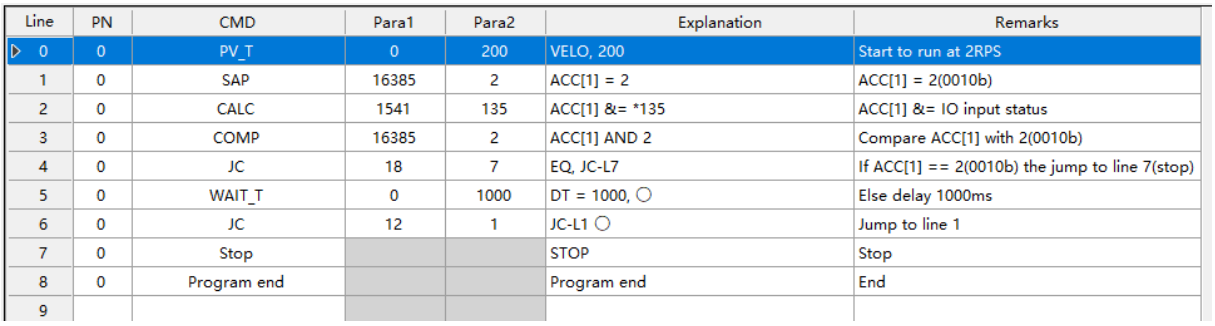
Paragraph End



Line	PN	CMD	Para1	Para2	Explanation
0	0	PV_T	0	200	VELO, 200
1	0	WAIT_T	0	3000	DT = 3000, ○
2	0	Stop			STOP
3	0	Program end			Program end
4	1	HM_T	0	100	HM = 0; VELO = 100, ○
5	1	Program end			Program end

- ## Code example

▪ Input condition program

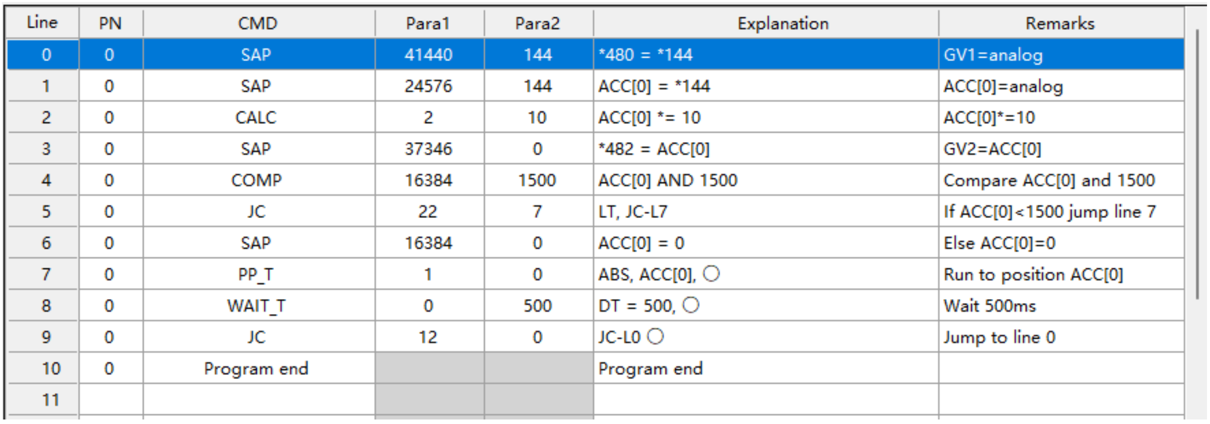


Line	PN	CMD	Para1	Para2	Explanation	Remarks
▷ 0	0	PV_T	0	200	VELO, 200	Start to run at 2RPS
1	0	SAP	16385	2	ACC[1] = 2	ACC[1] = 2(0010b)
2	0	CALC	1541	135	ACC[1] &= *135	ACC[1] &= IO input status
3	0	COMP	16385	2	ACC[1] AND 2	Compare ACC[1] with 2(0010b)
4	0	JC	18	7	EQ, JC-L7	If ACC[1] == 2(0010b) the jump to line 7(stop)
5	0	WAIT_T	0	1000	DT = 1000, ○	Else delay 1000ms
6	0	JC	12	1	JC-L1 ○	Jump to line 1
7	0	Stop			STOP	Stop
8	0	Program end			Program end	End
9						

Figure 15 : Input condition program example

The program start the running the motor at 2rps, compare with a mask the input port and 2 to know if IN2 is activate. If IN2 is activated, the movement is stop else the movement continue until the IN2 is activated.

▪ Analog input program



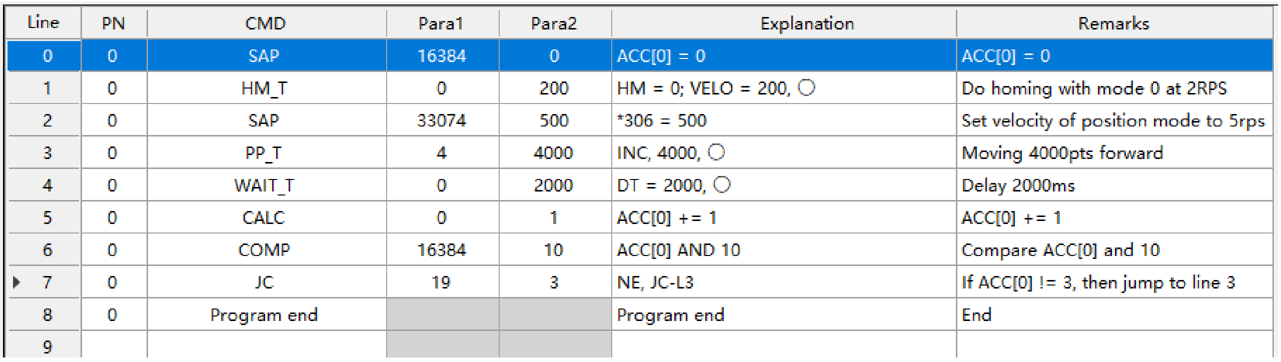
Line	PN	CMD	Para1	Para2	Explanation	Remarks
0	0	SAP	41440	144	*480 = *144	GV1=analog
1	0	SAP	24576	144	ACC[0] = *144	ACC[0]=analog
2	0	CALC	2	10	ACC[0] *= 10	ACC[0]*=10
3	0	SAP	37346	0	*482 = ACC[0]	GV2=ACC[0]
4	0	COMP	16384	1500	ACC[0] AND 1500	Compare ACC[0] and 1500
5	0	JC	22	7	LT, JC-L7	If ACC[0]<1500 jump line 7
6	0	SAP	16384	0	ACC[0] = 0	Else ACC[0]=0
7	0	PP_T	1	0	ABS, ACC[0], ○	Run to position ACC[0]
8	0	WAIT_T	0	500	DT = 500, ○	Wait 500ms
9	0	JC	12	0	JC-L0 ○	Jump to line 0
10	0	Program end			Program end	
11						

Figure 16 : Analog input program example

The program will set global variable 1 and accumulator 0 to the analog value. Multiply the analog value by 10. After run at target position that is 1000pt/V or 0 if analog value is outrange. After 500ms restart instruction.

For example if there are a tension of 2.5V on analog, the motor will move to position 25000.

- **Homing and for loop**



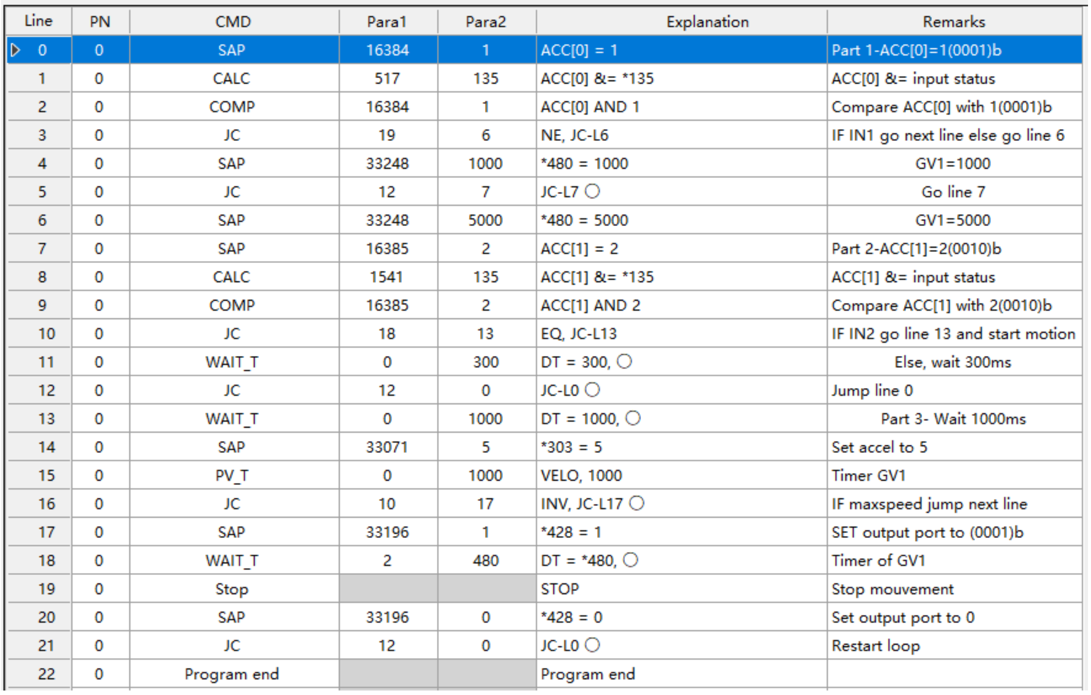
Line	PN	CMD	Para1	Para2	Explanation	Remarks
0	0	SAP	16384	0	ACC[0] = 0	ACC[0] = 0
1	0	HM_T	0	200	HM = 0; VELO = 200, ○	Do homing with mode 0 at 2RPS
2	0	SAP	33074	500	*306 = 500	Set velocity of position mode to 5rps
3	0	PP_T	4	4000	INC, 4000, ○	Moving 4000pts forward
4	0	WAIT_T	0	2000	DT = 2000, ○	Delay 2000ms
5	0	CALC	0	1	ACC[0] += 1	ACC[0] += 1
6	0	COMP	16384	10	ACC[0] AND 10	Compare ACC[0] and 10
▸ 7	0	JC	19	3	NE, JC-L3	If ACC[0] != 3, then jump to line 3
8	0	Program end			Program end	End
9						

Figure 17 : Homing and for loop program example

The program will start the homing method 0 (forward to origin) at 2 rps, an origin signal input must be assign to achieve the homing.

After the position mode velocity is set to 5rps, the motor will move forward 10 times of 4000 points.

- **Input, Output and speed mode**



Line	PN	CMD	Para1	Para2	Explanation	Remarks
▷ 0	0	SAP	16384	1	ACC[0] = 1	Part 1-ACC[0]=1(0001)b
1	0	CALC	517	135	ACC[0] &= *135	ACC[0] &= input status
2	0	COMP	16384	1	ACC[0] AND 1	Compare ACC[0] with 1(0001)b
3	0	JC	19	6	NE, JC-L6	IF IN1 go next line else go line 6
4	0	SAP	33248	1000	*480 = 1000	GV1=1000
5	0	JC	12	7	JC-L7 ○	Go line 7
6	0	SAP	33248	5000	*480 = 5000	GV1=5000
7	0	SAP	16385	2	ACC[1] = 2	Part 2-ACC[1]=2(0010)b
8	0	CALC	1541	135	ACC[1] &= *135	ACC[1] &= input status
9	0	COMP	16385	2	ACC[1] AND 2	Compare ACC[1] with 2(0010)b
10	0	JC	18	13	EQ, JC-L13	IF IN2 go line 13 and start motion
11	0	WAIT_T	0	300	DT = 300, ○	Else, wait 300ms
12	0	JC	12	0	JC-L0 ○	Jump line 0
13	0	WAIT_T	0	1000	DT = 1000, ○	Part 3- Wait 1000ms
14	0	SAP	33071	5	*303 = 5	Set accel to 5
15	0	PV_T	0	1000	VELO, 1000	Timer GV1
16	0	JC	10	17	INV, JC-L17 ○	IF maxspeed jump next line
17	0	SAP	33196	1	*428 = 1	SET output port to (0001)b
18	0	WAIT_T	2	480	DT = *480, ○	Timer of GV1
19	0	Stop			STOP	Stop mouvement
20	0	SAP	33196	0	*428 = 0	Set output port to 0
21	0	JC	12	0	JC-L0 ○	Restart loop
22	0	Program end			Program end	

Figure 18 : Program example condition on Inputs state , set output and speed motion

In the program there is 3 parts:

- Check input state 1 and action (line 0 to 6) : If the input state 1 is active global variable 1 is set to 1000 else it set to 5000. After go to line 0.
- Check input state 2 and action (line 7 to 12) : If the input state 2 is activate, start the motion.
- Motion (line 13 to 21) : Set the acceleration to 5, set target speed to 10 rps. When the motor reach the target speed the OUT1 is set to 1, the motor will rotate during 1000ms or 5000ms (in function of IN2). When the motor is stopped the output port is set to 0.

4. Supplemental directives

• Copyright

A2V Mécatronique owns the content of this user manual in its entirety, including but not limited to pictures, logos, trademarks, and resources.

Redistribution of sources or derived formats (for example, Portable Document Format or Hypertext Markup Language) must retain the above copyright notice, and the complete data sheet, user manual, and documentation of this product including associated application notes; and a reference to other available product-related documentation.

• Trademark Designations and Symbols

Trademark designations and symbols used in this documentation indicate that a product or feature is owned and registered as trademark and/or patent either by A2V Mécatronique or by other manufacturers, whose products are used or referred to in combination with A2V Mécatronique products and A2V Mécatronique product documentation.

This user manual is a non-commercial publication that seeks to provide concise scientific and technical user information to the target user. Thus, trademark designations and symbols are only entered in the Short Spec of this document that introduces the product at a quick glance. The trademark designation /symbol is also entered when the product or feature name occurs for the first time in the document. All trademarks and brand names used are property of their respective owners.

• Target user

The documentation provided here, is for programmers and engineers only, who are equipped with the necessary skills and have been trained to work with this type of product.

The Target User knows how to responsibly make use of this product without causing harm to himself or others, and without causing damage to systems or devices, in which the user incorporates the product.

• Disclaimer: Life Support Systems

A2V Mécatronique does not authorize or warrant any of its products for use in life support systems, without the specific written consent of A2V Mécatronique.

Life support systems are equipment intended to support or sustain life, and whose failure to perform, when properly used in accordance with instructions provided, can be reasonably expected to result in personal injury or death.

Information given in this document is believed to be accurate and reliable. However, no responsibility is assumed for the consequences of its use nor for any infringement of patents or other rights of third parties which may result from its use. Specifications are subject to change without notice.

Disclaimer: Intended Use

The data specified in this user manual is intended solely for the purpose of product description. No representations or warranties, either express or implied, of merchantability, fitness for a particular purpose or of any other nature are made hereunder with respect to information/specification or the products to which information refers and no guarantee with respect to compliance to the intended use is given.

In particular, this also applies to the stated possible applications or areas of applications of the product. A2V Mécatronique products are not designed for and must not be used in connection with any applications where the failure of such products would reasonably be expected to result in significant personal injury or death (safety-Critical Applications) without A2V Mécatronique consent.

A2V Mécatronique products are not designed nor intended for use in military or aerospace applications or environments or in automotive applications unless specifically designated for such use by A2V Mécatronique. A2V Mécatronique conveys no patent, copyright, mask work right or other trade mark right to this product. A2V Mécatronique assumes no liability for any patent and/or other trade mark rights of a third party resulting from processing or handling of the product and/or any other use of the product.

5. Annexs

Glossary

RPS : Rotate per seconds (1 RPS = 60RPM)

ACC : Accumulator, temporary data storage for programming

GV : Global variable for programming

Figures index

- **Tables index**