GUI Software User Guide

Ver. 3.0

Table of Contents

Revision History

Version	Date	Description	Remarks
1.0	DEC., 2015	Initial release	--
1.1	APR., 2016	1st revision	--
2.0	APR., 2018	2nd revision	Major updates
3.0	OCT., 2018	3rd revision	Amended and added several new functions, including: 1. **Interface element**: Added I/O viewer function (1.3.1). 2. **Wizard**: ● Added flow-setting page in Wizard. ● How to add/delete new motor (Ch. 3). 3. Setting-**Motor Protection** ● Revised the diagram of motor continuous current (4.3.1). 4. Settings-**Disable Option**: ● Revised Disable Option intro (4.4.2). ● Added descriptions of dynamic brake (4.4.2.3). ● Added "Disable Reaction End Vel" function (4.4.2.3-(b)). ● Deleted Motor Stuck function. 5. Settings-**Feedback**: ● Amended explanation of Encoder/Index Position (4.5-(f)). ● Added "Invert encoder polarity" function (4.5.1-(b)). ● Added new functions (Parameter & Calibration Tab) under Sine/Cosine encoder (4.5.2; 4.5.2.1; 4.5.2.2). ● Added Calibration Steps (4.5.2.3). ● Implemented Tamagawa encoder (4.5.3). ● Added "Event Limit" function (4.5.4). 6. Settings-**Boot Sequence:** ● Added RIGHT route (4.8) and its explanations (4.8).

Version	Date	Description	Remarks
			• Added "Don't abort on fault" function (4.8 chart). 7. Settings-**Position Comparator:** • New layout (4.10). 8. **Tune (general)** • Added functions of "Zoom Mode / Zoom Reset / Save as txt / Show Prevalue" for Bode plot as well as Frequency response of Tune-Current, -Velocity, and -Position. • Revised "sample count" description in Bode plot part. • Added new graph functions as follows: ▪ Current: VdDmd graph (5.1.4-(h)). ▪ Velocity: IdFdb (5.3.6-(l)). ▪ Pos.: PosError (5.4.4-(h)). 9. Tune-**Phase** • Phase Find (5.2.2): ▪ Rearranged introduction. ▪ Added Hall Theta Table function (5.2.2.2) and explained how to obtain its value by Scope (5.2.2.2.1). ▪ Added "Phase Find—Absolute encoder" function (5.2.2.3). • Added "invert commutation polarity" function (5.2.3-(b)). • Added a new chapter: Manually Set Phase (5.2.4). 10. Tune-**Gain Switch** • New layouts (5.5). 11. **Trial Run** • New layout (6.2.2). • Added "Estimated Runtime" and "Dwell Time" functions (6.2.2-(d & l)). • Allow users to set motor's current position as point A/B for Absolute-Move commands (6.2.2-(i & h)). 12. **Homing:** • Added "homing error trigger fault event" function (7.1-(i)).

Version	Date	Description	Remarks
			• Added warning about the functions "Transit….P.P. mode" and "Move to new zero…." (7.1-(g & h)). 13. And other miscellaneous revisions.

Copyright Notice

Disclaimer

1. Information furnished by cpc is believed to be accurate and reliable. However, no responsibility is assumed by cpc for its use, nor for any infringements of patents or other rights of third parties which may result from its use.
   cpc doesn't grant any license under its patent rights, nor the rights of others.
2. In addition, cpc assumes no responsibility for any errors that may appear in this document and for any claims or damages arising from information contained in this document.
3. The product specified in this document has been developed, produced, tested and documented in accordance with the relevant standards. cpc is not responsible for damages, accidents, or injuries caused by any deviation from the configuration and installation described in this guide;
4. Furthermore, cpc is not responsible for the performance of new measurements or ensuring that regulatory requirements are met.
5. The product specified in this document is not assumed to be used in critical application including, but not limited to, medical equipment, transportation, aerospace and nuclear instruments, undersea equipment, power plant equipment, as well as disaster prevention and crime prevention equipment.
6. We reserve the right to modify our products, including its hardware and software design, in order to improve its design and/or performance. The information in this document is subject to change without notice and does not represent a commitment by cpc.
7. Specifications are subject to change without notice.
8. Performance specification beyond those specified by safety regulations are guaranteed by design and not subject to production test.
9. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.
10. cpc assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using cpc products.

Contact Us

Headquarters

Chieftek Precision Co., Ltd.

NO.3, Dali 1st Rd., Xinshi Dist., Southern Taiwan Science Park,

Tainan City. 741-45, Taiwan (R.O.C.) 3

TEL: +886-6-505-5858

FAX: +886-6-505-5959

Email : service@mail.chieftek.com

China

Chieftek Machinery Kunshan Co., Ltd. ()

No.1188, Hongqiao Rd, Kunshan, Jiangsu, P.R. China 1186

Tel : +86-512-55252831

Fax : +86-512-55252851

Email : cn.service@mail.chieftek.com

Europe

cpc Europa GmbH

Industriepark 314, D-78244 Gottmadingen, Germany

Tel : +49-7731-59130-38

Fax : +49-7731-59130-28

Email : info@cpc-europa.de

USA

Chieftek Precision USA Co., Ltd.

4881 Murietta Street. Chino, CA. 91710

TEL: +1-909-628-9300

FAX: +1-909-628-7171

Email : info@usa.chieftek.com

Chapter 1 Installation & Interface

1.1 Software

There are two software tools for adjusting the driver: The cpc Graphical User Interface (GUI, also UI) and the Boot Loader.

- **User Interface (UI)**: To set driver parameters and tune the motor.
- **Boot Loader**: To flash the firmware to the driver via RS232 communication port. See chapter 4.8.
  Note: Boot loader is included in the cpc GUI software.

Please download GUI from cpc official website:
http://www.chieftek.com/product-tc.asp

1.2 Installation

1.2.1 System Requirements

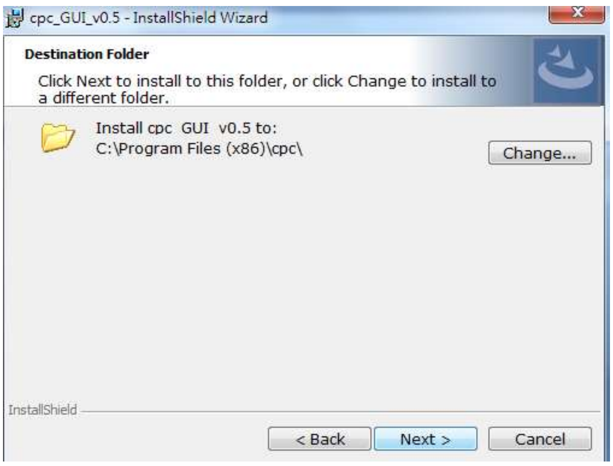
To execute software smoothly, your computer system should meet the following requirements:

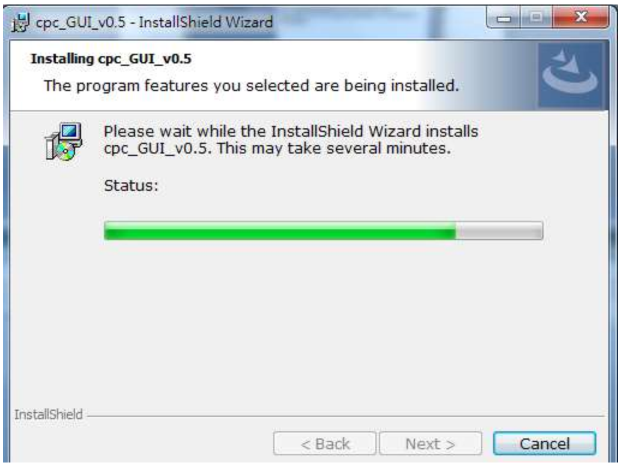
- Storage space: 500 MB
- Ram: 4 GB
- OS: windows-vista, win7 and win8/8.1 or later.
- Cable: RS232 to USB port.
- CD-ROM driver
- Framework V4.5 or later.

1.2.2 Steps

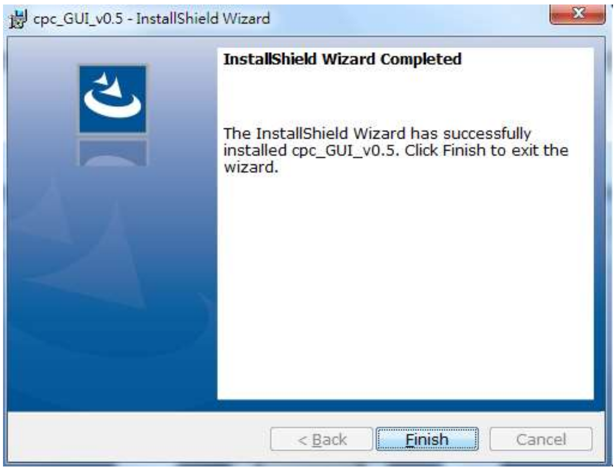
1. Download cpc UI software from cpc official website.
   http://www.chieftek.com/product-tc.asp
2. Framework V4.5 or later is necessary. Install Framework in host computer before installing GUI.
3. Click the file “setup.exe” to install UI.

4. Click “Next>”.



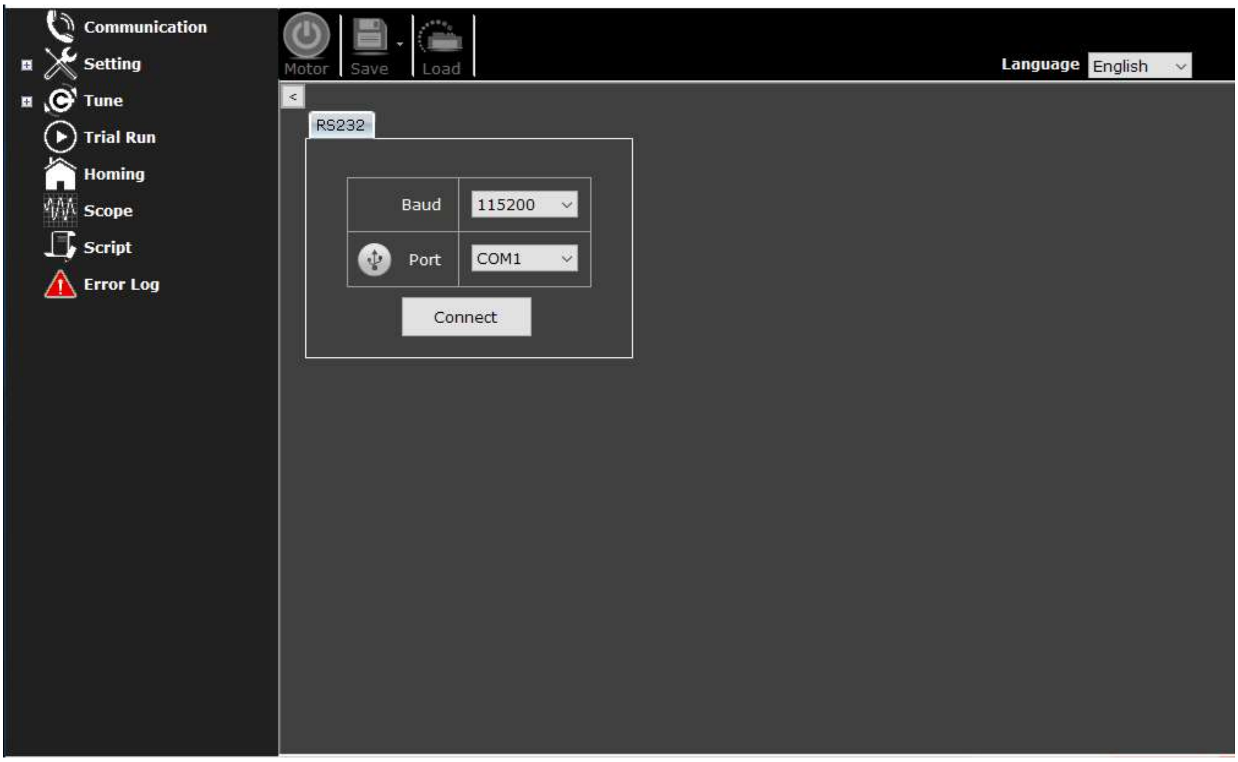


5. Click "Finish" to complete installation.



- Open the installed folder and double click the "GUI_TC1_ver_0_5" to access UI.

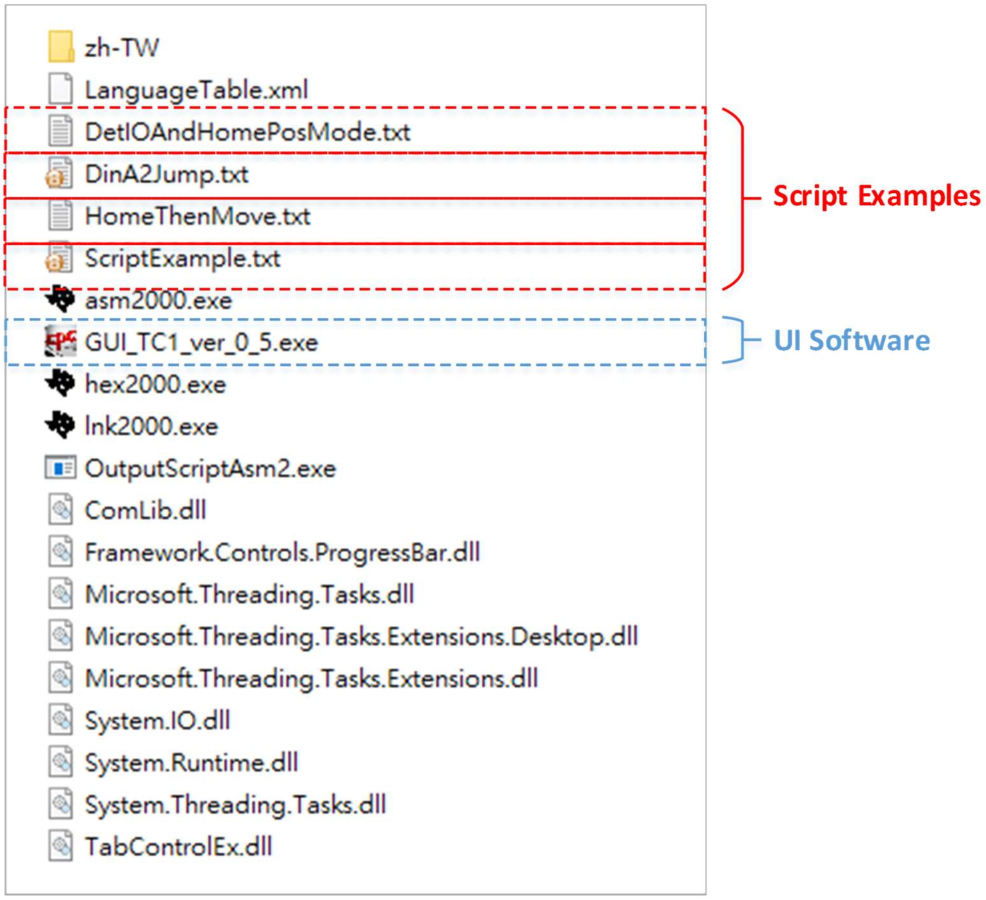
Software path for 64 bit system:	C:\Program Files (x86)\cpc
Software path for 32 bit system:	C:\Program Files\cpc



Note: If users can't open the UI software and see a message saying "lacking of dot Net Framework", please install Framework (v.4.5 or later).

1.2.3 Installed Files

There should be 4 text files in ".txt" format—which are the script examples—in the installed file folder in the host computer.

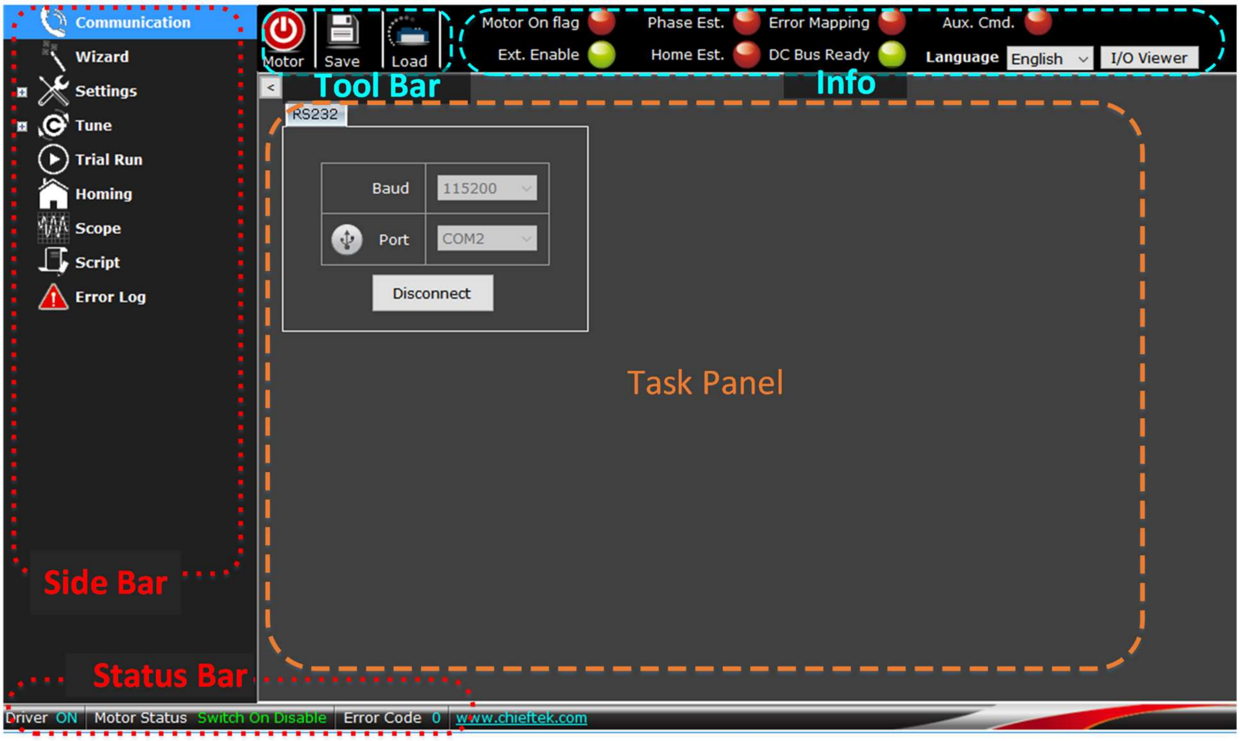


1.1 User Interface (UI)

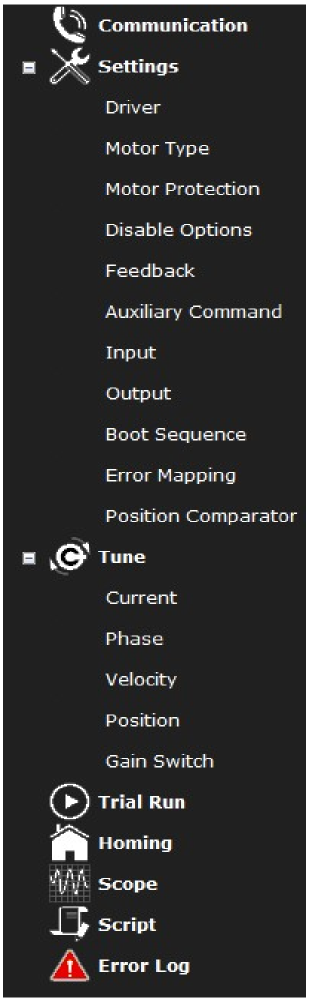
1.3.1 Interface Elements

The cpc UI has 5 function areas:

- Tool Bar
- Side Bar
- Task Panel
- Status Bar
- Info



Explanations:

Tool Bar	**Contains quick-access buttons for frequently used functions.**  **Motor:** The driver's internal **motor-off** switch. **Note**: It only switches the motor off, not on. **Save:** Saves configured parameters to Drive Flash or to host computer. **Load**: Loads user-defined parameters from the file at host computer.
Side Bar	Provides a menu to navigate the various function windows. 

	The sidebar can hide or show using the Arrow button.
Task Panel	**Allows users to set and test various parameters here and monitor corresponding values.**
Status Bar	**Shows the status of Driver, Motor, and Error Code (when fault occurs).** 
Info	**Contains indicating signals and UI language options to assist monitoring during use.**  **Motor On flag**: Shows green when driver's **internal** states meet the motor-on conditions. **Phase Est.**: Phase Established; shows green when phase-tuning is successfully completed. **Error Mapping**: Shows green when Error Mapping function is activated. **Aux. Cmd.**: Auxiliary Command; shows green when Auxiliary Command is enabled (see side bar). **Ext. Enable**: External Enable; shows green when ALL digital inputs defined as "External Enable Signal" are active. **If no digital input is defined**, the external enable is active by default. **Home Est.**: Home Established; shows green when Homing is completed successfully. **DC Bus Ready**: Driver's high voltage DC Bus supply is ready for operation. **Language**: Switches languages among English, Chinese (Simplified), Chinese (Traditional), German, and Japanese.

	I/O Viewer: Shows a popup window to indicate the status of Input and Output pins.

1.3.2 How to Input Parameter Values

1. To input data:
   - When configuring, key in values into the column and then press "Enter" to input value.
   - If you didn't press Enter, the data will not be sent to the driver; in this case the last valid value will be used.

2. Data validation:
   - If the keyed in value is **invalid**:

     The column becomes red 

     ; if you still press Enter, it will **show the last valid value.**
   - If the keyed in value is **valid**:

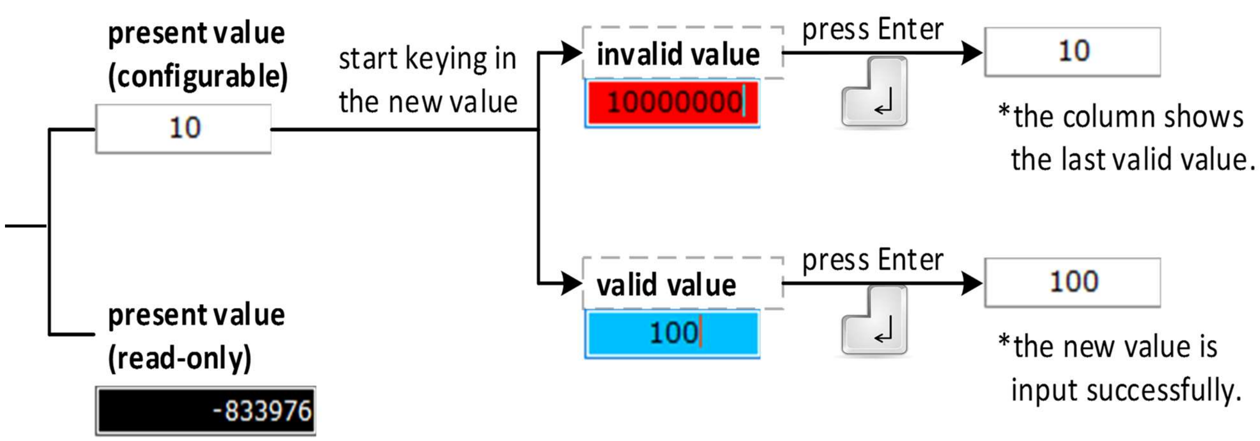
     The column becomes blue 

     ; after you press Enter, it reverts to white 

     .
   - Black columns 

     :

     Shows **read-only** parameters. Their values cannot be modified.



1.4 Boot Loader—Update Firmware

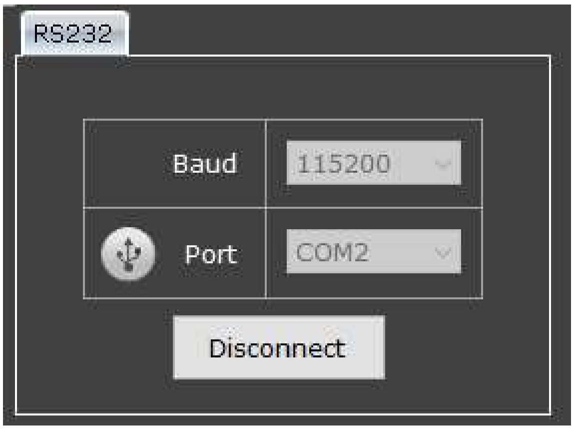
The "Boot-Loader" tool helps you update the driver firmware to the latest version through RS232 port. As long as a firmware xml file is available, users can update driver firmware.

Steps:

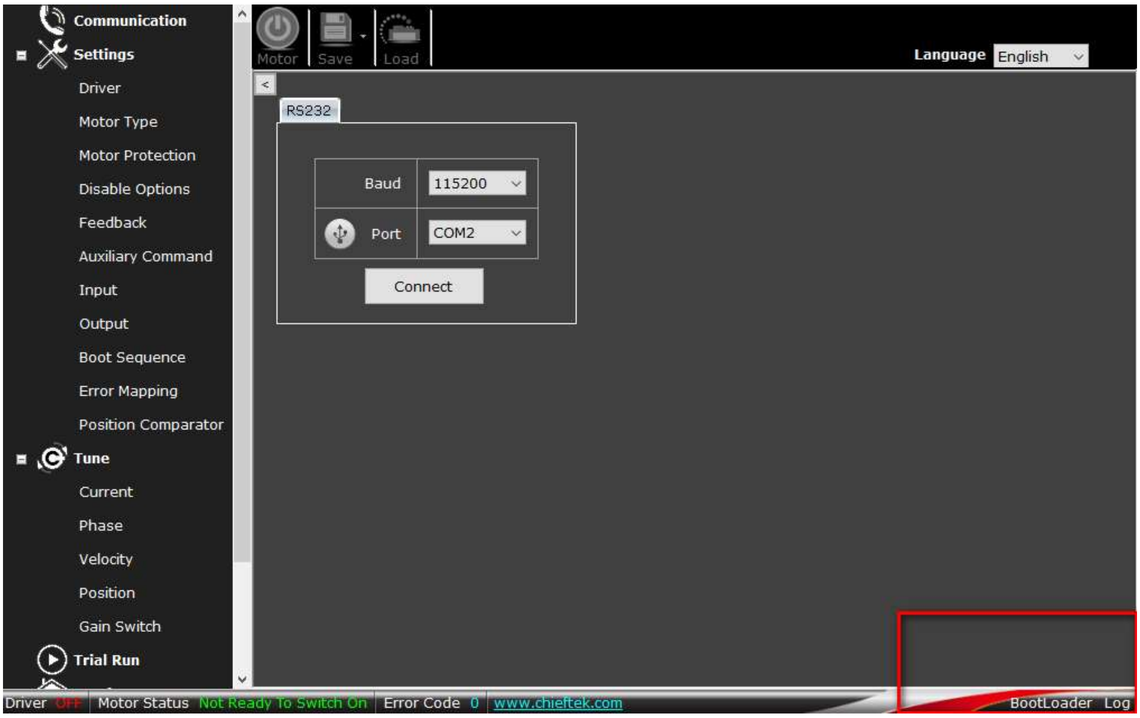
1. Download the latest firmware from the cpc official website. Choose the correct model name of the firmware you need.
   http://www.chieftek.com/product-tc.asp

Type	Description	
catalog	TC series (AC Linear Motor Servo Driver)	Download
DM	TC series - TC1&TC1-B	Download
manual	cpc GUI software user guide	Download
software	cpc GUI 0.4.48	Download
firmware	TC1-8_230 HW0.4 FW0.7.9	Download
	TC1-20_230 HW0.2 FW0.7.13	Download

2. Unzip the downloaded file.
3. Go back to the UI. Disconnect the driver (side bar > communication > click "Disconnect").



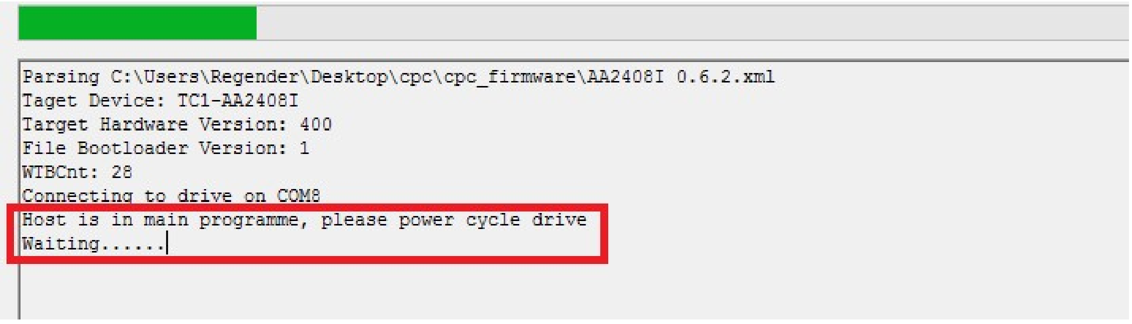
4. After disconnection, click on the bottom right corner of UI window to access the boot loader.

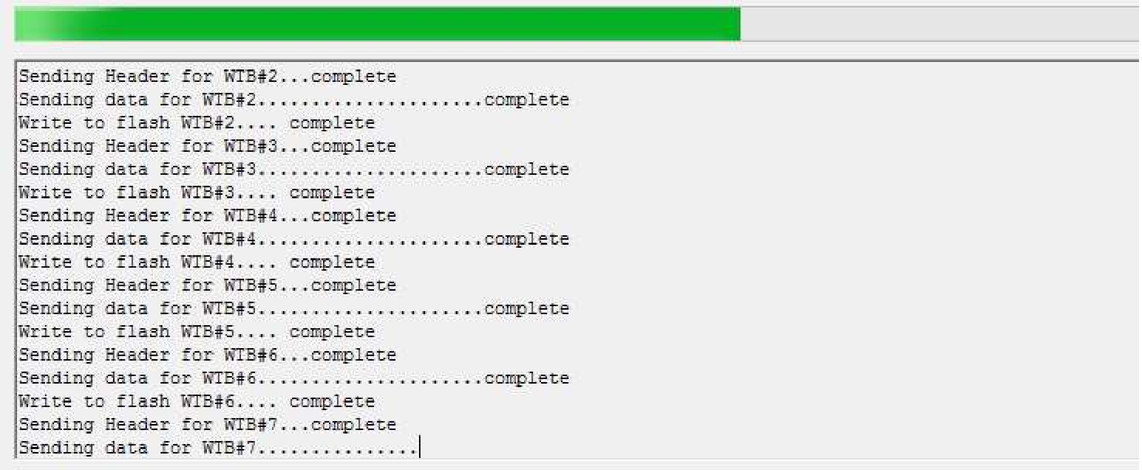


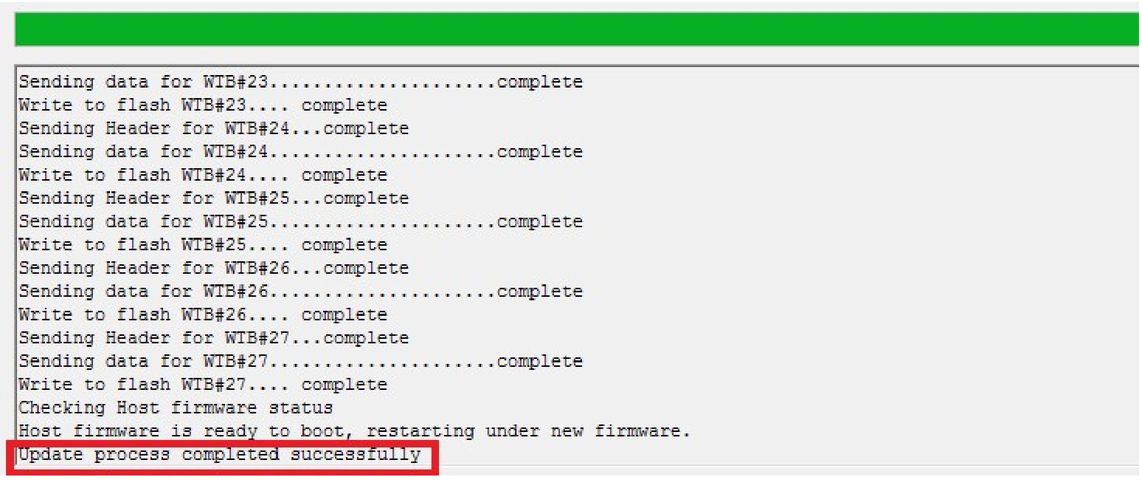
5. The Boot Loader interface will jump out automatically.
   Now, make sure again your communication cable (for example: USB-RS232 cable) is connected with the driver and host computer.

   Next,
   (1) Click "Scan Ports" button.
   (2) Choose serial port number.
   (3) Click on "Choose" and locate the firmware file (it is the downloaded file which is in ".xml" format. For example: TC1-8_230 HW0.5 FW0.7.14.xml).
   (4) Click "Start" button.

6. When bootloader shows the message “Host is in main programme, please power cycle drive Waiting……”, turn off the driver and turn on again.



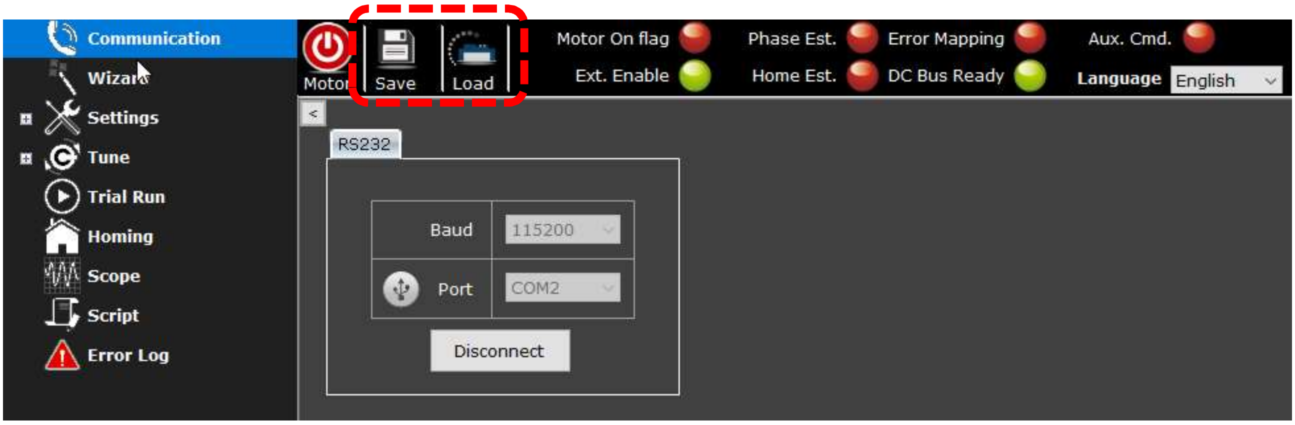




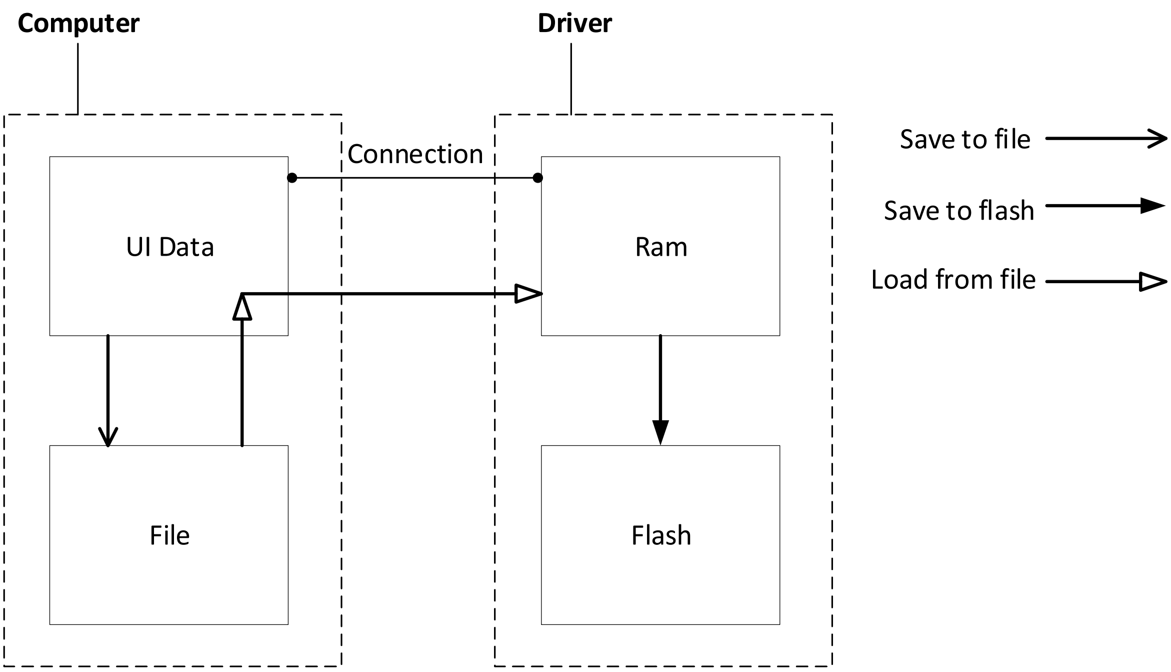
7. When the message “Update process completed successfully” shows up, the firmware burning is completed. The whole flash process may take 3~5 minutes.

1.5 Save and Load Driver Data

The driver data and parameters can be saved and loaded thorough UI. The “Save” and “Load” buttons are on the Tool bar.



- Upon connecting the driver with host computer, most of the driver data will be copied from driver’s RAM to UI Data.
- When parameters are modified, they can be saved to File (which is at host compute) or to driver’s Flash.
  Meanwhile, if parameters are saved in the File, UI can load these parameters from File to driver.



Save:

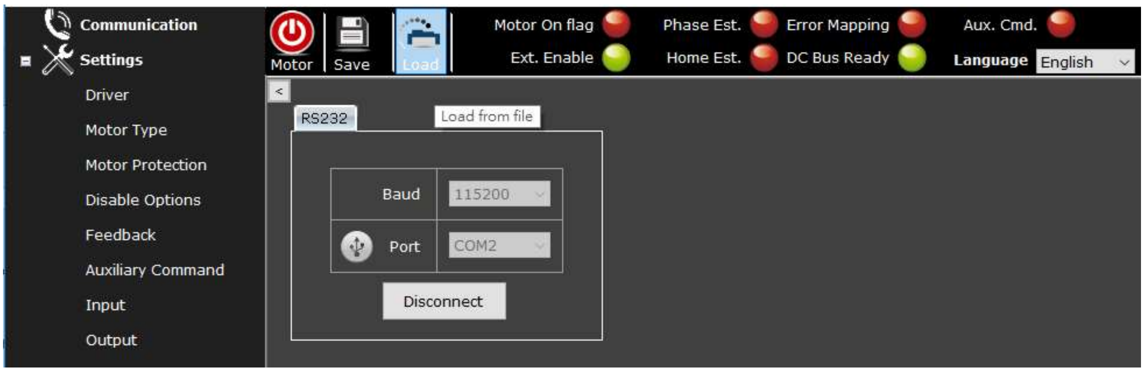
- Save to FILE: Save the parameters from UI Data to File.

- Save to FLASH: Save the parameters from driver's RAM to driver's Flash.
  Note:
  Users can only save to **Flash** while the **motor-on flag is off** (in red color).

Load:

- Load the parameters to driver's RAM and UI Data.
- Click Load and locate your file. The file should be in **.cfg** format.

Users may load driver parameters from the driver itself automatically, or load data from the File which is saved at host computer, or create new parameters. Choose a source to load data and then start setting up the cpc driver.



Meanwhile, the motor parameters—including values and units—are shown in the “Parameter” index under the route of Settings > Driver.

1.6 Setting Process

The information in this UI user guide is arranged in sequence as follows:

Chapter 2, Communication—explains how to connect the cpc driver with host computer.

Chapter 3, Wizard—explains how to set motor parameters and tune the driver step by step using Wizard.

Chapter 4, Setting—describes how to set motor limits, motor types, parameters, feedback, auxiliary command, I/O, boot sequence, and error mapping.

Chapter 5, Tune—describes the autotune procedure, trace, and fine tune procedure using bode plot and time response. The advanced tuning tools, such as gain switch and filter, are described herein.

Chapter 6, Trial Run—describes how to run a motor after the primary parameters, current loop, velocity loop, and position loop have been tuned.

Chapter 7, Homing—describes several homing methods and how homing is performed.

Chapter 8, Scope—explains how to use scope tools to measure and record driver data.

Chapter 9, Script—explains and shows how to edit motor motion commands using Script.

Chapter 10, Error Log—explains why an error message occurs and what each error code means.

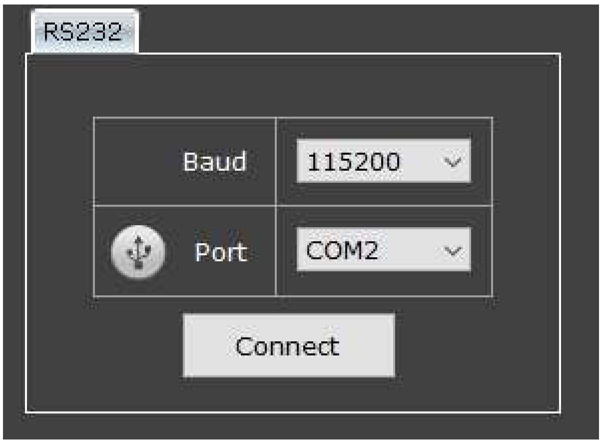
Chapter 2 Communication

RS232 Port:

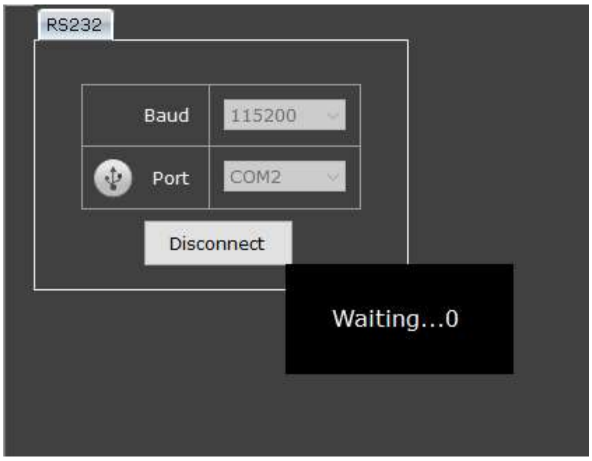
Users should prepare a USB-to-RS232 serial adapter as well as cable to connect the driver with host computer.

Setting steps:

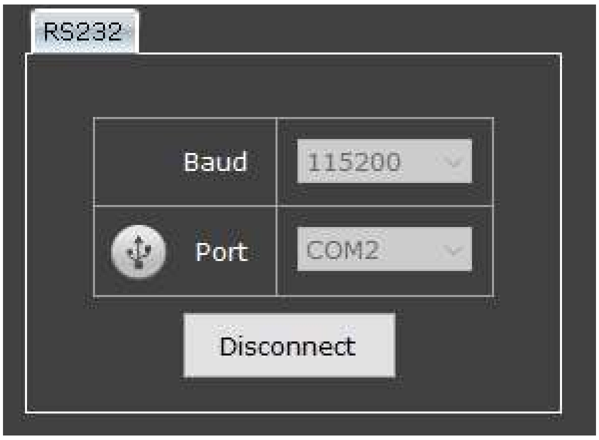
1. After accessing the UI, set Baud rate at 115200 and select a Port. Then, click "Connect".



2. Wait for connection.



3. When the columns turn grey, it means the driver and the host computer are successfully connected.



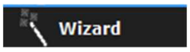
Chapter 3 Wizard

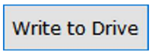
Wizard provides a step-by-step guidance for tuning motor parameters and configuring parameters. It enables users to set up rapidly and conveniently. Usually, after the drive is powered and successfully connected with a host computer, users will need to do the following: **Define motor parameters, set motor protection limits, define encoder resolution, specify I/O pins, tune motor, perform homing, and finally save all parameters to driver flash.**

The cpc UI wizard has a flow-setting page for users **to apply default parameters** of some of the cpc motors as well as Tamagawa motors AND **to designate setting step(s)**. In addition, users can define a new motor of their own.

< Flow-Setting Page>

To Use Wizard:

1. Click [Wizard] 

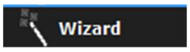
to show the popup window.
2. On the Feature panel, select a cpc or Tamagawa motor model that you're using.
3. Click [Write to Drive] 

.
4. Decide your setting flow by ticking/unticking the process boxes.

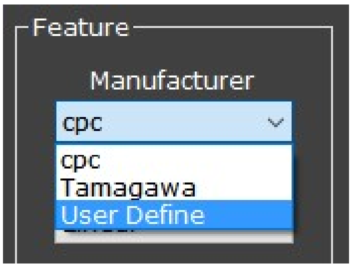
5. Go to popup window and click "Next" to continue.

6. Then keep following wizard's guide until the "Save to Flash" step.

Define a New Motor:

1. Click [Wizard] 

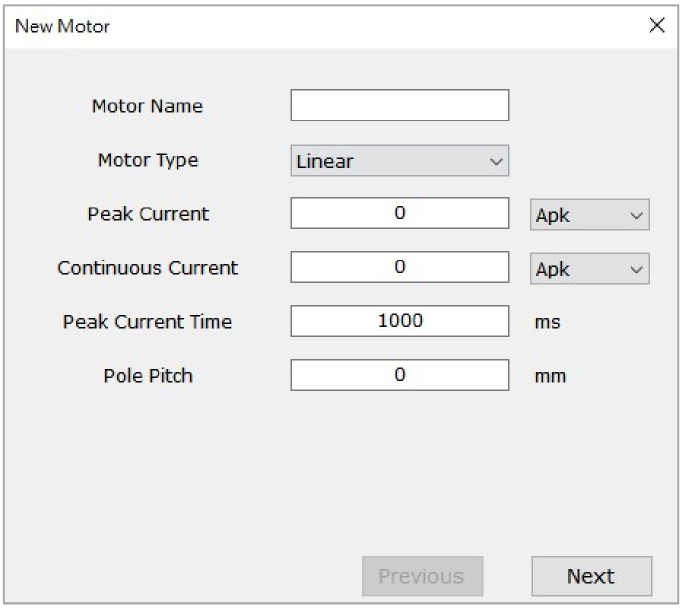
to show the popup window.
2. On the Feature panel, select "User Define" from the manufacturer list.



3. Click 

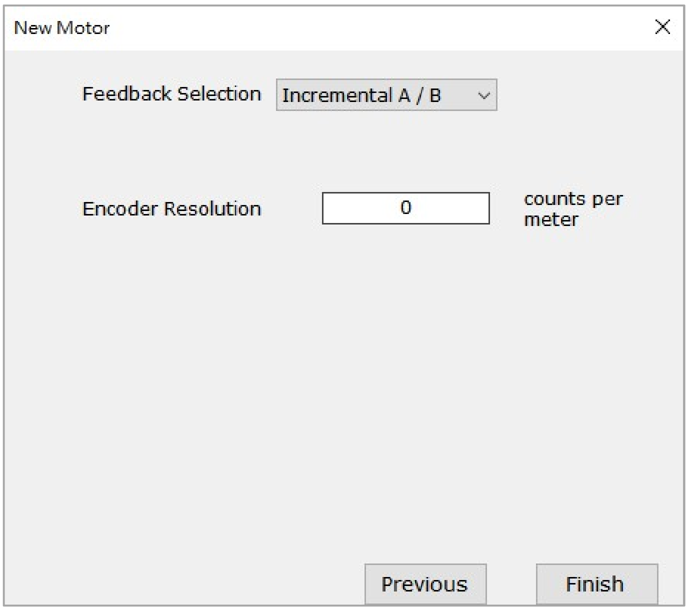
. A window will show up.

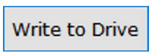
4. Name the new motor. Fill in each motor parameter and then click "Next".



Press
"Enter"
after keying in parameters.

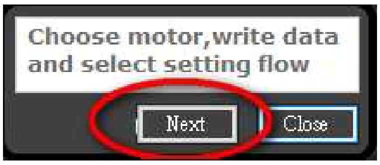
5. Select encoder type (Feedback Selection) and key in encoder resolution. Then click "Finish".



6. Click [Write to Drive] 

and select what to be included in the setting flow:

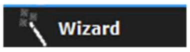
7. Go to the popup window and click “Next” to continue.



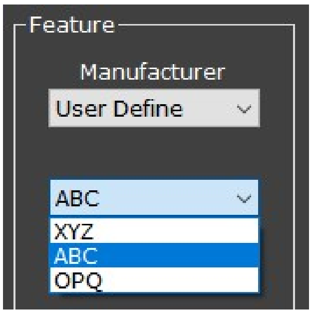
8. Then keep following wizard’s guide until the step “Save to Flash”.

- The data file of user-defined motor can be found at:
  C:\Users\ see Note* \AppData\Roaming\cpc
  Note*: Fill in the account username being used in host computer system.

Delete a User-Defined Motor:

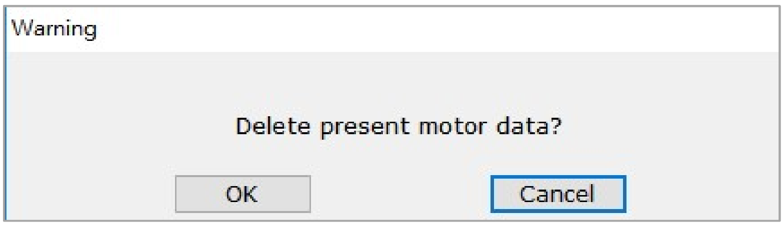
1. Click [Wizard]  to show the popup window.

2. On the Feature panel, select “User Define” from the manufacturer list; then select the name of the user-defined motor which is to be deleted.



3. Click .

4. A dialogue box will show up. Click “OK” to delete data.

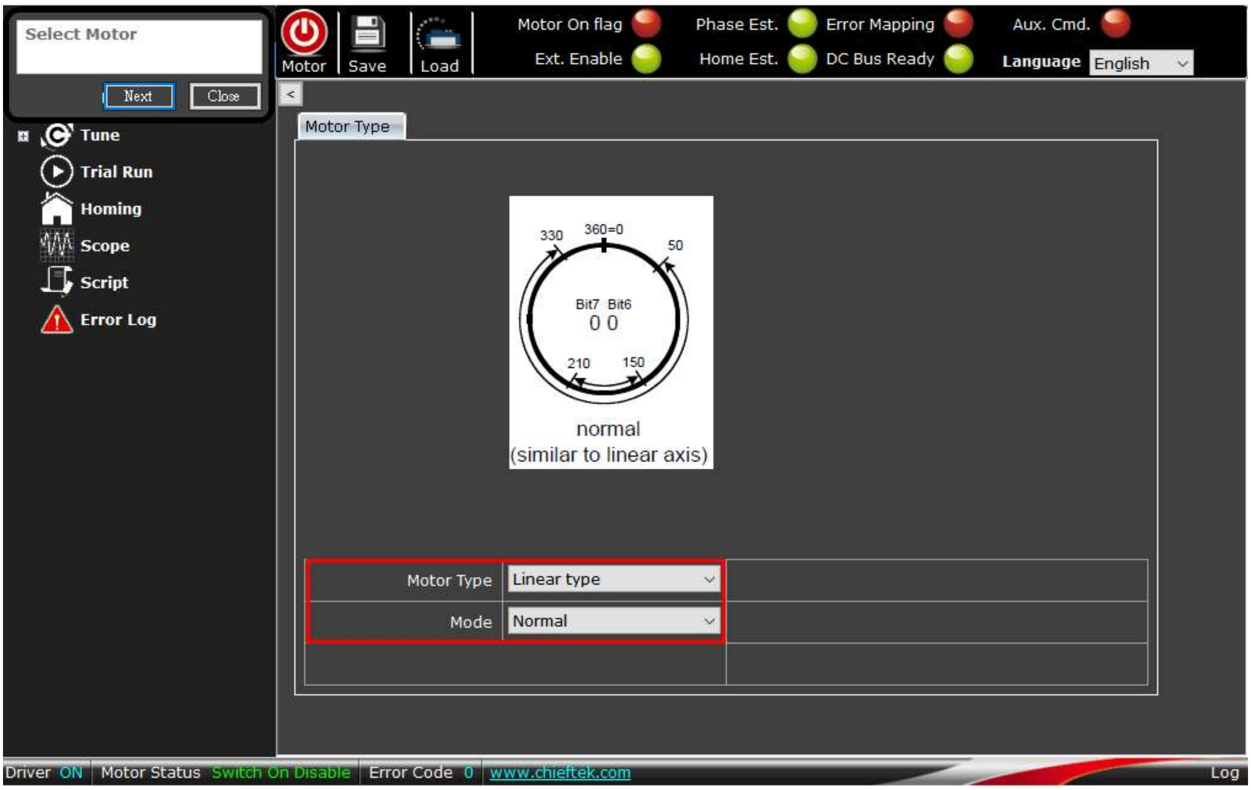


For more detailed and complete information about setting and tuning please refer to the following chapters:

Chapter. 4 Setting  and **Chapter. 5 Tune**  .

3.1 Motor Type

Select motor type (Linear or Rotary) and the mode you need.



- Linear motor has 2 operation modes:

  Normal and Optimized. See **chapter 4.2** for further information.
- Rotary motor has 4 operation modes:

  Normal, Optimized, Only Negative, and Only Positive. See **chapter 4.2** for further information.

3.2 Motor feedback

Set motor feedback parameters under the “Feedback” section in UI.

a. Feedback Selection:

Various signal types of encoder are supported by the cpc drivers:

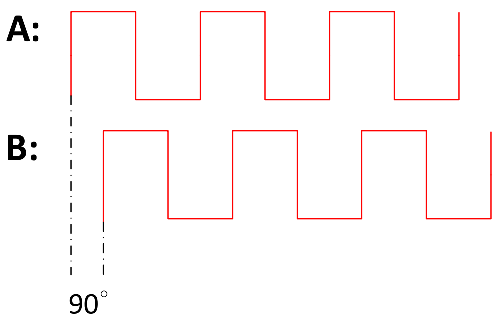
- TC1 Drivers:

  Supports Incremental signals.

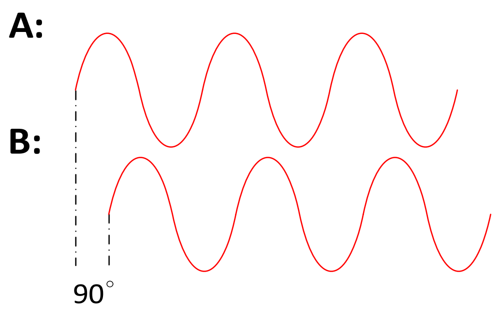
- TC1-B Drivers:

  Supports incremental A/B, Sin/Cos, BiSS-C, and Tamagawa encoder signals.

➢ Incremental Encoder A/B:



- Sin/Cos Encoder:



b. Hall Source:

- Feedback Port:

  Assigns the digital input pins at Feedback port as hall sensor signal source.

- Controller Port:

  Assigns the digital input pins at Controller port as hall sensor signal source.

c. Motor Encoder Resolution:

Sets the resolution (in micro meter) of the encoder.

Be careful while configuring the encoder resolution—especially when reducing the resolution scale.

Note:

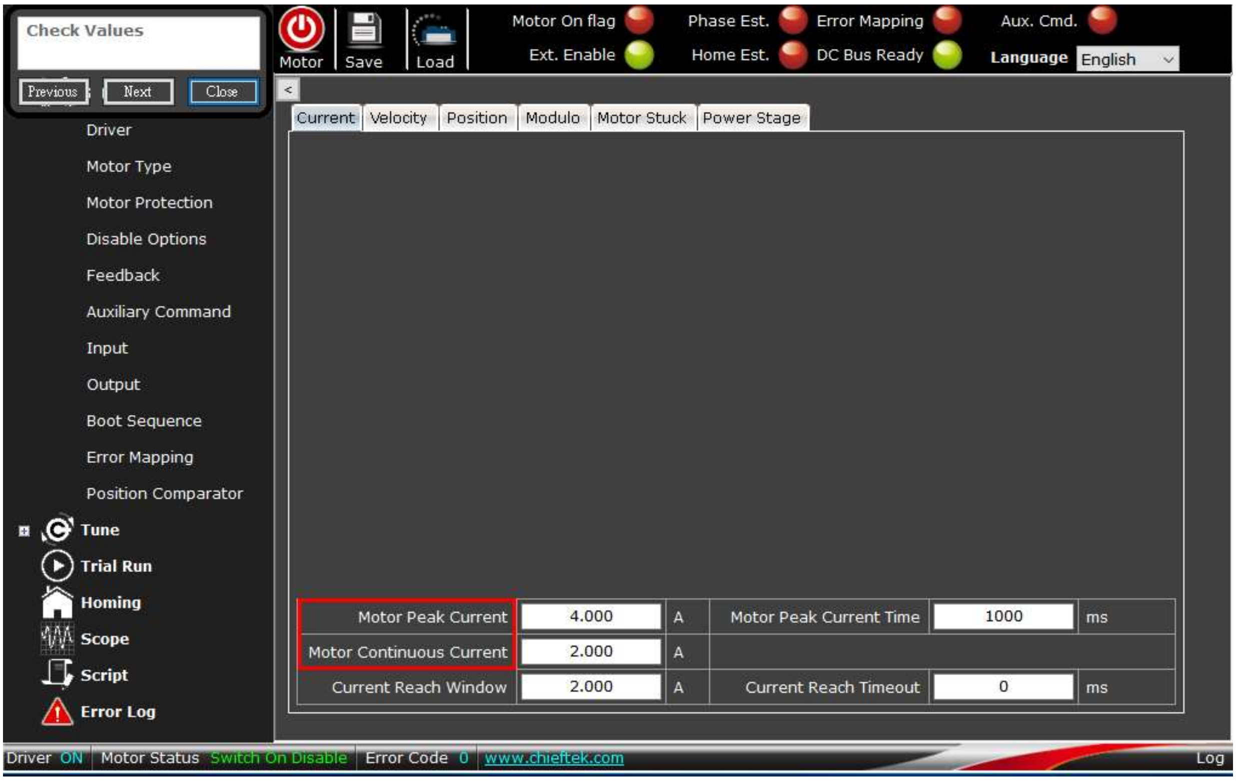
Position, velocity, and acceleration protection (count) values cannot be over 32-bit integer range.

d. Motor Pole Pitch:

Sets the length of one set of pole pitch (in millimeter) of the motor.

3.3 Motor Protection

Fill in motor parameters respectively:



a. **Motor Peak Current / time:**

The maximum transient current allowed for the motor and the corresponding time, in amperes and milliseconds.

b. **Motor Continuous Current:**

The maximum continuous current allowed for the motor, in amperes.

Note:

Please check the parameters mentioned in a. and b. carefully.

These parameters relate to the stability and safety of the system. Please make sure you fill in correct data!

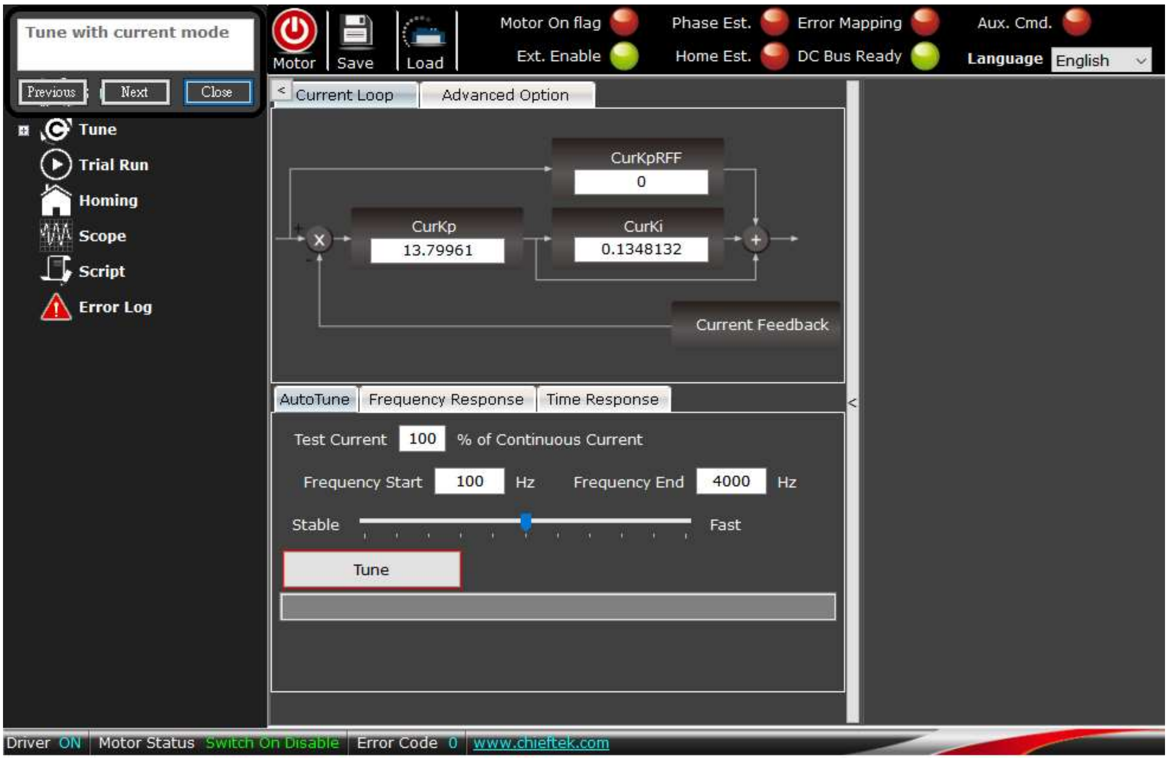
3.4 Input & Output

The I/O pin definitions of feedback and controller ports are elaborated in driver's *Installation Guides*.

Note: More detailed information please see **chapter 4.7 Input & Output.**

3.5 Tune – Current

After configuring motor data and digital input/output, users can start tuning the Current, Phase, and Velocity.



a. **Test Current**:

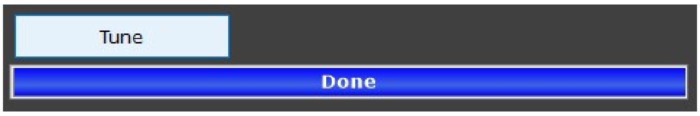
Percentage of continuous current.

b. **Frequency Start/End**:

Frequency response range.

To tune Current:

1. Click the "Tune" button, UI will tune the gain automatically.
2. Wait till the "Done" signal shows up, it means tuning is completed.



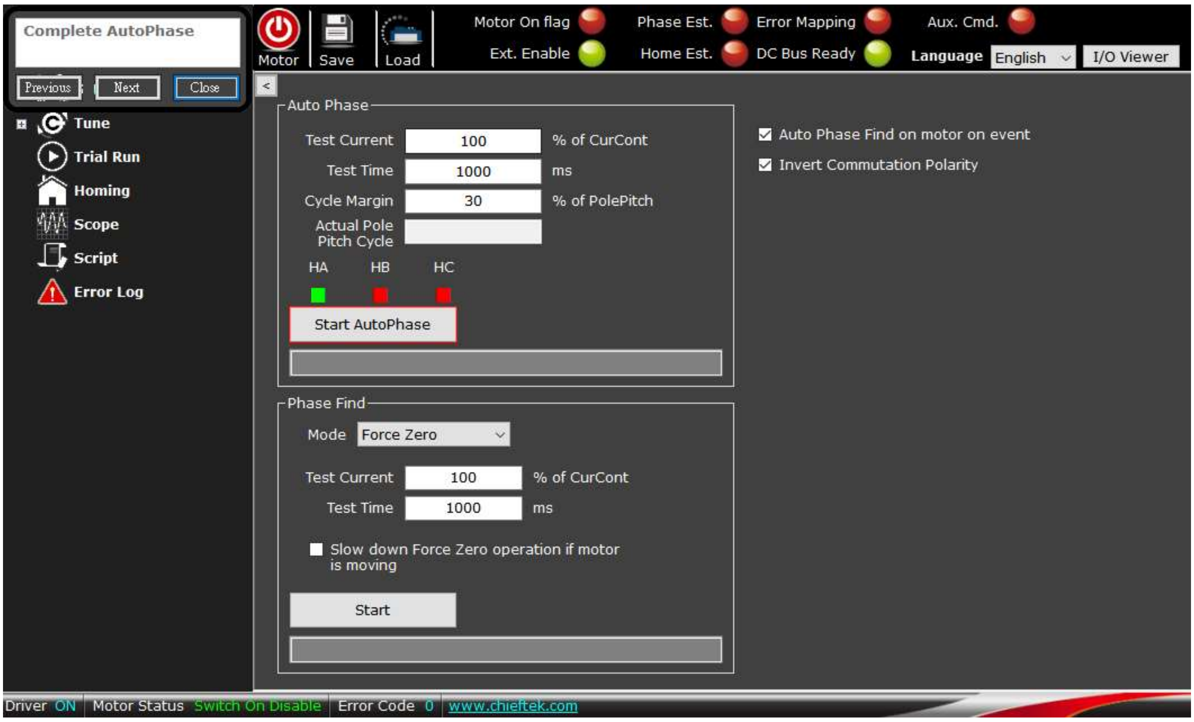
3. Next, tune the Phase.

Note: Auto tuning is only for reference!

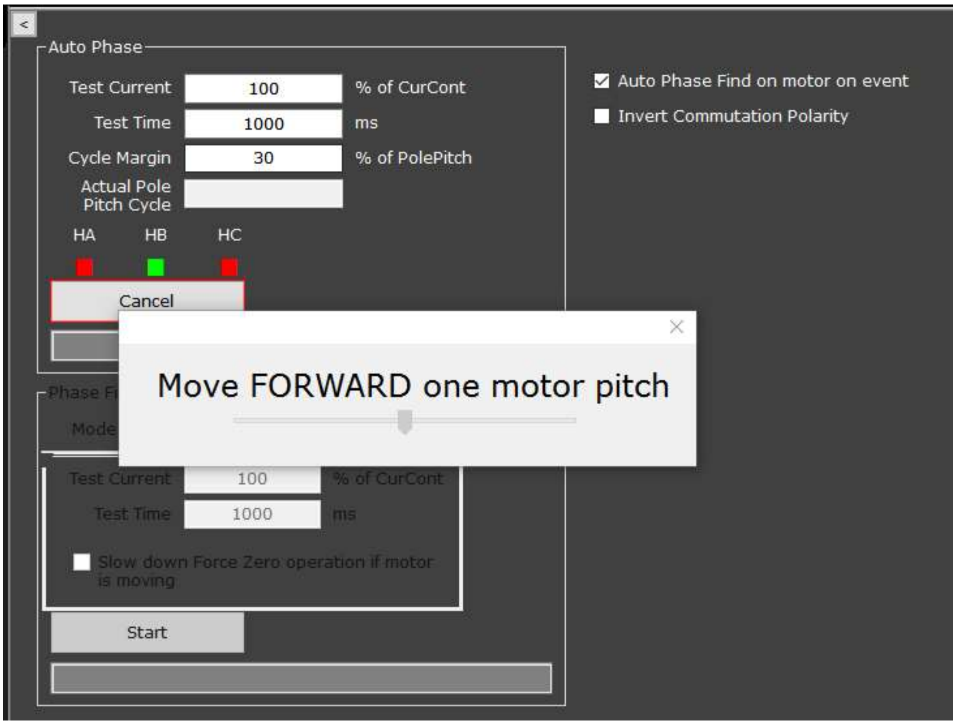
3.6 Tune – Phase

Steps:

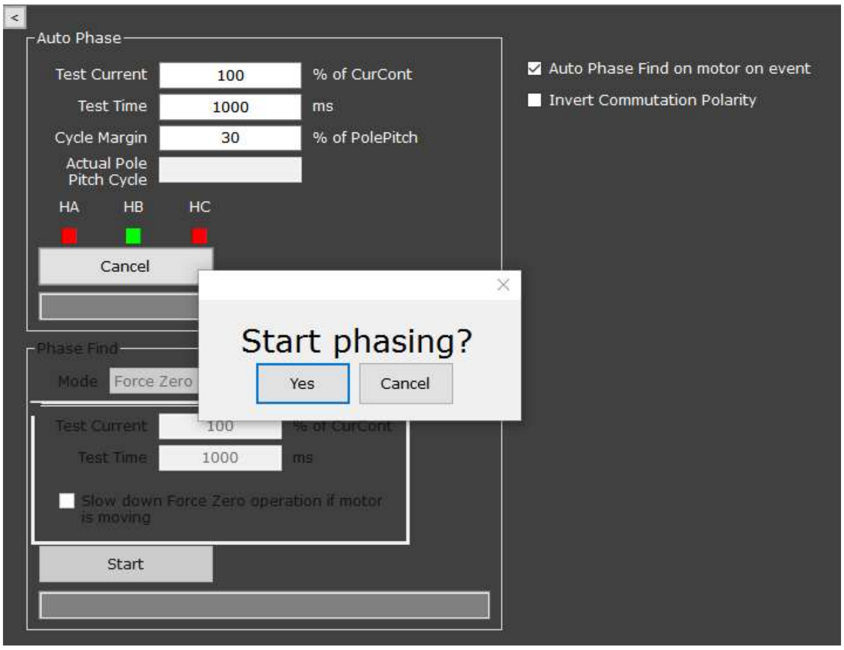
1. Click "Start AutoPhase".



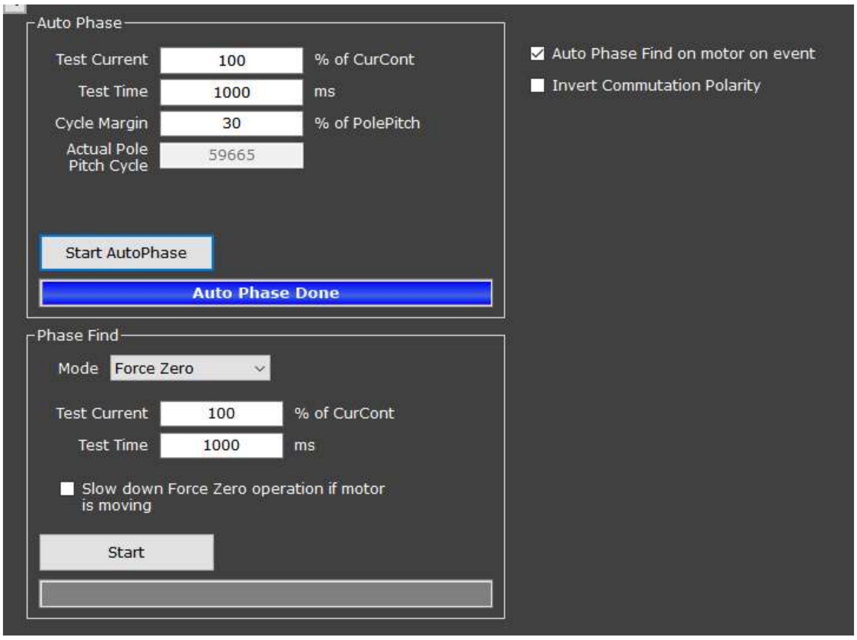
2. Move the motor manually at least one pole pitch in **positive direction**.



3. Click “Yes” to start phasing.



4. Wait until you see the message “Auto Phase Done”.



a. **Test Current:**

Percentage of continuous current.

b. **Test Time:**

Testing time.

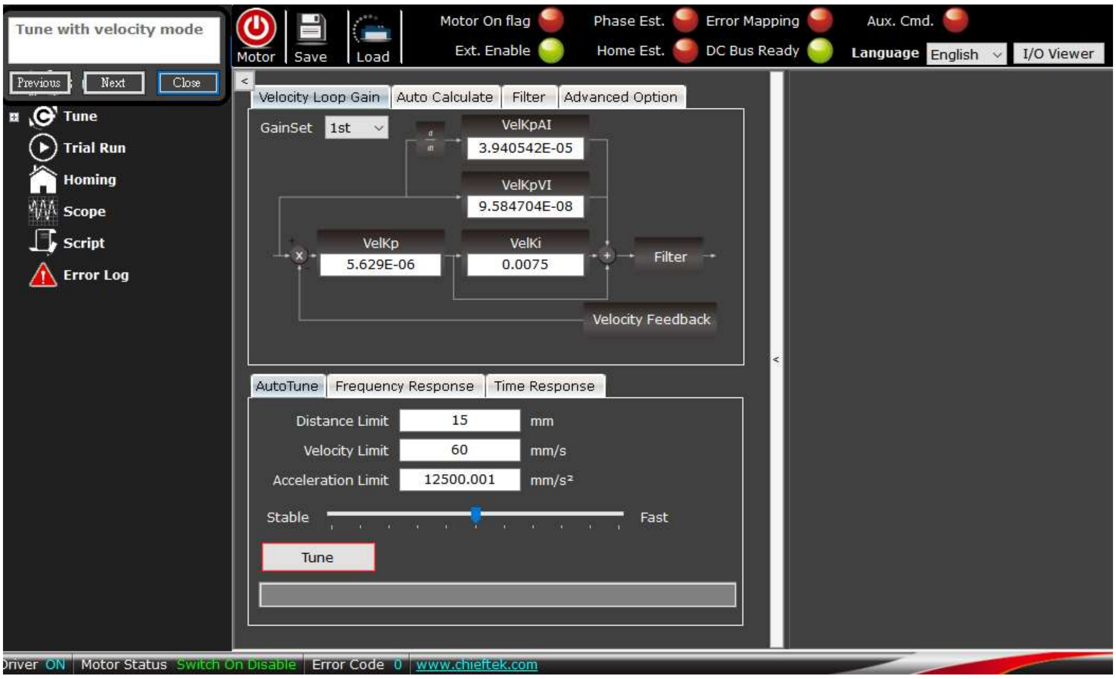
c. **HA, HB, HC:**

Hall sensors status.

3.7 Tune – Velocity

Tuning the velocity gain is similar to tuning the current gain.

1. Click “Tune” button.

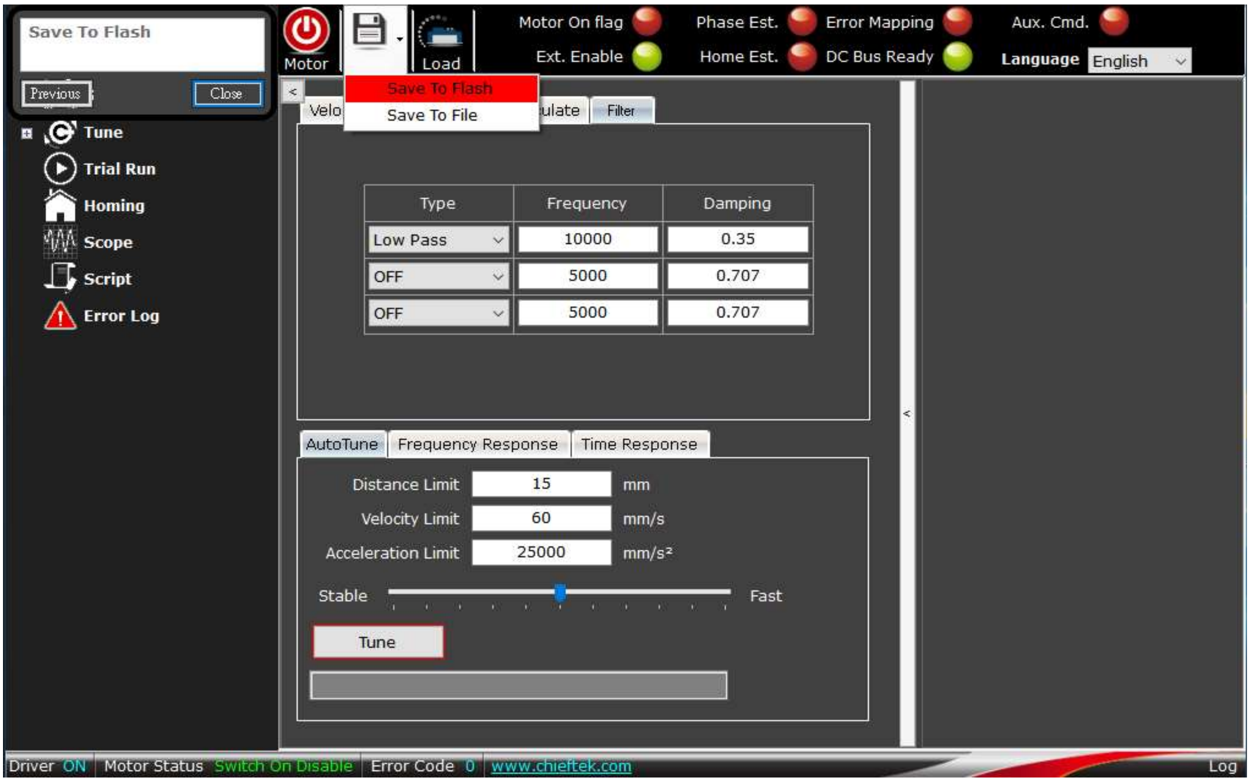


2. When the “Done” signal shows up, tuning is completed.

3.8 Save to Flash

Click "Save to Flash" to save parameters to the drive Flash.

If you wish to save to a file in your computer, select "Save to File" and then designate a route.



Chapter 4 Settings

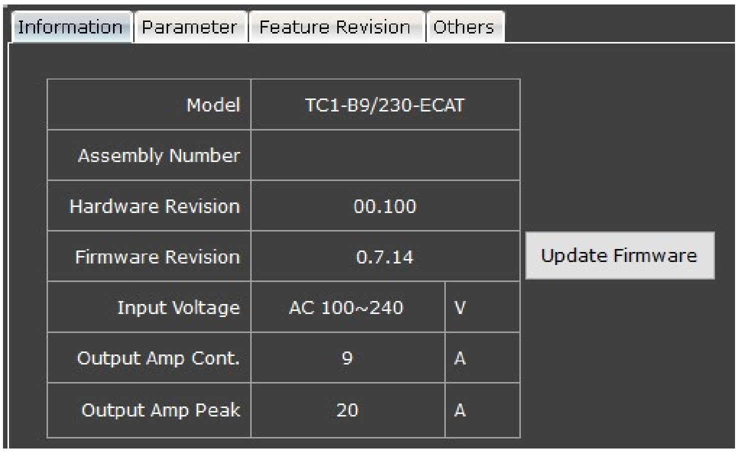
Setting procedure includes configuring the motor type, motor protections, disable option, boot sequence, and those mentioned in the Quick Start chapter (feedback, input, and output). In practice, the driver often needs to receive signals from controller; hence, the "Auxiliary Command" section is for setting the connection between controller and driver.

4.1 Driver

The "Driver" section contains basic information and parameters of cpc drivers.

4.1.1. Information

Shows basic information of the driver, including hardware/firmware revision, input voltage, output continuous current, and peak current.



a. **Update Firmware**:

Click to update driver's firmware.

<u>**Note**</u>:

Before updating, make sure you have downloaded the latest firmware revision from cpc official website.

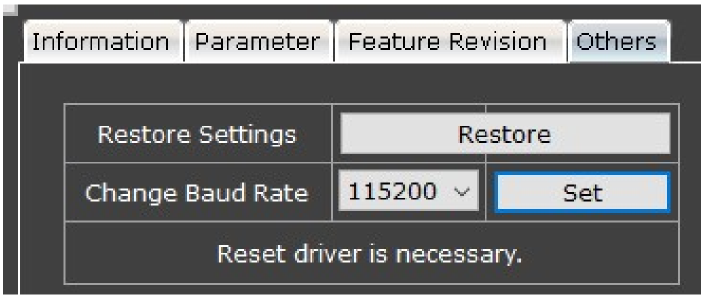
4.1.2 Parameter

Shows the values and units of variables.

4.1.3 Feature Revision

For cpc internal reference.
It shows the firmware revision of each subordinate part.

4.1.4 Others



a. **Restore Setting**:
   Click to reset to the default settings.

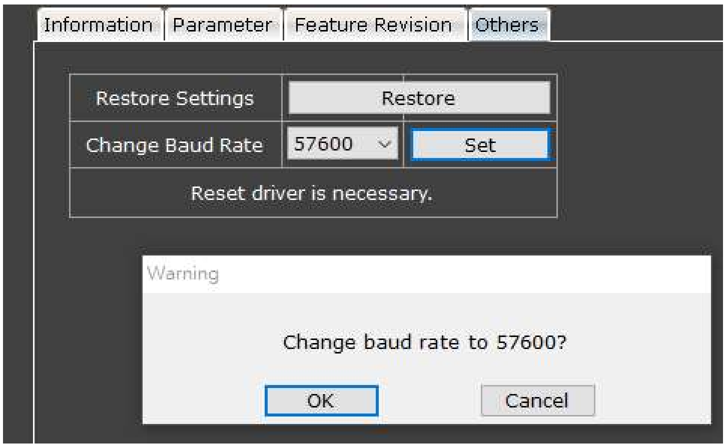
b. **Change Baud Rate**:
   The default baud rate is 115200.

   <u>Note</u>:
   - Any baud rate above 115200 is not supported.
   - Please RESET the driver after changing baud rate setting.

Baud rate setting steps:

(1) Select a driver baud rate, say 57600. Then click “Set”.

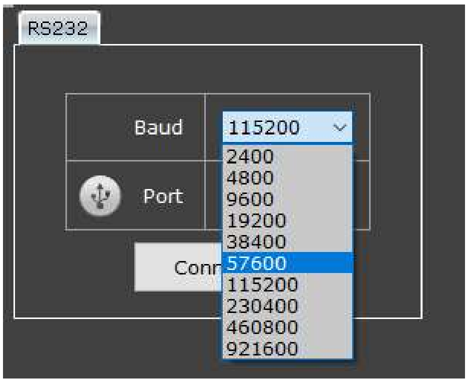
(2) Click “OK” when the warning message “Change baud rate to 57600?” shows up.



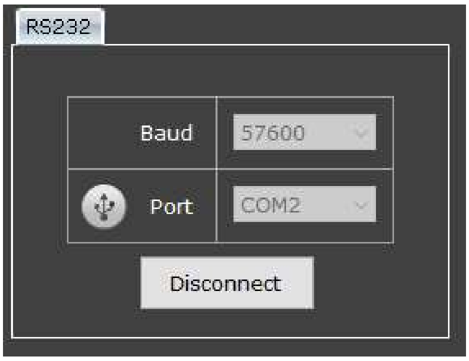
(3) Power off the driver.

(4) Power on the driver again.

(5) When the startup screen shows up, choose the new baud rate you selected (in this case “57600”) and choose connection port. Then click “Connect”.

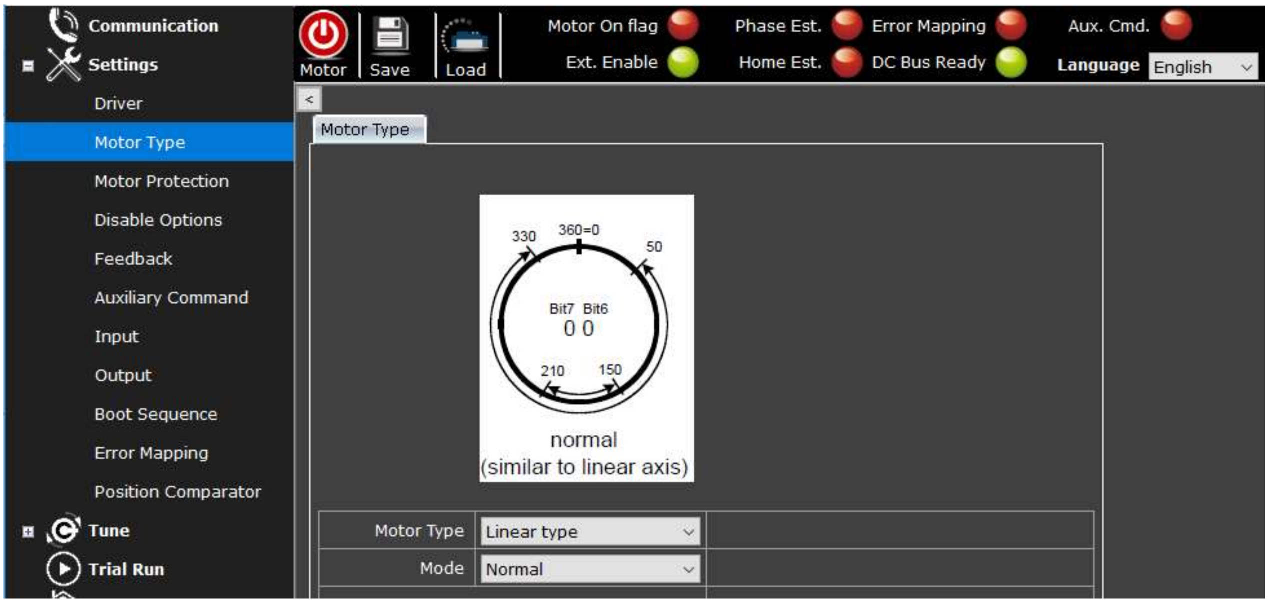


(6) Wait a few seconds. If words in the columns turn grey, this means the communication is successfully built.



4.2 Motor Type

Choose the Motor Type and Mode that suits your needs.



The "Motor Type and Mode" table below shows each motor-mode's effect on <u>Position Limit</u> (i.e., Command Forward/Backward Limit) **and <u>Position Range</u>** (i.e., modulo); see explanations on next page. The following chapters elaborate on details of these **6** type-mode combinations.

<u>< Motor Type and Mode ></u>

<table>
<tr><td>Motor Type :</td><td colspan="4">Linear</td><td colspan="6">Rotary</td></tr>
<tr><td>Motor Mode :</td><td colspan="2">Normal</td><td colspan="2">Optimized</td><td colspan="2">Normal</td><td>Only Negative (CCW)</td><td>Only Positive (CW)</td><td colspan="2">Optimized</td></tr>
<tr><td rowspan="2">Position Command Type</td><td>Abslt.</td><td>Rel.</td><td>Abslt.</td><td>Rel.</td><td>Abslt.</td><td>Rel.</td><td>Abslt.</td><td>Abslt.</td><td>Abslt.</td><td>Rel.</td></tr>
<tr><td colspan="10">Abslt.: absolute. Rel.: Relative.</td></tr>
<tr><td rowspan="2">Position Limit
(= Command Forward /Backward Limit)</td><td colspan="2">Y</td><td colspan="2">N</td><td colspan="2">Y</td><td colspan="2">Y</td><td colspan="2">Y</td></tr>
<tr><td colspan="10">Y: Users need to define. N: Users do not need to define.</td></tr>
<tr><td rowspan="2">Position Range
(= modulo)</td><td colspan="2">X</td><td colspan="2">X</td><td colspan="2">O</td><td colspan="2">X</td><td colspan="2">O</td></tr>
<tr><td colspan="10">O: User-defined position range is allowed.
X: User-defined position range is not supported.</td></tr>
</table>

1. **Explanations of the <Motor Type and Mode> Table**:
   (1) **Motor Type**:
   Linear or rotary.

   (2) **Motor Mode**:
   The modes that each motor type includes.
   See detailed characteristic of each mode in subsequent chapters.

   (3) **Position Command Type**:
   Giving **absolute** or **relative** position commands when operating motors.

(4) **Position Limit**:

i. It refers to the term “**607Dh**”—the **Software position limit**—in CiA® 402 Draft Standard Proposal.

ii. <u>**Note**</u>:

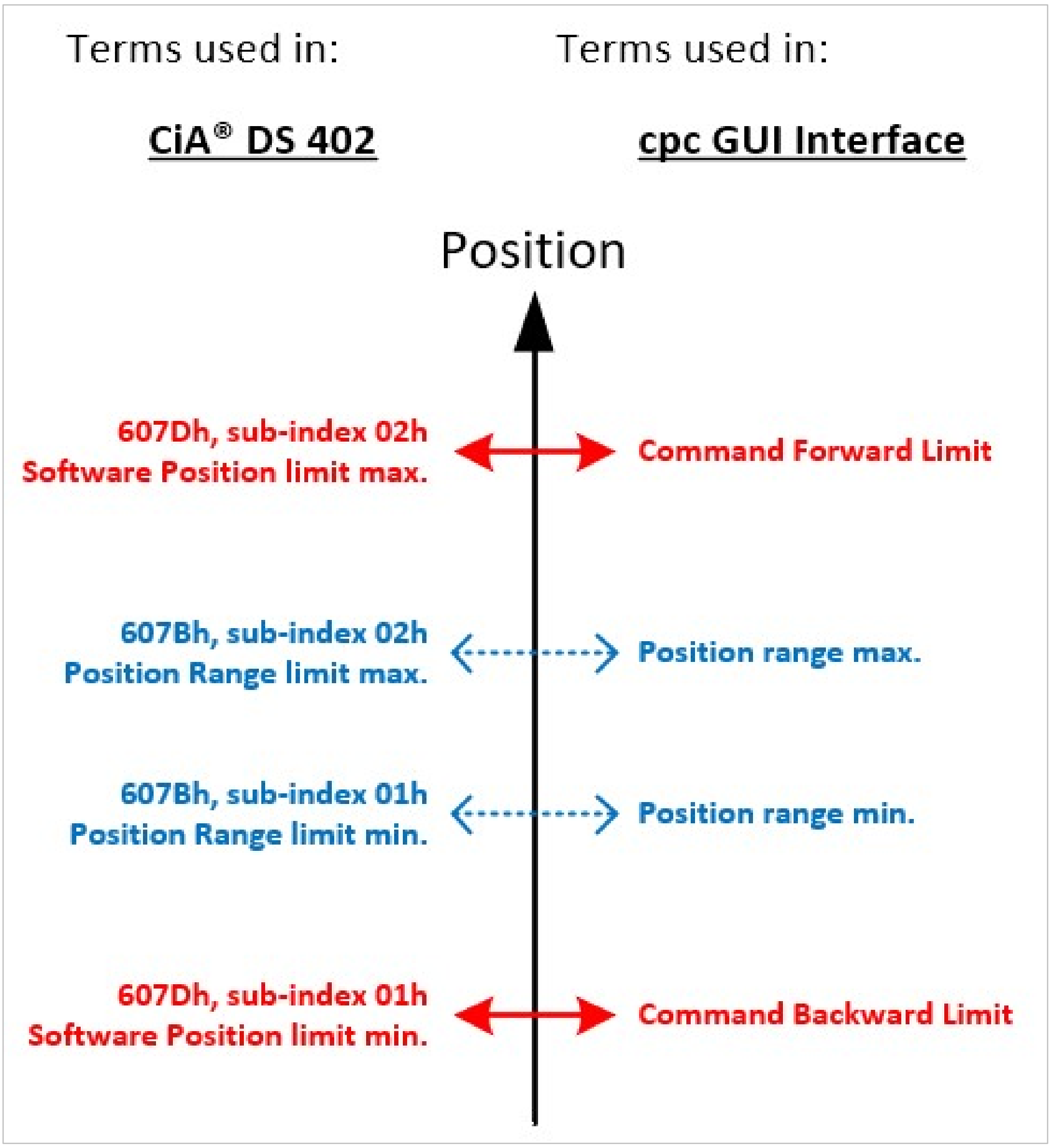
Perform **Homing** before setting the Command Forward /Backward Limit in UI.

iii. As explained in CiA® 402 Draft Standard Proposal: …*This object shall indicate the configured maximal and minimal software position limits. These parameters shall <u>define the absolute position limits for the position demand value and the position actual value</u> as specified in Figure 23. Every new target position shall be checked against these limits.*

<u>To disable the software position limits, the min position limit (sub-index 01h) and max position limit (sub-index 02h) shall be set to 0.</u> The limit positions shall be given in user-defined position units (same as target position). Supervision of software position limits requires a defined home position.

iv. <u>Where to set Position Limit in cpc GUI</u>:

Motor Protection > Position > the columns of **“Command Forward Limit”** and **“Command Backward Limit”**.



(5) **Position Range (i.e., Modulo)**:

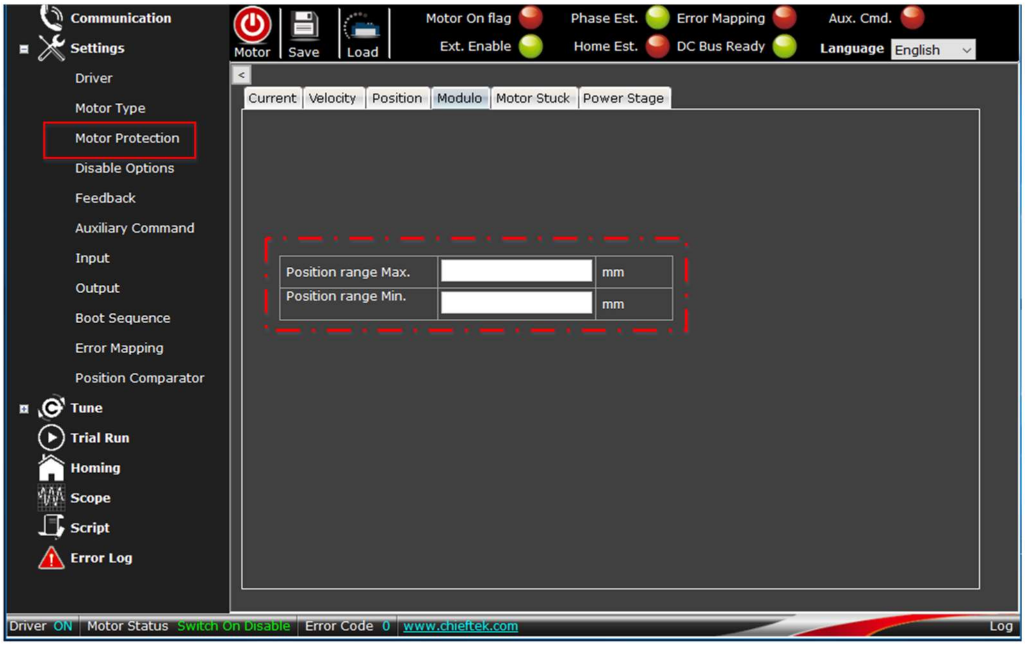
i. It refers to the term "**607Bh**"—the **Position range limit**—in CiA® 402 Draft Standard Proposal.

ii. <u>**Note**</u>:

Please perform **Homing** before setting Position range Max./Min. in cpc UI.

iii. As explained in CiA® 402 Draft Standard Proposal: …*This object shall indicate <u>the configured maximal and minimal position range limits</u>. It shall limit the numerical range of the input value. <u>On reaching or exceeding these limits, the input value shall wrap automatically to the other end of the range</u>. Wrap-around of the input value may be prevented by setting software position limits* *as defined in software position limit object (607Dh). To disable the position range limits, the min position range limit (sub-index 01h) and max position range limit (sub-index 02h) shall be set to 0.*

iv. Where to set Position range in cpc GUI:
Motor Protection > Modulo > the columns of "**Position range Max**." and "**Position range Min**."



2. This **major rule** applies to all combinations of motor type, motor mode, and position command type: **If position command exceeds or is lower than the Position Limit, this command will be ignored.**

3. **Note:**
Based on cpc's design for the setting of (4) Position Limit and (5) Position Range, if you set the minimum value to 0 and the maximum value to 1000, your target position command can be 0 or 1000.

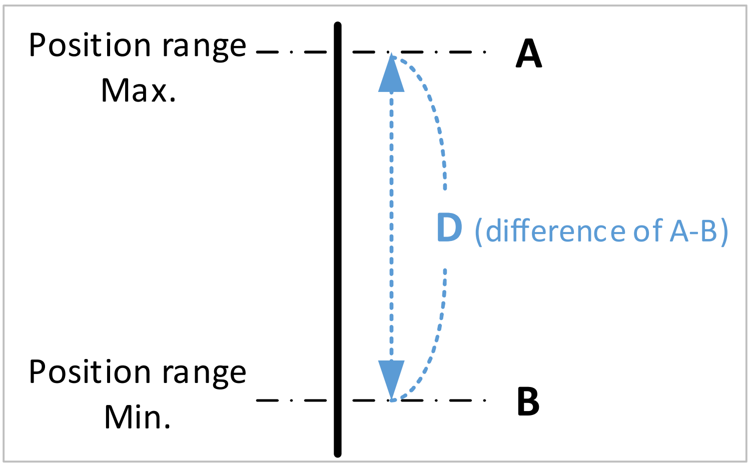
4. **"Optimized" mode** will take **the shortest way** to reach the target position. In addition, if you choose to give **relative** target position command under optimized mode, the "absolute value" of your position command **must be less or equal to half of Position Range (i.e., modulo); otherwise the command will be ignored.** See further explanation below.

Formula:

Suppose (1) the value of your **Relative** position command is **X** (units).

(2) The values of maximum and minimum of Position Range are **A** and **B**;

(3) The difference between A and B is **D** (units).



$$\boldsymbol{A - B = D} \ \textbf{units,}$$

$$|x| \leq \frac{1}{2}D \ .$$

Hence, $-\frac{1}{2}D \leq X \leq \frac{1}{2}D.$

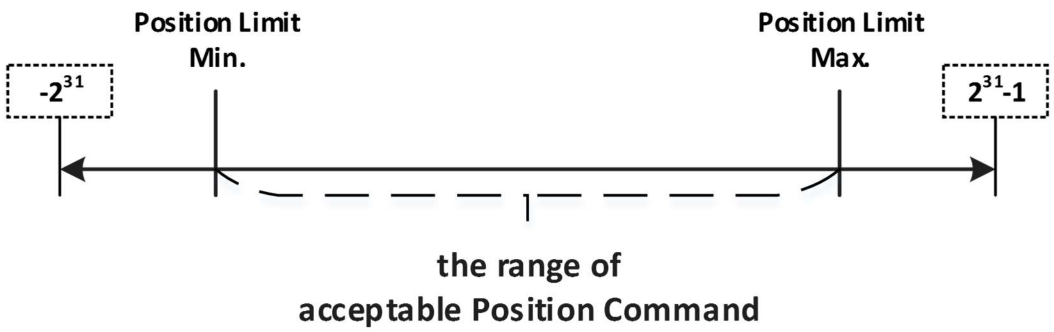
Please see characteristics of each mode on subsequent pages.

4.2.1 Linear Motor / Normal Mode

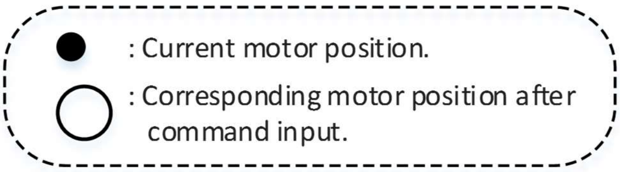
1. The motor moves within Position Limit (607D).
2. No Position Range (607B) (Position Range Max. and Min. will show 0 in GUI).
3. If target position command exceeds or is lower than the Position Limit, the command will be ignored.
4. If relative position command will exceed or be lower than the Position Limit, the command will be ignored.

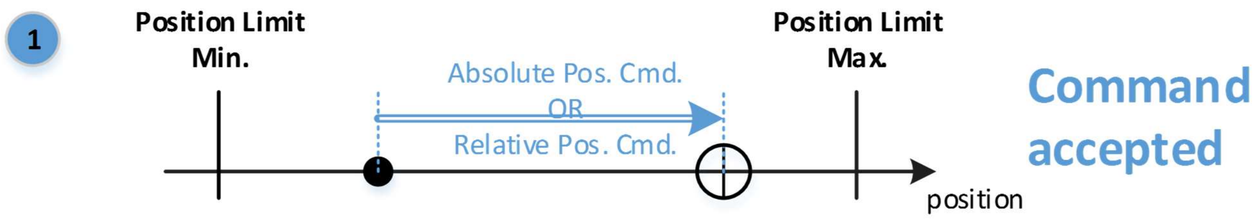
Linear / Normal

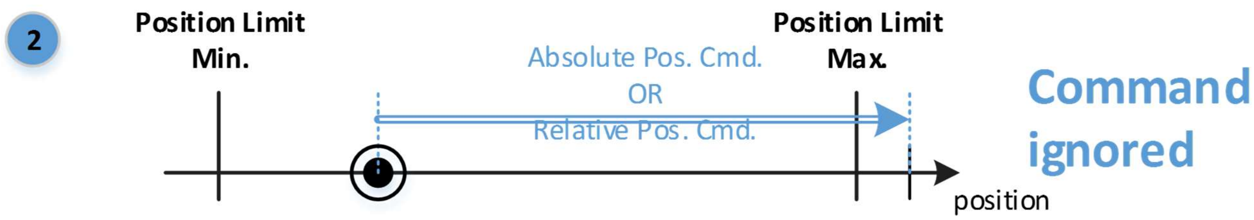
Rule



Examples (Linear / Normal)







4.2.2 Linear Motor / Optimized Mode

1. **This Linear/Optimized mode is for special application, such as operating a motor with high-resolution encoder**.
2. No Position Limit (607D) (Position Limit Max. and Min. will show 0 in GUI).
3. The Position Range (607B) under this mode is the maximum and minimum of 32bit (i.e., **from** -2^{31} **to** $2^{31}-1$).
4. **Absolute** position command:
   - When the [**dP**]* is less or equal to half of the Position Range (i.e., 2^{31}), the motor will move in positive direction; vice versa.

     *: [dP] is the distance between the absolute position command and the present position; namely, it is the "absolute value" of the distance difference between the absolute position command and the current position.
   - If the command exceeds or is less than the Position Range, it will be ignored.
5. **Relative** position command:
   The "absolute value" of your relative position command must be less or equal to half of the Position Range (i.e., 2^{31}); if relative command exceeds so, it will be ignored.

See illustration and examples on subsequent pages.

Linear / Optimzied

Rule

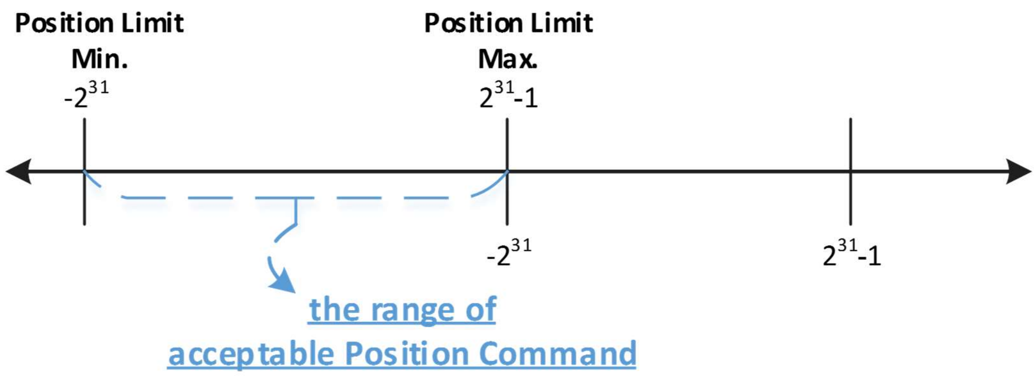
Position range = 2^{32}, from -2^{31} to $2^{31}-1$.

● : Current motor position.

○ : Corresponding motor position after command input.

➡ : Motor movement direction.

The driver supports up to 32 bit;
This Linear/Optimized mode allows to **wrap.**



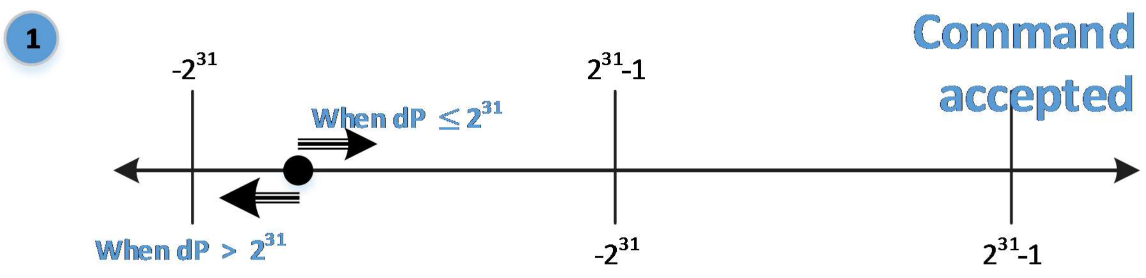
See examples on next page.

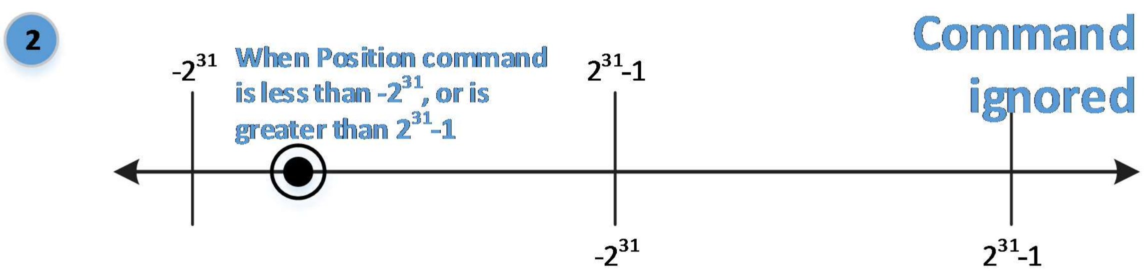
Examples (Linear / Optimized)

Absolute Position Command.

dP: | Pabs.cmd – Pcur | =
the "absolute value" of the difference between the absolute position command and the current position.

2^{31} = ½ of the Position range.

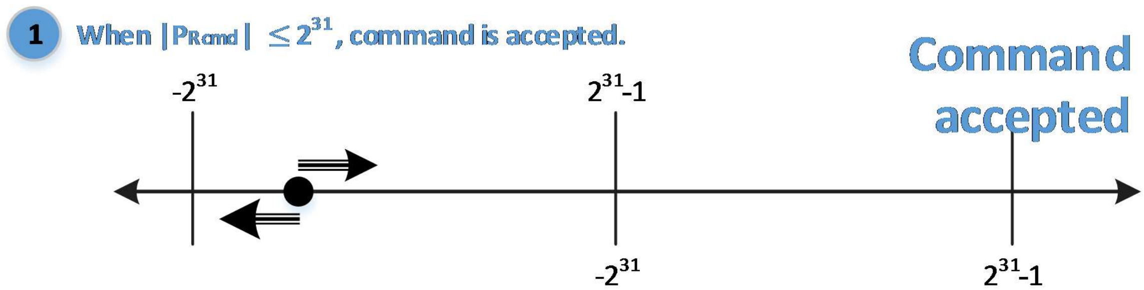


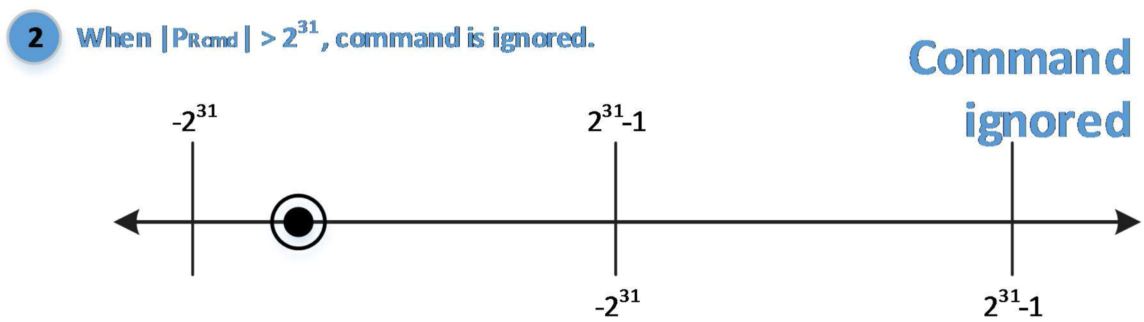


Relative Position Command

PRcmd : relative position command value.
|PRcmd|: the "absolute value" of the relative position command.

2^{31} = ½ of the Position range.





4.2.3 Rotary Motor /Normal Mode

1. Position commands can exceed Position Range; if reaching or exceeding the Position Range, the input value will **wrap**.
2. If **absolute** position command exceeds Position Limit, the command will be ignored.
3. If **relative** positioning will exceed Position Limit, the command will be ignored.

4.2.4 Rotary Motor / Only Negative (CCW)

1. No “607B” (Position Range)
   (Position Range Max. and Min. will show 0 in GUI).
2. Rotates only in negative direction.
3. If target position command is **higher** than actual position, the motor will rotate to the minimum Position Limit and then wrap.
4. If **absolute** position command exceeds Position Limit, the command will be ignored.

4.2.5 Rotary Motor / Only Positive (CW)

1. No “607B” (Position Range)
   (Position Range Max. and Min. will show 0 in GUI).
2. Rotates only in positive direction.
3. If position command is **lower** than actual position, the motor will rotate to the maximum Position Limit and then wrap.
4. If **absolute** position command exceeds Position Limit, the command will be ignored.

4.2.6 Rotary Motor / Optimized Mode

1. **Absolute** position command:
   (1) cannot exceed Position Limit and Position Range, otherwise the command will be ignored.
   (2) If the difference between the current position and Absolute position command is larger than half of Position Range, the motor will move in negative direction.

2. **Relative** position command:
   (1) cannot exceed Position Limit, otherwise the command will be ignored.
   (2) The "absolute value" of the **relative** position command **must be less or equal to half of Position Range**, otherwise the command will be ignored.

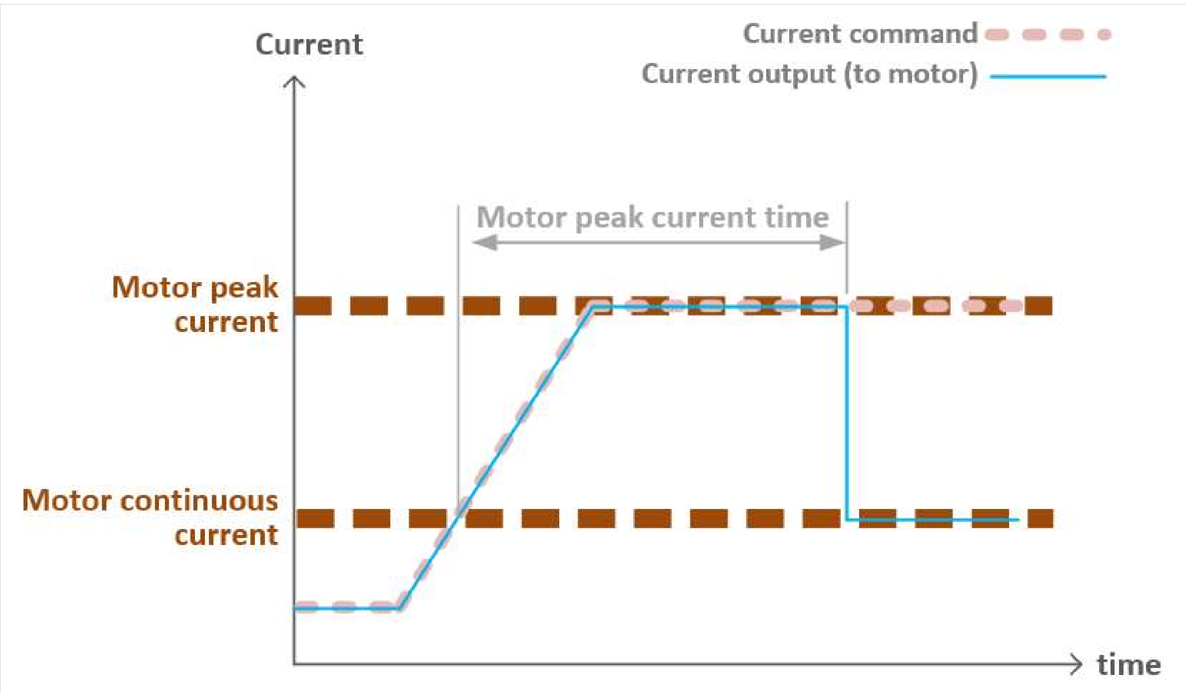
4.3 Motor Protection

Motor Protection is for safety and ensuring the stability of the motor.
The main protections—such as current limits, over speed limit, backward limit, and forward limit—are set under Motor Protection section.
For safety purpose, please check the parameters carefully.

4.3.1 Current

Motor Peak Current	4.000	A	Motor Peak Current Time	1000	ms
Motor Continuous Current	2.000	A			
Current Reach Window	2.000	A	Current Reach Timeout	0	ms

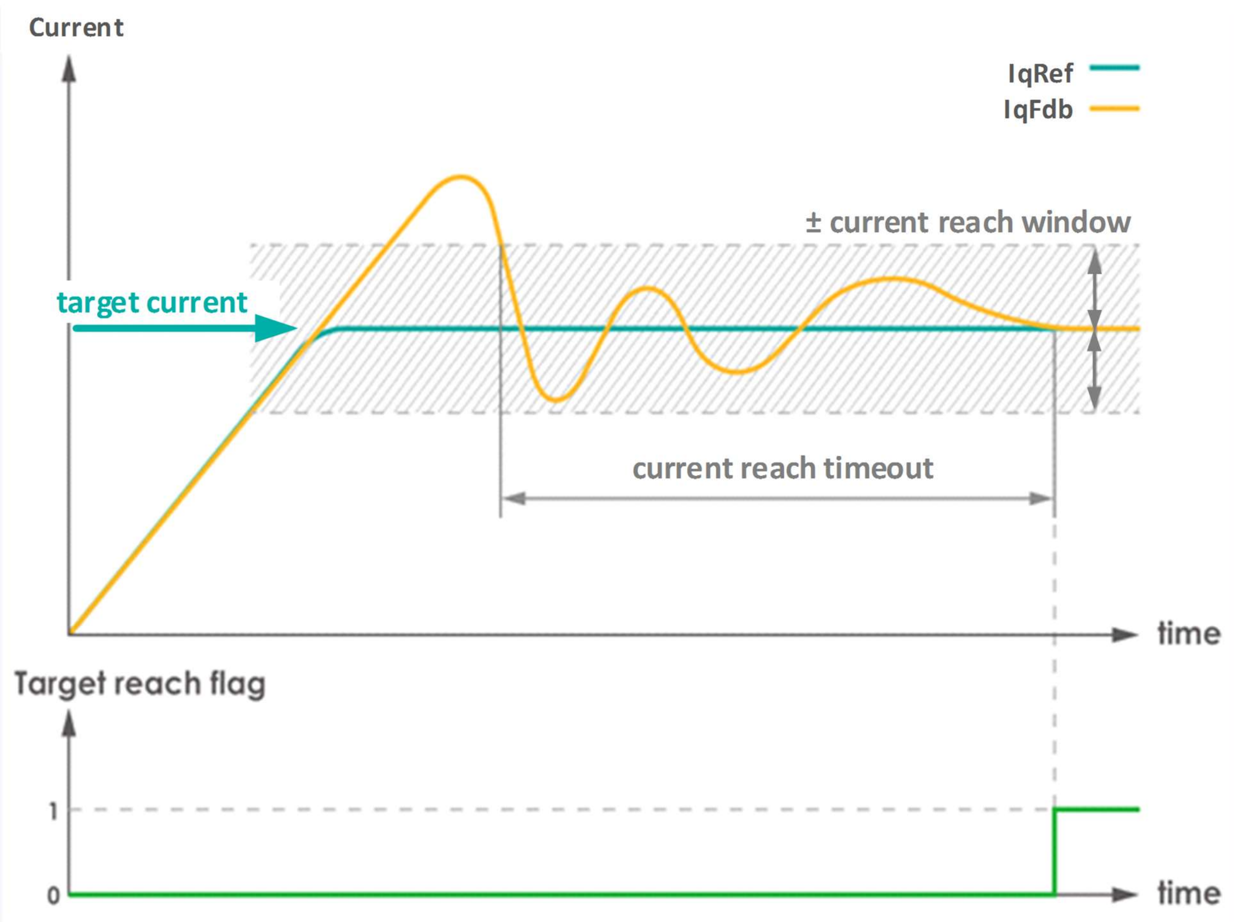
a. **Motor Peak Current:**
   The maximum transient current acceptable for the motor.
b. **Motor Peak Current Time:**
   The maximum time of peak current acceptable for the motor.
c. **Motor Continuous Current:**
   The maximum continuous current acceptable for the motor.



d. Current Reach Window / Timeout:

- The condition of current target reach flag and the corresponding time.
- When current feedback is within the window and continues for a period (reach timeout), the target reach flag will rise. **When the time is set to 0, this detection function will be deactivated.**

The current-time diagram below describes how “current reach window” and “current reach timeout” work.



IqFdb refers to **current feedback**; likewise, velocity applies the same method to rise target reach flag and to determine whether the reference command is met.

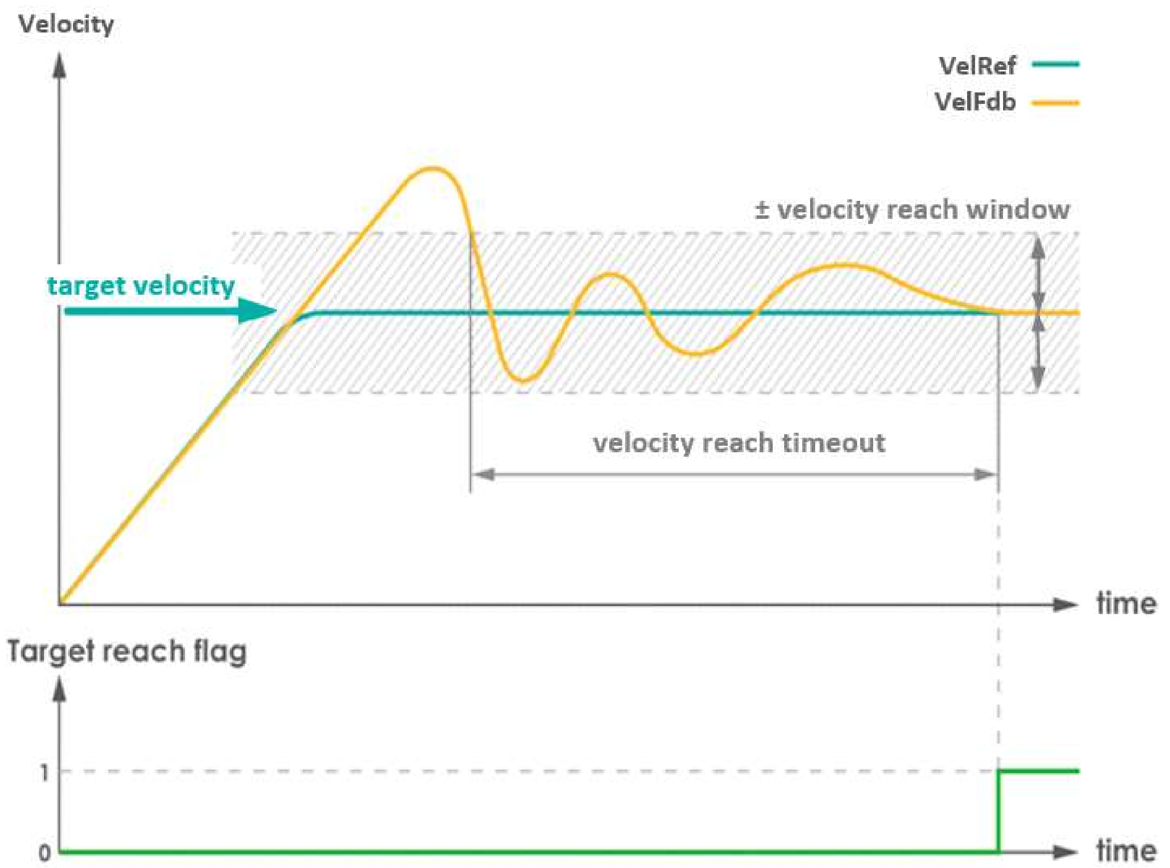
Please check chapter 5.5.4 Target reach flag

4.3.2 Velocity

Velocity Reach Window	2000	cnt/s	Velocity Reach Timeout	5	ms
Velocity Zero Threshold	1000	cnt/s	Velocity Threshold Timeout	1	ms
Motor Rated Speed	750.000	mm/s	QuickStop Deceleration	50000.000	mm/s²
Over Vel. Fault Limit	1250.000	mm/s			

a. Velocity Reach Window / Timeout:

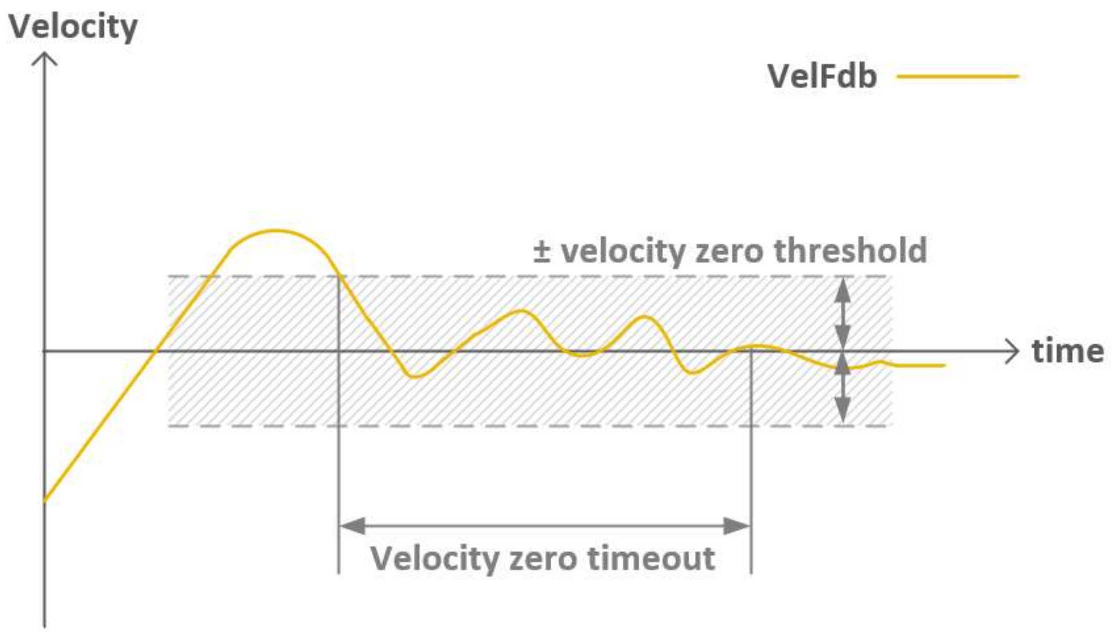
- The condition of velocity target reach flag and corresponding time.
- When the velocity feedback is within the reach window and continues for a period (reach timeout), the velocity target reach flag will rise.
- **When the time is set to 0, this detection function will be deactivated.**



Please check the Chapter 5.5.4 Target reach flag

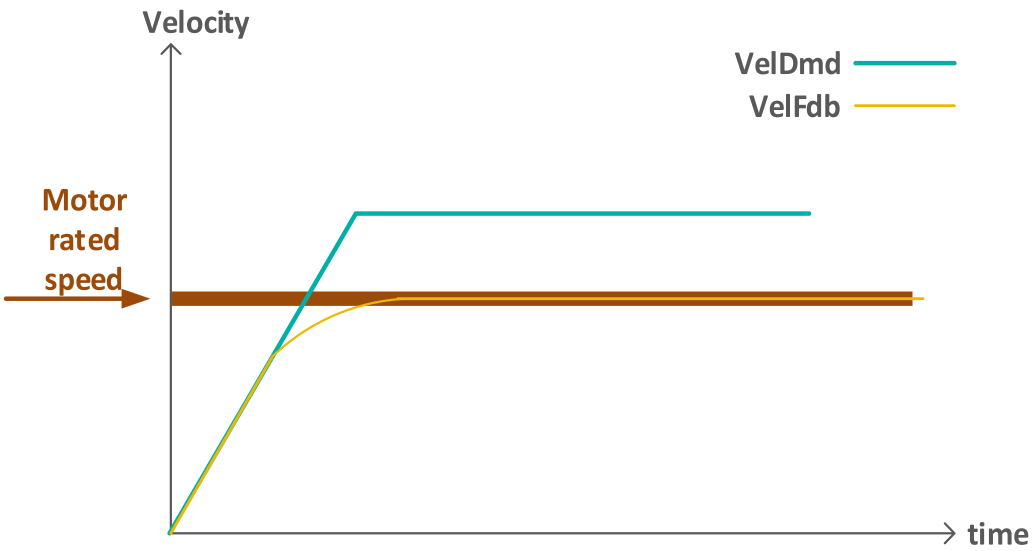
b. Velocity Zero Threshold / Timeout:

- When the velocity feedback is **within** the threshold and continues for a period (velocity zero timeout), such velocity feedback will be deemed as **zero**.
- **When the time is set to 0, this detection function will be deactivated.**



c. Motor Rated Speed:

The maximum speed allowed for the motor.



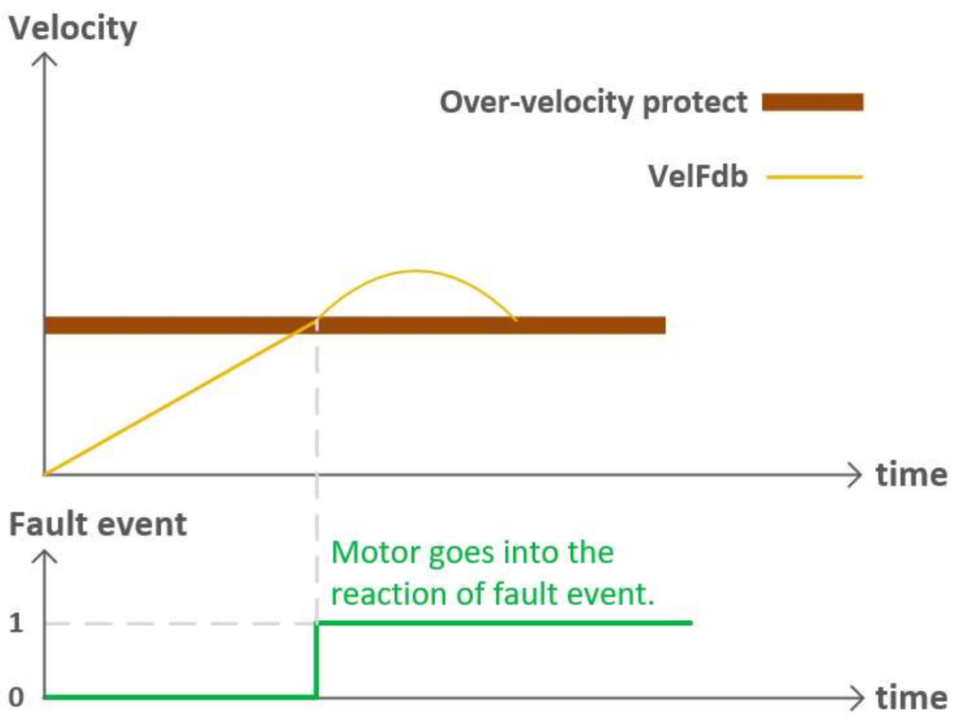
d. Quick Stop Deceleration:

Deceleration of the motor braking upon receiving the quick stop signal.

e. Over-Velocity Protect:

- The threshold of over velocity detection. If velocity exceeds over-velocity protect limit, the error code “**E8481**” will show up and **the motor will be turned off.**

- **If the value is set to 0, this detection function will be deactivated.**

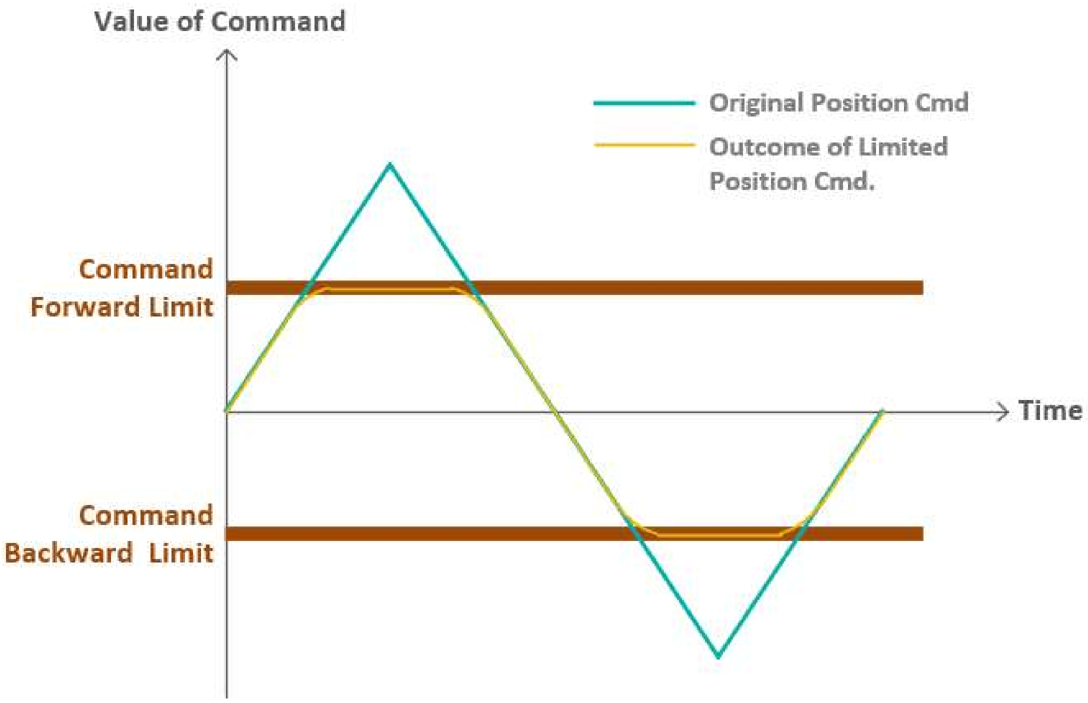


4.3.3 Position

Command Forward Limit	400.000	mm	Command Backward Limit	-250.000	mm
Over Pos. Fault Limit	500000.000	mm	Under Pos. Fault Limit	-500000.000	mm
Position Window	100	cnt	Position Window Timeout	5	ms
Following Error Window	5000	cnt	Following Error Window Timeout	0	ms

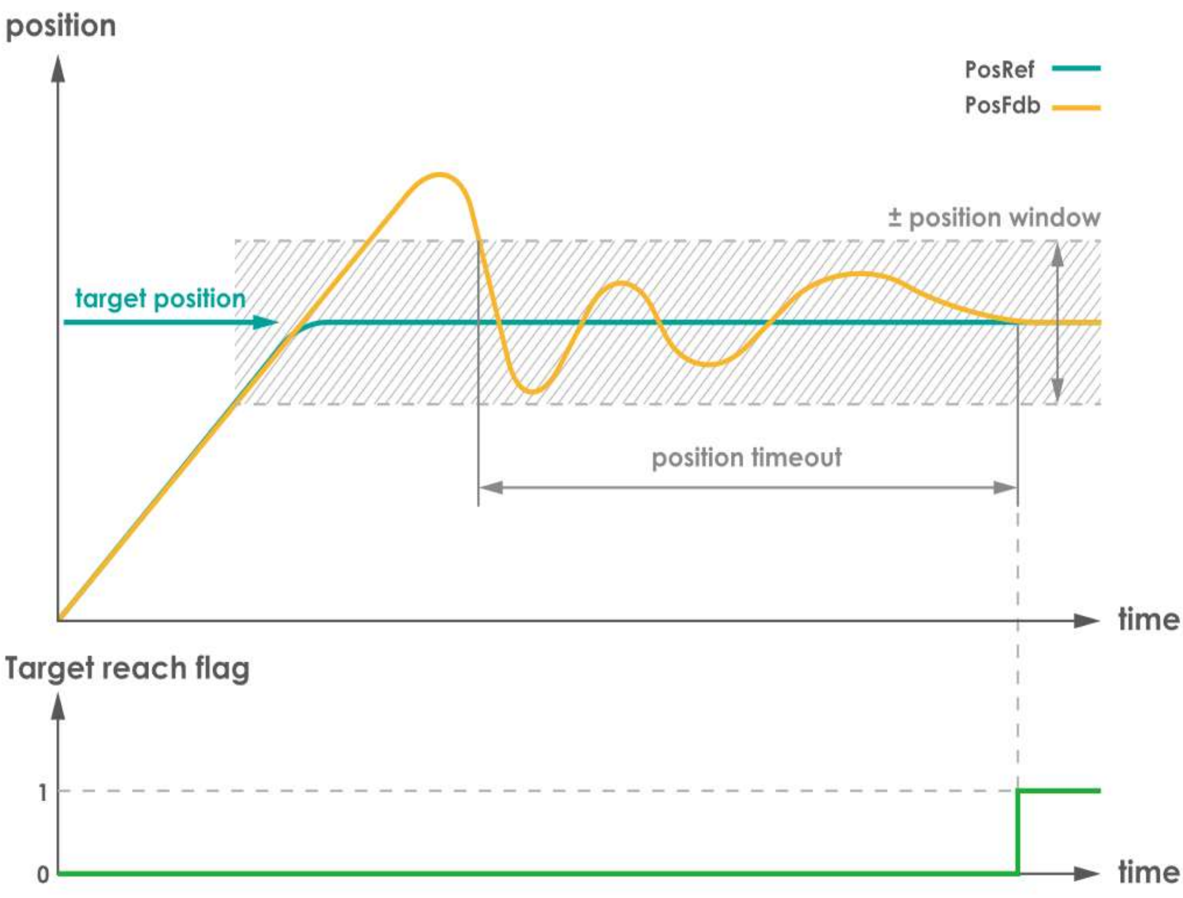
a. Command Forward / Backward Limit:

The maximum and minimum values of the position command that users can give.



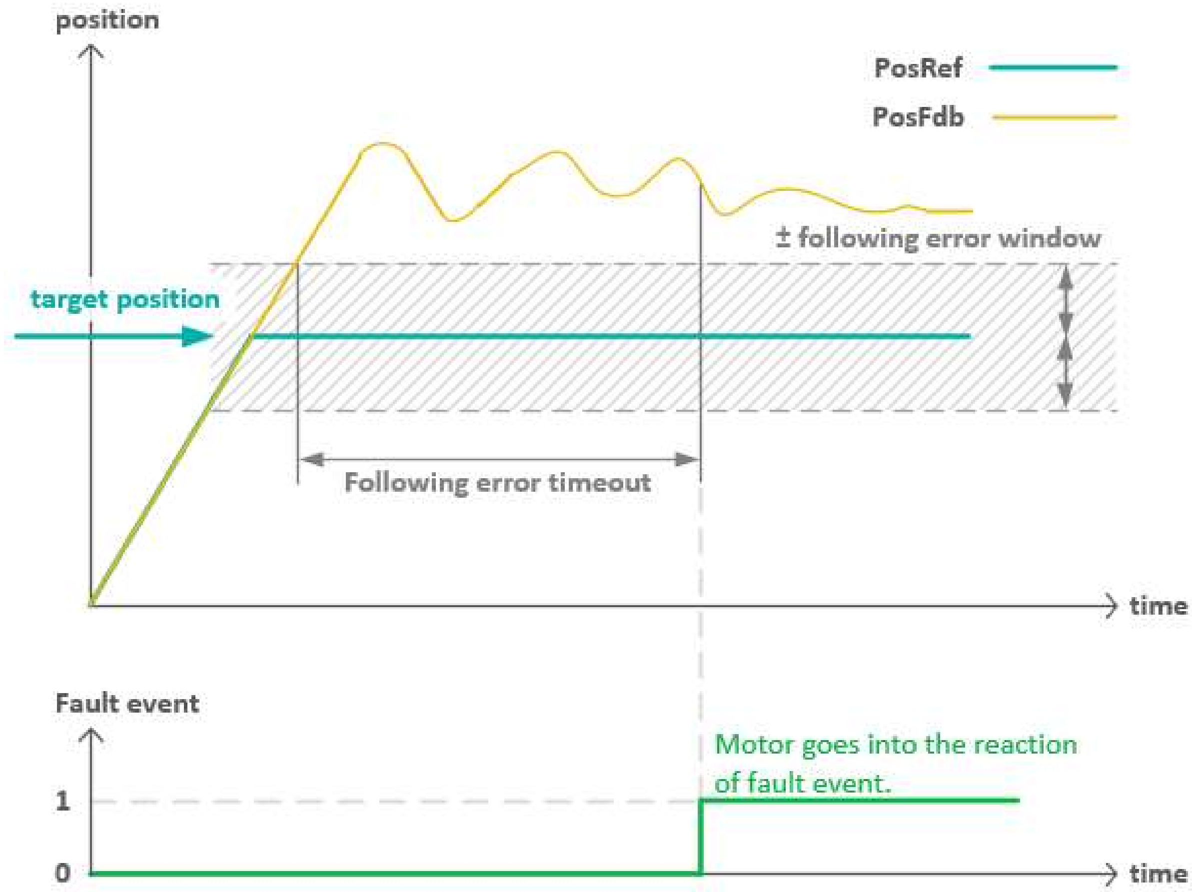
b. Position Window / Timeout:

- The condition of "position target reach" flag and the corresponding time.
- When the position feedback is **within** the position window and continues for a period (timeout), the target reach flag will **rise**.
- **When the time is set to 0, this detection function will be deactivated.**

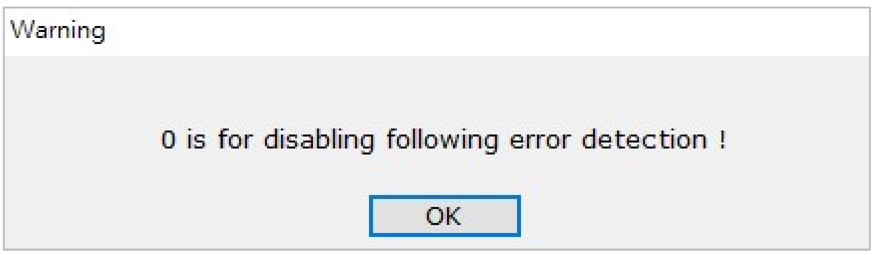


c. **Following Error Window / Timeout:**

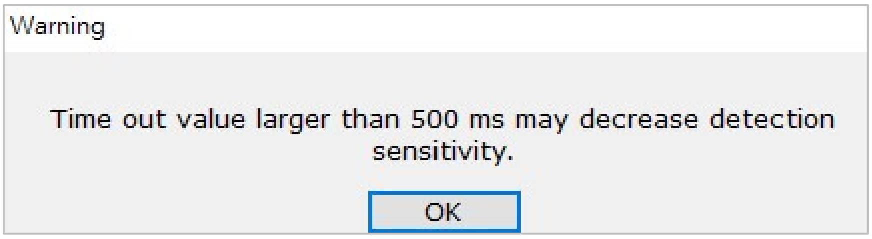
- The condition of "following error" flag and the corresponding timeout.
- The following error code is **E8611**.
- When the position feedback is **out of** the following error window and continues for a period (following error timeout), **a fault event will rise**. The motor will react according to user's setting of fault reaction.



- **When the time is set to 0, this detection function will be deactivated. For safety concern a warning will show up.**



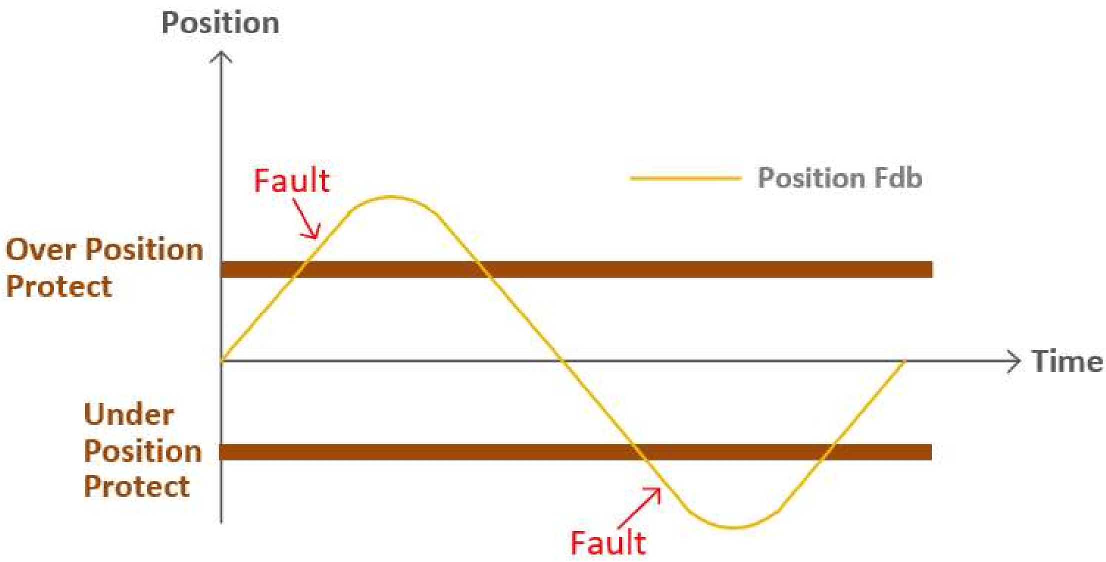
- **When the time is set to more than 500 ms, a warning will show up.**



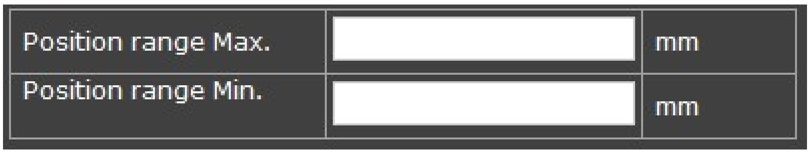
Please check the chapter 5.5.4 Target reach flag

d. Over / Under Position Protect:

The threshold of position-fault status. When motor position is **outside** the over/under range, **a fault event will occur**.



4.3.4 Modulo



Modulo is the Position Range. It also refers to the term “607Bh”, namely the “position range limit,” in CiA® 402 Draft Standard Proposal (see **chapter 4.2**, point 1 - (5)).

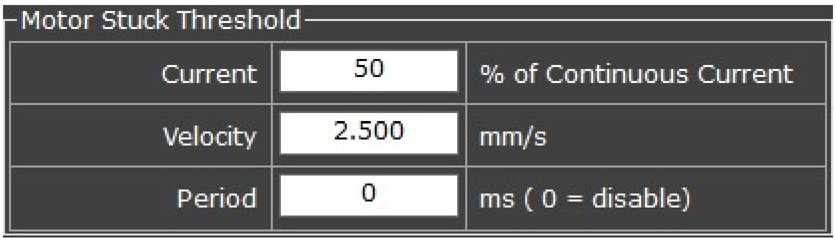
a. **Position range Max./Min.:**

Set user-defined Position Range here.

<u>Note</u>:

Some modes do not support user-defined parameters.

4.3.5 Motor Stuck



a. **Current:**

Sets the percentage of continuous current as the condition of “motor stuck”. If the motor requires more than this percentage of current to work, the motor will be considered **not moving (stuck)**.

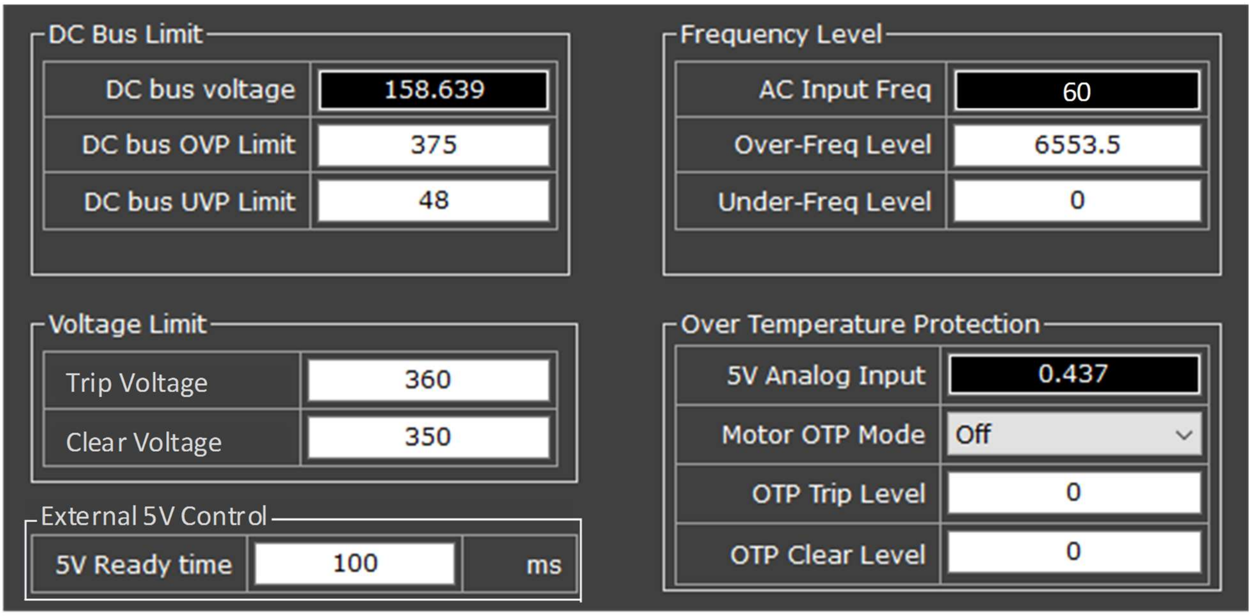
b. Velocity:

Velocity threshold. When velocity feedback is **lower** than this value, the motor will be considered **not moving** (i.e., motor stuck).

c. Period:

Set the time duration of Velocity threshold here. When velocity is lower than the Velocity threshold and continues for a period (Period), the motor will be considered not moving. **When the time is set to 0, this function will be deactivated.**

4.3.6 Power Stage



DC Bus Limit

This is for monitoring whether DC power supply is normal.

a. DC bus voltage:

monitors the DC bus voltage at present.

b. DC bus OVP Limit:

DC bus Over Voltage Protection Limit. Must be less or equal to **375 V**.

c. DC bus UVP Limit:

DC bus Under Voltage Protection Limit. Must be more than or equal to **48 V**.

Once DC power exceeds OVP limit or is lower than UVP Limit, this will become a **fault event** (Error code: **0x3210**; or **0x3220**); motor reactions will be as described in chapter 4.4 - Disable Option.

0x3210	DCLinkOverVoltage	DC capacitor over 375 V	Consider adding regenerate braking resistor
0x3220	DCLinkUnderVoltage	DC capacitor under 48V	Check high voltage supply

Voltage Limit

Settings of activating/deactivating the regenerative resistor.

d. **Trip Voltage:**

The trigger threshold of activating regenerative resistor. If voltage exceeds this value, regenerative resistor will be activated.

e. **Clear Voltage:**

Threshold of clearing the hysteresis effect. If voltage decreases to under this threshold, regenerative resistor will be deactivated.

External 5V Control

f. **5V Ready time:**

The time duration from starting supplying 5V power to when the equipment needing the 5V supply is ready for use. (default value is 100 ms).

Frequency Level

This is for monitoring whether **AC power supply** is normal, and/or if AC power is connected.

g. **AC Input Frequency:**

Monitors the AC power.

h. **Over-Freq Level:**

Over-Frequency Level.

i. Under-Freq Level:

Under-Frequency Level.

Once AC power exceeds Over-Freq Level or is lower than Under-Freq Level, it will become a fault event (when lower, error code is **0x3142**); motor reactions will be as described in the Disable Option chapter

Over Temperature Protection

The 5 V Analogue Input here is a general-purpose pin which is used to connect with the thermistor on the motor. GUI will show the monitored voltage on drive input.

As thermistors vary, users will need to calculate the resistance (ohm) according to the monitored voltage and then derive the corresponding temperature; see example formula [note*] and diagram below.

j. 5 V Analog Input:

Shows the voltage at the 5V Analogue Input pin.

k. Motor OTP Mode:

Motor Over Temperature Protection Mode.

- **Off:**

  turn off protection mode.

- **GreaterThan**:

  This option is for PTC thermistors. When voltage on drive input is **greater** than the "OTP Trip Level" value (a fault event), protection will be triggered.

- **LessThan**:

  This option is for NTC thermistors. When voltage on drive input is **less than** the "OTP Clear Level" value (a fault event), protection will be triggered.

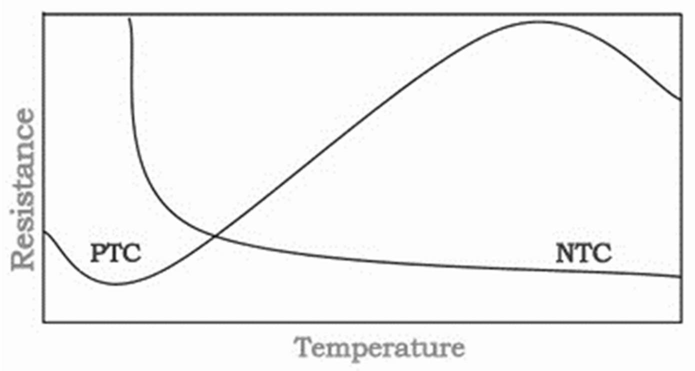
l. OTP Trip Level:

The voltage threshold value to trigger protection.

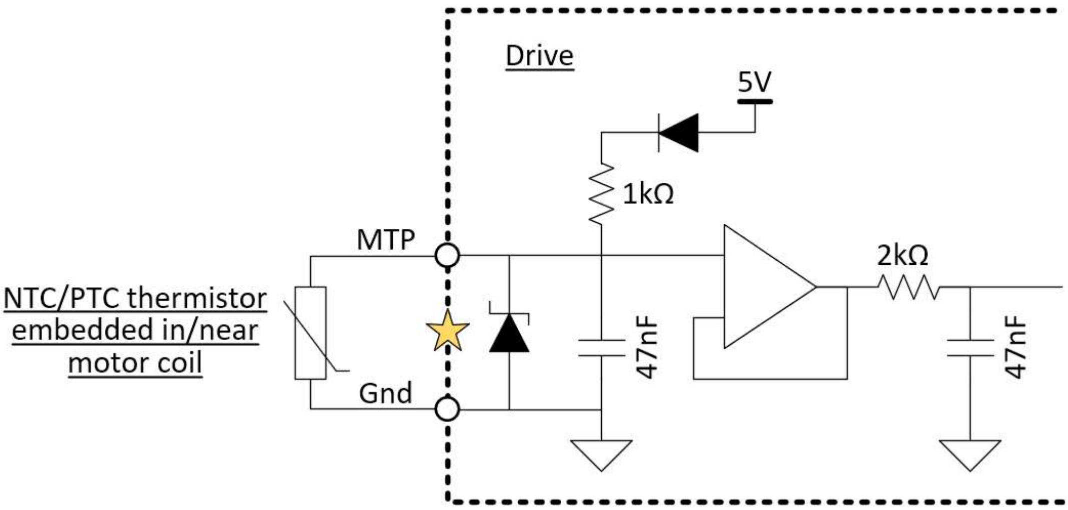
m. OTP Clear Level:

The voltage threshold value to deactivate protection.

< Thermistor—Diagram of Temperature & Resistance >



<Equivalent Circuit Diagram>



Note*:

The formula to acquire the resistance (ohm) of place ☆ :

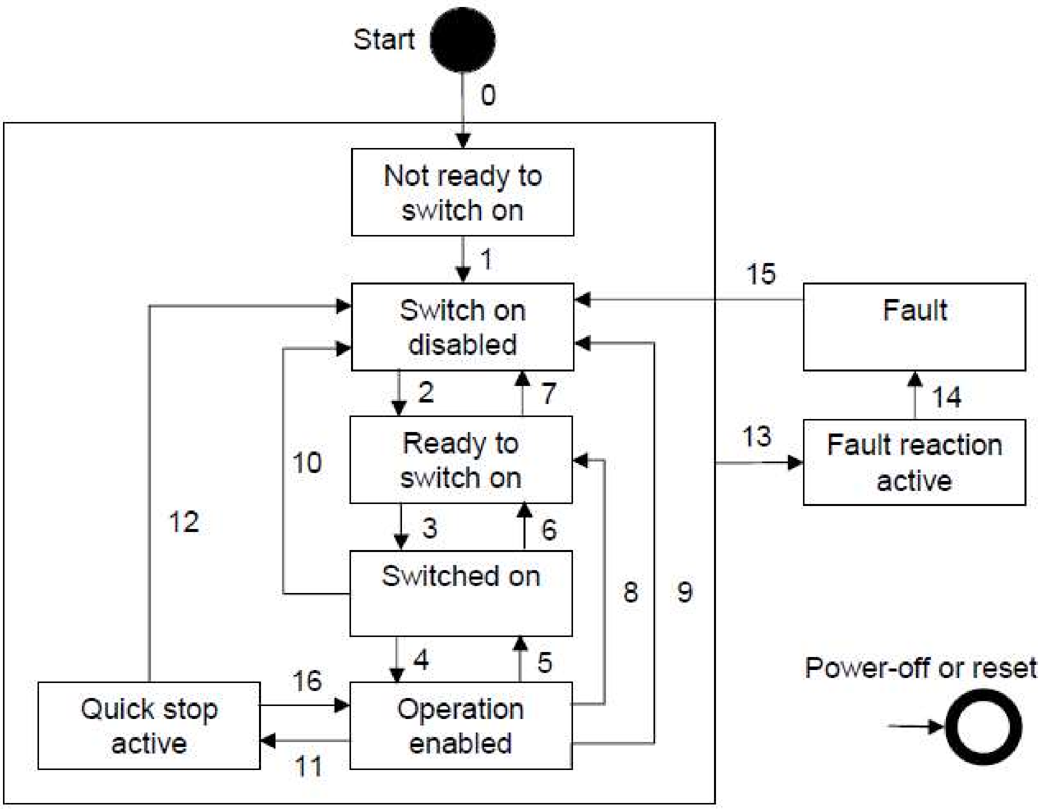
(The voltage of place ☆ is known, monitored by the UI.)

$$V = \frac{5R}{R + 1000} \quad , \quad R = \frac{1000V}{5 - V}$$

4.4 FSA & Disable Option

4.4.1 Finite State Automata

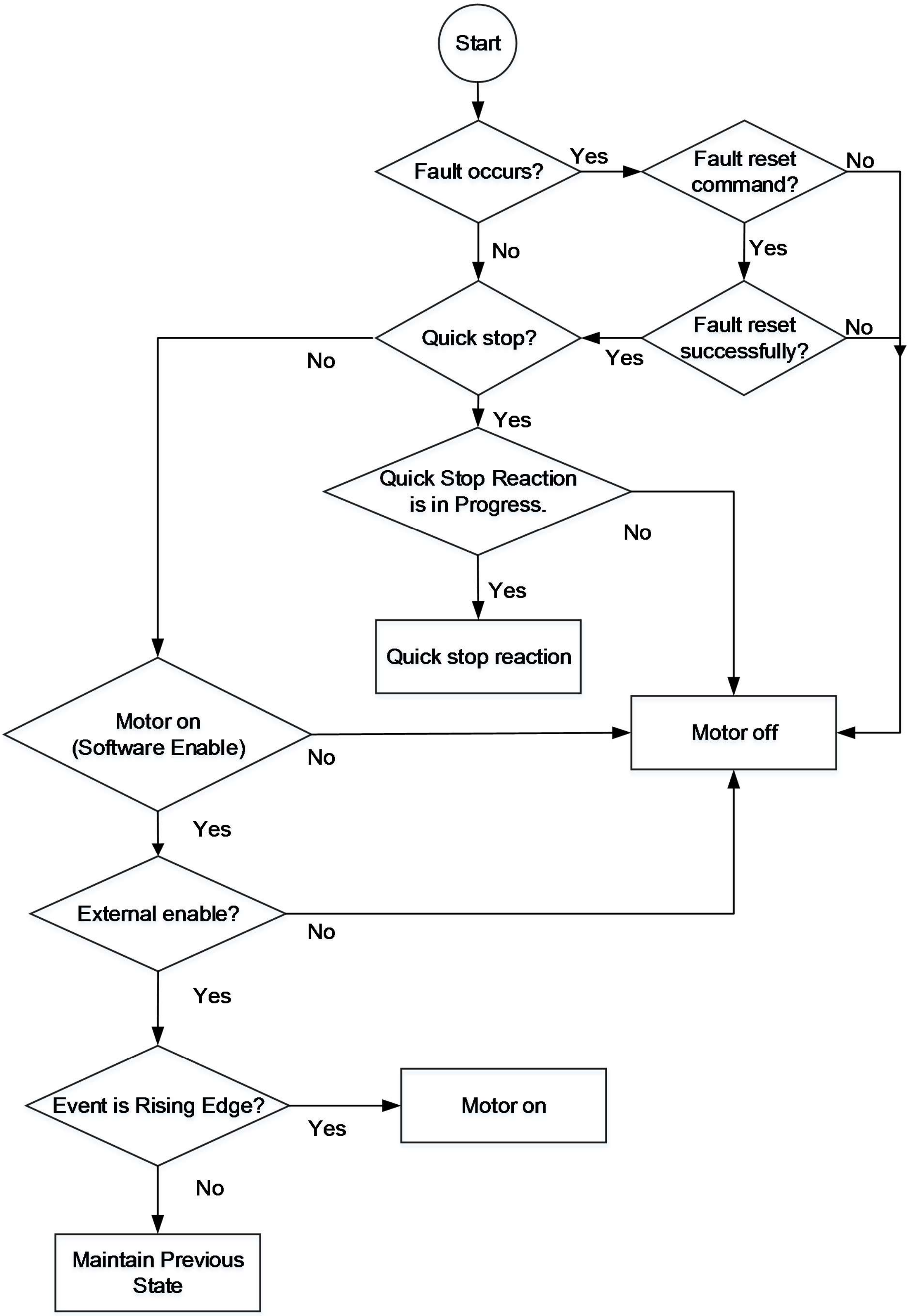
1. See the diagram below as described in CiA 402.



2. See the chart below from CiA 402 describing the triggering Event and Actions.

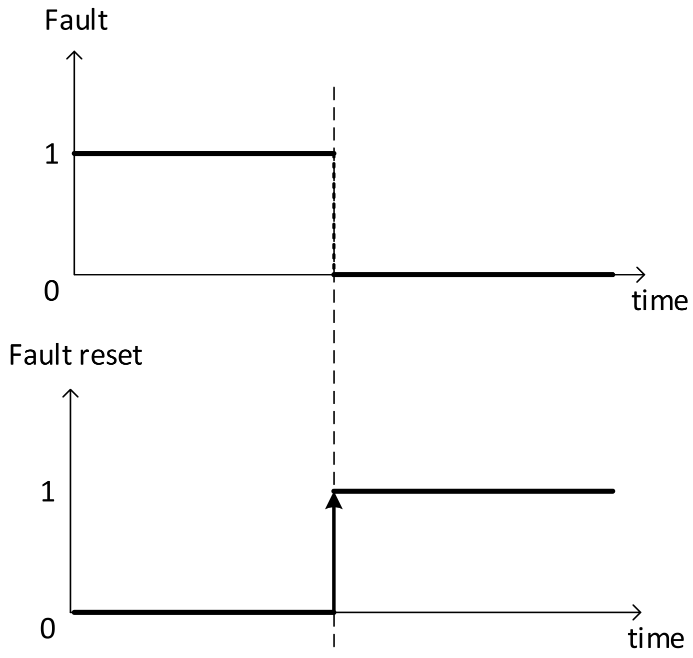
Transition	Event(s)	Action(s)
0	Automatic transition after power-on or reset application	Drive device self-test and/or self initialization shall be performed.
1	Automatic transition	Communication shall be activated.
2	Shutdown command from control device or local signal	None
3	Switch on command received from control device or local signal	The high-level power shall be switched on, if possible.
4	Enable operation command received from control device or local signal	The drive function shall be enabled and all internal set-points cleared.
5	Disable operation command received from control device or local signal	The drive function shall be disabled.
6	Shutdown command received from control device or local signal	The high-level power shall be switched off, if possible.
7	Quick stop or disable voltage command from control device or local signal	None
8	Shutdown command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
9	Disable voltage command from control device or local signal	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
10	Disable voltage or quick stop command from control device or local signal	The high-level power shall be switched off, if possible.
11	Quick stop command from control device or local signal	The quick stop function shall be started.
12	Automatic transition when the quick stop function is completed and quick stop option code is 1, 2, 3 or 4, or disable voltage command received from control device (depends on the quick stop option code)	The drive function shall be disabled, and the high-level power shall be switched off, if possible.
13	Fault signal (see also /CiA402-3/)	The configured fault reaction function shall be executed.
14	Automatic transition	The drive function shall be disabled; the high-level power shall be switched off, if possible.
15	Fault reset command from control device or local signal	A reset of the fault condition is carried out, if no fault exists currently on the drive device; after leaving the fault state, the fault reset bit in the controlword shall be cleared by the control device.
16	Enable operation command from control device, if the quick stop option code is 5, 6, 7, or 8	The drive function shall be enabled.
It is not recommended to support transition 16.		

3. Diagram of cpc's Motor Activation Logic.

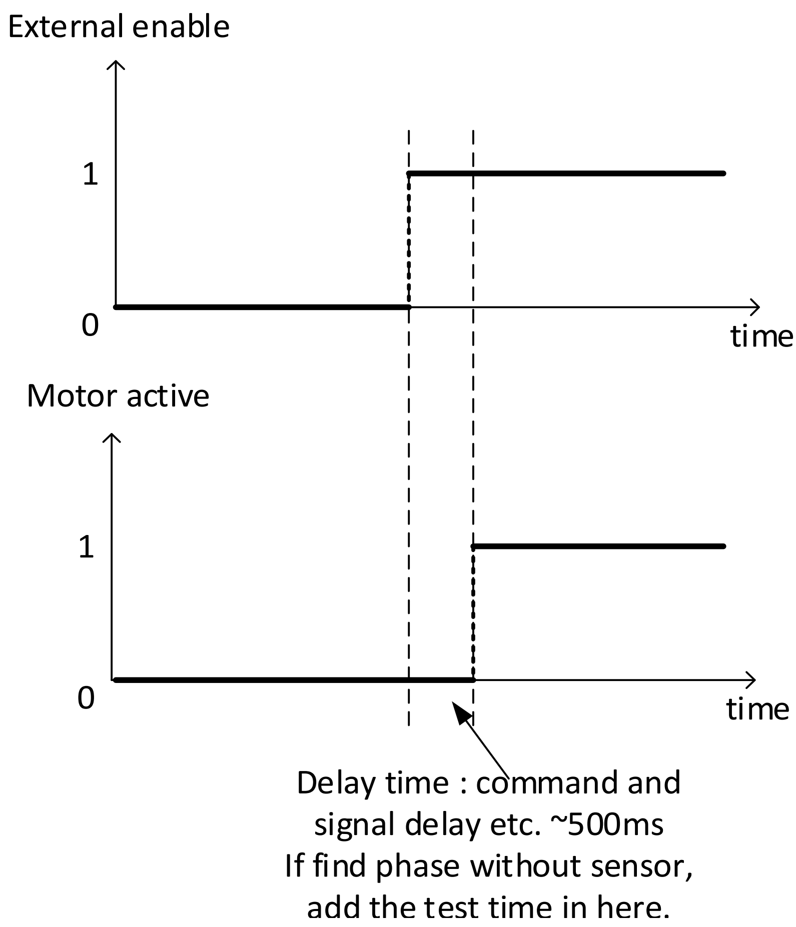


4. Fault event timing as follows:

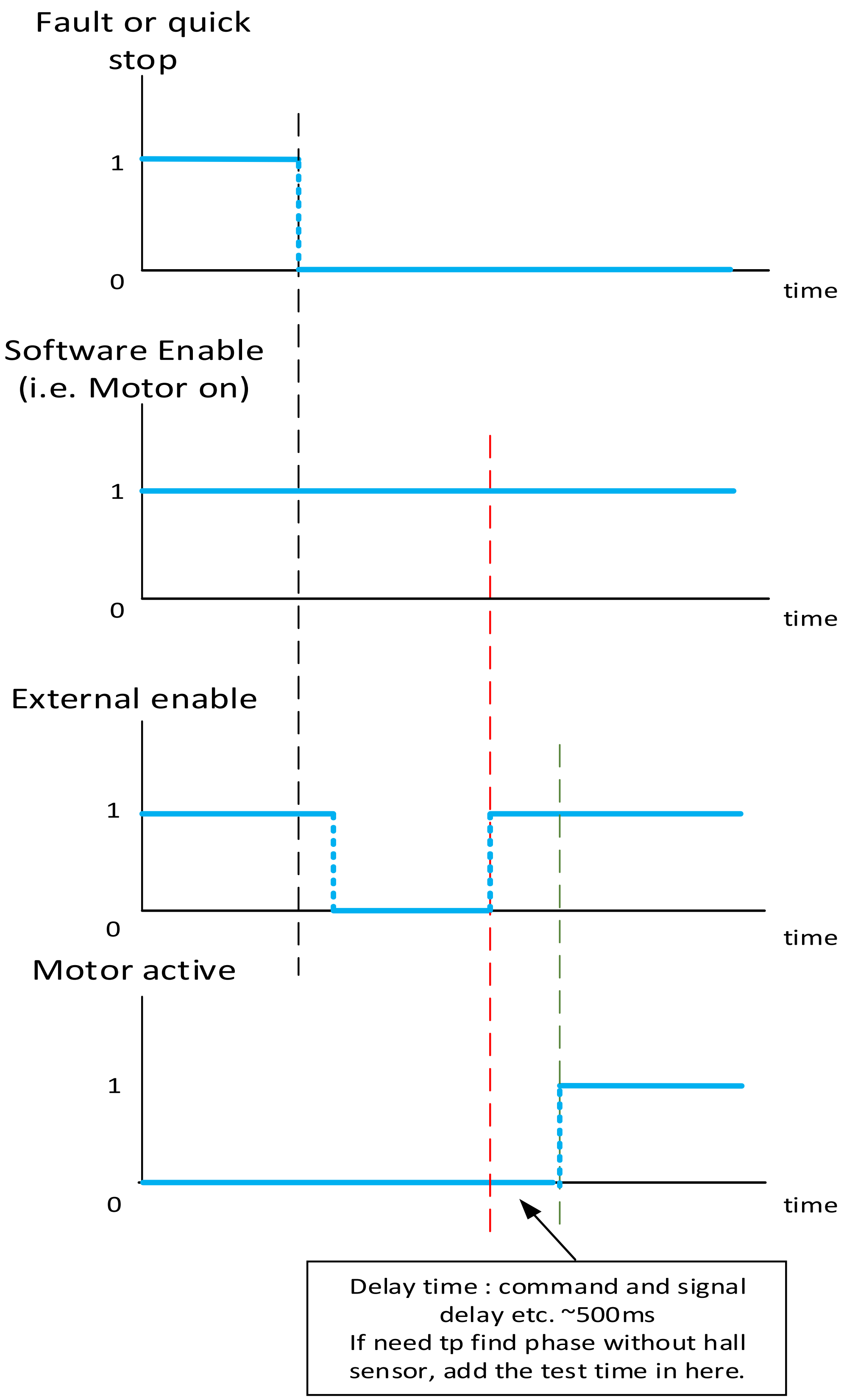
Note: If Fault conditions are not cleared, reset cannot be activated.



5. Motor event timing, as follows:



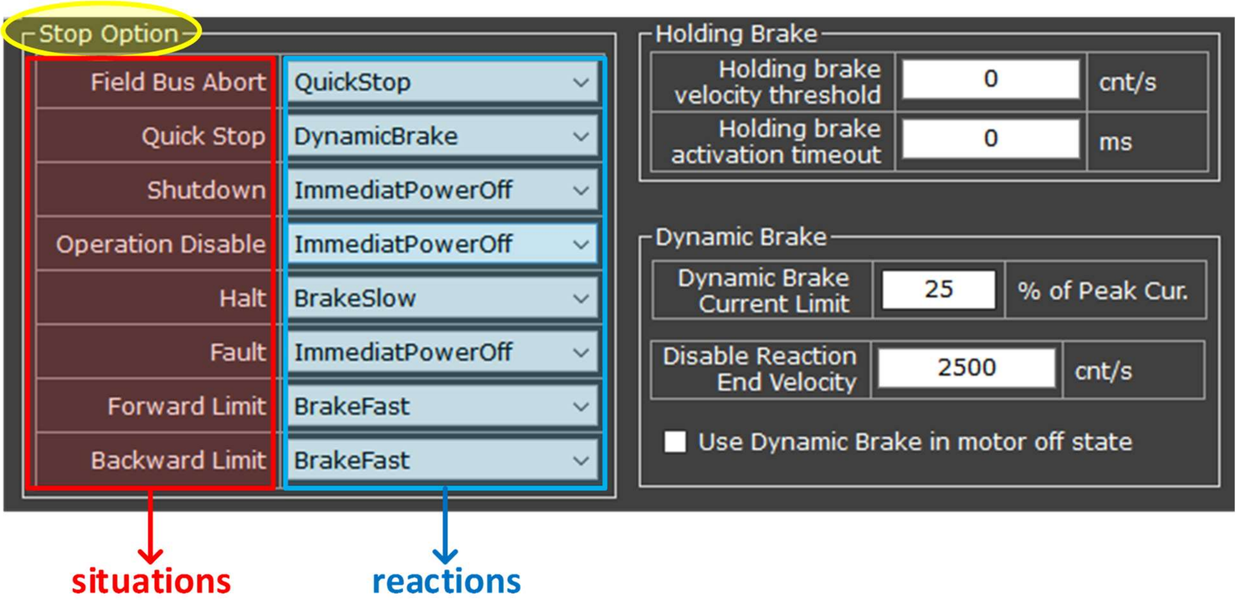
6. **If users want to turn on the motor again after fault or quick stop occurs, only rising edge of external enable can do, see the diagram below.**



4.4.2 Disable Option

Users can assign what **reaction** to perform when the **situations** listed on Stop Option panel occur.

On Stop Option panel, the situations from “Filed Bus Abort” to “Fault” correspond to the CiA® 402 DSP.



For further explanation about

- **Situations**, see chapter 4.4.2.1;
- **Reactions**, see chapter 4.4.2.2;
- **Dynamic Brake**, see chapter 4.4.2.3;
- **Holding Brake**, see 4.4.2.4

4.4.2.1 Situation Explanation

This chapter explains what causes situations “a” to “g” and shows the corresponding options of reaction.

a. Field Bus Abort:

This “Field Bus Abort” situation is due to that CANopen or EtherCAT stops broadcasting (usually because net cable is disconnected); therefore, the driver gives a command to itself to execute a reaction.

For example, if the reaction of Field Bus Abort is set to “FaultEvent”, when Field Bus Abort occurs, the driver will trigger a fault event and execute the designated reaction of Fault situation (situation “f”).

Stop Options		
Situation		**Reaction**
a.	**Field Bus Abort**	No action
		FaultEvent
		DisableVoltage
		QuickStop

Stop Options		
Situation		**Reaction**
f.	**Fault**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow
		BrakeFast
		BrakeCurrent
		BrakeVoltage

b. Quick Stop:

Quick Stop is triggered by controlword from control device, by physical switch, or when field bus is aborted (see Quick stop signal setting in Chapter 4.7 Input & Output).

Stop Options		
Situation		**Reaction**
b.	**Quick Stop**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow
		BrakeFast
		BrakeCurrent
		BrakeVoltage

c. Shutdown:

A situation of when AC power is shut down.

Stop Options		
Situation		**Reaction**
c.	**Shutdown**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow

d. Operation Disable:

When "disable command" is received from control device or local signal.

Stop Options		
Situation		**Reaction**
d.	**Operation Disable**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow

e. Halt:

Pause.

See Halt signal setting in Chapter 4.7 Input & Output.

Stop Options		
Situation		**Reaction**
e.	**Halt**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow
		BrakeFast
		BrakeCurrent
		BrakeVoltage

f. Fault:

When Fault event is triggered.

Stop Options		
Situation		**Reaction**
f.	**Fault**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow
		BrakeFast
		BrakeCurrent
		BrakeVoltage

g. Forward & Backward Limit:

When Forward/Backward limit switch is triggered.

See Forward & Backward limit signal setting in Chapter 4.7 Input & Output.

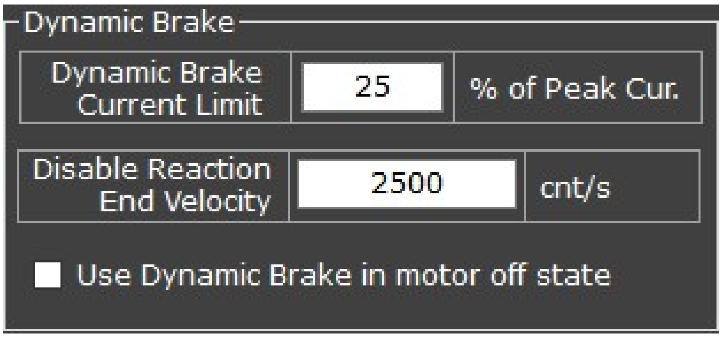
Stop Options		
Situation		**Reaction**
g.	**Forward & Backward Limit**	DynamicBrake
		ImmediatePowerOff
		BrakeSlow
		BrakeFast

4.4.2.2 Reaction Explanation

Name of the Reactions to Field Bus Abort	Explanation
No action	Event will be ignored.
FaultEvent	To trigger fault event.
DisableVoltage	Transit to "Switch on disabled" state.
Quick Stop	To trigger quick stop

Name of Reaction	Explanation
QuickStop	Transit to "Quick stop active" state.
DynamicBrake	Brake motor by means of a controlled motor short-circuit.
ImmediatePowerOff	Electrically disconnected from drriver.
BrakeSlow	Depends on presently used mode: If in ■ **Velocity Mode**: Bring velocity command to zero using Profile deceleration. ■ **Current(Torque) Mode**: Bring current command to zero using Profile current slope.
BrakeFast	Depends on presently used mode: If in ■ **Velocity Mode**: Bring velocity command to zero using Quick Stop deceleration. ■ **Current(Torque) Mode**: Bring current command to zero by powering off (that is to say: ImmediatePowerOff).
BrakeCurrent	Control stop to zero current using Torque mode's current slope.
BrakeVoltage	equals to ImmediatePowerOff.

4.4.2.3 Dynamic Brake



Brake motor by means of a controlled motor short circuit. This function can be applied **both during motor-on and motor-off states**, and can be used to:

1. Decrease the degree of unexpected motor reactions if the encoder is not functioning correctly.
2. Stop motor immediately.
3. Prevent motor from gliding freely even in motor-off state.

a. Dynamic Brake Current Limit % of Peak Cur.

Use how much percentage of peak current to brake the motor. This value will influence those stop options which are set to "Dynamic Brake".

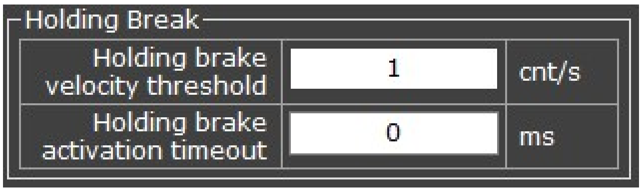
b. Disable Reaction End Velocity

When a disable reaction (in this case, the Dynamic Brake reaction) is in process, once motor speed drops below this value, end the disable reaction and transit motor state to next state.

c. Use Dynamic Brake in motor-off state

Tick this box if users wish to apply dynamic brake in motor off state also.

4.4.2.4 Holding Brake



This function is used in motor off state.
To apply this function, **a physical brake needs to be installed** on the motor system. Besides, users need to set a digital output pin as "Brake".

a. **Holding Brake Velocity Threshold:**
In motor off state, brake the motor when velocity is **below** this value cnt/s.

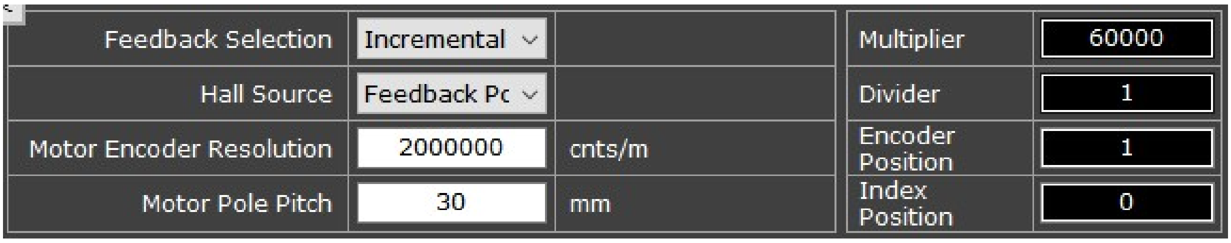
b. **Holding Brake Activation Timeout:**
In motor off state, brake the motor while motor has kept moving after a certain period (Timeout).

4.5 Feedback

In addition to as described (such as setting the encoder type, source of hall sensor, and the unit of encoder) in the Wizard chapter, please see detailed configuration on the subsequent pages.

Select first the type of your encoder ("Feedback Selection" on the panel). There are 4 types supported—Incremental Encoder A/B, Sine/Cosine Incremental, BiSS-C, and Tamagawa.

The Feedback panel's buttons are shared in all encoder types:



a. Feedback Selection:

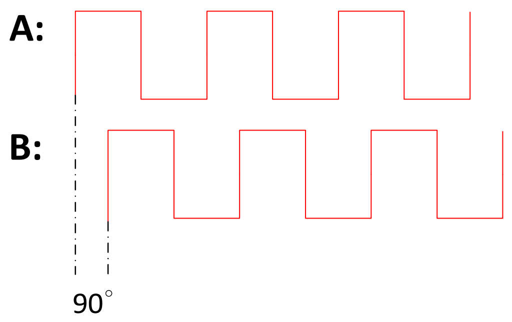
At present various signal types of encoder can be supported by TC1 Drivers:

- TC1 Drivers:

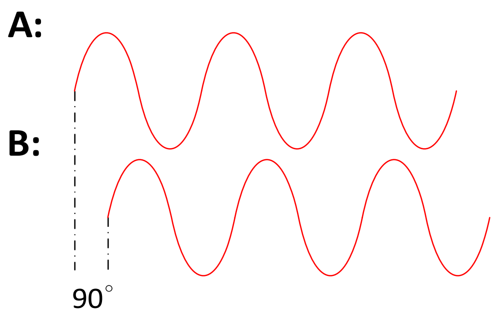
  supports Incremental A/B.
- TC1-B Drivers:

  supports incremental A/B, Sin/Cos, BiSS-C, and Tamagawa signals.

- Incremental Encoder A/B:



- Sine/Cosine Encoder:



b. Hall Source

- Feedback Port:

  Assign the digital input pins at Feedback port as hall sensor signal source.

- Controller Port:

  Assign the digital input pins at Controller port as hall sensor signal source.

c. Motor Encoder Resolution:

- Set the resolution of the encoder sensor (in cnts/m or cnts/rev).

  Note:

  The resolution of **sine/cosine encoder** = Interpolation Factor x Fundamental Cycle Count. See **chapter 4.5.2**.

d. Motor Pole Pitch:

- Rotary:

  Define how many sets of poles are in one revolution.

  (For firmware version starting **0.7.21**): **If unknown**, enter value **65535**, then go to Tune> Phase> Auto Phase and click "Start AutoPhase" button, the UI will test for you automatically.

- Linear:

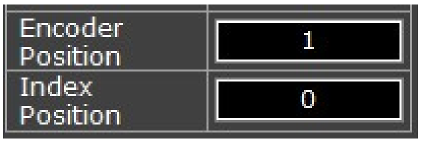
  Set the length of one set of pole of the motor (in millimeter).

e. Multiplier / Divider:

This ratio value is for firmware's use.

However, it also gives users an idea of how many counts per pole pair.

f. Encoder/Index Position:

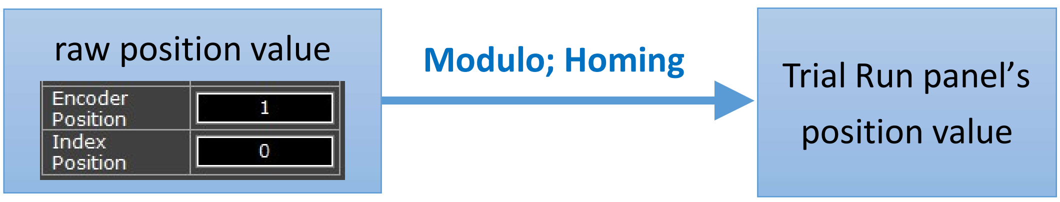


The **raw** position value of the encoder and index.

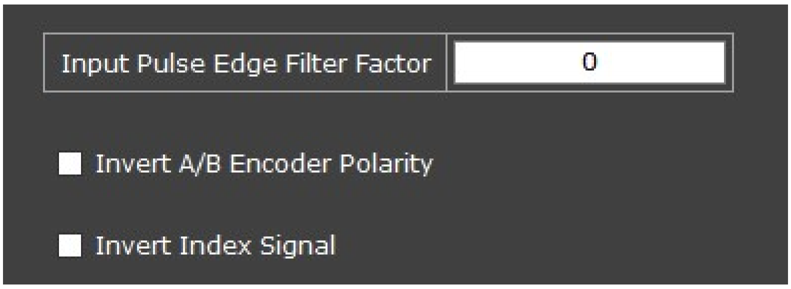
Note:

The position values shown on this panel might differ from the position feedback value shown on Trial Run panel.

The position feedback value on Trial Run panel has been processed by **modulo** (see chapter 4.3.4) and **homing** (see chapter 7).



4.5.1 Incremental Encoder A/B



a. Input Pulse Edge Filter Factor:

- **A length-of-time** **factor** (to be multiplied by the rest part of the formula mentioned below) **to produce the total length of time of a filter window which is used as the threshold to filter off the glitch in input signal.** This value can be **0 to 255**. "Input signal" means, for example, the signals sent from the motor encoder to the driver (value 0 = function disabled). The cpc driver provides such **pulse-width filter** function on such signal.

- **Formula**—Total length of time of Filter Window:

$$T_{FW} = \textbf{Factor} \times 10 \times \frac{1}{Fc\ (note*)}\ (ns)$$

- **Encoder's maximum detectable frequency**:

$$F_S = \frac{Y\ (note*)}{\text{Factor} \times 10 \times 4}\ (Hz)$$

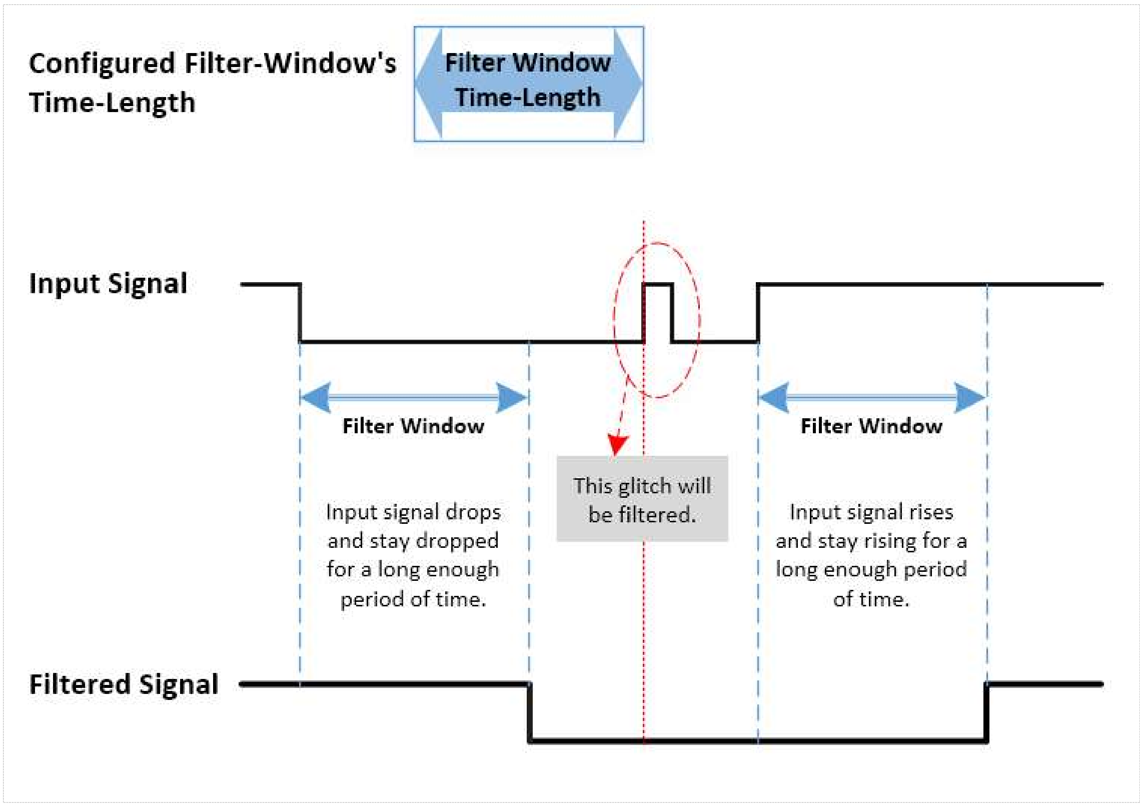
Note*:

Drive	**value of "Fc"**
TC1 Series	150,000,000
TC1-B Series	160,000,000

- When receiving a signal, the driver determines whether it is a glitch based on the user-configured total length of time of the Filter-Window.
  - ➢ If the input signal's duration is **shorter** than Filter Window, the signal is regarded **as a glitch** and will be ignored.

➢ If the input signal's duration is longer than or equals to the Filter Window, it is regarded as a true signal transition and will be displayed.

<Mechanism of Filter-Window>



b. Invert A/B Encoder Polarity

- Reverses the direction of encoder counting.

  To elaborate, if originally the motor's position feedback value increases while moving in positive direction, after ticking (or unticking) this option, the value will decrease instead.

  This function could be applied when AutoPhase is unusable—one of the possible causes is that the motor stroke cannot contain a complete motor pole pitch; to solve this issue, see manual-phase-setting solution in **chapter 5.2.4**.

- **Note**:

  Inverting encoder polarity will cancel the presently-established Phase and Home. **See SAFETY WARNING on next page.**

- **SAFETY WARNING**:

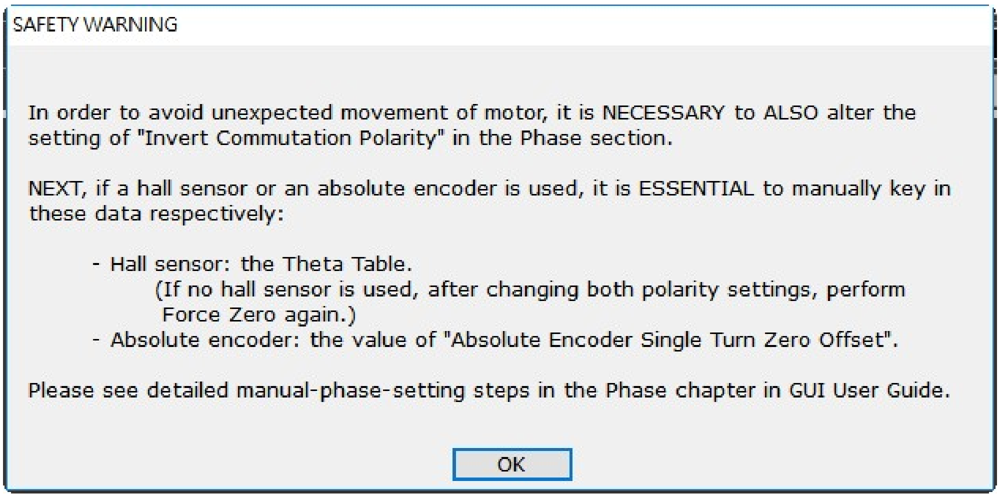
  To avoid unexpected movement of the motor, after changing the encoder polarity setting, it is NECESSARY to ALSO alter the "Invert Commutation Polarity" setting in the Phase section.

  Next, if a hall sensor or an absolute encoder is used, it is ESSENTIAL to manually key in these data respectively:

  - **Hall sensor**: The Theta Table. (If no hall sensor is used, after changing BOTH polarity settings, perform Force Zero again.)
  - **Absolute encoder:** the value of "Absolute Encoder Single Turn Zero Offset".

  Please see detailed steps of manual-phase-setting in chapter 5.2.4.

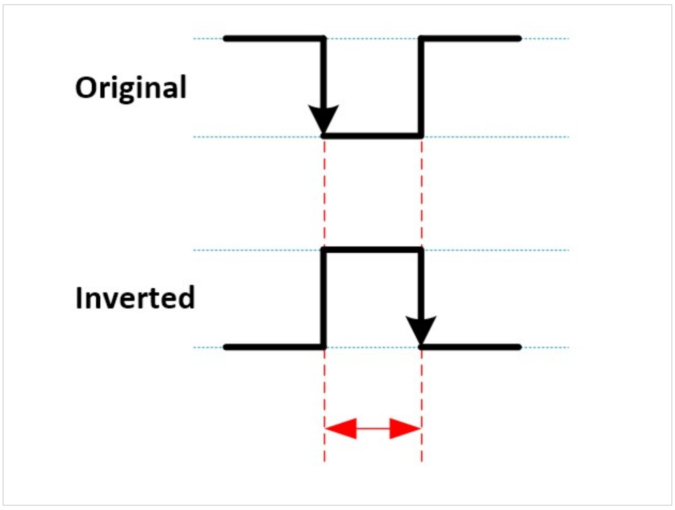
A warning will show up when trying to change the polarity.



c. Invert Index Signal

Inverts the polarity of index input signal.

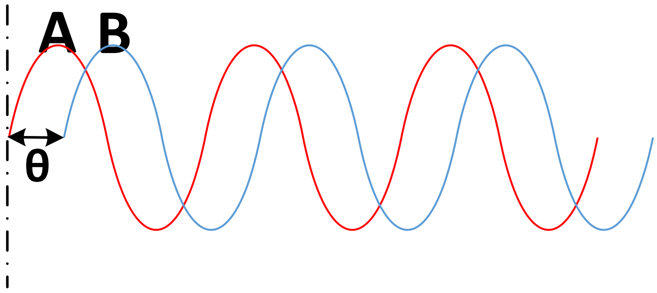
<Inverted index signal>



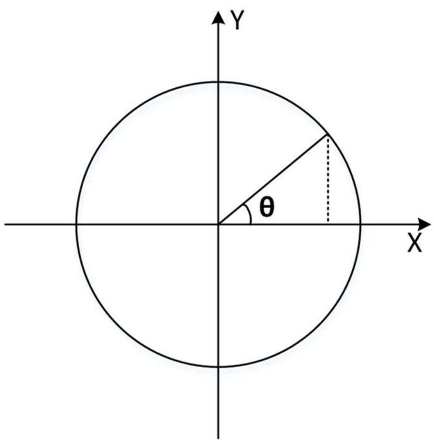
4.5.2 Sine/Cosine Encoder

Principle:

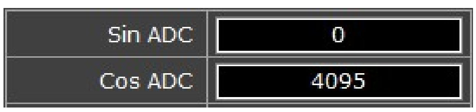
The sine/cosine encoder sends out two analog signals which are in sine wave forms and are **90 degrees out of phase**.



The sine and cosine signals can be drawn into a circle based on X (sine) and Y (cosine) co-ordinates and can therefore perform arctangent function (or, interpolation) to know the degree of **theta** in order to obtain finer position resolution.

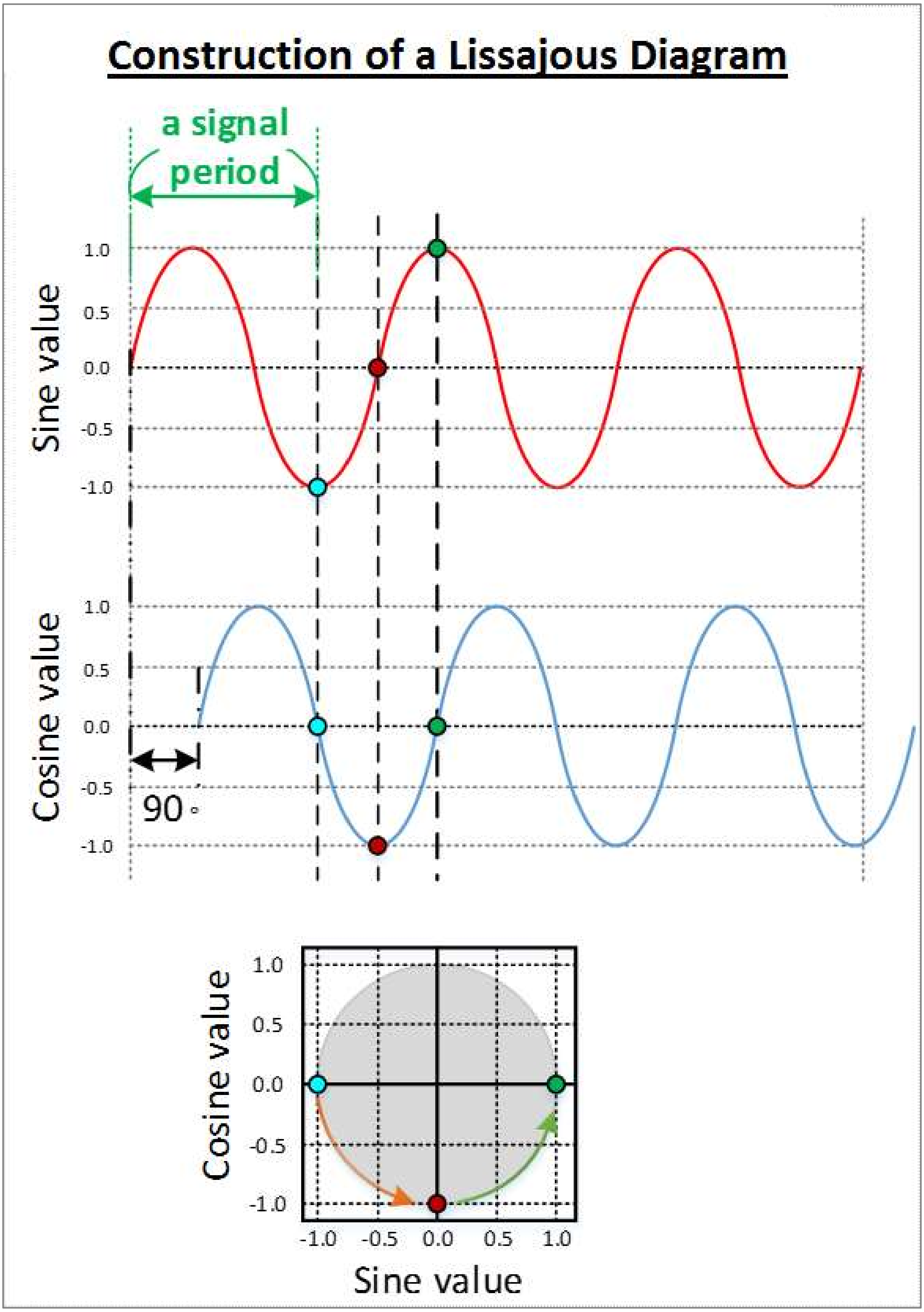


The cpc driver will digitalize sine and cosine signals into values ranging from **0 to 4095**.



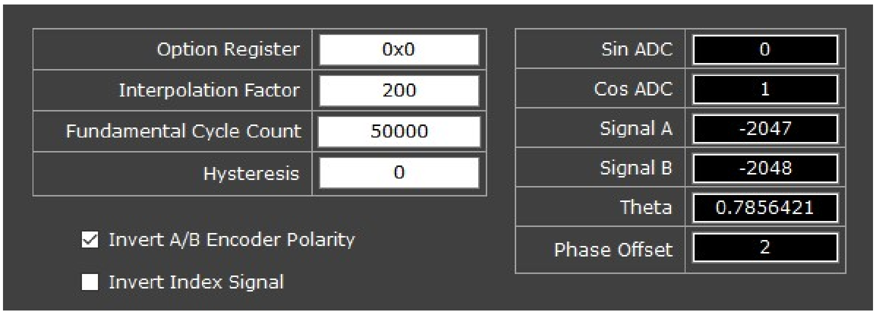
A sine/cosine circle can be viewed in **Lissajous** diagram (i.e., plotXY diagram) in the Scope window. Furthermore, there are 3 kinds of possible distortion in signals: Amplitude, Sine/Cosine Offset, and Phase. These will be elaborated on subsequent pages.

<Lissajous Diagram>

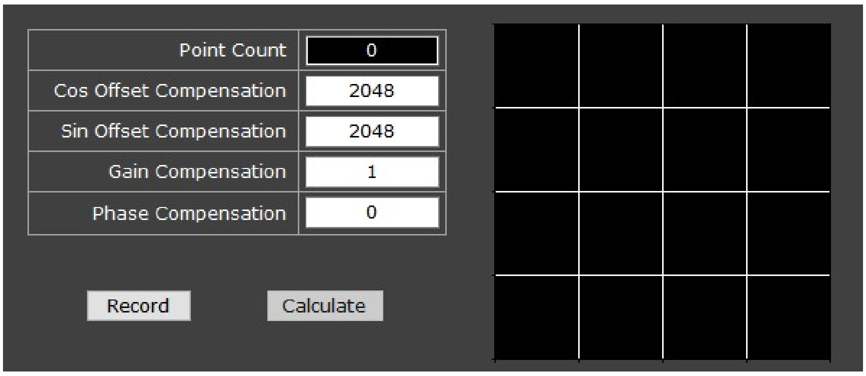


There are two function tabs under the sine/cosine encoder section:

- Parameter tab

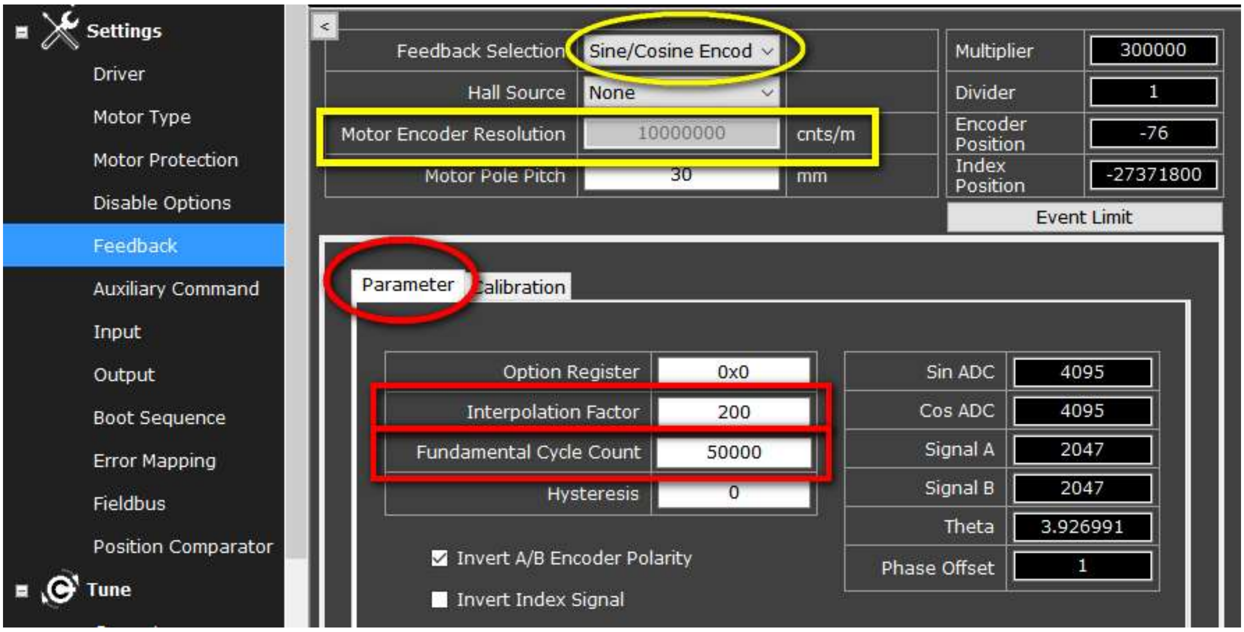


- Calibration tab

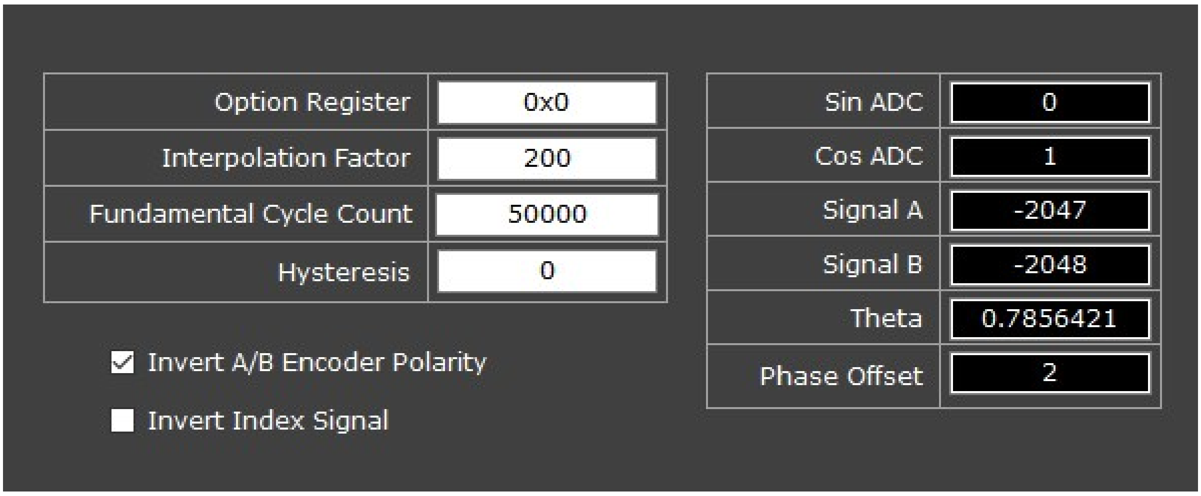


Note:

The value of "Motor Encoder Resolution" of the sine/cosine encoder is the result multiplying parameters "Interpolation Factor" and "Fundamental Cycle Count".



4.5.2.1 Parameter Tab



a. **Option Register**:

- 0x0:

  no use.

- 0x1:

  Averages the sine and cosine signals happened in one control loop cycle time. This function is for **filtering off noises**; the values of signal A, signal B and Theta are hence changed.

  Note:

  This function is **not** suitable for high resolution encoder. The sine/cosine wave frequency needs to be **no greater than 1/4 of the control loop frequency**.

b. **Interpolation Factor**:

Indicates how many counts per signal period.

c. **Fundamental Cycle Count**:

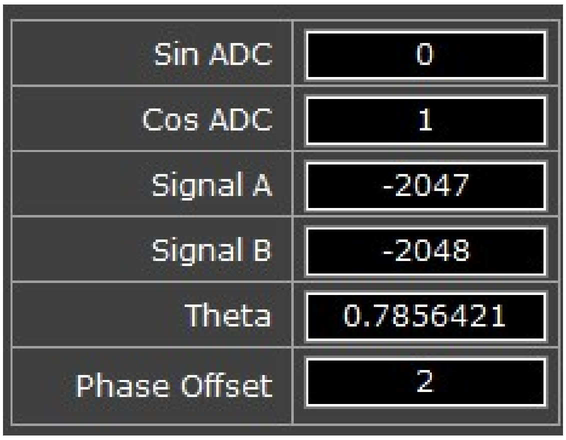
- Rotary motor: how many signal periods per cycle.
- Linear motor: how many signal periods per meter.

d. Invert A/B Encoder Polarity:

Same as described in **chapter 4.5.1, point (b)**.

e. Invert Index Signal

Same as described in **chapter 4.5.1, point (b)**.



f. Sin ADC:

The real time digitalized value of sine signal.

Value range: 0 to 4095.

g. Cos ADC:

The real time digitalized value of cosine signal.

Value range: 0 to 4095.

h. Signal A:

The outcome value of Sin ADC minus Sin Offset Compensation.

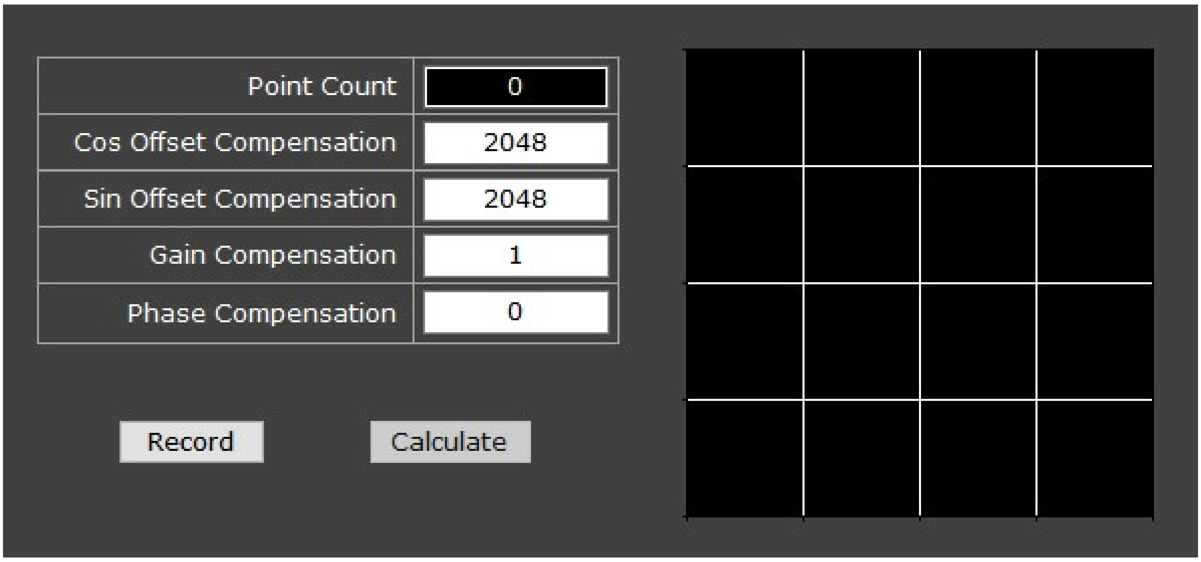
i. Signal B:

The outcome value of Cos ADC minus Cos Offset Compensation.

j. Theta:

The electrical theta within signal period (Unit: radian).

4.5.2.2 Calibration Tab



- **Point Count:**

  Shows how many dots are automatically recorded.

- **Cos Offset Compensation**:

  For calibration to make sure the outcome of computed value is right. When the circle (drawn based on the sin and the cos signals) is deviated from the center of lissajou diagram (shown in the Scope window), input this value to deduct from the cosine signal axis to relocate.

- **Sin Offset Compensation**:

  For calibration to make sure the outcome of computed value is right. When the circle (drawn based on the sin and the cos signals) is deviated from the center of lissajou diagram (shown in the Scope window), input this value to deduct from the sine signal axis to relocate the circle.

- **Gain Compensation**:

  For calibration. When the amplitude of the sine and the cosine signal is not equal (i.e., distorted), the ratio of them will not be **1**. Input a constant in this column to be multiplied by the present ratio so as to adjust the amplitudes to be equal.

e. Phase Compensation:

When the shape of the circle (drawn based on the sin and the cos signals) is oval-shaped (i.e., distorted) due to that the sine and cosine phase is not 90 degrees. Input a value ranged from **0 to 2** π (unit is radian).

f. Record:

click to start recording

g. Calculate:

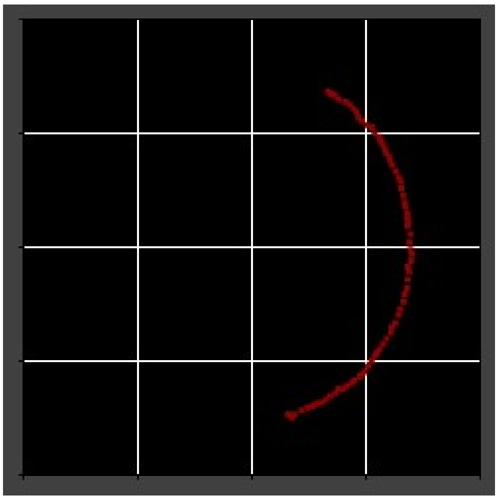
Click to automatically calibrate the sine/cosine encoder signals. Click this button, then manually rotate (or move) the motor to generate signal input.

4.5.2.3 Calibration Steps

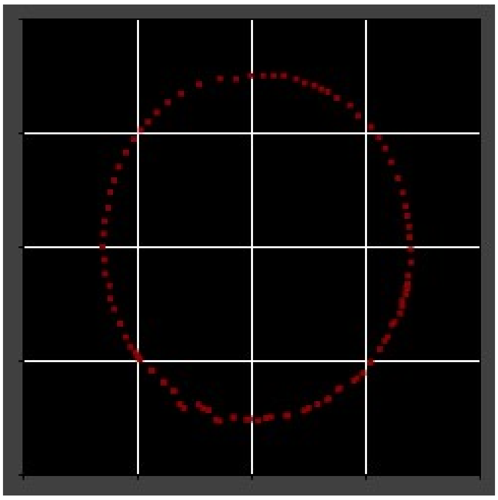
Before calibrating, make sure the encoder is connected properly.

1. Click 

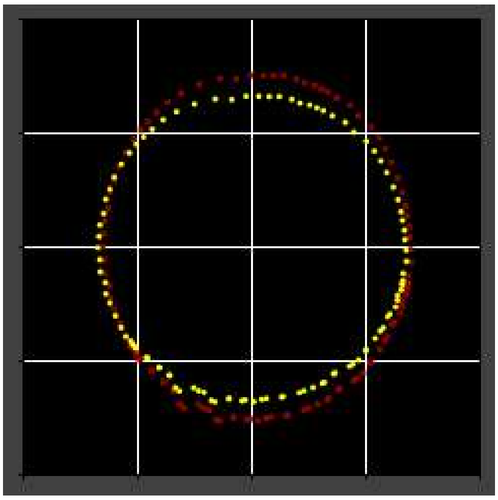
button.
2. Manually rotate the motor, you will see red dots appearing on the panel.



3. Keep rotating until there are more than 100 dots and until the dots can form a circle.
   The more dots recorded, the more accurate the calibration will be.
   To obtain more dots, rotate for more rounds.

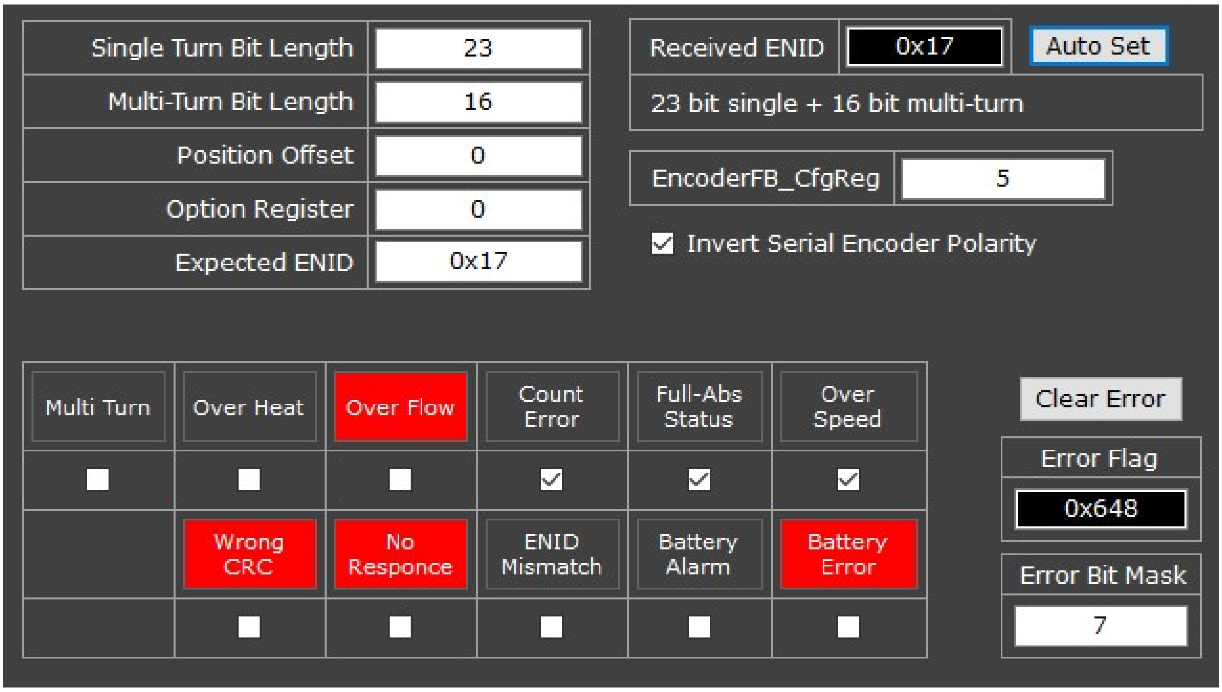


4. Click Calculate, a yellow circle will show up representing the automatic calibration result.



Calibration is now completed.

4.5.3 Tamagawa Encoder



Before configuring, **make sure the motor type (see chapter 4.2) is set to "Rotary type".**

a. Auto Set:

Automatically retrieves and shows the ENID* of the Tamagawa encoder.

*: The term "ENID" refers to "encoder ID". This ID is named by TAMAGAWA SEIKI Co., Ltd. Each Tamagawa encoder model has its own unique ID.

b. Received ENID:

(read-only)

Shows the ENID of the encoder which is now connected with your driver.

c. Single Turn Bit Length:

Configures how many bits—based on 2^{Nth}—can be recorded per motor turn [resolution].

d. Multi-Turn Bit Length:

Configures how many motor turns [resolutions]—based on 2^{Nth}—can be recorded.

e. Expected ENID:

Users can assign an "Encoder ID (ENID)" to write to the driver.

cpc provides 4 sets of default ENID: **0x0, 0x06, 0x11**, and **0x17.** Each of the default ENIDs carries default values of "single turn bit length" and "multi-turn bit length".

- Entering a default ENID will **automatically** change the values of the "single turn bit length" and the "multi-turn bit length".
- Entering a non-default ENID will not change above-mentioned settings of bit lengths. **Users will need to configure "single turn bit length" and "multi-turn bit length" on their own.**

Note:

1. **The ID in this column is the ID written to the driver.**
2. If the ID of the Tamagawa encoder that you are going to operate with is not of cpc's default, please key in the ENID of this encoder so that cpc driver is able to detect if the "Expected ENID" and "Received ENID" are consistent.

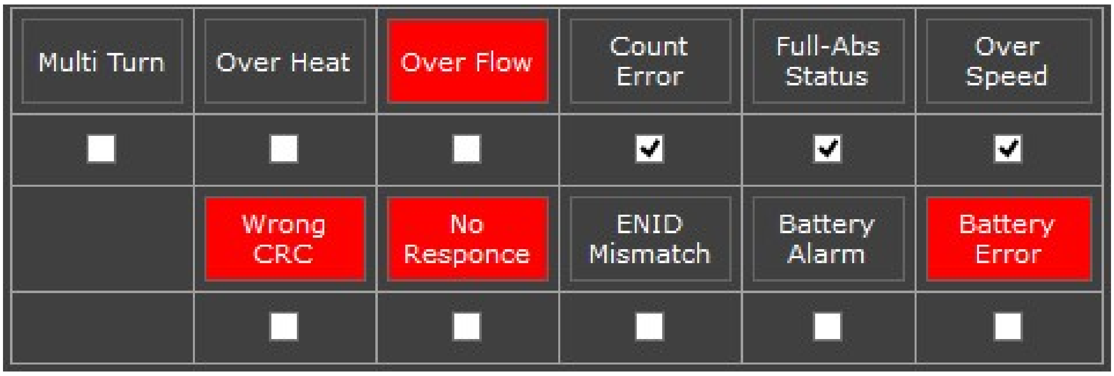
f. Position Offset: Reserved.

g. Option Register: Reserved.

h. Status Flag:

Also called **<u>status bit</u>**.

Each status flag/bit reflects the status of an **encoder error**. **A rising flag/bit** (represented in **red** background) **is seen as an encoder error.**



- The first 8 status bits (*Over Speed* ~ *Battery Alarm*) are from Tamagawa.
- The last 3 bits are cpc-specific. See meanings of signal high (shown in red) below:

- <u>ENID Mismatch</u>:

  Means that the values in user-defined "ENID" column 

 and in the read-only "Received ENID" column 

 don't match.

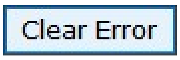
- <u>No Response</u>:

  Driver receives no response from encoder.

- <u>Wrong CRC</u>:

  The format of the message sent from encoder to driver is wrong. This might be caused by noise.

i. Clear Error:



Resets status flags [bits] to logic 0 level.

<u>Note</u>:

To reset successfully, the cause(s) of encoder error needs to be removed in advance.

j. **Error Flag**:

For internal use only.

It's the sum of all the status flags [bits] in hexadecimal.

k. **Error Bit Mask**:

For internal use only.

It's the sum (in decimal) of the ticked bits.

l. **EncoderFB_CfgReg**:

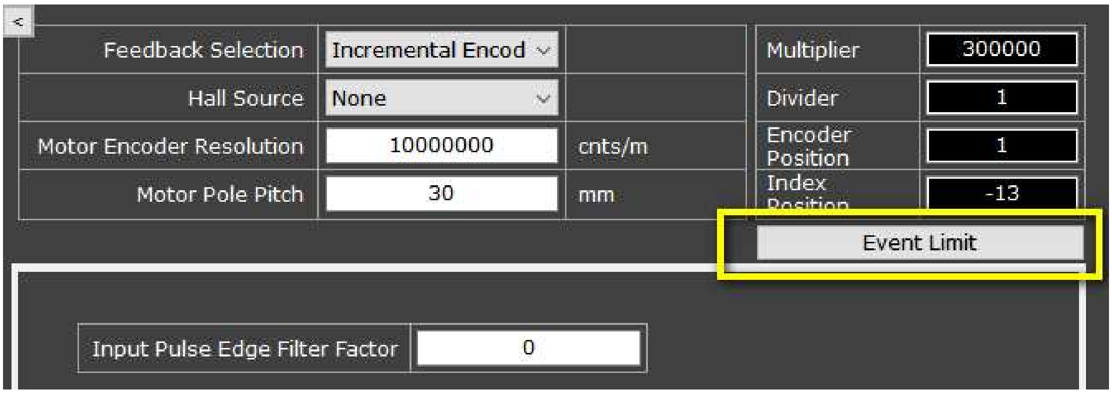
For internal use only.

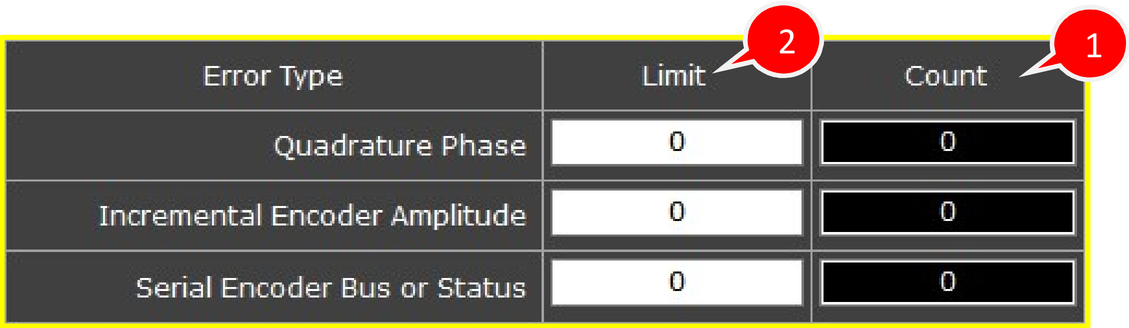
m. **Invert Serial Encoder Polarity**

Same as described in **chapter 4.5.1, point (b)**.

4.5.4 Event Limit

The Event Limit panel is shared by all encoder types. The hidden Event Limit panel will show up by clicking [Event Limit] button.





Each encoder type has its own definition of encoder error.

When the <u>encoder error count①</u> value—which derives from the formula of Encoder Error Calculation*—has accumulated and surpassed the <u>user-defined limit②</u>, a fault event will be triggered.

*** Calculation Formula of Encoder Error Count:**

- If an encoder error appears in one control loop's time, add value 16 to the <u>encoder error count</u>.
- If no encoder error appears, deduct value 1.

a. Count:

Shows the result of the encoder error calculation.

<u>Note</u>:

For Tamagawa encoder, only selected status flag (shown as ticked ) will be counted; namely, be included into error calculation.

b. Limit:

Sets the maximum limit (**0 to 65535**) of triggering a fault event. **When the value of encoder error calculation has exceeded this user-defined limit, a fault event will occur.**

c. Error Type:

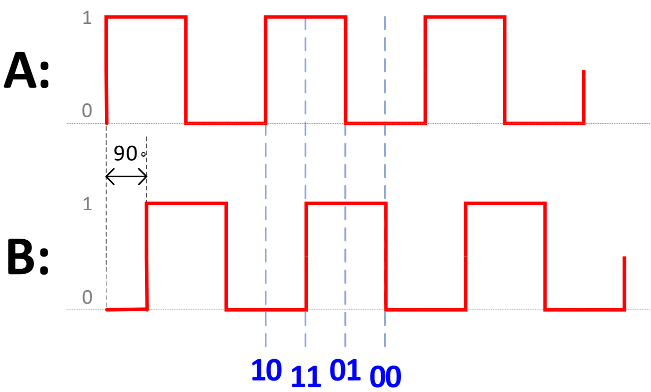
See each encoder type's definition of <u>encoder error</u> as follows:

Error Type	Limit	Count
Quadrature Phase	0	0
Incremental Encoder Amplitude	0	0
Serial Encoder Bus or Status	0	0

(1) **Quadrature Phase**

To set the fault-triggering limit of **<u>Incremental A/B encoder</u>**.

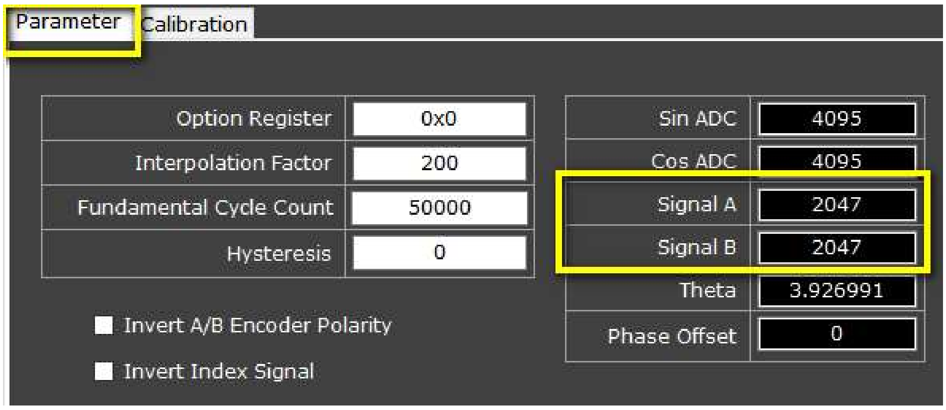
A normal incremental A/B quadrature signal received by the driver should be in line with the 10-11-01-00 sequence (see graph below). If not so, it will be regarded as an encoder error.



(2) **Incremental Encoder Amplitude**

To set the fault-triggering limit of **<u>sine/cosine encoder</u>**:

If the amplitude of sine or cosine is too low—according to the formula when $[(\text{signal A})^2 + (\text{signal B})^2] < 65536$—this will be regarded as an encoder error.

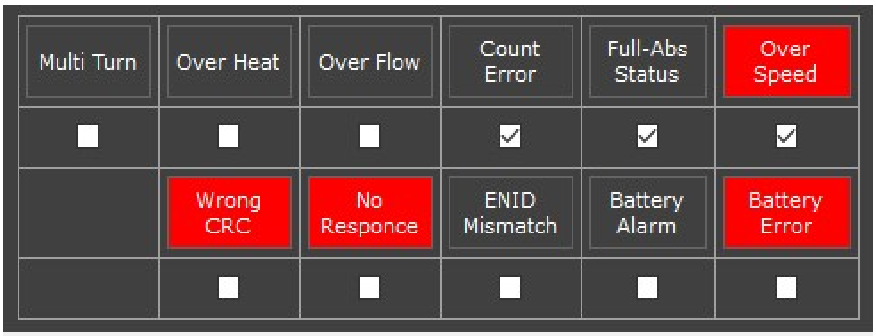


(3) **Serial Encoder Bus or Status**

To set the fault-triggering limit of **<u>Tamagawa encoder</u>**.

Any rising status flag (shown in red background) is viewed as an encoder error; however, **only the user-selected flag(s)** (shown as ticked) **will be included into the encoder error calculation to trigger fault event**.

<u>For example</u>:

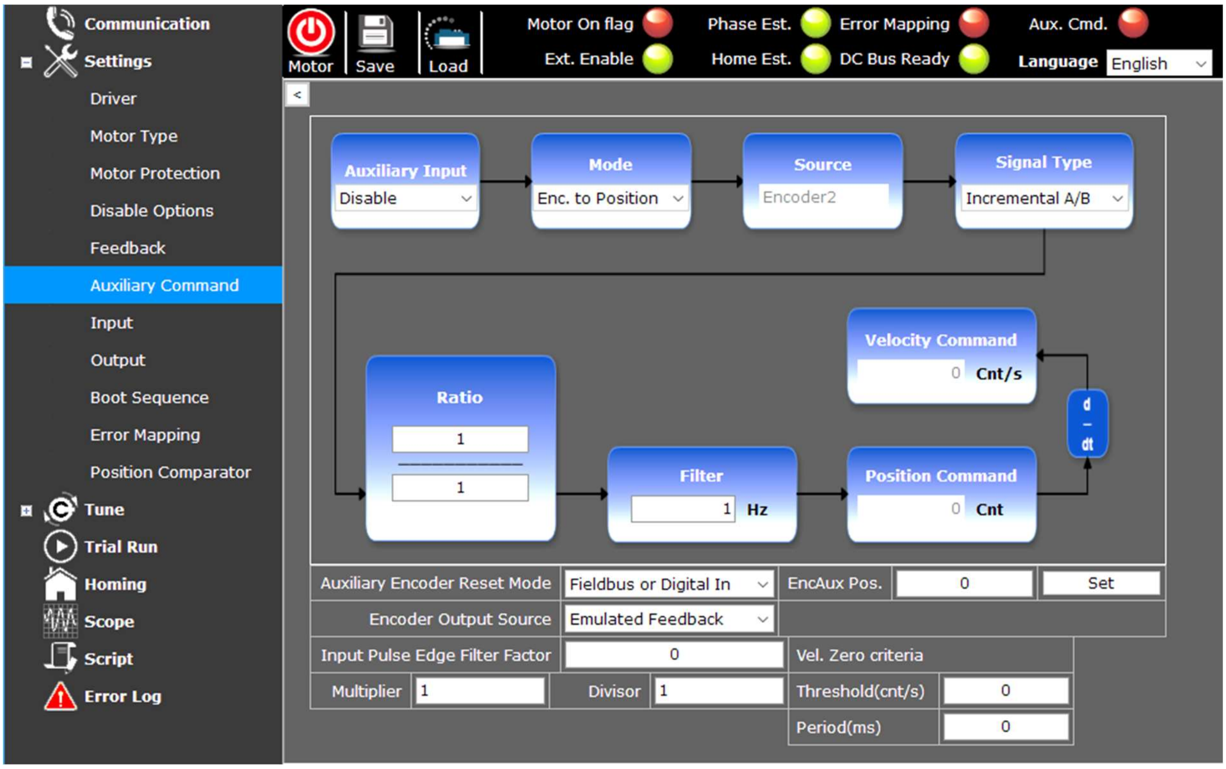


4 status flags have risen (represented in red) indicating the occurrence of encoder errors; yet, only the <u>selected</u> (shown as ticked) status flag "Over Speed" will be counted to trigger fault event.

4.6 Auxiliary Command

Use this tool to set up **how to give and take the commands from controller.** You can choose the signal type and gear ratio and can set the filter frequency.

Select "Enable" to allow Auxiliary Input, the driver then can execute commands from the controller.



There are 4 modes of auxiliary input:

- The first one (Enc. to Position, see **chapter 4.6.1**) uses **pulse signal** and corresponds to position mode;
- The rest three (Analog to Position & Velocity & Current mode, see **chapter 4.6.2**) use **analog signal**.

This panel below is shared in all modes:

a. Auxiliary Encoder Reset Mode:

This function allows users to choose **by which method to set the "received-and-processed position command value"** *note **to zero**. This function is especially used when checking whether the commands sent from controller side are correctly received by the driver.

***Note**:

The "received-and-processed position command value" is shown in this column:

There are 3 ways given by this function to set the received-and-processed position command value to zero:

- Operation Enable (default):

  Set it to 0 when motor-on state is triggered.

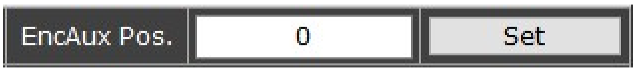
- Homing Attained:

  Set it to 0 when homing is successfully completed.

- Fieldbus or Digital In:

  Set it to 0 via CANbus or RS232, or digital I/O.

b. EncAux Pos.:



The EncAux Pos. value can only be configured in **motor-off** state. **Changing the value in the EncAux Pos. column will overwrite the value of the “received-and-processed position command” *[note].** After configuring, click “Set” button to overwrite.

***Note**:
The “received-and-processed position command value” is shown in this column:

Users can set the “received-and-converted position command” to the value they need by configuring the EncAux Pos. value. Value 0 is the mostly applied value.

Application example:
When setting up controllers with drivers, configure the EncAux Pos. value to be the same as the original command value sent from controller, so that you can check conveniently whether these two sides’ signals are in line with each other.

c. Encoder Output Source:

Selects from which source to output to the controller.

The word "encoder" in this title refers to **<u>the encoder feedback</u> that the controller received**.

- Motor Feedback:

  Copies the motor's feedback signal to the controller.

- Auxiliary Encoder Reference:

  Copies the controller's command to two or more drivers.

  Select this option if you wish to send controller's signals to two or more drivers to execute the same action.

- Emulated Feedback:

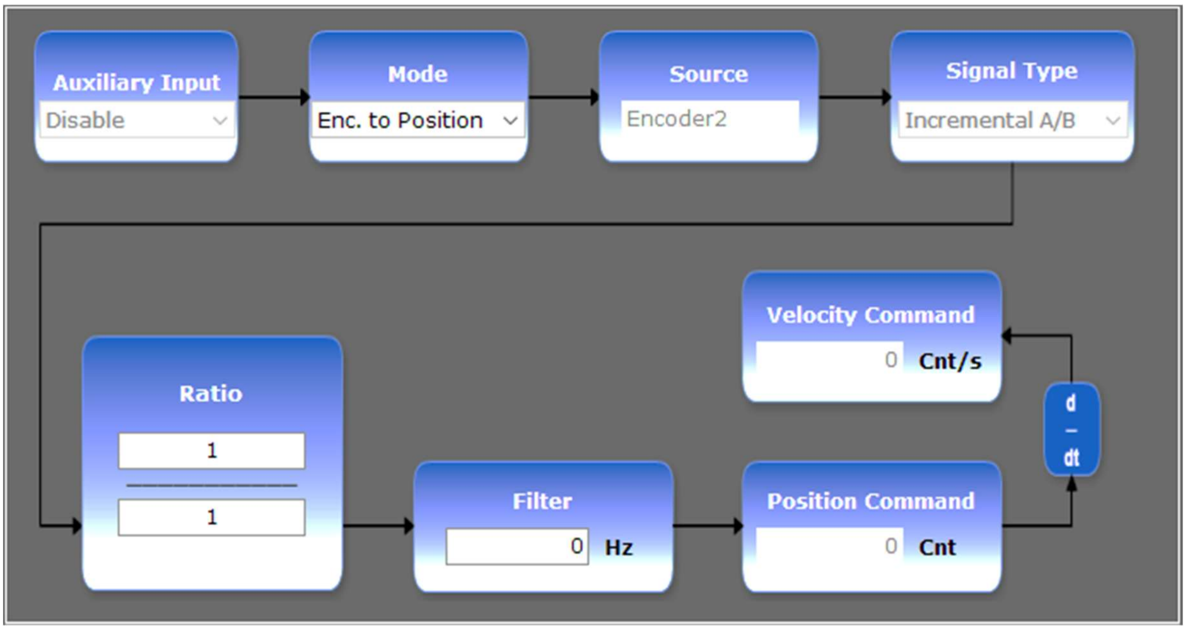
  Sends the "emulated" position feedback values to the controller.

  Select this option when using **Error mapping** function, or for emulating the motor encoder's signals; for the latter, you can set gear ratio by adjusting the multiplier and divisor.

  <u>Note</u>:

  Emulated Feedback function does <u>not</u> apply to the first version of TC1 Driver.

4.6.1 Encoder to Position mode



a. Signal Type:

Define the signal type from the controller.

- Incremental A/B
- Pulse & Direction
- CW/CCW

b. Ratio: cnt/pulse:

Define the ratio between the controller command and the motor movement.

c. Filter:

Low-pass filter.

Remove commands whose frequency is higher than this filter value.

Note: The final Position &Velocity command values are read-only and cannot be modified.

d. Input Pulse Edge Filter Factor:

Note: This function is for the "Encoder to Position" mode only.

- **A length-of-time factor** (to be multiplied by the rest part of the formula mentioned below) **to produce the total length of time of the Filter Window which is used as a threshold to filter off the glitch in input signals.** This value can be **0 to 255**. "Input signal" means, for example, the signals sent from the controller or external encoder to the driver (value 0 = function disabled). The cpc driver provides such **pulse-width filter** function on such signal.

- **Formula**: Length of time of the Filter Window:

$$T_{FW} = \text{Factor} \times 4 \times \frac{1}{Fc*} \text{ (ns)}$$ See note*.

(See next page)

- **Encoder's maximum detectable frequency**:

  (1) If the selected Signal Type is "Incremental A/B":

$$F_S = \frac{F_C*}{\text{Factor x 4 x 4}} \text{ (Hz)}$$

See note* below.

  (2) If the selected Signal Type is "CW/CCW" or "Pulse/Dir":

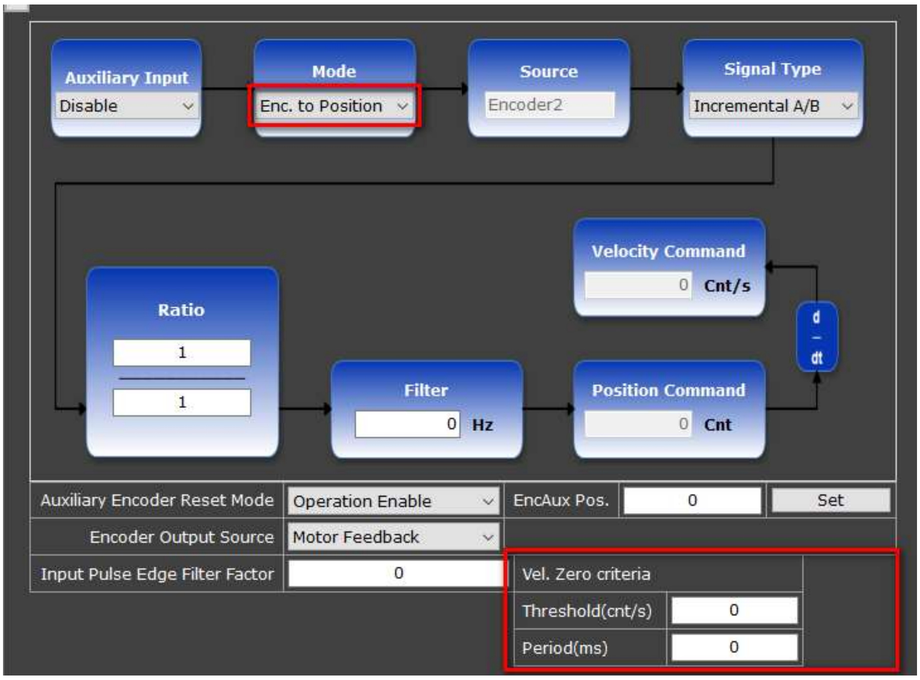
$$F_S = \frac{F_C*}{\text{Factor x 4 x 2}} \text{ (Hz)}$$

See note* below.

Note*: the corresponding value of F_C is

Driver	Value of "F_C"
TC1 Series	**150,000,000**
TC1-B Series	**160,000,000**

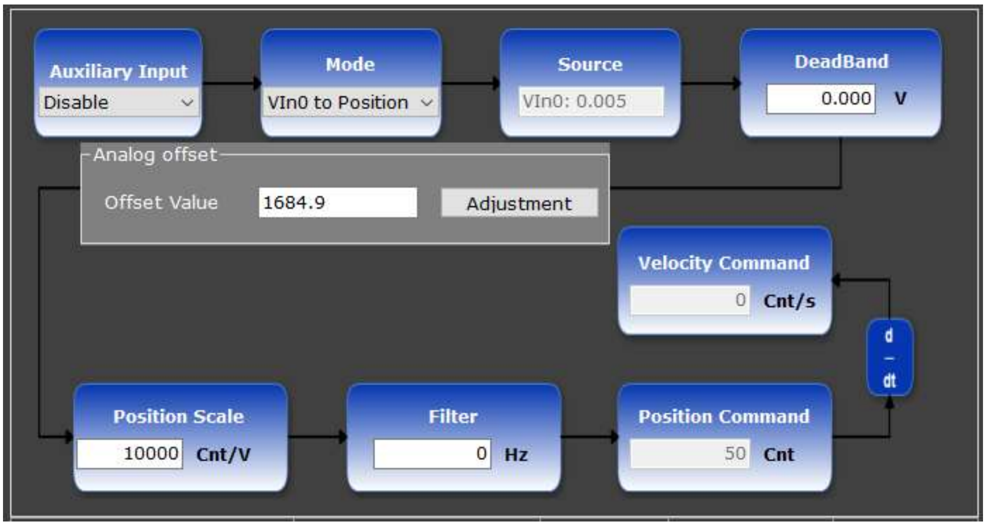
e. Vel. Zero Criteria / Threshold (cnt/s) / Period (ms):

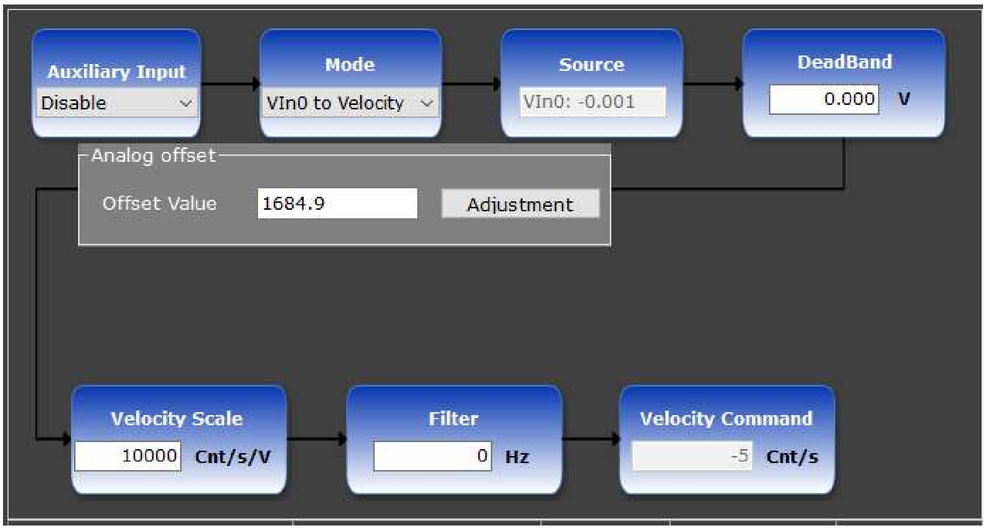


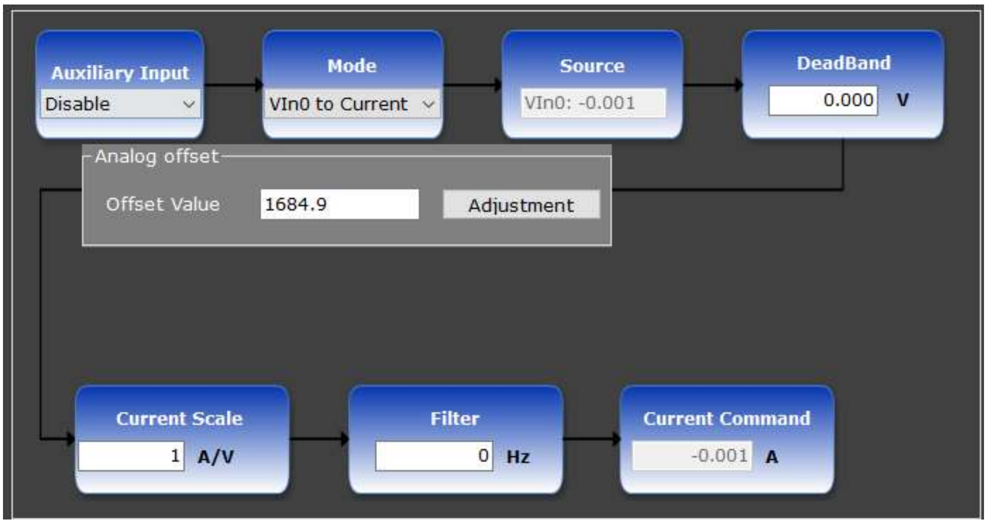
Note: This function is for the "Encoder to Position" mode only.

It is used to determine **whether the pulses of auxiliary command have stopped** – The pulse command which is **lower or equal to** this velocity (cnt/s) for a duration **longer than or equal to** this configured period (ms) will be regarded as zero (i.e., the auxiliary command has stopped giving orders).

4.6.2 Analog to Position & Velocity & Current mode







a. **Dead Band:**

To remove the noise.

The signal voltage **lower** than this value will be deemed to **zero**. The band will be the **+/- range** of the keyed-in value. For example, if the value is set at 1 V, the dead band will be +/-1 V.

b. **Analog offset:**

Digital offset value.

The controller sends out its analog signals (ranging from -10 V to 10 V) which will be converted into Analog-to-Digital-Converter (ADC) values (ranging from 0 to 4095). These analog signals need to be calibrated. To do so, the middle of the analog voltage signal (0 V) needs to align to the middle of the ADC value (2048) at driver. Click the "**Adjustment**" button to automatically calibrate the analog offset, or key in values to adjust manually.

c. **Position / Velocity / Current Scale:**

The gear ratio.

Converts the analog voltage command into position, or velocity, or current command.

d. **Filter:**

Low-pass filter.

Removes commands whose frequency is greater than this value.

Note: The final Position & Velocity & Current command values are read-only, cannot be modified.

4.7 Input & Output

The Input setting defines the trigger function that should occur when driver receives “true” logic from input signals. The Output setting defines what signals to be output while some events occur.

For pin-definition of input and output please refer to the cpc TC1/TC1-B Installation Guide.

The digital I/Os interfaces enable users to **configure** functionality as well as polarity of the digital I/Os and to **monitor** the state of all digital I/Os. In cpc GUI , the green light  represents logic “true”, and the red light  represents logic “false”.

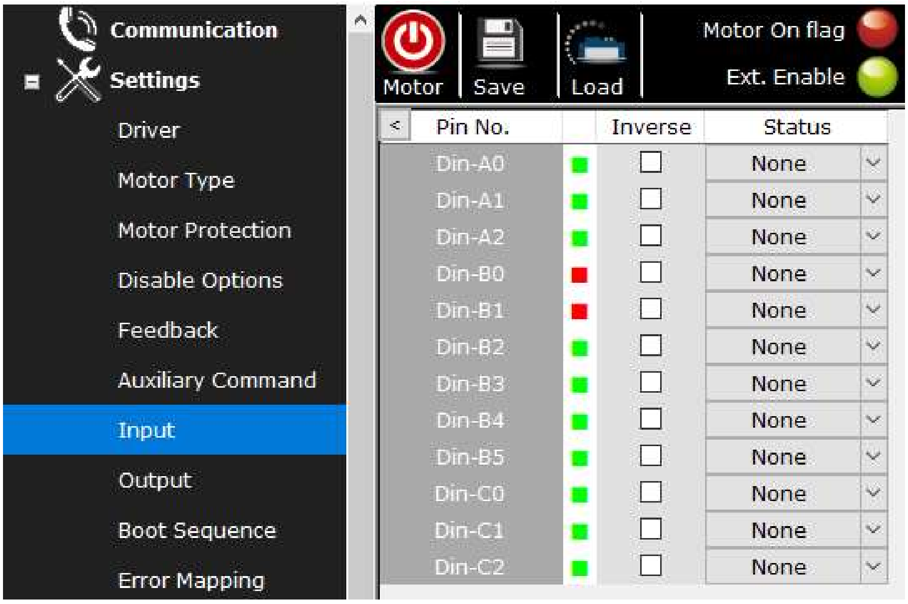
<u>**Note**</u>:

Only the “External Enable” function requires ALL the pins (which are assigned to be Ext. Enable) to be logic true to make this signal high. For the rest of I/O functions, it takes only ONE pin to be logic true to make signal high.

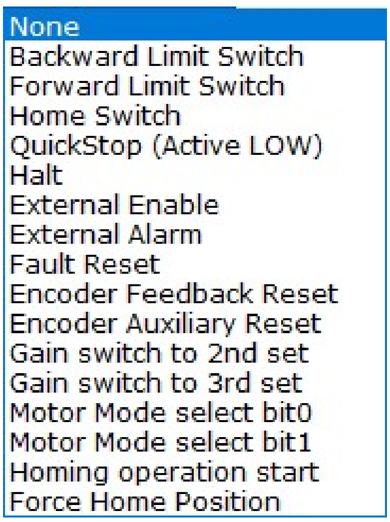
4.7.1 Input

Meaning: The signals that are sent **to the driver.**

- Input:



- Input panel explanation

INPUT	
State	A graphic element that toggles between green and red to reflect the on/off state of the actual input.
Pin No.	Specifies a pin.
Function	Defines the functionality of the digital input. 

INPUT	
Inverse	Click to invert the polarity of a digital input. As the result of inversion, the graphic element in the software immediately changes color.

4.7.1.1 Input Functions

Input Functions	Description
Backward Limit Switch	**Defines this pin as signal state of Backward Limit Switch**
Forward Limit Switch	**Defines this pin as signal state of Forward Limit Switch**
Home Switch	**Defines this pin as signal state of Home Switch**
QuickStop (Active LOW)	**Defines this pin as signal state of QuickStop** (QuickStop will be active when signal is Low).
Halt	**Defines this pin as signal state of Halt.** (Motor motion will be paused when signal is high)
External Enable	**Defines this pin as signal state of External Enable.** **Note:** if multiple pins are set to External Enable, ALL of these pins need to be High to activate External Enable event.
External Alarm	**Defines this pin as signal state of External Alarm.** (If the drive receives external alarm, Fault event (error code: **90F0**) will be triggered).
Fault Reset	**Defines this pin as signal state of Fault Reset.**
Encoder Feedback Reset	**Defines this pin as signal state of Encoder Feedback Reset.** (When signal is high, it will set the motor's present position as 0.)
Encoder Auxiliary Reset	**Defines this pin as signal state of Encoder Auxiliary Reset.** (When signal is high it will set the auxiliary command value as 0.)
Gain switch to 2nd set	**Switch to the 2nd set of gain.**
Gain switch to 3rd set	**Switch to the 3rd set of gain.**

Input Functions	Description

Motor Mode select bit0

This function is for **Boot Sequence** (See Ch. 4.8).

In boot sequence's "Motor-ON" setting, users can configure up to 4 sets of motor-on modes, numbered #0 to #3.

To enable altering the motor-on modes, users **need to equip at least 3 physical I/O switches; they are: External Enable, Select bit 0, and Select bit1.**

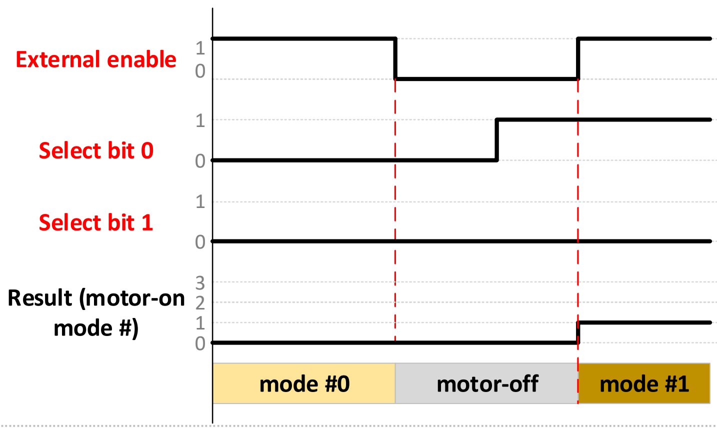
- The "select bit0" and "select bit1" together will designate a motor-on mode (see coding chart):

Coding chart

Bit 0	Bit 1	Result (motor-on mode #)
0	0	0
1	0	1
0	1	2
1	1	3

- **How to alter Motor-ON mode under Boot Sequence**:

Example of Switching Motor-on Mode in Boot Sequence
(From mode #0 to mode #1)



Steps:

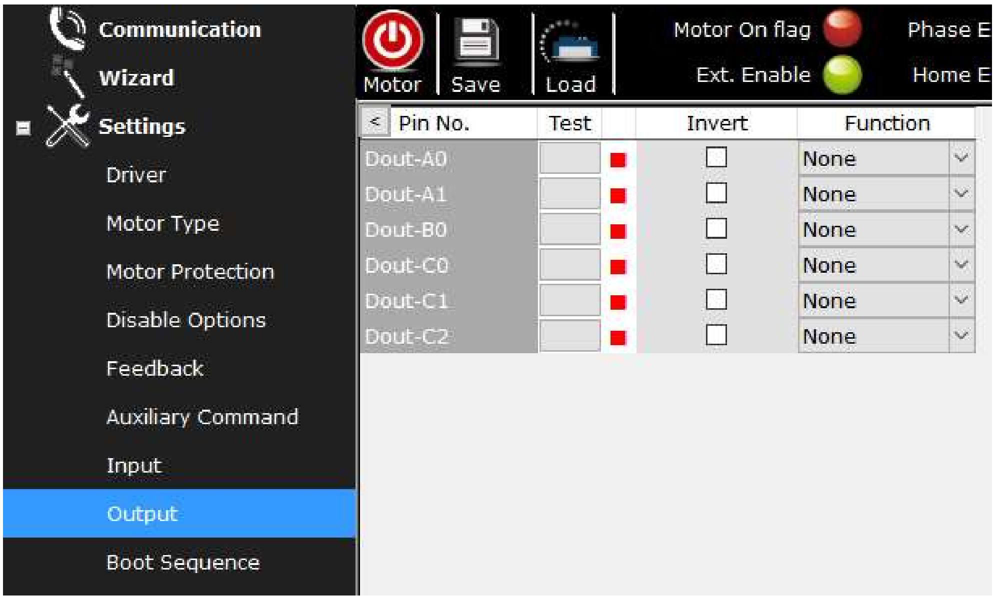
1. Equip these 3 sets of I/O:
   External Enable, Select bit 0, and Select bit1.

Input Functions	Description
	2. Go to UI > Input section, define the above-mentioned 3 sets of I/O pins. 3. Switch motor off (by setting the External enable I/O to 0). Then set the combination of pins “Select bit0” and “Select bit1” to the desired mode according to the coding chart. 4. Switch motor on (by setting the External enable I/O to 1). The motor is now turned on using newly-selected mode.
Motor Mode select bit1	**Defines this pin as Motor Mode select bit1.** See details in the column “Motor Mode select bit0”.
Homing operation start	This function is for **Boot Sequence** (see Ch. 4.8). If the Motor-ON mode under Boot Sequence is set to Homing mode, the driver will need to wait for a “start” signal to execute homing. **Once "Homing operation start" signal is activated, the motor will start performing the homing mode.**
Force Home Position	When Home is not found/set yet, make this input signal high to **force the driver to set the motor's present position as Home.** The corresponding Motor-ON modes should be position-related modes, such as *direct position* or *profile position* modes.

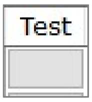
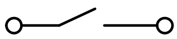
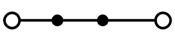
4.7.2 Output

Meaning: The signals that are sent **from the driver to other devices.**

- Output:



- Output panel explanation

OUTPUT	
State	A graphic element that toggles between green and red to reflect the on or off state of the actual input.
Test	Click this button to simulate output. 
Pin No.	Specifies a pin.
Function	Defines the functionality of the digital output.
Invert	Inverts the polarity of a digital output. Tick this box to invert the polarity. • **Unticked**: normal open.  • **Ticked**: normal close. 

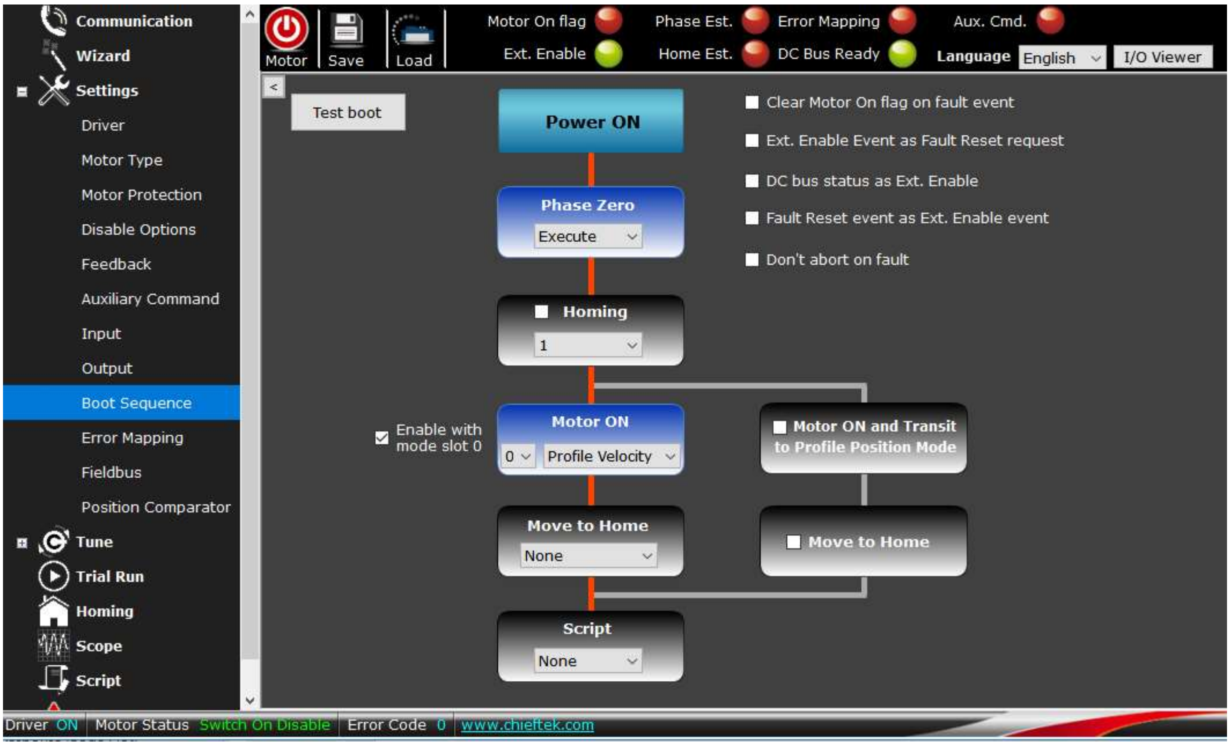
4.7.2.1 Output Functions

Output Functions	Description – When signal state is HIGH
Ready for Ext. Enable	If there is no error and the motor-on flag is on, send out a signal saying that **the drive is ready for taking commands.** (Hence, if External Enable is activated, the motor can respond to controller's commands.)
STW: Fault	**Defines this pin as signal state of the status word "Fault".** (High = Fault occurs)
STW: Target Reached	**Defines this pin as signal state of the status word "Target Reached".** (High = target reached)
Brake	**Links the Holding Brake demands to this pin.**
STW: Homing Attained	This is for Homing mode. **Defines this pin as signal state of the status word "Homing Attained".** (High = Home attained)
STW: Homing Error	This is for Homing mode. **Defines this pin as signal state of the status word "Homing Error".** (High = Homing error occurs)
Home Established	Sends out a signal **constantly** saying that Home is already established.

Motor at rest	Sends out a signal that **the motor is at rest**. [Definition]: In the Motor Protection's Velocity section, users can define Velocity Zero Threshold and Timeout parameters. If velocity is **greater** than this threshold and lasts for a period (Timeout) – the motor will be regarded as **moving**, vice versa. 
Pos. Compare 0 out	Sends out a signal when the #0 set condition of Position Comparator is satisfied.
Pos. Compare 1 out	Sends out a signal when the #1 set condition of Position Comparator is satisfied.
Pos. Compare 2 out	Sends out a signal when the #2 set condition of Position Comparator is satisfied.

4.8 Boot Sequence

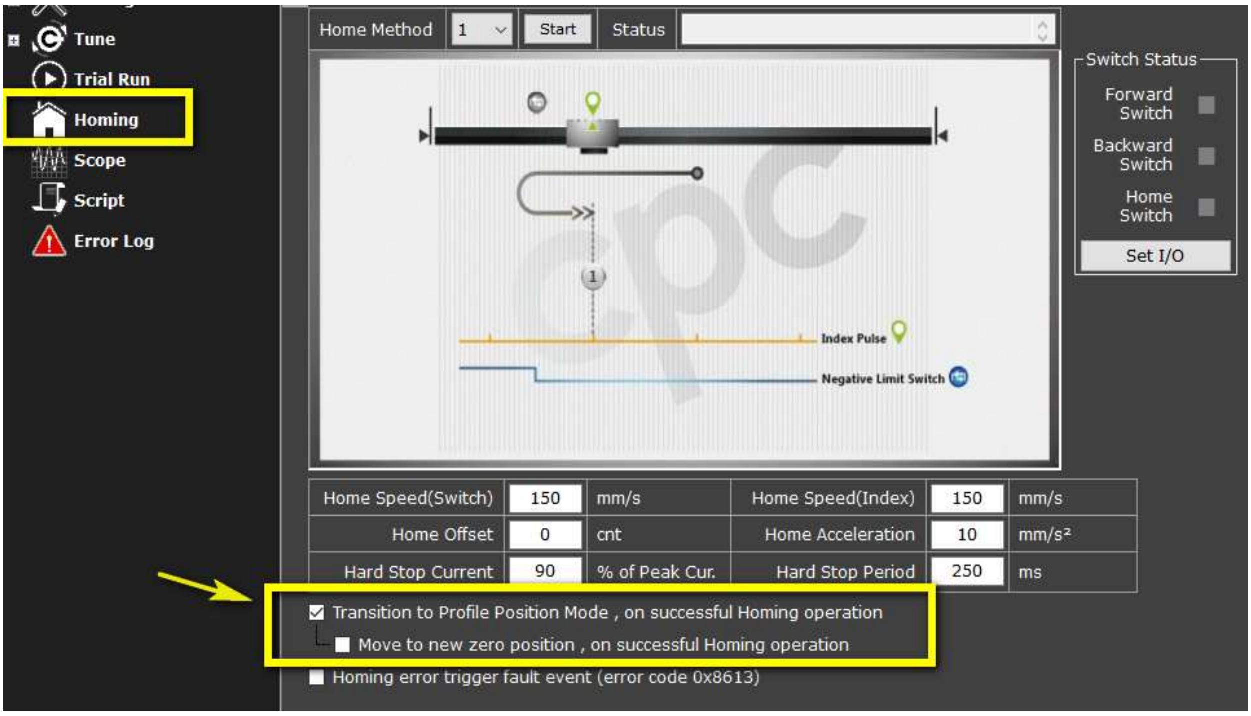
Boot Sequence is for **configuring what initial actions to take after the drive is powered on.** The initial actions may include find-phase, homing, motor-on, moving to home, and executing script.



There are two routes, LEFT and RIGHT, in boot sequence. **The chosen route will be highlighted in red.**

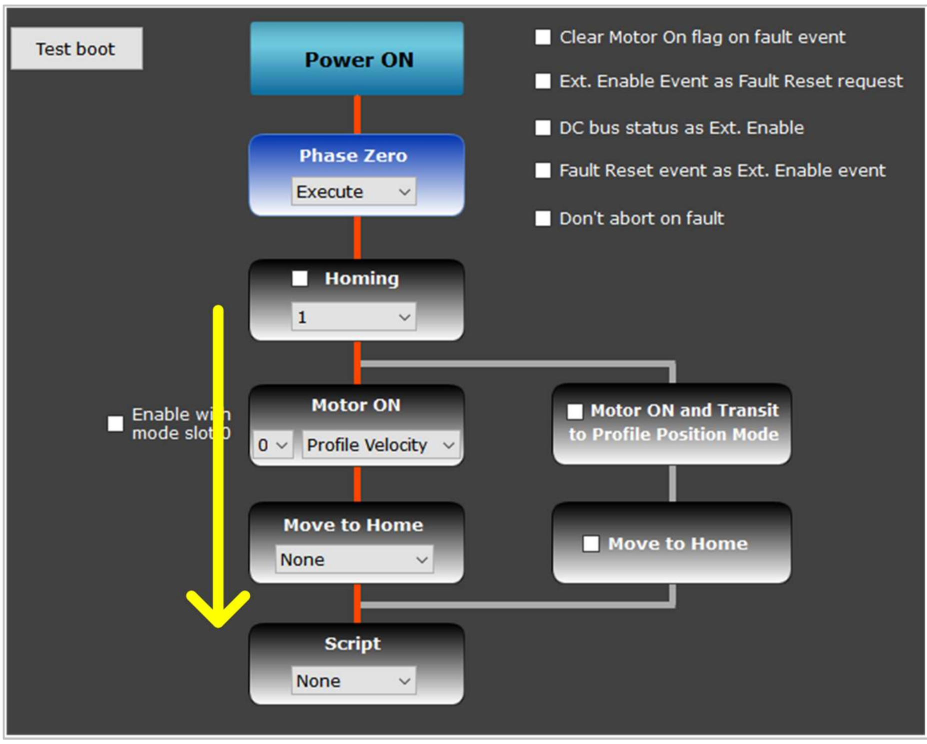
- The **LEFT** route allows users to do any configuration.
- The **RIGHT** route shows and alters the interlinked same settings in the Homing section. The settings of the RIGHT route (even when it is not activated) will be synchronized with those of the Homing section, see pictures below.

(↓ The interlinked same setting in Homing section)



See here for activation conditions:

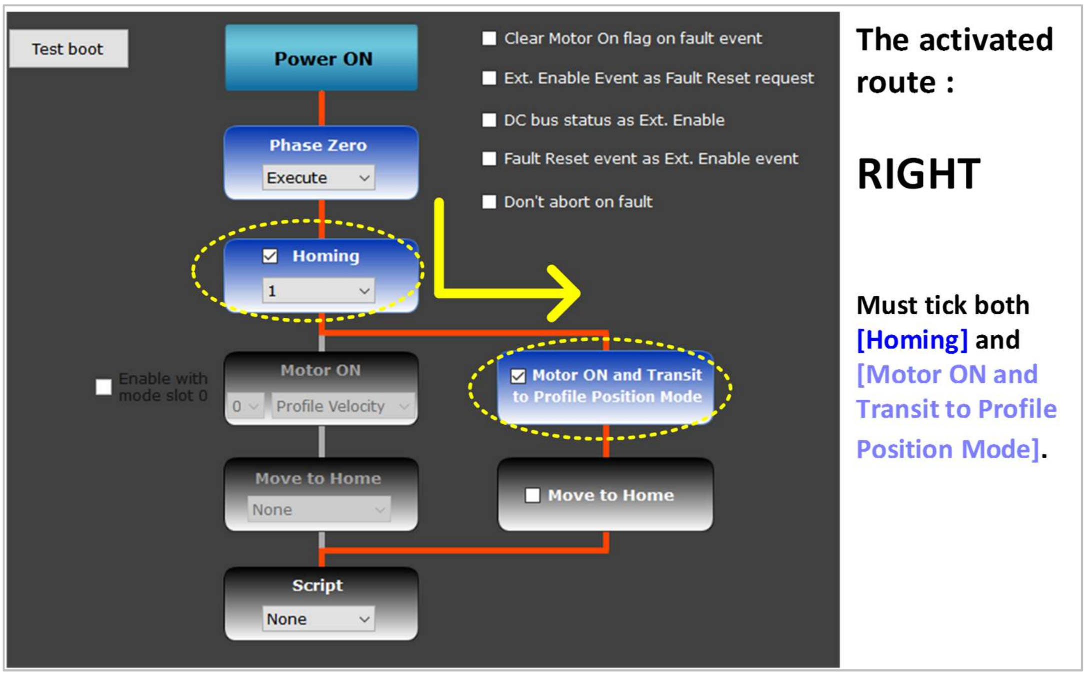
↓ ①: When LEFT route is activated



The activated route :

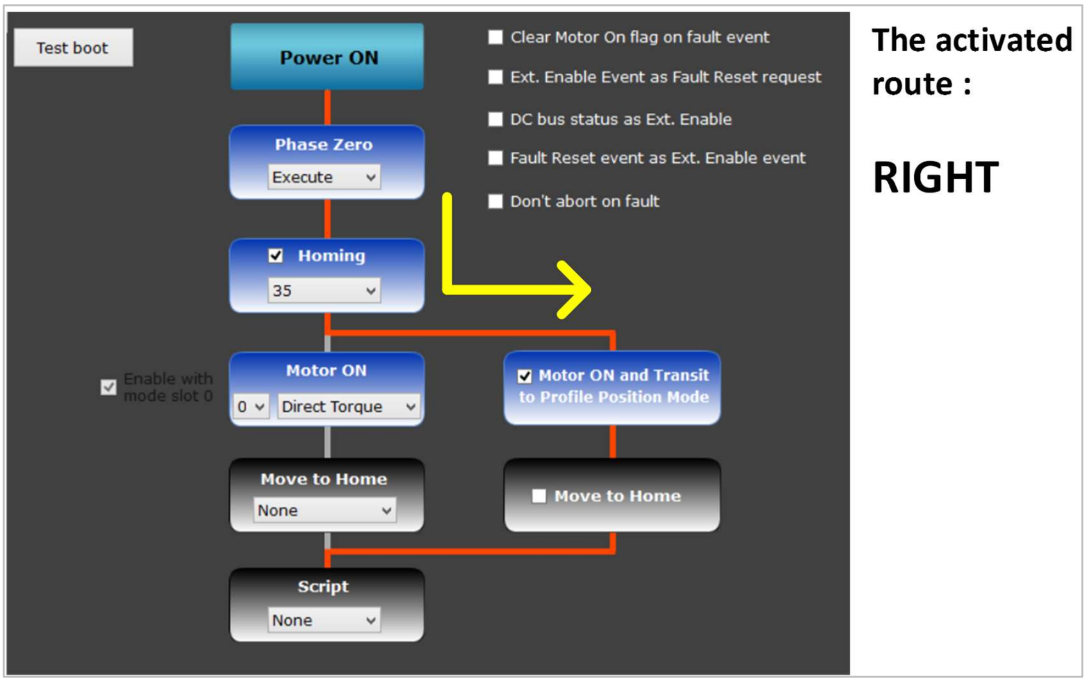
LEFT

↓ ②: Conditions to activate the RIGHT route



Note: In Boot Sequence, only when **BOTH** the boxes of [Homing] and [Motor ON and Transit to Profile Position Mode] are ticked, can the **RIGHT** route be activated.

↓ ③: If both routes are ticked, only the RIGHT route will be activated



Note: In Boot Sequence, if both routes are ticked, only the **RIGHT** route will be activated.

Please see the **configuration procedure** of boot sequence on subsequent pages.

Boot Sequence Configuration Procedure		
1	**Presetting**	• Clear Motor On flag on fault event : (Ticked): **Deactivates "Software Enable" when fault occurs.**
		• Ext. Enable Event as Fault Reset request : (Ticked): **When an External Enable event occurs, treat it as a fault reset action.** <u>Note</u>: ➢ "Event" means that the External Enable was off and then on. ➢ When a fault occurs, usually the following steps are taken: (1) Fault occurs → (2) Resolve the fault by personnel → (3) Perform "Fault Reset"→ (4) Perform External Enable "off" → (5) Perform External Enable "on". However, if users don't have an I/O for "Fault Reset", tick this box so that Fault Reset will be executed between step (4) and (5).
		• DC bus status as Ext. Enable **:** (Ticked): **Treats DC bus signal as External Enable event.** <u>Occasion example</u>: If AC power is cut off, the driver will detect that AC power is too low; hence a fault occurs. When AC power is turned on again and is successfully converted into DC power (in this case the DC bus signal will rise), treat the DC bus signal as an External Enable event.

Boot Sequence Configuration Procedure

- Fault Reset event as Ext. Enable event **:**

  (Ticked): **Treats Fault Reset action as an External Enable event.**

- Don't abort on fault **:**

  (Ticked): **Do not leave Boot Sequence when fault occurs.**

  When a fault occurrs during booting:

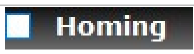
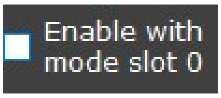
  - **If ticked,** the drive will stay in boot sequence (won't abort boot sequence); after the fault is cleared and has been reset, the drive will automatically restart boot sequence operation.
  - **If unticked,** after a fault is cleared and reset, users will need to power off and power on the device(s) to restart boot sequence operation.

- 

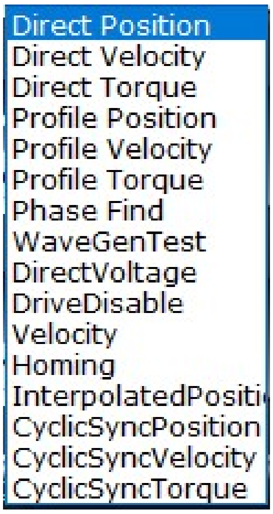
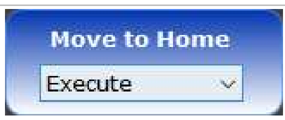
  Motor ON and Transit to Profile Position Mode.

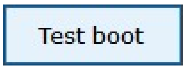
  To activate this function, users must also tick (activate) the box of Homing in boot sequence.
  It's only suitable to activate this function when users need to apply Profile Position mode.

  IMPORTANT:
  Ticking/Unticking this box in Boot Sequence section will **link with the same setting in Homing section** (see **chapter 7.1**); **vice versa**.

Boot Sequence Configuration Procedure		
		•  **Move to Home** (Move to the new zero position.) This box is a **subset** under the higher box [*Motor ON and Transit to Profile Position Mode*]. To use this function, the [*Motor ON and Transit…Mode*] box needs to be activated first.
2	**Phase Zero**	**Performs phase-find.** Note: If no hall sensor is installed, the motor will move for a certain distance while performing phase-find.
3	**Homing**	•  : Tick this box to include Homing in the boot sequence. •  : Choose a Homing method from # -12 to #35, see **chapter 7**.
4	**Motor ON**	**Motor-ON and use the selected mode for operation.** •  : Tick this box **to include Motor-ON in the boot sequence.** •  : ➢ Users can assign 4 sets of motor-on mode slot (#0 to #3). ➢ #0 is default for booting. ➢ #1~3: If users need to alter motor-on mode (e.g., from

<table>
<tr><th colspan="3">Boot Sequence Configuration Procedure</th></tr>
<tr><td></td><td></td><td>Profile Position to Profile Torque), they'll need to configure in this column.
➢ #1~3 need to work with physical I/Os.
Please refer to chapter 4.7.1.1 – "Motor Mode select bit0" for detailed configuration.

• Options of motor-on mode:

</td></tr>
<tr><td>5</td><td>Move to Home</td><td>

Move to Home or not.</td></tr>
<tr><td>6</td><td>Script</td><td>Run the script or not.</td></tr>
<tr><td>7</td><td>Test boot</td><td>

Click to test your boot sequence settings.</td></tr>
</table>

4.9 Error Mapping

It might be possible that the encoder feedback is not accurate.

For instance, the optical scale is not installed straight. In this case, suppose both the position command we give and the position feedback obtained from the present encoder is 1000 cnt. Yet, when measuring the position with a finer device—such as a laser interferometer—we might find the actual position is 1003 cnts instead.

This Error Mapping function is for troubleshooting abovementioned problems. To do so, users will need a device which has finer resolution than that of the currently-used encoder.

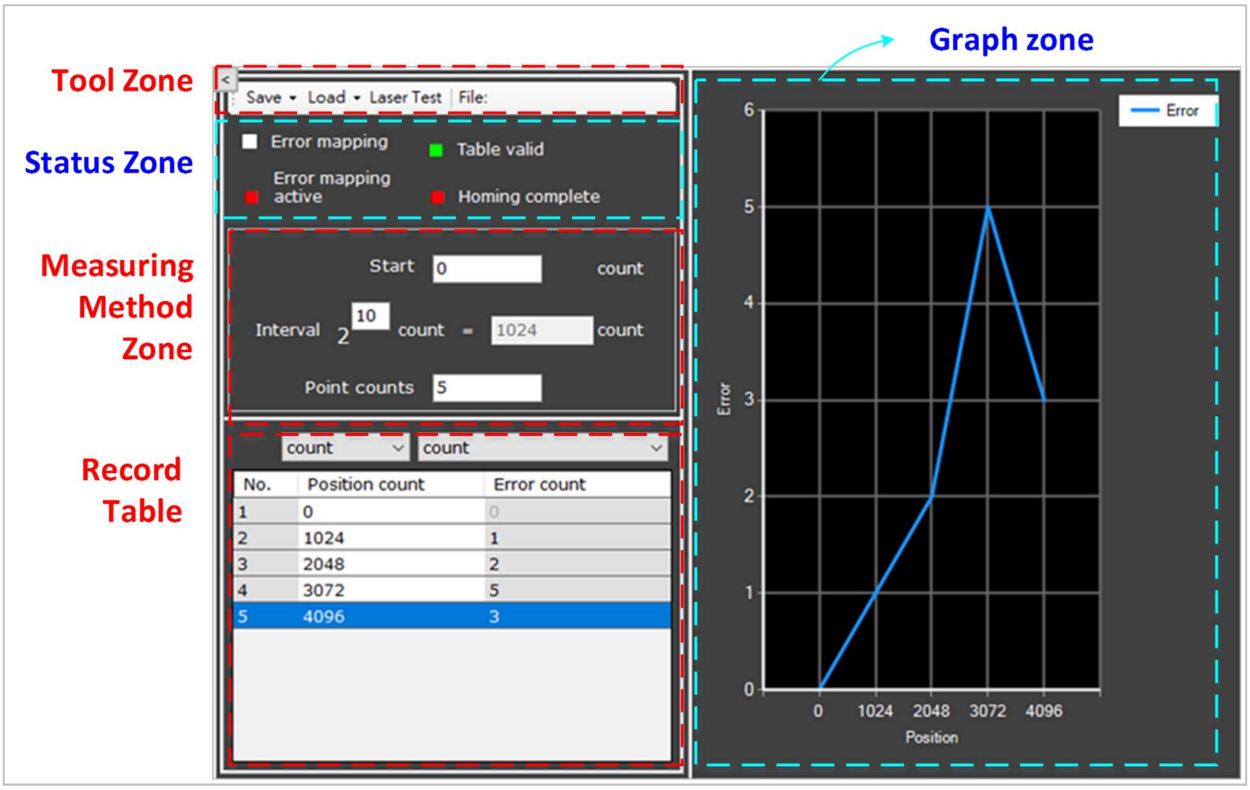
Next, users will need to perform homing first, designate a range (area) to be calibrated, and configure the error mapping settings. When error mapping function is activated, driver will automatically compensate the differences (i.e. errors) in the designated range*.

Note*: Errors occurred outside the designated error mapping range will NOT be compensated.

See explanation of each interface element in **chapter 4.9.1.**
See configuration steps in **chapter 4.9.2.**

4.9.1 Error Mapping Interface

Overview



4.9.1.1 Tool Zone

a. Save:

Save to File or To Flash.

Note:

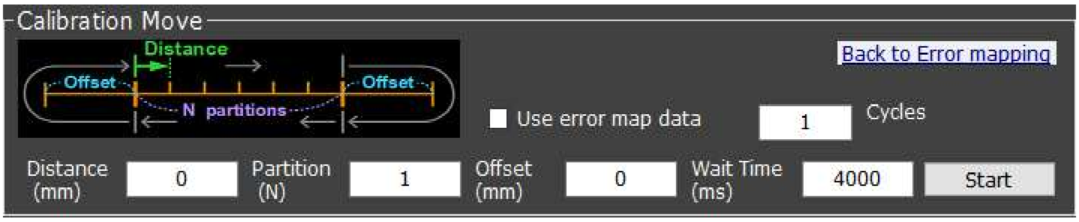
Your settings **must be saved to Flash** to be processed. If you only save to File, the parameters will NOT be loaded to the driver to execute.

b. Load:

To load settings from file or Flash.

c. Laser Test / Calibration Move:

Click it to jump to the [Calibration Move] panel.



This panel is a convenient tool for **designing a set of consecutive motor movements to work with a finer device.**

(1) Distance:

The length (mm) of <u>each</u> partition.

(2) Partition:

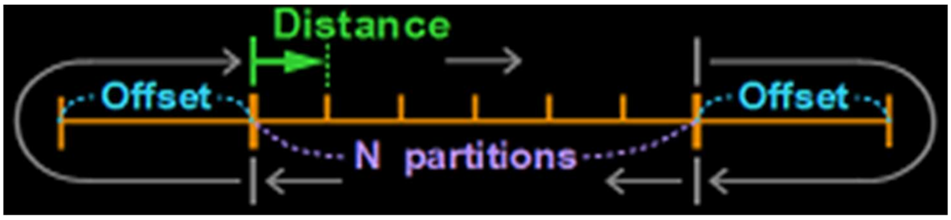
How many partitions (N) to be measured by the finer device.

(3) Offset:

After one cycle of consecutive movements [N partitions] is completed, <u>the distance (mm) that the motor needs to move further forward and then backward onto the end of the last partition</u>. See image below, marked in blue.

<u>Note</u>:

The Calibration Move cycles will be conducted in **round-trip**.



(4) Wait Time:

How much time (ms) the motor needs to pause motion after reaching one end of a partition. Key in values from **2000 to 10,000.**

(5) Cycles:

How many times to repeat the set of consecutive motor movements.

(6) Use error map data:

Tick to automatically set the parameters of “Distance” and “Partition” to be in line with the settings of “Interval” and “Point counts” on Error mapping panel.

For example:

4 Partitions = 5 Point counts.

(7) Start:

Start the whole configured calibration moves.

(8) Back to Error mapping:

Click to jump back to Error mapping panel.

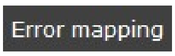
4.9.1.2 Status Zone

a. Error mapping:

Tick to check prerequisites and activate the error mapping function.

Note:

It is mandatory that BOTH the two prerequisites—Table valid and Homing complete—are fulfilled (signal ON, shown in green) to activate the Error Mapping function.

If this box  is ticked while either or both of the prerequisite signals are off, the activation of error mapping will wait until BOTH prerequisites are fulfilled.

b. Table valid:

Note:

You must first perform "Save to FLASH" to check the validity of data.

The driver will check:

- Table size (namely, point counts not exceeding maximum default).
- Validity of data version (namely, the version of Measuring Method Zone and Record Table is correct).

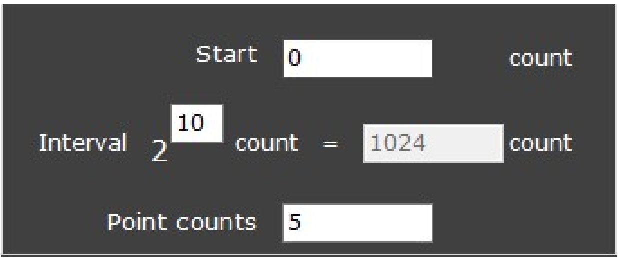
c. Homing complete:

Green if Home is established.

d. Error mapping active:

Turns green (meaning error mapping function is activated) on the condition that both "Table valid" and "Homing complete" signals are on.

4.9.1.3 Measuring Method Zone



a. Start:

Start measuring from what position (count). This zone is where you set the designated area to be calibrated.

b. Interval:

How big the internal between measurement nodes needs to be.
The Interval is based on {2 to the [user configuration]th power}.

c. Point counts:

How many nodes.

4.9.1.4 Record Table

Manually key in here the difference (i.e., error count) between the finer device and the encoder feedback.

No.	Position count	Error count
1	0	0
2	1024	1
3	2048	2
4	3072	5
5	4096	3

a. **Position count:**

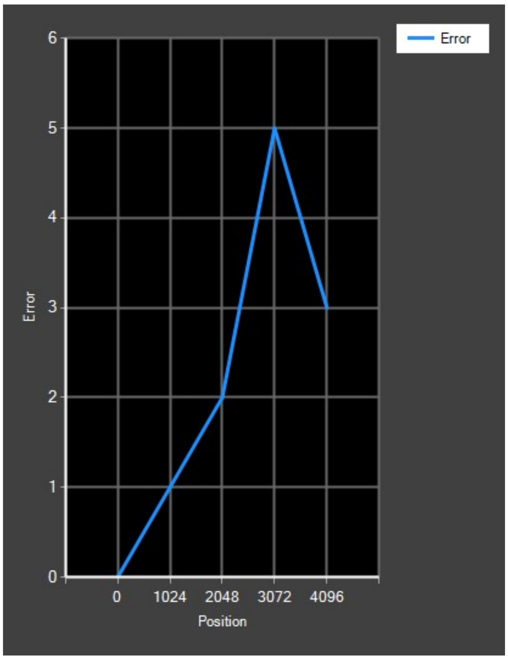
Shows the actual position of each node.

b. **Error count:**

Key in difference(s) manually.

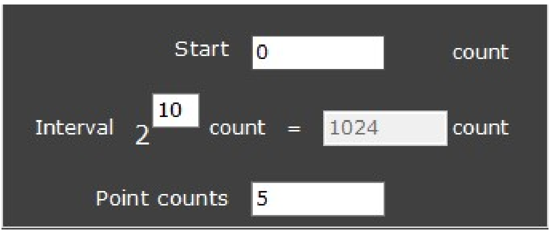
4.9.1.5 Graph Zone

The UI will automatically draw a graph according to the given record table.



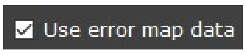
4.9.2 Configuration Steps

1. **Perform Homing first**.
2. **Configure the Measuring Method Zone**:



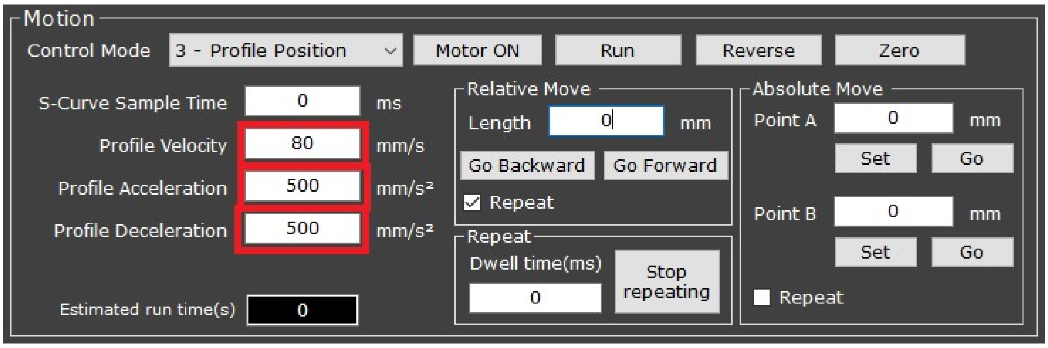
   (1) where to **start** measuring,
   (2) how many counts the **interval** is, and
   (3) how many nodes (i.e., **Point count**) you need. Each node is equally spaced.

3. **Configure "Calibration Move"**:
   (1) Click "Laser Test".

   (2) Configure the "Calibration Move" parameters (see **chapter 4.9.1.1**, point c.).
   OR, for convenience, click  button.

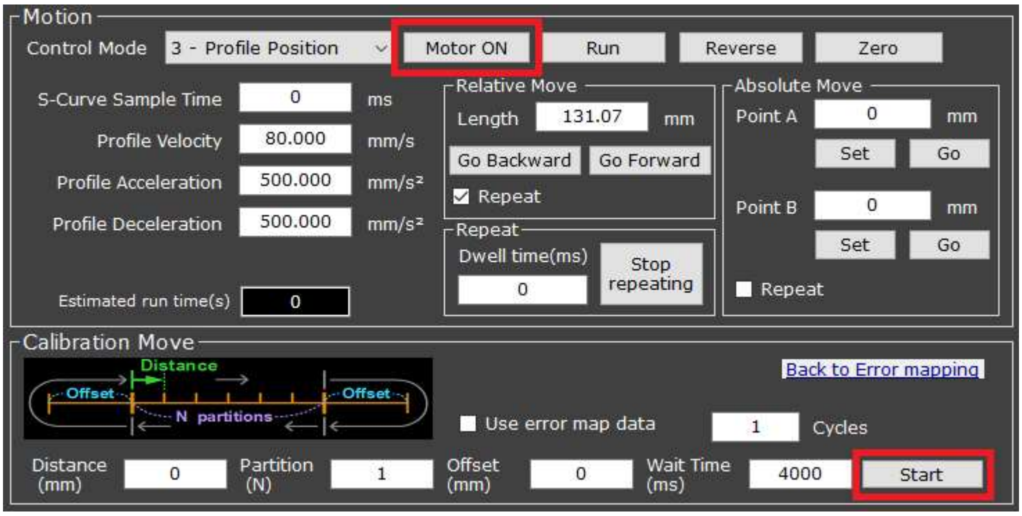
   (3) Then set the Offset, Wait Time, and Cycles. See picture below.

(4) On the “Motion” panel, choose <u>Profile Position</u> mode. Configure the motor parameters of Profile Velocity, Profile Acceleration and Deceleration.



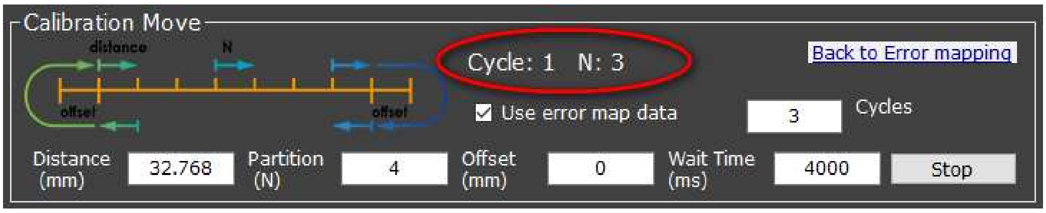
(5) **Click Motor-ON button.**

(6) **Click Start button.**



The motor will then move to the user-assigned-position to start measuring (see **Ch. 4.9.2**, point #2) and perform the Calibration Move.

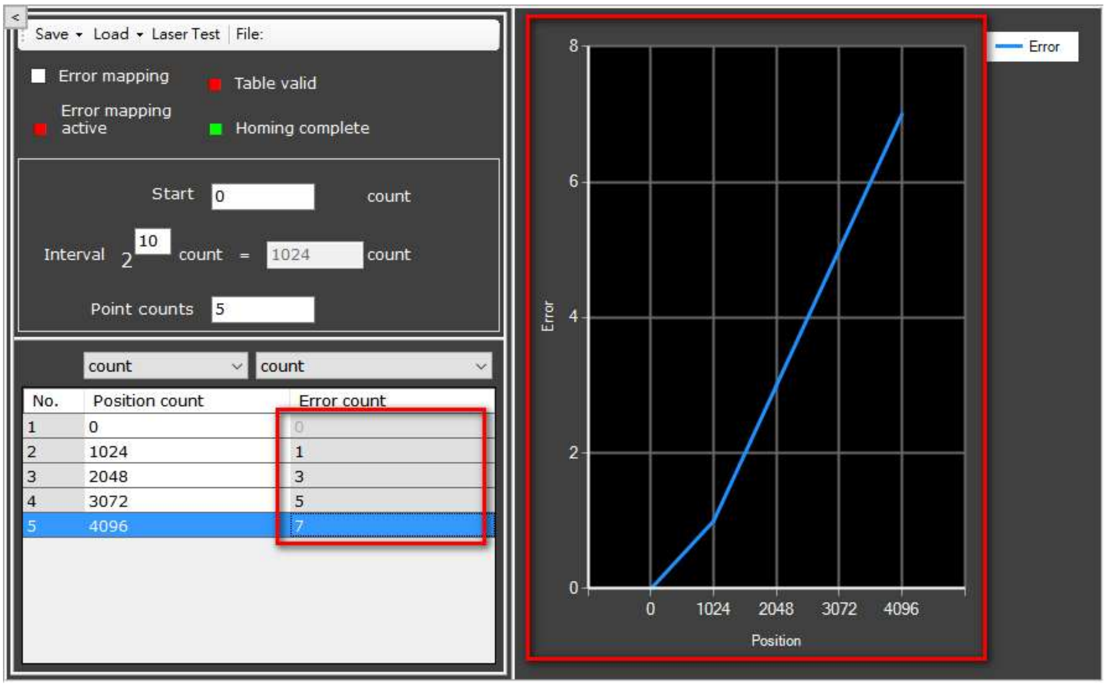
During calibration, the UI will show which cycle and which partition (N) is being performed.



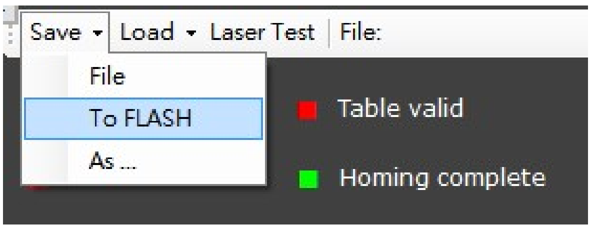
4. After calibration move(s) are finished, click **Back to Error mapping** button.

5. **Record the differences**:

Manually key in the difference(s) between the calibration result and the feedback of presently-used encoder to the **Record Table**. A graph shall automatically show on the right side.



6. **Save to Flash**.



<u>Note</u>:

The configuration parameters **must be saved to "Flash"** to be processed by the driver.

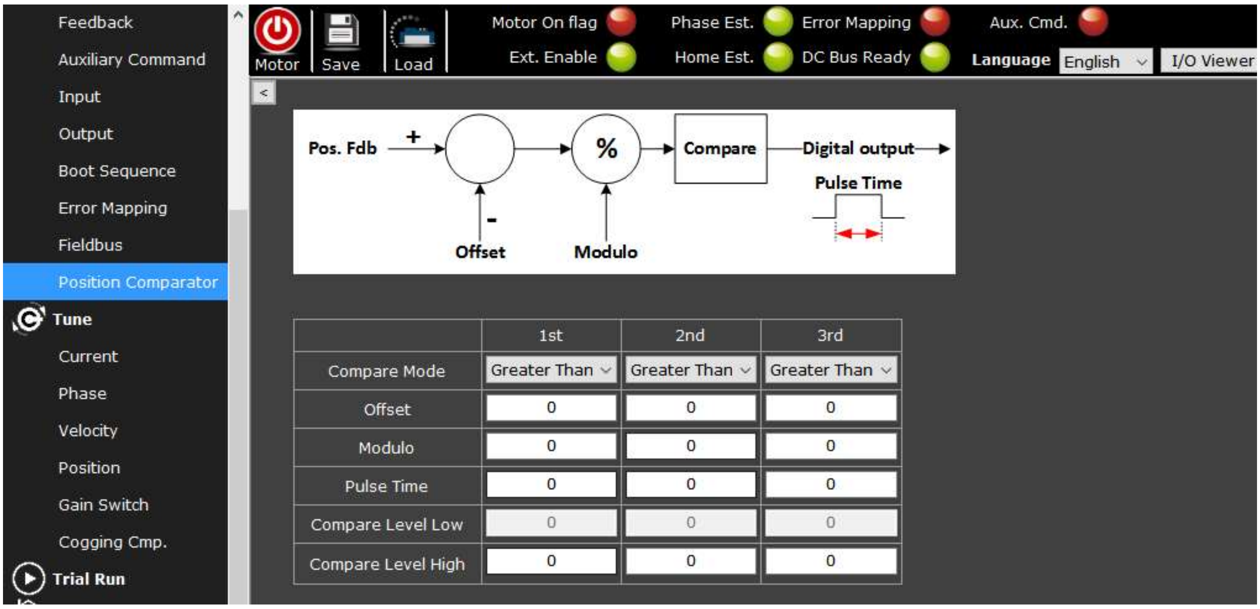
After saving to FLASH, system will check the validity of record table. If valid, the signal icon will turn green.

7. **Tick the "Error Mapping" box**.
   Tick this box so the UI will automatically check if BOTH the signals of [Table valid] and [Homing complete] are high (shown in green). If yes, the Error mapping function will be activated.

Now calibration is successfully completed.

4.10 Position Comparator

This function can be seen as a **virtual limit switch**.

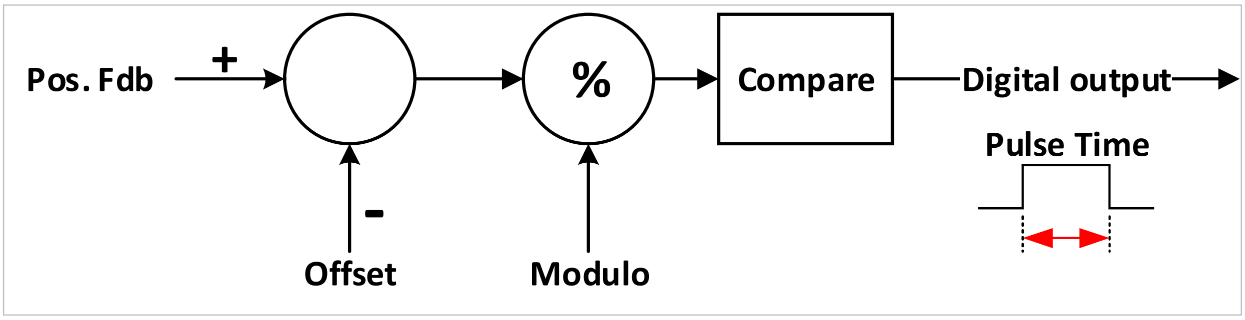


Application:

You can configure up to 3 sets of position conditions as limit switches.

Assign your output pin(s) to **"Pos. Compare 0~2 out"**. When the motor moves into a certain area or onto a spot, the position comparator signal will rise and trigger an output to other external devices.

< Position Comparator Processing >



	1st	2nd	3rd
Compare Mode	Greater Than	Greater Than	Greater Than
Offset	0	0	0
Modulo	0	0	0
Pulse Time	0	0	0
Compare Level Low	0	0	0
Compare Level High	0	0	0

a. Pulse Time:

For how much time (ms) the Position Comparator output signal needs to rise after the conditions you configured are satisfied. **If this is set to 0, the comparator function is deactivated.**

b. Compare Mode

This parameter works with the "Compare Level High/Low" parameters.

- Greater than:

  It corresponds to "Compare Level High". Signal rises if the compare outcome is greater than the configured value.

- Less Than:

  It corresponds to "Compare Level Low". Signal rises if the compare outcome is less than the configured value.

- In Range/Outside Range:

  Signal rises if the compare outcome is between the Compare Level High and Low values.

c. Offset

Offsets against the position feedback value then the comparator will start counting from the difference value.

For example:

If the motor is now at position 10000 cnt, configure “Offset” value to 2000 and set the “Compare Mode” to “Greater than”; then set the “Compare Level High” value to 15000. Hence:

- The comparator will start counting from 8000 (10000 – 2000 = 8000) and needs to move more than 7000 cnt to reach the configured goal (15000).
- The signal will rise when the motor is at pos. **17001** (not 15001).

d. Modulo

Sets a devisor value and use the remainder value for processing.
Hence, the remainder will range from **0** to **devisor value minus 1**.
It will look like the comparator counts from 0 to the devisor value minus 1, and then count from 0 again cyclically.

For example:

If you configure Modulo value to 3000, the comparator will count from 0 to 2999 and to 0, then up to 2999 again cyclically.

1 mod 3000	is 1
2 mod 3000	is 2, ……,
2999 mod 3000	is 2999
3000 mod 3000	is 0
3001 mod 3000	is 1
3002 mod 3000	is 2 ……, and so on.

e. Compare Level Low

A bench-mark to be compared when using “Less Than” in Compare Mode.

f. Compare Level High

A bench-mark to be compared when using “Greater Than” in Compare Mode.

Chapter 5 Tune

There are 4 basic auto-tuning functions in the cpc UI: **Current Loop, Phase, Velocity Loop,** and **Position Loop**. However, it’s possible that the performance of autotuned gains might not suit users’ needs—in this case, users can tune by themselves via adjusting parameters of each individual loop. Users can adjust the value of gain and use **Bode Plot** or **Time Response** function to view and evaluate the tuning result.

When tuning, **PI control** is applied.
In addition, during auto-tuning, only the feed forward in Velocity loop will be adjusted automatically; for other loops the feed forward needs to be adjusted by users.

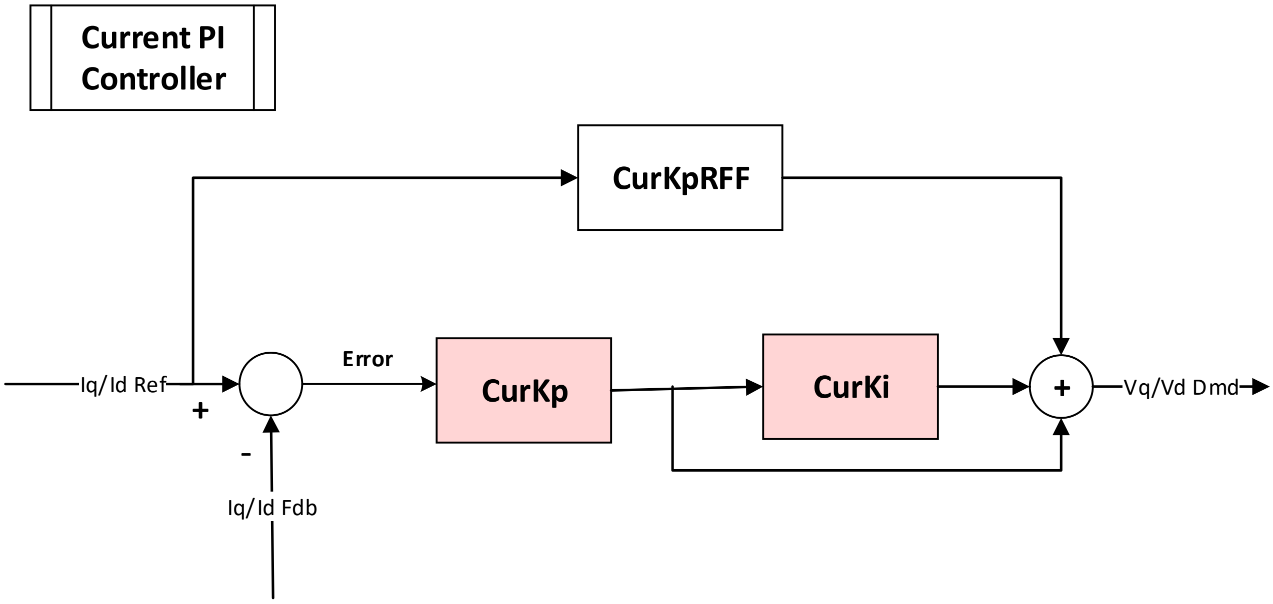
For the Bode Plot and Time Response functions, cpc provides convenient tools to view graphs better:

Loop \ Tool		Zoom Mode	Zoom Reset	Save as txt	Show Prevalue
Current	Frequency Response	V	V	V	--
	Time Response	--	V	V	V
Velocity	Frequency Response	V	V	V	--
	Time Response	--	V	V	V
Position	Time Response	--	V	V	--

See details in related chapters.

5.1 Current

5.1.1 Current Loop Gain



- **CurKpRFF:** Resistance feedforward. [CurKpRFF * Iq/Id Ref]
- **CurKp:** Proportional gain. [CurKp * error(t)]
- **CurKi:** Integral gain. [CurKi * $\int_0^t \mathrm{CurKp} * \ \mathrm{error}(\tau)\mathrm{d}\tau$]

✧ **Vq/Vd Dmd** = (CurKpRFF * Iq/Id Ref) + (CurKp * error (t)) + (CurKi

$*\int_0^t \mathrm{CurKp} * \ \mathrm{error}(\tau)\mathrm{d}\tau)$

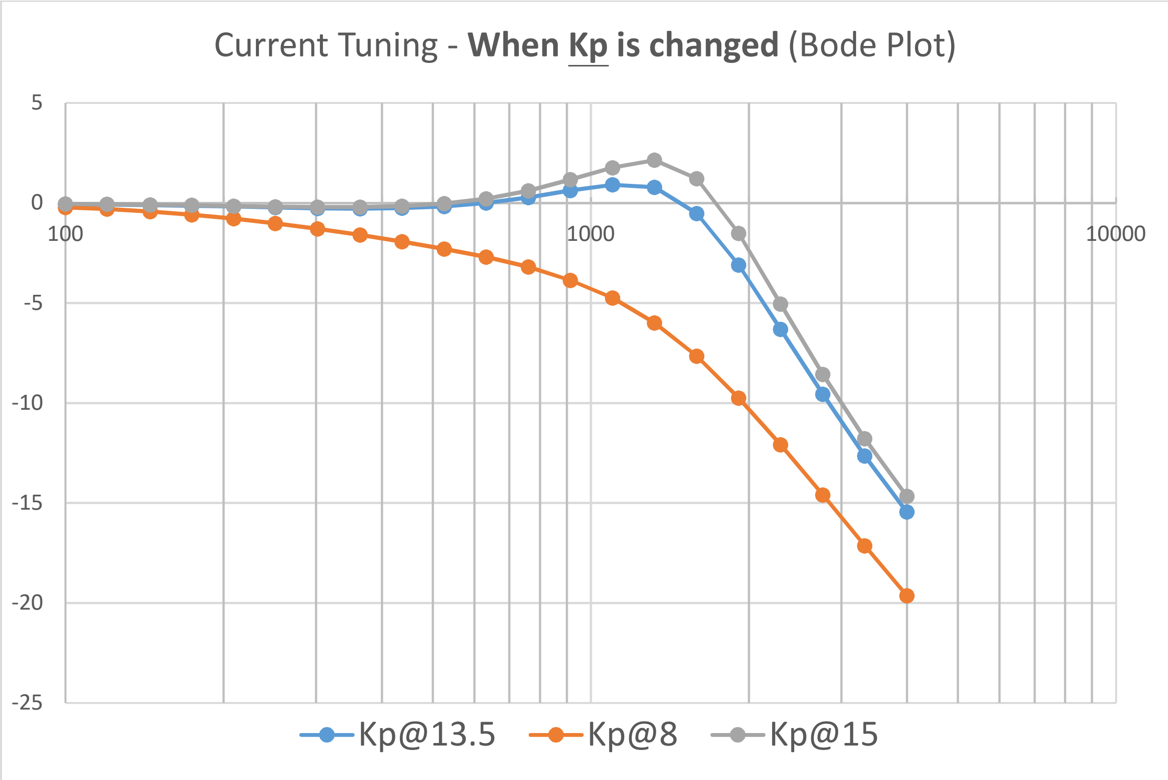
✧ **Error** = Iq/Id Dmd – Iq/Id fdb

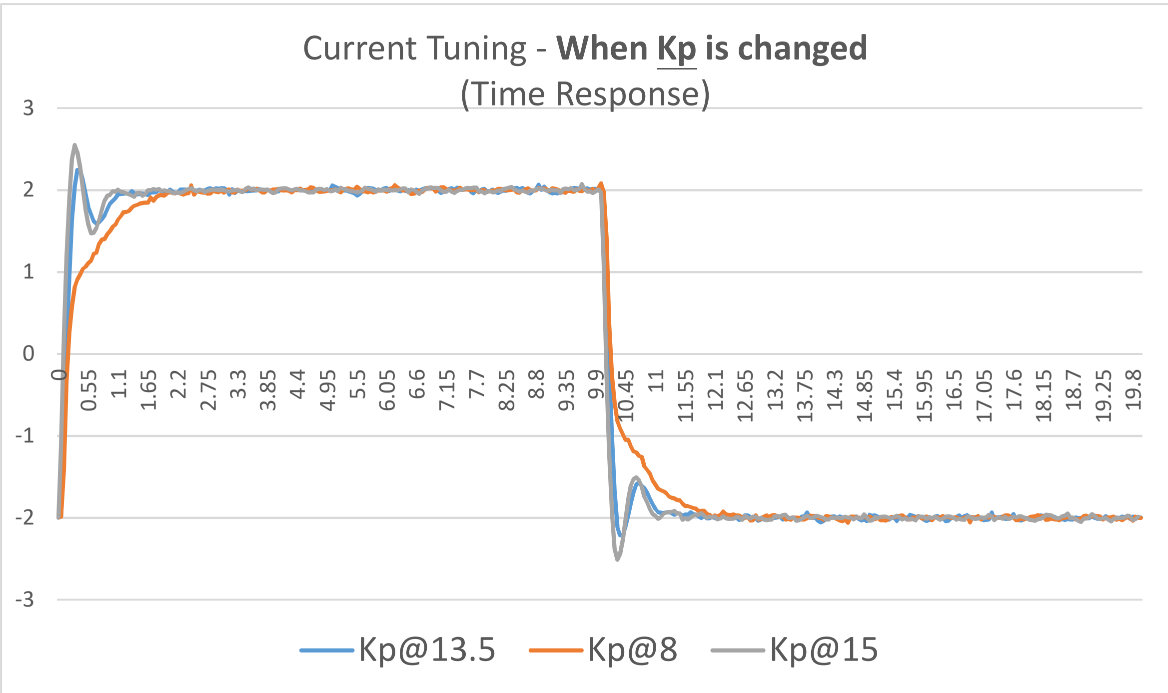
▪ Please check the symbol in chapter 6.1 Monitor.

- Iq/Id: Iq or Id.
- Vq/Vd: Vq or Vd.

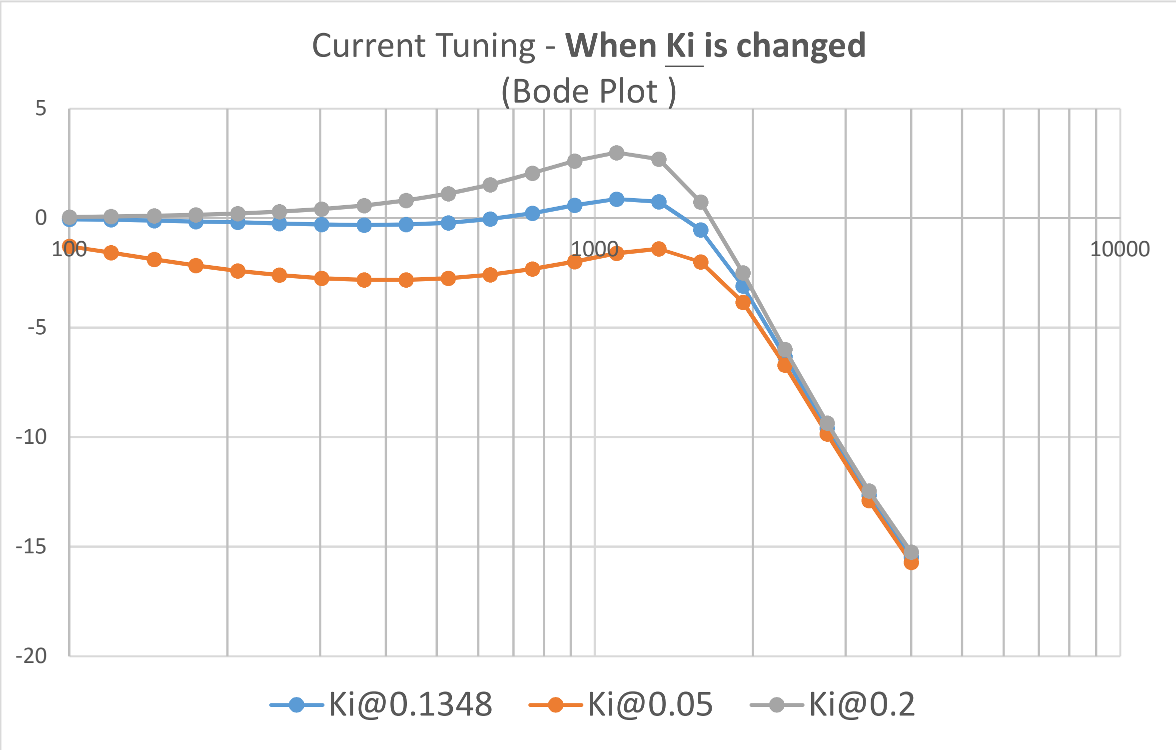
Examples of the result of adjusting Kp and Ki in Current Loop

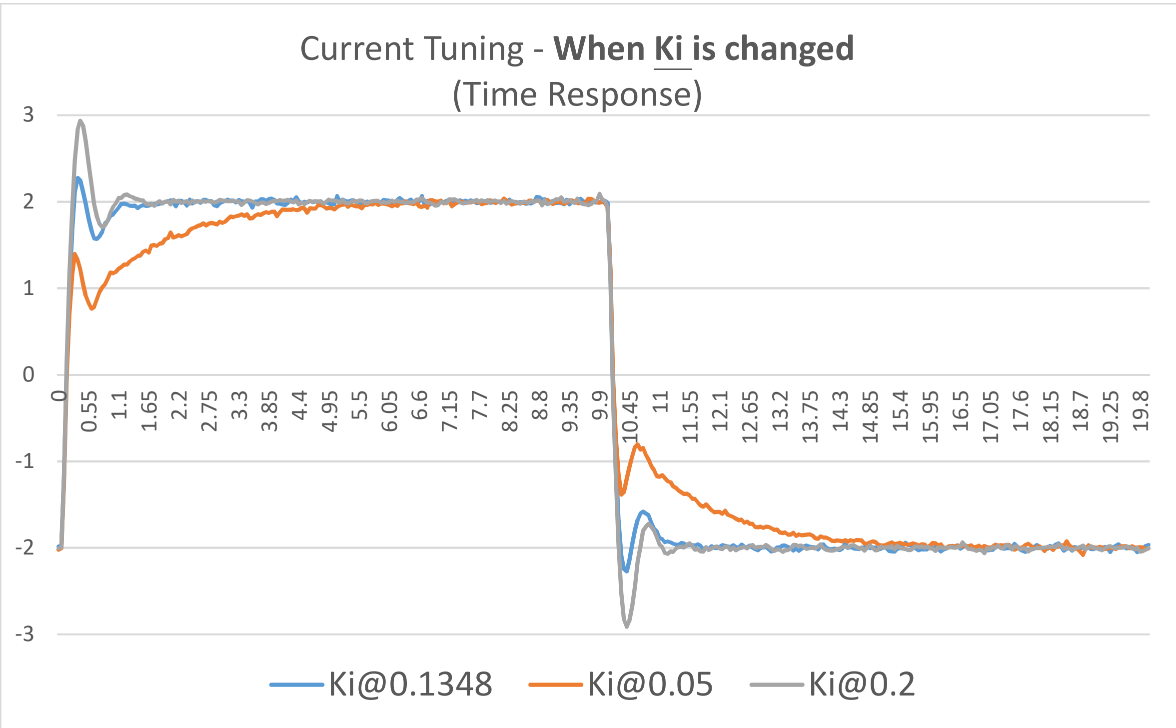
- **Kp:**



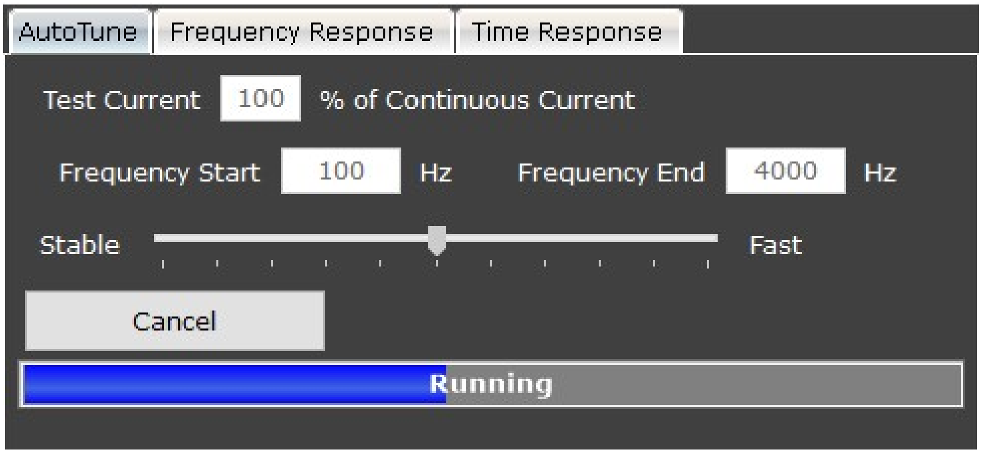


- **Ki**


Current Tuning - When Ki is changed
(Bode Plot)
5
0
-5
-10
-15
-20
100
1000
10000
Ki@0.1348
Ki@0.05
Ki@0.2


Current Tuning - When Ki is changed
(Time Response)
3
2
1
0
-1
-2
-3
0
0.55
1.1
1.65
2.2
2.75
3.3
3.85
4.4
4.95
5.5
6.05
6.6
7.15
7.7
8.25
8.8
9.35
9.9
10.45
11
11.55
12.1
12.65
13.2
13.75
14.3
14.85
15.4
15.95
16.5
17.05
17.6
18.15
18.7
19.25
19.8
Ki@0.1348
Ki@0.05
Ki@0.2

5.1.2 Auto Tune



a. **Test Current:**

Percentage of the continuous current for testing.

b. **Frequency Start/End:**

The frequency range of testing.

c. **Stable/Fast:**

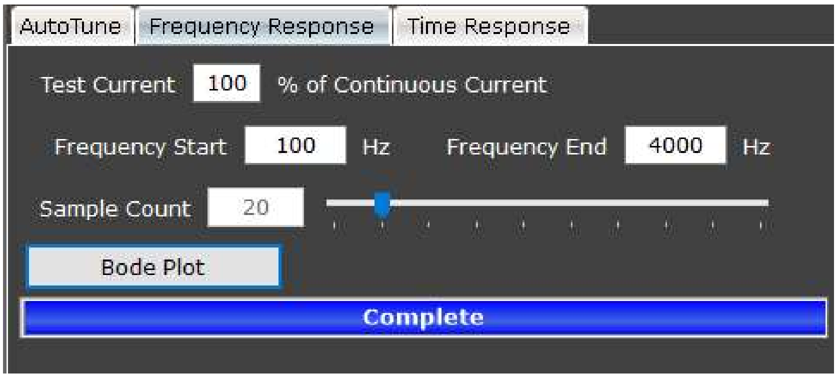
Expected feedback response speed toward demands.

d. **Dead time compensation:**

Compensate the time of the motor idle.

5.1.3 Frequency Response (Bode Plot)

You can plot the frequency response of the current loop.
Click the Frequency Response index, then click the Bode Plot button to start.



a. Test Current:

Percentage of the continuous current for testing.

b. Frequency Start/End:

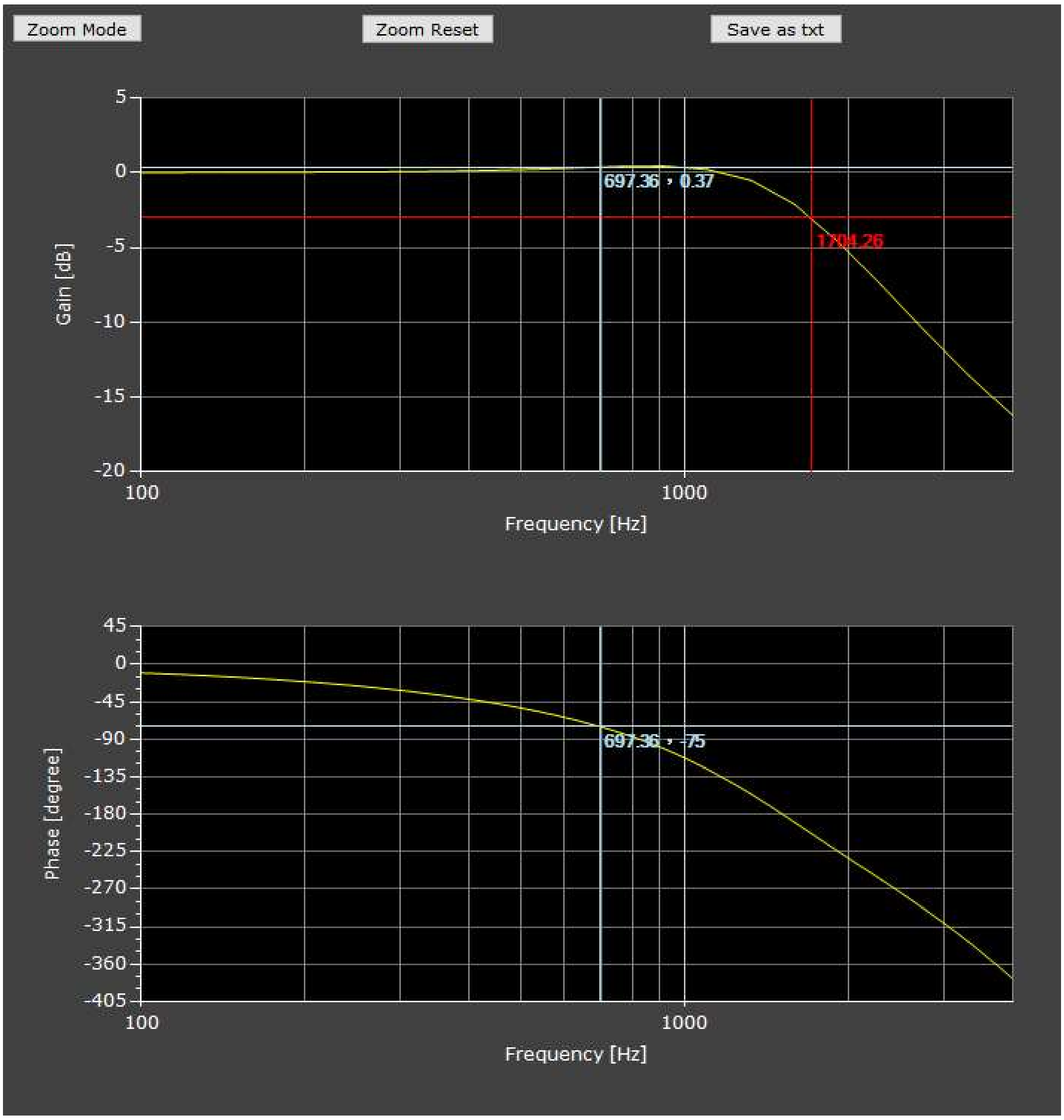
The frequency range.

c. Sample Count:

Defines how many data points (i.e., sample counts) will be captured within the configured frequency start and end. As a result, the more sample counts captured, the more data will be included in the exported .txt file, and the more time is needed to complete bode plot.

d. **Bode Plot:**

Click to start plotting. The result will show on the chart panel like this:



e. **Zoom Mode** 

(on chart panel)**:**

Click to turn off the focus-line in order to apply the zoom function.

f. **Zoom Reset** 

(on chart panel)**:**

Resets zoom.

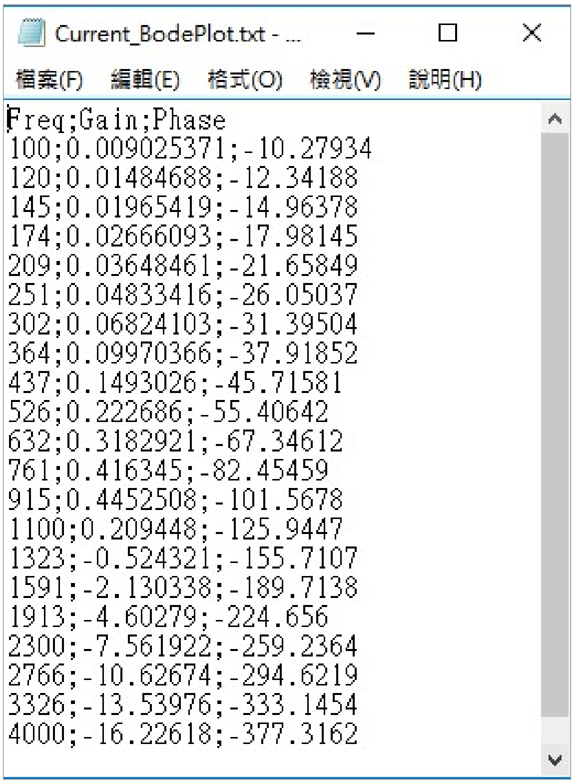
g. Save as txt [Save as txt] (on chart panel)**:**

Exports the graph data in .txt format. Users can rearrange the exported data via Excel.

Note:

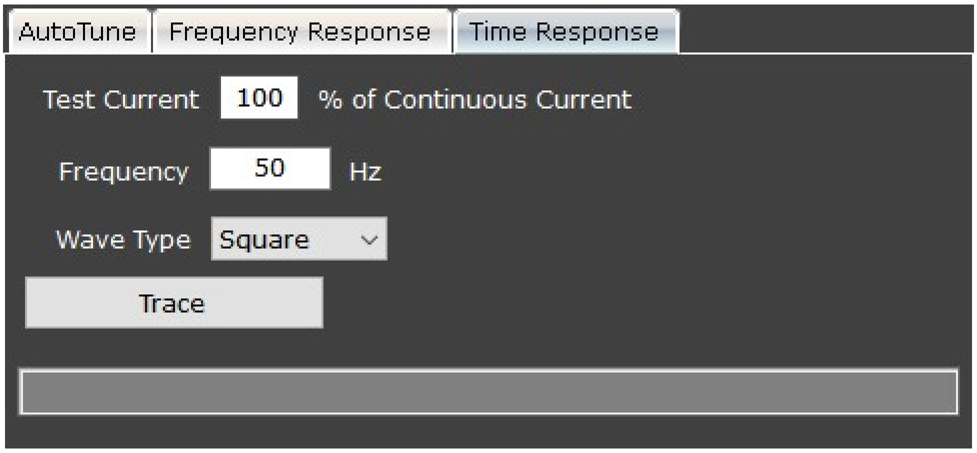
The exported data is separated by semicolon (see picture below). To rearrange the data, users may consider using the “text to column” function in Excel.

<The original exported graph data in .txt>




```
Freq;Gain;Phase
100;0.009025371;-10.27934
120;0.01484688;-12.34188
145;0.01965419;-14.96378
174;0.02666093;-17.98145
209;0.03648461;-21.65849
251;0.04833416;-26.05037
302;0.06824103;-31.39504
364;0.09970366;-37.91852
437;0.1493026;-45.71581
526;0.222686;-55.40642
632;0.3182921;-67.34612
761;0.416345;-82.45459
915;0.4452508;-101.5678
1100;0.209448;-125.9447
1323;-0.524321;-155.7107
1591;-2.130338;-189.7138
1913;-4.60279;-224.656
2300;-7.561922;-259.2364
2766;-10.62674;-294.6219
3326;-13.53976;-333.1454
4000;-16.22618;-377.3162
```

5.1.4 Time Response



a. **Test Current:**

Percentage of the continuous current for testing.

b. **Frequency:**

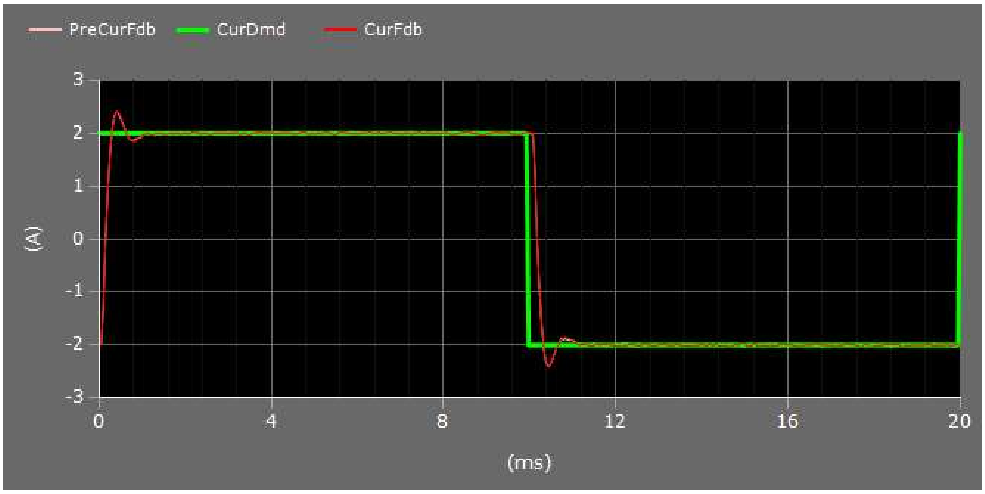
The frequency of the wave.

c. **Wave Type:**

Choose preferred wave form from Square, Triangle, and Sine.

d. **Trace:**

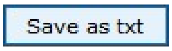
- Click to test and see how the motor feedback differs from the driver's command.
- View the upper graph (on the right side) to compare the difference between Current Demand (green) and Current Feedback (red). The graph helps you evaluate whether the tuning result is ideal.



e. Zoom Reset 

:

To reset zoom.

f. Save as text 

:

Exports the graph data into .txt format.

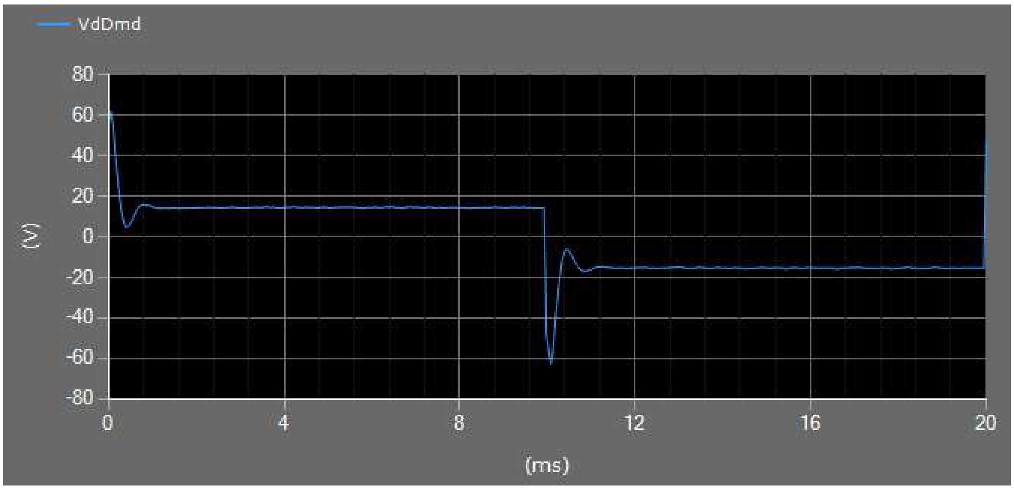
g. 

:

Click to see the previous Current Feedback and Voltage Demand.

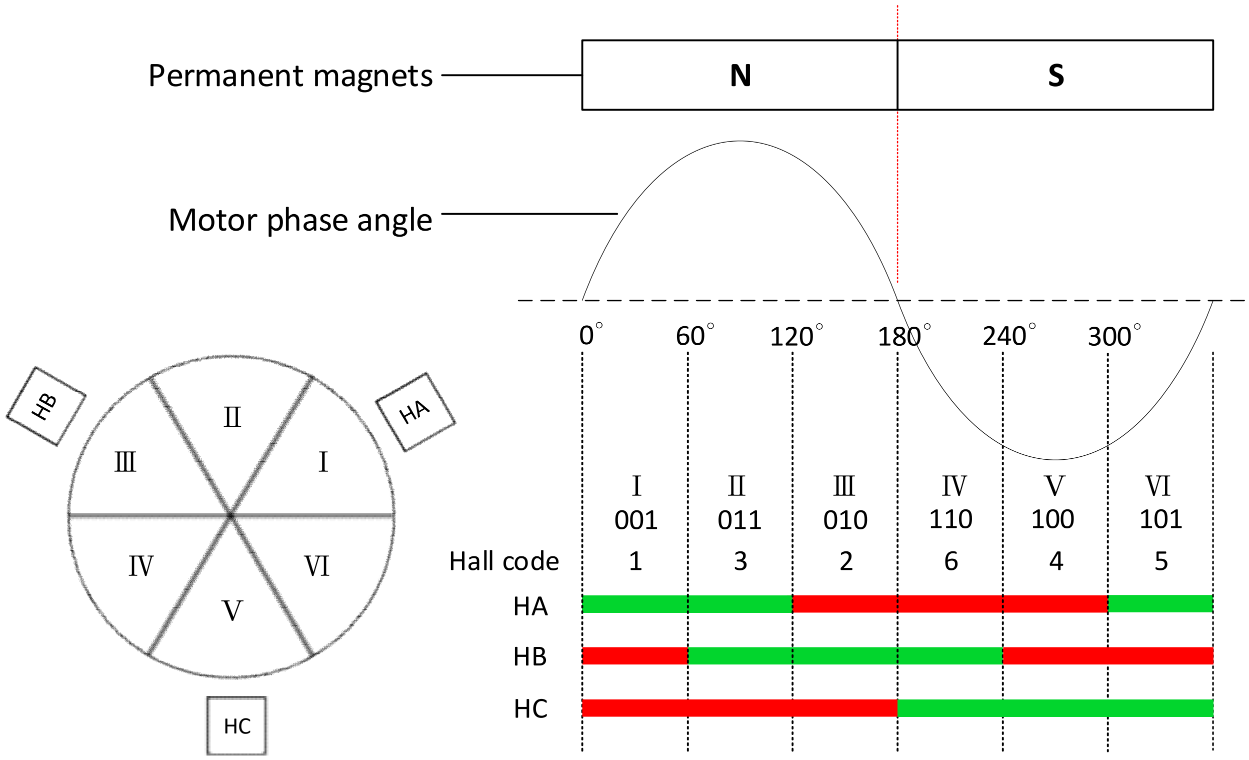
h. VdDmd graph:

To view the voltage demand.



5.2 Phase

The motor utilizes three-phase coils and permanent magnets to generate thrust force. The **coil phase** and the **motor electric angle** should be checked before operation. If you have installed hall sensors, these 3 signals—HA, HB and HC—will show you where the electric angle is now.



5.2.1 Auto Phase

Before operating the motor, there are two things to define:

(1) **User-defined motor direction.**

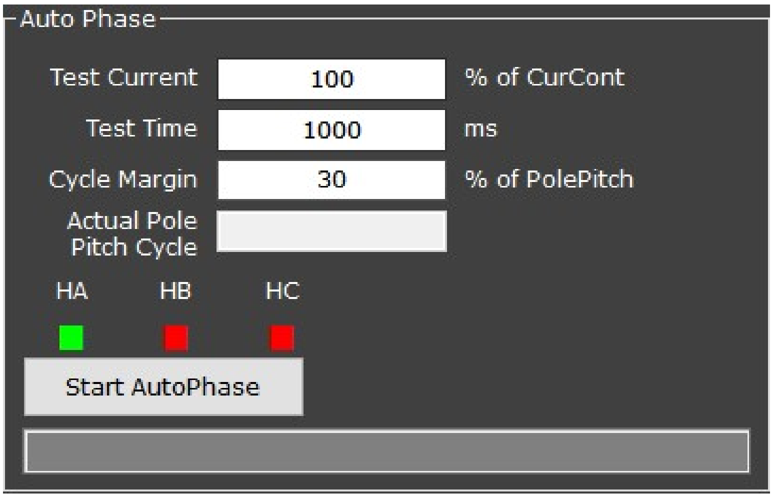
It's important that the user-defined opposite/negative (or CW/CCW) direction and the encoder's counting direction are **consistent**.

Performing Auto Phase function can configure them to be consistent.

(2) **Hall sensor**.

Click "Start AutoPhase" and manually move the motor in positive direction. In addition, the driver will check the pole pitch parameter automatically.

If Autophase function is unusable, please see how to manually set phase in **chapter 5.2.4.**

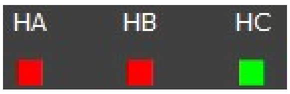


a. Test current / Test time:

Percentage of the continuous current for testing.

b. Cycle Margin % of PolePitch:

Allow this percentage of difference between the configured pole pitch value and the actual pole pitch value.

If hall sensors are installed, this panel 

will show the Hall code status.

5.2.2 Phase Find

Phase Find function is for searching for the **actual** phase automatically.

There are 2 ways to locate motor phase:

- If hall sensors are installed, the driver will calculate the difference between motor and magnets to find phase.
- If not, the driver will execute **forced excitation** to find phase.

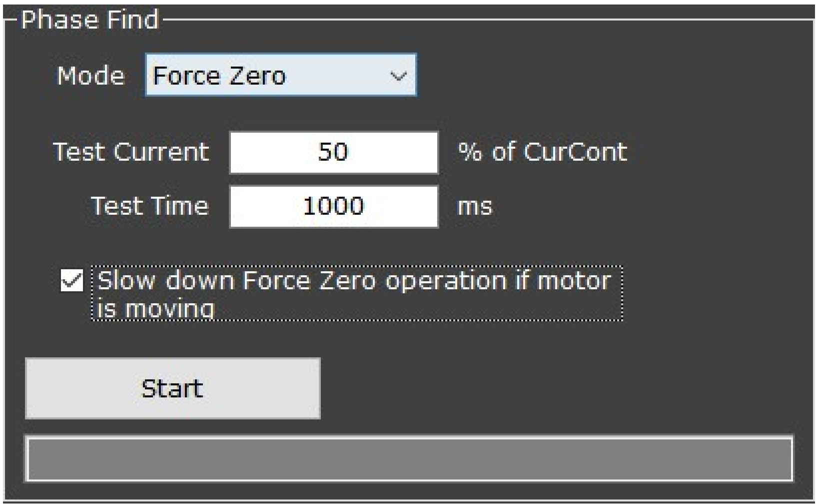
The cpc GUI provides 3 modes to find phase, click 

to perform:

- Force Zero
- Hall
- Abs. Enc. ST. Pos.

5.2.2.1 Force Zero

Click Start button to execute forced excitation.



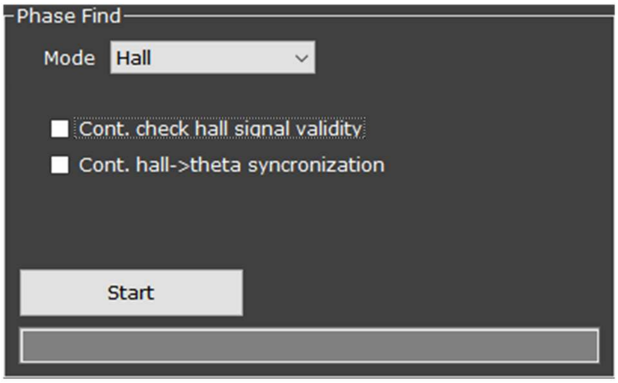
a. Test current / Test time:

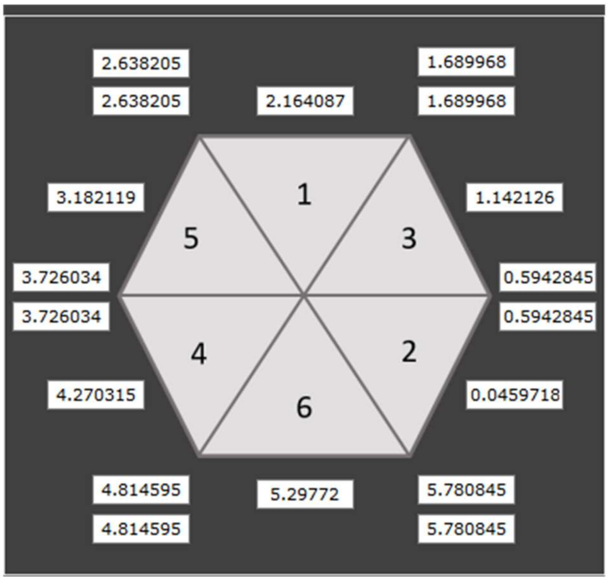
Percentage of the continuous current for testing.

b. Slow down Force Zero operation if motor is moving

Slows down the forced excitation current; hence, the motor will move **gradually** instead of being strongly pulled to place at one time.

5.2.2.2 Hall





a. Cont. check hall signal validity

Checks the validity of hall signal continuously.

If all the three bits of hall code are 1 or 0, a fault event will occur.

b. Cont. hall->theta synchronization

A fail-safe mechanism.

- Unticked:

  After the phase is found using hall sensor, use only the encoder feedback to update the phase angle.

- Ticked:

  Continuously uses the hall sensor data to update the motor phase.

c. Theta table

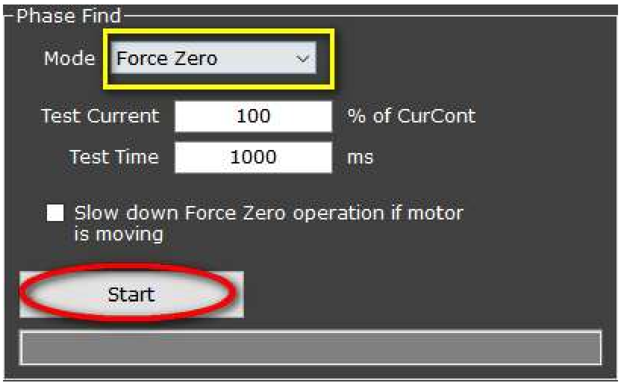
For manually keying in theta values. The columns can accept values from **0 to 2 pi**, unit is radium.

If AutoPhase is unusable, for instance the motor stroke cannot contain a complete motor pole pitch, users will need to manually set phase (see Ch. 5.2.4) and key in each electric angle's Theta value which can be obtained by using Scope function (see how in Ch. 5.2.2.2.1).

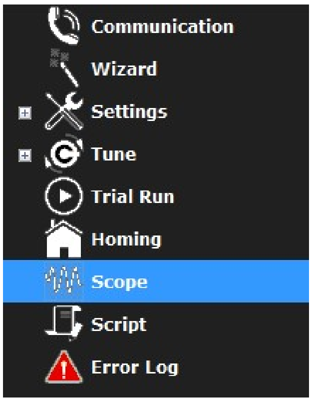
5.2.2.2.1 Steps—Obtaining the motor electric angle theta value

1. Execute Force Zero.

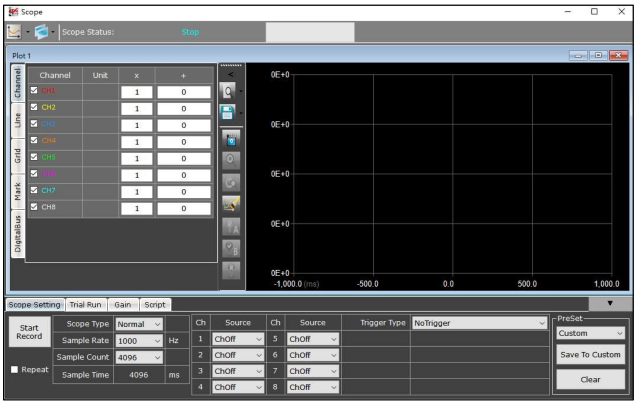
   Go to: Settings >Tune > Phase > Phase Find > select Force Zero mode, then click Start.



2. Go to Scope function.

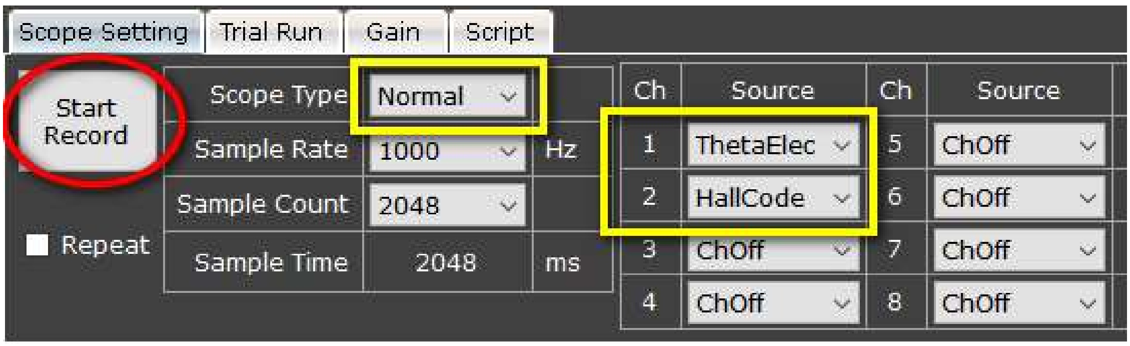


➔

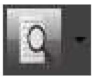
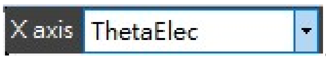


3. In Scope, set the source of channel 1 to **ThetaElec** and the source of channel 2 to **Hallcode**. Next, select "Normal" for Scope Type (see picture in step 4).

4. Click "Start Record" and manually push the motor for a distance.



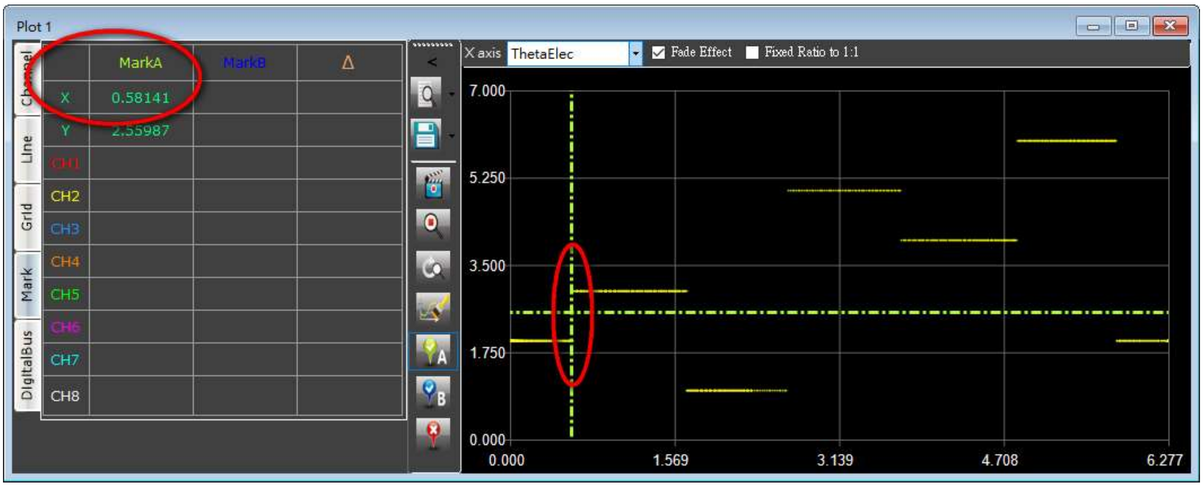
5. Now, wait for the “Receiving data complete” message to show up on top of the scope window. 

6. Select the “PlotXY” option from the Plot Type  list; then set “ThetaElec” to be the X axis value .

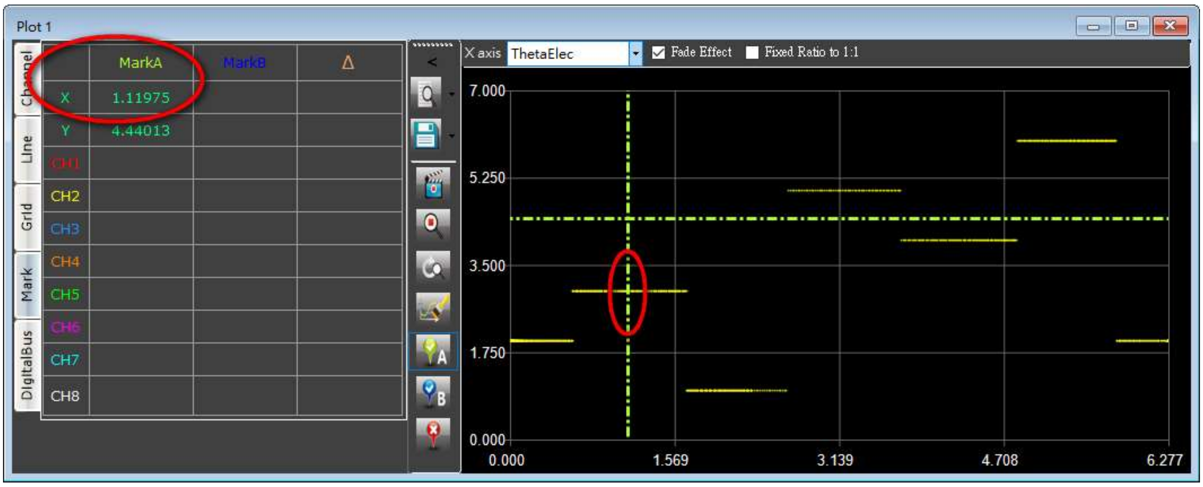
7. Next, click on the Mark Tab, the screen should appear like this:

8. Click on the Mark A icon . Put mouse curser on the junction of two signal sections and on the middle of a signal section (see images below). Users will need to note down the Mark A x-axis values shown on the Mark Tab.

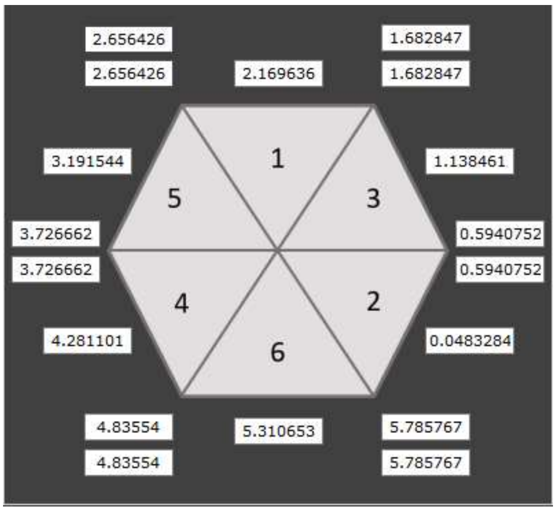
(↓ the junction of two signal sections)



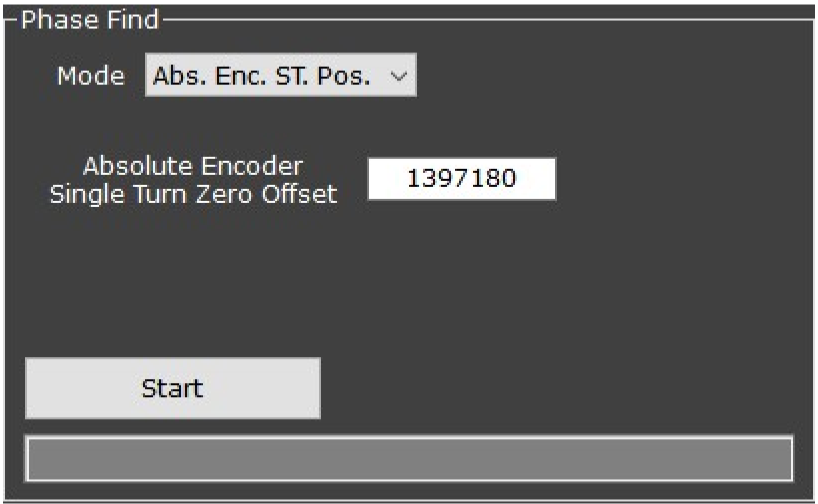
(↓ the middle of a signal section)



9. Keep moving curser and noting down all the Mark A x-axis values for the Theta table columns.

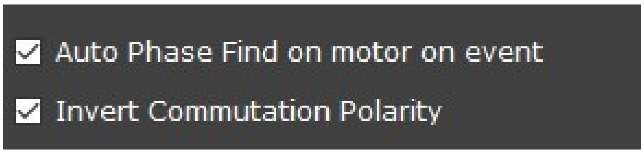


5.2.2.3 Abs. Enc. ST. Pos.



To operate with an absolute encoder, execute AutoPhase first in order to initially locate what position count corresponds to "0" electric angle (i.e. theta value 0)—this position count value is the "Absolute Encoder Single Turn Zero **Offset**" value.

5.2.3 Other Settings of Phase



a. Auto Phase Find on motor on event

Finds phase automatically on motor-on event.

b. Invert Commutation Polarity

CAUTION:

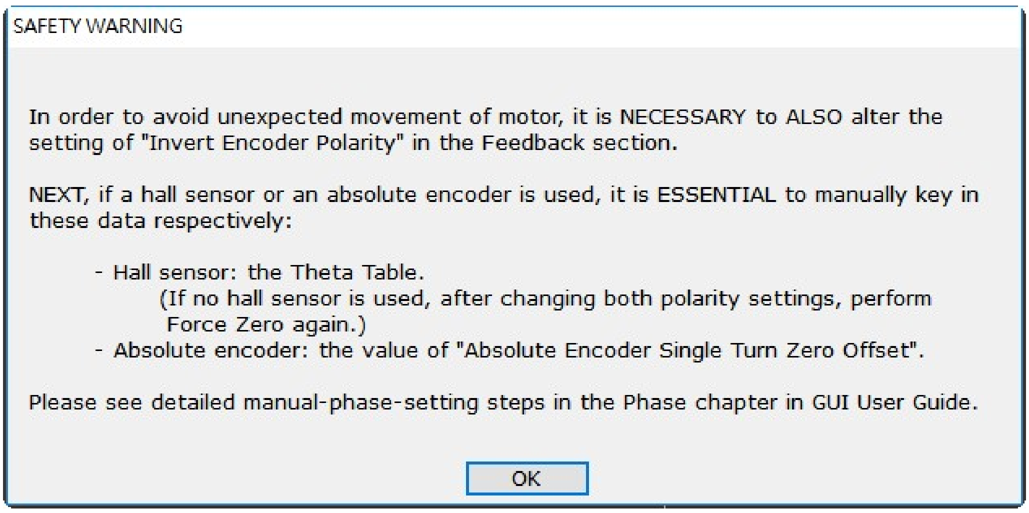
If the setting of “Invert Commutation Polarity” is changed, for the sake of safety, the setting of encoder polarity in Feedback section MUST also be altered. Otherwise, the motor may perform unexpected movements.

Next, if a hall sensor or an absolute encoder is used, it is ESSENTIAL to manually key in the following data respectively:

- Hall sensor: the Theta Table. (If no hall sensor is used, after changing BOTH polarity settings, perform Force Zero again.)
- Absolute encoder: the value of “Absolute Encoder Single Turn Zero Offset”.

Please see detailed steps of manual-phase-setting in **chapter 5.2.4.**

A warning will show up when trying to change polarity.



5.2.4 Manually Set Phase

Manually setting phase is to let users define the counting direction of encoder.

Steps:

1. Perform forced excitation first (IMPORTANT).

   (Go to: Tune > Phase > Phase Find > Force Zero mode.)

2. Change BOTH the settings of Invert Encoder Polarity* and Invert Commutation Polarity**.

   *: see **chapter 4.5.1**, point (b).

   : see **chapter 5.2.3, point (b).

3. Re-define phase:
   - If not using hall sensor:

     Perform “Force Zero” again.
   - If using a hall sensor:

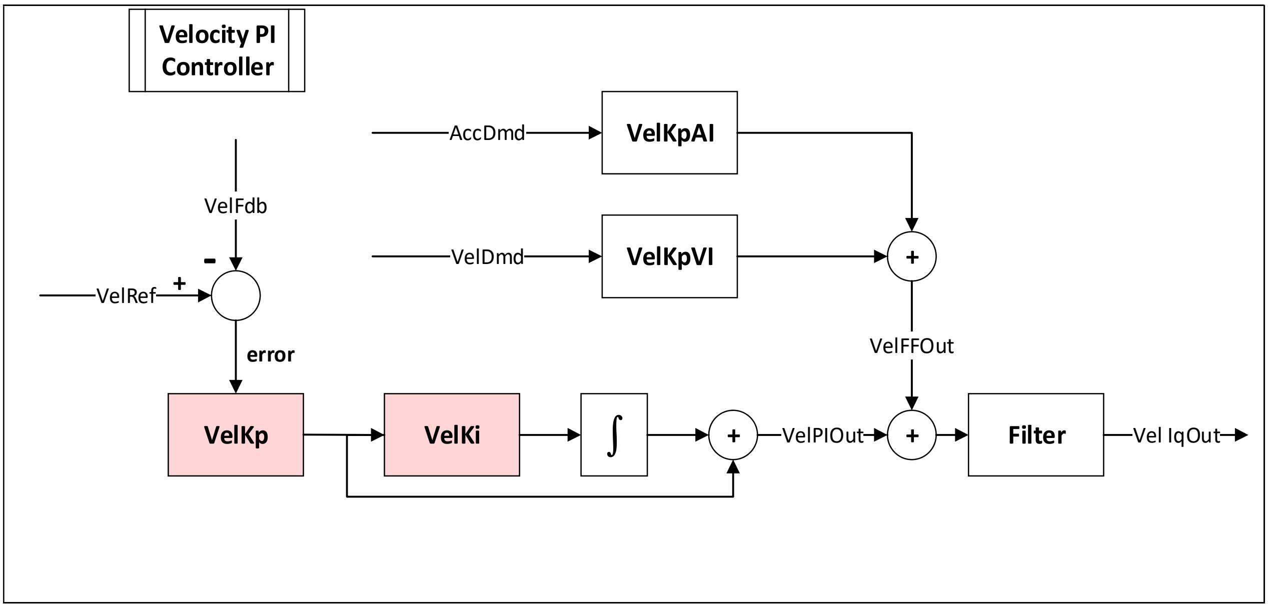
     Manually key in the Theta Table (see how in **chapter 5.2.2.2.1**).
   - If using an absolute encoder:

     Manually key in the value of “Absolute Encoder Single Turn Zero Offset” *(To be elaborated).*

5.3 Velocity

The flow diagram of velocity signal loop is shown in **chapter 5.3.1**.
The cpc firmware provides 3 sets of Kp and Ki. In addition, the "Auto Calculator" (see **chapter 5.3.2**) and the "Filter" (see **chapter 5.3.3**) are also useful tools for finer tuning.

5.3.1 Velocity Loop Gain

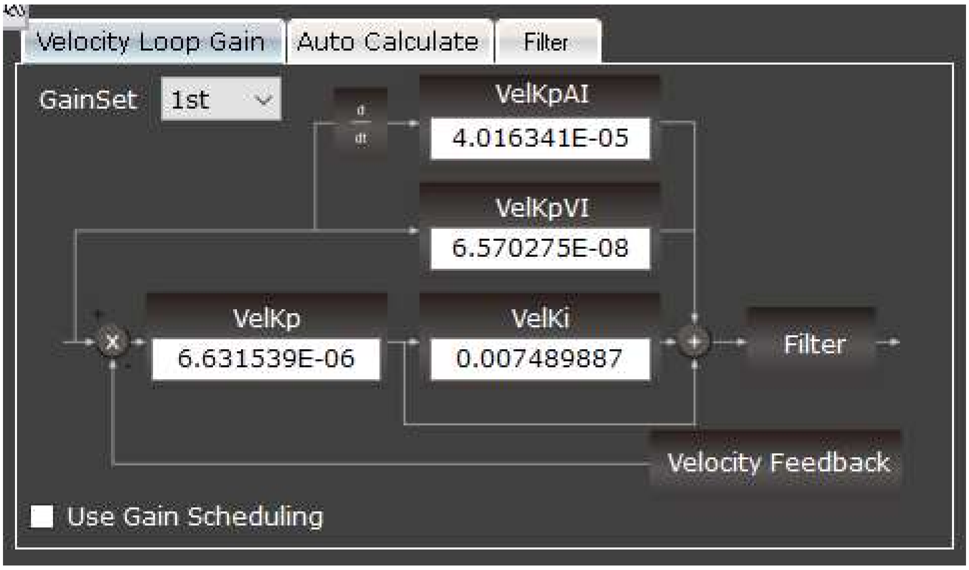


- **VelKpAI:** Acceleration feedforward gain, proprotional to load inertia. [VelKpAI * AccDmd]
- **VelKpVI:** Friction feedforward. [VelKpVI * VelDmd]
- **VelKp:** Proportional gain. [VelKp * VelErr]
- **VelKi:** Integral gain. [VelKi * $\int_0^t \mathrm{VelKp} * \mathrm{VelErr}(\tau)d\tau$]

Note:

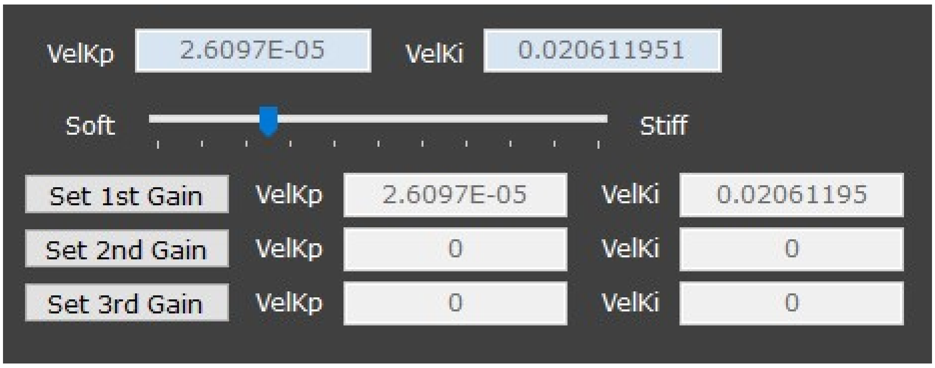
✧ VelIqOut = { VelKpAI * AccDmd + VelKpVI * VelDmd + VelKp * VelErr + VelKi * $\int_0^t \mathrm{VelKp} * \mathrm{VelErr}(\tau)d\tau$ } * Filter(ω)

▪ Please check the symbol in 6.1 Monitor.



Please refer to the line graphs in chapter 5.1.1 about how the results will be by adjusting Kp and Ki gains.

5.3.2 Auto Calculate

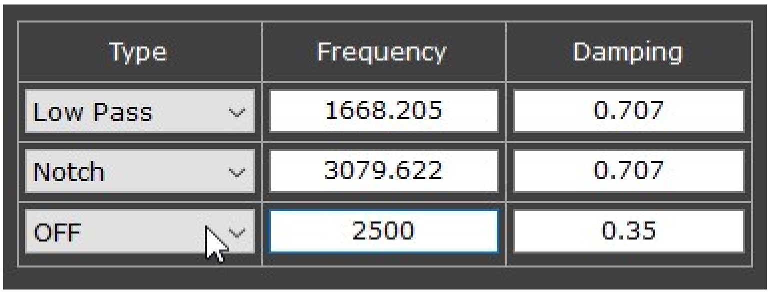


Drag the slider bar 

to generate a group of gains and then click 

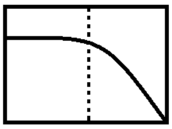
button—these auto calculated gains will be set into the driver.

5.3.3 Filter



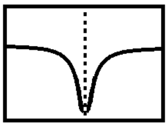
- **Low pass filter**:

  Attenuates signals with frequencies higher than the cutoff frequency.

- **Notch filter**:

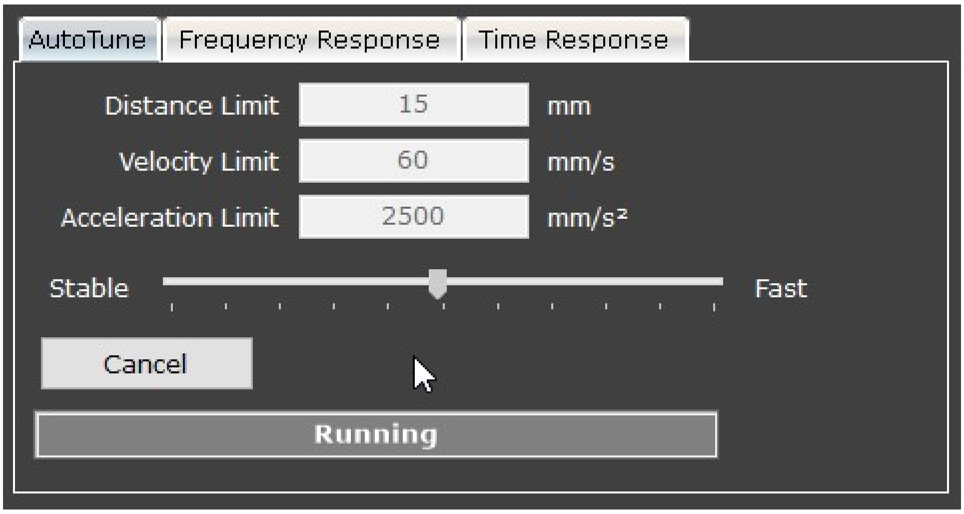
  Attenuates signals with frequencies in a specific range.

- **Off**:

  No filter.

5.3.4 Auto Tune



The motor will move back and forth while tuning velocity. Set the moving distance limit and click "Tune" button, the UI will tune the velocity loop gain automatically.

a. **Distance Limit:**

Maximum testing distance.

b. **Velocity Limit:**

Maximum testing velocity.

c. **Acceleration Limit:**

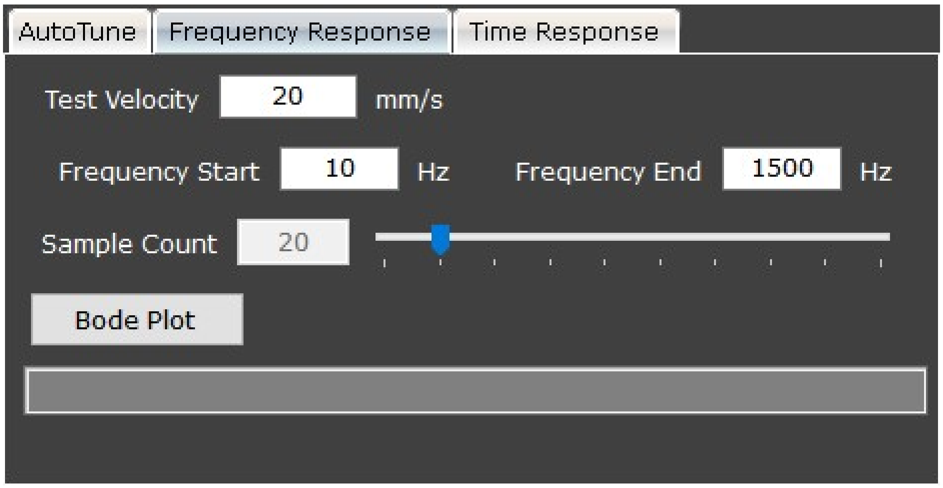
Maximum testing acceleration.

d. **Stable/Fast:**

User's expected feedback response speed toward demands.

5.3.5 Frequency Response (Bode Plot)

This function uses different frequencies of sine input from low frequency to high frequency then plot the response. Click "Bode Plot" to see the frequency response.



a. Test Velocity:

The testing velocity.

b. Frequency Start / End:

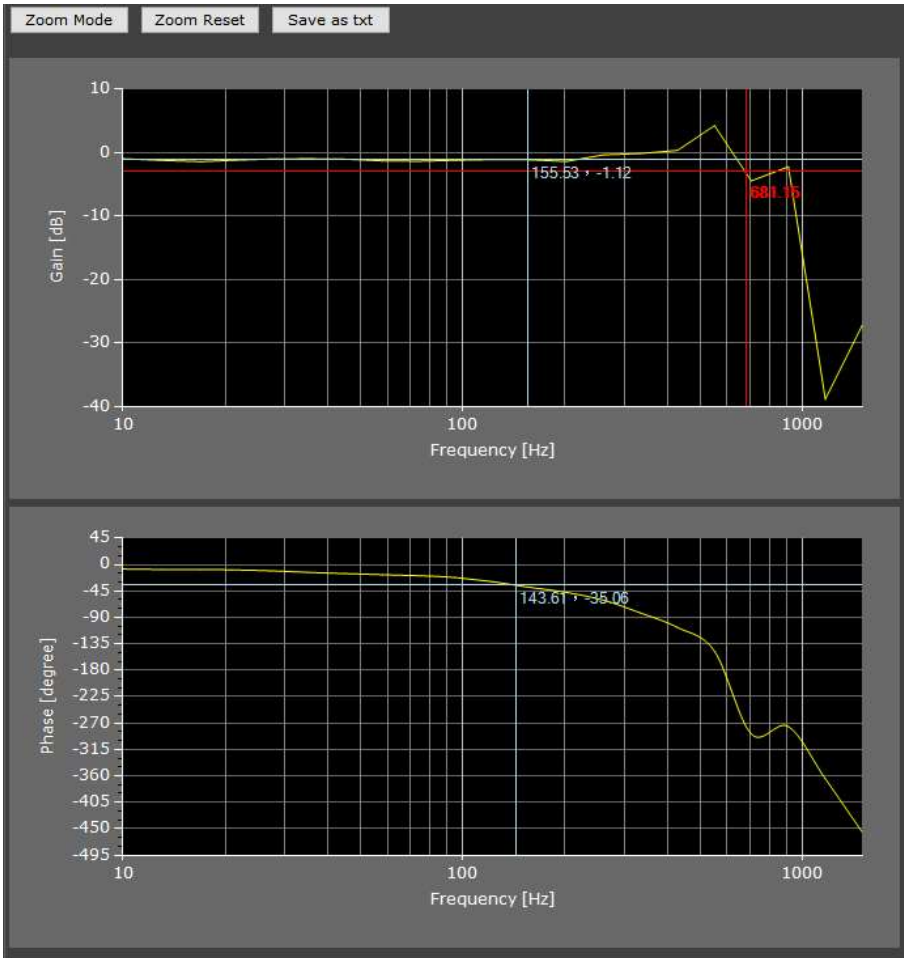
Range of the frequency.

c. Sample Count:

Defines how many data points (i.e., sample counts) will be captured within the configured frequency start and end. As a result, the more sample counts captured, the more data lines the exported .txt file will include, and the more time is needed to complete bode plot.

d. Bode Plot:

Click to start plotting. The result will show on the chart panel like this:



e. Zoom Mode Zoom Mode (on chart panel):

Click to turn off the focus-line in order to apply the zoom function.

f. Zoom Reset Zoom Reset (on chart panel):

Reset zoom.

g. Save as txt Save as txt (on chart panel):

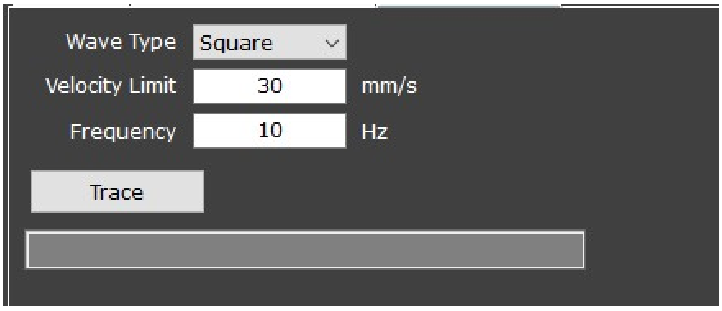
Export the graph data in .txt format. Users can rearrange the exported data via Excel.

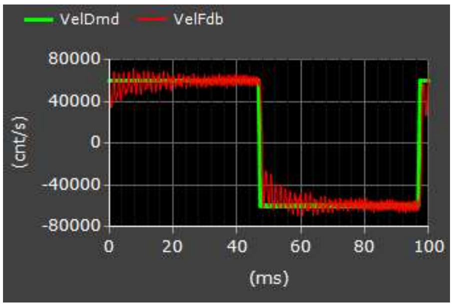
Note:

The content of exported data is separated by semicolon. To rearrange the data, users may consider using the “text to column” function in Excel.

5.3.6 Time Response

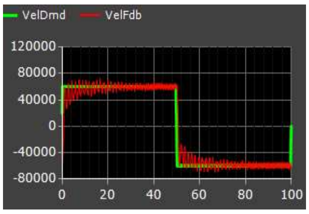
Click "Trace" to test and see how the motor feedback differs from the driver's command. View the upper graph to compare the response between the Velocity Demand (green) and the Velocity Feedback (red). It can help you observe whether the tuning result is ideal.

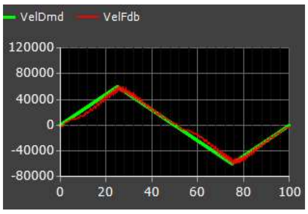


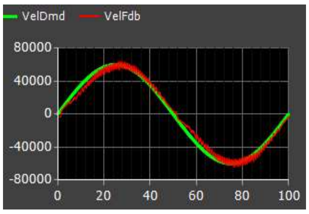


a. Wave Type:

Choose what type of wave form you want to trace: square, triangle, or sine wave forms.







b. Velocity Limit:

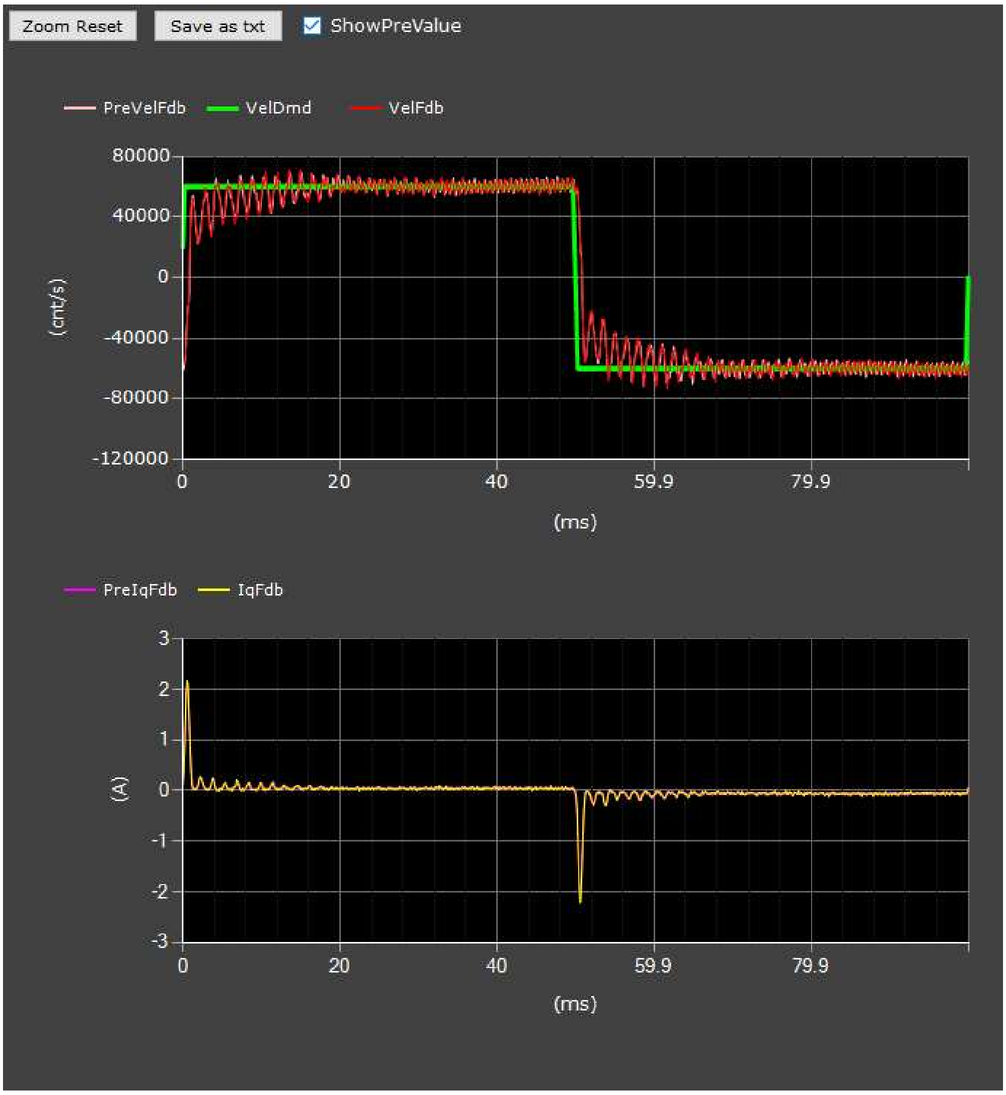
Maximum testing speed.

c. Frequency:

Testing frequency.

d. Trace:

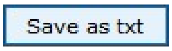
Click to start tracing. The chart panel should appear like this:



i. **Zoom Reset** 

:

To reset zoom.

j. **Save as text** 

:

Export the graph data in .txt format.

k. 

:

Click to see the previous Velocity Feedback and Iq Current Feedback.

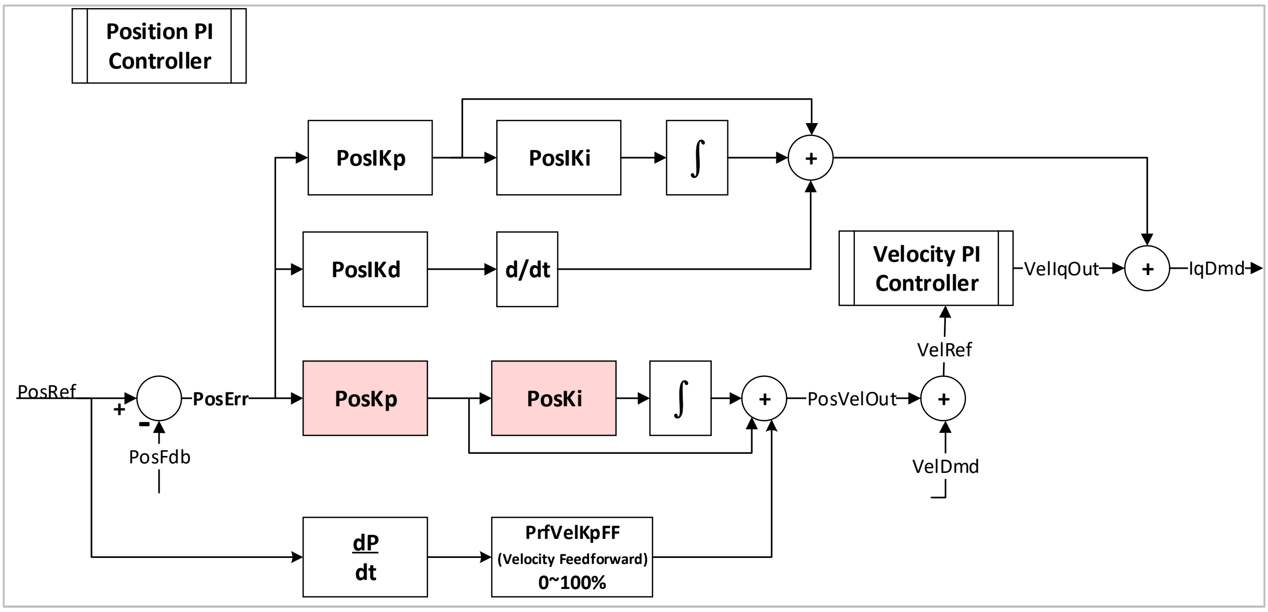
l. **IqFdb graph**:

To view the Iq current feedback.

5.4 Position

Position loop gain provides 3 sets of tuning parameters. Due to that the position loop gains tuning is integral in the velocity loop tuning procedure, after the velocity gains are tuned, the UI will automatically tune the position loop gains.

5.4.1 Position Loop Gain



- **PosIKp:** Position error to current demand proportional gain. [PosIKp * PosErr]
- **PosIKi:** Position error to current demand integral gain. [PosIKi * $\int_0^t PosErr(\tau)d\tau$]
- **PosIKd:** Position error to current demand derivative gain. [PosIKd * $\frac{d\,PosErr}{dt}$]
- **PosKp:** Proportional gain. [PosKp * PosErr]
- **PosKi:** Integral gain. [PosKi * $\int_0^t \mathrm{PosKp} * \mathrm{PosErr}(\tau)d\tau$]
- **PrfVelKpFF**: Profile velocity Kp feedforward.

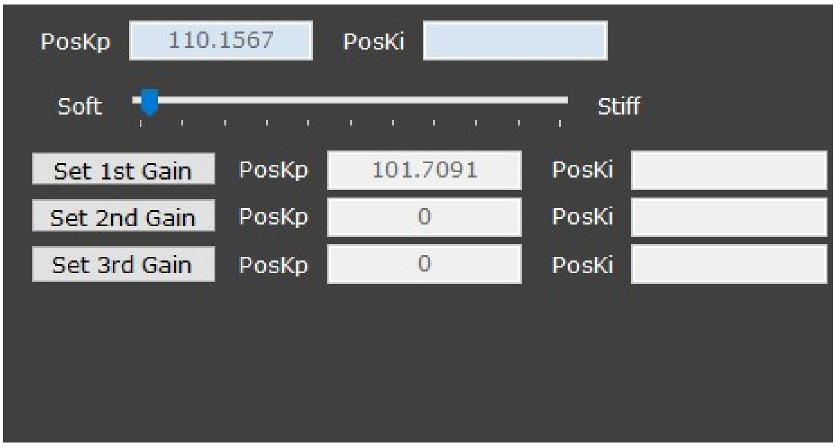
Note:

✧ PosVelOut = PosKp * PosErr + PosKi * $\int_0^t \mathrm{PosKp} * \mathrm{PosErr}(\tau)d\tau$

✧ IqDmd = VelIqOut + PosIKp * PosErr + PosIKi * $\int_0^t PosErr(\tau)d\tau$ + PosIKd * $\frac{d\ PosErr}{dt}$

▪ Please check the symbol in chapter 6.1 Monitor.

5.4.2 Auto Calculate

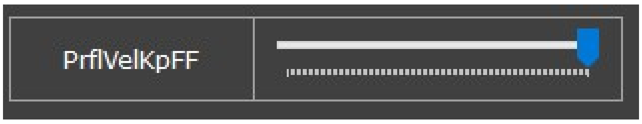


Drag the slider bar 

to generate a group of gain then click the 

button—these auto calculated gains will be set into the driver.

5.4.3 Other

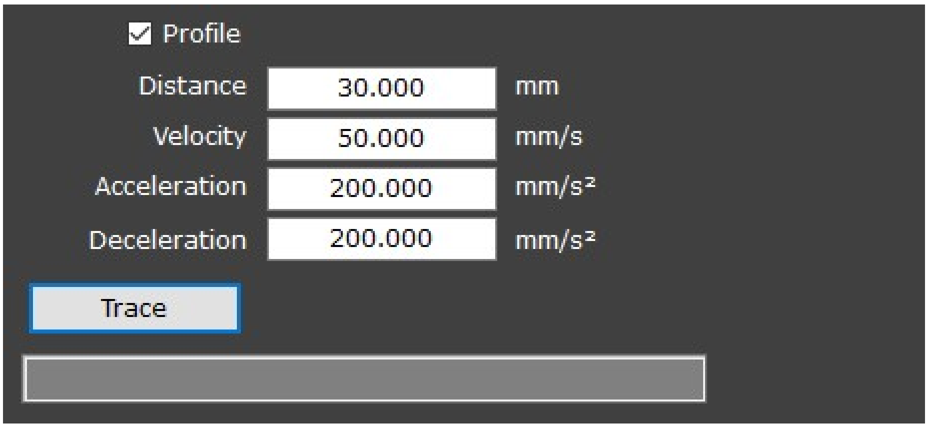


a. PrfVelKpFF: Profile Velocity Kp Feedforward

Ranges 0~100 %. Default is 100%.

- The profile velocity value is multiplied by the PrfVelKpFF value.
- The primary effect of this gain is **to decrease following error**.

5.4.4 Time Response



a. Profile:

Tick to apply the Acceleration and Deceleration.

b. Distance:

The moving distance of testing.

c. Velocity:

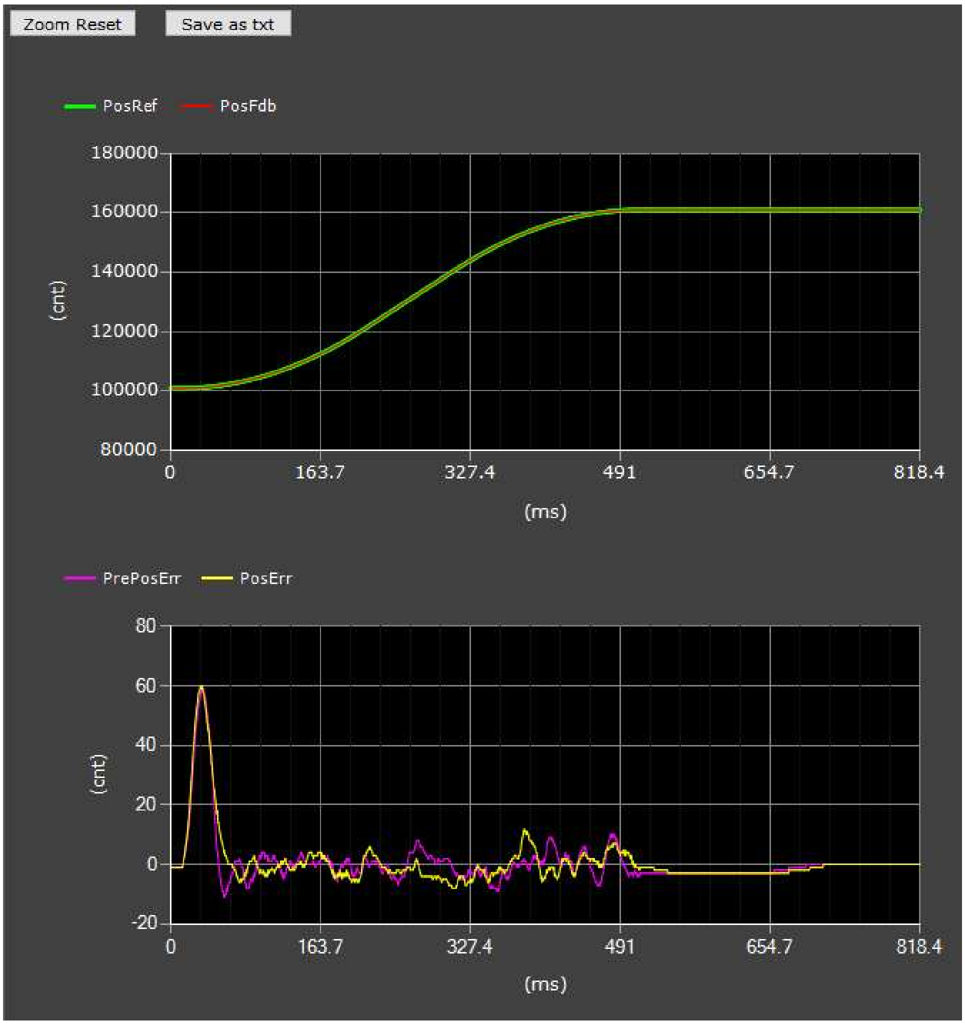
The maximum testing velocity.

d. Acceleration/Deceleration:

The slope of the velocity.

e. Trace:

Click to start tracing, the chart panel should appear like this.



f. Zoom Reset:

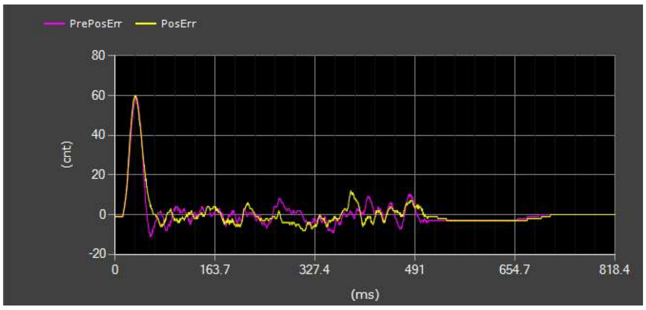
Resets zoom.

g. Save as txt:

Exports the graph data in .txt format.

h. PrePosErr / PosErr:

Shows the present and the previous differences (i.e., error) between the Position Demand and Position Feedback.



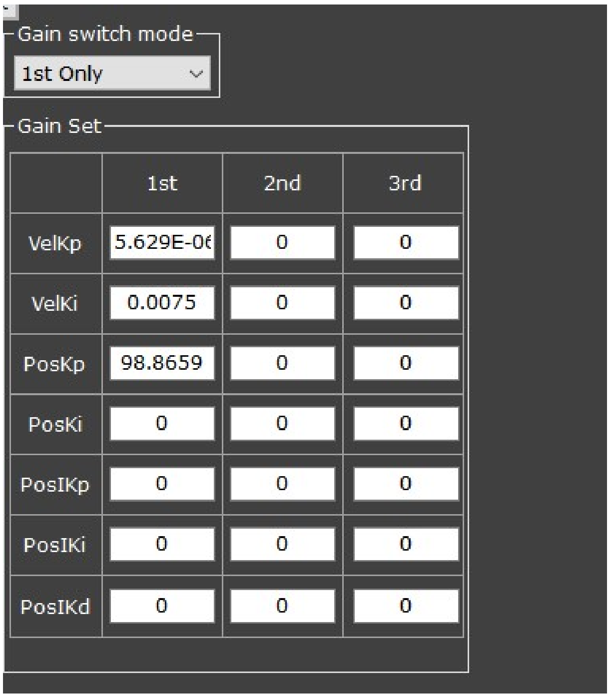
5.5 Gain Switch

The cpc UI provides 3 gain-sets.
Users can use Gain Switch functionality to shift gain-sets when certain user-defined conditions are reached. For instance, users can use conditions of **input signal**, **level trigger** (see **chapter 5.5.3**, the Switch Level), and **target reach flag** as triggers to shift gain-sets.

To start, select Gain Switch Mode and, if applicable, the Switch Source.

5.5.1 Single-set mode

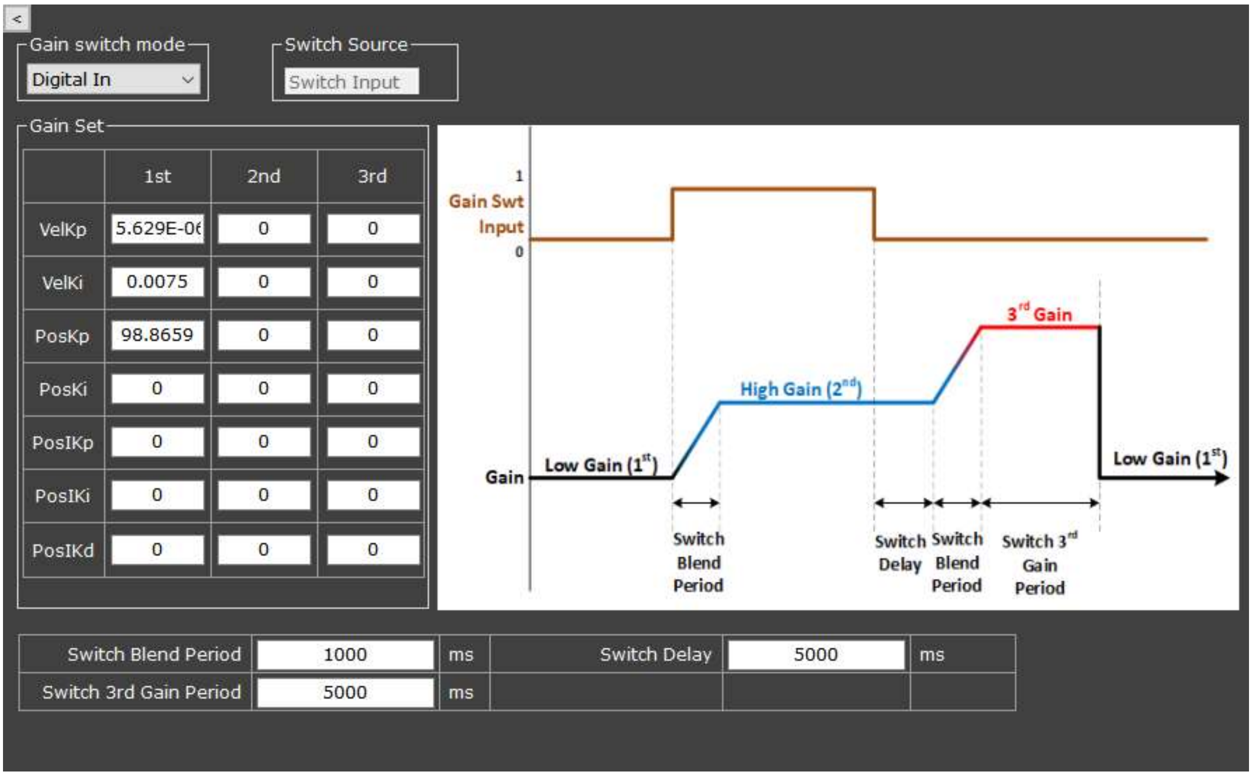
Only one gain-set will be used. Configure the parameters of that gain-set.



Options:

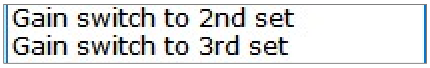
1^{st} Only: Use the 1^{st} gain-set only.
2^{nd} Only: Use the 2^{nd} gain-set only.
3^{rd} Only: Use the 3^{rd} gain-set only.

5.5.2 Digital In



Use digital signal input to determine which gain-set to be applied.

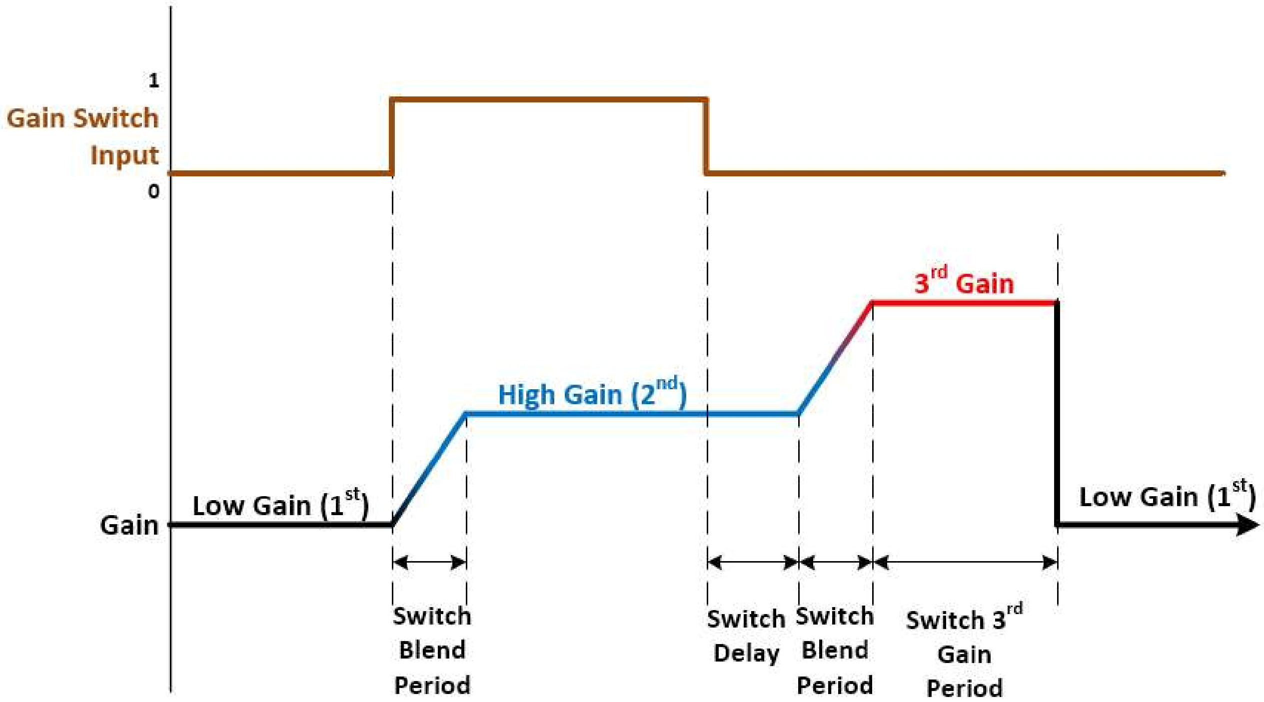
Go to Settings> Input (see **chapter 4.7.1**) to assign your input pin as "Gain switch to 2nd (or 3rd) set".



<u>For example</u>:

If there is only one input pin and it is set as "Gain switch to 2nd set," Gain-Set will switch to the 2nd one when a rising edge triggers signal input to the driver and will automatically switch to the 3rd one when a falling edge occurs; see the diagram below. For other occasions, please see the flow chart <Gain-set Switching Flow Chart> shown on subsequent page.

<Diagram—Gain-set Switching Determined by Digital Input>



a. Switch Blend Period:

The transition time (ms) from the 1st gain-set to the 2nd one, or from the 2nd to the 3rd.

b. Switch Delay:

Prolongs the 2nd gain-set for this time duration (ms) after the falling edge is triggered.

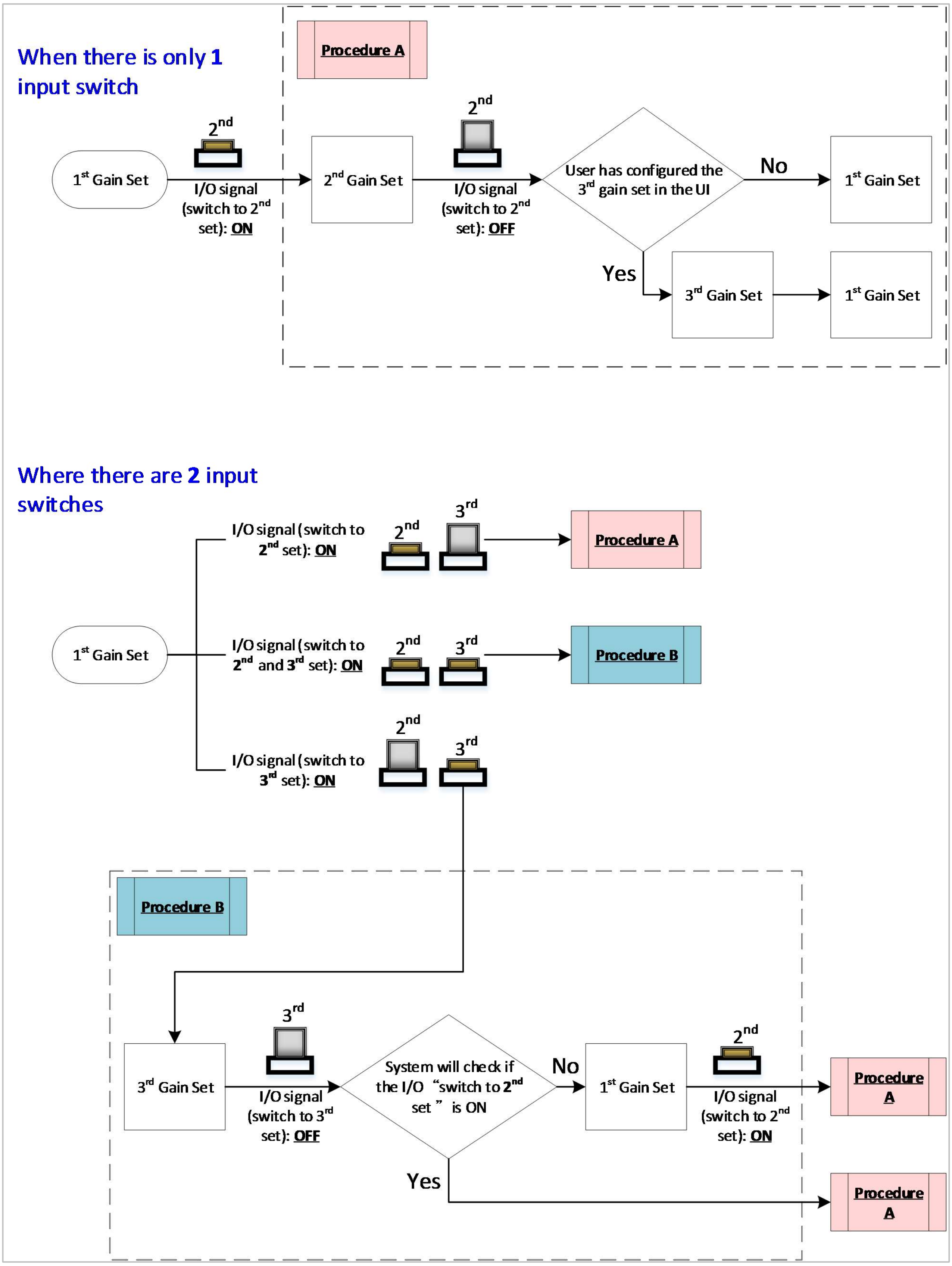
c. Switch 3rd Gain Period:

Performs the 3rd gain-set for this time duration (ms) after the 2nd gain-set ends.

Note:

When this value is set to 0, the 3rd gain is deactivated.

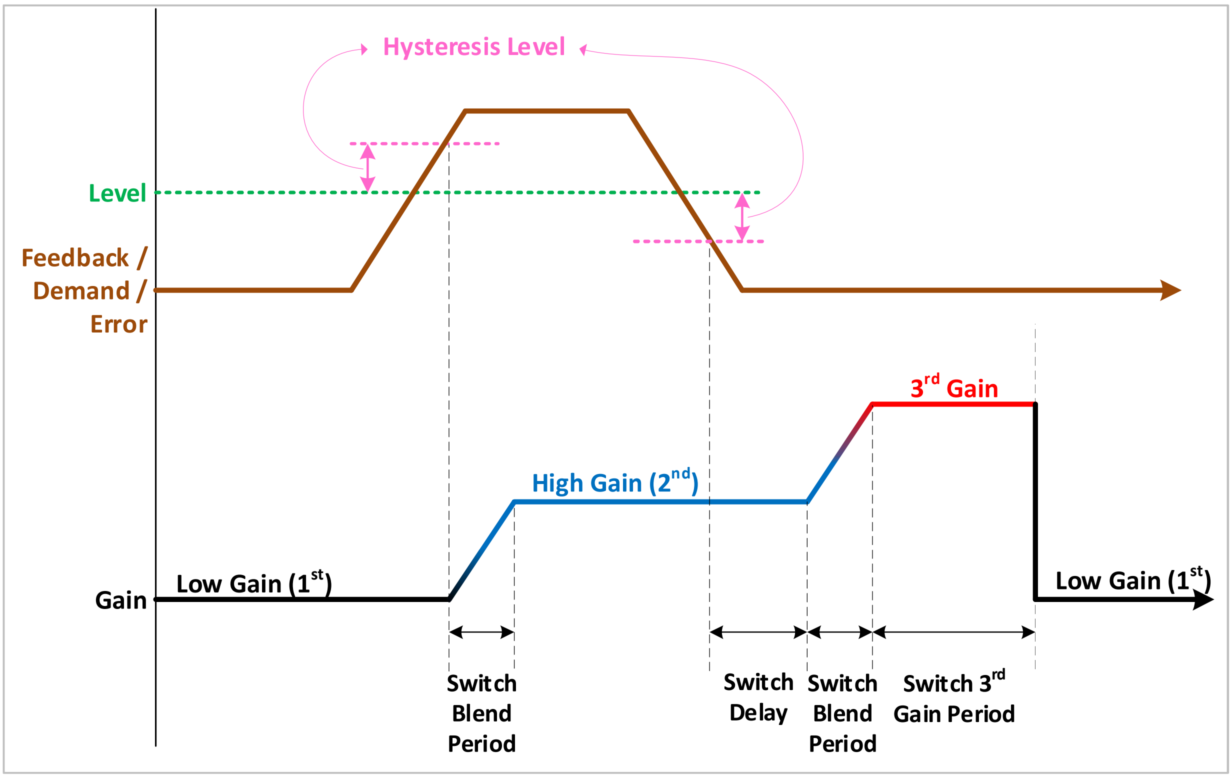
< Gain-set Switching Flow Chart >



5.5.3 Demand & Feedback & Error

These three gain switch modes—Demand, Feedabck, and Error—are to shift gain-sets **when user-configured conditions are reached**. See example on next page.

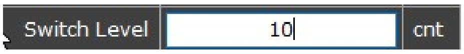
<Gain-set Switching Diagram>



Configuration example:

This setting example 

means:

Use the position demand value to determine which gain-set to apply. When the position demand value is greater than user-configured "Switch Level" 

, the gain-set will shift as described in the "Gain-set Switching Diagram".

a. Switch Blend Period:

The transition time from the 1st gain-set to the 2nd one, or from the 2nd one to the 3rd.

b. Switch Delay:

Prolongs the 2nd gain-set for this time duration (ms) after the Switch Source drops to lower than the hysteresis level (pink line in graph) below the Switch Level.

c. Switch 3rd Gain Period:

Performs the 3rd gain-set for this time duration (ms) after the 2nd gain-set ends. **When this value is set at 0, the 3rd gain is deactivated.**

d. Switch Level:

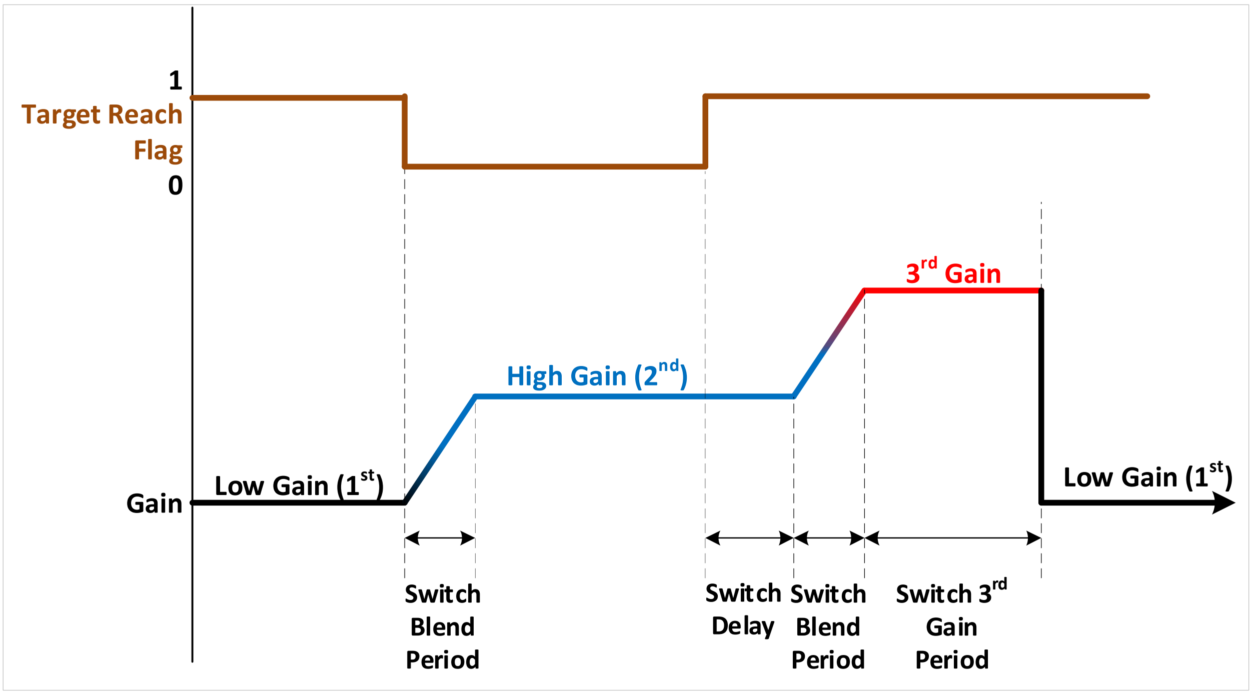
The threshold of switching gain-sets (green line in graph).

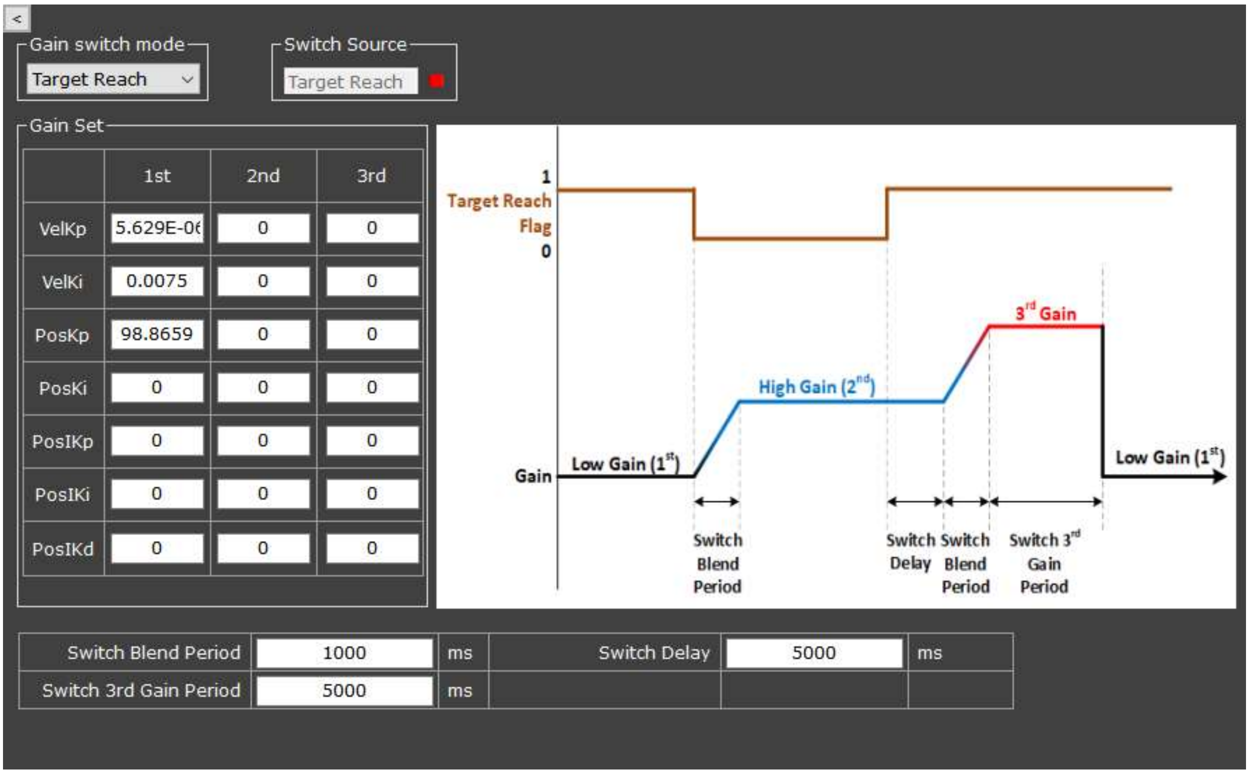
e. Switch Hysteresis Level:

A buffer between the reached Switch Level and the start of switching gain-sets.

5.5.4 Target reach flag

When the Gain switch mode is set to Target Reach, the target reach flag will be used as a gain switch trigger as shown in the diagram below. (For explanation of target reach flag, see **chapter 4.3.3**.)





a. Switch Blend Period:

The transition time (ms) from the 1st gain-set to the 2nd, or from the 2nd to the 3rd.

b. Switch Delay:

Prolongs the 2nd gain-set for this time duration (ms) after the falling edge is triggered.

c. Switch 3rd Gain Period:

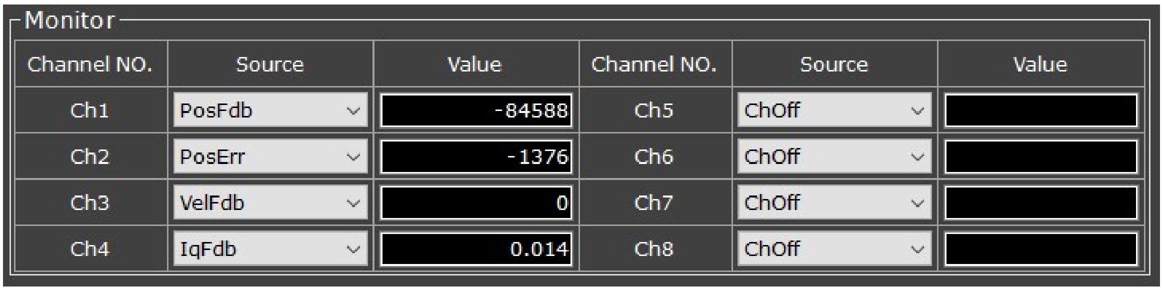
Performs the 3rd gain-set for this time duration (ms) after the 2nd gain-set ends.

When this value is set at 0, the 3rd gain is deactivated.

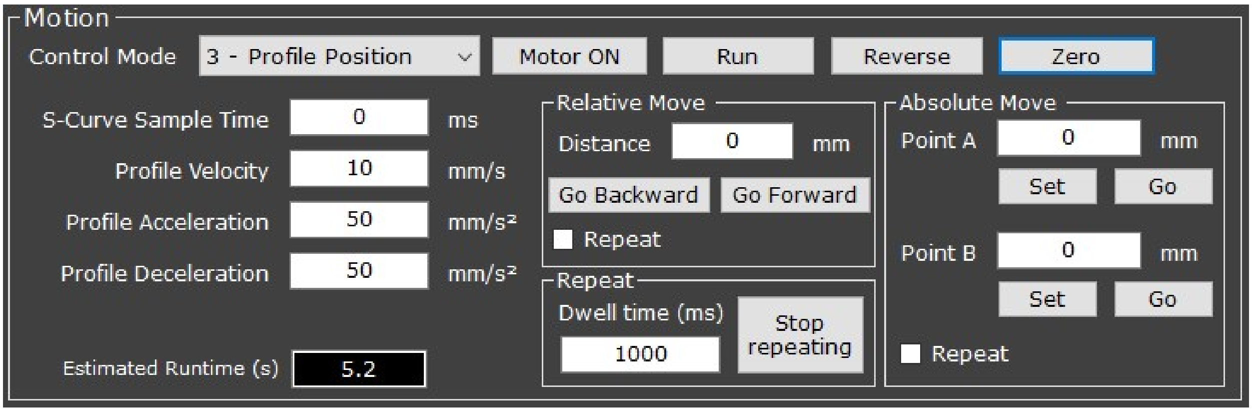
Chapter 6 Trial Run

There are 2 panels in Trail Run: **Monitor** and **Motion**.

Trial Run provides 8 channels for users to **monitor** the **numeric** data from the driver.



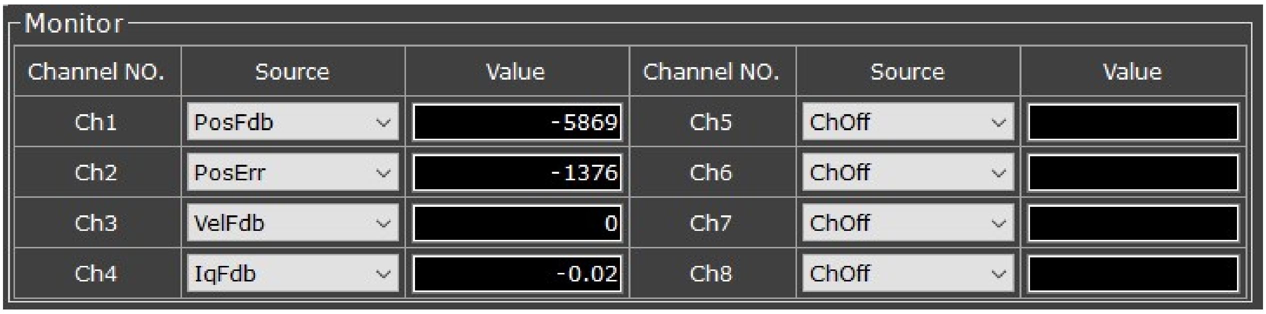
Besides, you can use the **Motion** panel to **test movements**.



<u>Application examples</u>:
Users can observe the encoder feedback value through the monitor panel and evaluate if the encoder is assembled correctly. Or, after tuning, users can test motor's performance (via the motion panel) using the profile velocity/position/torque mode.

There are several motion functions, such as motor operating mode, S-curve motion, and absolute/relative position movements.

6.1 Monitor



#	Option	Function
1	ChOff	Channel off
2	WvGnTheta	Reserved
3	WvGnSwpTmr	Reserved
4	WvGnFreq	Reserved
5	WvGnAmp	Reserved
6	WvGnOut	Reserved
7	VDCFdb	Reserved
8	IaFdb & IbFdb	One of the three phase current from the motor Ia, Ib, Ic formed 360°
9	Vin0 & Vin1	The voltage of analog input (See TC1-B Installation Guide – Chapter 3.7). **Vin0** Analog AI-0-; it is used for analog command. **Vin1** Analog ai-0+; reserved.
10	RegenSat	Reserved
11	IqRefSoft	Target current command, configured via UI
12	IqRefAux	Iq command from the controller
13	ThetaElec	The phase angle on the magnet
14	DdRef	Reserved
15	DqRef	Reserved
16	VdDmd	Reserved
17	VqDmd	Reserved

#	Option	Function
18	Ialpha & Ibeta	Map from the Ia, Ib, Ic model
		Ialpha=Ia , Ibeta= (Ib-Ic)/√3
19	IdFdb	Id feedback from motor
20	IdRef	Id command to the motor
21	IqFdb	Iq feedback from motor
22	IqRef	Torque command to the motor
23	VelUi	Reserved
24	VelPIOut	Velocity PI Controller output
25	VelIqOut	Velocity Iq Output
26	VelFFOut	Velocity feedforward Output
27	PosUi	Reserved
28	PosIUi	Reserved
29	PosRefSoft	Target position command, configured via UI
30	PosRefAux	Position command from the controller
31	PosRefPrfl	Position command from profile path
32	VelRefSoft	Target velocity command, configured via UI
33	VelRefAux	The command from the controller
34	VelRefPrfl	Velocity command from the profile path
35	VelDmd	Velocity demand (before limited)
36	VelRefStpMgr	Velocity reference step manager (after limited)
37	PosDmd	Position demand (before limited)
38	PosRefStpMgr	Position reference step manager (after limited)
39	PosFdb	Position feedback from motor
40	PosErr	The difference form PosRef and PosFdb
41	PosVelOut	Position PI controller output
42	VelRef	Velocity reference
43	VelFdb	Velocity feedback from motor
44	VelErr	The difference form VelRef and VelFdb
45	AccDmd	Acceleration demand
46	AccFdb	Acceleration feedback
47	AccErr	Acceleration error

#	Option	Function
48	Digital Input & Output	32-bit I/O code
49	MainISRLoad	Reserved
50	MainISRLoadP	Reserved
51	VDCRaw	Reserved
52	Ia & Ib Raw	The value of IaFdb & IbFdb analog signal
53	Vin0 & Vin1 Raw	The ADC (Analog-to-Digital converter) value of Vin0 & Vin1 analog signal
54	CtrlLoopCnt	Reserved
55	CtrlLoopLevel	control level in position, velocity or current
56	MachineState	Reserved
57	ErrorCode	Chapter 10 Error code
58	HallCode	Hall sensor signal
59	GainIndex	0: 1st gain-set
		1: 2nd gain-set
		2: 3rd gain-set
60	PosTgtFIFOCn	Reserved
61	STW	Reserved
62	ErrMapSts	0: off
		1: No table
		2: Waiting for Homing
		3: Active
63	EncSinFdb	Encoder sine feedback
64	EncCosFdb	Encoder cosine feedback
65	5VAnalog	Over-temperature protection 5V analog input
66	EncSinRaw	The ADC value of encoder sine analog signal
67	EncCosRaw	The ADC value of encoder cosine analog signal
68	MainsFreq	AC input frequency in 0.1 Hz.

6.2 Motion

a. Control Mode:

Choose a preferred control mode here.

b. Motor ON:

Click to enable motor-on.

c. Run:

Click to execute motion commands.

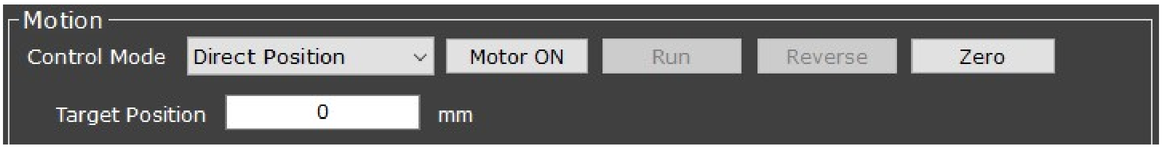
d. Reverse:

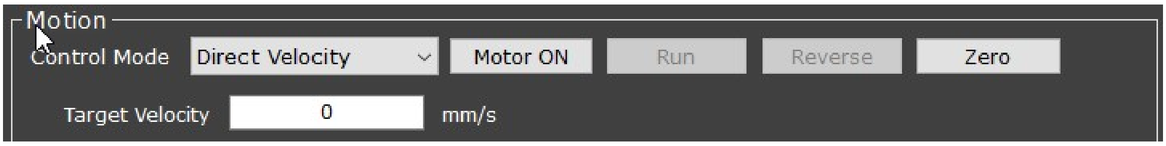
Reverse the value of target. (This function is only applicable when motor is on.)

e. Zero:

Make the motor's present position as position 0. (This function can only be enabled when motor is off.)

6.2.1 Direct Position & Velocity & Current



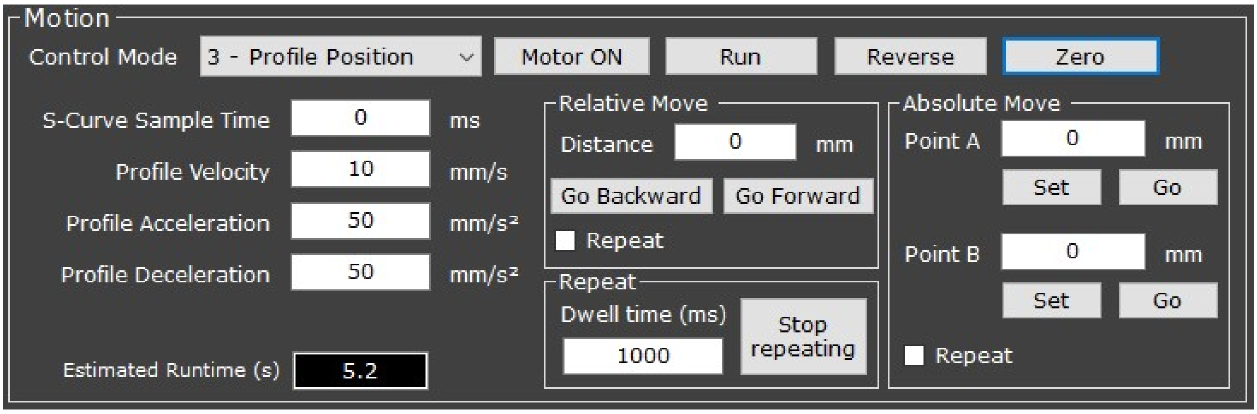


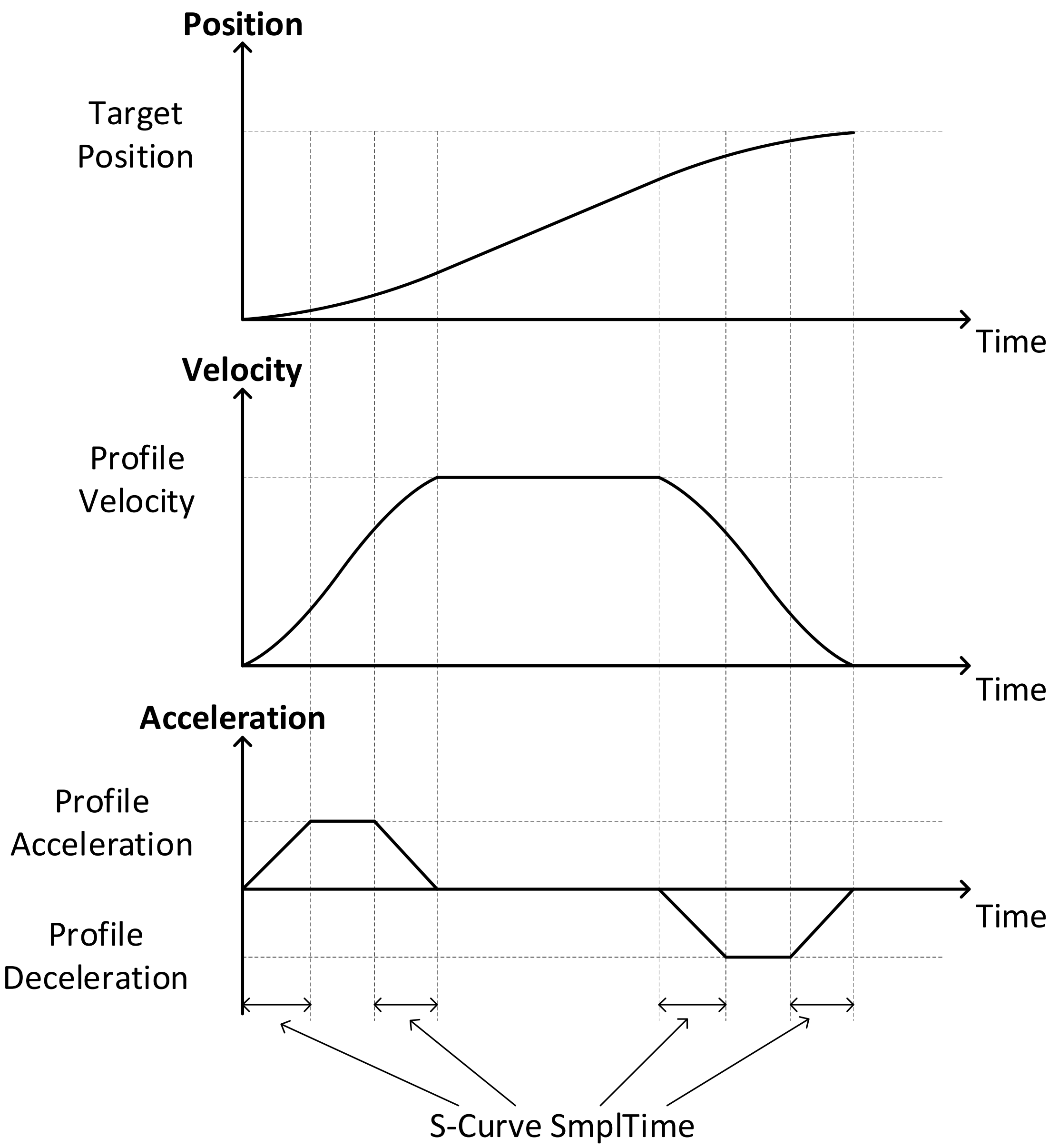
Set the target position, velocity, or current, the driver will reach the goal as soon as possible.

a. IdRefSoft:

for cpc internal use only.

6.2.2 Profile Position





a. **S-Curve SmplTime:**
   To smooth the acceleration slope to avoid too much vibration.

b. **Profile Velocity**:
   The maximum speed during movement.

c. **Profile Acceleration & Deceleration**:
   The maximum acceleration & deceleration during movements.

d. **Estimated Runtime (s)**:
   The estimated time (in second) needed to reach target.

[Relative Move Panel]

e. **Distance:**
   Sets the moving distance.

f. **Go Forward/Backward:**
   Sets the moving direction.

g. **Repeat:**
   Tick to activate the "repeat" function.

[Absolute Move Panel]

h. **Point A/B:**
   Manually key in the target position(s) here.

i. **Set:**
   Automatically fills in motor's current position into the columns of Point A/B. (Namely, this function sets motor's current position as point A/B).

j. **Go:**
   Click to make the motor move to point A/B.

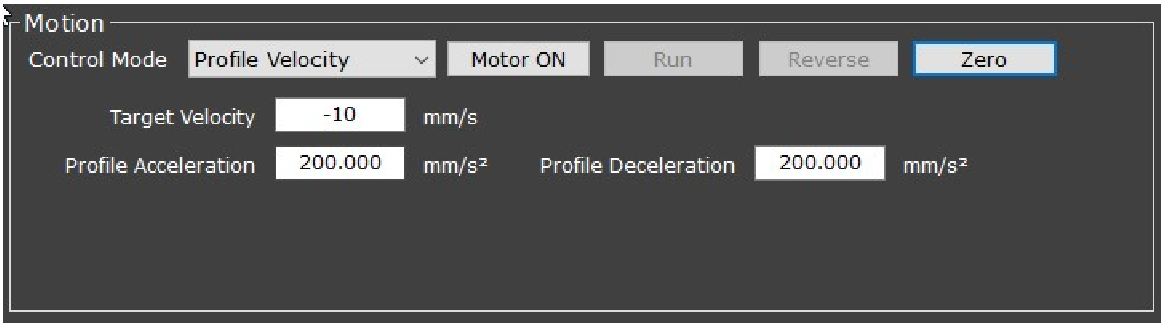
k. **Repeat:**
   Tick to activate the "repeat" function.

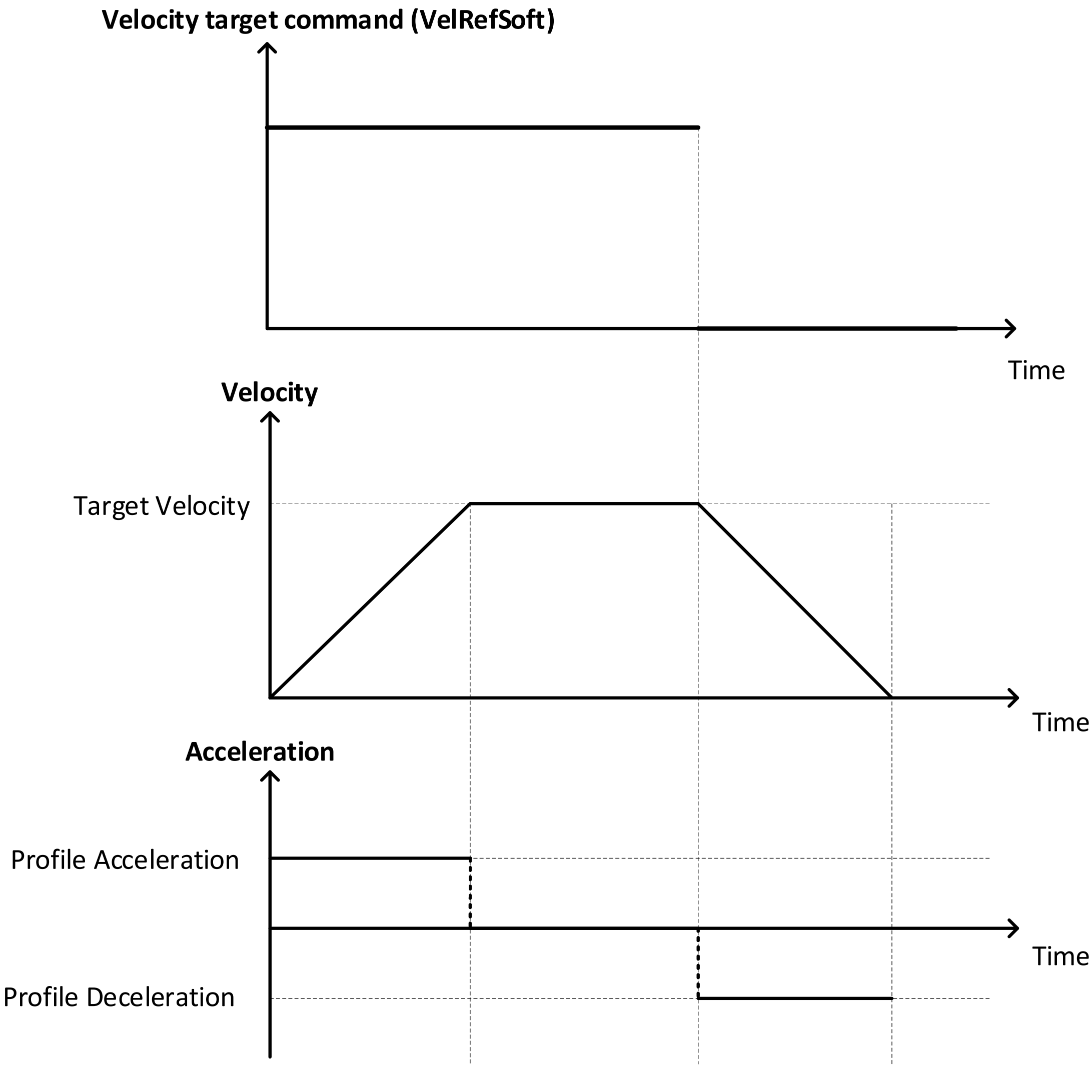
[Repeat Panel]

l. **Dwell time (ms):**
   Waits for this amount of time after the motor reaches target (i.e. point A/B or relative distance).

m. **Stop repeating:**
   Cancels the command of repeating.
   Note: Once you click this button, the motor won't stop until it reaches the designated target.

6.2.3 Profile Velocity





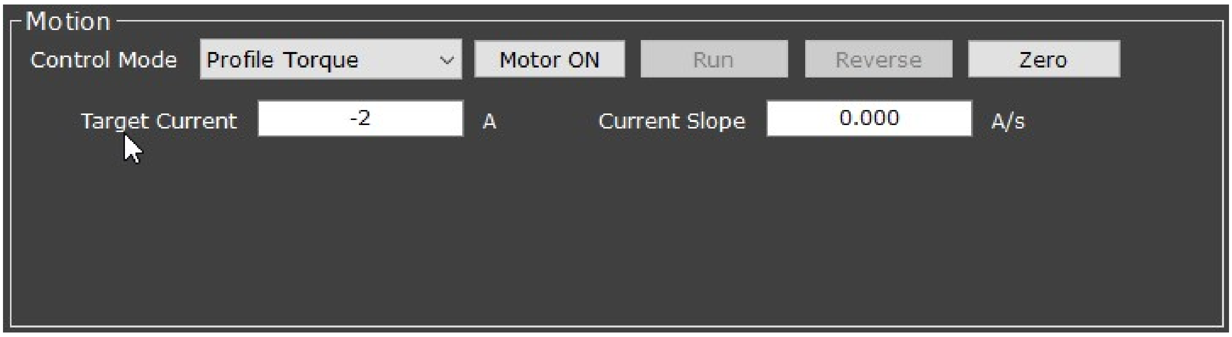
a. Target Velocity:

Key in the velocity command value here.

b. Profile Acceleration & Deceleration:

The maximum acceleration & deceleration during movements.

6.2.4 Profile Torque



a. **Current Slope:**

The rate of current increase, in amperes/sec.

Chapter 7 Homing

The operation of incremental encoder is based on calculating the increments (counts) between moves. However, for the operation of positioning the driver, an exact knowledge of the absolute position is normally required; we have to **define a start point (Home) before working**—which is called Homing—in order to know at what exact position the moves will be instead of knowing merely the relative moving distance.

There are several homing methods. Each method establishes:

- **Homing signal** (limit or home switch transition or encoder index pulse):
  (1) Backward limit switch
  (also called "negative limit switch" on the following pages)
  (2) Forward limit switch
  (also called "positive limit switch" on the following pages)
  (3) Home switch
  (4) Index pulse from an encoder.

- **Direction of motion** and, where appropriate, the relationship of the index pulse to limit or home switches.

7.1 Setting

There are several homing methods ranging from number **#-12 to #35**; the parameters to be set are the same, they are: speed, home offset, acceleration, hard stop current, and hard stop period.

a. **Home Method / Start:**

Selects a homing method from the list and click Start.

b. **Status:**

Shows the present status of the homing procedure.

c. **Home Speed (Switch/Index):**

The speed of moving to the switch / index.

d. **Home Offset:**

The offset counts from origin. Moreover, the offset value will be the position count(s) when homing is completed.

e. Home Acceleration:

The acceleration of homing.

f. Hard Stop Current / Period:

(1) When the driver continues to output **more** than or equal to a certain percentage (%) of peak current for a period, the motor will be regarded as hitting a hard stop.

(2) **Note:** The method of using mechanical hard stop as references can be applied only under homing method -1, -2, -3, -4, and -5. If the hard stop conditions are triggered under other homing methods, homing will fail.

g. Transition to Profile Position Mode, on successful Homing operation:

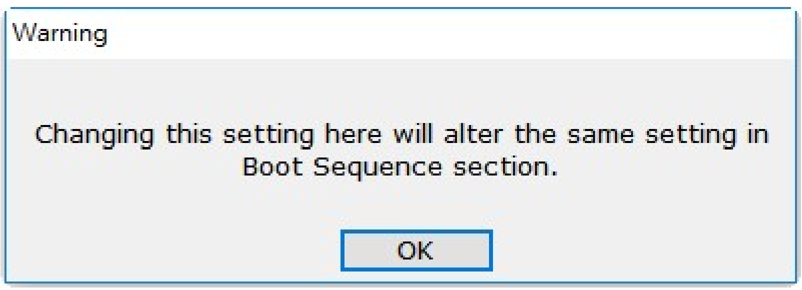
When ticked, once Homing procedure is completed, the system will automatically transit to Profile Position Mode.

Note:

- There is a same setting in Boot Sequence (see **chapter 4.8**) which **links** with the setting (e) here. **Changing the setting (e) in Homing section will alter the same setting in Boot Sequence. Vice versa.**
- On condition that in the Boot Sequence section the “Homing” step* is set to be performed, a warning** will show up when users change the setting (e) in Homing section.

*:

**:



h. Move to new zero position, on successful Homing operation:

When ticked, once Homing procedure is completed and system has switched to Profile Position mode, move to the newly-defined zero position. **This function is available only when the "*Transition to Profile Position Mode, on successful Homing operation*" option is ticked (activated).**

i. 

:

- Ticked: Triggers a fault event (error code "0x8613") when there is a homing error.
- Unticked: No fault event will be triggered when there is a homing error.

j. Switch status:

Displays the status of Forward/Backward/Home switches.
(Green: on; Red: off; Grey: undefined).

k. Set I/O:

A quick link to the Input panel.

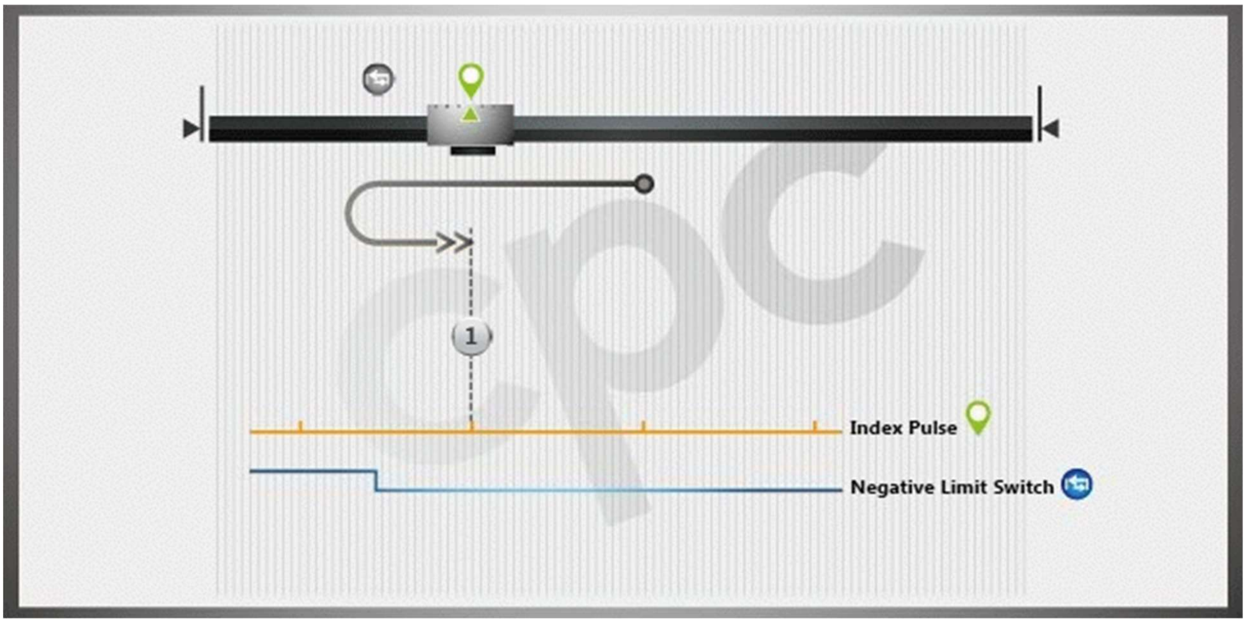
7.2 Homing Method

7.2.1 CiA 402 Standard Homing Method

By Limit Switch and Index Pulse

Method 1:

Home on the first **index** pulse after departing from the **negative** limit switch.

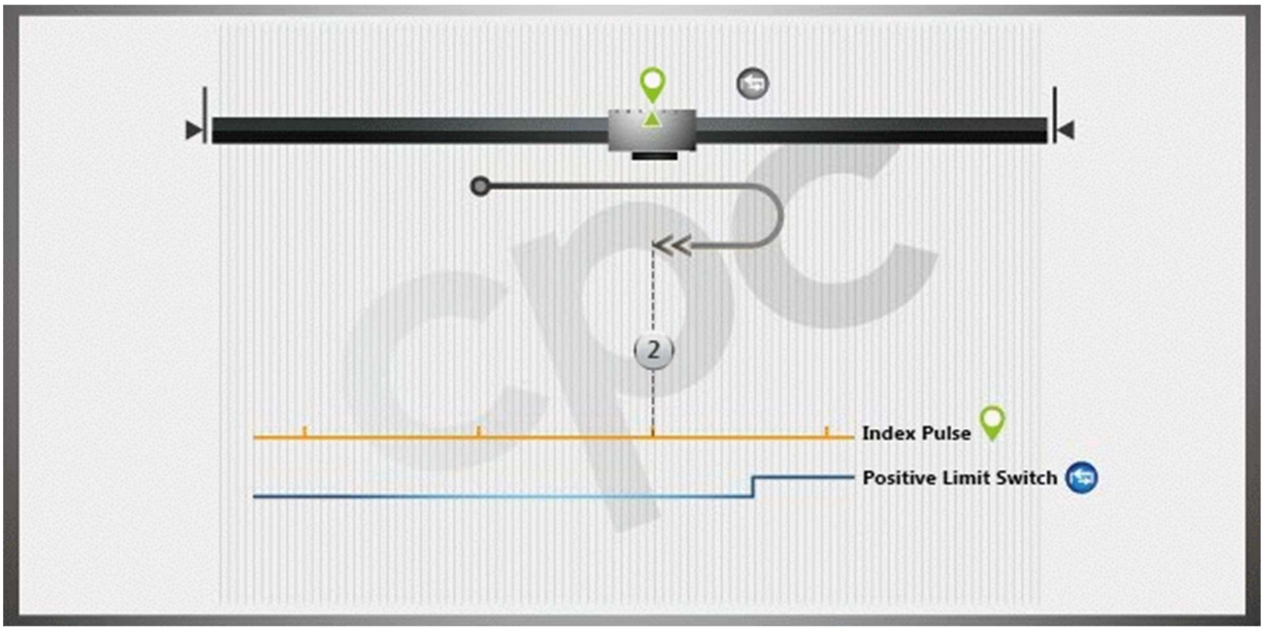


Homing process:

- Start with the negative motion unconditionally to the rising edge of the negative limit switch; then move in positive direction until the first index pulse is found.

Method 2:

Home on the first **index** pulse after departing from the **positive** limit switch.

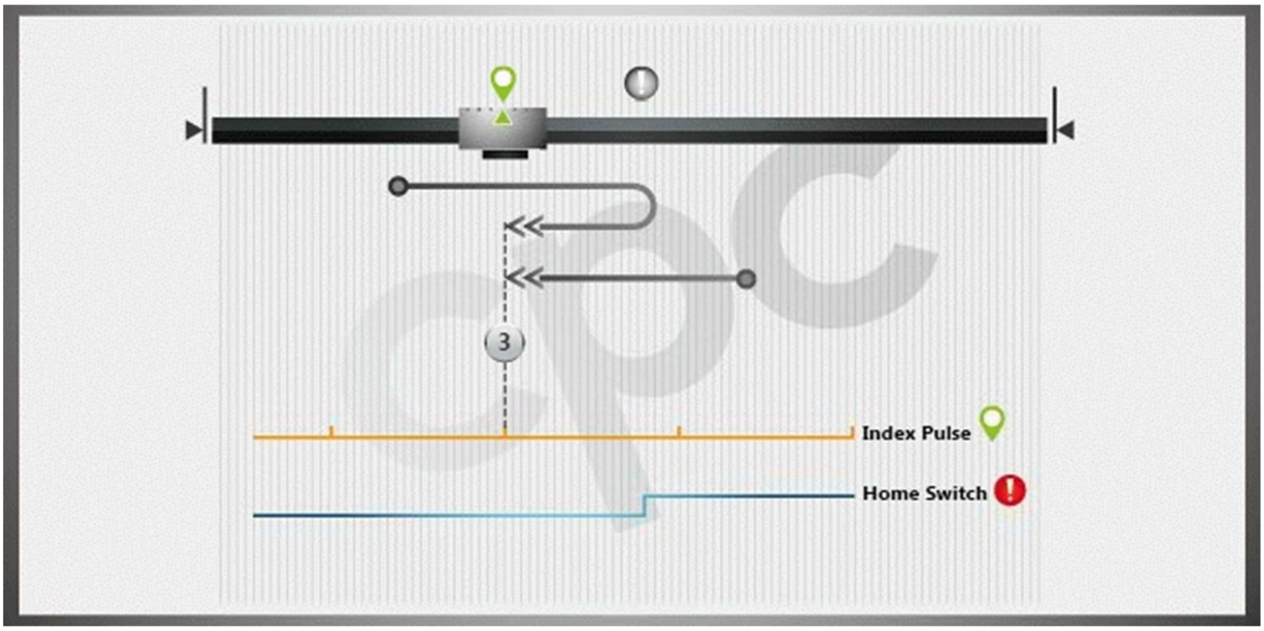


Homing process:

- Start with the positive direction unconditionally to the rising edge of the positive limit switch; then move in negative direction until the first index pulse is found.

Method 3:

Home on first **index** pulse after departing from home switch.



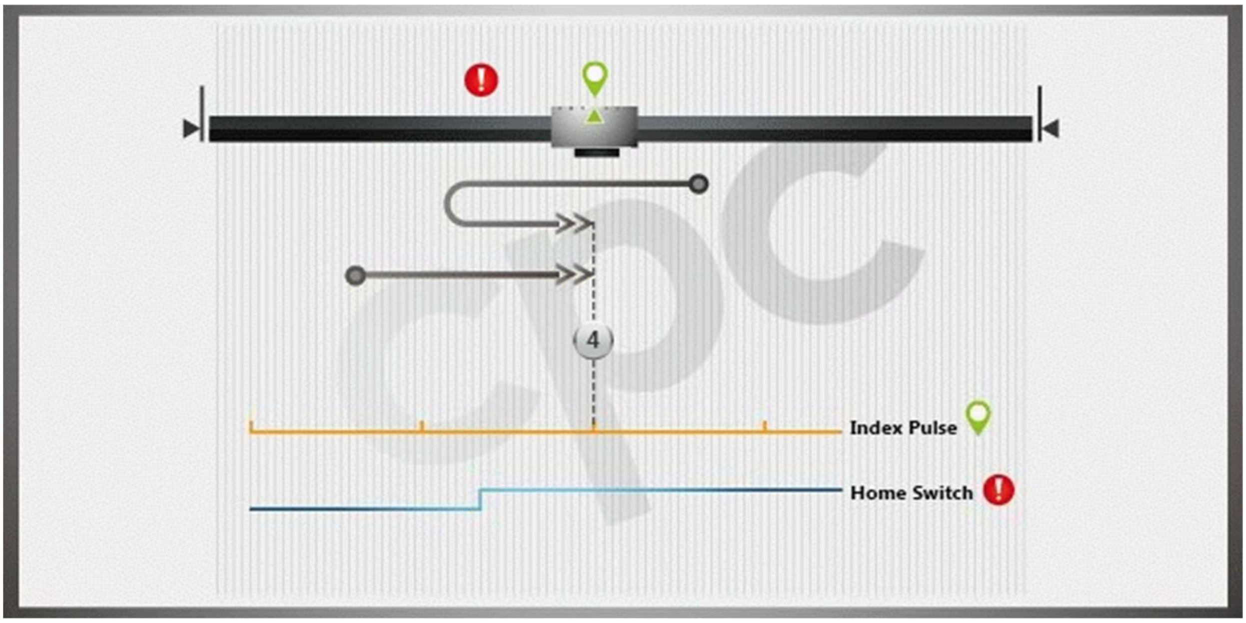
Homing process:

- If the home switch is inactive, start with the positive direction to the rising edge of the home switch; then move in negative direction until the first index pulse is found.
- If the home switch is active, start with the negative direction until the first index pulse is found.

By Home Switch and Index Pulse

Method 4:

Home on the first index pulse after engaging home switch.

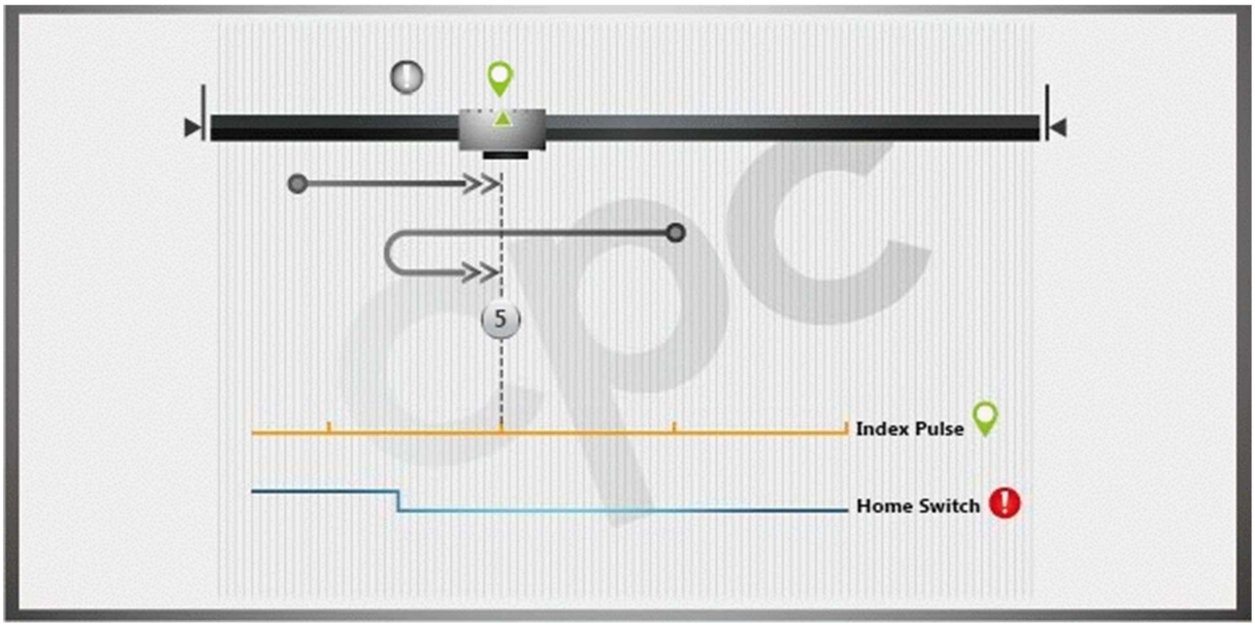


Homing process:

- If the home switch is active, start with the negative direction to the falling edge of the home switch; then move in positive direction until the first index pulse is found.
- If the home switch is inactive, start with the positive direction until the home switch is engaged, then keep moving in positive direction until the first index pulse is found.

Method 5:

Home on the first index pulse after departing from home switch.

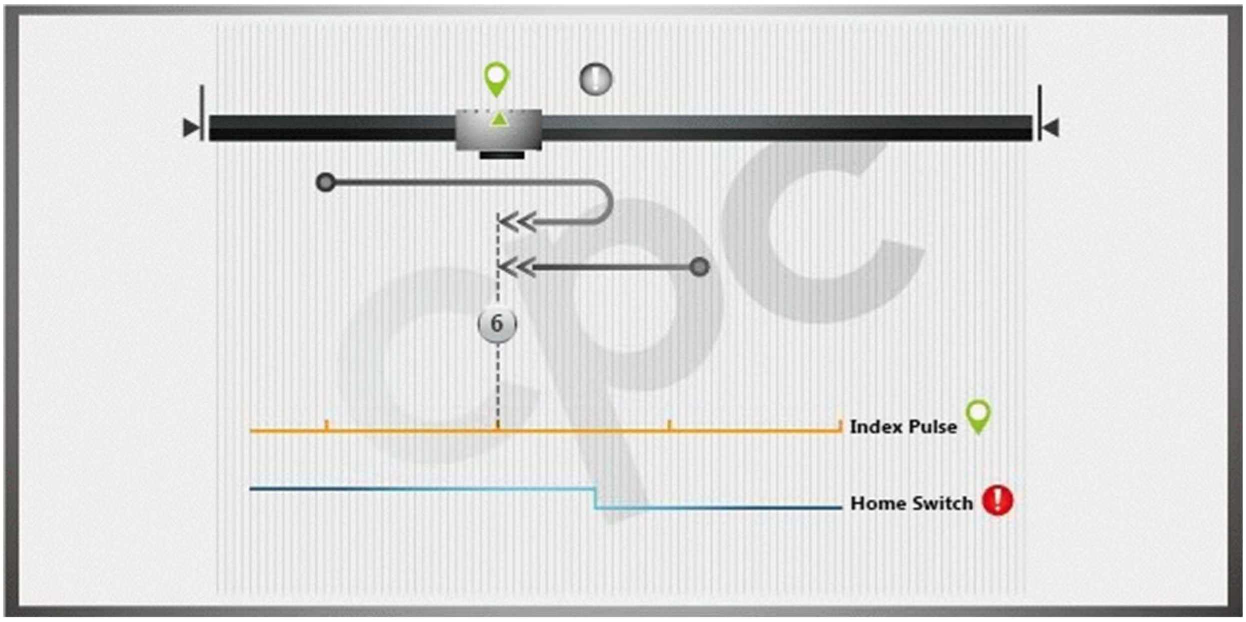


Homing process:

- If the home switch is active, start with the positive direction to the falling edge of the home switch; then keep moving in positive direction and home on the next index pulse found.
- If the home switch is inactive, start with the negative direction to the rising edge of the home switch; then move in positive direction until the first index pulse is found.

Method 6:

Home on the first index pulse after engaging home switch.



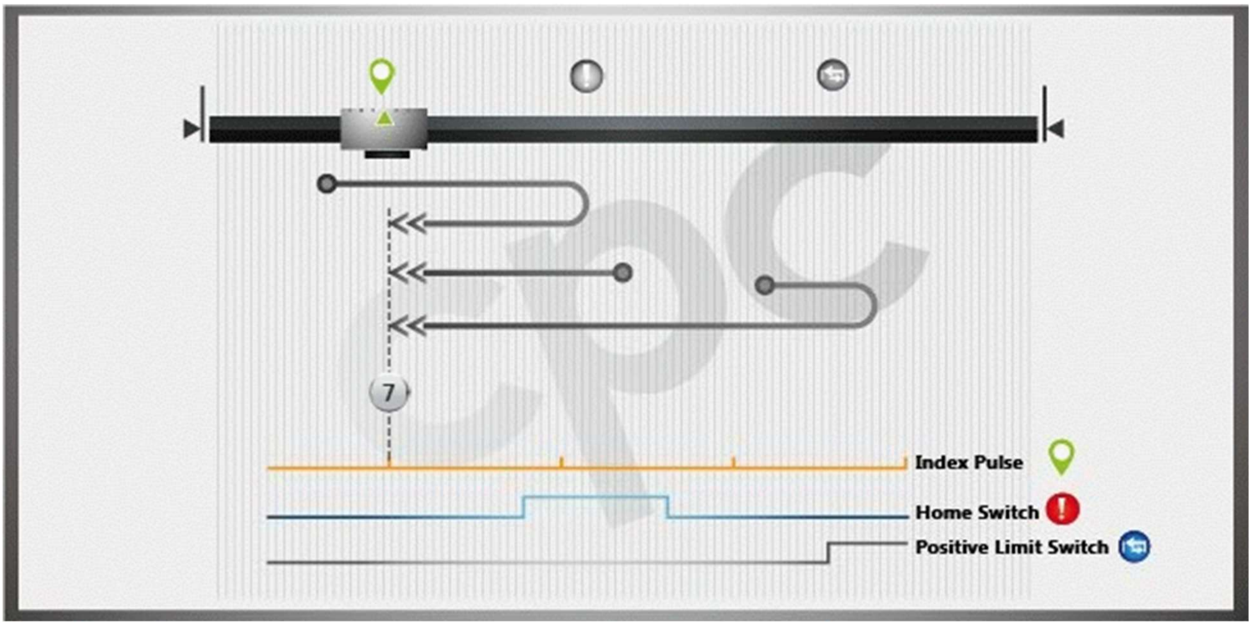
Homing process:

- If the home switch is active, start with the positive direction to the falling edge of the home switch; then move in negative direction until the first index pulse is found.
- If the home switch is inactive, start with the negative direction to the rising edge of the home switch; then keep moving in negative direction until the first index pulse is found.

By Home Switch, Index Pulse, and Limit Switch

Method 7:

Home on the first index pulse after departing from home switch while moving in negative direction.

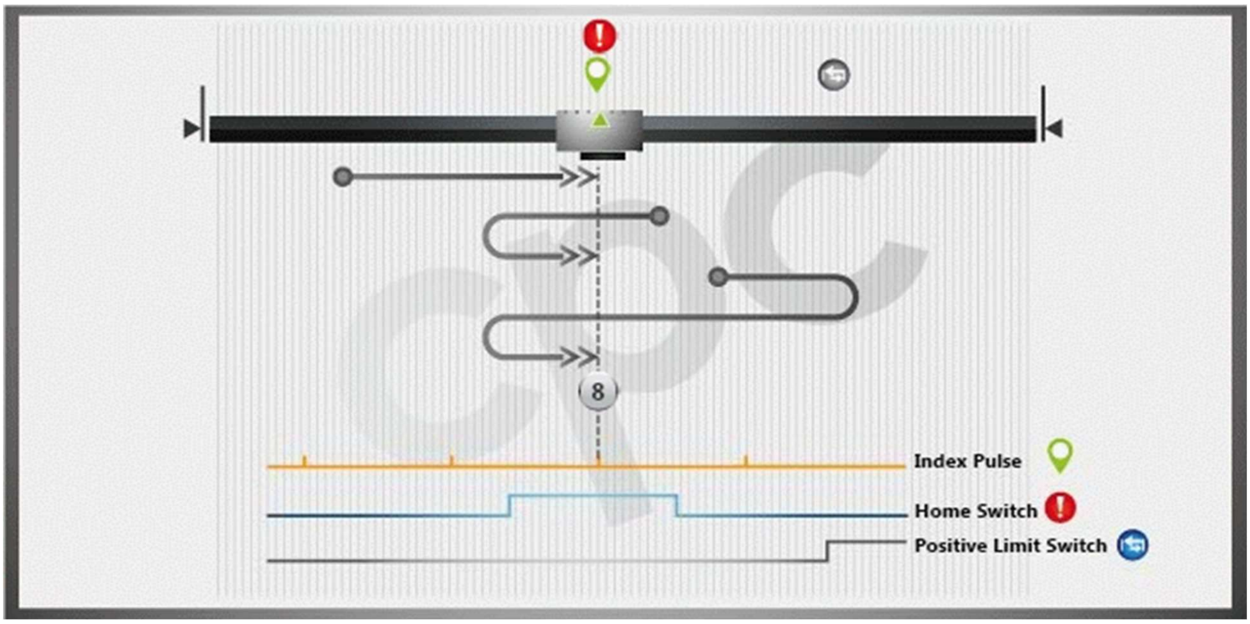


Homing process:

- If the home switch is inactive, start with the positive motion. If the home switch is engaged, move in negative direction until the home switch is disengaged, then find the first index pulse.
- If the home switch is active, start with the negative direction until the home switch is disengaged, then continue moving in negative direction until the first index pulse is found.
- If the home switch is inactive, start with the positive motion. If the positive limit switch is engaged, move in negative direction until the home switch is engaged and then disengaged, then move in negative direction until the first index pulse is found.

Method 8:

Home on the first index pulse after engaging home switch while moving in **positive** direction.

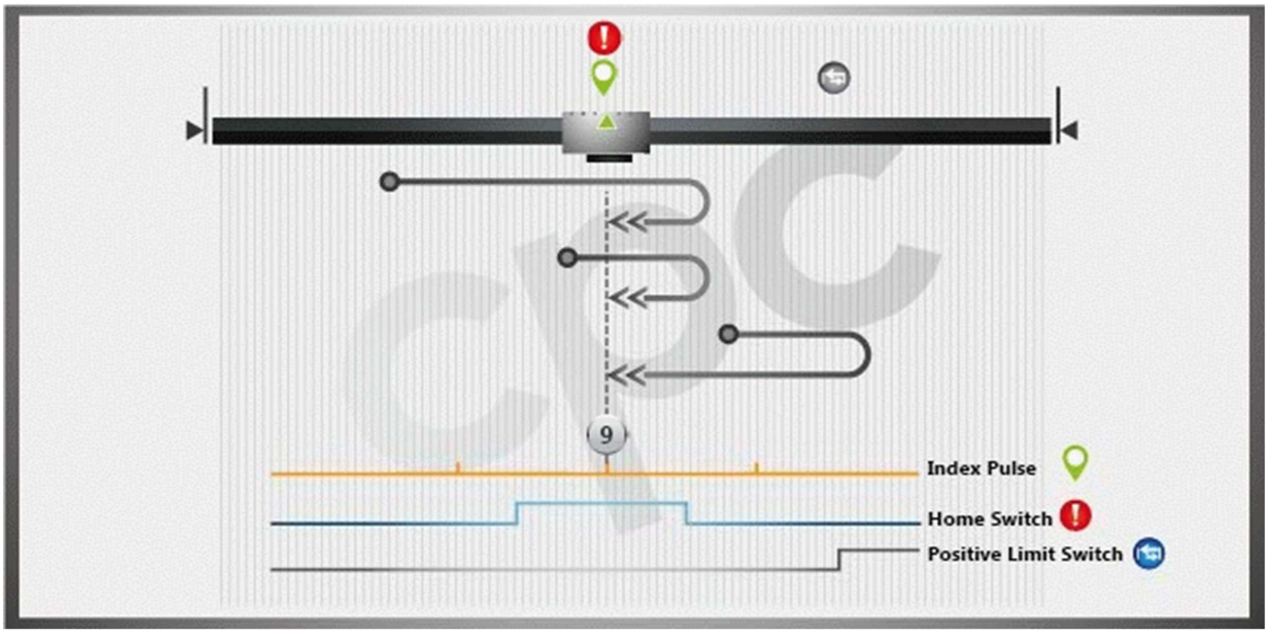


Homing process:

- If the home switch is inactive, start with positive direction until the home switch is met, then keep moving in positive direction until the first index pulse is found.
- If the home switch is active, start with negative direction until home switch is disengaged, then move in positive direction until home switch is engaged, and then find the first index pulse.
- If the home switch is inactive, start with positive direction; when the positive limit switch is engaged, move in negative direction until the home switch is engaged and disengaged, then move in positive direction until the first index pulse is found.

Method 9:

Home on the first **index** pulse after **engaging** home switch while **moving in negative** direction.

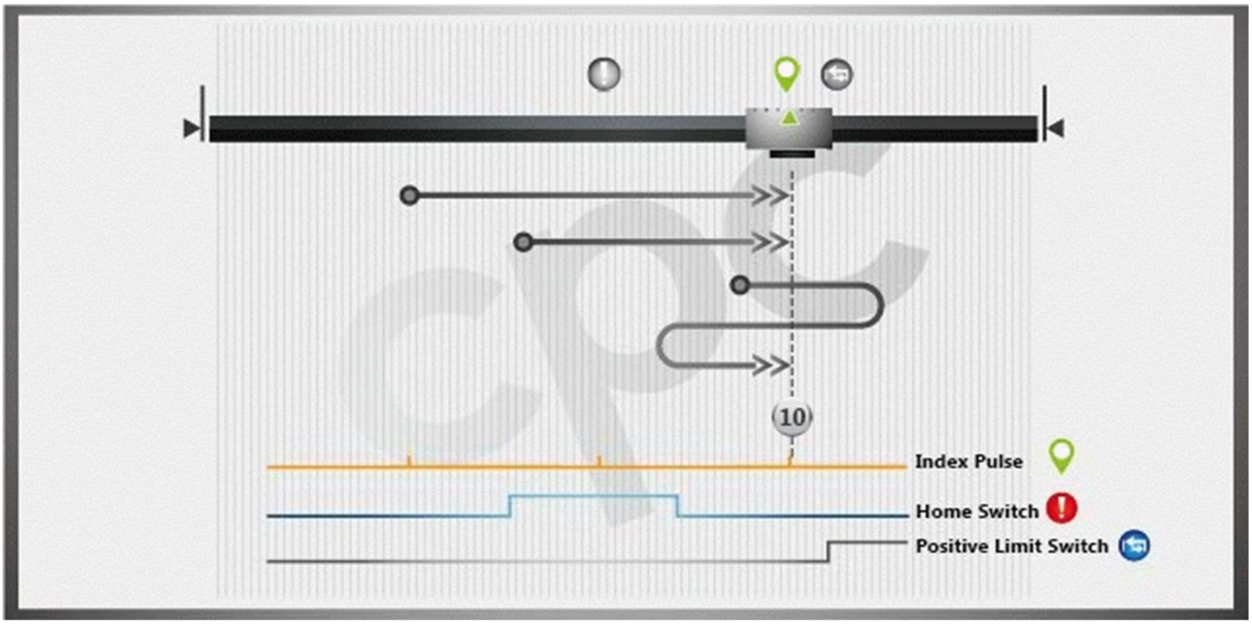


Homing process:

- Start with positive direction unconditionally. If home switch is engaged, keep moving in positive direction until home switch is disengaged, then move in negative direction until home switch is engaged, and then find the first index pulse.
- Start with positive direction unconditionally. If home switch is disengaged, move in negative direction until home switch is engaged, then find the first index.
- Start with the positive motion unconditionally. If the positive limit switch is engaged, move in negative direction until home switch is engaged, then find the first index pulse.

Method 10:

Home on the first **index** pulse after **departing** from home switch **while moving in positive** direction.

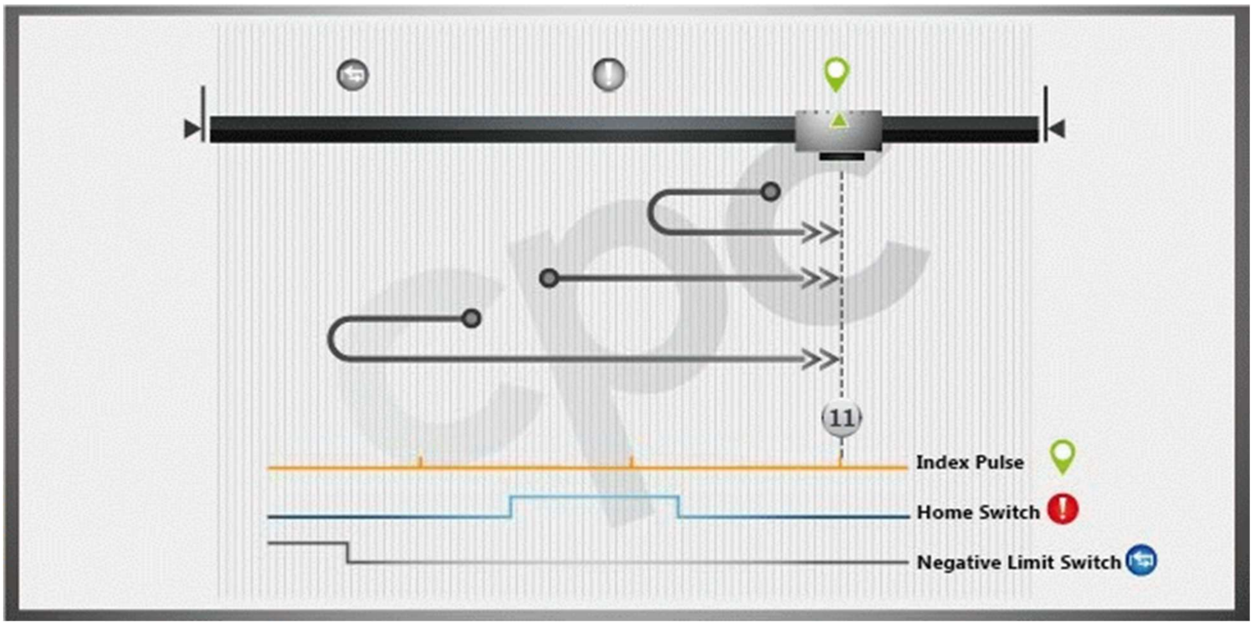


Homing process:

- Start with positive direction unconditionally. If home switch is then engaged, keeping moving in positive direction until home switch is disengaged, then find the first index pulse.
- Start with positive direction unconditionally. If home switch is active and then disengaged, keep moving in positive direction until the first index pulse is found.
- Start with position direction unconditionally. If positive limit switch is then engaged, move in negative direction. If home switch is engaged, move in positive direction until home switch is disengaged, then find the first index pulse.

Method 11:

Home on the first **index** pulse after **departing** from home switch **while moving in positive** direction.

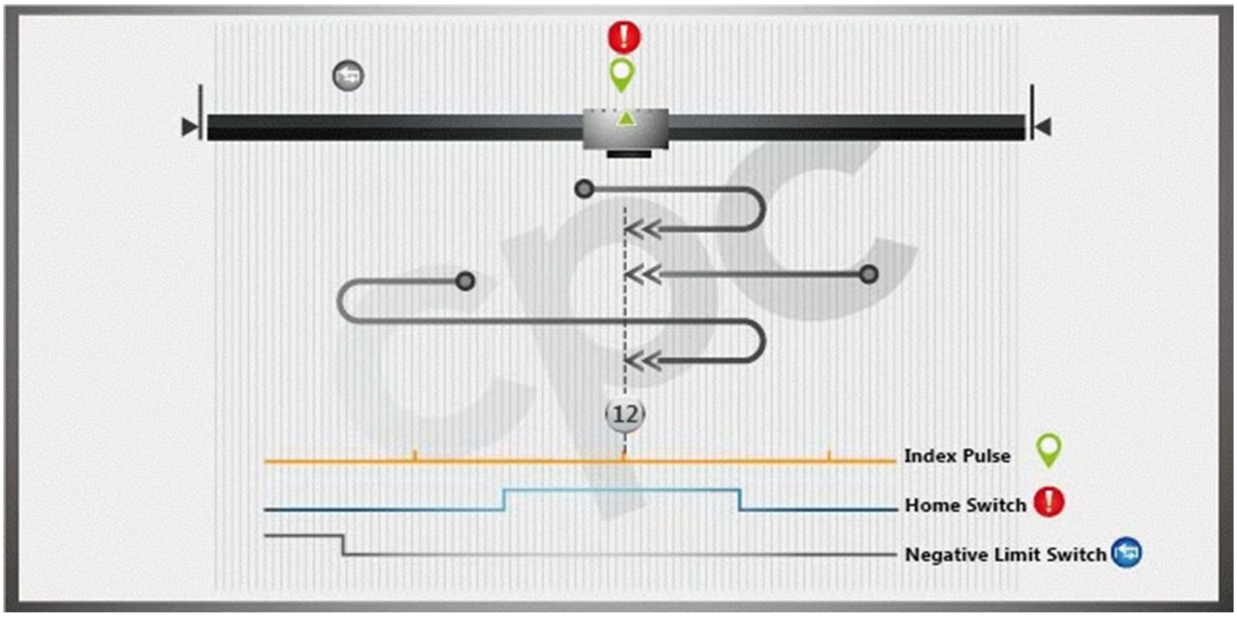


Homing process:

- If home switch is inactive, move in negative direction. If home switch is then engaged, move in positive direction until home switch is disengaged, then find the first index pulse.
- If home switch is active, move in positive direction until home switch is disengaged, continue moving in positive direction until the first index pulse is found.
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction until home switch is engaged, continue moving in positive direction until home switch is disengaged, then find the first index pulse.

Method 12:

Home on the first **index** pulse after **engaging** home switch **while moving in negative** direction.

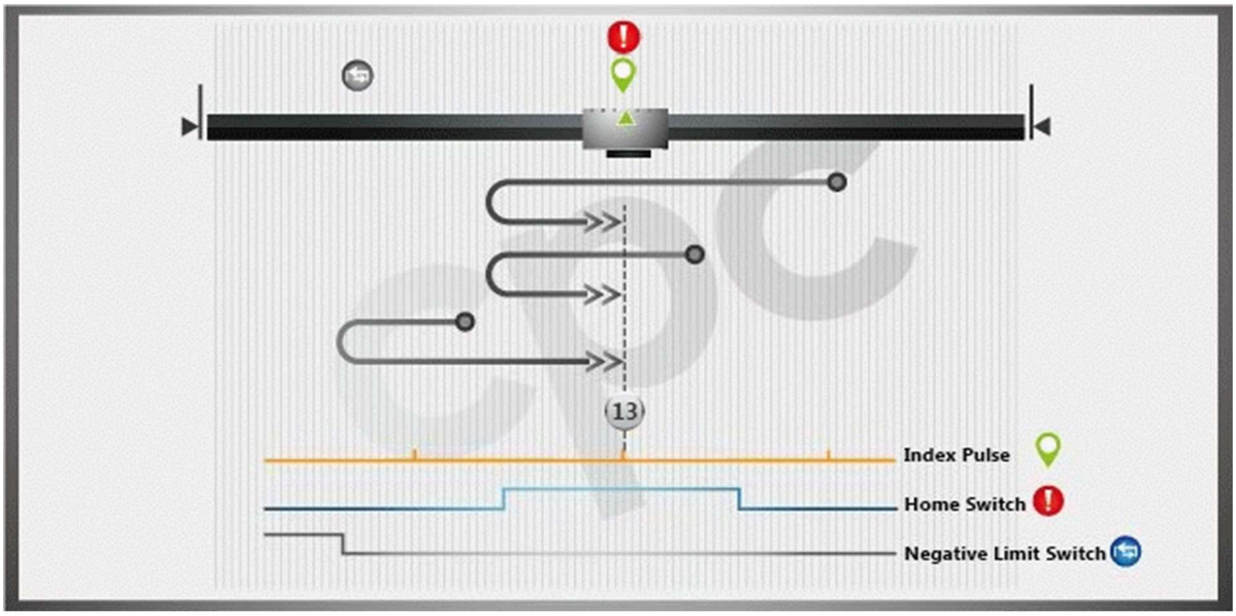


Homing process:

- If home switch is active, move in positive direction. If home switch is then disengaged, move in negative direction until home switch is engaged, then find the first index pulse.
- If home switch is inactive, move in negative direction until home switch is engaged, then, continue moving in negative direction until the first index pulse is found.
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction until home switch is engaged. If home switch is then disengaged, move in negative direction until home switch is engaged, then find the first index pulse.

Method 13:

Home on the first **index** pulse after **engaging** home switch **while moving in positive** direction.

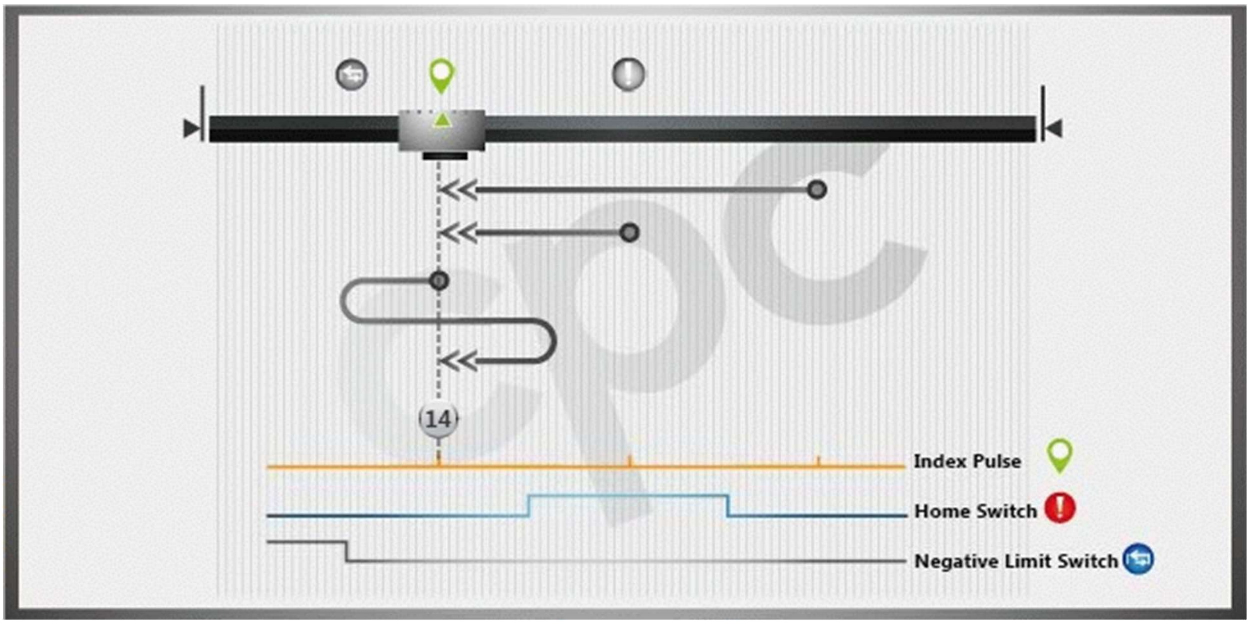


Homing process:

- Start with negative motion unconditionally. If home switch is then engaged, continue moving in negative direction. If home switch is then disengaged, move in positive direction until home switch is engaged, then find the first index pulse.
- Start with negative motion unconditionally. If home switch is then disengaged, move in positive direction until home switch is engaged, then find the first index pulse.
- Start with negative motion unconditionally. If negative limit switch is then engaged, move in positive direction until home switch is engaged, then find the first index pulse.

Method 14:

Home on the first **index** pulse after **departing** home switch **while moving in negative** direction.



Homing process:

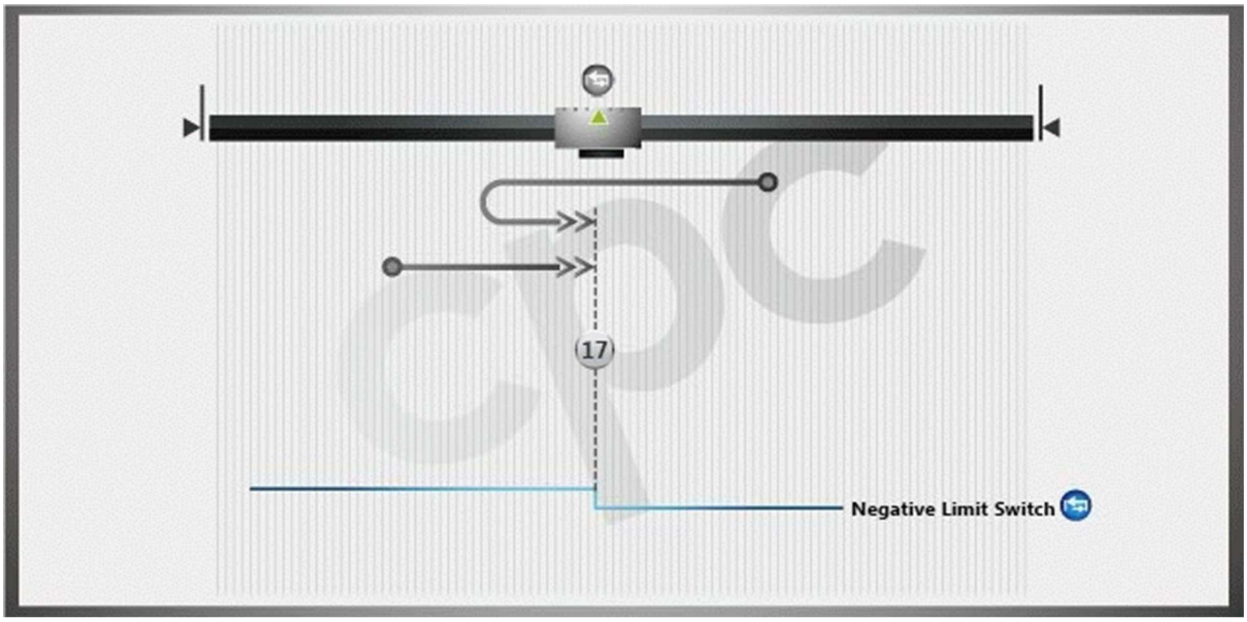
- Start with negative direction unconditionally. If home switch is inactive, move in negative direction. If home switch is then engaged, keep moving in negative direction until home switch is disengaged, then, find the first index pulse.
- Start with negative direction unconditionally. If home switch is then disengaged, keep moving in negative direction until the first index pulse is found.
- Start with negative direction unconditionally. If negative limit switch is then engaged, move in positive direction. If home switch is then engaged, move in negative direction until home switch is disengaged, then find the first index pulse.

Method 15 to 16: Reserved.

By Limit Switch

Method 17:

Home on negative limit switch without index pulse.

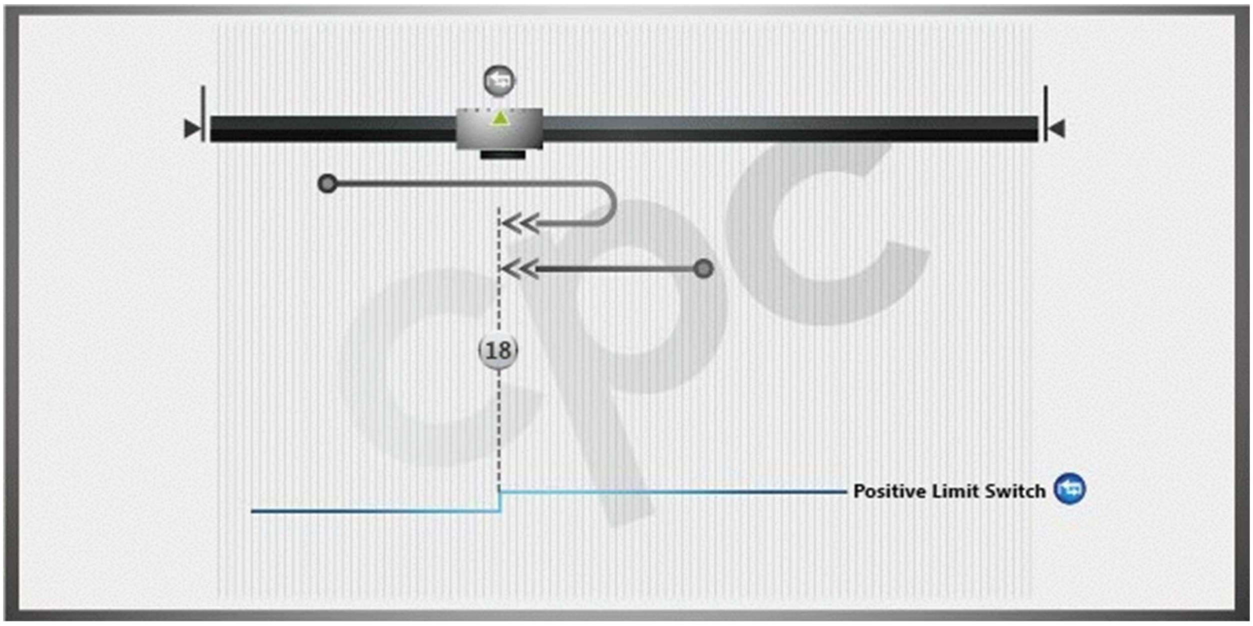


Homing process:

- If negative limit switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction to locate the falling edge of the negative limit switch.
- If negative limit switch is active, move in positive direction to locate the falling edge of the negative limit switch

Method 18:

Home on positive limit switch without index pulse.



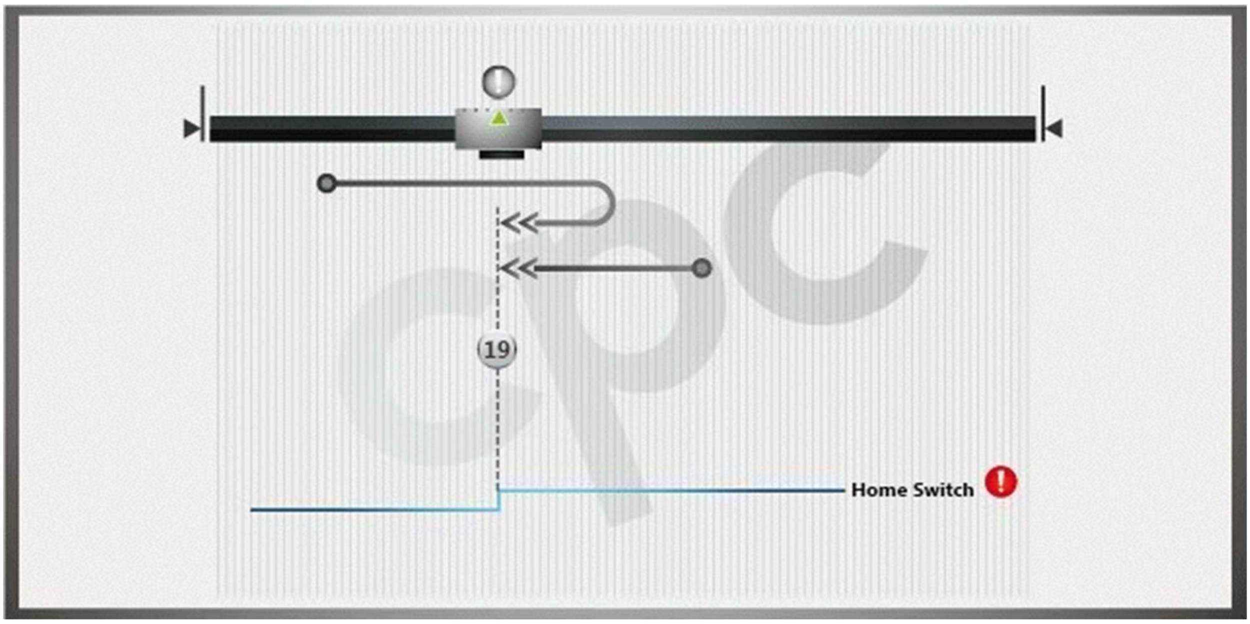
Homing process:

- If positive limit switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction to locate the falling edge of the positive limit switch.
- If positive limit switch is active, move in negative direction to locate the falling edge of the positive limit switch

By rising/falling edge of Home Switch

Method 19:

Home on falling edge of home switch without index pulse.

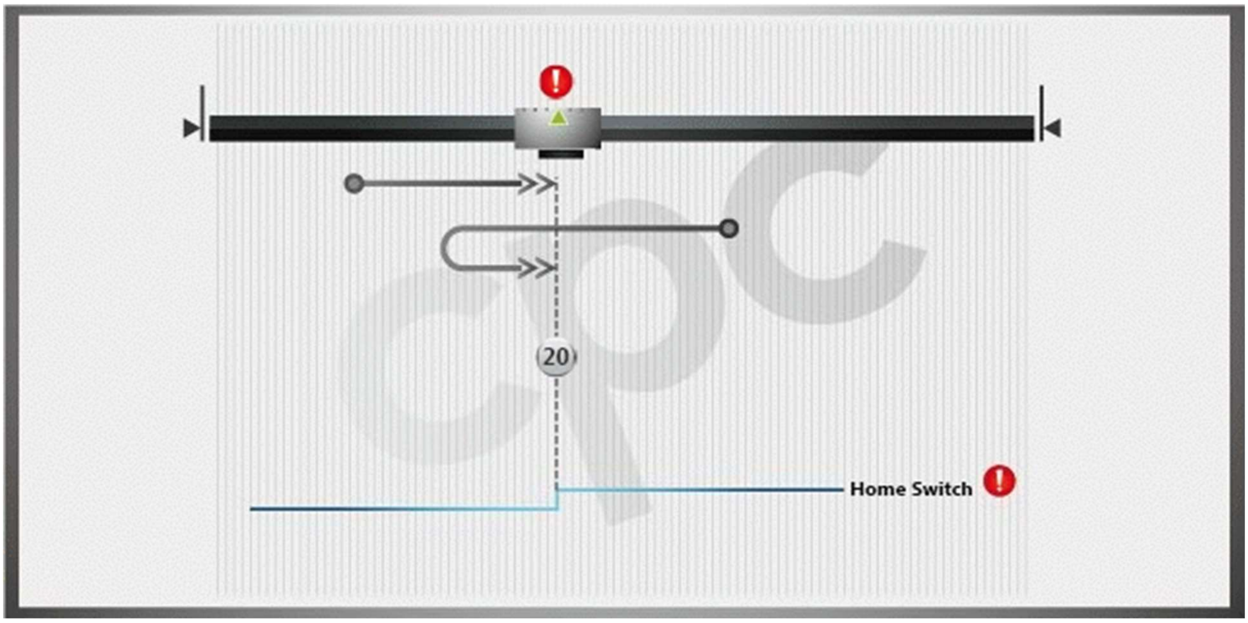


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, move in negative direction to locate the falling edge of the home switch.
- If home switch is active, move in negative direction to locate the falling edge of the home switch.

Method 20:

Home on rising edge of home switch without index pulse.

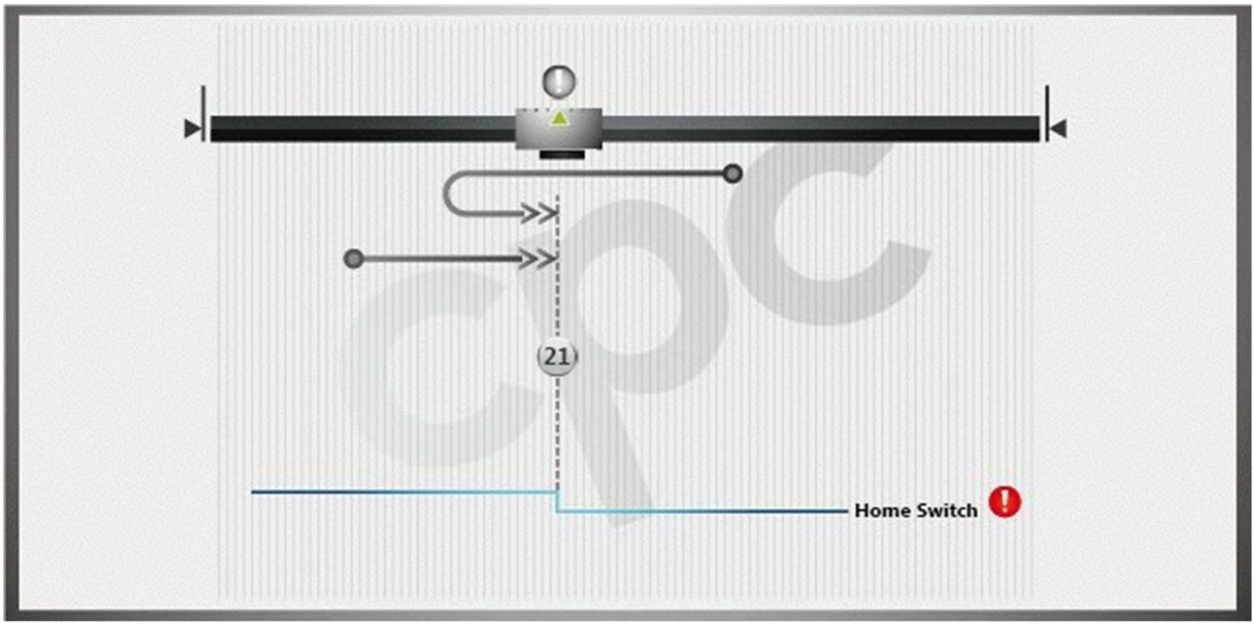


Homing process:

- If home switch is inactive, move to rising edge of the home switch in positive direction.
- If home switch is active, move in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.

Method 21:

Home on falling edge of home switch without index pulse.

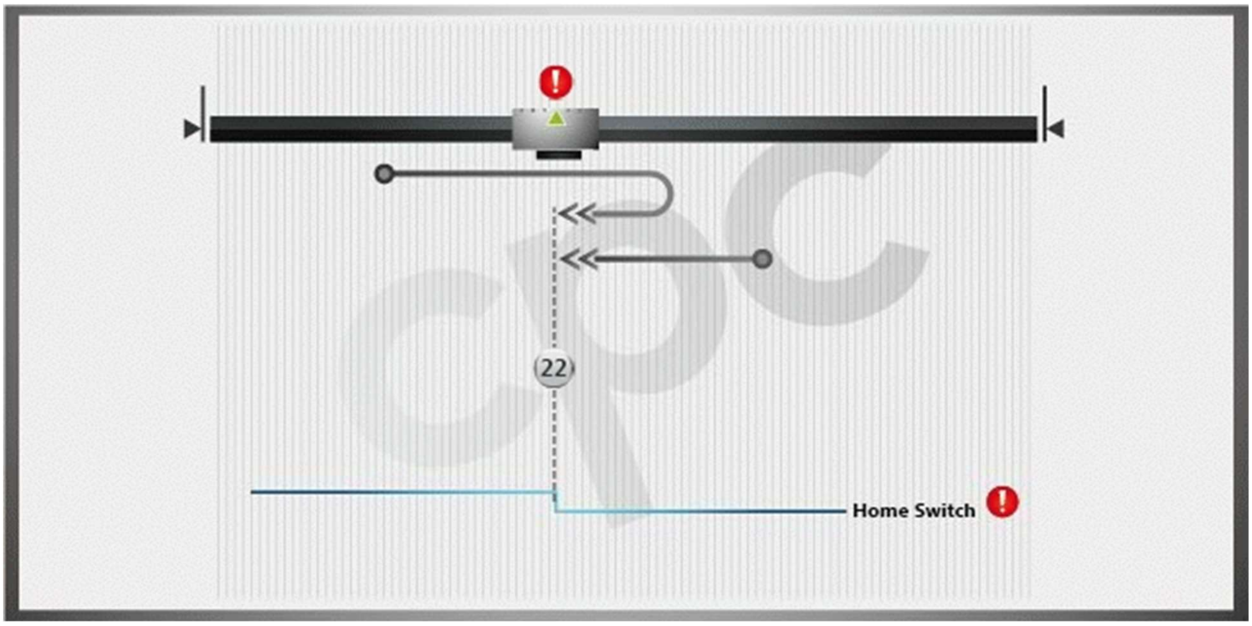


Homing process:

- If home switch is inactive, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.
- If home switch is active, move to falling edge of the home switch in positive direction.

Method 22:

Home on rising edge of home switch without index pulse.



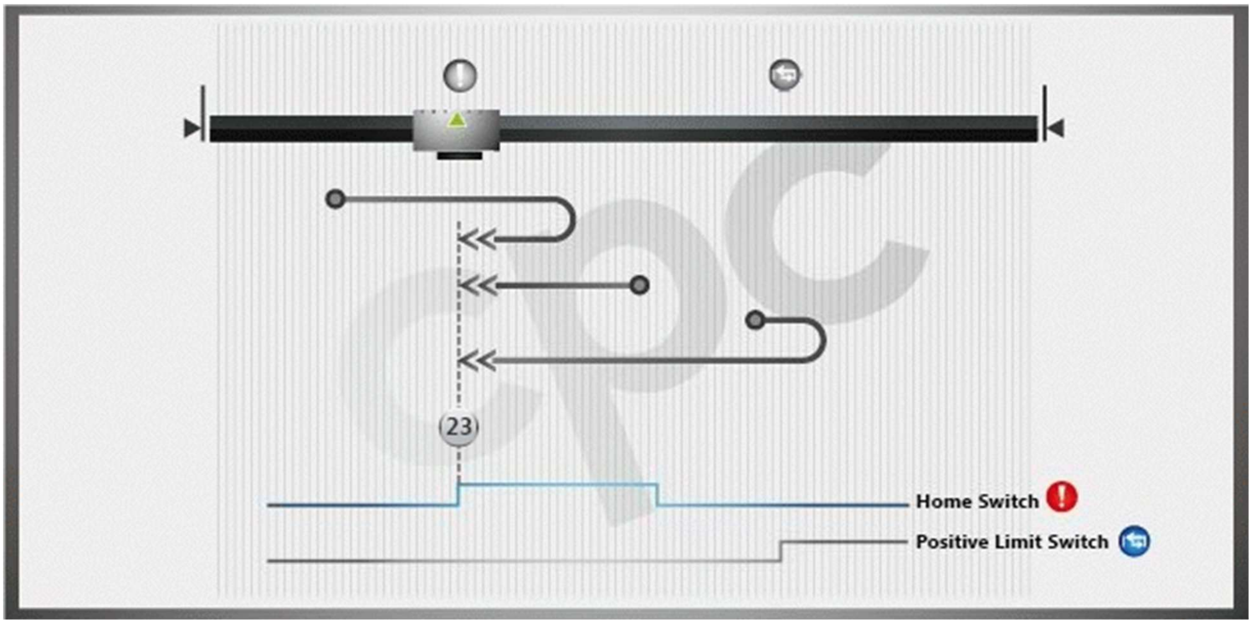
Homing process:

- If home switch is active, move in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is inactive, move to rising edge of the home switch in negative direction.

By Home Switch and Limit Switch

Method 23:

Home on falling edge of home switch while moving in negative direction.

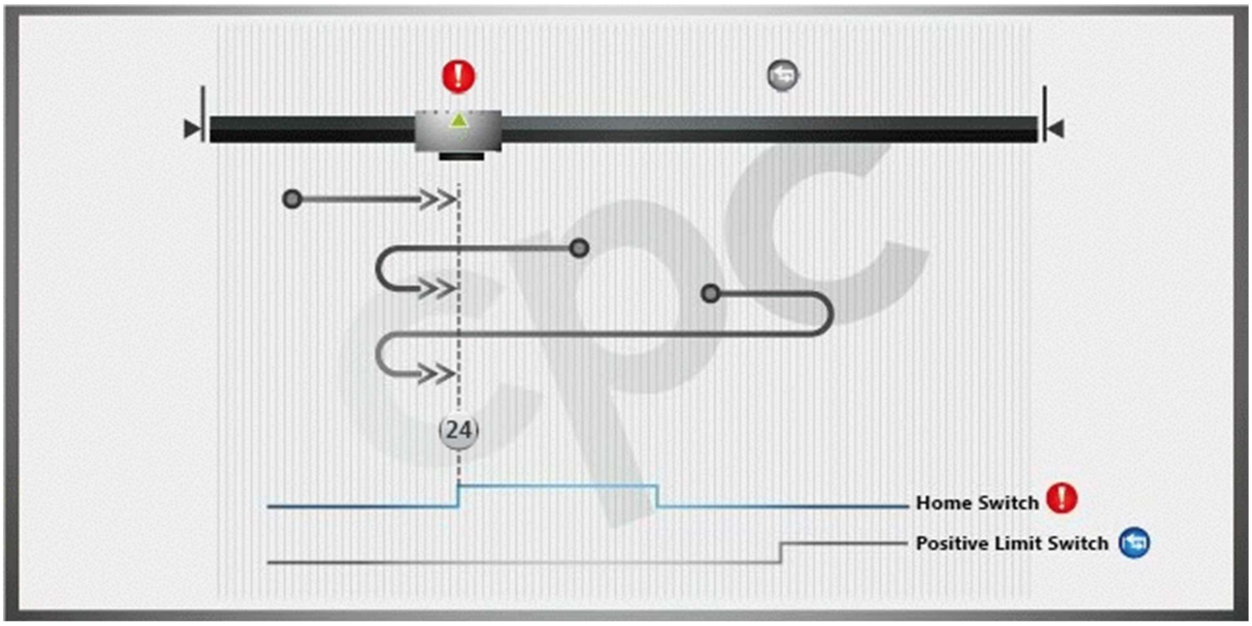


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in negative direction.
- If home switch is active, move to falling edge of the home switch in negative direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction. If home switch is then engaged, keeping moving to locate the falling edge of the home switch.

Method 24:

Home on rising edge of home switch while moving in positive direction.

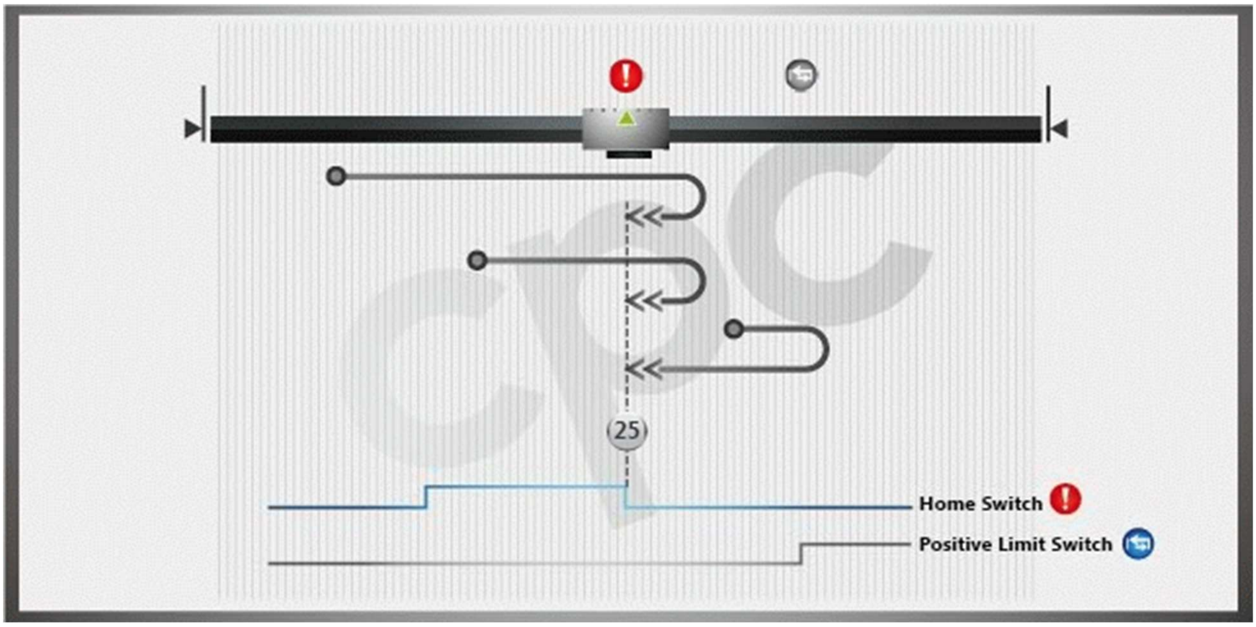


Homing process:

- If home switch is inactive, move to rising edge of the home switch in positive direction.
- If home switch is active, move in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction until home switch is engaged, continue moving in negative direction. If home switch is then disengaged, then move to rising edge of the home switch in positive direction.

Method 25:

Home on rising edge of home switch while moving in negative direction.

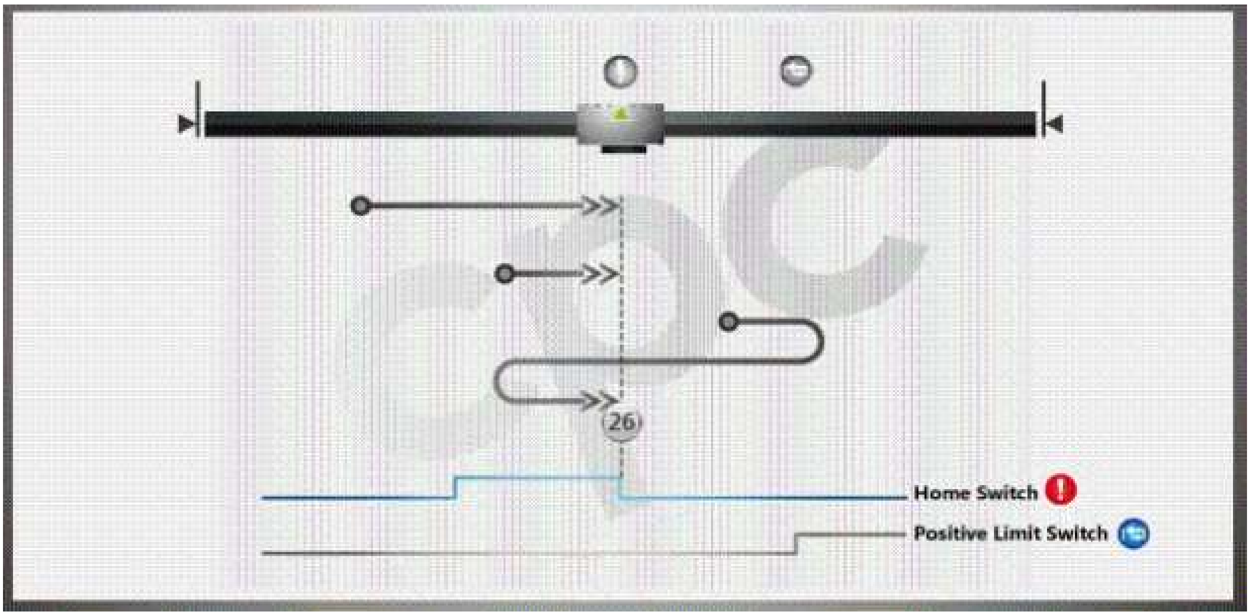


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, continue moving in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is active, move in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move to rising edge of the home switch in negative direction.

Method 26:

Home on falling edge of home switch while moving in positive direction.

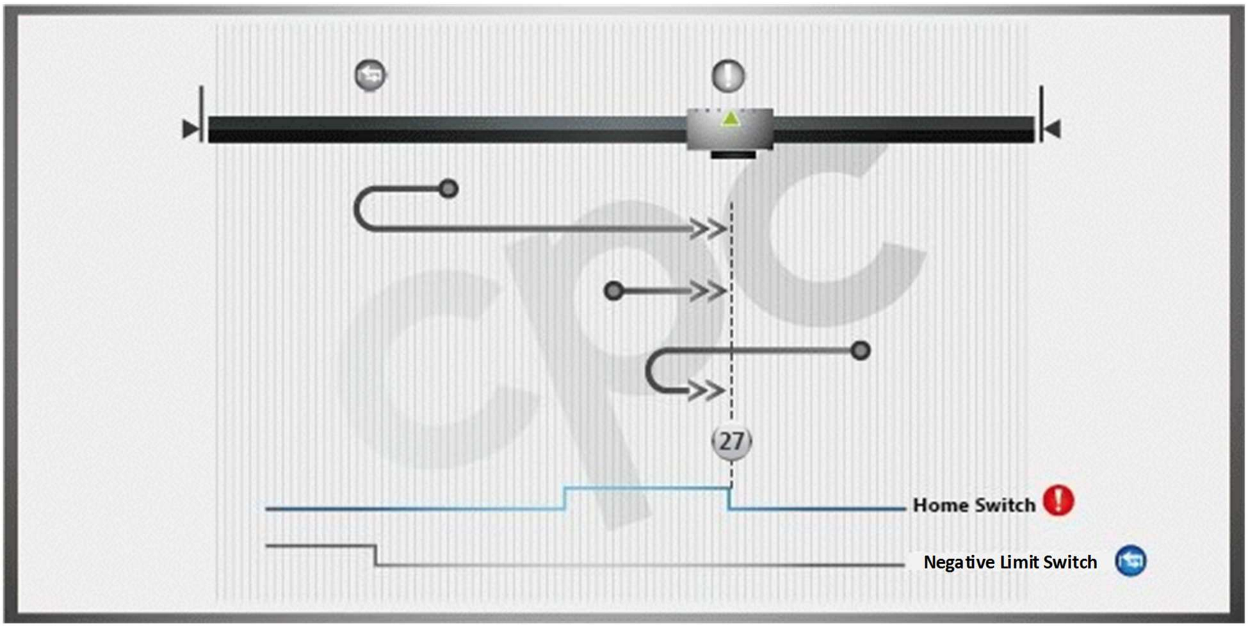


Homing process:

- If home switch is inactive, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.
- If home switch is active, move to falling edge of the home switch in positive direction.
- If home switch is inactive, move in positive direction. If positive limit switch is then engaged, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.

Method 27:

Home on falling edge of home switch while moving in positive direction.

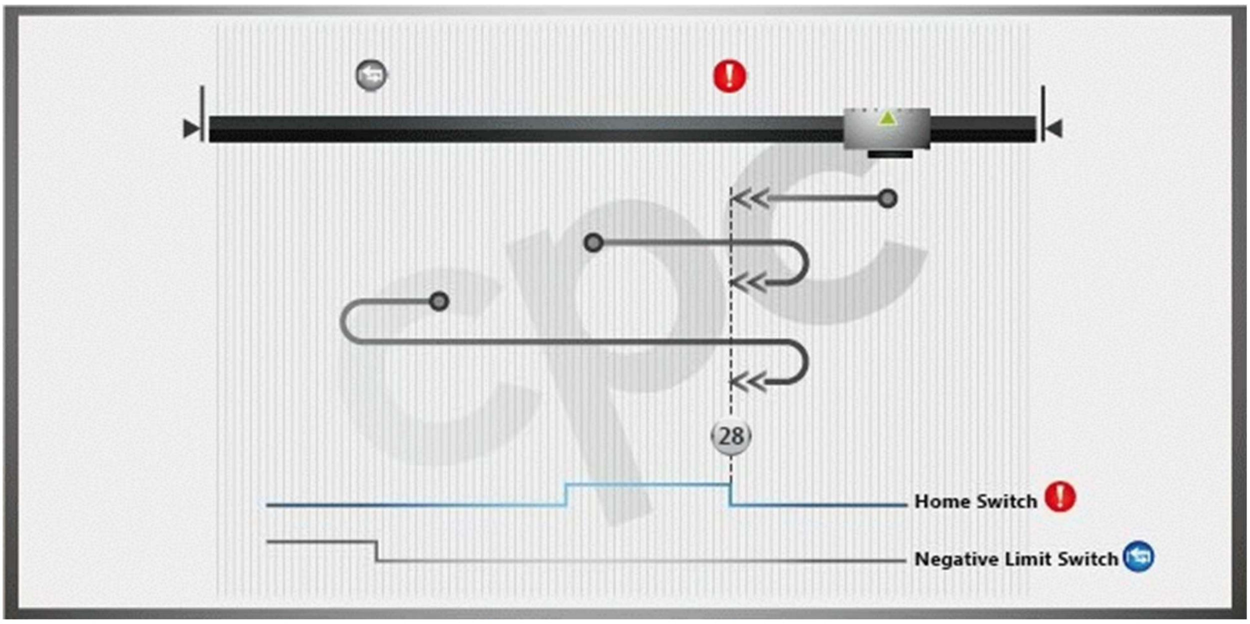


Homing process:

- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.
- If home switch is active, move to falling edge of the home switch in positive direction.
- If home switch is inactive, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in positive direction.

Method 28:

Home on rising edge of home switch while moving in negative direction.

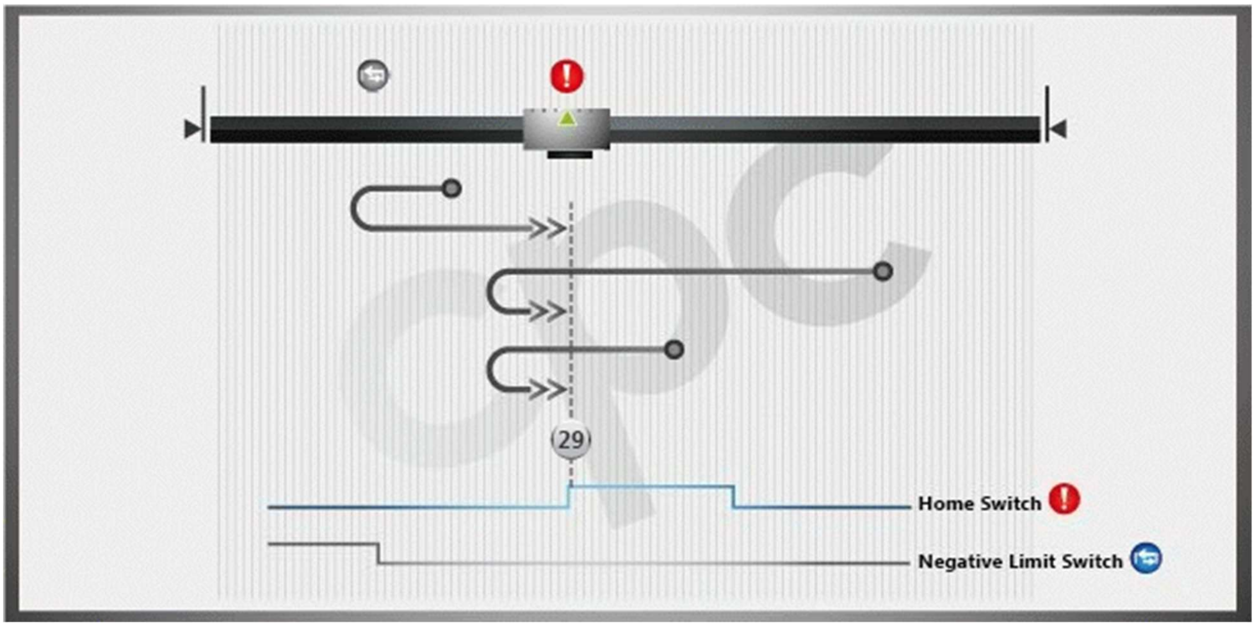


Homing process:

- If home switch is inactive, move to rising edge of the home switch in negative direction.
- If home switch is active, move in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction until home switch is engaged, continue moving in positive direction. If home switch is then disengaged, move to rising edge of the home switch in negative direction.

Method 29:

Home on rising edge of home switch while moving in positive direction.

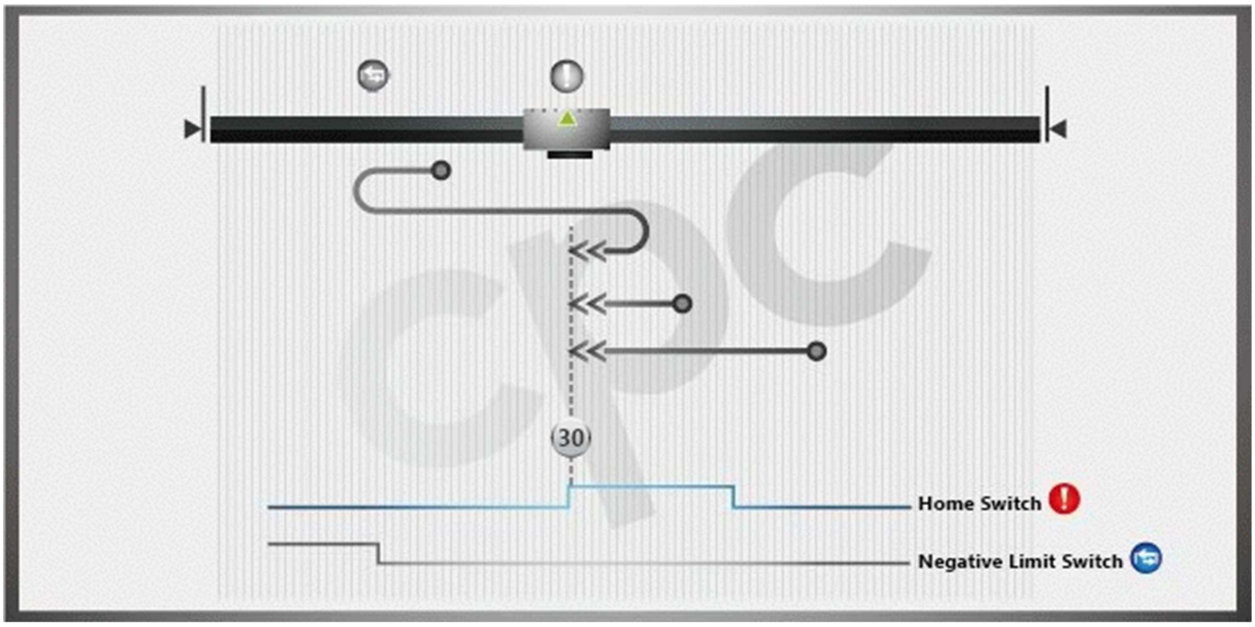


Homing process:

- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move to rising edge of the home switch in positive direction.
- If home switch is inactive, move in negative direction until home switch is engaged, continue moving in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.
- If home switch is active, move in negative direction. If home switch is then disengaged, move to rising edge of the home switch in positive direction.

Method 30:

Home on falling edge of home switch while moving in negative direction.



Homing process:

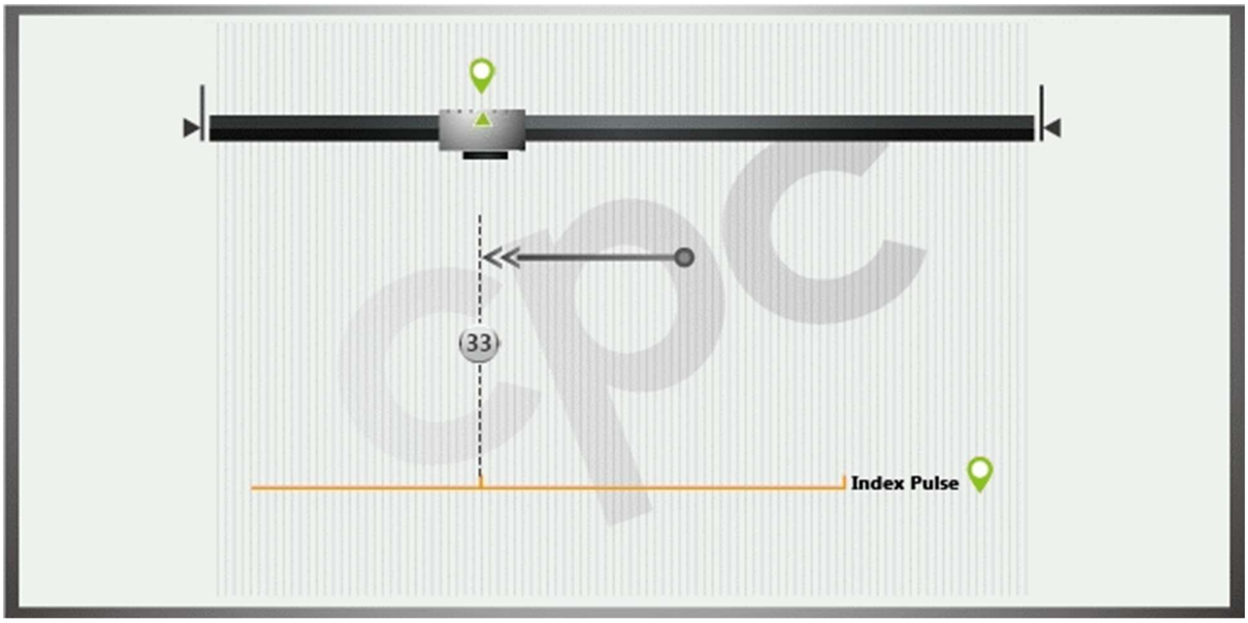
- If home switch is inactive, move in negative direction. If negative limit switch is then engaged, move in positive direction. If home switch is then engaged, move to falling edge of the home switch in negative direction.
- If home switch is active, move to the falling edge of the home switch in negative direction.
- If home switch is inactive, move in negative direction. If home switch is then engaged, move to falling edge of the home switch in negative direction.

Method 31 to 32: Reserved.

By First Pulse

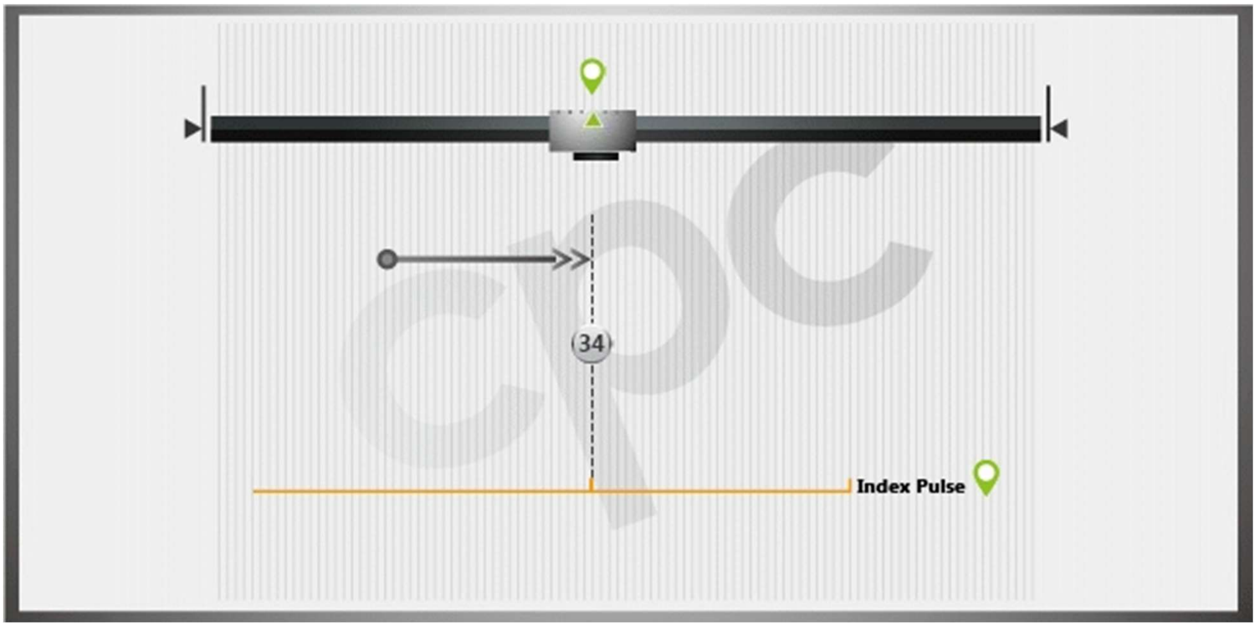
Method 33:

Home on the first pulse while moving in negative direction.



Method 34:

Home on the first pulse while moving in positive direction.



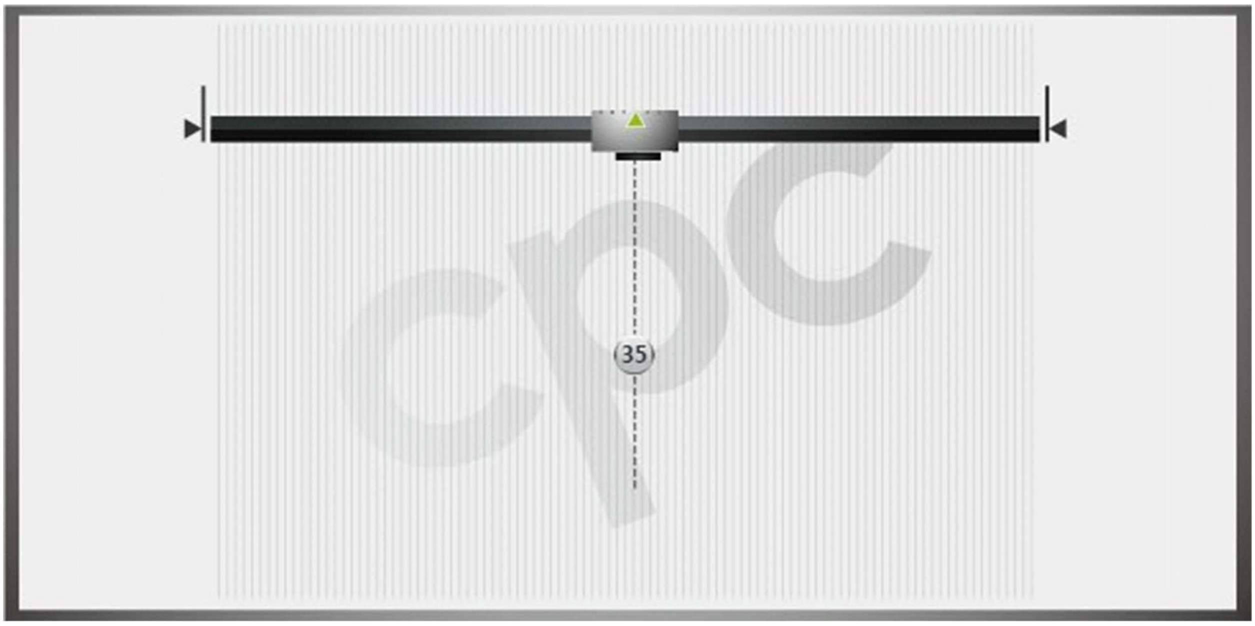
By Current Position

Method 35:

Home on the current position.

***Note**

Method 37 = Method 35

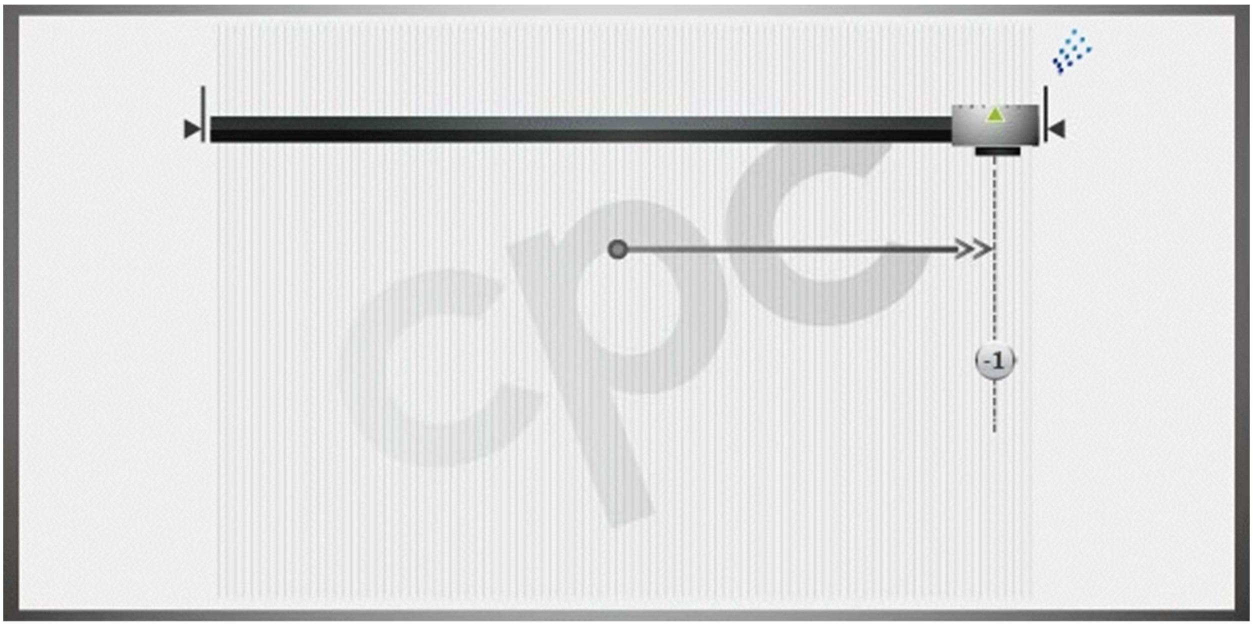


7.2.2 cpc-defined Homing Method

By Hard Stop

Method -1:

Home on the **point** of the **positive** hard stop.

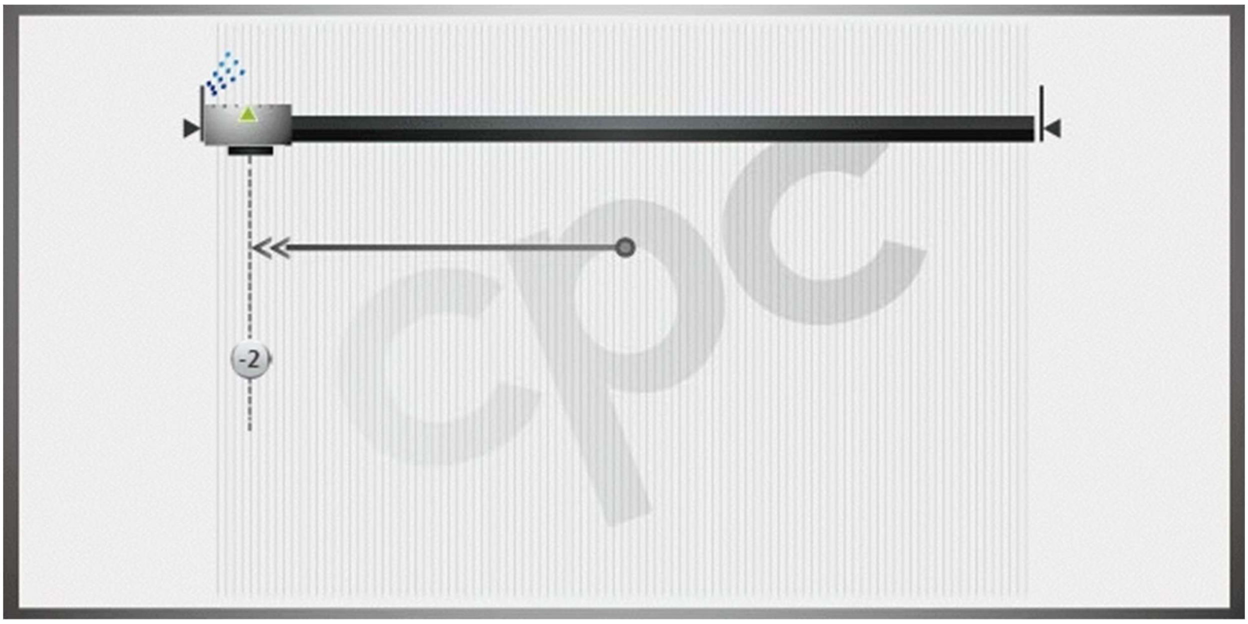


Homing process:

- Start with positive motion unconditionally until the positive hard stop is found.

Method -2:

Home on the **point** of the **negative** hard stop.



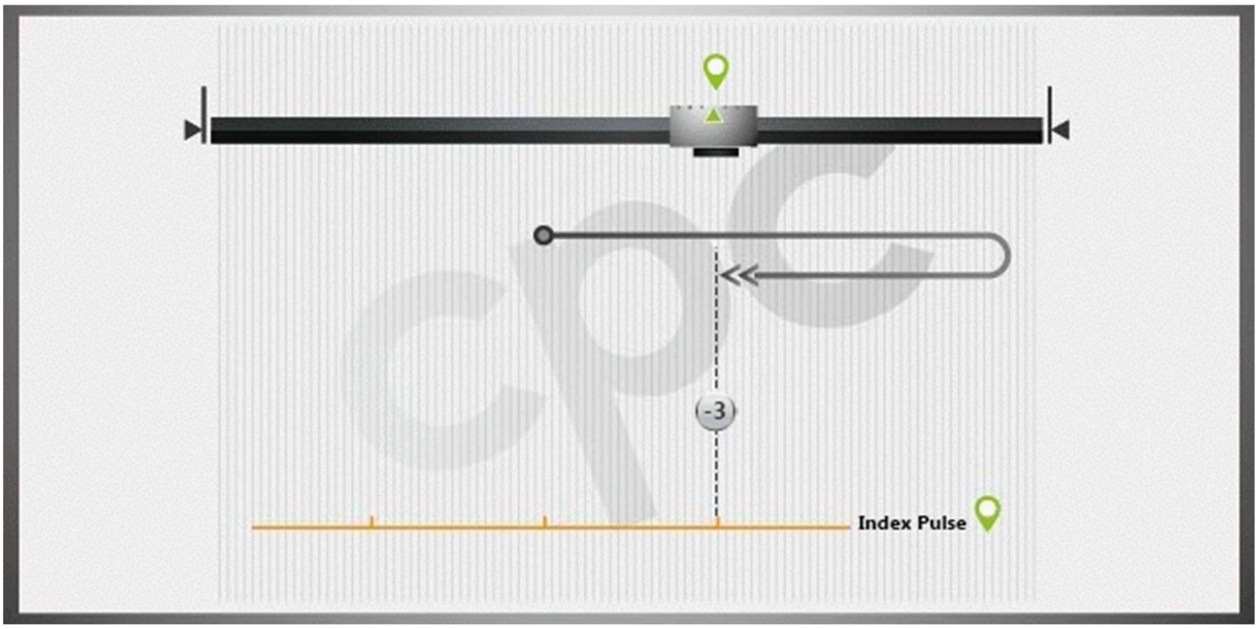
Homing process:

- Start with negative motion unconditionally until the negative hard stop is found.

By Hard Stop and Index

Method -3:

Home on the first **index pulse** after touching the **positive** hard stop.

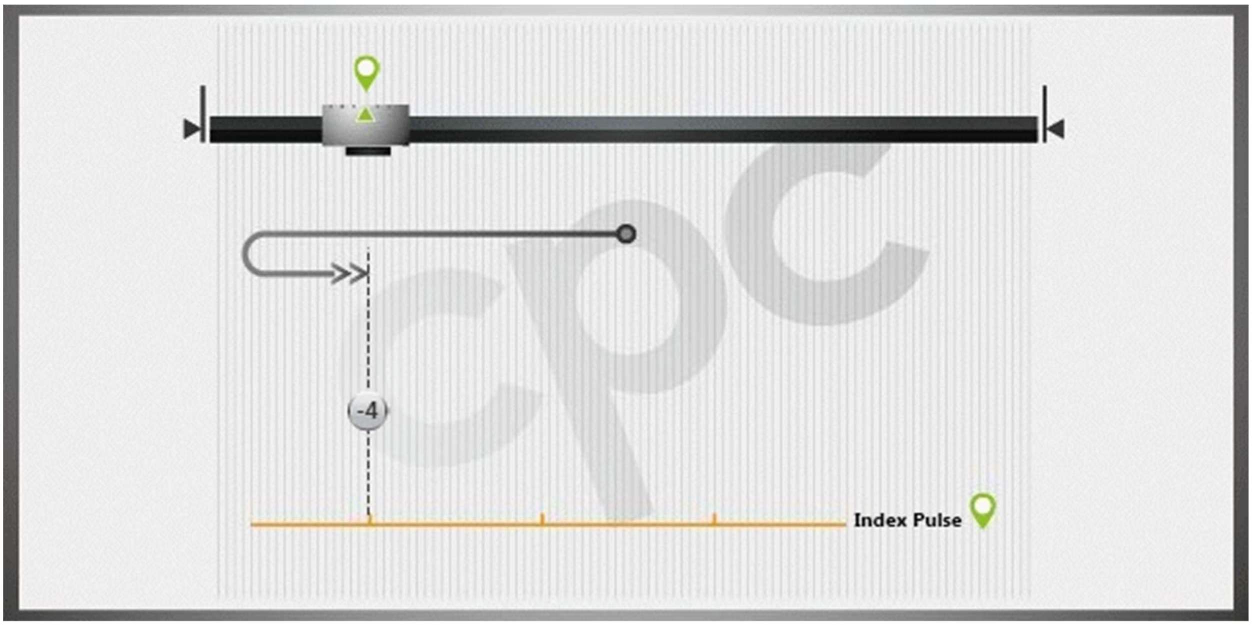


Homing process:

- Start with positive direction unconditionally. After touching the positive hard stop, move in negative direction until the first index is found.

Method -4:

Home on the first **index pulse** after touching the **negative** hard stop.



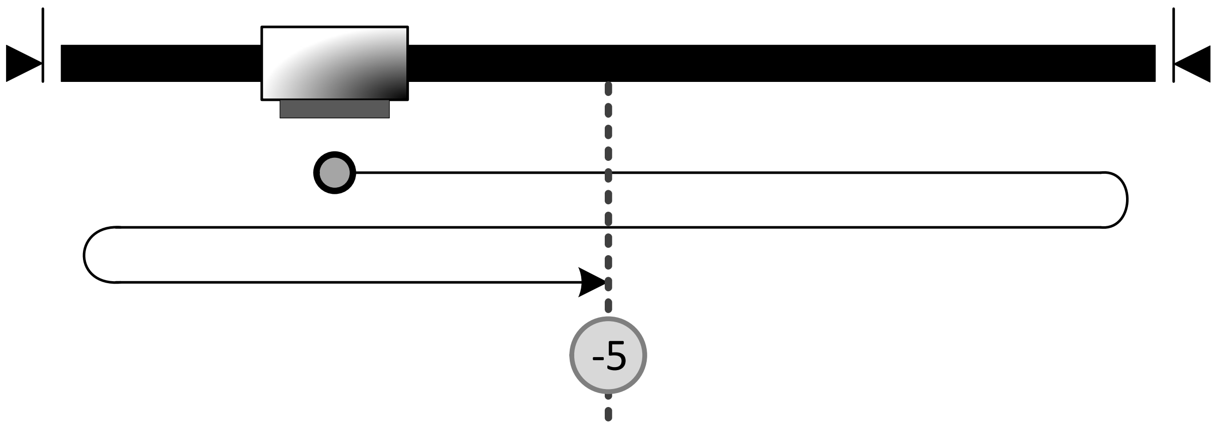
Homing process:

- Start with negative direction unconditionally. After touching the negative hard stop, move in positive direction until the first index is located.

By the middle of Hard Stop

Method -5:

Find middle between forward/backward hard stop, initial direction forward.

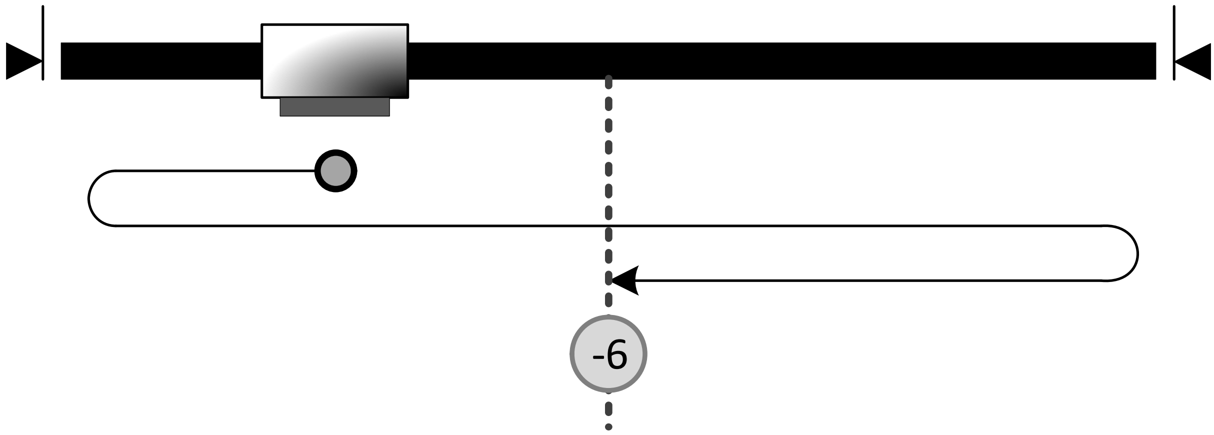


Homing process:

- Start with positive direction unconditionally. After touching the positive hard stop, move in negative direction until touching the negative hard stop, and then home on the middle of the two hard stops (found during homing).

Method -6:

Find middle between forward/backward hard stop, initial direction backward.



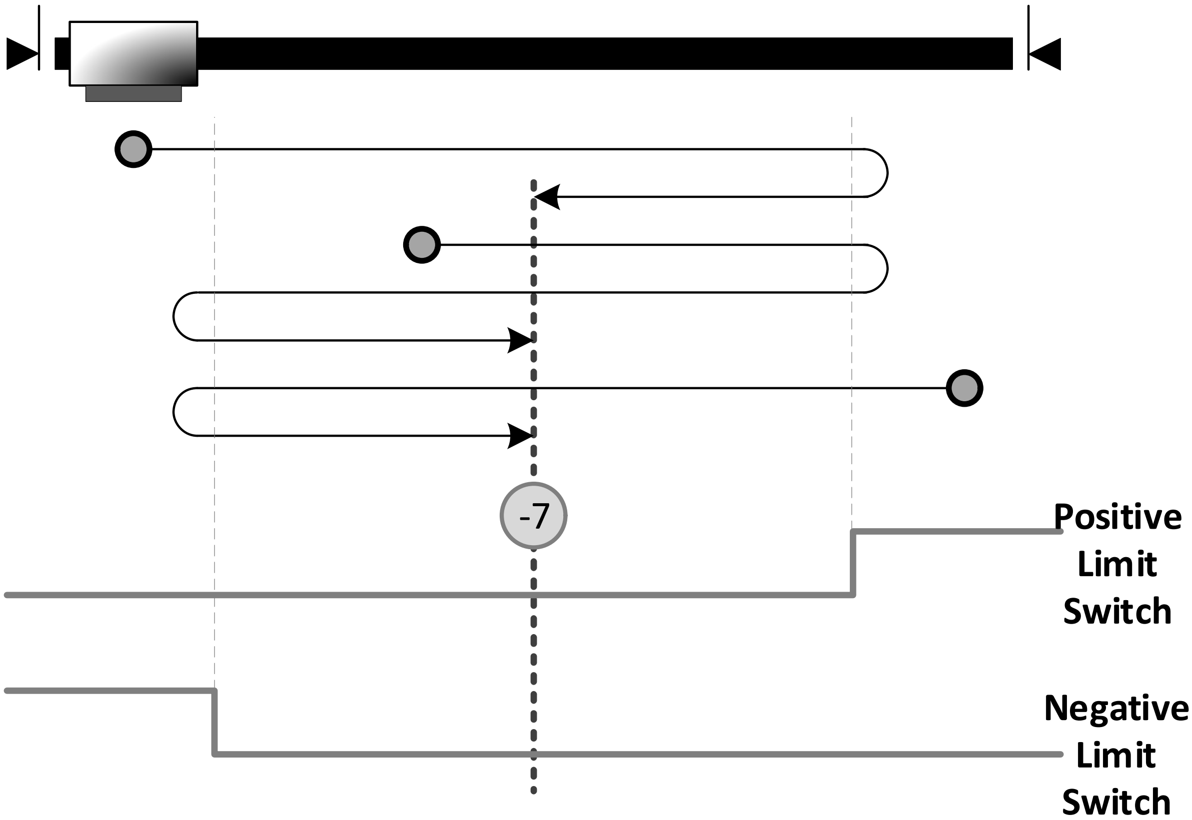
Homing process:

- Start with negative direction unconditionally. After touching the negative hard stop, move in positive direction until touching the negative hard stop, and then home on the middle of the two hard stops (found during homing).

By the middle of Limit Switch

Method -7:

Find middle between forward/backward limit switch falling edge, initial direction forward.

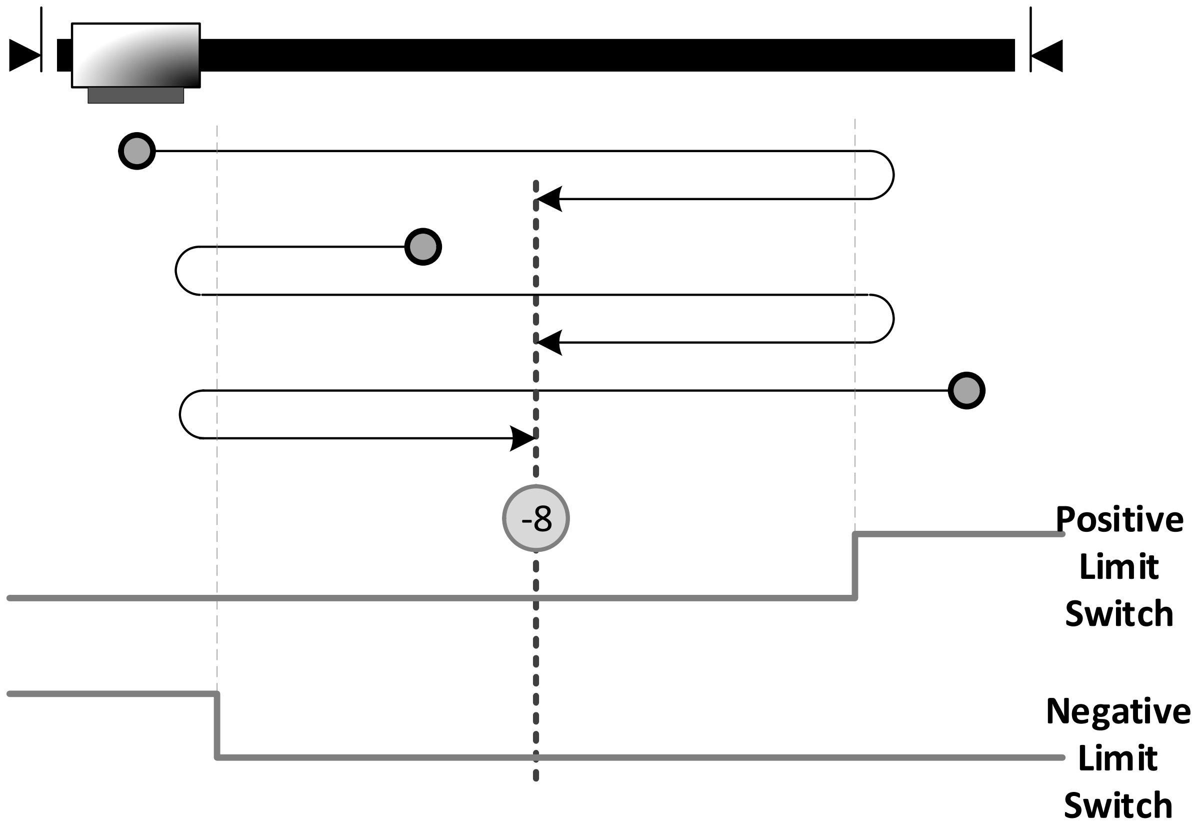


Homing process:

- If negative limit switch is active, start with positive direction until the positive limit switch is engaged; then, move in negative direction to find the middle of both switches.
- If negative limit switch is inactive, start with positive direction until the positive limit switch is engaged and then move in negative direction. If negative limit switch is engaged, move in positive direction until the middle of both switches is found.
- If the positive limit switch is active, start with negative direction until the negative limit switch is engaged; then, move in positive direction to find the middle of both switches.

Method -8:

Find middle between forward/backward **limit switch** falling edge, initial direction backward



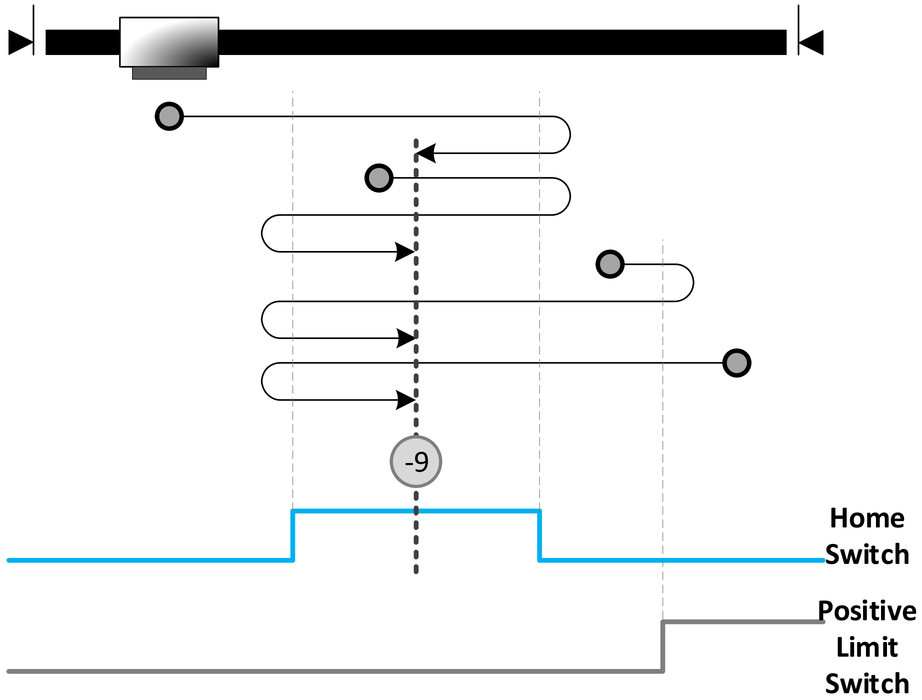
Homing process:

- If negative limit switch is active, start with positive direction until the positive limit switch is engaged; then, move in negative direction to find the middle of both switches.
- If negative limit switch is inactive, start with negative direction until the negative limit switch is engaged and then move in positive direction. If the positive limit switch is then engaged, move in negative direction until the middle of both switches is found.
- If the positive limit switch is active, start with negative direction until the negative limit switch is engaged; then, move in positive direction to find the middle of both switches.

By the middle of Home Switch

Method -9:

Find middle of home switch falling edge, initial direction forward, allow limit switch.

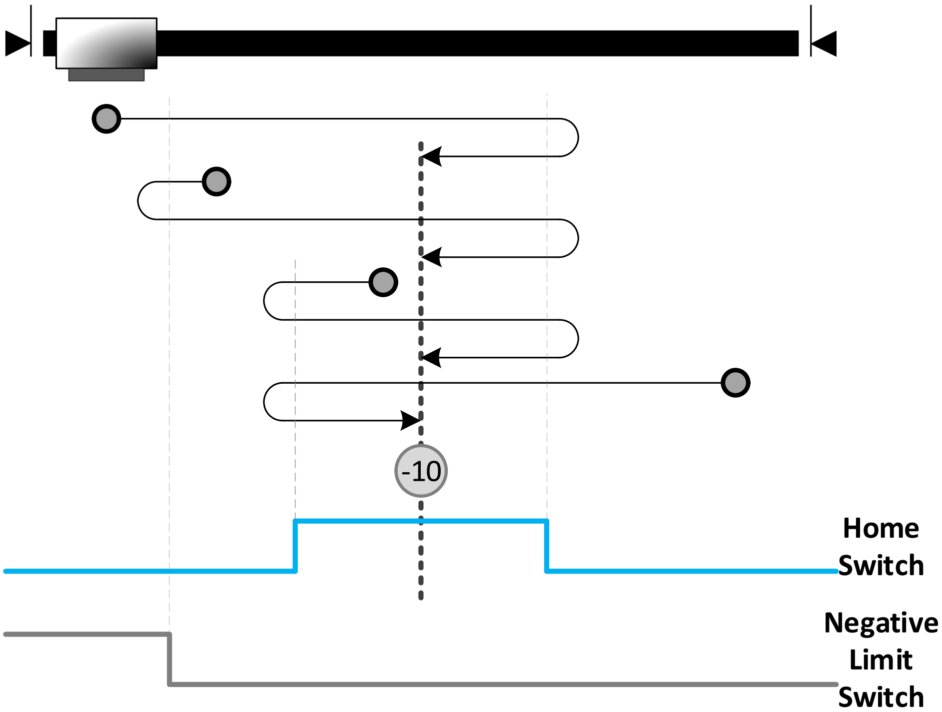


Homing process:

- If the home switch is inactive, start with positive direction until the home switch is engaged and disengaged, then move in negative direction until the middle of home switch is found.
- If the home switch is active, move in positive direction until the home switch is disengaged, then move in negative direction until the home switch is engaged and then disengaged. Reverse to move in positive direction and find the middle of home switch.
- If the home switch is inactive, move in positive direction. If the positive limit switch is then engaged, move in negative direction until the falling edge of the home switch is engaged, then, move in positive direction until the middle of the home switch is found.
- If positive limit switch is active, move in negative direction until the falling edge of home switch is met, then move in positive direction and find the middle of home switch.

Method -10:

Find middle of home switch falling edge, initial direction backward, allow limit switch

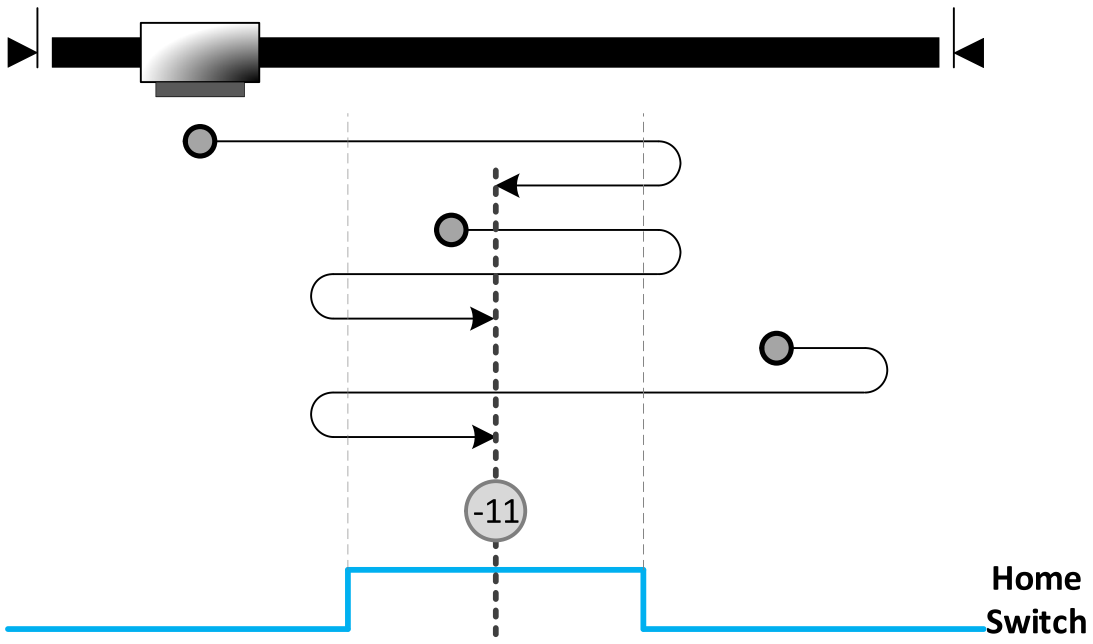


Homing process:

- If the negative limit switch is active, move in positive direction until the home switch is engaged and then disengaged. Then move in negative direction until the middle of the home switch is found.
- If the home switch is inactive, start with negative direction. If the negative limit switch is then engaged, move in right direction until the falling edge of the home switch is met. Then move in negative direction to find the middle of the home switch.
- If home switch is active, move in negative direction. If the home switch is then disengaged, move in positive direction until the home switch is engaged and disengaged. Then, move in negative direction until the home switch is engaged again and find the middle of home switch.
- If the home switch is inactive, move in negative direction until the falling edge of home switch. If the home switch is then disengaged, move in positive direction until the middle of the home switch is found.

Method -11:

Find middle of home switch falling edge, initial direction forward, allow hard stop.

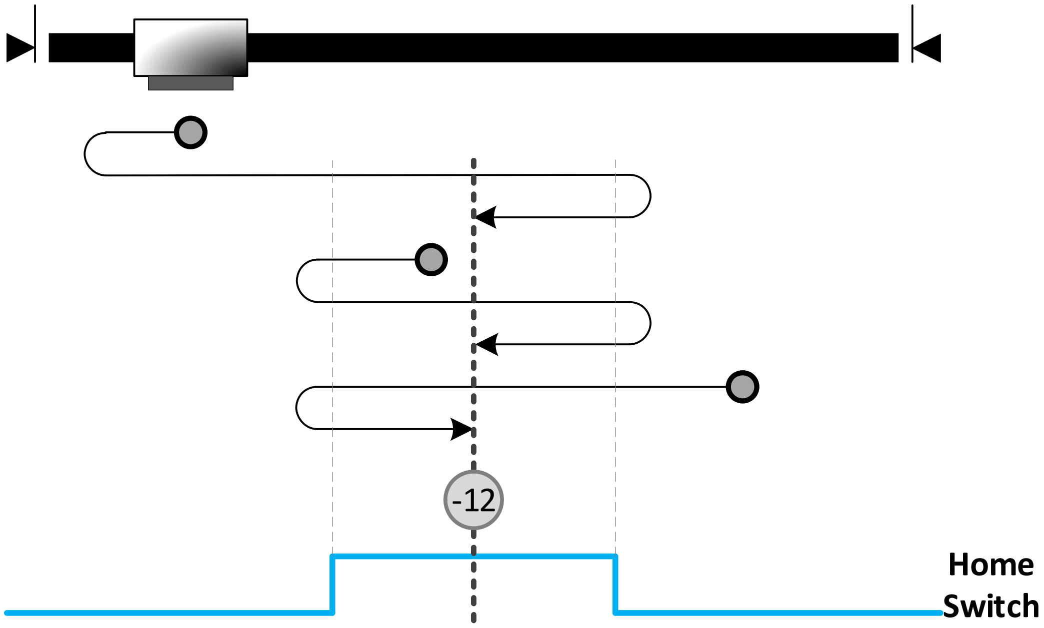


Homing process:

- Start with positive direction unconditionally. If home switch is inactive, keep going (and then if hard stop is met, reverse firstly) until meeting the rising and falling edge of the home switch, then reverse again to locate the middle of home switch.
- Start with positive direction unconditionally. If home switch is active, seek the positive side of home switch, then move in negative direction to meet the negative side of home switch, then reverse to find the middle of home switch.

Method -12:

Find middle of home switch falling edge, initial direction backward, allow hard stop.



Homing process:

- Start with negative direction unconditionally. If home switch is inactive, keep going (and then if hard stop is met, reverse firstly) until meeting the rising and falling edge of the home switch, then reverse again to locate the middle of home switch.
- Start with negative direction unconditionally. If home switch is active, go meet the negative side of home switch, then move in positive direction to meet the positive side of home switch, then reverse to find the middle of home switch.

7.3 Homing Error Code

Error code	Description
1	No error
2	Invalid FSM state
3	Invalid HmCfgBits
4	No valid homing method set
5	Wrong Home Switch edge encountered
6	Direction mismatch for motor motion when Home Switch is searched
7	Direction mismatch for motor motion when index is searched
8	Encountered overlapping limit switch
9	Unexpected encounter of Forward Limit Switch
10	Unexpected 2^{nd} encounter of Forward Limit Switch
11	Unexpected encounter of Backward Limit Switch
12	Unexpected 2^{nd} encounter of Backward Limit Switch
13	Unexpected encounter of Forward Hard Stop
14	Unexpected 2^{nd} encounter of Forward Hard Stop
15	Unexpected encounter of Backward Hard Stop
16	Unexpected 2^{nd} encounter of Backward Hard Stop
17	We have traveled from one limit switch to another without satisfying homing condition
18	We have traveled from one Hard Stop to another without satisfying homing condition

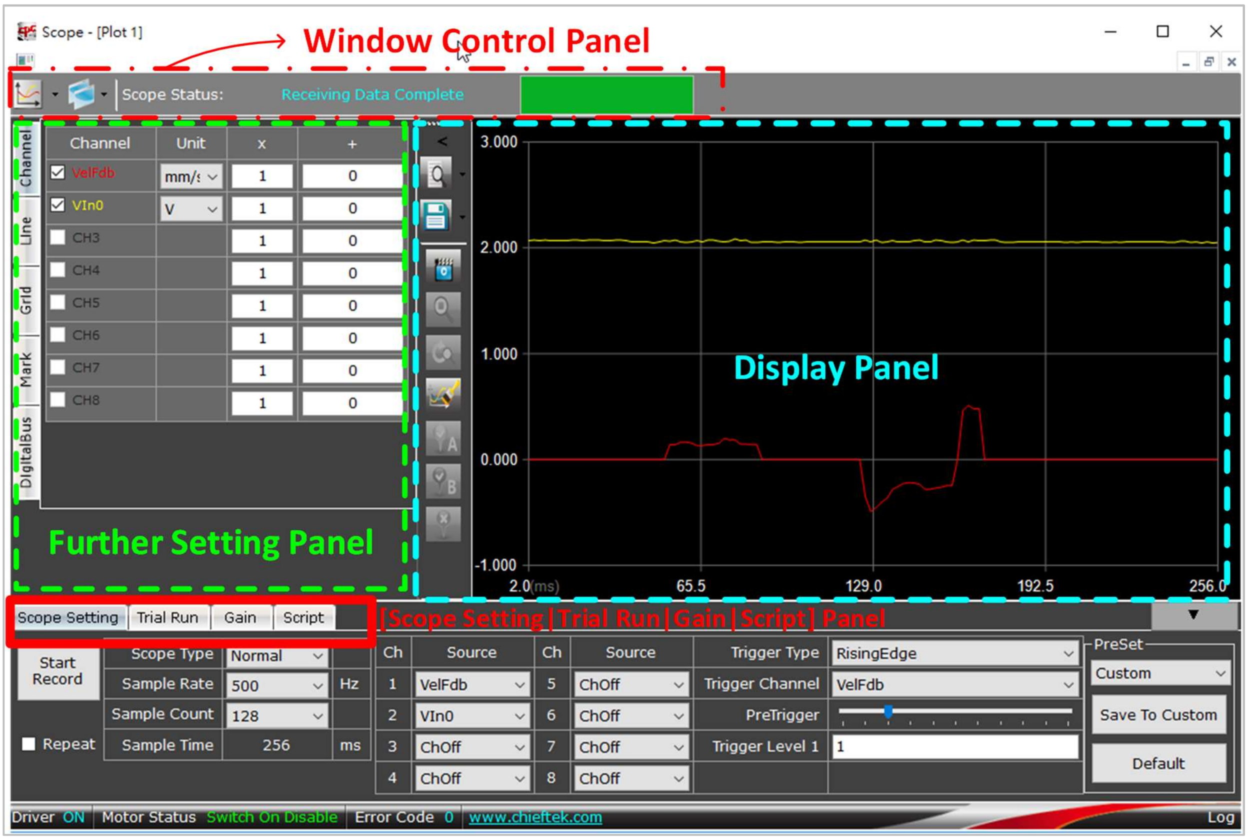
Chapter 8 Scope

8.1 Intro and Interface Tour

The function of Scope is similar to that of an oscilloscope.

The Scope interface has 4 main panels:

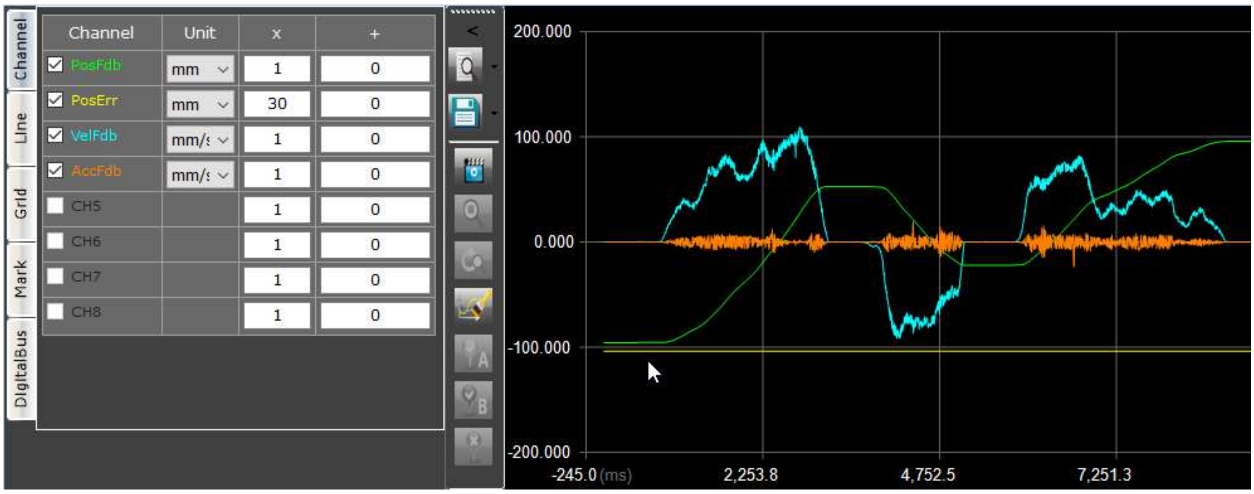
1. **Window Control Panel** – on the top of the interface.
2. **[Scope Setting|Trial Run|Gain|Script] Panel** – on the bottom of the interface, it's the major configuration panel.
3. **Display Panel** – on the right, for viewing results.
4. **Further Setting Panel** – on the left, finer settings for the display panel.



You can use Scope to monitor several variables by setting them as the channels' data source (see **chapter 6.1**). Also, you can configure the data unit and graph details. Moreover, Scope allows you to record data and alter the data presentation way.

After the recording and graph drawing is completed, you can observe the ripple and resonance frequency via the drawn plot.

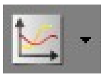
Take the picture below for example, you'll find position feedback, position error, velocity feedback, and acceleration feedback during a movement.



As for generating a movement(s), please refer to **chapter 6**. Trial run and **chapter 9.** Script.

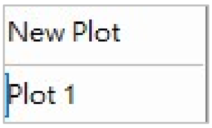
The recorded data can be saved in txt. format (text file) and the plot in .png format.

8.2 Window Control Panel

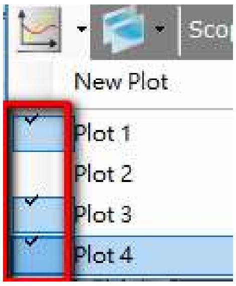
a.  **Plot**:

Move the cursor onto the downward arrow.

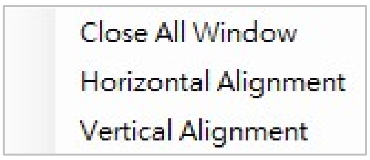
- Click “New Plot” to create up to 4 monitoring windows.



- You can select which window(s) to be shown on the Display Panel by ticking the title of the preferred window(s).



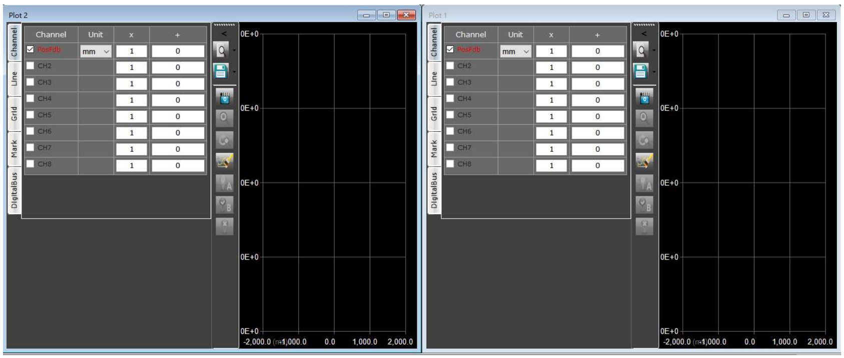
b.  **Window**:



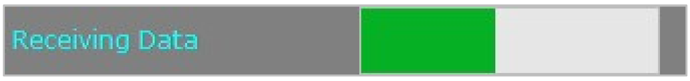
Select to align windows horizontally or vertically, or close them all. See images below.

(↓ Horizontal Alignment)

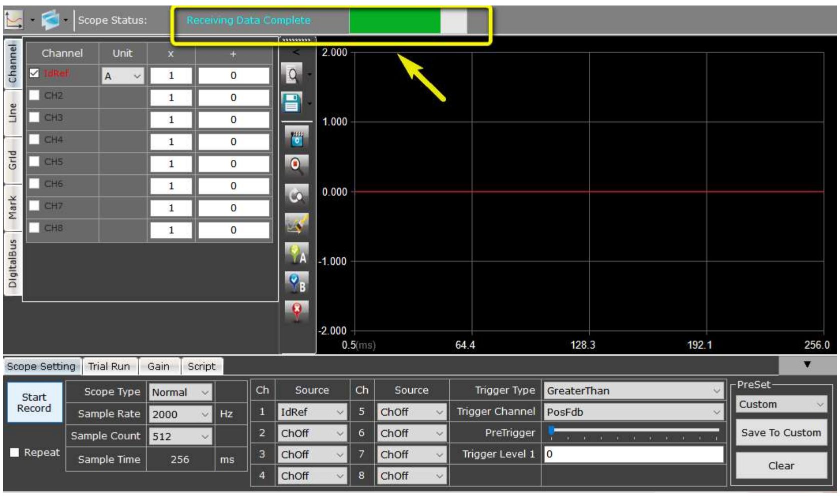
(↓ Vertical Alignment)



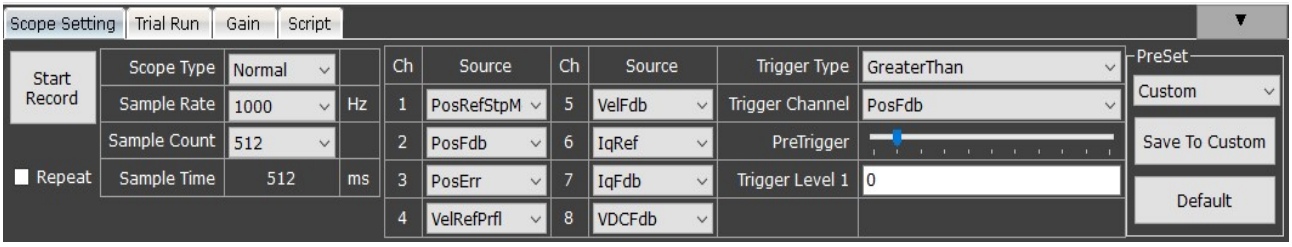
c. **Scope Status**:



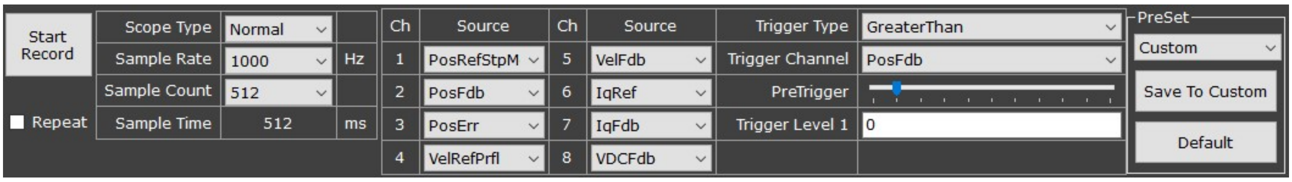
Indicates the status of data processing, e.g., **Waiting Trigger, Recording, Receiving Data,** and **Receiving Data Complete.**



8.3 [Scope Setting|Trial Run|Gain|Script] Panel



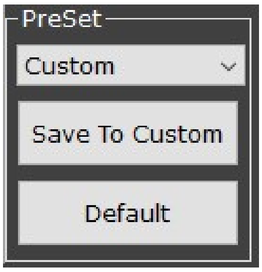
8.3.1 Scope Setting Tab



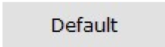
Like using an oscilloscope, users need to set up first the scope type, rate, sample count, and the data (channel number and the source).

Note: See the channel data source in **chapter 6.1** Monitor.

8.3.1.1 Preset



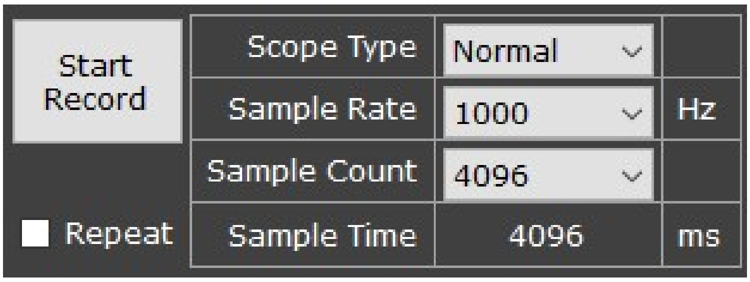
We cpc provide 2 factory presets (Profile Position and Profile Velocity) and 1 customized preset.

- You can click "Default" 

 to clear all settings then configure parameters as you wish.
- In Custom mode, click "Save To Custom"

 to save your preferred settings.

8.3.1.2 Recoding Setting



a. **Scope Type**:

Normal or Rolling; see **chapter 8.3.1.3.**

b. **Sample Rate:**

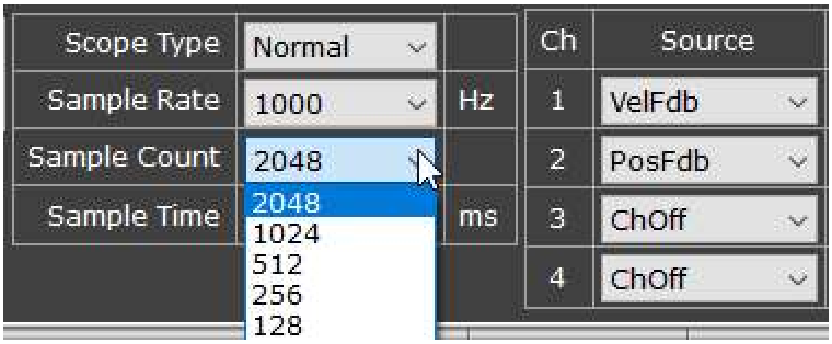
The frequency (Hz) of retrieving the data.

c. **Sample Count:**

The maximum data counts for each channel.

Note:

When using the "Normal" Scope Type, the system's maximum total data count is **4096** (counts). Hence, if there are 2 channels set, the maximum sample count for each channel shown on the list will become 2048 counts.



d. Sample Time:

The time duration needed for recording (based on the configuration of sample rate and sample count).

Sample Time (ms) **= Sample Count ÷ Sample Rate** (Hz) **x 1000**.

e. Start Record:

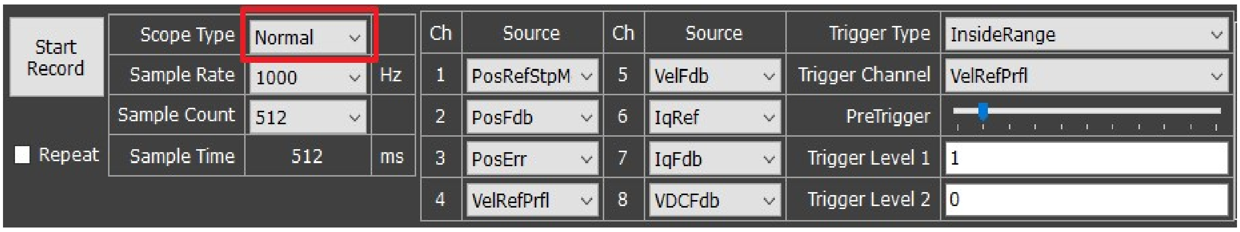
Click to start recording data.

When using the "Normal" Scope Type, if there are triggering conditions set, click this button; next, when conditions are satisfied the system will automatically start recording.

<u>**Note:**</u>

Any change of data source will need to be activated by clicking the "Start Record" button.

<u>8.3.1.3 Normal/Rolling and Trigger Setting</u>



There are 2 types of recording (or called “Scope Type”): **Normal** and **Rolling.**

- Normal type has higher sample rate than that of Rolling type.
- Normal type saves data in the driver first and then sends data to the UI <u>at one time</u>. Rolling type sends data continuously only to the UI.

a. Scope Type:

- Normal:

  Records the data for a certain period of time after the triggering conditions are satisfied. If there is no triggering condition set, click “Start Record” to record immediately.

- Rolling:

  Starts recording immediately upon clicking this button. The UI will refresh continuously.

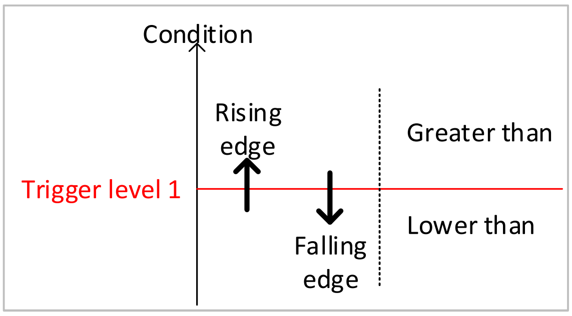
<u>The following functions (b and c) are for the "Normal" scope type only</u>:

b. ☐ Repeat **:**

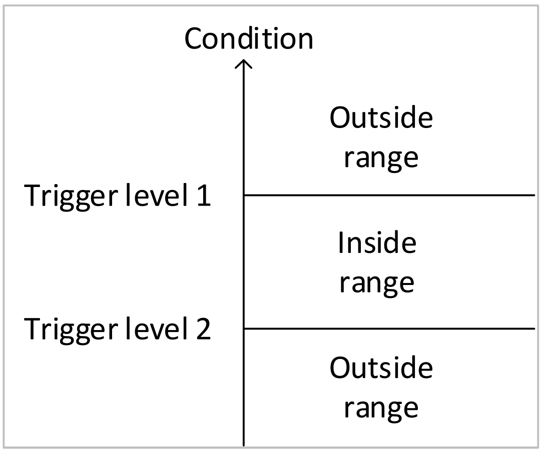
Only applicable under the **Normal** scope type. Tick to repeat the cycle of completing data receiving then recording (see **chapter 8.2**, **point c.**).

c. Trigger Type:

- NoTrigger:
  Start recording immediately upon clicking the button.

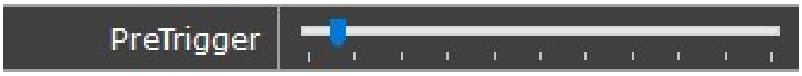


- RisingEdge:
  Rising edge trigger on the level.
- FallingEdge:
  Falling edge trigger on the level.
- LowerThan:
  The value lower than the level.
- GreaterThan:
  The value greater than the level.

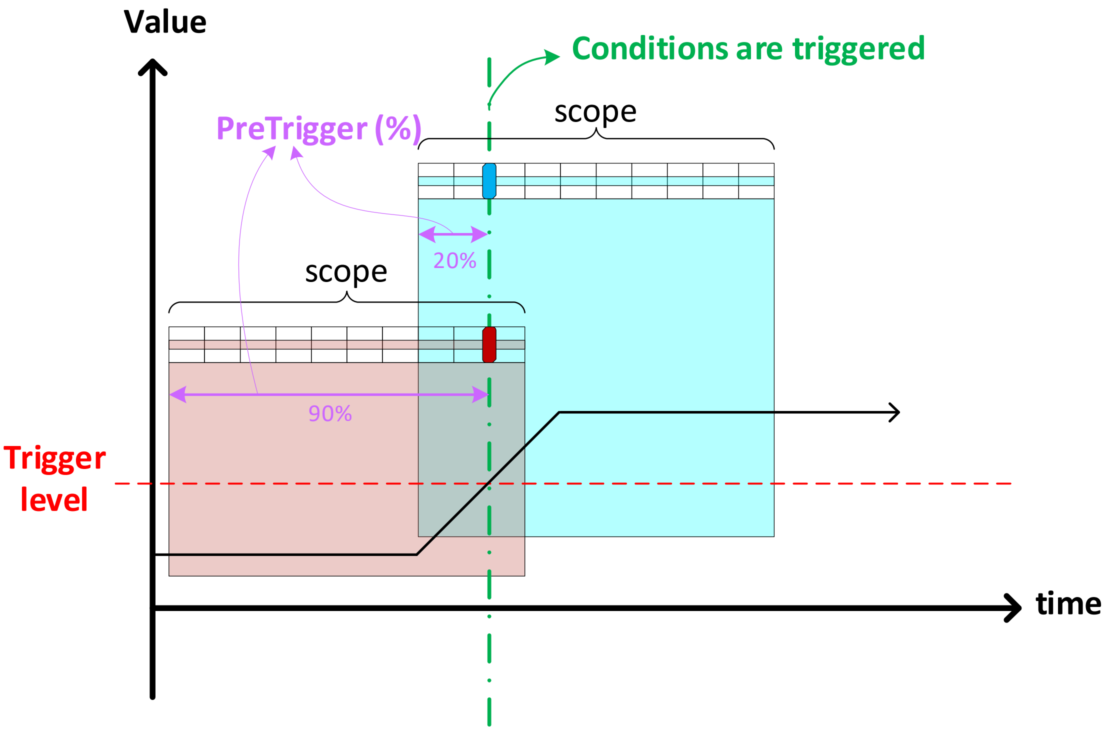


- InsideRange:
  Values between level 1 & level 2.
- OutsideRange:
  Values greater than the level 1 or lower than the level 2.
- Trigger Channel:
  Choose the trigger source.

- PreTrigger:



Drag to configure how much portion (%) of a record (namely, the recorded data) will be the part where the triggering conditions are not satisfied.



The scope on the left is with PreTrigger 90%, meaning that **90%** of the data shown on the *left* scope is below trigger condition(s). Likewise, for the scope on the right, **20%** of the data shown on the *right* scope is below trigger condition(s)

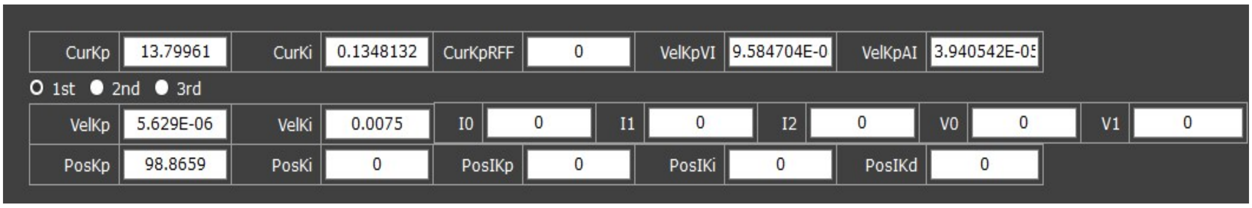
- Trigger Level 1 & 2:

Set the threshold of the trigger.

8.3.2 Trial Run Tab

Please refer to chapter 6 Trial Run.

8.3.3 Gain Tab

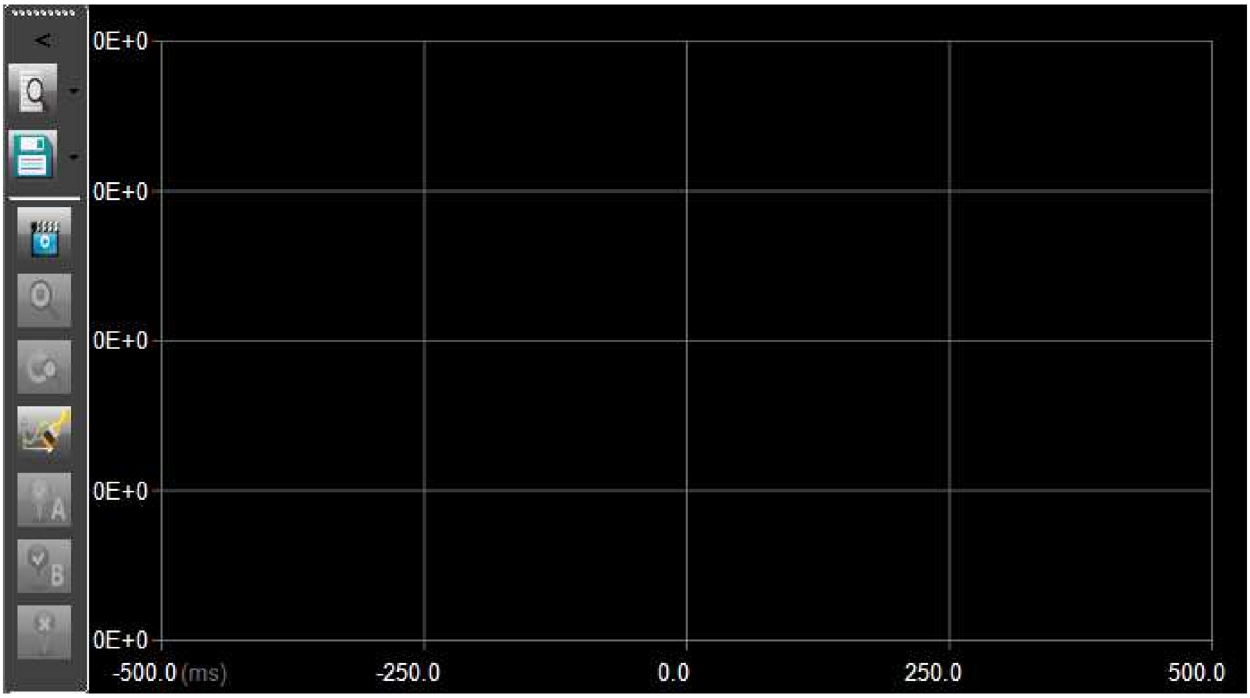


Users can set the gain parameters via this panel directly.

8.3.4 Script Tab

Click to execute script.

8.4 Display Panel



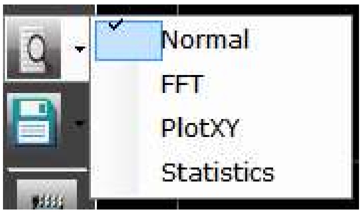
The display panel shows you the monitoring results.
You can change the data presentation way and save the monitored data in TXT., or PNG. format.

a. :

The Display Panel can be enlarged using the arrow button.

8.4.1 Data Presentation Way

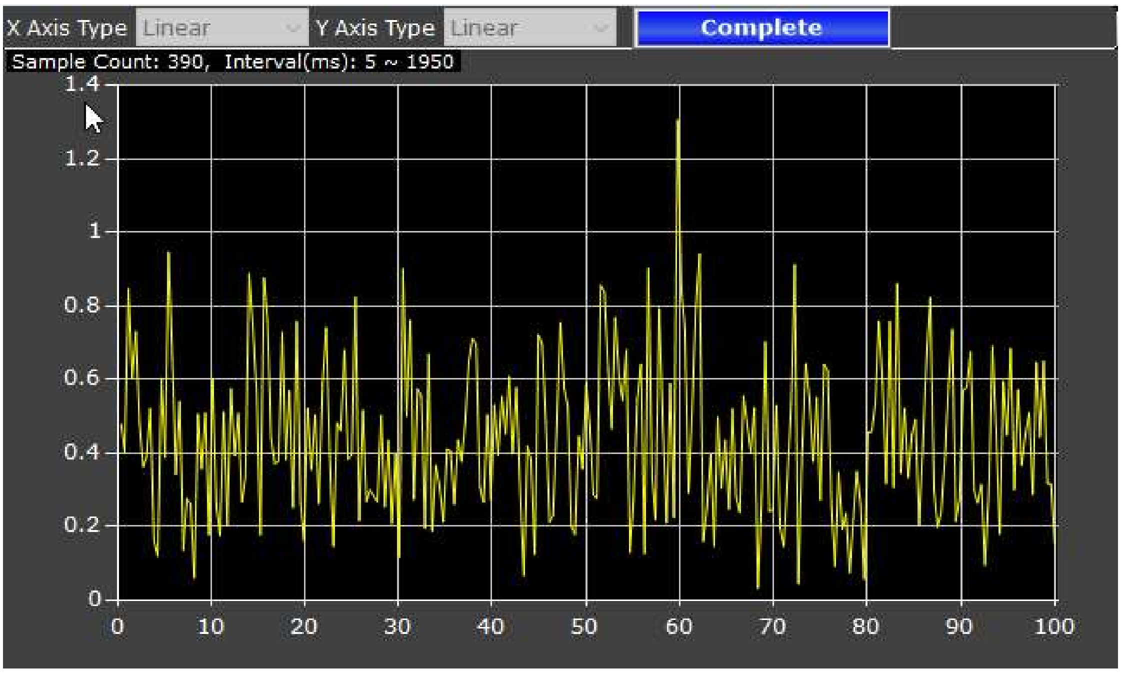
Click the arrow beside Plot Type icon to select a preferred way of data presentation. The ways are: Normal, FFT, PlotXY, and Statistics.



a. Normal

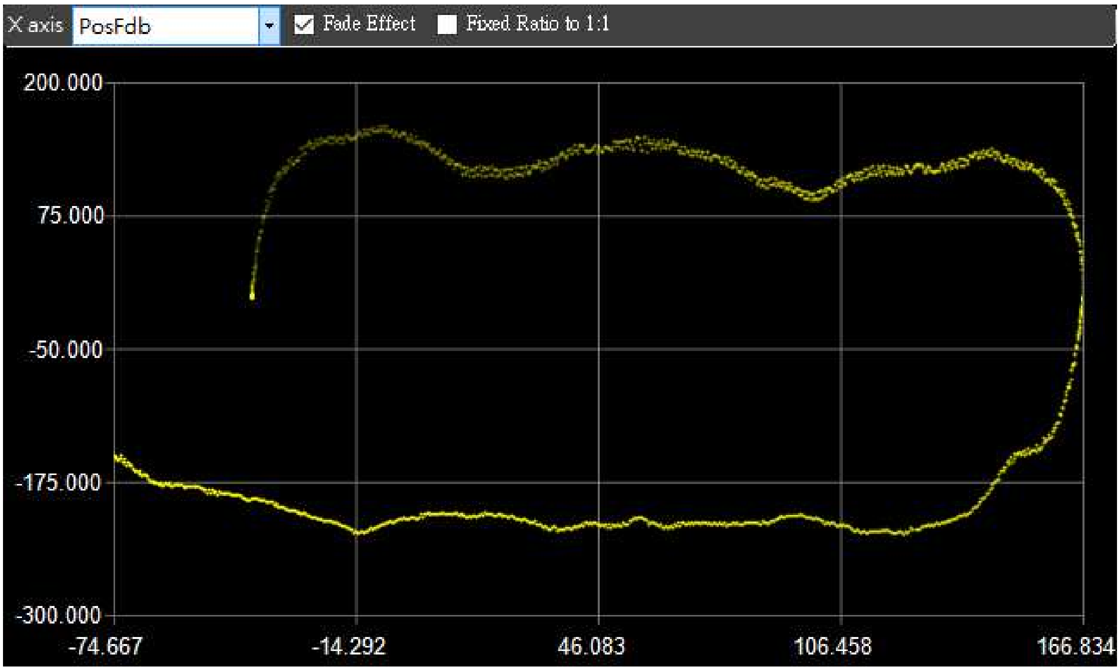
b. FFT

Fast Fourier Transform for observing the spectrum analysis.



c. PlotXY:

This X-Y plot can map 2 channels to the plane.



- **X axis**:

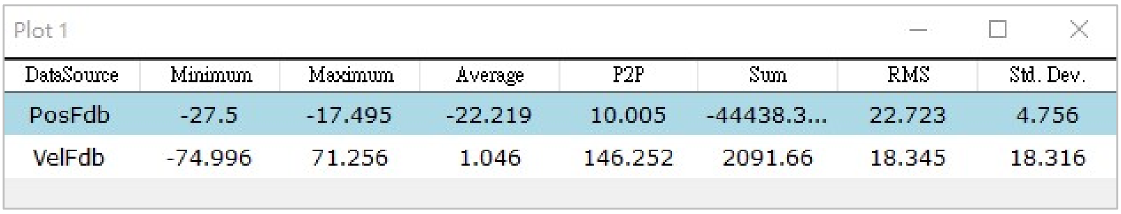
  Choose which data source to be the X axis.
- **Fade Effect:**

  When ticked, the curves shown on the display panel will fade out.
- **Fixed Ratio to 1:1**

  Fix the plane ratio to be 1:1.

d. Statistics:

Shows a variety of motion data, such as maximum and minimum values, peak to peak, root mean square, etc.

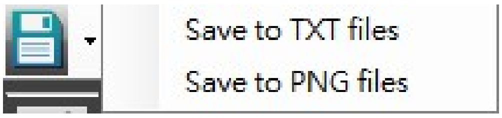


DataSource	Minimum	Maximum	Average	P2P	Sum	RMS	Std. Dev.
PosFdb	-27.5	-17.495	-22.219	10.005	-44438.3...	22.723	4.756
VelFdb	-74.996	71.256	1.046	146.252	2091.66	18.345	18.316

When the "marks A and B" are applied (see chapter 8.4.3, point e.), the statistics panel will show the various values that correspond to the marks A and B.

8.4.2 Save to Files

Choose to save the monitored data in TXT. or PNG. format.

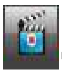


8.4.3 Scope Tools

a. Turn Scope ON/OFF:

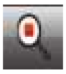
To switch the scope function ON/OFF. When function is off, the oscilloscope graph will be frozen.

➔ status: ON.

➔ status: OFF.

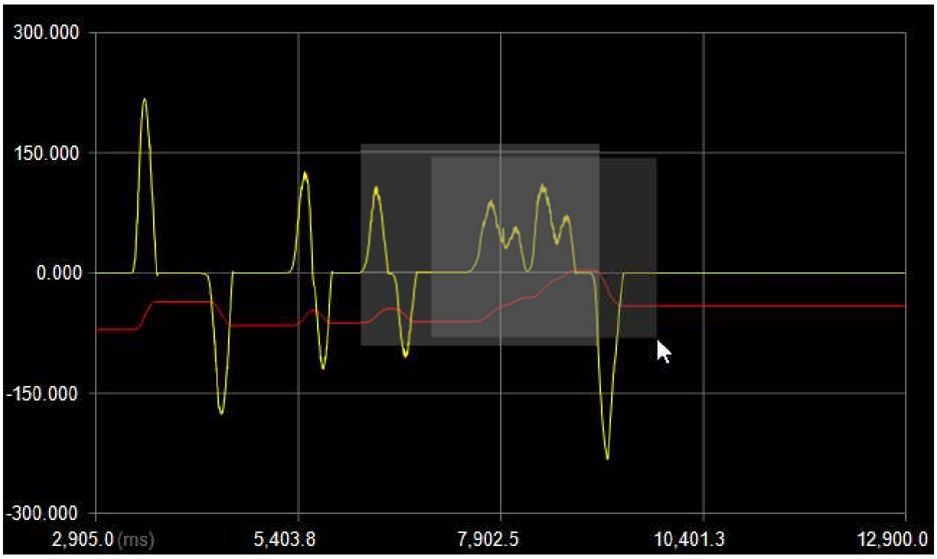
b. Zoom ON/OFF:

 ➔ status: ON.

 ➔ status: OFF.

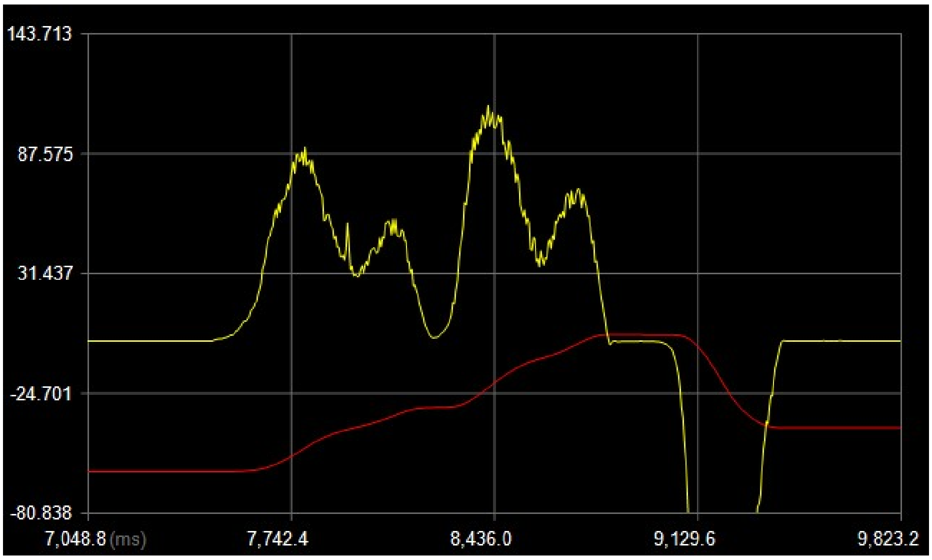
<u>Steps</u>:

1. Click the icon to activate the Zoom.
2. Hold the left mouse button and drag a box around the area of interest.



(↑ The selected area will appear in dark grey)

3. Release mouse button to let the display zoom in the selected area.



(↑ zoomed in)

c. Zoom Reset:

 Reset to the original graph.

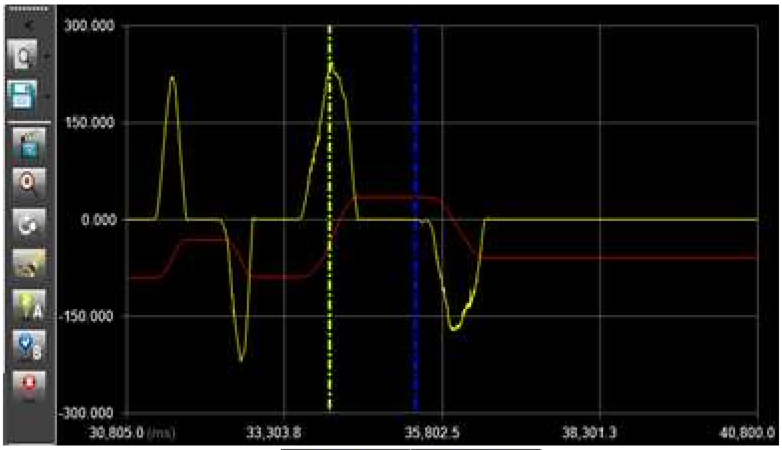
d. Clear Plot:

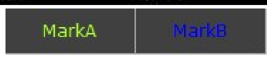
 Clear the display panel.

e. Mark A/B:

Click on the   icons to put marks A and B on the display panel.

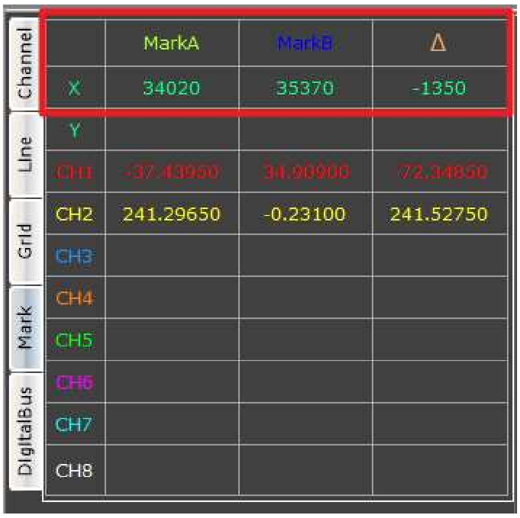
Drag to position the



The information 

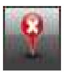
and differences 

between marks A and B are shown on the “Mark Tab” of Further Setting Panel.



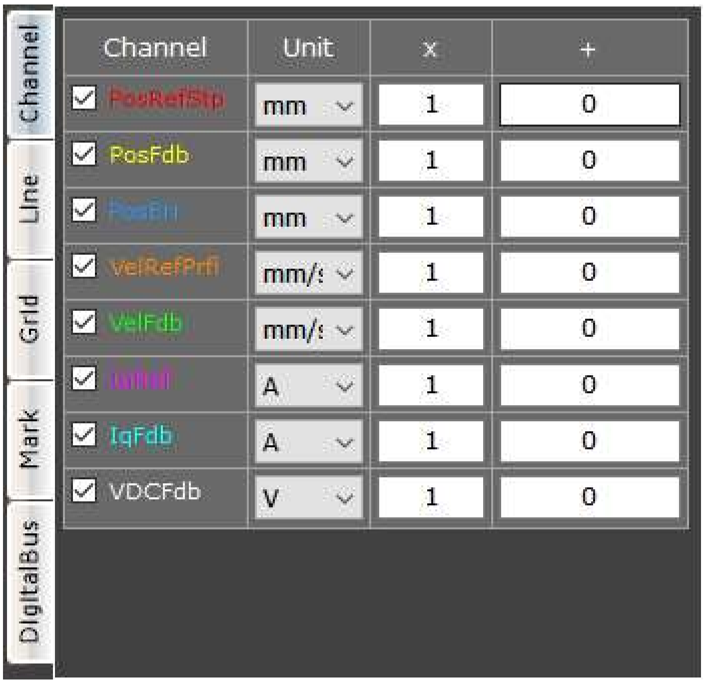
	MarkA	MarkB	Δ
X	34020	35370	-1350
Y			
CH1	-37.43960	34.90900	-72.34850
CH2	241.29650	-0.23100	241.52750
CH3			
CH4			
CH5			
CH6			
CH7			
CH8			

f. Clear Marks:

 Clear all marks on the display panel.

8.5 Further Setting Panel

This panel consisting of 5 tabs is to further configurate the graphs shown on the display panel; besides, the information of the marks A and B is shown on the Mark tab.



8.5.1 Channel Tab & Line Tab

Users can show or hide channels, change the color, unit or width of the line(s), and adjust the values (by multiplication x and/or addition +) to view the scope better.

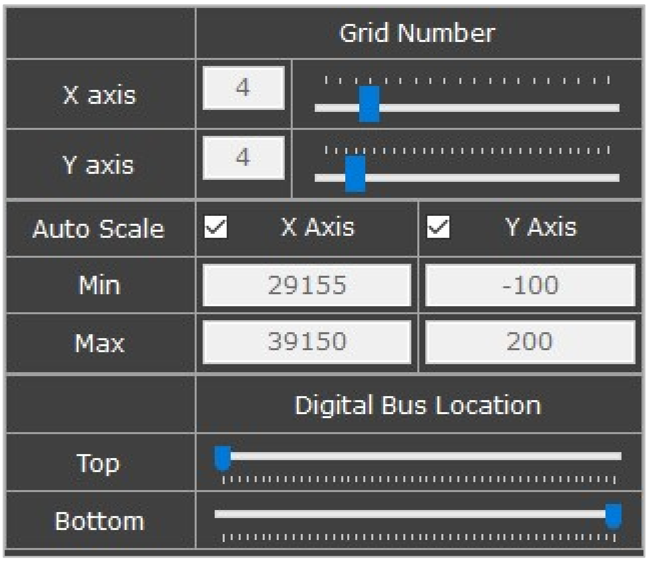
Channel Tab

Channel	Unit	x	+
☑ PosRefStp	mm	1	0
☑ PosFdb	mm	1	0
☑ PosErr	mm	1	0
☑ VelRefPrfl	mm/s	1	0
☑ VelFdb	mm/s	1	0
☑ IqRef	A	1	0
☑ IqFdb	A	1	0
☑ VDCFdb	V	1	0

Line Tab

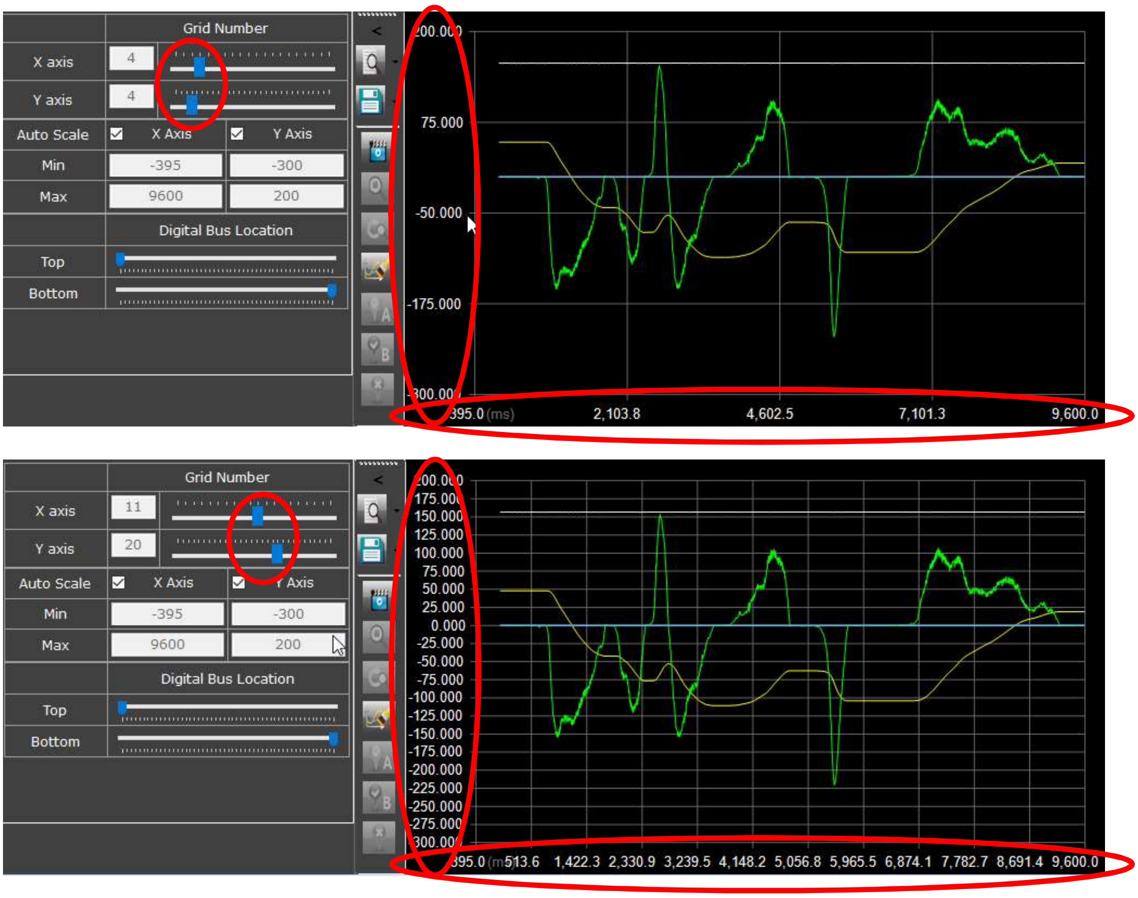
Channel	Color	Width
PosRefStpMg		1
PosFdb		1
PosErr		1
VelRefPrfl		1
VelFdb		1
IqRef		1
IqFdb		1
VDCFdb		1

8.5.2 Grid Tab



a. X/Y Axis:

To set how many grids to be shown on the X and Y axes.



b. Auto Scale:

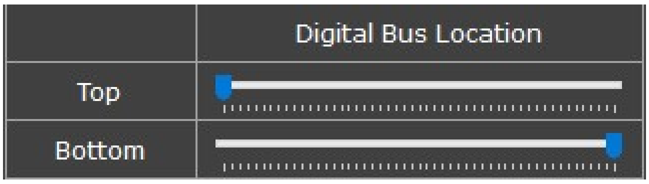
Automatically adjust the minimum and maximum limit of the X and Y axes.

c. Min/Max:

Customize the minimum and maximum limit of the X and Y axes.

d. Digital Bus Location:

Adjust the position of digital bus graphs shown on the display panel.



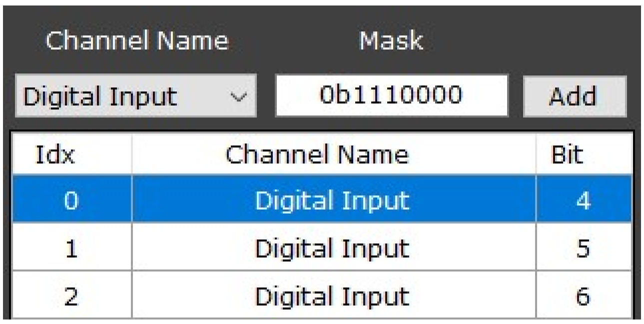
Example:

8.5.3 Mark Tab

Mark tab shows the values and differences (△) of the marks A and B.

8.5.4 Digital Bus Tab

The Digital Bus function is for observing the activity of certain **bits**. Users can also mask the bits which are not needed.



- **Channel Name:**

  Select the data source to be processed.

- **Mask:**

  Key in the binary matrix value of the bits to be observed.

- **Add:**

  Click to include these bits into the table of interest.

- **Idx:**

  The serial number for the bits of interest listed on this table.

- **Bit:**

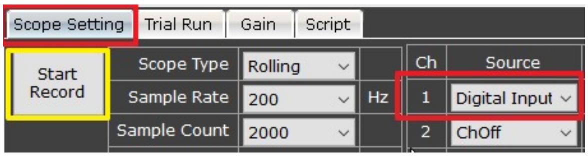
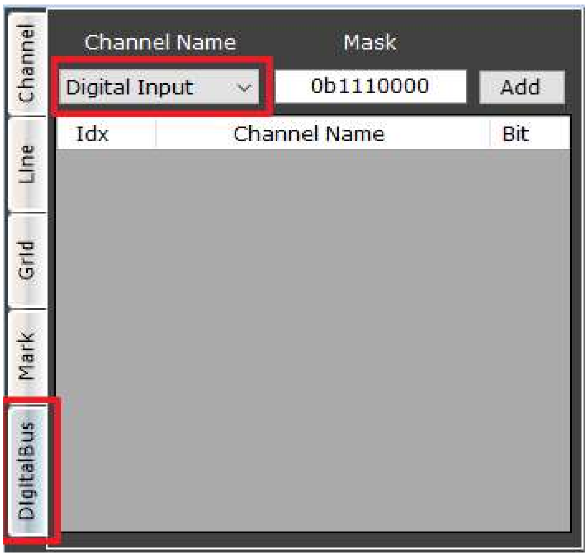
  Bit number.

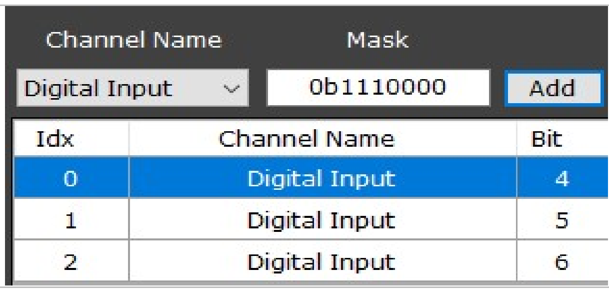
- **(right click) Delete / Clear:**

  Delete certain bits or clear all the bits of interest.

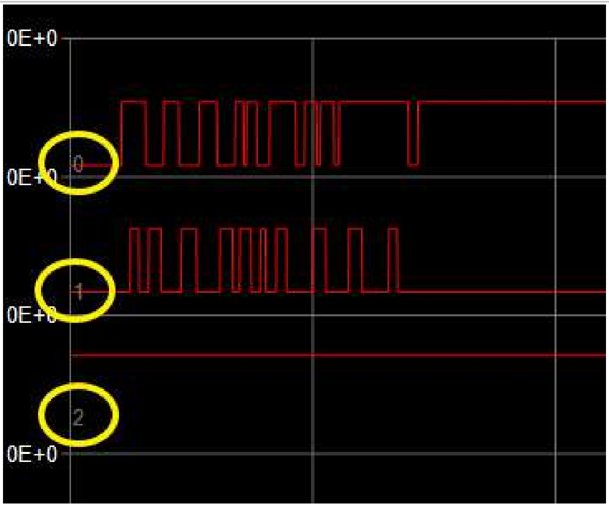
8.5.4.1 Configuration Steps

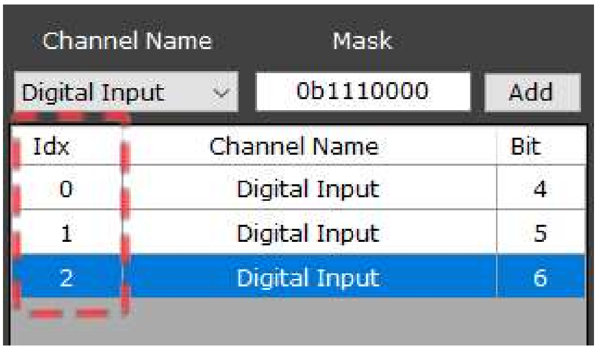
Let's take observing the **bit 4, 5, and 6 of digital input** for example:

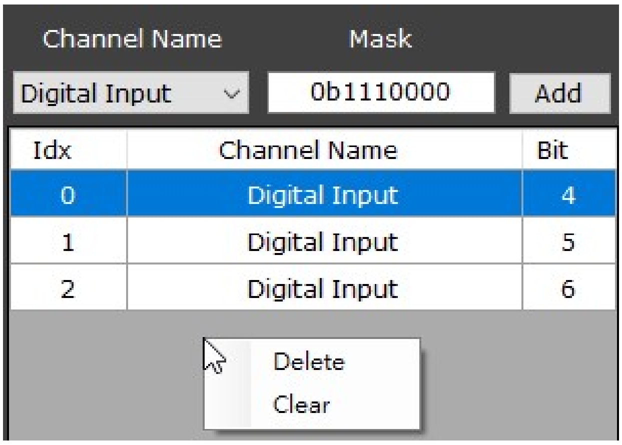
Step	Description	Example
1	Go to the Scope Setting tab on the [Scope Setting|Trial Run|Gain|Script] Panel. **Configure the data source of interest** (e.g. digital input) and then **click "Start Record**".	
2	Go to the DigitalBus tab on the Further Setting panel. **Select the data source** (choose from the Channel Name list).	
3	**Decide which bit(s) to show.** **Convert the bit's numeric figure into binary matrix value**. Next, key in this value into the "Mask" textbox and press enter; then click "Add". (note: "0b" means binary)	**In this case,** **to observe the bits 4, 5, and 6 ➔** (1) 4, 5, 6 equals to binary matrix value "1110000". (2) Key in 1110000 into the "Mask" textbox and press enter. (3) Click "Add".

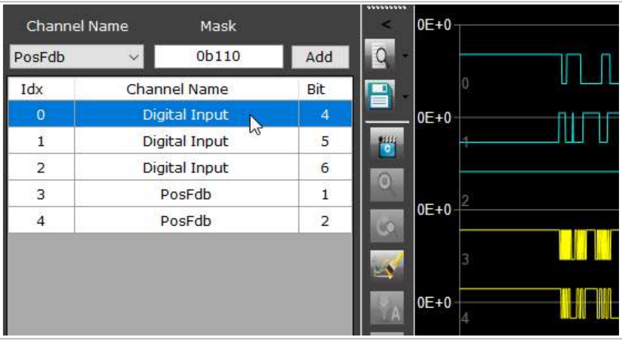
<table>
<tr>
<td></td>
<td></td>
<td>
</td>
</tr>
<tr>
<td>4</td>
<td>Now, observe the scope.

On the display panel, the numbers in dark grey color are the Idx number.

<u>For example:</u>
Idx #0 is bit 4,
Idx #1 is bit 5,
Idx #2 is bit 6.</td>
<td>


</td>
</tr>
<tr>
<td>5</td>
<td>To delete a specific line, choose the channel bit and right click, then select “Delete”.

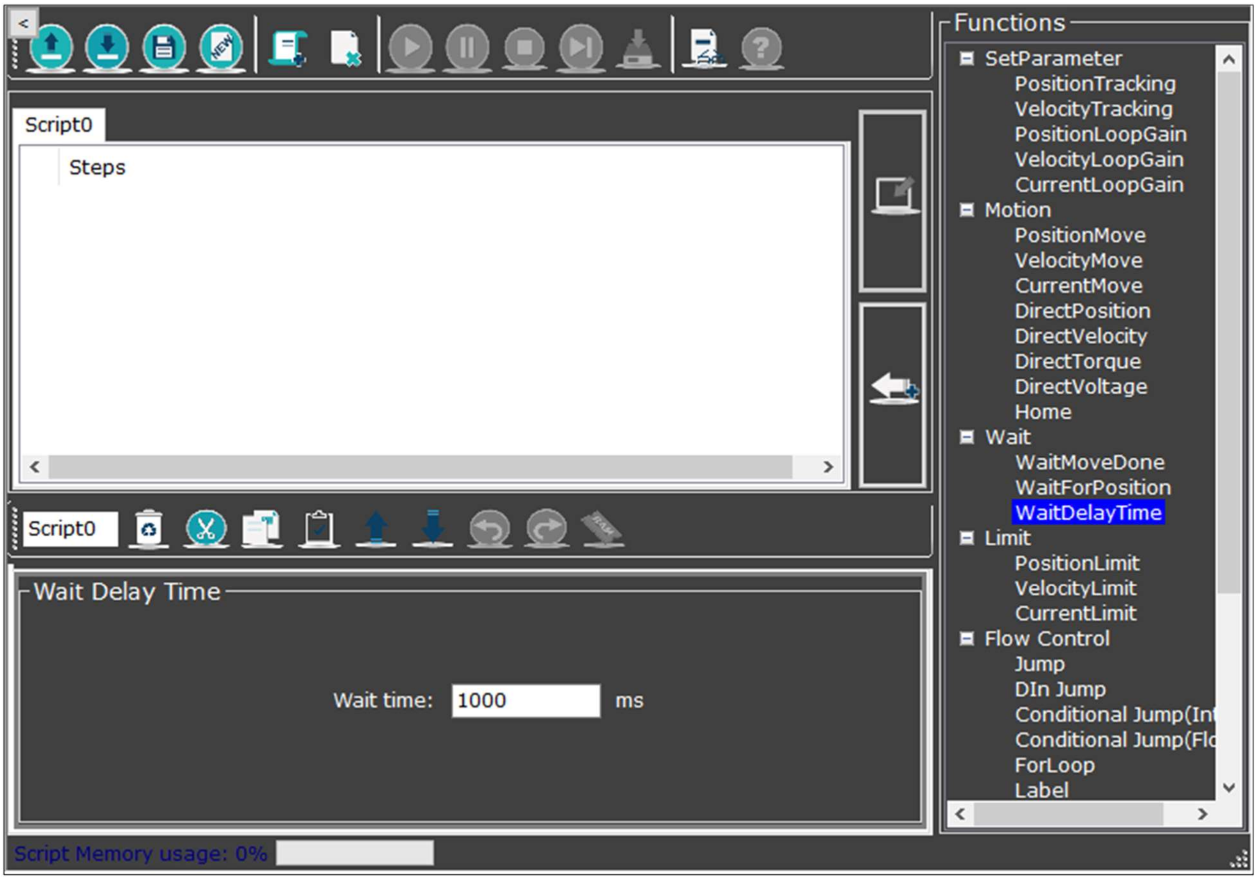
To clear all, right click on the table and select “Clear”.</td>
<td>
</td>
</tr>
</table>

6	Or, you can add more new channels by selecting other data source and then adding them into the table.	

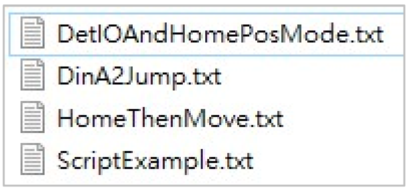
Chapter 9 Script

9.1 Intro and Interface Tour

Script can be seen as a state virtual machine; besides, script can simulate a series of motion commands. In application, script is usually used to test the performance of the driver executing or demonstrating a series of motion profiles.



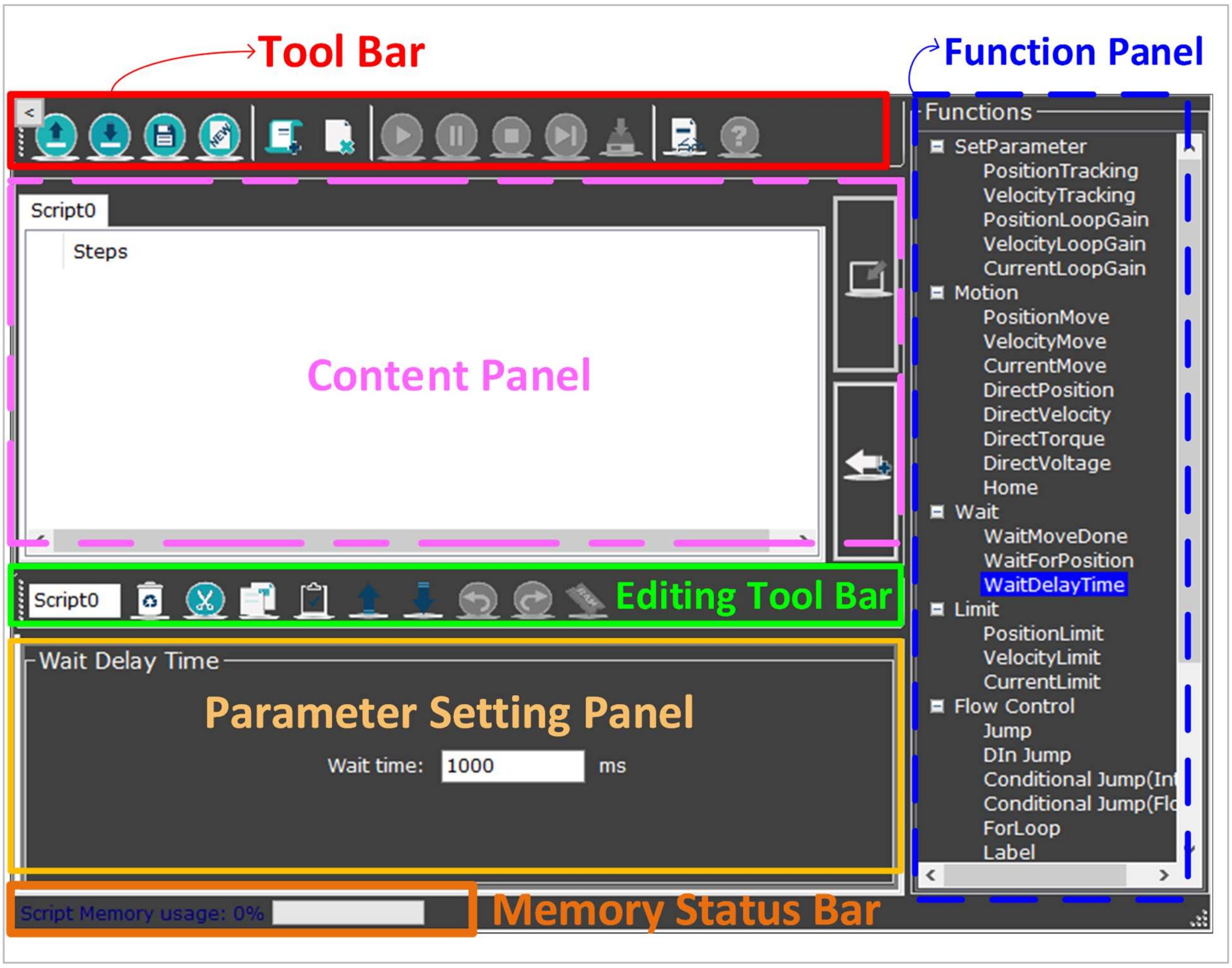
We cpc provide 4 script examples which can be found in the installation file.



On the subsequent pages you will see how to import script file(s), modify parameters, and add functions into a script.

Scripts can run automatically after the boot sequence setting is done. Please refer to **Chapter 4.8** Boot Sequence.

The Script interface has 6 areas:



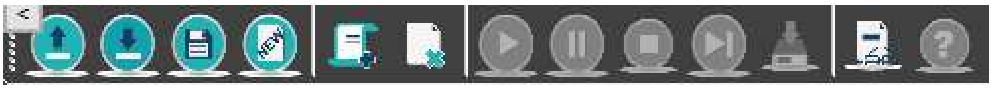
1. **Tool Bar** – on top of the interface, for processing file and executing script.
2. **Content Panel** – showing the editing result.
3. **Editing Tool Bar** – tools for editing.
4. **Function Panel** – a menu to navigate the various functions.
5. **Parameter Setting Panel** – to configure parameters of each function.
6. **Memory Status Bar** – showing the usage (%) of flash memory.

9.2 Interface Elements

1.

Tool Bar

For processing file and executing script:



Export

Exports script file in txt. format.

Import

Imports existing script file.

Save

Saves script.

New

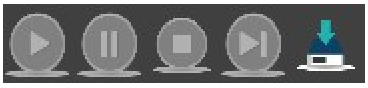
Opens a new blank file.

Note:

If you are editing a file, clicking this "New" button will clear up all the content. **Please make sure you've saved script files beforehand.**

~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~

For performing the script:



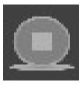
See **chapter 9.6** for the script execution flow.

Run

Runs each step of the script continuously.

Pause

Pauses the script. Click "Run" again to resume from where you paused at.



Stop

Stops the script. Click "Run" again to resume from the first step of script.

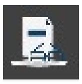
Step by Step

Executes only one step at a time and then pause. Click this button again to execute the next line.

Compiler

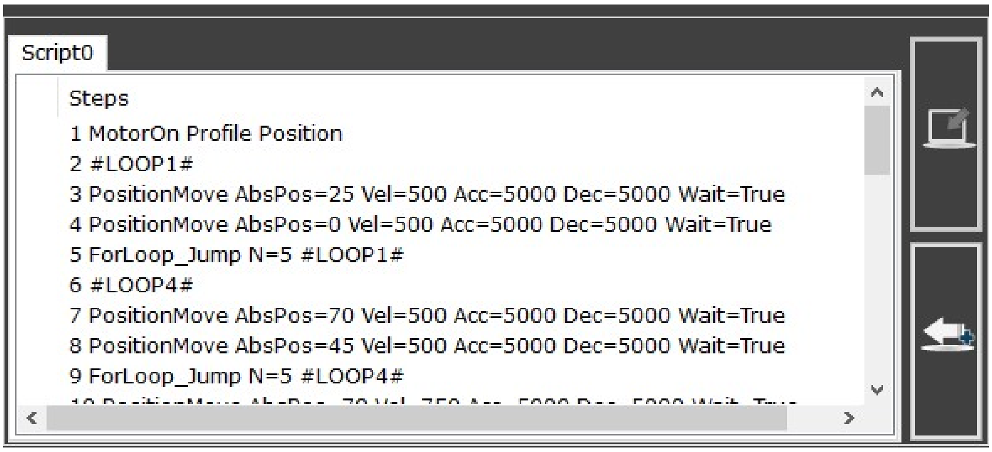
Click to convert the script into a machine-code.

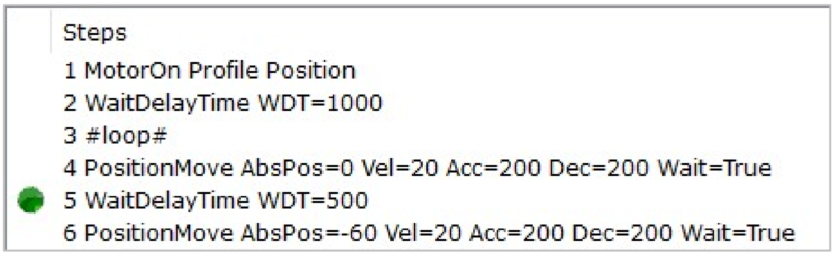
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~



Register Viewer

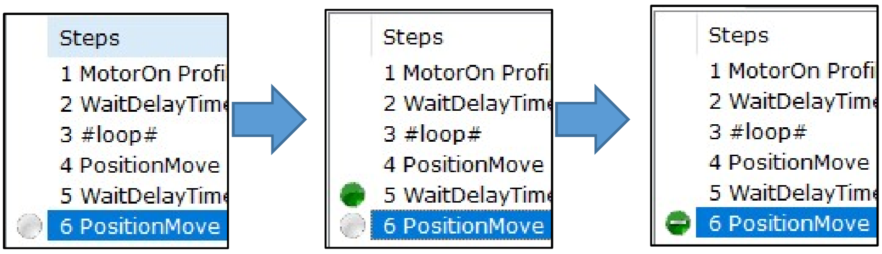
A table for monitoring the value of integer variables and float-point variables.

	 **Load FLASH Code** To load the script from the driver, it will overwrite the currently-edited script.
2. Content Panel	**Shows the editing result:**   **Insert selected function** Inserts the selected function prior to the chosen step.  **Add selected function to final step** Appends the selected function to the final step of script.  **Green Dot** When running the script, this green dot indicates the next step to be executed.



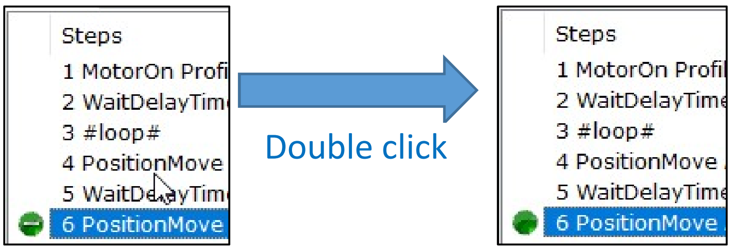
 **Grey Dot** (break point) **/**  **Green-stop Dot**

- Before running the script, users can double click on a selected step to set a <u>break point</u>, a <u>grey dot</u> will then appear in front of the selected step.
- A break point (grey dot) will make the execution pause at the end of the previous step.
- While running the script, a grey dot will become a green-stop dot indicating that the execution is paused at the previous step and that the step with the green-stop dot is <u>not</u> performed.



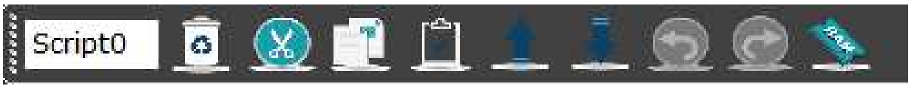
Before running Running Execution paused

To remove a break point, just double click the selected step.



3. Editing Tool Bar

Tools for editing:



Renames the script, key in the new name in the box.

 **Delete**

Deletes the selected step(s) in the script.

 **Cut**

Cuts the selected step(s).

 **Copy**

Copies the selected step(s).

 **Paste**

Pastes the copied step(s) before the selected step(s).

 **Move up**

Moves the selected step(s) up.

 **Move down**

Moves the selected step(s) down.

 **Undo**

Undo the last move.

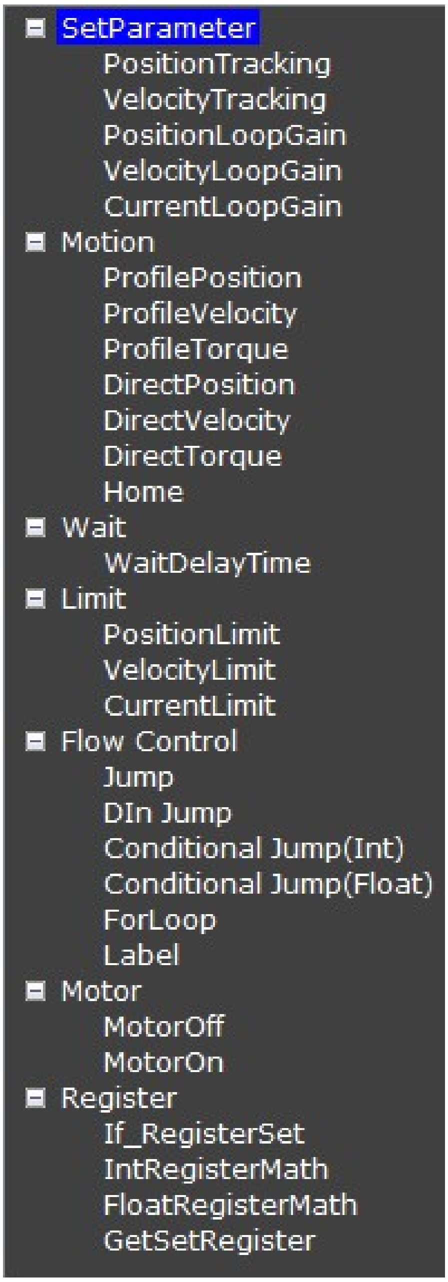
 **Cancel Undo**

 **Memory**

Calculates how much percentage (%) of the Flash the script will occupy.

4. Function Panel

A menu to navigate the various functions.

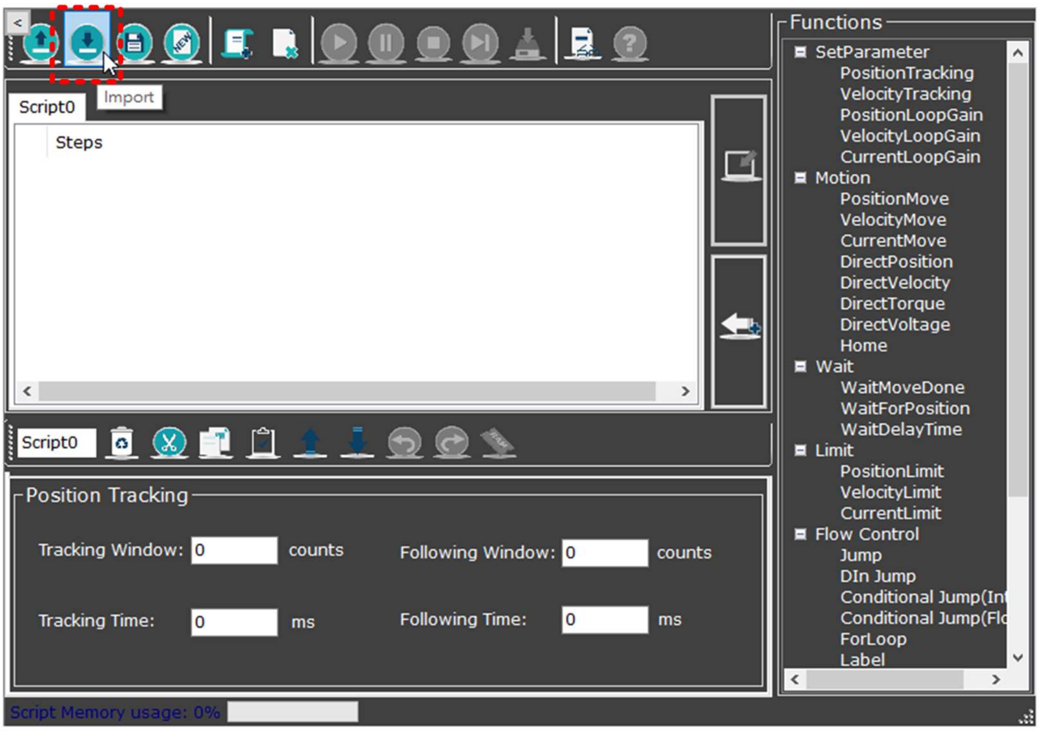


5. Parameter Setting Panel	To further configure parameters under each function (shown on the function panel).
6. Memory Status Bar	Showing the usage percentage of flash memory. 

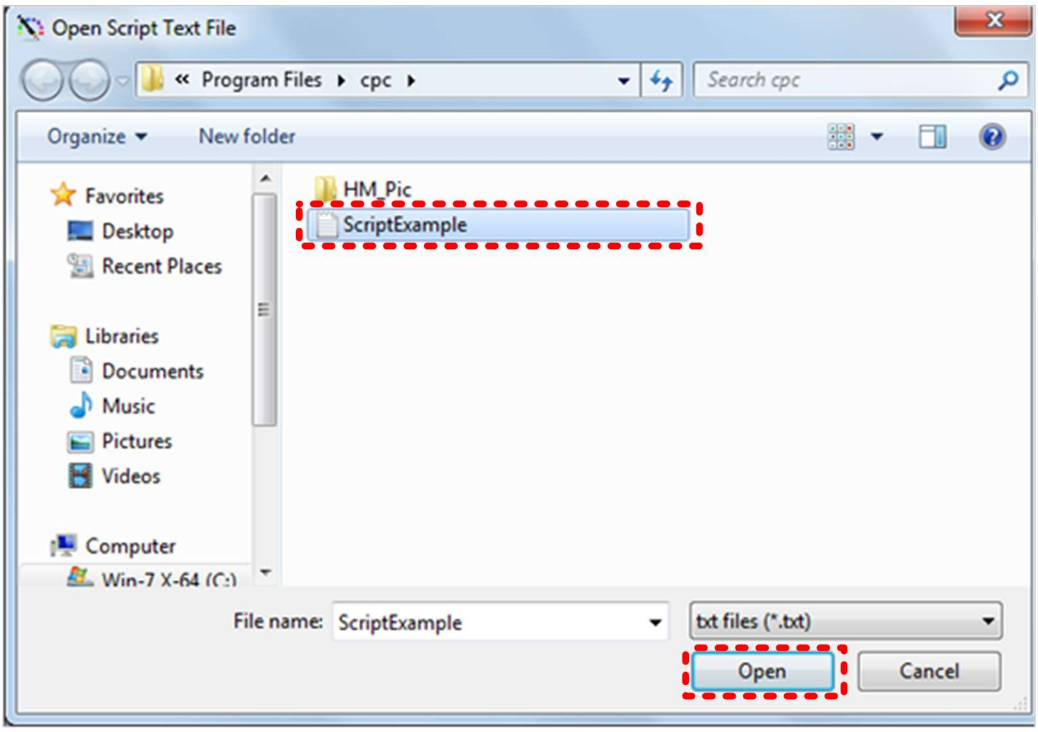
9.3 Import Scripts from File

Steps:

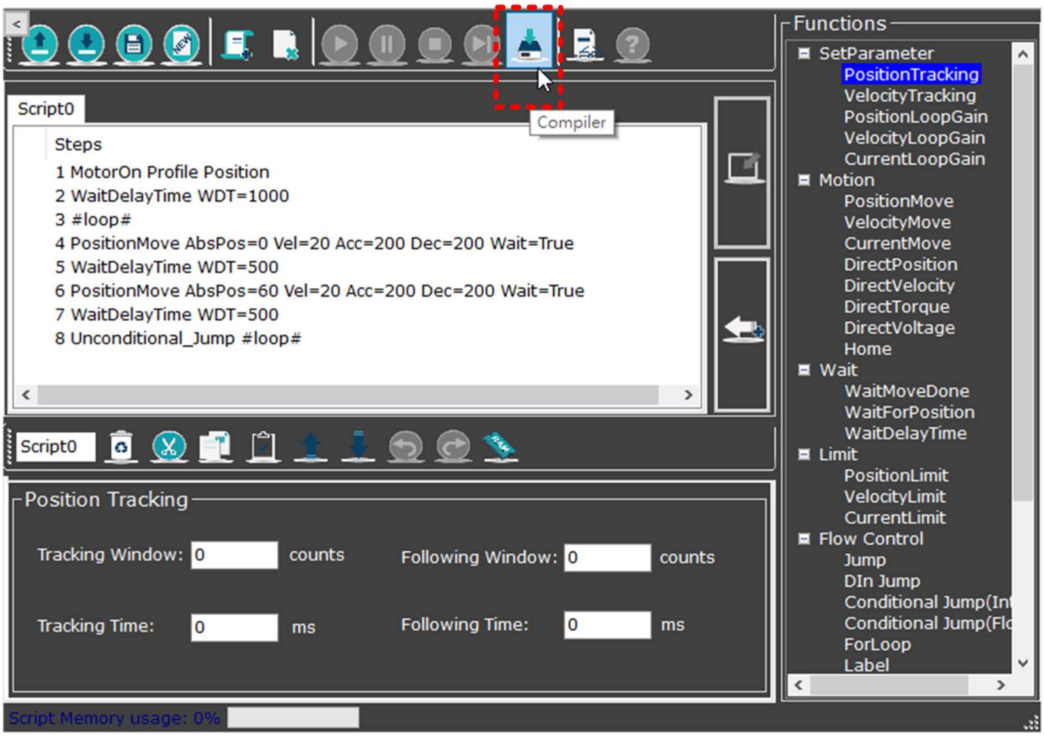
1. Click "Import".

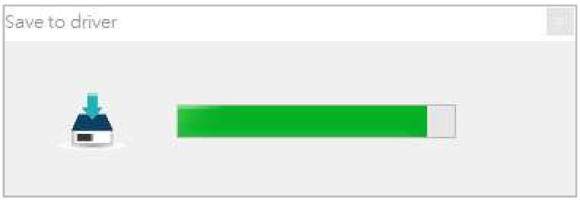


2. Locate your file and click "open".

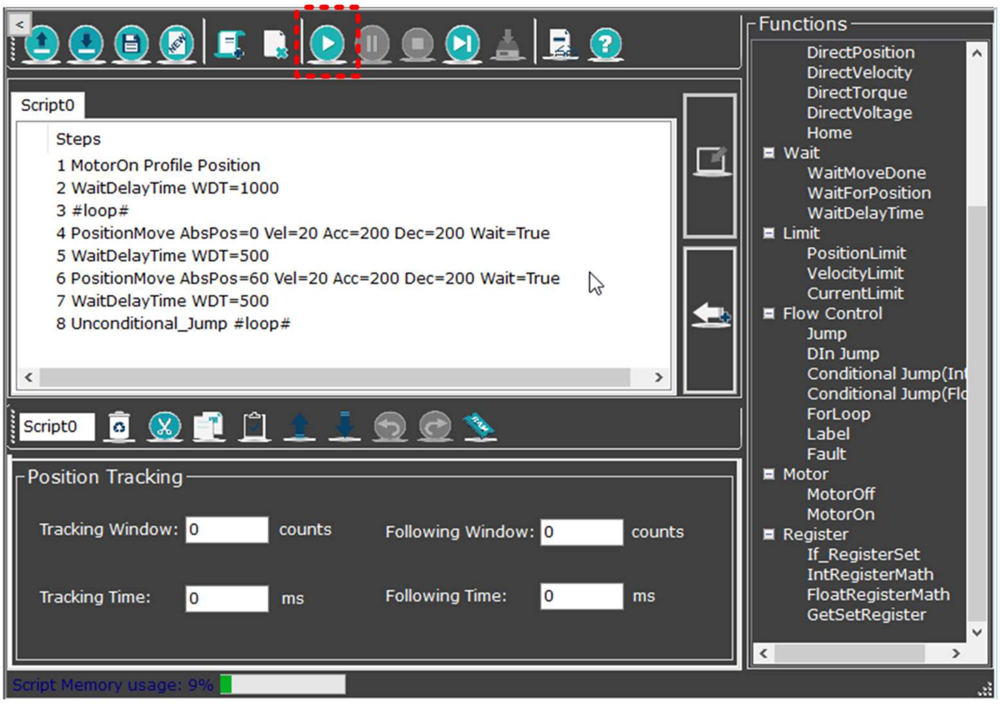


3. Click “Compiler” and wait for the process to finish.





4. After compiling is finished, click “Run” or “Step by Step” button to run script.

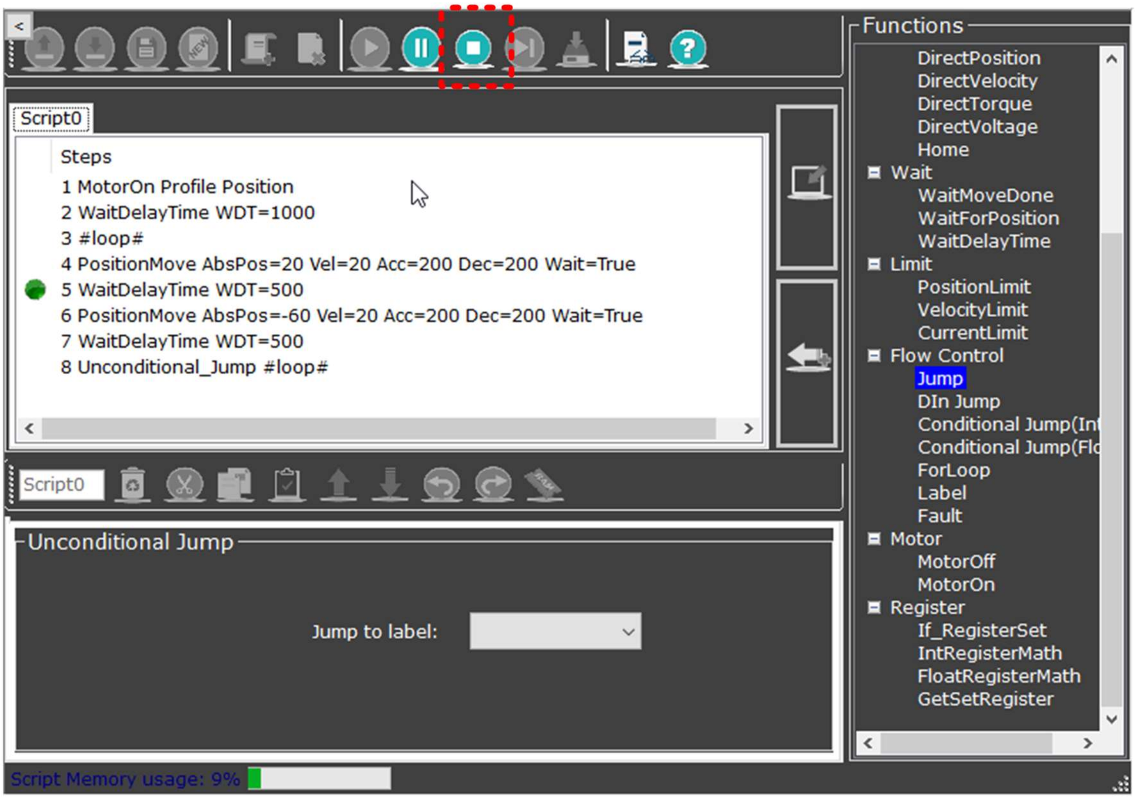


9.4 Modify Script Parameter

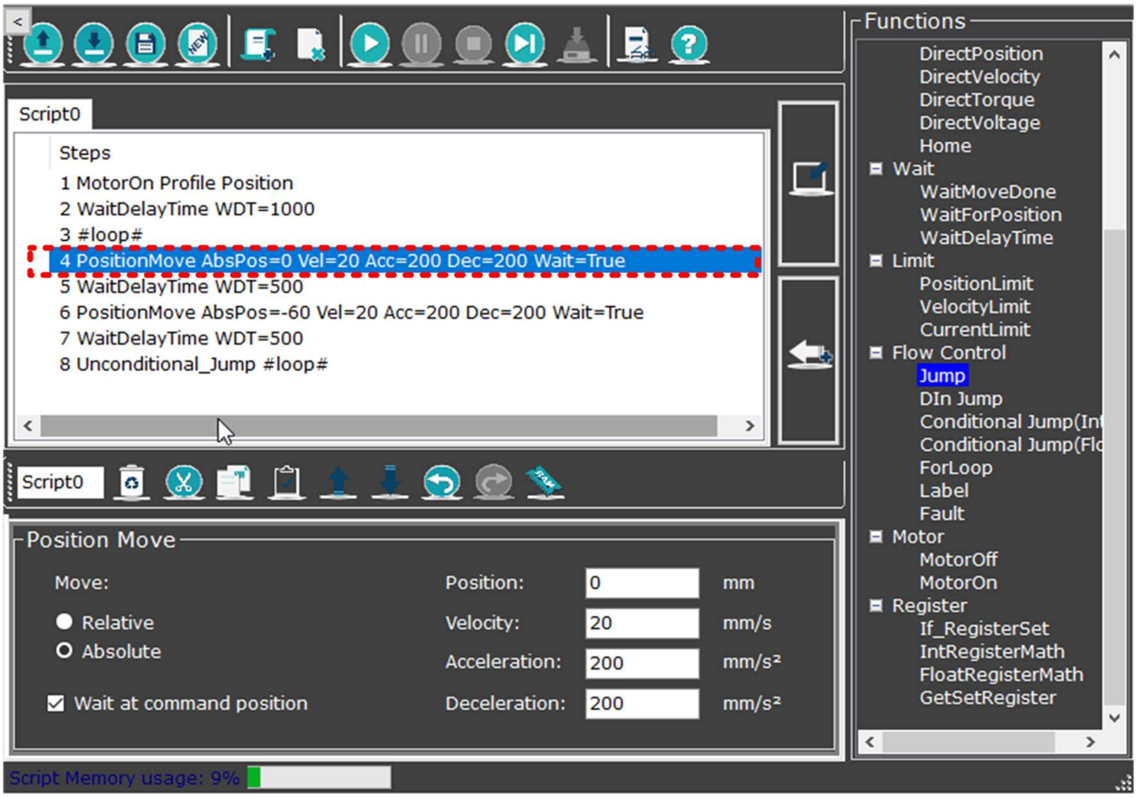
To modify script parameters, users must "stop" the script execution first.
When script is stopped, users can then modify parameters and compile them again.

Steps:

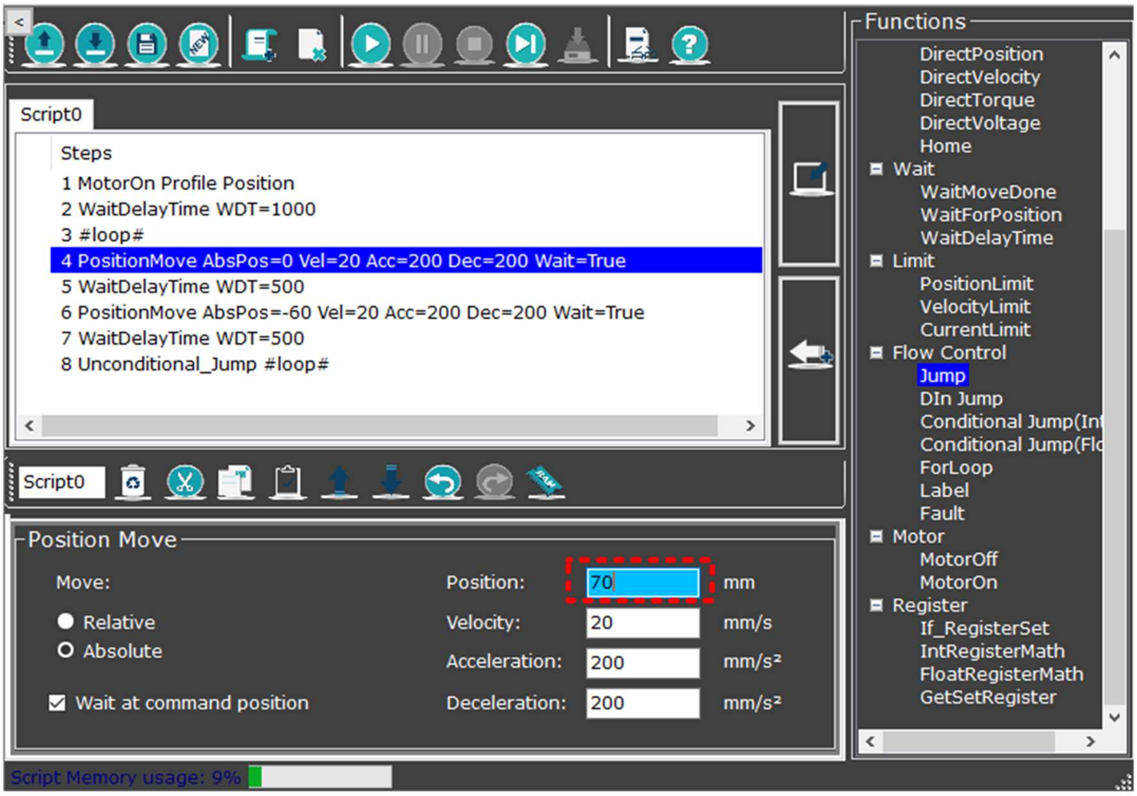
1. Click "Stop".



2. Click the step (line) that you wish to modify.



3. Modify the value on the Parameter Setting panel and then **press Enter.**



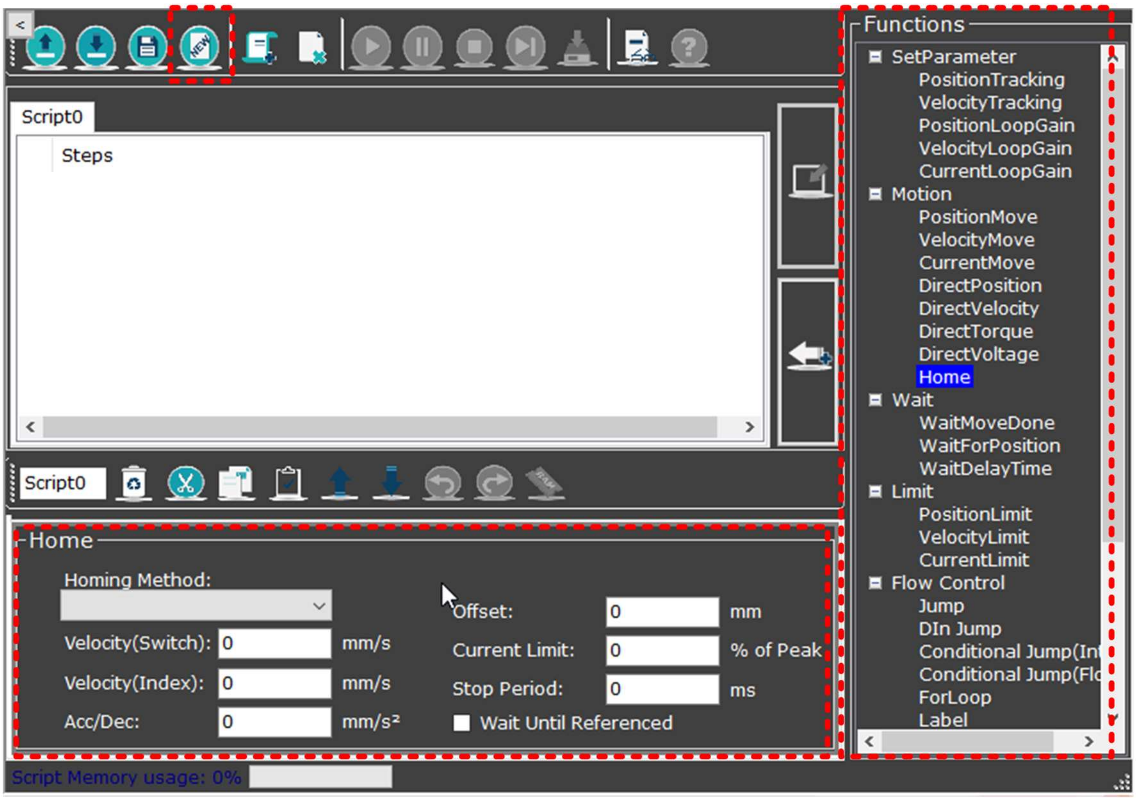
4. Click “**Compiler**” to compile the script again to be executed.

9.5 Add a Function

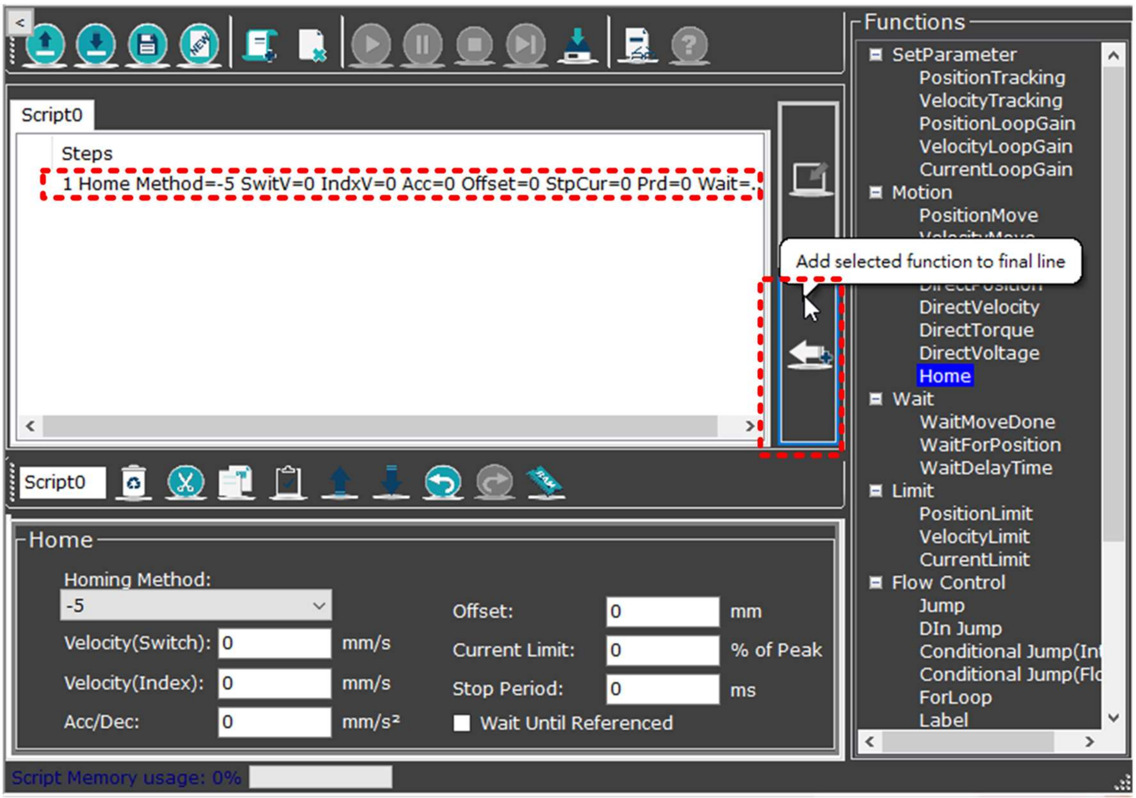
9.5.1 Into a New File

Steps:

1. Click “New”  button
2. Select a function from the function panel.
3. Go to the parameter setting panel. Modify the value or select a preferred action from the list.



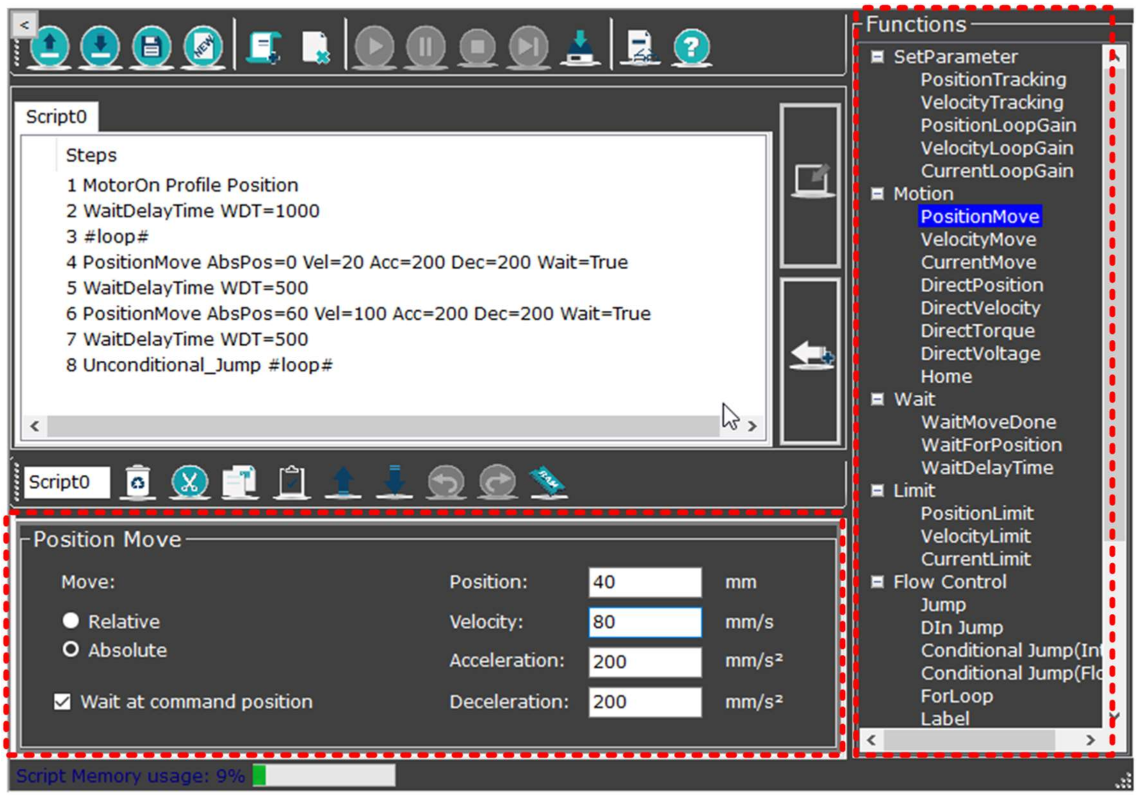
4. Click “Add” button, then the selected new function will appear on the content panel. Now you have successfully added a new function.



9.5.2 Into a Currently-Used File

Steps:

1. Select a function from the function panel.
2. Go to the parameter setting panel. Modify the value or select a preferred action from the list.

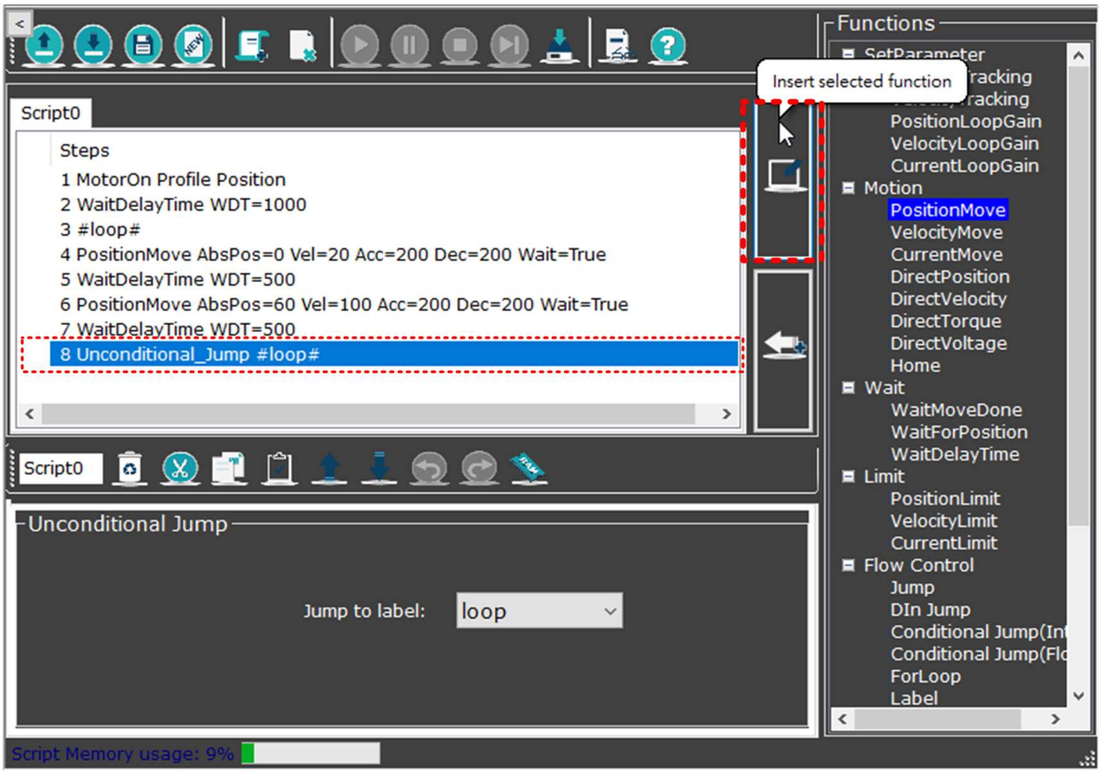


3. Go to the content panel, click on the line before which you wish to insert the new function, then click "Insert".

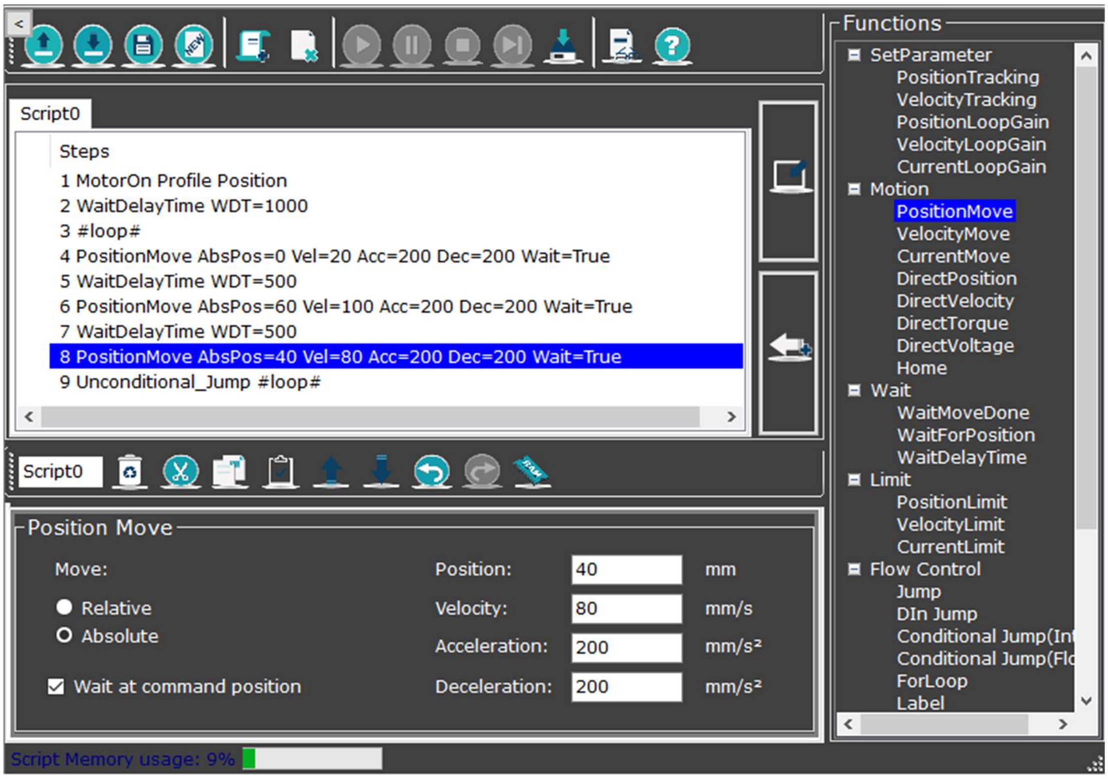
For example:

If you wish to insert a PositionMove **after step #7**, you will need to click step **#8** and Insert, so the new function will be placed after step #7 and before step #8. See images below.

(↓ To insert after step #7, you will need to click step **#8** and Insert)

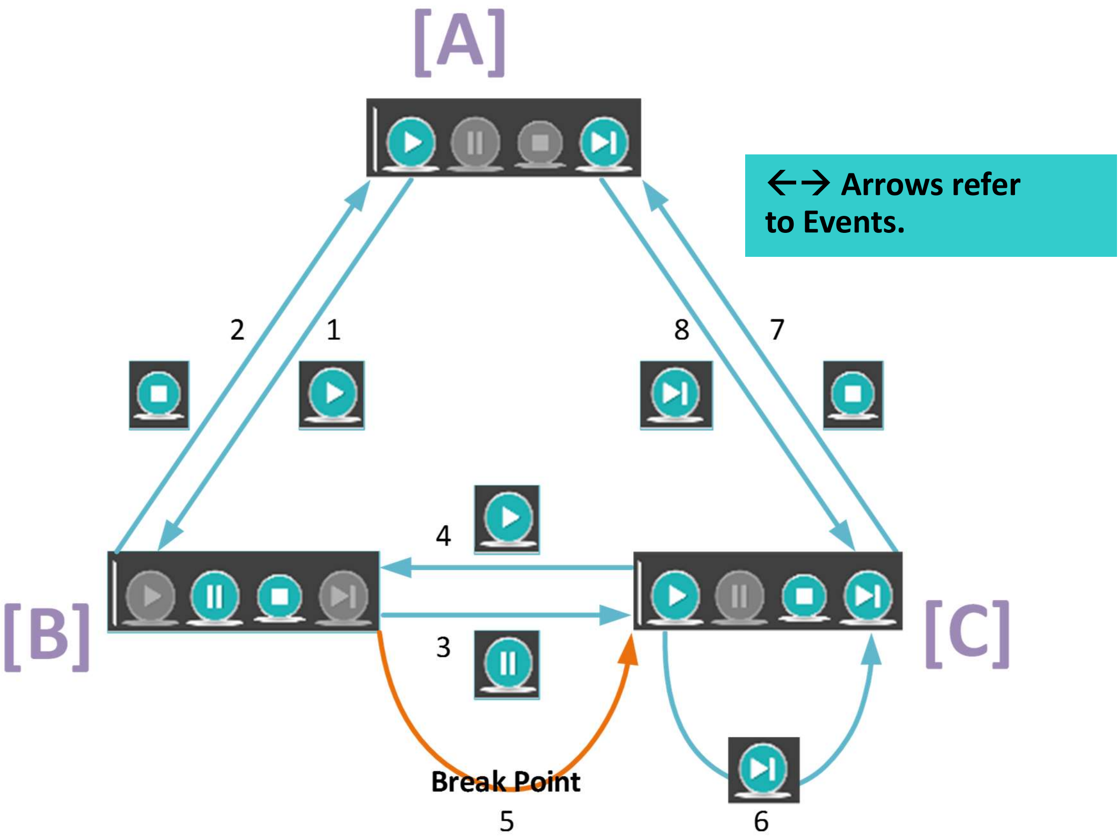


(↓ The newly-added function will be sequenced after step #7)



9.6 Run the Script

After the script is compiled, the tool bar will present the "Run"  and "Step by Step"  icons in green. The chart below shows the states and panel appearance while running the script.



- State [A]:

  After scripts are compiled, the panel will look as state [A].

- Arrow #5:

  If there is break point(s) set, the script execution will be paused and will look as state [C].

- Arrow #6:

  If users click the "Step by step" button, the panel will look the same.

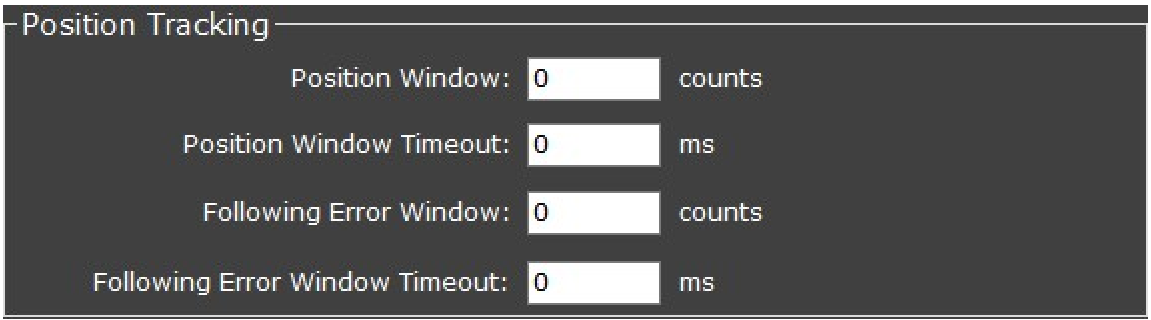
Note: Scripts must be compiled to be executed.

9.7 Function Description

Regarding the functions mentioned on subsequent pages, if users set parameters using IR (integer register) or FR (float number register), please note that **the unit of IR and FR is "count" instead of the descriptions shown on the UI panel**.

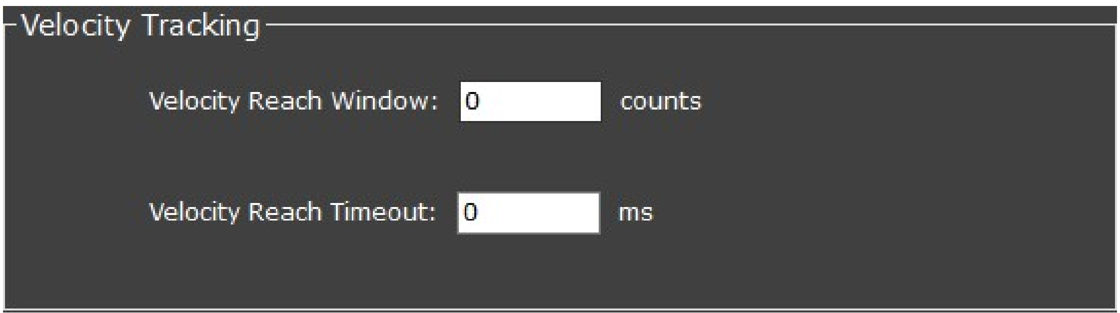
9.7.1 Set Parameter

- **Position Tracking**



Please refer to **chapter 4.3.3** Motor Protection—Position for parameter details.

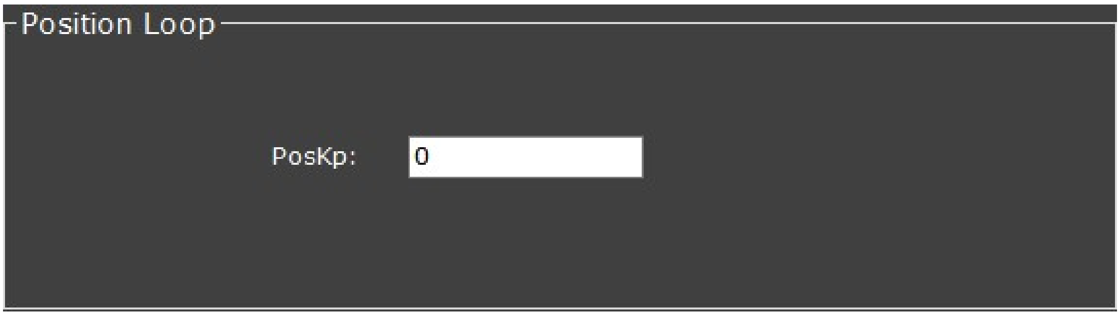
- **Velocity Tracking**



a. **Velocity reach window / timeout:**

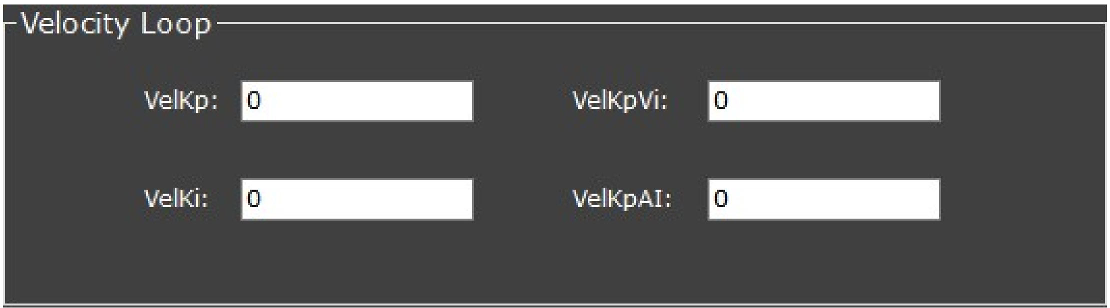
When the velocity feedback is within the reach window and continues for a period (reach timeout), the velocity target reach flag will rise.

- **Position Loop Gain**



Please see **chapter 5.4** Tune—Position for definition.

- **Velocity Loop Gain**



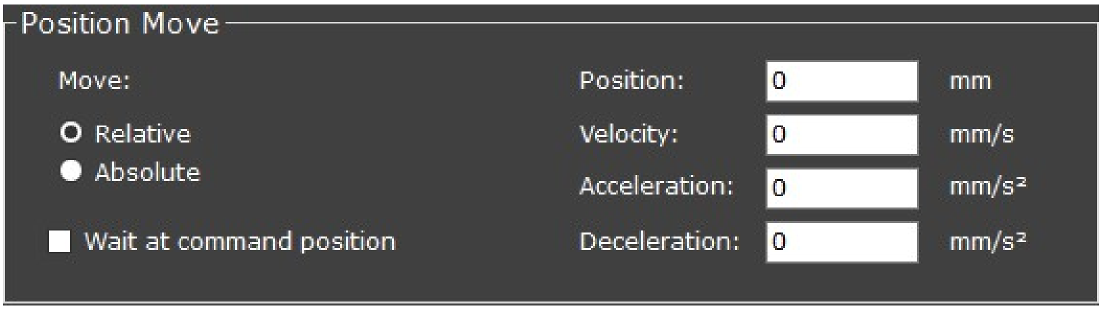
Please see **chapter 5.3.1** for further parameter information.

- **Current Loop Gain**

Please see **chapter 5.1.1** for further parameter information.

9.7.2 Motion

- **Position Move**



a. **Move**

Tick to give relative or absolute position command.

b. **Wait at command position**

- When ticked, the driver will wait for the target to be reached before executing the next command.
- When unticked, the driver will execute the newly received command right away (won't wait for target reach).

  <u>**Note**</u>:

  If this box is unticked, it might cause following error.

c. **Position**

Target position.

d. **Velocity**

Profile velocity of movement.

e. **Acceleration**

Profile acceleration.

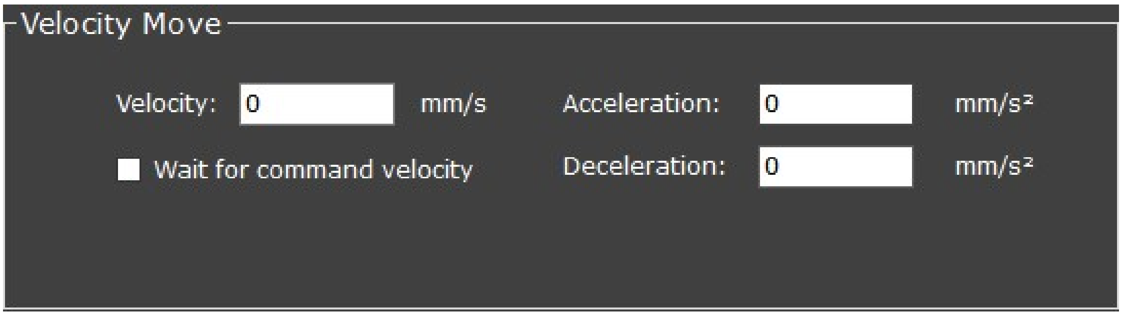
f. **Deceleration**

Profile deceleration.

<u>**Note**</u>**:**

The columns of Position, Velocity, Acceleration, and Deceleration can be filled in with **number or the name of IR** (Integer Register). The IR unit is "count".

- **Velocity Move**



a. Velocity

Value of profile velocity.

b. Wait for command velocity

- When ticked, the driver will reach the present command's target first before executing the next command.
- When unticked, the driver will execute the newly received command right away without fully finishing the current command.

c. Acceleration

Profile acceleration. The maximum acceleration speed during movements.

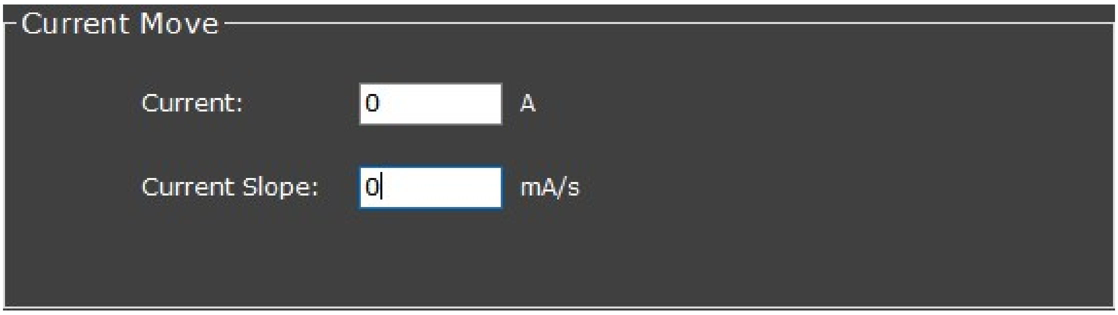
d. Deceleration

Profile Deceleration. The maximum deceleration speed during movements.

Note:

The columns of Velocity, Acceleration, and Deceleration can be filled in with number or the name if IR (Integer Register). The IR unit is "count".

- **Current Move**



Note:

The columns of current and current slope only accept either numbers or names FR register.

a. Current

Profile current. Current unit in Amperes.

b. Current slope

The rate of current increase, unit in mA per second.

Note:

This column can only accept either FR or numbers greater or equal to 1.

- **Direct Position**

a. Position

The value of the designated **absolute** position.

Note:

This column can be filled in with number or the name of IR (Integer Register). The IR unit is "count".

- **Direct Velocity**
  a. **Velocity**

     The value of the designated velocity.

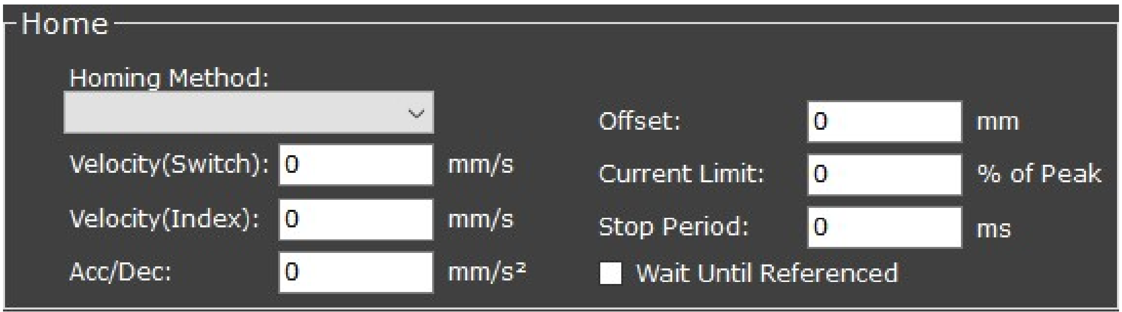
     Note:
     This column can be filled in with number or the name of IR (Integer Register). The IR unit is “count”.

- **Direct Torque**
  a. **Iq**

     The current (Amperes) on the “q” (quadrature) axis in the (d,q) coordinate system.

     Note:
     This column can be filled in with number or the name of FR (Float number Register). Unit is Amperes.

- **Home**



  a. **Homing Method**

     Select a preferred homing mode from the list.
  b. **Velocity (Switch / Index)**

     The speed to move to the switch/index.
  c. **Acc/Dec**

     Acceleration/deceleration speed.
  d. **Offset**

     The offset millimeter (mm) from origin. Moreover, the offset value will be the position mm when a homing is completed.

e. Current Limit / Stop Period

When the driver continues to output more than or equal to a certain percentage of peak current (Current Limit) for a period of time (Stop Period), the motor will be regarded as hitting a hard stop.

9.7.3 Wait

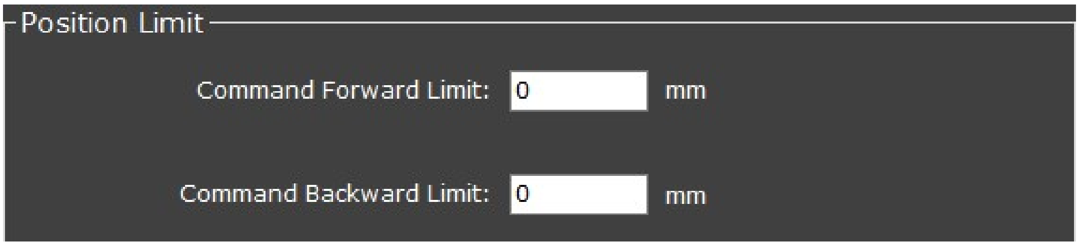
- **Wait Delay Time**

  Wait for some time (ms) first before running the next step of script.

9.7.4 Limit

- **Position Limit**

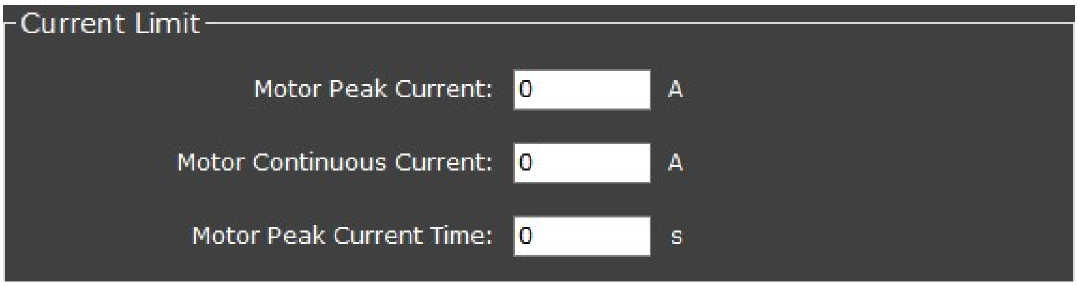
  Same as the parameter "Command Forward/Backward Limit" in the section of "motor protection—position".



- **Velocity Limit**

  The motor rated speed in mm/s.

- **Current Limit**



Please see **chapter 4.3.1** for further parameter information.

9.7.5 Flow Control

- **Jump**

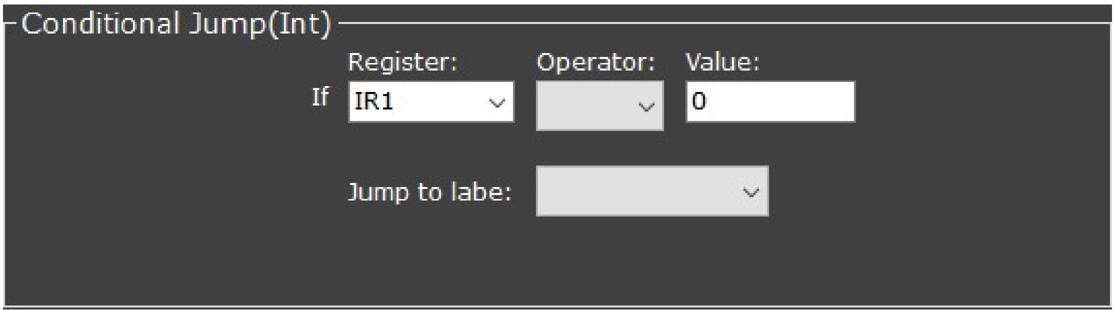
  Unconditionally jump to a selected label in script, no conditions required.

- **DIn Jump**

  Jump to a selected label depending on the status (Low/High) of digital in.

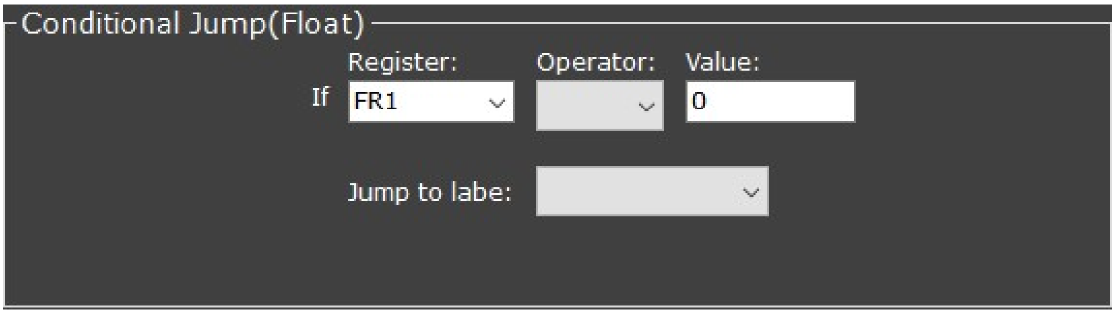
- **Conditional Jump (Int)**

  Conditional Jump (Integer)



  - ➢ Jump to a selected label when the value of integer register satisfies the configured condition.
  - ➢ The column of “Value” can be filled in with number or the name of IR (Integer Register).

- **Conditional Jump (Float)**



  - ➢ Jump to a selected label when the value of float number register satisfies the configured condition.
  - ➢ The column of “Value” can be filled in with number or the name of FR (Float number Register).

- **For Loop**

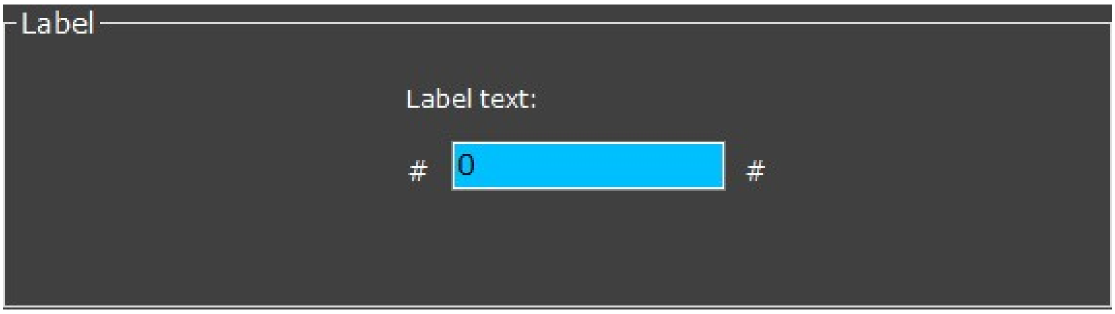
  Looping for this many times (N) then jump to the selected label.

Note:

- ➢ The first run is regarded as the first loop.
- ➢ **The ForLoop function cannot be enclosed in another ForLoop.**

- **Label**

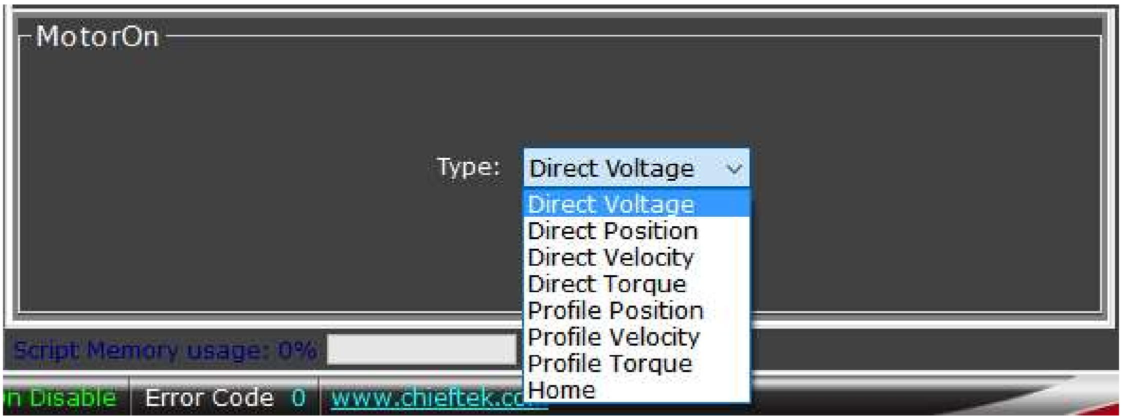
  Set a label and name it.



9.7.6 Motor

- **Motor Off**
- **Motor On**

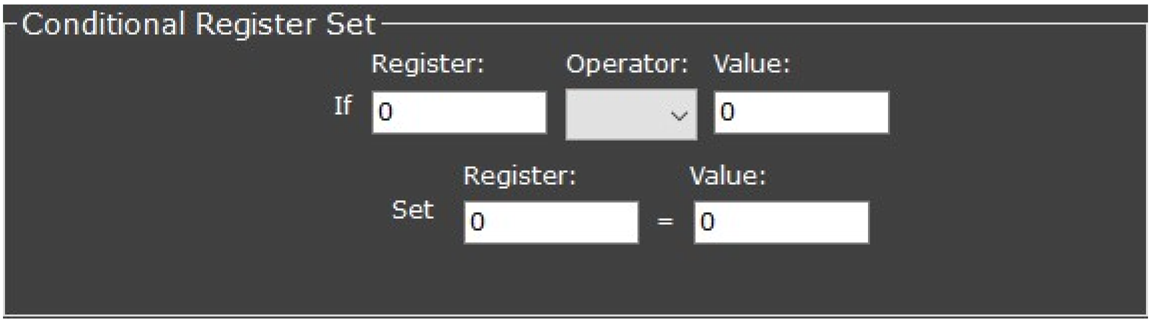
  Set motor on and select a motor-on operation mode from the list.



9.7.7 Register

- **If_ Register Set**

  Set triggering condition and the corresponding reaction.



a. Register:

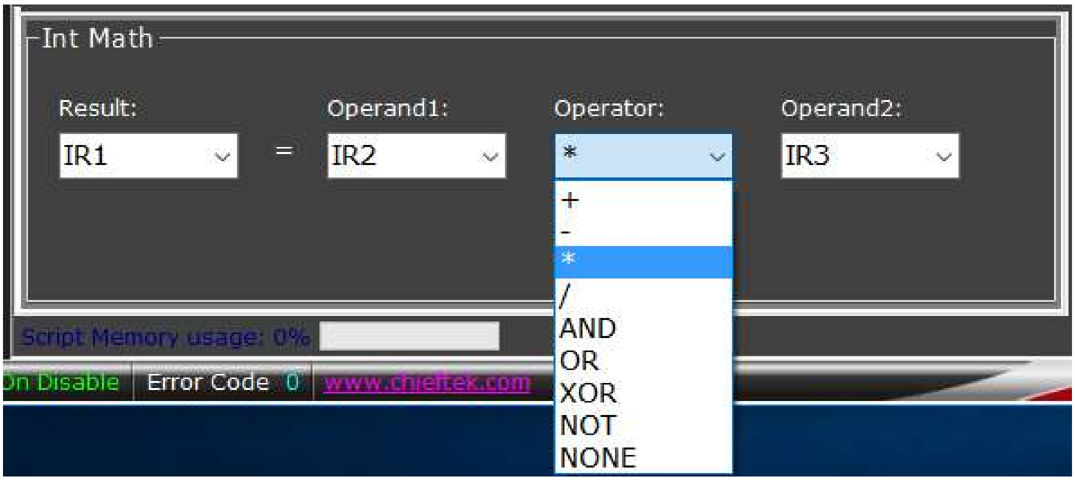
Key in names of IR or FR

b. Value:

Key in number, IR or FR.

- **Int Register Math**

Set an **integer** formula.



Note:

1. Operand1 and Operand2:

   If users key in float numbers into the columns Operand1 and Operand2 under this "Int Register Math" function, the UI will use the keyed-in number's corresponding integer value (according to the IEEE745 system) to process the formula.

2. Operator (mathematics):

   When using **division**, the UI will round down the quotient to an integer; all decimals will be chopped off.

3. Operator (bitwise):

   ➢ Be aware that when **bitwise operation** is used (i.e., AND, OR XOR, NOT), the **operands need to be 16 bit** (value ranging from -2^{15} to $2^{15}-1$).

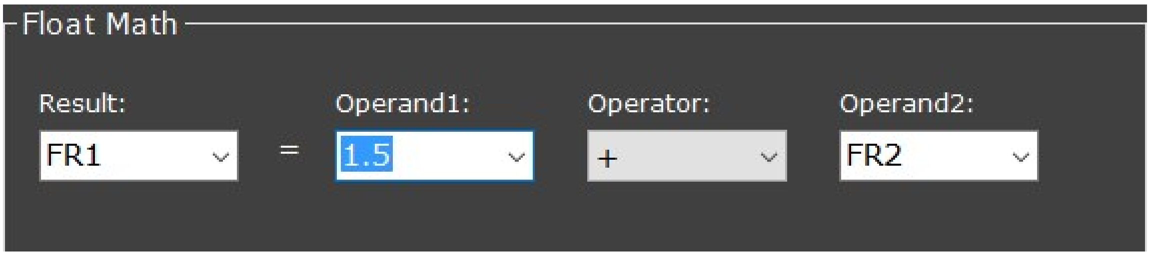
   ➢ Example of the bitwise operation:

Item		Example	a	b	C
AND	&	c = a & b	1100 1101	1001 1110	1000 1100
OR	|	c = a | b	1100 1101	1001 1110	1101 1111
XOR	^	c = a ^ b	1100 1101	1001 1110	0101 0011
NOT	~	c = ~ a	1100 1101		0011 0010

   ➢ The "NONE" option will ignore Operand2.

- **Float Register Math**

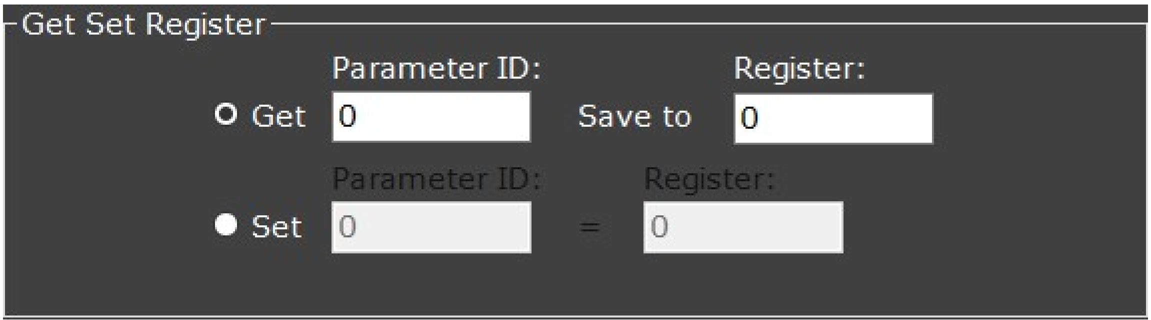
  Set a float number formula;



<u>Note</u>:

1. <u>Operand1 and Operand2</u>:
   Users can key in FR, float number, or integer.
2. <u>Operator (mathematics)</u>:
   The "NONE" option will ignore Operand2.

- **Get Set Register**



a. Parameter ID:

cpc's own ID numbers which corresponds to the numerous parameters mentioned in the CiA® 402 Draft Standard Proposal.

b. Register

The name of integer or float number register.

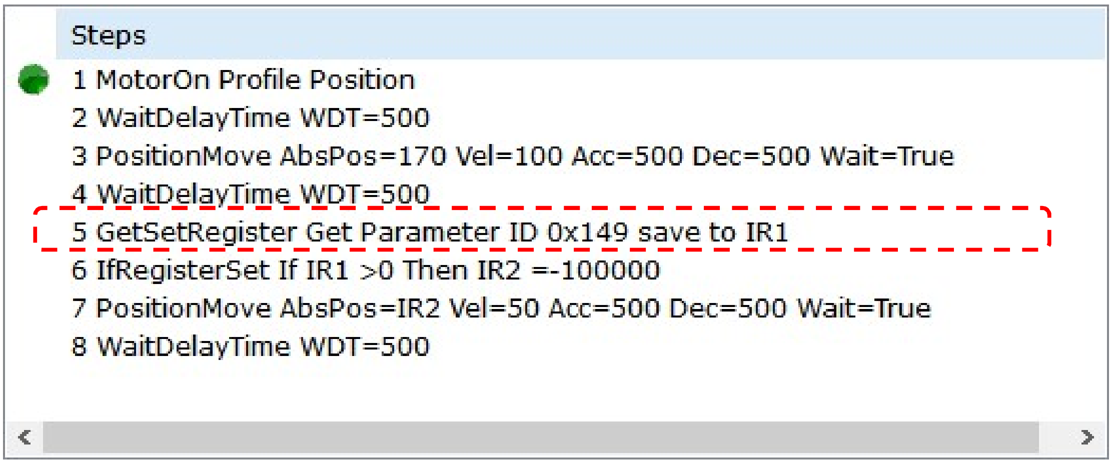
➢ **Get Parameter ID Save to Register**

Get the value out of a parameter ID and save this value to a designated register.

➢ **Set Parameter ID = Register**

Make the value of a parameter ID same as the value of a designated register.

Script Example:



In step 5:

Use “Get Set Register” function to acquire the position command value of the first move and save it as IR1 value.

In step 6:

Use “If_ Register Set” function. If IR1 is bigger than 0, then make IR2 as the configured value.

In step 7:

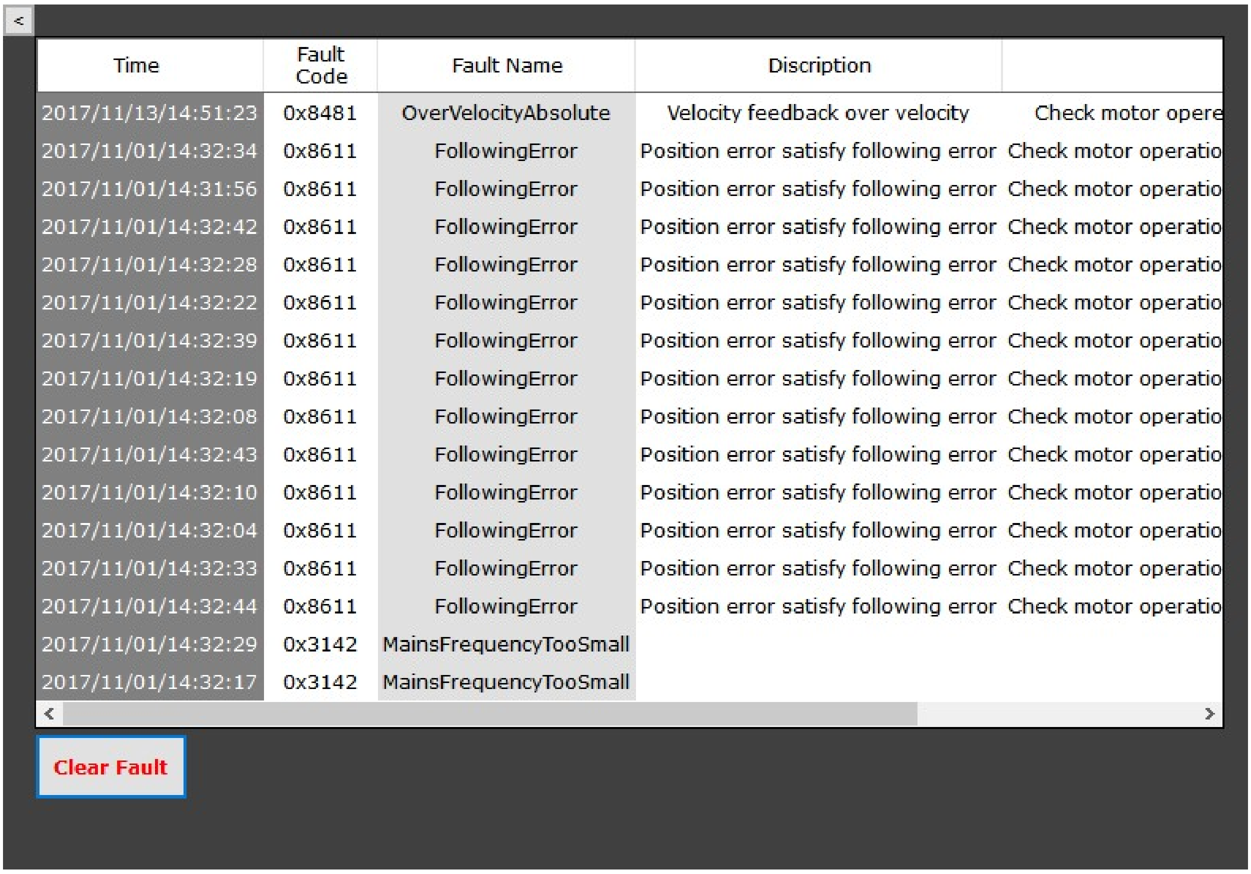
Give the second position command using the IR2 value (processed in “count”) as the target position value.

Chapter 10 Error Log

When a fault event occurs, such as over-temperature, protection threshold exceeded, incorrect wiring, difficulties in accessing signal, etc., the Error Log will record the occurrence time and code of the error. If the driver is shut down due to error(s), users can trace the cause and know how to troubleshoot with the information shown on error log.

Note:

- If the driver is powered off, the record will be cleared.
- The Error Log can record up to **16** instances.



Time	Fault Code	Fault Name	Discription	
2017/11/13/14:51:23	0x8481	OverVelocityAbsolute	Velocity feedback over velocity	Check motor opere
2017/11/01/14:32:34	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:31:56	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:42	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:28	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:22	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:39	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:19	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:08	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:43	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:10	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:04	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:33	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:44	0x8611	FollowingError	Position error satisfy following error	Check motor operatio
2017/11/01/14:32:29	0x3142	MainsFrequencyTooSmall		
2017/11/01/14:32:17	0x3142	MainsFrequencyTooSmall		

Error code definition is in accordance with the specification of CiA 402-2 DSP V3.0.0_Drives and motion. The table of driver error code on the subsequent pages defines all cpc driver error codes and includes descriptions as well as required troubleshooting actions.

<Table of cpc Driver Error Code>

Error Code	Error Message	Description	Action Required
2220	ContinuousOverCurrent_DeviceInternal	Drivers internal power stage is short	Check for unstable current loop gain
2310	ContinuousOverCurrent_MotorSide	Motor current exceeds limit	Check for unstable current loop gain
3210	DCLinkOverVoltage	Internal DC capacitor over 375 V	1. Check external AC supply. 2. Consider adding additional regenerative braking resistor.
3220	DCLinkUnderVoltage	Internal DC capacitor under 48V	Check high voltage supply
4310	ExcessTemperatureDrive	Drivers internal temperature over safe limit	Improve environment cooling condition
5520	ROM_EPROM	Factory calibration lost	Contact Customer Service
5530	EEPROM	Stored user parameters lost	Reload driver setting from file
7121	MotorBlocked	Motor stuck triggered	Check slide and guide or motor stuck setting
7122	MotorErrorOrCommutationMalfunc	Something wrong during phase-find	Check motor wiring and execute phase find again.

Error Code	Error Message	Description	Action Required
7305	IncrementalSensor1Fault	Encoder feedback signal error	Check encoder wiring or improve system noise
7306	IncrementalSensor2Fault	Auxiliary encoder signal transition error	Check controller wiring or improve system noise
8481	OverVelocityAbsolute	Motor velocity feedback over velocity protection limit	Check motor operation or over-velocity protection setting
8611	FollowingError	Motor position error satisfy the condition of following error.	Check 1. Motor operation and 2. The setting of Following Error Window/Timeout
8682	PositionLimitMinimum	Motor position feedback over position protection limit	Check motor operation or over-position protection setting
8683	PositionLimitMaximum	Motor position feedback under position protection limit	Check motor operation or under-position protection setting
90F0	ExternalAlarmDigitalInput	External alarm triggered	Check controller operation
FF01	MainISROverload	CPU overload.	Contact customer service.
FF02	CurrentSensorU	Motor current sensor error	Reboot driver
FF03	CurrentSensorV	Motor current sensor error	Reboot driver

Error Code	Error Message	Description	Action Required
FF05	HallSensorCodeInvalid	Invalid hall sensor code detected	Check hall sensor configuration
FF07	MotorCtrlOpModeInvalid	The code of Operation Mode is invalid	Check Operating Mode and wiring.
FF08	CommutationRequired	Attempting to activate motor without performing phase find first	1. Check phase find setting. 2. Execute phase-find again.

End of Document