Motion Creator Pro 2

Application Manual

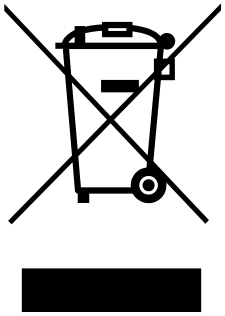
Manual Rev.: 0.1
Revision Date: Dec. 30, 2022
Part No.: 50M-00099-1000

Preface

Disclaimer

The information in this document is subject to change without prior notice in order to improve reliability, design, and function and does not represent a commitment on the part of the manufacturer.
In no event will the manufacturer be liable for direct, indirect, special, incidental, or consequential damages arising out of the use or inability to use the product or documentation, even if advised of the possibility of such damages.

Environmental Responsibility



ADLINK is committed to fulfill its social responsibility to global environmental preservation through compliance with the European Union's Restriction of Hazardous Substances (RoHS) directive and Waste Electrical and Electronic Equipment (WEEE) directive. Environmental protection is a top priority for ADLINK. We have enforced measures to ensure that our products, manufacturing processes, components, and raw materials have as little impact on the environment as possible. When products are at their end of life, our customers are encouraged to dispose of them in accordance with the product disposal and/or recovery programs prescribed by their nation or company.

Trademarks

Product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

Revision History

Revision	Release Date	Description of Change(s)
0.1	2022-12-30	Preliminary release

Table of Contents

1 Introduction

Motion Creator Pro 2 (MCP2) software is a user interface specifically developed for ADLINK motion control products. In a standard Windows environment, MCP2 provides simple configuration, testing, and verification functions without end users' writing code, reducing testing time and accelerating device development.

MCP2 software supports all ADLINK motion control products, including pulse control cards and EtherCAT control card, and provides basic motion control functions such as single-axis motion, tween motion, and homing. Functions such as axis status and IO status on the user interface can also be monitored in real time.

1.1 Functional Description

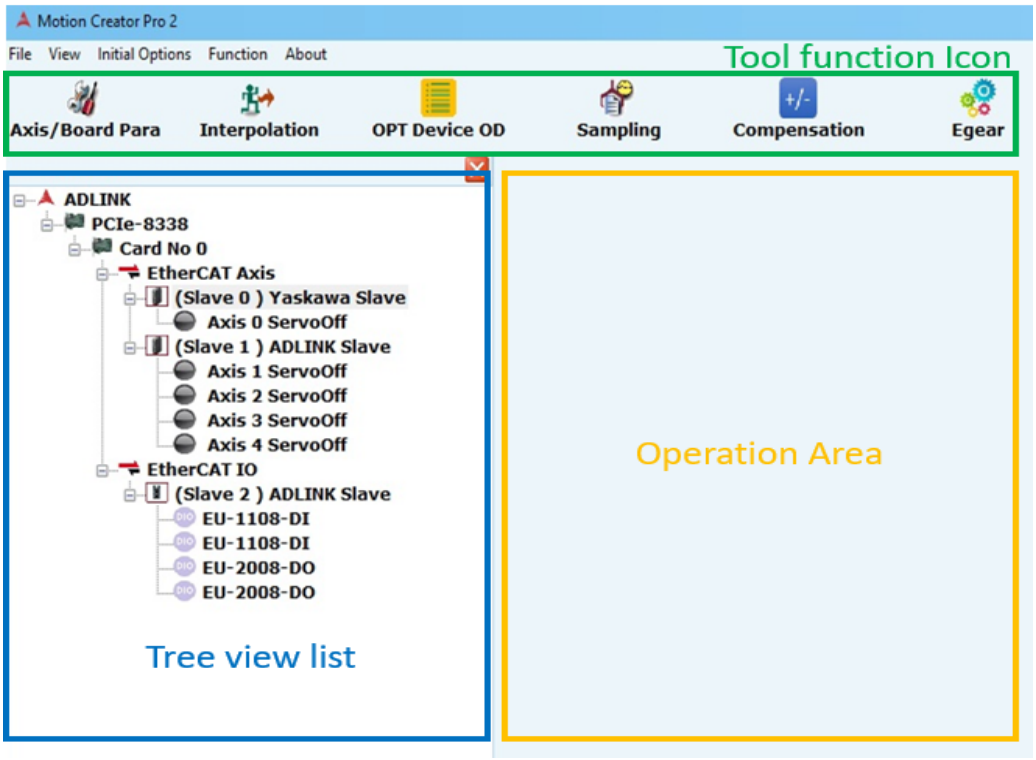
The Automation Product Software (APS) library provides users a uniform interface to access all ADLINK products that support it and covers many automation fields including machine automation. The APS library supports motion control with components. The following functions are provided by the software. (Refer to the actual motion control card being used for supported functions. Functions vary according to card type.)

- **Parameter Setting**: Parameters can be loaded and saved as files.
- **Single Move**: Operation buttons and setting for 3 modes—Absolute, Relative and Velocity (with partial support for JOG).
- **Multi-Axes Interpolation**: Supports multi-axes Linear, Circular and Spiral interpolation.
- **Homing Move**: This function is used to define the HOME position of the axis.
- **Interrupt**: This function is used to enable/disable interrupts of one board to host computer.
- **Gantry Function**: This is a structure used to straddle an object or workspace.
- **Egear Function**: Electronic gear.
- **Advanced Point Table**: This function is used to create a set of point table parameters to specified axes.
- **Compare Trigger**: Sends an output signal when triggered.
- **Latch**: Remembers the current location when receiving an input signal.
- **Axis Data Sampling**: This function is used to set sampling parameters such as sampling rate and sampling channel source.

2 Software Introduction

2.1 Interface

MCP2 supports ADLINK motion control and distributed I/O series products, providing a unified interface for users to configure parameters and perform motion control with simultaneous monitoring and management. The following figure shows the software interface, which is divided into three main parts: **Tool function icons**, **Tree view list** and **Operation Area**.



When MCP2 is started, all cards on the system will be displayed in the Tree view list. Tool function icons will vary according to card type. All control item interfaces will be uniformly displayed in the Operation Area.

2.2 Installation

This section describes the installation and use of ADLINK's APS SDK for motion control, which contains all motion control card driver installation files.

NOTES:

- The APS SDK is supported by Microsoft Windows only.
- Be sure to use proper ESD (Electrostatic discharge) protection when working with any electronic equipment.

1. Download and install the latest version of the APS SDK from ADLINK's official website.
2. Read the related manuals carefully to properly set up signal I/O.
3. Turn off power to your computer and all relevant terminal boards before inserting a device card into a PCI/PCIe slot in your computer.
4. Turn on the motion control limit switches and set up servo signals and general-purpose digital signal wiring.
5. Set up servo or stepper drive connections.
6. Turn on power to your computer and connected terminal boards.
7. Verify all I/O signals and servo operations are correct with Motion Creator Pro 2.

2.3 Troubleshooting

If the computer fails to power up normally, or the motion control system operates abnormally after installation, check the following troubleshooting scenarios for corrective action.

The card does not show up in Device Manager after its driver has been installed.
Turn off the computer and ensure that the card is properly seated in the PCI or PCIe slot. Ensure that the driver is properly installed by checking for it in **Control Panel>Add remove programs**.

Motion Creator Pro 2 cannot run after installing the driver.
Ensure that the latest version of APS SDK has been installed on the system.

The “without signal” indicator on Motion Creator Pro 2 lights up after the motor is connected and the motor does not work.
Ensure that 24V DC power is connected to the terminal board.

When using Motion Creator Pro 2 all the control indicators of the drive light correctly but the drive has a warning.
Ensure that axis parameter, alarm logic (ALM) and the EMG loop configuration have been set up correctly.

Value of output command differs from the feedback value.
From the encoder, ensure feedback signal settings (CW/CCW, 1xAB, 2xAB, 4xAB) align with those of the drive.

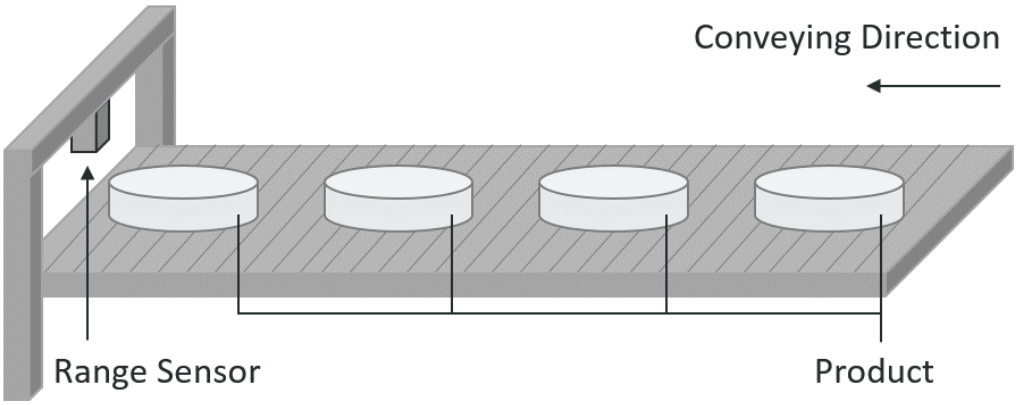
For motion control, the motor moves only in one direction rather than back and forth.
Ensure the signal pattern settings (CW/CCW, OUT/DIR) align with those of the motor drive. If the problem persists, contact your dealer for technical service.

3 Usage Scenarios

3.1 Latch Triggering

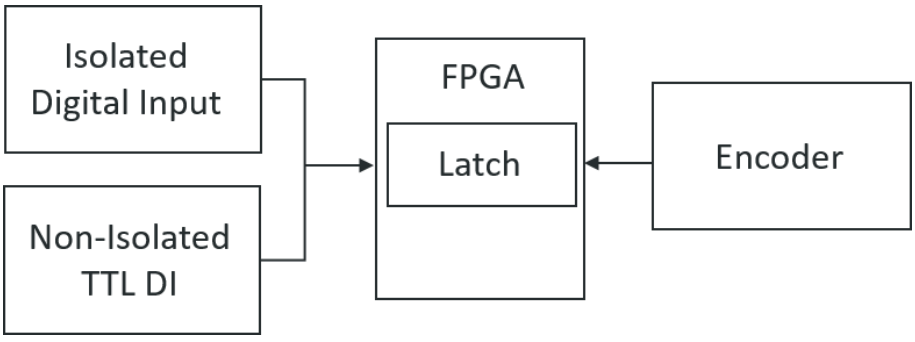
A conveyor belt is equipped with a range detector for fixed-point scanning. When a product enters the detection area, the range detector signal triggers the latch function, recording the location in a Latch Get Point Array each time a product passes through the area.

Similar usage scenarios include scanning labels to record the location of the product, product temperature scanning for inspection and screening, and grasping or pushing a product through distance detection.



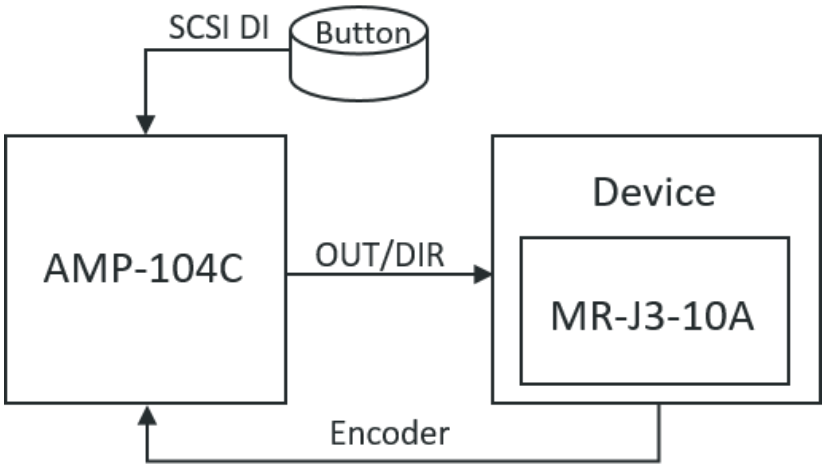
In this scenario, MCP2 is used to trigger the latch function of an AMP-104C. An AMP-104C provides four isolated digital inputs and four transistor–transistor logic (TTL) digital inputs as sources for latch signals.

After the detection device is triggered, the latch function records the location indicated by the encoder.



Hardware Wiring

This scenario requires the following hardware wiring, as shown in the diagram below.



- The AMP-104C sends an output pulse signal to the MR-J3-10A driver.
- The AMP-104C receives an encoded signal from the MR-J3-10A driver.
- A button uses the SCSI DI to simulate latch signal triggering.

The following table describes the hardware components used in this scenario.

Name	Hardware Component	Name	Function	Connector
Terminal Board	DIN-68S-01	OUT0+	Pulse Signal +	CN2
		OUT0-	Pulse Signal -	
		DIR0+	Direction Signal +	
		DIR0-	Direction Signal -	
Terminal Board	DIN-37D-01	EA0+	Encoder A-phase+	CN4
		EA0-	Encoder A-phase-	
		EB0+	Encoder B-phase+	
		EB0-	Encoder B-phase-	
Sensor	Button	DI0	Digital Input	CN2
Device	MR-J3-10A	LA	Encoder A-phase pulse	CN1
		LAR	differential line driver	
		LB	Encoder B-phase pulse	
		LBR	differential line driver	
		PP	Pulse F +	
		PG	Pulse F -	
		NP	Pulse R +	
		NG	Pulse R -	

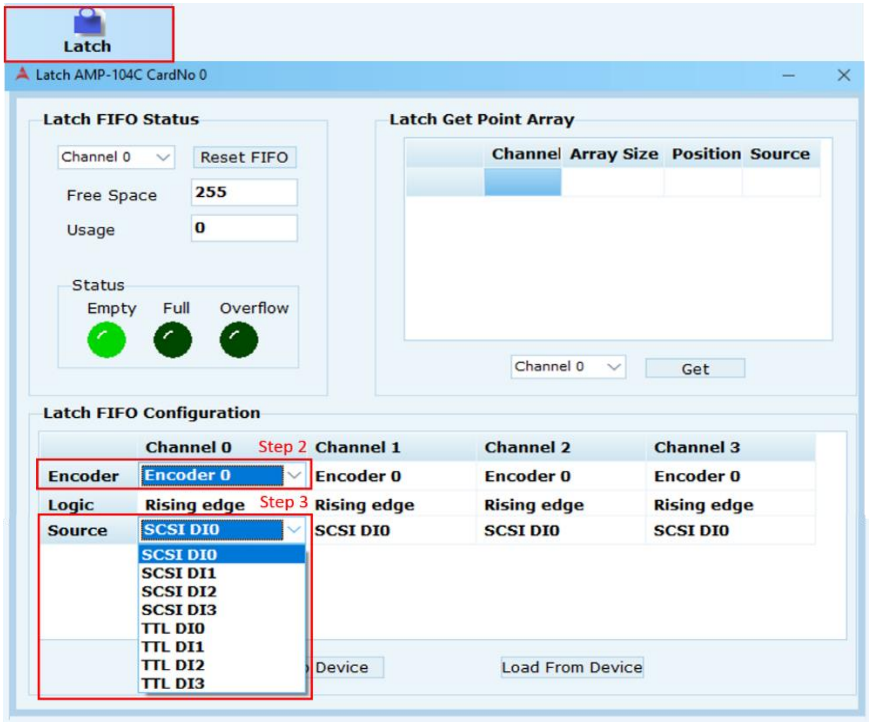
Configuration

These steps describe the Latch Triggering usage scenario configuration.

1. Set the **Pulse output mode** and **Pulse input mode** of the AMP-104C to match the driver.

2. Select the latch icon in MCP2 and set the **Latch FIFO Channel 0 Encoder** as **Encoder 0.** When a Latch event occurs, Latch FIFO Channel 0 records the current position of Encoder 0.

3. Set the **Latch FIFO Channel 0** signal source as **SCSI DI0.** When DI0 receives an input signal, Latch FIFO Channel 0 records the current position of Encoder 0.



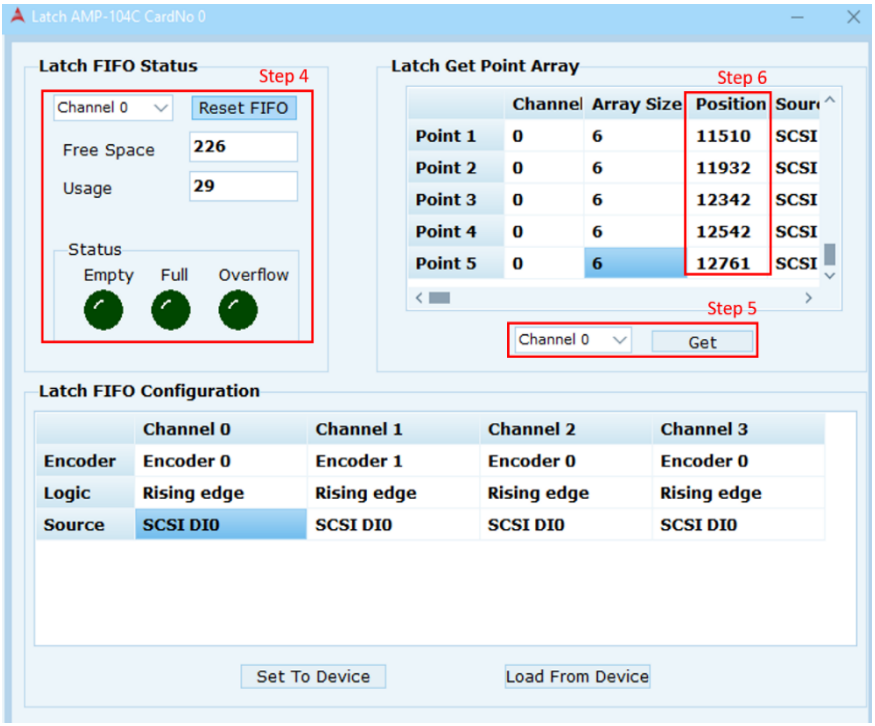
Results

The following results can be observed in MCP2 based on the Latch Triggering configuration settings.

4. When the latch signal has input and starts to record the location of the encoder, the Free Space, Usage and Status elements will show the current Latch FIFO status.

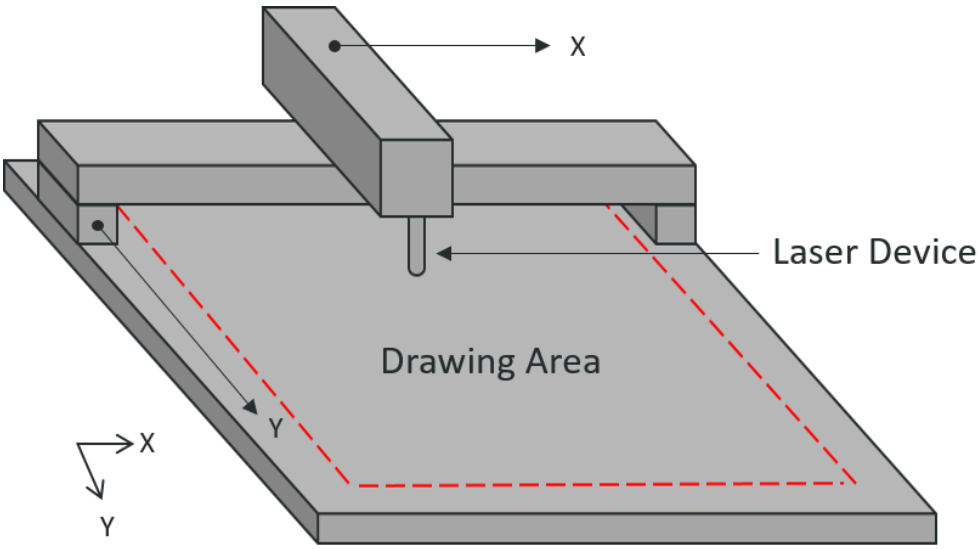
5. Select a channel to display and click Get to obtain the recorded position.

6. The recorded encoder positions are displayed in the Position column of the Latch Get Point Array.

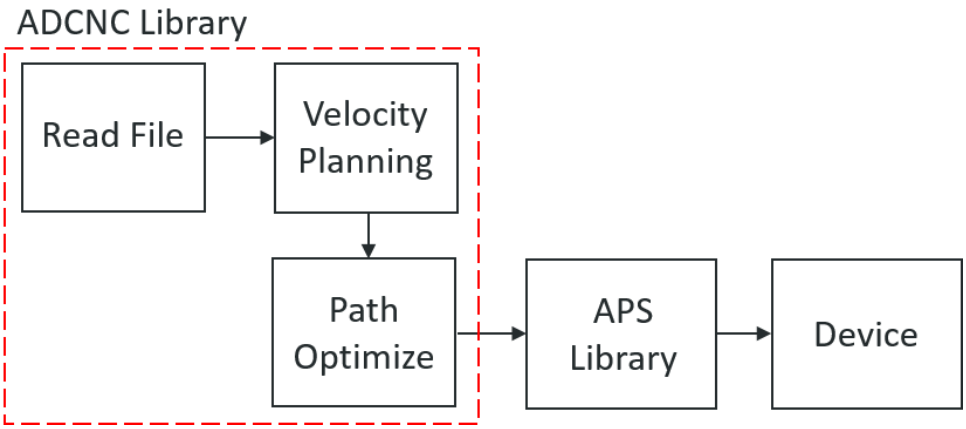


3.2 Advanced Point Table

A machine has two gantries to move a laser engraver. The Y-axis gantry moves back and forth across the drawing area, while the X-axis gantry is equipped with a laser and moves left and right across the drawing area. A product is placed in the drawing area where the laser is used to engrave a logo on it using an advanced point table to plan a continuous path for the laser to transit. This application similarly applies to steel plate cutting, or 2D painting, for example. The following example demonstrates how MCP2 can be used to manipulate the AMP-208C's Advanced Point Table function.

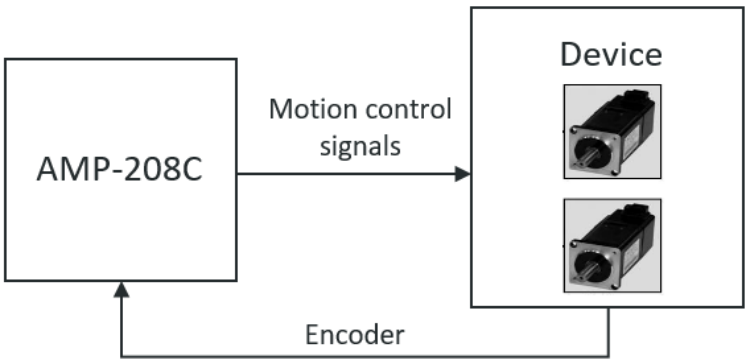


After the path is read by the ADCNC Library, path optimization and speed planning, or only speed planning, are performed and the APS Library is combined with the motion control card to control the equipment.



Hardware Wiring

This scenario requires the following hardware wiring, as shown in the diagram below.



- The AMP-208C sends an output pulse signal to the driver.
- The AMP-208C receives the Encoder from the driver.
- The CMP connector on the terminal board is connected to the device for driving the laser. When P1-A is connected, the CMP connector is at control axis 0 to 3. When P1-B is connected, the CMP connector is at control axis 4 to 7.

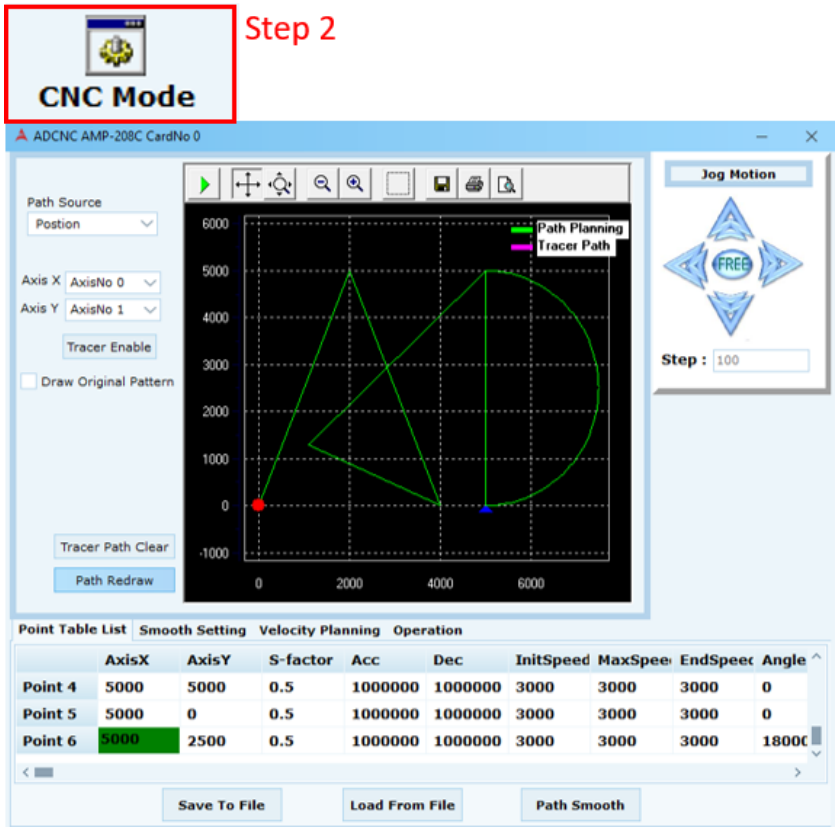
The following table describes the hardware components used in this scenario.

Name	Hardware Component	Axis	Function	Connector
Terminal Board	DIN-825-4P0	-	Motion control signals	P1
		Axis0	Connecting to servo drive	CMP1
		Axis1		CMP2
Device*1	MR-J3-10A	Axis0	Control Motion signals	CN1
Device*2	MR-J3-10A	Axis1		CN1

Configuration

These steps describe the Advanced Point Table usage scenario configuration.

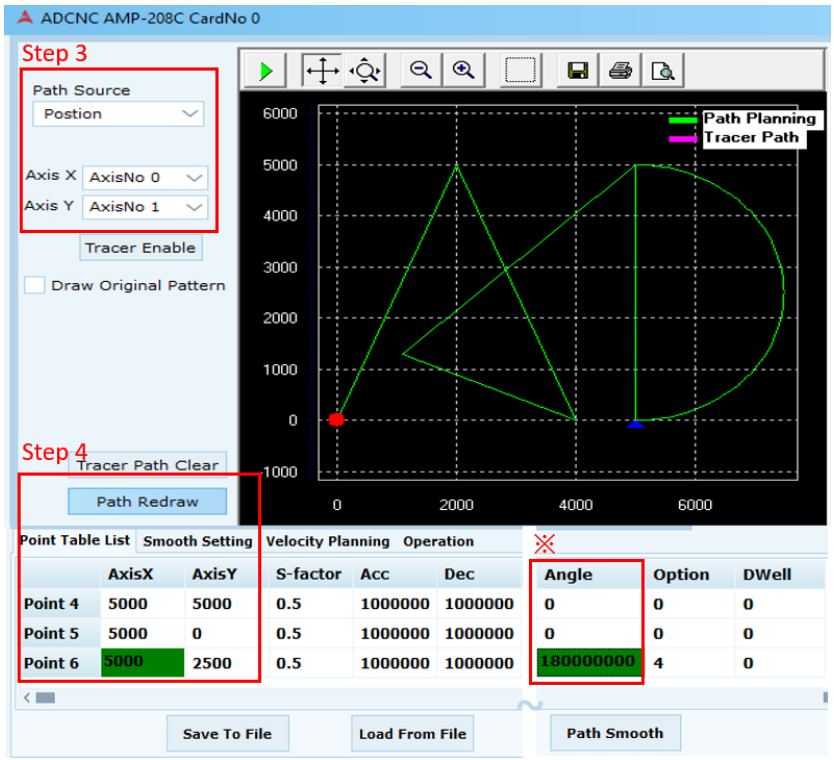
1. Match the **Pulse I/O mode** of the AMP-208C to the driver.
2. Click **CNC Mode** to call the related interface.



3. For **Path Source**, select **Position** to refer to the actual moving position and set **Axis X** as **AxisNo 0** and **Axis Y** as **AxisNo 1**.

4. Enter the path (see diagram above) in the **Point Table List**. You can preview the control command trajectory by clicking **Path Redraw** on the green line in the drawing area (Path Planning).

   Note: Point 6 sets the arc mode, and Angle is set to 180 degrees to draw the semicircle. (Angle is calculated as 0.000001 degree.)

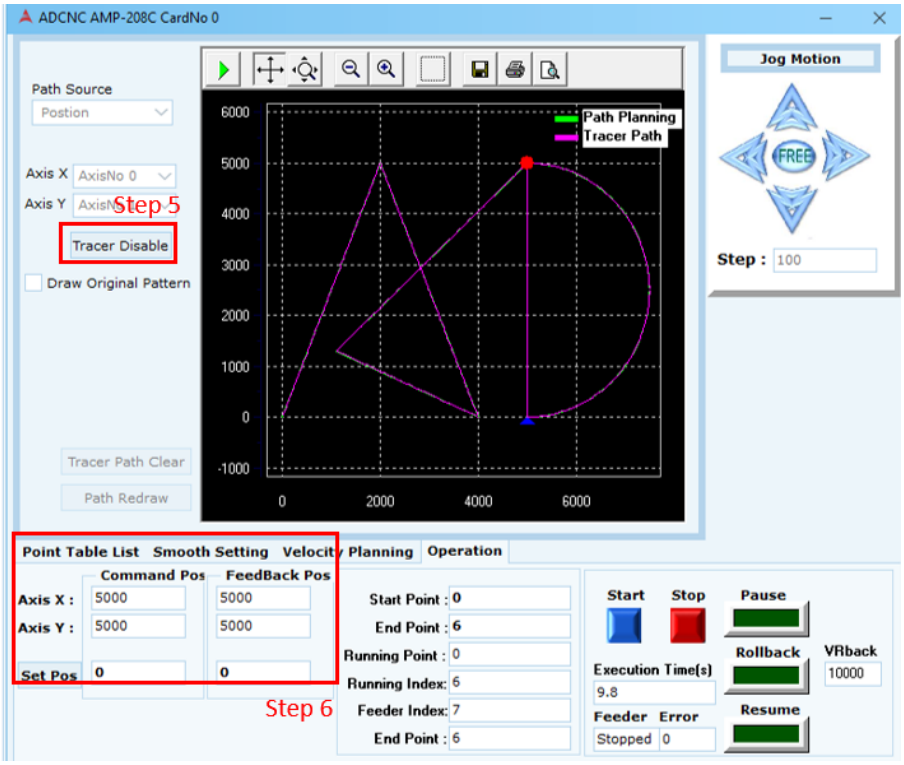


Results

The following results can be observed in MCP2 based on the Advanced Point Table configuration settings.

5. Click **Tracer Enable**. The drawing area displays a pink Tracer Path during execution.

6. The operation field can synchronously display the pulse signal sent by the motion control card, as well as the driver feedback position signal.

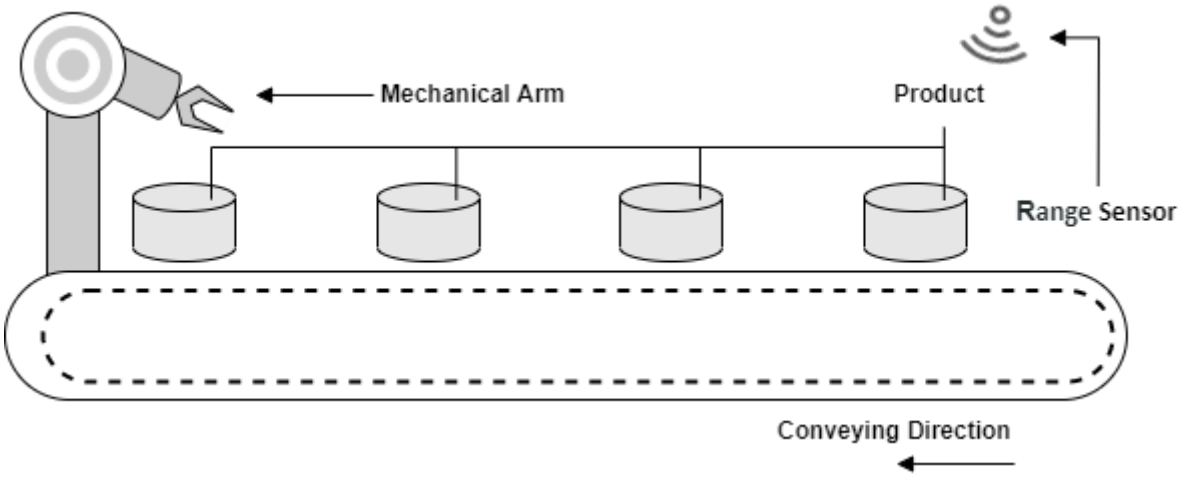


3.3 Compare Trigger

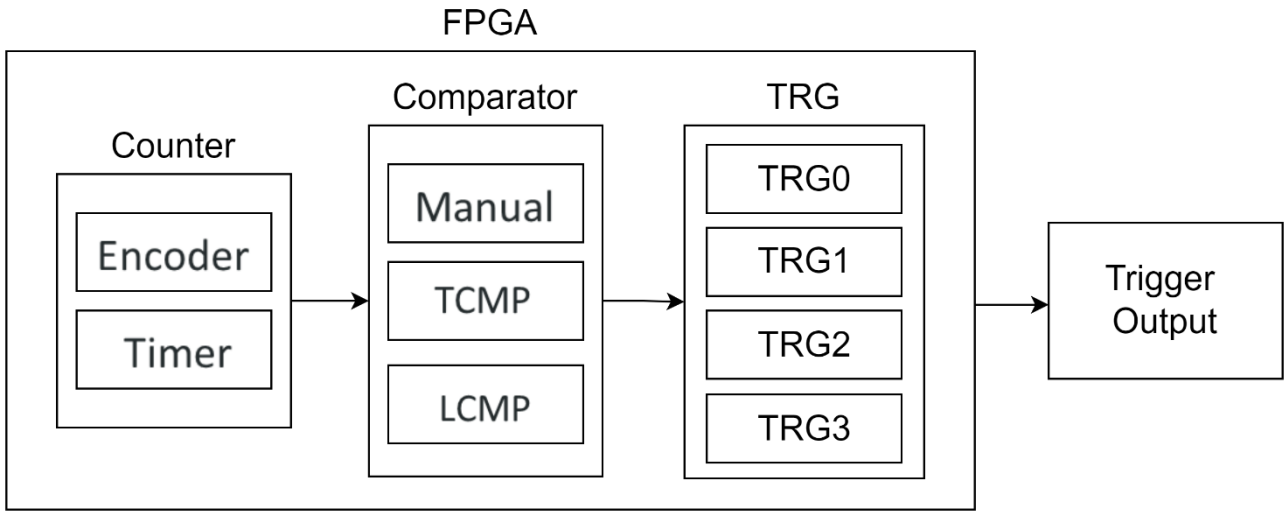
A range sensor obtains the location parameters of products on the conveyor, including the distances between the products, and stores the information in a table so that a robot arm at the end of the conveyor can use the information to obtain the correct grasping position for each product as it travels on the conveyor.

When the Compare function compares the location of the conveyor belt with the parameters in the table, it sends a Trigger signal to make the robot arm grasp the corresponding product in the correct position. Related applications include, factory logistics transport packaging, for example.

The following example demonstrates how to use MCP2 to manipulate the AMP-208C's Compare Trigger function.

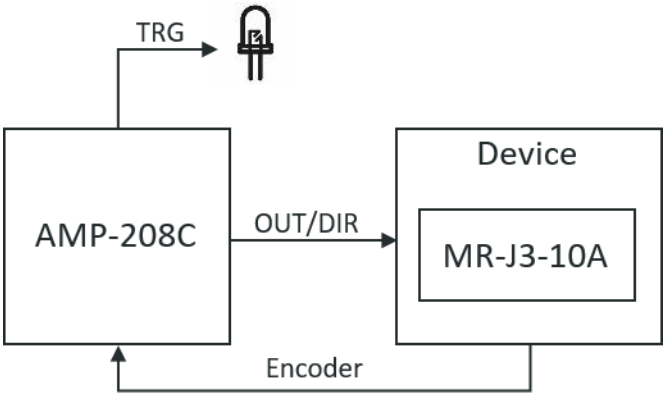


The AMP-208C signal counter sources include Encoder and Timer, while there are three comparator types: Manual (Manual), table (TCMP) and linear (LCMP). When the counter signal and comparator match, a trigger signal is sent.



Hardware Wiring

This scenario requires the following hardware wiring, as shown in the diagram below.



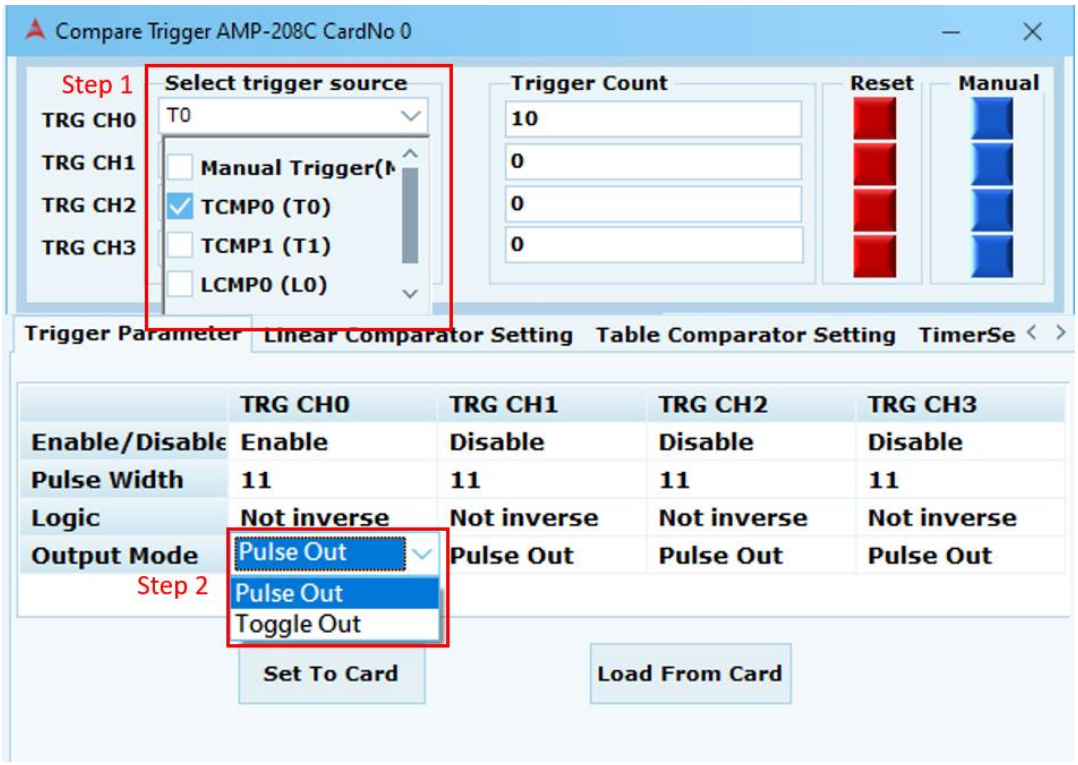
- The AMP-208C sends an output pulse signal to the driver.
- The AMP-208C receives the Encoder from the driver.
- The AMP-208C sends a TRG signal to an LED through the junction board to simulate the Compare Trigger function.

Name	Hardware Component	Function	Connector
Terminal Board	DIN-825-4P0	Motion control signals	P1
		Connecting to servo drive	CMP1
Device	MR-J3-10A	Control Motion signals	CN1
Device	LED	Trigger output	TRG0+

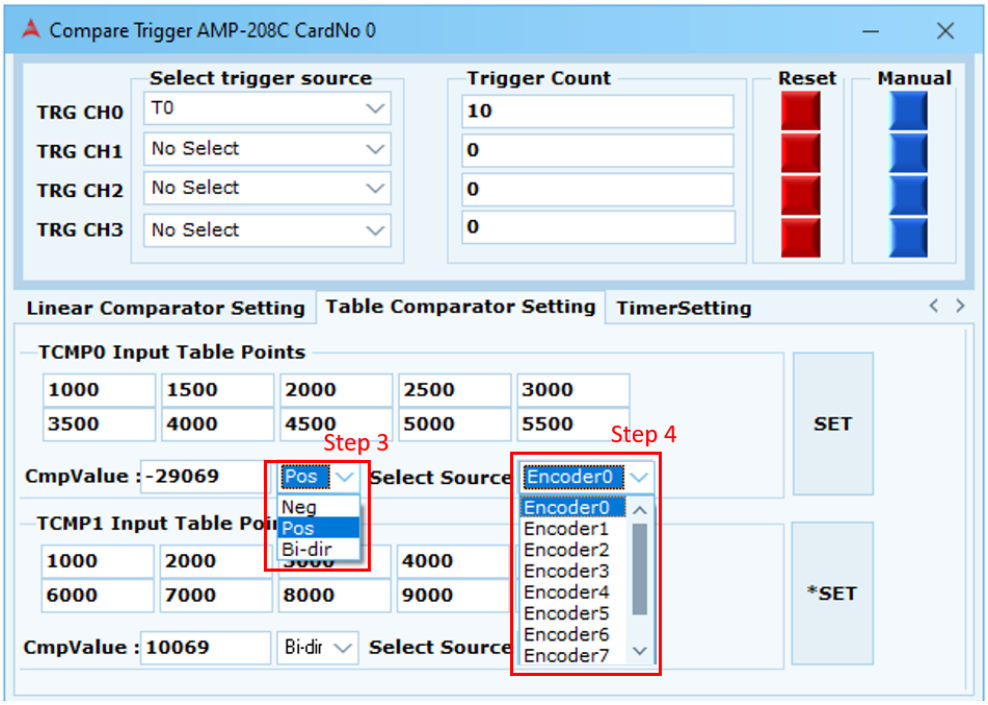
Configuration

These steps describe the Compare Trigger usage scenario configuration.

1. Select **TCMP0** as the channel trigger signal source.
2. The output of the comparator output mode switches to the actual device.



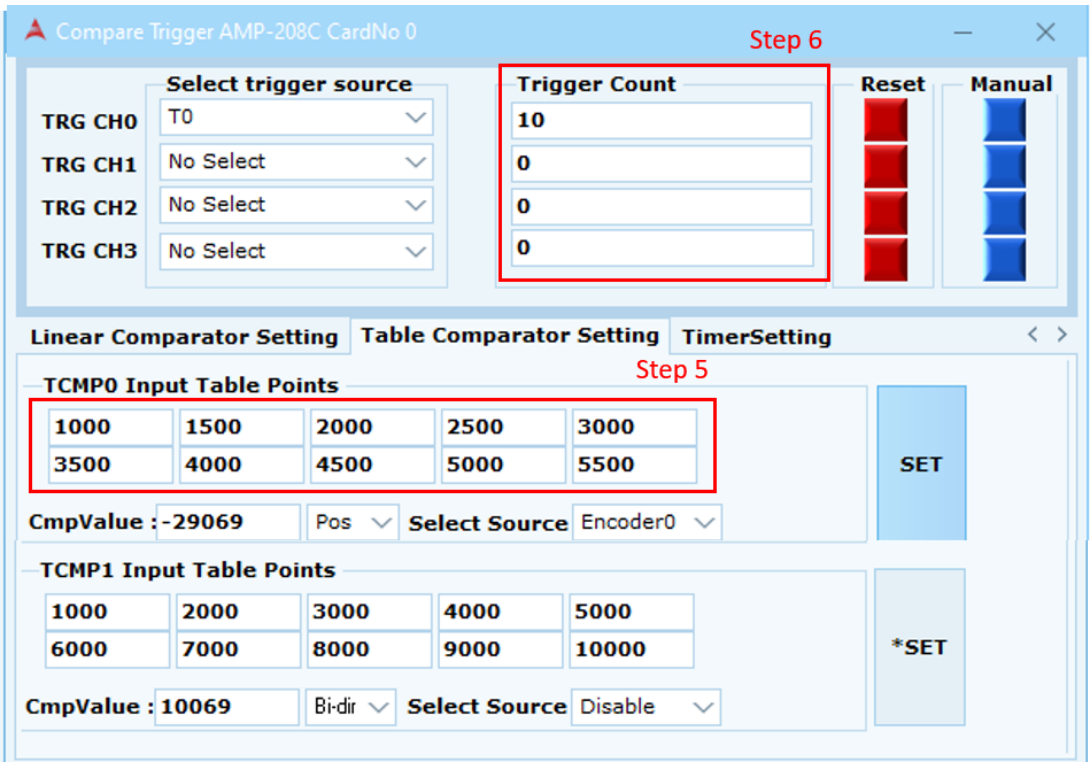
3. Set the direction of the comparator.
4. Select **Encoder0** as the location comparison source.



Results

The following results can be observed in MCP2 based on the Compare Trigger configuration settings.

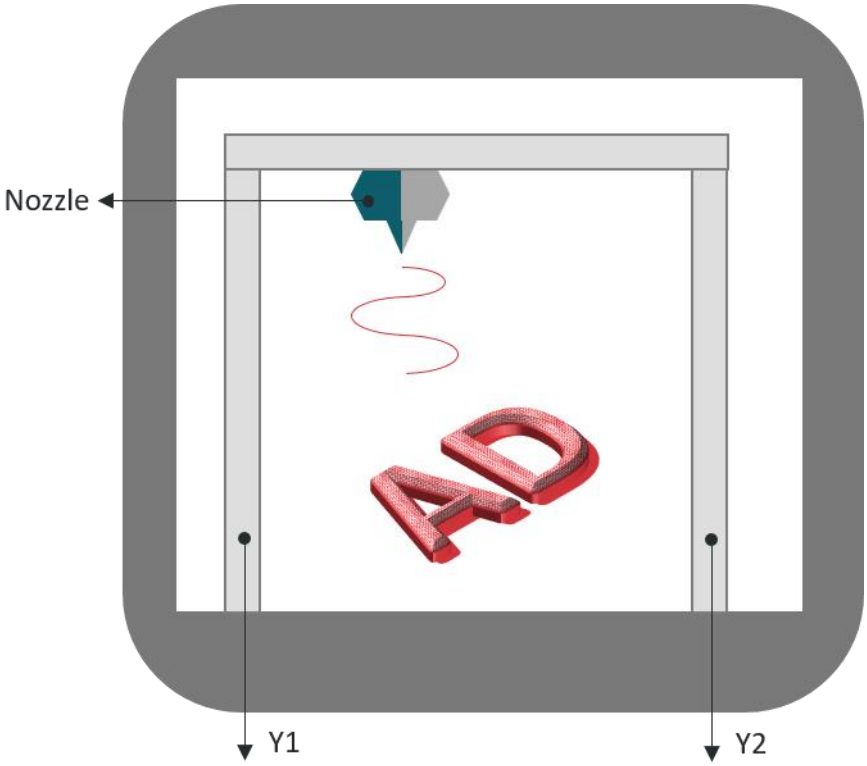
5. Fill the parameters to be compared into **TCMP0 Input Table Points**.
6. When **Compare Trigger** is enabled and triggered, **Trigger Count** will record the trigger times.



3.4 Gantry Mode

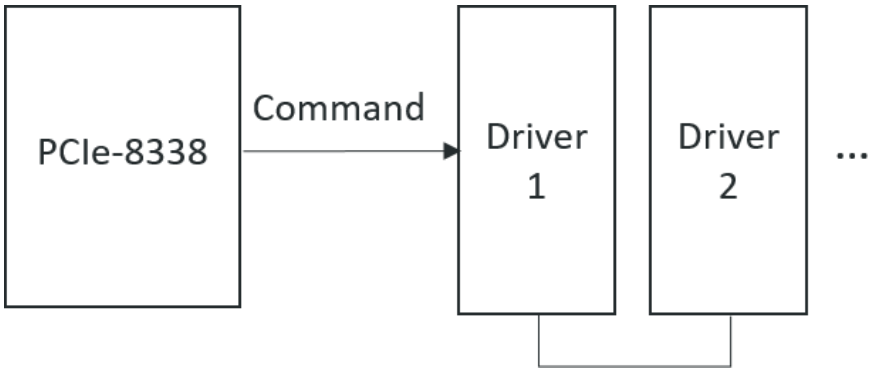
A large 3D printer uses fused deposition technology. The material must be placed in the heating area during production, and then extracted and stacked by the nozzle to be cooled and formed. With the Gantry function, the dual Y-axis drive is used to stabilize the machine, and the heating mechanism moves the nozzle to print at the same time. Related applications include, electronic material AOI and PCB automatic stamping, for example.

The following example demonstrates how to use MCP2 to operate the Gantry function with a PCI-8338 PCIe EtherCAT master motion controller.



Hardware Wiring

This scenario requires the following hardware wiring, as shown in the diagram below.



Note: The PCI-8338 belongs to the EtherCAT series and is connected to the driver through a network.

Name	Hardware Component	Function	Connector
Card	PCIe-8338	Basic EtherCAT Communication	ECAT0
Device*1	SDP-010E2C	Control Motion signals	CN3
Device*2	SDP-010E2C		CN3

Configuration

These steps describe the Gantry Mode usage scenario configuration.

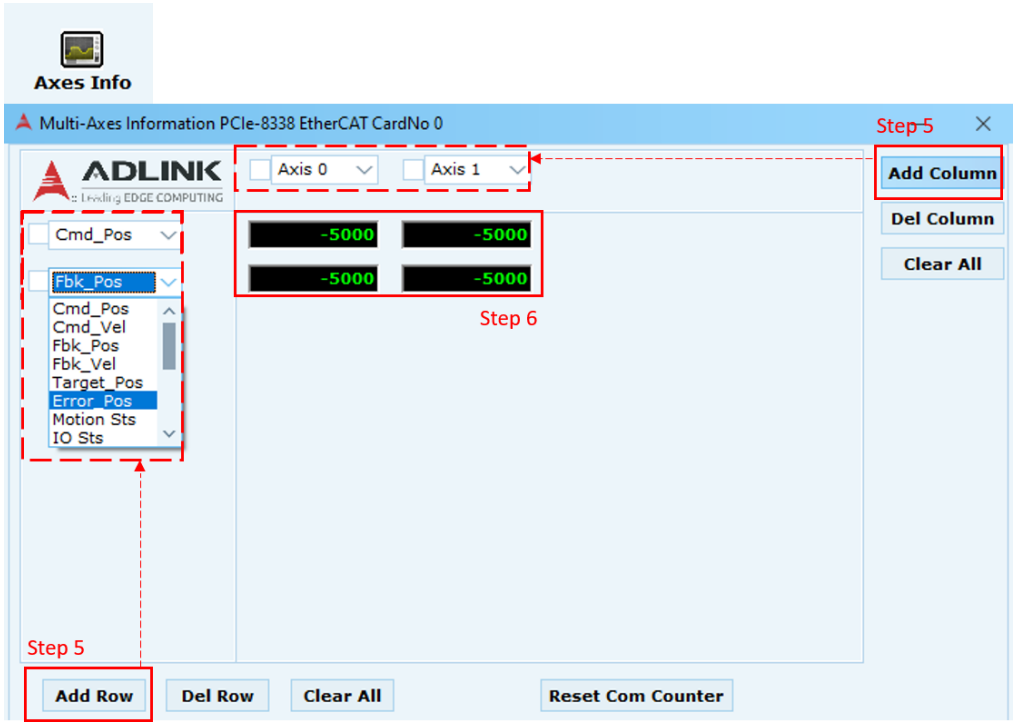
1. Select the **Egear** call interface and the select **Gantry** mode.
2. Set **Axis 0** to **Axis 1 Command Position** as the main control axis and set the **Level1** Gear Parameter to **2500** and **Level2** to **6000**.
3. Set **Master Axis** as **Axis 1**.
4. The control modes are absolute mode (ABS Move) and relative mode (REL Move), as shown.

Results

The following results can be observed in MCP2 based on the Gantry Mode configuration settings.

5. Use MCP2 to select **Axes Info** to observe data in movement. Use **Add Row** to add extra information. The number of axes to be observed can be selected from **Add Column**.

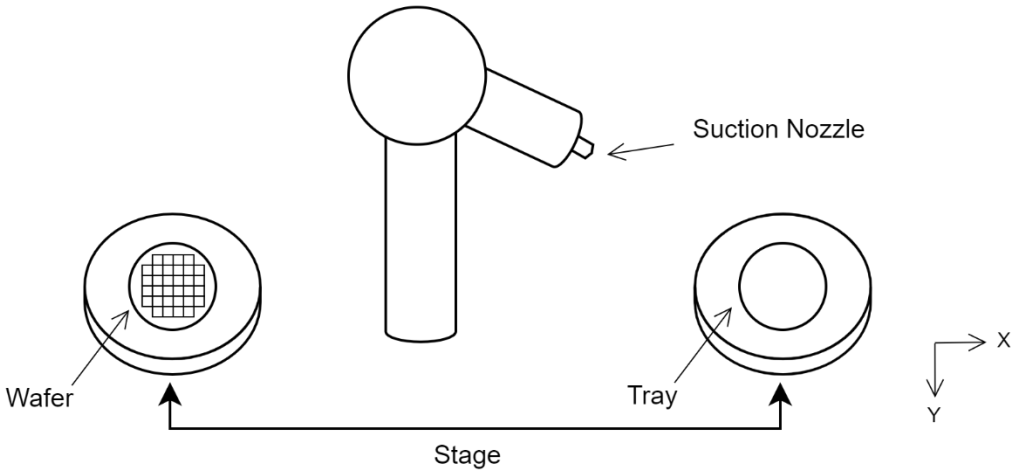
6. The operation results are shown in the display area.



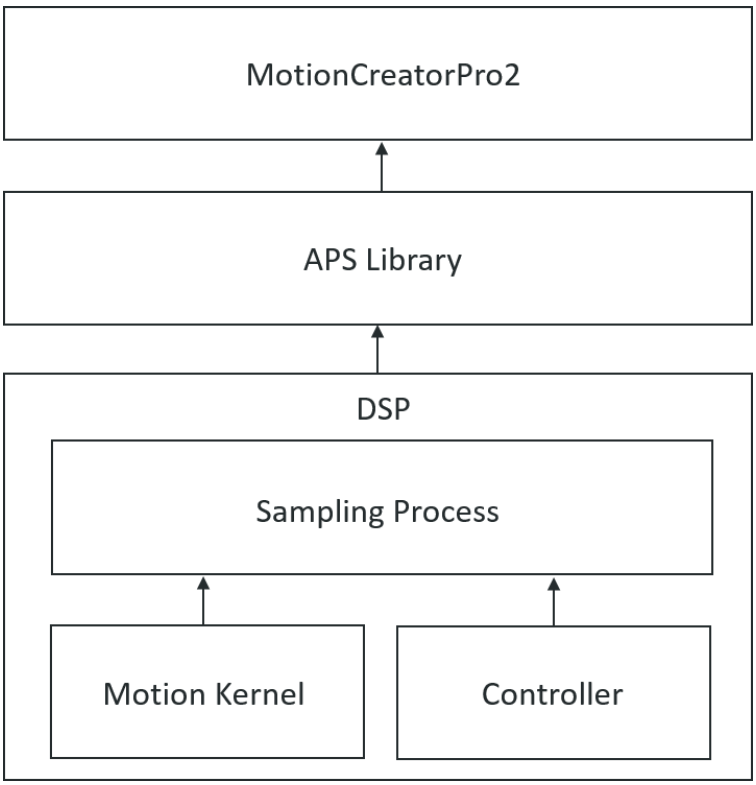
3.5 Sampling

This use case focuses on a high efficiency chip sorting machine with a mechanical arm with a suction nozzle that moves back and forth between a wafer and a tray. With Sampling, the relationship between the position and speed of the arm and the suction nozzle can be observed so the mechanical arm picks and places the wafer at the correct time and position to ensure that the machine maximizes chip sorting efficiency. Related applications include electronic material inspection or packaging and factory logistics transportation, for example.

The following example demonstrates how to use MCP2 to manipulate the Sampling function of an AMP-208C advanced pulse-train motion controller.



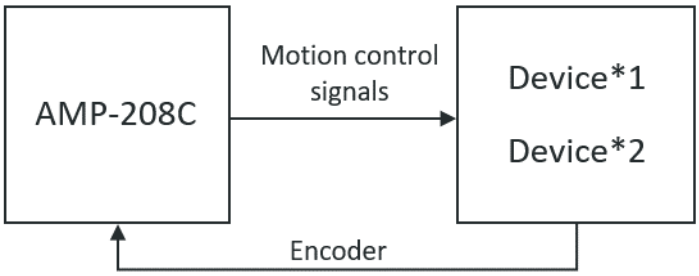
When implementing Sampling, the minimum sampling rate is 1ms. After DSP processing, data will be transmitted and displayed in Motion Creator Pro 2, or other application, by the APS Library.



Hardware Wiring

This scenario requires the following hardware wiring, as shown in the diagram below.

- The AMP-208C sends an output pulse signal to the driver.
- The AMP-208C receives an Encoder from the driver.

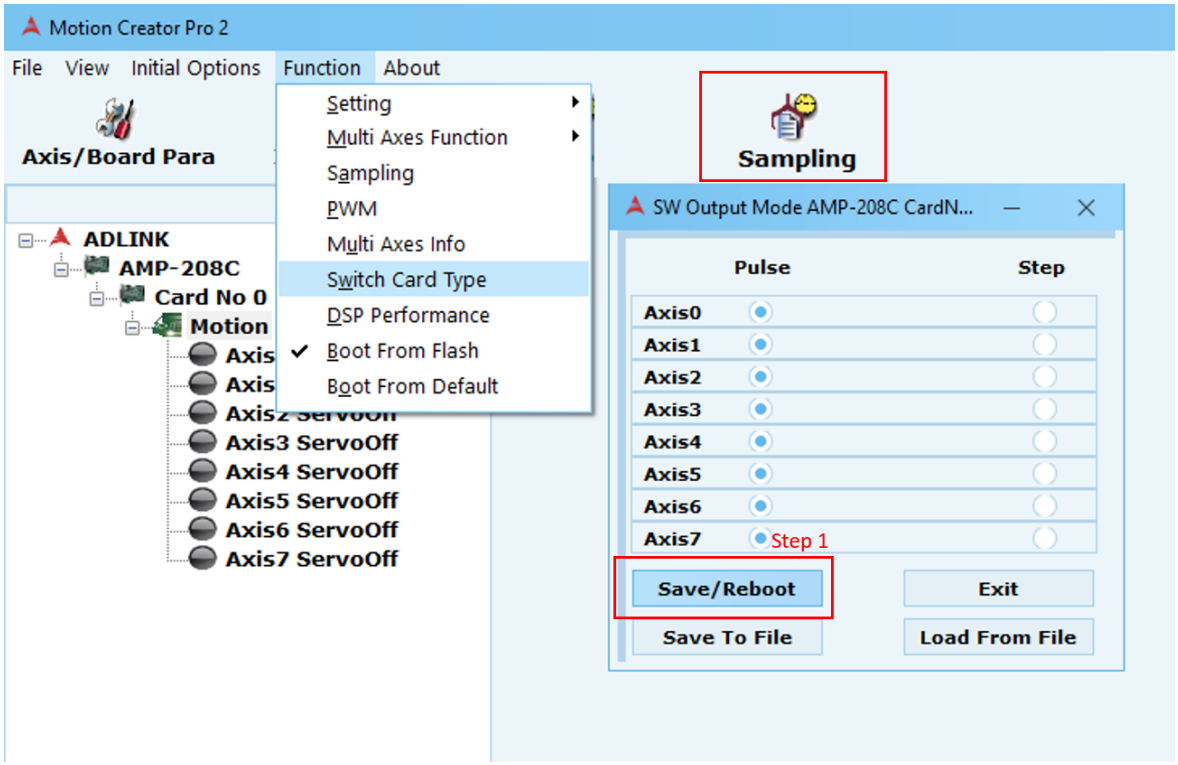


Name	Hardware Component	Function	Connector
Terminal Board	DIN-825-4P0	Motion control signals	P1
		Connecting to servo drive	CMP1
		Connecting to servo drive	CMP2
Device*1	MR-J3-10A	Control Motion signals	CN1
Device*2	MR-J3-10A	Control Motion signals	CN1

Configuration

These steps describe the Gantry Mode usage scenario configuration.

1. Set **Switch Card Type** to **Pulse** or **Step** and click **Save/Reboot** to save. Click **Sampling** to call the relevant interface.

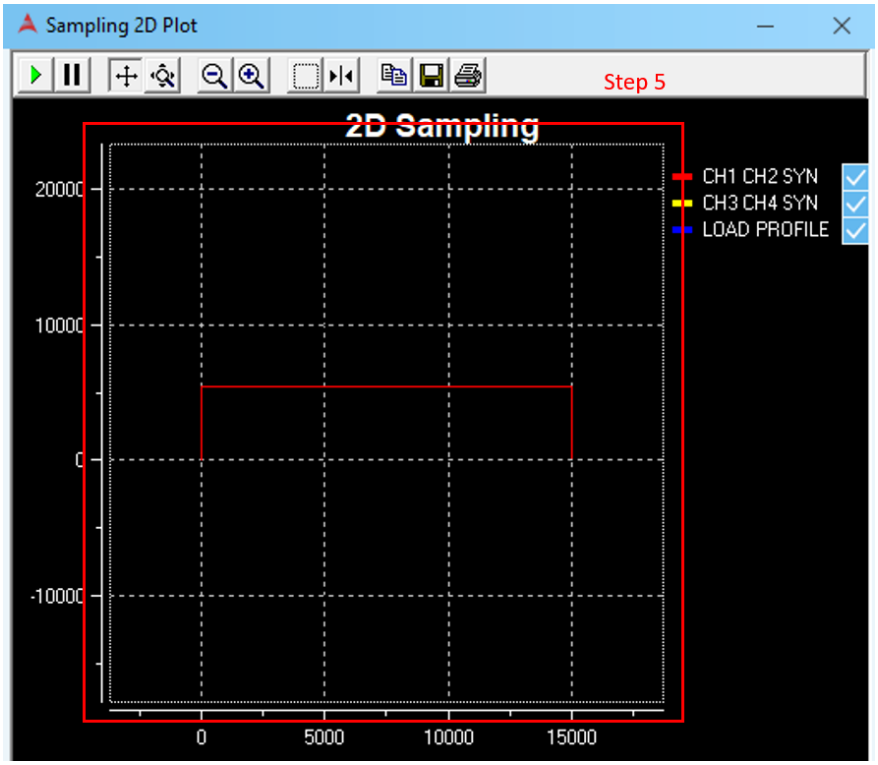


Results

The following results can be observed in MCP2 based on the Gantry Mode configuration settings.

2. Set channels 1-4 Command and Feedback Positions to either Axis 0 or Axis 1 as shown in the diagram below.
3. The display shows the pulse wave over time.
4. Click **2D Plot** to draw the plane movement path.

5. CH1 and CH2 are the Command Position of axis 0 and 1, and CH3 and CH4 are the Feedback Position of axis 0 and 1. Pulse waves sent at the same time are consistent with the actual movement trajectory, and 2D Plot will show any overlaps.



6. Assuming that the sent pulse wave is not synchronized with the actual movement trajectory, 2D Plot will show non-overlapping trajectories of CH1 and CH2, CH3 and CH4, as shown in the figure below.