Pulse & Direction Input Command

Application Note

agito

www.agito-akribis.com

Member of Akribis Systems group

Revision History

Version	Description	Date
1.2	Added AGD101 pinout description.	8 April 2026
1.1	Include information about position rollover	26 Feb 2024
1.0	Initial release	2 Oct 2023

Contact Information

Manufacturer	Agito Akribis Systems Ltd., Member of Akribis Systems Group
Address	56 Serangoon North Avenue 4, Singapore 555851
Email	info@akribis-sys.com
Website	www.agito-akribis.com

Copyright Notice

Products Rights

AGDx, AGCx, AGMx, AGAx, AGIx, and AGLx are products designed by Agito Akribis Systems Ltd. in Israel.

Sales of the products are licensed to Akribis Systems Pte Ltd. under intercompany license agreement.

Agito Akribis Systems Ltd. has full rights to distribute above products worldwide.

Disclaimer

This product documentation was accurate and reliable at the time of its release.

Agito Akribis Systems Ltd. reserves the right to change the specifications of the product described in this manual without notice at any time.

Trademarks

Agito PCSuite is a trademark of Agito Akribis Systems Ltd.

Contents

1 About this Application Note

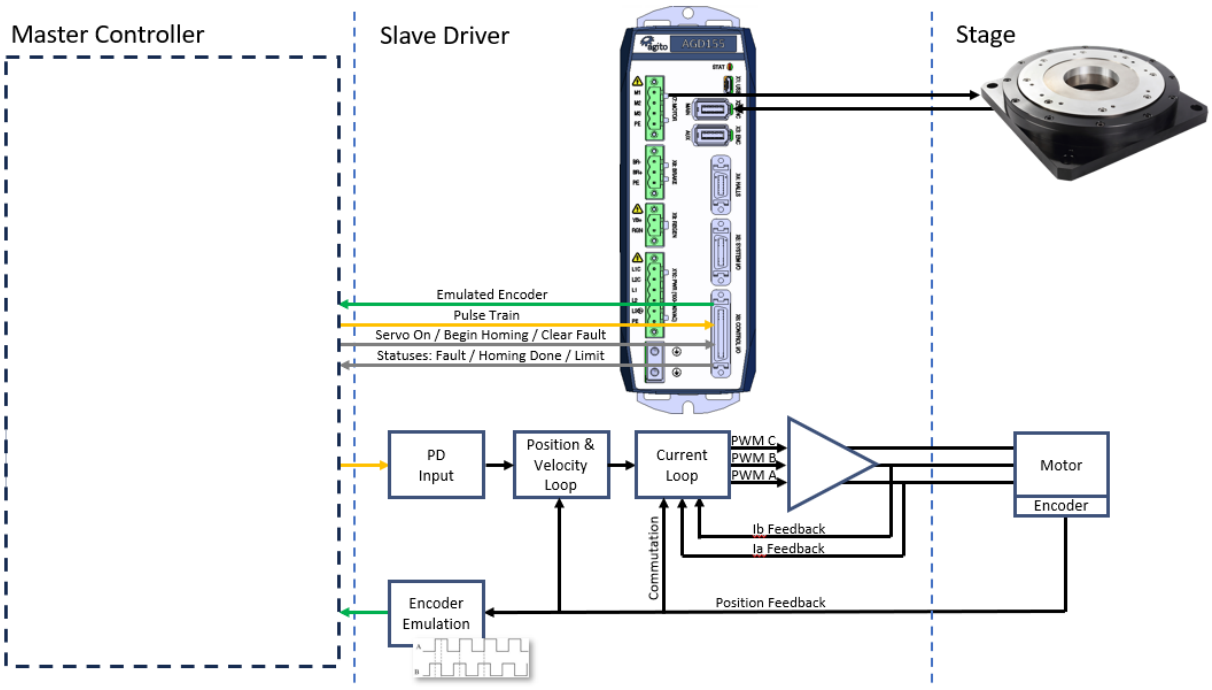
This application note explains how to setup and configure the Agito driver as a slave that receives pulse and direction position command. And to emulate the position feedback back to a 3rd party master controller.

The following topics will be covered in this document,

- PD Input wiring interface summary for Agito products
- IO Mode functionality for communication a 3rd party master controller
- PCSuite configuration steps and settings related to PD Input
- Motion and observation
- Relevant keywords summary

1.1 Introduction to Pulse & Direction

The Pulse & Direction command is a traditional method to decouple the controller that does the profiling, from the amplifier/driver that drives the motor. With this, controllers with profiling algorithm for specific applications (E.g., CNC Machine or Laser cutter) can be used with any stage system that supports this mode of command. Therefore, allowing for flexibility in pairing of drivers of different sizes and types (E.g., 3-phase servo/2-phase stepper/5-phase stepper drives).



Topology of Pulse & Direction Control

2 Wiring Interface

The PD Input functionality is product dependent. The following tables depict how the PD Input functionality is assigned to the inputs.

These are partial wiring descriptions, be sure to wire up other relevant lines such as ground for the differential IO.

Please refer to the relevant Product Manual for more information regarding the hardware and circuitry.

2.1 AGD101 Pinouts

For AGD101, the PD Input functionality is tied to specific inputs.

Input No.	Type	Connector	Function
7 (Diff IO 3)	Differential	X6: Pin 5,6	Pulse Input
8 (Diff IO 4)	Differential	X6: Pin 7,8	Direction Input

For AGD101, the emulated encoder functionality is tied to specific outputs.

Output No.	Type	Connector	Function
5 (Diff IO 1)	Differential	X6: Pin 1,2	AqB, Line A
6 (Diff IO 2)	Differential	X6: Pin 3,4	AqB, Line B

2.2 AGD155 Pinouts

For AGD155, the PD Input functionality is tied to specific inputs.

Input No.	Type	Connector	Function
17 (Diff In 1)	Differential	X6: Pin 24,25	Pulse Input
18 (Diff In 2)	Differential	X6: Pin 8,9	Direction Input

For AGD155, the emulated encoder functionality is tied to specific outputs.

Output No.	Type	Connector	Function
7 (Diff Out 1)	Differential	X6: Pin 1,2	AqB, Line A
8 (Diff Out 2)	Differential	X6: Pin 19,20	AqB, Line B

2.3 AGD200 Pinouts

For AGD200, the PD Input functionality is tied to specific inputs.

Input No.	Type	Connector	Function
15 (Diff In 1)	Differential	X10: Pin 5,6	A Axis – Pulse Input
16 (Diff In 2)	Differential	X10: Pin 7,8	A Axis – Direction Input
17 (Diff In 3)	Differential	X10: Pin 9,10	B Axis – Pulse Input
18 (Diff In 4)	Differential	X10: Pin 11,12	B Axis – Direction Input

For AGD200, emulated encoder functionality is tied to specific outputs.

Output No.	Type	Connector	Function
5 (Diff Out 5)	Differential	X10: Pin 21,22	A Axis – AqB, Line A
6 (Diff Out 6)	Differential	X10: Pin 23,24	A Axis – AqB, Line B
7 (Diff Out 7)	Differential	X10: Pin 25,26	B Axis – AqB, Line A
8 (Diff Out 8)	Differential	X10: Pin 27,28	B Axis – AqB, Line B

2.4 AGD301 Pinouts

For AGD301, the PD Input functionality is tied to specific inputs.

Input No.	Type	Connector	Function
28 (Diff IO 1)	Differential	X4: Pin 1,2	A Axis – Pulse Input
29 (Diff IO 2)	Differential	X4: Pin 19,20	A Axis – Direction Input
30 (Diff IO 3)	Differential	X4: Pin 3,4	B Axis – Pulse Input
31 (Diff IO 4)	Differential	X4: Pin 21,22	B Axis – Direction Input
32 (Diff IO 5)	Differential	X4: Pin 25,26	C Axis – Pulse Input
33 (Diff IO 6)	Differential	X4: Pin 9,10	C Axis – Direction Input

For AGD301, the differential IO's are bi-directional and the PD input functionality overlaps with the PD Output functionality. As such, emulated encoder is not supported for AGD301 while the PD input feature is used.

3 Configuration Setup (via PCSuite)

3.1 PD Input Settings

The steps below explain how to configure the driver for PD Input operation.

- Under the 

tab, navigate to the 

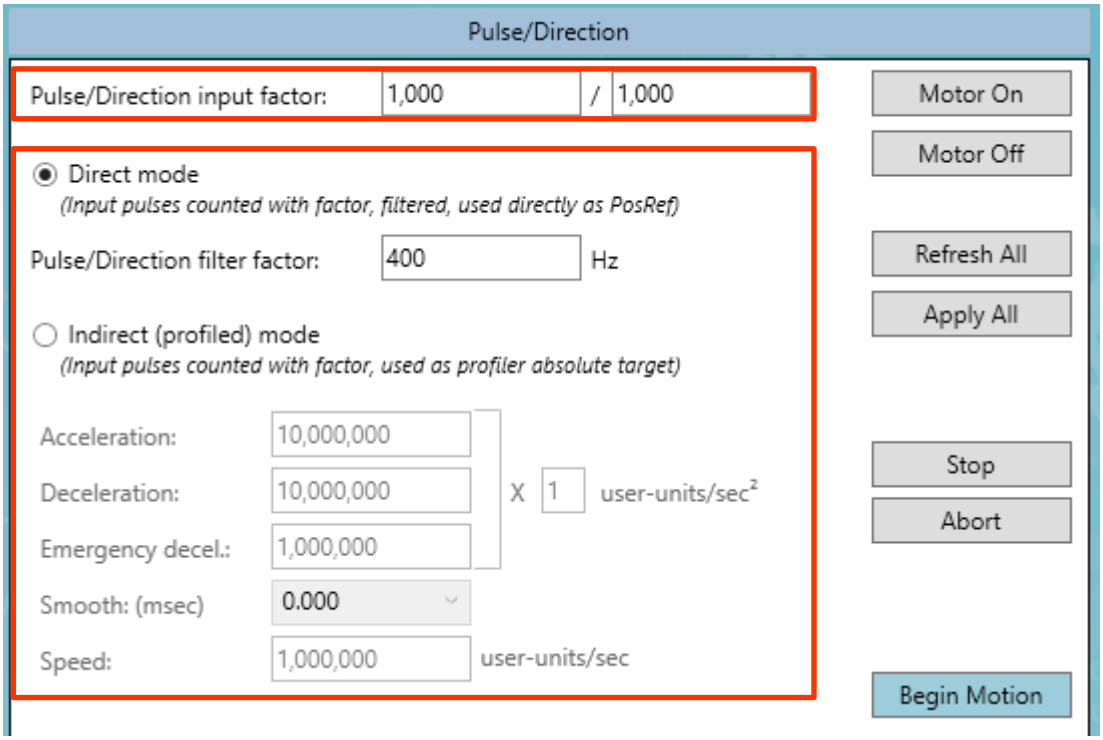
page.

  Configure the PD Input factor as necessary to scale the command coming from the master.

  Select “Direct Mode” or “Indirect Mode” as necessary.

  In direct mode, the driver does not perform any motion profiling; it will use the profile generated by the master. A low-pass filter is provided to smoothen the profile.

  In indirect mode, the driver will limit the acceleration and velocity to the values set by the user.



3.2 Emulated Encoder

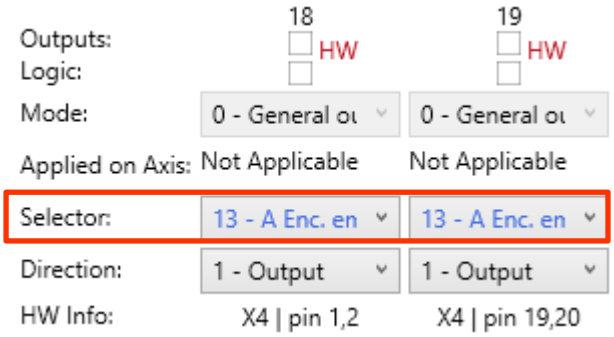
The steps below explain how to configure the driver to emulate the current position as feedback to the master controller.

- Under the 

tab, navigate to the 

page, Discrete Outputs tab.

  Configure the selector of the outputs to **“13 – x Enc. Emul. Line x”**.



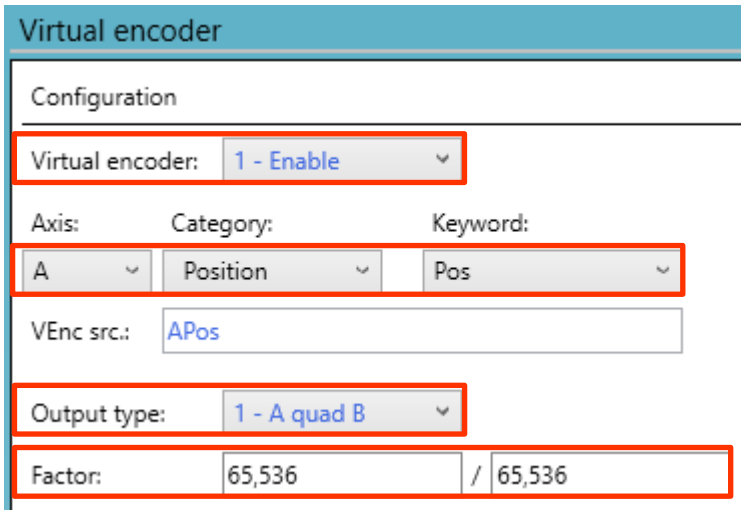
- Under the 

tab, navigate to the 

page. Configure the virtual encoder source to use “APos”, or “BPos” if you are configuring for B axis. The virtual encoder feature is used emulate the current position (Pos).

  Configure the scaling factor as necessary to convert the output from user-units to the preferred units.

  Configure the output type as necessary. PD Output and AqB are supported types.



- If the encoder is of incremental type, hardware emulation is supported. Hardware emulation will not have the 61us delay that is inherent in the virtual encoder feature. Nevertheless, the 61us delay will not affect performance as the system is in open-loop, and therefore of little concern.

  Hardware emulation can be scaled by configuring the keyword EmulRat. The emulation can only be scaled down. If EmulRat is set to 8, then 1 pulse will be sent every 8 counts.

  Disabling the virtual encoder feature puts the controller in hardware emulation mode.

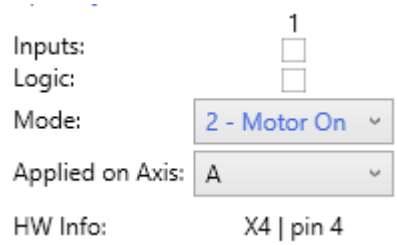
3.3 IO Mode Functionalities

The steps below explain how to setup the IO's to communicate the slave driver's status and to read the master controller's commands.

- Under the I/O tab, navigate to the DIG page, Discrete Inputs tab. Wire and configure the input signals as necessary.

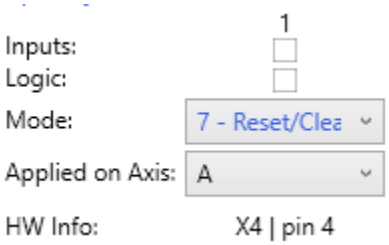
Motor On & Begin trigger – upon the rising edge of this input, the driver will do motor on and begin motion (if in PD Mode).

This signal is usually connected to the **Motor On status** output on the master controller side. When the master controller is enabled, the output will go high. In turn, the slave driver will be triggered to do motor-on and begin.



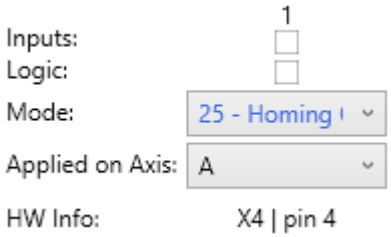
Reset/Clear Fault/Alarm trigger – upon rising edge of this input, the slave driver will clear its fault status.

This signal is usually connected to the **Fault status** output (with inverted logic) from the master controller. When a fault occurs, the output on the master controller will drop low due to the inversion of logic. When the error is cleared on the master controller, the signal will go high. In turn, the slave driver will be triggered to reset the faults on its side.



Homing On trigger – upon rising edge of this input, the homing sequence will start.

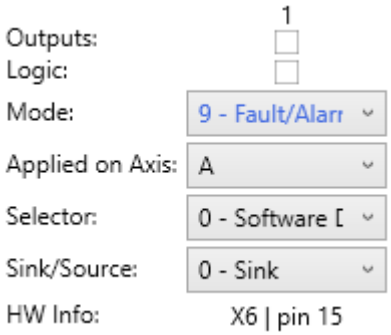
This signal is usually connected to the general output from the master controller.



- Under the I/O tab, navigate to the DIG page, Discrete Outputs tab. Wire and configure the output signals as necessary.

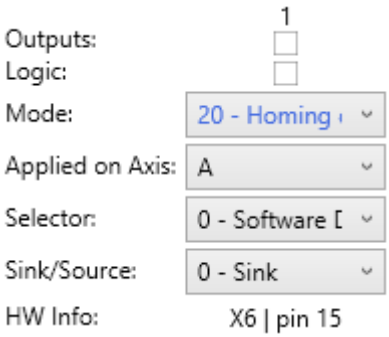
Fault/Alarm status – if there is a fault, the output will be high.

This signal is usually connected to the **Fault (ext. drive) trigger** on the master controller. If a fault occurs on the slave drive, the master controller will be alerted.



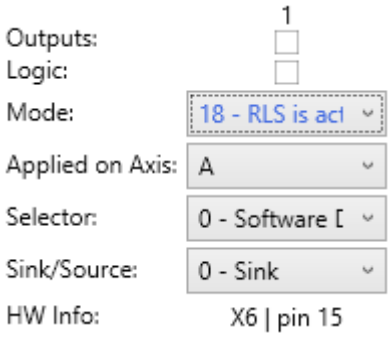
Homing Done status – if homing is successfully completed, the output will be high.

This signal is usually connected to a general input on the master controller to alert it when homing is done and it can begin PD motion.



Limit is Active status– if limit is active, the output will be high.

Limit sensors are usually connected to the slave drive for homing purposes. This signal is used to emulate the FLS/RLS signal to the master controller.



Note

Some of the discrete inputs/outputs' special functionalities assignments ("Mode") will operate properly only after power off/on or reset of the controller. This is mainly relevant for the case of turning off a special functionality. It might be still functional until the controller is powered off/on or reset.

Once setting up all special functionalities of the discrete inputs/outputs, apply them, save the new settings to the Flash, and consider resetting or power-cycling the controller.

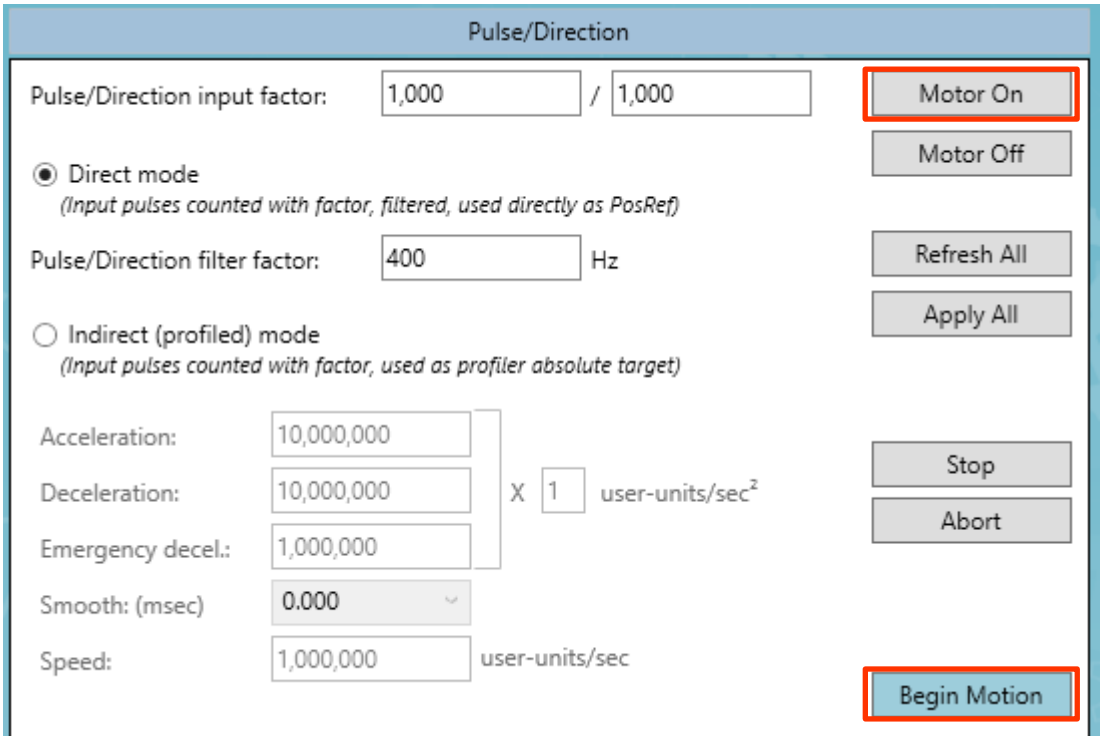
Note

There is no need to configure DInMode to specify the inputs as PD Inputs. The driver knows to which inputs to use as these are pre-defined and tied to the hardware.

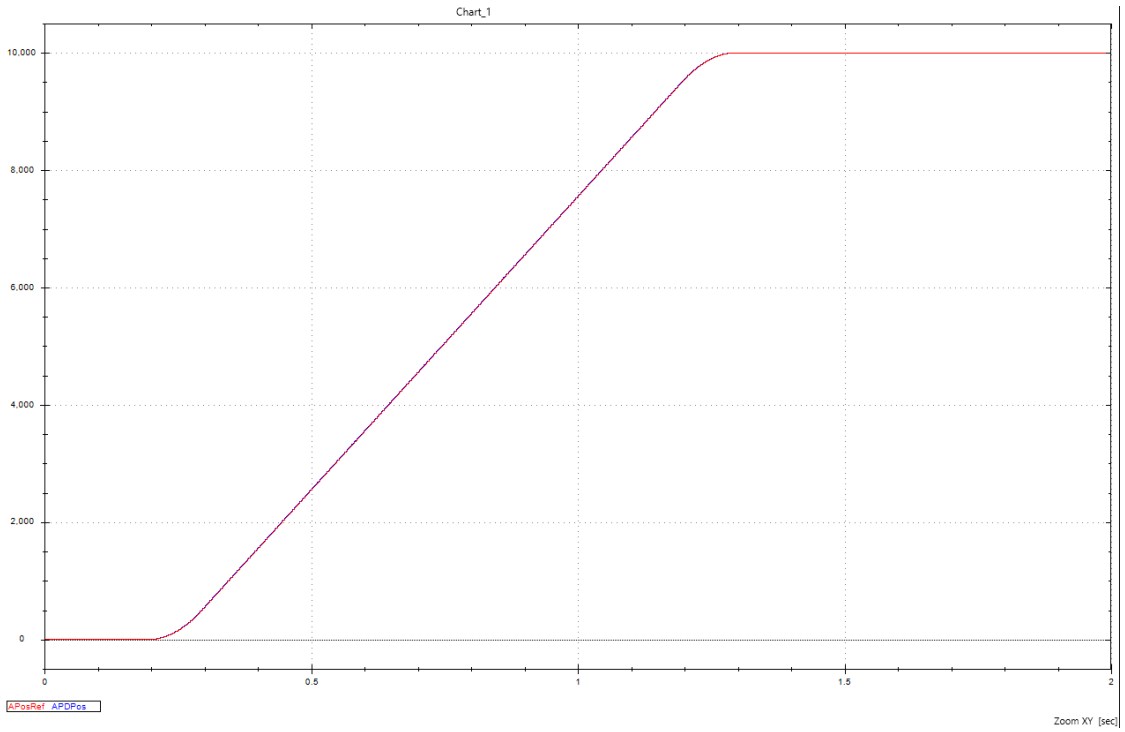
3.4 Motion

The step below explains how to execute a motion in PD Input mode (3rd party amplifier).

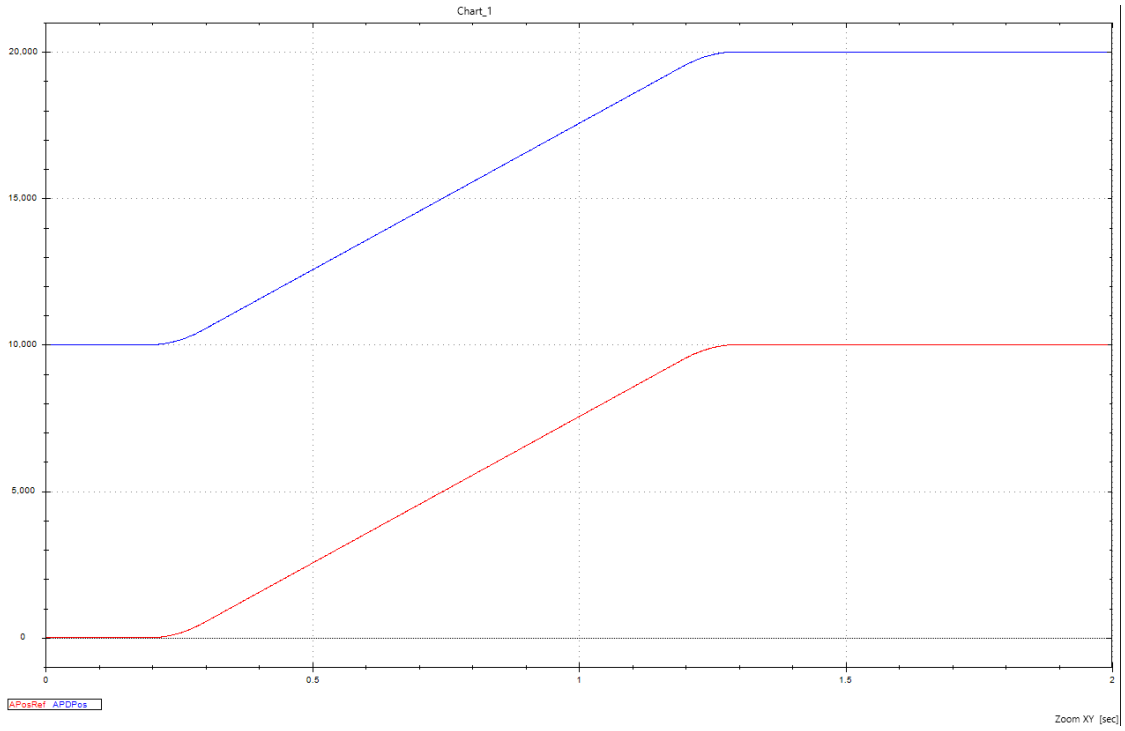
- Under the MOTION tab, navigate to the PD page. Do motor-on and begin. Once the master controller starts issuing pulses, the motor will start moving.



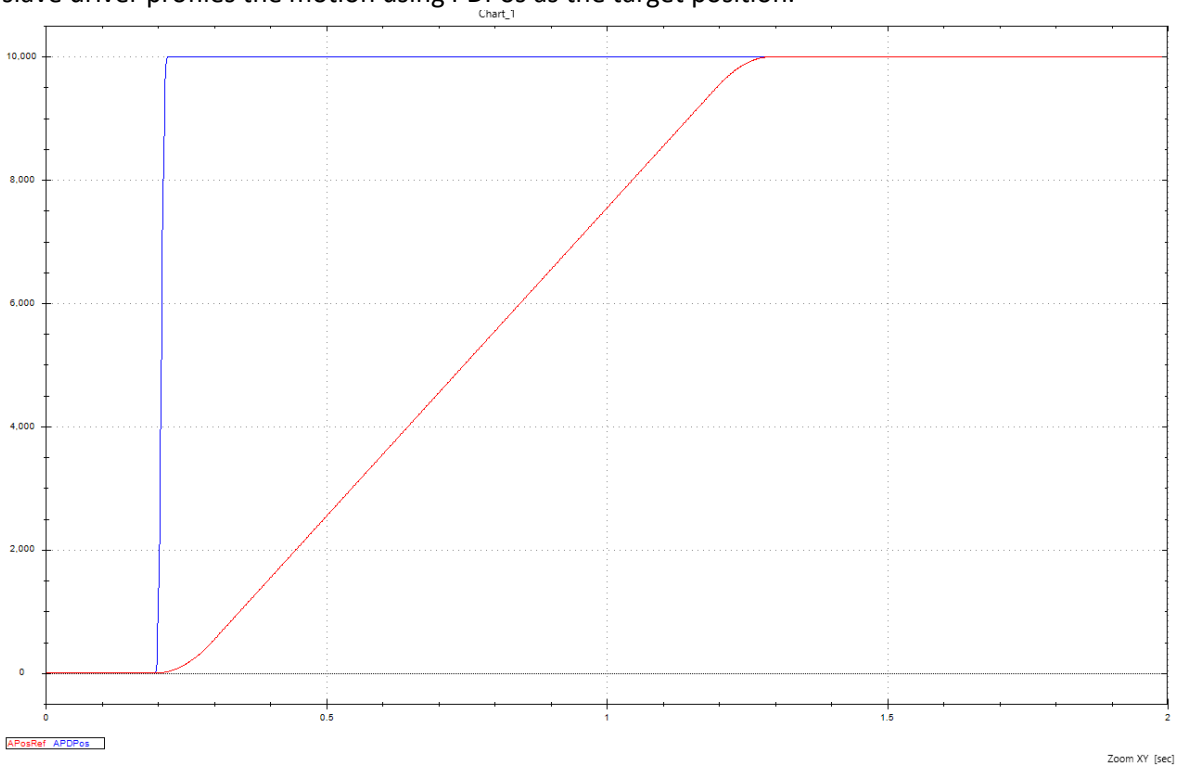
The graph below depicts the parameters PDPos (blue) and PDPosRef (red) while running in direct mode. PDPos records the number of pulses received by the driver since power on. PDPosRef is lagging behind PDPos by 61us; the profiler takes the delta of the input pulses of n'th interrupt and uses it to generate the profile for the n+1 interrupt.



In event that the motor is turned off, PosRef resets to 0 while PDPos remains at whatever value it was at. The graph below shows an offset of 10,000 counts between PDPos and PosRef, this depicts the PDPos had already received 10,000 pulses before the motor was turned on.



In indirect mode, the motion profile (PosRef – red) follows the acceleration and speed limitations set by the user. In the graph below, the pulse and direction command is rapidly fired (PDPos – blue). The slave driver profiles the motion using PDPos as the target position.



4 PD Input Limitations

The differential input is limited to a maximum frequency of 5MHz. Using PD input format, the speed is limited to 5,000,000 counts/s. If AqB format is used, the speed limitation will be increased by a factor of 4 to 20,000,000 counts/s. AqB format (in context of pulse and direction input command) is a future feature.

While internal tests have shown that it is possible to operate the feature at higher speeds of up to 9,990,144Hz, it is recommended to use output frequencies up to 5MHz.

5 Indefinite Motion In A Direction Limitations

Agito controllers work using signed 32-bit integers. As such, in many cases, position information is also limited to the range of -2,147,483,648 to 2,147,483,647. In general, roll-over is not supported with certain exceptions.

PD Mode may support position rollover whilst in “direct mode”. This allows the user to command the driver indefinitely in one direction. Rollover is also supported for emulation of the position via virtual encoder feature.

To allow the position to rollover, the forward and reverse software position limits “FwdPLim”, “RevPLim” should be set to the largest magnitudes 2,147,483,647 and -2,147,483,648 respectively.

6 PD Input Related Keywords

Keyword	Description
VEncOn	VEncOn enables/disables the virtual encoder feature. **VEncOn = 0** disables the virtual encoder feature. **VEncOn = 1** enables the virtual encoder feature. To emulate the position feedback, virtual encoder feature shall be enabled. Alternatively, if the encoder is of incremental type, it is possible to use hardware emulation instead. In this case virtual encoder feature shall be disabled.
VEncSrc	VEncSrc specifies the keyword/parameter to emulate. **VEncSrc = Complex CAN Code (CCC)** of parameter to be emulated. To use the motion controller in PD Output mode, VEncSrc is assigned the CCC of PosRef for the relevant axis. For ease, use PCSuite to compose the CCC. Alternatively, refer to the CCC Definition document for more information on how to compose CCC manually.
VEncType	VEncType specifies the output format of the virtual encoder feature. **VEncType = 0** sets the sub-type to **pulse and direction (PD)** mode. **VEncType = 1** sets the sub-type to **quadrature (AqB)** mode. Typically, pulse and direction mode is used. However, quadrature mode supports four times the max frequency of pulse and direction mode.
VEncFact	VEncFact scales (multiplies) the emulated output by a factor. **Default value:** **65,536** Max value: 16,777,215 Min value: -16,777,215 (A negative value inverts the direction)
VEncFactDen	VEncFactDen scales (divides) the emulated output by a factor. **Default value:** **65,536** Max value: 500,000,000 Min value: 1
VEncDelay	VEncDelay is applicable only if VEncType is set to pulse and direction (PD) mode. VEncDelay specifies the delay (in micro-seconds) to add in between the switching of directions. **Default value:** **0** Max value: 25 Min value: 0

PD Input related Keywords

Keyword	Description
PDFact	PDFact scales (multiplies) the input command by a factor. **Default value:** **1000** Max value: 16,777,215 Min value: -16,777,215 (A negative value inverts the direction)
PDFactDen	PDFactDen scales (divides) the input command by a factor. **Default value:** **1000** Max value: 16,777,215 Min value: 1
MotionMode	MotionMode specifies the way the motor will be commanded. **Motion Mode = 3** sets the mode to **PD Direct** mode. In direct mode, the drive follows the profile generated by the master. **Motion Mode = 4** sets the mode to **PD Indirect (Profiled)** mode. In indirect mode, the drive limits the profile to the specified acceleration and speed values.
PDPosFilt	PDPosFIlt is applicable on if MotionMode is set to PD Direct mode. In Direct Mode, PDPosFilt specifies the frequency for a first order filter (in 0.01Hz units) to “smoothen out” the command from the master controller. **Default value:** **12,800** Max value: 2,147,483,647 Min value: 4,150
PDPos	PDPos reports the pulse direction input position command reading in PD user units (PDUsrUnits). If PDUsrUnits = 1, then PDPos is in pulses. The value of PDPos is 0 upon reset. PDPos counts the position between -2,147,483,648 to 2,147,483,647. If the command input from PDPos exceeds these limits, the position command reading will roll. PDUsrUnits is used only to report the number of pulses that were counted. It is also possible to multiply the number on input pulses by a factor, and the result of the multiplication is used as the reference. **Default value:** **0 (upon power cycle)** Max value: 2,147,483,647 Min value: -2,147,483,648