

PMX Support FAQ

USB Support

What is the USB communication speed?

The time interval between sending and command from a PC and receiving a response is 1-3ms. This applies to most commands. Commands dealing with motor driver configuration or writing to flash memory will have a longer response time. These select commands will have response time ranging from 1-2 seconds.

USB communication is out of sync. What is the problem?

There may be a few different causes to this issue. Check each item below:

- **Buffer Flushing:** If USB communication begins from an unstable state (i.e. your application has closed unexpectedly), it is recommended to first flush the USB buffers of both the PC and the USB device. See the following function prototype below:

BOOL fnPerformaxComFlush(IN HANDLE pHandle)

- **USB Cable:** Another source of USB communication issues may come from the USB cable. Confirm that the USB cable being used has a noise suppression choke.
- **Time-outs:** If the timeout parameter for USB communication on your PC is either set too large or too small, out of sync communication may occur. The function prototype for the timeout function is below. Units are in milli-seconds and the recommended time-out value is 1000 milli-seconds.

BOOL fnPerformaxComSetTimeouts(IN DWORD dwReadTimeout, DWORD dwWriteTimeout)

USB driver installation is freezing. What is the problem?

Some systems will freeze during the USB driver installation, specifically during the transfer of the SiLib.sys file. This is likely do to the anti-virus protection on the PC. Disabling or removing the anti-virus software may be required for the USB driver to complete installation.

Unable to uninstall USB driver.

Sometimes because of anti-virus programs, certain batch files are not installed correctly during the initial install process. Under this condition, you will receive an error when trying to uninstall. See attached screen shot for an example.

To fix this, download the dummy batch file called: "PerformaxComDLLUninstall.bat" and place it in the directory shown in the error.

In this example the directory is "C:\Program Files\Arcus Technology\Drivers, Libraries, Source\Performax USB v1.30\"

Note that you will need to remove the actual batch file from the zip folder.

After this is complete, try uninstalling again.

I have occasional communication errors over USB. What is the problem?

The first thing to check is the USB cable. Because of the bandwidth of the data being transferred over the USB, a standard USB cable will drop messages every so often. This will lead to communication errors. Therefore, confirm that the USB cable being used has a noise suppression choke. See the photo below. A shielded cable is also advantageous.



Arcus Technology offers filtered USB cables, which we typically recommend to our customers on their first purchase. The part numbers are:

- CBL-USB-L3-S (A to B, 3 ft)
- CBL-USB-L3-M (A to mini-B, 3ft)

How to manually install USB driver

On the event that the Arcus USB driver is not automatically installed by a Windows system, the USB driver may need to be manually pointed to the correct file location. The following steps can be used for a Windows system.

1. Connect the USB device to the computer and verify that the device is powered.
2. Open the Window Device Manager
3. Find the USB device. It may be in a drop down menu for "Unknown Devices" or it may be the in the "Universal Serial Bus Controller" section with a warning.
4. Right click the USB device and select "Update Driver Software"
5. Select "Browse my computer for driver software"
6. Browse to the following file location
 - C:\Program Files (x86)\Arcus Technology\Drivers, Libraries, Source\Performax USB v4.01

- Check the "Include subfolders" check box
- Click next and the USB driver should install

Serial Communication Support

I am having serial communication problems. What is the problem?

There may be a few different causes to this issue. Check each item below:

- **RS-485/RS-232:** Check that your system is configured for the right type of serial communication.
 - PC Control: In most cases, a converter will be necessary for RS-485 communication.
 - PLC Control: Many PLCs have RS-485 or RS-232 communication built in.
- **Serial Port Configuration:** Check that the settings below match with your PC/PLC.
 - Baud Rate: Make sure to match with the device.
 - Byte Size: 8 bits
 - Parity: None
 - Flow Control: None
 - Stop Bit: 1
- **Device Address:** Check that the commands that are being sent are pre-pended with the correct device address. Check product manual for details.
- **RS-485 Wirings:**
 - Termination: For an RS-485 network, it may be required that a 120 Ohm resistor is placed at the beginning and end of the bus. Note that on short communication buses, or buses with a small number of nodes, termination resistors may not be needed.
 - Twisted Pair Wiring: To reduce noise, it is recommended to use twisted pair wiring for the 485+ and 485- lines.
 - Daisy chaining: For a multi-drop RS-485 network, be sure that the network uses daisy-chain wiring.
 - Number of Nodes: The maximum number of nodes recommended is 32. Increasing beyond this number will require special attention.

Digital Input/Output Support

How do I choose a digital input sensor?

Limit and Home sensors are split into two main groups:

- A NPN sensor is one that provides a sinking output. In other words, when the sensor is triggered, the output is sunk to GND. However, in the un-triggered state, the sensor outputs the high voltage.
- A PNP sensor operates opposite of an NPN sensor.

Most Arcus input opto-isolators trigger when the input is shorted to GND of the opto-supply. Therefore, a NPN sensor is the correct choice.

Other issues to note are the maximum sink/source current and voltage allowed. This will need to be addressed by the choice of the voltage supply and the current limiting resistor (if necessary).

How do I wire a digital output?

Outputs are split into two main groups:

- NPN outputs are open collector. In this case, an external supply and pull-up resistor must be supplied.
- PNP outputs are open-emitter. In this case, an external ground and pull-down resistor must be supplied.

The voltage supply and resistor value will depend on the current and voltage specification of the output.

An issue that users run into is that the voltage difference between the high and low state on the output is not high enough. This is due to the choice of the resistor value. Note that the voltage drop across the output will increase or decrease depending on the resistor ($V=IR$).

Encoder Support

How do I choose an encoder?

To add a close-loop position verification to your system, an encoder must be chosen. For the models that do not have an encoder integrated, it is important to choose the right encoder.

- **Absolute vs. Incremental:** An absolute encoder will provide absolute position even after power down. An incremental encoder keeps track of position but needs a point of reference. Arcus products only support incremental encoders.
- **A,B,Z:** The 3 signals used for incremental encoders are A, B, and Z. A and B provide the direction and incremental position moved. The Z signals provides a single pulse once per revolution. This can be used to home to a specific location within the 360 degrees of the motor.
 - There are A, B output encoders, as well as A, B, Z output encoders. If you do not need a Z index, an A, B output encoder will suffice.
- **Output Circuitry:** Arcus products accept 0 to 5V TTL signals.
- **Differential vs. Single-ended:** Encoders are available in differential or single-ended options. Differential output are necessary if your encoder signal is to travel long distances or in a noisy environment. Before choosing a differential encoder, be sure to check that your product supports differential encoder signals.
- **Resolution:** The resolution specifications for an incremental encoder is counts per revolution (CPR). Note that the resolution specification is known as the native resolution. The controller has the ability to interpolate the encoder signals to multiply the effective resolution.

Arcus products provide up to 4X decoding. For example, if a 1000 CPR encoder is selected, the Arcus product can interpolate the signal to $1000 \times 4 = 4000$ CPR.

What is StepNLoop closed-loop control?

StepNLoop closed loop control is an algorithm that Arcus has developed that will provide the user an easy way to add position feedback verification to a stepper system. Note that StepNLoop is not a real-time closed-loop system. In other words, position and speed is not corrected on-the-fly. Instead, position correction is performed after a move has been completed.

I am having trouble finding the correct StepNLoop ratio. What can I do?

The StepNLoop ratio is the parameter used by the StepNLoop algorithm to determine the relationship between the pulse count and encoder count. See the formula below:

StepNLoop Ratio = Pulse per revolution / Encoder counts per revolution

If you have trouble finding the StepNLoop ratio, the following simple test can be performed.

1. Disable StepNLoop
2. Set encoder position to 0
3. Set pulse position to 0
4. Move the axis 100 pulses
5. Get the resulting encoder value for the axis [ENC] (after the move)
6. StepNLoop ratio is : 1000/ENC

Motor/Drive Support

What is the difference between a servo and stepper motor?

The main difference between a servo motor and a stepper motor is that a stepper motor allows the motor to "lock" into place. A servo motor does not allow this; instead a position is held by closed-loop feedback. Because of this difference, the stepper motor provides more holding and low speed (<1000 RPM) torque.

Servo motors will provide less attenuation of torque at high speeds, however begin with a low holding torque. In systems where high precision closed-loop (speed and position) and high speed operation (>3000 RPM) is needed, a servo motor is appropriate.

The stepper motor, however, also provides a very viable solution. Low speed torque, cost-effectiveness and simplicity of the stepper motor systems make a good option for many applications.

What factors determine torque of a motor/driver combination?

A proper torque specification provides the desired torque and speed of your motor. Once this is established, torque information can be found by looking at torque curves that are provided with motor/driver products. Below are a few main factors that determine torque:

- **Current:** Increasing current will increase the torque output of the motor. Be sure to check the maximum current value of your unit.
- **Voltage:** Increasing voltage will allow your motor to have high torque at high speeds. Be sure to check maximum voltage value for your unit.
- **Motor Frame Size:** Choosing a larger motor will provide more torque. Arcus provides size NEMA 11, 17, and 23 frame motors.
- **Motor Stack Size:** Within a frame size, stepper motors come in different stack sizes. Torque increases with increasing stack size.

How do I prevent damage?

See below for stepper driver precautions:

- **Voltage:** Be sure to keep the voltage input below the maximum specifications. A regulated voltage supply is recommended; voltage spikes may damage the motor.
- **Current:** Keep the driver current below the specification of the motor. This will prevent over-heating which may damage the driver.
- **Motor:** If the motor is plugged into or unplugged from the driver while the driver is powered, it is likely that the driver will be damaged.

My stepper motor driver only moves in one direction. What is the problem?

See below for possible solutions:

- **1-CLK vs 2-CLK:** Make sure that the 1-CLK/2-CLK setting matches your pulse/direction input signals. 1-CLK is the US standard which uses one pulse input and one direction input. 2-CLK, which is more popular in Asia and Europe, uses two pulse signals. If the setting is incorrect, the driver will not operate properly. This setting can be made via the driver configurator.
- **Signal Wiring:** Make sure that the pulse, direction and opto-supply signals are wired correctly. Not that different wiring schemes are necessary depending on a single-ended or differential input driver.

What flags do I use to tell when the motor has completed a move?

All Arcus controller use the motor status command [**MST**] to determine whether the motor is moving or not. Typically, the first three bits of the motor status signify that the motor is accelerating, decelerating, or at constant speed, respectively. Therefore **MST&7** can be used to determine whether the motor is still moving. For a more detailed and specific description of the motor status, please refer to the product manual.

How can I tell if the motor has stopped moving?

If a controller is outputting pulses, thus moving the motor, it will be shown in the MST command. The MST command will respond with the current status of the controller. The first three bits of the response will let you determine if the motor is moving or not. See below for the meaning of the first three bit of the MST command:

Bit 0: Motor is moving at constant speed

Bit 1: Motor is accelerating

Bit 2: Motor is decelerating

Therefore, the following pseudo-code will demonstrate how to check if the motor is moving and ready to accept another move command:

```
//MOVE command sent to the controller
```

```
WHILE (MST&7 != 0) {
```

```
    //LOOP until MST is cleared
```

```
}
```

```
//MOVE command sent to the motor
```

PMX Software Support

What software comes with the purchase of a PMX?

The PMX comes with all the necessary software tools to begin evaluation and development. All software is available for download from the website free of charge.

How do I create a custom software application?

Many customers require the need to create a custom software application to interface with an PMX product. Sample source code is available to help ease the implementation of custom software. All source code is available for download from the company website free of charge.

How do I match firmware/software/manual for a product?

As each manual is updated to a new version, the corresponding firmware and software version is listed in the beginning of the document. This allows users to match a manual to the correct firmware and software for a product. Along with this information, errata as well as firmware /software update information is available for download form the company website.

I need customization (hardware/software/firmware). Is it possible?

Nippon Pulse is flexible and willing to make customization when appropriate. This includes high volume, cost-sensitive projects, but does not exclude lower volume projects. We will treat each customization situation case by case.

My standalone program gets stuck on a move command.

Standalone programs will not process a move command if the axis is already moving. This can create problems when the JOG function is used. Many times the following program is written:

```
WHILE 1=1
  IF DI1=1
    JOGX+
  ELSE
    STOPX
  ENDIF
ENDWHILE
```

This becomes a problem because, if digital input 1 is triggered for too long, multiple JOG commands can be sent. Once the second JOG command is issued, the standalone program will wait for motion to stop before proceeding. This prevents the standalone program from issuing the STOP command. To resolve this issue, a program like the following should be used:

```
V1=0
WHILE 1=1
  IF DI1=1
    IF V1=0
      JOGX+
      V1=1
    ENDIF
  ELSE
    STOPX
    V1=0
  ENDIF
ENDWHILE
```

This program will prevent the standalone program from issuing multiple JOG commands between STOP commands.

I'm running PMX software and getting a run-time error "383."

In this situation, your operating system is not allowing you to access a text file that is required for operation. To solve this, right click on the software icon that you are trying to run and click "Run As Administrator".