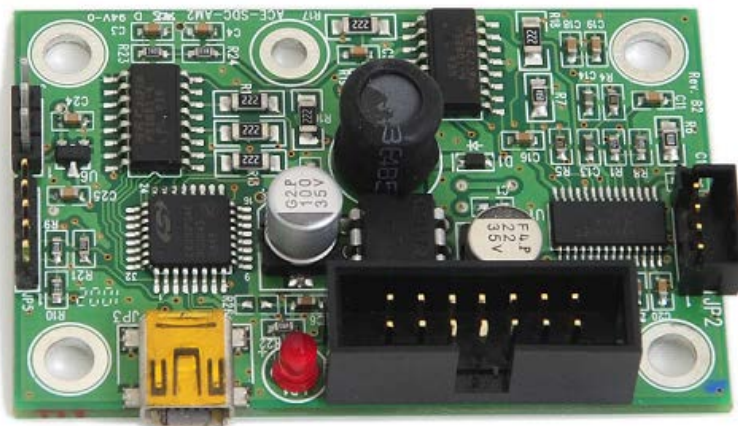


# ACE-SDC-V3

## Board Level Controller + Microstep Driver USB 2.0 Communication



---

COPYRIGHT © 2015 ARCUS,  
ALL RIGHTS RESERVED

First edition, April 2008

ARCUS TECHNOLOGY copyrights this document. You may not reproduce or translate into any language in any form and means any part of this publication without the written permission from ARCUS.

ARCUS makes no representations or warranties regarding the content of this document. We reserve the right to revise this document any time without notice and obligation.

**Revision History:**

- 1.00 – 1<sup>st</sup> Release
- 1.01 – 2<sup>nd</sup> Release
- 1.02 – 3<sup>rd</sup> Release
- 1.04 – 4<sup>th</sup> Release
- 1.05 – 5<sup>th</sup> Release
- 1.07 – 6<sup>th</sup> Release
- 1.08 – 7<sup>th</sup> Release

**Firmware Compatibility:**

†V212BL

†If your module's firmware version number is less than the listed value, contact Arcus for the appropriate documentation. Arcus reserves the right to change the firmware without notice.

# Table of Contents

<b>1. Introduction</b> .....	<b>5</b>
1.1 Features .....	5
<b>2. Electrical and Thermal Specifications</b> .....	<b>6</b>
<b>3. Dimensions</b> .....	<b>7</b>
<b>4. Connectivity</b> .....	<b>8</b>
4.1. JP1 3-Pin Connector Pin-outs .....	8
4.2. JP2 4-Pin Motor Connector Pin-outs.....	8
4.3. JP4 14-Pin IO Connector Pin-outs .....	9
4.4. Internal Interface Circuit .....	10
4.5. Digital Inputs.....	11
4.6. Digital Outputs.....	12
4.7. Analog Input .....	12
<b>5. Stepper Motor Driver Overview</b> .....	<b>13</b>
5.1. Microstep .....	13
5.2. Driver Current.....	13
<b>6. Communication Interface</b> .....	<b>14</b>
6.1. USB Communication .....	14
6.1.1. Typical USB Setup.....	14
6.1.2. USB Communication API.....	14
6.1.3. USB Communication Issues .....	16
6.2. Device Number.....	16
6.3. Windows GUI .....	16
<b>7. General Operation Overview</b> .....	<b>17</b>
7.1. Motion Profile .....	17
7.2. On-The-Fly Speed Change .....	17
7.3. Motor Position .....	18
7.4. Motor Power .....	18
7.5. Jog Move.....	18
7.6. Stopping .....	18
7.7. Motor Status .....	18
7.8. Digital Inputs / Outputs .....	19
7.8.1. Digital Inputs .....	19
7.8.2. Digital Outputs .....	19
7.9. Analog Speed Control .....	20
7.9.1. Current Used For Analog Speed Control.....	20
7.9.2. Digital IO Used For Analog Speed Control.....	21
7.10. Polarity .....	21
7.11. Storing to Flash .....	22
<b>8. Software Overview</b> .....	<b>23</b>
8.1. Main Control Screen .....	24
8.1.1. Status.....	24
8.1.2. Control .....	25
8.1.3. Digital Input.....	25
8.1.4. Digital Output .....	26
8.1.5. DO Boot .....	26
8.1.6. Terminal .....	27
8.1.7. Microstep .....	27
8.1.8. Upload / Download / Store to Flash .....	28
8.1.9. Control Mode .....	28
8.1.10. Polarity .....	29
8.1.11. Idle Current .....	29
8.1.12. DIO Control Mode Settings.....	30
8.1.13. Product Information.....	31

---

<b>9. ASCII Language Specification.....</b>	<b>32</b>
9.1 ASCII Command Set.....	32
9.2. Error Codes .....	34

---

## 1. Introduction

ACE-SDC-V3 is a single board cost-effective stepper driver and controller with USB 2.0 communication.

### 1.1 Features

- USB 2.0 Communication
- 12-24VDC voltage input
- Different configurable current settings from 0 mA to 2.0A peak current (100 mA resolution)
- Selectable Full, 1/2, 1/4, 16 microstep
- 6 Opto-isolated Digital inputs
- 2 Opto-isolated Digital outputs
- 1 Analog input
- Three different control modes:
  - DIO Control – motion is done using the digital inputs and outputs.
  - USB Control – motion is done from PC using USB communication.
  - Analog Control – speed control is done from analog input.

For technical support contact: [support@arcus-technology.com](mailto:support@arcus-technology.com)

Or, contact your local distributor for technical support.

## 2. Electrical and Thermal Specifications

Parameter	Min	Max	Units
Main Power Input <sub>1</sub>	+12	+24	V
	-	2	A
Opto-supply Power Input	+12	+24	V
Digital Input Forward Diode Current	-	50	mA
Digital Output Emitter Current	-	50	mA
Analog Input Range	0	+3.3	V
Operating Temperature <sub>2</sub>	-	+85	°C

Table 2.0

<sub>1</sub>The supply current should match the driver current setting.

<sub>2</sub>Based on component ratings.

### 3. Dimensions

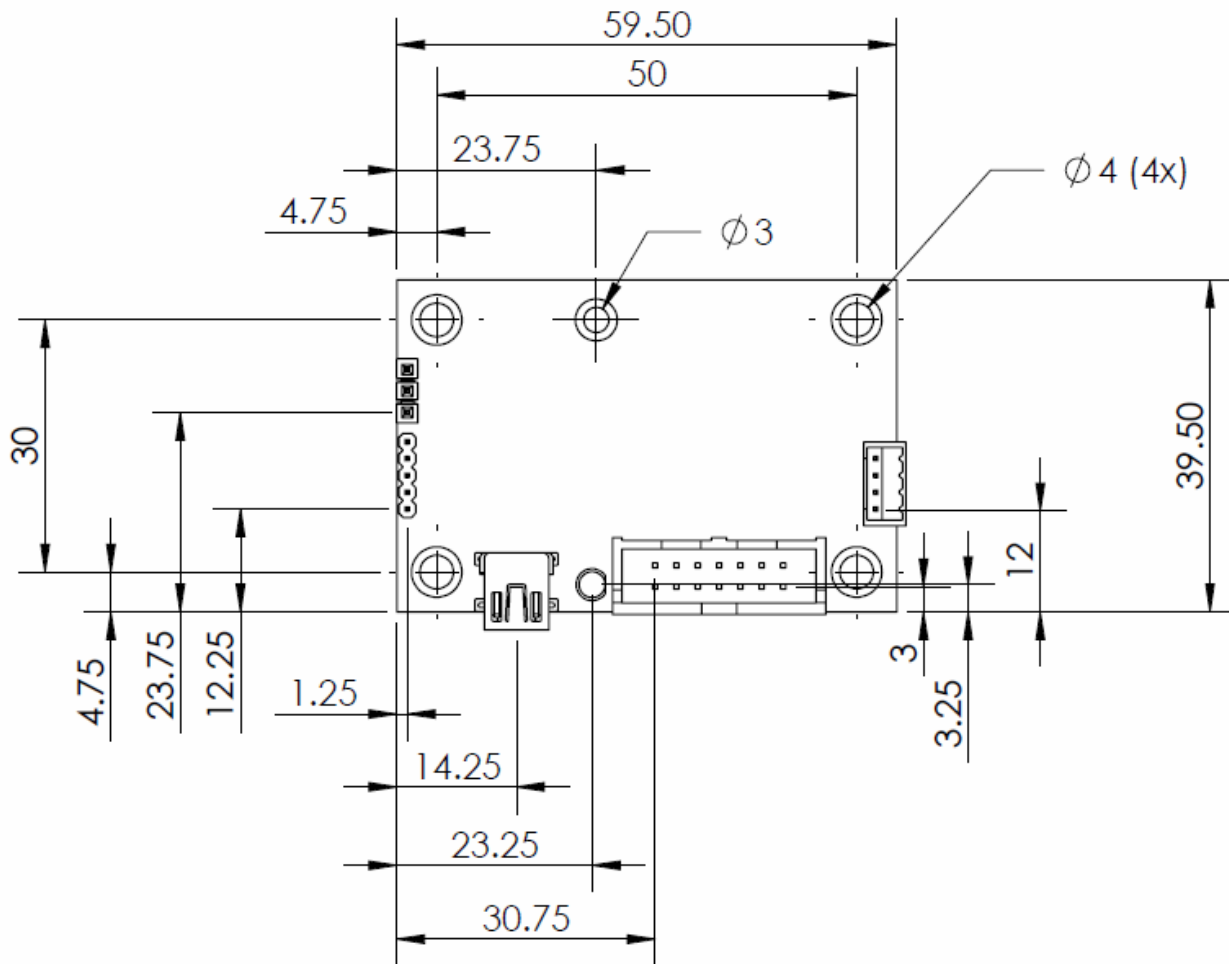


Figure 3.0

## 4. Connectivity

In order for ACE-SDC-V3 to operate, it must be supplied with +12VDC to +24VDC. Power pins as well as communication port pin outs are shown below.

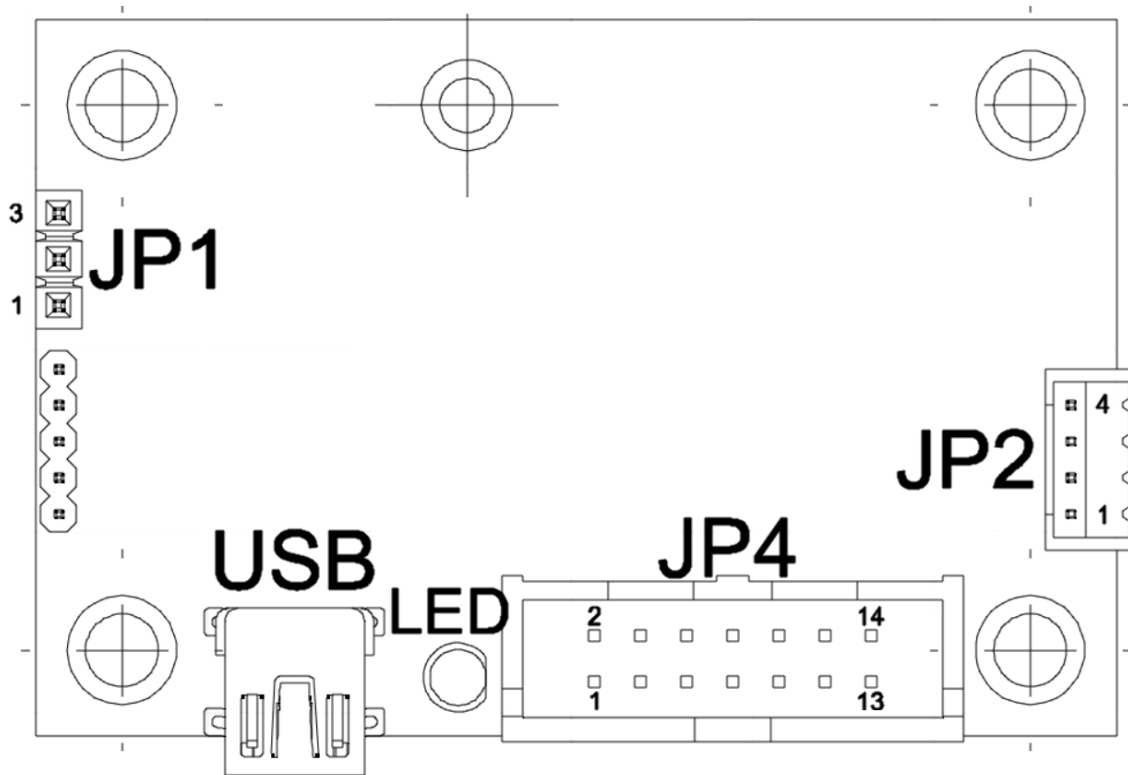


Figure 4.0

### 4.1. JP1 3-Pin Connector Pin-outs

Pin #	Name	In/Out	Description
1	3.3V Out	O	3.3V Output
2	Analog Input	I	Analog Signal Input used for Analog Speed Control (Section 7.9)
3	GND	O	Ground

Table 4.0

#### 3-pin Mating Connector Information

Mating Connector Description: Female 3 pin 0.1 inch single row  
 Mating Connector Manufacturer: Tyco/AMP  
 Mating Connector Manufacturer Part: 770602-3 (Connector)  
 770666-1 (Pin)

### 4.2. JP2 4-Pin Motor Connector Pin-outs

Pin #	Name	In/Out	Description
1	A	O	Phase A of Bi-polar Step Motor



2	/A	○	Phase /A of Bi-polar Step Motor
3	B	○	Phase B of Bi-polar Step Motor
4	/B	○	Phase /B of Bi-polar Step Motor

Table 4.1

#### 4-pin Mating Connector Information

Mating Connector Description: Female 10 pin 2mm single row  
Mating Connector Manufacturer: HIROSE  
Mating Connector Manufacturer Part: DF3-4S-2C (4 pin female connector)  
DF3-2428SC (female pin)

**Warning: Do not disconnect the motor connector while power is on. This can damage the product! Turn off the power before disconnecting the motor connector.**

#### 4.3. JP4 14-Pin IO Connector Pin-outs

Pin #	Name	In/Out	Description
1	OPTO	I	Opto-Supply Input (+12 to +24VDC)
2	DI1	I	Digital Input 1
3	DI2	I	Digital Input 2
4	DI3	I	Digital Input 3
5	DI4	I	Digital Input 4
6	DI5	I	Digital Input 5
7	DI6	I	Digital Input 6
8	DO1	O	Digital Output 1
9	DO2	O	Digital Output 2
10	NC	NC	No Connection
11	GND	I	Ground
12	PWR	I	Power (+12 to +24 VDC)
13	GND	I	Ground
14	PWR	I	Power (+12 to +24 VDC)

Table 4.2

#### 14-pin Mating Connector Information

Mating Connector Description: Female 14 pin 0.1 inch dual row  
Mating Connector Manufacturer: AMP  
Mating Connector Manufacturer Part: 1658621-2

## 4.4. Internal Interface Circuit

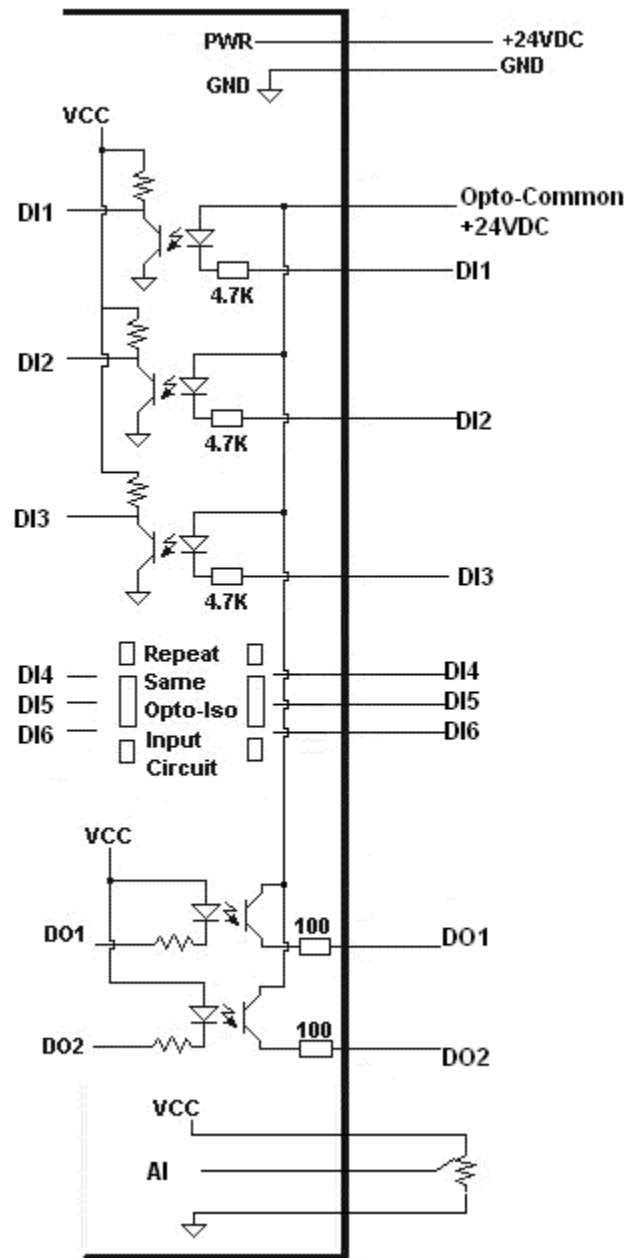


Figure 4.1

## 4.5. Digital Inputs

Figure 4.2 shows the detailed schematic of the opto-isolated general purpose digital inputs. All opto-isolated inputs are NPN type.

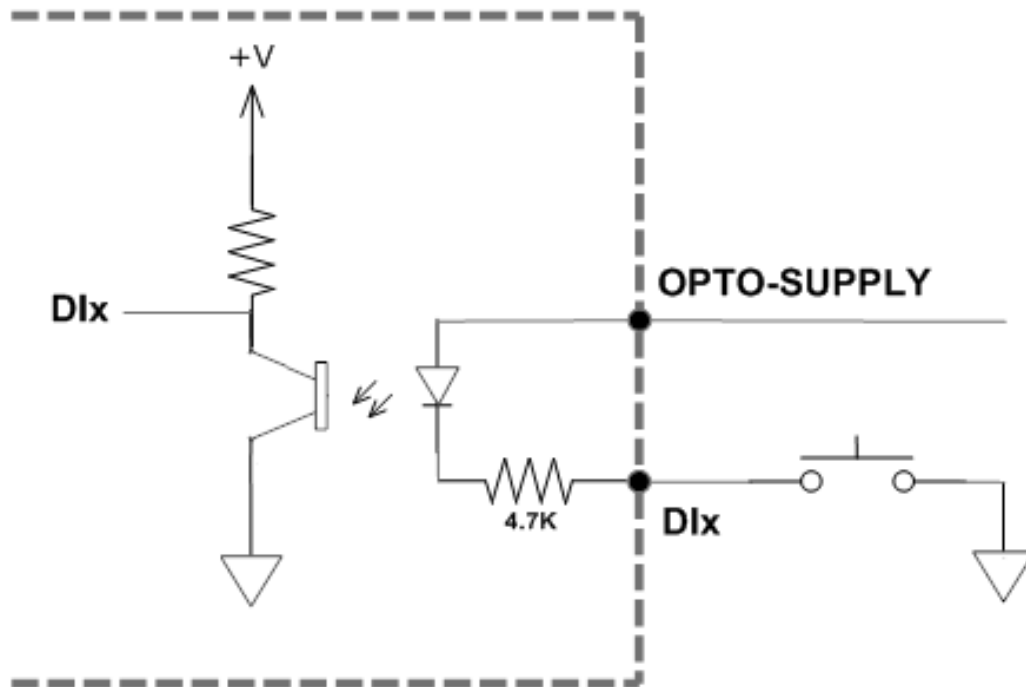


Figure 4.2

The opto-supply must be connected to +12 to +24 VDC in order for the digital inputs to operate.

When the digital input is pulled to ground, current will flow from the opto-supply to ground, turning on the opto-isolator and activating the input.

To deactivate the input, the digital input should be left unconnected or pulled up to the opto-supply, preventing current from flowing through the opto-isolator.

## 4.6. Digital Outputs

Figure 4.3 shows an example wiring of the digital outputs. All opto-isolated digital outputs will be PNP type.

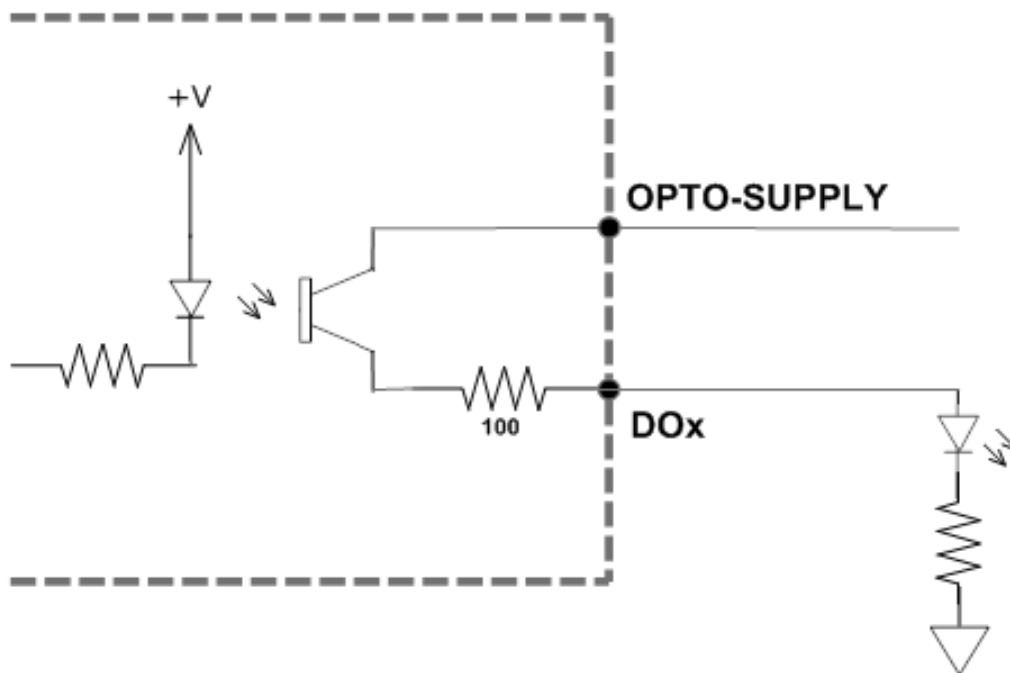


Figure 4.3

The opto-supply must be connected to +12 to +24VDC in order for the digital outputs to operate.

When activated, the opto-isolator for the digital output pulls the voltage on the digital output line to the opto-supply. The maximum sink current for digital outputs is 50mA. Take caution to select the appropriate external resistor so that the current does not exceed 50mA.

When deactivated, the opto-isolator will break the connection between the digital output and the power supply.

## 4.7. Analog Input

Analog inputs are 0 to 3.3V range and 10 bit in resolution. Using the analog input, analog speed control can be achieved. Section 7.9 will provide details on analog speed control mode. The maximum source current for the analog input is 10mA. The recommended potentiometer range is 1K $\Omega$  to 10K $\Omega$ .

---

## 5. Stepper Motor Driver Overview

The ACE-SDC-V3 has one built-in microstep driver, allowing the controller to directly drive a stepper motor.

### 5.1. Microstep

The ACE-SDC-V3 has four microstepping options including full step, ½ step, ¼ step, and 1/16 step. In general, the number of microsteps per revolution can be determined by the following equation, where “DEGREE” is the full step angle of the motor and “USTEP” is the microstep setting.

$$\text{Microsteps/Revolution} = 360^\circ / (\text{DEGREE} * \text{USTEP})$$

For example, using a 1.8° motor and a microstep setting of 1/16, the number of microsteps/revolution would be  $[360^\circ / (1.8^\circ * 1/16)] = 3200$ .

### 5.2. Driver Current

The ACE-SDC-V3 has a maximum rated current output of 2.0A. The current settings should not exceed the rated current of the motor being used. Setting the driver current higher than the maximum rated current will overheat and potentially damage the motor. It is recommended to use a current setting that is in the range of 60-80% of the maximum rated current for the motor.

## 6. Communication Interface

### 6.1. USB Communication

ACE-SDV-V3 USB communication is 2.0 compliant.

In order to communicate with ACE-SDV-V3 via USB, the proper software driver must first be installed. Before connecting the ACE-SDC-V3 controller, or running any programs, please go to the Arcus website, download the Arcus Drivers and Tools Setup, and run the installation.

All USB communication will be done using an ASCII command protocol.

#### 6.1.1. Typical USB Setup

The ACE-SDC-V3 can be connected to a PC directly via USB or through a USB hub. All USB cables should have a noise suppression choke to avoid communication loss or interruption. See a typical USB network setup in Figure 6.0 below.

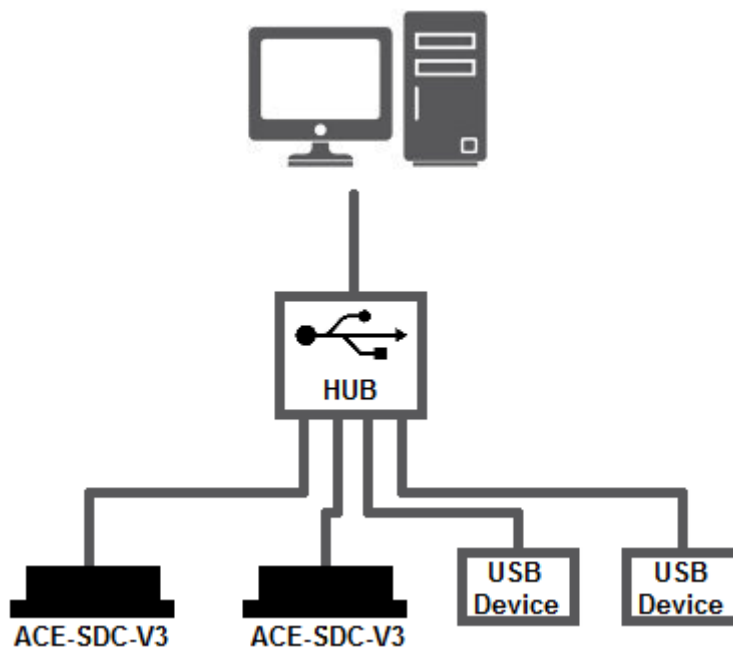


Figure 6.0

#### 6.1.2. USB Communication API

Communication between the PC and ACE-SDC-V3 is done using the Windows compatible DLL API function calls shown below. Windows programming languages such as Visual BASIC, Visual C++, LabVIEW, or any other programming language that can use a DLL can be used to communicate with the ACE-SDC-V3.

Typical communication transaction time between PC and ACE-SDC-V3 for sending a command from a PC and getting a reply from the controller using the **fnPerformaxComSendRecv()** API function is in single digit milliseconds. This value will vary with the CPU speed of the PC as well as the type of command.

For USB communication, following DLL API functions are provided.

**BOOL fnPerformaxComGetNumDevices**(OUT LPDWORD lpNumDevices);

- This function is used to get total number of all types of Performax and Performax USB modules connected to the PC.

**BOOL fnPerformaxComGetProductString**(IN DWORD dwNumDevices,  
OUT LPVOID lpDeviceString,  
IN DWORD dwOptions);

- This function is used to get the Performax or Performax product string. This function is used to find out Performax USB module product string and its associated index number. Index number starts from 0.

**BOOL fnPerformaxComOpen**(IN DWORD dwDeviceNum,  
OUT HANDLE\* pHandle);

- This function is used to open communication with the Performax USB module and to get communication handle. dwDeviceNum starts from 0.

**BOOL fnPerformaxComClose**(IN HANDLE pHandle);

- This function is used to close communication with the Performax USB module.

**BOOL fnPerformaxComSetTimeouts**(IN DWORD dwReadTimeout,  
DWORD dwWriteTimeout);

- This function is used to set the communication read and write timeout. Values are in milliseconds. This must be set for the communication to work. Typical value of 1000 msec is recommended.

**BOOL fnPerformaxComSendRecv**(IN HANDLE pHandle,  
IN LPVOID wBuffer,  
IN DWORD dwNumBytesToWrite,  
IN DWORD dwNumBytesToRead,  
OUT LPVOID rBuffer);

- This function is used to send commands and receive replies. The number of bytes to read and write must be 64 characters.

**BOOL fnPerformaxComFlush**(IN HANDLE pHandle)

- Flushes the communication buffer on the PC as well as the USB controller. It is recommended to perform this operation right after the communication handle is opened.

### 6.1.3. USB Communication Issues

A common problem that users may have with USB communication is that after sending a command from the PC to the device, the response is not received by the PC until another command is sent. In this case, the data buffers between the PC and the USB device are out of sync. Below are some suggestions to help alleviate the issue.

- 1) **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 the PC and the USB device. See the following function prototype below:

**BOOL fnPerformmaxComFlush(IN HANDLE pHandle)**

- 2) **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. See Figure 6.1.



Figure 6.1

## 6.2. Device Number

If multiple ACE-SDC-V3 devices are connected to the PC, each device should have a unique device number. This will allow the PC to differentiate between multiple controllers. In order to make this change to an ACE-SDC-V3, first store the desired number using the **DN** command. Note that this value must be within the range [CFG00 – CFG99].

To write the values to the devices' flash memory, use the **STORE** command. After a complete power cycle, the new device number will be written to memory. Note that before a power cycle is completed, the settings will not take effect.

By default: Device Name is set to: **CFG00**

## 6.3. Windows GUI

The ACE-SDC-V3 comes with a Windows GUI program to test, program, compile, download, and debug the controller. The Windows GUI will perform all communication via USB. See section 8 for further details.



## 7. General Operation Overview

**Important Note:** All the commands described in this section are defined as ASCII or standalone commands. ASCII commands are used when communicating over USB.

### 7.1. Motion Profile

The ACE-SDC-V3 incorporates trapezoidal velocity profile as shown in Figure 7.0.

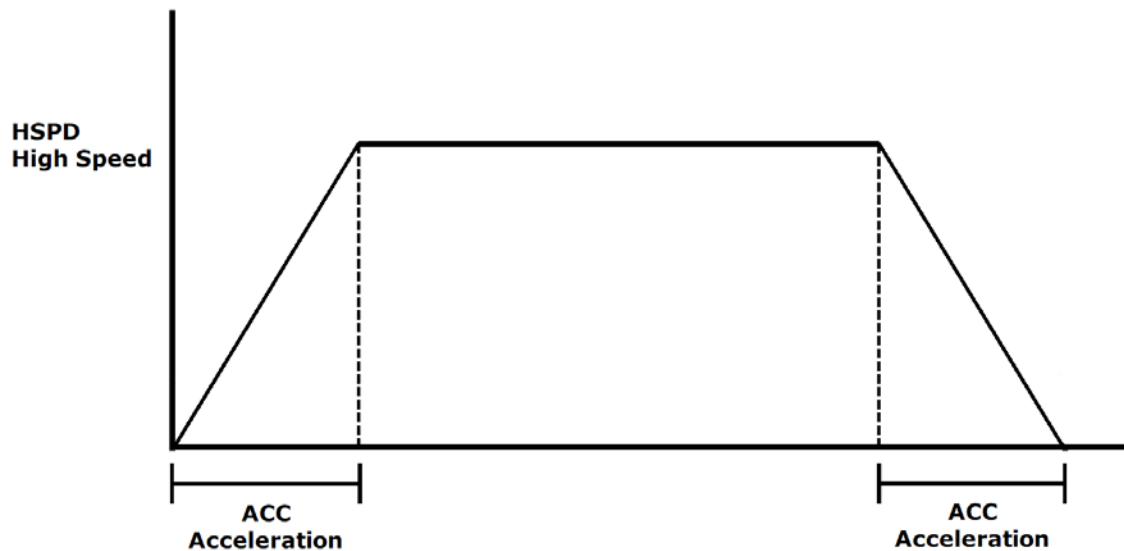


Figure 7.0

Once a typical move is issued, the motor will immediately start and accelerate to the high speed. Once at high speed, the motor will move at a constant speed until it decelerates and stops.

High speed is in pps (pulses/second). Use ASCII commands **HSPD** to set/get high speed settings. Acceleration times are in milliseconds. Use the **ACC** command to set/get acceleration values.

The minimum and maximum acceleration values depend on the high speed and low speed settings. Refer to Table A.0 and Figure A.0 in **Appendix A** for details.

ASCII	<b>HSPD</b>	<b>ACC</b>
-------	-------------	------------

### 7.2. On-The-Fly Speed Change

An on-the-fly speed change can be achieved at any point while the motor is in motion. The ASCII command **SSPD[Value]**, (no square brackets), can be used to perform the actual speed change.

ASCII	<b>SSPD</b>
-------	-------------

### 7.3. Motor Position

The ACE-SDC-V3 has a 32 bit signed step position counter. Range of the position counter is from -2,147,483,648 to 2,147,483,647. Motor positions can be read using the ASCII command **PX**, which returns the pulse position.

ASCII | **PX**

### 7.4. Motor Power

The **EO** command can be used to enable or disable the current to the motor. The effect of the enable output signals will depend on the characteristics of the motor drive. By default, the enable output is enabled at boot-up.

ASCII | **EO**

### 7.5. Jog Move

A jog move is used to continuously move the motor without stopping. Use the **J+/J-** command when operating in ASCII mode. Once this move is started, the motor will only stop if a limit input is activated during the move or a stop command is issued.

If a motion command is sent while the controller is already moving, the command is not processed. Instead, an error response is returned. See table 9.2 for details on error responses.

ASCII | **J[+/-]**

### 7.6. Stopping

When the motor is performing any type of move, motion can be stopped abruptly or with deceleration. It is recommended to use decelerated stops so that there is less impact on the system.

Use the **ABORT** (ASCII) command to immediately stop the motor. To employ deceleration on a stop, use the **STOP** (ASCII) command to stop the motor.

ASCII | **ABORT** | **STOP**

### 7.7. Motor Status

Motor status can be read anytime by reading the response to the **MST** command. The following is the bit representation of motor status:

Bit	Description
0	Motor running at constant speed
1	Motor in acceleration
2	Motor in deceleration

Table 7.0

ASCII | **MST**

## 7.8. Digital Inputs / Outputs

ACE-SDC-V3 module comes with 6 digital inputs and 2 digital outputs.

### 7.8.1. Digital Inputs

Read digital input status using the **DI** command.

Digital input values can also be referenced one bit at a time by the **DI[1-6]** commands. Note that the indexes are 1-based for the bit references (i.e. DI1 refers to bit 0, not bit 1)

Bit	Description	Bit-Wise Command
0	Digital Input 1	DI1
1	Digital Input 2	DI2
2	Digital Input 3	DI3
3	Digital Input 4	DI4
4	Digital Input 5	DI5
5	Digital Input 6	DI6

Table 7.1

If a digital input is on (i.e. input is pulled to GND of opto-supply), the bit status is 0. Otherwise, the bit status is 1.

ASCII	<b>DI</b>	<b>DI[1-6]</b>
-------	-----------	----------------

### 7.8.2. Digital Outputs

Read digital output status using the **DO** command.

Digital output values can also be referenced one bit at a time by the **DO[1-2]** commands. Note that the indexes are 1-based for the bit references (i.e. DO1 refers to bit 0, not bit 1)

Bit	Description	Bit-Wise Command
0	Digital Output 1 (Moving)	DO1
1	Digital Output 2 (General)	DO2

Table 7.2

If digital output is turned on (i.e. the output is pulled to GND), the bit status is 1. Otherwise, the bit status is 0.

The initial state of both digital outputs can be defined by setting the **DOBOOT** register to the desired initial digital output value. The value is stored to flash memory once the **STORE** command is issued.

ASCII	<b>DO</b>	<b>DO[1-2]</b>	<b>DOBOOT</b>
-------	-----------	----------------	---------------

## 7.9. Analog Speed Control

In analog speed control mode, the speed of the motor is determined by the analog input value [0 – 1023]. The direction of the motor is determined by the status of digital input 1.

### 7.9.1. Current Used For Analog Speed Control

There is only one current value used for the analog speed mode as shown below. Other current values including idle current are NOT used for analog speed control. Digital inputs 3 and 4 for current selection is NOT used for analog speed control.

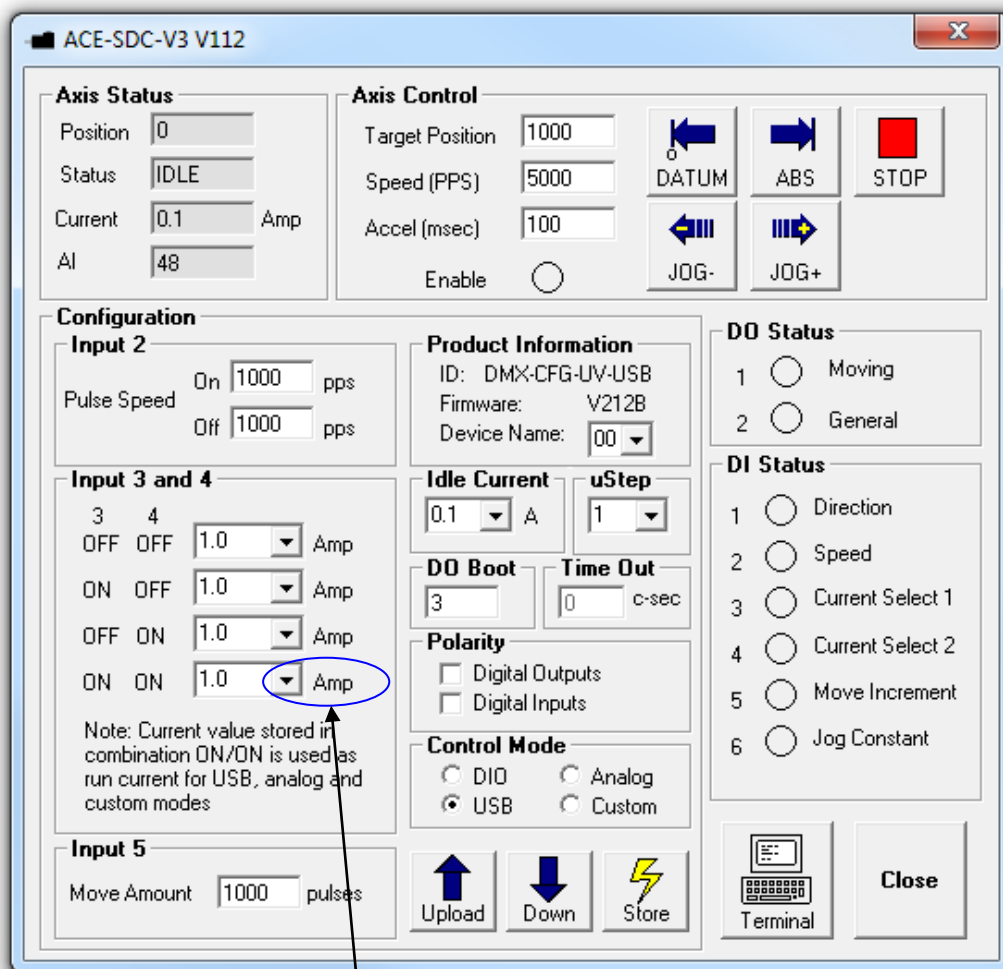


Figure 7.1

Current value used for Analog Speed Control

### 7.9.2. Digital IO Used For Analog Speed Control

There is only one digital input for direction used for the analog speed as shown below. Other digital IO must be turned OFF for proper analog speed control.

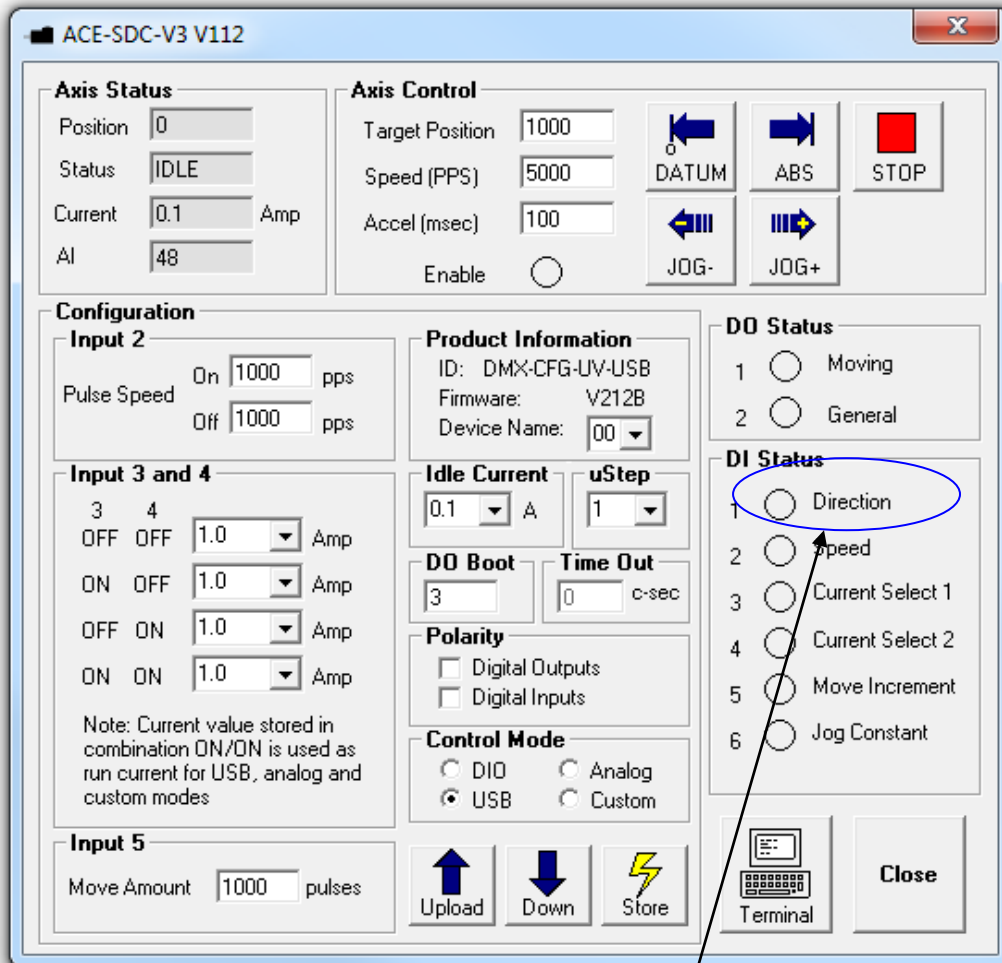


Figure 7.2

Only the direction input is used in analog speed control.

### 7.10. Polarity

Using the **POL** command, polarity of the following signals can be configured:

Bit	Description
0	Digital Outputs
1	Digital Inputs

Table 7.3

ASCII | **POL**

## 7.11. Storing to Flash

The following items are stored to flash:

ASCII Command	Description
DN	Device name
P8	Idle current
P9	Microstep
PA	Move mode
POL	Polarity Settings
DOBOOT	DO configuration at boot-up

Table 7.4

## 8. Software Overview

The ACE-SDC-V3 has Windows compatible software that allows for USB communication. It can be downloaded from the Arcus-Technology website. To communicate over a USB connection, make sure that the ACE-SDC-V3 is connected to one of the available ports on the PC. Startup the ACE-SDC-V3 GUI program and you will see the following screen in Figure 8.0.



Figure 8.0

To choose a particular device, select it and then press OK.

**Important Note:** In order to communicate with ACE-SDC-V3 via USB, the proper driver must first be installed. Before connecting the ACE-SDC-V3 device or running any program, please go to the Arcus-Technology website, download the USB driver installation instruction and run the USB Driver Installation Program.

## 8.1. Main Control Screen

The Main Control Screen provides accessibility to all the available function on the ACE-SDC-V3. All features can be tested and verified.

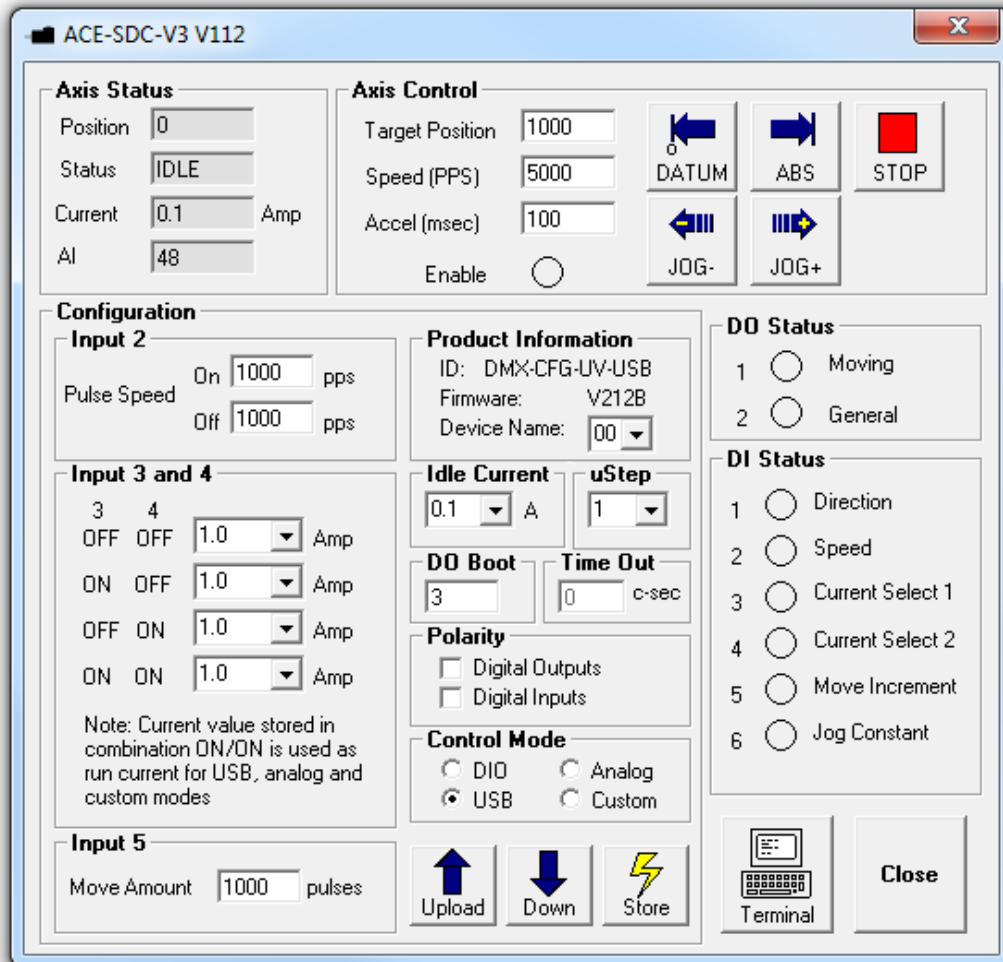


Figure 8.1

### 8.1.1. Status

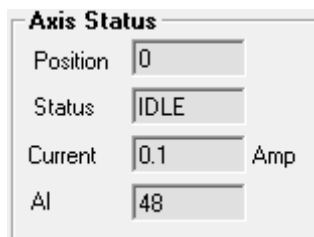


Figure 8.2

1. **Position Value** – Shows the value of the current position.
2. **Motor Status**
  - a. **IDLE** – Motor is idle and not moving.



- b. **ACCEL** – Motor is accelerating.
  - c. **DECEL** – Motor is decelerating.
  - d. **CONST** – Motor is moving at constant speed.
3. **Current**  
Displays current value.
4. **AI**  
Displays current analog input value (10 bit)  
Range is 0 to 1023 which corresponds to 0 to 3.3V.

### 8.1.2. Control

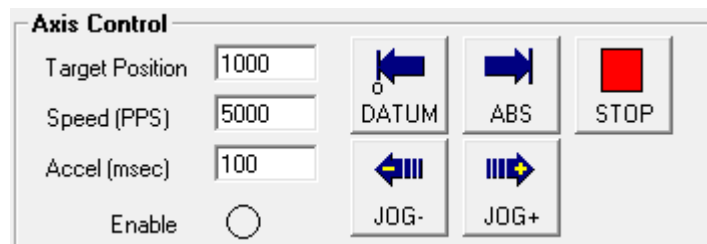


Figure 8.3

1. **Target Position** – Used to move the motor to the target position.
2. **Speed** – Set high speed setting of the motor (up to 16Kpps).
3. **Accel** – Set acceleration time (up to 300ms).
4. **Enable Status** – Displays the energized status of the motor
5. **Datum** – Moves the motor back to the zero position.
6. **ABS** – Moves the motor to the target position.
7. **JOG-** – Jogs the motor in the minus direction.
8. **JOG+** – Jogs the motor in the plus direction.
9. **STOP** – Stops the motor.

### 8.1.3. Digital Input

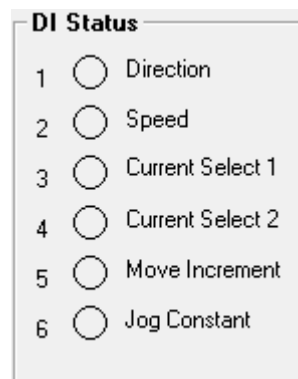


Figure 8.4

Displays the current digital input status. When used in DIO control mode, each input is designated for specific use:

Digital Input	Description
1	Direction
2	Speed
3	Current Select 1
4	Current Select 2
5	Move Increment
6	Jog Constant

Table 8.0

**Note: When in analog mode, only DI1 (direction input) is used.**

#### 8.1.4. Digital Output

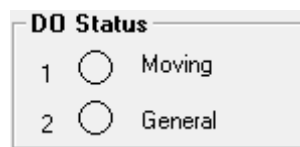


Figure 8.5

Displays the current digital output status.

DO1 is reserved as a moving output. When the motor is in motion, DO1 is turned on. DO2 is a general purpose output.

#### 8.1.5. DO Boot

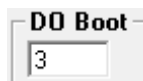


Figure 8.6

This configures the startup states of the two digital outputs. A value of '0' will disable both digital outputs, a value of '1' will cause DO1 to be enabled, a value of '2' will enable DO2, and a value of '3' will enable both digital outputs.

### 8.1.6. Terminal

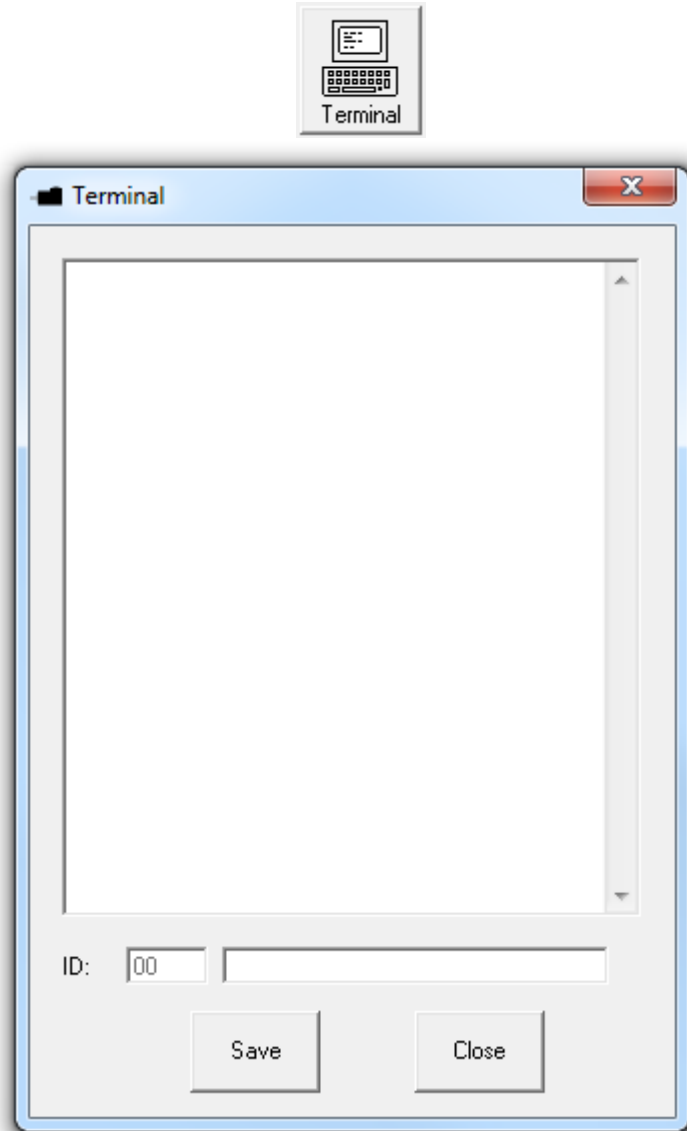


Figure 8.7

Terminal dialog box allows manual testing of the commands from a terminal screen as shown in Figure 8.7.

### 8.1.7. Microstep



Figure 8.8

The following microstep values are available: Full,  $\frac{1}{2}$ ,  $\frac{1}{4}$ ,  $\frac{1}{16}$ .

### 8.1.8. Upload / Download / Store to Flash



Figure 8.9

The following parameter settings can be uploaded or downloaded from the controller:

- Control Mode Selection (DIO, USB, or Analog)
- Idle Current
- Microstep
- DIO Motion Parameters (Speed, Current, and Increment amount)

Whenever the parameter is changed, it must be downloaded before taking effect. For example, if the idle current is changed from 0.35A to 0A, the new idle current will go into effect only after download. If the control mode is changed from DIO to USB from the screen, it must be downloaded for the new control mode to be used. Once the parameters are downloaded, they can be permanently stored to the flash memory (using the STORE command, or STORE button) so that when the controller is powered, the stored parameters are used.

The following items are stored to flash:

ASCII Command	Description
DN	Device name
P1	Pulse speed for DIO mode [DI2=0]
P2	Pulse speed for DIO mode [DI2=1]
P3	Run current for DIO mode [DI3=1/DI4=1]
P4	Run current for DIO mode [DI3=0/DI4=1]
P5	Run current for DIO mode [DI3=1/DI4=0]
P6	Run current for DIO, joystick, and USB mode [DI3=0/DI4=0]
P7	Move amount in DIO mode [DI5=1]
P8	Idle current
P9	Microstep
PA	Move mode

Table 8.1

### 8.1.9. Control Mode

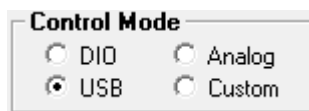


Figure 8.10

There are four control modes:

1. **DIO** – Controls the motor through digital inputs.
2. **USB** – Motor control is done through the PC using USB communication.  
 NOTE: All moves using the USB command uses current setting when input 3 and 4 are “ON”. Idle current is used when the motor is idle in USB mode.
3. **Analog** – Speed control is done using the analog input.  
 NOTE: Speed control using the analog input uses current setting when input 3 and 4 are on. Initial current setting for the analog speed control is used when the controller is powered initially. If the current is changed, download the new current, store to flash, and power cycle (turn off then back on) the controller for the new current to take effect. Idle current is not used when in analog control mode.
4. **Custom** – Custom move mode changes the names of digital outputs and inputs. It also disallows the changing of the Move Amount.

### 8.1.10. Polarity

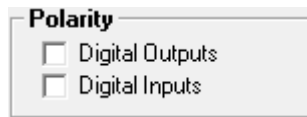


Figure 8.11

This allows the user to change to polarity setting or the Digital Outputs and/or the Digital Inputs.

### 8.1.11. Idle Current

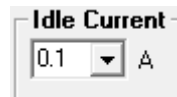


Figure 8.12

Idle current is used when the motor is not moving. If the idle current is set to 0, the driver is disabled. Only the DIO control mode and USB control mode use the idle current setting. For analog speed control mode, idle current is not used.

### 8.1.12. DIO Control Mode Settings

**Input 2**

Pulse Speed On  pps

Off  pps

---

**Input 3 and 4**

3	4	<input type="text" value="1.0"/>	Amp
OFF	OFF		
ON	OFF	<input type="text" value="1.0"/>	Amp
OFF	ON	<input type="text" value="1.0"/>	Amp
ON	ON	<input type="text" value="1.0"/>	Amp

Note: Current value stored in combination ON/ON is used as run current for USB, analog and custom modes

---

**Input 5**

Move Amount  pulses

Figure 8.13

Different move settings are selected through the digital inputs when in DIO control mode.

Digital input 1 is used to select the direction of the motion. For analog speed control mode, DI1 is also used for the direction of the motion.

Digital input 2 is used to select speed of the motion. There are two values, when the DI2 is on and when the DI2 is off.

Digital input 3 and 4 are used to select the current setting when in motion. Two bits allow four different settings of the current. Current setting selections are: 0 mA to 2.0 A (100 mA resolution). For analog USB control and analog speed control mode, current setting when input 3 and 4 are “ON” is used.

Digital input 5 is used to move a set amount, this amount is the same as the “target position” amount. For example: if the target position is 1000, the motor will move 1000 steps. Direction of the incremental move is determined by Digital input 1.

Digital input 6 is used to jog the motor. Direction of the jogging is set by digital input 1. There is no configuration value associated with digital input 6.

### 8.1.13. Product Information



**Product Information**  
ID: DMX-CFG-LV-USB  
Firmware: V212B  
Device Name: 00 ▾

Figure 8.14

Displays the product ID and the version number.

To change the device name, select a new one from the drop down list. In order for the setting to take effect, download the device name, store it to flash memory, and then perform a power cycle.

## 9. ASCII Language Specification

ACE-SDC-V3 language is case sensitive. All command should be in capital letters. Invalid commands return a “?”. Always check for the proper reply when sending a command.

For **USB communication**, send commands identical to the ones in the following table.

### 9.1 ASCII Command Set

Command	Description	Return
ABORT	Immediately stops the motor if in motion. For decelerate stop, use STOP command. This command can also be used to clear a StepNLoop error	OK
ACC	Returns current acceleration value in milliseconds.	milliseconds
ACC=[Value]	Sets acceleration value in milliseconds. Example: ACC=300	OK
AI	Get analog input status	0-1023
DI[1-6]	Get individual bit status of digital inputs	0 – on Non-zero - off
DO	Return status of digital outputs	2-bit number
DO[1-2]	Get individual bit status of digital outputs	0,1
DO[1-2] =[Value]	Set individual bit status of digital outputs	OK
DN	Get device name	[CFG00-CFG99]
DN=[Value]	Set device name	OK
EO	Returns driver power enable.	1 – Motor power enabled 0 – Motor power disabled
EO=[0 or 1]	Enables (1) or disable (0) motor power.	OK
HSPD	Returns High Speed Setting	PPS
HSPD=[Value]	Sets High Speed.	OK
ID	Returns product ID	DMX-CFG-UV-USB
J+	Jogs the motor in positive direction	OK
J-	Jogs the motor in negative direction	OK
MST	Returns motor status	Bit 0 – constant speed Bit 1 – accelerating Bit 2 – decelerating Bit 3 thru 8 – motor current (centi-A)
P1	Get pulse speed with DI2 = 0 [DIO mode only]	OK
P1=	Set pulse speed with DI2 = 0 [DIO mode only]	PPS
P2	Get pulse speed with DI2 = 2 [DIO mode only]	OK
P2=	Set pulse speed with DI2 = 2 [DIO mode only]	PPS
P3	Get run current when DI3 = 4, DI4 = 8 [DIO mode only]	1: 100 mA 2: 200 mA .... 20: 2000 mA
P3=	Set run current when DI3 = 4, DI4 = 8 [DIO mode	OK



	only]	
P4	Get run current when DI3 = 0, DI4 = 8 [DIO mode only]	1: 100 mA 2: 200 mA .... 20: 2000 mA
P4=	Set run current when DI3 = 0, DI4 = 8 [DIO mode only]	OK
P5	Get run current when DI3 = 4, DI4 = 0 [DIO mode only]	1: 100 mA 2: 200 mA .... 20: 2000 mA
P5=	Set run current when DI3 = 4, DI4 = 0 [DIO mode only]	OK
P6	Get run current when DI3 = 0, DI4 = 0 [DIO mode only] Note: This is the run current value used in joystick and USB mode.	1: 100 mA 2: 200 mA .... 20: 2000 mA
P6=	Set run current when DI3 = 0, DI4 = 0 [DIO mode only] Note: This is the run current value used in joystick and USB mode.	OK
P7	Get move amount when DI5 = 0 [DIO mode only]	32-bit
P7=	Set move amount when DI5 = 0 [DIO mode only]	OK
P8	Get idle current	1: 100 mA 2: 200 mA .... 20: 2000 mA
P8=	Set idle current	OK
P9	Get microstep	1: Fullstep 2: Halfstep 3: Quarter Step 4: 1/16 step
P9=	Set microstep	OK
PA	Get move mode	0 – DIO 1 – USB 2 – Joystick
PA=	Set move mode	OK
PX	Returns current position value	32-bit number
PX=[value]	Sets the current position value	OK
SSPD[Value]	On-the-fly speed change. Note that an “=” sign is not used for this command.	OK
STOP	Stops the motor using deceleration if in motion.	OK
STORE	Store settings to flash	OK
VER	Get firmware version	VXXX
X[value]	Moves the motor to absolute position value using the HSPD and ACC values	OK

Table 9.0

## 9.2. Error Codes

If an ASCII command cannot be processed by the ACE-SDC-V3, the controller will reply with an error code. See below for possible error responses:

Error Code	Description
?[Command]	The ASCII command is not understood by the ACE-SDC-V3
?Moving	A move or position change command is sent while the ACE-SDC-V3 is outputting pulses.
?SSPD not allowed	On-the-fly speed command, SSPD, is issued when the motor is not moving.

Table 9.1

## **Contact Information**

Arcus Technology, Inc.

3159 Independence Drive  
Livermore, CA 94551  
925-373-8800

[www.arcus-technology.com](http://www.arcus-technology.com)

The information in this document is believed to be accurate at the time of publication but is subject to change without notice.