

AB300-Series Automated Filter Wheels

User Manual

(CVI Document # 8-2016-A)
Copyright 1995, CVI Laser Corporation

CVI Laser Corporation

200 Dorado Place SE - P.O. Box 11308 - Albuquerque - NM -87192 - (505) 296-9541 - Fax (505) 298-9908

Instruments Group

NOTICES

Warranty; Hardware:

CVI Laser Corporation warrants this product to be free from defects in material and workmanship for a period of one year from the date of purchase.

Warranty; Software:

Any software associated with this product is provided "as is" with no warranty, expressed or implied. While it is CVI Laser Corporation's intent to provide error-free useful development tools, no guarantee is made regarding the accuracy or ultimate usefulness of this material.

Included Software:

Any software distributed with this product is provided free of charge as a service to the customer. The software is intended to be used as a tool for development and as an example of one possible method of code implementation. It is not intended to be a "user application."

Software Copyright:

CVI maintains the copyright on this material, but grants the customer rights to use or modify the software described herein without obtaining CVI's permission and without the requirement to reference CVI as the source of material.

Changes:

CVI reserves the right, without prior or further notice, to make changes to any of its products described or referred to herein to improve reliability, function, or design.

Liability:

CVI accepts no liability for incidental or consequential damages arising from the use of this product.

CVI Laser Corporation does not recommend the use of its components or software products in life support applications wherein a malfunction or failure of the product may directly threaten life or result in injury.

TABLE OF CONTENTS

1. DESCRIPTION	3
1.1 General Description.....	3
1.2 Detailed Operation Description.....	3
2. OPERATION	3
2.1 Mounting.....	3
2.2 Filter Installation.....	4
2.3 Connections.....	4
3. PROGRAMMING	4
3.1 Overview.....	4
3.2 Commands.....	5
3.2.1 Baud.....	5
3.2.2 Echo.....	6
3.2.3 EEPROM Read.....	6
3.2.4 EEPROM Write.....	6
3.2.5 Filter.....	7
3.2.6 Query Position.....	7
3.2.7 Reset.....	7
3.2.8 Step Down.....	8
3.2.9 Step Up.....	8
3.2.10 Zero.....	8
4. OPERATION HINTS	8
5. SPECIFICATIONS	10
5.1 Mechanical.....	10
5.1.1 AB301, Slit Aperture.....	10
5.1.2 AB301, Clear Aperture.....	11
5.1.3 AB302.....	12
5.2 Electrical.....	13
5.2.1 RS232 In.....	13
5.2.2 Power.....	13
5.2.2.1 AB301.....	13
5.2.2.2 AB302.....	13

5.2.3 Filter Wheel.....	14
5.3 Cables.....	14
5.3.1 Computer to filter wheel RS232.....	14
5.3.2 Filter wheel assembly to filter wheel controller.....	15

1. DESCRIPTION

1.1 General Description

The AB300-Series Automated Filter Wheels consist of a motor driven wheel, a computer based controller, a power supply, and one cable.

Filter wheel control is provided via RS232 from a host computer to the AB300-Series controller.

1.2 Detailed Operation Description

The filter wheel assembly has a stepper motor and a position sensing device. The motor drives the filter wheel via a belt. The position sensing device allows the filter wheel to determine a "home" position during a reset. This home position is some number of steps (the "offset") away from the filter 1 position. After home is found, the wheel automatically moves to the filter one position.

The relative position of filter 1 is determined by the Filter 1 Offset, stored in non-volatile memory. From the filter 1 position, all other positions are reached by a fixed number of steps and are not individually programmable. If fine tuning of each position is desired, the Step Up and Step Down commands may be used. This fine adjusting will not be saved however, and subsequent moves to each position will require re-adjusting.

The filter wheel controller responds to RS232 commands and requires software to control its operation. No "manual" mode of operation is provided.

2. OPERATION

2.1 Mounting

Mounting of the wheel itself is via 1/4-20 threaded holes (if provided) located on the sides of the filter wheel housing or via a mounting flange (or flanges) on the front and/or back of the housing.

Due to the vast possibilities of mounting arrangements, custom modification of the filter wheel housing may be necessary. Modified hardware will not be accepted by CVI for full refund. If replacement is deemed necessary, CVI will try to re-use modified parts, but is under no obligation to duplicate customer custom modifications. Mounting hole patterns and flange dimensions are provided in *Specifications; Mechanical*.

2.2 Filter Installation

Access to the filters is provided by removing four screws to remove the cover of the filter wheel housing.

Filters may be held in place by whatever means suit the customer's application. A popular method is to use small dabs of silicone RTV on the edge of a filter to "tack" the filter in place.

2.3 Connections

The filter wheel housing to filter wheel controller connection is made via a DB9-F to DB9-M cable (CVI pn: DK12AT, provided.) The connection is from the controller port "*To Filter Wheel*" to the filter wheel housing.

The host computer to filter wheel controller connection is made via a standard RS232 cable (not a "Null-Modem" type) to the "*RS232 In*" port on the filter wheel controller.

The power supply connects to the filter wheel controller at the "*Power*" jack.

3. PROGRAMMING

3.1 Overview

The AB300-Series controller responds to software commands via the RS232 line. This communication may come from any device capable of RS232, utilizing any language and operating system.

The AB300-Series controller is a DCE device and uses the following protocol:

- 8 Data bits, 1 Stop bit, No Parity
- Baud is user programmable (factory shipped configured at 9600)
- Flow control is via hardware handshaking (utilizing CTS and RTS)

Hardware handshaking for the AB300-Series controller is employed as follows:

CTS (from controller to host) signals if the controller is ready to receive a byte over the RS232 line. If the CTS line is asserted (ie, a positive voltage) the controller is ready to receive. If a byte(s) is sent while CTS is deasserted (ie, a negative voltage) the character may be lost. As the AB300-Series controller has a one character input buffer, multiple characters or commands cannot be received and stored. Since CTS held deasserted while the filter wheel is executing a command, the reassertion of CTS signals that the command has completed.

RTS (from host to controller) signals if the host is ready to receive a byte(s) from the controller. The AB300-Series controller does check this handshaking line and will wait

indefinitely for RTS to be asserted before sending a character. For this reason, the host system must ensure RTS is asserted for the filter wheel to work properly.

Proper handling of hardware handshaking is vital to smooth operation. Each programming language has different methods and defaults for this protocol which must be understood by the programmer.

The filter wheel controller is designed to be the sole device on an RS232 line. If multiple devices are desired and com ports are not available, a serial port switch box may be used (contact CVI for details and suggestions on multi-filter wheel systems.)

3.2 Commands

The communication to and from the controller utilizes bytes (8 bits) sent as characters (the reason for 8 data bits per character.) Each command consists of one or more bytes (characters) to the controller. Depending on the command, one or more bytes may be returned by the controller to return data or signify the end of a command .

Commands should be sent as indicated, without sending terminating characters (for example, do not send carriage return at the end of each command.) Undesired characters may have an adverse affect on controller operation.

For the following, a value surrounded by angle brackets <>, is the decimal (base 10) value of byte transmitted. Typical programming practice would be to send and receive these bytes as ASCII characters, converting them to a decimal value as needed by other programming tasks. How this is done is dependent on the language used. Refer to the sample programs for examples.

For commands that return a "Status Byte," that byte is interpreted as follows;

Bit	Meaning
7 (MSB)	0 if command accepted
6	1 if value specified = current value
5	0 = value too high, 1 = value too low (ignore if bit 7 = 0)
4	0 = moving to lower filter, 1 = moving to higher filter
3 - 0	Not used

The commands and their protocol are described as follows (in alphabetical order);

3.2.1 Baud

To controller: <58><Baud Rate Byte>
 From controller: <Status><24>
 Action: Communication baud rate switched to new rate after <24> sent

*Dec hex
58 -> 3A*

Baud rate bytes are as follows;

Byte	Baud	Byte	Baud
0	9600	4	600
1	4800	5	300
2	2400	6	150
3	1200	7	75

Note: The controller saves the last selected baud rate and will use that rate at the next power on or RESET. Therefore, the user must also remember the last baud rate. Powering the AB300-Series controller off then on will not reset the baud rate.

3.2.2 Echo

To controller: <27>
 From controller: <27>
 Action: None

27 → ^{hex} 1B

The ECHO command is used to verify communication.

3.2.3 EEPROM Read

To controller: <56><Address Byte>
 From controller: <Data High Byte><Data Low Byte><Status Byte><24>
 Action: None

56 → ^{hex} 38

Reads the word (two bytes) value currently stored in EEPROM non-volatile memory at the specified address (0 - 15.) The two returned data bytes form a word as follows;

$$(<Data High Byte> \times 256) + <Data Low Byte> = Word$$

3.2.4 EEPROM Write

CAUTION! Improper or indiscriminate use of this command may impair the functionality of the instrument. Caution and discretion is advised!

To controller: <59><Address Byte><Data High Byte><Data Low Byte>
 <Checksum Byte>
 Action: ~~Data~~ Data written to EEPROM address
 From controller: <Status Byte><24>

59 → 3B

Writes a word (two bytes) to the EEPROM non-volatile memory at the specified address (0 - 15.) The value of the word is given by

$$(\text{Data High Byte} \times 256) + \text{Data Low Byte} = \text{Word}$$

The "Checksum Byte" must be correct for the write to be allowed, and is determined by

$$\text{Address} + \text{Data High Byte} + \text{Data Low Byte} = \text{Checksum}$$

where Checksum is a byte long (truncated if necessary) and any carry is ignored. If the checksum byte is not correct, the command will be IGNORED (no status or <24> will be sent.)

This command is normally not used except at the factory.

3.2.5 Filter

To controller: <15><Position Byte>

Action: Filter wheel moves to specified position

From controller: <Status Byte><24>

Dec Hex
15 → F

Valid positions are 1 through 5. If position is valid, the wheel turns to the new position. If invalid, no movement occurs and status byte signals problem.

3.2.6 Query Position

To controller: <29>

From controller: <Position Byte><Status Byte><24>

Action: None

Dec Hex
29 1D

QUERY POSITION will return the present filter position in "Position Byte." (Eg: 1, 2, 3... decimal, *not* ASCII character "1", "2", "3".)

3.2.7 Reset

To controller: <255><255>

Action: Filter wheel re-homes and goes to filter position 1.

From controller: Nothing

255 → FF

This command effectively does a power-on reset, resetting the controller board and re-homing the wheel. It is used when there is suspicion the wheel has lost track of where it physically is (in the event of jamming for example.) In normal use this command should not be needed. It may, however, be used periodically for "just in case" reassurance that the wheel position is correct.

for any reason (e.g.: the wheel jams or becomes disconnected), the wheel loses its position, it will not be able to tell a problem exists.

It may be desirable or at least reassuring to “close the loop” by having the wheel re-find its home position. This is done by issuing the RESET command.

Using QUERY position: After power on or the RESET command a filter wheel goes to the filter one position. All subsequent movements from there are done by issuing FILTER commands. So, theoretically at least, the controlling software should always know the current filter position of any filter wheel.

Still, it may be reassuring to query the filter wheel controller to verify what it thinks is the current position matches what the controlling software thinks. This is done with the QUERY command. Note, the filter wheel is positioned “open loop” and cannot tell if a malfunction has prevented the wheel from reaching its desired target. For example, if a wheel became unplugged from the controller, no error would occur using FILTER commands, and a QUERY would return the intended position, not the actual filter position (which is long lost.) Only the RESET command will detect the problem by failing to home.

Fine tuning each position: Only the filter one position has a programmable “fine tuning” adjustment that is saved. For filter one position only, this is done by using STEP UP and STEP DOWN, then issuing a ZERO command to store the setting. All other filter positions are a fixed number of steps from the filter one position.

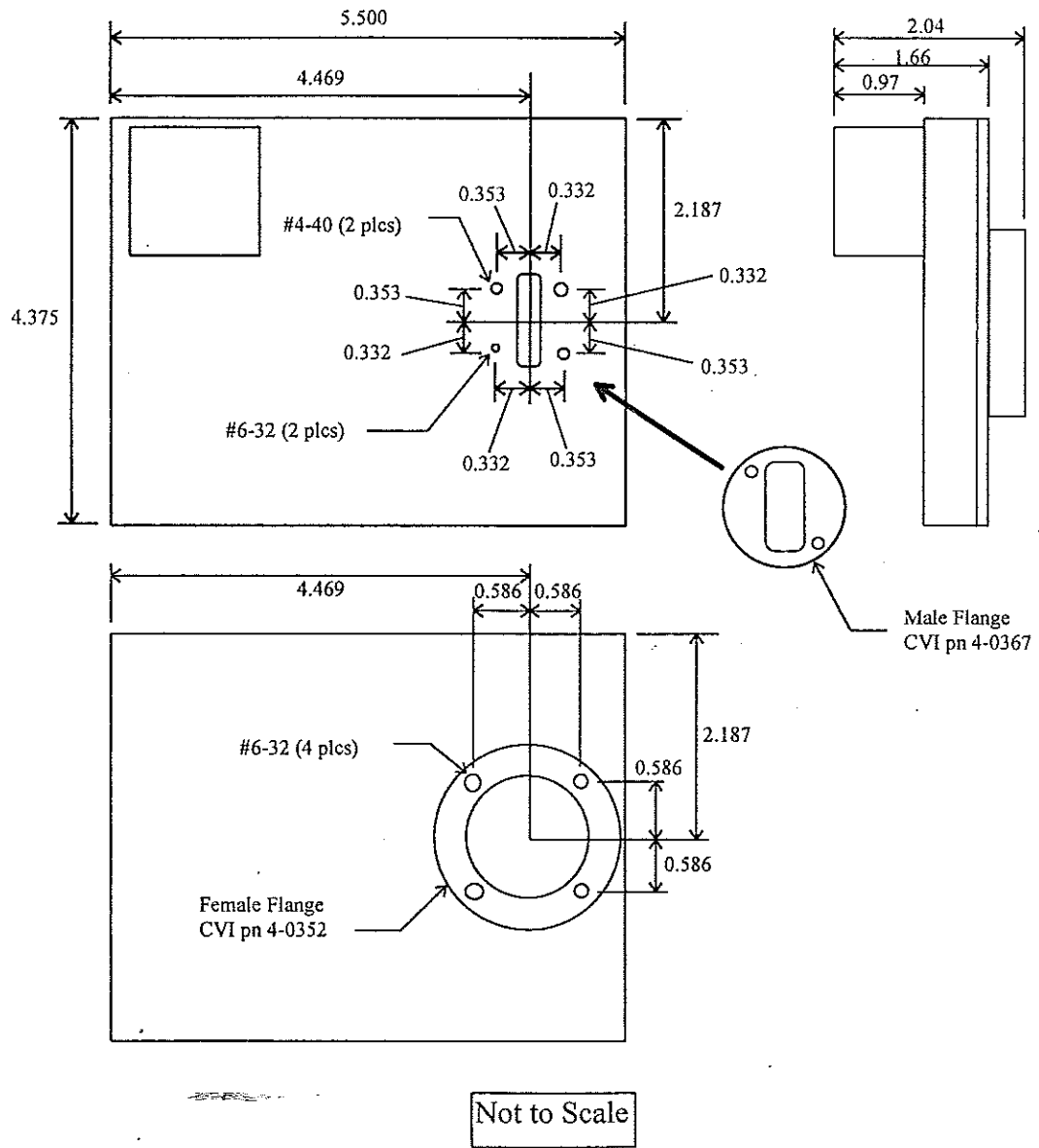
Fine tuning of each filter position may be accomplished using software techniques. The process is to go to a filter position, then issue STEP UP/DOWN commands to fine tune the position. The number of STEP UP/DOWN commands needed must be handled (saved) by the controlling software. Each time that position is to be recalled, a FILTER # and the appropriate number of STEPS would be issued.

Note: Adjusting a position using STEPs does not affect the “base” position of subsequent positions (E.g.: Using 3 step up commands for filter 2 does not make the subsequent filter 3 position off by 3 steps.)

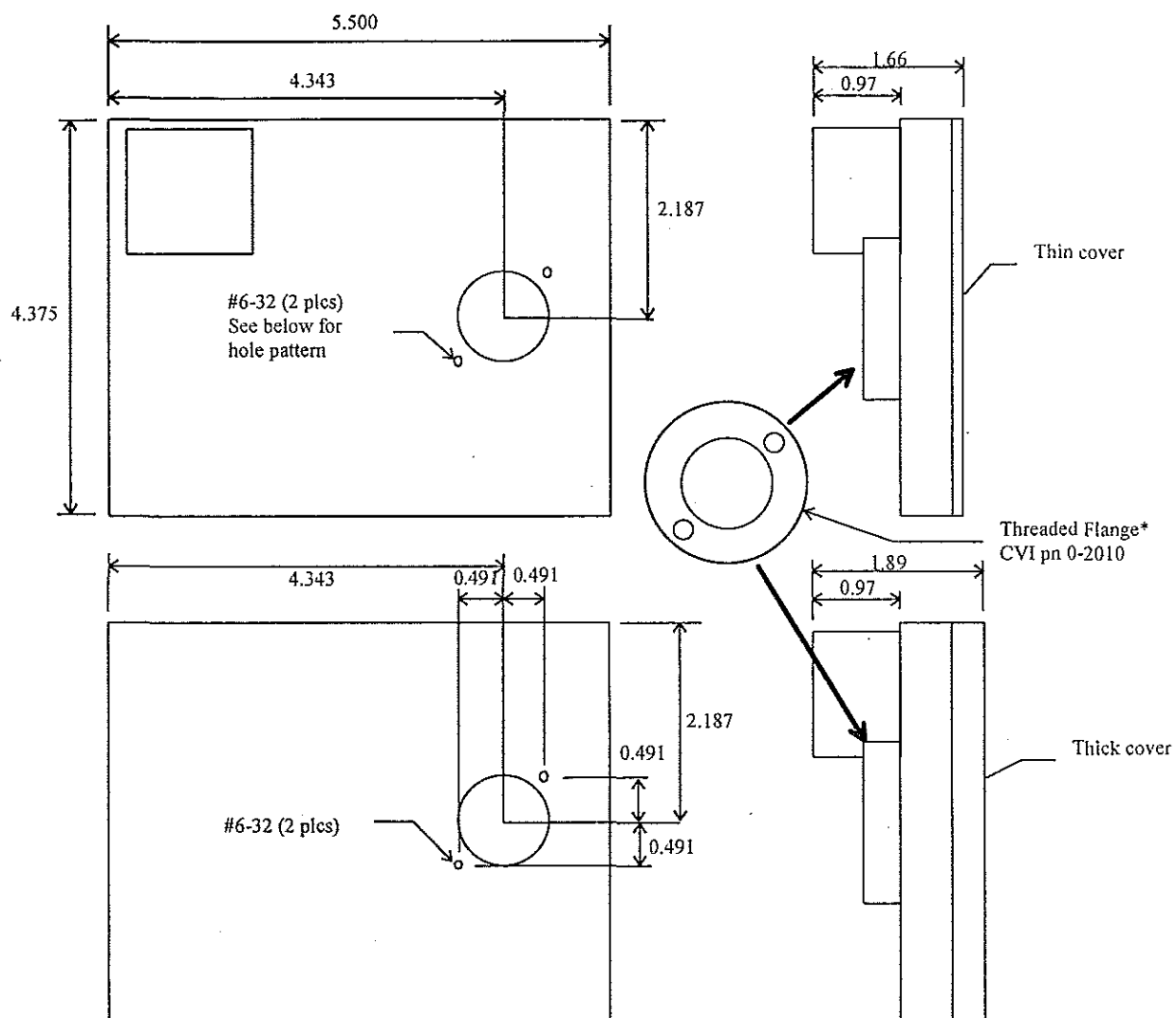
5. SPECIFICATIONS

5.1 Mechanical

5.1.1 AB301, Slit Aperture



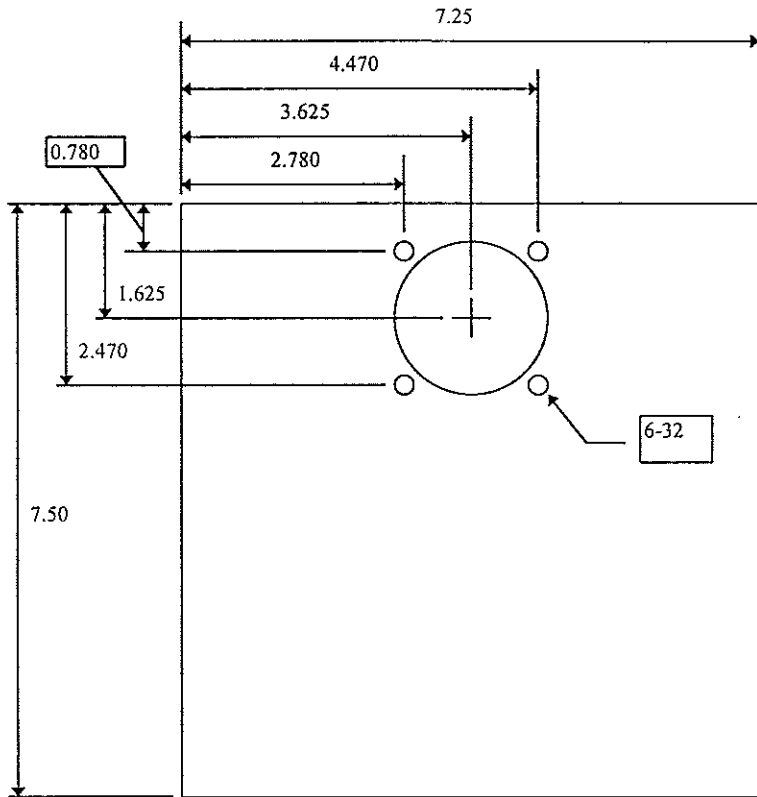
5.1.2 AB301, Clear Aperture



Not to Scale

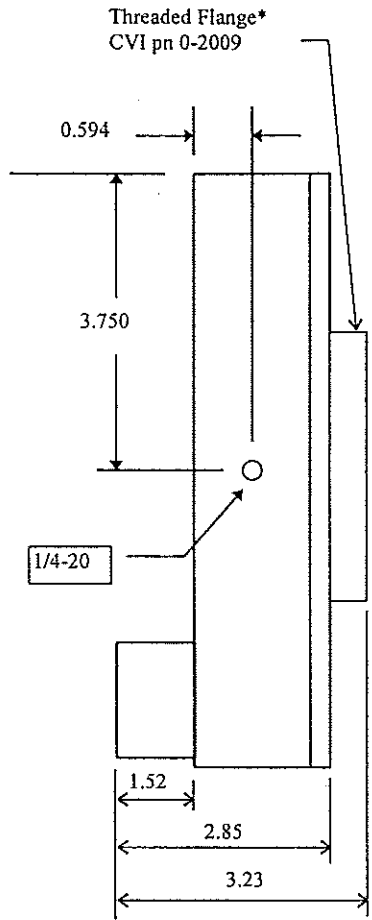
* Threaded flange is Thorlabs 1" tube compatible.

5.1.3 AB302



Not to Scale

* Threaded flange is Thorlabs 2" tube compatible.



5.2 Electrical

5.2.1 RS232 In

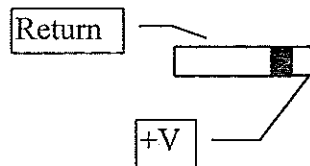
DB9 pin	Signal	Direction
2	RxD	From controller to computer
3	TxD	From computer to controller
5	GND	Signal Ground
7	RTS	From computer to controller
8	CTS	From controller to computer

The AB300-Series controller RS232 port (a DCE device) should be connected to a computer (DTE device) via a standard "straight through" cable (not a null modem type.) Connector is numbered.

5.2.2 Power

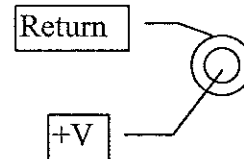
5.2.2.1 AB301

Tip: +5 VDC
Shaft: Return



5.2.2.2 AB302

Center: +12 VDC
Outside: Return



5.2.3 Filter Wheel

Pin out applies at wheel assembly and controller. Connectors are numbered.

DB9 Pin	Function
1	Motor Coil; Blue
2	Motor Coil; Black
3	Motor Coil; Red
4	Motor Coil; Yellow
5	Motor Coil; White
6	Motor Coil; Green
7	Home Contact
8	Home Contact
9	N/C

5.3 Cables

5.3.1 Computer to filter wheel RS232

RS232: DB9-F to DB9-M (CVI pn DK12AT), for 9 pin serial DTE to AB300-Series

From DB9-F pin	To DB9-M pin
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

RS232: DB25-F to DB9-M, for 25 pin serial DTE to AB300-Series

From DB25-F pin	To DB9-M pin
8	1
3	2
2	3
20	4
7	5
6	6
4	7
5	8
22	9

5.3.2 Filter wheel assembly to filter wheel controller

From DB9-F pin	To DB9-M pin
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9

It is recommended this cable use 26 AWG or larger wire, be shielded, and not exceed 12 feet in length.