

INSTRUCTION MANUAL

# **LAMBDA 10**

**OPTICAL FILTER CHANGER**



**SUTTER INSTRUMENT COMPANY**

## TABLE OF CONTENTS

### ■ SECTION I

INTRODUCTION

UNPACKING

SETUP

PAGE 4

### ■ SECTION II

MECHANICAL DESCRIPTION

PAGE 7

### ■ SECTION III

PRINCIPALS OF OPERATION

FRONT PANEL CONTROLS

PAGE 10

### ■ SECTION IV

ERROR DETECTION AND RECOVERY

STEPPING MOTOR OPERATION

PAGE 13

### ■ SECTION V

INTERFACE AND REMOTE CONTROL

INTERFACE MODES

PAGE 17

### ■ SECTION VI

SHUTTER OPTION OPERATION

PAGE 29

**■ SECTION VII**

**DISCLAIMER**

**WARRANTY**

## ■ SECTION I

### *INTRODUCTION*

This is a preliminary version of the Lambda-10 manual. Future versions (which will be available to all users) will contain more detailed instructions and visual aids, as well as all that 'taken for granted' material that is no doubt missing at this writing. In large part, those future editions will be a reflection of the experience of our users and the questions and comments they provide. We urge all investigators using this instrument to contact us regarding the instrument proper and the applications to which it is being applied. From this input will come future improvements to the Lambda-10 and more lucid, thorough editions of this manual.

After reading the manual, should you have ANY questions regarding the setup or operation of the Lambda-10, please contact the factory.

Thank you.

SUTTER INSTRUMENT COMPANY

## *UNPACKING*

The Lambda-10 and associated hardware comes packed in a single carton. The following is a list of those materials that should be found in the carton.

Lambda-10 Filterwheel

Controller

Support Base

2 Support Rods

2 Base Clamps

Lambda-10 to Controller Connecting Cable

Controller to Computer Interface Cable

Power Cord

2 Individual Filter Holders

Set of microscope adapters

Brass spanner wrench

Manual

The Lambda-10 is shipped with nine positions of the wheel containing retaining rings, spacer rings and blanking discs. One of the individual filter holders contains retainer rings, spacer rings and a blanking disc. The second individual filter holder and one position of the wheel (position #1) contain only retainers and spacers.

## *SETUP*

The Lambda-10 should be setup initially on its support base and run in the local mode to determine if it is functioning properly. Begin setup by assembling the two Support Rods to the Support Base. Flats have been machined into the end of these rods to facilitate tightening. The two Support Base Clamps should be used to anchor the base to some convenient location for testing. These clamps are designed to allow anchoring to any of the popular hole patterns found in optical benches.

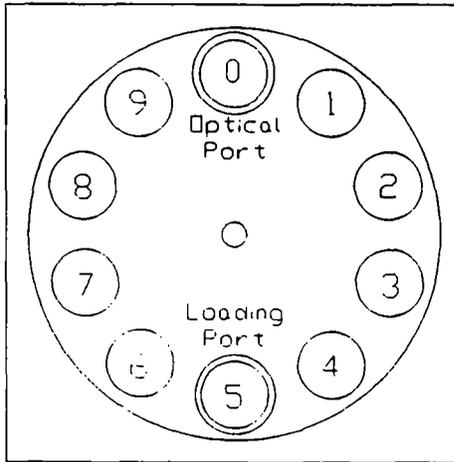
Slide the Lambda-10 Filterwheel down onto the support rods and anchor at a convenient position using the wing screws. Extend the tilt bail under the controller and set it on the bench near the Filterwheel. Connect the two units together with the appropriate cable. (This is a round cable with a DB-15 type connector at each end.) It is important that the screws anchoring the cable ends be tight to insure a good connection for the motor power leads and positioning signals. Check the back of the controller to make sure that it is set for the proper input voltage and that the power switch is in the OFF position; then connect the power cord to an appropriate power source. Since the controller is a microprocessor based device, a surge protector should be provided or the unit should be connected to a conditioned power source.

At this point you will want to go to the Operating section of this manual and verify that the unit is operating properly. However, once proper operation is verified, please return to the next section (Mechanical) for additional information and caveats on setting up the Filterwheel.

## SECTION II

### MECHANICAL

The Filterwheel is a sturdily built device, but extreme care must be taken during the filter loading procedure. Likewise, the support base arrangement is capable of supporting the Filterwheel and a limited additional load.



Port Orientation - Motor Side View

Selected Filter No.	0	1	2	3	4	5	6	7	8	9
Filter at Loading Port	5	6	7	8	9	0	1	2	3	4

The Filterwheel should be positioned approximately in its final location and attached to any other anchored piece of equipment (microscope?). The wing screws should then be tightened to lock the Filterwheel at its final height. If a light source of low mass is to be attached to the Filterwheel, it may be attached directly with no further support. However, if a heavy light source (such as a modern vapor lamp with integral igniter) is to be attached, it should have its own separate support system. All holes in the Filterwheel housing and Individual Filter Holder housings are threaded 1-32 (standard c-mount) allowing direct connection of like threaded units (camera adapters, etc). However, care should be taken that said threaded units not project more than 0.180 inches into any threaded port. Allowing threaded units to penetrate any further could result in contact with the enclosed Filterwheel or filter holders causing damage.

It can not be emphasized enough that loading the Filterwheel requires care and patience. To illustrate the method used to sequester filters in the wheel, one of the Individual Filter Holders has been shipped with inserts identical to those found in the positions around the Filterwheel.

Take the brass spanner wrench provided and insert it into the Individual Filter Holder that contains the blanking disc. There are three items in the filter holder 1) a threaded retaining ring; 2) a spacer ring (to carry the force of the retaining ring to the unthreaded portion at the bottom of the holder); and 3) a blank disc. The threaded retaining ring has two notches cut into it 180 degrees apart. Place the notched end of the brass spanner wrench into the notches and unscrew the ring. Once the retaining ring is removed, the spacer and blank should fall out. One should become familiar with the assembly of this retaining system. In particular, one should have a feel for tightening the retaining ring without pushing on the wrench (not a great deal of force is required). This is important because one cannot push on the Filterwheel in a direction perpendicular to the plane of the wheel without running the risk of distorting the wheel and rendering the position sensing mechanisms useless.

NOTE: It is most convenient to load and unload the Filterwheel and holders with the units flat on a table top. The Individual Filter Holders were intended to hold Heat and Neutral Density filters.

Loading the positions of the Filterwheel is the same procedure as for the Individual Filter Holders, with the additional complication of working through the Filterwheel housing. In order to gain access for the wrench, use a large coin or other suitable implement to remove the two Access Plugs located on the two faces of the Filterwheel housing. **Once again, it cannot be emphasized too strongly that one must take care during the loading and unloading procedure.** Be very careful when reinstalling a retaining ring that it goes in smoothly lest it becomes cross-threaded.

It is convenient to load the Filterwheel with the unit connected and operating. Since the filter being loaded is five positions removed from that indicated by the controller display, the diagram below is provided to clarify the filter position available at the access port.

When installing filters do not overtighten the retaining rings. If a retaining ring has been overtightened, it is possible that when it comes time to remove the filter later, the whole assembly holding the filter could begin to unscrew from the filterwheel. If you find that the filter holding assembly is unscrewing from the wheel, STOP. Turn the spanner wrench around and using the other end, insert it into the slots at the top of the assembly. Now screw the assembly back into the wheel. DO NOT push on the assembly, just screw it in until it seats firmly. It should now be possible to remove the retaining ring. If a filter 0.100" (2.5mm) thick is to be installed, insert a spacer ring before using the retaining ring; this will prevent the retaining ring from running out of threads and binding near the bottom of the filter holding assembly. After identifying the position to be loaded (controller powered up), it is advisable to turn off the controller and check each

filter position after loading to make sure that the retaining ring is not projecting above the top edge of the filter holding assembly.

It may appear that partial disassembly of the filterwheel housing (i.e. removing the coverplate) would facilitate loading filters - **THIS SHOULD NOT BE DONE**; it is just as fast to work through the access ports. **DO NOT REMOVE THE FILTERWHEEL FROM THE MOTOR SHAFT**; the procedure for aligning the filter ports, position sensors, etc. is involved. Further, removal of the wheel from the shaft constitutes abuse of the mechanism and voids the warranty. If you have any questions concerning disassembly of the unit for modification ( in a special situation), please contact the factory first.

One last **VERY IMPORTANT** word concerning the blanking discs. These discs should only be removed from those positions which will contain some other light attenuating device. The blanking discs are intended as a safety device for the occasion where one may be looking into the microscope at a time when the Filterwheel errs. We strongly urge that the blanking discs be used. Please be careful and avoid the possibility of direct visualization of high intensity light.

## ■ SECTION III

### *PRINCIPALS OF OPERATION*

The **LAMBDA 10** is designed to rapidly change optical filters in research applications. Although the **LAMBDA 10** is simple to use, proper operation depends on the user understanding the design and characteristics of the **LAMBDA 10**. This section provides the detailed information needed to begin operation and to control the unit from its keyboard. The Interface and Remote Control Section explains control of the unit from a remote device, as well as supplying some technical details relevant to local operation. Technical details which are not required for operation, but which may help you understand the behavior of the unit are found in Stepping Motor Operation. The primary design goal for the **LAMBDA 10** was to allow 2 optical filters of 1" (25 mm) diameter to be changed as quickly as possible. The final design allows such changes in less than 60 msec. The **LAMBDA 10** is also capable of holding up to 10 filters at a time and operating at 10 different speeds.

When the power is applied the **LAMBDA 10** begins to rotate at a slow rate until a sensor in the mechanical unit signals that filter 0 is in place. The display will then come on with a message such as:

**FILTER CONTROLLER PROGRAM**

**COPYRIGHT SUTTER INSTRUMENT CORP. 1989**

After about a second this display will be replaced by one such as:

**FILTER NO. 0 123 SPEED 1**

**KEYBOARD ON COMPUTER CONTROL OFF**

or

**FILTER NO. 0 123 SPEED 9**

**KEYBOARD OFF COMPUTER CONTROL ON**

The first line indicates the filter that is in position and the selected speed. The filter will always be 0 at this point, and the speed will be the speed last used when the unit was on before. The

three characters **123** are user selected labels that will be explained below. They have no operational role.

The second line of the display indicates if the unit is under control of a remote computer or the **LAMBDA 10** keyboard. If the unit is under remote control, the display may change immediately to reflect the speed and filter commands of the remote computer.

### ***FRONT PANEL CONTROLS***

***RESET*** Pressing the reset button at any time causes the control unit to start the power-up sequence. The sequence starts with a delay, then the old display will remain until the filter 0 is in position, then the power-up routine will rewrite the display.

***KEYBOARD*** There are 16 keys: 10 numerical keys and 6 function keys. Keys are read while the key is pressed but the appropriate action occurs only when the key is released. Keys pressed and released while a move is taking place are ignored. A key pressed while a move is in progress but released after the move is over will read normally. In the remote mode only the **LOCAL** key is active, but holding any other key down may cause the control unit to fail to read new commands from the remote computer. The keys are used as follows:

***0-9*** Pressing a numbered key will select a new filter if the unit is under local control and is not in the process of moving. If the unit is making a move or is under remote control all keys will be ignored. The control unit automatically determines the direction and distance the wheel needs to rotate to place the selected filter in position. The move will begin when the key is released. The number keys are also used with the **SPEED** and **LABEL** keys, as indicated below. When used with those keys there is no effect on the filter selected..

***LOCAL*** Pressing this key puts the unit under local control unless the unit is in the process of making a move during the entire time the key is pressed This is the only key that has an effect when the remote computer control is on.

***ON LINE*** This key transfers control to the remote computer. This can also be done by sending the correct signal from the remote unit as outlined under the interfacing section.

***F1*** Reserved for future use

***F2*** Reserved for future use

***LABEL*** This key is used to label a filter position with a three-digit number, such as the wavelength of the filter loaded in that position. When this key is pressed, assuming that the unit is not making a move and is in the local mode, the following display appears:

**WHICH FILTER DO YOU WISH TO LABEL?**

**VALUE OF THE FILTER (3 DIGITS)**

Press the number key for the filter you wish to label followed by the three digits you want to use. The display will now return to normal. The number entered will be retained, even if the unit is unplugged. The unit may arrive with characters other than digits stored as labels. This is not a malfunction. However, you will be able to enter 3 digit numbers only.

***SPEED*** When this key is pressed the next key pressed will be selected as the new speed. Use only the number keys following the speed key. This key is only read if the unit is not making a move and is in the local mode.

## ■ SECTION IV

### *ERROR DETECTION AND RECOVERY*

The **LAMBDA 10** contains two sensor systems that are used to monitor the actual position of the filter wheel. After the control unit sends the step sequence for a move to a new filter position, it compares the output of these sensors with the expected position. If there is no match, the error recovery sequence is started. First, the wheel is rotated until filter 0 is in position, as at power-up. Then the unit continues to rotate at a slow rate until the correct filter is in position. Signals are available from the interface that reflect the error status, and these are detailed in the interface section.

### *SELECTION OF SPEED*

The **LAMBDA 10** has 10 selectable speeds. When under local control the speed is selected by simply pressing the speed key on the keypad followed by the desired number key. Speed 0 is the fastest speed and speed 9 is the slowest speed. Entering the selected speed under computer control is covered in a separate section.

It is important to remember that the **LAMBDA 10** was optimized for the fastest exchange of 2 filters in research applications. When more filters are mounted, the speed at which any 2 adjacent filters may be exchanged may be reduced. The speed of operation of the **LAMBDA 10** is dependent on the weight of filters and mounting hardware on the moving filter wheel. This weight translates into an inertial load which must be accelerated and decelerated as quickly as possible for each move.

As the inertia increases, the force required to achieve a given acceleration and deceleration also increases. Since the step motor in the **LAMBDA 10** is operated at rated power, an increase in force could only be obtained by using a larger motor. Larger motors, however, have a higher intrinsic inertia. Increasing the motor diameter would require that the filter wheel be expanded and the filters moved out on the wheel so that the optical path clears the motor housing. This also would lead to increased inertia. Increasing the length of the motor would force an increase in the optical path length, which is not desirable. The net result might be a slight increase in speed when fully loaded, but at a cost of slower response when relatively few filters are loaded.

For each combination of selected speed and selected number of positions to move, the **LAMBDA 10** has a predetermined acceleration and deceleration profile. Since the force available is limited, the ability of the **LAMBDA 10** to successfully make the selected move at a given speed will depend on the weight of the filters.

If the weight is slightly excessive for the chosen speed the filter wheel may oscillate instead of stopping quickly when the selected filter has rotated into position. Even though this may not be detected as an error by the control unit, it is advisable to try a slower speed. A slight slowing of the acceleration and deceleration profile may actually allow the move to be completed more quickly by eliminating the oscillations. Naturally, if your application is not sensitive to small oscillations of the filter, this becomes a less important consideration.

If the weight is clearly excessive for the chosen speed the filterwheel may not always stop at the correct position on every move. Each time the control unit detects that a move was made incorrectly, it will enter the error recovery mode. If this occurs frequently, select a slower speed. Make sure you also address the question of the interval between commands below before you become very serious about the speed of operation.

Selecting a slower speed will generally eliminate errors caused by excessive load weight, but there are other considerations. For a given load weight some of the slow speeds may excite oscillations during movement. This is a result of stepping the motor near its resonant frequency. Selecting a higher or lower speed will generally solve this problem.

It is also possible to produce errors by commanding a new move too quickly after the previous move. If errors are detected with some frequency, it is wise to consider this possibility before addressing the selected speed. Test the unit for errors with a long time interval (1 second) between moves. If the errors are eliminated, the speed alone is not the problem. These errors reflect oscillations produced by a previous move which were carried over to the next move. The wheel was commanded to move before the oscillations had damped out.

In most applications, there is a required pause at each selected filter position in order to obtain data. In most cases, this should allow enough time delay between moves. There is a tendency for users to test the **LAMBDA 10** in their system using little, if any, delay between commands, even though a pause occurs during an actual experiment. Always determine the proper speed using the load weight and timing intervals that will be used in your experiments.

With an optimal combination of speed and load-weight, it may be possible to give a new command as soon as the last move is complete. On the other hand, if the selected speed is improper for the load, as outlined above, the oscillations may be sufficient to carry-over even if the next command is delayed over 50 msec from the end of the previous move.

If your application requires the delay between moves to be very slight, make sure the speed you have chosen allows the wheel to stop quickly, without oscillation. If possible, experiment with small changes in the delay to find the best value and then keep the delay at that value.

## ***STEPPING MOTOR OPERATION***

Stepping motors are not as familiar to most people as the common DC motor, but there are some similarities. The DC motor consists of an armature and an electromagnet mounted on a rotating shaft located inside a permanent magnet. Current is supplied to the electromagnet through brushes that rub on contacts on the armature. When the electromagnet is energized, the armature rotates to align the poles of the electromagnet with the opposite poles of the permanent magnet. Of course, before this can occur, the rotation of the armature changes the contact plates rubbing on the brushes so that the current is reversed. This causes the poles of the electromagnet to reverse, establishing a force for continued rotation. This switching action is called commutation.

In stepping motors the rotating element, called a rotor, is generally a permanent magnet while the fixed element, the stator, is the electromagnet. The key difference between stepping motors and DC motors is the method of commutation. The DC motor commutates automatically as it rotates. Thus the timing of the commutation is determined by the speed of rotation, which may vary with the load or applied power. The commutation of the stepping motor is set by external electronics, forcing the motor to rotate at a predetermined rate. If the load is such that the motor does not have the force to produce the correct rate of rotation, the rotation will become erratic and may even reverse.

The force exerted between two magnet poles is proportional to the square of the distance between the poles. A motor with a single electromagnet and only two poles would exhibit considerable loss of power when the distance between the poles of the permanent magnet and the electromagnet was greatest. It is understandable that, in most practical DC motors, the armature has more than 2 poles. This allows the commutation to occur over a smaller angle of rotation, so that the active poles can always be relatively close to the poles of the permanent magnet.

Stepping motors are also made with multiple poles on both the rotor and stator. The exact arrangement determines the number of steps per revolution. The motor used in the **LAMBDA 10** has 200 steps per revolution (1.8 degrees per step). There are usually two windings in the stator, and a single step of rotation is produced by reversing the current on one of the windings. Reversing the current on the second winding will then produce another step. If the first winding is then reversed again, returning to its original value, a third step will result. Finally, reversing the second winding, so that both windings are back to their original state, will produce a fourth step. This pattern may then be repeated to continue rotation in the same direction. Reversing the sequence produces steps of rotation in the opposite direction.

The rate and distance of rotation is determined by the rate and number of commutation steps. As long as the current is held constant in both windings the rotor will not rotate. This makes the stepping motor ideal for producing fast start and stop movements. There are some limitations that

should be considered. Given that there are only four states of the control electronics (2 polarities for each of the 2 windings) and 200 steps per revolution, it follows that, for each of these 4 states, there are 50 possible rotary positions. In order to establish the absolute position an external sensor must be added. Absolute position only needs to be determined once so long as the subsequent moves occur without errors.

Correct operation of stepping motors is not always obtained. If the motor lacks the torque to accelerate and decelerate the load at the rate indicated by the control electronics, the resulting move may differ from that commanded by the control electronics. For instance, if enough external force is applied, the motor shaft may be rotated even though the command signal has not changed. For small rotations (less than 3.6 degrees), the motor shaft will return to the correct position if the force is removed. For larger rotations, the shaft will stop at the nearest of the 50 correct positions for that command signal when the external force is removed.

A similar effect may occur if the motor is rotating a load when the stop command requires an abrupt stop. Momentum tends to force the wheel to rotate beyond the correct position. If the overshoot is slight, the motor may settle into the correct position after oscillating briefly around it. If the motor overshoots by the equivalent of 2 steps or more, it will continue ahead to the next of the 50 positions for that command signal.

Errors such as these may occur when the **Lambda 10** is operated at too high a speed for a given load, or when the wheel is rotated by an external force as may happen during the loading of the wheel. In order to detect such errors, an optical sensor has been mounted in the Filterwheel. This sensor is activated whenever a filter is correctly positioned in the port. An up/down counter counts the output of this sensor which was set to zero when filter zero was detected at power up. If the sensor does not indicate correct filter position, or if the counter has the wrong count at the end of a move, the control unit will identify this as an error.

## ■ SECTION V

### *INTERFACE AND REMOTE CONTROL*

The **Lambda 10** has been provided with an 8 bit parallel input port and 2 output lines in order to allow easy control from a remote computer or other source of logic-level signals. All interconnections are made through a 16 pin connector on the back of the control unit.

#### *INPUT LINES*

The 8 input lines are divided into two groups of 4. One group selects the desired filter while the other selects the speed of movement used in advancing to the next filter. Each group is treated as a four-place binary number, with the values 0 through 9 corresponding to the selected value. Binary codes with decimal values greater than 9 are ignored, with the exception of special command codes.

When the control unit is in the on-line (remote) mode, the 8 input lines are examined at 50 microsecond intervals. If the value of the speed or filter indicated by the input lines has changed from the previous time the lines were read, and if the new values are in the range of 0 to 9, the control unit will respond as commanded.

#### *OUTPUT LINES*

The first response of the unit to changing input is to set the **BUSY** output line high and write the new data to the display. The **BUSY** line will stay high until the control unit is ready to accept new data. The **BUSY** line stays high for 20msec after the move is complete. In the case of changes in speed value only, this will be 4 to 5 msec. If a new filter is selected, the **BUSY** line will stay high until the move is complete. If both a speed and a filter change have occurred, the speed change will be processed and the unit will then read the input again. If the same speed value is still there the new filter value will then be read and acted upon. It is likely that the speed will never be changed in a given session, so this should not cause a problem.

The status of the 8 input lines is not checked while the **BUSY** line is high! There is a short period (as much as 2 msec) between the time the control unit detects a new value and sets the **BUSY** line high. Thus checking the status of the **BUSY** line within about 2 msec of changing the input lines may give a misleading result.

The second output signal is the **ERROR** line. The control unit uses optical and magnetic sensors to verify that the correct filter is in position. If a command for a new filter is received and the cor-

rect new filter is not in position at the end of the change sequence, the sensors will indicate this to the control unit. The unit will then assert the error signal and enter the error recovery mode. The **BUSY** line will also be set high. In the error mode, the wheel is rotated continuously until the sensors indicate that filter zero is in position. The wheel will then make a slow move to put the correct filter in position. The **BUSY** line will stay high until the correct filter is in position, but the **ERROR** line will stay high only until filter zero is in position.

The various input and output signals are available at a 16 pin header on the rear of the control unit. As seen from the rear of the control unit the pins are

	2x	4x	6x	8x	10x	12x	14x	16x
	1x	3x	5x	7x	9x	11x	13x	15x
1	NC							
2	NC							
3	NC							
4	NC							
5	GROUND							
6	GROUND							
7	ERROR LINE OUTPUT							
8	BUSY LINE OUTPUT							
9	0 BIT PARALLEL PORT				(0 BIT FILTER)			
10	1 BIT PARALLEL PORT				(1 BIT FILTER)			
11	2 BIT PARALLEL PORT				(2 BIT FILTER)			

12	3 BIT PARALLEL PORT	(3 BIT FILTER)
13	4 BIT PARALLEL PORT	(0 BIT SPEED)
14	5 BIT PARALLEL PORT	(1 BIT SPEED)
15	6 BIT PARALLEL PORT	(2 BIT SPEED)
16	7 BIT PARALLEL PORT	(3 BIT SPEED)

The pins 1-4 are connected internally and reserved for future use. At present these lines **MUST NOT** be connected to an external cable as this will cause the control unit to malfunction. If you provide your own 16 line ribbon cable, make sure these 4 lines are cut at a point as close to the header as possible.

### *INTERFACE MODES*

#### *Input lines only.*

It is quite possible to control the filter position using only the 8 input lines. Reading the status of the **BUSY** and **ERROR** lines would not be needed in many cases ( as long as the weight of the filters mounted on the wheel does not change) since the operation of the wheel is very consistent. It is a simple matter to then determine the fastest speed that allows operation without errors, so that the **ERROR** line might be ignored.

If we can ignore the possibility of errors, the **BUSY** line would then only indicate that a move is in progress or that the unit is in the process of making the initial move to filter zero on power-up. The later case only requires a few seconds delay in sending commands to the control unit on power-up, which should be easily accommodated in any system. If a command is sent to the control unit for a new filter, so long as the unit is ready, the time to recognize the new command and complete the move should be consistent within less than a millisecond. Thus the control program in the host computer could simply allow for the time required to make the change rather than looking at the status of the **BUSY** line. This would be particularly easy in a system that required switching only between adjacent filter positions, since the time for each change would be the same.

In some applications not all of the 8 input lines will have to be connected to the host computer. It is likely that the same speed will be used at all times with a given set of filters. Thus the 4 lines that control speed could be directly wired so that the correct speed was always set. If desired, the

direct wiring could be through a switch so that the speed could be changed . If fewer than 9 of the ten filter positions will be used, it may be possible to direct wire some of the filter control lines as well. With only 1 line it is possible to select between 2 positions, 2 lines allows 4 positions, and 3 lines allows selection of 8 positions. In the extreme case this means that, in a system requiring switching between only two filter positions, a single logic level input line from the host computer to the control unit, together with direct-wired connections to the other 7 lines, would be sufficient.

### *Input lines and BUSY line.*

The conditions that cause the **BUSY** line to be asserted are noted above. Reading the status of the **BUSY** line would allow the host computer to determine if the unit is ready to receive a command, if a command is received, and when the new filter has been placed in position. Obviously, it is also possible for the host computer to use the **BUSY** line simply to verify that the unit is in the expected mode of operation at any point in a predetermined time sequence of commands. Thus the host computer would (as in the 8 input lines only example above) control the timing of events based on the expected behavior of the **Lambda 10**, and the **BUSY** line would be checked at certain points in the program to verify that it was in the expected state. This allows the quickest start of data collection since the **BUSY** line remains high for 20 msec after the move is complete.

Alternatively, the status of the **BUSY** line could be used to determine the timing directly. For instance, the signal that a move is complete (the dropping of the **BUSY** line) could be used as the triggering event for the opening of a shutter or for the beginning of data collection. If the host computer is programmed to wait until the **BUSY** line goes low before continuing, the time required for recovery from any errors would be automatically accommodated . The **BUSY** line can even be used as an indication that an error has occurred. If the **BUSY** line fails to go low at the expected time after a new command was issued, it may be assumed that the control unit has detected an error and is in the process of recovering.

### *Input lines, BUSY line and ERROR line.*

When the **input**, **ERROR** and **BUSY** lines are all used, the **input** and **BUSY** lines can be used as above. The **ERROR** line normally would not be used to determine timing of program execution in the host computer, since it is active only after an error and even then it is not held high during the entire error recovery process.

The use of the **ERROR** line will depend on two factors: First, what are the consequences of an error? Continuing normal program execution in the host computer during recovery from an error might cause problems such as wasted data storage capability, exposure of sensitive photodetec-

tors or the experimental preparation to strong light, or recording of faulty data. The other factor is the burden placed on the host computer in checking the **ERROR** line. The **ERROR** line will go high only after a move has been completed incorrectly, and will remain high only during part of the recovery from that error. Thus the host computer would have to read the status of the **ERROR** line just at the point in time at which data collection should begin. If the **ERROR** line acts via an interrupt, these problems might be eliminated.

### *CONNECTING TO A PARALLEL (PRINTER) PORT*

The standard PC parallel port provided for printer control may be used to control the **LAMBDA 10**. This, however, does not mean that instructions can be sent to the **LAMBDA 10** in the same manner that they are sent to a printer. A printer is designed to print at its maximum rate, and if the computer generates output at a higher rate, the printer will store the output in a buffer until the mechanism is ready. If the buffer becomes full, the printer will indicate to the computer that it is not ready to accept more data. It follows that there is not a tight time relationship between the output of a command from the computer and the execution of that command by the printer. The time relationship between output of data from the computer and response by the **LAMBDA 10**, in contrast, must be very tight. Thus it is understandable that the mode of use of the parallel port will be somewhat different for a printer and the **LAMBDA 10**.

The details of the input and output signals were described above. Now we will discuss the way in which these signals may be used to control the **LAMBDA 10** from a PC/XT or AT compatible computer's parallel port. This information is provided as a limited example and is only intended to show the elements useful in controlling the **LAMBDA 10**. The user's individual requirements, hardware, and software will dictate the actual program steps to be used. The approach in this example does involve direct control over the state of input and output ports. The example programs have been run successfully on computers from several manufacturers without encountering any difficulties. However, given the wide range of hardware and software in use, it is impossible to anticipate all possible problems. **Thus Sutter Instrument Company cannot be responsible for any damages that might result from use of the example programs.** The user should be particularly careful that the address used for each of the various I/O ports is correct for the user's system.

The parallel (printer) port is actually composed of three I/O ports at three consecutive addresses. The first address corresponds to an 8 bit output port, the second to a 4 bit input port, and the third to a 5 bit input/output port. The first two ports are used in our example. The address of the first (output) port, which is the base address, should be one of three possible values, depending on the individual system. If the printer adapter is located on a display adapter, it will be called

# FILTER

	0	1	2	3	4	5	6	7	8	9	
S	0	0	1	2	3	4	5	6	7	8	9
P	1	16	17	18	19	20	21	22	23	24	25
E	2	32	33	34	35	36	37	38	39	40	41
E	3	48	49	50	51	52	53	54	55	56	57
D	4	64	65	66	67	68	69	70	71	72	73
	5	80	81	82	83	84	85	86	87	88	89
	6	96	97	98	99	100	101	102	103	104	105
	7	112	113	114	115	116	117	118	119	120	121
	8	128	129	130	131	132	133	134	135	136	137
	9	144	145	146	147	148	149	150	151	152	153

TABLE 1: DECIMAL DATA VALUE TO BE OUTPUT

# FILTER

	0	1	2	3	4	5	6	7	8	9	
S	0	0	1	2	3	4	5	6	7	8	9
P	1	10	11	12	13	14	15	16	17	18	19
E	2	20	21	22	23	24	25	26	27	28	29
E	3	30	31	32	33	34	35	36	37	38	39
D	4	40	41	42	43	44	45	46	47	48	49
	5	50	51	52	53	54	55	56	57	58	59
	6	60	61	62	63	64	65	66	67	68	69
	7	70	71	72	73	74	75	76	77	78	79
	8	80	81	82	83	84	85	86	87	88	89
	9	90	91	92	93	94	95	96	97	98	99

TABLE 2: HEX DATA VALUE TO BE OUTPUT

LPT 1 and will have a base address of 956 (3BC H). If there is no active printer adapter on a display adapter card, the primary printer adapter card will be called LPT1 and will have a base address of 888 (378 H). A second printer adapter card in the same system will be called LPT2 and will have a base address of 632 (278 H). Check the documentation of your system and the jumpers on the various adapter cards to be sure of the base address on the adapter you wish to use.

The connections to the parallel port are made through a DB 25 connector on your computer. The cable supplied with your unit may be used to connect the control box to your computer's parallel printer port. As you view the rear of the control box, the 16 pin header for the interface is on the bottom right. The interface cable has four wires cut and removed from the 16 pin plug. The 16 pin plug is pressed onto the header pins with the cut wires to the left. The 25 pin plug will mate with a standard IBM PC type parallel printer port. **Make sure you connect to a parallel port and not a serial port.** Since the control unit connector is a 16 pin header, you must either use the cable supplied with the unit or construct your own cable.

The pin assignments for the parallel port socket are, as viewed from the plug side:

13X	12X	11X	10X	9X	8X	7X	6X	5X	4X	3X	2X	1X
	25X	24X	23X	22X	21X	20X	19X	18X	17X	16X	15X	14X

**DB 25 CONNECTOR**

**SUGGESTED 16 PIN HEADER CONNECTION**

1	STROBE	NC	
2	0 BIT DATA	9	0 BIT (FILTER)
3	1 BIT DATA	10	1 BIT (FILTER)
4	2 BIT DATA	11	2 BIT (FILTER)
5	3 BIT DATA	12	3 BIT (FILTER)
6	4 BIT DATA	13	4 BIT (0 BIT SPEED)
7	5 BIT DATA	14	5 BIT (1 BIT SPEED)
8	6 BIT DATA	15	6 BIT (2 BIT SPEED)
9	7 BIT DATA	16	7 BIT (3 BIT SPEED)
10	ACK	NC	
11	BUSY	8	BUSY LINE OUTPUT
12	PE	7	ERROR LINE OUTPUT
13	SELECT	NC	
14	AUTOFD	NC	
15	ERROR	NC	
16	INIT	NC	
17	SLCT IN	NC	
18	GROUND	NC	
19	GROUND	NC	
20	GROUND	NC	
21	GROUND	NC	
22	GROUND	NC	
23	GROUND	NC	
24	GROUND	5	GROUND
25	GROUND	6	GROUND

Commands may be sent through the printer adapter by using an **OUT** command from Basic, or an equivalent command from another language. In *Microsoft QuickBASIC* the form of the command is:

**OUT** *port, data*

Where *port* would be 956, 888, or 632 depending on your system, and *data* would be a number indicating the desired speed and filter number.

The number to use for *data* can be generated from the desired speed and filter as follows:

$$data = filter + (16 \times speed)$$

Thus, to command a move to filter 1 at speed 1 using a printer adapter having a base address of 888, you would use the following command:

**OUT 888, 17**

The various decimal values to use for particular combinations of speed and filter number are listed in Table 1. Table 2 shows the values in Hex. All other values, except as listed below, will be ignored by the control unit.

#### *SPECIAL CODES:*

**238** (EE H, 11101110 binary) When the control unit is in the keyboard mode, the unit will be forced to go on-line if this code is sent.

#### *READING THE STATUS OF THE LAMBDA 10*

The section titled interface modes has a detailed explanation of the **BUSY** and **ERROR** signals. That section also describes some of the factors to consider in deciding whether to read the **BUSY** and **ERROR** signals. If you wish to read the status of the unit from your PC, this can be done through the 4 bit input port on the printer adapter. The address of this port will be the base address of the printer adapter plus 1.

ADDRESS

	<u>BASE</u>	<u>INPUT PORT</u>
LPT1 ON GRAPHICS ADAPTER	956 (3BC HEX)	957 (3BD HEX)
LPT1 ON PRINTER ADAPTER	888 (378 HEX)	889 (379 HEX)
LPT2 ON PRINTER ADAPTER	632 (278 HEX)	633 (279 HEX)

The 4 bit input port uses the 4 most significant bits: 4,5,6 and 7. Bits 0,1,2 and 3 are not used by the printer adapter and will normally read as 1. Bit 5 (pin 12 on the DB 25 connector) is normally used to indicate paper empty. Bit 7 (pin 11 on the DB 25 connector) is normally used to signal printer busy. In this example we also use Bit 5 to indicate error and bit 7 to indicate busy, although the polarity of these signals may differ from the corresponding printer signal.

With these connections the input port can be read using *Microsoft QuickBASIC* as follows:

```
BUSY: OUT 890, 16                'DELAY TO ALLOW
    FOR U = 0 TO 5                'BUSY TO GO HIGH
    NEXT U
LOOK: D2 = INP(889)              'READ PORT 889
    IF D2 = 127 THEN GOSUB EFLAG 'ERROR FOUND
    IF D2 <> 223 THEN GOTO LOOK 'STILL BUSY SO LOOP
    RETURN                        'NOT BUSY SO RETURN
EFLAG: BEEP                      'INSERT YOUR OWN
    RETURN                        'ERROR RESPONSE
```

Obviously the value 889 corresponds to the input port address. The value 127 (7F HEX, 01111111 BINARY) which causes a branch to an error routine indicates that all bits except the busy bit (bit 5) were read as 1 (high). The error bit will read as 1 only when the **ERROR** signal is high, indicating that an error has occurred. The busy bit (bit 7) will read as 0 (low) even though the **BUSY** signal is high because the printer adaptor hardware inverts the data for bit 7. The value of 223 (DF HEX, 11011111 BINARY) which must be obtained to proceed to the RETURN statement, indicates that all bits except the error bit (bit 5) were read as 1 (high). This indicates that both the **ERROR** and **BUSY** signals were low and the **LAMBDA 10** is ready to accept a new command.

The example given will loop until the **BUSY** signal is low. When developing a program it is advisable to insert an escape device in this loop such as:

```
IF INKEY$ <> "" THEN RETURN
```

Pressing any key will then cause a return from the loop. If you are stuck in a tight, endless loop without an escape device you will have to reset the computer. This statement may be removed once the program is operating correctly.

An alternative command to use in *Microsoft QuickBASIC* is **WAIT**. This statement has the form:

**WAIT** *port*, *ANDdata*, *XORdata*

Where the *port* is the address of the input port, and *data* is an integer in the range 0 to 255. When the **WAIT** statement executes, program execution halts until the input port indicated by *port* reads as the indicated by *data* expressions. The *ANDdata* expression is evaluated as an AND operation. Thus we may test to see if the **BUSY** line is set as follows:

**WAIT** 889, 128

The program will suspend operation until the value read for the most significant bit is 1. Since the printer adapter inverts this bit, this condition will be met only when the **BUSY** line is low. An XOR operation can be used as well, as noted in the *Microsoft QuickBASIC Language Reference*. The **WAIT** statement will XOR the read data before the AND operation if a value for *XORdata* is provided. Consider the statement:

**WAIT** 889, 128, 128

The XOR of the read value of the port with 128 will invert the most significant bit. The program execution would thus be suspended until the **BUSY** signal went high, indicating that the unit was busy. Using the 2 statements in series would insure that the control unit had time to assert the **BUSY** line in response to a command before the program began to test to see if the **BUSY** line had come back down:

*WAIT 889, 128, 128*

*WAIT 889, 128,*

**THERE IS AN IMPORTANT CONSIDERATION WHEN USING THE *WAIT STATEMENT!*** The *WAIT* statement seems to be ideal because it provides a fast and easy way of directly testing input port bits. However, if the input port does not have the correct value, and you are unable to force the port to the correct value, you will need to reboot the computer. This should not be a significant problem in routine operation of a system, but it can provide considerable frustration when developing and testing the control program. Test the *WAIT* function with simple programs before you use it. Make sure that you will not lose important data if you are forced to reboot.

## ■ SECTION VI

### *SHUTTER OPTION OPERATION*

The shutter is simple to operate, but there are some important limitations on its operation. Please read this section carefully and completely.

If your **Lambda-10** control unit has the built-in shutter driver, you will see a toggle switch just to the left of the keypad. This switch is labeled **AUTO OFF OPEN**. If you have a built-in driver, the shutter will be connected through the 15 pin driver cable that controls the motor unit. The remote input is brought in through a BNC connector on the rear of the control unit.

If you have a stand alone shutter driver supplied by the Sutter Instrument Company, check to see that the line voltage selector on the back of the unit is set for the voltage you will be using. Line frequency may be 50 or 60 Hz. Maximum current draw is about 0.2 amps for 240V 50 Hz and 0.35 amps for 120V 60 Hz. The shutter cable must be connected to the control unit using the 4 pin connector. *Note: There is one pin that is larger than the other three pins; make sure that the connector is oriented so that the large pin mates with the corresponding large socket in the receptacle.* This pin is connected to ground and extends this connection to the shield in the cable. The shield in turn, is connected to the metal shutter so that it is also grounded. The 2 adjacent pins carry the drive current for the shutter. As much as 60V can appear on either of these pins, so always attach the connector with the power off.

If the unit has been connected to the line voltage (AC mains), it may be placed in operation by moving the power switch to ON. Before the unit is turned on, we suggest that the AUTO switch be placed in its center (OFF) position, which leaves the shutter closed. Once the power has been turned on, you can test the shutter by moving the AUTO switch down; this should cause the shutter to open. Return the switch to the center position and the shutter should close.

**AVOID LEAVING THE SHUTTER OPEN FOR EXTENDED PERIODS IF IT IS NOT REQUIRED.** The shutter driver must dissipate a significant amount of heat when the shutter is held open. Since the entire case acts as a heat sink, you may notice that the controller case becomes quite warm to the touch. So long as the ambient temperature is in the normal range, the power transistor should be kept well within its rated temperature by this heat sinking, even when the shutter is open for long periods of time. However, repeated cycles of heating and cooling within the rated range of operation of a transistor can cause a failure of that transistor. Therefore it is prudent to avoid leaving the shutter open for extended periods unless it is required.

The shutter can be controlled from almost any source of logic level signals. The BNC connector on the rear of the control box is used for the input. This input is active when the AUTO switch

is in the up (AUTO) position. If the input level is between 0 and 1 volts, the shutter will be closed. Increasing the input voltage to greater than 2 volts will cause the shutter to open. **Avoid input voltages between 1 and 2 volts; the state is not defined in this range. Avoid inputs below 2 volts.** The shutter may open if the input goes below -.2 volts. The maximum input is 15 volts. Within the range of 0 - 15 volts the input presents a load of about 100K ohms. Protective diodes limit the input voltage inside the unit to this range, and a 1K ohm resistor in series with the input limits the current if the protective diodes conduct.

**LIMIT THE FREQUENCY OF OPERATION TO UNDER 20 HZ.** When the input signal goes high the shutter controller delivers 60 volts to the shutter for a brief period to rapidly open the shutter. The output then drops to a holding level of about 5 volts. The shutter coil generates little heat when held open, but, as the interval between successive shutter openings decreases, the average power that must be dissipated by the shutter coil will increase. If a sufficiently high rate of opening is maintained for an extended period, the coil temperature will rise until its insulation melts. The shutter is able to respond mechanically up to 50 Hz, and the controller can go even faster. The shutter coil, however, may reach its operational temperature limit if the 20 Hz limit is exceeded for an extended period.

### **PERFORMANCE**

According to the shutter manufacturer, the typical time to open to 90% is 6 msec. Once the shutter starts to close, it takes 4.2 msec to go from 10% closed to completely closed. The latency from the command to close until the 10% closed status is not indicated. These specifications are for Teflon coated blades.

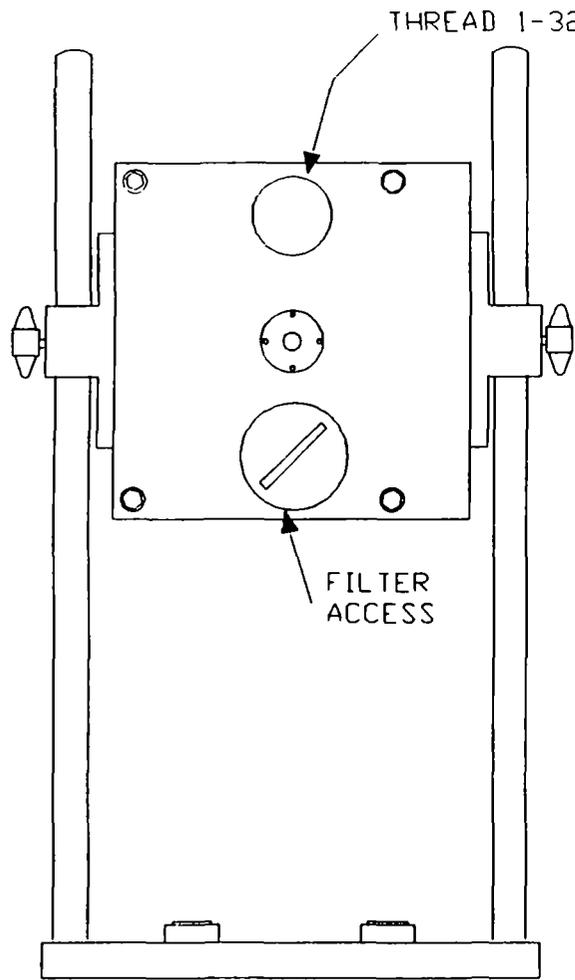
We have tested the units with our controller as follows: The shutter aperture was uniformly illuminated and the light passing through the shutter was focused on a high speed silicon photo diode. The photo diode was connected to a current to voltage converter circuit to obtain the best frequency response. The shutter command input and the photo diode signal were displayed on a *Gould 1600 Series* digital oscilloscope so that time intervals could be measured to within .01 msec. Typical delays in opening to 100% were 7.5 msec. Delays for complete closing, from the command to close, were typically 11.5 msec.

The shutter usually takes 4 to 5 msec from the first detectable change in aperture to completion of that change. The shutter remains completely closed for 2 to 3 msec after a command to open and remains completely open for 6 to 7 msec after a command to close. If the command pulse to open lasts less than 7 msec, the time to open may be increased. Command pulses of less than 5 msec may fail to open the shutter.

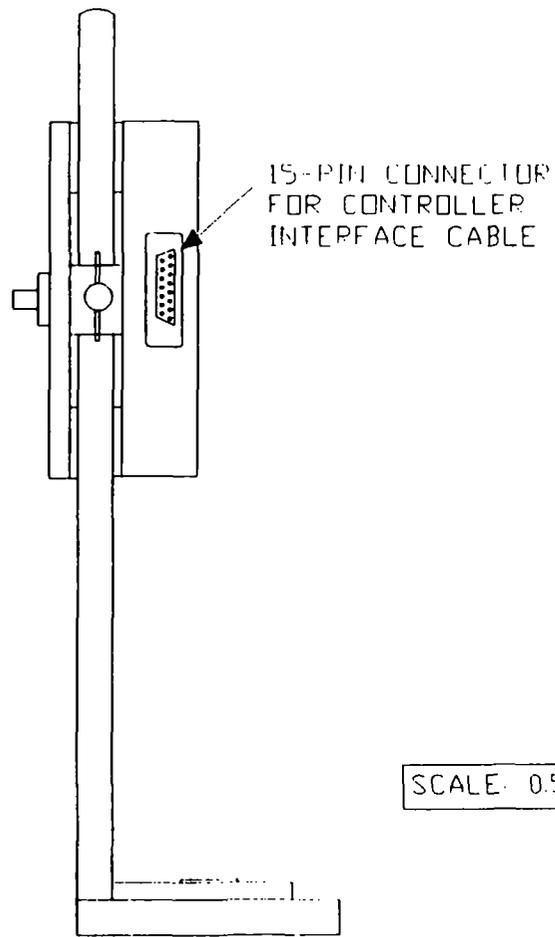
The shutter timing should be considered in establishing the timing of the control signals. Given the information above, it is clear that the command signals should be changed prior to the exact time at which you wish the shutter to open or close. If your timing requirement is very sensitive, it is best to determine the required time offset empirically.

The effective duration of an exposure is also effected by these timing considerations. If we consider the time from 50% open to 50% closed as the equivalent exposure time, the effective exposure will be about 3 to 5 msec greater than the duration of the command pulse.

The shutter option for the **Lambda 10** may be controlled form the parallel printer port of an IBM or similar personal computer. We have used pin 14 of the 25 pin connector, which is the autofeed pin. This pin is controlled by bit 1 ( where bit 0 is the least significant bit) of port 890 (assuming 888 is the base port). This bit may be set to open the shutter, from *Basic* , with a command such as **OUT 890, 2** and may be cleared to close the shutter as **OUT 890, 0**. We have found that some computers will not allow inputs on the parallel to be read unless bit 4 of port 890 has been set high. If you find this to be the case, **OUT 890, 18** can be used to open the shutter and **OUT 890, 16** to close the shutter leaving bit 4 high in either case.



FRONT VIEW  
(MICROSCOPE SIDE)



RIGHT SIDE

SCALE: 0.5"=1.0"

## LIMITED WARRANTY

Sutter Instrument Company, a division of Sutter Instrument Corporation, limits the warranty on this instrument to repair and replacement of defective components for one year after the date of shipment, provided the instrument has been operated in accordance with the instructions outlined in this manual.

Abuse, misuse or unauthorized repairs will void this warranty.

Limited warranty work will be performed only at the factory.

The cost of shipment both ways is to be borne by the user.

This instrument is designed for use on animal tissues. It is not intended to be used and should not be used in human experimentation or applied to humans in any way.

The limited warranty is as stated above and no implied or inferred liability for direct or consequential damages is intended.

## *DISCLAIMER*

The Optical Filter Changer Lambda 10 is designed for the specific use of changing filter positions on a wheel and no other use is recommended.

This instrument was designed to be used in a laboratory environment. It is not intended to be used and should not be used in human experimentation or applied to humans in any way. This is not a medical device..

Do not open or attempt to repair the instrument without expressed and explicit instructions from the Sutter Instrument Company. Extreme heat and high voltages are present and could cause injury.

Do not allow unauthorized and/or untrained operatives to use this device.

Any misuse will be the sole responsibility of the user/owner and Sutter Instruments assumes no implied or inferred liability for direct or consequential damages from this instrument if it is operated or used in any way other than for which it is designed.