LabVIEW Driver Supplement

to the

Spectrometer Control

Manual

Part Number 80123A

JOBIN YVON - SPEX

Instruments S.A., Inc.

Information in this document is subject to change without notice and does not represent a commitment on the part of the vendor.

About the Manuals

You may have more than one manual, depending on your system configuration. To find the manual that has the information you need, these guidelines may help.

- Each manual generally covers a product and the features and accessories peculiar to and/or contained within that product.
- Accessories that can be applied to other products are normally covered by separate documentation.
- Software that is exclusively used with one instrument or system is covered in the manual for that product.
- Software that can be used with a number of other products is covered in its own manual.
- If you are reading about a product that interacts with other products, you will be referred to other documentation as necessary.

This manual covers both the LabVIEW spectrometer drivers and CCD drivers. These products are provided separately therefore you may not have both sets.

Table of Contents:

Overview of LabVIEW drivers.

The LabVIEW graphical programming system for data acquisition and control, data analysis, and data presentation is popular with many instrument user-programmers. ISA offers a driver package to provide control and acquisition of ISA Spectrometer systems in the LabVIEW environment.

Many industrial I/O devices for process monitoring and control applications are also supported with LabVIEW drivers. Such devices as sensors, signal conditioners, and programmable logic controllers can be interfaced along with ISA components. Now you can easily control your Jobin-Yvon/SPEX spectrometers and related ISA accessories as well as other laboratory instrumentation such as temperature controllers or positioners from a single computer. LabVIEW can be run from a number of computer operating platforms such as Windows™, Windows™ 95, Windows™ NT, Macintosh™, or Sun Solaris™.

Please note that the driver Virtual Instruments (VI's) provided in this package are intended to serve as building blocks for user specific application development. A fundamental understanding of LabVIEW is necessary to successfully implement the ISA LabVIEW drivers. The documentation included with the LabVIEW development package is recommended as a starting point for those unfamiliar with LabVIEW.

This manual covers both sets of ISA LabVIEW drivers, spectrometer control and CCD SpectrumONE drivers. The CCD drivers also included a disk for the spectrometer drivers. However if only the spectrometer drivers were ordered, the CCD drivers will not be present.

LabVIEW Driver for ISA Spectrometer Controllers

The ISA LabVIEW drivers offer three approaches to the spectrometer controllers.

- A set of high level interface VI's provide the building blocks needed to control spectrometers, accessories, and acquire spectroscopic data.

- Single spectrometer scans, Synchronous dual spectrometer scans, and Time Base scans can be set up to run autonomously.

- For those who prefer low level command control, a communications VI (ISA DRIVER.VI) is provided that simply passes commands to the spectrometer control electronics. This allows the user-programmer to optimize specific applications.

The user level Virtual Instruments and Globals provided with the ISA LabVIEW Spectrometer driver are contained in the library file ISA_USER.LLB. These are described in the following pages. Please note that ISA_COMM.LLB, a library of low level VI's that are required to run the user VI's is also included with the package. These are not user level VI's, and no modification to these VI's is allowed. They are not documented here.

Installing the ISA Libraries:

To add the ISA VI's to a computer that has LabVIEW already installed, simply copy the EXE file ISA_TRX.EXE from the ISA LabVIEW Driver disk to a suitable location on the computer's hard drive. Copying the file to a new subdirectory called "ISA" is recommended. Then run the execute the file to extract the libraries.

Once the files are extract in the directory, they should be compiled. From the LabVIEW menu, pull down File / Mass Compile. To save time, it is recommended to compile ISA_COMM.LLB first, then ISA_USER.LLB.

To start using the ISA LabVIEW VI's, the first thing to be done is to call the SPECTROMETER SETUP global and enter and save the correct parameters for the spectrometer that will be controlled.

To Save:

- Under the Operate select and enter Edit Mode then change values.
- Again under Operate select Make Current Values Defualt
- Save

To start each session, run the Start Up. VI to establish communications. If the spectrometer is manually calibrated, then run the Calibrate.VI (not required for Autocalibrating Spectrometers).

Following these initial steps, further control actions depend on the application's requirements.

Virtual Instrument Libraries:

The ISA LabVIEW Driver package includes two Virtual instrument libraries. The User interface VI's are in ISA_USR.LLB.The other library, ISA_COMM. LLB, contains communications related VI's that the user does not need to (and probably shouldn't) access directly.

ISA_USR.LLB

Spectrometer Setup.GBL	Use this array to record the system operating parameters, and set the communications port or GPIB address to be used to access the spectrometer controller. This is used by the high level VI's.
Start up.VI	Performs communications initialization, and for autocalibrating spectrometers, also calibrates the wavelength drive and initializes the slits and mirrors.
Spectral Position.VI	Use this to display the present spectral position during a scan.
Spectral GOTO.VI	Provides a panel to enter the target spectral position.
Port & Grating.VI	Use this to change turret and mirror positions of automated spectrometers.
Shutter.VI	Use this to control a shutter.
Slits.VI	Use this to set slit widths of automated spectrometers.
Acquisition.VI	Use this to set the acquisition parameters.
HiVolt.VI	Use this to set the high voltage power supplies.
ISA DRIVER.VI	For those who wish to create specialized control programs, this low level command communications link passes individual driver calls to the spectrometer controller. It also establishes the communications port or GPIB address to be used to access the spectrometer controller.

SPECTROMETER SETUP.GBL

The SETUP global is used to capture the configuration data needed by the various control VI's. To find the correct values for these parameters, refer to the Monochromator Setup Parameters Table in the Appendix of the Spectrometer Control Manual. Parameters for both Spex and Jobin-Yvon spectrometers are listed.

- The **Spectrometer Number** is used to index each spectrometer in the system. This allows configurations to be set up for each of the spectrometers.

- The **Calibration** button toggles between Auto and Manual. Use **AUTO** for the 270M or HR460 autocalibrating spectrometers, otherwise set the button to **MANUAL.**

- The **Groove Density** and **Blaze Wavelength** describe the installed grating(s) in the selected spectrometer. If a turret is installed, enter the information about the second grating by selecting with the index control on the left of the indicator.

- The **Drive Type** is used to indicate which motor driver is needed to move control the instrument. Select *Wavelength* if using a TRIAX180, for all other instruments select *Step.*

The **Grating Motor Setup** section includes Indicators for:

- The **Base Grating** groove density. This is used as a reference value for determining the ratio of steps per spectral unit for any grating in use. Usually the base is 1200 grooves per millimeter.

- The **Steps / Spectral Unit** factor is used by other VI's to convert positions entered in spectral units to motor steps required to control the hardware.

- Enter the **Spectral Unit** of the scanning drive mechanism (i.e. nm or cm-1).

- Enter the **Min** and **Max Limits** of the drive in terms of spectral units for the base grating.

- Enter the **Backlash** correction required to assure positioning accuracy.

The **Slit Motor Setup** section applies only to spectrometers with motorized slits.

- The **Slit Installation** cluster contains buttons to install or remove each of the possible automated slits from the configuration.

- The **Steps / Unit, Backlash,** and **Min** and **Max Limits** for the slit motors are set in the same way as the Monochromator Setup Parameters mentioned above.

The **Communications Parameters** cluster is used to select the hardware connections to access the Spectrometer control electronics.

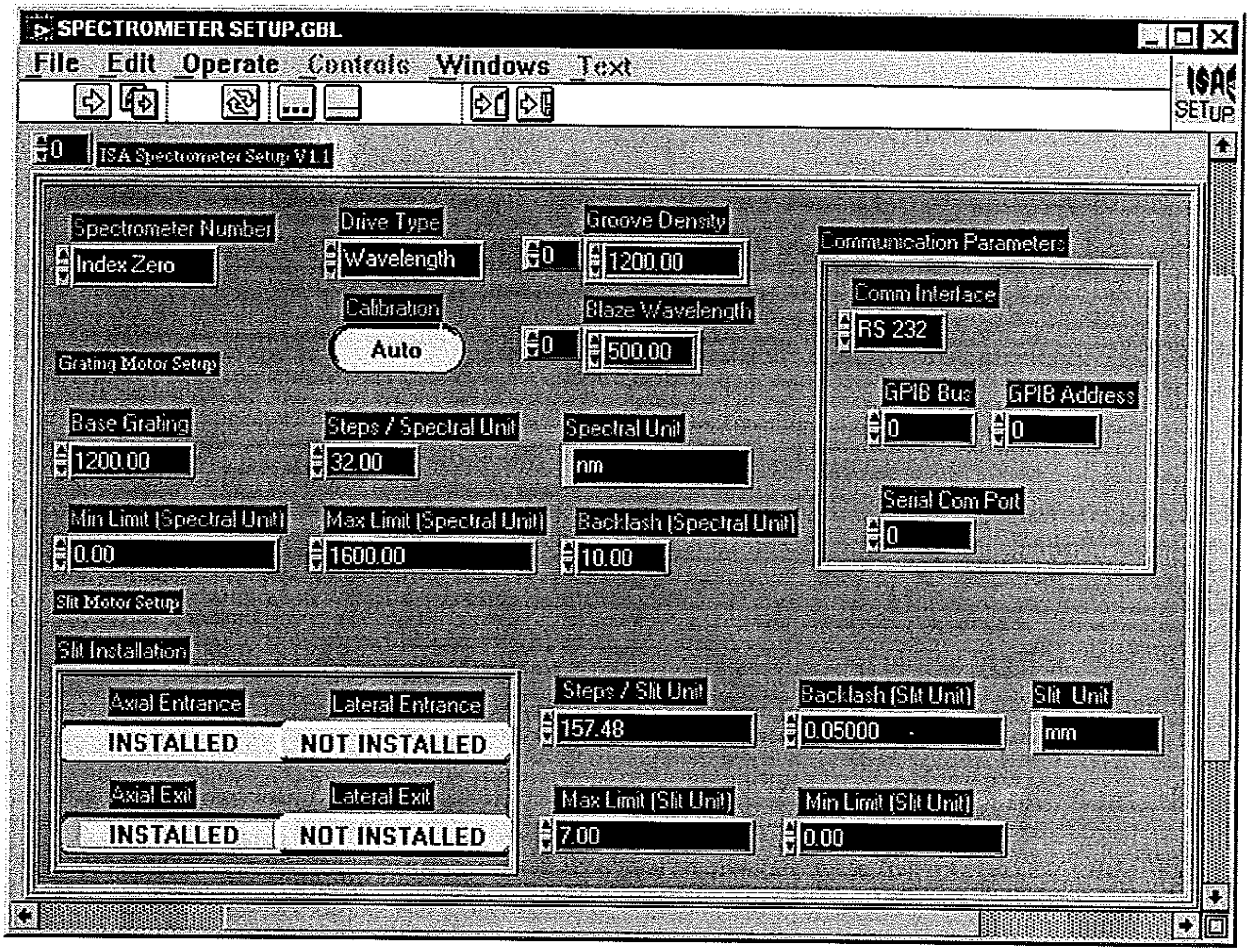
- The **Interface** ring control selects the RS-232 serial or IEEE488 GPIB communications hardware.

- If GPIB is selected as the interface, the **GPIB Bus** numeric control determines which of the GPIB interface cards plugged into the computer is used.

- The **GPIB Address** numeric control sets the address of the spectrometer controller on the Bus.

- If the RS-232 Interface is selected, the **Serial Com Port** control is used to select the serial port connection on the computer.

  COM1 = PORT Ø
  COM2 = PORT 1



0 ISA Spectrometer Setup V1.1

Spectrometer Number: Index Zero
Drive Type: Wavelength
Groove Density: 0 1200.00
Calibration: Auto
Blaze Wavelength: 0 500.00

Grating Motor Setup

Base Grating: 1200.00
Steps / Spectral Unit: 32.00
Spectral Unit: nm
Min Limit (Spectral Unit): 0.00
Max Limit (Spectral Unit): 1600.00
Backlash (Spectral Unit): 10.00

Communication Parameters
Comm Interface: RS 232
GPIB Bus: 0
GPIB Address: 0
Serial Com Port: 0

Slit Motor Setup

Slit Installation
Axial Entrance: INSTALLED
Lateral Entrance: NOT INSTALLED
Axial Exit: INSTALLED
Lateral Exit: NOT INSTALLED

Steps / Slit Unit: 157.48
Backlash (Slit Unit): 0.05000
Slit Unit: mm
Max Limit (Slit Unit): 7.00
Min Limit (Slit Unit): 0.00

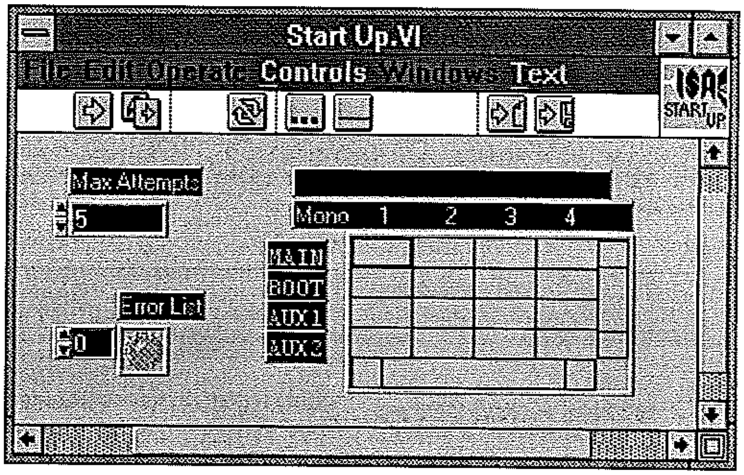
Start Up.VI

The Start Up.VI must be executed each time the computer or spectrometer controller is powered on. This VI establishes communications with the ISA spectrometer controller and sets it to the proper mode to accept the commands that will follow from the other VI's.

- The **Max Attempts** control sets the number of times the Start Up VI will try to establish communications with the Spectrometer Controller before returning an error.

- The **Error** LED is a binary indicator that will turn from green to red when the Start Up VI is unsuccessful in establishing communications. If failing to communicate, refer to the Computer and LabVIEW manuals for help with connections and settings.

- **The Controller Program Versions** Array returns the version numbers of the internal programs that reside in the spectrometer controller. This information may be helpful when contacting ISA Customer Service for assistance.



Max Attempts: 5

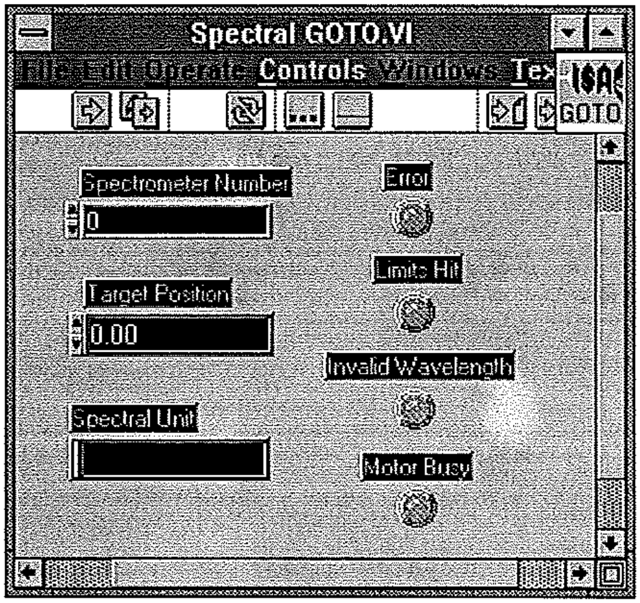
Error List: 0

Mono	1	2	3	4
MAIN				
BOOT				
AUX1				
AUX2				

Spectral GOTO.VI

The Spectral GOTO.VI is used in nearly all applications. Its function moves the grating drive to a specified position in terms of the spectral units of the spectrometer (i.e. nanometers or wavenumbers.) The VI will convert this to motor steps based on the factor entered in the Spectrometer Setup.GBL panel. Note that if the preferred working spectral unit is different from the base spectral unit of the spectrometer's grating drive mechanism, this may be accomodated by including further conversion calculations in the application VI. For example: if a wavelength (nm) drive spectrometer is to be positioned in terms of Relative Wavenumbers, the application VI will convert positions from Rcm-1 to nm before passing the value to the Spectral GOTO.VI.

- The **Spectrometer Number** selects the spectrometer to be controlled.
- The **Target Position** is where the drive will move to.
- The **Spectral Unit** indicator returns the working unit according to the Spectrometer Setup.
- The LED **Error** indicator turns red if there is an error preventing the VI from repositioning the spectrometer drive as requested.
- The **Limits Hit** indicator will turn red if the Spectrometer grating drive mechanism travels to an extreme position, sensed by a limit switch.
- The **Invalid Wavelength** indicator will turn red if the Target Position entered is outside the bounds defined by Min and Max Limits entered in the Spectrometer Setup global.
- The **Motor Busy** indicator turns red if the spectrometer's motor drive is in the process of repositioning, and cannot presently accept a new command to move.



Spectrometer Number: 0

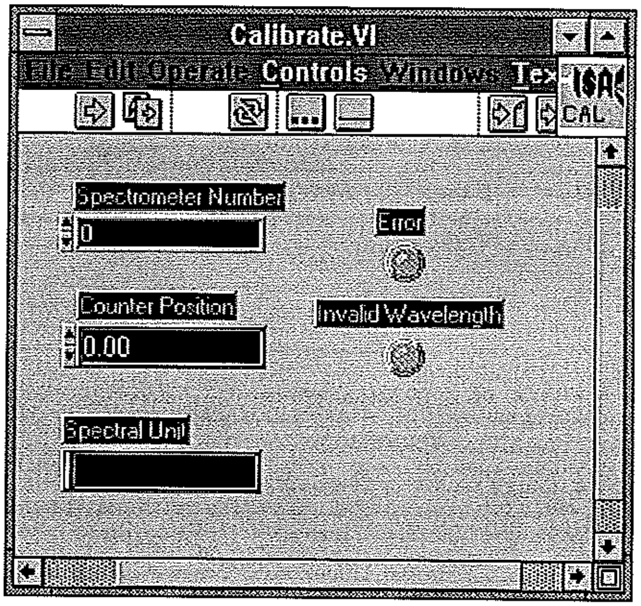
Target Position: 0.00

Spectral Unit:

Calibrate.VI

The Calibrate.VI is used for manual calibrating spectrometers to enter the position of the spectrometer grating drive when the system is powered on. To properly calibrate a spectrometer, the Spectrometer Setup Global must contain the correct base grating, spectral unit, and steps / unit parameters.

- The **Spectrometer Number** selects the spectrometer to be controlled.
- The **Counter Position** is the reading from the spectrometer's mechanical counter.
- The **Spectral Unit** indicator returns the working unit according to the Spectrometer Setup.
- The LED **Error** indicator turns red if there is a communications or out of range error .
- The **Invalid Wavelength** indicator will turn red if the Target Position entered is outside the bounds defined by Min and Max Limits entered in the Spectrometer Setup global.



Spectrometer Number 0

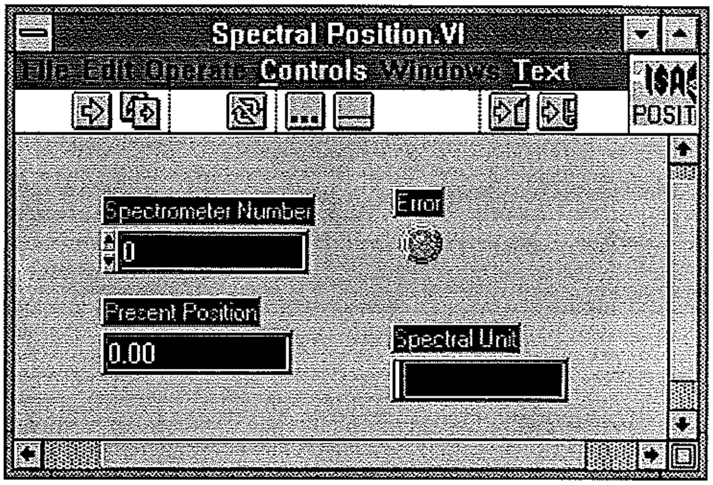
Counter Position 0.00

Spectral Unit

Spectral Position.VI

The Spectral Position.VI simply returns the present position of the spectrometer's grating drive. This is intended to be used as an easily updated digital indicator showing:

- The **Spectrometer Number** of the selected spectrometer.
- The **Present Position** for the selected spectrometer.
- The **Spectral Unit** indicator returns the working unit (i.e. Nanometers or Wavenumbers).
- A LED **Error** indicator will turn red if there is a communications failure.



Spectrometer Number: 0

Error

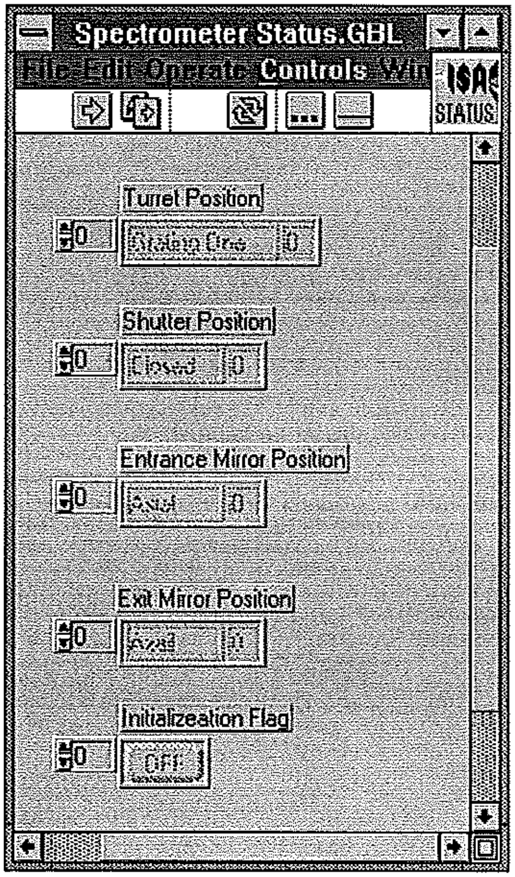
Present Position: 0.00

Spectral Unit

Spectrometer Status.GBL

The Spectrometer Status.GBL is used to determine the present positions of automated spectrometer accessories. For each indicator there is an index at the right to select the spectrometer number for status reporting.

- **Turret Position** indicates which grating is in use.
- **Shutter Position** reports that the shutter is open (1) or closed (0)
- **Entrance Mirror Position** reports the entrance port selection mirror setting: lateral (1) or axial (0)
- **Exit Mirror Position** reports the exit port selection mirror setting: lateral (1) or axial (0)
- **Initialization Flag** is a binary indicator that shows whether the spectrometer has been initialized (calibrated) or not.



Turret Position
0 Grating One 0

Shutter Position
0 Closed 0

Entrance Mirror Position
0 Axial 0

Exit Mirror Position
0 Axial 0

Initializeation Flag
0 OFF

Port and Grating.VI

The Port & Grating. VI allows the user to control the port selection mirrors and grating turret in an automated spectrometer that has those options installed. The front panel settings are as follows:

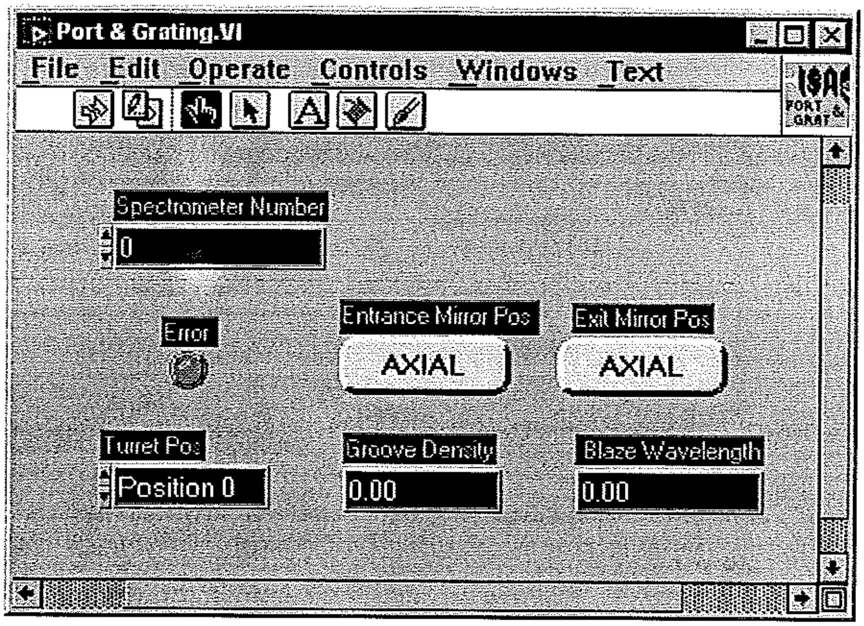
- **Spectrometer Number:** for multiple spectrometer systems, this selects the particular spectrometer to allow its mirrors and/ or turret to be positioned.

- **The Entrance Mirror Position** and **Exit Mirror Position** buttons toggle between the possible positions for the mirrors.

- **Turret Position:** Use to select the grating.

- **Groove Density** and **Blaze Wavelength** indicators for the selected grating will be updated when the VI is executed.

Note that when a port mirror moves, the shutter (if any) associated with the slit selected becomes active. The shutter (if any) associated with the deselected port becomes inactive, and defaults to closed.



Spectrometer Number: 0

Entrance Mirror Pos: AXIAL

Exit Mirror Pos: AXIAL

Turret Pos: Position 0

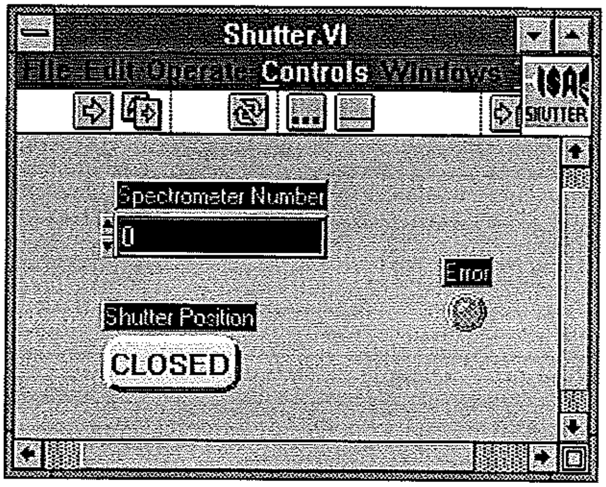
Groove Density: 0.00

Blaze Wavelength: 0.00

Shutter.VI

Shutter.VI provides for control of interfaced shutters in the system.

- The **Spectrometer Number** selects the spectrometer to be accessed for shutter control.
- The **Shutter Position** button toggles the active shutter open or closed.



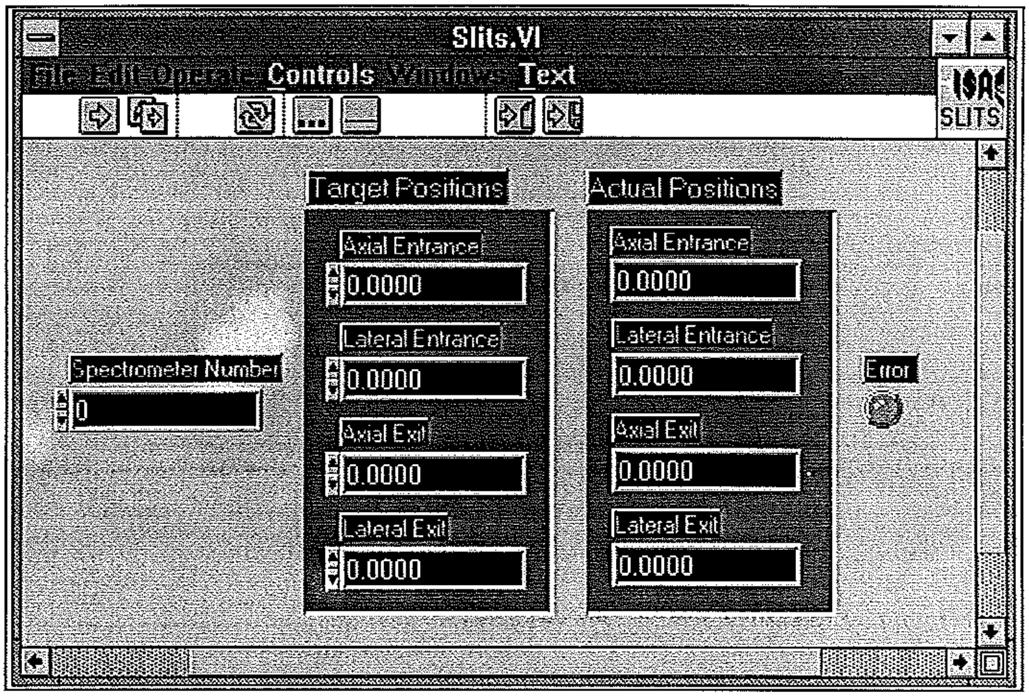
Spectrometer Number: 0

Shutter Position: CLOSED

Slits.VI

For spectrometers with automated Slits, the Slits.VI provides a means of setting the slit widths as required. The slit width is expressed in the slit base unit, usually mm.

- The **Spectrometer Number** selects the spectrometer to be accessed for shutter control.
- The **Target Positions**: Enter the value in the slit's base unit (generally mm) that is preferred. note that to calculate the resulting bandpass, one must multiply the slit width by the dispersion for the spectrometer. If not using the base groove density, apply the appropriate factor to take this into account also.
- The **Actual Positions**: These indicators will return the actual slit width to the nearest allowable step increment.



Target Positions

Axial Entrance 0.0000

Lateral Entrance 0.0000

Axial Exit 0.0000

Lateral Exit 0.0000

Actual Positions

Axial Entrance 0.0000

Lateral Entrance 0.0000

Axial Exit 0.0000

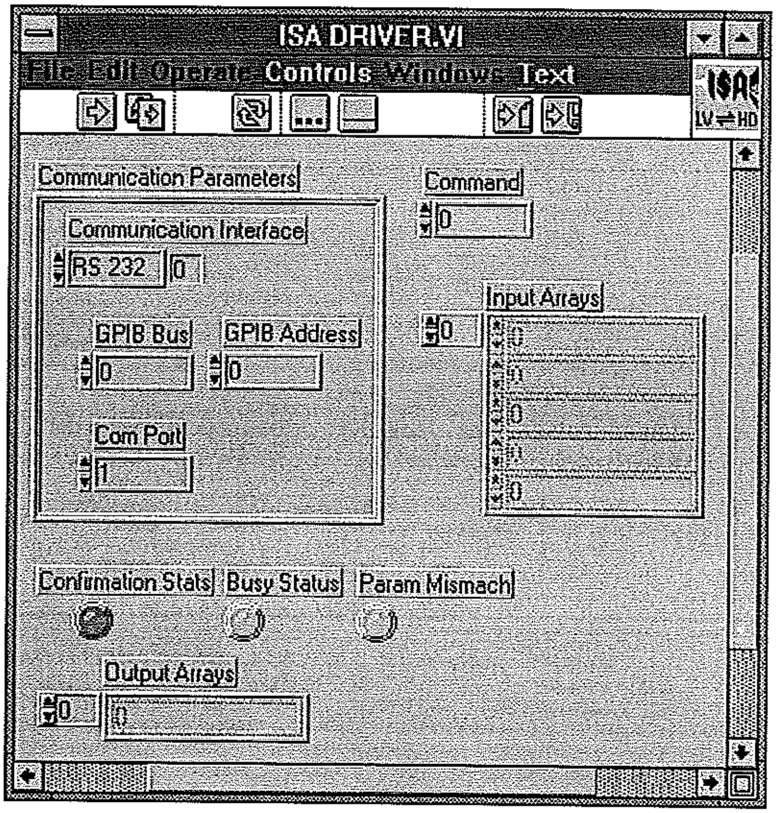
Lateral Exit 0.0000

Spectrometer Number 0

ISA DRIVER.VI

The ISA DRIVER.VI provides a low level communications interface to the spectrometer controller. The **Communications Parameters** cluster is used to define the hardware connection to the spectrometer controller.
The numeric control labeled "**Command**" receives the call number associated with the command to be sent. A call number is assigned to each command. This scheme allows the command to be indexed from an array that assists in formatting the command strings and parsing the returned data. The Driver Calls table on the next page lists the call numbers to be used with the ISA DRIVER.VI, and cross- references the calls to the original command character codes. For detailed description of each command and I/O Array values, refer to the Spectrometer Control Manual, Command Descriptions section.

- The **Input Array** receives the parameters associated with the command.
- The **Output Array** returns the response data from the spectrometer controller.
- The LED **Confirmation Status** indicator reports on the receipt of the communications confirmation from the Spectrometer controller.
- The **Busy Status** LED indicator advises when a device is busy by turning red.
- The **Parameter Mismatch** LED indicates whether the correct number of input parameters have been issued.



Communication Parameters

Communication Interface: RS 232 0

GPIB Bus: 0

GPIB Address: 0

Com Port: 1

Command: 0

Input Arrays: 0 — 0, 0, 0, 0, 0

Output Arrays: 0 — 0

Driver Calls for the ISA DRIVER.VI

Call	Command Name	Code	Input Array Size
0	Motor Init	A	0
1	Motor Set Speed	B	4
2	Motor Read Speed	C	1
3	Motor Busy Check	E	0
4	Motor Move Relative	F	2
5	Motor Set Position	G	2
6	Motor Read Position	H	1
7	Motor Limit Status	K	0
8	Motor Stop	L	0
9	Slit Set Speed	g	3
10	Slit Read Speed	h	2
11	Slit Set Position	i	3
12	Slit Read Position	j	2
13	Slit Move Relative	k	3
14	ACC Shutter Open	W	1
15	ACC Shutter Close	X	1
16	ACC Turret Position 1	a	1
17	ACC Turret Position 0	b	1
18	ACC ENTR Mirror Side	c	1
19	ACC ENTR Mirror Front	d	1
20	ACC Exit Mirror Side	e	1
21	ACC Exit Mirror Front	f	1
22	ACC Busy Check	l	0
23	ACQ Measure Offsets	w	1
24	ACQ Set Offsets	x	5
25	ACQ Channel Gain Set	R	2
26	ACQ Gain Read	S	1
27	ACQ Integration Time Set	O	2
28	ACQ Integration Time Read	P	1
29	ACQ Start	M	1
30	ACQ Stop	N	0
31	ACQ Busy	Q	0
32	ACQ Read Data	T	1
33	High Voltage Set	U	2
34	High Voltage Read	V	1
35	TTL Write Output	m	1
36	TTL Read Output	n	0
37	TTL Read Input	o	0
38	TTL Automatic Outputs	Z11	8
39	Threshold Write Value	I	4
40	Threshold Read Values	J	0
41	Scan Set Parameters	p	19
42	Scan Start	q	0
43	Scan Stop	v	0
44	Scan Current Status	r	0
45	Scan Cycle to Read	s	1
46	Scan Get Data Point No.	t	0
47	Scan Get Data	u	1

LabVIEW Drivers for ISA CCD systems:

Instruments SA has expanded its LabVIEW® library to include Virtual Instrument libraries for control of the Spectrum-One CCD line. The CCD package includes Virtual Instruments (VI's) to acquire spectra, define binning, and even acquire images. Also include are the drivers for control of SPEX and Jobin-Yvon spectrometers. With these drivers, all aspects of our spectroscopy systems can be controlled and run from the LabVIEW® environment.

System Requirements:

- An IBM compatible running National Instruments' LabVIEW® version 3.0.1*
- The computer must be running a ISA Spectrum-ONE CCD system with a CCD-2000 and RISC card or CCD-3000 controller with a National Instruments IEEE488 card.
- To use the Instrument SA (SPEX or Jobin-Yvon) spectrometer control LabVIEW® drivers, the spectrometer must be equipped with a SPEX232, SPEX488, JY232, or JY488 spectrometer interface or the DataLink, DataScan, or SpectrAcq spectrometer controller.

*If using LabVIEW 4.0, it must be installed as the 16-bit version (Windows 3.11). The DLL files for the CCD controller will only interface with the 16-bit version.

Installation:

The LabVIEW drivers for CCD control are virtually self-installing. The process takes between 5 and 10 minutes, depending on the speed of your computer.

The LabVIEW drivers for CCD control are delivered on a single 3½" disks which was prepared especially for your configuration. Prior to beginning the installation procedure, make sure you know (1) where the CCD controller hardware is installed (specifically, settings for the CCD control board) and (2) where you want the program installed (the default drive\directory is **C:\LABVIEW\ISACCD\.**

If you have a CCD 2000 controller, you will need to know the PORT, IRQ and DMA address of the CCD board. The default for new boards is PORT 3616, IRQ 5, and DMA 1. For older CCD boards with onboard memory (the main board will have a smaller board attached to it that contains the memory) you should enter the following settings: PORT 768, IRQ NONE, DMA NONE. If you have a CCD detector and are using the CCD 3000 the IEEE-488 device number is 5. If you have any question regarding your board or controller type and/or installation, please contact the service department.

To install The LabVIEW drivers for CCD control under Windows 3.x,

- **Turn on the computer and enter Windoes.**
- **Place the Disk in the appropriate floppy disk drive (a: or b:).**
- **Select File, Run**
- **Type "a:\setup" (or "b:", depending on where the disk is inserted) and press ENTER.**

To install LabVIEW Drivers under Windows™ 95,

- **Click on the Start key.**
- **Select *Run.***
- **Type "a:\setup" and press OK (where a: is the drive letter where the installation disk is located.**

The computer will access the floppy drive, begin to read the files and write information to the screen. You will see an *"Installing ISA"* message at the beginning of the installation procedure.

A dialog box suggesting the **C:\ LABVIEW\ISACCD\** directory will pop up.

Either,

- **Click *OK* to accept the default.**

Or

If you would prefer to have the program installed on another drive and/or in a different directory,

- **(Optional) Replace the C:\ LABVIEW\ISACCD\ suggestion with the preferred drive and/or location.**

Or

- **Click on the *Browse* button to view the contents of the hard drive and choose another directory**
- **Click *OK* to proceed.**

Notice that on the right lower portion of the screen are two icons: one for *Pause* and the other for *Exit*. Clicking on *Pause* will interrupt the installation without terminating the procedure, and clicking on *Exit* will terminate the installation and exit the installation program.

The *CCD Controller* dialog box will appear,

- **If using the CCD 2000 controller. Enter the information regarding the PORT, IRQ and DMA for the CCD card then click *OK* to confirm. (See information above).**

Or

- **If using the CCD 3000 controller. Enter the information regarding the IEEE-488 device number then click *OK* to confirm. (The unit is hardcoded as dev 5).**

The program will complete the installation and an option to access the *ReadMe* file will be presented to the screen. Any last minute changes to the software or other valuable information is contained in this file.

Virtual Instrument Libraries:

The ISA LabVIEW Driver package includes one Virtual instrument library. Some example interface VI's are in default directory. The library, ISA_CCDVI. LLB, contains the individual VI's that control different portions of the CCD operation.

ISACCDVI.LLB

CCDREAD.VI	This VI retrieves a 2D array of CCD data from the specified region
CCDBUSY.VI	CCD Busy, Check if a CCD acquisition is in progress.
CCDError.VI	Use this VI to decode the numeric error codes given.
CCDGO.VI	Starts the CCD acquisition with the given region(s) defined in RGNSET.VI.
CCDSTOP.VI	Use this VI to halt a CCD acquisition in progress.
OpenCCD.VI	Load and open the CCD driver DLL with the given parameters.
PROPGET.VI	This VI returns requested information relevant to the operation of the CCD. The available properties are listed in README.VI
PROPSET.VI	
ReadMe.VI	VI containing text file with CCD Properites (See Below)
RGNSET.VI	Use this VI to define all of the active acquisition regions within the CCD chip.

From ISACCD Directory

TestDEV.VI	This VI demonstrates acquisition and image display of CCD data utilizing the ISA CCD driver. Note: Specific help is available for each individual control, display and indicator.
TempMON.VI	Monitors the temperature.

Operation:

If using the CCD2000 controller, before calling any VI, run the HWINIT.BAT program to initialize the CCD2000 controller. The CCD3000 controller does not require any hardware initilization.

Supported IEEE 488 Computer Interface Boards (CCD3000 Only):

The IEEE 488 example programs work with several of the National Instruments PC interface boards.

- GPIB-PCII/PCIIA 488.2 Interface board: The driver supplied by National should be version 1.2 or newer, and their BASIC support disk should be version 2.0 or newer.
- Older GPIB-PCIIA boards require National's revision C13 or newer software.
- AT-GPIB 488 boards require National's revision E7 or newer software.
- AT-GPIB 488.2 board: Must be version 2.7.0 software, and their BASIC support disk version 2.2 or newer.
- GPIB-PCIII board: this model must be replaced with one of the above. Contact your local National Instruments dealer with regard to their current replacement program.

There are other boards by National and other suppliers for IBM compatible and MacIntosh computers. Many of these boards can function in *SIMILAR* fashion, however the drivers communicate specifically with the National Instruments drivers therefore we **do not** support or guarantee reliable communications with other boards and software, you **must** use the National Instruments products as described above.

Establishing GPIB Communications (CCD3000 Only):

The IEEE board and it's associated software driver must be installed in your computer as per National Instruments' instructions. The driver must be installed via your computer's CONFIG.SYS file.

This set up assumes that the National Instruments software driver has the PCII board name as GPIB0 and the first device name as DEV1. Run the IBCONF program and set the configuration as follows:

GPIB0	
Primary GPIB Address	0
Secondary GPIB Address	None
Timeout Setting	T3s
EOS byte	00H
Terminate read on EOS	no
Set EOI with EOS on write	no
Type of compare on EOS	7-bit
Set EOI w/ last byte of write	yes
System controller	yes
Assert REN when SC	no
Enable auto serial polling	yes
Timing	500 ns
Enable 488.2 protocols	yes
CIC protocols	no
Handler Type	PC2
Interrupt Setting	none
Base I/O address	02B8H
DMA channel	1

DEV5	
Primary GPIB address	5
Secondary GPIB address	None
Timeout Setting	T10s
EOS byte	0Dh
Terminate read on EOS	yes
Set EOI with EOS on write	no
Type of compare on EOS	7-bit
Set EOI w/ last byte of write	yes
Repeat addressing	no

(The above are the default settings)

General Notes:

Changes made to SYSTEM.INI

Here is an example section from a valid system.ini file:

```
[ISA CCD Driver]
debugflags=0
path=c:\labview\isaccd\
port=3616
dma=1
irq=5
```

Emulation

Sound Cards and CD-ROM drives:

Many sound card and CD-ROM controllers use IRQ 5 as the default setting. This causes a conflict with the CCD data acquisition board in the CCD2000. Change the IRQ address of the sound card or consult the SpectrumONE manual on how to change the address on the CCD card. If you change the address on the CCD card you must change the setting in the SYSTEM.INI or reinstall the software with the new IRQ.

Error Codes:

DLL Handle = 0
    DLL can't be found

DLL Handle ≠ 0 AND API Handle = 0
    Incorrect SYSTEM.INI setup
    Hardware conflict

Cannot find GSRGNIF.LSB

This file should be located in the same directory as the ISACCDVI.LLB file.

CCDBUSY.VI

Checks to see if a CCD acquisition is in progress.

Input:

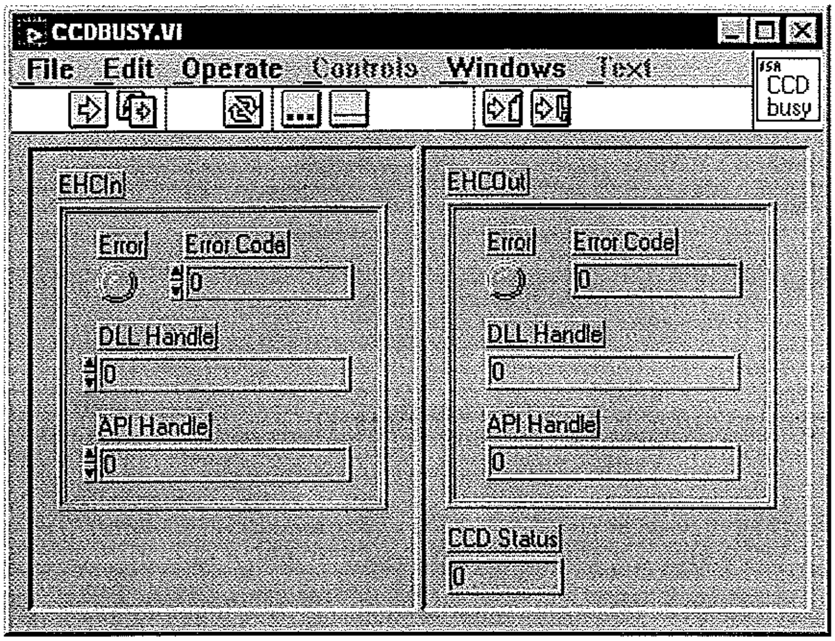
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.
- CCD status
  - Return 0 if no acquisition is in progress (idle).
  - Return non-zero if acquisition is in progress (busy).

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



CCDError.VI

Use this VI to decode the numeric error codes given

Input:

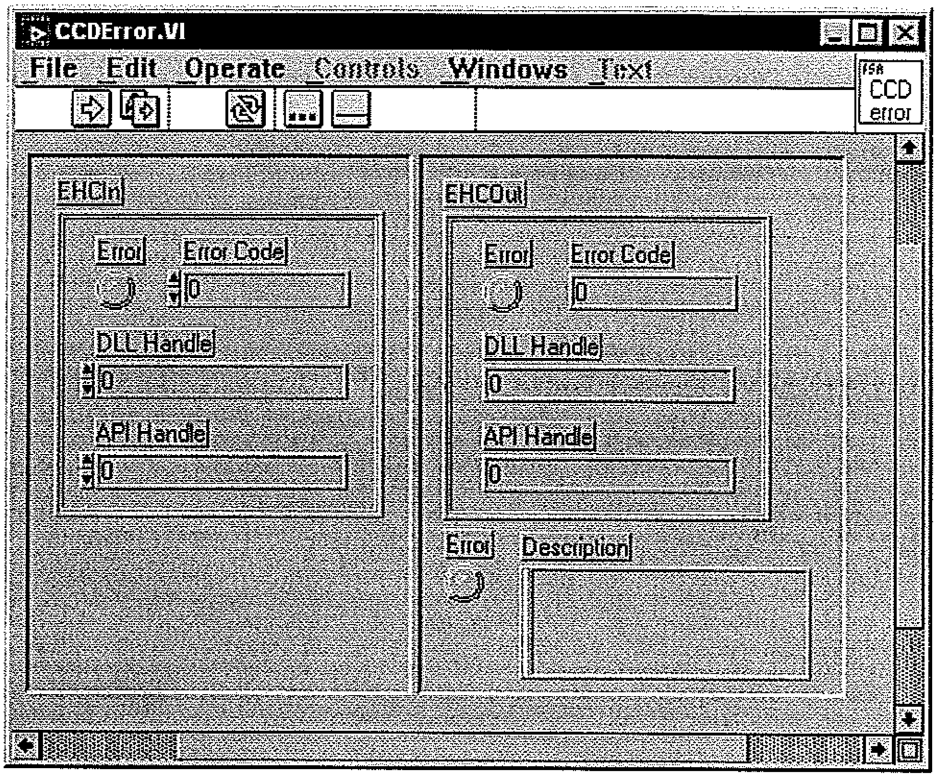
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.
- Error
  - Error LED indicates if an error has occurred.
- Description
  - Plane text error message.

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



EHCIn
Error
Error Code 0
DLL Handle 0
API Handle 0

EHCOut
Error
Error Code 0
DLL Handle 0
API Handle 0

Error
Description

CCDGo.VI

Starts the CCD acquisition with the given region(s) defined in RGNSET.VI

Input:

- EHCln - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.
- Blank
  - TRUE if the data is to be acquired with the shutter closed.
  - FALSE if the data is to be acquired with the shutter open.

Output:

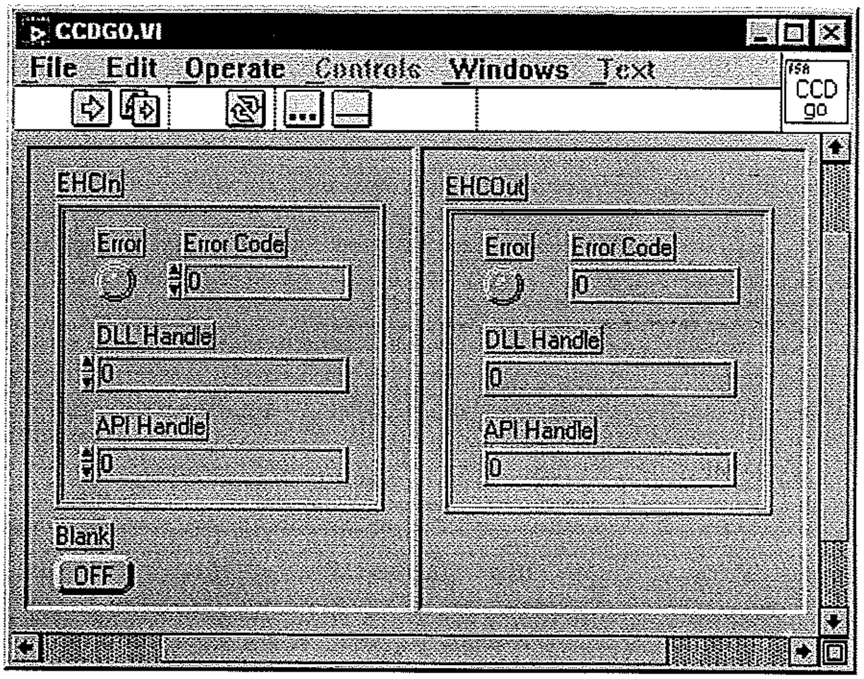
- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.

Notes:

- Do not run this VI during an acquisition. Wait until busy is done.



EHCIn
Error Error Code
0
DLL Handle
0
API Handle
0
Blank
OFF

EHCOut
Error Error Code
0
DLL Handle
0
API Handle
0

CCDRead.VI

This VI retrieves a 2D array of CCD data from the specified region

Input:

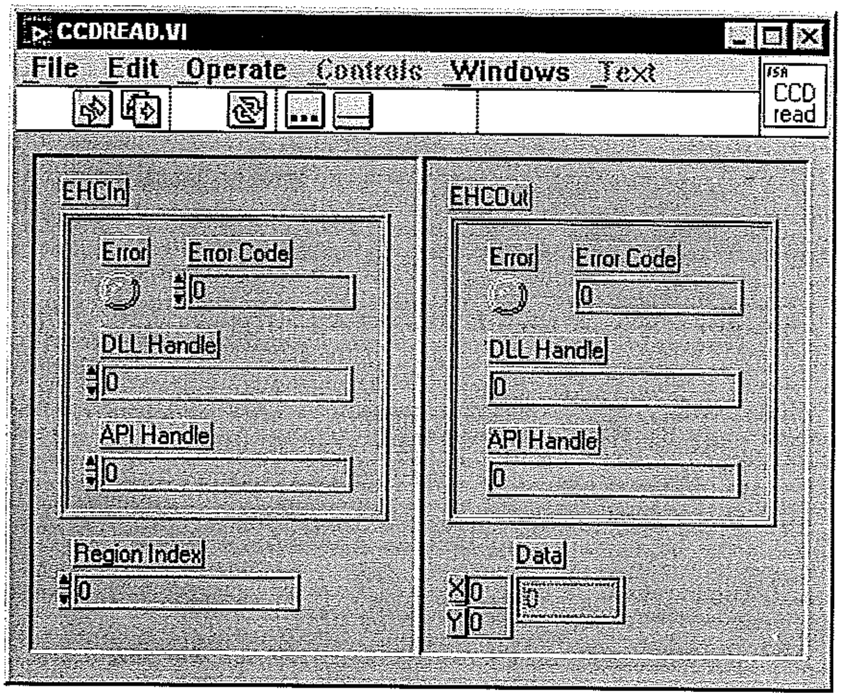
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.
- Region Index
  - Specifies the region to be retrieved.

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.
- Data
  - The DATA array contains the signal intensity read from all of the pixels or superpixels in the specified region. The display box shows the DATA acquired from the pixel or superpixel at the given X-Y data point coordinates within the selected region.

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



EHCIn
Error
Error Code 0
DLL Handle 0
API Handle 0
Region Index 0

EHCOut
Error
Error Code 0
DLL Handle 0
API Handle 0
Data
X 0
Y 0
0

CCDStop.VI

Use this VI to halt a CCD acquisition. It is important to call this VI for aborting an acquisition. (It resets the internal CCD state machine.)

Input:

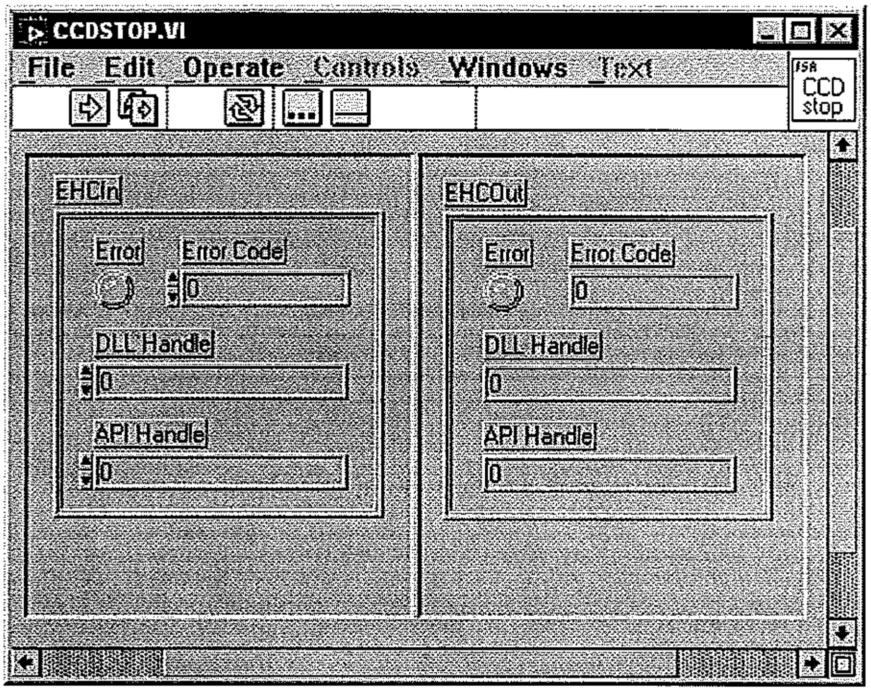
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



EHCIn

Error | Error Code: 0

DLL Handle: 0

API Handle: 0

EHCOut

Error | Error Code: 0

DLL Handle: 0

API Handle: 0

OpenCCD.VI

Load and open the CCD driver DLL with the given parameters. This VI must be called first to establish the links to the hardware.

Input:

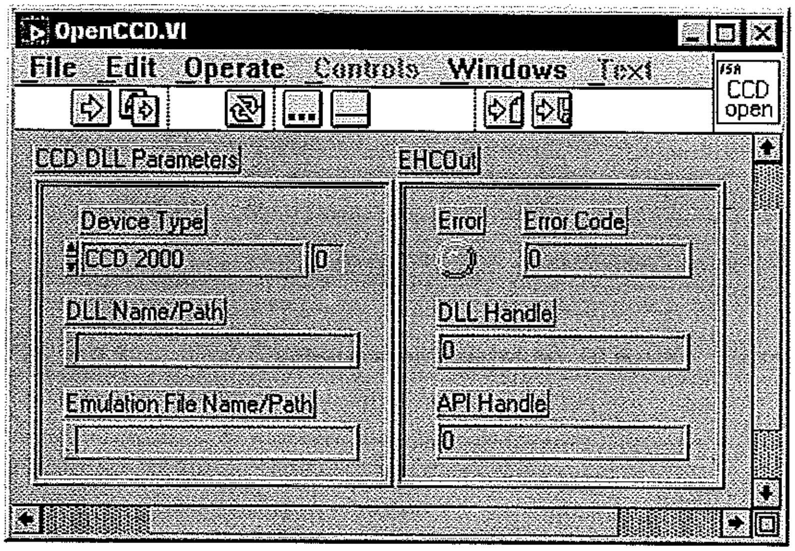
- CCD DLL Parameters
    - Specifies the initial settings required to load the driver.

Output:

- EHCOut - Error/Handles Cluster Output
    - This cluster reports errors and contains handles used by the driver.

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.
- Device Type - Specify the actual controller or an emulator to be used.
- DLL Name/Path - Specify the name and location of the CCD driver DLL.
- Emulation File Name/Path - Specify the name and location of the CCD emulation file. Note: This entry is needed only for emulation mode.



CCD DLL Parameters

Device Type: CCD 2000 | 0

DLL Name/Path:

Emulation File Name/Path:

EHCOut

Error | Error Code: 0

DLL Handle: 0

API Handle: 0

PropGET.VI

This VI returns requested information relevant to the operation of the CCD. The available properties are listed in the README.VI

Input:

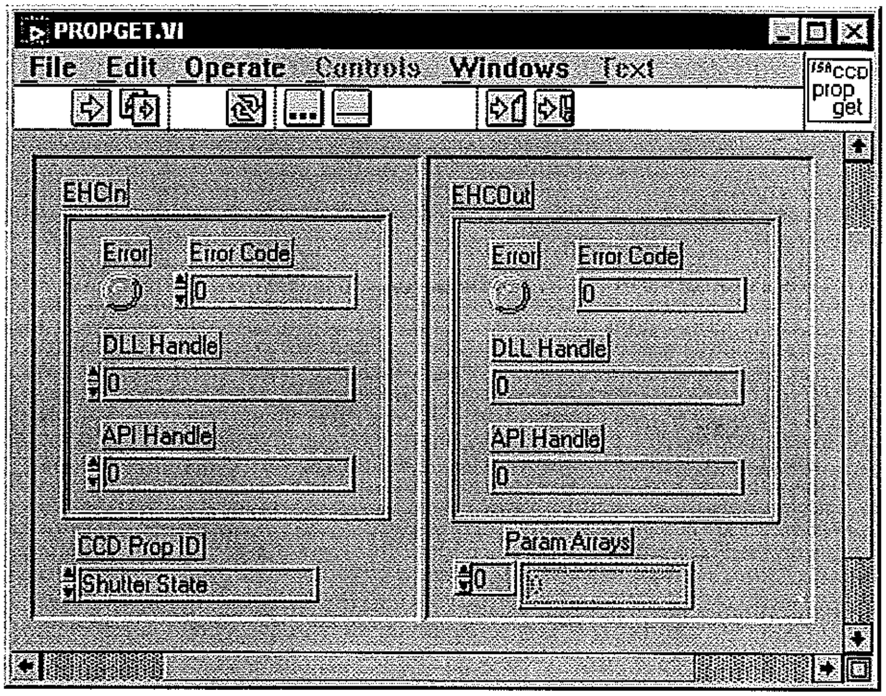
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.
- CCD Prop ID
  - This entry specifies the CCD property to be set. Refer to README.VI for a description of each property and the return parameter(s).

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.
- Param Arrays
  - This 1D array contains the current value(s) of the chosen CCD property. Refer to README.VI for a description of each property and the return parameter(s).

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



EHCIn

Error Error Code 0

DLL Handle 0

API Handle 0

CCD Prop ID Shutter State

EHCOut

Error Error Code 0

DLL Handle 0

API Handle 0

Param Arrays 0

PropSet.VI

Input:

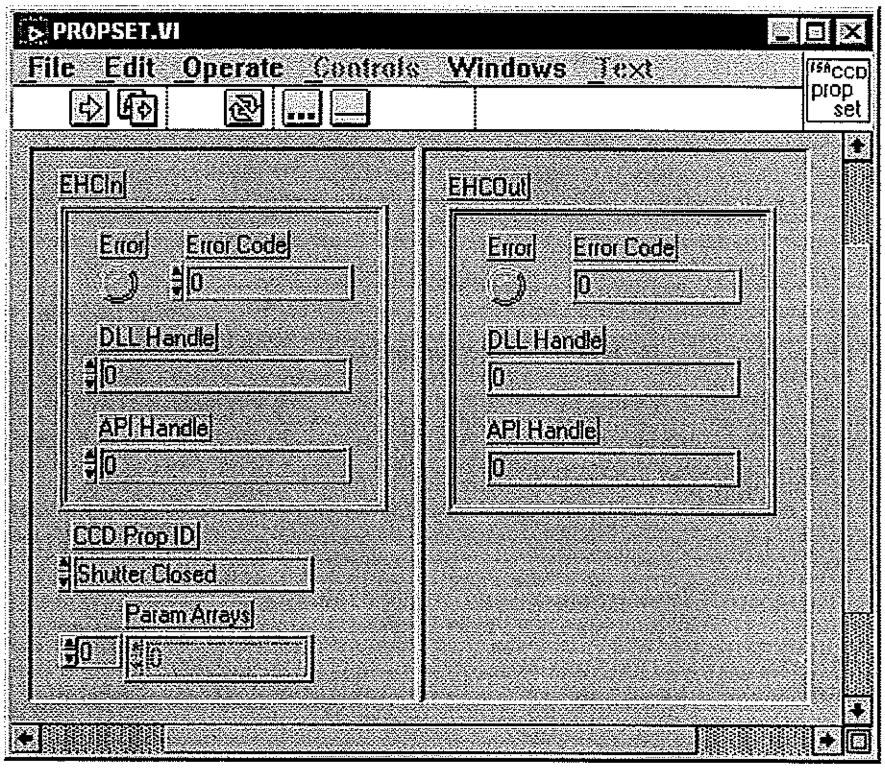
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.
- CCD Prop ID
  - This entry specifies the CCD property to be set. Refer to README.VI for a description of each property and the return parameter(s).
- Param Arrays
  - This 1D array contains the target value(s) of the chosen CCD property. Refer to README.VI for a description of each property and the return parameter(s).

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



EHCIn

Error

Error Code 0

DLL Handle 0

API Handle 0

CCD Prop ID Shutter Closed

Param Arrays 0 0

EHCOut

Error

Error Code 0

DLL Handle 0

API Handle 0

RGNSet.VI

Use this VI to define all of the active acquisition regions within the CCD chip

Input:

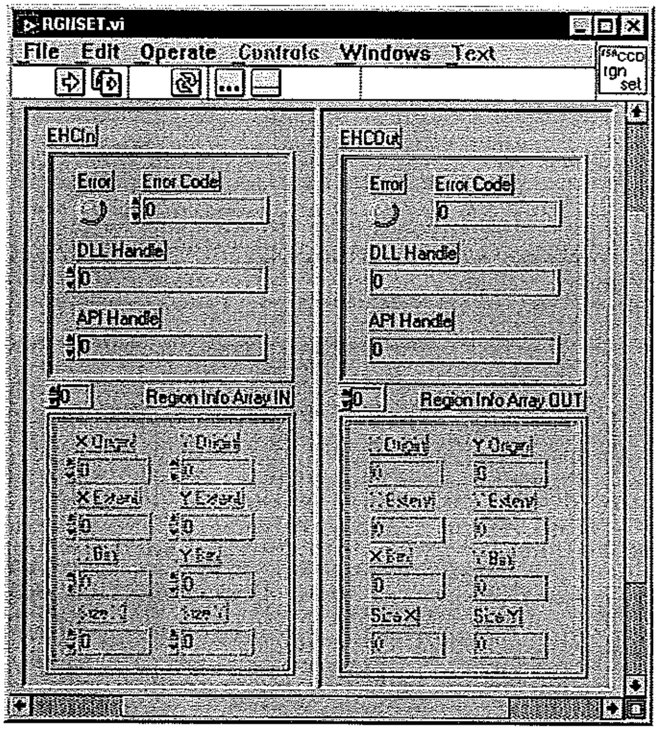
- EHCIn - Error/Handles Cluster Input
  - This cluster contains handles that are used by the driver and error information.
- Region Info Array IN
  - The CCD chip can be divided into 1 to 20 active regions. Each region can be individually defined by the parameters shown in the Region Info Cluster. Use the array index to select the region to be defined. NOTE: No two regions can overlap.

Output:

- EHCOut - Error/Handles Cluster Output
  - This cluster reports errors and contains handles used by the driver.
- Region Info Array OUT
  - The return array contains the actual region definition based on the Region Info Array IN. NOTE: Sometimes Region Info Array OUT may be slightly different from Region Info Array IN

Elements:

- Error - Error LED indicates if an error has occurred.
- Error Code - A numerical value will be returned when an error occurs. Use the CCDError.VI to decipher the Error Code numeric value.
- DLL Handle - Internally set by the driver. Do not change this.
- API Handle - Internally set by the driver. Do not change this.



EHCIn

Error

Error Code 0

DLL Handle 0

API Handle 0

0 Region Info Array IN

X Origin 0

Y Origin 0

X Extent 0

Y Extent 0

X Bin 0

Y Bin 0

Size X 0

Size Y 0

EHCOut

Error

Error Code 0

DLL Handle 0

API Handle 0

0 Region Info Array OUT

X Origin 0

Y Origin 0

X Extent 0

Y Extent 0

X Bin 0

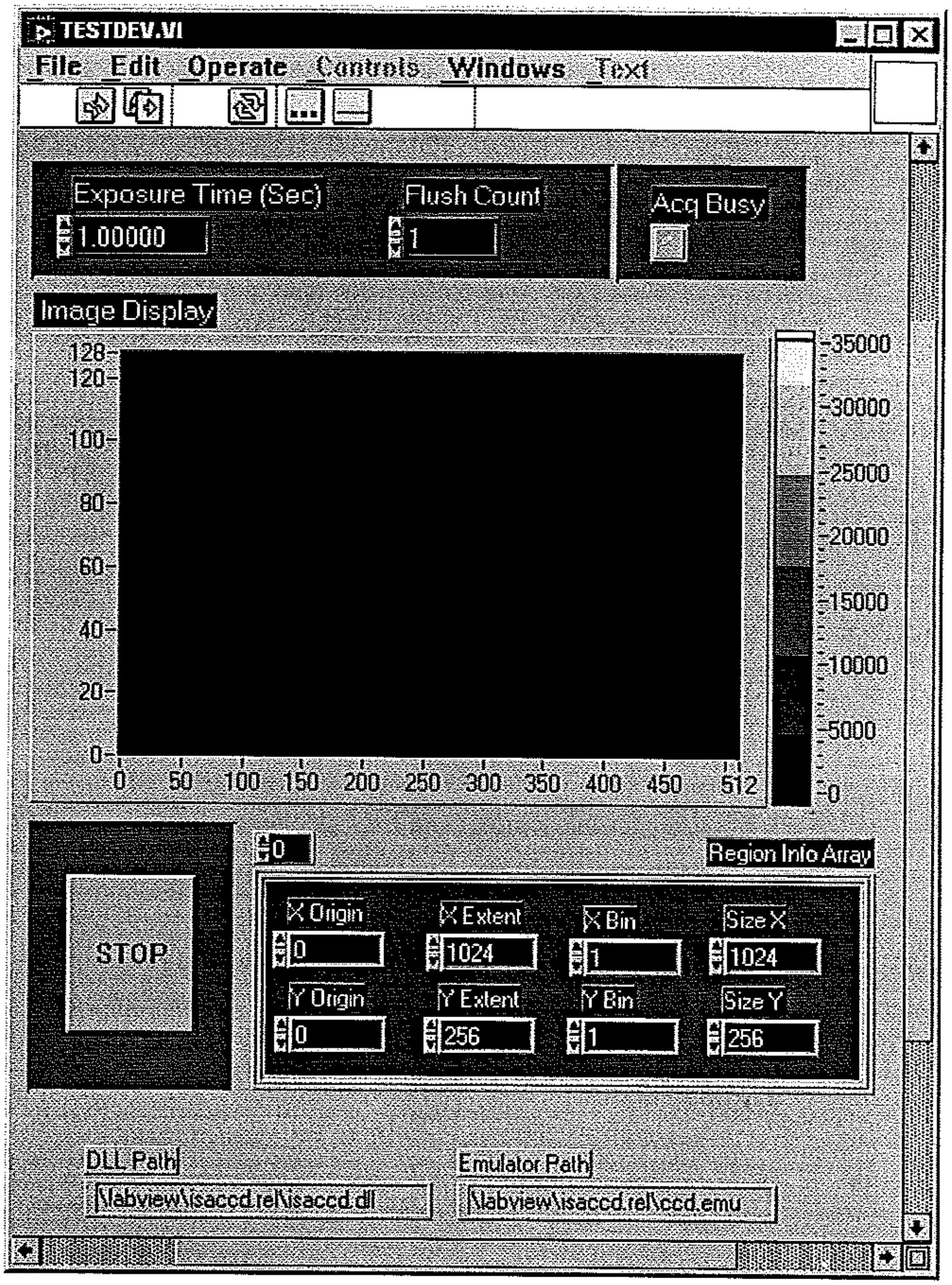
Y Bin 0

Size X 0

Size Y 0

TestDev.VI

This VI demonstrates acquisition and image display of CCD data utilizing the ISA CCD driver. Note: Specific help is available for each individual control, display and indicator.



Exposure Time (Sec) 1.00000

Flush Count 1

Acq Busy

Image Display

0

Region Info Array

X Origin 0	X Extent 1024	X Bin 1	Size X 1024
Y Origin 0	Y Extent 256	Y Bin 1	Size Y 256

DLL Path \labview\isaccd.rel\isaccd.dll

Emulator Path \labview\isaccd.rel\ccd.emu

TempMON.VI

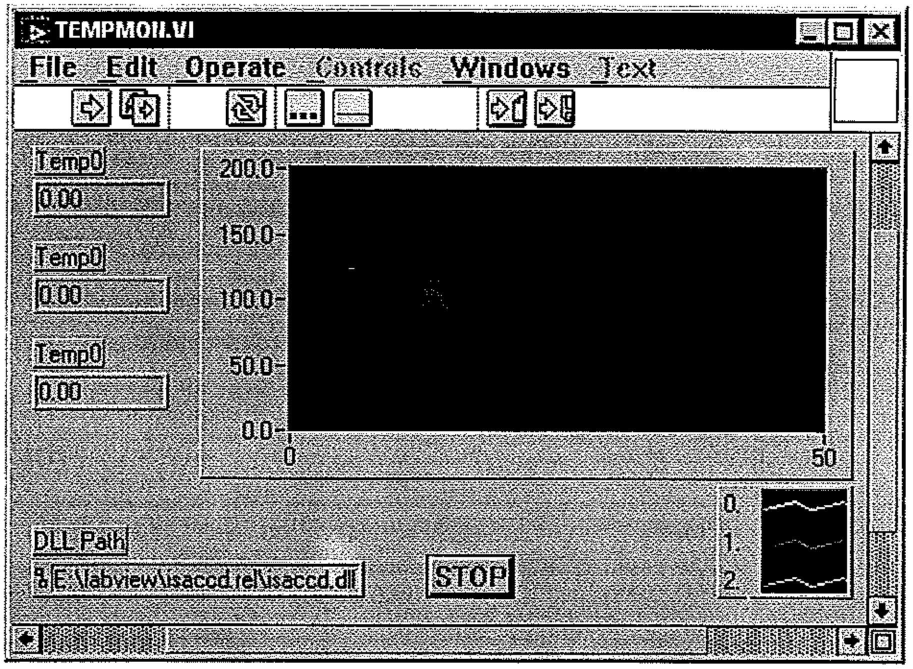
An example of how to monitors the temperature of the CCD.

Notes:

- Temp0 - CCD temperature
- Temp1 - Sink temperature
- Temp2 - V Therm temperature

DLL Path

- Must point to the isaccd.dll file.



Temp0
0.00

Temp0
0.00

Temp0
0.00

DLL Path
E:\labview\isaccd.rel\isaccd.dll

CCD General Properties

Flush Count (Index 0)
(PropGet/PropSet)
ParamArrays[0] - Number of Flushes

Exposure Time (ms) (Index 1)
(PropGet/PropSet)
ParamArrays[0] - Integration Time in millisecond

Valid Exposure Time Settings (Index 2)
(PropGet Only)
ParamArrays[0] - Minimum Integration Time in millisecond
ParamArrays[1] - Maximum Integration Time in millisecond

Temperature Set Point (Index 3)
(PropGet/PropSet)
ParamArrays[0] - Temperature Set Point in (0.1 Kelvin)

Valid Temperature Set Point Settings (Index 4)
(PropGet Only)
ParamArrays[0] - Min Temperature Set Point in (0.1 Kelvin)
ParamArrays[1] - Max Temperature Set Point in (0.1 Kelvin)

Double Correlation Integration Time (Index 5)
(PropGet/PropSet)
ParamArrays[0] - Double Correlation Time (Not Supported in CCD3000)

Gain (Index 6)
(PropGet/PropSet)
ParamArrays[0] - Detector Gain Level

Valid Gain Settings (Index 7)
(PropGet Only)
ParamArrays[0] - Minimum Detector Gain Factor
ParamArrays[1] - Maximum Detector Gain Factor

Detector Temperatures (Index 8)
(PropGet Only)
ParamArrays[0] - CCD Temperature
ParamArrays[1] - Sink Temperature
ParamArrays[2] - Thermostat Set Point

Chip Characteristics (Index 9)
(PropGet Only)
ParamArrays[0] - Chip Orientation
ParamArrays[1] - X Active Pixels
ParamArrays[2] - Y Active Pixels
ParamArrays[3] - X Pixel Spacing (μm)
ParamArrays[4] - Y Pixel Spacing (μm)

Shutter State (Index 10)
(PropGet/PropSet)
ParamArrays[0] - 1 to Open, 0 to Close(Index 0)

CCD 3000 Specific Properties

Note: Refer to the demo on the implementation of these new features.

Trigger Enable (Index 11)
(PropSet)
ParamArrays(0) - 0 Disable Trigger, 1 Trigger Per Acq, 2 Trigger Per Cycle

Blast Interval (Index 12)
(PropSet/PropGet)
ParamArray(0) - Interval Time in ms. Needed for a multi cycle acqusition. This is the total time (excluding data transfer time) of each cycle.

```
Cycle 0                                                   Cycle 1
>----------------------Interval Time----------------------<>----------------------Interval Time----------------------<
>-----------Exposure Time-----------<                        >-----------Exposure Time-----------<
```

Blast Cycle (Index 13)
(PropSet/PropGet)
ParamArray(0) - Number of cycles for the acquistion.
WARNING: This parameter must be set before defining acq regions.

Trigger Override (Index 14)
(PropSet)
Issue a software trigger instead of waiting for HW trigger pulse.

Blast Shutter Enable (Index 15)
(PropSet)
ParamArray(0) - 1 Enable shutter in multi cycle acq
0 Disable shutter open/close in multi cycle acq

ReadOut Time (Index 16)
(PropGet)
ParamArray(0) - Data Transfer time in ms for pervious acquisition