Biophotonic SOLUTIONS INC
ULTRAFAST PULSE SHAPING & COMPRESSION

MIIPS®2.0 Software Notes

Setting up a user-defined spectrometer through a TCP/IP connection

Revision 1.0

November 2014

Contents

1. Introduction

The MIIPS®2.0 software features built-in support of Ocean Optics miniature spectrometers as well as select Photon Control, Avantes spectrometers, and Andor cameras. Starting with version 2.077, it expands the portfolio of supported hardware by offering an ability to interface user-defined spectrometer-like devices with the core program via TCP/IP. This can be done by either adapting the provided open-source LV-based TCP/IP Spectrometer (Server) program or by utilizing any other means to establish the TCP/IP connection. In the later case, one has to follow the command syntax of the built-in TCP/IP client program. The notes below describe the open-source TCP/IP Spectrometer (Server) program and elaborate on the command syntax.

2. The TCP/IP Spectrometer Package

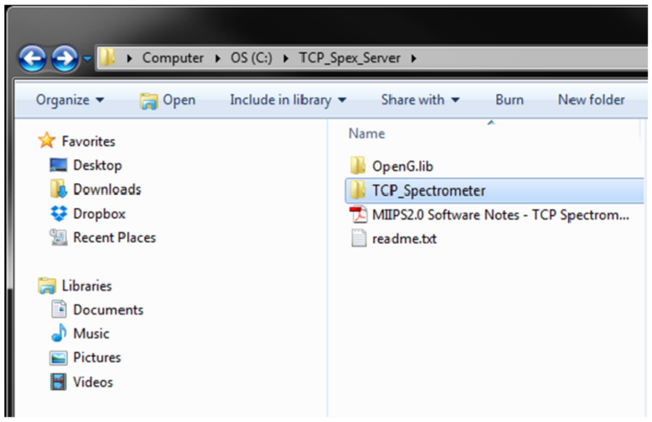
The TCP/IP Spectrometer package includes a built-in TCP/IP client, which is an integral part of the MIIPS®2.0 software, and an open-source TCP/IP Spectrometer (Server) program, which can be found in the software root directory (by default, it is C:\MIIPSv2\) as a compressed archive named "TCP_Spex_Server.zip". The archive can be unzipped into any suitable location on either the same computer or on any other network-connected computer that will be utilized to control the user-defined spectrometer.

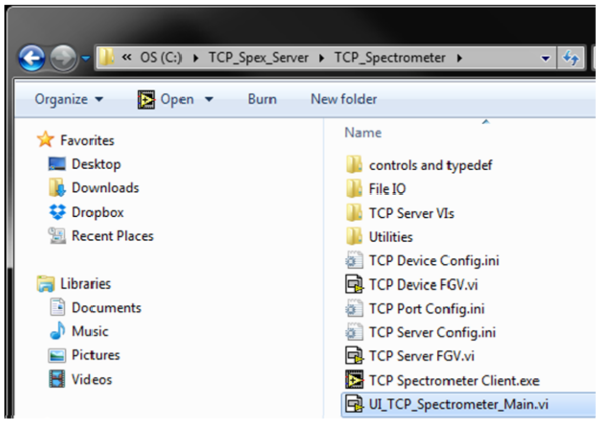
The open-source TCP/IP Spectrometer (Server) code is developed in LabView 7.1 and takes advantage of the OpenG library. It is highly recommended to use the same LV version to modify the program. If this is not an option, the open-source VIs can be up-converted to a later LV version, while the proper OpenG library can be added to the LV development module using the VI Package Manager from JKI (jki.net/vipm).

The OpenG library for LV7.1 is included into the distributed package. For 32-bit Windows operating systems, it is usually installed by the VI Package Manager into:

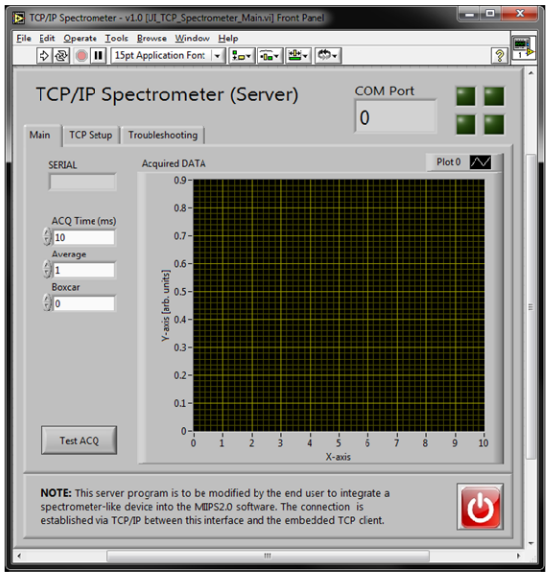
C:\Program Files\National Instruments\LabVIEW 7.1\user.lib\

It can also be copied there manually.





The "UI_TCP_Spectrometer_Main.vi" features the main user interface of TCP/IP Spectrometer (Server). All other VIs in the folder are sub-VIs of this program.


TCP/IP Spectrometer - v1.0 [UI_TCP_Spectrometer_Main.vi] Front Panel
File Edit Operate Tools Browse Window Help
TCP/IP Spectrometer (Server)
COM Port
Main
TCP Setup
Troubleshooting
Test ACQ
Plot 0
X-axis
Y-axis [arb. units]

COM Port: 0

SERIAL:

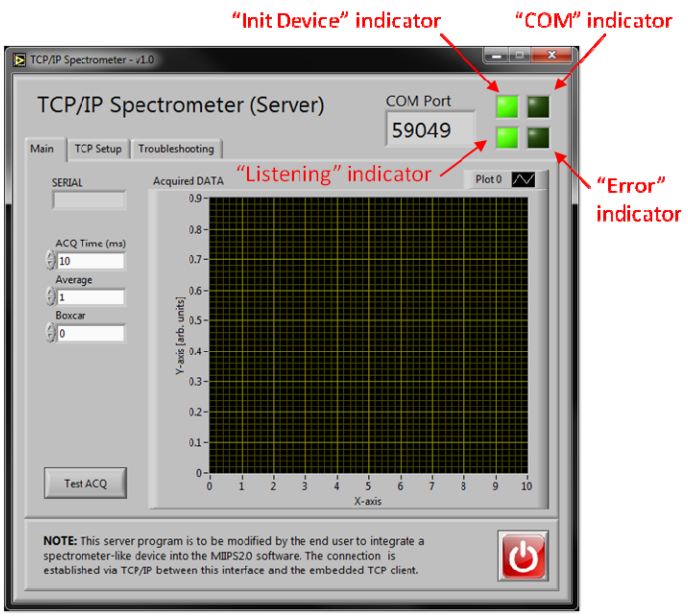
ACQ Time (ms): 10

Average: 1

Boxcar: 0

Acquired DATA

NOTE: This server program is to be modified by the end user to integrate a spectrometer-like device into the MIIPS2.0 software. The connection is established via TCP/IP between this interface and the embedded TCP client.



COM Port: 59049

SERIAL:

ACQ Time (ms): 10

Average: 1

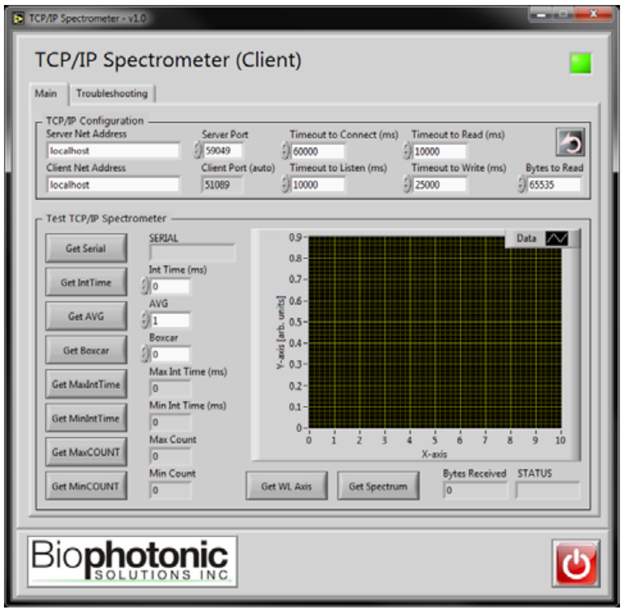
Boxcar: 0

Acquired DATA

NOTE: This server program is to be modified by the end user to integrate a spectrometer-like device into the MIIPS2.0 software. The connection is established via TCP/IP between this interface and the embedded TCP client.

The indicators in the top right corner summarize the current status of the program. Typically, the "Init Device" indicator is ON as a confirmation of successful device initialization. The "Listening" indicator is ON most of the time because the server is listening for connections whenever is doesn't process other requests. The "COM" indicator lights up only when data is received or transmitted via TCP/IP. The "Error" indicator lights up when an error occurs. It is reset as soon as the error is reported to the user.

The "TCP Spectrometer Client.exe" is a TCP/IP client program that can be used to test the server program without a need to run the MIIPS$^{®}$2.0 software.



TCP/IP Configuration

Server Net Address: localhost

Server Port: 59049

Timeout to Connect (ms): 60000

Timeout to Read (ms): 10000

Client Net Address: localhost

Client Port (auto): 51089

Timeout to Listen (ms): 10000

Timeout to Write (ms): 25000

Bytes to Read: 65535

Test TCP/IP Spectrometer

SERIAL:

Int Time (ms): 0

AVG: 1

Boxcar: 0

Max Int Time (ms): 0

Min Int Time (ms): 0

Max Count: 0

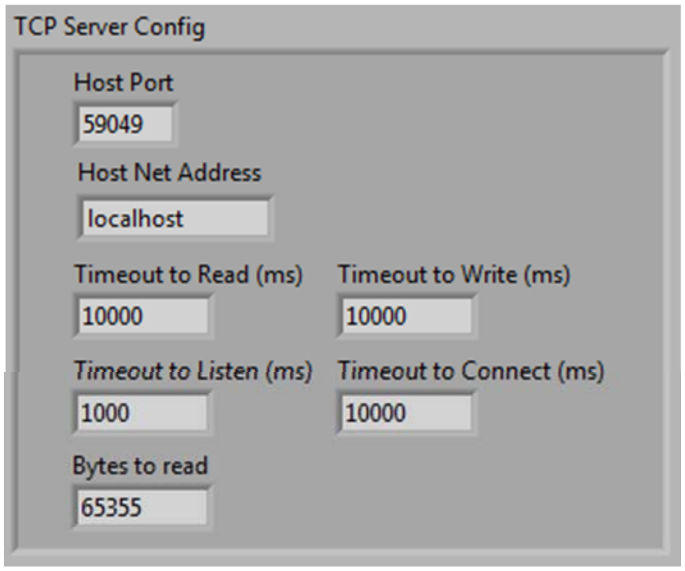
Min Count: 0

Bytes Received: 0

STATUS:

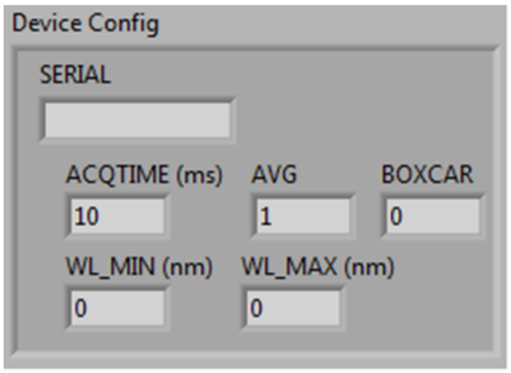
All start-up settings for the TCP/IP server program are stored in "TCP Server Config.ini" file. The start-up settings for the TCP/IP client program are stored in "TCP Port Config.ini" file, which is similar to one located in “globalfiles” sub-folder of the MIIPS®2.0 software. The device parameters can be stored in “TCP Device Config.ini” and are to be defined by the end user.

The “TCP Server Config.ini” and “TCP Device Config.ini” are loaded and updated in the TCP/IP server program by “TCP Server FGV.vi” and “TCP Device FGV.vi”, respectively. Their content can be modified by changing the definition of “TCP Server Config” and “Device Config” clusters.



TCP Server Config

Host Port: 59049
Host Net Address: localhost
Timeout to Read (ms): 10000
Timeout to Write (ms): 10000
Timeout to Listen (ms): 1000
Timeout to Connect (ms): 10000
Bytes to read: 65355



Device Config

SERIAL:
ACQTIME (ms): 10
AVG: 1
BOXCAR: 0
WL_MIN (nm): 0
WL_MAX (nm): 0

The “TCP Port Config.ini” is loaded and updated by the TCP/IP client program and should not be altered. It is implied the end user would only program the device-related content.

3. TCP/IP Spectrometer (Server)

The TCP/IP Spectrometer (Server) program is based on a single looped case structure controlled by a queue of commands.

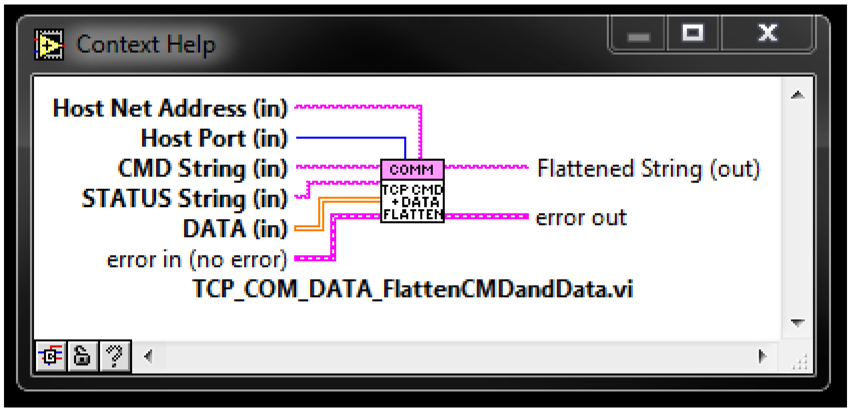
It starts up by loading information from the two ini-files (“Init” case), and then proceeds to initialize the device (“Init Device” case), the front panel controls & indicators (“Init FP” case), and finally, the listening port (“Init Listener” case).

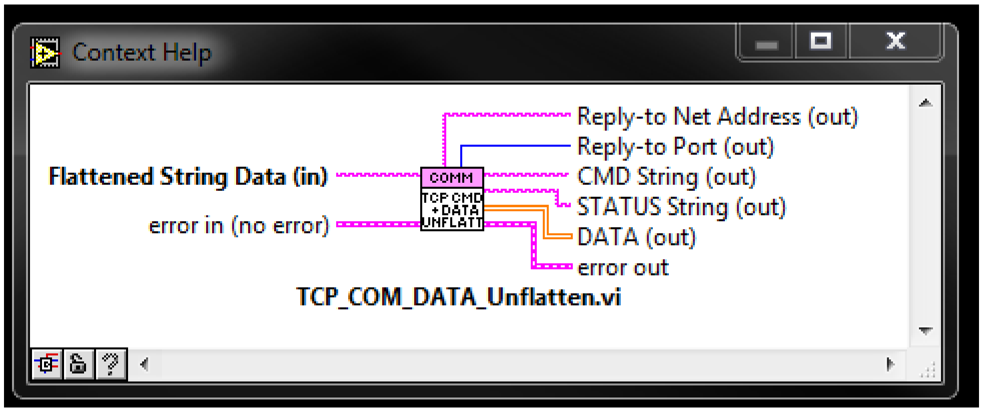
The initialization process is concluded to switching to the “Wait for User” case that processes the user activities on the front panel.

If no user activity is detected (Timeout), the program proceeds to listening for connections (“Listen” case); otherwise, it fulfills requests generated by the user activities.

If a connection is established during the listening session, the program reads the data (“Receive COM” case), interpret it (“Process COM” case), and follow the received instructions. If applicable, it compiles a reply and sends it using the provided net address and COM port. It is implied that the client program will be waiting for the reply.

The procedures of flattening and unflattering of the data in the format accepted by the MIIPS®2.0 software are performed by “TCP_COM_DATA_FlattenCMDandData.vi” and “TCP_COM_DATA_Unflatten.vi”, respectively.





There are several “cases” that are to be programmed by the end user integrating his or her device into the TCP/IP Spectrometer (Server) program.

They are:

a) “Init Device”
b) “Acquire DATA”
c) “Set ACQTIME”
d) “Set AVG”
e) “Set BOXCAR”
f) “Close Device”

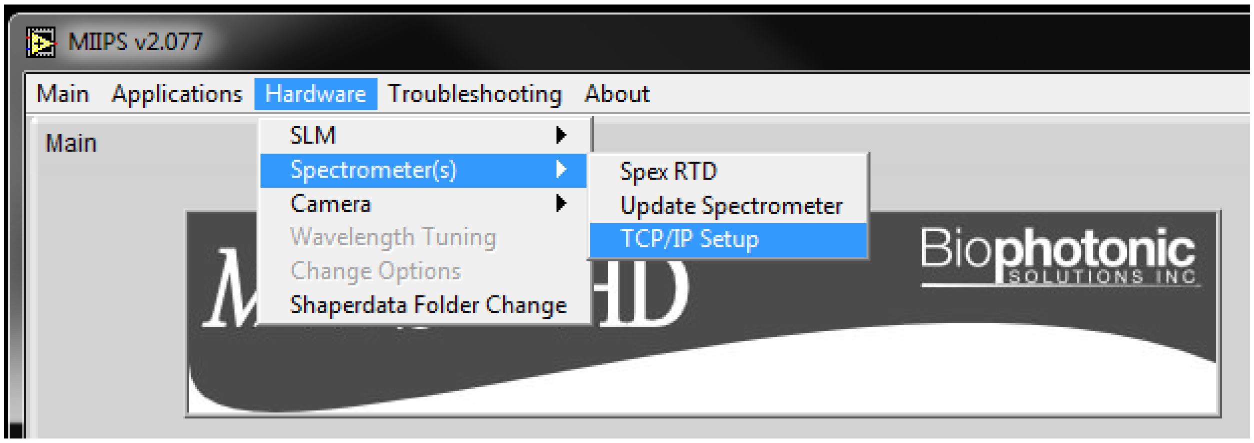
All of them imply communicating with the device; therefore, proper device-specific VIs should be imbedded into these “cases”; please refer to comments on the block diagram for further details.

When the “Exit” button is pushed, the program closes the device (“Close Device” case), updates the initialization files, and stops the while loop (“Exit” case).

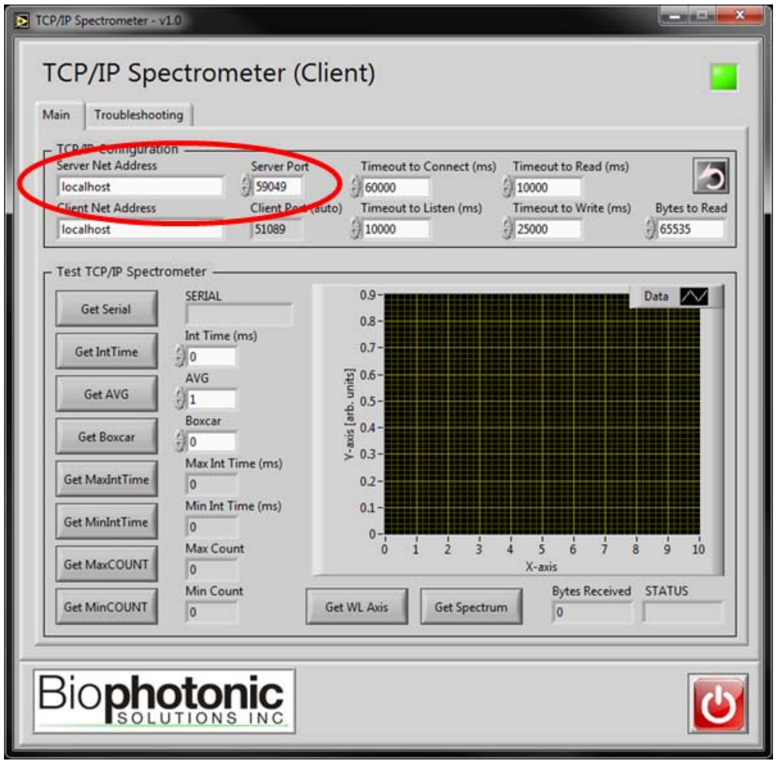
4. TCP/IP Spectrometer in MIIPS®2.0 Interface

Once the TCP/IP Spectrometer (Server) program is adapted to incorporate the user’s device and the server program is running, the user can set up the TCP/IP connection in the MIIPS®2.0 software by going to the program menu and selecting:

Hardware -> Spectrometer(s) - > TCP/IP Setup



A pop-up window with an interface similar to the test TCP/IP Client program would appear. It is important to confirm the TCP/IP setting for the server program. In particular, the COM port might have changed upon the initialization of the TCP/IP Spectrometer (Server) program. and the port # used by the MIIPS®2.0 software has to be matched to the one shown in the top right corner of the TCP/IP Spectrometer (Server) program interface.


TCP/IP Spectrometer (Client)

TCP/IP Configuration

Server Net Address: localhost	Server Port: 59049	Timeout to Connect (ms): 60000	Timeout to Read (ms): 10000	
Client Net Address: localhost	Client Port (auto): 51089	Timeout to Listen (ms): 10000	Timeout to Write (ms): 25000	Bytes to Read: 65535

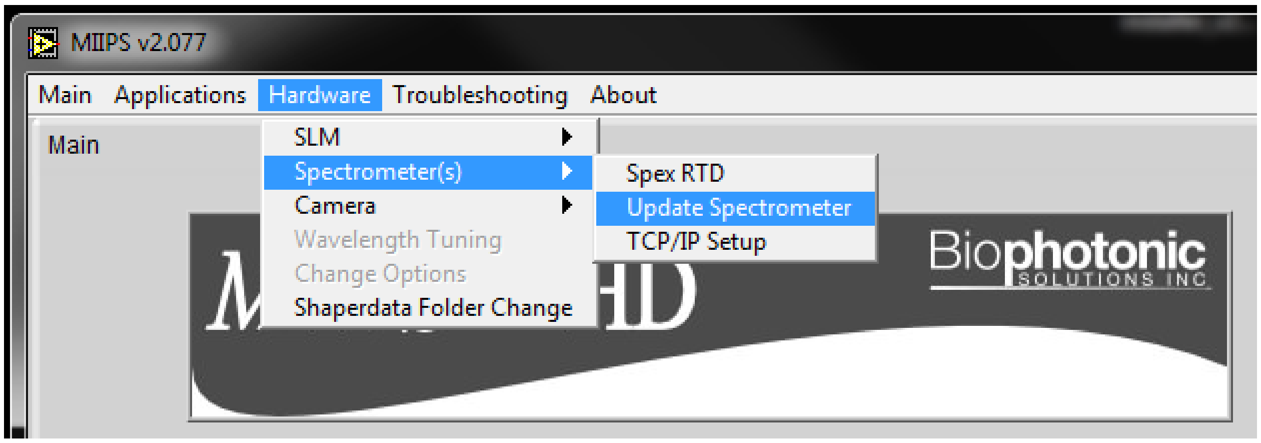
Test TCP/IP Spectrometer

SERIAL	
Int Time (ms)	0
AVG	1
Boxcar	0
Max Int Time (ms)	0
Min Int Time (ms)	0
Max Count	0
Min Count	0
Bytes Received	0
STATUS	

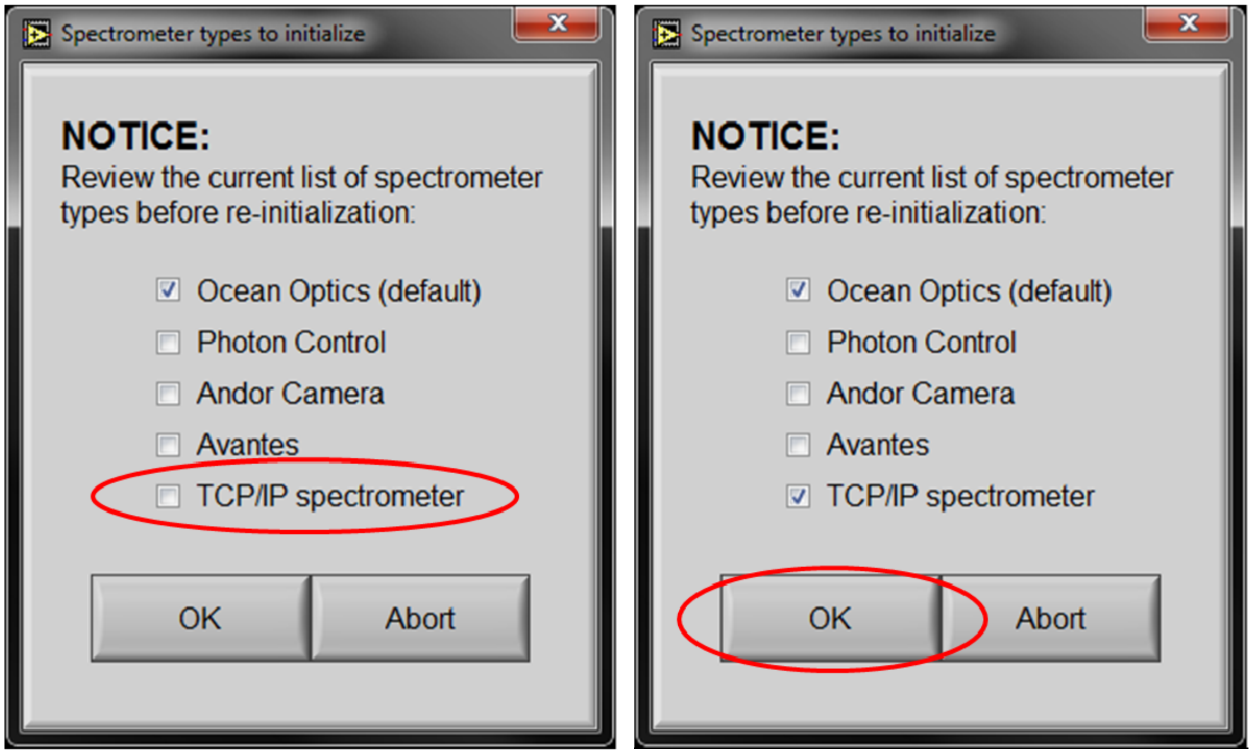
Once the TCP/IP settings are adjusted, the connection can be tested using controls in the "Test TCP/IP Spectrometer" section. When the window is closed (push the "Exit" button), the TCP/IP setting are updated.

To add the TCP/IP spectrometer, go to the program menu and select:

Hardware -> Spectrometer(s) - > Update Spectrometer



Select the TCP/IP spectrometer and click “OK”.



Spectrometer types to initialize

NOTICE:
Review the current list of spectrometer types before re-initialization:

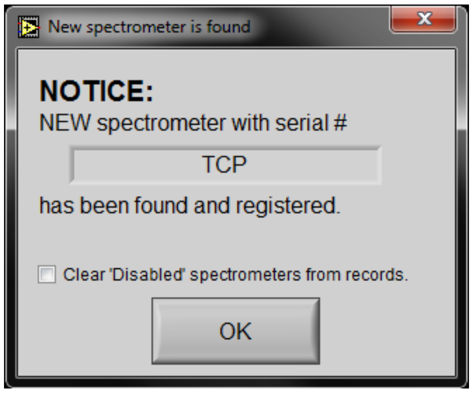
- [✓] Ocean Optics (default)
- [ ] Photon Control
- [ ] Andor Camera
- [ ] Avantes
- [ ] TCP/IP spectrometer

Spectrometer types to initialize

NOTICE:
Review the current list of spectrometer types before re-initialization:

- [✓] Ocean Optics (default)
- [ ] Photon Control
- [ ] Andor Camera
- [ ] Avantes
- [✓] TCP/IP spectrometer

If the spectrometer parameters are not found in the previously saved initialization data, a notification window will pop up, showing the serial number of the new device with the “TCP” prefix.


New spectrometer is found

NOTICE:
NEW spectrometer with serial #
TCP
has been found and registered.

Clear 'Disabled' spectrometers from records.

OK

It means that the TCP/IP-connected device has been successfully added to the list of the spectrometers accessible through the MIIPS®2.0 software interface.

5. Flattened Data Format

The flattened data received from or passed to the TCP/IP Spectrometer (Server) program or any other interface fulfilling the server functions is expected to be formatted as a single string built from an ordered sequence of the following sections:

1) //CMD_START//….//CMD_END//
2) //STATUS_START//…//STATUS_END//
3) //REPLY-TO-PORT_START//…/REPLY-TO-PORT_END//
4) //ADDRESS_START//…//ADDRESS_END//
5) //DATA_START//…//DATA_END//

The first section contains the command (CMD) string; see Appendix A for the list of commands The second section indicates the status of the task performed by the server. It is also used to pass the data type and the serial number of the device when requested. The third section contains the port # of the sender, converted into a string, and the fourth section is used to pass the net address. Finally, the firth section is used to pass a 2D array of data converted into a string.

For example, a command from a client program with net address "localhost" and reply-to COM port #59089 to set boxcar equal to 3 translates into:

```
//CMD_START//SET_BOXCAR//CMD_END////STATUS_START//0D_DBL//STATUS_END////REPLY
-TO-PORT_START//59089//REPLY-TO-
PORT_END////ADDRESS_START//localhost//ADDRESS_END////DATA_START//3.000000

//DATA_END//
```

The client's reply-to net address and port # are used by the server program to send back to the client requested data (if applicable), which could be an acquired spectrum, wavelength axis of the spectrometer or some device parameter value.

Appendix A. TCP/IP Commands and Associated Data

Command (CMD)	**STATUS**	**Description**
INIT_DEVICE	OK	Request to initialize the device remotely; reserved for future use.
CLOSE_DEVICE	OK	Request to close the device remotely; reserved for future use.
ACQ	2D_DBL (data size/type) Return: OK	Acquire spectrum; assumes to receive a two-row array, with the first row being X-axis and the second row – Y-axis data.
SET_ACQTIME	0D_DBL (data size/type) or ECHO Return: OK (if applicable)	Set the acquisition time (milliseconds). If STATUS=ECHO, a reply is expected with the actual parameter value.
SET_AVG	0D_DBL (data size/type) or ECHO Return: OK (if applicable)	Set the number of averaged data sets. If STATUS=ECHO, a reply is expected with the actual parameter value (integer number 1 or above).
SET_BOXCAR	0D_DBL (data size/type) or ECHO Return: OK (if applicable)	Set the BOXCAR value. If STATUS=ECHO, a reply is expected with the actual parameter value (integer number 0 or above).
?ACQTIME	0D_DBL (data size/type) Return: OK	Read the current acquisition time (milliseconds).
?BOXCAR	0D_DBL (data size/type) Return: OK	Read the current BOXCAR value (integer number 0 or above).
?AVG	0D_DBL (data size/type) Return: OK	Read the current number of averaged data sets (integer number 1 or above).
?MAX_ACQTIME	0D_DBL (data size/type) Return: OK	Read the upper bound of the device acquisition time (milliseconds). NOTE: This parameter is used to set the limit of the “Integration Time” control in the MIIPS®2.0 software.
?MIN_ACQTIME	0D_DBL (data size/type) Return: OK	Read the lower bound of the device acquisition time (milliseconds). NOTE: This parameter is used to set the limit of the “Integration Time” control in the MIIPS®2.0 software.
?MAX_COUNT	0D_DBL (data size/type) Return: OK	Read the upper bound of the acquired signal amplitude. NOTE: This parameter is used in the MIIPS®2.0 software for diagnostics; in particular, to check on the spectrometer saturation.
?MIN_COUNT	0D_DBL (data size/type) Return: OK	Read the upper bound of the acquired signal amplitude; reserved for future use.
?SERIAL	SERIAL Return: SERIAL # of the device	Read the serial # of the device.
?X	1D_DBL (data size/type) Return: OK	Read the X-axis, which would be the wavelength axis for a spectrometer-like device. The assumed units – nanometers.

Other possible values for the STATUS string are:

“N/A”, “UNKN_CMD”, “UNKN_STAT”, “ERR”.

If the received command is not applicable to the operation of the TCP/IP-connected device, the reply status might be set to “N/A”.

If the received command is unknown, the reply status might be set to “UNKN_CMD”.

If the status of the operation is unknown, the user might want to set the reply STATUS to “UNKN_STAT”.

Finally, if an error has occurred and the requested operation has not been completed, the reply STATUS might be set to “ERR”.

The built-in TCP/IP client and the supplied TCP/IP Spectrometer (Server) program offer limited error handling capabilities but the end user can enhance those by modifying the original open-source code.