Bristol Wavemeter Interface Documentation

Martha Roseberry

August 1, 2008

1 Basics

The LabVIEW wavemeter interface can be found in the Bistol Wavemeter folder on the desktop of the computer "qol". Within this folder is a folder entitled LabVIEWInterface, and all work on this interface is saved in this folder, including the final versions of the interface. There are two versions of the program, one entitled wavemeterInterface.vi and one entitled wavemeterInterfaceInternet.vi. Also there is a wavemeterReader.vi designed to work with the wavemeterInterface.vi. The other three VIs in the folder are earlier versions.

The internet version is designed to publish the information to the internet. The other interface is designed to work with and can only be run with the DataSocket server. These instructions are for the DataSocket version of the program. However, the internet version of the program operates exactly like the DataSocket writer program without the DataSocket parts.

The writer interface requires that several steps be preformed in the appropriate order. First make sure that the wavemeter is turned on and connected to the computer. Second, open the program and open the DataSocket server. Next, on the interface of the writer select the units that are desired, the medium through which the laser is traveling, and select the 'Log to File' button if you wish to save the data. All of these options must be selected before the programs begins running, and if you wish to change them afterwards, the program must be stopped and restarted.

After selecting the desired settings, press the white arrow in the upper left hand corner of the interface to start the program. If running properly, the program should update the wavelength and the power in the given units twice every second. On occasion, the program fails to receive the appropriate output, and displays a near zero number. This is normal but should happen infrequently.

To stop the program, press the large 'STOP' button. **DO NOT PRESS THE RED STOP SIGN BUTTON.** Pressing the 'STOP' button insures that the port will be closed. If the port is left open, the program will not run a second time without restarting LabVIEW.

To run the reader, simply open the reader and press the white arrow in the upper left hand corner to start. To stop the reader press the 'STOP' button. If the reader is not working, the most likely problem is that the address of

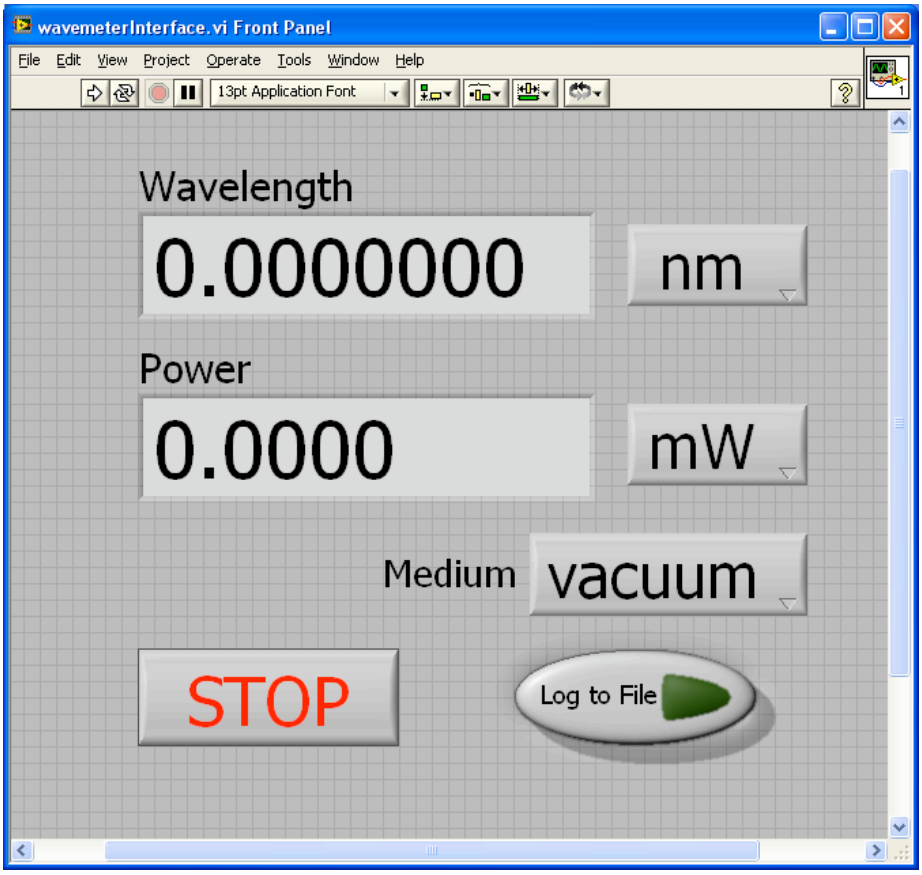

Figure 1: Front panel of the writer program. This program must be run on the computer to which the wavemeter is connected.

the server is incorrect. See the section on the reader block diagram to fix this problem.

2 Writer Front Panel

The writer front panel displays the wavelength and the power in text fields that the user cannot interact with. However, the front panel has six main parts with which the user can interact. Two are the start and stop buttons. The start button is the white arrow in the upper left hand corner of the interface, the stop button is the button with 'STOP' writen on it in large red letters. **DO NOT PRESS THE RED STOP SIGN.** The program is designed to write to the DataSocket server, which must be turned on in order for the program to operate. The DataSocket server can be found from the Start menu via the path

Programs:National Instruments:DataSocket:DataSocket Server.

There are three drop down menus which allow the user to specify the form he/she wishes to view the output. The first is the wavelength units menu, located next to wavelength text field. This menu has three options: nanometers (nm), inverse centimeters (1/cm) and gigahertz (GHz). The default setting for this menu is nanometers. The second menu is the power units menu located next to the power text field. This menu has two options: milliwatts (mW) or decibels (dBm). The default setting for this menu is milliwatts. The third menu specifies the medium through which the laser is traveling. This menu is located below the wavelength and power readings, and has two options: vacuum and air. The default setting for this menu is vacuum. **All three of these menus must be selected and set before the program is running.** Once the program is running, the user can change the value displayed, but the program will not check for the new settings until the program is restarted. This means it is possible to make the output units not match the units displayed on the menu. Thus, **it is recommended the menus are not changed while the program is running**, because it would be very easy to confuse which units are actually being displayed.

The last option the user can select is to log the data to a file. If the button 'Log to File' is selected, it will glow green and when the program starts it will output the time, the wavelength, and the power to a user specified file.

3 Writer Block Diagram

The block diagram of the writer program is shown in Figure 2. To access the data from the wave meter, the program uses multiple Call Library Function Nodes. The first of these, on the far left, is the library function to open the USB port that the wavemeter is connected to. The input for this library is the Comm Port number, which, unless it has changed, is Comm Port 3. The return of this function must be passed to the other Call Library Function Nodes. To configure this Library Node, right click on it and select Configure. You should see a window like that shown in Figure 3. If you click on the parameters tab, this function has two parameters. The return type has Type Numeric and Data type Signed 32-bit Integer. The commPort has Type Numeric, Data type Signed 32-bit Integer and Pass Value.

The Open function passes its return first to the SetUnits and SetMedium functions. The drop menus from the front panel, labeled Lambda Units, Power Units and Medium, input their values to these functions through the left I32 connection. The return type from the Open function is passed to each of these libraries through the left U32 connection. The first of these three functions is the SetLambdaUnits function. If you select to configure this function, a window like that shown in Figure 4 will appear. This configuration is the same for both the SetPowerUnits function and the SetMedium function, except their function names will be CLSetPowerUnits and CLSetMedium, respectively. The parameters for all three functions are the same. A return type which has Type

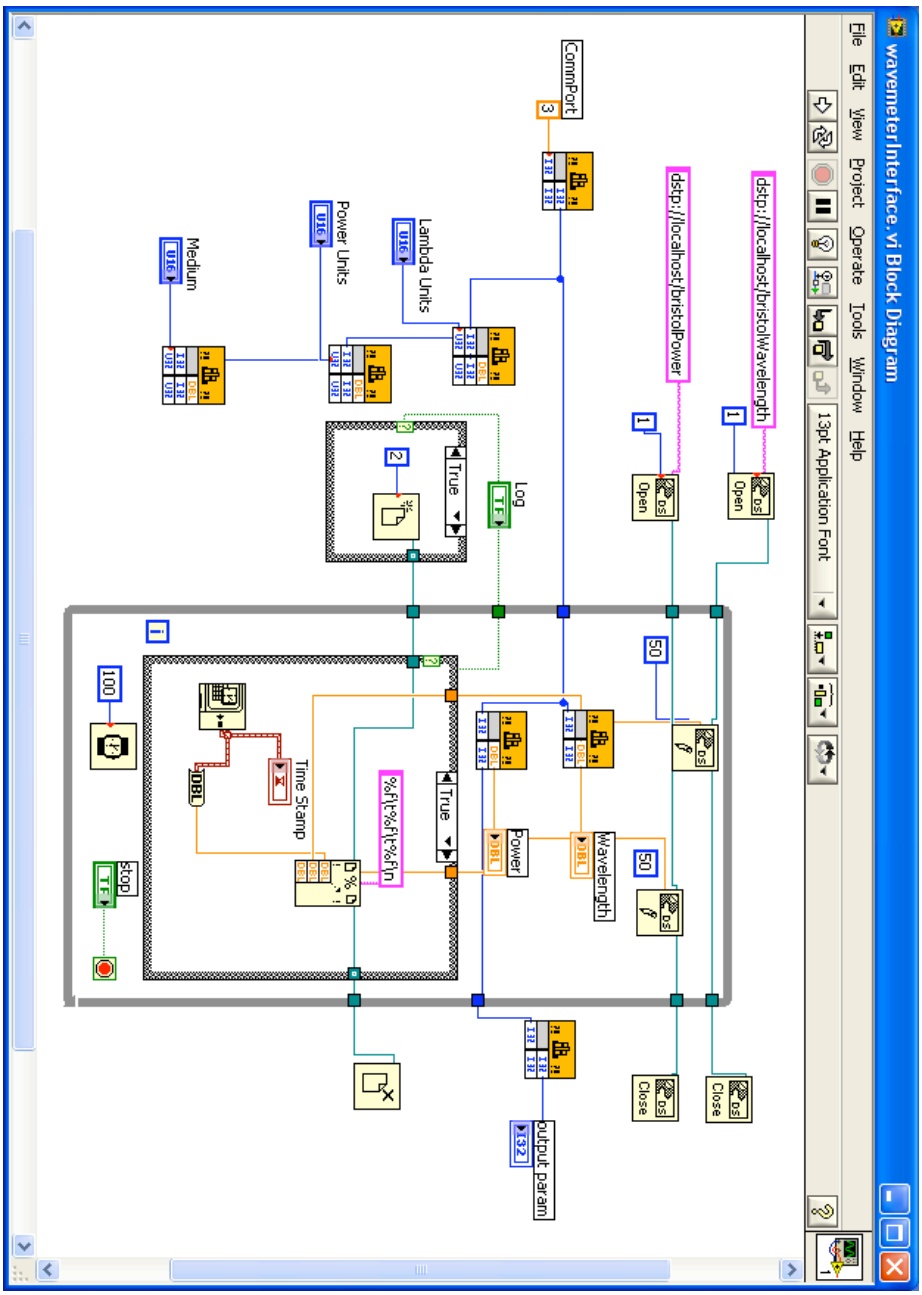

Figure 2: The block diagram for the writer program. Program is designed to run with DataSocket and will only run properly if the DataSocket server is turned on and the writer has the correct address for the server.

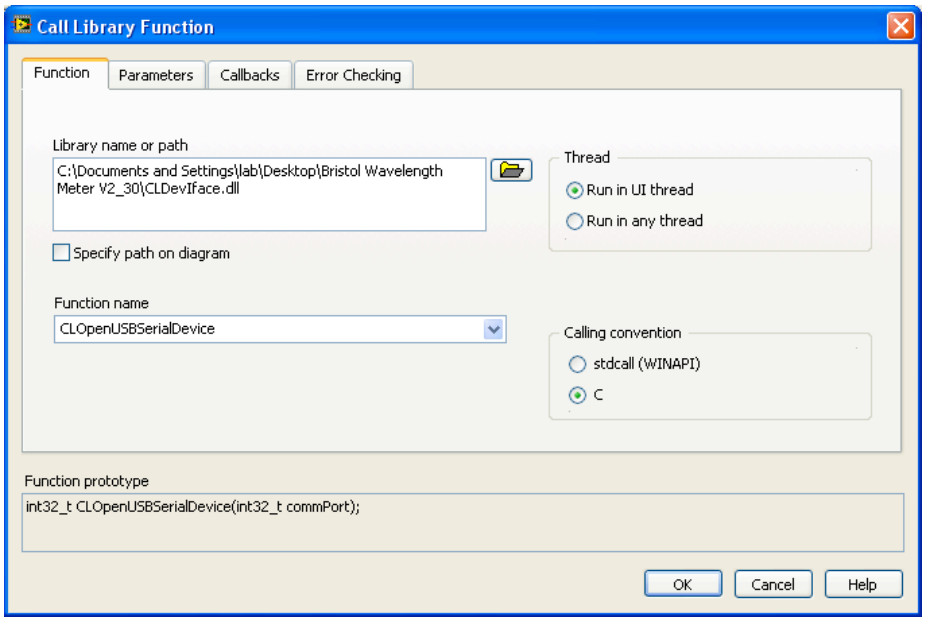

Figure 3: Configuration window for the Open Device Call Library Function Node.

Numeric and Data type 8-byte Double. A devHandle which has Type Numeric, Data type Signed 32-bit Integer and Pass Value. The devHandle is the port number returned by the Open function. Lastly an arg2 with Type Numeric, Data type Unsigned 32-bit Integer, and Pass Value.

The Open function also passes its return to the GetLambdaReading and the GetPowerReading. These functions are configured as shown in Figure 5. The function name for the the GetPowerReading function is CLGetPowerReading. Both functions have two parameters. The return type has Type Numeric and Data type 8-byte Double. The devHandle has Type Numeric, Data type Signed 32-but Integer and Pass Value. These functions output the wavelength and the power given to them by the wavemeter to the text fields on the front panel. They also give these values to the DataSocket server and to a file, which will be explained later.

The Open funtion returns passes from the GetReading functions to the Close function. The Comm Port must be closed for the program to work a second time. The configuration of the Close function library is shown in Figure 6. This function has two parameters. The return type has Type Numeric and Data type 32-bit Integer. The arg1 has Type Numeric, Data type Signed 32-bit Integer and Pass Value. The Close function must return a value, which is why it is connected to the output parameter.

On the block diagram, three case structures are present. These allow the program to write the data to a file. The Log to File button is connected to each

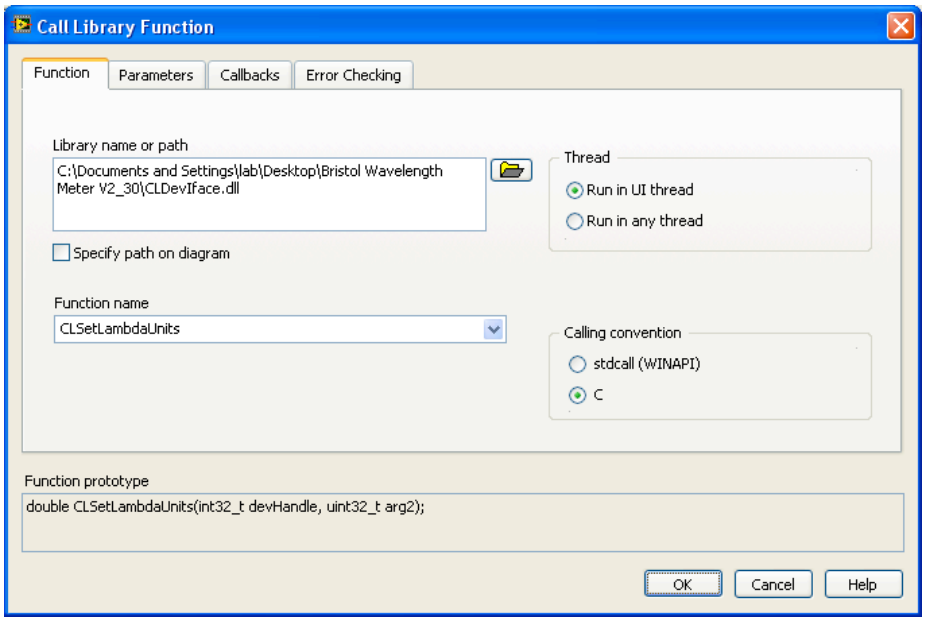

Figure 4: Configuration window for the Lambda Units Call Library Function Node.

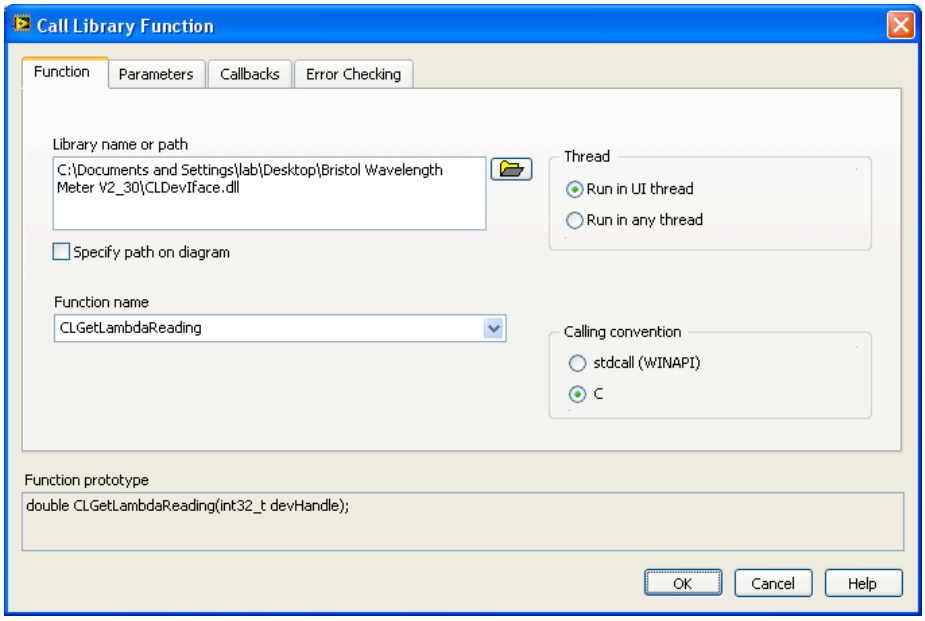

Figure 5: Configuration window for the Get Lambda Reading Call Library Function Node.

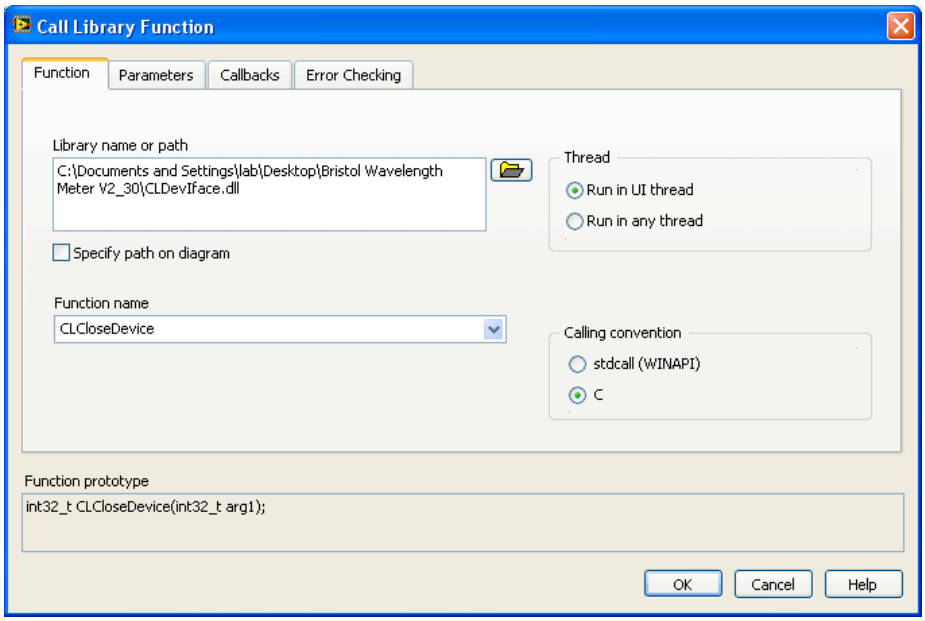

Figure 6: Configuration window for the Close Device Call Library Function Node.

case structure. If the log to file button is True, then the program will open a file and request the user to name the file. Then the name of the file is passes to the Format Into File function. The pink box tells the function how to format the file. This function takes three double values. The first is the time, provided by the TimeStamp. The second is the wavelength from the GetLambdaReading function and the third is the power from the GetPowerReading. Once the stop button has been pressed, the file will be closed by the Close File function.

Also present in this program are the DataSocket VIs. This program is intended to be run with the DataSocket server and will not work unless the DataSocket server is running. The pink boxes in the upper left corner of the block diagram provide the address to which the data will be written. In this case, the server is being run on the same computer as the program, and 'localhost' is used. The server can be run on any computer, but then 'localhost' must be replaced by the correct address for the computer on which the DataSocket server is running. These address are passed to the DataSocket Open VIs, which allow the program to write to the server. The DataSocket Open functions are connected to the DataSocket Write functions. One of these functions is connected to the GetLambdaReading function, and the other to the GetPowerReading function. They are also connected to a 50 millisecond delay. These function will take the wavelength and power values and write them to the DataSocket server. The DataSocket Write functions are connected to the DataSocket Close functions.

The last structure in the block diagram is the while loop. This large loop

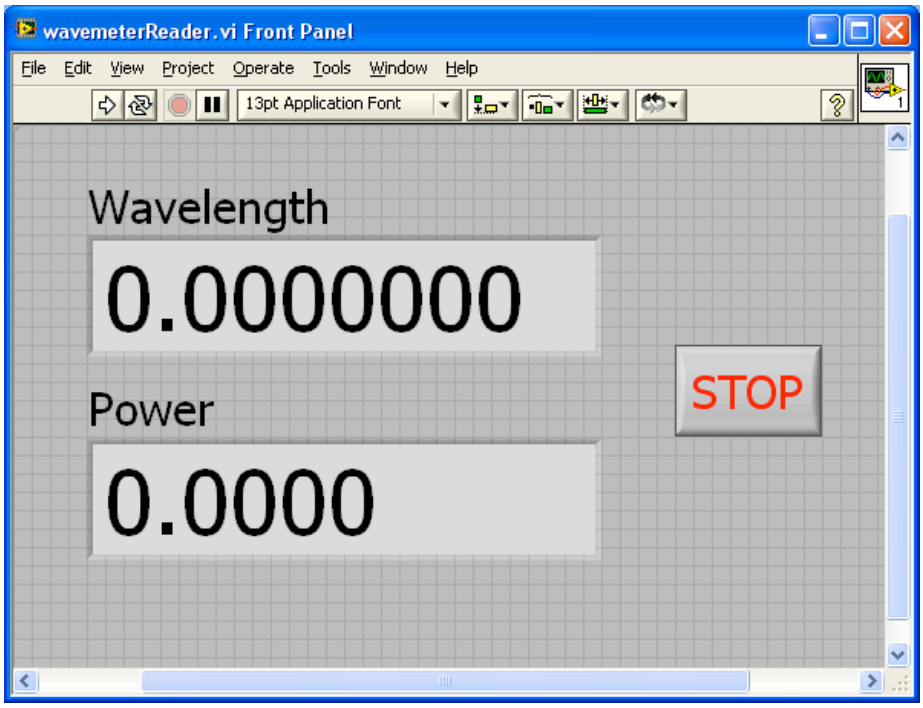

Figure 7: The front panel of the reader program.

encloses most of the program. Its function is to cause the reading of the data to the interface, the writing to the DataSocket Server, and the writing to the file to happen continuously as the long as the program is running. The loop has a built in 100 millisecond delay. The loop will run as long as the red STOP button has not been pressed. Once the STOP button has been pressed, the program will exit the loop. Everything on the left of the loop happens before the loop has started. Everything on the right of the loop happen after the STOP button has been presses. If the stop sign is pressed rather than the STOP button, these three close functions on the left of the loop will not happen.

4 Reader Front Panel

The reader front panel is entirely for viewing the wavemeter output, not for changing any settings. The user should start the writer first, see the section on the writer front panel, making any necessary changes to the default settings, and then start the reader. To run the reader, press the white arrow in the upper left hand corner. The program should then begin to display the wavelength and power in the units set on the writer front panel. The reader will update its value approximately one update behind the writer. To stop the reader simply press

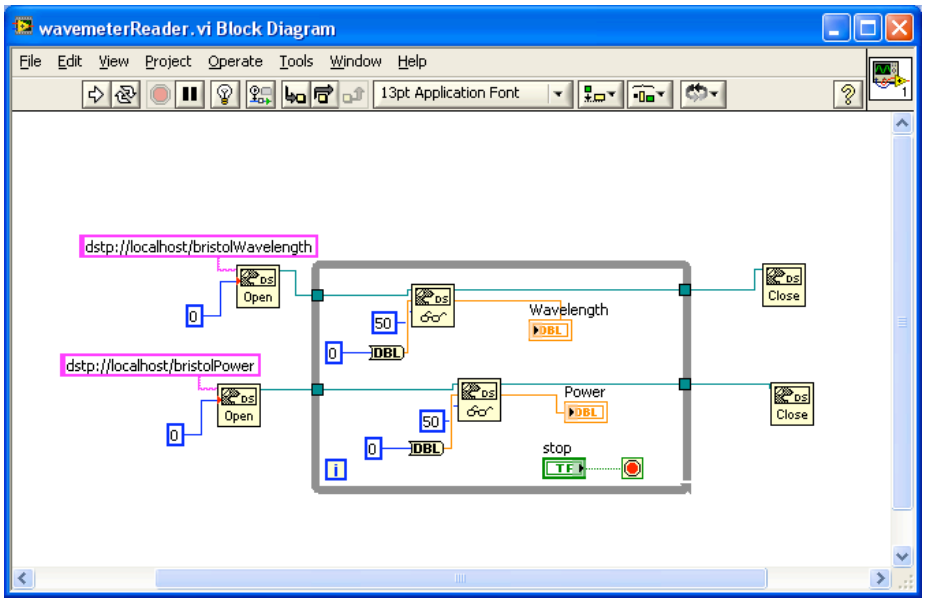

Figure 8: The block diagram of the reader program. The reader program will do nothing if it is not correctly directed to the DataSocket server. Also, the writer must be writing to the server for the reader to work.

the 'STOP' button. Stopping the reader will not stop the writer or DataSocket. These must be closed separately.

5 Reader Block Diagram

The block diagram for the reader program is shown in Figure 8. Like in the writer, the pink boxes provide the address of the DataSocket server, and unless the program is being run on the same computer as the server, 'localhost' must be changed to the correct address of the server. The addresses are input for the DataSocket Open functions. Inside the while loop, the return from the Open functions are connected to the DataSocket Read functions. These functions have a 50 millisecond time delay, and are given a type Double. They output the values from the server to the text fields on the front panel. Once the program is stopped, the server is closed with the DataSocket Close functions.

6 DataSocket

Figure 9 shows a view of the DataSocket server window. The server is opened by the path Start:Programs:NationalInstruments:DataSocket:DataSocket Server. When the programs are running, the statistics for the server should be nonzero.

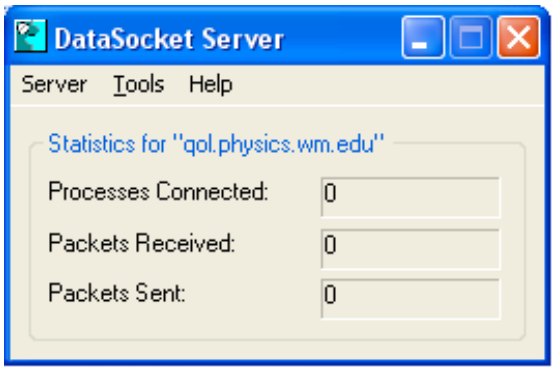

Figure 9: DataSocket server window.