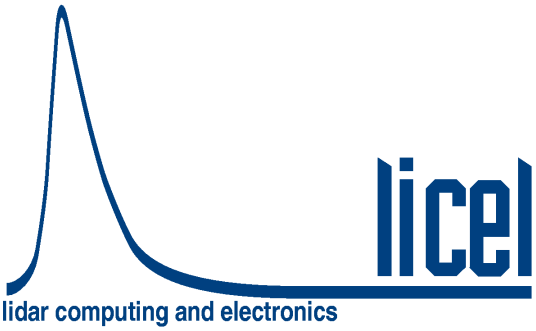


Licel Waverider – Installation and Reference Manual

Licel GmbH

May 8, 2024

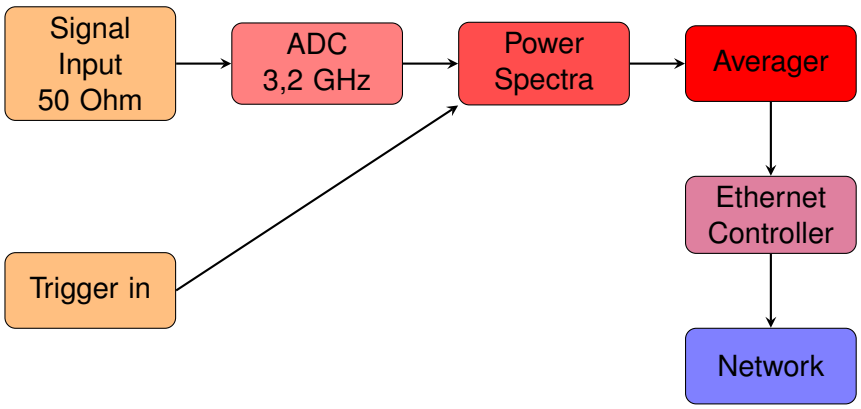
Contents

Chapter 1

Introduction

1.1 System description

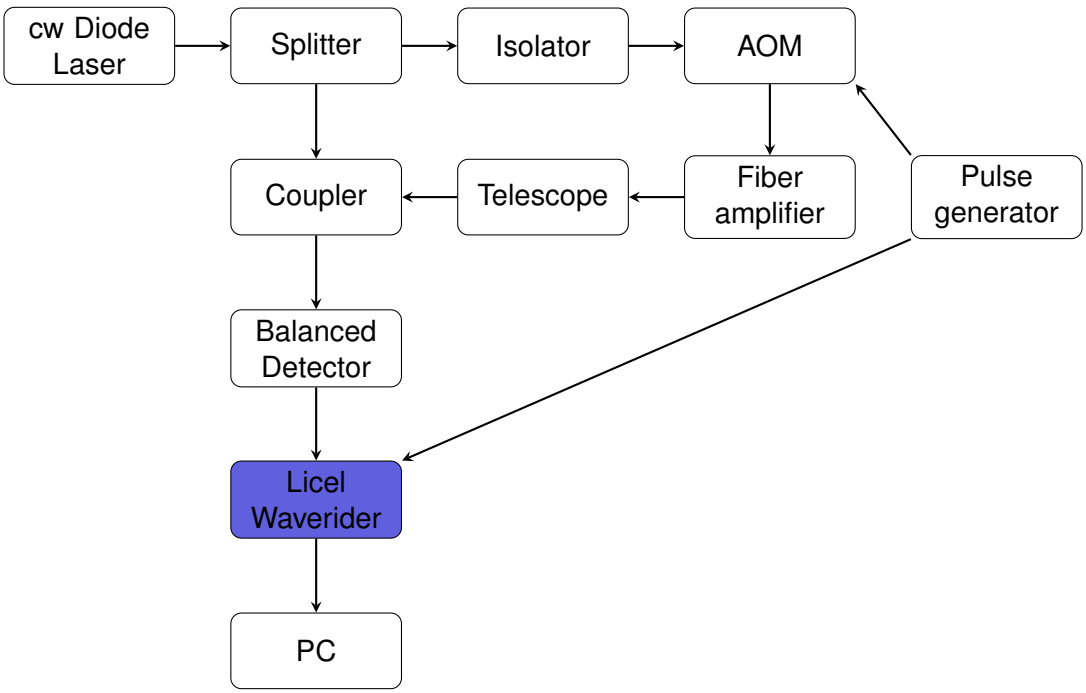
The Licel Waverider module is designed to be used in pulsed coherent Doppler Wind systems. It takes the output of balanced fiber detector together with a synchronizing trigger pulse and computes power spectra from an 3,2 GHz AC coupled 12 bit ADC signal which are then averaged and transmitted over Ethernet to a PC. This hardware power spectra computation and averaging allows to run the system with a small amount of laser shots lost while keeping the SNR much higher due to the 12 bit ADC.



The FFT size is fixed 2048, which corresponds to 96m. The power spectra are averaged over a predefined number of shots and the accumulated power spectra are transferred over a TCPIP connection to the PC. Licel provides software modules to acquire the data and store the data into data files. They can later be processed to compute the wind signal. Further the software is provided to view the data files, to have an oscilloscope like interface to the instrument and to configure the network interface.

1.2 Typical Experimental Setup

The instrument is designed to be a component of larger lidar system. The white boxes should be provided by the system integrator. The blue box is the data acquisition system described here.



We also provide a complete acquisition software which can serve both as a starting point to acquire and store data and as a example how to implement a wind lidar acquisition software. The computation of the wind speed is intentional not provided and put on the integrators, as this instrument serves as a instrument, where the user can completely control the process from the raw power spectra to the final wind speeds. We provide a peak finding algorithm implementation based on *Engelmann, R. 2010 Aerosol vertical exchange in the convective planetary boundary layer: Turbulent particle flux measurements with combined wind and aerosol lidar, PhD Thesis, University of Leipzig,Germany, 135 pp.*

For proper finding of the peak position one should do a background measurement, so that a background slope does not influence the peak position. For a background measurement the emission from the laser should be blocked.

1.3 Cable Connections

1.3.1 Rear

Trigger input, DIO Cable
Input High >2.0V
Input Low <0.8V

Signal Input, SMA 50Ω

Chapter 2

Software Installation

Licel provides a package of software modules for setting up the Licel Waverider Controller for network operation, and for operating the Licel Control Modules. These software modules are written in LabVIEW's G language. The software is provided as LabVIEW source for users who have LabVIEW 2018 or later installed, or alternatively as a set of Windows applications. The Windows applications come within a Windows Installer package for an easy installation on your Windows 10 computer. Licel provides the software on a CD ROM and for download (http://www.licel.com/wind.htm).

2.1 Preparation

Windows Application Users

If you have used older versions of Licel Windows applications it is recommended to backup existing initialization files (`*.ini`).
Search the existing installation directory of the older version of Licel Windows applications (standard: `<Program Files Directory> \Licel`) and backup all files with the ending `*.ini` to an archive file (zip, ARJ, TAR, etc...).

LabVIEW Users

If you have used older versions of Licel LabVIEW libraries it is necessary to remove and backup older versions.

1. Backup all your current Licel software libraries, in case you want to restore them, by either compressing them (zip, ARJ, TAR, etc...) .
2. Scan your disks to find all versions of the following files and delete them once you have made backups of them

```
project\WIND_src.lvproj
shared\ctl\*.*
shared\Licel TCPIP.llb\*.*
shared\LicelUtil.llb\*.*
shared\user.lib\errors\*.*
WIND_PC\SearchControllers.llb\*.*
WIND_PC\driver\*.*
WIND_PC\WindTDMS.llb\*.*
WIND_PC\WindViewer.llb\*.*
WIND_PC\WindAcquis.llb\*.*
WIND_PC\WindLiveDisplay.llb\*.*
```

```
WIND_PC\WindTCP_Server.llb\*.*
WIND_PC\LicelGraph.llb\*.*
WIND_Simulation\*.*
```

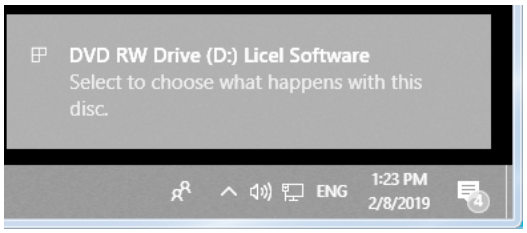
Please note: Licel may have provided individual software solutions with additional or less LabVIEW library files than noted in the list above.

3. Search the directory your older version of Licel LabVIEW libraries reside and backup all initialization files (`*.ini`).
4. The LabVIEW sources are delivered including the following files and directories:
   - `installation.txt` a short description file
   - `WIND PC` a directory containing the LabVIEW vis and initialization files that run on the PC.
   - `WIND Simulation` a directory containing the Virtual controller for simulation of a Waverider system.
   - `project` a directory containing the LabVIEW project `Wind.lvproj`
   - `shared\user.lib\errors\ Licel-errors.txt` Licel error code file.
   - `shared\user.lib\errors\ WIND-errors.txt` Wind error code file.
5. The `Licel-errors.txt` and `Wind-errors.txt` files should be copied to the user error directory below the LabVIEW installation path. A typical path would be `C:\Program Files (x86) \National Instruments \LabVIEW 2018\user.lib\errors`.

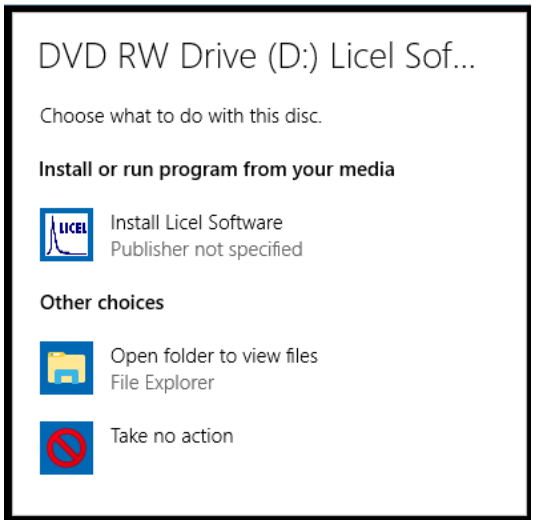
2.2 The Licel CD ROM

The standard CD ROM provided by Licel contains both, the LabVIEW sources and the Windows Installer for installing the Windows applications, and furthermore a documentation folder. Licel may add customer specific components on the CD ROM.

1. Insert the Licel CD into your CD ROM drive.
2. In Windows 10 you will normally be notified by a pop-up message at the bottom right corner of the main monitor.

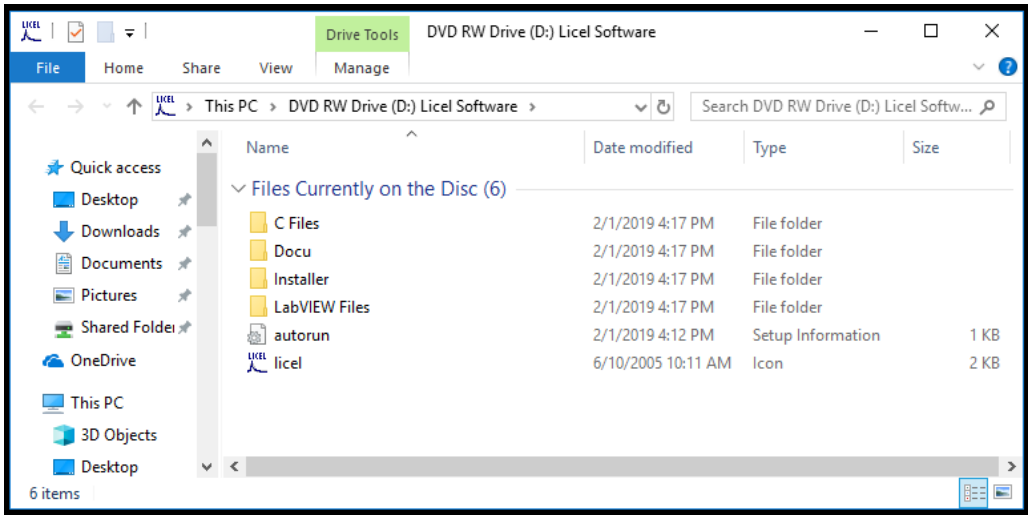


   Please click on the pop-up message.
3. The following selection dialog should appear:

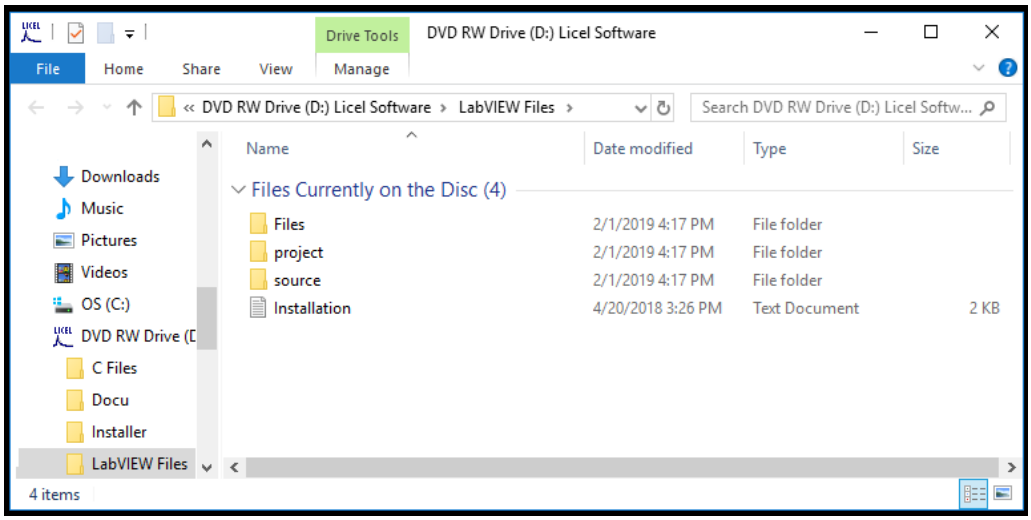


In older Windows operation systems a similar dialog will automatically come up.

- Press *Install Licel Software* to start the Windows Installer which will guide you through the installation of the Licel Applications. Please proceed to the section 2.3.
- Press *Open folder to view files* to start the File Explorer (Windows Explorer) to see the content of the CD:



- The LabVIEW source files are located in the folder *LabVIEW files*. From there you may copy them to a directory of your choice on your local PC.

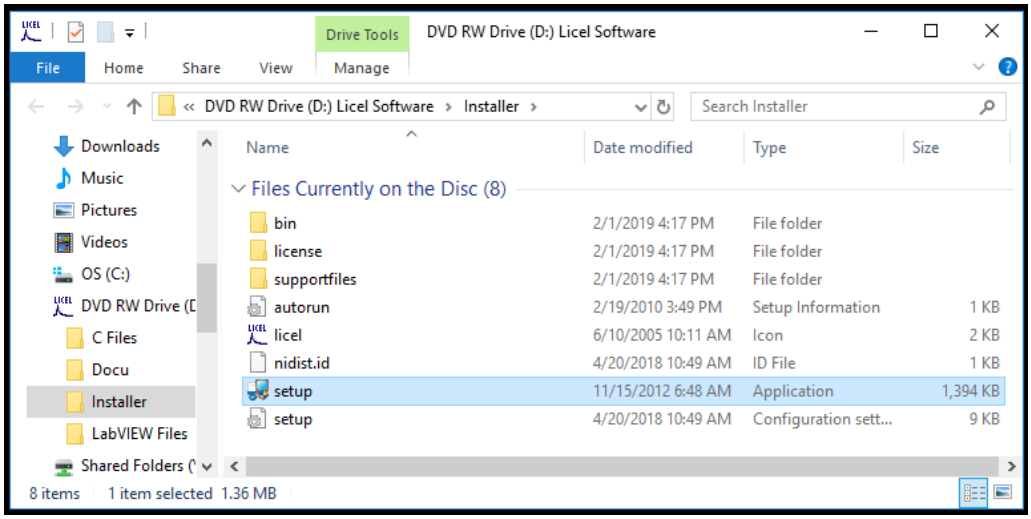


Please note the remarks according to existing LabVIEW library files. Please refer to the section 2.4 for further details.

- In the folder *Docu* you will find some documentation.
- The folder *C Files* contains Licel's C sources.

4. If the selection dialog does not come up automatically after inserting the CD into your CD/DVD drive, please manually open the File Explorer (Windows Explorer) and navigate to the CD/DVD drive of your PC.

   - Either go to the folders *LabVIEW Files*, *Docu*, or *C Files* to get the LabVIEW source files, read the documentation, or copy the C source files,
   - or open the folder *Installer* and run *setup.exe* by double click to start the Windows Installer.



Please proceed to the section 2.3 afterwards.

2.3 Download

The Licel software is frequently maintained. The most recent version is available on the download page (http://www.licel.com/wind.htm). Licel provides both packages described in this chapter, the LabVIEW sources as well as the Windows installer to deploy the Windows applications. The packages come as zipped archive files, `WaveriderLVSource.zip` contains the LabVIEW sources, while `WaveriderLVInstaller.zip` is the corresponding zip archive with the Windows installer. Note that you may have changed these files names while downloading the archives.

Unpacking the Windows Installer

If you downloaded the Windows Installer package (`WaveriderLVInstaller.zip`) please unzip all files to a temporary directory. Locate the setup routine `setup.exe` in that directory and run it by double-clicking the program entry in the Windows Explorer. Please proceed to the section 2.3.

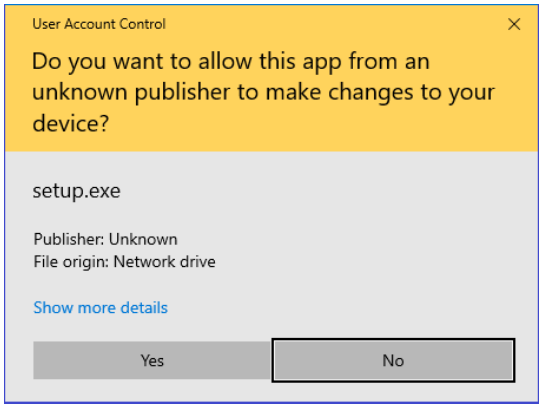
Unpacking the LabVIEW Sources

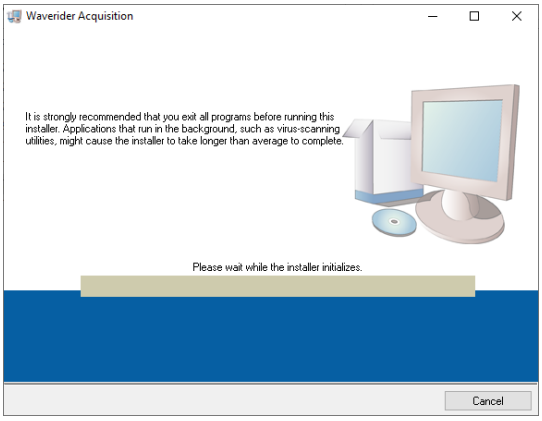
The Licel LabVIEW libraries and initialization files contained in the zip file `WaveriderLVSource.zip` may directly be unzipped to a destination folder of your choice. Please note the remarks according to existing LabVIEW library files. Please refer to the section 2.4 for further details.

2.4 Installing the Windows Applications

This subsection describes the installation process for the Licel Windows applications. To operate the Licel Windows applications a LabVIEW runtime environment needs to be installed, as well. The Windows applications together with the LabVIEW runtime environment come as a Windows Installer package. For the installation of the LabVIEW runtime part of the installer package local administrator privileges are required.
The following items describe the installation process after starting the Windows Installer's setup routine (`setup.exe`). The setup program is automatically started when using the CD ROM and pressing **Install Applications** in the setup selection dialog. `setup.exe` is located on the Licel CD ROM in the subdirectory `Installer` or in the temporary directory you unzipped the downloaded Licel Installer package. You may directly start the setup routine from the corresponding directories.
The Windows Installer dialogs will guide you through the installation process.

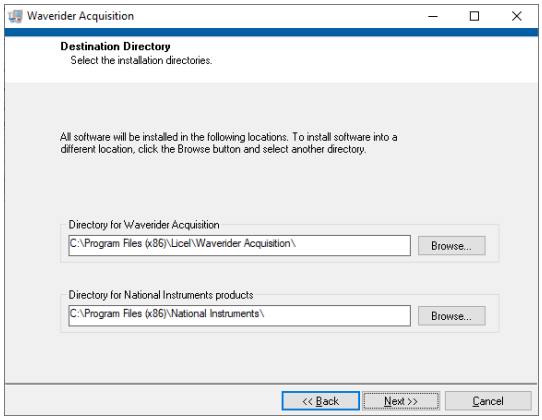
1. At the very first start of the installation the User Account Control dialog may appear. Click **Yes** to continue the installation process.

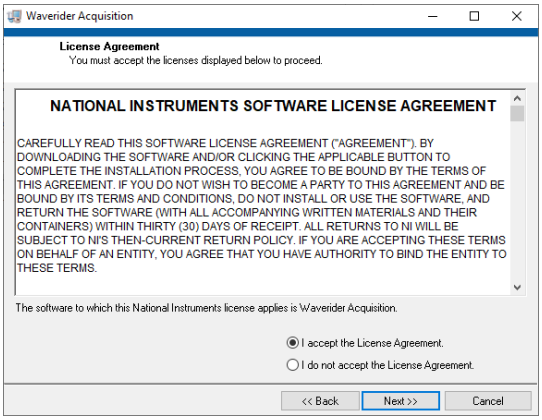




   Afterwards, the Installation will be initialized and the destination directories will be shown.

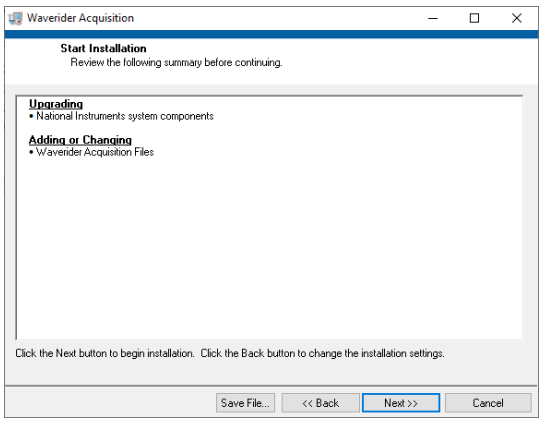
2. You may change the installation directories using the **Browse** button. Click **Next** to proceed.

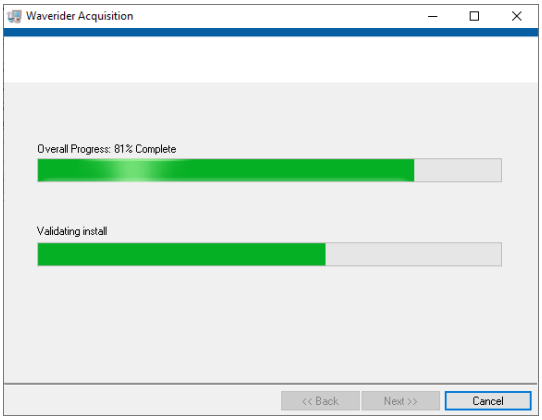




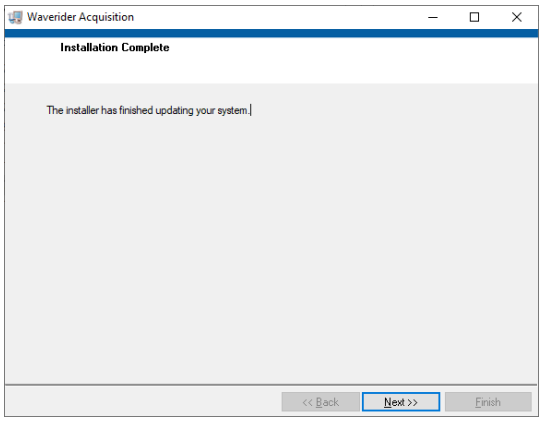
   In the next dialog you have to accept the License Agreement(s). Choose *I accept the License Agreement* and **Next** to proceed.

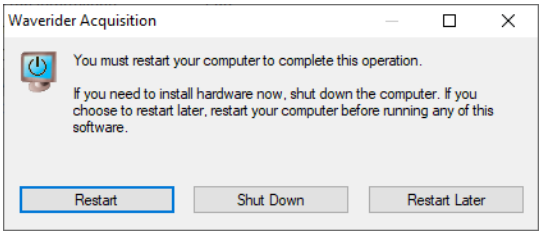
3. Confirm the following dialog using the **Next** button or click **Back** to change your installation settings. After starting the installation the progress will be indicated by a progress bar.





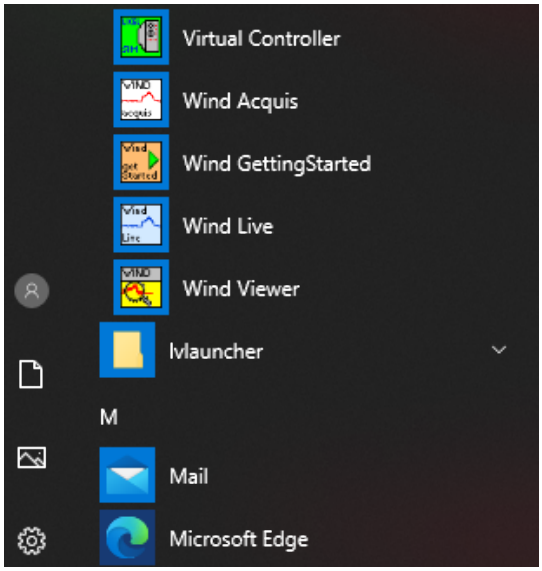
4. After the installation process is completed an information window will be shown. Click **Next** to proceed.





To finish the installation you may have to restart your computer. If a restart is required click **Restart** to complete the installation.

5. After the installation has successfully been completed you are able to start the Windows applications through the corresponding folder **Licel**. To open the folder go to the Windows start menu and browse the Letter L.



6. If you have backuped your initialization files from an older version of Licel Ethernet Software you may copy the TCP/IP parameters from the corresponding old initialization files to the files

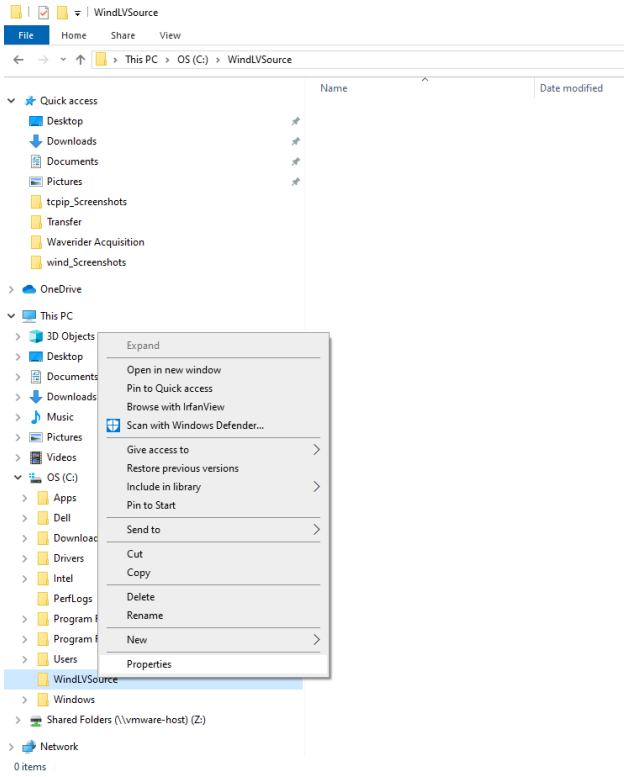
of the current installation. Please note that copying information from older to new initialization files should be done value by value (line by line).

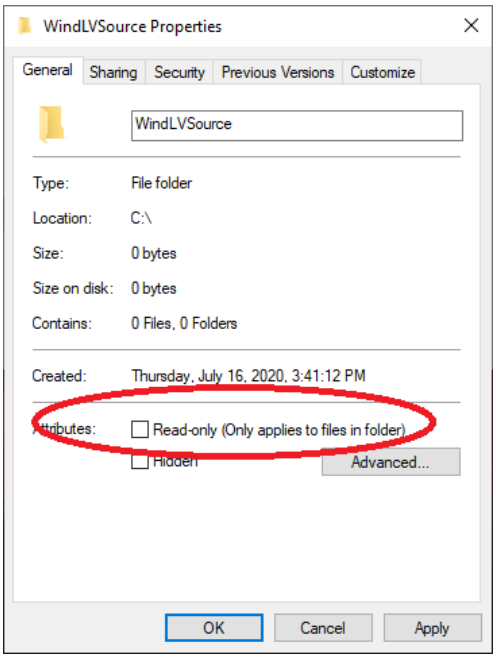
2.5 Installing the Licel LabVIEW Libraries

To install the Licel LabVIEW libraries you may choose between the following options:

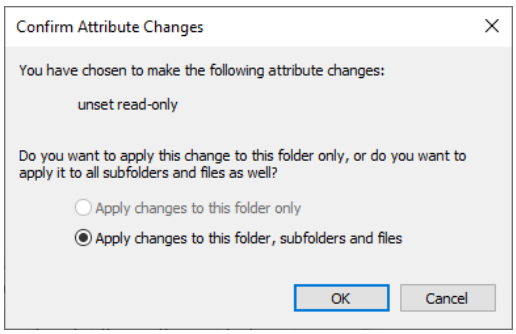
- The Licel LabVIEW Libraries will be copied automatically from the Licel CD ROM by pressing **Copy LabVIEW Source** in the setup selection dialog. You will be asked to select or create a target folder.
- You may manually copy all files contained in the directory `LabVIEW Files` on the CD ROM to a directory of your choice.
- If you downloaded the Licel software from http://www.licel.com/wind.htm please unpack the content from the downloaded zip file and copy it to a directory of your choice.

Please note that in the case the software is copied from a CD you may have to unselect the *"Read-only"* attribute for the destination folder. This is done by selecting the directory and right-clicking on it. Select **Properties** from the context menu.



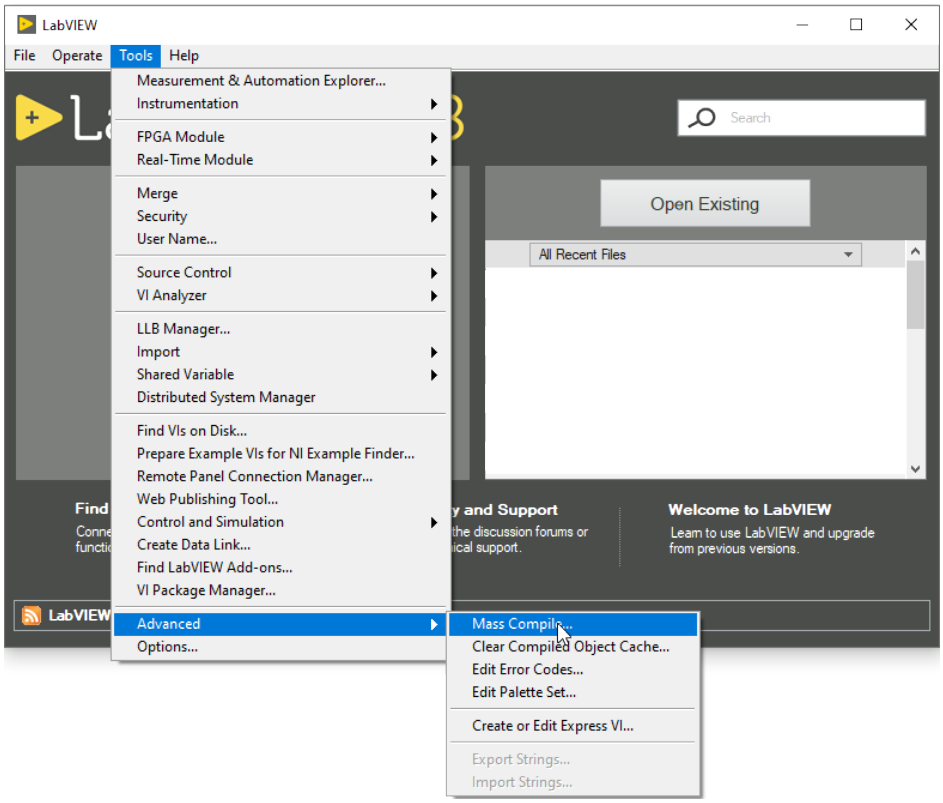


Verify that the `"Read-only"` attribute is not checked, uncheck it if necessary. Click *OK* and check in the next dialog *Apply changes to this folder, subfolders and files*. Leave the dialog by clicking *OK*

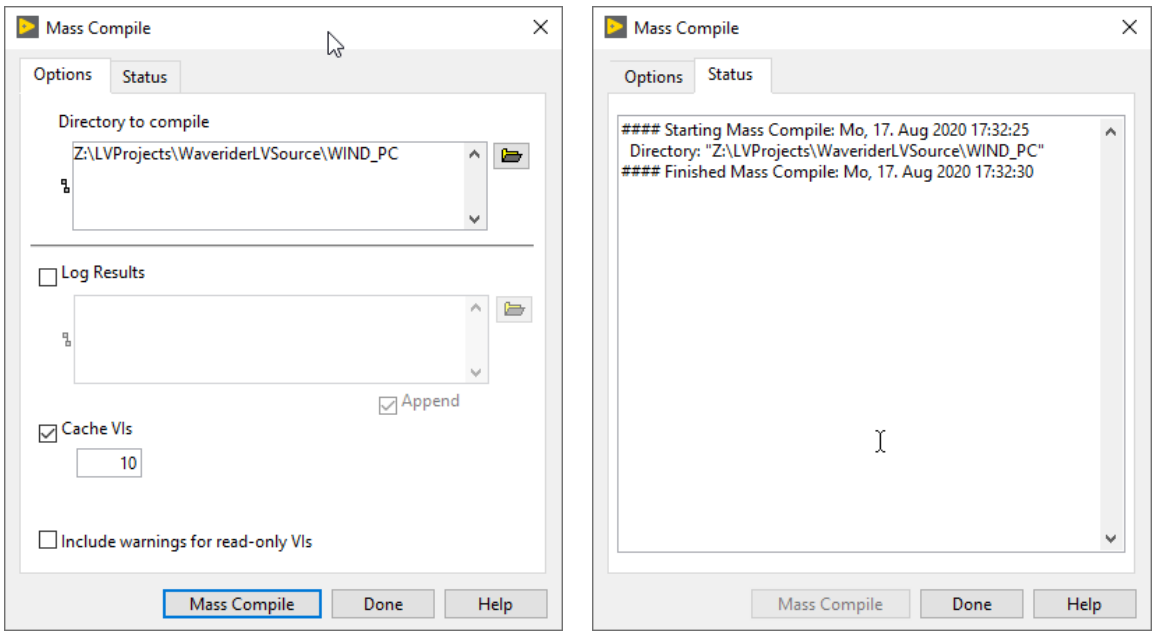


Now you should be able to run all the files. If you are still having problems, apply a mass compile to the directory where the software was extracted to:

1. Start LabVIEW.
2. Select the menu **Tools**, then **Advanced**, and finally **Mass Compile...**.



3. You will be asked to select a directory, select the target directory of the LabVIEW source files.
4. Press *Mass Compile* in the next dialog.



5. Later the mass compile status will be shown.

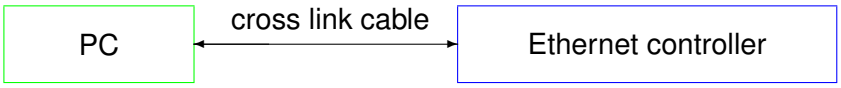
Please note that the removal of older libraries is a necessity, since LabVIEW often links to various libraries with the same name. As a result, if a library is installed twice, one can not be certain which library is actually being used.
If you still have any problems, please contact Licel for further assistance.

Chapter 3

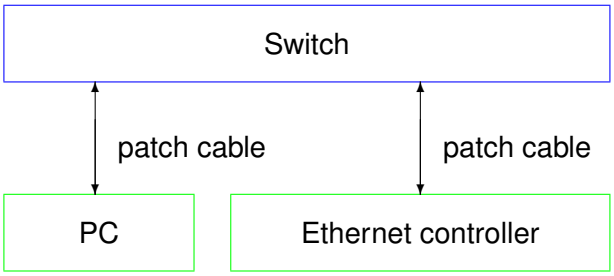
Setting up the Network

3.1 Network Introduction

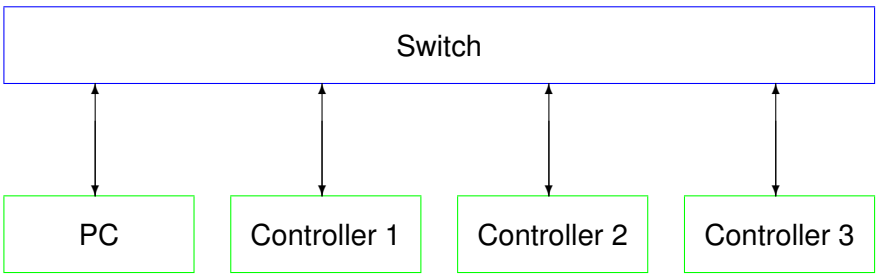
To control a Waverider controller a working TCPIP connection is required. This can be reached by two ways, using a cross link cable, which creates a one to one connection between the PC and the Waverider Controller or with patch cables and a switch



The cross link cable might be a perfect setup for single controller, but as soon as the PC needs to communicate over the same network connector with other nodes locally or the Internet, the usage of a switch is mandatory.



This configuration has the big advantage that it is easily scalable if more than one controller needs to be connected.



There are two concepts for the switch either:

- Use the local infrastructure, this requires coordination with your local network administrator as

she/he will define network addresses to be used for the PC and the Ethernet controllers or require DHCP for the nodes to be used.

- add a second Ethernet controller to the PC, so that Ethernet controllers can be moved to a private network and you become the administrator of this private network.

  http://en.wikipedia.org/wiki/Private_network describes the available address ranges, selecting a network subset in the 192.168.0.0 192.168.255.255. seems like a good choice

In all of these configurations the PC and the controllers should be finally in the same subnet but have **different** IP addresses within this subnet. To achieve this, each controller needs to be specially setup as all controller ship with the same default network address. If more than controller needs be setup the procedure below needs to be repeated for each controller individually. **Never** connect more than one controller with the factory default to a network.

3.2 Preparations

To operate the Waverider controller in your local network you will have to carry out the following required steps described in the corresponding subsections:

1. Get the required **Network Information**.
2. Prepare the PC to communicate with the Ethernet controller using a cross-link cable (**Network Preparation**).
3. Setup the Ethernet controller for your local area network either by setting a fixed IP address or by activating the DHCP mode (**Network Setup**).
4. **Reconfigure the PC** for your local area network and test the communication with the Ethernet controller.

3.3 Network Information

The Waverider controller is shipped with dynamically assigned IP address (DHCP).
The network parameters should be aligned according to your local network environment. Before doing this, the system administrator should be contacted. He should provide the following information:

1. Should the Ethernet controller use a dynamically assigned IP address (DHCP)?
   (a) If yes, the network parameters will be set by a DHCP server residing in your LAN. Refer to the subsection DHCP Mode (3.5.2) to enable the Waverider controller to automatically receive the network parameters from the DHCP server.
   (b) If a static address configuration is to be used,
      i. the IP address,
      ii. the network mask,
      iii. and the gateway

      should be set by yourself. Refer to the subsection Fixed IP Address (3.5.1).
2. The default ports used by the Ethernet controller are 2055, 2056 and 2057. Can these ports be used?
3. Is it necessary to change the configuration of any firewall in the case you need to access the controller outside of the LAN boundaries?

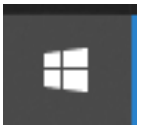
3.4 Network Preparation

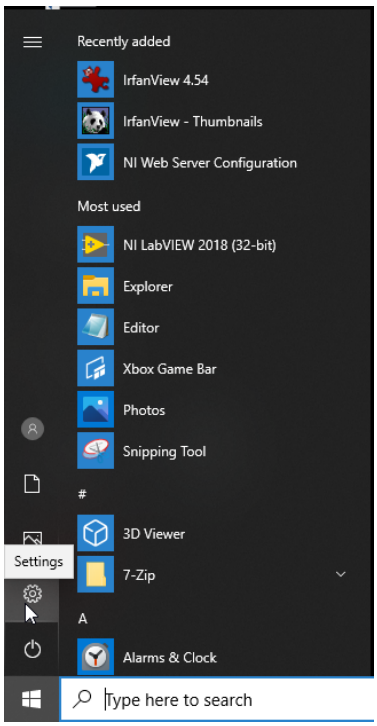
After having installed the Licel Windows applications or the Licel LabVIEW modules on your PC you are ready to change the network configuration parameters of the Waverider controller according to the local network settings described in the previous section.

3.4.1 Establish the Connection

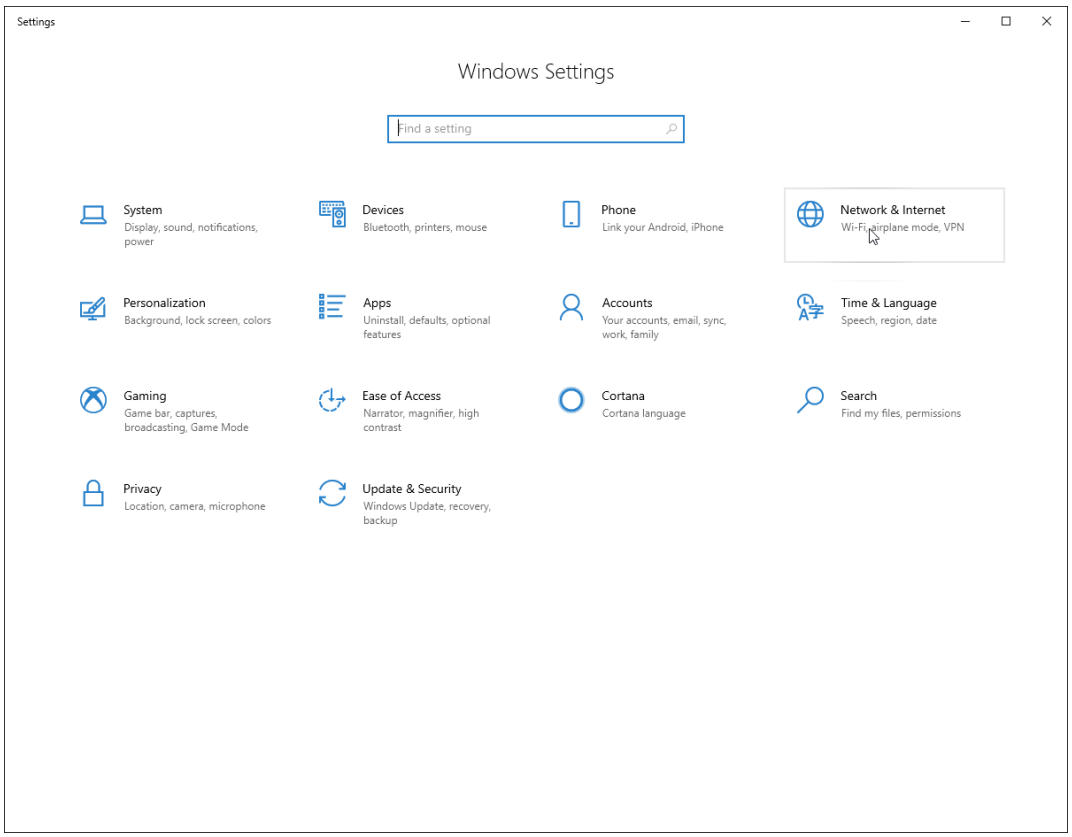
A straight-forward way to do this is the following procedure. You will need local administrator rights on your PC for the following steps:

1. Disconnect the PC from the local network.
2. Open the *Properties* dialog of the network connection your Ethernet adapter is assigned to. Usually you will find the appropriate network connection by opening *Network Connections* from the Windows start menu or the *System Settings*. The following list shows the steps to follow on a Windows 10 operating system:

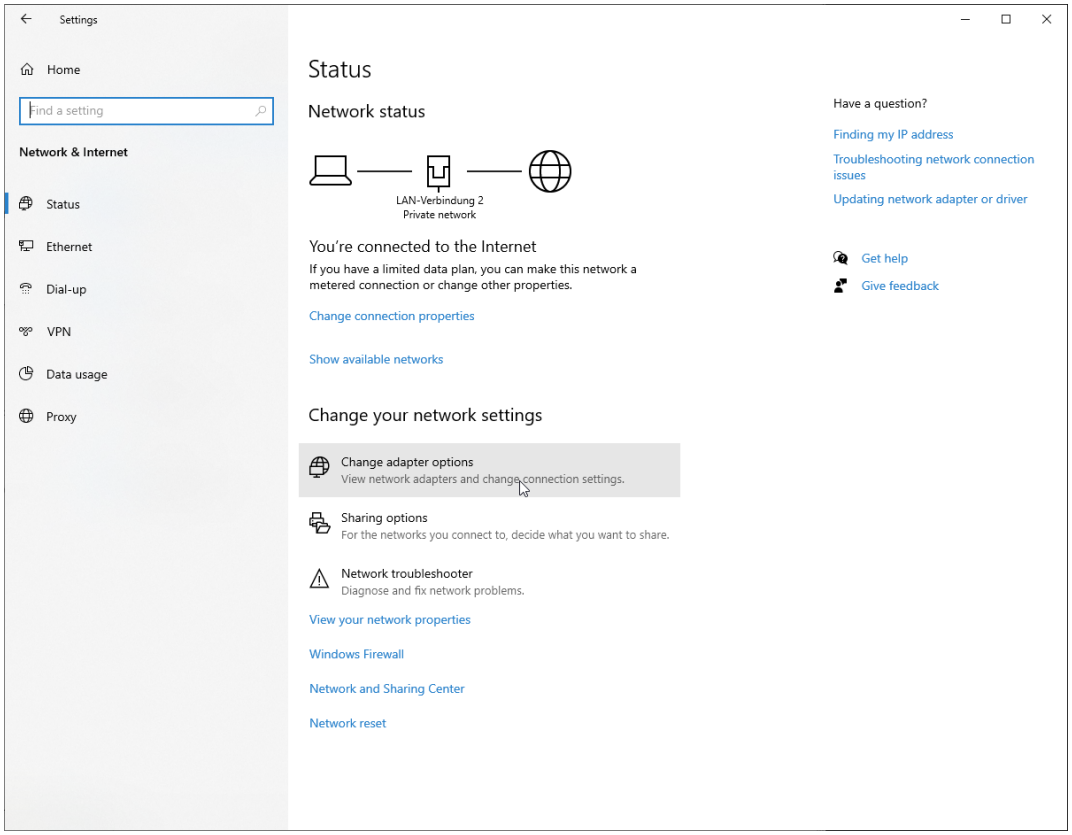
   (a) Click on the  button, and then on *Settings*.



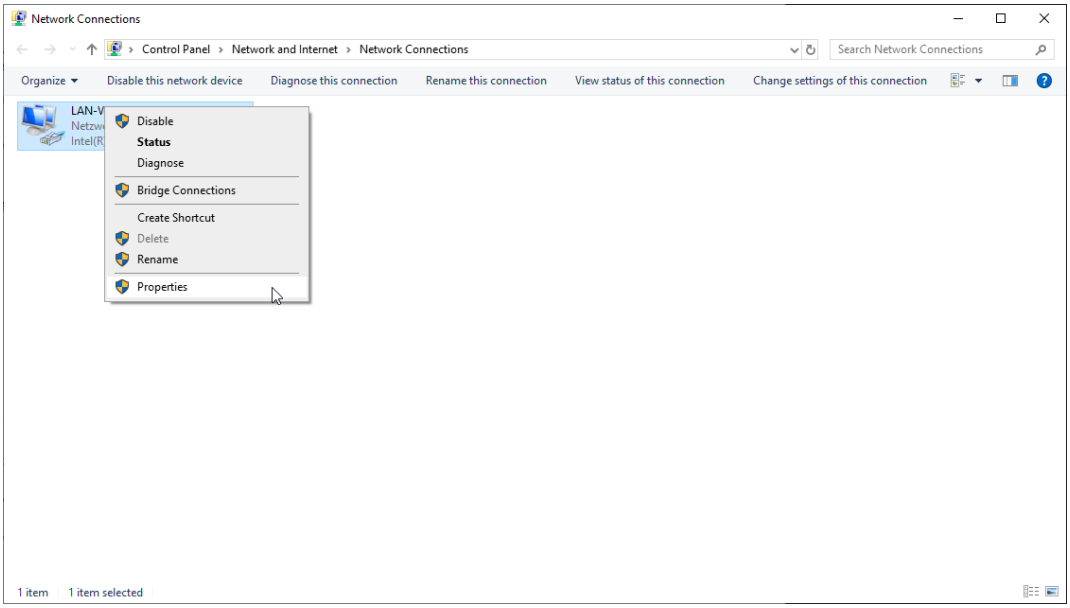
   (b) Once the control panel has come up click on *Network and Internet*.



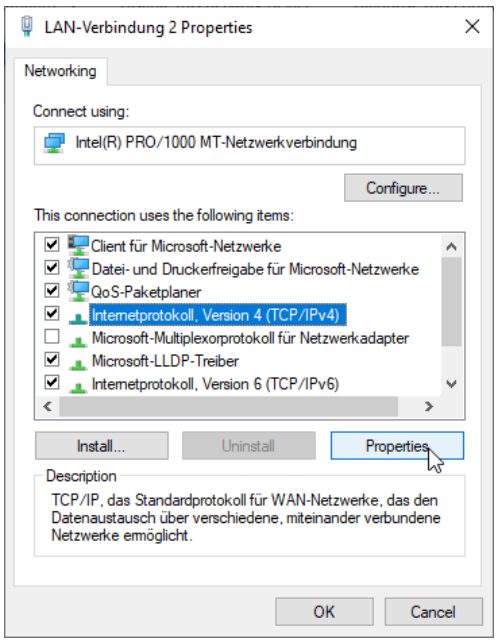
(c) In the next window click on *Change Adapter Options*.



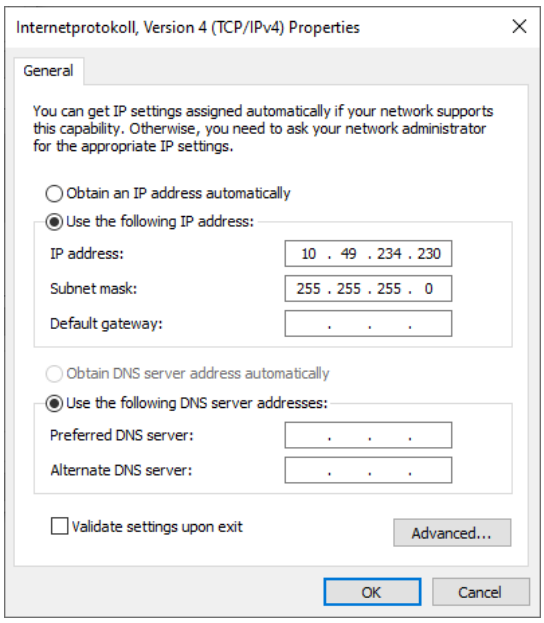
(d) The installed network connections will be shown, right-click on the local Ethernet connection to be used with the Waverider controller and choose *Properties* from the context menu.



3. Click on the TCP/IP protocol entry in the lists of components used by the assigned Ethernet adapter card / LAN connection and press the *Properties* button.



4. Write down your current TCP/IP settings i.e. all settings seen in the following graphics. You will need this information to reconfigure your PC to access the LAN again.



5. If activated disable DHCP (checkbox *Obtain an IP address automatically*) and manually assign an IP address within the default address range of the Waverider controller. A good choice would be `10.49.234.230`. **Never use the default address (`10.49.234.234`) of the Waverider controller as IP address for your PC.**
6. Quit the dialog by pressing *OK*.
7. Reboot your PC.
8. Power up the rack with the Waverider controller and connect the PC with the controller using a **cross-link cable** shipped together with your hardware.

Now you should be able to access the Waverider controller via your Ethernet card. Please test this first connection with the methods given in the next section.

3.4.2 Diagnostics

Please carry out the following steps to verify that the connection of the Waverider controller with the PC is established.

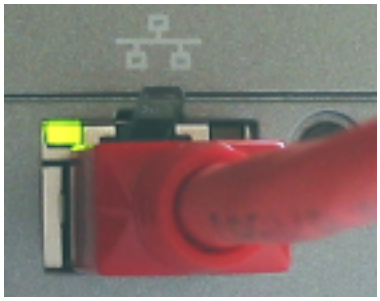
1. Verify that the green **LNK** LED lights up indicating a correct electrical connection.
2. Verify that in case of a 100Mbit Ethernet connection the **Spd** lights up.
3. Verify that the network settings of your PC have changed according to your settings:
   (a) Open a command prompt window (DOS box).
   (b) Type `ipconfig` and press enter. At least one of the Ethernet adapters should show the address that you previously set (`10.49.234.230`). The response should be similar to the following:

```
1 Ethernet Adapter :
        IP-Address. . . . . . . . . : 10.49.234.230
        Subnet Mask . . . . . . . . : 255.255.255.0
        Standard-Gateway. . . . . . :
```

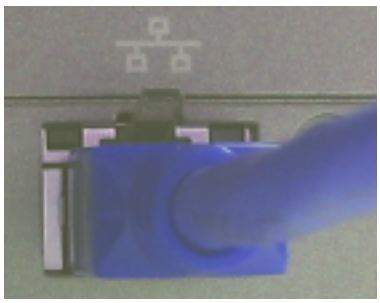
4. Verify that the Waverider controller is accessible via the network now:

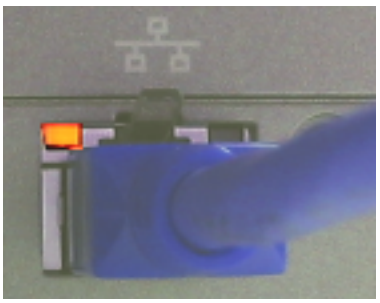
(a) Open a command prompt window (DOS box) or use the one from above.

(b) Type `ping 10.49.234.234` and press enter. The Licel Ethernet Controller should respond without loss of any packet. If the controller is not responding check if the network cable is correctly mounted and that an appropriate cable is used, i.e. a cross-link cable when working with a direct connection from the computer. Most Ethernet adapters indicate a correct connection with a green LED:



A non-existent or incorrect connection is often identified by an unlighted LED (left) or red LED (right).





Please note that these indicators may be different on your PC.

(c) If the network cable connection is correct and the controller is still not responding execute a hardware reset and repeat the procedure with the default IP address.

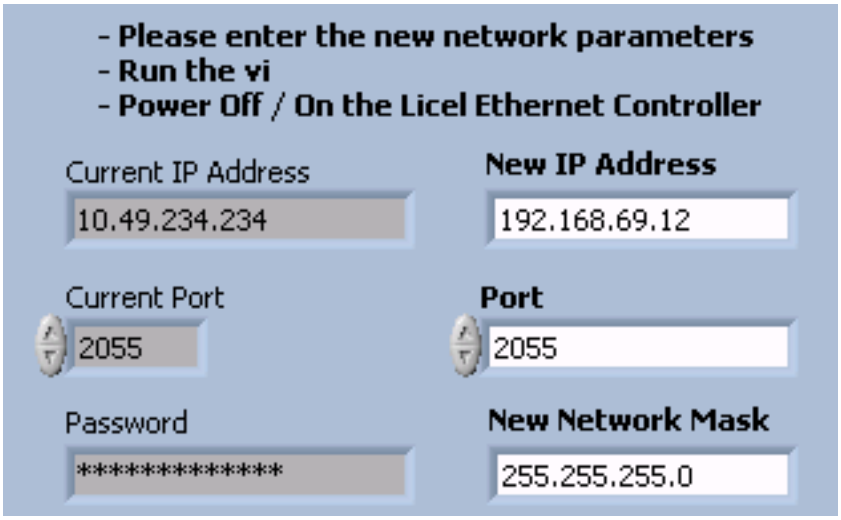
3.5 Network Setup

In order to configure the Ethernet controller, you need either to set the controller to a fixed IP address or invoke the DHCP Mode. Whether a fixed or dynamic (DHCP) mode is used or not will depend upon your network type. Dependent on this, please refer either to the subsection Fixed IP Address or DHCP Mode and skip the corresponding other subsection. Please contact your administrator if you have not yet requested the information described in the above subsection Network Setup.
Afterwards you will have to reconfigure your PC for operating in the local network.
Once you have set the **IP Address** and **Port** for the Licel Ethernet Controller you should define these values to be used by the software.

3.5.1 Fixed IP Address

If you need to set the controller to a fixed IP address carry out the following steps. Skip the steps described in next subsection DHCP Mode.

1. Open `Licel TCPIP Set New Fixed IP Address.vi` or the corresponding Windows application from the Windows start menu.

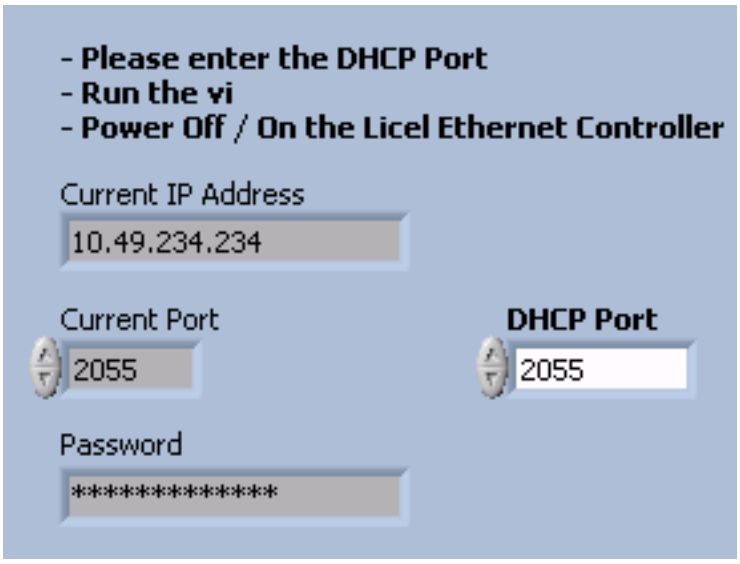


2. Set the desired network parameters in the fields **New IP Address**, **Port**, and **New Network Mask**.
3. Do not forget to enter the correct administrator **Password**.
4. Run the vi by pressing the start button. It should finish without opening an error message dialog.
5. Turn the Waverider controller off and switch it on again. Wait **approximately 20 – 30 seconds**.
6. A `ping 10.49.234.234` executed from a command prompt (DOS box) should now time-out.

3.5.2 DHCP Mode

In order to configure the Waverider controller for DHCP operation carry out the following steps. You must have skipped the steps described in the last subsection Fixed IP Address.

1. Open `Licel TCPIP Activate DHCP Mode.vi` or the corresponding Windows application from the Windows start menu.



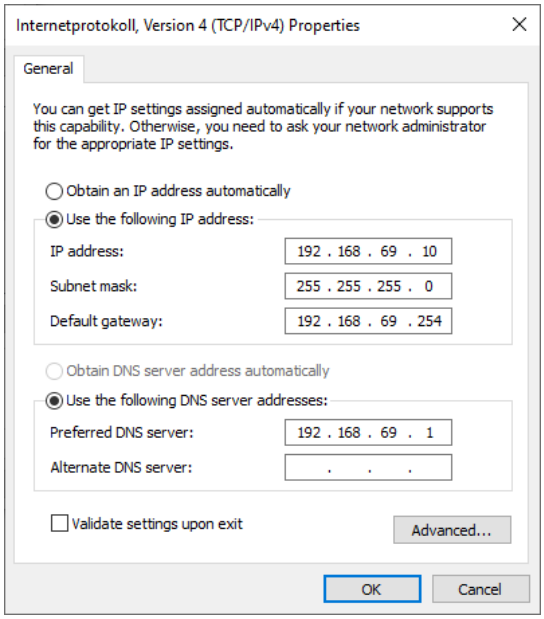
2. Set the desired **DHCP Port** number.
3. Do not forget to enter the administrator administrator **Password**.

4. Run the vi by pressing the start  button. It should finish without opening an error message dialog.
5. Turn the Waverider controller off and switch it on again. Wait **approximately 20 – 30 seconds**.
6. A `ping 10.49.234.234` executed from a command prompt (DOS box) should now time-out.

3.6 Reconfigure the PC

After you successfully configured the Waverider controller the following last steps have to be carried out to reconfigure your PC for the local network and to test the connection to the Waverider controller:

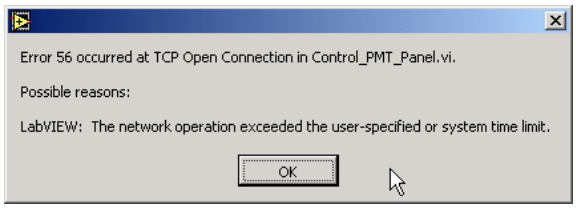
1. Reconnect the PC to the local network.
2. Open the *Properties* dialog of the network connection your Ethernet adapter is assigned to. A more detailed instruction has been given above.
3. Open the *Properties* dialog of the TCP/IP protocol entry in the lists of components used by the assigned Ethernet adapter card.
4. Reset your current TCP/IP settings to the values you recorded while processing the subsection to establish a network connection.



   Note that the values shown here are just example settings. You must exactly use the settings present on your PC before configuring the Waverider controller.

5. Quit the dialog by pressing *OK*.
6. Reboot your PC.
7. Connect the Waverider controller with your local network through a hub or switch using an **ordinary patch cable**.
8. Execute a `ping` command from a command prompt (DOS box). Use the IP address you assigned to the Waverider controller. If the Ethernet controller is in DHCP mode try first `ping LicelWind` if this does not return value, you need to ask your system administrator for the assigned network address. The `ping` command's response should indicate a correctly working connection.

9. Test the access using `Licel Wind TCP.vi`.
10. A TCP/IP timeout error with LabVIEW's error code 56 may be caused by a wrong IP address.



Please check carefully that the values for **IP Address** and **Port** match with the parameters set at the Waverider controller. Set the correct values as defaults for future operation. Other reasons for errors with code 56 are non-existing connections (check if the cable in use is correct) or unstable network operation.

3.7 TCP/IP Connection Parameters

To work properly with the Waverider controller both the Windows applications and the LabVIEW software must be able to establish a TCP/IP connection. The user of the software must define the **IP Address** and **Port** – these values must be equal to the parameters that have been for the Waverider controller following the network setup section.

Windows Applications: Initialization Files

The Windows applications are communicating with the `WindTCP_Server.vi` in the background over a queue interface. This `WindTCP_Server.vi` is communicating with the Waverider controller and uses a initialization file to read the TCP/IP parameters **IP Address** and **Port**.

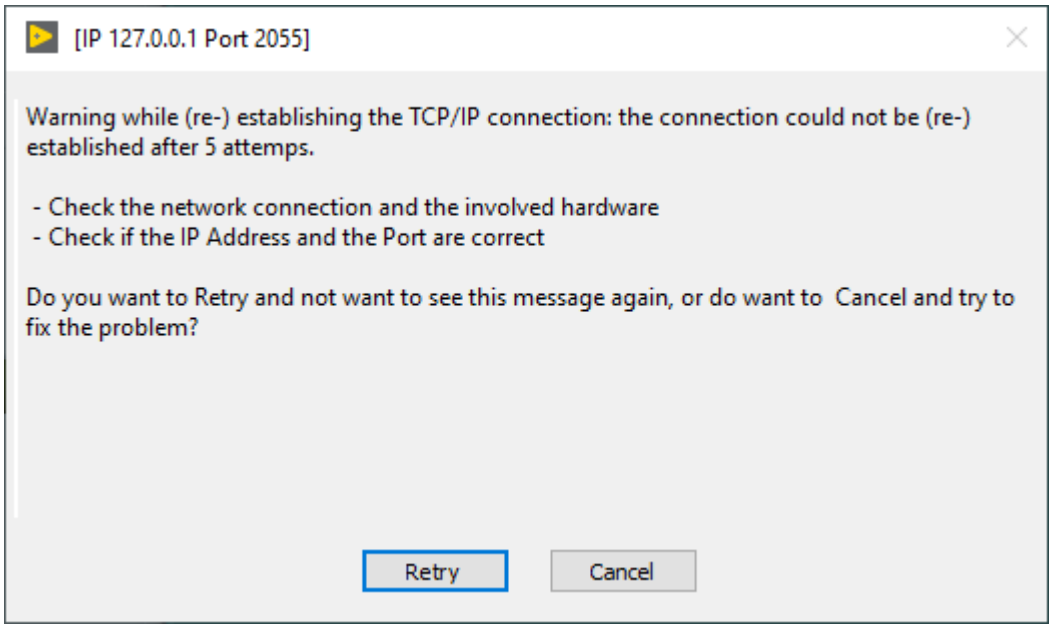
For Windows applcations this file is located at: `C:\Program Files (x86)\Licel\Waverider Acquisition\WIND.ini`.
An example for an initialization file holding the TCP/IP information is given below:

```
[Client]
IP = 10.49.234.234
PORT = 2055
```

You may edit this file using a text editor like `Notepad` which is installed by default when setting up a Windows operating system. You may use `Notepad` as well to create a required initialization file if it does not exist in the installation directory. Make sure that you save the file before leaving the editor. You must change the values for IP address and port to the values you will set following the Instructions in the network setup section.
This file will be also used by the other applications to store information.
If the network address is not correct a error dialog will show up

3.7.1 TCP/IP Connection Problems (Software)

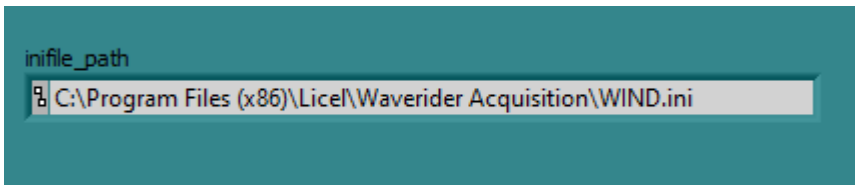
The `WindTCP_Server.vi` has a built-in mechanism to re-establish the TCP/IP connection to the Waverider controller when the connection is lost or when the connection is not successful after the program start.
If the reconnection mechanism is not successful after 5 attempts the software assumes that some basic TCP/IP settings may be incorrect. Therefore the following error message is displayed:



In the case that this dialog comes up please

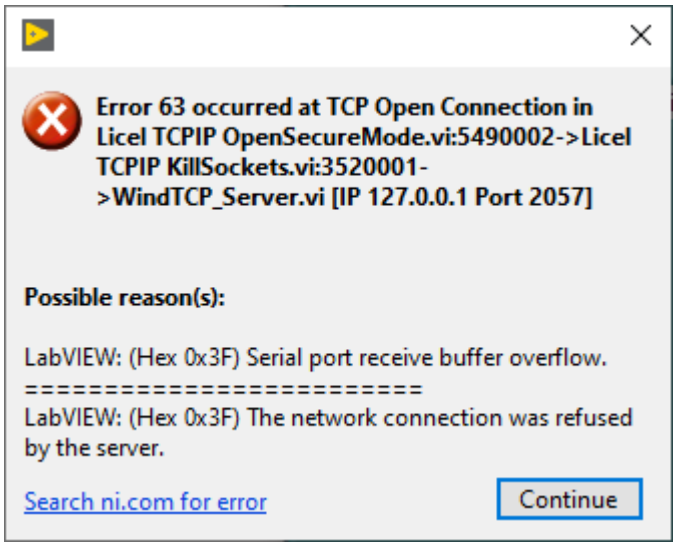
- check the network connection and the involved hardware. Check whether the Waverider Controller is switched on. Check that the Ethernet cable is plugged correctly, and that the correct Ethernet cable is used.
- check whether the **IP Address** and the **Port** the software is using equal to the values of the Waverider-Controller (refer to the network setup).

  1. Before you start please enter the correct values for the **IP Address** and **Port**. You should already have set these values for the Waverider Controller
     – You should set the values in the corresponding initialization file. You will see the full path of the file in a file path indicator on the *System* page.



You have two choices to leave the message dialog:

1. Click *Retry* to continue to reconnect to the Waverider controller.
2. Click *Cancel* to exit. The program will display an error message



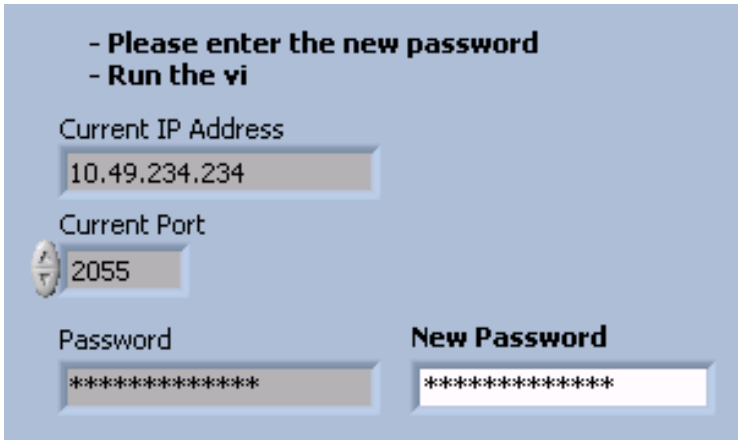
3.8 Network Security

Certain administrative tasks use an administrator password. An example is the change of the IP address of the controller. The administrator password has to be sent with the related commands.

3.8.1 Changing the Administrator Password

The Waverider controller is shipped with the default administrator password "'*Administrator*"'. In order to change this password which grants administrative access to the controller, please carry out the following steps:

1. Open `Licel TCPIP Set New Password.vi` or start the corresponding Windows application from the Windows start menu.



2. Enter the current administrator **Password**.
3. Enter the **New Password**.
4. Run the vi by pressing the start  button. It should finish without opening an error message dialog. Please note that the password is case sensitive.

Chapter 4

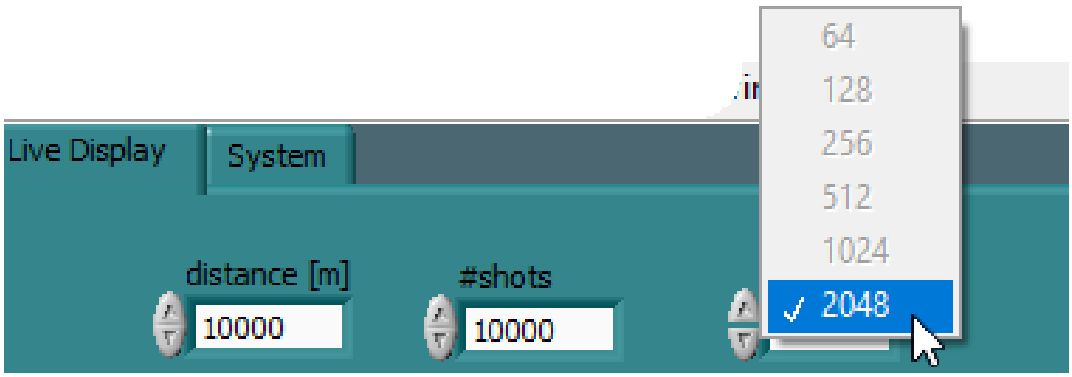
Waverider Software Tutorial

4.1 Overview

This software tutorial describes how to use the data acquisition software as well as the functions of the individual controls and indicators. In order to actually try the information in this tutorial, the hardware and network setup must be completed. This tutorial is broken into two parts. In the Live Display section a short introduction to viewing actual spectra is given. Then Acquisition with Wind Acquis contains instructions for recording your first spectra using Wind Acquis. Which is then followed by the Viewer to display the previously recorded data.

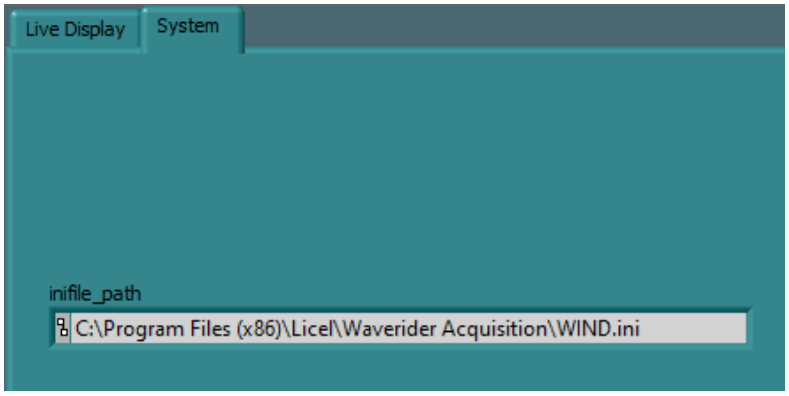
4.2 LiveDisplay

The intention of this module is to give you a tool to verify that the trigger works and a valid signal is connected to the input. Once properly configured and connected it will show for every run the peak amplitude and peak frequency of the power spectra in Vrms2 units versus height. You need to setup first the controller IP and the start settings via the .ini File. For executables this is: `C:\Program Files (x86)\ Licel\Waverider Acquisition\WIND.ini`. For the LabVIEW Sources the ini file is in the WindTCP_Server.llb `D:\Waverider\WindTCP_Server.llb \WIND.ini`. The software recognizes which controller is connected and automatically selects the possible FFT size and writes the change to the ini file. The not allowed FFT size options are disabled and greyed out.

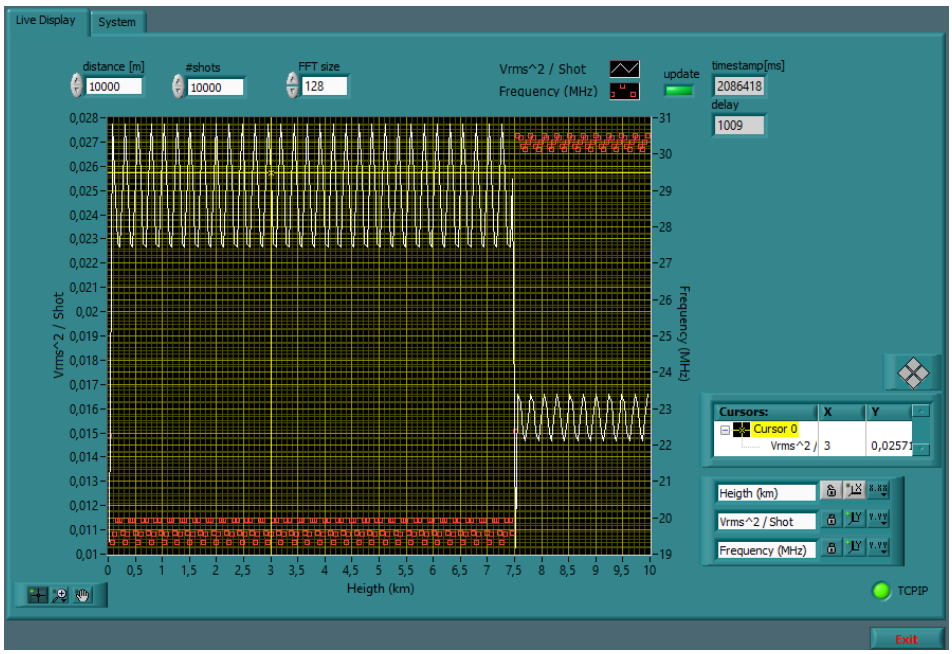


```
[Client]
IP = 10.49.234.234
PORT = 2055

[global_info]
Location = "Berlin"
Longitude = 13.384373
Latitude = 52.542185
Height_asl = 45.000000
working_directory = "/C/temp"
Zenith = 0.000000
Azimuth = 0.000000

[Live]
Distance = 10000
Shots = 10000
FFTsize = 5 ;;  0 = 64, 1 = 128, 2 = 256, 5 = 2048
```

Start settings are: Distance (in meter) Shots and FFTsize. The ini file path can be checked on the `System` tab of the LiveDisplay application.



The measurement starts as soon as the application is started and the TCPIP connection is established. The TCPIP connection is established by the WindTCP_Server. The exchange between server and application takes place through a queue mechanism refer to Queue Programming Interface.



This module will give you a live signal for every run. The peak powers and the range of a run are plotted against the frequency in the graph. You can change the parameters while the acquisition is running, the changes will take effect right away.

4.2.1 Stop the Live Display

The execution of the `Wind Live Display` can be stopped using the button *Exit* or by clicking the window's close symbol.

4.3 Acquis

An acquisition program is provided as part of the delivered LabVIEW source project (*WindAcquis.vi*, found in the appropriate project folder) as well as a Windows executable *Wind Acquis.exe*.This program acquires and stores the acquired data in a `TDMS` file.

4.3.1 Running Wind Acquis

Before running the acquisition program, the initialization file `WIND.ini` must be checked for the correct TCP/IP connection settings. The `WIND.ini` file can be found at the installation path like `C:\Program Files\Licel\Waverider Acquisition\`, if the waverider application was installed. Otherwise (sources only) the `WIND.ini` file is stored at `WIND_PC\WindTCP_Server.llb\WIND.ini`. Please refer to the Network Setup to correctly configure the Waverider for the TCP/IP operation. The `IP` address and the `PORT` assigned to the controller during the network setup must be written to `WIND.ini`:

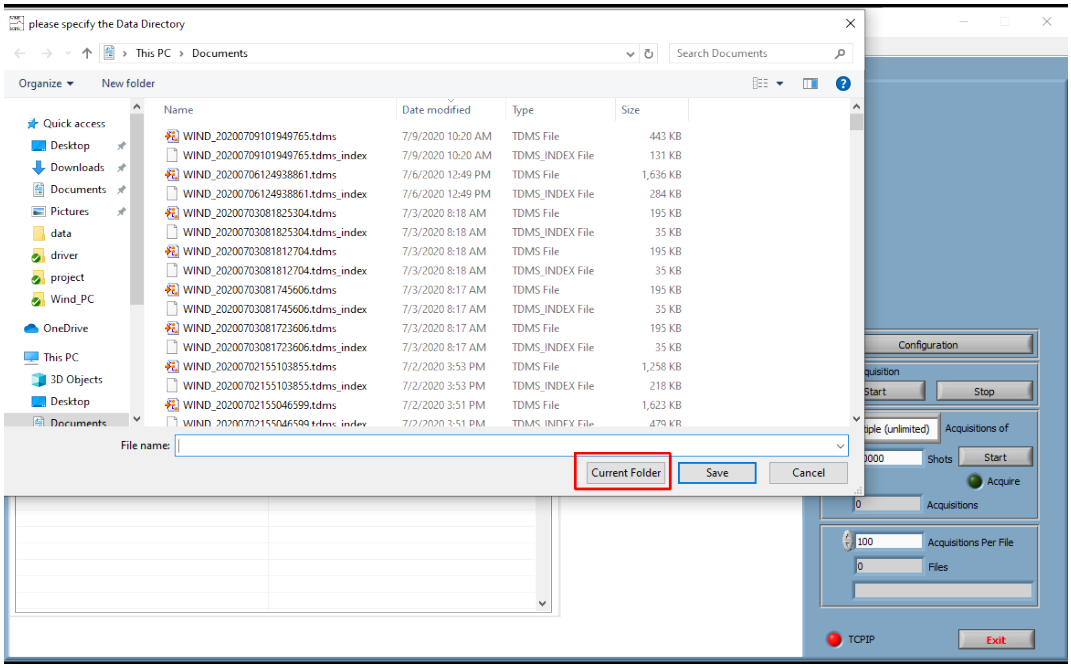
```
[Client]
IP = 10.49.234.234
PORT = 2055
```

Once these values are correct the *Wind Acquis* can be started.

4.3.2 Checking the Data Directory

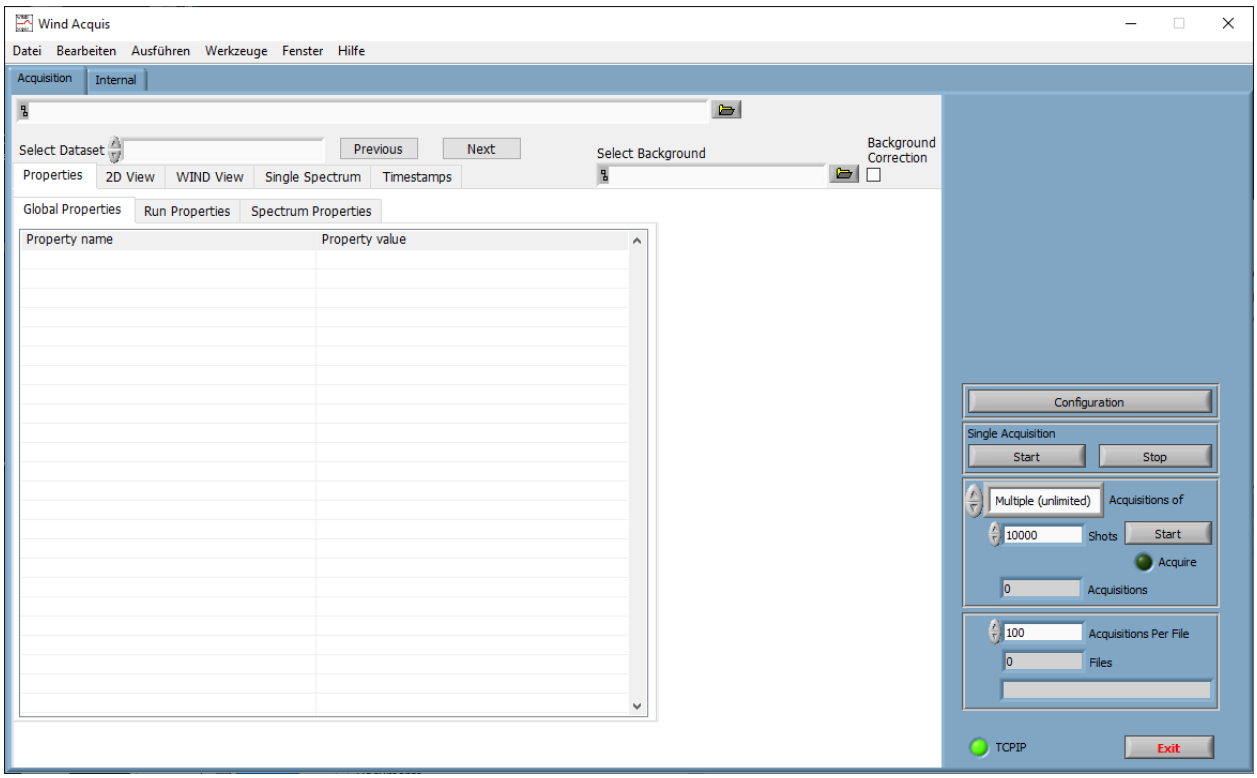
The data directory is automatically checked at the start of the programs. If the formerly used directory is not valid a directory selection dialog comes up directly after starting *WindAcquis*. There, it is possible to select or create a directory to use to save the acquired data to TDMS files using the button *Current Folder*

Current Folder.



4.3.3 Wind Acquis Front Panel

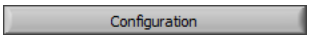
The front panel of *Wind Acquis* consists of mainly two parts: a *view area* on the left and an *acquisition control area* on the right.



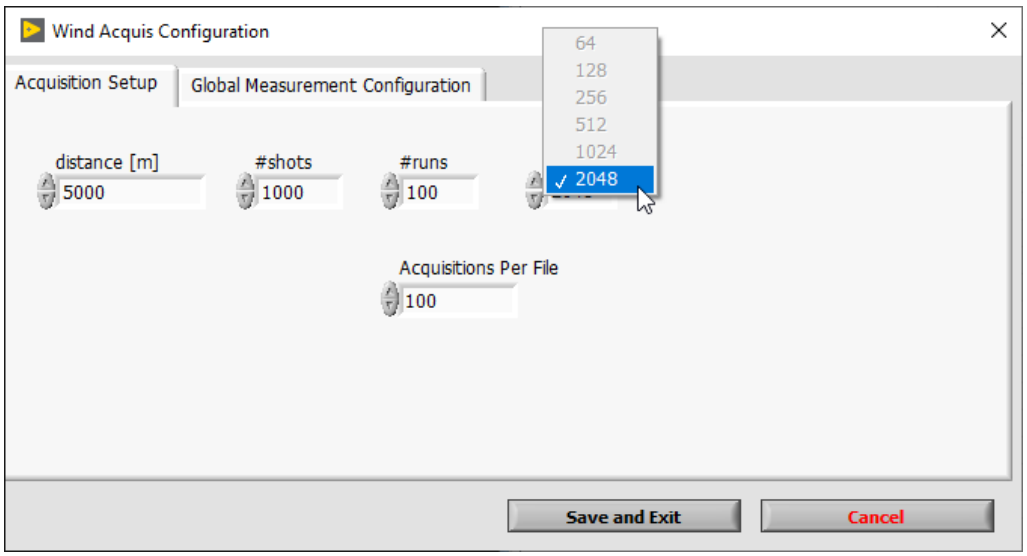
At the most bottom right a TCP/IP indicator and the button *Exit* are located.

- The LED should always light green during operation. If it changes the color to red the TCP/IP connection has been lost. In the case that this is caused by incorrect entries in the initialization file `WIND.ini` or if build-in TCP/IP connection repair mechanisms fail an error message giving more details will be shown and the program terminates.
- The *Exit* button is used to exit the program.

4.3.4 Configuration

In the *acquisition control area* the button *Configuration* 

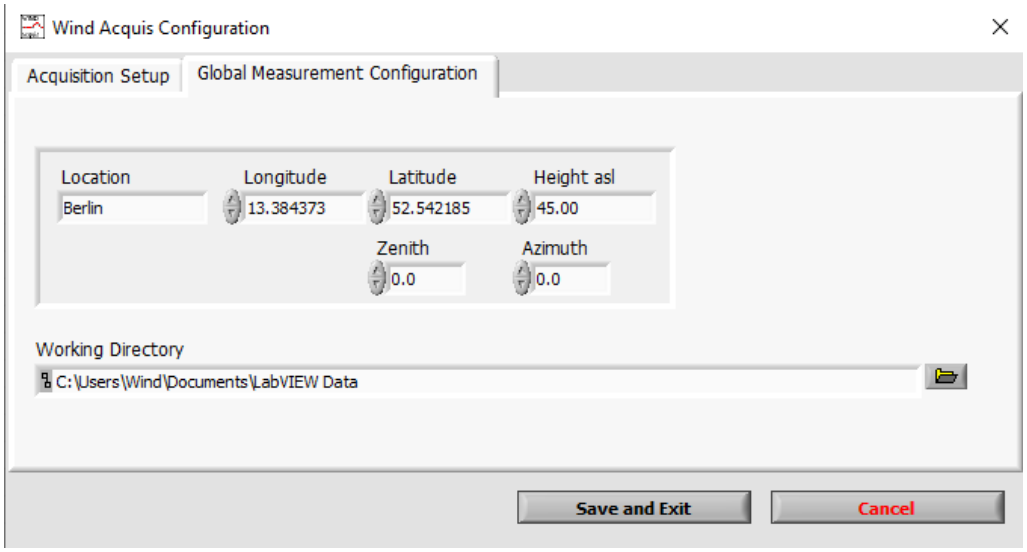
will open the acquisition and measurement site configuration dialog. The values displayed in the configuration dialog correspond to the current values in use. The first tabulator page *Acquisition Setup* contains the controls to specify the acquisition settings: The software recognizes which controller is connected and automatically selects the possible FFT size and writes the change to the ini file.



The following parameters can be entered:

distance[m]	the distance (range) of the acquisition in meters,
#shots	the number of shots to acquire,
#runs	the number of runs to acquire (-1 is allowed indication *run until Stop is manually pressed*),
FFT size	the FFT size,
Acquisitions Per File	the (maximum) number of acquisitions per file (-1 = unlimited, not recommended).

The second tabulator page *Global Measurement Configuration* allows to enter measurement site data and the detection angles.



Location	the name of the measurement site / location,
Longitude	the corresponding longitude in degrees,
Latitude	and latitude in degrees,
Hight_asl	the height above sea level in meters,
Zenith	the zenith angle,
Azimuth	and the azimuth angle,

Working Directory the current data directory for the TDMS data files.

To check out of the configuration dialog two options are available:

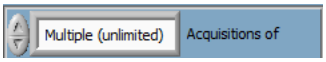
Save and Exit save all parameters to the initialization file for the next program start and exit the configuration dialog,

Cancel reset the parameters to the values when the configuration dialog has been opened.

4.3.5 On-the-Fly Change of Parameters

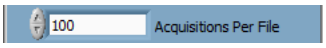
Some of the parameters can be changed on-the-fly without saving them to the initialization file. Nevertheless the values will be displayed when opening the configuration dialog!

Number of Acquisitions the number of runs (acquisitions) is either -1 (*unlimited* = run until Stop) or a finite positive Number. The *unlimited* setting can be achieved by setting the selector to *Multiple (unlimited)*

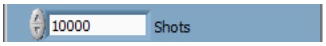


or by explicitly setting the run number to -1:

Acquisitions Per File the *Acquisitions Per File* (-1 = unlimited, not recommended) can be changed at the bottom



Number of Shots the *Number of Shots* can directly be entered, as well:

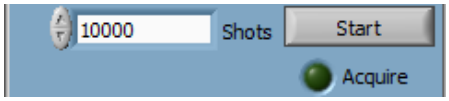


4.3.6 Run an Acquisition

A single acquisition (acquisition series with only one run (acquisition)) is started (and stopped) with the appropriate buttons.

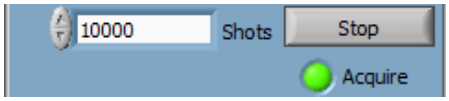
.

A series of limited or unlimited acquisitions (runs) is started using the *Start* button below:



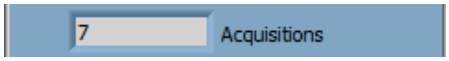
.

Once a series of acquisitions is running the corresponding LED *Acquire* changes the color to bright green. A *Stop* button is available.

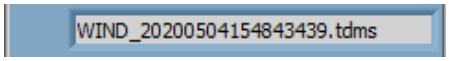


.

The run (acquisition) counter increases . . .

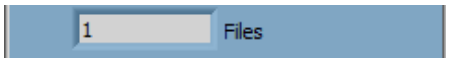


. . . and a file is created and the file name is displayed:



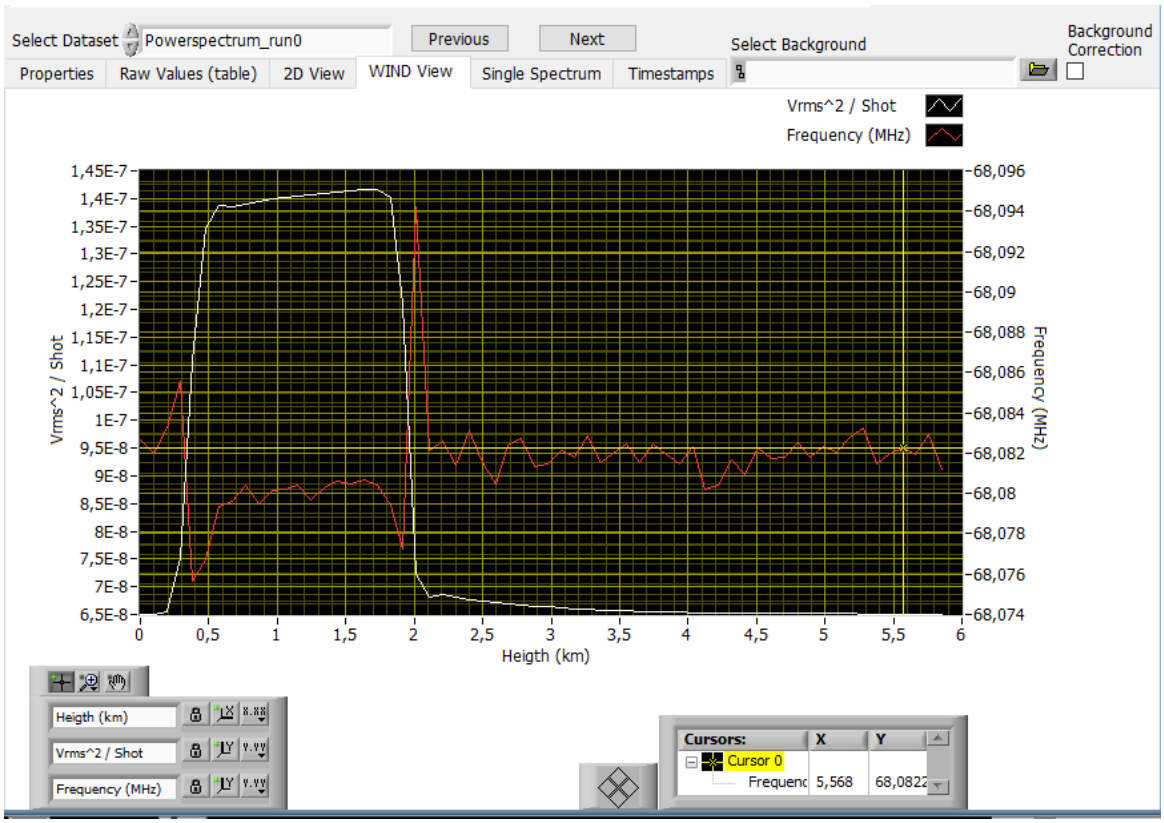
.

Acquired data is displayed on the left side in the *view area*. There, during an acquisition, the file controls are locked. Once the *Acquisitions Per Files* has been reached a new file will be generated. The file counter increases:



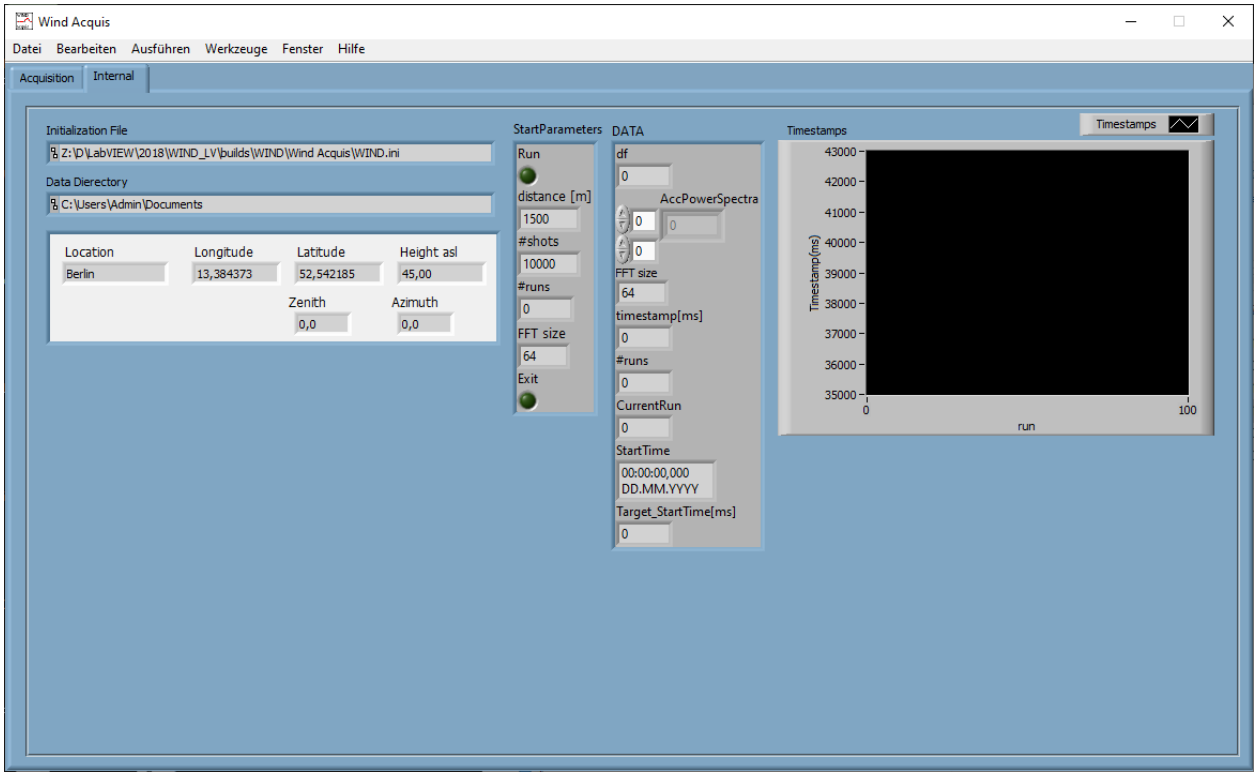
.

The *View Area* on the left side of the *Wind Acquis* front panel contains the Wind Viewer as a plug-in. When no acquisition is running it can be used to display and inspect any acquired wind data file as described in the next section. During a running series of acquisitions the *View Area* is automatically filled and shares the current acquired file with *Wind Acquis*.



.

On the tabulator page *Internal* some more internal and service information is available:



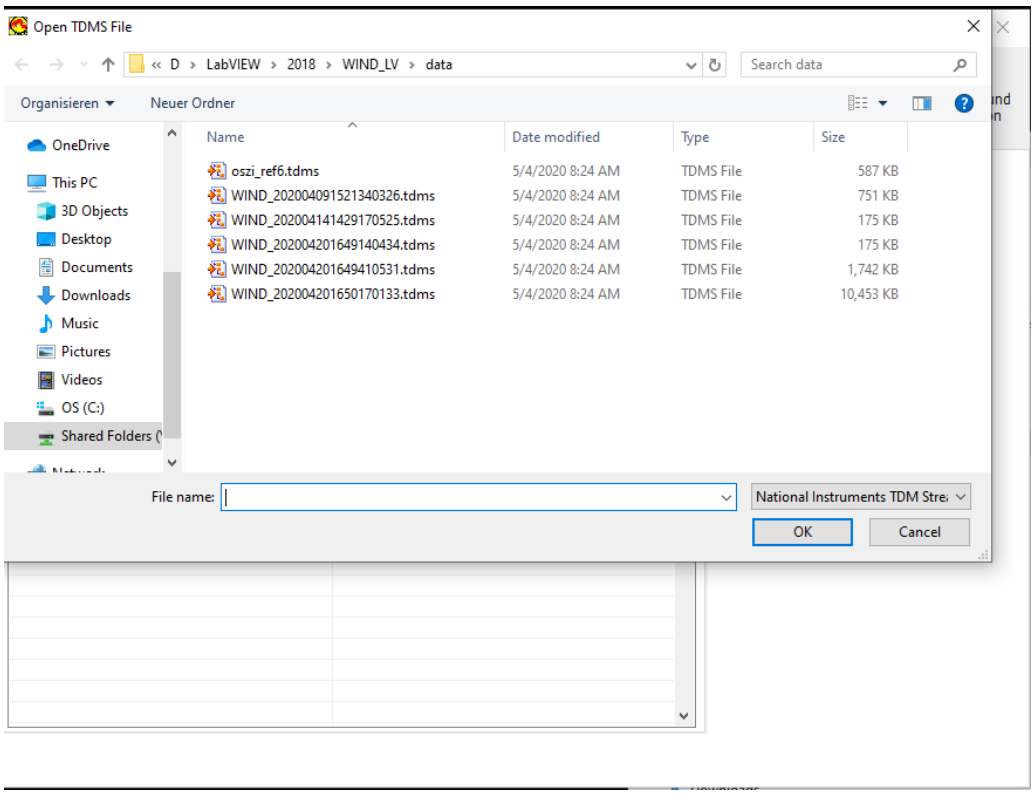
Initialization File	path of the initialization file,
Data Directory	current data directory, can be changed in the configuration dialog,
Global Parameters	measurement situation parameters, can be changed in the configuration dialog,
StartParameters	communication parameters submitted to the TCP/IP Server while starting a series of acquisitions (for service),
DATA	latest data received by the TCP/IP Server (for service),
Timestamps	relative timestamps of the series of acquisitions.

4.4 Viewer

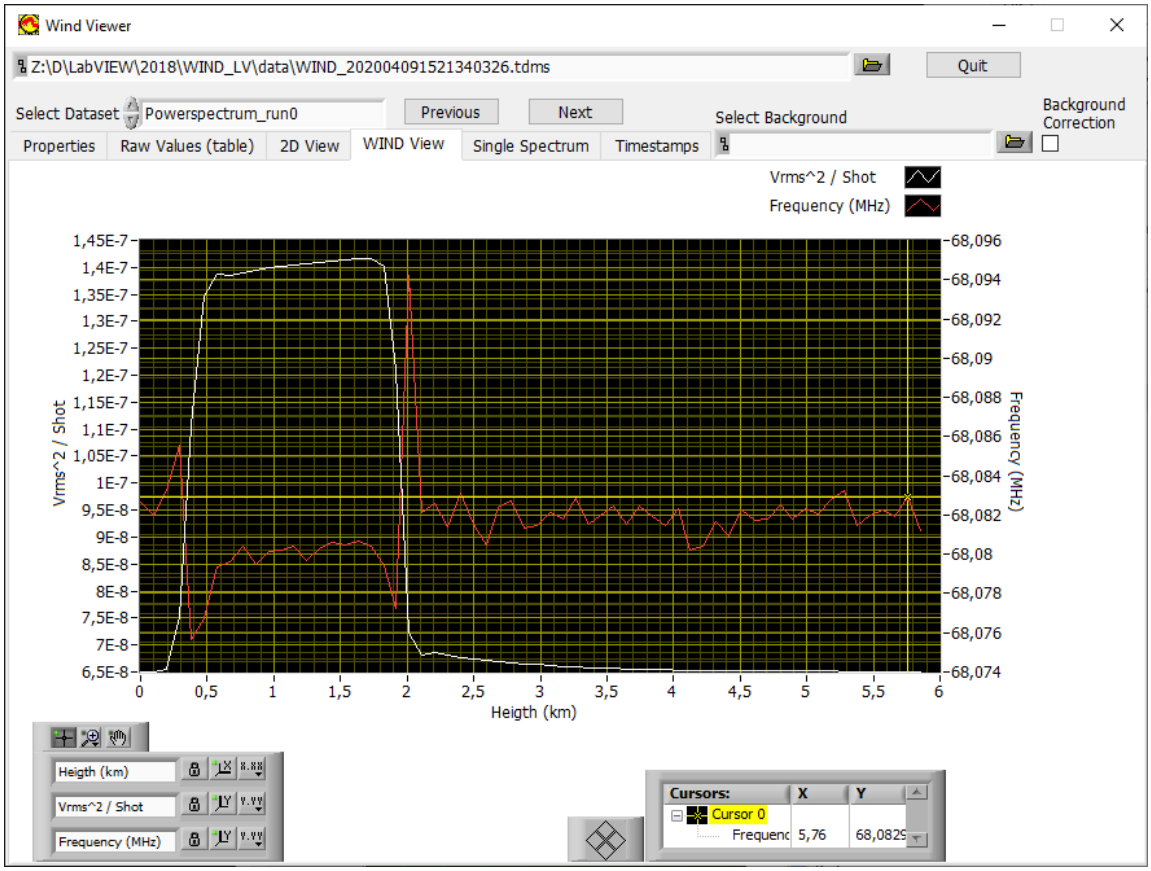
A data Viewer component (stand-alone) is provided as part of the delivered LabVIEW source project (*WindViewer.vi*, found in the corresponding project folder) as well as a Windows executable *Wind Viewer.exe*. With this reader you can load the `TDMS` file generated with the `WindAcquis` software. It is possible to carry out a background correction. To do this, the background file must be selected and activated with the `Background Correction` tick.

4.4.1 Run the Viewer

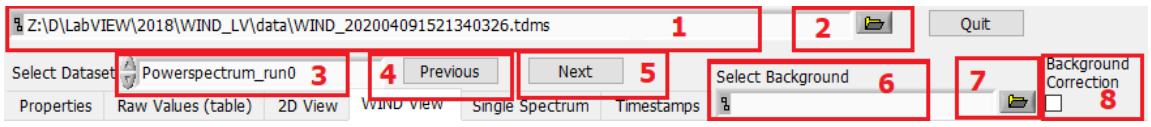
After running the program a (TDMS) file selection dialog comes up and allows the user to select a Waverider data file:



after clicking the button *OK* OK the selected file will be loaded into the Waverider data viewer (Wind Viewer).



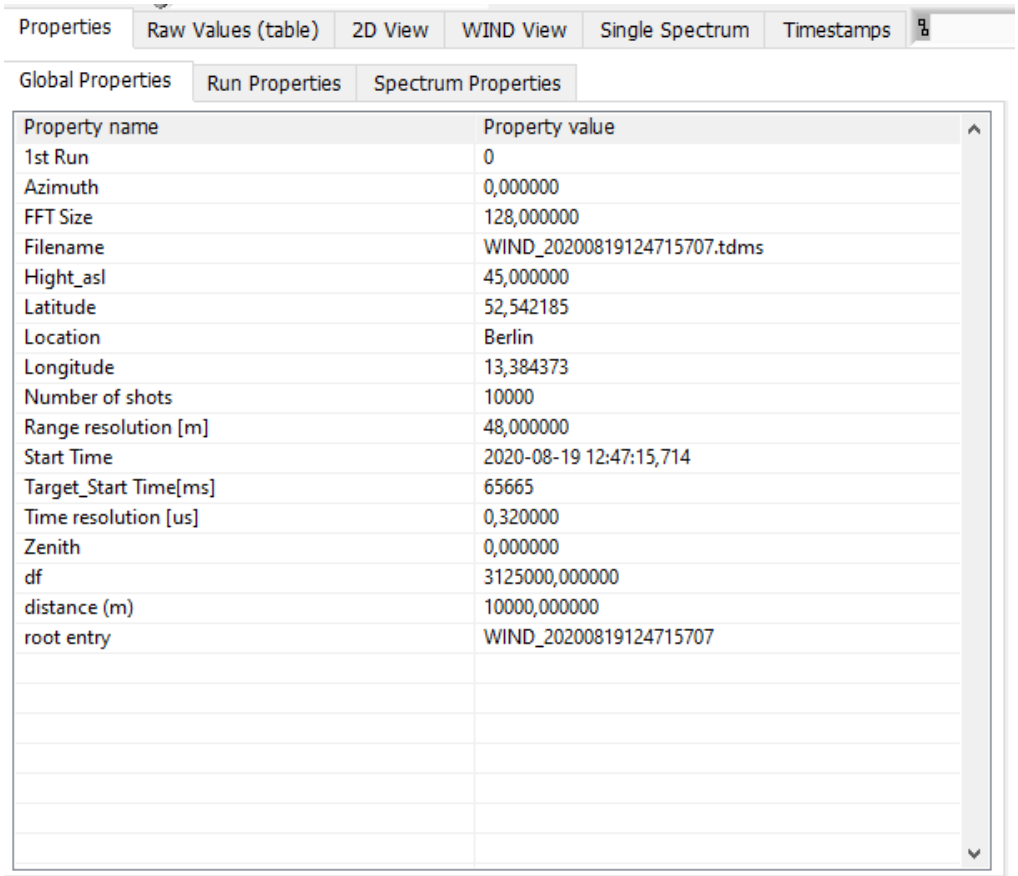
The data file related control elements are placed at the top of the viewer's front panel:



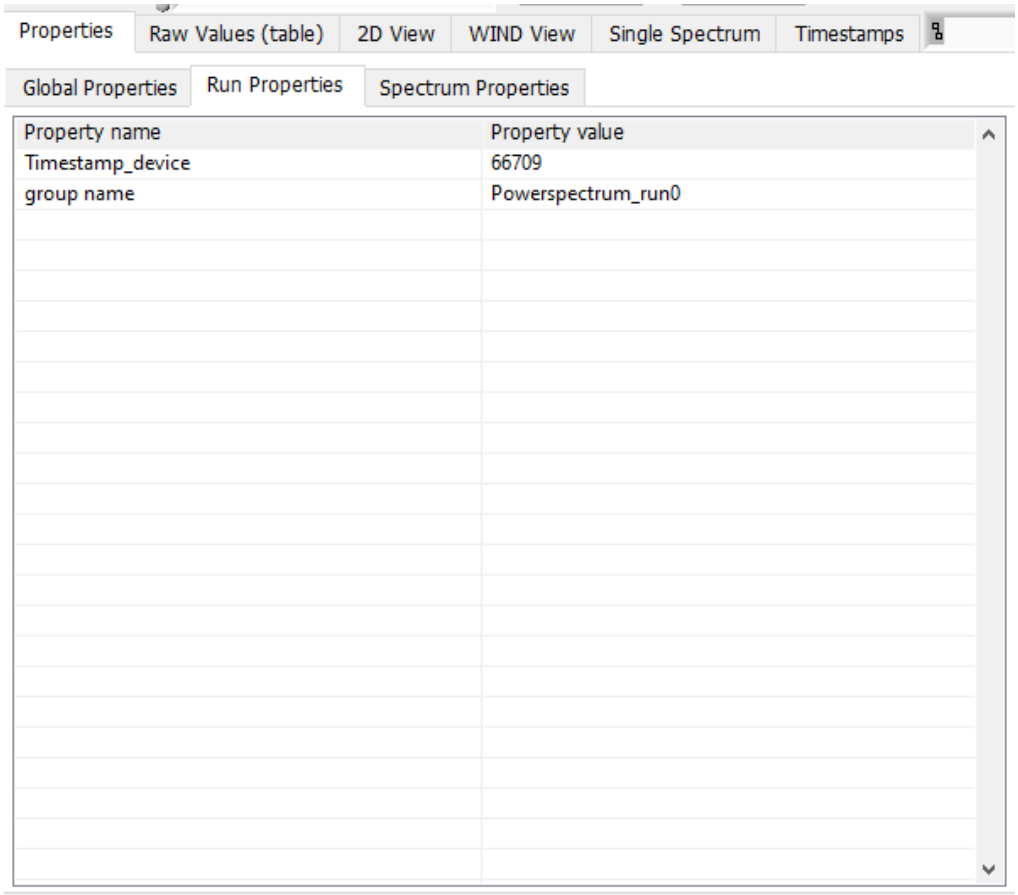
1. The file path is shown.
2. The browse button allows to open the file selection dialog to change the loaded file.
3. Select the dataset manually.
4. A click on the *Previous* button will load the previous dataset, if the current dataset is the first dataset of the file then the previous file in the directory will be loaded.
5. A click on the *Next* button will load the next dataset, if the current dataset is the last dataset of the file then the next file in the directory will be loaded.
6. Selected background file path is shown.
7. The browse button allows to open the file selection dialog to change the loaded background file.
8. Enable or disable the background correction, if no path is selected the correction is automatically disabled.

4.4.2 View Acquisition Details

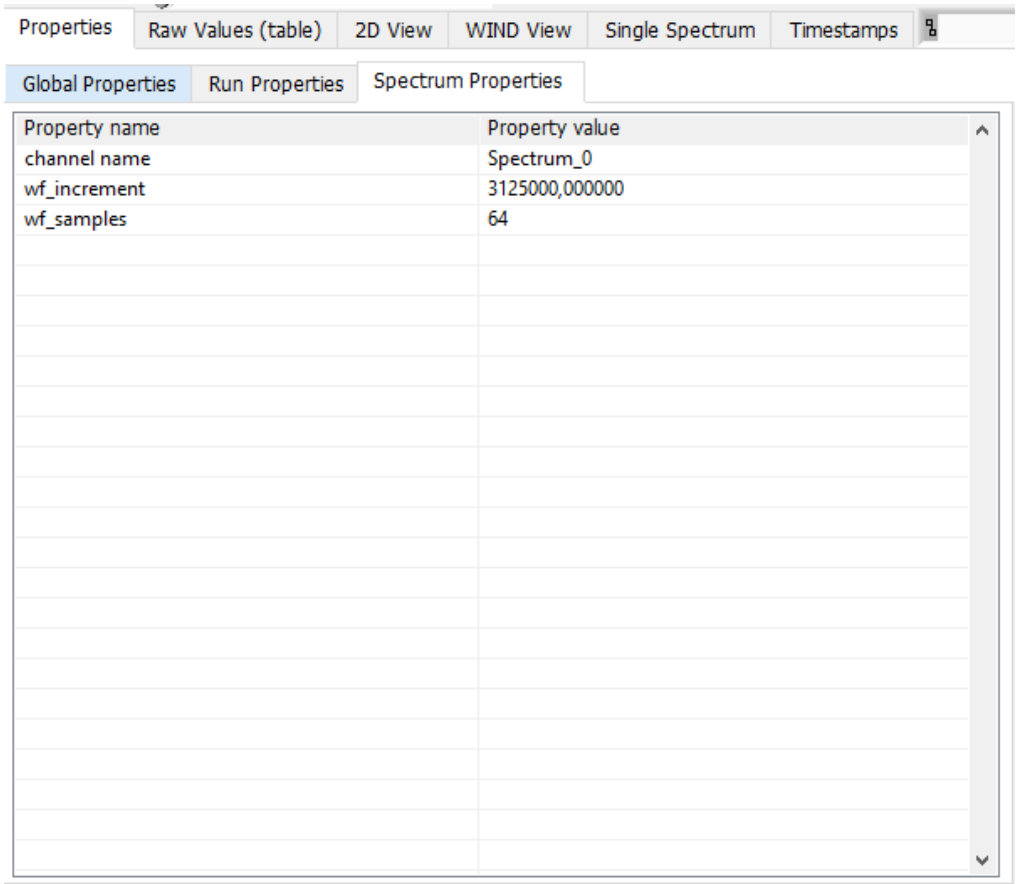
It is possible to inspect several details corresponding to the selected dataset. Some of the details are specific for the acquisition series, the runs (acquisitions), or the spectra.
The `Global Properties` of the `acquisition` include the acquisition parameters and the global measurement site settings.



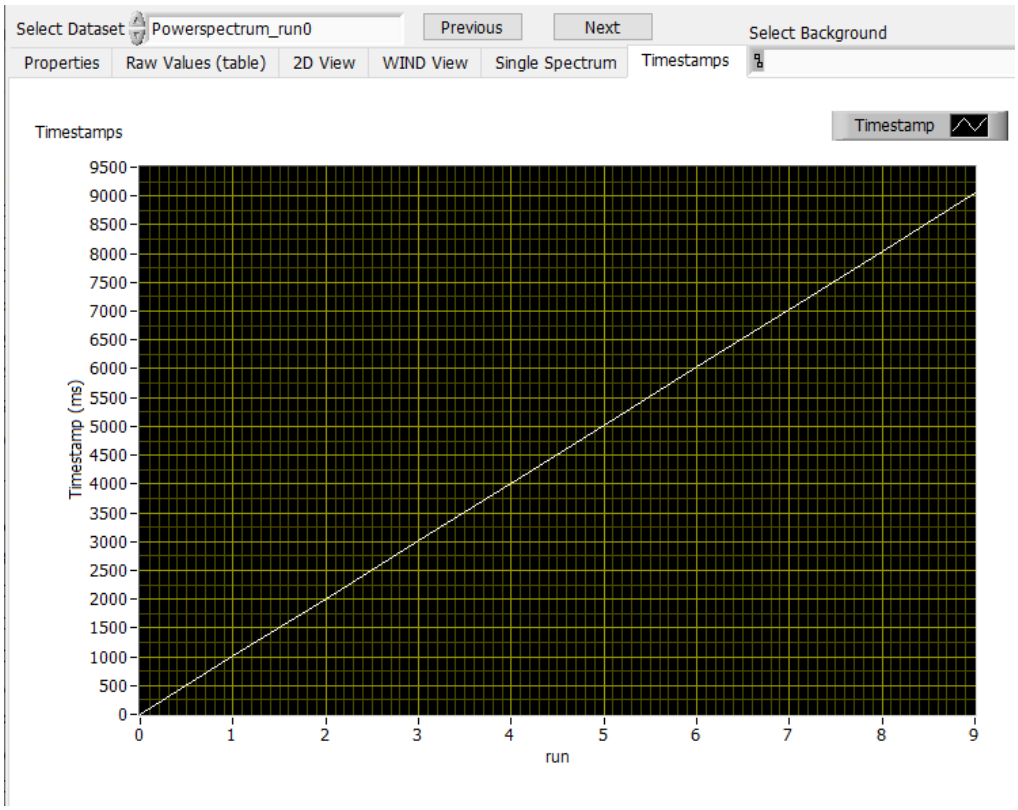
`Run Properties` contains the timestamp information of the selected run. The difference between `Target_Start Time[ms]` and `Timestamp_device` gives an information about the time required for acquiring the selected dataset.



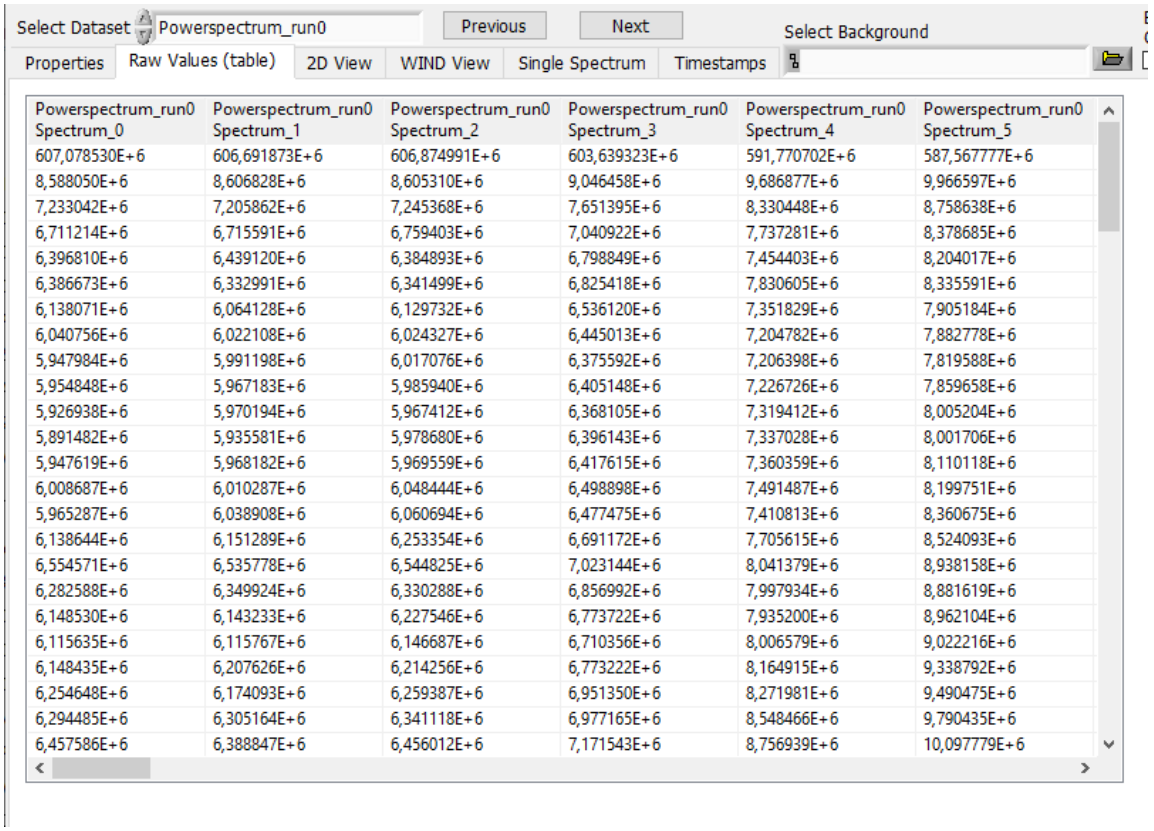
The Waveform information of the spectrum can be looked up in the `Spectrum Properties` Tab.



The relative *Timestamps* of all *runs (acquisitions)* of the *acquisition series* consist of a diagram plotting the controller time (in milliseconds with respect to the 1st run) as a function of the run index. It should show a linear behavior if all records have been correctly acquired.

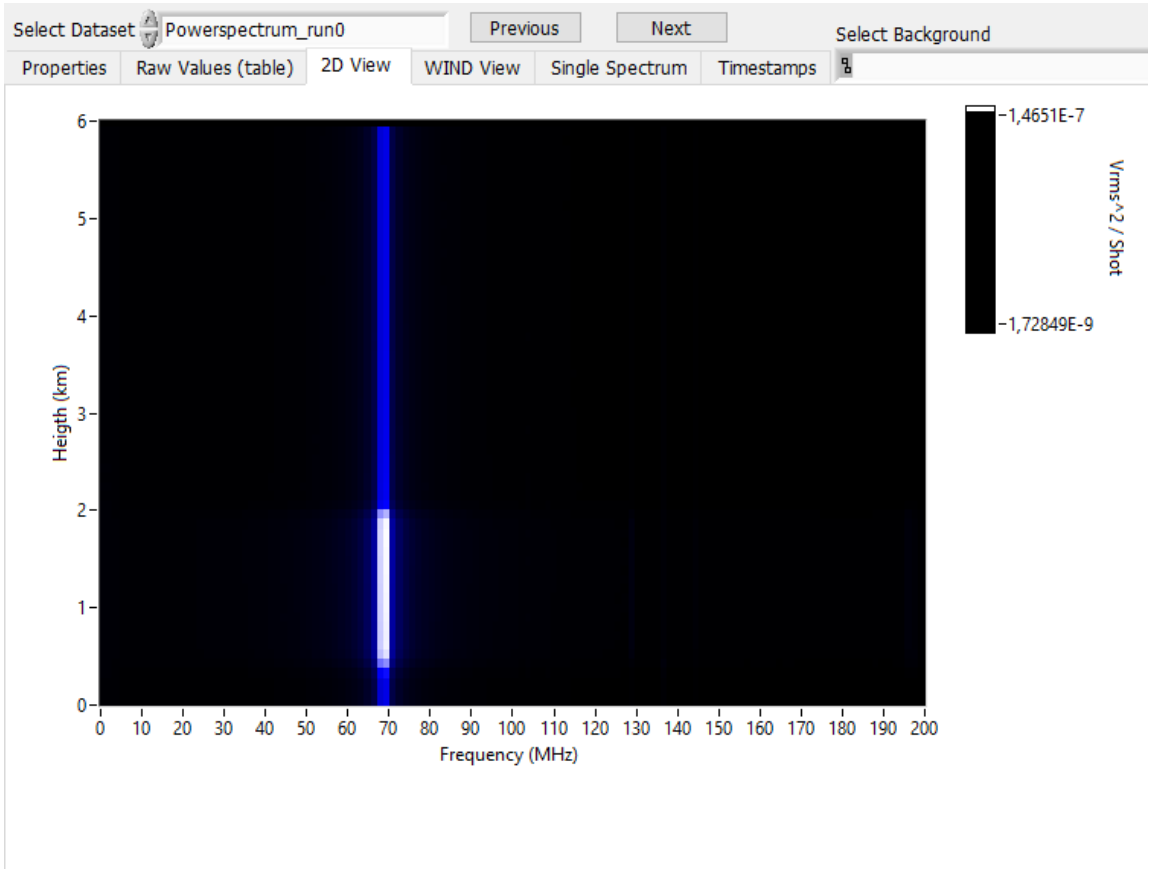


The `Raw Values` of a run (acquisition) or a spectrum can be inspected at the appropriate tab page.

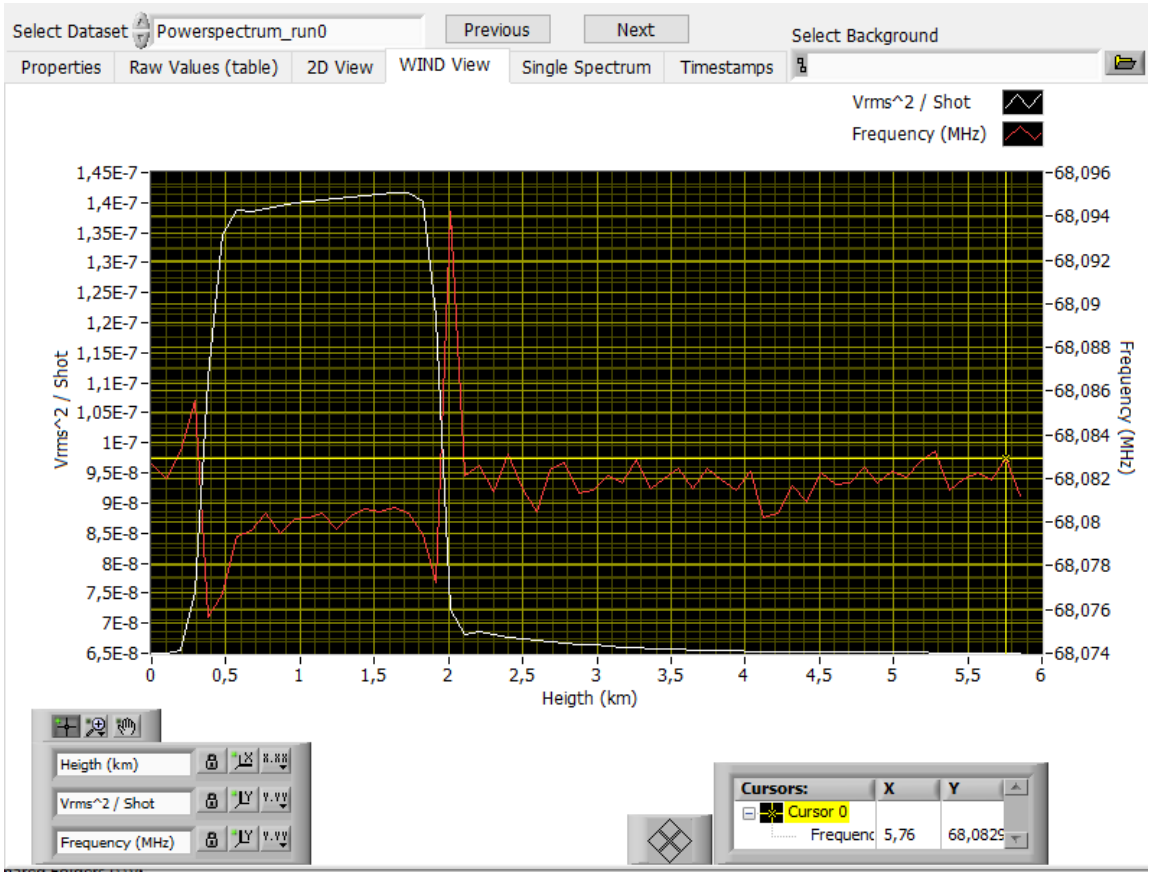


Powerspectrum_run0 Spectrum_0	Powerspectrum_run0 Spectrum_1	Powerspectrum_run0 Spectrum_2	Powerspectrum_run0 Spectrum_3	Powerspectrum_run0 Spectrum_4	Powerspectrum_run0 Spectrum_5
607,078530E+6	606,691873E+6	606,874991E+6	603,639323E+6	591,770702E+6	587,567777E+6
8,588050E+6	8,606828E+6	8,605310E+6	9,046458E+6	9,686877E+6	9,966597E+6
7,233042E+6	7,205862E+6	7,245368E+6	7,651395E+6	8,330448E+6	8,758638E+6
6,711214E+6	6,715591E+6	6,759403E+6	7,040922E+6	7,737281E+6	8,378685E+6
6,396810E+6	6,439120E+6	6,384893E+6	6,798849E+6	7,454403E+6	8,204017E+6
6,386673E+6	6,332991E+6	6,341499E+6	6,825418E+6	7,830605E+6	8,335591E+6
6,138071E+6	6,064128E+6	6,129732E+6	6,536120E+6	7,351829E+6	7,905184E+6
6,040756E+6	6,022108E+6	6,024327E+6	6,445013E+6	7,204782E+6	7,882778E+6
5,947984E+6	5,991198E+6	6,017076E+6	6,375592E+6	7,206398E+6	7,819588E+6
5,954848E+6	5,967183E+6	5,985940E+6	6,405148E+6	7,226726E+6	7,859658E+6
5,926938E+6	5,970194E+6	5,967412E+6	6,368105E+6	7,319412E+6	8,005204E+6
5,891482E+6	5,935581E+6	5,978680E+6	6,396143E+6	7,337028E+6	8,001706E+6
5,947619E+6	5,968182E+6	5,969559E+6	6,417615E+6	7,360359E+6	8,110118E+6
6,008687E+6	6,010287E+6	6,048444E+6	6,498898E+6	7,491487E+6	8,199751E+6
5,965287E+6	6,038908E+6	6,060694E+6	6,477475E+6	7,410813E+6	8,360675E+6
6,138644E+6	6,151289E+6	6,253354E+6	6,691172E+6	7,705615E+6	8,524093E+6
6,554571E+6	6,535778E+6	6,544825E+6	7,023144E+6	8,041379E+6	8,938158E+6
6,282588E+6	6,349924E+6	6,330288E+6	6,856992E+6	7,997934E+6	8,881619E+6
6,148530E+6	6,143233E+6	6,227546E+6	6,773722E+6	7,935200E+6	8,962104E+6
6,115635E+6	6,115767E+6	6,146687E+6	6,710356E+6	8,006579E+6	9,022216E+6
6,148435E+6	6,207626E+6	6,214256E+6	6,773222E+6	8,164915E+6	9,338792E+6
6,254648E+6	6,174093E+6	6,259387E+6	6,951350E+6	8,271981E+6	9,490475E+6
6,294485E+6	6,305164E+6	6,341118E+6	6,977165E+6	8,548466E+6	9,790435E+6
6,457586E+6	6,388847E+6	6,456012E+6	7,171543E+6	8,756939E+6	10,097779E+6

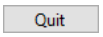
The scaled data of a run (acquisition) or a spectrum can be seen in the *2D View* where the power amplitude is plotted in a color magnitude against the height and frequency.



The peak power and the heights of a run (acquisition) are plotted against the frequency in the *WIND View* graph.



4.4.3 Stop the Viewer

The execution of the *Wind Viewer* can be stopped using the button *Quit* 

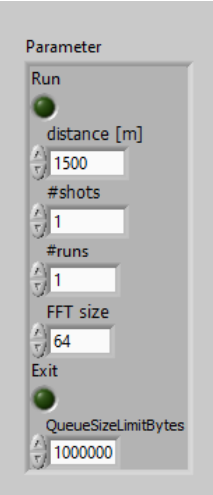
(leaves the front panel open for the next start) or by clicking the window's close symbol × at the top right (this will close the front panel after execution).

Chapter 5

Queue Programming Interface

In addition to the direct TCPIP interface for the controller an easy accessible queue interface is available. The interface has two queues the `WIND_CMD` queue which takes a parameter cluster containing the required information for controlling an acquisition and the `WIND_DATA` providing the raw data and auxiliary data to convert the raw data into physical values. All applications described above use this interface. Both queues are handled by the TCPIP Server which translates the queues into TCPIP socket handling.

The WIND_CMD Queue The Parameter cluster is shown below:



Setting the `Run` boolean causes a START command issued. Resetting it causes a STOP command. The desired distance together with the FFTSIZE is translated into the required number of FFT.
The `#shots` define the number of shots per run. And the `#runs` define how many runs should be aquired. Setting the `Exit` boolean causes a termination of the TCPIP Server.
The `QueueSizeLimitBytes` define the limit of the queue size in bytes.

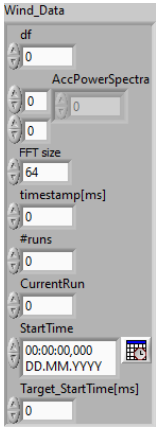
5.1 TCPIP Server

The TCPIP server is usually launched as a background service by calling the `WIND Acquisition Start TCPIP Server.vi`. After start the TCPIP Server reads the `WIND.ini` file. The user should enter the corresponding IP address in the client section.

```
[Client]
IP = 192.168.69.58
```

```
PORT = 2055
```

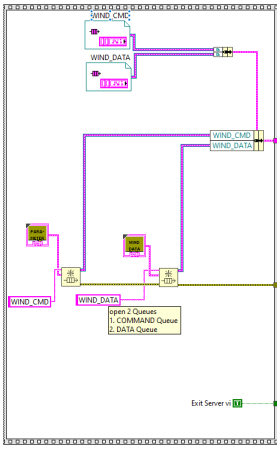
With this setting the TCPIP server would connect to the controller at `192.168.69.58` and open two sockets the command socket at `2055` and the push data socket `2056`
The TCPIP server then waits till the `Parameter` cluster is send over the `WIND_CMD` queue and executes then the required actions on the command socket and waits for data on the push data socket. Once data arrives there it sends the data over the `WIND_DATA` queue back to client program. The `WIND_DATA`



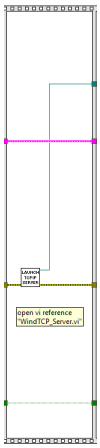
contains the

- **df** - the frequency increment of the power spectra in MHz.
- **AccPowerSpectra** - accumulated power spectra for each range bin there is one FFT and they are following each other, knowing the FFT size this array can be converted to a 2D array.
- **FFTsize** - number of points for the FFT, the power spectra contains only half of this number points.
- **timestamp** - time stamp of data (compare MSEC? on the target) this are the milliseconds since Waveride controller start when the data became available in the Waverider controller. To convert the timestamp into a real world time compute (timestamp - Target_StartTime)* 0.001 + StartTime.
- **#runs** - selected number of runs
- **CurrentRun** - indicates the already acquired number of runs
- **PC Start Time** - PC time when the start command has been issued
- **Target_StartTime[ms]** - Start time of the target in [ms], this are the Waverider milliseconds since start at the moment when the start command has been issued.

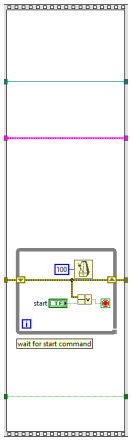
5.2 Queue Client - Getting Started



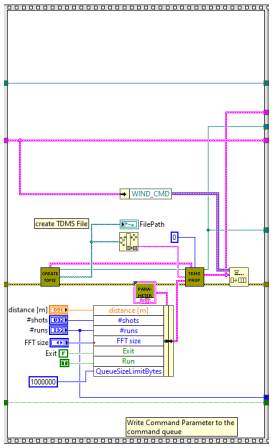
First we create the queue based on the ctl typedefs. Set the constant `Exit Server vi` true, to exit the server vi after the process.



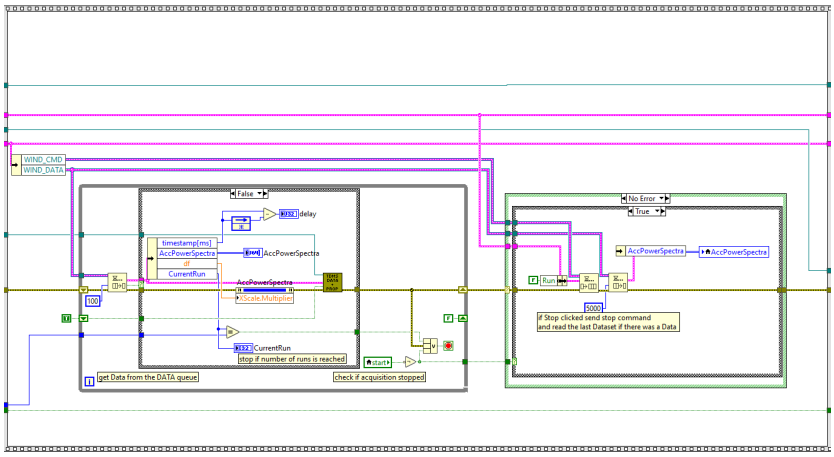
The we launch the TCPIP server. It will connect it self to the waverider. If not it will pop up with an error.



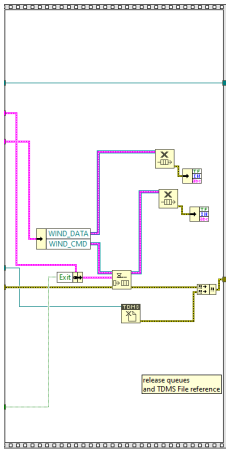
Then we wait for a user to start, this could be some external synchronization required



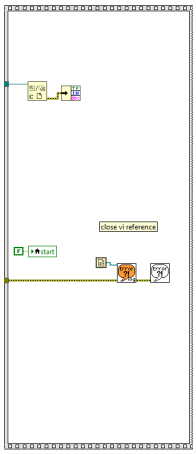
A TDMS file gets prepared and a request for data is send to the `WIND_CMD` queue.



As data arrives it is written into the TDMS file and if the user stops the loop the data acquisition is aborted. After the acquisition cycle the run variable is set to false issuing a STOP



The TDMS file is closed and the queues are released. If the constant `Exit Server vi` was true, the server vi will be closed too.



Then one closes the VI reference to the server and shows errors if they occurred.

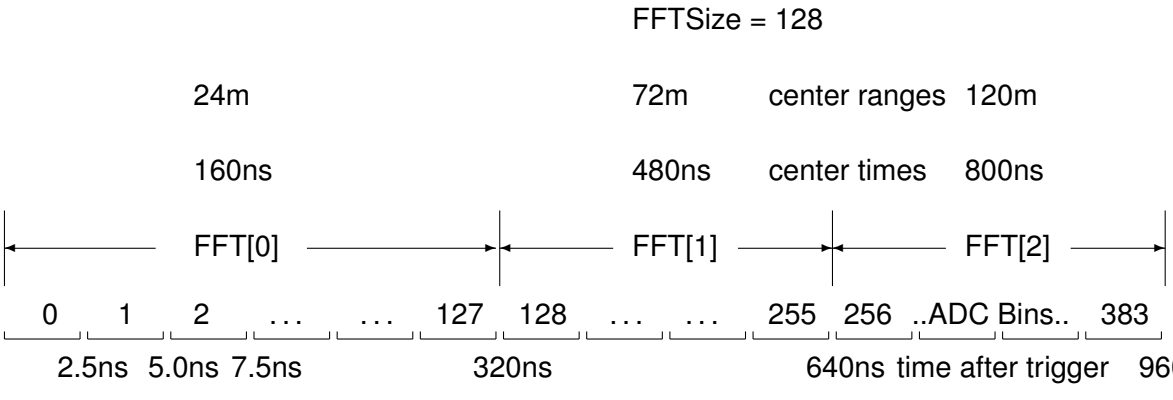
Chapter 6

Data transfer - Low Level Description

6.1 Operation principle

6.1.1 FFTSize

The FFTsize defines how many original ADC samples go into one FFT. The ADC stream after the trigger is cut into chunks defined by the FFTSize. Currently 64, 128 and 256 are the possible chunk sizes for Waverider v1 (400MHz). Waverider v2 (3,2GHz) has only one possible chunk size 2048. Below follows a description what that means for a 128 chunk size.



For a 400 MHz sampling frequency with 128 Samples the spectral resolution of the power spectra is

$$\Delta \mathrm{f} = \frac{400\mathrm{MHz}}{128} = 3.125\mathrm{MHz} \tag{6.1}$$

For a 3200 MHz sampling frequency with 2048 Samples the spectral resolution of the power spectra is

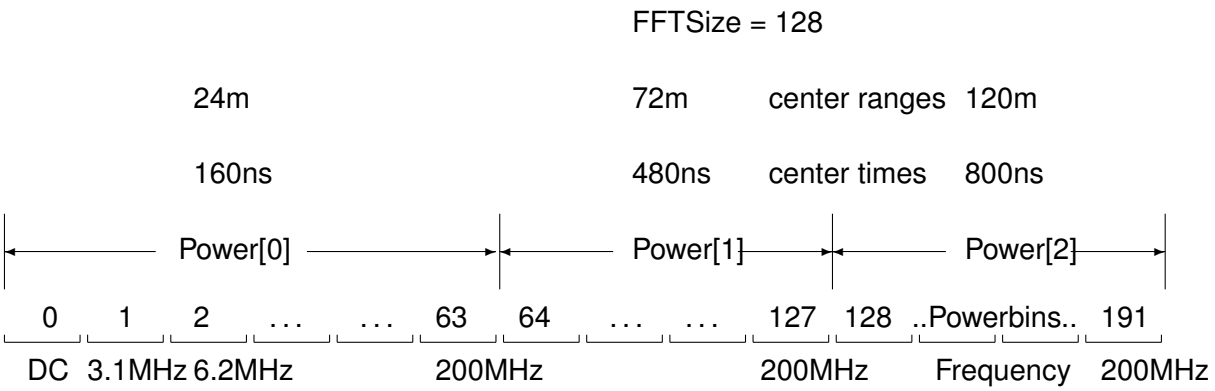
$$\Delta \mathrm{f} = \frac{3200\mathrm{MHz}}{2048} = 1.5625\mathrm{MHz} \tag{6.2}$$

6.1.2 Distance

Defines the maximum distance the ADC trace should cover. For a 15km trace this would correspond to 100 μs. This would be 100(μs) * 400 (MHz) = 40000 ADC bins. The trace should be a aligned to the FFTsize so 312 FFT would fit into this. -The system will then work 312 *128 = 39936 ADC bins.

6.1.3 Power spectra

The result of the FFT is a power spectra. The number of elements in the power spectra is half of the number the FFTSize chunks. The system will then work 312 *128 / 2 = 19968 power spectra bins.



6.1.4 Shots

Each trigger event (laser shot) results in a power spectra vector. The power spectra are not individually stored but are accumulated.

6.1.5 Runs

One run consists of accumulation of `shots`. The results are then as described below transmitted to the PC. One can define how many transmissions are requested. -1 is special value requiring an infinite amount of tranmissions. This infinite mode can be terminated by a STOP command.

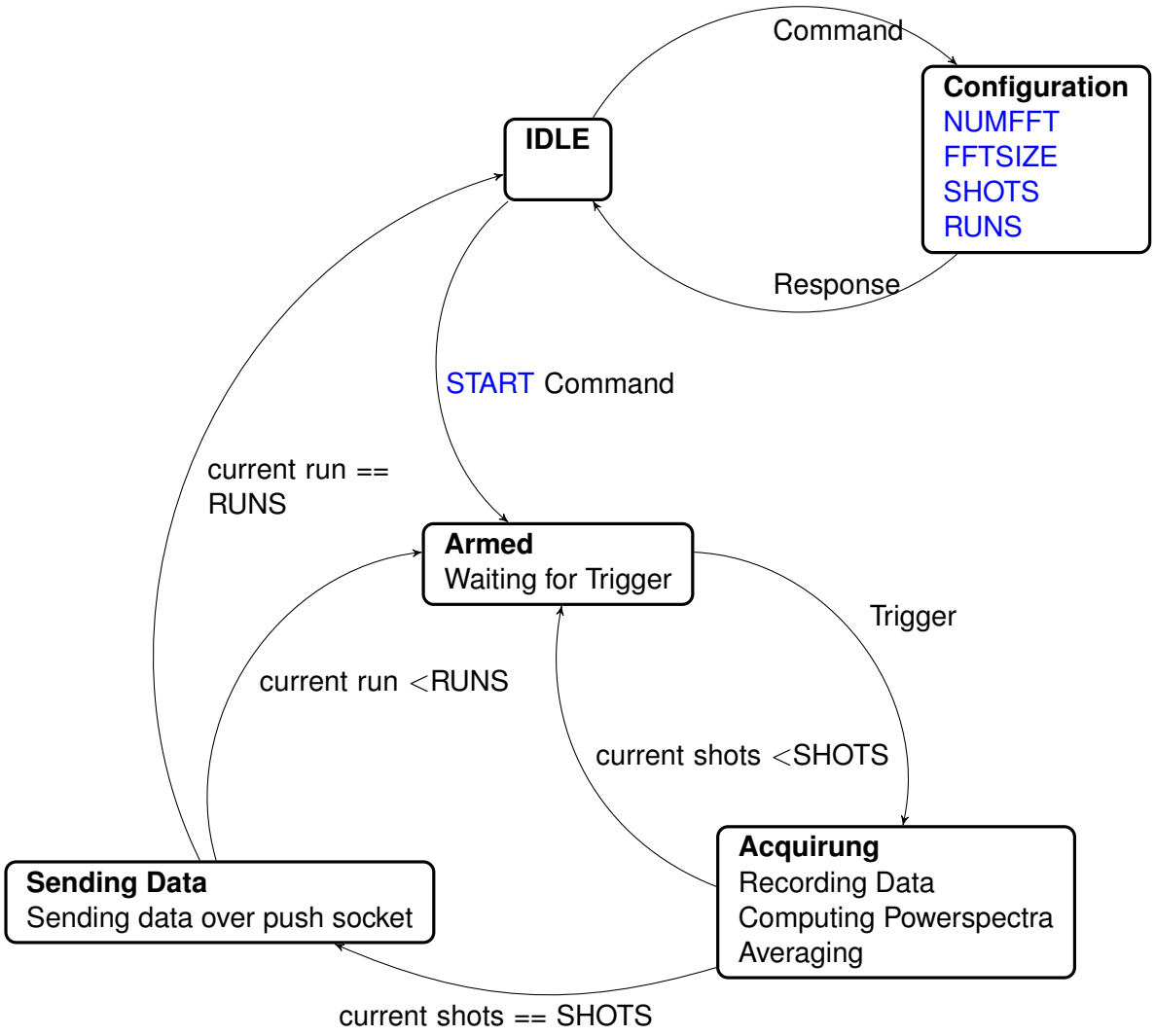
6.2 TCPIP Communication

The communication between the PC and the controller is implemented as TCPIP sockets. There are two sockets. First there is the command socket. The communication here is bidirectional. The PC is sending a command and the controller is answering with a response. The acquired data is sent over the second socket, the push socket. The data is send asynchronously when a run is finished. There is no command to request the data, the data is pushed towards the PC. The START command will fail if the push socket connection has not been established.

Before starting an acquisition the acquisition parameters should be set. After sending the START command the controller will change into the armed state. Once a valid trigger is received the acquisition and later the power spectra computation runs. After each acquisition the system checks if the desired number of shots per run has been reached. If not it changes again into the armed state. If yes it transmits the data over the push socket.

The system then checks if more runs are required, if yes it changes again into the armed state if not it goes back into the idle state.

The `Wind Acquisition_TCPIP.vi` demonstrates how this works. By implementing a queue interface this VI frees the user from handling these communication details. See the Queue Interface for more explanations.


Command
Configuration
NUMFFT
FFTSIZE
SHOTS
RUNS
IDLE
Response
START Command
current run ==
RUNS
Armed
Waiting for Trigger
Trigger
current run <RUNS
current shots <SHOTS
Acquirung
Recording Data
Computing Powerspectra
Averaging
Sending Data
Sending data over push socket
current shots == SHOTS

6.3 Push Data Package format

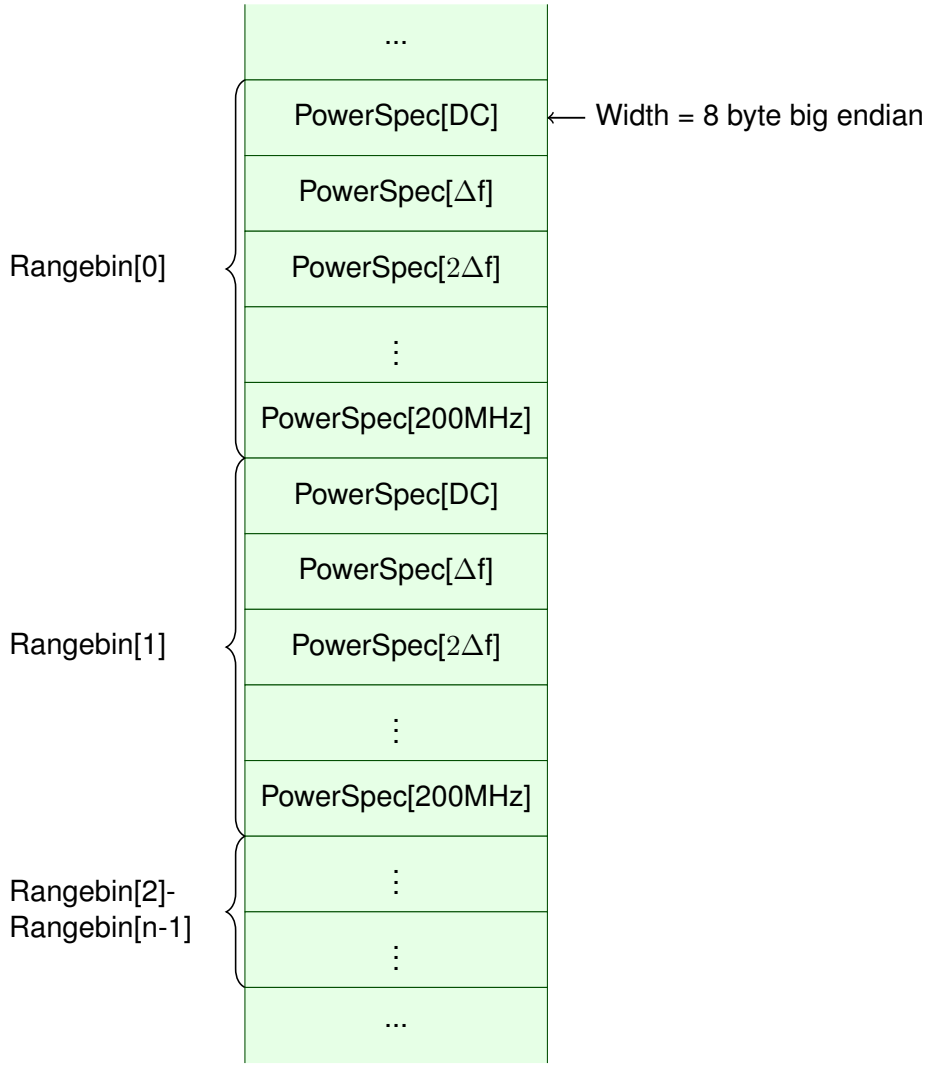
The push data is 8 byte aligned. The data is big endian. The first 8 bytes contain a marker and a time stamp. Then comes the number of data points in the data set. And then the data set itself.

data set marker = 0xFFFFFFFF [4 bytes]
times tamp [4 bytes]
number of data points [8 bytes]
power spectra entries [8 bytes]

This structure allows a resynchronization by searching for the marker and a reasonable time stamp. See the push loop in the `Wind Acquisition_TCPIP.vi` for an example of this resynchronization.

6.4 Accumulated Power Spectra

The Push data contains the accumulated power spectra as 64 bit integers. Its a 2D matrix where the columns are for the different frequencies between DC and 200MHz and the rows are the different range bins. This matrix gets then flattened into a vector which is described below.



6.5 Raw Data to Physical Value Conversion

The conversion starts with a normalization with the shot number. After this step the analog data shows the mean ADC bit values. The stored raw values equals the summation of FFT^2.

The analog data needs then to be scaled by the ADC max value.

$$\texttt{FFT}^2 = \frac{AccPowSpectra}{Number\ of\ shots * (2^{ADCbits-1})^2}$$

For a 12 Bit ADC and Number of shots equal to 100 this means

$$\texttt{FFT}^2 = \frac{AccPowSpectra}{100 * (2048)^2}$$

The FFT^2 can be converted to the Unit $Vrms^2$ by using the `FFT size` Parameter.

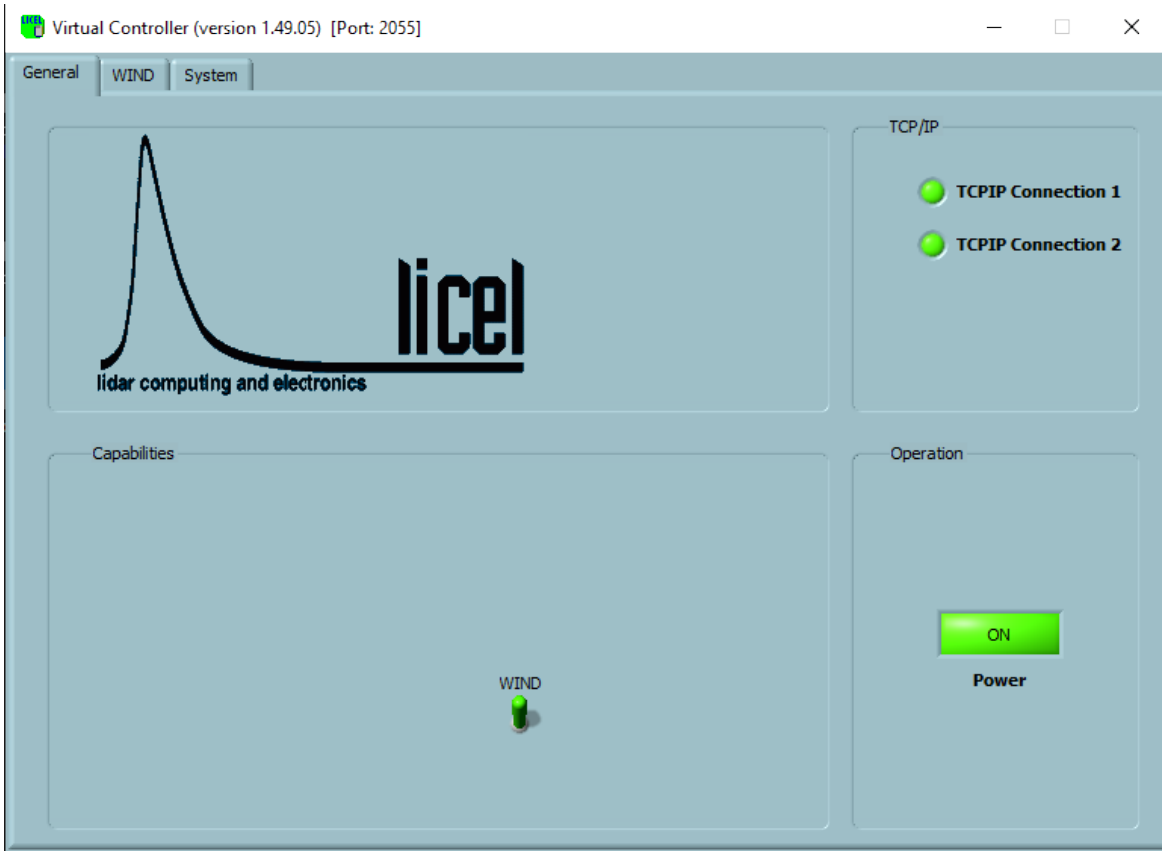
$$Vrms^2 = \frac{FFT^2}{(FFT\ size)^2}$$

Chapter 7

Simulation

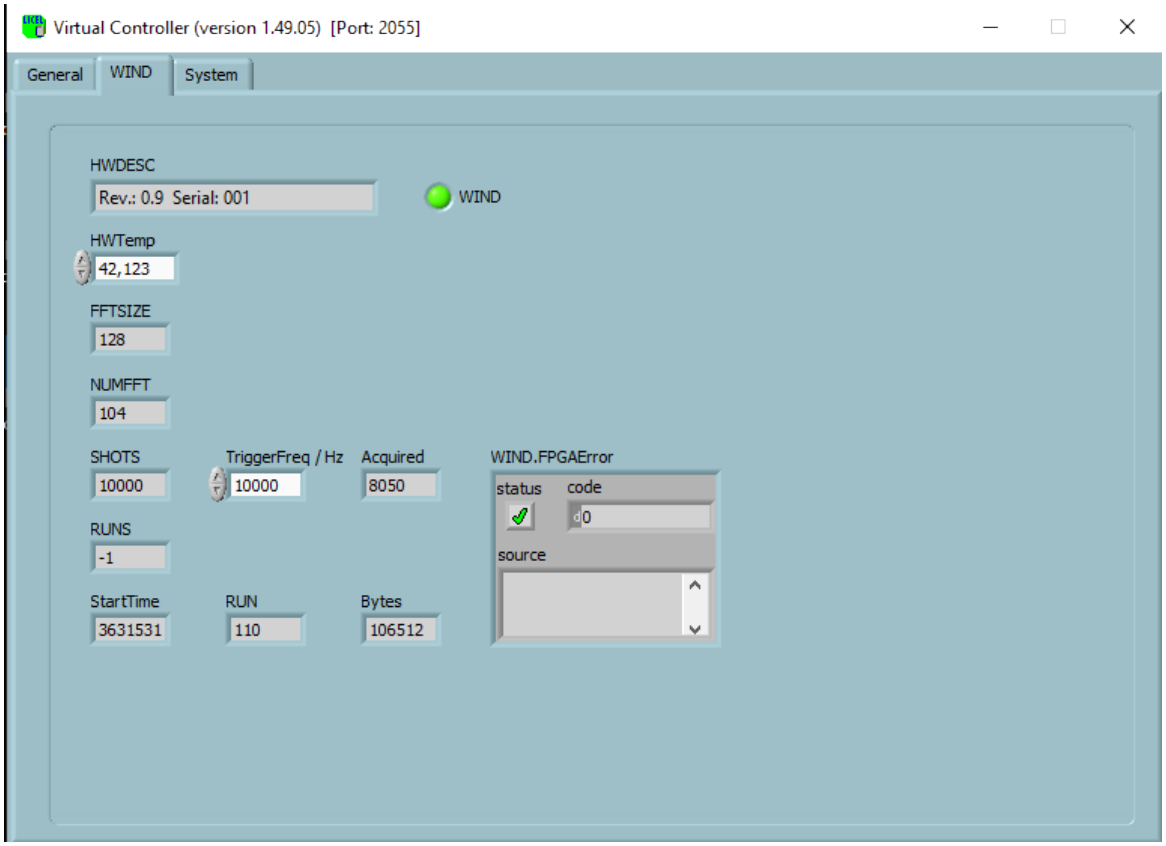
All software module can be also tested in Simulation mode. One needs a Virtual Controller. This is part of the executables in http://licel.com/download/ethernet/WaveriderLVInstaller.zip

It should start up as



Please note the green `Wind` switch, indicating the support for the Wind commands. The general capabilities are described at in the Licel Ethernet Controller Installation and Reference Manual chapter 8.

In addition to the there described capabilities is one tab with the Wind support.



To run the simulation start first the `Virtual Controller.exe`. Enter `127.0.0.1` as the IP number in the `Wind.ini` in the `[Client]` section and start the desired software module. Once it runs the TCPIP indicators at the entrance panel should become green and the software should run with simulated data.

Chapter 8

Specifications

Analog acquisition:

Signal input range:	+- 1V
A/D Resolution:	12 Bit
Impedance	50 Ω
Sampling rate:	3200 MSamples/second.
Lidar spatial resolution:	96 m.
Bandwidth:	AC- 3200 MHz.
SNR:	58db
trace length max:	11km
max. shots per run	$4 * 10^9$

Trigger input:

Impedance:	20 kΩ
V_{IH}:	2V
V_{IL}:	0.8V
V_{max}:	3.6V

Chapter 9

Appendices

9.1 TCP/IP Command List and Syntax

This section lists and describes the TCP/IP command syntax for Licel TCP/IP Ethernet Controllers. Most commands can be sent either in a short form or a long form. <CRLF> is carriage return line feed. All commands sent to the TR should end with <CRLF>, and all replies from the Licel TCP/IP controller end with <CRLF> which will not explicitly be shown in this document.
If the controller detects an unknown command it will return the string

<command> unknown command

back to the caller where <command> is the command originally sent.
The following commands are available dependent on the Waverider controller you ordered.

Short	Long
CAP?	CAP?
*IDN?	IDENTIFICATION?
	KILL
MSEC?	MILLISEC?
PASS	PASS
	HOSTNAME?
	HOSTNAME
START	START
STOP	STOP
FFTSIZE	FFTSIZE
FFTSIZE?	FFTSIZE?
NUMFFT	NUMFFT
NUMFFT?	NUMFFT?
SHOTS	SHOTS
SHOTS?	SHOTS?
RUNS	RUNS
RUNS?	RUNS?
CURRENTSHOTS?	CURRENTSHOTS?
ERROR?	ERROR?
TCPIP	TCPIP
HWDESC?	HWDESC?

CAP?

Requests the control capabilities of the controller.

The controller's response is

CAP: [List of Capabilities],

where List of Capabilities is a space-separated list with on or more of the following items:

WIND for a Wind Waveridercontroller	
TR	for controlling transient recorder
APD	for APD remote control
PMT	for PMT remote control
TIMER	for the trigger timing controller
CLOUD	for transient recorder controller cloud mode

A response could be

CAP: TR

for a controller which is able to control transient recorders, only, while

CAP: APD PMT TIMER

indicates a controller capable of controlling APDs PMTs and the timing generator.

*IDN?

IDENTIFICATION?

Asks the controller to send its identity and firmware revision. The reply from the controller is e.g.

KILL <SOCKETS> <Password>

Causes the controller to close all TCP/IP connections. Password is the internal password of the controller. This command can be used only at a TCP/IP connection with the controller on the 3rd supported Ethernet port, i.e. on Port + 2 when Port is he Ethernet port used for the bidirectional communication. The default is $2055 + 2 = 2057$. If required, the base port can be changed using the TCPIP command, the internal password (default: *Administrator*) can be changed with the PASSWORD command. KILL SOCKETS must be sent before reopening the TCP/IP communication with the controller.
Usage:

1. Open a TCP/IP connection to the controller at the 3rd Ethernet port, i.e. Port + 2 (default $2055 + 2 = 2057$).
2. Immediately send KILL SOCKETS (terminated by <CRLF>).
3. Ignore all communication errors, the controller will close the connection on Port + 2, as well.

MILLISEC?

MSEC?

Requests the millisecond timer value of the controller. The reply is

MILLISEC: time

where time is a number with the milliseconds since the start of the controller.

HOSTNAME?

Requests the hostname of the controller. The reply is

HOSTNAME: LicelWIND

If the controller is controlled via DHCP and the name resolution is working properly one can access the controller also with this symbolic name for instance

ping LicelWind should work form the MSDOS command line.

HOST <"NewHostName"><"Password">

After sending this command, the controller replies with the string

`Hostname set to LicelWIND`

If the <"NewHostName">was `LicelWind`

It will set the hostname of the controller. If the controller is controlled via DHCP and the name resolution is working properly one can access the controller also with this symbolic name.

If the password is wrong it will return

`HOST failed due to invalid password`

PASSWORD <"Old Password"> <"New Password"> <"New Password">

PASS <"Old Password"> <"New Password"> <"New Password">

Changes the password for the controller. The actual password is required to change the IP configuration of the transient recorder. The user needs to enter the old password and then the new password twice. The default password is "'*Administrator*"'. The password will be reset to this if a hardware reset is executed on the controller. For example

`PASS "Administrator" "MyPassword" "MyPassword"`

will change the password to MyPassword. The controller replies with

`PASSWORD set to "MyPassword"` ,

if an error occurs (wrong `Old Password`, nonequal `New Password` entries, or empty `New Password`s) the reply is

`PASSWORD not set.`

START

START

After sending this command, the controller replies with

`START executed.`

If the push socket has not been opened before the command returns

`Push socket not active.`

On the push socket after a run of shots averages has been finished it will the the push data b blocks

The push data is 8 byte aligned. The data is big endian. The first 8 bytes contain a marker and a time stamp. Then comes the number of data points in the data set. And then the data set itself.

data set marker = 0xFFFFFFFF [4 bytes]
times tamp [4 bytes]
number of data points [8 bytes]
power spectra entries [8 bytes]

This structure allows a resynchronization by searching for the marker and a reasonable time stamp. See the push loop in the `Wind Acquisition_TCPIP.vi` for an example of this resynchronization.

The time stamp corresponds to what MSEC? will return. The first data point comes from the first power spectra. Each power spectra will be half of the FFTSIZE.

The flow of data telegramms will repeat NUMFFT times.

STOP

After sending this command, the controller replies with the string
`STOP executed.`

FFTSIZE <0|1|2>

Here

- 0 corresponds to a 64 points FFT
- 1 corresponds to a 128 points FFT
- 2 corresponds to a 256 points FFT
- 5 corresponds to a 2048 points FFT.

After sending this command, the controller replies with the string
`FFTSIZE executed`

FFTSIZE?

queries the previously set FFT size.
the controller responds with
`FFTSIZE: 0`
Here

- 0 corresponds to a 64 points FFT
- 1 corresponds to a 128 points FFT
- 2 corresponds to a 256 points FFT
- 5 corresponds to a 2048 points FFT.

NUMFFT <number >

Sets the number of FFT to be computed after each trigger event.
After sending this command, the controller replies with the string
`NUMFFT executed`

NUMFFT?

queries the previously set number of FFTs.
`NUMFFT: 50`

SHOTS <number >

Sets the number of shots to be averaged for one run.
After sending this command, for instance
`SHOTS 5000`
the controller replies with the string
`SHOTS executed`

SHOTS?

queries the previously set number of shots for one run and a typical return would be
`SHOTS: 5000`

RUNS <number >

Sets the number of runs. Each run contains the previously described number of shots averages
After sending this command, for instance
`RUNS 4`
the controller replies with the string
`RUNS executed`
If a `-1` is passed as an argument, the system will acquire data infinetely until it receives a STOP

RUNS?

queries the current number runs and a typical return would be
`RUNS: 0`
if no acquisition has been finished.

CURRENTSHOTS?

queries the current number of shots in the actual run and a typical return would be
`CURRENTSHOTS: 5070`

ERROR?

queries the presence of an internal error
if no previous error has been recorded it will return
`ERROR 0 0`
if an error has been recorded it will return for instance
`ERROR 1 130042 FPGA error.`
the command will clear the error also so on the next run one will see again
`ERROR 0 0`

TCPIP <"ip#"> <"subnet mask"> <"Gateway"> <"Port"> <"Password">

TCP <"ip#"> <"subnet mask"> <"Gateway"> <"Port"> <"Password">

Sets the IP address, subnet mask, gateway and Ports that are used for TCP connections. Please note that the port numbers `Port`, `Port` + 1 and `Port` + 2 are used by the controller. This command will only be executed if the password corresponds with the controller's internally stored password. The defaults are

IP Address	10.49.234.234
Subnet Mask	255.255.255.0
Gateway	empty
Port	2055 .

In this case port 2055, port 2056, and port 2057 are used by default. Port 2055 is used for the bidirectional communication with the controller. The communication on port 2056 is monodirectional and contains the data that is pushed to the acquisition computer. Furthermore, port 2057 is used to enforce the controller to close all TCP/IP connections on the other ports (KILL SOCKETS). In order to restore the default values, the reset button needs to be pressed when powering up the controller

(hardware reset). The default password is "Administrator." To change the password, see the PASS command. For example

```
TCPIP "197.13.17.23" "250.250.250.29" " " "2013" "Administrator"
```

will change the IP Address to 197.13.17.23, the Subnet mask to 250.250.250.39, the gateway would be empty and the ports 2013 and 2014 would be used. The controller replies

```
IP "197.13.17.23" Subnet "250.250.250.39" Gateway " " Port "2013"
executed.
```

If the password is incorrect, then the reply is

```
TCPIP failed due to invalid password.
```

TCPIP "DHCP" <"Port"> <"Password">

TCP "DHCP" <"Port"> <"Password">

Enable DHCP mode on the network controller. The controller will listen at the specified port and at Port+1. This command will only be executed if the password corresponds with the controller's internal password. If not

```
TCPIP failed due to invalid password
```

will be returned. If the command is successfully executed the controller replies

```
DHCP activated.
```

The controller comes with the defaults described for the TCPIP IP command. A hardware reset will disable the DHCP mode.

HWDESC?

The controller returns the HW revision and serial number

```
HWDESC: Rev.:  1.0 Serial:  001
```

9.2 Data File Format

This appendix describes the file format written by `WindAcquis.vi`. The file format is a TDMS format where the measurement situation was written as properties, below the dataset description.

9.2.1 Filename

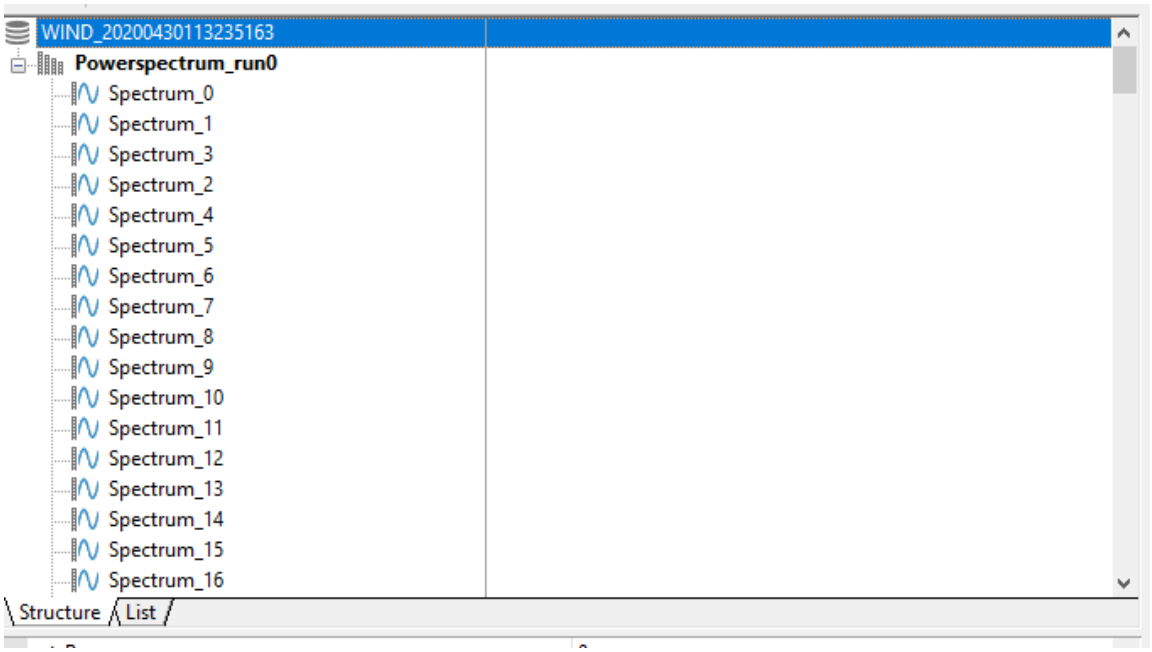
string, format	`WIND_YYYYmmddHHMMSSuuu`
`YYYY`	the year (decimal, four digits)
`mm`	the month (decimal, two digits)
`dd`	the day (decimal, two digits)
`HH`	the hour (decimal, two digits)
`MM`	the minute (decimal, two digits)
`uuu`	three decimal places of the seconds (decimal, three digits)

9.2.2 Properties

Location	Location where the measurement takes place.
Longitude	Longitude of the measurement site.
Latitude	Latitude of the measurement site.
Hight_asl	hight above sea level of the measurement site in meter.
Zenith	The zenith angle is the angle between the sun and the vertical.
Azimuth	The azimuth is the angle between a celestial body (sun, moon) and the north.
distance(m)	the distance (range) of the acquisition in meters.
FFT Size	Number of points per spectra.
Number of shots	selected number of shots.
Filename	Name of the File.
df	calculated delta f of the FFT.
Range resolution(m)	Range resolution in meter.
Time resolution(us)	Time resolution in microseconds.
1stRun	number of Run.
Timestamp_device	Time stamp of the accumulated data set in Waverider milliseconds when the summation has been completed at the Waverider controller. (Accuracy 1ms)
Target_Start Time(ms)	Time of the Waverider controller in Waverider milliseconds when the START command has been received. (Accuracy 1ms)
Start Time	PC start time. (Accuracy OS dependent for Win10 this is typical: 15ms)
root entry	TDMS root name is equivalent to Filename.
group name	TDMS group name.
channel name	TDMS channel name.
wf_increment	Interval in Hz between data points in the waveform.
wf_samples	The number of points stored in this channel.

9.2.3 File structure

The TDMS file is structured as shown below.



The TDMS root entry corresponds to the acquisition and is represented by the original file name. Each run is saved as a group named `Powerspectrum_runX`, where X is the number of run. Each group contains individual spectra, the following is used as the channel name, `Spectrum_Y`, where Y is the index of the spectrum.

Root entry	`WIND_YYYYmmddHHMMSSuuu`	Filename
Group name	`Powerspectrum_runX`	X describes the number of run starting from 0
Channel name	`Spectrum_Y`	Y describes the position of the spectra starting from 0

The following list shows which properties are stored in which section.

section	properties
Root entry	`Location`, `Longitude`, `Latitude`, `Hight_asl`, `Zenith`, `Azimuth` , `distance(m)`, `FFT Size`, `Number of shots`, `df`, `Filename`, `Range resolution(m)`, `Time resolution(us)`, `1stRun`, `Target_Start Time(ms)`, `Start Time`
Group name	`Timestamp_device`
Channel name	`wf_increment`, `wf_samples`

9.2.4 Power Spectra data storage

The TDMS format stores the power spectrum as raw accumulated power spectra values (see the Raw Data to Physical Value Conversion) and defers the computation of physical values to the display phase.

9.3 VI List

9.3.1 WIND GettingStarted.vi

This Vi shows step by step how a acquisition can be done.

9.3.2 WindTDMS.llb

WindTDMS NewDataPath.vi

This vi produces a TDMS Filename with current Time information.

- **input**
  - **DataDirectory:** Directory for data files.
- **output**
  - **Path:** Absolute datapath of the current TDMS file.
  - **Filename:** Filename of the current TDMS file.

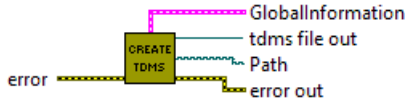
WindTDMS CheckDataDirectory.vi

Check if the data directory is available otherwise prompt user for new one.

- **input**
  - **IniFilePath:** Path of the found initialization file.
  - **DataDirectory:** Data directory where the data file should be created if necessary.
- **output**
  - **DataDirectoryOut:** Data directory where the data file will be created.

WindTDMS Create.vi

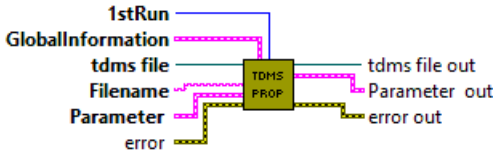
Create a TDMS File with the current time in the filename.



- **output**
  - **Globalinformation:** Cluster combining global measurement information. These settings will be used for the data file headers.
  - **tdms file out:** TDMS file handle.
  - **path:** Absolute datapath of the current TDMS file.

WindTDMS WriteProperty.vi

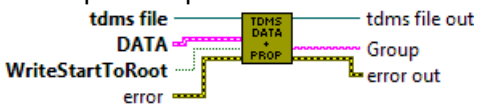
Write the acquisition parameters to a TDMS file.



- **input**
  - **tdms file:** TDMS file handle.
  - **1stRun:** First run written to the file.
  - **GlobalInformation:** Cluster combining global measurement information. These settings will be used for the data file headers.
  - **Filename:** Filename of the current TDMS file.
  - **Parameter:** Cluster contains the command parameter for starting or stopping the acquisition.
- **output**
  - **tdms file out:** TDMS file handle.
  - **Parameter out:** Cluster contains the command parameter for starting or stopping the acquisition.

WindTDMS WriteDataProp.vi

Write the power spectrum and additional time information to a TDMS file



- **input**
  - **tdms file:** TDMS file handle.
  - **DATA:** Cluster contains the data retrieved from RealTime Operating System. Additionally contains the information of the number of runs and the current run.
  - **WriteStartToRoot: True** Write start time information to the TDMS file root entry.
- **output**
  - **tdms file out:** TDMS file handle.
  - **Group:** Generated TDMS group name like `powerspectrum_run0`.

9.3.3 WindTCP_Server.llb

WindTCP_Server StartTCPIP_Server.vi

Launchs a background TCPIP server which is controlled by queue commands.

- **input**
  - **OpenFP: True** Open front panel.
- **output**
  - **reference out:** Vi reference of the TCPIP server.

WindTCP_Server.vi

`WindAcquisition`, `WindLiveDisplay` and `WIND GettingStarted` software use the WindTCP_Server interface for communication with the LicelWind system.

WindTCP_Server PushParser.vi

Seek in the data stream the marker and return a complete spectra array if it is in the buffer. Otherwise move forward to find the next marker.

- **input**
  - **Buffer In:** contains the data that was read out from the controller but could not be parsed due to insufficient data points. This data gets stored here until enough data is read from the controller to generate the complete data sets. `U8 Array`.
  - **data points:** Expected number of spectra elements.
- **output**
  - **Buffer out:** contains the data that was read out from the controller but could not be parsed due to insufficient data points. This data gets stored here until enough data is read from the controller to generate the complete data sets. `U8 Array`.
  - **RawDataPackage:** Contains the marker, the timestamp, the acc power spectra length and the accumulated power spectra as an array of raw `u8` values.
  - **DataComplete:** Indicate whether data is complete or not.

WindTCP_Server SplitRawData.vi

Convert the byte array into a u64 array and get the millisecond timer from the data.

- **input**

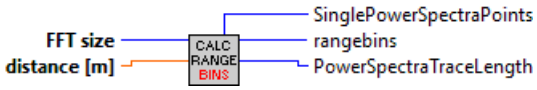
  - **RawDataPackage:** Contains the marker, the timestamp, the acc power spectra length and the Accumulated Power Spectra as an array of raw `u8` values.

- **output**

  - **PushDataPackage:** raw spectra for each range bin there is one FFT and they are following each other, knowing the FFT size this array can be converted to a 2D array.
  - **msec:** Time information of the Package in milliseconds.

WindTCP_Server CalcRangebins.vi

Calculate the powerspectrum tracelength needed for the FFT in the FPGA.
Equation :
Rangebins = round ((distance[m]/150$\frac{m}{\mu s}$ * 0,0025 μs)/FFT size))
PowerspectraTraceLength = Rangebins * FFT size / 2



- **input**

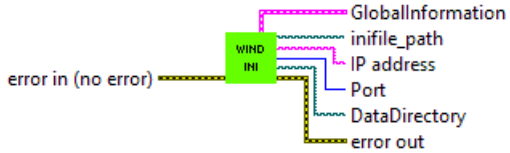
  - **FFT size:** Number of points per spectra. There are 3 opportunities
    1. 64
    2. 128
    3. 256
  - **distance [m]:** max. trace distance in meter.

- **output**

  - **SinglePowerPectraPoints:** Number of points for a single power spectra.
  - **rangebins:** Number of FFT's to be acquired.
  - **PowerspectraTraceLength:** Powerspectrum tracelength of the current measurement. The trace from the FPGA should always be a integer multiple of entire spectras.

WindTCP_Server Ini.vi

Read the configuration values from the ini file.

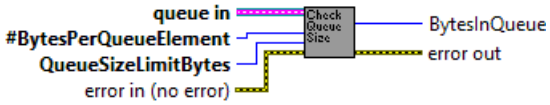


- **output**

  - **GlobalInformation:** Cluster combining global measurement information. These settings will be used for the data file headers.
  - **inifile_path:** Path of the found `Wind.ini` file.
  - **IP address:** IP adress of the LicelWind system.
  - **Port:** Port number of the service with which a connection should be established.
  - **DataDirectory:** Data Directory, where the data file should be created if necessary.

WindTCP_Server CheckQueueSize.vi

Get the number of elements remaining in the queue. The elements will be converted into bytes and compared with the selected queue size limit in bytes. If the limit is exceeded, an error message message will be generated.

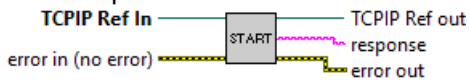


- **input**
  - **queue in:** Queue reference.
  - **#BytesPerQueueElement:** Indicate the number of bytes per queue element.
  - **QueueSizeLimitBytes:** Limit of the queue size in bytes.
- **output**
  - **BytesInQueue:** Indicate the number of bytes remaining in the queue.

9.3.4 driver

WIND_PC Driver StartAcquisition.vi

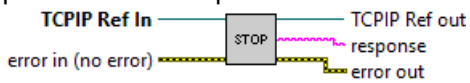
Start an acquisition.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.

WIND_PC Driver StopAcquisition.vi

Stop the current acquisition.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.

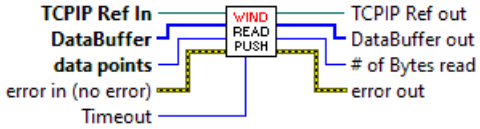
WIND_PC Driver SendParameters.vi

send WIND acquisition parameters via TCPIP..

- **input**
  - **Wind Acq Param:** low level paramters for the LicelWind system.
    - **shots:** number of shots per run.
    - **FFT size:** Number of points per FFT
    - **numFFT:** number of FFTs to be acquired.
    - **runs:** number of runs, each run contains shots acquisitions.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.

WIND_PC Driver ReadPushData.vi

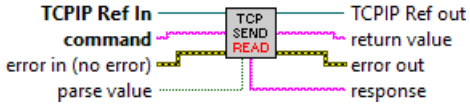
Read entire datasets if possible from the push socket, if this is not possible get as much data as possible.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
  - **DataBuffer:** contains the data that was read out from the controller but could not be parsed due to insufficient data points. This data gets stored here until enough data is read from the controller to generate the complete data sets.
  - **data points:** Expected number of spectra elements.
  - **Timeout:** Timeout value for the TCPIP communication in ms.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **DataBuffer out:** Contains the data that was read out from the controller but could not be parsed due to insufficient data points. This data gets stored here until enough data is read from the controller to generate the complete data sets.
  - **Number of Bytes read:** The number of bytes that were read at the last read.

WIND_PC Driver SendRead.vi

Send TCP Commands and get the response.



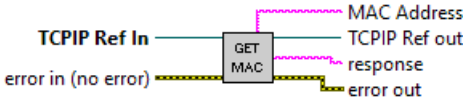
- **input**

  - **TCPIP Ref In:** TCPIP reference for the connection.
  - **command:** String that will be send to the controller (typecasted data).
  - **parse value:** If true the TCPIP response will be shown without checking the String. Otherwise the response will be checked if an executed command was written.

- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **return value:** Return requested parameter.
  - **Response:** TCPIP command response.

WIND_PC Driver GetMAC.vi

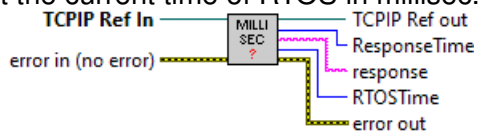
Get the MAC Address of the RTOS target.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.
  - **MAC Address:** MAC Address retrieved from the target Device.

WIND_PC Driver MILLISEC.vi

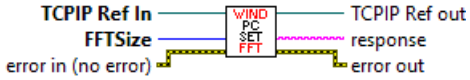
Get the current time of RTOS in millisec.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **ResponseTime:** Time for response the request.
  - **Response:** TCPIP command response.
  - **RTOSTime:** Current RTOS time.

WIND_PC Driver SetFFTSize.vi

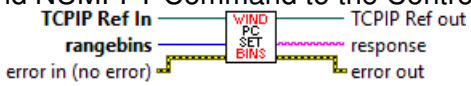
Set the FFT size.
Send FFTSIZE command to the controller.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
  - **FFT size:** Number of points per FFT. There are 3 opportunities
    1. 64
    2. 128
    3. 256
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.

WIND_PC Driver SetNumFFT.vi

Set the rangebin.
Send NUMFFT Command to the Controller.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
  - **rangebins:** number of FFTs to be acquired.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.

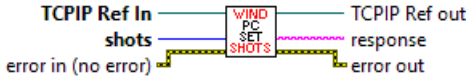
WIND_PC Driver SetRuns.vi

Set the number of runs.
Send RUNS command to the controller.

- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
  - **runs:** number of runs (each run contains a number of shots).
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.

WIND_PC Driver SetShots.vi

Set the number of shots.
Send SHOTS command to the controller.



- **input**
  - **TCPIP Ref In:** TCPIP reference for the connection.
  - **shots:** number of shots per run.
- **output**
  - **TCPIP Ref out:** TCPIP reference for the connection.
  - **Response:** TCPIP command response.