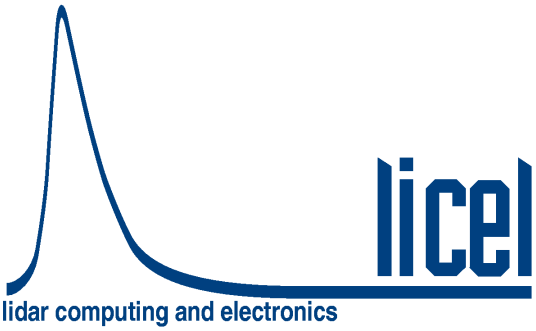


Licel Ethernet Controller – Installation and Reference Manual

Licel GmbH

May 15, 2025

Contents

Chapter 1

Introduction

The ethernet-based control modules for Licel detection systems open the path for truly remote controlled experiments. The Licel Ethernet Control Modules use a TCP/IP based protocol with a syntax similar to conventional GPIB based instruments. Each module use an ASCII command set with a structure similar to SCPI in order to be compatible with common measurement devices. A LabVIEW driver library for easy integration is supplied. The modules can either be operated using a static IP address or a dynamically assigned IP address (DHCP).
In the next chapter the control modules currently available at Licel are introduced. Then the installation of the software and setup of the network is described. The following chapter gives an introduction to the acquisition software. After that acquisition tutorial further software control modules are described. Finally the appendix contains information about the TCP/IP command set, the file format, initialization files, and the LabVIEW TCP/IP driver library.
The most up to date version of this manual can be found at https://www.licel.com/software.htm.

Chapter 2

Licel Control Modules

Currently several modules are available to control transient recorders, APDs, PMTs, the trigger timing/gating and the recording of laser power. Two integrated packages to make polarization sensitive measurements and to remotely control the detector alignment extend the family of Ethernet controllers. The control modules are equipped with a twisted pair ethernet connector which allows for 10/100 Mbit network based access.

2.1 The Transient Recorder Control Module

The Licel transient recorder control module can control up to 16 transient recorders. It translates the ASCII based commands received via TCP/IP into low level transient recorder commands. The data from the transient recorders is then sent back to the PC. This eliminates the need of a special interface card to control the transient recorder. The typical data transfer rate is 200 kb/sec. This is lower than for a PCI-DIO-32HS but offers a cost sensitive solution for small systems.
The transient recorder ethernet control module introduces a new data transfer mode: the push mode. In the push mode the transient recorders get their start, stop, and readout commands from the ethernet controller without any direct interaction with the PC. The ethernet controller then pushes the data to the PC. At the PC level, a periodic task reads the data when it becomes available from the TCP/IP buffer. This frees the PC from controlling the transient recorders by itself and reduces the communication load. The advantages of the push mode are important for single shot acquisitions.

2.2 The Photomultiplier High Voltage Control Module

The Licel photomultiplier high voltage control module can control up to 8 PMT modules. The control voltage ranges between 0 and 1V, which is generated by a precision DAC and monitored. PMTs with an activated high voltage are indicated by a LED.

2.3 The APD High Voltage Control Module

The Licel APD high voltage control module can control up to 4 APD modules. The control voltage ranges between 0 and 1.8V, (which corresponds to 0 to 450V APD HV), which is generated by a precision DAC and monitored. APDs with an activated high voltage are indicated by a LED (1-4). The thermoelectrical cooler can also be remotely activated. Once a stable temperature is reached the T_{Set} LED is activated.

2.4 The Licel Trigger Module

The Licel Trigger Module incorporates one trigger input and 4 different outputs to build up compact detection systems. The trigger input can be used to synchronize the system to an external laser flash lamp or Q-switch trigger. The module can also run internally triggered. The Licel Trigger Module consists of a timing sub-board which is able to generate:

- a lamp trigger
- a pretrigger for the transient recorder
- a Q-Switch trigger
- and a Gating trigger for gated PMT-Modules.

All timings are derived from a quartz based oscillator ensuring nanosecond timing stability.
Optionally the Licel Trigger Module can be extended with more timing sub-boards. Then, each of the sub-boards is able to generate the trigger outputs listed above. When programming such a multi-board Trigger Module the parameter `BoardID` in the TRIGGERMODE and TRIGGERTIME must be used.

2.5 The Licel Power Meter Controller

Monitoring every laser shot is the optimum approach to detect laser pulse energy fluctuations, SHG and THG efficiency changes and flashlamp degradation of your laser . A laser spot reflection can be measured using a photodiode or a laser power meter head. The photo diode signal or the output of the power meter head is directly analyzed at the Power Meter Controller.
After an external trigger is received a short trace from detector will be analyzed. The light pulse is measured and the result together with a time stamp is sent to the PC over a Ethernet connection. A trace mode is available to inspect a single pulse.
The most recent Power Meter Controller can be delivered with up to three detector inputs.

2.6 The Licel Polarotor

The Licel Polarotor adds (de-)polarization measurements to multispectral detection systems. A rotating stepper motor driven Glan Thompson prism is used to separate p- and s-polarized signal contributions. The integrated polarotor trigger generator synchronizes

- a lamp trigger
- a Q-Switch trigger
- a pretrigger for the transient recorder at the s polarization detection chain
- and a pretrigger for the transient recorder at the p polarization detection chain.

A laser repetition rate of up to 50 Hz is supported. All timings are derived from a quartz based oscillator ensuring nanosecond timing stability.
Like the Trigger Module the Licel Polarotor can optionally be extended with additional timing sub-boards. When programming those sub-boards the `BoardID` in the TRIGGERMODE and TRIGGERTIME must be used.

2.7 The Licel Bore Sight Alignment Controller

In many LIDAR applications, daylight is among the major limiting parameters for the achievable signal range. By implementing a continuous monitoring and correction of the alignment, the telescope field of view can be reduced close to the laser beam divergence. This can improve operation of narrow field of view Raman or micropulse lidars and unattended operation.
The LICEL Bore sight alignment controller evaluates the image of the laser return from two user defined height ranges on a multi anode photomultiplier. It computes correction parameters for the beam steering.

Chapter 3

Software Installation

Licel provides a package of software modules for setting up the Licel Ethernet 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 (beginning with version 2016) 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 (7|8|10) computer. Licel provides the software on a CD ROM and for download (https://www.licel.com/software.htm).
It is not necessary to install both, the LabVIEW sources and the Windows applications but one will at least need one of these pachages. In the next section *Preparation* some required steps to prepare for the installation are discussed. The following section *The Licel CD ROM* describes the content of Licel's CD ROM. Alternatively you may download the most recent software as described in the section *Download*. Finally, the installation is described in the sections *3.4 Installing the Windows Applications* and *3.5 Installing the LabVIEW sources*.

3.1 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...) or onto a CD ROM.

LabVIEW Users

If you have used older versions of Licel LabVIEW sources 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...) or burning them onto a CD ROM.
2. Scan your disks to find all versions of the following directories (or files with similar names if you migrate from an older version of Licel's Acquisition Software) and delete them once you have made backups of them. Delete all other `LLB` files.

```
1. project\LicelTCPIP_src.lvproj
2. Files\user.lib\*.*
3. source\AdvancedViewer.llb
```

4. source\ControlAPD-PMT.llb
5. source\ControlTiming.llb
6. source\Datafile.llb
7. source\DriveControl.llb
8. source\LicelAcquis.llb
9. source\LicelFile.llb
10. source\LicelGraph.llb
11. source\LicelMaintenance.llb
12. source\LicelModule.llb
13. source\LicelTCPIP.llb
14. source\LicelTCPIP_API.llb
15. source\LicelUtil.llb
16. source\M-Acquis.llb
17. source\Postan.llb
18. source\PowerMeter.llb
19. source\SearchControllers.llb
20. source\TCPIP_Acquis.llb
21. source\TCPIP_LiveDisplay.llb
22. source\TCPIP_LiveSquare.llb
23. source\TCPIP_MPushAcquis.llb
24. source\TCPIP_Pulse.llb
25. source\TCPIP_Track.llb
26. source\Gating Control.vi
27. source\Licel Main.vi
28. source\Licel Main-M.vi
29. source\Multi Power Meter Control.vi
30. source\Polarotor.vi

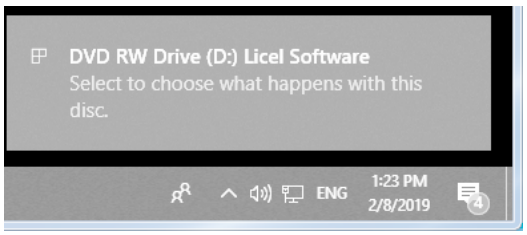
Please note: Licel may have provided individual software solutions with additional or less directories and/or LabVIEW library files than noted in the list above.

3. Search the directory your older version of Licel LabVIEW sources 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
   - `source` a directory containing the above listed LaVIEW LLBs and VIs and initialization files
   - `project` a directory containing the LabVIEW project `LicelTCPIP_src.lvproj`
   - `Files\user.lib\errors\Licel-errors.txt` Licel error code file

3.2 The Licel CD ROM

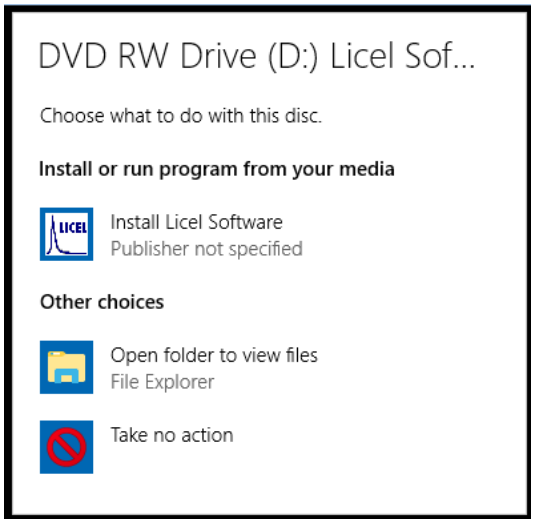
The standard CD ROM provided by Licel contains both, the LabVIEW sources and the Windows Installer for installing the Windows applications, and furthermore a documentation folder. Licel may add customer specific components on the CD ROM.

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.



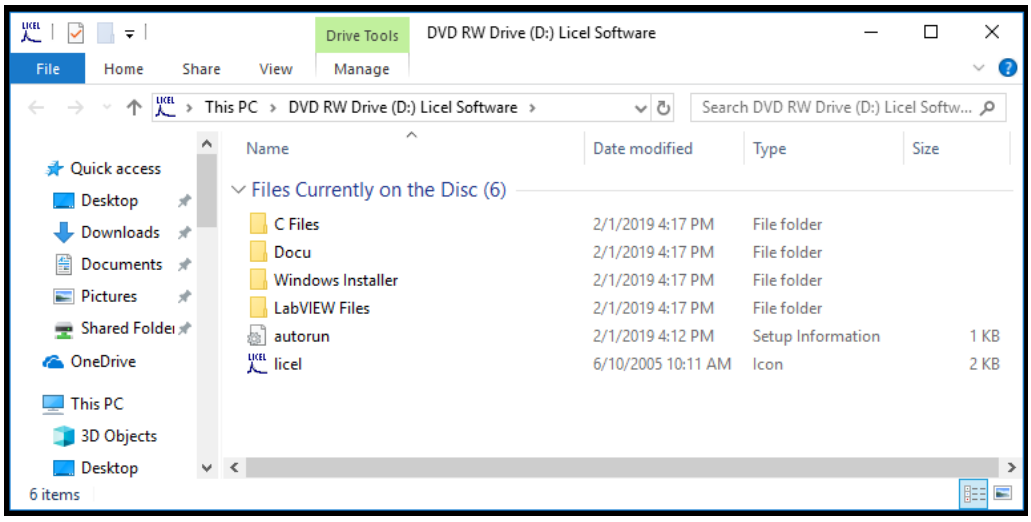
Please click on the pop-up message.

3. The following selection dialog should appear:

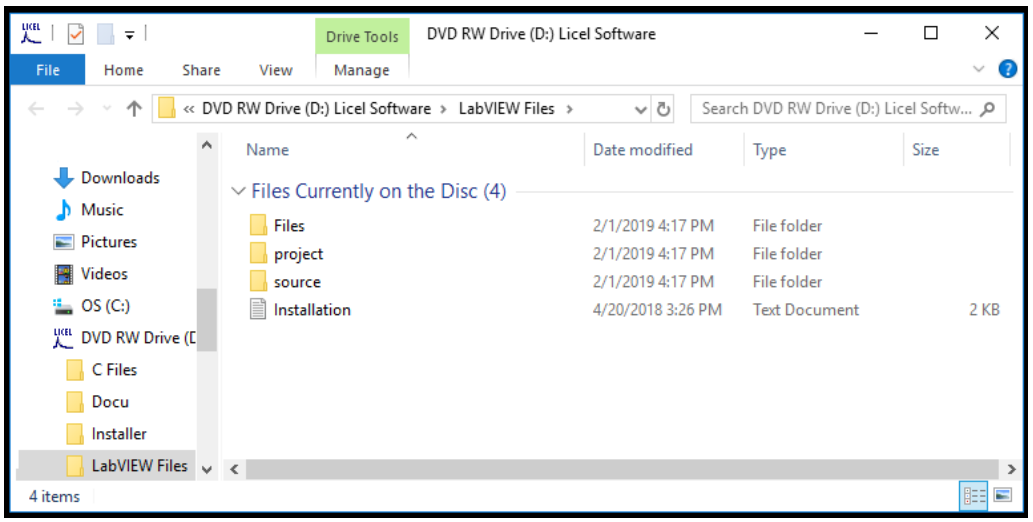


In older Windows operation systems a similar dialog will automatically come up.

- 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 3.4.
- Press *Open folder to view files* to start the File Explorer (Windows Explorer) to see the content of the CD:



  - The folder *LabVIEW files* contains the LabVIEW sources. Please copy them from there into a directory of your choice on your local PC.

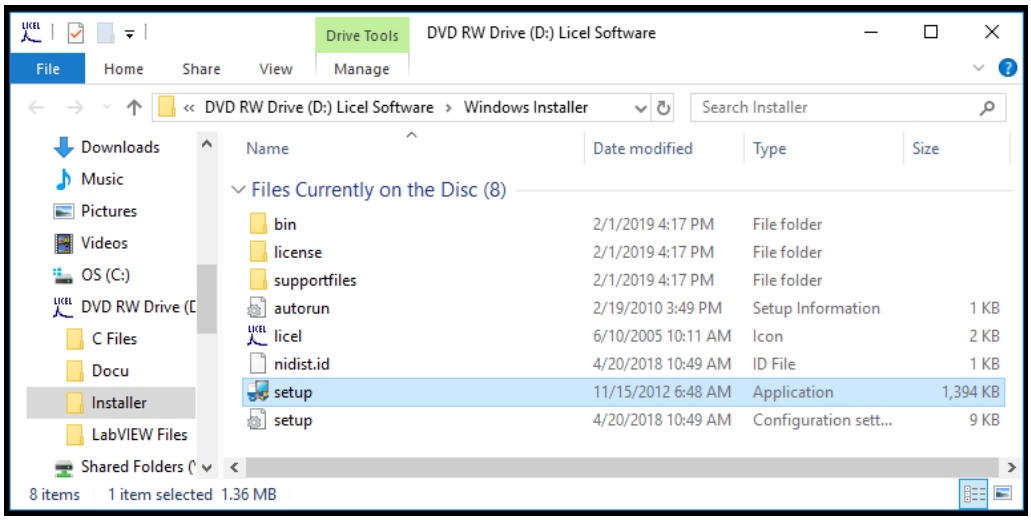


Please note the remarks according to existing LabVIEW library files. Please refer to the section 3.5 for further details.

– 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 3.4 afterwards.

3.3 Download

The Licel software is frequently maintained. The most recent version is available on the download page (https://www.licel.com/software.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, `TREthernet.zip` contains the LabVIEW sources, while `LVInstaller.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 (`LVInstaller.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 3.4.

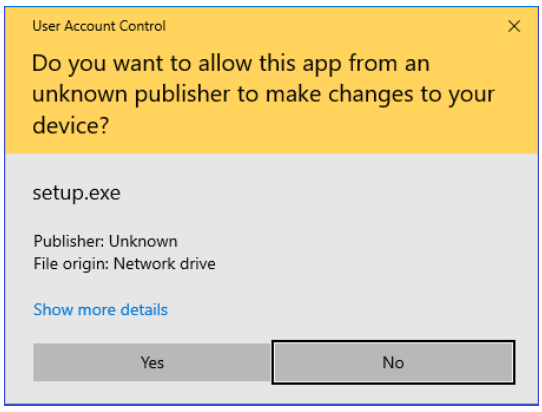
Unpacking the LabVIEW Sources

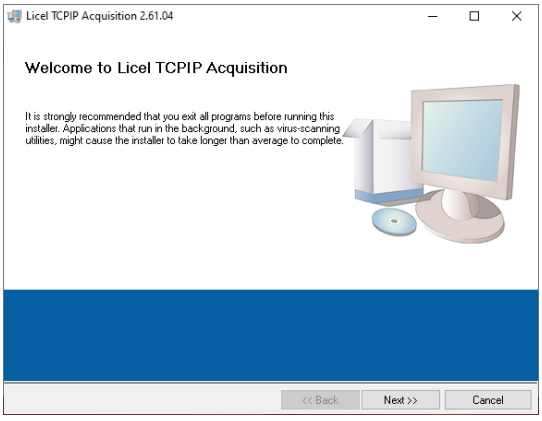
The Licel LabVIEW sources and initialization files contained in the zip file `TREthernet.zip` may directly be unzipped to a destination folder of your choice. Please note the remarks according to existing LabVIEW library files. Please refer to the section 3.5 for further details.

3.4 Installing the Windows Applications

This subsection describes the installation of the Licel Windows applications. To operate the Licel Windows applications the LabVIEW Runtime Engine needs to be installed, as well. The Windows applications together with the LabVIEW Runtime Engine come as a Windows Installer package. For the installation of the LabVIEW applications administrator rights are required. If you plan to work with the LabVIEW sources within a LabVIEW installation, only, you may skip this section.
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.

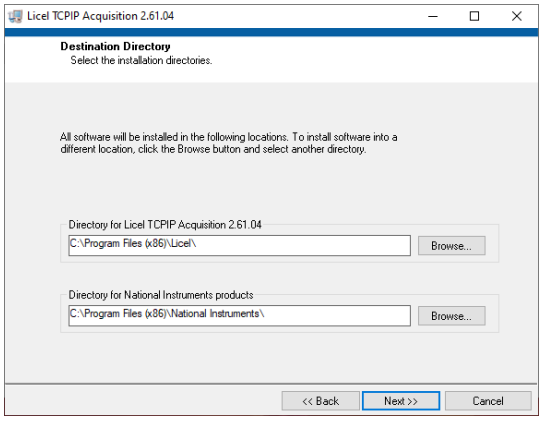
1. At the very first start of the installation the User Account Control dialog may appear. Click **Yes** to continue the installation process.

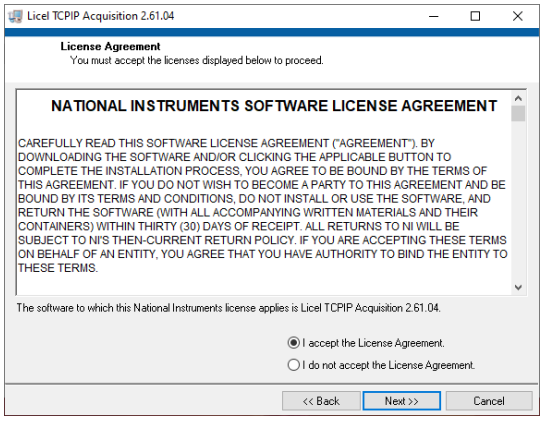




   Afterwards, the Installation will be initialized and a welcome window will be shown. Press the **Next** button to continue.

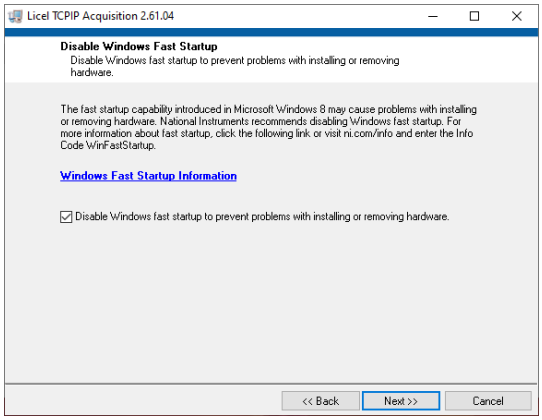
2. At the destination directories window you may change the installation directories using the **Browse** button. Click **Next** to proceed.

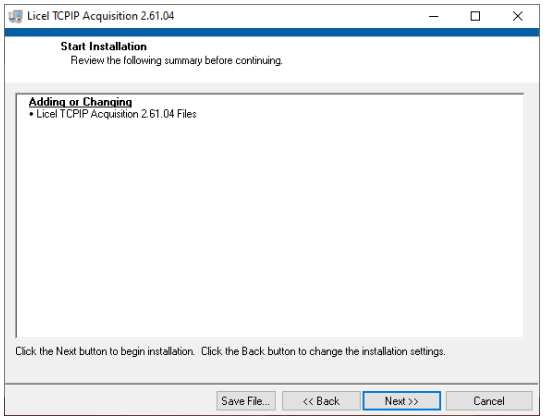




In the next dialog you have to accept the License Agreement(s). Choose *I accept the License Agreement* and **Next** to proceed.

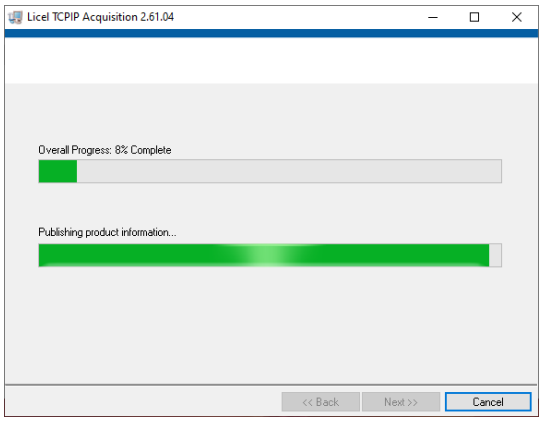
3. In the following dialog you can coose to check the mark *Disable Windows fast startup...* and click the **Next** button to proceed.

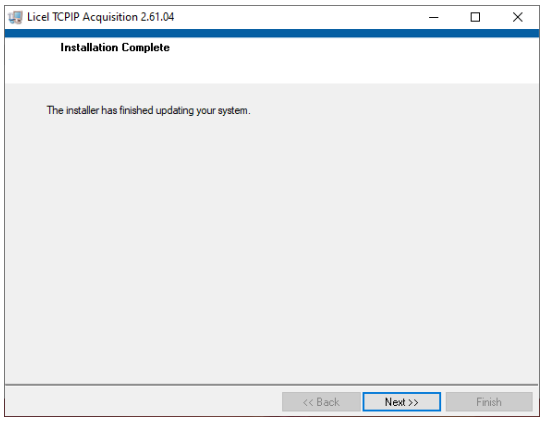




Confirm the following dialog using the **Next** button or click **back** to change your installation settings.

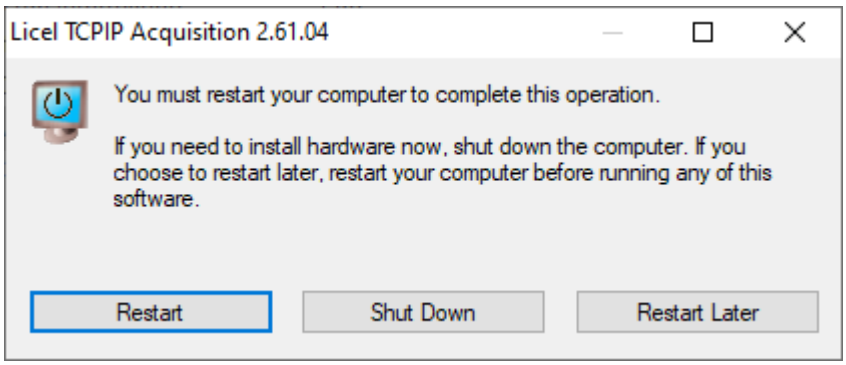
4. After the installation has started the progress will be indicated by a progress bar.





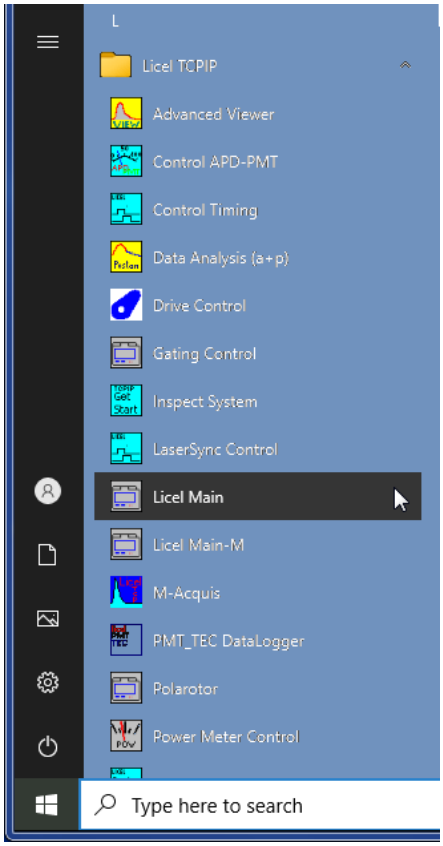
After the installation process is completed an information window will be shown. Click **Next** to proceed.

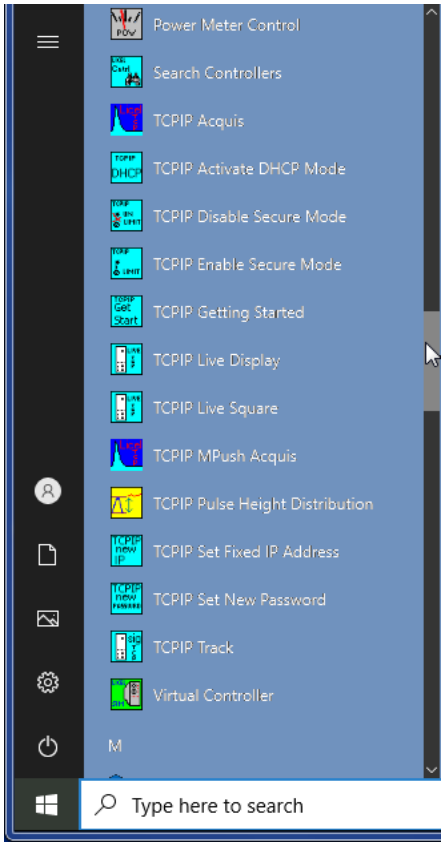
5. To finish the installation you may have to restart your computer. If a restart is required click **Restart** to complete the installation.



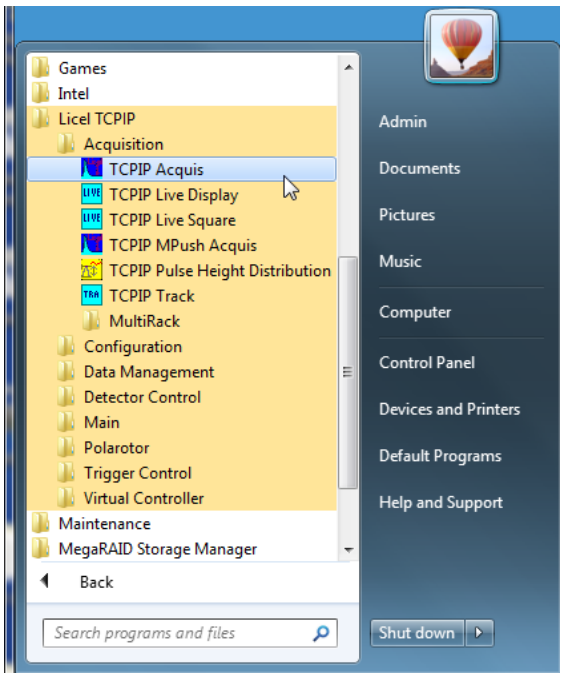
6. After the installation has successfully been completed you are able to start the Windows applications through the corresponding folder *Licel TCPIP* in the Windows start menu. The appearance of the start menu depends on your operating system.

   - Windows 10 To open the folder go to the Windows start menu and navigate to the Letter *L*. There, you will find the folder *Licel* with the links to the applications (left). Please scroll down to see more applications (right).





   - Windows 7 You will find the link to the installed applications in the sub folders of the folder *Licel TCPIP* in the *Programs* section of the Windows start menu.



7. 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).

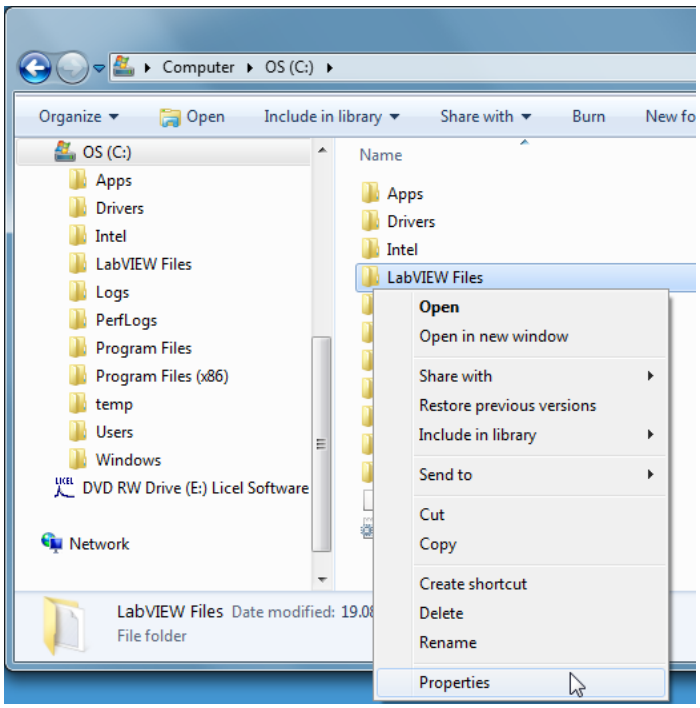
3.5 Installing the Licel LabVIEW Sources

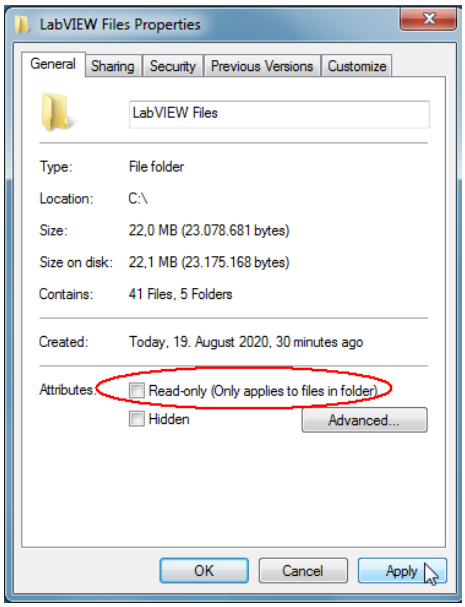
This section describes how to install Licel's LabVIEW Sources. If you are using the *Windows Applications* and don't have LabVIEW or don't want to modify the programs you may skip this section.
To install the Licel LabVIEW sources you may choose between the following options:

- Copy all files contained in the directory `LabVIEW Files` from the CD ROM to a directory of your choice.
- If you downloaded the Licel software from https://www.licel.com/software.htm please unpack the content from the downloaded zip file and copy it to a directory of your choice (keep all directory hierarchies!).

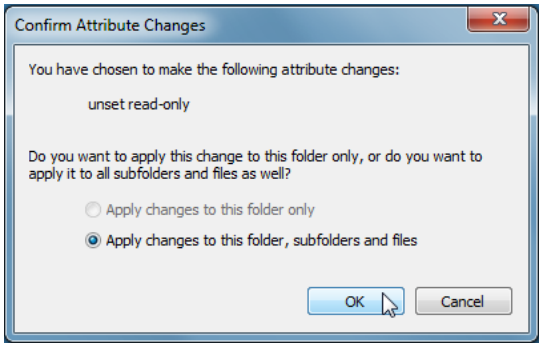
Please note that in the case the software is copied from a CD you may have to unselect the *"Read-only"* attribute for the destination folder.

1. This is done by selecting the directory and right-clicking on it. Select **Properties** from the context menu.



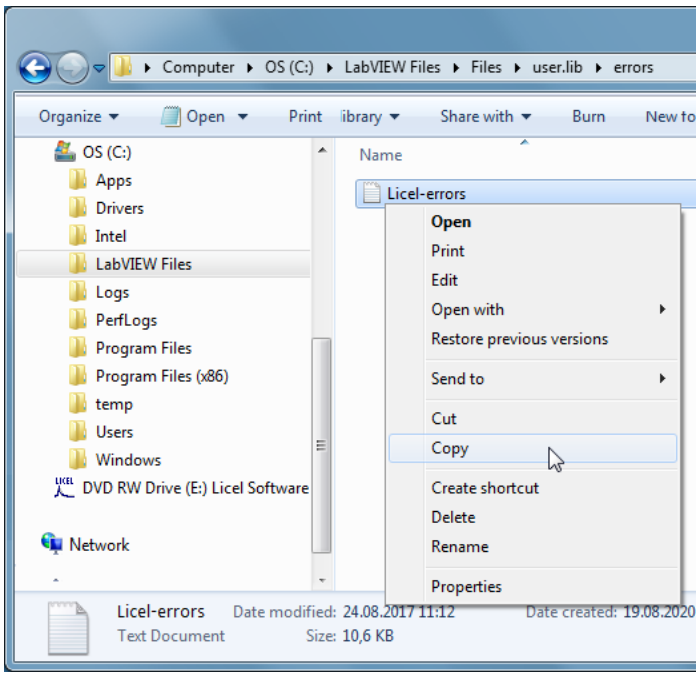


2. Verify that the "`Read-only`" attribute is not checked, uncheck it if necessary. Click *OK* and check in the next dialog *Apply changes to this folder, subfolders and files*. Leave the dialog by clicking *OK*

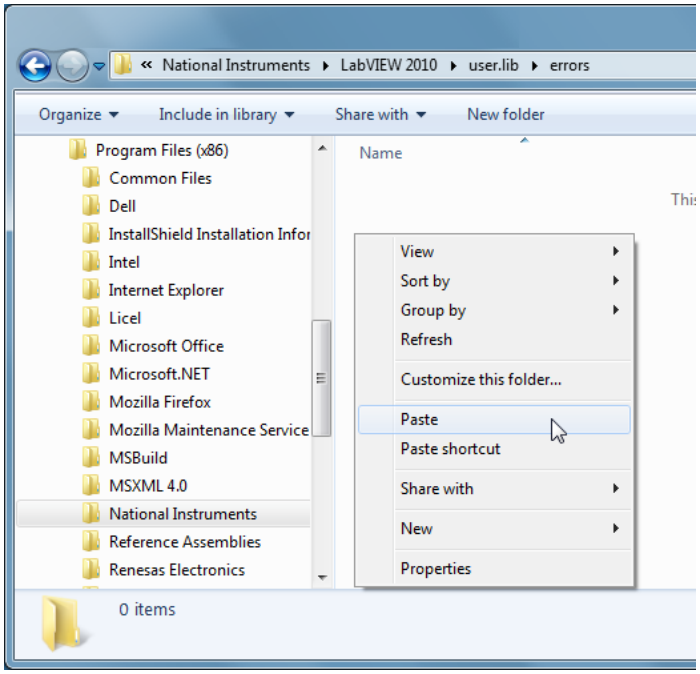


Licel provides one or more user-defined LabVIEW error code files. LabVIEW will use these files to generate hints in error messages. Before you will have to copy these error code files to an appropriate location where LabVIEW will find them. For this

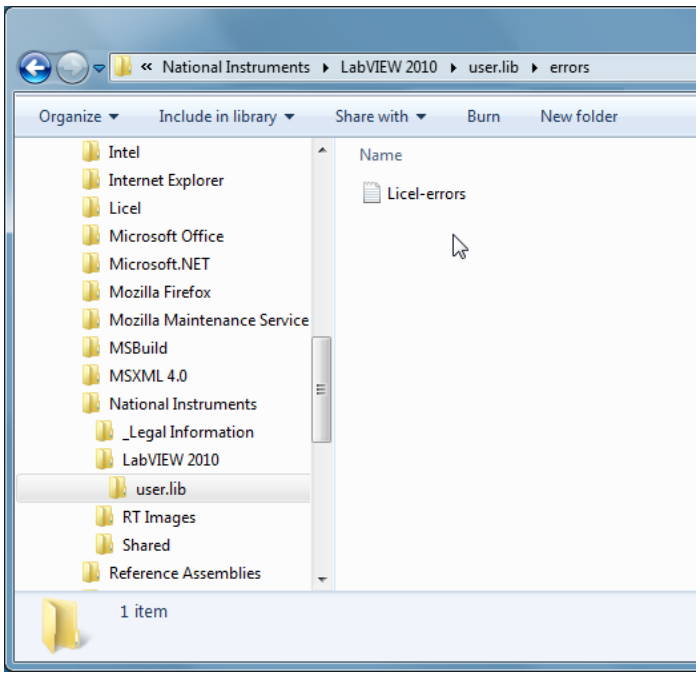
1. Locate the error code files in Licel's LabVIEW sources: they are located in the sub folder `<LabVIEW Files Folder>\Files\user.lib\errors`
2. Select all files `*-errors.txt` and copy them



3. Navigate to the folder
   `<LabVIEW installation directory>\user.lib\errors`,
   create the sub folder `errors` if necessary. Paste the copied file(s) to that directory

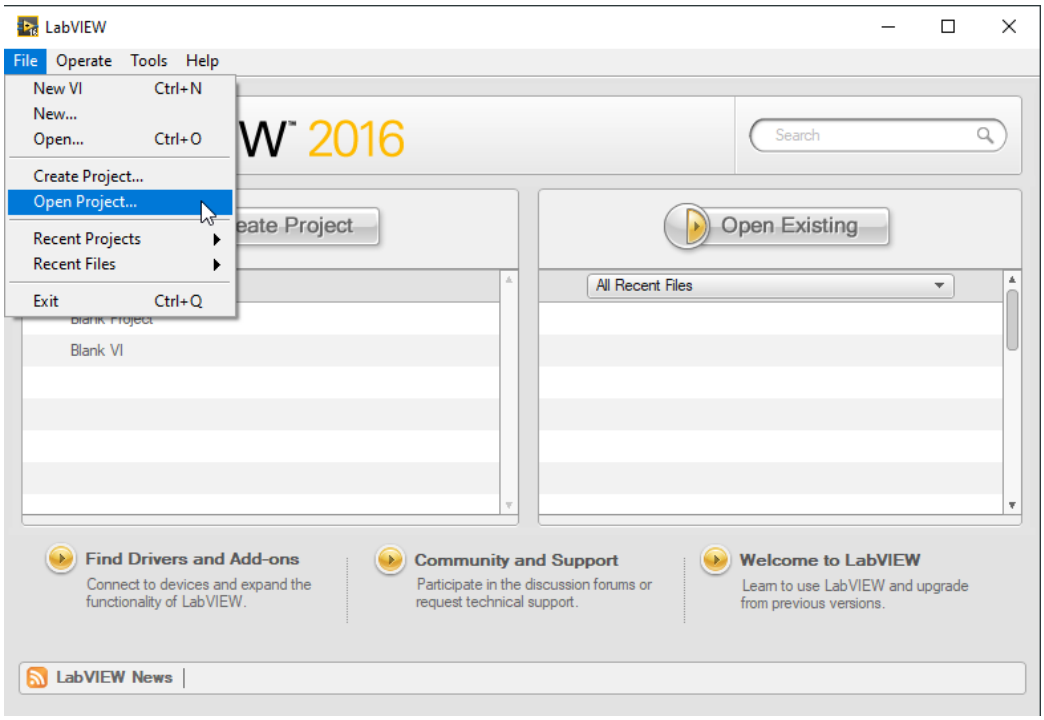


4. All copied LabVIEW error code files should be seen now:

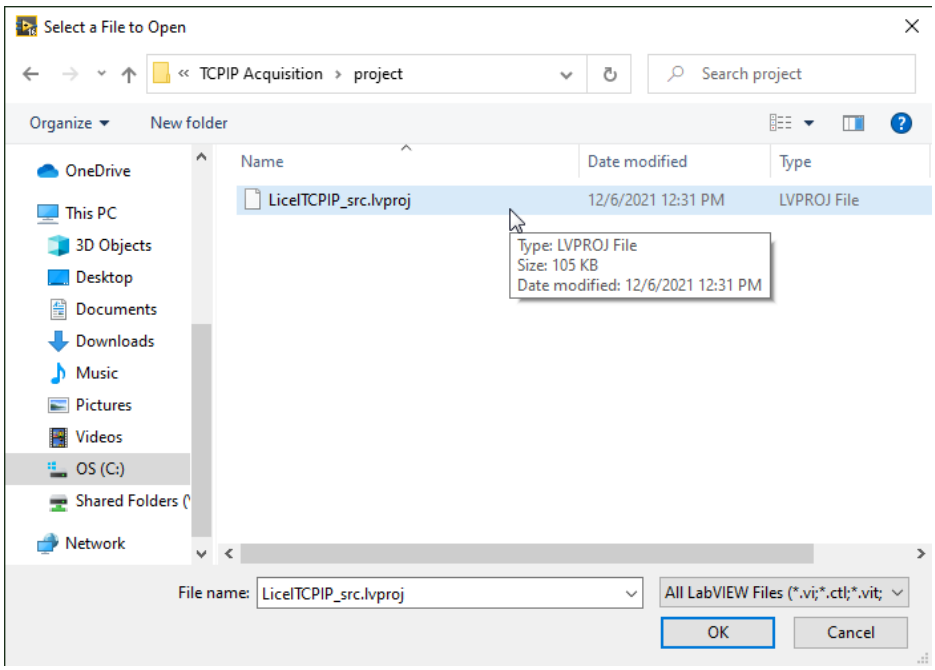


Now you should be able to run all the files. If you are still having problems, apply a mass compile to the directory where the software was extracted to:

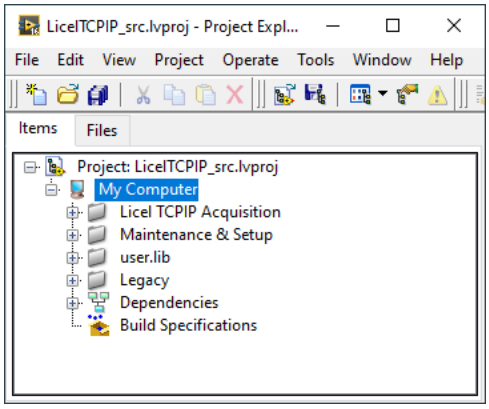
1. Start LabVIEW.
2. Select the menu entry *Open Project...* in the *File* menu



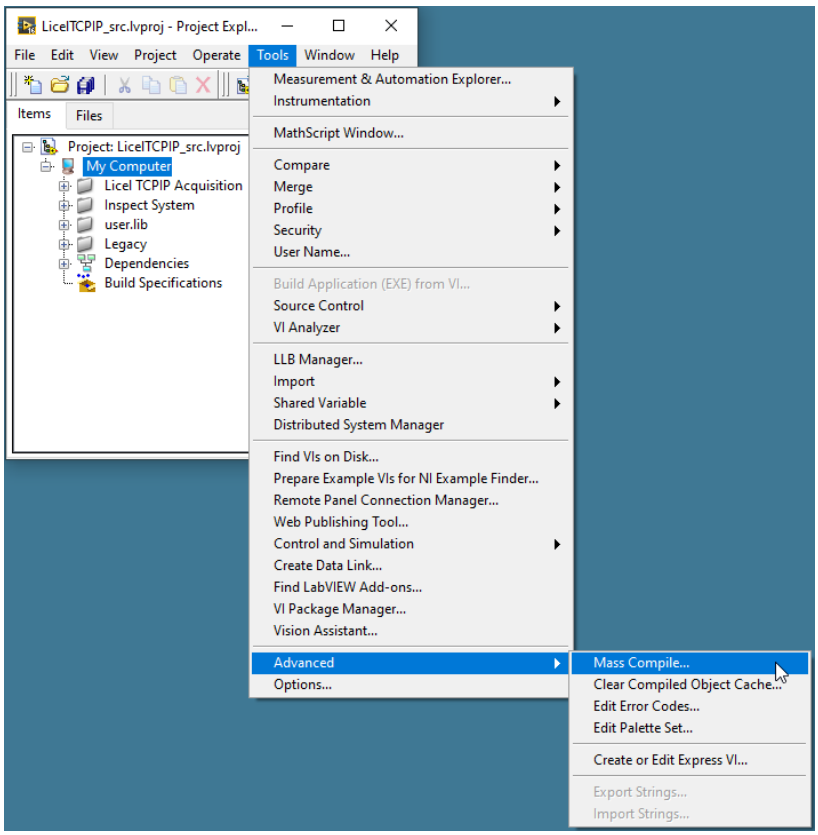
3. Navigate to the folder where you copied the Licel LabVIEW sources to



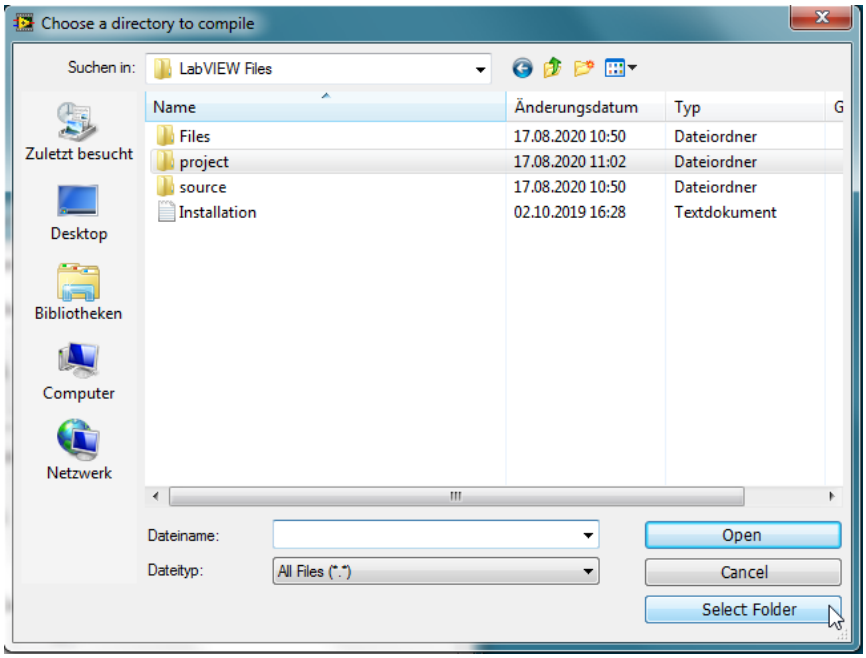
4. Open the project `LicelTCPIP_src.lvproj` in the subfolder `project`



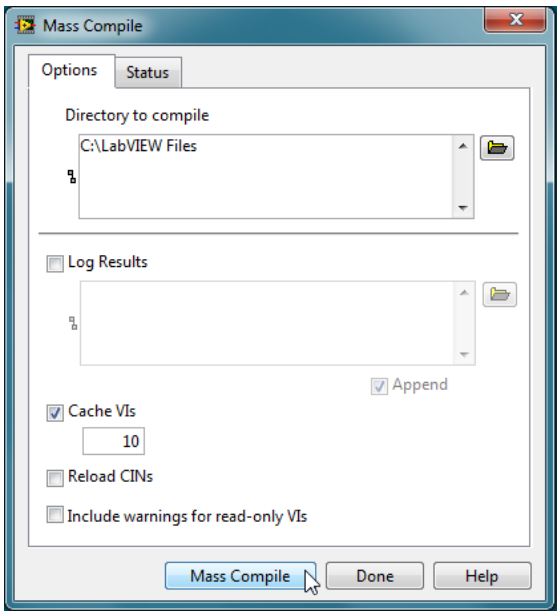
5. Select the menu *Tools*, then *Advanced*, and finally *Mass Compile...*.

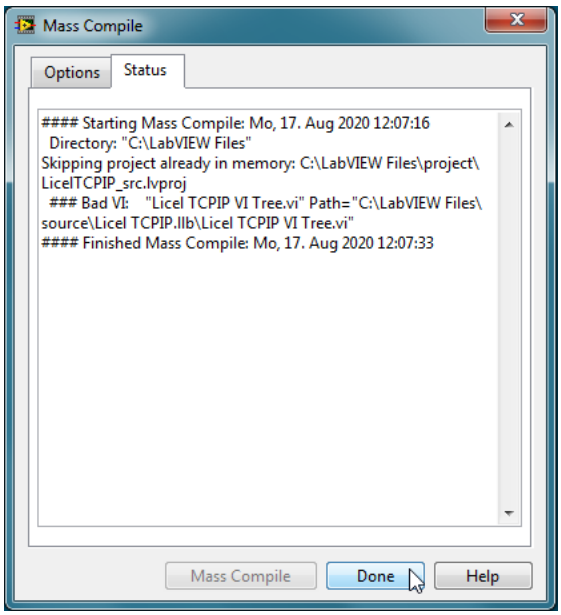


6. You will be asked to select a folder, select the target directory of the LabVIEW source files.



7. Press *Mass Compile* in the next dialog.





8. Later the mass compile status will be shown. The LabVIEW project `LicelTCPIP_src.lvproj` will be indicated as skipped because it has already been loaded. Please ignore that the VI `Licel TCPIP VI Tree` is marked as a ”Bad VI”.

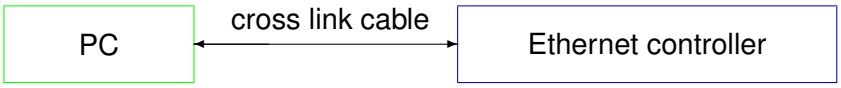
Please note that the removal of older libraries is a necessity, since LabVIEW often links to various libraries with the same name. As a result, if a library is installed twice, one can not be certain which library is actually being used.
If you still have any problems, please contact Licel for further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

Chapter 4

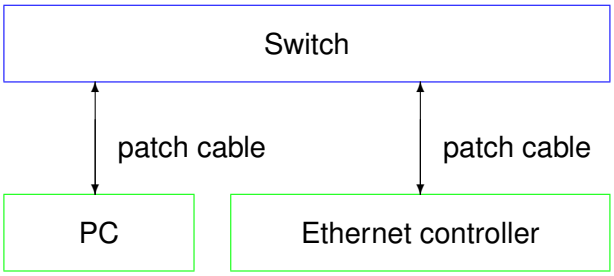
Setting up the Network

4.1 Network Introduction

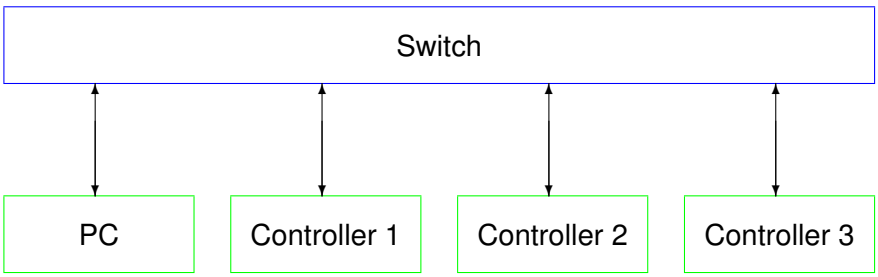
To control a Licel Ethernet Controller a working TCPIP connection is required. This can be reached by two ways, using a cross link cable, which creates a one to one connection between the PC and the Ethernet Controller or with patch cables and a switch



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. Never use IP addresses beginning with `169.254.` because the corresponding IP address range `169.254.x.y` is reserved as the DHCP fallback range for network clients that try to connect to a DHCP server but do not find any in the local network.

4.2 Preparations

To operate the Licel Ethernet 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.
5. Test the TCP/IP connection from your PC to the Licel Ethernet Controller.

4.3 Network Information

The Licel Ethernet Controller is shipped with a default static IP address. The default parameters are:

IP address	10.49.234.234
network mask	255.255.255.0
gateway	
port	2055

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 (4.5.2) to enable the Licel Ethernet 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. **Please make sure that the IP address is unique in your network.** If you have more than Licel Ethernet Controller make sure that they use different addresses. Refer to the subsection Fixed IP Address (4.5.1). The system ships with all Ethernet controllers set to the default address of `10.49.234.234`. In order to setup a system with multiple controllers one needs to do this procedure with each controller in sequence where only one controller is connected to the network at a time. Otherwise one would end up with multiple controllers sharing the same default address which would prevent a successful setup procedure. Never use a fixed IP address beginning with `169.254.` because the corresponding IP address range `169.254.x.y` is reserved as the DHCP fallback range for network clients that try to connect to a DHCP server but do not find any in the local network.

2. The default ports used by the Ethernet controller are 2055 and 2056. Can these ports be used? If you have more than one Licel Ethernet controller the addresses should be different but the ports can be identical for them.

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?

4. Is the default network mask `255.255.255.0` suitable for the communication bewtween the PC and the Licel Ethernet Controller? A "`255`" at the first n positions of the network mask mean that the first n numbers of the IP addresses of both, the PC and the Licel Ethernet Controller, at the corresponding positions must be the same.

4.4 Network Preparation

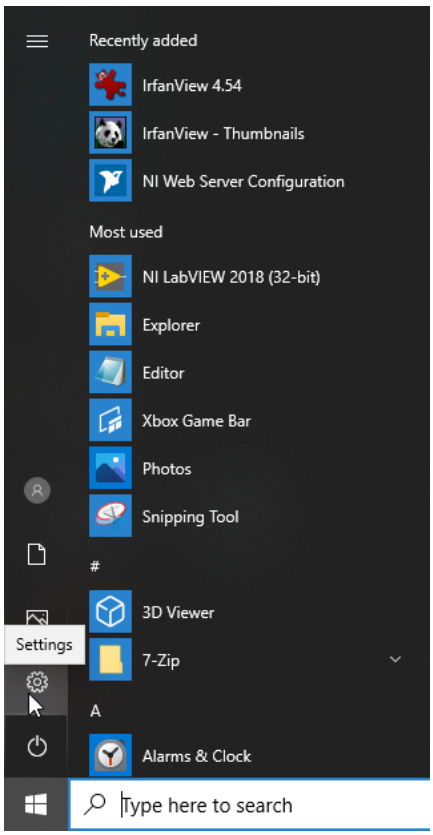
After having installed the Licel Windows applications or the Licel LabVIEW modules on your PC you are ready to change the network configuration parameters of the Licel Ethernet Controller according to the local network settings described in the previous section.

4.4.1 Establish the Connection

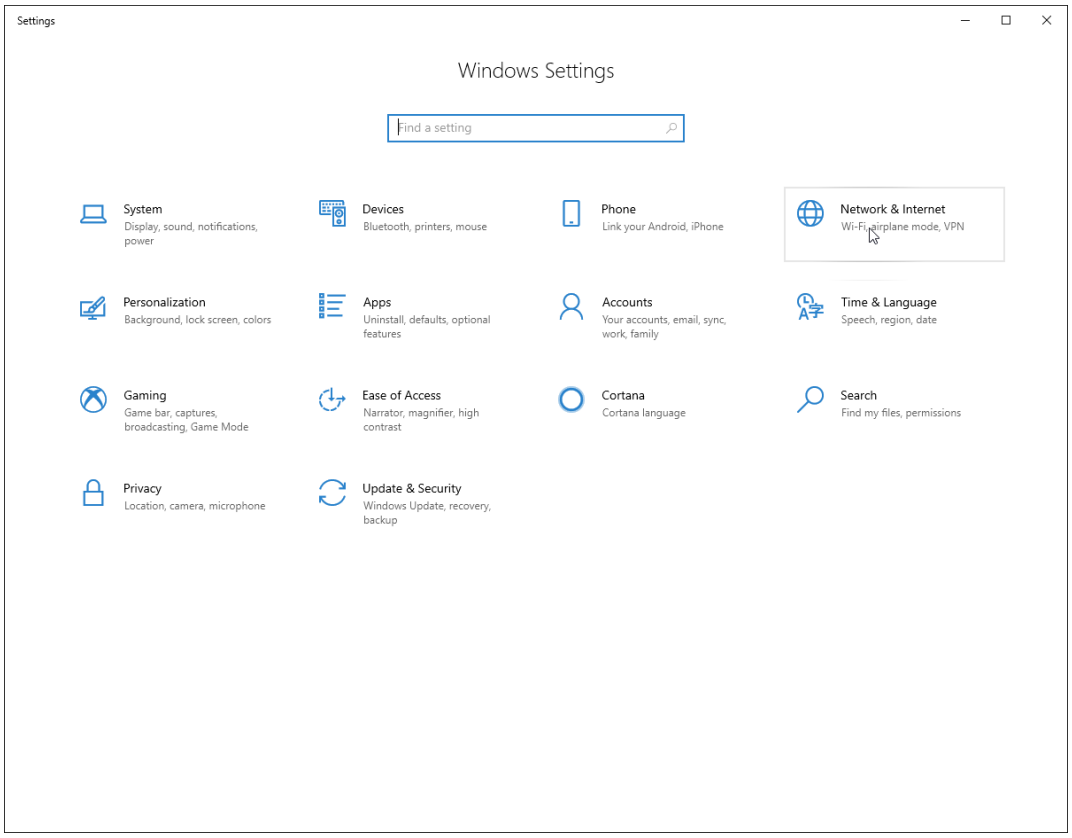
A straight-forward way to do this is the following procedure. You will need local administrator rights on your PC for the following steps:

1. 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:

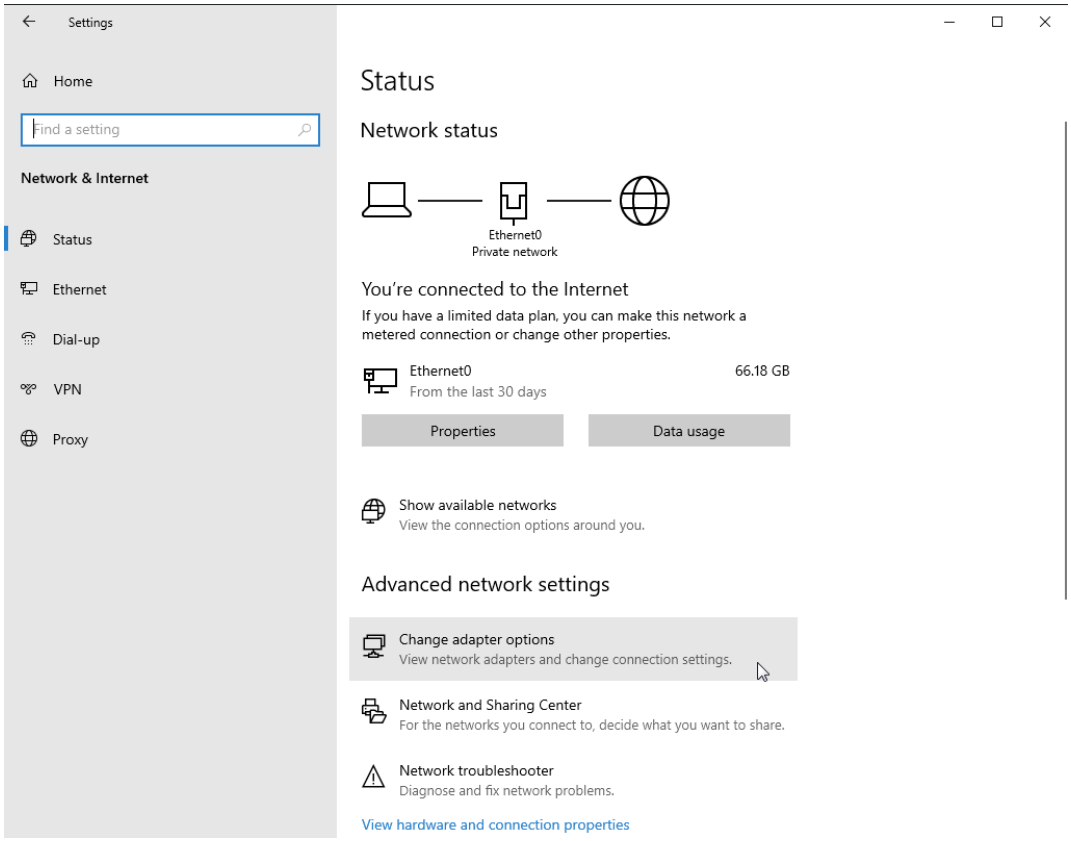
   (a) Click on the  button, and then on *Control Panel*.



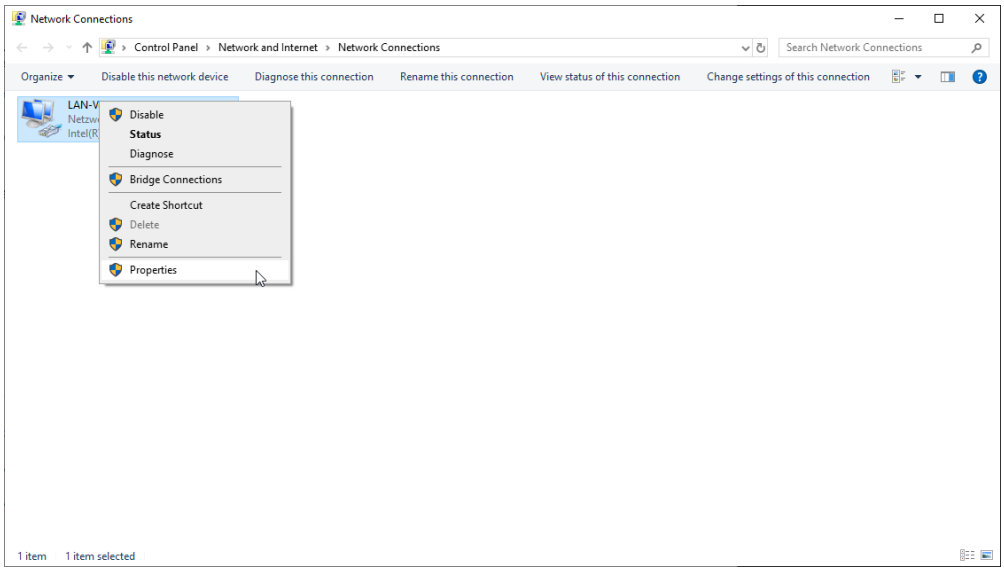
(b) Once the control panel has come up click on *Network and Internet*.



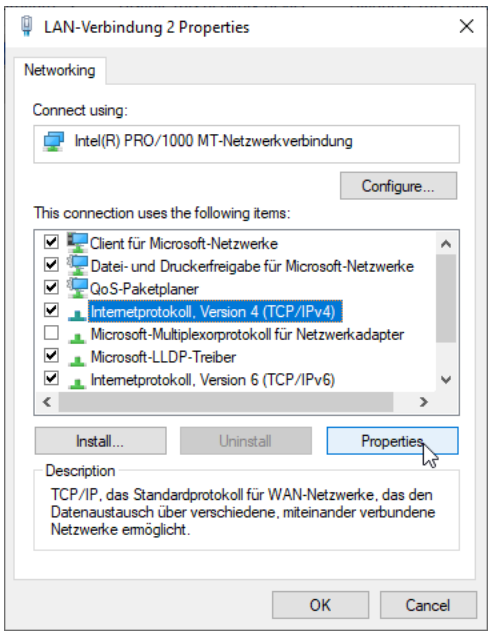
(c) In the next window click on *Change adapter options* in the *Advanced network settings* section.



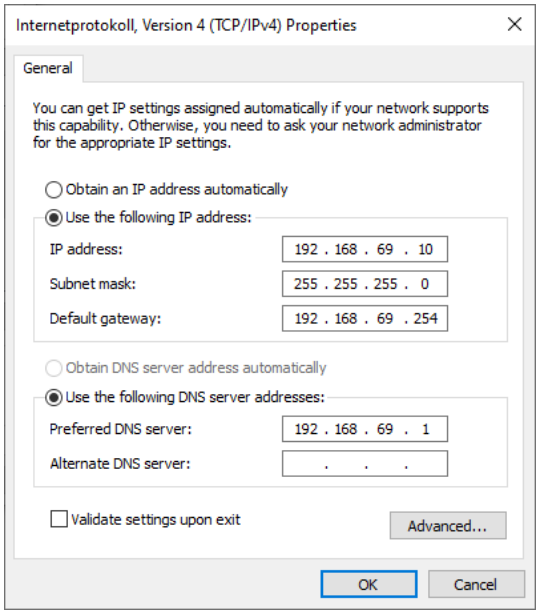
(d) The installed network connections will be shown, right-click on the local Ethernet connection to be used with the Licel Ethernet Controller and choose **Properties** from the context menu.



2. 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.



3. 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. Please note that the addresses and settings given here are examples only!



   (a) If *Obtain an IP address automatically* is active on the Ethernet adapter you will use for the communication with your Licel Ethernet Controller you will work in a configuration where both, your PC and your Licel Ethernet Controller will finally connect to your local network via a switch. In that case you should find out information about the PC's current IP address (range) to be able to assign a fixed IP address to the Licel Ethernet Controller later:

      i. Open a command prompt window (DOS box).

      ii. 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. . . . . . . . . : 192.168.69.10
        Subnet Mask . . . . . . . . : 255.255.255.0
        Standard-Gateway. . . . . . :
```

iii. If the shown IP address begins with `169.254.` there is a network problem: your PC attemps to connect to a DHCP server but does not find any. Therefore, an IP address in the reserved DHCP fallback range `169.254.x.y` is assigned. Please fix your netwirk problem before you continue.

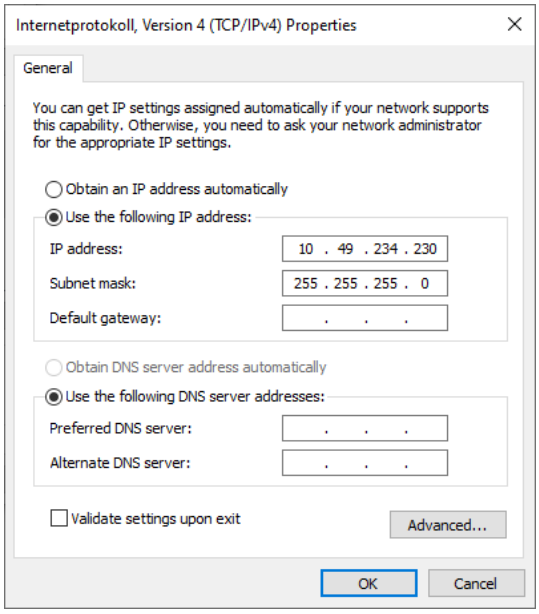
iv. If not remember the shown IP address (in the example: `192.168.69.10`).

(b) if *Obtain an IP address automatically* is not checked, note the *IP address*, the *Subnet mask*, and the *Default gateway*.

(c) Remember the checkbox *Obtain DNS server address automatically*, and

(d) if *Obtain DNS server address automatically* is not checked, note the *DNC server* addresses if available.

4. If activated disable DHCP (checkbox *Obtain an IP address automatically*) and manually assign an IP address within the default address range of the Licel Ethernet Controller. A good choice would be `10.49.234.230`. **Never use the default address (`10.49.234.234`) of the Licel Ethernet Controller as IP address for your PC.**



5. Quit the dialog by pressing *OK*.

6. Power up the rack with the Licel Ethernet Controller and connect the PC with the controller using the red **cross-link cable** shipped together with your hardware.

Now you should be able to access the Licel Ethernet Controller via your Ethernet adapter. Please test this first connection with the methods given in the next section.

4.4.2 Diagnostics

Please carry out the following steps to verify that the connection of the Licel Ethernet 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).

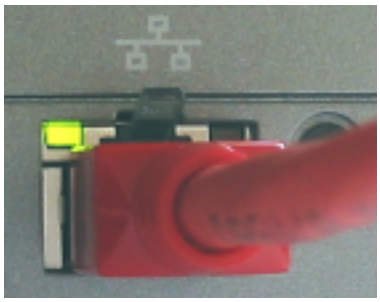
(b) Type `ipconfig` and press enter. At least one of the Ethernet adapters should show the address that you previously set (`10.49.234.230`). The response should be similar to the following:

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

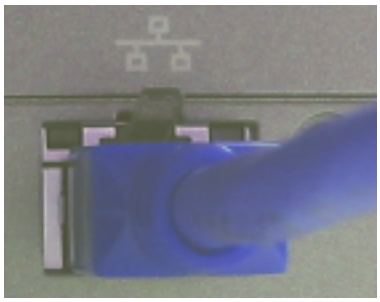
4. Verify that the Licel Ethernet Controller is accessible via the network now:

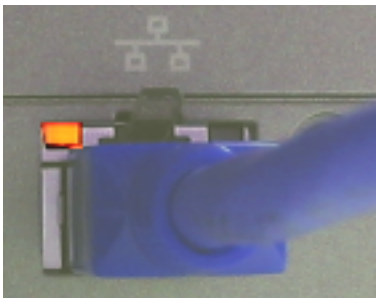
   (a) Open a command prompt window (DOS box) or use the one from above.

   (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.

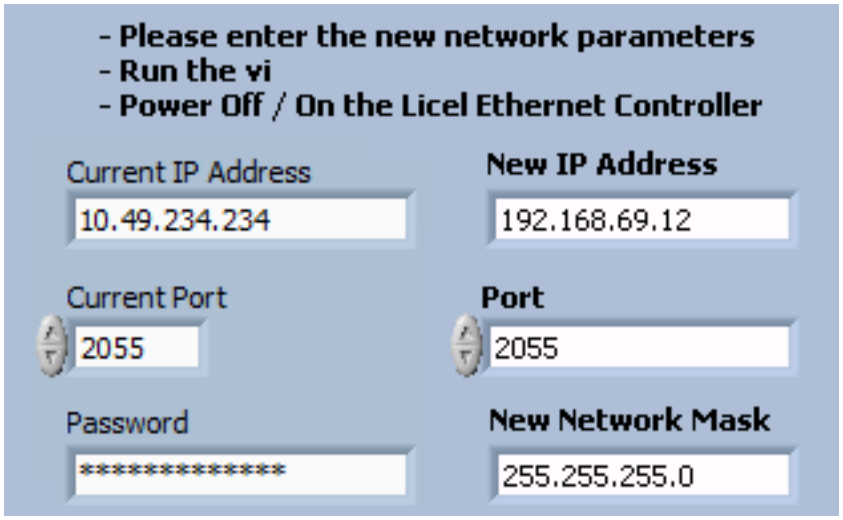
4.5 Network Setup

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.

4.5.1 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 `LicelTCPIP SetFixedIP_Address.vi` or the corresponding Windows application from the Windows start menu.



2. Set the desired network parameters in the fields **New IP Address** and **Port**. In this eaxample we set the Licel Ethernet Controller's IP address to `192.168.69.12` because this IP address lies in the same IP address range where we found our PC (in our example `192.168.69.10`). Never use a fixed IP address beginning with `169.254.` because the corresponding IP address range `169.254.x.y` is reserved as the DHCP fallback range for network clients that try to connect to a DHCP server but do not find any in the local network.

3. Check the **New Network Mask**: is the default network mask `255.255.255.0` suitable for the communication bewtween the PC and the Licel Ethernet Controller? A "`255`" at the first n positions of the network mask mean that the first n numbers of the IP addresses of both, the PC and the Licel Ethernet Controller, at the corresponding positions must be the same.

4. Do not forget to enter the correct administrator password of the Ethernet controller. The default password when shipped is *Administrator*, which is set into the password field by default. Enter a different password only if you have changed the Ethernet controller's administrator **Password**.

5. Run the vi by pressing the start  button. It should finish without opening an error message dialog.

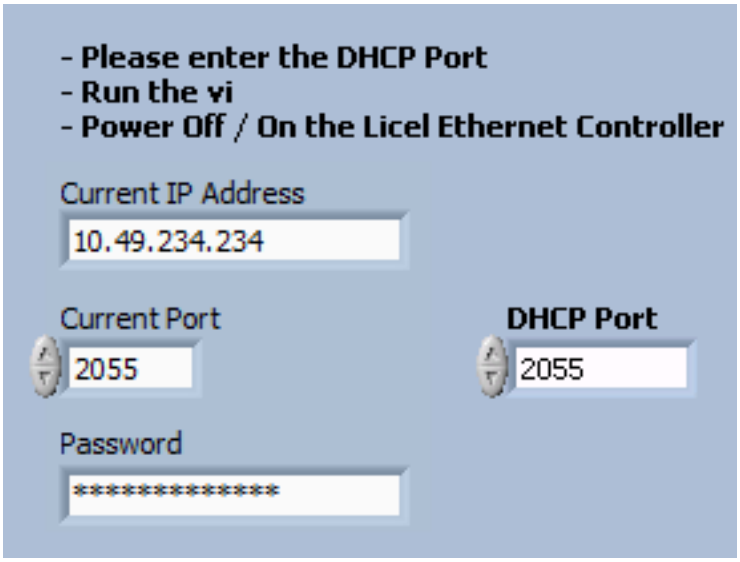
6. Turn the Licel Ethernet Controller off and switch it on again. Wait **approximately 20 – 30 seconds**.

7. A `ping 10.49.234.234` executed from a command prompt (DOS box) should now time-out.

4.5.2 DHCP Mode

In order to configure the Licel Ethernet 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 `LicelTCPIP ActivateDHCP_Mode.vi` or the corresponding Windows application from the Windows start menu.

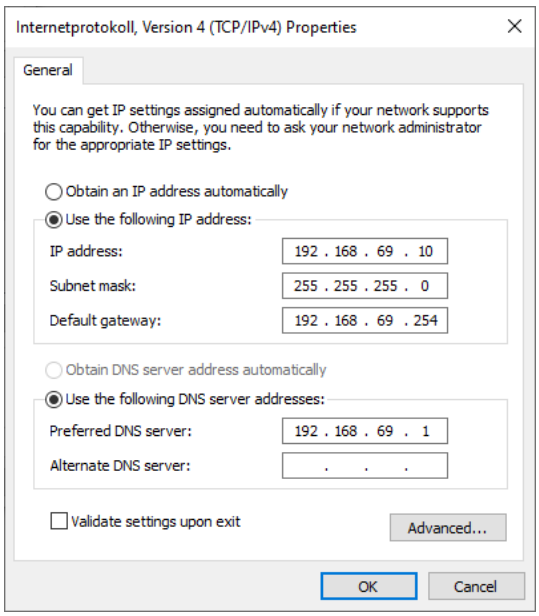


2. Set the desired **DHCP Port** number.
3. Do not forget to enter the correct administrator password of the Ethernet controller. The default password when shipped is *Administrator*, which is set into the password field by default. Enter a different password only if you have changed the Ethernet controller's administrator **Password**.
4. Run the vi by pressing the start button. It should finish without opening an error message dialog.
5. Turn the Licel Ethernet 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.

4.6 Reconfigure the PC

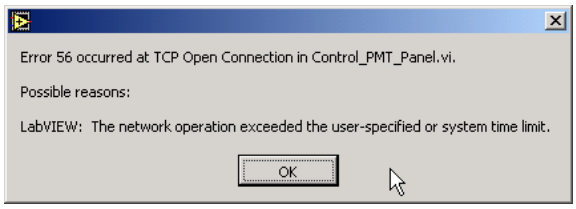
After you successfully configured the Licel Ethernet Controller the following last steps have to be carried out to reconfigure your PC for the local network and to test the connection to the Licel Ethernet Controller:

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 Licel Ethernet Controller.

5. Quit the dialog by pressing *OK*.
6. Reboot your PC.
7. Connect the Licel Ethernet 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 Licel Ethernet Controller. If the Ethernet controller is in DHCP mode, 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 *TCPIP GettingStarted.vi* or the corresponding Windows application to be started from the Windows start menu.
10. A TCP/IP timeout error with LabVIEW's error code 56 may be caused by a wrong IP address.

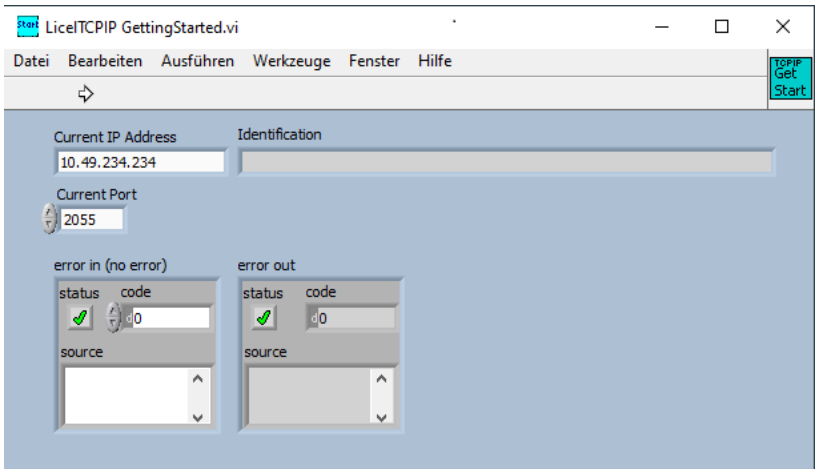


Please check carefully that the values for *IP Address* and *Port* match with the parameters set at the Licel Ethernet Controller. Set the correct values as defaults for future operation. Other reasons for errors with code 56 are non-existing connections (check if the cable in use is correct) or unstable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

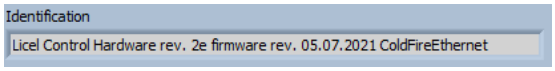
4.7 Test the TCP/IP Setup – Getting Started

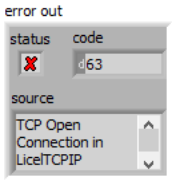
Once you assigned an IP address to your Licel Ethernet Controller, turned it off and switched it on again, and reconfigured the PC's IP address you may like to test whether or not a TCP/IP connection is possible from your PC to the Licel Ethernet Controller using the new TCP/IP settings.
For this you could start the Windows Application *TCPIP Getting Started* from the Windows start menu. Open the corresponding VI `LicelTCPIP GettingStarted.vi` from the LabVIEW project

in the subfolder `Licel TCPIP Acquisition\TCPIP` in the case that you are using the LabVIEW sources.
The front panel of *Getting Started* will open.



Please continue as follows:

1. Enter the *Current IP address* and *Port* as you have set them for the Licel Ethernet Controller in one of the previous sections.
2. Run the program using the run button 
3. The program will attempt to open a TCP/IP connection to the Licel TCPIP controller and request the identification string using a low level TCPIP command.
4. The *Identification* field should hold the controller's identification string now.
   

5. In the case of an error *error out* will show up with an error mark, an error *code* and an error *source*, here an example is shown:
   

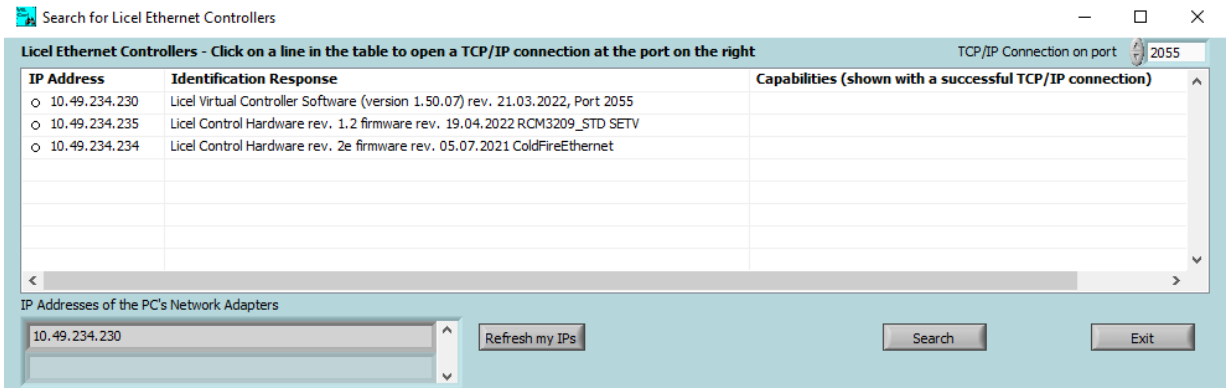
   The error codes are the codes LabVIEW is using. E.g. code 63 means:
   The network connection was rejected by the server. For TCP/IP, check that the server is running and polling the correct interface. Problems with connections are also often caused by firewalls.
   Please refer to the LabVIEW network error page https://www.ni.com/docs/en-US/bundle/labview-api-ref/page/errors/networking-error-codes.html for more information.

4.8 Search Controllers

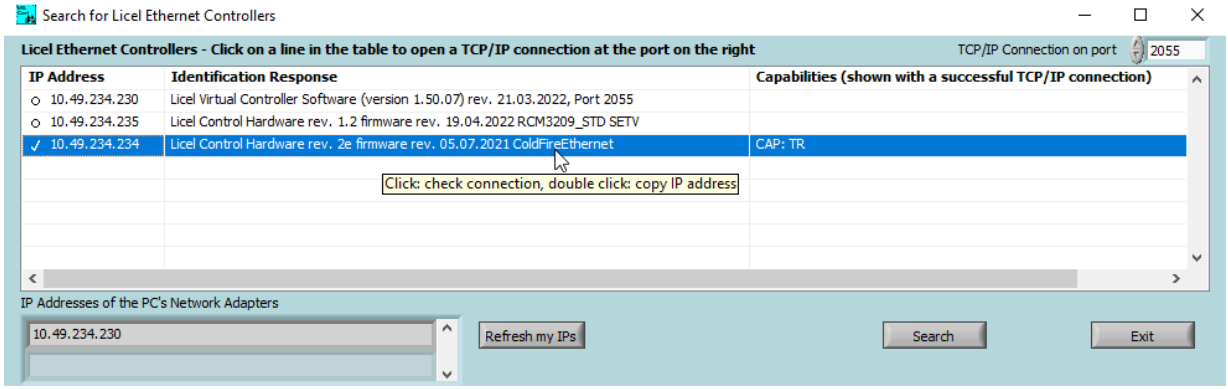
Licel provides the utility *Search Controllers* to search for Licel Ethernet Controllers in your Local Network.

- You can start the Windows Application *Search Controllers* from the Windows start menu. Then the program opens and immediately starts to run.
- If you are using the LabVIEW sources open the corresponding VI `Search Controllers.vi` from the LabVIEW project in the subfolder `Licel TCPIP Acquisition\TCPIP`. Run the VI by using LabVIEW's run button  .

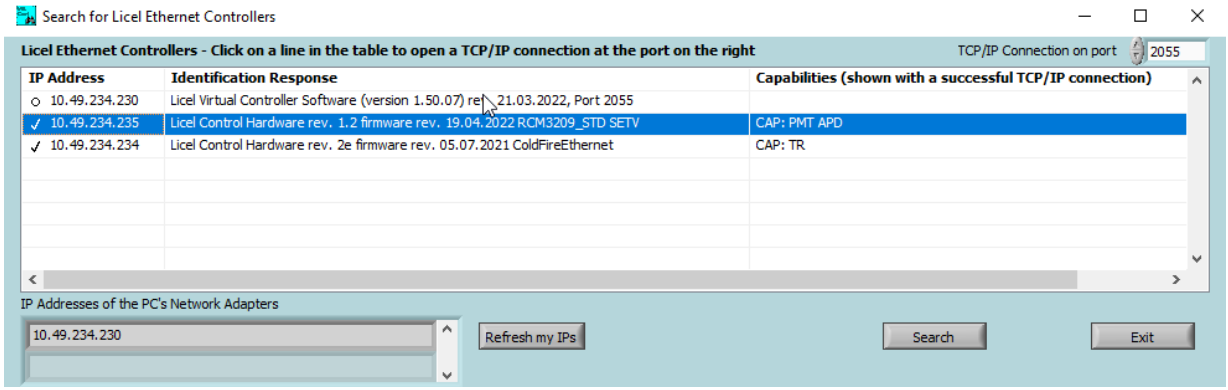
Search Controllers uses UDP polls to find Licel Ethernet Controllers in your network. Your firewall might ask for your allowance to do that. Use *Search Controllers* to make sure that all your controllers have been set to the correct IP addresses or to identify a specific controller e.g. after you have set it to the DHCP mode. After starting *Search Controllers* will display the found controllers in a table. Please click on the Search button if the table remains empty or if you would like to repeat a search because you have switched some controllers on or off.



In the example we see two controllers at the IP addresses 10.49.234.234 and 10.49.234.235, and additionally we see a *Virtual Controller*running on the PC's IP address (10.49.234.230 in this example). On your system you will see the IP addresses you assigned to your Licel Ethernet Controllers.
At the left of each line you see a circle symbol which indicates that no deatils are known about the corresponding controller at the moment ○.
Now click on one line representing one of the controllers. *Search Controllers* will attemps to open a TCP/IP using the port top right TCP/IP Connection on port 2055. If the TCP/IP connection could be successfully opened a checkmark ☑ will appear on the left. Furthermore *Search Controllers* will requst the controller's capabilities using a low level TCP/IP command. In the following example you will recognize that the controller at the IP address 10.49.234.234 has the capability *TR* i.e. it supports the control of transient recorders,



while the controller at 10.49.234.235 is a PMT and APD remote controller:



If the capability field is empty please check that no other software is communicating with the selected controller. If a connection is not successful (e.g. because the controller has been switched off in the meantime or the port is wrong) an appropriate sign is displayed on the left .
A double click on a line representing a controller will copy it's IP address to the clipboard.
At the bottom left you will find a list with the current IP address(es) of the PC. The list can be refreshed with the appropriate button.

4.9 TCP/IP Connection Parameters (Software)

To work properly with the Licel Ethernet 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 set for the Licel Ethernet Controller following the network setup section.
If a connection to the controller using the *IP Address* and *Port* in the corresponding control fields is not successful the applications will continiue to try to connect to the controller. The user may change the *IP Address* and *Port* during these reconnection attempts.
Defining the *IP Address* and *Port* is different for the Windows applications and the LabVIEW sources.

Windows Applications: Initialization Files

The Windows applications use initialization files to read their TCP/IP parameters *IP Address* and **Port** right after starting them. The applications will attempt to connect to the controller at the given TCP/IP parameters.
An example for an initialization file holding the TCP/IP information is given below:

```
[TCPIP]
UseValues=TRUE
IPAddress=10.49.234.234
Port=2055
```

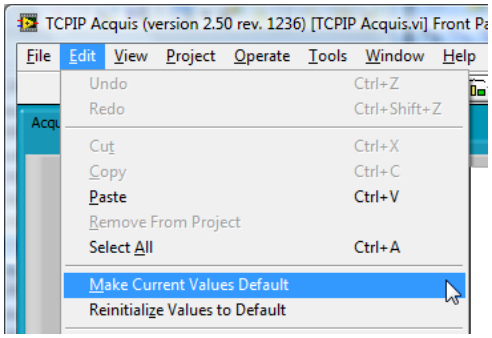
You may directly edit the corresponding initialization file using a text editor like `Notepad`. You must change the values for the *IP Address* and *Port* to the values you will set following the Instructions in the network setup section. Or change the *IP Address* and *Port* while the application is running and not yet connected, the application will try to build up a connection until it has success.
When the applications is able to open a TCP/IP connection with the given *IP Address* and *Port* these values will be written to the initialization file. The Windows applications will use these values at the next start of the program.
Here is an overview of the initialization file names used by the Windows applications of the Licel TCPIP Acquisition software for reading the TCP/IP information:

Windows Application	Initialization File
`Control APD-PMT.exe`	`Control APD-PMT.ini`
`Control Timing.exe`	`Control Timing.ini`
`Power Meter Control.exe`	`Power Meter Control.ini`
`TCPIP Acquis.exe`	`Acquis.ini`
`M-Acquis.exe`	`Acquis.ini`
`TCPIP Live Display.exe`	`TCPIP Live Display.ini`
`TCPIP Pulse Height Distribution.exe`	`TCPIP Pulse Height Distribution.ini`
`TCPIP Track.exe`	`TCPIP Track.ini`
`Licel Main.exe`	`Licel Main.ini`

LabVIEW: Setting Default TCP/IP Parameters

The LabVIEW VIs will not read the `IPAddress` and `Port` from the initialization file as the Windows applications do. However, when opening a LabVIEW VI within a LabVIEW development environment, default values can be defined for controls on the panel of a LabVIEW VI. This is especially convenient and recommended for the TCP/IP parameters **IP Address** and *Port*. Change the values to the values you set following the Instructions in the network setup section.

1. Open the vi using LabVIEW, do not run the vi.
2. Enter the value for the IP address into the control named *IP Address*.
3. Right-click on the control *IP Address* ⟶ the context menu opens.
4. Select **Data Operations** ⟶ a sub menu opens.
5. Select **Make Current Value Default**.

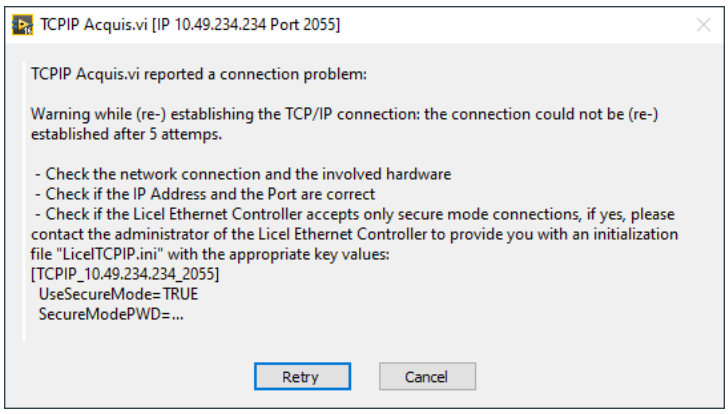


6. Repeat this procedure for *Port*.
7. Save the vi.

Although the LabVIEW VIs do not read the *IP Address* and *Port* from the initialization file they will save these values to the file for documentation whenever a TCP/IP connection could successfully be opened.

4.9.1 TCP/IP Connection Problems (Software)

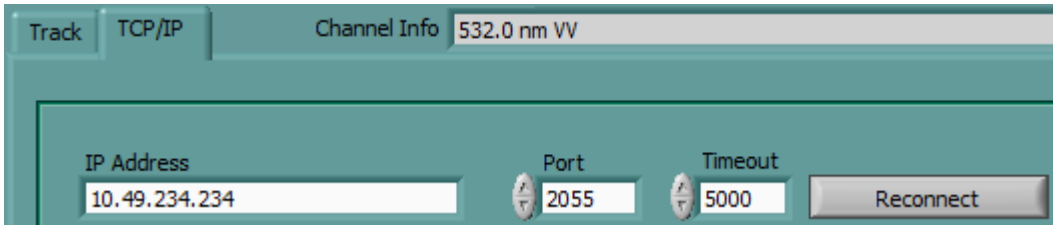
The parent application *Licel Main* as well as the software modules when run stand-alone (*Track*, *Live*, *Acquis*, *Control APD-PMT*, *Control Timing*, etc.) have a built-in mechanism to re-establish the TCP/IP connection to the Licel Ethernet Controller when the connection is lost or when the connection 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:



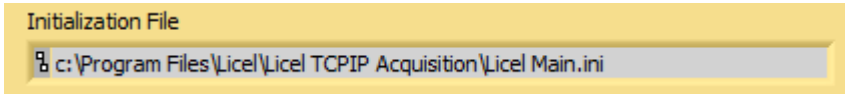
In the case that this dialog comes up please

- check the network connection and the involved hardware. Check whether the Licel Ethernet Controller and all other Licel hardware 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 Licel Ethernet 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 Licel Ethernet Controller
     - Using the LabVIEW vi, just enter the required values on the *TCP/IP* page and save them as defaults.



     If the warning dialog is not closed by a user (Retry or Cancel) it will close automatically and the program will continue to attempt to connect to the Licel Ethernet Controller using the given *IP Address* and *Port*.
     - If you run a Windows application you should check the values in the corresponding initialization file. You will see the full path of the file in a file path indicator on the *TCP/IP* page.



     While a Licel Windows application **is running** (and has not yet a TCP/IP connection) you may enter the *IP Address* and the *Port* directly. If a connection can be established (i.e. the values are correct) the parameters will be written to the appropriate initialization file directly after successfully establishing a TCP/IP connection.

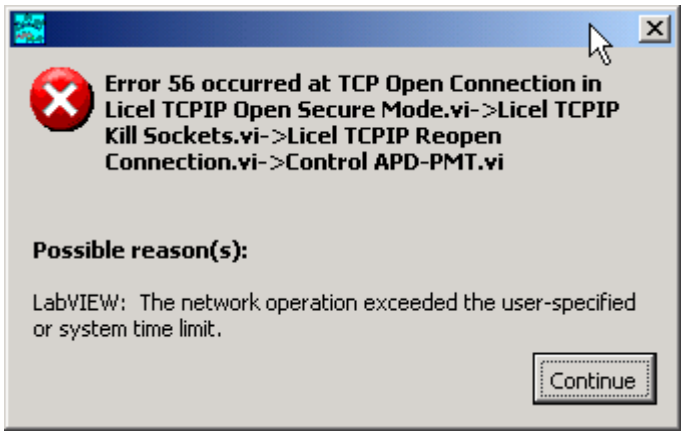
- check whether the Licel Ethernet Controller operates in secure mode. If secure mode is used please check the following section in the file `LicelTCPIP.ini`:

```
[TCPIP_<Controller-IP-Address>_<Controller-Port>]
UseSecureMode=TRUE
SecureModePWD=<SecureModePassword>
```

where `Controller-IP-Address` and `Controller-Port` are the IP address and port of the Licel Ethernet Controller, respectively. If necessary, ask your administrator for the correct password for usage in secure mode.

You have the following choices to continue when the warning dialog appears:

1. Click *Retry* to continue to reconnect to the Licel Ethernet Controller.
2. Click *Cancel* to exit. The program will display an error message (here an example for *Control APD-PMT*, the dialog's appearance may depend on the LabVIEW version):



3. Do nothing – the application will automatically close the warning dialog and try again to connect to the TCP/IP controller with the current *IP Address* and *Port*.

4.9.2 Change the Ethernet Controller (in the Application)

If you have more than one Licel Ethernet Controllers (e.g. 2 Detector Remote Controls) and recognize that you accidentially connected to the wrong controller you may easily change the controller:

1. Enter the *IP Address* and *Port* of that controller you really would like to connect to.
2. Press the 

button.

The application will then close the open TCP/IP connection and reconnect with the new *IP Address* and *Port*. Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

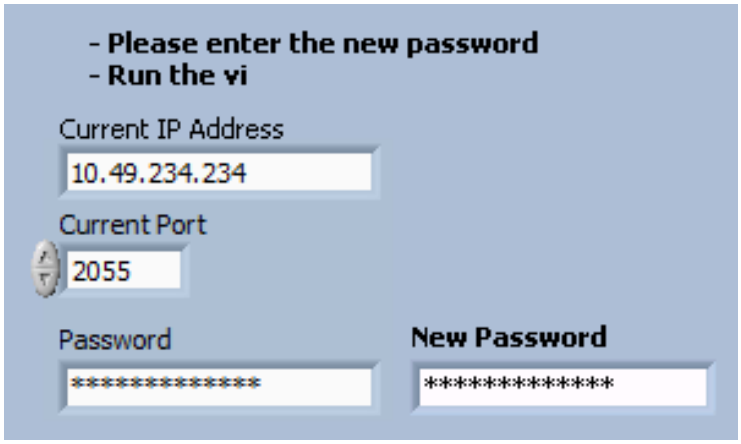
4.10 Network Security

The Licel Ethernet Controller provides two levels of 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. Furthermore a *Secure Mode* based on an encryption mechanism is available.

4.10.1 Changing the Administrator Password

The Licel Ethernet 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 `LicelTCPIP SetNewPassword.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.

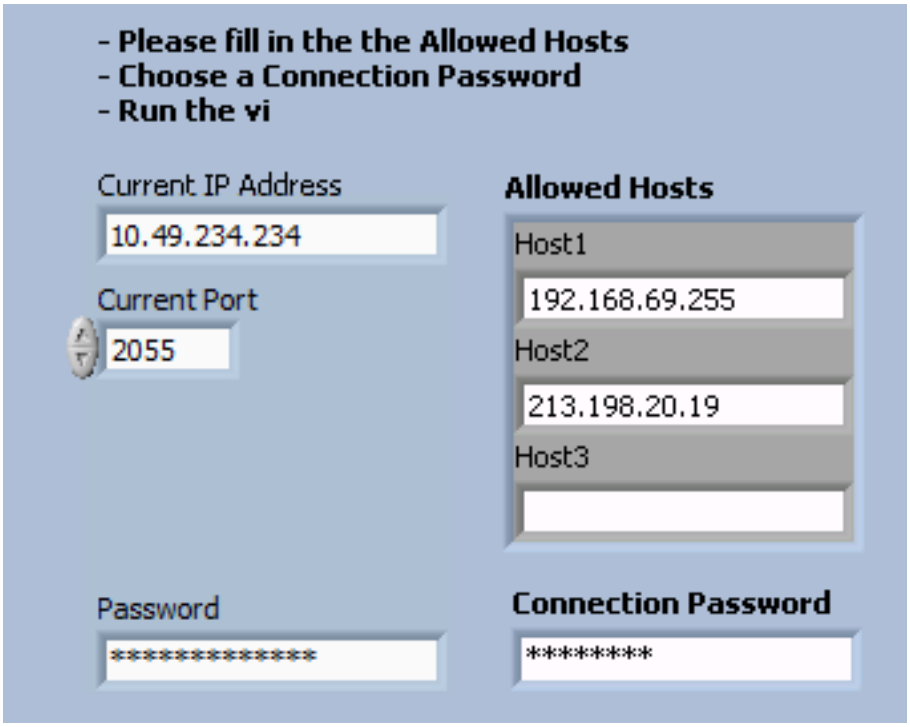
4.10.2 Secure Mode

The Licel Ethernet Controller might be the target of an attack. The best protection against this is to run the controller with a private IP address beyond a firewall. Firewalls are designed to protect against various types of attacks that can not be covered by the Ethernet controller. Licel strongly recommends the use of a firewall/router combination to prevent unauthorized use of the hardware.
Starting with firmware versions from 2005-02-22 (`state53`) the Licel Ethernet Controller has an additional level of security that can be additionally used.
This secure mode combines whitelisting of allowed hosts with an encrypted password transmission scheme.

Enabling the Secure Mode

In order to enable the Secure Mode for the Licel Ethernet Controller carry out the following steps:

1. Open the LabVIEW VI `LicelTCPIP EnableSecureMode.vi` from the LabVIEW project in the subfolder `Licel TCPIP Acquisition\TCPIP` or the corresponding Windows application from the Windows start menu.



2. Set the desired whitelist of allowed host IP addresses or address ranges. An entry in the list of **Allowed Hosts** is either

   - a host specified by its IP address `xx.xx.xx.xx` ,
   - an IP address range `xx.xx.xx.255` ranging from 0 to 255, or
   - empty .

   At least one valid entry must be submitted. Do not forget to include the IP address of the PC you are currently using.

3. Set the **Connection Password**. This password must be used by clients accessing the Licel Ethernet Controller as long as the secure mode is enabled. Refer to the `LOGON` command for further details. Please note that the password is case sensitive.

4. Do not forget to enter the administrator **Password**.

5. Run the vi by pressing the start button. It should finish without opening an error message dialog.

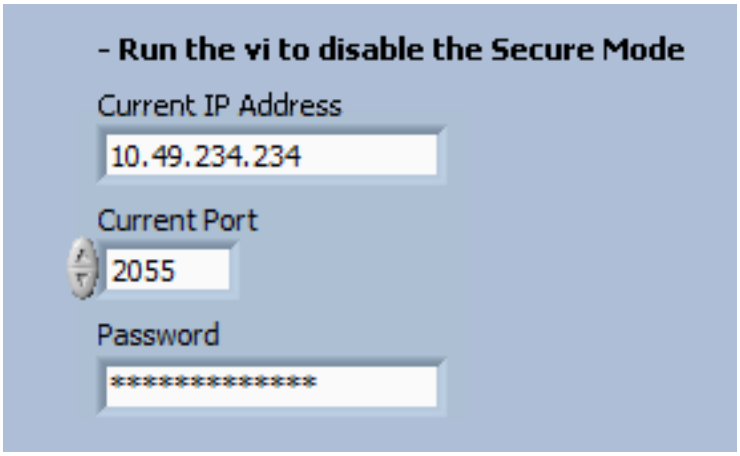
6. The vi will write an initialization file `LicelTCPIP.ini` with appropriate keys and values. These values are used by the sample applications while establishing a connection to the controller. Distribute the initialization file to all PCs the sample programs are installed on.

7. Test the access using `LicelTCPIP GettingStarted.vi` or the corresponding Windows application to be started from the Windows start menu.

Disabling the Secure Mode

In order to disable the Secure Mode for the Licel Ethernet Controller carry out the following steps:

1. Open the LabVIEW VI `LicelTCPIP DisableSecureMode.vi` from the LabVIEW project in the subfolder `Licel TCPIP Acquisition\TCPIP` or the corresponding Windows application from the Windows start menu.



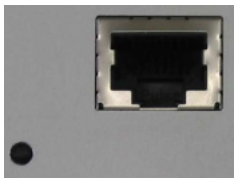
2. Do not forget to enter the administrator administrator **Password**.

3. Run the vi by pressing the start button. It should finish without opening an error message dialog. Note that this vi can only be used if the Licel Ethernet Controller is running in secure mode and if a valid initialization file `LicelTCPIP.ini` is located in the directory where the vi's library resides.

4. The vi will update the initialization file `LicelTCPIP.ini` with appropriate keys and values. These values are used by the sample applications while establishing a connection to the controller. Distribute the initialization file to all PCs the sample programs are installed on.

4.11 Hardware Reset

A reset is performed by pressing the reset switch while powering up the controller. The reset switch is located inside a hole close to the RJ45 connector.



To reset the system

- turn off the controller unit
- press the switch inside the hole with a small screw driver, Allen key or anything similar
- turn the rack on while keeping the switch pressed, release the switch 5 seconds after switching the unit on, wait for 45 seconds.

After a reset

- the controller has the default IP address
- the port number is reset to the default value
- the controller operates in its fixed IP address mode
- the password is reset to the default password.

Chapter 5

Transient Recorder Software Tutorial

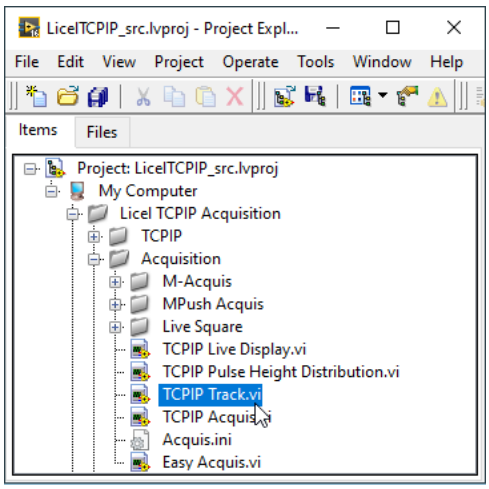
5.1 Overview

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. The quick tour gives a brief introduction to recording spectra with the software module TCPIP Track and TCPIP Live Display. The Acquisition Software contains instructions for recording your first spectra using TCPIP Acquis/M-Acquis and TCPIP MPush Acquis.

5.2 Quick Tour

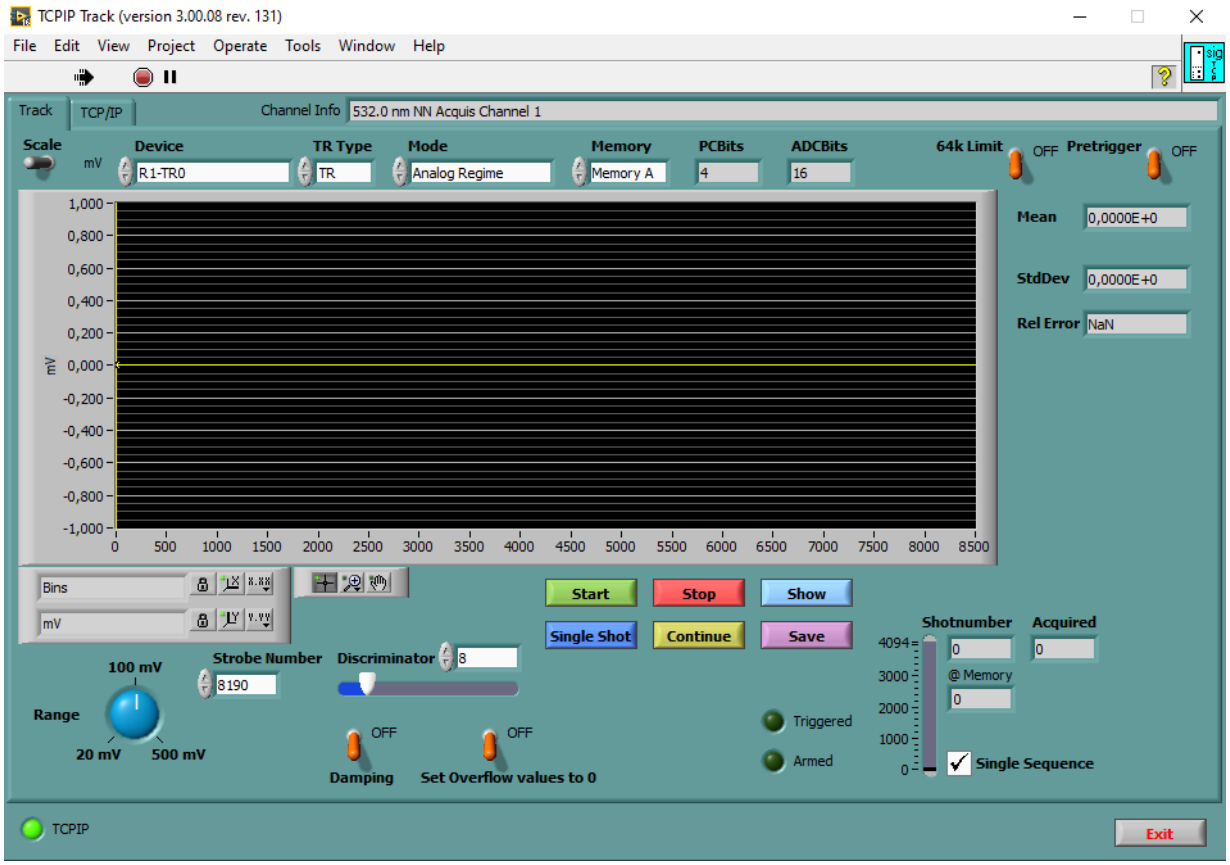
5.2.1 TCPIP-Track

- If you are using the LabVIEW sources open *TCPIP Track* from the LabVIEW project by navigating to the corresponding entry *TCPIP Track.vi* and double-clicking it.



- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

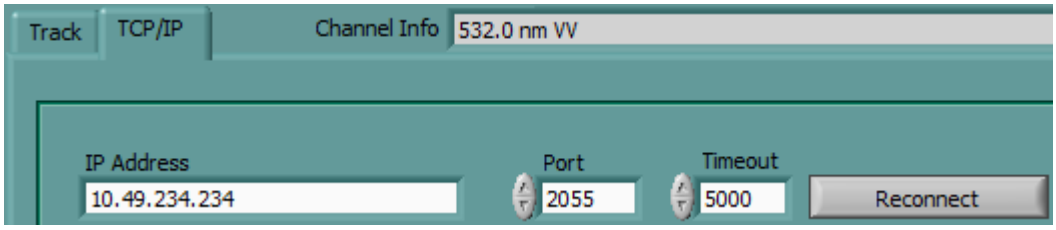
After doing so, you should see a screen similar to the one below. *TCPIP Track* is a program that can be used to access all the individual functions of the transient recorders. It allows you to control one individual recorder at a time.



If you have completed the hardware setup and configured the network you should be ready to experiment with the software. Do the following steps to get a brief introduction of the software *TCPIP Track*.

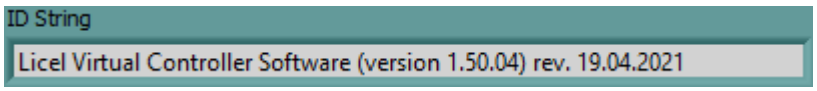
1. First of all the *IP Address* and *Port* have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

   - Using the LabVIEW vi, just enter the required values on the *TCP/IP* page and save them as defaults.

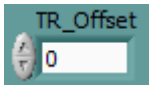


   - If you use the Windows application you must directly enter the correct values into the corresponding control fields on the *TCP/IP* page. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.
     You may also set the values in the initialization file `TCPIP Track.ini`. You will see the full path of the file in a file path indicator on the *TCP/IP* page.

   - If you run *TCPIP Track* within a sub panel on a page from *Licel Main* the latter is responsible for managing the TCP/IP connection.
   - In recent versions of *TCPIP Track* the *IDN string of the Licel Ethernet Controller is displayed.

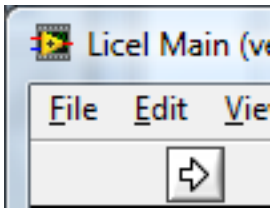


- If you are working with the multi-rack acquisition software M-Acquis and you are directing *TCPIP Track* to the second (or third ...) rack, you need to add a constant offset to the device number to set up the device selection list and to obtain correct values from the initialization file `Acquis.ini`. This is done by selecting the *TR_Offset* (1st rack: 0, 2nd rack 16, ...).

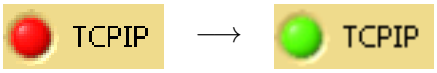
  If you run *TCPIP Track* within a sub panel on a page from *Licel Main* the latter is responsible for submitting the correct *TR_Offset*.

2. To start the program press the *Run* button at the top left of the screen.

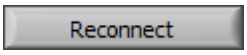
   The Windows application will start automatically when called for the first time.

3. After a short time the *TCPIP* indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for *IP Address* and *Port*, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct. The LED of the transient recorder should be lit up.

4. If you recognize that you are not connected to the Ethernet Controller you would like to use, just

   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and

   (b) press the 

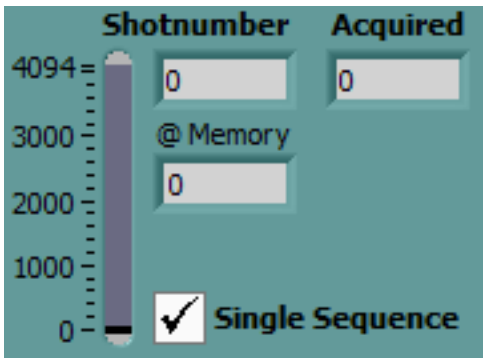
   button.

   Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

5. Press the *Start* button directly below the waveform graph.

6. After pressing Start, the *Shotnumber* should start increasing. The shot number is increased by one for every trigger pulse that is received.



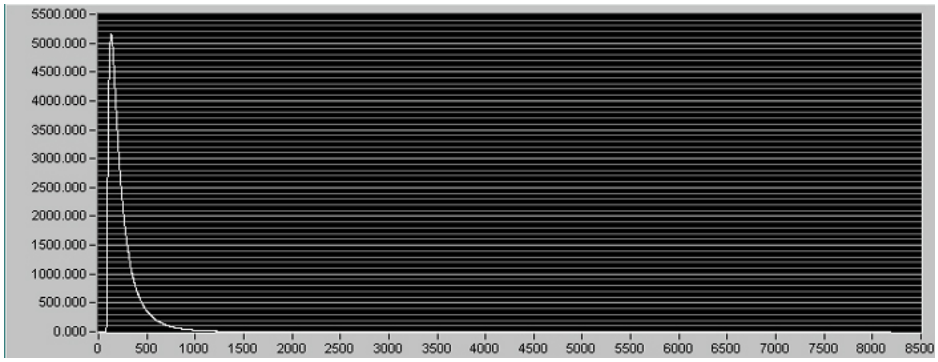
If the 

-option is checked the acquisition will stop at 4094 (or 65534 if *64k Limit = ON*) received trigger pulses. The behavior of the unchecked option will be explained below.

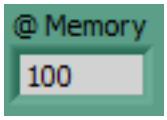
7. Press the *Stop* button after a few seconds to stop the acquisition

8. Press the *Show* button to display the results.

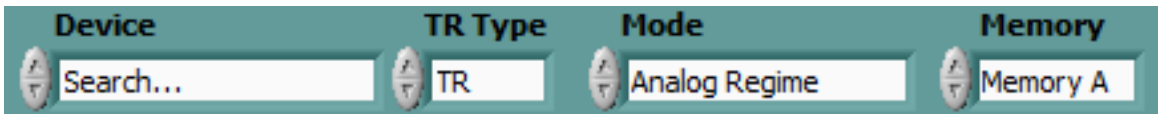
9. You will see the acquired signal in the graph window. Your signal will differ depending upon your system configuration. The example below is the simulation of a well aligned low noise system.



10. Once the display has been updated, the individual shot number at the selected memory is read and displayed in (*@Memory*) when separate shot counters are available at the transient recorder.

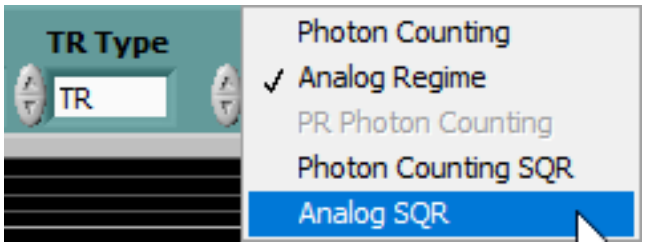


11. The data set that is displayed after pressing the *Show* button is selected by the *Device*, *TR Type*, and *Mode* switches at the top of the waveform graph.



Device refers to the hardware address of the transient recorder. Before selecting the *Mode* you must set the *TR Type* in compliance with the external hardware module that you are addressing. The transient recorder is either a TRxx-xx or PRxx-xx, *TR Type* must be set to *TR* or *PR*, respectively. Both types differ in their memory layout as you might see in the programming manual (https://www.licel.com/manuals/programmingManual.pdf, page 24 ”Memory organization”). *Mode* depends on the *TR Type*:

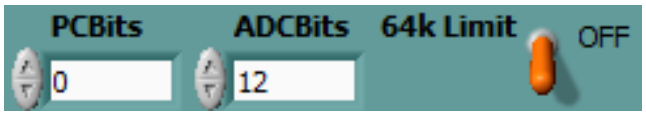
TR Type	*Mode*	
TR	*Photon Counting*	the accumulated data from the counting chain is displayed
	Analog Regime	the ADC data is displayed
	Photon Counting SQR	show the raw squared photon counting data
	Analog SQR	show the raw squared analog data
PR	*PR Photon Counting*	the photon counting data is displayed



The data set is further specified by the *Memory* switch. *Memory A* corresponds to acquisitions which were triggered by Trigger A, and *Memory B* corresponds to acquisitions triggered at input B. If only one trigger input is connected to the trigger source, only one memory can hold data different from 0.

12. Depending on the current *Device*, *Mode*, and *Memory* the corresponding wavelength, laser and detection polarizations, and custom information from the înitialization file `acquis.ini` is displayed (if available):

13. Further settings for the selected transient recorder (*Device*) are available at *PCBits*, *ADCBits*, and *64k Limit*.



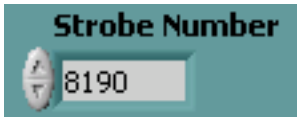
The numbers of photon counting bits and analog bits (*PCBits* and *ADCBits*) of the selected transient recorder are read from the Licel Ethernet Controller (supported since spring 2011). In the case that the controller does not support the request the numbers of photon counting bits and analog bits can be set by the user. The values must correspond to the installed transient recorders.

With the *64k Limit* the maximum number of acquirable shots (shot limit) is selected: *OFF* and *ON* correspond to the shot limits 4094 and 65534, respectivlely. *64k Limit* = *ON* is supported only by 16 bit transient recorders connected to an appropriate controller (available since 2011).

14. The *Pretrigger* control is supported only if the corresponding feature is supported by the selected transient recorder (*Device*) and the Licel Ethernet Controller.

If the *Pretrigger* is enabled, the acquired trace consists of 1/16th of the transient recorder memory tracelength before the trigger and the remaining points after the trigger point.

15. The horizontal scale is given in bins. The number of displayed bins is controlled by the *Strobe Number* control. This number should not exceed the memory length of the transient recorder; please refer to the memory selection guide http://licel.com/model_select.htm to find out the maximum allowed value of your transient recorder.



16. The vertical scale has different meanings for the analog and the photon counting mode.
    - For photon counting data the mean number of counts per bin is displayed. For a given range bin the number of accumulated counts is divided by the shot number.
    - For the analog data the display can be either in mV or in least significant bits (LSB) and may be selected using the most top-left switch:

      The LSB scale corresponds to the ADC reading, for a 12 bit transient recorder it can vary between 0 and 4095.

17. Press *Continue* to continue accumulation without clearing the memory. Pressing *Start*would clear both memories.

18. After a few seconds press the *Stop* button followed by the *Show* button and notice how the signal-to-noise ratio has improved.

19. The acquired data can be saved using the *Save* button. The data is written to an ASCII file in a single column. The path can be chosen in a separate file selection dialog.

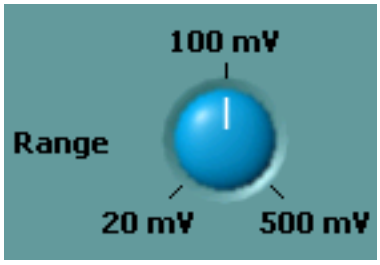
    The data is written to a binary file (single precision 4 byte floating point) using LabVIEW's `vi.lib\Utility\file.llb\Write To SGL File.vi`.

20. If the 

-option is not checked the acquired data will be read from the transient recorder and displayed once the shot counter reaches 4094 (or 65534 if *64k Limit = ON*). The acquired shot number will be displayed in the *Acquired* field. Then the transient

resorder is restarted to acquire the next up to 4094 (64534) shots. The next time the counter reaches 4094 (64534) the data read from the transient recorder will be added to the data acquired before.

21. Change the input sensitivity with the *Range* knob located at the bottom left.



The three displayed millivolt values indicate the full scale negative voltages.

22. Take a new acquisition by repeating steps 1 through 7 to see the influence of the input range.
23. Press the *Stop* button to stop the acquisition.
24. Connect a photomultiplier to the signal input on the transient recorder and switch to photon counting mode.

25. Using the *Discriminator* slide, you can set the discriminator level between 0 and 63.

Change to the photon counting mode and make a few acquisitions to see how the count rate is influenced by the discriminator setting. For details about setting the discriminator refer to the Pulse Height Distribution section.

26. Change back to analog mode
27. Turn the *Damping* switch on and make a new acquisition

This reduces the counting rate since the discriminator level is set four times as high.

28. The *Set Overview Values to 0* switch helps you to analyze whether the signal exceeds the acquisition range.

Once you have acquired a real signal and the overflow at the transient recorder flashes switch the *Set Overflow Values to 0* switch and see if you see spikes towards zero. Those spikes will give you the exact position where the overflow occurred. With this information you can decide

if you have overflows or underflows. Overflows can be avoided by increasing the signal input range or lowering the amplitude of the input signal underflows might require recalibration of the instrument as shown in https://www.licel.com/manuals/TR40-16bit3U_Manual.pdf (section 4.3 Analog Background).

Underflows are a serious thread to signal integrity as they are hard to notice and might distort the background computation. They should be fixed as otherwise the far field signal will be incorrect.

29. The *triggered* lamp is turned on if a shot is acquired while the program makes a status request. The *armed* lamp below the trigger lamp lights up when the transient recorder is waiting for the next trigger event while the program makes a status request.

30. The following switch is available only if *TCPIP Track* is running as a sub module in `Licel Main`.

When switched `ON` it allows to keep the memory block setting from the configuration of `TCPIP Acquis` (default behavior when running as a sub module). The only condition to achieve keeping the memory block setting is that memory blocking is active for the selected transient recorder and that `Licel Main`'s tab page running `TCPIP Acquis` has been entered at least once. When the Licel Ethernet Controller does not support the `BLOCK` command the setting of the switch has no influence.

To make sure that keeping the memory block settings is active you should once

(a) Activate *Keep MemBlock* (ON)
(b) Change to the Acquisition Tab (TCPIP Acquis)
(c) Open the configuration in TCPIP Acquis
(d) Check *Block Trigger* for all desired transient recorders
(e) Save the configuration
(f) When leaving the configuration dialog of `TCPIP Acquis` the memory blocking will be set at the Ethernet controller
(g) Switch back to Track
(h) After starting you will only see shots at non-blocked memories

2 The next time you start *TCPIP Track* inside of `Licel Main` it is sufficient to switch once to the Acquisition tab page to achieve keeping `TCPIP Acquis`'s memory blocking.

Keeping the memory block settings may be switched off (*Keep MemBlock* (OFF)). To switch it on again, you must

(a) Activate *Keep MemBlock* (ON)
(b) Change to the Acquisition Tab (TCPIP Acquis)
(c) Switch back to Track

Please note: when running stand-alone *TCPIP Track* will not show this switch. All triggers received at any memory of the transient recorder will be enabled and included into the accumulated shot number.

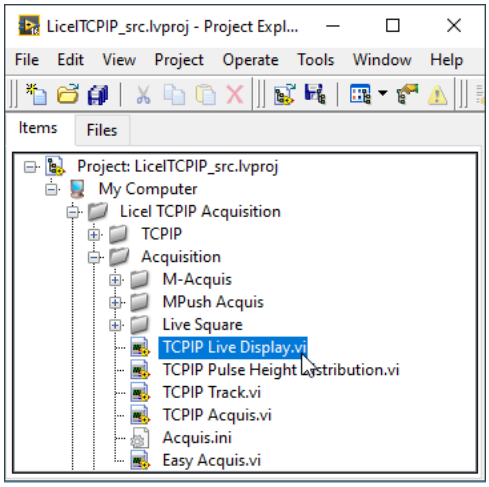
31. Use the *Exit* switch or the Window close button ✕ to stop the program, when running as a Windows or Linux application the front panel window will close.

That's the end of the short introduction into the capabilities of the data acquisition software *TCPIP Track*.

5.2.2 TCPIP Live Display

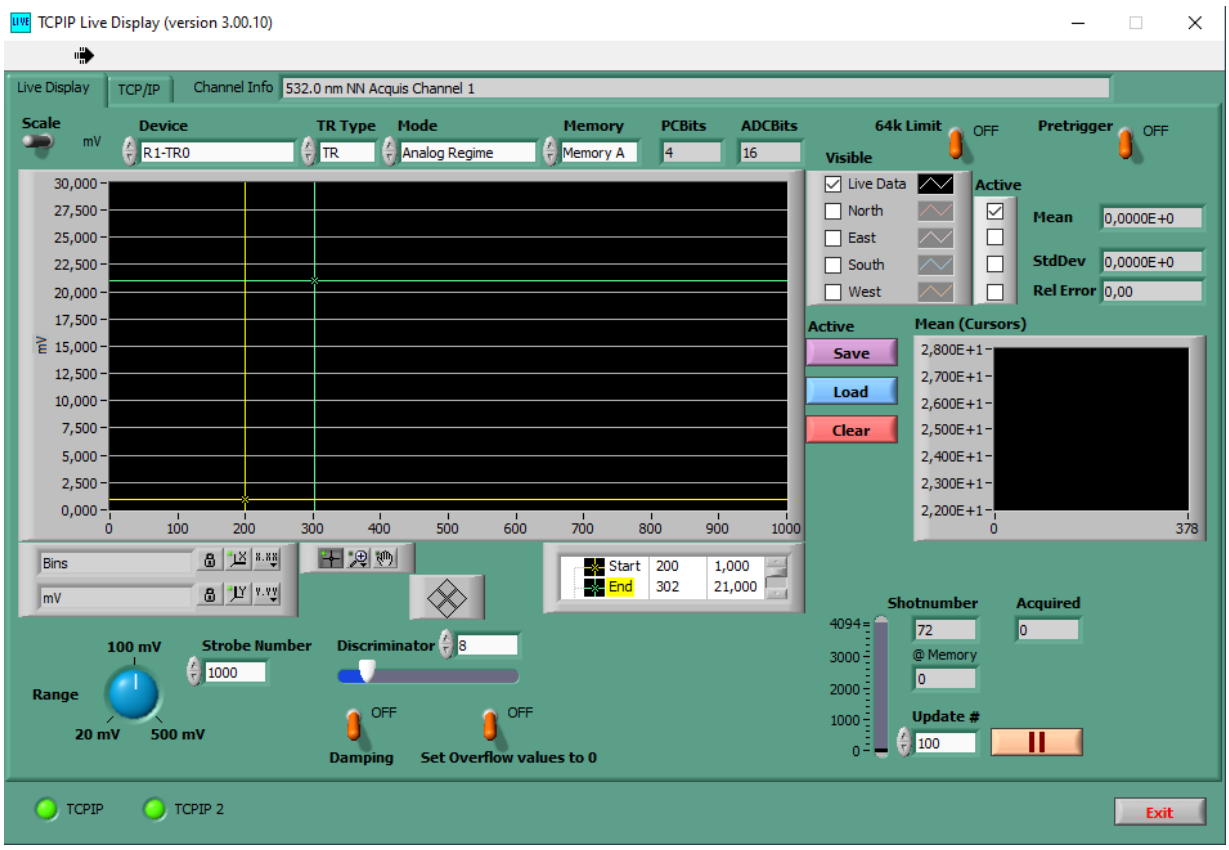
TCPIP Live Display allows you to operate the transient recorder in an oscilloscope mode, where the display is updated every X number of shots. This mode is very useful when you are trying to align the optics on your system and would like to see how the changes affect the signal.

- If you are using the LabVIEW sources open *TCPIP Live Display* from the LabVIEW project by navigating to the corresponding entry *TCPIP Live Display.vi* and double-clicking it.



- If you installed the Windows application please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

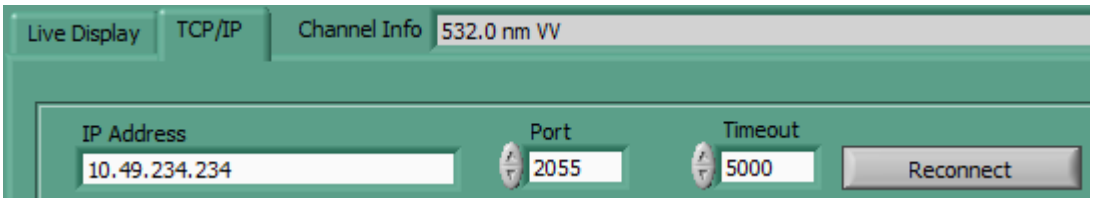
After this the following interface appears:



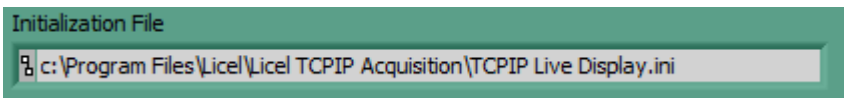
As you can see, the interface is quite similar to that of TCPIP Track. The main difference is that the buttons *Start*, *Stop*, *Show*, *Single Shot*, *Continue*, and *Save* are missing (In Live Display *Save* has a different functionality than in Track). Instead, there is an *Update #* control and a second graph called *Mean(Cursors)*. Furthermore there are control elements related to displaying reference signals.

1. First of all the *IP Address* and *Port* have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

   - Using the LabVIEW vi, just enter the required values on the *TCP/IP* page and save them as defaults.

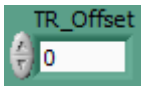


   - If you use the Windows application you must directly enter the correct values into the corresponding control fields on the *TCP/IP* page. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.
     You may also set the values in the initialization file `TCPIP Live Display.ini`. You will see the full path of the file in a file path indicator on the *TCP/IP* page.



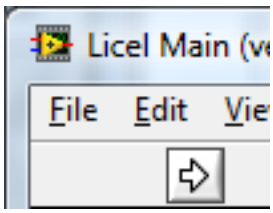
   - If you run *TCPIP Live Display* within a sub panel on a page from `Licel Main` the latter is responsible for managing the TCP/IP connection.
   - If you are working with the multi-rack acquisition software M-Acquis and you are directing *TCPIP Live Display* to the second (or third ...) rack, you need to add a constant offset to

the device number to set up the device selection list and to obtain correct values from the initialization file `Acquis.ini`. This is done by selecting the *TR_Offset* (1st rack: 0, 2nd rack 16, ...).



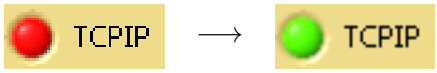
If you run *TCPIP Track* within a sub panel on a page from *Licel Main* the latter is responsible for submitting the correct *TR_Offset*.

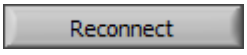
2. To start the program press the *Run* button at the top left of the screen.



The Windows application will start automatically when called for the first time.

3. After a short time the *TCPIP* indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for *IP Address* and *Port*, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct. The second TCPIP indicator should change to a green color after some time, too. This second connection is used to directly transfer data from the transient recorders to the acquisition computer if the *update #* is smaller than 15.



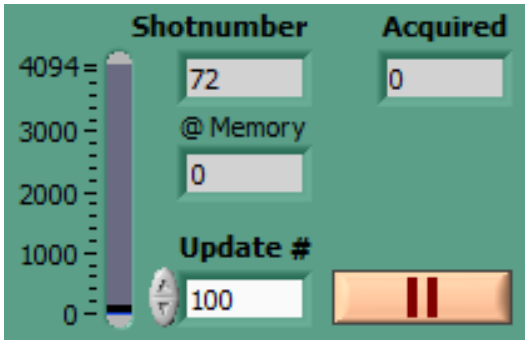
4. If you recognize that you are not connected to the Ethernet Controller you would like to use, just
   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and
   (b) press the 

button.

   Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

5. Notice that the *Shotnumer* indicator immediately starts to increase once a TCP/IP connection has successfully been establihed. When the *Shotnumber* reaches the *Update #*, the signal data is read from the transient recorders and displayed.

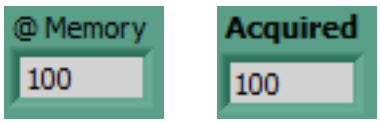
6. Set the *Update #* to 100.



If your trigger is operating at 10Hz, the data display will now be updated every ten seconds. By changing this value, you decide how many shots will be taken between subsequent updates

of the display. If the shot number is set a value larger than 4094 the transient recorder's data is read when reaching a multiple of 4094 and immediately displayed. The transient recorder is restarted, and the next acquired data is added until the target shot number is reached. The shot number corresponding to the last displayed data is shown in the *Acquired* field. Please note that the shot number is obtained from the reponse to the command, which returns the sum of the shot numbers of the triggers A and B (if both triggers are switched on).

7. Once the display has been updated, the individual shot number at the selected memory is read and displayed in (*@Memory*) when separate shot counters are available at the transient recorder. The field *Acquired* is relevant if an *Update #* has been chosen that is larger than the transient recorders shot limit (here: 4094 shots).

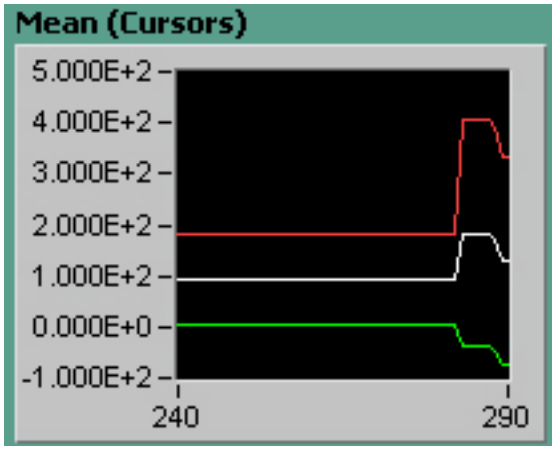


8. Set the Strobe Number to 4000:

notice that the length of the signal on the x-axis is now 4000, that is the number of bins read from the transient recorder. This number should not exceed the memory length of the transient recorder; please refer to the memory selection guide http://licel.com/model_select.htm to find out the maximum allowed value of your transient recorder.

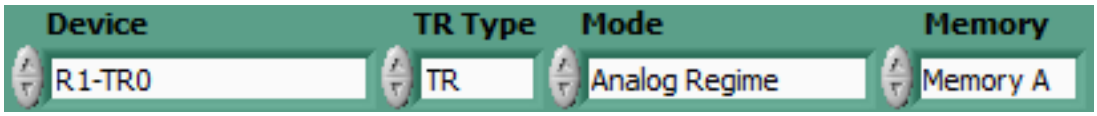
This indicates that only the first 4000 bins of the transient recorder memory are being read out and displayed.

9. Move the cursors to select part of the signal on the waveform graph. The region between the two cursors is averaged and displayed in *Mean (Cursors)* as the white curve. Additionally the +/- one standard deviation lines (red/green) are shown.



The further control elements have the same function as in TCPIP Track:

1. The data set that is displayed after the *Shotnumber* reaches the *Update #* is selected by the *Device*, *TR Type*, and *Mode* switches at the top of the waveform graph.



Device refers to the hardware address of the transient recorder. Before selecting the *Mode* you must set the *TR Type* in compliance with the external hardware module that you are addressing. The transient recorder is either a TRxx-xx or PRxx-xx, *TR Type* must be set to *TR* or *PR*,

respectively. Both types differ in their memory layout as you might see in the programming manual (https://www.licel.com/manuals/programmingManual.pdf, page 24 ”Memory organization”). *Mode* depends on the *TR Type*:

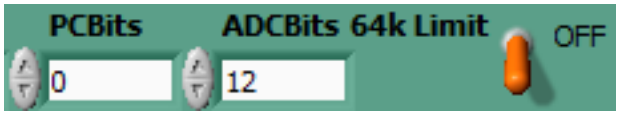
TR Type	*Mode*	
TR	*Photon Counting*	the accumulated data from the counting chain is displayed
	Analog Regime	the ADC data is displayed
PR	*PR Photon Counting*	the photon counting data is displayed

The data set is further specified by the *Memory* switch. *Memory A* corresponds to acquisitions which were triggered by Trigger A, and *Memory B* corresponds to acquisitions triggered at input B. If only one trigger input is connected to the trigger source, only one memory can hold data different from 0.

A change of these settings will restart the selected *Device*.

2. Depending on the current *Device*, *Mode*, and *Memory* the corresponding wavelength, laser and detection polarizations, and custom information from the înitialization file `acquis.ini` is displayed (if available):

3. Further settings for the selected transient recorder (*Device*) are available at *PCBits*, *ADCBits*, and *64k Limit*.



The numbers of photon counting bits and analog bits (*PCBits* and *ADCBits*) of the selected transient recorder are read from the Licel Ethernet Controller (supported since spring 2011). In the case that the controller does not support the request the numbers of photon counting bits and analog bits can be set by the user. The values must correspond to the installed transient recorders.

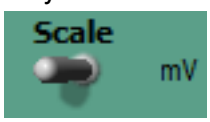
With the *64k Limit* the maximum number of acquirable shots (shot limit) is selected: *OFF* and *ON* correspond to the shot limits 4094 and 65534, respectivlely. *64k Limit* = *ON* is supported only by 16 bit transient recorders connected to an appropriate controller (available since 2011).

4. The *Pretrigger* control is supported only if the corresponding feature is supported by the selected transient recorder (*Device*) and the Licel Ethernet Controller.

If the *Pretrigger* is enabled, the acquired trace consists of 1/16th of the transient recorder memory tracelength before the trigger and the remaining points after the trigger point.

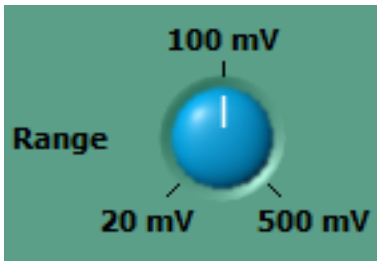
5. The vertical scale has different meanings for the analog and the photon counting mode.

   - For photon counting data the mean number of counts per bin is displayed. For a given range bin the number of accumulated counts is divided by the shot number.
   - For the analog data the display can be either in mV or in least significant bits (LSB) and may be selected using the most top-left switch:



The LSB scale corresponds to the ADC reading, for a 12 bit transient recorder it can vary between 0 and 4095. A change of the *Scale* setting will restart the selected *Device*.

6. Change the input sensitivity with the *Range* knob located at the bottom left.

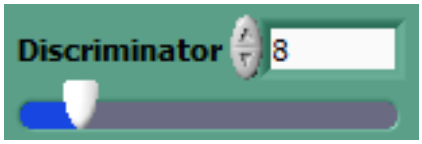


The three displayed millivolt values indicate the full scale negative voltages. A change of the *Range* setting will restart the selected *Device*.

7. If you have a photomultiplier connected to the signal input on the transient recorder you may switch to photon counting mode.

A change of the *Mode* will restart the selected *Device*.

8. Using the *Discriminator* slide, you can set the discriminator level between 0 and 63.



The discriminator setting influences the count rate in photon counting mode. For details about setting the discriminator refer to the Pulse Height Distribution section. A change of the *Discriminator* setting will restart the selected *Device*.

9. Turn the *Damping* switch

to reduce the counting rate since the discriminator level is set four times as high.

10. The *Set Overview Values to 0* switch helps you to analyze whether the signal exceeds the acquisition range.

Once you have acquired a real signal and the overflow at the transient recorder flashes switch the *Set Overflow Values to 0* switch and see if you see spikes towards zero. Those spikes will give you the exact position where the overflow occurred. With this information you can decide if you have overflows or underflows. Overflows can be avoided by increasing the signal input range or lowering the amplitude of the input signal underflows might require recalibration of the instrument as shown in https://www.licel.com/manuals/TR40-16bit3U_Manual.pdf (section 4.3 Analog Background).

Underflows are a serious thread to signal integrity as they are hard to notice and might distort the background computation. They should be fixed as otherwise the far field signal will be incorrect.

A change of this setting will restart the selected *Device*.

11. The following switch is available only if *TCPIP Live Display* is running as a sub module in `Licel Main`.

When switched `ON` it allows to keep the memory block setting from the configuration of `TCPIP Acquis` (default behavior when running as a sub module). The only condition to achieve keeping the memory block setting is that memory blocking is active for the selected transient recorder and that `Licel Main`'s tab page running `TCPIP Acquis` has been entered at least once. When the Licel Ethernet Controller does not support the `BLOCK` command the setting of the switch has no influence.

To make sure that keeping the memory block settings is active you should once

(a) Activate *Keep MemBlock* (ON)
(b) Change to the Acquisition Tab (TCPIP Acquis)
(c) Open the configuration in TCPIP Acquis
(d) Check *Block Trigger* for all desired transient recorders
(e) Save the configuration
(f) When leaving the configuration dialog of `TCPIP Acquis` the memory blocking will be set at the Ethernet controller
(g) Switch back to Live Display
(h) After starting you will only see shots at non-blocked memories

The next time you start *TCPIP Live Display* inside of `Licel Main` it is sufficient to switch once to the Acquisition tab page to achieve keeping `TCPIP Acquis`'s memory blocking.

Keeping the memory block settings may be switched off (*Keep MemBlock* (OFF)). To switch it on again, you must

(a) Activate *Keep MemBlock* (ON)
(b) Change to the Acquisition Tab (TCPIP Acquis)
(c) Switch back to Live Display

Please note: when running stand-alone *TCPIP Live Display* will not show this switch. All triggers received at any memory of the transient recorder will be enabled and included into the accumulated shot number.

12. *TCPIP Live Display* can additionally to the live data show up to four definable reference signals labeled *North*, *East*, *South*, and *West*. For that the following control elements are available:

To work with reference signals, proceed as follows:

(a) First of all select the *Active* reference signal by either checking the corresponding checkbox *North*, *East*, *South*, or *West*. All Button actions for *Save*, *Load*, and *Clear* will be applied to the active signal.

(b) A click on the button 

will save the current live data to a file and assign the data as the active reference signal. The corresponding *Visible* checkbox will be checked 

so the reference signal trace is shown together with the Live Data in the display. Furthermore an additional cursor becomes visible. The assignment and the cursor position will be saved for the next start of *TCPIP Live Display*. The file contains one column of ASCII data, it is saved in the sub folder `live`.

(c) A click on the button 

will allow to manually assign an existing live data file to the *Active* reference signal.

(d) A click on the button 

clears an assignment to the *Active* reference signal. The formerly assigned file is not deleted.

(e) All assigned reference signals can be displayed together with the live data. When a *Visible* checkbox is unchecked the graph will be set invisible while the cursor remains in the graphic display as the assignment of the reference data will.

13. The *Freeze* button  can be used to interrupt an acquisition to keep the current display unchanged for further inspection or to make a screenshot. An acquisition will be restarted after clicking on  .

14. Use the *Exit* switch or the Window close button ✕ to stop the program, when running as a Windows or Linux application the front panel window will close.

5.3 Acquisition Software

In this section, you will be introduced to the *TCPIP Acquis*, the *TCPIP MPush Acquis*, and *M-Acquis* software modules.

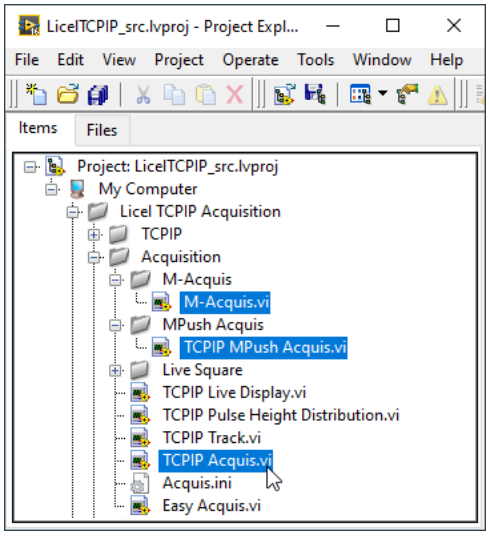
For a basic LabVIEW example refer to the example VI `Easy Acquis.vi` and extract the code you need for your own application.
The technical difference between these three programs is the way of receiving transient recorder data and the number of supported Licel Ethernet Controllers.

- *TCPIP Acquis* reads the data after explicitly sending a `DATA?` request to the Licel Ethernet Controller followed by reading the acquired data. Use this program to do acquisitions with many shots from transient recorders controlled by a single Licel Ethernet Controller.
- *TCPIP MPush Acquis* asynchronously reads all acquired data using a second Ethernet connection to the Licel Ethernet Controller. Use this program to do acquisitions with one or few shots from transient recorders controlled by a single Licel Ethernet Controller.
- *M-Acquis* (*Multiple Controllers Acquis*) works as *TCPIP Acquis* but supports controlling transient recorders at up to six Licel Ethernet Controllers.

Before starting acquisitions, you should configure the configuration parameters concerning the transient recorders and certain global parameters. In the case of using *TCPIP Acquis* or *M-Acquis* you may also define the program behavior in the case that no trigger is received, and the power meter integration.
If you have already aligned these parameters, you can directly jump to the TCPIP Acquis/M-Acquis or TCPIP MPush Acquis section to make an acquisition. Please note that both programs, *TCPIP Acquis* and `TCPIP MPush Acquis`, use the initialization file `acquis.ini`.

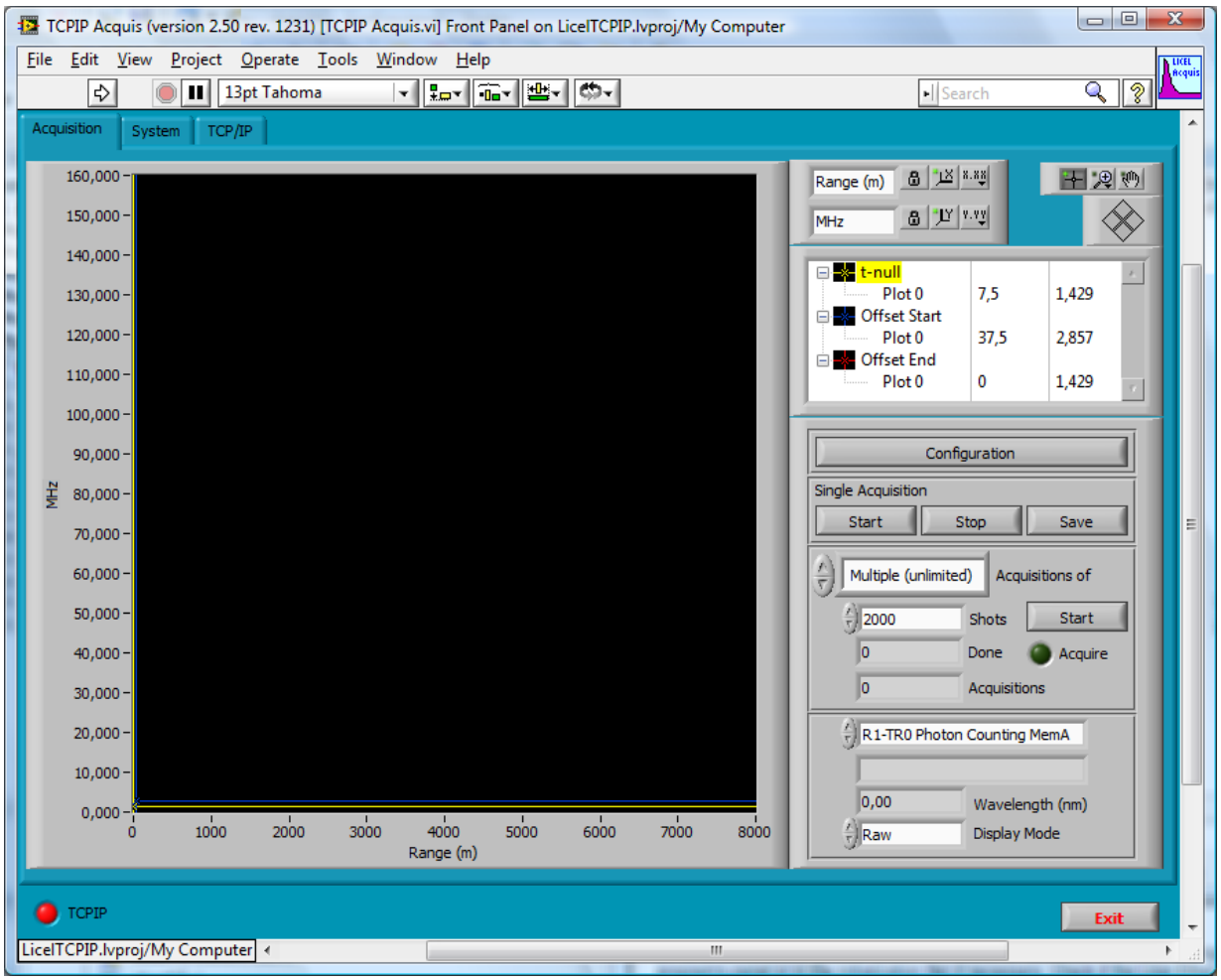
5.3.1 Starting TCPIP Acquis, TCPIP MPush Acquis, or M-Acquis

- If you are using the LabVIEW sources open *TCPIP Acquis*, *TCPIP MPush Acquis*, or *M-Acquis* from the LabVIEW project by navigating to the corresponding entry *TCPIP Acquis.vi*, *TCPIP MPush Acquis.vi*, or `M-Acquis.vi`, respectively, and double-clicking it.



- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

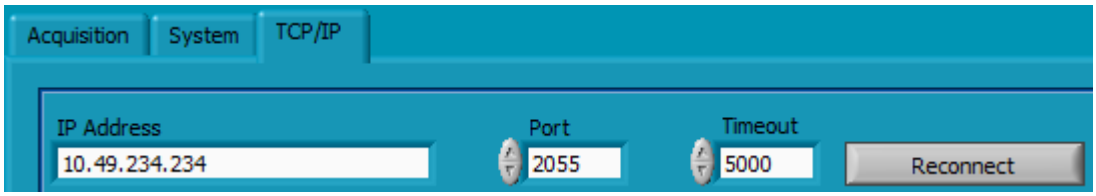
The Front Panel of *TCPIP Acquis* is seen in the next picture:



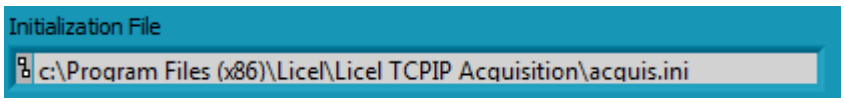
The front panel of *TCPIP MPush Acquis* is similar to that of *TCPIP Acquis* displayed above. The differences are important for acquiring acquisitions but not for establishing the TCP/IP connections described here.
The front panel of *M-Acquis* is similar to that of *TCPIP Acquis*, as well. The differences are described below.

1. First of the **IP Address** and the **Port** have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

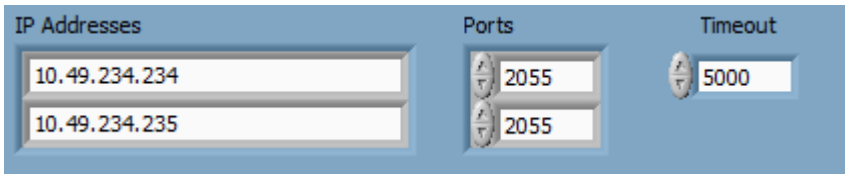
   - Using the LabVIEW vi, just enter the required values on the *TCP/IP* page and save them as defaults.



   - If you use the Windows application you must directly enter the correct values into the corresponding control fields on the *TCP/IP* page. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.
   You may also set the values in the initialization file `acquis.ini`. You will see the full path of the file in a file path indicator on the *TCP/IP* page.



- In *M-Acquis* up to six controllers are supported. Therefore, all IP Addresses and Ports must be set (here as an example: two controllers)

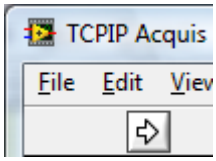


The number of controllers can be changed by entering the *No of Controllers* and using the *Change* button:

The number of fields to enter the IP Addresses and Ports will rescale after some time according to the number of controllers. All **IP adresses** and **Ports** will be saved to the initialization file `acquis.ini` when using the Windows application.

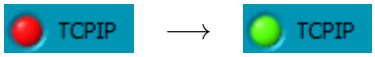
- If you run *TCPIP Acquis* within a sub panel on a page from *Licel Main* the latter is responsible for managing the TCP/IP connection(s).

2. To start the program press the **Run** button at the top left of the screen.



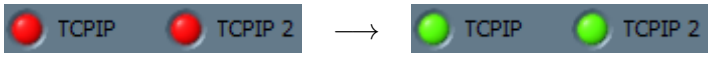
The Windows application will start automatically when called for the first time.

3. After a short time the **TCPIP** indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for **address** and **Port**, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct. The LED of the transient recorder should be lit up.



If you run *TCPIP Acquis* within a sub panel on a page from *Licel Main* the LED indicator is shown in *Licel Main*'s front panel.

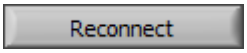
4. In the case *MPush Acquis* is in use, a second TCP/IP connection for receiving the transient recorder data is used. Therefore, the TCP/IP LED indicators will show up as follows:



If you run *MPush Acquis* within a sub panel on a page from *Licel Main* the LED indicators are shown in *Licel Main*'s front panel.

5. If you recognize that you are not connected to the Ethernet Controller(s) you would like to use, just

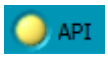
   (a) Enter the *IP Address*(es) and *Port*(s) of the controller(s) you really would like to connect to and

   (b) press the 

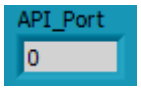
button.

Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

6. *TCPIP Acquis* and *M-Acquis* may be controlled externally via a TCP/IP API. If such a connection is active, the indicator at the top right of the front panel is displayed:



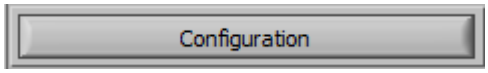
The used Port is shown on the *TCP/IP* tab page if *TCPIP Acquis* and *M-Acquis* are running stand-alone (not as a sub module in *Licel Main*):



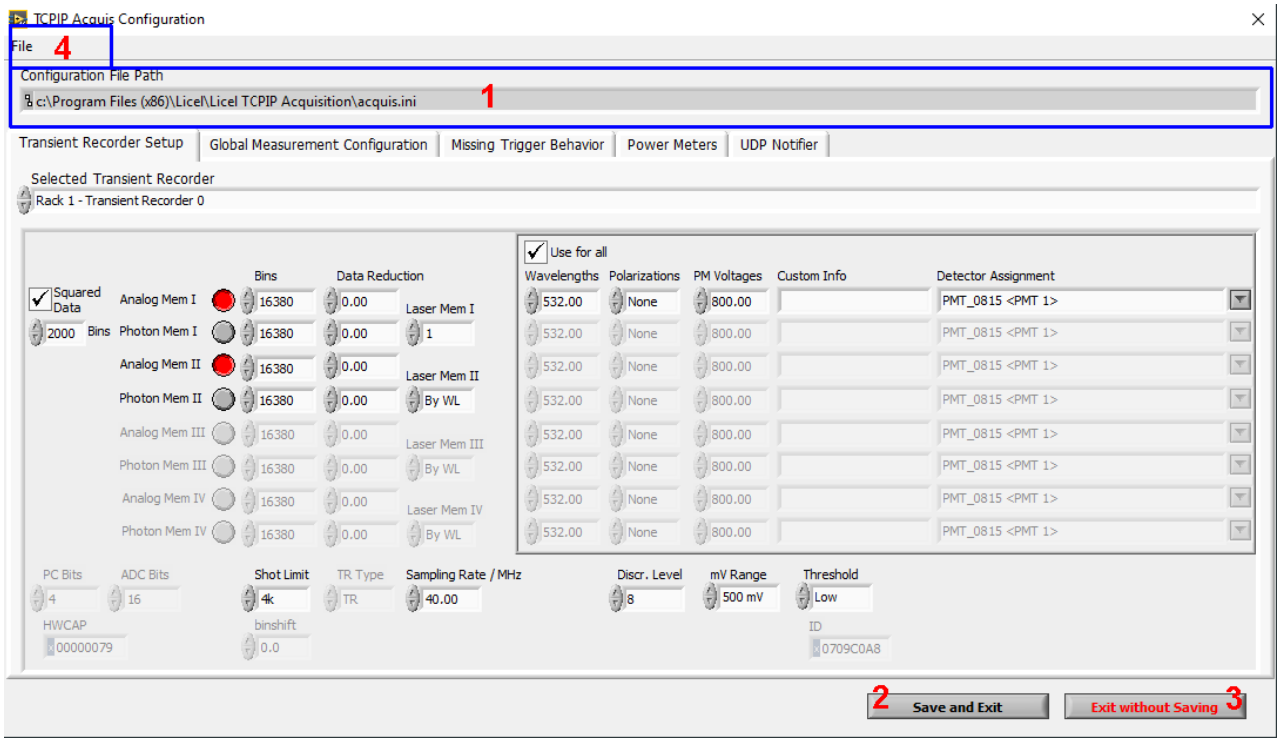
If the program is already running and not acquiring data then continue. Otherwise press the **Stop** button and then you may continue with the following steps.

5.3.2 The Configuration Dialog

The configuration parameters concerning the transient recorders, certain global parameters, the program behavior in the case that no trigger is received, and the power meter integration and are set in a separate program dialog. This configuration dialog is accessed by pressing the *Configuration* button you find on the front panel of the on the right-hand side. Setting the trigger behavior and integrating power meters is not supported by *TCPIP MPush Acquis*.



The configuration dialog will show the information found in the file `acquis.ini`. If a value is not found it will be set to a default value.
The configuration dialog is organized in terms of four tab pages each of them corresponding to one of the following sub-subsections.

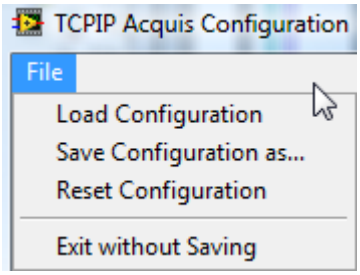


1. The full path of the initialization file is displayed at the top.

2. Use *Save and Exit* to close the configuration dialog and to save all configuration data to the configuration file.

3. Use *Exit without Saving* to close the configuration dialog without saving the configuration. Please note that when you exit the program without saving, any unsaved data is lost! Thus if you have configured the data and wish to keep it, you need to choose *Save and Exit*.

4. The *File* menu provides the following entries:

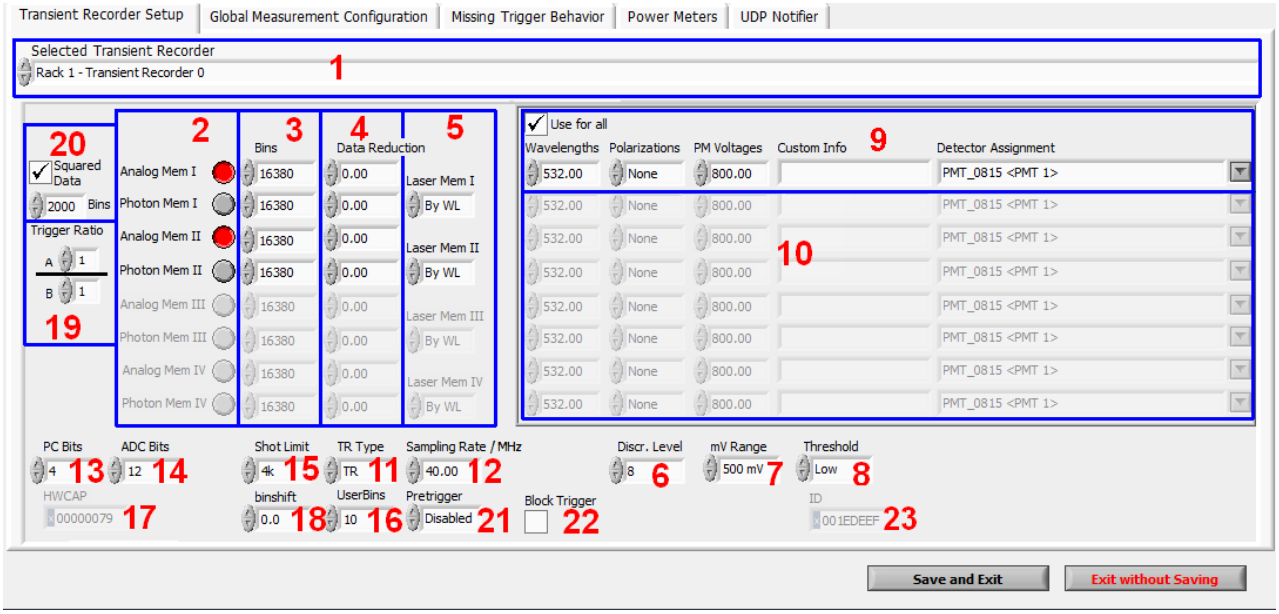


   - *Load Configuration* allows to select a configuration file anywhere in the file system to load the configuration parameters from there.
   - *Save Configuration as...* saves all configuration parameters to a selectable file in the file system.
   - *Reset Configuration* resets all changes since the configuration dialog has been opened.
   - Exit without Saving closes the configuration dialog without saving the configuration. Please note that when you exit the program without saving, any unsaved data is lost! Thus if you have configured the data and wish to keep it, you need to choose *Save and Exit* from above.

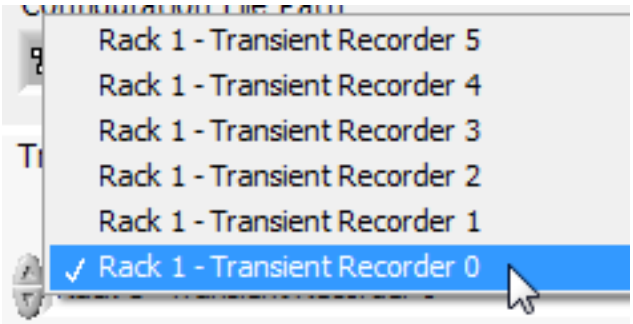
Please note: If you want the current configuration to be used as the default configuration for the next start of the program, you must save this data as `acquis.ini` in the directory where TCPIP-Acquis.llb or the Windows applications, respectively, are located.

Transient Recorder Setup

The *Transient Recorder Setup* is displayed on the corresponding tab page and contains all the information needed to configure the transient recorders. Several properties of the individual installed transient recorders can be set here. Some of the properties have fixed settings dependent on the capabilities of the transient recorder and the Licel Ethernet Controller. In such a case the corresponding input field will not be shown or is disabled and greyed out.



1. A selection list allows to select one of the installed transient recorder for configuration.



   Please completely configure a selected transient recorder before switching to the next one. If you call the configuration dialog from Licel's multi-rack acquisition software *M-Acquis* the list entries easily allow to identify the transient recorder's racks.

2. Select the memory banks for data transfer. The memory banks are Analog Memory A, Photon Memory A, Analog Memory B, and Photon Memory B. Analog Memory C and Photon Memory C are enabled if supported. For a PRxx-xx transient recorder (**TR Type**=*PR*, see position 11) use the photon counting switches, only.

3. Position 3 allows to specify the number of bins to read. The maximum number of bins is given by the *hardware defined tracelength*/($2^{\text{data reduction}}$). For old TR 20-160 the hardware defined tracelength is 16k bins. For newer its dependent on the memory length dip switch configuration see https://licel.com/manuals/TR40-16bit3U_Manual.pdf#subsection.3.4

4. Set the data reduction which allows for (software-) binning. A data reduction level of 0,1 and 2 corresponds to a height resolution of $1\times$, $2\times$, and $4\times$ the length corresponding to a primary bin. For a 40 MHz transient digitizer these values correspond to 3.75 m, 7.5 m, and 15 m, respectively. Each increasing in value reduces the height resolution by 1/2 and doubles the number of bins that are combined together to make a superbin. Thus the levels 0,1,2 correspond to 1 , 2, and 4 bins per data point, respectively. Binning results in an increase of the effective number of ADC bits written to the data files.

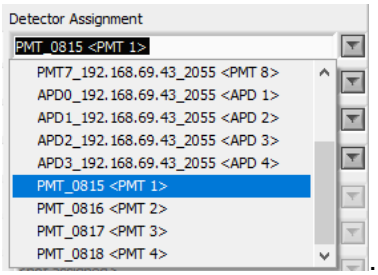
5. Assignment of the used laser to a memory. *By WL* will assign the laser by using the wavelength specified in regions 9 and 10. The further selectable list entries correspond to the lasers in the global configuration.

6. Set the discriminator level for the transient recorder. There are 64 discriminator levels (values 0 – 63) which correspond to either a range of 0 – 24 mV without gain reduction or 0 – 96mV with gain reduction. For details about setting the discriminator refer to the Pulse Height Distribution section.

7. Set the range value of the transient recorder. Valid values are 0 – 20 mV, 0 – 100mV and 0 – 500mV.

8. Setting the threshold mode is possible at position 8. In the *High* level the discriminator level is set four times as high.

9. The parameters in region 9 allow to enter further parameters which indicate the type of equipment that is used in the channel for memory A, analog acquisition. This information is stored as a header in the data files, so that the user (or anybody who evaluates the data) can see the parameters used while taking the data. The fields are used for the laser **Wavelength**, the corresponding **Polarization**, and the detector voltage (**PM HV**). **Polarization** has the allowed values *None* (0), *Parallel* (1), *Crossed* (2), *R*ight *Circ*ular (3), and *L*eft *Circ*ular (4). **Custom Info** can be used to specify any further channel dependent paramters. The information entered in these fields has no effect whatsoever upon the data acquisition. It is used purely to store information about the experimental setup in the data files. The content of **Custom Info** will be saved enclosed by quotes as entered by the user. The list **Detector Assignment** is intended to assign a detector to an acquisition channel:



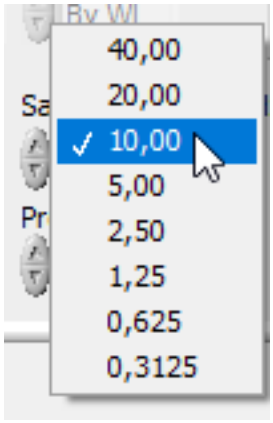
The assignments will be written to the initialization file. When the assignment information is loaded it is checked whether of not the assigned detector is listed in the initialization file `DetectorTR_Assignment.ini`. Whenever an acquisition is started and a detector has been assigned to an acquisition channel *Acquis* will read the current HV of the detector a channel is assigned to and update the acquisition channel's **PM Voltage** with the current HV value. The detector's ID and it's description will be added to the data file's header lines.

If the checkbox **Use for All** is checked the values for memory A, analog are used for all other acquisition channels. If not, the input fields in region 10 allow to enter individual values for each acquisition channel.

10. **Wavelengths**, the corresponding **Polarizations**, **PM Voltages**, **Custom Info**, and **Detector Assignments** for the memory A (photon counting) and memory B, and C and D (if available, analog and photon counting) channels. The values are set equal to those of region 9 if **Use for All** is checked. If **Use for All** is not checked the input fields can be used to enter individual values.

11. The **TR Type** is the transient recorder type. The transient recorder is either a TRxx-xx (analog and photon counting transient recorder) or PRxx-xx (pure photon counting device), **TR Type** must be set to *TR* or *PR*, respectively. Align the memory switches (position 2) to the **TR Type**.

12. The **Sampling Rate** is the sampling rate of the transient recorder. This value defaults to 40 MHz. If a transient recorder does not support the *frequency divider* feature (`HWCap ∧ 0x40`, see below) the sampling rate is a fixed property: if your transient recorder has a different sampling rate, you will need to change the value to the corresponding sampling rate. For instance enter *40* for a TR-**40**-xxx or *10* for a TR-**10**-xx. New transient recorders will report their sampling rates, then the corresponding value is not changeable. For a transient recorder supporting the *frequency divider* feature the **Sampling Rate** is a list of selectable values. Then a decrease of the **Sampling Rate** will result in an increase of the effective number of ADC bits written to the data files.



13. The number of photon counting bits **PC Bits** must be specified here according to the transient recorder. This value is automatically set if supported by the controller. Then this field cannot be changed manually.

14. The number of ADC bits **PC Bits** must be specified here according to the transient recorder. This value is automatically set if supported by the controller. Then this field cannot be changed manually.

15. The **Shot Limit** (*4k* = 4094 shots or *64k* = 65534 shots) is set here. The shot limit must be the same for all transient recorders.

16. The input field to set the **UserBins** is visible only if the corresponding feature is supported by the transient recorder. In that case the SETMAXBINS command is used to transfer the desired **UserBins** to the transient recorder.

17. The non-changeable **HWCAP** field indicates individual transient recorder capabilities. **HWCAP** equals zero for older transient recorders. The following transient recorder capabilities are coded:

`0x01`	separate shot counter B
`0x02`	separate shot counter C
`0x04`	separate shot counter D
`0x08`	pretrigger
`0x10`	memory blocking
`0x20`	squared data support
`0x40`	frequency divider
`0x80`	*reserved*
`0x100`	apd-flex

18. If supported by the controller and the transient recorder the **binshift** is read from the active transient recorder and the value cannot be changed in the configuration dialog. Otherwise the value is changeable. The binshift is saved to the data file in the variable lines. A non-changeable **binshift** can get a user defined offset settable in the configuration file `acquis.ini`.

19. The **Trigger Ratio** is only available if acquisitions are enabled for both memories, A and B and no separate shot counters are available. Then the ratio of the trigger frequencies A and B must be set according to the ratio of the trigger frequency inputs for memories A and B at the corresponding transient recorder.
This setting is not available in *TCPIP MPush Acquis*.

20. The **Squared Data** with corresponding **Bins** is available if the individual hardware capability `0x20` is available and squared data handling is supported by the controller (newer than 2019-12-17). If **Squared Data** is checked the summed squared counts will be read for active analog and photon counting memoryA-channels. The data will then be saved as $\sqrt{N\sum(x^2)-(\sum x)^2}$

(N: shots, x: counts in each bin). Reading this enables to calculate the standard deviation and standard error. In *TCPIP Acquis* and *M-Acquis* the standard error is shown. Refer to the appendix for calculation details.
This setting is not available in *TCPIP MPush Acquis*.

21. The **Pretrigger** is available if the individual hardware capability `0x08` is available and the PRETRIG command is supported by the controller. If the **Pretrigger** is enabled, the acquired trace consists of 1/16th of the transient recorder memory tracelength before the trigger and the remaining points after the trigger point. For a TR with 16k memory configuration to store 15 km after the trigger with 40MHz, you will need 15km / 3.75 = 4000 bins + 1024 bins pretrigger = 5024 bins. For 16bit TR shipped before 2018 the pretrigger is 128 bins long. Newer units have the scaled length of the pretrigger.

22. The **Block Trigger** use case for blocking rack trigger signals for each transient recorder individually is a system with 2-4 trigger signals for multiple lasers or for alternating laser wavelengths (DIAL).

    For example a DIAL system with two alternating wavelengths would use the trigger input ”A” to write lambda-on signals into memory 1 and use trigger ”B” to write lambda-off signals into memory 2. Without checking the ”block-trigger” checkbox all transient recorders in a rack would be triggered by trigger ”A” AND trigger ”B”.

    If you want a specific TR channel start only on trigger ”A” or only trigger ”B” you can check the ”Block Trigger” checkbox and disable the ”Analog Mem II” and ”Photon Mem II” checkboxes for this TR. It will then not respond to trigger B. Other TR channels where ”Analog/Photon Mem I AND II” are enabled will trigger on trigger ”A” AND trigger ”B”.

    The **Block Trigger** checkbox is visible if the individual transient recorder's hardware capability `0x10` is available and the memory blocking command is supported by the controller. When checked the triggers corresponding to the non-used memories are blocked at the individual transient recorder by using the `BLOCK` command.

23. The **ID** is a unique identifier of the transient recorder displayed in hexadecimal notation. The ID is not supported at older transient recorders nor when using a controller without support of the `TRTYPE?` command, then the field is not shown.

If you have filled the control fields above with appropriate values, your transient recorders should now be configured for the usage with *Acquis*.

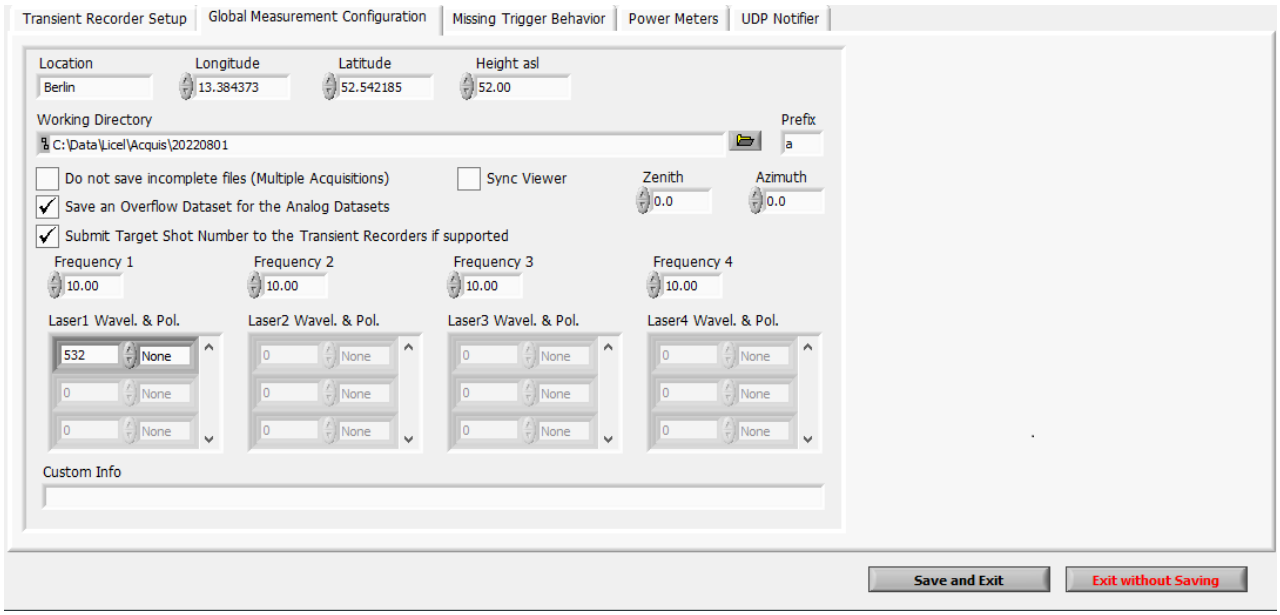
Monitoring the shot number During an acquisition *Acquis* will monitor the acquired number of shots. For this *Acquis* uses the first active transient recorder shown (item 2 from the transient recorder parameter list above with a number of bins > 0). We refer to this transient recorder as the *Leading Device*. Please make sure, that the leading device receives trigger pulses and that it receives them for memories which are not blocked. If more than one trigger is used in the acquisition system its important to have the fastest running trigger on the *leading device* as this transient recorder will determine when the desired shotnumber is reached.

Note that older transient recorders return the received number of shots of both triggers A & B (`I` & `II`).

Global Measurement Configuration

The global information allows you to set values that are stored in the data file headers which will tell you later about what sort of conditions were existent at the time the data was acquired. These are global values which usually do not vary from measurement to measurement and so it is named

Global Measurement Configuration. The *Global Measurement Configuration* is displayed on the corresponding tab page.



1. The **Location**, **Longitude**, **Latitude**, and **Height asl** (above sea level) represent the location of your acqusition system and will be stored in the data file headers.

2. The **Working Directory** is the location where you want data files to be stored and the **Prefix** contains one or two letters that will be used as a prefix for the file names. Directly enter the path of the **Working Directory** into the control field or browse your file system using the browse button. The format of the file names is
   `??YYmddHH.MMSSuuu`
   where `??` is the **Prefix**, `YY` is the year of the century, where `m` is the month (hexadecimal, 0 – C), `dd` is the day of the month, `HH` is the current hour of the day, `MM` are the minutes, `SS` the seconds, and `uuu` the first 3 decimal places of the seconds.
   For example the filename
   `a1981211.2816504`
   is a file that would have been taken on August 12, 2019. The operator set `a` to be the prefix (as in the screenshot) and the time was 11:28:16.504.

3. A checkmark at **Do not save incomplete files (Multiple Acquisitions)** will make *Acquis* not to save files when a multiple acquisition is manually stopped (available with version 1.70.01 and higher).

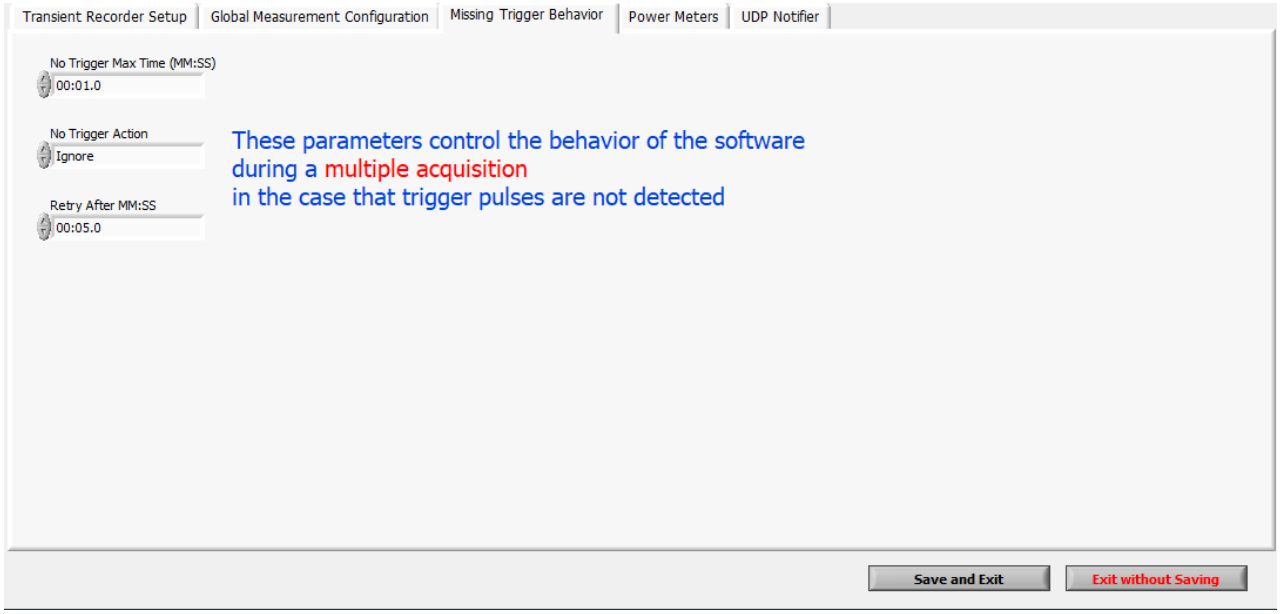
4. A checkmark at **Save an Overflow Dataset for the Analog Datasets** will make *Acquis* to add an additional dataset containing bitwise information whether or not an overflow appeared in a range bin of an analog dataset. For the first analog dataset in an acquisition bit0 (= 1) will be set, for the second bit1 (= 2) is set, and so on. *Acquis* and the Viewer will use this information to mark the affected range bins in the graphic display.

5. A checkmark at **Submit Target Shot Number to the Transient Recorders if supported** will make *Acquis* to submit the target shot number directly to the transient recorders when starting an acquisition using the `SETMAXSHOTS` command. Then, the transient recorders will automatically stop. `SETMAXSHOTS` will not be used even when the checkbox is checked if any of the active transient recorders do not support that feature. If the defined laser frequencies are not equal the checkbox will automatically be disabled.

6. A checkmark at **Sync Viewer** will make *Acquis* to start the *Advanced Viewer* to get synchronized with *Acquis*' data file notifier. Unchecking the box and saving the settings will make *Acquis* to close the *Advanced Viewer*. This option is not available for *MPush Acquis*.
7. The **Zenith** and **Azimuth** angles are the angles of the used telescope/sending and receiving optics. They are stored as header information in the data files.
8. The next group of input fields contains the repetition rates and wavelengths with polarizations of the lasers (**Frequency1**, **Laser 1 wavel**ength, **Frequency2**, **Laser 2 wavel**ength), **Frequency3**, **Laser 3 wavel**ength, and **Frequency4**, **Laser 4 wavel**ength (Laser 4 settings are available with version 1.70.01 and higher). At least one wavelength and polarization for laser 1 should be set. The allowed polarization values are *None* (0), *Vert*ical (1), *Hor*izontal (2), *R*ight *Circ*ular (3), and *L*eft *Circ*ular (4).
9. **Custom Info** is a free text field which is enquoted and attached to the second line of the data files.

Missing Trigger Behavior

The parameters to respond to missing trigger pulses during an acquisition are settable on the tab page *Missing Trigger Behavior*.
This setting is not available in *TCPIP MPush Acquis*.



The following parameters may be set here:

1. **No Trigger Max Time** Maximum time the software accepts missing trigger pulses. When this time has elapsed during an acquisition (*single or multiple*) the **No Trigger** LED in the acquisition program will become visible. When this time has elapsed during a *multiple acquisition* the **No Trigger Action** will be processed.
   The value is set in the format `MM:SS.u` with the minutes `MM`, the seconds `SS`, and the one-digit fractional part of the seconds.
2. **No Trigger Action** Action to apply after the specified **No Trigger Max Time** has elapsed without having received a trigger during a multiple acquisition.
   (a) *Ignore* The missing trigger is ignored if the already acquired shots are less than (target **Shots** -32). Otherwise the running acquisition is stopped, data is saved with the currently acquired shots, and a new acquisition is started.

(b) *Stop – NoSafe* The running acquisition is stopped without saving data.

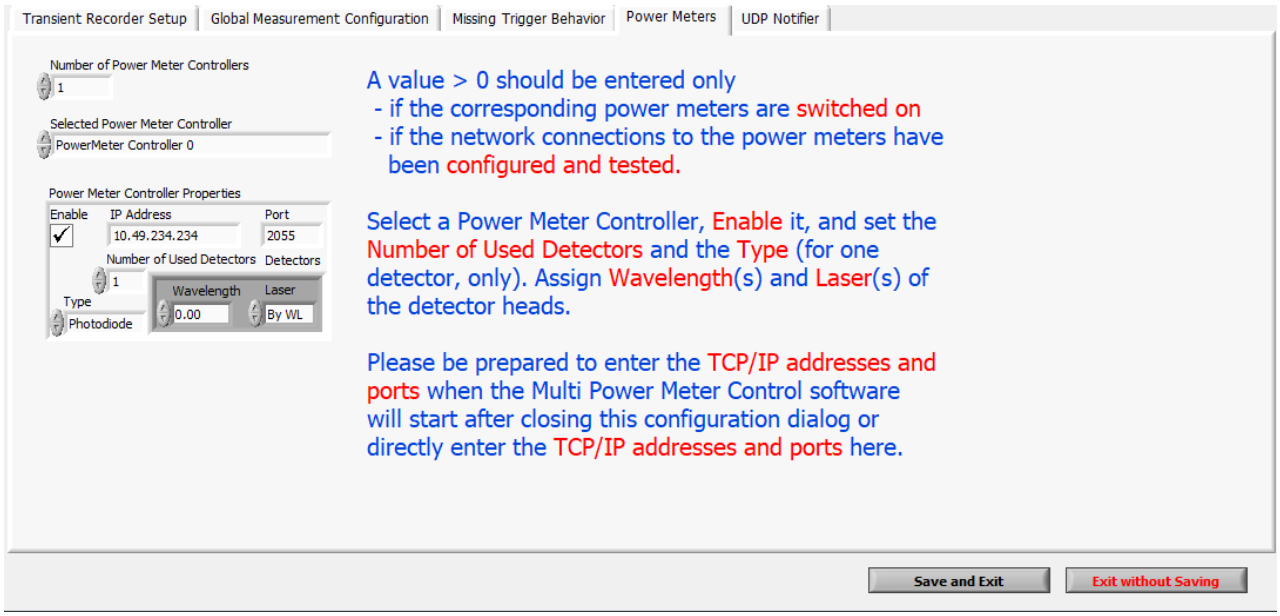
(c) *Stop – NoSafe – Retry* The running acquisition is stopped without saving data. After the **Retry After** time a new acquisition is started.

(d) *Exit – NoSafe* The running acquisition is stopped without saving data. The acquisition program is terminated.

3. **Retry After** Restart an acquisition when the specified time has elapsed after the previous acquisition has been stopped in the case that the **No Trigger Action** is set to *Stop- NoSafe - Retry*
   The value is set in the format `MM:SS.u` with the minutes `MM`, the seconds `SS`, and the ono-digit fractional part of the seconds.

Power Meters

The Acquisition software is capable to start and communicate with 1 or more instances of Licel's `Power Meter Control` software and to save the power meter data together with the transient recorder data.
This setting is not available in *TCPIP MPush Acquis*.

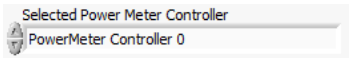


1. First of all it is required to configure the TCP/IP connection from the PC to the Power Meter Controller hardware. Please refer to the network setup and make sure that each IP address is uniquely used.

2. Please test the connection to each Power Meter Controller using the `Power Meter Control` software.

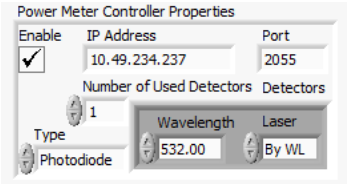
3. Top-right you may specify the **Number of Power Meter Controllers** you will use.
   

4. When the number is > 0 further control fields are available:

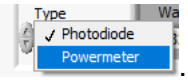
   - A selection list to select the power meter controller you would like to change:
     

     .

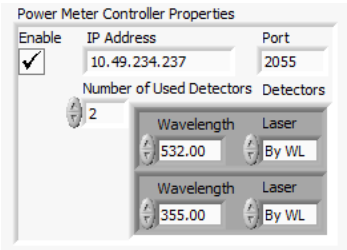
- A control field to edit the power meter controller properties of the selected controller:



Here, the **Enable** (checkbox), the **IP Address** and the TCP/IP **Port**, the **Number of Used Detectors** and the **Type** of the power meter controller's sensor are set:



- The **Wavelength** and **Laser** assignment can be defined for each detector of the power meter controller.



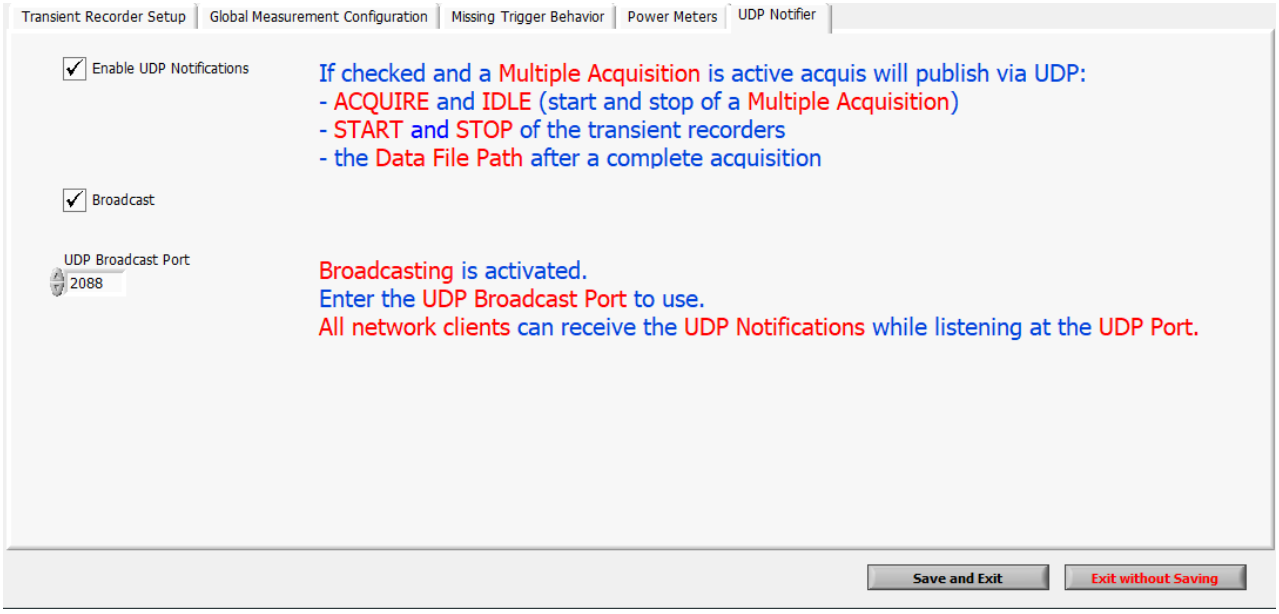
If you change the **Number of Used Detectors**, the list of **Wavelength** / **Laser** assignments is scaled. Please note that a number > 1 is allowed only for power meter controllers supporting at least this number of detectors. It is allowed to use less than the number of supported detectors.

- The sensor **Type** cannot be changed for power meter controllers with more than one sensor.

5. After leaving the configuration dialog the *TCPIP Acquis* will prepare an initialization file for the required software `Multi Power meter Control`, and the latter will be run in a sub panel on the tab page *Power Meters* in the acquisition software. If it is started for the first time **you will have to check the IP addresses and ports of the involved Power Meter Controllers**. Please refer to the subsection 6.4.3 for the details.

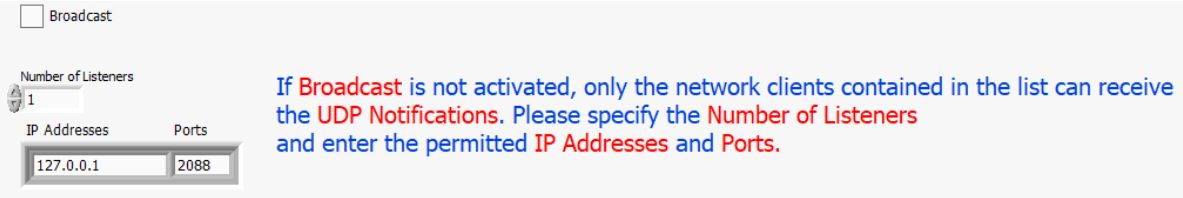
UDP Notifier

The Acquisition software is capable to publish the path of a data file via UDP once a multiple acquisition has been saved to disc. Furthermore `START` and `STOP` will be send when the active transient recorders are started or stopped during a multiple acquisition.
This setting is not available in *TCPIP MPush Acquis*.



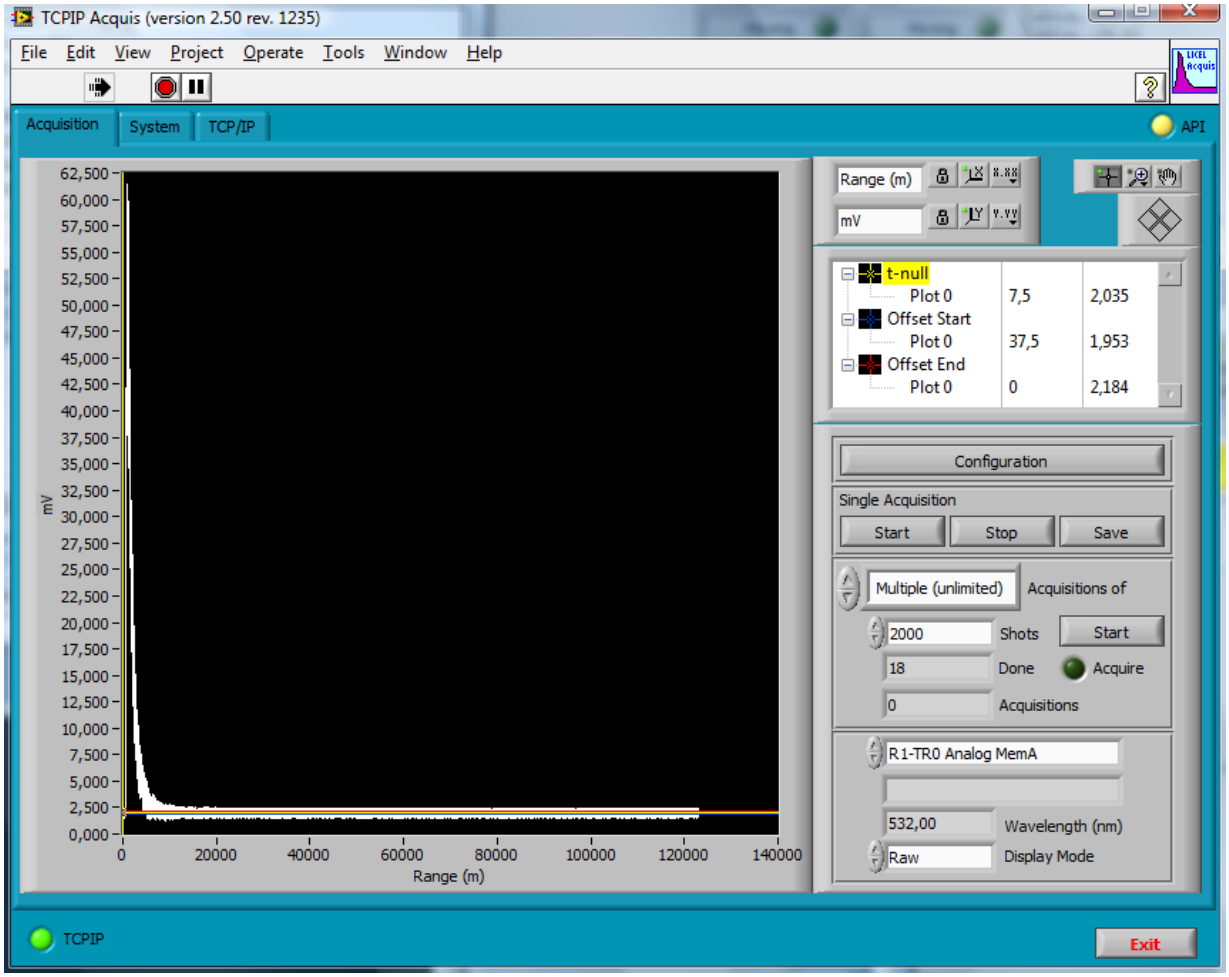
- Publishing the file path of a data file and `START` and `STOP` notifications via UDP ist enabled or disabled by checking or unchecking *Enable UDP Notifications*.

- Once *Enable UDB Notifications* is activated the *UDP Port* must be set. Please note that the selected port must not be blocked by firewalls in the network.
- If *Broadcast* is activated as shown above the file path will be published via UDP so that it can be received by any client in the network.
- If *Broadcast* is inactive the permitted clients must be specified by their *IP Addresses*. Please set the *Number of Listeners* and enter one IP address per line.

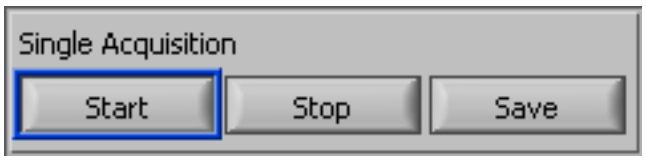


5.3.3 Acquisitions with TCPIP Acquis and M-Acquis

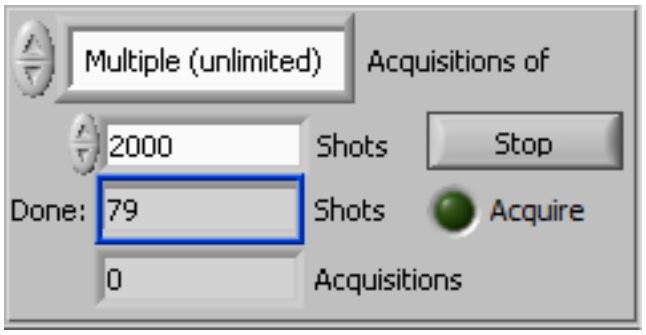
Now that the configuration is complete, it is time to take the first acquisitions with TCPIP Acquis or M-Acquis in the case that you are using more than one controller.
As the front panel elements for controlling acquisitions in M-Acquis and TCPIP Acquis are the same, there is no extra section for M-Acquis.
The Front Panel of TCPIP Acquis is shown again:



1. Press the **start** button in the **Single Acquisition** section of the right-hand side panel.



The transient recorders having activated data sets should now acquire data if a sufficient trigger signal is connected to the input. If the acquisition has been started, the **number of shots done** in the **Multiple Acquisitions** sections of the right-hand side panel should start increasing. Please note that the **Acquire** Indicator is for multiple, automatically saved acquisitions, only.

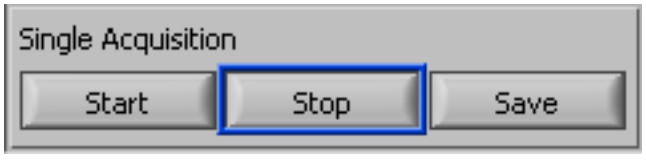


Monitoring the shot number During an acquisition *Acquis* will monitor the acquired number of shots. For this *Acquis* uses the first active transient recorder shown (item 2 from the transient recorder parameter list above with a number of bins > 0). We refer to this transient recorder as the *Leading Device*. Please make sure, that the leading device receives trigger pulses and that it receives them for memories which are not blocked. If more than one trigger is used in the acquisition system its important to have the fastest running trigger on the *leading device* as this transient recorder will determine when the desired shotnumber is reached. Note that older transient recorders return the received number of shots of both triggers A & B (`I` & `II`).

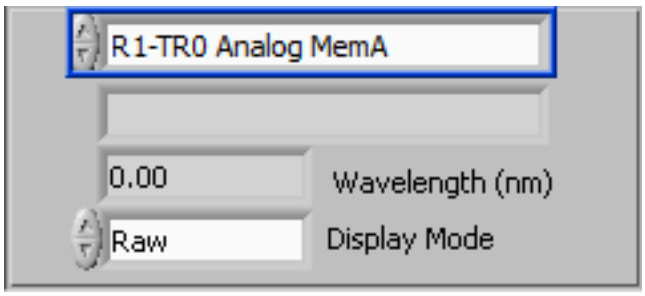
If the acquisition software does not receive an increasing shot number for a time larger than the **No Trigger Max Time** specified in the configuration an LED indicator becomes visible at the top right corner of the front panel window:

The data acquisition of the individual transient recorders can be checked by seeing if the Acquire LED of the specified transient recorder is brightly lit up. If not, the trigger is either insufficient, or the data sets are not activated as described in the section Changing the Transient Recorder Information.

2. Stop the acquisition in the same group by pressing the corresponding button.



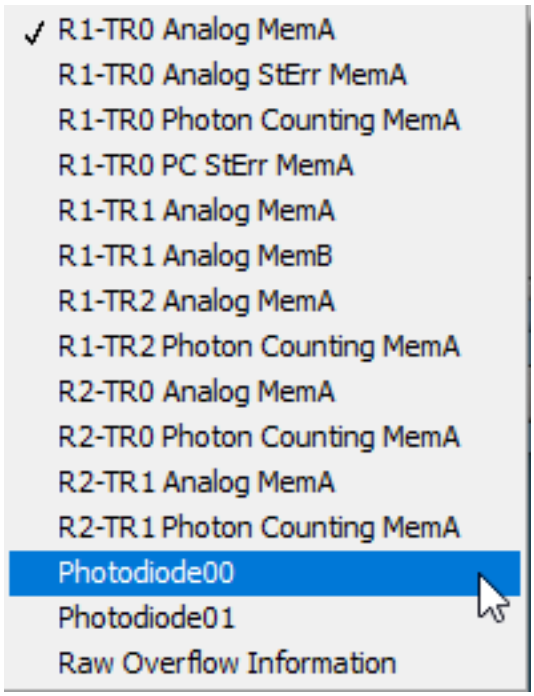
3. By changing the **data set selection**, you can now view the various data sets that were acquired. The entries in the selection list contain information about the used transient recorder (with it's rack), the acquisition mode (analog/photon counting) and the read memory. The wavelength is indicated in a seperate field.



4. The entries in the list are dependent on

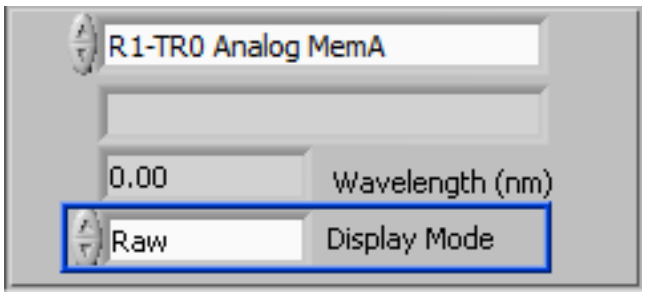
   (a) the activation of individual *Analog and Photon Counting channels*,
   (b) the availability and activation of *Squared Data*,
   (c) the integration of *Power Meter data* acquisition, and
   (d) the activation of the *Overflow* dataset (then, the raw overflow data can be inspected).

   In the case that power meter or Overflow data is acquired, as well, the list contains the corresponding entries at the end:

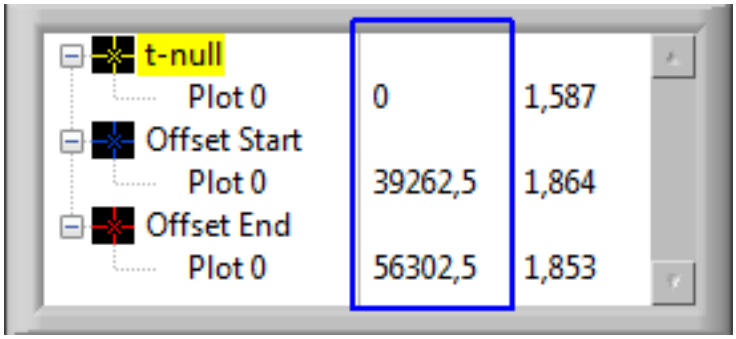


5. The horizontal scale of the data display is given in meters. The vertical scale is in mV for the analog and in MHz for the photon counting data. An incorrect scaling of the photon counting data might be the result of a wrong entry for the *Sampling Rate* in the transient recorder configuration (Configuring the Transient Recorders in section 5.3.2).

6. The **Display Mode** setting allows you together with the positions of the cursors to view the data in raw format (*Raw*), base line-corrected (*Offset Corr.*), or corrected to match the power loss due to the length of travel of the signal (*PR2*).

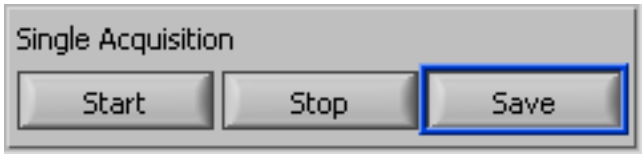


7. The background for the baseline correction is defined by the cursors *Offset Start* and *Offset End*, while the point-of-zero for the *PR2* mode corresponds to the cursor *t-null*. These values may either be set by moving the cursors in the graphic or by entering the appropriate values in the cursor control menu:

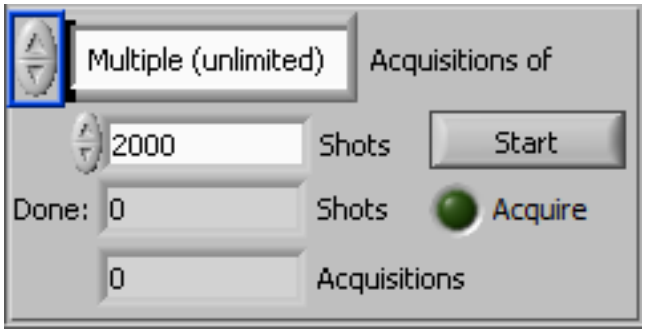


   Please refer to the Advanced Viewer section for a descriptive example.

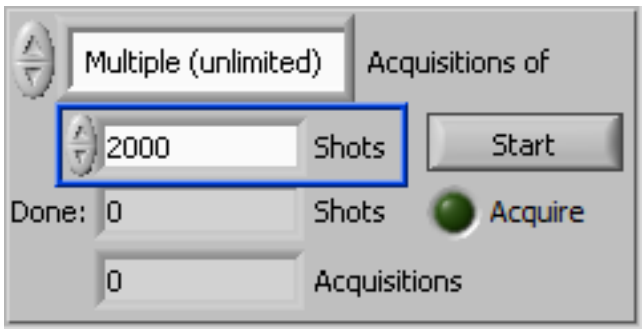
8. If you are satisfied with the data press **save**. The file is now renamed from temp.dat to a unique identifier according to Licel's data file format. The file contains the raw, uncorrected data as acquired by the transient recorders.



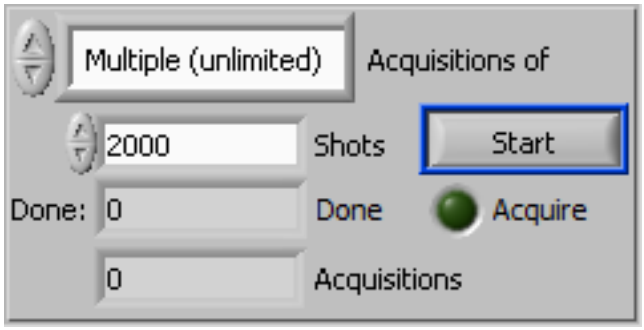
9. In order to automatically make multiple acquisitions, you must first decide whether to make acquisitions until the process is manually stopped or to enter the desired number of **Acquisitions**. Each acquisition will contain the specified number of **Shots** and will be saved to a separate file.



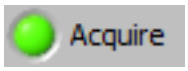
10. Then you will have to set the number of **Shots** which will be acquired for each file.



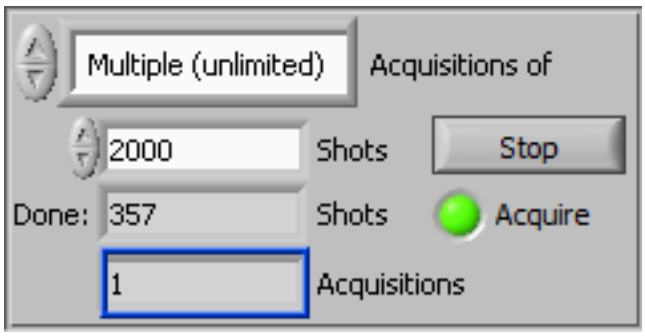
11. To start an automatic series of acquisitions, press the **Start** button in the multiple acquisition group. This button then turns into a **Stop** button.



The LED Indicator **Acquire** will change to light green. This indicator is for multiple acqusiitons, only.



The number of shots **done** will now start increasing and when the number *done* is equal to the number of **Shots**, the data sets will be written to a file. The number of shots per acquisition is not limited by the shot limit (4094/65534 shots) in the multiple acquisitions mode; once the transient recorder's shot limit is reached TCPIP Acquis will read the acquired data and shots and internally store them for later summation. Acquis will automatically restart the acquisition. When the desired shot number is reached Acquis will combine the acquired new data with any internally stored data and write it to the file. After this, the acquisition counter will increase and the program automatically starts acquiring the next data sets.



12. This process of automatically acquiring data sets of the defined number of **Shots** continues until you press the **Stop** button or — if specified — the number of acquisitions has been reached. In the case that acquiring data is terminated by pressing the **Stop** button the data acquired up to that time will be saved in a last file.

    If the acquisition software does not receive an increasing shot number for a time larger than the **No Trigger Max Time** specified in the configuration an LED indicator becomes visible at the top right corner of the front panel window:

In this case the action specified in the configuration will be processed.

13. You can change the data set configuration or global information by pressing the **Configuration** button. The changes to the data set configuration are applied to any acquisitions that you make after changing the configuration.

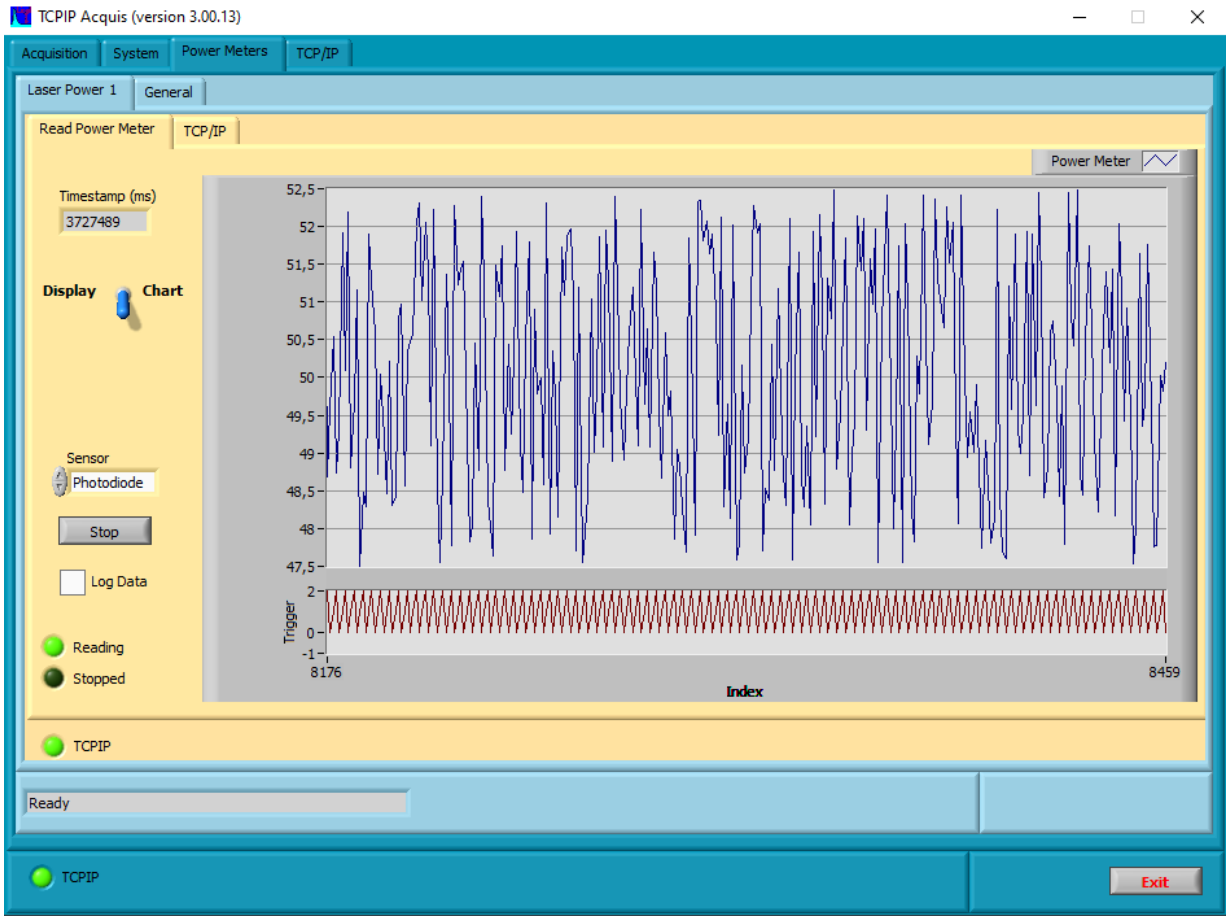
14. If an overflow at the ADC has been detected an overflow LED appears red at the top of the front panel window.

15. If you are done taking data and want to leave the program, press the **Exit** or the Window close button ✕ to stop the program, when running as a Windows or Linux application the front panel window will close.

You should now have a first impression of the capabilities of the Licel data acquisition software and the capabilities of the transient recorders. You can use all vi's as raw material for your acquisition software.

5.3.4 Power Meter Integration

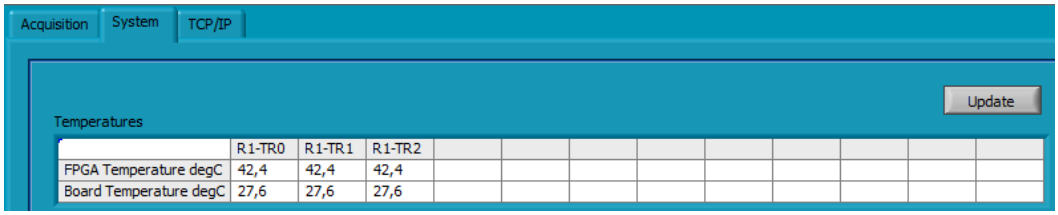
Once power meters have been configured the `Multi Power meter Control` software will be run in a sub panel on the tab page *Power Meters* in the acquisition software.



Please refer to the subsection 6.4.3 for the details.

5.3.5 Transient Recorder Temperature Monitoring

The most recent Transient Recorders and Licel Ethernet controllers support to read the FPGA temperature and at some transient recorders the board temperature, as well. If available, the temperatures are shown on the tab page *System*.



The temperatures in centigrade are updated when one enters the *System* tab or after pressing the *Update* button. Table field are empty if temperature reading is not supported.

5.3.6 Monitoring and Controlling TCPIP Acquis from Outside

TCP/IP Server

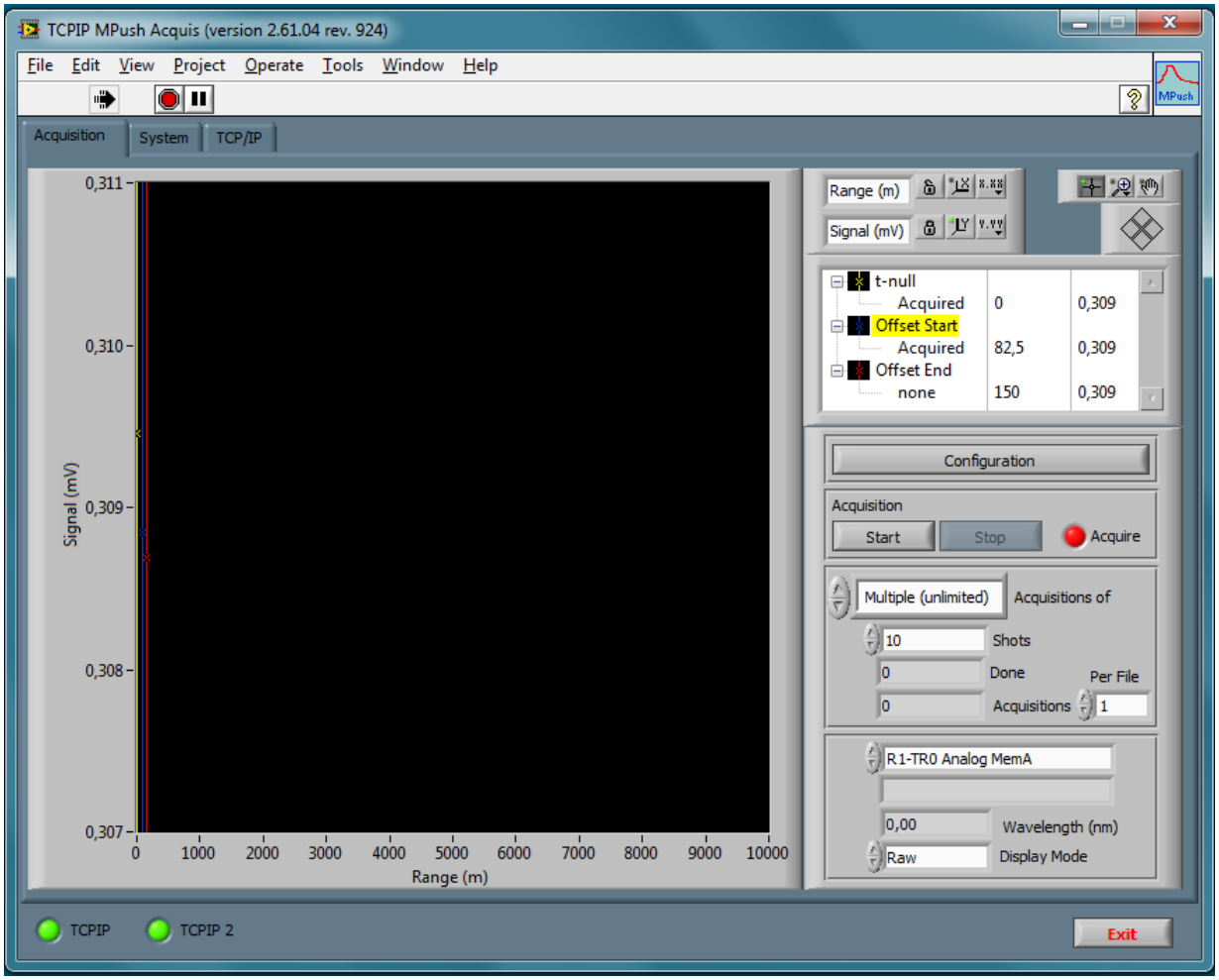
The basic functions of the TCPIP Acquis software can be accessed from third party applications via TCP/IP. For this TCPIP Acquis implements a TCP/IP Server listening on a defineable port (initialization file).
Please refer to the appendix Controlling TCPIP Acquis via TCPIP for the configuration of the TCP/IP server and the syntax of the supported commands.

Queue Control

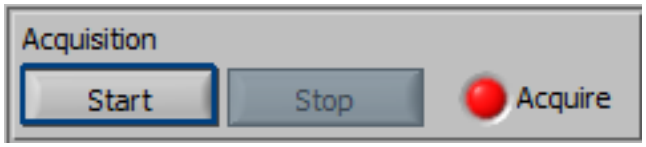
Some basic functions of the LabVIEW version of the TCPIP Acquis software can be accessed from third party LabVIEW VIs using LabVIEW's named queue mechanism. Since version 2.31 TCPIP Acquis uses a listening queue named `ACQUIS_LISTEN` to accept commands, and a reply queue `ACQUIS_REPLY` to send answers to the commands received via the listening queue. If your TCPIP Acquis is controlled by this queue mechanism please remember to wait for the reply to the command you sent. The queue interface is no longer extended.
Please refer to the appendix Controlling TCPIP Acquis from Outside for the syntax of the queue commands.

5.3.7 Acquisitions with TCPIP MPush Acquis

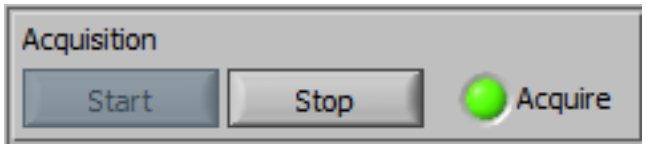
The Front Panel of TCPIP MPush Acquis is shown here:



1. Press the **Start** button in the **Acquisition** section of the right-hand side panel.

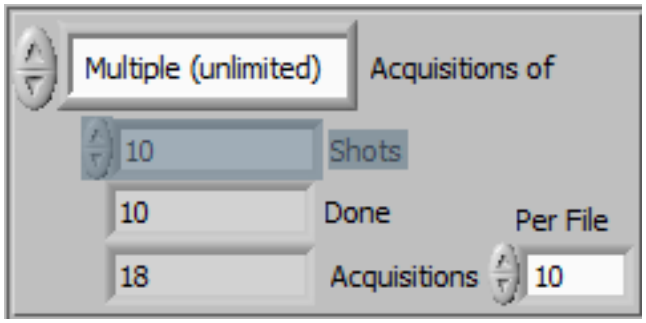


The **Acquire** Indicator should change to light green.



Stop the acquisition by pressing the corresponding button.

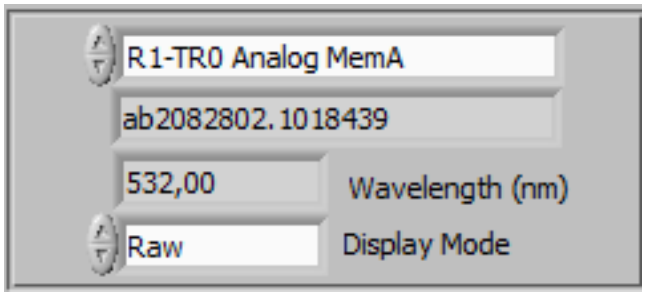
The transient recorders, in which you have activated data sets, should now acquire data if a sufficient trigger signal is connected to the input. If the acquisition has been started, the number of shots **Done** on the right-hand side panel should display the acquired shot number after some time. Each time tha target shot number (**Shots**) has been reached the number of **Acquisitions** should increase. Each time the number of **Acquisitions** is a multiple of the number of acquisitions **Per File**, a seperate file is created. You must enter the number of **Shots** before starting an acquisition.



2. With the setting seen above (*Multiple (unlimited)*) the program will acquire and save data until you press **Stop**. You may change this behavior and specify how many acquisitions with the specified number of **Shots** will be recorded by changing the *Multiple (unlimited)* selector: then, you will be able to enter the number of desired acqusiitions, and the program will automatically stop acquiring when that number is reached.

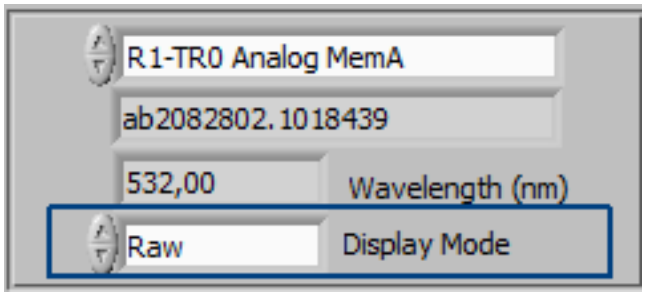
3. So, to summarize, the process of automatically acquiring and saving data sets of the defined number of **Shots** continues until you press the **Stop** button or — if specified — the number of acquisitions has been reached.

4. By changing the **data set selection**, you can now view the various data sets that were acquired. The entries in the selection list contain information about the used transient recorder (with it's rack), the acquisition mode (analog/photon counting) and the read memory. The wavelength is indicated in a seperate field.

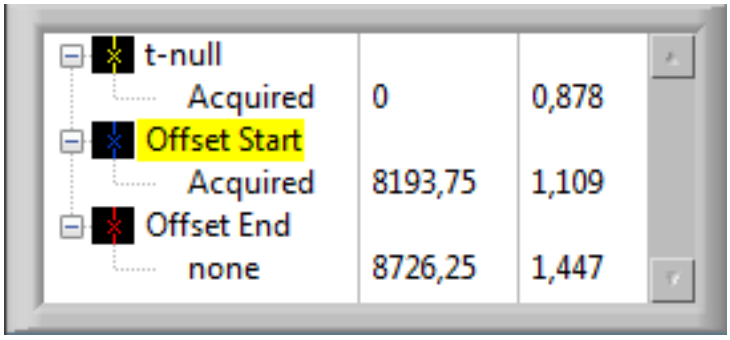


5. The horizontal scale of the data display is given in meters. The vertical scale is in mV for the analog and in MHz for the photon counting data. An incorrect scaling of the photon counting data might be the result of a wrong entry for the *Sampling Rate* in the transient recorder configuration (Configuring the Transient Recorders in section 5.3.2).

6. The **Display Mode** setting allows you together with the positions of the cursors to view the data in raw format (*Raw*), base line-corrected (*Offset Corr.*), or corrected to match the power loss due to the length of travel of the signal (*PR2*).



7. The background for the baseline correction is defined by the cursors *Offset Start* and *Offset End*, while the point-of-zero for the *PR2* mode corresponds to the cursor *t-null*. These values may either be set by moving the cursors in the graphic or by entering the appropriate values in the cursor control menu:



   Please refer to the Advanced Viewer section for a descriptive example.

8. You can change the data set configuration or global information by pressing the **Configuration** button. The changes to the data set configuration are applied to any acquisitions that you make after changing the configuration.

9. If you are done taking data and want to leave the program, press the **Exit** or the Window close button ✕ to stop the program, when running as a Windows or Linux application the front panel window will close.

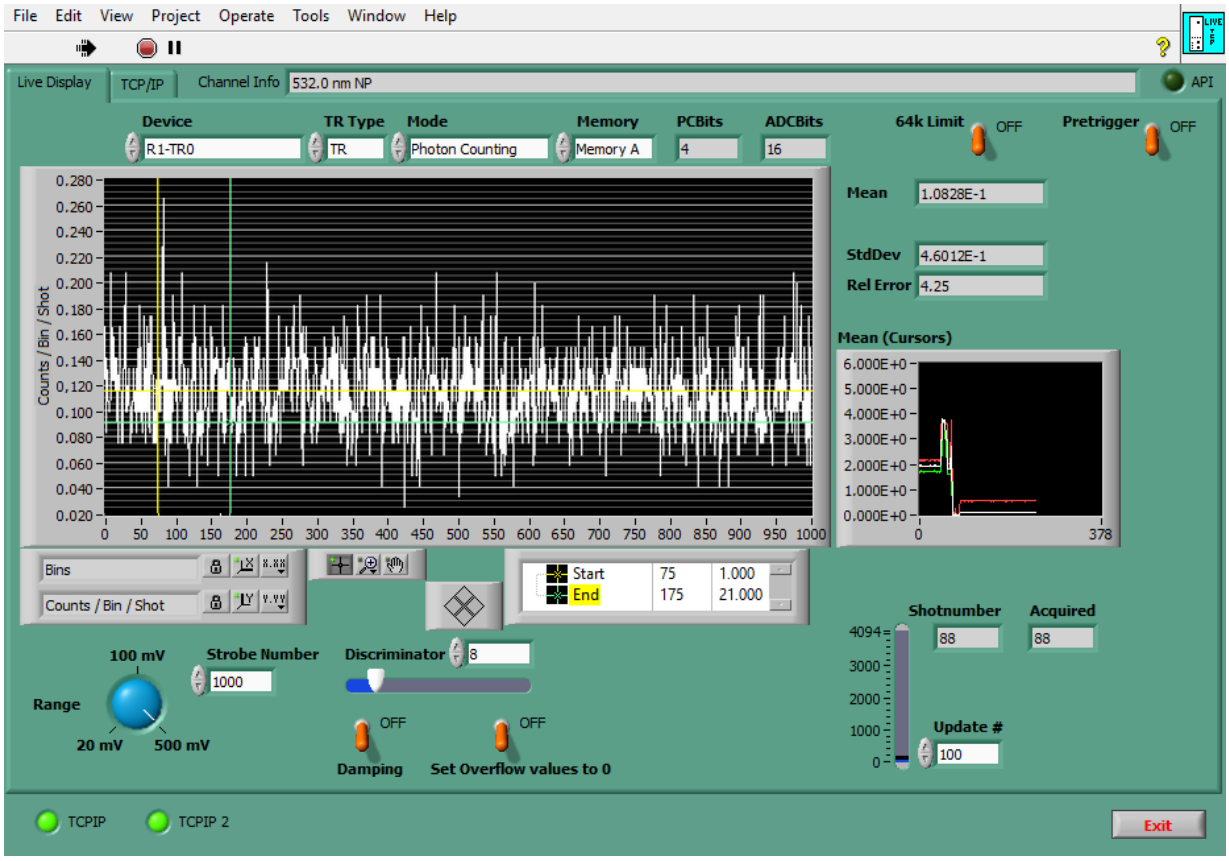
5.4 Pulse Height Distribution

When operating a PMT in photon counting mode the question is: Where to put the discriminator level? From the signal to noise ratio point of view the optimum will be a level where most of the noise counts are removed while only a minor part of the signal is lost. This point is called the valley point. The pulse height distribution can be shown using two modes:

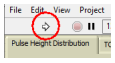
Integral Mode: All counts below the specified the discriminator level are shown.

Differential Mode: Here the counts with a pulse height between the current and the adjacent discriminator level are shown.
In order to acquire a pulse height distribution set the PMT to a desired voltage and illuminate the PMT using a continuous light level. We have found, that a signal with a mean count rate of 2 MHz or 0.1 count per 50 nsec bin is just fine. A sky with scattered clouds is definitely not stable enough. A LED powered by a DC supply is preferred.
So before running the PHD software the corresponding light levels should be set up. As reference level we use a discriminator level of 10, which corresponds to -3.96 mV. A typical dark PMT will show only few counts or none at all after 1000 shots.
When the continuous light is applied the signal rises to 0.1 count per bin or 2MHz count rate at

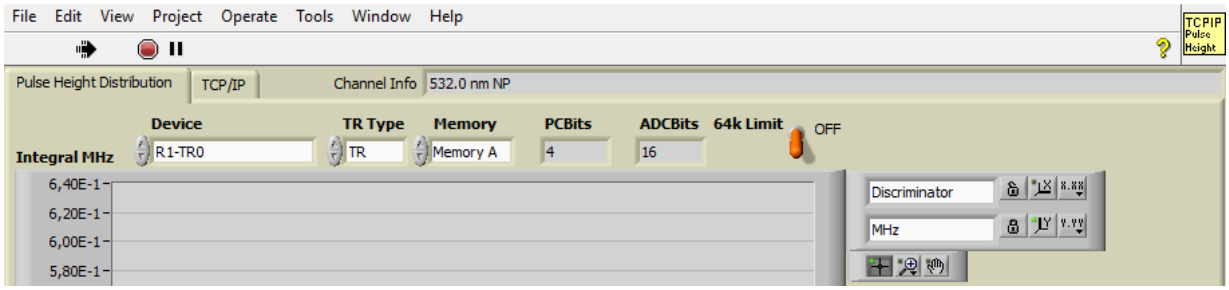
discriminator level 8 (-3.1mV).



Use the `TCPIP Pulse Heigth Distribution.vi` from the `TCPIP-Pulse.llb` or the `TCPIP Pulse Height Distribution.exe` if you are using the Windows applications.

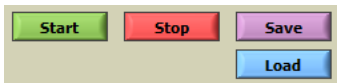

The first step is to press the LabVIEW `Run` button.

At the top of the *Pulse Height Distribution*'s panel you can set the parameters of the current transient recorder similar to the settings for *TCPIP Track*.



While a shot number of 100 shots per discriminator level is fine for an integral pulse height distribution, you should select significantly more for a differential pulse height, for instance 4000 shots. Acquire 4000 * 63 shots to get the PHD. A typical pulse height measurement using a 30Hz laser as the trigger source would require 2.3h of total acqisition time. The TR 20-160 can run at 300Hz, which will cut the acquisition time down to 840 sec or 14min. Using a faster trigger source is therefore recommended. The illumination level must be kept constant over this period of time as it would otherwise distort the the PHD.

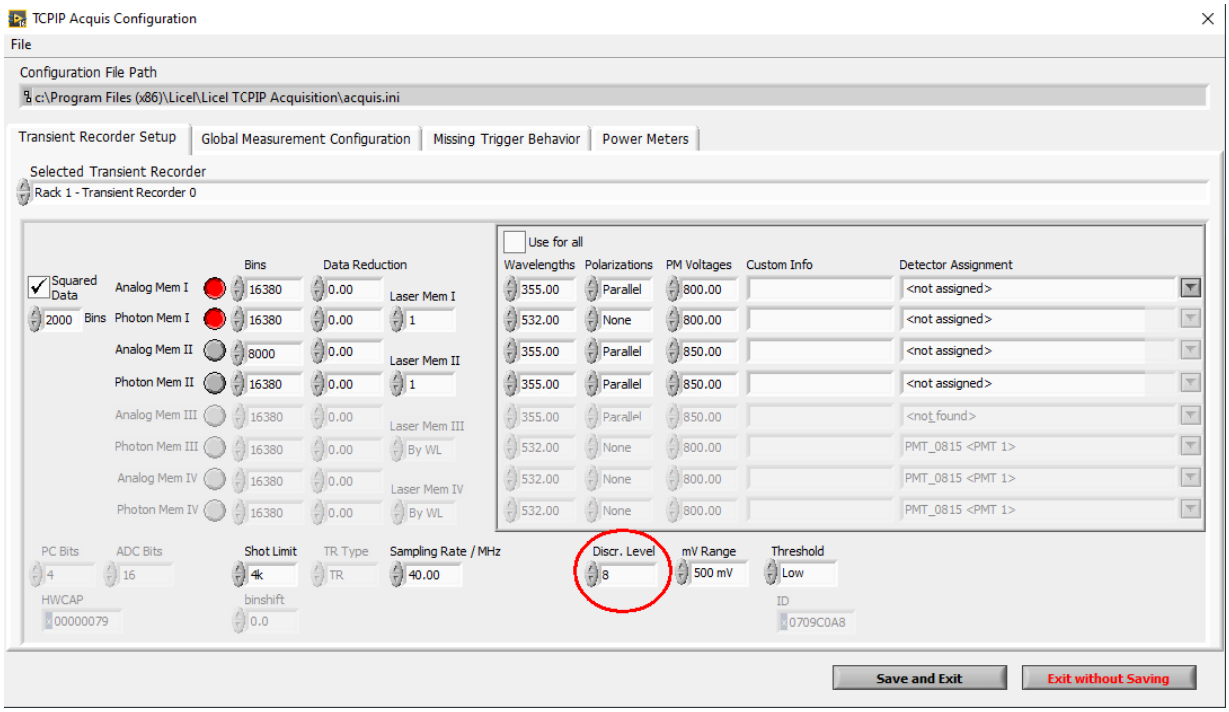
After all settings have been entered *Pulse Height Distribution* is controlled by the following buttons:



- A pulse height distribution is started with the *Start* button. For each discriminator level from 0 ... 63 the desired number of shots will be acquired. The acqusition stops after the acquisition for the discriminator level 63 has been completed or when the *Stop* button has been pressed.
- Use the *Save* button to save the (integral) data to a text file (each number in a separate line). The default file extension is `.phd`. The data values are saved with a `.` as decimal separator.
- Use the *Load* button to load a distribution from a text file and display it.

A typical sample of a integral PHD with 100 shots per discriminator level is shown below. Press the `Start` button to begin the data acquisition. The data display will be updated once the shots for a discriminator level are acquired.

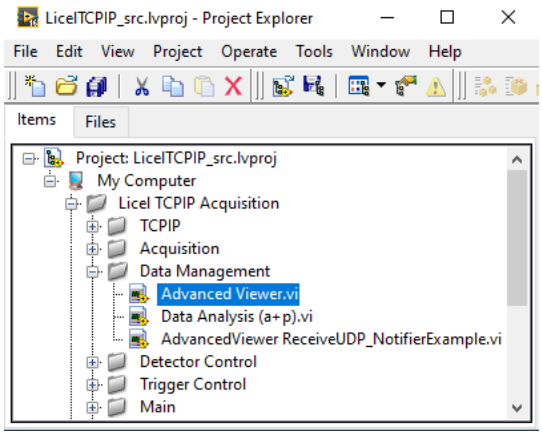
Please notice the change in the slope around discriminator level 10. The differential PHD for the same light intensity makes that more obvious. Select the differential field on the right side of the data display to get the differential PHD.
Here the valley point can be seen at discriminator level 10 which should then be used in the acquisition program. As the valley is not very sharp a discriminator level of 8 might also be a good choice.
If no clear valley point is visible but the differential PHD shows still a significant level at 63 then repeat the mesurement with the Damping switch in the ON position. This will give more complete coverage. The valley point then needs to be multiplied by 4 as the discriminator level is 4 times larger with damping on.
These values can then be used in the configuration data of the transient recorder in the acquisition program.



5.5 Advanced Viewer

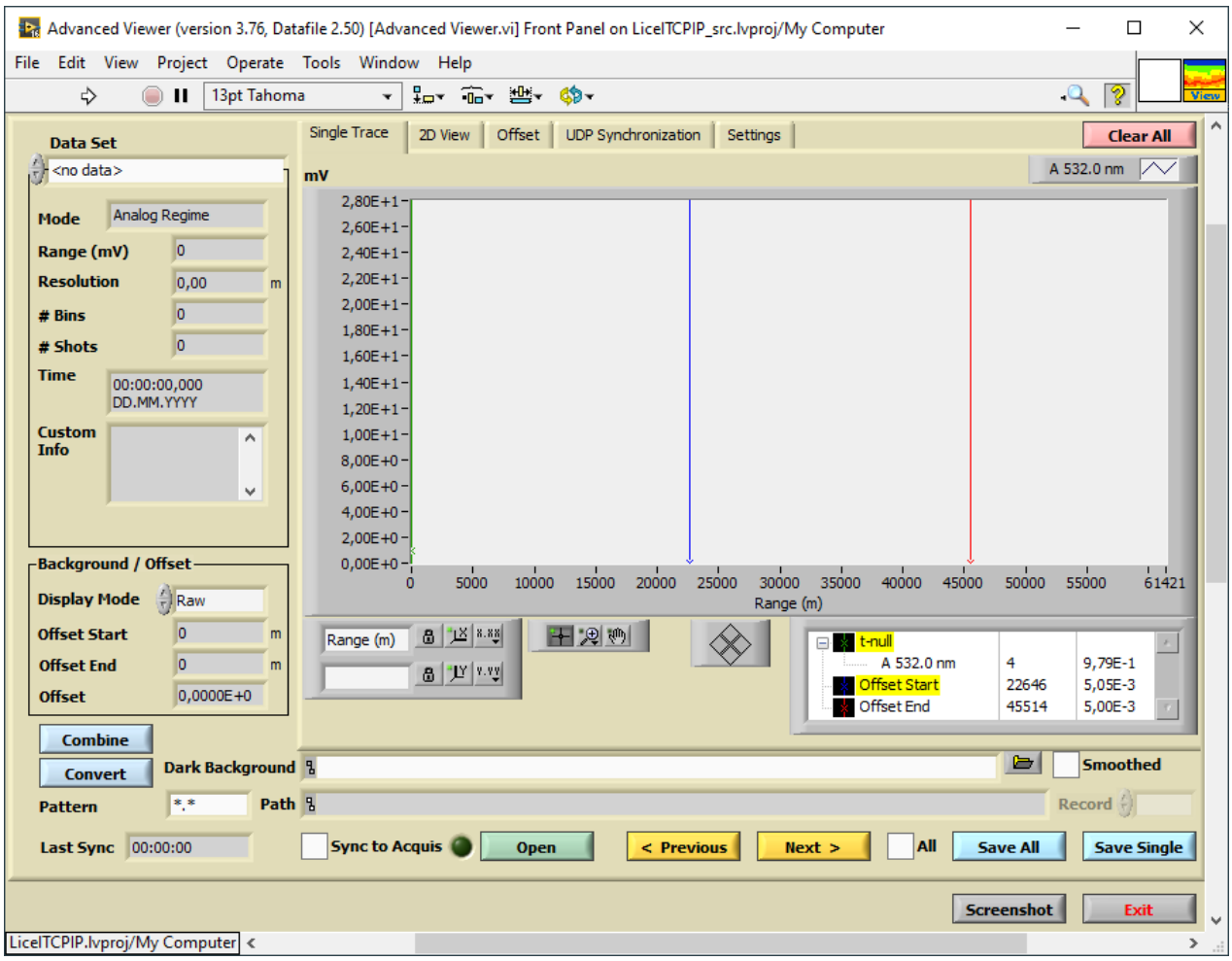
The Advanced Viewer is designed for reading data files created by Licel acquisition software like *TCPIP Acquis*. In order to demonstrate the capabilities of this program, you must already have acquired at least one dataset and saved it. For more information about acquiring data, please refer to the manuals for Acquis, Track and Live Display.

- If you are using the LabVIEW sources open the Advanced Viewer from the LabVIEW project by navigating to the corresponding entry *Advanced Viewer.vi* and double-clicking it.

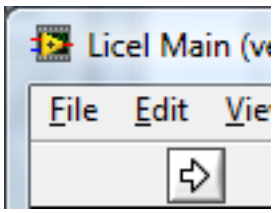


- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

You should see the following screen:

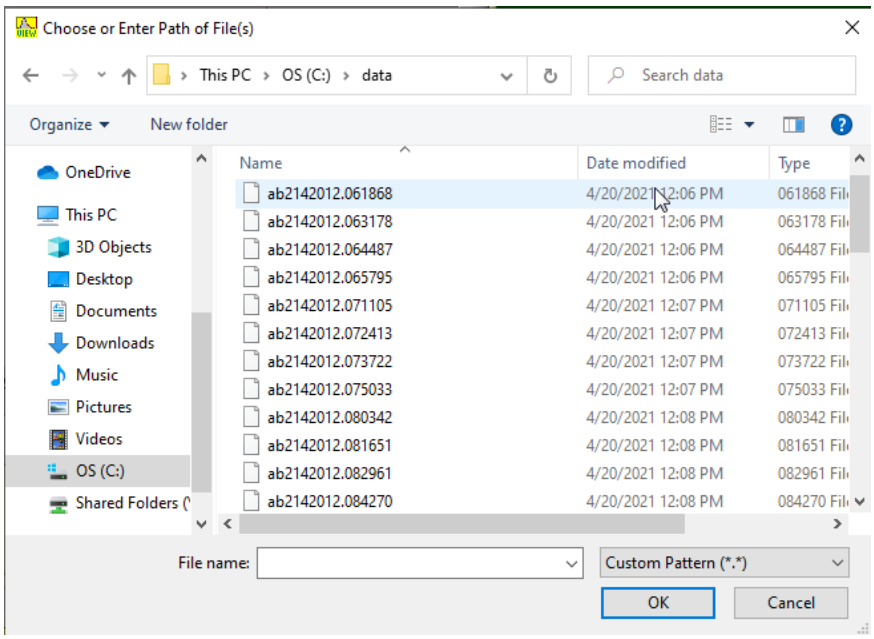


To start the program press the *Run* button at the top left of the screen.



The Windows application will start automatically when called for the first time.
A desired data file may be loaded by pressing the *Open* button.

A file selection dialog will appear. At the program start this dialog will come up without pressing the button.

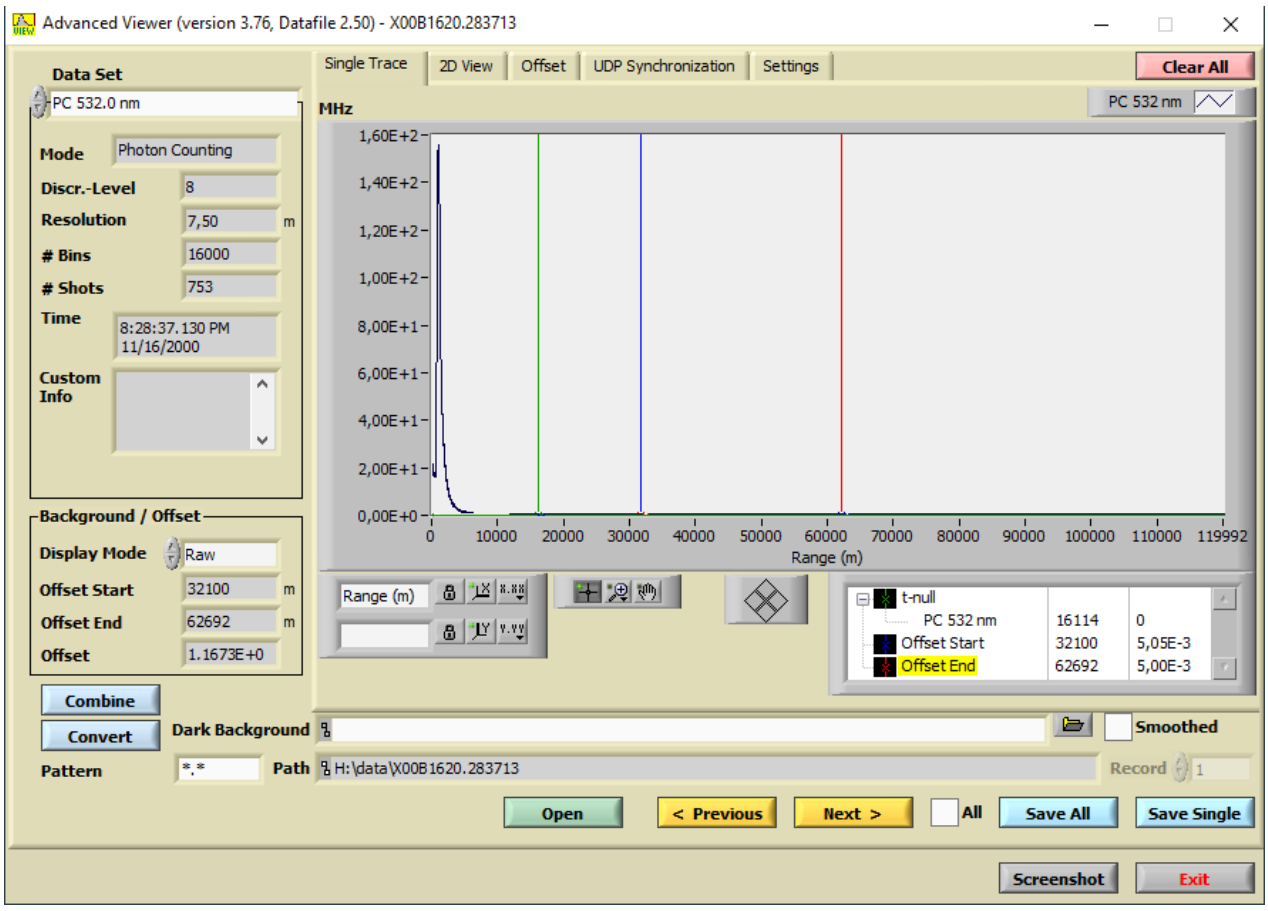


As start directory for browsing data files for the first time the *Advanced Viewer* uses the last storage directory of *TCPIP Acquis/M-Acquis/MPush Acquis* as set in the configuration dialog. If you would like to work independently of the acquisition software you could set the path in `acquis.ini`:

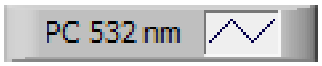
```
[global_info]
...
working_directory = "/C/temp"
```

The specified file *Pattern*

is used in the file dialog, it can be changed there.
Select a valid data file, press open and the dataset number specified by the *Data Set* control will appear in the graph indicator.



The signal type and wavelength are displayed in the graph legend

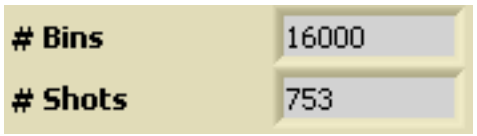


and the units used for the y-axis, which can be either megahertz or millivolts, are displayed in the upper left hand corner of the graph

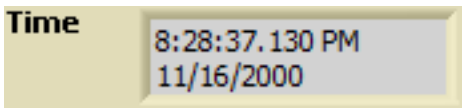
For photon counting data the discriminator level used during the acquisition is displayed

Furthermore the bin resolution is given in meters

and the number bins and the number of acquired shots are displayed:



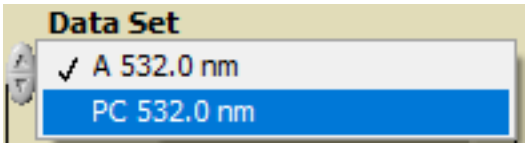
Additionally the acquisition's start time is shown.



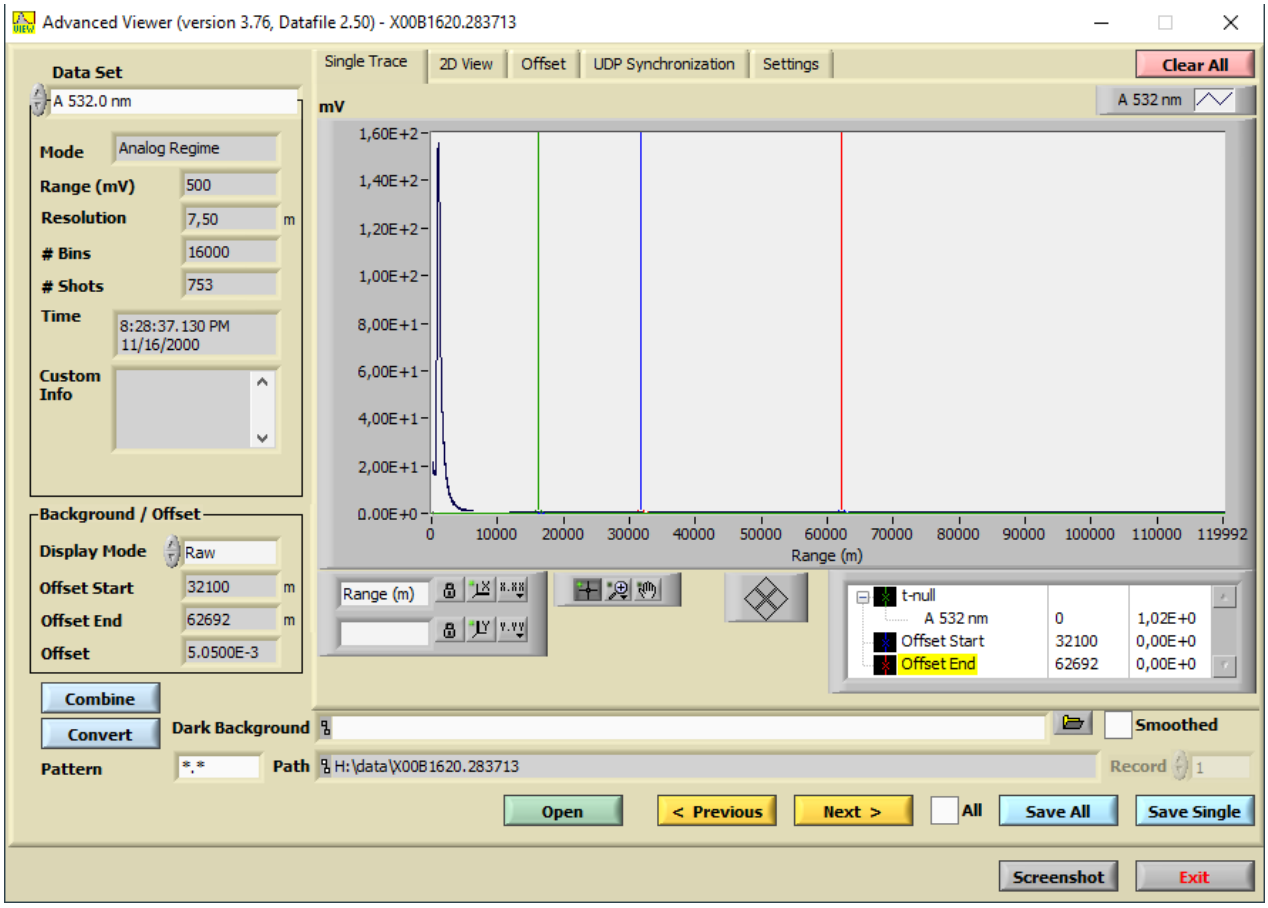
A scrollable field displays the custom info of the data set (here, it is empty):

The full path to the current data file is shown in the *Path* indicator.

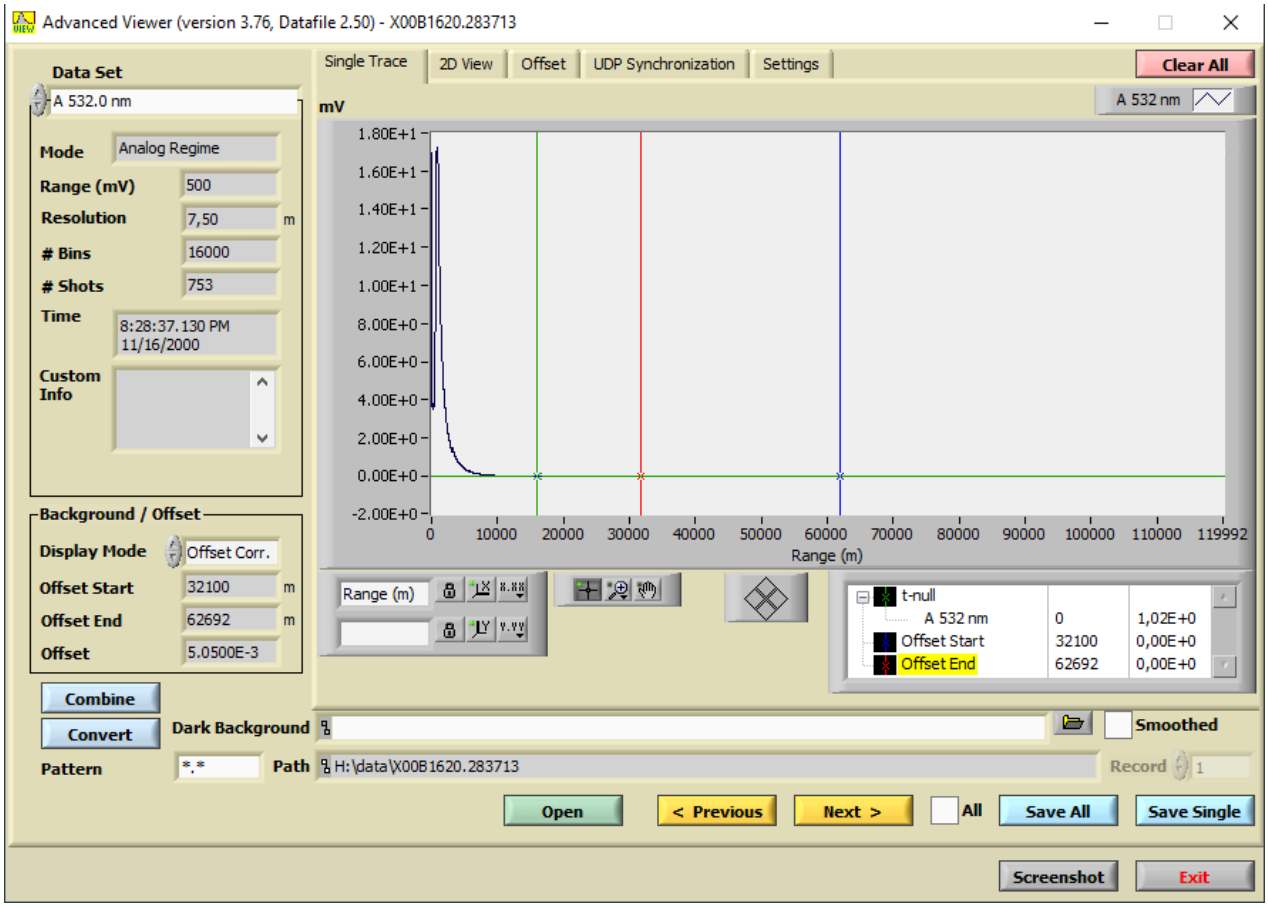
If you would like to see a different dataset from the file, use the *Data Set* control to choose it.



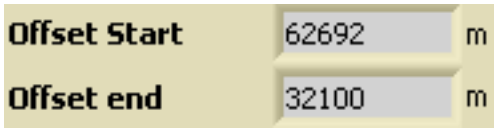
In this example, by switching to data set to *A 532 nm*, the analog 532 nm channel is displayed. Now, instead of the discriminator level the range used during the acquisition is displayed



Note that the legend has changed to *A 532 nm* and the units have changed to millivolts. Currently the data is being shown in raw mode and we see that in the example above, that the baseline appears to have a value of about 1000. The *Offset Start* and *Offset End* cursors can be used to correct for the baseline offset. If the *Display Mode Offset Corr.* is used, then the mean value of the signal between these two cursors will be subtracted from the signal in order to create a baseline corrected signal. Use the cursor controls to move the blue and red cursors (Offset Start and Offset End) to a region which will be evaluated to generate the new baseline. Change the *Display Mode* to *Offset Corr.* and observe that the baseline of the signal changes.



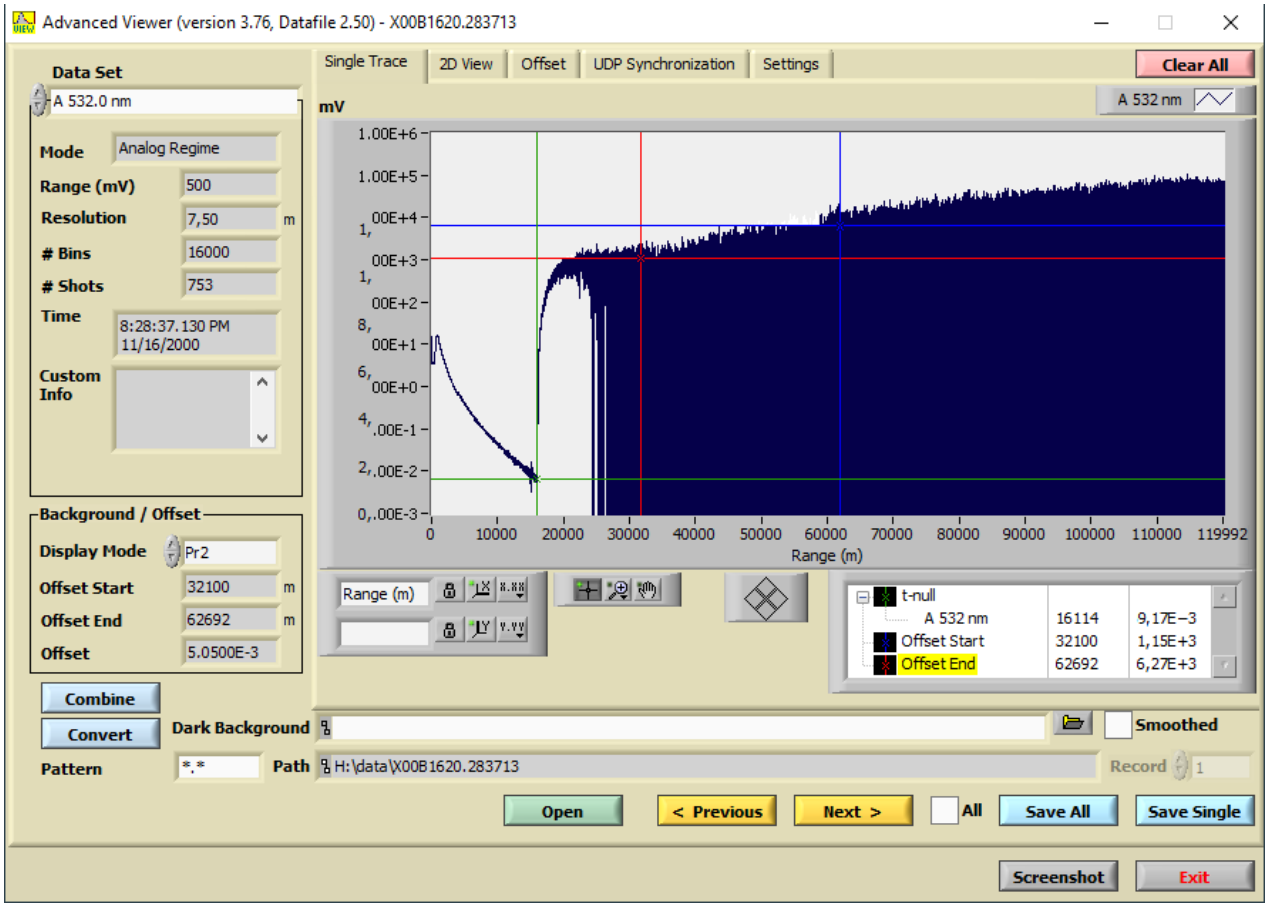
In the image above, you can see that the baseline is now close to zero. The end values of the region to be used to evaluate the baseline are shown in the *offset Start* and *offset End* indicators as well as in the cursor controls.



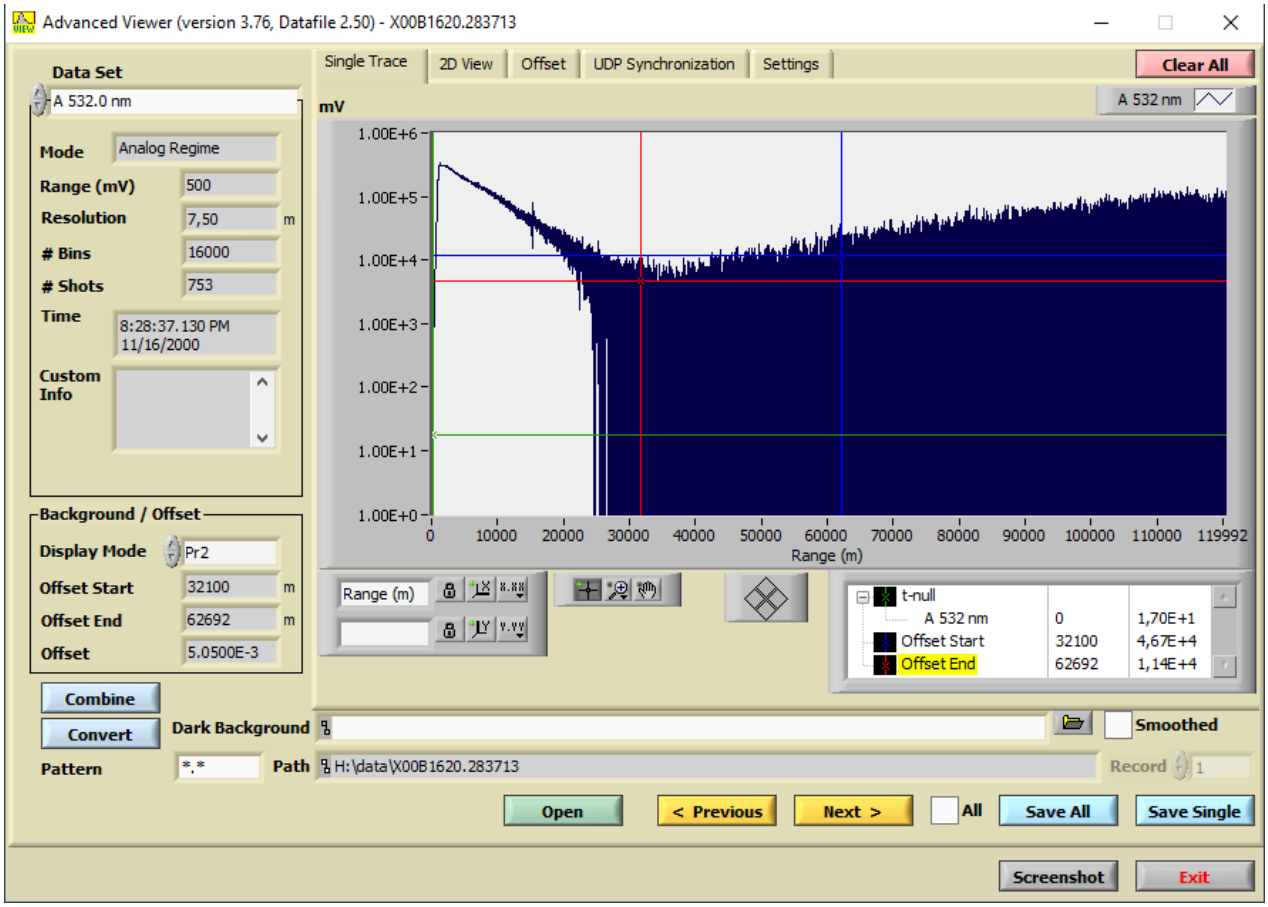
The offset, i.e. the mean value of the raw signal between *offset Start* and *offset End* is separately displayed:

.

The data can also be displayed in the *Pr2* mode which corrects for the power loss due to the length of travel of the signal. The key parameter for the *Pr2* mode is *t-null* which defines the starting point of the signal. When switching to the *Pr2* mode, the data will look similar to the following.

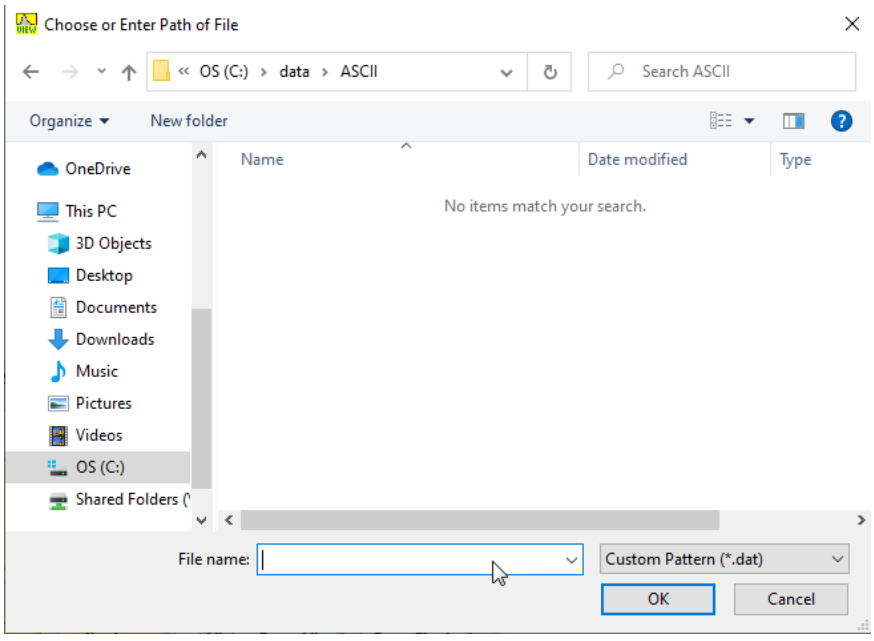


Note that the signal is only corrected for power loss after the green cursor, which is the t-null point. All values before *t-null* are left unchanged and those after *t-null* are corrected for the power loss due to distance. The difference in the display is due to the fact that the scaling has been changed to a logarithmic scale for easier viewing. After moving the cursor to the new t-null point, the signal changes to represent the new start time of the signal.



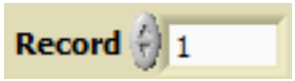
Please remember that the *Pr2* signal is offset corrected, as well.
Once you have adjusted your signal and would like to save it to a file in ASCII format as it is displayed, then press the *Save Single* button.

A file dialog appears asking you to name the ASCII file.



Enter the desired file name press *Save*. The file extension `.dat` will automatically be added to the file name unless you choose a different extension. Afterwards the data is saved in ASCII format as a column and can be imported into other programs for further evaluation.

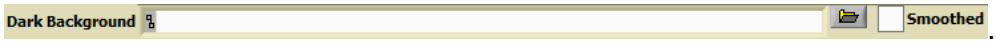
converts all datasets contained in the actual data file to an ASCII format file and appends the extension `.dat` to the end of the actual file's name. The whole file can then be imported into other programs.
A Licel data file may contain more than one record (a record is an acquisition with the specified target shot number). *TCPIP MPush Acquis* can write more than one acquisition into the same file. If a file contains more than one record the displayed record can be selected using the record selector at the right of the path.



Checking the checkbox *All* all these records of a file will be loaded when opening the file

.

A *Dark Background* file can be selected using the browse button at the right of the Dark Background control. If the selected file is valid and matching with the loaded data file it's data will be subtracted from the displayed plot. This raw subtraction will reduce clock noise but will add random noise. Optionally the selected background data can be smoothed by checking the checkbox *Smoothed* before the subtraction. Then the standard deviation is reduced but not the clock noise.



.

Use <*Previous* or *Next*> buttons if you would like to load the next file or previous file in a time series or the next or previous record in a file,

By pressing one of these buttons, either the file acquired before or after the current file (or, if existing, the previous or next record inside the same file) will be displayed if it exists in the same directory.
<*Previous* or *Next*> will use the specified file *Pattern*

One can then manipulate the data using the aforementioned capabilities of *Advanced Viewer* and save the data from the new datasets to ASCII if desired.
The control elements shown below are available if the *Advanced Viewer* is running in the LabVIEW development environment or if it has been started by *Acquis*. If the Windows or Linux application is running stand-alone, these controls are not present.

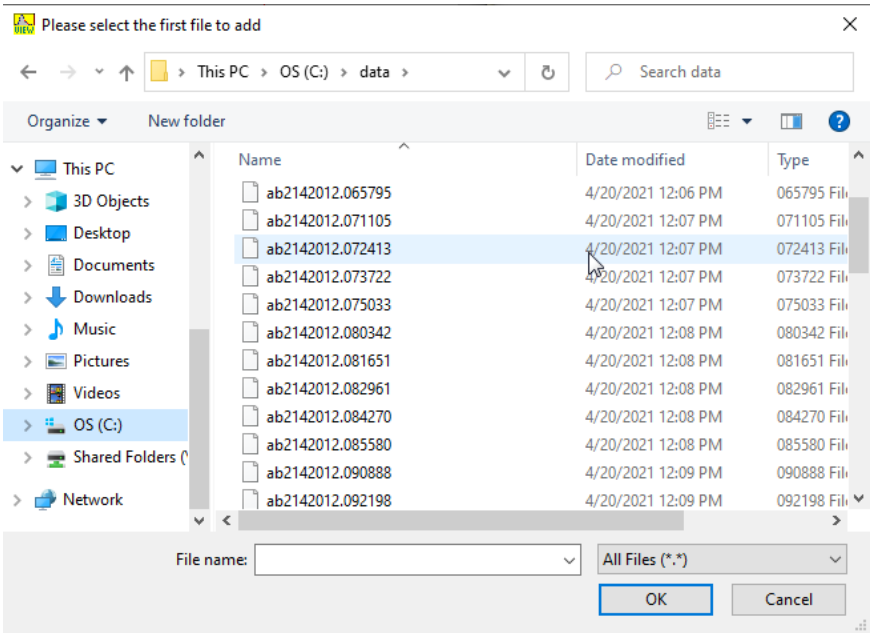
.

Check the box *Sync to Acquis* will make the *Advanced Viewer* to listen to the data file notifier. Whenever a new data file been written, it will be displayd in the *Advanced Viewer*. Then, it is possible to inspect a different data set than in *Acquis* or to view the 2D or the Offsets plots. A green LED light will indicate that the notifier served by *Acquis* can be read. Whenever an update is received the *Last Sync*ronization time is updated.
Two data file utilities may be called from the *Advanced Viewer*, one to sum the data values of several files to one single file (`Datafile Addfiles Interface.vi`), and another to convert the data from several files to corresponding ASCII files (`Datafile Batch Converter.vi`).
By pressing *Combine* `Datafile Addfiles Interface.vi` is interactively called to sum the data contained in a set of subsequently recorded data files.

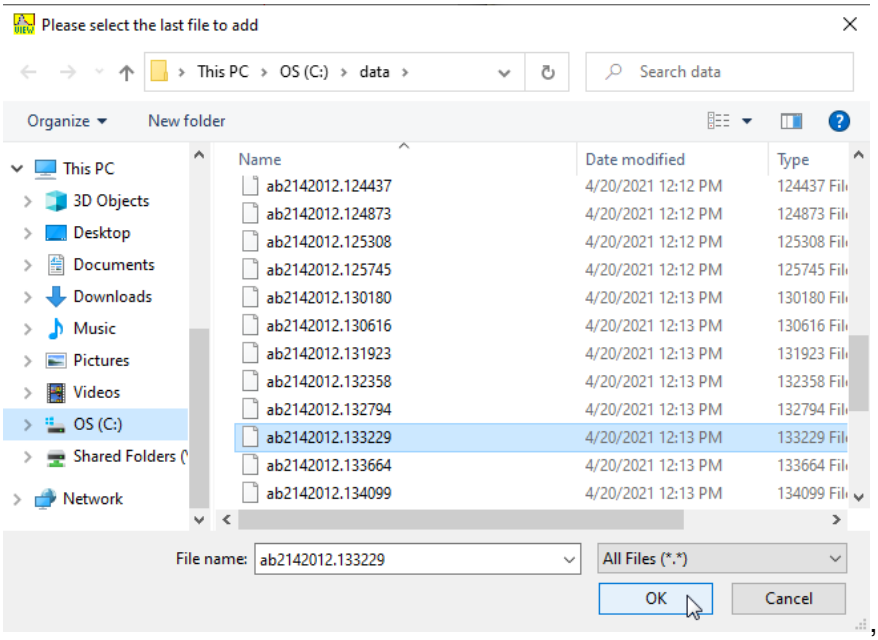
Combine

You will have to specify:

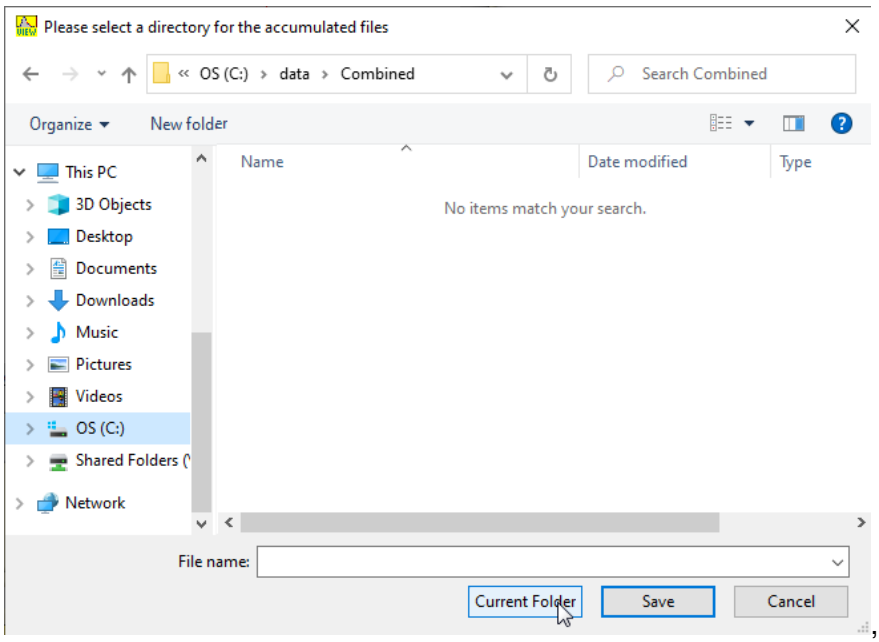
1. the name of the first file to add



2. the name of the last file to add

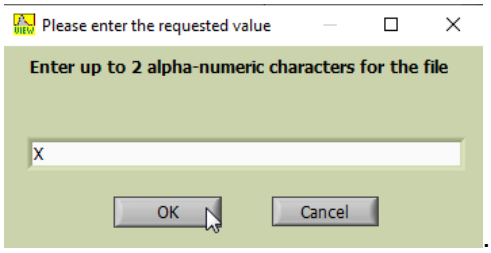


3. the name of the target directory for the file containing the summed data.



, and

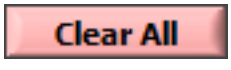
4. the first (two) letter(s) of the file name



.

Both the first and the last file must reside in the same directory. The data from the files with acquisition dates/times lying between the first and the last files (including them) are summed and written to a target file into the target directory. The target file's name begins with the first letter, and the rest of the name is taken from the first selected file.
By pressing *Convert* `Datafile Batch Converter.vi` is interactively called to convert the data contained in a set of subsequently recorded data files to ASCII files.

The selection mechanism to select the first file, the last file, and the target directory is the same as for the sum operation.
Both the first and the last file must reside in the same directory. The data from the files with acquisition dates/times lying between the first and the last files (including them) are converted as described above for the *Save All* operation. Each data file will result in an ASCII file named by the original file name with the additional extension `.txt`.
The *Clear All* button will empty all data from the plots.



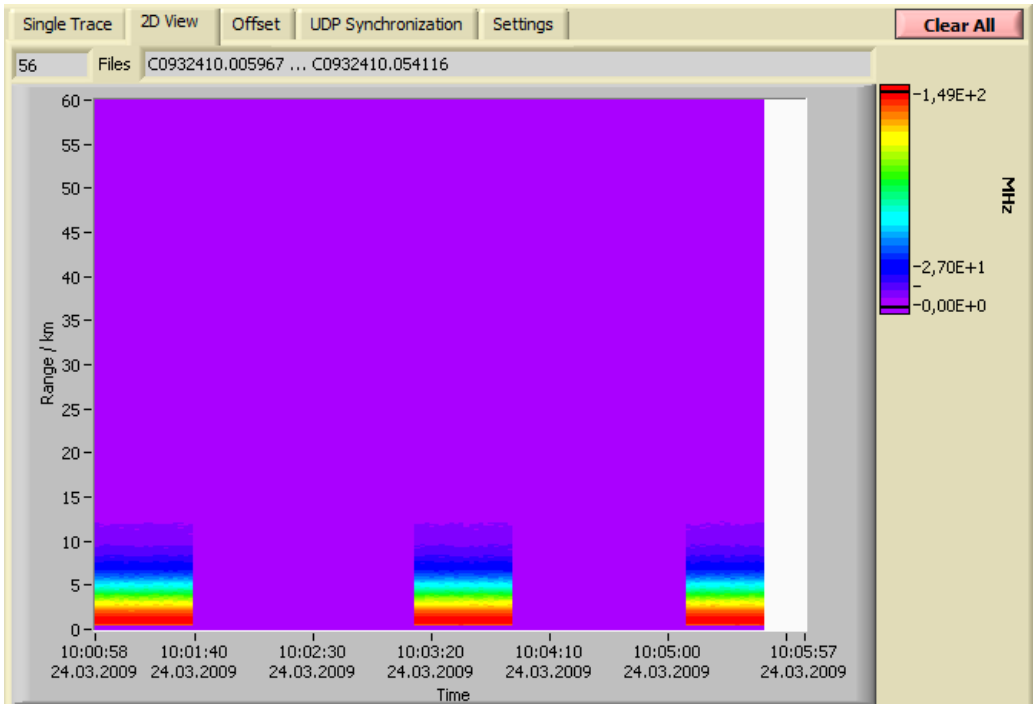
A screenshot of the current Viewer window can be saved to a portable network graphics (`png`) file by clicking the *Screenshot* button and selecting a file name in the subsequent file dialog.

The *Advanced Viewer* is terminated by pressing the *Exit* button or the Window close button ×, when running as a Windows or Linux application the front panel window will close.

2D Viewing of Multiple Data Files

The Advanced Viewer supports a 2D display for several data files. To switch to the 2D View please use the *2D View* page of the tab page selector.

In the *2D View* the signal is displayed in a color map as a function of time and range.



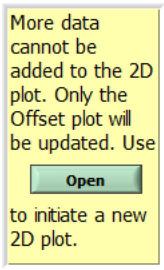
Data is added to the 2D plot with the following buttons:

clears the 2D plot and adds the selected *Data Set* of the new loaded file according to the *Display Mode* setting.

adds the selected *Data Set* from the next file in the directory to the 2D plot and displays it according to the *Display Mode* setting.

adds the selected *Data Set* from the previous file in the directory to the 2D plot and displays it according to the *Display Mode* setting. If the data is not compatible to the previously loaded files the 2D plot will be cleared before displaying the new file's data.

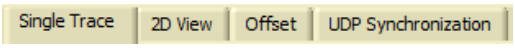
The number of loaded *Files* and the file name range is shown on the top of the *2D View* page. The data from the last file loaded by these buttons is always shown in the *Single Trace* plot according to the *Data Set* selection and *Display Mode* setting. If a file is already plotted in the 2D plot no data will be added. The `temp.dat` file created by the Acquis program will not be displayed in the 2D plot.
To avoid memory problems with the 2D plot a warning message will be shown when a limit is reached:



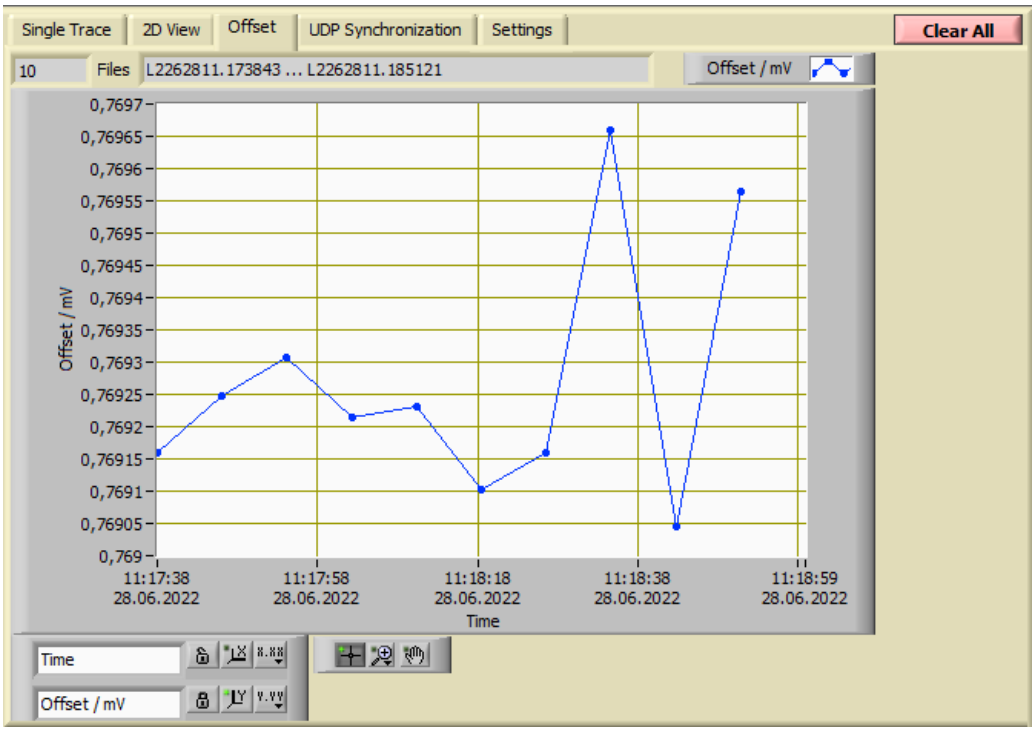
Data from further data files will not be added, but the *Offsets* plot will be updated. Use the *Open* button to start with a new 2D plot.

Offset Timeline of Multiple Data Files

The Advanced Viewer supports a timeline view of the signal offset between the offset cursors. To switch to the offset timeline use the *Offset* page of the tab page selector.



The offset time line allows to inspect whether or not the offset is constant or shows a certain trend.



Data is added to the plot in the same way as it is added to the 2D plot with the following buttons:

clears the offset timeline and adds the offset of the selected *Data Set* of the new loaded file according to the *Display Mode* setting.

adds the the offset of the selected *Data Set* from the next file in the directory to the timeline and displays it according to the *Display Mode* setting. If the data is not compatible to the previously loaded files the offset timeline will be cleared before displaying the new file's data.

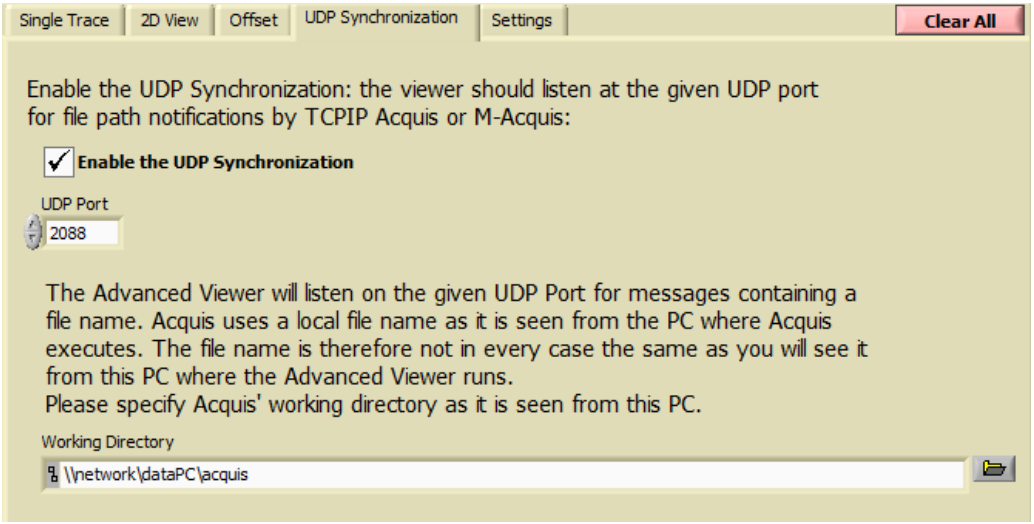
adds offset of the selected *Data Set* from the previous file in the directory to the timeline and displays it according to the *Display Mode* setting. If the data is not compatible to the previously loaded files the 2D plot will be cleared before displaying the new file's data.

The time axes of the 2D plot and the offset timeline are equal.

UDP Synchronization

While the *Advanced Viewer* can be synchronized with *TCPIP Acquis* using a LabVIEW Notifier when both software components are running on the same PC, there is a more general method for file synchronization. *UDP Synchronization* will work even if the *Advanced Viewer* runs on a different PC than the acquisition software. To achieve UDP synchronization follow the steps below:

1. Configure *Acquis* to publish it's file path via UDP. Choose either *Broadcast* or set an IP address equal to the IP address of the PV where the *Advanced Viewer* is running. Make sure that the selected port must not be blocked by firewalls in the network.
2. In the Advanced Viewer go to the tab page *UDP Synchronization*.



3. Enable the UDP synchronization by checking the corresponding checkbox.
4. Enter the UDP Port. This should be the same port as configured in the UDP configuration of the acquisition software.
5. Now browse for the *Working Directory* where *Acquis* writes it's files to. This step is important, because locations in a network may look different from different computers.
6. Start a multiple acquisition at *Acquis*. From then the *Advanced Viewer* will automatically update it's display right after a new file has been written by the acquisition software.
7. The Synchronization information (*Last Sync*hronization Time, *TR Acquisition* LED) will show up without a checkbox:

Settings

Another Option is available on the tab page *Settings*:

- The *Range Scale Unit* can be switched between *m/km* (meters in the single plot, kilometers in the 2D plot) and *µs*.

5.6 Further Data Analysis

The analysis of the acquired data depends strongly on the individual application of Licel detection systems. Therefore, we do not provide a complete data analysis package.
In an appendix we provide a strategy and an example to combine analog and photon counting data acquired with Licel transient recorders. Then a dynamic range of more than 5 orders of magnitude can be achieved.
The data file format is described in an appendix.

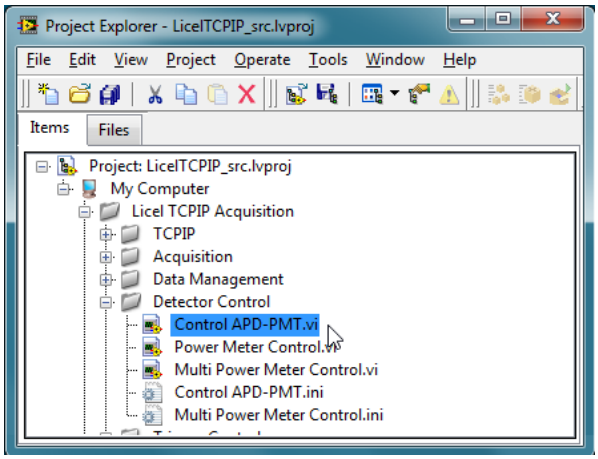
Chapter 6

Detector and Timing Control Utilities

6.1 The Combined APD and PMT Control Panel

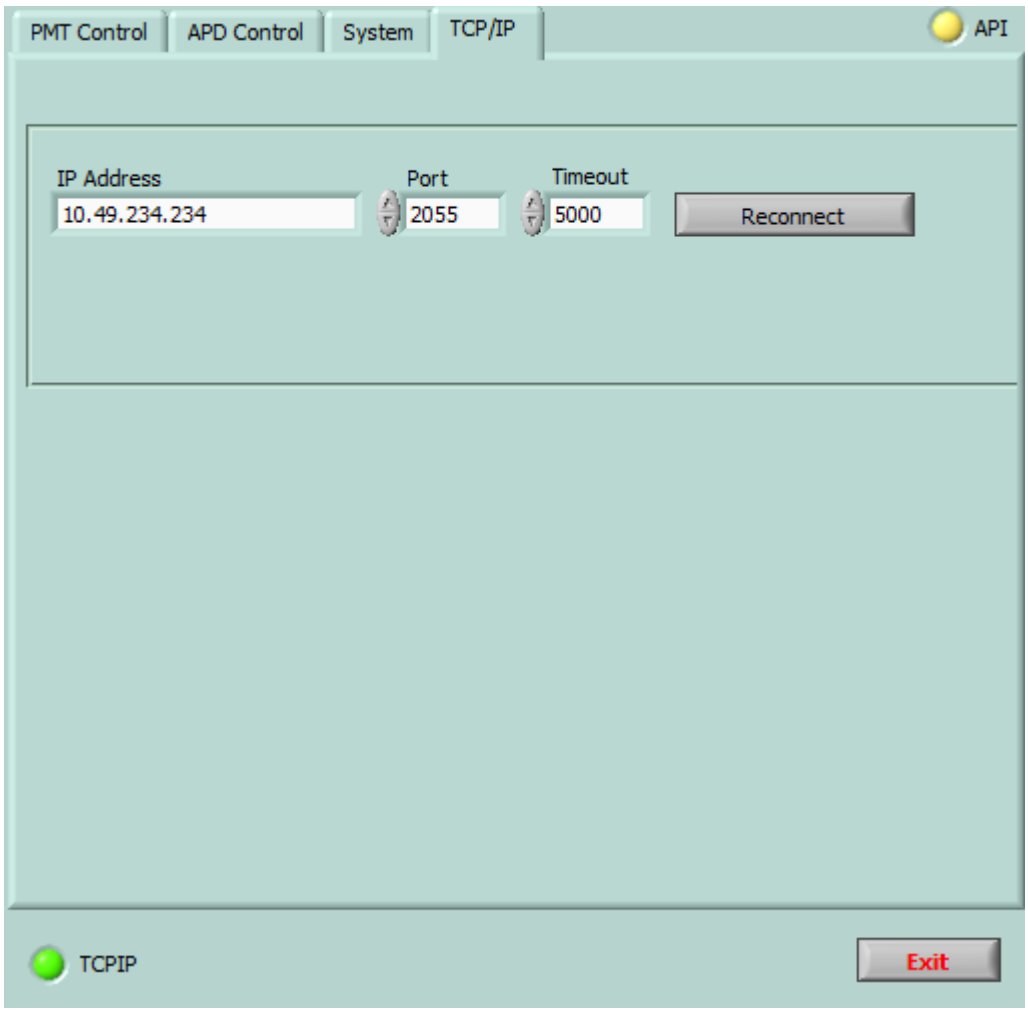
In this section the application *Control APD-PMT* is described which is capable to control APDs and/or PMTs dependent on the remote controller's capability support.
For basic LabVIEW examples refer to the example VIs `Easy Control APD.vi` and `Easy Control PMT.vi` and extract the code you need for your own application.

- If you are using the LabVIEW sources open Control APD-PMT from the LabVIEW project by navigating to the corresponding entry *Control APD-PMT.vi* and double-clicking it.



- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

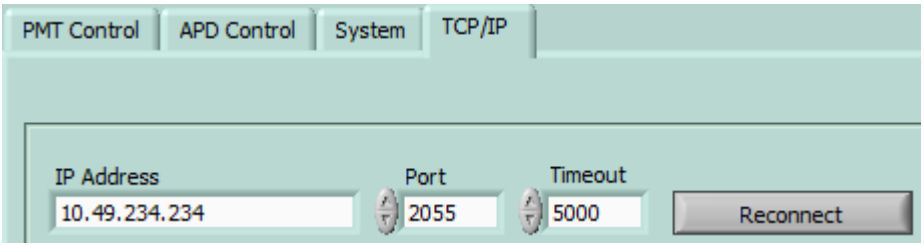
After opening it you should see a screen like the following:



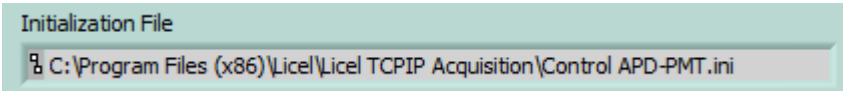
6.1.1 Starting the Application

1. First of all the **IP Address** and **Port** have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

   - Using the LabVIEW vi, just enter the required values on the *System* page and save them as defaults.



   - If you use the Windows application you must directly enter the correct values into the corresponding control fields on the *TCP/IP* page. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.
   You may also set the values in the initialization file `Control APD-PMT.ini`. You will see the full path of the file in a file path indicator on the *TCP/IP* page.

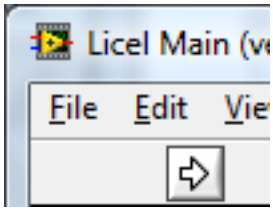


- Please note that the inizialization file name is `Control APD-PMT`n`.ini` when you are running `Control APD-PMT` as a software module in `Licel Main`. There, n corresponds to the n^{th} instance of *Control APD-PMT* you are operating.
- If you run `APD-PMT Control` within a sub panel on a page from `Licel Main` the latter is responsible for managing the TCP/IP connection.
- Enter the number of available PMTs and APDs to the initialization file `Control APD-PMT.ini` (or `Control APD-PMT`n`.ini`) in the same directory where the LabVIEW llb or the executable program resides:

```
...
[Number_Of_PMTs]
Number_Of_PMTs=5

[Number_Of_APDs]
Number_Of_APDs=1
...
```

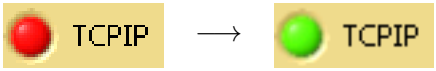
  In recent program versions the number of APDs and PMTs may be changed just by entering the corresponding numbers on the page *System*.

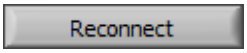
2. To start the program press the **Run** button at the top left of the screen.



   The Windows application will start automatically when called for the first time.

3. After a short time the **TCPIP** indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for **address** and **Port**, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct.

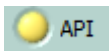


4. If you recognize that you are not connected to the Ethernet Controller you would like to use, just
   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and
   (b) press the 

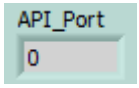
button.

   Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

5. Since version 1.70.01 *Control APD-PMT* may be controlled externally via a TCP/IP API. If such a connection is active, the indicator at the top right of the front panel is displayed:



   The used Port is shown on the *TCP/IP* tab page if *Control APD-PMT* is running stand-alone (not as a sub module in *Licel Main*):

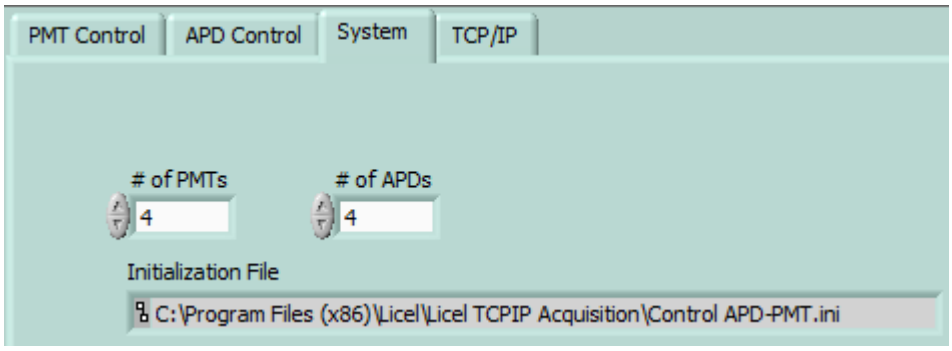


6. Use the **Exit** switch or the Window close button ✕ to stop the program, when running as a Windows or Linux application the front panel window will close.

6.1.2 Operation

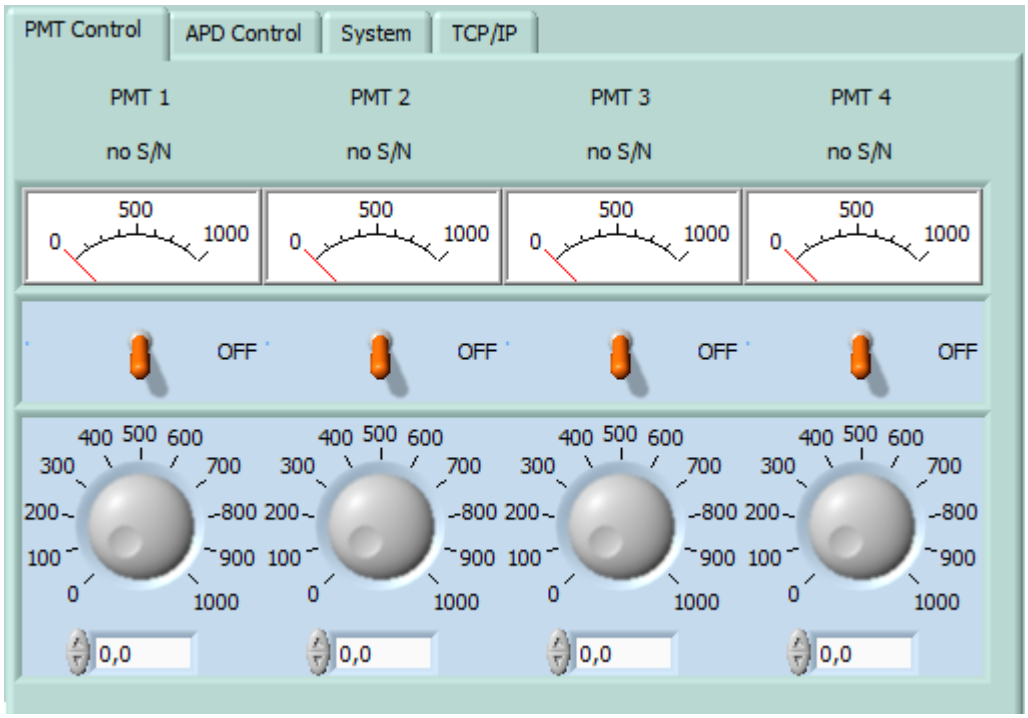
Dependent on the Licel Ethernet Controller capabilities only those tab pages will be shown with the corresponding functionality. If a controller is capable to control both, PMTs and APDs, the pages *PMT Control* and *APD Control* will be present.

System Information



On the page *System* you may inspect and change (since version 1.70.01) the number of PMTs and APDs and see the full path of the initialization file. On change of the number of APDs or PMTs the new values are written to the file, and the front panel with it's elements to control the APDs and PMTs will be be resized.

PMT Control

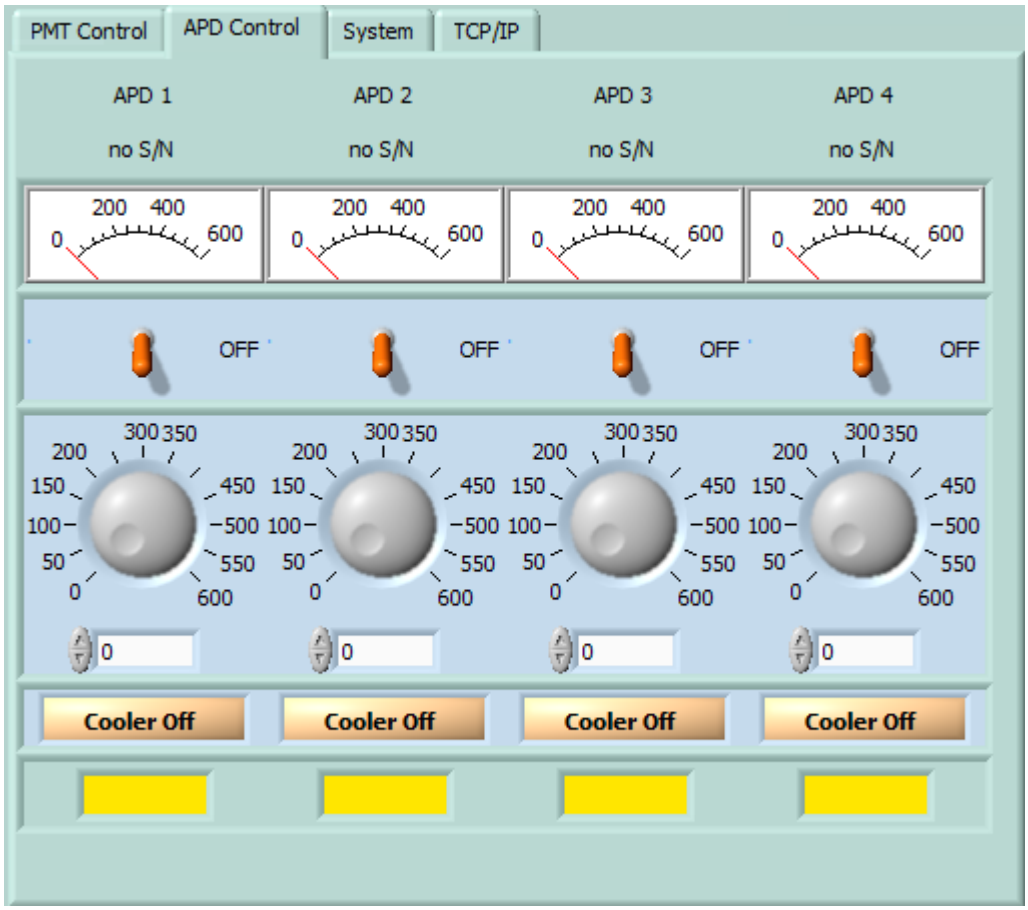


On the page *PMT Control* the voltage of the PMTs can be set via the knobs at the bottom. The displays at the top show the actual voltages. Turning the switches on, will set the desired voltages at the PMT.
Above the voltage displays you will find individual label (description) and serial number fields for each PMT. The content must be defined in the intialization file:

```
...
[PMT<index>]
Description = "my description"
Serial = "my serial"
...
```

If the initialization file keys for the description or serial are empty, `PMT %d` (where `%d` is the PMT index + 1) and `no S/N` are displayed, respectively.
Recent Quad-HV remote controllers support *PMT label (description)* handling using the `PMTDESCR` and `PMTDESCR?` commands. The PMT label will be visible at the display on the controller's front. As this label should be the same as in the *Control APD-PMT* software, these labels are read from the controller and will be used instead of labels found in the initialization file. When a Quad-HV controller is in use the labels can be changed on the front panel to modify the display at the controller.

APD Control



On the page *APD Control* The voltages can be set via the knobs at the bottom. The displays at the top show the actual voltages. Turning the switches on, will set the desired voltages at the APD. After switching from the passive to the active cooling mode the cooling status indicator will first turn red indicating that the APD temperature is not in range later will become green when the temperature is stabilized.
Above the voltage displays you will find individual description and serial number fields for each APD. The content must be defined in the intialization file:

```
...
[APD<index>]
Description = "my description"
Serial = "my serial"
...
```

If the initialization file keys for the description or serial are empty, `APD %d` (where `%d` is the APD index + 1) and `no S/N` are displayed, respectively.

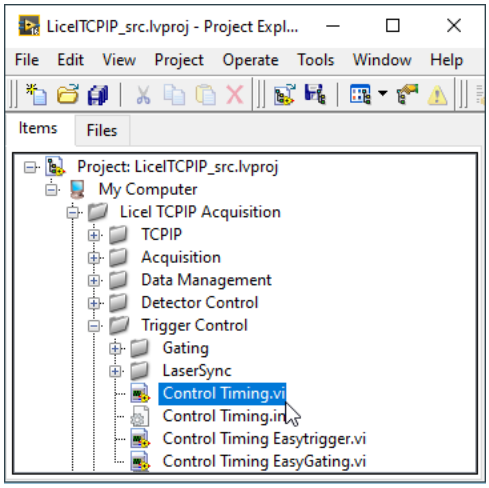
6.1.3 Assign Transient Recorder Channels to a detector

The transient recorder data acquired by *TCPIP Acquis* or *M-Acquis* is saved to data files containing the current detector (APD or PMT) high voltage as header information.
For this an initialization file-based mechanism is available to link transient recorder channels with detectors served by any instance of *Control APD-PMT* and to update the transient recorder data with the current voltages. Details are explained in the Appendix.

6.2 The Trigger Module Control Panel

In this section the application *Control Timing* is described which is capable to control Licel's trigger controller.
For basic LabVIEW examples refer to the example VIs `Control Timing Easytrigger.vi` and `Control Timing EasyGating.vi` and extract the code you need for your own application.

- If you are using the LabVIEW sources open *Control Timing* from the LabVIEW project by navigating to the corresponding entry *Control Timing.vi* and double-clicking it.



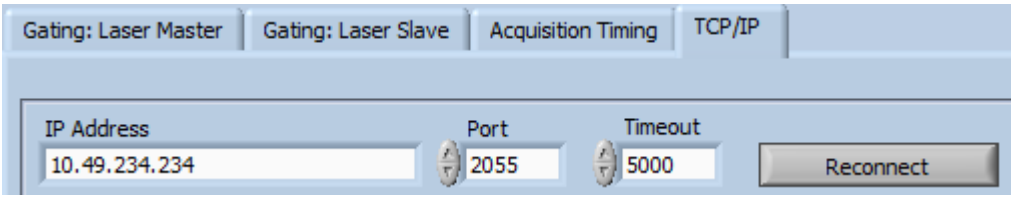
- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

In the case that the Licel Trigger Module is equipped with more than one timing sub-board the application `Gating Control.vi` (`Gating Control.exe`) must be used.

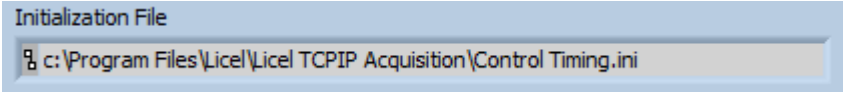
6.2.1 Starting the Application

1. First of all the **IP Address** and **Port** have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

   - Using the LabVIEW vi, just enter the required values on the *TCP/IP* page and save them as defaults.

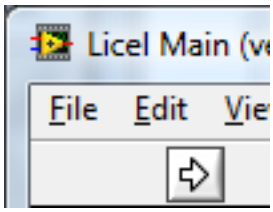


   - If you use the Windows application you must directly enter the correct values into the corresponding control fields on the *TCP/IP* page. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.
   You may also set the values in the initialization file `Control Timing.ini`. You will see the full path of the file in a file path indicator on the *TCP/IP* page.



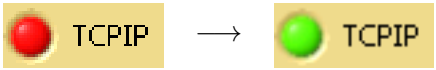
   - If you run `Control Timing` within a sub panel on a page from `Licel Main` or `Gating Control` the latter is responsible for managing the TCP/IP connection.

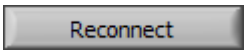
2. To start the program press the **Run** button at the top left of the screen.



The Windows application will start automatically when called for the first time.

3. After a short time the **TCPIP** indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for **address** and **Port**, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct.

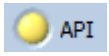


4. If you recognize that you are not connected to the Ethernet Controller you would like to use, just
   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and
   (b) press the 

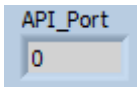
button.

   Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

5. Since version 1.70.01 *Control Timing* may be controlled externally via a TCP/IP API. If such a connection is active, the indicator at the top right of the front panel is displayed:



   The used Port is shown on the *TCP/IP* tab page if *Control APD-PMT* is running stand-alone (not as a sub module in *Licel Main*):

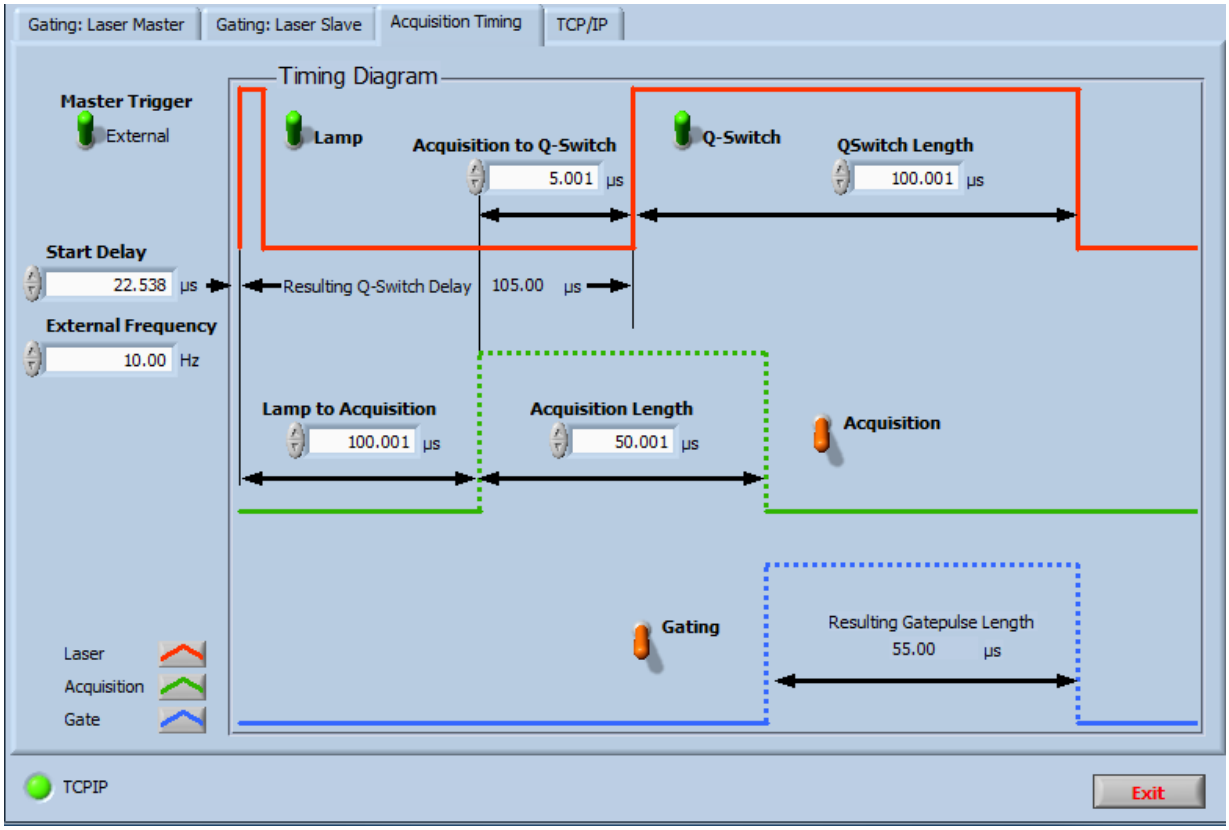


6. Use the **Exit** switch or the Window close button ✕ to stop the program, when running as a Windows or Linux application the front panel window will close.

Note that in each of the following scenarios the pretrigger pulses are only generated if the corresponding switches are in the *On* position.

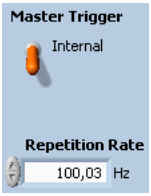
6.2.2 Direct Control of the Timing Parameters

In this scenario the user has to directly set the same timing parameters as the Licel Trigger Module is using. For a basic LabVIEW example refer to the example VI `Control Timing Easytrigger.vi` and extract the code you need for your own application.



The following parameters may be changed or set:

- **Master Trigger**: Determines whether the triggers are internally (*Internal*) generated or an external trigger is supplied (*External*). On timing sub-boards of a Licel Trigger Module where the external trigger is supplied by another timing sub-board through a fixed electronic connection the **Master Trigger** must be set to *External* because from the sub-board's point-of view the trigger is supplied from outside. Licel refers to such a sub-board as a *Slave-Only-Board*. For newer Licel Trigger Modules (shipped after April 2020) the **Master Trigger** is automatically set for *Slave-Only-Board*s and the **Master Trigger** switch is hidden. Please look at the example below.
  Dependent on the **Master Trigger** setting different parameters are available:
  - **Master Trigger** = *External*
    - **Start Delay**: Initial delay time at the start of the time chain. The **Start Delay** is available in the *External* mode. For trigger controllers shipped before June 2007 a firmware update is required to use the **Start Delay**. Otherwise the delay time set is ignored (start delay 0 µs).
    - **External Frequency**: Estimated frequency of the external trigger source in Hz. This value will help newer controllers to avoid unexpected irregularities by suppressing changes of the timing parameters while a laser trigger cycle is processed.
  - **Master Trigger** = *Internal*
    - **Repetition Rate**: Frequency in Hz of the internally generated trigger pulses, i.e. of the laser **Lamp**, **Acquisition**, **Q-Switch**, and **Gating** pulses. The **Repetition Rate** is available in the *Internal* mode.

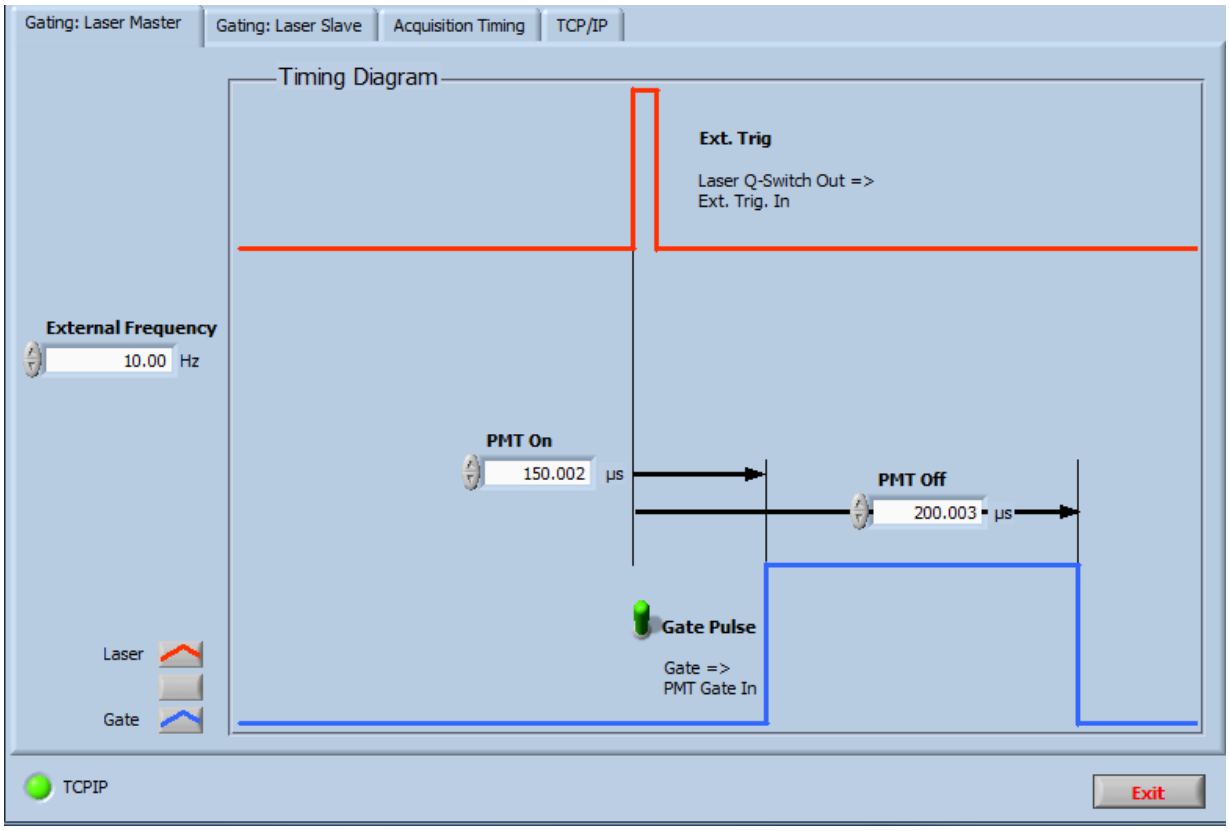


- **Lamp to Acquisition**: Delay time between the laser lamp pulse and the acquisition pretrigger pulse (µs)
- **Acquisition Length**: Length of the acquisition pretrigger pulse (µs)
- **Acquisition to Q-Switch**: Delay time between the acquisition pretrigger pulse and the Q-switch trigger pulse (µs)
- **Q-Switch length**: Length of the Q-switch trigger pulse (µs).

Lamp, **Acquisition**, **Q-Switch**, and **Gating** have to be set *On* to generate the corresponding trigger pulses.

6.2.3 Gating: Laser Master

Use this scenario if the laser's Q-switch output is used as the trigger source. Then the gate pulse is characterized by its start and end times with respect to the external trigger. This option is not available for *Slave-Only-Board*s shipped after April 2020. For a basic LabVIEW example refer to the example VI `Control Timing Easygating.vi` and extract the code you need for your own application.



The following parameters may be set (µs):

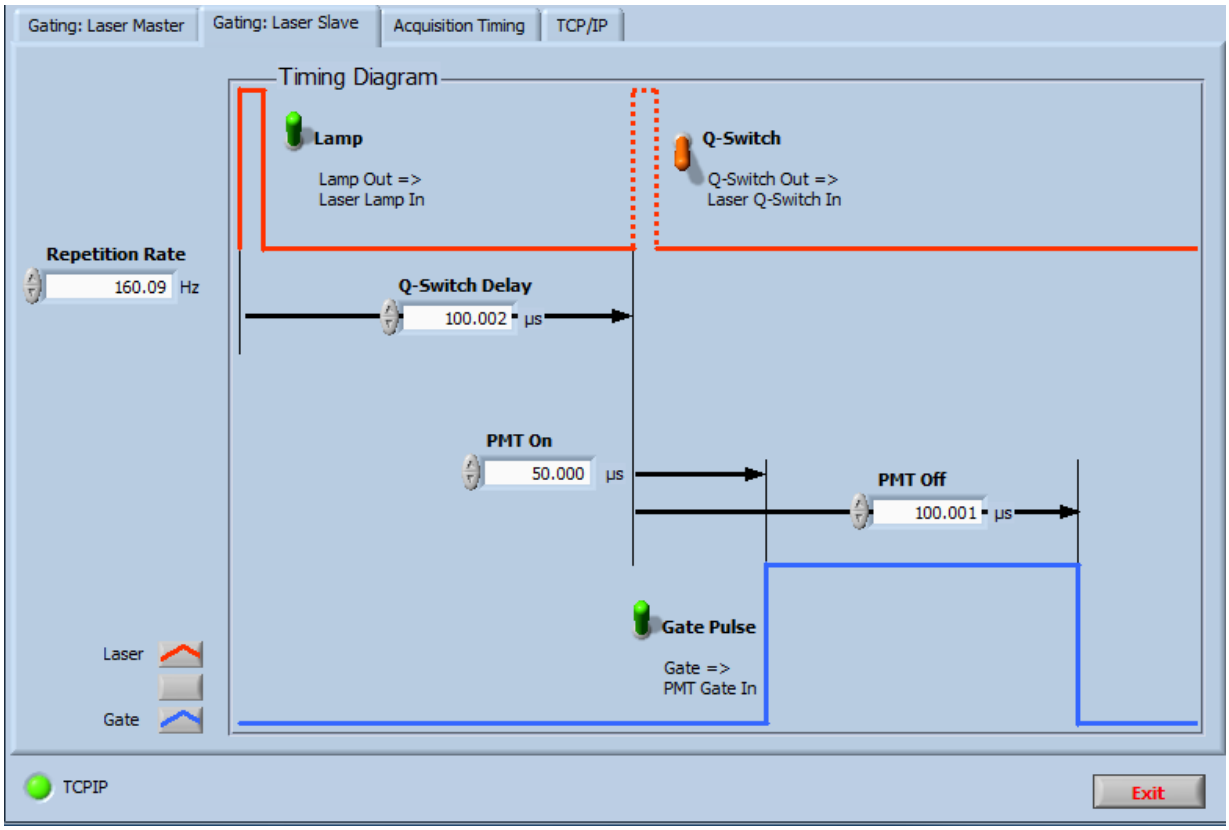
- **External Frequency**: Estimated frequency of the external trigger source in Hz. This value will help newer controllers to avoid unexpected irregularities when changing the timings.
- **PMT On**: Gate pulse start time with respect to the external trigger (laser Q-switch out)

- **PMT Off**: Gate pulse stop time with respect to the external trigger (laser Q-switch out).

Gate Pulse needs to be set *On* to enable the gate pulse output.

6.2.4 Gating: Laser Slave

Use this scenario to trigger the laser lamp and the Q-switch from the Licel Trigger Module. Then the gate pulse is characterized by its start and end times with respect to the Q-switch pulse. This option is not available for *Slave-Only-Board*s shipped after April 2020.



The following parameters may be changed or set:

- **Repetition Rate**: Frequency in Hz of the internally generated trigger pulses, i.e. of the laser **Lamp**, **Q-Switch**, and **Gating** pulses.
- **Q-Switch Delay**: Time between the lamp trigger output and the Q-Switch output (µs)
- **Gate On**: Gate pulse start time with respect to the Q-switch out (µs)
- **Gate Off**: Gate pulse stop time with respect to the Q-switch out (µs).

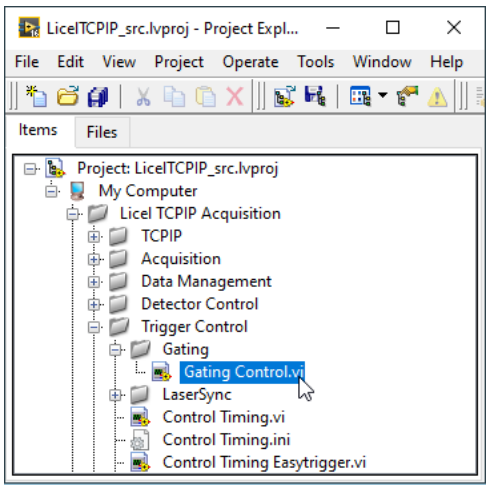
Lamp, **Q-Switch**, and **Gate Pulse** have to be set *On* to generate the corresponding trigger pulses.

6.3 The Gating Control Module (Multiple Timing Sub-boards)

This application must be used in the case that a Licel Trigger Module is equipped with more than one timing sub-board. The user interface of *Gating Control* allows to control each timing sub-boards in seperate tab page. Each tab page contains the necessary switches and timing parameters for a single sub-board.

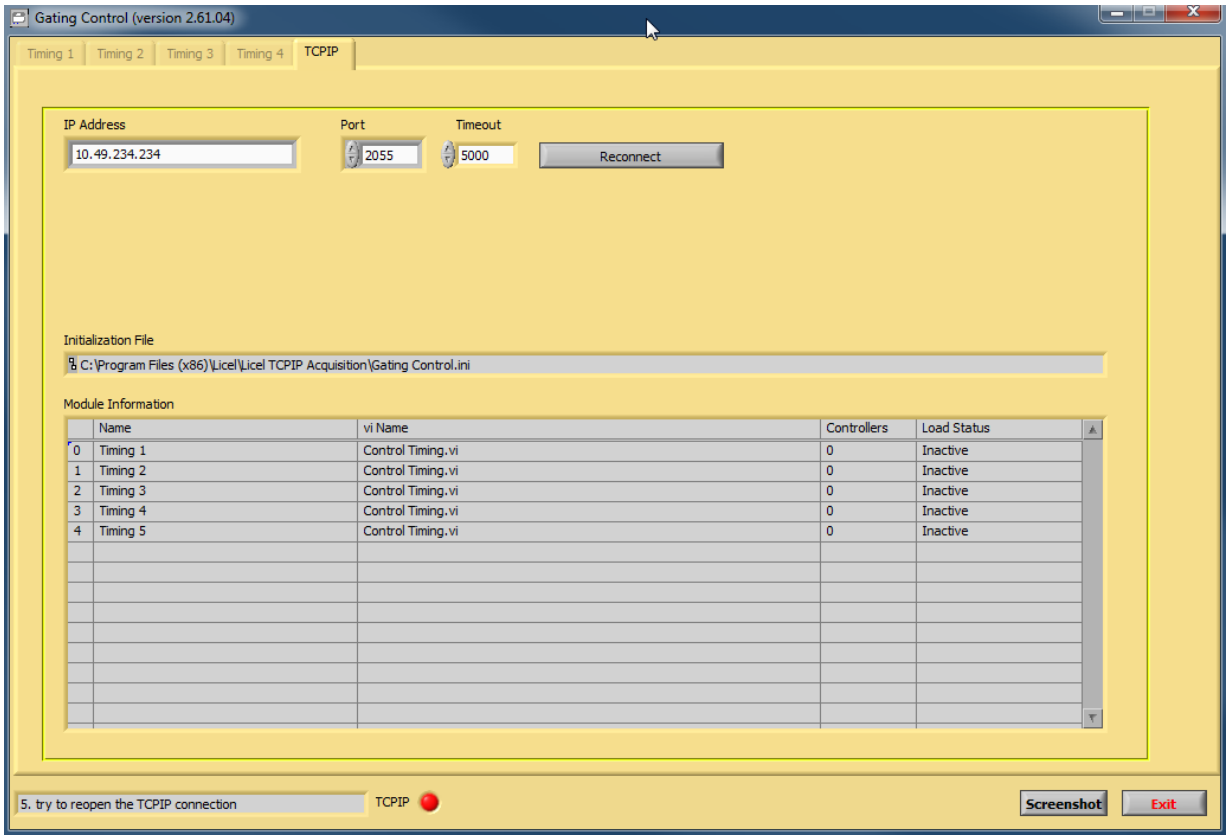
6.3.1 Starting the Application

- If you are using the LabVIEW sources open the gating control application from the LabVIEW project by navigating to the corresponding entry *Gating Control.vi* and double-clicking it.



- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

The front panel of the Gating Control application is seen in the next picture:



1. First of all the *IP Address* and *Port* have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

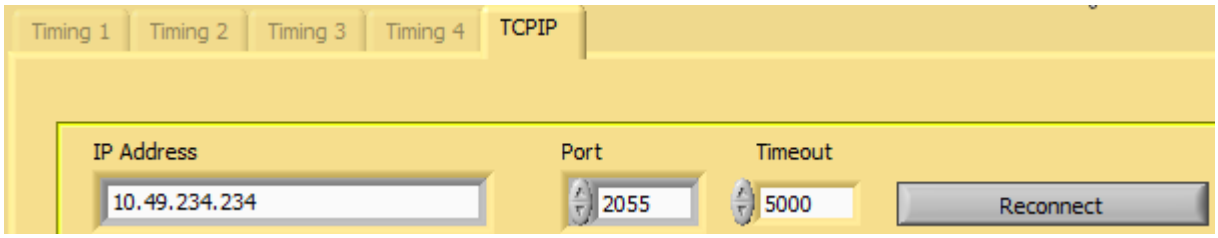
2. It is recommended to enter the correct values to the initialization file `Gating Control.ini`:

```
[TCPIP]
```

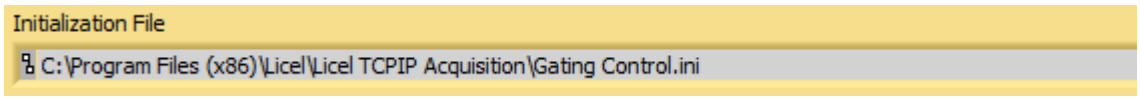
```
UseValues=TRUE
IPAddress=10.49.234.234
Port=2055
```

Both, the LabVIEW VI and the Windows and Linux application will read these values while starting.

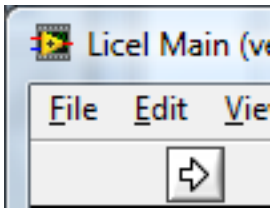
3. You may also enter the correct values into the corresponding control fields on the *TCP/IP* page *when the program is running*. *Gating Control* will using these values when opening a TCP/IP connection.



4. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.

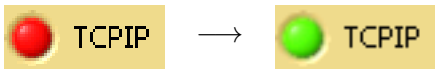


5. To start the program press the *Run* button at the top left of the screen.



The Windows application will start automatically when called for the first time.

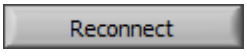
6. After a short time the *TCPIP* indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for *IP Address* and *Port*, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct.



7. Once the TCP/IP connection has been established *Gating Control* will check for `TIMER` capabilities at the connected ethernet control. For each `TIMER` sub-board it will load a separate instance of `Control Timing.vi`, each in a separate tab page. As *Licel Main* `Gating Control` uses an initialization file which can be found in the same directory as the *LabVIEW vi / Windows application*, it's name is `Gating Control.ini`.

8. If you recognize that you are not connected to the Ethernet Controller you would like to use, just

   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and

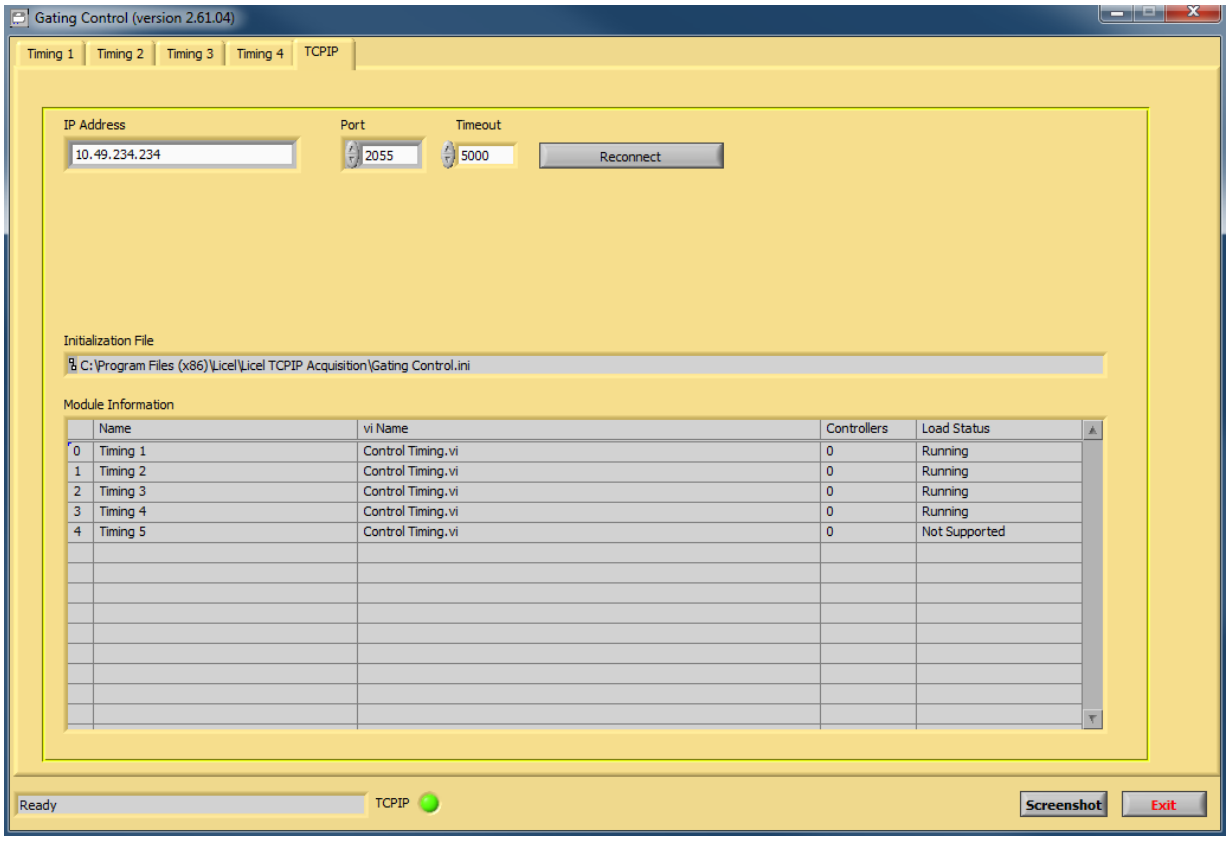
   (b) press the 

button.

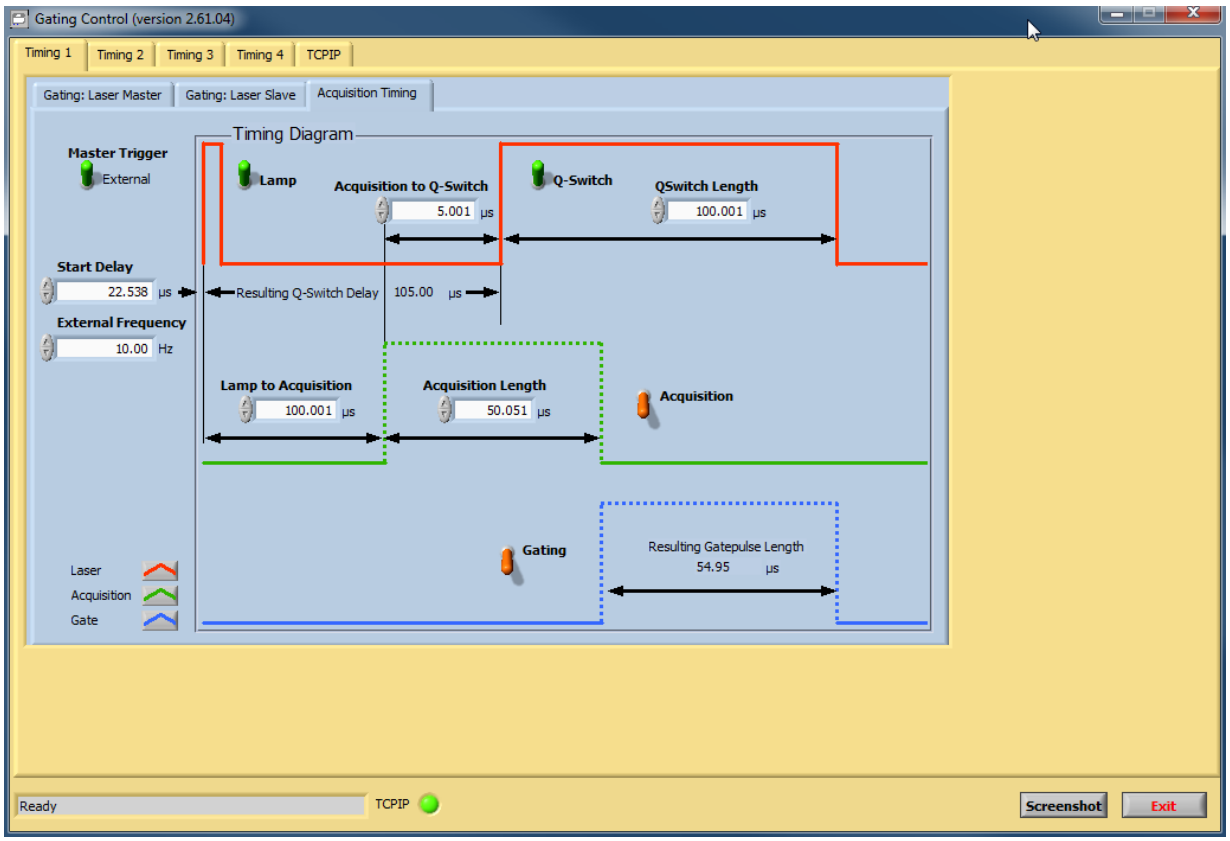
Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

6.3.2 Gating Control Operation

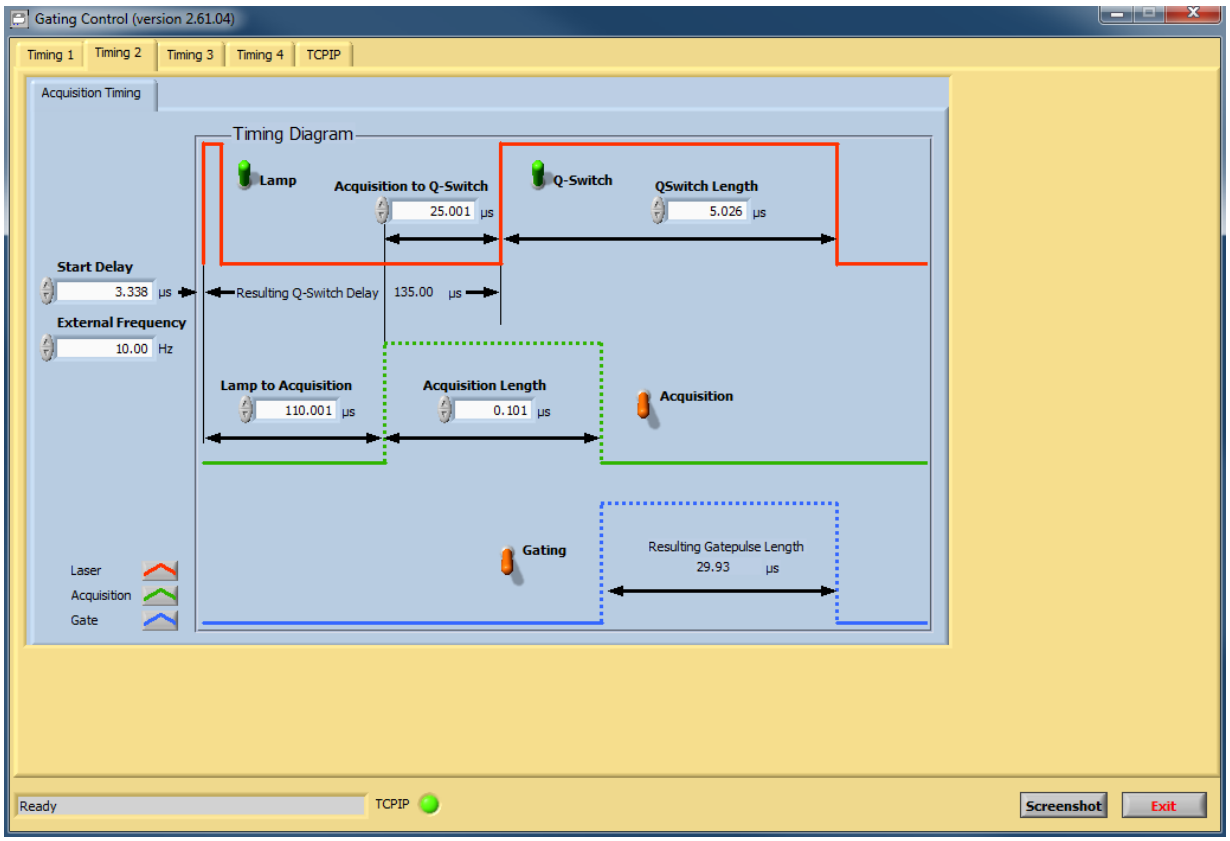
In the next picture *Gating Control* is connected with a Licel Trigger Module equipped with four sub-boards with the capabilities TIMER, TIMER1 ... TIMER3. With the shipped initialization file *Gating Control* will try to find a sub-board matching the TIMER4 capabilities, as well. In this example this fifth sub-board is not present:



The tab pages *Timing 1*, *Timing 2*, *Timing 3*, and *Timing 4* are active. On each of these tab pages the user interface of the *Control Timing* is embedded, the operation is controlled as described above.



In our example the second sub-board (related to the capabilty TIMER1) is a *Slave-Only-Board* in a Licel Trigger Module shipped after April 2020: In a *Slave-Only-Board* the master trigger is wired inside the Licel Trigger Module, i.e. the *Master Trigger* of the *Control Timing* user interface must be set to *External*. Here, the Licel Trigger Module is newer than April 2020, therefore the *Control Timing* software can ask the controller whether or not a sub-board is a *Slave-Only-Board*; as a consequence the correct *Master Trigger* is automatically set.



6.3.3 Gating Control Initialization File

The initialization file `Gating Control.ini` is located in the same directory as `Gating Control.vi` / `Gating Control.exe`. The keys labelled "`Name`" define the captions of the tab pages.

```
[Module0]
Active = TRUE
Path = "Control Timing.vi"
Name = "Timing 1"
Controllers = 0
CAPs = TIMER
Push = FALSE

[Module1]
Active = TRUE
Path = "Control Timing.vi"
Name = "Timing 2"
Controllers = 0
CAPs = TIMER1
Push = FALSE

[Module2]
Active = TRUE
Path = "Control Timing.vi"
Name = "Timing 3"
Controllers = 0
CAPs = TIMER2
```

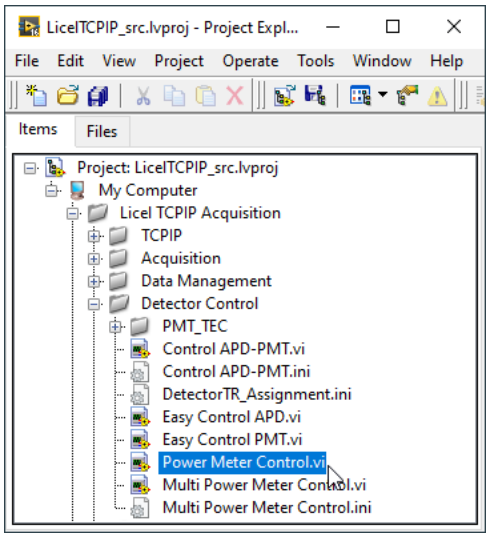
```
Push = FALSE

[Module3]
Active = TRUE
Path = "Control Timing.vi"
Name = "Timing 4"
Controllers = 0
CAPs = TIMER3
Push = FALSE

[Module4]
Active = TRUE
Path = "Control Timing.vi"
Name = "Timing 5"
Controllers = 0
CAPs = TIMER4
Push = FALSE
```

As described above, the TCP/IP parameters are included, as well.

```
[TCPIP]
UseValues = TRUE
Port = 2055
IPAddress = 10.49.234.234
```

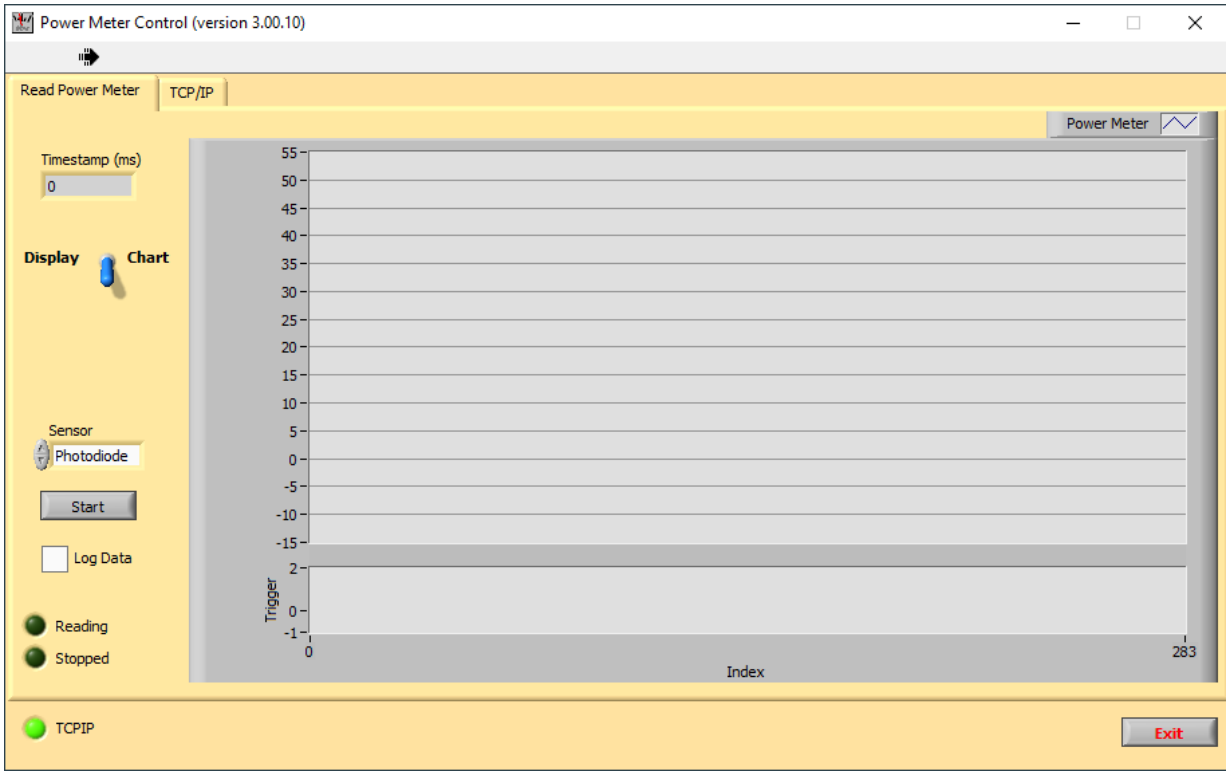
6.4 Power Meter Control

- If you are using the LabVIEW sources open the power meter control program from the LabVIEW project by navigating to the corresponding entry *Power Meter Control.vi* and double-clicking it.



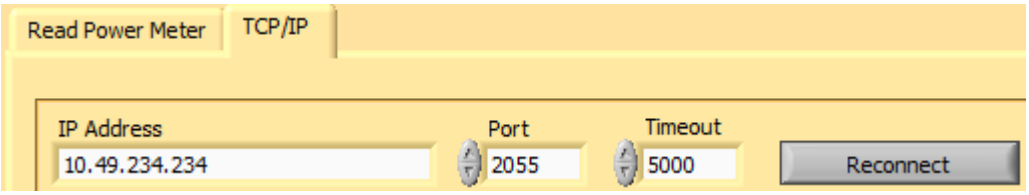
- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

The Power Meter Control software is capable to control one Power Meter controller. Licel's most recent Power Meter controllers may be equipped with up to three detector inputs. The front panel of the Power Meter Control software is seen in the next picture:

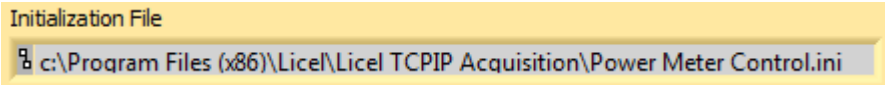


1. First of all the **IP Address** and **Port** have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

   - Using the LabVIEW vi, just enter the required values on the *TCP/IP* page and save them as defaults.

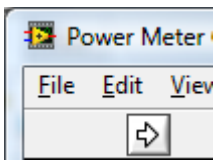


   - If you use the Windows application you must directly enter the correct values into the corresponding control fields on the *TCP/IP* page. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.
   You may also set the values in the initialization file `Power Meter Control.ini`. You will see the full path of the file in a file path indicator on the *TCP/IP* page.



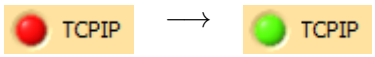
   Please note that when you run *Power Meter Control* from *TCPIP Acquis/M-Acquis* that the initialization file name(s) of the used instances of *Power Meter Control* will be `Power Meter Control<n>.ini` where *n* is a number starting at 1. Just watch the initialization file indicator to be sure to use the correct file.

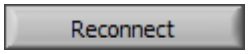
2. To start the program press the **Run** button at the top left of the screen.



The Windows application will start automatically when called for the first time.

3. After a short time the **TCPIP** indicator should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for **address** and **Port**, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct. The LED of the transient recorder should be lit up.

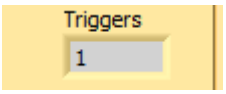


4. If you recognize that you are not connected to the Ethernet Controller you would like to use, just
   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and
   (b) press the 

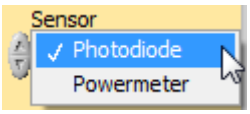
button.

   Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

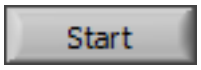
Once the TCP/IP connection is established the Power Meter Control software will request the number of power meter detectors (number of triggers) from the Power Neter Controller using the `POW NUMTRIG?` command. At older Power Meter Controllers the request fails and the number of triggers will be set to 1. The number of triggers will be shown on the tab page *TCP/IP* at the right of the TCP/IP controls.



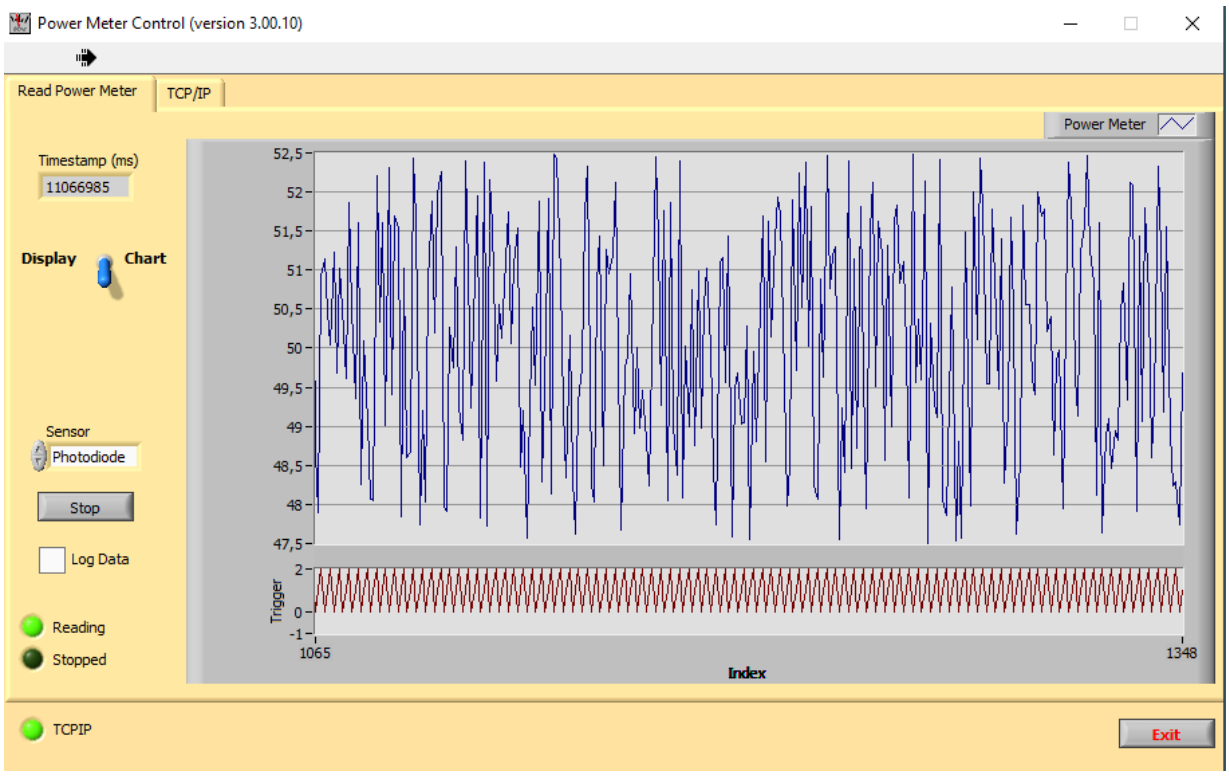
Then the **Sensor** needs to be selected so that it corresponds to the external detector (*Photodiode* or *Power Meter*).



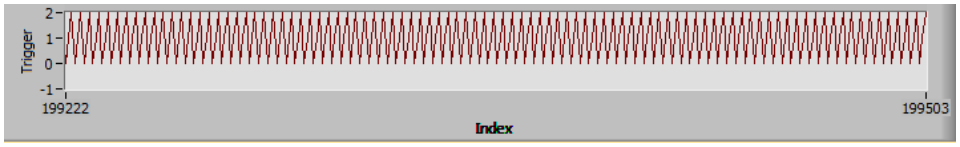
Press the start button to start acquiring data from the power meter controller.



The button text will change to *Stop*. The **Reading** LED will turn to light green. The acquired data is calibrated using the trigger-dependent calibration factors and offsets obtained from the initialization file and displayed in the top graphic indicator. At the bottom the indices of the received triggers are shown (here: always 0, we have only one trigger). In the case that an older controller does not return trigger information -1 is displayed as the trigger.

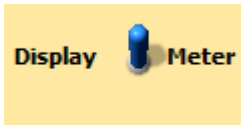


The following example results from a Power Meter Controller with three detector heads, all acquiring at the same frequncy.

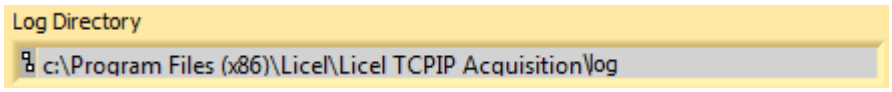


The display of the data can be in two modes, either history chart that shows the last 1024 readings or a power scale. The button text will change to *Stop*. Use the **Display** selection switch to change the display mode.

or



Check **Log Data** to write the acquired data to a file. Tab-seperated ASCII columns are written: the first column contains the timestamp corresponding to the time when the data was read from the controller. The second column contains the millisecond timer value returned by the controller, and the third contains the sensor values. Finaly, the fourth column consists of the trigger indices. The files are written to the sub directory `log`. The file name is YYYYmmdd_HH-MM-SS_uu.log where YYYYmmdd is the date, and HH-MM-SS the 24 hour time. uu stands for the first decimal places of the seconds. The complete path of the log file directory can be inspected on the *TCPIP* tab page:



An acquisition is stopped using the *Stop* button. After stopping the **Stopped** LED may turn to light green until all data already sent by the controller has been received.

6.4.1 Initialization File

The following settings are used in the initialization file `Power Meter Control.ini`.
Please note that the initialization file name(s) of the used instances of *Power Meter Control* will be `Power Meter Control<n>.ini` where *n* is a number starting at 1 when running *Power Meter Control* from *TCPIP Acquis*/*M-Acquis*. Just watch the initialization file indicator to be sure to use the correct file.

```
[TCPIP]
UseValues = TRUE
Port = 2055
IPAddress = "10.49.234.234"

[Power Meter Control]
HideRootWindow = True

[Data]
CalibrationFactor = 1
UnitLabel = ""
Offset = 0
CalibrationFactor1 = 1
UnitLabel1 = ""
Offset1 = 0
CalibrationFactor2 = 1
UnitLabel2 = ""
Offset2 = 0
```

`UseValues`	Use TCPIP values from ini file (Only for Windows applications)
`Port`	TCPIP Port
`IPAddress`	TCPIP address
`CalibrationFactor`	Conversion from binary controller data to physical unit (1st trigger)
`UnitLabel`	physical unit (1st trigger), this setting is used for the unit display in the GUI
`Offset`	Offset in physical units (1st trigger)
`CalibrationFactor1`	Conversion from binary controller data to physical unit (2nd trigger)
`UnitLabel1`	physical unit (2nd trigger, not used in the software)
`Offset1`	Offset in physical units (2nd trigger)
`CalibrationFactor2`	Conversion from binary controller data to physical unit (3rd trigger)
`UnitLabel2`	physical unit (3rd trigger, not used in the software)
`Offset2`	Offset in physical units (3rd trigger)

In older Power Meter Controllers or in controllers with only one detector the calibration factor and offset corresponding to the 1st trigger are used.

6.4.2 LabVIEW TCPIP Power Meter VIs

The low level TCP/IP commands supported by the Licel Power Meter Controller is described in the corresponding appendix.

6.4.3 Data Acquisition with TCPIP Acquis

The acquisition program *TCPIP Acquis* is capable to communicate with 1 or more instances of the Power Meter Control software to configure, start, and stop power meter data acquisitions and to read

the acquired and calibrated data to store it in the *TCPIP Acquis* data files together with the transient recorder data. For the configuration please refer to the TCPIP Acquis configuration.
When controlled by the Acquis software the needed instances of *Power Meter Control* are integrated into the container application *Multi Power Meter Control* which is described in the next subsection.

6.4.4 Multi Power Meter Control

Multi Power Meter Control is a variant of *Licel Main*. *Multi Power Meter Control* is intended to load several instances of `Power Meter Control.vi`. This mechanism is defined in an initialization file.

Initialization File

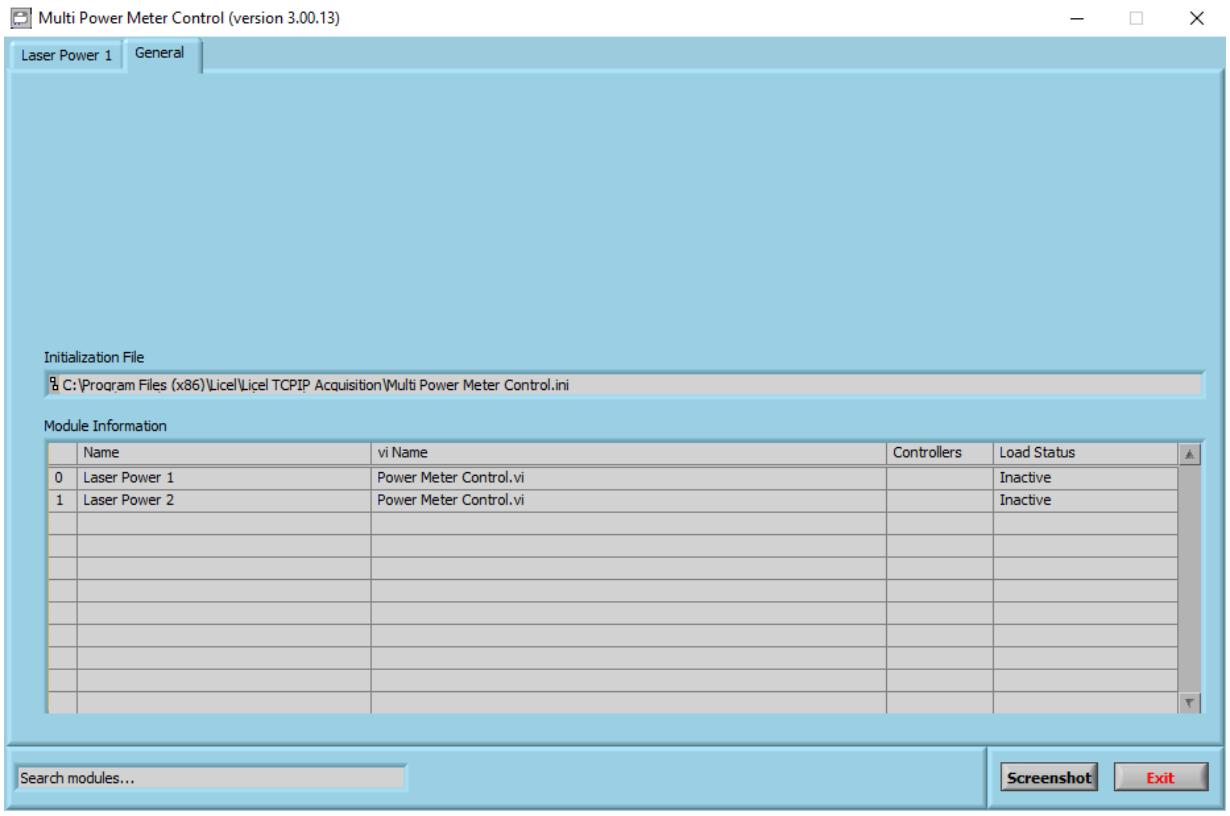
The syntax of the initialization file `Multi Power Meter Control.ini` is equal to that used by *Licel Main*. Each section `[Module<n>]` defines how to load an individual instance of `Power Meter Control.vi`.

```
; you may duplicate the sections named [Module<x>]
; containing the key 'Path = "Power Meter Control.vi"'
; to run more than 1 Power meter
; the key value of 'Name = ..' should always end up with
; a number TCPIP Acquis will automatically handle this file
[Module0]
Active = "TRUE"
Path = "Power Meter Control.vi"
Name = "Laser Power 1"
Controllers = "0"
CAPs = "POW"
Push = "FALSE"
IndependentTCPIP = "TRUE"
```

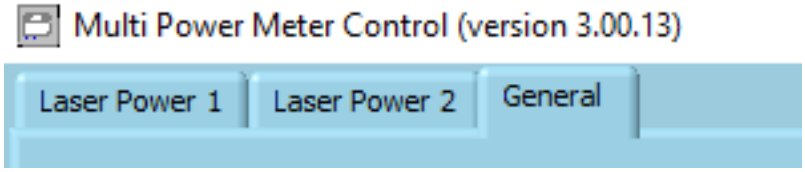
Each section must contain the initialization file key `IndependentTCPIP = "TRUE"` as each `Multi Power Meter Control` will communicate with a different Power Meter Controller.
In the case that *TCPIP Acquis* starts *Multi Power Meter Control* the initialization file is automatically generated or modified by the acquisition program — in the configuration dialog the number of power meters can be specified.

Starting the Multi Power Meter Control

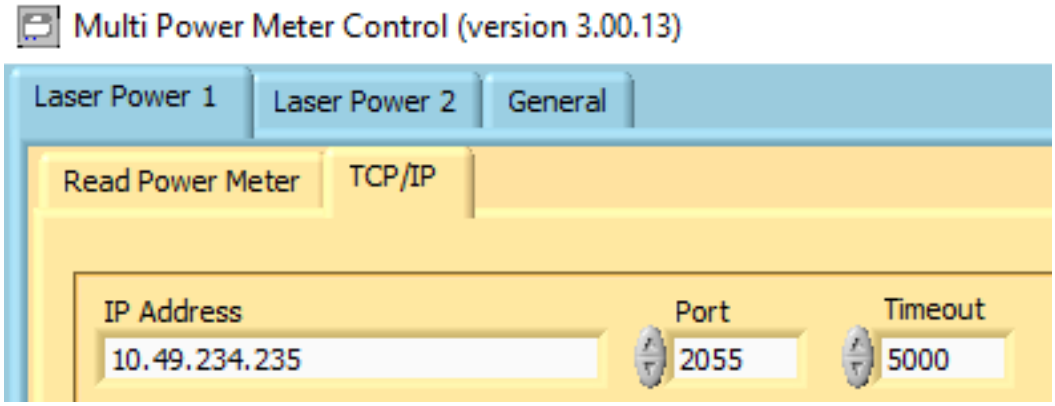
The Program can be started in the same way as *Licel Main*.
Please note that **in the case that the power data acquisition is integrated into *TCPIP Acquis*/*M-Acquis*** the used Acquis application will prepare all needed initialization files and will start *Multi Power Meter Control*. The individual IP addresses of all used Power Meter Controllers will already be entered in the Acquis configuration. Just the calibration factors and offsets have to be entered manually into the individual initialization files of the *Power Meter Control* instances.
The front panel contains a tab page with a table containing the information about the modules to load.



After starting the program will load and initialize the requested instances of the *Power Meter Control* software. For each instance a tab page will be added according to the settings in the initialization file.



When starting *Multi Power Meter Control* for the 1st time the user must set the IP addresses and ports for each of the loaded Power meter Control Instances. These values will be saved in the initialization files `Power Meter Control<n>.ini` for the *n*th instance and reloaded at the next start of the program. This mechanism known from the behavior of most Licel TCPIP Windows applications is used here in the case that the LabVIEW development environment is in use, as well.



Each instance of *Power Meter Control* will continue to run even if the tab page of the container application *Multi Power Meter Control* changes.

When controlled by *TCPIP Acquis*/*M-Acquis* the Sensor (*Photodiode* or *Powermeter*) will automatically be set according to the setting in the TCPIP Acquis configuration as well as the IP address and port. The start and stop of the power meter acquisition will automatically be controlled.

Chapter 7

Acquisition System Integration

Licel provides a number of standard software modules or example applications to communicate via TCP/IP with the available control modules. The software modules have been described above in this manual. All these TCP/IP software modules are capable to run stand alone.

7.1 Licel Main

In many cases a single Licel Ethernet Controller supports the control of different control modules, for example transient recorder and PMT high voltage control modules. To support all these capabilities attached to one IP address with a single software component a parent application named *Licel Main* has been designed. *Licel Main* embeds one or more TCP/IP software modules and is responsible for the handling of the TCP/IP connection.
Since the release of Licel Acquisition Software TCPIP version 2.61.00 *Licel Main* is principally capable to manage the control of up to six Licel Ethernet Controllers each having it's own IP address.
With the installation of the Licel Acquisition Software TCPIP *Licel Main* is installed in the following variants:

1. `Licel Main.exe`: classic variant for one rack with one Ethernet controller at one IP address
2. `Licel Main-M.exe`: basicly for two Ethernet Controllers, can be configured for up to six Ethernet controllers at six different IP addresses.

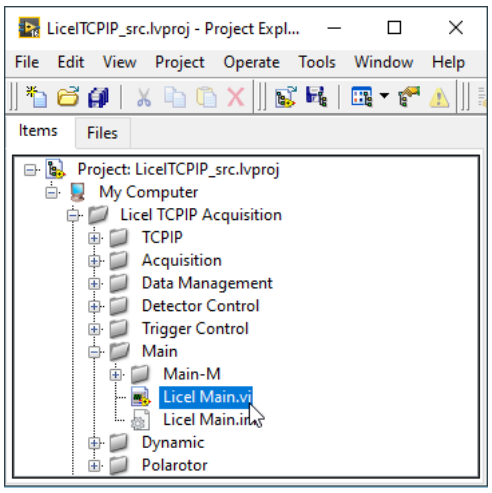
A Licel TCP/IP software module is capable to run within a sub panel on a tabulator page of the parent application *Licel Main*.

7.2 Licel Main – Short Tutorial

To understand how *Licel Main* is configured and how it is working it is recommended to run it first against Licel's simulation program *Virtual Controller*. Then you can try out all functions of the program. After this it is useful to inspect how *Licel Main* recieves TCP/IP information and definitions for software modules to load from the initialization file `Licel Main.ini`.
Please start the *Virtual Controller* from the Windows start menu as described in the corresponding chapter. Make sure that the switches *TR*, *PMT*, *APD*, and *TIMER* are active. That means that the *Virtual Controller* will support the simulation of transient recorders, a PMT and APD remote control, and a trigger generator. *Licel Main* will load the corresponding software modules to control the simulated hardware.

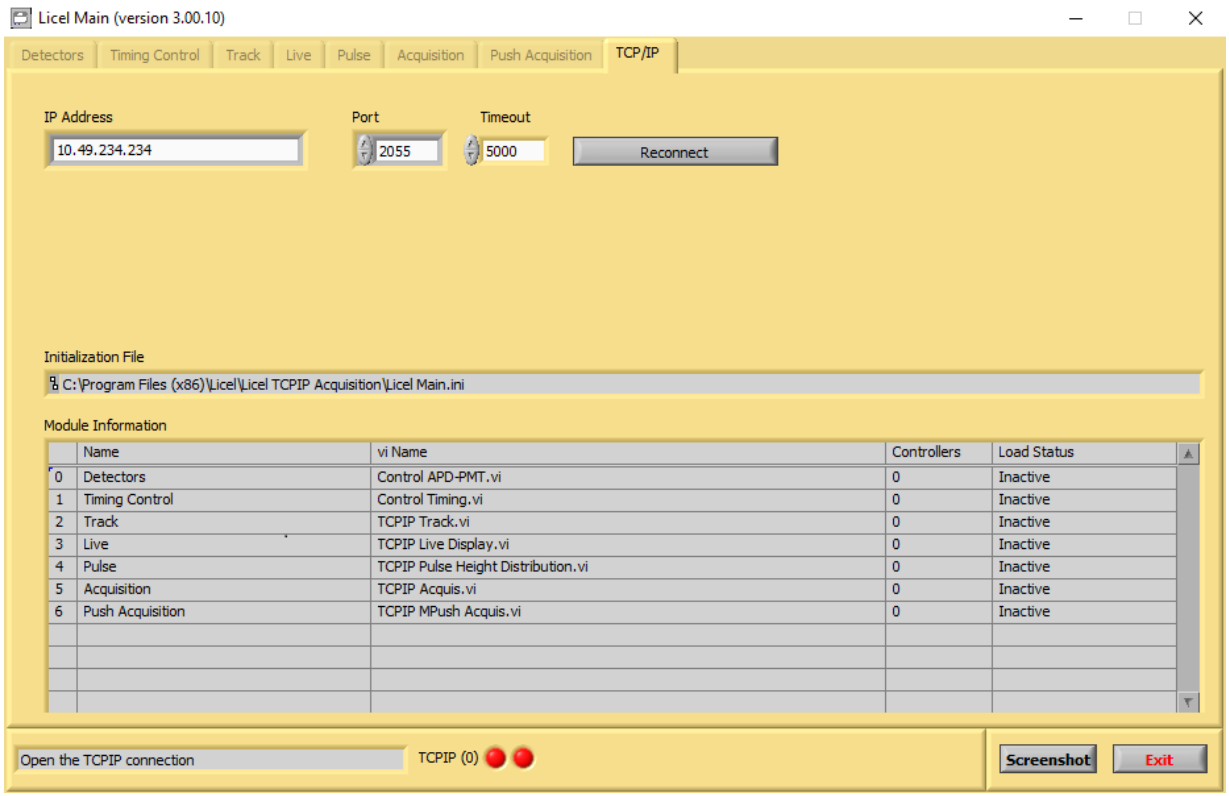
7.2.1 Starting Licel Main

- If you are using the LabVIEW sources open *Licel Main* from the LabVIEW project by navigating to the corresponding entry *Licel Main.vi* and double-clicking it.



- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

After loading a screen like the following will be visible (here the Windows application is shown):



1. First of all the *IP Address* and *Port* have to be set. You should already have set these values for the Licel Ethernet Controller following the network setup section above.

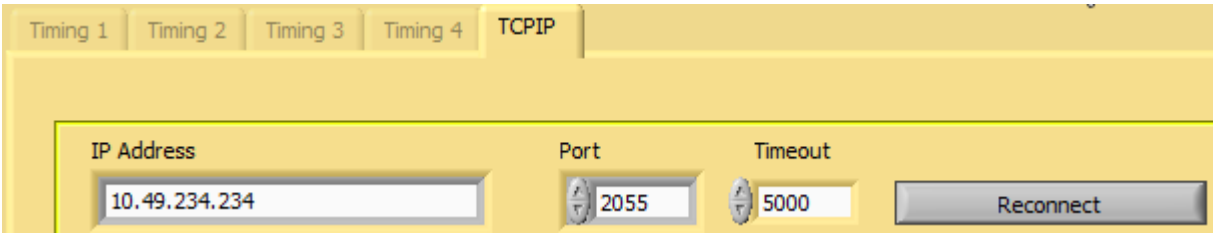
2. It is recommended to enter the correct values to the initialization file `Licel Main.ini`:

```
[TCPIP]
UseValues = TRUE
```

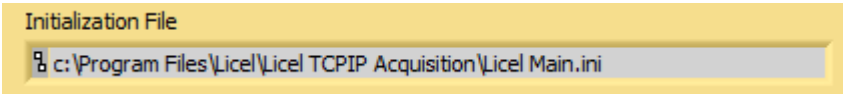
```
IPAddress = 10.49.234.234
Port=2055
```

Both, the LabVIEW VI and the Windows and Linux application will read these values while starting. If you run the tutorial with the *Virtual Controller*, just enter the IP address `127.0.0.1` (localhost).

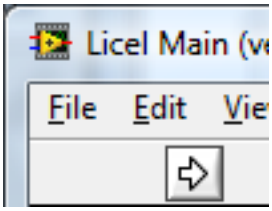
3. You may also enter the correct values into the corresponding control fields on the *TCP/IP* page *when the program is running*. *Gating Control* will using these values when opening a TCP/IP connection.



4. The values will be saved to the initialization file in the case that the TCP/IP connection has successfully been established.

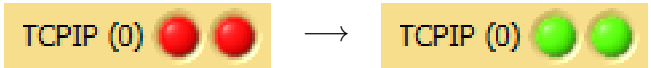


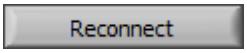
5. To start the program press the *Run* button at the top left of the screen.



The Windows application will start automatically when called for the first time.

6. After a short time the *TCPIP* indicator at the bottom should change its color from red to green indicating a successful connection with the Licel Ethernet Controller. If the indicator remains red and/or an error is indicated, please check the values for
textsIIP Address and *Port*, change them (on the program's panel or in the initialization file) if necessary. Check if the Licel Ethernet Controller is running and that all network connections are correct. The LED of the transient recorder(s) should be lit up if transient recorder control modules are loaded. If TCP-Live Display is loaded, a second LED indicator as seen below will be visible.



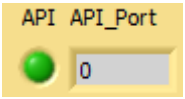
7. If you recognize that you are not connected to the Ethernet Controller you would like to use, just
   (a) Enter the *IP Address* and *Port* of that controller you really would like to connect to and
   (b) press the 

button.

   Also use the *Reconnect* button after changing the *IP Address* and/or *Port* while no connection is active.

8. Since version 1.70.01 *Licel Main* may be controlled externally via a TCP/IP API. If such a connection is active, the LED indicator and the used Port are displayed at the top right of the

TCP/IP tab page:



9. If the TCP/IP connections are alive when exiting the application (button *Exit*) the *IP Address* and *Port* will be written to the initialization file for the next start.

After starting the application will load all requested software control modules to the sub panels on the different tabulator pages if

1. the corresponding vi has been found
2. the controller's capabilities support the requested capabilities
3. the module could be successfully initialized.

Please refer to the initialization file `Licel Main.ini`.
The status of the software modules *Licel Main* attemps to load and run can be viewed in the table on the *TCP/IP* page. A software module can be loaded when the corresponding hardware is found.
In the last column it is indicated whether or not a software module is supported by the current Licel Ethernet Controller. In the example below `Control Timing` is not supported because the addressed Licel Ethernet Controller is installed at a Rack6 equipped with transient recorders and a PM-Remote control.

Module Information

	Name	vi Name	Controllers	Load Status
0	Detectors	Control APD-PMT.vi	0	Running
1	Timing Control	Control Timing.vi	0	Not Supported
2	Track	TCPIP Track.vi	0	Running
3	Live	TCPIP Live Display.vi	0	Running
4	Pulse	TCPIP Pulse Height Distribution.vi	0	Running
5	Acquisition	TCPIP Acquis.vi	0	Running
6	Push Acquisition	TCPIP MPush Acquis.vi	0	Running

7.2.2 Operating Licel Main

The following controls are available while operating *Licel Main*:

- *IP Address*: IP address of the accessed controller. It is recommended to set this value before starting the program, or to enter it to the initialization file `Licel Main.ini` for operating the windows application.
- *Port*: TCP/IP port for the communication with the accessed controller. It is recommended to set this value before starting the program, or to enter it to the initialization file `Licel Main.ini` for operating the windows application.
- *Timeout*: Timeout in milliseconds for the TCP/IP communication (default value: 5000 ms).
- Status indicators at the bottom:

  - *TCPIP* LEDs: TCP/IP alive indicators (see above)
  - *Status*: Status message indicator. After starting the program the following message sequence will be shown: *Open TCPIP connection(s)...*, *Search Modules...*, *Load Modules...*, *Initialize Module* <...> for all modules to be loaded, *Ready*.

- *Screenshot* A click on this button will send a screenshot of the current sub panel to the standard printer.
- *Exit* A click on this button will exit the application.

The different software modules may be accessed just by clicking on the corresponding tab pages. The software work as if they would run stand-alone. Only the monitoring of the TCP/IP connections is managed by *Licel Main*. If a TCPIP connection is lost *Licel Main* will take the control from the currently active vi, try to reconnect to the Licel Ethernet Controller, reset all currently used values (PMT voltage, timing values, etc.) after a successfully established connection, and return the control to the currently active vi.

7.2.3 TCP/IP Settings

The TCP/IP settings of *Licel Main* are always read from the initializtion file `Licel Main.ini` and always be written to it when a TCP/IP connection has successfully been established. Please note that this behavior is the same for the LabVIEW sources (`Licel Main.vi`) and the Windows application (`Licel Main.exe`) (in most other Licel applications the LabVIEW sources will use the *IP Address* and *Port* values from the corresponding control fields). Please note that for *Licel Main* with **one** Ethernet controller the IP address is read from the initialization file key `IPAddress` in the section `TCPIP`. `NoOfControllers` has to be set to 1.

```
[TCPIP]
UseValues = TRUE
IPAddress = 10.49.234.234
Port = 2055
NoOfControllers = 1

[TCPIP00]
IPAddress = 10.49.234.234
Port = 2055
```

Please note that the entry `NoOfControllers = 1` indicates that only one controller is in use. The section `[TCPIP00]` with distinct TCP/IP settings of this 1st controller is for documentation.

7.2.4 Software Module Load Definition

The initialization file `Licel Main.ini` contains also information for *Licel Main* about the software modules to load and run. As only **one** controller is in use all modules have to be assigned to the first controller at the index 0. This is done by the initialization key `Controllers = 0`. The complete module definition part of the initialization file `Licel Main.ini` right after the software installation will look like the following.

```
[Module0]
Active = TRUE
Path = Control APD-PMT.vi
Name = Detectors
Controllers = 0
CAPs = PMT,APD,PMTSPI
Push = FALSE
TCPIP_API = FALSE

[Module1]
```

```
Active = TRUE
Path = Control Timing.vi
Name = "Timing Control"
Controllers = 0
CAPs = TIMER
Push = FALSE
TCPIP_API = FALSE

[Module2]
Active = TRUE
Path = TCPIP Track.vi
Name = Track
Controllers = 0
CAPs = TR
Push = FALSE

[Module3]
Active = TRUE
Path = TCPIP Live Display.vi
Name = Live
Controllers = 0
CAPs = TR
Push = TRUE
TCPIP_API = FALSE

[Module4]
Active = TRUE
Path = TCPIP Pulse Height Distribution.vi
Name = Pulse
Controllers = 0
CAPs = TR
Push = FALSE

[Module5]
Active = TRUE
Path = TCPIP Acquis.vi
Name = Acquisition
Controllers = 0
CAPs = TR
Push = FALSE
TCPIP_API = FALSE

[Module6]
Active = TRUE
Path = TCPIP MPush Acquis.vi
Name = Push Acquisition
Controllers = 0
CAPs = TR
Push = TRUE
```

A software module load description corresponds to an initialization file section `[Module<number>]`. The `number`s define the order of the corresponding tab pages and must be unique. The initialization file keys of such a software module section contains to the following parameters:

ACTIVE	TRUE | FALSE	load the vi or not
Path	<vi path>	name of the vi (without llb)
Name	tab label	is displayed as tab label
Controllers	controller index	(the 1st index is 0)
CAPs	comma-separated list of capabilities	Licel Main will check if at least one of these capabilities is supported by the controller(s)
Push	TRUE | FALSE	use the push socket of the controller(s) (TCPIP Live Display, MPush Acquis)
IndependentTCPIP	TRUE | FALSE	the loaded module will use a different Ethernet Controller not managed by *Licel Main*, the feature is supported by the software modules *Control APD-PMT*, *Power Meter Control*, and *PMT_TEC Datalogger*. When changing the tab page in *Licel Main* the module will continue to run.
TCPIP_API	TRUE | FALSE	used when *Licel Main* is controlled via the TCP/IP API and when it should pass incoming TCP/IP commands to the sub module. Please refer to the linked appendix for details.
IniName	<IniFileName>	used when a loaded module should explicitly use a certain initialization file. The *IniFileName* should include the ending .ini. This setting is supported by *TCPIP Acquis*, *TCPIP M-Acquis*, *TCPIP MPushAcquis*, *TCPIP Live Display*, *Control Timing*, *Power Meter Control*. Always use this setting if you include several instances of a module VI accessing different Ethernet controllers.

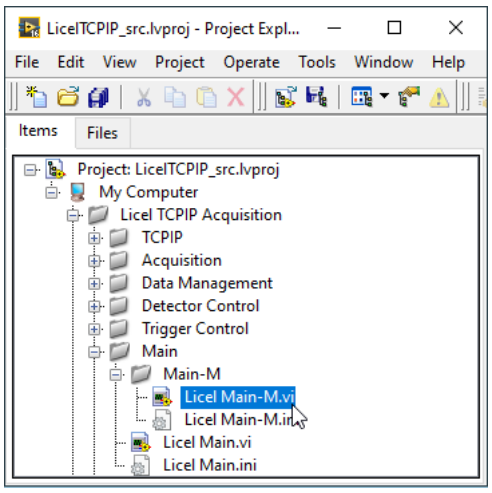
In the case of *Licel Main* the controller index is always 0 (Controllers = 0), i.e. *Licel Main* will always use the first (and only) Licel Ethernet Controller for each software module. Modules for which the capabilities are not supported by the controller are just disabled; that means that no tab page is shown for such a software module.
With it's standard initialization file Licel Main.ini *Licel Main* supports **one** Licel Ethernet controller for the control of transient recordes, PMT and APD remote controls, and/or a trigger generator. If you are using the hardware simulation *Virtual Controller* you may also simulate a rack consisting of transient recorders and a PMT remote control: just disable the switches *APD* and *TIMER* before starting *Licel Main*. You will see that *Licel Main* will not have a timing tab, and Control APD-PMT will not show APD-relevant information.

7.3 Licel Main-M

Licel Main-M is a variant of *Licel Main* to support the communication and data acquisition with up to six Licel Ethernet Controllers with individual IP addresses. *M* stands for *Multiple*. *Licel Main-M* is available as a LabVIEW VI (Licel Main-M.vi) and as a Windows application (Licel Main-M.exe). In it's standard configuration after the installation *Licel Main-M* will support **two** Ethernet controllers, each of them controlling transient recorders and/or APD/PMT remote controls.

7.3.1 Starting Licel Main-M

- If you are using the LabVIEW sources open *Licel Main-M* from the LabVIEW project by navigating to the corresponding entry *Licel Main-M.vi* and double-clicking it.



- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu.

Licel Main-M uses the initialization file `Licel Main-M.ini`. With this initialization file *Licel Main-M* will work with two racks, each equipped with transient recorders and PM-Remote controller. This ini file may be exended for the use of up to six Licel Ethernet Controllers.

7.3.2 TCP/IP Settings

The TCP/IP settings of *Licel Main-M* are always read from the initializtion file and always be written to it when the TCP/IP connection is alive while terminating the application using the *Exit* button and `[TCPIP] UseValues = TRUE`. Please note that this behavior is the same for the LabVIEW sources (`Licel Main-M.vi`) and the Windows application (`Licel Main-M.exe`) (in most other Licel applications the LabVIEW sources will use the *IP Adress* and *Port* values from the corresponding control fields). The following part of the initialization file `Licel Main-M` shows the TCP/IP definition:

```
[TCPIP]
UseValues = TRUE
Port = 2055
IPAddress = 10.49.234.234
NoOfControllers = 2

[TCPIP00]
IPAddress = 10.49.234.234
Port = 2055

[TCPIP01]
IPAddress = 10.49.234.235
Port = 2055
```

In the section `TCPIP` the key `NoOfControllers = 2` defines that **two** Ethernet controllers are in use. The individual TCP/IP properties are obtained from the sections
`[TCPIP00]`, `[TCPIP01]` (, ...).
The settings `IPAddress` and `Port` below `[TCPIP]` are used for compatibility with the settings for *Licel Main* (`NoOfControllers = 1`).
The user interface of *Licel Main-M* is equal to that of *Licel Main* with the exception of the IP addresses and ports and the status indicator. The controls allow to set all IP addresses and ports corresponding to the definition in the initialization file `Licel Main-M.ini`.

Additional TCP/IP indicator LEDs are shown for the second TCP/IP connection.

7.3.3 Software Module Load Definition

The behavior of *Licel Main-M* and the loaded software modules is defined in the initialization file `Licel Main-M.ini`, as well. Each software module section `[Module<number>]` contains the entry `Controllers` to specify whether the first (`Controllers` = 0) or the second rack (`Controllers` = 1) is used. The `number`s define the order of the corresponding tab pages and must be unique. Here you see the standard content of the module definitions in `Licel Main-M.ini` right after the software installation:

```
[Module0]
Active = TRUE
Path = Control APD-PMT.vi
Name = Detectors Rack 1
Controllers = 0
CAPs = PMT,APD
Push = FALSE
TCPIP_API = FALSE

[Module1]
Active = TRUE
Path = Control APD-PMT.vi
Name = Detectors Rack 2
Controllers = 1
CAPs = PMT,APD,PMTSPI
Push = FALSE
TCPIP_API = FALSE

[Module2]
Active = TRUE
Path = TCPIP Live Display.vi
Name = Live 1
Controllers = 0
CAPs = TR
Push = TRUE

[Module3]
Active = TRUE
Path = TCPIP Live Display.vi
Name = Live 2
Controllers = 1
CAPs = TR
Push = TRUE
TCPIP_API = FALSE
```

```
[Module4]
Active = TRUE
Path = TCPIP Track.vi
Name = Track 1
Controllers = 0
CAPs = TR
Push = FALSE

[Module5]
Active = TRUE
Path = TCPIP Track.vi
Name = Track 2
Controllers = 1
CAPs = TR
Push = FALSE

[Module6]
Active = TRUE
Path = TCPIP Pulse Height Distribution.vi
Name = Pulse 1
Controllers = 0
CAPs = TR
Push = FALSE

[Module7]
Active = TRUE
Path = TCPIP Pulse Height Distribution.vi
Name = Pulse 2
Controllers = 1
CAPs = TR
Push = FALSE

[Module8]
Active = TRUE
Path = M-Acquis.vi
Name = Acquisition
Controllers = 0;1
CAPs = TR
Push = FALSE
TCPIP_API = FALSE
```

Modules with the key `Controllers = 0` are assigned to the first Ethernet controller, those with `Controllers = 1` are using the second one. Licel's multi-rack acquisition software *M-Acquis* uses both controllers to obtain synchronized data from both Ethernet controllers: here, `Controllers = 0;1` assigns the module to both.

7.4 Licel Main Applications

7.4.1 When do I use *Licel Main*?

You can use *Licel Main* without modifying the module load descriptions in the initialization file `Licel Main.ini` when you have **one** rack with **one** Licel Ethernet Controller with **one** single IP address

containing one or more of the following hardware:

- transient recorders
- APD and/or PMT remote control
- Trigger Controller (one sub board only)

Just enter the `IPAddress` and `Port` in the `TCPIP` section of the initialization file.

7.4.2 When do I use *Licel Main-M*?

You can use *Licel Main-M* without modifying the load descriptions in the initialization file `Licel Main-M.ini` when you have **two** Licel Ethernet Controllers with individual IP addresses containing the following hardware:

- transient recorders
- APD and or PMT remote controls

Just enter the `IPAddress`es and `Port`s in the `TCPIP<number>` sections of the initialization file.

7.4.3 What should I do when I ...

... have one TR Rack and a Remote Control Rack?

Use *Licel Main-M* after modifying the module load description in `Licel Main-M.ini`:

```
[Module0]
Active = TRUE
Path = Control APD-PMT.vi
Name = Detectors
Controllers = 1
CAPs = PMT,APD,PMTSPI
Push = FALSE
TCPIP_API = FALSE

[Module1]
Active = TRUE
Path = TCPIP Track.vi
Name = Track
Controllers = 0
CAPs = TR
Push = FALSE

[Module2]
Active = TRUE
Path = TCPIP Live Display.vi
Name = Live
Controllers = 0
CAPs = TR
Push = TRUE
TCPIP_API = FALSE

[Module3]
```

```
Active = TRUE
Path = TCPIP Pulse Height Distribution.vi
Name = Pulse
Controllers = 0
CAPs = TR
Push = FALSE

[Module4]
Active = TRUE
Path = TCPIP Acquis.vi
Name = Acquisition
Controllers = 0
CAPs = TR
Push = FALSE
TCPIP_API = FALSE

[Module5]
Active = TRUE
Path = TCPIP MPush Acquis.vi
Name = Push Acquisition
Controllers = 0
CAPs = TR
Push = TRUE
```

Please note that the first controller (0) corresponds to the transient recorder rack, the second (1) to the PMT remote control. The IP addresses and ports have to be specified in the `TCPIP<number>` sections.
Do not forget to enter the `IPAddresses` and `Ports` in the `TCPIP<number>` sections of the initialization file.

... have a PMT_TEC Remote Control?

Please make sure that the following software modules are part of `Licel Main.ini` or `Licel Main-M.ini`. Note that the numbers in the `Module<Number>` must be unique and could be different as in this example. The Controller assignment of the *Control APD-PMT* can be different, too.

```
[Module0]
Active = TRUE
Path = PMT_TEC DataLogger.vi
Name = TEC Control
IndependentTCPIP = TRUE

[Module1]
Active = TRUE
Path = Control APD-PMT.vi
Name = Detectors
Controllers = 0
CAPs = PMT,APD,PMTSPI
Push = FALSE
TCPIP_API = FALSE
```

Here, the *PMT_TEC Datalogger* is part of the initialization file. `IndependentTCPIP = TRUE` will make the data logger to use it's own IP address and to continue operation when it's tab page in *Licel Main* is not active.

7.5 Licel Main Derivates

With the Licel Acquisition software further applications based on *Licel Main* are installed. All of them are technically equal to *Licel Main*. Just the front panel (window) appearance varies.

1. *Gating Control* is used to control a Trigger Controller with more than one sub-board.
2. *Polarotor* is used to control a Polarotor.
3. *Multi Power Meter Control* is used to control one or more Power Meters from *TCPIP Acquis*. *TCPIP Acquis* will dynamically start and stop *Multi Power Meter Control* and it is responsible to generate the required initialization file according to the number of power meters.

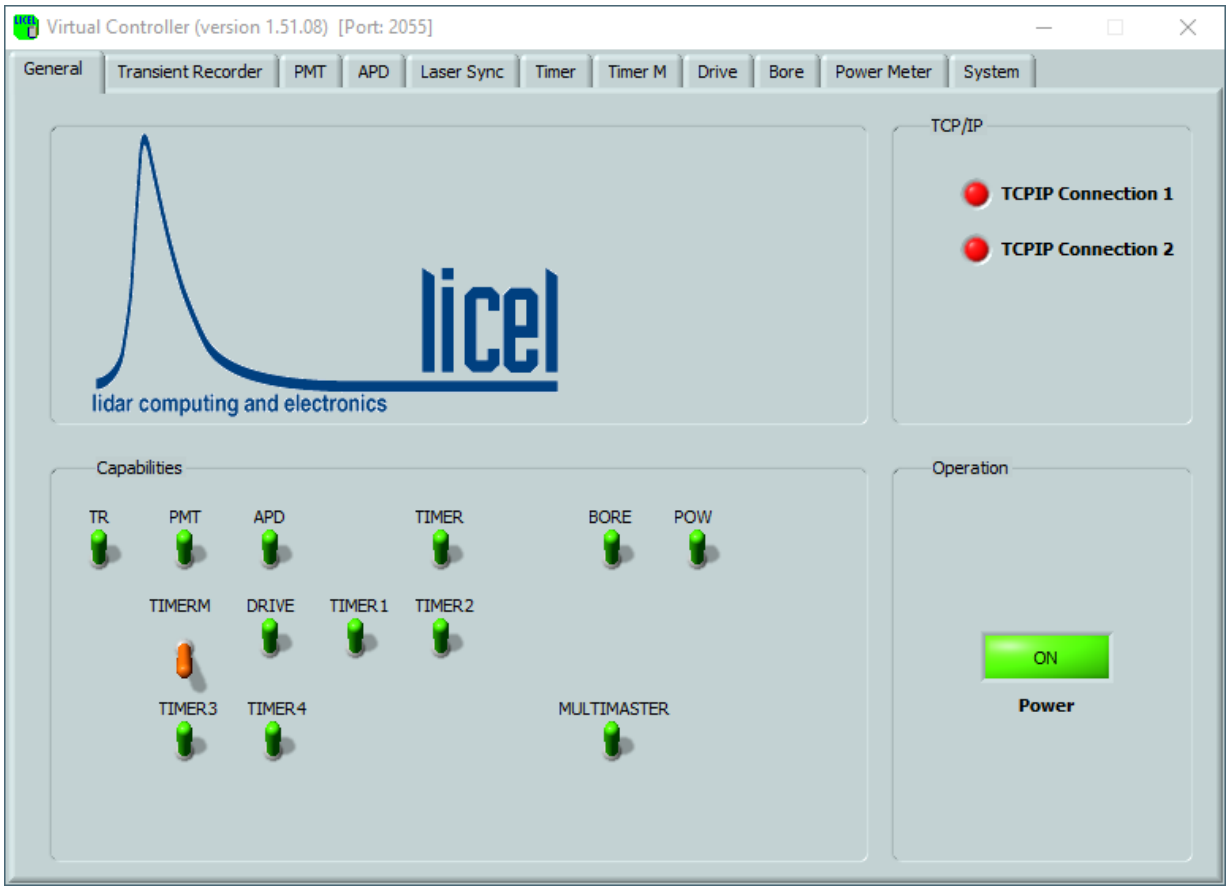
Chapter 8

The Licel Virtual Controller

The Licel Virtual Controller simulates the behavior of a Licel Ethernet Controller. This will help a user to develop and test his own software as debugging is possible without having the hardware installed. The software can substitute hardware while testing the other LabVIEW modules provided by Licel or while developing custom software. The Licel Virtual Controller supports the TCPIP commands related with the standard functionality of a Licel Ethernet Controller. Furthermore it is able to simulate realistic backscatter signals to test LIDAR acquisition software.

8.1 Starting the Application

The Licel Virtual Controller is started directly from the Windows Start menu and opens with the following screen.



The Licel Virtual Controller is waiting for applications to connect to it via TCP/IP. These applications must use the IP address of the computer where the Licel Virtual Controller is running. If the appli-

cation you would like to run against the Virtual Controller is located on the same machine, just use `127.0.0.1` (localhost) as the IP address. The Licel Virtual Controller is listening on port 2055 which equals the standard port of the Licel hardware. The port may be changed on the *System* tab page. Note that the choosen port must not be used by other programs and not be blocked by any firewall.
The Virtual Controller provides tab pages for each capability and furthermore the pages *General* and *System*.

8.2 Initialization File

On start the Virtual Controller searches for an initialization file named like the executable application (default: `Virtual Controller.ini`) with the following entries:

```
[TCPIP]
Port=2055
[UDP]
Port=2000
[CAPS]
File="standard.cps"
```

`Port` in the `[TCPIP]` section determines the port for the TCP/IP command socket. The Virtual Controller uses the ports `Port`...`Port+2`. `Port` in the `[UDP]` section determines the UDP port. The entry `File` in the `[CAPS]` section may only be used if Licel provided you with such a file. `cps` encode information about the virtual capabilities of the Virtual Controller. If no file is specified the Virtual Controller has the default capabilities.
If no initialization file is present the Virtual Controller uses default values.
Initialization files must be used to run more than one Virtual Controller. Follow the steps below to configure a system with 2 Virtual Controllers:

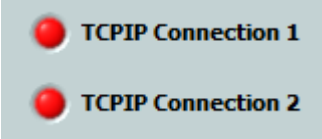
1. Create a file `Virtual Controller.ini` and enter the values for the `[TCPIP]` and `[UDP]` sections
2. Copy the application `Virtual Controller.exe` to `Virtual Controller1.exe`
3. Copy the initialization file `Virtual Controller.ini` to `Virtual Controller1.ini`
4. Change the `port` values in `Virtual Controller1.ini` to 2065 for `[TCPIP]`, and 2001 for `[UDP]`
5. Start both Virtual Controllers, one is now accessable by your software at port 2055, the second at 2065.

8.3 General

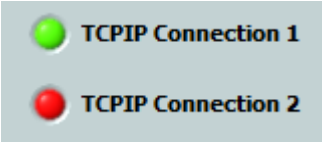
On the tab page *General* you may inspect and set the following parameters:

Capabilities Each switch here represents a capability of the Virtual Controller. The standard capabilities are **TR**, **PMT**, **APD**, **TIMER**, **BORE**, **POW**, **TIMERM**, **DRIVE**, **TIMER1**, **TIMER2**, **TIMER3**, **TIMER4**, and **MULTIMASTER**. Licel may provide more capabilities for customized systems. The active capabilities will be part of the answer to the CAP? command. If you switch a capability off, corresponding commands will no longer be accepted. This is useful for debugging purposes to implement an appropriate error handling at the calling application.

TCP/IP Indicators **TCPIP Connection 1** and **TCPIP Connection 2** indicate wether applications are connected via TCP/IP with the Virtual Controller.



If both indicators are red, no client is connected. **TCPIP Connection 1** become green when an application uses the command socket on the specified port.



If an application used the second TCP/IP socket on a port number advanced by 1 (push socket), as well, the second indicator would become green. You may test this with the example applications `TCPIP Live Display` with an **Update #** smaller than 15 for 12 bit TR and 1 Shot for 16bit unit or `TCPIP MPush Acquis` from the Licel TCPIP Acquisition Software package without any shot restriction.

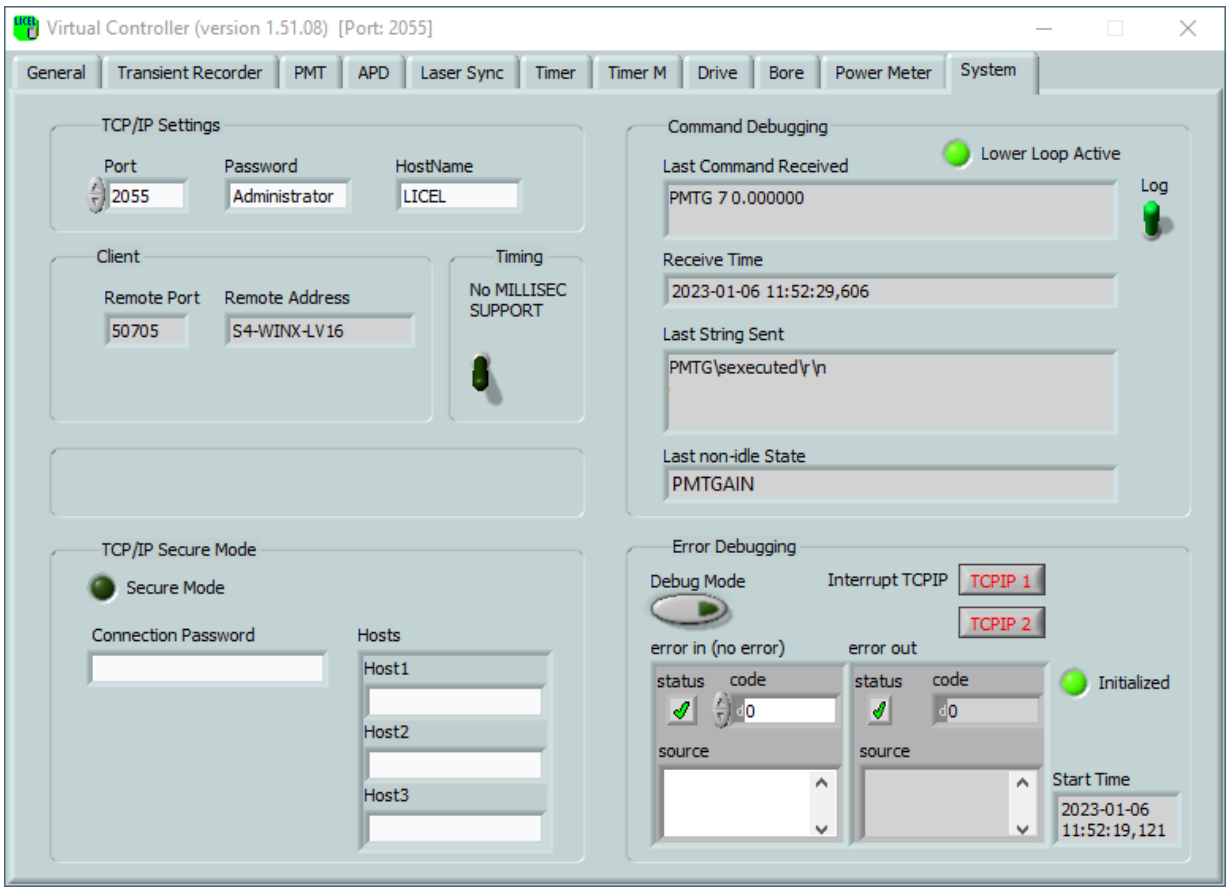
Power By clicking the **Power** button the Virtual Controller will be stopped.

You may restart it using LabVIEW's run button 

Close the *Virtual Controller*'s window as usual with the Windows close button ✕ at the top right of the window.

8.4 System

On the *System* tab page system and debugging options are available.



TCP/IP Settings In this section the **Port** and the **Password** for the Virtual Controller may be changed. Note that **Port** is used directly for the command socket (TCPIP Connection 1), and **Port**+1 is the port for the push socket (TCPIP Connection 2). The **Password** is the administrator password of the Virtual Controller which is needed to be sent with certain commands like the command to change the IP address. Note that a change of the IP address of the Virtual Controller is not possible, as the IP address is set in the computer's system setup. The **Hostname** is used to simulate Ethernet Controllers with the corresponding capability.

Client When a client is connected to the Virtual Controller it's **Remote Port** and **Remote Address** are displayed here.

Timing Here, the millisecond timer support (MILLISEC? command) of the Virtual Controller can be switched off to simulate very old Ethernet controllers.

TCP/IP Secure Mode This section is related to the secure mode and the related TCP/IP commands WHITELIST and ACCESS.

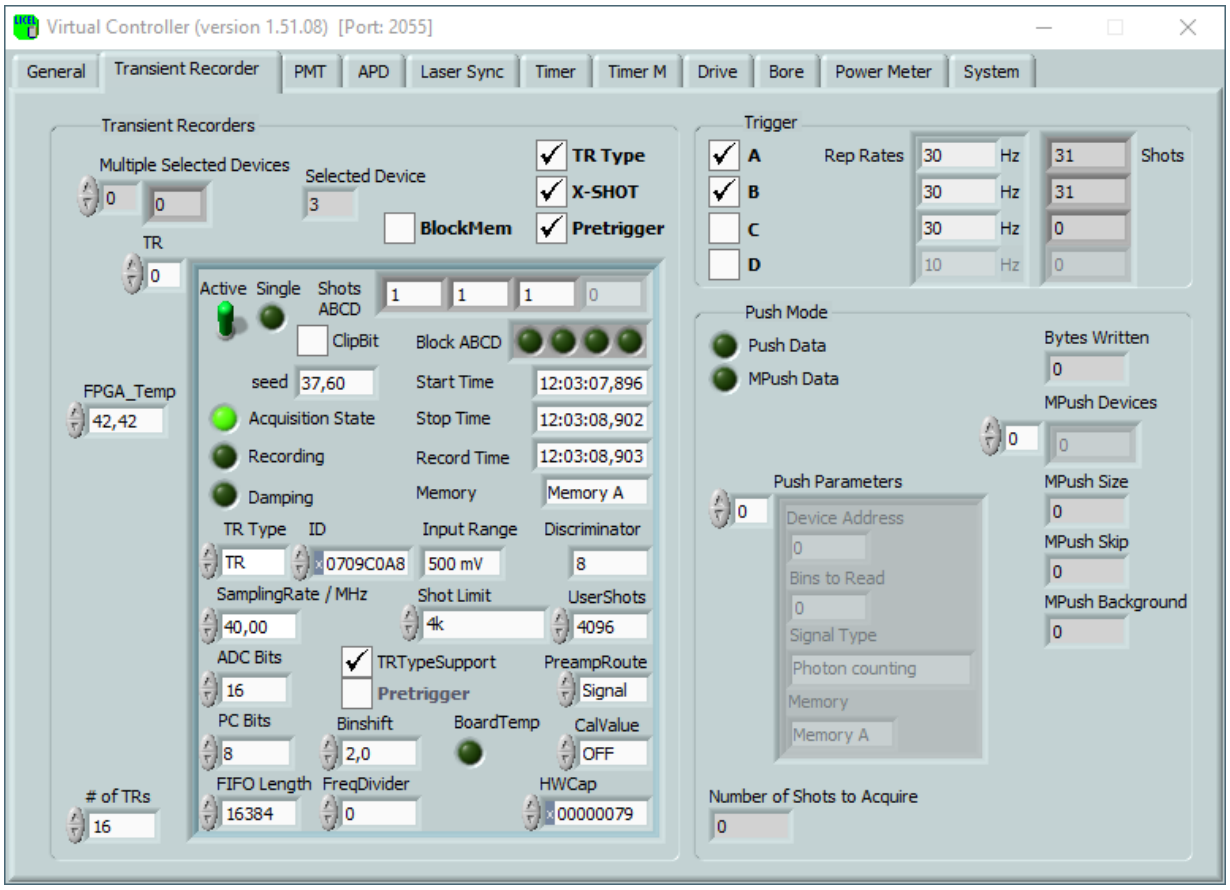
- **Secure Mode** Controls whether or not the secure mode is active. Normally the secure mode is enabled with the ACCESS command, for debugging it can be activated by clicking.
- **Connection Password** The connection password to be used to login.
- **Hosts** The allowed host or address ranges set by the WHITELIST command.

Command Debugging This part of the front panel contains information about the TCP/IP communication flow (**Last Command Received**, the corresponding **Receive Time** and **Last String Sent**), and about the loop activities of the Virtual Controller (**Last non-idle State** and **Lower Loop Active**). **Last non-idle State** describes the internal state of the command socket loop of the program while **Lower Loop Active** indicates that the push socket loop to handle requests on **Port**+1 is alive. This indicator should always light in a green color. Switching the **Log** switch to the up position will make the `Virtual Controller` write all received commands to a log file named `Virtual Controller.exe.log`.

Error Debugging This section contains the standard LabVIEW error clusters. If **Debug Mode** is set errors in the Virtual Controller are shown. Timeout errors (code=56) are normal during operation. The buttons **Interrupt TCPIP** (**TCPIP 1** and **TCPIP 2**) may be used to interrupt the TCP/IP connections to inspect the behavior of an application to such an event. This is especially useful if a mechanism to reconnect has been implemented to the application when TCP/IP relevant errors occur. The LED indicator **Initialized** shows whether or not the program has been initialized.

8.5 Transient Recorder

The controls and information located on this tab page correspond to the TR capability. When a transient recorder setting is changed or while an acquisition is running you will find the corresponding information here.



The following controls and indicators are available:

Left: Transient Recorders

Selected Device The currently SELECTed device number for single functions like START and STOP.

Multiple Device Numbers An array containing the device IDs of the active transient recorders if any have been selected using SELECT.

# of TRs Number of transient recorders at the Virtual Controller.

TR Type Control whether or not the Virtual Controller understands the TRTYPE? command.

X-Shot Control whether or not the Virtual Controller understands the extended shot commands SHOTAB? and MSHOTSAB?.

Pretrigger Control whether or not the Virtual Controller understands the PRETRIG command.

BlockMem Control whether or not memory blocking is supported.

TR Contains the current status of the transient recorders. You may switch between the different TRs by changing the device number located top-left of this control. **Tr** contains in detail:

- **Active | OFF** Set the transient recorder at the current index (TR address) active or not (= not installed). This setting is stored in initialization file for future usage.
- **Shots ABC** Number of shots acquired at the memories A,B, and C still stored in the transient recorder that are to be added to the new acquisition.
- **Start Time** Time of the last start.
- **Stop Time** Time of the last stop. If the **Start Time** is larger than the **Stop Time** the transient recorder is acquiring data, otherwise the transient recorder is inactive.
- **Record Time** The Record Time is used to control whether or not a new seed for simulated data generation should be created or not. In this way, the seed for generating the data is not changed if the acquisition is continued instead of a new acquisition being started.
- **Seed** Used for generating random white noise for the simulated lidar signal.
- **Acquisition State** TRUE when the transient recorder returns from the armed state, FALSE, when an acquisition is running.
- **Recording** TRUE during acquisition-time, e.g. the ADC or the photon counting is acquiring data. Recording is FALSE during summation and when the device is waiting for a new trigger.
- **Damping** Indicates the threshhold range in photon counting mode. *Off*: the current discriminator level is used, *On* the current discriminator level multiplied by 4 is used.

Block ABC Indicate whether or not the memory blocking is active for the meories A, B, or C.

Input Range Shows the analog input range of the preamplifier. The signal start at 0 and reaches -20, -100, -500 mV. (0=-500mV,1=-100mV,2=-20mV).

Discriminator For a device with a photon counting unit, the discriminator threshold can be set in 64 steps between 0 and 63.

Memory Shows to which summation memory (Memory A or B) the last acquisition was added.

TR Type Type of the transient recorder: 0 (TR) = analog and photon counting device, 1 (PR) = pure photon counting device. This value is changeable here. This setting is stored in initialization file for future usage.

ID An ID of the transient recorder. This is a simulated value dependent on the **Port** and the device ID (index of the transient recorder array). The value is set 0 only if both checkboxes **TR Type** and **TRTypeSupport** are checked.

Sampling Rate Sampling rate in MHz of the transient recorder. This value is changeable here if this information is not stored in the transient recorder. The setting is always stored in initialization file.

ADC Bits Number of ADC Bits. If supported this information will be submitted with the the `TRTYPE?` command.

PC Bits Number of PC Bits. If supported this information will be submitted with the the `TRTYPE?` command.

Shot Limit

- *Not Supported* Setting the shot limit is not supported i.e. the `LIMIT` command returns an error.
- *4k* The shot limit is 4k shots.
- *64k* The shot limit is 64k shots.

UserShots Limit of the `SETMAXSHOTS` command, range 2k . . . 64k.

TRTypeSupport Check this box if the individual simulated transient recorder should supports the `TRTYPE?` command.

Pretrigger Check this box if the individual simulated transient recorder should supports the `PRETRIG` feature.

Bin Shift Bin shift of ADC data bins with respect to photon counting bins. If supported this information will be submitted with the the `TRTYPE?` command.

FreqDivider Freqeuncy divider - control the hardware binning of the transient recorder. Supported only if the 7th bit(HWCap) is set (HWCap AND 0x40).

FIFO Length FIFO length. If supported this information will be submitted with the the `TRTYPE?` command. Currently the Virtual Controller does not use this value when generating and returning data.

HWCap Hardware capabilities of the transient recorder.

- `0x01` shot counter B support
- `0x02` shot counter C support
- `0x04` shot counter D support (not yet used)
- `0x08` pretrigger support (not used in simulations)
- `0x10` block memory support
- `0x20` squared data support
- `0x40` frequency divider support
- `0x80` triple reference voltage support
- `0x100` apd-flex

Right Top: Trigger

Checkboxes Here the triggers A, B, and C can be enabled (checked) or disabled

Laser Rep Rates Enter the laser repetition rates (Hz) here

ShotsABC Acquired shots at for memories A, B, and C

Right Bottom: Push Mode

Push Data Shows whether or not the PUSH mode is active.

MPush Data Shows whether or not the multiple push mode MPUSH is active.

Push Parameters Push parameters used for push and mpush modes:

Device Address Device address of the active transient recorder.

Bins to Read The number of bins to read from the appropriate memory.

Signal Type Specifies which part of the raw information should be transfered from the device to the computer.

Memory Shows the memory to read from.

Number of Shots to Acquire The number of shots to acquire in push mode.

MPush Devices An array containing the device IDs of the transient recorders addressed in the MPush mode.

MPush Size Total number of bytes to send in MPush mode.

MPush Skip Number of bins to skip after **MPush Size** in MPush mode as defined by the `MPUSHBACK` command. Defines the background start = MPush Size + MPush Skip.

MPush Background Number of background bins to append in MPush mode as defined by the `MPUSHBACK` command.

MPush Mode The MPush Mode determines whether or not a timestamp is sent with the transmitted data and how the data is organized.

- *No Timestamp* The MPush data does not contain a timestamp (for older controllers).

- *Standard* A timestamp is send in MPush mode. This format is expected by `TCPIP MPush Acquis`
- *MPush AB* The MPush AB mode is available only in customized controllers.

8.5.1 LIDAR Signal Simulation

The Licel Virtual Controller simulates backscatter signals and sends the data via TCPIP to the calling client. Therefore, the Virtual Controller functions in exactly the same way as a real detection system. As a result the software acquisition programs will observe no difference between a simulated and a real signal.
The simulation starts with molecular back scattering for 550 nm at the standard height read from the files `standard_back.txt` and `standard_height.txt`, respectively. A backscatter signal is simulated and the overlap with the optical detection system is calculated. Finally some random white noise is added and the data is prepared to be sent via TCPIP. For the simulation a default laser repetition rate of 10 Hz is assumed, the rate is changeable on the front panel.
The shape of the overlap function is controlled by the coefficients read from the initialization file `Overlap_Globals.ini`:

```
[Overlap]
Beam_Radius=2.000000
Beam_Divergence=0.002000
Beam_Inclination_Parallel=0.000000
Beam_Inclination_Perpendicular=0.000000
Telescope_Diameter=0.600000
Detector_Radius=1.000000
Focal_Length=1800.000000
Axial_Separation=800.000000
Noise_Amplitude=1.000000
Sqrt_Signal_Noise_Amplitude=0.120000
```

These parameters are read when the Licel Virtual Controller is started. They may be changed in the initialization file according to your own optical system to simulate a realistic backscatter signal you will measure with your hardware.

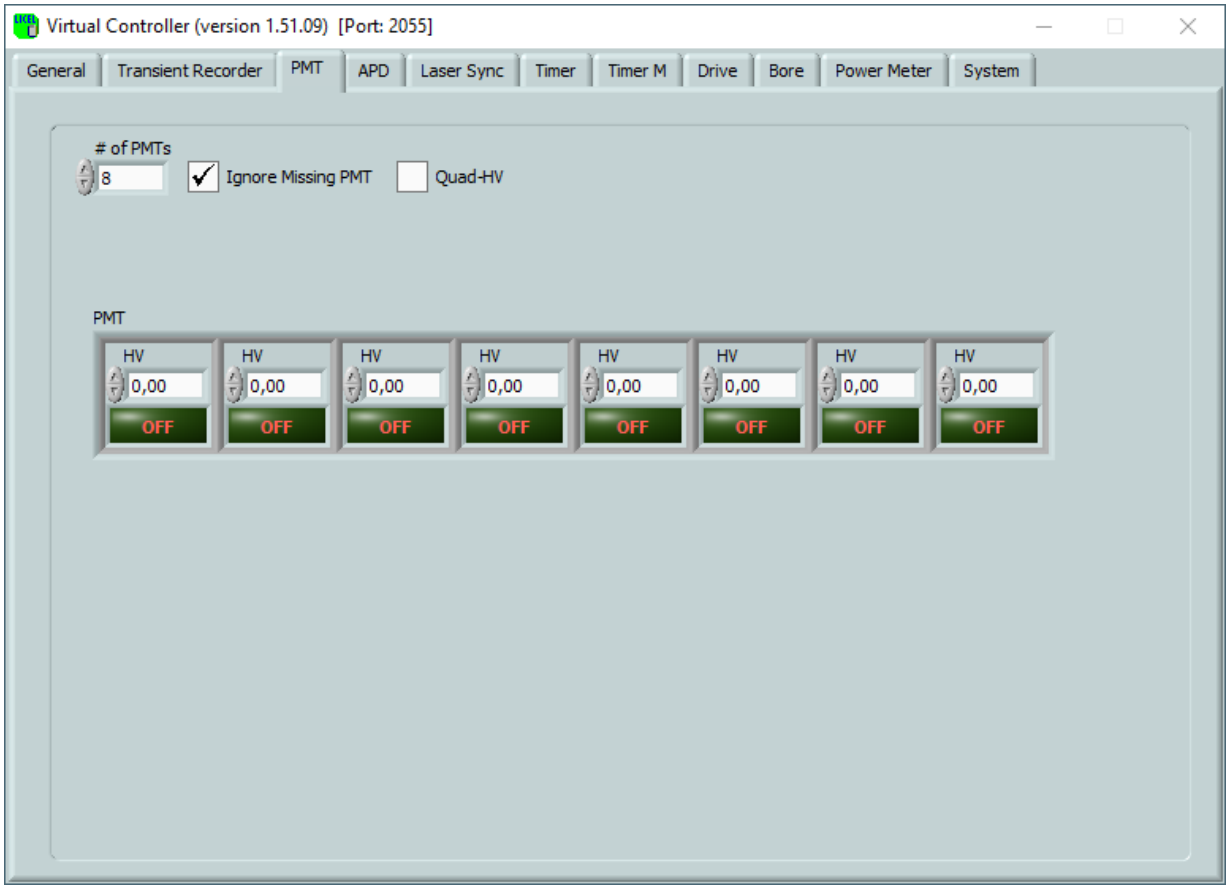
`Beam_Radius`	Radius of the assumed beam in mm
`Beam_Divergence`	Divergence of the assumed laser beam in rad
`Beam_Inclination_Parallel`	Assumed parallel beam inclination in rad
`Beam_Inclination_Perpendicular`	Assumed perpendicular beam inclination in rad
`Telescope_Diameter`	Telescope diameter in m
`Detector_Radius`	Assumed detector radius in mm
`Focal_Length`	Focal length of the receiving telescope in mm
`Axial_Separation`	Axial separation between the receiving telescope and outgoing laser beam
`Noise_Amplitude`	The noise amplitude allows for control of the amplitude of the random white noise added to the Lidar Data. The Lidar Data has a range of 0-2000, 0-10000, or 0-50000 depending upon which input range is taken
`Sqrt_Signal_Noise_Amplitude`	Square root of the signal noise amplitude for the simulation

LabVIEW Source Options

If you purchased the LabVIEW source code for the Licel Virtual Controller you are able to change the optical alignment while an acquisition is active. After having read the initialization file the values are written to global variables. These global variables are accessed at the time the simulated signal is generated. You may change them in `Virtual Globals Overlap Globals.glb.vi` in the library `Virtual Globals.llb`. The simulated signal will immediately follow this ”change” of the optical alignment.
The variable `Virtual Globals Laser Rep Rate.glb.vi` controls the laser repetition rate (located in `Virtual Globals.llb`, as well).
Furthermore you will be able to replace the model used for the simulated backscatter signal. You can substitute the code to generate the backscatter signal with your own model in the vis `Lidar-Sim Simulated Lidar with Noise.vi` and `Lidar-Sim interpolate sigma test.vi`. Both vis are located in the library `lidar-sim.llb`.

8.6 PMT

The controls and information located on this tab page correspond to the PMT capability. When the photomultilpier high voltage is changed you will find the corresponding information here.



The following information is available:

# of PMTs number of simulated PMTs.

Ignore Missing PMT a checkmark will make the Virtual Controller to send valid replies even if a PMT does not exist. This simulates the behavior of older remote controllers. Recent controllers will send an error message when a PMT does not exist (no checkmark).

Quad-HV a checkmark will make the Virtual Controller to simulate the support of PMT labels as in Quad-HV remote controllers. If checked the additional control *Labels* will be visible. The labels can be set and read using the `PMTDESCR` and `PMTDESCR?` commands.

PMT An array containing the status information of the PMTs. Each element represents a PMT with the following detailed information:

HV Shows the value of the high voltage power supply for the PMT.

ON/OFF switch Shows whether or not the HV is being applied to the PMT.

8.7 APD

The controls and information located on this tab page correspond to the APD capability. When the APD parameters are changed you will find the corresponding information here.

The following information is available:

# of APDs Number of simulated APDs.

Ignore Missing APD a checkmark will make the Virtual Controller to send valid replies even if an APD does not exist. This simulates the behavior of older remote controllers. Recent controllers will send an error message when a APD does not exist (no checkmark).

APD An array containing the status information of the APDs. Each element represents an APD with the following detailed information:

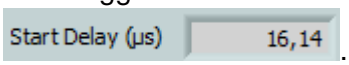
HV	Shows the value of the high voltage power supply for the APD.
ON/OFF switch	Shows whether or not the HV is being applied to the APD.
Temp Regulation	Shows whether or not the APD is being passively or actively cooled.
Temp in Range	Controls whether or not the temperature of the apd is in or out of range. This value may be changed here to test the reaction of a client application.

8.8 Timer

The controls and information located on this tab page correspond to the TIMER, TIMER1,and TIMER2 capabilities. They are changed in response to the `TRIGGERMODE` and `TRIGGERTIME` commands.

The following information is available:

Internal/External	This switch indicates whether the internal trigger of the Licel Trigger Module or an external trigger will be used as master trigger.
Lamp	Indicates whether or not the trigger output for the laser lamp is enabled.
Transient Recorder	Indicates whether or not the transient recorder (acquisition) will be triggered.
Q-Switch	Indicates whether or not the Q-switch output will be triggered.
Gating	Indicates whether or not a gate pulse will be generated.

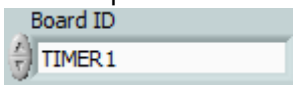
Repetition Rate Frequency in Hz of the internally generated trigger pulses. Here, the repetition rate is displayed in the case that the internal trigger is used as the master trigger. If the external trigger is used the set start delay is shown instead at the same position: 

.

Estimated Period (ms) Estimated periodin milliseconds.

Timing Diagram Timing Diagram displaying the trigger pulses. Please refer to the explanations in the TCP/IP Command List.

Board ID This value reflects the latest accessed sub board of a Licel Trigger Module. Allowed values reflect the capabilities TIMER, TIMER1 . . . TIMER4. The control can be changed to inspect the latest values set at a certain sub bourd.

The example 

indicates an access to the simulated sub-board corresponding to the capability TIMER1.

Trigger Slave 

This checkbox relects the feature of the active sub board set by *Board ID*: if checked it is estimated that the active board always gets an external trigger from another sub board. The simulation will not allow an internal master trigger.

Triggers OFF 

This switch can be used to switch all triggers OFF for debugging.

In this example the controller internally generates a trigger (**Master Trigger** dark) with the repetition rate 160.09 Hz. The controller generates lamp, q-switch, and gating pulses.

8.9 Bore

The controls and information located on the tab page *Bore* correspond to the BORE capability to simulate Licel's Bore Site Aligment Detector. This is described in the corresponding manual available at https://www.licel.com/manuals/BoreManual.pdf.

8.10 Power Meter

The controls and information located on this tab page correspond to the POW capability. They are changed in response to the commands to control a Licel Power Meter. Please refer to the Power Meter Control section 6.4.

8.11 TimerM and Drive

The controls and information located on these tab pages correspond to the TIMERM and DRIVE capabilities. They are changed in response to the commands to control a Licel Polarotor. Please refer to the corresponding manual for the Licel Polarotor found at https://www.licel.com/manuals/polarotor.pdf.

Chapter 9

Appendices

9.1 TCP/IP Communication

9.1.1 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. In this description the abbreviations TR, PMT, and APD are used to denote a Licel transient recorder, a Licel photomultiplier module, or a Licel avalanche photodiode, respectively. <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 Licel Ethernet Controller you ordered.

Short	Long
	ACCESS
	ALIGNDATA
	ALIGNSIGN
	ALIGNTIME
APD?	APDSTAT?
APDT	APDTEMPERATURE
APDG	APDGAIN
	BLOCK
	BOARDTEMP?
CAP?	CAP?
CLE	CLEAR
CONT	CONTINUE
DATA?	DATA?
DISC	DISCRIMINATOR
	DRIVEMODE
	DRIVERESET
	DRIVESPEED
	DRIVESPEED?
	DRIVESTATUS?
	FREQDIV
	FREQDIV?
HOST	
	HOSTNAME?

*IDN?	IDENTIFICAT?
	INTERNALTRIGA
	KILL
	LIMIT
	LOGON
MCL	MCLEAR
MCON	MCONTINUE
	MILLISEC?
MPUS	MPUSH
	MPUSHAB
	MPUSHBACK
	MSHOTS?
	MSHOTSAB?
MSTA	MSTART
MSTO	MSTOP
MWA	MWAIT
PASS	PASS
PMT?	PMTSTAT?
PMTG	PMTGAIN
PMTDESCR?	
PMTDESCR	
	POW
	PRETRIG
PUSH	PUSH
RANG	RANGE
SEL	SELECT
	SETMAXBINS
	SETMAXSHOTS
SING	SINGLE
	SHOTAB?
SLAV	SLAVE
STAR	START
STAT?	STATUS
STOP	STOP
TCPIP	TCPIP
	TEMP?
THR	TRESHOLD
	TRIGCYCLE
	TRIGGERMODE
	TRIGGERTIME
	TRIGGERTIMEM
	TRIGTRIGMINWIDTH
	TRIGOFFSET
	TRIGSCALE
	TRIGSLAVE
	TRTYPE?
	WHITELIST

ACCESS <LIMIT "Password" "Connection Password" | FREE "Password">

Switches the secure mode on or off.

If used with the keyword LIMIT the secure mode is switched on. The administrator password ("Password") and the password for client connections ("Connection Password") have to be transmitted together with the LIMIT keyword. Access to the controller is limited to clients operating from hosts specified with the WHITELIST command. After establishing his TCPIP connection a client must use the LOGON command to login in secure mode. The example

ACCESS LIMIT "Administrator" "Connected"

will start the secure mode with the Connection Password Connected (if the current controller password equals Administrator). In case of a non-correct controller password or bad command syntax the controller will return

ACCESS not accepted,

other wise the return value is

ACCESS Limited.

If the ACCESS command is used with the keyword FREE the secure mode is switched off. The administrator password ("Password") has to be transmitted together with the LIMIT keyword. The response of the controller after a successful ACCESS command is

ACCESS Unlimited.

Note that one has to establish a secure mode connection using the LOGON command (i.e. one has to know the Connection Password) before switching the secure mode off with the ACCESS command. The only other way to disable the secure mode is a hardware reset.

ALIGNDATA <START EVERY #shot SHOTS #cycle CYCLES>| <STOP>

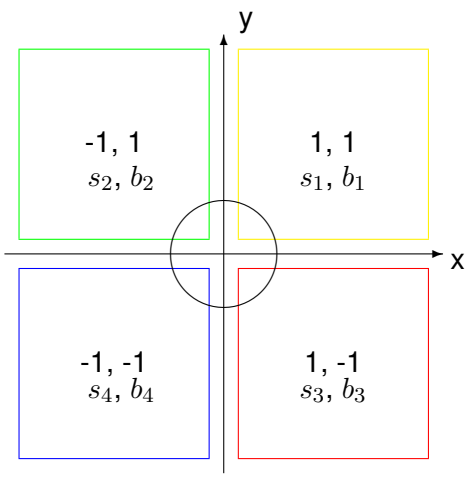
Starts or stops the controller to send bore site alignment data via the data push socket. After ALIGNDATA START the controller will average the involved channels over #shot shots and write the data to the data push socket in the following form:

Align Info: id s_1 s_2 s_3 s_4 b_1 b_2 b_3 b_4.

The reply ends with a <CRLF>.

where id is a counter which absolutely increases with each transmitted data set. The sign of the counter can be toggled via the ALIGNSIGN command. $s_{1..4}$ and $b_{1..4}$ are the averaged signal and background values for the involved channels and corresponding to the background and signal regions defined by the ALIGNTIME command, i.e. by Background Start and Background Stop, and Signal Start and Signal Stop. This step is repeated for #cycle times. If #cycle equals -1, the controller must explicitly be stopped by sending ALIGNDATA STOP to the controller. The indices of $s_{1..4}$ and $b_{1..4}$ correspond to the Licel Bore Sight Detector as seen in the figure below. The controller replies

ALIGNDATA START executed or ALIGNDATA STOP executed.



Sketch of the quadrants of the Licel Bore Sight Detector (cathode)

ALIGNSIGN

Sending this command toggles the counter sign in the data that is send when the bore system is acquiring data. The main purpose is to synchronize the bore data with the alignment moves. Once a movement is finished one could send ALIGNSIGN command and wait that the counter sign in the received data toggles to make sure that the data has been recorded after the last alignment move.
If the command is successful the controller replies:

```
ALIGNSIGN executed.
```

ALIGNTIME <Background Start><Background Stop><Signal Start><Signal Stop>

Sets the timing parameters for bore site alignment. The parameters correspond to the bins of the acquired data. The region between `Background Start` and `Background Stop` is assumed to correspond to the background, while `Signal Start` and `Signal Stop` define the signal region. The following restrictions apply:

$$\begin{aligned} 0 < &\ \texttt{Background Start} &&< 1024 \\ \texttt{Background Start} < &\ \texttt{Background Stop} &&< 1024 + \texttt{Background Start} \\ \texttt{Background Stop} < &\ \texttt{Signal Start} &&< 1024 + \texttt{Background Stop} \\ \texttt{Signal Start} < &\ \texttt{Signal Stop} &&< 1024 + \texttt{Signal Start} \end{aligned}$$

where `Background Start` and `Background Stop` may pairwise be interchanged with `Signal Start` and `Signal Stop`.
If the command is successful the controller replies:

```
ALIGNTIME executed.
```

An example for the command is

```
ALIGNTIME 400 850 1200 1600.
```

APDSTAT? <Device Number>

APD? <Device Number>

Returns the current status of the APD with the given `Device Number`. For example to get the status of APD number 3 send

```
APD? 3
```

to the controller. The reply is of the following form:

```
APD <Voltage> <HV control state> <temperature regulation>
<T in range?> <T control state>
```

with the values

```
Voltage                 HV voltage
HV control state        HV_local | HV_remote
temperature regulation  T_on | T_off
T in range?             T_in_range | T_out_of_range
T control state         T_local | T_remote.
```

`Voltage` is the gain voltage and indicates whether the power supply of the APD is switched on or off. The `HV control state` indicates whether the APD HV is being controlled locally (`HV_local`) or remotely (`HV_remote`). Valid answers for the `temperature regulation` are `T_on` and `T_off`. If the temperature is in range, then the `T in range?` value is `T_in_range`, otherwise `T_out_of_range` is returned. The `T control state` returns `T_local` or `T_remote`. An example of a reply is

```
APD 750.0 HV_local T_on T_in_range T_remote.
```

In this case the gain voltage is 750.0 volts, the APD HV is controlled locally, and the temperature is being regulated, is in range and remotely controlled. If the APD with the specified device number is not installed at recent APD controllers the reply contains

```
APD not available
```

Older controllers will not generate such an error message. Valid values for the device number are 0 – 3. `APDSTAT?` works in both remote and local control modes.
Please note that the long command `APDSTAT?` is not supported on all controllers, please use the short command `APD?` instead.

APDT <Device Number> <on|off>

APDTEMPERATURE <Device Number> <on|off>

Turns the temperature regulation for the APD specified by `<device number>` either `on` or `off`. For example to turn on the temperature regulation on the APD with device number 3, send

```
APDT 3 on
```

to the controller. The reply is

```
APDT executed
```

If the APD with the specified device number is not installed at recent APD controllers the reply contains

```
APD not available
```

Older controllers will not generate such an error message. Valid values for the device number are 0 – 3. The long form breaks the SCPI convention since it is longer than 12 characters.

APDG <Device Number> <HV Voltage>

APDGAIN <Device Number> <HV Voltage>

Sets the gain voltage for the specified APD to the given `<HV Voltage >` value. For example to set the gain of APD with device number 3 to 300 Volts, send

```
APDG 3 300
```

to the controller. A successful execution is indicated by the reply

```
APDG executed.
```

If the APD with the specified device number is not installed at recent APD controllers the reply contains

```
APD not available.
```

Older controllers will not generate such an error message. Valid values for the device number are 0 – 3.

BLOCK <Memory|OFF>

Block a trigger related to the acquisition at the specified `Memory` = `A`, `B`, `C`, or `D`. The typical use case is when the rack trigger A and B are driven but a certain channel should be active only when trigger A or B arrives. `BLOCK OFF` unblocks all memories.
To use the `BLOCK` command it must be supported by the Licel Ethernet Controller and the `HWCAP` returned with `TRTYPE?` command for the individual transient recorder must contain the bit `0x10`.
The controller's response is

```
BLOCK executed
```

or if the command fails

```
BLOCK failed error:  <error number >.
```

To block triggers at the memories B and C of the selected transient recorder, subsequently send

`BLOCK B` and

`BLOCK C` to the controller.

To unblock previously blocked triggers send
`BLOCK OFF.`

BOARDTEMP?

Get the FPGA temperature of the SELECTed transient recorder. If the command is supported by the Ethernet controller and the transient recorder supports reading the temperature the reply is
`BOARDTEMP <BoardTemp>`
where `BoardTemp` is the temperature in centigrade as a fractional number.

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:

`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
`BORE`	Boresight alignment system .

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.

CLEAR

CLE

Clears both memories (A and B) of the previously selected transient recorder, if the TR is in SLAVE mode. After sending this command, the controller replies with the string
`CLEAR executed.`
If this command is sent while PUSH or MPUSH mode is active, the reply is
`CLEAR ignored due to active PUSH mode.`
If the selected TR does not answer, the response will be:
`CLE failed for TR <TR#>, Can't write.`
`CLE failed for TR <device number >, <Can't clear Memory >`
indicates a memory access error to `Memory` (A or B).

CONTINUE

CONT

Continues data acquisition without clearing the memory of the selected transient recorder if the TR is in SLAVE mode. After sending this command the controller replies with the string
`CONTINUE executed.`
If this command is sent while PUSH or MPUSH mode is active, the reply is
`CONTINUE ignored due to active PUSH mode.`
The error message
`CONTINUE failed for TR <Device Number>, Can't write`

is sent if the transient recorder identified by `Device Number` is not responding.

DATA? <Device Number> <Number to Read> <Signal Type> <Memory>

Requests data from the transient recorder with the corresponding `Device Number` if the TR is in SLAVE mode.

- The `Number to Read` determines the number of bins to be read.
- The `Signal Type` may have one the following values:
  - `PC` photon counting
  - `PHM` photon counting upper memory
  - `MSW` most significant word for analog and PR transient recorders
  - `LSW` least significant word for analog and PR transient recorders
  - `PHM` photon counting and analog upper memory
  - `P2L` least significant word for photon counting squared data
  - `P2M` middle word for for photon counting squared data
  - `A2L` least significant word for analog squared data
  - `A2M` middle word for analog squared data
  - `A2H` highest word for analog squared data

  Squared data is supported by Licel's most recent transient recorders with Ethernet controller software newer than 2019-12-17.
- The `Memory` can be either `A`, `B`, `C`, or `D`, for memory A, memory B, memory C, or memory D, respectively. Memory C and D support is determined by reading the bits `0x02` and `0x04` of the `HWCAP` received with the TRTYPE? command.

```
DATA? 6 8000 PC B
```

which would return the first 8000 bins of the photon counting Memory B of transient recorder #6. The controller replies to the DATA? request by returning the data. As the transient recorder's data is an array of 16-bit numbers the returned number of bytes equals twice the number of requested bins. The 16-bit numbers are sent as little endian, i.e. the least significant byte of a number comes first followed by the most significant byte. If the command is sent while PUSH or MPUSH mode is active, the reply is

```
DATA? ignored due to active PUSH mode.
```

If `Device Number` is not in range the reply is

```
Device ID <Device Number> is currently not supported.
```

The error message

```
DATA failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #`Device Number` is not responding.

DISCRIMINATOR <Integer>

DISC <Integer>

Sets the discriminator level. Valid values for the discriminator are 0–63. To set the discriminator level to 16, send

```
DISCRIMINATOR 16
```

to the controller. The reply is

```
DISCRIMINATOR set to 16.
```

If the `Integer` value is out of range the reply is

```
DISCRIMINATOR value is out of range.
```

The error message

```
DISCRIMINATOR failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #`Device Number` is not responding.

DRIVEMODE <DriveMode><Count>

Sets the drive mode for a polarotor controller. `DriveMode` is a decimal number with the following settings encoded into the five least significant bits:

`00000`	0	`STOP`	no motion, stops motion
`01000`	8	`START`	free run, starts motion with the programmed `DRIVESPEED` setting
`11001`	25	`WHILE 1234 HIGH`	run the stepper motor with the programmed `DRIVESPEED` *as long as* the angular position detector is active
`11010`	26	`WHILE 1234 LOW`	run the stepper motor with the programmed `DRIVESPEED` *until* the angular position detector is active
`11011`	27	`WHILE SYNC HIGH`	run the stepper motor with the programmed `DRIVESPEED` *as long as* the synchronization detector is active
`11100`	28	`WHILE SYNC LOW`	run the stepper motor with the programmed `DRIVESPEED` *until* the synchronization detector is active
`11101`	29	`X STEPS`	move the stepper motor `Count` number of steps.

`Count` is used only for the mode `11101`. For the modes with bit 4 set, a small velocity (recommended: `320`) must be set using the `DRIVESPEED` command.
If the command is successfully executed the controller replies

```
DRIVEMODE executed [Steps: numSteps].
```

The current drive mode can be read with the DRIVESTATUS? command.
`Steps: numSteps` is returned by controllers since march 2020. `numSteps` is the number of moved steps since *submitting a `DRIVEMODE` command before the current `DRIVEMODE`* command. This number can be used to determine the number of steps for one revolution (always use the `DRIVESTATUS?` command to obtain the level of the synchronization detector):

1. Obtain the current level of the synchronization detector using the `DRIVESTATUS?` sommand.
2. If the synchronization detector is inactive send `DRIVEMODE 28 0` and wait until the synchronization detector is active.
3. Send `DRIVEMODE 27 0` to move the polarotor until the synchronization detector is inactive.
4. Send `DRIVEMODE 28 0` to move the polarotor until the synchronization detector is active.
5. Send `DRIVEMODE 27 0` to move the polarotor until the synchronization detector is inactive, remember the returned `numSteps` (the number of steps moved since the execution of 4).
6. Now, the polarotor has moved one roundtrip, you need the number of steps since the execution of 5)
7. You will get this number by sending `DRIVEMODE 0 0`, take the returned `numSteps` and add them to the number returned with the execution of 5.

DRIVERESET

Reset the polarotor controller. This command should be used after the stepper motor of the polarotor has been stopped by `DRIVEMODE 0 0` while it was free running.
If the command is successfully executed the controller replies

```
DRIVERESET executed.
```

DRIVESPEED <Speed>

Sets the velocity for the polarotor's stepper motor in counts per second.
If the command is successfully executed the controller replies

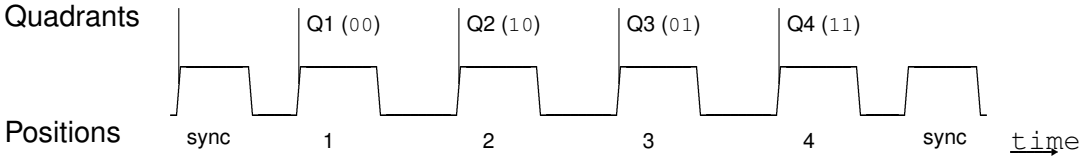
DRIVESPEED executed.
The current stepper motor velocity can be read with the DRIVESPEED? command.

DRIVESPEED?

Reads the programmed stepper motor velocity at the polarotor.
If the command is successfully executed the controller replies
DRIVESPEED Speed where Speed is the velocity of the stepper motor in counts per second.
The stepper motor velocity can be set with the DRIVESPEED command.

DRIVESTATUS?

Returns the status of the polarotor.
If the command is successfully executed the controller replies
DRIVESTATUS Quadrant SyncLevel 1234Level DriveMode.
The 2 least significant bits of Quadrant contain the quadrant information. 00 is the quadrant beginning with the first angular position following the synchronization position. The sequence of quadrants is encoded by 00 (1) – 10 (2) – 01 (3) – (11) (4).
SyncLevel indicates whether the synchronization position is active (1) or not (0).
1234Level indicates whether or not an angular position is active (1) or not (0).



DriveMode is the current drive mode, a number with the following settings encoded into the five least significant bits:

00000	0	STOP	stopped
01000	8	START	free running
11001	25	WHILE 1234 HIGH	run the stepper motor *as long as* the angular position detector is active
11010	26	WHILE 1234 LOW	run the stepper motor *until* the angular position detector is active
11011	27	WHILE SYNC HIGH	run the stepper motor *as long as* the synchronization detector is active
11100	28	WHILE SYNC LOW	run the stepper motor *until* the synchronization detector is active
11101	29	X STEPS	move the stepper motor Count number of steps.

FREQDIV <freqDividerExponent>

The command will only work if the hardware capabilities of the transient recorder support this. This is indicated by bit 7 in the HWCAP field of the TRTYPE? command. Allowed values for the freqDividerExponent are 0 ...7. This command will set the frequency divider of the SELECTed transient recorder: it changes the sampling rate before the summation. The resulting range resolution is obtained by rangeResolution = primaryBinWidth(1 << freqDividerExponent). Therefore, with freqDivExponent = 0 and a primary bin width of 3.75 m you will get 3.75 m range resolution. With a freqDividerExponent of 3 you will get 30 m range resolution.
The frequency divider works before the summation, so that the number of bins in the trace is not influenced by it. This is different from the software binning that is done in acquis.
If successfully executed the command returns
FREQDIV <freqDividerExponent> 0

or if the command fails

```
FREQDIV failed error:  <error number >
```

FREQDIV?

The command will only work if the hardware capabilities of the transient recorder support this. This is indicated by bit 7 in the HWCAP field of the TRTYPE? command. This command will request the current frequency divider of the SELECTed transient recorder. Possible values are 0 ...7. The resulting range resolution is obtained by rangeResolution = primaryBinWidth(1 << freqDividerExponent).
If successfully executed the command returns

```
FREQDIV <freqDividerExponent> 0
```

or if the command fails

```
FREQDIV? failed error:  <error number >
```

HOST <"NewHostName"> <"Password">

Set the host name of the Licel Ethernet Controller to NewHostName. Password is the administrator password (default: *Administrator*, changeable using the PASSWORD command).
Example:

```
HOST "MyNewHostName" "Administrator"
```

gives the reply

```
HOSTNAME changed to MyNewHostName.
```

If a command parameter is not correct the reply is

```
HOST failed.
```

HOSTNAME?

Returns the current host name set by the HOST command:

```
HOSTNAME?
```

gives the reply

```
LICEL
```

if the host name has been set to LICEL.

IDENTIFICAT?

*IDN?

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

```
Licel Control firmware rev.  22.03.2024 ARM_Ethernet 00:03:F4:FF:FF:FF
```

INTERNALTRIGA

Causes an internal trigger and can be used for software testing purposes.
The command is acknowledged by

```
INTERNALTRIGA executed
```

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.

LIMIT <64K|4K >

Switches the maximum number of shots acquired by the transient recorder between 65536 and 4096 shots, where the initial clearing shots are also included. The command will work for 16 bit transient recorders starting from 2011. Starting from 2014 the `SETMAXSHOTS` allows arbitrary shot numbers for newer transient recorders.
If the transient recorder supports the command, the response will be:

```
LIMIT executed
```

otherwise

```
LIMIT not supported for TR <TRNumber >
```

LOGON <"Encrypted Hexcode">

Is used to log in while the secure mode is active. Directly after establishing the TCP/IP connection with the controller the latter will send two 4 byte unsigned integer numbers in a hex-encoded string. The client has to decode these numbers from the hexadecimal string and use them to encrypt the connection password set by the `ACCESS` command using the Blowfish encryption algorithm. The resulting two 4 byte unsigned integer numbers have to be converted to a hexadecimal string and sent to the controller with the `LOGON` command. While secure mode is active the controller will close the TCP/IP connection without any comment if it does not receive the correct code within 20 seconds.

MCLEAR

MCL

Clears all memories of the SELECTed transient recorders, if the TR is in SLAVE mode. The answer is

```
MCLEAR executed.
```

If this command is sent while PUSH or MPUSH mode is active, the reply is

```
MCLEAR ignored due to active PUSH mode.
```

If a selected TR does not answer, the response will be:

```
MCLEAR failed for TR <TR#>, Can't write.
MCLEAR failed for TR <device number >, <Can't clear Memory >
```

indicates a memory access error to `Memory` (A or B).

MCONTINUE

MCON

Restarts the SELECTed transient recorders without clearing the memories, if the TRs are in SLAVE mode. The reply is

```
MCONTINUE executed.
```

If this command is sent while PUSH or MPUSH mode is active, the reply is

```
MCONTINUE ignored due to active PUSH mode.
```

The error message

```
MCONTINUE failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #`Device Number` is not responding.

MILLISEC?

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.

MPUSH <Shots> <Device Number> <Number to Read> <Signal Type> <Memory> [<Device Number> <Number to Read> <Signal Type> <Memory>[...]]

MPUS <Shots> <Device Number> <Number to Read> <Signal Type> <Memory> [<Device Number> <Number to Read> <Signal Type> <Memory>[...]]

Causes the controller to enter a state where data of `Signal Type Memory` is directly pushed from the transient recorder(s) `Device Number` to the computer. The `Signal Type` can be either `PC`, `MSW`, or `LSW` for photon counting, analog MSW, or analog LSW, respectively. The `Memory` can be either `A` or `B`, for memory A or memory B, respectively. The transient recorders acquire n shots, n is given by `Shots` and in opposition to the PUSH Mode there is only the internal shot limit of the TR (4094 by default).

After having acquired the requested number of `Shots` the controller reads `Number to Read` bins from the corresponding `Memory` and `Signal Type` from each transient recorder with the given `Device Number`. Additionally, the controller reads the `Background Bins` defined by the MPUSH-BACK command. Signal and Background are combined and sent to the computer. As the transient recorder's data is an array of 16-bit numbers the returned number of bytes equals twice the number of requested bins. The data have a header consisting of 2 marker bytes `0xFF` and a 4 byte integer with a timestamp defined as the milliseconds since the start of the controller. The data sets for each transient recorder are preceded by the number of shots as a 16-bit number. Note that the number of shots has an offset of 2 caused by the clear shots. The length of each device-specific data set has to be known by the acquiring computer.

Then, the transient recorders automatically continue to collect data sets for pushing them to the computer. The SLAVE command stops the MPUSH command. The example

```
MPUSH 5 1 8000 PC B 4 6000 LSW A
```

would cause the data from the transient recorders 1 and 4 to be pushed to the data acquisition computer after recording 5 shots. From device 1, 8000 bins of the photon counting Memory B will be sent. From device 4, 6000 bins of analog LSW memory A will be sent. Having sent the data the TRs will automatically be restarted by the controller and the next set of data will be acquired and sent. The reply is

```
MPUSH executed.
```

If the command syntax is not correct the controller replies

```
MPUSH syntax is wrong,
```
if the PUSH mode is active the controller will return
```
MPUSH ignored due to active PUSH mode.
```
If the number of shots is not in range the controller returns
```
Illegal Push shot number.
```
The mpush mode data will flow over the push socket and show a header field followed by the 16-bit binary data.

```
0xFF
0xFF
<4 byte timestamp> (milliseconds, unsigned)
<2 byte shot number data(0)> (unsigned)
 <data(0) (16bit wide) ....>
<2 byte shot number data(1)> (unsigned)
 <data(1) (16bit wide) ....>
...
```

All header fields and the 16-bit numbers are sent as little endian, i.e. the least significant byte of a number comes first. Please note that older Ethernet controllers with a Coldfire processor will return (only) the timestamp as big endian. You may recognize such a controller by determining whether the response to the `*IDN?` command contains `ColdFireEthernet`.
The mpush mode is stopped by sending the `SLAVE` command.

MPUSHAB

starts a single shot acquisition over multiple transient recorders. For each shot the trigger source will be recorded (trigger A or trigger B).
The mode is activated by
```
MPUSHAB <tr> <bins> <signal type> [<tr> <bins> <signal type>[...]]
```
The command is acknowledged by
```
MPUSHAB executed
```
or rejected with
```
MPUSHAB syntax is wrong
```
or if no push socket has been previously opened with
```
Push socket not ready for transmission.
```
The push mode data will flow over the push socket and show a header field

```
0xFF
0xFF
<8 byte timestamp> (double)
<2 byte shot number> (unsigned)
<1 byte memory> Mem A|B
<...data (16bit wide) ....>
shot number(2 byte)
Mem A|B (1 byte)
<...data (16bit wide) ....>
```

The push mode is stopped by sending the `SLAVE` command.

MPUSHBACK <Skip> <Background Bins>

Defines the background for the data in the MPUSH mode. `Skip` is the number of bins to skip between the signal bins (`Number to Read`) and the first background bin. `Background Bins` is the number

of background bins the controller will read and attach to the `Number to Read` signal bins in the MPUSH mode.
if the command is correctly executed the controller replies

```
MPUSHBACK executed.
```

MSHOTS? [numMemories]

Returns the acquired shots and summation memories of the last acquisitions of all `SELECT`ed transient recorders.
The reply is

```
MSHOTS [shots(0) memory(0) ... shots(n-1) memory(n-1)]
```

where `n` transient recorders have been `SELECT`ed before. `shotsi` contains the decimal number of shots of the i^{th} selected transient recorder and `memory(i)` is either 0 (memory A), 1 (memory B), 2 (memory C), or 3 (memory D).

The optional argument `numMemories` will request the information about the the shots that have been acquired into the first `numMemories` memories. If 2 is choosen for `numMemories` the command will return the shots at the memories A and B. Then the response will be

```
MSHOTS [shotsA(0) shotsB(0) ... shotsA(n-1) shotsB(n-1)]
```

MSHOTSAB?

Returns the shots in each of the memories of all `SELECT`ed transient recorders.
The reply is `MSHOTS [shotsA(0) shotsB(0) ... shotsA(n-1) shotsB(n-1)]`
where `n` transient recorders have been `SELECT`ed before. `shotsAi` contains the decimal number of shots of the i^{th} selected transient recorder that have been acquired to memory A and `shotsBi` those that have been aquired to memory B.
The return values will be only valid if the hardware capabilities of the transient recorder support this. This is indicated by the lowest bit in the HWCAP field of the `TRTYPE?` command.

MSTART

MSTA

starts the SELECTed multiple TRs, if the TRs are in SLAVE mode. As an example

```
MSTART
```

would start selected Devices. The reply is

```
MSTART executed.
```

If this command is sent while PUSH or MPUSH mode is active, the reply is

```
MSTART ignored due to active PUSH mode.
```

The error message

```
MSTART failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #`Device Number` is not responding,

```
MSTART failed for TR <device number >, <Can't clear Memory >
```

indicates a memory access error to `Memory` (A or B).

MSTOP

MSTO

Stops the SELECTed multiple TRs, if the TRs are in SLAVE mode.

```
MSTOP
```

will stop the currently selected devices. The reply is

```
MSTOP executed.
```

If this command is sent while PUSH or MPUSH mode is active, the reply is
`MSTOP ignored due to active PUSH mode.`
The error message
`MSTOP failed for TR <Device Number>, Can't write`
is sent if the transient recorder #`Device Number` is not responding.

MWAIT <Timeout in ms>

MWA <Timeout in ms>

Waits until all SELECTed TRs stop or until the timeout time is exceeded. The range for `Timeout in ms` is between 0 and 400. If all TRs are ready within the timeout limit, the reply is
`MWAIT executed.`
If `Timeout in ms` is not in range the controller replies
`MWAIT failed delay: <Timeout in ms> should be between 0 and 400ms.`
If a selected TR does not answer, the response will be:
`MWAIT failed for TR <TR#>, Can't write.`

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.`

PMT? <Device Number>

PMTSTAT? <Device Number>

Returns the measured HV at the PMT with the specified device number. The reply parameters are `<HV value in Volts>` followed by a `on remote`.
`PMT? 5`
to the controller. An example of a reply is
`PMT 970 on remote`
which indicates that the PMT's high voltage is 970 Volts. If the PMT with the specified device number is not installed at recent PMR controllers the reply contains
`PMT not available`
Older controllers will not generate such an error message. Valid values for the device number are 0 – 7.
Please note that the long command `PMTSTAT?` is not supported on all controllers, please use the short command `PMT?` instead.
If you use the `PMT?` command for a PMT which is not installed, a HV value of ≈ 356 V is returned. You may use this fact to find out all installed PMTs:

1. Set the *manual*|*off*|*remote* switches of all mounted PMT cassettes to *remote*

2. Set the high voltages of all possible PMTs to 0: `PMTG n 0.0`, where `n` runs from `0` to `7`.
3. Request all high voltages: `PMT? n`, where `n` runs from `0` to `7`. When a PMT reply contains ≈ 356 V the corresponding cassette/PMT is not installed.

PMTG <Device Number> <Voltage>

PMTGAIN <Device Number> <Voltage>

This command sets the gain voltage applied to the dynodes of the PMT with the specified device number. For example

```
PMTG 3 980
```

will set the gain voltage to 980 volts. The reply is

```
PMTG executed.
```

If the PMT with the specified device number is not installed at recent PMT controllers the reply contains

```
PMT not available.
```

Older controllers will not generate such an error message. Valid values for the device number are 0 – 7.

PMTDESCR? <Device Number>

Request the PMT label (description) of the detector specified by `Device Number`. This command is supported by Quad-HV remote controllers, only. Valid values for the device number are 0 – 7.
The reply is

```
PMTDESCR <PMT label>
```

where the `PMT label` is the label used at the corresponding display of the Quad-HV remote controller's front. If the PMT with the specified device number is not installed the reply contains

```
PMT not available.
```

PMTDESCR <Device Number> <PMT label>

Set the PMT label (description) of the detector specified by `Device Number`. This command is supported by Quad-HV remote controllers, only. The `PMT label` is used at the corresponding display of the Quad-HV remote controller's front. The `PMT label` should not be longer than 10 characters. Valid values for the device number are 0 – 7.
The reply is

```
PMTDESCR executed.
```

If the PMT with the specified device number is not installed the reply contains

```
PMT not available.
```

POW <Command> [Number]

Submit a command to a Power Meter Controller. The usage of `Number` depends on the `Command`. The reply is

`POW <Command> executed` for all values of `Command` excepted `TRACE` (see below).

The following values for `Command` are supported:

`CHANNEL` Selects the ADC channel for the data acquisition. The ADC channel `Number` can be either be 0 (photodiode) or 2 (power meter head).
example: `POW CHANNEL 0` selects the ADC channel 0. The controller response is:
`POW CHANNEL executed.`

`NUMTRIG?` Request the number of detector inputs (number of triggers) from the Power Meter Controller. The controller response is
`POW NUMTRIG <numTriggers>` where $1 \leq$ `numTriggers` ≤ 3.
The `NUMTRIG?` option is supported by recent Power Meter Controllers. Older controllers will return `POW NUMTRIG? unknown command` which has to be interpreted as a single detector input (trigger).

`START` (`Number` is not used) Activates the data acquisition and data transmission over the previously opened push socket. The controller response is:
`POW START executed`.
For every received trigger one ASCII line will be sent with the following format
`<millisecondsSinceControllerStart> <pulseAmplitude><CRLF>`
Recent Power Meter Controllers (supporting the `NUMTRIG?` option) will add the `triggerIndex` after the `pulseAmplitude` ($0 \leq$ `triggerIndex` $<$ `numTriggers`).

`STOP` (`Number` is not used) Deactivates the data acquisition and stops the data transmission over the push socket. The controller response is:
`POW STOP executed`.

`TRACE` (`Number` is not used) Starts a single pulse acquisition and returns one pulse in the following ASCII format:
`<Number of points:N>` $<Y_0>$ $<Y_1>$ ... $<Y_{N-1}>$`<CRLF>`.

PRETRIG <1|0>

The command will only work if the hardware capabilities of the transient recorder support this. This is indicated by bit 3 in the HWCAP field of the `TRTYPE?` command. The pretrigger is 1/16 of the hardware tracelength. 1 enables the pretrig and 0 disables it. The startup state for a transient recorder is disabled. The command can be used only if the `HWCAP` contains the bits `0xF9`.
The command returns if successfully executed

```
PRETRIG executed
```

or if the command fails

```
PRETRIG failed error:  <error number >
```

PUSH <Shots> <Number to Read> <Signal Type> <Memory>

Causes the controller to enter a state where data from `Signal Type Memory` is directly pushed from the SELECTed transient recorder to the computer. The `Signal Type` can either be `PC`, `MSW`, or `LSW`, for photon counting, analog MSW (default), or analog LSW, respectively. The `Memory` can either be `A` (memory A) or `B` (memory B). The example

```
PUSH 3 8000 PC B
```

would return the first 8000 bins of the photon counting Memory B after 3 shots have been acquired. The controller will start an acquisition of n `Shots`. n is limited to a maximum value of 14. After having acquired the requested number of `Shots` the controller reads `Number to Read` bins from the corresponding `Memory` and `Signal Type` from the transient recorder and sends them to the computer. As the transient recorder's data is an array of 16-bit numbers the returned number of bytes equals twice the number of requested bins. These data have a header consisting of 2 marker bytes `0xFF` followed by the number of shots as a 16-bit number. The 16-bit shot number and the 16-bit data values are sent as little endian, i.e. the least significant byte of a number comes first followed by the most significant byte. Note that the number of shots has an offset of 2 caused by the clear shots. Then, the controller forces the TR to collect the next data for pushing it to the computer. The SLAVE command stops the PUSH command. The reply is

```
PUSH executed.
```

if the MPUSH mode is active the controller will return

```
PUSH ignored due to active MPUSH mode.
```

If data from more than one transient recorder or more than the LSW should be pushed to the acquisition computer the MPUSH command should be used.
If the number of shots is not in range the controller returns

```
Illegal Push shot number.
```

RANGE <0|1|2>

RANG <0|1|2>

Sets the input range to either -500mV (0), -100mV (1), or -20mV (2). The command

```
RANGE 0
```

sets the input range to -500mV. The TR replies with

```
RANGE set to -500mV.
```

If an illegal value for the range is submitted to the controller the reply is

```
Illegal Range Value.
```

The error message

```
RANGE failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #`Device Number` is not responding.

SELECT <Device Number List>

SEL <Device Number List>

Selects or unselects the active transient recorders. The parameter `<Device Number List>` is a comma-separated list of transient recorder numbers or -1 to unselect all selected transient recorders. For example to activate transient recorder #8, send

```
SELECT 8
```

If a TR with the given device number is available the answer by the controller is

```
SELECT 8 executed
```

To select more than one TR, separate the transient recorder numbers with a comma. For example, to select the transient recorders 1, 3, 8, and 12, send

```
SELECT 1, 3, 8, 12
```

Note that the separator is a comma and the empty spaces between the TRs will be ignored. Thus,

```
SELECT 1,3,8,12
```

is equivalent to the previous command. The answer by the controller is

```
SELECT 1, 3, 8, 12 executed
```

if any device number is out of range, the controller does not execute the command while replying

```
Device ID %d is currently not supported,
```

where `%d` is the first illegal device number. To unselect the active transient recorders send

```
SELECT -1
```

to the controller, the reply is

```
SELECT executed.
```

SETMAXBINS <numMaxBins>

Sets the tracelength if the memory configuraton switch 5 is in the ON Position. A user defined tracelength allows a better usage of the acquisition time for high repetition rate systems.

```
SETMAXBINS executed
```

if the command fails

```
SETMAXBINS failed error:  <error number >
```

SETMAXSHOTS <numMaxShots>

Sets the maximum number of shots that the TR should acquire. Per default this number is 4096. The command allows a determined number of shots other than 4096. The functionality is similiar to the LIMIT command but it allows a arbitrary number between 2 and 64K. The command returns if successfully executed

```
SETMAXSHOTS executed
```

or if the command fails

```
SETMAXSHOTS failed error:  <error number >
```

SINGLE

SING

Clears the TR memory and causes the transient recorder to take a single shot. The reply is

```
SINGLE executed
```

If this command is sent while PUSH or MPUSH mode is active, the reply is

```
SINGLE ignored due to active PUSH mode.
```

An access error at a transient recorder with the device number <device number> is indicated by

```
SINGLE failed for TR <device number>, Can't write.
```

SHOTAB?

Returns the shots in each of the memories of the previously SELECTed transient recorder.
The reply is

```
SHOTAB shotsA shotsB
```

shotsA contains the decimal number of shots of the selected transient recorder that have been acquired to memory A and shotsB those that have been aquired to memory B.
The return values will be only valid if the hardware capabilities of the transient recorder support this. This is indicated by the lowest bit in the HWCAP field of the TRTYPE? command.

SLAVE

SLAV

Ends the PUSH or MPUSH mode. The reply is

```
SLAVE executed.
```

START

STAR

Clears the memories and starts the data acquisition of the selected transient recorder, if the TR is in Slave mode. After sending this command, the controller replies with

```
START executed.
```

If this command is sent while PUSH or MPUSH mode is active, the reply is

```
START ignored due to active PUSH mode.
```

The error message

```
START failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #Device Number is not responding,

```
START failed for TR <device number >, <Can't clear Memory >
```

indicates a memory access error to Memory (A or B).

STATUS?

STAT?

Returns the current status of the SELECTed transient recorder. If more than one TR is selected, the Status of the selected TR with the lowest device number is returned (e.g. when 5, 7, 12, 14 are selected, then STAT? returns the status of TR #5). The values returned are the shotnumber, the acquisition state, the recording state, and the transient recorder's summation memory of the last acquisition. The shotnumber is returned as an integer. The acquisition state can be either armed or disabled, hence the string Armed is returned if the TR is armed, otherwise an empty string is returned. Whether or not the TR recorder is collecting data is shown by the recording state. If the TR is storing data in its memory the string Acquiring is returned or an empty string. The summation memory can either be Memory A or Memory B. If Memory B has been used for the last acquisition MemB, otherwise an empty string is returned. An example reply would be

```
Shots 8032 Armed Acquiring
```

indicating that the TR has acquired 8032 shots, is armed and currently accumulating data. Another example is

```
Shots 8032 Armed,
```

here, TR has acquired 8032 shots of data, is armed and is not storing data.
The error message

```
STAT? failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #Device Number is not responding,

STOP

Stops the data acquisition of the selected transient recorder, if the TR is in Slave mode. After sending this command, the controller replies with the string

```
STOP executed.
```

If this command is sent while PUSH or MPUSH mode is active, the reply is

```
STOP ignored due to active PUSH mode.
```

The error message

```
STOP failed for TR <Device Number>, Can't write
```

is sent if the transient recorder #Device Number is not responding.

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 when it is in PUSH or MPUSH mode. 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.

TEMP?

Get the FPGA temperature of the SELECTed transient recorder. If the command is supported by the Ethernet controller and the transient recorder supports reading the temperature the reply is
```
TEMP <FPGA_Temp>
```
where `FPGA_Temp` is the temperature in centigrade as a fractional number.

THRESHOLD <0|1>

THR <0|1>

Sets the damping state to either on or off. If a value of 1 is sent then damping is turned on. If a value of 0 is sent, the damping is turned off. To turn Damping on, send
```
THRESHOLD 1,
```
to turn the damping back off, send
```
THRESHOLD 0.
```
The controller replies with either
```
THRESHOLD executed :  Damping on
```
or
```
THRESHOLD executed :  Damping off.
```
The error message
```
THRESHOLD failed for TR <Device Number>, Can't write
```
is sent if the transient recorder #`Device Number` is not responding.

TRIGGERMODE[BoardID] <mode >

Enable/Disable the trigger in and outputs at Trigger and Polarotor controllers. `BoardID` is a board identifier which is used only if the Licel Trigger Module is equipped with more than 1 timing unit. Then `BoardID` is in the range 1 ...number of timing sub-boards. `BoardID` is not used for the polarotor timing (but used for additional timing sub boards) in a Polarotor Controller.
Mode is a bitfield where for every set bit the corresponding output is enabled:

	Trigger Controller	Polarotor Controller
0x01	Laser Lamp trigger	Laser Lamp trigger
0x02	Pretrigger (Acquisition)	Acquisition +
0x04	Q-Switch	Q-Switch
0x08	Gating	Acquisition ||
0x10	External Trigger	External Trigger

.

If the External trigger bit is set an external trigger will be accepted, if not the internal trigger will be used. The internal trigger will be controlled via the RepetitionRate in the TRIGGERTIME command. If successful the controller will return:

TRIGGERMODE executed

otherwise the returned string is

TRIGGERMODE failed.

If the parameter is out of the range (not a byte) the reply is:

TRIGGERMODE: invalid parameter.

TRIGGERTIME[BoardID] <RepetitionRate|StartDelay> <Pretrigger> <PretriggerLength> <QSwitch> <QswitchLength> <TriggerPeriod>

Set the timing parameter in ns at a Trigger Controller. BoardID is a board identifier which is used only if the Licel Trigger Module is equipped with more than 1 timing unit. Then BoardID is in the range 1 . . . number of timing sub-boards.

RepetitionRate|StartDelay	in internal mode delay between two pulses in ns, in external mode start delay between external trigger and lamp in ns. The StartDelay is supported at controlers newer than 2007-06
Pretrigger	delay between lamp and pretrigger (acquisition) in ns
PretriggerLength	length in ns of the pretrigger (acquisition) pulse
QSwitch	delay between pretrigger (acquisition) start and Q-Switch start in ns
QswitchLength	length in ns of the Q-Switch pulse.
TriggerPeriod	estimated time in milliseconds between subsequent laser triggers .

If successful the controller will return:

TRIGGERTIME executed,

in the case that the parameters cannot be interpreted the reply is

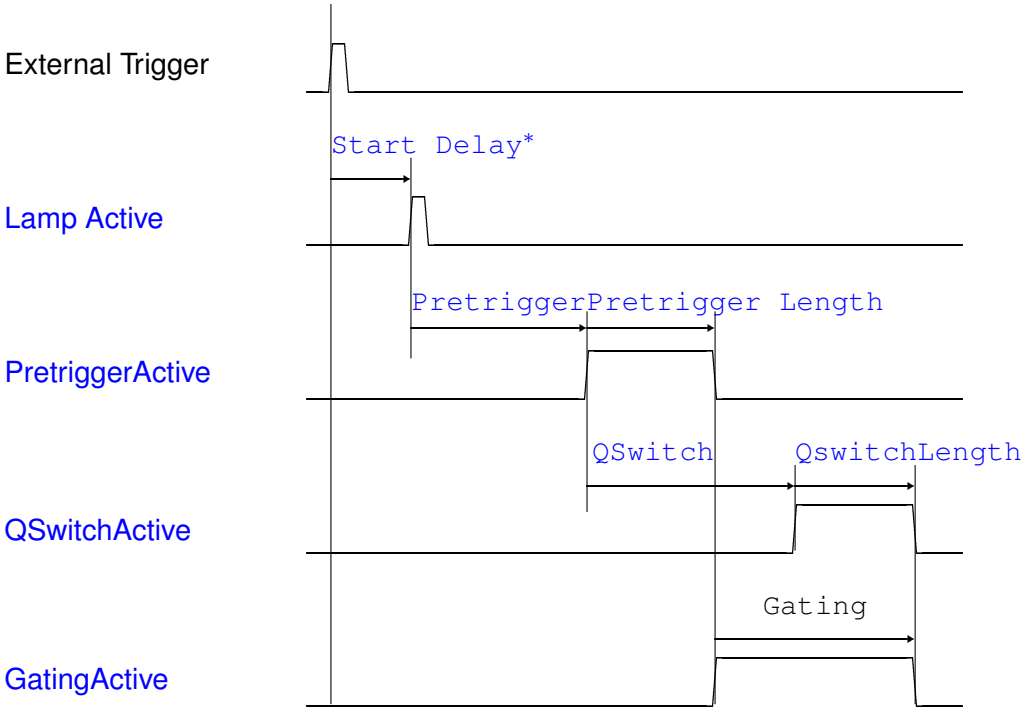
TRIGGERTIME: incorrect or invalid parameters.

The Gating pulse will be high from the end of the Pretrigger pulse till the end of the Q-Switch Pulse. The duration is

Gate = QSwitch + QswitchLength - PretriggerLength.

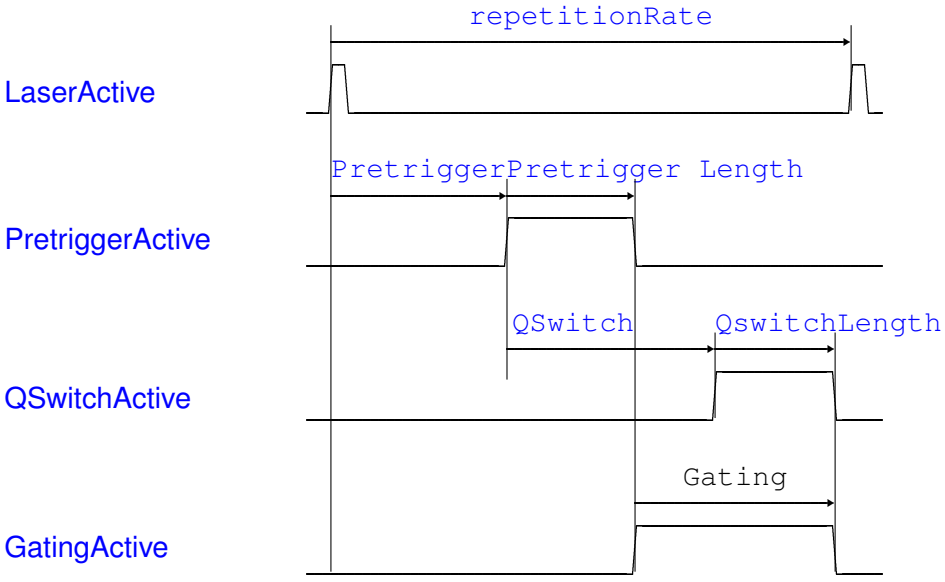
Timing Parameter Explanation

External trigger (ExternalTrigger = True)



∗ Available at controllers newer than 2007-06

Internal trigger (ExternalTrigger = False)



The Laser Lamp pulse has a fixed length of 5μs.

TRIGGERTIMEM<RepetitionRate> <Pretrigger> <PositionTriggersEnable> <QSwitch> <StartDelay> <TriggerPeriod>

Set the timing parameters and position triggers in ns for a Polarotor Controller.

`RepetitionRate`	in internal mode delay between two pulses in ns, in external (chopper) mode ignored.
`Pretrigger`	delay between lamp and pretrigger (acquisition) in ns
`PositionTriggersEnable`	the 4 least significant bits of this number represent the trigger switches (0 = off, 1 = on) of the triggers at the 4 angular positions of the polarotor. Bit 0 corresponds to the quadrant 1 (00) (see the `DRIVESTATUS` command)
`QSwitch`	delay between pretrigger (acquisition) start and Q-Switch start in ns
`StartDelay`	start delay between internal or external (chopper) trigger and lamp in ns
`TriggerPeriod`	estimated time in milliseconds between subsequent laser triggers .

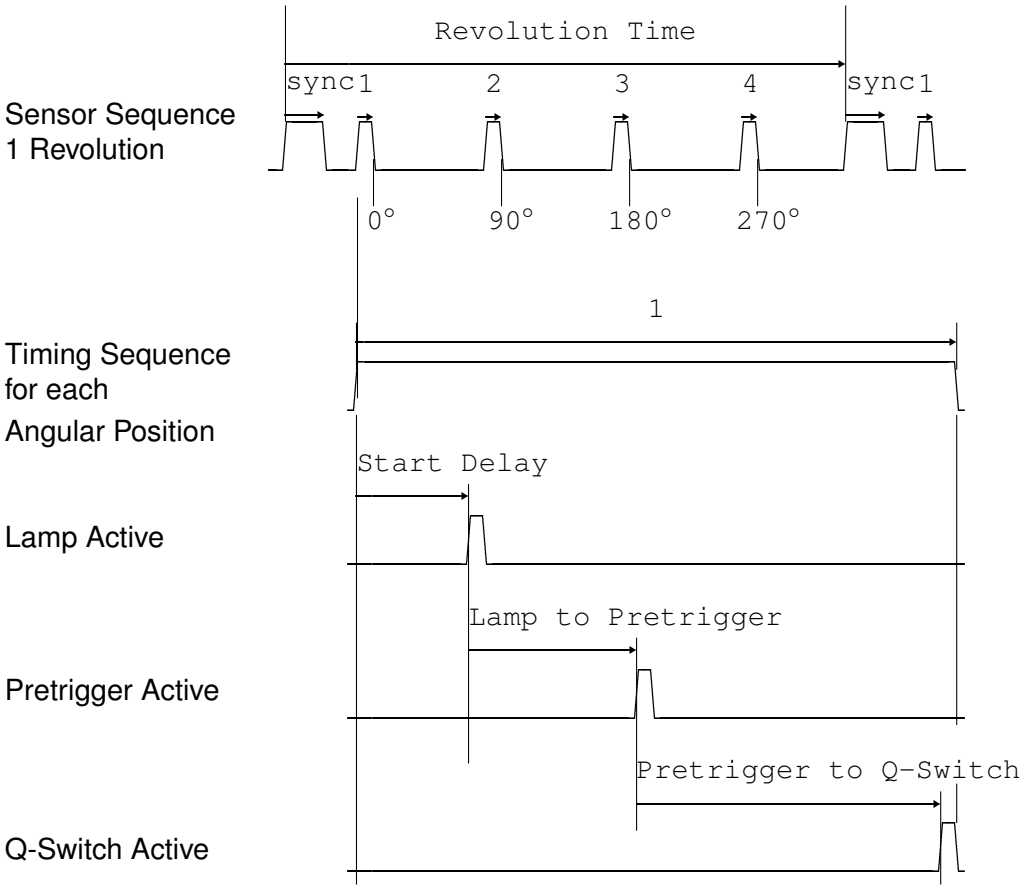
If successful the controller will return:

`TRIGGERTIMEM executed,`

in the case that the parameters cannot be interpreted the reply is

`TRIGGERTIMEM: incorrect or invalid parameters.`

Polarotor Timing Parameters



TRIGCYCLE[BoardID]?

Gives back the internal clock of the timerboard in nano seconds.

A typical response would be

`TRIGCYCLE 12.500001`

for a 12.5ns clock which is the default.

TRIGMINWIDTH[BoardID]?

Return the minimum width of the Pretrigger and Q-switch length of the board specified by `BoardID`; The return value `TrigMinWidth` is given in clock cycles:

`TRIGMINWIDTH <TrigMinWidth>`

and has to be added to the `Pretrigger` and `Q-switch` offsets to obtain the final offset for these parameters.

TRIGSCALE[BoardID]?

Gives back the increment in master clock cycles of the counters a typical response would be

`TRIGSCALE 512 1 1 1 1`

The first number is the lamp counter which control the start delay, it goes with an increment of

512 * 12.5ns = 6400ns

The Pretrigger, the PretriggerLength, the QSwitch and the QswitchLength all have a 12.5ns increment.

TRIGOFFSET[BoardID]?

Gives back the offset in master clock cycles of the counters. A typical response would be

`TRIGOFFSET 267 7 2 4 2`

The first number is the lamp counter which control the minimum start delay, it goes with an increment of

267 * 12.5ns = 3337.5 ns

The Pretrigger has

7 * 12.5ns = 87.5ns

The PretriggerLength has

2 * 12.5ns = 25ns

The QswitchLengthhas

4 * 12.5ns = 50ns

The QswitchLength has

2 * 12.5ns = 25ns

TRIGSLAVE[BoardID]?

Returns the slave state of a timer board, by default boards can react as master or as slaves (see Direct control of the timing parameters)

typical response would be

`TRIGSLAVE 1`

This means the board is internally wired as a slave of another master board. It can not act a master. If a TRIGGERMODE is sent where the 0x10 flag is set, the controller will return an error message

`TRIGGERMODE failed, board is an internal slave and can not act as a master`

If the board is has no internally wired external trigger the controller will return

`TRIGSLAVE 0`

TRTYPE?

Returns hardware information about the `SELECT`ed transient recorder. If more than one TR is selected, the Status of the selected TR with the lowest device number is returned (e.g. when 5, 7, 12, 14 are selected, then `TRTYPE?` refers to TR #5).

If successful the controller will return the number of ADC bits, the number of PC bits, the length of the FIFO, the binwidth in meters, and the ID of the device in controllers delivered after 11-2013 followed by a number representing further hardware capabilities of the transient recorder and the the binshift in primary bins. In the example

```
TRTYPE ADC Bits 12 PC Bits 4 FIFOLength 16384 binwidth 3.75 ID 0
HWCAP 0x00000000 binshift 0.0
```

the SELECTed transient recorder is a TR 40-160 with 12 ADC bit device, 4 PC bits, a FIFO length of 16384, a binwidth of 3.75 meters, and the ID 0. No further hardware capabilities (0x00000000) are available. The hardware capabilities are bitwise coded into the returned hexadecimal number: bit 0 (0x01) indicates whether or not a separate shotcounter for memory B is available, bit 1 (0x02) is the corresponding flag for a separate memory C. The example is completed by the binshift 0.0 (in primary bins). The following hardware capabilities are available (Mai. 2025):

0x01	separate shot counter B
0x02	separate shot counter C
0x04	separate shot counter D
0x08	pretrigger
0x10	memory blocking
0x20	squared data support
0x40	frequency divider
0x80	*reserved*
0x100	apd-flex

The TRTYPE? request is implemented since spring 2011.

WHITELIST <"Password"> <"Host1"> <"Host2"> <"Host3">

Sets the allowed hosts for secure mode operation. Password is the controller password, and Host# is either

- a host specified by its IP address xx.xx.xx.xx ,
- an IP address range xx.xx.xx.255 ranging from 0 to 255, or
- empty .

The example

```
WHITELIST "Administrator" "192.168.69.255" "213.198.20.19" ""
```

grants secure mode access to clients operating from any IP address between 192.168.69.0 and 192.168.69.255, and furthermore from the IP address 213.198.20.19. If successful the controller will return:

```
WHITELIST executed.
```

in case of an invalid password or syntax the reply is

```
WHITELIST not accepted.
```

The secure mode must be enabled after specifying the allowed hosts using the ACCESS command. Clients will have to log in using LOGON.

9.1.2 TCP/IP Command Logging

In Licel's Windows and Linux Applications and by default in Licel's LabVIEW sources the logging of the TCP/IP comands and replies is enabled. To achieve that in the LabVIEW sources the *Conditional Disable Symbol* CMDLOG is set TRUE in the LabVIEW project. Each controller (identified by the IP address and port) will have it's own log file.
There are two ways to activate TCP/IP command logging:

1. Activate the command logging by entering or changing the following lines in the initialization file `LicelTCPIP.ini`:
   ```
   [LogCmd]
   LogCmd = TRUE
   ```
   The command logging will then be active from the start of a program.
2. Activate or deactivate the command logging by using the TCP/IP API command `CMDLOG`.

The log files are located in the sub folder `\log` with respect to the location of the application (Windows, Linux) or to the `<source` folder (LabVIEW sources). The names of the log files are like
`YYYY-mm-dd_HHMMSS_<IPaddress>_<port>_<application>.log`
e.g. `2023-11-01_133233_10.49.234.234_2055_Licel-Main.vi.log`.

9.1.3 TCP/IP Reconnection Error Logging

Errors that lead to the loss of a TCP/IP connection or appear during the connection to a Licel Ethernet controller are always written to separate log files. There is no need to enable this mechanism.
The log files are located in the sub folder `\log` with respect to the location of the application (Windows, Linux) or to the `<source` folder (LabVIEW sources). The names of the log files are like
`<IPaddress>_<port>_<TCPIPversion>.log`
e.g. `10.49.234.234_2055_3.00.09.log`.

9.2 Data File format

This appendix describes the file format written by `TCPIP Acquis`. The files are interoperable between the different platforms. The file format is a mixed ascii-binary format where the first lines describe the measurement situation, below follow the dataset description and then raw data as unsigned 32-bit integer arrays itself.

9.2.1 Sample file header

```
lc1980211.305684
Berlin   02/08/2019 11:30:23 02/08/2019 11:30:56 0045 0013.384373 0052.542185 00.0 00.0
0001000 0010 0001000 0010 12 0000000 0000 0000000 0000
1 0 2 02000 1 0800 7.50 00366.o 0 0 00 000 12 001000 0.500 BT0
1 2 2 02000 1 0800 7.50 00366.o 0 0 00 000 12 001000 0.500 S2A0
1 1 2 02000 1 0800 7.50 00366.o 0 0 00 000 00 001000 3.1746 BC0
1 3 2 02000 1 0800 7.50 00366.o 0 0 00 000 00 001000 3.1746 S2P0
1 0 1 02000 1 0800 7.50 00532.o 0 0 00 000 12 001000 0.500 BT1
1 1 1 02000 1 0800 7.50 00532.o 0 0 00 000 00 001000 3.1746 BC1
1 0 1 02000 1 0800 7.50 00532.o 0 0 00 000 12 001000 0.500 BT1
1 1 1 02000 1 0800 7.50 00532.o 0 0 00 000 00 001000 3.1746 BC1
1 5 0 02000 0 0000 1.00 00000.o 0 0 00 000 00 000000 0.000 OF0
```

Each line of the header is completed with a carriage return and a line feed `0x0D0A` (`CRLF`).

Line 1

Filename string, format: `?[?]YYmddHH.MMSSuu[u]`

- *?(?)* - The first 1 or 2 letters can freely be chosen
- *YY* - two digits showing the years in the century
- *m* - the month (hexadecimal, one digit) (December=C)
- *dd* - the day (decimal, two digits)
- *HH* - the hour (decimal, 24 hours per day, two digits)
- *.* - a period (.)
- *MM* - the minute (decimal, two digits)
- *SS* - the seconds (decimal, two digits)
- *uu(u)* - two or three decimal places of the seconds (decimal, two or three digits)

Line 2

Location	string with 8 characters
Start Time	`dd/mm/YYYY HH:MM:SS`
Stop Time	`dd/mm/YYYY HH:MM:SS`
Height a.s.l.	four digits (meter)
Longitude	11 digits (including - sign). six digits for decimal grades.
Latitude	11 digits (including - sign). six digits for decimal grades.
zenith angle	four digits in degrees, 1 decimal place

azimuth angle	four digits in degrees, 1 decimal place
info	a custom information field enclosed by quotation marks (””) [optional]

Line 3

Laser 1 Number of shots	integer 7 digits
Pulse repetition frequency for Laser 1	integer 4 digits
Laser 2 Number of shots	integer 7 digits
Pulse repetition frequency for Laser 2	integer 4 digits
number of datasets in the file	integer 2 digits
Laser 3 Number of shots	integer 7 digits
Pulse repetition frequency for Laser 3	integer 4 digits
0000000 0000	(reserved)
timestamp	timestamp of the controller corresponding to the data in milliseconds [optional]

Dataset description

Active	1 if dataset is present, 0 otherwise
Dataset type	0 ≡ Analog, 1 ≡ Photon Counting, 2 ≡ Analog squared, 3 ≡ Photon Counting squared, 4 ≡ Power Meter dataset, 5 ≡ Overflow dataset
Laser source	one digit Laser 1 ≡ 1, Laser 2 ≡ 2, Laser 3 ≡ 3, Laser 4 ≡ 4.
Number of bins	5 digits
Laser polarization	none ≡ 0, vertical ≡ 1, horizontal ≡ 2, right circular ≡ 3, left circular ≡ 4
PMT high voltage	four digits in Volt
bin width	in meters four digits including decimal separator (.) and decimal places
Laser wavelength	in nm, five digits period
Polarisation	one letter, o ≡ none, p ≡ parallel, s ≡ crossed, r ≡ right circular, l ≡ left circular
0 0	backward compatibility
bin shift, whole-number	bin shift (primary bins, integer rounded down, 2 digits, 00 if not supported or zero)
bin shift, decimal places	decimal places of the bin shift (3 digits, 000 if not supported or zero)
number of ADC bits	2 digits, in case of an analog dataset, otherwise 0
number of shots	6 digits
analog input range/discriminator level	

analog input range in Volt in case of analog dataset , discriminator level in case of photon counting, one digit period 3/4 digits.

Dataset descriptor

BT analog dataset

BC photon counting

S2A $s\sqrt{N(N-1)}$ analog

S2P $s\sqrt{N(N-1)}$ PC

s sample standard deviation

N shot number

PD Powermeter (Photodiode)

PM Powermeter (Powermeter)

OF Overflow.

The following number is the address of the transient recorder or the power meter controller and detector indices as a hexadecimal number.

info

a custom information field enclosed by quotation marks (””) [optional]. If a transient recorder channel is assigned to a detector, `<detectorID: detector description>` will be part of the custom info field which is always present in that case.

9.2.2 Sample file data

The datasets are arrays of 32bit wide unsigned integer values saved as little endian. Each binary data array is introduced by a `CRLF`. A final `CRLF` is added after the last data array. The `CRLF` can be used – together with the array length information (number of bins) – as check points for file integrity. So the example header from above (already having a `CRLF` after the last line) would be followed by

```
<CRLF><2000 32bit values from the memory analog A, TR 0>
<CRLF><2000 32bit values s√N(N−1) (analog A, TR 0)>
<CRLF><2000 32bit values from the memory photon counting A, TR 0>
<CRLF><2000 32bit values s√N(N−1) (photon counting A, TR 0)>
<CRLF><2000 32bit values from the memory analog A, TR 1>
<CRLF><2000 32bit values from the memory photon counting A, TR 1>
<CRLF><2000 32bit values from the memory analog B, TR 1>
<CRLF><2000 32bit values from the memory photon counting B, TR 1>
<CRLF><2000 32bit values with overflow information for the analog
datasets "analog A, TR 0" and "analog A, TR 1">
<CRLF>
```

The line breaks in this list are for readability, real breaks in the file are marked by ”`<CRLF>`”. `TR` is an abbreviation of *transient recorder*, $s =$ sample standard deviation, $N =$ shot number as in the previous subsection. The data sets can be converted to quantities with physical units by using the number of shots and additionally the number of bits and the data range in case of the analog signal. https://www.licel.com/manuals/programmingManual.pdf#subsection.5.3 contains a description of the conversion of raw data to physical values.

9.2.3 Overflow data set

The overflow data set contains bitwise information whether or not an overflow appeared in a range bin of an analog dataset. For the first analog dataset in an acquisition bit0 (= 1) will be set, for the second bit1 (= 2) is set, and so on. E.g. a decimal value of 3 (binary 11)in a certain range bin indicates that the first and the second analog dataset of the acquisition have an overflow.

9.3 Standard Deviation Data

This appendix describes how standard deviation data is saved for transient recorders supporting the acquisition of squared data.
The sum of counts c at each bin after acquiring N shots is read from the transient recorders (standard photon counting or analog memories):

$$c_{\text{bin}} = \sum_{i=0}^{N-1} x_i \tag{9.1}$$

with the counts x_i of the ith shot.
The mean value μ (counts per shots) acquired in a bin is then

$$\mu = \frac{c_{\text{bin}}}{N} = \frac{\sum_{i=0}^{N-1} x_i}{N}.$$

The *sample standard deviation* is (https://en.wikipedia.org/wiki/Standard_deviation):

$$s = \sqrt{\frac{\sum_{i=0}^{N-1}(x_i - \mu)^2}{N-1}}. \tag{9.2}$$

From there we get the *standard error of the mean* (https://en.wikipedia.org/wiki/Standard_error):

$$\sigma_\mu \approx \frac{s}{\sqrt{N}}. \tag{9.3}$$

As the x_i are the counts for each shot, but the counts are already summed by the transient recorders there is no direct access to standard deviation nor to standard error when reading from standard analog or PC memories.
The solution is to read squared data c^2_{bin} from transient recorders supporting the summation of squared counts. Then one can calculate the sample standard deviation using

$$\begin{aligned} s &= \sqrt{\frac{c^2_{\text{bin}} - (c_{\text{bin}})^2/N}{N-1}} \\ &= \sqrt{\frac{\sum_{i=0}^{N-1} x_i^2 - (\sum_{i=0}^{N-1} x_i)^2/N}{N-1}} \\ &= \sqrt{\frac{N\sum_{i=0}^{N-1} x_i^2 - (\sum_{i=0}^{N-1} x_i)^2}{N(N-1)}} \\ &= \frac{1}{\sqrt{N(N-1)}}\sqrt{N\sum_{i=0}^{N-1} x_i^2 - (\sum_{i=0}^{N-1} x_i)^2} \end{aligned} \tag{9.4}$$

(https://en.wikipedia.org/wiki/Algorithms_for_calculating_variance).
The acquired squared data sqr_{bin} returned from the transient recorders is

$$sqr_{\text{bin}} = \sum_{i=0}^{N-1} {x_i}^2. \tag{9.5}$$

With this and the acquired data from eq. 9.1 the right square root in eq. 9.4 can be written as

$$sqd_{\text{bin}} = \sqrt{N\sum_{i=0}^{N-1} x_i^2 - (\sum_{i=0}^{N-1} x_i)^2} \tag{9.6}$$

$$= \sqrt{N sqr_{\text{bin}} - (c_{\text{bin}})^2}. \tag{9.7}$$

This will fit into 4 byte numbers used in the standard data files and will therefore be saved by TCPIP Acquis.
Reading sqd_{bin} back from the data files will enable to restore the *sample standard deviation* s and the *standard error of the mean* σ_μ as

$$s = \frac{sqd_{\text{bin}}}{\sqrt{N(N-1)}} \text{ and} \tag{9.8}$$

$$\sigma_\mu = \frac{s}{\sqrt{N}}. \tag{9.9}$$

In TCPIP Acquis and the Viewer software data according to eq. 9.9 is displayed.

9.4 The Initialization File acquis.ini

9.4.1 TR configuration

The initialization file acquis.ini contains definition sections for each transient recorder. The data here corresponds to the values set while configuring the transient recorders. The data entries may appear in a different order within a section named [TR<address>]. Most of the entries are automatically written by *Acquis*. Here, the section for the transient recorder with the device address 0 is shown:

```
[TR0]
Discriminator = 0
Range = 0
PM = 0
WavelengthA = 532.000000
PolarisationA = 0
AnalogA = TRUE
A-binsA = 16000
A-reductA = 0
PC A = TRUE
P-binsA = 16000
P-reductA = 0
WavelengthB = 1024.000000
polarisationB = 0
Analog B = FALSE
A-binsB = 0
A-reductB = 0
PC B = FALSE
PC-binsB = 0
PC-reductB = 0
SamplingRate = 20
TRType = 0
TriggerFractionA = 1
TriggerFractionB = 1
Threshold = 0
WavelengthApc = 555.000000
WavelengthBpc = 555.000000
PolarisationApc = 0
PolarisationBpc = 0
PM1pc = 450.000000
PM2 = 450.000000
PM2pc = 450,000000
ShotLimit = 0
ADCBits = 12
PCBits = 4
HWCAP = 0
Binshift = 0.000000
Analog C = FALSE
A-binsC = 16380
A-reduct-C = 0.000000
PC C = FALSE
P-binsC = 16380
P-reductC = 0.000000
```

```
WavelengthC = 555.000000
WavelengthCpc = 555.000000
PolarisationC = 0
PolarisationCpc = 0
PM3 = 450.000000
PM3pc = 450.000000
ID = 0
MemoryDepth = 4
Pretrigger = 0
BlockMemory = FALSE
UserBins = 16384
SquaredData = TRUE
SQR-Bins = 2000
IDhex = "0x00000000"
FreqDivider = 0
LaserA = 0
LaserB = 0
LaserC = 0
Info1 = ""
Info1P = ""
Info2 = ""
Info2P = ""
Info3 = ""
Info3P = ""
AnalogD = FALSE
A-binsD = 16380
A-reductD = 0,000000
WavelengthD = 0,000000
PolarisationD = 0
PM4 = 0,000000
Info4 = ""
LaserD = 0
PC D = FALSE
P-binsD = 16380
P-reductD = 0,000000
WavelengthDpc = 0,000000
PolarisationDpc = 0
PM4pc = 0,000000
Info4P = ""
Detector1 = "<not assigned>"
Detector1p = "<not assigned>"
Detector2 = "<not assigned>"
Detector2p = "<not assigned>"
Detector3 = "<not assigned>"
Detector3p = "<not assigned>"
Detector4 = "<not assigned>"
Detector4p = "<not assigned>"
UseForAll = FALSE
binshiftOffset = 0,000000
```

A section always begins with `[TR<n>]` where n indicates the address of the transient recorder.

`Discriminator` Discriminator level between 0 and 63.

`Range` Input range of the transient recorder. Valid values are

Range Value	Input Range
0	0 – 500 mV
1	0 – 100 mV
2	0 – 20 mV

`PM` Photomultiplier voltage (analog, memA).

`WavelengthA` Wavelength 1 (analog).

`PolarisationA` corresponding detection polarization 1 (analog). Valid values are

Value	Polarization
0	none
1	parallel
2	crossed
3	right Circular
4	left Circular

`AnalogA` (`TRUE` | `FALSE`) Enable or disable analog acquisition for memory A.

`A-binsA` Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)

`A-reductA` Corresponding data reduction level.

`PC A` (`TRUE` | `FALSE`) Enable or disable photon counting acquisition for memory A.

`P-binsA` Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)

`P-reductA` Corresponding data reduction level.

`WavelengthB` Wavelength 2 (analog).

`polarisationB` Corresponding polarization 2 (analog). Valid values are as above for `PolarisationA`.

`Analog B` (`TRUE` | `FALSE`) Enable or disable analog acquisition for memory B.

`A-binsB` Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)

`A-reductB` Corresponding data reduction level.

`PC B` (`TRUE` | `FALSE`) Enable or disable photon counting acquisition for memory B.

`PC-binsB` Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)

`PC-reductB` Corresponding data reduction level.

`SamplingRate` Sampling rate of the transient recorder.

`TRType` Type of the transient recorder. 0: TR, transient recorder with analog and photon counting acquisition capabilities, 1: PR, pure photon counting device. The type must correspond to the hardware you are addressing (TRxx-xx or PRxx-xx, respectively).

TriggerFractionA,B	Fraction of the number of shots used for memories A and B. These parameters are neglected if an acquisition uses only one memory. If N shots have been acquired at the transient recorder the number of shots for the channel j is calculated as $N\frac{\text{TriggerFraction}_j}{\sum_{A,B}\text{TriggerFraction}_i}$. The fractions must be set according to the ratio of the trigger frequency inputs for memories A and B at the corresponding transient recorder.
Threshold	Threshold set in the configuration. Currently not used.
WavelengthApc	Wavelength 1 (photon counting).
WavelengthBpc	Wavelength 2 (photon counting).
PolarisationApc	Polarization 1 (photon counting). Valid values are as above for PolarisationA.
PolarisationBpc	Polarization 2 (photon counting). Valid values are as above for PolarisationA.
PM1pc	Photomultiplier voltage (photon counting, memA).
PM2	Photomultiplier voltage (analog, memB).
PM2pc	Photomultiplier voltage (photon counting, memB).
ShotLimit	Shot limit.
ADCBits	ADC bits.
PCBits	PC bits.
HWCAP	Transient recorder capabilities, for documentation, only.
Binshift	Bin shift of the transient recorder.
Analog C	(TRUE | FALSE) Enable or disable analog acquisition for memory C.
A-binsC	Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)
A-reduct-C	Corresponding data reduction level.
PC C	(TRUE | FALSE) Enable or disable photon counting acquisition for memory C.
P-binsC	Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)
P-reductC	Corresponding data reduction level.
WavelengthC	Wavelength 3 (analog).
WavelengthCpc	Wavelength 3 (photon counting).
PolarisationC	Polarization 3 (analog). Valid values are as above for PolarisationA.
PolarisationCpc	Polarization 3 (photon counting). Valid values are as above for PolarisationA.

PM3	Photomultiplier voltage (analog, memC).
PM3pc	Photomultiplier voltage (photon counting, memC).
ID	Transient recorder ID if available, for documentation.
MemoryDepth	Memory depth if supported.
Pretrigger	(0 | 1) Pretrigger support if available, for documentation.
BlockMemory	(TRUE | FALSE) Memory blocking support.
UserBins	User bins if available
SquaredData	Enable the reading of squared data to calculate the standard error if supported.
SQR-Bins	Number of primary bins to read from squared data memory.
IDhex	The ID in hexadecimal representation.
FreqDivider	Frequency divider (set in the configuration dialog if supported).
LaserA	Laser assignment to memory A (0: by wavelength).
LaserB	Laser assignment to memory B (0: by wavelength).
LaserC	Laser assignment to memory C (0: by wavelength).
Info1	Custom information for memory A, analog.
Info1P	Custom information for memory A, photon counting.
Info2	Custom information for memory B, analog.
Info2P	Custom information for memory B, photon counting.
Info3	Custom information for memory C, analog.
Info3P	Custom information for memory C, photon counting.
Analog D	(TRUE | FALSE) Enable or disable analog acquisition for memory D.
A-binsD	Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)
A-reduct-D	Corresponding data reduction level.
WavelengthD	Wavelength 4 (analog).
PolarisationD	Polarization 4 (analog). Valid values are as above for PolarisationA.
PM4	Photomultiplier voltage (analog, memD).
Info4	Custom information for memory D, analog.
LaserD	Laser assignment to memory D (0: by wavelength).
PC D	(TRUE | FALSE) Enable or disable photon counting acquisition for memory D.

P-binsD	Corresponding number of bins to be read out. The maximum number of bins is given by 16380/($2^{datareduction}$)
P-reductD	Corresponding data reduction level.
WavelengthDpc	Wavelength 4 (photon counting).
PolarisationDpc	Polarization D (photon counting). Valid values are as above for PolarisationA.
PM4pc	Photomultiplier voltage (photon counting, memD).
Info4P	Custom information for memory D, photon counting.
Detector1	Detector assignment for memory A, analog (done in the configuration dialog).
Detector1p	Detector assignment for memory A, photon counting (done in the configuration dialog).
Detector2	Detector assignment for memory B, analog (done in the configuration dialog).
Detector2p	Detector assignment for memory B, photon counting (done in the configuration dialog).
Detector3	Detector assignment for memory C, analog (done in the configuration dialog).
Detector3p	Detector assignment for memory C, photon counting (done in the configuration dialog).
Detector4	Detector assignment for memory D, analog (done in the configuration dialog).
Detector4p	Detector assignment for memory D, photon counting (done in the configuration dialog).
UseForAll	TRUE: use the pm HV, the wavelength and polarization, the detector assignment, and the custom info of memory A, analog for all other channels within the transient recorder.
binshiftOffset	additional (software) offset which is added to the binshift obtained from the transient recorder.

To completely disable a transient recorder AnalogA, PC A, Analog B, and PC B (and, if avalable, AnalogC, PC C, AnalogD, and PC D) must be set to FALSE. This will be done by saving the configuration after configuring the transient recorders.
Important Note: Customers with at least one transient recorder shipped before October 2009 must add for each old transient recorder the following initialization file key to the acquis.ini file when using the TCP/IP 2.44 (Windows or LabVIEW) software or higher:

```
[TR<address>]
PC_Device=FALSE
```

M-Acquis (multi-rack software) users must be aware that the address numbers in the initialization file sections [TR<address>] of transient recorders in further racks are the original device addresses increased by $16 \times$ rackIndex. So the block [TR16] corresponds to the transient recorder with device address 0, block [TR17] corresponds to the transient recorder with device address 1 in the second rack.

9.4.2 TCP/IP Settings

The TCP/IP related initialization keys are interpreted only in the case that the Windows applications (.exe) are in use. LabVIEW users should enter the appropriate TCP/IP values and save them as default.

```
[TCPIP]
UseValues = TRUE
Port = 2055
IPAddress = 10.49.234.234
;; NoOfControllers is for the multi-rack acquis (M-Acquis)
NoOfControllers = 2

;; [TCPIP00], [TCPIP01], ... are for the multi-rack acquis (M-Acquis)
[TCPIP00]
UseValues = TRUE
Port = 2055
IPAddress = 10.49.234.234

[TCPIP01]
UseValues = TRUE
Port = 2055
IPAddress = 10.49.234.235

...
```

UseValues, Port, and IPAddress in the section [TCPIP] are used by the Windows application TCPIP Acquis.exe and TCPIP MPush-Acquis.exe.
The 2-rack software M-Acquis.exe uses NoOfControllers = 2 in the [TCPIP] section and the values in the sections [TCPIP00] and [TCPIP00] for the connections to rack 1 and rack 2, respectively.

9.4.3 Global Configuration Values

In software versions older than TCP/IP 2.50 these values had been stored in the separate initialization file global info.ini.

```
[global_info]
Location = "Berlin"
Longitude = 13,384373
Latitude = 52,542185
Height_asl = 45,000000
working_directory = "/C/temp"
first_letter = "a"
Zenith = 0,000000
Azimuth = 0,000000
frequency1 = 10,000000
frequency2 = 10,000000
frequency3 = 10,000000
frequency4 = 10,000000
NoSafeIncompleteFiles = FALSE
SaveOverflow = TRUE
SyncViewer = FALSE
```

```
[global_info_Laser1]
Laser1_Wavelength0 = 532,000000
```

The values are updated when saving the configuration, wavelengths will be added or removed.

9.4.4 Missing Trigger Behavior

The parameters defining the missing trigger behavior are as follows:

```
[NoTrigger]
DoActionSeconds = 2,000000
Action = 2
RetrySeconds = 5,000000
```

The parameters correlate with the control fields shown in the configuration dialog.

9.4.5 Power Meter Settings

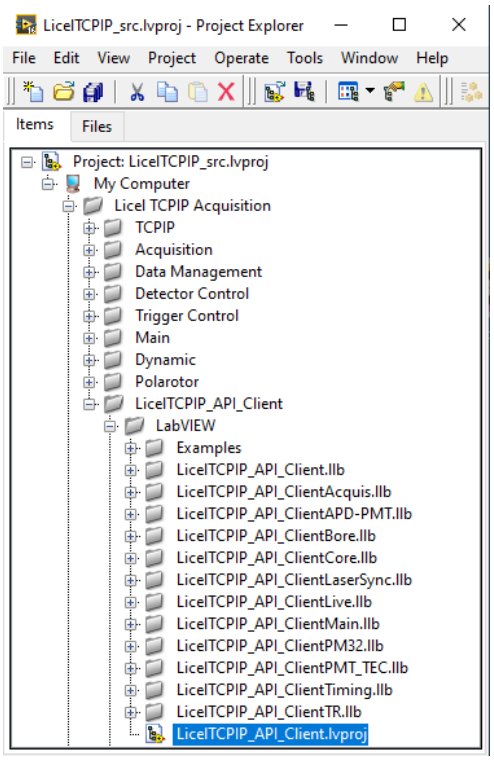
The power meter settings defined in the configuration dialog are stored in acquis.ini:

```
[Powermeter]
Number = 1
Active00 = TRUE
Detectors00 = 3
Channel00 = 0
Wavelength00_0 = 532,000000
Laser00_0 = 1
Wavelength00_1 = 1064,000000
Laser00_1 = 2
Wavelength00_2 = 266,000000
Laser00_2 = 0
```

Here, one power meter with three detector heads is defined.

9.5 Monitoring and Controlling Software Components

In this section the TCP/IP API for controlling the Licel TCPIP Acquisition software from outside is described. You may write a client for the TCP/IP API using any programming language which supports TCP/IP socket handling.
As a starting point for your own programming project you are invited to use basic programming examples in Perl and Python https://www.licel.com/download/ethernet/LicelMain_Basic_PerlPython.zip.



A TCP/IP API client driver for LabVIEW is included in the LabVIEW sources. Simple examples to control *Control APD-PMT*, *Control Timing*, *TCPIP Acquis*/*M-Acquis*, and *Licel Main* are included.

Furthermore simple perl and python example scripts are provided.

9.5.1 Licel Main, Licel Main-M

TCP/IP Server

The basic functions of the *Licel Main* software can be accessed from third party applications via TCP/IP. For this *Licel Main* implements a TCP/IP server listening on a defineable port. To activate the TCP/IP server the following initialization file keys in `Licel Main.ini` have to be aligned:

```
[TCPIP_API]
Active = TRUE
Port = 2088
```

If `Active` is set `TRUE` a listener will be started using the specified TCP/IP port (`Port = 2088`).

Controlling Sub Modules

If a module supports implementing a TCP/IP server and has the key `TCPIP_API = TRUE` in the initialization file Licel Main.ini

```
[Module0]
...
TCPIP_API = TRUE
```

Licel Main will activate the TCP/IP server in the software module using a generated listener port. The module's TCP/IP server is automatically accessed by *Licel Main*: Whenever *Licel Main* receives an unknown TCP/IP API command while the active tab page contains the module, the command is passed through to the module's TCP/IP API server and handled there.

Command List

The following list contains the supported commands. The commands must be sent with an additional <CRLF> (0x0D0A) and the responses will end with a <CRLF>, as well.

- `MAIN: VER?`
  Parameters
  Description Return the program version number as displayed in the Windows title bar
  Reply `VER <version>`
- `MAIN: *IDN?`
  Parameters
  Description Return the *IDN?-information from the controller(s). This command is available only when the current tab page is the *System* or *TCP/IP* page.
  Reply `<IDN(0)>[; IDN(1) ...]`
- `MAIN: LIST?`
  Parameters
  Description Return the supported TCP/IP server commands
  Reply `<"command0"> <"command1"> ...`
- `MAIN: NUMTABS?`
  Parameters
  Description Return the number `numTabs` of visible tab pages
  Reply `MAIN: NUMTABS <numTabs>`
- `MAIN: TAB?`
  Parameters
  Description Return the index `currentTab` of the current tab page within the visible tab pages
  Reply `MAIN: TAB <currentTab>`
- `MAIN: TABNAME?`
  Parameters
  Description Return the `tabName` of the current tab page
  Reply `MAIN: TABNAME <"tabName">`
- `MAIN: TABNAMES?`
  Parameters
  Description Return the `tabName`s of the visible tab pages
  Reply `MAIN: TABNAMES <"tabName(0)"> ["tabName(1)" ...]`
- `MAIN: CURRENTVI?`
  Parameters
  Description Return the name `currentVI` of the VI running in the current tab page. If the command is sent while the *System* or *TCP/IP* tab page is active, the name of the main VI (*Licel Main.vi* is returned).
  Reply `MAIN: CURRENTVI<"currentVI">`

- `MAIN: TAB`
  Parameters `<setTab>`
  Description Set the current tab by submitting the index `setTab` of the current tab page within the visible tab pages. This is not allowed if the current sub module blocks the tab page navigation. `MAIN: TAB -1` will activate the *System* or *TCP/IP* tab page.
  Reply `MAIN: TAB executed` or `ERROR MAIN: TAB Blocked` or `ERROR MAIN: TAB out of range`
- `MAIN: STATUS?`
  Parameters
  Description Return status information, `READY` indicates whether or not all modules have been loaded, `TCPIP_OK` indicates whether or not the TCP/IP connection is alive, `BUSY` indicates whether or not the current module is busy
  Reply `MAIN: STATUS READY=0|1 TCPIP_OK=0|1 BUSY=0|1`
- `MAIN: CONFIG?`
  Parameters `[SIZE]`
  Description request the name of the initialization file, optionally return the file `size` in bytes
  Reply `CONFIG <filepath> [SIZE=size]`
- `MAIN: GETCONFIG`
  Parameters
  Description request the binary file data of the initialization file
  Reply `CONFIG BYTES=size<CRLF><file>` where `size` is the number of bytes of the file, `file` is the file content. Note that as usual a `<CRLF>` will be added.
- `MAIN: CONFIG`
  Parameters `<BYTES=size><CRLF><file>`
  Description Transfer `size` bytes in `file` to replace the current initialization file. Note that as usual a `<CRLF>` must be added and that there are no further white spaces in the command parameters.
  Reply `CONFIG executed` if the transferred file is suitable to replace the current file, or `CONFIG ERROR: error message`
- `MAIN: RELOAD`
  Parameters
  Description Stop and unload all modules, read the initialization file and reload and restart all modules defined in it. The TCP/IP API connection stays alive during the reload process.
  Reply `RELOAD initiated`
- `other commands`
  Parameters
  Description *Licel Main*'s API will pass the commands through to the software module loaded in the current tab page if the module's TCP/IP API is available and controlled by *Licel Main*
  Reply `reply received from the module`
- `MAIN: CMDLOG?`
  Parameters
  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources.
  Reply `MAIN: CMDLOG 0|1`

- `MAIN: CMDLOG`
  Parameters `0|1`
  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources.
  Reply `MAIN: CMDLOG executed`
- `MAIN: TABS_LOCKED`
  Parameters `<0|1>`
  Description Lock (`1`) or unlock (`0`) manual tab switches. Tab locking is disabled when an API client disconnects or a user explicitely clicks on the *Unlock* button.
  Reply `MAIN: TABS_LOCKED executed`
- `MAIN: TABS_LOCKED?`
  Parameters
  Description Return the current tab locking status.
  Reply `MAIN: TABS_LOCKED <0|1>`
- `MAIN: PANEL_LOCKED`
  Parameters `<0|1>`
  Description Lock (`0`) or unlock (`0`) the panel on the current tab page. Panel locking is disabled when an API client disconnects, the tab is changed programmatically or manually, or a user explicitely clicks on the *Unlock* button..
  Reply `MAIN: PANEL_LOCKED executed`
- `MAIN: PANEL_LOCKED?`
  Parameters
  Description Return the locking status of the panel on the current tab page.
  Reply `MAIN: PANEL_LOCKED <0|1>`
- `MAIN: SCREENSHOT?`
  Parameters
  Description Generate a screenshot and return it as a saveable PNG-string.
  Reply `SCREESHOT BYTES=size<CRLF><pngString>` where `size` is the number of bytes of the `pngString` which could directly be saved to a binary PNG file by the API client. Note that as usual a `<CRLF>` will be added.
- `MAIN: QUIT`
  Parameters
  Description exit the program, allowed only when the following key in `Licel Main.ini` is set:
  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```
  Reply `MAIN: QUIT executed` or `MAIN: QUIT ERROR not allowed`

9.5.2 TCPIP Acquis, M-Acquis

TCP/IP Server

The basic functions of the *TCPIP Acquis* software can be accessed from third party applications via TCP/IP. For this *TCPIP Acquis* implements a TCP/IP server listening on a defineable port. To activate the TCP/IP server the following initialization file keys in `Acquis.ini` have to be aligned:

```
[TCPIP_API]
Active = TRUE
Port = 2088
```

If `Active` is set `TRUE` a listener will be started using the specified TCP/IP port (`Port = 2088`).
If *TCPIP Acquis* is run within *Licel Main* (itself running a TCP/IP API server) and the key `TCPIP_API = TRUE` is set in the Module section of `Licel Main.ini`,

```
[Module5]
Active = TRUE
Path = TCPIP Acquis.vi
...
TCPIP_API = TRUE
```

the TCP/IP server of *TCPIP Acquis* is accessed automatically via *Licel Main* using a generated listener port. Whenever *Licel Main* receives an unknown TCP/IP API command while the active tab page contains *TCPIP Acquis*, the command is passed through to the *TCPIP Acquis* TCP/IP API server and handled there.

TCP/IP API Command List

The following list contains the supported commands. The commands must be sent with an additional `<CRLF>` (`0x0D0A`) and the responses will end with a `<CRLF>`, as well.

- `VER?`
  Parameters
  Description Return the program version number as displayed in the Windows title bar
  Reply `VER <version>`
- `*IDN?`
  Parameters
  Description Return the *IDN?-information from the controller(s)
  Reply `<IDN(0)>[; IDN(1) ...]`
- `NUMDEV?`
  Parameters
  Description Return the number of installed transient recorders (`numDev`)
  Reply `NUMDEV <numDev>`
- `INSTALLED?`
  Parameters
  Description Return the list of installed transient recorders, (`TR_List` corresponds to the list in the configuration dialog and to the `[TRi]`-entries in the initialization file)
  Reply `INSTALLED <TR_List>`
- `CONFIG?`
  Parameters `[SIZE]`
  Description request the name of the initialization file, optionally return the file `size` in bytes
  Reply `CONFIG <filepath> [SIZE=size]`
- `CHECKCONFIG`
  Parameters
  Description check the current initialization file without applying any changes
  Reply `CONFIG OK`: the current ini file is suitable to run *Acquis*, or `CONFIG ERROR`: followed by one or more of certain error keywords

- CONFIG
  - *Parameters* `[ACCEPTTR] <BYTES=size><CRLF><file>`
  - *Description* Transfer `size` bytes in `file` to replace the current initialization file. Note that as usual a `<CRLF>` must be added and that there are no further white spaces in the command parameters. The optional parameter `ACCEPTTR` will accept changes in the submitted initialization file with respect to the last configuration (number of transient recorders, completeness of initialization keys, hardware mismatches, and individual TR settings). *Acquis* will always coerce these settings to a valid configuration.
  - *Reply* `CONFIG executed` if the transferred file is suitable to replace the current file, or `CONFIG ERROR`: followed by one or more of certain error keywords
- GETCONFIG
  - *Parameters*
  - *Description* request the binary file data of the initialization file
  - *Reply* `CONFIG BYTES=size<CRLF><file>` where `size` is the number of bytes of the file, `file` is the file content. Note that as usual a `<CRLF>` will be added.
- SHOTS
  - *Parameters* `<number>`
  - *Description* set the target shot number
  - *Reply* `SHOTS executed`
- SHOTS?
  - *Parameters*
  - *Description* get the target shot number
  - *Reply* `SHOTS <targetShots>`
- RECORDS
  - *Parameters* `<number>`
  - *Description* set the number of acquisitions (records) (0: unlimited)
  - *Reply* `RECORDS executed`
- RECORDS?
  - *Parameters*
  - *Description* get the target number of acquisitions (records)
  - *Reply* `RECORDS <targetRecords>`, (0: unlimited)
- START
  - *Parameters*
  - *Description* start a single acquisiton
  - *Reply* `START executed`
- STOP
  - *Parameters*
  - *Description* stop a running acquisiton
  - *Reply* `STOP executed`
- AUTO
  - *Parameters* `<0|1>`
  - *Description* start (0) or stop (1) multiple acquisitons
  - *Reply* `AUTO executed`
- FILE?
  - *Parameters* `[SIZE]`
  - *Description* request the name of the last written file, optionally return the file `size` in bytes
  - *Reply* `FILE <filepath> [SIZE=size]`

- GETFILE
  Parameters
  Description request the binary file data of the latest data file
  Reply FILE BYTES=size<CRLF><file> where size is the number of bytes of the file, file is the file content. Note that as usual a <CRLF> will be added.
- STATUS?
  Parameters
  Description request the current acquisition status. SINGLE indicates whether or not a single acquisition is running, AUTO indicates whether or not a multiple acquisition is running, records is the current record number, shots is the current shot number, TR_SET 0: the program is waiting for settings the transient recorder parameters, 1: the transient recorder parameters have been set, NOTRIG indicates whether or not the program does not receive a trigger during a multiple acquisition, WTRIG indicates whether or not the program waits for receiving a trigger during a multiple acquisition, TCPIP_OK indicates whether or not the TCP/IP connection is alive
  Reply STATUS AUTO=<0|1> SINGLE=<0|1> SHOTS=<shots> RECORDS=<records> TR_SET=<0|1> NOTRIG=<0|1> WTRIG=<0|1> TCPIP_OK=<0|1>
- SETDIR
  Parameters <directory>
  Description set the storage directory of the data files
  Reply SETDIR executed
- DIRECTORY
  Parameters <directory>
  Description set the storage directory of the data files (for compatibility)
  Reply DIRECTORY executed
- SETFIRSTLETTER
  Parameters <firstletter>
  Description set the first (and second) letter of the data files
  Reply SETFIRSTLETTER executed
- SETSTOPNOSAFE
  Parameters <0|1>
  Description disable or enable the prevention of saving incomplete files in multiple acquisition mode
  Reply SETSTOPNOSAFE executed
- SETLOC
  Parameters <location>
  Description set the location (measurement site), <location> must have a length of 1 . . . 8 characters
  Reply SETLOC executed
- SETLONG
  Parameters <longitude>
  Description set the longitude of the measurement site, <longitude> must be in the range -180 . . . 180
  Reply SETLONG executed

- SETLAT
  Parameters <latitude>
  Description set the latitude of the measurement site, <latitude> must be in the range -90 ... 90
  Reply SETLAT executed
- SETALT
  Parameters <altitude>
  Description set the altitude (height above sea level) of the measurement site
  Reply SETALT executed
- SETZENITH
  Parameters <zenithAngle>
  Description set the zenith angle
  Reply SETZENITH executed
- SETAZIMUTH
  Parameters <azimuthAngle>
  Description set the azimuth angle
  Reply SETAZIMUTH executed
- SETMAXSHOTS
  Parameters <0|1>
  Description enable (1) or disable (0) setting the target shot numbers directly at the transient recorders if this feature is supported by the transient recorders. Enable (1) is not allowed if the defined laser frequencies are not equal.
  Reply SETMAXSHOTS executed
- SETFREQ
  Parameters <number> <frequency>
  Description set the frequency of the laser with the given number (1 ... 4). If the frequencies are not equal after executing the command, an active SETMAXSHOTS will be disabled
  Reply .

  SETFREQ executed
- SETLASER
  Parameters <number> <wavelength[0] polarization[0] > [wavelength[1] polarization[1] ...]
  Description set the wavelengths and polarizations of the laser with the given number (1 ... 4), allowed values for polarizations are 0 (none), 1 (vertical), 2 (horizontal), 3 (right circular), and 4 (left circular).
  Reply SETLASER executed
- SETCUSTOMINFO
  Parameters <"customInfo">
  Description set global custom info string
  Reply SETCUSTOMINFO executed

- `MTB?`
  - *Parameters*
  - *Description* request the current missing trigger behavior as defined in the configuration dialog. The command returns the maximum time in seconds the software accepts missing trigger pulses before showing the missing trigger LED (`Action_s`), the defined `Action` applied after`Action_s` has elapsed without having received a trigger (values: 0 (Ignore), 1 (Stop – NoSafe), 2 (Stop – NoSafe – Retry), 3 (Exit – NoSafe)), and is the time in seconds to wait for the next start after loosing a trigger if `Action` = 2 (`Retry_s`).
  - *Reply* `MTB <Action_s> <Action> <Retry_s>`
- `MTB`
  - *Parameters* `<Action_s> <Action> <Retry_s>`
  - *Description* set the missing trigger behavior as in the configuration dialog. `Action_s` is the maximum time in seconds the software accepts missing trigger pulses before showing the missing trigger LED. `Action` is applied after `Action_s` has elapsed without having received a trigger, allowed values: 0 (Ignore), 1 (Stop – NoSafe), 2 (Stop – NoSafe – Retry), 3 (Exit – NoSafe). `Retry_s` is the time in seconds to wait for the next start after loosing a trigger if `Action` = 2. The initialization file will be updated after successfully executing the command.
  - *Reply* `MTB executed` or an error message
- `CUSTOMINFO`
  - *Parameters* `<trAddress> <memory> <mode> <"customInfo">`
  - *Description* Define the custom info for the channel in the data file specified by the transient recorder address `trAddress`,`memory` = `A|B|C|D`, and `mode` = `AN|PC`
  - *Reply* `CUSTOMINFO executed` or an error message `CUSTOMINFO ERROR: <description>`
- `DATASET`
  - *Parameters* `<trAddress> <memory> <mode> <active> <noBins> <reduction> <wavelength> [Laser [Pol PM_HV]]`
  - *Description* Define a dataset to acquire at the transient recorder with the address `trAddress` with the given parameters.
    `memory` = `A|B|C|D`, `mode` = `AN|PC` (analog or photon counting), `active` = `0` (disable) or `1` (enable and use for the next acquisition), textttnoBins, `reduction` = number of (primary) bins to read with the given reduction, `wavelength` in nm, `Laser` laser number (`1|1|2|3|4`), `Pol` polarization (detection), `0` (None), `1` (Parallel), `2` (Crossed), `3` (Right Circular), or `4` (Left Circular), `PM_HV` detector voltage. The parameters will be checked, and if they are correct they will be applied and the initialization file will be updated.
  - *Reply* `DATASET executed` or an error message `DATASET ERROR: <description>`
- `TR?`
  - *Parameters* `<TR_index>`
  - *Description* request the dataset-independent transient recorder parameters of the transient recorder with the index `TR_index`
  - *Reply* `TR <TR_JSON_string>` where `TR_JSON_string` is a JSON string
- `TR`
  - *Parameters* `<TR_index> <TR_JSON_string>`
  - *Description* set the dataset-independent transient recorder parameters of the transient recorder with the index `TR_index` using the `TR_JSON_string`
  - *Reply* `TR executed` or an error message `TR ERROR: <description>`

- `POW?`
  Parameters
  Description Return the current power meter configuration
  Reply `POW <numPow>[Active(0) numDet(0) Channel(0) IP(0) Port(0) Wavelength(00) Laser(00) [Wavelength(01) Laser(01) ...] [Active(1) ...]]` where `numPow` is the number of power meter controllers followed by the controller parameters: `Active =0|1` is the *Active* flag in the configuration dialog, `numDet` is the number of used detectors at the controller, `Channel = 0` (Photodiode) or `2` (Powermeter), `IP` and `Port` are the IP address and TCP/IP port followed by `numDet` pairs with the `Wavelength` in nm and the assigned *Laser* (`0`: by wavelength) for each detector. Then the parameters for the next power meters are transferred.
- `POW`
  Parameters `<numPow>[Active(0) numDet(0) Channel(0) IP(0) Port(0) Wavelength(00) Laser(00) [Wavelength(01) Laser(01) ...] [Active(1) ...]]`
  Description Set the current power meter configuration. As in the `POW?` command, `numPow` is the number of power meter controllers followed by the controller parameters: `Active =0|1` is the *Active* flag in the configuration dialog, `numDet` is the number of used detectors at the controller, `Channel = 0` (Photodiode) or `2` (Powermeter), `IP` and `Port` are the IP address and TCP/IP port followed by `numDet` pairs with the `Wavelength` in nm and the assigned *Laser* (`0`: by wavelength) for each detector. Then the parameters for the next power meters have to be transferred. Please note that at least one of the power meter controllers must be set active. `POW 0` will disable all power meters.
  Reply `POW executed` or an error message `POW ERROR: <description>`
- `POW`
  Parameters `<START>`
  Description Starts the configured power meters and transfer data to *Acquis*. After that the last power meter readings can be obtained for inspection using `LASTPOWERMETERREADINGS`.
  Reply `POW START executed`
- `POW`
  Parameters `<STOP>`
  Description Stop the data transfer from the configured power meters to *Acquis*.
  Reply `POW STOP executed`
- `LASTPOWERMETERREADINGS`
  Parameters
  Description Return the last readings of the *active* power meters specified by their controller indices `pmIndex` and detector indices `pmDetIdx` within the controller (for power meters supporting more than 1 detector).
  Reply `POWER <pm[0] value[0] <pm[1] value[1] <...>>>` where `pm[index]` is the index of a power meter detector (= `[pmIndex] [pmDetIdx]`). If a power meter has not yet received data, a minus sign `"-"` is returned instead of a numerical value.

- `CMDLOG?`

  Parameters

  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Acquis* is running under *Licel Main*

  Reply `CMDLOG 0|1`

- `CMDLOG`

  Parameters `0|1`

  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Acquis* is running under *Licel Main*

  Reply `CMDLOG executed`

- `QUIT`

  Parameters

  Description exit the program, allowed only when the following key in `acquis.ini` is set:

  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```

  and *TCPIP Acquis* is *not* run within *Licel Main*

  Reply `QUIT executed` or `QUIT ERROR not allowed`

Remarks and further Explanations

`CHECKCONFIG`, `CONFIG` error keywords

`TR_SettingsMismatch`	the transient recorder settings in the file do not match the installed hardware
`NumberTR_Changed`	the number of used transient recorder entries changed since the last run (e.g. when you switch between *TCPIP Acquis* and *M-Acquis*), then a check is necessary
`GlobalConfigIncomplete`	the global configuration is incomplete
`DirectoryWrong`	the working directory is not correct (`GlobalConfigIncomplete` is returnded in that case , as well)
`TCP/IP_Mismatch`	the TCP/IP settings are not correct (only applied if you are not using *Licel Main* to run *Acquis*)
`Error=<code>`	an error specified with it's `code` occurred
`API_Mismatch`	the TCP/IP API settings are not correct

Transient Recorder Parameters in JSON Format

The transient recorder recorder parameters in JSON format are used to get and modify transient recorder parameters which are not part of dataset specific properties. To get and modify the parameters the `TR?` and `TR` commands are used. A JSON string sent to the TCP/IP API must not contain `<CRLF>`. The transient recorder recorder parameters in JSON format look as in the following example:

```
{"Device":0,
 "ID":2057,
 "HWCAP":121,
 "TRType":0,
```

```
"SamplingRateMHz":40,
"Discriminator":8,
"Range_mV":1,
"Threshold":0,
"ADC_Bits":16,
"PC_Bits":8,
"ShotLimit":0,
"SquaredData":0,
"SQR_Bins":4090,
"Binshift":0,
"UserBins":90,
"Pretrigger":0,
 "BlockTrigger":0,
"TriggerRatioA":1,
"TriggerRatioB":1}
```

<CRLF> are not allowed. Please note that one or more of these properties may not be changeable on your system because they are not supported by the individual transient recorder or by the Licel Ethernet Controller in use. Refer as well to the description of the configuration dialog.

`Device`	lindex of the transient recorder in the configuration
`ID`	ID of the transient recorder, read-only
`HWCAP`	indicates the transient recorder's hardware capabilities (decimal, read-only). `HWCAP` equals zero for older transient recorders. The following transient recorder capabilities are coded: `0x01` separate shot counter B `0x02` separate shot counter C `0x04` separate shot counter D `0x08` pretrigger `0x10` memory blocking `0x20` squared data support `0x40` frequency divider `0x80` *reserved* `0x100` apd-flex
`TRType`	O: TR, 1: PR (photon counting only)
`SamplingRateMHz`	sampling rate in MHz
`Discriminator`	discriminator lvel. range: 0 . . . 63
`Range_mV`	mV-range, 0: 500 mV, 1: 100 mV, 2: 20 mV
`Threshold`	0: low, high: 1 (set the discriminator level four times as high)
`ADC_Bits`	analog bits, ca't be changed in new transient recorders
`PC_Bits`	analog bits, ca't be changed in new transient recorders
`ShotLimit`	0: 4k, 1: 64k
`SquaredData`	read squared data, 0: disable, 1: enable
`SQR_Bins`	number of bins for squared data
`Binshift`	binshift (if supported)
`UserBins`	user-defineable number of bins, currently not used
`Pretrigger`	pretrigger, 0: disable, 1: enable
`BlockTrigger`	block trigger, 0: disable, 1: enable
`TriggerRatioA`	Fraction of the number of shots acquired at trigger A with respect to the value at trigger B (older transient recorders)
`TriggerRatioB`	Fraction of the number of shots acquired at trigger B with respect to the value at trigger A (older transient recorders)

Recommendations while using the TCP/IP API
Please consider the following recommendations while using the TCP/IP API of *TCPIP Acquis*:

1. Check the current configuration file using the `CHECKCONFIG` command.
2. Before changing the configuration file using the `CONFIG` command, compare the file you would like to send with the file you obtain with the `GETCONFIG` command.
3. Note that the path you submit via the TCP/IP API must be accessable from the host PC where *Acquis* is running.
4. Before sending transient recorder parameters it is recommended to obtain the current parameters using the `TR?` command and then modify the desired parameters, only.

Queue Control

The basic functions of the LabVIEW version of the *TCPIP Acquis* software can be accessed from third party LabVIEW VIs using LabVIEW's named queue mechanism. *TCPIP Acquis* uses a listening queue named `ACQUIS_LISTEN` to accept commands, and a reply queue `ACQUIS_REPLY` to send answers to the commands received via the listening queue. If your TCPIP Acquis is controlled by this queue mechanism please remember to wait for the reply to the command you sent. Further development of this feature has terminated.

Queue Command List

The following table lists the queue commands:

- `SETSHOT`
  Parameters `<number>`
  Description set the target shot number
  Reply `SETSHOT <number> executed`
- `SETACQ`
  Parameters `<number>`
  Description set the number of acquisitions (0: unlimited)
  Reply `SETACQ <number> executed`
- `START`
  Parameters
  Description start a single acquisiton
  Reply `START executed`
- `STOP`
  Parameters
  Description stop a running acquisiton
  Reply `STARTMULTI executed`
- `STARTMULTI`
  Parameters
  Description start multiple acquisitons
  Reply `STARTMULTI executed`
- `GETSHOT`
  Parameters
  Description request the current number of shots
  Reply `SHOT <number>`

- GETACQ
  Parameters
  Description request the current number of acquisitions
  Reply ACQ <number>
- GETFILE
  Parameters
  Description request the name of the last written file
  Reply FILE <filepath>
- SETFIRSTLETTER
  Parameters <firstletter>
  Description set the first letter of the data files
  Reply SETFIRSTLETTER <firstletter> executed
- SETDIR
  Parameters <directory>
  Description set the storage directory of the data files
  Reply SETDIR <directory> executed
- SETFREQ
  Parameters <number> <frequency>
  Description set the frequency of the laser with the given number (1,2, or 3)
  Reply SETFREQ <number> <frequency> executed
- SETZENITH
  Parameters <zenithAngle>
  Description set the zenith angle
  Reply SETZENITH <zenithAngle> executed
- DATASET
  Parameters <trAddress> <memory> <mode> <active> <noBins> <reduction> <wavelength>
  Description Define a dataset to acquire at the transient recorder with the given address trAddress with the transmitted parameters. memory = A|B|C, mode = AN|PC (analog or photon counting), active = 0 (disable) or 1 (enable and use for the next acquisition), noBins, reduction = number of (primary) bins to read with the given reduction, wavelength in nm
  Reply DATASET executed
- QUIT
  Parameters
  Description exit the acquisition program
  Reply QUIT executed

Notifier to Send the File Name

Whenever TCPIP Acquis saves a file it will use a named notifier ACQUIS_FILE to report the full path of the written file. This mechanism can be used by other LabVIEW VIs to monitor whether or not new acquired data is available in the file system for further processing.

9.5.3 TCPIP Live Display

TCP/IP Server

Some functions of the *TCPIP Live Display* software can be accessed from third party applications via TCP/IP. For this *TCPIP Live Display* implements a TCP/IP server listening on a defineable port. To

activate the TCP/IP server the following initialization file keys in `TCPIP Live Display.ini` have to be aligned:

```
[TCPIP_API]
Active = TRUE
Port = 2088
```

If `Active` is set `TRUE` a listener will be started using the specified TCP/IP port (`Port = 2088`).
If *TCPIP Live Display* is run within *Licel Main* (itself running a TCP/IP API server) and the key `TCPIP_API = TRUE` is set in the Module section of `Licel Main.ini`,

```
[Module3]
Active = TRUE
Path = TCPIP Live Display.vi
...
TCPIP_API = TRUE
```

the TCP/IP server of *TCPIP Live Display* is accessed automatically via *Licel Main* using a generated listener port. Whenever *Licel Main* receives an unknown TCP/IP API command while the active tab page contains *TCPIP Live Display*, the command is passed through to the *TCPIP Live Display* TCP/IP API server and handled there.
The TCP/IP server of *TCPIP Live Display* may be used to quickly check whether or not the received signals are in the expected range.

Command List

The following list contains the supported commands. The commands must be sent with an additional `<CRLF>` (`0x0D0A`) and the responses will end with a `<CRLF>`, as well.

- `LIVE: VER?`
  Parameters
  Description Return the version number as displayed in the Windows title bar
  Reply `VER <version>`
- `LIVE: *IDN?`
  Parameters
  Description Return the *IDN?-information from the controller
  Reply `<IDN>`
- `LIVE: INSTALLED?`
  Parameters
  Description Return the list of installed transient recorders, (`TR_List` corresponds to the list in the configuration dialog and to the `[TRi]`-entries in the initialization file)
  Reply `LIVE: INSTALLED <TR_List>`
- `LIVE: BINS`
  Parameters `<strobeNumber>`
  Description Set the number of bins (*Strobe Number*) to read from the active transient recorder
  Reply `LIVE: BINS executed`
- `LIVE: BINS?`
  Parameters
  Description Request the current number of bins (*Strobe Number*)
  Reply `LIVE: BINS <strobeNumber>`

- `LIVE: SHOTS`
  Parameters `<targetShots>`
  Description Set the number of target shots to acquire from the active transient recorder
  Reply `LIVE: SHOTS executed`
- `LIVE: SHOTS?`
  Parameters
  Description Request the current number of target shots
  Reply `LIVE: SHOTS <targetShots>`
- `LIVE: RANGE`
  Parameters `<range>`
  Description Set the ADC range of the active transient recorder, allowed values for the `range` are 0 (500 mV), 1 (100 mV), and 2 (20 mV)
  Reply `LIVE: RANGE executed`
- `LIVE: DISCRIMINATOR`
  Parameters `<discriminator>`
  Description Set the discriminator of the active transient recorder for photon counting acquisitions, allowed values for the `discriminator` are 0 ... 63
  Reply `LIVE: DISCRIMINATOR executed`
- `LIVE: DAMPING`
  Parameters `<0|1>`
  Description Set the discriminator to four times the set value (`1`) or to the originally set discrimnator value (`0`)
  Reply `LIVE: DAMPING executed`
- `LIVE: SETOVERFLOWZERO`
  Parameters `<0|1>`
  Description Set the acquired values to zero in case of an overflow (`1`) or not (`0`)
  Reply `LIVE: SETOVERFLOWZERO executed`
- `LIVE: STATUS?`
  Parameters
  Description Return some status information about push mode activity (`PUSH=0|1`), the number of `acquiredShots`, the TCP/IP connection status (`TCPIP_OK=1`: connected), and whether or not the freeze button is active (`FREEZE=0|1`)
  Reply `LIVE: STATUS PUSH=0|1 SHOTS=acquiredShots TCPIP_OK=0|1 FREEZE=0|1`
- `LIVE: FREEZE`
  Parameters `0`
  Description Switch off the freeze button (continue live acquisitions)
  Reply `LIVE: FREEZE executed`
- `LIVE: CURSORS`
  Parameters `<minCursorBins> <maxCursorBins>`
  Description Set the cursor positions in range bins (format: `%f`) to define the range for the mean calculation
  Reply `LIVE: CURSORS executed`
- `LIVE: MEAN?`
  Parameters
  Description Return the current mean calculation values `mean`, `stdDev`, and `relError` and as shown on the front panel (format: `%f`) and the number of acquired `cycles` (completed acquisitions with the requested target shots) used in the mean plot
  Reply `LIVE: MEAN VALUE=mean STDDEV=stdDev RELERROR=relError CYCLES=cycles`

- `LIVE: CLEARMEAN`
  Parameters
  Description Clear the plot showing the acquired mean values
  Reply `LIVE: CLEARMEAN executed`
- `LIVE: DEVICE`
  Parameters `<trAddress>`
  Description Activate (select) the transient recorder with the device address `trAddress`
  Reply `LIVE: DEVICE executed`
- `LIVE: MODE`
  Parameters `<0|1|2>`
  Description Set the mode for the data acquisition from the selected transient recorder, `0`=Photon Counting, `1`=Analog Regime, `2`=PR Photon Counting
  Reply `LIVE: MODE executed`
- `LIVE: MEMORY`
  Parameters `<A|B|C|D>`
  Description Set the memory for the data acquisition from the selected transient recorder
  Reply `LIVE: MEMORY executed`
- `LIVE: SHOTLIMIT`
  Parameters `<0|1>`
  Description Set the shot limit for the data acquisition to `0` (4k) or `1` (64k)
  Reply `LIVE: SHOTLIMIT executed`
- `LIVE: DATA?`
  Parameters
  Description Return the current plot data
  Reply `LIVE: DATA BYTES=size<CRLF><valueList>` where `size` is the number of the following bytes containing the `valueList` as a `<SPACE>`-separated list of numbers in `%f` format. Note that as usual a `<CRLF>` will be added.
- `LIVE: CMDLOG?`
  Parameters
  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Live Display* is running under *Licel Main*
  Reply `LIVE: CMDLOG 0|1`
- `LIVE: CMDLOG`
  Parameters `0|1`
  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Live Display* is running under *Licel Main*
  Reply `LIVE: CMDLOG executed`
- `LIVE: QUIT`
  Parameters
  Description exit the program, allowed only when the following key in `TCPIP Live Display.ini` is set:
  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```
  and *TCPIP Live Display* is *not* run within *Licel Main*
  Reply `LIVE: QUIT executed` or `LIVE: QUIT ERROR not allowed`

9.5.4 Control Timing

TCP/IP Server

The basic functions of the *Control Timing* software can be accessed from third party applications via TCP/IP. For this *Control Timing* implements a TCP/IP server listening on a defineable port. To activate the TCP/IP server the following initialization file keys in `Control Timing.ini` have to be aligned:

```
[TCPIP_API]
Active = TRUE
Port = 2088
```

If `Active` is set `TRUE` a listener will be started using the specified TCP/IP port (`Port = 2088`).
If *Control Timing* is run within within a sub panel of *Licel Main* (itself running a TCP/IP API server) and the key `TCPIP_API = TRUE` is set in the Module section of `Licel Main.ini`,

```
[Module1]
Active = TRUE
Path = Control Timing.vi
...
TCPIP_API = TRUE
```

the TCP/IP server of *Control Timing* is accessed automatically via *Licel Main* using a generated listener port. Whenever *Licel Main* receives an unknown TCP/IP API command while the active tab page contains *Control Timing*, the command is passed through to the *Control Timing* TCP/IP API server and handled there.

Command List

The following list contains the supported commands. The commands must be sent with an additional `<CRLF>` (`0x0D0A`) and the responses will end with a `<CRLF>`, as well.

- `TIMING: VER?`
  Parameters
  Description Return the version number as displayed in the Windows title bar
  Reply `VER <version>`
- `TIMING: *IDN?`
  Parameters
  Description Return the *IDN?-information from the controller
  Reply `<IDN>`
- `TIMING: NUMTABS?`
  Parameters
  Description Return the number `numTabs` of visible tab pages
  Reply `TIMING: NUMTABS <numTabs>`
- `TIMING: TAB?`
  Parameters
  Description Return the index `currentTab` of the current tab page within the visible tab pages
  Reply `TIMING: TAB <currentTab>`
- `TIMING: TABNAME?`
  Parameters
  Description Return the name `tabName` of the current tab page
  Reply `TIMING: TABNAME <"tabName">`

- `TIMING: TAB`
  Parameters `<setTab>`
  Description Set the current tab by submitting the index `setTab` of the current tab page within the visible tab pages. The operation is not possible when trigger switches are set which are not allowed in the new tab.
  Reply `TIMING: TAB executed`, `TIMING: TAB Blocked`, or `ERROR TIMING: TAB out of range`
- `TIMING: MODE?`
  Parameters
  Description Return the current `mode`. The current mode is a number containing the switch states of the trigger switches in it's bits: `0x01` Lamp, `0x02` Acquisition, `0x04` Q-Switch, `0x08` Gating, `0x10` Master Trigger
  Reply `TIMING: MODE <mode>`
- `TIMING: MODE`
  Parameters `<mode>`
  Description Set the `mode` i.e. the switch states of the trigger switches in it's bits: `0x01` Lamp, `0x02` Acquisition, `0x04` Q-Switch, `0x08` Gating, `0x10` Master Trigger. This command is not allowed on the *TCP/IP* tab page or if a submitted trigger switch state is not allowed on the current tab.
  Reply `TIMING: MODE executed` or `ERROR TIMING: Mode Blocked`
- `TIMING: TIMES?`
  Parameters
  Description Return the times in microseconds and the frequencies in Hz currently set at the controller. Please note that the returned times correspond to the values shown on the tab *Acquisition Timing*.
  Reply `TIMING: TIMES <Repetition Rate><Lamp to Acquisition> <Acquisition Length> <Acquisition to Q-Switch> <Q-Switch Length> <Start Delay> <External Frequency>`
- `TIMING: TIMES`
  Parameters `<Repetition Rate> <Lamp to Acquisition> <Acquisition Length> <Acquisition to Q-Switch> <Q-Switch Length> <Start Delay> <External Frequency>`
  Description Set the times in microseconds and the frequencies in Hz at the controller. Please note that the times and frequencies correspond to the values shown on the tab *Acquisition Timing*. The reply includes the coerced times and frequencies after executing the TRIGGERTIME command. This command is not allowed on the *TCP/IP* tab page.
  Reply `TIMING: TIMES <Repetition Rate> <Lamp to Acquisition> <Acquisition Length> <Acquisition to Q-Switch> <Q-Switch Length> <Start Delay> <External Frequency>` or `ERROR TIMING: TIMES Blocked`
- `TIMING: CMDLOG?`
  Parameters
  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Control Timing* is running under *Licel Main*
  Reply `TIMING: CMDLOG 0|1`

- `TIMING: CMDLOG`
  Parameters `0|1`
  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Control Timing* is running under *Licel Main*
  Reply `TIMING: CMDLOG executed`
- `TIMING: QUIT`
  Parameters
  Description exit the program, allowed only when the following key in `Control Timing.ini` is set:
  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```
  and *Control Timing* is *not* run within *Licel Main*
  Reply `TIMING: QUIT executed` or `TIMING: QUIT ERROR not allowed`

9.5.5 LaserSync Control

TCP/IP Server

The basic functions of the *LaserSync Control* software can be accessed from third party applications via TCP/IP. For this *LaserSync Control* implements a TCP/IP server listening on a defineable port. To activate the TCP/IP server the following initialization file keys in `LaserSync.ini` have to be aligned:

```
[TCPIP_API]
Active = TRUE
Port = 2088
```

If `Active` is set `TRUE` a listener will be started using the specified TCP/IP port (`Port = 2088`).
If *LaserSync Control* is run within a sub panel of *Licel Main* (itself running a TCP/IP API server) and the key `TCPIP_API = TRUE` is set in the Module section of `Licel Main.ini`,

```
[Module5]
...
TCPIP_API = TRUE
```

the TCP/IP server of *LaserSync Control* is accessed automatically via *Licel Main* using a generated listener port. Whenever *Licel Main* receives an unknown TCP/IP API command while the active tab page contains *LaserSync Control*, the command is passed through to the *LaserSync Control* TCP/IP API server and handled there.

Command List

The following list contains the supported commands. The commands must be sent with an additional `<CRLF>` (`0x0D0A`) and the responses will end with a `<CRLF>`, as well.

- `LSYNC: VER?`
  Parameters
  Description Return the version number as displayed in the Windows title bar
  Reply `VER <version>`
- `LSYNC: *IDN?`
  Parameters
  Description Return the *IDN?-information from the controller
  Reply `<IDN>`

- `LSYNC: MULTIMASTER?`
  Parameters
  Description Return the current MULTIMASTER parameters `masterCycles`, `laser1Omit`, `laser1Offset`, `laser2Omit`, `laser2Offset`, `laser3Omit`, `laser3Offset`, and `triggerMode`. The parameters are the same as those used when directly communicating with the Laser Synchronization Module. Please refer to https://www.licel.com/manuals/LaserSync.pdf#section.6 for details
  Reply `LSYNC: MULTIMASTER <masterCycles> <laser1Omit> <laser1Offset> <laser2Omit> <laser2Offset> <laser3Omit> <laser3Offset> <triggerMode>`
- `LSYNC: MULTIMASTERSTORE?`
  Parameters
  Description Return the MULTIMASTER parameters `masterCycles`, `laser1Omit`, `laser1Offset`, `laser2Omit`, `laser2Offset`, `laser3Omit`, `laser3Offset`, and `triggerMode` stored in the flash memory of the Laser Synchronization Module. The parameters are the same as those used when directly communicating with the Laser Synchronization Module. Please refer to https://www.licel.com/manuals/LaserSync.pdf#section.6 for details
  Reply `LSYNC: MULTIMASTERSTORE <masterCycles> <laser1Omit> <laser1Offset> <laser2Omit> <laser2Offset> <laser3Omit> <laser3Offset> <triggerMode>`
- `LSYNC: MULTIMASTER`
  Parameters `<masterCycles> <laser1Omit> <laser1Offset> <laser2Omit> <laser2Offset> <laser3Omit> <laser3Offset> <triggerMode>`
  Description Set the current MULTIMASTER parameters `masterCycles`, `laser1Omit`, `laser1Offset`, `laser2Omit`, `laser2Offset`, `laser3Omit`, `laser3Offset`, and `triggerMode`. The parameters are the same as those used when directly communicating with the Laser Synchronization Module. Please refer to https://www.licel.com/manuals/LaserSync.pdf#section.6 for details
  Reply `LSYNC: MULTIMASTER executed`
- `LSYNC: MULTIMASTERSUBMIT`
  Parameters
  Description Store the current MULTIMASTER parameters `masterCycles`, `laser1Omit`, `laser1Offset`, `laser2Omit`, `laser2Offset`, `laser3Omit`, `laser3Offset`, and `triggerMode` into the flash memory of the Laser Synchronization Module.
  Reply `LSYNC: MULTIMASTERSTORE executed`
- `LSYNC: CMDLOG?`
  Parameters
  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *LaserSync Control* is running under *Licel Main*
  Reply `LSYNC: CMDLOG 0|1`

- `LSYNC: CMDLOG`
  Parameters `0|1`
  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *LaserSync Control* is running under *Licel Main*
  Reply `LSYNC: CMDLOG executed`
- `LSYNC: QUIT`
  Parameters
  Description exit the program, allowed only when the following key in `LaserSync.ini` is set:
  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```
  and *LaserSync Control* is *not* run within *Licel Main*
  Reply `LSYNC: QUIT executed` or `LSYNC: QUIT ERROR not allowed`

9.5.6 Control APD-PMT

TCP/IP Server

The basic functions of the *Control APD-PMT* software can be accessed from third party applications via TCP/IP. For this *Control APD-PMT* implements a TCP/IP server listening on a defineable port. To activate the TCP/IP server the following initialization file keys in `Control APD-PMT.ini` have to be aligned:

```
[TCPIP_API]
Active = TRUE
Port = 2088
```

If `Active` is set `TRUE` a listener will be started using the specified TCP/IP port (`Port = 2088`).
If *Control APD-PMT* is run within a sub panel of *Licel Main* (itself running a TCP/IP API server) and the key `TCPIP_API = TRUE` is set in the Module section of `Licel Main.ini`,

```
[Module0]
Active = TRUE
Path = Control APD-PMT.vi
...
TCPIP_API = TRUE
```

the TCP/IP server of *Control APD-PMT* is accessed automatically via *Licel Main* using a generated listener port. Whenever *Licel Main* receives an unknown TCP/IP API command while the active tab page contains *Control APD-PMT*, the command is passed through to the *Control APD-PMT* TCP/IP API server and handled there.

Command List

The following list contains the supported commands. The commands must be sent with an additional `<CRLF>` (`0x0D0A`) and the responses will end with a `<CRLF>`, as well. Commands beginning with `APD:` and `PMT:` are supported only if the corresponding hardware capability (CAP?) is available. If a command is not supported because of a missing capability `unknown command` is returned.

- `APD: VER?`
  Parameters
  Description Return the version number as displayed in the Windows title bar
  Reply `VER <version>`

- `PMT: VER?`
  Parameters
  Description Return the version number as displayed in the Windows title bar
  Reply `VER <version>`
- `APD: *IDN?`
  Parameters
  Description Return the *IDN?-information from the controller
  Reply `<IDN>`
- `PMT: *IDN?`
  Parameters
  Description Return the *IDN?-information from the controller
  Reply `<IDN>`
- `APD: NUMTABS?`
  Parameters
  Description Return the number `numTabs` of visible tab pages
  Reply `APD: NUMTABS <numTabs>`
- `PMT: NUMTABS?`
  Parameters
  Description Return the number `numTabs` of visible tab pages
  Reply `PMT: NUMTABS <numTabs>`
- `APD: TAB?`
  Parameters
  Description Return the index `currentTab` of the current tab page within the visible tab pages
  Reply `APD: TAB <currentTab>`
- `PMT: TAB?`
  Parameters
  Description Return the index `currentTab` of the current tab page within the visible tab pages
  Reply `PMT: TAB <currentTab>`
- `APD: TABNAME?`
  Parameters
  Description Return the name `tabName` of the current tab page
  Reply `APD: TABNAME <"tabName">`
- `PMT: TABNAME?`
  Parameters
  Description Return the name `tabName` of the current tab page
  Reply `PMT: TABNAME <"tabName">`
- `APD: TAB`
  Parameters `<setTab>`
  Description Set the current tab by submitting the index `setTab` of the current tab page within the visible tab pages.
  Reply `APD: TAB executed` or `ERROR APD: TAB out of range`
- `PMT: TAB`
  Parameters `<setTab>`
  Description Set the current tab by submitting the index `setTab` of the current tab page within the visible tab pages.
  Reply `PMT: TAB executed` or `ERROR PMT: TAB out of range`

- `APD: NUMDEV?`
  Parameters
  Description Get the current number of APDs
  Reply `APD: NUMDEV <numberAPD>`
- `PMT: NUMDEV?`
  Parameters
  Description Get the current number of PMTs
  Reply `PMT: NUMDEV <numberPMT>`
- `APD: HVSETTING?`
  Parameters `<device>`
  Description Return the current set value of the high voltage `setHV` and the current value of the high voltage switch `ON|OFF` of the APD at the index `device`
  Reply `APD: HVSETTING <setHV> <ON|OFF>`
- `APD: HVSETTING`
  Parameters `<device> <setHV>`
  Description Set or change the current set value of the high voltage `setHV` of the APD at the index `device`
  Reply `APD: HVSETTING executed`
- `APD: HV`
  Parameters `<device> <ON|OFF>`
  Description Switch the high voltage of the APD at the index `device` on or off
  Reply `APD: HV executed`, or `APD: HV cooling must be active` if the cooling is not active while attempting to switch the APD on
- `APD: HV?`
  Parameters `<device>`
  Description Return the current displayed high voltage `HV_val` of the APD at the index `device`
  Reply `APD: HV <HV_val>`
- `APD: COOLING`
  Parameters `<device> <ON|OFF>`
  Description Switch the TEC cooler of the APD at the index `device` on or off
  Reply `APD: COOLING executed`
- `APD: COOLING?`
  Parameters `<device>`
  Description Request the status of the TEC cooler of the APD at the index `device`
  Reply `APD: COOLING <Inactive>` or `APD: COOLING <Active InRange|NotInRange>`
- `APD: MAXVOLT?`
  Parameters `<device>`
  Description Return the allowed maximum high voltage `HV_max` of the APD at the index `device`
  Reply `APD: MAXVOLT <HV_max>`
- `APD: READ`
  Parameters `<device>`
  Description Update the current HV voltage `HV_val` of the APD at the index `device` or of all devices if `device` = -1
  Reply `APD: READ executed`

- `PMT: HVSETTING?`
  Parameters `<device>`
  Description Return the current set value of the high voltage `setHV` and the current value of the high voltage switch `ON|OFF` of the PMT at the index `device`
  Reply `PMT: HVSETTING <setHV> <ON|OFF>`
- `PMT: HVSETTING`
  Parameters `<device> <setHV>`
  Description Set or change the current set value of the high voltage `setHV` of the PMT at the index `device`
  Reply `PMT: HVSETTING executed`
- `PMT: HV`
  Parameters `<device> <ON|OFF>`
  Description Switch the high voltage of the PMT at the index `device` on or off
  Reply `PMT: HV executed`
- `PMT: HV?`
  Parameters `<device>`
  Description Return the current displayed high voltage `HV_val` of the PMT at the index `device`
  Reply `PMT: HV <HV_val>`
- `PMT: MAXVOLT?`
  Parameters `<device>`
  Description Return the allowed maximum high voltage `HV_max` of the PMT at the index `device`
  Reply `PMT: MAXVOLT <HV_max>`
- `PMT: READ`
  Parameters `<device>`
  Description Update the current HV voltage `HV_val` of the PMT at the index `device` or of all devices if `device` = -1
  Reply `PMT: READ executed`
- `APD: CHANNEL?`
  Parameters `<device>`
  Description Return the `channel Description` of the APD at the index `device`
  Reply `APD: CHANNEL <"channel Description">`
- `APD: SERIAL?`
  Parameters `<device>`
  Description Return the `serialNumber` of the APD at the index `device`
  Reply `APD: SERIAL <"serial Number">`
- `PMT: CHANNEL?`
  Parameters `<device>`
  Description Return the `channel Description` of the PMT at the index `device`
  Reply `PMT: CHANNEL <"channel Description">`

- `PMT: SERIAL?`
  Parameters `<device>`
  Description Return the `serialNumber` of the PMT at the index `device`
  Reply `PMT: SERIAL <"serial Number">`
- `APD: CMDLOG?`
  Parameters
  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Control APD-PMT* is running under *Licel Main*
  Reply `APD: CMDLOG 0|1`
- `APD: CMDLOG`
  Parameters `0|1`
  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Control APD-PMT* is running under *Licel Main*
  Reply `APD: CMDLOG executed`
- `PMT: CMDLOG?`
  Parameters
  Description Request whether or not the TCP/IP commands are currently logged. Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Control APD-PMT* is running under *Licel Main*
  Reply `PMT: CMDLOG 0|1`
- `PMT: CMDLOG`
  Parameters `0|1`
  Description Switch the TCP/IP command logging on (`1`) or off (`0`). Command logging is supported in Licel's Windows and Linux applications and by default in the LabVIEW sources. The command is not supported when *Control APD-PMT* is running under *Licel Main*
  Reply `PMT: CMDLOG executed`
- `APD: QUIT`
  Parameters
  Description exit the program, allowed only when the following key in `Control APD-PMT.ini` is set:
  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```
  and *Control APD-PM* is *not* run within *Licel Main*
  Reply `APD: QUIT executed` or `APD: QUIT ERROR not allowed`
- `PMT: QUIT`
  Parameters
  Description exit the program, allowed only when the following key in `Control APD-PMT.ini` is set:
  ```
  [TCPIP_API]
  ...
  AllowQUIT = TRUE
  ```
  and *Control APD-PM* is *not* run within *Licel Main*
  Reply `PMT: QUIT executed` or `PMT: QUIT ERROR not allowed`

9.6 Assign Detector Voltages to Transient Recorders

The high voltages of APD and PMT detectors set by *Control APD-PMT* may be assigned to transient recorder channels defined by *TCPIP Acquis*, *TCPIP MPush Acquis*, or *M-Acquis*.

- When the HV value read from the APT/PMT controller has changed, the current HV value is written to the file `DetectorTR_Assignment.ini`. The detector description from the initialization file `Control APD-PMT`n`.ini` is written to `DetectorTR_Assignment.ini`, as well:

  ```
  [DetectorIdentifier]
  Description = "my description"
  HV = HV_value
  ```

  The `DetectorIdentifier` is either

  - `[APD_S/N]` or `[PMT_S/N]` if the corresponding serial number `S/N` of an APD or PMT, respectively, is defined, or
  - `[APDi_IPAddress_Port]` or `[PMTi_IPAddress_Port]` if no serial number is available. `i` is the APD or PMT index, `IPAddress` and `Port` are given by the TCP/IP connection settings of the detector controller.

- The assignment of a transient recorder channel to a detector can easily be done in the configuration dialog.
- When starting or restaring an acquisition in *TCPIP Acquis*, *TCPIP MPush Acquis*, or *M-Acquis* the last written HV values from `DetectorTR_Assignment.ini` will be assigned to the transient recorder channels following the definitions from above.

The following text shows two examples

```
[PMT1_10.49.234.236_2055]
HV = 850.0

[PMT_ab23x07]
HV = 800.0
```

9.7 Analysis Example: Gluing Analog and Photon Counting Data

Abstract

The algorithm for combining analog and photon counting data (gluing) is described. A discussion when the signals need to be combined is followed by stepwise procedure to do this with real data.

9.7.1 Introduction

The Licel transient recorder systems have a parallel analog and photon counting detection chain. The combination of both signals gives the high linearity of the analog signal for strong signals and the high sensitivity of the photon counting for weak optical signals. The integration of both detection mechanism into a single device avoids ground loops and other problems that make the combination otherwise cumbersome. The main idea of the signal combination is that there is a region where both signals are valid and have a high signal to noise ratio. For typical Mini-PMT that region extends from 0.5 to 10 MHz in the photon counting. To combine (glue) both signals, the photon counting needs a dead time correction. There are two typical dead-time scenarios, while the Licel photon counter can be best described as nonparalyzable.

9.7.2 Paralyzable System

$$N = S\exp(-S\tau_d) \tag{9.10}$$

Where:
N - is the observed count rate
S - is the true count rate
τ_d - is the system dead time

9.7.3 Nonparalyzable System

$$N = \frac{S}{1 + S * \tau_d} \tag{9.11}$$

N - is the observed count rate
S - is the true count rate
τ_d - is the system dead time

While the paralyzable case is nonlinear equation, the nonparalyzable case can be easily inverted to

$$S = \frac{N}{1 - N * \tau_d} \tag{9.12}$$

As both cases are only a theoretical model, they are valid for lower count rates but fail when $S * \tau_d$ becomes larger than one. From a numerical point of view Eq. 9.12 can be only applied to a signal as long as

$$S < \frac{1}{\tau_d} \tag{9.13}$$

As an example the correction factor for a time constant of 4ns and a observed count rate of 5 MHz is 1.02. As typical averaged maximum observed count rate is 160MHz the correction factor would be 2.77. This would imply an maximum count rate of 470MHz. The glued profiles however show a virtual count rate in the 2GHz region for a 20mV peak.

9.7.4 The glueing algorithm

In the valid region of both signals between the lower toggle rate (typical 0.5MHz) and the upper toggle rate (typical 10MHz) one seeks the linear regression coefficients to transfer the analog data into photon counting data:

$$\sum_{i=1}^{n} (PC(z_i) - (a * Analog(z_i) + b))^2 = min \tag{9.14}$$

The coefficients a and b are applied to the analog signal and above the upper toggle rate the scaled analog is used and below the photon counting data.

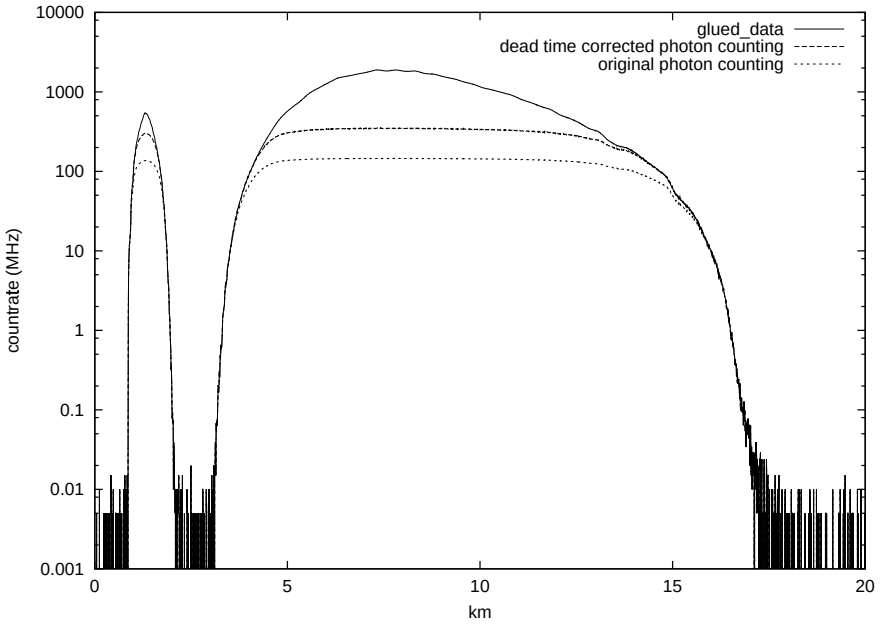


Figure 9.1: Glued data

The zoomed plot shows that the dead time correction function is valid up to 130 MHz.

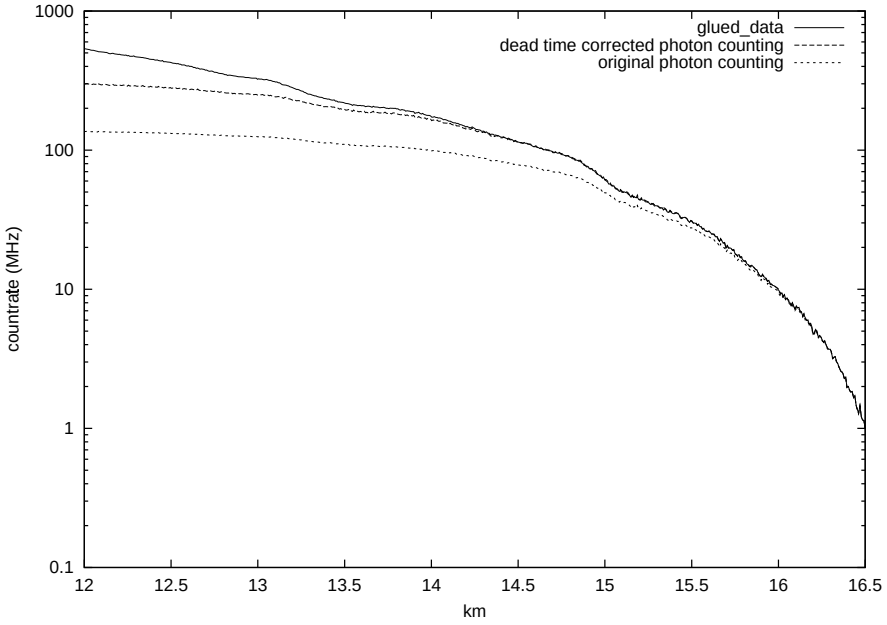


Figure 9.2: Zoomed plot

If one varies the upper toggle frequency between 5 and 10 MHz the standard deviation for the signal maximum is only 3MHz or 0.1%. This proves the numerical stability of the proposed algorithm.
The figure below shows the necessity of applying the dead-time correction first. Without correction the signal maximum becomes stronger dependent from the max. toggle rate.

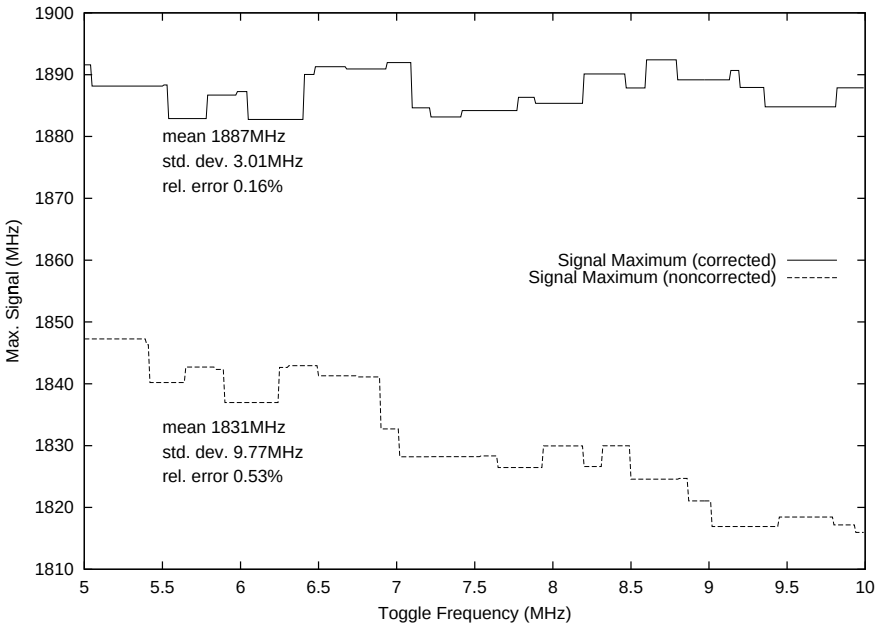


Figure 9.3: Signal maximum for different max. toggle frequencies without dead-time correction

Figure 9.4 demonstrates the advantages of the photon counting in the low light level region. While the analog signal shows the noise coming from the ADC, the photon counting is still able to follow the input signal and extends the dynamic signal range from the analog signal by another 2 orders of magnitude.

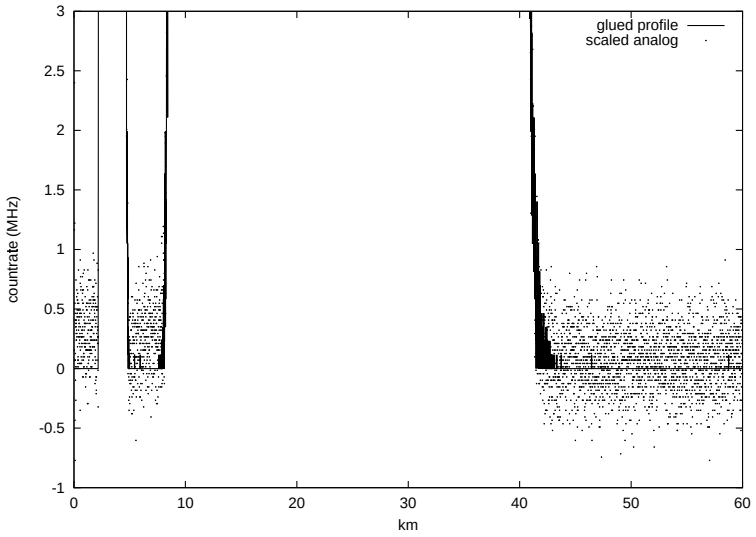


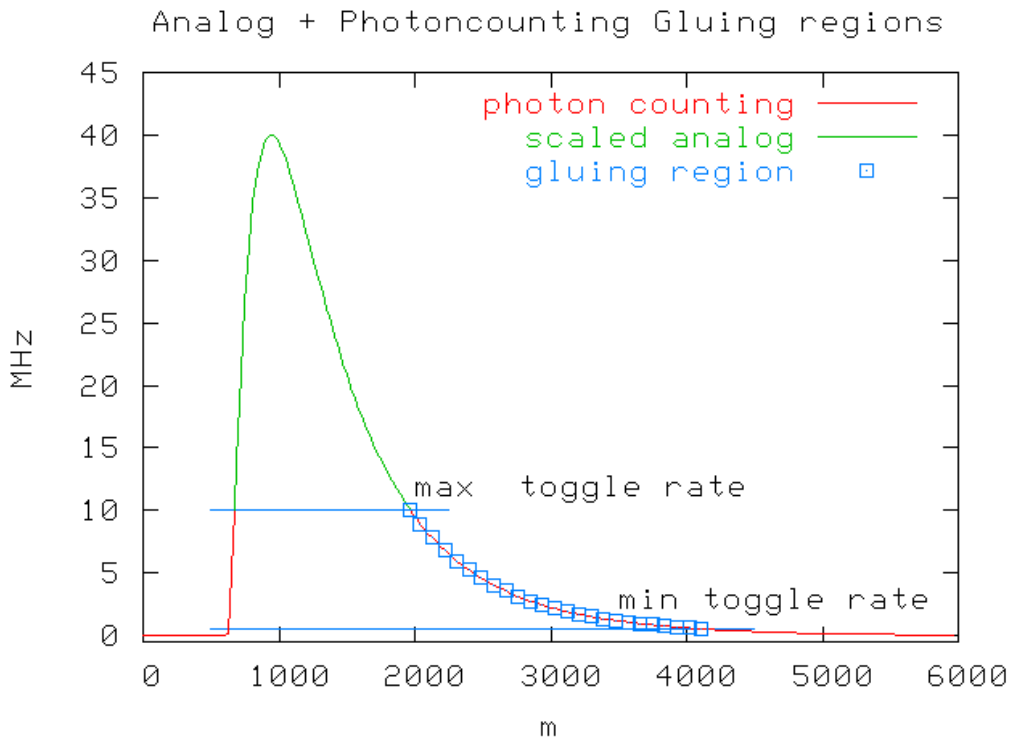
Figure 9.4: Increased dynamic range under low light level conditions

9.7.5 Gluing strategy

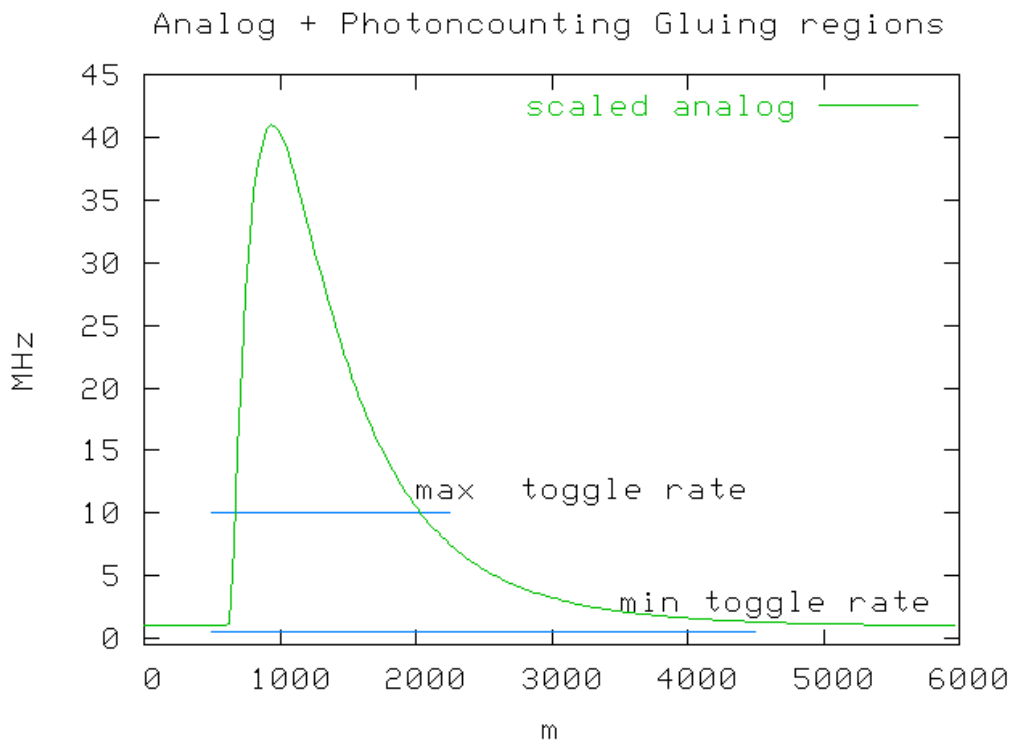
In principle one should glue two signals only if it is necessary. The only scenario when one really need to glue is when:

1. the peak value of the dead-time corrected photon counting is above the maximum toggle rate and
2. the background of the dead-time corrected photon counting is below the minimum toggle rate.

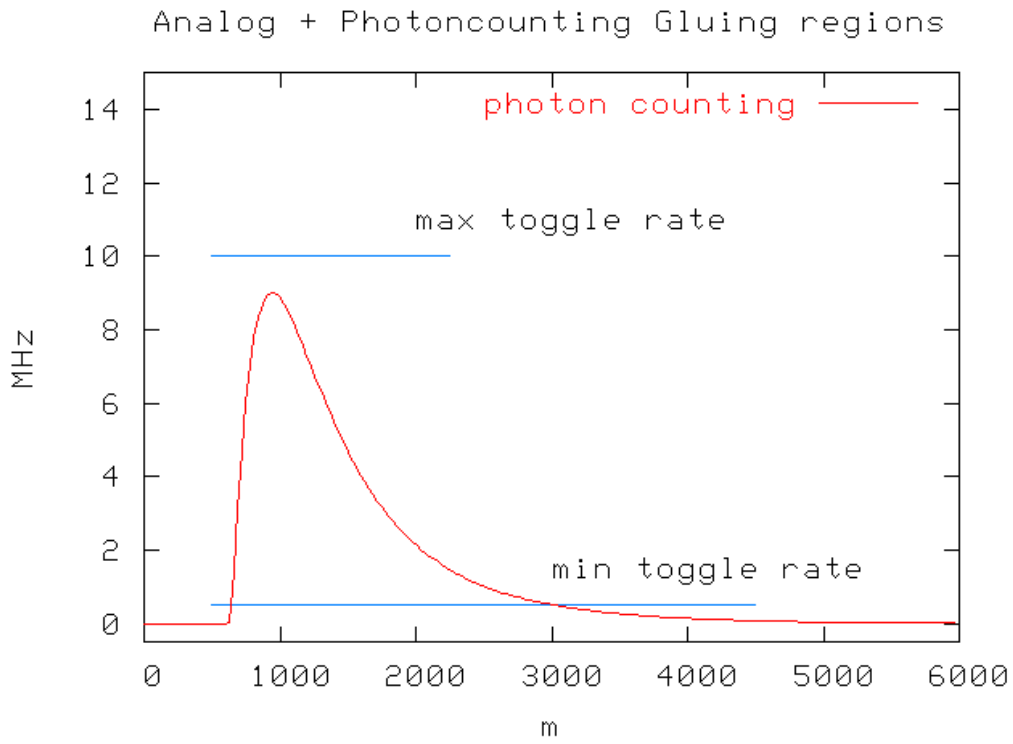
This situation is shown below:



If one assumes that the analog is valid enough to compute a regression curve then there is no need to compute a regression if the photon counting background exceeds the minimum toggle rate. In this case one can use the scaled analog.



If the peak count rate does not exceed the max. toggle rate there is no need to glue either and the dead-time corrected photon counting should be used.



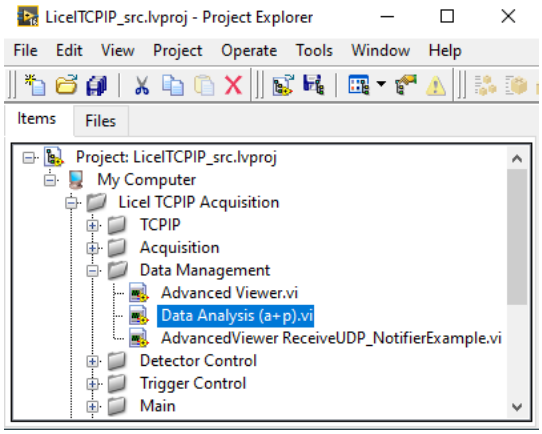
The use of a glued profile instead of a pure photon counting profile if the peak value is only slightly above the max. toggle rate. say at 12 MHz for 10MHz max. toggle rate could also be avoided.

9.7.6 Tutorial

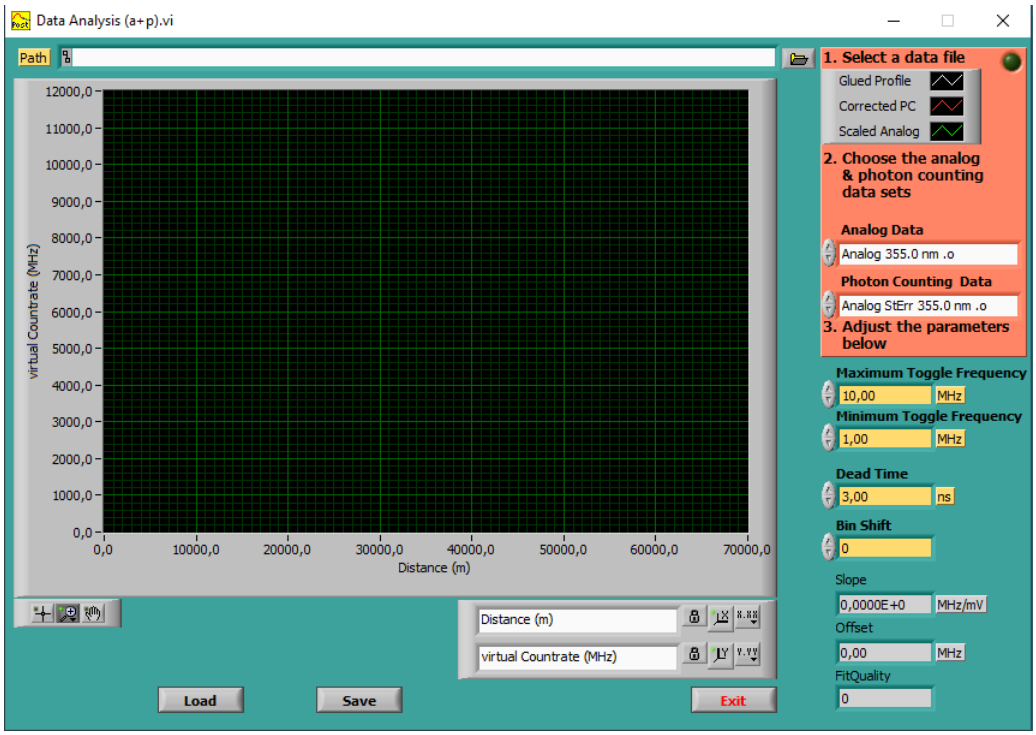
Licel provides a sample code in LabVIEW for combining analog and photon counting data. The sample code assumes that the provided data has been previously recorded with the Acquis Software. One needs a LabVIEW license to look into the code. Reuse of this code in your projects is desired and permitted. The compiled Windows application is included in the Windows installer.

Loading the VI

- If you are using the LabVIEW sources open the post analysis utility from the LabVIEW project by navigating to the corresponding entry *Data Analysis (a+p).vi* and double-clicking it.

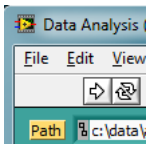


- If you installed the Windows applications please start the program by selecting the corresponding entry in the Licel section of the Windows Start menu. The Windows application will automatically start.



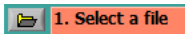
Start the VI in LabVIEW

If you opened the VI in LabVIEW please press the run button in the upper left corner.

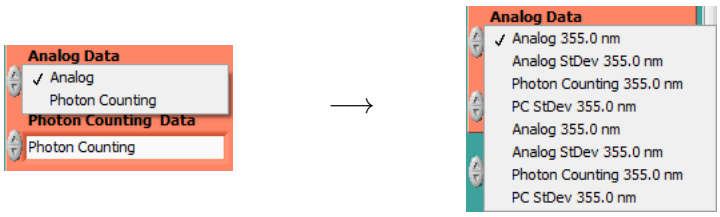


Selecting a data file

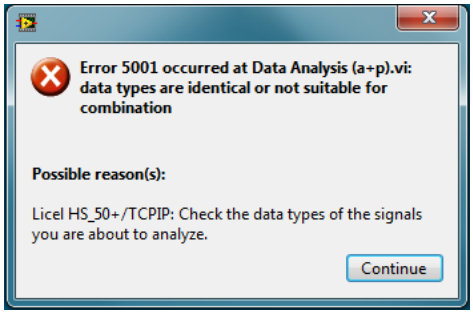
First click the browse button



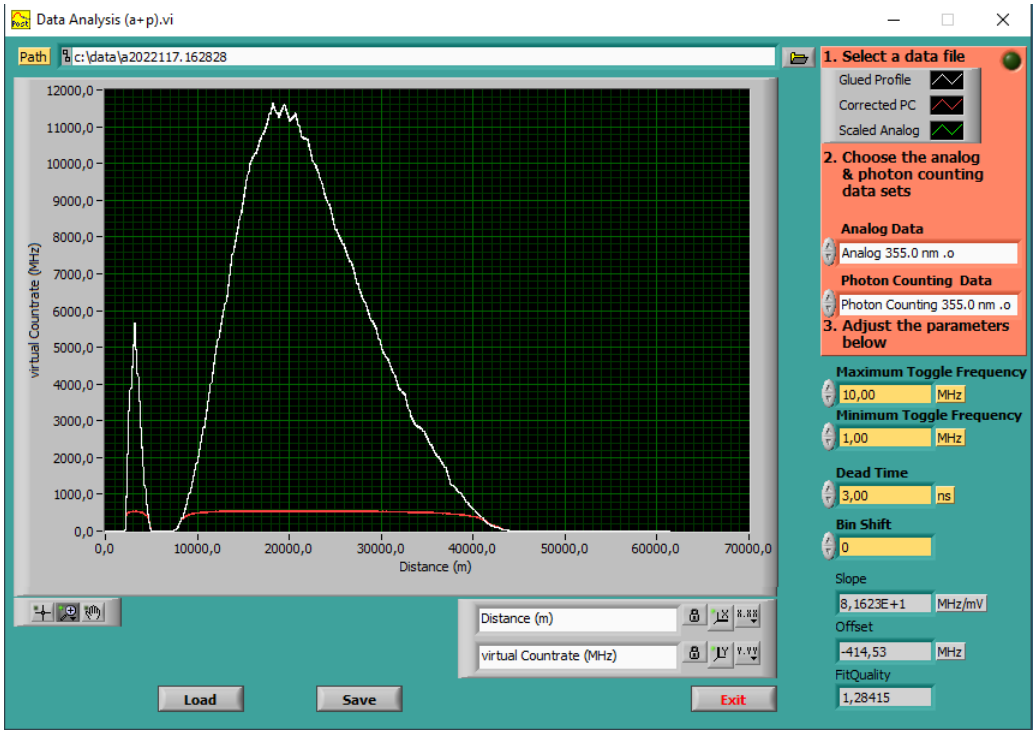
in the upper right part of the program window and select a data file that has previously been recorded with the Acquis-Module. At https://www.licel.com/download/gluetestfile.zip one can find the data file which has been used for this demonstration.
First off all the program will read information about all datasets and the dataset selection lists *Analog Data* and *Photon Counting Data* will update as for the selection lists in the Acquis-Module and the Viewer:



This will help to make sure that the first and the second data sets are really analog and photon counting data sets, respectively. If the selected data sets are not suitable for a glueing procedure an error message dialog will be displayed:



In that case you will have to correctly select the data sets for glueing and press the run button again. Finally one should see the following curves:



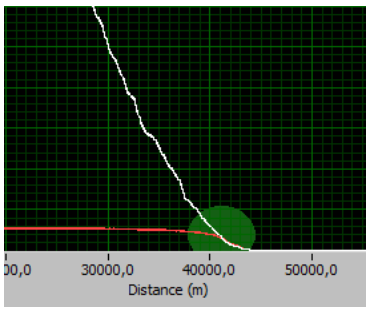
The white curve shows the combined signal.

Bin shift

The analog and the photon counting data has a fixed shift between them. This is a result of two factors

1. Analog Bandwidth, the preamplifier contains an antialias filter which has a bandpass of half the sampling frequency this delays the analog signal with respect to the photon counting by 2 bins
2. ADC pipelining, modern ADCs sample the voltage in a multiple step process so that the sample result will be available several clock cycles later after the actual sampling took place.

To demonstrate this zoom into the profile



There is a shift of the scaled analog signal versus the photon counting data (the green vs. white curve)

Setting the bin shift to 2 will result in a much more perfect match.

- For 12 bit TR shipped between before 2002 this would be 2-3.
- For 12 bit TR shipped between 2002 and 2018 this would be 9.
- For 16 bit TR shipped between 20010 and 2018 this would be 16.
- For TR shipped since this would be 0-1.

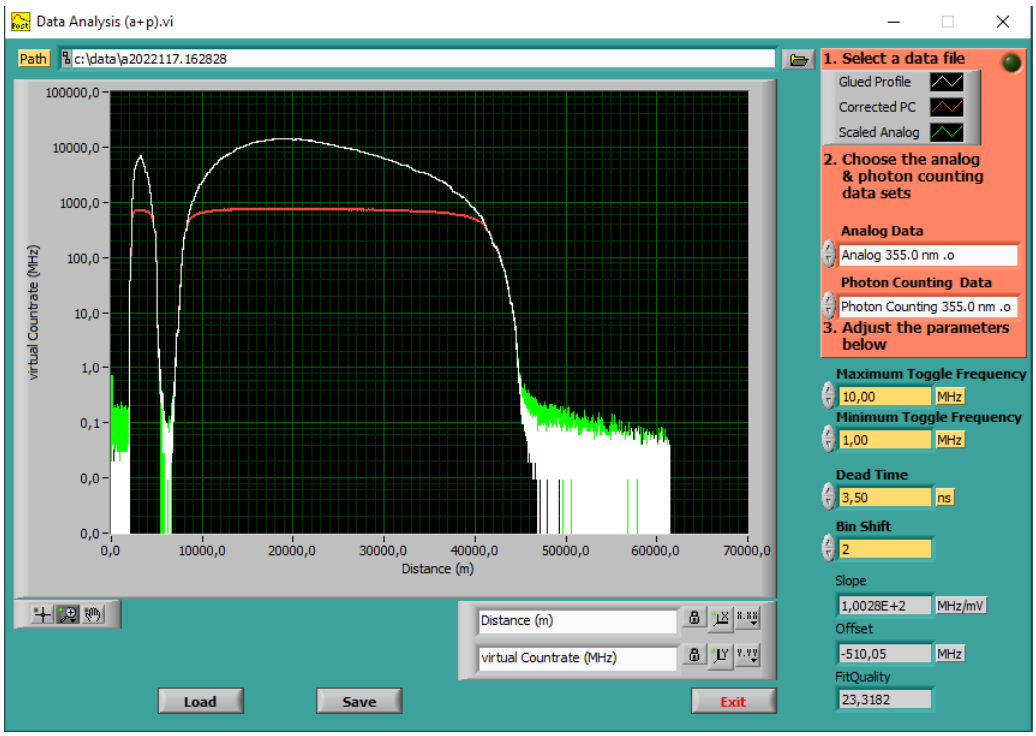
The binshift is a transient property which can be retrieved with the `TRTYPE?` command. Its stored in the data files see the `Data File format` appendix.

Photon counting dead-time correction

The default value of 3.0 ns (120 MHz) is rather conservative approach for the dead-time correction. Lowering this value increases the dead-time correction. In the region above the max toggle rate a perfect dead-time correction will show a longer region where the glued curve and the dead-time corrected photon counting coincide.

5 Orders of magnitude

Changing the y-scale from linear to logarithmic reveals the potential of this signal combination.



The red curve shows that the photon counting becomes nonlinear and saturates. The green curve shows that signals which are close to the analog baseline are difficult to distinguish. But the combination of both signal prevents the nonlinearity for strong signals and gets the good baseline from the photon counting.

Save and Load

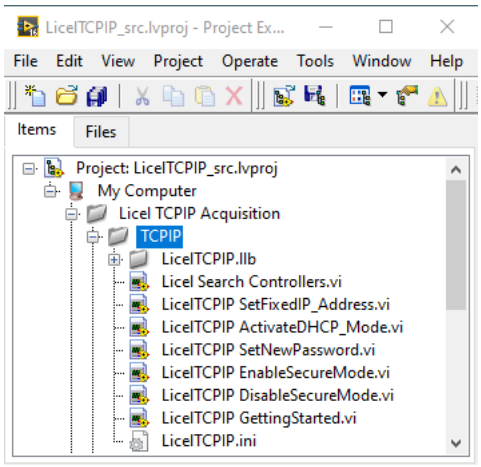
With the button Save a gluing result can be saved to a text file. Using Load such a text file can be reloaded.

Next steps

Code similar to that in *data analysis (a+p)* needs to be integrated into the data retrieval software. For this the core procedure *Postan.llb\Postan DataAnalysisCore.vi* can be found in the LabVIEW sources. Experience shows that recording background file without a laser signal and subtracting the averaged background from real signals will improve the analog background flatness and give more consistent gluing results. Once the transfer coefficients are found one could use them instead of searching in every signal for a new set of coefficients. The coefficient should stay constant if the detector has the same applied high voltage.

9.8 LabVIEW TCPIP Driver VIs

In this section an overview about the provided LabVIEW TCPIP Driver VIs is given.
The TCPIP driver VIs are located into the folder `source\LicelTCPIP.llb` of the Licel's LabVIEW source distribution. In the LabVIEW source project `project\LicelTCPIP_src.lvproj` they can be found in the virtual folder `Licel TCPIP Acquisition\TCPIP`.



Please note that the VIs described below are located in the directory `source\LicelTCPIP.llb`. In older versions of the Licel TCPIP Acquisition software before 2.70.01 the VIs were located in the LabVIEW LLB `source\Licel TCPIP.llb` and had slightly different names.
The top level VIs described in the next subsection are directly located in the virtual folder while the low-level driver VIs described in the further subsections can be found in the virtual project subfolder `LicelTCPIP.llb`.

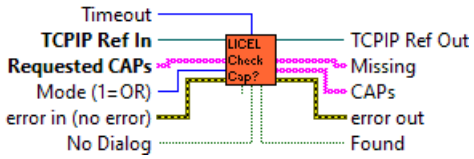
9.8.1 Top Level VIs

LicelTCPIP ActivateDHCP_Mode.vi

This VI is used to activate DHCP for the transient recorder controller.
This VI uses the default password **Administrator** and the default port **2055**. If the port has been changed, you must change the **current port** to the proper value. The **DHCP port** is the port that will be used for DHCP communication. After DHCP mode has been set, communication will be lost until the acquisition computer is configured for DHCP communication as well.

LicelTCPIP CheckCapabilities.vi

Check the controller for the Requested CAPs. Generates a warning if not found.



LicelTCPIP DisableSecureMode.vi

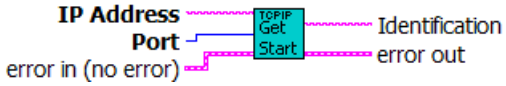
This VI is used to disable the Secure Mode of the Licel Ethernet Controller. The initialization file LicelTCPIP.ini is modified to allow future access without using the Secure Mode login.

LiceITCPIP EnableSecureMode.vi

This VI is used to enable the Secure Mode of the Licel Ethernet Controller. The initialization file LiceITCPIP.ini is modified to allow future access using the Secure Mode login. This file should be copied to the same directory where Licel TCPIP.llb resides on all PCs from where access is allowed.

LiceITCPIP GettingStarted.vi

This VI gets the identification information from the transient recorder controller.



LiceITCPIP SetFixedIP_Address.vi

This VI is used for setting the new IP configuration for the transient recorder controller.

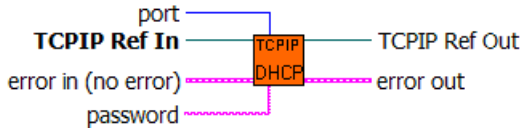
LiceITCPIP SetNewPassword.vi

This VI is used for setting the new password for the Licel Ethernet Controller.

9.8.2 Controller Related VIs

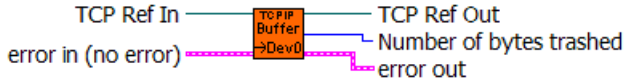
LiceITCPIP ActivateDHCP.vi

This VI is used to activate the DHCP mode of the transient recorder controller.
In order to do so, the user must enter the proper password and port number for the controller. After DHCP mode has been set, communication will be lost until the acquisition computer is configured for DHCP communication as well.



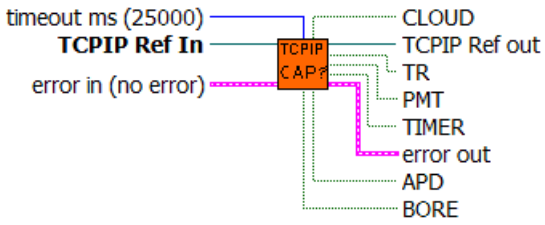
LiceITCPIP DumpTCPIP_Buffer.vi

This VI empties the TCPIP buffer by reading all the data that is available in the buffer.
The **Number of bytes trashed** shows how many bytes were read from the buffer and disposed of.



LiceITCPIP GetCapabilities.vi

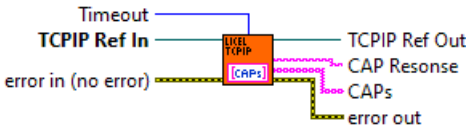
This vi returns the controller's capabilities.



Each of the capabilities TR, PMT, TIMER, TIMERM, TIMER1, TIMER2, APD, DRIVE, POW, CLOUD, BORE, BATT, and APDCOOL corresponds to an own boolean output value.

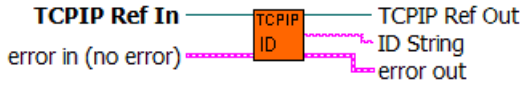
LicelTCPIP GetID.vi

This vi returns the controller's capabilities as a string array.



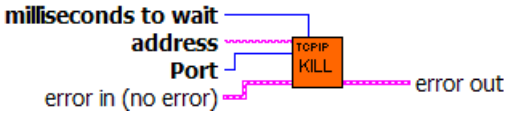
LicelTCPIP GetID.vi

gets the identification string from the transient recorder controller.



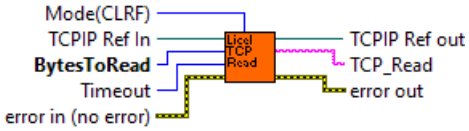
LicelTCPIP KillSockets.vi

This VI opens a new connection to the TR and sends the command to close down and reset all TCPIP connections. After doing this, the VI shuts down its TCPIP connection and waits the specified number of milliseconds, **milliseconds to wait**, before returning.



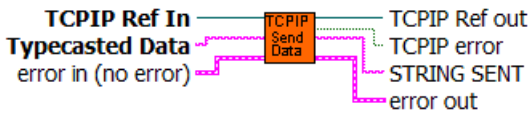
LicelTCPIP Read.vi

Wrapper around LabVIEW's TCP Read, adds logging feature if enabled.



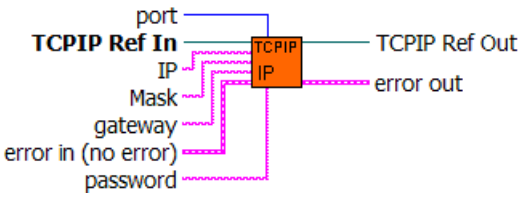
LicelTCPIP SendData.vi

adds a `CRLF` to the end of the string and sends it via TCPIP using the TCPIP reference input



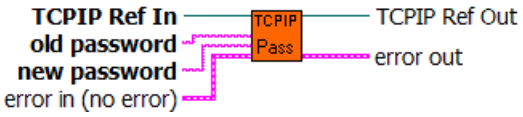
LiceITCPIP SetIP_Parameter.vi

This VI is used to configure the transient recorder controller for static IP communication. With it, the values of the **IP** address, **port** number, subnet **mask**, and **gateway** can be set.



LiceITCPIP SetPassword.vi

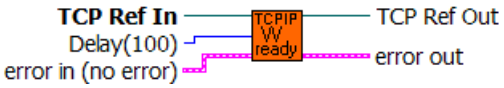
This VI is used for setting the password of the transient recorder controller.
This password must be given in order to change the IP configuration of the controller.



9.8.3 Transient Recorder

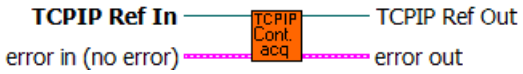
LiceITCPIP WaitForReady.vi

Waits for return of the device from the armed state. If the waiting time is longer than the time specified by delay than the device remains armed and will be return to the idle state with next reading of binary data



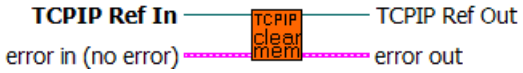
LiceITCPIP ContinueAcquisition.vi

Continues the recording process for the specified device without reinitializing the memory.



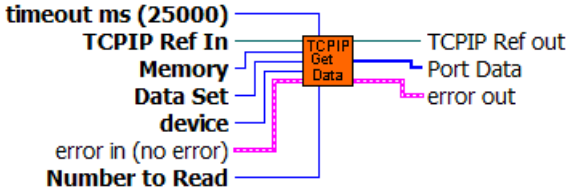
LiceITCPIP ClearMemory.vi

Clears all memories of the specified device.



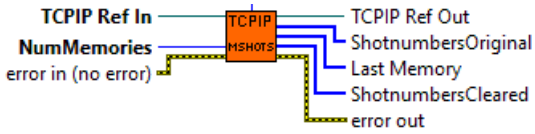
LiceITCPIP GetDatasets.vi

is a vi for reading raw data sets (analog LSW, analog MSW or photon counting and upper word photon counting) from the specified device.



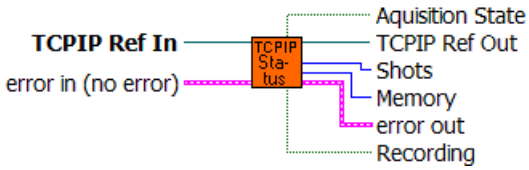
LiceITCPIP GetShots.vi

Get the acquired shots from the selected transient recorders.
The VI uses the `MSHOTS?` command. It will analyze the reply and put the returned shot numbers to the output arrays.



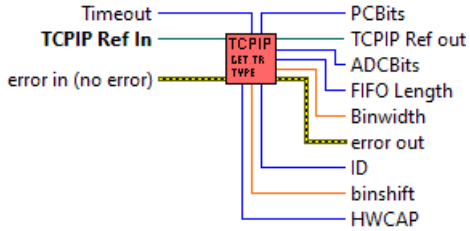
LiceITCPIP GetStatus.vi

Returns the status information for the specified device (cycles,memory,acquisition state and whether the device is just recording).
If an error parsing the status information occurs, the VI returns an error 5765.



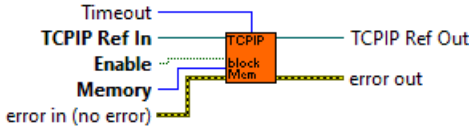
LiceITCPIP GetTRType.vi

Get transient recorder type information about the selected transient recorder:



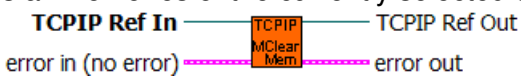
LiceITCPIP MemoryBlock.vi

Block the memory from receiving trigger pulses.



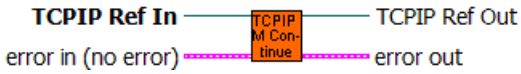
LiceITCPIP MultipleClearMemory.vi

Clears all memories of the currently selected devices.



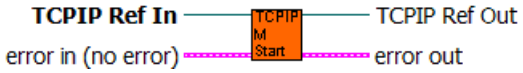
LiceITCPIP MultipleContinueAcqusition.vi

The acquisition process of the selected multiple devices will be restarted without clearing their memories.



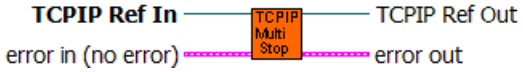
LiceITCPIP MultipleStart.vi

The acquisition process will be started after the next received trigger for multiple devices



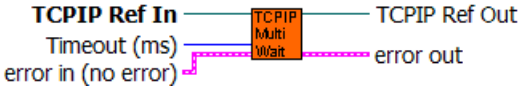
LiceITCPIP MultipleStopAcqusition.vi

The acquisition process will be stoped after the next received trigger for multiple devices



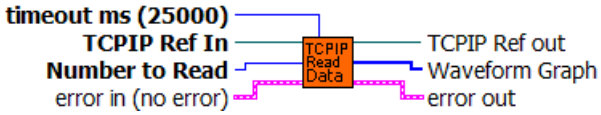
LiceITCPIP MultipleWaitForReady.vi

The vi waits until all devices returned from the armed state.



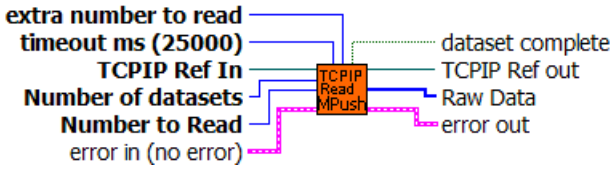
LiceITCPIP ReadData.vi

This VI waits until the number of scans defined by **Number to Read** is available and reads them or returns a timeout error if the **timeout ms** is exceeded.



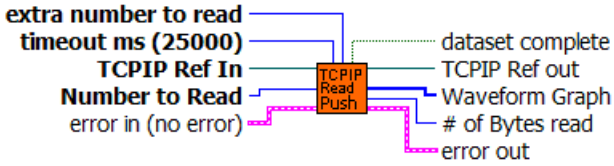
LiceITCPIP ReadMPushedData.vi

This VI reads the pushed data from multiple transient recorders at once. The data from the various transient recorders is concatenated together and must still be separated.



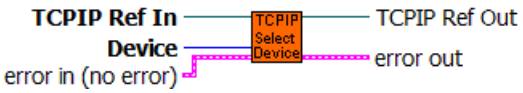
LiceITCPIP ReadPushedData.vi

This VI is used for reading a single pushed data set.



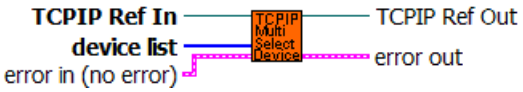
LiceITCPIP SelectDevice.vi

selects the device specified by the input **device number**. Selecting a device makes it active for all future commands that do not have a required **device number** input. The previously selected devices become deselected when this command is issued.



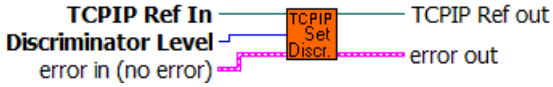
LiceITCPIP SelectMultipleDevices.vi

This VI is used to select multiple transient recorders.
The devices corresponding to the numbers in the **device list** array will be selected which means that they will become sensitive to all future commands that do not require a **device number** input. The devices will be deselected if another **select** command is issued.



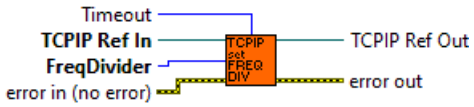
LiceITCPIP SetDiscriminatorLevel.vi

Set the discriminator level between 0 and 63 for the selected transient recorders.



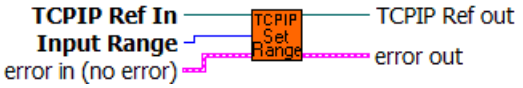
LiceITCPIP SetFrequencyDivider.vi

Set the current frequency divider of the selected transient recorder. The frequency divider is supported only if the transient recorder's HWCAP supports bit 0x40.



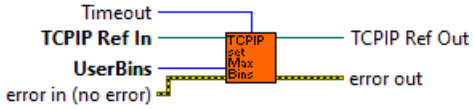
LiceITCPIP SetInputRange.vi

The vi changes the input voltage range.



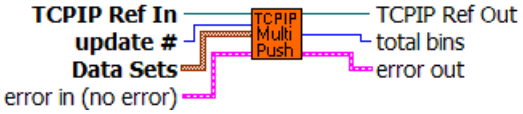
LiceITCPIP SetMaxBins.vi

Set the maximum bin number the transient recorder will acquire



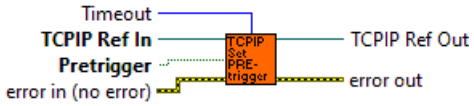
LiceITCPIP SetMultiplePushMode.vi

This VI is used to start the push mode for one or more devices.
This VI takes the **Data Sets** information and the **update #**, which is the number of laser pulses to acquire, as input parameters. Based upon these inputs, the VI generates and sends a command to start the push mode for the transient recorders specified by **Data Sets**.



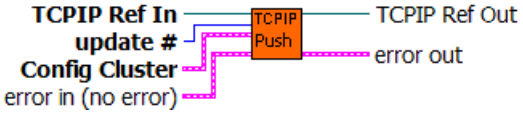
LiceITCPIP SetPretrigger.vi

Set the pretrigger of the selected transient recorder.



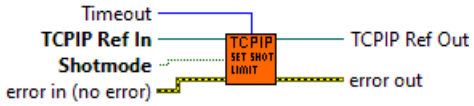
LiceITCPIP SetPushMode.vi

sets the push mode for the currently selected transient recorder.



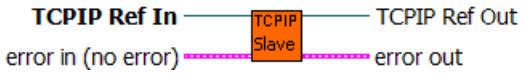
LiceITCPIP SetShotLimit.vi

Set the shot limit of the selected transient recorder



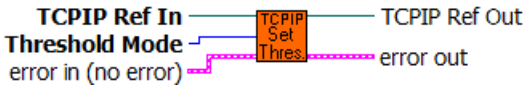
LiceITCPIP SetSlaveMode.vi

This VI stops the push mode and sets the transient recorder controller back in to the slave mode.



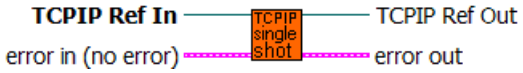
LiceITCPIP SetThresholdMode.vi

Set Threshold Mode sets the scale of the discriminator level. In the low threshold mode the discriminator level 63 corresponds to -25mV while in the high threshold mode it corresponds to -100mV.



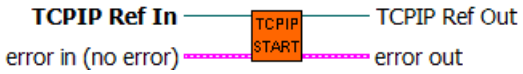
LiceITCPIP SingleShot.vi

Acquires one shot with the currently selected device.



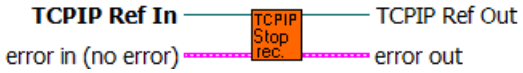
LiceITCPIP Start.vi

starts the currently selected transient recorder.



LiceITCPIP StopAcqusition.vi

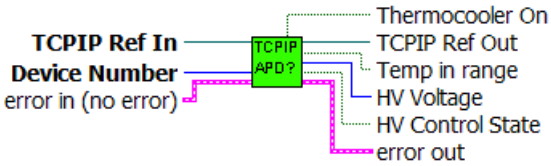
This VI stops the acquisition process after the next received trigger.



9.8.4 APD

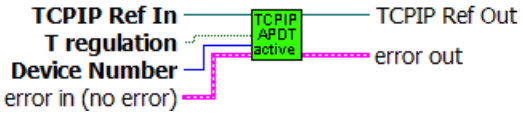
LiceITCPIP APD_GetStatus.vi

This VI gets the status of the APD with the corresponding device number.
The values that are returned are the
HV Voltage : this is the actual gain voltage
On : this boolean is true if the gain voltage power supply is on,
otherwise it is false **control state** : if true, the APD is being controlled remotely,
if false, then the APD is being controlled locally
T regulation: if true, then the cooling has been activated
if false, then the cooling is inactive; i.e. passive



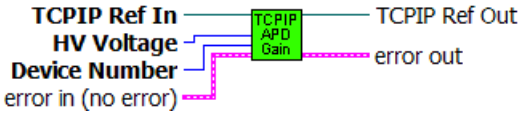
LiceITCPIP APD_SetCoolingState.vi

This VI sets the cooling state for the APD with the corresponding device number.
True = the current to the peltier cooling will be activated
False = the cooling will not be activated. Only passive cooling occurs.



LiceITCPIP APD_SetGain.vi

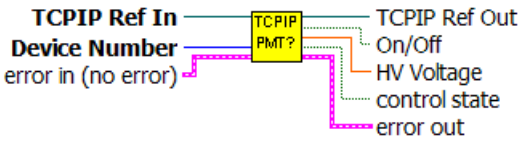
Sets the Gain Voltage for the APD specified by the **Device Number** to the value specified by **HV Voltage**.



9.8.5 PMT

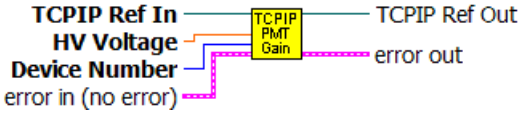
LiceITCPIP PMT_GetStatus.vi

This VI gets the status of the PMT with the corresponding device number.
The values that are returned are the
HV Voltage : this is the actual gain voltage
On : this boolean is true if the gain voltage power supply is on, otherwise it is false
control state : if true, the PMT is being controlled remotely, if false, then the PMT is being controlled locally



LiceITCPIP PMT_SetGain.vi

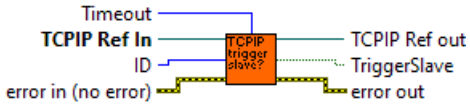
Sets the Gain Voltage for the PMT specified by the Device Number to the value specified by HV Voltage



9.8.6 Trigger

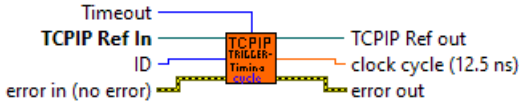
LiceITCPIP GetTriggerSlave.vi

Find out whether or not the timing board is a slave trigger board i.e. it receives a trigger by another build-in sub board.



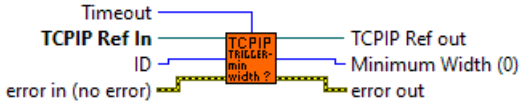
LiceITCPIP GetTriggerTimingCycle.vi

Read the clock cycle time for the board specified by ID



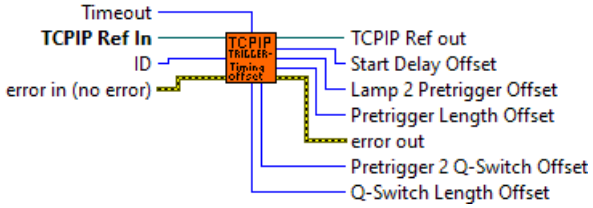
LiceITCPIP GetTriggerTimingMinWidth.vi

Read the minimum width of the Pretrigger and Q-switch length of the board specified by ID; the value is given in clock cycles.



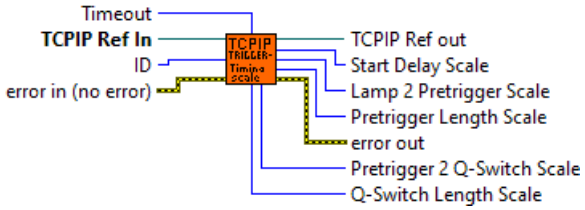
LiceITCPIP GetTriggerTimingOffset.vi

Read the offsets for the times for the Lamp, pretrigger delay, pretrigger length, Q-Switch delay and Q-switch length for the board specified by ID; all offsets are given in clock cycles.



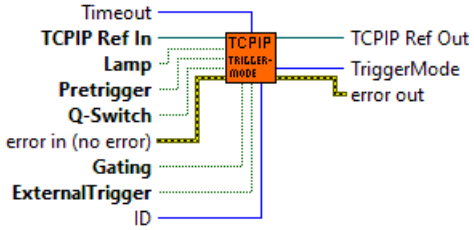
LiceITCPIP GetTriggerTimingScale.vi

Read the scales for the times for the Lamp, pretrigger delay, pretrigger length, Q-Switch delay and Q-switch length; all scales are given in clock cycles.



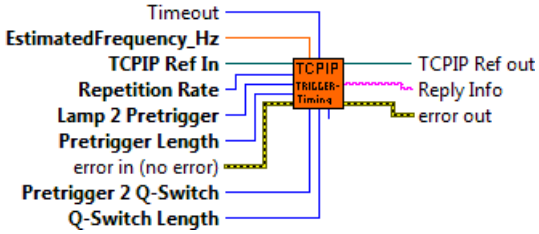
LiceITCPIP SetTriggerMode.vi

The vi enables or disables the trigger mode for the Lamp, Pretrigger (Acquisition Trigger), Q-Switch, and Gating. The user can also switch between the internal and an external trigger using the External-Trigger control.



LiceITCPIP SetTriggerTiming.vi

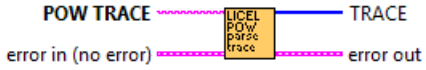
The vi allows the user to set the times in ns for the Lamp, pretrigger delay, pretrigger length, Q-Switch delay and Q-switch length.



9.8.7 Power Meter

LiceITCPIP POW_ParseTrace.vi

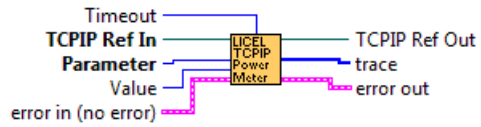
Parses the string reply of the **POW** TRACE command and returnes the data as an array.



LiceITCPIP PowerMeter.vi

The vi sends the POW command with the **Parameter** (0: START, 1: STOP, 2: RESET) to the controller.

START	causes the controller to send power meter data wheneverr receiving a trigger
STOP	stops transferring data.
CHANNEL	sets the ADC channel
TRACE	starts a single pulse acquisition and returns the last trace of data points

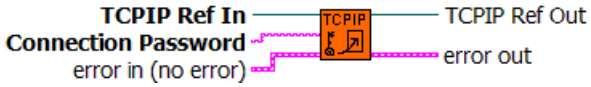


An error is generated if the vi does not receive the appropriate reply (`POW <Parameter>executed` or for TRACE a sequence of decimal string numbers).

9.8.8 Network Security

LicelTCPIP LoginSecureMode.vi

Send the LOGON command to work in secure mode. Reads a string from TCPIP, attempts to convert the string to 2 U32 numbers used to encrypt the password to 2 output U32 numbers using the Blowfish encryption algorithm. These output numbers are converted to a hexadecimal string to be used in the LOGON command. If the LOGON command fails the controller will close the connection without any notification.

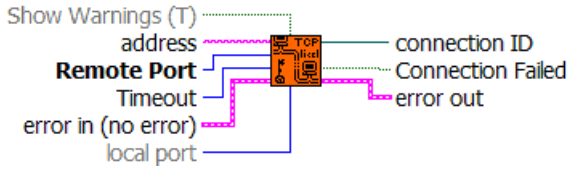


LicelTCPIP OpenSecureMode.vi

Open a TCP/IP connection to the Licel controller in secure mode. The vi tries to open the initialization file LicelTCPIP.ini to read the values for the keys UseSecureMode and SecureModePWD from the SecureMode section:

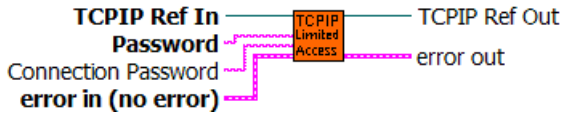
```
[SecureMode]
UseSecureMode=TRUE
SecureModePWD=ConnectMe
```

If the initialization file is found and UseSecureMode is true and SecureModePWD is found the vi will send the password using the LOGON command (Licel TCPIP Login Secure Mode.vi). Otherwise just the TCP/IP connection will be opened.



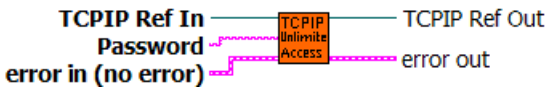
LicelTCPIP SetAccessLimited.vi

Enables the limited access to the controller, i.e. activates the secure mode. Access is granted only for IP addresses as specified with the WHITELIST command. Moreover the connection password is specified.



LicelTCPIP SetAccessUnlimited.vi

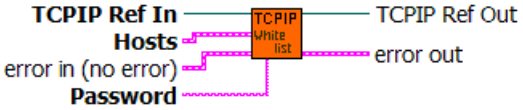
Disables the limited access to the controller, i.e. deactivates the secure mode. Access is granted for everybody.



LicelTCPIP SetWhitelist.vi

This VI is used to set the allowed hosts at the controller. In order to do so, the user must enter the appropriate password and 3 host strings to allowed IP addresses or IP address ranges. Such a string must be specified in the following format:

xx.xx.xx.xx	a single IP address,
xx.xx.xx.255	an IP address range (0:255),
	or may be empty.

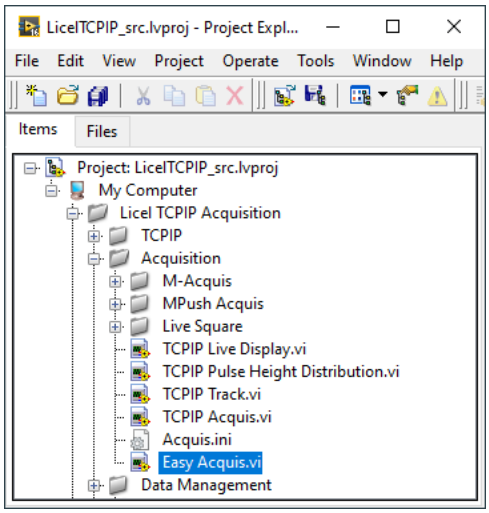


9.9 Basic LabVIEW Example VIs

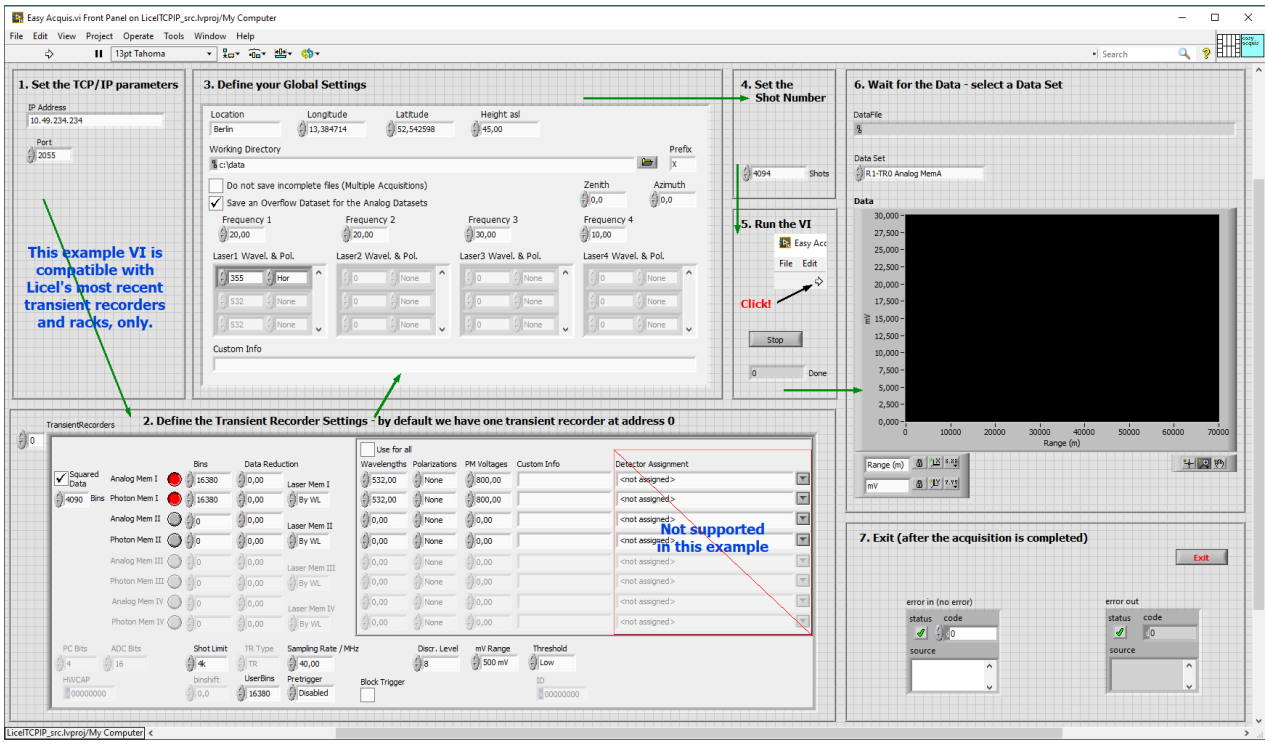
Licel provides a set of basic, easy to use LabVIEW example VIs. You are free to extract code from these VIs for integration into your own LabVIEW code.

9.9.1 `Easy Acquis.vi`

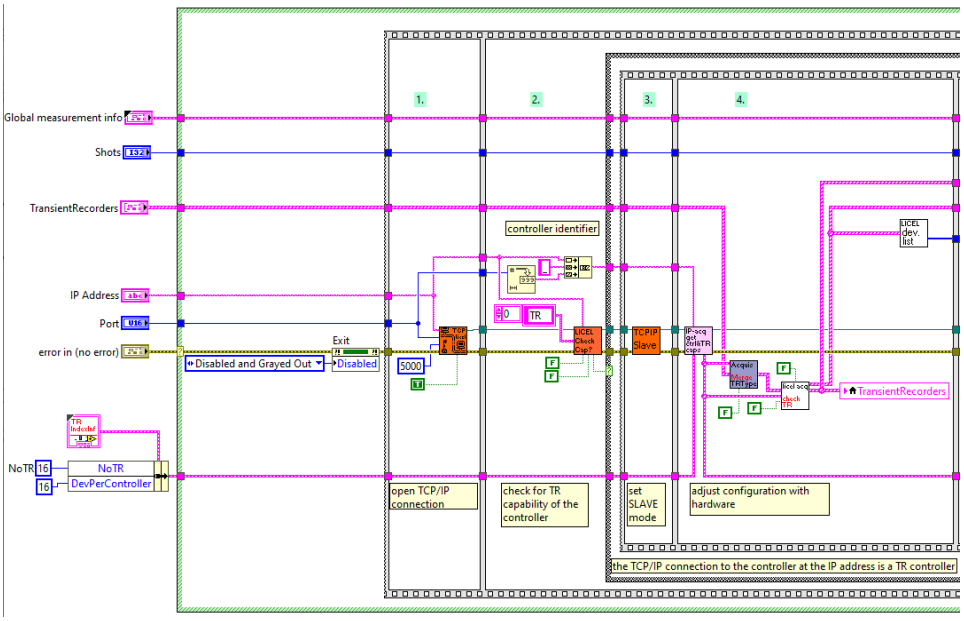
Find the VI `Easy Acquis.vi` in the LabVIEW source project in the virtual folder `Acquisition`.



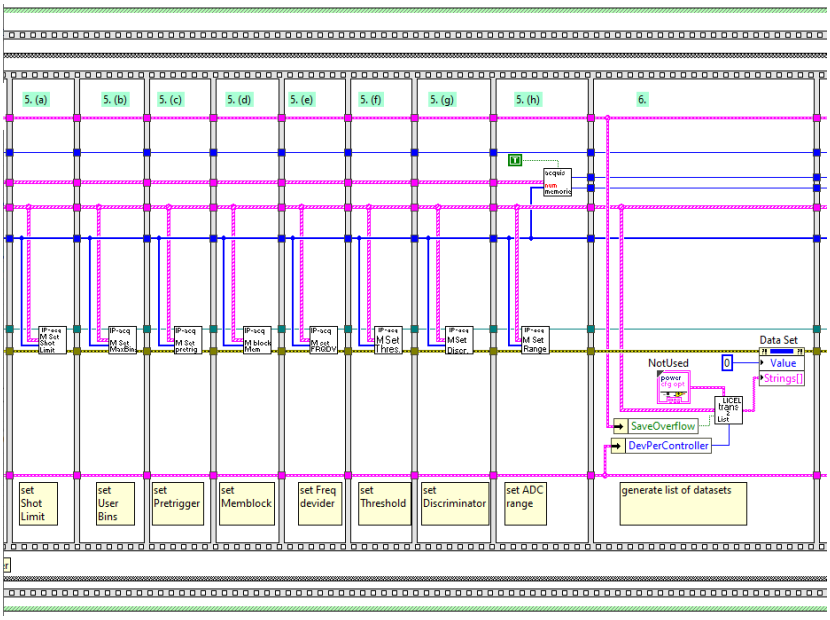
Please note that the example VI is designed for the usage with Licel's most recent Ethernet controller and transient recorder hardware.
The front panel comes up as follows:



The used front panel elements are the same as used in the configuration of Licel's acquisition software. Please note that some of the configuration features are not supported in `Easy Acquis.vi`.
Follow the green arrows to test the example VI with your Ethernet controller.
On the block diagram you will see and understand a step by step acquisition:



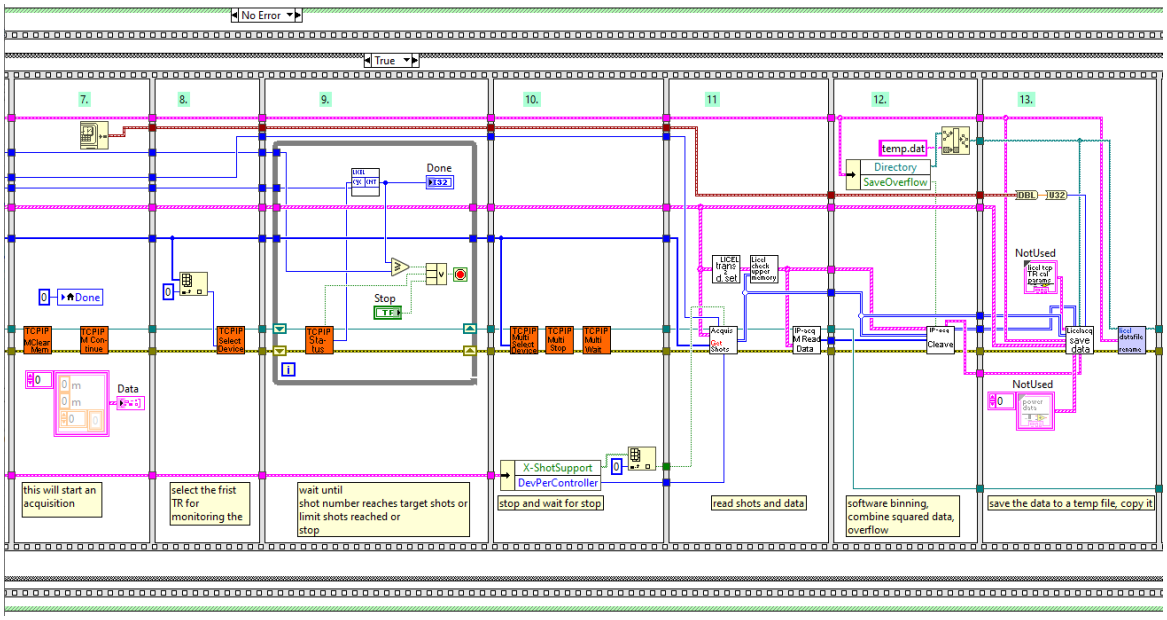
1. the TCP/IP connection is opened
2. verify that the Ethernet controller supports the transient recorder (TR) capability
3. a `SLAVE` command is sent to make sure that the `PUSH` mode is inactive
4. the available hardware is matched with the configuration settings on the front panel



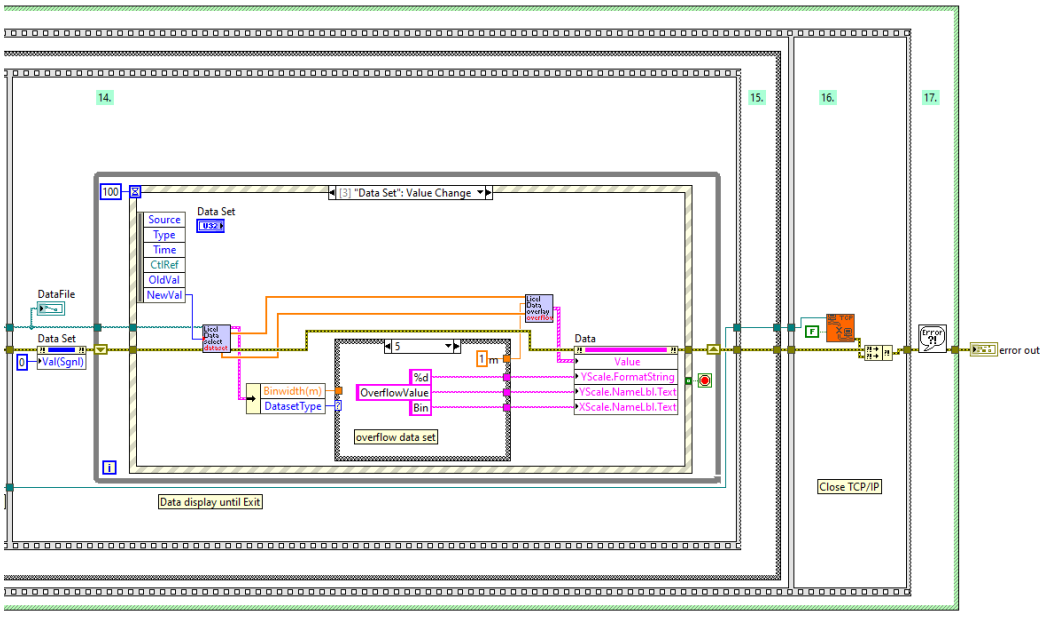
5. all transient recorder parameters are set for the acquisition:
   (a) SHOT `LIMIT`
   (b) user bins (
   (c) `PRETRIG`ger
   (d) memory `BLOCK`ing
   (e) frequency divider (`FREQDIV`)
   (f) set `THRESHOLD`
   (g) set `DISCRIMINATOR`

(h) set `RANGE`

6. generate the list of datasets to acquire



7. start the acquisition using `MCLEAR` and `MCONTINUE`
8. `SELECT` the leading device for monitoring the acquired shots
9. wait until the desired number of shots has been reached, the transient recorder's status is armed, or the user user has clicked the *Stop* button
10. explicitely stop all acquiring transient recorders (*MSTOP*), wait until all have stopped (*MWAIT*)
11. read the acquired shots and the raw data
12. combine the acquired raw data to the requested datasets
13. save the data using Licel's data file format

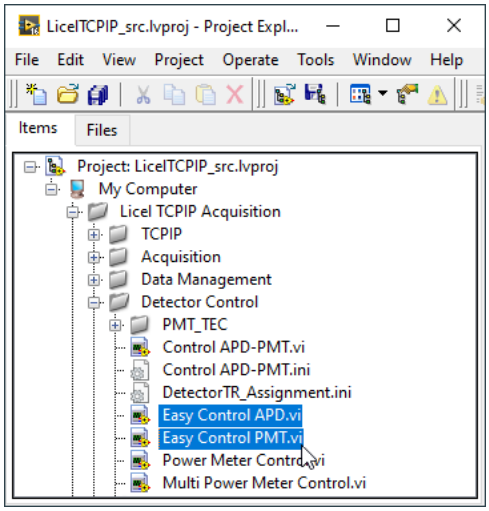


14. wait for the user to click the *Exit* button, meanwhile a change of the dataset list will result in reading the corresponding dataset from the written data file

15. close the TCP/IP connection
16. handle errors using LabVIEW's simple error handler.

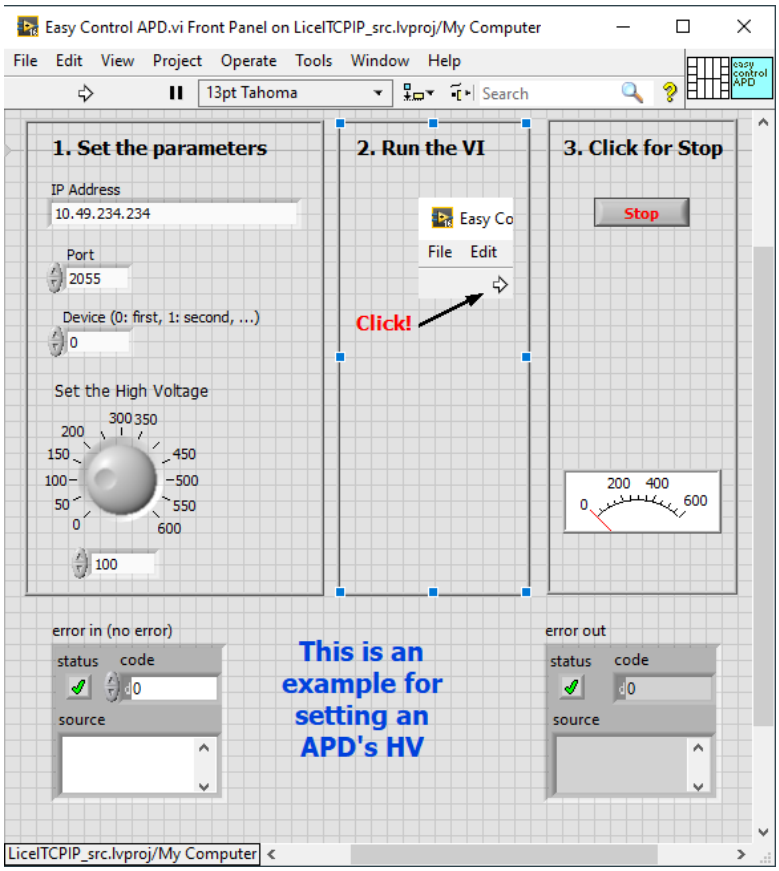
9.9.2 `Easy Control APD.vi` and `Easy Control PMT.vi`

Find the VIs `Easy Control APD.vi` and `Easy Control PMT.vi` in the LabVIEW source project in the virtual folder `Detector Control`.



`Easy Control APD.vi`

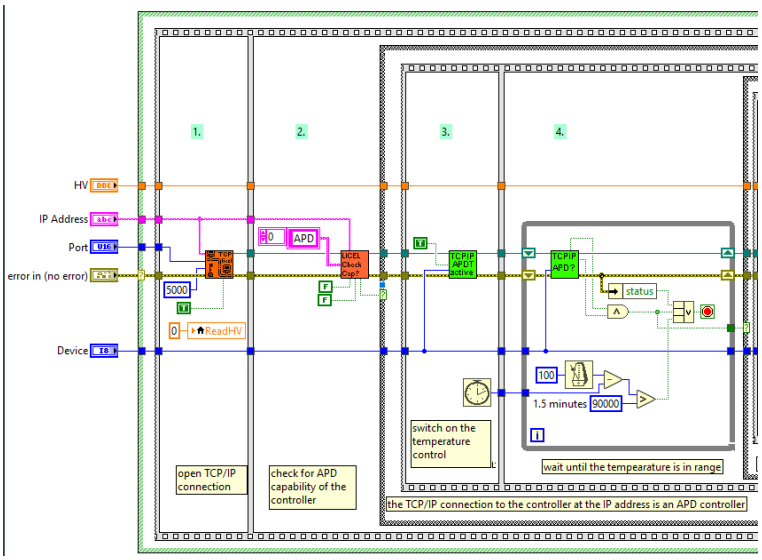
Please note that the example VI is designed to control one APD. It can easily be scaled for more APDs.
The front panel comes up as follows:



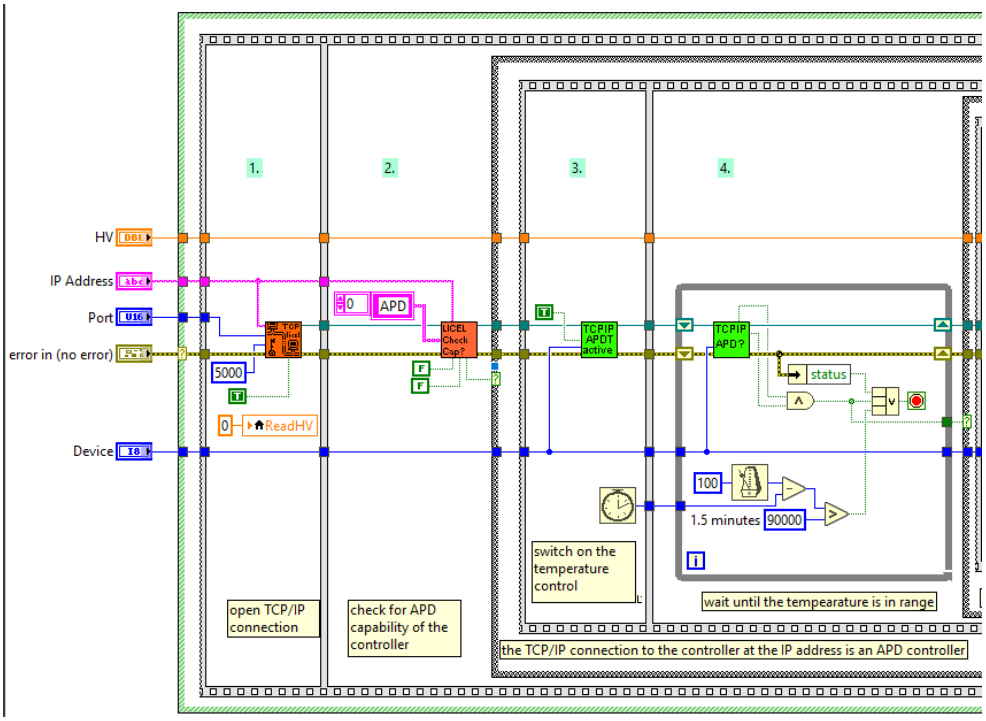
The used front panel elements are the same as used in the *Control APD-PMT.vi*. Please follow the instruction on the front panel:

1. set the parameters *IP Address*, *Port*, *Device* (= index of the APD cassette), choose the high voltage.
2. run the VI by clicking on the arrow button, wait until the VI is running
3. click on Stop to stop the VI's execution.

On the block diagram you will see and understand the VI's execution step by step:



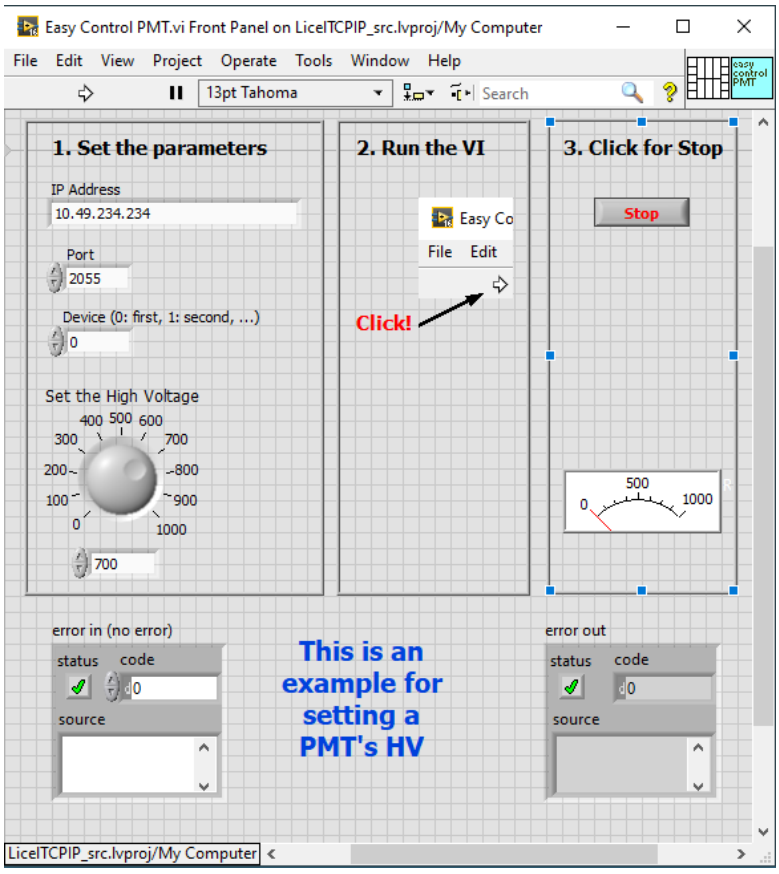
1. the TCP/IP connection is opened
2. verify that the Ethernet controller supports APD capability
3. start the TEC cooler using the *APDT* command
4. wait until the temperature is in range (use the *APD?* command)



5. set the high voltage using the *APDG* command
6. read the high voltage using the *APD?* command
7. wait until the user clicks the *Stop* button
8. set the high voltage to 0 V
9. switch off the TEC controller
10. close the TCP/IP connection
11. handle errors using LabVIEW's simple error handler.

`Easy Control PMT.vi`

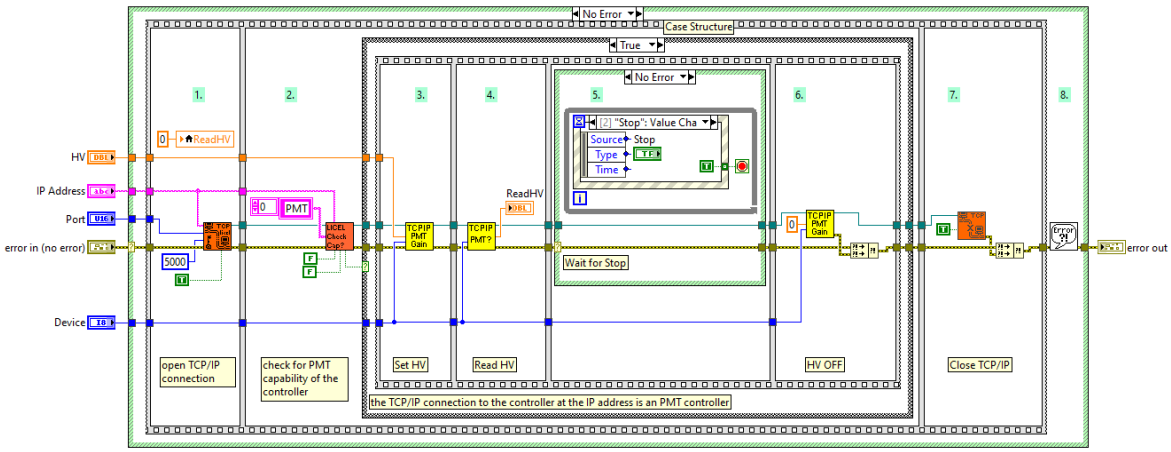
Please note that the example VI is designed to control one single PMT. It can easily be scaled for more PMTs.
The front panel comes up as follows:



The used front panel elements are the same as used in the *Control APD-PMT.vi*. Please follow the instruction on the front panel:

1. set the parameters *IP Address*, *Port*, *Device* (= index of the APD cassette), choose the high voltage.
2. run the VI by clicking on the arrow button, wait until the VI is running
3. click on Stop to stop the VI's execution.

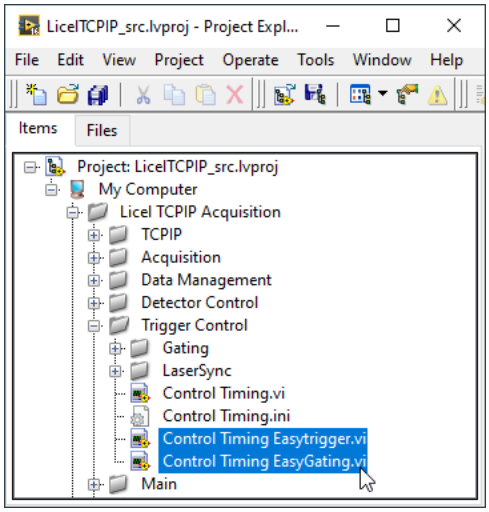
On the block diagram you will see and understand the VI's execution step by step:



1. the TCP/IP connection is opened
2. verify that the Ethernet controller supports PMT capability
3. set the high voltage using the *PMTG* command
4. read the high voltage using the *PMT?* command
5. wait until the user clicks the *Stop* button
6. set the high voltage to 0 V
7. close the TCP/IP connection
8. handle errors using LabVIEW's simple error handler.

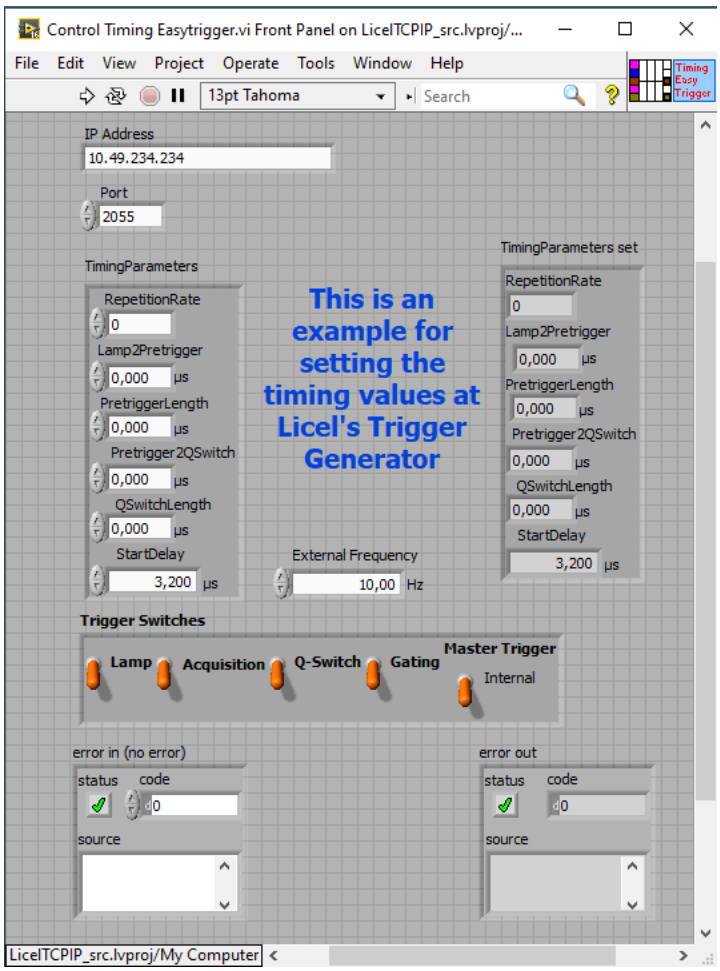
9.9.3 Timing Examples

Find the VIs `Control Timing Easytrigger.vi` and `Control Timing EasyGating.vi` in the LabVIEW source project in the virtual folder `Trigger Control`.



`Control Timing Easytrigger.vi`

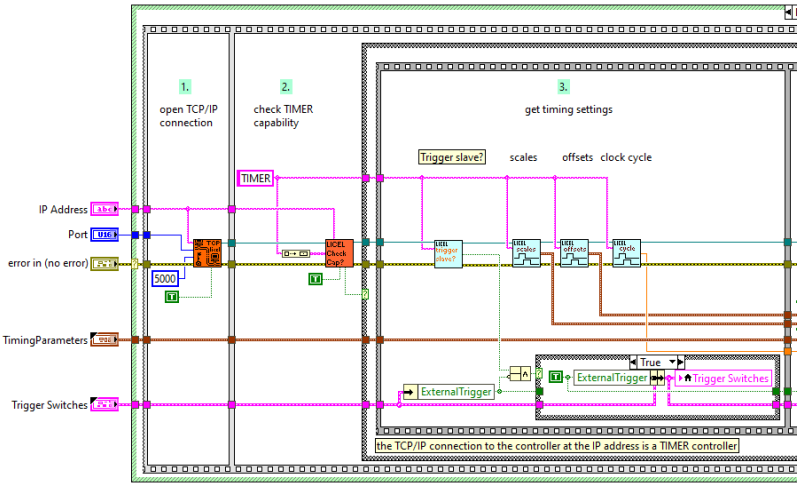
This VI sets the desired trigger times and enable active trigger outputs using the TRIGGERTIME and TRIGGERMODE commands.
The front panel comes up as follows:



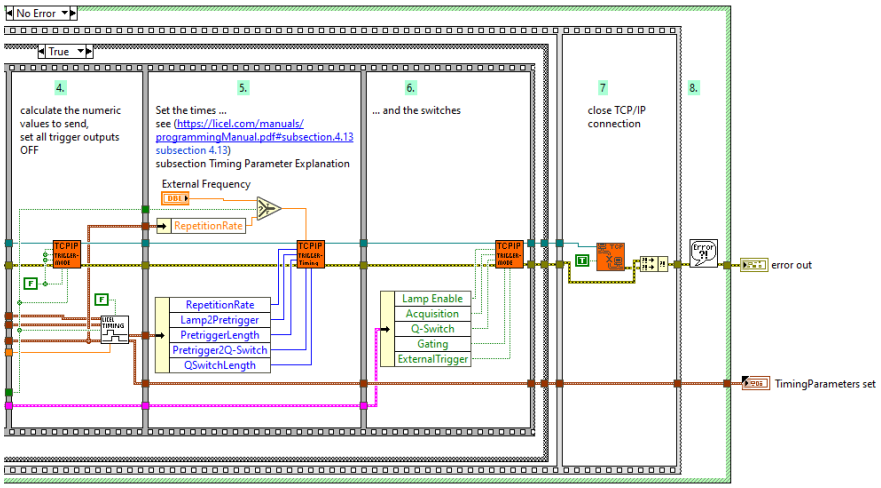
Please proceed as follows to run the VI and set the trigger times and outputs:

1. set the *IP Address* and *Port*
2. set the *Timing parameters* according to the diagram in the documentation of the TRIGGERTIME command
3. set the *External Frequency*
4. run the VI by clicking on the arrow button
5. the settings will be sent to the trigger controller.

On the block diagram you will see and understand the VI's execution step by step:



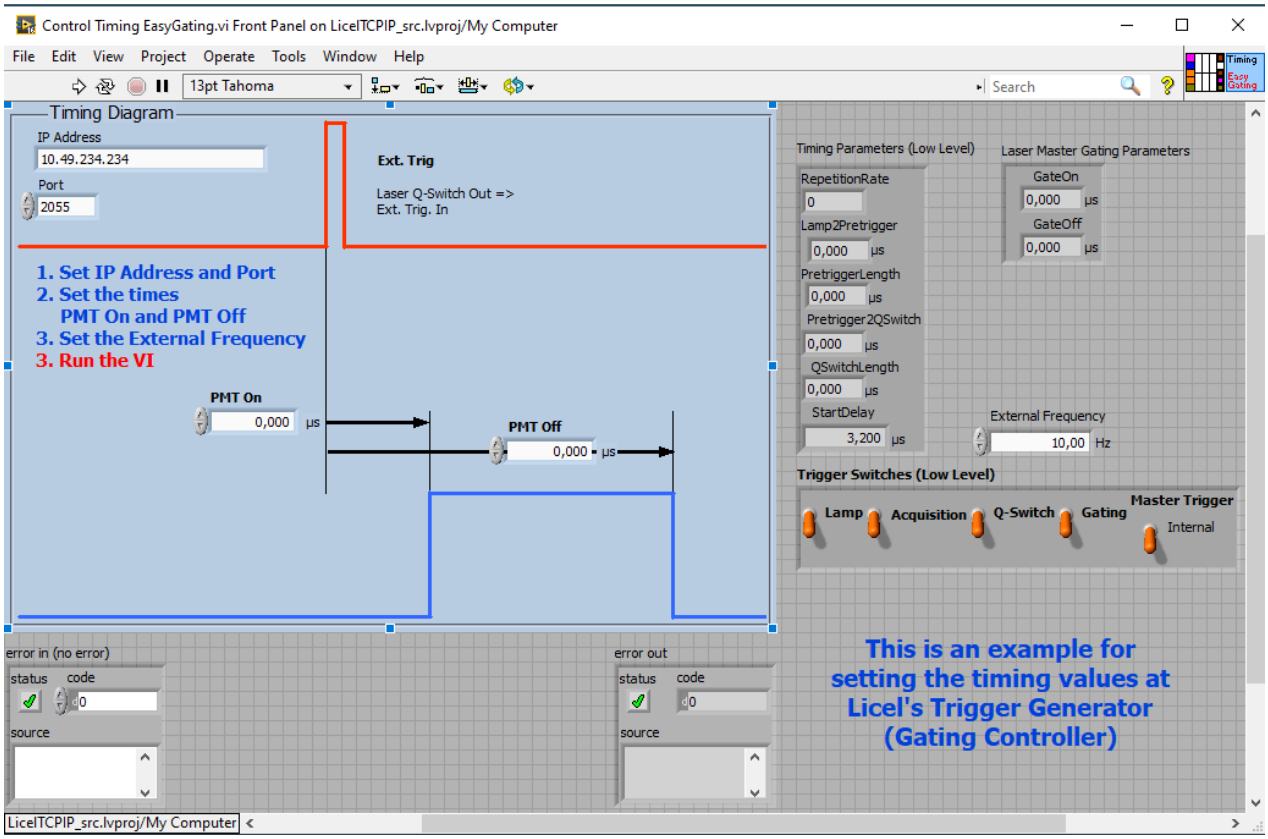
1. the TCP/IP connection is opened
2. verify that the Ethernet controller supports TIMER capability
3. find out if the controller board refers to a trigger slave (internally wired master trigger), read the scales, offsets, and the clock cycle. These parameters will be needed to calculate the timing values to submit to the controller



4. calculate the timing values, set all trigger outputs OFF
5. set the timing values
6. enable the trigger outputs
7. close the TCP/IP connection
8. handle errors using LabVIEW's simple error handler.

`Control Timing Easygating.vi`

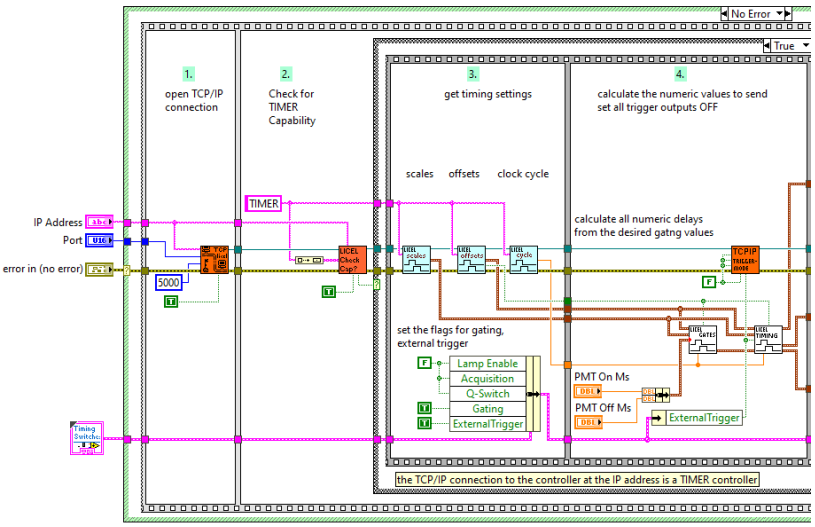
This VI will set trigger times and outputs using for an externally triggered gating acquisition. The low level trigger times and outputs are set using the TRIGGERTIME and TRIGGERMODE commands. The front panel comes up as follows:



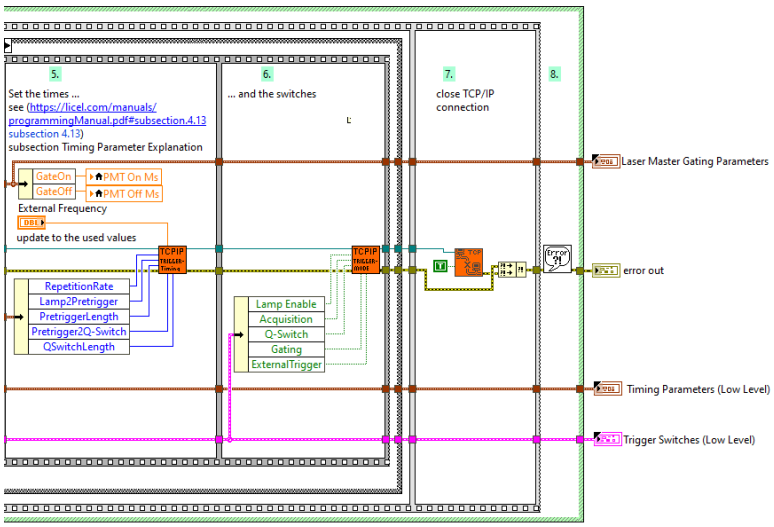
Please proceed as follows to run the VI and set the trigger times and outputs:

1. set the *IP Address* and *Port*
2. set the desired *PMT On* and *PMT off* times
3. set the *External Frequency*
4. run the VI by clicking on the arrow button
5. the VI will calculate the necessary low level trigger times and outputs according to the diagram in the documentation of the TRIGGERTIME command
6. the settings will be sent to the trigger controller.

On the block diagram you will see and understand the VI's execution step by step:



1. the TCP/IP connection is opened
2. verify that the Ethernet controller supports TIMER capability
3. read the scales, offsets, and the clock cycle. These parameters will be needed to calculate the timing values to submit to the controller
4. calculate the timing values from the gating times, set all trigger outputs OFF



5. set the timing values
6. enable the trigger outputs
7. close the TCP/IP connection
8. handle errors using LabVIEW's simple error handler.