

LogView Studio

Manual

Table of Contents

Part I	Introduction	1
Part II	About LogView Studio	2
1	LogView Studio on the Web	2
2	What's New	2
3	Getting Support	2
4	Requirements	3
5	Contact	4
Part III	English	4
1	Donation	5
2	Installation	5
.	.Net Framework	5
.	Install LogView Studio	6
.	Update LogView Studio	11
.	Uninstall LogView Studio	11
.	Remove User Data	12
.	Use ZIP Archive	13
.	Remove ZIP Blocking	13
.	Firewall	15
3	Quickstart	16
4	Basics / FAQ	21
.	What is LogView Studio?	22
.	What is a Project?	22
.	What is a Device?	23
.	What is a Source?	24
.	What is a Channel?	25
.	What is a Dataset?	26
.	What is a Sensor?	27
.	What is a View?	27
.	Static and Live View	28
.	Garbage - the trash can	29
.	GUI / CLI - What's that?	29
.	LogView Studio on Linux	30
.	LogView Studio and Network	31
5	User interface	33
.	Tool Windows	33
.	MDI Windows	35
File	36
.	New Project	37
.	New Remote Project	38
.	Open Project	39
.	Save Project	40
.	Close Project	41
Project	41

Device	41
Device Monitor.....	43
Chart	44
GPS 3D	45
Table	46
Circular Gauge.....	47
Last Values.....	47
Folder	48
Delete	49
Export	50
Chart - Graphic.....	50
Chart - CSV.....	51
Dataset - CSV	51
Dataset - XML.....	51
Dataset - JSON.....	52
Dataset - KML.....	52
Special - Application Screenshot.....	52
Special - Desktop Screenshot.....	53
Analyse	53
Dataset - Min Max Average.....	53
Dataset - Average Filtered.....	54
Tools	55
Change Start Time.....	56
Open Settings Folder.....	57
Add Host to Database.....	57
View	57
Tool Windows	57
Windows	58
Help	58
6 Working with LogView Studio	58
Device	58
Device Start and Stop.....	59
Change Source.....	59
Fileimport	61
Autodetect Serial / HID.....	61
Zero Sensor Devices.....	62
Device Monitor	63
Chart	65
Zoom und Pan.....	65
Layout	66
Templates.....	67
Measurement.....	69
7 Samples	70
Simple Project	71
Network Usage	77
Part IV Deutsch	81
1 Spenden	81
2 Installation	82
.Net Framework	82
LogView Studio installieren	83
LogView Studio updaten	88

LogView Studio deinstallieren	88
Benutzerdaten entfernen	89
ZIP Archiv nutzen	90
ZIP Blockade entfernen	90
Firewall	92
3 Schnellstart	93
4 Grundlagen / FAQ	98
Was ist LogView Studio?	99
Was ist ein Project?	99
Was ist ein Device?	100
Was ist eine Source?	101
Was ist ein Channel?	102
Was ist ein Dataset?	103
Was ist ein Sensor?	104
Was ist eine View?	104
Static und Live View	105
Garbage - der Mülleimer	106
GUI / CLI - Was ist das?	106
LogView Studio unter Linux	107
LogView Studio im Netzwerk	108
5 Oberfläche	110
Tool Fenster	110
MDI Fenster	112
File	113
New Project	114
New Remote Project	114
Open Project	116
Save Project	117
Close Project	118
Project	118
Device	118
Device Monitor	120
Chart	121
GPS 3D	122
Table	123
Circular Gauge	124
Last Values	124
Folder	125
Delete	126
Export	127
Chart - Graphic	127
Chart - CSV	128
Dataset - CSV	128
Dataset - XML	128
Dataset - JSON	128
Dataset - KML	129
Application Screenshot	129
Desktop Screenshot	129
Analyse	130
Min Max Average	130
Average Filtered	131
Tools	132
Change Start Time	132

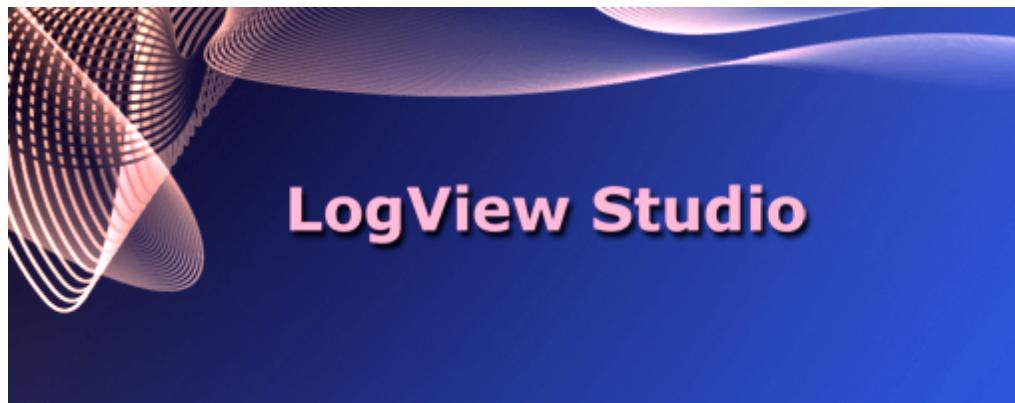
Open Settings Folder.....	133
Add Host to Database.....	133
View	134
Tool Windows.....	134
Windows.....	134
Help	135
6 Arbeiten mit LogView Studio	135
 Device	135
Device Starten und Stoppen.....	135
Source Konfiguration.....	136
Device mit Dateiimport.....	137
Autodetect Serial / HID.....	138
Zero Sensor Devices.....	139
 Device Monitor	140
 Chart	141
Zoom und Pan.....	141
Layout	142
Templates.....	144
Messfunktionen.....	146
7 Beispiele	147
einfaches Projekt	147
Netzwerk Nutzung	154

Part V Linux Prepare 157

1 SSH Connect	158
2 Raspberry Pi	160
System Preparation	161
Mono	162
Install Mono.....	163
Compile Mono from TAR.....	164
Compile Mono from GIT.....	165
Testing	166
3 Cubietruck	167
System Preparation	168
Mono	169
Install Mono.....	170
Compile Mono from TAR.....	171
Compile Mono from GIT.....	172
Testing	173
4 Lubuntu	174
System Preparation	175
Mono	175
Install Mono.....	176
Compile Mono from TAR.....	176
Compile Mono from GIT.....	178
5 CentOS	179
System Preparation	180
Mono	180
Compile Mono from TAR.....	181
Compile Mono from GIT.....	183

Part VI Command Line Tool LvCmd	184
1 Webserver	188
Part VII Development	189
1 OpenFormat Zero	189
Description	190
Header	192
Time Formats	194
Data	195
Multi Channel Usage	196
Sample : File Import	196
Sample : Arduino TCP	199
Sample : Arduino Serial	204
2 OpenFormat INI	205
Description	206
Sample INI	210
Index	0

1 Introduction



LogView Studio is the next generation milestone in collecting and visualizing data from various sources.

This manual will give you some information about using LogView Studio and its tools. Additionally it will offer some guidelines for special topics.

The most parts of this manual are in English. Only the most important Chapter are also available in German language.

Topic	Description
About LogView Studio	Contains support, licensing and requirement information about LogView Studio.
English	Contains basic information about LogView Studio. It's a good advice to read this sections if you new to LogView Studio
Deutsch	In diesem Kapitel werden grundlegende Informationen für LogView Studio bereit gestellt. Wenn Sie neu auf dem Gebiet LogView Studio sind sollten sie die folgenden Kapitel aufmerksam durchlesen.
Linux Prepare	Basic information how to setup a Linux system to run LogView Studio (LvCmd)
Command Tool LvCmd	Information about the Command Line Tool LvCmd which runs on Windows and Linux
Development	Here you will find development information like OpenFormat

LogView Studio is created by

Dominik Schmidt dominik@logview.info

Holger Hemmecke holger@logview.info

Nobody is perfect. Even our English translation may contain logical or syntactical errors. In case you feel some text description shall be adjusted please send your suggestion to

dominik@logview.info.

Copyright for LogView Studio by Dominik Schmidt & Holger Hemmecke

It is not allowed to use LogView Studio for commercial usage. Contact us if you like to use it in such an environment.

All code parts are intellectual property of Dominik Schmidt and Holger Hemmecke.

2 About LogView Studio

This section contains support, licensing and requirement information about LogView Studio.

Sections

This chapter is divided into different sections:

- [LogView Studio on the Web](#)
- [What's New](#)
- [Getting Support](#)
- [Requirements](#)
- [Contact](#)

2.1 LogView Studio on the Web

LogView Studio can be found on the [LogView website](#). We host discussion forums, provide support and inform about LogView Studio updates on this site. Make sure to visit our website regularly to stay up-to-date with LogView Studio news.

Links

- LogView website: <http://www.logview.info/>
- LogView forum: <http://www.logview.info/forum/>

2.2 What's New

This page lists all new features, bug fixes and changes of the latest LogView Studio releases. Please note that not all changes are inserted here. This page will only cover major changes.

LogView 1.0.XXX.XXX

This release contains the following changes:

- Change1
- Change2

2.3 Getting Support

We provide support over different channels such as this help file, community-supported forums and email support.

Please read this manual carefully in order to answer the basic questions about LogView Studio.

Links

- Forum: <http://www.logview.info/forum/index.php>
The Forum is our preferred support platform since the information is available for all other user.
- Mail: dominik@logview.info

2.4 Requirements

LogView Studio can be used in two ways. One is the graphical Windows application (GUI) and the other one is the command line tool (CLI).

Supported Platforms GUI

The following operating systems and platforms are supported by LogView Studio GUI.

Type	
Operating System	Windows XP Service Pack 3 and later
CPU (MHz)	Not really important but on Windows ~1,5GHz would be Ok.
Memory / Ram	Minimum is 512MB (more is better)
.Net Framework	.Net 4.0 Full (4.0 Client profile is not enough !)
MySQL Database	MySQL from version 5.6 MiraDB from version 5.5 / 10.0

Supported Platforms CLI

The following operating systems and platforms are supported by LogView Studio CLI.

Type	
Operating System	Linux
CPU (MHz)	Runs from ~700MHz like the Raspberry Pi
Memory / Ram	Minimum is 256MB (more is better / could be run with 128MB but untested)
.Net Framework	Mono from version 3.0 Full (which supports .Net 4.0 Full)
MySQL Database	MySQL from version 5.6 MiraDB from version 5.5 / 10.0

Hint

Mono 2.x lacks on some Arm derivatives like the Raspberry Pi since it has some trouble with the floating point unit. So better use Mono 3.x as that is already available.

2.5 Contact

If you need to contact us the mail addresses below can be used. Please note that we currently cannot answer support questions by phone. Thank you for your understanding.

Contact by Email

The best way to contact us is by email. Please see below for a list of email addresses for different purposes.

dominik@logview.info

General questions and information.

holger@logview.info

General questions and information.

LogView Software Address

Dominik Schmidt

Bruchstrasse 2

59929 Brilon Madfeld

Germany

Copyright

Copyright 2013-2014 Dominik Schmidt & Holger Hemmecke. All rights reserved. Any duplication or distribution of parts or as a whole of this content is forbidden without prior written approval of the LogView owner.

Please see our general Business and Licensing Conditions in this manual for more information.

3 English

This chapter contains basic information about LogView Studio. It's a good advice to read these sections if you are beginner with LogView Studio.

Sections

This chapter is divided into different sections.

- [Donations](#)
- [Installation](#)
- [Quickstart](#)
- [Basics](#)
- [User Interface](#)
- [Working with LogView Studio](#)
- [Samples](#)

3.1 Donation

As you may know LogView is based on donations and till now we do this non profit in our spare time. We use the money from your donations to buy software licenses for the development process and for monthly costs like our web server.

If you like our software LogView Studio feel free to send us a donation. You will support and confirm our work.

Paypal

You can donate via Paypal using the email address

dominik@logview.info

Alternatively you can use this Link which will redirect you to our donation page within our forum

<http://logview.info/forum/index.php?donate/>

We will insert some donation goals at that page which are important for our work.

Bank transfer

If you like to donate some money via bank transfer please send en email to dominik@logview.info.

We had some trouble in the past with bank data abuse so we don't longer show the bank details here.

3.2 Installation

This chapter contains information about Install, Uninstall, Firewall.

Sections

This chapter is divided into different sections:

- [.Net Framework](#)
- [Install LogView Studio](#)
- [Update LogView Studio](#)
- [Uninstall LogView Studio](#)
- [Remove User Data](#)
- [Use ZIP Archive](#)
- [Firewall](#)

3.2.1 .Net Framework

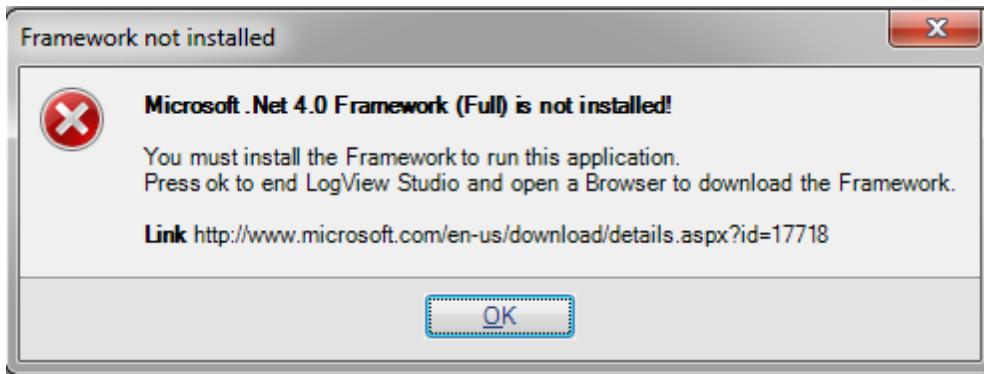
Logview Studio is based on the Microsoft .NET Framework. Therefore .NET Framework has to be installed mandatory.

Wikipedia

The .NET Framework is Microsoft's comprehensive and consistent programming model for building applications that have visually stunning user experiences, seamless and secure communication, and the ability to model a range of business processes.

For proper function of Logview Studio it is necessary to have the complete edition of .NET 4.0 installed. Any preinstalled reduced version (like the .Net 4.0 Client profile) is not sufficient.

In case the .NET Framework (Full) is missing on your computer, the following message will appear on your screen.



Download

The installer is available for downloading at Microsoft's Internetpage

<http://www.microsoft.com/de-de/download/details.aspx?id=17718>

Alternatively you may search for ".Net Framework 4.0 Full" in your favorite searchengine.

Installation

The Installation is well documented and don't need any further explanation.

3.2.2 Install LogView Studio

Download

LogView Studio can be downloaded from the LogView Webpage.

<http://www.logview.info/forum/index.php?resources/>

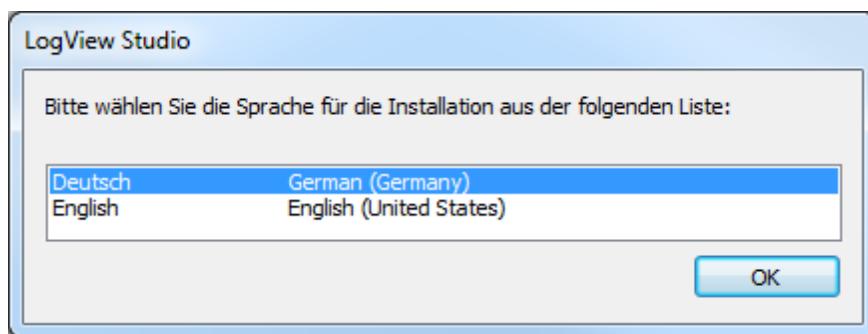
There is always the latest installer available.

Important Note

The installer will be available only when the testing period has been finished. Before that date the installer will be sent out by email to the registered testuser only.

Installation

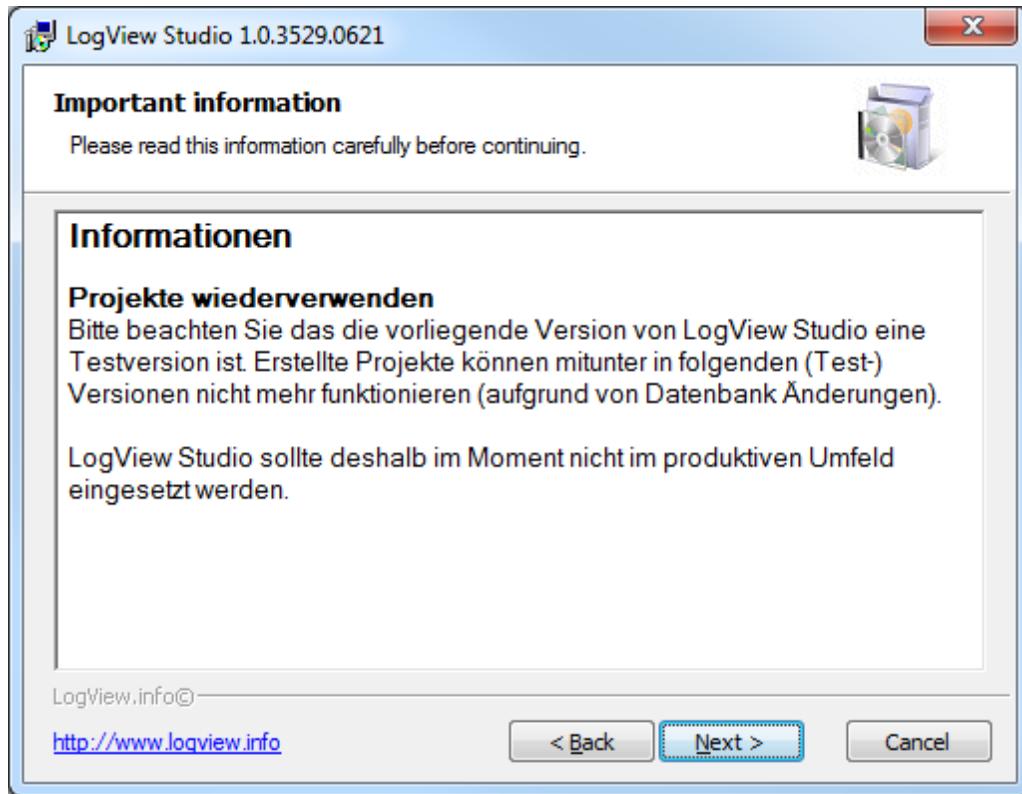
After downloading and saving on your harddrive please start the installer and select your language. This selection is valid for the installer only.



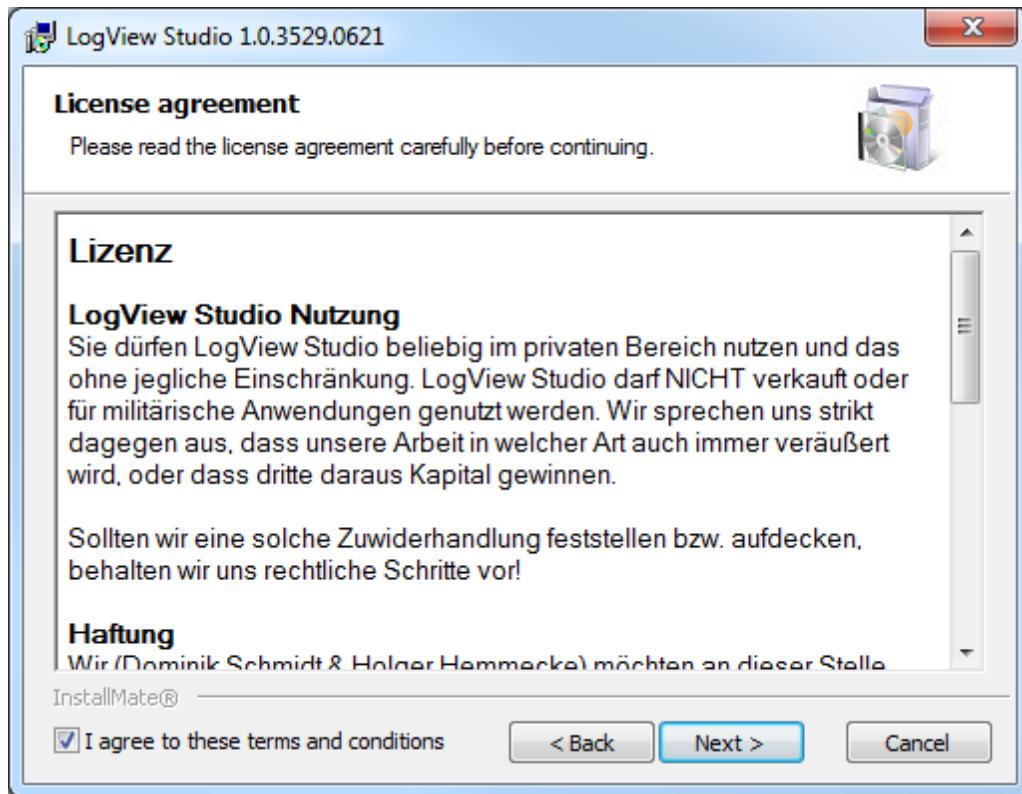
Read the Startpage and continue with Next.



Read the Informationpage carefully and continue with Next.



Please read the terms of license carefully and mark the checkbox bottom left 'I agree ...' before continuing with Next.

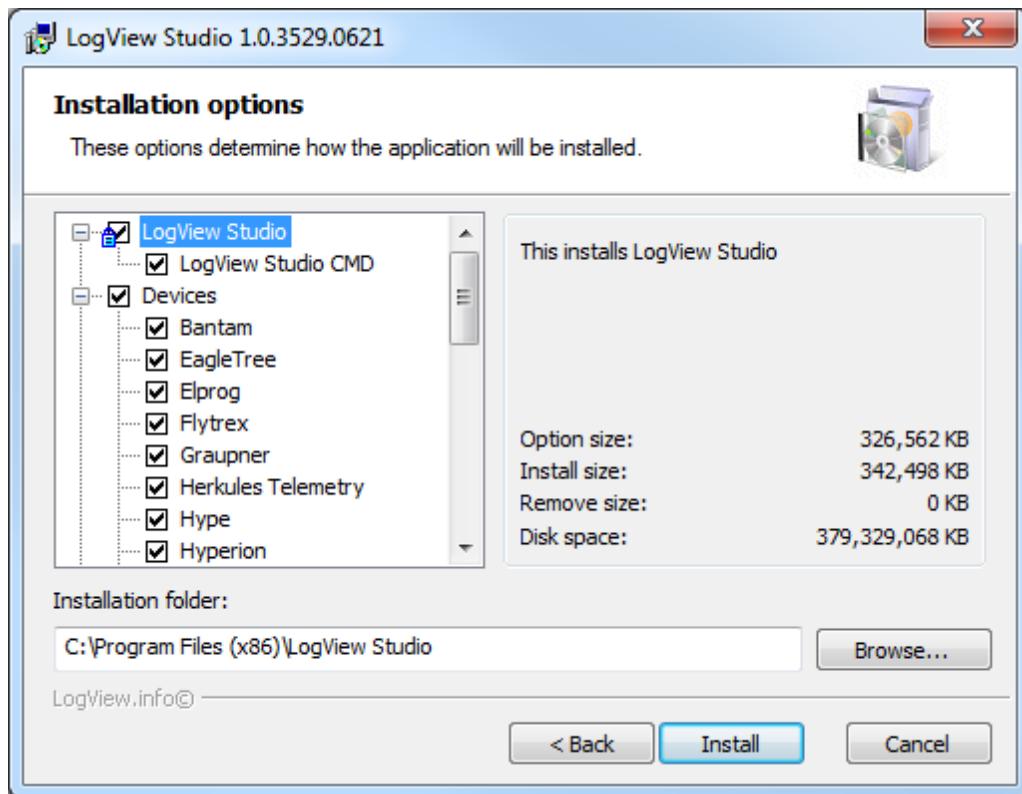


Please select your preferred installation folder.

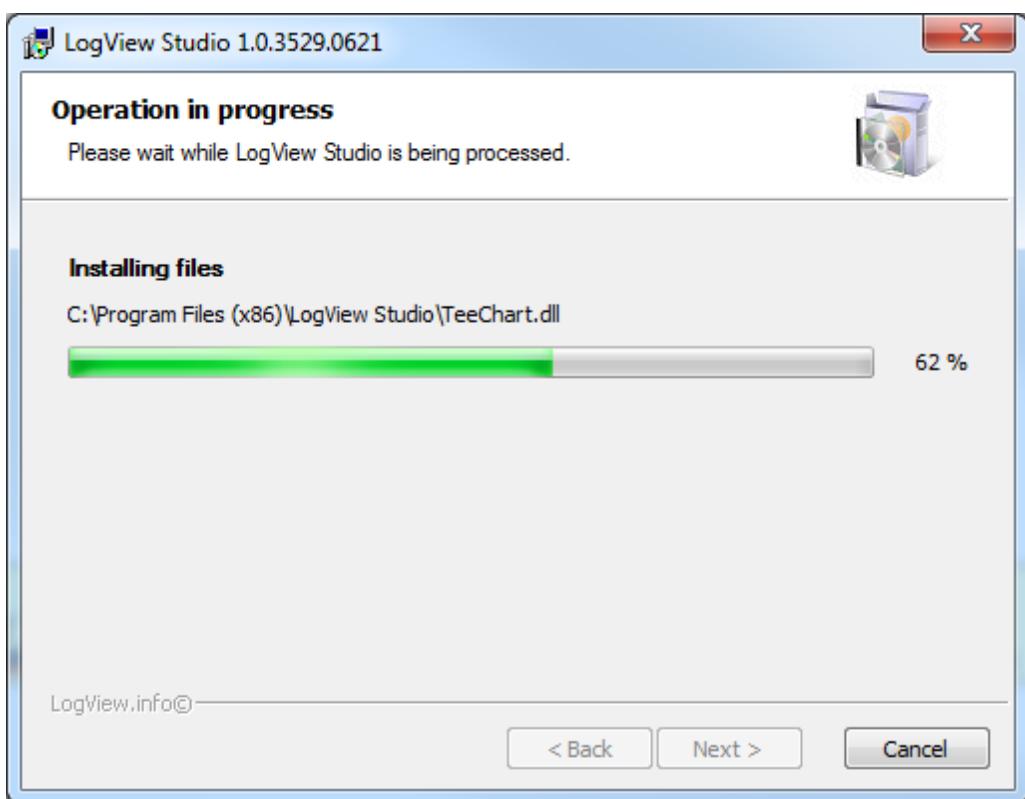
Note

It is recommended to reduce the selection of devices in the 'Install Options' to those Manufacturers which you intend to use. In case the selection remains as marked to all devices the Start of LogView Studio may be delayed unnecessarily because all (or only) selected devices must be registered in the database first.

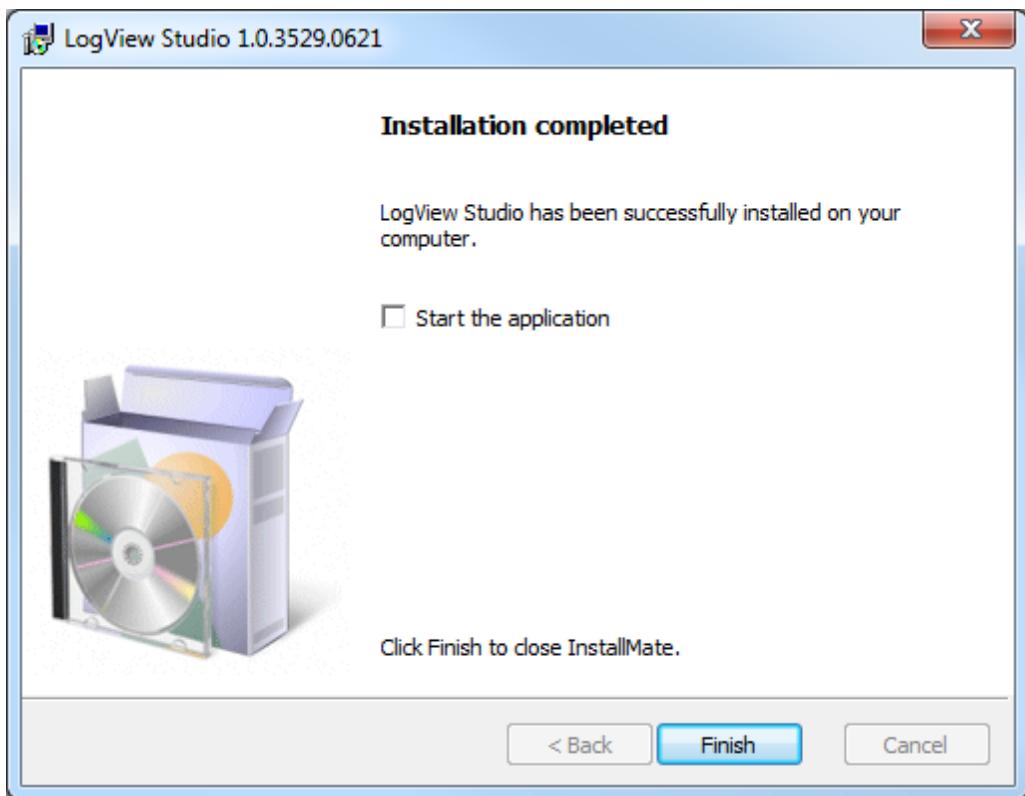
Then you can start the Installation using the Install Button.



LogView Studio Installation will be started ...



In the final window you may select immediate Start of LogView Studio if you like and the Installation is completed.



3.2.3 Update LogView Studio

Download

LogView Studio can be downloaded from the LogView Webpage.

<http://www.logview.info/forum/index.php?resources/>

There is always the latest installer available.

Update

The Installer is the same for first installation as for updating. In case the installer will find a previous installation of LogView Studio then only the relevant updated files will be exchanged.

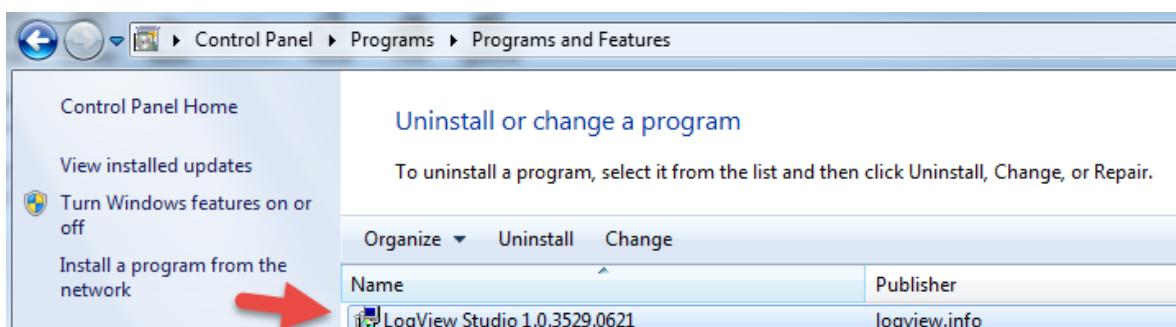
There is no major difference between updating process and [first installation process](#).

Tip

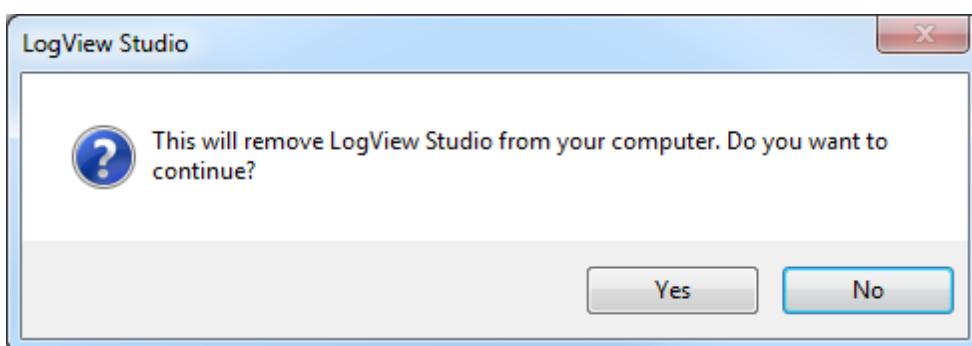
You can also add or remove your selected devices during an update process.

3.2.4 Uninstall LogView Studio

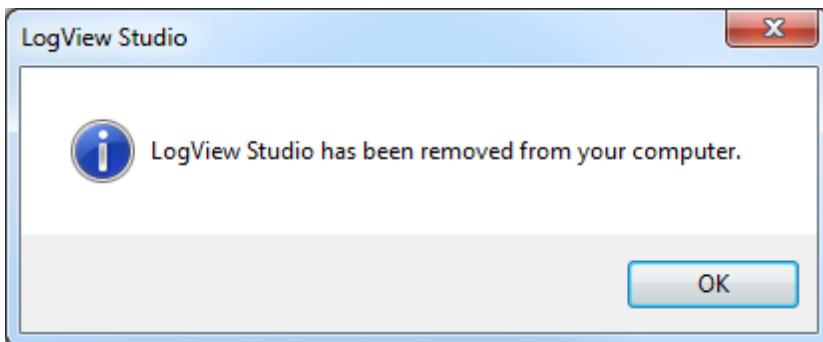
Please use Windows 'Control Panel' system function 'Uninstall Software' to uninstall LogView Studio from your Computer.



Select the entry 'LogView Studio' from the list and click Uninstall.
Continue with Yes.



Next windows will confirm the successful removing of the program.



Remaining Files

During the uninstall process only the installation folder and all links will be removed. Following files will remain existing:

- Projects
Projects which have been created / saved on your harddrive are not deleted by Installer. If you want to or need to delete those files from your harddrive then please do this manually using Windows File Explorer or a similar tool. The complete project folder can be deleted. As you have specified the exact position of the project folder during the creation this should not be a problem.
- Userdata
LogView Studio need to save some few datafiles inside the Userfolder like for example the last used projects. This folder will remain as well and will not be deleted during uninstall process. Here you can find further informations about [Remove User Data](#).

3.2.5 Remove User Data

LogView Studio has to store some data inside a user folder. Beside others these are:

- recent used projects
- Error Logs and Screenshots
- global graphic templates (if applicable)

These data files and/or sub folders will remain on the computer and will not be deleted during the uninstallation process. They have to be removed manually as per individual needs.

Folder

The folder can be found on different locations of the folder structure within the various operation systems as follows:

System	Folder
Windows XP	C:\Documents and Settings\<USERNAME>\Local Settings\\Applicationdata\LogView Studio C:\Dokumente und Einstellungen\<USERNAME>\Lokale Einstellungen\Anwendungsdaten\LogView Studio
Windows Vista, 7, 8, 8.1	C:\Users\<USERNAME>\AppData\Local\LogView Studio C:\Benutzer\<USERNAME>\AppData\Local\LogView Studio
Linux	/home/<USERNAME>/LogView Studio

Registry

LogView Studio generally does not save any data in the Registry.

Note

The only exemption of that is the Installer itself: The Installation info is registered in the registry which is necessary for the Uninstallation process (amongst other things).

3.2.6 Use ZIP Archive

LogView Studio is also available as pure Command Line Version. That Edition is normally used as download source only under Linux operating system. In case the Installation with the Installer package is not possible due to whatever reason this ZIP-Archive can be used as Alternative Solution.

Download

Download is available at following location:

<http://www.logview.info/lvstudio/LVStudio.zip>

Unzipping

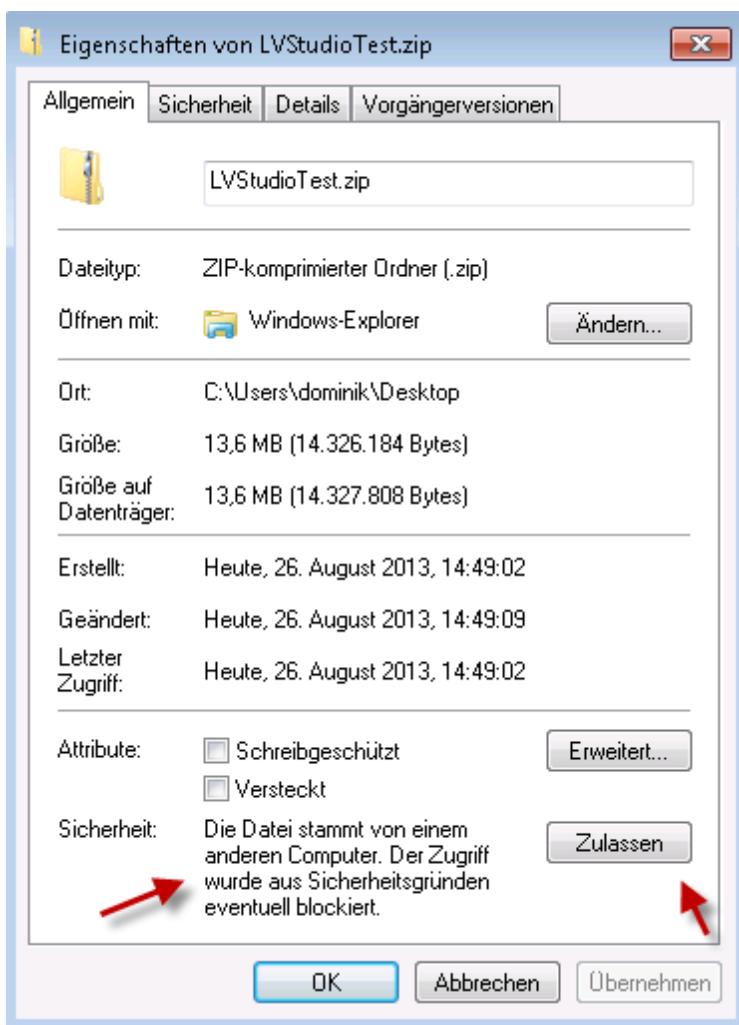
To unzip the file you can use any standard tool which can handle the 'ZIP' compression.

Important is only to keep the folder structure as given inside the archive file.

When using MS Windows it might be helpful to [check if Windows has blocked the zip file](#).

3.2.6.1 Remove ZIP Blocking

Since MS Windows XP SP2 (and higher) any ZIP files are classified as potential dangerous after downloading from the internet. You can recognize that by checking the ZIP-file Properties.



Additionally you will see a security warning after starting the program.



You can unzip the primary file without problems but if that is done with the original Windows UNZIP function also the unzipped files will be blocked as potential unsafe.

That leads to the problem that amongst others the LogView Studio can not open and load the needed files for the devices. In the Log you may see following information:

```
Mo 14:54:44.805 : [ 1, Main      ] An attempt was made to load an assembly from a network location which would have caused the assembly to be sandboxed in previous versions of the .NET Framework. This release of the .NET Framework does not enable CAS policy by default, so this load may be dangerous. If this load is not intended to sandbox the assembly, please enable the loadFromRemoteSources switch. See http://go.microsoft.com/fwlink/?LinkId=155569 for more information.  
Mo 14:54:44.810 : [ 1, Device    ] Jeti.dll is not a valid LogView Studio Device.
```

Solution 1

Don't use the Windows unzipping tool but choose a third-party free ZIP-Tool like 7ZIP or similar. With such program the blocking will not be handed over to the unzipped files.

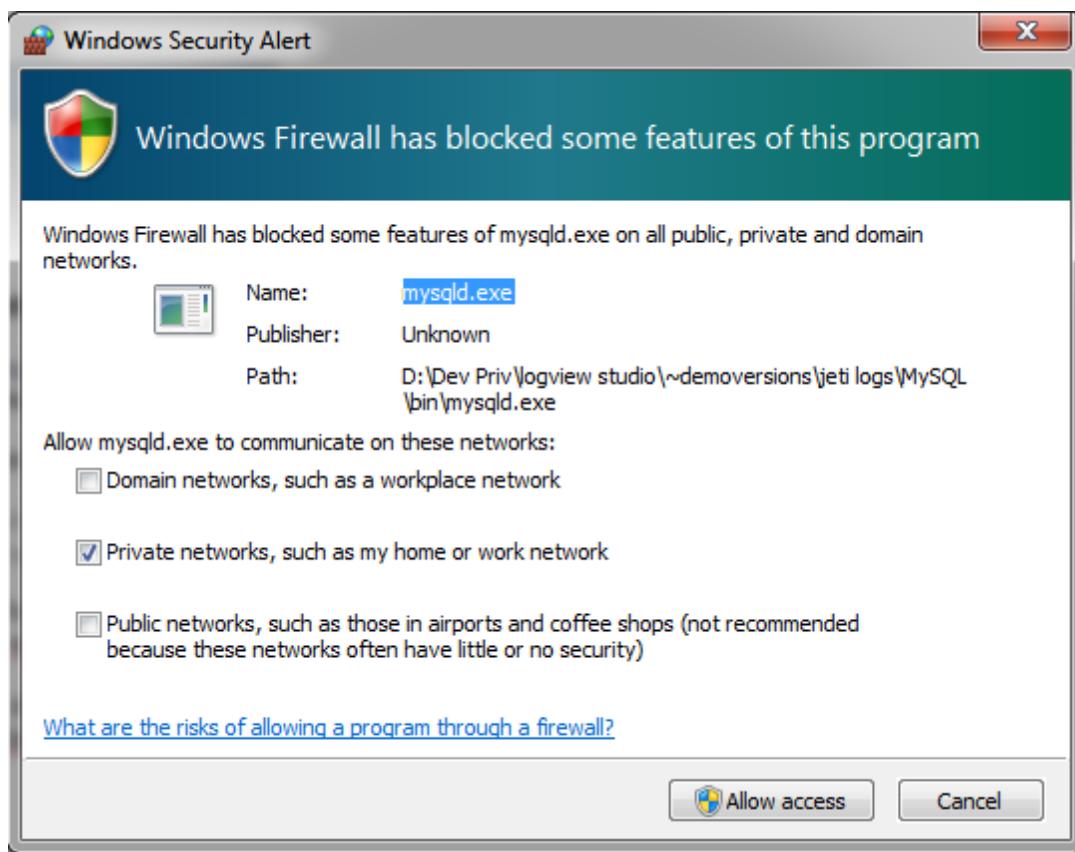
Solution 2

Open the properties of the ZIP-file and click to the button "allow". This will remove the blocking and the file can be unzipped using any unzipping tool without creating further problems.

3.2.7 Firewall

Dialog for mysqld.exe

After the first start of LogView Studio you may see one or two firewall dialog windows:



LogView Studio is using a MySQL Database Server for saving data. Please select 'allow access' for private networks such as your home or work network for the 'mysqld.exe'

Dialog für LvGui.exe / LvCmd.exe

A similar dialog may appear also for LogView Studio itself. In that case the relevant files will be 'LvGui.exe' or 'LvCmd.exe'. These LogView Studio files need also access to the network in order to reach the database or execute Internet access. Please also allow them access to the private network.

If this access is not granted some functional errors may appear when using LogView Studio.

If you have a different Firewall installed the dialog windows may look similar. Also then you should allow access to the network for the 'mysqld.exe' and eventually the LogView Studio files 'LvGui.exe' or 'LvCmd.exe' if such dialog window will appear.

Installer

The installer prepares and installs normally a relevant exemption rule for the Microsoft Firewall.

3.3 Quickstart

In this chapter we like to explain in short words how to create a project in LogView Studio, how to add a device, how to configure a device and how to start logging in order to create a graph at the end. Hereby we only show parts of the powerful options

which are possible with LogView Studio.

Tip

The following documented Quickstart show in general the function of LogView Studio. If you are familiar with the previous LogView V2 you should not have any problems with the following demonstration and the functions of LogView Studio. Generally we highly recommend to read all the additional information about LogView Studio carefully to understand the full functionality.

Requirements

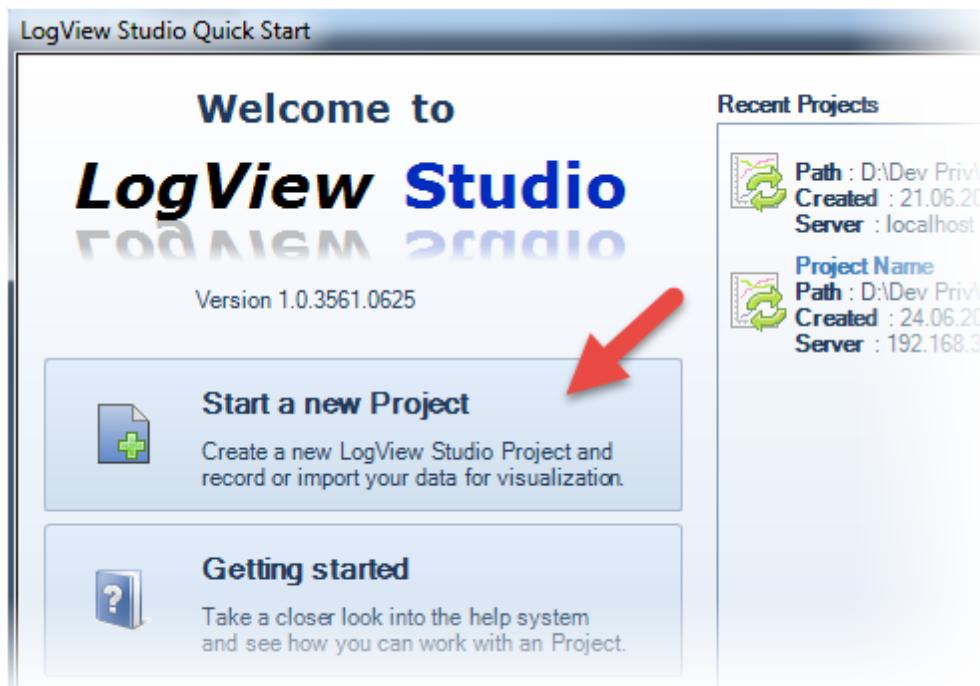
- [.Net Framework 4.0 Full](#) must be installed
- [LogView Studio must be installed](#)

Start LogView Studio

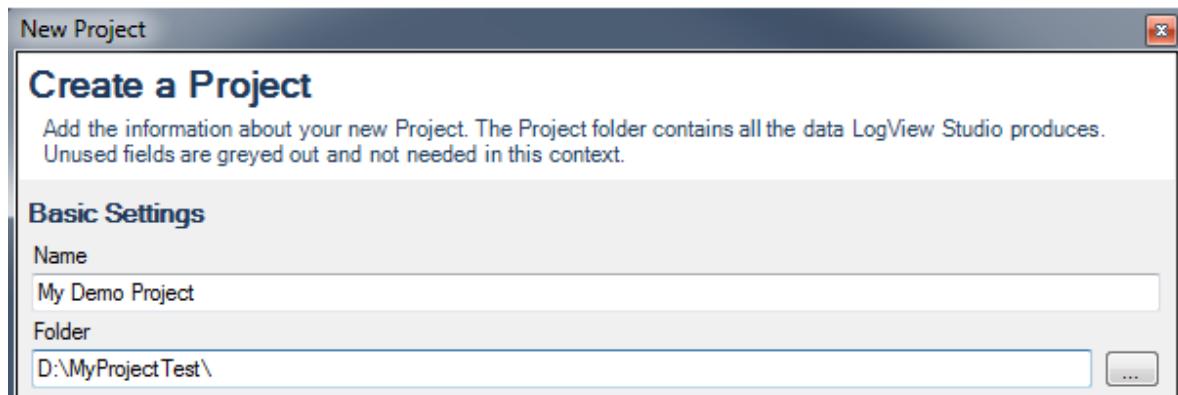
- Start LogView Studio from the Windows Startmenu or start 'LvGui.exe' directly.
- It may happen that [Firewall](#) dialog windows appears on the screen where access to the private network shall be granted.
- Now LogView Studio should appear on the screen with a 'LogView Studio Quick Start' dialog window.

New Project

Click 'Start a new Project' inside the Quick Start Dialog



In the following window 'New Project' enter a Project Name and Folder as location where the project shall be created and saved. Press 'OK' to proceed creating the new Project.

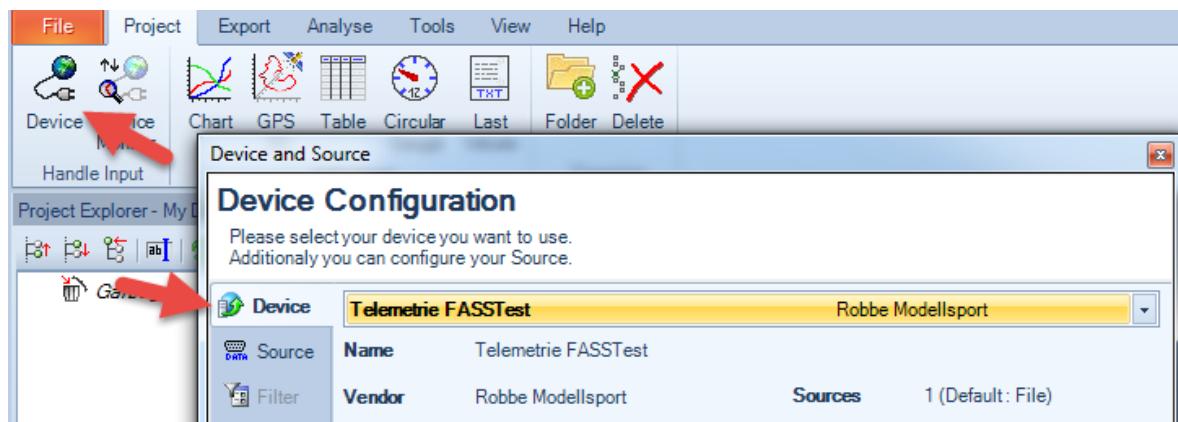


After successfull creation of the Project you should see in the left side window 'Project Explorer' only one entry called 'Garbage'.

Add Device

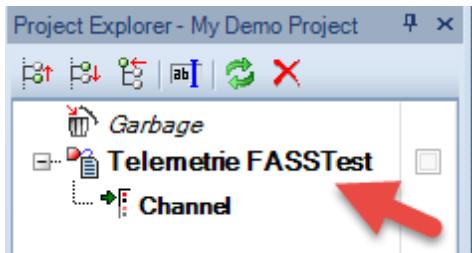
Next step is to add a device you want to read data from and/or process such data.

Select from the Main Menu 'Project' and click in the 'Handle Input' section on 'Device'.



In the following 'Device and Source' Configuration Dialog you can select the device of your choice. (Note that here only those devices are available which have been installed during the first main Installation of LogView Studio. If you need to add other devices you must run the [Update procedure](#).)

Confirm with 'OK'. Now the new device will be added to the previous created Project.

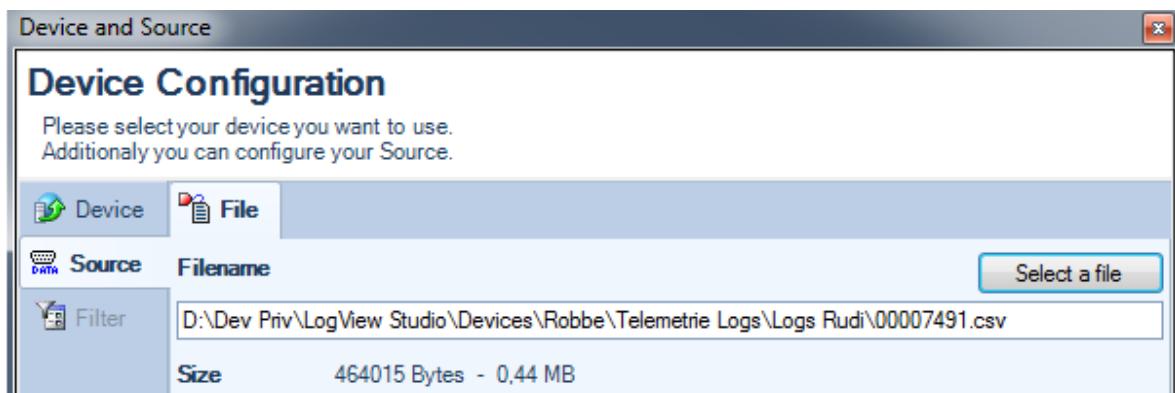
**Note**

We showed this intentionally without Configuring the source.
Normally you can configure Device and Source in one step.

Configure Source

Now the Source (data-source) must be configured for the device. LogView Studio is able to handle a various amount of different data sources like serial, HID, TCP, UDP, Files. Which type of sources are finally available to be configured depends on the selected device. Example: In case of a FASSTest telemetry only Textfiles can be used as source.

With double click on the source 'Telemetrie FASSTest' inside the Project Explorer the 'Device and Source' Configuration Dialog window will appear.



As this device only supports one source 'File' no other type of sources are available. In case a device supports more than one type of sources you can select one source here which will be used for this device.

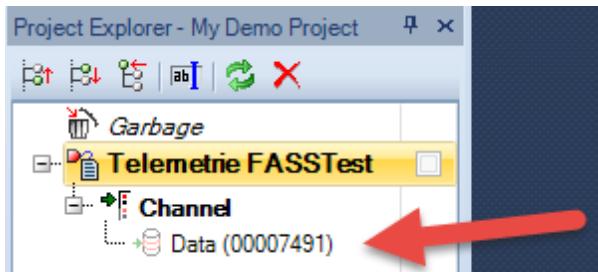
Confirm the Dialog with the 'OK' button and the data import will be started.

Note

Only text and binary files will be read immediately after closing the dialog window. All other type of sources must be started individually by selecting the Checkbox beside the device inside the Project Explorer Tree.



After completing the data import or after starting the device with the Checkbox new elements will appear in the Project Explorer - the data records.



The data records are representing a collection of data within the database. And in a different way from LogView V2 there will NOT be created a related graphic immediately.

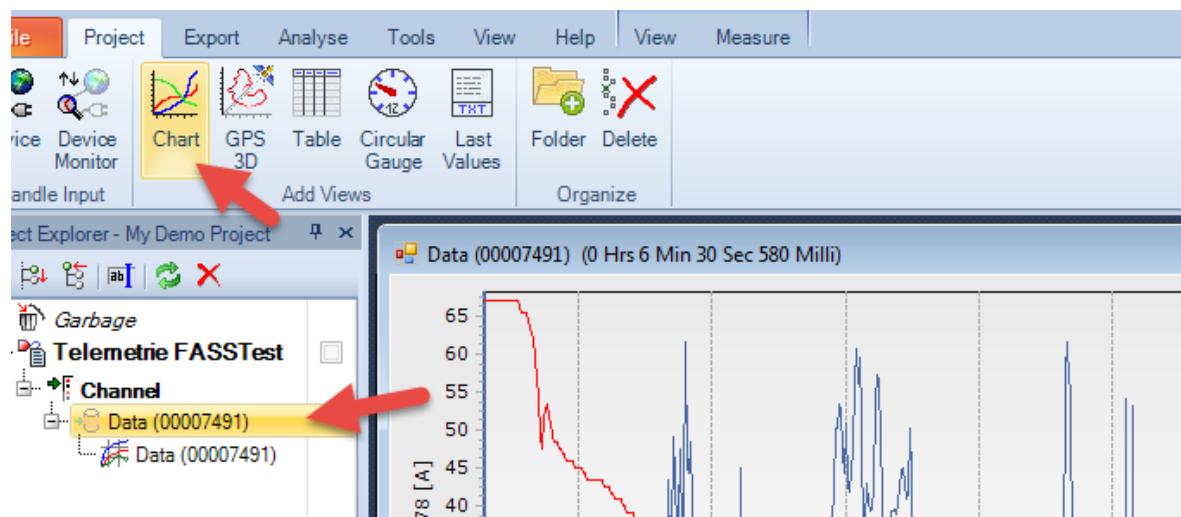
How to create a graphic from a data record will be shown further down.

Import further datafiles

In case you want to import further datafiles for a specific device you can also use the Drag and Drop function to move them from the Windows File Explorer to the relevant device in the Project Explorer of LogView Studio. The data import and processing will start immediately.

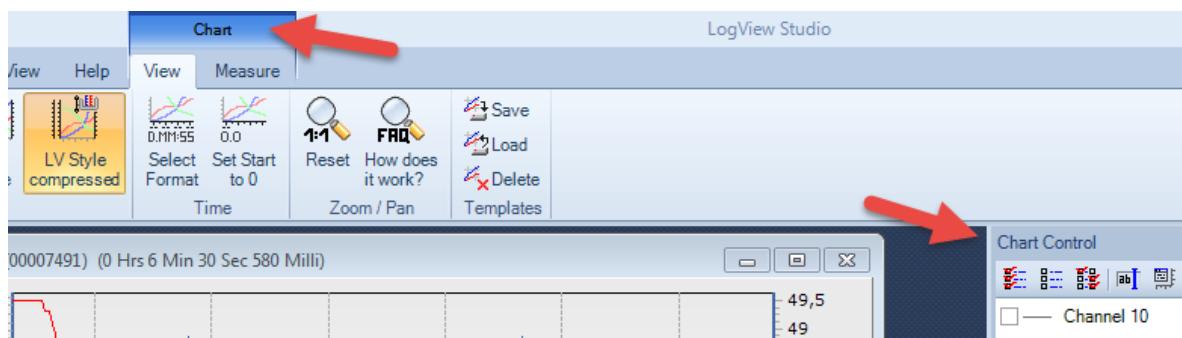
Create static Chart

To create a static chart of a data record mark the relevant data record and then go to the Main Menu 'Project' section 'Add Views' and click on 'Chart'.



Now the selected data record will be shown in a static Chart.

Using the window 'Chart Control' and the menu 'Chart' the graphic can be configured to your own preferences.



Display Live Data

Beside the static graphic (no further changes inside the graphic) LogView Studio can also display live data. This is mainly interesting with devices which send continuously data like battery charger or weather station sensors.

To achieve that you must select a channel of a device in the Project Explorer. Then by selecting 'Chart' or 'Circular Gauge' a Live chart can be displayed which is fed every x seconds with new data.

Load / Reload Project

A Project can be loaded in three different options:

- After starting LogView Studio inside the 'Quick Start' Dialog
- using the List in File Menu
- via File - Open Project ...

Final Remarks

Maybe this Quickstart Introduction seems to be slightly complex on the first view. But once you have created your first project in LogView Studio and have used it you will find it more and more easy to handle. Additionally there are help texts in tool tips on almost all buttons in LogView Studio where the function of the specific button is explained. Within these tool tips you find also direct links to further information inside this Help File.

3.4 Basics / FAQ

This chapter is about basics around LogView Studio. The following information did not cover all aspects but should be a good starting point.

Sections

This chapter is divided into different sections:

- [What is LogView Studio?](#)
- [What is a Project?](#)
- [What is a Device?](#)
- [What is a Source?](#)
- [What is a Channel?](#)
- [What is a Dataset?](#)
- [What is a View?](#)
- [Static and Live View](#)

- [GUI / CLI - What's that?](#)
- [LogView Studio on Linux?](#)
- [LogView Studio and Network](#)

3.4.1 What is LogView Studio?

LogView Studio is a free Software to read and process different data values of various different devices. The main focus is actually on the hobby model building market where many dataloggers, battery chargers and telemetry units are supported by LogView Studio. Beside that you can use LogView Studio also for your own developments. The device itself plays only a minor role. As soon as you know the communication protocol of the device in question you can normally start logging the data.

After successful received / reading and storing the data you can do process visualization in various different graphic formats. Also Export and saving in various different formats is possible.

Features

LogView Studio differs from other Software by some unique selling propositions:

- Any quantity of devices can be used simultaneously (limited only to the number of usable resources of your computer)
- The received data will be stored in a database. This database can be used for your own developments of data analysis or applications.
- The data storage is completely decoupled from the data output process.
- LogView Studio is network-compatible. Data can be logged on different computers and stored in a common project.
- A command line Edition is available and also compatible with LINUX. LogView Studio including the database can even run on a Raspberry Pi.
- Various reports can be open active simultaneously.
- Receiving/reading data can be done through various input ports like serial, network, or directly from files. Even a cyclical request of a webpage could be realized.
- A built-in webserver for fast presentation of the data within a network.

Additional Information

[Quickstart](#) - quick starting a project with one device

[Sample Project](#) with several devices

3.4.2 What is a Project?

A Project in LogView Studio is some kind of a 'container' where all the data are collected and stored. Various elements like devices, data records, graphics etc. can be added to such a 'container' called 'project'. The items are sorted within the 'Project Explorer' window in a tree structure which allows a good overview and control of the whole project.

A project can also be extended by adding individual folders (context menu 'Organize' - 'Insert Folder'). Hereby you can sort and group your data to your individual preferences.

The User is completely free to specify the individual meaning of his project. This is a

very flexible option and there are many different ways possible how to organize a project:

- Collecting all data for a specific device only like battery charger in one project.
- Collecting data from various devices in one project which have a common theme like e.g. all devices and data related to a specific flight model: Battery charger, flight data logger, etc.
- It is also possible to collect and organize all data and devices related to hobby modelling in one project only but this is not recommended as the overview can probably be lost when too many various devices are collected within one project only.

Finally it remains to the User how to organize the various devices and data and which devices are collected in one specific project. LogView Studio gives just a higher-level framework for this purpose.

Note

Only one project can be active at the same time.

Note

The quantity of elements like devices, data records etc. within LogView Studio is unlimited and only restricted by the available computer resources.

Note

LogView Studio saves the data not in one single data file but in a specific project folder.

Additional Information

[Create a new project](#)

[Add a new device to a project](#)

3.4.3 What is a Device?

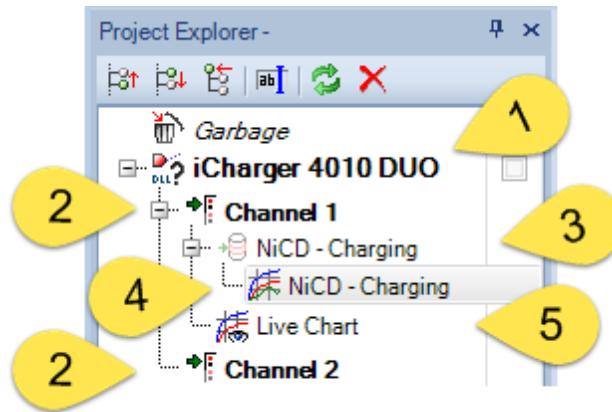
A device in LogView Studio is a hardware unit which can provide data via a specific port to the computer. LogView Studio supports various different type of ports to receive the data in the computer.

Else than the number of active projects (only one at the same time) the quantity of different devices within one project is unlimited and only restricted by the available computer resources. Inside the active project it is possible to use and run multiple devices on different ports simultaneously.

Samples of devices

- data logger
- battery charger
- weather station with various sensors
- Arduino / Microcontroller with sensors

Sample Screen



1 Device

2 Channel

3 Dataset

4 Chart View (static)

5 Chart View (live)

Additional Information

[Quickstart](#)

[How to configure Input-devices within a project](#)

[Sample Project](#)

3.4.4 What is a Source?

Whenever a device is added to a project in LogView Studio it is always necessary to configure this device by selecting a related source for this device. In other words each device needs a specific source from where the data will be sent to the computer. LogView Studio supports a wide selection of various possible sources which can be used for devices:

- **Serial Port** (✉) Data will be received through a serial port.
- **USB HID** (Human Interface Device) (✉) Data will be received through an USB Port. The USB Port is working in HID mode: http://en.wikipedia.org/wiki/Human_interface_device
- **TCP Client** (✉) Data will be read over network from a TCP Server.
- **UDP Receiver** (✉) Data will be read over network from a UDP sender.
- **Text Files** (TXT) Data is from a text file (normally CSV).
- **Binary Files** (BIN) Data is from a binary file.

Note

The feasible sources of a specific device is always preassigned during the creation / programming of the device file. It doesn't make sense to have always all possible sources available. Therefore the options of selecting a source is restricted to those sources which are useful and proper working with that specific device, e.g. it is useless for a battery charger to provide a TCP Client Source - which charger has already a network connection.

Each device can have different sources (if applicable) but only one source of each device can be used at the same time. LogView Studio can not for example read data of a specific device from the serial port and import a Textfile of that device at the same time.

The source can be specified directly when adding a device. When doing the device adding there is another tab in the dialog window called 'Source'. Alternatively a source can also be specified or changed later by double click on the relevant device or by selecting 'Source' inside the device context menu (right mouse click on the device) and opening the dialog window 'Device Configuration'.

Additional Information

[Quickstart](#)

[How to configure Input-devices within a project](#)

[Sample Project](#)

3.4.5 What is a Channel?

Each device which has been added to a project in LogView Studio has at least one Channel. The received data are separated by channels into logical interrelated arrays.

Note

A Channel can neither be moved nor deleted directly as they are a vital part of a device.

The number of available channels is preassigned during the creation / programming of the device file. It can not be changed by the User.

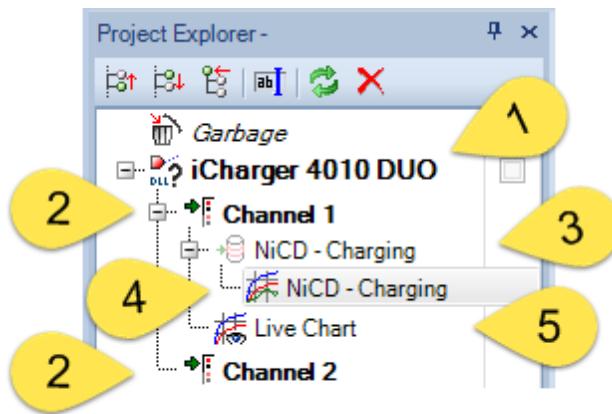
Sample 1

A battery charger which can handle two rechargeable batteries at the same time will have two channels in LogView Studio. The data sent from the charger to LogView Studio will be separated automatically and assigned to the related channel.

Sample 2

A data logger which stores different measured values on a SD-card has normally only one channel in LogView Studio. The data read from the SD-card data file will be assigned to this channel.

Sample Screen



1 Device

2 Channel

3 Dataset

4 Chart View (static)

5 Chart View (live)

3.4.6 What is a Dataset?

A Dataset is a connected block of data which have been created by a device for a specific channel. A Dataset is always related to a specific Channel and is movable to your self created folder within the tree structure of the Project Explorer.

Most easy way to explain and understand the function of datasets is using a sample case study:

A battery charger in a standard case can be used to charge or to discharge a rechargeable battery. After start of the charging process LogView Studio will create a dataset called 'Charging' (see sample screen below). When the charger shifts from Charge to Discharge mode LogView Studio will create a new dataset called 'Discharging'.

(Remark: The final exact naming depends on the selected device and provided data. Normally it contains also the type of the actual connected battery type depending on the actual selected mode settings of the charger. Furthermore it may contain the date and time info when the data or the dataset has been created.)

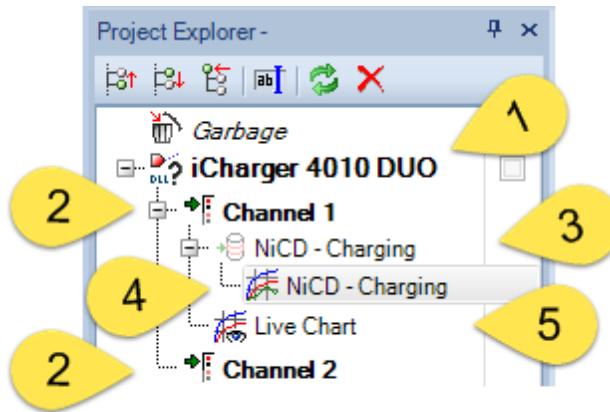
Note

A dataset is a logically accumulated group of data within the database. A dataset is used as a reference for chart views like a graphic screen in order to select and find the relevant data from the database. A dataset itself does not create a graphic (or chart view) automatically and should not be equated with that.

Note

A Dataset will be created automatically by the device after activating the data processing.

Sample Screen



1 Device

2 Channel

3 Dataset

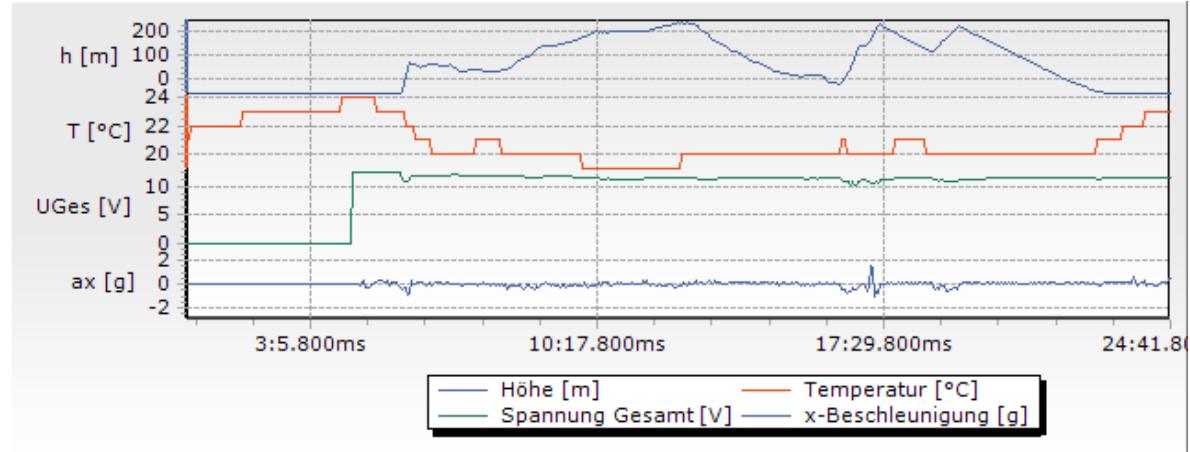
4 Chart View (static)

5 Chart View (live)

3.4.7 What is a Sensor?

A sensor in LogView Studio describes a measuring value which is to be logged and processed. Sensors are dedicated to a specific channel and they are specified normally with the Device (an exemption are the '[Zero Sensor Devices](#)').

The best way to describe sensors is to look at a chart:



Each displayed (or available) value is from the LogView Studio perspective a sensor.

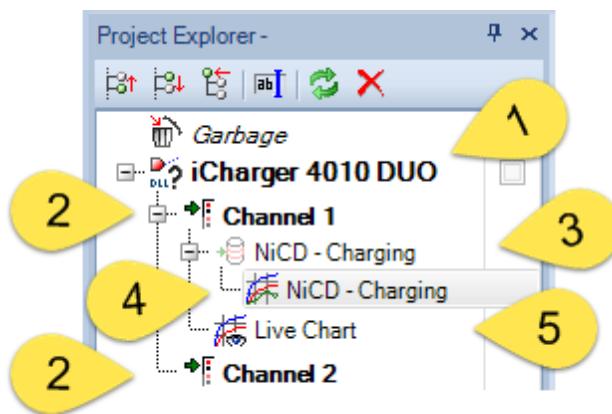
3.4.8 What is a View?

A View is the graphical display of data from a dataset or from a channel. Actually there are following Views available in LogView Studio:

- Chart - data visualization using curves
- GPS 3D - Visualization of GPS Data in a 3D View (only when GPS Data are available)

- Table - Tabular Display of the received data
- Circular Gauge - Display of the actual received data in circular gauge ('Live Gauge')
- Last Values - Display of the actual received data in text form ('Live Text')

Sample Screen



- 1** Device
- 2** Channel
- 3** Dataset
- 4 Chart View (static)**
- 5 Chart View (live)**

Additional Information

Further Information see next chapter '[Static and Live View](#)'.

3.4.9 Static and Live View

If you start recording data with LogView Studio you will notice that there is no visual display after start of recording / reading. The display of your choice has to be added first to a dataset or to a Channel within the specific Project.

Generally there are two different type of Views available:

Static Views

Static Views are created once for a specific dataset and will not change their values afterwards. (Only the graphical appearance can still be adjusted afterwards by the User.)

Live Views

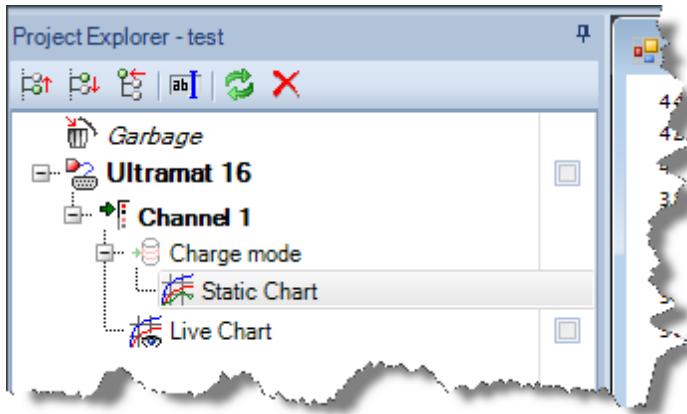
Live Views are assigned to a specific Channel and they are reading intermittently and display the latest data from the database of the selected Channel.

Note for Live Views

Live Views must be activated (started) by the User similar as to be done with a device. In the Project Explorer window there is a checkbox on the right hand side of each Live View entry which can be used to start and stop the Live View

thread. If it is not possible to start and stop these Live Views it may lead to a performance problem.

Sample Screen



Inside this Project a device has been added (Ultramat 16) which has one Channel (Charge mode data line).

The device was activated earlier and has recorded the charge process into the self created dataset 'Charge Mode'.

In order to see the graphic of the received data there has been added a static View named 'Static Chart'. This chart contains all data of the related dataset. Once the Chart has been created the data will not be changed anymore by LogView Studio.

The Live Chart is assigned to a Channel. Here are displayed the latest received and stored data of Channel 1. The graphical display Live Chart will be updated continuously when activated (started). The display is slightly similar to an oscilloscope - but considerably slower.

3.4.10 Garbage - the trash can

Whenever any element will be deleted from the Project they will be removed first to the Garbage (trash can). This function is similar to the Windows 'Recycle Bin'. With this function you can restore elements which have been deleted by mistake.

Note

Garbage is the first and only element within a new created project. This is a good indication for a successful creation of a new project.

3.4.11 GUI / CLI - What's that?

From time to time you will see the expression 'GUI' or 'CLI' (or even 'CMD') Tool.

Explanation GUI

GUI is the abbreviation of Graphical User Interface. In other words it's a software which can present a graphical display instead of text only. From LogView Studio the GUI is the most commonly used and known application.

Note

The file which starts the LogView Studio GUI edition is called 'LvGui.exe'.

Explanation CLI

CLI is the abbreviation of Command Line Interface and CMD or Command Tool means basically he same. It is a software program which runs in pure Textmode and will run typically within a Shell. With LogView Studio CLI there are no graphical outputs possible.

Note

The file which starts the LogView Studio CLI edition is called 'LvCmd.exe'.

Comparison and differences

Following table shows the different functions of GUI und CLI

Function	GUI (LvGui.exe)	CLI (LvCmd.exe)
Quantity of parallel useable Devices	unlimited - restricted by computer resources only	1
graphical Output via Views	yes - limited by computer resources only	nil
Running on Linux	no	yes - using Mono Framework
Needs of resources	high - depends on the extent of the Project	low
Debug Properties	User Interface must be started before	fast and direct

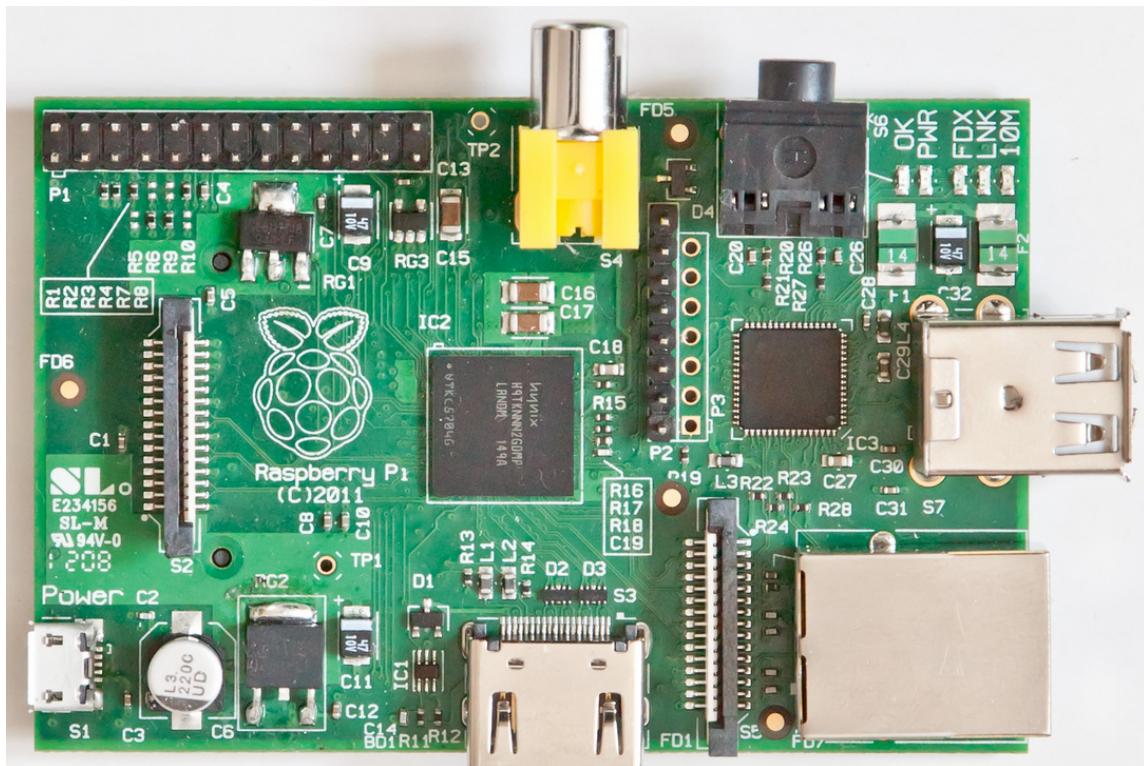
Basically it can be said that both versions have their own advantages and disadvantages. It always depends on the result what the User expect when using LogView Studio.

For the 'standard User' the GUI is in most cases and will be the competitive edge.

3.4.12 LogView Studio on Linux

The CLI (Command Line Interface) of LogView Studio can easy be used on Linux systems with little efforts only. To do so only the Mono Framework is necessary (similar to the .Net Framework under Windows). As said earlier on Linux only the CLI can work. But the CLI edition of LogView Studio has additionally a built-in Webserver function in order to do graphical analyzing of the data.

This idea was born when the Raspberry Pi computer system was available on the market.



During tests of this little low-cost system it was confirmed that even with such a simplified hardware platform it is possible to get the LogView Studio running. With the CLI Edition of LogView Studio there is now a solution available to run a project on a very energy efficient system even for a long period.

This is a very good solution for long-term data recording.

But LogView Studio also works flawless on a standard Linux system without any major problems.

Data analyzing with Windows

It should be mentioned here that data analysis can be done also on a computer with Windows operating system. The LogView Studio GUI can connect to an existing project (similar as the CLI) locally or using a network to perform data visualization.

Additional Information

Raspberry Pi: <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

Linux: <http://www.linux.org/>

3.4.13 LogView Studio and Network

LogView Studio is working excellent also in a network. That means primarily that various LogView Studio programs can connect to a common project via network. This is possible within a local network LAN / WLAN or even the world-wide Internet.

Note

For a proper use of the Internet solution you must allow the opening a port of the

router for an IP address and in best case you also have configured a DynDNS adress. To explain these procedures in detail will go beyond the task of this manual - but your favorite searchengine like Google should give you sufficient further information needed to proceed:

*Port forwarding : http://en.wikipedia.org/wiki/Port_forwarding
DynDNS : http://en.wikipedia.org/wiki/Dynamic_DNS*

In fact each LogView Project is already a 'LogView Studio in a network'. Each project has its own database and LogView Studio connect to this database using your local network. When using such a standard project case the User is not forced to add further configurations.

Benefits

The advantage of LogView Studio within a network is obvious: data can be logged into a common project from various locations and on the other hand the data from the common project can be read out from different locations.

In other words you can compare the networking of LogView Studio also with the setup of an internet newsgroup. There is one server located somewhere in the world and connected to the internet where the newsgroup software is running and providing the news - that is in our case the computer with the LogView Studio project collecting and providing the data. In the other case various different User from various different locations can connect to a newsgroup and read or write information. In case of the LogView Studio that would be further computer providing data to the one and common project or read out data from this project for further local processing.

Samples

Here are some sample configurations to explain the networking with LogView Studio:

- "Basement Computer"

You have a computer in the basement running and collecting / reading data from some sensors e.g. a battery charger. From time to time you want to check the charging progress on the 'Basement Computer' but you are sitting in the living room in the second floor. Now it's not longer necessary to run down to the 'Basement computer' instead you build up a LAN or WLAN connection from your living room computer to the 'Basement computer'. Now all the information from project running on the 'basement computer' are also available on your living room computer.

- Weather station

You want to collect and read out weather data from two different locations using a small Arduino system. Both stations can (also via Internet) provide their local data into a common LogView Studio project which is running on your computer at home.

- Linux with more than one device

With Linux you can use the LogView Studio 'CLI' edition. The limitation on this is that you can use only one device. But if you start another LogView Studio CLI instance then this can connect to the same project and drop its data there as well.

Network devices

Beside the option of multiple connections to one common project LogView Studio offers

also the option for different devices to provide their data via network into one common project. For such purpose the 'OpenFormat' offers simple options to connect a hardware to LogView Studio using a network. Further details you can find down under '[Development](#)'.

Internet of Things

There is actually a big hype in the Internet called 'Internet of Things' (http://en.wikipedia.org/wiki/Internet_of_Things). It is about advanced connectivity of devices, systems and services that goes beyond machine-to-machine communications (M2M) and covers a variety of protocols, domains and application e.g. washing machine can send an email when the washing program is finished.

From the view of LogView Studio it can be said that there are lots of 'Things' available which can provide data. And LogView Studio itself can also work as distributor of data in the Internet.

If you like the 'Internet of Things' you will see that LogView Studio is also the flexible and multifunctional application for such purpose.

Additional Information

[Network Sample](#)

3.5 User interface

This chapter is about the user interface. You will get detailed information about each menu item.

Sections

This chapter contains following thematic sections:

- [Tool Fenster](#)
- [MDI Fenster](#)
- [File Menü](#)
- [Project Menü](#)
- [Export Menü](#)
- [Analyse Menü](#)
- [Tools Menü](#)
- [View Menü](#)
- [Hilfe Menü](#)

3.5.1 Tool Windows

LogView Studio has a few tool windows like for example the "Project Explorer". These windows can be docked to the side of the User Interface, minimized or even completely closed.

Note

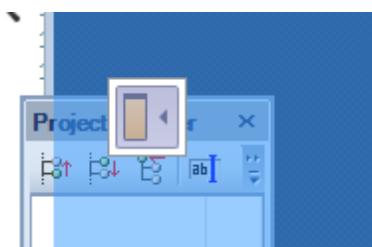
LogView Studio has a few tool windows like for example the "Project Explorer". These windows can be docked to the side of the User Interface, minimized or even completely closed.

Docking

To dock a window select the title row of the window with the left mouse key and hold the key. Then link areas will be displayed at the frame borders as target to where the selected window can be moved.

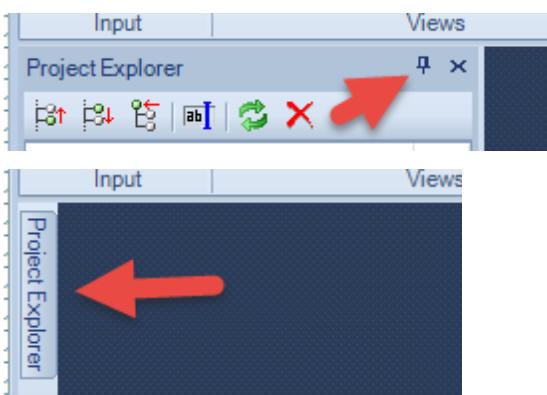


The light blue preview show the new area of the window.



Minimize

With the 'Pin'-tool in the title row the tool windows can be minimized to a small tab.



By selecting the tab the tool window will reappear on the screen. If then the 'Pin'-tool will be selected again the window remains visible in its normal size.

Close

A tool window can be closed completely by using the 'x' in der title row.

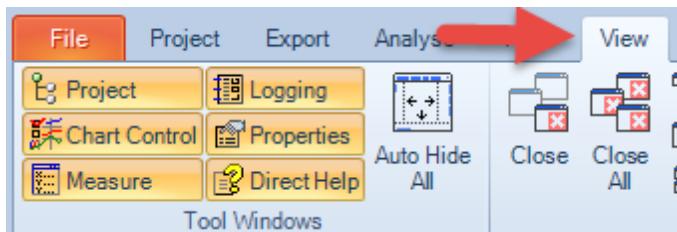


Note

Closed windows can be reopened by selecting the related switch in the 'Tool Window' section of the 'View' menu.

View Menu

The 'View' Menu offers the option to switch on/off the tool windows. Furthermore all central windows can be rearranged in various options.



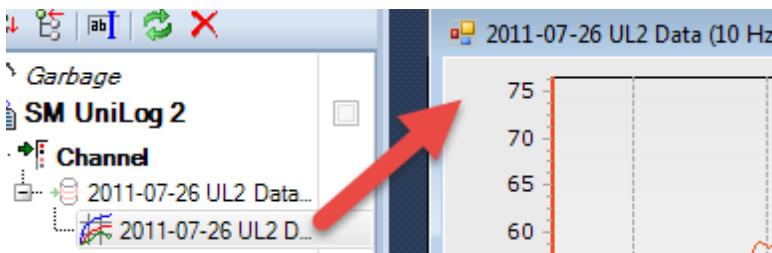
3.5.2 MDI Windows

LogView Studio use MDI = Multiple Document Interface to display the Views. That means only that the application can open any number of View windows.

Note

The maximum effective quantity of View depends on the available resources of the computer.

Every MDI windows in LogView Studio has always a related entry in the Project Explorer.



Switching the active MDI window in the application will also select (highlight) the related entry in the Project Explorer automatically.

A double click on an entry in the Project Explorer opens the related MDI window (if available).

Minimizing

A window in LogView Studio can be minimized using the standard system Icons.



Maximize

A window in LogView Studio can be maximized using the standard system Icons.

**Note**

The system icons of a maximized window will appear in the Main menu row right beside the Help-'?'-icon.

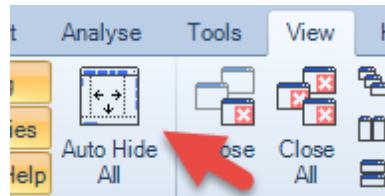
**Close**

To close a window select the 'X'-Icon.

**Maximize the central display area**

In case you need a maximum of available central display area you can arrange this in three simple steps:

- Minimize all Tool windows



- Hide Main Menu ribbon (top right corner left beside the Help-'?')



- Maximize the View

**3.5.3 File**

This chapter explains the File menu and its various functions.

Note

The File functions are available in the File menu as well as in the LogView Studio Wizard. The functions are always identical.

Sections

This chapter contains following thematic sections:

- [New Project](#)
- [New Remote Project](#)
- [Open Project](#)
- [Save Project](#)
- [Close Project](#)

3.5.3.1 New Project

Before LogView Studio can record and analyze data a [Project](#) must be available. There are two options to create a new project:

- using the LogView Studio Wizard which appears after starting the program
- via the 'File' menu - 'New' > 'Project'

The procedure is the same in both options. Therefore only the option 'File' menu is described as follows:

- Select 'New' in the menu 'File' or select 'Project' in the following sub menu (alternatively you can select 'New Project' of the LogView Studio Wizard options)



- Enter a project title in the LogView Studio Wizard and select a suitable folder of the hard drive's Filesystem. (The folder must be empty to be available for a project creation.)

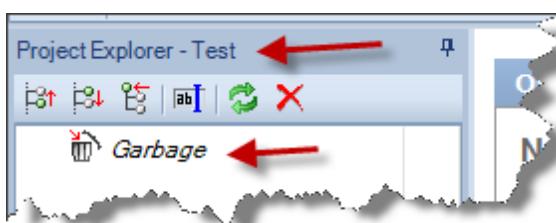
Note

A new folder can be created using the button on the right side of the window. (The new folder will be created as sub folder of the actual marked (highlighted) folder in the tree structure.)

- In case empty folders are available for LogView Studio as location for a new project they are marked with a green dot.
-
- Select 'Finish' in the Wizard to proceed with the project creation
 - The creation progress in LogView Studio can be observed in the following window and in the logging tool window.

Project created?

When the project has been created successfully the headline in the Project Explorer will change and display the Project name. Furthermore the first and only entry in the Explorer tree shall be the 'Garbage' trash can.



3.5.3.2 New Remote Project

LogView Studio can connect to a project which is running on a remote computer. It is possible to read out the data of the remote project but also to send (write and store) data of the local system to the project on the remote system.

Note

Working with LogView Studio in a network is explained more detailed in a separate chapter.

In order to get this feature alive it is necessary that LogView Studio creates a project on the local / own computer. Somehow this sounds like a paradoxical step as there is already a project running on the remote computer. But the reason is that LogView Studio needs to save some data locally like for example settings or views. For this purpose the existence of a local project structure is mandatory.

Note

The difference to a normal local project is that using a remote project is not necessary to start a local database.

The procedure to create a new remote project is almost similar to the creation of a (normal) local project. Only a few additional settings have to be done.

- Select 'New' > 'Connect to remote Project' in the menu 'File' (alternatively you can select 'Remote Project' of the LogView Studio Wizard options)



- Enter a project title in the LogView Studio Wizard and select a suitable folder of the hard drive's Filesystem. (The folder must be empty to be available for a project creation.)

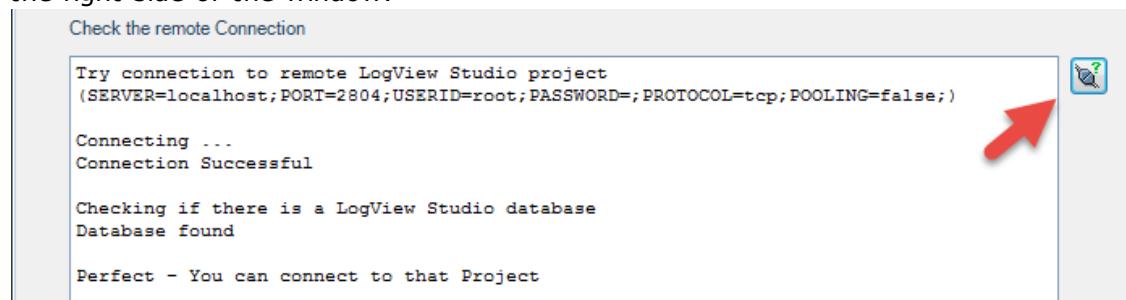
Note

New folder can be created using the button on the right side of the window. (The new folder will be created as sub folder of the actual marked (highlighted) folder in the tree structure.)

- In case empty folders are available for LogView Studio as location for a new project they are marked with a green dot.
- Select 'Next' to proceed to the next dialog page
- Now please enter the Hostname or the IP address of the remote computer respectively. The default values of Port, User and Password are normally correct and must be adjusted only in special cases.

Hostname / IP-Adress	Port
localhost	2804
User	Password
root	

- If it is possible to connect to the remote database can be tested using the button on the right side of the window.



The screenshot shows a window titled 'Check the remote Connection'. It displays the following text:

```
Try connection to remote LogView Studio project
(SERVER=localhost;PORT=2804;USERID=root;PASSWORD=;PROTOCOL=tcp;POOLING=false;)

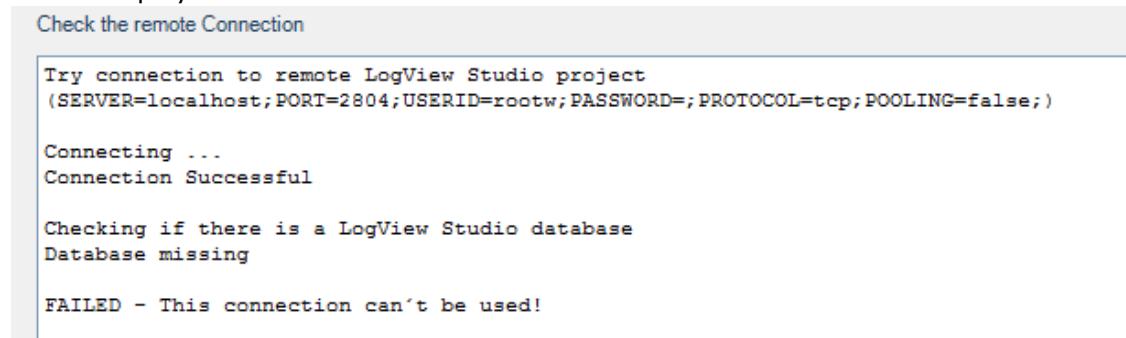
Connecting ...
Connection Successful

Checking if there is a LogView Studio database
Database found

Perfect - You can connect to that Project
```

A red arrow points to the 'Check' button in the top right corner of the window.

In case the remote LogView Studio database is not available a related message will be displayed.



The screenshot shows a window titled 'Check the remote Connection'. It displays the following text:

```
Try connection to remote LogView Studio project
(SERVER=localhost;PORT=2804;USERID=rootw;PASSWORD=;PROTOCOL=tcp;POOLING=false;)

Connecting ...
Connection Successful

Checking if there is a LogView Studio database
Database missing

FAILED - This connection can't be used!
```

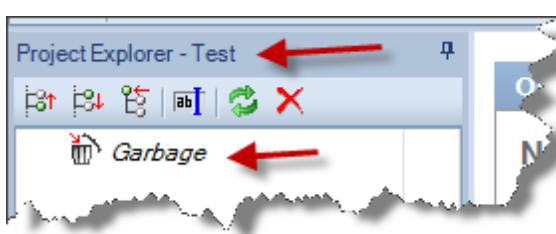
Note

To allow the local computer connecting to the remote computer it is necessary to store / activate the IP address of the local computer in the database of the remote project. Further Information to this are available in the chapter 'Samples - Network sample'.

- Select 'Finish' in the Wizard to proceed with the remote project creation
- The creation progress in LogView Studio can be observed in the following window and in the logging tool window.

Project created?

The project has been created successfully when the headline in the Project Explorer shows the Project name and the first entry in the Explorer tree is the 'Garbage' trash can. Furthermore the devices and datasets created on the remote computer shall be visible.



3.5.3.3 Open Project

To open a project in LogView Studio can be done in two different ways:

- Open Project ...
- Recent Project

Open Project ...

Select 'Open Project' in the 'File' menu. A new window will display the folder structure of the available hard drives. In case there is somewhere a valid LogView Studio project in one of the folders such folder symbol is yellow and marked with a green dot. Otherwise the folder symbol is grey.



After selecting a valid project folder it can be continued with selecting the 'Finish' Button at the bottom of the dialog window. The project will be opened and the progress of opening can be monitored in the next screen.

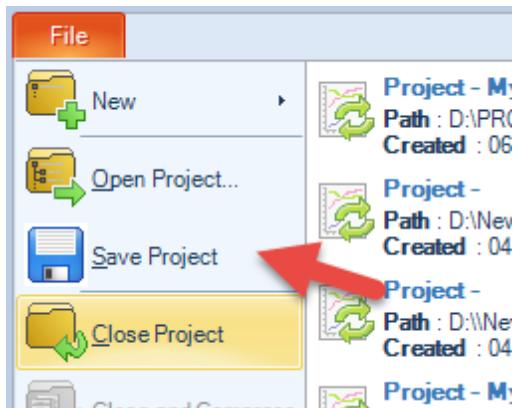
Recent Project

Up to ten (10) recent projects will be displayed in the list of available recent projects. Here a recent project can be selected and with a click on 'Finish' the selected recent project will be opened.

Alternatively the project can also be opened with a double click on the selected project name.

3.5.3.4 Save Project

Select 'Save Project' in the 'File' menu to save an active open project.

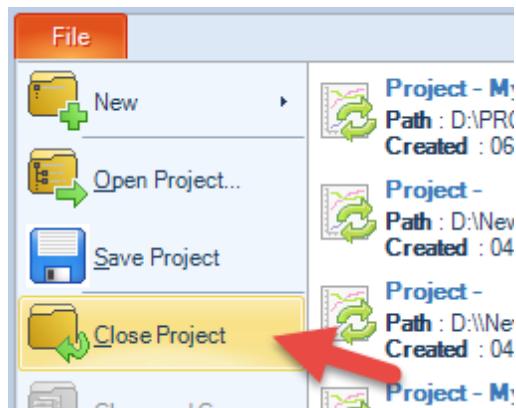


Note

During the saving process mainly the open views and settings will be saved. Recorded or imported data will be saved immediately direct to the database.

3.5.3.5 Close Project

To close an open active project select 'Close Project' in the 'File' menu.



Note

All devices recording / receiving data and all Views displaying the data will be stopped automatically before closing the project.

3.5.4 Project

This chapter describes the Project Menu and his sub functions.

Sections

This chapter contains following sections:

- [Device](#)
- [Device Monitor](#)
- [Chart](#)
- [GPS 3D](#)
- [Table](#)
- [Circular Gauge](#)
- [Last Values](#)
- [Folder](#)
- [Delete](#)

3.5.4.1 Device

To read, log or import data into LogView Studio there must be added a device to the active project first. With the dialog 'Device and Source' the device can be selected and the source for this device can be configured.

In LogView Studio a 'device' is always the compulsory base for any further operation.

Menu Section

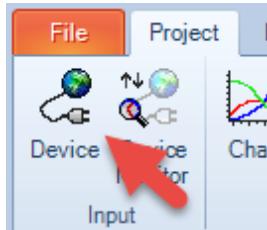
Input

Hierarchy level

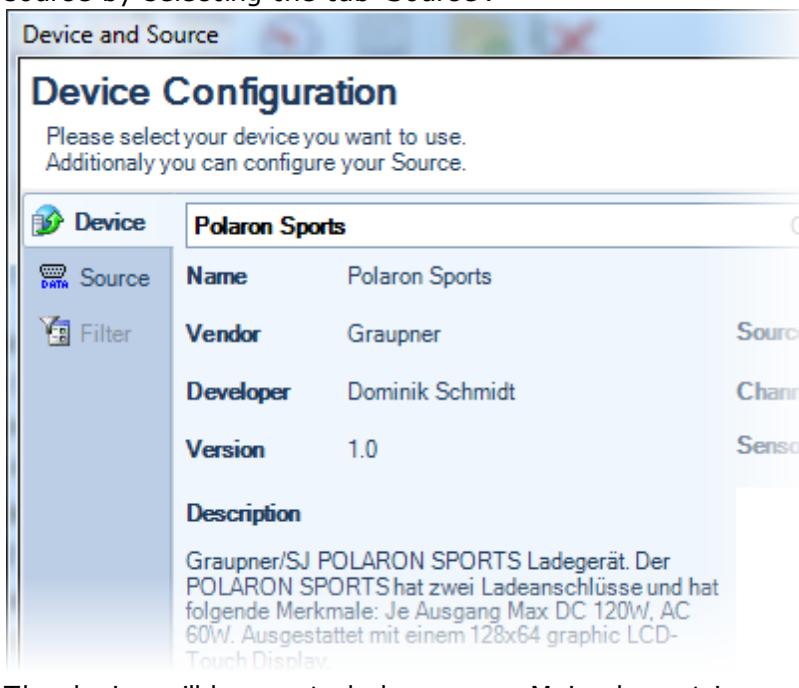
Main element of a Project

Procedure 'Device'

- Requires an existing project (reloaded or new created)
- Select 'Device' in the Input Section of the Menu 'Project'



- Select a device within the dialog 'Device and Source'. Then specify the data source by selecting the tab 'Source'.



- The device will be created always as a Main element in a project with all its connected Channels.



Note

The quantity of connected devices in LogView Studio is unlimited. Two or more different devices can read / receive and store data simultaneously. A restriction is only given by the number of available ports of the computer or the computer resources.

Additional Information

[Working with LogView Studio - Device](#)

3.5.4.2 Device Monitor

The 'Device Monitor' function in LogView Studio is available to monitor the incoming and outgoing data of a selected device. Furthermore it can be used to send your own data to the device.

The function of the Device Monitor is primary for the ambitious User of LogView Studio User who works with e.g. the OpenFormat and self developed hardware. For the normal (standard) use of LogView Studio this component gives not much extra advantages.

Menu Section

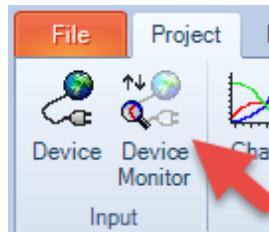
Input

Hierarchy level

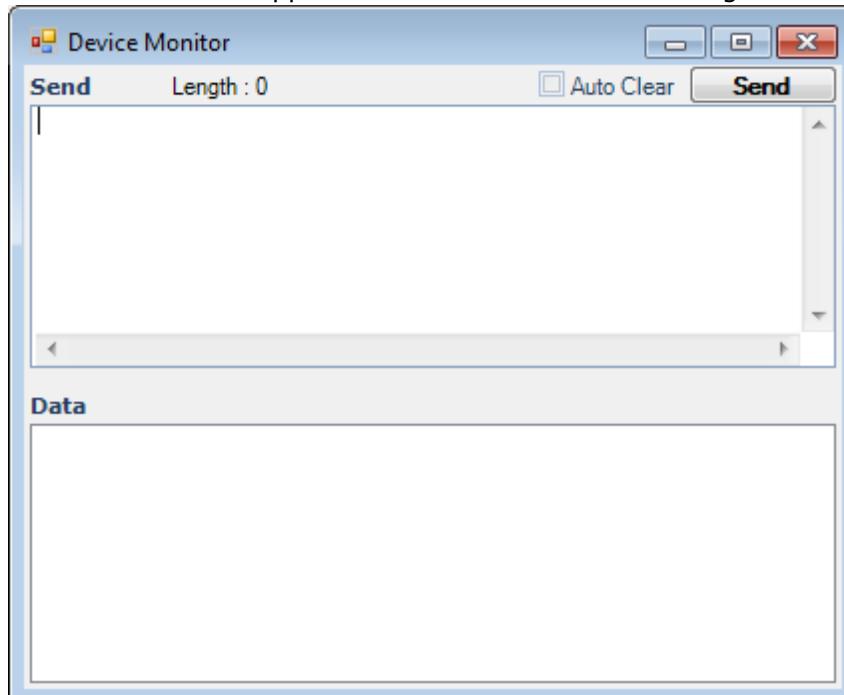
Sub element of a Device

Procedure 'Device Monitor'

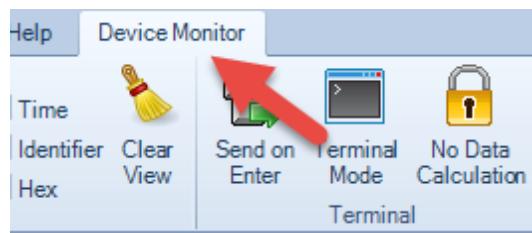
- Select a device within your project first.
- Select 'Device Monitor' in the Input Section of the Menu 'Project'



- A new window will appear called 'Device Monitor' in LogView Studio.



- Once this window is activated (and always when the Device Monitor entry in the Project Explorer tree will be selected) a separate additional main menu tab 'Device Monitor' will show up and is available with various specific options.



This menu 'Device Monitor' offers numerous options to observe data, manipulate outgoing data, or to protocol the data.

Additional Information

[Working with LogView Studio - Device Monitor](#)

3.5.4.3 Chart

The function 'Chart' is the data visualization as graphical display using curves. It is divided in static Charts and Live Charts.

Menu Section

Views

Hierarchy level

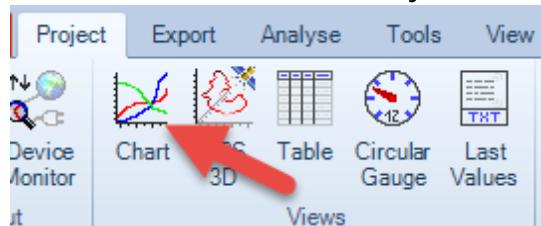
static Chart : Sub element of a Dataset

live Chart : Sub element of a Channel

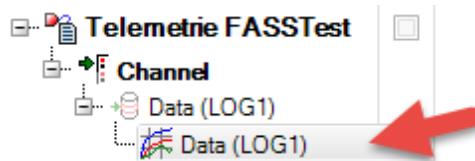
Procedure 'static Chart'

A static View is created once for a specific dataset and will not change their values afterward. (Only the graphical appearance can still be adjusted afterward by the User.)

- Select a dataset in the Project Explorer (shall be highlighted)
- Select 'Chart' in the Menu 'Project'



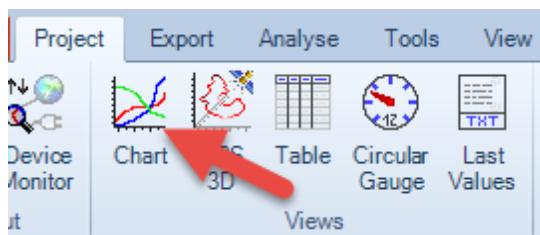
- The static Chart will be created as a sub-element of the dataset



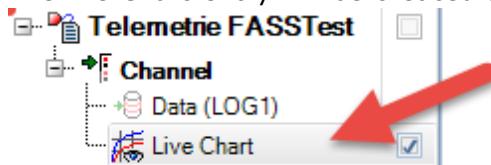
Procedure 'live Chart'

A Live Chart always show the latest received (and stored) data of one Channel of a specific device. The graphical display update can be configured using 'Interval (ms)' and 'X Value Count'.

- Einen Channel im Project anklicken



- Select a Channel in the Project Explorer (shall be highlighted)
- The live Chart entry will be created as a sub-element of the Channel.



- Live Views must be activated (start and stop function) by the User with the checkbox on the right hand side of the Live View entry in the Project Explorer.

Note

The quantity of charts (static or live) within LogView Studio is unlimited. It is also possible to create more than one chart for a specific dataset.

Additional Information

[Working with LogView Studio - Chart - Layout](#)

[Working with LogView Studio - Chart - Templates](#)

[Basics / FAQ - Stativ and Live View](#)

3.5.4.4 GPS 3D

The function 'GPS 3D' will display data with embedded GPS information in a 3D diagram. The GPS 3D View shall neither substitute the related function in Google Earth nor has it a comparable scope of operational functions. This option is only to display the GPS Track in a simple and easy way from various sides.

The GPS View shows always a full completed dataset. Once the data have been read to display the graphic Chart they can not be changed anymore - beside some visual options.

To create this GPS 3D View it is necessary that GPS coordinates are available inside the dataset. Normally they are specially marked direct from the device itself. LogView Studio will detect and analyze these informations accordingly. In case there are no GPS data available or not detected, a related message will appear.

Menu Section

Views

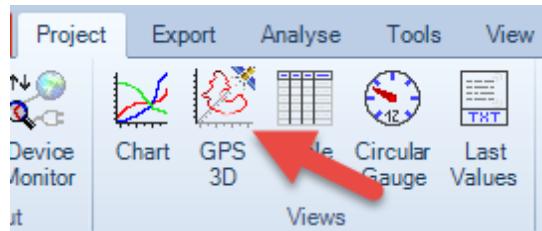
Hierarchy level

Sub element of a Dataset

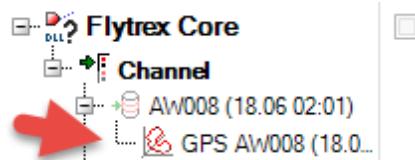
Procedure 'GPS 3D'

- Select a device within your project

- Select 'GPS 3D' in the Views Section of the Menu 'Project'



- The GPS View chart will be created as Sub-element of the dataset



Note

The quantity of GPS 3D Views in LogView Studio is unlimited. It is also possible to create more than one GPS 3D View for the same dataset.

3.5.4.5 Table

The View 'Table' can be used to display Datasets as spreadsheet. The dataset will be read out once from the database and will be displayed in form of a spreadsheet.

The 'Table' shows always a full completed dataset. Once the data have been read to display the spreadsheet they can not be changed anymore.

Note

Using the 'Table' View function the data values of the database itself can not be changed.

Menu Section

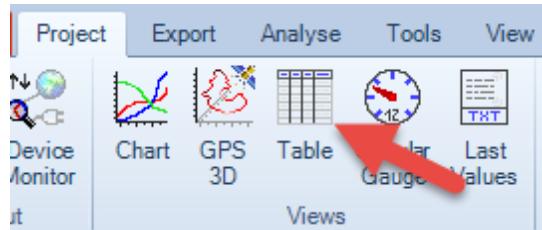
Views

Hierarchy level

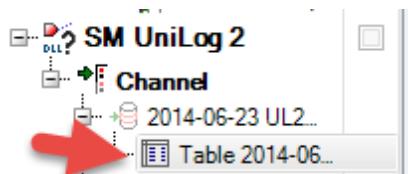
Sub element of a Dataset

Procedure 'Table'

- Select a Dataset within your project (shall be highlighted)
- Select 'Table' in the Views Section of the Menu 'Project'



- The Table View spreadsheet will be created as Sub-element of the dataset

**Note**

The quantity of Table Views in LogView Studio is unlimited. It is also possible to create more than one Table View for the same dataset.

3.5.4.6 Circular Gauge

The Live View 'Circular Gauge' can be used to display values of a Channel as circular Gauge. The data values will be read out from the database periodically and displayed as circular gauge.

Menu Section

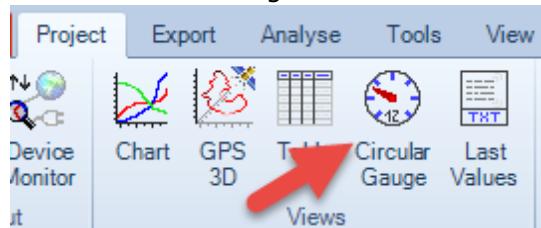
Views

Hierarchy level

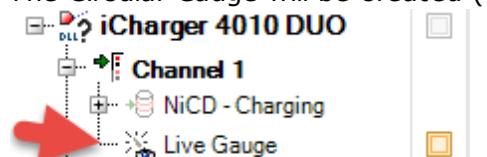
Sub element of a Channel

Procedure 'Circular Gauge'

- Select a Channel within your project (shall be highlighted)
- Select 'Circular Gauge' in the Views Section of the Menu 'Project'



- The Circular Gauge will be created ('Live Gauge') as sub-element of the Channel.



- The Live View of the Circular Gauge must be activated (start and stop function) by the User with the checkbox on the right hand side of the 'Live Gauge' entry in the Project Explorer.

Note

The quantity of Gauge Views in LogView Studio is unlimited. It is also possible to create more than one View for the same Channel.

3.5.4.7 Last Values

The Live View 'Last Values' can be used to display the latest data of a channel (most recent timestamp) of a Channels inside a textwindow. The data values will be read out

from the database periodically and displayed accordingly.

Menu Section

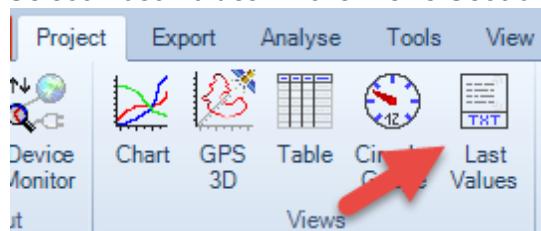
Views

Hierarchy level

Sub element of a Channel

Procedure 'Last Values'

- Select a Channel within your project (shall be highlighted)
- Select 'Last Values' in the Views Section of the Menu 'Project'



- The Last Values View will be created ('Live Text') as sub-element of the Channel.



- The Live View of the 'Last Values' must be activated (start and stop function) by the User with the checkbox on the right hand side of the 'Live Text' entry in the Project Explorer.

Note

The quantity of Text Views in LogView Studio is unlimited. It is also possible to create more than one Text View for the same Channel.

3.5.4.8 Folder

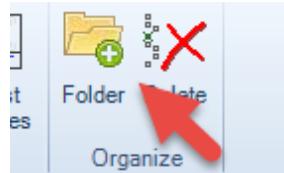
A Folder is used to organize the Project. A Folder can also have multiple sub folders. Only Datasets can be moved into Folders using the Drag and Drop function to arrange a better overview of a the Project. Devices, Channels or Live Views can not be moved. In case a Dataset will be moved to a Folder, all related existing View of that Dataset will be moved as well.

Menu Section

Organize

Procedure

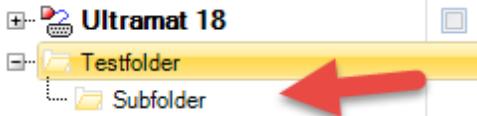
- Select 'Folder' in the 'Organize' Section of the Menu 'Project'



- In case there was no Folder selected (highlighted) before, a new Root Folder will be created.



- In case there was a Folder selected (highlighted) before, LogView Studio will create a new sub folder.



Note

The Folder Function is also available in the context-menu of the Project Explorer ('Insert Folder'). Folders can only be created below (or parallel to) the Device tree hierarchy but never direct inside the Device structure.

3.5.4.9 Delete

With the 'Delete' Function selected elements of a project can be deleted.

Note

'Garbage' (the trash can) and Channels can not be deleted: The 'Garbage' entry itself can not be deleted at all, Channels can only be deleted together with their related Device.

Menu Section

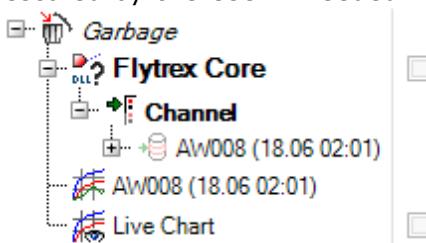
Organize

Procedure

- Select the element in the Project Explorer shall be deleted
- Select 'Delete' in the 'Organize' section of the Menu 'Project'



- The selected element will be moved to the 'Garbage'. From here it still can be restored by the User if needed.



- Elements inside the 'Garbage' trash will be deleted finally only by another 'Delete' procedure from there. The process is comparable with the 'Recycle Bin' function in Windows.

Note

The 'Delete' Function is also available in the context-menu of the Project Explorer.

Additional Information

[Garbage - The trash can](#)

3.5.5 Export

This chapter explains the Export menu and its functions.

LogView Studio offers a wide field of options to export data as graphic diagram or in data format (e.g. CSV).

Abschnitte

This chapter contains following thematic sections:

- [Chart - Graphic](#)
- [Chart - CSV](#)
- [Dataset - CSV](#)
- [Dataset - XML](#)
- [Dataset - JSON](#)
- [Dataset - KML](#)
- [Application Screenshot](#)
- [Desktop Screenshot](#)

3.5.5.1 Chart - Graphic

The selected Chart will be exported as graphic file. Various graphic export formats are available.

Menu Section

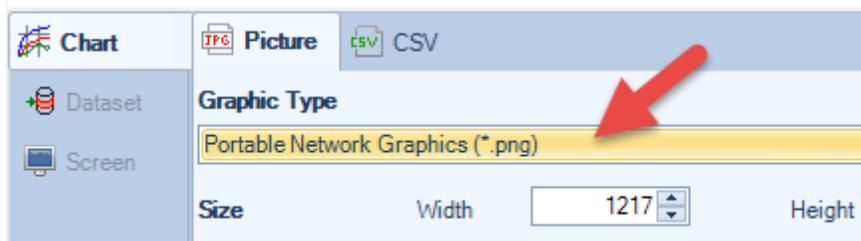
Chart

Procedure

- Select a Chart Element in the Project Explorer
- Select 'Graphic' in the Menu 'Export'
- Select your preferred Exportformat (use optional the scroll down menu):

Data Export

Export your Charts or Datasets as Image or Datafile.
Use Copy for the Clipboard and Save to store a file.



- With 'Copy' secure the graphic export to the clipboard (and continue processing of

your choice using the clipboard memory), or select 'Save' to save the graphic as selected file.

3.5.5.2 Chart - CSV

Export the selected chart as CSV (Comma-separated values).

Menu Section

Chart

Procedure

- Select a Chart Element in the Project Explorer
- Select 'CSV' in the Menu 'Export' (Chart section)
- The Data Export dialog window offers some additional selectable options to modify the final export, like title, names colors. Try out which is the best selection for your needs.
- With 'Copy' secure the graphic export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the graphic as selected file.

3.5.5.3 Dataset - CSV

Export of the selected Dataset as 'CSV' (Comma-separated values).

Menu Section

Dataset

Procedure

- Select a Dataset Element in the Project Explorer
- Select 'CSV' in the Menu 'Export' (Dataset section)
- The Data Export dialog window offers some additional selectable options to modify the final export, like name, value types etc. Try out which is the best selection for your needs.
- With 'Copy' secure the data export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the data as file.

3.5.5.4 Dataset - XML

Export of the selected Dataset as 'XML' (Extensible Markup Language).

Menu Section

Dataset

Procedure

- Select a Dataset Element in the Project Explorer
- Select 'XML' in the Menu 'Export' (Dataset section)
- The Data Export dialog window offers some additional selectable options to modify the final export result. Try out which is the best selection for your needs.
- With 'Copy' secure the data export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the data as file.

3.5.5.5 Dataset - JSON

Export of the selected Dataset as 'JSON' (JavaScript Object Notation).

Menu Section

Dataset

Procedure

- Select a Dataset Element in the Project Explorer
- Select 'JSON' in the Menu 'Export' (Dataset section)
- The Data Export dialog window offers some additional selectable options to modify the final export result. Try out which is the best selection for your needs.
- With 'Copy' secure the data export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the data as file.

3.5.5.6 Dataset - KML

Export of the selected Dataset as 'KML' (Keyhole Markup Language).

Note

This export type of file is only applicable for datasets which contains also GPS coordinates data (beside other values).

Menu Section

Dataset

Procedure

- Select a Dataset Element in the Project Explorer
- Select 'KML' in the Menu 'Export' (Dataset section)
- The Data Export dialog window offers some additional selectable options to modify the final export result. Try out which is the best selection for your needs.
- With 'Copy' secure the data export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the data as file.

3.5.5.7 Special - Application Screenshot

Export of the LogView Studio window as 'PNG' Graphic.

Menu Section

Special

Procedure

- Select 'Application Screenshot' in the Menu 'Export' (Section 'Special')
- With 'Copy' secure the graphic export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the graphic as file.

3.5.5.8 Special - Desktop Screenshot

Export of the complete desktop screen as 'PNG' Graphic.

Menu Section

Special

Procedure

- Select 'Desktop Screenshot' in the Menu 'Export' (Section 'Special')
- With 'Copy' secure the graphic export to the clipboard (and continue processing of your choice using the clipboard memory), or select 'Save' to save the graphic as file.

3.5.6 Analyse

This chapter describes the 'Analyse' menu and its functions.

Sections

This chapter contains following thematic sections:

- [Min Max Average](#)
- [Average Filtered](#)

3.5.6.1 Dataset - Min Max Average

Calculating the following values for a specific dataset:

- Start Date and Time
- End Date and Time
- Difference (time span)
- Minimum (per Sensor)
- Maximum (per Sensor)
- Average (per Sensor)

Menu Section

Dataset

Procedure

- Select a Dataset Element in the Project Explorer
- Select 'Min Max Average' in the Menu 'Analyse'
- Following display output will be created:

```
Analyse Min Max Avg
Minimum / Maximum / Average Analysis

Start Date : 21.08.2014 20:06:01
End Date   : 21.08.2014 20:26:47
Difference  : 00:20:45.8000000

Value Count : 6230

Höhe
Minimum : -0,033
Maximum : 203,384
Average : 96,289

Temperatur
Minimum : 22,0
Maximum : 24,0
Average : 23,232

Steig/Sinkrate
Minimum : -3,0
Maximum : 8,4
Average : -0,043
```

Note

The results are not added to the Project Explorer.

3.5.6.2 Dataset - Average Filtered

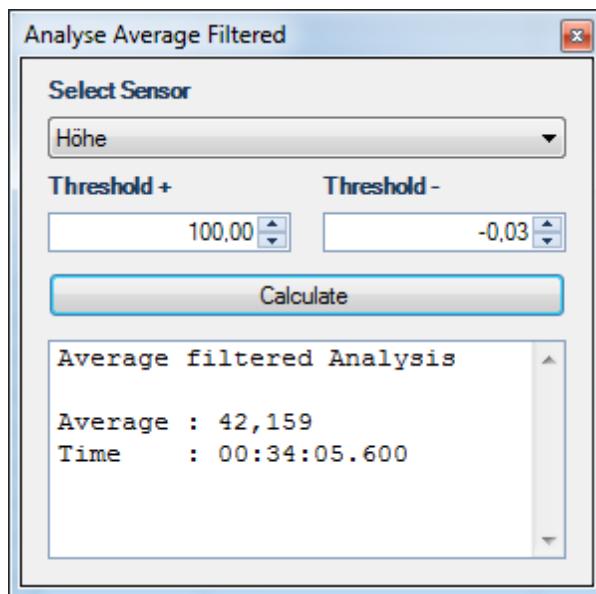
Calculating the average value for a specific sensor. It can be selected which values shall be included in the calculation (with Threshold).

Menu Section

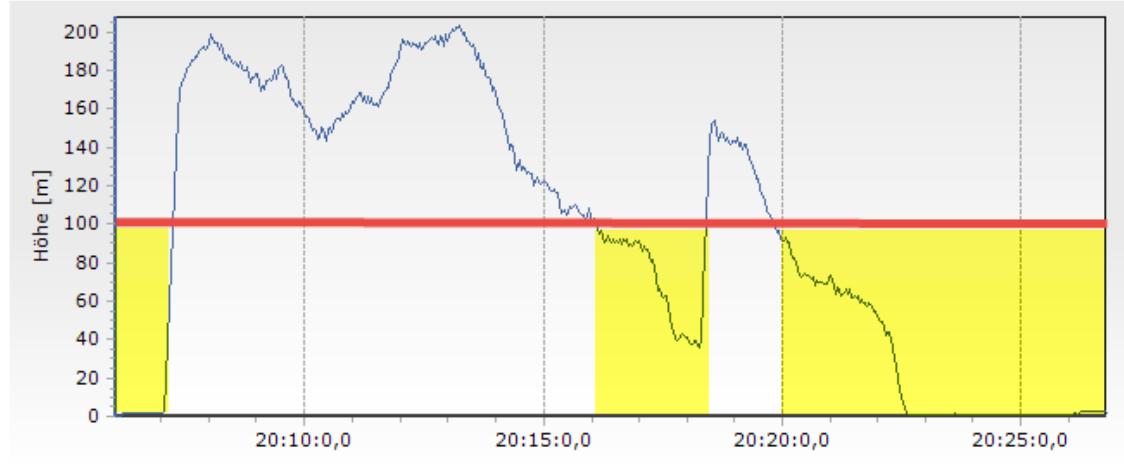
Dataset

Procedure

- Select a Dataset Element in the Project Explorer
- Im Menü Analyse auf 'Average Filtered' klicken
- The following dialog window will appear:



- By adjusting the Threshold +/- values the range of data which shall be included in the calculation can be specified. In the following sample all values between 100 and -0.03 will be considered (yellow marked range of the curve):



Note

The results are not added to the Project Explorer.

3.5.7 Tools

This chapter describes the 'Tools' menu and its functions.

Sections

This chapter contains following thematic sections:

- [Change Start Time](#)
- [Open Settings Folder](#)
- [Add Host to Database](#)

3.5.7.1 Change Start Time

Due to various reasons it may happen that the recorded dataset has not the right starting- / ending time. This may occur e.g. when the device did not give specific time information and LogView Studio used the system time as reference. Also devices with GPS information are using in most cases UTC (GMT) time information and not the local time.

In such case the tool 'Change Start Time' can be used to adjust the start time to the Users preferences.

This tool can be used for dataset elements as well as for chart elements.

Note

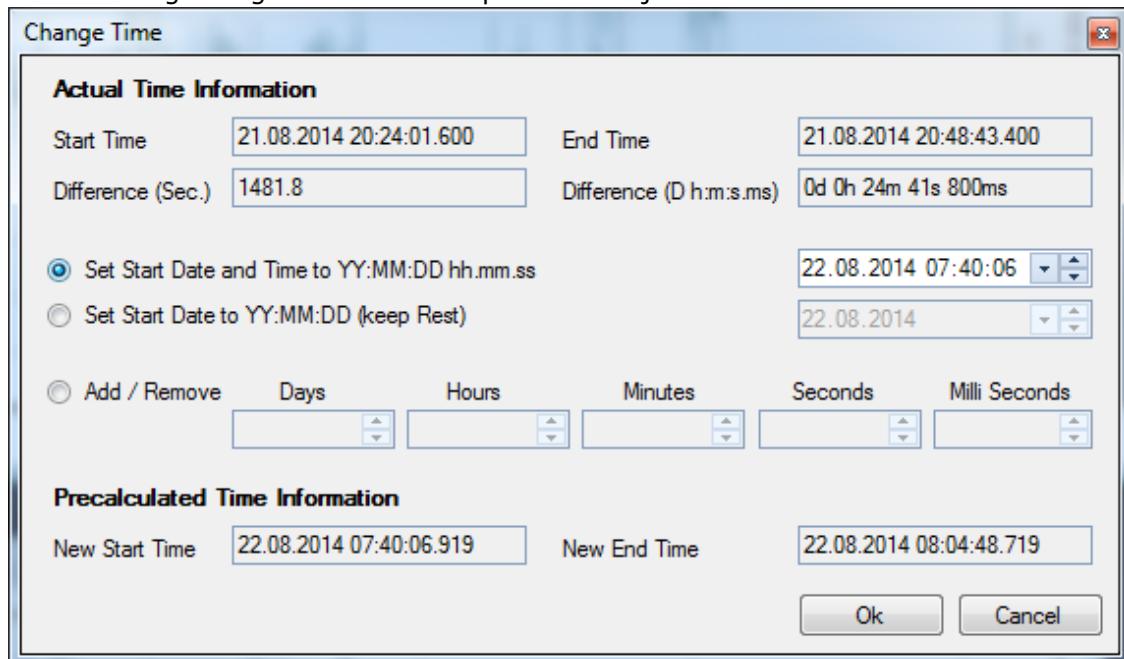
Using this tool with a dataset will modify the data within the database. So the original data information will be manipulated!

Menu Section

Data

Procedure

- Select a Dataset or a Chart Element in the Project Explorer
- Select 'Change Start Time' in the Menu 'Tools'
- The following dialog offers various options to adjust the start time:



The original time information is always shown in the top part of this dialog window. The adjusted or modified calculated time information is shown in the bottom area of the dialog window.

- Select 'OK' to confirm the new settings. Changing the time data in a dataset will be stored in the database. When changing the time in a Chart Element only the time axis will be adjusted.

3.5.7.2 Open Settings Folder

Some important data in LogView Studio must be saved in specific user folder. In this folder there will be stored information about the last used projects and also the error logs.

With the function 'Open Settings Folder' this folder can be opened and displayed directly without searching.

Menu Section

Special

3.5.7.3 Add Host to Database

The function 'Add Host to DB' is used to allow other computers the access to the current project.

The procedure is described in the [Network sample](#).

Menu Section

Database

3.5.8 View

This chapter describes the 'View' menu and its functions.

Sections

This chapter contains following thematic sections:

- [Tool Windows](#)
- [Windows](#)

3.5.8.1 Tool Windows

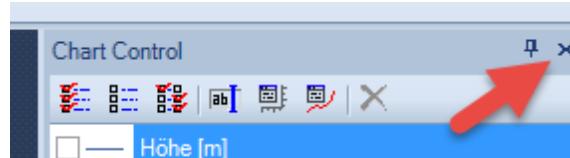
LogView Studio has some specific tool windows like the 'Project Explorer' or the 'Chart Control'. The buttons in this section of the 'View' Menu can be used to change the display status of the various tool windows.

Menu Section

Tool Windows

Procedure

- Select one of the available tool windows buttons in the section 'Tool Windows' to close or open the related tool window.
- Alternatively the tool windows can also be closed with their little 'X' symbol in the top right corner:



- The tool windows can also be minimized using the 'Pin'. The 'Pin' will change to horizontal position indicating the windows actual 'minimized' status. The windows

will be reduced to a tab at the screen border:



Moving the mouse over the tab will reopen the minimized window, selecting the 'Pin' again will change back the display status of the window ('Pin' displayed in vertical position).

- Further option in this section is the button 'Auto Hide All' to minimize all tool windows in one step.

3.5.8.2 Windows

The display status of the centered 'normal' Views as for example a chart or a gauge can be changed in LogView Studio using the standard window functions available in the 'Windows' section of the 'View' Menu.

The various functions in this section shall be self explanatory.

Menu Section

Windows

3.5.9 Help

The 'Help' Menu offers direct access or link to the main help topics of LogView Studio.

3.6 Working with LogView Studio

This Chapter describes the major and often used functions of LogView Studio.

Sections

This chapter contains following thematic sections:

- [Information about Devices](#)
- [Using the Device Monitor](#)
- [Using and adjusting Charts](#)
- [Samples](#)

3.6.1 Device

This Chapter describes the setup and configuring of devices in LogView Studio.

Sections

This chapter contains following thematic sections:

- [Start / Stop a Device](#)
- [Source Configuration](#)
- [Device with File Import](#)
- [Autodetect Serial- and HID Ports](#)
- [Zero Sensor Devices](#)

3.6.1.1 Device Start and Stop

Each Device in the Projekt Explorer has a Checkbox on the right side of the name:



Activating this Checkbox will start the Device. That means the incoming data of the related Source will be received, processed and stored into the database.

This process can be stopped by selecting the Checkbox again (removing the activating mark):



Note

Using a File Import or Binary Source the Checkbox is marked grey (deactivated). The process of reading the data will be started immediately after closing ('OK'-Button) the Source Configuration Dialog window.

Logging

Starting and Stopping is shown also inside the Logging protocol:

```
Mo 18:21:02.232 : [ 9, Serial      ] Open Port, Start IO Thread (Port : COM1)
Mo 18:21:02.251 : [ 9, Serial      ] GetSetting : COM1||\\?\acpi#pnp0501#1#\{..\}||9600||8||One||None||False||False
Mo 18:21:53.067 : [ 9, Device     ] Stopping Source
Mo 18:21:53.068 : [ 9, Serial      ] Stop IO Thread, Close Port
Mo 18:21:53.068 : [ 9, Device     ] Stopping Calc Thread
Mo 18:21:53.068 : [ 9, Device     ] Closing Database
Mo 18:21:53.081 : [37, Serial      ] Serial IO Thread stopped.
```

In case of problems the possible reason of the problem can be here.

Further Information

[What is a Device?](#)

[What is a Source?](#)

[Add a Device](#)

[Samples](#)

3.6.1.2 Change Source

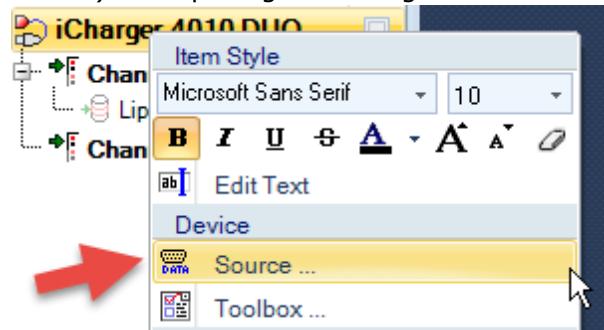
Each device has (minimum) one source in LogView Studio. The source is the port where the data comes to the computer (so for example a serial port).

Normally the source will be specified and used directly when [adding a device](#). Alternatively a source can also be specified or changed later.

Change Source

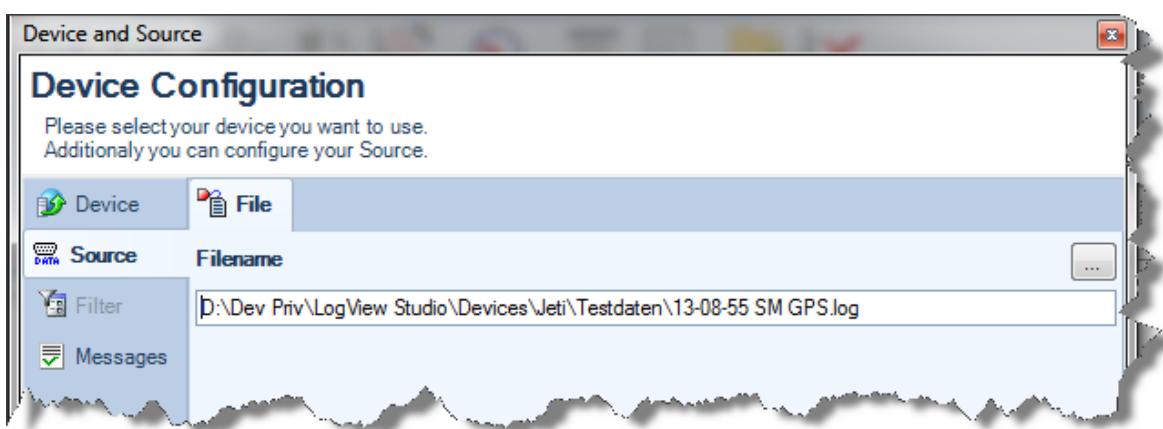
There are different ways to reach the source configuration of a specific device:

- by double click on the relevant device name in the Project Explorer.
- by selecting 'Source ...' inside the device context menu (right mouse click on the device) and opening the dialog window 'Device Configuration'.



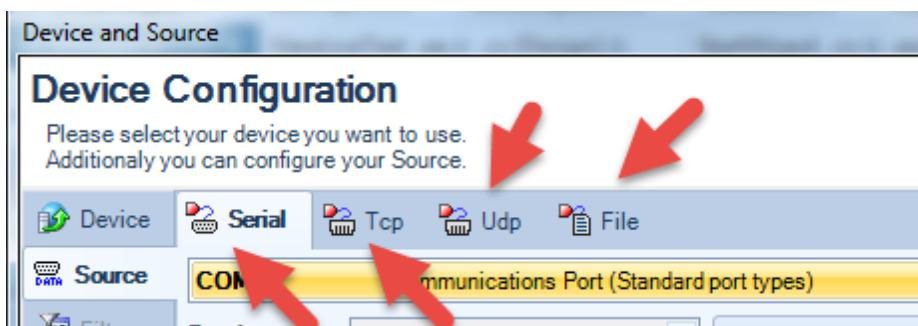
- using the Command Line Tool

The 'Device Configuration' dialog window will appear on the screen (except when using the Command Line Tool).



Devices with more than one source

A device may have (more than one) different sources, e.g. the OpenFormat can read data from various different sources. In this case the correct source must be selected and configured by selecting the related tab.



After configuring the device and source the setup dialog must be confirmed and closed

with the 'OK' button.

Note

Only one source of each device can be active (used) at the same time.

Further Information

[What is a Device?](#)

[What is a Source?](#)

[LogView Studio Command Line Tool](#)

3.6.1.3 Fileimport

Compared with other sources the [File and Binary Sources](#) are a special type of sources. When using a file source all the data are available immediately. Therefore LogView Studio start reading and processing the datafile direct after the user has confirmed the Device Dialog window with the 'OK' button.

The option to start or stop the device is not possible with File / Binary Sources (greyed out checkbox).

Note

When the 'Device Configuration' Dialog is re-opened and confirmed again with 'OK' the file will be read and processed again!

A multiple import for more than one file is actually not (yet) available.

Import of more than one Logfile into a Project

Especially when using a device with File or Binary Source the user can be confused how to import different (more than one) logfiles. The right process would be the following:

- [Add a new device](#) to the Project (once).
- Specify the first logfile immediately when configuring the File/Binary Source.
- Confirm the dialog with 'OK' and the first Logfile will be imported.
- After that double click on the device in order to open the Source Dialog again (alternatively also possible with a right click on the device using the context menu by selecting "Source ...").
- Then specify or select the next logfile and confirm the dialog with 'OK' (it is NOT necessary to add a new device)
- Repeat the last two steps until all logfile have been importet.

3.6.1.4 Autodetect Serial / HID

Both the Serial and the HID Sources offers the option of an Autodetect of the port to be used.

The Autodetect function is only working for ports which are connected to the Computer later, means primary everything which works via USB port.

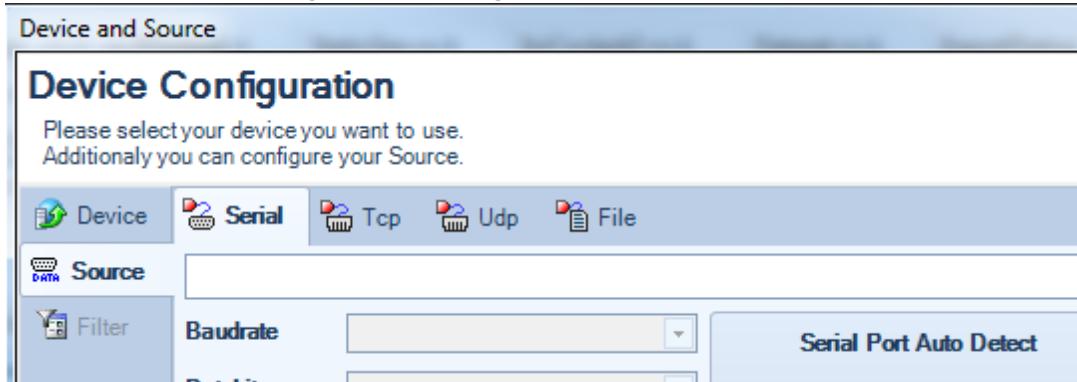
Note

Autodetect is not designed to detect or identify a device. It is only to identify the correct com-port for the source.

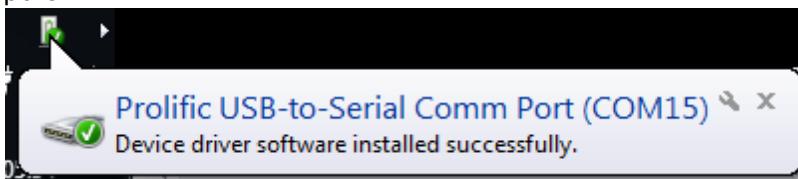
Procedure

The following sample shows the Autodetect with a USB / Seriell Adapter. The process with a USB HID Source is the same.

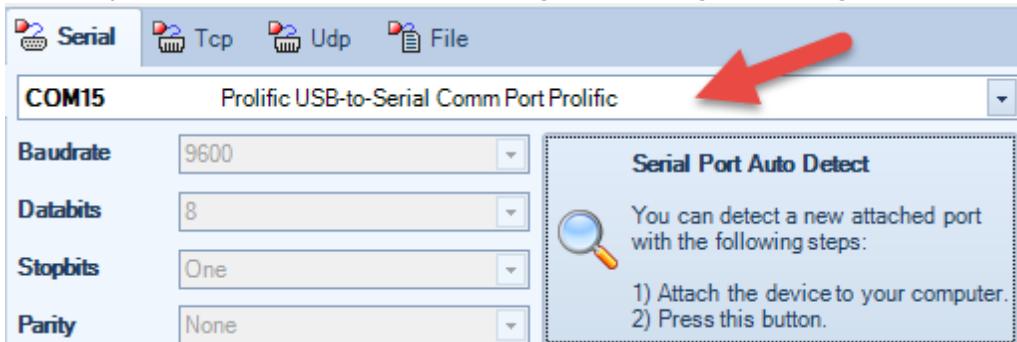
- The device should NOT be connected (yet) to the computer!
- Start the 'Device Configuration' Dialog as described before



- Connect the device to the Computer and wait until Windows has detected the unit. It may happen that this takes a while when first a driver has to be installed (by windows automatically). Please observe the messages on the screen. As mentioned earlier this detection is only working if the device is connected via USB port.



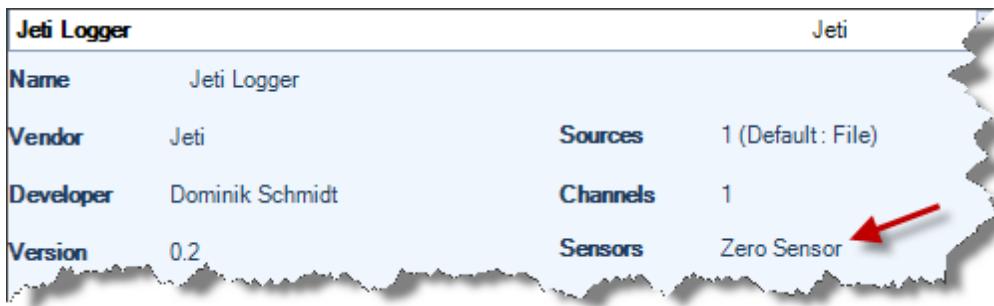
- Then select the button 'Serial Port Auto Detect'. With HID select the related button 'USB HID Port Auto Detect'.
- The new Port should be selected automatically. Please observe and be sure the correct port is selected before confirming and closing the dialog with 'OK'.



3.6.1.5 Zero Sensor Devices

LogView Studio may have devices available which do not have preconfigured sensors (= processable measured values). Those devices can be correlated to their sensors with the runtime e.g. according to the length of the processable logfile and the database can be extended accordingly.

Such devices can be recognized within the 'Device Configuration' dialog:



Normally there will be no problems with such devices. The data will be read, the database will be adjusted, and the data will be processed / displayed accordingly.

Problem : Different logfiles

Sometimes it can happen that the results are looking slightly curious in case of importing data which have used different sensors.

Sample

A User has a logfile A with current and voltage data and another logfile B with speed.

He add and configures the proper device and import the logfile A. The database will be expanded with the two sensors current and voltage for the selected device. Now he create a chart for these imported data and he see only the two curves for voltage and current.

Now the user imports with the same device the logfile B. The device will now be extended with an additional sensor 'speed'. This is generally not a problem yet and is working fine.

But when the user now creates a chart for the second dataset (from logfile B) then this chart has three (3) curves: current, voltage and speed but the first two curve have no data in.

Recommended Procedure

In case the sensors (measured data) of the imported (or to be imported) datafiles are different it is recommended to add the same device in the project again for a second time and import or read the data with that new device. Then only the necessary and reasonable sensors will be created.

3.6.2 Device Monitor

This Chapter describes the use of the 'Device Monitor' with the help of a sample.

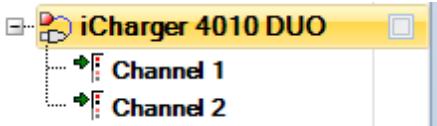
Note

The Device Monitor is not necessary under normal working conditions of LogView Studio. This monitor becomes interesting for data analysis which are sent from the device to the Computer or vice versa. The device monitor is primarily useful for developers.

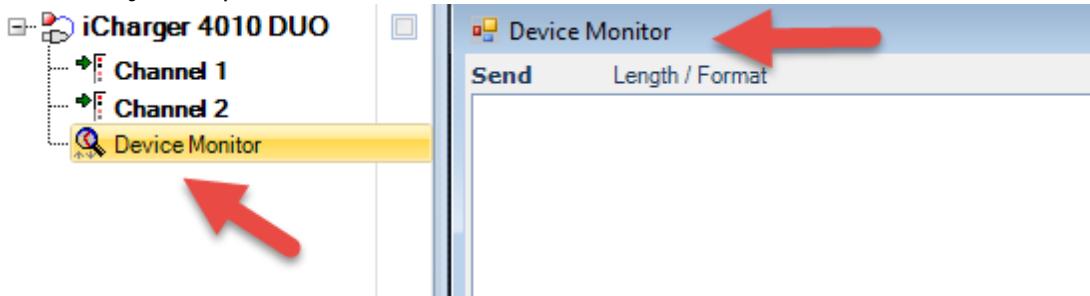
Procedure

The following sample illustrates the handling of the Device Monitor using a Junsil battery charger 4010 DUO with HID Interface.

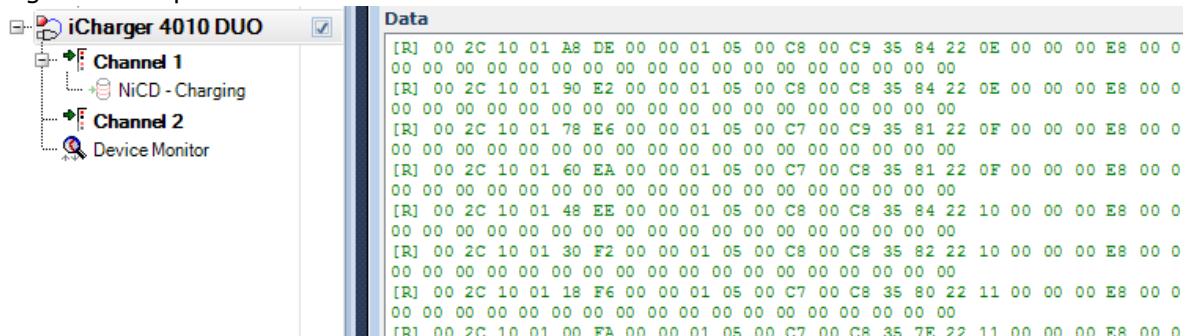
- The device which shall be observed must be available in the Project:



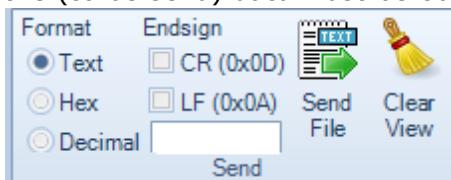
- Select the device in the Project Explorer and then select 'Device Monitor' in the Menu 'Project'. A new window will appear and a new element 'Device Monitor' will be added to the Project Explorer tree:



- Now start the device (with the checkbox). As soon as data have been received the log will be expanded:



The Device Monitor can also be used to send data to the device. Doing so the format the (to be send) data must be configured in the menu:



In this case shall be a text in the 'Send' frame. This will be sent to the device.

No Data Calculation

The menu 'Device Monitor' has a special option 'No Data Calculation'. When this option is activated LogView Studio will not calculate or store the data. The data will only be displayed in the Device Monitor and will be neglected after that.

This function is very helpful when doing tests with the device and the analysis or processing function of the data is only playing a minor role.

3.6.3 Chart

This chapter describes more detailed the use of Charts

Sections

This chapter contains following thematic sections:

- [Zoom und Pan](#)
- [Graphic layout](#)
- [Templates](#)
- [Measure](#)

3.6.3.1 Zoom und Pan

List of the Zoom- & Panning functions of a graphic (chart):

Zoom

left mousekey -> rectangle (from top/left to bottom/right)	Zoom rectangle
left mousekey -> rectangle (from bottom/right to top/left)	UNDO Zoom
left mousekey + STRG	only X Zoom
left mousekey + SHIFT	only Y Zoom
mousewheel rotate up	Zoom OUT at mousecursor
mousewheel rotate down	Zoom IN at mousecursor
mousewheel rotate + STRG	Zoom x
mousewheel rotate + SHIFT	Zoom y
mousewheel rotate + SHIFT	Zoom from center

Pan

pressed mousewheel	X / Y panning
pressed mousewheel + STRG	X panning
pressed mousewheel + SHIFT	Y panning

Zoom & Pan

pressed mousewheel + mousewheel rotate	X / Y panning & Zoom at Mouse
pressed mousewheel + mousewheel rotate + STRG	X panning & X Zoom at Mouse
pressed mousewheel + mousewheel rotate + SHIFT	Y panning & Y Zoom at Mouse

3.6.3.2 Layout

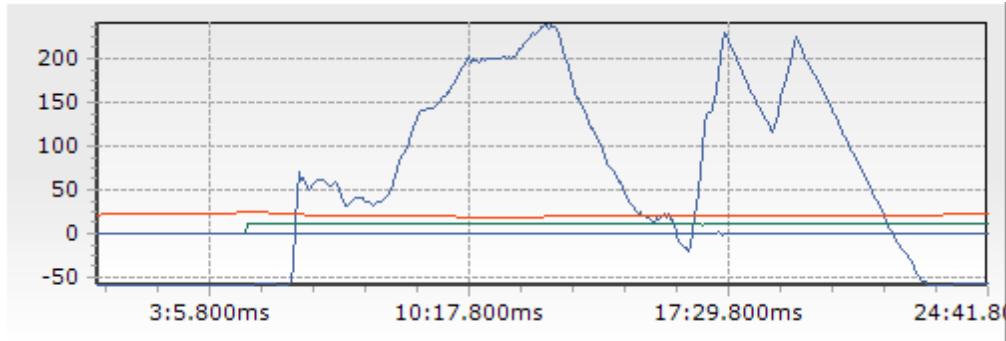
The charts in LogView Studio can be configured graphically in various different ways.

Axis Style

For the layout of the charts there are three different basic formats available:

- Left only

There is only one axis for all curves.



- Stack

Each curve has her own axis on the left side of the chart. All curves are displayed in stacked mode.



- LV Style

Each curve has her own axis displayed either on the left or right side of the chart.



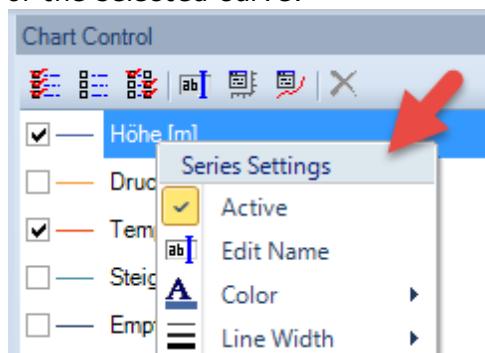
The selection of the axis style can be selected in the 'Axis Styling' section of the menu 'Chart' -> 'Settings':

**Note**

The 'compressed' versions combines similar curves on one axis. By using this for example the single cell voltages of a rechargeable battery can be combined on one Axis.

Context Menu

Each curve of a chart has her own entry in the 'Chart Control' window. By using the context menu (rightclick on the curves entry in the 'Chart Control') the graphical layout of the selected curve:



The menu offers multiple various options to format the graphical appearance of the selected curve. Additionally also the Axis Settings can be adjusted there (Endvalues, Offsets, ...).

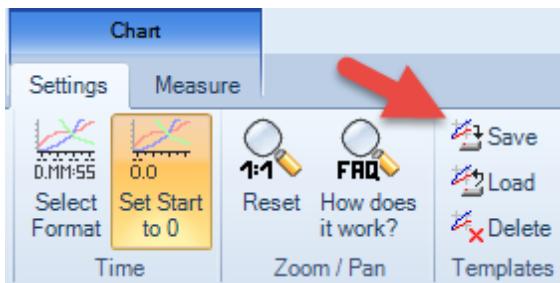
3.6.3.3 Templates

The layout and styling of charts in LogView Studio can be stored in templates. These templates can be used later for easy formatting similar charts with the same style.

In the menu 'Chart' -> 'Settings' there is the section 'Templates' with the functions 'Load', 'Save' and 'Delete'. These functions are linked to the actual active chart.

Save Template

To save a template first select the chart of your choice. Then select the 'Save' button in the 'Templates' section.



A new dialog will open to enter the (file-)name of the template. After confirming with 'OK' the template will be saved.

Note

The templates will be saved in the LogView Studio User folder. Under Windows 7 this is for example

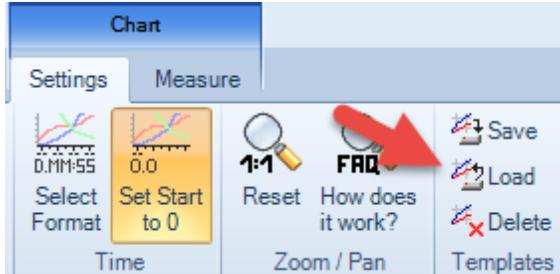
`C:\Users\<USERNAME>\AppData\Local\LogView Studio\ChartTemplate`

Note

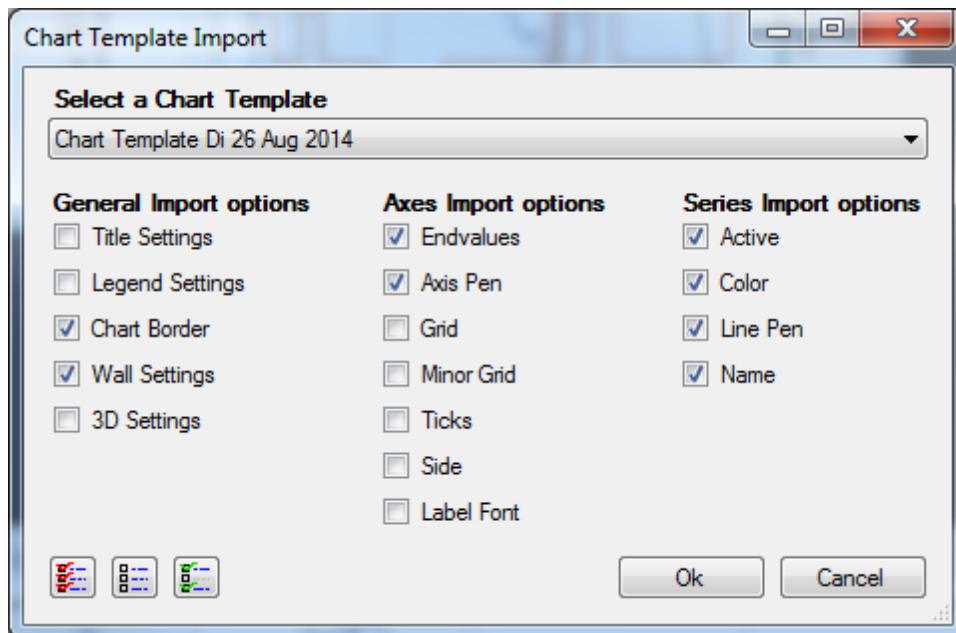
During Deinstallation of LogView Studio the templates will not be removed. The User folder will remain on the computer and must be deleted manually. See also 'Remove User Data'.

Load Template

Again first select a chart of your choice. Then select 'Load' in the 'Templates' section of the 'Chart' -> 'Settings' menu:



The following dialog window will open:



In the top area an existing (saved) template can be selected. In the bottom area various settings of the saved template which shall be applied to the active chart can be pre-selected.

The Default selection is in most cases fully sufficient.

Templates in different charts

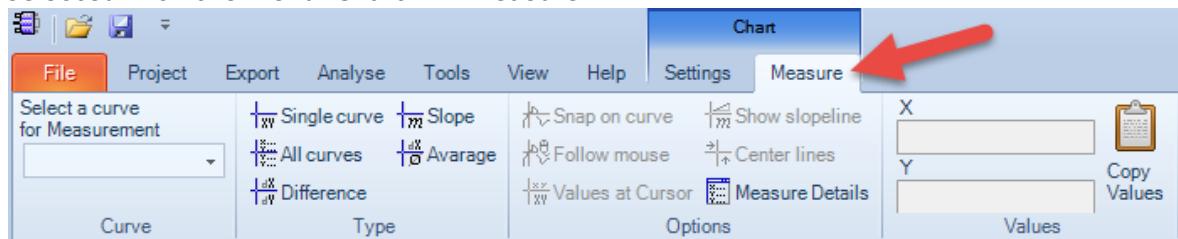
In general it is recommended to load templates only for similar charts - e.g. for charging curves of a specific battery charger. In case the number of curves are different due to (for example) a different type of charger the template will work anyway. But some curves will then remain in the actual / previous format or some information from the template will be simply ignored. This depends whether the active chart or the template has more curves (or curve information).

Additional Information

[Remove User Data](#)

3.6.3.4 Measurement

LogView Studio offers a lot of different measuring functions for charts. These can be selected with the Menu 'Chart' -> 'Measure':

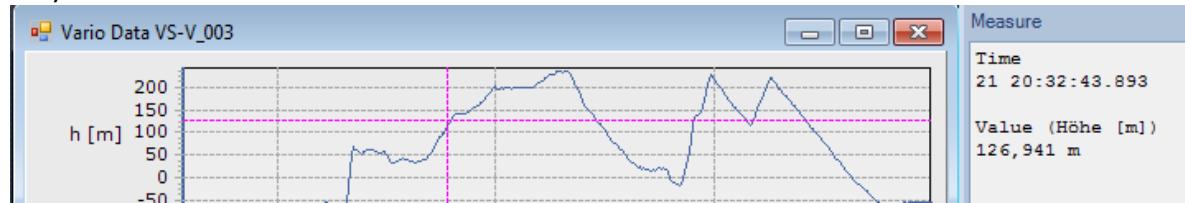


To start or stop the measure function it is just needed to select a curve and the type of measuring.

Following options are available:

- **Single Curve**

Only one curve will be measured.



- **All Curves**

All displayed curves will be measured. There is only one vertical measuring mark which can be moved with the mouse.



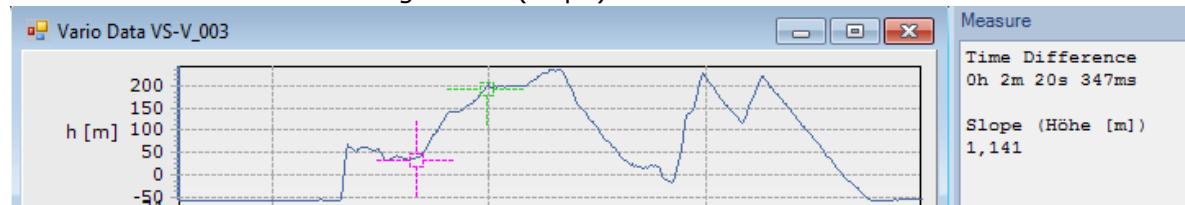
- **Difference**

With two cross hairs a differential measure can be done.



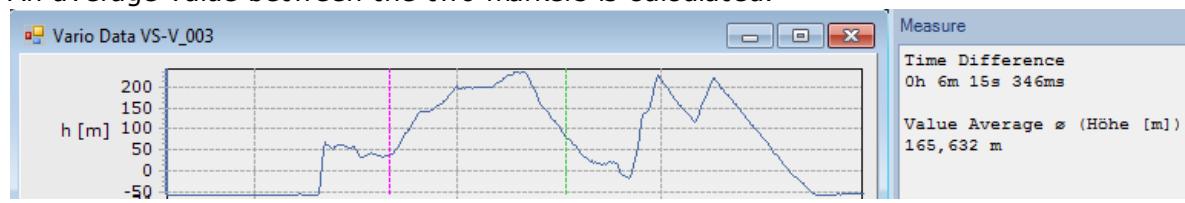
- **Slope**

With the two cross hairs the gradient (slope) of the curve can be calculated.



- **Average**

An average value between the two markers is calculated.



With 'Copy Values' the measured values can be copied to the clipboard anytime.

3.7 Samples

This chapter explains the LogView Studio usage with some examples.

Sections

This chapter is divided into different sections:

- [Simple Project](#)
- [Network Usage](#)

3.7.1 Simple Project

The following case study shall try to describe the use of LogView Studio a little bit more pragmatic.

The situation

A user has an electric model airplane, two batteries, a charger and a data logger (= telemetry). Now he would now like to collect data about the flights but also get and collect information about his batteries.

Projecting

The first step now is to think about how to organize the one (or even different) [project\(s\)](#) for this purpose.

Note

The organization of a project remains the task of the User and his preferences. LogView Studio offers with the project structure just a flexible framework to combine interrelated data into one project.

The following basic ideas are possible:

- **One Project "Airplane"**

The project needs two Devices "Charger" and "Data logger" which have to be added. All data will be collected with these two devices. The collected data are then sorted in folder structures.

- **Two Projects "Charger" and "Data logger"**

One Project for the Charger and one project for the data logger will be created and the data will be collected and stored in different projects. Then there is no direct relation to be made possible between the data of the telemetry and the batteries, but in case this is not relevant for the User this is a possible solution.

- **Two Projects, one for each battery**

In this scenario each project "Battery 1" and "Battery 2" will get a device "Charger" und "Data logger". But the collection and storing of the data will be done for each battery separately.

You see already in this rather simple example that there are various options in LogView Studio to setup and arrange your projects. There are not any rules or restrictions given to the user. A project consists finally of such elements which seems to make the most sense to the User.

We will continue the description only for the first option "Airplane" now.

Start LogView Studio and create the project

To start LogView Studio is the most easy step once you have read the <Quickstart> chapter of this Help file. Most simple option is to start 'LogView Studio' from the Windows Start Menu. Alternatively you can start the 'LvGui.exe' using a File Explorer.

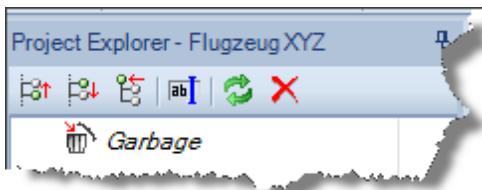
Next step is to create the new project via File -> New -> Project (alternatively in the LogView Studio Wizard click on 'New Project')



Enter a project title in the LogView Studio Wizard and select a suitable folder of the hard drive's Filesystem (available empty folders are marked with a green dot).

The project name should reflect the content of our planned project. All data of the project including the database will be stored in this folder. (It must be empty before a project can be created in it.)

After click on 'Finish' the creation process of the project will be started by the Wizard and the progress can be observed. Various messages will appear, the database will be started and various template files will be copied into the project folder until finally the Wizard window will close itself. When the project has been created successfully the headline in the Project Explorer will change and display the Project name. Furthermore the first and only entry in the Explorer tree shall be the 'Garbage' trash can:



Add Devices to the Project

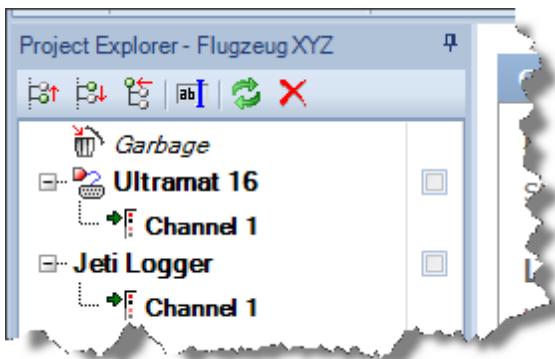
Select 'Device' in the 'Project'-Menu and add the Charger as device. Also select and configure the related Source in the same step. In our sample the device is a 'Graupner Ultramat 16' and a serial port COM22 as source configuration. (The User must choose his own configuration.) Confirm the dialog with 'OK'.

Now add the data logger (or telemetry analysis) in the same way. Click on 'Device' and select the 'Jeti Logger' as device to be added. The source leave open at this point. Confirm with 'OK' and the devices have been added.

Note

The devices in this sample project are just examples. If you want to replicate this sample project you should select your own available devices and ports.

The screen should now look like the following sample screen:



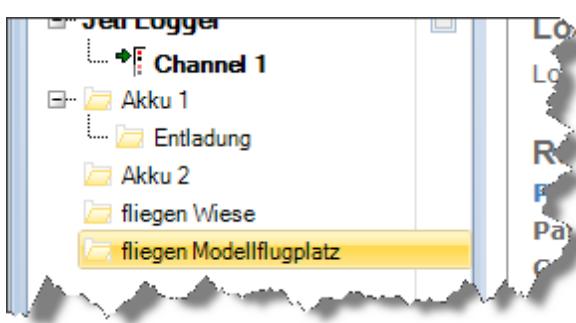
Add Folder to the Project Tree structure

Actually we have now a project with two devices added. In order to keep a good overview with the expected datasets we will now add some additional folder. They will be used later.

Select in the 'Project' Menu the icon 'Folder' in the 'Organize' section. In the next window enter the folder name - e.g. 'Akku 1' - and confirm with 'OK'.

If an existing Folder is highlighted then the selection of the 'Folder' icon will create a sub folder to the highlighted folder. Also possible is to use the context-menu of the Project Explorer (right mouse-click).

After adding some folder the structure of the Project should look similar as to the following sample screen:

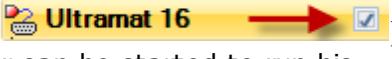


Now the basics of the Project is already configured to our (Project-)requirements and we can start receiving the data ...

The Charger

Our virtual User now start to prepare his rechargeable batteries with the battery Charger. First step is to discharge 'Akku 1'. After that the same battery will be fully charged for the next flight.

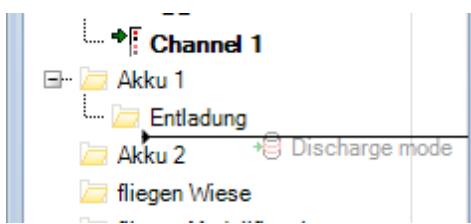
The battery is connected to the charger and the charger is connected to the Computer.

The Charger entry in LogView Studio is activated ( in order to receive the related data. Then the Charger can be started to run his discharging process once and then the charging process of the battery. Our 'Ultramat 16' device should now look similar to the following Screenshot sample:



Now the User like to sort all datasets of 'Akku 1' in the folder 'Akku 1'. As the discharging process is normally not used very often our User like to store this in a sub folder 'Entladung' (Discharging).

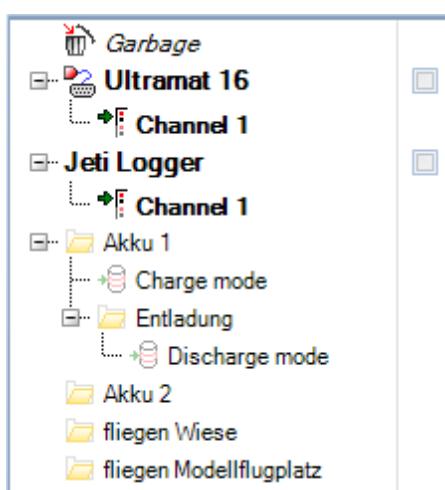
This can be done with the Drag-and-Drop function: click and hold down the left mouse key on the dataset 'Discharge Mode' and then move the dataset to the folder below the folder 'Entladung'. A black line with a little arrow indicate the position.



Now release the mouse key and the dataset has been moved to that folder - similar as it works in Windows Explorer.

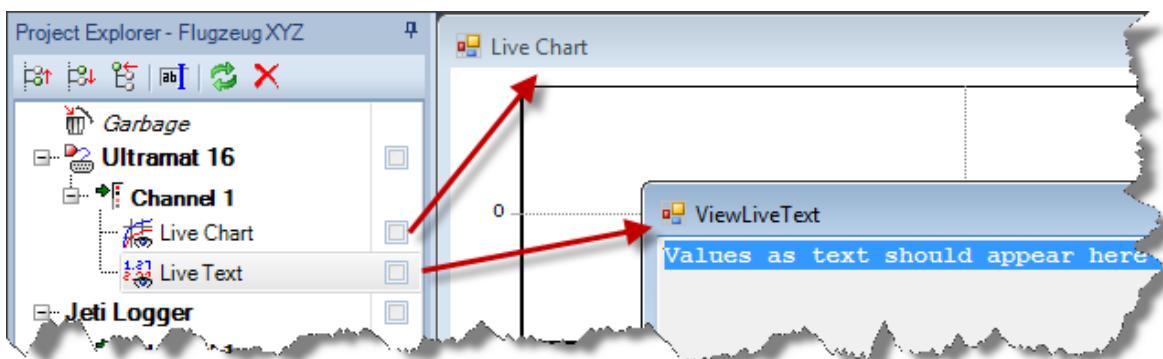
The same procedure will be done with the dataset 'Charge Mode' - but this dataset should be moved to the folder 'Akku 1'.

The project structure should now look like following screen sample:



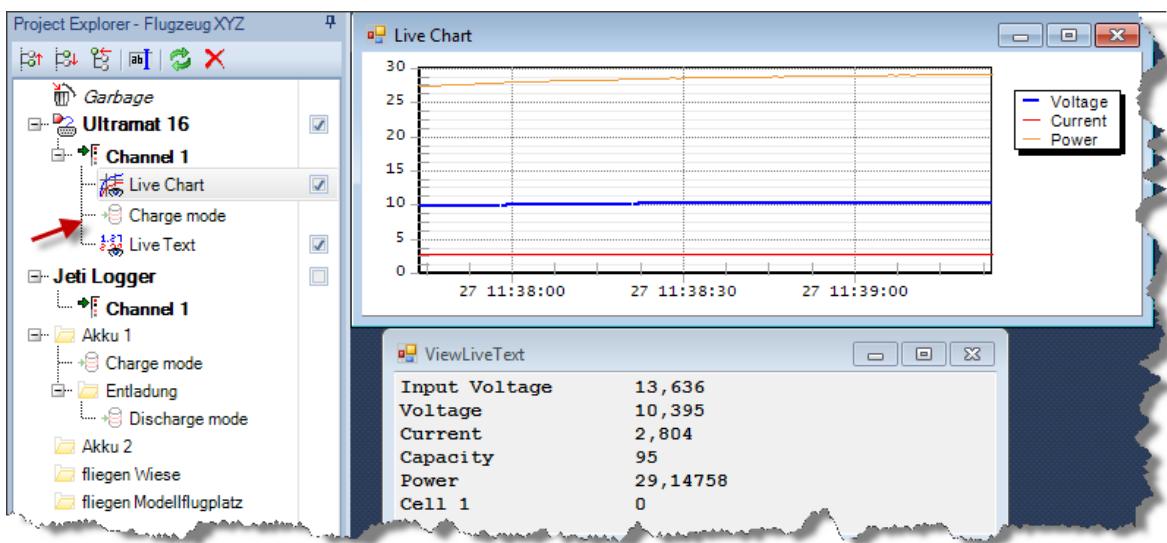
In the next step our User will load the second battery with the same Charger. The dataset will be received in LogView Studio again via Channel 1 of the 'Ultramat 16'.

But this time the User like to see the actual graphic chart and the last data of the second battery charge process. For this purpose he add a 'Live Chart' and a 'Live Text' to the Channel 1 of the Ultramat 16 Charger:



As soon as these 'Live Views' are activated they show the last data from the database and will update their display periodically.

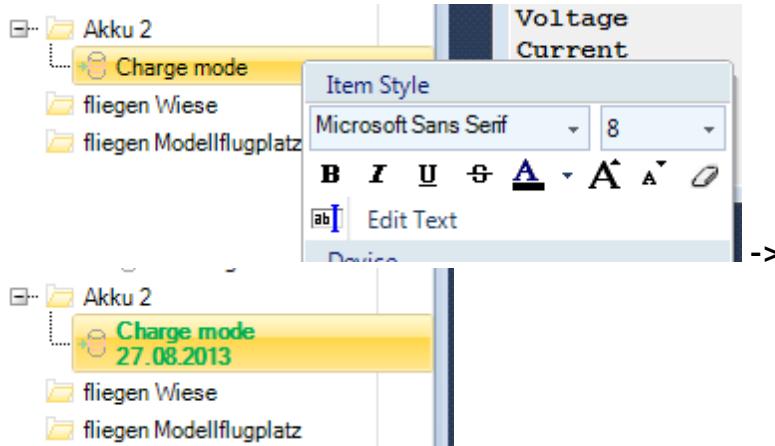
After starting the Charging process, activating the device and the two Live Views with their checkboxes, the charging process of 'Akku 2' in LogView Studio may look like the following Screenshot sample:



The 'Live Chart' shows the charging process like an oscilloscope with a diagram and the 'Live Text' shows the latest actual values in text form.

Additionally a new dataset was created automatically (after receiving the first data from the Charger) called "Charge mode".

When the charging process was finished, our User moves the dataset to folder 'Akku 2'. Using the context-menu (right mouse click) the name and style can be adjusted to the personal preferences (green text, bold, adding a date to the text):



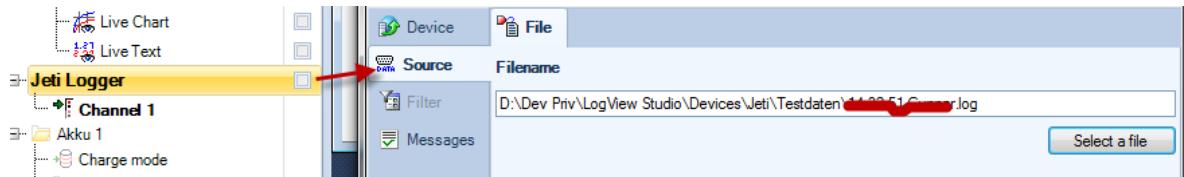
Preliminary result

The User has used his batteries and moved (and stored) the results in the folder tree. Furthermore he observed the data flow from the charger in two different Live View screens.

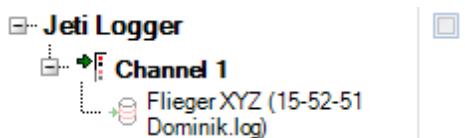
The User flew the model - flight data (telemetry) are available

Now the advantages of LogView studio become obvious when using different devices. The user can import various data into the same project which are somehow related to the batteries and also to the charge procedures but coming from complete different devices.

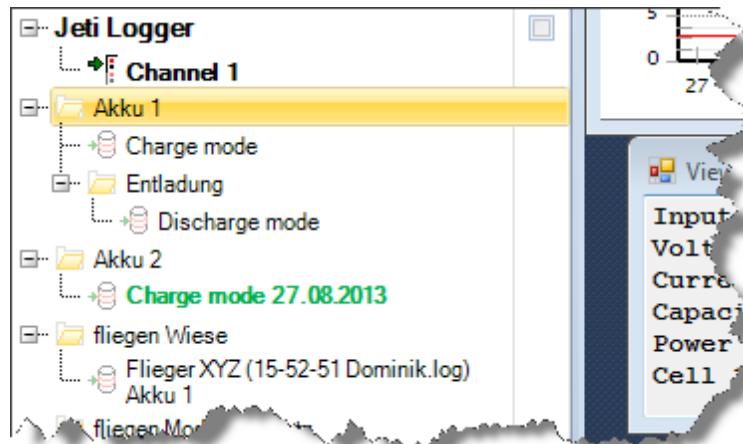
The User has flown the model on his lawn ('fliegen Wiese'), used the 'Akku 1' and produced a related telemetry data Logfile. The Logfile can be imported by double click on 'Jeti Logger'.



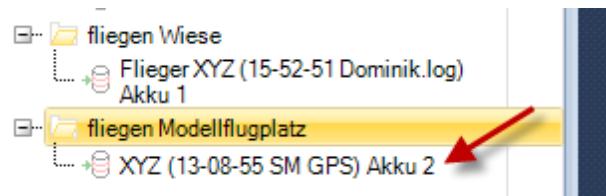
In the dialog window 'Device and Source' the User select as the source (in this case it is a file) the telemetry Logfile and confirms with OK. The Logfile will be imported immediately and will be added to the project as separate dataset under Channel 1 of the 'Jeti Logger':



To keep the project in good order this dataset was moved to the correct position in the folder tree as well and the title was modified adding the information of the used battery 'Akku 1':



After that the User went to the model airfield and flow down the second battery 'Akku 2' and also took logging data of the telemetry. Again he can double click on the 'Jeti Logger' and does not need to add a new device. In the device dialog the User selected as Source the new file and confirms with OK. The new Logfile will be imported similar as the first Logfile. Also this dataset can be moved to the folder model airfield ('fliegen Modellflugplatz') and the info 'Akku 2' was added to the title:

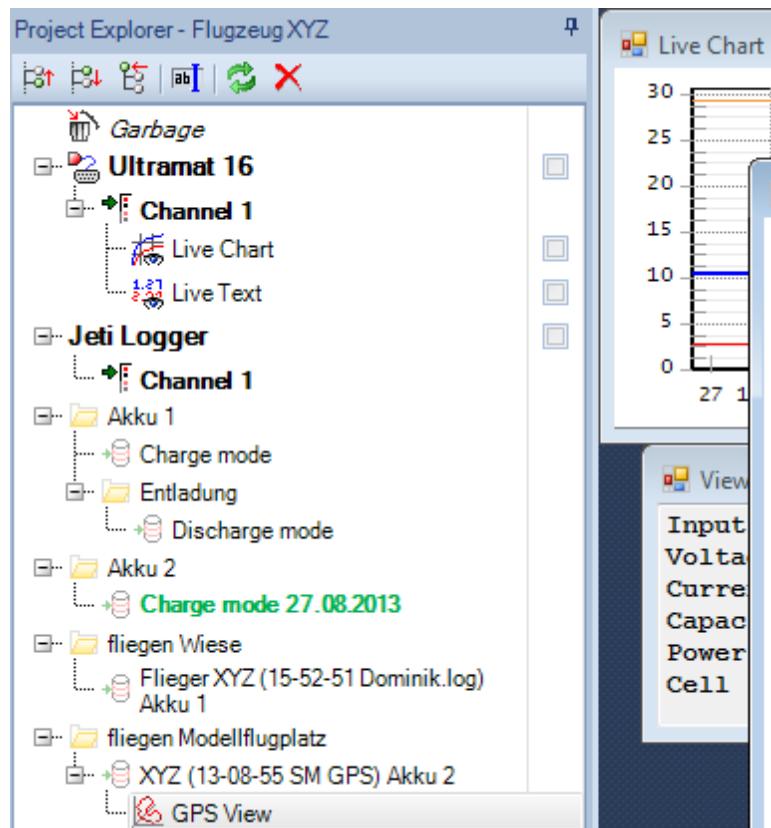


Analysis with (static) Views

Finally the User like to see additionally a static view of the last telemetry GPS data. He

select the related dataset (highlight) and then select 'GPS 3D' in the Views section of the 'Project' menu.

Now the project should look like following sample Screenshot:



All following battery charging processes or telemetry data files can be handled similar and added to this project as well.

Summary

The above theoretic example gives hopefully a better insight view to the options of LogView Studio and how it could be (or should be) used. The advantage are very clear located in the possibility to handle more than one device simultaneously (or in row) and to structure the collected data in a common project - fully flexible according to the Users preferences.

In case you need further informations you are kindly invited to connect to our Forum under www.logview.info where also questions can be placed.

3.7.2 Network Usage

The following sample describes the use of LogView Studio in a network setup. We use two different Windows Computer to show how it works.

Requirements

- You should be familiar with the 'normal' use of LogView Studio.
- The Introduction '[LogView Studio and Network](#)' should have been read and understood.

- The computer must see each other within the network.
- LogView Studio must be installed on both computers.
- It may be necessary under some circumstances to open the TCP Port 2804 in the Firewall to allow communication with the database.

Setting

For the following sample we use two different Windows systems (Windows 7 64Bit and Windows XP 32 Bit). Both computers are connected with the same network and LogView Studio is installed on both PC's.

Computer 1 - Windows XP

- LogView Studio has been started
- A Project has been created and one device has been added (in this case a 'Jansi 4010 DUO' Charger)
- The device is connected to the computer, ready for operation and can deliver data to LogView Studio.

Computer 2 - Windows 7

- LogView Studio has been started
- No project has been (yet) created

Procedure

Computer 2 shall get access to the project of computer 1 and display the data (on computer 2).

- First identify the IP address of computer 2. The most easy way to do this under windows is by using the command function:
 - Press 'WINDOWSkey' + 'R'
 - Enter '[cmd](#)' and confirm with the 'Return'-key
 - The console window will open. Enter '[ipconfig](#)' and press 'Return'
 - You will see the list of available network adapters and their details. Look for the actual connection (LAN or WLAN) and write down the mentioned IP address.A good help in this respect maybe also to read the following:
http://www.tippscout.de/windows-xp-ip-adresse-des-computers-ermitteln-und-herausfinden_tipp_2676.html
<http://www.wikihow.com/Find-the-IP-Address-of-Your-PC>

```
C:\> C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\>ipconfig

Windows IP Configuration

Wireless LAN adapter Wireless Network Connection:

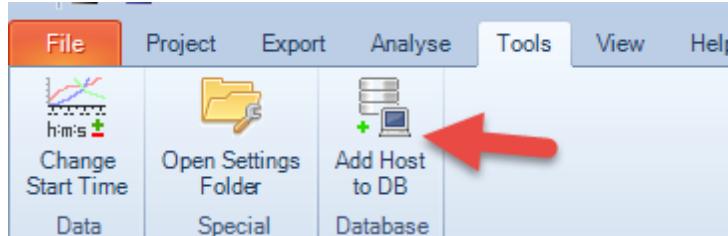
  Connection-specific DNS Suffix . : fritz.box
  Link-local IPv6 Address . . . . . fe80::3845:8728:751c:fbe5%15
  IPv4 Address . . . . . 192.168.30.140
  Subnet Mask . . . . . 255.255.255.0
  Default Gateway . . . . . 192.168.30.1

Ethernet adapter LAN:

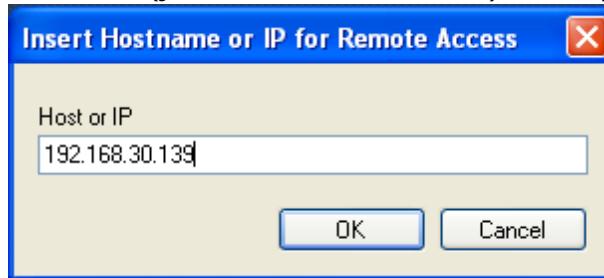
  Connection-specific DNS Suffix . : fritz.box
  Link-local IPv6 Address . . . . . fe80::5c60:858d:889c:7604%11
  IPv4 Address . . . . . 192.168.30.139
  Subnet Mask . . . . . 255.255.255.0
  Default Gateway . . . . . 192.168.30.1
```

In this sample the IP address (of computer 2) is 192.168.30.139.

- Next step is to open computer 1 for an access of computer 2. There is a separate button in LogView Studio's 'Tools' menu called 'Add Host to DB':

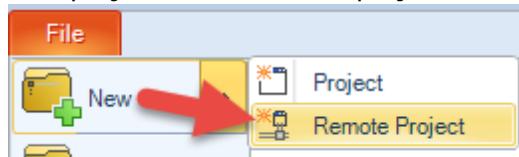


A new dialog window will open where the IP Address (or Hostname) of the remote computer which shall get access to the existing project must be entered. In our case this is the (just identified IP address) of computer 2 (Windows 7).



Confirm and close the window by selecting the 'OK' button.

- Now you can start on computer 2 (the LogView Studio Program is already running) a new project but a remote project. Select 'Remote Project' in the 'File' menu:



- In the following dialog you should Create a new Project Folder. This folder is necessary locally to store data (like e.g. charts).
- Select 'Next' to go to the next page of the dialog. Here you have to enter the IP

address (or Hostname) of that computer where the active Project is running (in our case computer 1 - XP).

Project Settings

Note

You can only connect to a running LogView Studio Project. It is r

Hostname / IP-Adress

192.168.30.141

- Then you can check the network connection:

Check the remote Connection

```
Try connection to remote LogView Studio project
(SERVER=192.168.30.141;PORT=2804;USERID=root;PASSWORD=;PROTOCOL=tcp;POOLING=false;Conn
ection Timeout=20;)
```

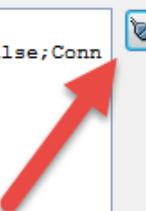
Connecting ...

Connection Successful

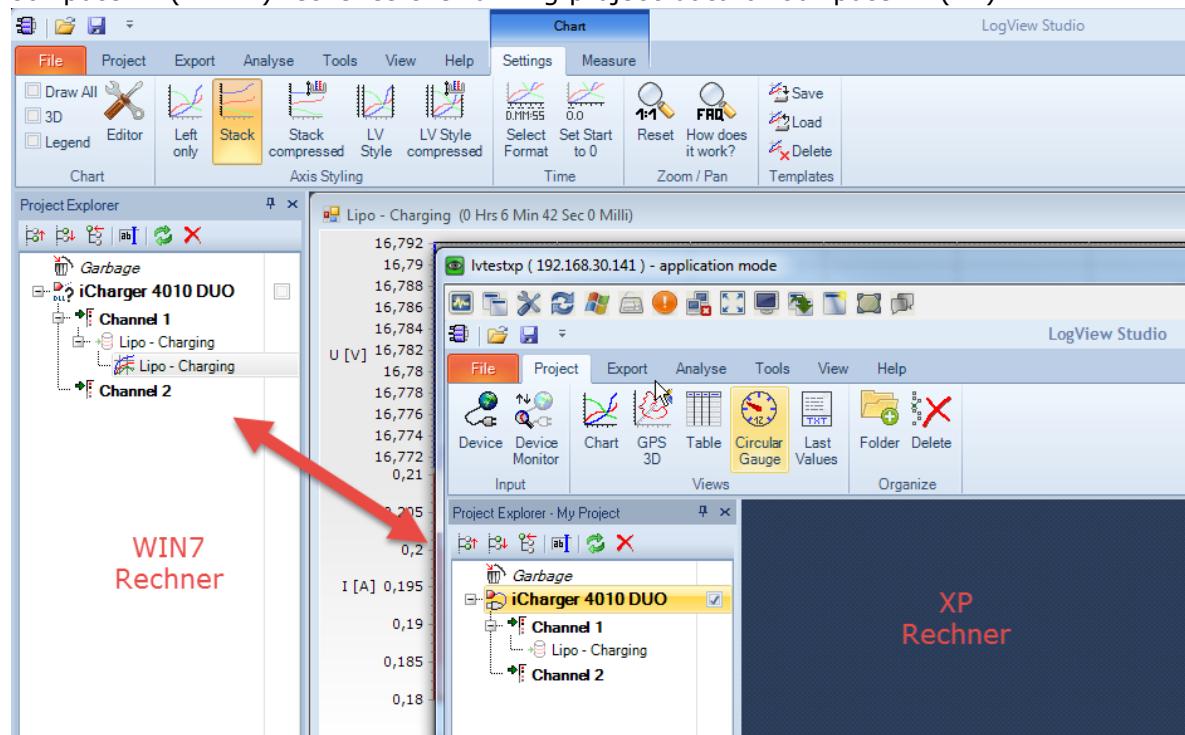
Checking if there is a LogView Studio database

Database found

Perfect - You can connect to that Project



- If everything is OK, close the dialog by selecting the 'Finish' button.
- Final status result should be now a running LogView Studio on both computers where computer 2 (WIN 7) retrieves the running project data of computer 1 (XP):



Note

You can connect to one project as many (remote-)computer as you like.

Note

Also with LogView Studio Command you can establish a network connection to such a project!

Accessibility via Internet

The above setup is also possible if one or more computer shall access to the project via Internet. But in that case the Firewall and/or the router must be configured accordingly. The keyword is then amongst others the 'Portforwarding'.

Important to know is that the remote computer should have access to Port 2804.

Additional Information

In case of further questions feel free to check our Forum (<http://www.logview.info/forum/index.php>) and look for 'LogView Studio' for possible solutions or raise your questions to the community. Also we as Authors are checking the Forum to give support whenever possible.

4 Deutsch

In diesem Kapitel werden grundlegende Informationen für LogView Studio bereit gestellt. Wenn Sie neu auf dem Gebiet LogView Studio sind sollten sie die folgenden Kapitel aufmerksam durchlesen.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Spenden](#)
- [Installation](#)
- [Schnellstart Anleitung](#)
- [Grundlagen / FAQ](#)
- [Oberfläche](#)
- [Arbeiten mit LogView Studio](#)
- [Beispiele](#)

4.1 Spenden

Das Projekt LogView basiert komplett auf Spenden und wird von uns in unserer Freizeit voran getrieben. Die Spenden verwenden wir hauptsächlich um Software Lizenzen für unsere Entwicklung zu kaufen (oder zu verlängern) und um monatliche Kosten wie den Webserver für die Webseite zu decken.

wenn Ihnen unsere Software LogView Studio gefällt wäre es nett uns eine Spende zukommen zu lassen. Sie unterstützen und respektieren damit unserer bisherige und weitere Arbeit.

Paypal

Sie können über Paypal mit der folgenden EMail Adresse spenden

dominik@logview.info

Alternativ gibt es in unserem Forum eine Spenden Seite

<http://logview.info/forum/index.php?donate/>

Wir tragen dort von zeit zu Zeit Spendenziele ein die gerade für unsere Arbeit wichtig sind.

Bank Überweisung

Wenn Sie eine Überweisung an uns tätigen möchten schreiben sie uns bitte eine EMail an dominik@logview.info.

Aufgrund zahlreicher Probleme mit Datenmissbrauch in der Vergangenheit können wir unsere Bankinformationen hier nicht mehr direkt zur Verfügung stellen.

4.2 Installation

In diesem Kapitel werden Informationen u.a. zu den Themen Installation, Deinstallation, Firewall gegeben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [.Net Framework](#)
- [LogView Studio installieren](#)
- [LogView Studio updaten](#)
- [LogView Studio Deinstallieren](#)
- [Benutzerdaten entfernen](#)
- [ZIP Archiv nutzen](#)
- [Firewall](#)

4.2.1 .Net Framework

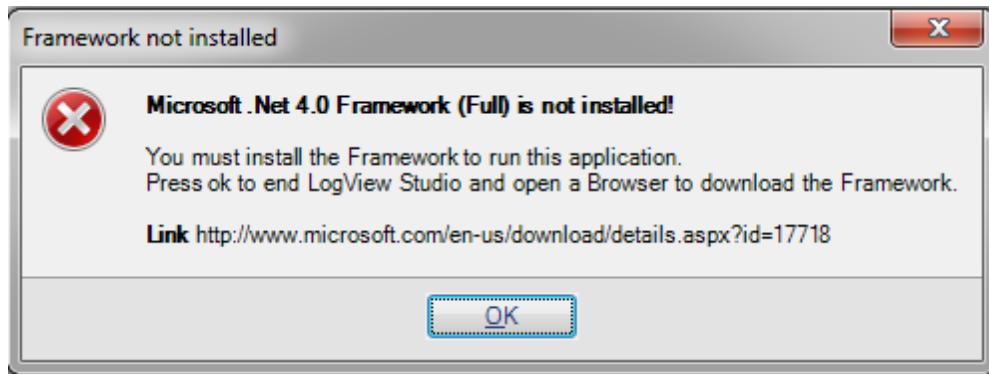
LogView Studio basiert auf dem .Net Framework von Microsoft. Es muss daher zwingend installiert sein.

Wikipedia

.NET bezeichnet eine von Microsoft entwickelte Software-Plattform zur Entwicklung und Ausführung von Anwendungsprogrammen.

Für LogView Studio wird .Net 4.0 Full benötigt. Das .Net 4.0 (welches teilweise vorinstalliert ist) reicht zum Ausführen nicht aus!

Sollte das Framework nicht installiert sein erscheint nach dem Start folgende Meldung.



Download

Der Installer steht bei Microsoft zum Download

<http://www.microsoft.com/de-de/download/details.aspx?id=17718>

Alternativ kann über Google nach ".Net Framework 4.0 Full" gesucht werden.

Installation

Die Installation des Frameworks ist selbsterklärend und bedarf keiner weiteren Erläuterung.

4.2.2 LogView Studio installieren

Download

LogView Studio kann von der LogView Webseite geladen werden.

<http://www.logview.info/forum/index.php?resources/>

Dort wird immer der aktuelle Installer zur Verfügung gestellt.

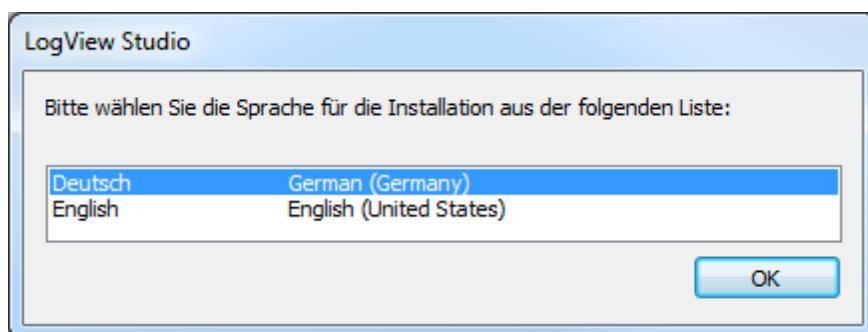
Wichtiger Hinweis

Der Installer wird erst auf der Webseite eingestellt wenn die Testphase beendet ist!

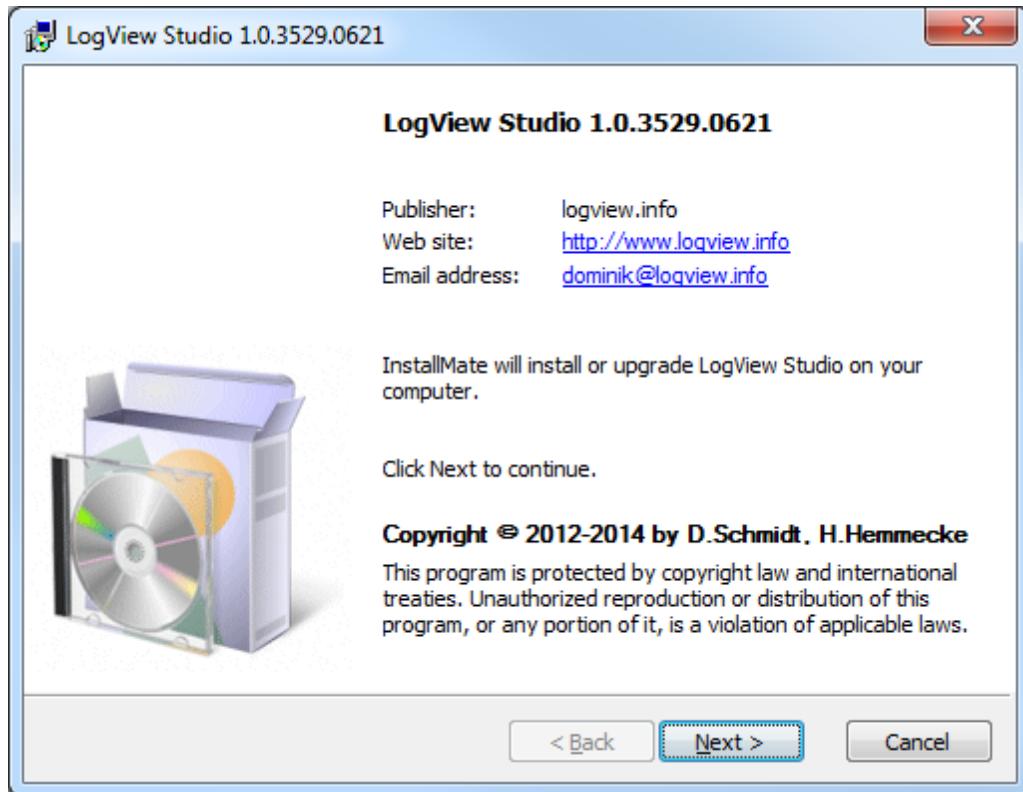
Vorher wird der Installer per mail an die Testuser geschickt.

Installation

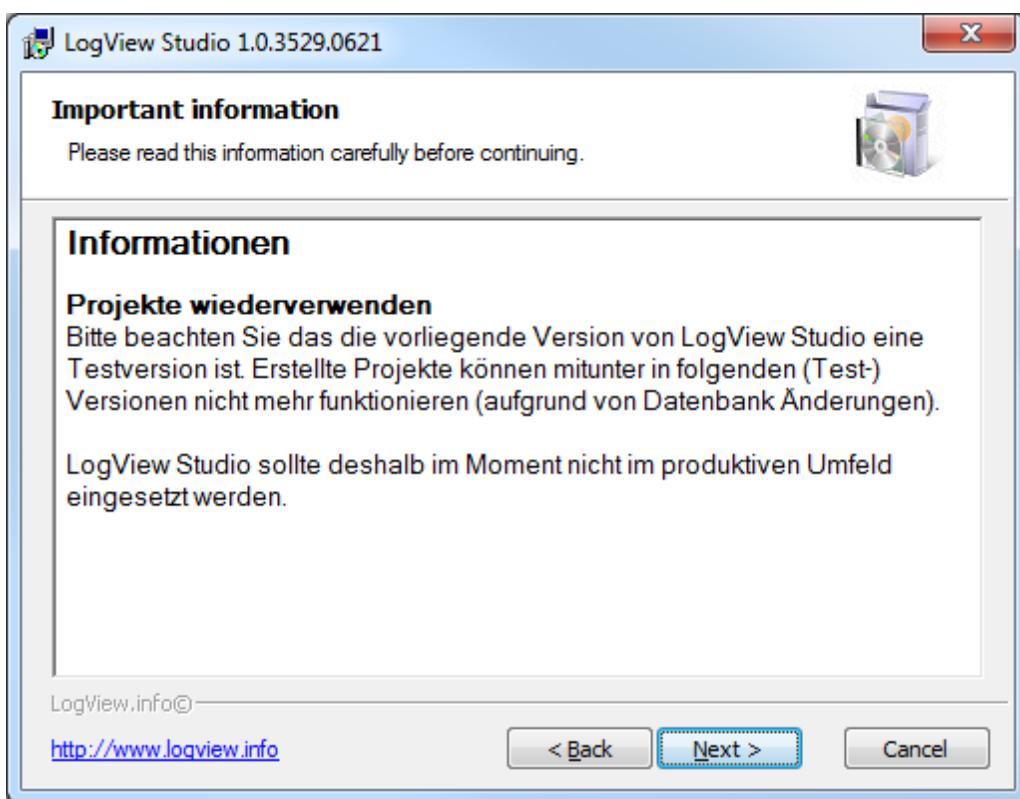
Nach dem Download den Installer per Doppelklick ausführen und eine Sprache wählen.
Die Sprache gilt nur für den Installer.



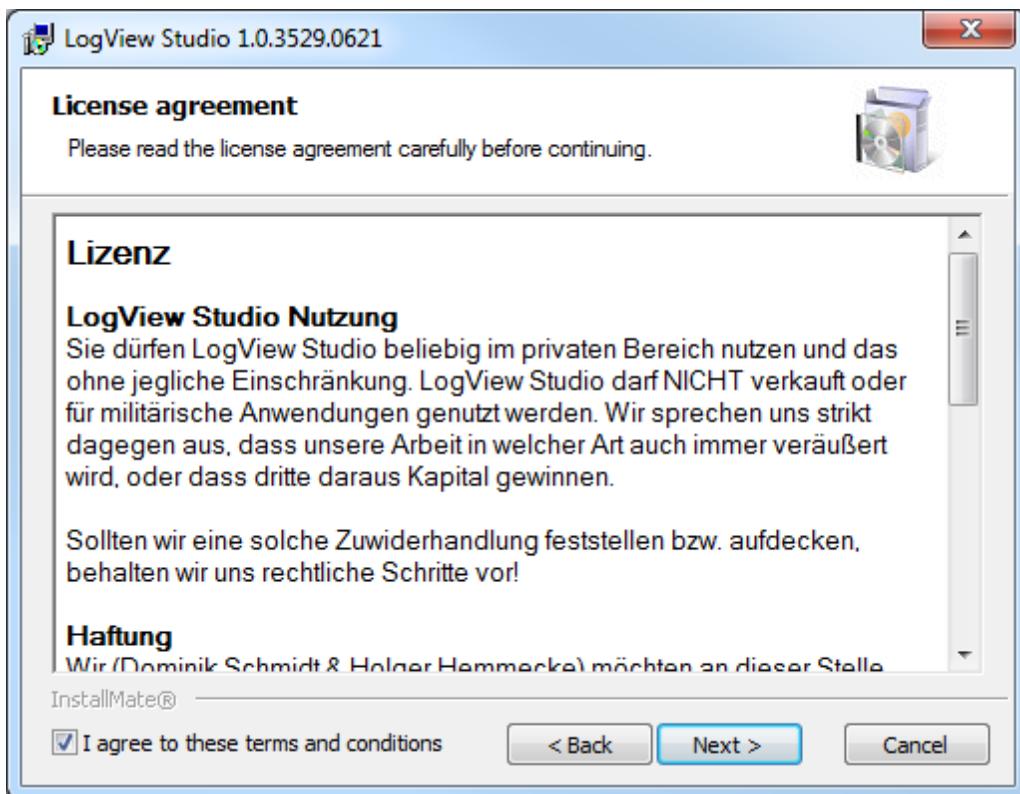
Die Infoseite mit Next überspringen.



Die Informationsseite (sofern vorhanden) lesen und beachten.



Die Lizenzbedingungen lesen und unten rechts mit der Checkbox akzeptieren.

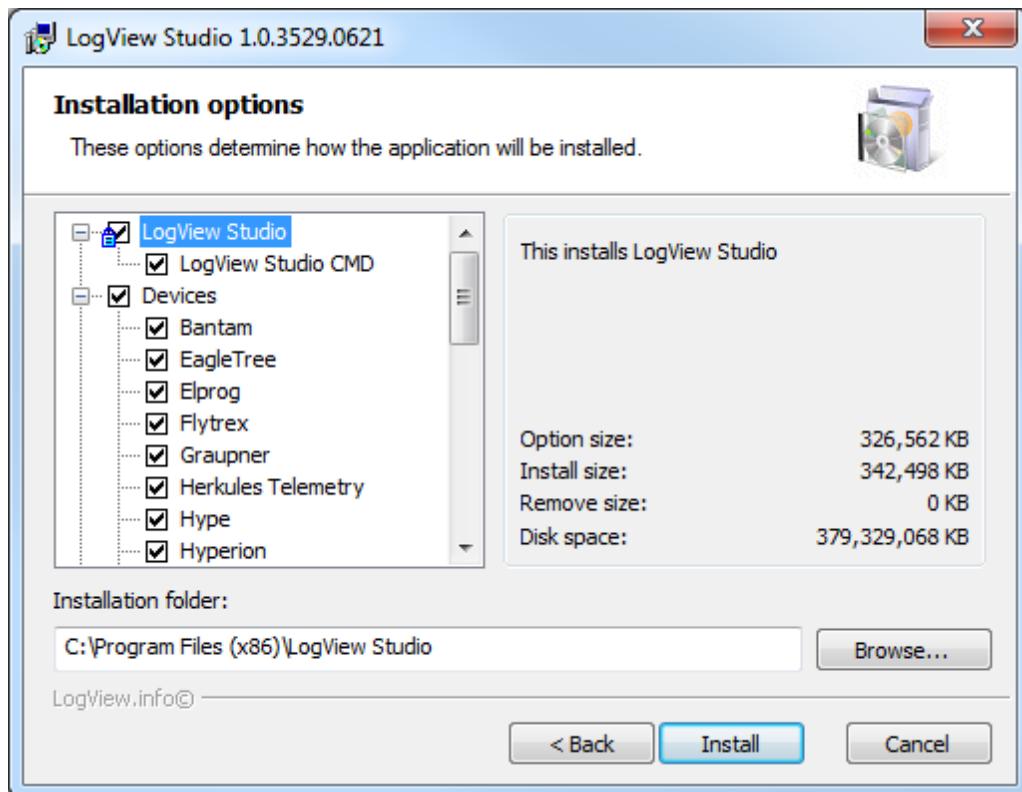


Je nach Bedarf den Installationsordner anpassen.

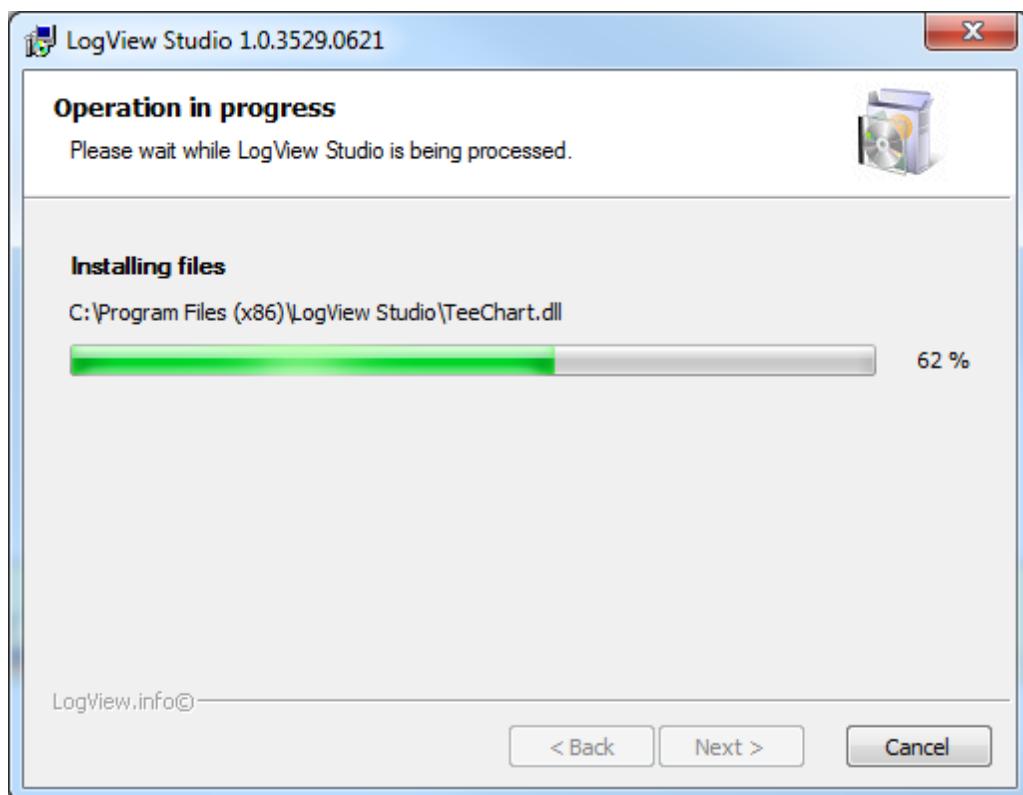
Hinweis

*Es ist durchaus von Vorteil wenn an dieser Stelle nur die Hersteller unter dem Reiter "Devices" aktiviert sind von dem Sie auch ein Gerät besitzen.
Bleiben hier alle Hersteller aktiviert kann das den LogView Studio Start eventuell verzögern weil alle Geräte beim Start erst in der Datenbank registriert werden.*

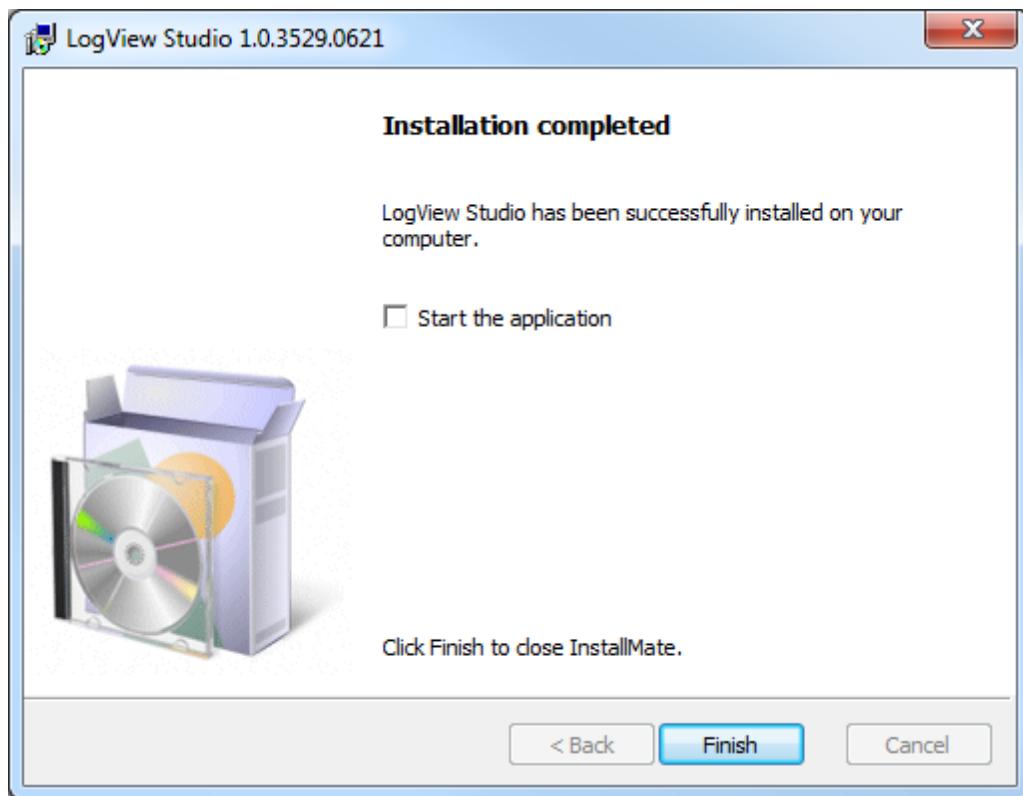
Den Installationsvorgang mit dem Install Button starten.



LogView Studio wird nun installiert ...



Abschließend kann LogView Studio nach Bedarf gestartet werden und die Installation ist beendet.



4.2.3 LogView Studio updaten

Download

LogView Studio kann von der LogView Webseite geladen werden.

<http://www.logview.info/forum/index.php?resources/>

Dort wird immer der aktuelle Installer zur Verfügung gestellt.

Update

Der Installer ist für eine Erstinstallation und ein Update der gleiche. Sollte der Installer eine vorhandene LogView Studio Installation vorfinden, werden die relevanten Dateien ersetzt.

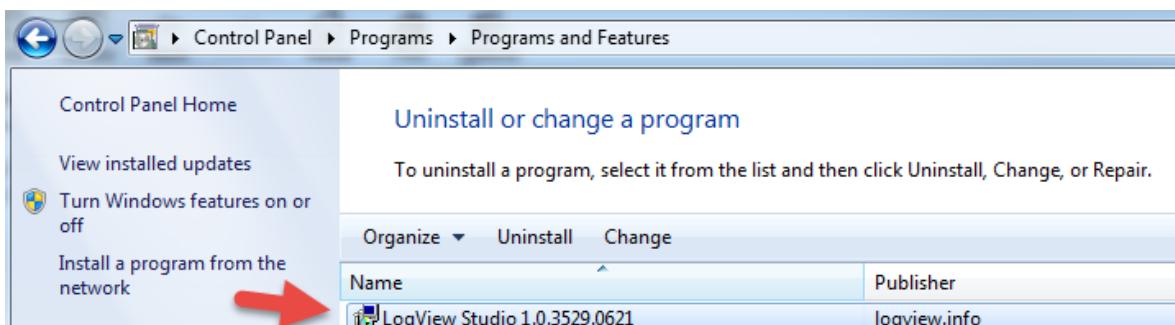
Der Update Vorgang unterscheidet sich kaum von dem [Installationsvorgang](#).

Hinweis

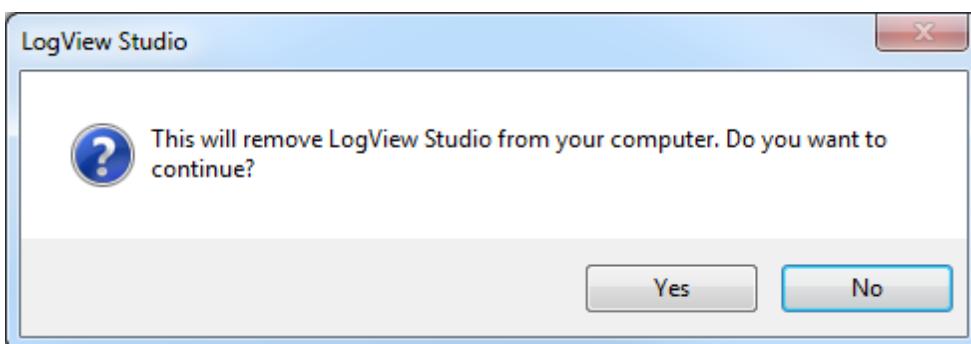
Über ein Update können auch leicht Devices hinzugefügt oder entfernt werden!

4.2.4 LogView Studio deinstallieren

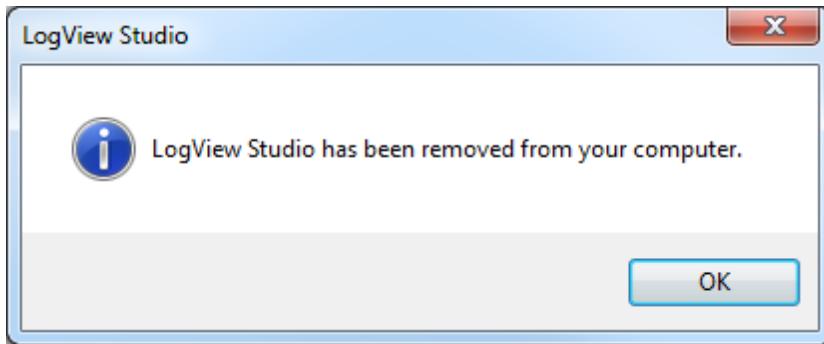
Zur Deinstallation von LogView Studio wird die Windows Funktion "Programme Deinstallieren" verwendet.



Dort den LogView Studio Eintrag auswählen und auf Uninstall klicken.
Den Dialog mit Yes / Ja beantworten.



LogView Studio ist deinstalliert.



Daten die nicht gelöscht werden

Bei der Deinstallation wird nur das Installationsverzeichnis gelöscht und alle Links entfernt. Nicht gelöscht werden:

- erstellte Projekte

Projekte die auf der Festplatte gespeichert wurden werden vom Installer nicht gelöscht. Sollten Sie diese Daten wirklich löschen wollen muss das z.B. über einen Datei Explorer erfolgen. Es kann der gesamte Project Folder gelöscht werden. Die genaue Lage des Project-Folders wurde bei der Erstellung vom User vorgegeben.

- Userdaten

LogView Studio muss einige wenige Daten im Userverzeichnis ablegen (wie z.B. die zuletzt geöffneten Projekte). Dieses Verzeichnis wird ebenfalls nicht gelöscht bei einer Deinstallation. Hier gibt es weiterführende Informationen zum Thema [User Daten löschen](#).

4.2.5 Benutzerdaten entfernen

Einige Daten müssen von LogView Studio im Benutzer Verzeichnis abgelegt werden. Das sind unter anderem folgende Daten

- zuletzt geöffnete Projekte
- Error Logs und Screenshots
- ggf. globale Grafik Templates

Diese Daten werden bei einem Deinstallieren nicht gelöscht und müssen nach Bedarf vom User entfernt werden.

Verzeichnisse

Das Verzeichnis liegt je nach System an einer anderen Stelle im Dateisystem

System	Verzeichnis
Windows XP	C:\Documents and Settings\<USERNAME>\Local Settings\Applicationdata\LogView Studio C:\Dokumente und Einstellungen\<USERNAME>\Lokale Einstellungen\Anwendungsdaten\LogView Studio
Windows Vista, 7, 8, 8.1	C:\Users\<USERNAME>\AppData\Local\LogView Studio C:\Benutzer\<USERNAME>\AppData\Local\LogView Studio
Linux	/home/<USERNAME>/LogView Studio

Registry

LogView Studio speichert keine Informationen in der Registry.

Hinweis

Die einzige Ausnahme bildet der Installer. Hier wird die Installation in der Registry eingetragen was u.a. für eine Deinstallation nötig ist.

4.2.6 ZIP Archiv nutzen

LogView Studio ist auch in einer reinen Command Line Version vorhanden. Diese Version wird normalerweise nur unter Linux als Download Quelle verwendet. Sollte (aus welchen Gründen auch immer) eine Installation mittels Installer nicht möglich sein kann dieses ZIP Paket als Ersatz verwendet werden.

Download

Die Datei lässt sich über folgenden Link laden:

<http://www.logview.info/lvstudio/LVStudio.zip>

Entpacken

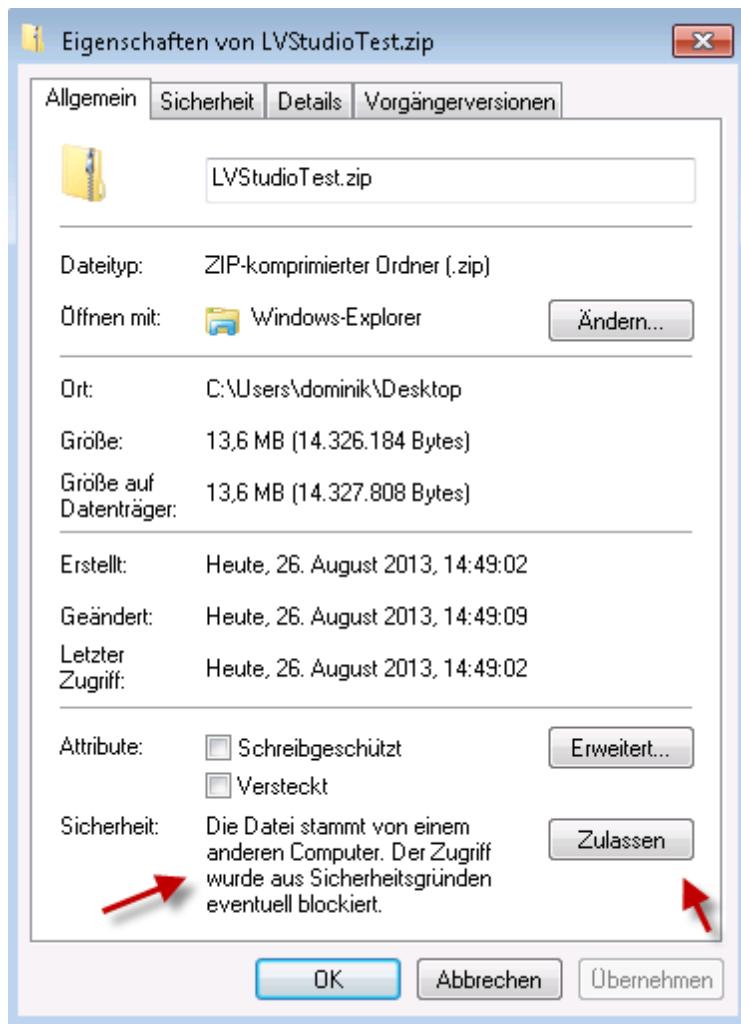
Die Datei lässt sich mit jedem üblichen Tool entpacken welches ZIP Komprimierung beherrscht.

Wichtig dabei ist nur das die Ordnerstruktur komplett erhalten bleibt.

Es wäre hilfreich unter Windows vor dem entpacken zu prüfen ob die [ZIP Datei eventuell von Windows geblockt](#) wurde.

4.2.6.1 ZIP Blockade entfernen

Seit Windows XP SP2 werden in Windows ZIP Dateien als potentiell gefährlich beim Download eingestuft. Man kann das erkennen wenn man die Datei Eigenschaften von der ZIP Datei ansieht.



Des weiteren kommt bei dem Start von LogView Studio eine Sicherheitswarnung.



Die ZIP Datei lässt sich trotzdem entpacken. Wenn das allerdings mit den Windows Board Mitteln für ZIP Dateien geschieht werden auch alle entpackten Dateien geblockt.

Das führt dazu das LogView Studio u.a. die nötigen Dateien für die Geräte nicht laden kann. Im Log erscheinen dann folgende Infos:

```
Mo 14:54:44.805 : [ 1, Main      ] An attempt was made to load an assembly from a network location which would have caused the assembly to be sandboxed in previous versions of the .NET Framework. This release of the .NET Framework does not enable CAS policy by default, so this load may be dangerous. If this load is not intended to sandbox the assembly, please enable the loadFromRemoteSources switch. See http://go.microsoft.com/fwlink/?LinkId=155569 for more information.
Mo 14:54:44.810 : [ 1, Device     ] Jeti.dll is not a valid LogView Studio Device.
```

Problemlösung 1

Die Datei nicht mit den Windows Mitteln entpacken sondern 7Zip oder ein ähnliches Packtool verwenden. Dabei wird die Blockade nicht an die entpackten Dateien weitergereicht.

Problemlösung 2

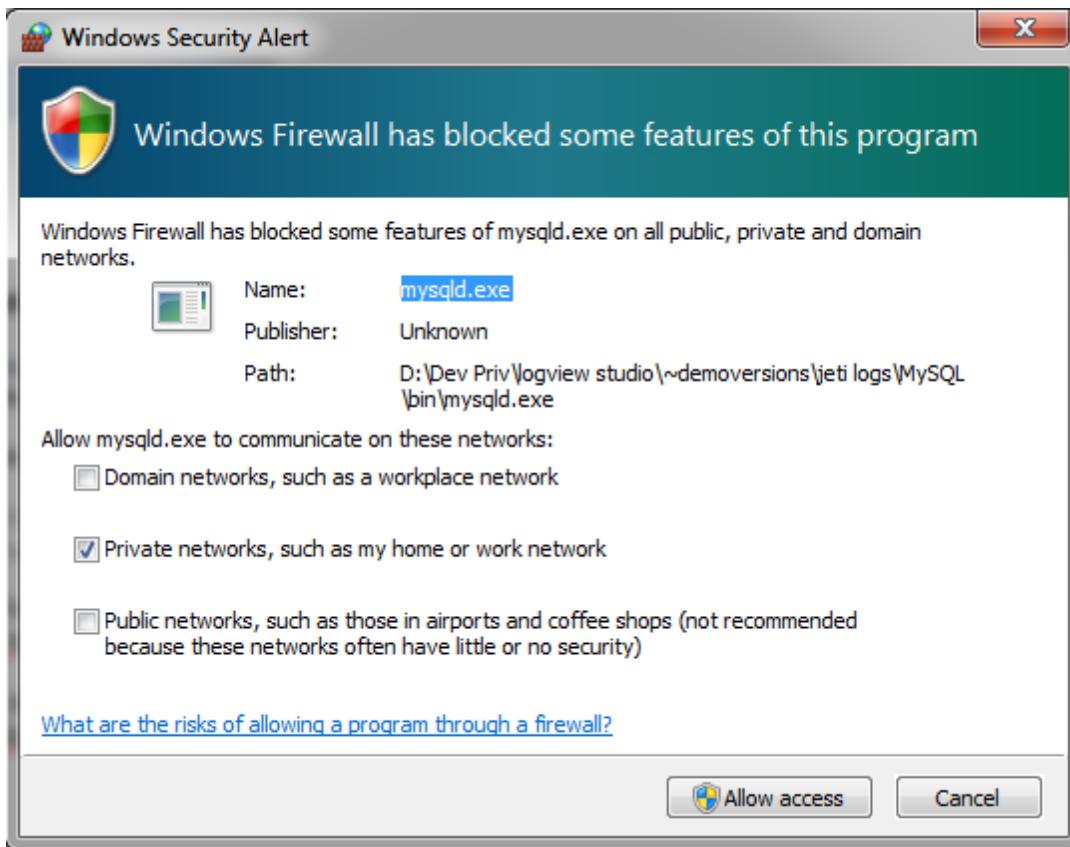
Die Eigenschaften der ZIP Datei aufrufen und auf den Button "Zulassen" klicken. Damit wird die Blockade aufgehoben und die Datei kann mit allen Tools sauber entpackt werden.

4.2.7 Firewall

Dialog für mysqld.exe

Nach dem ersten Start von LogView Studio können mitunter ein oder zwei Firewall

Dialoge erscheinen.



LogView Studio nutzt einen MySQL Datenbankserver für die Datenspeicherung. Dem Server wird mit den Einstellungen der Zugriff auf das lokale Netzwerk gewährt.

Dialog für LvGui.exe / LvCmd.exe

Des weiteren kann ein ähnlicher Dialog noch für LogView Studio selber erscheinen. In dem Fall wäre der Name dann LvGui.exe oder LvCmd.exe. Diese LogView Studio Dateien benötigen ebenfalls Zugriff auf das Netzwerk um die Datenbank zu erreichen oder abfragen im Internet durchzuführen. Auch hier sollte ein Zugriff auf das private Netzwerk erlaubt werden.

Wird der Zugriff verweigert kann es zu Funktionsstörungen im Betrieb kommen.

Bei anderen Firewall Herstellern werden die Dialoge sicherlich ähnlich aussehen. Auch hier muss der Datenbank und ggf. LogView studio Zugriff auf das Netzwerk gegeben werden.

Installer

Der Installer richtet bei der Installation im Normalfall eine entsprechende Regel für die Microsoft Firewall ein.

4.3 Schnellstart

In diesem Kapitel wird kurz und knapp erklärt wie man mit LogView Studio ein Projekt erstellt, ein Gerät (Device) einfügt, das Device konfiguriert, eine Aufzeichnung startet

um am Ende eine Grafik darzustellen. Dabei wird aber nur ansatzweise aufgezeigt welche Möglichkeiten LogView Studio weiterhin bietet.

Hinweis

Der hier gezeigte Schnellstart spiegelt ganz grob die Funktionsweise von LogView V2 wieder. Wer also mit dem alten LogView gearbeitet hat sollte sich mit dieser Anleitung einigermaßen schnell zurecht finden.

Generell ist natürlich zu empfehlen die additiven Informationen zu LogView Studio zu lesen.

Voraussetzungen

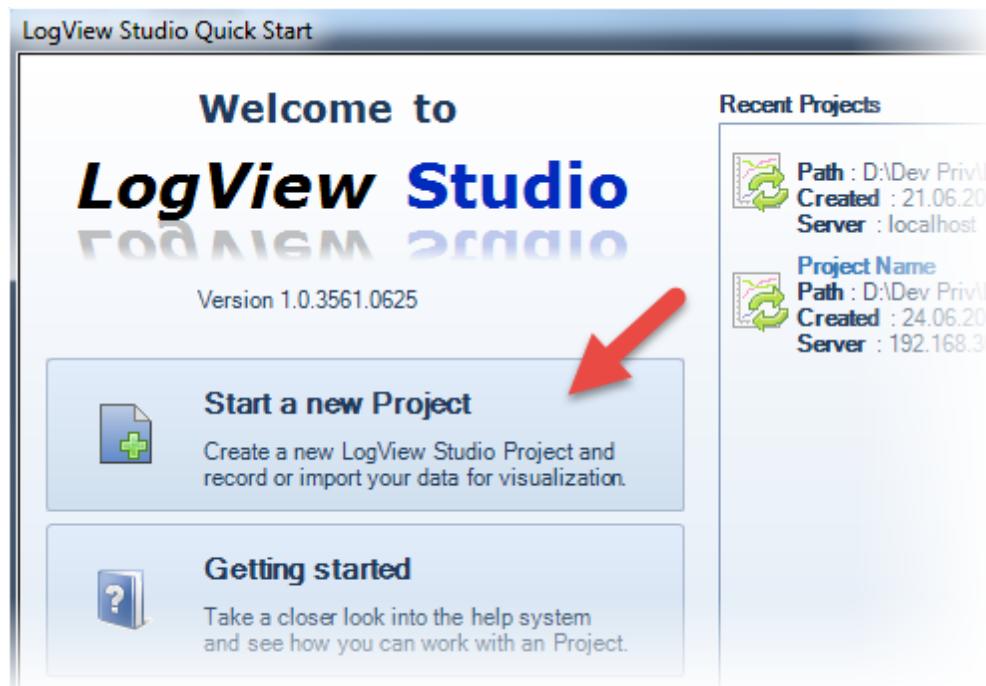
- [.Net Framework 4.0 Full](#) muss installiert sein
- [LogView Studio ist installiert](#)

LogView Studio starten

- LogView Studio aus dem Startmenü starten oder 'LvGui.exe' direkt ausführen
- Es können [Firewall Meldungen](#) erscheinen wo der Zugriff auf das lokale Netzwerk zugelassen werden muss.
- LogView Studio sollte nun mit einem Quick Start Dialog erscheinen

Projekt anlegen

In dem Quick Start Menü "Start a new Project" anklicken



Im Dialog "New Project" einen Projekt Namen eintragen und darunter einen Ordner wo das Projekt gespeichert wird. Mit Ok das Projekt erstellen lassen.

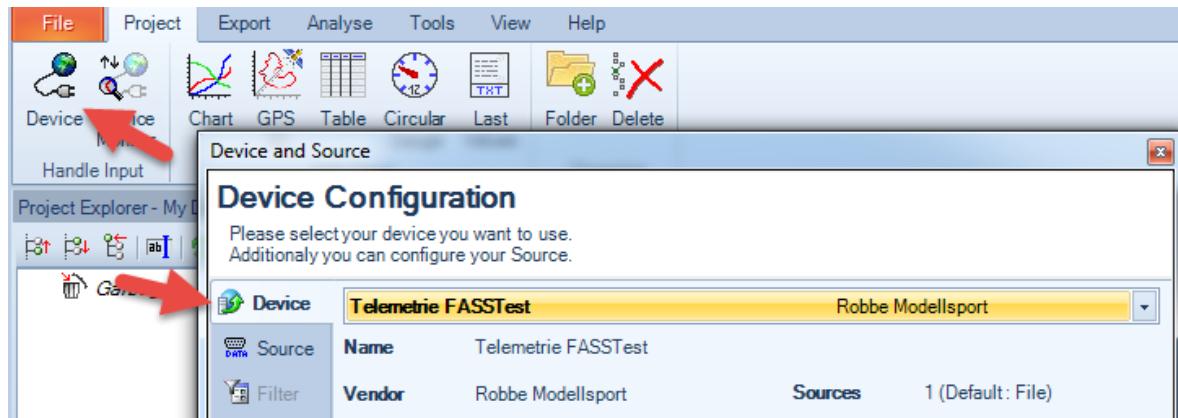


Wenn das Projekt erfolgreich erstellt wurde, sollte im Project Explorer nur ein Eintrag "Garbage" vorhanden sein.

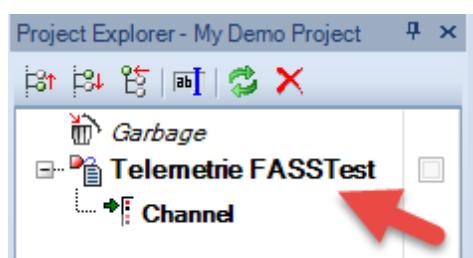
Device hinzufügen

Als nächster Schritt wird ein Device hinzugefügt von dem wir Daten aufzeichnen bzw. verarbeiten möchten.

Im Menü Project / Handle Input auf Device klicken und in dem folgenden Dialog das Device auswählen.



Den Dialog mit Ok beenden. In dem zuvor erstellten Projekt wird nun ein neues Device eingetragen.



Hinweis

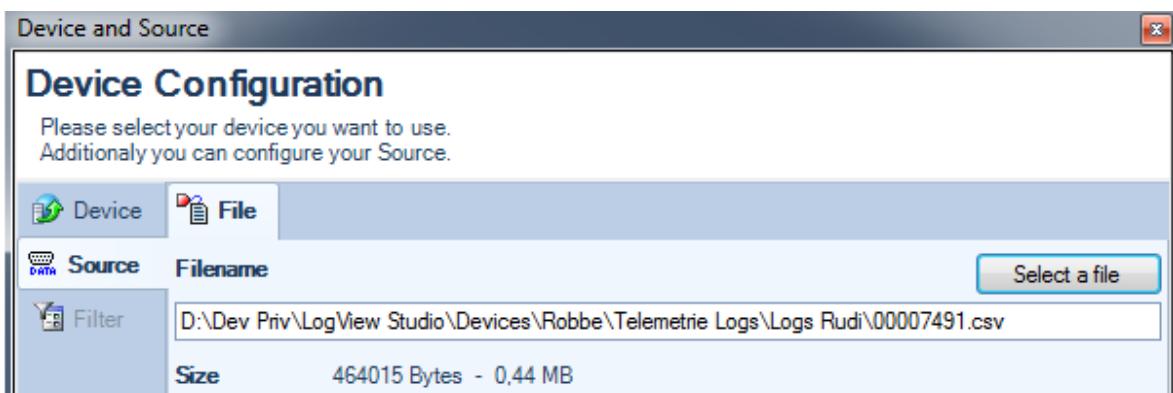
An dieser Stelle wurde bewusst auf das Einrichten einer Source (Datenquelle) verzichtet.

Im Normalfall kann natürlich Device und Source in einem Schritt konfiguriert werden!

Source konfigurieren

Nun wird die Source (also die Datenquelle) für das Device eingerichtet. LogView Studio kann eine ganze Palette von Datenquellen handhaben wie z.B. Seriell, HID, TCP, UDP, Dateien, welche Datenquellen letztlich konfiguriert werden können wird für das Device individuell festgelegt. Im Fall der Telemetrie FASSTest können z.B. nur Textdateien eingelesen werden.

Mit einem Doppelklick auf das Device "Telemetrie FASSTest" öffnet sich der Source Dialog.



Da das Device nur eine Datei Source unterstützt wird hier auch nicht mehr angezeigt. Hat ein Device mehrere Sources kann eine Source ausgewählt werden die anschließend für das Device verwendet wird.

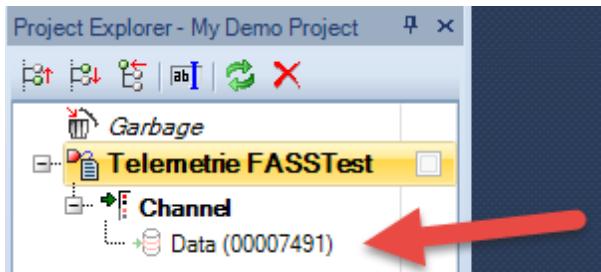
Mit Ok den Dialog schließen und direkt den Datei Import starten.

Hinweis

Text und Binärdateien werden von LogView Studio sofort nach Schließen des Dialogs verarbeitet. Alle anderen Sources müssen explizit gestartet werden über die Checkbox rechts neben dem Device im Project Explorer.



Nach dem Datenimport bzw. dem Start über die Checkbox werden im Projekt Explorer neue Elemente eingetragen - die Datensätze.



Diese Datensätze repräsentieren eine Ansammlung von Daten innerhalb der Datenbank. Und anders als in LogView V2 wird nicht sofort eine Grafik erzeugt.

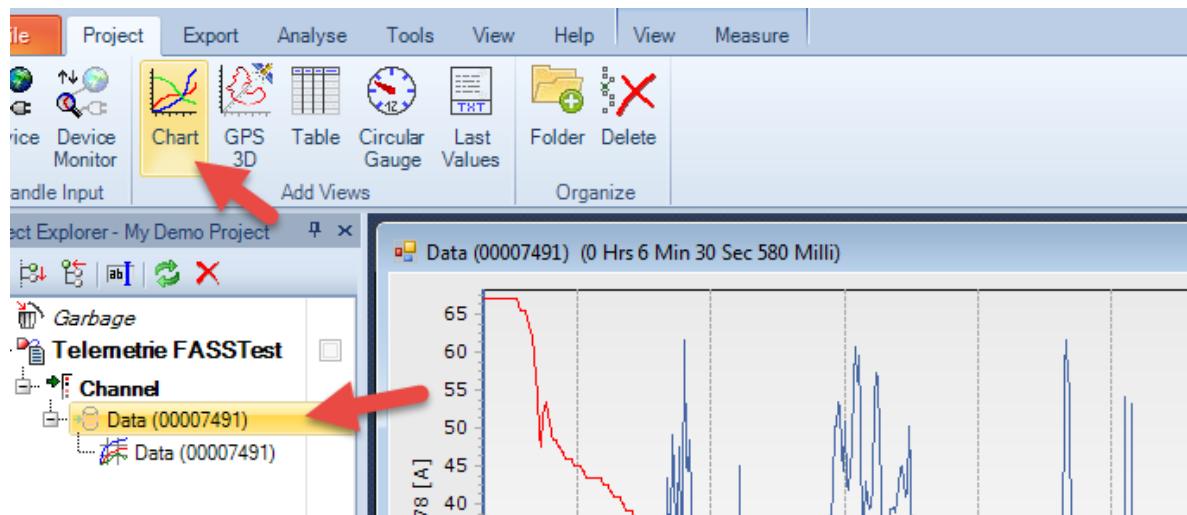
Wie aus einem Datensatz Grafiken erzeugt werden wird anschließend erklärt.

Weitere Datendateien importieren

Sollen weitere Dateien für ein Device importiert werden so können diese auch sehr bequem per Drag and Drop einfach aus dem Datei Explorer auf das Device im Project Explorer von LogView Studio gezogen werden. Die Daten werden sofort importiert und verarbeitet.

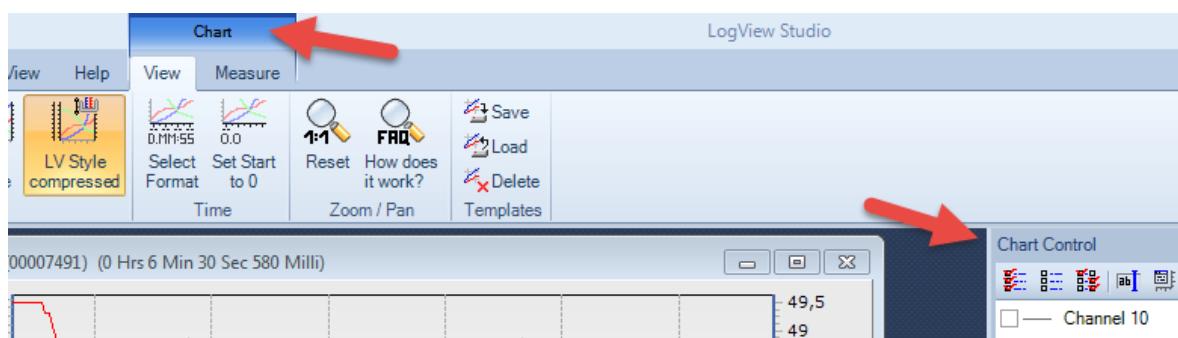
Statisch Daten anzeigen

Wenn für einen Datensatz eine Grafik erzeugt werden soll, dann muss dazu der Datensatz selektiert werden. Danach im Menu Project / Add Views auf Chart klicken.



Die Daten werden nun in einem Chart dargestellt.

Über das Fenster Chart Control und das menü "Chart" kann die Darstellung nun nach Belieben angepasst werden.



Live Daten anzeigen

Neben statischen Anzeigen (die Daten ändern sich nicht mehr) kann LogView Studio auch Live Daten darstellen. Dies ist vor allem bei Geräten interessant die kontinuierlich Daten liefern wie z.B. Ladegeräte oder Wetterstationen.

Dazu muss im Projekt Explorer ein Channel eines Devices selektiert werden. Anschließend kann mit Chart oder Circular Gauge eine Live Anzeige angezeigt werden die sich alle x Sekunden mit neuen Daten versorgt.

Projekt neu laden

Ein Projekt kann über drei Wege neu geladen werden.

- Nach dem Start im Quick Start Dialog
- Über die Liste im File Menü
- Über File - Open Project ...

Abschließende Anmerkung

Auf den ersten Blick mag dieser Schnellstart etwas komplex wirken. Wer jedoch einmal erfolgreich ein Projekt in LogView Studio angelegt und genutzt hat sollte sich recht schnell zurecht finden. Zudem gibt es bei fast allen Buttons in LogView Studio Tooltip Hilfen mit direktem Link zu dieser Hilfe.

4.4 Grundlagen / FAQ

In diesem Kapitel werden Grundlagen rund um LogView Studio besprochen. Die Texte gehen nicht immer auf alle Aspekte ein, sollen aber einen ersten Überblick verschaffen.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Was ist LogView Studio?](#)
- [Was ist ein Project?](#)
- [Was ist ein Device?](#)
- [Was ist eine Source?](#)
- [Was ist ein Channel?](#)
- [Was ist ein Dataset?](#)
- [Was ist eine View?](#)
- [Static und Live View](#)
- [GUI / CLI - Was ist das?](#)

- [LogView Studio auf Linux?](#)
- [LogView Studio im Netzwerk](#)

4.4.1 Was ist LogView Studio?

LogView Studio ist eine Software zum Aufzeichnen und Verarbeiten von Daten die von unterschiedlichsten Geräten geschickt werden können. Der Fokus liegt derzeit im Modellbau Bereich wo viele Datenlogger, Ladegeräte und auch Telemetrie unterstützt wird. Danaben kann LogView Studio aber auch für Eigenentwicklungen genutzt werden. Letztlich spielt das Gerät aber eine untergeordnete Rolle. Sofern das Datenprotokoll bekannt ist können die Daten im Normalfall aufgezeichnet werden.

Die aufgezeichneten Daten können anschließend in diversen Ausgabe- und Exportformaten dargestellt oder gespeichert werden.

Eigenschaften

LogView Studio hat einige Herausstellungsmerkmale die es von anderer Software unterscheidet:

- Es können beliebig viele Geräte gleichzeitig genutzt / aufgezeichnet werden (hängt von den verfügbaren Ressourcen des Computers ab)
- Die Daten werden in einer Datenbank gespeichert. Damit können User eigene Datenauswertungen erstellen oder Applikationen entwickeln.
- Die Datenaufzeichnung ist komplett von der Datenausgabe entkoppelt.
- LogView Studio ist netzwerkfähig. Daten können an verschiedenen Rechnern aufgezeichnet und in ein gemeinsames Projekt gespeichert werden.
- Es gibt eine Kommandozeilen Version die auch unter Linux lauffähig ist. Selbst auf einem Raspberry Pi kann man LogView Studio samt Datenbank betreiben.
- Die Oberfläche kann mehrere Auswertungen parallel geöffnet haben.
- Die Daten können über diverse Eingangskanäle aufgezeichnet werden wie z.B. Seriell, Netzwerk oder Dateien. Selbst die zyklische Abfrage einer Webseite wäre umsetzbar.
- Es gibt einen eingebauten Webserver für die schnelle Darstellung der Daten im Netzwerk.

Weitere Informationen

[Schnellstart](#) - schneller Einstieg in ein Projekt mit einem Gerät

[Beispiel Projekt](#) mit mehreren Geräten

4.4.2 Was ist ein Project?

Unter einem Project muss man sich in LogView Studio einen Container vorstellen in dem alle erzeugten Daten gespeichert werden. Innerhalb dieses "Containers" werden unterschiedliche Elemente eingefügt wie Geräte, Datensätze, Ansichten, etc. Diese einzelnen Elemente werden in einer Baumstruktur eingesortiert und erleichtern so den Überblick über das gesamte Project.

Ein Project kann weiterhin mit einer eigenen Ordnerstruktur erweitert werden. Dadurch können die erzeugten Daten nach User Vorstellung sortiert und eingruppiert werden.

Wenn ein Project angelegt wird dann ist es dem User überlassen in welchem Kontext er dieses Project betrachtet. Es gibt eine ganze Reihe von möglichen Ansätzen was man

mit einem Project realisieren könnte:

- Es werden alle Daten für ein bestimmtes Gerät (z.B. ein Ladegerät) in einem Project gesammelt.
- Es werden Daten von verschiedenen Geräten in einem Project gesammelt um z.B. alle Daten zu einem Flieger zentral zu organisieren.
- Es wäre auch vorstellbar all seine Daten in einem Project zu sammeln die z.B. unter den Begriff "Hobby Modellbau" fallen. Dieses Vorgehen wird aber nicht empfohlen.

Letztlich ist es aber dem User überlassen welche Geräte er einem Project hinzufügt und welche Daten er in einem Project speichert. LogView Studio bietet mit den Projekten nur den Rahmen.

Hinweis

Es kann nur ein Project in LogView Studio geöffnet werden.

Hinweis

Die Anzahl der Elemente (Geräte, Datensätze, ...) ist in LogView Studio nicht limitiert und wird lediglich über die verfügbaren Rechner Ressourcen eingeschränkt.

Hinweis

LogView Studio nutzt zum Speichern der Daten nicht eine einzige Datei sondern ein Verzeichnis.

Weitere Informationen

[Ein Project anlegen](#)

[Elemente in einem Project hinzufügen](#)

4.4.3 Was ist ein Device?

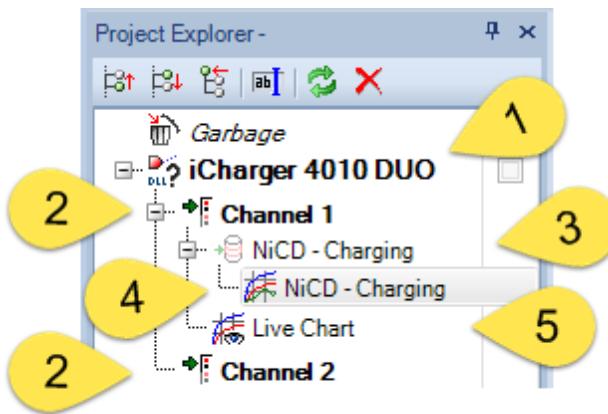
Ein Device ist aus LogView Studio Sicht ein Gerät welches über eine Schnittstelle Daten liefern kann. LogView Studio unterstützt dabei eine Vielzahl von [Schnittstellen](#) also wie die Daten an den Rechner geliefert werden.

Die Anzahl der Devices die in einem Project genutzt werden können ist nicht begrenzt (nur durch die Ressourcen des Rechners). Es können also in einem Project mehrere Devices mit unterschiedlichen Schnittstellen parallel genutzt / aufgezeichnet werden.

Beispiele für Devices

- Datenlogger
- Ladegerät
- Wetterstation
- Arduino mit Sensoren

Beispielansicht



1 Device

2 Channel

3 Dataset

4 Chart View (statisch)

5 Chart View (Live)

Weitere Informationen

[Schnellstart](#)

[Anlegen eines Devices in einem Project](#)

[Beispiel Project](#)

4.4.4 Was ist eine Source?

Wenn in LogView Studio ein Device in ein Project hinzugefügt wird, dann muss auch immer einer Source angegeben werden. Die Source gibt vereinfacht gesagt an wie die Daten in den Rechner gelangen. LogView Studio unterstützt inzwischen eine ganze Reihe von möglichen Sources die für ein Gerät genutzt werden können:

- **Serielle Schnittstelle** (COM)

Die Daten werden über einen seriellen Port empfangen.

- **USB HID** (Human Interface Device) (USB)

Die Daten kommen über einen USB Port. Der USB Port arbeitet im HID Modus:

http://en.wikipedia.org/wiki/Human_interface_device

HID Geräte sind im Normalfall Mäuse oder Tastaturen die über USB angeschlossen werden. Die Technik der Datenübertragung wird aber gerne auch für Geräte genutzt da die Übertragung von Daten von und zum PC sehr einfach gehalten ist.

- **TCP Client** (TCP)

Die Daten kommen per Netzwerk von einem TCP Server.

- **UDP Receiver** (TCP)

Die Daten kommen per Netzwerk von einem UDP Sender.

- **Textdateien** (File)

Die Daten kommen aus einer Datei.

- **Binärdateien** (File)

Die Daten kommen aus einer binären Datei.

Hinweis

Welche Sources bei einem Device bereit gestellt werden wird bei der Programmierung des Devices festgelegt. Es würde auch keinen Sinn ergeben wenn ein Ladegerät auch immer eine TCP Client Source hat - welcher Lader hat schon einen Netzwerk Anschluss.

Für jedes Devices können in LogView Studio mehrere Sources bereitgestellt werden. Davon kann aber immer nur eine genutzt werden. LogView Studio kann also nicht für ein Gerät Daten über eine serielle Schnittstelle einlesen und gleichzeitig ein Textfile des Laders importieren.

Die Source wird beim Anlegen des Devices direkt mit angegeben. Dazu gibt es in dem Dialog einen extra Reiter Source. Alternativ kann eine Source auch später hinzugefügt oder geändert werden. Entweder per Doppelklick im Project Explorer auf das Device oder aber per Rechtsklick auf das Device und dann 'Source Configuration'.

Weitere Informationen

[Schnellstart](#)

[Anlegen eines Devices mit Source in einem Project](#)

[Beispiel Project](#)

4.4.5 Was ist ein Channel?

Wenn in LogView Studio ein Device zu einem Project hinzugefügt wird, dann hat dieses Device immer mindestens einen Channel. Die Channel separieren die eingehenden Daten in logisch zusammenhängende Bereiche.

Hinweis

Channel können weder verschoben noch direkt gelöscht werden weil sie immer zu einem Device gehören.

Die Anzahl der Channels wird bei der Entwicklung / Einbindung eines Devices festgelegt. Sie kann nicht vom User beeinflusst werden.

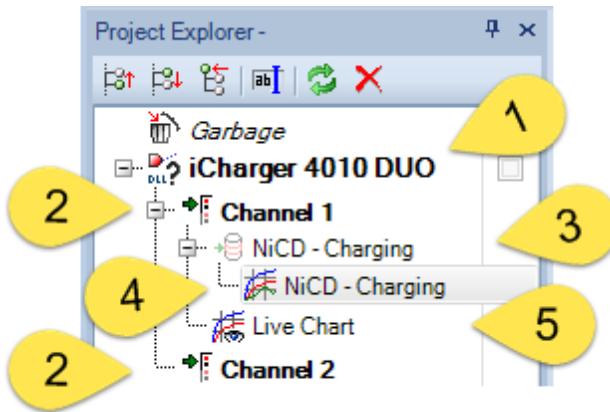
Beispiel 1

Ein Ladegerät an dem 2 Akkus gleichzeitig bearbeitet werden können hat in LogView Studio zwei Channel. Die Daten die vom Ladegerät gesendet werden splittet LogView Studio automatisch und trägt sie entweder bei Channel 1 oder 2 ein.

Beispiel 2

Ein Datenlogger der auf einer SD Karte einige Messwerte speichert wird im Normalfall in LogView Studio nur einen Channel besitzen. Die einzulesenden Daten werden dann diesem Channel zugeordnet.

Beispielansicht



1 Device

2 Channel

3 Dataset

4 Chart View (statisch)

5 Chart View (Live)

4.4.6 Was ist ein Dataset?

Ein Dataset (Datensatz) ist ein zusammenhängender Block von Daten die von einem Device für einen bestimmten Channel erzeugt wurden. Ein Dataset wird immer einem Channel zugeordnet und ist verschiebbar in einen Ordner.

Datasets lassen sich am einfachsten anhand eines Beispiels erklären:

Ein Ladegerät kann (im einfachsten Fall) dazu genutzt werden einen Akku zu entladen oder zu laden. Wenn das Ladegerät also einen Akku lädt erzeugt LogView Studio einen Datensatz mit dem Namen "Akku Laden" (Beispiel). Wechselt der Lader zum Entladen erzeugt LogView Studio einen neuen Datensatz mit "Akku Entladen".

(Anmerkung: Die abschließende Benamung des Dataset ist abhängig vom verwendeten Device und den daraus empfangenen Daten. Normalerweise sollte der Name auch den Typ des aktuell angeschlossenen Akkus enthalten - entsprechend der gewählten Einstellung am Ladegerät. Weiterhin kann gelegentlich auch eine Datums- und Zeitinformation enthalten sein, wann der Datensatz erstellt wurde.)

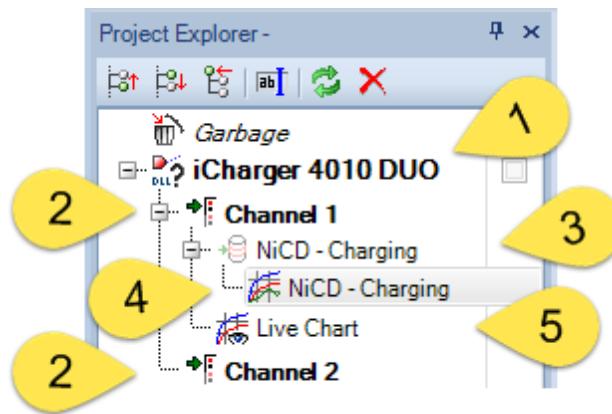
Hinweis

Ein Dataset ist nur eine logische Gruppierung von Daten in der Datenbank. Ein Dataset dient als Referenz für Anzeigen wie eine Grafik um die relevanten Daten aus der Datenbank zu bestimmen. Ein Dataset erzeugt aber nicht automatisch eine Grafik noch ist er damit gleichzusetzen!

Hinweis

Datasets werden immer automatisch von einem Device erzeugt.

Beispielansicht

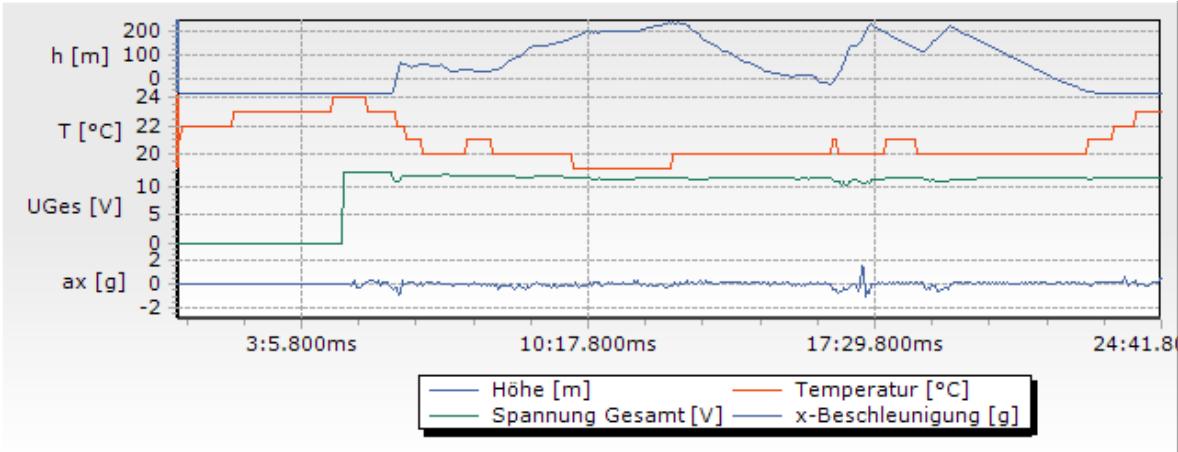


- 1 Device
- 2 Channel
- 3 Dataset
- 4 Chart View (statisch)
- 5 Chart View (Live)

4.4.7 Was ist ein Sensor?

Ein Sensor beschreibt in LogView Studio eine Messgröße die erfasst wird. Sensoren sind dabei einem Channel zugeordnet und werden in der Regel über das Device definiert (eine Ausnahme bilden hier [Zero Sensor Devices](#)).

Sensoren lassen sich am einfachsten beschreiben wenn man einen Chart betrachtet:



Jeder der dargestellten (bzw. verfügbaren) Messwerte stellt aus LogView Studio Sicht einen Sensor dar.

4.4.8 Was ist eine View?

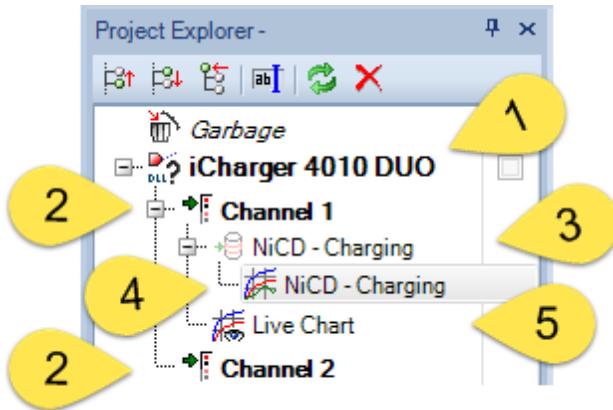
Eine View ist die Darstellung der Daten eines Datasets oder eines Channels. In LogView Studio gibt es derzeit folgendes Views:

- Chart - grafische Darstellung von Daten in Kurven
- GPS 3D - Darstellung von GPS Daten in einer 3D Ansicht (nur wenn auch GPS

Daten vorhanden sind)

- Table - tabellarische Darstellung der Daten
- Circular Gauge - Darstellung der letzten Daten als Rundinstrument
- Last Values - Darstellung der letzten Daten als Text

Beispielansicht



1 Device

2 Channel

3 Dataset

4 Chart View (statisch)

5 Chart View (Live)

4.4.9 Static und Live View

Wer mit LogView Studio Daten aufzeichnet wird feststellen das nach dem Start keine visuellen Ausgaben vorhanden sind. Diese müssen erst einem Element Dataset oder Channel im Project hinzugefügt werden.

Generell gibt es zwei verschiedene Arten von Views (Ansichten):

Statische Views

Statische Views werden für einen vorhandenen Dataset (Datensatz) einmal angelegt und ändern danach nicht mehr den Datenbestand. (Das Layout der View kann aber vom User angepasst werden.)

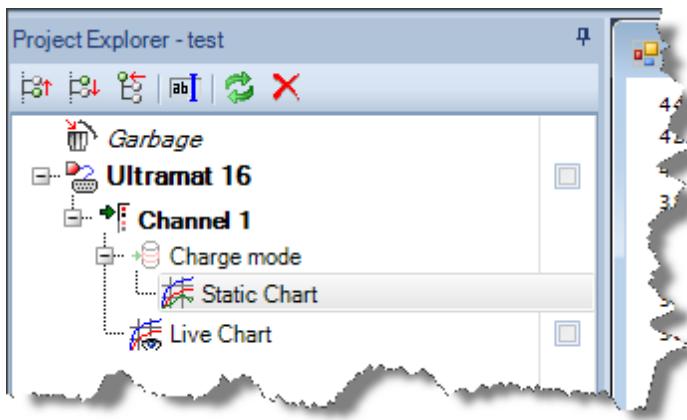
Live Views

Live Views werden einem Channel (Kanal) zugeordnet und lesen in zyklischen Abständen die neuen Daten eines Channels aus der Datenbank.

Hinweis zu Live Views

Live Views müssen immer vom User (genau wie ein Device) gestartet werden. Jeder Live View Eintrag im Project Explorer hat dafür eine Checkbox am rechten Rand die aktiviert werden muss. Würden alle Live Ansichten nicht an- bzw. abschaltbar sein würde es irgendwann zu Performance Problemen kommen.

Beispiel



In diesem Project ist ein Device eingefügt (Ultramat 16) welches einen Channel (einen Ladeausgang) besitzt.

Das Device wurde gestartet und die Daten für einen Ladevorgang wurden aufgezeichnet woraus ein Dataset "Charge Mode" entstanden ist.

Um den Datensatz nun darzustellen wurde ein statischer Chart eingefügt. Dieser Chart enthält alle Daten die zu dem Datensatz gehören. Die Daten des Chart werden von LogView Studio nach dem Erstellen nicht mehr verändert.

Der Live Chart ist an den Channel gebunden. Er zeigt die zuletzt eingetragenen Daten für den Channel 1. Die Daten im Live Chart werden stetig (sofern er aktiviert wurde) aktualisiert. Die Ansicht ähnelt ein wenig dem eines Oszilloskops - nur bedeutend langsamer.

4.4.10 Garbage - der Mülleimer

Wenn der User im Project Elemente löscht werden diese zunächst im Garbage (Mülleimer) abgelegt. Das Vorgehen ist sehr ähnlich dem des Windows Recycle Bin. Versehentlich gelöschte Elemente können so wieder hergestellt werden.

Hinweis

Nachdem ein neues Project angelegt wurde ist der Garbage immer das erste Element im Project. Es ist also ein guter Indikator dafür ob das Projekt erfolgreich angelegt wurde.

4.4.11 GUI / CLI - Was ist das?

Es ist immer mal wieder die Rede von einer GUI oder auch von einem CLI (oder auch CMD) Tool.

Erklärung GUI

GUI ist eigentlich die Abkürzung für Graphical User Interface. Oder anders gesagt eine Software die grafische anstatt rein Text basierte Ausgaben macht. In LogView Studio ist die GUI das was die meisten User nutzen und kennen werden.

Hinweis

Die Datei welche die LogView Studio GUI startet ist LvGui.exe.

Erklärung CLI

CLI steht für Command Line Interface und CMD oder Command Tool besagt im Grunde das gleiche. Es ist eine Software die im reinen Textmodus arbeitet und üblicherweise in einer Shell gestartet wird. Mit LogView Studio CLI sind keine grafischen Ausgaben möglich.

Hinweis

Die Datei welche die LogView Studio GUI startet ist LvCmd.exe.

Unterschiede

Folgende Unterschiede ergeben sich zwischen GUI und CLI

Funktion	GUI (LvGui.exe)	CLI (LvCmd.exe)
Anzahl parallel nutzbarer Devices	begrenzt durch Rechner Ressourcen	1
grafische Ausgaben mittels Views	ja - begrenzt durch Rechner Ressourcen	keine
Läuft auf Linux	nein	ja - mittels Mono Framework
Ressourcen Bedarf	hoch - je nach Project	gering
Debugeigenschaften	Oberfläche muss vorher starten	schnell und direkt

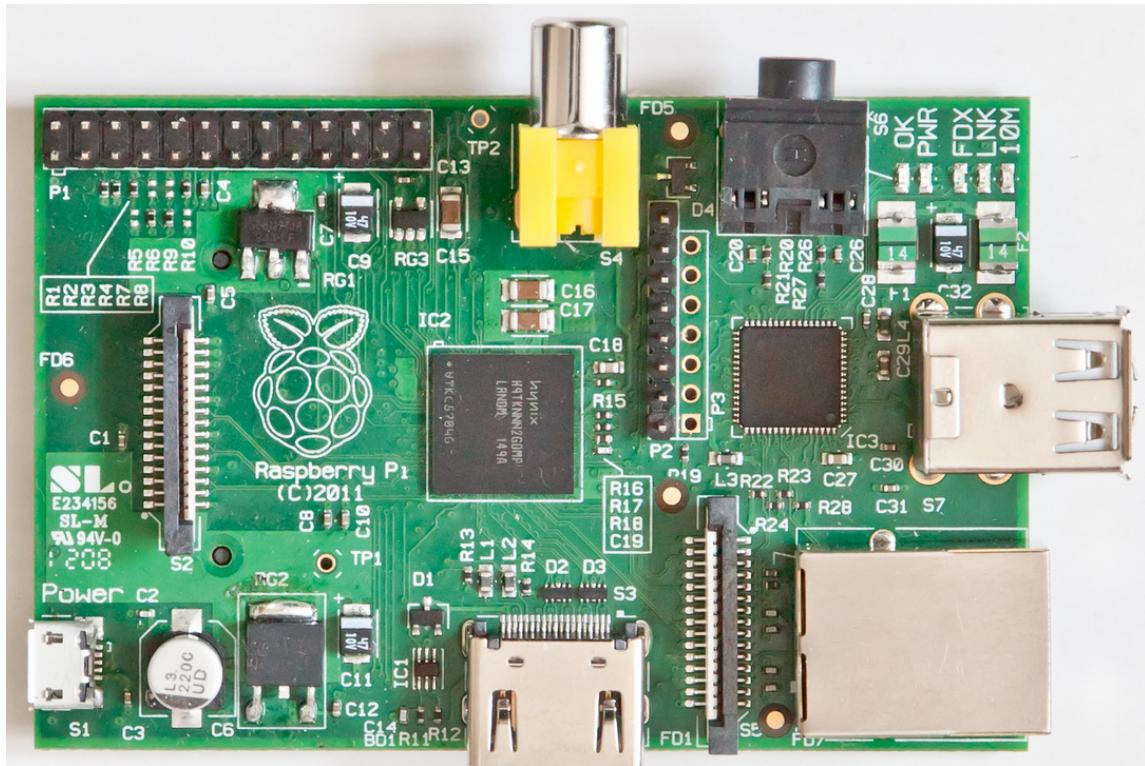
Grundlegend kann man sagen das beide Varianten ihr Vor- und Nachteile haben. Es kommt letztlich immer darauf an was man gerade mit LogView Studio erreichen möchte.

Für den "normalen User" wird aber sicherlich die GUI das Mass der Dinge sein und bleiben.

4.4.12 LogView Studio unter Linux

Das CLI (Command Line Interface) von LogView Studio lässt sich mit eher geringem Aufwand leicht auf Linux nutzen. Dafür ist allerdings das Mono Framework nötig (was unter Windows das .Net Framework ist). Wie schon im vorherigen Kapitel angedeutet kann unter Linux nur das CLI verwendet werden. Allerdings hat die CLI Version von LogView Studio einen eingebauten Webserver um die Daten visuell aufzubereiten.

Entstanden ist die Idee mit dem Erscheinen des Raspberry Pi Rechners.



Bei Tests hat sich schnell gezeigt das es durchaus möglich ist auf dieser eher bescheidenen Hardware ein LogView Studio zu betreiben. Durch die CLI Version von LogView Studio besteht nun die Möglichkeit das Project auf einem sehr stromsparenden System auch über längere Zeit laufen zu lassen.

Vor allem bei Langzeitaufzeichnungen eignet sich diese Variante.

Aber auch auf einem normalen Linux läuft LogView Studio ohne nennenswerte Probleme.

Datenauswertung mit Windows

An dieser Stelle sei noch zu erwähnen das die Datenauswertung auch mit einem Windowsrechner erfolgen kann. Die LogView Studio GUI kann sich (genauso wie das CLI) mit einem bestehenden Project lokal oder auch im Netzwerk verbinden um dann die Daten zu visualisieren.

Weitere Informationen

Raspberry Pi: <http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>

Linux: <http://www.linux.org/>

4.4.13 LogView Studio im Netzwerk

LogView Studio lässt sich sehr gut im Netzwerk einsetzen. Das bedeutet nichts anderes als das sich mehrere LogView Studio über ein vorhandenes Netzwerk auf ein Project verbinden. Dabei spielt es keine große Rolle ob mit Netzwerk nun das heimische LAN / WLAN gemeint ist oder aber auch das Internet.

Hinweis

Für die Internet Variante muss natürlich in den meisten Fällen der Router passend mit einer Portfreigabe versehen werden und im besten Fall hat man eine DynDNS Adresse. Diese Themen sprengen aber den Rahmen dieser Dokumentation - Google sollte aber genug Informationen liefern mit den Schlagwörtern.

*Portfreigabe : <http://de.wikipedia.org/wiki/Portweiterleitung>
DynDNS : <http://de.wikipedia.org/wiki/Dyndns>*

Genau genommen ist jedes LogView Project schon ein "LogView Studio im Netzwerk". Jedes Project hat seine eigene Datenbank und auf diese Datenbank greift LogView Studio über das Netzwerk zu. Nur muss der User bei einem einfachen Project keine weitere Konfiguration vornehmen.

Nutzen

Der Vorteil eines vernetzten LogView Studio liegt klar auf der Hand. Es können von verschiedenen Stellen Daten in ein Project geloggt werden. Oder es können von verschiedenen Stellen Daten aus einem Project ausgelesen werden.

Man kann die Vernetzung von LogView Studio mit einem Forum im Internet vergleichen. Es gibt einen Server im Internet der das Forum bereit stellt - das wäre in unserem Fall der Rechner mit dem Project wo die Daten gespeichert / gelesen werden sollen. Bei einem Forum können dann unterschiedliche Nutzer von beliebigen Orten auf das Forum zugreifen und Informationen lesen oder schreiben. In LogView Studio Fall wären das dann weitere Rechner die Daten ablegen oder Daten zur Auswertung abrufen.

Beispiele

Einige Beispiele sollten das Thema LogView Studio und Netzwerk verdeutlichen.

- "Kellerrechner"
Sie haben im Keller einen Rechner der gerade Daten aufzeichnet. Um gelegentlich den Fortschritt zu begutachten können Sie aus dem Wohnzimmer eine Verbindung zu dem "Keller Project" aufbauen. Alle Informationen sind nun auch über den Wohnzimmer Rechner abrufbar.
- Wetterstation
Sie möchten von zwei Standorten die Wetterdaten mit einem kleinen Arduino Projekt überwachen. Beide Stationen könnten (auch über das Internet) in einem Project die Daten ablegen.
- Linux mit mehreren Geräten
Unter Linux wird die LogView Studio CLI verwendet. Diese hat die Limitierung das nur ein Device genutzt werden kann. Startet man unter Linux aber eine weitere LogView Studio CLI Instanz kann diese sich auf das gleiche Project verbinden und die Daten dort ablegen.

Netzwerk Devices

Neben dem mehrfachen Verbinden von verschiedenen Rechnern auf ein gemeinsames Project bietet LogView Studio natürlich auch die Möglichkeit das verschiedene Devices über das Netzwerk Daten in ein gemeinsames project liefern können. Vor allem das OpenFormat bietet hier einfache Möglichkeiten um eine Hardware über das Netzwerk an LogView Studio anzubinden. Weitere Informationen zu diesem Thema finden sich im Kapitel [Development](#).

Internet der Dinge

Ein großer Hype im Internet ist derzeit das "Internet der Dinge" (http://de.wikipedia.org/wiki/Internet_der_Dinge). Vereinfacht gesagt geht es dabei darum das "Dinge" immer mehr miteinander vernetzt werden. Also z.B. eine Waschmaschine die Sich per Mail meldet das die Wäsche fertig ist.

Aus LogView Studio Sicht könnte man aber auch sagen das es eine große Anzahl von "Dingen" gibt die Daten liefern können. Und LogView Studio könnte selber als Datenlieferant im Internet auftreten.

Wer sich mit dem Thema "Internet der Dinge" etwas beschäftigt wird feststellen das LogView Studio auch hier gute Dienste leisten kann.

Weitere Informationen

[Netzwerk Beispiele](#)

4.5 Oberfläche

In diesem Kapitel wird das Menü mit seinen Funktionen und einige Grundlagen zur Oberfläche beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Tool Fenster](#)
- [MDI Fenster](#)
- [File Menü](#)
- [Project Menü](#)
- [Export Menü](#)
- [Analyse Menü](#)
- [Tools Menü](#)
- [View Menü](#)
- [Hilfe Menü](#)

4.5.1 Tool Fenster

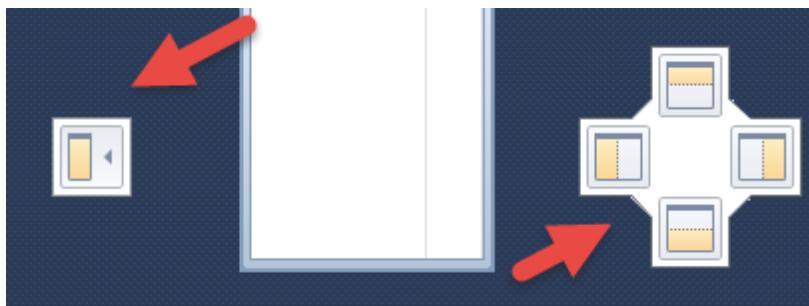
LogView Studio hat zahlreiche Tool Fenster wie z.B. den "Project Explorer". Diese Fenster lassen sich in der Oberfläche an die Seiten andocken, minimieren und auch komplett schließen.

Hinweis

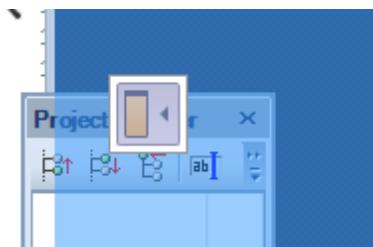
*Das Layout der Tool Fenster wird in der Datei LogViewStudio.lay gespeichert.
Diese Datei ist Bestandteil eines Project Ordners. Das Löschen dieser Datei setzt das Tool Fenster Layout wieder auf den Standard zurück.*

Docking

Um ein Fenster zu "docken" muss die Titelzeile mit der linken Maustaste angeklickt werden (Maustaste gedrückt halten). Es erscheinen daraufhin an den Rändern Laschen auf die das selektierte Fenster geschoben wird.

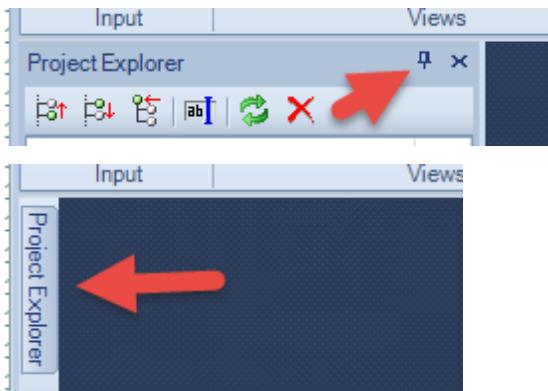


Die hellblaue Vorschau zeigt den neuen Bereich für das Fenster.



Minimieren

Über die "Stecknadel" in der Titelzeile lassen sich die Tool Fenster auf eine einfache Textlasche minimieren.



Wird die Lasche angeklickt erscheint das Tool Fenster wieder. Wird dann erneut die Stecknadel angeklickt bleibt das Fenster wieder in normaler Größe angezeigt.

Schließen

Ein Tool Fenster lässt sich auch komplett schließen über das 'x' in der Titelzeile.

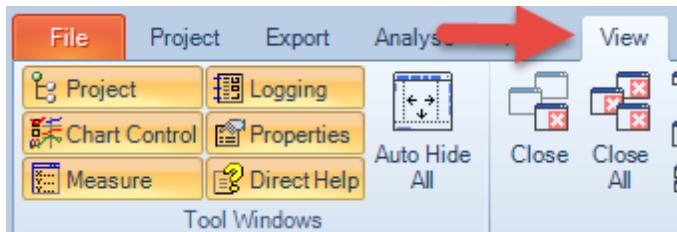


Hinweis

Geschlossene Fenster können über das Menü 'View' wieder angezeigt werden.

View Menü

Das View Menü bietet ebenfalls die Möglichkeit die Tool Fenster aus- bzw. einzublenden. Es können zudem alle Fenster Minimiert werden.



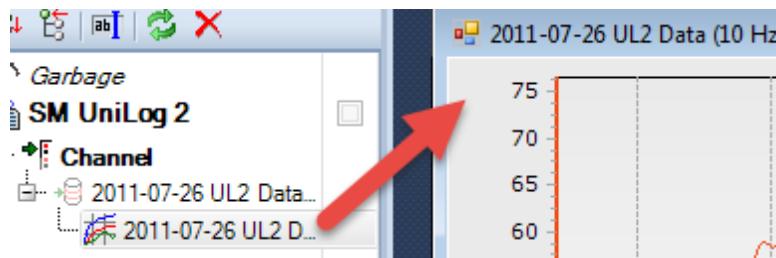
4.5.2 MDI Fenster

Für die Anzeige der Views nutzt LogView Studio MDI = Multiple Document Interface. Das bedeutet nur das in der Anwendung beliebig viele View Fenster geöffnet werden können.

Hinweis

Da tatsächliche maximale Anzahl der Views hängt stark von den verfügbaren Ressourcen des Rechners ab.

Jedes MDI Fenster in LogView Studio hat immer einen zugehörigen Eintrag im Project Explorer.



Wird ein MDI Fenster in der Oberfläche gewechselt, so wird auch der dazugehörige Project Eintrag selektiert.

Ein Doppelklick auf einen Project Explorer Eintrag öffnet immer das zugehörige Fenster (sofern vorhanden).

Minimieren

Ein Fenster lässt sich in LogView Studio über das normale System Icon minimieren.



Maximieren

Ein Fenster lässt sich in LogView Studio über das normale System Icon maximieren.



Hinweis

Die System Icons eines maximierten Fenster erscheinen daraufhin rechts neben dem Hauptmenü.

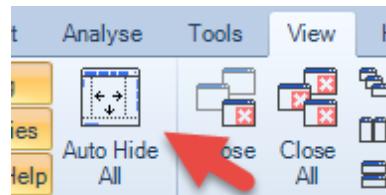
**Schließen**

Schließen lässt sich ein Fenster über das X Icon.

**Platz maximieren**

Wer für eine View besonders viel Platz benötigt kann dies in drei einfachen Schritten realisieren:

- Alle Tool Fenster minimieren



- Das Hauptmenü einklappen (rechts oben neben dem Hilfe Fragezeichen)



- View Maximieren

**4.5.3 File**

In diesem Kapitel wird das File Menü mit seinen Funktionen beschrieben.

Hinweis

Die File Funktionen sind sowohl im File Menü als auch im LogView Studio Wizard vorhanden. Die Funktionen sind immer identisch.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Neues Project anlegen](#)
- [Mit einem Remote Project verbinden](#)
- [Project öffnen](#)
- [Project speichern](#)

- [Project schließen](#)

4.5.3.1 New Project

Damit in LogView Studio Daten aufgezeichnet und ausgewertet werden können muss ein [Project](#) erstellt werden. Es gibt zwei Möglichkeiten ein neues Project zu erstellen:

- über den LogView Studio Wizard der nach dem Start angezeigt wird
- über das File Menü 'New > Project'

Die Vorgehensweise ist bei beiden Varianten identisch. Deswegen wird im Folgenden auch nur File Menü Variante erklärt:

- Im Menü 'File' auf New klicken oder in dem Submenü auf Project (alternativ im LogView Studio Wizard auf den Eintrag 'New Project')



- Im LogView Studio Wizard einen Project Titel eintragen und einen leeren Ordner im Dateisystem auswählen.

Hinweis

Neue Ordner können mit dem Button auf der rechten Seite erzeugt werden.

- Ist der Ordner aus LogView Studio Sicht verwendbar wird er mit einem grünen Punkt markiert.
- Im Wizard unten auf 'Finish' klicken und das Project wird erstellt
- Während der Erstellung des Project zeigt LogView Studio den Fortschritt in einem extra Fenster und im Logging an.

Project erstellt?

Das Projekt ist dann fertig erstellt wenn im Project Explorer der Garbage Eintrag vorhanden ist und der Projektname im Projekt Explorer Titel steht.



4.5.3.2 New Remote Project

LogView Studio ist in der Lage sich mit einem Project das auf einem anderen Rechner läuft zu verbinden. Es können sowohl die Daten des Remote Projects ausgelesen werden als auch vom eigenen Rechner Daten in dem Project abgelegt werden.

Hinweis

Das Arbeiten im Netzwerk mit LogView Studio ist in einem weiteren [Kapitel](#) noch

genauer beschrieben.

Damit das funktioniert muss LogView Studio auf dem lokalen / eigenen Rechner ein Project erzeugen. Das klingt zunächst etwas paradox wo doch auf dem Remote Rechner schon ein Project läuft. Der Grund dafür liegt darin das LogView Studio auf dem lokalen Rechner Daten wie z.B. Einstellungen oder Views speichern muss. Dafür wird auf dem lokalen Rechner zwingend eine eigen Project Struktur benötigt.

Hinweis

Der Unterschied zu einem normalen Project besteht darin das bei einem Remote Project keine Datenbank lokal gestartet wird.

Das Vorgehen ein neues Remote project anzulegen gleicht dem eines normalen Projects. Es müssen lediglich ein paar weitere Einstellungen vorgenommen werden.

- Im Menü 'File' auf New > remote Project klicken (alternativ im LogView Studio Wizard auf den Eintrag 'Remote Project')

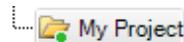


- Im LogView Studio Wizard einen Project Titel eintragen und einen leeren Ordner im Dateisystem auswählen.

Hinweis

Neue Ordner können mit dem Button auf der rechten Seite erzeugt werden.

- Ist der Ordner aus LogView Studio Sicht verwendbar wird er mit einem grünen Punkt markiert.



- Mit 'Next' die nächste Dialog Seite aufrufen
- Hier muss der Hostname bzw. die IP des Remote Rechners eingetragen werden. Port, User und Passwort sind im Normalfall korrekt vorbelegt und müssen nur im Sonderfall angepasst werden.

Hostname / IP-Adresse	Port
localhost	2804
User	Password
root	

- Ob die Datenbank erreicht werden kann lässt sich mit dem Button auf der rechten Seite testen.

```
Check the remote Connection

Try connection to remote LogView Studio project
(SERVER=localhost;PORT=2804;USERID=root;PASSWORD=;PROTOCOL=tcp;POOLING=false;)

Connecting ...
Connection Successful

Checking if there is a LogView Studio database
Database found

Perfect - You can connect to that Project
```



Ist die Datenbank nicht erreichbar wird ein Fehler ausgegeben

```
Check the remote Connection

Try connection to remote LogView Studio project
(SERVER=localhost;PORT=2804;USERID=rootw;PASSWORD=;PROTOCOL=tcp;POOLING=false;)

Connecting ...
Connection Successful

Checking if there is a LogView Studio database
Database missing

FAILED - This connection can't be used!
```

Hinweis

Damit der lokale Rechner auf den remote Rechner zugreifen kann muss im remote Project die IP des lokalen Rechners in der Datenbank hinterlegt / freigeschaltet werden. Infos dazu finden sich im [Netzwerk Kapitel](#).

- Im Wizard unten auf 'Finish' klicken und das Remote Project wird erstellt
- Während dem Erstellen des Projects zeigt LogView Studio den Fortschritt in einem extra Fenster und im Logging an.

Project erstellt?

Das Projekt ist dann fertig erstellt wenn im Project Explorer der Garbage Eintrag vorhanden ist und der Projektname im Projekt Explorer Titel steht. Weiterhin sollten vom Remote Rechner eingetragene Devices und Datensätze sichtbar sein.



4.5.3.3 Open Project

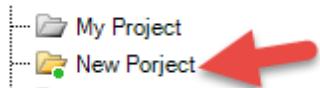
Ein Project kann in LogView Studio über zwei Wege geöffnet werden:

- Open Project ...
- Recent Project

Open Project ...

Im File Menü gibt es den Punkt 'Open Project ...'. Hier wird ein Fenster geöffnet in dem

eine Verzeichnisstruktur mit den verfügbaren Festplatten angezeigt wird. Wenn ein Ordner ein Projekt beinhaltet, ist der Ordner mit einem grünen Punkt markiert. Ansonsten ist der Ordner grau.



Wird ein Project Ordner ausgewählt kann unten im Dialog der 'Finish' Button angeklickt werden. Nun wird das Project geöffnet und der Fortschritt in einem Fenster angezeigt.

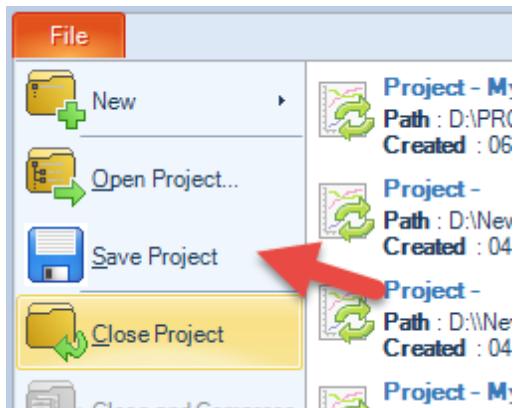
Recent Project

Die Recent Project Auswahl zeigt immer die zuletzt geöffneten Projects an (maximal 10). Hier kann ein Project ausgewählt und über 'Finish' geöffnet werden.

Alternativ wird das Project auch bei einem Doppelklick geöffnet.

4.5.3.4 Save Project

Ein geöffnetes Project kann im File Menü über 'Save Project' gespeichert werden.

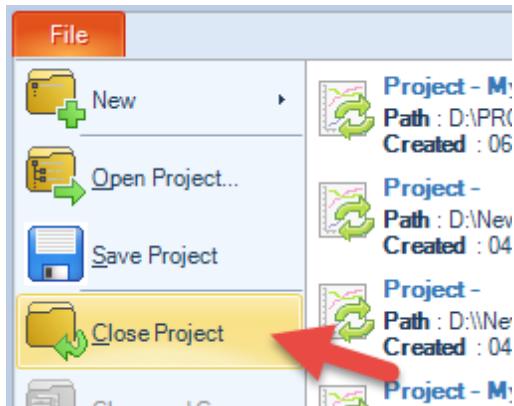


Hinweis

Beim Speichern werden vornehmlich die offenen Views und Settings gespeichert. Aufgezeichnete oder importierte Device Daten werden direkt in der Datenbank abgelegt.

4.5.3.5 Close Project

Ein geöffnetes Project kann im File Menü über 'Close Project' geschlossen werden.



Hinweis

Alle Geräte die Daten aufzeichnen und alle Views die Daten ausgeben werden vorher gestoppt.

4.5.4 Project

In diesem Kapitel wird das Project Menü mit seinen Funktionen beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Device](#)
- [Device Monitor](#)
- [Chart](#)
- [GPS 3D](#)
- [Table](#)
- [Circular Gauge](#)
- [Last Values](#)
- [Folder](#)
- [Delete](#)

4.5.4.1 Device

Um in LogView Studio Daten aufzuzeichnen oder zu importieren muss ein Device im Project eingefügt werden. Im Dialog 'Device and Source' wird dann das Device aus einer Liste ausgewählt und die Source konfiguriert.

Ein Device bildet in LogView Studio immer die Grundlage für alle weiteren Operationen.

Menü Gruppe

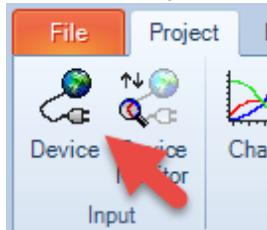
Input

Hierarchische Position

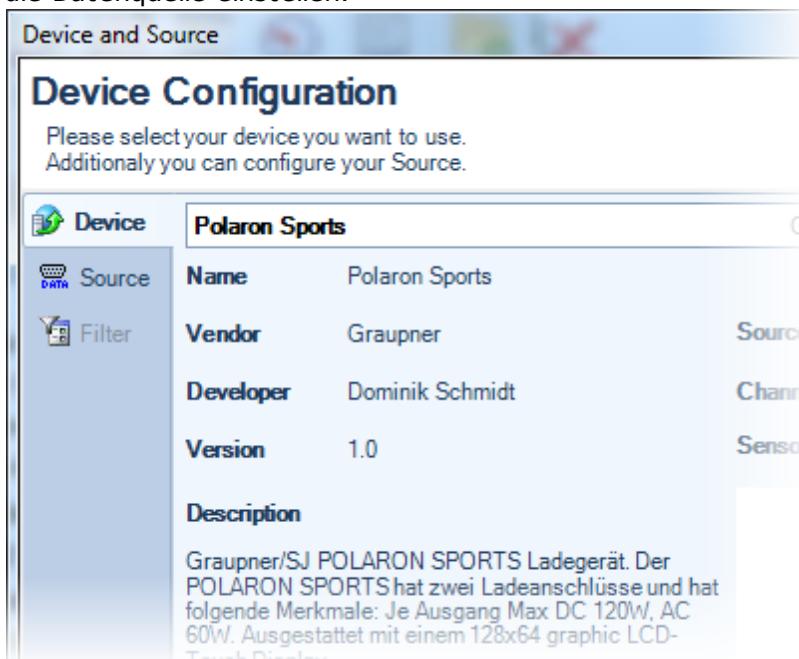
Hauptelement im Project

Vorgehensweise 'Device'

- Voraussetzung ist ein geladenes oder erstelltes Project
- Im Menü Project die Funktion 'Device' anklicken



- Im Dialog 'Device and Source' ein Device auswählen. Anschließend unter Source die Datenquelle einstellen.



- Das Device wird als Hauptelement im Project eingetragen mit seinen zugehörigen Channels.



Hinweis

Die Anzahl der Devices ist in LogView Studio nicht limitiert. Es können auch mehrere Devices gleichzeitig Daten aufzeichnen. Eine Begrenzung stellt hier eher die Anzahl verfügbarer Ports dar oder die Rechner Ressourcen.

Weitere Informationen

[Device Informationen](#)

4.5.4.2 Device Monitor

Der Device Monitor dient in LogView Studio dazu die ein- und ausgehenden Daten eines Devices zu monitoren. Des weiteren können auch Daten zum Device geschickt werden.

Die Funktion des Device Monitors ist eher für den ambitionierten LogView Studio User gedacht der sich z.B. mit dem OpenFormat und selbst entwickelter Hardware beschäftigt. Im "normalen" täglichen Einsatz wird diese Komponente keinen nennenswerten Vorteil bringen.

Menü Gruppe

Input

Hierarchische Position

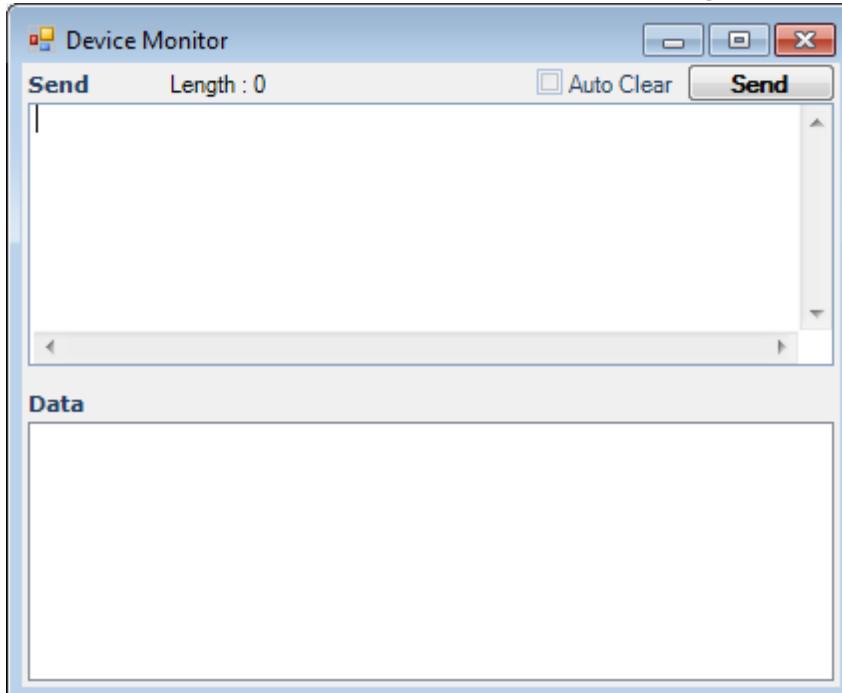
Subelement eines Devices

Vorgehensweise 'Device Monitor'

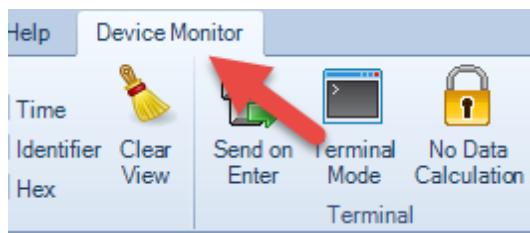
- Ein Device im Project auswählen.
- Im Menü Project die Funktion 'Device Monitor' anklicken



- Es erscheint ein neues Fenster 'Device Monitor' in LogView Studio.



- Ist dieses Fenster aktiviert wird zusätzlich ein neues Menü 'Device Monitor' eingeblendet.



Dieses Menü bietet zahlreiche Möglichkeiten die Daten zu protokollieren oder auch zu beeinflussen.

Weitere Informationen

[Arbeiten mit dem Device Monitor](#)

4.5.4.3 Chart

Die Funktion 'Chart' wird dazu verwendet um Daten in einem Diagramm als Kurven darzustellen. Es wird dabei zwischen statischen und live Charts unterschieden.

Menü Gruppe

Views

Hierarchische Position

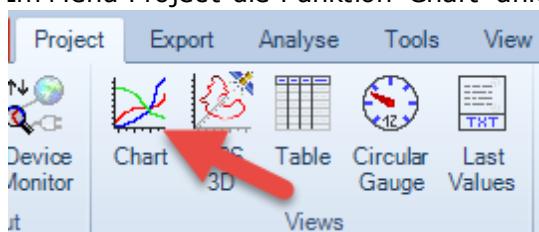
statischer Chart : Subelement eines Dataset

live Chart : Subelement eines Channels

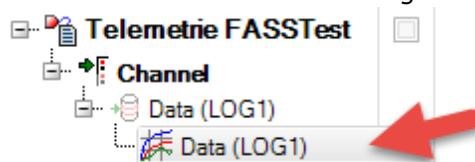
Vorgehensweise 'statischer Chart'

Ein statischer Chart zeigt immer einen abgeschlossenen Datensatz an. Nachdem die Daten einmal gelesen wurden kann der Chart nicht mehr geändert werden (bis auf das Erscheinungsbild).

- Einen Dataset im Project anklicken
- Im Menü Project die Funktion 'Chart' anklicken



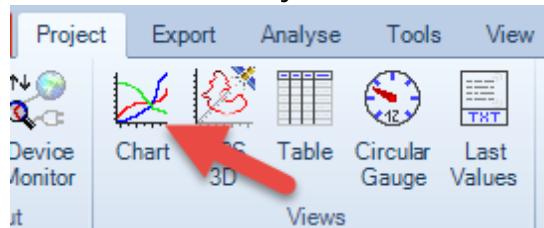
- Der statische Chart wird erzeugt als Subelement des Dataset



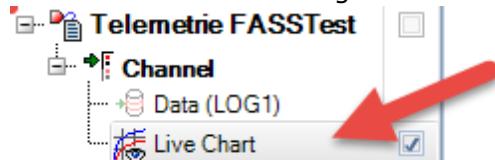
Vorgehensweise 'live Chart'

Ein live Chart zeigt immer die letzten Daten von einem Channel eines Devices die letzten X Daten an. Anzahl der Datenpunkte und die Update Geschwindigkeit können dabei konfiguriert werden.

- Einen Channel im Project anklicken



- Im Menü Project die Funktion 'Chart' anklicken
- Der live Chart wird erzeugt als Subelement des Channels



- Die live View wird über die Checkbox auf der rechten Seite im Project Explorer gestartet bzw. gestoppt.

Hinweis

Die Anzahl der Charts (statisch oder live) ist in LogView Studio nicht limitiert. Es können auch mehrere Charts für einen Dataset erzeugt werden.

Weitere Informationen

[Anpassen des Chart Layouts](#)

[Chart Templates speichern / laden](#)

[Static und Live View](#)

4.5.4.4 GPS 3D

Die Funktion 'GPS 3D' wird dazu verwendet um in den Daten vorhandene GPS Informationen in einem 3D Diagramm darzustellen. Die Funktion des GPS 3D View soll dabei weder Google Earth ersetzen noch bietet es auch nur ansatzweise den gleichen Funktionsumfang. Die Funktion ist nur dafür gedacht einen GPS Track schnell und einfach zu betrachten.

Eine GPS View zeigt immer einen abgeschlossenen Datensatz an. Nachdem die Daten einmal gelesen wurden kann der Chart nicht mehr geändert werden (bis auf das Erscheinungsbild).

In dem Datensatz müssen dafür allerdings GPS Daten vorhanden sein. Diese Daten werden im Normalfall vom Device automatisch markiert. LogView Studio erkennt diese Daten und wertet sie entsprechend aus. Sind keine GPS Daten vorhanden wird eine Meldung angezeigt.

Menü Gruppe

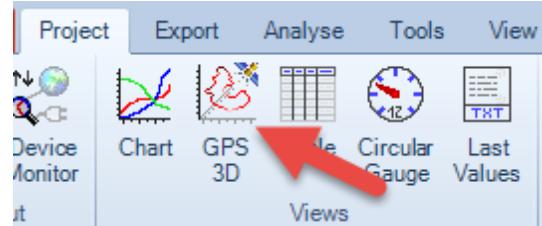
Views

Hierarchische Position

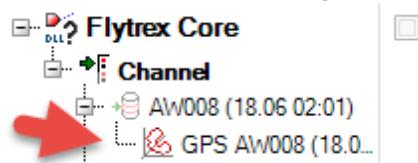
Subelement eines Dataset

Vorgehensweise 'GPS 3D'

- Einen Dataset im Project anklicken
- Im Menü Project die Funktion 'GPS 3D' anklicken



- Die GPS View wird erzeugt als Subelement des Dataset



Hinweis

Die Anzahl der GPS Views ist in LogView Studio nicht limitiert. Es können auch mehrere Views für einen Dataset erzeugt werden.

4.5.4.5 Table

Die View 'Table' wird genutzt um Datasets (Datensätze) als Tabelle darzustellen. Dabei werden die darzustellenden Daten einmalig aus der Datenbank gelesen und in der Tabelle eingetragen.

Eine Table View zeigt immer einen abgeschlossenen Datensatz an. Nachdem die Daten einmal gelesen wurden können die Daten nicht mehr geändert werden.

Hinweis

Mit der Table View können nicht die Daten der Datenbank geändert werden!

Menü Gruppe

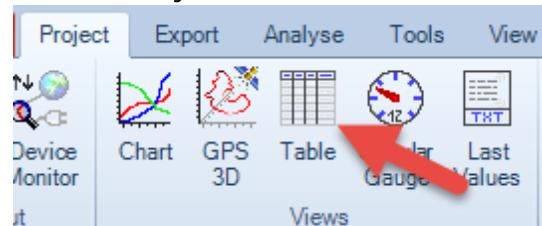
Views

Hierarchische Position

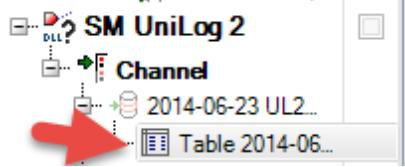
Subelement eines Dataset

Vorgehensweise 'Table'

- Einen Dataset im Project anklicken
- Im Menü Project die Funktion 'Table' anklicken



- Die Table View wird erzeugt als Subelement des Dataset



Hinweis

Die Anzahl der Table Views ist in LogView Studio nicht limitiert. Es können auch mehrere Views für einen Dataset erzeugt werden.

4.5.4.6 Circular Gauge

Die Live View 'Circular Gauge' wird genutzt um Channels als analog Anzeige (Gauge) darzustellen. Dabei werden die darzustellenden Daten zyklisch aus der Datenbank gelesen und im Gauge eingetragen.

Menü Gruppe

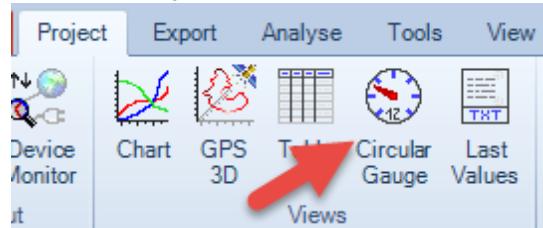
Views

Hierarchische Position

Subelement eines Channels

Vorgehensweise 'Circular Gauge'

- Einen Channel im Project anklicken
- Im Menü Project die Funktion 'Circular Gauge' anklicken



- Die Gauge View wird erzeugt als Subelement des Channels



- Die live View wird über die Checkbox auf der rechten Seite im Project Explorer gestartet bzw. gestoppt.

Hinweis

Die Anzahl der Gauge Views ist in LogView Studio nicht limitiert. Es können auch mehrere Views für einen Channel erzeugt werden.

4.5.4.7 Last Values

Die Live View 'Last Values' wird genutzt um die jeweils zuletzt eingetragenen Daten (jüngster Zeitstempel) eines Channels in einem Textfeld darzustellen. Dabei werden die

darzustellenden Daten zyklisch aus der Datenbank gelesen.

Menü Gruppe

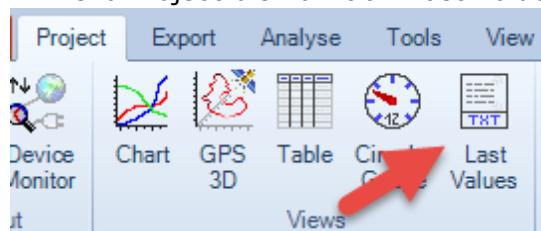
Views

Hierarchische Position

Subelement eines Channels

Vorgehensweise 'Last Values'

- Einen Channel im Project anklicken
- Im Menü Project die Funktion 'Last Values' anklicken



- Die Last Values View wird erzeugt als Subelement des Channels



- Die Live View wird über die Checkbox auf der rechten Seite im Project Explorer gestartet bzw. gestoppt.

Hinweis

Die Anzahl der Text Views ist in LogView Studio nicht limitiert. Es können auch mehrere Views für einen Channel erzeugt werden.

4.5.4.8 Folder

Ein Folder (Ordner) dient dazu ein Projekt zu strukturieren. Folder können untereinander verschachtelt werden. Datasets können in die Folder per Drag and Drop verschoben werden um so einen besseren Projektüberblick zu haben.

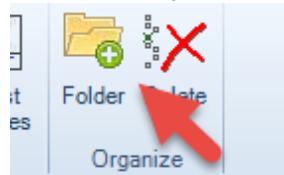
Wird ein Dataset in einen Folder verschoben werden alle Views des Dataset mit verschoben.

Menü Gruppe

Organize

Vorgehensweise

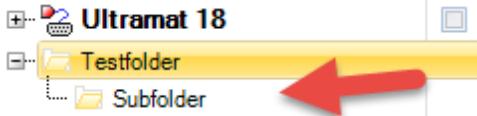
- Im Menü Project die Funktion 'Folder' anklicken



- Ist vorher kein Order selektiert worden wird ein neuer Root Ordner angelegt.



- Wird ein Ordner selektiert so erstellt LogView Studio einen Unterordner.



Hinweis

Die Folder Funktion ist auch im Kontextmenü des Project Explorers vorhanden.
Ordner können nicht innerhalb einer Device Struktur eingefügt werden.

4.5.4.9 Delete

Mit Hilfe der Delete Funktion können aus einem Project einzelne Elemente gelöscht werden.

Hinweis

Der Mülleimer Garbage und Channels lassen sich nicht im Project löschen.
Channels werden nur in Verbindung mit einem Device gelöscht.

Menü Gruppe

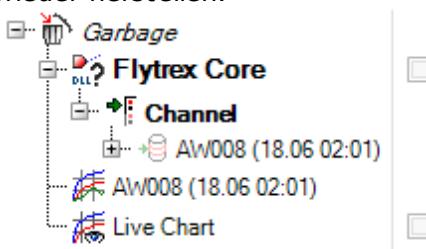
Organize

Vorgehensweise

- Ein Element im Project anklicken
- Im Menü Project die Funktion 'Delete' anklicken



- Das Element wird in den Garbage kopiert. Von hier kann es der User bei Bedarf wieder herstellen.



- Elemente im Garbage lassen sich durch erneutes 'Delete' endgültig löschen.

Hinweis

Die Delete Funktion ist auch im Kontextmenü des Project Explorers vorhanden.

Weitere Informationen

[Garbage - der Mülleimer](#)

4.5.5 Export

In diesem Kapitel wird das Export Menü mit seinen Funktionen beschrieben.

LogView Studio bietet ein breites Spektrum an Möglichkeiten um Daten als Grafik oder als Datenexport (z.B. CSV) zu exportieren.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Chart - Graphic](#)
- [Chart - CSV](#)
- [Dataset - CSV](#)
- [Dataset - XML](#)
- [Dataset - JSON](#)
- [Dataset - KML](#)
- [Application Screenshot](#)
- [Desktop Screenshot](#)

4.5.5.1 Chart - Graphic

Exportiert das selektierte Chart als Grafik. Es stehen dabei mehrere Grafik Exportformate zur Auswahl

Menü Gruppe

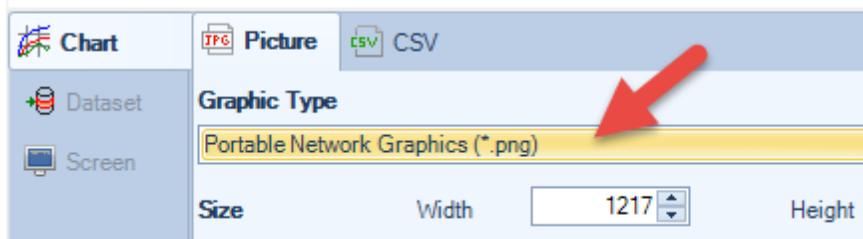
Chart

Vorgehensweise

- Ein Chart Element im Project auswählen
- Im Menü Export auf 'Graphic' klicken
- Das passende Exportformat auswählen:

Data Export

Export your Charts or Datasets as Image or Datafile.
Use Copy for the Clipboard and Save to store a file.



- Den Grafik Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.2 Chart - CSV

Exportiert das selektierte Chart als CSV (Comma-separated values).

Menü Gruppe

Chart

Vorgehensweise

- Ein Chart Element im Project auswählen
- Im Menü Export auf 'CSV' klicken (Gruppe Chart)
- Der Export Dialog bietet noch einige Einstellungen um den Export zu beeinflussen
- Den Daten Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.3 Dataset - CSV

Exportiert den selektierten Datensatz als CSV (Comma-separated values).

Menü Gruppe

Dataset

Vorgehensweise

- Ein Dataset Element im Project auswählen
- Im Menü Export auf 'CSV' klicken (Gruppe Dataset)
- Der Export Dialog bietet noch einige Einstellungen um den Export zu beeinflussen
- Den Daten Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.4 Dataset - XML

Exportiert den selektierten Datensatz als XML (Extensible Markup Language).

Menü Gruppe

Dataset

Vorgehensweise

- Ein Dataset Element im Project auswählen
- Im Menü Export auf 'XML' klicken (Gruppe Dataset)
- Der Export Dialog bietet noch einige Einstellungen um den Export zu beeinflussen
- Den Daten Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.5 Dataset - JSON

Exportiert den selektierten Datensatz als JSON (JavaScript Object Notation).

Menü Gruppe

Dataset

Vorgehensweise

- Ein Dataset Element im Project auswählen
- Im Menü Export auf 'JSON' klicken (Gruppe Dataset)
- Der Export Dialog bietet noch einige Einstellungen um den Export zu beeinflussen
- Den Daten Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.6 Dataset - KML

Exportiert den selektierten Datensatz als KML (Keyhole Markup Language).

Hinweis

Dieser Export Typ ist nur für Datensätze geeignet die auch GPS Daten beinhalten.

Menü Gruppe

Dataset

Vorgehensweise

- Ein Dataset Element im Project auswählen
- Im Menü Export auf 'KML' klicken (Gruppe Dataset)
- Der Export Dialog bietet noch einige Einstellungen um den Export zu beeinflussen
- Den Daten Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.7 Application Screenshot

Exportiert das LogView Studio Fenster als PNG Grafik.

Menü Gruppe

Special

Vorgehensweise

- Im Menü Export auf 'Application Screenshot' klicken
- Den Grafik Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.5.8 Desktop Screenshot

Exportiert den kompletten Desktop als PNG Grafik.

Menü Gruppe

Special

Vorgehensweise

- Im Menü Export auf 'Desktop Screenshot' klicken
- Den Grafik Export mit 'Copy' in die Zwischenablage sichern, oder mit 'Save' als Datei speichern

4.5.6 Analyse

In diesem Kapitel wird das Analyse Menü mit seinen Funktionen beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Min Max Average](#)
- [Average Filtered](#)

4.5.6.1 Min Max Average

Berechnet für einen Datensatz folgende Werte:

- Startzeit
- Endzeit
- Zeitspanne
- Minimum (je Sensor)
- Maximum (je Sensor)
- Durchschnitt (je Sensor)

Menü Gruppe

Dataset

Vorgehensweise

- Ein Dataset Element im Project auswählen
- Im Menü Analyse auf 'Min Max Average' klicken
- Es wird folgende Ausgabe erzeugt:

```
Analyse Min Max Avg
Minimum / Maximum / Average Analysis

Start Date : 21.08.2014 20:06:01
End Date   : 21.08.2014 20:26:47
Difference  : 00:20:45.8000000

Value Count : 6230

Höhe
Minimum : -0,033
Maximum : 203,384
Average : 96,289

Temperatur
Minimum : 22,0
Maximum : 24,0
Average : 23,232

Steig/Sinkrate
Minimum : -3,0
Maximum : 8,4
Average : -0,043
```

Hinweis

Die Daten werden nicht im Project Explorer eingehängt.

4.5.6.2 Average Filtered

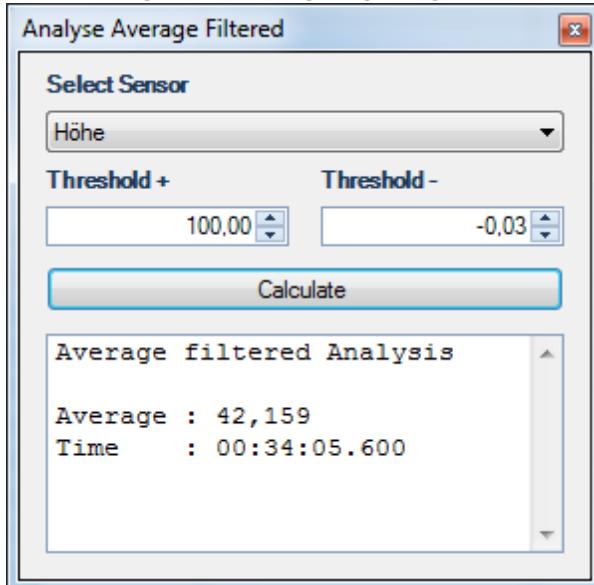
Berechnet für einen Sensor den Durchschnittswert. Dabei kann eingestellt werden welche Werte in die Berechnung einbezogen werden (mittels Threshold).

Menü Gruppe

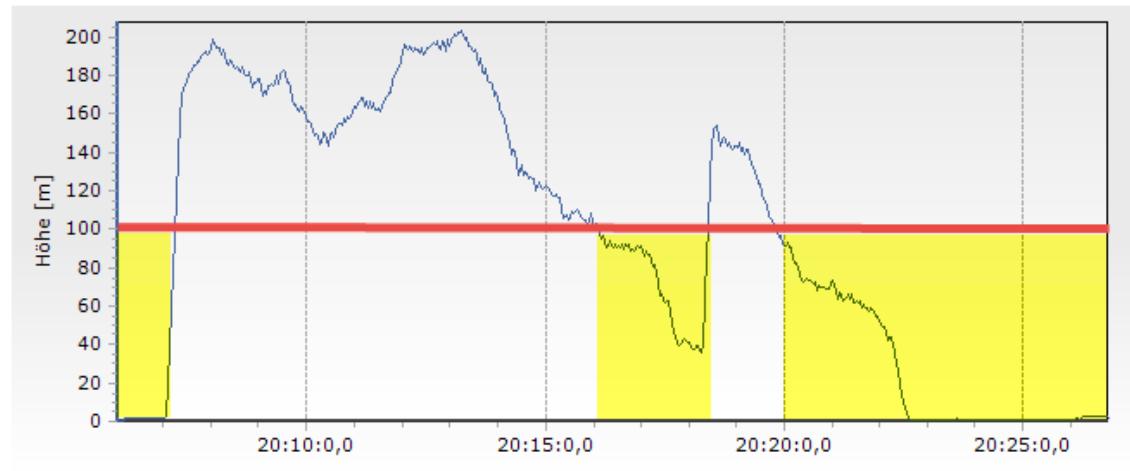
Dataset

Vorgehensweise

- Ein Dataset Element im Project auswählen
- Im Menü Analyse auf 'Average Filtered' klicken
- Es wird folgender Dialog angezeigt:



- Mit Threshold +/- kann der Bereich eingestellt werden der bei der Berechnung eingeschlossen werden soll. In dem Beispiel werden alle Werte zwischen 100 und -0,03 berücksichtigt (gelber Bereich der Kurve):

**Hinweis**

Die Daten werden nicht im Project Explorer eingehängt.

4.5.7 Tools

In diesem Kapitel wird das Tools Menü mit seinen Funktionen beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Change Start Time](#)
- [Open Settings Folder](#)
- [Add Host to Database](#)

4.5.7.1 Change Start Time

Aufgezeichnete Daten können aus unterschiedlichen Gründen die falsche Start- / Endzeit haben. Dies kann u.a. vorkommen wenn das Gerät keine eindeutigen Zeitstempel liefert und LogView Studio die Systemzeit als Referenz nutzt. Geräte mit GPS Informationen nutzen auch häufig UTC (GMT) Zeitstempel und nicht die lokale Ortszeit.

In einem solchen Fall kann das Tool 'Change Start Time' die Zeit nach eigenen Vorgaben anpassen.

Das Tool kann sowohl auf Datasets als auch auf Charts angewendet werden.

Hinweis

Wird das Tool auf einen Dataset angewendet werden die Daten in der Datenbank geändert. Es werden also die Originaldaten manipuliert!

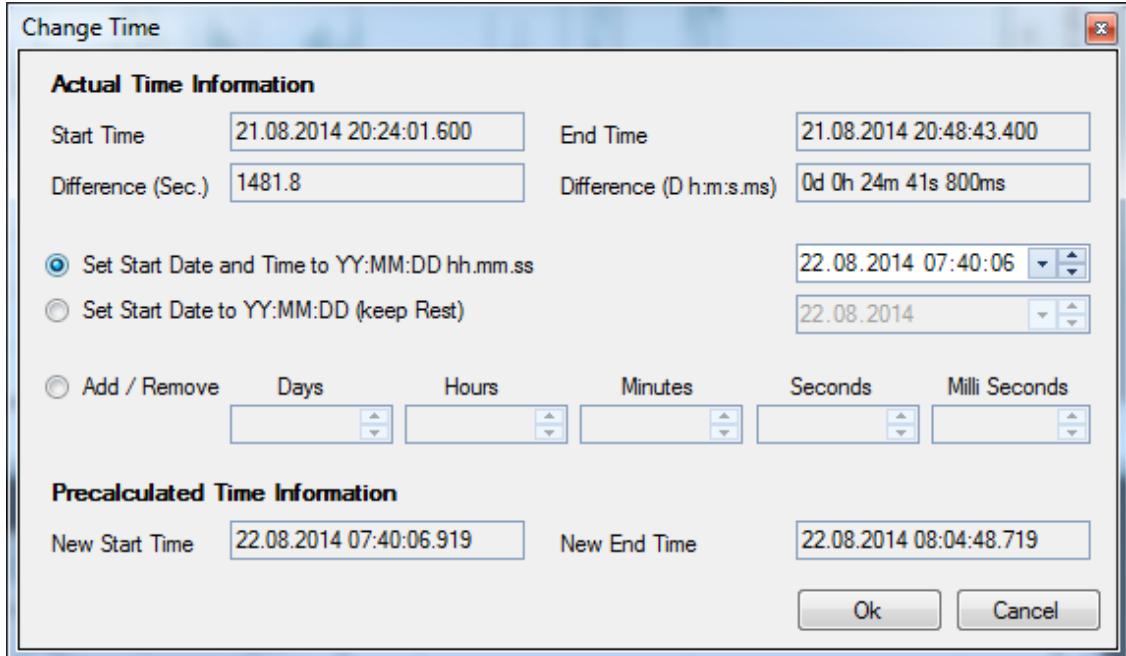
Menü Gruppe

Data

Vorgehensweise

- Ein Dataset oder ein Chart Element im Project auswählen

- Im Menü Tools auf 'Change Start Time' klicken
- Der Dialog bietet eine Reihe von Möglichkeiten die Startzeit anzupassen:



Im oberen Bereich des Dialogs werden immer die ursprünglichen Zeitinformationen angezeigt. Im unteren Bereich werden die neu berechneten Zeiten dargestellt.

- Mit 'Ok' werden die neuen Einstellungen übernommen. Bei einem Datensatz werden die Daten in der Datenbank geändert. Bei einem Chart wird lediglich die Zeitachse angepasst.

4.5.7.2 Open Settings Folder

LogView Studio muss einige wenige Daten im Benutzer Verzeichnis ablegen. Dort werden z.B. die letzten genutzten Projekte gespeichert oder die Error Logs gesammelt.

Mit der Funktion 'Open Settings Folder' kann dieser Ordner direkt geöffnet werden.

Menü Gruppe

Special

4.5.7.3 Add Host to Database

Die Funktion 'Add Host to DB' wird benötigt um weiteren Rechnern den Zugriff auf das aktuelle Project zu erlauben.

Die Vorgehensweise ist im [Netzwerk Beispiel](#) beschrieben.

Menü Gruppe

Database

4.5.8 View

In diesem Kapitel wird das View Menü mit seinen Funktionen beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Tool Windows](#)
- [Windows](#)

4.5.8.1 Tool Windows

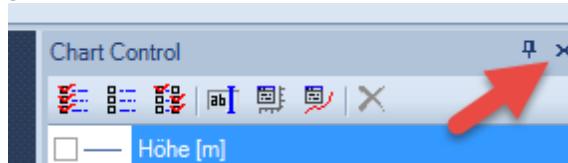
LogView Studio bietet einige spezielle Tool Windows wie z.B. den Project Explorer oder das Chart Control. Mit den Buttons in diesem Menü können die Fenster ein- bzw. ausgeschaltet werden.

Menü Gruppe

Tool Windows

Vorgehensweise

- Im Bereich Tool Windows auf einen Button klicken um das entsprechende Tool Window zu öffnen oder zu schließen.
- Alternativ können die Tool Windows auch mittels dem kleinen 'x' rechts oben geschlossen werden:



- Die Tool Windows können zudem minimiert werden über die 'Stecknadel'. Die Fenster werden danach nur noch als Lasche am Fensterrand angezeigt:



- Das Menü bietet weiterhin die Möglichkeit alle Tool Windows zu minimieren (Auto Hide All).

4.5.8.2 Windows

Für die "normalen" Views wie z.B. ein Chart oder ein Gauge bietet LogView Studio die normalen Funktionen um Fenster zu verwalten.

Die Funktionen in diesem Menübereich sollten sich von alleine erklären.

Menü Gruppe

Windows

4.5.9 Help

Das Help Menü bietet Zugriff auf die wichtigsten Hilfe Themen von LogView Studio.

4.6 Arbeiten mit LogView Studio

In diesem Kapitel werden die oft genutzten Arbeitsabläufe mit LogView Studio beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Informationen zu Devices](#)
- [Device Monitor benutzen](#)
- [Charts anpassen und benutzen](#)
- [Beispiel](#)

4.6.1 Device

In diesem Kapitel wird der Umgang mit Devices genauer beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Device Starten und Stoppen](#)
- [Source ändern](#)
- [Device mit Dateiimport](#)
- [Autodetect Serial- und HID Ports](#)
- [Zero Sensor Devices](#)

4.6.1.1 Device Starten und Stoppen

Ein Device hat im Projekt Explorer rechts neben dem Namen eine Checkbox:



Ist diese Checkbox aktiviert wird das Device gestartet. Das bedeutet die Daten der Source werden empfangen, ausgewertet und in der Datenbank eingetragen.

Das Device wird gestoppt wenn der Haken wieder entfernt wird:



Hinweis

Bei einer File oder Binary Source ist die Checkbox ausgegraut. Die Daten werden sofort importiert beim Schließen des Source Dialogs.

Logging

Das Starten und Stoppen wird auch im Logging protokolliert:

```
Mo 18:21:02.232 : [ 9, Serial      ] Open Port, Start IO Thread (Port : COM1)
Mo 18:21:02.251 : [ 9, Serial      ] GetSetting : COM1||\\?\acpi#pnp0501#1#\{..\} ||
9600||8||One||None||False||False
Mo 18:21:53.067 : [ 9, Device     ] Stopping Source
Mo 18:21:53.068 : [ 9, Serial      ] Stop IO Thread, Close Port
Mo 18:21:53.068 : [ 9, Device     ] Stopping Calc Thread
Mo 18:21:53.068 : [ 9, Device     ] Closing Database
Mo 18:21:53.081 : [37, Serial      ] Serial IO Thread stopped.
```

Bei Problemen sollten dort Informationen bezüglich des Fehlers zu finden sein.

Weitere Informationen

[Was ist ein Device?](#)

[Was ist eine Source?](#)

[Device hinzufügen](#)

[Beispiele](#)

4.6.1.2 Source Konfiguration

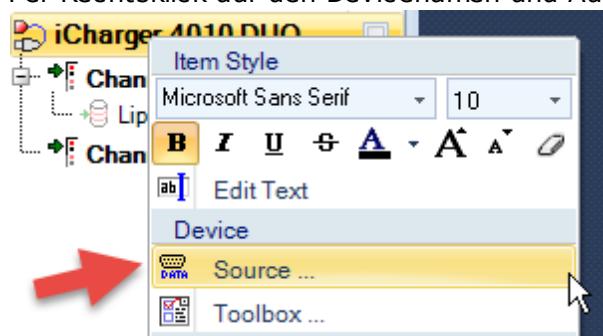
Zu jedem Device gehört in LogView Studio eine Source. Die Source ist die Schnittstelle über welche die Daten in den Rechner gelangen (also z.B. ein serieller Port).

Im Normalfall wird die Source direkt beim [Anlegen eines Devices](#) mit angegeben und danach genutzt. Die Source kann aber auch später ausgewählt und auch geändert werden.

Source ändern

Um die Source eines Devices anzupassen gibt es verschiedene Wege:

- Per Doppelklick auf den Devicenamen im Project Explorer
- Per Rechtsklick auf den Devicenamen und Auswahl von 'Source ...'



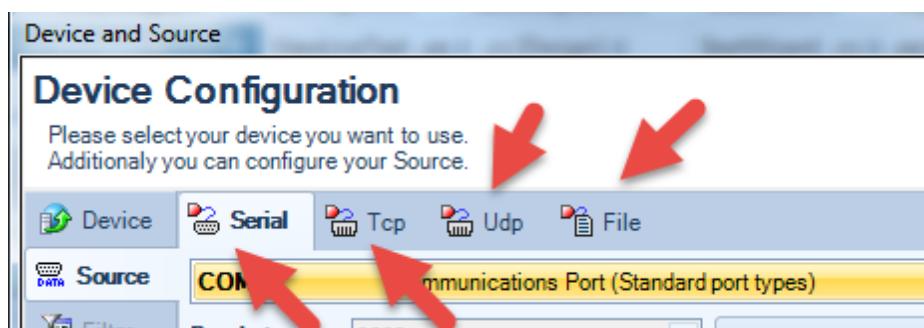
- über das Command Line Tool

Abgesehen vom Command Line Tool wird der Source Dialog zur Auswahl angezeigt.



Devices mit mehreren Sources

Ein Device kann durchaus mehrere Sources haben. So kann z.B. das OpenFormat Daten aus verschiedenen Quellen entgegen nehmen. In dem Fall muss die richtige Source über den entsprechenden Reiter ausgewählt und eingestellt werden.



Nachdem Device und Source eingestellt sind wird der Dialog mit OK bestätigt.

Hinweis

Es kann immer nur eine Source aktiv sein.

Weitere Informationen

[Was ist ein Device?](#)

[Was ist eine Source?](#)

[LogView Studio Command Line Tool](#)

4.6.1.3 Device mit Dateiimport

Im Vergleich zu den anderen Sources nehmen die [File und Binary Source](#) eine gewisse Sonderstellung ein. Bei einer Datei stehen alle Daten sofort zur Verfügung. Aus diesem Grund werden Dateien in LogView Studio direkt eingelesen und verarbeitet wenn der User den Device Dialog mit OK bestätigt.

Die Möglichkeit das Device zu starten bzw. zu stoppen ist bei einer File / Binary Source nicht möglich.

Hinweis

Wenn der User den Device Dialog erneut öffnet und wieder mit OK bestätigt wird die Datei erneut eingelesen und verarbeitet!

Einen Massenimport für mehrere Dateien ist derzeit noch nicht vorhanden.

Import mehrere Logs in ein Project

Gerade bei einem Device mit File oder Binary Source kann es für einen Benutzer verwirrend sein wie er mehrere Logs importieren kann. Der richtige Weg wäre folgender:

- Einmal ein [neues Device anlegen](#)
- Das erste Log gleich bei der File/Binary Source angeben
- Mit OK den Dialog bestätigen und das erste Log importieren lassen
- Danach mit einem Doppelklick auf das Device den Source Dialog neu öffnen (Alternativ auch Rechtsklick auf das Device und Source Configuration anklicken).
- Jetzt in der Source das neue Log File auswählen und mit OK den Dialog bestätigen (Kein neues Device anlegen)
- Die letzten beiden Schritte für alle anstehenden Logs wiederholen.

4.6.1.4 Autodetect Serial / HID

Sowohl die Serial als auch die HID Source bieten die Möglichkeit eines Autodetect für den verwendeten Port.

Der Autodetect funktioniert natürlich nur mit Ports die nachträglich am Rechner angeschlossen werden. Also vornehmlich alles was mit USB angeschlossen wird.

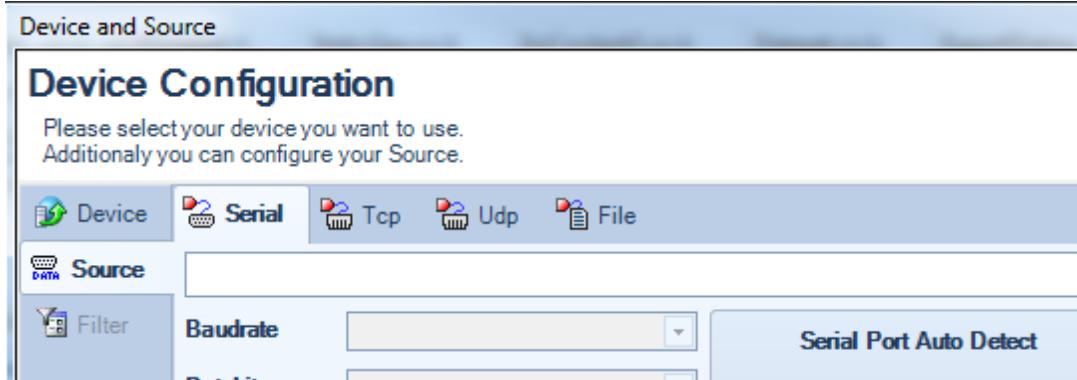
Hinweis

Autodetect ist nicht dazu gedacht ein Gerät zu erkennen. Es geht nur darum den richtigen Port für die Source zu bestimmen.

Vorgehensweise

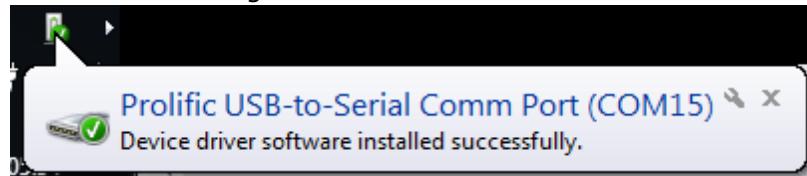
Das folgende Beispiel zeigt den Autodetect mit einem USB / Seriell Adapter. Mit einer USB HID Source ist das Vorgehen gleich.

- Das Device darf noch nicht am Rechner angeschlossen sein!
- Der Dialog für die Source des Device öffnen

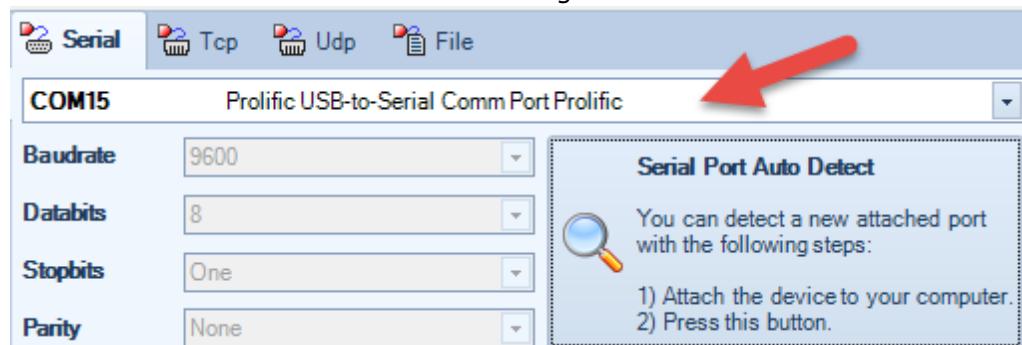


- Das Gerät an den PC anschließen und warten bis Windows das Gerät erkannt hat. Hier kann es mitunter etwas dauern wenn erst der Treiber installiert werden muss. Wie schon zuvor erwähnt funktioniert diese Erkennung nur wenn das Gerät über

einen USB Port angeschlossen wird.



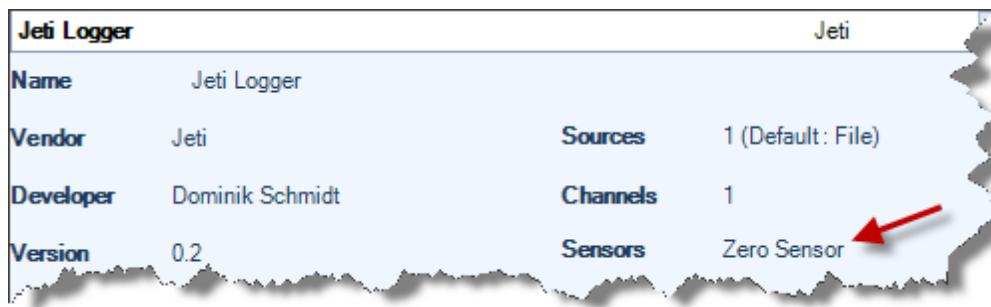
- Jetzt den Button "Serial Port Auto Detect" drücken. Bei HID entsprechend auf "USB HID Port Auto Detect".
- Der neue Port sollte nun automatisch ausgewählt sein



4.6.1.5 Zero Sensor Devices

LogView Studio kann Devices beinhalten die keine vorkonfigurierten Sensoren (= auswertbare Messwerte) haben. Diese Devices können ihre Sensoren zur Laufzeit z.B. anhand der auszuwertenden Logdateien bestimmen und die Datenbank dementsprechend erweitern.

Erkennen kann man solche Devices im Device Dialog:



Im Normalfall gibt es mit solchen Devices keine Probleme. Die Daten werden eingelesen, die Datenbank angepasst und die Daten werden ausgewertet.

Problem : Unterschiedliche Logfiles

Wenn nun aber Daten importiert werden die bei den Messungen unterschiedliche Sensoren genutzt haben kommt es mitunter zu merkwürdigen Auswertungen.

Beispiel

Ein User hat eine Logdatei A mit Spannung und Strom und eine Logdatei B mit Geschwindigkeit.

Er legt das entsprechende Gerät an und importiert Datei A. Die Datenbank wird nun um die beiden Sensoren Spannung und Strom für das Device erweitert. Wenn er jetzt zu

den importierten Daten einen Chart öffnet sieht er nur die beiden Kurven.

Nun importiert der User mit dem gleichen Device noch Logdatei B. Das Gerät wird nun zusätzlich noch um den Sensor Geschwindigkeit erweitert. Das stellt an sich kein Problem dar und funktioniert auch.

Wenn der User aber nun einen Chart anlegt für den zweiten Datensatz (von Logdatei B), dann hat dieser Chart 3 Kurven. Spannung, Strom und Geschwindigkeit. Die ersten beiden Kurven enthalten aber keinerlei Daten.

Beste Vorgehensweise

Wenn sich die Sensoren (Messdaten) bei den eingelesenen Daten unterscheiden sollte man das Gerät erneut im Projekt anlegen und die Daten mit dem neuen Device importieren. Dort werden dann nur die nötigen Sensoren eingetragen.

4.6.2 Device Monitor

In diesem Kapitel wird der Umgang mit dem Device Monitor anhand eines Beispiels beschrieben.

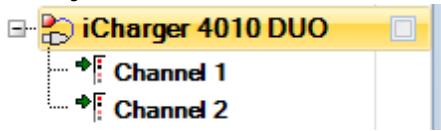
Hinweis

Der Device Monitor ist bei der normalen Benutzung von LogView Studio nicht notwendig. Der Monitor wird erst interessant wenn man die Daten analysieren möchte die vom Gerät zum PC geschickt werden oder umgekehrt. Der Device Monitor ist also in erster Linie für Entwickler interessant.

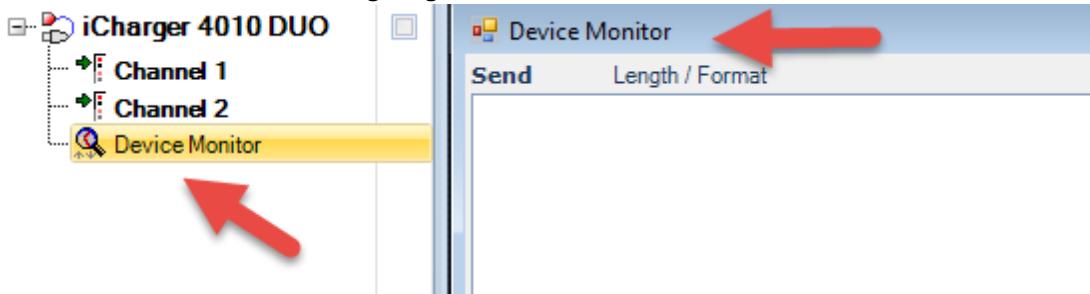
Vorgehensweise

Das folgende Beispiel zeigt die Verwendung des Device Monitors anhand des Junsi Laders 4010 DUO mit HID Interface.

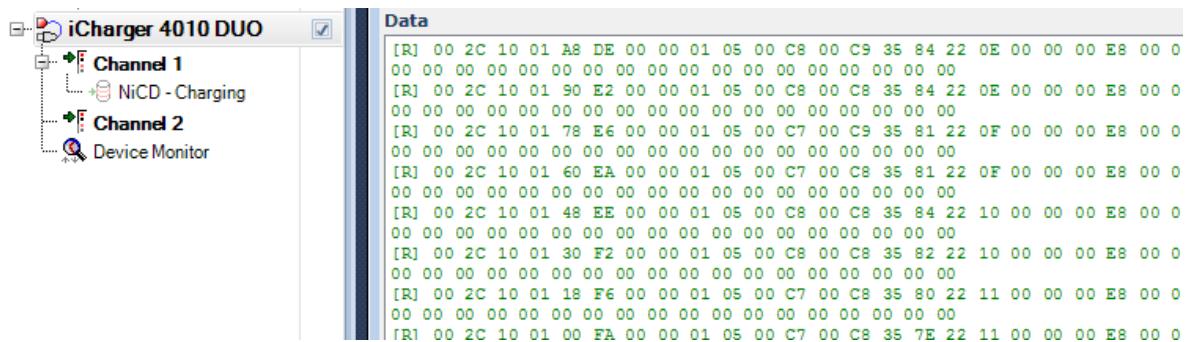
- Im Projekt muss das Device vorhanden sein welches überwacht werden soll:



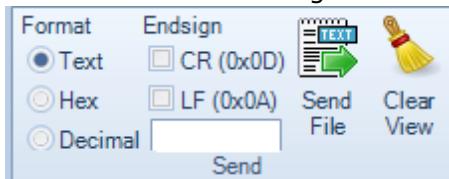
- Das Device selektieren im Project Explorer und im Menü Project auf 'Device Monitor' klicken. Es wird ein neues Fenster eingeblendet und im Project Explorer wird ein 'Device Monitor' Element eingefügt:



- Jetzt das Device starten. Wenn Daten empfangen werden wird das Log erweitert:



Der Device Monitor kann ebenfalls dazu genutzt werden an das Device Daten zu senden. Dazu muss im Menü eingestellt werden wie die Daten eingegeben wurden:



In diesem Fall muss im Send Bereich ein Text stehen. Dieser wird dann an das Device übertragen.

No Data Calculation

Das Menü 'Device Monitor' hat einen speziellen Eintrag 'No Data Calculation'. Wenn diese Funktion aktiviert wird werden die Daten von LogView Studio nicht berechnet und gespeichert. Die Daten werden also nur im Monitor angezeigt und anschließend verworfen.

Die Funktion ist sehr hilfreich wenn man mit dem Gerät Tests durchführt und die Auswertung erst mal eine untergeordnete Rolle spielt.

4.6.3 Chart

In diesem Kapitel wird der Umgang mit Charts genauer beschrieben

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [Zoom und Pan](#)
- [Grafik layouten](#)
- [Templates](#)
- [Messen](#)

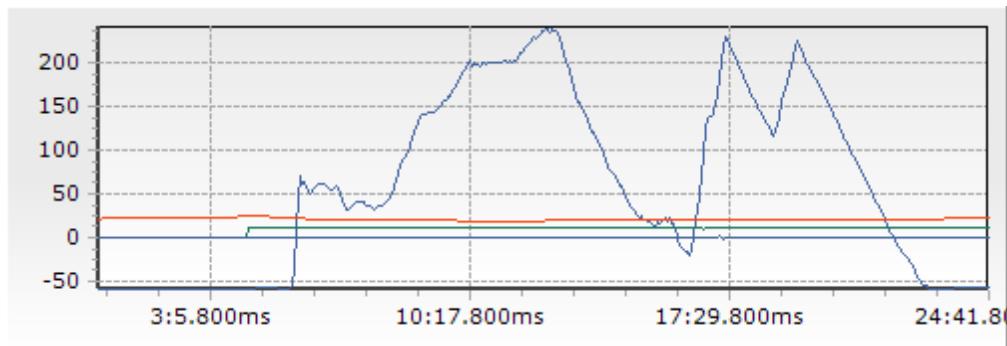
4.6.3.1 Zoom und Pan

Im folgenden sind die Zoom- und Panning (Verschieben) Optionen der Grafik (Chart) aufgeführt:

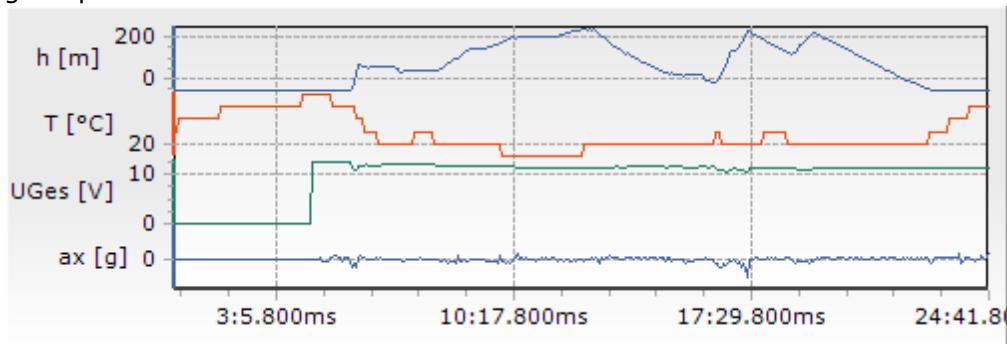
Zoom

left mousekey -> rectangle (from top/left Zoom rectangle
to bottom/right)

left mousekey -> rectangle (from UNDO Zoom



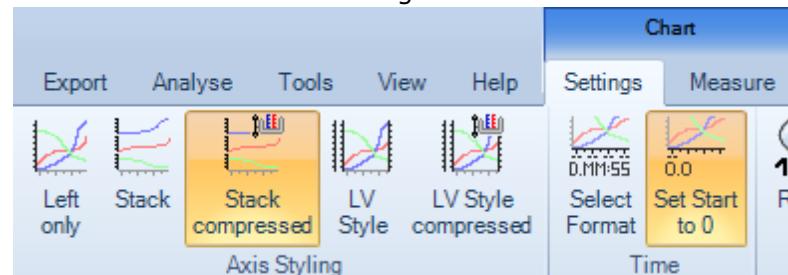
- **Stack**
Jede Kurve hat eine eigene Achse auf der linken Seite. Alle Kurven sind übereinander gestapelt.



- **LV Style**
Jede Kurve hat eine eigene Achse die entweder links oder rechts dargestellt wird.



Die Auswahl der Ansicht erfolgt über das Menü 'Chart -> Settings':



Hinweis

Die 'compressed' Varianten kombinieren gleiche Kurven auf eine Achse. So lassen

sich z.B. Zellspannungen auf einer Achse zusammenlegen.

Kontext Menü

Jede Kurve wird bei einem Chart im 'Chart Control' eingetragen. Über einen Rechtsklick auf eine Kurve bekommt man ein Kontext Menü für Anpassungen für die selektierte Kurve:



Das Menü bietet eine Fülle an Möglichkeiten die Kurve optisch aufzubereiten. Zusätzlich kann dort auch die Achse im Detail eingestellt werden (Endwerte, Offsets, ...).

4.6.3.3 Templates

Das Aussehen und Styling von Charts kann in LogView Studio mittels Templates gespeichert werden. Diese Templates können danach genutzt werden um Charts mit dem gleichen Style zu belegen.

Im Menu 'Chart' -> 'Settings' ist die Sektion 'Tempates' mit den Funktionen 'Load', 'Save' und 'Delete'. Diese Funktionen sind jeweils auf das aktive Chart bezogen.

Template speichern

Um ein Template zu speichern muss ein chart selektiert werden. Im 'Chart' Menü befindet sich die Funktion Templates Save.



Wird die Funktion ausgeführt kann ein Template Name vergeben werden. Nach der Bestätigung mit OK wird das Template gespeichert.

Hinweis

Die Templates werden im Userverzeichnis gesichert. Bei Windows 7 ist das z.B. C:\Users\<USERNAME>\AppData\Local\LogView Studio\ChartTemplate

Hinweis

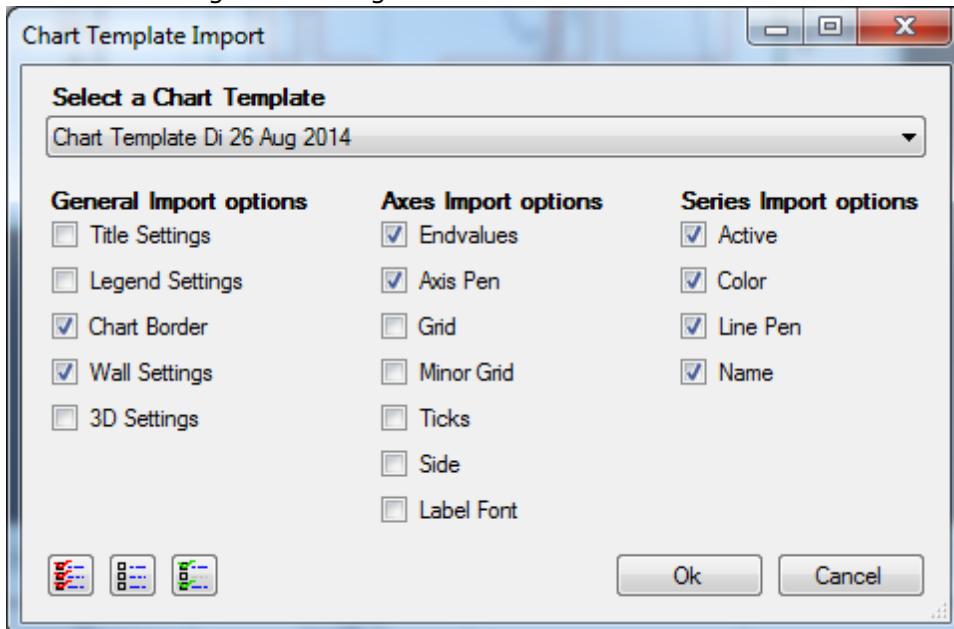
Die Templates werden beim Deinstallieren von LogView Studio nicht gelöscht. Das Userverzeichnis bleibt erhalten und muss ggf. von Hand gelöscht werden.

Template laden

Zunächst muss wieder ein Chart selektiert werden. Danach im Chart Menü auf Template -> Load klicken:



Es erscheint folgender Dialog:



Im oberen Bereich kann ein gespeichertes Template selektiert werden. Im unteren Bereich können die Elemente des Templates selektiert werden die in das bestehende Chart übertragen werden sollen.

Die Default Auswahl reicht in den meisten Fällen vollkommen aus.

Templates in verschiedenen Charts

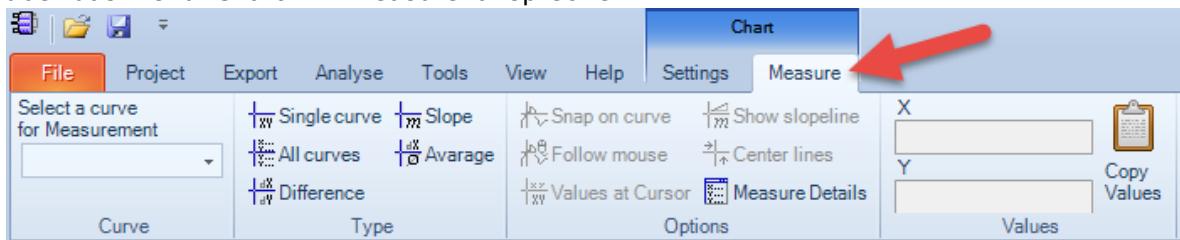
Prinzipiell sollte man Templates nur dann laden wenn es sich um ähnliche Charts handelt - also z.B. Ladekurven eines bestimmten Ladegerätes. Sollte die Anzahl der Kurven z.B. aufgrund unterschiedlicher Geräte nicht passen funktionieren die Templates trotzdem. Es bleiben dann mitunter Kurven im alten Zustand oder vom Template werden Einstellungen ignoriert. Je nachdem ob das Template oder der Chart mehr Kurven ausweist.

Weitere Informationen

[Benutzer Daten entfernen](#)

4.6.3.4 Messfunktionen

LogView Studio bietet für Charts umfangreiche Messmöglichkeiten an. Diese lassen sich über das Menü 'Chart' -> 'Measure' ansprechen:

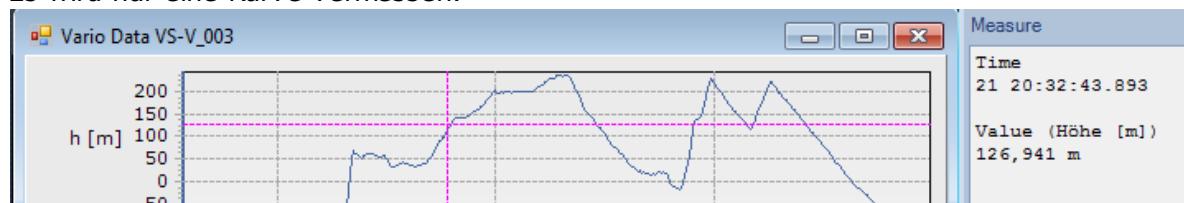


Zum Ein- bzw. Ausschalten muss lediglich eine Kurve eingestellt und ein Messtyp selektiert werden.

Folgende Messmöglichkeiten stehen zur Auswahl:

- **Single Curve**

Es wird nur eine Kurve vermessen.



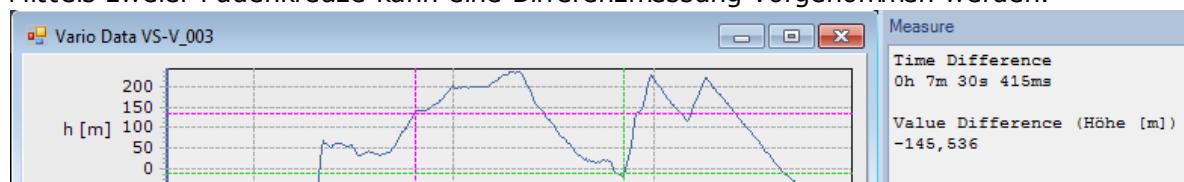
- **All Curves**

Alle dargestellten Kurven werden vermessen. es gibt nur einen horizontalen Messstrich.



- **Difference**

Mittels zweier Fadenkreuze kann eine Differenzmessung vorgenommen werden.



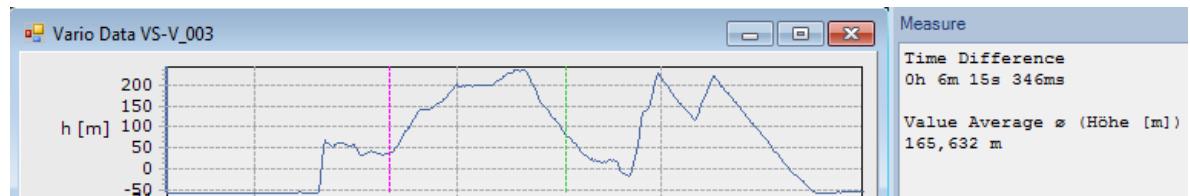
- **Slope**

Mittels zweier Fadenkreuze kann die Steigung einer Kurve berechnet werden.



• Average

Es wird für eine Kurve der Durchschnitt zwischen den Markern berechnet.



Mittels 'Copy Values' lassen sich die Messwerte jederzeit in die Zwischenablage kopieren.

4.7 Beispiele

In diesem Kapitel werden Beispiele zur Nutzung von LogView Studio beschrieben.

Abschnitte

Dieses Kapitel beinhaltet folgende Themen:

- [einfaches Projekt](#)
- [Netzwerk Nutzung](#)

4.7.1 einfaches Projekt

Das folgende Fallbeispiel soll versuchen die Benutzung von LogView Studio etwas pragmatischer zu beschreiben.

Die Umstände

Ein Benutzer hat ein Elektro Flugzeug, zwei Akkus, ein Ladegerät und einen Datenlogger (= Telemetrie). Er möchte nun Daten zu den Flügen sammeln aber auch Informationen über seine Akkus gewinnen.

Die Projektsuche

Der erste Schritt besteht nun darin sich zu überlegen wie das (oder auch die) [Projekt\(e\)](#) aufgebaut werden sollen.

Hinweis

Es obliegt vollkommen dem Benutzer was er unter einem Project zusammenfasst. LogView Studio bildet mit der Project Struktur lediglich einen Rahmen um zusammengehörende Daten abzubilden.

Folgende Grundideen wären denkbar:

- **Ein Projekt "Flugzeug"**
Dem Projekt wird jeweils ein Device "Ladegerät" und "Datenlogger" hinzugefügt. Alle Daten werden mit diesen beiden Devices erfasst. Die erhobenen Daten werden anschließend über Ordnerstrukturen sortiert.
- **Jeweils ein Projekt für das Ladegerät und den Datenlogger**
Die Daten werden jeweils in unterschiedlichen Projekten gespeichert. So wäre zwar kein direkter Bezug herzustellen zwischen Telemetriedaten und Akkudaten, aber wo diese Beziehung irrelevant ist wäre das Szenario durchaus denkbar.
- **Jeweils ein Projekt für die beiden Akkus**

Hier könnten jedem Projekt jeweils das Device "Ladegerät" und "Datenlogger" hinzugefügt werden. Die Erhebung der Daten erfolgt aber für jeden Akku getrennt.

Man sieht also an diesem doch eher einfachen Beispiel das in LogView Studio diverse Möglichkeiten bestehen ein Projekt zu erstellen und zu betrachten. Dem Benutzer werden hier auch keinerlei Vorgaben oder gar Restriktionen vorgegeben. Ein Projekt setzt sich letztlich aus den Elementen zusammen die für einen Benutzer am sinnvollsten erscheinen.

Wir werden im Folgenden die erste Projektvariante 'Projekt Flugzeug' weiter verfolgen.

LogView Studio starten und das Projekt anlegen

LogView Studio starten sollte noch die einfachste Aufgabe sein. Im einfachsten Fall 'LogView Studio' aus dem Start Menü aufrufen. Alternativ im Datei Explorer die Datei LvGui.exe starten.

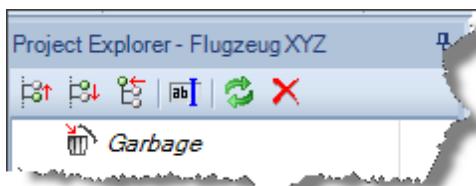
Als Nächstes legen wir das Projekt an über File -> New -> Project (alternativ im LogView Studio Wizard auf den Eintrag 'New Project')



Im LogView Studio Wizard einen Project Titel eintragen und einen leeren Ordner im Dateisystem auswählen.

Der Projektname sollte unser angestrebtes Projekt widerspiegeln. Der Project Folder ist der Pfad auf dem Dateisystem wo alle relevanten Daten zum Projekt (inklusive der Datenbank) gespeichert werden.

Wir warten bis das Projekt erstellt wurde. Im Hintergrund wird die Datenbank gestartet und diverse Vorlagen Dateien in den Project Folder kopiert. Wenn das Projekt erstellt ist ändert sich die Überschrift im Projekt Explorer und der Mülleimer (Garbage) ist im Projekt vorhanden:



Geräte im Projekt einfügen

Im Menü 'Project' auf 'Device' klicken und das Ladegerät einfügen. Dabei auch gleich die Source (also die Datenquelle) einstellen. In unserem Beispiel ist das Device ein Graupner Ultramat 16 mit einem Seriellen Port (Source) auf COM22. Den Dialog mit OK bestätigen.

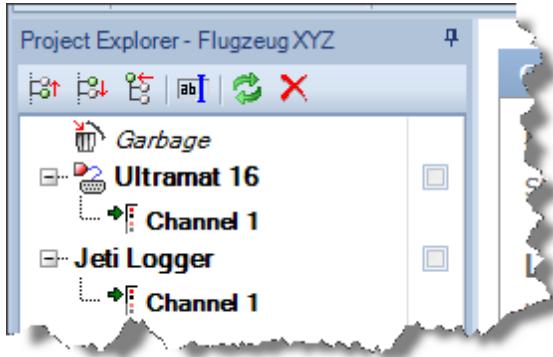
Jetzt den Datenlogger (bzw. die Telemetrie Auswertung) hinzufügen. Wieder auf 'Device' klicken und den Jeti Logger auswählen. Die Source lassen wir zu diesem Zeitpunkt mal unbeachtet. Mit OK bestätigen und wir haben die Devices eingefügt.

Hinweis

Die Geräte sind an dieser Stelle natürlich frei ausgewählt. Sollten Sie das Project

nachstellen wollen dann gleich die eigenen Geräte einfügen.

Das Ergebnis sollte so ausschauen:



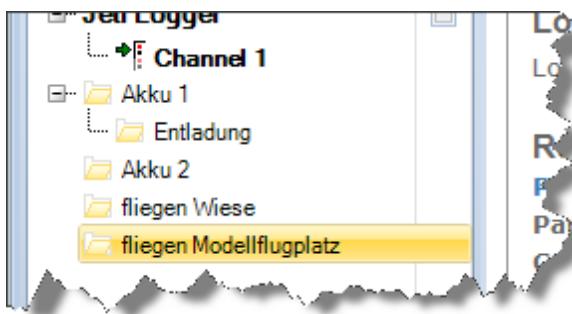
Projektstruktur um Ordner erweitern

Wir haben nun ein Project mit zwei hinzugefügten Devices. Um später nicht den Überblick bei den ganzen Datensätzen zu verlieren erzeugen wir uns nun ein paar Ordner. Die nutzen wir später um Ordnung zu schaffen.

Im 'Project' Menü auf Folder klicken. In dem aufgehenden Dialog einen Ordnernamen - z.B. Akku 1 - eingeben und mit OK bestätigen.

Wenn ein bestehender Ordner angeklickt wird kann man mit einem Klick auf 'Folder' einen Unterordner anlegen. Oder einfach in den Project Explorer an eine frei Stelle klicken und dann auf Folder um einen neuen Haupt Ordner zu erzeugen.

Die Struktur könnte nach wenigen Kicks so aussehen:

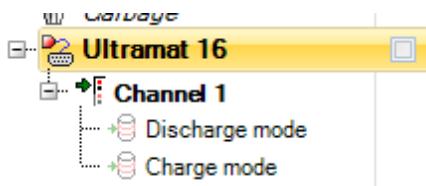


Zu diesem Zeitpunkt haben wir unser Projekt schon weitestgehend nach unseren (Projekt-) Bedürfnissen eingerichtet. Nun könnten Daten kommen ...

Das Ladegerät

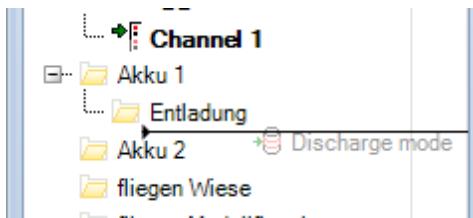
Unser Benutzer fängt nun an seine Akkus mit dem Ladegerät zu bearbeiten. Er nimmt sich also Akku 1 vor und entlädt ihn zunächst. Danach wird der Akku für den nächsten Flug aufgeladen.

Nachdem der Lader also gestartet () und der Akku erst entladen und dann voll geladen wurde sieht unser Ultramat Device folgendermaßen aus:



Der Benutzer möchte nun alle Ladevorgänge mit Akku 1 unter dem Ordner Akku 1 ablegen. Da eher selten entladen wird sollen diese Datensätze unter Akku 1 -> Entladung abgelegt werden.

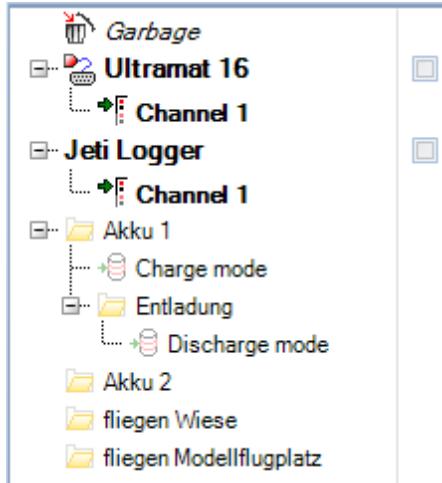
Der Datensatz "Discharge Mode" wird also mit der linken Maustaste festgehalten und dann verschoben nach Entladung.



Hier wird dann die Maustaste losgelassen und der Datensatz ist in den Ordner verschoben - genauso wie es im Windows Explorer funktioniert.

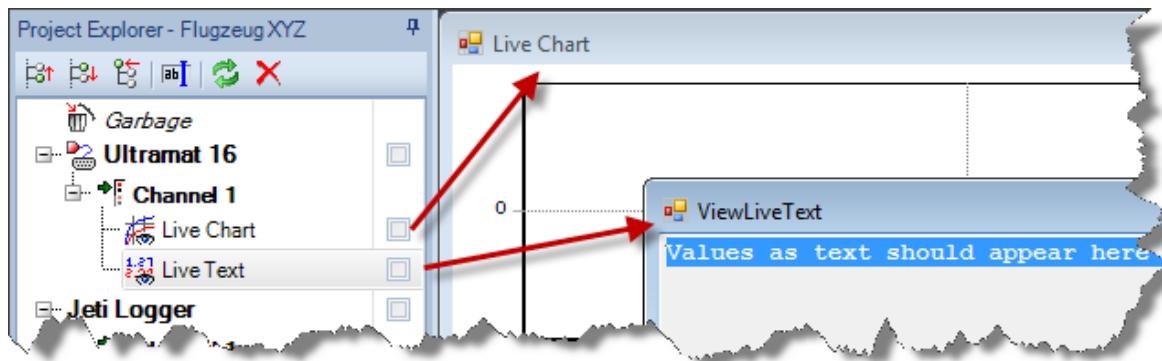
Mit dem Datensatz 'Charge Mode' wird genauso verfahren - nur das der im Ordner Akku 1 landet.

Unsere Projektstruktur sieht nun wie folgt aus:



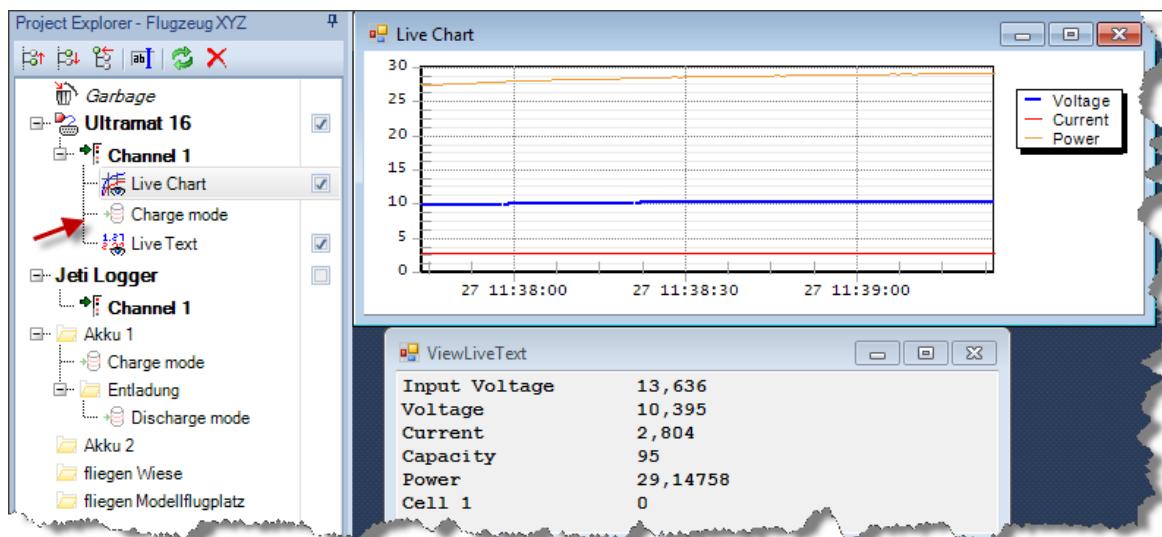
Also nächstes lädt der Benutzer den 2. Akku mit dem gleichen Ladegerät. Die Daten werden nun wiederum am Channel 1 des Ultramat 16 eingefügt. Da wir aber unser Projekt (also die Behandlung von Akku 1) schon bereinigt haben stellt das kein Problem dar.

Der Benutzer möchte aber während dem Laden des Akku 2 zum einen die aktuellen Werte des Ladegerätes sehen und zudem eine Grafik mit den letzten Daten. Dazu fügt er dem Channel 1 des Ultramat 16 einen Chart und eine Text View hinzu:



Diese Live Views zeigen nun - wenn sie aktiviert werden - die letzten Daten aus der Datenbank und aktualisieren sich in regelmäßigen Abständen.

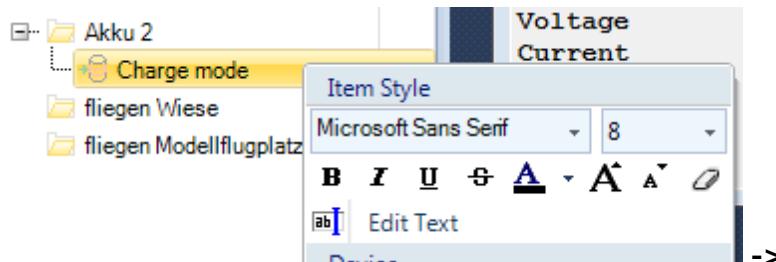
Nachdem das Gerät und die Live Views gestartet wurden könnte das Laden von Akku 2 so aussehen:

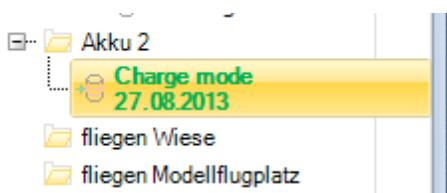


Der Live Chart zeigt ähnlich wie ein Oszilloskop den Ladeverlauf und die Text View zeigt die letzten Werte.

Zudem wurde ein neuer Datensatz für das Laden angelegt - 'Charge Mode'.

Nachdem der Ladevorgang abgeschlossen ist verschiebt der User den Datensatz nach Akku 2. Über einen Rechtsklick auf den Datensatz kann der Name und der Style angepasst werden:





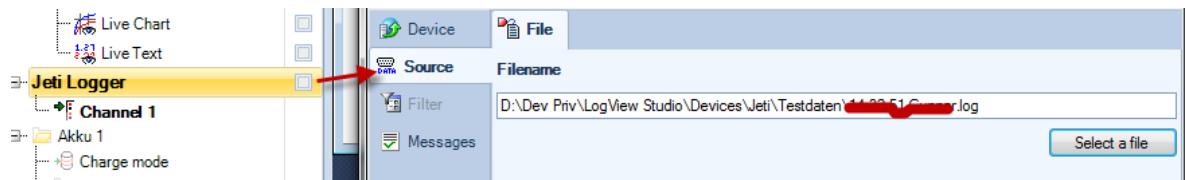
Zwischenstand

Der Benutzer hat seine Akkus behandelt und die Ergebnisse in der Projektstruktur einsortiert. Zudem kann bei dem Ladegerät die Datenaufzeichnung mittels Live Views verfolgt werden.

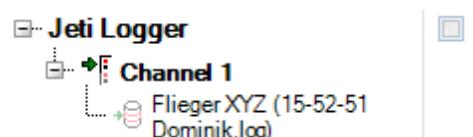
Benutzer ist geflogen - Telemetriedaten sind verfügbar

Jetzt kommen langsam die Stärken von LogView Studio zum tragen wenn es um den Umgang mit mehreren Geräten geht. Denn nun werden in das gleiche Projekt Daten importiert die zwar im Zusammenhang mit den Akkus und den Ladungen stehen, aber eben von ganz anderen Geräten erfasst wurden.

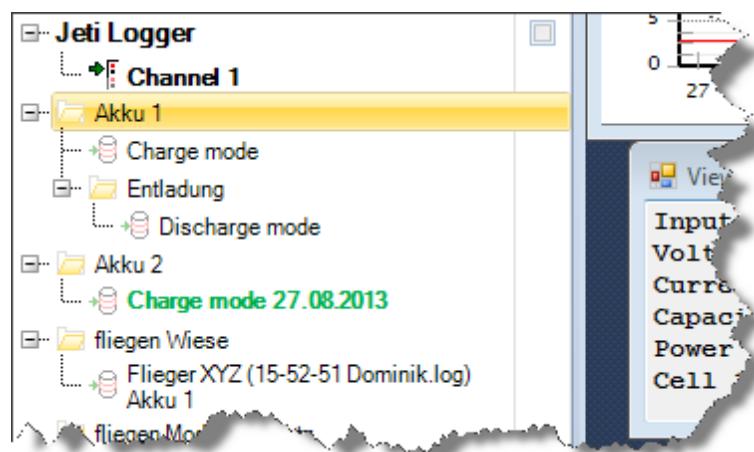
Der Benutzer ist also auf seiner Wiese geflogen, hat Akku 1 genutzt und ein entsprechendes Telemetrie Logfile. Das Logfile wird geöffnet indem man auf den Jeti Logger einen Doppelklick macht.



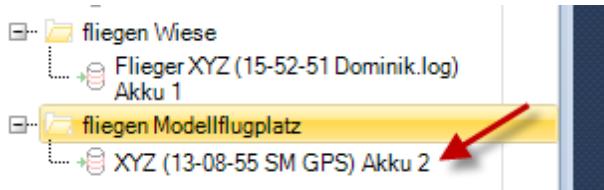
Im aufgehenden Device Dialog wählt man für die Source (in dem Fall File) das Logfile aus und bestätigt mit OK. Das Logfile wird ohne das Zutun des Benutzers sofort importiert und im Projekt als Datensatz eingetragen:



Der Ordnung halber wird auch dieser Datensatz direkt wieder an die richtige Stelle verschoben und um die Info Akku 1 ergänzt:



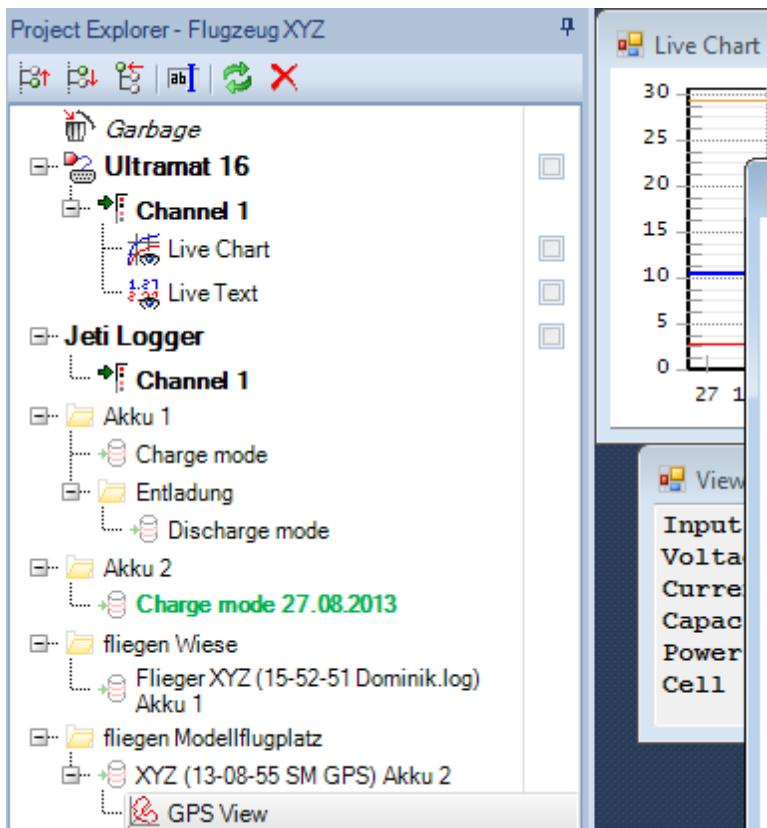
Der Benutzer geht danach auf den Modellbauplatz und fliegt Akku 2 leer und loggt ebenfalls die Telemetriedaten. Auch hier machen wir wieder einen Doppelklick auf den Jeti Logger (und legen nicht ein neues Device an). In dem Device Dialog wählen wir bei der File Source die neue Datei aus und bestätigen wieder mit OK. Auch den neuen entstehenden Datensatz verschieben wir passend in unserem Projekt:



Auswertung mit Static Views

Als Abschluss möchte der Benutzer noch eine Static View von den letzten Telemetrie GPS Daten sehen. Er wählt als den entsprechenden Datensatz aus und klickt im Menü 'Project' auf GPS 3D.

Das Project sieht danach folgendermaßen aus:



Alle nachfolgenden Ladungen oder Logfiles können auf ähnliche Art und Weise gehandhabt werden.

Zusammenfassung

Das obige abstrakte Beispiel zeigt hoffentlich etwas genauer wie LogView Studio eingesetzt werden kann und sollte. Die Stärken liegen hier ganz klar in der Fähigkeit mehrere Geräte gleichzeitig zu bedienen als auch die resultierenden Daten sauber zu

strukturieren.

Sollten trotz dieser Ausführung weitere Fragen bestehen sollten diese in unserem Forum unter www.logview.info platziert werden.

4.7.2 Netzwerk Nutzung

Das folgende Fallbeispiel beschreibt die Nutzung von LogView Studio im Netzwerk. Dabei kommen zwei Windows Rechner zum Einsatz.

Voraussetzungen

- Der "normale" Umgang mit LogView Studio sollte bekannt sein.
- Die [Einführung zum Netzwerk Thema](#) sollte verstanden sein.
- Die Rechner müssen sich über das Netzwerk erreichen können.
- Auf den Rechnern muss [LogView Studio installiert](#) sein.
- Es kann unter Umständen nötig sein den TCP Port 2804 in der Firewall zu öffnen um die Kommunikation mit der Datenbank zu erlauben.

Umgebung

Für das folgende Beispiel werden zwei Windows Systeme (Windows 7 64Bit und Windows XP 32 Bit) verwendet. Beide Rechner sind mit dem selben Netzwerk verbunden und es ist LogView Studio installiert.

Rechner 1 - Windows XP

- LogView Studio ist gestartet
- Es ist ein Project angelegt in dem ein Device hinzugefügt wurde (in diesem Fall ein Jansi 4010 DUO Lader)
- Das Gerät ist an den Rechner angeschlossen, betriebsbereit und kann Daten an LogView Studio liefern

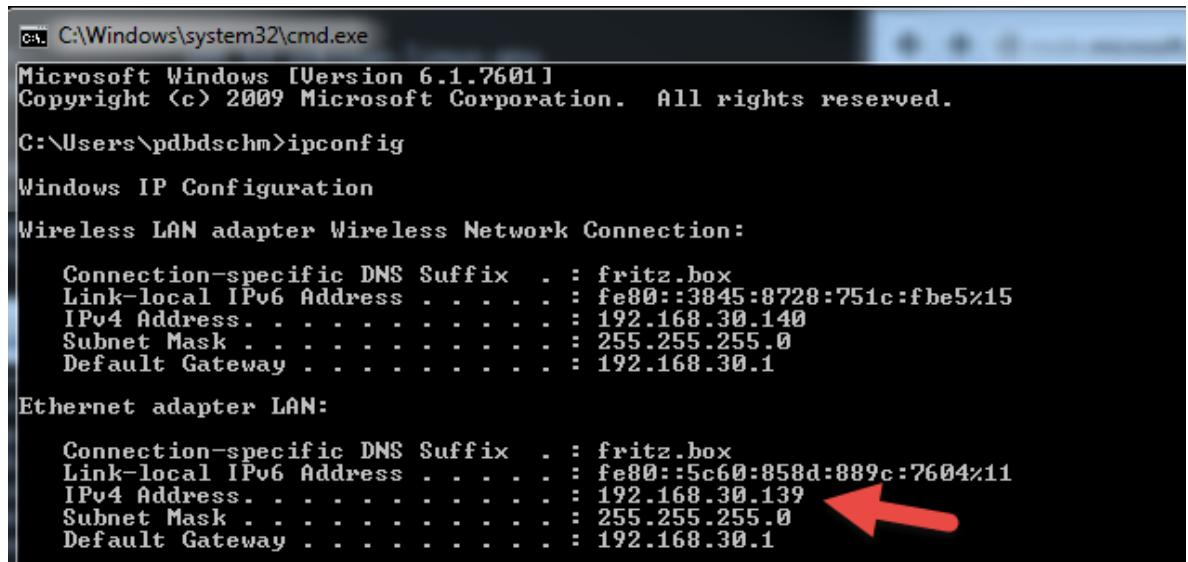
Rechner 2 - Windows 7

- LogView Studio ist gestartet
- Es ist noch kein Project vorhanden

Vorgehensweise

Rechner 2 soll auf das Project von Rechner 1 zugreifen und die Daten darstellen.

- Als erstes muss die IP Adresse von Rechner 2 ermittelt werden. Am einfachsten geht das unter Windows mittels der Commando Zeile.
 - Windowstaste + R drücken
 - `cmd` eingeben unter mit Enter bestätigen
 - in dem Konsolenfenster `ipconfig` eingeben und Enter
 - Es wird nun eine Liste angezeigt mit Netzwerkadaptersn. Je nach Verbindung nach LAN oder WLAN suchen und die IP Adresse notieren.
- Eine sehr gute Hilfe zu dem Thema findet man hier:
- http://www.tippscout.de/windows-xp-ip-adresse-des-computers-ermitteln-und-herausfinden_tipp_2676.html
- <http://de.wikihow.com/Die-IP-Adresse-des-PCs-herausfinden>



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright <c> 2009 Microsoft Corporation. All rights reserved.

C:\Users\pdbschm>ipconfig

Windows IP Configuration

Wireless LAN adapter Wireless Network Connection:

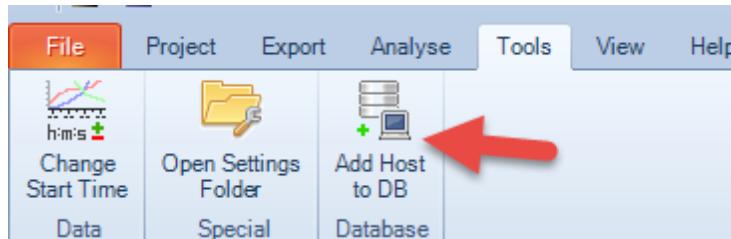
  Connection-specific DNS Suffix . : fritz.box
  Link-local IPv6 Address . . . . . fe80::3845:8728:751c:fbe5%15
  IPv4 Address . . . . . 192.168.30.140
  Subnet Mask . . . . . 255.255.255.0
  Default Gateway . . . . . 192.168.30.1

Ethernet adapter LAN:

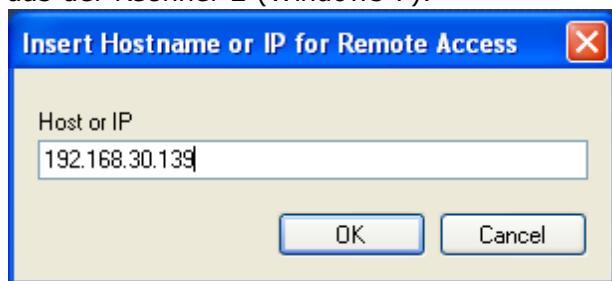
  Connection-specific DNS Suffix . : fritz.box
  Link-local IPv6 Address . . . . . fe80::5c60:858d:889c:7604%11
  IPv4 Address . . . . . 192.168.30.139
  Subnet Mask . . . . . 255.255.255.0
  Default Gateway . . . . . 192.168.30.1
```

Im dem Beispiel ist die IP Adresse 192.168.30.139.

- Der nächste Schritt besteht darin, auf Rechner 1 den Rechner 2 frei zu geben. Dafür gibt es in der Oberfläche von LogView Studio einen Button im Tools Menü - 'Add Host to DB':



Es erscheint ein Dialog in dem die IP Adresse (oder der Hostname) des Rechners eingetragen wird, der auf das bestehende Project zugreifen soll. In unserem Fall ist das der Rechner 2 (Windows 7).



Den Dialog mit OK schließen.

- 'Jetzt kann auf Rechner 2 (LogView Studio läuft bereits) ein neues Project gestartet werden, aber jetzt ein 'Remote Project'. Im File Menü auf 'Remote Project' klicken:'



- In dem folgenden Dialog einen Project Ordner anlegen. Dieser Ordner wird lokal benötigt um Daten (wie z.B. Charts) zu speichern.

- Mit Next auf die nächste Seite des Dialogs wechseln. Dort muss die IP des Rechners angegeben werden wo das Project aktiv ist (also in unserem Fall Rechner 1 - XP).

Project Settings

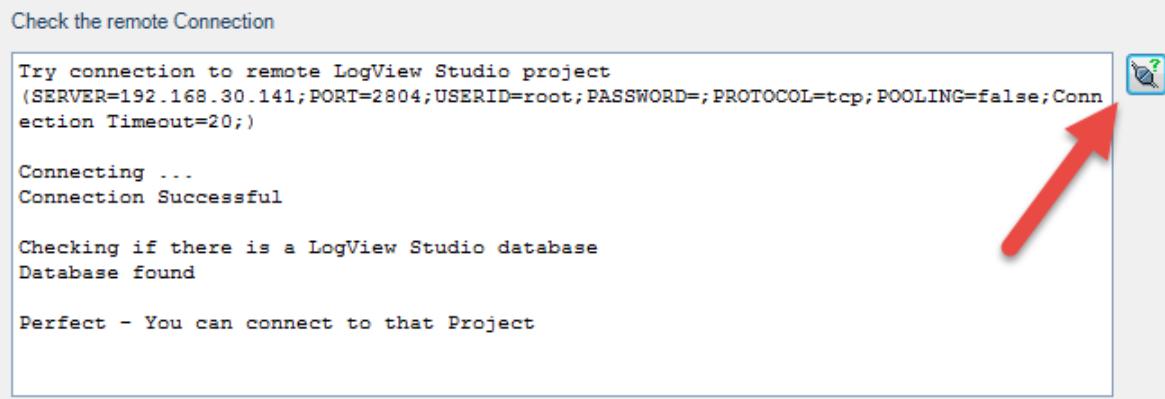
Note

You can only connect to a running LogView Studio Project. It is r

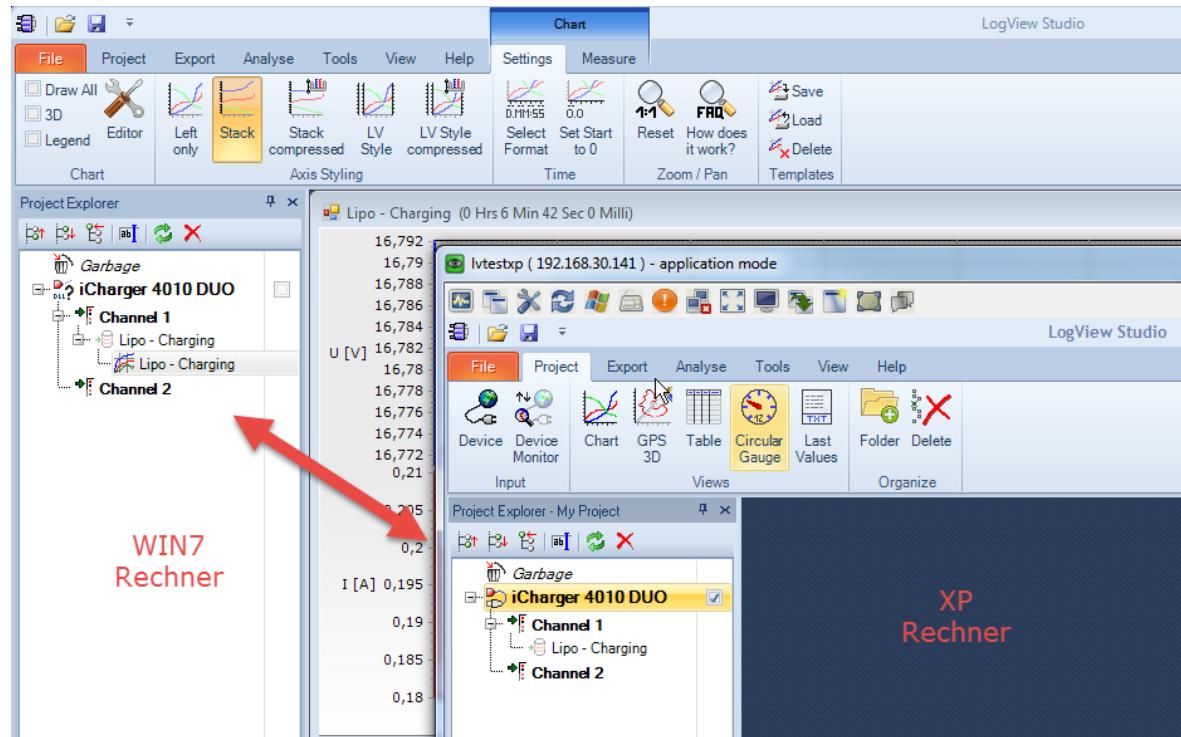
Hostname / IP-Adress

192.168.30.141|

- Jetzt kann die Verbindung geprüft werden:



- Wenn alles Ok ist, den Dialog mit Finish beenden.
- Als Ergebnis sollte jetzt auf beiden Rechnern ein LogView Studio laufen. Wobei Rechner 2 die Daten von Rechner 1 abholt:



Hinweis

Es können beliebig viele Rechner an ein Projekt gebunden werden.

Hinweis

Auch mittels LogView Studio Command kann man auf ein solches Project zugreifen!

Erreichbarkeit aus dem Internet

Die obige Konstellation wäre auch denkbar wenn ein oder mehrere Rechner aus dem Internet zugreifen wollen. In dem Fall müsste allerdings die Firewall und / oder der Router entsprechend eingerichtet werden. Ein Schlagwort ist hier u.a. Portforwarding.

Wichtig ist dabei zu wissen das der entfernte Rechner auf Port 2804 zugreifen können muss.

Weitere Informationen

Bei weiteren Fragen empfehlen wir unser Forum (<http://www.logview.info/forum/index.php>). Dort die Sektion 'LogView Studio' anwählen und nach bereits möglichen Lösungen suchen oder Fragen an die Community stellen. Wir die Autoren sind ebenso im Forum vertreten und geben gerne unsere Unterstützung sofern möglich.

5 Linux Prepare



This Topic will explain how to install different Linux Systems in order to get LogView Studio Command running.

At the Moment we can support these Systems:

- Arm6 32Bit - e.g. [Raspberry Pi](#)
- Arm7 32Bit - e.g. [Cubietruck](#)
- Generic x86 / x64 Linux (we will cover (L)Ubuntu and CentOS in this document)

Procedure

There are typically 4 steps to do:

- Install the Operating System and apply the latest updates
- Install (or Compile) Mono
- Copy the LogView Studio Files to the System
- Run LogView Studio and get the Database Files

5.1 SSH Connect

To get a connection to your Linux system it's very handy to have SSH up and running.

For a SSH connection you need some software on your Windows system (SSH client) and some software on your Linux system (SSH server).

This topic will cover Putty as a SSH client.

Download

Download putty.exe from here

<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>

64Bit Version is available here

<http://www.dzinovic.net/projects/windows/putty-64-bit-builds/>

Or use this direct link for downloading

<http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe>

Installation

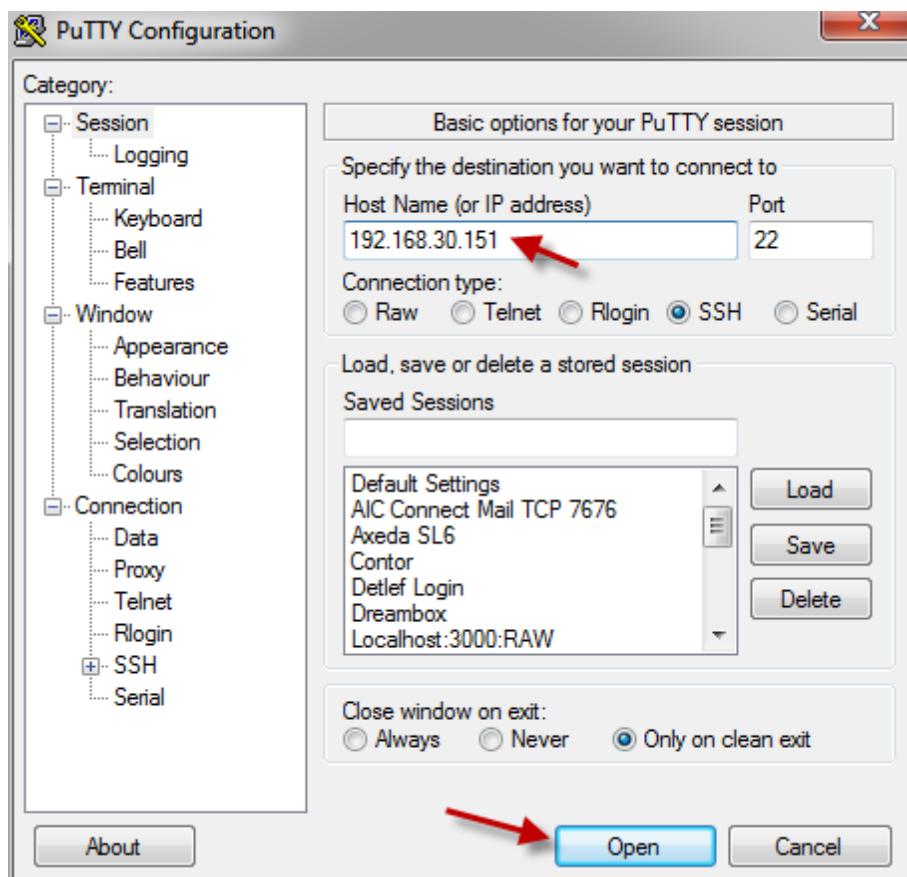
Just execute the file and you are done.

Get the IP from your Linux system

```
pi@raspberrypi ~ $ ifconfig
eth0      Link encap:Ethernet  HWaddr b8:27:eb:c9:53:f7
          inet addr:192.168.30.151  Bcast:192.168.30.255
          Mask:255.255.255.0
                  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
                  RX packets:69364 errors:0 dropped:0 overruns:0 frame:0
                  TX packets:10413 errors:0 dropped:0 overruns:0 carrier:0
                  collisions:0 txqueuelen:1000
                  RX bytes:92340960 (88.0 MiB)   TX bytes:980385 (957.4 KiB)
```

Run Putty

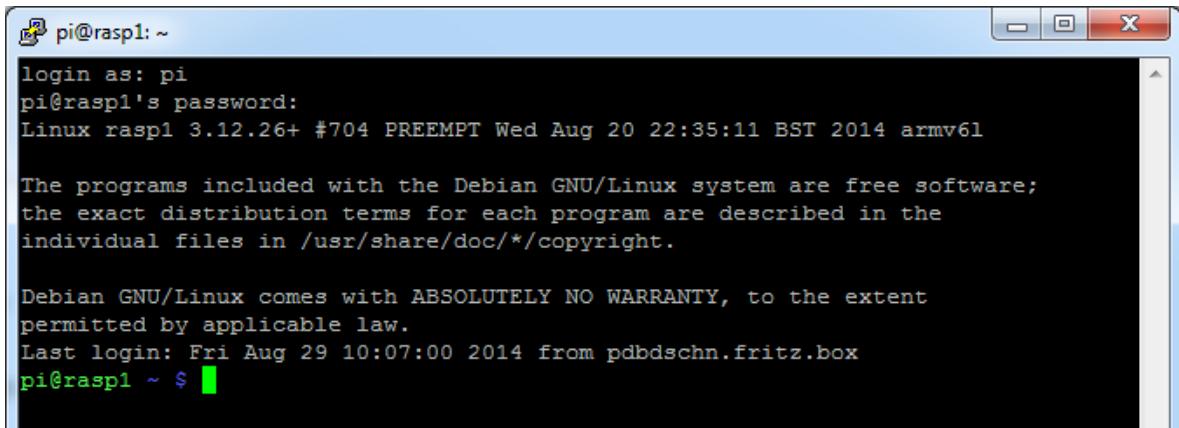
Start putty.exe, enter the IP-address and click on Open



Accept the key



Enter username and password and you are connected.



A screenshot of a terminal window titled "pi@rasp1: ~". The window shows a successful login as user "pi" with the password. The system information includes the kernel version (3.12.26+), date (Wed Aug 20 22:35:11 BST 2014), and architecture (armv6l). It also displays the standard Debian GNU/Linux copyright notice and the last login information (Fri Aug 29 10:07:00 2014 from pdbdschn.fritz.box).

```
pi@rasp1: ~
login as: pi
pi@rasp1's password:
Linux rasp1 3.12.26+ #704 PREEMPT Wed Aug 20 22:35:11 BST 2014 armv6l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug 29 10:07:00 2014 from pdbdschn.fritz.box
pi@rasp1 ~ $
```

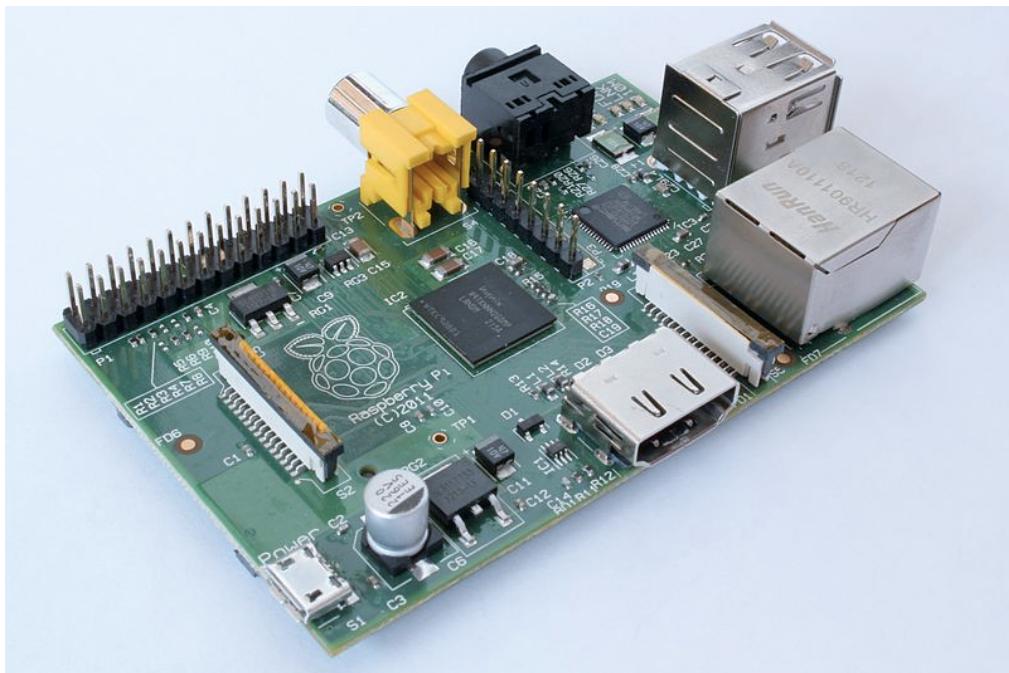
Additional Information

Mobaxterm as an alternative to putty
<http://mobaxterm.mobatek.net/>

Xming X Server for Windows
<http://sourceforge.net/projects/xming/>

5.2 Raspberry Pi

The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. (http://en.wikipedia.org/wiki/Raspberry_Pi)



System

700MHz CPU (ARM6)

256MB or 512MB Memory

SD Card Storage

Webpage

<http://www.raspberrypi.org/>

Default Credentials

User pi
Password raspberry

Additional Information

RPi Low-level peripherals (http://elinux.org/RPi_Low-level_peripherals)

5.2.1 System Preparation

The following section will explain how to setup a normal Raspberry Pi. It will not cover every detail but will give a good overview on the important steps.

Hint

For the next steps you must connect a display, mouse and a keyboard to your Pi.

Download

For the Raspberry Pi we use the normal NOOBS Installer (actually version 1.3.9). You can download it from here:

<http://www.raspberrypi.org/downloads>

System Installation

Place the extracted files on a clean SD card, put it into the Raspberry and boot it up. Now you can install different Operating Systems with the installer.

We install the recommended Raspbian System.

System Preparation

After the installation finished and the Raspberry has rebooted it will show you the configuration (raspi-config). Here you can tweak some settings.

- Expand the File system in order to use the whole SD card (may not needed with NOOBS)
- Change the User Password (if you like)
- **Don't change the Local** (Translation of the System). This will cause trouble during compilation.
You can switch the Local after LogView Studio is running fine.
- To get some more power set the Overclocking (900MHz is safe, 950MHz should work, 1000MHz often results in errors)
If the system is busy you can check the result with
`watch -n 1 cat /sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq`
- Ensure that SSH is enabled (Advanced options)
We will use SSH to connect to the Raspberry after the installation is done.

Reboot the system to activate the changes

From this point we will use [SSH to connect](#) to the Raspberry. You can disconnect any keyboard, mouse and display if you like.

We only need the Raspberry powered up and connected with a network cable.

Software Install / Update

- Update the software package lists
`sudo apt-get update`
- Update the installed software packages (this may take a while)
`sudo apt-get dist-upgrade`
- Install some additional useful packages (maybe some are preinstalled)
`sudo apt-get install sudoku htop git build-essential autoconf automake screen curl binutils`
(It's always a good idea to have [Sudoku](#) installed :-)
- Install the latest Raspberry Pi Firmware
`sudo rpi-update`
- Reboot
`sudo reboot`

Additional Information

German Information about Raspberry installation

<http://linuxwelt.blogspot.de/2013/10/raspberry-pi-raspbian-installieren.html>

Raspberry Pi Firmware-Update

<http://raspberrypiguide.de/howtos/raspberry-pi-firmware-update/>

General Information about Debian software installation / updates

<https://help.ubuntu.com/community/AptGet/Howto>

5.2.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

<http://www.mono-project.com>

How to get Mono?

There are at least three different ways for getting Mono installed on your Raspberry Pi. All of them have their pros and cons:

Install it from the Raspbian package repository

- + very easy to handle
- + fast installation
- + works with LogView Studio (LvCmd)
- mostly an old Mono release
- WinForms applications produce errors with some components (like a Textbox)

Compile it from a TAR package

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
 - o mostly newer than the Raspbian package but older than the GIT version
- takes a long time to compile
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

Compile it from the GIT repository

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- takes a long time to compile
- needs some extra packages for compiling

Which method is the best?

For LogView Studio it depends on what you like to do. If you only need the LvCmd version you are just fine with the normal Raspbian package.

If you like to run WinForms applications it's a better choice to compile it from GIT.

Note

If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.

AOT

Ahead of Time Compilation or AOT is a feature of the Mono runtime code generator. We don't need it to run LogView Studio and in some cases it results in compilation problems on ARM. So we just disable it.

<http://www.mono-project.com/docs/advanced/aot/>

5.2.2.1 Install Mono

This will install the Mono package from the Raspbian repository.

Install

Installing Mono on the Raspberry Pi is a simple task. Just use the following command:

```
sudo apt-get install mono-complete libgdiplus
```

If you want to try C# development as well just add `monodevelop`, too.

```
sudo apt-get install mono-complete libgdiplus monodevelop
```

The installation will take some minutes.

Uninstall

Uninstalling is as simple as installing:

```
sudo apt-get autoremove monodevelope (only if you installed it)  
sudo apt-get autoremove libgdiplus
```

Hint

libgdiplus removes mono-complete as well.

5.2.2.2 Compile Mono from TAR

This will compile Mono from a TAR package. **It takes about 9 hours in sum @ 950MHz.**

Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev
libexif-dev gettext libcairo2-dev
```

- Install libgdiplus

```
sudo apt-get install libgdiplus
```

(You can also compile the latest version. See 'Compile Mono from GIT' for details.)

- Create a new folder where we place the files. You can use your home directory for that

```
cd ~
mkdir sources
cd sources
```

- Check for the latest TAR version

```
wget -qO- http://download.mono-project.com/sources/mono/ | grep -o -E
'href="([^\"]+)"' | cut -d '"' -f2 | sort | uniq
```

or just watch at the website

<http://origin-download.mono-project.com/sources/mono/>
<http://download.mono-project.com/sources/mono/>

- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)

```
wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2
```

- Extract the TAR file

```
tar -jxvf mono-3.6.0.tar.bz2
```

- Enter the new directory

```
cd mono-3.6.0
```

- run configure (this will take ~10 minutes @ 950MHz)

```
./configure --prefix=/usr/local/
```

/usr/local/ is the installation path of the compiled files

- run make - compile all the files (this will take ~8:30 hours @ 950MHz)

```
sudo SKIP_AOT=true make
```

- run make install - this will install the needed files into your system (this will take ~25 minutes @ 950MHz)

```
sudo SKIP_AOT=true make install
```

Uninstall

```
sudo make uninstall
```

Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv

<https://gist.github.com/pjvds/7185693>

5.2.2.3 Compile Mono from GIT

This will compile Mono from the GIT repository. **It takes about 10:30 hours in sum @ 950MHz.**

Hint

Since this is updated very often it could include some bugs. But in most cases it works great.

Procedure

- Install additional needed packages
`sudo apt-get install libtool libglib2.0-dev libxrender-dev
libfontconfig-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev
libexif-dev gettext libcairo2-dev`
- Create a new folder where we place the files. You can use your home directory for that
`cd ~
mkdir sources
cd sources`

libgdiplus Compilation (this will take overall ~16 minutes @ 950MHz)

- Get the libgdiplus sources from GIT
`git clone git://github.com/mono/libgdiplus`
- Enter the new directory
`cd libgdiplus`
- Run autogen
`./autogen.sh --prefix=/usr`
- Compile the sources
`sudo make`
- Install the files
`sudo make install`
- Leave the directory
`cd ..`

Mono Compilation (tak ????)

- Get the Mono sources from GIT (~45 minutes)
`git clone git://github.com/mono/mono.git`
- Enter the new directory
`cd mono`
- Run autogen (~20 minutes)
`./autogen.sh --prefix=/usr/local`
- Install and configure gmcs
`sudo make get-monolite-latest
cd mcs/class/lib/monolite
cp basic.exe gmcs.exe
cd /home/pi/sources/mono`

Hint

You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.

- Compile the sources (this will take ~9:00 hours @ 950MHz)
If you used sudo make get-monolite-latest use this:
`sudo SKIP_AOT=true make EXTERNAL_MCS="/home/pi/sources/mono/mcs/class/lib/monolite/gmcs.exe"`
If you use apt-get install mono-gmcs use this:
`sudo SKIP_AOT=true make`
- Install the files (this will take ~30 minutes @ 950MHz)
`sudo SKIP_AOT=true make install`

Uninstall

- Run the uninstall
`sudo make uninstall`
- remove the source files
`rm -rf ~/sources/mono`

Additional Information

Raspberry Pi Mono libgdiplus

<http://www.shatalmic.com/raspberry-pi-mono-libgdiplus>

Mono Compile from Git

<http://stackoverflow.com/questions/13365158/installing-mono-3-x-3-0-x-and-or-3-2-x>

<https://gist.github.com/pjvds/7185693>

5.2.2.4 Testing

These tests work for all types of installation.

First Test

If the installation is done just type the following command:

`mono --version`

```
Mono JIT compiler version 3.2.8 (Debian 3.2.8+dfsg-4+rpi1)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and Contributors. www.mono-project.com

TLS:          __thread
SIGSEGV:      normal
Notifications: epoll
Architecture: armel,vfp+hard
Disabled:     none
Misc:         softdebug
LLVM:         supported, not enabled.
GC:          sgen
```

Ensure that you find the verb "hard" in the Architecture line.

If you get the hard info in the output you are done with the mono installation.

The version should match your installed / compiled version.

Check Hard Float Unit

There were some hard float issues in the past releases of Mono. You can simple check if it works with the csharp shell.

```
pi@raspberrypi ~ $ date  
Sun Mar 16 13:12:07 UTC 2014  
pi@raspberrypi ~ $ csharp  
Mono C# Shell, type "help;" for help  
  
Enter statements below.  
csharp> DateTime.Now;  
03/16/2014 13:12:39  
csharp>
```

You must see the correct datetime. Otherwise your mono is damaged.

Check WinForms

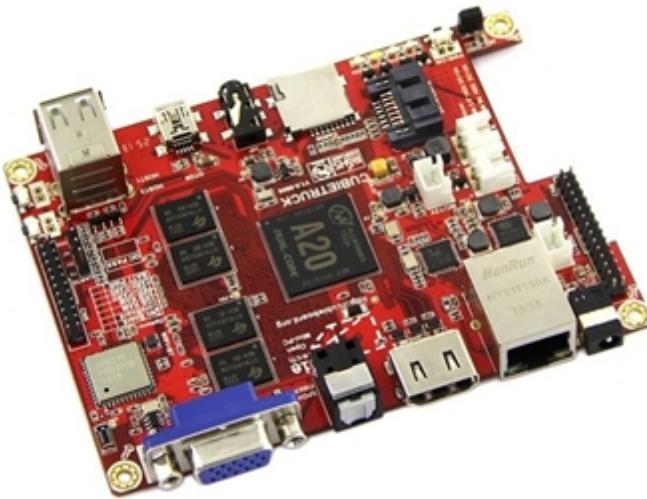
If you need to run WinForms applications just compile a simple demo with a TextBox and ComboBox. You will receive errors when you start that application under Mono. If the application starts fine your Mono is ok.

Hint

It could be important to run the application with sudo. Just check that.

5.3 Cubietruck

Cubietruck is the 3rd board of Cubieteam, so we also name it Cubieboard3. It's a new PCB model adopted with Allwinner A20 main chip, just like Cubieboard2.



System

1000MHz ARM® Cortex™-A7 Dual-Core

1GB or 2GB Memory

HDD and SD Card Storage

Webpage

<http://cubieboard.org/>

Image

<http://cubian.org/>

Default Credentials

User cubie

Password cubie

5.3.1 System Preparation

The following section will explain how to setup a Cubietruck. It will not cover every detail but will give a good overview on the important steps.

Hint

Display, mouse and keyboard are not needed.

Download

For the Cubietruck we use Cubian SD Image (actually version **Cubian-desktop-r1-a20-ct**). You can download it from here:

<http://cubian.org/downloads/>

System Installation

- Unpack the 7z file to get the img file
- Use Win32DiskImager (<http://sourceforge.net/projects/win32diskimager/>) to bring the image to the SD card. You need at least a 4GB SD card.
- Put the SD card in the Cubietruck und power it on

System Preparation

- Connect to 'cubian' via [SSH](#) (Port 36000, User cubie, Password cubie)
- Bring Cubian to the NAND-Flash ...
- `sudo bash` (Password cubie)
- Get a PGP Key for debian:
`wget -O - http://packages.cubian.org/cubian.gpg.key | apt-key add -`
- Update the software package lists
`apt-get update`
- Update the installed software packages (this may take a while)
`apt-get dist-upgrade`
- Get the Script for NAND Install
`apt-get install cubian-nandinstall`
- Run the Script for NAND Install (takes about 10 minutes)
`cubian-nandinstall`
If the script finished it will shut down your Cubietruck (after you enter Y).
- Remove the SD card and power on the Cubietruck again. It should boot from NAND now.

- Set Timezone
`sudo dpkg-reconfigure tzdata`
- Locales in English (en_US.UTF-8)
`sudo dpkg-reconfigure locales`
- Kernel update
`sudo apt-get update && sudo apt-get install cubian-update`

If you get messages like this:

```
locale: Cannot set LC_CTYPE to default locale: No such file or directory
locale: Cannot set LC_MESSAGES to default locale: No such file or directory
locale: Cannot set LC_ALL to default locale: No such file or directory
```

try to reinstall the locales:

```
sudo apt-get install --reinstall locales
```

Software Install / Update

- Install some additional useful packages (maybe some are preinstalled)
`sudo apt-get install sudoku htop git build-essential autoconf automake screen curl binutils`
(It's always a good idea to have [Sudoku](#) installed :-)

Hints

General Information about Debian software installation / updates

<https://help.ubuntu.com/community/AptGet/Howto>

Cubian Webpage

<http://cubian.org/>

Very good Tutorial how to install Cubian (German)

<http://www.kriwanek.de/cubieboard.html>

<http://www.kriwanek.de/cubieboard/393-cubian-auf-sd-card-installieren.html>

<http://www.kriwanek.de/cubieboard/402-cubian-auf-nand-flash-uebertragen.html>

<http://www.kriwanek.de/cubieboard/396-cubian-auf-sata-festplatte-verlagern.html>

5.3.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

<http://www.mono-project.com>

How to get Mono?

There are at least three different ways for getting Mono installed on your Cubietruck. All of them have their pros and cons:

Install it from the debian package repository

- + very easy to handle
- + fast installation
- + works with LogView Studio (LvCmd)

- mostly an old Mono release
- WinForms applications produce errors with some components (like a Textbox)

Compile it from a TAR package

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
 - o mostly newer than the Raspbian package but older than the GIT version
- takes a long time to compile
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

Compile it from the GIT repository

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- takes a long time to compile
- needs some extra packages for compiling

Which method is the best?

For the Cubietruck the best choice is to compile it from GIT.

Note

If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.

AOT

Ahead of Time Compilation or AOT is a feature of the Mono runtime code generator. We don't need it to run LogView Studio and in some cases it results in compilation problems on ARM. So we just disable it.

<http://www.mono-project.com/docs/advanced/aot/>

Mono Bug

https://bugzilla.xamarin.com/show_bug.cgi?id=17017

5.3.2.1 Install Mono

This will install the Mono package from the Debian armhf repository.

Install

Installing Mono on the Raspberry Pi is a simple task. Just use the following command:

```
sudo echo "deb http://ftp.debian.org/debian sid main" >> /etc/apt/
sources.list
sudo apt-get update
sudo apt-get install mono-complete libgdiplus -t sid
```

If you want to try C# development as well just add `monodevelop`, too.

```
sudo apt-get install mono-complete libgdiplus monodevelop -t sid
```

The installation will take some minutes.

Uninstall

Uninstalling is as simple as installing:

```
sudo apt-get autoremove monodevelop (only if you installed it)
sudo apt-get autoremove libgdiplus
```

Hint

libgdiplus removes mono-complete as well.

5.3.2.2 Compile Mono from TAR

This will compile Mono from a TAR package. **It takes about 4 hours in sum.**

Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev
libexif-dev gettext libcairo2-dev
```

- Install libgdiplus

```
sudo apt-get install libgdiplus
```

(You can also compile the latest version. See 'Compile Mono from GIT' for details.)

- Create a new folder where we place the files. You can use your home directory for that

```
cd ~
mkdir sources
cd sources
```

- Check for the latest TAR version

```
wget -qO- http://download.mono-project.com/sources/mono/ | grep -o -E
'href="([^\"]+)"' | cut -d '"' -f2 | sort | uniq
```

or just watch at the website

<http://origin-download.mono-project.com/sources/mono/>
<http://download.mono-project.com/sources/mono/>

- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)

```
wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2
```

- Extract the TAR file

```
tar -jxvf mono-3.6.0.tar.bz2
```

- Enter the new directory

```
cd mono-3.6.0
```

- run configure (this will takes ~5 minutes)

```
./configure --prefix=/usr/local/
```

/usr/local/ is the installation path of the compiled files

- run make - compile all the files (this will take ~3:30 hours)

```
sudo SKIP_AOT=true make
```

- run make install - this will install the needed files into your system (this will take ~10 minutes)

```
sudo SKIP_AOT=true make install
```

Uninstall

```
sudo make uninstall
```

Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv

<https://gist.github.com/pjvds/7185693>

5.3.2.3 Compile Mono from GIT

This will compile Mono from the GIT repository. **It takes about 4 hours in sum.**

Hint

Since this is updated very often it could include some bugs. But in most cases it works great.

Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev  
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev  
libexif-dev gettext libcairo2-dev
```

- Create a new folder where we place the files. You can use your home directory for that

```
cd ~  
mkdir sources  
cd sources
```

libgdiplus Compilation (this will take overall ~10 minutes)

- Get the libgdiplus sources from GIT

```
git clone git://github.com/mono/libgdiplus
```

- Enter the new directory

```
cd libgdiplus
```

- Run autogen

```
./autogen.sh --prefix=/usr
```

- Compile the sources

```
sudo make
```

- Install the files

```
sudo make install
```

- Leave the directory

```
cd ..
```

Mono Compilation

- Get the Mono sources from GIT

```
git clone git://github.com/mono/mono.git
```

- Enter the new directory

```
cd mono
```

- Run autogen (~11 minutes)

```
./autogen.sh --prefix=/usr/local
```

- Install and configure gmcs


```
sudo make get-monolite-latest
cd mcs/class/lib/monolite
cp basic.exe gmcs.exe
cd ~/sources/mono
```

Hint

You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.

- Compile the sources (this will take ~3:30 hours)

If you used sudo make get-monolite-latest use this:

```
sudo SKIP_AOT=true make EXTERNAL_MCS="/home/pi/sources/mono/mcs/class/
lib/monolite/gmcs.exe"
```

If you use apt-get install mono-gmcs use this:

```
sudo SKIP_AOT=true make
```

- Install the files

```
sudo SKIP_AOT=true make install
```

Uninstall

- Run the uninstall


```
sudo make uninstall
```
- remove the source files


```
rm -rf ~/sources/mono
```

Additional Information

Mono Compile from Git

<http://stackoverflow.com/questions/13365158/installing-mono-3-x-3-0-x-and-or-3-2-x>
<https://gist.github.com/pjvds/7185693>

5.3.2.4 Testing

These tests work for all types of installation.

First Test

If the installation is done just type the following command:

```
mono --version
cubie@Cubian:~$ mono --version
Mono JIT compiler version 3.2.8 (Debian 3.2.8+dfsg-7)
Copyright (C) 2002-2014 Novell, Inc, Xamarin Inc and Contributors. www.mono-
project.com
      TLS:          __thread
      SIGSEGV:      normal
      Notifications: epoll
      Architecture: armel,vfp+hard
      Disabled:    none
      Misc:         softdebug
      LLVM:         supported, not enabled.
      GC:           sgen
```

Ensure that you find the verb "hard" in the Architecture line.

If you get the hard info in the output you are done with the mono installation.

The version should match your installed / compiled version.

Check Hard Float Unit

There where some hard float issues in the past releases of Mono. You can simple check if it works with the csharp shell.

```
cubie@Cubian ~ $ date  
Sun Mar 16 13:12:07 UTC 2014  
cubie@Cubian ~ $ csharp  
Mono C# Shell, type "help;" for help
```

```
Enter statements below.  
csharp> DateTime.Now;  
03/16/2014 13:12:39  
csharp>
```

You must see the correct datetime. Otherwise your mono is damaged.

Check WinForms

If you need to run WinForms applications just compile a simple demo with a TextBox and ComboBox. You will receive errors when you start that application under Mono. If the application starts fine your Mono is ok.

Hint

It could be important to run the application with sudo. Just check that.

5.4 Lubuntu

Lubuntu is a fast and lightweight operating system developed by a community of Free and Open Source enthusiasts. The core of the system is based on Linux and Ubuntu.



System

x86 / X64

Webpage

<http://lubuntu.net/>

5.4.1 System Preparation

For this documentation we use Lubuntu (<http://lubuntu.net/>). It should be possible to use this topic for other Debian based systems as well like Ubuntu, Kubuntu, ...

Download

You can download 32 or 64 Bit images from the Lubuntu website:

<https://help.ubuntu.com/community/Lubuntu/GetLubuntu>

System Installation

There are tons of information in the internet how to install Lubuntu. We won't cover the details here.

<https://help.ubuntu.com/community/Lubuntu/InstallingLubuntu>

System Preparation

- Install SSH
`sudo apt-get install openssh-server`
- Ensure that you use English locales until LogView Studio runs!

From this point it would be ok to use [SSH to connect to Lubuntu](#) only.

Working on the machine directly would be fine, too.

Software Install / Update

- Update the software package lists
`sudo apt-get update`
- Update the installed software packages (this may take a while)
`sudo apt-get dist-upgrade`
- Install some additional useful packages (maybe some are preinstalled)
`sudo apt-get install sudoku htop git-core build-essential autoconf automake screen curl binutils`
(It's always a good idea to have [Sudoku](#) installed :-)

Hints

General Information about Debian software installation / updates

<https://help.ubuntu.com/community/AptGet/Howto>

5.4.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common Language Runtime.

<http://www.mono-project.com>

How to get Mono?

There are at least two different ways for getting Mono installed on your Cubietruck. All

of them have their pros and cons:

Compile it from a TAR package

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
- o mostly newer than the Raspbian package but older than the GIT version
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

Compile it from the GIT repository

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- needs some extra packages for compiling

Which method is the best?

For x86/x64 the best choice is to compile it from GIT because it takes not really much time.

Note

If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.

5.4.2.1 Install Mono

This will install the Mono package from the (L)Ubuntu repository.

Install

Installing Mono is a simple task. Just use the following command:

```
sudo apt-get install mono-complete libgdiplus
```

If you want to try C# development as well just add `monodevelop`, too.

```
sudo apt-get install mono-complete libgdiplus monodevelop
```

The installation will take some minutes.

Uninstall

Uninstalling is as simple as installing:

```
sudo apt-get autoremove monodevelope (only if you installed it)
```

```
sudo apt-get autoremove libgdiplus
```

Hint

libgdiplus removes mono-complete as well.

5.4.2.2 Compile Mono from TAR

This will compile Mono from a TAR package.

Procedure

- Install additional needed packages
`sudo apt-get install libtool libglib2.0-dev libxrender-dev libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev libexif-dev gettext libcairo2-dev`
- Install libgdiplus
`sudo apt-get install libgdiplus`
(You can also compile the latest version. See 'Compile Mono from GIT' for details.)
- Create a new folder where we place the files. You can use your home directory for that
`cd ~`
`mkdir sources`
`cd sources`
- Check for the latest TAR version
`wget -qO- http://download.mono-project.com/sources/mono/ |grep -o -E 'href="([^\"]+)"' | cut -d '"' -f2 | sort | uniq`
or just watch at the website
<http://origin-download.mono-project.com/sources/mono/>
<http://download.mono-project.com/sources/mono/>
- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)
`wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2`
- Extract the TAR file
`tar -jxvf mono-3.6.0.tar.bz2`
- Enter the new directory
`cd mono-3.6.0`
- run configure
`./configure --prefix=/usr/local/`
`/usr/local/` is the installation path of the compiled files
- run make - compile all the files
`sudo make`
- run make install - this will install the needed files into your system (this will take ~10 minutes)
`sudo make install`

Uninstall

- Run the uninstall
`sudo make uninstall`
- remove the source files
`rm -rf ~/sources/mono`

Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv
<https://gist.github.com/pjvds/7185693>

5.4.2.3 Compile Mono from GIT

This will compile Mono from the GIT repository.

Hint

Since this is updated very often it could include some bugs. But in most cases it works great.

Procedure

- Install additional needed packages

```
sudo apt-get install libtool libglib2.0-dev libxrender-dev  
libfontconfig1-dev libpng12-dev libgif-dev libjpeg8-dev libtiff5-dev  
libexif-dev gettext libcairo2-dev
```
- Create a new folder where we place the files. You can use your home directory for that

```
cd ~  
mkdir sources  
cd sources
```

libgdiplus Compilation

- Get the libgdiplus sources from GIT

```
git clone git://github.com/mono/libgdiplus
```
- Enter the new directory

```
cd libgdiplus
```
- Run autogen

```
./autogen.sh --prefix=/usr
```
- Compile the sources

```
sudo make
```
- Install the files

```
sudo make install
```
- Leave the directory

```
cd ..
```

Mono Compilation

- Get the Mono sources from GIT

```
git clone git://github.com/mono/mono.git
```
- Enter the new directory

```
cd mono
```
- Run autogen

```
./autogen.sh --prefix=/usr/local
```
- Install and configure gmcs

```
sudo make get-monolite-latest  
cd mcs/class/lib/monolite  
cp basic.exe gmcs.exe  
cd ~/sources/mono
```

Hint

You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.

- Compile the sources (~15 minutes)

If you used sudo make get-monolite-latest use this:

```
sudo make EXTERNAL_MCS="~/sources/mono/mcs/class/lib/monolite/gmcs.exe"
```

If you use apt-get install mono-gmcs use this:

```
sudo make
```

- Install the files

```
sudo make install
```

Uninstall

- Run the uninstall

```
sudo make uninstall
```

- remove the source files

```
rm -rf ~/sources/mono
```

Additional Information

Mono Compile from Git

<http://stackoverflow.com/questions/13365158/installing-mono-3-x-3-0-x-and-or-3-2-x>

<https://gist.github.com/pjvds/7185693>

5.5 CentOS

The CentOS Linux distribution is a stable, predictable, manageable and reproducible platform derived from the sources of Red Hat Enterprise Linux (RHEL)



System

x86 / X64

Webpage

<http://www.centos.org/>

5.5.1 System Preparation

For this documentation we use CentOS 6.5 (<http://www.centos.org/>). It should be possible to use this topic for other Redhat based systems as well.

Download

You can download 32/64 Bit images from the CentOS website:
<http://www.centos.org/download/>

System Installation

There are tons of information in the internet how to install CentOS. We won't cover the details here.

<http://wiki.centos.org/>

<http://www.centos.org/docs/>

System Preparation

Ensure that you use English locals until LogView Studio runs!

From this point it would be Ok to use [SSH to connect to Lubuntu](#) only.

Working on the machine directly would be fine, too.

Software Install / Update

Command may need `su` - for admin rights.

if the network is down use

`ip addr`
`dhclient ethXX` (XX for your interface)

- Install nano (give vi a kick ...)
`yum install nano`
- Update the installed software packages (this may take a while)
`yum update`
- Install some additional useful packages (maybe some are preinstalled)
`yum install gnome-sudoku git-core gcc* gcc-c++ kernel-devel make autoconf automake screen curl binutils wget unzip bzip2`
(It's always a good idea to have [Sudoku](#) installed :-)

Hints

General Information about software installation / updates with yum

https://www.centos.org/docs/5/html/5.1/Deployment_Guide/s1-yum-useful-commands.html

5.5.2 Mono

Mono is a software platform designed to allow developers to easily create cross platform applications. Sponsored by Xamarin, Mono is an open source implementation of Microsoft's .NET Framework based on the ECMA standards for C# and the Common

Language Runtime.

<http://www.mono-project.com>

How to get Mono?

Hint

CentOS has no package for Mono. You must compile it by yourself.

There are at least two different ways for getting Mono installed on your Cubietruck. All of them have their pros and cons:

Compile it from a TAR package

- + works with LogView Studio (LvCmd)
- + sometimes the GIT version has compilation errors which the TAR version don't have
- o mostly newer than the Raspbian package but older than the GIT version
- needs some extra packages for compiling
- WinForms applications produce errors with some components (like a Textbox)

Compile it from the GIT repository

- + works with LogView Studio
- + WinForms applications work from Mono 3.8.1
- + always the newest release
- needs some extra packages for compiling

Which method is the best?

For x86/x64 the best choice is to compile it from GIT because it takes not really much time.

Note

If the TAR package gets to a 3.8 version this would also be Ok. At the moment of writing this text there is only a 3.6.0 version available.

5.5.2.1 Compile Mono from TAR

This will compile Mono from a TAR package. **It takes about 4 hours in sum.**

Procedure

- Get admin rights
`su -`
- Preparation for installation ...
`yum install glib2-devel libpng-devel libjpeg-devel giflib-devel libtiff-devel libX11-devel fontconfig-devel bison gettext libtool cairo-devel`
- Create a new folder where we place the files. You can use the tmp directory for that
`mkdir /tmp/sources`
`cd /tmp/sources`

libgdiplus Compilation

- Get the libgdiplus sources from GIT

```
git clone git://github.com/mono/libgdiplus
```

- Enter the new directory
`cd libgdiplus`

- Run autogen

```
./autogen.sh --prefix=/usr
```

- Compile the sources

```
make
```

Solving "libtool: Version mismatch error"

```
autoreconf -ivf
```

- Install the files

```
make install
```

- Leave the directory

```
cd ..
```

Mono Compilation

- Check for the latest TAR version

```
wget -qO- http://download.mono-project.com/sources/mono/ |grep -o -E 'href="([^\"]+)"' | cut -d '"' -f2 | sort | uniq
```

or just watch at the website

<http://origin-download.mono-project.com/sources/mono/>
<http://download.mono-project.com/sources/mono/>

- Download the latest release (at the moment of writing it is mono-3.6.0.tar.bz2)

```
wget http://download.mono-project.com/sources/mono/mono-3.6.0.tar.bz2
```

- Extract the TAR file

```
tar -jxvf mono-3.6.0.tar.bz2
```

- Enter the new directory

```
cd mono-3.6.0
```

- run configure

```
./configure --prefix=/usr/local/
```

/usr/local/ is the installation path of the compiled files

- run make - compile all the files

```
sudo make
```

- run make install - this will install the needed files into your system (this will take ~10 minutes)

```
sudo make install
```

Uninstall

- Run the uninstall

```
sudo make uninstall
```

- remove the source files

```
rm -rf ~/sources/mono
```

Additional Information

Notes about compiling Mono 3.2.3 from a TAR Archiv

<https://gist.github.com/pjvds/7185693>

5.5.2.2 Compile Mono from GIT

CentOS has no package for Mono. This will compile Mono from the GIT repository.

Hint

Since this is updated very often it could include some bugs. But in most cases it works great.

Procedure

- Get admin rights
`su -`
- Preparation for installation ...
`yum install glib2-devel libpng-devel libjpeg-devel giflib-devel libtiff-devel libX11-devel fontconfig-devel bison gettext libtool cairo-devel`
- Create a new folder where we place the files. You can use the tmp directory for that
`mkdir /tmp/sources`
`cd /tmp/sources`

libgdiplus Compilation

- Get the libgdiplus sources from GIT
`git clone git://github.com/mono/libgdiplus`
- Enter the new directory
`cd libgdiplus`
- Run autogen
`./autogen.sh --prefix=/usr`
- Compile the sources
`make`
Solving "libtool: Version mismatch error"
`autoreconf -ivf`
- Install the files
`make install`
- Leave the directory
`cd ..`

Mono Compilation

- Get the Mono sources from GIT
`git clone git://github.com/mono/mono.git`
- Enter the new directory
`cd mono`
- Run autogen
`./autogen.sh --prefix=/usr/local`
- Install and configure gmcs
`sudo make get-monolite-latest`
`cd mcs/class/lib/monolite`
`cp basic.exe gmcs.exe`
`cd ../../..`

Hint

You can also use `apt-get install mono-gmcs` for the 4 gmcs steps.

- Compile the sources (15 minutes)

If you used sudo make get-monolite-latest use this:

```
sudo make EXTERNAL_MCS="/tmp/sources/mono/mcs/class/lib/monolite/
gmcs.exe"
```

If you use apt-get install mono-gmcs use this:

```
sudo make
```

- Install the files

```
sudo make install
```

Uninstall

- Run the uninstall

```
sudo make uninstall
```

- remove the source files

```
rm -rf ~/sources/mono
```

Additional Information

Installing Mod_Mono and Xsp4 on CentOS 6.3

<http://stackoverflow.com/questions/14901271/installing-mod-mono-and-xsp4-on-centos-6-3>

<http://wiki.phonicuk.com/Installing-Mono-in-CentOS-5-x.ashx>

6 Command Line Tool LvCmd

LogView Studio comes with a Command line tool called LvCmd. This tool will run on Windows (uses .Net 4.0) and Linux like Raspberry Pi, Cubietruck, Linux x32, Linux x64 with the mono framework. Take a look at the previous [Linux Prepare](#) chapter how to get your Linux ready for running mono applications.

LvCmd can be used to run LogView Studio without a graphical environment. The following text will guide you through the steps of installing and using this utility.

Windows Installation

LvCmd is part of the normal LogView Studio installation. Normally it's located in the Program Files folder (..\LogView Studio).

Anyway you can download the ZIP file to a Windows system unpack and use it.

Linux Installation

Since there is no dedicated installer for Linux systems you have to do it manually. Just download the latest ZIP file for example with wget. Unpack the ZIP file and you are done.

Download File : <http://www.logview.info/lvstudio/LVStudio.zip>

Sample from Raspberry Pi:

```
pi@rasp1 ~ $ mkdir lvstudio
pi@rasp1 ~ $ cd lvstudio/
pi@rasp1 ~/lvstudio $ wget http://www.logview.info/lvstudio/LVStudio.zip
--2014-03-18 15:02:38--  http://www.logview.info/lvstudio/LVStudio.zip
Resolving www.logview.info (www.logview.info)... 85.13.139.189
Connecting to www.logview.info (www.logview.info)|85.13.139.189|:80...
connected.

HTTP request sent, awaiting response... 200 OK
Length: 18710468 (18M) [application/zip]
Saving to: `LVStudio.zip'

100%
[=====>] 18,710,468 4.16M/s in 7.7s

2014-03-18 15:02:46 (2.31 MB/s) - `LVStudio.zip' saved
[18710468/18710468]
```

```
pi@rasp1 ~/lvstudio $ unzip LVStudio.zip
Archive: LVStudio.zip
      FILE EXTRACT .....
pi@rasp1 ~/lvstudio $
```

That's it. Only **4 steps** and you are done with the "installation".

LvCmd Usage

Using LvCmd is a straight forward task. Create a project, add a device, configure the device and run the device. The next example is on a Windows 64Bit machine. Keep in mind that the usage is exactly the same on Linux and Windows. The only difference is that you start LvCmd on Linux with the prefix mono.

Create a project:

Just run LvCmd <projectpath> and you are done. Assume we will create our project in d:\data\project we would call LvCmd like this:

```
LvCmd "d:\data\project"
```

on Linux we would prefix it with mono:

```
mono LvCmd.exe "\data\project"
```

Hints:

If you start LvCmd the first time it is possible that there are no database executables available. In that case LvCmd will automatically detect your system and download the necessary files from the LogView web page.

If the project is created you will get a command prompt like this:

Type "h" or "?" for Help. Enter for Command prompt.

```
CMD >
```

Add a Device

Insert **pa** as a command (if you miss the command prompt just hit enter):

Insert the Device Id which you want to add to your project:

```
Insert Device ID : 5
Main     1      V 15:16:36.2168395 : Insert into Project
Main     1      V 15:16:36.2643455 : Create Table : d5_ch1_20140318_151636
Main     1      15:16:36.2648456 : SQL Statement
Main     1      V 15:16:36.2888487 : Create Channel for Device
```

Configure the Device

Press enter to get the command prompt back. Enter **dc** for Device Configuration.

```
CMD > dc
```

```
-----  
Select a Device from the Project  
-----
```

```
Device : 2 - Ultramat 18          (DevId :5)
```

```
Insert Device ID [<Enter> for Exit] : 2
```

```
Device 1      V 15:18:01.6581892 : Device found !
```

```
-----  
Select the Source for the Device  
-----
```

```
1 - DsCom.DsSerial
```

```
2 - DsCom.DsSerialLinux

Insert Source ID [<Enter> for Exit] : 1

Main    1    V 15:18:02.8828447 : [0] Type : [COM1] COM1 (Communications
Port)

Main    1    V 15:18:02.8833448 : [1] Type : [COM3] COM3 (Intel(R) Active
Management Technology - SOL)

Main    1    V 15:18:02.8833448 : [2] Type : [COM4] COM4 (Sierra Wireless
NMEA Port)

Main    1    V 15:18:02.8833448 : [3] Type : [COM5] COM5 (Sierra Wireless
DM Port)

Main    1    V 15:18:02.8838448 : [4] Type : [COM20] COM20 (Virtual
Serial Port (Eltima Software))

Main    1    V 15:18:02.8838448 : [5] Type : [COM21] COM21 (Virtual
Serial Port (Eltima Software))

Select Port : 1
```

Start the Device

Press enter to get the command prompt back. Enter [da](#) for Device Start.

```
CMD > da
```

```
-----  
Start the Device : Ultramat 18  
-----
```

```
Project 1    > 15:19:15.5915775 : ..HandleRawDatasets (setlast : True)
Project 1    V 15:19:15.5925777 : ..Process Channel : 1
Project 1    V 15:19:15.6020789 : ..Found 0 RawDatasets in ProjectBase
Table
Project 1    V 15:19:15.6125802 : ..Found 0 Datasets (Break Set) in Table
d5_ch1_20140318_151636
Project 1    < 15:19:15.6130803 : ..HandleRawDatasets
```

Stop the Device

Press enter to get the command prompt back. Enter [do](#) for Device Stop.

Basically that's all you have to do. If you want to see some data you have to use a web browser. The next chapter will cover this topic.

6.1 Webserver

If you start LvCmd a small embedded web server is started as well. You can see it after the LvCmd start:

```
Web      1      V 15:14:32.6811525 : Webserver Port 8080 available = True
Web      1      V 15:14:32.6811525 : Starting embedded LogView Studio
Webserver ...
Web      1      V 15:14:32.6821526 : Webserver IP Addresses [PDBDSCHN3] ...
Web      1      V 15:14:32.6826527 :   IP V4 : 192.168.30.139,
192.168.232.1, 192.168.130.1
```

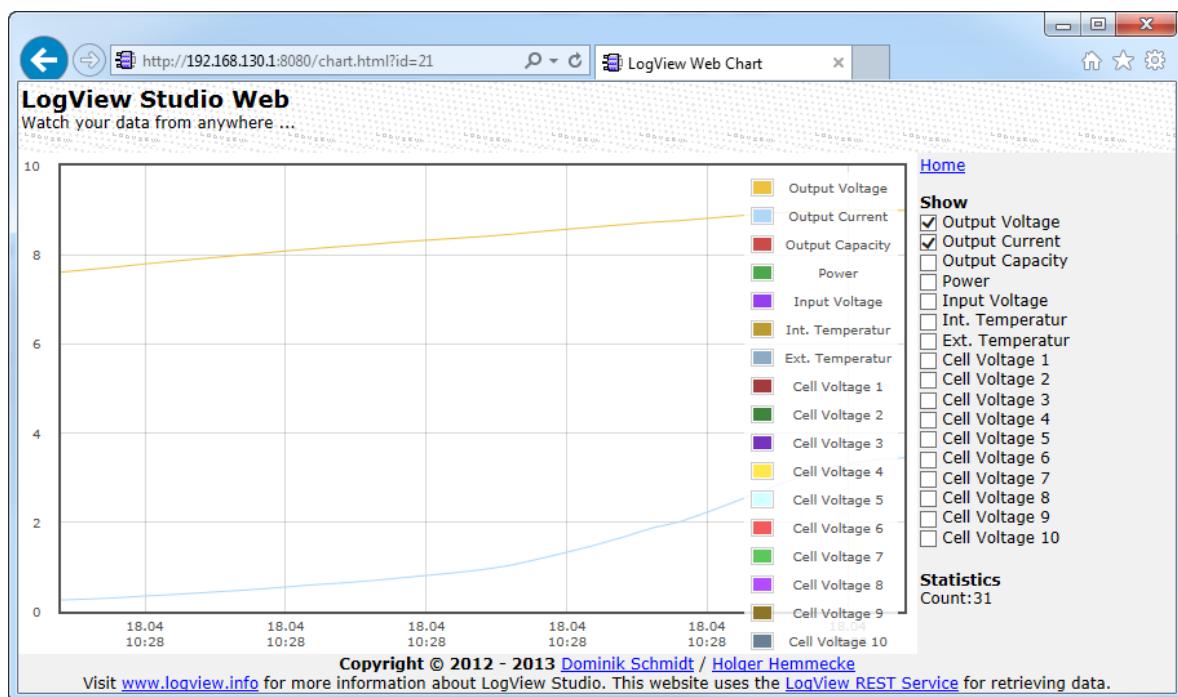
So pick up one of the IP addresses and enter it in your web browser like this: <http://<IP>:8080>



If all works well you will get a list of available datasets. Click on a dataset und you will get a chart view:

10	2014-3-18 11:29:38	NiCD - Charging
11	2014-3-18 11:31:34	NiCD - Charging
12	2014-3-18 11:37:39	NiCD - Charging
13	2014-3-18 11:41:50	NiCD - Charging

Result:



At the moment the website is very basic. But it will get more features in the next releases.

7 Development

In this chapter you will find information about LogView Studio development.

You will find information how to ...

- use the new [OpenFormat Zero](#)
- use the "old" [OpenFormat which uses an INI file](#)
- *write your own Device library*
- *use the REST API*

(*italic* topics will be introduced later)

7.1 OpenFormat Zero

OpenFormat Zero is used to send data from a device to LogView Studio. Instead of using a configuration file (like the old OpenFormat) OpenFormat Zero gets the configuration from the device. The handling of your device can't be easier:

- Connect your device to your computer (or Network)
- Add a OpenFormat Zero Device in LogView Studio

- Select the correct Source for your Device and configure it
- Start your device
- Start the recording in LogView Studio

At the moment OpenFormat Zero is limited to one Channel. This will be enhanced in a future release.

OpenFormat Zero can use different Sources. At the moment there is support for:

- TCP Client (your device must realize a TCP server)
- UDP Receiver
- Serial Port (Default is 57600 Baud)
- File Import

The next chapter will explain the OpenFormat Zero in detail and give two examples on Arduino bases.

7.1.1 Description

If you use OpenFormat Zero there are some points you have to know in order to get it working with LogView Studio.

Line Start / End

Every data lines (including the headers) start with \$ and ends with \r\n (0Dh, 0Ah).

Every dataset has it's header

Every dataset in OpenFormat Zero starts with it's own header. The smallest header is a simple dataset name:

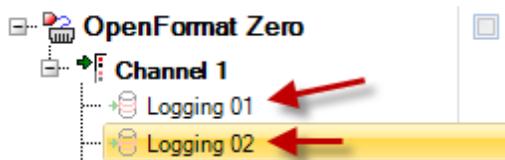
\$N\$;MyDatasetName

A dataset is a block of data which has it's own Entry in LogView Studio.

Example:

```
$N$ ;Logging 01
$5.1;2.3
$6.1;3.3
$N$ ;Logging 02
$15.1;2.13
$61.1;3.23
```

Result in LogView Studio:



Decimal separator

You can use . (dot) or , (comma) as decimal separator. LogView Studio will convert it automatically to the correct value which is used in your system.

Be aware of sending data with 1000th separator like this:

1.500,45

Send values without the separator:

1500,45

Default Time

OpenFormat Zero uses the time when a package is received as time stamp for the data as Default.

Take a look at the [Time format](#) chapter for more information.

Column, Data, Factor, Offset and SumOffset count must be the same

If you send a header followed by data you can transmit some information where the count must match.

First of all your Column count must be the same as your Data count.

If you send additional Factor, Offset or SumOffset values the count must also be the same as your Column / Data count.

Good Example:

```
$C$;Temp 1 [°C,T];Voltage[V]
$F$;1.2;1.0
$5.1;2.3
$6.1;3.3
```

Bad Example:

```
$C$;Temp 1 [°C,T];Voltage[V]
$F$;1.2;1.0;2.0
$5.1;2.3
$6.1;3.3;5.0
```

Two lines have different item counts !

Watch the samples

Things will get more clear if you take a look at the attached samples.

7.1.2 Header

OpenFormat Zero supports different Header information to setup the data handling in LogView Studio. As a minimum you must send the \$N\$ and \$C\$ header because LogView Studio uses it for a dataset change detection and Database enhancement. All other header are optional. As mentioned in last chapter all header lines have to end with CR LF (\r\n).

You can use all command with an additional channel setting (<CHN>) to use the OpenFormat Zero in a multi Channel scenario.

Please take a look at the chapter [Multi Channel Usage](#) to get a better understanding how this works.

If you don't send a channel information LogView Studio will always use Channel 1 as a default.

General Information for the Header Settings

Header Settings are valid as long as you don't change them. You can set the Columns once and send multiple datasets. All datasets will use the same Columns. Same for Factor, Offsets and Timesettings.

\$N<CHN>\$;DatasetName

Set the Dataset Name in LogView Studio and indicates the start of a new dataset.

If you only send \$N\$; LogView Studio will use a Default dataset name.

Note: This header is mandatory for a new dataset!

\$C<CHN>\$;Temp 1 [°C,T];Voltage[V];Current

Set the columns which will be received in the data part. Each column is separated by a ; (semicolon).

As you can see the Column name can be supplemented by one or two values in [].

The first value is the Unit and the second value is the Symbol. You can also use only one additional value. In that case you have to transmit the Unit.

It's also possible to send only the Column name.

Note: The Column count must match the data item count !

\$V<CHN>\$;VALUE

You can set the value format for the data lines. Valid Values are HEX and DOUBLE.

If you don't set this value the Default is DOUBLE.

\$L<CHN>\$;Level;LogText

You can send Log Messages to LogView Studio. They appear in the LogView Logging system. You can use this feature to send some Debug information.

Level can be set to the following values:

E = Error

W = Warning

V = Verbose

If you only set the LogText the Default Level is Verbose.

\$F<CHN>\$;Factor1;Factor2;....

\$O<CHN>\$;Offset1;Offset2;....

\$S<CHN>\$;SumOffset1;SumOffset2;....

LogView can calculate the value based on three values : Factor, Offset, Sum Offset.

The formula for this is : **y = (x + t1) * m + t2**

y is the resulting value which will be stored in LogView Studio.

x is the value from your Device.

m is the Factor

t1 is the Offset

t2 is the Sum Offset

Note : Factor, Offset and SumOffset must have the same count as the data items count!

\$R<CHN>\$

Reset the Time Settings to Time Mode Receivedate.

\$T<CHN>\$;1;yyMMdd

Set the Time Settings to Time Mode TimeGiven.

The first parameter (1) indicates the Time position in the data line (value starts with Index 0 !)

The second parameter sets the given DateTime Format.

\$I<CHN>\$;1000;28042013_120005123

Set the Time Setting to Time Mode Increment.

The first parameter is the increment in Milliseconds.

The second parameter is optional and indicates the Start Date Time. The Format is : DDMMYYYY_HHMMSSFFF (FFF = Milliseconds).

Note : See [Time Formats](#) for more details.

7.1.3 Time Formats

OpenFormat Zero supports different Time formats to get the best support for your specific device. The following Time Modes are available and will be set automatically through your Time Settings:

Receive Date \$R\$

This is the Default Time Format which will be used if you send %R% or nothing.

The date value for the data package will be the receive time of the raw data. This mode is well suited for Serial and Network connections.

Importing a file with this Time Mode is possible but will result in strange views. The data import is very fast in LogView and the time span for the data will be very slow. If possible use Increment or Time Given for File Import.

Increment \$I\$;1000

Use this Time Mode if you have fixed time intervals - like a charger which sends every second a value.

But keep in mind that there is no synchronisation between the received raw data time and the calculated incremented time.

Check if you can't use the Receive Date Format instead.

This Mode will start at 01.01.0000 00:00:00 because no Start Time is given.

Increment Start Time \$I\$;1000;28042013_120005123

This increment Mode is the same as the previous mode.

The difference is the given Start Time.

28042013_120005123 = 28. April 2013 12:00:05 123ms

Time Given \$T\$;1;yyMMdd

This Time Mode gets the Time for LogView from the Device data.

All data packages from your device must contain a Date / Time value. The position is given at the first parameter.

The second parameter indicated the DateTime Format. Use the following link for more

details on this topic:

[http://msdn.microsoft.com/en-us/library/vstudio/8kb3ddd4\(v=vs.100\)](http://msdn.microsoft.com/en-us/library/vstudio/8kb3ddd4(v=vs.100))

Also see the File Import Sample how to use this feature.

Usage Examples

Receive Date can always be used if you receive data from a Serial or Network device.

Time Given can be used if your data contains any Date Time information regardless of the used Source.

Increment can be used in situations where you import data from a file and no Date Time value is given.

7.1.4 Data

Each dataset contains some header information and 1-n lines of data. The data lines represent the values you want to store in LogView Studio.

Some general notes for data lines:

- Must start with \$
- Must end with CR LF (\r\n)
- If you add a time value ensure that it is not in HEX format
- The data item count must be the same as the column name count
- You can send the values in HEX format if you set `%v% ;HEX` in the header
- The Decimal separator can be . or ,
LogView Studio will transfer it automatically to the correct system separator.
- Don't add a thousand separator
- Values are separated by a ; sign

Sample data lines:

```
$1;2;3\r\n
$1.2;2.3;3.44\r\n
```

Invalid data lines:

1;2;3\r\n	No \$ at the beginning
\$1;2;3\r	No \n at the end
\$1:2:3\r\n	wrong separator
\$1.000,3;2;3\r\n	thousand separator added

7.1.5 Multi Channel Usage

TBD

7.1.6 Sample : File Import

1. Incremental Fileimport

```
$N$;My Data
$C$;Temp 1 [°C,T];Voltage[V]
$I$;1000
$5.1;2.3
$6.1;3.3
$L$;My Logtext
$7.1;4.3
$6.1;5.3
$5.1;6.3
```

Description

- No Channel is given so LogView Studio will use Channel 1 as Default
- Dataset name is "My Data"
- Two columns will be created in LogView Studio : "Temp 1" and "Voltage"
- Temp 1 has a given Unit "°C" and a Symbol "T"
- Voltage has a given Unit "V"
- Time for the Data will be incremented by 1000 Milliseconds for each data line
- Time will start at 01.01.XXXX 00:00:00 (XXXX = actual year)
- Values are send in Double format (Default)
- A verbose Logtext will be added to the LogView Studio Logging system called "My Logtext"

2. Timegiven HEX Fileimport

```
$N$;My Data 2
$C$;Temp 1 [°C,T];Voltage[V]
$T$;0;ddHHmmss
$V$;HEX
$28160001;A;CC
$28160002;1A;C5
$L$;My Logtext 2
$28160005;2B;C7
$28160007;5F;C0
$28160010;FFF;CF
```

Description

- No Channel is given so LogView Studio will use Channel 1 as Default
- Dataset name is "My Data 2"
- Two columns will be created in LogView Studio : "Temp 1" and "Voltage"
- Temp 1 has a given Unit " $^{\circ}\text{C}$ " and a Symbol "T"
- Voltage has a given Unit "V"
- Time is given in data lines at position 0
- Time Format is ddHHmmss (Day Hour Minute Second)
You can find sample here : <http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx>
- Values are send in HEX Format (\$V\$ Header)
- A verbose Logtext will be added to the LogView Studio Logging system called "My Logtext 2"
- All values are in HEX Format and represent integer values
You can convert the integer values to double by using the Offsets and the Factor.

3. Timegiven Fileimport with Offsets and Factor

```
$N$;My Data 3
$C$;Temp 1 [  $^{\circ}\text{C}$ ,T];Voltage[V,U]
$T$;1;ddHHmmss
$F$;2;1.5
$V$;DOUBLE
$5.1;28160001;2.3
$6.1;28160002;3.3
$7.1;28160003;4.3
$6.1;28160004;5.3
$5.1;28160005;6.3
```

Description

- No Channel is given so LogView Studio will use Channel 1 as Default
- Dataset name is "My Data 3"
- Two columns will be created in LogView Studio : "Temp 1" and "Voltage"
- Temp 1 has a given Unit " $^{\circ}\text{C}$ " and a Symbol "T"
- Voltage has a given Unit "V" and a Symbol "U"
- Time is given in data lines at position 1
- Time Format is ddHHmmss (Day Hour Minute Second)
You can find sample here : <http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx>
- Values are send in Double format (Default)
- The default factor (1) will be replaced by 2 (for data item 1) and 1.5 (for data item 2)
The resulting values for the first data line:
 $5.1 * 2 = \mathbf{10.2}$

2.3 * 1.5 = 3.45

Note : The time column is ignored in the Factor header line !

- Offset and Offset Sum use the Default value 0

Note : \$C\$ and \$F\$ have the same item count !

4. Incremental Fileimport with multiple channels

```
$N1$;My Channel1 Data
$C1$;Temp 1[°C,T];Voltage[V]
$I1$;1000
$1$5.1;2.3
$L1$;My Logtext
$1$6.1;3.3
$N3$;My Channel3 Data
$C3$;Cell 1[V];Cell 2[V];Cell 3[V]
$I3$;1000
$1$7.1;4.3
$3$3.5;3.6;3.65
$1$6.1;5.3
$3$3.6;3.62;3.66
$1$5.1;6.3
$3$3.62;3.6;3.64
$3$3.5;3.6;3.6
```

Description

- Channel 1 & 3 will be used
- Channel 1
 - Dataset name is "My Channel1 Data"
 - Two columns will be created in LogView Studio : "Temp 1" and "Voltage"
 - Temp 1 has a given Unit "°C" and a Symbol "T"
 - Voltage has a given Unit "V"
 - Time for the Data will be incremented by 1000 Milliseconds for each data line
 - Time will start at 01.01.XXXX 00:00:00 (XXXX = actual year)
 - Values are send in Double format (Default)
 - A verbose Logtext will be added to the LogView Studio Logging system called "My Logtext"
- Channel 3
 - Dataset name is "My Channel3 Data"
 - Three columns will be created in LogView Studio : "Cell 1", "Cell 2", "Cell 3"
 - All have a given Unit "V"
 - Time for the Data will be incremented by 1000 Milliseconds for each data line

- Time will start at 01.01.XXXX 00:00:00 (XXXX = actual year)
- Values are send in Double format (Default)

Note : You don't have to set the Channel information for Channel 1 explicitly. If you don't send the channel number it will be treated as channel 1.

7.1.7 Sample : Arduino TCP

```
#include <SPI.h>
#include <Ethernet.h>
#include <OneWire.h>

// TCP Server
// http://forum.arduino.cc/index.php?topic=123756.0

// the media access control (ethernet hardware) address for the shield:
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
//the IP address for the shield:
byte ip[] = {
  192, 168, 30, 177 };
// the subnet:
byte subnet[] = {
  255, 255, 255, 0 };

// Use TCP Port 12345 as Default
EthernetServer server = EthernetServer(12345);

// DS1820 Variablen
byte i;
byte present = 0;
byte data[12];
byte addr[8];

boolean headersend = false;

// DS18S20 Temperature chip I/O
OneWire ds(2); // on pin 2
```

```
void setup()
{
    Serial.begin(57600);

    Serial.println("LogView Studio - OpenFormat Zero Demo (TCP Server for Data
sending)");
    Serial.println("2013 by Dominik Schmidt\n");
    Serial.println("This demo uses a DS1820 temperature sensor on Pin 2 for data.");

    Serial.println("Init DS1820 Sensor");
    if ( !ds.search(addr)) {
        Serial.println("No more addresses.\n");
        ds.reset_search();
        return;
    }

    Serial.print("R=");
    for( i = 0; i < 8; i++) {
        Serial.print(addr[i], HEX);
        Serial.print(" ");
    }
    Serial.println("");

    if ( OneWire::crc8( addr, 7) != addr[7]) {
        Serial.println("CRC is not valid!\n");
        return;
    }

    if ( addr[0] == 0x10) {
        Serial.println("Device is a DS18S20 family device.\n");
    }
    else if ( addr[0] == 0x28) {
        Serial.println("Device is a DS18B20 family device.\n");
    }
    else {
        Serial.print("Device family is not recognized: 0x");
    }
}
```

```
    Serial.println(addr[0], HEX);
    return;
}

// initialize the ethernet device
Ethernet.begin(mac, ip, subnet);

// start listening for clients
server.begin();
}

void loop()
{
    float temp;
    String tempst;

    // Check if a client is available
    // The Server only starts if LogView Studio sends some data.
    // So check für the 02h as startbyte.
    EthernetClient client = server.available();

    if (client == true)
    {
        Serial.println("Client Connected");

        // Check if the client is still connected
        while (client.connected())
        {
            // Has the client send Data ?
            if (client.available())
            {
                char c = client.read();
                switch (c)
                {
                    case 83: // "S" -> Start
                        Serial.println("Start detected");
                        // Send Header information

```

```
tempst = "$N$;Room Temp Logging\r\n";
server.print(tempst);
tempst = "$C$;Temp 1 [°C,T];Temp 2\r\n";
server.print(tempst);
headersend = true;
break;

case 69: // "E" -> End
Serial.println("End detected");
headersend = false;
break;
}

}

// Send data if header was send
if (headersend) {
    temp = GetTemp();
    char StrBuffer [10];
    tempst = dtostrf(temp, 6, 2, StrBuffer);
    Serial.println(tempst);
    tempst = "$" + tempst + ";2.0\r\n";
    server.print(tempst);
}
}

Serial.println("Client Disconnected");
}

}

// Read Temperature from DS1820
float GetTemp()
{
    int HighByte, LowByte, TReading, SignBit, Tc_100, Whole, Fract;
    float value;

    ds.reset();
    ds.select(addr);
    ds.write(0x44,1);           // start conversion, with parasite power on at the end
```

```
delay(1000);      // maybe 750ms is enough, maybe not
// we might do a ds.depower() here, but the reset will take care of it.

present = ds.reset();
ds.select(addr);
ds.write(0xBE);      // Read Scratchpad

//    Serial.print("P=");
//    Serial.print(present,HEX);
//    Serial.print(" ");
for ( i = 0; i < 9; i++) {           // we need 9 bytes
    data[i] = ds.read();
    //    Serial.print(data[i], HEX);
    //    Serial.print(" ");
}
//    Serial.print(" CRC=");
//    Serial.print( OneWire::crc8( data, 8), HEX);
//    Serial.println();

LowByte = data[0];
HighByte = data[1];
TReading = (HighByte << 8) + LowByte;
SignBit = TReading & 0x8000; // test most sig bit
if (SignBit) // negative
{
    TReading = (TReading ^ 0xffff) + 1; // 2's comp
}

Tc_100 = (TReading*100/2);

value = Tc_100 / 100;
if (Tc_100 % 100 == 50)
{
    value += 0.5;
}
if (SignBit)
{
```

```
    value = value * -1;  
}
```

```
    return value;  
}
```

7.1.8 Sample : Arduino Serial

```
boolean headersend = false;  
double counter = 0.0;  
  
int led = 13;  
  
void setup()  
{  
    Serial.begin(57600);  
  
    Serial.println("LogView Studio - OpenFormat Zero Demo (Serial Port for Data  
sending)");  
    Serial.println("2013 by Dominik Schmidt\n");  
  
    pinMode(led, OUTPUT);  
}  
  
void loop()  
{  
    String tempst;  
    double temp;  
  
    if (Serial.available() > 0)  
    {  
        char c = Serial.read();  
        switch (c)  
        {  
            case 83: // "S" -> Start  
                Serial.println("Start detected");  
                // Send Header information  
                tempst = "$N$;Data Logging\r\n";
```

```

        Serial.print(tempst);
        tempst = "$C$;Value 1;Value 2\r\n";
        Serial.print(tempst);
        headersend = true;
        break;

    case 69: // "E" -> End
        Serial.println("End detected");
        digitalWrite(led, LOW);
        headersend = false;
        break;
    }

}

if (headersend)
{
    digitalWrite(led, !digitalRead(led));
    counter += 0.01;
    if (counter > 6.28) {counter = 0.0;}

    Serial.print("$");
    temp = sin(counter);
    Serial.print(temp);
    Serial.print(";");
    temp = sin(cos(counter) * tan(counter / 2));
    Serial.print(temp);
    Serial.print("\r\n");

    delay(250);
}
}

```

7.2 OpenFormat INI

OpenFormat is the old developer format which was introduced with LogView V2. For compatibility issues the format is supported by LogView Studio, too.

You should use [OpenFormat Zero](#) for new implementations!

OpenFormat can use different Sources. At the moment there is support for:

- TCP Client (your device must realize a TCP server)
- Serial Port (Default is 9600 Baud)
- File Import

7.2.1 Description

Format Overview

The OpenFormat is designed to enable developers to interface their own hardware with LogView via a configurable serial data protocol. To use the OpenFormat (short name: OF) you have to create an INI file and an optional picture for your device. The INI file contains all the needed configuration information for LogView to display your serial data.

The format OF is pure ASCII data where the items are separated by ";".

An OF packet starts with "\$", then has several values separated by ";" and always ends up with **CR + LF** (#13#10). An example of a packet your device may produce is:

```
$1;1;0;1133;318;0;0;0;0;20;21;0;16<cr><lf>
```

We will explain what you can make all the values do in a bit.

Device Settings

Name - The name of your device

Vendor - Your (company) name

Vendor Link - Company link [optional]

LogView Link - If we have a special info on our website [optional]

TimeStep - Time difference between two packets within one channel. This is a global setting for all channels.

Time given from device - If set, the time must be included in the packet and will be used by LogView. If not set, the time is calculated from TimeStep and counting your packets.

We will go into more detail later, see chapter 4c.

Value format - At the moment LogView only supports ASCII format.

Checksum - If you can make your device generate a checksum, LogView can check it. Again more detail later, see chapter 4e

none - Checksum isn't used. Checksum = 0 !

xor - Your device must use the XOR method for checksum calculations.

Port Settings

Baudrate - Device baud rate

Databits - Device data bits (7 / 8 / 9)

Stopbits - Device stop bits (1 / 1,5 / 2)

Flowcontrol - Flowcontrol (None / Xon,Xoff / Hardware)

State DTR - Controls the state of the DTR line

State RTS - Controls the state of the RTS line

Clustersize - Defines the size of the packet in Bytes.

Don't change this value if you don't know the exact value.

Negative values tell LogView that the size is variable.

DelayTime - These values control the receiving and sending of serial data and are related to the used baudrate.

Increase the values for RTOCharDelay if you have problems receiving your data correctly.

Useable values for xxxCharDelay are 200-2000.

WTxxxx controls the timeouts for sending. Normally you don't have to change this.

Packet parts

An average packet may look like the one shown below. Have a quick look as in the next few chapters we will explain it in detail.

```
$1;1;0;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>
.....-----> End Sign #13#10
.....----> Checksum (see chapter 4e)
.....-----> Data Stream (see chapter 4d)
....-> Time Stamp (see chapter 4c)
....-> States (see chapter 4b)
.-> Channel Number (see chapter 4a)
-> Start Sign $"
```

a. Channel Number

The OF can handle up to 8 separated channels with 40 data streams each. We class data streams as the values you want to display, such as voltage, current, etc. Channels are something most devices don't make use of. To explain say for instance you built a charger that can independantly change 3 batteries at once but you still only wanted one serial connection to your PC. Logview can cope with all of them at once by using channels and assigning each individual charger to a seperate channel.

In our example from chapter 1. the **bold** entry is the **channel number**.

```
$1;1;0;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>
```

b. States

```
[Stati]
```

```
StatiAnzahl = 1
```

```
001 = Flight logging
```

So in state 1 Logview will name the data set "Flight logging".

You have to define at least one state within your INI file. The states are used in LogView to build useful dataset names.

Some example state names might be : charging, discharging, balancing, Charging in Lipo Mode, ... etc.

States are only labels which can contain any ASCII character.

Again in our example the **bold** entry is the **state**.

```
$1;1;0;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>
```

c. Time Stamp

```
$1;1;;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>
```

```
....^^
```

The Timestamp (if given by your device) should be given in seconds. You can use fractions, for instance 200ms = 0.2

Always start at 0 !

Again in our example the **bold** entry is the **timestamp**.

```
$1;1;0;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>
```

If you don't want your device to give a timestamp you have to set "Time given from device" in "Device Settings" to off, then LogView will use the "TimeStep" value in "Device Settings" multiplied by the packet count to work out the passage of time.

If no timestamp is sent you have to send ; ; !

d. Data Stream

You have to let LogView know how to interpret each of the data items which are part of your packet. To explain this let's use the example from above:

```
$1;1;0;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>
```

The items in **bold** are the **data items**, this example has 10 seperate data items.

Each of the data items must be specified in the INI file. These are the settings used:

Name (Messgröße) - name of the value (for example Voltage)

Unit (Einheit) - unit (U for voltage)

Symbol - symbol (V for voltage)

Factor (Faktor) - floating value for the value calculation (see explanation) - default : 1,0

Pre Factor Offset (OffsetWert) - floating value for the value calculation (see explanation) - default : 0,0

Post Factor Offset (OffsetSumme) - floating value for the value calculation (see explanation) - default : 0,0

If you want LogView to display the values from your device verbatim you don't have to change the last 3 settings.

If you want to add an offset or a scaling factor to the sent value you can specify it with "Factor", "Pre Factor Offset" and "Post Factor Offset".

LogView calculates the resulting value with this formula:

```
y = ( x + [Pre Factor Offset] ) * [Factor] + [Post Factor Offset]

y:           displayed value
x:           input value, sent in data packet
Pre Factor Offset:   OffsetWert
Factor:         Faktor
Post Factor Offset: OffsetSumme
```

e. Checksum

You can use XOR and AND as Checksum calculation.

Example code for XOR Calculation (Pascal / Delphi Code !!):

```
CheckCalc := 0;
Data := '$1;1;0;1133;318;0;0;0;0;0;20;21;0;' // see Example above !
For _i := 1 to Length(Data) do begin
  CheckCalc := CheckCalc XOR Byte(Data[_i]);
end;
```

This means LogView takes the data bytes until the last ";" for the checksum calculation.

\$1;1;0;1133;318;0;0;0;0;0;20;21;0;16<cr><lf>

Data used for the checksum calculation

7.2.2 Sample INI

[Geraet]	
Name	= Tiny Logger
Hersteller	= Schmidt electronics
Gruppe	= 09
Device_ID	= ID_OPENFORMAT
Used	= 1
Abbildung	= OpenFormat.jpg
ChangeSettings	= 1
HerstellerLink1	=

```

HerstellerLink2          =
LogViewLink              = www.logview.info
TimeStep_ms               = 1000
TimeGiven                 = 0
KanalAnzahl               = 1
WerteFormat                = ASCII
Prüfsummenberechnung       = keine

[Stati]
StatiAnzahl               = 1
001                       = Flight logging

[serielle Schnittstelle]
Port                      = COM1
Baudrate                  = 9600
Datenbits                 = 8
Stopbits                   = 0
Parität                     = 0
Flusskontrolle             = 0
ClusterSize                = -10
SetDTR                     = 0
SetRTS                     = 0

[Schnittstelle TimeOuts]
RTOCharDelayTime           = 290
RTOExtraDelayTime          = 100
WTOCharDelayTime            = 290
WTOExtraDelayTime          = 100

[Anzeige Einstellungen Kanal 01]
Zeitbasis                  = Zeit
Einheit                     = s
Symbol                      = t
WerteAnzahl                 = 3

Messgröße1                  = Voltage
Einheit1                     = V

```

```
Symbol1          = U
Faktor1          = 1.0
OffsetWert1      = 0.0
OffsetSumme1     = 0.0
```

```
Messgröße2      = Current
Einheit2         = A
Symbol2          = I
Faktor2          = 1.0
OffsetWert2      = 0.0
OffsetSumme2     = 0.0
```