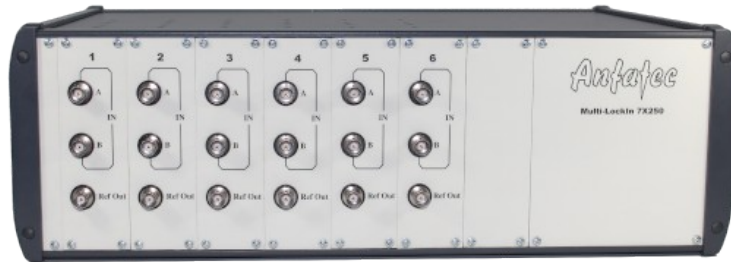


# MULTILOCKIN AMPLIFIER



**UP TO 7 USB LOCKIN AMPLIFIERS**  
**10 mHz to 250 kHz**  
**in one 3 HE 19 " Housing**



Anfatec Instruments AG  
Melanchthonstr. 28  
08606 Oelsnitz /V.  
Germany  
Tel.: +49 (0) 37421 24212  
Tel.: +49 (0) 37421 24221  
<http://www.anfatec.de>  
email: [mailbox@anfatec.de](mailto:mailbox@anfatec.de)

# • Table of Contents

- [General Information](#).....4
  - [Specification of the Multi-Lockin Amplifier 7x250](#).....4
    - [General Parameters](#).....4
    - [Signal Input](#).....4
    - [Reference Output](#).....5
    - [General](#).....5
    - [Standard Part List](#).....5
  - [Licence for the software](#).....5
- [Installation](#).....6
  - [System requirements](#).....6
  - [Get Started](#).....6
  - [Provided Software](#).....6
  - [Software Installation](#).....7
  - [Driver Installation](#).....9
  - [Driver Update](#).....9
  - [Connections to the MultiLockIn](#).....10
- [LockIn Amplification Basics](#).....11
  - [The general idea of Lockin Amplification -> LockIn](#).....11
  - [Mathematical description](#).....12
  - [Noise Measurements](#).....13
  - [Example: Electrical Force Microscope](#).....14
- [Software Description](#).....16
  - [Overview](#).....16
  - [Functions in the Menu Line](#).....17
    - [File](#).....17
    - [Option](#).....17
    - [View](#).....17
    - [Help](#).....17
  - [Functions in the Function Line](#).....17
    - [Frequency Sweep](#).....17
    - [Oscilloscope](#).....17
  - [Parameter settings](#).....19
    - [Time constant](#).....19
    - [RollOff](#).....19
    - [Dynamic](#).....19
    - [Coupling](#).....20
    - [The Meters](#).....20
    - [Frequency](#).....20
    - [Amplitude](#).....20
    - [Phase](#).....20
    - [Harmonic](#).....20
  - [Sweep Frequency](#).....20
    - [Window Description](#).....20

- [Options for the frequency sweep](#).....21
- [DLL Description](#).....24
  - [Working with the DLL in C++ projects](#).....24
  - [DLL-Functions](#).....25
  - [Example programs](#).....28
- [Remote Control with LabView](#).....29
  - [Files and Locations](#).....29
  - [General Programming Directions in LabView](#).....29
  - [Description of Example VI Files](#).....31
  - [Address the Multiple LockIn Amplifiers in LabView](#).....36
- [Revision History](#).....37
  - [Driver Revisions](#).....37
  - [Hardware Revisions](#).....37
  - [Changes in the Manual](#).....37

Copyright 2002-2021 Anfattec Instruments AG. All rights reserved. Anfattec, and AMU are trademarks of Anfattec. Other product and brand names may be trademarks or registered trademarks of their respective owners.

Anfattec Instruments AG assumes no responsibility for any damage or loss resulting from use of this manual.

Anfattec Instruments AG assumes no responsibility for any damage or loss resulting from use of the software. Anfattec Instruments AG assumes no responsibility for any damage or loss by deletion of data as a result of malfunction, dead battery, or repairs. Be sure to make backup copies of all important data on other media to protect against data loss.

Important: Please read Anfattec Licence Agreement contained in this handbook before using the accompanying software programs. Using any part of the software indicates that you accept the terms of Anfattec Software Licence Agreement.

# GENERAL INFORMATION

The Multi-Lockin is a 3 HE 19" unit that contains up to seven independent lockin amplifiers with differential inputs.

## SPECIFICATION OF THE MULTI-LOCKIN AMPLIFIER 7x250

### GENERAL PARAMETERS

Up to 7 x Digital Dual-Phase Lock-In Amplifier	
Dynamic Reserve	> 135 dB <sup>(1)</sup>
Input Noise (low noise mode)	< 4 nV <sub>rms</sub> /Hz <sup>0.5</sup> @ 100 kHz
Remote Control	USB 2.0
Time Constants	10 μs ... 5 s / 1 μs in sync mode
Sensitivity	10 nV ... 10 V
Phase Resolution	0.0001°
Amplitude Deviation (0..250 kHz)	< 1 %
Phase Deviation (0..250 kHz)	< 0.5 °
Maximum Frequency	250 kHz
Weight	1.9 kg
Housing Size	44 cm (W) x 25 cm (D) x 14.5 cm (H)

### SIGNAL INPUT

Voltage Input	BNC (A and B // A-B)
Input coupling:	dc or ac (f <sub>-3dB</sub> = 2 Hz)
Input Impedance	10 MΩ    18 pF
Damage Threshold	+/- 12 V
Input sampling rate	20 MHz
Bandwidth	dc to 250 kHz (f <sub>-3dB</sub> > 250 kHz)
Input Ranges	± 3.5 V <sub>rms</sub> , ± 350 mV <sub>rms</sub> , ± 35 mV <sub>rms</sub>
CMRR* (A-B mode; 1 V, 1 kHz input)	> 60 dB, > 80 dB, > 100 dB (low noise)
Input Sensitivity	10 nV to 10 V
Typical Input Noise:	U <sub>ac</sub> = 0 V, 50 Ω @ input → see table
Input Offset (Low Noise Range)	< 1 mV (Mode: DC, InA only)
RollOff	6 dB/oct, 12 dB/oct, 24 dB/oct
Time Constants	10 μs ... 5 s
Amplitude Accuracy (dc to 250 kHz)	< 1 %
Gain deviations between dynamic ranges:	< 1 %
Phase shift accuracy (dc to 250 kHz)	< 0.5°

Input Noise measured with U<sub>ac</sub> = 0 V and 50 Ω @ input shown in nV<sub>rms</sub>/Hz<sup>0.5</sup>

Frequency	Filter \ Range	High dynamic reserve	Normal	Low Noise
250 kHz	10 ms	< 80 (typ. 70)	< 10 (typ. 8)	< 5 (typ. 2.6)
100 kHz	10 ms	< 60 (typ. 50)	< 7 (typ. 6)	< 5 (typ. 2.6)
10 kHz	10 ms	< 60 (typ. 35)	< 7 (typ. 5)	< 6 (typ. 2.6)
1 kHz	1 s	< 300 (typ. 150)	< 30 (typ. 15)	< 10 (typ. 4)

\* The CMRR determined by connecting the same input signal (e.g. 1 kHz sine wave with 1 Vrms amplitude) to both inputs A and B and detecting the remaining signal at the same frequency in mode A-B.

## REFERENCE OUTPUT

Internal Oscillator	10 mHz .. 250 kHz (2 MHz)
Frequency Resolution	< 10 mHz
Frequency Accuracy	+/- 50 ppm from 0 °C to 70 °C
Amplitude Accuracy (dc to 250 kHz)	< 0.5 %
Reference Output Voltage	< 1 mVpp ... 15 Vpp
Output Noise @ 100 kHz	@ Uac = 1 mV, < 200 nV <sub>rms</sub> /Hz <sup>0.5</sup>
Output sampling rate	>= 20 MHz

## GENERAL

Interface	Plug & Play USB 2.0 interface
Drivers	Win7, Win10 - 32 Bit and 64 Bit
Power consumption	12 Vdc, 1 A
Warranty	2 years

## STANDARD PART LIST

Multi Lockin 7x250	1
EEC Power Supply cable	1
USB cable M-M , 2 m long	1
Manual	1
Software on USB-stick	1

## LICENCE FOR THE SOFTWARE

The Source code for the remote control software and the DLL source code are provided with the General Public Licence (GPL). All examples for LabView can be freely implemented and changed by the user.

# INSTALLATION

## SYSTEM REQUIREMENTS

- compatible PC
- 1 MB hard disk space
- Windows XP (32 bit) / Windows 7 (32/64 bit) / Windows 10 (32/64 bit)

## GET STARTED

- Connect the power supply to "Power"
- Connect the USB interface to your PC
- When Windows asks for a driver, then provide the path to the driver on the USB-Stick under "Driver" depending on your Windows Version. Otherwise see "[Driver Installation](#)".
- One needs to start the software "lockin.exe" once as **Administrator**<sup>1</sup>



Figure 1: Power Supply Connector & USB Connector at the backside of the controller

## PROVIDED SOFTWARE

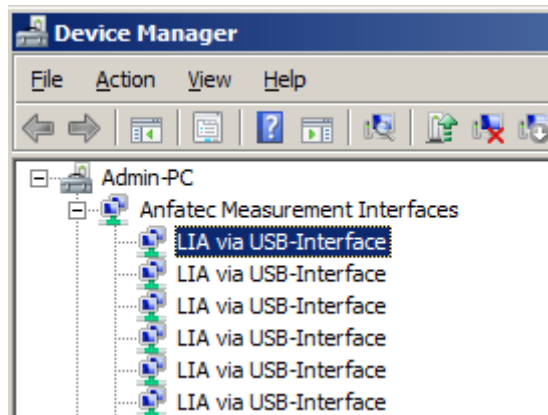
On the delivered USB Stick, you find

Driver	in	.\USBLockin_Driver
executable application	in	.\DLL_and_App\Apps\Win32 and .\DLL_and_App\Apps\Win64
Source Code of application	in	.\DLL_and_App\Apps\Delphi7 or .\DLL_and_App\Apps\DelphiXE7
DLL file	in	.\DLL_and_App\DLL\DLLTestApp_DelphiXE7\Win32 and .\DLL_and_App\DLL\DLLTestApp_DelphiXE7\Win64
Source Code of DLL	in	.\DLL_and_App\DLL\DLLSourceXE7 or .\DLL_and_App\DLL\DLLSourceDelphi7
Source Code of DLL Test App	in	.\DLL_and_App\DLL\DLLTestApp_DelphiXE7
LabView examples	in	.\DLL_and_App\DLL\Labview
Python examples	in	.\DLL_and_App\DLL\Python
Scilab examples	in	.\DLL_and_App\DLL\Scilab

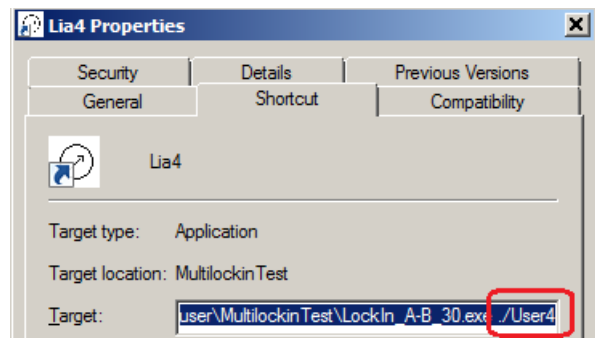
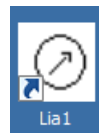
<sup>1</sup> This copies the scaling data for the lockin amplifier into the registry of the PC. Afterwards, all other applications and all users can use these registry entries.

# SOFTWARE INSTALLATION

- Install the correct driver (32 bit or 64 bit) – if not yet done during plug-in. After correct installation of Device Driver, the Device Manager of the PC should display each lockin amplifier unit separately like this:

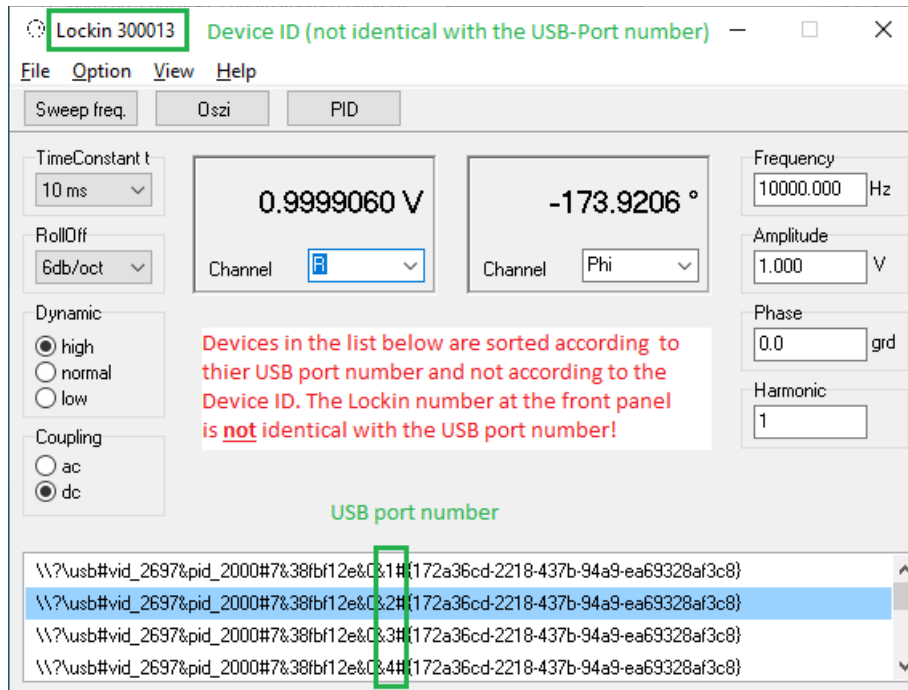


- Copy the LockIn.exe into any local folder (it is required that one has the right to write into this folder, because the ini-files will be generated here)
- Send a link from this LockIn.exe to the Desktop
- change the name of the link into "Lia1" for the 1st lockin amplifier
- Right click onto the link and open the Link Properties
- Behind "Target:" add a separate name for the ini-file for this lockin amplifier No 1 (e.g. './Lia1' or './User1')
- Run the LockIn.exe **once** as Administrator from this link. This generates registry entry for the 1st lockin amplifier.
- When running for the 1st time - the software will ask to generate this new ini-file -> OK

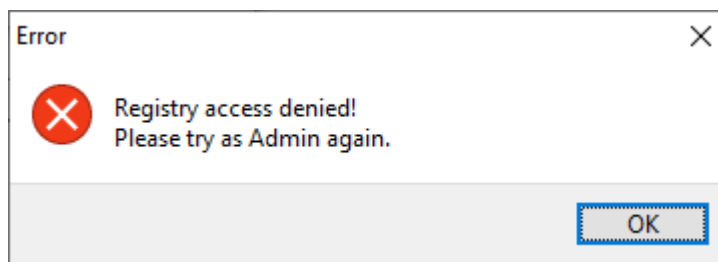


For **each** further lockin amplifier:

- Copy the 1st link and paste it again onto the PC's Desktop
- Change the ini-file name behind target in the link properties (e.g. into './User2' ...)
- Run the LockIn.exe (**not as** Administrator for the 1st time)
- Confirm that the new ini-file is generated
- Close the software
- Restart the software as Administrator
- Select the next device (At this point, a new Registry entry will be generated. Start the Registry Editor and check the entries in HKEY\_LOCAL\_MACHINE → SOFTWARE → Anfatec → Lockin. The names of the new entries are equal to the Device ID, e.g. 300013)

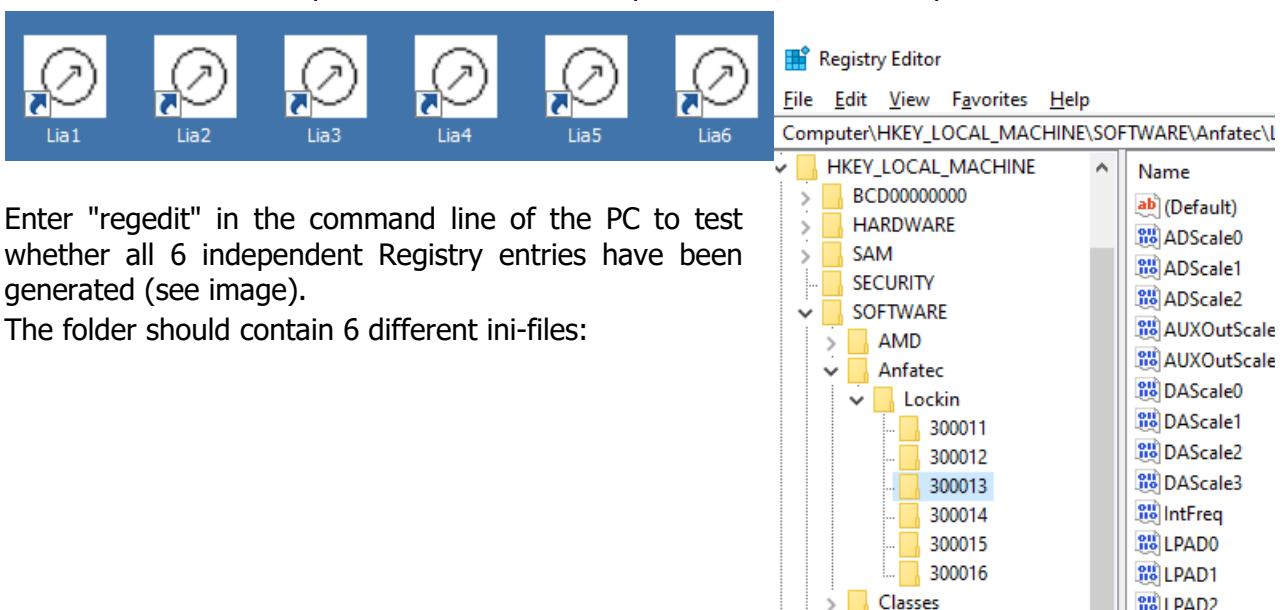


- If one selects the next (not yet used) device (the devices are sorted according to the USB port number) from the list in the software surface AND the Registry Entry with the correct scaling does not yet exist, the software will show this error message



and close itself.

After this has been completed for all 6 lockin amplifier units, the Desktop should look like this:



Enter "regedit" in the command line of the PC to test whether all 6 independent Registry entries have been generated (see image).

The folder should contain 6 different ini-files:



Name	Date modified
LockIn.exe	10/21/2020 11:14 AM
Device6.ini	7/27/2021 1:43 AM
Device5.ini	7/27/2021 2:18 AM
Device4.ini	7/27/2021 1:45 AM
Device3.ini	7/27/2021 1:45 AM
Device2.ini	7/27/2021 1:45 AM
Device1.ini	7/27/2021 2:18 AM

For DLL-based applications:

- Check the whether the application is 32 bit or 64 bit
- Copy the related DLL (.....\DLL\Win32 or ..... \DLL\Win64) to your project folder

## DRIVER INSTALLATION

Connect the USB lockin first. If the system asks for a driver, browse to the USB-Stick  
 .\Driver\version\x86 for 32 bit or  
 .\Driver\version\x64 for 64 bit operating system.  
 Select the inf-file.

If the system does not ask automatically, one has two possibilities:

A) install through device manager

- Click on Start
- Enter "devmgmt.msc" in the Search Line to open the Device Manager
- Select the "Anfatec Lockin Interface" and right-click to "Update Driver Software"
- "Browse for driver software on your computer"
- provide the path "%YourPath/Driver/version/.."
- select the inf-file
- Say OK when Windows recommends not to install from an unknown driver vendor

## DRIVER UPDATE

Install updated drivers through device manager

- Click on Start
- Enter "devmgmt.msc" in the Search Line to open the Device Manager
- Select the "Anfatec Lockin Interface" and right-click to "Update Driver Software"
- select "**Browse my computer for driver software**"
- provide the path "%YourPath/Driver/version/.."
- select the inf-file

**Important Note:** It is necessary to update the driver for **each USB port** at which the Lockin Amplifier will be connected. Otherwise, the old driver will be used

# CONNECTIONS TO THE MULTILOCKIN

## Back Side:



Figure 2: Backside of the Multi-LockIn Amplifier. Connectors for Power Supply and USB-Connection to PC. Power Switch and Fuse.

**Power Supply** – 230/115 V ac , 6.3 A

**USB** – interface for remote control via USB2.0

## Front Side:

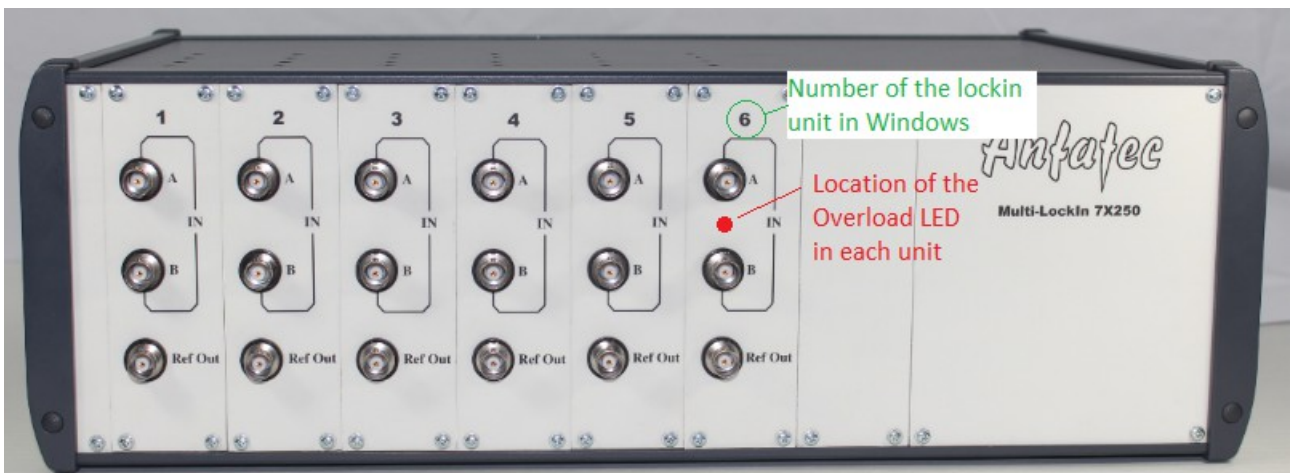


Figure 3: Front side of the Multi-LockIn Unit. There are 7 slots available to mount a separate lockin amplifier. Each lockin amplifier Unit provides the connectors InA , InB and RefOut.

**LEDs** ... each lockin amplifier unit has an LED that indicates an overload of the AD converters at the input. During normal operation this LED is invisible.

**IN A** ... is the analogue input of the lockin in which the signal to be detected is plugged in.

**IN B** ... is the 2nd input, when the lockin is used differentially.

**RefOut** ... is a sine wave output whose amplitude is set by software and whose frequency equals the currently selected centre frequency of the lockin amplifier.

# LOCKIN AMPLIFICATION BASICS

## THE GENERAL IDEA OF LOCKIN AMPLIFICATION -> LOCKIN

A lockin amplifier is a phase sensitive bandpass filter with a centre frequency  $f$  and a bandwidth  $t$ . It has two output signals in parallel: either the amplitude  $R$  and the phase  $j$ , or the real part  $X$  and the imaginary part  $Y$ .

- The centre frequency is the frequency at which the LockIn is searching for its signal.
- The bandwidth gives the frequency range around this centre frequency, from which the output signal is derived.
- The output signal  $R$  is the measured amplitude of the analysed signal at the centre frequency  $f$ .
- The phase  $j$  is the relative phase shift between the analysed signal and an internal reference signal with the same frequency.

For analysing the input signal, the LockIn needs an internal frequency reference with a certain amplitude and phase. (see Fig. 8).

The bandwidth of a LockIn can be visualised by a simple setup. A generator gives a constant frequency  $f_{in}$  signal to the input and sweeping the centre frequency  $f$  around the input frequency  $f_{in}$ . Figure 10 shows that the amplitude at the input frequency is always measured with the same value. For other frequencies, the measured amplitude is dependent on the distance between the actual centre frequency and the input signal frequency.

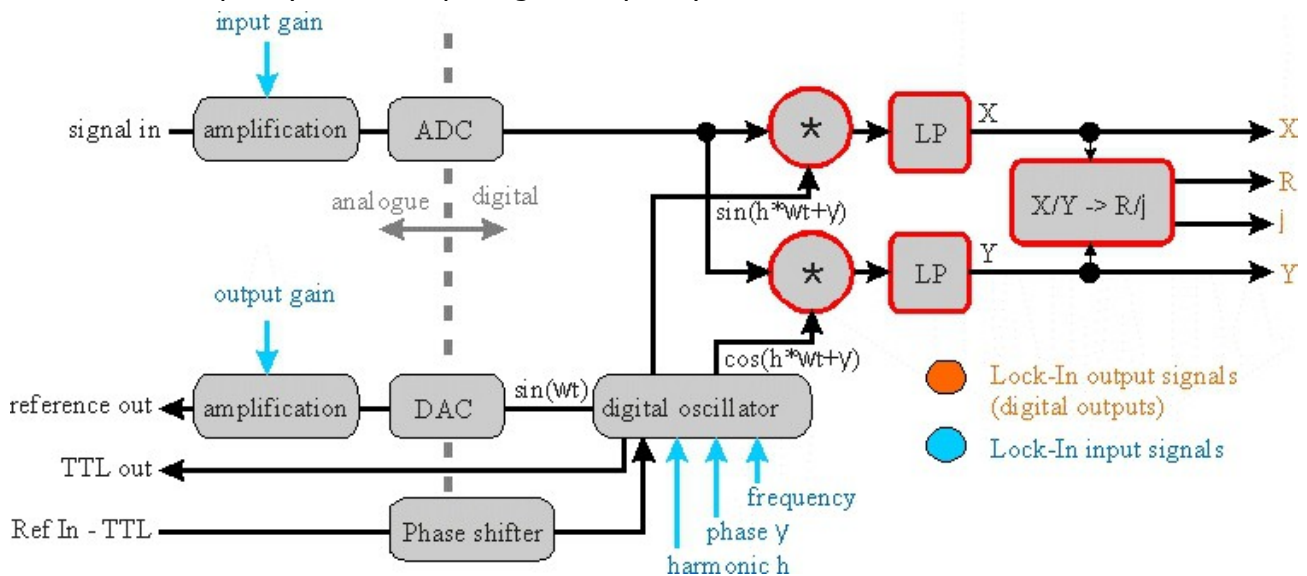


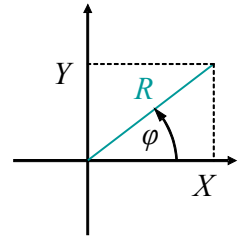
Figure 4: Schematic diagram with the basic components of the lock-in amplifier.

A schematic diagram with the basic components of the lock-in amplifier is shown in Figure 5 on page 13. First of all the received signal is amplified and digitized. Divided into two separate channels, the signal is multiplied by the mentioned reference signal (with frequency  $\omega_{ref}$ ) and the  $90^\circ$  phase-shifted reference signal respectively. The reference signal is generated with the lock-in's digital oscillator. An additional implemented phase displacement  $\psi$  enables to compensate phase differences caused by the measuring equipment. After multiplication, the resulting signals are low-pass filtered and provide now information about the real part  $X$  and the imaginary part  $Y$  of the analysed signal relating to the phase position of the reference signal. Out of them the amplitude  $R$  as well as the relative phase shift  $\varphi$  are calculated.

# MATHEMATICAL DESCRIPTION

A digital LockIn realises its filtering by certain mathematical procedures.

Think,  $Y(t)$ , the signal to be analysed, is dependent on a value  $X(t)$ . For instance: we measure a photo voltage  $U(t)$  which depends on the light intensity given by an LED  $I(t)$ . If the background light intensity is much too high to detect small intensity changes of the LED light, one can modulate the light intensity  $I(t)$  at a certain frequency  $\omega_{ref}$  and detect with a narrow band filter at this frequency.



Thus,  $X(t)$  is modulated as  $X(t) = X_0 + X_1 \cos(\omega_{ref} t)$ . Then,  $Y(X(t))$  can be developed into a Taylor-Series:

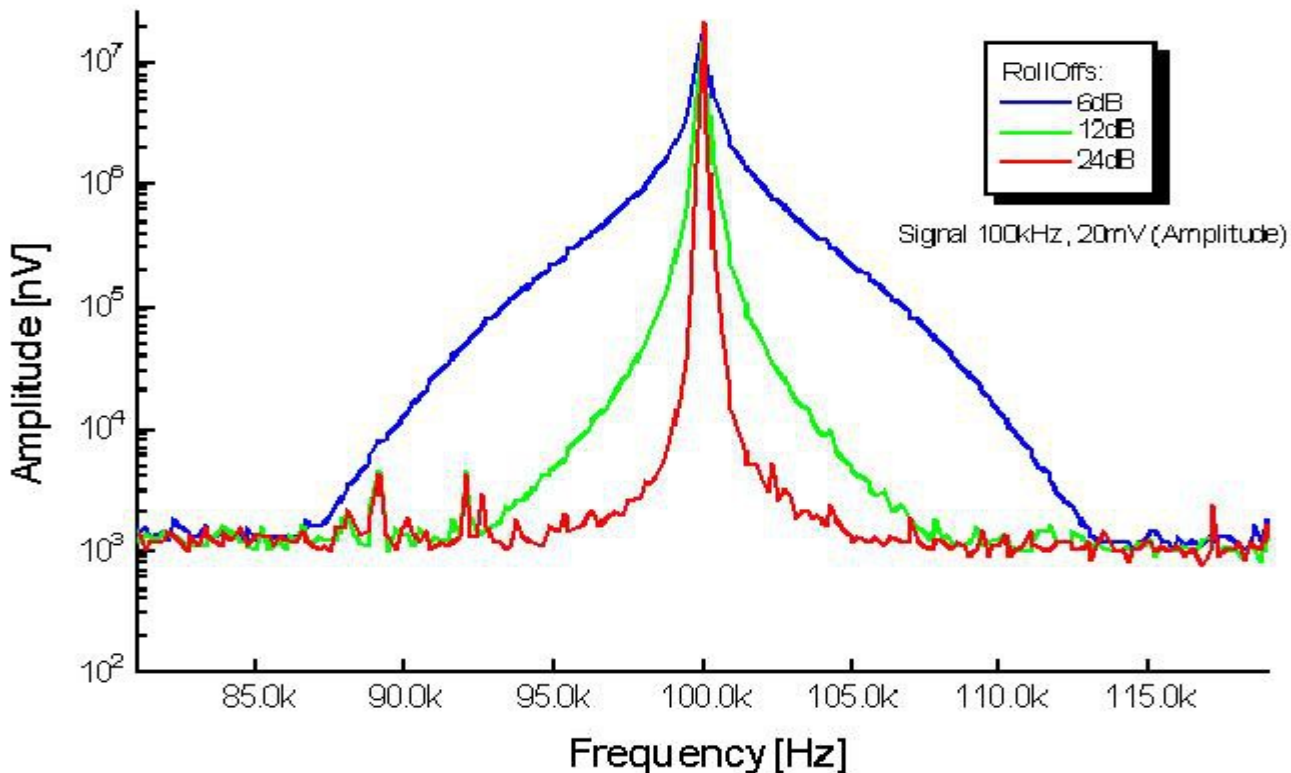


Figure 5: Output signal of a LockIn amplifier for a single frequency input signal of 100 kHz and 20 mV<sub>rms</sub> amplitude at the input. The time constants have been kept constant.

$$Y(X_0 + X_1 \cos(\omega_{ref} t)) = \sum_{k=0}^{\infty} Y^{(k)}(X_0) \cdot \frac{X_1^k}{k!} \cdot \cos^k(\omega_{ref} t) \cdot Y^{(k)}(X) = \frac{d^k Y}{dX^k} \Big|_{X_0} \quad k \in \mathbb{N}$$

Equ.(1)

For the value of  $Y$  at the time  $t$ , all measurement values  $Y(t)$  detected in the period  $\tau$  are used. This period  $t$  should be a whole-numbered multiple of  $2\pi\omega_{ref}$ . Due to the modulation of  $X(t)$  at  $\omega_{ref}$ , modulations of  $Y(t)$  at the frequencies  $m \omega_{ref}$  with  $m = 1, 2, \dots$  are expected. These modulations equal the Fourier components  $Y(m \omega_{ref})$  with  $m = 1, 2, \dots$  of the input signal  $Y(t)$ , which are given as

$$Y(m \omega) = \frac{1}{\tau} \int_{-\tau/2}^{\tau/2} Y(X(t)) e^{-im\omega t} dt$$

and together with Equ. 1

$$Y(m \omega_{ref}) = \sum_{k=0}^{\infty} Y^{(k)}(X_0) \frac{X_1^k}{k!} K_m^k$$

with

$$K_m^k = \frac{1}{\tau} \int_{-\tau/2}^{\tau/2} \cos^k(\omega_{ref} t) e^{-im \omega_{ref} t} dt .$$

The factors  $K_m^k$  get zero for  $k$  to infinity. For small  $k$ , the  $K_m^k$  remain too big to be neglected. In order to neglect  $n$ -th order parts, the  $n$ -th derivative of  $Y(t)$  to  $X(t)$  has to be negligible.

The first 10 ( $k = 1 \dots 10$ ) coefficients  $K_m^k$  for the first 4 harmonics  $m = 1 \dots 4$  are given in the following table:

	k =									
m	1	2	3	4	5	6	7	8	9	10

1	$\frac{1}{2}$		$\frac{3}{8}$		$\frac{5}{16}$		$\frac{35}{128}$		$\frac{63}{256}$	
2		$\frac{1}{4}$		$\frac{1}{4}$		$\frac{15}{64}$		$\frac{7}{32}$		$\frac{105}{512}$
3			$\frac{1}{8}$		$\frac{5}{32}$		$\frac{21}{128}$		$\frac{21}{128}$	
4				$\frac{1}{16}$		$\frac{3}{32}$		$\frac{7}{64}$		$\frac{15}{128}$

## NOISE MEASUREMENTS

Lock-in amplifiers are capable to measure noise. They detect a signal at a certain centre frequency  $\omega_{ref}$  with an equivalent noise bandwidth. For Gaussian noise, the equivalent noise bandwidth of a real low pass filter is the bandwidth which passes the same amount of noise as a perfect rectangular filter with the equivalent noise band width.

The equivalent noise bandwidth of the lockin amplifier is determined by the time constant and the slope of the used Butterworth filters. It is calculated by

$$B_n = \int_0^{\infty} \frac{1}{1 + \omega^{2n}} d\omega \quad (4)$$

The normalized Butterworth filter noise bandwidths are:

<b>Filter order</b>	<b>Bandwidth / <math>\tau</math></b>
1 = 6 dB/oct	1.570796
2 = 12 dB/oct	1.110721
4 = 24 dB/oct	1.026172

In order to measure noise spectra, the resulting data should be divided by the square-root of the used bandwidth.

The integrated spectrum is taken with a time constant of 10 ms and a slope of 24 dB/oct (4<sup>th</sup> order) – bandwidth factor ~ 1,03. The related bandwidth is then 1/10 ms = 100 Hz. In order to interpret the result as noise in units of V/Hz<sup>0.5</sup>, the spectrum should be divided by 10 Hz<sup>0.5</sup> (= sqrt(100 Hz)).

## EXAMPLE: ELECTRICAL FORCE MICROSCOPE

In this chapter electrical force microscopy (EFM) is briefly presented as an example for the application of lock-in amplifiers.

EFM is a related technique to the well-established atomic force microscopy (AFM). Its special aim is to detect electrical forces to learn something about the electrical properties of a surface, for example about the local distribution of surface potentials on electronic devices or different dopant

concentrations in semiconducting materials.

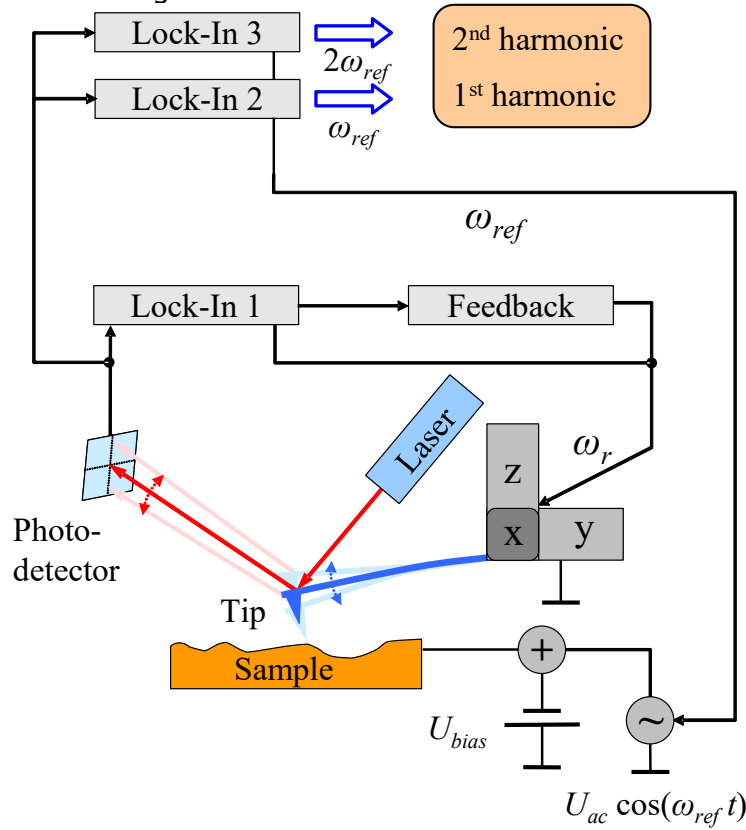


Figure 6: Schematic diagram of the EFM experimental setup.

The fundamental experimental setup shown in Figure 6 is based on a conventional atomic force microscope: In the non-destructive dynamic non-contact mode, an oscillating metallic tip fixed to a cantilever is scanning over the surface by means of a piezo scanning device. The distance between tip and sample can be controlled by monitoring the oscillation amplitude of the cantilever, because its value is influenced by short-range van der Waals forces. Therefore a reflected laser beam and a position-sensitive photo-detector are used. A lock-in amplifier analyses the detector signal at the cantilever resonance frequency  $\omega_r$  and passes the determined amplitude value to a feedback control system that re-adjusts the tip-sample distance. The required displacement of the z-piezo can be recorded as topography signal.

In addition to the topography, EFM detects electrical forces, too. These forces are proportional to the derivative of the capacitance  $C$  of the tip-sample arrangement with respect to the tip-sample distance  $z$  and proportional to the potential difference  $U$  squared:

$$F_{el} = -\frac{1}{2} \frac{dC}{dz} U^2 \quad (5)$$

A possible voltage dependency of  $C$  is neglected in this consideration.

The voltage  $U$  contains a direct voltage part  $U_{DC}$  and an alternating voltage part  $U_{AC}$ :

$$U = U_{DC} + U_{AC} \cdot \cos(\omega_{ref} t) \quad (6)$$

$U_{DC}$  consists of an additional applied bias voltage and, what is especially interesting from the physical point of view, potential differences caused by different electronic work functions and charges. For separating the impact of these electrical forces from other forces (e.g. van der Waals forces),  $U$  has to be modulated at the frequency  $\omega_{ref}$ . As a result of this, the measurable photo-detector signal is modulated, too.

Inserting equation (6) in (5) and using power-reduction formulae of trigonometric functions one can expect a force between tip and sample at the frequency  $\omega_{ref}$  as well as at  $2 \cdot \omega_{ref}$ . Figure 7



confirms this prediction: It shows the frequency spectrum of the cantilever oscillation with clear signals at the mechanical excitation frequency  $\omega_r$  and at the frequencies of the first and second harmonic of the electrical excitation.

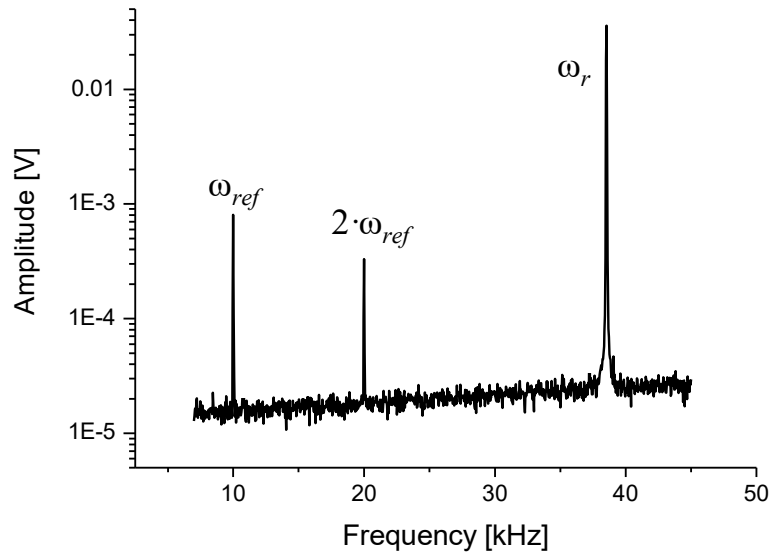


Figure 7: Frequency spectrum of the oscillating cantilever

Now two additional lock-in amplifiers (see Figure 6) can be used to analyse the photo-detector signal at the frequency  $\omega_{ref}$  and  $2 \cdot \omega_{ref}$  simultaneously. The measured amplitudes of the signals are proportional to the strength of the electrical forces.

# SOFTWARE DESCRIPTION

## OVERVIEW

The GUI software *lockin.exe* opens with the following surface:

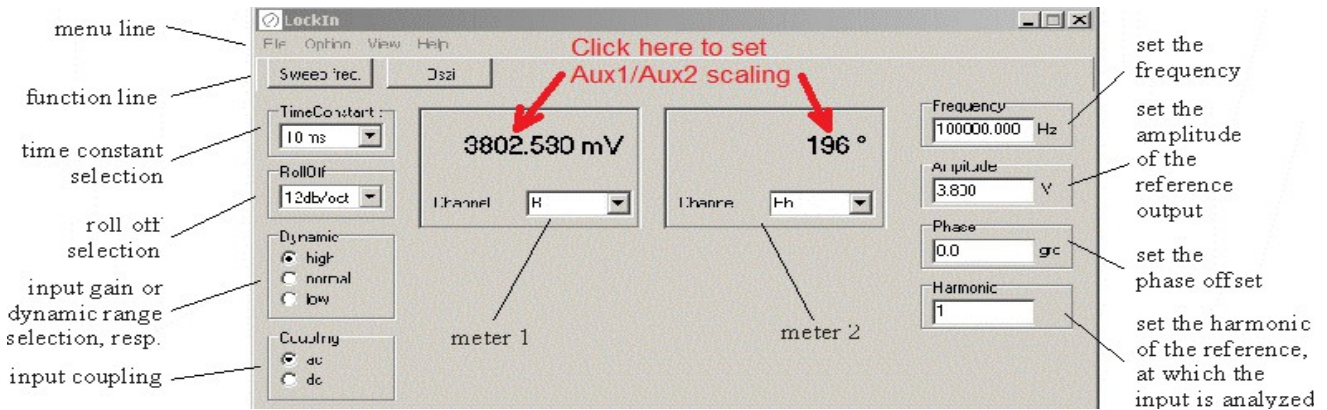


Figure 8: Main window of the LockIn program with description of the functions.

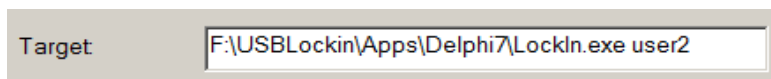
Figure 8 shows the main window of the LockIn program, which appears when the program is opened. For checking the actually detected values, the two meters are used. Basic input parameters (frequency, amplitude, offset phase and harmonic) can be selected in the right part of the window. Parameters, which concern the input stage (time constants, RollOff, and input gain) are chosen in the left part of the window.

The frequency is either the internal frequency (black numbers) or the detected external frequency from the reference input (grey numbers).

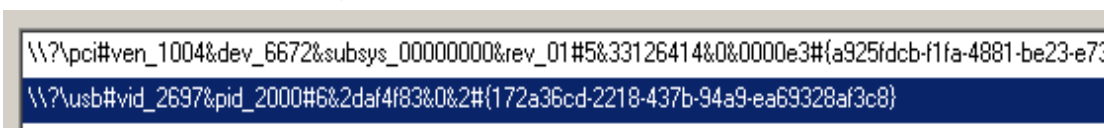
The menu line allows typical Windows functions, while the button in the function line open new windows with specific functions.

This software can be started as often as many Lockin Amplifier Units are mounted in the 3HE rack unit. In order to address a certain unit with dedicated parameters, follow these steps:

1. Create as many desktop shortcuts to the *lockin.exe* as many devices you wish to operate
2. Right-click on each desktop shortcut and choose "Properties"
3. add a parameter (e.g. user2) behind the entry in the target line:



4. Start each shortcut once, select the device to be associated with this shortcut:



The same procedure can be used to open the software with different settings.

# FUNCTIONS IN THE MENU LINE

## FILE

*Exit* - Exit the program

## OPTION

- **overload**

Overload occurs, when the dc input signal exceeds the full scale sensitivity for the selected range. This full scale sensitivity is 7 V<sub>rms</sub> for high reserve, 700 mV<sub>rms</sub> for normal reserve and 70 mV<sub>rms</sub> for low reserve. With this option can be selected whether a beep and/or a color change is shown in case of overload.

- **ext. reference**

For LockIn-amplifier versions with PLL (Phase locked loop), the reference frequency can be either the internal oscillator frequency or an external reference frequency. If the "ext. reference" is selected, the internal PLL is enabled. The locked frequency is shown in grey in the frequency window. If there is no input signal connected to the reference input, a default frequency of 2 mHz is shown, but the LockIn is not working properly.

With disabled PLL, the frequency given in the frequency window is written in black.

## VIEW

It can be selected, which of the meters is shown.

- **Devices**

Select whether the strings for the lockin amplifier(s) should be shown. Useful when using multiple lockin amplifiers to select which one should be used. Is also used for LabView (see chapter "Address the Multiple LockIn Amplifiers in LabView").

## HELP

**About** – shows the current program version and the version of the used hardware driver.

**LockIn help** – calls the table of contents of the HTML help file supplied with the program

**Help as PDF** – calls the current manual (Manual.pdf) and opens it in the Microsoft Internet Explorer.

# FUNCTIONS IN THE FUNCTION LINE

## FREQUENCY SWEEP

Opens the "[frequency sweep](#)" window

## OSCILLOSCOPE

The oscilloscope works like a real 3-channel-oscilloscope. Content, scaling type and offset of the

three channels is selectable.

**Channel selection:** is done from a drop down list, which shows only the available channels.

The two *numbers* above the drop down list for channel selection display the “scaling factor per vertical division” (= left number, hint: “y-scale in /div”, example: 12.90 nV/div) and the mean value. Both are calculated from all data acquired from the left oscilloscope edge till the current oscilloscope time. Therefore, these numbers are subsequently re-calculated.

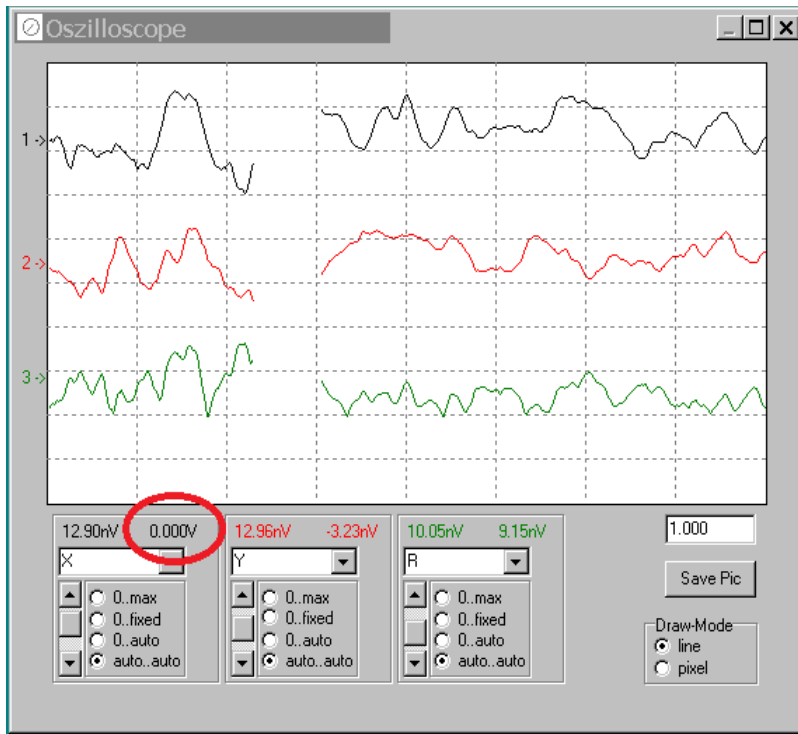


Figure 9: Screen shot of the software oscilloscope.

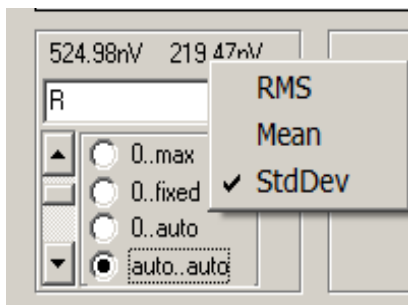
**Vertical scaling types:**

- 0..max                    the scaling is set to maximum value of the channel
- 0..fixed                 the maximum value can be changed by a slider appearing on the right sight of the scaling type selection
- 0..auto                  the program calculates the the optimum, but takes always “0” as minimum
- auto..auto                automatically scaled

**Time scaling** is done with the edit window (right sight) in seconds.

**“Save Pic”** saves the oscilloscope screen in a bitmap file.

**“Draw mode”** selects whether the data are drawn as dots or lines.



Right-click on the right number to change its meaning from RMS to Mean or StdDev.

# PARAMETER SETTINGS

## TIME CONSTANT

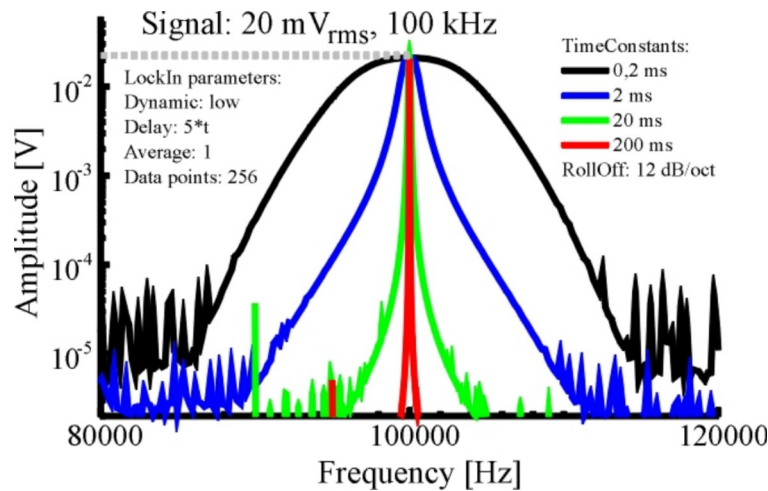


Figure 10: Effect of different time constants.

This option selects the used time constant  $\tau$  for the low-pass filter. If applicable, the closest approximation of the digital equivalent to an analogue Butterworth filter is used. The corner frequency of the Butterworth filter is  $1/\tau$ .

The effective noise bandwidth of this filter is

$$B_n = \int_0^\infty \frac{1}{1 + \omega^{2n}} d\omega \quad .$$

The normalized Butterworth filter noise bandwidths are:

<b>Filter order</b>	<b>Bandwidth</b>
1 → 6 dB/oct	1.570796
2 → 12 dB/oct	1.110721
4 → 24 dB/oct	1.026172

## ROLLOFF

The meaning of the "RollOff" is shown in the curves in Figure 5. It equals the degree of the lowpass filter. One can chose between 6 dB/oct (1<sup>st</sup> order), 12 dB/oct (2<sup>nd</sup> order) and 24 dB/oct (4<sup>th</sup> order).

## DYNAMIC

This switches the input amplification of the LockIn. With "high" dynamic, input amplification is 1. The maximum signal amplitude is then +/-10 V. The "normal" input amplification is 10, which equals maximum signal amplitudes of +/- 1 V. When the low dynamic is chosen, the resolution of the LockIn is highest, but the signals cannot exceed 100 mV.

## COUPLING

If the specification of the instrument allows it, this option switches between DC coupled input and AC coupled input. Note: The 3dB corner frequency of the input high pass is around 2Hz. Reference frequencies around 2 Hz and below may cause misleading results.

## THE METERS

The meters display the LockIn output channels, which are X and Y as well R and Phi, in physical units.

## FREQUENCY

If written in black letters, this is the actual reference frequency which is used at the reference output and as reference frequency for the signal evaluation of the input.

Click with the right mouse button to switch from external to internal reference. In case of internal reference, the numbers are written in grey.

## AMPLITUDE

This is the amplitude of reference output.

## PHASE

Allows to give a phase offset between the reference output and the input. In the schematic in Fig. 4, this phase is equal to the LockIn input parameter "Phase  $\gamma$ ".

## HARMONIC

Selects, which harmonic of the reference frequency is evaluated. The possible values range from 1 to 15. When selecting higher harmonics, take care, that, due to lowpass filtering, the maximum input frequency the LockIn cannot be higher than 2 MHz.

## SWEEP FREQUENCY

This window serves the acquisition of frequency dependent spectra of any of the LockIn input channels. The frequency sweep uses the internal reference. When the PLL is enabled for normal operation, it will be disabled during frequency sweep.

The number of data points, parameters for the visualisation as well as for the saving and copying the acquired data can be changed in the [option window](#).

## WINDOW DESCRIPTION

**"Delay"**: is the time delay between each acquired data point. During spectrum acquisition, the frequency is set to the next value. Then, the system waits "Delay" and takes one single value from the acquired Channel.

As this delay has to be related to the time constant of the LockIn, the options in the drop down list for the delay are given in multiples of  $\tau$ . Thus, independently on the [time constant  \$\tau\$  given in the main window](#), the time constant for the acquisition of the spectrum is always correct.

**"from" and "to"** define the values of the start frequency and the stop frequency. For the spectrum's acquisition, one chooses the wanted frequency range, and presses the "start-button". If the time constant was very high, the spectrum might take a while. In order to stop the acquisition, the start-button can be pressed a second time.

“Channel” is a drop down list of available data channels (X, Y, R, and Phi).

“Range back” - click with the right mouse button in the data screen, and a pop-up menu with list of four frequency ranges appears. The upper one is a standard range, which can be changed in the “Option/acquire” part. The next three are, from the bottom to the top, the last three used frequency ranges.

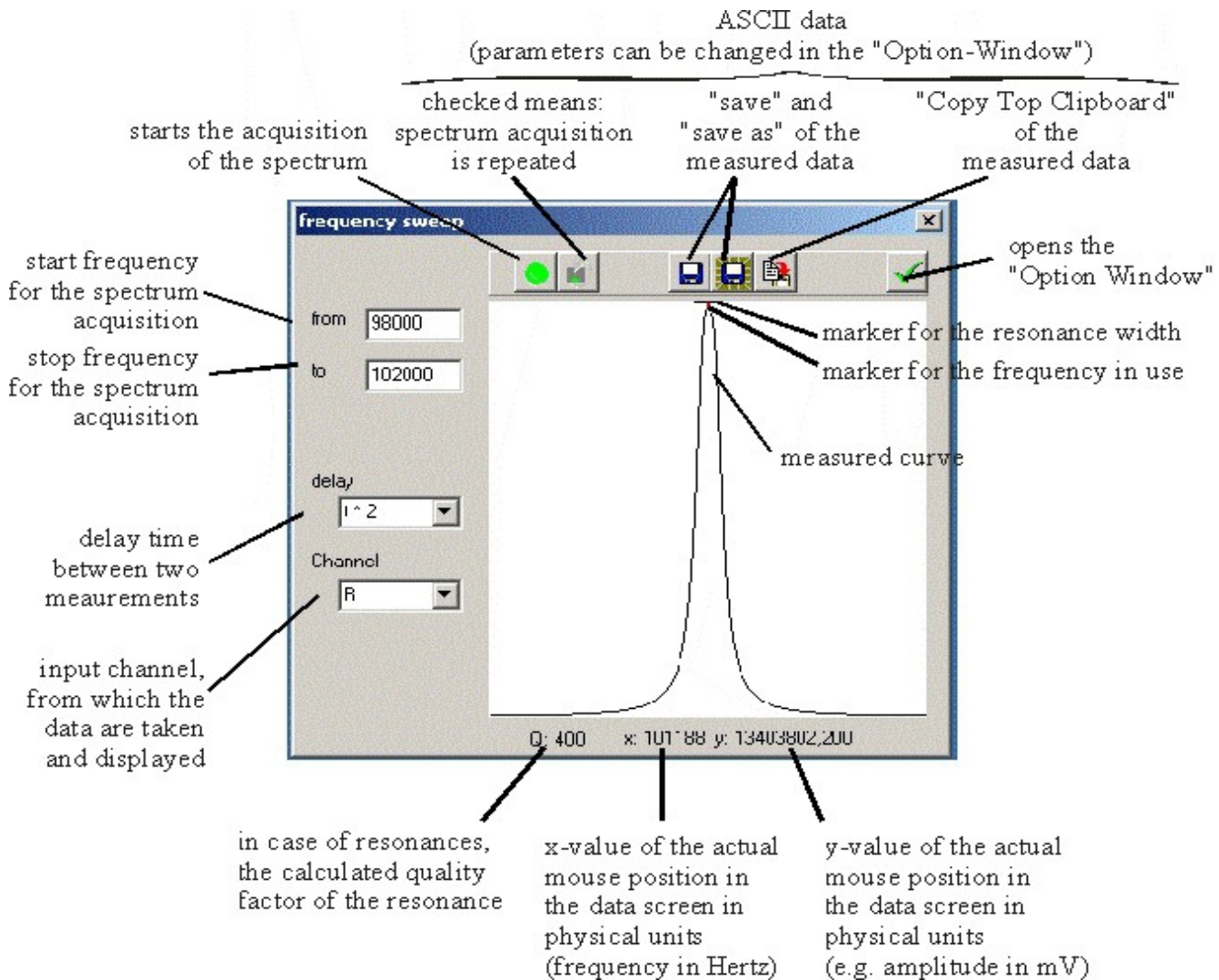


Figure 11: Window for the spectra acquisition.

## OPTIONS FOR THE FREQUENCY SWEEP

The option window provides three cards: “save” - parameters about the saving and copying format of data, “acquire” – parameters about the data acquisition, and “view” – parameters around the screen of showing the data.

### Save-Tab:

The saved files and the data copied to the clipboard have an ASCII structure. The data are written in lines (each frequency value one line) and delimited by the given delimiter (“TAB” in the example) are saved. The frequency values are only saved, if “Save x-Axis” is checked. All history data are saved too and also delimited the same character.

Data file example:

1000,00      234,09  
 1200,00      237,98 ...

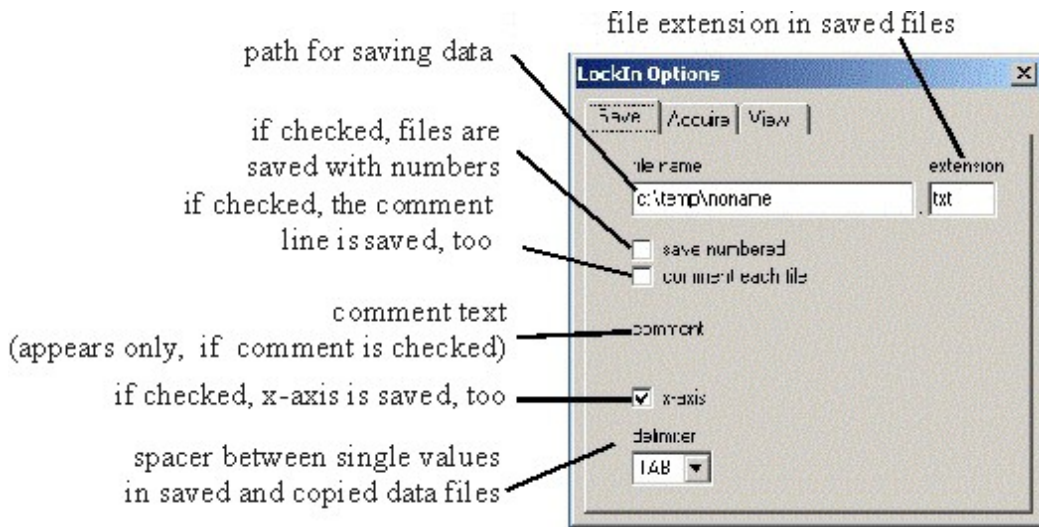


Figure 12: description of the card "save" in the sweep frequency options.

**Acquire-tab:**

**"Wobble"** - if a large range is scanned for overview purposes and the frequency peak, which should be found, is too small to be excited (because the single frequency steps are too big), the wobble option can help. If wobble is checked, the frequency is not kept constant during scan. It is varied (wobbled) between the neighbored values while the data are taken. This makes sure, that even small peaks can be found in an overview spectrum with only some 100 data points.

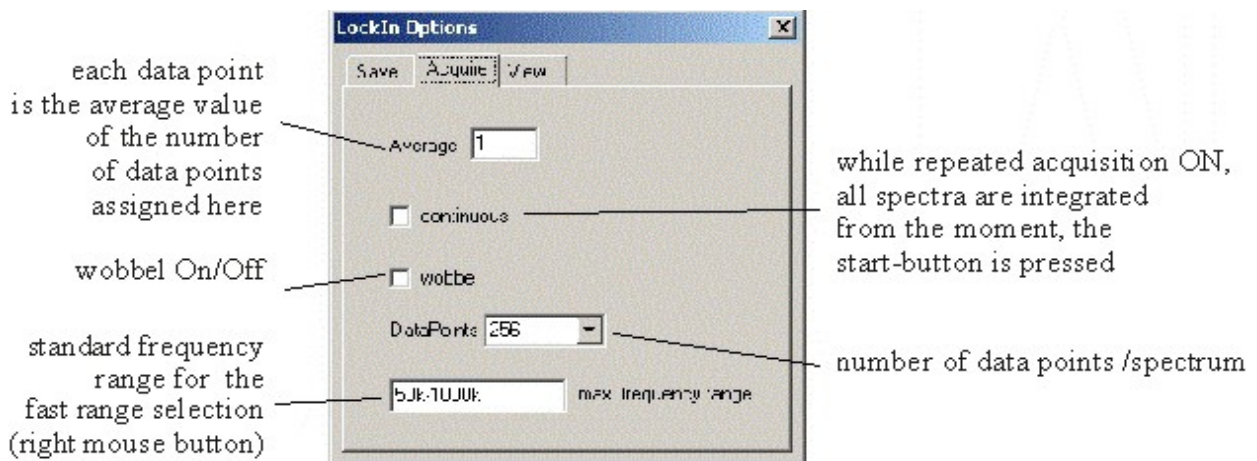


Figure 13: Description of the card "acquire" of the sweep frequency options.

**"Standard frequency"** - is the range, which appears at the topmost position, if the right mouse button is used in the data screen of the "Sweep Frequency" window.



### **View-tab:**

In the data screen, several data curves can be displayed simultaneously.

Therein, the "History depth" is the number of old curves added to the actual one. If the depth is 2, the actual, the last and the last but one curves are displayed. The actual curve is always of black colour. The last is red, and the last but one is green. More curves get the usual next colours from the Windows™ palette.

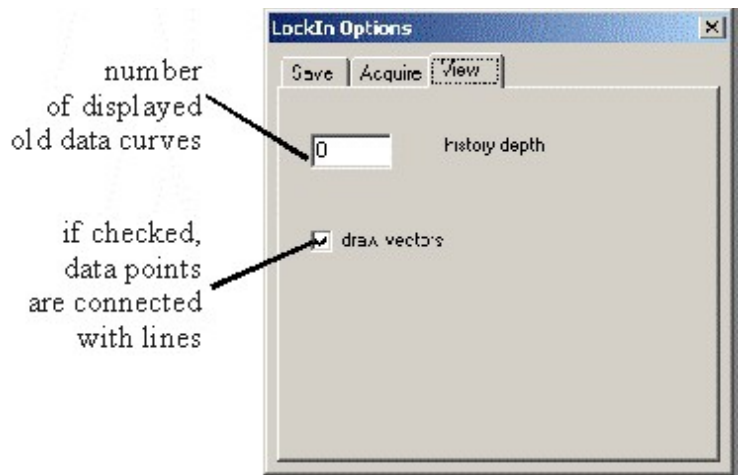


Figure 14: View options for the frequency sweep.

If "**Draw vectors**" is checked, the lines are closed, otherwise data points are drawn as pixel, only.

# DLL DESCRIPTION

The lockin amplifiers in the MultiLockin can be addressed from other programs and LabView™ by calling functions provided in a DLL. In order to make use of these functions as transparent as possible, the following files are provided:

- source code of the DLL
- example in Borland C++ Builder 5.0 for using the DLL
- example in Delphi for using the DLL
- current versions of the DLL

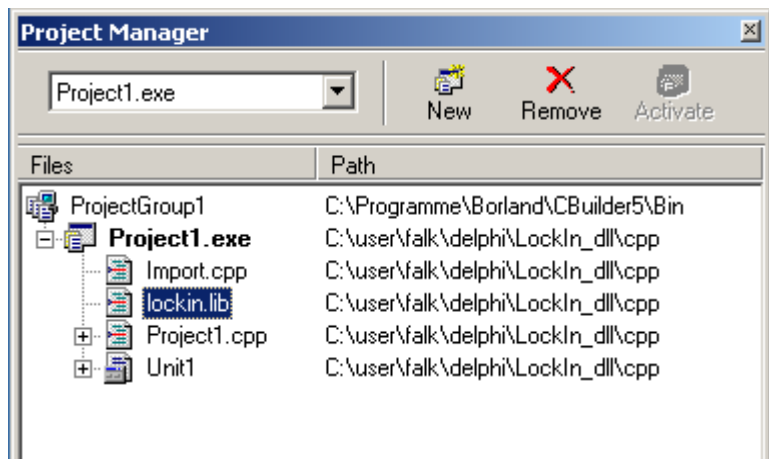
This DLL is compatible with C calling conventions. The convention of the functions in C++ need an underline in front of function names.

Locations of the different DLL file version are described in Section [Installation](#).

## WORKING WITH THE DLL IN C++ PROJECTS

Add the lib-file generated from the DLL to the project in Borland C++ Builder 5.0 by opening 'Project / Add to Project' (see figure).

Chose the **lockin.lib** as the file to be added.



If the lockin.lib does not exist, it can be created from the command line:

- open the command prompt by running 'cmd' in 'Start/Run..' under the Windows Start Menu.
- Go to the directory in which the lockin.dll is located (e.g. '[C:\Windows](#)' )
- type: 'implib lockin lockin.dll'

The lockin.lib is created in the same directory based on the lockin.dll.

To call functions residing in the DLL, headers are necessary, so add 'import.cpp' also to the project, which contains:

```
extern "C" {  
  
double    _declspec(dllimport) cdecl SetLockInFreq (double freq);  
void      _declspec(dllimport) cdecl SetLockInAmpl (double Ampl);  
double    _declspec(dllimport) cdecl SetLockInPhase (double Phase);  
long      _declspec(dllimport) cdecl SetLockInHarm (long Harm);  
long      _declspec(dllimport) cdecl SetLockInPllOn (long PllOn);  
long      _declspec(dllimport) cdecl SetLockInTimeConst (long Time);
```

```

long      _declspec(dllimport) cdecl SetLockInRollOff (long Time);
double    _declspec(dllimport) cdecl GetLockInChannel (long Channel);
long      _declspec(dllimport) cdecl GetLockInStatus;
long      _declspec(dllimport) cdecl SetLockInHardGain (long Gain);
long      _declspec(dllimport) cdecl SetLockInCoupling (long Coupling);
void      _declspec(dllimport) cdecl FillDataArrays (long *length,
double *XData[], double *YData[], double *Rdata[], double
*PhiData[]);
long      _declspec(dllimport) cdecl SetLockInTimeBase (double TimeBase,
long PointsPerDivision);
long      _declspec(dllimport) cdecl IsUSBLockin;
long      _declspec(dllimport) cdecl SetLockInSync;
void      _declspec(dllimport) cdecl SetLockInAUX (long *Channel, long
*Input, double *MaxRange);
char*     _declspec(dllimport) cdecl GetVersion;
long      _declspec(dllimport) cdecl SetUsedDevice (char *device);
long      _declspec(dllimport) cdecl SetLockInPID (long *Channel, long
*Input, double *Offset, double *Kp, double *Ti, double *Td)
}

```

All these functions give the set value back when they are called, except 'SetLockInAmpl' and 'FillDataArrays'. 'SetLockInFreq' gives the set value back, when the system uses the internal oscillator (PLL = Off) or allows the user to read the external frequency, when then PLL is ON. 'SetLockInTimeBase' gives back the IdleCount which is calculated from the given TimeBase and PointsPerDivision. If the setting of the current device succeeded is given back by 'SetUsedDevice' and 'GetVersion' gives back a string that includes the hardware number of the lockin and the versions of sys- and dll-files.

## DLL-FUNCTIONS

### SetLockInFreq

```
function SetLockInFreq (freq : double) : double;
```

this function sets the centre frequency of the lockin to a value in Hz and returns the value of the frequency in Hz. If the PLL is OFF, the return value equals the sent value. If the PLL is ON, this function can be used to read the external frequency value.

### SetLockInAmpl

```
procedure SetLockInAmpl (Ampl : double);
```

sets the output amplitude of the reference output in V and gives no value back.

### SetLockInPhase

```
function SetLockInPhase (Phase : double) : double;
```

sets the phase offset in degree and returns this phase offset in degree.

### SetLockInHarm

```
function SetLockInHarm (Harmonic : integer) : integer;
```

sets the harmonic at which the signal is analysed. The harmonic can vary between 1 and 15. It returns the harmonic value.

### SetLockInPllOn

```
function SetLockInPllOn (PllOn : integer) : integer;
```

switches the PLL on (1) or off (0) and gives the state of the PLL back. The state 1 stands for "PLL ON", while the state 0 stand for "PLL OFF". When using the external trigger, please make sure, that later calls of SetLockInHarm(H) or SetLockInPhase do not reset the state of the PLL to zero.

Starting from software package Version 1.05, the functions "SetLockInPhase,", "SetLockInHarm", "SetLockInFreq" and "SetLockInAmpl" do not overwrite the PLL setting anymore.

### SetLockInTimeConst

```
function SetLockInTimeConst (TimeConst : integer) : integer;
```

sets the time constant for the low pass filter.

TimeConst =

0	10 $\mu$ s
1	20 $\mu$ s
2	50 $\mu$ s
3	0.1 ms
4	0.2 ms
....	
17	5 s

### SetLockInRollOff

```
function SetLockInRollOff (RollOff : integer) : integer;
```

sets the RollOff for the low pass filter:

0	6 dB/oct
1	12 dB/oct
2	24 dB/oct

### GetLockInChannel

```
function GetLockInChannel (Channel : integer) : double;
```

reads the results in the channel number x, where x is:

0	real part = X
1	imaginary part = Y
2	amplitude = R
3	phase = phi

It gives the read value of this channel in V back.

Note: It is necessary to call **SetLockInHardGain** first.

### GetLockInStatus

```
function GetLockInStatus : longint;
```

Returns a value in which bit 1 (corresponding to  $2^1$ ) equals 1, if the input of the lockin amplifier is in overload state. The values of the other bits might be undefined.

### SetLockInHardGain

```
function SetLockInHardGain (Gain : integer) : integer;
```

Switches the input gain of the lockin between

1 high reserve  
10 normal  
100 low noise

#### NOTE:

**This function "SetLockInHardGain" should be called at least once at the beginning of a program. Otherwise, the scaling of the R, X, and Y as well as of the outputs Aux1 and Aux2 might be wrong.**

### SetLockInCoupling

```
function SetLockInCoupling (Ac : integer) : integer;
```

Switches the input coupling of the lockin between ac mode and dc mode

1 ac coupling  
0 dc coupling

### FillDataArrays

```
procedure FillDataArrays (pLength: PInteger; XData, YData, RData, PhiData: PArray);
```

Fills the given arrays with the corresponding X-, Y- and R-Data. The length of all four arrays must be 1024 (0...1023). LabView should reserve the memory and provide the pointer to these arrays for the DLL. The parameter "*Length*" is to the number of indexes of the array that were actually filled.

### SetLockInTimeBase

```
function SetLockInTimeBase(timeBase: Double): Double;
```

the input to this function "*timeBase*" is the time between two data points in seconds. The function returns the used time between data points in seconds.

### IsUSBLockin

```
function IsLockInUSB: Longint;
```

This function returns "1", if the connected device is a USB-LockIn250 and "0" if it is not.

### SetLockInSync

```
function SetLockInSync (Sync: Longint): Longint;
```

This function allows to set the number of sync cycles for synchron filtering mode and returns the value set.

### SetLockInAux

```
procedure SetLockInAUX(Channel, Input: Longint; MaxRange : Double);
```

```
HW SerNr. :0  
sys: 0.07.0.3  
dll: 1.0.6.2
```

Example result

The USB-LockIn250 has two analogue auxiliary outputs named with "Aux1" and "Aux2" – see also page Error: Reference source not found. The parameter "*channel*" selects which connector is used (0 ↔ Aux1, 1 ↔ Aux2). The parameter "*input*" selects the type of signal that is provided at this

output (In ↔ 0 , X ↔ 10, Y ↔ 11, R ↔ 12, Phi ↔13). "MaxRange" scales the signals according to the amplitude at the selected input channel. MaxRange of Phi is always 180° while for the other channels it can be any value between -10 V and 10 V.

Examples:

SetLockInAux(1,10,0.01) → provides the measured value of X at the output AUX2 scaled to 10 mV.

SetLockInAux(0,1,2.35) → Aux1 is set to 2.35 V.

If the parameter "input" is set to 1, then "MaxRange" is used as output voltage.

Examples:

SetLockInAux(1,1,5.32) → sets the output of Aux2 to 5.32 V

## SetUsedDevice

```
function SetUsedDevice (FullName: PChar) : longbool;
```

Each lockin amplifier inside this device can be selected by its ID. As all of them are connected through one USB hub, the IDs start with an identical number and count up for each unit. This command allows to enter a specific device information so that this multiple lockins can be addressed independently. (see also page 38)

## GetVersion

```
function GetVersion : PChar;
```

Returns the hardware serial number (e.g. 280007), the version number of the driver (e.g. 0.07.0.3) and the version of the dll itself (e.g. 1.0.6.2).

```
HW SerNr. :0  
sys: 0.07.0.3  
dll: 1.0.6.2  
Example result
```

When the DLL is initialized, it sets the following default values:

Phase	0 degree
Harmonic:	1
PLL.:	off
Time constant:	10 ms
RollOff:	12 dB
Input Gain:	1 (high reserve)

## SetLockInPID

```
function SetLockInPID (Channel, Input : LongInt; Offset, Kp, Ti, Td:  
Double):boolean;
```

This functions sets the parameters for PID control.

The parameter "*channel*" selects which connector is used (0 ↔ Aux1, 1 ↔ Aux2).

The parameter "*input*" selects the type of signal that is used as input for the PID control (In ↔ 0 , X ↔ 10, Y ↔ 11, R ↔ 12, Phi ↔13).

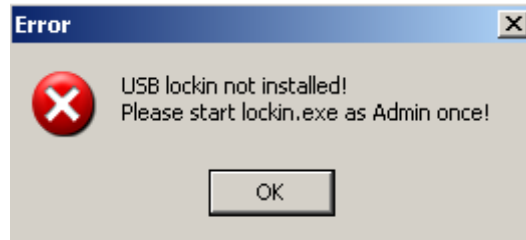
The parameter "*Offset*" equals the set-point. "*Kp*", "*Ti*" and "*Td*" are the feedback parameters (see page Error: Reference source not found).

## EXAMPLE PROGRAMS

Currently, there is a program examples provided in Borland Delphi 6.0 (demo.exe). It can be found in the directory %YourPath/DLL/Delphi

# REMOTE CONTROL WITH LABVIEW

**Note: LabView and DLL-functions will not run properly, when the test software "lockin.exe" never has been started as Administrator.** It requires properly set entries in the registry for the right scaling. When LabView is started before the entries in the registry exist, the following error message appears:



## FILES AND LOCATIONS

The LabView driver is based on the provided DLL file "lockin.dll". The 'lockin.dll' is best located in the same directory as the vi-files or in any path that is known to LabView as library path.

Anfatec has prepared LabView examples for LabView6.1, LabView11 and saved some special applications also for Labview9.

Nine different vi-files are provided, the first four in two versions LabView6.1 and LabView11, and all other just for LabView11:

Output_Lockin.vi	read the four output channels X, Y, R and Phi
Input_Lockin.vi	sets the input and internal parameters of the lockin amplifier
Named_Input_Lockin.vi	this is required to operate the single lockin amplifiers in this Multiple Lockin Amplifier Unit simultaneously
Frequency_Response_LockIn.vi	acquires a spectrum of R and Phi
SetAUX_Simple.vi	shows how the AUX output channel reacts to different maximum ranges
AUXRamp_Lockin.vi	demonstrates how one can create a ramp at the AUX output channels
SetAUX_Lockin.vi	has the same functionality as the "lockin.exe"
GetUnitAndRangeFromString.vi	divides a supplied string into physical unit and range
Oszi_LockIn.vi	shows X, Y, and R in one oscilloscope screen

Additionally, in LabView 9, a special application for a phase feedback called "Nose" has been programmed. (see page ).

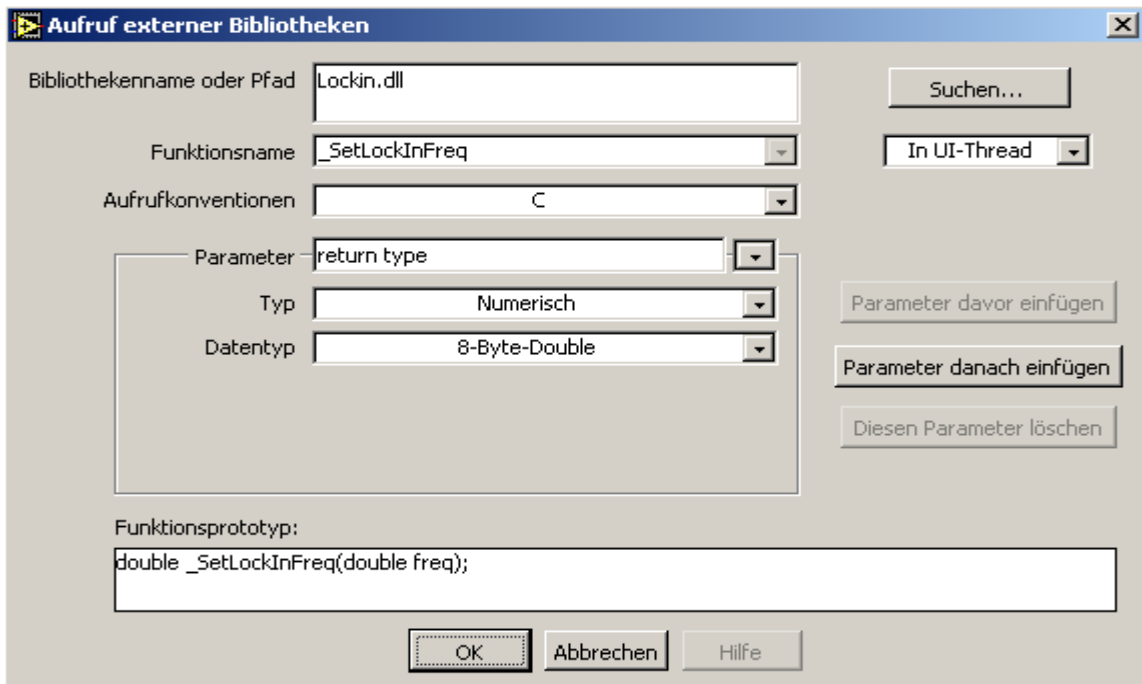
## GENERAL PROGRAMMING DIRECTIONS IN LABVIEW



Switch into the diagram view of the vi-files. The parameters are set and read with the DLL functions or double click into these functions to open the window, which allows to correlate the function with a DLL-function and its exchange value:

The used library is the 'lockin.dll'. The example sets the frequency with the function 'SetLockInFreq'.

The value is of numerical type and 8-Byte Double long.



Needed data types in the DLL and their equivalent in LabView are:

- double numerical type, 8-Byte-Double
- integer numerical type, signed 32-Bit-Integer
- longint numerical type, signed 32-Bit-Integer
- PInteger numerical type, signed 32-Bit-Integer (Pointer to value)
- PdataArray array type, 8-Byte-Double (1 dimension, array pointer)
- longbool numerical type, signed 32-Bit-Integer
- PChar string type, pointer to C-String

The frequency sweep example, for instance, consists of a loop, in which each single frequency value in the spectrum is calculated based on the input parameters 'Start Frequency', 'Stop Frequency' and 'Number of Steps', while the results are stored and displayed in a graphical window.

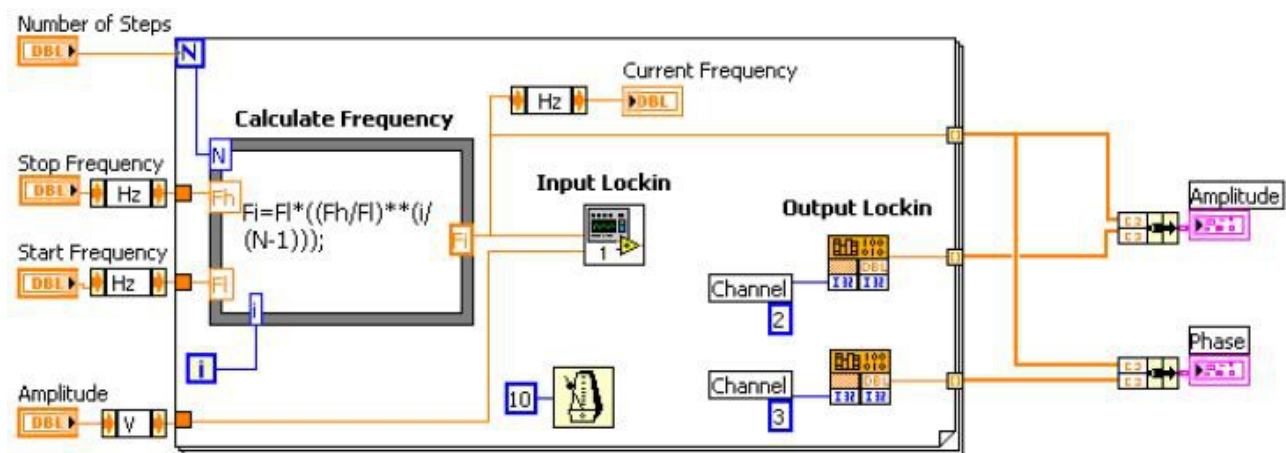


Figure 15: Diagram of the Frequency\_Response\_Lockin

Both sub-vi's 'Input\_Lockin' and 'Output\_LockIn' are used in the frequency sweep. They only connect the numerical data and DLL function.



# DESCRIPTION OF EXAMPLE VI FILES

The file **Output\_Lockin.vi** reads the channels 0 to 3. Channel 0 equals X, channel 1 equals Y. Channel 2 and 3 are R and Phi, respectively. The numbers given here are in the unit shown behind the numbers.

The file **Input\_Lockin.vi** allows to set the main parameters of the lockin. Input Gain, Time Constant and RollOff are given as integers. The relation between the value and the integer is described together with the DLL. Frequency, Harmonic, Amplitude and Phase are of the numeric type 'Double'. They are used unit-less in this file, but their units are meant in Hz, Volts and Degree, respectively.

Beside the frequency input, the current frequency is displayed. Also, an additional switch allows to enable the PLL function. When the PLL is ON, this frequency display shows the external frequency, onto which the system is locked. If the PLL is OFF, it shows the same frequency that the user has set in the frequency input.

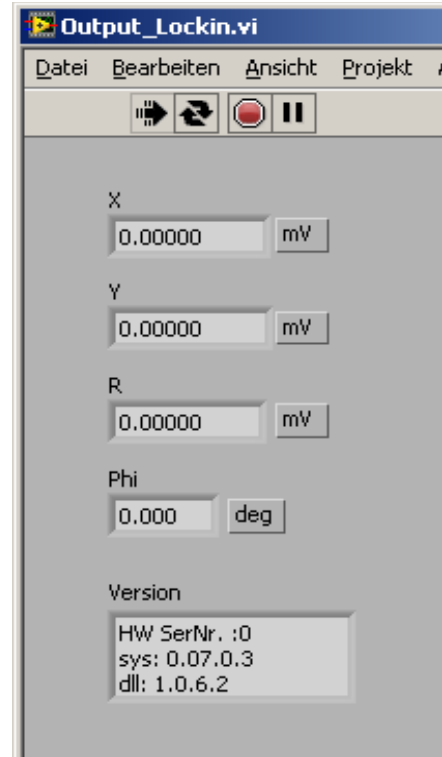


Figure 16: Frontpanel of the Output\_Lockin.vi

For the use of **Named\_Input\_LockIn.vi** please read the chapter "Address the Multiple LockIn Amplifiers in LabView" on page 38.

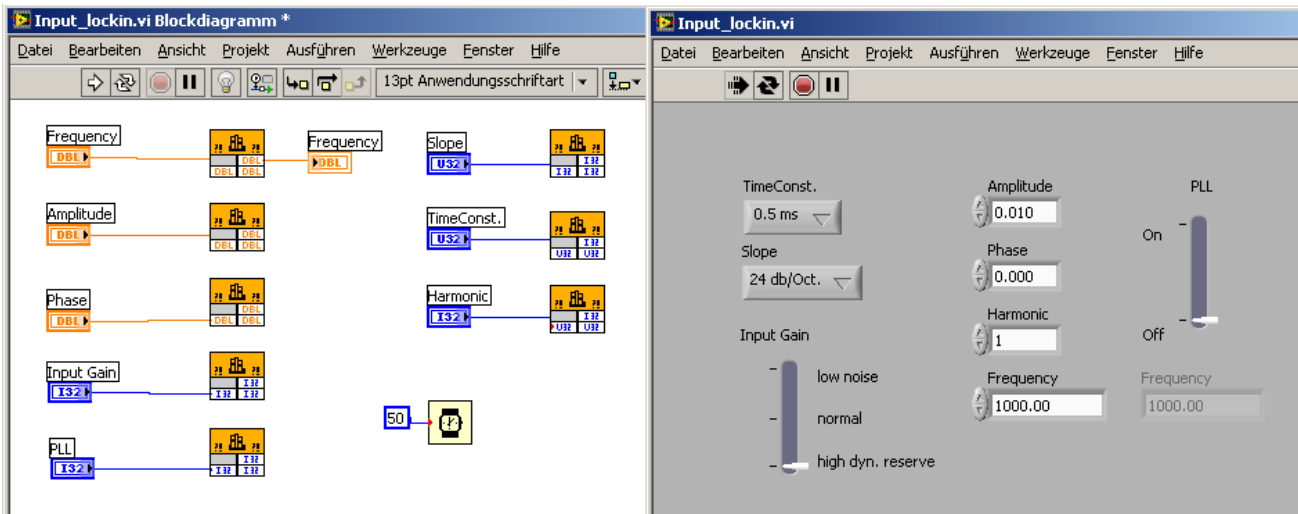


Figure 17: Frontpanel of the Input\_Lockin.vi

The **Frequency\_Response\_LockIn.vi** is one possible example which shows how spectra might be acquired with the lockin amplifier in LabView. The user provides the corner data for the spectra acquisition (Start Frequency, Stop Frequency, Number of Values). When the VI-file runs, one spectrum is acquired. The data values are set here with units. The Output Amplitude in this front panel equals the Amplitude in the Input\_LockIn.vi. During the spectra acquisition, the current frequency is monitored.

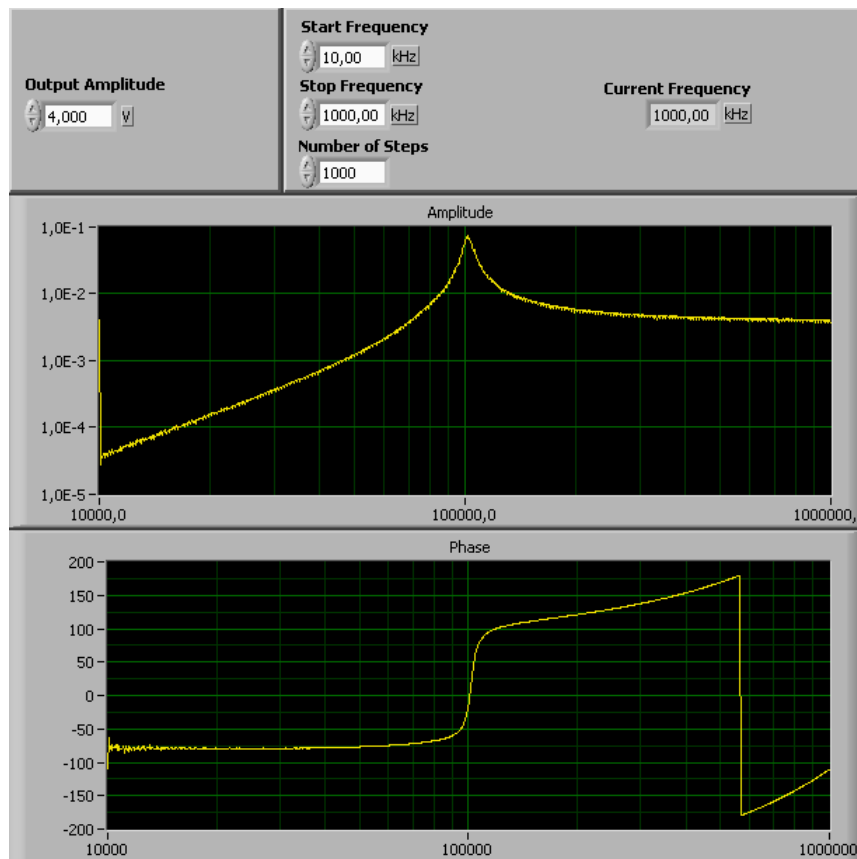


Figure 18: Frontpanel of the Frequency\_Response\_Lockin.vi

**SetAUX\_Simple.vi** uses the DLL-function "SetLockInAUX" to set the AUX0 channel to the selected maximum range. Therefore, it uses the SubVI "GetUnitAndRangeFromString.vi" to get MaxRange in Volts from the string provided by the "Display Range"-Combobox (e.g. "0.01" from "10 mV"). With an Output Amplitude of 10 mV, the AUX0 output at a display range of 20 mV should be 5 Volts. For channel number 13 (Phi) the maximum range is always 180 degrees. To get different AUX outputs here, use the Phase offset in the LabView-Frontpanel.

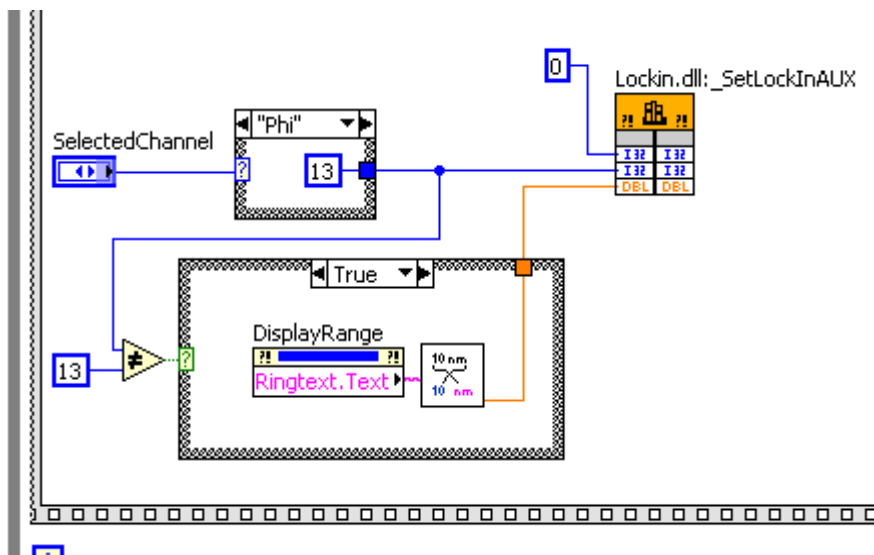
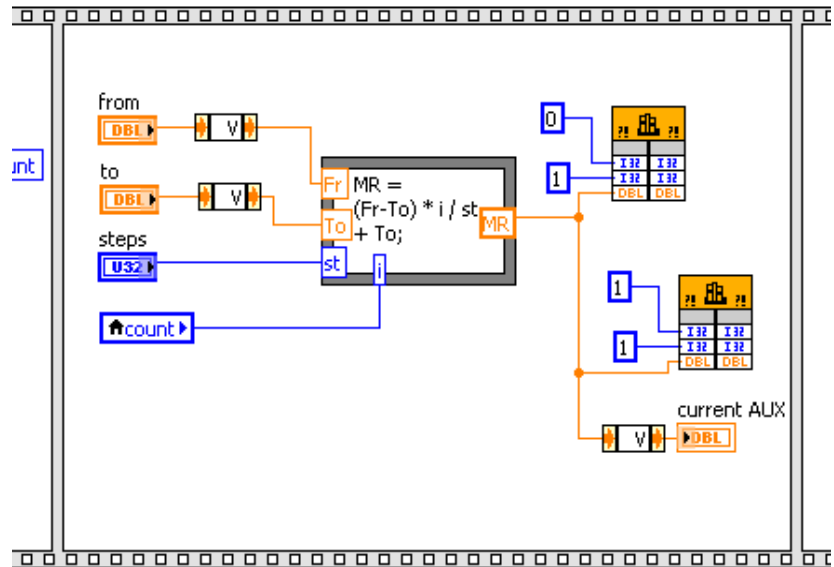


Figure 19: Part of the Block Diagram of the SetAUX\_Simple.vi

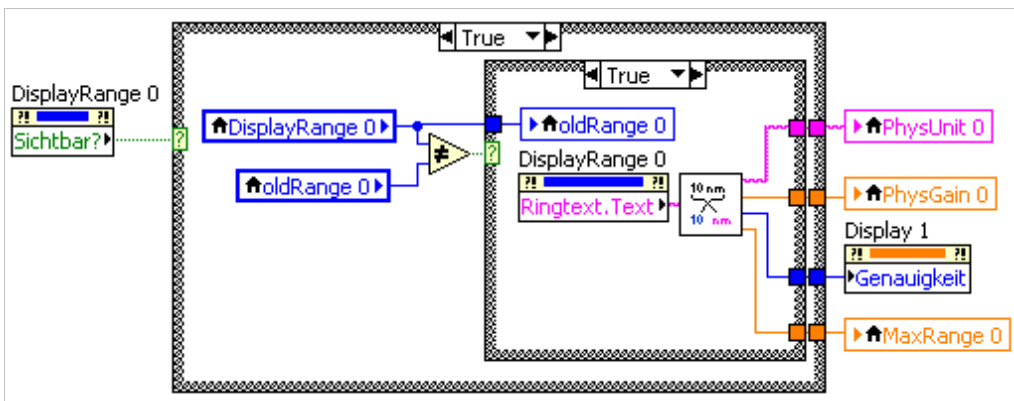
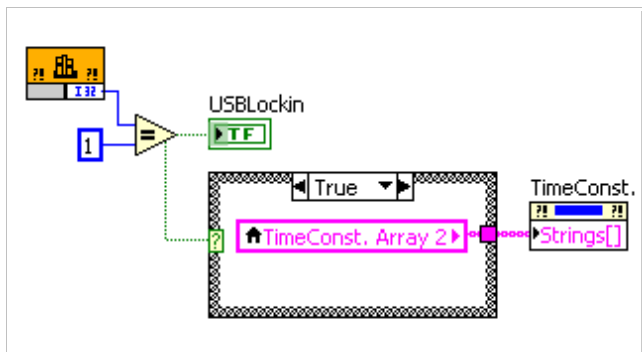
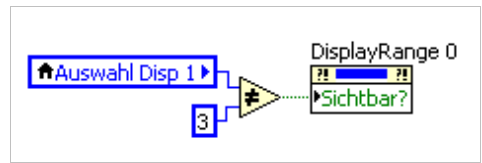
**AUXRamp\_Lockin.vi** uses output channel 1 to create a ramp at AUX channel 0. The boundaries are given with "from" and "to". With these parameters and a supplied steps variable the formular knot calculates a MaxRange for each step at "count" and gives back the "current AUX" in Volts.

• Figure 20: Part of the Block Diagram of the AUXRamp\_Lockin.vi

The example "SetAUX\_Lockin.vi" is not valid for the MultiLockIn Amplifier, as there are no Auxiliary Outputs available.

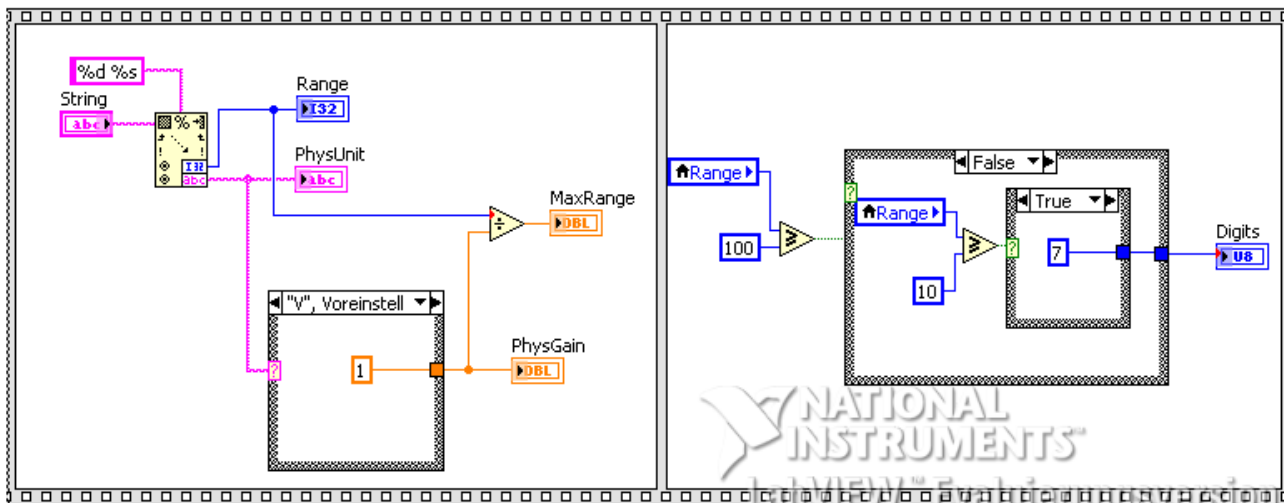


If the chosen output channel is "Phi" the "Display Range"-Combobox is not shown because the maximum range for Phi is always 180 degrees. For the other channels the MaxRange is calculated via the SubVI "GetUnitAndRangeFromString.vi".

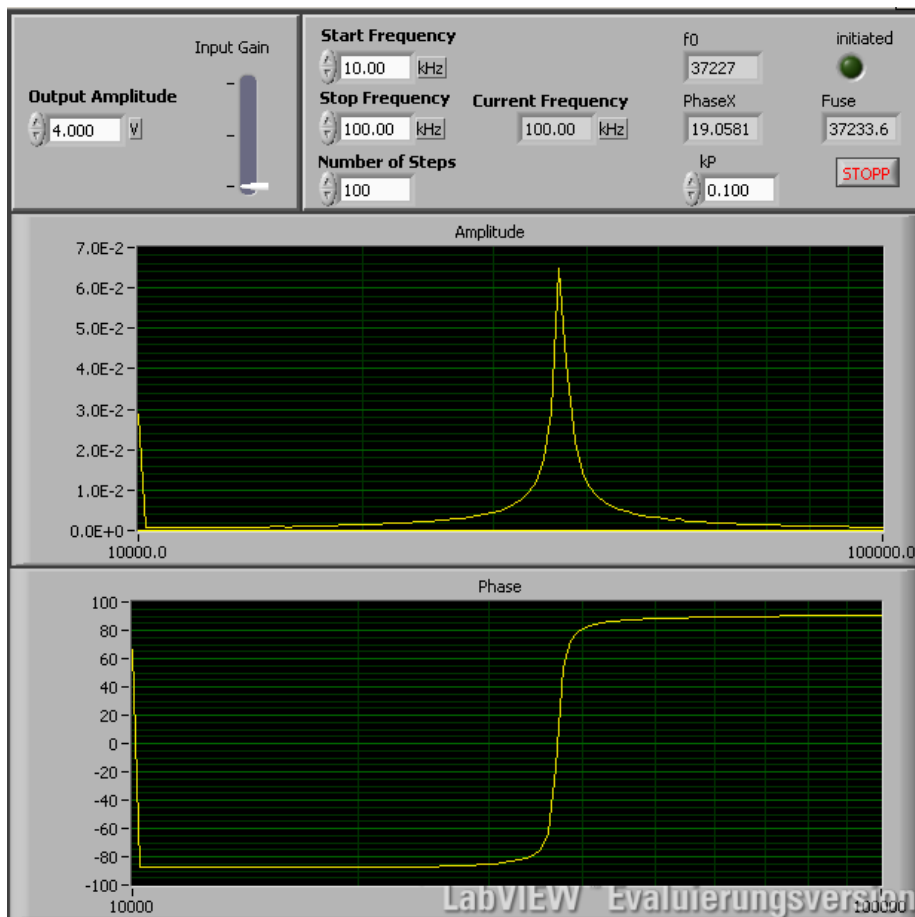


The SubVI **GetUnitAndRangeFromString.vi** cuts a string consisting of value and unit into these

two parts. The value from this operation (Range) is divided by the physical gain that is supplied by the physical unit (e.g. PhysGain is 1000 for a PhysUnit of "mV"). The count of digits to be shown in "SetAUX\_Lockin.vi" is calculated in the second part of the VI. It depends on the Range of the supplied String.



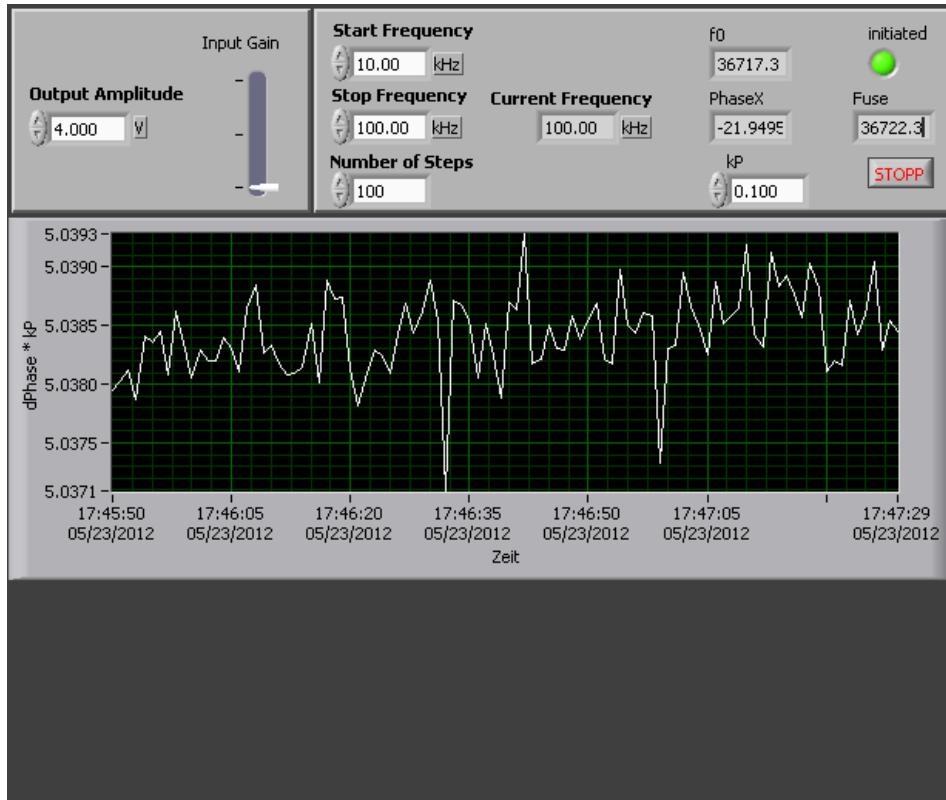
The VI "**Nase\_LockIn.vi**" (in LabView9 folder, only) consists of two part. First comes an initialization. Start the VI after you connected everything and selected an output amplitude, the input gain as well as the start and stop frequency and the number of steps to be taken. The VI



now takes a spectrum with the selected settings and shows the resulting amplitude and phase in the diagrams.

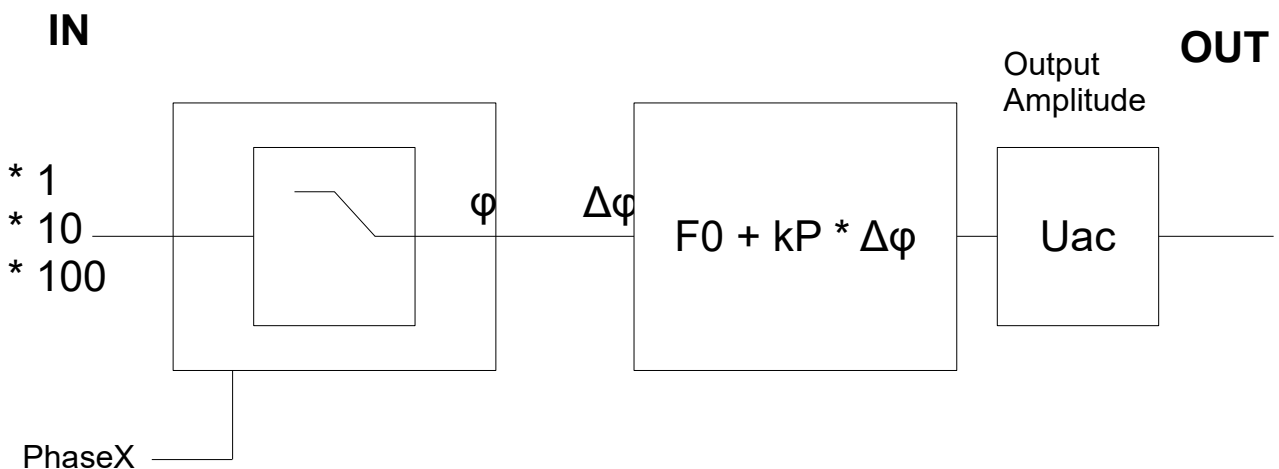
The corresponding block-diagram shows how the frequencies are calculated for each step and set via the VI "Input\_LockIn.vi". Then the channels 2 and 3 are polled for getting the amplitude and phase at the actual frequency.

For the second step, please click with the left mouse button into the amplitude-graph at the resonance frequency. This frequency will be set and the phase at this point will also be set as the reference phase. These two values will be shown as "f0" and "PhaseX" accordingly. Now the two diagrams become invisible and a new graph appears. The new diagram shows the new phase multiplied with the chosen  $K_p$  over the passing time, like shown in the picture below:




$K_p * \text{Phase}$  will be added to the set frequency "f0" and stored in "F<sub>use</sub>" to set the new used frequency of the LockIn amplifier.

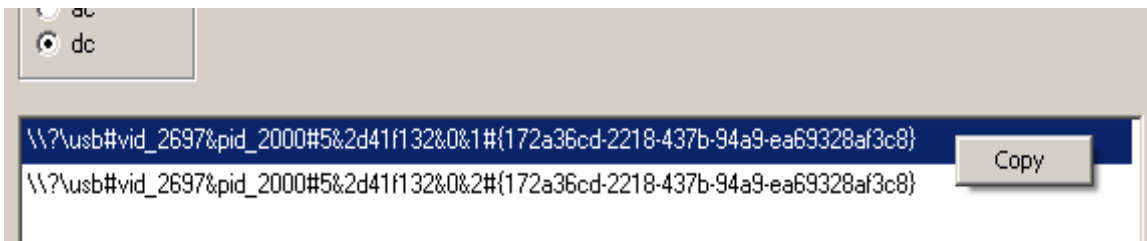
The corresponding scheme of this nose application is shown below:



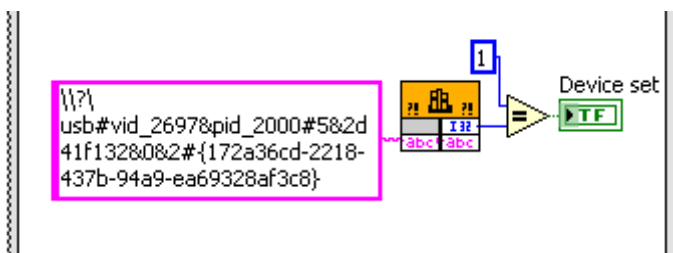
# ADDRESS THE MULTIPLE LOCKIN AMPLIFIERS IN LABVIEW

In order to address the single amplifiers of teh MultiLockIn 7x250 unit, one has to follow these steps:

- generate a copy of the "Lockin.dll" with a different name (e.g. Lockin2.dll)
- open "Named\_Input\_lockin.vi" in LabView
- double-click on each 'external library'-symbol  and change the used library to the new ".dll"
- open the program "Lockin.exe"
- enable "View > Devices" (if not already shown)
- right-click on the device string that corresponds to the chosen USBLockIn250 and copy it



- paste the copied string into the input of the function "SetUsedDevice" in the vi-file



Now, this ".vi" is set specifically for the chosen lockin. Repeat the above steps to create more ".dll" and ".vi" for the other lockin amplifier units.

(Don't forget to close the 'Lockin.exe' again for testing the '.vi'!)

# REVISION HISTORY

## DRIVER REVISIONS

June 2018

- 0.7.0.9 sys

This is the first driver that acts as exclusive driver. Only **one** application can access the driver at a time.

## HARDWARE REVISIONS

no revisions yet

## CHANGES IN THE MANUAL

This manual was generated based on the Manual of the USB lockin amplifier USBLockIn250. Not all functions described here may apply to the multi-Lockin amplifier.

## • **Alphabetical Index**

• Address the Multiple LockIn Amplifiers in LabView.....	36
• Amplitude.....	20
• Connectors.....	10
• Coupling.....	20
• Description of Example VI Files.....	31
• DLL.....	24
• DLL in C++ projects.....	24
• DLL-Functions.....	25
• Driver Installation.....	9
• Driver Update.....	9
• Dynamic.....	19
• Example programs.....	28
• File.....	17
• Files and Locations.....	29
• Frequency.....	20
• Frequency Sweep.....	20f.
• Functions in the Function Line.....	17
• Functions in the Menu Line.....	17
• General Information.....	4
• Harmonic.....	20
• Help.....	17
• Input.....	4
• Installation.....	6
• LabView.....	29
• Licence for the software.....	5
• Locations.....	29
• LockIn.....	11
• LockIn Amplification Basics.....	11
• Mathematical description.....	12
• Menu Line.....	17
• Meters.....	20
• Noise Measurements.....	13
• Option.....	17
• Options for the frequency sweep.....	21
• Oscilloscope.....	17
• Output.....	5
• Parameter settings.....	19
• Phase.....	20



- Provided Software.....6
- Reference Output.....5
- Remote Control with LabView.....29
- Revision History.....37
- RollOff.....19
- settings.....19
- Signal Input.....4
- Software Description.....16
- Software Installation.....7
- Specification of the M.....4
- Sweep Frequency.....20f.
- System requirements.....6
- Time constant.....19
- View.....17