TECHNISCHE UNIVERSITÄT WIEN

Diplomarbeit

Process Control for a Large Area High Resolution Hall Scanner

Ausgeführt am

Atominstitut der TU Wien

unter Anleitung von

Priv. Doz. Dipl.-Ing. Dr. techn. Michael Eisterer

durch

David Bader, BSc

Matr. Nr.: 1026256

Hauswaldgasse 10
2620 Neunkirchen

Wien, 25. Juli 2017

Abstract

The goal of this thesis was to build the software for a new nano Hall scanner with an interferometry system for positioning feedback. The software is responsible for controlling the position and movement of the Hall probe of this scanner, as well as communicating with and controlling all other devices used for the scanning system. Also part of this work was setting up and connecting these devices with the controlling PC. Additionally, the software was used to measure and plot interference patterns to help setting up the interferometry system correctly.

Kurzfassung

Das Ziel dieser Arbeit war, die Software für einen neuen Nano-Hallscanner mit einem Interferometriesystem zur Positionsregelung zu schreiben. Diese Software ist dafür verantwortlich, die Position und Bewegung der Hallsonde des Scanners zu steuern und zu regeln, sowie mit allen anderen Geräten, die für das Scanner-System verwendet werden, zu kommunizieren und diese zu steuern. Ein weiterer Teil dieser Arbeit war der Aufbau dieser Geräte und deren Verkabelung mit dem Steuerungs-PC. Weiters wurde die Software dazu verwendet, um Interferenzmuster zu messen und darzustellen und damit den Aufbau des Interferometriesystems zu unterstützen.

Contents

Introduction

The molecular structure of high-temperature superconductors (HTS) can contain a lot of inhomogeneities that have an impact on their superconducting properties. Magnetic flux enters a type-II superconductor in the form of quantized vortices with normal conducting cores, that experience a Lorentz force caused by a current in the superconductor. This force induces the vortices to move, causing energy dissipation, which determines the resistance of the superconductor. However, the Lorentz force can be counteracted by effectively pinning the vortices to pinning centers consisting of defects in the material.

The understanding of pinning centers and the structure of the flux-line lattice plays an important role in understanding and improving the superconducting properties of HTS. The most significant pinning centers are grain boundaries, however, small fissures in the µm range or other crystallographic defects like dislocations or stacking faults can also play an important role.

A useful measurement technique for better understanding these influences is Scanning Hall Probe Microscopy (SHPM): The externally magnetized superconductor is scanned with a Hall probe (see section 1.1), in order to measure the local magnetic field with a high spacial resolution. With this data one can calculate the local critical current density J_c using the Maxwell equation

$$\vec{\nabla} \times \vec{H} = \vec{J_c}. \tag{1}$$

With a high enough spatial resolution it is also possible to see the flux-line lattice, which can yield important information about the pinning centers as mentioned above, especially, if the scan range is large.

There are already two Hall scanners in our laboratory: One is a large sample scanner that can cover a range of $30 \times 10\,\mathrm{cm}^2$, with a spatial resolution of $50\,\mu\mathrm{m}$, which is used for experiments at liquid nitrogen temperature. The second one is a micro Hall scanner with a resolution of $1\,\mu\mathrm{m}$ and a smaller scan range of $3 \times 3\,\mathrm{mm}^2$ that can be inserted into an $8\,\mathrm{T}$ cryostat and therefore operate down to liquid helium temperatures.

The new nano Hall scanner will combine the advantages of both these scanners in order to get a scanner with a large scan range of approximately $27 \times 17\,\mathrm{mm}^2$ and a high spatial resolution of 50 nm at various temperatures. This is accomplished by using two different piezo actuators for each axis, one coarse actuator for moving large distances, and one fine actuator for accurate positioning.

The nano Hall scanner will utilize an interferometry setup (section 2.2.7) to accurately measure the position of the Hall probe. This position information is fed back into the controlling software in order to enable more precise positioning than with the previous scanners.

The large scan range brings the obvious benefit of being able to scan large samples, while maintaining accuracy because of the double axis setup. With its high resolution, the scanner will be able to measure the magnetic fields in individual grains and at their boundaries. This is not always possible with the existing scanners, because typical grain sizes vary between 100 nm and 50 µm, depending on the material, so the spacial resolution of the existing scanners is often not sufficient for this purpose. These grain boundaries can have a boosting, but also a limiting effect on the achievable critical current J_c of a superconductor, as mentioned previously. It will also be possible to see individual flux-lines with this new scanner, which would provide interesting information about their pinning as explained above.

Section 1 of this thesis describes and explains the physical basics and effects that are needed for the scanning process. Section 2 describes the hardware setup of the new nano Hall scanner. This section also contains a description of all devices used in the scanner system. A description of interference experiments for the interferometer can be found in subsection 2.3. Section 3 contains a detailed description of the user interface of the scanning software and explains the handling of the application. Section 4 describes the architecture of this software and the interaction of its modules, as well as the scripting capabilities of the software. Lastly, section 5 contains a short outlook for the next steps that have to be taken to get the new scanner up and running.

1 Physical Basics

This section describes and explains the physical basics and effects that are needed for the scanning process. First, the Hall effect that enables accurate measuring of a magnetic field is explained. The next part describes the working principle of the piezo actuators used for moving the cantilever and the sample, and the last part of this section discusses the function of a PID controller, which we use for controlling the temperature of the sample.

1.1 The Hall Effect

The Hall effect can be used for measuring magnetic fields very accurately. The basis of the effect is the generation of a voltage difference across a current carrying electric conductor in a magnetic field. This voltage difference is perpendicular to the current and the magnetic field and is caused by the Lorentz force on the charge carriers in the conductor

$$\vec{F} = q\vec{v} \times \vec{B}. \tag{1.1}$$

This force causes the charge carriers to move on a curved path instead of a straight one, so charges accumulate on one side of the conductor. This in turn creates an electric field $\vec{E}$ in the conductor that balances the Lorentz force, such that

$$q(\vec{E} + \vec{v} \times \vec{B}) = 0. \tag{1.2}$$

With $\vec{v}$, $\vec{E}$ and $\vec{B}$ perpendicular to each other, in x-, y- and z-direction respectively, this can be simplified to a non-vector equation

$$E_y - v_x B_z = 0, \tag{1.3}$$

where $\vec{v} = (v_x, 0, 0)$ and $\vec{B} = (0, 0, B_z)$, which means that $\vec{I} = (I_x, 0, 0)$ and $\vec{E} = (0, E_y, 0)$. With $\vec{j} = nq\vec{v}$, $\vec{j} = \frac{\vec{I}}{t \cdot w}$ and $E_y = \frac{V_H}{w}$, one gets

$$V_H = A_H \frac{I_x B_z}{t}, \tag{1.4}$$

where A_H is the Hall constant, V_H is the Hall voltage, t is the thickness of the probe in z-direction and w the width in x-direction, so $t \cdot w$ is the cross sectional area of the Hall probe, $\vec{I}$ is the electric current and $\vec{j}$ is the current density. For the simple case of just one type of charge carriers, the Hall constant is $A_H = \frac{1}{nq}$. From this Hall voltage, which in our case is measured with the nanovoltmeter described in section 2.2.2, one can calculate the magnetic field by using equation (1.4).

1.2 The Piezoelectric Effect and Piezo Actuators

The piezoelectric effect can occur in crystals with an asymmetric charge distribution in the unit cell. By applying mechanical stress to the crystal, the positive and negative charge concentrations no longer coincide, which generates a measurable electric field in the crystal. The effect is reversible, such that the application of an electric field causes a mechanical deformation of the crystal. This so-called converse piezoelectric effect makes it possible to build piezoelectric actuators that can be used for movement with sub nm accuracy. If a voltage V is applied to a piezoelectric crystal, this voltage causes a change Δl in the length of the crystal

$$\frac{\Delta l}{l_0} = d \cdot E = d \cdot \frac{V}{l}, \tag{1.5}$$

where l_0 is the unperturbed length of the crystal, d is the piezoelectric constant, depending on the material and the temperature, E is the electric field created by V and $l = l_0 + \Delta l$ is the current length of the crystal. With $\Delta l \ll l_0$ this can be simplified to

$$\Delta l = d \cdot V, \tag{1.6}$$

$$\Delta l_{2,1} = d \cdot \Delta V, \tag{1.7}$$

which gives a linear correspondence of an applied voltage difference $\Delta V = V_2 - V_1$ with the resulting length difference of the piezoelectric crystal $\Delta l_{2,1} = \Delta l_2 - \Delta l_1$ caused by ΔV.

The maximal expansion of the crystal and consequently the movement range of a piezoelectric actuator is relatively small, in the range of µm. To get piezo actuators with a larger movement range, it is possible to use the stick-slip-effect and apply the piezo to power a stepping motor, as depicted and explained in figure 1.1. By applying a sawtooth-like voltage wave form to the piezo, it expands slowly and then contracts very fast, or vice versa. Because of the high inertia forces caused by the fast movement, the motor can overcome the static friction and the actuator moves in the regime of the lower kinetic friction, while the slow movements cannot overcome the static friction and reset the actuator for a new step. These motors can be built as linear as well as rotational actuators.

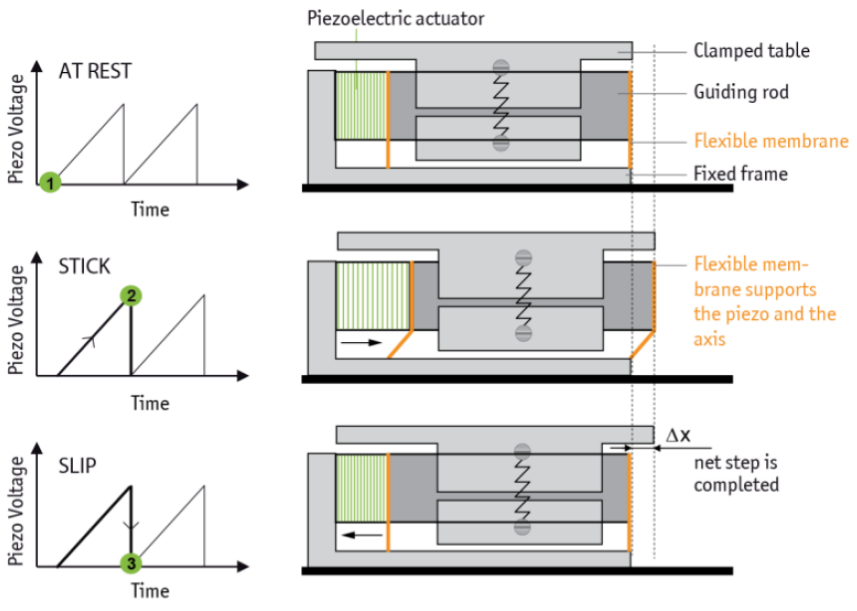


Figure 1.1: Working principle of a piezo stepping motor using the stick-slip-effect, taken from [4]. The moving table is clamped to a guiding rod, which is connected to the piezo actuator. At (1) the motor is at rest. During the slow flank of the sawtooth voltage until point (2) the clamped table sticks to the guiding rod and is moved over a distance Δx. During the steep flank of the sawtooth voltage at (3), the inertia of the table overcomes the friction because of the rapid acceleration of the guiding rod, such that the table disengages from the rod and remains nearly stationary. Now the table has moved the distance Δx, ready for another step.

1.3 PID Controller

A PID controller (Proportional-Integral-Derivative controller) is a control loop feedback mechanism. It continuously compares a measured process variable to a desired setpoint and calculates an error $e(t)$ as difference of these two. From this error value the value of a control variable $u(t)$ is calculated to minimize the error:

$$u(t) = P \cdot e(t) + I \cdot \int_0^t e(\tau)\,\mathrm{d}\tau + D \cdot \frac{\mathrm{d}e(t)}{\mathrm{d}t}. \tag{1.8}$$

For temperature control as used in our case (see section 2.2.4), $u(t)$ is the power supplied to the heater, in general however, $u(t)$ could be any control mechanism like the position of a valve or the speed of a motor. P, I and D are the coefficients for the proportional, integral and derivative terms, respectively, that determine the behavior of the controller. Not every part of the sum in equation (1.8) is needed for all applications, each part can be "disabled" by setting the according coefficient to zero. The controller is then called a P, PI, PD, or I controller, depending on which coefficients remain non-zero. A pure differential controller is not used, because it does not yield a useful control variable on its own.

2 The Nano Hall Scanner Setup

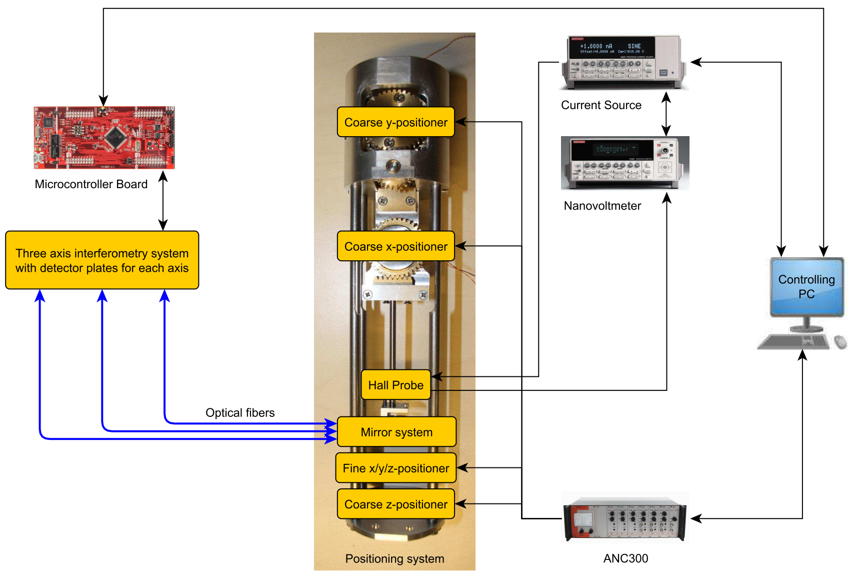


Figure 2.1: System overview of the nano Hall scanner with the interferometry system, the positioning system, the controlling PC and the devices relevant for measuring and controlling the position of the Hall probe.

Figure 2.1 shows a schematic overview of the whole nano Hall scanner setup, while figure 2.2 shows a photo of the scanning system with all its devices. The six piezo actuators of the positioning system are controlled by an Attocube ANC300 Piezo Positioning Electronic (see section 2.2.1). The position of the Hall probe is measured by an interferometer that is being built at the moment (section 2.2.7). A combined system of a Keithley 6220 Precision Current Source and a Keithley 2182A Nanovoltmeter is used to accurately supply and measure the output of the Hall probe (sections 2.2.2 and 2.2.3). The temperature of the cryostat is monitored and controlled by a Cryocon Model 32B Temperature Controller (section 2.2.4), while the liquid helium level is monitored by a Cryomagnetics LM-500 Liquid Helium Gauge (section 2.2.5). Lastly, the superconducting magnet in the cryostat is powered and controlled by a Cryomagnetics 4G Magnet Power Supply (section 2.2.6).

The whole system is controlled from a PC by a software written in Python, described in sections 3 and 4, which is the main part of this thesis.

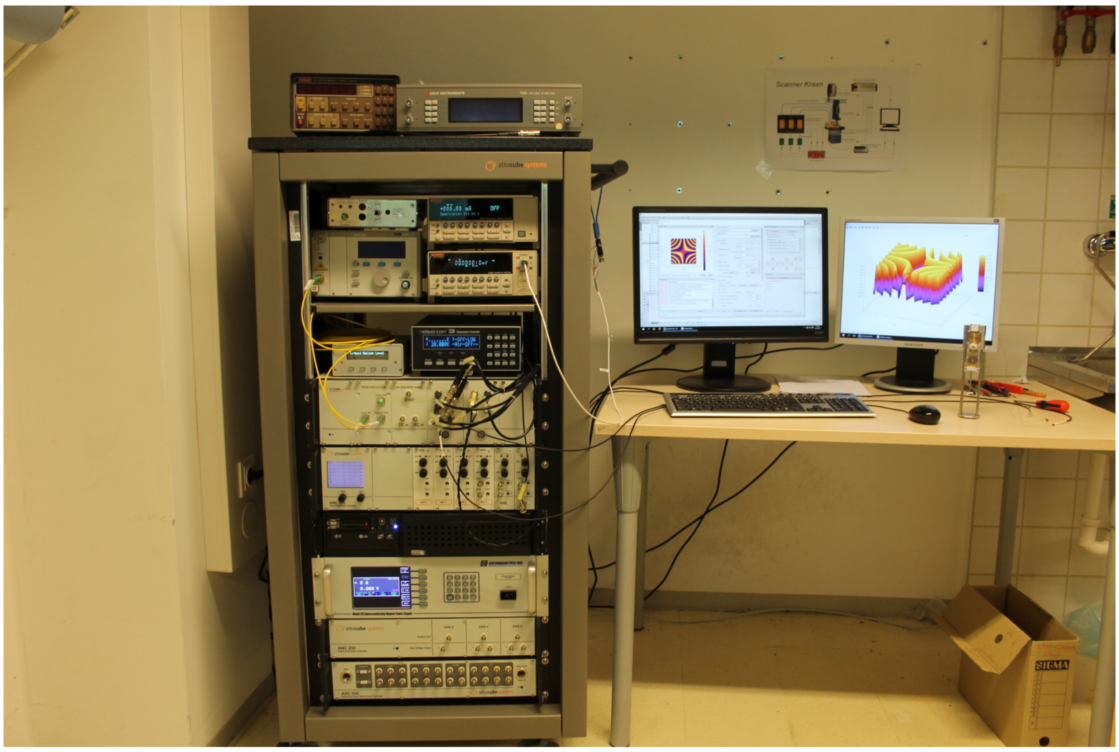


Figure 2.2: Photo of the scanning system with the device-rack and the controlling PC showing a simulated sample scan. The first prototype of the positioning system can be seen standing in front of the right monitor.

2.1 The Positioning System

The positioning system for the Hall scanner is depicted in figure 2.3 and consists of six piezo actuators, two for each axis, to enable coarse and fine positioning. The coarse x- and y-axes consist of rotational piezo actuators, which use gear systems to move a cantilever that holds the Hall probe. The gear of the x-axis rotator moves a toothed rack that is connected to the cantilever, while the y-axis rotator moves a threaded rod that in turn moves the cantilever system together with the x-axis rotator. Because of the threaded rod and an additional gear transmission, the y-axis is significantly slower than the x-axis. They cover a range of about $17 \times 27\,\mathrm{mm}^2$.

For the coarse z-axis, as well as for all three fine axes, linear piezo actuators

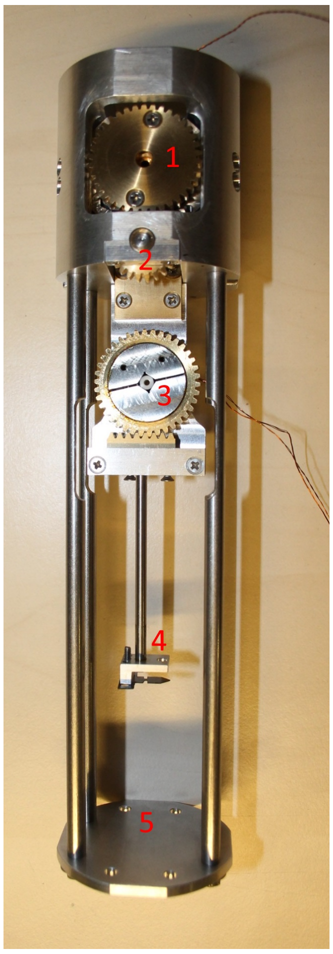


Figure 2.3: Prototype of the positioning system of the nano Hall scanner. The big gears (1) and (3) are driven by piezo rotators. The gear at (1) drives a smaller gear (2), which turns a threaded rod to move the cantilever (4) together with the other gear system (3) in y-direction (backward and forward through the picture plane). The gear at (3) moves the cantilever in x-direction via a toothed rack. The stack consisting of the coarse z-positioner and all three fine positioners will be placed at position (5) beneath the cantilever.

are used as a stack to move the sample. The coarse z-axis has a range of 5 mm, while the fine axes can cover $50 \times 50 \times 24\,\mu\text{m}^3$ at room temperature (300 K) and $30\times30\times15\,\mu\text{m}^3$ at cryogenic temperature (4 K). While the expansion of piezos, and thus the scan range of the fine positioners depends on temperature, the scan range of the coarse positioners does not, because they operate using the stick-slip-effect as explained in section 1.2. Therefore the scan range is limited by the geometry of the positioning system, not the expansion of the piezos. The movement speed of the coarse axes, however, is also dependent on the temperature because the step size of the actuators is considerably smaller at low temperatures.

2.2 Devices

Each device is connected to the PC that controls them by means of a Python software (see sections 3 and 4). Most devices are connected with either an RS-232 cable or with a USB cable simulating an RS-232 port, while for other devices a GPIB connection is necessary.

2.2.1 Attocube ANC300 Piezo Positioning Electronic

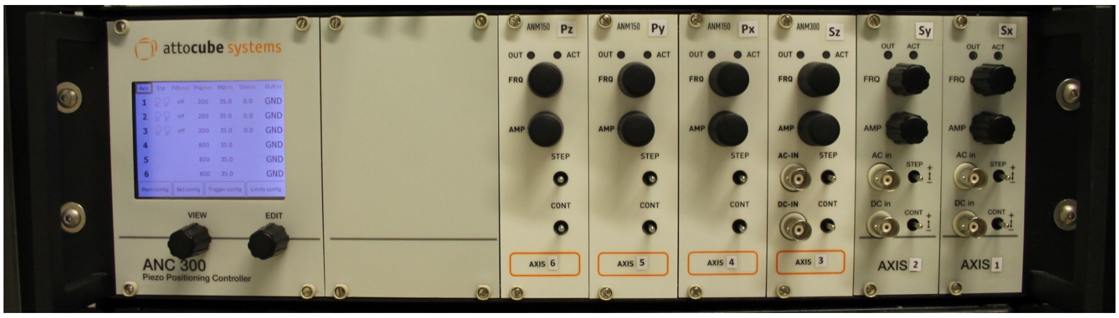


Figure 2.4: ANC300 with six modules, three ANM150 and three ANM300. From left to right, these are the coarse z-, y- and x-axes (labeled with "Pz", "Py" and "Px" for "positioning") and then the fine z-, y- and x-axes (labeled with "Sz", "Sy" and "Sx" for "scanning").

The Attocube ANC300 Piezo Positioning Electronic and its modules, shown in figure 2.4, are used to generate the electric signals to control the piezo actuators. It is connected to the PC via USB cable.

Modules The ANC300 uses a modular architecture, where up to six independent modules can be inserted into the controller. Each module is connected to one piezo actuator and sends the desired output signals to move this piezo. For this setup, all six modules are necessary, because for each of the three axes, x, y and z, there is a coarse and a fine positioning piezo used.

Two different modules are used for coarse and fine positioning, i.e. the ANM150 and the ANM300, respectively, which can be seen in figure 2.4. The main difference between the different modules is their capability of generating output signals for stepping and/or scanning with the corresponding piezo actuator. These modules, together with their modes will be explained in detail in the next two sections.

ANM150 stepping module This module supports stepping mode only. It is used for the three coarse positioning axes. Stepping mode means that the controller produces a saw tooth-like output, which, together with an appropriate piezo actuator, utilizes the stick and slip effect to move the actuator in distinct steps. The step size and frequency can be chosen and changed as needed, within certain power limits explained later in this section. While stepping mode enables fast movements that can cover the whole scan range, there are considerable vibrations because of the stepping character of the motion. Additionally, the positioning is not as accurate as with scanning mode, because the actual step sizes can vary and some steps can be skipped entirely, due to the design of the actuators. The step size also depends on the temperature of the piezo. To remedy all this, the fine positioners, which operate with scanning mode, are used whenever possible. Furthermore the interferometer (section 2.2.7) is used to measure and correct the position of the cantilever.

ANM300 combined stepping and scanning module This module supports stepping as well as scanning. It is used for the three fine positioning axes. Only its scanning capabilities are utilized, as the stepping is already performed by the coarse positioners, controlled by the ANM150 modules.

Scanning, in contrast to stepping, refers to a mode, where the controller is set to maintain a given voltage output, in order to expand or compress the piezo to the degree needed and keep it there. A specific voltage change ΔV corresponds to a specific movement Δd of the piezo axis, depending on the temperature as explained in section 1.2. Scanning mode is much more accurate than stepping, but at the same time covers a vastly smaller range. This is the reason for combining coarse and fine axes, in order to get a large scan range while also maintaining high accuracy.

Axis Modes Each module of the ANC300 can be in one of the following different modes, that determine the current capability of the axis.

Ground (`gnd`): This is the default mode. It disables all outputs and connects them to chassis mass.

Input (`inp`):	In this mode, AC-IN and DC-IN can be enabled, stepping and offset are disabled. This mode is only available for the ANM300 module.
Capacitance (`cap`):	Setting to this mode starts a capacitance measurement. When the measurement is finished, the axis returns to ground mode automatically. It is not necessary to switch to ground mode beforehand.
Step (`stp`):	In this mode, stepping is enabled. AC-IN and DC-IN functions are not modified, while an offset function would be turned off. This mode is available for both, the ANM150 and ANM300 modules.
Offset (`off`):	This mode outputs a given offset voltage. AC-IN and DC-IN functions are not modified, while any stepping would be turned off. This mode is only available for the ANM300 module.
Step+ (`stp+`):	This is additive offset and stepping mode. Stepping waveforms are added to an offset. AC-IN and DC-IN functions are not modified. This mode is only available for the ANM300 module.
Step- (`stp-`):	This is subtractive offset and stepping mode. Stepping waveforms are subtracted from an offset. AC-IN and DC-IN functions are not modified. This mode is only available for the ANM300 module.

Power Limitation The maximum stepping output power is limited to prevent damaging of the piezo actuators. The output power depends on the capacitance of the piezos, therefore the limit depends on the measured capacitance. In order to enforce this limit, the ANC300 will only allow certain combinations of stepping frequency and voltage (see table 2.1). For this reason, the capacitance needs to be measured before scanning or after large temperature changes that influence the capacitance of the piezos. For the limitation, the last measured capacitance value is used, or if no value was measured (if the ANC300 display shows “?” for the capacitance), the most restrictive limit will be applied.

Max. Frequency [Hz]		Capacitance [µF]								
		1	2	2.5	3	3.5	4	4.5	5	5.5
Voltage [V]	10	10000	3815	3624	3433	3338	3242	3147	2400	1329
	20	10000	3052	2899	2747	2670	2594	2518	1800	1208
	30	5500	2441	2319	2197	2136	2075	2014	1600	1098
	40	3100	1953	1855	1758	1709	1660	1611	1300	998
	50	2000	1563	1484	1406	1367	1328	1289	1100	908
	60	1300	1250	1188	1125	1094	1063	1031	900	825
	70	1000	1000	950	900	875	850	825	800	750

Table 2.1: ANC300 power limitation, taken from [5].
This table shows the maximal usable stepping frequency in Hz for a given stepping voltage (left column) and a given measured capacitance (top row). If the capacitance is not known, the most restrictive (lowest) value will be used as limitation. The typical capacitance for the piezos used in our setup ranges from about 1.1 µF to 2.1 µF at 300 K and from about 0.15 µF to 0.4 µF at 4 K.

2.2.2 Keithley 2182A Nanovoltmeter

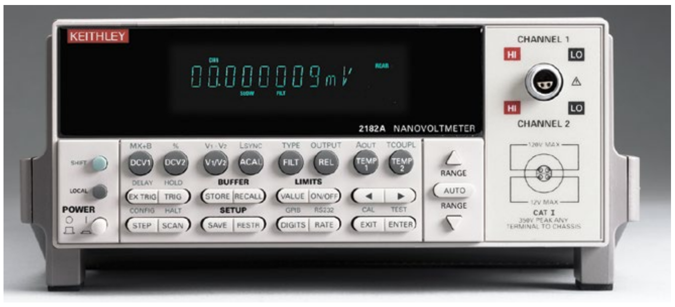


Figure 2.5: Keithley 2182A Nanovoltmeter, taken from [11]

The nanovoltmeter, depicted in figure 2.5, is used to measure the Hall voltage from the Hall probe. For scanning, it is not directly connected to the PC. Instead it is controlled by the Precision Current Source, which allows faster and more accurate measurements (see section 2.2.3). For this purpose, the nanovoltmeter needs to be connected to the current source via RS-232 cable and via trigger link cable, while the current source is connected to the PC via GPIB.

The nanovoltmeter can also be used without the Precision Current Source, when it is connected directly to the PC via RS-232 or GPIB. This mode of operation can be used to measure a voltage signal over time. This method was chosen to test the optical setup of the interferometer by connecting the nanovoltmeter to a photo diode and plotting the output, in order to see interference patterns (see section 2.3). This mode is also possible when leaving the nanovoltmeter connected to the current source, which is then only used as a relay to put commands sent from the PC through to the nanovoltmeter. This eliminates the need to reconnect the cables, but slows down the communication speed and thus limits the measuring rate.

2.2.3 Keithley 6220 Precision Current Source

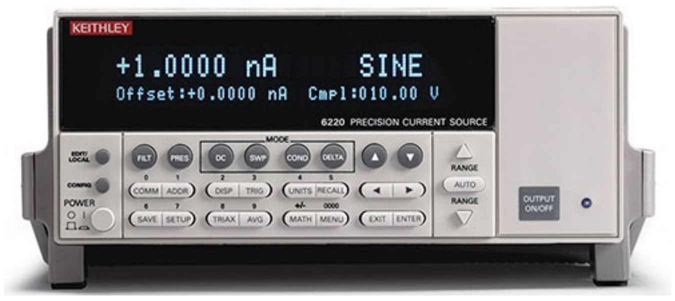


Figure 2.6: Keithley 6220 Precision Current Source, taken from [13]

The current source, depicted in figure 2.6, is used to supply the Hall probe with a precise current, and to control the nanovoltmeter for measuring the output. It can be connected to the PC via RS-232 or GPIB. To control the nanovoltmeter, both devices have to be connected together with their RS-232 and trigger link cables. This leaves only the GPIB port to connect it to the PC, which is realized through an Agilent GPIB to USB converter, such that a USB port of the PC can be used.

The measurement is performed using delta voltage measurements. This uses a current reversal technique to cancel the effects of the thermal electromotive forces (thermal EMFs), created by the connecting leads. The simplest version to realize this is for the supply voltage of the Hall probe to be applied normally and then with reversed polarity, and measuring the output both times. This eliminates the thermal EMFs in the end result

$$V_{delta} = \frac{V_1 - V_2}{2}, \tag{2.1}$$

where $V_1 = V_0 + V_{EMF}$ and $V_2 = -V_0 + V_{EMF}$ are the output voltages from the positive and the negative supply voltage, respectively, V_0 is the "real" voltage and V_{EMF} is the voltage due to the thermal EMFs. Since the polarity of V_0 is reversed between the measurements, while V_{EMF} stays the same, the latter is eliminated in the end result.

The current source actually uses an improved delta process with three single measurements for each delta measurement and a moving average algorithm. An alternating supply voltage is used as shown in figure 2.7, and the delta voltage is calculated with

$$V_{delta} = \frac{V_1 - 2V_2 + V_3}{4}(-1)^n, \tag{2.2}$$

where V_1, V_2 and V_3 are the three single measurements and n is the cycle number starting with 0 for a positive supply.

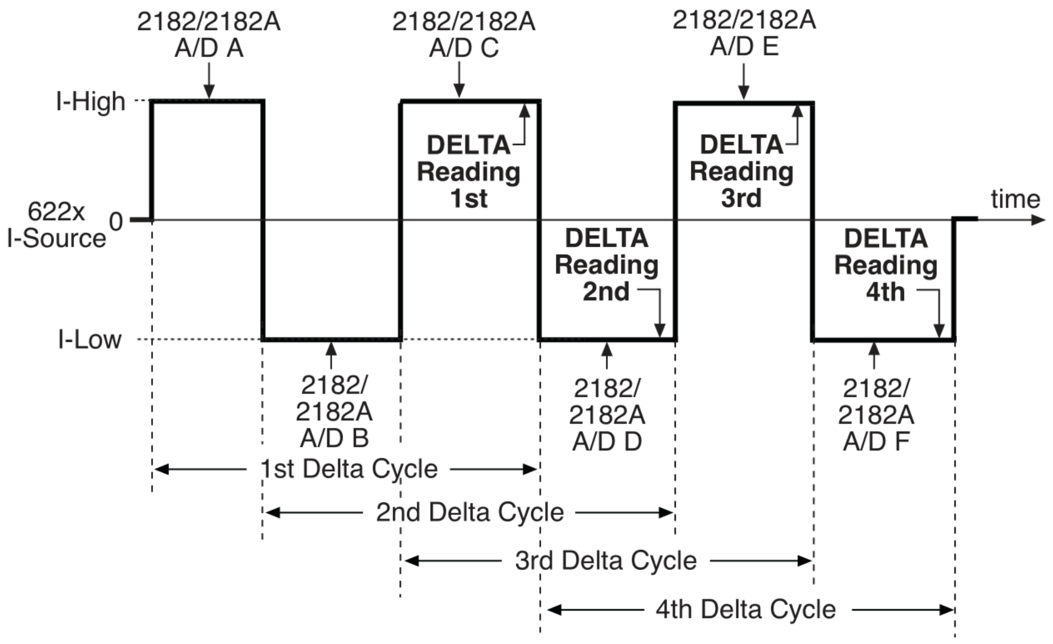


Figure 2.7: Three point delta measurement cycles, taken from [13]
This graphic shows the variation of the output current of the current source over time, as well as which peaks are used for each delta measurement cycle.

2.2.4 Cryocon Model 32B Temperature Controller

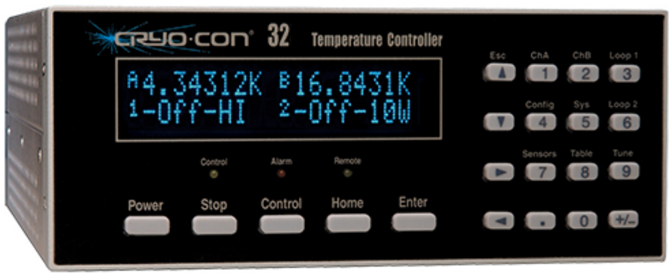


Figure 2.8: Cryocon Model 32B Temperature Controller, taken from [7]

The temperature controller, depicted in figure 2.8, is used to measure and control the temperature of a sample inside the cryostat. It is connected to the PC via RS-232 cable. The device uses a temperature sensor as feedback for a PID controller (see section 1.3), where the P, I and D values can be set remotely. This PID controller controls the power output connected to a heater. It can be set to three different power ranges, high, medium and low, which limit the maximum output power. This range has to be set such that the maximum output power will not damage the equipment. The output power for these ranges, which can be seen in table 2.2, depends on the load resistance value of either 25 Ω or 50 Ω that has to be set on the device to match the connected heater.

Range	Compliance Voltage		Full-Scale Current	Max. Output Power	
	25Ω	50Ω		25Ω	50Ω
High	25	50	1.0A	25 Watts	50 Watts
Medium	25	50	0.333A	2.5 Watts	5.0 Watts
Low	25	50	0.100A	0.25 Watts	0.50 Watts

Table 2.2: Heater output ranges for the Cryocon Model 32B Temperature Controller, taken from [7]. Compliance voltage, full-scale current and maximum output power are shown for load resistance values of either 25 Ω or 50 Ω for the three heater output ranges high, medium and low.

2.2.5 Cryomagnetics LM-500 Liquid Helium Gauge

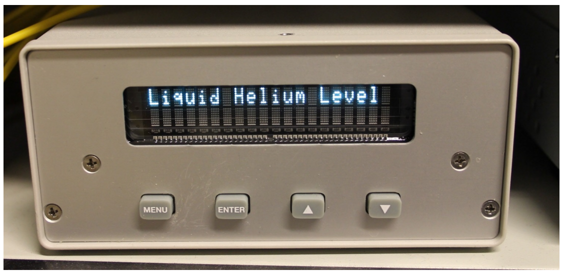


Figure 2.9: Cryomagnetics LM-500 Liquid Helium Gauge

The helium gauge, depicted in figure 2.9, measures the helium level inside the cryostat. If a critical He level is reached, the helium gauge can warn the user by sounding an alarm. It is connected to the PC via RS-232 cable.

2.2.6 Cryomagnetics 4G Magnet Power Supply

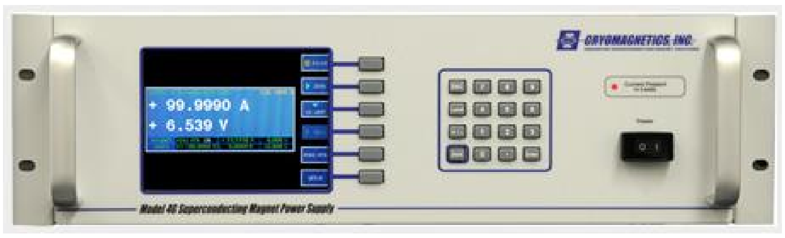


Figure 2.10: Cryomagnetics 4G Magnet Power Supply, taken from [9]

The magnet power supply, depicted in figure 2.10, controls and supplies the superconducting magnet located in the cryostat. It is connected to the PC via USB cable.

Superconducting magnets have the great advantage that they do not need to be supplied with power, once a current has been placed in them. The magnet just has to be short-circuited and it will retain its current with no need for external power. To be able to control the magnet, this short-circuiting is performed by using a superconducting wire, the so-called persistent switch, to connect the magnet's input terminals. This persistent switch can be heated above its transition temperature with a small heater, the persistent switch heater, which effectively severs the connection and allows the power supply to control the magnet's current. Once the heater is turned off, the persistent switch will cool down and become superconducting once again and the magnet will retain its present current. Then the power supply can be switched off and the field will remain. To deactivate the magnet, the power supply has to be brought up to match the current of the magnet and then, after enabling the persistent switch heater, it can sweep the magnet down to zero or to another field.

2.2.7 Interferometer

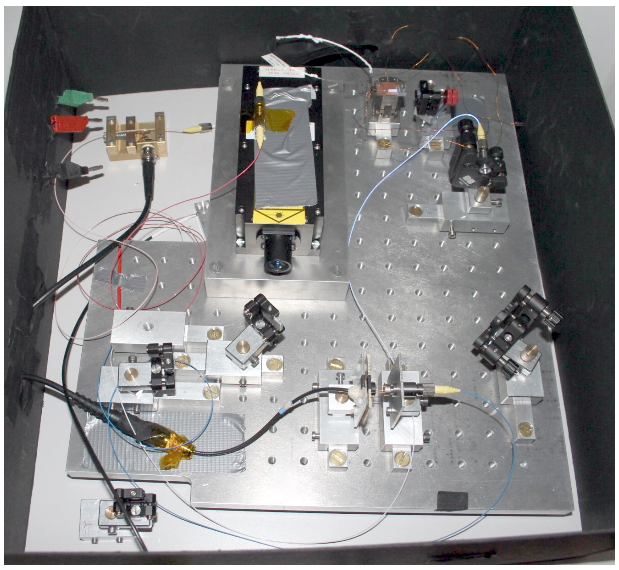


Figure 2.11: Photo of the prototype of the interferometry system. On the top left the fiber stretcher can be seen, the laser is in the middle next to it, and on the bottom in the middle there is the photo detector to measure the laser intensity.

The interferometer, of which a prototype setup is depicted in figure 2.11, is used to measure the position of the Hall probe over the sample, i.e. to accurately control the movement along the axes. It is developed and built in house and connected to the PC via USB. A laser with a wavelength of 473 nm is used and split into three beams to measure all three axes simultaneously.

The interferometers of the x- and y-axes measure the relative position of the Hall probe over the sample, to facilitate accurate positioning, while the z-axis interferometer measures the bending of the cantilever. That way it is possible in the ideal case to determine, when the cantilever is close enough to the sample to bend down due to atomic forces. This enables a very accurate z-positioning above the sample without needing to know its absolute position or geometry very accurately in advance.

This feedback is used in the following way: For each movement of any axis, the software calculates the needed number of steps (for the coarse positioners) or the voltage output (for the fine positioners) from a saved distance-per-voltage value and initiates the movement. When the movement is finished, the actually reached position as reported by the interferometer is compared to the desired position. If the error is small enough, the movement is finished, otherwise this process is repeated. During this process, the distance-per-voltage value is updated to more accurately represent the actual value that is also temperature dependent.

2.3 Measuring Interference

The nanovoltmeter described in section 2.2.2 was also used to measure interference patterns to help setting up the laser and the mirrors for the interferometer. To do this, a variable gain current amplifier was connected to a photo diode that measured the intensity of the laser as the output of a Michelson interferometer. The output of this amplifier was then connected to the nanovoltmeter in order to measure the signal over time.

For first adjustments this was used to measure the intensity of the laser reaching the photo diode, while aligning the mirrors, in order to adjust the laser correctly. After this was completed, interference patterns were created by placing a mirror on a piezo in one arm of the Michelson interferometer. The piezo was then supplied with a variable voltage pattern, either a sine wave or a square wave, in order to periodically expand and contract the piezo. For later experiments, in order to minimize the need to accurately adjust many mirrors, the interferometer was realized by using a fiber optic splitter instead of mirrors, and the piezo mirror was replaced by a piezo fiber stretcher seen in figure 2.16. The fiber stretcher works by slightly changing the length of the prestressed fiber of one arm of the Michelson interferometer and therefore the length of this arm. This leaves only the two mirrors to adjust that couple the laser into the fiber.

For these testing purposes the piezo mirror and the fiber stretcher, respectively, were used to produce measurable interference patterns, because there is not yet a finished positioning system to use for real measurements. For measuring the position of the cantilever in the final system, the fiber stretcher will be used to superimpose a sine wave onto the position signal in order to employ a lock-in technique to filter out the background noise. Here the frequency of the superimposed wave will be in the range of some kHz in contrast to the measurements described above, where some Hz were used, because the nanovoltmeter is not fast enough to measure at higher frequencies. This will not be a problem in the finished system, because the measurements will be performed by a dedicated micro-controller.

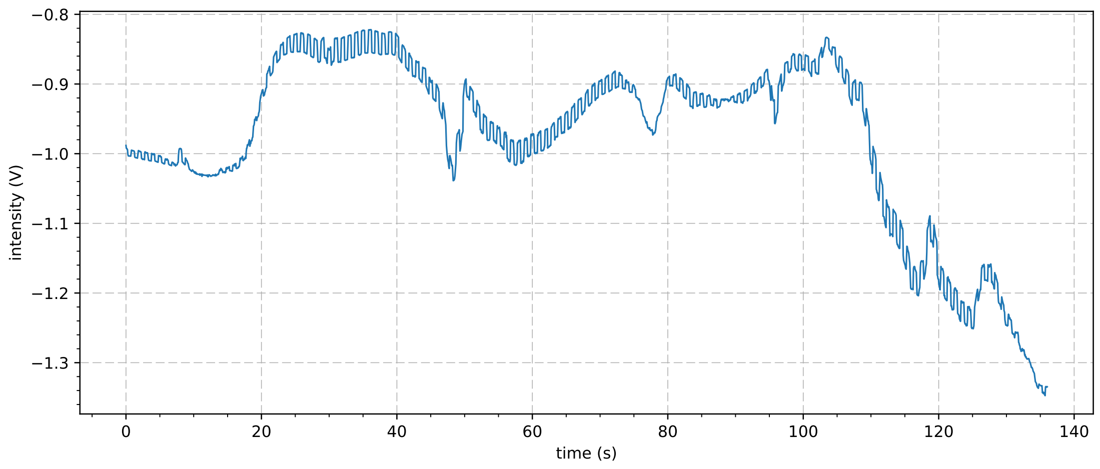


Figure 2.12: Interference pattern, where the piezo was supplied with a square wave with a frequency of 1 Hz. All other fluctuations except this square wave are caused by background noise.

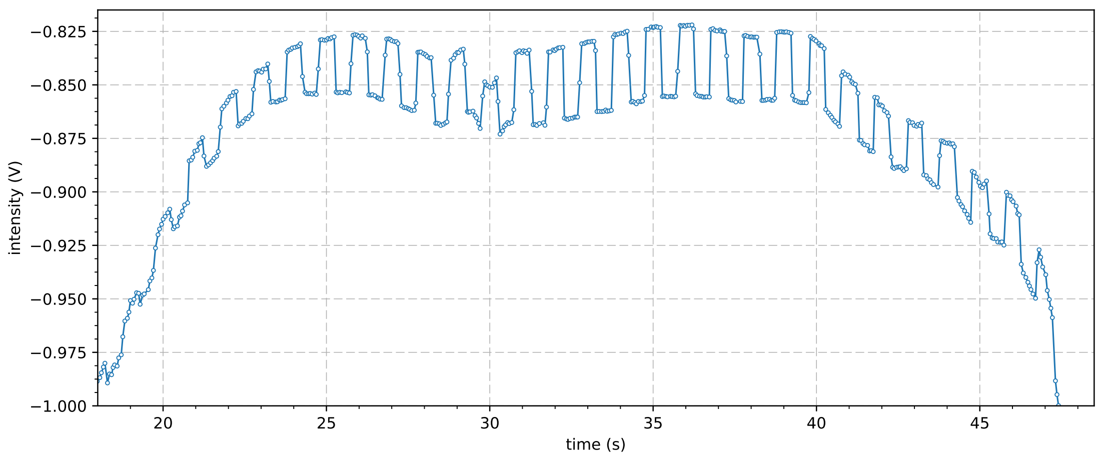


Figure 2.13: Zoomed in section of the interference pattern from figure 2.12 with a piezo-induced square wave with a frequency of 1 Hz.

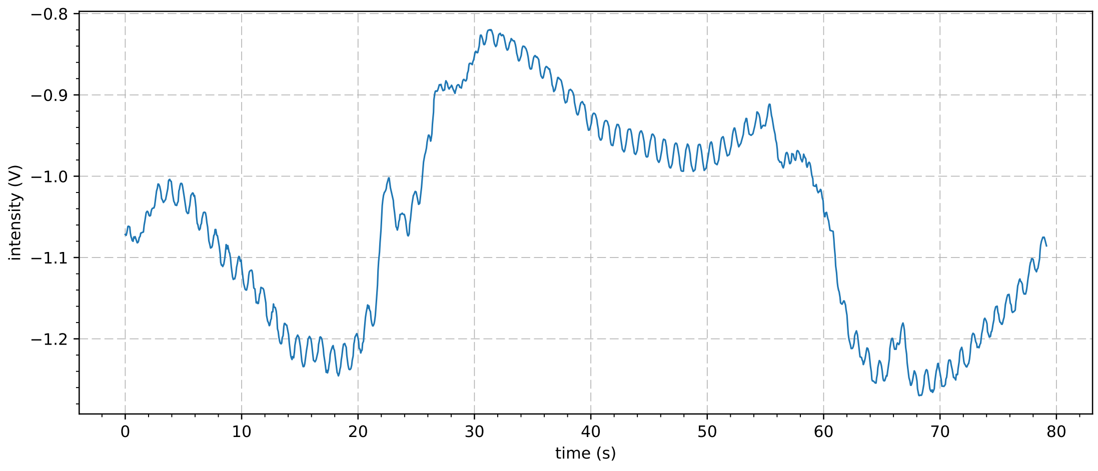


Figure 2.14: Interference pattern, where the piezo was supplied with a sine wave with a frequency of 1 Hz.

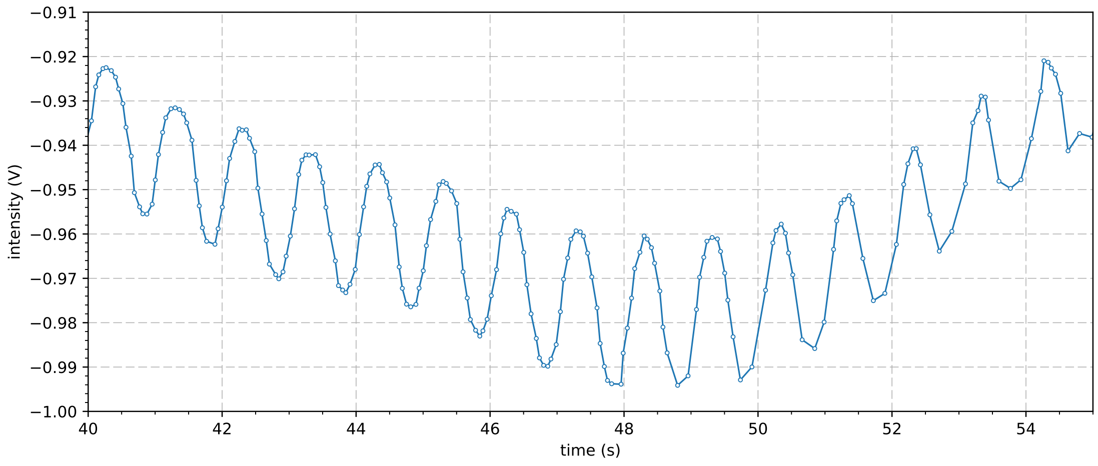


Figure 2.15: Zoomed in section of the interference pattern from figure 2.14 with a piezo-induced sine wave with a frequency of 1 Hz.

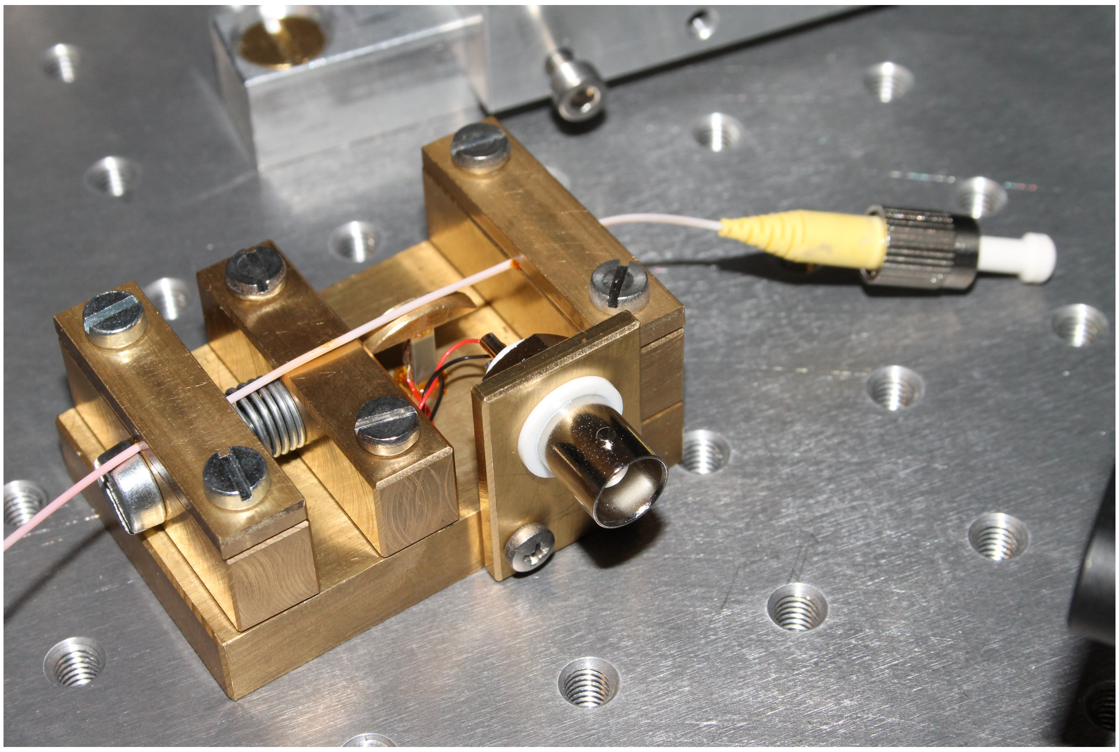


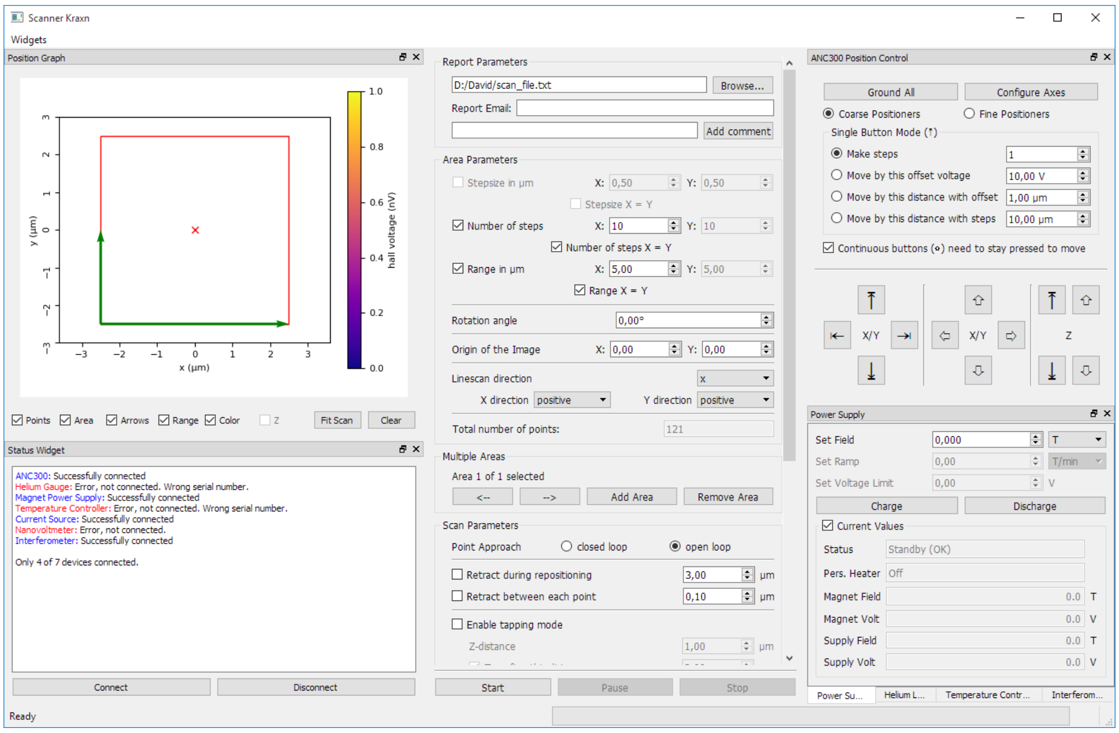
Figure 2.16: Fiber stretcher for the interferometer. The fiber is fixed on both ends with Varnish and the screws from the top and can then be pre-stretched using the screw on the left. The piezo to modulate the length of the fiber is located in the center, behind the BNC connector.

Figure 2.12 shows a measurement where the piezo was supplied by a square wave with a frequency of 1 Hz. Figure 2.13 shows the same data, but zoomed into a section of it, in order to better show the clearly visible square wave. Figure 2.14 shows another plot, where the piezo was supplied with a sinus wave, also with a frequency of 1 Hz, and figure 2.15 shows a magnified section of this plot. These plots show the very strong background noise, but this is mostly due to the experimental setup of the interferometer and vibrations caused by people working on it. The background noise is expected to be much less pronounced in the finished interferometer. Both plots were created before updating the setup to use the fiber optic splitter and the fiber stretcher.

3 The User Interface

The functionality of the Python software is accessible through the user interface or by writing a Python script. The user interface and its usage is discussed in detail in this section, for the scripting capabilities see section 4.3. The main window can either be started by using the desktop link “Scanner Kraxn Main Window”, or by directly running the “main_window” module in the program’s root folder (see section 4).

3.1 The Main Window



ANC300: Successfully connected
Helium Gauge: Error, not connected. Wrong serial number.
Magnet Power Supply: Successfully connected
Temperature Controller: Error, not connected. Wrong serial number.
Current Source: Successfully connected
Nanovoltmeter: Error, not connected.
Interferometer: Successfully connected

Only 4 of 7 devices connected.

D:/David/scan_file.txt
Report Email:

☐ Stepsize in µm X: 0,50 Y: 0,50
☐ Stepsize X = Y
☑ Number of steps X: 10 Y: 10
☑ Number of steps X = Y
☑ Range in µm X: 5,00 Y: 5,00
☑ Range X = Y
Rotation angle 0,00°
Origin of the Image X: 0,00 Y: 0,00
Linescan direction x
X direction positive Y direction positive
Total number of points: 121

Area 1 of 1 selected

Point Approach ○ closed loop ● open loop
☐ Retract during repositioning 3,00 µm
☐ Retract between each point 0,10 µm
☐ Enable tapping mode
Z-distance 1,00 µm

● Coarse Positioners ○ Fine Positioners
Single Button Mode (↑)
● Make steps 1
○ Move by this offset voltage 10,00 V
○ Move by this distance with offset 1,00 µm
○ Move by this distance with steps 10,00 µm
☑ Continuous buttons (⇧) need to stay pressed to move

Set Field 0,000 T
Set Ramp 0,00 T/min
Set Voltage Limit 0,00 V
☑ Current Values
Status Standby (OK)
Pers. Heater Off
Magnet Field 0.0 T
Magnet Volt 0.0 V
Supply Field 0.0 T
Supply Volt 0.0 V

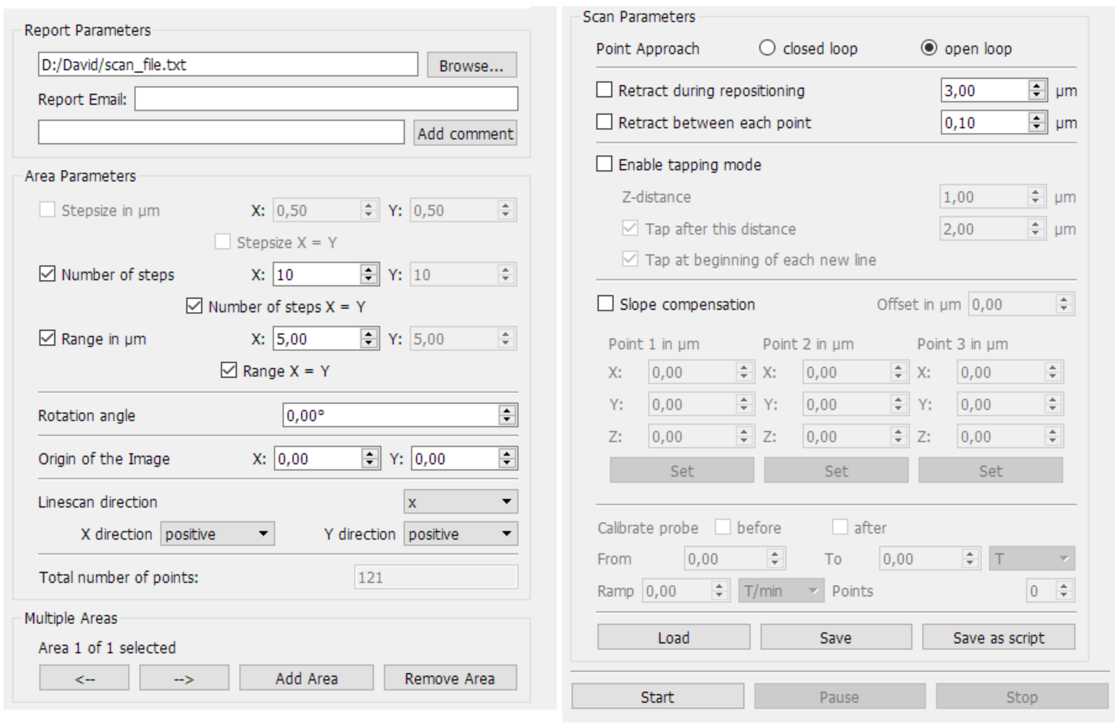
Figure 3.1: The main window of the user interface. On the left side there are the graph widget and the status widget, the scan parameter widget is in the middle and on the right side there are the position control widget and the power supply widget. Under this widget is a row of tabs to switch to the other device widgets.

Figure 3.1 shows the main window of the scanning software, that is used to setup and monitor a scan and for controlling the devices of the scanning system. If the main window is closed, all connected devices will be disconnected, after performing some device specific cleanup actions. For example, all ANC axes are automatically grounded before the connection to the ANC is closed. When the user tries to close the main window during a scan, a warning message pops up, which prevents an accidental abortion of the scan by forcing the user to confirm the closing of the application.

To aid the user, every text field, button or drop-down-box has an informative tool tip that briefly describes the function of this element. In addition to that, a help mode can be entered by pressing the F1 key on the keyboard. In this mode, clicking on any one of theses elements shows a more detailed description of its function.

For debugging purposes, info and error messages are saved in a log file continuously while the application is running. These log files are consolidated for each day. When a scan is started, a log file specific for this scan is created. Logging is described in more detail in section 4.2.2.

3.2 The Scan Parameter Widget


Report Parameters
D:/David/scan_file.txt — Browse...
Report Email:
Add comment

Area Parameters
Stepsize in µm X: 0,50 Y: 0,50
Stepsize X = Y
[✓ Number of steps] X: 10 Y: 10
[✓ Number of steps X = Y]
[✓ Range in µm] X: 5,00 Y: 5,00
[✓ Range X = Y]
Rotation angle: 0,00°
Origin of the Image X: 0,00 Y: 0,00
Linescan direction: x
X direction: positive Y direction: positive
Total number of points: 121

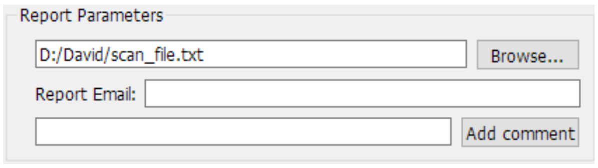
Multiple Areas
Area 1 of 1 selected
<-- --> Add Area Remove Area

Scan Parameters
Point Approach: closed loop [✓ open loop]
Retract during repositioning 3,00 µm
Retract between each point 0,10 µm
Enable tapping mode
Z-distance 1,00 µm
[✓ Tap after this distance] 2,00 µm
[✓ Tap at beginning of each new line]
Slope compensation Offset in µm 0,00
Point 1 in µm X: 0,00 Y: 0,00 Z: 0,00 Set
Point 2 in µm X: 0,00 Y: 0,00 Z: 0,00 Set
Point 3 in µm X: 0,00 Y: 0,00 Z: 0,00 Set
Calibrate probe before after
From 0,00 To 0,00 T
Ramp 0,00 T/min Points 0
Load Save Save as script
Start Pause Stop

Figure 3.2: The scan parameter widget (split in half). This widget is used to configure the settings for the scan. Available options are "'Report Parameters", "Area Parameters", settings for "Multiple Areas" and "Scan Parameters".

This is the main part of the user interface and controls all parameters of the scan. It is located at the center of the main window. The following sections will describe each part of the scan widget and its functions. Each section contains a screenshot of the relevant part, while figure 3.2 shows the full scan widget. While the scan settings can be changed during a scan, this has no effect until the current scan is finished or aborted and a new scan is started.

3.2.1 Report Parameters



Report Parameters

D:/David/scan_file.txt Browse...

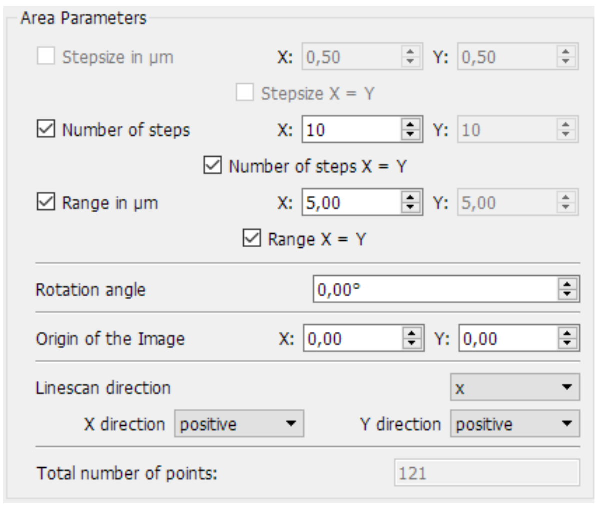
Report Email:

Add comment

Figure 3.3: Settings for the report parameters for the scan, where the path of the scan file, email addresses to report to and additional comments for the log file can be set.

The report parameters section, seen in figure 3.3, controls the saving and reporting of the gathered data. The first field is mandatory and sets the location of the scan file. Pressing the “Browse...” button opens an explorer dialog where the file can be selected. “Report Email” contains a comma separated list of email addresses. If this field is not empty, an email is automatically sent to all addresses when an error occurs or the scan is finished. With the “Add Comment” button, the text of the adjacent text field is added to the log file of the scan, which is explained in section 4.2.2.

3.2.2 Area Parameters



Area Parameters

☐ Stepsize in µm X: 0,50 Y: 0,50

☐ Stepsize X = Y

☑ Number of steps X: 10 Y: 10

☑ Number of steps X = Y

☑ Range in µm X: 5,00 Y: 5,00

☑ Range X = Y

Rotation angle 0,00°

Origin of the Image X: 0,00 Y: 0,00

Linescan direction x

X direction positive Y direction positive

Total number of points: 121

Figure 3.4: Scan area parameters to define the size, orientation and point density of a scan area.

With this part of the scan widget, depicted in figure 3.4, the scan area as well as the point density of a rectangular scan can be set. It is possible to define multiple areas for one scan, see figure 3.5 and section 3.2.3 for more information on that. All distances are entered in µm.

The size and point density of the scan area are defined by selecting two out of the three available settings: “Stepsize”, “Number of steps”, and “Range”. Only two of these can be enabled at the same time, in order to enable an other pair, one checkbox must first be disabled. The disabled settings are automatically given values that result from the active settings. Each setting has a corresponding “X = Y” checkbox, that sets the y-value to be equal to the x-value. This checkbox has to be disabled, in order to set different values for the x- and y-direction.

“Stepsize” defines the size of each step, that is, the distance between two scan points. “Number of steps” determines the number of steps taken in each direction, which is one less than the number of scan points. The allowed values that can be entered start at 0, which means only one line of scan points in the respective direction. “Range” defines the total size of the scan area.

The “Rotation angle” field sets the clockwise rotation of the scan area in degrees, negative values are allowed. With “Origin of the Image”, the center point of the scan range, and with it, the relative position of different scan areas, as explained in section 3.2.3, can be adjusted. “Linescan direction” can be either “x” or “y” and determines the direction in which the Hall probe will move first to measure points. This axis is indicated in the graph widget by the longer arrow, as can be seen in figures 3.5 and 3.15. In case of an area scan, the line scan axis has to move back the whole scan range for each scanned line, so for this purpose the x-axis should be the line scan axis most of the time, because this axis is substantially faster than the y-axis. “X direction” and “Y direction” determine, if the axes are scanned in positive or negative direction. The direction is also indicated by the arrows in the graph widget (see figure 3.5). Lastly, the total number of points to be scanned is displayed at the bottom part of the area parameters section.

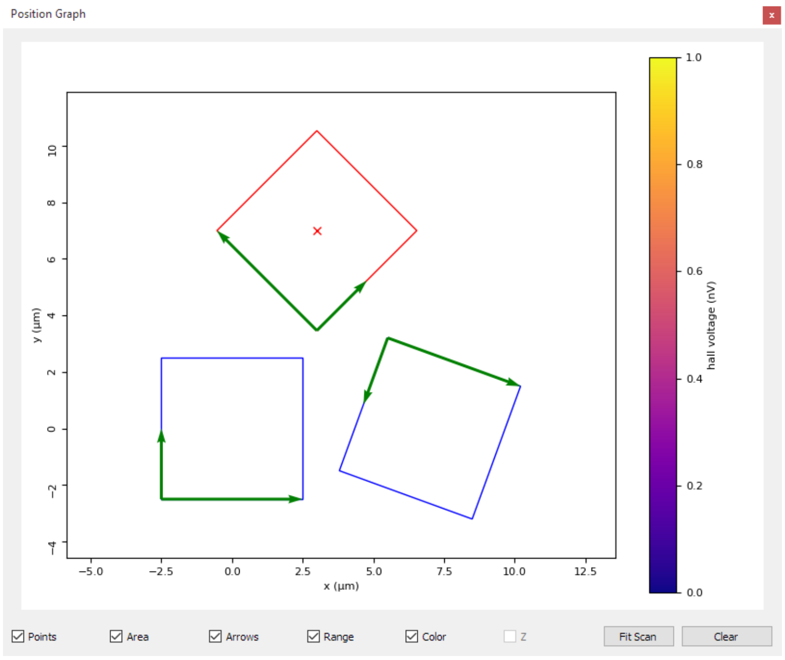


Figure 3.5: Graph widget with three scan areas and different scan directions. The currently active scan area is shown in red, while the others are blue. The green arrows extend from the start point of their scan area to show the scan direction, the longer arrow marks the linescan direction that is scanned first.

3.2.3 Multiple Areas

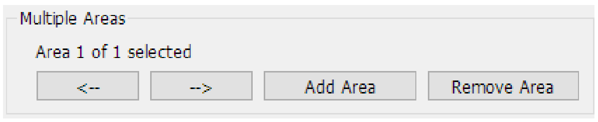
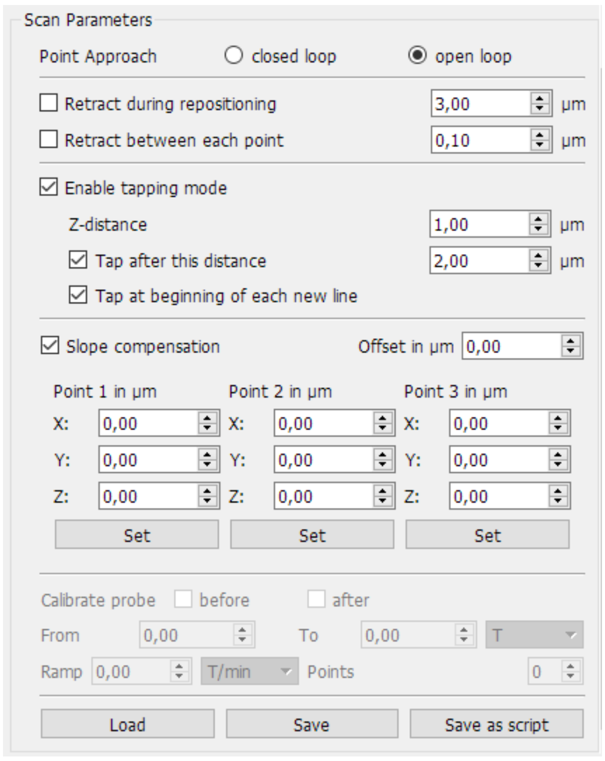


Figure 3.6: Configuration of multiple scan areas, the "<–" and "–>" buttons allow to cycle through the selected areas.

With the section shown in figure 3.6, it is possible to define multiple scan areas, with each having its own independent settings from the area parameters. To add a new area, click the "Add Area" button. A new scan area is then added with the same settings as the previous one, so they are overlayed until the settings are changed. The active area can be cycled with the "<–" and "–>" buttons and is indicated in the graph widget by a red box. All other areas are drawn in blue. Changes of the area parameters only affect the currently active scan area. An area can be removed with the “Remove Area” button; the last remaining area can not be removed.

3.2.4 Scan Parameters


Scan Parameters

Point Approach ○ closed loop [✓ open loop]

☐ Retract during repositioning 3,00 µm

☐ Retract between each point 0,10 µm

☑ Enable tapping mode

Z-distance 1,00 µm

☑ Tap after this distance 2,00 µm

☑ Tap at beginning of each new line

☑ Slope compensation Offset in µm 0,00

	Point 1 in µm	Point 2 in µm	Point 3 in µm
X:	0,00	0,00	0,00
Y:	0,00	0,00	0,00
Z:	0,00	0,00	0,00

Set Set Set

Calibrate probe ☐ before ☐ after

From 0,00 To 0,00 T

Ramp 0,00 T/min Points 0

Load Save Save as script

Figure 3.7: Scan Parameters with settings for point approach, retraction, tapping, slope compensation and saving, loading and exporting of files.

Figure 3.7 shows the scan parameters section. Here the parameters of the scan, that apply to all scan areas, can be set. Each part of these settings is described in detail in the following sections, accompanied by an image of the corresponding part of the widget.

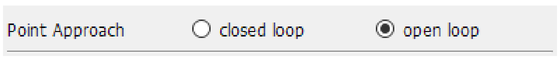
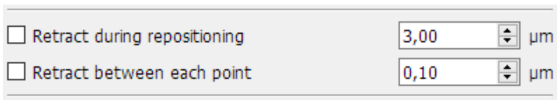

Point Approach ○ closed loop [✓ open loop]

Figure 3.8: Point approach settings to control if the interferometer should (closed loop) or should not (open loop) be used for the positioning of the Hall probe.

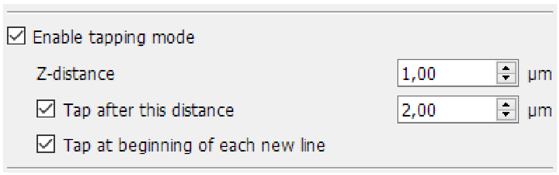
Point Approach The point approach can be set to open or closed loop mode with the settings shown in figure 3.8. Closed loop means, that the interferometer feedback is used to position the Hall probe, while open loop does not make use of the interferometer and determines the position only by the number of moved steps of the coarse axes and the applied offset voltages for the fine axes. This mode is faster, but can be very inaccurate, so for accurate scanning the user should always use closed loop mode.



☐ Retract during repositioning	3,00	µm
☐ Retract between each point	0,10	µm

Figure 3.9: Retraction settings with the corresponding retraction distances.

Retraction With the settings seen in figure 3.9, the Hall probe can be configured to retract at certain times to be farther away from the sample during long movements. "Retract during repositioning" means that the probe is retracted whenever a new line is started and the axis has to move back over the scan area. "Retract between each point" retracts the probe for every movement. The distances to retract can be set separately for these two modes with the corresponding spin boxes.



☑ Enable tapping mode

Z-distance	1,00	µm
☑ Tap after this distance	2,00	µm
☑ Tap at beginning of each new line		

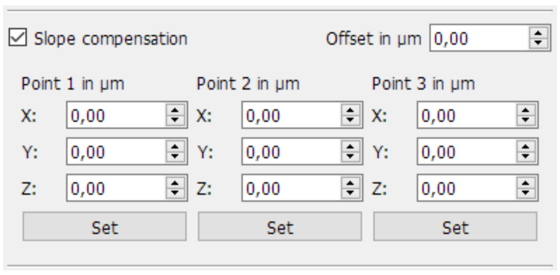
Figure 3.10: Tapping mode settings with settings for the Z-distance and the tapping distance.

Tapping Mode The scanning software supports tapping mode, which means that the probe is moved closer and closer towards the sample, until the z-axis of the interferometer detects the bending of the cantilever and thus contact with the sample. The interferometer is always used for this, regardless of the point approach setting discussed previously. The available settings are shown in figure 3.10. Tapping mode can be enabled or disabled with the "Enable tapping mode" checkbox.

"Z-distance" is the distance, the probe shall be moved away from the sample after contact, thus the scanning height.

If "Tap after this distance" is enabled, a tapping is performed whenever the distance between the last tapping point and the current scan point exceeds the value set at the corresponding spin box. If this value is set to 0 or any value smaller than or equal to the step size, a tapping is performed for every scan point.

"Tap at the beginning of each line" means exactly that: if enabled, a tapping is performed each time a line is finished, for the first point of the next line.



☑ Slope compensation — Offset in µm: 0,00

	Point 1 in µm	Point 2 in µm	Point 3 in µm
X:	0,00	0,00	0,00
Y:	0,00	0,00	0,00
Z:	0,00	0,00	0,00
	Set	Set	Set

Figure 3.11: Slope compensation settings to define the angled plane of the sample. The widget includes text fields to manually set points and "Set" buttons to set them to the current position of the cantilever.

Slope Compensation With slope compensation, for which the available settings are shown in figure 3.11, the error of a not planarly aligned sample can be corrected. It effectively changes the scanning plane to an angled one. To do this, three points on the angled plane have to be set. This can either be done manually for the x-, y- and z-coordinates of each point, or by using the "Set" buttons. When pressed, these buttons populate the coordinate fields above them with the current position of the probe. The "Offset in µm" field allows to set an offset for the angled plane to the set points in the z-direction.

Slope compensation also works together with tapping mode, by automatically readjusting the z-position of the angled plane each time a tapping is performed, but leaving the orientation of the plane unchanged.

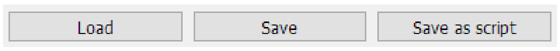


Figure 3.12: Buttons to save and load settings and to export a scan script

Save and Load Settings With these three buttons, shown in figure 3.12, all scan parameters can be saved, loaded or exported to a script. The “Save” button saves all settings to a “`*.ini`” file that can be restored again using the “Load” button. This can be used to save time when frequently using the same settings. By overwriting the “`default_settings.ini`” file in the “`scan_settings`” directory of the program with the “Save” function, the default startup settings of the scan widget can be changed, as this file is loaded when the application is started.

The “Save as script” button generates a script to run the scanning software with the current settings without the user interface as will be described in section 4.3.

3.2.5 Starting, Pausing, Resuming and Aborting the Scan

The scan process can be controlled with the three buttons at the bottom of the scan widget. When no scan is started, only the “Start” button is enabled. Clicking it starts the scan, after showing a summary message with all scan settings. This message allows the user to abort the scan, if any of these settings are not set as intended. After the scan is started, the “Pause” button is enabled to allow the user to pause the scan. After the scan is paused, it can be resumed with the same button, now captioned “Resume”, or aborted with the “Stop” button. Figure 3.13 shows the possible states of the buttons for each state of the scan.

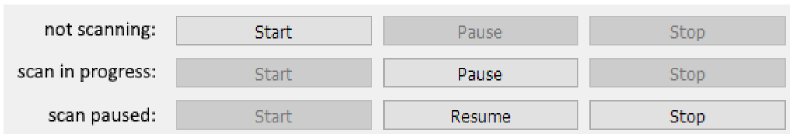


Figure 3.13: Possible states of the start, stop, and pause buttons. Each row shows the state of these buttons for a different status of the scan: not scanning, scan in progress and scan paused.

3.3 Graph, Status and Device Widgets

Each device has its own widget, from which this device can be controlled. Additionally, there is a status widget and a graph widget. Most of the widgets can have a single instance only, but the “ANC300 Axis Configuration” widget can be opened multiple times to be able to view the settings of multiple axes simultaneously. The widgets can be freely docked inside the main window, positioned somewhere on the screen as independent floating windows, or closed, if not needed. To re-enable a closed widget, the “Widgets” menu from the menu bar can be used. It is shown in figure 3.14, where a check mark indicates a currently displayed widget.

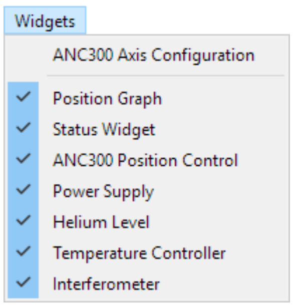


Figure 3.14: Widgets menu to enable and disable widgets. A check mark indicates that this widget is currently displayed. A widget can be displayed or hidden by clicking on its name in the list. The ANC300 Axis Configuration widget has no check mark because there can be multiple instances of this widget. Clicking on its entry opens another instance.

3.3.1 The Graph Widget

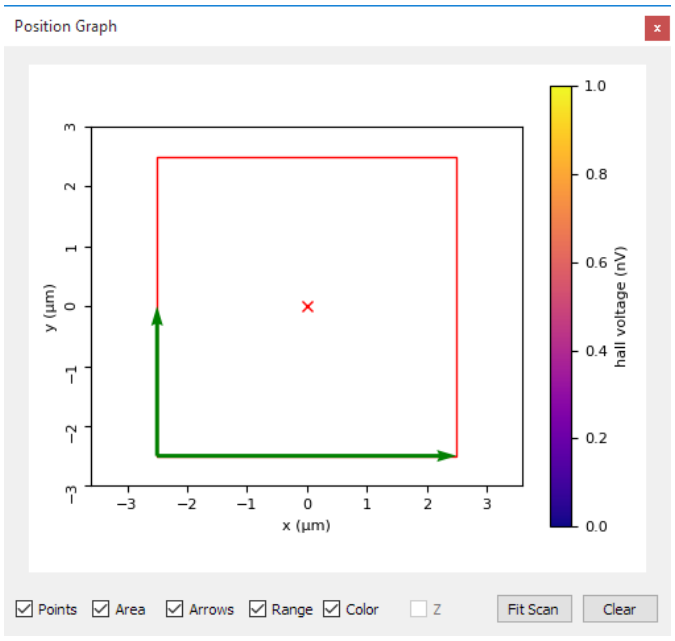


Figure 3.15: Graph widget with one scan area before scanning. The red rectangle shows the scan area, the green arrows indicate the direction of the scan, which is started at their origin. The longer arrow points out the linescan axis.

The graph widget, shown in figure 3.15, displays scan areas, scanned points and the total scanning range. During setup it can be used to see the position, size and aspect ratio of the scan areas, as well as the scan directions. The latter are represented by green arrows, where the longer arrow points out the linescan axis. The origin of both arrows is at the starting point of the scan in one of the corners of the scan range, depending on the set scan directions discussed in section 3.2.2. When using multiple scan areas, the currently active one is displayed in red, while all others are colored blue. During the scan, the graph widget can display all scanned points, optionally color coding their measured values.

The bottom row of the widget contains check boxes and buttons to change the parameters of the widget. The "Points" checkbox enables or disables the display of the scanned points. If it is disabled, only the most recent point is displayed. This checkbox should be disabled during scanning, because the continuous updating of the graph is CPU-intensive and possibly slows down the scan, if there are many points to plot.

The “Area” checkbox enables or disables the rectangles representing scan areas, while the “Arrows” checkbox is responsible for the display of the direction-arrows. The “Range” checkbox activates or deactivates the display of the whole scan range of the scanning system. If the “Color” checkbox is enabled, a color bar is displayed next to the graph and the scanned points are displayed with a color corresponding to the measured voltage. If it is disabled, all points have the same color and the color bar vanishes.

The “Fit Scan” button adjusts the visible range of the graph, such that all scan areas are displayed, the “Clear” button deletes all saved points. If it is pressed during a scan, all points up to this point in time are deleted, but new points are still displayed. Deletion only affects the graph widget, no point is ever deleted from the scan file. Figures 3.16 and 3.17 show the graph widget during and after scans with a different number of scan points, where the results are simulated with the equation $V = \sin(4 \cdot x \cdot y)$.

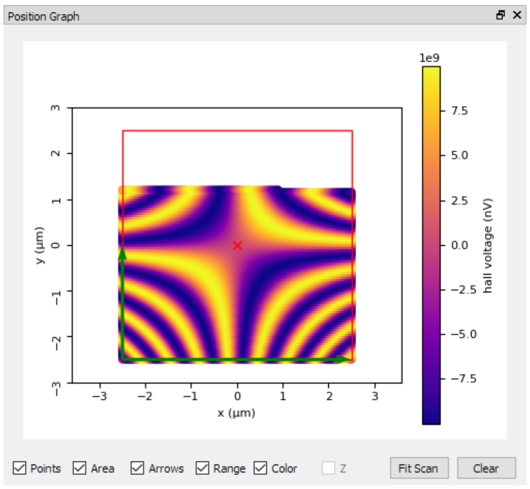


Figure 3.16: Graph widget during a simulated scan with 100×100 steps

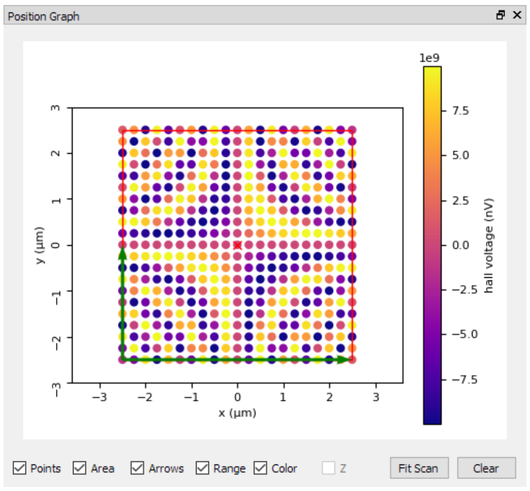
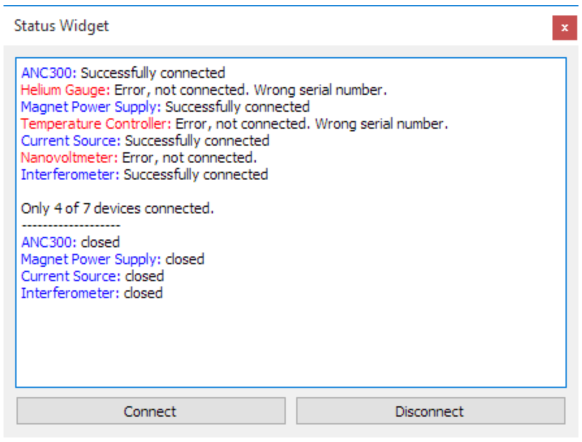


Figure 3.17: Graph widget after a simulated scan with 20×20 steps

3.3.2 The Status Widget



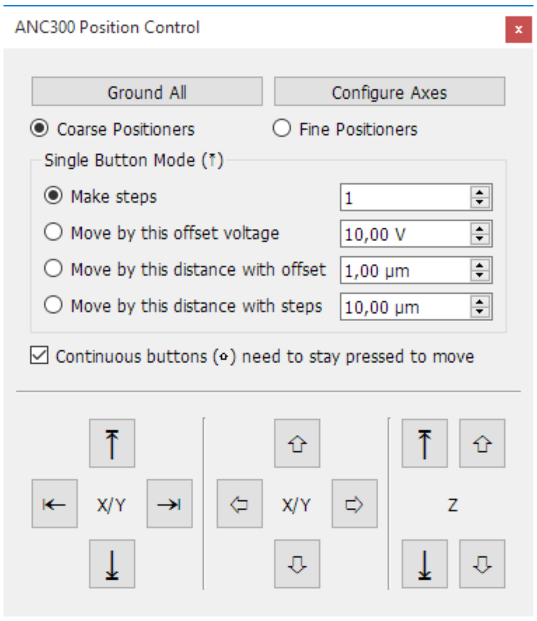
```
ANC300: Successfully connected
Helium Gauge: Error, not connected. Wrong serial number.
Magnet Power Supply: Successfully connected
Temperature Controller: Error, not connected. Wrong serial number.
Current Source: Successfully connected
Nanovoltmeter: Error, not connected.
Interferometer: Successfully connected

Only 4 of 7 devices connected.
-------------------
ANC300: closed
Magnet Power Supply: closed
Current Source: closed
Interferometer: closed
```

Figure 3.18: Status widget. Various status messages and errors are displayed in this widget.

The status widget, shown in figure 3.18, is used to display status and error messages. All devices can be connected and disconnected using the “Connect” and “Disconnect” buttons of the status widget. The connection status for each device is displayed in the text field. If the connection fails for one device, the cause of the error is printed next to the name of the device. All other error messages are also printed here. If the “Connect” button is pressed, while some devices are still connected, their connection is left unchanged and each unconnected device will try to connect.

3.3.3 The ANC300 Position Control Widget



[✓ Coarse Positioners] Fine Positioners

Single Button Mode (⤒)

[✓ Make steps]	1
Move by this offset voltage	10,00 V
Move by this distance with offset	1,00 µm
Move by this distance with steps	10,00 µm

[✓] Continuous buttons (⇧) need to stay pressed to move

Figure 3.19: Position control widget. This widget is used to manually move the piezo actuators by a set number of steps, a set distance, or simply as long as the button stays pressed.

This widget, shown in figure 3.19, is used to manually move the piezo actuators. If the ANC300 is not connected or gets disconnected, this widget will be automatically disabled until a connection is established again.

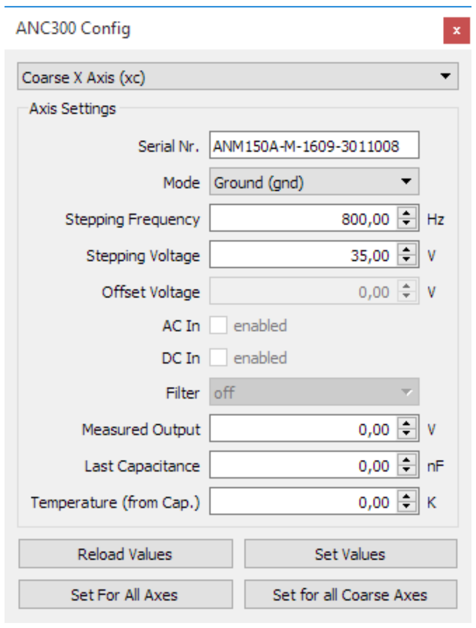
The "Ground All" button puts every axis into ground mode, thereby stopping all movement. The successful grounding of the axes will be confirmed by the status widget. A piezo actuator should not be disconnected, unless its axis is in ground mode. The "Configure Axes" button will open the axis settings widget that will be described in the next section.

The two option buttons, "Coarse Positioners" and "Fine Positioners", select which axes will be moved, the coarse or the fine axes respectively. For the actual movement, there are twelve buttons, labeled with arrows. Six "single" buttons with a thin arrow, and six "continuous" buttons with a thick, bordered arrow. The up and down buttons labeled with "X/Y" move in positive or negative y-direction, while the left and right buttons move the x-axis. The buttons labeled with "Z" move up and down along the z-axis.

For the continuous buttons, there are two movement modes, controlled by the status of the checkbox “Continuous buttons need to stay pressed to move”. Both modes make the axis step continuously in the indicated direction. If the checkbox is checked, the axis moves as long as the button stays pressed, and stops, when it is released. If the checkbox is not checked, the movement is started with one click, and stopped with another.

The single buttons have more different modes, selected in the “Single Button Mode” group. “Make steps” instructs the respective piezo actuator to perform the chosen number of steps when a button is pressed, “Move by this offset voltage” increases or decreases the offset voltage. The two “Move by this distance with ...” modes move the axis either by changing the offset voltage or by using steps. These movements are performed in open loop mode, i.e. the number of steps or the needed offset voltage is calculated by using preset values.

3.3.4 The Axis Settings Widget



ANC300 Config

Coarse X Axis (xc)

Axis Settings

Serial Nr.	ANM150A-M-1609-3011008	
Mode	Ground (gnd)	
Stepping Frequency	800,00	Hz
Stepping Voltage	35,00	V
Offset Voltage	0,00	V
AC In	☐ enabled	
DC In	☐ enabled	
Filter	off	
Measured Output	0,00	V
Last Capacitance	0,00	nF
Temperature (from Cap.)	0,00	K

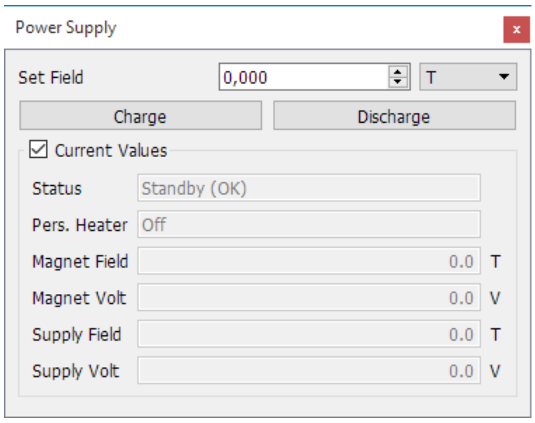
Figure 3.20: Axis settings dialog. Here the settings for each axis can be adjusted.

This widget, shown in figure 3.20, is used to configure the settings of each module of the ANC300 and therefore the settings of each axis of the positioning system. It is opened by either clicking the “Configure Axes” button of the position control widget, or by using the “Widgets” menu of the main window. Like the position control widget, this widget will also be disabled if the ANC300 is not connected to the PC.

The widget lists all settings for the currently selected axis, some of which can be changed, while others are just informative. The selected axis can be changed with the combo box that displays the axis name. When the axis is changed, the settings for the new axis are automatically loaded. On opening the widget, the axis that was last moved manually with the position control widget will be selected. There can be multiple instances of this widget, to allow the user to view and edit the parameters of multiple axes simultaneously.

If the value of a parameter is changed, the change is not immediately communicated to the ANC300. To apply the changes, the “Set Values” button has to be pressed. The “Reload Values” button re-reads all values from the ANC300, so it can be used to refresh the widget if some values were changed on the ANC300 itself, or to discard the changes made with the widget. The “Set For All Axes” button sets the displayed settings for all axes of the ANC300, while the “Set for all ... Axes” just sets the settings for all axes in the same group of the currently selected axis. Here “...” stands for either “Coarse” or “Fine”.

3.3.5 The Power Supply Widget


Power Supply

Set Field: 0,000 T

Charge | Discharge

☑ Current Values

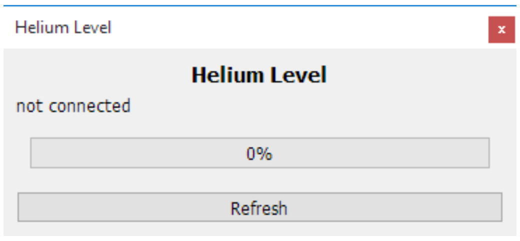
Status: Standby (OK)
Pers. Heater: Off
Magnet Field: 0.0 T
Magnet Volt: 0.0 V
Supply Field: 0.0 T
Supply Volt: 0.0 V

Figure 3.21: Power supply widget to monitor the status of the magnet power supply and to charge and discharge the magnet in the cryostat.

This widget, shown in figure 3.21, is used to control the Cryomagnetics 4G Magnet Power Supply, which in turn controls the superconducting magnet in the cryostat. The “Charge” button sweeps the magnet up to the field entered in the “Set Field” box, while the “Discharge” button sweeps the magnet back to zero. After being charged to the desired field, the magnet is kept at this field using the persistent switch heater (see section 2.2.6), and the power supply is then swept back to zero. The field can also be changed using the “Charge” button, if the magnet is already powered. In this case, the software automatically matches the field of the magnet and then enables the persistent switch heater, before sweeping to the new field.

If the “Current Values” check box is checked, the status of the power supply is periodically queried and displayed in the following fields. The units for the set field and the current field can be changed with the drop-down box next to the value for the set field. Available units are T, mT, G, kG, A, and mA. The power supply itself only supports kG or A, all other conversations are done by the software. For the magnetic field units to be correct, the “Gauss to Amp Ratio” has to be set correctly for the power supply to match the superconducting magnet. It can be set at the at the display of the power supply by first selecting “SETUP” and then “MAGNET”.

3.3.6 The Helium Level Widget



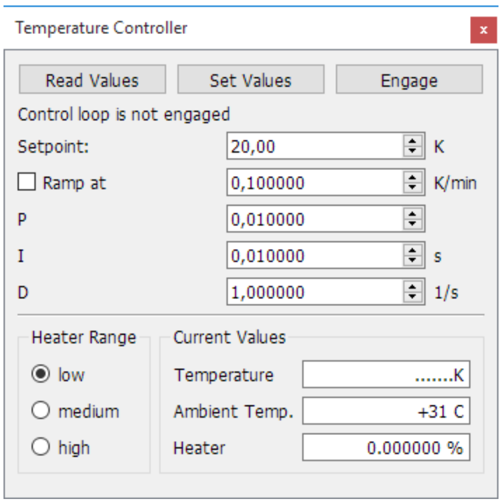
not connected

0%

Figure 3.22: Helium level widget to show the liquid helium level of the cryostat.

This widget, shown in figure 3.22, displays the current helium level of the cryostat using the LM-500 Liquid Helium Gauge (section 2.2.5). It is periodically refreshed as long as the helium gauge is connected to the PC, and can also be refreshed manually by using the “Refresh” button.

3.3.7 The Temperature Controller Widget



Control loop is not engaged

Setpoint:	20,00	K
☐ Ramp at	0,100000	K/min
P	0,010000	
I	0,010000	s
D	1,000000	1/s

Heater Range: [● low] ○ medium ○ high

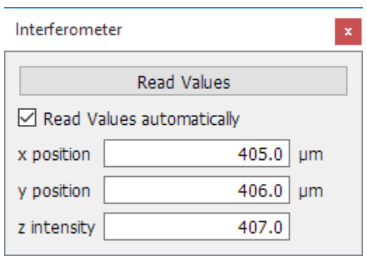
Temperature	.......K
Ambient Temp.	+31 C
Heater	0.000000 %

Figure 3.23: Temperature controller widget for changing the settings of the PID temperature controller and enabling or disabling the controlling.

This widget, shown in figure 3.23, controls the Cryocon Model 32B Temperature Controller (section 2.2.4). The setpoint and the P, I and D values (see section 1.3) can be set with the respective fields and sent to the device with the “Set Values” button. The “Read Values” button reads the stored values from the device and overwrites any previously made changes that have not yet been sent to the temperature controller. The “Engage” button starts the temperature controlling. It then changes its label to “Disengage”, which stops the controlling.

If the “Ramp at” checkbox is checked, the setpoint is approached with the selected ramp rate, then the standard PID type regulation will be performed as usual. The maximum heater output power can be set in three steps, “low”, “medium” and “high” (see section 2.2.4) with the corresponding “Heater Range” option buttons. The “Current Values” will be periodically queried from the temperature controller.

3.3.8 The Interferometer Widget



☑ Read Values automatically

x position 405.0 µm

y position 406.0 µm

z intensity 407.0

Figure 3.24: Interferometer widget to show the current position of the Hall probe over the sample as measured by the interferometer.

This widget, displayed in figure 3.24, shows the x-, y- and z-position of the Hall probe measured by the interferometer. The values can be read automatically by enabling the corresponding check box, or manually, using the “Read Values” button.

4 Software

The software is written in Python 3.5. This programming language was chosen because of its relative ease of use compared to many other languages. This allows the software to be modified by any user as easily and quickly as possible, without much preparation time. To further aid this and also improve the understandability and maintainability of the software, each function is documented in code in the NumPy documentation style. Algorithm 1 contains an example for this style, using a function of the ANC300 that puts some or all of its axes into ground mode.

```python
def set_all_to_ground_mode(self, axes=None):
    """
    Sets all axes to ground mode.

    Parameters
    ----------
    axes : None, int, str or list of int or str
        Axes to be moved. Axes can be referred to by name or number,
        eg [1, 'xc', 4].
        If None, all axes are set to ground mode.

    Returns
    -------
    bool
        True, if successful for all axes, False otherwise.
    """

    return self.set_all_mode(self.mode_ground, axes)
```

Algorithm 1: Example of the NumPy documentation style using a function of the ANC300. The red text delimited by triple quotation marks is the documentation string.

4.1 Needed Packages

The program needs at least Python 3.5 and should work with all higher versions. For this project, the Anaconda distribution in version 4.3.11 was used, because many helpful scientific packages are already pre-installed, although most of them are not needed for this program. A list of all needed, non standard packages is given below. Most of them can be installed with either the Python standard package manager (`pip`) or the anaconda package manager (`conda`), although some packages are only available via `pip`. If so, this will be indicated in the list. Each package should work with the newest version, the used version of each package will be included in the list.

- `pySerial 3.2.1` to communicate via serial connection (RS-232 or USB with simulated RS-232), this can only be installed with `pip`, because `conda` doesn't have the newest version at the time of writing
- `numpy 1.11.3` to facilitate fast array calculations (already installed with anaconda)
- `pyQt 5.6.0` for the user interface
- `pyVISA 1.8` to communicate with the GPIB to USB controller
- `matplotlib 2.0.0` for plotting (already installed with anaconda)
- `mido 1.1.24` to read `"*.midi"` files, not available on `conda`

The user interface was created with Qt, therefore an editor for Qt's `"*.ui"` files is needed to modify the user interface. For this project, Qt Creator 4.0.2 with Qt 5.7.0 was used.

4.2 File and Class Structure

The Python software, together with all its packages, is located at "D:/David/python_scanner_kraxn". It consists of three main packages: the "devices" package, for communicating with the different devices, the "gui" package responsible for the graphical user interface and a "misc" package, containing additional miscellaneous modules. The devices package also contains a sub package for helper functions. The main window for the user interface is managed by the "main_window" module, while "main" contains functions to connect and disconnect all devices and imports some modules needed for scripting (see section 4.3).

A module, in this context, is one Python file (with the extension ".py") that contains python functions and classes. A package is a collection of multiple modules and/or other packages, which are bundled together in a folder with the package's name. Figures 4.1 and 4.2 show the directory structure of the whole Python project.

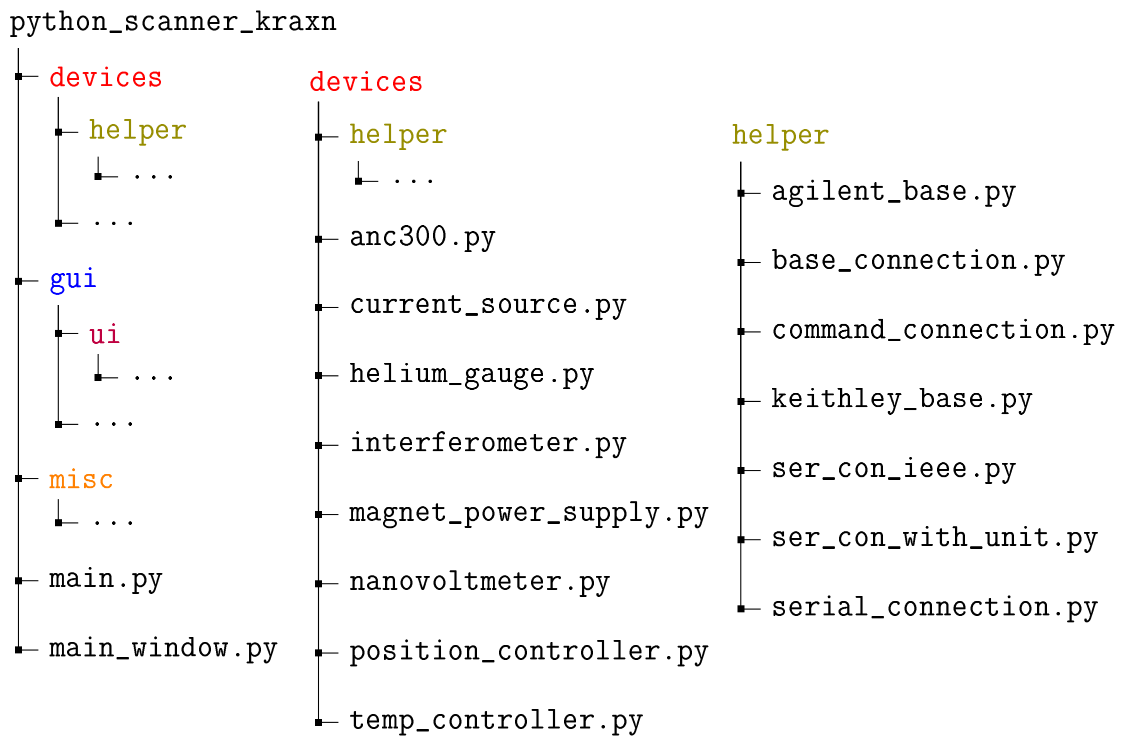


Figure 4.1: File structure of the whole Python project, part 1 (for part 2 see figure 4.2). All files in the root program directory "python_scanner_kraxn", as well as in the "devices" and "helper" packages are shown, except for the empty "__init__.py" files that are needed for every package. "..." denotes the contents of a package that are shown extra, to make the graphic more compact. Packages are color coded, so that every package can be easily found in every tree.

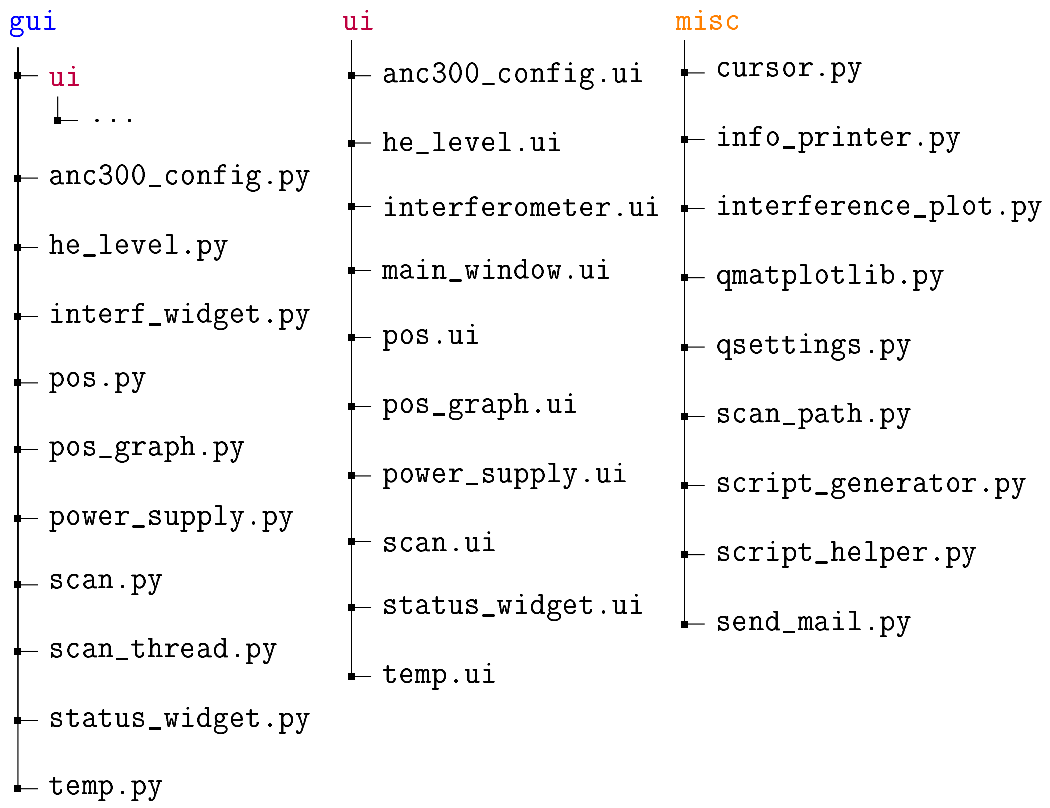


Figure 4.2: File structure of the whole Python project, part 2. All files in the "gui", "ui" and "misc" packages are shown, except for the empty "__init__.py" files that are needed for every package. "..." denotes the contents of a package that are shown extra, to make the graphic more compact. Packages are color coded, so that every package can be easily found in every tree. The "ui" folder contains the files that define the user interface.

4.2.1 The "devices" package

All code for communication with the different devices is contained in the "devices" package. The inheritance hierarchy for the devices is shown in figure 4.3, where the "devices/" prefix for every node is omitted for readability. Each device is contained in a module in the "devices" package, while the "helper" sub-package contains helper classes to avoid code duplication.

All devices inherit from "helper/base_connection", so the basic API for all devices is similar. For example, device names and signals for connecting and disconnecting are defined in this module. Additionally, "helper/base_connection" inherits from "misc/info_printer" (not contained in the devices-package), to print and log status as well as error messages. “command_connection” implements sending and receiving commands as strings, without requiring a specific connection type. The latter is done by the “serial_connection” module, which implements an RS-232 connection (or a simulated RS-232 connection via USB). “ser_con_ieee” implements some IEEE standard commands, which are shared by many devices, and “ser_con_with_unit” defines standard commands to get and set units. “current_source” and “nanovoltmeter” both share some basic Keithley-specific functions which are contained in “keithley_base”. Lastly, “agilent_base” establishes communication with the Agilent GPIB to USB controller. The next-to-last row in figure 4.3 contains all individual devices; “position_controller” combines the “interferometer” and “anc300” modules, to be able to move to a desired position with feedback.

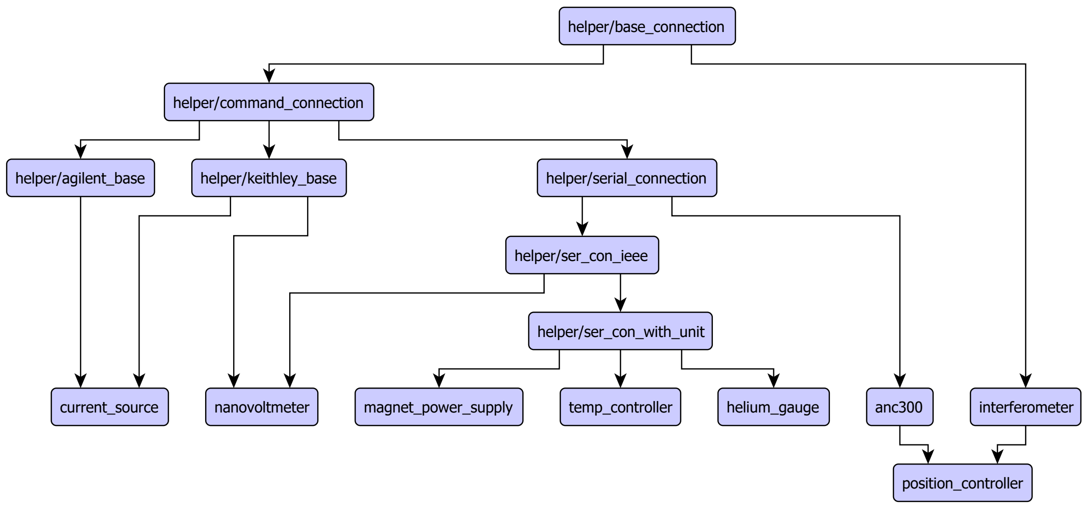


Figure 4.3: Inheritance hierarchy for the devices-package. Each node represents a module, the two last lines show the modules to communicate with each device, while all other modules contain helper functions that are used by multiple devices.

The “anc300” module This module contains the classes and functions for communicating with the ANC300. It consists of two classes, one representing the ANC300 controller, the other one representing one piezo axis. Each axis, consisting of an ANM150 or an ANM300 module connected to a piezo actuator, thus has an associated “Axis” class that contains axis specific settings and handles the communication with its individual axis module. This class contains all the command and query functions for the axis, e.g. a function to set the axis to a specific mode or to get its currently set stepping voltage.

The "Anc300" class is instantiated once, to connect to the ANC300 controller. It contains settings that affect all axes equally. Upon connecting, it creates an "Axis" class instance for each axis, or module, of the ANC300. Most functions of the "Axis" class have a corresponding function in the "Anc300" class, with the difference that here the functionality will be carried out for some or all axes simultaneously.

4.2.2 The "misc" package

The "misc" package contains some miscellaneous modules, that will be discussed in detail in this section.

The "info_printer" module is used to print status messages, as well as error messages. It also implements the logging capabilities, which enable the application to save these messages to a log file. If the application crashes unexpectedly, the resulting error message is also saved to the log file. All messages that occur during a scan are saved to a file in the log directory with the same name as the scan file, but with the added extension `''.log''`. All other messages that do not occur during a scan are consolidated for each day and saved to a file named with the current date. Log files are never overwritten, new data is always appended.

The "scan_path" module contains classes that define and save the scan area parameters as discussed in section 3.2.2. The `ScanPath` class defines an iterable object that returns a new `PointInfo` class on each iteration. It defines the points to scan and is iterated over during a scan. The `PointInfo` class contains the x- and y-coordinates of the next scan point, together with progress information, to print useful status messages.

There are two types of scan paths, inheriting from the basic `ScanPath` class: the `RectangleGrid` and the `CombinedPath`. The `RectangleGrid` represents a rectangular grid of points as discussed and adjusted in section 3.2.2. The `CombinedPath` can contain a list of `ScanPaths` (possibly including other `CombinedPaths`) and iterates over all of them, one after the other. This enables the combination of multiple scan areas for one scan, as explained in section 3.2.3.

The "script_generator" module is used to create a script from the settings made in the user interface (see paragraph "Save and Load Settings" of section 3.2.4).

The "script_helper" module contains functions to simplify the writing of a script as will be discussed in section 4.3.

The "send_mail" module implements email capabilities, so the scanning software can send an email when the scan is finished or an error occurred.

The "interference_plot" module is not used for the main application. It can continuously plot measurements made with the nanovoltmeter connected directly to the PC (see section 2.2.2). This was used to measure interference patterns for setting up the interferometer as discussed in section 2.3.

The "qsettings" module contains the functions to export and import scan settings to a ``*.ini'' file as described in section 3.2.4.

The "cursor" and "qmatplotlib modules enable zooming and panning in the graph widget, which is plotted using the matplotlib library.

4.3 Scripting Capabilities

Everything that can be done with the user interface, and more, can be accomplished with a Python script, using the in-built scripting capabilities. The easiest way to create a script, is to define a scan with the user interface and then use the "Export script" button to create a script that defines a scan with all the settings defined in the user interface. This script can then be modified as needed.

What follows, is an example script that was generated by the script generator module, followed by a detailed description of all the commands used.

```
import main


def start():

    main.connect()

    scan_path = main.scan_path.RectangleGrid(
        center_point=[0, 0],
        stepsize=[1, 1],
        size=[2, 2],
        angle=0,
        line_axis_index=0,
        back_and_forth=False)

    scan_thread = main.ScanThread(
        main.anc, main.cur, main.itf, main.hel, main.tco, main.mag)

    # For testing without devices
    # scan_thread = main.ScanThread(
    #     main.anc, None, None, None, None, None)

    scan_thread.set_file('D:/David/scan_file.txt', False)
    scan_thread.set_scan_path(scan_path)
    scan_thread.set_report_mail('')

    scan_thread.set_tapping_mode(
        tapping_enabled=False,
        tap_z_distance=1,
        tap_at_distance_enabled=True,
        tap_every_distance=1,
        tap_at_newline_enabled=True)
    scan_thread.set_closed_loop_approach(True)
    scan_thread.set_retract_offset(0)
    scan_thread.set_retract_every_point_offset(0)
    scan_thread.set_slope_disabled()

    main.script_helper.connect_and_start_scan_thread(scan_thread)


if __name__ == '__main__':
    start()
```

The first line, `import main`, imports the "main" module of the scanner software, which contains some helper functions and references to all devices. The function defined by `def start():` contains all the relevant code, while the very last two lines of the code just start this function, when the script is run.

`scan_path = main.scan_path.RectangleGrid(...)` creates the scan path element defining the points to be scanned. The first parameter, `center_point=[0, 0]` defines the x- and y-coordinates of the center of the scan rectangle, `stepsize=[1, 1]` sets the step size in x- and y-direction in µm, `size=[2, 2]` sets the total size of the scanned rectangle in µm, `angle=0` sets the rotation angle to 0 degrees, `line_axis_index=0` sets the x-axis as the line scan axis (1 would set the y-axis) and `back_and_forth=False` disables back-and-forth-mode. This means that each line is scanned in the same direction, while `back_and_forth=True` would scan alternating lines in alternating directions.

`scan_thread = main.ScanThread(...)` creates the scan thread that carries out the actual scanning. Here `main.anc` is the "Anc300" class, `main.cur` is the current source, `main.itf` is the interferometer, `main.hel` is the helium gauge, `main.tco` is the temperature controller and `main.mag` is the magnet power supply. If any of these devices cannot be connected, the scan will not be started, but if `None` is given instead of a device, this device will be simulated and the scan will be performed. This can be used for testing purposes. Each of these devices can also be controlled directly by the script using these variables.

The line `scan_thread.set_file('D:/scan_file.txt', False)` sets the scan file, where all the data will be saved. The Boolean value defines, if an already existing file should be overwritten (`True`) or if the new data should be appended to it (`False`).

The line `scan_thread.set_scan_path(scan_path)` gives the scan thread the previously created scan path, and with `scan_thread.set_report_mail('')` the email addresses to report errors or completion of the scan are set. Multiple email addresses can be set by separating them with a comma.

`scan_thread.set_tapping_mode(...)` enables or disables the tapping mode and sets its parameters. `tapping_enabled=False` disables tapping mode. To enable tapping mode, it has to be set to `True`, otherwise all other parameters are ignored. `tap_z_distance=1` sets the distance to retract after tapping in µm, `tap_at_distance_enabled=True` performs a tap, when the next scan point is more than `tap_every_distance` away from the last tapping point. `tap_at_newline_enabled=True` performs a tap whenever a new scan line is started.

`scan_thread.set_closed_loop_approach(True)` enables the use of the interferometer for positioning, `False` would disable it.

`scan_thread.set_retract_offset(0)` sets the offset to retract the probe when returning for a new line, an offset of 0 disables the retraction. `scan_thread.set_retract_every_point_offset(0)` sets the offset to retract between each scan point, this retraction can also be disabled by setting the offset to 0.

`scan_thread.set_slope_disabled()` disables the slope compensation. To enable slope compensation, the function `scan_thread.set_slope(p1, p2, p3)` is used, where p1, p2, and p3 are each a list of the x-, y- and z-coordinates of the three points defining the slope plane (see section 3.2.4).

`main.script_helper.connect_and_start_scan_thread(scan_thread)` connects all devices as well as the callback functions of the scan thread and then starts the scan. Status information will be printed to the console.

5 Conclusion and Outlook

The software for communicating with and controlling the devices needed for the Nano Hall Scanner is completed. It is possible to define and perform scans with a user interface or with a scripting language and to control all six axes of the positioning system, as well as all devices. The user interface consists of multiple widgets, that can be reordered or disabled to fit the user's preferences. For more fine grained control over all aspects of the devices and the scan, there is the scripting interface. This allows the user to intervene at every level of the software using normal python code, while still being able to utilize the existing functions. Because these functions are well documented in the code, their purpose is self explanatory.

The next step for the scanner setup will be to test the scanner together with the interferometer. For this a known sample will be scanned in order to make sure that the setup works properly. This can be done as soon as the interferometer, as well as the mirror setup for the positioning system, are completed. Depending on the properties of the interferometer and the results of these tests, it may be decided to refine the positioning algorithm of the scanning software, to account for any changes or apply improvements. The very versatile concept of the software makes it possible to easily change some aspect of the software, like the communication with the interferometer, should the need be recognized during testing.

List of Tables

List of Figures

References

[1] attocube systems AG. *ANRv220 open loop, rotary stepper positioner with horizontal rotation axis*, April 2013.

[2] attocube systems AG. *ANRv51 Technical Specifications*, 2015.

[3] attocube systems AG. *ANSxyz100std Technical Specifications*, 2015.

[4] attocube systems AG. *User Manual - Premium Line: Positioners & Scanners*, June 2015.

[5] attocube systems AG. *User Manual Piezo Positioning Electronic ANC300*, July 2016.

[6] attocube systems AG. *ANPz102 Technical Specifications*, 2017.

[7] Cryogenic Control Systems, Inc. *User's Guide Model 32 & 32B Cryogenic Temperature Controller*, March 2008.

[8] Cryomagnetics, Inc. *Operating Instruction Manual for the Model LM-500 Cryogenic Liquid Level Monitor*, May 2004.

[9] Cryomagnetics, Inc. *Operating Instruction Manual for the 4G Magnet Power Supply*, February 2010.

[10] Keithley Instruments, Inc. *Model 2182/2182A Nanovoltmeter Service Manual*, July 2004.

[11] Keithley Instruments, Inc. *Model 2182/2182A Nanovoltmeter User's Manual*, June 2004.

[12] Keithley Instruments, Inc. *Model 6220 DC Current Source Model 6221 AC and DC Current Source Reference Manual*, October 2008.

[13] Keithley Instruments, Inc. *Model 6220 DC Current Source Model 6221 AC and DC Current Source Users Manual*, October 2008.