

Setting up the scanning system

Threaded rods for fixing horizontal absorbers:

- 9 st - M4 (4 mm diameter) - length = 10 mm (Al base) + 70 mm (tungsten pieces = $2 \times 19 + 2 \times 13 + 3 \times 2 = 70$) ~ 80 mm
- <https://tools.se/produkterSe/fastteknik/industriinfastning/helgangad-stang/Gangstang-HGS-4.8-ANS-FZB-1411093-sv-se#v=SE-2605977>
- Malmö!

These have been processed and tested.

Screws for setting up LYCCA ring:

12 cylindrical distance pins for LYCCA ring top <-> bottom - 2x12 M6 screws:

- From top: $l = 10 + 10 = 20$ mm, no protruding head
- <https://tools.se/produkterSe/fastteknik/industriinfastning/maskingangad-insexskruf/-SchaferPeters-A2-1411468-sv-se?categoryId=1073748027#v=SE-2027270>
- Here we chose M6x25mm with no head!
-
- From bottom: Drill small larger “holes” for head? $-> l = 10$ (Al base) + 10 (Plexi bottom) + 10 (in distance pin) = 30 mm
- <https://tools.se/produkterSe/fastteknik/industriinfastning/maskingangad-insexskruf/Insexskruf-MF6S-10.9-Forsankt-cyl.-huvud-ANS-FZB-1411311-sv-se>
- Here we chose M6x35 or M6x30mm (if former does not work) with försänkt head!

2 screws for correct positioning of the LYCCA base on the table

- $l = 10 + 10 + 15 + (\text{depth in table}) = 15 = 50$ (max), this is M6!
- They cannot have head or something like that. See storage in lab.

12 M3 screws for fixing each LYCCA module in the bottom plate

- Or is it possible with the final module configuration?
- $l = 5$ mm, head not very important!?
- This length does not exist!!
- WE HAVE IT IN_HOUSE!! This is true!

Dubbelhäftande tape to fix the 4 plastic boxes on the preamplifiers and the preamps to the bottom plate.

- <https://tools.se/produkterSe/fastighet/tejp-och-tatningslist/ovrigt-tejp-och-hallare/Mattejp-Universal-561705617156172-TESA-1405469-sv-se>

Screws for collimator

The following has been measured (photo also exists):

- Collimator <-> Source holder:
 - M4x15 mm, head 3.5x6mm² (thickness x diam)
- Collimator <-> Lock:
 - M6x50 (25 threaded) mm, head 5x9mm² (thickness x diam)
 - **OBS** with this we only use 10mm of the thread in the collimator!!!
 - There are no perfect one in house!

Screws top-table and positioning system

- KK50-bottom <-> top plate: 6 st; HEAD=8x4, THRU=4.5 (no thread), thickness top plate = 10 mm
 - Final: 6 st, M4x25mm with M4 nut + M4 plate
- KK50-wagon <-> KK60-base:
 - Wagon: 4 st; M4x0.7
 - KK60-base; 4?=2; HEAD 9.5x4.7, THRU=5.5
 - * Final: 2 st M4x10mm. These have been optimised for the usage of the most threads in the KK50 wagon. M4x12mm was protruding! Fix these really tight as they need to sustain a large load.
- Linear guide base <-> top-plate: 4 st; 7.5(measured~8)x5.3, THRU=4.5x(15-5.3=9.7), thickness top plate = 10 mm
 - Final: 6 st M4x25mm + M4 nut + M4 plate
- Linear guide wagon <-> KK60-base:
 - Wagon: M4 THRU
 - KK60-base; 4?=2; HEAD 9.5x4.7, THRU=5.5
 - Final: M5x25mm. Nuts and plates were too tedious to install.
- Collimator arm <-> KK60-wagon:
 - Wagon: 4 st; M5x0.8
 - Thickness collimator arm = 10 mm

* Final: M5x18 + plate M5 + plate M5 underneath the arm (!)
for the two back screws. (Perhaps the machine drill bent it? :()

Remarks: Sadly no plates matches the space for the screw heads on the base of the linear units, unclear why this construction.

- Top-table legs <-> top-table:
 - 4 st M8x35 head 7x12mm² (thickness x diam)
 - These were the longest found in-house.
- Top-table legs <-> bottom-table:
 - 8 st “drilling” screws. Measured: “M”3(4?)x15mm and head 1x11mm² (thickness x diam)
 - **OBS** these are measured values

Mounting the wagon onto the linear guide

1. First a plastic temporary filler was removed from underneath the oiled up wagon.
2. The wagon needs to be slowly and carefully slid onto the guide.
 - At first try 2 of the bearing balls were pushed out of their course.
 - With a forceps (pincett) the cleaned balls were moved back.
 - At second try it was successful.
3. It is important that the guide is not fixed when run, as it needs to adapt its path to horizontal with the linear unit.

Mounting the collimator arm on the KK60 and placing the collimator

1. Make sure that you mount it in such a way that the arm extends on that side of the KK60 where there is no “frame” sticking out.
2. Fix the two screws (see above for which screws to use) with the longest distance from the arm first. Fix as much as possible and verify that they are positioned correctly by observing horizontal arm (both on wagon and w.r.t. table).
3. Fix the other screws as good as possible.
4. Place the collimator and observe no change in the set-up.

Setting up the Raspberry Pi - Gertbot - stepper motors

1. Before booting the *Lundiumberry* (Raspberry Pi 3, Model B installed with Raspbian): The Gertbot goes on top of the pins on the raspberry pi (only one way, otherwise it will hit ethernet connector)

2. The 2 phase stepper motors *17H261-02S/D* are run with 1.2 A per phase. On the specification sheet it is stated that each phase has a resistance of 4 Ohm. According to <http://compotech.se/blogg/2014/06/sa-valjer-du-ratt-spanning-din-stegmotordrift/> at least 4.8 V should be enough. 6 V was chosen as an objective.
3. The Gertbot controller requires at least 8 V (8-18 V is recommended, see gertbot.com) to function. This has the consequence that it is necessary to connect a resistor in series with the phase of the stepper motors. A 15 V nominal input voltage was chosen which implies 6 Ohms resistors.
4. 6.7 Ohms, 5.5 W resistors were connected via a connection board. Assuming a motor current of 1.2 A we need ~ 10 W resistors and these are getting somewhat too hot.
5. The stepper motors were connected to the Gertbot as illustrated in the figures below!

Important convention

- Green wire connected to A1 and striped green wire to A2, red wire connected to B1 and striped red wire to B2. Correspondingly for 2nd connected stepper motor. Important since this determines the polarity of rotation and this way it is congruent with + = going away from motor housing.
- The bottom stepper motor (running the KK50, i.e. the smallest) should be connected to the first 2 gates of the Gertbot. This means the 6 pin contacts close to the long side of the gertbot. In the end this convention rules which stepper motor (#0 and #2 in the GUI) is called what in the program and that the end-stop is activated for the correct stepper motor+direction.
- The motors are referred to **STEPPER_X** and **STEPPER_Y** where the view of reference is: the wagons are as close as possible to the motor housing and one looks in the direction of the bottom linear unit. This position can be referred to the motor origin. Hence, **STEPPER_X** = **Top-motor** (KK60) and **STEPPER_Y** = **bottom-motor** (KK50). This is especially used in the stepper software but also in the cableing of the ScanningComputer.

Configuring *Lundiumberry* with Gertbot

1. From <https://www.gertbot.com/download.html> download:
 - Gertbot debug GUI, executable for Rasbian
 - Drivers for Python 3 (.py)
 - Executable to enable the uart
2. Follow: https://www.isnr.de/images/Tools/low_cost_CT/How-to-set-up-the-Raspberry-Pi-and-Gertbot-30-08-2016.pdf

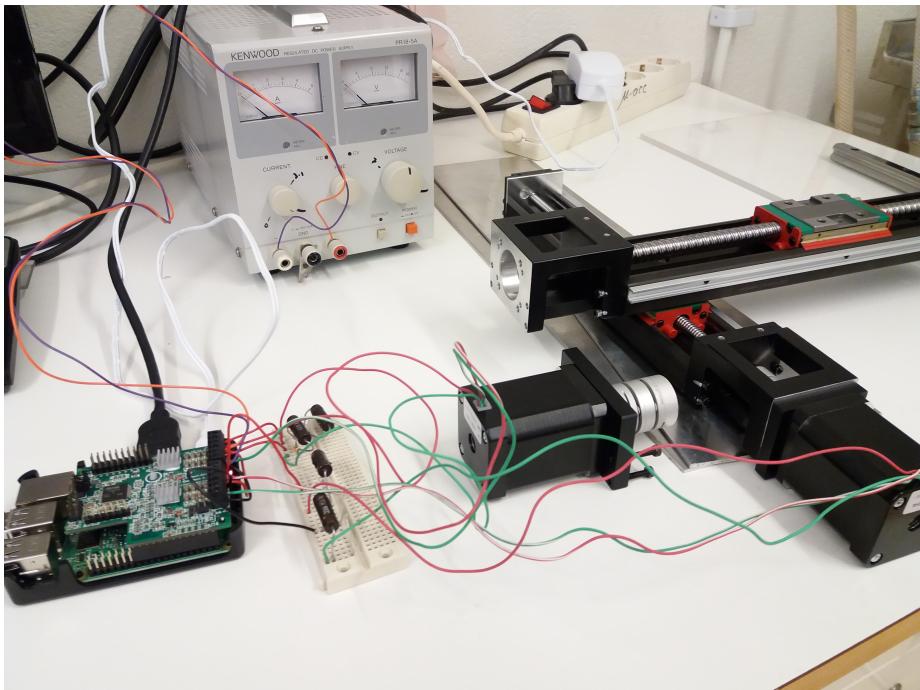


Figure 1: Stepper motor connection

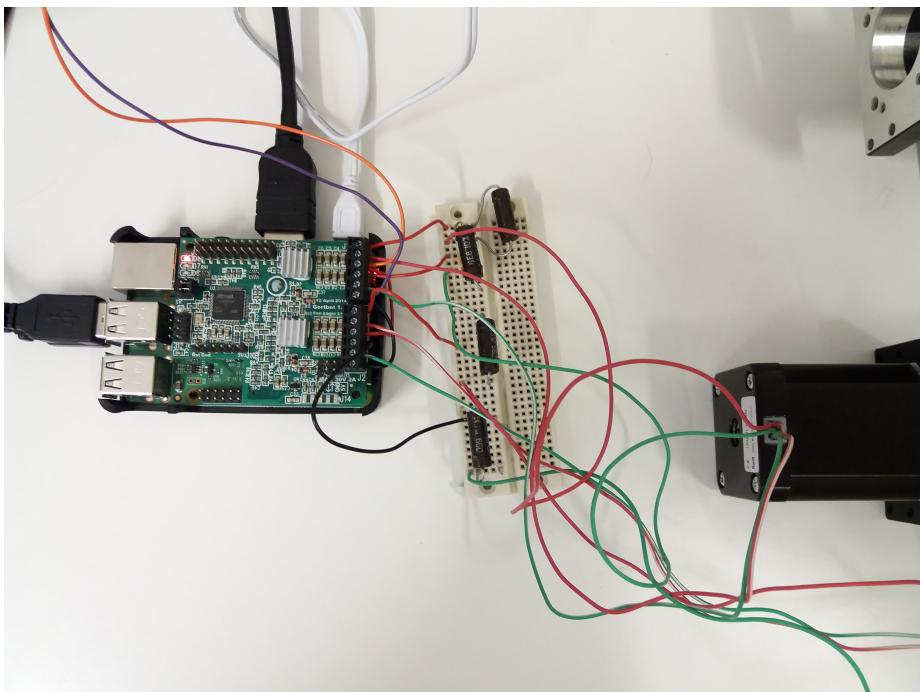


Figure 2: Stepper motor connection (close up)

3. Execute GUI for debug with: `sudo ./gertbot`
 - Click connect and one board (#3) should be found
 - For channels 1 and 3 choose `Step gray off` and try it out.
 - For details of the GUI: https://www.gertbot.com/gbdownload/man/Gertbot_GUI.pdf

Activating end-stops

The following fork couplers were purchased:

5 st (OPB990T51Z): https://www.elfa.se/sv/laesgaffel-18-mm-40-ma-30-ma-40-70-optek-opb990t51z/p/30009110?channel=b2b&price_afd=48.2&gclid=CjwKEAjwja_JBRD8idHpxaz0t3wSJAB4rXW5Xf4yJTT7wcB

Documentation: https://www.elfa.se/Web/Downloads/_t/ds/OPB960-990-series_eng_tds.pdf?mime=application%2Fpdf

Wires:

- Red - anode (diode)
- Black - cathode (diode)
- White - V_CC (phototransistor)
- Blue - Output (phototransistor)
- Green - Ground (phototransistor)

If only one sensor is connected then this limiting resistor should be sufficient:

330 Ohms resistor -> Diode current ~ 24 mA

OBS : Check the J3-pins (the ones to connect the sensor output to) in order to connect it properly. Be aware every stepper motor has two end-stops: *A* and *B*. The *B* is the upper most pin (in the config. in the figures). Each end-stop is configured for a specific polarity of the stepper motor rotation (*A*=+ and *B*=-).

If the GUI is used for testing the end-stops you might run into trouble when it is activated. The *Gertbot* is programmed to change to `Step Gray PWR` mode when the end-stop is active. This has been circumvented in the .py script where the mode is reset if such an event would occur. In the script the motor status is checked in order to disentangle if an error (end-stop) is apparent.

Installation of ScanningComputer

The detailed connections of the stepper motor wires to the Gertbot and endstops to the Gertbot see above section. Here goes the fine tune cableing to set up the *ScanningComputer*, which is the box with cables that goes to and from the raspberry pie, Gertbot and the power supplies. See figure.

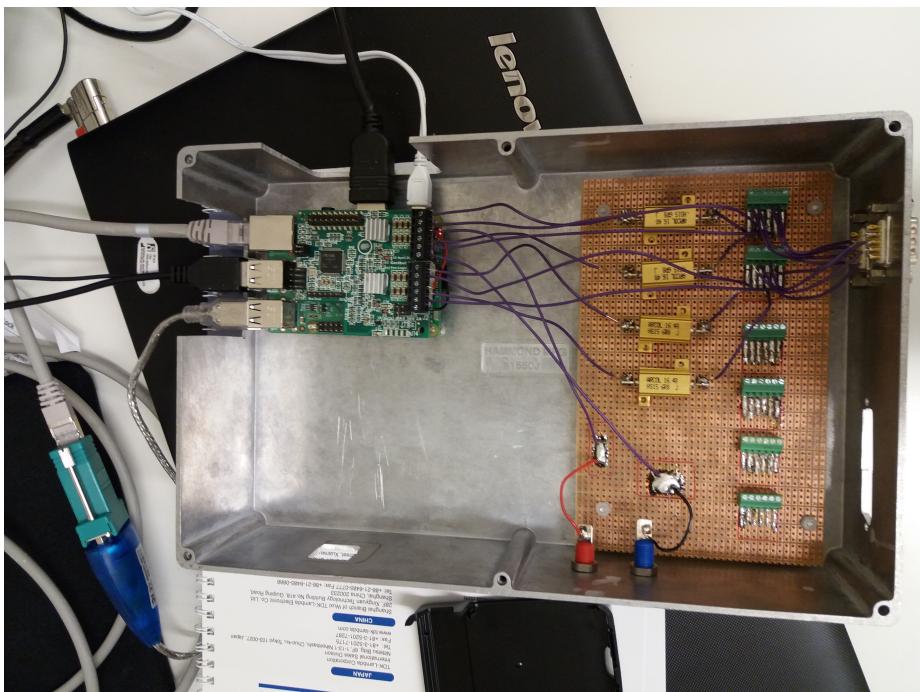


Figure 3: ScanningComputer

** Stepper motors -> dsub 9 **: Green -> 1, striped green -> 2, red -> 3 and striped red -> 4.

** Sensors -> dsub 25 **: Red -> 1, black -> 2, white -> 3, blue -> 4 and green -> 5. This pattern continued for all 4 sensors. Further the following was connected:

1. $X0$ -> 1-5
2. $XMAX$ -> 6-10
3. $Y0$ -> 11-15
4. $YMAX$ -> 16-20

Here ($X0, Y0$) is referred to the sensor activated when the origin (see earlier discussion) has been reached. ($XMAX, YMAX$) is referred to the sensor activated when the maximum dislocation, with respect to the motor housing, is reached.

Software control

The control software has been implemented in cd `ScanningSystem/atlundiumberry`. The program is run with: `python3 move_collimator.py`. If no command line options are provided a substantial help information is given.

Code documentation can to a large extent be found in the source files. The program enables:

Usage:

- Specify how many steps that should be taken in as: ‘steps_x steps_y’
- [-h] [-step STEPS STEPS] [-N NEW_XY NEW_XY] [-new] [-xy XY XY]
- [-file_xy FILE_XY]

Optional arguments:

- -h, -help show this help message and exit
- -step STEPS STEPS steps_x steps_y
- -N NEW_XY NEW_XY Start a new run from scratch with new origin.
- -new Start a new run from scratch with origin the same last saved.
- -xy XY XY Provide the new coordinates as: ‘x y’ (mm)
- -file_xy FILE_XY Provide the file name from which the position data is to be read from.