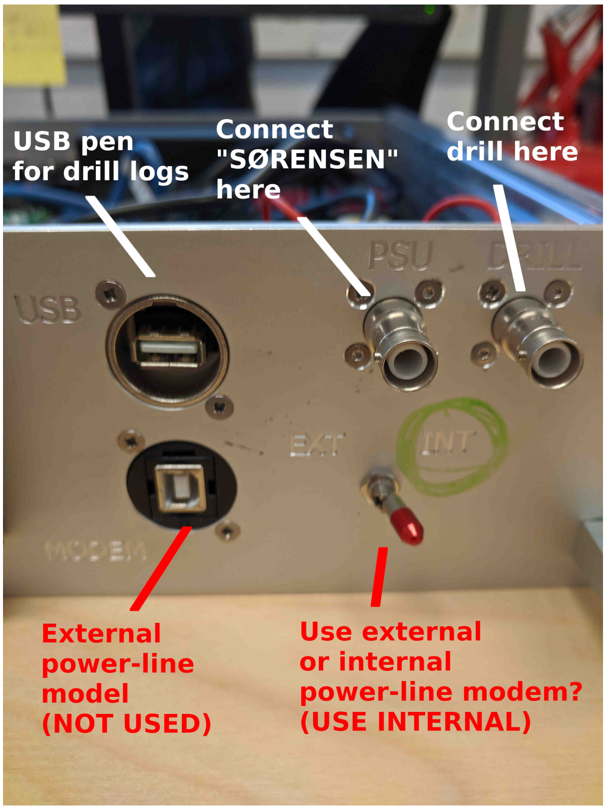
Surface Unit documentation

Niels Bohr Institute, Uni. Copenhagen

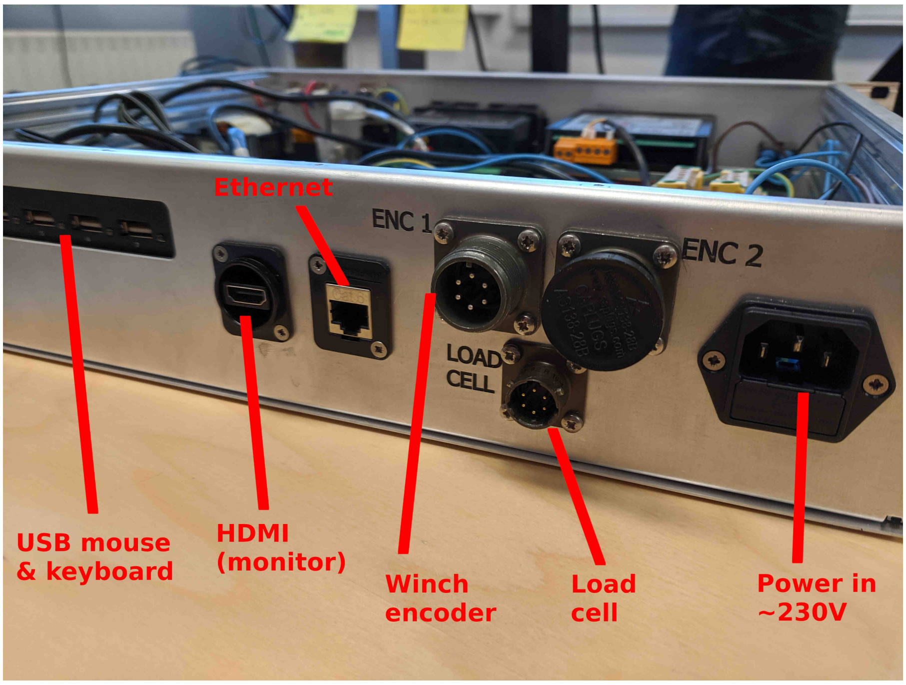
Nicholas Rathmann and Aslak Grinsted
Version 2024, June

Connecting the Surface Unit

Frontside

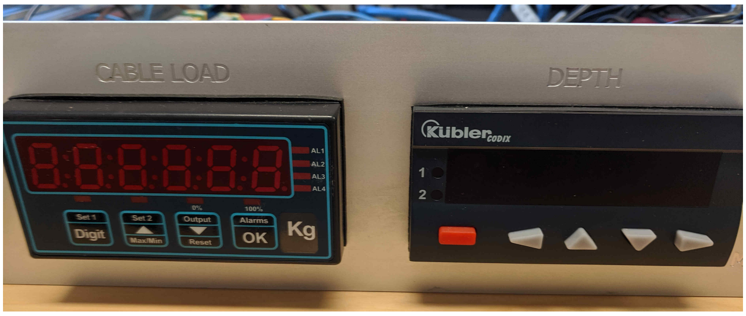


Backside



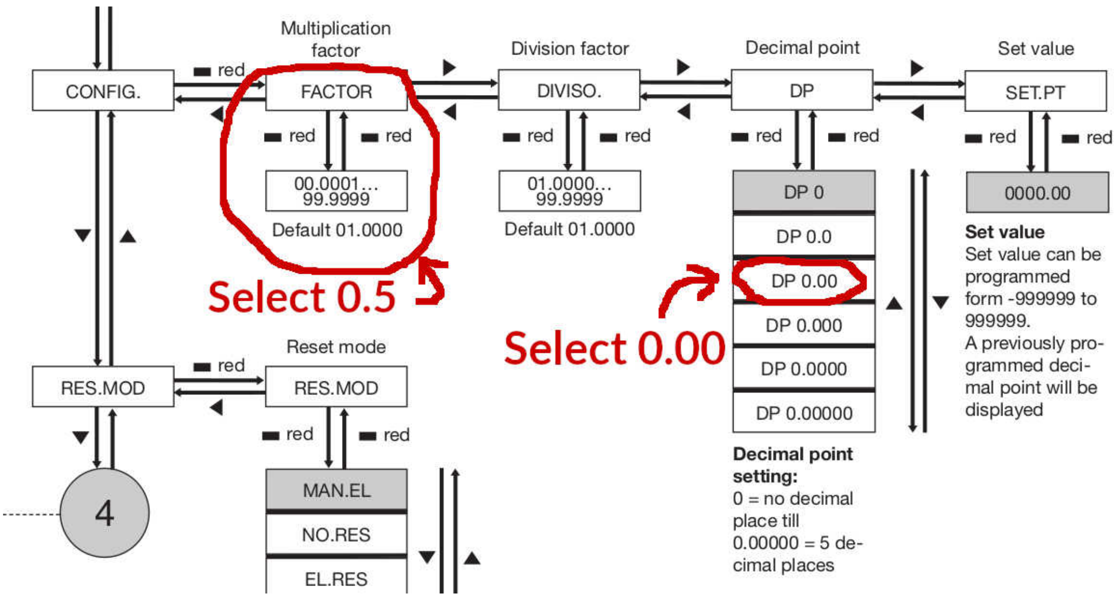
Configuring the displays

The surface unit has two front-panel displays, a load cell and depth counter:



Depth counter

- Maker, model: Kubler, CODIX 560
- Manual can be found here (or at `documentation/manuals/kuebler_codix_en.pdf`)
- Device menu navigation can be found here (or at `documentation/manuals/kuebler_codix_en_menuguide.pdf`)
- The settings used with the Danish deep drill are (slice of menu navigation):



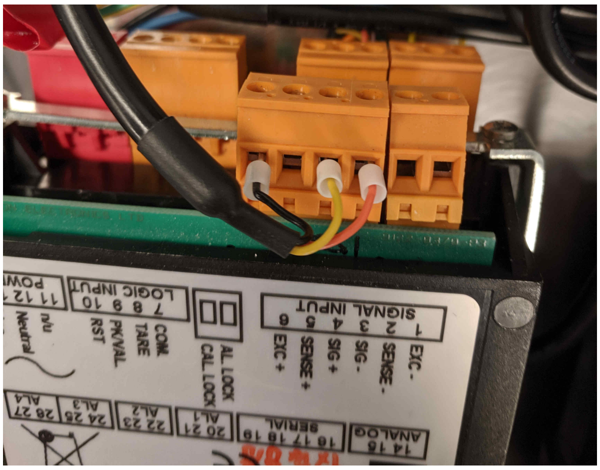
Load cell

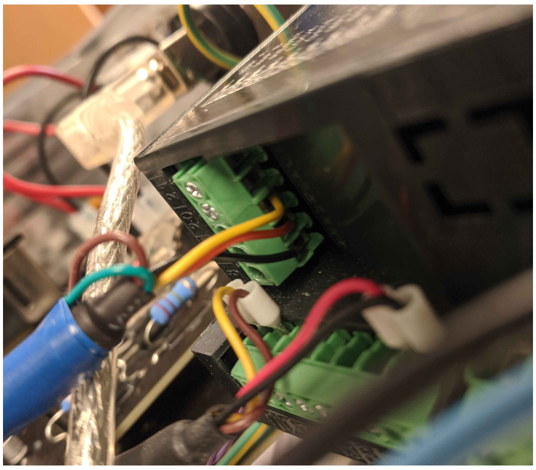
- Maker, model: LCM Systems, PMD-STRAIN
- Manual can be found here (or at `documentation/manuals/PMD-Strain.pdf`)

- The load cell must be calibrated using the "Direct Calibration" method, which requires applying zero load and a large load, preferably 1500kg to 2000kg.
- The relevant pages of the manual for performing the calibration routine can be found here (or at `documentation/surface-displays-calibration/pmd-strain-calibration.pdf`).

Serial wiring

The serial (USB) wiring for the load cell and depth counter are, respectively:





The serial cable documentation can be found here (or at `documentation/manuals/USB_RS232_cables.pdf`).

Software overview

- The Surface Unit software is available here.
- On the Surface Unit, all the software is located at `/home/drill/surface-unit`.
- All manuals, datasheets, and other documentation are located at `/home/drill/surface-unit/documentation`.
- The software can be updated to the latest version by opening a terminal and running

```
cd surface-unit
bash update.sh
```

Installing

If wanting to install the software from scratch, note that:

- The software relies on both `python2` and `python3`, and

```
apt-get install dialog redis-server
```

- The following python packages are required by the Surface Unit software

```
pip2 install redis minimalmodbus docopt termcolor pyvesc pythoncrc
pip3 install redis minimalmodbus docopt termcolor pyvesc pythoncrc
pip3 install numpy scipy ahrs PyQt5 redis pyqtgraph
```

- The following *additional* packages are required for processing logs etc.

```
pip3 install pandas
```

Booting the Surface Unit

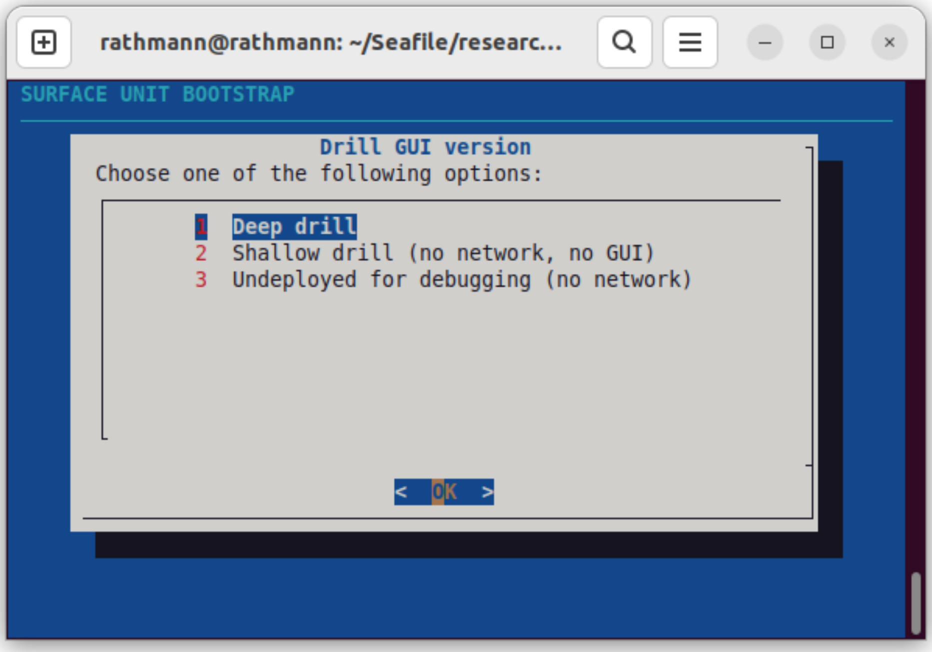
- When powered on, a bootstrapping script should run automatically

If not, open a terminal and type:

```
bash surface-unit/drill-bootstrap.sh
```

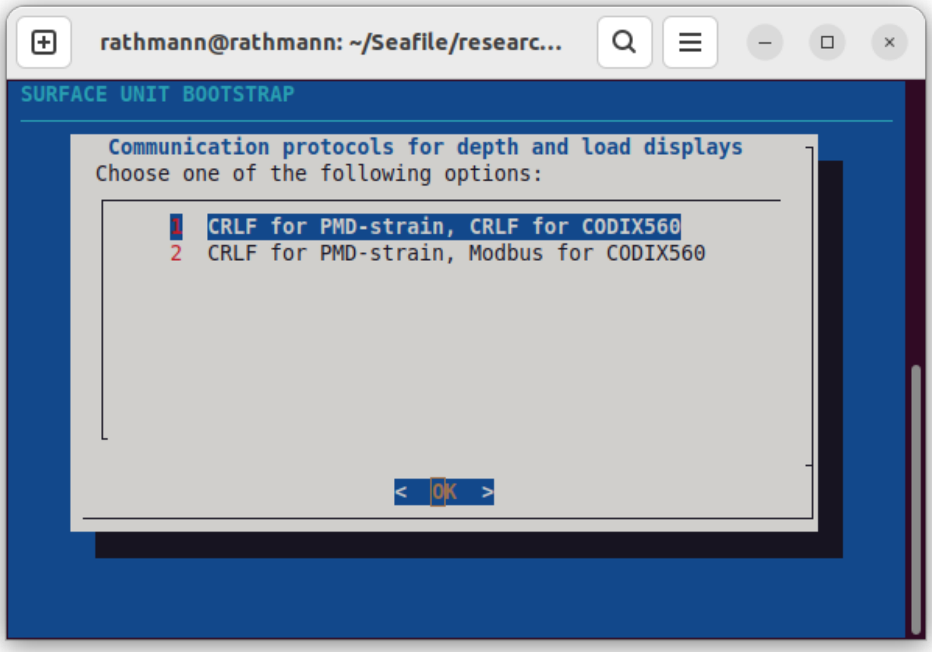
- Select the appropriate deployment

- "Deep drill" assumes a full deployment with internet access
- "Shallow drill" assumes a standalone deployment without internet access
- "Undeployed" is for debugging and should not be selected



- Select the communication protocol for load and depth displays

This is written on labels above the displays (*CRLF* or *MODBUS*).



The bootstrap script

The bootstrap script runs the following tasks:

- Network configuration

If full (deep drill) deployment is selected, the surface unit is assumed to be on camp network with a fixed IP address and internet access:

```
sudo dhcpcd -S ip_address=10.2.3.10/16 \
    -S routers=10.2.1.1 -S domain_name_servers=10.2.1.1 eth0
```

but if you wish to instead get an IP address from a local DHCP server, run in a terminal:

```
sudo dhcpcd eth0
```

If unable to resolve hostnames, try moving "dns" forward (immediately after “myhostname”) in `/etc/nsswitch.conf`.

- Clock synchronization

```
sudo systemctl restart systemd-timesyncd.service
sudo timedatectl set-ntp true &
sudo ntpdate 0.arch.pool.ntp.org
```

but if you want to set the time manually, run e.g.:

```
sudo timedatectl set-time '2022-11-20 16:14:50'
```

- Mount the USB pen for saving drill logs

```
sudo mount /dev/sda1 /mnt/logs/ -o umask=000
```

- Run the communication script for the depth display

```
python2 surface-displays/codix560.py
```

- Run the communication script for the load display

```
python2 surface-displays/pmdstrain.py
```

- Run the drill communication backend

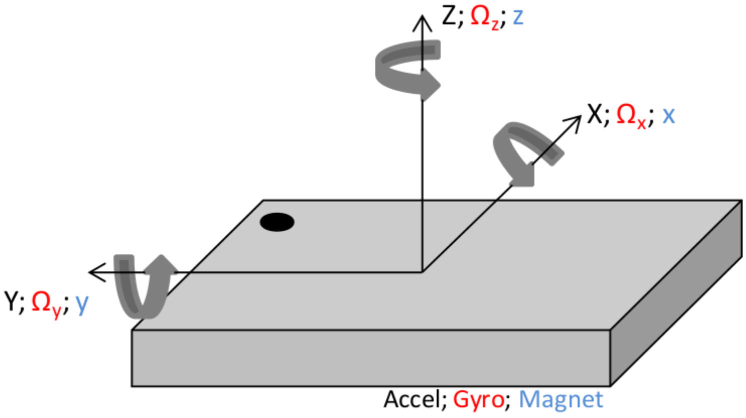
```
python3 drill-dispatch/dispatch.py --debug --port=/dev/ttyAMA0
```

- Run the drill control GUI

```
python3 drill-control/drill-control.py
```

Drill orientation

The BNO055 is a System in Package (SiP), integrating a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope with a range of ±2000 degrees per second, a triaxial geomagnetic sensor and a 32-bit cortex M0+ microcontroller running Bosch Sensortec sensor fusion software, in a single package.



Calibrating the BNO055

Though the sensor fusion software runs the calibration algorithm of all the three sensors (accelerometer, gyroscope and magnetometer) in the background to remove the offsets, some preliminary steps had to be ensured for this automatic calibration to take place. The accelerometer and the gyroscope are relatively less susceptible to external disturbances, as a result of which the offset is negligible. Whereas the magnetometer is susceptible to external magnetic field and therefore to ensure proper heading accuracy, the calibration steps described below have to be taken. Depending on the sensors been selected, the following simple steps had to be taken after every 'Power on Reset' for proper calibration of the device.

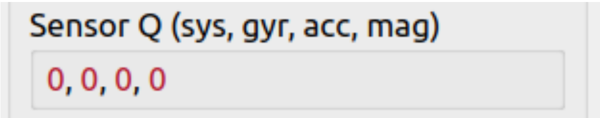
Routine recommended in datasheet

- Place the device in 6 different stable positions for a period of few seconds to allow the accelerometer to calibrate.
  - Make sure that there is slow movement between 2 stable positions.
  - The 6 stable positions could be in any direction, but it is best if the device is lying at least once perpendicular to the x, y and z axis.
- Place the device in a single stable position for a period of few seconds to allow the gyroscope to calibrate.

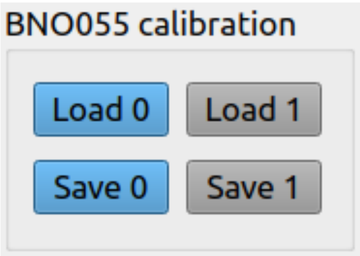
Routine for drill deployment

At EastGRIP, the following routine was found to be sufficient:

- Power on drill while laying horizontal tower and let it rest for 10 seconds.
- Rotate drill slowly around own axis at 90 deg. intervals and let it rest for a few seconds.
- Tilt tower to 45 deg. and repeat rotation.
- Tilt tower to 90 deg. and repeat rotation.
- If sensor quality values (`Sensor Q` in drill control GUI) are larger than 1 for most sensors, the calibration profile is useful (`Q>=1` is preferred).



- **If calibration is poor, return tower to horizontal and try again.**
- Once satisfied, save the calibration profile in slot `1` or `2` so that it can be reloaded later in case the BNO055 automatically re-calibrates (happens once in a while):



- Descend the drill with power on throughout descent.

If you want consistent heading (azimuth) and roll information from the drill, e.g. when using the spring for directional drilling, an additional step is needed:

- Make sure the spring (or some other marker on the drill) is pointing away from the tower (i.e. pointing straight upwards when tower is horizontal) and bring the tower to 10 to 30 deg. from plumb and press `Zero ref.`. This will align the drill orientation with the tower/trench frame-of-reference (the azimuth and roll dials should then align with zero).

Drill logs

Drill logs are saved on the inserted USB pen in `/mnt/logs`.

If a USB pen is NOT inserted, drill logs will NOT be saved!

Processing

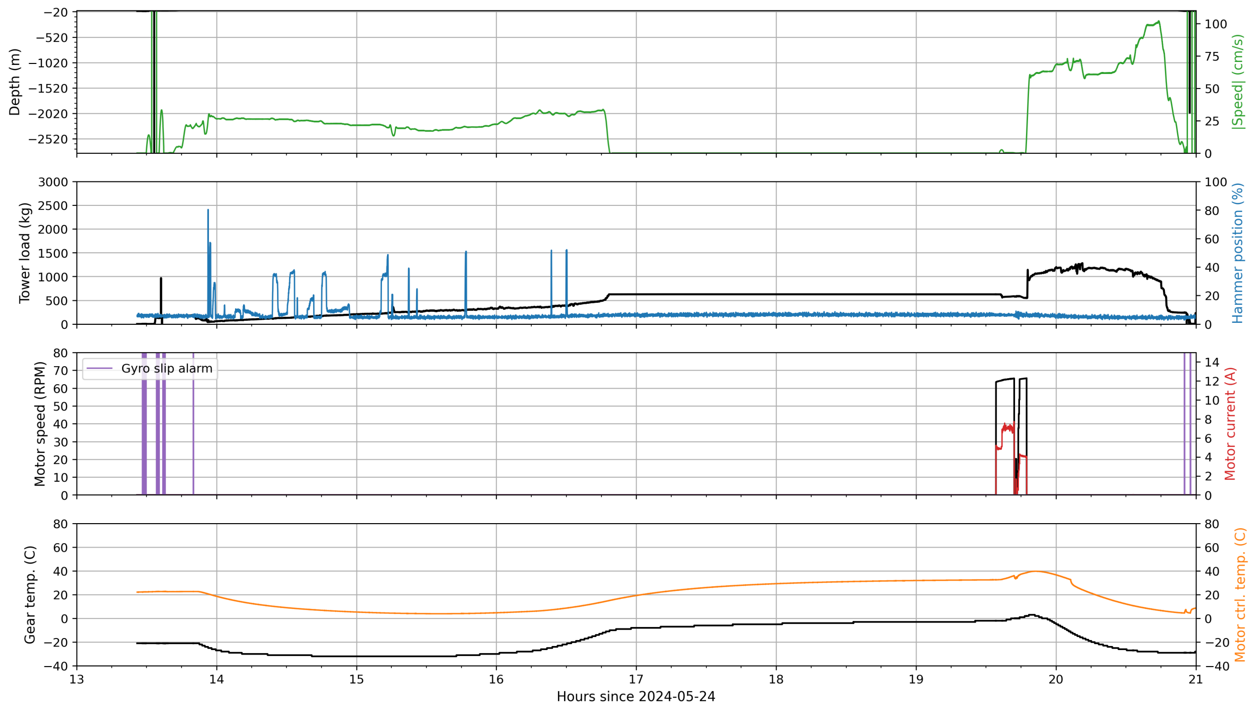
The log files must be processed to get the equivelent comma-separated-value (.csv) file type. This should be done externally and is not possible on the Surface Unit.

The following steps demonstrate how to do this:

- Power down the surface unit and remove the USB pen.
- Insert the USB pen into a Linux computer where the processing will take place.
- Download the surface unit software package from here and extract the .zip file.
- Move the log files from the USB pen into the `logging/drill-logs` subdirectory.
- Process a given log file by entering the `logging/` subdirectory and running (where 8 and 24 are the hours during that day to consider, but can be replaced by other hours)

```
python3 plot-drill-log.py drill-logs/drill.log.YYYY-MM-DD 8 24
```

- The processed .csv log file is dumped in `logging/drill-logs-processed` alongside a summary plot:



Calculating orientation profile

Once the drill log has been processed, the drill orientation can be calculated using standard AHRS algorithms that rely on the measured vectors of acceleration, magnetic field, gyro, etc.

The surface unit software supports only the SAAM algorithm which gives results close to estimates based on the tilt of the acceleration vector, but is more accurate as it includes the magnetic field. While these methods give the full orientation quarternion, only inclination and heading (azimuth) angles are outputted.

To generate a .csv file of drill inclination and heading, run the following command for a given processed drill log file (note that 12 and 17 is the interval of hours during that day to consider, but can be replaced by other hours)

```
python3 plot-drill-orientation.py \
    drill-logs-processed/drill.log.processed.YYYY-DD-HH.csv 12 17
```

The processed .csv file is dumped in `logging/drill-logs-processed` alongside an orientation profile plot: