NOMADe Manual

Android app, DCU and MATLAB

Contents

Android app and DCU

1 Hardware components

All the components needed to execute a measurement are listed below and are shown in the following picture.

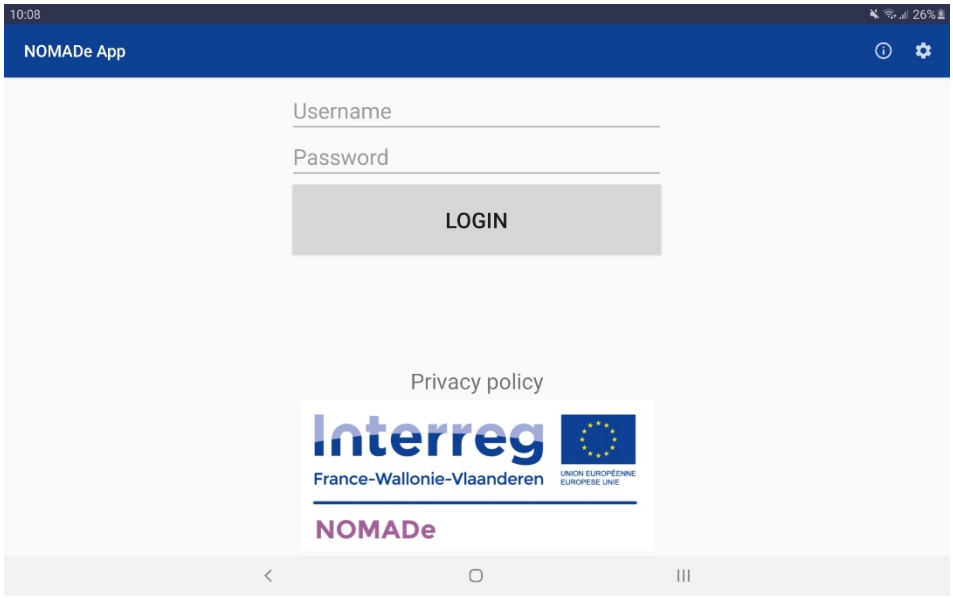
- DCU board
- 1 x IMU sensor module
- 2 x micro USB cable
- Micro-SD card
- Android device with NOMADe app installed
- Computer or power supply

2 Software components

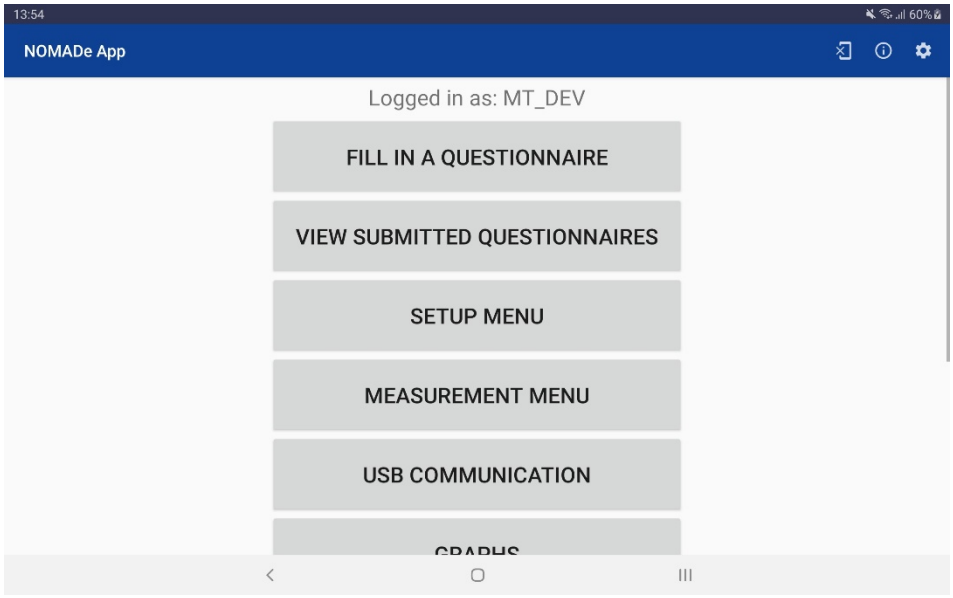
For these measurements you will need to install the NOMADe app. The app can be downloaded from Google Play by searching for 'NOMADe app' or scanning the QR code. After downloading the app, follow the instructions below.

3 Logging in

Underneath is the main screen of the app. You can use the username and password supplied to you by the NOMADe team to log in.



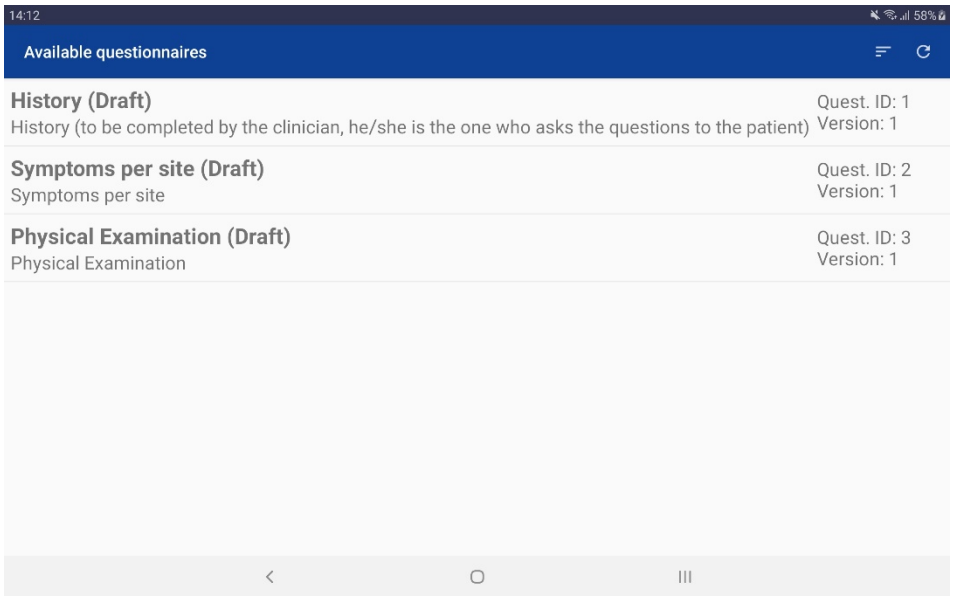
Once you are logged in, you should see the main menu. The available menu options may differ based on your account.



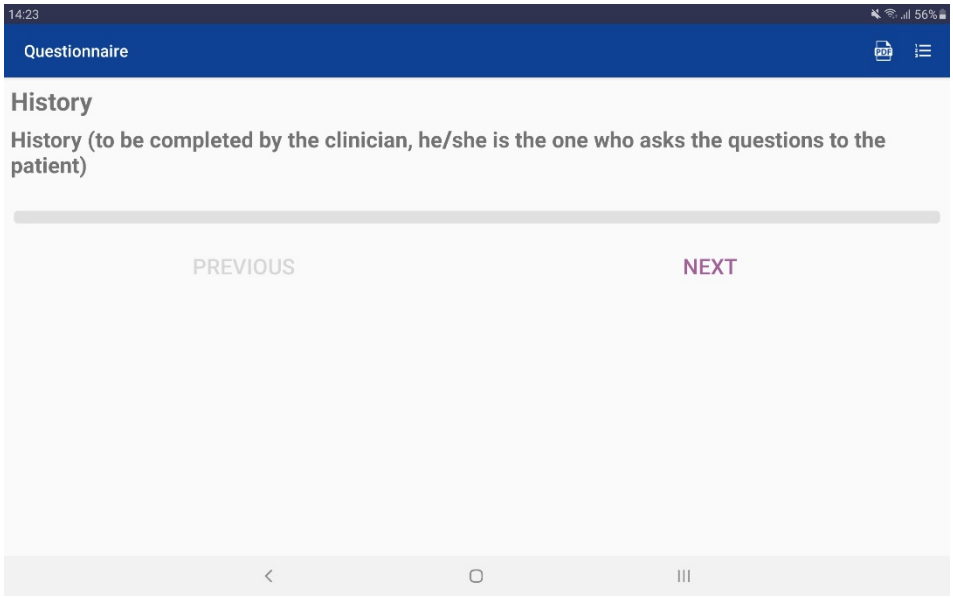
4 Questionnaires

4.1 Filling in a questionnaire

You can fill in questionnaires by clicking the corresponding button in the main menu. Now you can see a list of available questionnaires[1]. To fill in a questionnaire, simply click on one of the options.



You should see the intro screen of the questionnaire now[2]. While filling in a questionnaire you can navigate back and forward using the “Previous” and “Next” buttons at the bottom of the screen. Once you reach the end of the questionnaire, you get an overview of your answers.



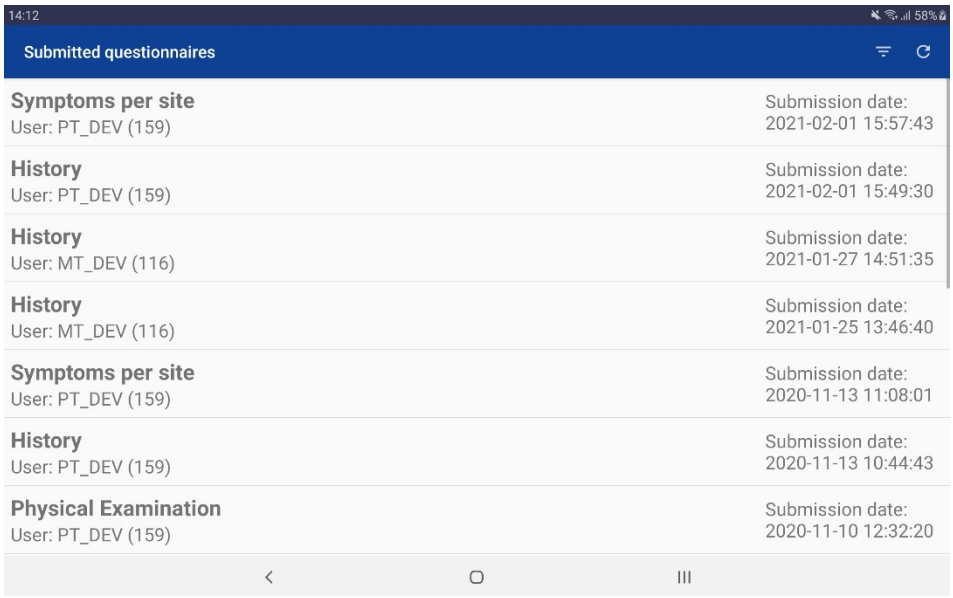
[1] If your account has the right permissions, you can see draft questionnaires by going to settings (cog wheel in the top right corner of the main menu) and enabling the “Display drafts” option.

[2] If your account has the right permissions, you will get a popup window asking whether you are filling in this questionnaire for yourself or for someone else.

4.2 Viewing a submitted questionnaire

You can view submitted questionnaires by clicking the corresponding button in the main menu.

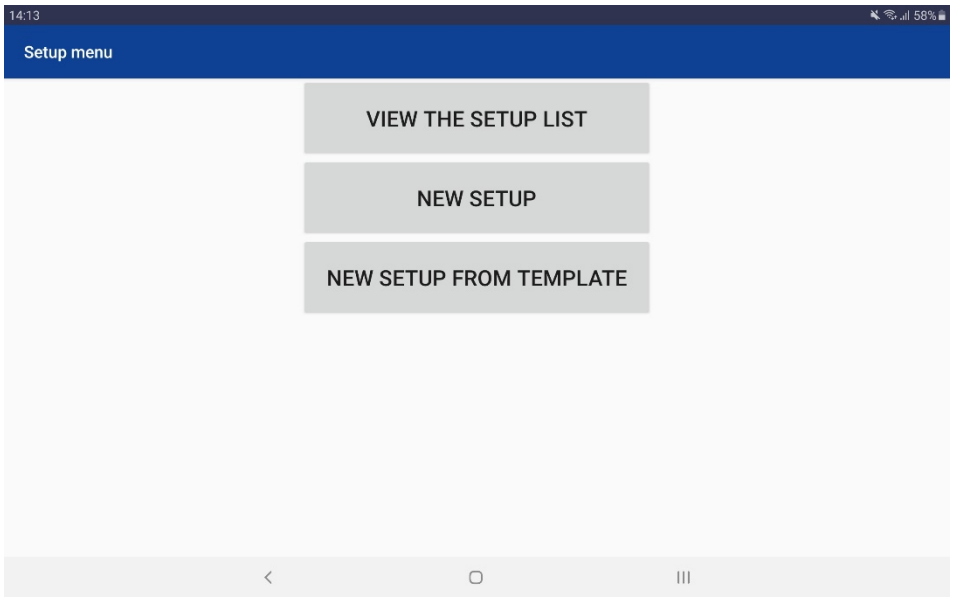
Now you can see a list of your submitted questionnaires[3].



5 Setups and measurements

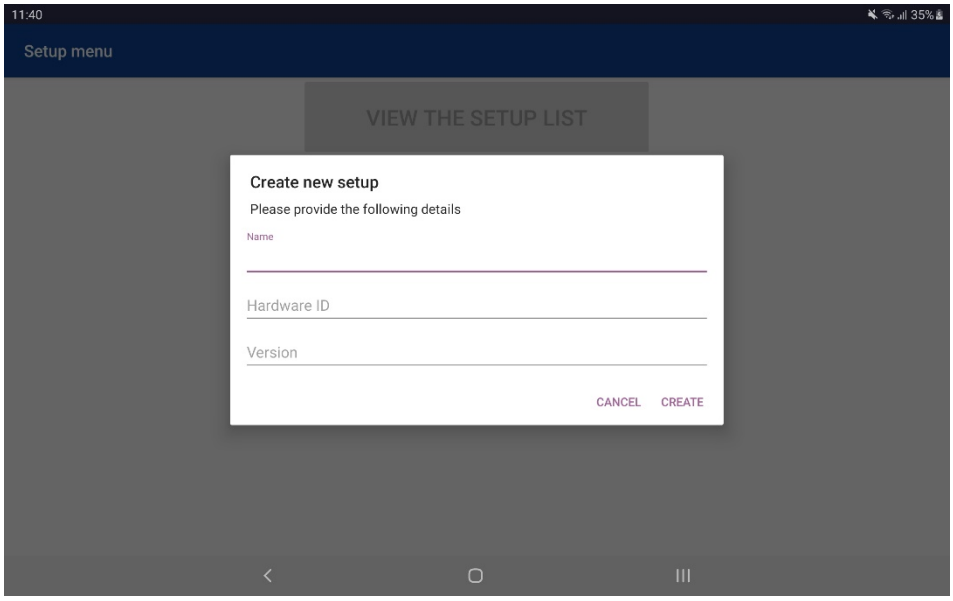
5.1 Creating a setup

To create a new setup, you first need to go to the "Setup menu", this is available by clicking the corresponding button in the main menu. In this menu you can view a list of the existing setups, you can create a new empty setup or you can create a new setup starting from an existing one.

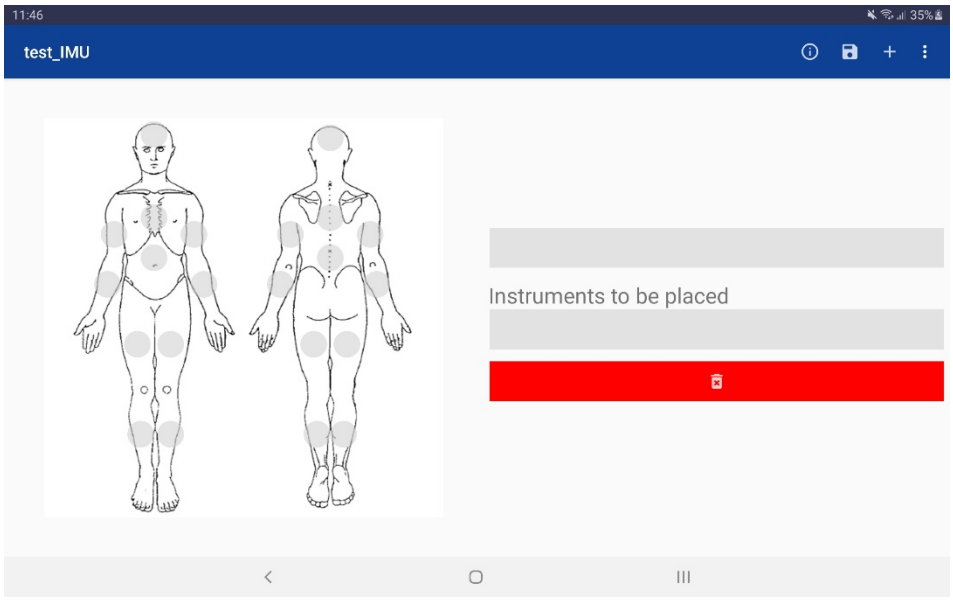


[3] If your account has the right permissions, you may also see submitted questionnaires of other people.

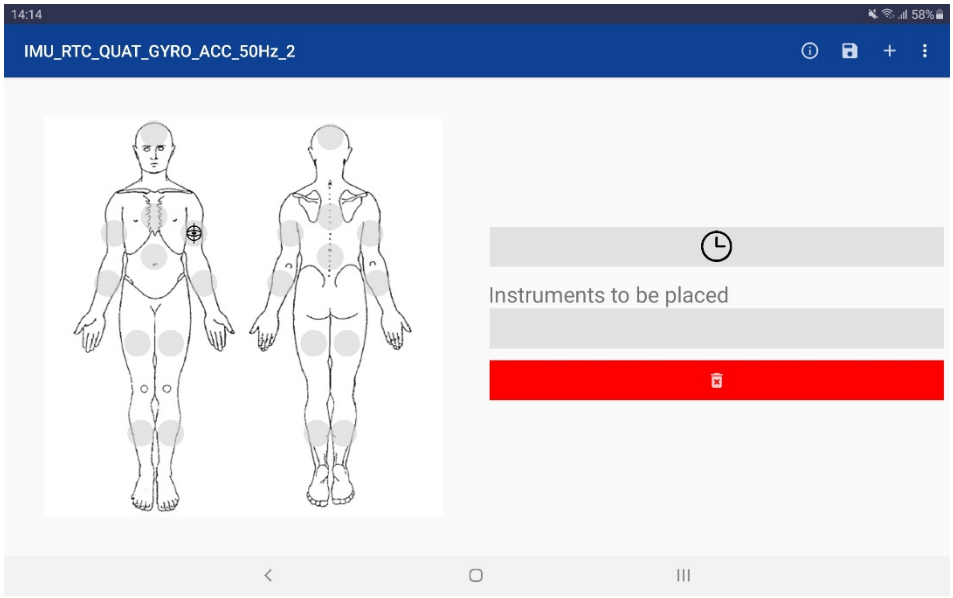
When you create a new setup (from template), you need to provide a name for the setup, a hardware ID and a version number.



A new empty setup should look like the screen underneath. You can start adding instruments (e.g. IMU sensors) using the "+" button in the top right corner. For the moment the "Real Time Clock" and the "Wireless IMU" are the only relevant instruments. The "Real Time Clock" will be added to the top grey box and the "Wireless IMU" will be added to the bottom grey box, this means that the "Real Time Clock" doesn't need a specific place on the body while the "Wireless IMU" does need a place on the body. To drag the "Wireless IMU", simply long press on the IMU symbol and drag it to one of the locations on the body indicated by the grey circles, the grey circle will turn green and will increase in size if you are in the right location, if you let go of the instrument it will be placed on the location.

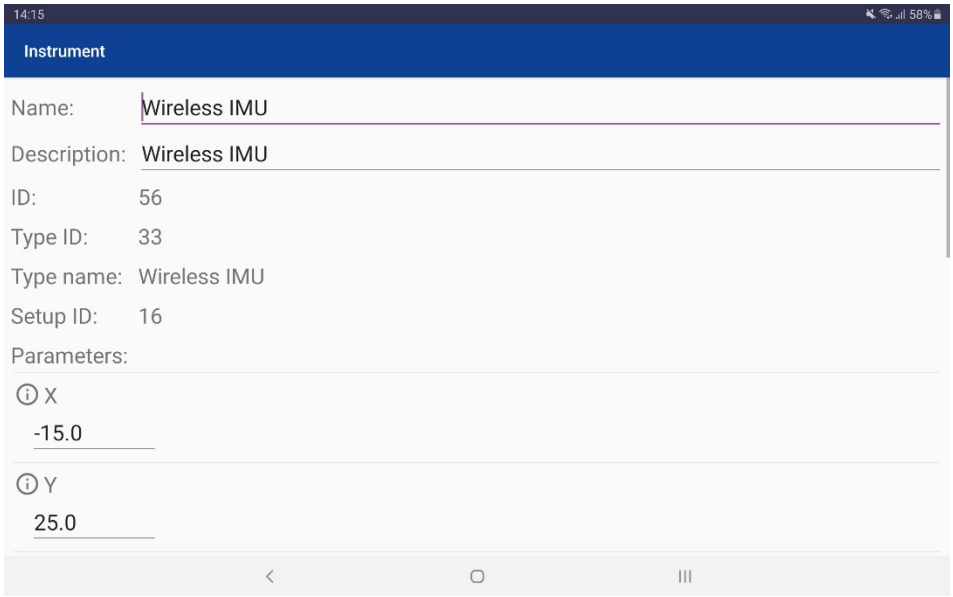


For the moment only one “Wireless IMU” is supported, so you should only add one “Wireless IMU” to each setup. Each setup only needs one “Real Time Clock”, so you should only add one “Real Time Clock”. Your setup should look similar to the screen below.



When an instrument is added to the setup, all parameters of that instrument have their default value. So don’t forget to change the needed parameters while configuring a setup. You can view and adjust the parameters by clicking the instrument you want to edit. For the RTC, you shouldn’t change any parameters.

For the “Wireless IMU”, you should change the Bluetooth MAC address so it corresponds to the MAC address of the IMU you are using, you could also change the sample rate and output datatype (“IMU_QUAT”, “IMU_QUAT_GYRO_ACC” or “IMU_QUAT_GYRO_ACC_100Hz”) if needed. Both 50 Hz and 100 Hz sample rates are supported. The output datatype depends on the chosen sample rate. You can use “IMU_QUAT” and “IMU_QUAT_GYRO_ACC” with a 50 Hz sample rate and “IMU_QUAT_GYRO_ACC_100Hz” with a 100 Hz sample rate. As the name of the output datatype suggest, “IMU_QUAT” will only collect the quaternions data, while “IMU_QUAT_GYRO_ACC” and “IMU_QUAT_GYRO_ACC_100Hz” will collect the quaternions data, gyroscope data and accelerometer data. Parameters “X” and “Y” are automatically changed when you drag and drop the instrument on the body, so you shouldn’t change those parameters in the edit screen.

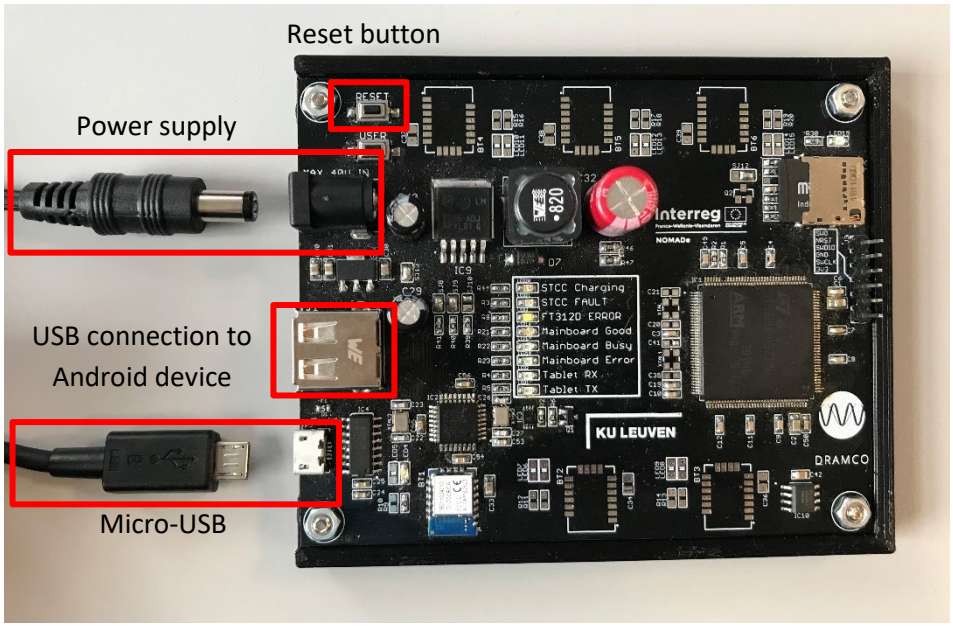


You can save your setup at any time using the “Save” icon in the top right corner. When leaving the setup screen, the app will ask if you want to save any unsaved changes if there are any.

5.2 Connecting the Android device to the DCU

First you need to power the DCU board before you connect the DCU to the Android device. Before powering the DCU board, insert the SD card into the DCU. In this case the SD card will only be used to store the setup and not to store the data from a measurement.

Connect the DCU board to a computer using a micro-USB cable to power the DCU board. Another method to power the DCU board is using a power supply. The correct connections for both methods are shown in the figure below. On the DCU board a red LED will light up. Next to this LED the message ‘Mainboard Error’ is displayed. This means that the DCU board is powered and that there is no IMU sensor module connected yet.

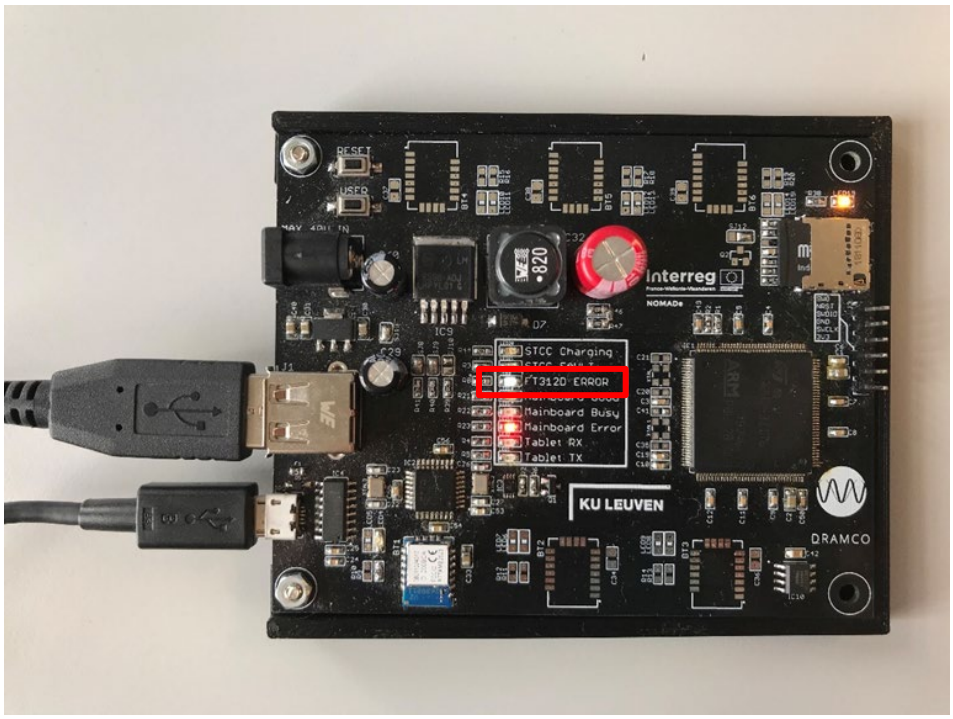


Once the DCU board is powered, the tablet can be connected to the DCU board. When connecting the Android device to the DCU, make sure that the NOMADe app is closed, this makes sure no issues occur

during the connection process. The easiest way to make sure the app is closed, is by opening your recent apps screen and closing the NOMADe app (depends on manufacturer).

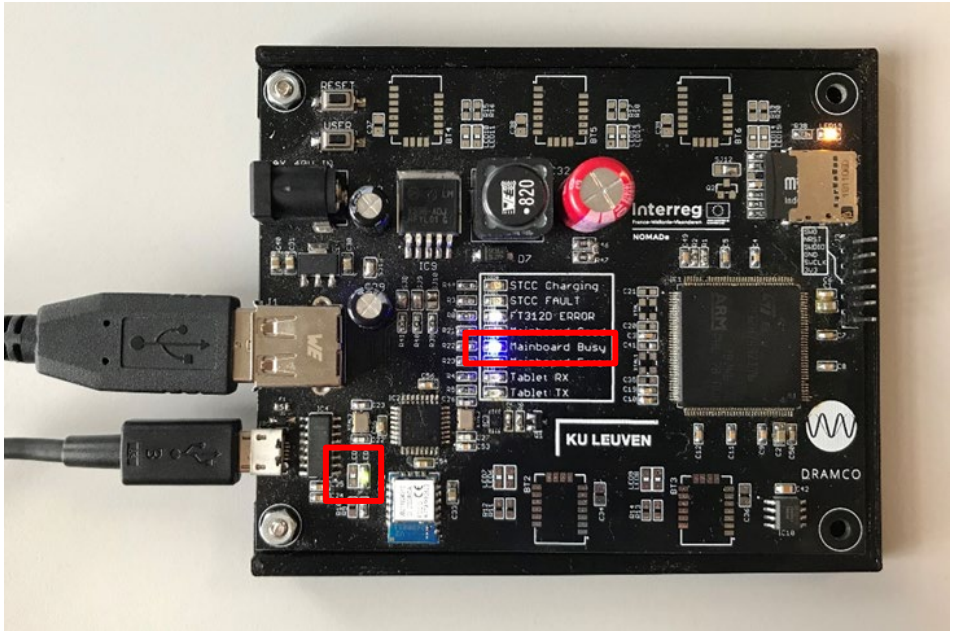
When the app is closed, you can connect the USB cable to the DCU as shown in the above picture. Always press the reset button on the DCU board after connecting the USB cable. After a few moments the app should open automatically. The first time it is possible that you need to give the app permission to open and/or that you need to select it as default.

When the communication between the DCU board and the tablet is established, a white LED will light up on the DCU board as shown in the next picture. The message 'FT312D ERROR' is displayed on the DCU board next to this LED. If this LED does not light up or the light goes off, disconnect the Android device from the DCU and repeat the process.

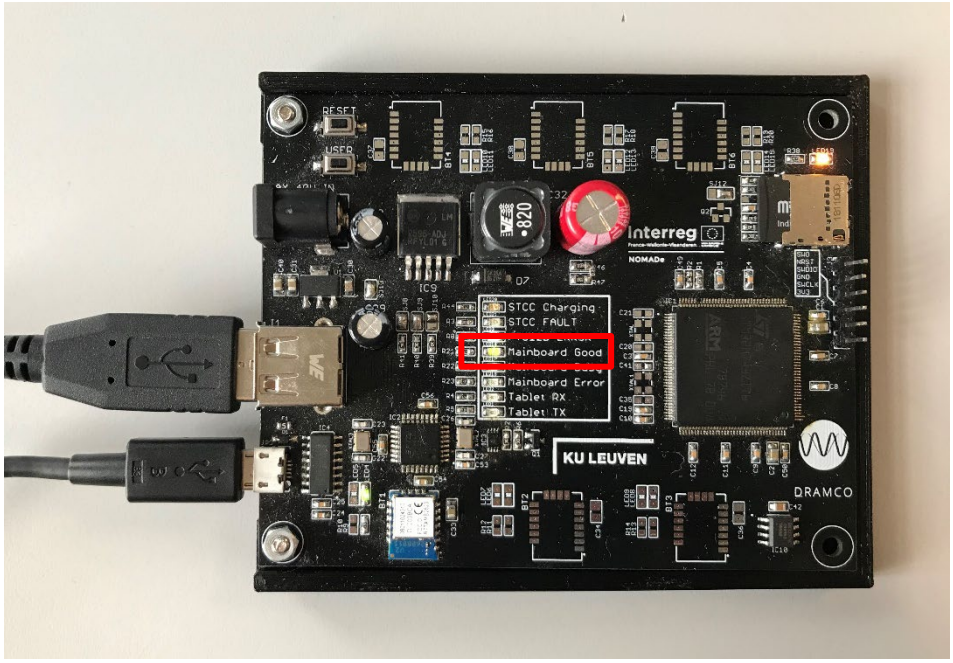


Once the NOMADe app has started, you are ready to add an IMU sensor module. Leave the IMU sensor module on the table and don't place it on the patient's body yet. Turn on the IMU sensor module. This module will automatically connect with the DCU board if you have entered the correct MAC address. If the green LED next to the Bluetooth module lights up the IMU sensor module is connected to the DCU board. The blue LED on the IMU sensor module will start blinking.

The IMU sensor module will automatically calibrate, so leave it on the table during this process. During this calibration process a blue LED on the DCU board will light up. Next to this LED the message 'Mainboard Busy' is displayed. On the IMU sensor module the blue LED will start blinking faster.



When the blue LED on the DCU board goes off and the green LED (with message ‘Mainboard Good’ lights up. This means that the calibration process has completed and the IMU sensor module is ready to start a measurement. The sensor module can now be placed on the patient’s body.



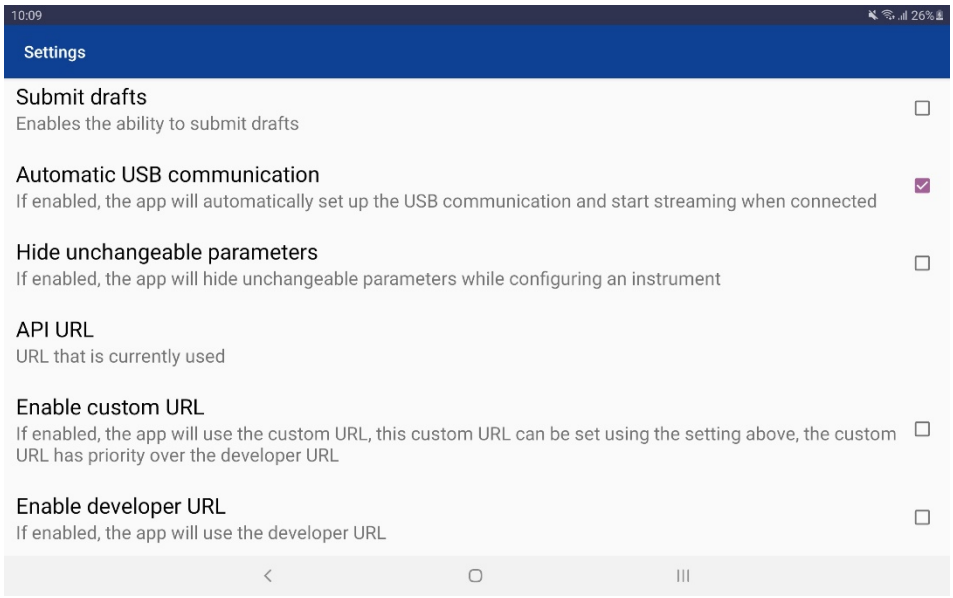
Disconnecting can be done at any time, but preferably not when a measurement is active. To reconnect, you again have to close the app first using the recent apps screen on your Android device.

5.2.1 Manual initialization

Once the app is open, you can go to “USB communication”, firstly click on “Initialise communication”, the DCU should assign an address to the Android device and it will also send information about the setup that’s currently loaded (or not loaded) in the DCU. The DCU will send watchdog messages continuously to maintain the communication link.

5.2.2 Automatic initialization

You can also enable “Automatic USB communication” in the settings. When this is enabled, the initialisation will be performed automatically in the background. This enables the user to directly go to the measurement menu, provided that there’s already a setup loaded in the DCU.

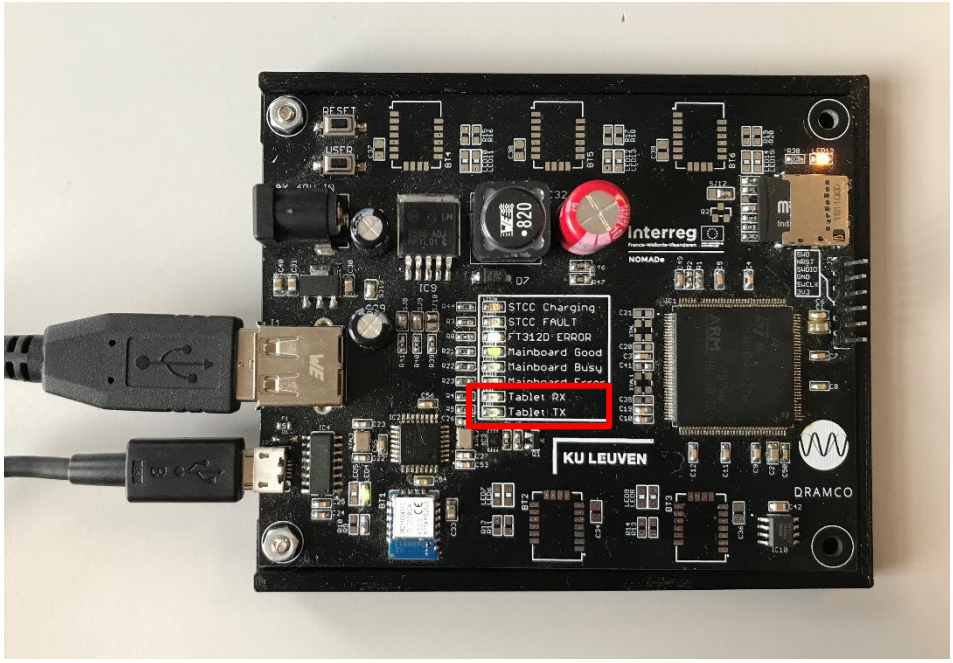


5.3 Sending the setup to the DCU

When the communication is initialised (watchdog messages are exchanged), you can send a setup to the DCU. This can be done via the “USB communication” screen by clicking on the “Send setup” button, then you can select a setup from the list. Once selected, the setup will be transferred to the DCU. At this point, reboot the DCU to make sure the setup is loaded correctly. Turn the IMU sensor module off and on again and repeat the process described above for connecting the IMU sensor module to the DCU.

5.4 Testing the setup

Once the setup is loaded in the DCU, you can test the setup by using the “Start stream” button in the “USB communication” screen, provided that the communication is initialised (watchdog messages are exchanged). The communication was initialised correctly if two blue LEDs (with messages ‘Tablet RX’ and ‘Tablet TX’ displayed next to them) will blink on the DCU as shown in the next picture. If these LEDs are not blinking, it means that there is no communication between the DCU and the tablet. If this is the case, reboot the DCU.

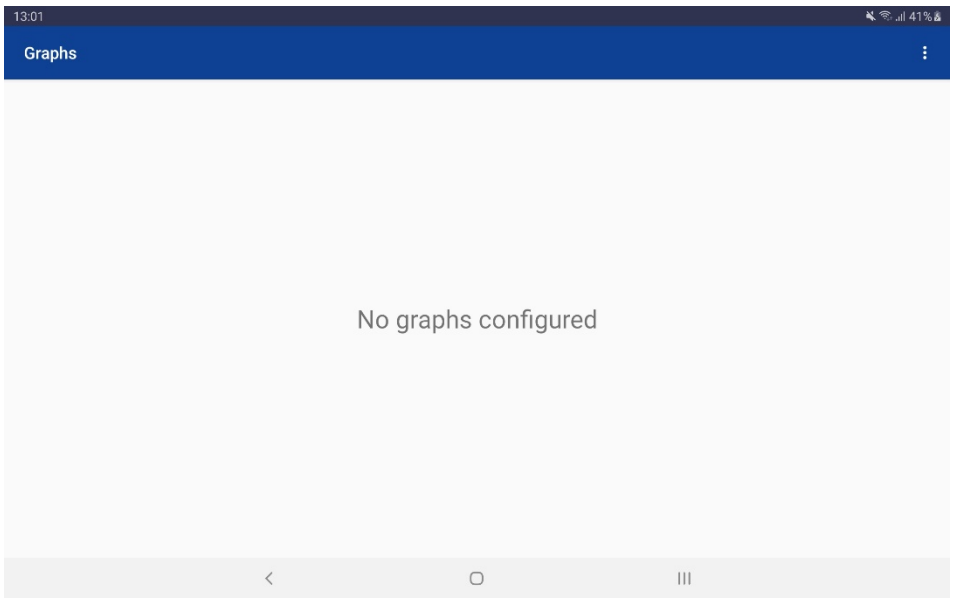


Once the stream has started, you can test the setup by going to the “Graphs” and/or “Values” screen.

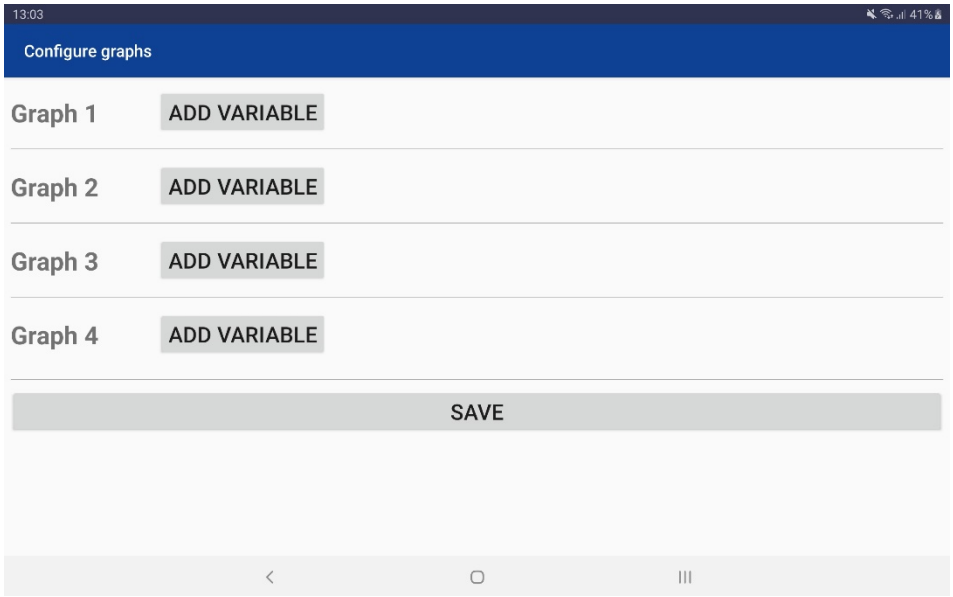
5.4.1 Graphs

From the main menu, you can navigate to the “Graphs” screen by clicking the corresponding button.

Initially, you should see an empty screen since you haven’t configured the graphs. To configure the graphs, click the three dots in the top right corner and select “Configure”.



In this screen you can configure 4 different graphs to display multiple variables. To add a variable, simply click “Add variable” and then select the instrument, the variable, the name and the color to display in the graph. Click “Add” to add the variable to the graph. Once added, you can edit or remove the variable by clicking on the variable itself. Once you are ready, click the “Save” button at the bottom to save and go back to the “Graphs” screen.



Once you're back at the "Graphs" screen, the configured graphs appear and the live data is displayed for the variables you have configured.

5.4.2 Values

A second option to display the data is the "Values" screen, you can navigate to the it by clicking the corresponding button in the main menu. This screen will display a live view of all the instruments with all their variables.

Via the cog wheel in the top right, you can set an update frequency for the values (1-50 Hz, default 50 Hz).

5.5 Locking the setup

Once the setup is tested and everything is working, you should lock the setup. A locked setup enables you to do measurements where the data is streamed in real-time to the cloud database for further analysis.
Note: if the setup needs some modifications, first make the needed modifications, send it again to the DCU (using the steps mentioned above) and test it again. Once a setup is locked, it can't be unlocked so make sure everything is functioning as intended before locking it.

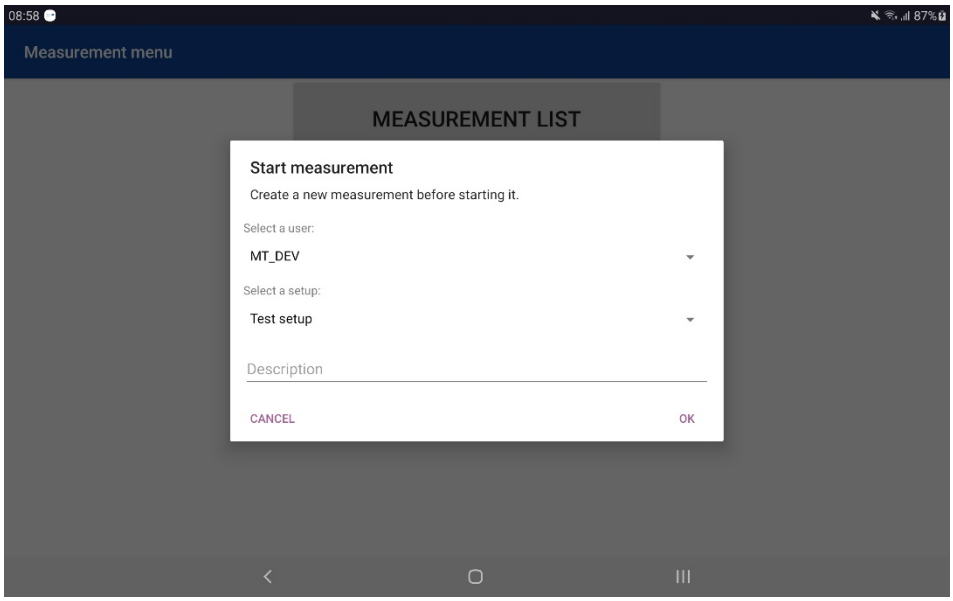
To lock a setup, go to the setup screen, click the three dots in the top right corner and click "Lock setup". If you didn't make any changes, you don't have to resend the setup to the DCU, otherwise make sure to resend the setup.

5.6 Starting a measurement

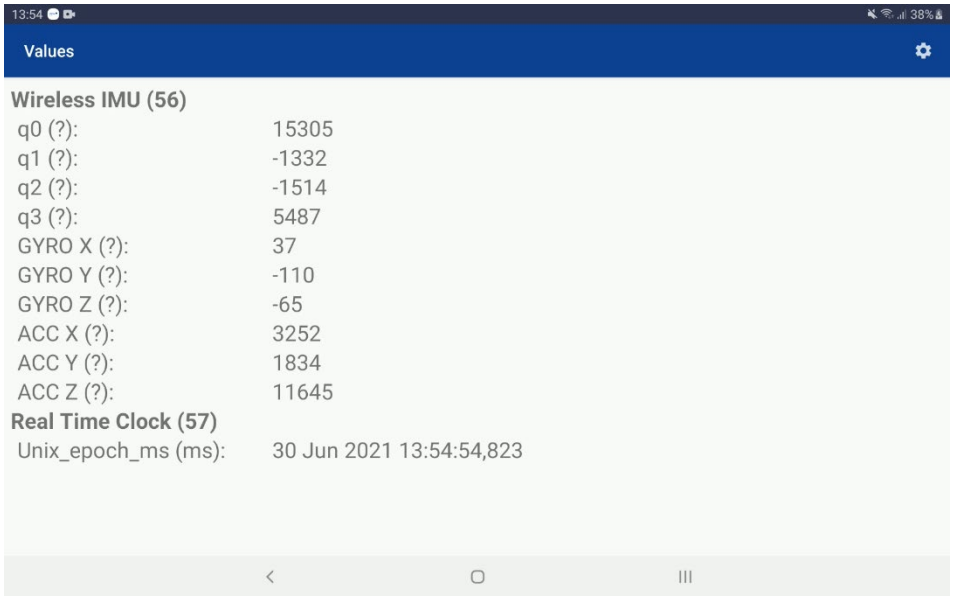
When the setup is locked, you can start doing measurements, in the app it looks similar to the testing of the setup using the "Start stream" button. The only difference is that the data is streamed to the cloud database in real-time.

Note: earlier tests have proven that the streaming of the data to the cloud database is blocked on most hospital/university Wi-Fi networks. The app can't detect whether or not the data goes through, so it's recommended to first do a short test measurement to check whether or not the measured data arrives in the cloud database. Using 4G mobile data is one of the workarounds to avoid issues concerning the connectivity.

Starting a measurement can be done while the Android device and the DCU are communicating (watchdog messages are exchanged), either from the "USB communication" screen or the "Measurement menu" screen using the "Start measurement" button. Clicking the button opens a popup window where you need to fill in the user, the setup and name of the measurement. The user and setup should be filled in automatically.



Once started, the data can be monitored in the “Graphs” or “Values” screen. The data is transferred to the cloud database in the background.

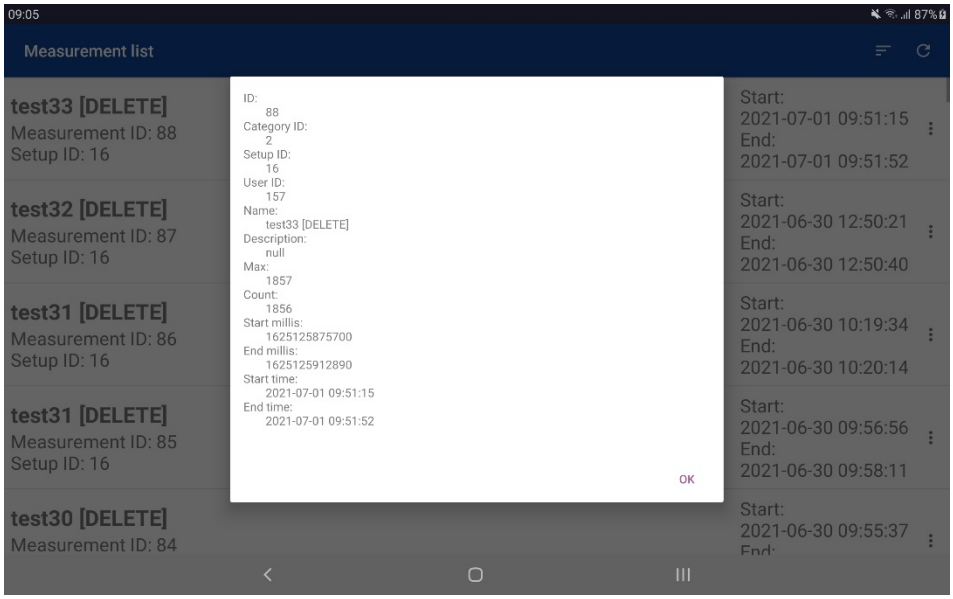


Stopping a measurement can be done using the “Stop measurement” button that can be found on the “USB communication” screen or the “Measurement menu” screen.

For the moment each new measurement gets an “[UNKOWN]” tag added to the end of the name. If you stop the measurement, a popup window will appear whether or not you want to keep the measurement.
If you choose to keep it, the “[UNKOWN]” tag will be removed from the measurement name. If you choose not to keep the measurement, the “[UNKOWN]” tag will be replaced with “[DELETE]”. In this way we can easily identify which measurements we need to keep, which measurements we need to delete and which measurements were not stopped correctly (“[UNKOWN]” tag will remain).

You can check the details about your recent measurements using the measurement list, accessible via the corresponding button on the main menu. You can view the details by clicking on the measurement.

The most important details in the popup window are “Count” and “Max”. “Count” contains the amount of cycle counters (with data) in the database. “Max” contains the highest cycle counter in the database. The difference between these two are the missing cycle counters for this measurement. If something went wrong with the measurement, e.g. the data didn’t reach the database due to network constraints, these two values will be “null”, this indicates that there is no data in the database.



MATLAB Live Script

1 Needed software & Toolboxes

1. Git
2. MATLAB R2019a or higher
   The following toolboxes are needed:

   a. Database_toolbox
   b. Pred_maintenance_toolbox

2 Installing the Git repository

Download, if not yet installed, Git on the following website https://git-scm.com/downloads. Install Git with the default settings.

When the error below occurs in MATLAB follow the steps below the error:

Error: 'git' is not recognized as an internal or external command, operable program or batch file.

Windows 10:

1. In the Start Menu or taskbar search, search for "environment variable".
2. Select "Edit the system environment variables".
3. Click the "Environment Variables" button at the bottom.
4. Double-click the "Path" entry under "System variables".
5. With the "New" button in the PATH editor, add C:\Program Files\Git\bin\ and C:\Program Files\Git\cmd\ to the end of the list.
6. Close and re-open your console.

Windows 7:

1. Right-click "Computer" on the Desktop or Start Menu.
2. Select "Properties".
3. On the very far left, click the "Advanced system settings" link.
4. Click the "Environment Variables" button at the bottom.
5. Double-click the "Path" entry under "System variables".
6. At the end of "Variable value", insert a ";" if there is not already one, and then C:\Program Files\Git\bin\;C:\Program Files\Git\cmd\. Do not put a space between ";" and the entry.
7. Close and re-open your console.

Source: https://stackoverflow.com/questions/4492979/git-is-not-recognized-as-an-internal-or-external-command

3 Pulling the repository NOMADe-Data-Visualization

Go to the wanted directory and execute the following command:

```
!git clone https://github.com/DimitriDeSchuyter/NOMADe-Data-Visualization.git
```

The repository is now cloned in the folder “NOMADe-Data-Visualization”. Next, download the JDBC drive which is needed for the database connection inside MATLAB. Unzip the file inside the new created repository folder “NOMADe-Data-Visualization”. Copy the content into the parent folder and rename this folder to “jdbc”. See Figure 1 for the structure after the JDBC driver is downloaded and copied. If this doesn’t match the “connect to the database” section will generate an error that the JDBC is not found.

https://downloads.mysql.com/archives/get/p/3/file/mysql-connector-java-8.0.18.zip

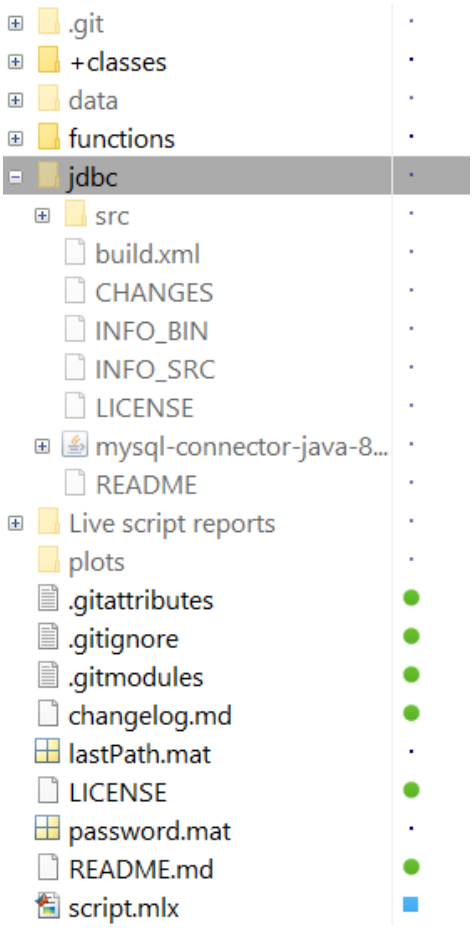


Figure 1: Folder structure

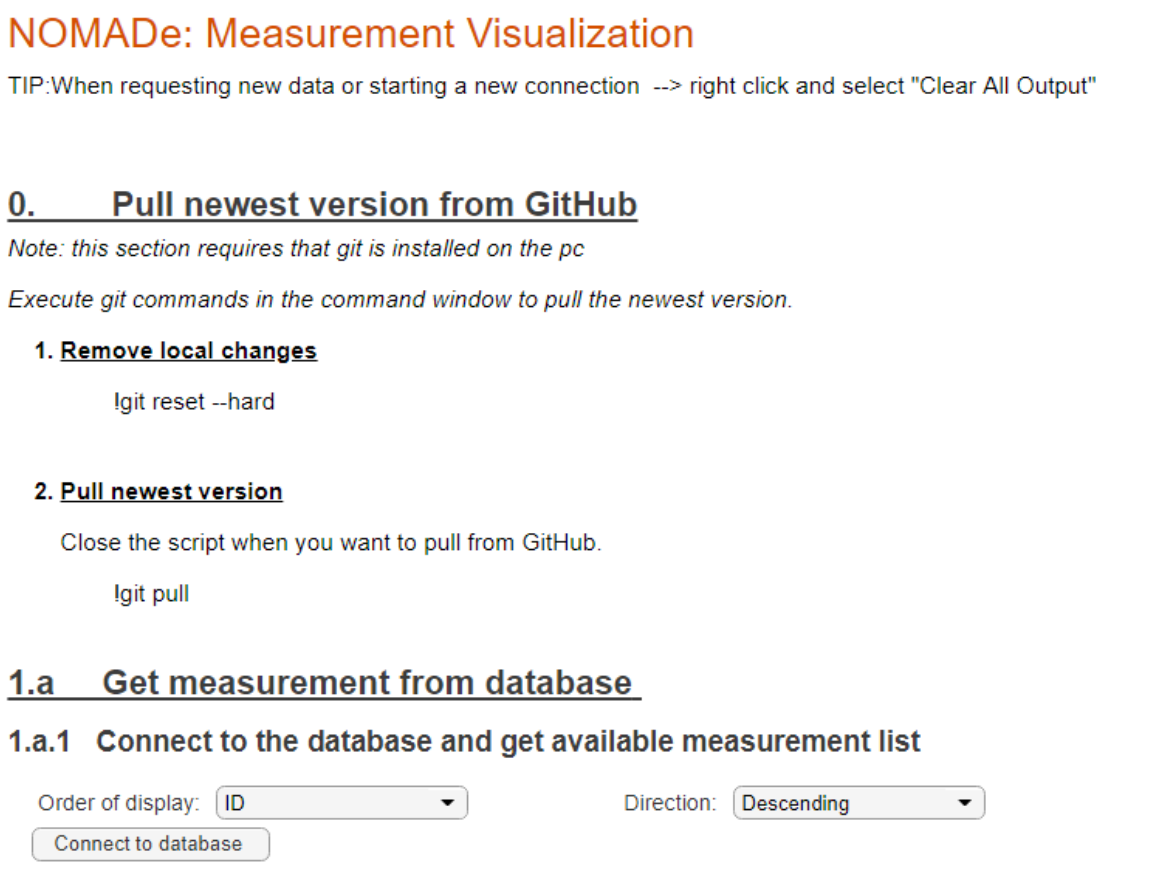
4 Running the Live Script

The code is segmented in a series of steps/functions. Those steps/functions are listed below:

0. Pull newest version from GitHub
1. .a Get measurement from database
    1. Connect to the database and get available measurement list
    2. Selection of measurement:

   .b Get measurement from Local .mat file
2. Functions
    1. Cutting in time: fit to start and end time
3. Plot all measurement information
4. Extraction from M object to workspace variables in data folder
5. Save m object in data folder
6. Save workspace to .mat file in data folder
7. Save live script (PDF, HTML, Word)

Each run of the script it is recommended to right click inside the Live Script and select “Clear All Output”.



4.1 Pull newest version from GitHub

This section shows the needed commands to get the newest version of the repository. When executing the last command, it is recommended to close all the scripts in MATLAB that are inside the repository.

4.2 Get measurement

There are two ways to get the measurement data and information. The first method is to pull it from the cloud database directly. The last method is that the measurement data and information are pulled from the cloud database earlier and stored to a measurement object inside a .mat file.

4.2.1 Get the measurement from the database

4.2.1.1 Section "1.a.1 Connect to the database and get available measurement list"

This will create a measurement object and establish a connection to the cloud database. For this connection a password must be inputted in the command window. There is an option to store this password to a local .mat file.

When the credentials are correct a table will be displayed with all the measurements that are stored in the cloud database. When another order is wanted the dropdowns can be used before pushing the "connect to database" button.

An example of this table is shown in Figure 2.

1.a.1 Connect to the database and get available measurement list

Order of display: ID Direction: Descending

Connect to database

```
NOMADe DB username: nomade-dev
NOMADe DB password (retrieved from file)
```

ans = 88×9 table

	id	name	description	started_at	stopped_at	setup	user	count	max
1	88	"test33 [DE...	"empty"	"2021-07-01 09:5...	"2021-07-01 09:5...	"IMU_RTC...	"SG_DEV"	1856	1857
2	87	"test32 [DE...	"empty"	"2021-06-30 12:5...	"2021-06-30 12:5...	"IMU_RTC...	"SG_DEV"	987	988
3	86	"test31 [DE...	"empty"	"2021-06-30 10:1...	"2021-06-30 10:2...	"IMU_RTC...	"SG_DEV"	2005	2007
4	85	"test31 [DE...	"empty"	"2021-06-30 09:5...	"2021-06-30 09:5...	"IMU_RTC...	"SG_DEV"	3751	3753
5	84	"test30 [DE...	"empty"	"2021-06-30 09:5...	"2021-06-30 09:5...	"IMU_100Hz"	"SG_DEV"	326	327
6	83	"test26 [DE...	"empty"	"2021-06-29 12:0...	"2021-06-29 12:0...	"IMU_100Hz"	"SG_DEV"	462	463
7	82	"test27 [DE...	"empty"	"2021-06-29 11:5...	"2021-06-29 11:5...	"IMU_100Hz"	"SG_DEV"	987	987
8	81	"test26"	"empty"	"2021-06-29 11:5...	"2021-06-29 11:5...	"IMU_100Hz"	"SG_DEV"	304	305
9	80	"test25 [DE...	"empty"	"2021-06-29 11:5...	"2021-06-29 11:5...	"IMU_100Hz"	"SG_DEV"	398	400

Figure 2: Measurement list.

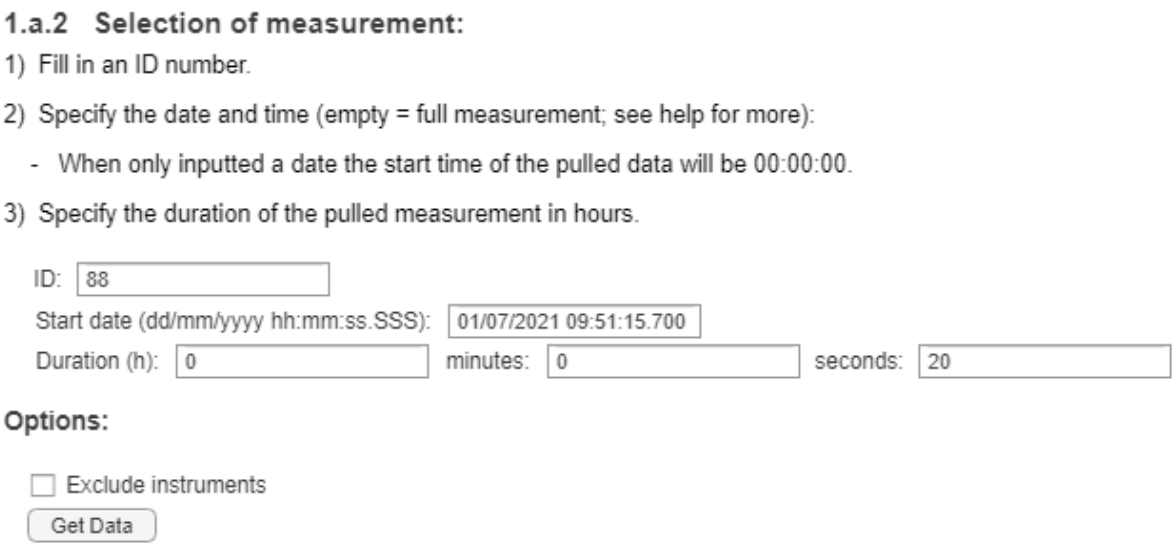
4.2.1.2 Section "1.a.2 Selection of measurement"

In this section the data from a specific measurement is pulled and processed. The measurement can be chosen with a unique measurement ID which is shown in the measurement list in 1.a.1.

The execution time of this section depends on the duration of the measurement. For this an option is added to pull a specific period of this measurement. When the full measurement is wanted put "full" in the start date field. Otherwise give the start date with same syntax as specified in the script. Also, the duration that must be pulled must be entered.

If only interested in some instruments a function is implemented to show a popup where the instruments can be excluded when "Exclude instruments" has been checked.

To execute this section the "Get Data" button must be pressed. The extracting and processing time per instrument will be shown and if there are missing cycle counters a warning will be shown at the end of the displayed times.



4.3 Functions

4.3.1 Cutting in time

If the start of the measurement recording is not the same as the start of the exercise the cut function can be used. A popup will be shown where the new start and end time can be chosen. When this function is executed the cycles outside the new measurement period are deleted in the measurement object. In case of a wrong time was inputted the data needs to be pulled again from the cloud database.

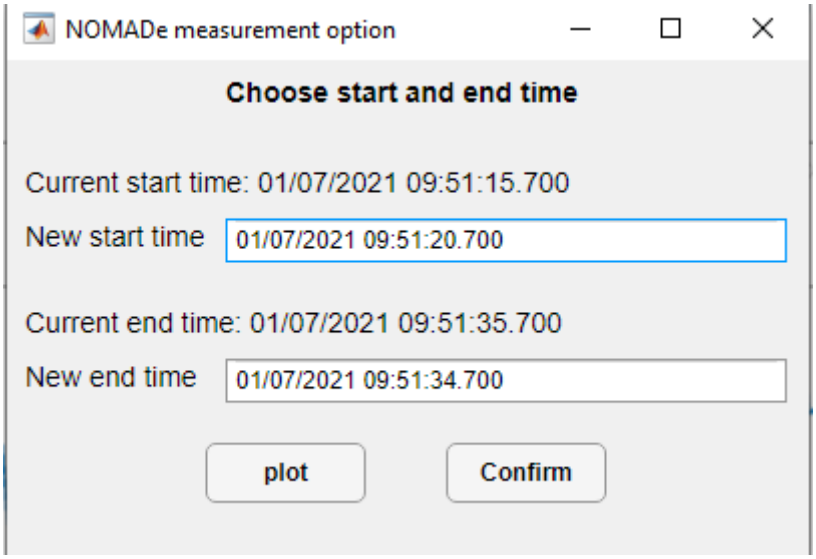


Figure 3: Cutting in time popup.

4.4 Plot all measurement information

In this section all the data from the instruments can be plotted automatically inside the Live Script. There is an option to choose which instruments that must be plotted.

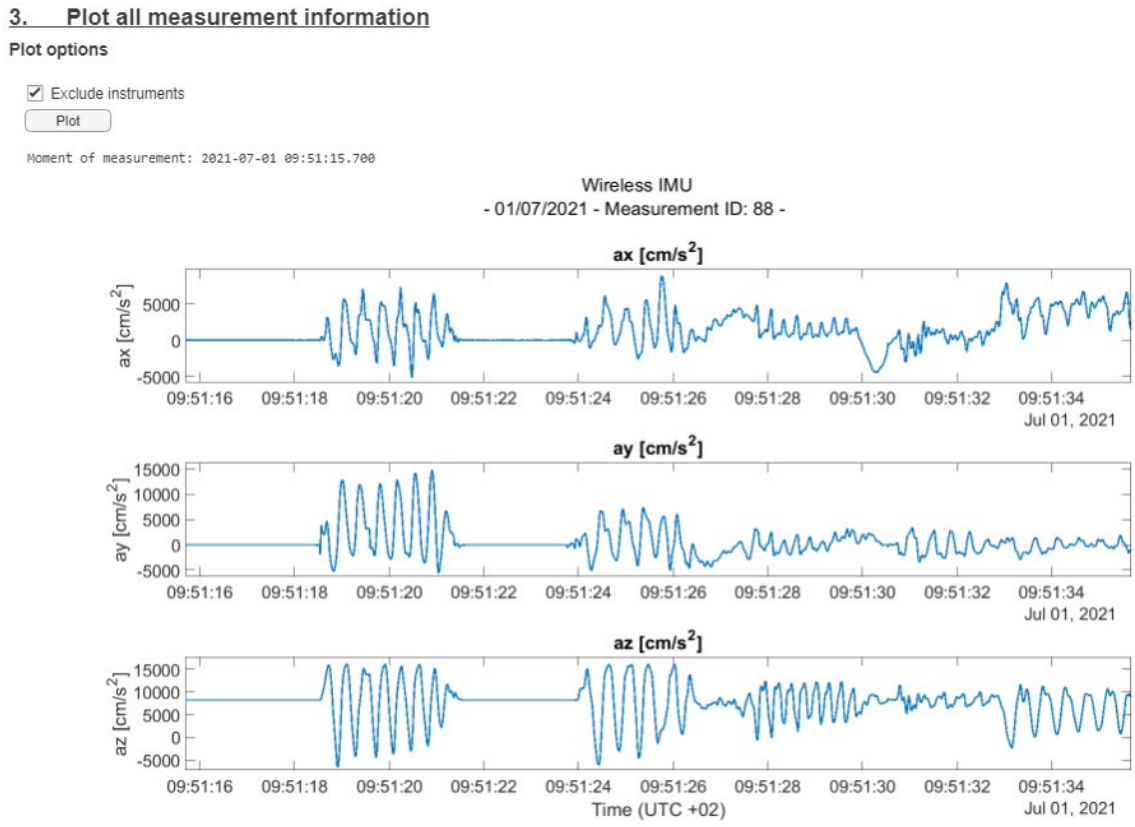


Figure 4:Plotting section.

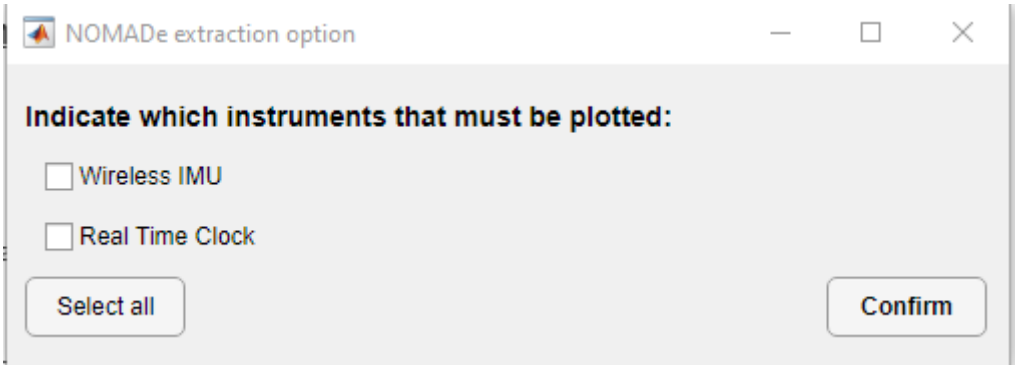


Figure 5:Plotting exclude instruments popup.

4.5 Extraction from M object to workspace variables in data folder

When own plots or analysis is wanted it can be useful to have all the data for each instrument in the workspace rather than a measurement object in the workspace. The same data will be available but without the structure of the measurement object. Inside an info variable is most of the important information of the measurement stored, e.g. the start and stop time.

4.6 Save m object in data folder

As mentioned before the measurement object can be stored to a .mat file. This comes of great use for long measurements. It reduces the extraction and processing time to a single execution which can take hours/ days. When the database toolbox is not installed/possessed it is also possible to use this .mat file in section 1.b. The .mat file will be stored in the data folder with a name structure as follows "IDXXX_ m-extracted_YYYY_MM_DD_hhmmss". Also, a custom name can be chosen.

4.7 Save workspace to .mat file in data folder

Saving all the workspace variables into a .mat file is possible. It will be stored in the same folder as the m object and with a name structure as follows:

"IDXXX_workspace_instruments_YYYY_MM_DD_hhmmss".

4.8 Save live script (PDF, HTML, Word)

As last step the live script with the executed steps will be stored. A PDF, Word and/or HTML version can be stored inside the "Live script reports" folder.