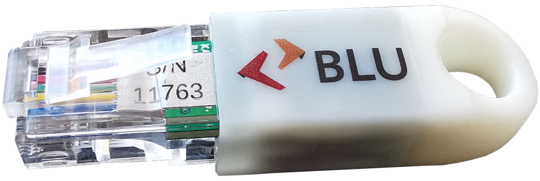
BLU & Kanja — Manual

July 2024



Revision 1.9

Contact Information

Publisher and producer:
Kanardia d.o.o.
Lopata 24a
SI-3000
Slovenia

Tel: +386 40 190 951
Email: info@kanardia.eu

A lot of useful and recent information can be also found on the Internet. See `http://www.kanardia.eu` for more details.

Copyright

Credits

This document was written using TeX Live (LaTeX) based document creation system using Kile running on Linux operating system. Most of the figures were drawn using Open Office Draw, Inkscape and QCad applications. Photos and scanned material was processed using Gimp. All document sources are freely available on request under the licence mentioned above and can be obtained by email. Please send requests to info@kanardia.eu.

Some icons used in this manual and in Horis are contributed to `https://www.flaticon.com`, in particular, by Freepik, Flat icons, Maxim Basinski, Prosymbols, Juicy Fish, Vectors Market, Maswan, Muhammad Usman.

WEEE Statement

Disposal of Waste Electrical and Electronic Equipment. This electrical item cannot be disposed of in normal waste. Check with your local authority for kerbside collection, or recycle them at a recycling centre.

Revision History

The following table shows the revision history of this document.

Rev.	Date	Description
1.9	July 2024	Installation & pairing reorganization
1.8	May 2024	Installation, minor fixes
1.7	Jan 2024	Pairing troubleshoot
1.6	December 2022	Recorder function
1.5	August 2021	Kanja App updated
1.4	July 2020	Instrument Screen Layout Changes updated
1.3	February 2020	Revised with LaTeX

Contents

1 Introduction

First of all we would like to thank you for choosing Kanardia.

Kanardia `BLU` is a CAN-to-Bluetooth interface plug which in combination with the `Kanja` Android application provides ability to upgrade and configure most CAN based Kanardia products. It is also well suited for various diagnostic.

When your system consist mostly from Indu devices, Horis or similar, it is essential component for the software update.

This manual covers `BLU` plug in combination with its `Kanja` Android app.

- `BLU` is the interface plug which plugs into Kanardia CAN bus network. It listens to the CAN bus traffic and transmits the traffic to the `Kanja` app via the Bluetooth connection.
- `Kanja` is an Android app used to manipulate Kanardia CAN bus devices. Most of this manual is devoted to Kanja.

1.1 System Requirements

To run `Kanja` application and control Kanardia devices with `BLU` interface plug you will need an Android tablet or telephone with:

- Android version 7 or higher,
- WiFi/4G/5G data connection to the Internet,
- Bluetooth v2.1 or later.

1.2 Kanja Meaning

In our language (Slovene) `Kanja` shown in Figure 1 is a name for a predator bird very common in Europe. See wiki `https://en.wikipedia.org/wiki/Common_buzzard` for more details.

2 Installation

In order to get `BLU` and `Kanja` combination running on your Android, you have to perform the following general steps.

Figure 1: Kanja (Common Buzzard) in flight. The photo is credit of Andreas Trepte, `http://avi-fauna.info/`.

- Hardware installation - insert `BLU` into CAN network and make sure it is properly terminated.
- Install the `Kanja` app into your Android device.
- Pair `BLU` and your Android device.

2.1 Hardware Installation

`BLU` dongle has to be connected to Kanardia CAN network. Normally the easiest way to do it is to connect it in a place where RJ45 CAN bus terminator plug is connected. Shortly after the power is applied to Kanardia devices (CAN network) the `BLU` will indicate initialization routine by blinking red and blue LEDs. Once the initialization is complete the blue LED will blink in a rate of one blink per second.

You have to assure that at least one CAN bus terminator is present on the CAN bus! If terminator is not present, the CAN bus will not be working properly. If your instrument is standalone, than put `BLU` into one CAN port and a terminator plug into the second CAN port.

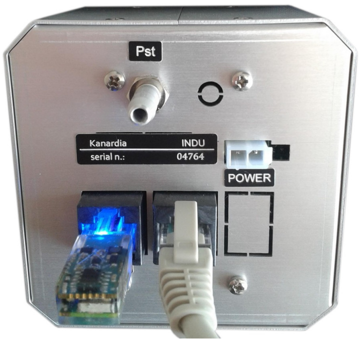


Figure 2: Blu inserted into an Indu device into one of the CAN ports.

2.2 App Installation

The correct source of the `Kanja` app is the Google Play Store. In some special cases, the app can be also obtained from our web site.

2.2.1 Installation From The Google Play Store

`Kanja` is installed directly from the Google Play Store. Follow next steps:

1. Open the `Google Play Store` app on your Android device.
2. Enter `Kanja` into the search field.
3. Locate the `Kanja` app and install it in a standard way. Figure 3 shows the app icon. A similar icons shall appear in the result list.

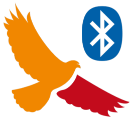


Figure 3: The `Kanja` app image in the Google Play Store.

If the installation fails, please check if `Kanja` was already installed. In this case, remove it from you Android and try again.

2.3 Installation From Our Web Site

Installation from our web site is not recommended. Use this only if `Kanja` is not available from the Google Play Store or if we explicitly instructed you to do so. Furthermore, the `Kanja` app will be hidden on our web site for most of the time and we will make it available only in special cases.

1. If you have `Kanja` already installed on your Android device, then we suggest that you *uninstall* if first. Especially, if it was initially installed from the Google Play Store.
   Note: On some Androids, you have to go into `Settings|Application` and uninstall the old version from there.

2. Download the `Kanja` app file from `https://www.kanardia.eu/support/firmware/` to your Android device.

3. The Kanja.apk file will be stored *somewhere* on the device. Try to use a file managing app on your device and locate the file. Again, this depends on Android version and apps in use.

4. Open/run the Kanja.apk file. Most probably how will have to configure your phone to allow/enable installation from third party source. As before, the actual procedure strongly varies with Android version.

2.4 Pairing

Before you can use `BLU` you have to pair it with your Android. Use your Android system settings for this. Usually this is done only once as Android will remember the paired devices. The exact details depend on your device and Android version. In principle, follow:

1. Plug the `BLU` device into a free CAN slot on Kanardia CAN network. See section 2.1.

2. Make sure the Kanardia CAN network is powered on and make sure that at least one CAN bus terminator is present.

3. Make sure that the BLU initializes properly (blue LED blinking in a rate of one blink per second).

4. Open `Settings | Bluetooth` on your Android and make sure that the Bluetooth module and its visibility to others is enabled.

5. Once the Bluetooth page is active, various devices start appearing on the `Available Devices` list. Search for the `ESP Kanardia BLU SN`, where SN stands for its serial number.
6. Tap on your `BLU` device and pair it. Confirm the pairing. No action on `BLU` is necessary.

Once `BLU` is paired with your Android, you may start using the `Kanja` app.

3 Kanja App

3.1 Access Grants

On the very first run, the system will ask you to allow Kanja some access. In reality Kanja only needs access to files, where the latest firmware, user layouts and configurations are stored. However, the system may ask you for more permissions, Location for example. Please allow access to them otherwise you will not be able to use `Blu`. It is beyond our knowledge why all these additional permissions are needed.

3.2 The Start Page

When Kanja starts, a page similar to Figure 4 appears. Figure 4 shows individual elements on the page.

① Kanja app version is shown on top and last data download date & time is shown below.

② Opens a new page with a list of all Kanardia devices detected on the CAN bus. You have to connect Kanja and BLU first.

③ The connection button. Click it to establish connection between Kanja and BLU. BLU must be paired first. The hexadecimal number on the button shows the address of the Bluetooth device it will connect to.

④ A click on the `Select BLU` button opens a list of detected BLU devices. In your case there will probably only one. Select it.

⑤ Status bar is briefly shown on the app start. Once the latest data has been downloaded, `Done` is displayed and shortly after the line disappears.

⑥ Kanja global menu allows access to certain special commands.

⑦ Manual BLU address entry. This is seldom used. You may use it in cases where selection process fails.

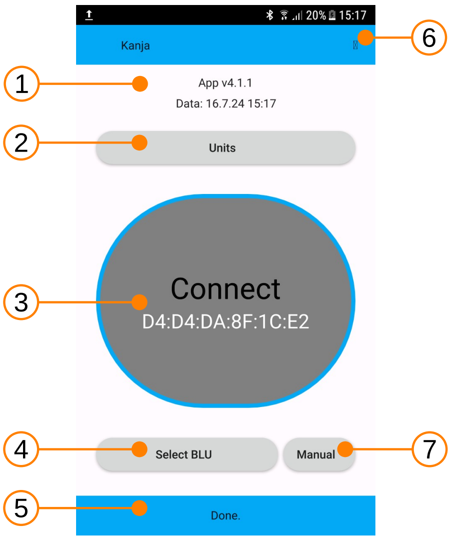


Figure 4: Typical page shown on Kanja start

3.3 First Connection

The `Kanja` app and `BLU` must be paired before first connection to the CAN network. See Section 2.4.

1. Plug `BLU` into a free CAN slot on Kanardia CAN network. See section 2.1.
2. Make sure the CAN network is powered on and make sure that at least one CAN bus terminator is present.
3. Make sure that `BLU` initializes properly (blue LED blinking in a rate of one blink per second).
4. Make sure that the Bluetooth mode is active on your Android.

5. Start `Kanja` app. The page similar to one shown on Figure 4 shall appear.
6. Click the `Select BLU` button and select your `BLU` from the list. In most cases, there will be only one item in the list. Select it.
7. Press the `Connect` button. Finally, page changes to a list of Kanardia devices detected on the CAN network.

3.4 Connection

If Kanja and BLU were already paired (see section 3.3), then simply:

1. Plug BLU into a free CAN slot and make sure you have at least one terminator in the network.
2. Make sure the CAN network is on. BLU shall be blinking.
3. Press the `Connect` button to establish connection between Kanja and BLU.

You may keep BLU plugged in permanently, if this suits you.

3.5 Devices Page

Once connection is established, Kanja shall switch to the device page. This page lists all devices found on the CAN bus. Device name, serial number and hardware number are shown together with their name. Figure 5 shows an example.

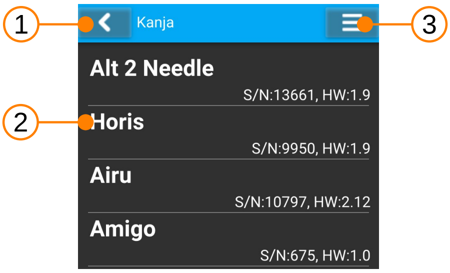
Please note that certain instruments may consists of several devices. Horis, for example, consist of the Horis core device and an AD-AHRS-GPS module called Airu.

The following actions are possible:

① Blue arrow closes this page and switches back to the previous one. Always use this command to get previous page. Do not use Android *back*.

② Select a device from the list to open more options for this device.

③ Access to special functions, which are not necessarily device specific.



Alt 2 Needle
S/N:13661, HW:1.9

Horis
S/N:9950, HW:1.9

Airu
S/N:10797, HW:2.12

Amigo
S/N:675, HW:1.0

Figure 5: An example of devices detected on the CAN bus network.

4 Standard Options

Once a device is selected a list of options appears in a new page. Many of these options are standard for all devices, Figure 6. They will be covered in this section.

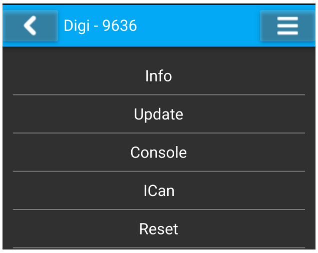


Figure 6: Standard options for all devices on the CAN network.

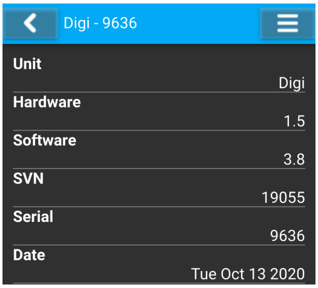
4.1 Info

The `Info` command, opens yet a new page, where some general device information is shown. Use blue back button, to switch to previous page. Figure 7 illustrates and example.

Unit shows the device name.

Hardware shows the device hardware version. This shall never change.

Software shows the software version as major.minor release. This changes with each update.



Unit
Digi
Hardware
1.5
Software
3.8
SVN
19055
Serial
9636
Date
Tue Oct 13 2020

Figure 7: Device general information page.

SVN is special software build number (subversion number), which exactly identifies software situation at the time software was created. This number increases in steps with every update.

Serial number of the device.

Date of the device creation.

4.2 Update

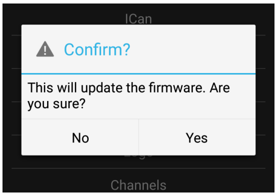
This command is used to transfer latest firmware (software) into a CAN device. Most devices can be updated this way, but not all. Nesis, Aetos and Emsis are exceptions – they must be updated using USB memory stick or micro SD card.

On every start, Kanja connects to our server and automatically downloads the latest firmware along with some other configuration files. The date and time of the last *data* is shown on the start page, see Figure 4, mark (1) below. The downloaded files are then kept on your Android device.

Selecting the `Update` option starts an update process. A confirmation window appears to prevent accidental updates. After the decision is confirmed, firmware programming starts.

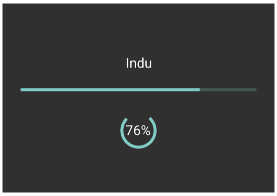
Kanja updates only one device – the selected one. If you have several devices, you have to repeat the process for all of them.

Note that certain device may have a different *Unit* name, but they share same firmware. For example all Indu devices and Digi share the same firmware. Hence, you may see generic firmware name instead the device name during the programming. In the example from Figure 8, Digi was updated but *Indu* text appeared in the progress page.



This will update the firmware. Are you sure?

(a) Confirmation window.



(b) Update progress.

Figure 8: Device update process.

4.2.1 Troubleshooting

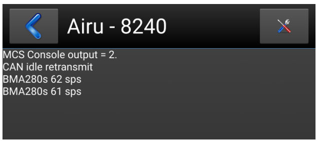
If for any reason the firmware update procedure fails the application will automatically try to repeat the update.

In case the application could not successfully update the device (device not working properly) shortly disconnect the power from Kanardia CAN network/device and connect it back. Once you will reconnect Kanja with Kanardia BLU the application will automatically start processing the problematic unit.

4.3 Console

The `Console` option may come handy during some troubleshooting and debugging. It shall be seldom used.

This command redirects the system output of the device over the CAN bus network and allows you to see various text messages that devices sends. In most cases you will not be able to understand them, but they can be useful to us and help us to understand certain problems. Figure 9 shows an example.

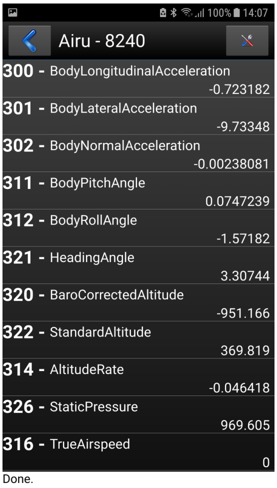


```
MCS Console output = 2.
CAN idle retransmit
BMA280s 62 sps
BMA280s 61 sps
```

Figure 9: A console page example. Device specific information and diagnostic messages appear on the page.

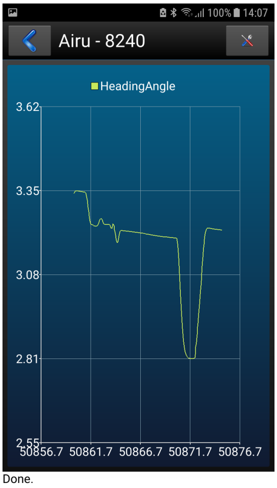
4.4 ICan

The `ICan` is also a debugging and troubleshooting tool. It opens a new page which shows all flight and engine messages sent by this device on the CAN bus. Service messages are not shown. Each message consists of its id and value. Values are received and shown in real-time, Figure 10a. Furthermore, when a parameter is selected, a new page with the real time chart is opened, Figure 10b.



```
100% 14:07
Airu - 8240
300 - BodyLongitudinalAcceleration
                         -0.723182
301 - BodyLateralAcceleration
                          -9.73348
302 - BodyNormalAcceleration
                       -0.00238081
311 - BodyPitchAngle
                         0.0747239
312 - BodyRollAngle
                          -1.57182
321 - HeadingAngle
                           3.30744
320 - BaroCorrectedAltitude
                          -951.166
322 - StandardAltitude
                           369.819
314 - AltitudeRate
                         -0.046418
326 - StaticPressure
                           969.605
316 - TrueAirspeed
                                 0
Done.
```

(a) ICan main page.



(b) ICan chart page.

Figure 10: An Example of Airu (AD-AHRS-GPS) device messages.

4.5 Reset

The `Reset` options sends the reset command to the device. Device shall restart as it was turned off and back on. Not all devices accept this command. This command shall be seldom used.

5 Special Commands

When a global menu symbol is selected (denoted as ⑥ on Figure 4), a page with special commands appears. Figure 11 shows an example.

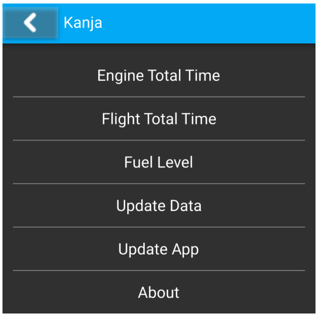


Figure 11: A page with special commands.

5.1 Engine Total Time

The `Engine Total Time` command opens a dialog where the engine time can be changed. See Figure 12 for instance.

1. Write the desired engine time value (in decimal hours) into the field.
2. Click `OK` button to update the engine total time.
3. Click on left arrow button to leave special commands page.

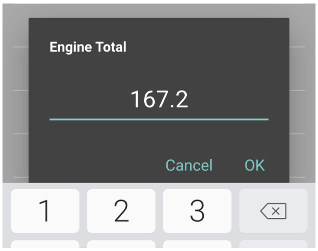


Figure 12: Setting a new engine total time.

When a new time was set, Kanja sends special message over CAN bus to all connected devices to accept the new time. Most devices ignore this message, but devices which have logging capability accept it. Please wait about 10 seconds after the command was sent before you turn the CAN bus network off.

5.2 Flight Total Time

Same principles as for the engine total time (see section 5.1) are also used for the flight total time.

5.3 Recorder

Recorder opens a recording page, which allows storing CAN bus information to some internal file. Once recording has been complete the file can be exported and further analysed on a PC.

For the time being recoding is possible only for Airu and/or Daqu devices.

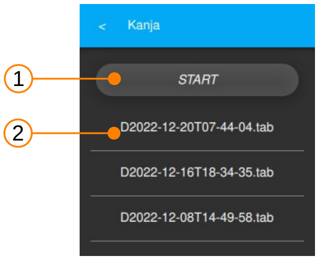
5.3.1 Principle of Operation

When the recording starts, Kanja builds list of messages sent by Airu (AD-AHRS-GNSS device) and Daqu (EMS device) and stores this list. Then it makes snapshots of corresponding values in 200 ms intervals (5 times per second) and stores them into a file.

Once recording was finished it is transfered to a PC or similar device for further analysis.

5.3.2 Start

Figure 13 shows the situation before the `Start` button is pressed. ① is the `Start/Stop` button and ② is a list of recorder files.



D2022-12-20T07-44-04.tab

D2022-12-16T18-34-35.tab

D2022-12-08T14-49-58.tab

Figure 13: Illustration of the recorder window before start.

On start an internal file is created. It is stored in the private application folder. The file name has the following format `Dyyyy-mm-dd`, which defines the date, where `yyyy`, `mm` and `dd` stand for year, month and day, respectively.

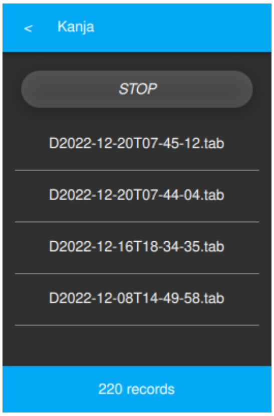
It continues with `Thh-mm-ss.tab` which defines time (hours, minutes, seconds) and the `.tab` file extension. Newly created file appears on the top.

Approximately five record snapshots will be added every second to the file. During the recording, the bottom part of the page is flushing every 4 seconds displaying number of records made so far.

Important: Please make sure that all Kanardia devices are running and you are receiving all messages before pressing the start option. Make sure there are no *red crosses*. Your engine shall be running before you press the start button.

5.3.3 Stop

Press the `Stop` button in order to stop recorder. This will close the file and make it ready for further processing. See also Figure 14.



D2022-12-20T07-45-12.tab
D2022-12-20T07-44-04.tab
D2022-12-16T18-34-35.tab
D2022-12-08T14-49-58.tab
220 records

Figure 14: Recorder window during the recording session. The bottom part is flushing.

5.3.4 Managing The Files

This part may be a bit tricky as each Android version behaves slightly differently. All files are stored in the private app folder, which can't be directly accessed.

Touching a filename on the list gives you two options:

Delete simply removes the selected file.

Export copies the selected file into the Android `Downloads` section. From there you use some other Android app to transfer it to the PC. Sharing it with Google Drive or similar may be the simplest solution.

Please note that certain Android versions may give you headache as the file will simply not appear in the Download section. We suggest trying with some other device rather than spending hours of searching for the solution. We do not like this, but there is not much we can do about it.

5.3.5 Opening The TAB file

The `tab` file format is a plain text format, where each row represents one record and parameters in the record are separated by a tab character. Each record has several flight and engine parameters like: date, time, position, altitude, static pressure, velocities, wind speeds, engine temperatures, engine pressures, RPMs and many others. Typically, the file is opened with Microsoft Excel or with LibreOffice Calc. Here are the steps needed to open the file in LibreOffice Calc. Steps in Microsoft Excel are similar.

1. Start the LibreOffice Calc.
2. Select the `File|Open` from the menu.
3. In the selection window, set `Filter` to `All Files`.
4. Search for a file with the tab extension.
5. `Calc` detects that a text file is being imported and it opens a configuration window. Please make sure that the `tab` option is selected as the separator and English (USA) as the language. This makes sure that decimal values are properly imported.

Please also note that the exact procedure may vary from spreadsheet version to version, but principles are similar.

5.4 Fuel Level

This is very special function limited to `Digi`. When software based tank (without fuel level sensors) is used with Digi, there is no way to enter the initial fuel level value as Digi has no input capability. However, when `Blu` is connected, then the fuel level can be adjusted using this feature. The fuel level is typically set before flight, but it can be set during the flight as well. It takes about six seconds for the system to remember the change.

5.5 Update Data

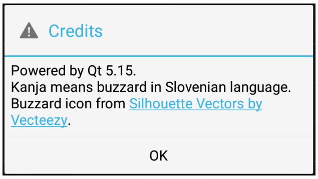
This command is obsolete. It can be used to download all firmware and special data from the server. Note that this is done automatically, when Kanja is started.

5.6 Update App

This command is obsolete. It opens a window, which tells you to check Play Store for the latest version.

5.7 About

The command opens a window with some publishing data and credits.



Powered by Qt 5.15.
Kanja means buzzard in Slovenian language.
Buzzard icon from Silhouette Vectors by Vecteezy.

Figure 15: The about and credits window.

6 Daqu Options

When selecting Daqu from list of connected devices, the extra options are available as on shown on Figure 16. They are explained in next subsections.

6.1 Channels

This option is used to configure Daqu and miniDaqu channels. An engine or aircraft sensor is connected to one of Daqu channels. Once a sensor is wired to a Daqu channel the channel must be properly set to support the sensor.

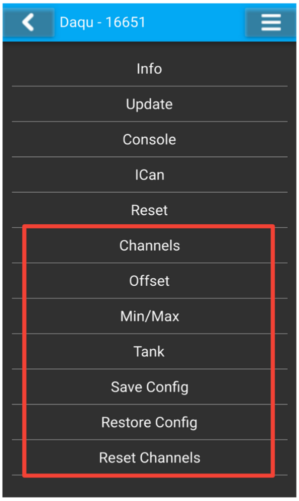


Figure 16: Daqu specific options are marked with a red frame.

For each sensor/channel pair, we must define the function of the sensor, the sensor make/type/model and some other information.

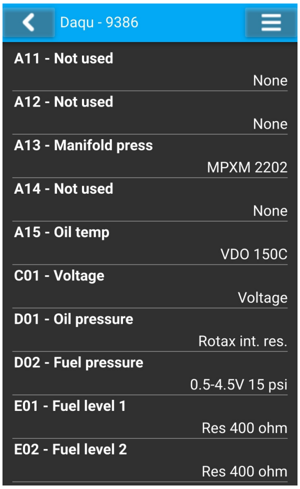
A proper channel configuration is a complex topic and it depends on a sensor type and a sensor function. These details are explained in the Daqu and miniDaqu manual.

When `Channels` option is selected, Kanja lists names of all detected channels together with their current function and sensor name. An example is shown on Figure 17.

Details on a channel configuration will be shown on two examples; a fuel pressure sensor and a fuel level sensor.

6.1.1 Fuel Pressure Example

Select a channel to which a fuel pressure sensor is connected. It is usually connected to channel D02 on Daqu or to channel F01 on miniDaqu. For this example we assume an active 15 PSI pressure sensor with 0.5-4.5V signal output range. 0.5 V represents 0 and 4.5 V represents max sensor range value. The sensor range value is 15 PSI, which equals to 1.034 bar. Such sensor is typically used on Rotax engines with carburetors. Note: Pressure in Kanja must be always given in bars.


Daqu - 9386

A11 - Not used
None

A12 - Not used
None

A13 - Manifold press
MPXM 2202

A14 - Not used
None

A15 - Oil temp
VDO 150C

C01 - Voltage
Voltage

D01 - Oil pressure
Rotax int. res.

D02 - Fuel pressure
0.5-4.5V 15 psi

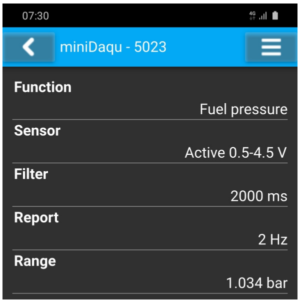
E01 - Fuel level 1
Res 400 ohm

E02 - Fuel level 2
Res 400 ohm

Figure 17: Illustration Daqu channels screen. Some channels are not used and some channels are not visible.

Please note that your case may be significantly different. You may use a different channel, sensor may work on a different principle and it may have a different range.

From the list of channels select channel D02. A new window appears where channel specific settings are specified.



miniDaqu - 5023

Function
Fuel pressure

Sensor
Active 0.5-4.5 V

Filter
2000 ms

Report
2 Hz

Range
1.034 bar

Figure 18: An example of fuel pressure sensor configuration.

1. The `Function` lists all possible functions for given channel. For this example, search for the `Fuel pressure` and select it.

2. The `Sensor` lists all possible sensors that can be connected to this channel. Our fuel pressure sensor has a 0.5 to 4.5 V output range. Thus, we search for the `Active 0.5-4.5 V` item.

3. The `Filter` tells how quickly Daqu responds to a sensor change. A small filter results in fast respond and a large filter in slower respond. We do not need a very fast response on fuel pressure change, hence a `2000` ms, which equals to 2 secs is selected. Note that Daqu manual show recommended filter values in seconds, but Kanja requires them in milliseconds.

4. The `Report` tells how frequently Daqu transmits sensor readings on the CAN bus. A value of `2 Hz` seems reasonable. This means that Daqu will transmit fuel pressure reading two times per second. Daqu manual specifies this value in seconds. Two times per seconds means every 500 ms or 0.5 sec.

5. `Range`: Finally, sensor working range must be defined. Here we tell that sensor at its max value of 4.5V equals to `1.034 bar`. Remember, for pressure sensors, this value must be always given in bars.

Once all data is entered, you shall get something similar to Figure 18.

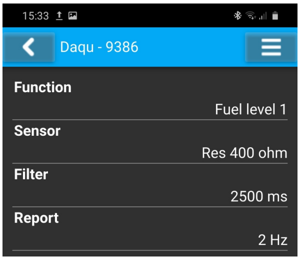
Once all parameters are entered, press the `Back` arrow on top left screen corner to get back to the list of channels. Please note that the new channel settings were sent to Daqu, but they were not stored permanently yet. If you want to make this permanent, you have to press `Back` again to close the `Channels` page and go back to Daqu options. Now the change is permanent.

6.1.2 Fuel Level Example

Fuel level example is similar to one above. Here we assume that a resistive sensor with range between 3 and 180 Ohm is connected to the E01 channel. Again, your case may be significantly different: a different sensor may be used, connected to a different channel.

Once this operation is complete, Daqu will be able to recognize the fuel level sensor and read from it – it will be able to read Ohms from the sensor. However, it will still not be able to convert the sensor readings to fuel volume (liters), which is what we need. To get the fuel volume, a further step is needed. This step is explained in section 6.4.

1. Select `Function` and then search the list for the `Fuel level 1` item.


15:33

Daqu - 9386

Function
Fuel level 1

Sensor
Res 400 ohm

Filter
2500 ms

Report
2 Hz

Figure 19: An example of fuel level sensor setting for channel E01.

2. Open `Sensor` option and select the `Res 400 ohm`.
3. Set `Filter` to maximum – `2500` ms, which equals to 2.5 secs. Daqu manual show these values in seconds, but Kanja requires them in milliseconds.
4. Set `Report` to `2 Hz`. This means that Daqu will transmit fuel level information two times per second. Daqu manual specifies this value in seconds. Two times per seconds equals to 0.5 sec (500 ms).

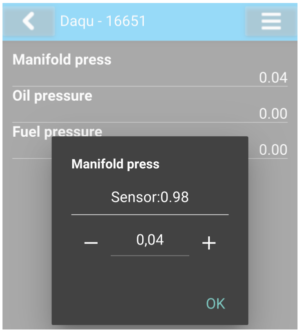
6.2 Offset

On selecting the `Offset` item new dialog opens, which enables user to edit various offset values for sensors connected to the system. These is useful in cases where you know the exact (correct) value and you would like to adjust (offset) the sensor for some constant.

Following offset values can be edited:

- Fuel pressure – adjust the sensor to show zero, when the sensor is not loaded. In some cases, Rotax iS, for example, the sensor must be removed and the ECU off.
- Oil pressure – adjust the sensor to show zero. Remove the sensor from engine if possible. Namely, there could be some small residual pressure in the system.
- Current 1 and 2 – adjust to show zero and make sure that no current is flowing trough sensor.
- Manifold pressure – adjust the sensor to current static pressure.

When adjusting offset value observe `Sensor` value to adjust to appropriate value.


Daqu - 16651

Manifold press 0.04

Oil pressure 0.00

Fuel pressure 0.00

Manifold press

Sensor:0.98

– 0,04 +

OK

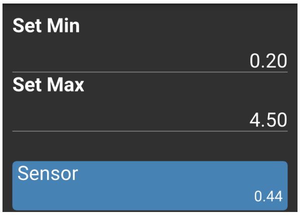
Figure 20: An example of manifold pressure offset window.

6.3 Min/Max

This option is used to define min and max positions for flaps and trims. It shall be used after corresponding channel was set.

The example given next shows flap position situation. Procedure for a trim position is very similar.

Once `Min/Max` was selected, a list of detected flaps and trims are shown. Select the one you want to set. In our case, it is a `Flap pos`. Your case may be different.


Set Min 0.20

Set Max 4.50

Sensor 0.44

Figure 21: Flap position example.

Figure 21 illustrates a flaps position case. The `Sensor` item shows the value detected by the sensor. Depending on the channel type it will show either voltage or resistance. When flaps are moving, this value must change accordingly. If it does not, please check the channels settings and wiring.

1. Extend the flaps and remember the sensor value when fully extended. In the example this value is 0.20. Retract the flaps and remember the sensor value when retracted. In the example, this is 4.50. Your case will be different, of course.
2. Click on the `Set Min` and set this value to the *extended* value.
3. Click on the `Set Max` and set this value to the *retracted* value.

If your flap/trim position indicator moves in the wrong direction, then exchange the `Set min` and `Set max` values.

6.4 Tank

The goal of the fuel level (tank) calibration is to convert the sensor readings (which are in Ohms for resistive sensors or in voltages for active sensors) into fuel volume. *Fuel volume must be always given in liters!*

Before a calibration can be started, make sure that:

- Fuel level sensor is properly wired to the Daqu. See Daqu or miniDaqu manual for more details.
- The channel to which the sensor is wired is properly configured. See section 6.1.2 for an example.

First you have to select which tank you wish to configure. `Tank 1` corresponds to `Fuel Level 1` and `Tank 2` to `Fuel Level 2` in Daqu channel/function configuration.

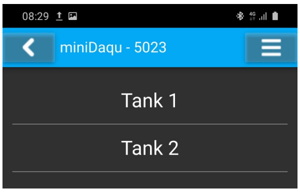


Figure 22: Select tank to configure.

Once a tank is selected, a window will open. It shows some options on the top and tank *shape* in the form of a diagram. An example is shown on Figure 23.

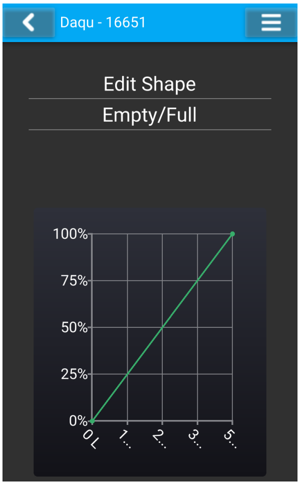


Figure 23: Tank configuration options.

Vertical axis of the diagram corresponds to sensor readings expresses in percentages. 0 % equals to the sensor value when tank is empty and 100 % equals to the sensor value when tank is full. Horizontal axis shows actual volume expressed in liters.

6.4.1 Edit Shape

In most cases you will start with the `Edit Shape` option. This is where sensor readings are combined with the actual fuel volume. You have to define at least two and at most 15 volume-sensor pairs.

The edit shape window consist of there parts. The top part has two commands. In the middle you see current sensor reading. It shows either Ohms (for resistive sensors) or voltages for active/capacitive sensors. The bottom part lists volume-sensor pairs.

It is important that the sensor value is *alive*. Its value must change logically if the fuel level changes. This indicates that sensor has been properly connected and its channel has been properly configured. See section 6.1.2 for more details.

Please note that some sensors indicate largest value when tank is full and smallest value when tank is empty. This is perfectly normal.

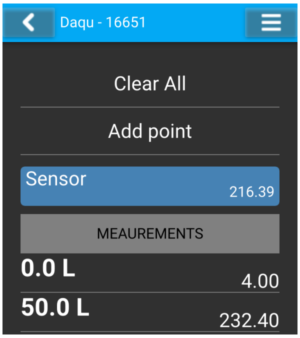


Figure 24: Tank editing window illustration.

Clear All

This command will remove all volume-sensor pairs from the list. You may start with it, if you want to calibrate tank from scratch. Beware the command does not ask for a confirmation.

Adding a Point

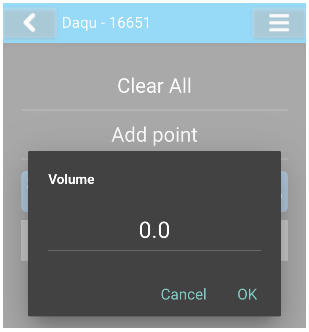
Click the `Add Point` option to add a new volume-sensor pair – a point. Figure 25 You will have to enter current volume inside the tank and sensor value for this point. The points entered do not have to be in order. A new point will be inserted to the appropriate place, soreted by volume. However it is important that all sensor values are either in ascending or in descending order. If they are not, a warning dialog will open when you will try to close this page.

Removing a point

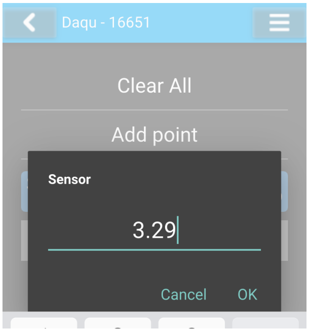
If you want to delete a measurement point click on the measurement volume-sensor pair and delete dialog will appear. Figure 26 illustrates and example.

6.4.2 Verification With ICan

Fuel level indication can be verified with the ICan option. Once Tank page was closed tank settings were stored permanently in Daqu (miniDaqu). If all is correct, Daqu will start transmitting fuel level CAN messages on the bus. Open the ICan option and search for the `FuelTankQuantity_0` and



(a) Set volume.



(b) Set sensor.

Figure 25: Adding a volume-sensor pair. Initial sensor value is read directly from sensor. You may change it, if you like.

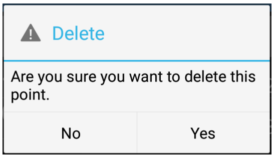
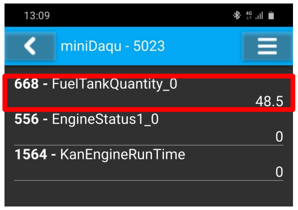


Figure 26: Deleting a measurement point from the list.

`FuelTankQuantity_1`, which equal to `Fuel level 1` and `Fuel level 2`, respectively.



13:09

miniDaqu - 5023

668 - FuelTankQuantity_0	48.5
556 - EngineStatus1_0	0
1564 - KanEngineRunTime	0

Figure 27: Checking tank indication value on the bus. Fuel level 1 is identified as `FuelTankQuantity_0`. Value on the bus is always shown in liters.

6.4.3 Example

In this example we will calibrate a tank of 12.5 US gal. This will act as fuel level 1 (=Tank 1). Calibration will be done in 3 gal steps. Note that we will have to convert US gal to liters as liters must be entered into Kanja.

One US gallon equals to 3.7854 liters.

1. Make sure sensor was properly connected and its channel configured accordingly.

2. Have a notebook and pencil at hand.

3. Make sure you have enough fuel and enough cans ready.

4. Drain fuel tank completely.

5. Make sure you have your cans calibrated and marked. Fuel will be added in steps. Steps of 3 US gal or 10 liter steps are sufficient. Do not try to be super precise. Fuel sensors are not very precise. The maximal number of measuring points is 18. In practice, 5 points are usually enough.

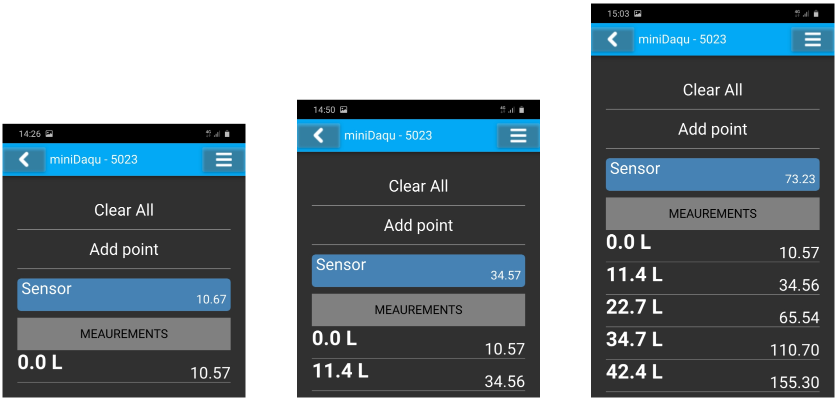
6. Make sure the aircraft is level for the cruise position.

7. In Kanja app, select Daqu/miniDaqu from the list of devices and then select `Tank` and then `Tank 1`. This shall open a page similar to Figure 24. There may be some measurement points listed.

8. Select `Clear All` to remove any previous measurements.

9. Select `Add point`, enter `0 liters` for the volume and keep the current sensor value. Tank is empty at the moment and sensor is at its bottom position. Write down both values into the notebook. Figure 28a shows this initial situation. This will the first point in the measurements list. It identifies the empty tank condition.

10. Pour some known quantity of fuel into the tank. In this example this is 3 US gal, which equals to 11.36 liters. Observe the *Sensor* value at the bottom of the window. This value must be alive – it must change. This value shows the raw sensor reading in Ohms for the resistance based sensors or voltage for active fuel level sensors (capacitive, or pressure based). Note: Usually, you have to put some fuel before the fuel reaches the bottom of the sensor and the sensor starts reacting. This amount


14:26
miniDaqu - 5023
Clear All
Add point
Sensor 10.67
MEAUREMENTS
0.0 L 10.57

14:50
miniDaqu - 5023
Clear All
Add point
Sensor 34.57
MEAUREMENTS
0.0 L 10.57
11.4 L 34.56

15:03
miniDaqu - 5023
Clear All
Add point
Sensor 73.23
MEAUREMENTS
0.0 L 10.57
11.4 L 34.56
22.7 L 65.54
34.7 L 110.70
42.4 L 155.30

(a) One point. (b) Two points. (c) All points.

Figure 28: Adding points to the measurements list.

varies from case to case and it may be as small as 0 or as large as 15 liters (4 US gal) or even more.

Wait for sensor to stop changing and press `Add Point`. Enter 11.36 for volume and keep the sensor value. This adds a second point to the list, Figure 28b. Write these two values down into notebook, too.

11. Pour some more fuel into the tank. Say another 3 US gal. Once the sensor value stabilizes, write down the total amount of fuel (now 6 US gal = 22.71 liters) and sensor value into the notebook.

12. Repeat the previous step until the tank is almost full. Be careful here. You will probably have a case where sensor stops reacting (has reached its max point) way before the tank is completely full. Stop at the point when sensor does not react anymore. In our case the sensor reached its top position at 11.2 US gal (=42.4 liters). This will be our last point, although the actual tank is a bit larger (12.5 US gal). Final situation is shown on Figure 28c.

13. Go one page back - return to the initial tank page. Now you shall see the shape of the tank. Observe the non-linear shape of the tank curve, Figure 29.

14. Close all pages until you get back the list of devices. Select Daqu (or miniDaqu) and then `ICan` option. This opens a page which lists mes-

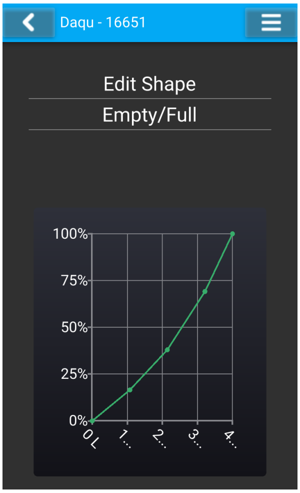
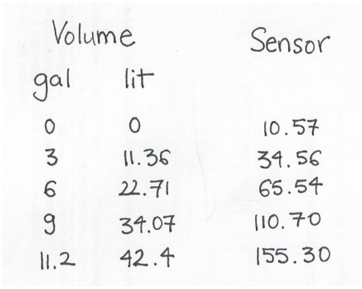


Figure 29: Shape of the calibrated tank.

sages detected on the CAN bus. You should see something like shown on Figure 27.

Taking notes after each measuring point is a very good idea. It can easily happen that something goes wrong and without the notes you have to repeat the whole procedure. But if you have notes, you can reconstruct the tank points without messing with fuel. The only difference here is that you have to change the sensor value according to the value from the notes. Notes from our example are illustrated on Figure 30.



Volume		Sensor
gal	lit	
0	0	10.57
3	11.36	34.56
6	22.71	65.54
9	34.07	110.70
11.2	42.4	155.30

Figure 30: Notes taken during tank calibration example.

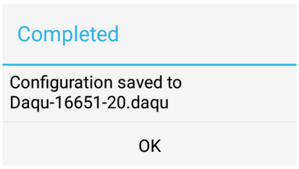
6.5 Configuration

This section presents options to save, restore and import Daqu configurations. Configuration consists of:

- Daqu channels settings. See section 6.1.
- Sensor offsets, section 6.2.
- Tank calibration, section 6.4.

6.5.1 Save Config

To save your configuration press `Save Config`. A window will open as shown in Figure 31. Press OK to close the window. You can have multiple configurations saved. For easier use later, we suggest renaming your configurations. This procedure is described in section 6.5.3.


Completed

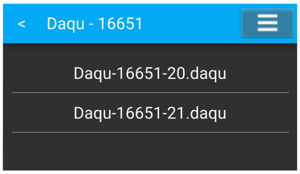
Configuration saved to Daqu-16651-20.daqu

Figure 31: An example of saved configuration.

6.5.2 Restore Config

This option offers you a list of previously saved configurations. An example is shown in Figure 32. Choose the one, you want to restore.

Caution, this will overwrite your existing configuration in Daqu and you may lose recent changes.



Daqu-16651-20.daqu

Daqu-16651-21.daqu

Figure 32: List of saved configurations.

6.5.3 Config Manager

When a global menu symbol is selected (top right corner in Figure 16), a page with special commands appears. Figure 33 shows an example. Choose `Config Manager` and the list of saved configurations shows up. You have options of deleting, renaming or importing.

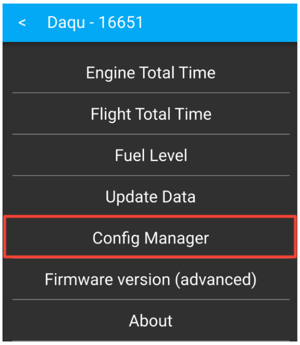


Figure 33: Special commands.

Rename

Renaming is done in 3 steps:

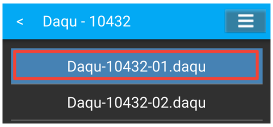
- Select configuration which you want to rename. See Figure 34a.
- Select `Rename`, example is in Figure 34b.
- Type in the name you want, then press OK . See Figure 34c.

Delete

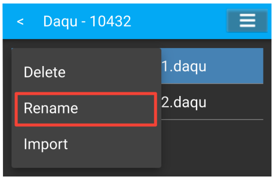
Deleting is done in 2 steps:

- Select configuration which you want to delete. See Figure 35a.
- Select `Delete`, example is in Figure 35b.

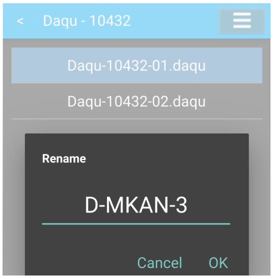
Caution, once configuration was deleted, it cannot be restored.



(a) Choose configuration to rename.

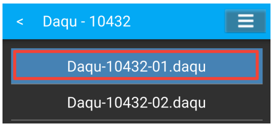


(b) Select rename.

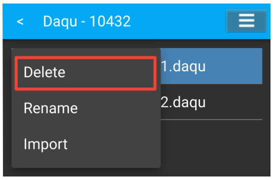


(c) Choose new configuration name.

Figure 34: Config Manager - renaming selected configuration.



(a) Choose configuration to delete.



(b) Select delete.

Figure 35: Config Manager - deleting selected configuration.

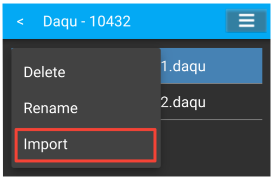
Import

Importing is done in 2 steps:

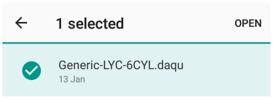
- Choose any existing configuration to access the `Import` command and select it. See Figure 36a.
- Select Daqu configuration file which you want to import and press open (file name must end as *.daqu*). Example is in Figure 36b.

6.6 Reset Channels

This option will reset all Daqu sensor channels to default values. Be aware that all channel settings will be gone and you will have to set each channel again. Think twice, before you use this option.



(a) Select import.



(b) Choose the file you want to import.

Figure 36: Config Manager - importing Daqu configuration.

7 Indu & Digi Options

When selecting `Indu` or `Digi` from list of connected devices, the extra options are available as on shown on Figure 37. They are explained in next subsections.

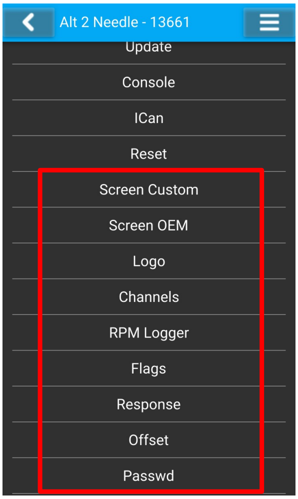


Figure 37: Indu and Digi specific options are marked with a red frame.

Although Digi and Indu are very different in the form, they share the same code base. In fact the same software is running in their micro-controllers.

Some options are not supported in certain Indu/Digi device. This depends from one device to other as they may have slightly different hardware built-in. Such limitations will be specified in each sub section.

7.1 Screen Custom

The `Screen Custom` option allows you to change the look of your LCD display. Be careful with it, as the changes can't be undone – Indu or Digi will always remember the last screen that was received.

This option is mostly used with Digi, which layout is tuned according to sensors and engine specifics. It can be also used for any Indu, but there is seldom a need for that.

The `Screen Custom` option used together with the `Customizer` app. The Customizer app is used to prepare some Indu or Digi layout for its LCD display. The layout is stored in a file with *iml* extension. This file must be copied to your Android device. Once the file is in the Android device, the *iml* file is transferred into Indu or Digi.

More details about how to prepare the layout and transfer the file into the Android device are given in the Customizer manual.

1. Select the `Screen Custom` option.
2. A list of files available in your Android device will appear. The files will have *iml* extension.
3. Select the file you would like to transfer.
4. The transfer requires about 2-3 seconds to complete and you should see a new layout on the screen afterwards. New layout is permanently saved on the device.

Note that at the time being you can't reverse the operation. You can not copy the file from the Indu/Digi into Android device.

7.2 Screen OEM

This option was designed to allow our OEM customers to access various layouts and copy them into Indu or Digi. These layouts are copied from our server into Android each time Kanja is started. Here you simply select one of many pre-build options, which are specific to OEM companies.

Do not use this option unless you know exactly what you are doing. You may change your Digi/Indu layout into something completely useless by accident. Such change can not be undone.

OEM files have *isb* extension. This is a binary version of the *iml* original.

1. Select the `Screen OEM` option.
2. A list of folders and files will appear, see Figure 38 for an example.
3. Select a folder first and then select a file with *isb* extension.
4. This will copy the layout file into the Digi/Indu.

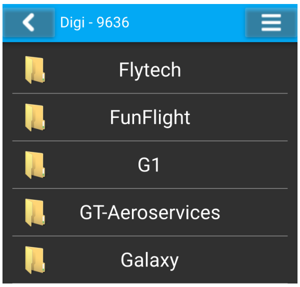


Figure 38: Some folders from the Screen OEM option. One or more folder levels may be used before correct *isb* file is found.

7.3 Logo

This command shall only be used by OEM manufacturers. They can choose their logo screen to appear on the screen when the instrument starts. All these screens are predefined.

You can't add/use your own screen for the startup logo unless we approve it and copy to our server. We approve only logos from OEMs. Please contact `support@kanardia.eu` for the details.

7.4 Channels

This option shall **NOT** be used for most Indu and Digi instruments. The channels in Indu and Digi should be configured only if the hardware supports external sensors and if the sensor is actually connected to the instrument. In all other cases the channel function shall be set as *Not Used*.

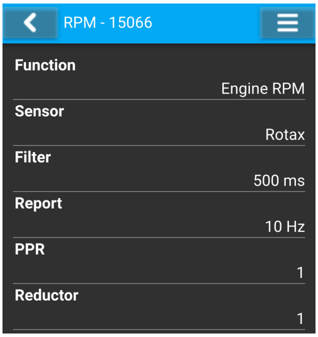
A few of Digi or Indu instruments were equipped with special additional hardware, which acts like a miniature Daqu. The most typical case is a standalone Indu RPM indicator.

The channels behave in a similar way as this was already described in section 6.1 on page 22.

Note that channels that are listed are not necessarily supported by the hardware. Only a subset of these are supported. There is nothing that will prevent you from editing a channel, even if it is not supported.

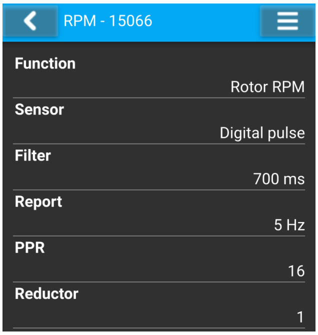
7.4.1 Standalone Indu RPM

Here we will give an example for the most frequently used one – Indu RPM standalone indicator. This Indu version supports channels Z01, Y01 and Y02. Others are not used. Figure 39a illustrates channel setting for the Rotax UL(S) 912 engine.


RPM - 15066

Function	Engine RPM
Sensor	Rotax
Filter	500 ms
Report	10 Hz
PPR	1
Reductor	1

(a) Z01 channel settings for the Indu RPM indicator for Rotax engine.


RPM - 15066

Function	Rotor RPM
Sensor	Digital pulse
Filter	700 ms
Report	5 Hz
PPR	16
Reductor	1

(b) Y01 channel settings for some RPM rotor with 20 pulses per revolution (PPR).

Figure 39: Channel settings examples for Indu instruments with special additional hardware.

Sometimes similar indicators are also used as rotor RPM indicators or even both engine RPM and rotor RPM indicators. Rotor sensor are usually connected to Y01 or to Y02 channel, Figure 39b.

Please refer also to the Daqu manual for some details about channel settings. The underlying principles used in Indu are identical to those in Daqu. There are some terminology differences, though. Here we use `Report`. This means how frequently will be this function reported to the CAN bus. In the Daqu manual we define the same value in seconds (s).

7.5 RPM Logger

This option shall **NOT** be used for most Indu and Digi instruments.

If you use a set of several Indu instruments, it is not always obvious who stores the engine time information. This may not be the RPM, but it may be the Altimeter or a Combo instead. Please contact `support@kanardia.eu` if unsure.

In the cases where an Indu or a Digi also stores engine time information this settings define how the engine total time is calculated.

RPM Threshold defines the RPM threshold that must be reached in order the we can consider that an engine is running. When engine RPM are below the threshold the logging system does not count this into total engine time.

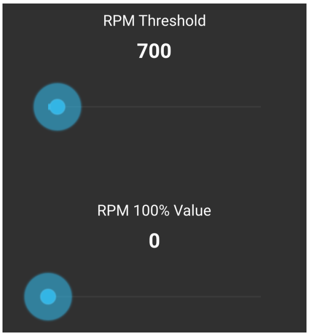
RPM 100% Value gives us two cases.

1. The value is set to `zero`. This is the *standard case* for most engines, Figure 40a.This means that whenever the engine RPM are above the threshold, the actual time elapsed is added to the total engine time.
2. When the value is not zero then this defines the reference RPM. Let's say the 100% value is set to 5000 RPM. In this case, the engine total time increased by the logger depends on RPM. Say the RPM are 3000. In this case only 3000/5000 $\rightarrow 0.6s$ will be added to total time. When RPM are 5000 then 5000/5000 $\rightarrow 1.0s$ will be added and if RPM are 5800/5000 $\rightarrow 1.16s$ will be added. This means that logger counts time faster when RPM exceed the 100% value and slower when RPM are below. Figure 40b illustrates the example.

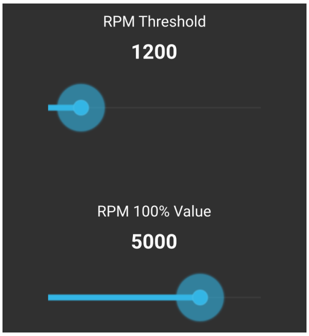
7.6 Flags

This option shall **NOT** be used for most Indu and Digi instruments. All instruments shall arrive pre-configured and you should never change any flag.

Use this option only in the case of some failure or some shortcoming. The actually *flags* that you see depend on the instrument type and also on the additional hardware, which may not appear in all Indus. In addition the correct value of some flags may also depend on other instruments connected to the CAN bus. We strongly advice the you consult us before you do any change. Contact `support@kanardia.eu` for the details. Figure 41 illustrates an example.

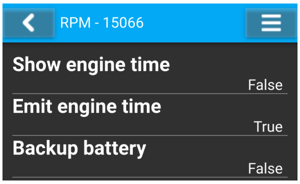

RPM Threshold
700
RPM 100% Value
0

(a) Standard settings.


RPM Threshold
1200
RPM 100% Value
5000

(b) Special settings.

Figure 40: Engine total time logging settings. Two typical examples.



Show engine time
False
Emit engine time
True
Backup battery
False

Figure 41: Flag values depend on different circumstances and also on other instruments connected to the same CAN bus.

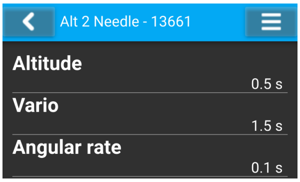
7.7 Response

This option is seldom used and in most cases it shall not be changed.

This option allows you to change a time constant of a low pass filter for some frequently used parameters.

Lets assume that some sensor readings is zero and then it suddenly jumps to 100. The sensor output is filtered with a low pass filter. This means it will not react immediately on a change, but it will gradually raise from 0 to 100 over some time. The time constant tells how much time (in seconds) is needed for an indicated value to reach 63. Large time constant yields slower response while short constant yields swift response.

When the `Response` command is selected a page with various parameters appear, Figure 42. You will see also parameters that have nothing to do with this particular Indu/Digi. Change only the one that is related with the actual Indu. If you change anything else the change will be ignored.


Alt 2 Needle - 13661

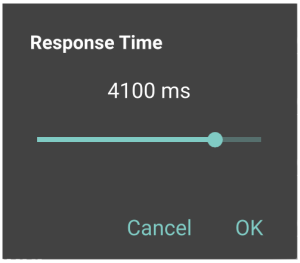
Altitude 0.5 s

Vario 1.5 s

Angular rate 0.1 s

Figure 42: A list of adjustable parameters. The complete list is not shown on this example.

Maybe the best candidate the response change is vertical speed a.k.a. *Vario*. Some pilots like a swift response and some want a lazy one. Select the `Vario` from the list and adjust its response time. Figure 43 shows an example where the time constant is set to 4.1 seconds.


Response Time

4100 ms

Figure 43: Time constant for vertical speed (Vario).

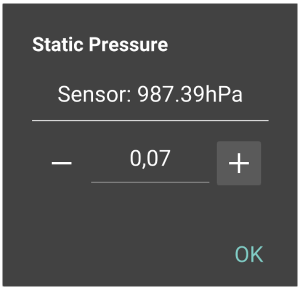
7.8 Offset

This option is useful only for Indu instruments with built-in static and/or dynamic pressure, like master altimeter, master airspeed indicator or combo. It allows sensor offset of static or dynamic pressure. Electronic sensors may drift a bit over longer period of time and here we can adjust their offset.

Before any sensor adjustments please make sure that instruments are turned on for at least 10 minutes. This will allow internal temperature to stabilize.

7.8.1 Static Pressure

In order to adjust static pressure, you have to know correct current static pressure. Some barometric device or another reference sensor is needed for this operation. Adjust the static pressure offset until the Indu pressure do not match the reference static pressure. The static pressure in this window is always shown in hPa. If your reference pressure was given in other units, you have to convert it to hPa.


Static Pressure

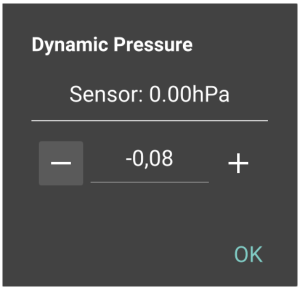
Sensor: 987.39hPa

Figure 44: Example of static pressure offset.

7.8.2 Dynamic Pressure

Luckily, the dynamic pressure offset does not require any reference. It only requires that pitostatic system is clean, not blocked (remove any pitot tube or static port cover) and there must be no wind – do this inside hangar on a calm day.

Adjust the dynamic pressure offset so that the sensor value will be zero or close to zero.


Dynamic Pressure

Sensor: 0.00hPa

Figure 45: Change the offset until the sensor shows zero.

7.9 Password

This is special option which allows access to a couple of *rescue* commands. Don't use it unless we ask you to do so. A special password value must be entered. If the password is correct, some background will be executed.

8 Joyu Options

When selecting Joyu from the list of connected devices, an extra option `Joyu Config` is available as shown in Figure 46.

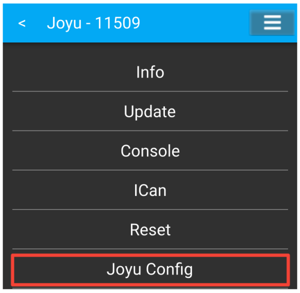


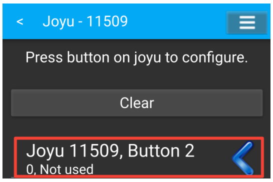
Figure 46: Joyu extra option.

8.1 Joyu Config

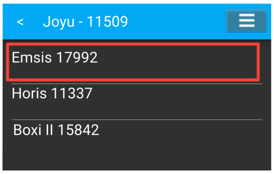
This option gives you the ability to assign Joyu buttons/wheel commands to available devices in the CAN bus network. Each Joyu button/wheel can only be assigned to a single device.
Configuration is done in next steps:

1. Press the button or scroll the wheel that you want to configure on Joyu. Select the item marked with the blue arrow. Example is shown in Figure 47a, see red box.
2. List of available devices will show up. Select the one, you want to assign button/wheel to. Example is in Figure 47b.
3. All available actions for the selected device will appear as shown in Figure 47c. Choose your preferred action. If choosing `Not used`, button will not be assigned.

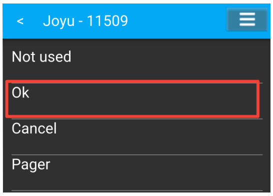
4. Repeat steps 1-3 to assign other Joyu buttons/wheel actions.



(a) A response to Joyu.



(b) Device select.



(c) Action select.

Figure 47: Joyu Config steps example.

Example of Joyu S/N(11509) `Button 2` assigned to Emsis S/N(15784) as `Ok` action is shown in Figure 48.

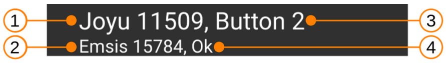


Figure 48: Joyu button/wheel details.

① Joyu and its serial number.

② Receiving device and its serial number.

③ Button on Joyu.

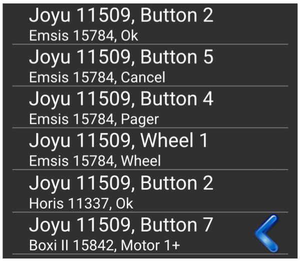
④ Action on the receiving device.

Example of Joyu buttons assigned to Emsis, Horis and Boxi II actions is shown in Figure 49.

Replacing the device requires a new configuration because the serial number changes.

9 Limited Conditions

Although a great care was taken during the design, production, storage and handling, it may happen that the Product will be defective in some way. Please read the following sections about the warranty and the limited operation to get more information about the subject.



Joyu 11509, Button 2
Emsis 15784, Ok
Joyu 11509, Button 5
Emsis 15784, Cancel
Joyu 11509, Button 4
Emsis 15784, Pager
Joyu 11509, Wheel 1
Emsis 15784, Wheel
Joyu 11509, Button 2
Horis 11337, Ok
Joyu 11509, Button 7
Boxi II 15842, Motor 1+

Figure 49: List of assigned Joyu buttons.

9.1 Warranty

Kanardia d.o.o. warrants the Product manufactured by it against defects in material and workmanship for a period of twenty-four (24) months from retail purchase.

Warranty Coverage

Kanardia's warranty obligations are limited to the terms set forth below:

Kanardia d.o.o. warrants the Kanardia-branded hardware product will conform to the published specification when under normal use for a period of twenty-four months (24) from the date of retail purchase by the original end-user purchaser ("Warranty Period"). If a hardware defect arises and a valid claim is received within the Warranty Period, at its option and as the sole and exclusive remedy available to Purchaser, Kanardia will either (1) repair the hardware defect at no charge, using new or refurbished replacement parts, or (2) exchange the product with a product that is new or which has been manufactured from new or serviceable used parts and is at least functionally equivalent to the original product, or, at its option, if (1) or (2) is not possible (as determined by Kanardia in its sole discretion), (3) refund the purchase price of the product. When a refund is given, the product for which the refund is provided must be returned to Kanardia and becomes Kanardia's property.

Exclusions and Limitations

This Limited Warranty applies only to hardware products manufactured by or for Kanardia that have the "Kanardia" trademark, trade name, or logo affixed to them at the time of manufacture by Kanardia. The Limited Warranty does not apply to any non-Kanardia hardware products or any software, even if packaged or sold with Kanardia hardware. Manufacturers, suppliers, or publishers, other than Kanardia, may provide their own warranties to the Purchaser, but Kanardia and its distributors provide their products *AS IS*, without warranty of any kind.

Software distributed by Kanardia (with or without the Kanardia's brand name including, but not limited to system software) is not covered under this Limited Warranty. Refer to the licensing agreement accompanying such software for details of your rights with respect to its use.

This warranty does not apply: (a) to damage caused by use with non-Kanardia products; (b) to damage caused by accident, abuse, misuse, flood, fire, earthquake or other external causes; (c) to damage caused by operating the product outside the permitted or intended uses described by Kanardia; (d) to damage caused by service (including upgrades and expansions) performed by anyone who is not a representative of Kanardia or an Kanardia Authorized Reseller; (e) to a product or part that has been modified to significantly alter functionality or capability without the written permission of Kanardia; (f) to consumable parts, such as batteries, unless damage has occurred due to a defect in materials or workmanship; or (g) if any Kanardia serial number has been removed, altered or defaced.

To the extent permitted by applicable law, this warranty and remedies set forth above are exclusive and in lieu of all other warranties, remedies and conditions, whether oral or written, statutory, express or implied, including, without limitation, warranties of merchantability, fitness for a particular purpose, non-infringement, and any warranties against hidden or latent defects. If Kanardia cannot lawfully disclaim statutory or implied warranties then to the extent permitted by law, all such warranties shall be limited in duration to the duration of this express warranty and to repair or replacement service as determined by Kanardia in its sole discretion. Kanardia does not warrant that the operation of the product will be uninterrupted or error-free. Kanardia is not responsible for damage arising from failure to follow instructions relating to the product's use. No Kanardia reseller, agent, or employee is authorized to make any modification, extension, or addition to this warranty, and if any of the foregoing are made, they are void with respect to Kanardia.

Limitation of Liability

To the extent permitted by applicable law, Kanardia is not responsible for indirect, special, incidental or consequential damages resulting from any breach of warranty or condition, or under any other legal theory, including but not limited to loss of use; loss of revenue; loss of actual or anticipated profits (including loss of profits on contracts); loss of the use of money; loss of anticipated savings; loss of business; loss of opportunity; loss of goodwill; loss of reputation; loss of, damage to or corruption of data; or any other loss or damage howsoever caused including the replacement of equipment and property, any costs of recovering, programming, or reproducing any program or data stored or used with Kanardia products and any failure to maintain the confidentiality of data stored on the product. Under no circumstances will Kanardia be liable for the provision of substitute goods or services. Kanardia disclaims any representation that it will be able to repair any product under this warranty or make a product exchange without risk to or loss of the programs or data. Some jurisdictions do not allow for the limitation of liability for personal injury, or of incidental or consequential damages, so this limitation may not apply to you.

9.2 TSO Information — Limited Operation

This product is not TSO approved as a flight instrument. Therefore, the manufacturer will not be held responsible for any damage caused by its use. The Kanardia is not responsible for any possible damage or destruction of any part on the airplane caused by default operation of instrument.