

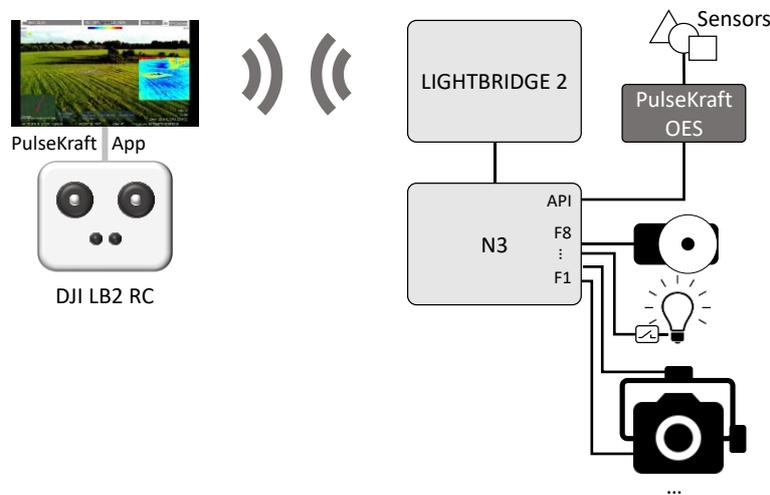
# PulseKRAFT 2.2.2

## Enabling 12+ PWM Channels, Enhanced Telemetry & Missions for DJI Matrice 100, Matrice 600, A3 + LB2, N3 + LB2

### Contents

1. <b>Overview</b> - Enabling Control to additional Onboard Devices on DJI Platforms + Enhanced Mission Planning & Control.....	1
2. Installing the “PulseKRAFT” App (Android).....	3
3. Preparation of the Onboard Device (“software only” option).....	9
4. Wiring the Onboard Device.....	11
5. Activation of the Onboard Software.....	15
6. Taking Control.....	17
7. Restart / Safe shutdown.....	20
8. Additional App Features.....	20
9. Waypoint Mission Editor.....	24
10. Mission Control.....	25
11. Add-on for further PWM Channels and additional Sensor Data.....	26
12. Troubleshooting.....	30
13. Copyright.....	31
14. Privacy Policy, Terms of Use and End User License Agreement (EULA).....	31

### 1. Overview - Enabling Control to additional Onboard Devices on DJI Platforms + Enhanced Mission Planning & Control



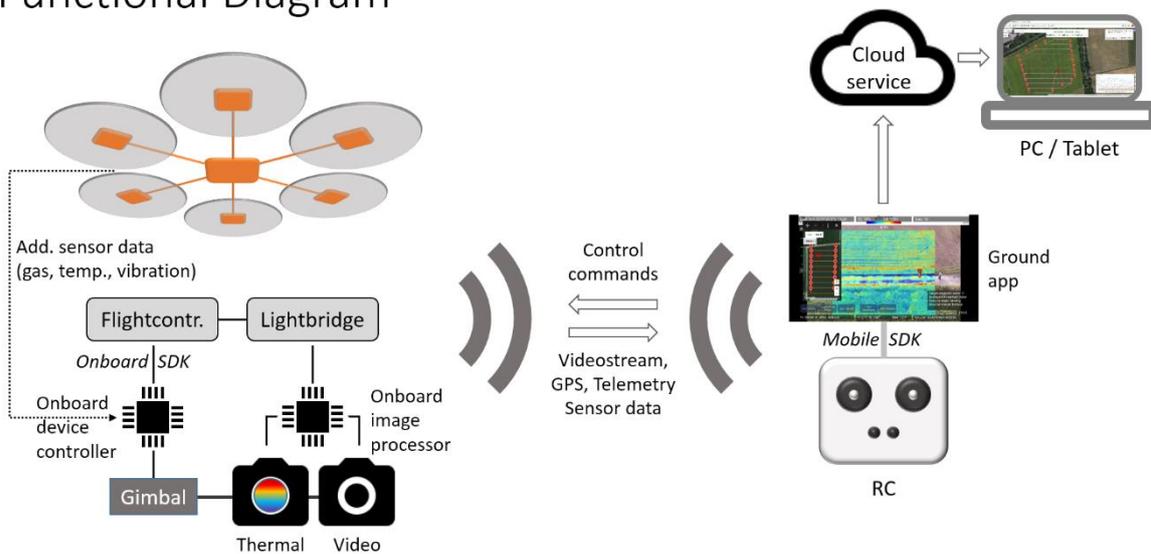
At the time of this writing, neither the DJI Matrice platforms (M100, M200 series, M600) nor the A3 / N3 flight controllers in combination with Lightbridge 2 provide out-of-the-box options to map PWM

output signals of F1 ... F8 ports to the wheels, buttons and switches of the DJI remote controller (except landing gear switch and shutter button).

The PulseKRAFT solution allows mapping more than 12 PWM channels, extended telemetry, backward data channel and some enhanced waypoint mission features. With a little add-on board, you will get further PWM outputs, a universal serial (UART) data interface for enhanced integration scenarios like LiDAR based distance control and altimeter. SBus + MAVlink output will follow soon. It only requires DJI devices supporting the Mobile SDK and Onboard SDK (through the API port on the flight controller).

**Application example** (some parts are not part of the PulseKRAFT solution):

## Functional Diagram



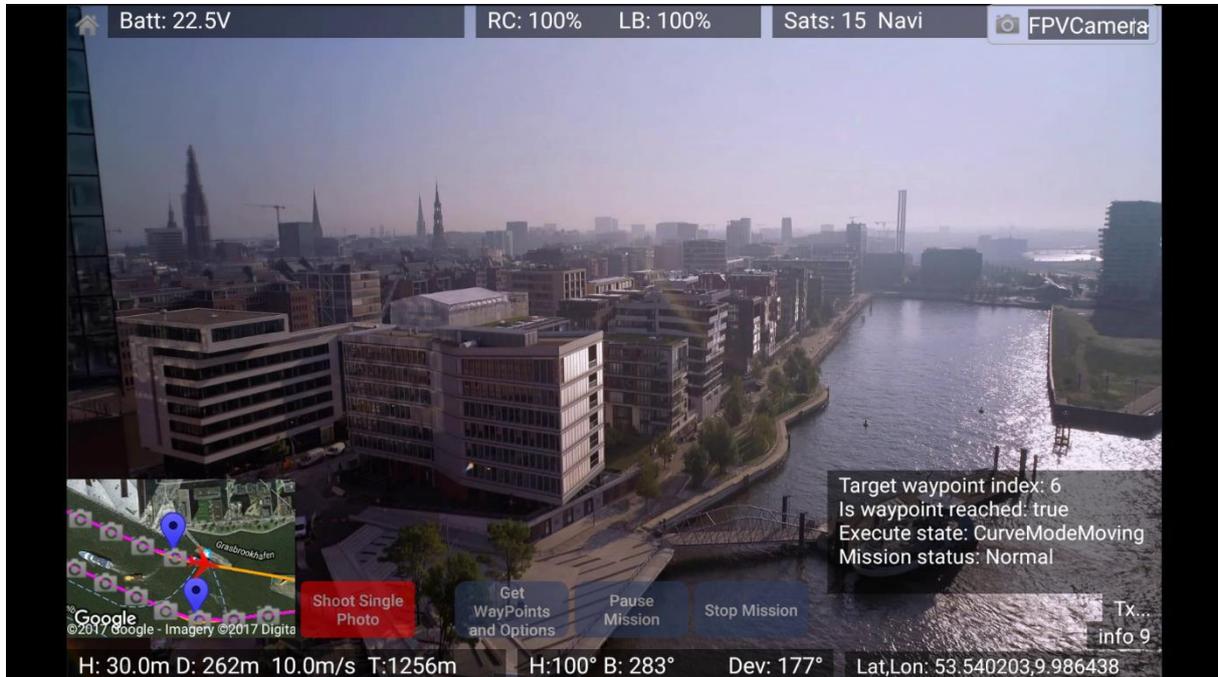
It is possible to grab data directly from additional onboard sensors (e.g. temperature) through 1-wire protocol as well as from any other device supporting USB-to-serial communication (e.g. an FTDI / CP210x USB->UART adapter or an Arduino, Teensy etc.) and send them through the transparent data transmission to the app running on ground.

*Feature-list summarized:*

- More than 12 PWM channels controllable from the wheels, buttons and switches of a LB2 (+A3/N3) / M100 / M200 / M600 remote controller
- Up to 16 channels controllable with S-Bus add-on (*available soon*)
- Up to 32 channels controllable through serial interface; app with additional buttons / sliders on the touch user interface of the connected ground app available on request
- Backward data channel for 1-wire sensors (default: DS18B20 temperature sensor) and almost any other sensors through Serial-to-USB interface (e.g. FTDI / Arduino)
- Position, status and sensor data received from drone are logged to local files on the tablet as well as optionally transmitted to a cloud service where it can be followed real-time in any browser
- Waypoint missions including curves can be prearranged in a graphical mission editor that runs on almost any device capable to display "Google Maps". Missions are sent via PulseKraft

app to the drone. Editing can happen also in advance on any Windows PC / Mac / Linux PC / iOS / Android device and mission data can be sent by mail or stored for later usage.

- During execution of a waypoint mission, PWM signals can be triggered on each waypoint and in equally spaced distances automatically in order to trigger 3<sup>rd</sup> party camera shutters or devices for agricultural applications.



*Currently not supported:*

- The app currently does not intend to provide support for camera / gimbal functions of the DJI X3 / X4 / X5 / XT camera series like ISO or focus setting
- The app currently does not provide support for enhanced configuration functions like compass calibration or video transmission bitrate configuration etc. Please use the DJI GO App for that purpose.

## 2. Installing the “PulseKRAFT” App (Android)

The “PulseKRAFT” App will be used instead the “DJI GO App”, if you want to make use of the onboard unit’s PWM and communication features. Besides controlling the onboard unit, PulseKRAFT App will provide additional features like

- uploading prearranged waypoint missions (including curves)
- enhanced mission control with integrated map view
- live display and logging of additional telemetry data
- optional mission status and sensor data transfer
- additional logging of data received from drone connected devices

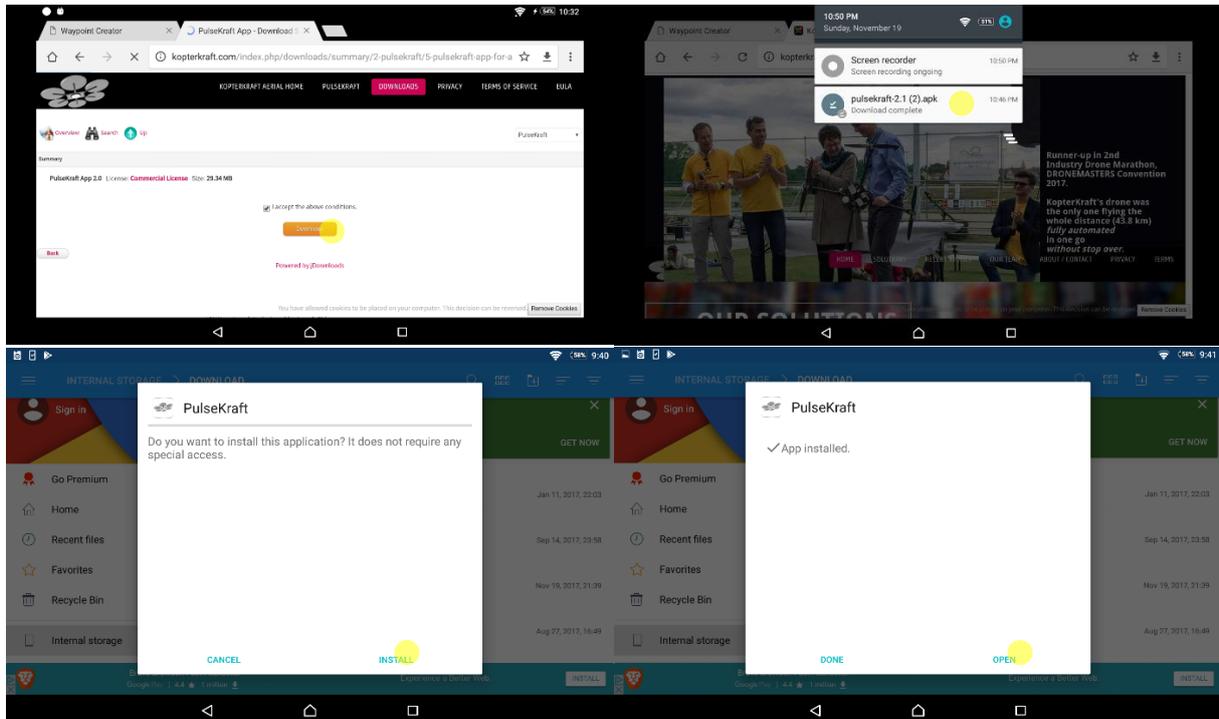
- described in detail later.

Currently this app is available for Android devices (version 5 or higher recommended). Please visit this page to download the latest version: <http://www.kopterkraft.com/index.php/downloads>

The following Android settings are required for direct download and installation:

- Installation of apps from unknown sources should be enabled in the security settings – at least temporarily, when the app is downloaded from the KopterKraft.com web site (app may be available in Google Play store later).
- USB-debugging must be enabled (in the developer options). If you previously used DJI GO app, this setting will already be enabled

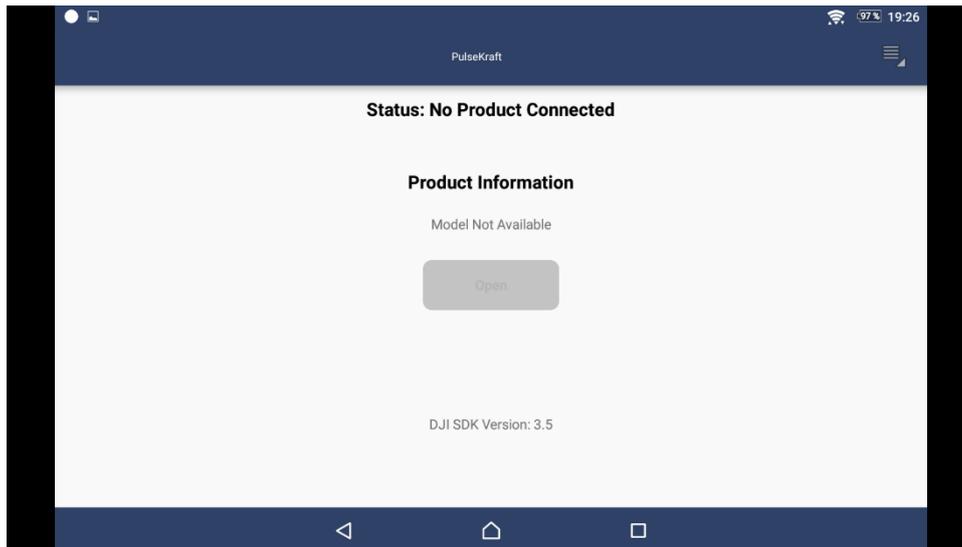
Please follow these steps that should look similar for your device:



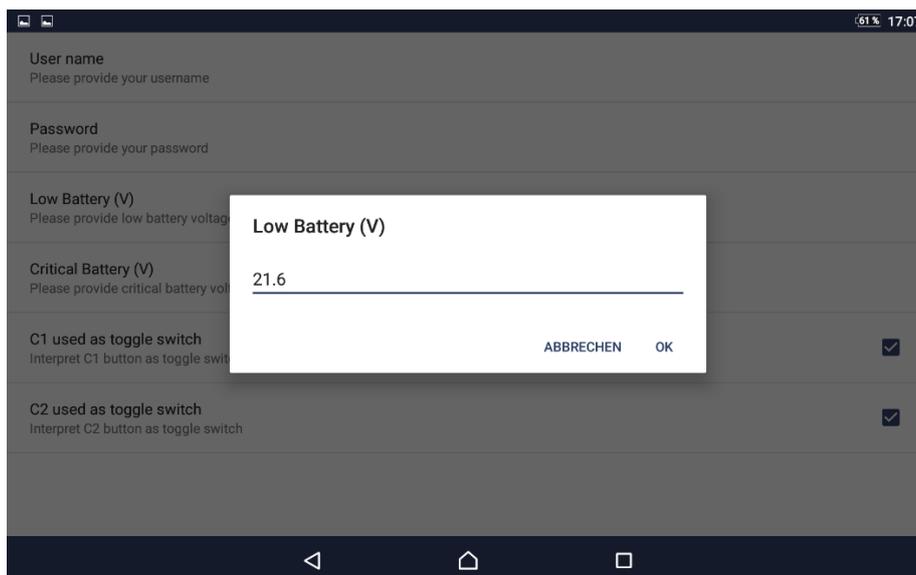
During the download, you will see a little indicator on the upper status bar. If the installation of the APK file does not start automatically, you will find it in the upper status bar and/or in the download folder (using a file manager app). If you tap on this file, you will be prompted for installation.

When you open the app for the first time, your tablet must be connected to the internet. Please **do not** connect the tablet via USB to the remote controller at the first start. As the app is using the DJI mobile SDK, it needs activation and will do some background communication with a DJI server when opened for the first time. If activation was successful, the underlying Mobile SDK may request some further permissions for localization, file system access and phone options – depending on your Android version.

When the initialization procedure has finished, you will see a start screen with some general status information:



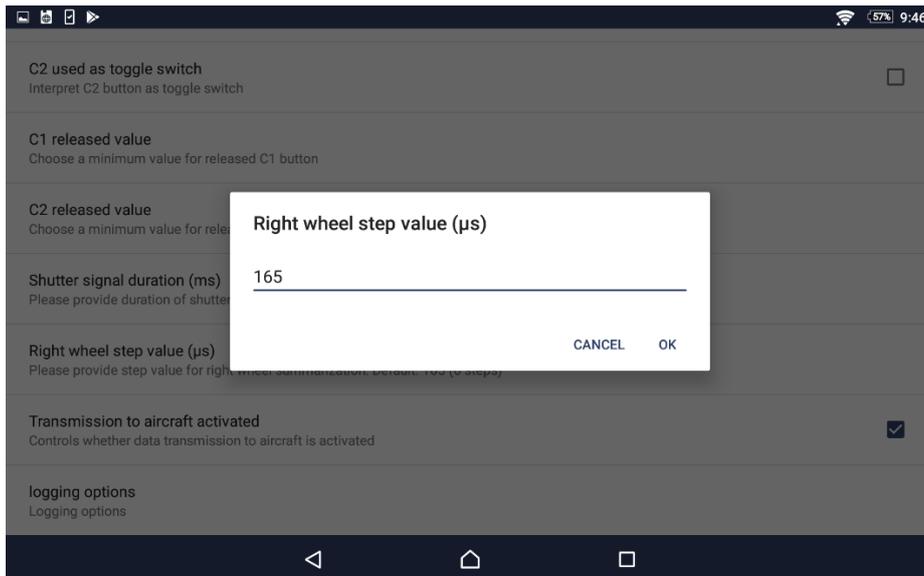
On the upper status bar on the right side you find a menu icon providing access to some preference settings. In the preferences you can set a user name and password (required if you want to use the online mission status monitor) as well as threshold values for low battery and critical battery warnings (initially contain some default values for a 6S battery setup).



Those voltage levels should match the values you may have set earlier via DJI Assistant or Go App. Additionally you can set a voltage offset that should be added to the voltage reported by your flight controller in case that this measurement is inaccurate (this offset may also be negative, e.g. -0.35).

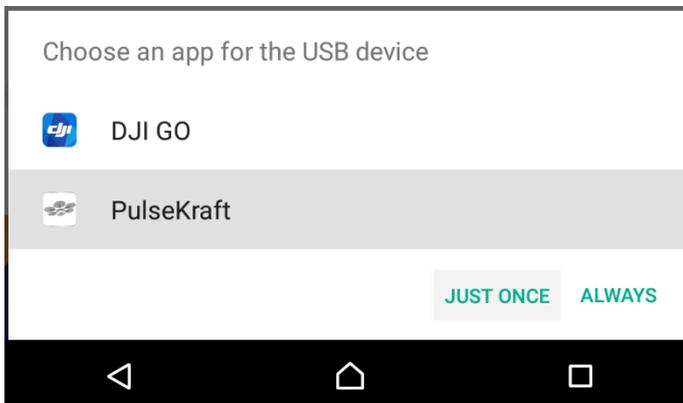
You can also configure, if the C1 and C2 buttons should be interpreted as push buttons or as toggle switches and a duration for the shutter trigger (default: 200 ms).

With the “Right wheel step value ( $\mu\text{s}$ )” you can configure the increment for the PWM signal with every step of the right wheel. Default is  $165\mu\text{s}$  which results in 6 steps (distributed between  $1000\mu\text{s}$  to  $2000\mu\text{s}$ ).



Now it's time for a test together with the DJI RC and the flight controller:

1. Make sure, the tablet is **DIS**connected from the USB port of the DJI RC handset
2. Switch on the RC handset
3. Switch on your drone (Lightbridge + flight controller).  
The PulseKRAFT Onboard Device should **not yet** be connected.  
A camera delivering a picture to the Lightbridge should be connected.
4. When the RC, the LB2 and the flight controller are linked, please plug in the USB connection between the RC handset and the tablet. You should see a pop up similar to this:

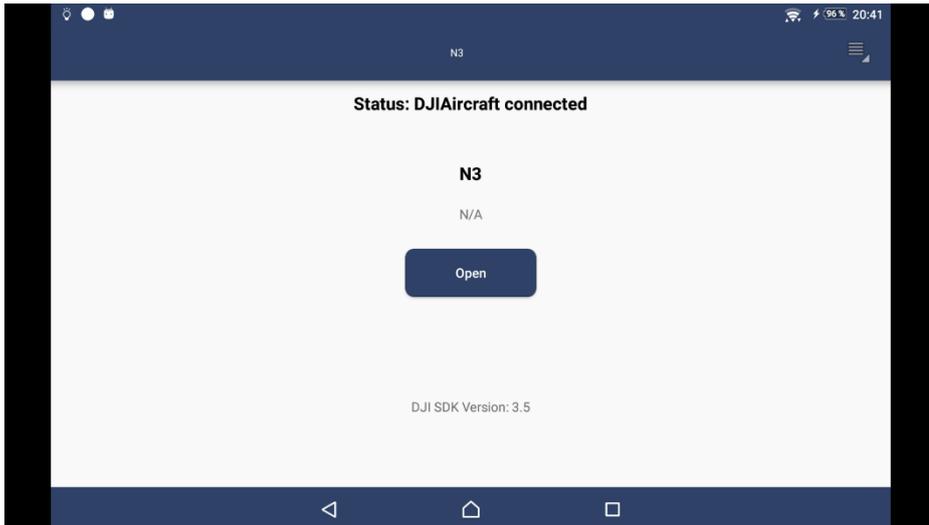


If you installed further apps compatible to the USB device (like "Litchi") you may see also these apps in the list. Please select the "PulseKraft" app (and – if you see this choice on your device as well - click on "just once").

**Note:**

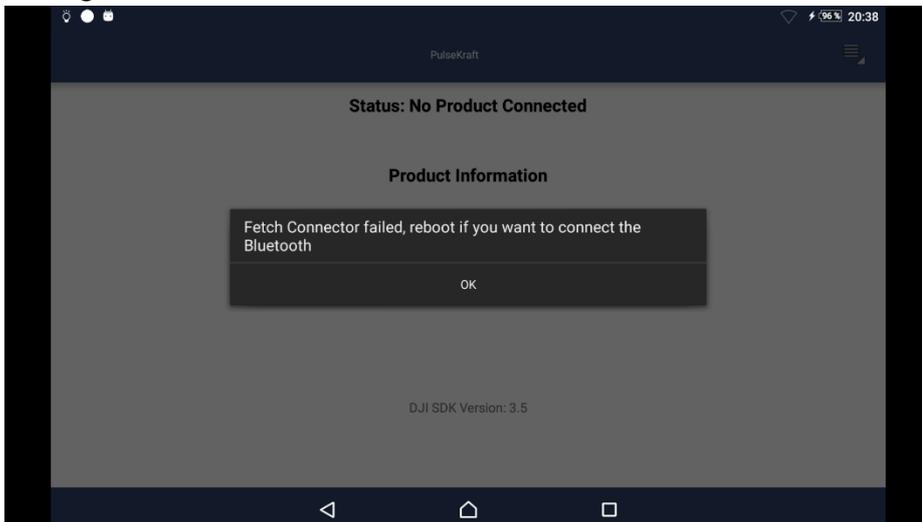
*Some manufacturers changed this behavior and configured the device to set the selected app as default automatically. E.g. Samsung Galaxy S7 and S8 and also tablet S3 will behave this way. In this case you will see no further prompt to choose on the alternative apps when connecting the USB cable to the RC. Please follow these instructions to change this default behavior: <https://www.howtogeek.com/305364/how-to-stop-the-galaxy-s7-and-s8-from-setting-default-apps-automatically/>*

5. The app should start now and show you the start screen with a detection of your drone / flight controller type like this:



The “Open” button should be enabled now as well.

If no model is detected and the “Open” button does not get enabled and instead you see a message like this:

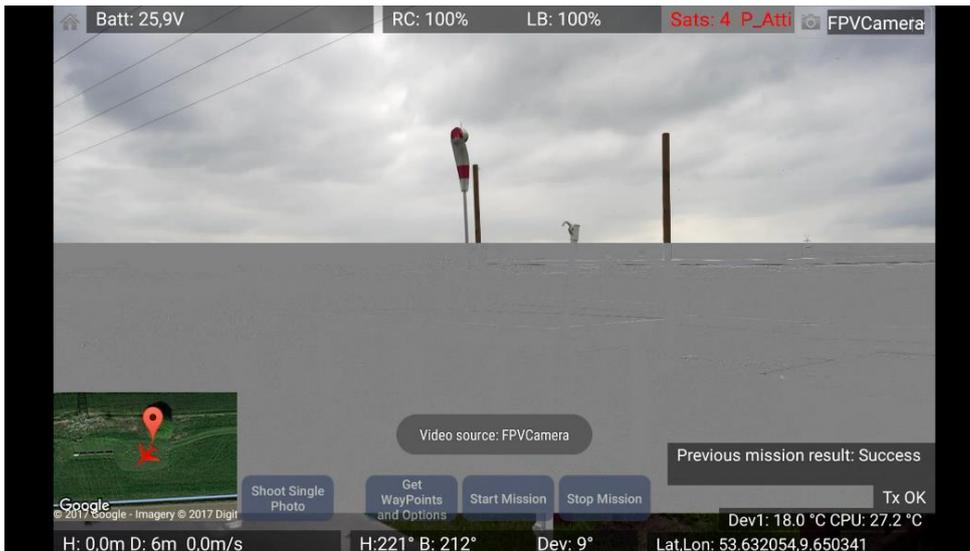


You should close the app, disconnect the USB connection to your remote controller and reboot your tablet. Make sure, the tablet has an internet connection after the reboot and reconnect the USB connection to your remote controller.

**Hint:**

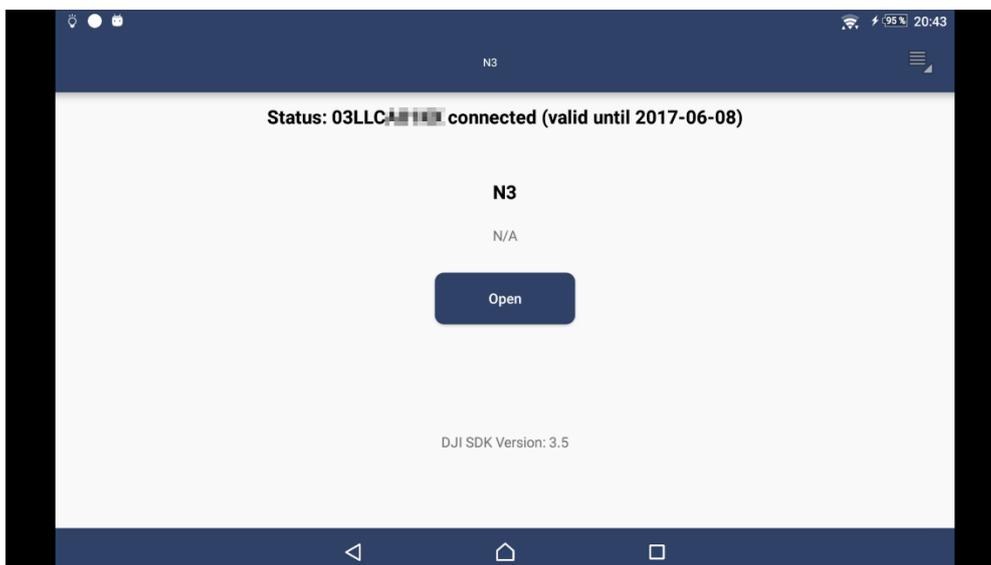
*The current version (3.5.1) of the Mobile SDK detects the drone as an “Inspire 1” when an X3 camera is connected and as “Inspire 1 Pro”, if an X5 camera is connected to the LB2. In this case, the app will not communicate correctly to the Onboard SDK (it will report a TxErr and battery level will not be reported correctly). This will be fixed by DJI in the future.*

6. After pushing the “Open” button, you should see the camera view building up a live picture from your camera (connected to the HDMI input of the Lightbridge), a map and some telemetry like this:



If you have several video sources connected to your Lightbridge (e.g. via composite AV and HDMI) You can switch between video sources with selector in the upper right corner. Please note that there is only a simple switch option for quick access. If you want to configure your video setup in more detail (like bitrate, resolution and bandwidth configuration) please use the HD configuration tab of the DJI GO app.

The first time when you push the “Open” button from the entrance screen, an activation of the PulseKraft app takes place in the background. The app will receive a temporary activation key for your particular device that is valid for a trial period of 30 days. The next time, when you connect your remote controller via USB and start the PulseKraft app, the entrance screen will show the result of your activation (serial number of your device and the validity period of this activation) like this:



For a perpetual activation, you can purchase a license key at [KopterKraft.com](http://KopterKraft.com).

Software installation and preparation is almost finished now.

**Please note** that you cannot change between the DJI GO app and the PulseKRAFT app without disconnecting and reconnecting the USB cable. Both of these apps use a background process that keeps the USB connection busy even if you already exited from the app window. When you disconnect

and reconnect the USB cable, you will see the selection pop up of compatible apps again to choose from. If you pushed “always” instead of “only once” from that pop up, next time the selected app will directly start after USB connection but we do not recommend that. You still should be able to choose between these apps each time you connect the USB cable of the DJI RC. If you already have chosen one of the app as default app earlier, you can undo that in the general Android settings. There you can go into installed apps and select (in this case undo) their default behavior.

If you have a Samsung S7, S8 or tablet S3, there may be no “only once” option and the selected app will be default for that USB device in the future. Please follow this description to change this behavior: <https://www.howtogeek.com/305364/how-to-stop-the-galaxy-s7-and-s8-from-setting-default-apps-automatically/>

### 3. Preparation of the Onboard Device (“software only” option)

If you received PulseKRAFT as “software only”, you need a Raspberry Pi minicomputer.

Recommended due to compactness, weight and price is a Raspberry Pi Zero, the “5\$ computer”.



Currently there is only a limited number of official suppliers. For example in USA it is available at “Adafruit”, in Europe / UK you can buy it e.g. through dealers “Pimoroni” or “thePiHut”. For a complete list, see here: <https://www.raspberrypi.org/products/raspberry-pi-zero/>

The new RaspBerry Pi Zero W (includes a WiFi and Bluetooth module) is expected to be available in larger quantities and can be used as well but the wireless modules will not be activated in order to avoid interferences with the drone’s radio control.

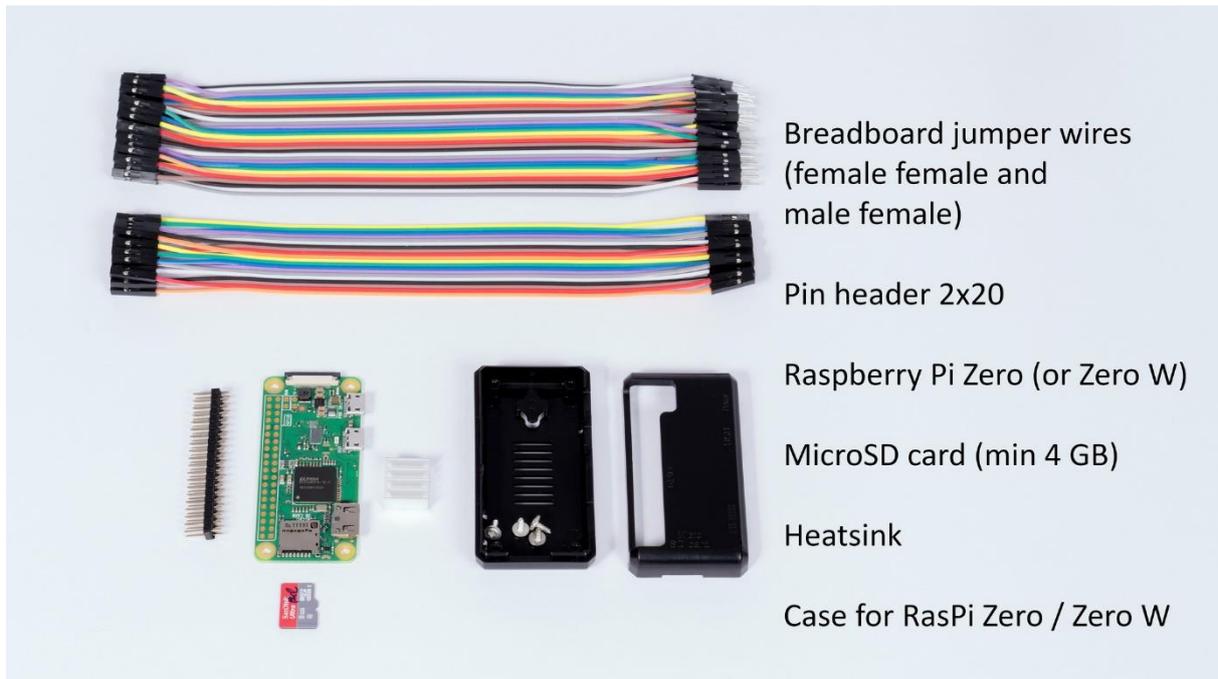
As the Pi Zero comes quite naked, you need some additional components for our use case.

*Required:*

- Raspberry Pi Zero (or Zero W or 2 B; Pi 3 can be used as well with a little config change, see below)
- a pin header (male 2x20 pins to be soldered to the Pi Zero)
- a microSD memory card, minimum 4 GB, recommended class 10
- 5V power supply (via USB or an UBEC), should be able to provide 1-2A depending on additional devices attached
- Cables (bread board jumper wires female-male and female-female, see “wiring”)

*Optionally (strongly recommended):*

- a heat sink (about 14mmx14mm or 15x13mm)
- a case providing access to the pins and enough space for the heat sink



*Nice to have (helpful in case of error analysis):*

- an adapter from mini-HDMI to HDMI to connect a display monitor
- an “OTG” adapter from micro USB to USB A (female) to connect an USB keyboard

Almost any of these parts can be found and ordered at the suppliers mentioned above. As an example, a shopping cart from Pimoroni could look like this:

<https://shop.pimoroni.com/products/raspberry-pi-zero> (RPI-011)

<https://shop.pimoroni.com/products/heatsink> (6mm, COM1202)

<https://shop.pimoroni.com/products/male-40-pin-2x20-hat-header> (COM1101)

<https://shop.pimoroni.com/products/jumper-jerky> (CAB 0005 + CAB 0006)

<https://shop.pimoroni.com/products/4gb-microsd-card> (COM2200)

(or <https://shop.pimoroni.com/products/noobs-8gb-sd-card> containing an SD card adapter)

If you do not already have a stable 5V power source, we recommend voltage regulators from Pololu, like <https://www.pololu.com/product/2850>

### **Onboard software download:**

Latest software and documentation can be downloaded here:

<http://www.kopterkraft.com/index.php/downloads>

The image file for the onboard device is provided ZIP compressed and needs to be uncompressed before installation. Please make sure, you have about 1.2 GB disk space available for the img file in the target folder. This will be needed only temporary.

In order to write the image file to a microSD memory card, you will need a card reader that can handle microSD cards or an adapter that adapts the microSD card to the form factor of a SD card. Usually microSD cards come packaged together with such an adapter.

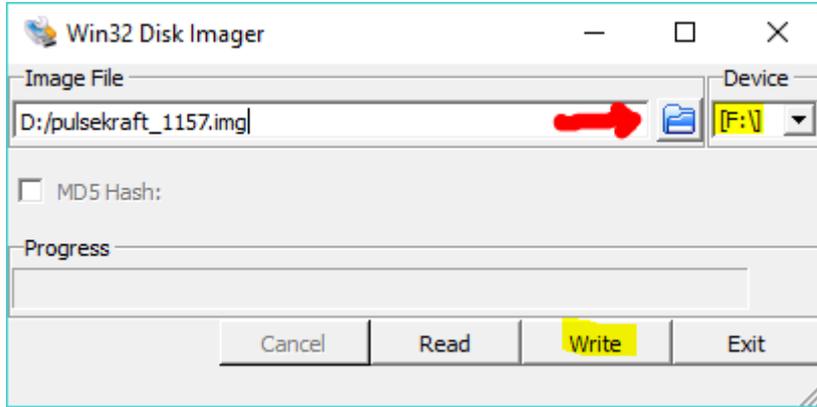
In Windows you can use “Win 32 Disk Imager”, available here:

<https://sourceforge.net/projects/win32diskimager/>

Under Linux you can use “dd” and on Mac OS you can use “Etcher” (see <https://etcher.io>).

Please make sure that the target device drive is the SD card – the image writing software will overwrite data on that volume irreversible.

*In this example, it is drive F: but it may be assigned to a different drive letter on your PC.*



After writing the image, you should see a volume (drive) named “boot” in Windows explorer.

If you have the adapters available to connect a HDMI monitor and a keyboard, you can make a quick check now if the image is executable. When the Pi Zero is powered on with inserted microSD card, you will see some boot messages until it reaches a login prompt and an endless loop trying to connect to the flight controller. Now you can disconnect power again. There will be cleaner methods for shutdown described later but as the system is prepared to run in a read-only mode, there is no hassle.

If you want to use the image on a Raspberry Pi3 (B or B+), then the config.txt file on the boot volume needs a little modification because Raspi 3 assigns the serial UART to the Bluetooth module by default. When the microSD card is inserted in a Windows PC, you see the boot volume as a separate drive letter. At the end of the file config.txt please add the line:

**dtoverlay=pi3-disable-bt**

It is recommended to use a text editor that supports Unix style line breaks (e.g. Notepad++).

#### 4. Wiring the Onboard Device

For a quick setup, the onboard device needs the following connections (e.g. using breadboard jumper wires female-male / female-female):

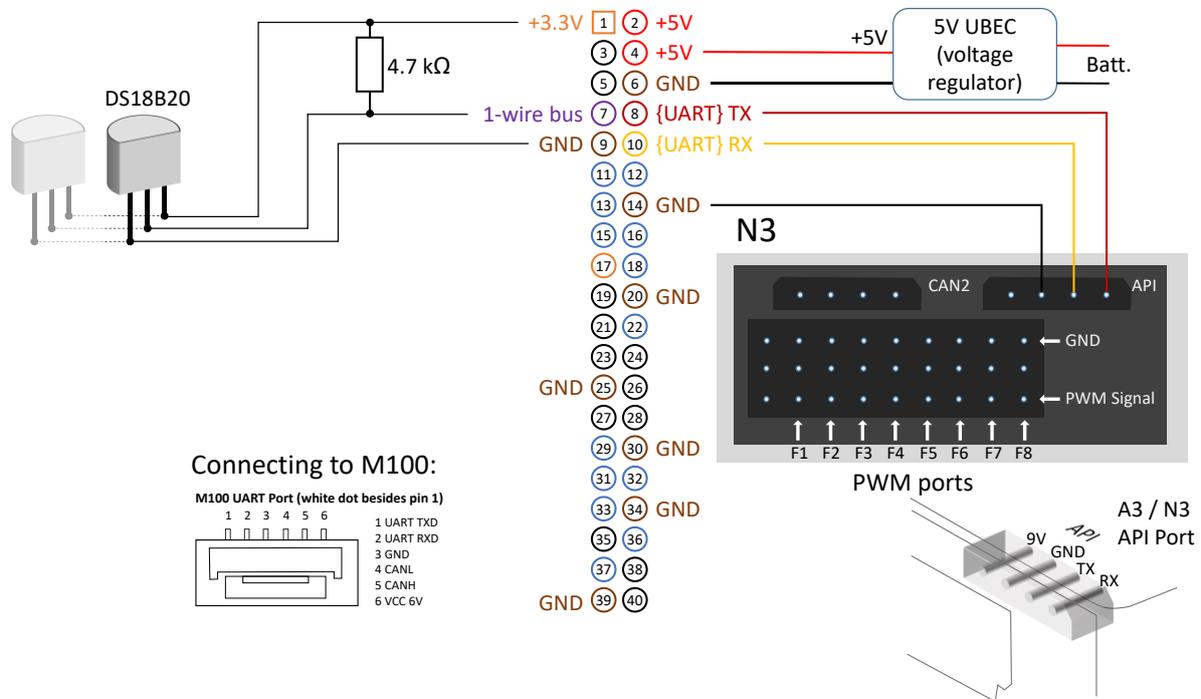
- 5V power supply (e.g. from an UBEC or from USB) that should be able to provide 1-2A (depending on further connected devices)
- Serial connection to the API connector of the flight controller (A3 or N3) or the UART/CAN2 port of the Matrice 100 (please use the UART cable that is provided with the Matrice 100). Please note that the TX and RX pins of the API port and the onboard device are cross-wired.

You may also consult the hardware setup guide from DJI, which can be found at <https://developer.dji.com/onboard-sdk/documentation/development-workflow/hardware-setup.html>

The PulseKraft solution will support Matrice 200 series soon, the required connection port is mentioned in the M200 series manual as “OSDK port” in section “Expansion Ports”. The diagrams in DJI’s hardware setup guides indicate that you need an UART-USB converter between M200 and Linux

Onboard computer. This is not required when you connect the RaspBerry Pi as recommended for PulseKraft solution, because it has Rx, Tx and GND pins for direct connection (as shown in the wiring diagram below).

The following diagram shows the layout and assignment of the 40 pins of the onboard computer's pin header as well as some typically connected devices:



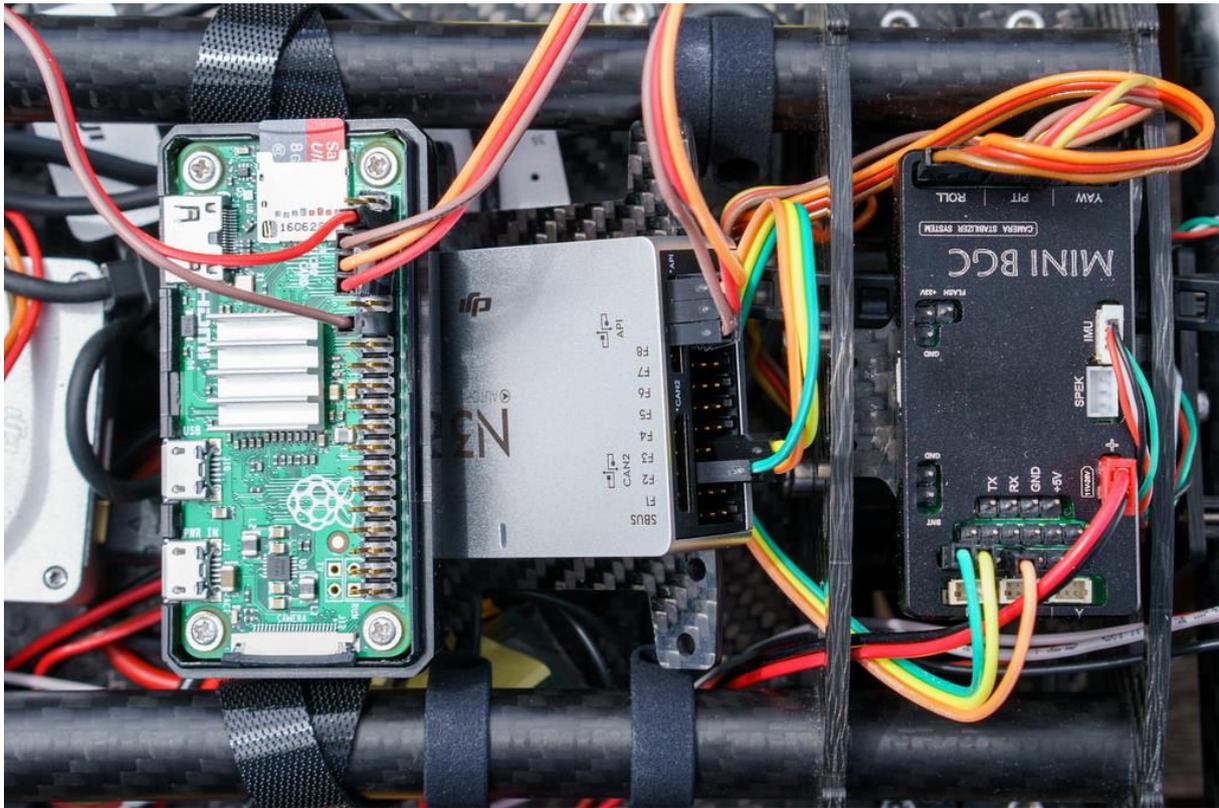
If you provide power to the onboard device from a USB port, please connect it to the micro USB connector of the onboard computer labeled with “power”. If you provide 5V from an UBEC, please connect – (GND) to one of the pins labeled as “GND” (pins 6, 9, 14, 20, 25, 30, 34, 39) and “+5V” to one of the pins labeled +5V (pin 2 or 4). Please do **NOT** connect external power to the +3.3 V output pin. This is intended to be used only for some lower power devices like sensors.

**WARNING 1:** Please note that the F-port pins in the middle between GND and PWM signal do not provide +5V voltage on the A3 / N3 controller. If you want to drive servos or a landing gear, it is strongly recommended to provide a separate 5V power supply as they may consume high current when moving heavy load or when blocked. If they share the 5V power supply of the onboard computer, it may hang, shutdown or do other unpredictable things at voltage drops. Voltage peaks may destroy the onboard computer. It is also **not** recommended to use the 6V power output of the M100 UART connector or the 9V output of the N3 / A3 API connector.

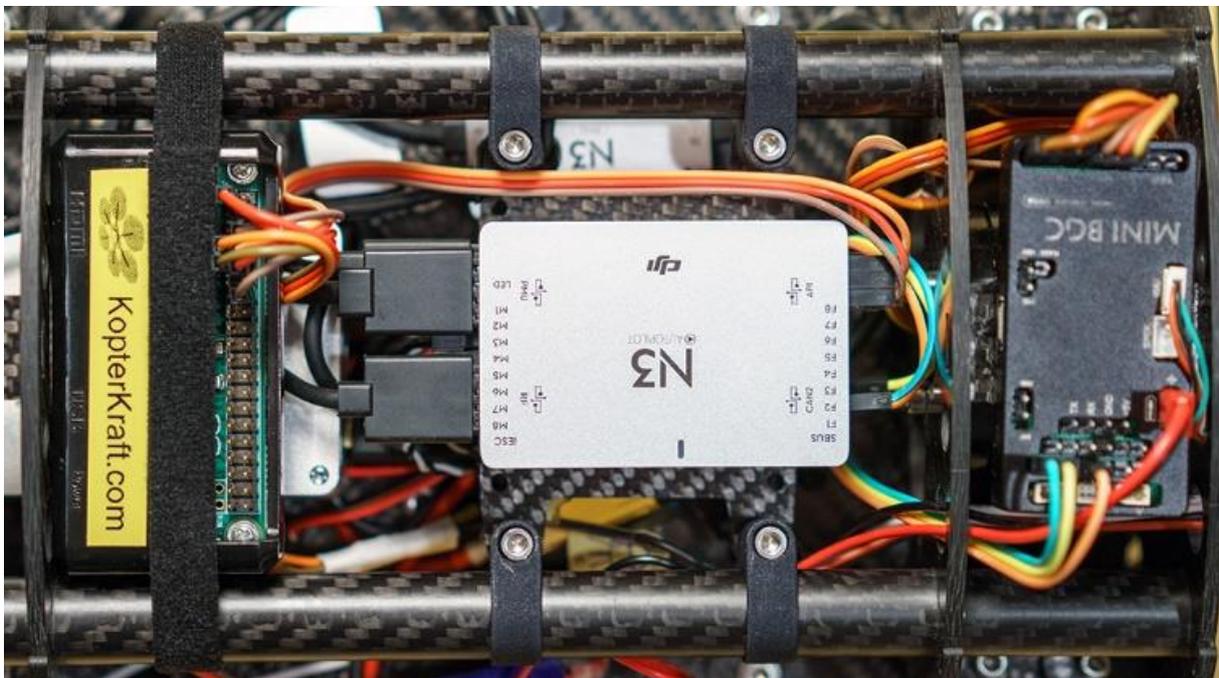
**WARNING 2:** The I/O pins of the onboard computer as well as the RX/TX pins of the UART connector of the flight controller are sensitive as they are directly connected with the internal micro controller chips. They are **not** 5V tolerant, voltages above 3.3V on the signal pins will damage the controller. Be careful and double check the wiring. We accept no liability for any damage or injury incurred from wrong wiring.

When you connect the RX / TX pins from the onboard computer to the UART / API connector of the flight controller, please make sure that the RX pin (10) is connected to the TX pin of the flight controller and the TX pin (8) of the onboard computer with the RX pin of the flight controller.

The following picture illustrates an example setup using DJI N3, Lightbridge 2 and an Alexmos 32 bit SimpleBGC gimbal controller (connected to F2 and F3 ports):



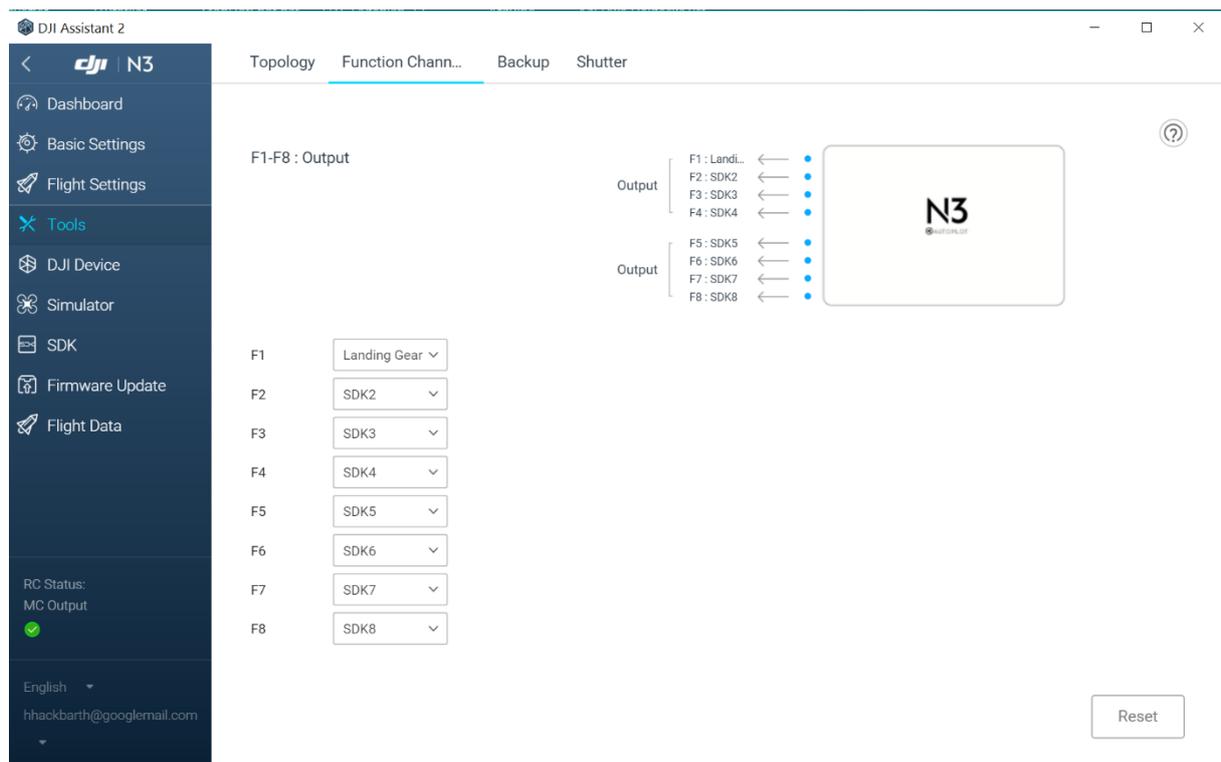
Final setup (including wiring to a DS18B20 temperature sensor):



(See another video showing the solution here: <https://www.youtube.com/watch?v=Uh30eD15li8> please note that this video shows wiring based on an earlier hardware version, where the PWM devices were connected to the PulseKraft onboard device)

If you want to control a 3<sup>rd</sup> party gimbal or a camera controller (like CAMremote or gentWIRE) through PWM channels, you need to connect only one of the PWM pins from the onboard computer to one of the signal pins of the gimbal controller. GND is most probably shared already through the power connections. In other words: You should **not** share a 5V connection between the onboard computer and the peripheral controllers and one GND connection should be sufficient. E.g. if you want to control a 3-axis gimbal, 3 wires for the PWM signal lines should be sufficient.

In order to be able to use the F-ports of the A3 / N3 flight controller, you need to map them in the DJI Assistant:



**NOTE:** It is recommended to let the F1 port in the default setting mapped to control a landing gear. This makes sure, that it will also be activated in case of a fail-safe / auto landing, if you configured it that way in the DJI Assistant.

If you cannot map all 8 F-ports, you probably use version 1.1.6 or higher of DJI Assistant 2. Version 1.1.2 (and 1.1.2\_2) were able to map all 8 F-ports as SDK output ports. Download is still available at the time of this writing at

[https://dl.djicdn.com/downloads/dji\\_assistant/20170619/DJI+Assistant+2+1.1.2\\_2.zip](https://dl.djicdn.com/downloads/dji_assistant/20170619/DJI+Assistant+2+1.1.2_2.zip)

As the Matrice 100 (and also Matrice 200 series) do not provide F-Ports onboard, the add-on solution described in chapter 11 will be required to get the desired PWM output on these models.

If you want to use devices that can only be controlled using a SBus connection, you can either use a PWM-> SBus converter (e.g. the "RMILEC High-Precision PWM/PPM/SBus Signal Converter V2") or wait until the SBus add-on for the PulseKRAFT solution becomes available.

If you plan to connect one or more DS18B20 digital thermometer sensors, you need a 4.7 kΩ pull-up resistor connected between 3.3V (pin 1) and pin 7 (the I/O port for 1-wire bus sensor devices) as indicated in the wiring diagram.

At the moment, the solution only supports DS18B20 sensors directly but there will be an add-on later providing an interface for lots of other sensors.

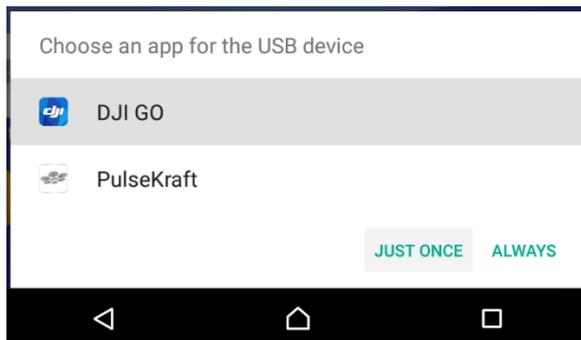
## 5. Activation of the Onboard Software

Like for the ground app, it is also required by DJI for software based on their Onboard SDK to be activated. Initial activation can be done *only* through DJI's GO app.

Prerequisite is that you enabled the API on the flight controller using the DJI Assistant and that you set the baud rate to the default value of 230000 baud.

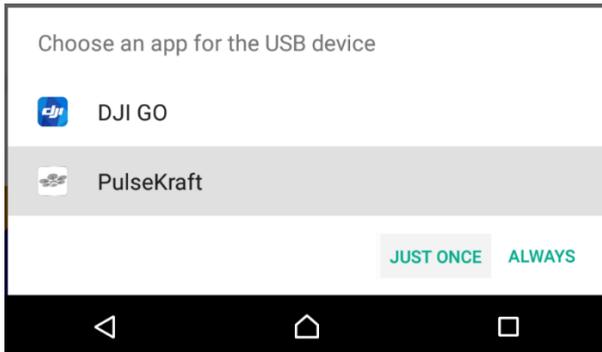
After API is enabled in DJI assistant, please follow these steps:

1. Connect your tablet to the internet
2. Switch on the remote controller, set the little mode switch (on the upper left side of the remote controller) to "F" (Matrice 100, N3) or "P" (A3, N3).
3. Connect the tablet via USB.



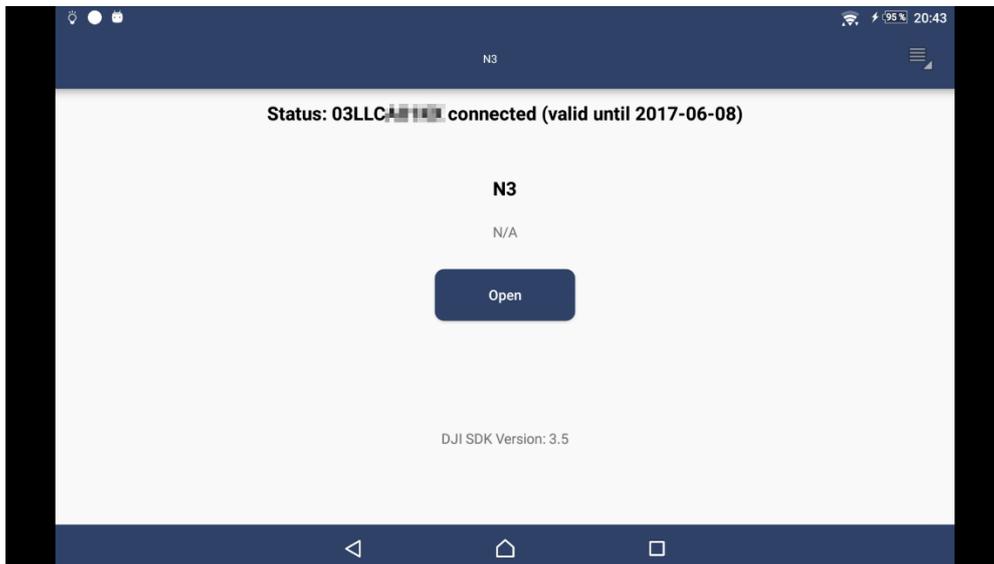
As there is more than one app able to handle the connected remote controller (as USB device), Android will let you choose from compatible apps. If you installed further apps like "Litchi", you will probably see even more options to select from. Please select the DJI GO app, but "only once". *If you have a Samsung S7, S8 or tablet S3, there may be no "only once" option and the selected app will be default for that USB device in the future. Please follow this description to change this behavior: <https://www.howtoqeeek.com/305364/how-to-stop-the-galaxy-s7-and-s8-from-setting-default-apps-automatically/>*

4. Power on the aircraft's flight controller (with "PulseKRAFT" onboard computer already connected like described in chapter "Wiring"). The PulseKRAFT device needs about 20 seconds to boot and to load its software.  
*(It can be helpful to monitor the HDMI output of the onboard computer, if you have a display available. You can also connect the HDMI output of the onboard computer to the HDMI input of the Lightbridge if you have a cable or adapters from Mini-HDMI to Micro HDMI available.)*
5. When the GO App detects the connected flight controller, please enter the "camera" view of the GO App. Wait for about 1 minute for the activation to go through.  
If you have a display connected to the HDMI port of the onboard device, you can see, if activation was successful.
6. Leave the camera view of the DJI GO App and exit the app.
7. Disconnect the USB cable from the remote controller
8. Reconnect the USB cable and select again the app that should handle the device. This time select the "PulseKraft" app...

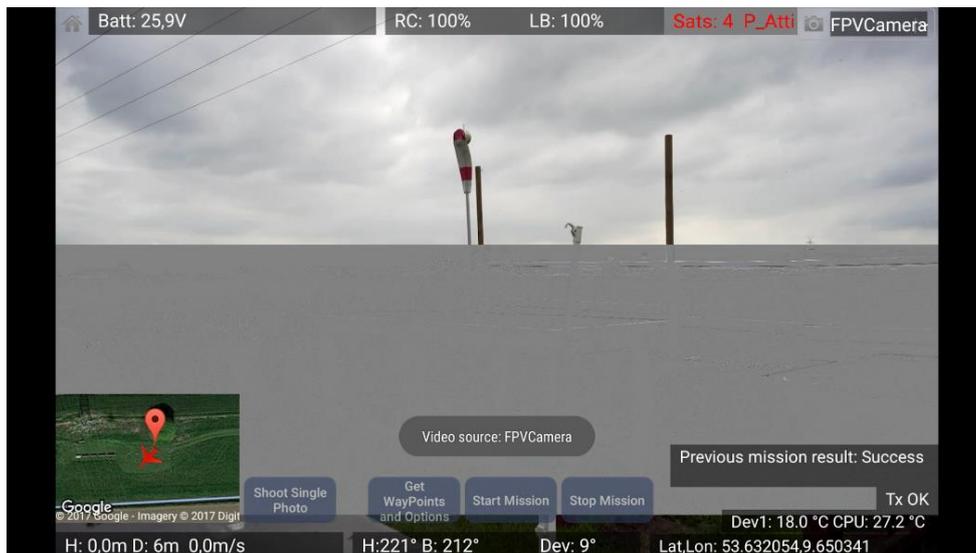


...and tap on “only once”.

Now the PulseKraft app should open and show a connection to your flight controller (in this example the **N3**):

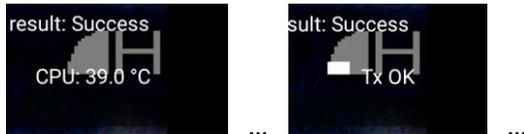


This time the “Open” button should be enabled and can be tapped to enter the camera view:



In the camera view, you should see the live view of your onboard camera (connected to the Lightbridge 2). If you do not see the live view, please go back to the “DJI GO App” and check the HD configuration for the Lightbridge.

On the bottom right side you should see a little indicator for the status of the data transmission between the PulseKraft app and the onboard computer as well as the CPU temperature of the onboard device sent back via the data connection (in this example it shows “CPU: 39.0 °C”). If you have further sensors connected, you will also see their data:



#### Troubleshooting:

If you see “TxErr: Not Support” or if you only see “Tx OK” but no temperature from the backward channel, there was a communication problem during the startup phase. If you see “Tx OK” but no CPU temp, it may be sufficient to push the “play” and the “record” buttons together for a short moment. This will cause the onboard software to restart (if it was able to receive that command).

If this does not help, you should leave the live camera view (with a swipe gesture from the bottom of the screen to see the backward button) and exit the PulseKraft app. Now, disconnect the USB cable from the RC and wait for *at least* 10 seconds (this will trigger a “watchdog” on the onboard device forcing it to reinitialize). Now reconnect the USB cable and again choose the “PulseKraft” app from the list of compatible apps.

After a short moment you should see the start screen with the connection status for the detected DJI drone controller (A3, N3, M100, M600 etc.) and the “Open” button should be enabled. Push that button to enter the live camera view again. Now you should see the temperature(s) and “Tx OK”. If this still does not help, please see the hints in the “Troubleshooting” section.

## 6. Taking Control

After an initial wiring (according to chapter “Wiring”) and activation, you should be able to control PWM devices now. It is strongly recommended to change wiring only when power is disconnected. The following table explains the generated signals according to RC actions in more detail with default channel configuration:

Channel - > F-port	RC stick/wheel/button	Operation
Ch. 0	-	Internal use.
Ch. 1	Left stick (vertical)	Throttle (controlling elevation)
Ch. 2	Left stick (horizontal)	Rotation (rudder)
Ch. 3	Right stick (vertical)	Forward / backward pitch
Ch. 4	Right stick (horizontal)	Left / right pitch
Ch. 5 -> F2	Left shoulder wheel	Continuous control for PWM servo pulses between minimum and maximum values (default: 1000 and 2000 $\mu$ s, center position generates 1500 $\mu$ s pulses) (configurable)
Ch. 6 -> F4	Right shoulder wheel summarized	As the right wheel physically produces only short pulses, this output sums up pulses to generate a PWM signal in 8 steps between 1000 and 2000 $\mu$ s (configurable). Initially, this output generates 1500 $\mu$ s as a “center position”. From there you have 4 steps in both directions to reach either the lower

		or the upper limit. Further steps in the same direction after reaching the limit will have no effect.
Ch. 7	Landing gear switch	Switching a PWM servo signal between minimum and maximum position (1000 and 2000 $\mu$ s, configurable). If you plan to use it for a landing gear, please make sure the controller interprets it correctly.  <i>WARNING: As we take no responsibility / liability for malfunction of the onboard device, we DO NOT recommend to use this output for a landing gear. Instead you should use the F1 port of your flight controller which includes failsafe functionality.</i>
Ch. 9	Go-Home button	Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed and 1500 $\mu$ s (like a stick in center position, configurable) when released.  <i>When playback button is pressed before and together with Go-Home button, the onboard device will be safely shut down. Use this combination only, when aircraft is on ground.</i>
Ch. 10 -> F6	Record button	Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed and 1500 $\mu$ s (like a stick in center position) when released.  <i>When playback button is pressed before and together with record button, the onboard device will stop and restart. Use this combination only, when aircraft is on ground.</i>
Ch. 11 -> F5	Shutter button	Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed and 1500 $\mu$ s (like a stick in center position) when released.
Ch. 12	Playback button	Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed and 1500 $\mu$ s when released.  Additionally it is interpreted as "shift" key together with left wheel, record button and Go-Home button.
Ch. 13 -> F7	C1 button	Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed and 1500 $\mu$ s (like a stick in center position) when released.
Ch. 14 -> F8	C2 button	Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed and 1500 $\mu$ s (like a stick in center position) when released.
Ch. 15 -> F3	Shifted left wheel	When playback button is pressed together with operation of the left wheel, it generates continuous control for PWM servo pulses between 1000 and 2000 $\mu$ s (configurable), center position generates 1500 $\mu$ s pulses on this pin. The signal of the last wheel position will be kept, when the playback button is released. This can be used to memorize a certain position until the "shift" (playback) button is operated together with the left wheel again.
Ch. 16	Pushed right wheel summarized	Like "right wheel summarized" but on this output the signal changes when the right wheel is operated in "pushed in" position.

Ch. 21	Right wheel pushbutton	This pin only indicates if the right wheel is pushed in or released. Generates a maximum PWM servo pulse (2000 $\mu$ s, configurable) when pushed. 1500 $\mu$ s (like a stick in center position) when released.
Ch. 22	Right wheel pulse	This pin generates short pulses when the right wheel is turned. Pulses are 1000 $\mu$ s (configurable) when wheel is turned downwards (clockwise) and 2000 $\mu$ s (configurable) when wheel is turned upwards (counterclockwise). Between the maximum / minimum pulses, the PWM servo pulses are 1500 $\mu$ s (like a stick in center position).

#### a) Channel mapping

In order to map these channels to “GPIO” pins (F-ports of the A3 / N3 flight controller), the PulseKraft onboard software uses a configuration file **UserConfig.txt** that can be found in the boot volume of the microSD card. If you can live with the Channel -> F-Port mapping as shown in the table above, you do not need to edit this file and can use it with default values.

If you would like to change the mapping, you can insert the card (using an adapter) into the card reader of your PC and edit the file with an editor like Notepad++ or WordPad that is able to handle UNIX text files (containing only ‘\n’ characters as line break). Default is:

```
ChannelGpioMap:-1 -1 -1 -1 -1 1 3 -1 -1 -1 5 4 -1 6 7 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

Each value stands for an index of a F-port starting at 0 for F1 port (0 = F1, 1 = F2, 2 = F3, ..., 7 = F8). -1 means that this channel is not mapped to an F-port. In this list you find 32 values assigned to the channels described in the table above starting with channel 0.

#### Examples:

- The 6<sup>th</sup> value in that list is “1”. This means that channel 5 (remember: counting starts at 0) representing the left shoulder wheel is assigned to F2 port.
- The 7<sup>th</sup> value in that list is “3”. This means that channel 6 representing the right shoulder wheel (as summarized value) is assigned to F4 port.
- The 16<sup>th</sup> value in that list is “2”. This means that channel 15 representing the shifted left shoulder wheel (wheel turned together with pressed play button) is assigned to F3 port.

It is also possible to map two buttons together to one PWM port (e.g. if you want to drive a zoom function with C1 and C2) and let these buttons create different PWM pulse widths. Let’s say you want to map C1 as well as C2 to F7 port on your A3 / N3 flight controller. In that case you would set the mapping to:

```
ChannelGpioMap:-1 -1 -1 -1 -1 1 3 -1 -1 -1 5 4 -1 6 6 2 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
```

In order to assign them different PWM pulse widths when pressed, please see the following section “Adjusting the PWM signal range”. The F-port will have the value according to the latest status change of the C1 and C2 buttons. You can do this with “Record” and “Shutter” buttons as well.

If you use the PulseKraft app connected to a slave RC **in dual operator (master/slave) mode**, you can also map the main sticks (usually used to control throttle, yaw etc.) to drive the PWM ports e.g. in order to control a gimbal. Please make sure, the activation of the app happened when connected to the slave RC. You may see some delay between stick movement and execution onboard of the aircraft as the master/slave connection adds some latency to the transmission of the PulseKraft command messages.

## b) Adjusting the PWM signal range

The minimum and maximum PWM pulse width (duty cycle) can be configured for each channel as well. Default values are 1000  $\mu$ s for minimum pulse width, 2000  $\mu$ s for maximum pulse width and 1500  $\mu$ s for middle position.

If you want to change these values, you find them as well in the **UserConfig.txt** file in the boot volume of the microSD card. If you put the card (using an adapter) into the card reader of your PC, you can edit them. Default values are:

```
ChannelMinPWM:1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000  
1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000
```

```
ChannelMaxPWM:2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000  
2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000
```

Each figure stands for a particular channel counting from 0 to 31, the values must be separated by blanks. E.g. if you want to change the values for the left shoulder wheel, which is channel 5 (according to the table above), you should edit the 6<sup>th</sup> value in the list.

The value for the middle position will be half between the minimum and maximum value (1500  $\mu$ s by default).

If you want to control more than 7 PWM channels, you can use an add-on device described in the according chapter.

## 7. Restart / Safe shutdown

As long as you have a Tx connection to the onboard device, you can restart the software by pressing the playback button of the remote control together with the record button. This may help in case the downward data connection was not initialized properly (e.g. you see no CPU temperature).

As the software boots the partitions in a read-only mode, you simply can disconnect the power for a hard shutdown without the risk to corrupt the data on the microSD card.

Nevertheless, you can safely shutdown the onboard device by pressing the playback button together with *short* push on the Go-Home button.

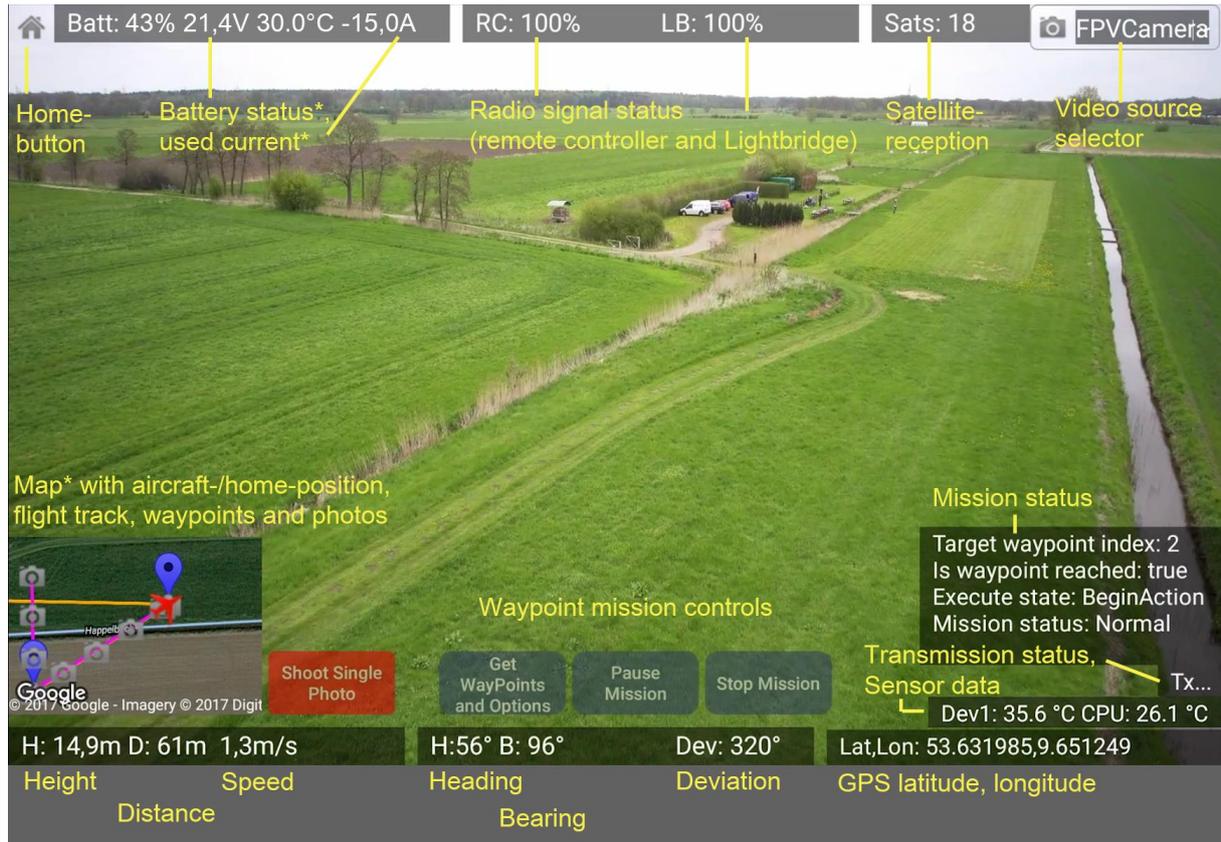
*Warning: Do not use this function as long as the drone is in the air as it may be misinterpreted by the drone when you press the Go-Home button too long. After shutting down the onboard device, you will not have control over the PWM outputs until the onboard computer is power cycled.*

The safe shutdown feature can be useful if you have further devices connected to the onboard computer. If they listen for the safe shutdown command as well, they can finish their operation (e.g. a recording) safely.

## 8. Additional App Features

The PulseKRAFT app displays some telemetry, you may already know from other apps and it provides some additional status and telemetry information as well as a map and mission control features.

### c) UI Elements and Enhanced Telemetry

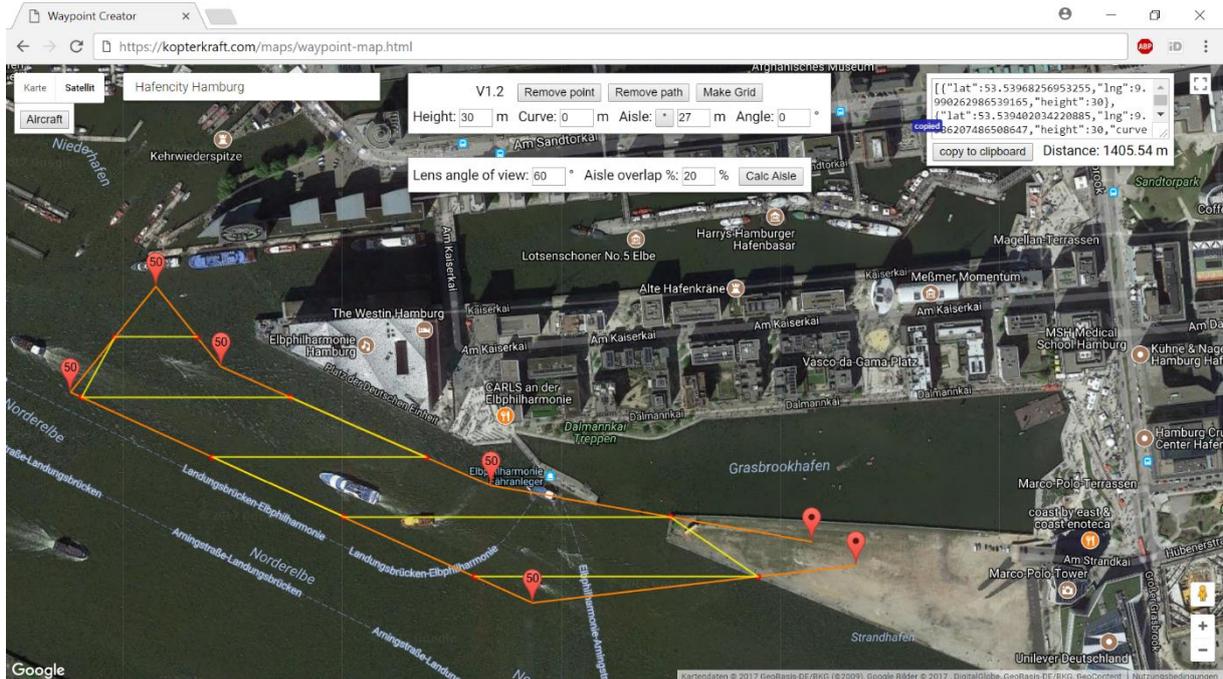


\* Enhanced battery status and used current information is only available with DJI intelligent batteries. The battery status will turn yellow, when low battery level (or 30% charge) is reached. The battery status will turn red when critical low battery level (or 10% charge) is reached. Please be aware, that these levels are handled by the app separately from the drone controller's configuration that you may have set through DJI Assistant or DJI GO App. There may be also different levels configured in your drone for failsafe / return-to-home (RTH) and for immediate landing at critical low battery. If you use standard batteries, you will see only the total battery voltage. The low and critical voltage levels for the app's display colors can be configured in the preferences accessible through the menu icon on the app's start screen.

- **Heading** means the "compass" direction of the drone itself. 0° (= 360°) is north.
- **Bearing** means the direction of the visual line of sight from pilot to drone (the "compass" direction you look, when you watch the drone)
- **Deviation** means the difference between heading and bearing. If deviation is 0° this means, the drone is flying straight away from the pilot. A deviation of 180° means, the drone is flying straight towards the pilot (e.g. at a "Go-Home").
- **Lat,Lon** are latitude and longitude of the drone's current GPS coordinates (only available when the GPS antenna receives enough satellites)

## d) Mission Control

In order to upload a prearranged waypoint mission, the mission can be prepared in the mission editor

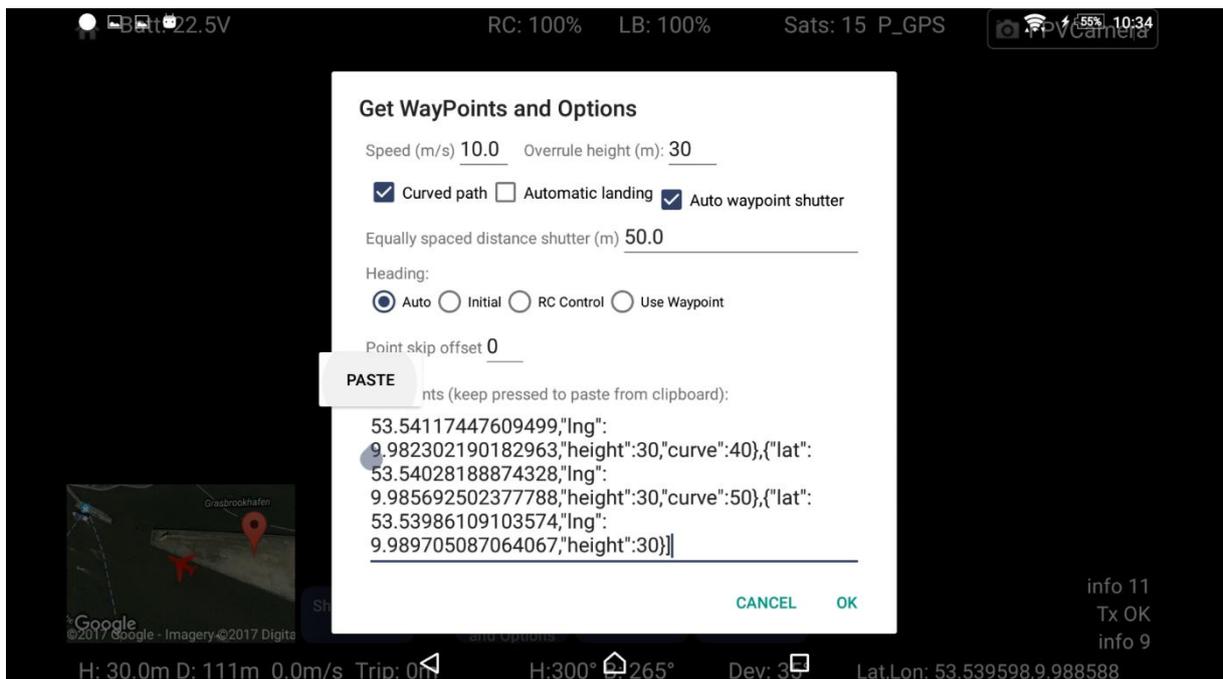


...explained in the next chapter “Waypoint Mission Editor” (chapter 9).

If the mission is planned on the tablet, you can directly take it over into the app via the clipboard. If the mission was planned elsewhere, you can for example send the mission data by E-mail to the tablet and copy it from the E-mail.

Assuming, you have the mission data already copied into the clipboard, you tap on the screen button “Get waypoints and speed”. It will open a pop up a dialog with further mission options. If required, you can overrule the height that was previously defined in the waypoint editor now.

In the text field for the mission data you can paste the data previously copied to the clipboard (touch and hold your finger on the input field until the paste option pops up).



The mission waypoints will be shown on the app's map as well. The map size can be maximized and reduced by simply tapping on the map.

The speed that you set in the options dialog can be overruled during the mission execution using the command sticks. You can even stop and reverse the movement of the drone during mission execution using the sticks of the remote controller.

When you push the "Start Mission" button after the motors of the drone already have been started, the drone will directly go to the first mission waypoint. It is your responsibility that nothing is obstructing the flight path. After the mission has been started, the button toggles between a "Pause" button and a "Continue" button.

You can see a video demonstrating the map and mission features here:

<https://www.youtube.com/watch?v=q4r9RsyoWCA>



The mission can be stopped using the "Stop Mission" button (or by switching the mode switch from 'F' to 'P' (Matrice 100) or from 'P' to 'A' (A3 / N3)). When it reaches the last waypoint, the drone hovers at the last position waiting for stick commands or lands automatically, when you checked the "Automatic landing" option.

During the mission, the PWM channel assigned to the shutter button is automatically "fired" at each waypoint. This allows to trigger the shutter also for 3<sup>rd</sup> party cameras when linked to the shutter PWM channel.

If you checked the option "Automatic waypoint shutter" and entered a valid value in the text field "Equally spaced distance shutter (m)", then the drone will trigger the shutter additionally whenever its position changed the specified number of meters from the last trigger point.

This option may be useful also in agriculture applications where you want to activate a certain action on the drone in equally spaced distances (e.g. dropping Parasitic Wasps to fight the European corn borer).

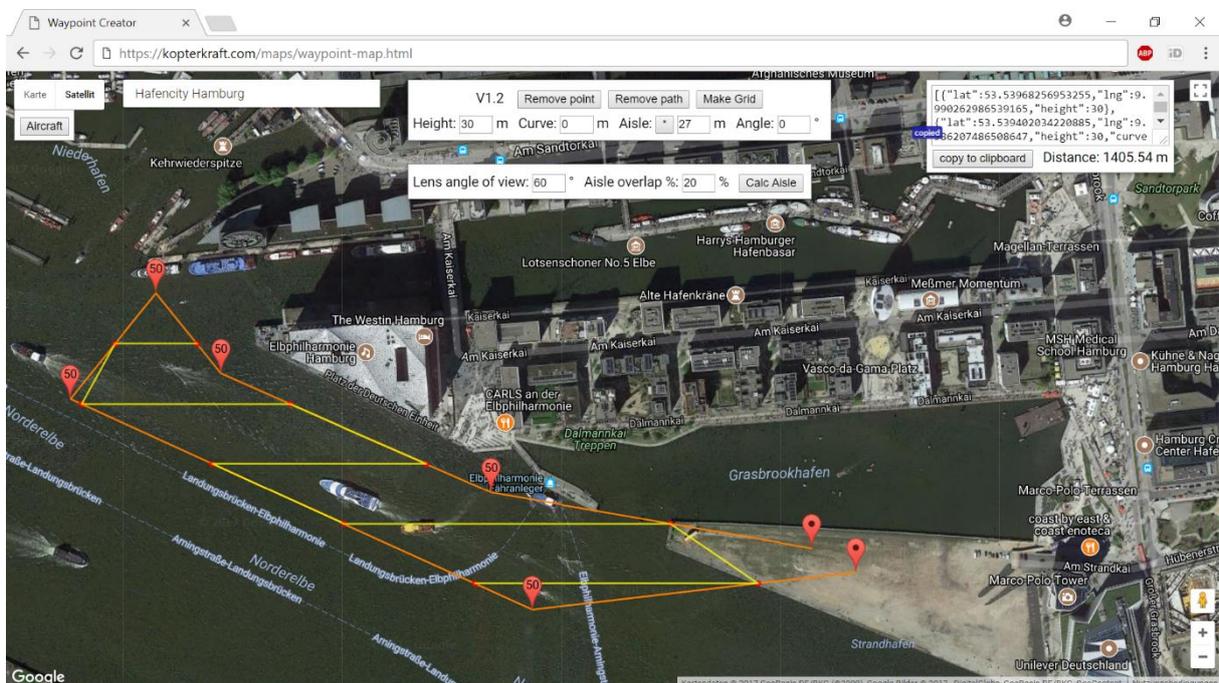
It is **STRONGLY RECOMMENDED** to test the waypoint mission functionality using the DJI simulator, which comes with the DJI assistant before executing the first “live” mission!

## 9. Waypoint Mission Editor

The waypoint mission editor is a pure web application based on Google Maps API that can be used on any HTML5 capable browser (tested on Windows PC/Mac/Linux PC, IOS & Android tablets & smartphones).

If you got credentials for the waypoint mission editor (free trial available), you can access it at <https://kopterkraft.com/maps/waypoint-map.html>. When the page is opened after login with your credentials, it will first ask for permission to use the device’s location service. If this is granted, the map is automatically centered to that location. Alternatively the powerful Google location search functionality can be used with the “Search Box” or – if the aircraft already received a valid GPS location – the map can be centered to the current aircraft location by clicking on the “Aircraft” button.

On the map you can set waypoint markers for a polyline that the drone should follow. For each point an individual height can be set. The preview will also indicate with yellow lines a grid that can be created from the polygon. The angle and direction of this grid can be adjusted with the “Angle” field. Updates will become visible when you set (or remove) a new point on the map. In order to calculate an aisle width for that grid corresponding to your flight height and lens angle of view, you can click on the “\*” button beside the aisle value. It will open a pop up with further parameters for the lens angle of view and the required overlapping:



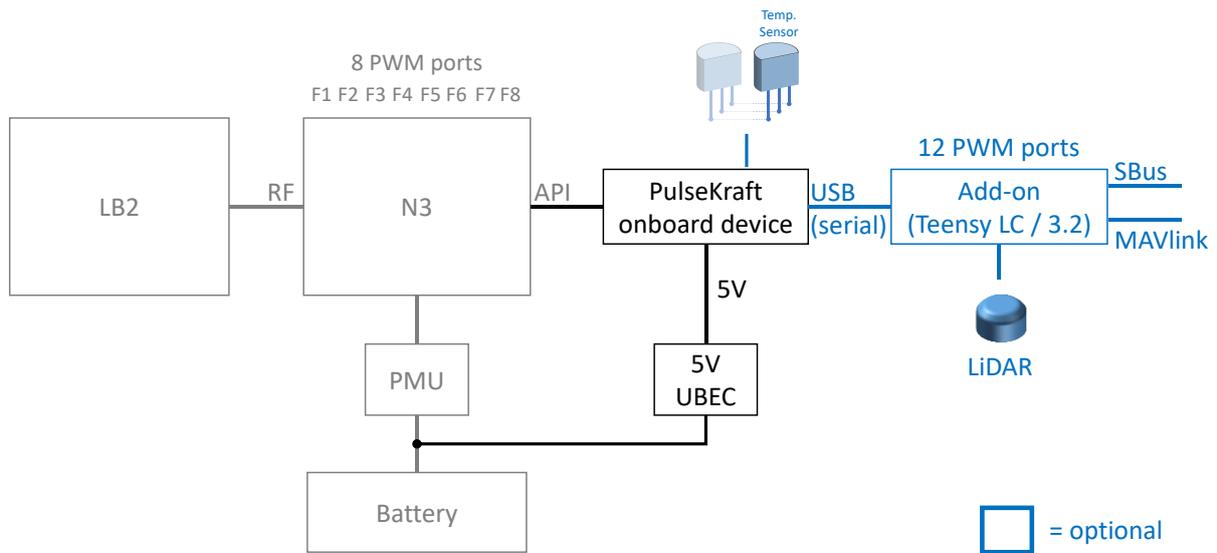
If you wish to use that grid as a waypoint mission, the polygon can be converted using the “Make Grid” button. This action can also be reverted.

Since version 1.2 of the Waypoint Mission Editor, you can also define waypoints as “curved” waypoint with a certain radius. The radius (in meters) can be set in the control panel on top as a value between 0 (no curve) and 999. It will be indicated as figure on the waypoint. Later versions of the Waypoint Editor will visualize the curve also as path. Please be aware, that in the current version,



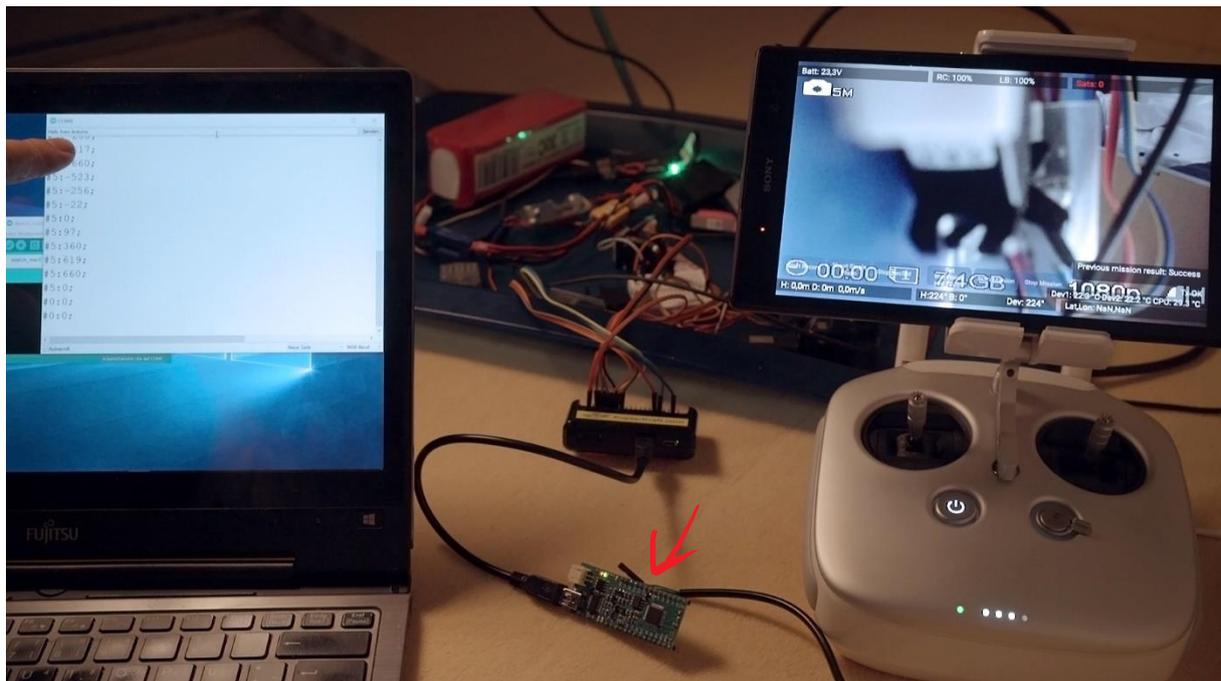


A schematic block diagram for an optional configuration could look like this:

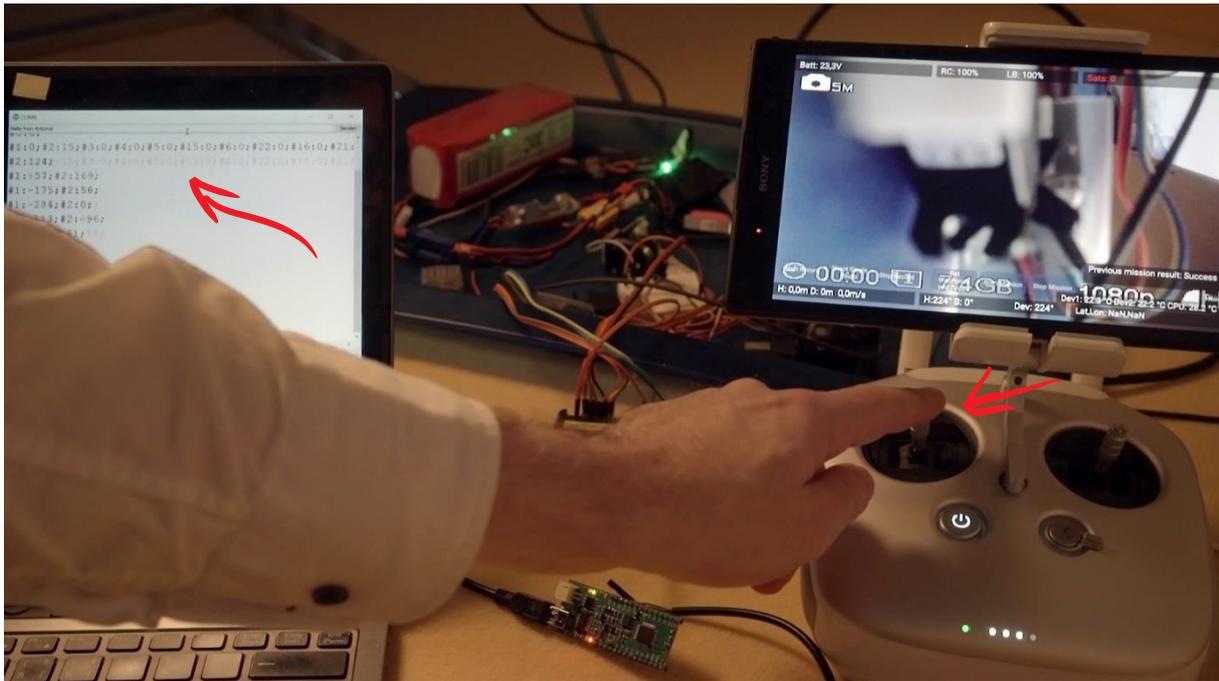


The **blue elements** are optional add-ons. Support for SBUS, MAVLink and LiDAR is planned.

The following image shows an Arduino connected to the PulseKRAFT device via USB (OTG cable). The Arduino itself is connected through another serial-to-USB converter to a laptop where the transmitted data are shown in an Arduino IDE serial monitor:



In the following image you can see status data received from the RC handset:



The format can be parsed easily with key/value/checksum blocks following this pattern:  
**#key:value;checksum**

For the sticks, wheels and buttons of the remote controller, the key is the channel number assigned to this control and the values will vary between 1000 and 2000 (1500 is the default center position) or whatever you configured in the UserConfig.txt file of the onboard device as minimum and maximum PWM servo values.

*Example* for the left shoulder wheel in upmost position:

**#5 : 2000 ; 0D**

*Example* for the left shoulder wheel in shift state (play button pressed) in almost down most position:

**#15 : 1000 ; 3F**

The channel number for the different RC controls is given in the table in section 6 “Taking Control”.

For other kind of data, the key can also be alphanumeric, like “LAT”, “LON” and “ALT” for the GPS latitude, longitude and the (barometric) altitude of the drone position. You will find this in the serial output already now but at the moment these data may be “0”. This issue is caused by a weakness in the Raspberry Pi Zero design. It may work on a RasPi 3 (not yet tested).

The checksum, which is appended after the semicolon is calculated as hexadecimal byte by XOR-ing all characters *between* (but not including) the ‘#’ and ‘;’ delimiters, similar to the way, how checksums are calculated in NMEA sentences.

Data sent from the Arduino to the PulseKRAFT device through the serial USB connection will be transmitted to the ground app, where it is written to a log file and displayed on the tablet:

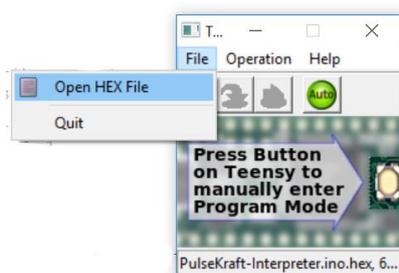


Optionally these sensor data will be transmitted to the cloud service as well. As there is no guarantee for uncorrupted transmission, you should implement some kind of checksum and optionally some kind of acknowledgement mechanism for the data you transmit as well.

As an Arduino or a Teensy has far more options to retrieve Data from almost any type of analog or digital sensor and can control PWM / I2C / SPI devices, you can integrate and control almost anything that comes to your imagination – quick and easy!

In the PulseKraft download section you will find code compiled for the Teensy LC (my favorite micro controller for these kinds of projects) as a hex-file. It can be uploaded to the Teensy LC using the teensy.exe tool which is part of the TeensyDuino plugin for the Arduino IDE (see [https://www.pjrc.com/teensy/td\\_download.html](https://www.pjrc.com/teensy/td_download.html)). Under Windows, the tool will be located in "%ProgramFiles(x86)%\Arduino\hardware\tools\teensy.exe"

If you do not want to install the full TeensyDuino plugin and Arduino IDE, you can also find the simple loader application here: <https://www.pjrc.com/teensy/loader.html>



You simply open the hex file in the Teensy uploader and afterwards press the button on the Teensy which will bring it into programming mode and the code will be uploaded to the Teensy automatically.

For more information including a pin layout of the Teensy LC please see <https://www.pjrc.com/teensy/teensyLC.html>

If you use the hex-file from the PulseKraft Download section, you get the following channel-to-pin-mapping (using the channels from the table in chapter 6 “Taking Control” and the pin numbers according to the pin layout in the Teensy LC documentation):

Channel 5 -> Pin 3  
Channel 6 -> Pin 4  
Channel 7 -> Pin 6  
Channel 10 -> Pin 9  
Channel 11 -> Pin 10  
Channel 13 -> Pin 16  
Channel 14 -> Pin 17  
Channel 15 -> Pin 20  
Channel 21 -> Pin 22  
Channel 22 -> Pin 23

For a different mapping please consult the author of the PulseKraft software.

There will be an add-on providing some preconfigured sensor interfaces and SBus signal output later.

## 12. Troubleshooting

Due to the filigree architecture with many hard- and software components that have to play together in that scenario, there is some potential for obstructions during initialization and activation. The “Activation” section already contains some basic troubleshooting hints.

Further hints:

- Is the mode switch on the RC controller in a position supporting the Onboard SDK (“F” on Matrice 100, “P” on A3 and N3)?
- Is the API activated on the flight controller and set to 230.400 baud (using DJI Assistant)?
- Make sure, your tablet has an internet connection when first time starting the PulseKRAFT app.
- Make sure, the onboard device was activated through “DJI GO App” connected to the internet before first time using the PulseKRAFT app as described in the “Activation” section.
- If you see temperature data from the backward channel and RC commands are transmitted but you see no live picture from the camera, please check the HD configuration in the DJI GO App. In rare cases the tablet may have to be rebooted in order to be able to display the live view again.

If there is no reception of data from the onboard device (like the CPU temperature) but you have a Tx connection to the onboard device, you can restart the onboard software by pressing the playback button of the remote control together with the record button. This may help in case the downward data connection was not initialized properly (e.g. you see no CPU temperature).

If there is no transmission of RC commands to the onboard device, please exit the PulseKRAFT app, disconnect it from USB port, wait 10 seconds, reconnect it after 10 seconds and choose PulseKRAFT

app from the list of compatible apps. Open the live camera view from the start screen after the app displays that it is connected to your drone.

If you still do not get it working, please connect an HDMI monitor to the HDMI output of the onboard device. The output you see there after the boot-phase may contain helpful information about the type of error. Please report the output [support@KopterKraft.com](mailto:support@KopterKraft.com) .

### 13. Copyright

The copying, distribution and utilization of this document as well as the communication of its contents to others without expressed authorization is prohibited. Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of patent, utility model or ornamental design registration.

### 14. Privacy Policy, Terms of Use and End User License Agreement (EULA)

The Privacy Policy of PulseKRAFT software and KopterKraft services can be found at:  
<http://www.kopterkraft.com/index.php/privacy>

The Terms of Use / Terms of Services for PulseKRAFT software and KopterKraft services can be found at:  
<http://www.kopterkraft.com/index.php/terms-of-service>

The End User License Agreement for the PulseKRAFT software (app, onboard software) and KopterKraft online services (like the waypoint creator) can be found at:

<http://www.kopterkraft.com/index.php/eula>