

# MikroKopter LiveView

Documentation

## Inhalt

General.....	3
Visualisation and operation.....	4
Elements.....	4
TopBar telemetry data .....	4
Left sidebar      Right sidebar .....	5
Settings.....	6
Connect .....	6
Style .....	8
Map.....	9
Thresholds .....	10
Map elements style .....	10
Data .....	11
Serial.....	11
Waypoints.....	12
Waypoint edit window .....	13
Various.....	14
Setting waypoints .....	14
Moving waypoints .....	14
Map zoom.....	14
Map scroll.....	14
Additional Software:.....	16
Obligatory .....	17

## General

MK-LiveView was created with the intention to show relevant telemetry data of the copter clear on the screen.

The data is send by a serial connection between the debug port of the copter and the pc.

By implementing the code from GreatMaps ( <http://greatmaps.codeplex.com/> ) the actual position of the copter can be displayed on different kind of maps.

With the Versatile Web Cam Library by lepipele (<http://www.codeproject.com/Articles/125478/Versatile-WebCam-C-library> ) it is also possible to display a camera window displaying a live image of a connected camera.

The programm depends on the Navi-Ctrl with gps module.

At the time of creation the firmware of the copter used was:

Flight-Ctrl: V2.15k

Navi-Ctrl: V2.15h

BL-Ctrl: V1.10

The software was tested with the following hardware:

Flight-Ctrl: V2.2 (V2.1 with ACC upgrade)

Navi-Ctrl: V2.0

BL-Ctrl: V3.0

The programm was created with VisualStudio 2015 in WPF/C# .

The programm depends on the Visual C++ Redistributable for Visual Studio 2015 ( <https://www.microsoft.com/de-de/download/details.aspx?id=48145> )

The „Mikrokopter Serial Control Tutorial“ (<http://hdl.lib.byu.edu/1877/2747>) by John C. Macdonald has contributed essentially for the implementation oft he programm. He created a C#-project for controlling a MikroKopter by PC.

# Visualisation and operation

## Elements

The screenshot shows the MK LiveView v1.0 interface. A central satellite map displays a drone's position over a residential area with a river. The interface includes a top bar with telemetry data, a left sidebar with menu buttons, a right sidebar with copter status, and a settings window. Callout boxes identify the following elements:

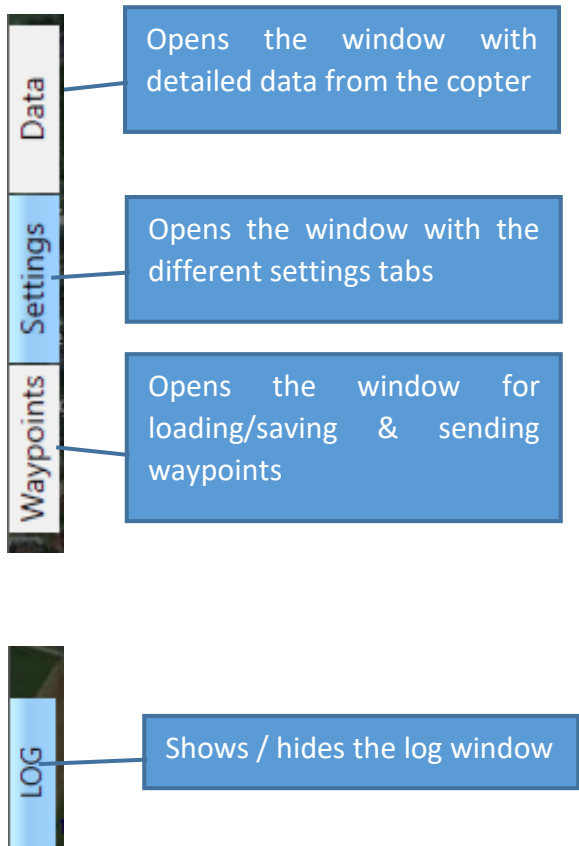
- Topbar with telemetry data**: Located at the top of the window, displaying various flight metrics.
- Serial port and connection status (opens/closes connection with click)**: A button in the top bar used to manage the connection.
- Toggle fullscreen button**: A button in the top right corner of the window.
- Settings window**: A panel on the left side containing options for serial port connection, status, port, baudrate, databits, stopbits, parity, and timing updates.
- Log message windows displaying general notifications or warnings**: Two empty white boxes at the bottom of the map area.
- Artificial horizon with indicators for heading and homeposition**: A circular indicator in the bottom right corner of the map area.
- left sidebar with menu buttons**: A vertical bar on the left side with buttons for Data, Settings, Waypoints, and LOG.
- Right sidebar with copter status and menu buttons**: A vertical bar on the right side with buttons for OSD, Motors, and WP-Edit.

## TopBar telemetry data

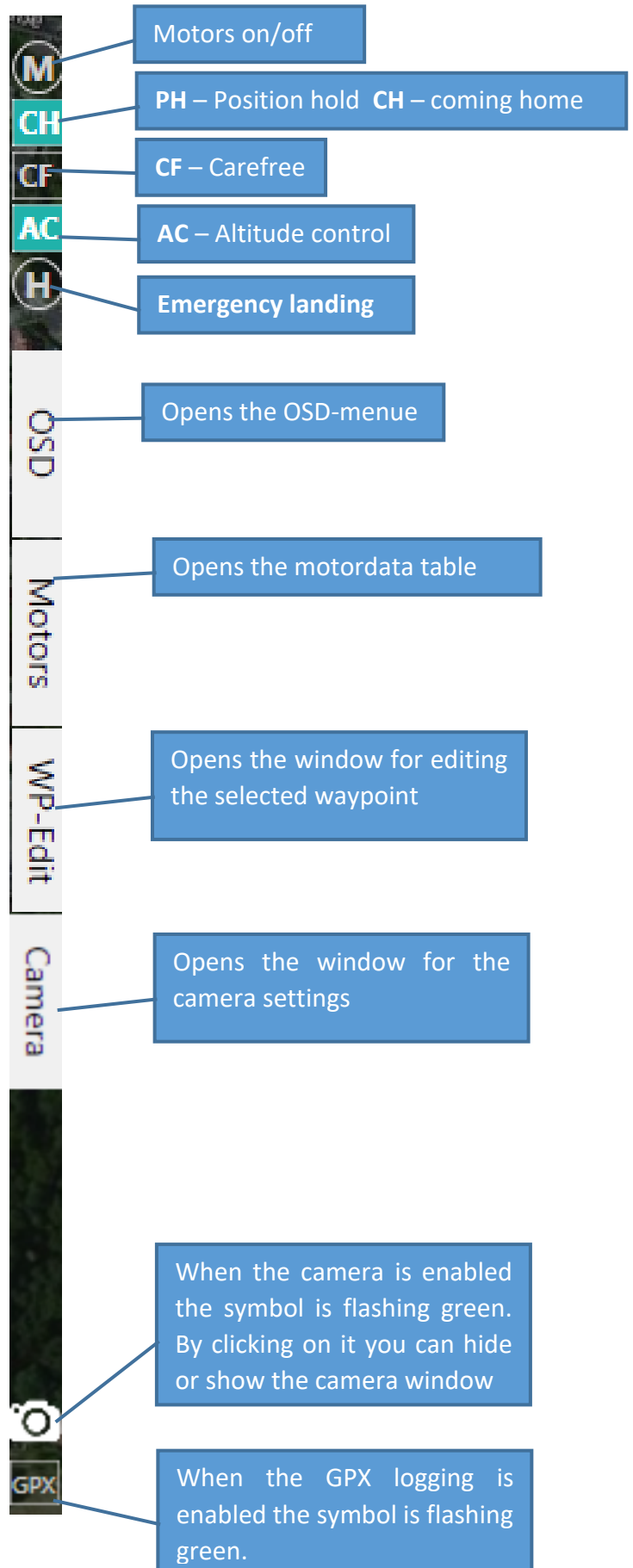
A close-up of the top bar telemetry data. Each icon and value is highlighted with a callout box:

- Capacity used**: 16.3 V
- Remaining voltage**: 16.3 V
- flighttime**: 125 mAh
- Current consumption**: 0.5 A
- height**: 10.2 m
- Distance to homeposition**: 0.0 m
- Speed over ground**: 0.07 m/s
- Earth magnetic field**: 106%
- Satellites in use**: 5
- RC level quality**: 200

Left sidebar

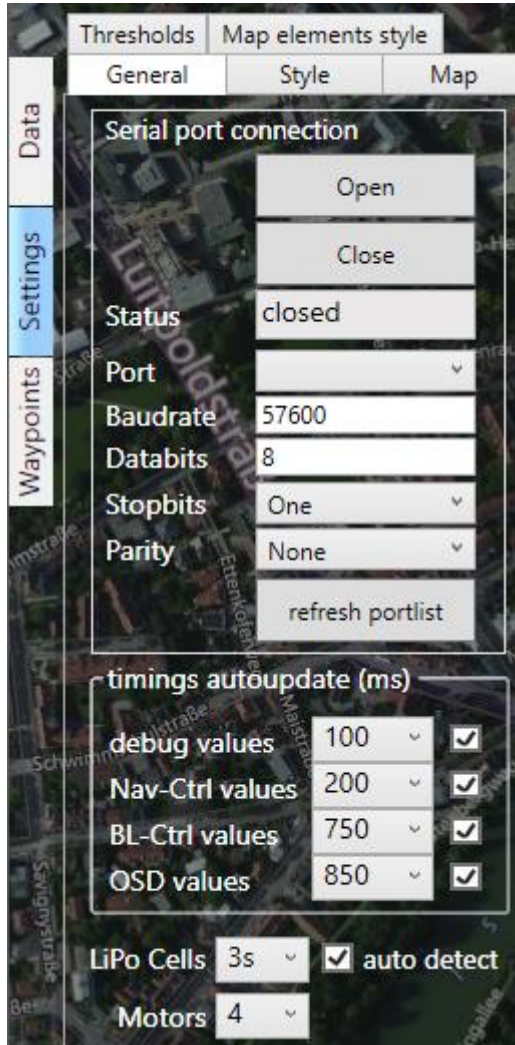


Right sidebar



## Settings

## Connect



When start up the subwindow Settings is opened, you first need to check if the right COM-Port is selected.

The other settings are already set to the necessary communication settings – don't change them unless You know what You're doing!

By clicking 'Open' the COM-Port is initialized and the program starts requesting the telemetry data from the copter.

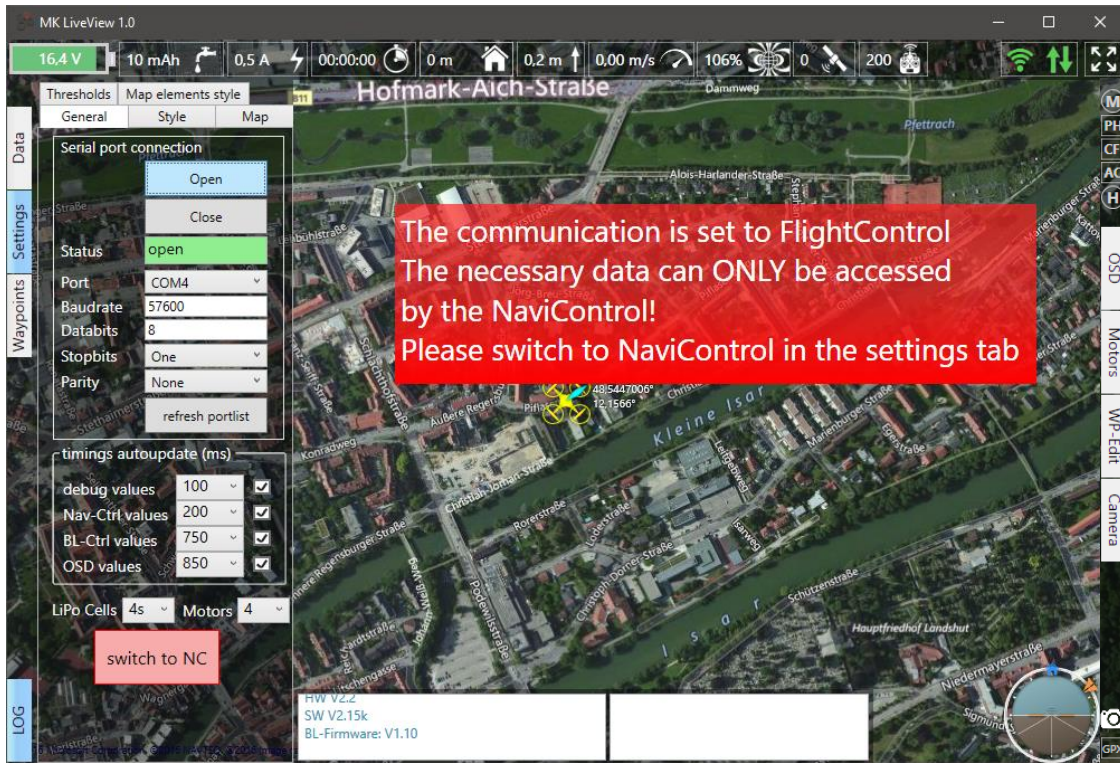
The timings usually should work with the current settings.

The debug and NavCtrl timings are very fast in order to give a 'floating' experience.

The LiPo Cells settings affects the warning thresholds for battery level.

By checking 'auto detect' the value is set by the copter.

The motors settings determines how many rows are shown in the motors tab



In case the active controller on the copter is set to FC, a warning is displayed and a button 'switch to NC' is displayed on the settings tab.

## Style



Most of the subwindows and bars can be scaled to fit your needs.

These settings may be stored for normal window state and for fullscreen state.

By 'Reset' the scalings are set back to default.

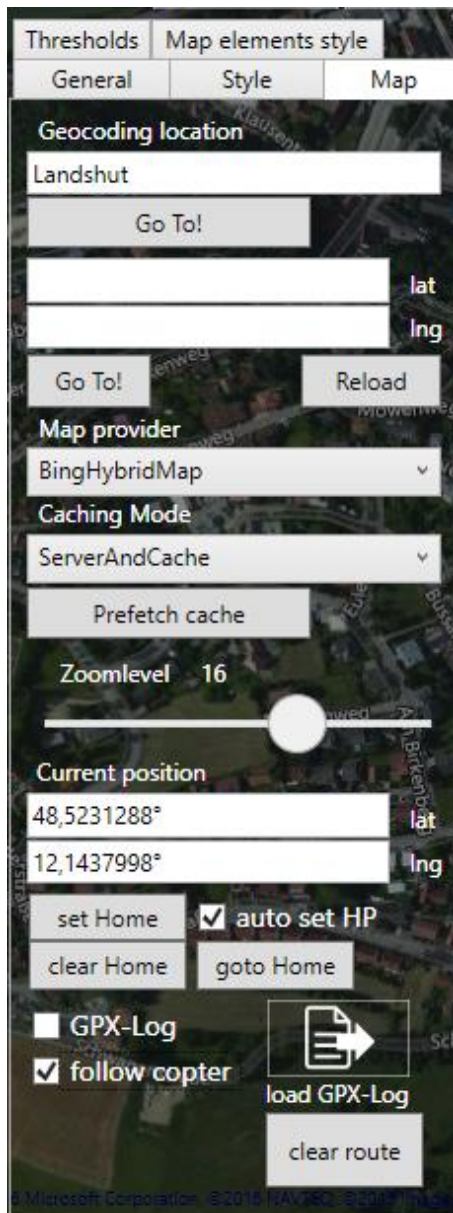
Each of the topbar items except the connection status and the fullscreen button can be hidden

The artificial horizon in the lower left corner can be hidden too.

All these settings are stored when closing the programm and restored at next start.



## Map



**Geocoding location** lets you move the map to a known location like a city.

You can also put the coordinates directly in the fields below.

**Map provider** lets you chose from different online map providers.

There are a lot more available, but I selected the most reliable and useful providers.

**At this point I have to notice You, that some providers like google don't agree that their services are accessed directly like gmap does. There has been a lot of discussion about this on their website with some officials of the providers.**

**The problem is, that their APIs are only available for webapplications but not for desktop apps like gmaps.**

**So it's up to you if you want to use google maps service nevertheless or not.**

**As far as I know bing maps/microsoft don't have a problem with accessing their data like this...**

The great thing about GreatMaps is that you can **prefetch** necessary map areas for later offline use.

For this you need to select an area of the map by **holding the ALT key and drawing a rectangle with the left mousebutton pressed**. Finally click '**prefetch cache**'

The programm will download all tiles from the current zoomlevel upward to maximum zoomlevel. **Therefore**

**you should check before wich levels you really need as this is quite a lot of data at higher levels.**

You will always be asked after one level is finished if you want to proceed or skip next level or if you want to abort (finish) prefetching.

When **auto set HP** is checked, the homeposition is set when at least 5 sattelites are available.

If you want to change, correct the position, you can always use '**set home**' / '**clear home**'  
The position is taken from the center of the map.

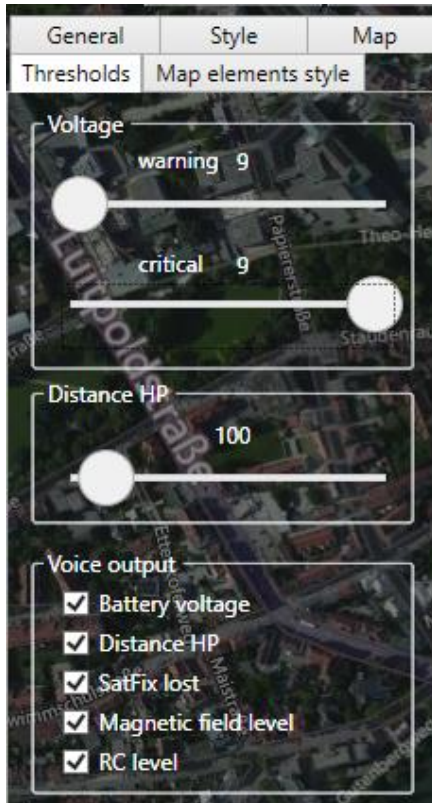
**HINT:** You can also set the homepositon by clicking the '**distance to home**' field in the topbar

**Follow copter** keeps the copter centered on the map as it is moving.

**GPX-log** logs the position and time beginning whe the copter is in the air.

It is stored in the general gpx format and can be viewed with any compatible gpx viewer. You can also load the logged route with this programm.

## Thresholds



Here you can set warning levels for voltage and distance.

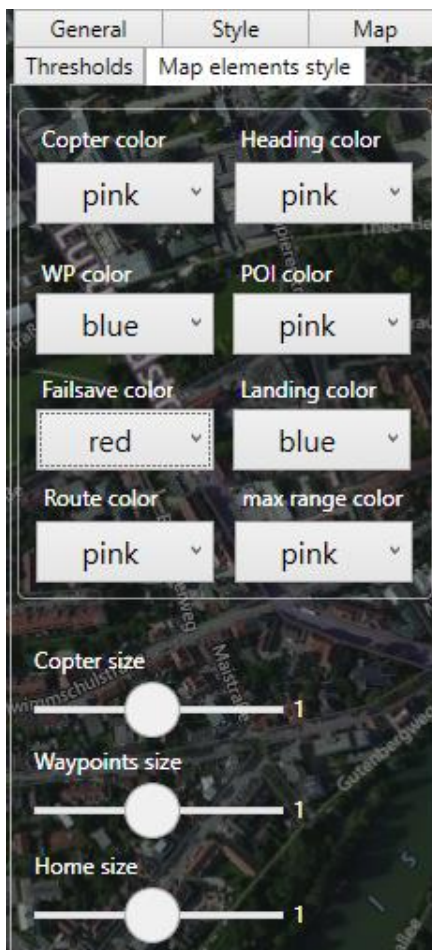
You may also de-/activate voice output on certain conditions.

All conditions listed here are also visualized with red flashing in the top bar.

The voice warnings have been generated with a [text to speech engine](#) in the web.

**Hint:** If want to use a different voice or language you can record the voice there (with a recording programm like audacity) and replace the appropriate file in the "Voice" subfolder of the programm.

## Map elements style

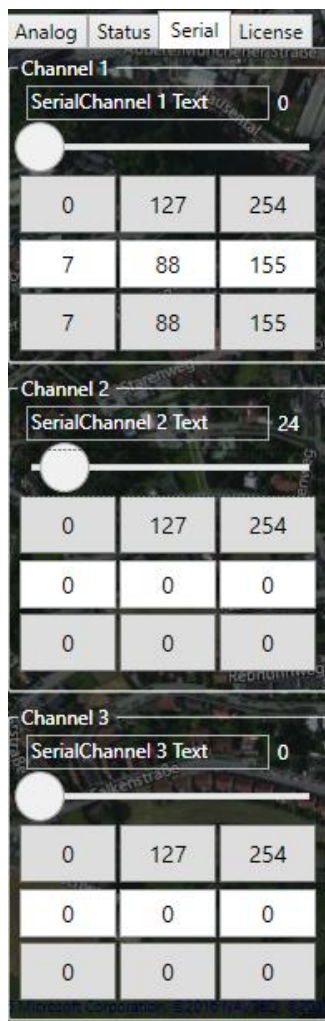


Here you can change the colors for the different map elements.

Below the sizes of the Coptersymbol and the waypoints can be changed.

Data

Serial



The serial channels might be useful if you don't have enough channels on your RC.

With serial channels you could extend the RC by 8 channels.

At moment 3 of them are implemented.

**One thing to mention is that by design always all 8 channels will be set when changing just one.**

So please make sure that you haven't assigned any other channels in the MK tool when using this here.

In the first line you can assign a name to the channel.

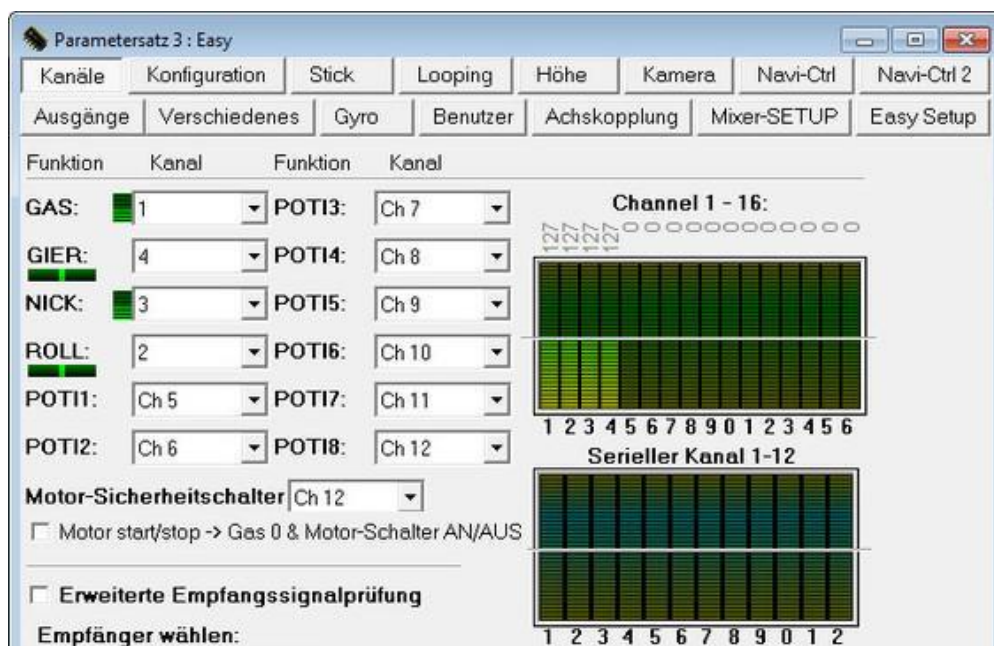
The slider control immediately assigns the set value to the channel.

There are three buttons below that may act like a three state switch on the RC.

Below you can assign 3 individual values to the buttons below. These might be values that you need often like certain cam angles...

In order to make this work you have to set up the assignment of the serial channels in the MK-Tool. Normally you assign them to

any of the pots on channel page:



There are also other possibilities to assign the channels to – just look and see...

## Waypoints

The screenshot shows the MK LiveView 1.0 interface. At the top, there's a status bar with battery (16.3 V), power (0.5 A), and other flight parameters. Below that is a table of waypoints. The main area is a map showing a flight path connecting waypoints P1 through P13, ending at a Point of Interest (POI) O14. A control panel at the bottom contains buttons for sending/receiving waypoints, loading/saving files, and clearing lists. A simulation panel on the right has checkboxes for showing route, statusbar, max range, and auto show edit window. A status box at the bottom left shows hardware and software versions.

Index	Type	Name	Latitude	Longitude	Altitude	Heading	Speed	ClimbRate	Radius	HoldTime	AutoTrigger	CamAngle	Out1Timer
1	WP	P1	48,5229832	12,1441282	22.5 m	WP 14	1,2 m/s	0,5 m/s	10 m	5 s	---	Auto	1
2	WP	P2	48,5228642	12,1441899	33,6 m	WP 14	2,2 m/s	0,5 m/s	10 m	5 s	---	---	8
3	WP	P3	48,5227061	12,144316	44,7 m	WP 14	3,3 m/s	0,5 m/s	10 m	5 s	---	Auto	8
4	WP	P4	48,5226226	12,1445547	55,8 m	83°	4,0 m/s	0,5 m/s	10 m	5 s	---	Auto	8
5	WP	P5	48,522699	12,1449221	66 m	83°	5,0 m/s	0,5 m/s	10 m	5 s	---	Auto	8
6	WP	P6	48,5228518	12,145067	77 m	83°	6,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
7	WP	P7	48,5229939	12,1450938	88 m	83°	7,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
8	WP	P8	48,523104	12,1450723	99 m	83°	8,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
9	WP	P9	48,5231999	12,1448792	22 m	83°	9,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
10	WP	P10	48,5232319	12,1444528	22 m	83°	10,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
11	WP	X11	48,5232017	12,1442597	22 m	83°	11,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
12	WP	Y12	48,5231022	12,1441336	22 m	83°	12,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
13	WP	Z13	48,5230383	12,1441309	22 m	83°	13,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1
14	POI	O14	48,5229317	12,1445359	100 m	83°	6,0 m/s	0,5 m/s	10 m	5 s	---	Auto	1

This diagram highlights the file management buttons in the control panel with callouts:

- send to copter**: Sends the list to the copter
- receive from copter**: receives the list stored on the copter
- load file**: Loads a \*.wpl file
- save file**: Saves the list to a \*.wpl file
- clear local list**: Clear the local list
- clear copter list**: Clear the list on the copter

This diagram highlights the simulation settings panel with callouts:

- show route**: Show/hide the routepath
- show statusbar**: Show/hide the statusbar on top
- show max range**: Show/hide the maximum range circle according to your license
- auto show edit window**: Automatically open the waypoint edit window when selecting a point from the list

## Waypoint edit window



Here you can edit a selected waypoint

You can find a detailed description of the values in the [MK WaypointEditor wiki](#)

When you have finished editing you need to save the changes with the button 'save changes'

Hitting 'add WP' adds the current point the end of the list.

'Delete WP' deletes the point from the list.

'Send to copter' sends the waypoint data to the copter (it's the data from this window, so changes you haven't stored yet are send nevertheless)

'Set as active WP' activates the WP as active target – **this is only working with the new beta FW (2.15h) and waypointflight has to be active (at moment)**

'Add current Pos' adds the current position of the copter as new waypoint to the list. Depending on the selected options 'use copter height' and 'use copter heading' these values will be

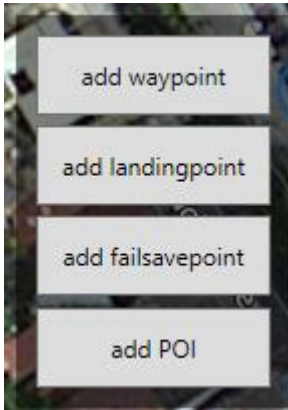
taken from the copter or will be set to the default values.

'save as default' stores the current values as default values. Next time you add a waypoint these values will be taken.

**Up/down** arrows move the waypoint in the list accordingly. This affects the order the waypoints are processed.

## Various

### Setting waypoints



You can set waypoints either as described before by clicking **'Add current Pos'** or by double clicking with the left mouse button or double tapping on a touch display on the desired position.

**The option 'add landingpoint' is only working with the new beta FW (2.15h) and the upcoming final release 2.16 (and onward)**

### Moving waypoints

You can simply move the waypoints by dragging the waypoint with the left mousebutton pressed or by touch (needs exact positioning of the finger)

### Map zoom

You can either zoom with the scroll wheel of the mouse, the touch zoom gesture with two fingers or with the zoom slider on the map settings tab

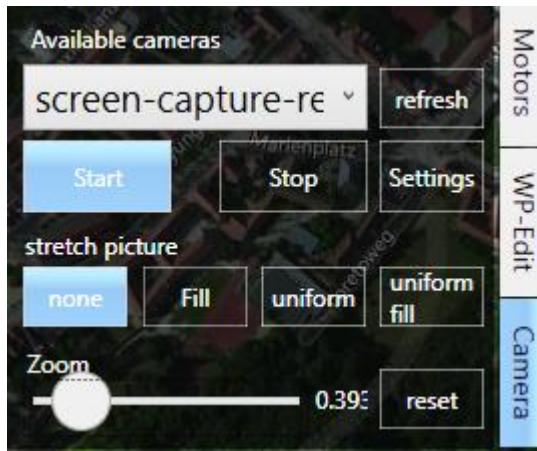
### Map scroll

The map is scrolled with right mouse button pressed or by touch dragging

## Camera

With the appropriate hardware like a usb-videograbber you can display the videosignal from your copter in in the main window.

In order to work the device has to be recognized properly by the system.



In the dropdown box all available devices are listed. You can refresh the list (e.g. when attaching a device when the programm is already started) by clicking **refresh**.

The camera device gets connected and videostream is captured by clicking **start**.

If supported by the device you can change display settings by clicking **settings**

You can choose to display the video signal in a small window or to fill the main window uniform or stretched.

The small window can be positioned and sized as you like. Therefore **none** has to be selected. In case the window moves out of sight you can reset the window by pressing **reset**

**Fill** fills the whole window regardless of the aspect ratio.

**Uniform** fills the window at maximum size by keeping the aspect ratio.

**Uniform fill** fills the whole window by keeping the aspect ratio – parts may be cut off.

### Additional Software:

The artificial horizon is based on the discussion [Creating an advanced HUD](#)

The map display and processing by [GreatMaps](#) – really great!

I made some changes and additions to fit my needs – so if you want to recompile the project you can't use the original source of greatMaps – the changed sources are available in the [MK svn](#).

The camera library [Versatile Web Cam Library by lepipele](#)

And last but not least parts of the sources from the „**Mikrokopter Serial Control Tutorial**“ (<http://hdl.lib.byu.edu/1877/2748> )



## Obligatory

```
///=====
/// MKLiveView
/// Copyright © 2016 Steph
///
///This file is part of MKLiveView.
///
///MKLiveView is free software: you can redistribute it and/or modify
///it under the terms of the GNU General Public License as published by
///the Free Software Foundation, either version 3 of the License, or
///(at your option) any later version.
///
///MKLiveView is distributed in the hope that it will be useful,
///but WITHOUT ANY WARRANTY; without even the implied warranty of
///MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
///GNU General Public License for more details.
///
///You should have received a copy of the GNU General Public License
///along with cssRcon. If not, see <http://www.gnu.org/licenses/>.
///
///=====
///Credits:
/// radioman (http://www.codeplex.com/site/users/view/radioman)
/// for his really GreatMaps! (http://greatmaps.codeplex.com/)
///
/// I made some changes to the source, so You need all files from this project here in
order to compile and run
///
/// JOHN C. MACDONALD at Ira A. Fulton College of Engineering and Technology
/// for his MIKROKOPTER SERIAL CONTROL TUTORIAL (http://hdl.lib.byu.edu/1877/2747)
/// and the sourcecode (http://hdl.lib.byu.edu/1877/2748)
/// By his work I finally managed to get the communication with the Mikrokopter
controllers to run
/// Some of his code was used in this programm like the SempelSerialPort class (with
lots of changes)
/// and the FilghtControllerMessage class
///
///=====
/// DISCLAIMER
/// =====
///
/// I created this software with my best knowledge and belief.
///
/// IN NO EVENT, UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING,
/// SHALL I, OR ANY PERSON BE LIABLE FOR ANY LOSS, EXPENSE OR DAMAGE,
/// OF ANY TYPE OR NATURE ARISING OUT OF THE USE OF,
/// OR INABILITY TO USE THIS SOFTWARE OR PROGRAM,
/// INCLUDING, BUT NOT LIMITED TO, CLAIMS, SUITS OR CAUSES OF ACTION
/// INVOLVING ALLEGED INFRINGEMENT OF COPYRIGHTS,
/// PATENTS, TRADEMARKS, TRADE SECRETS, OR UNFAIR COMPETITION.
///
/// This means: use it & have fun (but @ Your own risk...)
/// =====
```