

MikroKopter LiveView

Dokumentation

Inhalt

Allgemein.....	3
Anzeige und Bedienung.....	4
Elemente	4
Kopfleiste Telemetriedaten.....	4
Linke Seitenleiste Rechte Seitenleiste	5
Einstellungen	6
Verbinden	6
Stil.....	8
Karte	9
Grenzwerte.....	10
Stil der Kartenelemente	11
Daten	12
Serielle Kanäle	12
Wegpunkte	13
Wegpunkte Editierfenster	15
Verschiedenes	16
Wegpunkte setzen.....	16
Wegpunkte setzen mit der ‚Toolbox‘	16
Wegpunkte bewegen/verschieben	16
Karten Zoom.....	16
Karte verschieben.....	16
Kamera	17
Zusätzliche Software:	18
Obligatorisch	19

Allgemein

MK-LiveView wurde mit der Absicht erstellt relevante Telemetriedaten übersichtlich auf dem Bildschirm darzustellen.

Die Daten werden über eine serielle Verbindung zwischen der Debug-Schnittstelle des Kopters und dem Pc ausgetauscht.

Durch die Implementierung des Codes von GreatMaps (<http://greatmaps.codeplex.com/>) kann die aktuelle Position des Kopters auf verschiedenen Karten angezeigt werden.

Mithilfe der "Versatile Web Cam Library" von lepipele (<http://www.codeproject.com/Articles/125478/Versatile-WebCam-C-library>) ist es auch möglich ein Kamerafenster mit einem Live-Bild einer angeschlossenen Kamera anzuzeigen.

Das Programm benötigt eine Navi-Ctrl mit GPS-Modul.

Zum Zeitpunkt der Erstellung wurde folgende Firmware auf dem Kopter verwendet:

Flight-Ctrl: V2.15k (BETA)

Navi-Ctrl: V2.15h (BETA)

BL-Ctrl: V1.10

Die Software wurde mit folgender Hardware getestet:

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

Navi-Ctrl: V2.0

BL-Ctrl: V3.0

Das Programm wurde mit VisualStudio 2015 in WPF/C# erstellt.

Das Programm benötigt die „Visual C++ Redistributable for Visual Studio 2015“ (<https://www.microsoft.com/de-de/download/details.aspx?id=48145>)

Das „Mikrokopter Serial Control Tutorial“ (<http://hdl.lib.byu.edu/1877/2747>) von John C. Macdonald hat wesentlich zur Umsetzung des Programms beigetragen. Dort wird beschrieben wie man mit einem C#-Projekt einen MikroKopter mittels PC steuert.

Anzeige und Bedienung

Elemente

The screenshot shows the MK LiveView v1.0 software interface. The central part is a satellite map of a city area. On the left, there is a sidebar with menu items: 'Daten', 'Einstellungen', 'Wegpunkte', and 'LOG'. On the right, there is another sidebar with status indicators: 'OSD', 'Motoren', 'WP-Editor', and 'Kamera'. At the top, there is a toolbar with various icons for navigation and settings. A 'Serial port connection' window is open on the left, showing connection status and settings like baud rate (57600) and port selection. A 'Kopfleiste' (header bar) at the top displays various telemetry data. A 'Künstlicher Horizont' (artificial horizon) is visible in the bottom right corner of the map area. A 'Meldefenster' (notification window) is located at the bottom center of the map area.

Labels pointing to the interface elements:

- Kopfleiste mit Telemetriedaten
- Serieller Port und Verbindungsstatus (öffnet/schließt die Verbindung mit Klick)
- Button zum Wechsel in den Vollbildmodus
- Einstellungsfenster
- Meldefenster mit generellen Nachrichten oder Warnungen
- Künstlicher Horizont mit Anzeige der Flugrichtung und Homeposition
- Linke Seitenleiste mit Menü Buttons
- Rechte Seitenleiste mit Kopter-Status und Menü Buttons

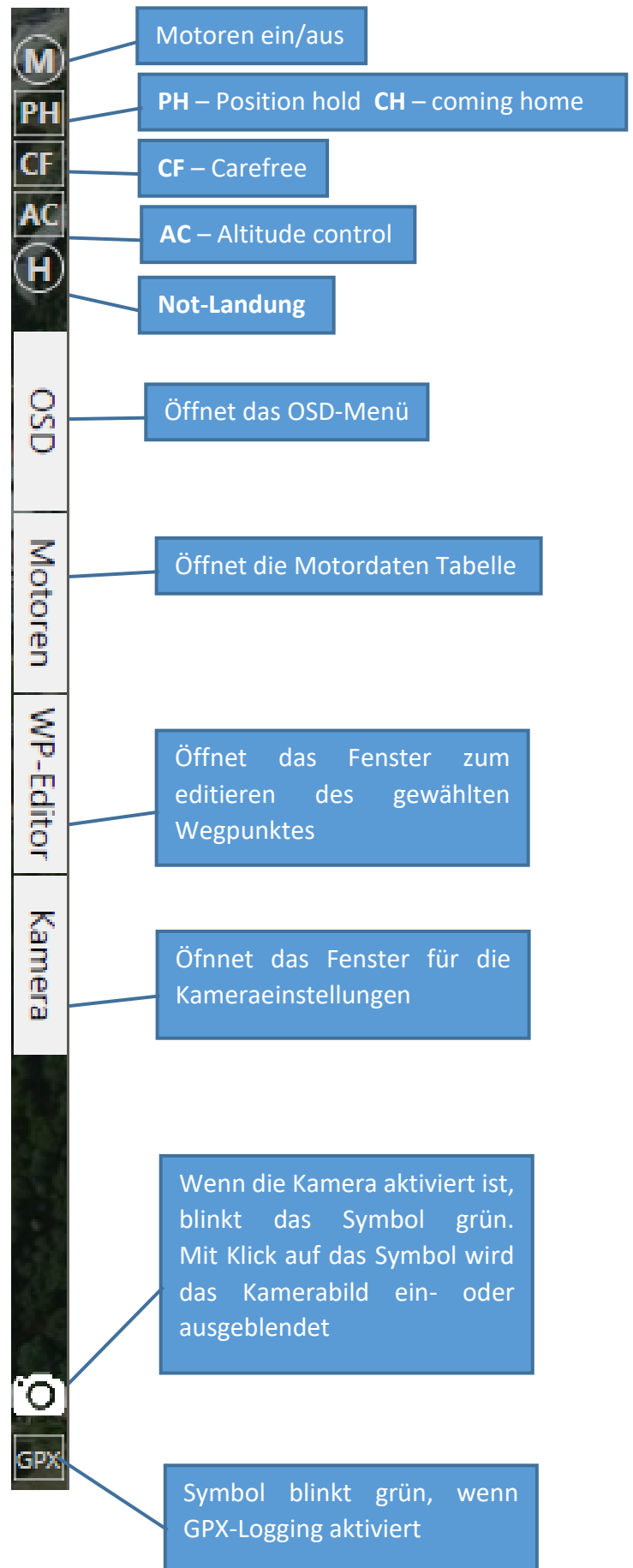
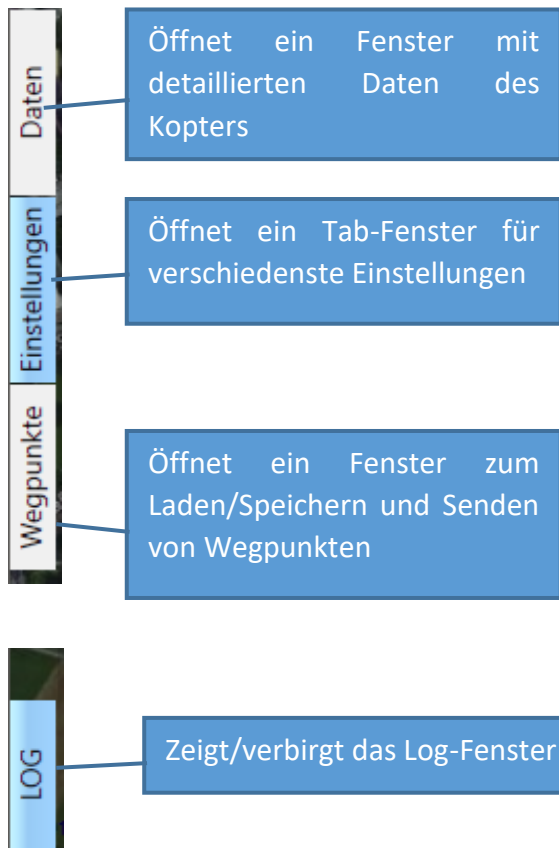
Kopfleiste Telemetriedaten

The close-up shows the telemetry header bar with the following data points and their labels:

- Verbrauchte Kapazität: 16.3 V (LiPo-Spannung)
- 125 mAh (Stromverbrauch)
- 0.5 A (Stromverbrauch)
- 00:00:00 (Flugzeit)
- 10.2 m (Entfernung zur Homeposition)
- 0,0 m (Entfernung zur Homeposition)
- 0,07 m/s (Horizontale Geschwindigkeit)
- 106% (Erdmagnetfeld)
- 5 (Verfügbare Satelliten)
- 200 (RC Signalqualität)

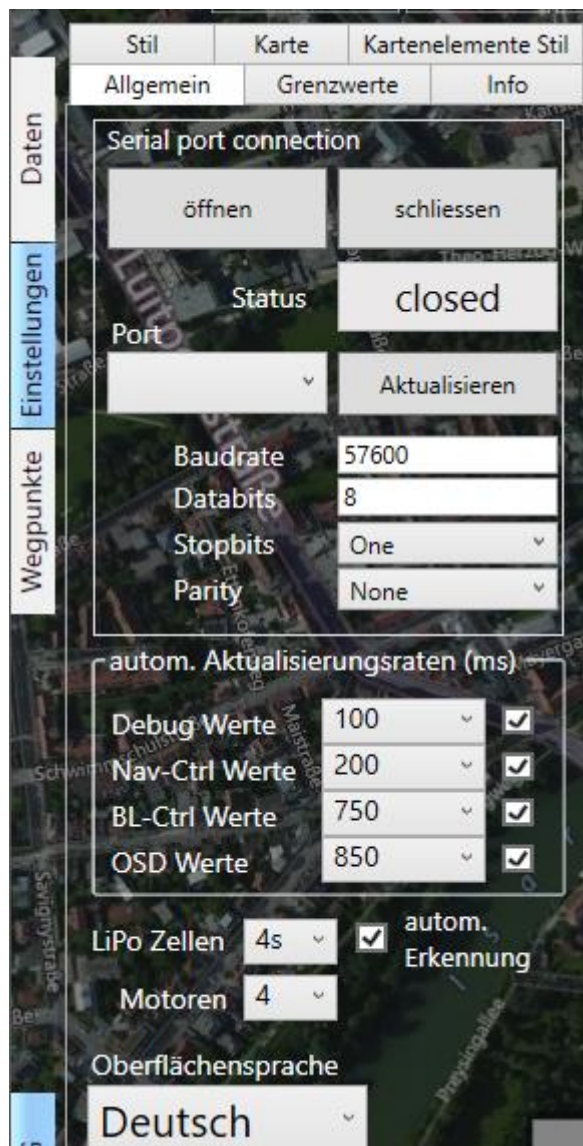
Linke Seitenleiste

Rechte Seitenleiste



Einstellungen

Verbinden



Programmoberfläche stehen momentan in Deutsch und Englisch zur Verfügung und können hier nach Bedarf umgeschaltet werden.

Bei Programmstart ist das Einstellungsfenster geöffnet. Hier muss geprüft werden ob der richtige COM-Port eingestellt ist.

Die anderen Einstellungen habe bereits die richtigen Werte für die Kommunikation mit dem Kopter – man sollte sie nicht ändern, außer man weiß was man tut!

Durch Klick auf '**öffnen**' wird die COM-Schnittstelle initialisiert und das Programm beginnt die Telemetriedaten vom Kopter auszulesen.

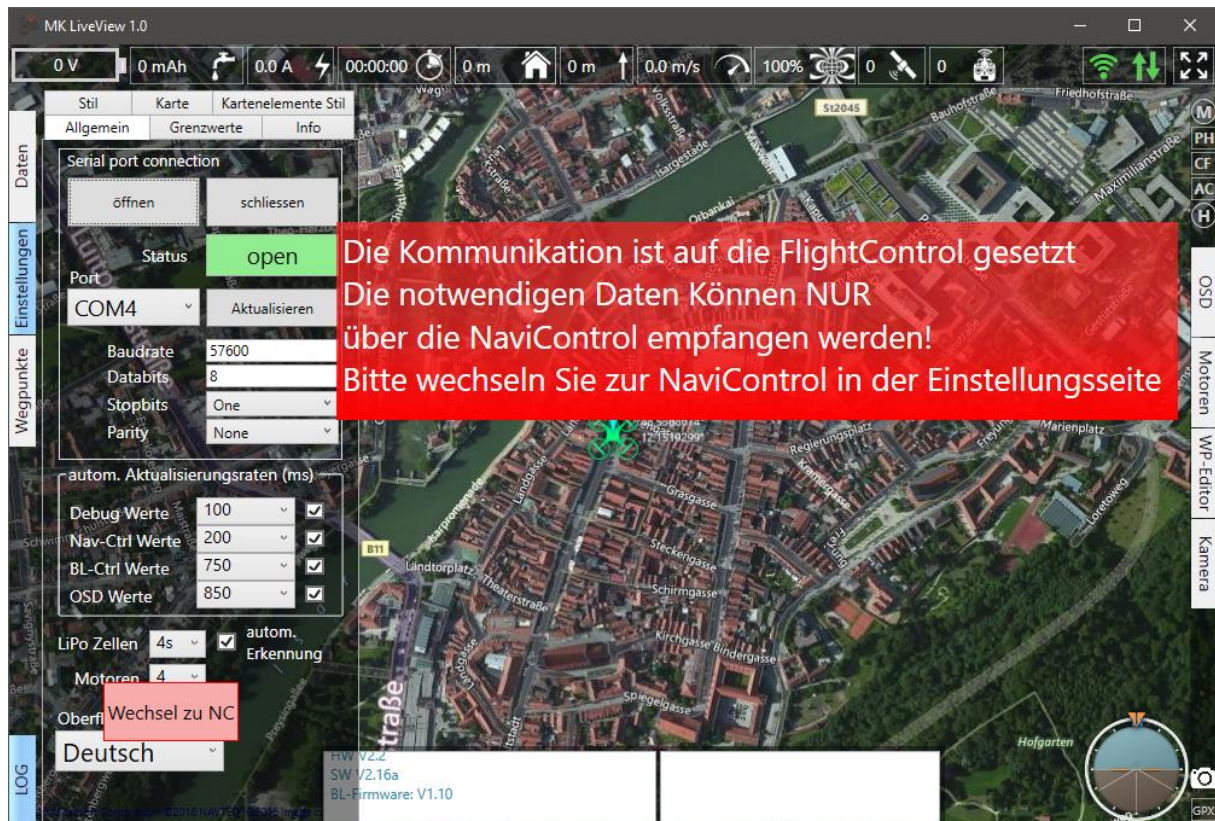
Die **autom. Aktualisierungsraten** sollten im Normalfall mit den aktuellen Einstellungen funktionieren.

Die Debug- und Nav-Ctrl Timings sind sehr schnell damit die Anzeige ‚flüssig‘ dargestellt wird

Die '**LiPo Zellen**' Einstellung beeinflusst die Grenzwerte für die Batterie Warnungen. Mit '**autom. Erkennung**' wird der Wert durch den Kopter gesetzt.

Mit der '**Motoren**' Einstellung wird die Anzahl der anzuzeigenden Zeilen in der Motor-Daten Tabelle festgelegt.

Oberflächensprache - Die Anzeigetexte der



Falls die FlightControl im Kopter als aktiver Controller gesetzt ist, erscheint eine Warnung und ein roter Button 'switch to NC' im Einstellungsfenster mit dem man die 'Navi-Ctrl' als aktiven Controller setzen kann.

Dies ist für den ordnungsgemäßen Ablauf des Programms unabdingbar!

Stil



Die meisten (Unter-)Fenster und Leisten lassen sich je nach Bedürfnis skalieren.

Diese Einstellungen können für den normalen Fenstermodus (**Normale Fensterskalierung speichern**) und für das Vollbild (**Vollbild Skalierung speichern**) gespeichert werden.

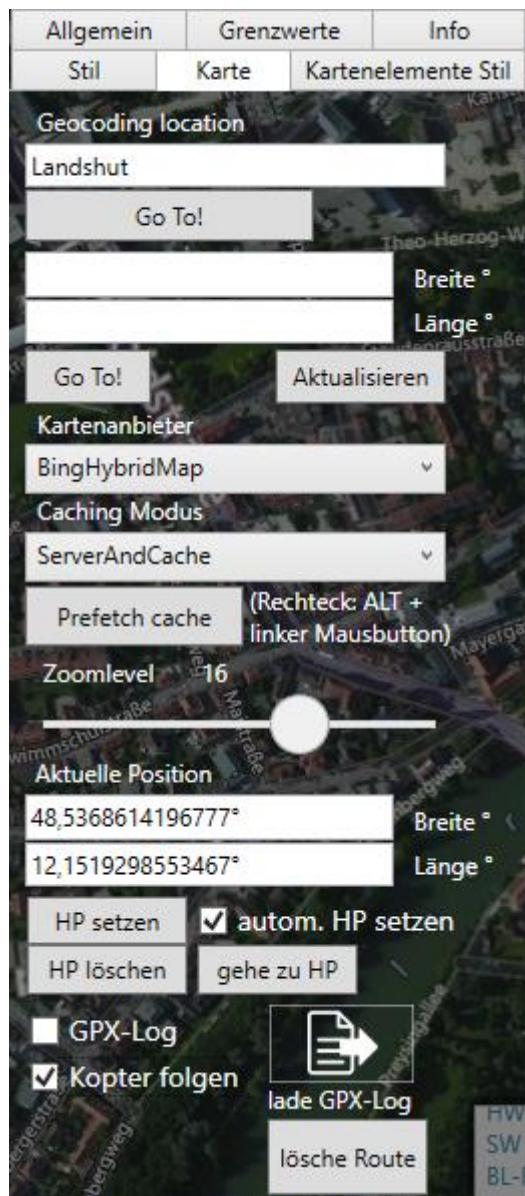
Mit **Rücksetzen** werden alle Skalierungen wieder auf die Standardwerte gesetzt.

Alle Elemente der Kopfleiste (außer dem Verbindungsstatus und dem Vollbild Button) können bei Bedarf ausgeblendet werden.

Der künstliche Horizont kann ebenfalls ausgeblendet werden.

Diese Einstellungen werden bei Beendigung des Programms gespeichert und beim nächsten Start wieder geladen.

Karte



Mit **Geocoding location** kann man mit der Karte zu einer bekannten Lokalität (zum Bsp. eine Stadt) springen.

U.U. kann es nötig sein das Land mit Komma getrennt anzuhängen. („Berlin“ wird zum Beispiel nicht erkannt, wohl aber „Berlin, Deutschland“ oder „Berlin, Germany“)

Man kann auch die Koordinaten direkt in den Feldern darunter eingeben.

Kartenanbieter enthält eine Liste verschiedener Online karten Anbieter.

Es sind erheblich mehr verfügbar - hier wurde eine Auswahl der zuverlässigsten und relevantesten gemacht.

An dieser Stelle muss ich darauf hinweisen, dass einige Anbieter wie Google nicht mit der Art des Zugriffs auf ihren Dienst einverstanden sind, wie es mit gMaps realisiert worden ist.

Die Nutzungsbedingungen von Google sind diesbezüglich aber nicht eindeutig weshalb es im Forum von gMaps einige Diskussionen mit offiziellen Vertretern einiger Anbieter gab. Das Problem ist, dass von den Anbietern nur APIs für Webapplikationen zur Verfügung stehen aber nicht für Desktopanwendungen wie gMaps.

So müßt ihr letztenendes Entscheiden ob ihr Google Maps verwenden wollt – soweit ich weiß,

hat Microsoft mit bing maps keine Einwände dagegen, dass auf ihre Daten so zugegriffen wird wie es mit gMaps realisiert worden ist.

Das großartige an GreatMaps (GroßartigeKarten) ist, dass man die benötigten Kartendaten vorladen (**Prefetch cache**) kann um sie dann zu einem anderen Zeitpunkt offline zu verwenden.

Hierzu zieht man um den gewünschten Bereich ein Rechteck mit gedrückter „ALT“-Taste und gedrückter linker Maustaste.

Zuvor sollte man unbedingt prüfen welche Zoomlevel benötigt werden!

In der Regel beginnt man bei Level 16 und beendet das Laden nach Level 20.

Nachdem man das Rechteck gezogen hat drückt man ‚**Prefetch cache**‘ um den Ladevorgang zu starten.

Man wird für jeden Level (ab dem aktuell eingestellten) gefragt, ob man diesen Level laden will.

Mit 'ja' wird geladen mit 'nein' dieser Level übersprungen - 'Abbrechen' beendet das Vorladen.

Mit **'autom. HP setzen'** wird die Home-Position automatisch gesetzt sobald 5 Satelliten zur Verfügung stehen.

Mit **, HP setzen '** kann die Position manuell gesetzt werden – hierzu wird der Kartenmittelpunkt verwendet. Mit **, HP löschen'** wird die Position gelöscht.

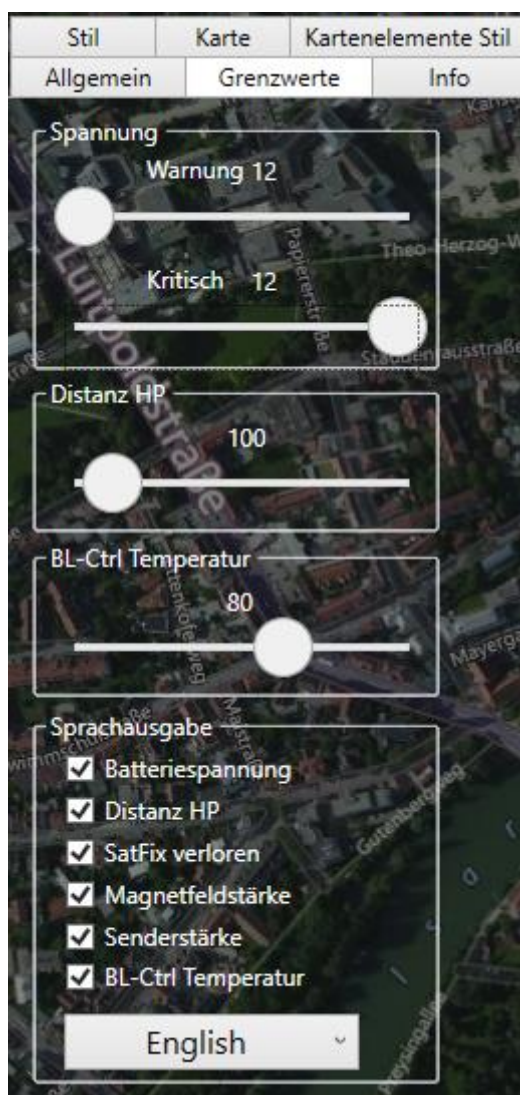
Tip: Man kann die Position auch durch Klick auf das Symbol **'distance to home'** in der Kopfleiste setzen.

'Kopter folgen' zentriert den Kopter in der Mitte der Karte – die Karte bewegt sich mit dem Kopter.

,GPX-Log' zeichnet die Route des Kopters auf sobald dieser abhebt.

Die Daten werden im GPX-Konformen Format gespeichert. So kann man sie in jedem kompatiblen GPX-Betrachter laden. Die gespeicherte Route lässt sich aber auch hier laden.

Grenzwerte



Hier lassen sich Grenzwerte für verschiedene Telemetriedaten einstellen. Im Moment sind Warnschwellen für die LiPo Spannung, die Distanz zur Home Position und die Motorregler Temperatur einstellbar.

Das über-/unterschreiten der Schwellen wird sowohl optisch durch rotes Blinken des entsprechenden Symbols in der Kopfzeile signalisiert als auch durch akustische Meldungen.

Momentan steht die Sprachausgabe in Englisch und in Deutsch zur Verfügung.

Die Sprachmeldungen wurden mit einer [text to speech engine](#) im Internet erstellt.

Tip: Wenn man andere Stimmen oder auch Sprachen für die Sprachausgabe verwenden möchte, kann man die entsprechenden Dateien in dem Ordner "Voice" austauschen.

Man kann zum Beispiel die Sprachausgabe auf der genannten Webseite mit einem Audiorecorder aufnehmen (ich benutze hierzu [Audacity](#)).

Stil der Kartenelemente

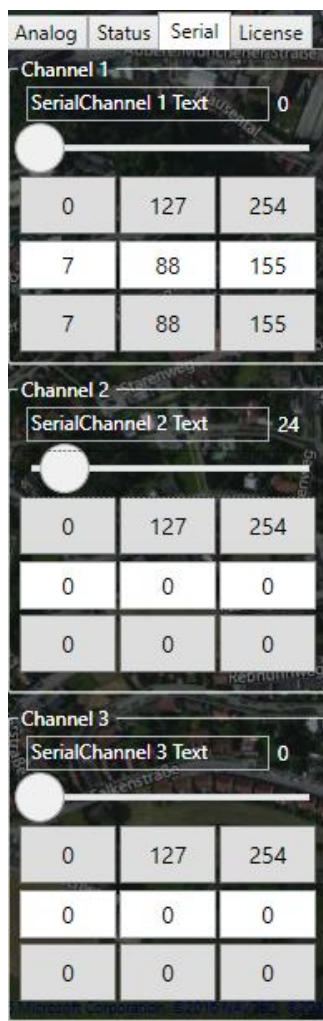


Hier lassen sich die Farben der Kartenelemente ändern.

Hier kann man die Symbolgröße von Kopter, Wegpunkten und Home Symbol anpassen.

Daten

Serielle Kanäle



Die seriellen Kanäle sind eine nützliche Option, wenn einem die Kanäle der Funksteuerung nicht ausreichen.

Mit den seriellen Kanälen können bis zu 8 zusätzliche Kanäle definiert werden.

Allerdings sind hier im Programm nur 3 verfügbar.

Wichtiger Hinweis: Programmbedingt werden immer alle 8 Kanäle gesetzt wenn ein Kanal gesendet wird.

Daher sollte sichergestellt sein, dass keine weiteren Kanäle im Koptertool definiert wurden, da diese dann mit „0“ überschrieben werden.

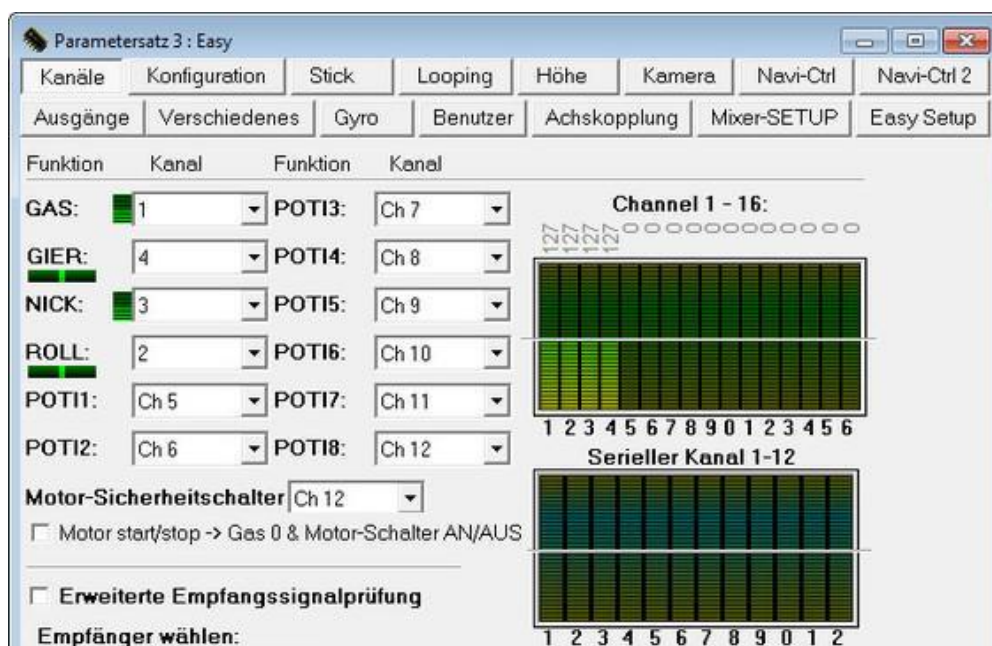
In der 1. Zeile kann man dem Kanal einen sinnvollen Namen geben.

Änderungen an dem Slider werden sofort gesendet.

Die erste Knopfreihe kann entsprechend einem 3-Stufigen Schalter der Funksteuerung verwendet werden.

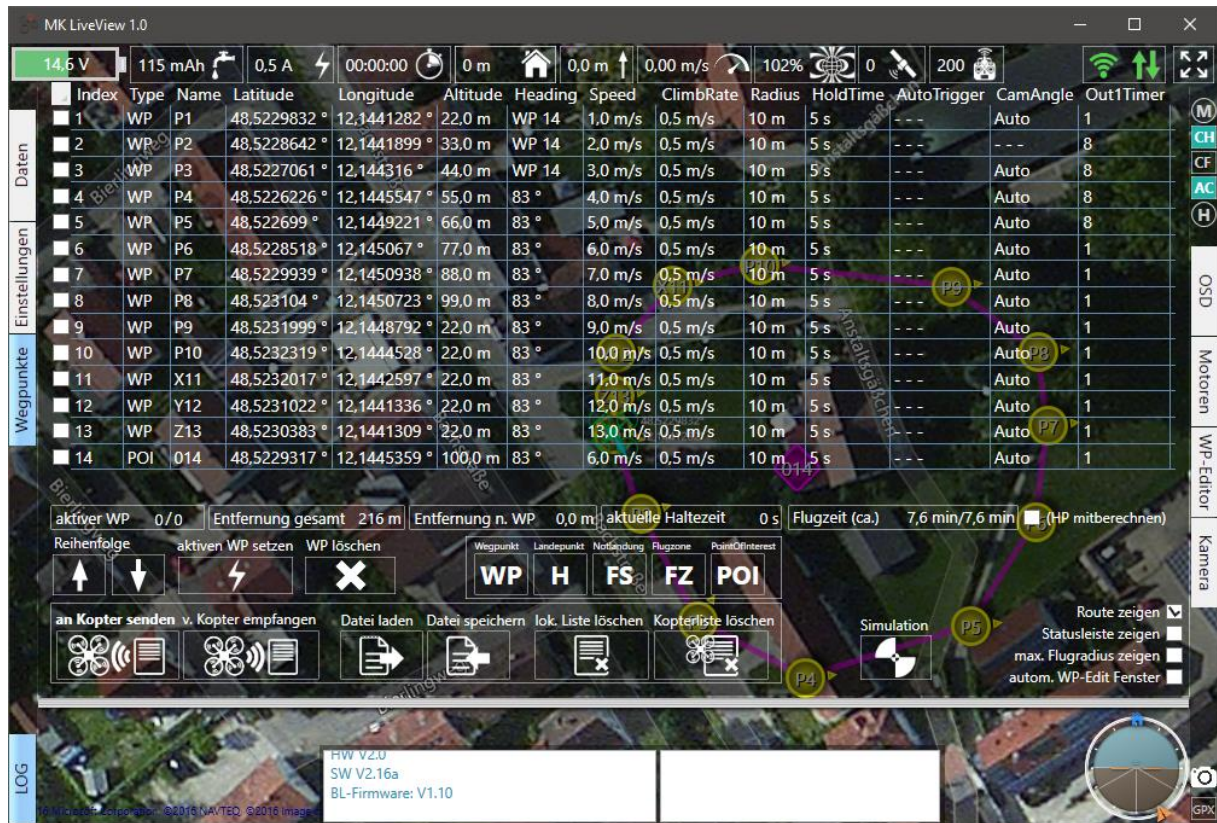
Die Knopfreihe darunter lässt sich frei parametrieren – man kann dort zum Beispiel Werte hinterlegen die man immer wieder benötigt.

Damit man die seriellen Kanäle nutzen kann, müssen diese im MK-Tool zugewiesen werden. Normalerweise weist man sie auf der Kanalseite einem der Potis zu:



Im Tool gibt es noch andere Möglichkeiten die Kanäle zuzuweisen – einfach mal ansehen...

Wegpunkte



Lädt eine *.wpl Datei

Speichert die Liste in eine *.wpl Datei



Sendet die Liste an den Kopter

Lädt die aktuell aktive Liste vom Kopter

Löscht die lokale Liste

Löscht die aktive Liste auf dem Kopter



Simulieren des Wegpunkteflugs bei angeschlossenem Kopter

Anzeigen/ausblenden der Route


Anzeigen/ausblenden der oberen Statusleiste für Wegpunkte

Anzeigen/ausblenden der maximalen Flugweite (entsprechend der installierten Lizenz) Standard: 250m

Automatisches öffnen des Editierfensters bei Auswahl eines Punktes in der Liste

Verschieben des
gewählten Punktes
in der Liste

Löscht den
gewählten WP aus
der Liste

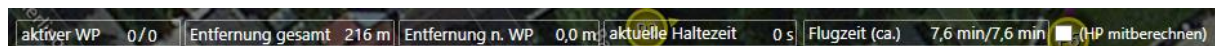


aktiviert den aktuell gewählten WP als aktives Ziel – man kann somit Punkte in der Liste überspringen oder nochmal abfliegen –
Dies funktioniert aber nur ab der aktuellen Firmare (2.16) – der Wegpunkteflug muss (momentan noch) aktiv sein, sonst wird der 1. Punkt der Liste aktiviert.

„WP-Toolbox“
Durch Auswahl eines Punkttyps können durch wiederholten Klick auf die entsprechende Kartenposition schnell mehrere Punkte des gewählten Typs gesetzt werden.

Die Option Flugzone ist noch im Betastadium und funktioniert nur mit der aktuellen Beta version (v2.17a)

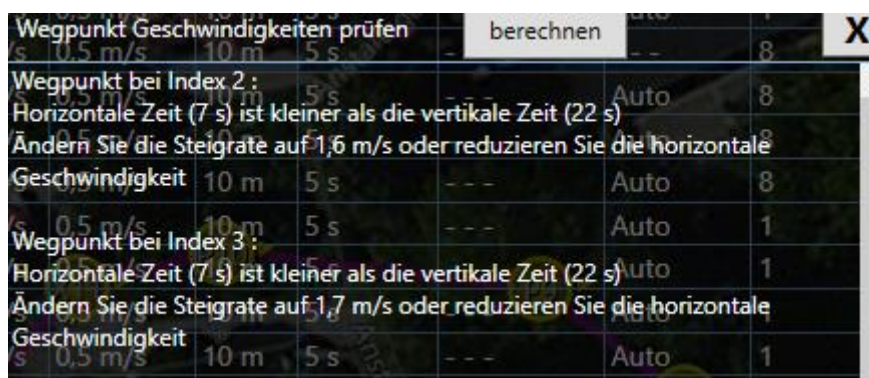
Einige Statuswerte zur aktuellen WP-Liste



Die Flugzeit ist ein grober Richtwert, da nicht alle Faktoren, wie zum Beispiel Wind, Beschleunigung und Abbremsen, miteinberechnet werden können.

HP mitberechnen addiert die Flugzeit von der Homeposion zur Flugzeit hinzu.

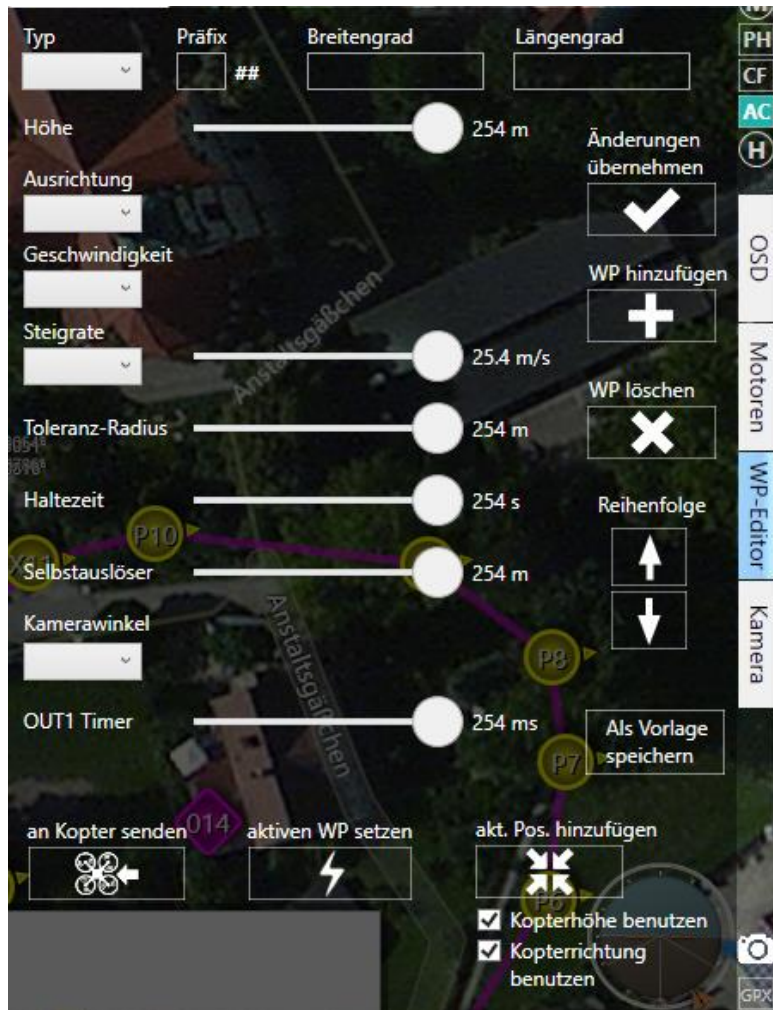
Durch Klick auf das Feld öffnet sich ein kleines Toolfenster:



Hier wird auf Punkte hingewiesen, bei denen der Kopter den Punkt schneller in der Waagrechten erreicht als in der Höhe.

Es wird die notwendige Steigräte angezeigt, damit der Kopter den Punkt gleichschnell in der Waagrechten und in der Höhe erreicht.

Wegpunkte Editierfenster



Hier kann ein Punkt, der in der Liste angewählt wurde, bearbeitet werden.

Eine detaillierte Beschreibung der verschiedenen Werte findet man hier: [MK WaypointEditor wiki](#)

Die Änderungen müssen abschließend mit dem Button **“Änderungen übernehmen”** gespeichert werden.

Mit **‘WP hinzufügen’** kann der aktuell gewählte Punkt ans Ende der Liste dupliziert werden.

‘WP löschen’ löscht den aktuell gewählten Punkt aus der Liste.

Reihenfolge

Mit den **Pfeilen** kann der aktuell gewählte Wegpunkt in der Liste hoch und runter

verschoben werden.

‘Als Vorlage speichern’ speichert die aktuellen Werte als **Standardwerte** – diese Werte werden dann bei der Erstellung eines neuen Wegpunktes verwendet.

‘an Kopter senden’ sendet den aktuell gewählten Punkt an den Kopter – es werden die hier angezeigten Werte gesendet; d.h. man kann auch Änderungen senden ohne diese zu speichern.

‘aktiven WP setzen’ aktiviert den aktuell gewählten WP als aktives Ziel – man kann somit Punkte in der Liste überspringen oder nochmal abfliegen –

Dies funktioniert aber nur mit der aktuellen Firmware (2.16) – der Wegpunkteflug muss (momentan noch) aktiv sein, sonst wird der 1. Punkt der Liste aktiviert.

‘akt. Pos. Hinzufügen’ fügt die aktuelle Kopterposition als neuen Wegpunkt zur Liste hinzu. Abhängig von den gewählten Optionen **‘Kopterhöhe benutzen’** und **‘Kopterrichtung benutzen’** werden die Höhe und Ausrichtung des Kopters mit eingetragen – ansonsten werden die Standardwerte gesetzt.

Verschiedenes

Wegpunkte setzen



Man kann Wegpunkte entweder, wie oben beschrieben, mit der ‚**WP Toolbox**‘ im Wegpunktenfenster, mit ‚**akt. Pos. Hinzufügen**‘ oder durch Doppelklick mit der linken Maustaste (oder Doppeltippen bei einem Touchdisplay) auf die entsprechende Position in der Karte setzen.

Dabei öffnet sich ein kleines Popup-Menü zur Auswahl des Punkttyps.

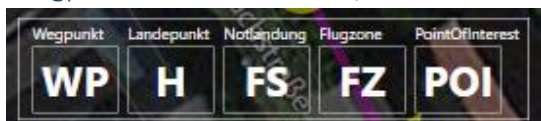
Landepunkt – eine neue Option die nur mit der neuen Firmware (2.16) (und folgende) funktioniert – ein Landepunkt zum automatischen Landen – wird auch als Failsafe Punkt verwendet.

Failsafepunkt – im Störfall wird der nächstgelegene Failsafepoint angefliegen.

POI – Point of interest – kann zur Ausrichtung des Kopters verwendet werden.

FlyZone – Flugzone – eine neue Option die nur mit der neuen beta Firmware (2.17a) (und folgende) funktioniert – hiermit wird ein Polygon/Feld aufgezo- gen, welches der Kopter nicht verlassen darf – näheres dazu siehe in der Beschreibung zur aktuellen Betaversion.

Wegpunkte setzen mit der ‚Toolbox‘



Durch Auswahl eines Punkttyps können durch wiederholten Klick auf die entsprechende Kartenposition schnell mehrere Punkte des gewählten Typs gesetzt werden.

Bei Klick auf einen der Buttons wird das Wegpunktenfenster nach oben verkleinert.



Der Mauszeiger ändert seine Form in ein Kreuz und neben dem Kreuz erscheint das Kürzel des gewählten Punkttyps. Durch erneutes Klicken auf den gewählten Button wird die Funktion wieder deaktiviert.

Wegpunkte bewegen/verschieben

Man kann Wegpunkte einfach mit gedrückter linker Maustaste verschieben – mit Touchdisplay ist das auch möglich, allerdings muss der Finger exakt auf dem Punkt positioniert werden.

Karten Zoom

Der Kartenausschnitt kann mit dem Scrollrad der Maus, dem Zoomslider in der Karteneinstellungsseite oder mit der “Touchzoom Geste” auf einem Touchdisplay vergrößert oder verkleinert werden

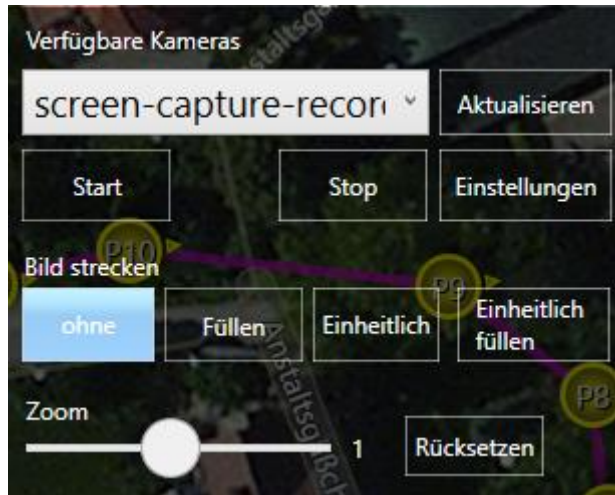
Karte verschieben

Die Karte kann mit gedrückter rechter Maustaste oder mit dem Finger auf Touchdisplays verschoben werden

Kamera

Mit entsprechender Hardware, wie zum Beispiel einem USB-Videograbber, kann das Videosignal der Kopterkamera im Programmfenster angezeigt werden.

Die Hardware muss hierzu vom System als Videodevice erkannt werden.



In der Auswahlliste werden alle vom System erkannten Video-Geräte aufgelistet. Mit dem „**Aktualisieren**“ Button kann die Liste aktualisiert werden, falls ein Gerät bei laufendem Programm angesteckt wurde.

Mit dem „**Start**“ Button wird die Videoschnittstelle initialisiert und der Videostream in einem Bildfenster angezeigt.

Je nach Gerät können verschiedene Einstellungen wie Helligkeit und Kontrast durch Klick auf „**Einstellungen**“ gemacht werden.

Das Videobild kann entweder in einem kleinen Bildfenster angezeigt werden oder auch als füllendes Hintergrundbild.

Das kleine Bildfenster kann frei im Programmfenster positioniert und in der Größe angepasst werden.

Hierzu muss unter „**Bild strecken**“ die Option „**ohne**“ gewählt sein.

Der **Zoom Slider** funktioniert nur für diese Option.

Falls das Bildfenster mit der Maus nicht mehr erreichbar sein sollte, kann man die Einstellungen mit „**Rücksetzen**“ wieder zurücksetzen.

„**Füllen**“ füllt das komplette Programmfenster – das Seitenverhältnis wird dabei außer acht gelassen.

„**Einheitlich**“ füllt das Programmfenster soweit wie möglich unter Berücksichtigung des Seitenverhältnisses.

„**Einheitlich füllen**“ füllt das komplette Programmfenster – das Seitenverhältnis bleibt erhalten – dadurch können Bereiche des Videobildes abgeschnitten sein.

Zusätzliche Software:

Der künstliche Horizont wurde mithilfe folgender Fragestellung erstellt: [Creating an advanced HUD](#)

Die Kartenanzeige und Funktionen wurden durch [GreatMaps](#) ermöglicht – really great!
Ich habe einige Anpassungen für das Progeamm gemacht. Daher ist es notwendig die geänderten Quelldateien zu verwenden, falls man das Projekt neu kompilieren möchte. Alle Quellen findet man im [MK svn](#).

Die Kamerabibliothek [Versatile Web Cam Library by lepipele](#)

Und schließlich Teile aus dem Projekt „**Mikrokopter Serial Control Tutorial**“ (<http://hdl.lib.byu.edu/1877/2748>)

Ideen und Anregungen sind ausdrücklich erwünscht!

Guten Flug & Cheers!
Steph

Obligatorisch

```
///=====
/// 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...)
/// =====
```