MikroKopter LiveView

Dokumentation

V1.1 September 2016

Allgemein

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

Die Daten werden hierbei mittels einer seriellen Verbindung zwischen dem PC und dem Debug-Anschluss des Mikrokopters übertragen.

Das Programm wurde vornehmlich für die Kommunikation mit der Navi-Ctrl optimiert, kann aber auch zur alleinigen Anzeige der Daten der Flight-Ctrl verwendet werden.

Zum Zeitpunkt der Programmerstellung hatte die verwendete Firmware des Mikrokopters folgende Versionen:

Flight-Ctrl: V2.14c

Navi-Ctrl: V2.14e

BL-Ctrl: V1.10

Die Software wurde auf folgender Hardware getestet:

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

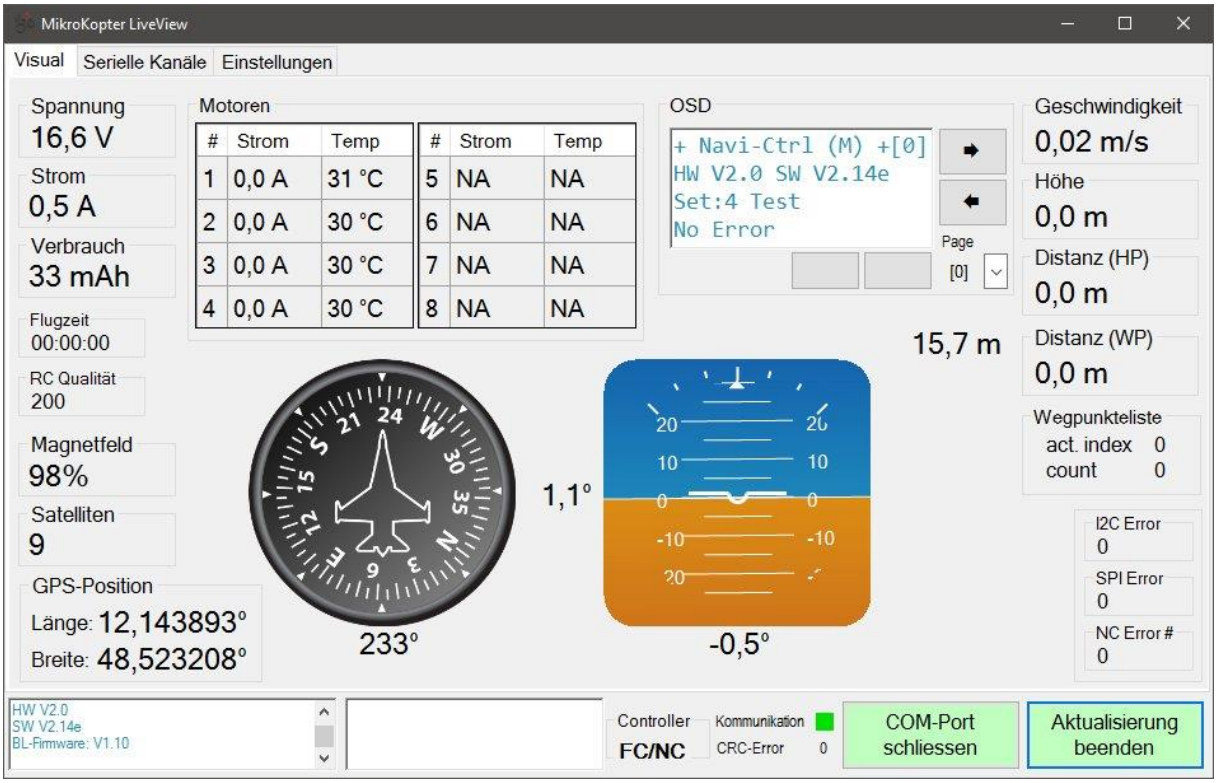
Navi-Ctrl: V2.0

BL-Ctrl: V3.0

Das Programm wurde mit VisualStudio 2015 in C# erstellt.

Zur Umsetzung hat wesentlich das „Mikrokopter Serial Control Tutorial“ (http://hdl.lib.byu.edu/1877/2747) von John C. Macdonald beigetragen, der ein Projekt zur Steuerung eines Mikrokopters mittels PC in C# erstellt und dokumentiert hat.

Anzeige und Bedienung



Das Startfenster im Tab „Visual“ dient zur Anzeige der Telemetriedaten.

Ist keine Navi-Ctrl vorhanden, oder die Flight-Control angewählt, werden die nicht verfügbaren Daten als „NA“ angezeigt.

Die Ströme und Temperaturen der BL-Ctrl können momentan auch nur über die Navi-Ctrl abgerufen werden. (Ich habe allerdings eine Erweiterung für die FC-Source geschrieben, mit der es dann auch möglich ist die Motoren Ströme und Temperaturen auszulesen. Evtl. wird sie beim nächsten Release integriert…)

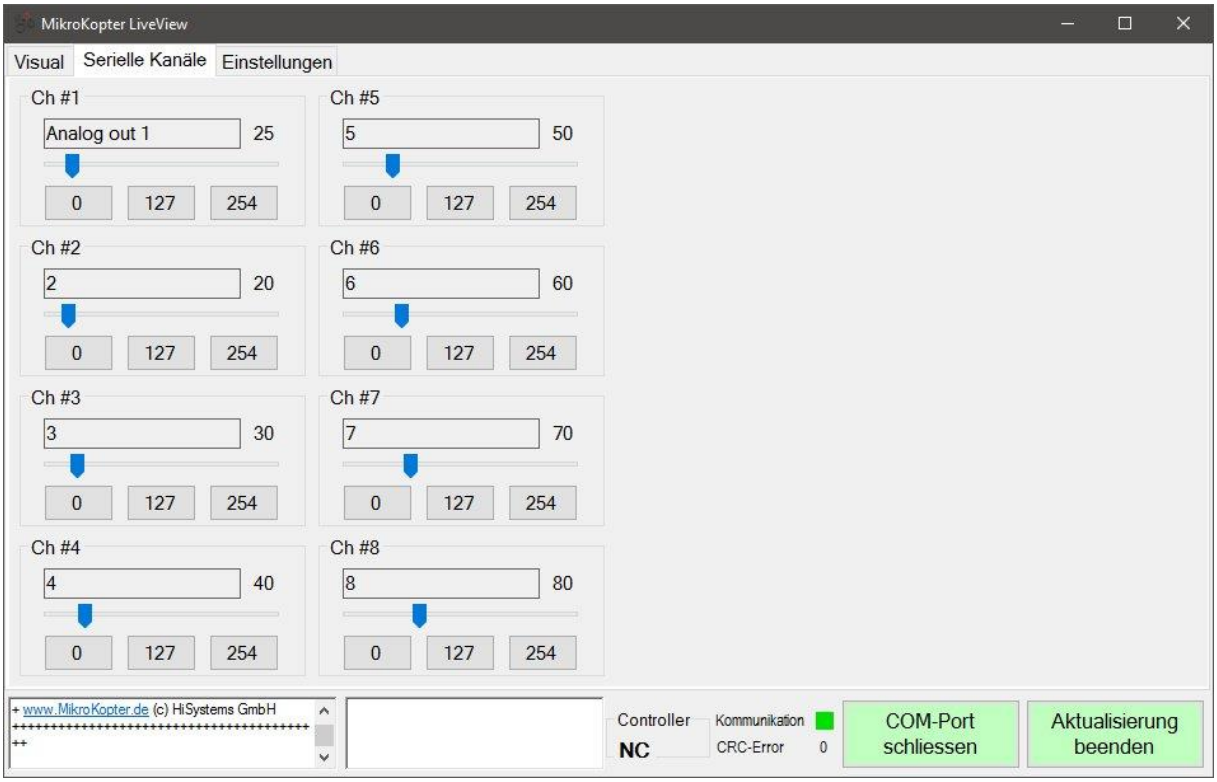
In der unteren Leiste befindet sich links ein Logfenster mit allgemeinen Nachrichten des aktiven Controllers und des Programms.

Rechts daneben werden Störmeldungen des Kopters angezeigt.

Der Button rechts daneben öffnet oder schließt die serielle Schnittstelle (COM-Port öffnen/schliessen)

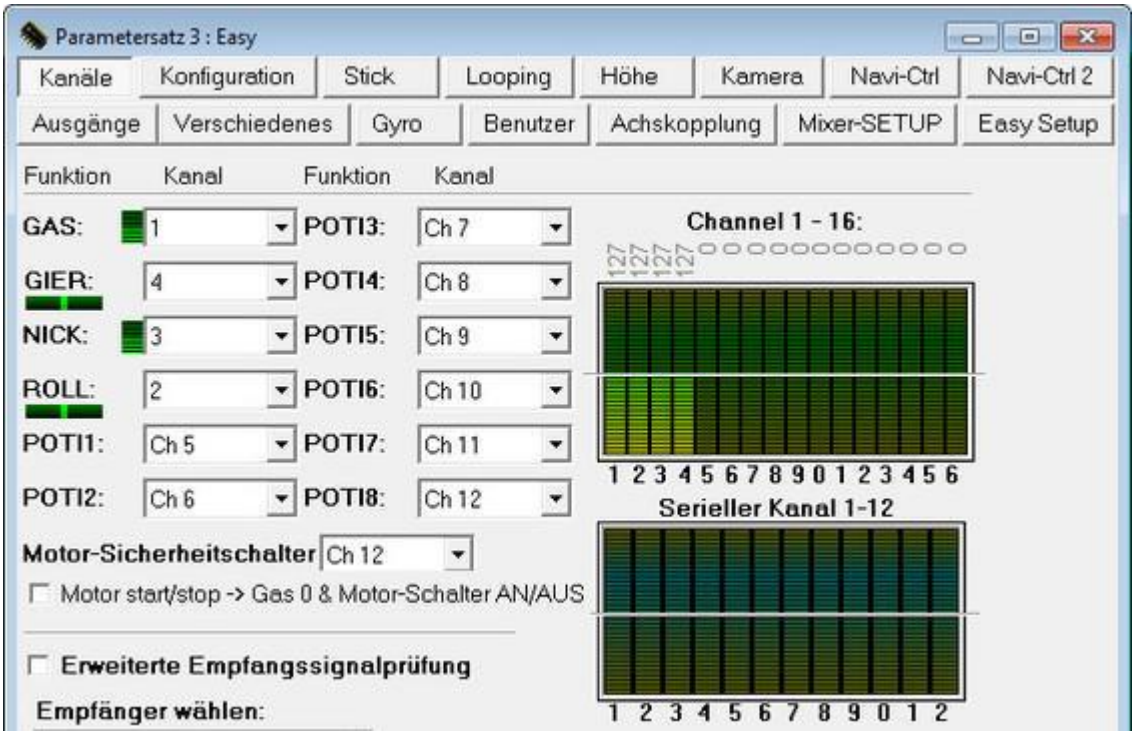
Über den ganz rechten Button (Aktualisierung starten/beenden) wird zyklisch eine Anforderung der Daten an den Kopter gesendet.

Das OSD-Fenster entspricht dem OSD wie im Mikrokopter Tool.
Man kann hier wie gewohnt blättern oder über das Dropdown Menü direkt auf eine Menüseite springen.
Funktionen werden über die beiden Buttons unterhalb bedient.



Auf der Seite „Serielle Kanäle“ hat man die Möglichkeit bis zu 8 serielle Kanäle zu steuern.

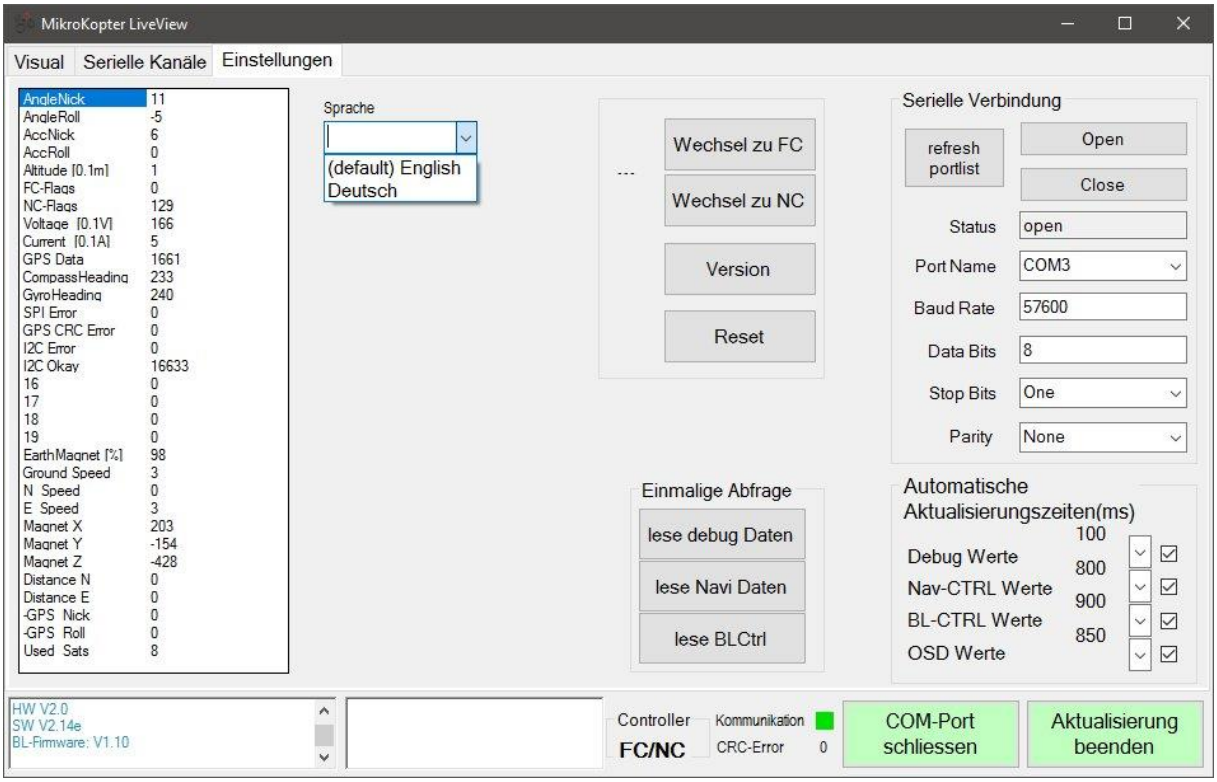
Hierzu müssen die Kanäle zuerst im MK-Tool zugeordnet werden. Im Normalfall ordnet man die benötigten seriellen Kanäle einem der 8 Potis auf der Seite Kanäle zu:



Es gibt aber auch noch andere Funktionen, denen man einen Seriellen Kanal zuordnen kann… einfach mal schauen.

Als Anwendungsmöglichkeit könnte man zum Beispiel das Timing des Analogausgangs steuern.

Die eingestellten Werte und Texte werden bei Beendigung des Programms gespeichert und beim nächsten Start wieder geladen.



Im dritten Tab „Settings“ können Einstellungen zur seriellen Schnittstelle und dem zyklischen Datenabruf gemacht werden.

Im Fenster links werden alle Analogdaten des aktuellen Controllers angezeigt.

Bei der seriellen Schnittstelle sollte nur, wenn nötig, der verbundene Port angepasst werden. Die anderen Einstellungen sollten bis auf weiteres so passen.
Mit dem Button „refresh portlist“ kann man die Seriellen Schnittstellen nochmals einlesen, falls zum Beispiel nach dem Start des Programms ein USB-Adapter angeschlossen wurde, über welchen die Kommunikation stattfinden soll.

Bei den Automatischen Aktualisierungszeiten für das Autoupdate der Daten können die Zyklen in denen der Kopter-Controller die Daten sendet eingestellt werden. Die abzufragenden Datenstrukturen können auch ganz deaktiviert werden.
Mit den Timings sollte man vorsichtig sein, da sie zum einem den Prozessor des Kopters unnötig belasten können, wenn sehr kurze Intervalle gewählt werden. Zum anderen wird auch der Anzeigefluss gestört, wenn zu oft große Datenpakete wie das OSD-Menü abgerufen werden.

Aber am besten man probiert es aus – es ist sicherlich auch vom eigenen Rechner abhängig wie schnell die Daten ausgewertet und angezeigt werden können.

(Für eine flüssige Anzeige des Horizonts und des Kompasses sollte das Timing für die debug Werte auf 100 ms eingestellt sein).

Für das Programm sind 2 Sprachanzeigen möglich – der Standard ist Englisch. Als weitere Sprache steht Deutsch zur Verfügung. Ist das Gebietsschema des Rechners auf dem das Programm ausgeführt wird Deutsch (de-DE), wird die Sprache automatisch beim Start geladen. Man kann aber auch hier die Sprache nach dem Start des Programms ändern.

Mit den Buttons „Wechsel zu FC“ und „Wechsel zu NC“ kann man, wie der Name schon sagt, den aktiven Controller für den Datenabruf setzen. (wie auch im MK-Tool)
Im Flug habe ich das noch nicht getestet – es könnte aber sein, dass der Kopter nicht so ‚freundlich‘ reagiert – also bitte Vorsicht walten lassen.

„Reset“ macht einen Warmstart des Controllers – **meines Wissens ist die Funktion während des Flugs gesperrt – ich würde es aber nicht unbedingt darauf ankommen lassen…**

Mit „Einmalige Abfrage“ kann man die entsprechenden Datenstrukturen einmalig anfordern. (Ist eher zum Testen gedacht und werde ich wohl entfernen…)

Da aber eine Anforderung beim Controller für 4 Sekunden ‚abonniert‘ wird, kommen die Daten auch 4 Sekunden lang.

Sonstiges

Zusätzliche Software:

„**Avionic Instrument Controls**“ von Chootair
(http://www.codeproject.com/Articles/27411/C-Avionic-Instrument-Controls)

Hier habe ich mich des Codes für die Kompassanzeige bedient – der Kompass selber wurde mit Google Bildsuche und Gimp erstellt.

„**Artificial horizon**“ von Tom Pycke (http://tom.pycke.be/mav/100/artificial-horizon)
Sehr schön gemacht!

Und natürlich die Quellen zum „**Mikrokopter Serial Control Tutorial**“
(http://hdl.lib.byu.edu/1877/2748)