

Anleitung zu „Udo's Suite“



Inhaltsverzeichnis

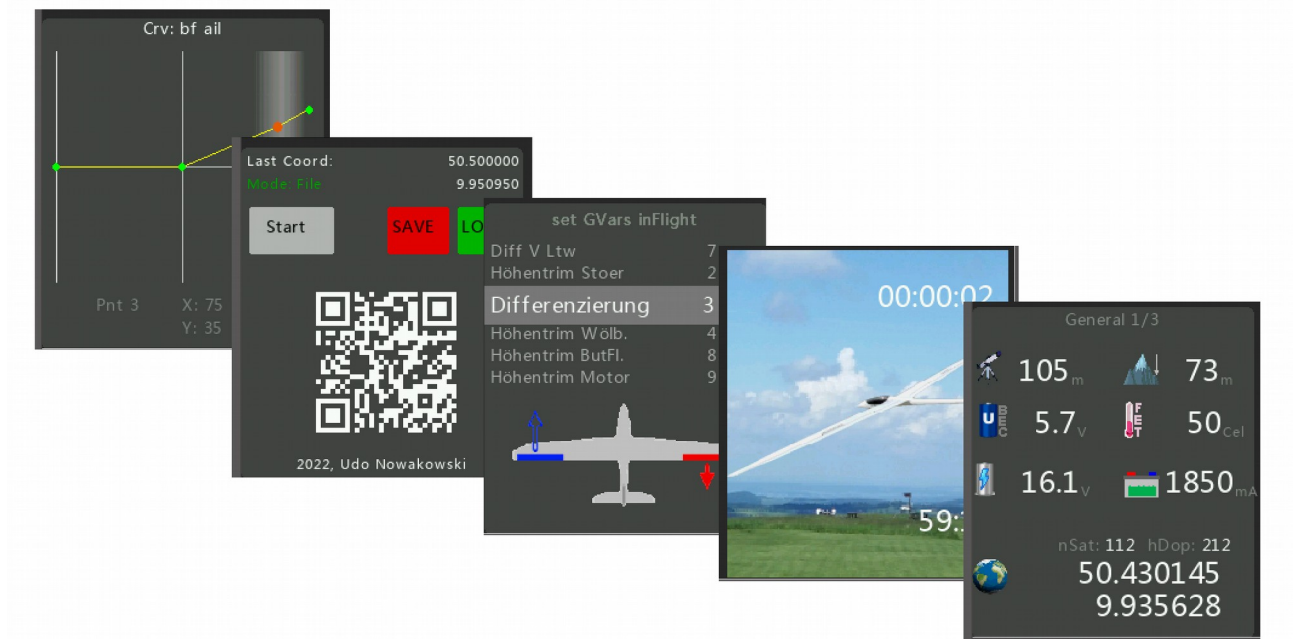
Einleitung.....	3
Allgemeines.....	3
Was ist das besondere ?.....	3
Grundidee:.....	4
Funktionsweise:.....	6
Die einzelnen Apps:.....	7
Telemetrie.....	7
Modell Image.....	7
Kurveneditor.....	8
QR Modelfinder.....	8
TopBar „Full“.....	9
TopBar „Standard“.....	9
TopBar „light“.....	9
(Future: GVAR editor).....	10
(Future: Ansagen).....	11
Installation:.....	12
(1) Dateien & Ordner kopieren.....	12
(2) Screenlayout auswählen.....	13
(3) widget allgemein einrichten.....	14
(4) einzelne Apps konfigurieren.....	14
Telemetrie.....	14
Picture.....	14
Kurveneditor.....	14
TopFull.....	15
Anhang.....	16
Modellspezifische Konfiguration.....	16
Standardsettings der Apps.....	16
Der „Telemetrikatalog“.....	16
Landesspezifische Texte / Sprachdateien.....	16
Beispieltemplate „Alpina“.....	17
Sonstige Optionen.....	18
Sprachdateien.....	18
Sensorverzeichnis.....	18
Modellspezifische Parameter.....	18
Internals.....	19

Einleitung

Allgemeines

Udos Suite ist eine Sammlung von Ethos widgets für Tandem Sender von Frsky.

Es existieren z.B. Widgets für umfangreiche Telemetrie Anzeigen, QR Code basierte Google Maps „Modellfinder“, ein Kurveneditor der im Flug mittels Akustik feedback genutzt werden kann, openTx ähnliche Topbars etc...



Die Tandem Sender unterstützen grundsätzlich die Touch Bedienung, was auch in den widgets genutzt wird

Was ist das besondere ?

Herkömmliche Widgets unter Ethos stellen EIN lua „Programm“ dar, welches sich in einen vom Anwender zuvor definierten „Rahmen“ konfigurieren lassen.

Das Layout der Rahmen gibt die Anzahl von widgets per screen vor.

Möchte man mehr widgets implementieren, muss man diese auf verschiedene Seiten platzieren und die Seiten immer kompoett umschalten.

Zur Zeit des Verfassens dieser Anleitung werden auch noch keine TopBars wie bei oTx üblich unterstützt.

Diese Einschränkungen werden durch die Suite aufgehoben.

Bereits in der einfachsten Form können in einem Rahmen 3 Apps abgerufen werden.

Die „maximale“ Konfiguration lässt zwei Rahmen mit jeweils 3 Apps sowie eine (ggf individuell gestaltbare) TopBar zu.

Die einzelnen Apps besitzen eigene „Sprachfiles“, so dass sich beliebige Sprachdarstellungen umsetzen lassen (zur Zeit deutsch und englisch)

Grundidee:

Ich komme von oTx.

Dieses OS konnte ich extrem gut auf meine persönlichen Bedürfnisse zuschneiden.

Eine besondere Rolle spielten dabei die widgets.

Auf Basis von lua wurde man in die Lage versetzt selbst kleine „Apps“ zu schreiben, die genau das machten was man von einer guten Senderanlage erwarten würde.

Also entwickelte ich auf mich zugeschnittene Apps

Sie dienten mir z.B.

- zum einen als umfangreicher Informationsbildschirm (Darstellung etlicher Sensorwerte innerhalb eines Frames),
- zum Feintrim des Modells im Flug (Selektion einer gewünschten Globalen Variable per Schalter, Einstellen per Poti, Selektion der nächsten Gvar, alles mit Ansageunterstützung um nicht auf den screen schauen zu müssen)
- zur erhöhten Sicherheit durch Signalisierung von Zuständen einiger LSW's die Mischer bedienen, oder dem Zustand des Motor „Kill switches“ bzw eines möglichen Motoranlaufs in der Topbar..

Standardmässig nutzte ich auf einer X12 das Layout zur Darstellung mit 4 Widgets pro Seite und TopBar. Durch den Wechsel auf eine X18 unter Ethos ergaben sich einige Änderungen

- Ethos selbst unterstützt derzeit keine eigene „Topbar“
- der X18 Bildschirm ist kleiner, wodurch das „4 widget layout“ erheblich an Präsentationsfläche verliert
- Ethos unterstützt unter lua aber auch touch Ereignisse auf den Tandem Sendern.

Der Wechsel auf die X18 war eine bewusste Entscheidung, die Eigenschaften bzw. Pros und Cons waren vorher klar, nun galt es daraus eine Umsetzung zu entwickeln, die der oTx Lösung um nichts Nachstand sondern sie „übertrumpfte“

Daraus entstand folgendes Grob-Konzept / Anforderungsprofil:

- Anstelle von vier zuvor „kleinen“ widgets auf der X12 werden nur noch zwei in dafür „grösseren“ Frames auf der X18 dargestellt
- per Touch lassen sich innerhalb eines Frames mehrere Apps aufrufen
- per Touch lassen sich innerhalb einer App mehrere Seiten aufrufen
- da keine generische Ethos „lua TopBar“ möglich ist, wird eine eigene programmiert.
- einfache Adaption auf andere Bildschirmauflösungen wie X20 und zukünftige Hardware
- unterstützung mehrerer Sprachen durch selbst editierbare Sprachdateien
- mehrere „Themes“, individuell anpassbar

Letztendlich bedeutete das auch eine eigene widget Verwaltung, im Ergebnis aber deutliche Vorteile gegenüber der oTx Lösung:

- pro widget selbst auf der X18 eine grössere Präsentationsfläche
- weniger HW Schalter und pods notwendig, da einige Funktionalität per touch abrufbar ist
- ein höheres Mass an Individualisierung
- eine deutlich verbesserte Konfigurationsmöglichkeit der Widgets unter lua durch den „Config Handler“

Das ganze sollte dann im Fullscreen Layout laufen

Während der Entwicklung kam es noch zur Idee, das ganze auch in einem einzelnen Frame (Suite1) bzw ohne eigene Topbar in zwei Frames (Suite2) zu adaptieren.

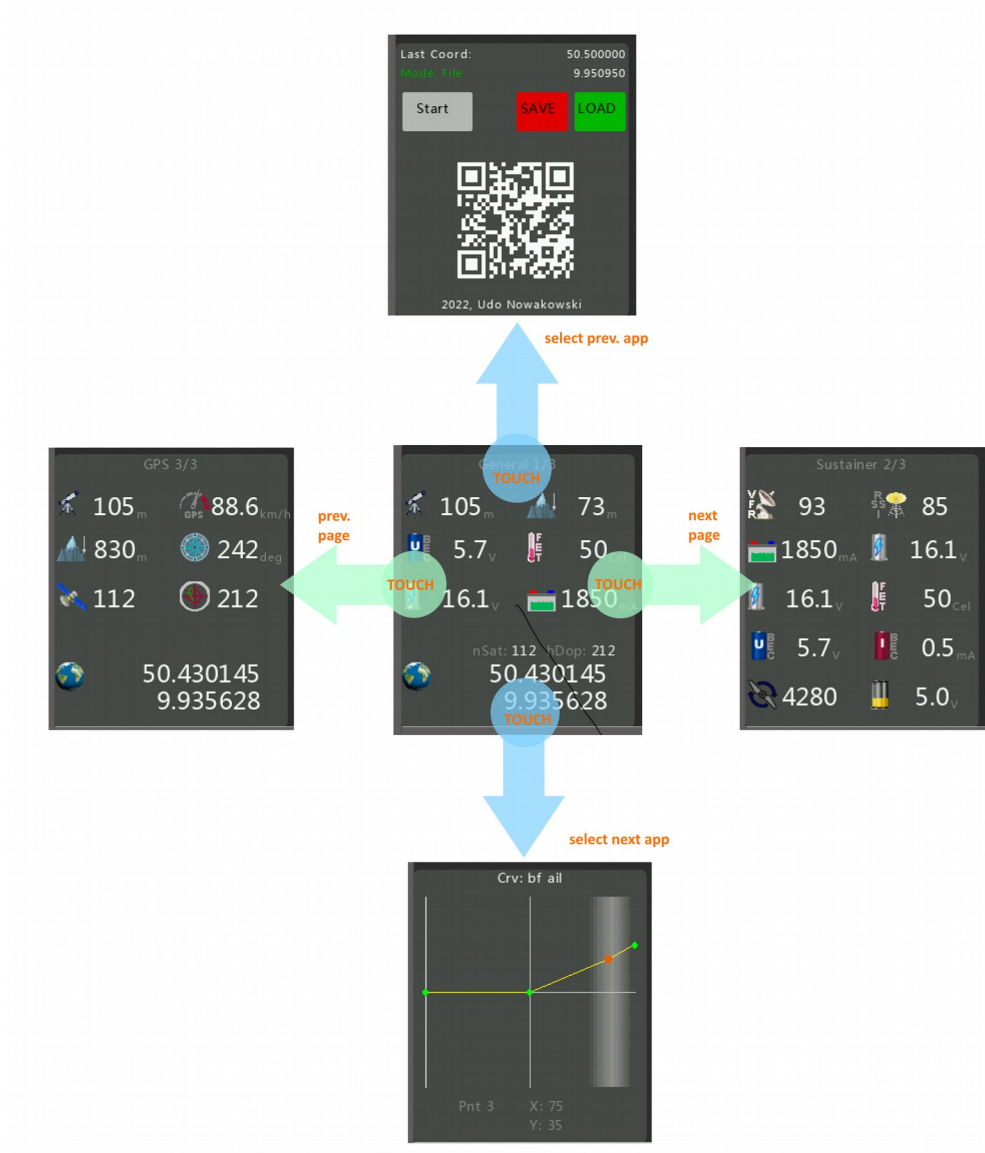
Das Ergebnis ist nun diese Sammlung von Apps, die ich „Udo's Suite“ titliert habe und durch ein übergeordnetes Programm (wrapper) ansteuere

Innerhalb eines Rahmens/Frames lassen sich bis zu drei Apps konfigurieren.
Jede App kann bis zu drei Seiten Umfang besitzen.

In den Frames existieren „reservierte“ Touch Bereiche, um zwischen den Apps umschalten zu können, bzw innerhalb einer App blättern zu können.

Im u.a. Bild ist im Zentrum der Hauptbildschirm der ersten (Telemetrie) App zu sehen.
Um auf eine andere Telemetrie-seite zu gelangen drückt man in die grün markierten Bereiche.
Links wir zurück geblättert, rechts wird vorwärts geblättert.

Die Blau markierten Bereiche ermöglichen es in eine andere App zu springen.
Oben wird die vorherige App angewählt (Google Maps Modelfinder), unten die nächste App (Kurveneditor)

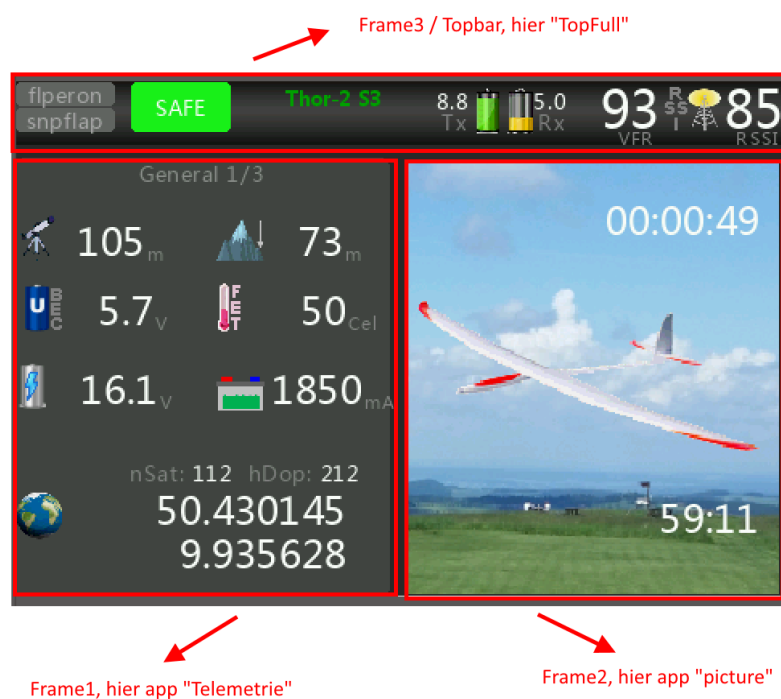


Funktionsweise:

Das „Hauptprogramm“ dient dazu die aktuell ausgewählte app darzustellen, gibt dieser app die „Arbeitsfläche“ (den Frame) bekannt, überprüft touch Eingaben und wertet diese aus, sowie startet Hintergrundaufgaben von apps, falls diese es erfordern.

Das Hauptprogramm „umschliesst“ logisch die eigentlichen apps und organisiert ihren Ablauf, stellt quasi eine eigene widget-Verwaltung dar.
Der Fachausdruck dieses Types von Programm ist „wrapper“

Werden mehrere Apps/Frames bedient, werden die dazugehörigen Apps nacheinander abgearbeitet.



Bsp: Suite3 im „Fullframe“ modus mit drei apps/frames

Die Apps selbst werden in der widget Konfiguration parametrisiert.

Da zu Anfang der Wrapper gar nicht weiss, welche Apps der Anwender wählen wird, sieht man bei der Grundinstallation lediglich eine Auswahl für das Farbschema („thema“) und die Auswahlfelder für die Apps (maximal drei per frame).

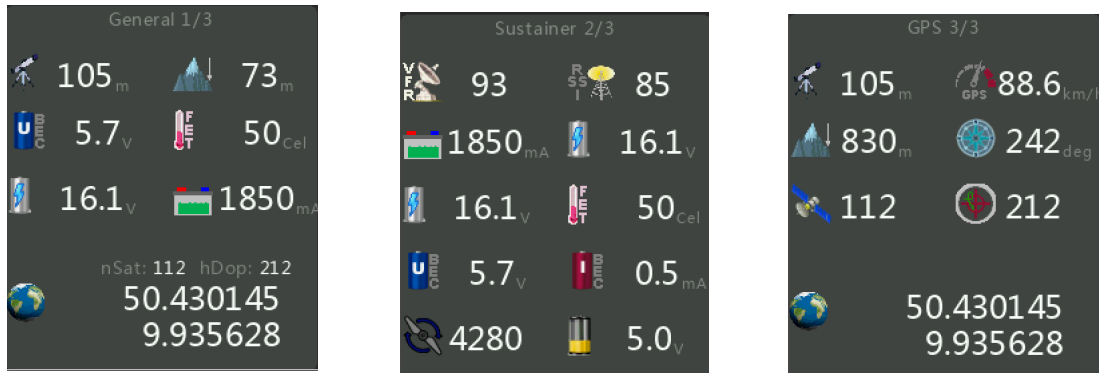
Sobald eine App in der Konfig angewählt wurde, erscheinen die dazugehörigen Felder zur Parametrisierung.

Ist eine App durchkonfiguriert, kann das Konfigmenu verlassen werden und die entsprechende App startet im Frame.

Die Apps werden also dynamisch in die Oberfläche via wrapper eingebunden.

Die einzelnen Apps:

Telemetrie



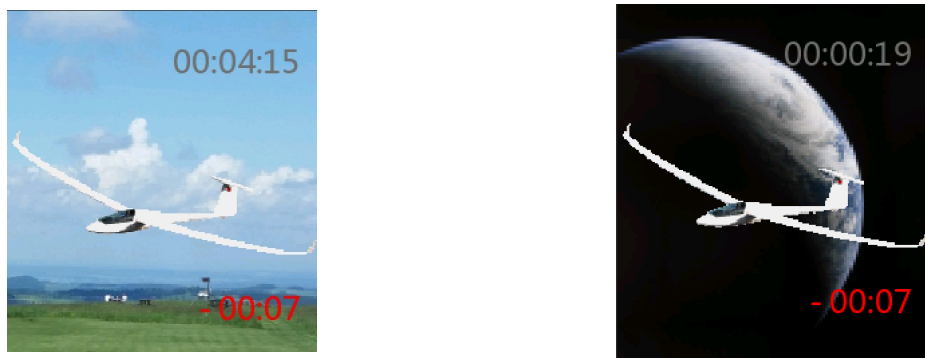
Die Telemetrie App präsentiert etliche Sensorwerte individuell in einem Frame.

Die Werte können über bis zu drei Seiten dargestellt werden.

Ein Wert kann mehrfach präsentiert werden

Die komplette Konfiguration, welcher Wert wo platziert ist, wird als „set“ in einer Datei hinterlegt und kann so von etlichen Modellspeichern aus genutzt werden, ohne jedesmal neu konfigurieren zu müssen

Modell Image



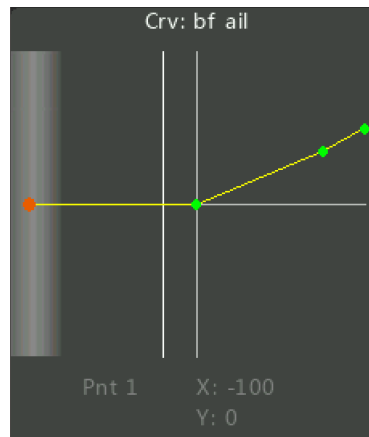
Die App zur Darstellung eines Modellfotos unterscheidet sich im wesentlichen durch zwei Punkte vom Standardwidget:

- es lassen sich Modellfoto (als freigestelltes png Format) und der Hintergrund unabhängig voneinander wählen
- Da das Bild „alleine“ viel Fläche des Displays nutzt können optional zwei Timer zusätzlich eingeblendet werden (ich nutze meistens den FlugzeitTimer sowie den MotorlaufzeitTimer)

Bildgrößen werden automatisch in den Frame skaliert

Kurveneditor

Der Kurveneditor dient dazu eine Kurve im Flug anpassen zu können.



Dazu wird der gewünschte Kurvenpunkt z.B. durch einen Trimmer angewählt, mittel des „zur Kurve gehörigen Inputs“ wird der Punkt angesteuert (akustik feedback, so dass man nicht auf das Display schauen muss), ein analog Geber ermöglicht den Feintrim auf die gewünschte Position.

Per Schalter lässt sich der Urzustand wiederherstellen (reset).

Ideal um z.B. die HR-Kompensationskurve im Butterfly anzupassen u.a.

QR Modelfinder

Ideal wenn ein GPS im Modell verbaut ist.



Die app „merkt“ sich die letzte empfangene GPS Position (diese Routine arbeitet auch, wenn die App nicht im Vordergrund läuft).

Per Touch lässt sich der Wert in einen QR Code darstellen

(Achtung lange rechenzeit, z.B. 15 Sekunden)

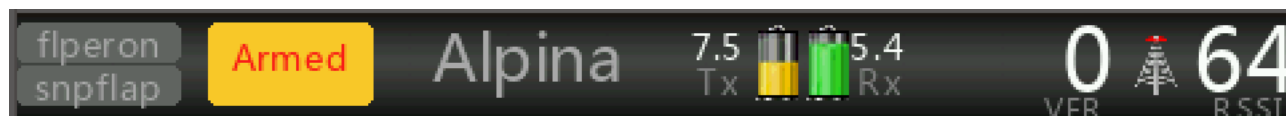
Der QR Code kann per neuerem Handy abfotografiert werden, Google maps öffnet sich dann automatisch mit der Positionsangabe auf der dazugehörigen Kartenregion

Abspeichern der Koordinaten & Wiederaufruf sind möglich

Die App ist eine Portierung des „standalone“ widgets.

TopBar „Full“

Diese Topbar enthält die komplette Funktionalität, wie ich sie unter oTx genutzt hatte.



Zwei schmale Schaltflächen zeigen, ob ich snapflap bzw Flaperon Mischer (Wölkappen als Querruder) aktiviert habe

Eine grössere Schaltfläche signalisiert, ob mein Sicherheitsschalter den versehentlichen Anlauf des E-Motors unterbindet, wenn er aktiv (Motor „unscharf“) ist und der Motor-Input eauf aus steht, ist die Anzeige grün. Sollte man auch bei unscharfen Sicherheitsschalter einenMotoranlauf gesetzt haben, blinkt die Fläche rot und gibt Warntöne aus (Sicherheitsrisiko, da bei deaktivieren des Schalters der Motor SOFORT anläuft)

Ist der Schalter deaktiviert, also der Motor „scharf“, wird eine Orange Darstellung gewählt.

Relativ Mittig steht der Modellname

Dann folgt eine Darstellung der Sender und Empfängerspannung. Die Symbole ändern sich je nach Spannungslage Grün auf Gelb nach Rot. Die Empfängerspannungsgrenzen sind modellspezifisch in einer Datei hinterlegbar.

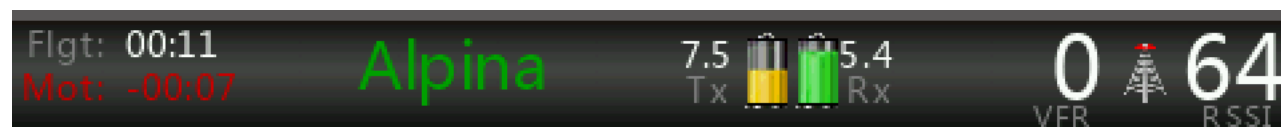
Ganz rechts davon werden die VFR sowie RSSI Empfangsgüten dargestellt.

Das Symbol ändert die Farbe / den Status auf den „niedrigeren“ Status beider Sensoren.

Bsp: RSSI ist 55 (gelb), aber VFR 98 (grün) >> Status gelb

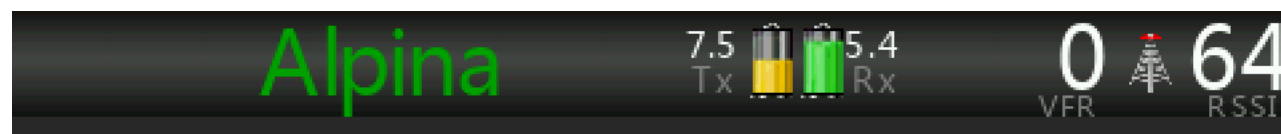
TopBar „Standard“

Wie TopBar „full“, jedoch ohne Mixer-Signalfächen bzw Schutzschalter Display, dafür mit zwei Timern



TopBar „light“

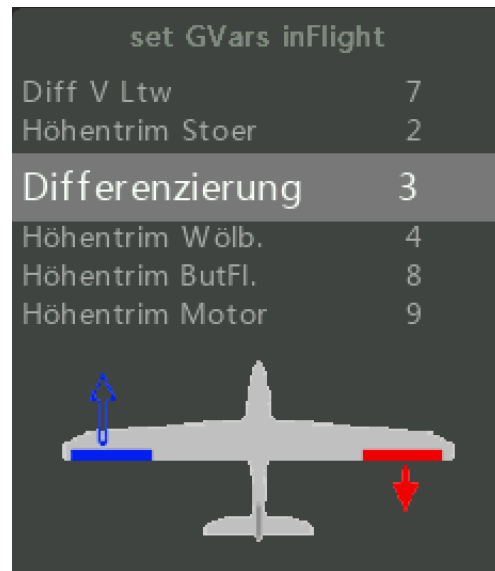
Eine sehr einfache Variante, lediglich mit Modellname, Akkuspannungen und Empfangsgüte



(Future: GVAR editor)

Dies ist ein Vorentwurf zur Anpassung von Globalen Variablen im Flug

(aufgrund Akustik feedback wird keine Notwendigkeit existieren dabei auf den Sender zu schauen).



Über einen Selektionsschalter (z.B. Trimmer / Tastschalter) wird die gewünschte Gvar angewählt, es erfolgt eine Durchsage der Selektion.

Dann kann via Poti der Wert innerhalb gewisser Grenzen (z.B. +/-20) angepasst werden, die Anpassungen werden sofort auf das Modell angewendet.

Ist der richtige Wert gesetzt, kann mit erneuter Wahl des Selektionsschalters der Wert final „gespeichert“ werden, die Anpassung via Poti wird dadurch geblockt.

Das Poti MUSS nun wieder in die Neutralposition zurückgesetzt werden (Piepston & Ansage „Go“ wenn erreicht)

Nun kann die nächste gewünschte Gvar selektiert werden.

Ein „Resettaster“ ermöglicht das Rücksetzen auf die (zuvor intern gespeicherten) Ursprungswerte.

Dieses widget ist zunächst ein Preview und wird erst im Laufe von Ethos 1.5 fertiggestellt.

Noch sind nicht alle notwendigen Methoden unter lua Implementiert (Zugriff auf GVars)

(Future: Ansagen)

Dieser Vorentwurf generiert zyklische Ansagen von Sensorwerten.

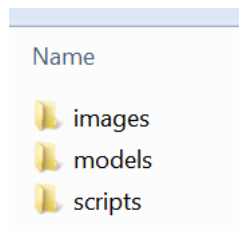
Dadurch können z.B. Spannungs- / Kapazitäts- / Höhen-/ Entfernungsansagen etc.. in regelmässigen zeitlichen Abständen durchgeführt werden, ohne Logische Schalter bzw Spezialfunktionen konfigurieren zu müssen.

Welche Sensoren angesagt werden sollen bzw welche Zykluszeiten gewünscht sind lassen sich in der Konfig bzw per touch einstellen

Installation:

(1) Dateien & Ordner kopieren

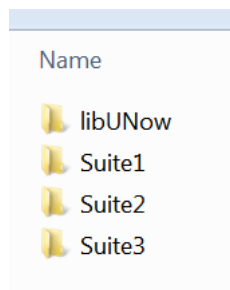
Der Download besteht (im git-Unterordner „src“) aus einer Reihe von Ordnern (sowie Unterordnern), die auf das entsprechende „Laufwerk“ des Senders kopiert werden müssen.



Im Folgenden wird das Haupt- bzw Wurzelverzeichnis des Senders mit „/“ gekennzeichnet

Bitte die Ordner wie durch die Downloadstruktur vorgegeben auf den Sender kopieren:

- „images“ in das Hauptverzeichnis
- alle Ordner unterhalb von „scripts“ in das „/scripts“ Verzeichnis des Senders
das sind die Ordner „libUNow“, „Suite1“, „Suite2“ sowie „Suite3“



- wer möchte kann sich aus dem Ordner „models“ die Modellvorlage „scale“ in das /models Verzeichnis kopieren, das template enthält eine durchkonfigurierte Beispielvorlage
- Sender durchstarten, damit die scripte „bekannt“ gemacht werden

(2) Screenlayout auswählen

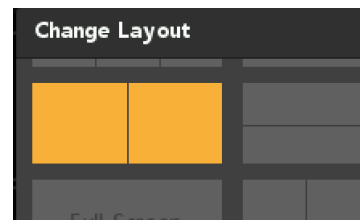
Wie üblich wird ein widget einem Frame / Rahmen unter Ethos zugeordnet.
Ethos bietet bekanntermassen eine Vielzahl verschiedener Layouts an.

Udo's Suite unterstützt folgende Layouts / Rahmen:

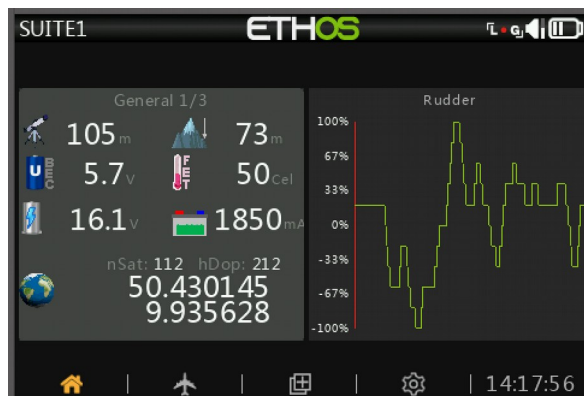
- **ein einzelner Frame**, in dem max 3 apps ausgewählt werden können

Hier wird das widget / der wrapper „SUITE1“ genutzt
Als Layout folgende Variante nutzen:

nur einer der beiden Frames darf durch den wrapper
genutzt werden !

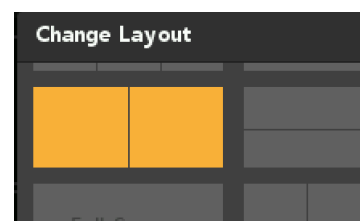


Das Ergebnis nach Konfiguration könnte dann so aussehen:



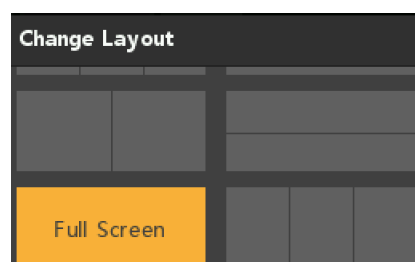
- **zwei Frames**, jeweils maximal 3 apps, Ethos original TopBar wird weiter genutzt

Hier wird das widget / der wrapper „SUITE2“ genutzt
Als Layout folgende Variante nutzen:



- **zwei Frames und Suite eigene TopBar**

Hier wird das widget / der wrapper „SUITE3“ genutzt
Als Layout die VollBild Variante nutzen:



(3) widget allgemein einrichten

(4) einzelne Apps konfigurieren

Telemetrie

App 01L	Tele01 ▼
DemoMode:	OFF <input type="radio"/> ON <input type="radio"/>
SensorSet:	Neuron & oXs ▼

Picture

App 01R	Picture ▼
Model File	alpina2 ▼
Background	SimmelH ▼
Timer display	OFF <input type="radio"/> ON <input type="radio"/>
Timer color	#C8CCC8 <input type="color"/> ▼

Kurveneditor

Widget ETHOS	
App 01L	Curve Edit ▼
Curve	bf ail ▼
Input	Throttle ▼
Trim Src	Slider Left ▼
Pt. Selector	Trim5 ▼
Restore Switch	SB↓ ▼

TopFull

Widget

ETHOS

24

TopLine	Top full
Status1, LSW:	LS Flaperon
Status1, label	flperon
Status2, LSW:	LS Snapflap
Status2, label	snpflap
VFR value	OFF ON
Mot Safety status "SAFE",LSW:	Safe Mot disarm
status "WARN/pre-engaged"	Mot Warning
Motor RUNNING	Motor running

Anhang

Modellspezifische Konfiguration

Standardsettings der Apps

Der „Telemetrikatalog“

Landesspezifische Texte / Sprachdateien

Beispieltemplate „Alpina“

Sonstige Optionen

Wer es sich zutraut kann in der Dateien weiteres Feintuning vornehmen.

Sprachdateien

Sensorverzeichnis

Modellspezifische Parameter

Internals

Für diejenigen, die etwas mehr Interesse an Lua Programmierung besitzen, ein paar Erklärungen:

Rev 0.8
unow, Aug 2023