

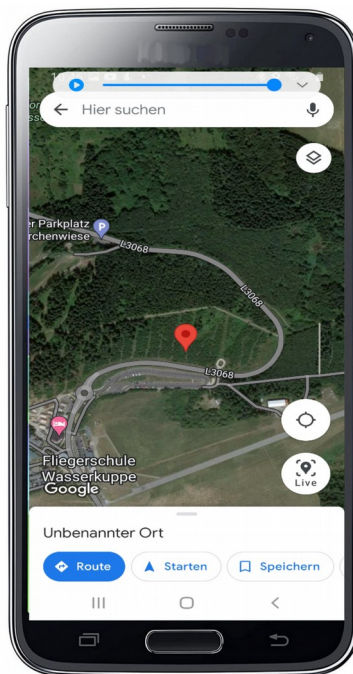
Anleitung zum widget „MODELFINDER“ Rev 1.0



„dark“



„bright“



Inhaltsverzeichnis

Einleitung.....	2
Hinweis „Source zur QR Code Kalkulation“:.....	3
Anwendung:.....	4
grobe Funktionsweise.....	4
Bedienung.....	5
Speichern von Koordinaten:.....	5
Starten / Zyklische Kalkulation des QR Codes.....	6
Laden von Koordinaten:.....	7
Installation:.....	8
(1) Dateien & Ordner kopieren.....	8
(2) source script einrichten.....	8
(3) widget einrichten.....	9
(4) nur Horus: Logische Schalter einrichten.....	10

Einleitung

Modelfinder erleichtert die Suche und das Auffinden eines Modells, was mit einem GPS ausgestattet ist:

Die letzte bekannte Position wird bei Bedarf im Widget als QR Code dargestellt.
Dieser QR Code wird mittels einem handelsüblichen Handys im Fotomodus „anvisiert“.
Smartphones erkennen den Inhalt des QR Codes heutzutage automatisch, auf dem Handy wird daher vorgeschlagen auf Google Maps zu wechseln.

Wird dieser Vorschlag akzeptiert startet das Handy Google Maps und zeigt dort die letzte bekannte Position.

Das Widget läuft (Stand Nov. 2022) auf Tandem und Horus Sendern.

Hinweis in eigener Sache:

Die Kalkulation des QR Codes ist sehr Speicher & Rechenaufwendig !
Die Berechnung benötigt auf einem Tandem Sender ca 6 Sekunden, auf einem Horus Sender ca 9 Sekunden.
Aufgrund der ETHOS / LUA Implementierung ist während dieser Zeit ist keine Bedienung an der Ethosoberfläche möglich, selbst ein Ausschalten kann nicht initiiert werden !

Um unnötige Last zu verhindern muss die QR Kalkulation manuell gestartet werden.
Alle 30 Sekunden wird dann ein neuer QR Code berechnet.

Für die Funktionalität des Widget ist es nicht zwingend notwendig die GPS Koordinaten zu loggen.

Es ist möglich die zuletzt empfangenen Koordinaten in ein File abzuspeichern, so dass sie auch später, ggf nach Neustart des Senders, noch abrufbereit sind.

Auf Tandem/Twin Sendern erfolgt die Bedienung mittels Touch Display, auf Horus Sender müssen Logische Schalter definiert werden, die das Starten bzw das Speichern ermöglichen.

Die Konfiguration erfolgt über das Ethos übliche Widget Konfig Menue

Wurde der Sender „deutsch“ lokalisiert (Systemeinstellung „Sprache“) werden automatisch deutschsprachige Texte „gezogen“, ansonsten englisch.

Über die Widget Konfiguration lässt sich eine „dunkle“ und ein „helle“ Darstellung aufrufen.

Zu Übungszwecken kann dort auch ein Testmodus aktiviert werden, der einen etwas variierenden GPS Empfang simuliert.

Die Darstellung besitzt jeweils eigene „Konfigs“ für X20 & Horus Sender, X18 sollte funktionieren, sobald ein Sim dazu zur Verfügung steht kann die Darstellung optimiert werden.

Horus Sender müssen Logische Schalter definiert bekommen, um die „Buttons“ zu simulieren.
(siehe „Installation“)

Die Zuordnung welcher LSW welche Buttonfunktion übernimmt kann in der Widget Konfig vorgenommen werden.

Hinweis „Source zur QR Code Kalkulation“:

Der Source zur Erstellung eines QR Codes ist sehr komplex und wurde via OpenSource zur Verfügung gestellt:

<https://github.com/speedata/luagrcode>

Ich bedanke mich ausdrücklich beim speedata Entwicklerteam, ohne dessen Arbeit dieses widget nicht möglich gewesen wäre !

Anwendung:

„Eigentlich“ ist die Bedienung einfach:
Nach Aufruf „Start“ Drücken und QR Code zeichnen lassen, mit Handy anvisieren,

- fertig -

Wer möchte kann noch „Save“ Drücken um die Koordinaten abzuspeichern bzw „Load“ um sie hochzuladen.
Um Rückfragen weitestgehend zu minimieren habe ich dennoch etwas mehr zur Bedienung geschrieben.
Wer „Old school“ mässig tatsächlich noch Anleitungen liest,

Bitteschön (-;

Im Folgenden wird die grundlegende Bedienung also etwas genauer beschrieben.
Es wird vorausgesetzt, dass das Widget funktionsbereit installiert wurde, näheres dazu im Kapitel „Installation“

grobe Funktionsweise

Das widget ist in zwei scripte unterteilt.

(1)

Ein „source script“ übernimmt die Hintergrundaktivitäten.
Es liest die aktuellen GPS Telemetriedaten, puffert diese falls kein Empfang mehr vorliegt, und speichert bzw liest Koordinaten über Dateioperationen.
Wird das „Speichern“ ausgelöst, erstellt das script eine kleine Datei mit zwei Zeilen, den Lat bzw Lon Koordinaten.

Die Datei erhält den Namen des Modellspeichers und wird unter
“/scripts/SRC_Gps_UN/data“ abgelegt.

Es werden immer 3 Generationen einer Datei vorgehalten, so dass auch „ältere“ Koordinaten im Notfall per PC ausgelesen werden können.

Das „source script“ stellt also die Koordinaten für das eigentliche widget bereit.

(2)

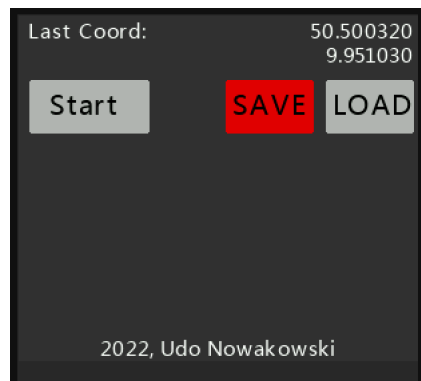
Das widget ist das „FrontEnd“, hier startet der Anwender die QR Kalkulation bzw. initiiert das Speichern oder Lesen von Koordinaten.
Wie bereits erwähnt ist die QR Kalkulation aufwändig weshalb diese bewusst über eine Schaltfunktion (Horus) oder Button (Touchdisplay) gestartet werden muss.

Auch wenn das Modell nach einem crash keine Stromversorgung mehr besitzt, wird die letzte empfangene Position noch visualisiert und kann abgespeichert werden.

Bedienung

Nach Installation des Widgets erscheint die Oberfläche, ohne dass eine QR Kalkulation stattfindet. Man erkennt es daran, dass der Start Button grau ist. Stellen wir uns vor, das Modell ist am Hang „abgesoffen“ oder hat weshalb auch immer in einem Maisfeld „hart gestoppt“.

Wir schauen uns das Widget an:



In diesem Beispiel zeigt der Sender noch eine gültige Koordinate an. Sollten sich diese Koordinaten nicht mehr verändern ist es sehr wahrscheinlich, dass kein GPS Empfang mehr vorliegt. Sollte das Modell nach einem Absturz keine Stromversorgung mehr besitzen ist auch die akustische Meldung „Telemetrie verloren“ erfolgt.

Speichern von Koordinaten:

Es ist auf jeden Fall ratsam im ersten Schritt das Speichern auszulösen (per Touch oder Schalter)

Der Save Button wird kurz blau, was bedeutet, dass die letzte gespeicherte Position durch die Telemetrie nicht mehr aktualisiert wurde. Im Crashfall wird dieser Button also blau bleiben !



Hinweis: eine „alte“ Datei wird nicht einfach überschrieben, sondern umbenannt. Es stehen maximal 3 Datei Versionen (die aktuelle und zwei Vorläufer) zur Verfügung.

Starten / Zyklische Kalkulation des QR Codes

Dann wird „Start“ ausgelöst, und sobald der Zeitpunkt für eine Neukalkulation erreicht ist, wird der QR Code prozessiert und auf dem Display dargestellt.

Der Code wird nun zyklisch neu berechnet. Sollte vorher ein Speichern durchgeführt worden sein, und die Koordinaten haben sich geändert, wird der Save Button wieder auf die Farbe rot wechseln, siehe Bsp:



Sollte eine Kalkulation nicht mehr notwendig sein:

UNBEDINGT wieder Stoppen um die CPU Last zu verringern !

Der oben erwähnte Farbwechsel des Save Buttons kann für folgendes Szenario angewendet werden:

Das Modell macht eine Aussenlandung, Stromversorgung bleibt intakt.

Durch RF Abschottung allerdings kein GPS Empfang am Sender, man speichert also die letzten Koordinaten. Der Button wird daher blau, keine Änderung der Anzeige.

Nun begibt man sich via Maps in die Umgebung der letzten „Standorts“ vom Modell.

Irgendwann wird der Empfang wieder eintreten, durch den Farbwechsel des Save Buttons auf rot erkennt man, dass das Modell weiter unter Spannung ist und die nun erhaltenen Koordinaten präziser sein werden.

Ein gewisses Rauschen, also kleine Schwankungen in den Koordinaten, sind selbstverständlich und sollten bekannt sein.

Findet man das Modell trotzdem nicht, empfiehlt es sich bei leicht wechselnden Koordinaten drei unterschiedliche Datensätze / Dateien abzuspeichern (siehe oben, es rotieren drei „Generationen“ der Dateien). Durch Triangulation im Nachgang (per PC) dürfte eine noch präzisere Angabe möglich sein.

Kleiner Hinweis:

Sollte man einen „Restart“ der QR Kalkulation vornehmen, und die letzte Kalkulation ist deutlich älter als die „Refreshzeit“ erfolgt i.d.Regel eine unmittelbare Berechnung des QR Codes, ohne dass dies durch einen Anzeigetext bekannt gemacht wird.

(.. Ein Fall für die Rev2.0)

Laden von Koordinaten:

Sollte der Sender zwischendurch ausgeschaltet worden sein, und man erhält nach Einschalten kein GPS Signal, kann die letzte Datei „hochgeladen“ werden.

Unmittelbar nach Upload wird der Code kalkuliert und dargestellt.



Anhand des grünen „Lade“ Buttons sowie des grünen Modustextes oben links erkennt man, dass aktuell die Koordinaten einer Datei dargestellt werden.

Auch hier gilt es den Farbwechsel des „Speichern“ Buttons als zusätzliche Informationsquelle zu beachten:

>> Ist der Button blau, haben sich die Koordinaten seit dem Laden nicht verändert.

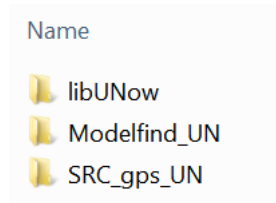
>> Ist der Button rot, wurden neue Koordinaten empfangen (siehe Beispiel oben)

Installation:

(1) Dateien & Ordner kopieren

Nach download des Zip Files gilt es dieses zu entpacken

Nun stehen drei Ordner zur Verfügung, die in das script Verzeichnis des Senders kopiert werden müssen:



Das „Modelfind_UN“ Verzeichnis enthält das Haupt-Widget

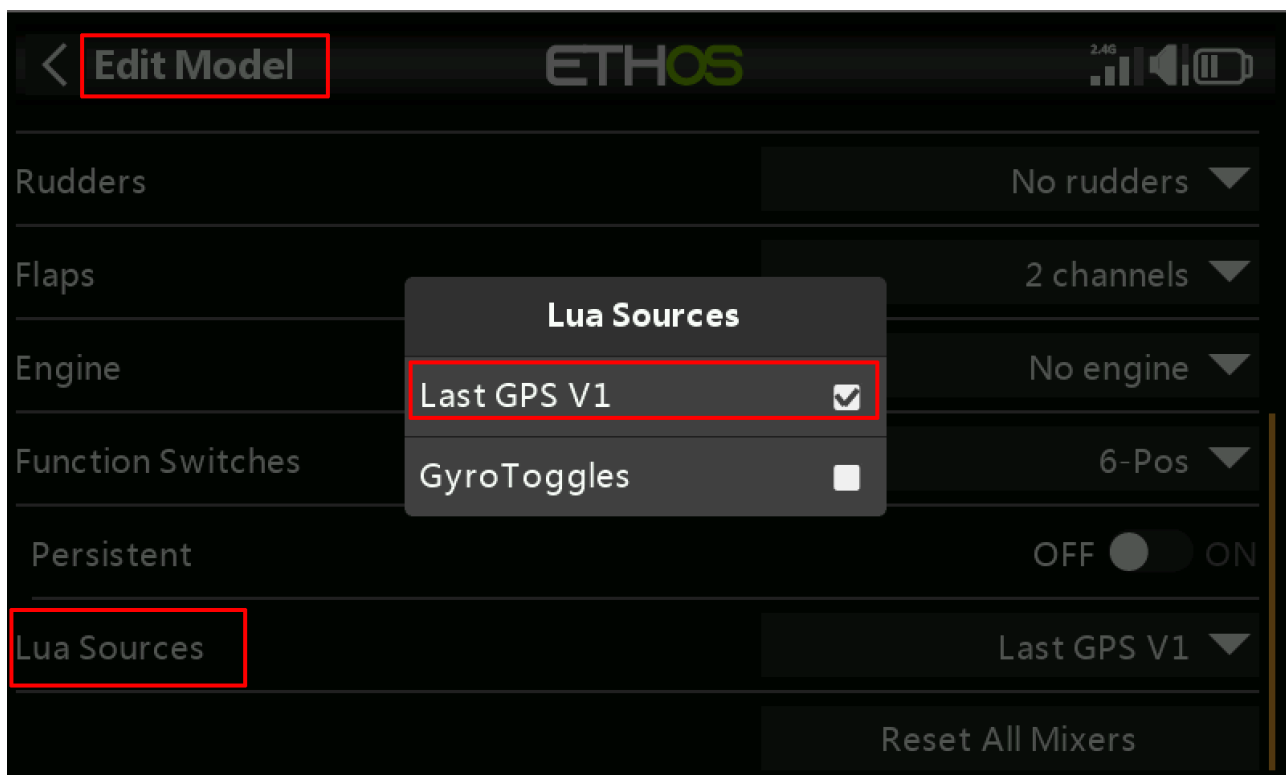
Der „SRC_gps_UN“ Ordner beinhaltet das source script

Das „libUNow“ Verzeichnis enthält allgemeine lua Funktionsdefinitionen, die von mehreren widgets genutzt werden können, u.a. auch das modelfinder widget greift darauf zu.

(2) source script einrichten

Source scripts werden unter Ethos immer in der **Modelleinstellung** aktiviert (s.a. „Edit Model“)

Unter dem entsprechenden Formulareintrag ganz unten („Lua Sources“) ist das „Last GPS“ script zu aktivieren:



(3) widget einrichten

Das widget wird in einem Frame mit halber Seite Grösse eingerichtet.

Die Kenntnis darüber wie Ethos Seiten eingerichtet werden / Layouts zu wählen sind setze ich voraus.

- In der Widget Konfig wird zunächst das „Modelfinder“ widget ausgewählt.
- der Titel ist unbedingt abzuschalten
- die Darstellung (Thema / Farbschema) kann zwischen hell & dunkel gewählt werden
- wenn gewünscht kann der Testmodus scharf geschaltet werden (kein GPS / Empfang notwendig)
- Für Horus Sender **MÜSSEN** die logischen Schalter selektiert werden (näheres dazu weiter im Text..), Touch Sender können diese Parameter unangetastet lassen. Aufgrund der automatischen Sendertyperkennung im widget haben sie keinerlei Funktion.

Setting	Value
Widget	ModelFinder 0.9
Title	OFF
Theme	Dark
Test mode	ON
Horus: LSW Start	LSW_1 A
Horus: LSW Save	LSW_1 A
Horus: LSW Load	LSW_1 A

(4) nur Horus: Logische Schalter einrichten

Da Horus Sender keine Touch Oberfläche bieten, müssen die entsprechenden Funktionen per logische Schalter abgebildet werden.

Das widget „reagiert“ immer auf einen Statuswechsel eines LSW's um die Touch Bedienung zu simulieren

Also immer dann, wenn ein Log.Schalter von „Aus“ nach „Ein“ bzw umgekehrt wechselt, verhält sich das widget so, als ob eine Taste am Touch screen gedrückt wurde.

Dementsprechend sind drei LSW's unter Ethos anzulegen.

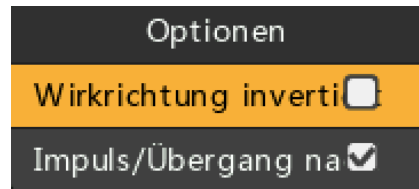
Sind die LSW's definiert, werden sie in der widget Konfig wie oben beschrieben zugeordnet.

Die interne Zuordnung im Widget erfolgt mittels des NAMENS des LSW's, dies sichert die persistente Zuordnung, auch wenn andere LSWS „mittendrin“ in der Modellkonfig verschoben, gelöscht oder neu angelegt werden und sich somit die Nummerierung ändert !

Wie aber definiere ich einen solchen LSW?

Es bieten sich dazu Trimtaster als Quelle an:

- Der LSW erhält die Funktionalität „FlipFlop / Sticky“
- Jedesmal, wenn der Taster bedient wird, soll der Schalterstatus gewechselt werden werden.
Dazu nutzt man unter Ethos die Funktion auf „Impuls/Flanke“ reagieren
Um das zu erreichen wird zunächst der Trimtaster unter „Trigger Ein“ ausgewählt,
danach wird durch „long press“ auf diesen Eintrag die Impulsvariante angewählt



Die „Trigger Aus“ Bedingung wird genau so definiert, nur dass in den Optionen zusätzlich „Wirkrichtung invertiert“ mit angewählt wird

Auf diese Weise erhält man einen Logischen Schalter, der bei beim ersten „T5 nach links“ Klick eingeschaltet, beim nächsten „T5 nach links“ Klick ausgeschaltet wird.

Zur Realisierung hier ein screenshot, wie z.B. T5 als Schalter für „Save“ definiert wurde:



analog dazu werden die anderen zwei Schalter definiert