

Source script „Integral Vario“
2024 by Udo Nowakowski
(draft // beta 0.81)

verwendbar ab Ethos 1.5

Das script erlaubt es einen „Custom“ Telemetriesensor anzulegen, der einen gemittelten Variowert ausgibt.

Somit wird es möglich, einen Varioton zu generieren, der einem über einem längeren Zeitraum (z.B. 3..4 Thermik-Kreise) Auskunft darüber gibt, ob man tatsächlich unter sehr schwachen Bedingungen ein „langfristiges“ Steigen erzielt.

Letztlich wird über eine Zeitkonstante die jeweilige Höhendifferenz ermittelt, und daraus das durchschnittliche Steigen kalkuliert

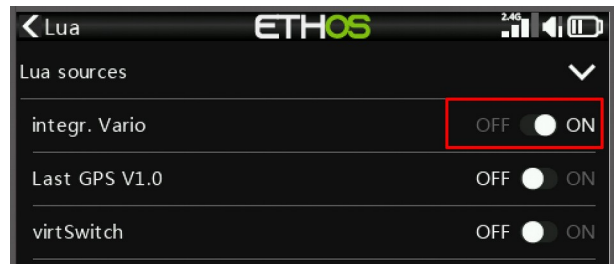
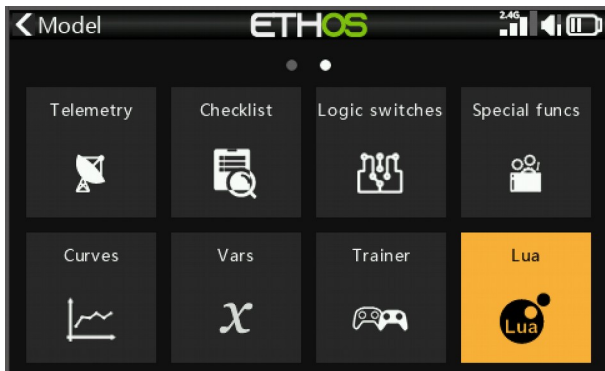
Mit Ethos 1.5 wird eine Spezialfunktion „playVario“ eingeführt, die es ermöglicht beliebige Sensoren der Einheit „m/s“ als Varioton umzusetzen.

Dies ist eine Grundvoraussetzung und daher ist das script erst ab 1.5 sinnvoll einsetzbar.

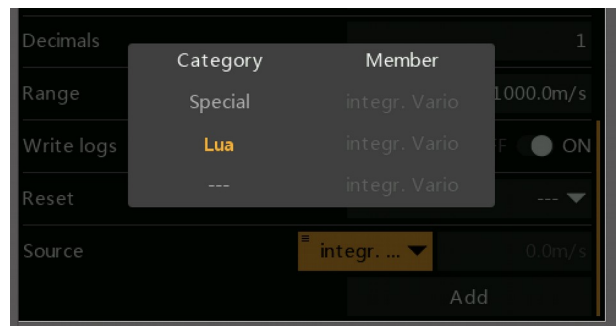
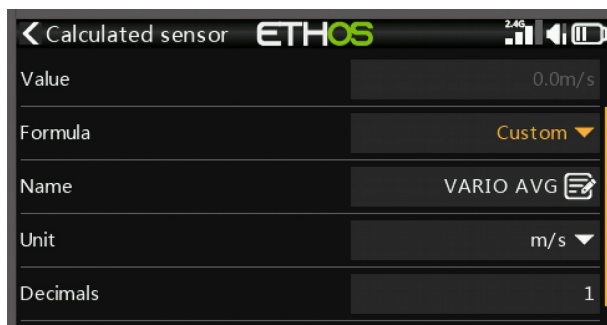
Ausserdem besteht die Möglichkeit einen beliebigen Sensor in die Einheit m/s in Echtzeit zu konvertieren

Kurzanleitung:

- (1) Das script unter dem Ordner „/scripts“ in ein eigenes Verzeichnis kopieren
- (2) Sender neu starten, und im Modellspeicher > Modell >> lua das source script aktivieren

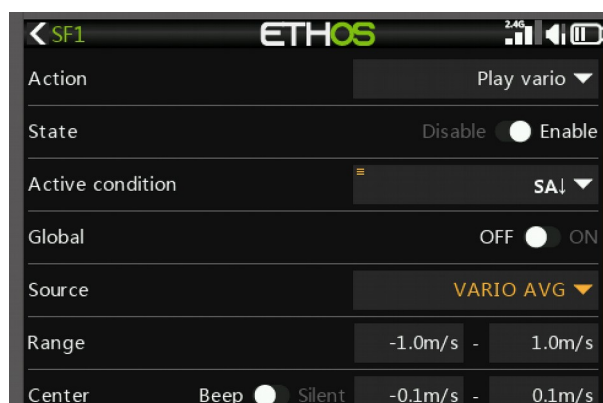


- (3) unter Telemetrie einen benutzerspezifischen Sensor anlegen
Als Quelle dabei unter der Lua Kategorie das skript angeben
Bei nur einem „scharfen“ Source script erhält man autom. diese Auswahl



- (4) Nun die SF „play Vario“ mit Quelle des o.g. Sensors anlegen

So konfigurieren, dass man z. B. über einen 3 Stufen Schalter auswählen kann zwischen „ Vario Aus <> Vario Normal <> Vario integral “
oder für das IntegralVario einen Taster für „adhoc“ Abfragen verwenden



Benutzerkonfiguration

a) typisch (Ausgabe durchschn. Steigen)

Je nach Anwender möchte der eine eine längere Zeitkonstante, der andere eine kürzere. Manche möchten ggf sogar die Samplingrate (1 Datensample/Sekunde) persönlich anpassen. Ggf muss je nach Hardware/Konfig auch die Quelle des eigentlichen Varios adaptiert werden.

Dazu existieren 3 Codezeilen im script:

```
1  -- *****
2  -- user personalized config / customizing
3  -- *****
4  local sensor = "Altitude" -- please change in case altitude sensor has another label
5  local RECORDduration <const> = 20 -- duration in seconds for building average
6  local RESOLUTION <const> = 1 -- save/datapoint-interval in seconds
7
8
```

- „sensor“ muss ggf auf das label/ die Benennung des Varios angepasst werden
- RECORDduration ist die Zeitkonstante / die Aufzeichnungsdauer in Sekunden
- RESOLUTION gibt an, wie häufig (Intervall / alle x Sekunden) ein Wert gespeichert wird.

Dies bitte auf die individuellen Bedürfnisse anpassen

b) absoluten Differenzwert ausgeben:

eine zweite Alternative kann es sein, nicht das durchschnittliche Steigen, sondern die reine Höhendifferenz als Sensorwert auszugeben
(dann auch als Einheit m/s)

Das ermöglicht z.B. per Taster eine Sprachansage, wieviele Meter man im eingestellten Zeitraum gewonnen / verloren hat.

Dazu wird der Parameter „DIFFonly“ auf von **false** auf **true** gesetzt

```
38
39
40 local DIFFonly = true
41
42
```

den Wert kann man immer noch als Ton modulieren, dazu entsprechend den „Bereich“ in der Spezialfunktion anpassen

„Sensor“ - Echtzeit Konverter

Das script lässt sich auch dafür nutzen, einen Sensor, der keine „m/s“ Einheit anbietet, direkt und ohne Mittelung in die Einheit „m/s“ zu übersetzen.

Dabei kann falls notwendig auch direkt skaliert werden

dazu unter sensor den entsprechenden Telemetriesensor eintragen,
und die Konstante „RESOLUTION“ auf den Wert **-1** setzen

```
35  
36 local RECORDduration <const>    = 20  
37 local RESOLUTION <const>        = -1
```

Jetzt kann unter der Telemetrie manuell , wie oben bereits gezeigt, ein benutzerspezifischer Sensor mit der Einheit m/s ergänzt werden.

Sollte eine Skalierung notwendig sein, kann z.B. der Wert um einen Faktor bereits im Coding verändert werden:

```
141 else  
142     local sensVal      = input:value()  
143     source:value(sensVal * 1.0 )  
144 end
```

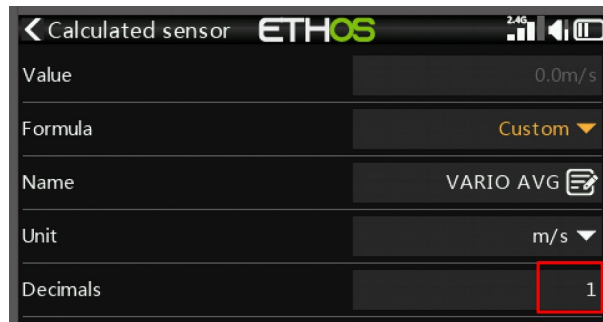
(hier der Faktor 1.0)

Tipps:

A) akustische Auflösung

bei schwacher Thermik wird das Integral Vario kleine Werte liefern.
Die SF lässt sich im Wertebereich minimal von -1m/s bis +1m/s konfigurieren.
Dadurch wird eine eher grobe „akustische Auflösung“ produziert.

Möchte man „feiner“ auslösen kann folgender Trick angewendet werden:
durch setzen der „Kommastellen“ von 1 auf 0 wird der Sensorwert quasi verzehnfacht
hier die „Stelle“:



Aus 0,25m/s werden also 2,5m/s
Danach den Range auf die individuell bevorzugten Werte spreizen

B) Simulation

Im Flug wird es schwierig sein den Varioton auf das individuelle Wunschverhalten einzustellen
Ich habe daher einen „Simulationsmodus“ eingebaut.
Über den Gasknüppel wird eine „aktuelle“ Höhe zwischen 0 und 20m simuliert (Mitte entspricht also rund 35m)
Durch langsame Bewegung simuliert man Höhenveränderung des Modells
Setzt man die Konstante „SIM“ von false auf true wie unten gezeigt, arbeitet man im Simulationsmodus und kann den varioton auch ohne Modell einstellen.

```
19  
20 local SIM <const> = true  
21
```

Als Hilfe / Bsp habe ich ein 1.5er Modell „IntVario“ mit in die Files gepackt
Das „Integralvario“ lässt sich über SD bedienen / aktivieren.

viel Spaß & Gruß

Udo Nowakowski