

# FrSKY Taranis GPS-Racing Telemetrie-Skript Bedienungsanleitung



David Beach  
[david.s.beach@gmail.com](mailto:david.s.beach@gmail.com)  
8. Juni 2016

## Inhaltsverzeichnis

Historie.....	3
Einführung .....	3
Installation .....	4
Dateien auf den Sender kopieren.....	4
Installationsüberprüfung .....	4
Verwenden des simulierten GPS.....	5
Beschreibungen der Kursparameter .....	6
Lass uns Rennen fliegen! .....	6
Triangle Course.....	8
2 Line Course.....	9
Line / Cylinder Course.....	9
Custom Course.....	10
Festlegen von benutzerdefinierten Kurswegpunkten über telem2.lua .....	11
Festlegen der Standardeinstellungen über telem2.lua .....	11
Zukünftige Verbesserungen .....	12
Verbesserungen der Telemetrie .....	12

## Historie

### 8. Juni 2015 - Release 1.0

Skript jetzt kompatibel mit OpenTx 2.0.13 und OpenTx 2.1.7 (und hoffentlich späteren Versionen)  
Berechnung der Aufgabengeschwindigkeit hinzugefügt (überprüft anhand der Dreiecksaufgaben von RC Electronics T3000)  
XCRestrt-Mix-Skript hinzugefügt (Unterstützung für das Beenden / Neustarten über einen vom Benutzer wählbaren Schalter)  
Aktualisierungen der Dokumentation  
Kleinere Fehlerbehebungen und Änderungen an der Benutzeroberfläche

### 5. Januar 2014 - Beta 0.2

Unterstützung für Linien- / Zylinder- und benutzerdefinierte Kurse hinzugefügt. Zusätzliche Dokumentation veröffentlicht.  
Fehler im Zusammenhang mit Null-Telemetriewerten behoben.

### 20. Dezember 2014 - Beta 0.1

Erstveröffentlichung nur mit Triangle- und 2 Line-Kursen. Die Verbreitung war begrenzt und minimal Dokumentation wurde zur Verfügung gestellt.

## Einführung

Willkommen und vielen Dank, dass Sie das GPS Racing Telemetry Script für die FrSKY Taranis verwenden. Die Skripte werden unterstützt im Rahmen des RCGroups-Thread unter

<http://www.rcgroups.com/forums/showthread.php?t=2219128> , oder per E-Mail an [mailto: david.s.beach@gmail.com](mailto:david.s.beach@gmail.com) .

Wenn Sie ein Problem finden und es reproduzieren können, gibt es eine gute Chance, dass ich es reparieren kann. Wenn es scheinbar zufällige Fehler gibt, melden Sie diese bitte und ich werde versuchen eine Lösung zu finden.

Das System wurde für die Sender FrSKY Taranis und Taranis Plus mit den OpenTX Softwareversionen 2.0.13 oder 2.1.7 (oder höher) entwickelt. Ihre Sender-Firmware muss die LUA-Option enthalten, es gibt jedoch keine weiteren spezifischen Firmware-Anforderungen. Die Telemetrieskripte verwenden keine globalen Flugmodi, Variablen, logische Schalter, Sonderfunktionen oder benutzerdefinierte einmalige Skripte. Die Installation der Telemetrie Skripte sollten keine Nebenwirkungen auf vorhandene Modell-Setups haben.

Diese Version enthält jetzt ein optionales Mix-Skript (XCRestrt.lua), das das Beenden / Neustarten durch den Benutzer über einen wählbarer Schalter ermöglicht.

Die Skripte erfordern ein integriertes FrSKY-kompatibles GPS, das Längen- und Breitengradinformationen liefert (verfügbar im Standard-Telemetriedatenstrom). Sie wurden getestet mit dem FrSKY GPS V2 Sensor und einem generischen GPS, das an den FrSKY-Sensor-Hub angeschlossen wurde. Anwender haben auch von Erfolgen mit dem Altis GPS und selbstgebaute GPSs basierend auf OpenXSensor berichtet. Besitzer von FrSKY GPS V2-Sensoren sollten in Betracht ziehen die Firmware zu aktualisieren (siehe <http://www.rcgroups.com/forums/showthread.php?t=2600818> ) oder einen kompatiblen Sensors mit einer höheren Aktualisierungsrate zu kaufen.

Wenn die entsprechende Telemetrie-Seite ausgewählt wurde, erfolgt sowohl eine akustische Unterstützung der Kursnavigation als auch eine visuelle (über das Display).

Dieses System enthält eine GPS-Simulationsoption, mit der Sie sich mit den verwendeten Skripten vertraut machen können. Dazu benötigen sie nur ihren Sender oder die OpenTX Companion-Software (verfügbar unter <http://www.opentx.org/downloads.html>)

## Installation

### Kopieren Sie die Dateien auf den Sender

Alles wird in einem einzigen Zip-Archiv bereitgestellt, das entpackt werden muss. Die einzelnen Dateien werden dann in das entsprechende Verzeichnis auf der SD-Karte des Senders gelegt.

Alle xc \*.wav-Dateien müssen im Verzeichnis / SOUNDS / de / abgelegt werden.

OpenTx 2.0.x-Benutzer sollten für alle Modelle, die für GPS-Rennen verwendet werden, die Dateien telem1.lua und telem2.lua im Verzeichnis / SCRIPTS / < Modellname > / ablegen.

OpenTx 2.1.x-Benutzer sollten die Skripte telem1 und telem2 in das Verzeichnis / SCRIPTS / TELEMETRY kopieren. Verweisen Sie dann auf die Skripte auf der Telemetrie-Konfigurationsseite bei allen Modellen, die für GPS-Rennen verwendet werden.

### HINWEIS:

Die Datei telem1.lua wird im LUA-Bytecode-Format bereitgestellt und sollte nicht mit einem typischen Texteditor geändert werden. Ich habe zwar vor, den Quellcode zur Verfügung zu stellen, aber er ist zu groß, um ihn erfolgreich auf einem Sender zu kompilieren und zu starten.

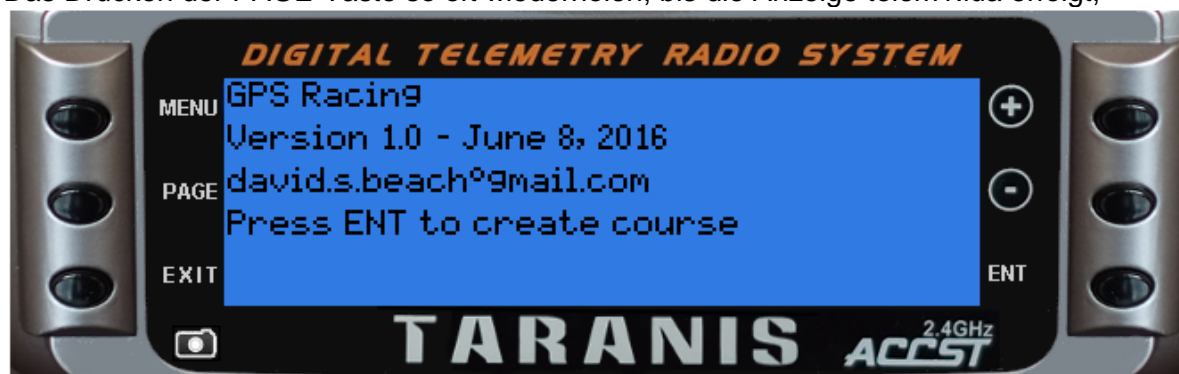
Die Datei telem2.lua ist eine normale Textdatei und kann von Anwender geändert werden, um Kursstandards festzulegen und andere Parameter zu überschreiben. Sie ist auch optional und kann bei Bedarf in einem Modell weggelassen werden, um mehr Speicherplatz für andere Telemetrie Skripte zu schaffen.

Unter OpenTX 2.0 kann die Datei telem1.lua umbenannt werden, muss jedoch bei Verwendung vor telem2.lua geladen werden. Wenn beispielsweise telem1.lua in telem4.lua umbenannt wird, muss telem2.lua in telem5.lua (oder höher) umbenannt werden.

## Installationsüberprüfung

Bei ordnungsgemäßer Installation können Sie die Telemetrieanzeige durch langes Drücken der PAGE-Taste auswählen.

Das Drücken der PAGE-Taste so oft wiederholen, bis die Anzeige telem1.lua erfolgt,

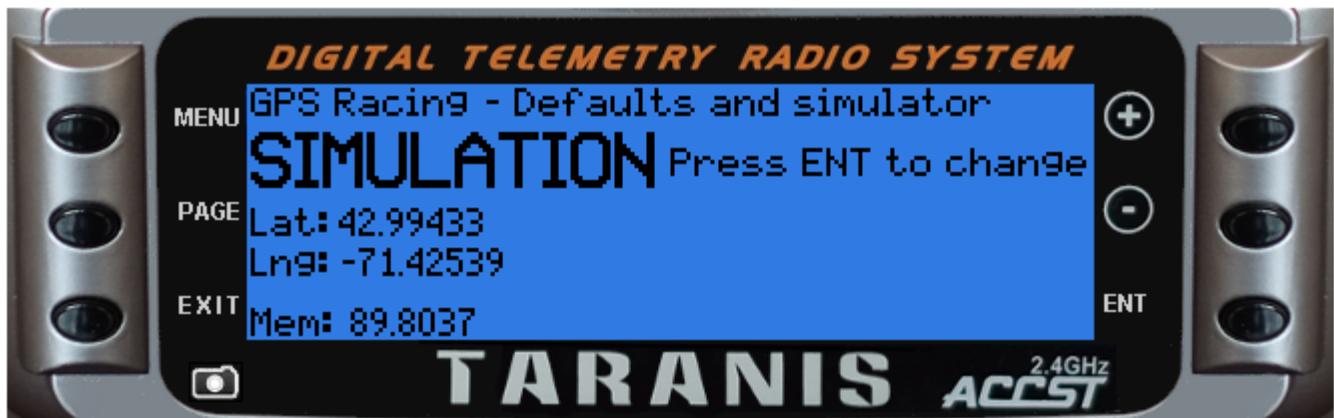


gefolgt von der Anzeige telem2.lua



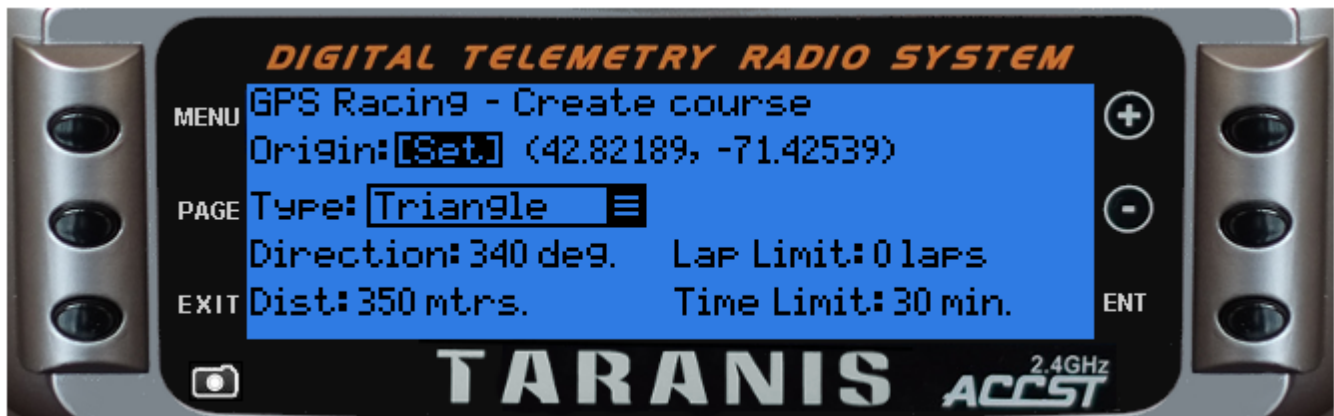
### Verwendung des simulierten GPS

Aktivieren Sie das simulierte GPS, indem Sie zum Telem2-Display gehen und die Taste ENT drücken. Es sollte jetzt ungefähr so aussehen:



Im Simulationsmodus steuert der Gashebel die Geschwindigkeit und der Ruder- und Querruderhebel die Drehrate. (Ich habe nur in Modus zwei getestet. Wenn Sie Modus eins fliegen, lassen Sie mich wissen, ob sich die Knüppel so verhalten, wie Sie es erwarten). Wenn die Simulation aktiv ist, werden Sie Änderungen in den angezeigten Lat: und Lng: feststellen, sofern die der Gashebel nicht auf Null steht.

Drücken Sie nun wiederholt PAGE, bis Sie wieder auf dem telem1-Display sind. Drücken Sie in diesem Bildschirm auf ENT. Danach können Sie den GPS-basierten Kurs eingeben.



Wenn (wie oben gezeigt) [Set] ausgewählt ist und ENT gedrückt wird, blinkt [Set] und die angezeigten GPS Koordinaten werden automatisch aktualisiert. Jetzt können Sie den Gashebel anheben und durch das Bewegen des Querruderhebels kann man überprüfen, ob die Positionsaktualisierung erfolgt. (Beachten Sie, dass bei aktivierter TELEMETRIE-Option die Bewegung des Flugzeugs (sobald es einen GPS-Fix hat) auch bewirkt, dass die Koordinaten aktualisiert werden.)

Drücken Sie erneut ENT, um den Ursprungspunkt für den Kurs zu speichern.

Mit den Tasten + und - (bzw. Drehen der Walze) navigieren Sie zu den anderen Optionen.

Drücken Sie ENT in einem beliebigen Feld um es zu bearbeiten und drücken Sie erneut ENT, um den Bearbeitungsmodus des Feldes wieder zu verlassen.

## Beschreibungen der Kursparameter

**Origin:** Hiermit werden der Startlängengrad und Startbreitengrad für alle Kurstypen festgelegt (außer für benutzerdefinierte Kurse).

**Type:** Triangle, 2 Lines, Line / Cyl, and Custom (benutzerdefiniert) sind als Kurstypen verfügbar und werden weiter unten im Detail erläutert.

**Direction:** Dies ist die Startrichtung (0 bis 360 Grad) vom Ursprung (Origin) bis zum ersten Wegpunkt. (0° bedeutet der erste Wegpunkt liegt im oben/Norden, 90° bedeutet er liegt rechts/Osten, usw...)

**Dist:** Dies ist die „Beinlänge“ in Metern für die Wegpunkte relativ zum Ursprung.

**Lap Limit:** Geben Sie einen Wert größer als Null ein, wenn ein Rundenlimit gewünscht wird.

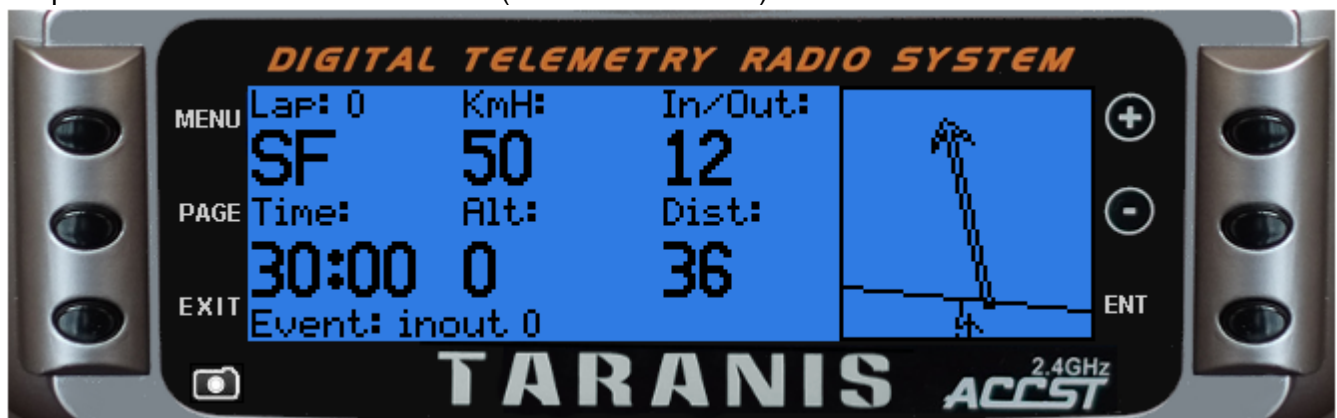
**Time Limit:** Geben Sie einen Wert größer als Null ein, wenn ein Zeitlimit gewünscht wird.

Zu beachten: Wenn sowohl "Rundenlimit" als auch "Zeitlimit" angegeben sind, wird der Kurs beendet, wenn eines der beiden Limits erreicht wird.

## Lasst uns Rennen fliegen!

Wenn Sie mit den Kursparametern zufrieden sind, drücken Sie EXIT, um die Kursnavigation zu starten. (wichtig: Aufgrund eines Fehlers in älteren oTx Versionen kann es auch sein, dass man bei manchen Sendertypen MENU nehmen muss. Dazu muss man in telem1.lua EVT\_EXIT\_BREAK durch EVT\_MENU\_BREAK ersetzen.)

Beachten Sie, dass alle hier vorgenommenen Änderungen verloren gehen, wenn der Sender ausgeschaltet wird. Allerdings können alle Kursparameter und Standardwerte durch Bearbeiten des Skripts telem2.lua vordefiniert werden (siehe Details unten).



Oben ist der Hauptbildschirm von telem1 dargestellt, wenn ein Kurs aktiv ist.

Bevor ich Sie mit dem grafischen Teil der Anzeige vertraut mache, werde ich alle Textfelder beschreiben.

**Lap:** Die aktuelle Rundennummer wird angezeigt. Darunter befindet sich der Name des nächsten Wegpunkts. Hier "SF" steht für Start / Ziel. (Wp1 steht für Wegpunkt-1 usw.)

**KmH:** berechnete Fahrgeschwindigkeit in Metern pro Sekunde. (Hinweis: Auch wenn die Geschwindigkeit per Telemetrie verfügbar ist wird der Telemetriewert derzeit nicht verwendet.) (Anzeige muss bei kleinen Displays entfallen.)

**In/Out:** Dies ist die Abweichung vom idealen Kurs zwischen Wegpunkten, gemessen in Metern. Positiv Zahlen befinden sich außerhalb (rechts) der Ideallinie, negative Zahlen befinden sich innerhalb (links).

**Time:** - Zeigt die verstrichene (Count up) oder verbleibende (Countdown) Zeit an, je nachdem, ob eine Zeit Limit vorgegeben wurde oder nicht.

**Alt:** Höhe, wie von einem Variometer angegeben, falls vorhanden. (Die Maßeinheit wird durch die Telemetrieinstellungen bestimmt.) (Anzeige muss bei kleinen Displays entfallen.)

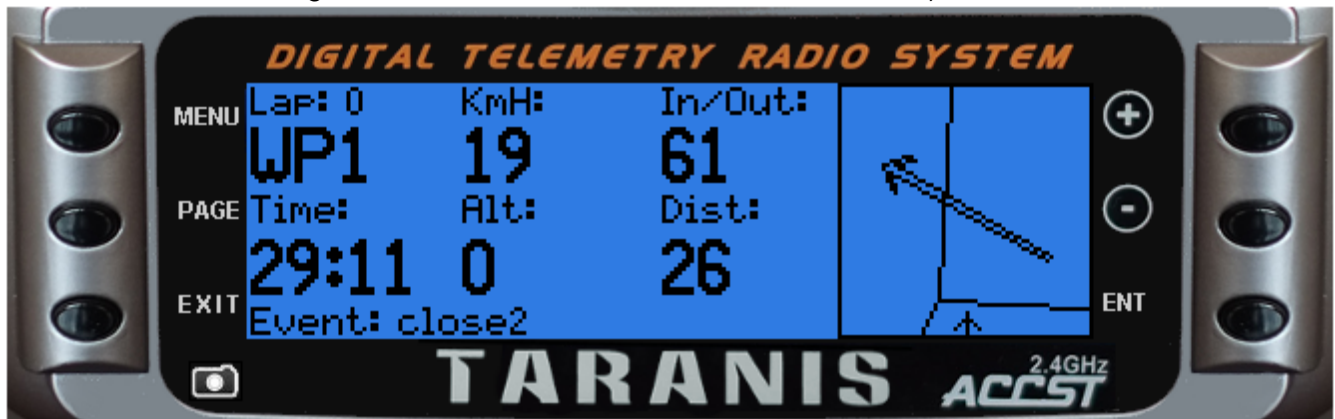
**Dist:** Entfernung in Metern zum nächsten Wegpunkt.

Bei Dreieckskursen gilt: Ist man innerhalb, wird der Abstand zum Ursprung angezeigt.

Ist man außerhalb, wird der Abstand zur Linie des nächsten Wegpunkts angezeigt.

### Seite 7

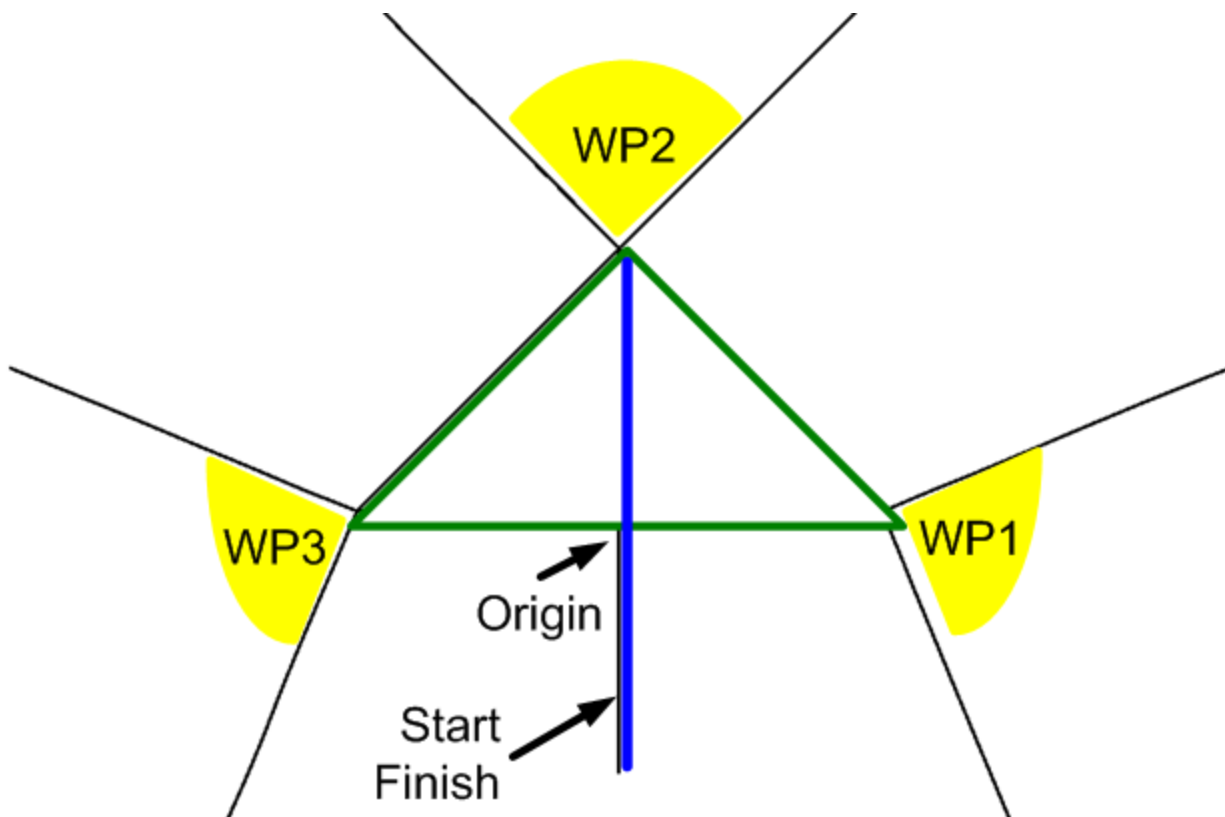
**Grafik** - Die Spitze des kleinen Pfeils am unteren Rand der Grafik zeigt die relative Position des Flugzeugs zum Kurs an. Gerade Linien sind skalierte Darstellungen des nächsten Wegpunkts. Der große umrandete Pfeil zeigt die relative Richtung zum Wegpunkt aus der Perspektive des Flugzeugs. Im Allgemeinen gilt, wenn der Umrisspfeil gerade nach oben zeigt fliegt man in Richtung des Wegpunkts. Wenn man sich einem Wegpunkt nähert dreht sich der Pfeil wie unten gezeigt in Richtung des Wegpunkts. (Zeigt der Pfeil dabei nach links umfliegt man den Wegpunkt auf der Außenseite. Zeigt der Pfeil nach rechts umfliegt man ihn auf der Innenseite – was falsch wäre)





## Triangle Course

Das folgende Diagramm zeigt den Dreieckskurs:



Bei einer gegebenen Ursprungscoordinate (Origin) wird WP1 in der angegebenen Richtung lokalisiert. Angenommen in diesem Bild ist Norden oben, dann ist dieses Beispiel ein 90-Grad-Kurs.

Alle drei Wegpunkte sind gleich weit entfernt vom Ursprung.

Sobald ein Kurs (durch Überfliegen der Start/Finish Linie in Richtung WP1) aktiviert wird, läuft die Zeit.

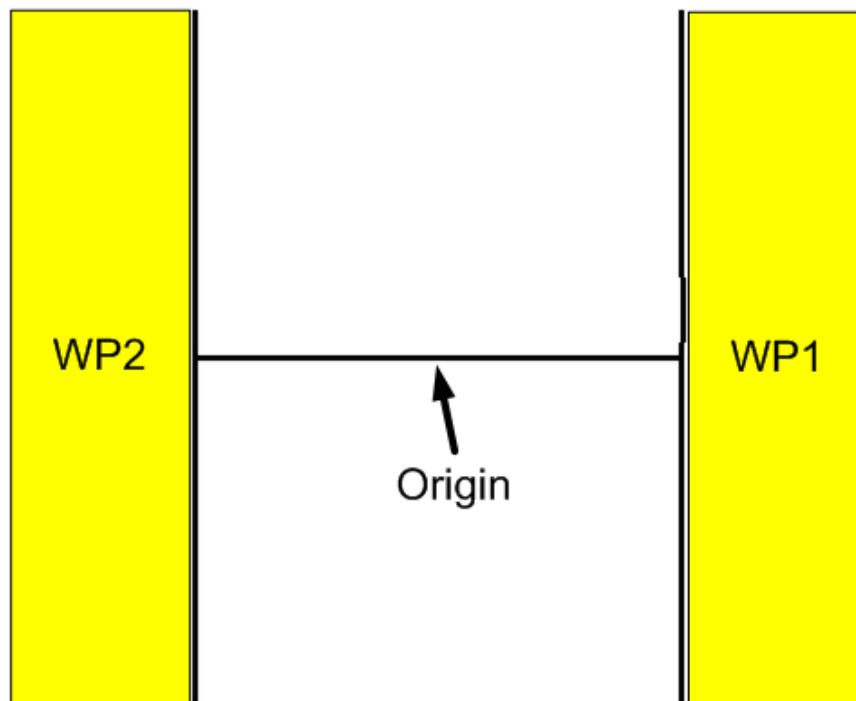
Der Wegpunkt gilt jedes Mal als erreicht, wenn das Flugzeug in den gelben Bereich eintritt (einschließlich des erweiterten nicht hervorgehobene Bereich).

Eine Runde ist beendet, wenn alle Wegpunkte in der richtigen Reihenfolge erreicht wurden und die Start- / Finishlinie wieder überflogen wurde.



## 2 Line Course

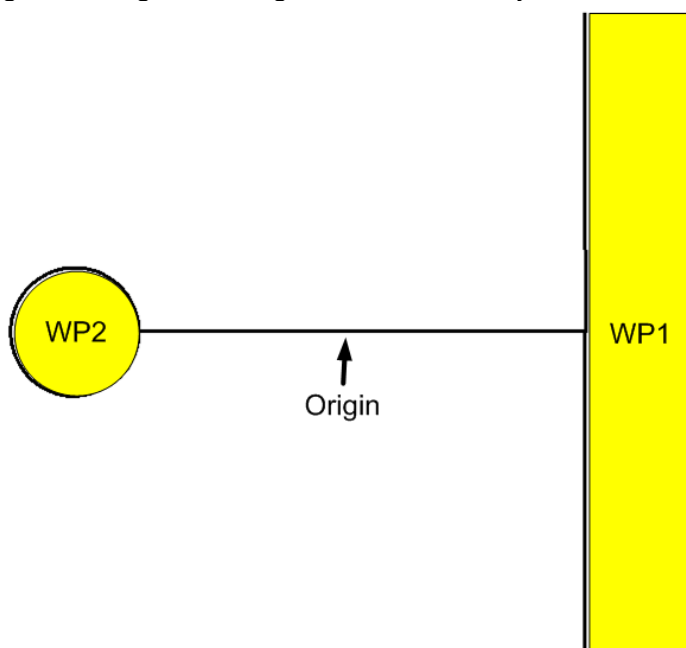
Das folgende Diagramm zeigt den 2-Linien-Kurs.



Bei einer gegebenen Ursprungscoordinate wird WP1 durch die Verwendung der angegebenen Richtung und Entfernung definiert. Die definierende Linie WP2 befindet sich in der entgegengesetzten Richtung. Der Kurs beginnt, wenn das Flugzeug den gelben Bereich WP1 in Richtung zum Ursprung verlässt. WP2 wird erreicht, wenn das Flugzeug in einen Teil des erweiterten WP2 Bereichs eintritt. Eine Runde ist beendet, wenn das Flugzeug zu WP1 zurückkehrt.

## Line / Cylinder Course

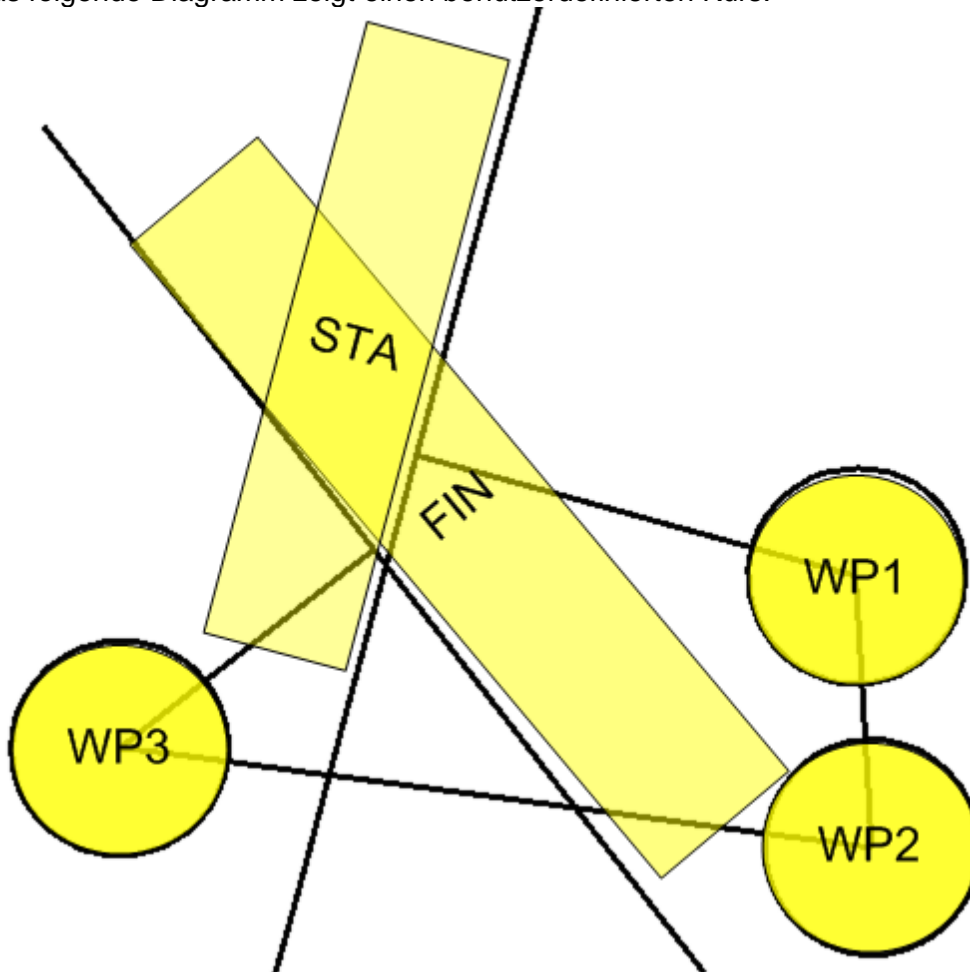
Das folgende Diagramm zeigt einen Linien- / Zylinderverlauf.



Bei einer gegebenen Ursprungsordinate wird WP1 in der angegebenen Richtung lokalisiert. Ähnlich wie beim 2-Linien-Kurs. WP2 befindet sich in der entgegengesetzten Richtung, ist jedoch als vertikal ausgerichteter Zylinder definiert. Dessen nächstgelegener Punkt wird durch die angegebene Entfernung bestimmt. Der Kurs beginnt, wenn das Flugzeug aus dem Gelb markierten WP1-Bereich in Richtung zum Ursprung herausfliegt. WP2 wird nur erreicht, wenn das Flugzeug in den WP2-Zylinder eintritt. Eine Runde ist abgeschlossen, wenn das Flugzeug zu WP1 zurückkehrt.

## Custom Course

Das folgende Diagramm zeigt einen benutzerdefinierten Kurs.



Benutzerdefinierte Kurse müssen in der Datei Telem2.lua definiert werden. Es können beliebig viele Wegpunkte mit beliebigen Namen und Koordinaten definiert werden. Für einen Benutzerdefinierten Kurs sind mindestens drei Wegpunkte erforderlich. Der erste und der letzte Punkt werden als Linien/Bereiche (Start/Finish) interpretiert, alle anderen sind Zylinder. Der Kurs beginnt, wenn das Flugzeug den Startbereich (hier STA) in Richtung WP1 verlässt. Eine Runde ist beendet, wenn alle Zylinder in der richtigen Reihenfolge erreicht wurden und das Flugzeug in den Zielbereich (hier FIN) einfliegt.

Beachten Sie bitte: Da sich im Diagramm oben Start- und Finish-Bereich überschneiden, ist es möglich eine Runde innerhalb oder außerhalb des Startbereichs zu beenden.

Wenn Sie sich außerhalb des Startbereichs befinden, wenn eine Runde beendet ist, müssen Sie zum Startbereich zurückkehren, um eine nachfolgende Runde zu starten.

## Festlegen von benutzerdefinierten Kurswegpunkten über telem2.lua

Hier ist ein Beispiel für eine benutzerdefinierte Kursdefinition:

```
xcCustom = {
{"STA", 42.824101, -71.423681}, - Parameter sind Wegpunktname, Breite, Länge
{"WP1", 42.824348, -71.428071}, - Es sind mindestens drei Wegpunkte erforderlich
{"WP2", 42.819736, -71.424117}, - Duplikate sind in Ordnung, solange sie nicht benachbart sind
{"WP3", 42.819988, -71.427973}, - Start und Ende (erster und letzter) werden als Linien gerendert
{"FIN", 42.824101, -71.423681}, - Alle anderen Wegpunkte werden als Zylinder gerendert
}}
```

## Standardeinstellungen über telem2.lua festlegen

Telem2.lua ist eine optionale Lua-Quelldatei, die mit jedem Texteditor bearbeitet werden kann.

Jede Änderung an dieser Datei muss eine gültige Lua-Syntax sein, sonst kann der gefürchtete „telem2: Script-Syntaxfehler“ auftreten.

lokale Funktion init ()

```
xcC = {{
42.82189, - Ursprungsbreite
-71.42539, - Ursprungslänge
340, - Kursüberschrift (0 - 360)
350 - Beinlänge
},
{}},
0, - Standardkursart 0 = Dreieck, 1 = 2 Linie, 2 = Linie / Zyl., 3 = Benutzerdefiniert
30, - Zeitlimit in Minuten - Null ist unbegrenzt
0 - Rundenlimit - Null ist unbegrenzt
}}
sim = false - Simulatoroptionen sind true oder false
- Legen Sie die Standardstartposition für den Simulator fest (wird überschrieben, wenn eine GPS-Sperre
auftritt).
cPos [1] = 42,82189
cPos [2] = - 71,42539
xclOWidth = 50 - Breite der In / Out-Ansagen in Metern
Ende
lokaler Funktionslauf (e)
lcd.clear ()
lcd.drawText (1,1, "GPS Racing - Standardeinstellungen und Simulator", 0)
wenn sim dann
lcd.drawText (1,11, "SIMULATION", DBLSIZE)
sonst
lcd.drawText (1,11, "TELEMETRY", DBLSIZE)
Ende
lcd.drawText (106, 16, "Drücken Sie ENT, um den Modus zu ändern", 0)
lcd.drawText (1,31, "Lat:", 0)
lcd.drawText (lcd.getLastPos () + 2,31, rnd (cPos [1], 5), 0)
lcd.drawText (1,41, "Lng:", 0)
lcd.drawText (lcd.getLastPos () + 2,41, rnd (cPos [2], 5), 0)
lcd.drawText (1, 56, "Mem:", 0)
lcd.drawText (lcd.getLastPos () + 4, 56, rnd (collectgarbage ("count"), 4), 0)
wenn e == EVT_ENTER_BREAK, dann ist sim = nicht sim end
Ende
return {init = init, run = run}
```

## **Zukünftige Verbesserungen**

### **Verbesserungen der Telemetrie**

Das Standard-GPS von FrSKY bietet nur Längen- und Breitengrade. Derzeit sind diese Informationen die Grundlage zur Berechnung von Kurs und Geschwindigkeit. Idealerweise möchte ich Kurs und Geschwindigkeit vom GPS selbst erhalten, wenn es im Telemetristream verfügbar ist.