

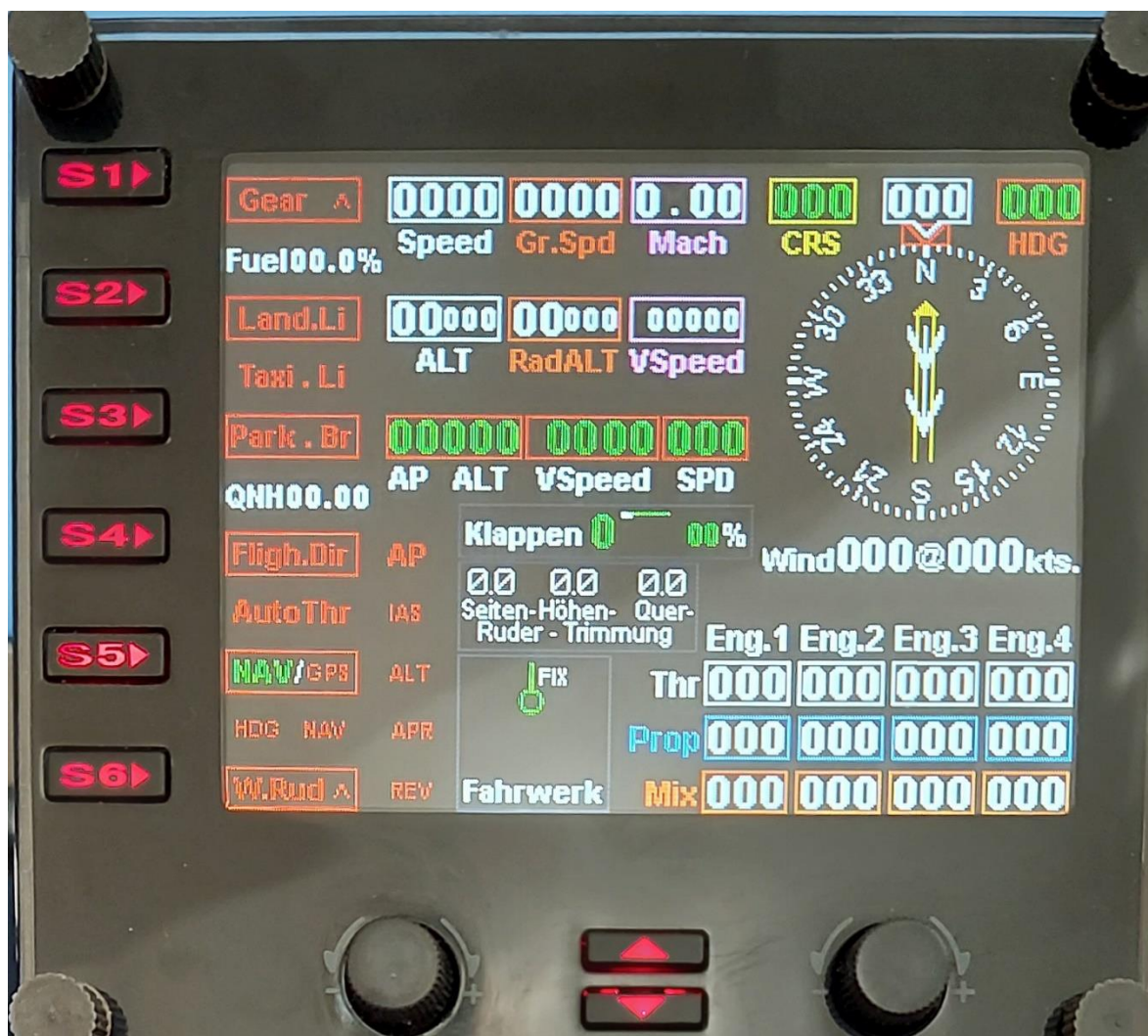
# Info-Gauge (V1.0)

## Index:

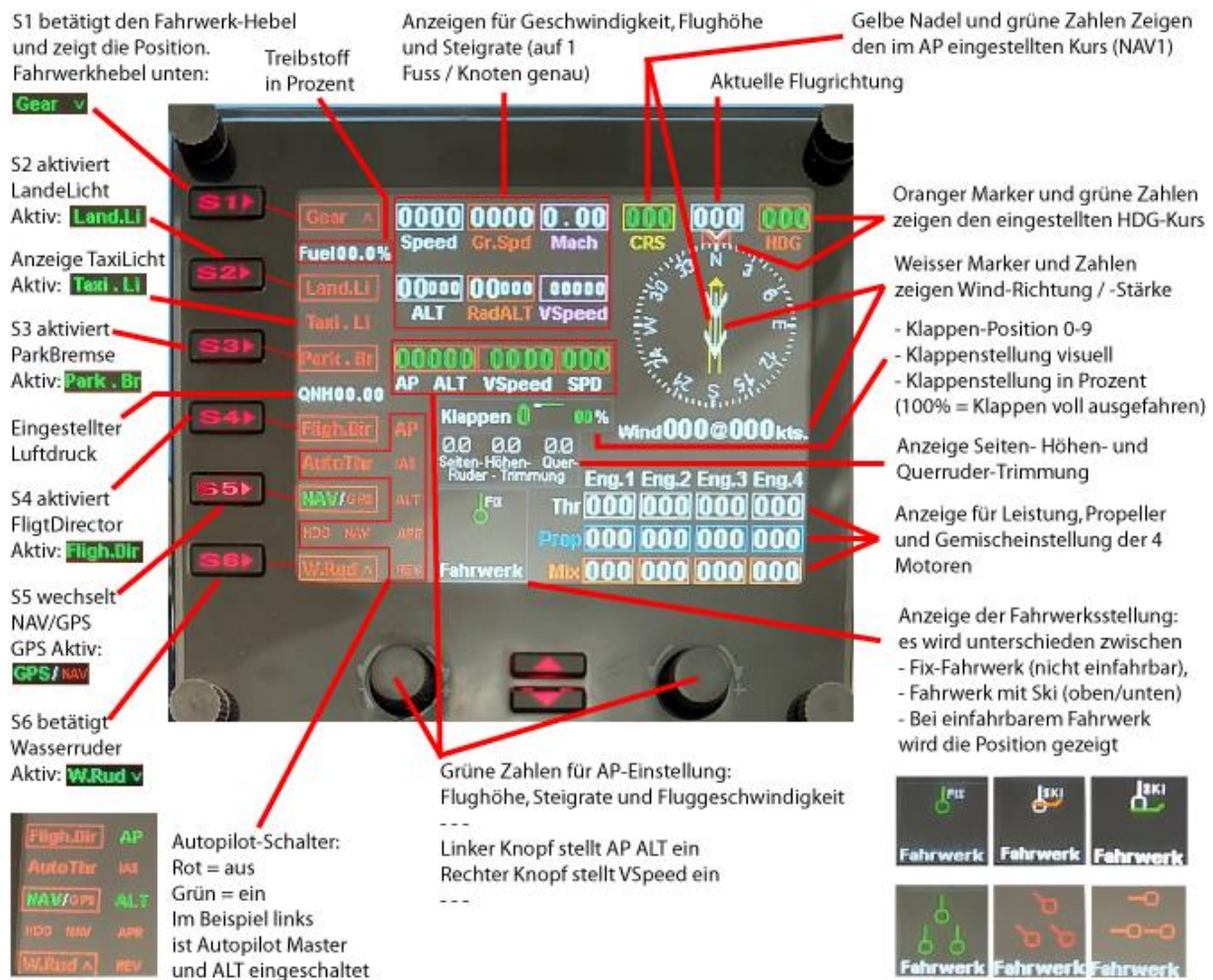
- A) Übersicht
- B) Beschreibung der Anzeigen
- C) Zusätzliche Info zu den Anzeigen
- D) Bekanntes Problem
- E) Warum dieses Gauge
- F) Entstehung des Gauge
- G) Das Gauge selber umbauen

## A) Übersicht

So sieht die Anzeige dieses Gauge aus:



## B) Beschreibung der Anzeigen



## C) Zusätzliche Info zu den Anzeigen

1) "S1 Fahrwerk-Hebel" zeigt nur die Position des Fahrwerk-Hebels. Das Fahrwerk kann trotzdem eine andere Position haben, weil z.B. die Fluggeschwindigkeit für das Ausfahren zu hoch ist.

2) **Anzeigen für Geschwindigkeit** (Speed, Gr.Speed, Mach) zeigen trotz Stillstand am Boden eine gewisse Geschwindigkeit an (z.B. 004 Knoten). Das liegt daran, dass der Wind mit in die Geschwindigkeitsberechnung einfließt.

Bei "004 Knoten" also ein direkter Gegenwind von 4 Knoten, oder ein von schräg-vorne kommender Gegenwind von 10 Knoten.

3) **Warum 3 Anzeigen für Geschwindigkeit** (Speed, Gr.Speed, Mach), ist doch im Prinzip das gleiche? Speed (gemessene Geschwindigkeit im Flugzeug) und Ground-Speed (tatsächliche Geschwindigkeit über Boden) sind in Bodennähe nahezu identisch, mit steigender Höhe weichen diese Werte aber deutlich ab, wegen geringerem Luftdruck (siehe 3a).

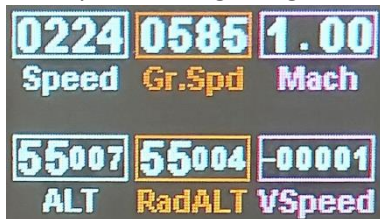
### 3a) Am deutlichsten sieht man die Unterschiede, beim Flug mit der Concorde:

- Auf 28000 Fuss Höhe, mit gemessenen 280 Knoten (Mach 0.69), fliegt man in Wirklichkeit mit 422 Knoten über den Boden.



- Auf 55000 Fuss Höhe, mit gemessenen 224 Knoten (Mach1.00), fliegt man in Wirklichkeit mit 585 Knoten über den Boden.

Laut Speed-Anzeige fliegt man langsamer als in 28000 Fuss Höhe, in echt ist man aber viel schneller.



- Auf 55000 Fuss Höhe, mit gemessenen 454 Knoten (Mach2.02), fliegt man in Wirklichkeit mit erstaunlichen 1170 Knoten über den Boden.



### 4) Die Anzeigen "ALT, RadALT und VSpeed" sind in meinem Gauge genauer als üblich und nicht limitiert:

- Die Anzeige ALT (Flughöhe) zeigt bei meinen bisherigen Gauges nur bis 45000 Fuss Höhe an, alles darüber wird nicht angezeigt. Klar, ausser der Concorde und Kampffjets fliegt kein Flugzeug höher. Da ich aber beides nutze, endet meine Anzeige erst bei 99999 Fuss.

- RadALT (Flughöhe über Boden) zeigt bei meinen bisherigen Gauges nur bis 2000 Fuss über Boden einen Wert. Bei meiner Anzeige endet die Anzeige nicht.

- VSpeed (Steig-/Sink-Rate) zeigt bei meinen bisherigen Gauges erst Abweichungen von +/- 10 Fuss. Meine Anzeige ist auf +/- 1 Fuss genau. Sogar noch etwas genauer, denn die Zeichen +/- schwenken ins Bild, bevor 1 Fuss Unterschied auftritt.

Wenn man ohne Autopilot fliegt, ist es sehr hilfreich, auch kleinste Höhenabweichungen sofort zu erkennen, um entsprechend zu trimmen.



## 5) Klappen:



- Links wird die Klappen-Position 0-9 angezeigt, unabhängig davon, wie die Klappen wirklich stehen. (Bei Motorausfall /keine Energie, um die Klappen zu bewegen / defekten Klappen, kann der Klappenhebel durchaus eine andere Position haben, als die tatsächliche Klappenstellung.)
- In der Mitte wird gezeigt, wie die Klappen stehen, bzw. wie sie gerade aus/ein-fahren.
- Rechts die aktuelle Klappen-Position in Prozent. 0 % = Klappen eingefahren, 100 % = Klappen vollständig ausgefahren.

Bitte nicht verwechseln mit Klappen auf 40° (40Grad) ausgefahren.

Da jedes Flugzeug andere "Grad-Werte" bei den einzelnen Klappenstellungen hat, kann ich die Klappenstellung nicht global in Grad angeben.

Dieses Gauge soll bei möglichst allen Flugzeugen funktionieren.

Wenn ein Gauge für ein bestimmtes Flugzeug gemacht wird, kann man auch problemlos die korrekten Grad-Angaben machen.

## D) Bekanntes Problem

Problem mit der Fahrwerks-Anzeige:

Bei manchen Flugzeugen kommt es gelegentlich vor, dass es so aussieht, als ob das Fahrwerk nicht komplett aus- oder ein-gefahren ist.

Erneutes Klicken auf Aus- und Ein- Fahren (zweimaliges betätigen des Fahrwerk-Hebels) behebt in der Regel den Anzeigefehler.

Allerdings bin ich nicht sicher, ob das ein Problem meiner Anzeige ist, denn andere Gauge zeigen genau das gleiche:

Korrekt ausgefahrenes Fahrwerk.

Links ist meine Anzeige, daneben die Anzeigen von anderen Gauge-Erstellern.



Fahrwerk nicht komplett ausgefahren:

- Anzeige Links: Meine Anzeige, selbsterklärend.
- Anzeige Mitte: Roter Punkt bedeutet, "Fahrwerk beim Ausfahren", grün zeigt "Ausgefahren".
- Anzeige Rechts: Gelb = Ausgefahren, Rot = "Fahrwerk beim Ausfahren".



Fahrwerk nicht komplett eingefahren:

- Anzeige Links: Meine Anzeige, selbsterklärend.
- Anzeige Mitte: Roter Punkt bedeutet, "Fahrwerk beim Einfahren".
- Anzeige Rechts: Rot = "Fahrwerk beim Einfahren".



## E) Warum dieses Gauge

Die meisten meiner Gauges zeigen nicht das Gewünschte, oder nur Teile davon.

Darum habe ich versucht, alles auf einem Bildschirm anzuzeigen. Erstaunlich, wie viel auf diesem kleinen Bildschirm gezeitigt darstellbar ist.

Dies ist mein erstes selbst gebastelte Gauge für Logitech-FIP (flight instrument panel). Bisher habe ich nur vorhandene Gauge für meine Bedürfnisse leicht angepasst.

Andere Gauges versuchen eine Anzeige für Boeing 737, oder Cessna nachzuahmen. Darum ist es nicht verwunderlich, wenn die Anzeigen bei 45000 Fuss Flughöhe enden, denn beide Maschinen fliegen nicht höher.

Mein Gauge soll möglichst für alle Maschinen funktionieren.

Mir ist klar, dass meine Anzeigen nicht den realen Anzeigen eines Flugzeuges entsprechen.

Aber da dies in einem Flugsimulator verwendet wird, ist sowieso nichts wie in Real:

- In Real kann ich den Kopf nach links rechts bewegen, um neben dem Steuerhorn vorbeischaun zu können.
- In Real kann ich mich erheben, um im Landeanflug die Piste besser sehen zu können.
- In Real "spürt" man kleinste Steig/Sink-Änderungen.
- In Real bewegen sich die Leistungshebel, wenn der Autopilot die Leistung ändert.
- usw.

Darum finde ich es legitim, in dem Gauge etwas mehr zu zeigen, als im realen Flugzeug zu sehen ist.

## F) Entstehung des Gauge

**Falls es dich interessiert, wie dieses Gauge entstanden ist:**

Ohne Programmierkenntnisse und ohne geeignetes Programm, habe ich alles im simplen "Text-Editor" von Windows gebastelt.

- Ein Riesen Aufwand, alles im Blindflug zu schreiben, denn jeder Text, jedes Bild, oder Anzeigewert muss Pixelgenau platziert werden (X und Y Koordinaten).

Das Problem dabei: Man kann nicht einfach den Text speichern und das Ergebnis anschauen.

Man muss den Text speichern und den Flugsimulator starten, damit die FIP-Bildschirme aktiviert werden. Um zu testen, ob die programmierten Werte (z.B. Fahrwerk ein/ausfahren) funktionieren, muss man nun einen Flug Starten, abheben und das Fahrwerk einfahren.

- Weiteres Problem: Man muss erst die richtigen "Simulation Variablen" ausfindig machen. Klar, bei vielen Variablen konnte ich bei vorhandenen Gauges "abschauen", aber ich wollte ja anzeigen, welche andere Gauges nicht haben...

- Variable gefunden und alles ist gut? Von wegen! Nun muss man die Variable in das xml-Dokument einfügen, damit auch der richtige Wert ausgegeben wird.

Es wäre schön, wenn man z.B. "Schubhebelstellung von Motor1" abfragen könnte (der zugehörige Parameter lautet übrigens: "A:GENERAL ENG THROTTLE LEVER POSITION:1, percent") und den Wert einfach als Zahl, in Schrift Arial, Grösse 12, ausgeben könnte.

Leider funktioniert das so nicht, man kann auf den Bildschirmen nur Bilder anzeigen, keine Schrift, oder Zahlen.

Wie löst man das Problem?

- 1) Man bastelt eine "Maske", quasi ein Guckloch, durch welches man schaut.
- 2) Man bastelt ein Bild mit allen benötigten Zahlen.
- 3) Nun schiebt man das Bild (Parameter-gesteuert) an der Maske vorbei.

Die Maske ist hier rot dargestellt, in schwarz/weiss das Bild, von 000, über 001, 002, 003, bis 999. (Beachte: Das Zahlen-Bild ist hier beschnitten/gekürzt angezeigt, in echt ist es Meter-hoch!)



Hier wird der komplette schwarze Balken (je nach Wert) nach unten/oben verschoben.

So ein Bild mit allen Zahlen basteln ist extrem aufwändig, wenn man kein geeignetes Programm hat, welches das automatisch generiert.

Darum bin ich einen anderen Weg gegangen:

Ich habe eine 1-stellige Zahlenreihe gebastelt und verwende diese 3x, für eine 3-stellige Zahl.



Hier werden die 3 schwarzen Balken einzeln (je nach Wert) nach unten/oben verschoben.

- Allerdings hat meine Methode einen Haken: Ich muss für alle 3 Stellen einen anderen Parameter verwenden, besser gesagt, ich muss den vorhandenen Parameter auslesen und berechnen, bevor ich

den Wert ausgabe.

Da ich die "xml-Sprache" nicht kenne und nicht Programmieren kann, musste ich bei anderen abgucken und versuchen zu verstehen, was, warum, so gemacht wurde und das Gesehene für meine Zwecke anpassen.

---

Nicht einfach, versuch es selber:

Verstehst du die 4 Berechnungen?

1) Bei obigem Beispiel mit "Schubhebelstellung von Motor1", wird der Wert für die 100er-Stelle (1xx) so berechnet:

$(A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot 100 \div -$

2) Der Wert für die 10er-Stelle (x1x) so:

$(A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot 10 \div (A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot 100 \div 10 * -$

3) Der Wert für die 1er-Stelle (xx1) so:

$(A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot (A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot 10 \div 10 * - \%$

Die Berechnungen funktionieren, trotzdem wird bei "Vollem Schub" nur 99 angezeigt. \*grrr\*

Egal, ob mit dem ThrottleQuadrant (Hardware-Schubhebel von Logitech), oder direkt mit der Maus im Virtuellen Cockpit, selbst "F4" (Tastaturbefehl für vollen Schub), wird nur 99 angezeigt.

Man kann damit leben, aber mich störte das extrem.

Nach langem Suchen habe ich eine Lösung gefunden:

Der Flugsimulator rechnet intern auf mehrere Kommastellen genau, voller Schub wird wohl als 99.9999% gewertet.

Meine Abfrage nach 100% wird also nie erfüllt.

4) Also habe ich diese Berechnung ausgetüftelt, für 100%:

$(A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot (A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \% - 10 * \% (A:GENERAL\ ENG\ THROTTLE\ LEVER\ POSITION:1, percent) \cdot 1 \div 10 * + 999 ==$

Hast du die Berechnungen verstanden?

---

Es geht noch eine Stufe schwerer:

Ich wollte anzeigen, wie viel Prozent Treibstoff noch im Tank ist. Dafür gibt es keinen Parameter, den man auslesen kann.

Man kann nur auslesen, wie viel maximal getankt werden kann und wie viel aktuell im Tank ist.

Also muss man aus diesen beiden Werten ermitteln, wie viel Prozent das ist und diesen Wert für die Anzeige weiter berechnen.

**Langer Rede kurzer Sinn: Das Basteln des Gauge hat rund 1 Monat gedauert, bei 2-3h pro Tag.**

**Hier die Entwicklungsstufen:**

Tag 1, erste Versuche mit einigen Parametern.



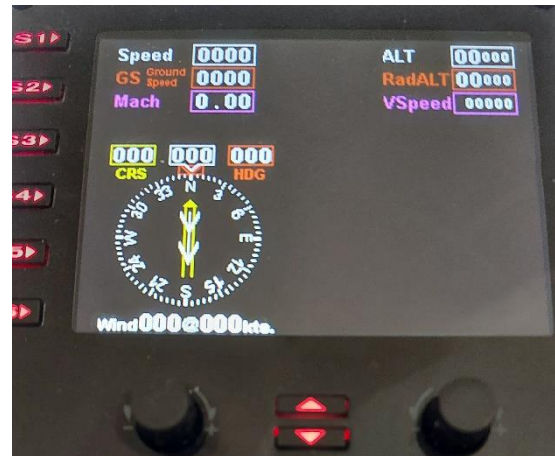
Tag 2, Windrichtung, Flugrichtung und HDG-Zeiger eingebaut.



Tag 3, Alt, Radar Alt und V Speed eingebaut.



Tag 5, CRS, HDG und Wind als Text eingefügt.



Tag 15, alles neu angeordnet, damit Anzeigen für die S-Tasten Platz haben.

Das war ein mühsames herum Geschiebe, denn jeder Rahmen, jede Zahl, jedes Zeichen musste Pixelgenau mit X und Y Koordinaten im Texteditor umgeschrieben werden.

Throttle, Propeller und Gemisch für 4 Triebwerke eingebaut.





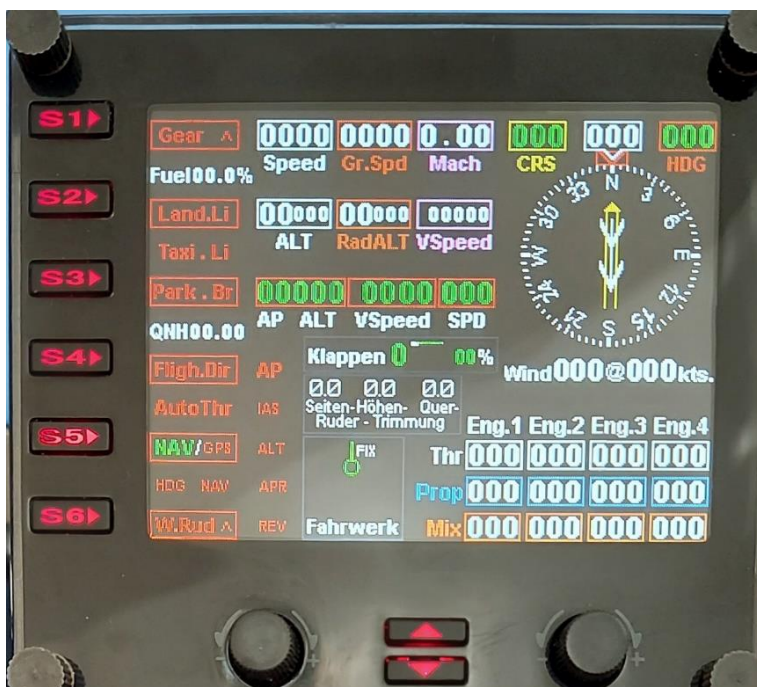
Tag 20, Funktionen für die S-Tasten und diverse Anzeigen des Autopiloten eingebaut.



Tag 29, Fahrwerk und Klappenanzeige eingebaut. Beim Fahrwerk wird unterschieden zwischen Fix-Fahrwerk (nicht einfahrbar), Fahrwerk mit Ski (oben/unten) und bei einfahrbarem Fahrwerk wird die Position (unten/unterwegs/oben) gezeigt.



Tag 31, QNH, Treibstoff und Trimmung eingebaut. Fertig!



## G) Das Gauge selber umbauen

Du darfst das Gauge nach Belieben den eigenen Bedürfnissen anpassen.

Wenn du zum Beispiel die Drehregler nicht mit AP-ALT und AP-VSpeed belegt haben willst, sondern mit Seiten- und Quer-Ruder-Trimming, öffne mit dem Editor von Windows das "Info.xml" und ersetze den Text ab <!-- Drehregler --> (ganz unten im Dokument), durch folgende Zeilen:

```
<!-- Drehregler -->
<!-- Rechter Regler stellt Aileron (Querruder) Trimmung ein -->
<Mouse>
<Area>
<Area>
<Cursor Type="DownArrow" />
<Click Event="AILERON_TRIM_LEFT" Repeat="Yes" />
</Area>
<Area>
<Cursor Type="UpArrow" />
<Click Event="AILERON_TRIM_RIGHT" Repeat="Yes" />
</Area>
</Area>
</Mouse>

<!-- Linker Regler stellt Rudder (Seitenruder) Trimmung ein -->
<Mouse>
<Area>
<Area>
<Cursor Type="DownArrow" />
<Click Event="RUDDER_TRIM_LEFT" Repeat="Yes" />
</Area>
<Area>
<Cursor Type="UpArrow" />
<Click Event="RUDDER_TRIM_RIGHT" Repeat="Yes" />
</Area>
</Area>
</Mouse>

Achtet darauf, dass unterhalb noch das steht:

</Gauge>
```

Da dieser Text nicht problemlos aus diesem pdf kopiert werden kann, findest du den zu kopierenden Text in der beiliegenden Textdatei "Trimm.txt".

-----

Ich hoffe, dir gefällt das Gauge, gib doch bitte eine Rückmeldung an [p3d@andi20.ch](mailto:p3d@andi20.ch).  
Verbesserungsvorschläge und Fehlermeldungen werden gerne angenommen.