

Test Handling Turbinenmodelle:

1. Installation:

Lupa App über setup batch installieren, bei Registrierung alle zusätzlich (auch leere Apps) deinstallieren. **Es darf nur die App Template aktiv sein!!!**

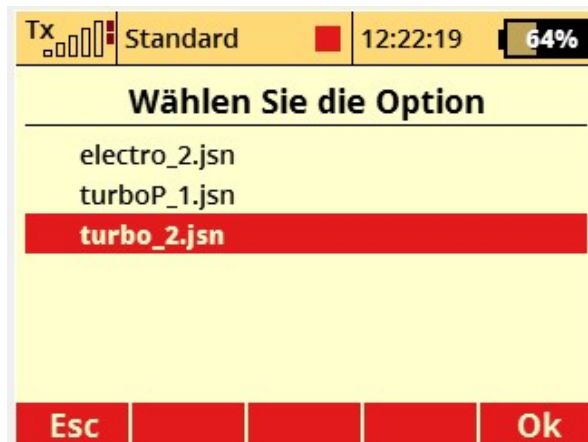


Dann Telemetriefenster registrieren



2. Datafile laden:

Für jedes Modell wird ein jsn file mit Modellname.jsn angelegt. In diesem File werden alle Sensorkonfigurationen gespeichert. Bei Inbetriebnahme wird das erste Datafile aus der Templateliste geladen.



turboP_1 ist ein Template für nur eine Telemetrie-seite

turboP_2 ist ein Template für aufgeteilt auf zwei Telemetrieseiten.

Electro_2 ist ein Template für Elektromodelle auf zwei Telemetrieseiten

3. Schalter zur Umschaltung der Telemetrieseite konfigurieren (steht nur bei Doppelseiten zur Konfiguration):

Tx [Signal] Standard [Red Light] 12:22:06 64%

App Template

Konfiguration

DataFile turbo_2.jsn

TeleScreen2 Sf X

Kapazität (ml) 2400

Zählwert Kapazität 100

ECU Typ JetCat

Powered by Geierwally - V1.1.1 Mem max: 66K

ScrLib [] [] [] Ok

Konfiguration von zwei getrennten Telemetrieseiten ist aus Speichergründen auf den 14/16-er Sendern nicht möglich. Aus diesem Grund wurde das Konzept über Schalter implementiert. Das hat auch den Vorteil, dass man z.B. zwei Seiten in Abhängigkeit des Flugzustandes schalten kann.

4. ECU Config laden (nur bei turboP Modellen):

Tx [Signal] Standard [Red Light] 12:22:20 64%

App Template

Konfiguration

DataFile 0001Turb.jsn

ECU Typ HORNET

Powered by Geierwally - V1.1.1 Mem max: 65K

ScrLib [] [] [] Ok

Turbine im ECU Config auswählen für Statusanzeige

5. Sensoren zuweisen:






Tx	Standard	12:22:57	64%
Anzeigen Bibliothek			
Konfiguration			
Zuweisung Sensoren			
Label	Tankinhalt		
Sensor	PBS GPS		
SensParam	SPEED		
Powered by Geierwally - V1.1.1 Mem max: 65K			
			Ok

Jedem Rahmenlabel den entsprechenden Sensor zuweisen. Bei Erstinbetriebnahme ist zum Einlesen der Sensoren erforderlich, den Empfänger einzuschalten. Dabei kann kommt es bedingt durch fehlende Sensorzuweisung zu akustischen Alarmausgaben. Diese werden zurückgesetzt wenn alle Sensoren gebunden sind.

6. Prüfung der Telemetriehauptseiten:

Tx	Standard	12:22:31	64%	Tx	Standard	12:22:51	64%
D-FLAC Turbo Porter Scr1				D-FLAC Turbo Porter Scr2			
Temperatur Turbine	0%	Drehzahl Turbine	0 1/min	Drehzahl Propeller	0%	Geschw	0 km/h
0 °C	V	0 1/min		0 1/min	V	0 km/h	
Tank		Status Turbine	OFF	Höhe		Strecke	0 km
0 ml		OFF		0 m		0 km	
		Bat1 0.00 V 0 mAh					
		Bat2 0.00 V 0 mAh					
			Start				Start
		Clr				Clr	

Die Hauptseiten bei geladenem turboP_2.jsn Template (zwei Telemetrieseiten)

Tx 		Standard 	12:22:28	 64%
D-FLAC Turbo Porter Scr1				
Turb. 0 1/min	0%	Prop. 0 1/min		
Höhe 0 m	 V	Tank 0 ml		
Geschw 0 km/h		Stat OFF		
Strecke 0 km	 	Temp. 0 °C		
Bat1 0.00 V 0 mAh		L		
Bat2 0.00 V 0 mAh				
				Start Clr

Hauptseite bei geladenem turboP_1.jsn Template (eine Telemetrieseite)

Von jeder geladenen Konfiguration bitte zur Ermittlung des Speicherbedarfes Screenshots der Konsole speichern

7. Hinweise zur Datafile – Struktur der konfigurierten Frames:

Die verwendeten Frames sind wie folgt definiert:

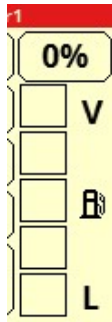
Mittelteil Tankanzeige:

```

Main Turbine:
[1] 3           Modelltyp 3      (1 Elektro  2 Verbrenner 3 Turbine , 4 Segler)
[2] "Tankinhalt"  Framelabel, Text wird in der Konfig der Sensorzuweisung aufgelistet
[3] "ml"         Einheit ist ml , wird im Frame dargestellt
[4] 1           Wert wird in Templateapp in % zur Füllstandsanzeige umgerechnet
[5] 30          Ab 30% erfolgt Fehlermeldung
[6] 100         Bei 100 % (voll getankt) wird Fehler wieder gelöscht
[7] 0           Platzhalter Anzahl der Nachkommastellen (wird hier nicht verwendet)
[8] 0           von Screenlib gesetzt aktueller Wert
[9] 0           von Screenlib gesetzt wenn 1, ist Alarm aktiv
[10] 1          von Screenlib gesetzt zugewiesene SensorID
[11] 1          von Screenlib gesetzt zugewiesener Sensor Parameter
[12] 0          von Screenlib gesetzt (Platzhalter für spätere Anwendung)

Beispiel nach obiger Beschreibung:
[3,"Tankinhalt","ml",1,30,100,0,0,0,1,1,0]

[3,"Tankinhalt","ml",1,30,100,0,0,0,1,1,0]
```



Frametype 1 Normalfenster :

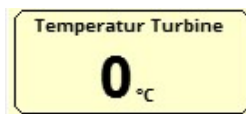
Normalfenster

```
[1] 1
[2] "Temperatur Turbine"
[3] "°C"
[4] 0
[5] 0
[6] 0
[7] 0
[8] 0
[9] 0
[10] 1
[11] 1
[12] 0
```

Beispiel nach obiger Beschreibung:

```
[1,"Temperatur Turbine","°C",0,0,0,0,0,0,1,1,0],
```

```
[1,"Temperatur Turbine","°C",0,0,0,0,0,0,1,1,0],
```



Frametype 1 (1 Normalfenster, 2 MinMax, 3 geteiltes Fenster, 4 Al
Framelabel, Text wird in der Konfig der Sensorzuweisung aufgelist
Einheit ist °C , wird im Frame dargestellt
Wert kommt direkt vom Telemetriesensor, wenn > 0 dann von der App
Alarmwert in dem Fall kein Alarm
Alarmwert gelöscht, in dem Fall ist kein Alarm konfiguriert
Anzahl der Nachkommastellen
von Screenlib gesetzt aktueller Wert
von Screenlib gesetzt wenn 1, ist Alarm aktiv
von Screenlib gesetzt zugewiesene SensorID
von Screenlib gesetzt zugewiesener Sensor Parameter
von Screenlib gesetzt (Platzhalter für spätere Anwendung)

Frametype 3 geteiltes Fenster :

Geteilte Fenster

```
[1] 3
[2] "Höhe"
[3] "m"
[4] 0
[5] 0
[6] 0
[7] 0
[8] 0
[9] 0
[10] 1
[11] 1
[12] 0
```

Beispiel nach obiger Beschreibung:

```
[3,"Höhe","m",0,0,0,0,0,0,0,1,1,0],
```

```
[3,"Höhe","m",0,0,0,0,0,0,0,1,1,0],
```



Frametype 3 (1 Normalfenster, 2 MinMax, 3 geteiltes Fenster, 4 Abschl
Framelabel, Text wird in der Konfig der Sensorzuweisung aufgelistet
Einheit ist m , wird im Frame dargestellt
Wert kommt direkt vom Telemetriesensor, wenn > 0 dann von der App
Alarmwert in dem Fall kein Alarm
Alarmwert gelöscht, in dem Fall ist kein Alarm konfiguriert
Anzahl der Nachkommastellen
von Screenlib gesetzt aktueller Wert
von Screenlib gesetzt wenn 1, ist Alarm aktiv
von Screenlib gesetzt zugewiesene SensorID
von Screenlib gesetzt zugewiesener Sensor Parameter
von Screenlib gesetzt (Platzhalter für spätere Anwendung)

Frametype 5, Abschluss mit 4 Anzeigewerten:

Jedem Frame ist in der scrib config ein Sensorparameter zuzuordnen!!!

```
Abschlussfenster 5 Werte es sind vier Zeilen zu konfigurieren, welche sich nur im Fensterwert unterscheiden
[1] 5                               Frametype 5 , 4 Anzeigewerte
[2] "Bat1"                         Framelabel Zeile 1 ist Bat1
[3] "v"                             Einheit ist Volt
[4] 0                               Wert kommt direkt vom Telemetriesensor
[5] 0                               Alarmwert kein Alarm konfiguriert
[6] 0                               Alarm zurückgesetzt , in dem Fall kein Alarm
[7] 2                               Anzahl der Nachkommastellen für Spannung 2 , für Strom 0
[8] 0                               von Screenlib gesetzt aktueller Sensorwert
[9] 0                               von Screenlib gesetzt wenn 1 Alarm aktiv
[10]1                               von Screenlib gesetzt SensorID
[11]1                               von Screenlib gesetzt Sensor Parameter
[12]0                               vorbereitet für Outputcontrol
[13]1                               Framewert 1 (bei diesem Frametyp von 1 - 4)
[14]""                              optional Label für jeden Wert , hier aus Platzgründen nicht definiert
```

Beispiel nach obiger Beschreibung:

```
[5,"Bat1","v",0,0,0,2,0,0,1,1,0,1,""],
[5,"Bat1","mAh",0,0,0,0,0,0,1,1,0,2,""],
[5,"Bat2","v",0,0,0,2,0,0,1,1,0,3,""],
[5,"Bat2","mAh",0,0,0,0,0,0,1,1,0,4,""],
```

```
[5,"Bat1","v",0,0,0,2,0,0,1,1,0,1,""],
[5,"Bat1","mAh",0,0,0,0,0,0,1,1,0,2,""],
[5,"Bat2","v",0,0,0,2,0,0,1,1,0,3,""],
[5,"Bat2","mAh",0,0,0,0,0,0,1,1,0,4,""],
```

Bat1 0.00 v 0 mAh
Bat2 0.00 v 0 mAh