

## 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 Name des Fensters setzt sich aus Modellname + SCR1 zusammen



### 2. Datafile laden:

Bei Inbetriebnahme wird das erste Datafile aus der Templateliste geladen.



turboP\_1 ist ein Template für nur eine Telemetrieseite

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 icon] 12:22:06 64%

### App Template

**Konfiguration**

DataFile turbo\_2.json

**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 icon] 12:22:20 64%

### App Template

**Konfiguration**

DataFile 0001Turb.json

**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 [signal] Standard [red icon] 12:22:57 64%

### Anzeigen Bibliothek

**Konfiguration**

**Zuweisung Sensoren**

**Label** Tankinhalt

Sensor PBS GPS

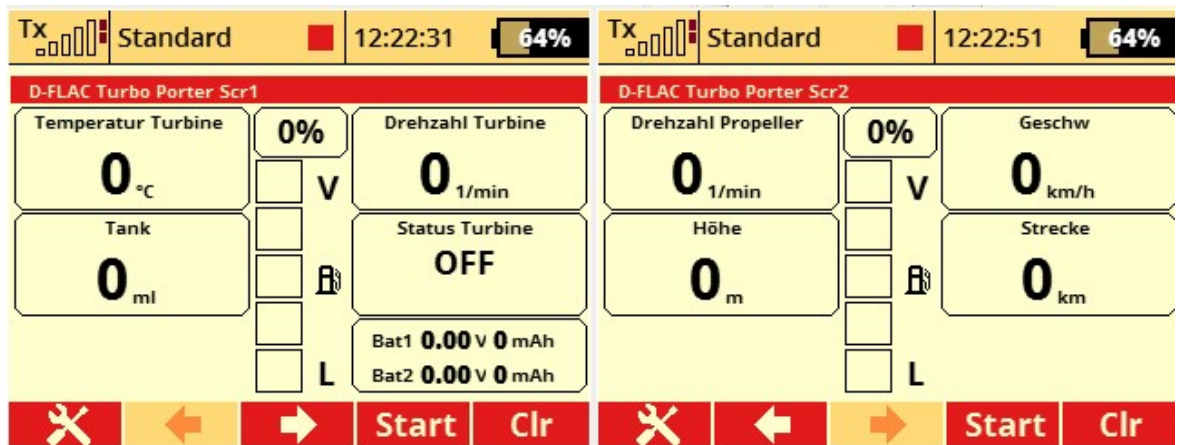
SensParam SPEED

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

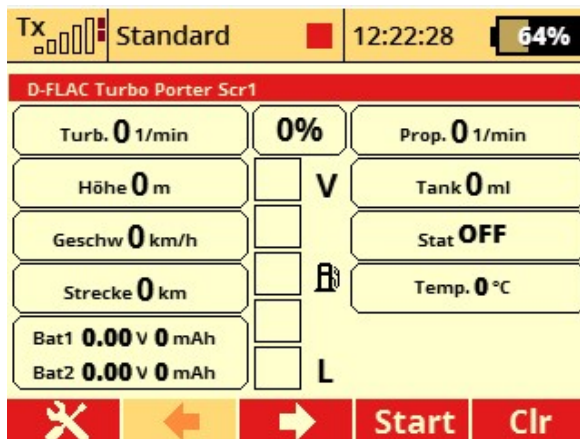
[ ] [ ] [ ] Ok

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

## 6. Prüfung der Telemetriehauptseiten:



Die Hauptseiten bei geladenem turboP\_2.jsn Template (zwei Telemetrieseiten)



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:

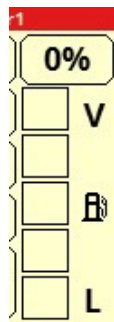
Die verwendeten Frames sind wie folgt definiert:

### 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]

```

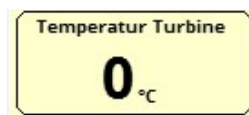


### Frametype 1 Normalfenster :

```

Normalfenster
[1] 1           Frametype 1 (1 Normalfenster, 2 MinMax, 3 geteiltes Fenster, 4 Al
[2] "Temperatur Turbine" Framelabel, Text wird in der Konfig der Sensorzuweisung aufgelistet
[3] "°C"        Einheit ist °C , wird im Frame dargestellt
[4] 0           Wert kommt direkt vom Telemetriesensor, wenn > 0 dann von der Ap
[5] 0           Alarmwert in dem Fall kein Alarm
[6] 0           Alarmwert gelöscht, in dem Fall ist kein Alarm konfiguriert
[7] 0           Anzahl der Nachkommastellen
[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:
[1,"Temperatur Turbine","°C",0,0,0,0,0,0,0,1,1,0],
[1,"Temperatur Turbine","°C",0,0,0,0,0,0,0,1,1,0],

```



### Frametype 3 geteiltes Fenster :

Geteilte Fenster	
[1] 3	Frametype 3 (1 Normalfenster, 2 MinMax, 3 geteiltes Fenster, 4 Abschl
[2] "Höhe"	Framelabel, Text wird in der Konfig der Sensorzuweisung aufgelistet
[3] "m"	Einheit ist m , wird im Frame dargestellt
[4] 0	Wert kommt direkt vom Telemetriesensor, wenn > 0 dann von der App
[5] 0	Alarmwert in dem Fall kein Alarm
[6] 0	Alarmwert gelöscht, in dem Fall ist kein Alarm konfiguriert
[7] 0	Anzahl der Nachkommastellen
[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,"Höhe","m",0,0,0,0,0,0,1,1,0],
```

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

Höhe **0** m

### 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