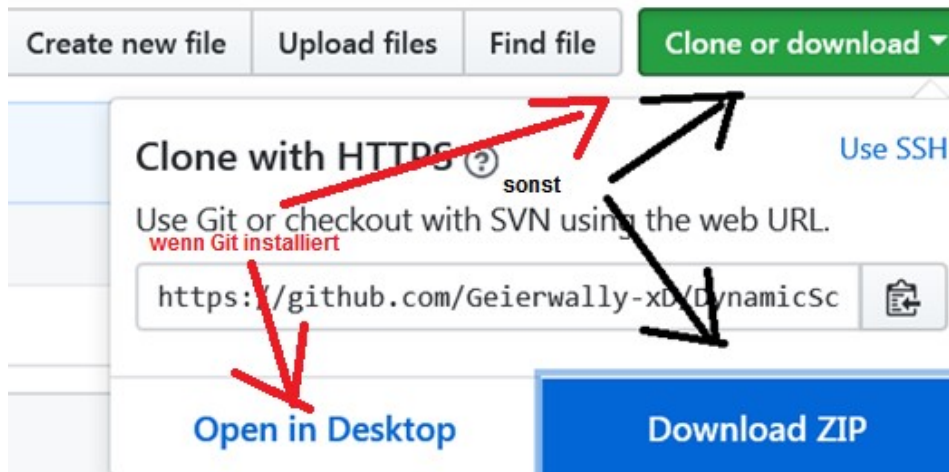


## Screenlib Testprogramm 1 (Formatierung der Ausgaben auf 14, 16 und 24 er Sendern)

### Installation:

1. Git downloaden und installieren (nur wenn noch nicht installiert)  
<https://git-scm.com/downloads>
2. Link <https://github.com/Geierwally-xD/DynamicScreenLibrary/tree/ScreenFormatTests> öffnen
3. Wenn das Repository noch nicht auf lokalem PC existiert, zip downloaden und entpacken bzw. wenn Git installiert ist über open in Desktop in ein lokales Verzeichnis auf den PC laden.



4. Sender über USB verbinden und im Verzeichnis DynamicScreenLibrary-ScreenFormatTests\AppTemplate AppTemplate\_Setup.bat ausführen. Wenn fehlerfrei beendet, USB Verbindung trennen.

### Konfiguration:

1. App Template registrieren unter Zusatzfunktionen\Benutzerapplikationen (keine weitere App aktivieren!!!)



2. Telemetrieseiten registrieren unter Stoppuhren\Sensoren\Telemetrieanzeige (hier auch nur diese beiden Seiten registrieren, Name setzt sich aus Modellname + Scr1 bzw. Scr2 zusammen) Telemetrie Zoom auf S stellen Reihenfolge ist 1 Scr1 und 2 Scr2

Telemetrieanzeige			
		Telemetrie:	Zoom
1	Test Scr1	▼	... ▼
2	Test Scr2	▼	... ▼

3. Unter Systemfunktion\Senderkonfiguration einen Schalter für die Screenshots konfigurieren

Senderkonfiguration	
Temperaturbereich	Celsius
Tx Frequenz	100Hz
Startmeldung unterdrücken	Ja
Reversierung Menünavigation	Ja
Endlosmenue	Ja
<b>Screenshot-Schalter</b>	<b>Sj ✓</b>
PPM Einstellungen	Aus

4. Unter Hauptmenü App Template folgendes konfigurieren: Als Datafile die electro.jsn, Anzahl Lipozellen =1 , Kapazität und Zählwert Kapazität bleibt auf Defaultwerten.

App Template	
<b>Konfiguration</b>	
DataFile	electro.jsn
<b>Anzahl Lipozellen</b>	<b>1</b>
Kapazität (mAh)	2400
Zählwert Kapazität	100
<b>SensorSimulation</b>	
simCellVoltage	Aus

5. Für simCellVoltage, simCapacity und simRPM jeweils einen Proportionalgeber wie folgt konfigurieren: Prop ohne Mittelstellung für SimCapacity noch Revers aktivieren

Tx	Standard	12:22:29	64%	Tx	Standard	12:22:15	64%
<b>App Template</b> Kapazität (mAh) 2400 ▾ Zählwert Kapazität 100 ▾ <b>SensorSimulation</b> simCellVoltage P5 <input type="checkbox"/> simCapacity P6 <input type="checkbox"/> simRPM P2 <input type="checkbox"/> Powered by Geierwally - V1.1.1 Mem max: 49K				<b>Geber auswählen</b> <div style="border: 1px solid black; padding: 10px; text-align: center;"> <b>P6</b>          87%     </div>			
ScrLib			Ok	Mitte	Prop.	Rev.	Clr Ok

Über die Proportionalgeber werden Sensorwerte für Akkukapazität, Zellspannung und Drehzahl simuliert.

6. Wechsel auf Hauptseite Telemetriescreen 1 und mit den konfigurierten Gebern Limits von Kapazität, Spannung und Drehzahl testen. Kommt alles korrekt formatiert? Jeweils einen Screenshots von Normal und Limit speichern.

Tx	Standard	12:22:35	64%	Tx	Standard	12:22:50	64%
<b>Test Scr1</b> <div> <div>Drehzahl <b>12156</b> 1/min min:12156 max:25000</div> <div>58% </div> <div>Velocity <b>83</b> km/h</div> <div>Acceleration <b>3.3</b> g</div> <div>Current <b>4.12</b> A</div> <div>Kapazitätsverbrauch <b>3250</b> mAh</div> <div>Empfängerspannung U1: <b>5.70</b> V U2: <b>4.50</b> V</div> <div>4.01V</div> <div>RX1 A1: <b>93</b> % A2: <b>63</b> % RX2 A1: <b>83</b> % A2: <b>75</b> %</div> </div>				<b>Test Scr1</b> <div> <div>Drehzahl <b>20856</b> 1/min min:4656 max:25000</div> <div>43% </div> <div>Velocity <b>83</b> km/h</div> <div>Acceleration <b>3.3</b> g</div> <div>Current <b>4.12</b> A</div> <div>Kapazitätsverbrauch <b>3250</b> mAh</div> <div>Empfängerspannung U1: <b>5.70</b> V U2: <b>4.50</b> V</div> <div>3.20V</div> <div>RX1 A1: <b>93</b> % A2: <b>63</b> % RX2 A1: <b>83</b> % A2: <b>75</b> %</div> </div>			
			Start				Start Clr

7. Wechsel auf Telemetriescreen 2 und Formatierung der Ausgaben prüfen 1 Screenshot speichern.

Tx	Standard	12:22:14	64%
<b>Test Scr2</b> <div> <div>FIDir: <b>255.25</b> °</div> <div>70% </div> <div>Pos: <b>125.25</b> °</div> <div>Entf: <b>1250</b> m</div> <div>B: <b>0° 0.000000'</b> N</div> <div>L: <b>0° 0.000000'</b> E</div> <div>Strecke: <b>1250</b> m</div> <div>Vario: <b>0.75</b> m/s</div> <div>3.70V</div> </div>			
			Start Clr

8. Wechsel auf App Template Config Seite und Datafile Stroke.jsn laden und zurück auf Telemtriescreen 1

9. Auch hier wieder Formatierung der Ausgaben mit Limit prüfen und jeweils einen Screenshot speichern. Füllstand wird über Kapazitätsgeber simuliert.

10. Telemtriescreen 2 entspricht dem der der electro.jsn 1 Screenshot speichern



11. Zurück in die Konfig und testTyp1.jsn laden



12. Auf Telemetriescreen 1 und 2 jeweils Formatierung von Anzeigetyp 1 normal sowie im Limit prüfen und jeweils einen Screenshot speichern



13. Die gespeicherten Screenshots vom Sender laden und gezippt auf PC speichern.

14. Zurück in die Konfig und testTyp2.jsn laden

**App Template**

**Konfiguration**

DataFile testTyp2.jsn

Kapazität (ml) 2400

Zählwert Kapazität 100

**SensorSimulation**

simCapacity P6

ScrLib Ok

15. Auf Telemetriescreen 1 und 2 jeweils Formatierung von Anzeigetyp 2 normal sowie im Limit prüfen und jeweils einen Screenshot speichern. Die Drehzahl wird mittels dafür konfigurierten Proportionalgeber simuliert

**Test Scr1**

Drehzahl 5338 1/min min:1413 max:25000

Drehzahl 5338 1/min min:1413 max:25000

RX1 A1: 82 % A2: 72 %

RX2 A1: 94 % A2: 63 %

**Test Scr2**

Drehzahl 25000 1/min min:1413 max:25000

Drehzahl 25000 1/min min:1413 max:25000

RX1 A1: 53 % A2: 54 %

RX2 A1: 94 % A2: 63 %

Start Clr

16. Zurück in die Konfig und testTyp3.jsn laden

**App Template**

**Konfiguration**

DataFile testTyp3.jsn

Kapazität (ml) 2400

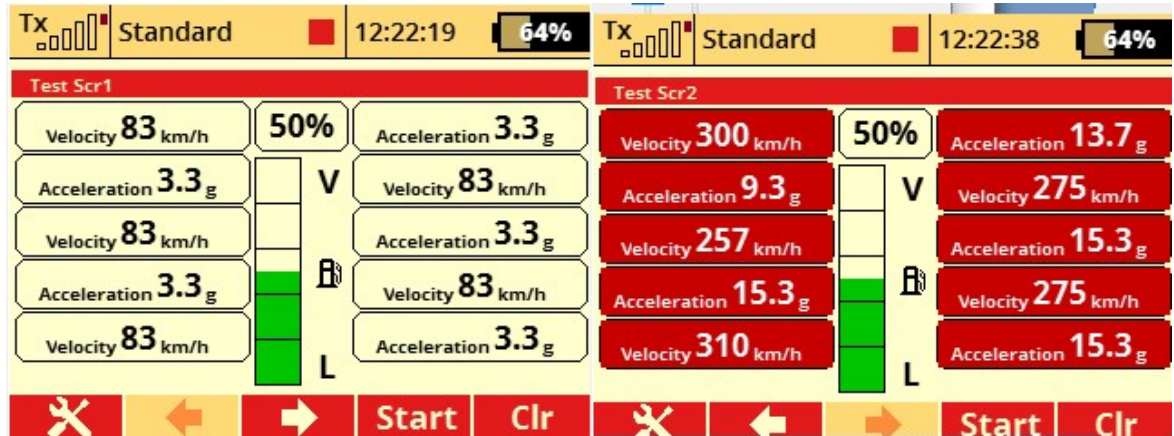
Zählwert Kapazität 100

**SensorSimulation**

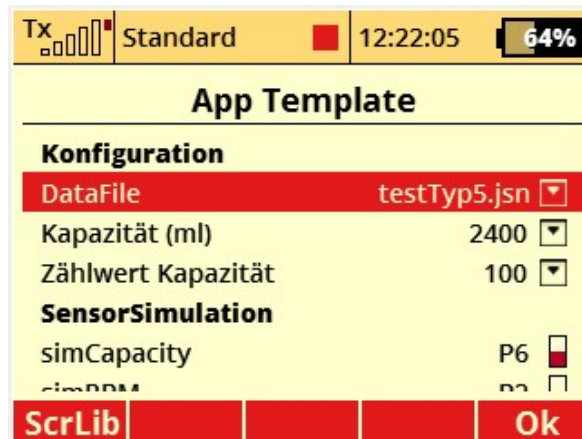
simCapacity P6

ScrLib Ok

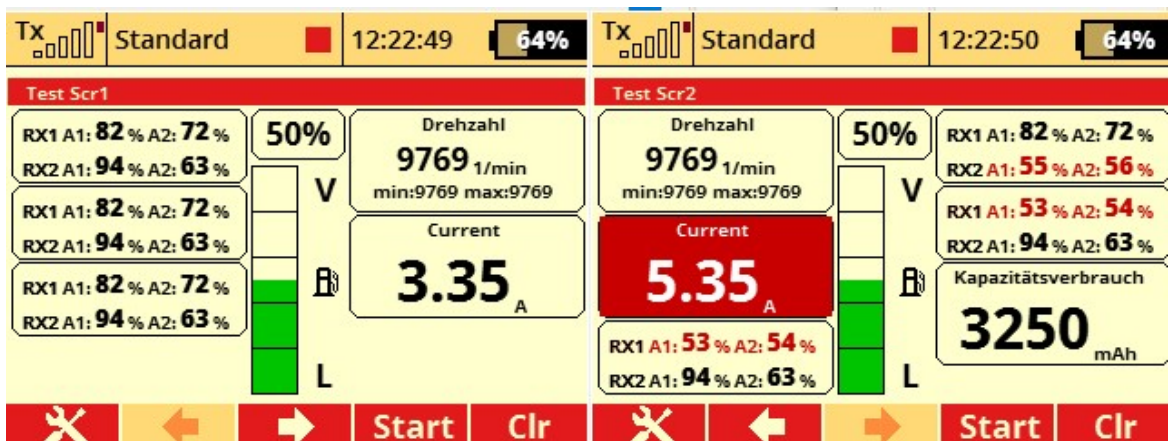
17. Auf Telemetriescreen 1 und 2 jeweils Formatierung von Anzeigetyp 3 normal sowie im Limit prüfen und jeweils einen Screenshot speichern.



18. Zurück in die Konfig und testTyp5.jsn laden



19. Auf Telemetriescreen 1 und 2 jeweils Formatierung von Anzeigetyp 5 normal sowie im Limit in Verbindung mit den Anzeigetypen 1 und 2 prüfen und jeweils einen Screenshot speichern. Drehzahl kann über den konfigurierten Probgeber simuliert werden. Gleiches gilt für Füllstandsanzeige



20. Zurück in die Konfig und in der letzten Zeile ganz rechts den Speicherbedarf der App prüfen. Sollte bei den 14-ern und 16-ern < 50K sein. Screenshot speichern und dann die electro.jsn als Datafile laden sowie über ScrLib (Key1) in die Screenlib Konfig wechseln

Tx Standard 12:22:51

### App Template

DataFile testTyp5.jsn

Kapazität (ml) 2400

Zählwert Kapazität 100

**SensorSimulation**

simCapacity P6

simRPM P2

Powered by Geierwally - V1.1.1 Mem max: 50K

ScrLib Ok

21. Hier die simulierte Sensorzuweisung testen. Über Selectbox Label einen Fensterlabel wählen und über Sensor dem Fenster einen simulierten Telemetriesensor zuweisen. (Unterschiedliche Werte speichern) Dann Sender aus \ ein und prüfen, ob die zugewiesene Konfiguration übernommen wurde.

Tx Standard 12:22:48

### Anzeigen Bibliothek

**Konfiguration**

**Zuweisung Sensoren**

Label Kapazität

Sensor Sens8

Powered by Geierwally - V1.1.1 Mem max: 51K

Ok