

# Stand LowFake

Sebastian Steinhauser

6. Februar 2025

## 1 Stand des Projekts

Aktuell sieht das Skript vor, eine bereits bestehende `.s12`-Datei zu nutzen. Diese wird dann decodiert und in eine `.csv`-Datei umgewandelt. In der CSV-Datei werden alle Informationen, die in der `.s12`-Datei als Offset enthalten sind, in einzelne Spalten (Comma Separated Values) aufgeteilt.

Zur Findung der Offsetwerte wurden hauptsächlich zwei Quellen genutzt:

- Zum Einen:  
<https://wiki.openstreetmap.org/wiki/SL2> (Abgerufen am 05.02.25). Diese Quelle enthält allerdings mehr unbekannte Offsetabschnitte als die nachfolgende.
- Zum Anderen:  
[https://www.memotech.franken.de/FileFormats/Navico\\_SLG\\_Format.pdf](https://www.memotech.franken.de/FileFormats/Navico_SLG_Format.pdf) (Abgerufen am 05.02.25). Hier befinden sich die Offset-Tabellen ab dem Eintrag "Frame for Format 2" (für `.s12`-Dateien).

Das `.s12`-File ist im LittleEndian-Format geschrieben. Als Header werden lediglich die ersten 10 Byte verwendet, die nur einmal zu Beginn der Datei erscheinen. Anschließend folgt vor jedem eigentlichen Sonardatenblock ein lokaler Header von 144 Byte Länge, der unter anderem unbekannte Offset-Bereiche enthält. Darauf folgen die Sonardatenblöcke, die in den vorliegenden Dateien stets 1920 Byte lang sind.

Derzeit ist es möglich, die `.s12`-Daten in ein `.csv`-File zu konvertieren. Allerdings bleiben die unbekannten Offset-Bereiche unaufgeklärt. Falls diese für die Verarbeitung in ReefMaster relevant sein sollten, müssen sie ermittelt werden. Erste Annahmen gehen jedoch dahin, dass diese Bereiche nicht ausschlaggebend sind, da im Skript versucht wurde, sie komplett „durchzureichen“.

Aktuell liegt das Problem eher in der Datenverarbeitung der Python-Skripte, wobei Fehler (möglicherweise fehlerhafte TypeCasts) zu einer fehlerhaften Behandlung der Datentypen führen.

## 2 Ziel

Das Hauptziel besteht darin, die `.csv`-Dateien eines Deeper Chirp+ Sonars so aufzubereiten, dass sie in ReefMaster 2.0 importiert werden können.

Da ReefMaster 2.0 keine Unterstützung für Deeper-Dateien (`bathymetry.csv`, `sonar.csv`) bietet, soll die Konvertierung von Deeper-CSV-Dateien in ein ReefMaster-kompatibles `.s12`-Format erfolgen. Dabei wird angenommen, dass unbekannte Bereiche des lokalen Headers mit Mock-Daten befüllt werden können, sofern sie den Import nicht negativ beeinflussen. Außerdem müssen die Einheiten und Projektionen auch noch in die im `.s12`-Format verwendeten Formate konvertiert werden.

## 3 Vorgehen

Das Gesamtverfahren unterteilt sich in mehrere Skripte:

### 1. **SL2 → CSV: [lowranceToHumanReadable.py]**

Das Skript `lowranceToHumanReadable.py` decodiert `.s12`-Dateien in ein lesbares CSV-Format. Dabei werden Datenfelder (z. B. Zeit, Position, Geschwindigkeiten) konvertiert bzw. umgerechnet. Hierzu gehört das Config-File `lowFakeConfig.yaml`, in welchem bestimmt wird welcher Einheit die Daten vorliegen sollen. Diese sollte später auch dafür verwendet werden, um Dateipfade zu konfigurieren. Dieses Skript war lediglich notwendig, da die Konvertierung von Lowrance-Daten in ReefMaster getestet werden sollte. Damit soll das Fehlerpotential, welches zusätzlich bei der Verwendung von Deeper-Daten, welche noch nicht in der Lowrance-Struktur sind, beseitigt werden.

Bei Deeper-Daten besteht das `.csv`-File bereits, sodass dieser Schritt übersprungen werden kann.

### 2. **CSV (Ursprung: Lowrance) → SL2: [onlyLowranceCsvToS12.py]**

Das Skript `onlyLowranceCsvToS12.py` soll die im ersten Schritt entstandene CSV-Datei wieder in eine `.s12`-Datei umwandeln. Dabei werden die unbekannten Teile des lokalen Headers mit Original-Lowrance Daten befüllen. Der Grund warum dieses Skript existiert ist, da hier die Konvertierung von Lowrance-Daten von `.s12` zu `.csv` und dann wieder zu `.s12` getestet werden soll um so direkt festzustellen ob Grundmechanik der Konvertierung funktioniert. Erste Tests zeigen jedoch, dass die von diesem Prozess erzeugten `.s12`-Dateien in ReefMaster nur wenige Punkte importieren. Vermutlich aufgrund fehlerhafter TypeCasts oder einem fehlerhaften Mapping der CSV-Daten in die Binärstruktur.

### 3. **CSV (Ursprung: Deeper) → SL2: [csvToS12]**

Das Skript ist unvollständig und funktioniert nicht. Grund hierfür ist, dass die Konvertierung eines Original-Lowrance-`.s12`-Files in ein `.csv`-File und wieder zurück in ein `.s12`-File zuerst getestet werden sollte.

Das Skript `csvToS12.py` liest eine CSV-Datei ein, aber diesmal von einem von Deeper-Sonardaten erzeugtem `.csv`-File. Erst dann macht es Sinn sich im nächsten Schritt an die Deeper-Daten zu wagen und dessen Konvertierung in ein `.s12`-File zu implementieren. Die CSV-Datei heißt `synchedDeeperData.csv` und wurde in einem anderen Skript außerhalb dieses Projekts bereits erstellt. In diesem befin-

den sich alle nutzbaren Informationen, welche durch das Deeper-Sonar geschrieben werden.

## 4 Probleme und Nächste Schritte

- **SL2 (Lowrance) → CSV: [`lowranceToHumanReadable.py`]**
  - Das Skript `lowranceToHumanReadable.py` ist funktionsfähig, jedoch könnte auch hier die Konvertierung mit fehlerhaften Datentypen erfolgen, was später zu Problem führen könnte.
  - Zu beachten ist hier, dass bei der Funktion `_decode_record()` Alle Offset werde als integers interpretiert wurden, obwohl teilweise laut Dokumentation auch float-Werte vorkommen können. Hintergrund hierfür ist, dass sich bei Blick mit einem hxd-Viever ergeben hat dass die Werte als Integer dargestellt werden, obwohl sie laut Dokumentation auch als Float-Werte vorkommen können.
  - Der erste Eintrag in der originalen Lowrance-Datei zeigt, dass der erste time1-Offseteintrag einen einmaligen Unix-Zeitstempel in Millisekunden liefert, der zur Einordnung der Messung gedacht ist, aber bisher unzureichend verarbeitet wird.
  - Des weiteren ist in der original Lowrance-Datei immer ein Bereich "vor Beginn" der eigentlichen Messdaten, welcher aktuell als fehlerhaft oder unvollständig interpretiert wird und herausgefiltert wird. Dieser Bereich könnte jedoch wichtige Informationen enthalten, die später für die Verarbeitung der Daten relevant sein könnten. Von diesem Bereich oder Schema wird leider nichts in den beiden Dokumentationen erwähnt. (Vergleiche `sl2ToCsvOutput.csv` mit `sl2ToCsvOutput_raw.csv`).
- **CSV (Ursprung: Lowrance) → SL2: [`onlyLowranceCsvToSl2.py`]:**

ReefMaster importiert von den generierten SL2-Dateien nur wenige Punkte. Ursachen könnten fehlerhafte TypeCasts oder ein falsches Mapping der CSV-Daten in die Blockstruktur sein. Generell wir mittlerweile gerade das "Jonglieren" von Datentypen in Python als problematisch angesehen. Der Fehler könnte sich wahrscheinlich im Bereich der Funktion `_create_block()` befinden, da evtl. wieder Datentypen falsch interpretiert werden. Aktuell werden alle Einheiten als Integer behandelt, obwohl laut Dokumentation auch float-Werte vorkommen können. Die Konvertierungen der Einheiten fehlen.
- **CSV (Ursprung: Deeper) → SL2: [`csvToSl2`]**

Der zusätzlicher Konvertierungsweg, um Deeper-CSV-Dateien in das ReefMaster-kompatible SL2-Format zu integrieren, fehlt (bzw ist unvollständig). Die Konvertierungen der Einheiten fehlen. Das Skript erkennt keine nutzbaren Zeilen im CSV-File.

Bei Rückfragen oder zur weiteren Abstimmung (gerne auch per Teams) stehe ich selbstverständlich zur Verfügung. (Email-Adresse: [Sebastian.Steinhauser@newtec.de](mailto:Sebastian.Steinhauser@newtec.de))