

# Großes 'Karten Kochbuch'

## 1. Vorbemerkungen

Diese Ausarbeitung ist die Weiterführung des 'Kleinen Karten Kochbuches'. Es setzt voraus, dass die Software wie im 'Kleinen Kochbuch' beschrieben installiert ist, und der prinzipielle Ablauf verstanden wurde. Es werden in 2 Kapiteln Techniken beschrieben, wie man eigene individuelle Karten erstellt. Das 1. Kapitel beschreibt Techniken, die lediglich die OSM-Datenbasis nutzen. Im 2. Kapitel werden Techniken beschrieben, die zusätzlich zur OSM Datenbasis weitere Daten verarbeiten und in die Karte integrieren.

## 2. Individuelle Karten, die ausschließlich aus der OSM Datenbasis erzeugt werden

### 2.1. SPLITTER betreffende Hinweise

Spätestens bei der Erzeugung der 2. eigenen Karte sollte man der Übersicht wegen schon beim Splitten des Länder -oder Kontinentfiles sich eine eigene Systematik anlegen und die beim Splitten erzeugten Files entsprechend der eigenen Systematik benennen. Täte man das nicht, hätten alle gesplitteten osm.pbf einen Dateinamen mit 62340000 beginnend. Wir ergänzen deshalb das File <mein Land>splitten.bat um einige Zeilen. Mindestens aber um `--mapid=90000001` .90000001 steht hier exemplarisch für eine 8-stellige Zahl, welcher der Dateiname der ersten Kachel wird. Die Zahl wird automatisch vom SPLITTER hochgezählt. Manchmal scheitert SPLITTER mit dem Fehler sinngemäß 'zu viele nodes im area'. Das kann passieren, falls z.B. eine Großstadt sehr viele Strassen hat, die trotz normalerweise kleiner Fläche zu einem Überlauf führen. Das kann man verhindern, indem man die Option `--max-nodes=1500000` benutzt. 1,5 Mio sind normalerweise ein üblicher Wert, den man ev. kleiner wählen könnte.

Das File <meinLand>splitten.bat könnte dann so aussehen :

E:

```
del E:\Karten\Iceland\9000*.osm.pbf
rem alte Files vorheriger Durchläufe löschen
java -Xmx5500m -jar E:\Karten\splitter-r653\splitter.jar --output-dir=E:\Karten\Iceland
--mapid=90000001 --geonames-file=E:\Karten\cities500.zip --max-nodes=1500000
E:\Karten\Iceland\iceland-latest.osm.pbf
pause
```

Sehr selten kommt es vor, daß mkgmap meldet sinngemäß 'nicht genügend Platz in einem einzelnen File für zu schreibende Elemente'. Es bedeutet, dass eine vom SPLITTER erzeugte Kachel zu groß ist. Abhilfe kann nur die Verkleinerung dieser Kachel schaffen. Theoretisch müßte man den gesamten Splitvorgang mit heruntergesetzter Option `--max-nodes=1000000` neu starten, mit entsprechenden Durchlaufzeiten. Das ist das Standardverfahren bei mittelgrossen Areas. Zum Verständniss der vom SPLITTER erzeugten areas.list nachfolgend eine manuelle Korrekturmethode. Ist zumindest am Anfang nicht wirklich schneller, fördert aber das Verständniss. Wir helfen uns folgendermaßen: nur die beanstandete Kachel wird in 2 Kacheln gesplittet. Die betreffende Kachel wird umbenannt durch vorsetzten von 'X' an den Beginn des Dateinamens, heißt dann z.B X90000026.osm.pbf. Dann öffnen wir das File <name\_>areas.list mit Notepad++ und suchen dort die Zeile die mit 90000026 beginnt. Dahinter stehen vorerst unverständliche Zahlen, die aber ganz leicht zu verstehen sind. Die darunterstehende Zeile ist mit # als Kommentarzeile markiert und stellt die Koordinaten in Dezimalgrad dar. Die unverständlichen Zahlen sind die Koordinaten der Seiten der Kachel, aber in 'GarminUnits'. Es beginnt mit Süden, Westen, Norden, Osten; also im Uhrzeigersinn bei Süd beginnend. Die Umrechnungsformel von Dezimalgrad in 'GarminUnits'(GU) lautet :  $360^\circ \text{Dezimal} = 2 \text{ hoch } 24 \text{ GU}$ . So genau brauchen wir es aber nicht, wir benutzen die Formel  $1 \text{ GU} = 0,000021457 \text{ Dezimalgrad}$ . Selbst diese Rechnung brauchen wir nur in Spezialfällen. Wir halbieren einfach die Kachel. Entweder waagrecht, dann müssen wir die Süd- und Nordkoordinaten anpassen, oder senkrecht, dann eben die West- und Ostkoordinaten. Aus 1 Kachel werden dann 2. Im folgenden Beispiele halbieren wir waagrecht. Um die Sache möglichst einfach zu machen, kopieren wir aus der <name>areas.list die 2 Zeilen der fraglichen Kachel und fügen diese doppelt in eine leeren Datei ein. Wir beabsichtigen die Originalkachel unter ihrem Namen, aber neuen Grenzen zu behalten und in der 2. Kopie vergeben wir einen neuen Namen, am besten eine Zahl, die auf die höchste vergebene folgt. In der MyKartenname.args müssen wir ebenfalls diese neue Kachel hinzufügen. Jetzt ändern wir die Grenzen in der Originalzeile so, dass die Nordgrenze, es ist die 3.Zahlenkolonne etwa auf halber Höhe liegt. Rechnerisch wäre das :  $\text{neuer Nordwert} = \text{Südwert} + (\text{Nordwert} - \text{Südwert}) / 2$  . Der neue Wert muß nur 1

Zusatzbedingung erfüllen, er muß durch 4 ohne Rest teilbar sein. Wie machen den Test mit dem vorerst ermittelten Wert und nehmen dann den am nächsten liegenden ohne Rest durch 4 teilbaren Wert. Diesen Wert benutzen wir in der 2. Kachel als Südwert (1.Zahlenkolonne). Damit geht die Originalkachel nur noch bis zur halben der ursprünglichen Breite und die neue Kachel beginnt dort. Das funktioniert zu 99%. Es gibt aber Gebiete, genannt sei hier der Großraum Kairo, wo rundum nur Wüste ist und im Stadtgebiet sehr dichte Bebauung. Dort kann die einfache Teilung in der Mitte scheitern, dann müßte man intelligenter teilen, mit der oben mitgeteilten Formel eigene GU ermitteln. Schließlich speichern wir die Liste mit den 2 Kacheln als AreasRep.list und starten die mit einer Splitter-rep.bat :

```
java -Xmx5500m -jar E:\Karten\mkgmap\Splitter\splitter-r653\splitter.jar --output-dir=E:\Karten\Kanaren --split-file=E:\Karten\Kanaren\AreasRep.list E:\Karten\Kanaren\X90000026.osm.pbf
```

Dabei wird nur die eine Kachel in 2 Kachel gesplittet, was rasend schnell geht.

Falls man eine Karte für ein nicht standardmäßig angebotenes Gebiet erstellen will, so kann man aus einem größeren Gebiet wie Europa ein kleines wie Pyrenäen ausschneiden. Dazu braucht man ein sogenanntes Poly-file. Dieses kann man sehr leicht mittels des auf der SD Card befindlichen gpx2poly.exe selbst generieren. Dazu wird in Mapsource /oder Basecamp ein Track zusammengeclickt, im Uhrzeigersinn und ohne Überschneidungen. Der letzte Trackpunkt soll 100 m bis 10 km vor dem 1.Trackpunkt enden, also ein offenes Polygon, das speichern wir als name.gpx und öffnen es mit der gpx2poly.exe. Diese erzeugt daraus das gewünschte Poly-file. In Splitter geben wir dann die zusätzliche option : *polygon-file=<Pfad zum polyfile>\meineregion.poly an.*

## 2.2 mkgmap und STYLE

Die Erstellung thematischer Karten erfolgt prinzipiell, indem man nur die Datenelemente der OSM-Datenbank benutzt, die den Zweck am besten erfüllen. Für eine zur KFZ Navigation bestimmte Karte sind z.B. Gebäudedarstellungen nicht hilfreich, für eine Wanderkarte sehr wohl. Letztendlich muß jeder für sich entscheiden, welche Elemente in der Karte sein sollen. Das Hilfsmittel dazu ist der sogenannte STYLE. Der STYLE ist ein Ordner, der mehrere Files enthält, je eines für die Kartenelementtypen Punkt, Linie und Fläche und einige weitere, die jedoch selten einer Bearbeitung bedürfen. Entsprechend heißen die Files *'lines'*, *'points'* und *'polygons'*. mkgmap benutzt ohne weitere Angaben die im Unterordner `\examples\styles\default` stehenden diesbezüglichen Files. Wir werden das ändern und den eigenen STYLE benutzen. Sinnvollerweise erstellt man sich dazu eine Kopie des Ordners `\default` an anderer Stelle und benennt den Kopieordner mit sprechendem Namen, z.B. *MyStreetStyle*. Alle Files sind reine Textfiles und bevorzugt mit dem Editor Notepad++ zu bearbeiten. Bevor wir jetzt beginnen, Änderungen an den *'points'* oder *'lines'* oder *'polygons'* zu tätigen, sollten wir die Software JOSM installiert haben. Zu beziehen unter <https://josm.openstreetmap.de/download/windows/josm-setup.msi>. Ich verzichte bewußt auf eine Kopie auf der micro SD Card, weil die schnell veralten würde, deshalb besser den Download der aktuellen Version vom Link. JOSM werden wir nutzen, um die genaue Schreibweise und das Vorhandensein von 'Tag 's' festzustellen. Dazu laden wir in JOSM eines der gesplitteten 6234000\*.osm.pbf Files. Mit 'Tag' bezeichnet man die Beschreibung genau einer Eigenschaft eines OSM-Elementes. OSM Elemente haben fast immer mehrere Eigenschaften. Beispiel : *highway=service* und *name=Obere Hauptstraße*. Diese Tag 's' gehören zu einer Linie, welche eine Hauszufahrt, Parkplatzzufahrt u.ä. bezeichnet. Der Tag *name=Obere Hauptstraße* ist wohl selbsterklärend. Wobei uns das 'ß' später noch Probleme bereiten kann und im Style behandelt werden sollte. Das Zeichen '#' am Beginn einer Zeile macht diese Zeile zur Kommentarzeile, die nicht ausgewertet wird.

Die einfachste Bearbeitung ist, alle nicht gewünschten Kartenelemente durch Setzen des # an den Beginn der Zeile zu deaktivieren. Bei einer Strassenkarte würde man die Gebäude und viele Landnutzungspolygone deaktivieren, ev. noch in den points Berggipfel, Sehenswürdigkeiten, Brunnen, Biergärten ? Der eigenen Kreativität sind keine Grenzen gesetzt. Nur mit rausstreichen kann man allerhand Dateivolumen der resultierenden Karte sparen. Mit eigenen Ergänzungen wird die Karte noch spezieller. In den als Vorlage genommenen defaultstyle sind bei weiten nicht alle Tags der OSM-Datenbasis ausgewertet. Zum Beispiel könnte man bei Wanderkarten die Skipisten mit einbauen. Dazu die *lines* ergänzen mit :

```
# ski
```

```
piste:type=downhill & piste:difficulty=novice [0x10f08 resolution 22]
```

```
piste:type=downhill & piste:difficulty=easy [0x10f09 resolution 22]
```

```

piste:type=downhill & piste:difficulty=intermediate [0x10f0a resolution 22]
piste:type=downhill & piste:difficulty=advanced [0x10f0a resolution 22]
piste:type=downhill & piste:difficulty=expert [0x10f0b resolution 22]
piste:type=downhill & piste:difficulty=freeride [0x10f0c resolution 22]

```

Da Garmin Probleme mit deutschem 'ß' hat, sollte man das 'ß' nach 'ss' wandeln. Dazu die Zeile

```
addr:street=* {set addr:street='${addr:street|subst:ß=>ss}'}
```

ziemlich am Beginn jedes Files einfügen.

```
name=* {set name='${name|subst:ß=>ss}'}
```

Für Motorradkarten könnte man die Zeiten für Streckensperrungen einarbeiten. Unter *lines* :

```
highway=* & motorcycle:conditional=* {set name '${name} bike=${motorcycle:conditional}' }
```

mit dieser Zeile wird der Strassenname ergänzt durch die im Tag motorcycle:conditional hinterlegten Werte. Oder unterscheiden Sie Parkplätze nach gebührenpflichtig oder nicht gebührenpflichtig. Dazu müssen Sie im points -File den Tag `amenity=vending_machine + vending=parking_tickets` auswerten und ggf. unterschiedliche Symbole zuweisen. Das sind nur einige Beispiele zur individuellen Kartengestaltung. Die Syntax für alle Regeln sind im style-manual.pdf ausführlich beschrieben.

Spätestens jetzt sollte man sich mit der Bedeutung der Zeichengruppen [0x1a resolution 20] vertraut machen. 0x10f4 oder ähnlich steht für die Bezeichnung eines Kartenelementes (Linie, Punkt oder Polygon) wie es Garmin benutzt. Die Bedeutung ist in der Datei GarminTypes.txt erklärt. Es ist wichtig, dass man sich so nah wie möglich bei der Umsetzung der OSM-Elemente in die GarminElemente an diese Vorgaben hält. Beispiel : highway=motorway soll übersetzt werden in [0x01 resolution 15], weil 0x01 laut Liste ein MajorHighway ist. Nur dann kann damit geroutet werden. Und **resolution 15** wird die Autobahn bereits in der Übersichtskarte anzeigen. Siehe dazu auf der sd Card die Datei StrassenIceland.args, Zeile 14 : `overview-levels=5:17,6:15,7:12` . resolution 15 ist hier aufgeführt.

Würde man fälschlicherweise das mit [0x18 resolution 22] übersetzten, so würde daraus ein Fluss, der nur beim reinzoomen angezeigt wird. Weil resolution 18 nicht in der Definition für die overview-levels =... enthalten ist. Betreffs Strassen muß streng darauf geachtet werden, das dafür nur routingfähige GarminTypen benutzt werden. Bei prinzipiell nicht routingfähigen Kartenlinien wie Stromleitungen, Pipelines machen falsche Zuordnungen weniger aus.

Viele STYLE Regeln ergeben nur Sinn, falls man in den mkgmap-Optionen zugehörige Werte benutzt. Deshalb nachfolgend Erklärungen die Optionen betreffend.

Bereits im Kleinen Kochbuch habe ich einige Optionen in den KanarenTemplate.args eingebaut. Da ein template.args jedesmal vom splitter erzeugt wird, benutzen wir es, um alle mkgmap-optionen dort zusätzlich hinein zu schreiben. Und zwar ganz am Anfang und bevor die 1.Zeile mit input-file :...kommt. Die Option `family-id=123...` weist der Karte eben diese Ganzzahl zu, der Wertebereich reicht von 1 bis 65534. Derzeit von Garmin verwendete Werte liegen alle < 3000. Diese family-id ist wichtig zur Darstellung auf einem PC und GPS. Jede Karte muß eine nur 1x vorkommende unterschiedliche family-id besitzen. Das gilt auch für Karten auf dem GPS. Sollten 2 Karten die selbe family-id haben, würde nur 1 Karte davon in Basecamp/Mapsource bzw. dem GPS dargestellt, die 2. würde ignoriert. `product-id=1` hat den selben Wertebereich. Üblicherweise wird nur der Wert =1 benutzt. Andere Werte sind möglich, aber nicht erforderlich. Sollten Sie die Absicht haben, mehrere Karten zu erstellen, ist es empfehlenswert, sich die bereits verwendeten familiy-id privat zu notieren, um den Überblick zu behalten. Mein Tool img2ms kann sowohl die family-id's (dort FID genannt) aller installierten Karten auslesen als auch nachtraglich andern. `family-name= StrassenBayern` und `series-name=StrassenBayern` sind selbsterklarend. `add-pois-to-areas` ist immer sinnvoll. Hufig sind z.B. Parkplatze als Polygon in der Datenbank, Sie mochten aber in der Karte einen POI "Parkplatz", dann bewirkt diese Option, da ein POI `emenity=parking` ggf. mit weiteren Tag's (Gebuhrenpflichtig ?) der Datenbank hinzugefugt wird, der dann unter pois ausgewertet werden kann. `show-profiles=1` mu bei Topografischen Karten gesetzt werden, um enthaltene Hohendaten (dazu im nachsten Kapitel) angezeigt zu bekommen. `output-dir=<beliebigen vollstandigen Pfad>` ist selbsterklarend.

`index` erzeugt einen globalen Adresssuchindex, vergroert aber die Dateigroe der Karten erheblich.

`poi-address` erzeugt den Suchindex fur POIS

`route` macht das, was der Name schon sagt

*style-file=E:\Karten\MyStreetStyle* bewirkt, dass der eigene STYLE verwendet wird,  
*order-by-decreasing-area* ist immer zu empfehlen, verbessert die Darstellung der Polygone  
*ignore-fixme-values* ist immer zu empfehlen, vermeidet die Auswertung von Tags, die unsicher sind  
*location-autofill=is\_in*  
*process-exits*

*process-destination* empfehlenswert für Strassennavigation, ergänzt die Anzeige von Abbiegehinweisen  
*code-page=1252* Darstellung in lateinischen Buchstaben, für kyrillisch oder osteuropäische Karten sind andere *code-page* zu empfehlen u.U.auch UTF-8  
*levels=0:24,1:22,2:20,3:18*  
*overview-levels=5:17,6:15,7:12* Die hier eingetragenen Werte sollten mit den im STYLE gesetzten *resolution xy* übereinstimmen. Wenn im STYLE eine *resolution 17* oder *12* für eine Element (z. B. *highway=motorway*, =deutsche Autobahn ) gesetzt wird, dann erscheint dieses in der Übersichtskarte.

Nachdem wir das vom Splitter erzeugte *template.args* mit unseren eigenen Eintragungen ergänzt haben, speichern wir es unter neuem Namen z.B. *StrassenIceland.args*, ansonsten würde es beim nächsten Splitterlauf überschrieben. Die Beispiele zu dieser 2. eigenen Karte sind auf der sd Card unter dem Ordner *E:\Karten\Iceland* gespeichert.

Alle Optionen sind in der Hilfe bzw. auch auf meiner micro-SD card zu finden. *mkgmap* setzt während eines Durchlaufes auch sogenannte *#interne tags#*, deren man sich ebenfalls bei der Definition des eigenen STYLES bedienen kann. Die internenTags sind nur auf meiner sd Card.

Da wir jetzt alle *mkgmap*-Optionen bereits in der z.B *StrassenIceland.args* gespeichert haben gestaltet sich der Start mit einer *StrassenIceland-erzeugen.bat* sehr einfach :

```
cd E:\Karten\mkgmap-r4907
java -XX:-UseGCOverheadLimit -jar mkgmap.jar --read-config=E:\Karten\Iceland\StrassenIceland.args
E:\OSMStra.typ
pause
```

Um diese Karte in Basecamp zu sehen, verschieben wir die erzeugte *StrassenIceland.gmap* nach *C:\ProgramData\Garmin\Maps\StrassenIceland.gmap*

## 2.2 mkgmap und Typfiles

in der letzten gezeigten *StrassenIceland-erzeugen.bat* ist bereits ein *Typfile* benutzt worden.

Für Strassenkarten, die eng an die Garminkonventionen anlehnen, ist ein *Typfile* nicht unbedingt erforderlich. Garmin selbst verwendet *Typfiles* für Strassenkarten nur zur Darstellung von POI. Dennoch ist es allgemein Brauch geworden, einer Karte ein wenigstens minimales *Typfile* mitzugeben. **Ein Typfile ändert lediglich die Optik**, aber keine sonstigen Eigenschaften. Insbesondere ist es **falsch zu erwarten**, dass ein in der Optik auf Strasse/ Autobahn/Wanderweg getrimmtes Kartenelement, welches aber nicht mittels *0x10* usw. als routingfähiges Element erzeugt wurde, durch ein *Typfiles*, welches Strassen vorgibt, routingfähig zu machen. Jedes Element behält seine (Routing-) Eigenschaften, die es mittels *0x10* usw. mitbekommen hat. *mkgmap* bringt einen eigenen *Typcompiler* mit. Darauf wird hier nicht eingegangen, weil es eine bessere Variante mit schöner grafischer Benutzeroberfläche gibt : *Typviewer V4.6.5* von user *michel40* : <https://sites.google.com/site/sherco40/>.

Bei der Bearbeitung eines *Typfiles* muß man streng darauf achten, die *family-id* des beabsichtigten Kartensets auch im *Typfile* zu benutzen. Anderenfalls würde die geänderte Optik nicht wirken. Mit dem *Typviewer*, wie oben verlinkt, ist das selbsterklärend möglich. Die Feinheiten der Programmierung sind im *Usermanual* von *cgpsmapper* hinreichend erklärt. Das Hinzufügen des *Typfiles* erfolgt, indem man in der *<kartenerzeugen>.bat* den Pfad zum *Typfiles* hinzufügt. Im letzten Beispiel ist dies bereits erfolgt. *mkgmap* ist so freundlich und passt ggf.falsche FID des *Typfiles* an. Wenn man es also wie oben beschrieben dem *Mapset* bereits mit der *.bat* zuordnet, ist die FID gleichgültig.

## 3. Karten die weitere Quelldaten beinhalten

### 3.1 Karten mit Digitalem Höhenmodell

Der nächste, noch relativ einfache Schritt zur Verbesserung der Karten ist die Integration eines Digitalen Höhenmodells. Dazu ist noch keine zusätzliche Software erforderlich, da *mkgmap* ausreicht. Beschafft

werden muß aber die Datengrundlage, in diesem Fall sind das sogenannte <name>.HGT -Files. Es handelt sich um rein binäres Format. Jedes File beinhaltet genau den Bereich von 1° Länge x 1° Breite. Der Dateinamen ist standardisiert, er bezeichnet die linke untere Ecke jedes 1° x 1° Quadranten. N40W125.hgt steht demnach für den Bereich von N 40 ° bis N 41 ° und W 125° bis W 124 °. Diese Dateien werden sowohl von Landesämtern als auch von Internationalen und Nationalen Organisationen bereitgestellt. Dabei gibt es teilweise beachtliche Qualitätsunterschiede. Landesämter geben meist höhere Genauigkeiten als z.B. die beliebten der NASA. Ein genereller Qualitätsunterschied liegt in der Punktweite, mit der die Daten erfasst wurden. Bei der Nasa ist das beste 1 ArcSekunde Punktabstand. Andere Quellen liegen deutlich enger. Für den Bereich von 60 ° südlicher Breite bis 60 ° nördlicher Breite bietet die Nasa aber bequemen Download an. Nördlich 60° Breite wird man bei Jonathan de Ferranti fündig. Einen Einstieg bietet die website [vterrain.org/Elevation/SRTM/](http://vterrain.org/Elevation/SRTM/). Ganz neu bietet der user sony auf der website <https://sonny.4lima.de> aufbereitete Daten an. Die website hat bereits vorgfertigte Downloadbereiche, die umfangreiche Suchaktionen erleichtern. Man muss sich also zunächst eine Sammlung von <Koordinaten>.hgt Files anlegen. Am besten auf einem separaten Datenträger. Wenn das geschehen ist, nimmt man folgende zusätzliche Optionen(=Zeilen) in das <name>.args File auf :

```
#vollständiger Pfad zum Ordner mit den Koordinaten.hgt-files
dem=E:\Karten\ISECHGT\
dem-dists=3312,13248,26512,53024
```

Für das Beispiel 2-Iceland sind die meisten HGT-Files auf der sd Card enthalten. Außerdem ein Linkverzeichnis zum NASA ftp-server.

### 3.2 Karten mit zusätzlichen Contourlines (Höhenlinien)

Für diese Karten müssen die Höhenlinien zunächst mit weiterer u.U. Bezahlt Software erstellt werden. Die gängigste Software dazu ist GlobalMapper von [www.blumarblegeo.com](http://www.blumarblegeo.com). An freier Software wäre zunennen IDL-VIRTUAL -MACHINE, gepaart mit DEM2TOPO.sav. Habe ich aber seit dem Umstieg auf GlobalMapper nicht mehr benutzt. Funktioniert hat es schon, aber alles manuell ohne batch macht es langwierig. IDL-VIRTUAL MACHINE ist immer noch frei verfügbar, und das dazu benötigte DEM2TOPO.sav kann von mir bereitgestellt werden. GlobalMapper kann aus den <Koordinaten>.hgt Files Höhenlinien erzeugen. Als Eingangsformat ist auch geeignet das häufig anzutreffende GeoTiff-Format. GM kann auch aus GeoTiff hgt-files erzeugen. Besonders durch die Möglichkeit der Batchverarbeitung ist GM hervorragend geeignet. Zur Erzeugung wählt man in GM Analysis--> generate contours, kann dann Rahmenwerte für Abstand vorgeben, Kachelung. Und Exportiert die erzeugten Contourlines im Format <polish>.mp als <Koordinaten>.mp Files.

Man sollte konsequent die GM-Script-Language nutzen. Ein Beispiel für die Erzeugung von Contourlines aus den HGT für den Bereich der Alpen mittels GM-Script ist auf der sd Card. Dieses Beispiel benutzt nicht die Möglichkeiten von Schleifendurchläufen, was aber das händische Schreiben vereinfachen würde. Ich erzeuge es momentan mittels VB6-IDE automatisiert. Hier ein Ausschnitt :

```
GLOBAL_MAPPER_SCRIPT VERSION=1:00
UNLOAD ALL
IMPORT FILENAME="L:\HGT\Alps\N45E005.hgt"
GENERATE_CONTOURS ELEV_UNITS=METERS INTERVAL=25 GEN_SPOT_ELEVATIONS=NO
EXPORT_VECTOR FILENAME="L:\HGT\Alps\N45E005.mp" TYPE=POLISH_MP MAP_NAME=ContourLines
TEMPLATE_FILENAME=D:\mkgmap\SwissSued.mp
UNLOAD ALL
IMPORT FILENAME="L:\HGT\Alps\N45E006.hgt"
GENERATE_CONTOURS ELEV_UNITS=METERS INTERVAL=25 GEN_SPOT_ELEVATIONS=NO
EXPORT_VECTOR FILENAME="L:\HGT\Alps\N45E006.mp" TYPE=POLISH_MP MAP_NAME=ContourLines
TEMPLATE_FILENAME=D:\mkgmap\SwissSued.mp
UNLOAD ALL
```

.....  
.....  
.....

Das im GlobalMapper.script erwähnte Template\_Filename =D:\mkgmap\SwissSued.mp ist ein Musterfile, aus dem GM die Einträge im Header, wie TRE, Transparent und die Angaben zu Layer/resolution

übernimmt. Dazu sollte man im Usermaual des cgpsmapper nachschlagen.

Die von GM exportierten <Koordinaten>.mp Files kann mkgmap genauso wie osm.pbf in Garmin Kacheln <12345678>.img wandeln. Man gibt dazu in einer .bat für mkgmap dem Pfad zu einer `contourlines.args` an. Diese beginnt dann wie folgt :

```
code-page=1252
transparent
output-dir=E:\Karten\Contourlines\IMG
draw-priority=30
```

```
mapname=40005000
description=ContourLines 40005000
input-file=E:\Karten\Contourlines\N45E005.mp
```

```
mapname=40005001
description=ContourLines 40005001
input-file=E:\Karten\Contourlines\N45E006.mp
```

### 3.2.1 wie kommen die Contourlines in die Karte ?

der oben bezeichnete Weg erstellte 2 Sätze von <12345678>.img, nämlich die eigentlichen OSM-Kacheln, deren Dateinamen mit 90000\* beginnt und einen Satz reine Contourlines-Kacheln, deren Dateinamen mit 4000\* beginnt. Die Contourline-Kacheln sind transparent erzeugt und mit höherer DrawPriority als die OSM-Kacheln, werden deshalb über die OSM Kacheln gerendert. Ich vermeide den anderen Weg, nämlich die Contourlines zum untrennbaren Bestandteil der Karten zu machen, was man mit OSMIUM könnte, aus gutem Grund : so schön die Contourlines auf einem großen PC-Bildschirm auch aussehen, auf dem kleinen GPS sind die eher störend, wenn man in steilen Gegenden nichts mehr von der Topografie sieht, weil die viel zu eng den Bildschirm unlesbar machen. Meine Methode gestattet es, wahlweise die Contourlines mit zum GPS zu übertragen oder eben nicht, je nach Belieben kann man die auf den meistens GPS auch als aktiv stellen oder deaktivieren. Nachdem also die Contourlines als separate und immer wiederverwendbare Kacheln vorliegen, ändern wir nur geringfügig die <MeinLand>\_erzeugen.bat wie folgt :

```
cd E:\Karten\mkgmap-r4907
```

```
java -XX:-UseGCOverheadLimit -jar mkgmap.jar --read-config=E:\Karten\Iceland\StrassenIceland.args
E:\Karten\Contourlines\400*.img E:\OSMStra.typ
```

```
pause
```

Achtung, die rote Schrift ist 1 Zeile ohne Zeilenumbrüche !

### 3.3 Karten mit zusätzlichen Inhalten

GM kann fast alle denkbaren Aufgaben der Konvertierung von Vektordaten fast beliebigen Ursprungs in OSM-Daten erledigen. Prinzipell läuft das dann so, dass man aus beschafften Daten z.B. CSV-Tabellen mit Hauskoordinaten und Adressangaben der Landesvermessungsverwaltungen diese mit GM bearbeitet und exportiert als OSM-Datei. Die Vereinigung dieser aus GM exportierten OSM-Daten mit der originären OSM-Datenbank darf aus prinzipiellen Gründen nur auf dem Lokal PC erfolgen. Es ist strikt verboten und würde zu einem Chaos führen, falls man auf die Idee käme, solche von GM erzeugten Daten in die offizielle Datenbank hochzuladen. Die Erklärung dazu ist : in der offiziellen OSM-Datenbank hat jeder node, way und relation eine durchnummerierte ID (IdentityNumber). Diese ID ist eine positive Ganzzahl. Von GM erzeugte node, way, relation haben **negative** Ganzzahlen, daran würde die OSM Datenbank erkennen, das es sich um neue unbekannte node usw. handelt. Beim Hochladen würde sich eine hohe Anzahl Dopplungen ergeben, deshalb sind solche Batch Uploades nur nach Rücksprache und Diskussion mit den Betreibern der OSM Datenbank zulässig.

Der privaten Verwendung steht aber nichts entgegen. Im Gegenteil, solche Karten sind ein Alleinstellungsmerkmal und qualitativ den Standardkarten überlegen. Jetzt muß nur noch die Vereinigung der originären OSM Daten mit den selbst erzeugten bewerkstelligt werden. Das Tool der Wahl ist die commandline Software *OSMIUM*.

#### 3.3.1 OSMIUM

OSMIUM ist auf Github verfügbar, leider nicht als kompilierte Version für Windows 64 bit. Der offizielle

Weg führt über Anaconda3 -Installation, diese enthält eine unter W64 bit lauffähige OSMIUM.exe im Pfad C:\ProgramData\Anaconda3\Library\bin. Mit Anaconda3 wird auch eine Anaconda-eigene shell installiert. Mit dieser shell kan man bei Bedarf neue oder aktualisierte Pakete (z.B. die aktuelle OSMIUM V15) nachinstallieren. Die aktuelle OSMIUM v15 für Anaconda kann man downloaden von [https://anaconda.org/conda-forge/osmium-tool/files/win-64/osmium-tool-1.15.0-h8558f88\\_0.conda](https://anaconda.org/conda-forge/osmium-tool/files/win-64/osmium-tool-1.15.0-h8558f88_0.conda) . Das package ist auch auf sd Card. In der conda shell kann man es mit *conda install -c conda-forge <pfadzumPackage>* nachinstallieren/aktualisieren. Ich habe diese Osmium.exe zusammen mit den notwendigen Dll in einem Ordner \OSMIUM\Library\bin\ auf der sd Card kopiert und konnte damit arbeiten.

### 3.3.2 Vereinigen

Die Vereinigung läuft in 3 Schritten :

1. stellen wir fest, welche ID's in der originalen latest.osm.pbf verwendet werden. Diese höchsten Werte notieren wir.

```
osmium fileinfo -e D:\Europa\europa-latest.osm.pbf
```

2. wir sortieren die eigene.osm

```
osmium sort -O -o D:\Europa\Sorted-eigene.osm.pbf D:\Europa\europa-latest.osm.pbf
```

3.wir renumerieren die sortierte eigene.osm.pbf so, dass die Nummerierung mit deutlich höheren ID's beginnt, als bei der originalen latest.osm.pbf verwendet werden. Die niedrigste ID sollte ruhig 100 Mio höher liegen als die höchste der originalen.osm.pbf.

```
osmium renumber -O --start-id 12000000000,2100000000,16400000 -o D:\Europa\renumbered-eigene.osm.pbf D:\Europa\Sorted-eigene.osm.pbf
```

4.wir mergen beide

```
osmium merge -O -o D:\Europa\merged-all.osm.pbf D:\Europa\europa-latest.osm.pbf D:\Europa\renumbered-eigene.osm.pbf
```

#rot ist je 1 Zeile ohne Zeilenumbruch !

Nach diesem Schritt erfolgt die Weiterverarbeitung wie gewohnt mit splitter und mkgmap.

## 4.0 Beispiele für Karten mit zusätzlichem Inhalt

### 4.1 Spanien, Islas Canarias, Adressdaten

Vom spanischen Geografischem Institut, [www. Centrodascargas.cnig.es](http://www.Centrodascargas.cnig.es) sind shp\_files verfügbar je Provinz, hier Santa Cruz de Tenerife, in der Kategorie 'Mapas vectoriales y Bases Cartograficas y Topograficas; Information geografica tematica : CartoCiudad. (auf der sd Card enthalten als Cartociudad\_Callejero\_tenerife.zip). Die shp-files enthalten die georeferenzierten Adressen mit Strassennamen, Hausnummer usw. Man öffnet diese zip mit GlobalMapper und verschafft sich einen Überblick, welche Attribute verwendet werden. Viele Attribute sind überflüssig und werden im 2. Schritt gelöscht, die brauchbaren Attribute werden belassen und im STYLE ausgewertet. Zusätzlich wird ein Attribut, im Sprachgebrauch der OSM-Karten als tag bezeichnet, hinzugefügt, welches die Herkunft der Daten markiert. Unser Ziel ist es, diese Adressen in die Karte aufzunehmen. Im Beispiel sind folgende Attribute verwendbar, deren Namen wir belassen und im STYLE wie folgt auswerten : TIPO\_VIAL und NOMBRE\_VIA====> addr: street='\$(TIPO\_VIAL) \$(NOMBRE\_VIA)'; NUMERO ====> addr:housenumber='\$(NUMERO)'; POBLACION ====> addr:city='\$(POBLACION)'; COD\_POSTAL====> addr: postcode='\$(COD\_POSTAL)'. Damit haben wir PLZ, Stadt, Strassennamen und Hausnummer. Alle anderen Attribute werden gelöscht mittels eines GlobalMapper.gms Files, das schreiben wir mit Editor selbst wie folgt :

```
GLOBAL_MAPPER_SCRIPT VERSION=1:0
```

```
EDIT_VECTOR_ATTR_TO_DELETE=EXTENSION ATTR_TO_DELETE=DGC_VIA ATTR_TO_DELETE=CODIGO  
ATTR_TO_DELETE=ID_POB ATTR_TO_DELETE=ID_PORPK ATTR_TO_DELETE=ID_PORPKD  
ATTR_TO_DELETE=SENTIDOPK ATTR_TO_DELETE=SENTIDOPKD ATTR_TO_DELETE=FUENTE  
ATTR_TO_DELETE=FUENTED ATTR_TO_DELETE=FECHA_ALTA ATTR_TO_DELETE=ID_TRAMO  
ATTR_TO_DELETE=ID_VIAL ATTR_TO_DELETE=TIPOPORPKD ATTR_VAL=source=geoSpain
```

#Achtung, im pdf lassen sich die Zeilenumbrüche nicht vermeiden, Es sind nur 2 Zeilen ! Jede Zeile hat 1 Farbe!

In GM lassen wir das script durchlaufen und haben jetzt nur noch die benötigten Daten, diese exportieren wir als ==> ExportVector ==> OSM -XML File. Das exportierte <tenerifa>.osm wird wie oben

beschrieben mit *OSMIUM sort ; remumber* und *merge* mit den latest-osm.pbf vereinigt.

## 4.2. Spanien, Gebäudepolygone

Unter der Kategorie BTN kann man ebenfalls Landnutzungspolygone, Gebäudeumrisse, auch gegliedert nach Wohnbebauung, Industrie usw. beziehen. Die Einarbeitung erfolgt nach dem Schema wie unter 4.1. angegeben.

## 4.3 Adressen deutscher Bundesländer, Schweiz und Österreich

Einige Bundesländer wie Sachsen, Thüringen, Hessen, NRW, Bremen, Berlin Brandenburg stellen die georeferenzierten Adressangaben kostenfrei unter Lizenz der Quellenangabe zur Verfügung. Zu beziehen bei den jeweiligen Landesvermessungsverwaltungen. In der Regel sind das je Bundesland 2 CSV-Dateien. Eine davon, die SCHLÜSSEL.CSV, enthält nur numerische Schlüsselwerte für Orte, Regierungsbezirk, Ortsteil und deren Klarnamen. Die 2. CSV- Datei enthält die Koordinaten, numerische Schlüssel für die 1. Datei und die Strassennamen, Hausnummern, ggf.auch PLZ. Um das verwerten zu können, müssen zuerst die Ortsnamen im Klartext an Hand der numerischen Schlüssel aus der 2. CSV- Datei ausgelesen und zusammengeführt werden. Wegen der hohen Anzahl Datensätzen (Millionen..) ist das nur mittels Datenbank Software möglich. Ich habe für meine private Nutzung ein unter VB6 -IDE laufendes Programm geschrieben, was dies erledigt. Aus Copyrightgründen kann ich nur den Quelltext auf Wunsch zur Verfügung stellen. Zur Weitergabe der Chilkat-csv.dll und der Demoversion von VB6-IDE auf CD habe ich keine Lizenz. Zumindest ist aber kostenlos die Chilkat-csv.dll bei Chilkat beziehbar. Wer noch eine VB6-IDE laufen hat, kann mit dem Quelltext das erledigen. Dem Buch #VB6# von Herbert Spona lag eine solche VB6-Demo IDE bei. Das funktioniert auch mit den csv-Daten der Schweiz und Österreichs, wenngleich diese etwas anders konfiguriert sind. Grundkenntnisse und etwas Erfahrung mit VB6 Code sollte vorhanden sein. Die aufbereiteten Adressen-neu.csv kann von GM gelesen werden und exportiert als Adressen.osm, weitere Verarbeitung wie unter 4.1 mit OSMIUM.

## 5. Ausblick

Laut EU-Verordnung sind alle Mitgliedsländer angehalten die Hauskoordinaten und Polygone kostenfrei zur Verfügung zu stellen. Die Umsetzung ist nicht vollständig erfolgt. Für Schleswig-Holstein soll das erst ab 2024 erfolgen. Bayern hat nur die Polygone, aber keine Hauskoordinaten bereitgestellt. Schweiz und Österreich haben diese bereitgestellt. Portugal, Italien habe es umgesetzt, aber in so kleinräumigen Maßstab, dass die Zusammenführung zum Komplettland für mich wegen des Aufwandes nicht in Betracht kommt. Andere Länder habe ich bisher nicht untersucht.

Für Hinweise auf Downloadquellen anderer europäischer Länder bin ich dankbar.

Fazit : durch die Einarbeitung speziell der Hauskoordinaten als Adressen wird ein Schwachpunkt der OSM-Datenbasis, nämlich fehlende Adressdaten, beseitigt und solche Karten haben bezüglich der Adressuche Profiniveau. Eine weitere Aufwertung ist durch die Einarbeitung der Landnutzungspolygone möglich.

Thomas Morgenstern, April 2023