BuildDEMFile

BuildDEMFile[1] ist in erster Linie ein Demo-Programm um zu zeigen, dass man DEM-Daten für Garmin-Karten selbst erzeugen kann. Dafür werden frei verfügbare HGT-Dateien[2] verwendet. Daraus erzeugt das Programm DEM-Dateien, die in eine fertige Garmin-Karte integriert werden müssen.

Technischer Hintergrund

Garmin-Karten bestehen aus einzelnen Kartenkacheln. Die Anzahl der geografischen Objekte je Kachel ist offensichtlich begrenzt. Deshalb kann die geografische Größe einer Kachel sehr unterschiedlich sein. Sind z.B. nur wenige Straßen und sonstige Objekte vorhanden, ist die Kachel im allgemeinen sehr groß. Die Aufteilung der OSM-Daten auf einzelne Kacheln erfolgt z.B. mit dem Programm splitter[3]. Die Größe und Position der Kartenkacheln wird bei jeder Veränderung in den OSM-Daten natürlich etwas unterschiedlich sein.
Im nächsten Schritt werden die OSM-Daten jeder einzelnen Kachel mit mkgmap[4] verarbeitet. I.A. entstehen für jede Kachel jeweils eine TRE-, LBL-, RGN-, NET- und NOD-Datei mit dem gleichen Basisnamen. Ohne an dieser Stelle weiter auf deren Inhalt und deren Bedeutung einzugehen: die TRE-Datei enthält u.a. die exakte Position und Größe der Kartenkachel.
Bei Originalkarten von Garmin findet man manchmal eine Verzeichnisstruktur, in der diese Dateien direkt abgelegt sind. Mkgmap fasst diese „Sub-Dateien“ je Kartenkachel jeweils in einer IMG-Datei zusammen. Z.B. enthält eine Datei 70210016.IMG dann die Subdateien 70210016.TRE, 70210016.LBL usw.. Garmin verwendet an dieser Stelle manchmal auch GMT-Dateien, deren Struktur aber nicht hinreichend genug aufgeklärt ist.
Bei einer Karte für ein GPS-Gerät sind alle Subdateien in einer einzigen IMG-Datei zusammengefasst.
Es gibt 2 wesentliche Unterschiede zwischen den Karten für Mapsource/Basecamp und für ein GPS-Gerät. Zum einen existiert für die PC-Variante i.A. eine zusätzliche Kartenkachel für das gesamte Kartengebiet. Sie kann natürlich nur relativ wenige Daten enthalten. Diese Kachel wird als Übersichtskarte verwendet. Für das GPS-Gerät existiert eine solche Kachel nicht. Zum anderen wird für den PC eine TDB-Datei verwendet. In ihr ist i.W. eine Auflistung aller Subdateien und deren Dateilänge enthalten. Natürlich sind das redundante Informationen und für das GPS-Gerät wird diese Datei auch nicht benötigt.
Ein folgenschweres technisches Problem gibt es bei den IMG-Dateien: sie können nicht um zusätzliche Subdateien erweitert werden! Will man eine Subdatei hinzufügen, müssen die vorhanden Subdateien extrahiert werden und anschließend mit den zusätzlichen Dateien wieder zu einer neuen IMG-Datei zusammengefügt werden.

Prinzipielle Vorgehensweise

Aus dem bisher Beschriebenen folgt prinzipiell folgende Vorgehensweise:

1. Aus einer mit mkgmap fertig erzeugte Karte müssen zunächst die Subdateien extrahiert werden.
2. Mit den geografischen Angaben der TRE-Dateien wird jeweils mit BuildDEMFile und den HGT-Dateien eine maßgeschneiderte DEM-Datei erzeugt.
3. Für den PC werden die jeweils zusammengehörenden Subdateien wieder in IMG-Dateien zusammengefasst **oder** für ein GPS-Gerät werden alle Subdateien und sonstigen Dateien in eine einzige IMG-Datei zusammengefasst.
4. Für den PC wird noch eine neue TDB-Datei erzeugt.

1 https://github.com/FSofTlpz/Garmin-DEM-Build/blob/master/BuildDEMFile/bin/BuildDEMFile.exe
2 https://dds.cr.usgs.gov/srtm/version2_1/SRTM3/
http://www.viewfinderpanoramas.org/
3 http://www.mkgmap.org.uk/download/splitter.html
4 http://www.mkgmap.org.uk/download/mkgmap.html

Vorgehensweise als Beispiel

Im Folgenden wird meine Lösung für die 4 oben genannten Schritte beschrieben. Das heißt nicht, dass es nicht auch andere Möglichkeiten gibt!
Insbesondere für das Extrahieren der Subdateien und deren späteres Zusammenfügen, aber auch die Gewinnung der nötigen Informationen verwende ich mein experimentelles Programm gmtool[5].
Und noch ein Hinweis: Ich kann natürlich keine Garantie für die korrekte Funktion der Programme übernehmen. Es ist immer eine gute Idee, vorher eine Sicherungskopie anzulegen.

Zunächst die Lösung für das GPS-Gerät (sie ist deutlich einfacher). Die folgenden Kommandozeilen-Befehle werden in Textdatei, z.B. builddem4gps.cmd eingetragen und je nach Wunsch angepasst. Dann wird diese neu erzeugte Kommandodatei gestartet.

```
set SRCFILE=old.img
set DSTFILE=new.img
set DESCRIPTION=CzechR, 8.11.2017
set DLON=--dlon=0.00027761 --dlon=0.00049 --dlon=0.00075 --dlon=0.00106 --dlon=0.0017 --dlon=0.0025
set OPTIONS=--usedummydata
set HGTPATH=%OSM_DATA%\srtm_zip
set TMPSPLIT=~tmp

rmdir /S /Q "%TMPSPLIT%"
gmtool -i "%SRCFILE%" --split -o "%TMPSPLIT%"
for /d %%S in (%TMPSPLIT%\*) do move "%%S\*" "%TMPSPLIT%" && rmdir "%%S"
for %%S in (%TMPSPLIT%\*.tre) do BuildDEMFile --dem="%TMPSPLIT%\%%~nS.DEM" -tre="%%S"
                                              --hgtpath="%HGTPATH%" %OPTIONS% --overwrite %DLON%
gmtool -i "%TMPSPLIT%\*.*" --join=device -o %DSTFILE% --description="%DESCRIPTION%" -O
```

Erklärung
Zunächst werden eine Reihe von Umgebungsvariablen gesetzt. Damit ist die Wiederverwendung und Anpassung der Datei einfacher.
SRCFILE ist natürlich die von mkgmap erzeugte Datei.
DSTFILE ist der Name der neuen Datei.
DESCRIPTION ist die Kartenbeschreibung, die das GPS anzeigt (zumindest mein Oregon 600).
DLON ist der Abstand der Höhenangaben (in Grad) für die DEM-Datei. Kleinere Abstände führen zu detailgenaueren Karten aber auch zu größeren Datenmengen. Außerdem ist zu bedenken, das die HGT-Daten schon keine besonders hohe Auflösung haben. Es sollte für jeden Maplevel eine Angabe erfolgen, sonst fehlt bei einigen Zoomstufen die Anzeige der DEM-Date (Maplevel sind im verwendeten mkgmap sytle in der Datei options z.B. so definiert: levels=0:24, 1:23, 2:22, 3:21, 4:20, 5:19).
HGTPATH ist natürlich das Verzeichnis, dass die HGT-Dateien enthält.
TMPSPLIT ist ein Verzeichnis dass nur temporär für die Subdateien verwendet wird.
Die nächsten beiden Befehle legen ein neues temporäres Verzeichnis an und erzeugen darin, leider in eigenen Unterverzeichnissen, alle Subdateien. Die folgende for-Schleife schiebt alle Subdateien direkt in das temporäre Verzeichnis und löscht die unnötigen Unterverzeichnisse. Die nächste for-Schleife erzeugt zu jeder TRE-Datei die DEM-Datei (Achtung: die 2 Zeilen müssen in **einer** Zeile stehen). Der letzte Befehl packt alle Dateien wieder zusammen und setzt die Kartenbeschreibung.

5 https://github.com/FSofTlpz/GmTool/blob/master/bin/Release/GmTool.exe
https://github.com/FSofTlpz/GmTool/blob/master/bin/Release/GarminCore.dll

Für eine PC-Karte ist das Verfahren deutlich komplizierter, weil hier eine ganze Reih zusätzlicher Informationen benötigt werden. Deshalb verwende ich im 1. Schritt ein Perl[6]-Script, dass diese Informationen „einsammelt“ und eine Kommando-Datei builddemcmd.cmd erzeugt. Diese Kommando-Datei wird dann im 2. Schritt gestartet.
Im Kartenverzeichnis, dass z.B. so aussieht

```
34.018.304 70210001.IMG
12.512.768 70210002.IMG
...
 9.031.168 70210053.IMG
    33.863 fsoft3.TYP
   232.960 osmmap.IMG
       650 osmmap.mdx
     9.139 osmmap.tdb
       379 osmmap_license.txt
12.337.152 osmmap_mdr.img
```

wird das Script gestartet.

Einige Variablen im Bereich CONFIG müssen vorher angepasst werden.

```
# create a command file (for windows) to create and insert the DEM-data
#
# gmtool and BuildDEMFile are necessary

use strict;

# ---------- CONFIG ----------
my $ovmap      = 'osmmap.img';                  # name of overview-IMG
my $tdb        = 'osmmap.tdb';                  # name of TDB
my $hgtpath    = 'd:\\osmdata\\srtm_zip';       # path to HGT's
my @dlon       = (                              # point distance in degree for every maplevel
                  0.00027761,
                  0.00049,
                  0.00075,
                  0.00106,
                  0.0017,
                  0.0025,
                 );
my @dlonovmap  = (                              # point distance in degree for overview map
                  0.01851797,
                 );

# ----------------------------------------------------------------------------

my $lastcolstd = 0;                   # with 1 will be the rightmost subtile the standard width 64
my $hgtoutput  = 0;                             # with 1 create textdata from HGT
my $usedummydata = 1;                           # use dummy data when HGT not exist
my $destdir    = "~dem";                        # destination for new IMG's and TDB
my $splitdir   = '~dem2';                       # temp. directory
my $cmdfile    = 'builddemcmd.cmd';             # name of the new command file

# ----------------------------------------------------------------------------
# read the names of all IMG's, but not MDR-IMG
opendir(my $h, './') or die $!;
my @imgfiles = grep { /\.img$/i && !/mdr\.img$/i && -f "$_" } readdir($h);
closedir($h);

open(my $cmd, ">$cmdfile") or die $!;
print $cmd "\@echo off\n";
print $cmd "mkdir $destdir\n";
print $cmd "mkdir $destdir\\hgtoutput\n";
print $cmd "\n";

foreach my $img (@imgfiles) {
   print STDERR "$img ...\n";
   Do4Img(substr($img, 0, length($img) - 4), $splitdir, $destdir, $hgtpath, $usedummydata,
          $lastcolstd, $hgtoutput, uc($img) eq uc($ovmap) ? @dlonovmap : @dlon);
```

6 https://www.perl.org/get.html

```
}

print STDERR "$tdb ...\n";
my (  $codepage,
      $famid,
      $pid,
      $prodvers,
      $routing,
      $maxrouting,
      $contour,
      $lowestmaplevel,
      $famname,
      $series,
      $description,
      $copyright) = InfoFromTDB($tdb);
print "PID:                $pid\n";
print "Prodvers:           $prodvers\n";
print "FamID:              $famid\n";
print "Codepage:           $codepage\n";
print "Routing:            $routing\n";
print "max. routingtype:   $maxrouting\n";
print "Contour:            $contour\n";
print "Maplevel:           $lowestmaplevel\n";
print "Famname:            $famname\n";
print "Series:             $series\n";
print "Description:        $description\n";

print $cmd "cd $destdir\n";
print $cmd "gmtool -i . -i ..\\*.typ -i ..\\$tdb -mapsource=ov:$ovmap;tdb:";
print $cmd "$tdb;noov;notyp;nomdx;nomdr;noinst";
print $cmd " --mapfamilyname=\"$famname\" --mapseriesname=\"$series\"";
print $cmd " --description=\"$description\" --routable=$routing -highestroutable=$maxrouting";
print $cmd " --maxbits4overview=$lowestmaplevel --hasdem=1 --hasprofile=$contour -copyright=*I*";
print $cmd " -o . --overwrite\n";
print $cmd "cd ..\n";

print $cmd "\n";
print $cmd "echo The new files are in directory '$destdir'.\n";
print $cmd "echo The old files will be replaced by the new files.\n";
print $cmd "echo cancel with CTRL^C\n";
print $cmd "pause\n";
print $cmd "move /Y \"$destdir\\*.img\" .\n";
print $cmd "move /Y \"$destdir\\*.tdb\" .\n";
print $cmd "rem rmdir /S /Q \"$destdir\"\n";
print $cmd "\n";
print $cmd "echo running Mapsource\n";
print $cmd "echo cancel with CTRL^C\n";
print $cmd "pause\n";
print $cmd "del /Q \"%APPDATA%\\GARMIN\\MapSource\\TileCache\\*.*\"\n";
print $cmd "\"%ProgramFiles(x86)%\\Garmin\\MapSource.exe\"\n";

close($cmd);

print "**********************\n";
print "With the new file '$cmdfile' you can create and insert the DEM.\n";
print "**********************\n";


# ------------------------------------------------------------------------------

sub Do4Img {
my $basename = shift;
my $splitdir = shift;
my $destdir = shift;
my $hgtpath = shift;
my $usedummydata = shift;
my $lastcolstd = shift;
my $hgtoutput = shift;
my $dlon = '';
   foreach my $para (@_) {
      $dlon .= " --dlon=$para";
   }
my ($description, $mapid) = InfoFromIMG($basename . '.img');
   if ($description !~ /$mapid/) {
```

```
      $description .= ", ($mapid)";
   }

   print $cmd "rmdir /S /Q \"$splitdir\"\n";
   print $cmd "gmtool -i $basename.img --split -o \"$splitdir\"\n";
   print $cmd "for /d %%S in ($splitdir\\*) do move \"%%S\\*\" \"$splitdir\" && rmdir \"%%S\"\n";
   print $cmd "BuildDEMFile --dem=\"$splitdir\\$mapid.DEM\" --tre=\"$splitdir\\$mapid.TRE\"".
                          " --hgtpath=\"$hgtpath\" --overwrite $dlon" .
                ($usedummydata ? " --usedummydata": "") .
                ($lastcolstd ? " --lastcolstd": "") .
                ($hgtoutput ? " --hgtoutput=\"$destdir\\hgtoutput\\dat$mapid.txt.zip\"": "") .
                "\n";
   print $cmd "gmtool -i \"$splitdir\\$mapid.*\" --join=tile -o \"$destdir\\$basename.IMG\"" .
                    " --description=\"$description\" --overwrite\n";
   print $cmd "rmdir /S /Q \"$splitdir\"\n";
   print $cmd "\n";
}
# ------------------------------------------------------------------------------
# get infos from IMG file
sub InfoFromIMG {
my $img = shift;
my $description = '';
my $mapid = -1;
   # my @stdout = `gmtool -i osmmap.img -I`;

   open my $pipe, '-|', "gmtool -i $img -I";
   my @stdout = <$pipe>;
   close($pipe);

   foreach my $line (@stdout) {
      if ($line =~ /Kartenbeschreibung:\s*(.*)$/i) {  # from IMG
         $description = $1;
         next;
      }
      if ($line =~ /MapID:\s*(\d+)/i) {     # from TRE-Subfile
         $mapid = $1;
         next;
      }
   }

   return ($description, $mapid);
}
# ------------------------------------------------------------------------------
# get infos from TDB file
sub InfoFromTDB {
my $tdb = shift;
my $codepage      = 1252;
my $famid         = 0;
my $pid           = 0;
my $prodvers      = 0;
my $routing       = 0;
my $maxrouting    = 0;
my $contour       = 0;
my $lowestmaplevel= 0;
my $famname       = 'Family';
my $series        = 'Series';
my $description   = 'Description';
my $copyright     = '';

   open my $pipe, '-|', "gmtool -i $tdb -I";
   my @stdout = <$pipe>;
   close($pipe);

   #  CodePage:            1252 (0x000004E4)
   #  Family-ID:           7026 (0x1B72)
   #  Product-ID:          1 (0x0001)
   #  ProductVersion:      100 (0x0064)
   #  FamilyName:          A, aio (70260000)
   #  SeriesName:          A, aio (70260000)
   #  Routingföhig:        1 (0x01)
   #  gröter Routing-Typ: 24 (0x18)
   #  mit Contourlinien:   1 (0x01)
   #  DEM:                 1 (0x01)
   #  niedrigster MapLevel:18 (0x12)
```

```
   #  Beschreibung:        A, aio (70260000)
   #  Copyright:           [CopyrightInformation, ProductInformationAndPrinting]
   #                       created by gmtool
   foreach my $line (@stdout) {
      if ($line =~ /CodePage\s*:\s*(\d+)/i) {
         $codepage = $1;
         next;
      }
      if ($line =~ /ly-ID\s*:\s*(\d+)/i) {
         $famid = $1;
         next;
      }
      if ($line =~ /ct-ID\s*:\s*(\d+)/i) {
         $pid = $1;
         next;
      }
      if ($line =~ /ctVersion\s*:\s*(\d+)/i) {
         $prodvers = $1;
         next;
      }
      if ($line =~ /hig\s*:\s*(\d+)/i) {
         $routing = $1;
         next;
      }
      if ($line =~ /Routing-Typ\s*:\s*(\d+)/i) {
         $maxrouting = $1;
         next;
      }
      if ($line =~ /Contourlinien\s*:\s*(\d+)/i) {
         $contour = $1;
         next;
      }
      if ($line =~ /MapLevel\s*:\s*(\d+)/i) {
         $lowestmaplevel = $1;
         next;
      }
      if ($line =~ /FamilyName\s*:\s*(.*)$/i) {
         $famname = $1;
         next;
      }
      if ($line =~ /SeriesName\s*:\s*(.*)$/i) {
         $series = $1;
         next;
      }
      if ($line =~ /Beschreibung\s*:\s*(.*)$/i) {
         $description = $1;
         next;
      }
   }

   return (
               $codepage,
               $famid,
               $pid,
               $prodvers,
               $routing,
               $maxrouting,
               $contour,
               $lowestmaplevel,
               $famname,
               $series,
               $description,
               $copyright,
          );
}
```

Die Kommando-Datei builddemcmd.cmd sieht dann z.B. so aus (Achtung, die Zeilen sind z.T. umbrochen):

```
@echo off
mkdir ~dem
mkdir ~dem\hgtoutput

rmdir /S /Q "~dem2"
gmtool -i 70210001.img --split -o "~dem2"
```

```
for /d %%S in (~dem2\*) do move "%%S\*" "~dem2" && rmdir "%%S"
BuildDEMFile --dem="~dem2\70210001.DEM" --tre="~dem2\70210001.TRE" -hgtpath="d:\osmdata\srtm_zip"
             --overwrite  --dlon=0.00027761 --dlon=0.00049 --dlon=0.00075 -dlon=0.00106
             --dlon=0.0017 --dlon=0.0025 --usedummydata
gmtool -i "~dem2\70210001.*" --join=tile -o "~dem\70210001.IMG" -description="DE-Muenchen,
             (70210001)" --overwrite
rmdir /S /Q "~dem2"

...

cd ~dem
gmtool -i . -i ..\*.typ -i ..\osmmap.tdb
       --mapsource=ov:osmmap.img;tdb:osmmap.tdb;noov;notyp;nomdx;nomdr;noinst
       --mapfamilyname="OSM, Austria" --mapseriesname="Austria, Basis, 5.6.2017"
       --description="AUT, (70210000)" --routable=1 --highestroutable=24 -maxbits4overview=18
       --hasdem=1 --hasprofile=1 --copyright=*I*  -o . --overwrite
cd ..

echo The new files are in directory '~dem'.
echo The old files will be replaced by the new files.
echo cancel with CTRL^C
pause
move /Y "~dem\*.img" .
move /Y "~dem\*.tdb" .
rem rmdir /S /Q "~dem"

echo running Mapsource
echo cancel with CTRL^C
pause
del /Q "%APPDATA%\GARMIN\MapSource\TileCache\*.*"
"%ProgramFiles(x86)%\Garmin\MapSource.exe"
```