



© batsheba, 123RF

QMapShack zeigt nicht nur unterschiedlichstes Kartenmaterial an, sondern ermöglicht darüber hinaus das einfache Planen und Verwalten von Tracks und Routen.

Karsten Günther

Lange Zeit galt QLandkarte(GT) als beste Möglichkeit, um freies Kartenmaterial unter Linux anzuzeigen und um es auf Garmin-Geräte zu laden. Entsprechend lange Gesichter gab es, als das Projekt vor einigen Jahren sanft entschlief. Glücklicherweise fand sich schnell eine Reihe von Entwicklern zusammen und hob ein neues Programm mit erweiterten Funktionen unter dem Namen QMapShack [🔗](#) aus der Taufe.

QMapShack erlaubt es, Kartendaten aus unterschiedlichsten Quellen, online

und offline, zu verwenden und anzuzeigen. GPX-Tracks und Routen lassen sich visualisieren, erstellen und bearbeiten. Sogar ein automatisches Routing ist möglich. Dabei kombiniert QMapShack Informationen aus verschiedenen Quellen als „Projekt“, was das Programm für größere Vorhaben mit mehreren Karten, Tracks oder Routen interessant macht. Alles in allem lässt sich QMapShack als universelles Karten-, Planungs- und Routing-Werkzeug beschreiben.

## QMapShack in der Praxis

QMapShack präsentiert sich nach dem Start einfach und übersichtlich **1**. Die wichtigsten Funktionen finden Sie wie üblich im Menü und der Werkzeugleiste. Um Karten darstellen zu können, müssen Sie sie zunächst einmal bereitstellen. Dabei akzeptiert QMapShack diverse Online- und Offline-Formate, insbesondere

### README

Karten anzeigen kann jeder Webbrowser. Wollen Sie aber bestimmte Aspekte von Karten visualisieren, wie Wegpunkte oder Tracks, dann kommen Programme wie QMapShack ins Spiel.

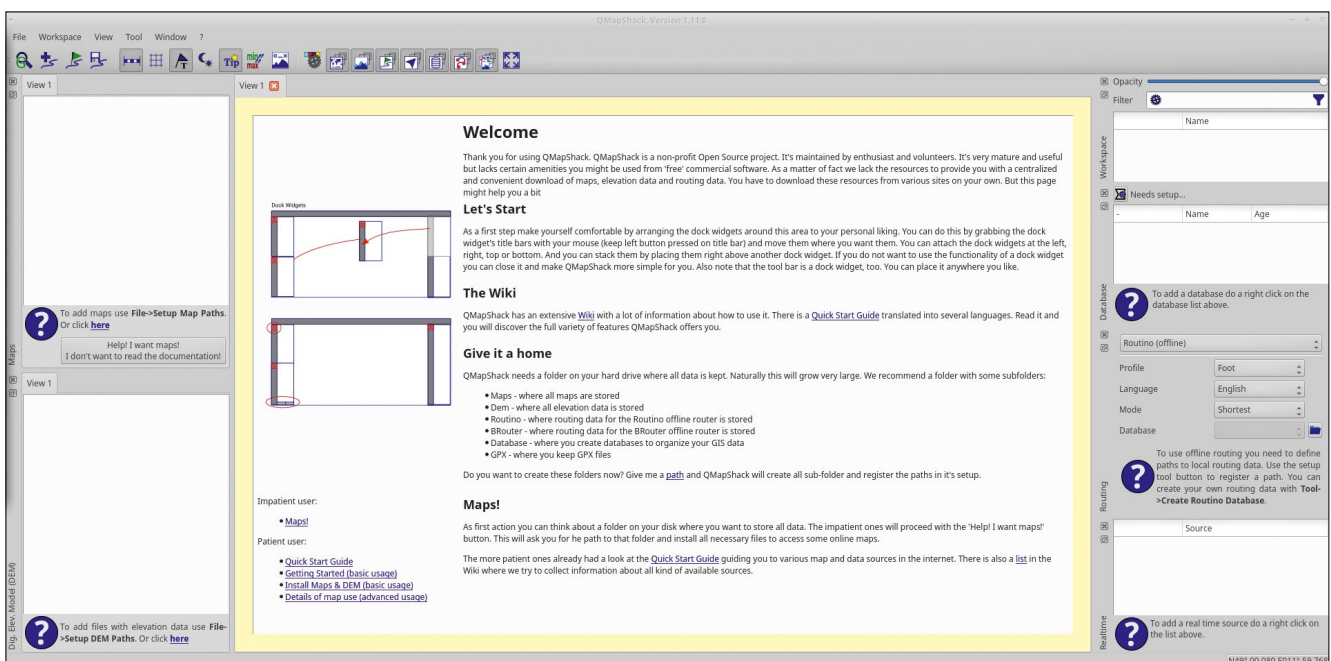
### Listing 1

```
$ hg clone https://bitbucket.org/maproom/qmapshack QMapShack
$ mkdir build_QMapShack
$ cd build_QMapShack
$ ccmake ../QMapShack
$ make
$ sudo make install
```

VRT, TMS, WMTS, RMAP und IMG. Letzteres bezieht sich auf durch Mkgmap erzeugte Karten für Garmin-Geräte (siehe Kasten [GMAP](#)).

Die Konfiguration erfolgt bei Offline-Karten über *File | Setup map paths* [2](#). Dort sehen Sie den Pfad, unter dem die Software die Tiles – die angezeigten „Kacheln“ der Karten – abspeichert (in der Standardkonfiguration `~/ . QMapShack/`). Zusätzlich lassen sich weitere Ordner eintragen, unter denen QMapShack Karten suchen soll. Dabei lädt die Anwendung nur Karten aus dem angegebenen Ordner, nicht aber aus dessen Unterordnern.

Verfügen Sie nicht über Offline-Karten, können Sie auch auf Online-Karten zurückgreifen. Zu deren Definition dienen beispielsweise \*.tms-Dateien („Tile Map Server“). Der Schalter mit dem Plus-Symbol erlaubt das Hinzufügen von Karten und Kartenverzeichnissen. Legen Sie neue \*.tms-Dateien an, dann erscheinen diese im linken Ansichtsfenster von QMapShack unter dem verwendeten Dateinamen. Identische Dateinamen in unterschiedlichen Verzeichnissen lassen sich hier nicht mehr unterscheiden, achten Sie also beim Benennen der Dateien auf eine Unterscheidung [3](#).



**1** Ansichten und Einstellungen strukturieren die Oberfläche von QMapShack, das eine gut verständliche Anleitung zum Einrichten gibt.

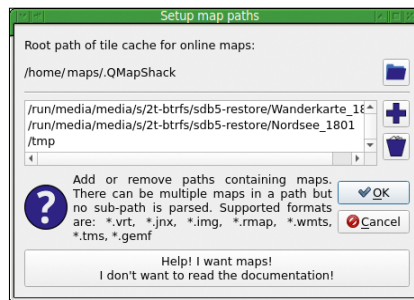
## QMapShack bauen

Die meisten gängigen Distributionen führen die aktuelle QMapShack-Version 1.11.0 noch nicht in den Paketquellen. Sie müssen die Anwendung in der neuesten Version also meist selbst bauen, was jedoch nicht schwerfällt. Sie benötigen dazu das Versionierungssystem Mercurial (ab Version 4, besser 4.5) und Cmake beziehungsweise Cmake in Version 3.1. Außerdem brauchen Sie GDAL, Proj4 und Routino sowie QuaZip und Qt in Version 5.

[Listing 1](#) zeigt die einzelnen Arbeitsschritte zum Übersetzen des Programms. Zunächst klonen Sie das Mercurial-Repository auf die lokale Festplatte. Danach legen Sie ein Arbeitsverzeichnis an und wechseln anschließend direkt in den neuen Ordner. Mittels Cmake erzeugen Sie nun ein Makefile. Dazu nehmen Sie in der Ncurses-

Oberfläche mit [C] die Konfiguration vor. [G] erstellt dann die Makefiles, [Q] beendet Ccmake zu guter Letzt.

Anschließend baut der Aufruf `make` die Programme unter `bin/`. Dabei entstehen mehrere Binaries: das Hauptprogramm `qmapshack`, das Hilfsprogramm `qmaptool` zur Kartenanzeige sowie `qmt_map2jnx` und `qmt_rgb3pct` für das Konvertieren von Daten. Mit `sudo make install` spielen Sie das Programm gemäß der zuvor getroffenen Vorgaben nach `/usr/local/` ins System ein. Bei Bedarf aktualisieren Sie QMapShack direkt aus dem geklonten Verzeichnis heraus. Wechseln Sie wieder in das Arbeitsverzeichnis, und holen Sie mit `hg pull && hg update` den aktuellen Code. Ein erneutes `make` generiert die neue Version; mit `sudo make install` installieren Sie das Update.



**2** Vor der Nutzung steht die Konfiguration: Geben Sie zunächst ein oder mehrere Verzeichnisse an, in denen Sie das Kartenmaterial vorhalten.

Neben echten Karten zeigt QMapShack auch beliebige Tiles an, wie etwa die von Microsoft via Bing bereitgestellten Luftbilder. Deren Einbinden gestaltet sich nicht gerade intuitiv, wie Listing 2 zeigt. Die hierfür benötigte `.tms`-Datei und weitere Beispiele laden Sie über das Blog [Mtb-touring.net](http://Mtb-touring.net) als ZIP-Datei herunter. Bitte beachten Sie, dass die rechtliche Si-

tuation nicht bei allen Kartenquellen eindeutig ist: So erlaubt Google die Nutzung seiner Tiles nicht mit anderen Programmen als Google Earth oder den hauseigenen Online-Diensten. Zudem lassen sich nicht alle der angebotenen Karten wirklich sinnvoll nutzen, da sie teilweise nur kleine Bereiche abdecken oder nur geringe Zoom-Bereiche bereitstellen. Andere, wie beispielsweise die „Alpenkarte“, haben hingegen einen beachtlichen Umfang und zeigen viele Details.

Neben reinen Kartendaten unterstützt QMapShack auch die Verwendung von DEM-Daten. Das Kürzel steht für „Digital Elevation Model“ und bezeichnet digitale Höhen- oder Geländedaten. Eine Reihe von Servern stellen entsprechende Daten zur Verfügung, die sich im Programm als Online-Karten integrieren lassen. Die Originaldokumentation widmet dem Thema DEM-Karten ein eigenes Kapitel [\[2\]](#).

Zum Navigieren im Kartenfenster verschieben Sie bei gedrückter mittlerer Maustaste den angezeigten Kartenaus-

## GMAP

Zu den besonderen Eigenschaften von QMapShack zählt die Unterstützung für IMG-Dateien. Sie beschränkt sich auf die durch Mkgmap erzeugten Karten für Garmin-Geräte, mit denen QMapShack jedoch sehr gut klarkommt. Das ermöglicht, IMG-Dateien mit QMapShack zu testen, was viel besser klappt, als das mit dem Navi möglich wäre. Zudem lassen sich die IMG-Karten auch offline mit QMapShack nutzen, also genau dort, wo es wichtig ist: draußen, wo es kein Internet gibt. Derzeit arbeiten die Entwickler an der Möglichkeit, Navi-Geräte als GPS-Koordinatengeber für QMapShack einzubinden.

## Konfiguration

Eine wesentliche Neuerung der Version 1.11 findet sich in der überarbeiteten Benutzerführung während des Einrichtens von QMapShack. Für viele Aspekte des Programms mussten Sie früher spezielle Verzeichnisse anlegen und in der Konfiguration eintragen. Das vereinfacht die aktuelle Version deutlich: Über den Pseudo-Link *Give me a path...* auf der Willkommenseite **1** geben Sie nur noch ein Basisverzeichnis an, alle benötigten Ordner erzeugt QMapShack automatisch und trägt sie in die Konfiguration ein. Allerdings sammeln sich unterhalb des Basisverzeichnisses schnell 10 GByte und mehr an Daten an. Gegebenenfalls müssen Sie daher die unterschiedlichen Verzeichnisse auf mehrere Laufwerke verteilen, was sich nur manuell konfigurieren lässt.

## Listing 2

```
<TMS>
<Title>Bing</Title>
<Layer idx="0">
<Script><![CDATA[
(
function convert(z1, x1, y1) {
  serverpart = 0
  serverpart = (serverpart + 1) % 4;
  function encodeQuadTree(zoom, tilex, tiley) {
    var tileNum = []
    for (var i = zoom - 1; i >= 0; i--) {
      var num = (tilex % 2) | ((tiley % 2) << 1);
      tileNum[i] = new String(num);
      tilex >>= 1;
      tiley >>= 1;
    }
    return tileNum.join("");
  }
  return "http://ecn.t" + serverpart + ".tiles.
  virtualearth.net/tiles/a" + encodeQuadTree(z1,x1,y1)
  + ".jpeg?g=1036";
}
)
]]></Script>
</Layer>
<Copyright>Microsoft - Bing</Copyright>
</TMS>
```

schnitt, mit dem Mausrad skalieren Sie dessen Größe. Beides funktioniert auch beim Anlegen von Tracks und Routen noch. Die Drehrichtung des Mausrads ändern Sie mit der Option *Mausrad umdrehen* im Menü *Ansicht*.

## Karten installieren

Manuell installierte Karten, die Sie etwa wie Garmin-Karten mittels Mkgmap erzeugt und im Kartenverzeichnis hinterlegt haben, müssen Sie zuerst aktivieren. Das erledigen Sie über eine Funktion im Kontextmenü der eingebundenen Karten **5**. Aktivierte Karten sortiert

QMapShack immer oben in der Liste ein, wo Sie sie dann umgehend wiederfinden.

### Listing 3

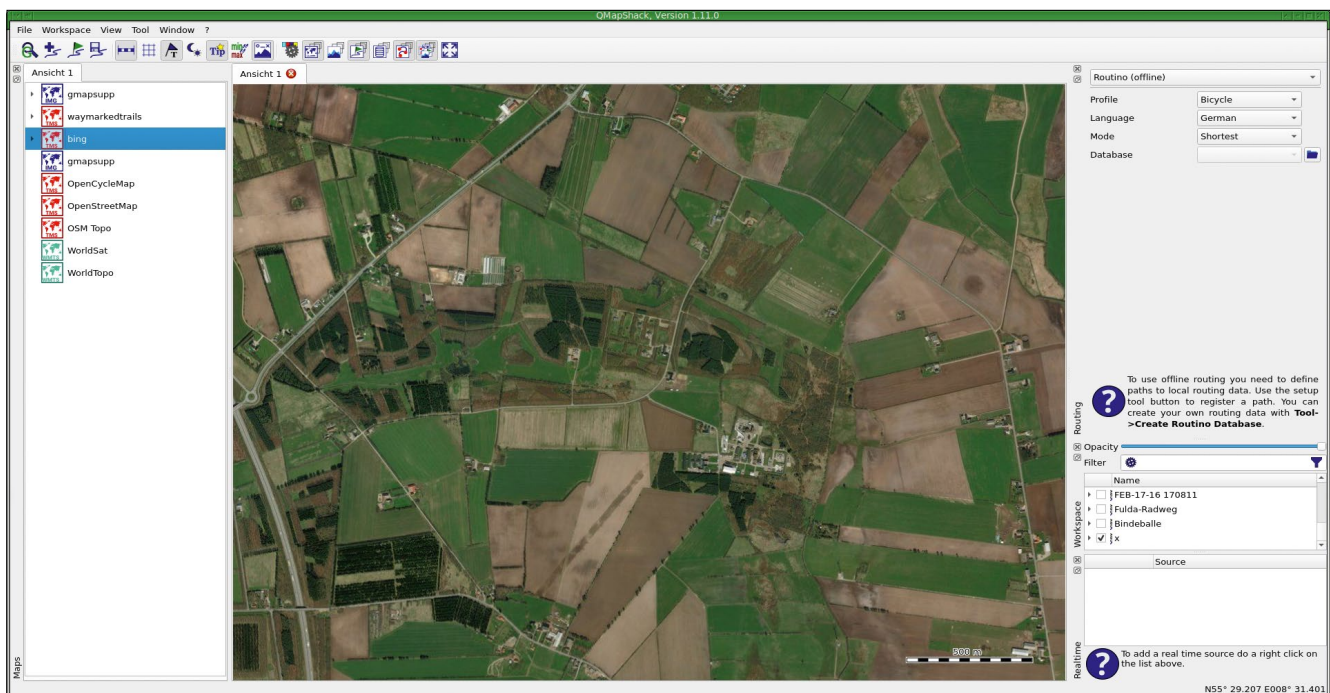
```
<TMS>
<Title>Opencyclemap</Title>
<MinZoomLevel>1</MinZoomLevel>
<MaxZoomLevel>1024</MaxZoomLevel>
<Layer idx="0">
  <ServerUrl>http://a.tile.thunderforest.com/cycle/%1/%2/%3.
  png?apikey=API-Schlüssel</ServerUrl>
</Layer>
<Copyright>Map data: (c) OpenStreetMap contributors, ODbL | Rendering:
(c) OpenCycleMap , CC-BY-SA</Copyright>
</TMS>
```

## API-Key für OpenCycleMap

Seit einiger Zeit verlangt der von Thunderforest.com betriebene OpenCycleMap-Server (und einige weitere) einen sogenannten API-Key, um Kacheln ohne eine Meldung in der Form *API Key Required* als Wasserzeichen auszuliefern **4**. Den Schlüssel erhalten Sie im Rahmen eines Hobbyprojekts nach einer Registrierung kostenfrei [4](#). Nach einer Bestätigungsmail finden Sie den API-Key im Dashboard des Diensts [4](#). Danach müssen Sie QMapShack so konfigurieren, dass die Anwendung den API-Schlüssel bei jeder Anfrage an den Tile-Server von OpenCycleMap mitsendet. Dazu öffnen Sie die Datei *OpenCycleMap.tms* im Kartenverzeichnis, die nur wenige Zeilen XML-Code um-

fasst (Listing 3). In der Zeile *<ServerUrl>* ergänzen Sie dann den API-Schlüssel in der Art *?apikey=API-Key* vor dem schließenden *</ServerUrl>*. Mit dieser Modifikation erhalten Sie wieder „ungestörte“ Tiles vom OpenCycleMap-Server.

Wo in den Konfigurationsdateien sich die Datei *OpenCycleMap.tms* befindet, hängt von mehreren Faktoren ab und lässt sich bei der Konfiguration als *Setup map paths* angeben. Beim Import alter Daten von QLandkarte(GT) finden Sie sie oft auch unter *~/qlandkarte-gt-Konto/OpenCycleMap.tms*. Notfalls suchen Sie diese Datei mittels `find ~ -iname OpenCycleMap.tms`.



**3** QMapShack vermag gleichzeitig mehrere Karten einzubinden und diese übereinandergelegt anzuzeigen.

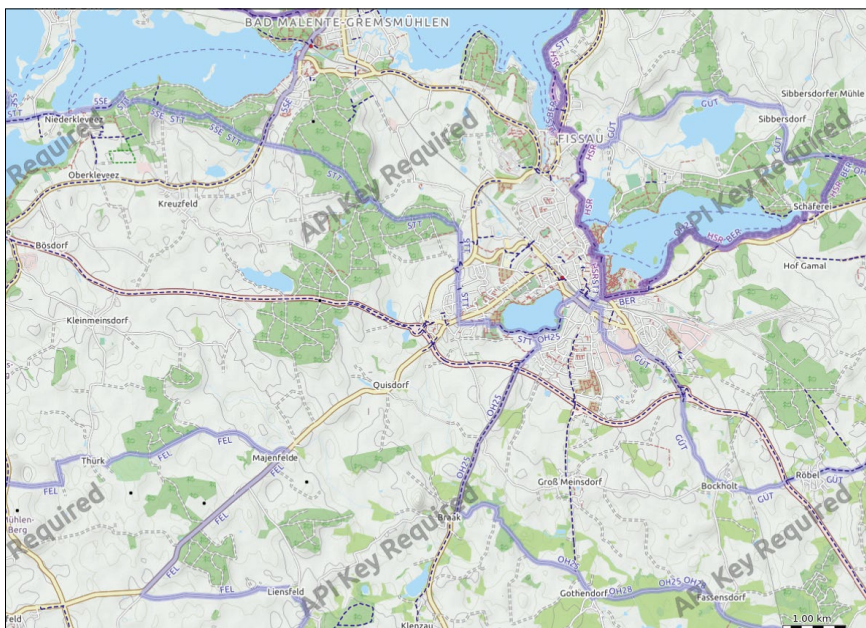
QMapShack erlaubt es, mehrere Karten gleichzeitig zu aktivieren. So lassen sich etwa Luftbilder mit einer Kartendarstellung überlagern. Das erlaubt ganz neue, bisher nur durch Kombination in einer Bildbearbeitung zu erzeugende Formen von „realistischen“ Kartenbildern.

Die Reihenfolge der Karten spielt eine wichtige Rolle: Weiter unten liegende Karten überdecken die darüberliegenden. Die Deckkraft jeder aktivierten Karte regeln Sie über einen versteckten Dialog, der sich nach einem Mausklick auf das kleine Dreieck vor dem Karten-Icon öffnet **6**. Der oberste Schieberegler steuert dabei die Deckkraft der Ebene.

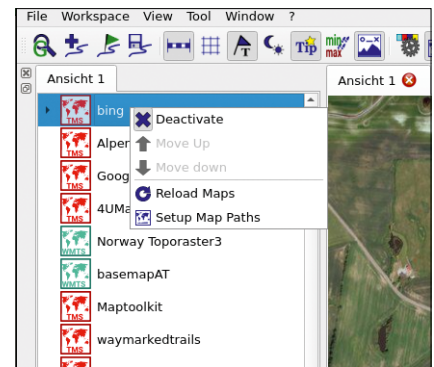
Schenken Sie auch den zwei Schieberegler darunter Beachtung: *Cache Size (MB)* begrenzt den auf der Festplatte belegten Speicherplatz pro Kartendarstellung, und *Expiration (Days)* sorgt dafür, dass QMapShack automatisch veraltete Kacheln entfernt und später beim Aufruf der Region durch neue ersetzt.

## Drucken und Speichern

Über [Strg]+[P] oder den Menüpunkt *Datei | Drucken* aktivieren Sie die Druck- und Speicherfunktion für den angezeigten Kartenausschnitt **7**.



**4** Eine Reihe von Servern markieren inzwischen die Karten mit einem Wasserzeichen, wenn Sie keinen API-Key angeben.



**5** Über das Kontextmenü lassen sich Karten in der Darstellung aktivieren und wieder abschalten.

Zunächst wählen Sie dazu den fraglichen Kartenausschnitt aus und stoßen über die Buttons oben rechts das Speichern beziehungsweise das Drucken an. Das funktioniert sowohl für die Kartendarstellungen selbst als auch für Luftbilder und überlagernde Darstellungen. Die Vorschau zeigt hierbei allerdings nur sehr eingeschränkt, welchen Bereich Sie wirklich bearbeiten. Dafür blendet QMapShack die reale Auflösung in Metern pro Pixel ein.

In der Praxis mit der Alpenkarte und Bing funktionieren beide Funktionen jedoch selbst in der aktuellen Version der Software nur unbefriedigend: Beim Speichern erhalten Sie zwar einen Screenshot (auch für Luftbilder), aber die Qualität überzeugt nicht **8**.

Die Druckfunktion arbeitet noch weniger zufriedenstellend: Trotz aktiviertem Farbdruck und einem *Drucken in Datei* lieferte QMapShack im Test nur einen Schwarzweißdruck mit weitem Rand. In Kombination mit der OpenStreetMap und OpenCycleMap wussten beide Funktionen allerdings zu überzeugen.

## Tracks und Routen

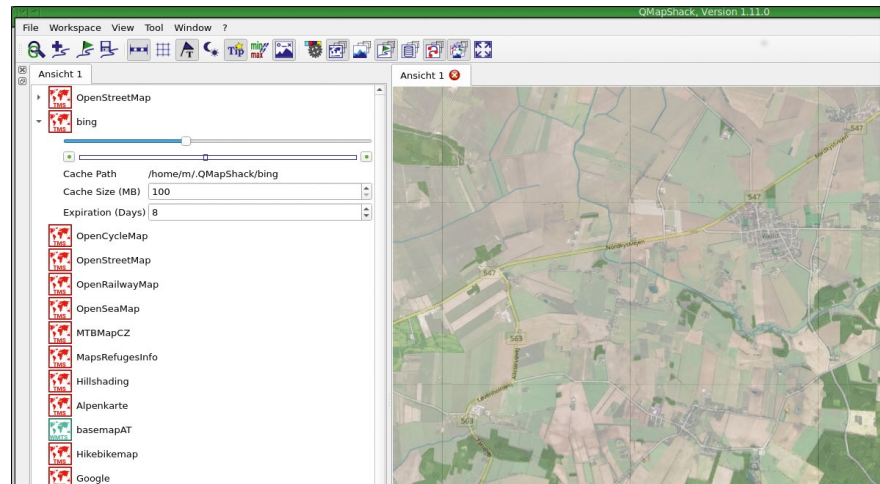
Eines der Einsatzgebiete von QMapShack liegt im bequemen Planen von Routen am Bildschirm. Zum Anlegen eines Tracks klicken Sie mit der rechten Maustaste in das Kartenfenster und wählen *Track hinzufügen* im Kontextmenü **9**. Das aktiviert den „Gummibandmodus“ und erzeugt eine gerichtete Linie, die Sie mit

Stützpunkten versehen in der Karte platzieren. Mit der rechten Maustaste beenden Sie diesen Modus und legen dann den so erzeugten Track über *Als neu speichern* als GPX-Track auf der Festplatte ab. Gespeicherte Tracks erscheinen automatisch auch in der Projektübersicht, die QMapShack voreingestellt unten in der rechten Seitenleiste anzeigt.

Auch Tracks aus Portalen wie Waymarkedtrails.org lassen sich in QMapShack integrieren. Dazu drücken Sie [Strg]+[L] oder wählen den Menüpunkt *Datei | GIS Daten laden*. Die Anwendung unterstützt unter anderem den Import von GPX- und TCX-Dateien, auch mit dem neuen FIT-Format aktueller Garmin-Geräte kommt sie zurecht. Die entsprechenden Tracks erscheinen ebenfalls in der Projektübersicht. Falls die Wegspur die Kartenansicht zu stark stört, regeln Sie über den Schieber *Transparenz* die Darstellung nach.

Routen erstellen Sie analog zu Tracks über das Kontextmenü. Dort wählen Sie

die Option *Route hinzufügen*, stellen die Route zusammen und speichern sie ab. Anders als bei einem Track lässt sich eine solche Route jederzeit nachträglich korrigieren. Den Editor-Modus reaktivieren Sie,



**6** Die Deckkraft von Kartendarstellungen lässt sich stufenlos einstellen und erlaubt so ein teilweises Überdecken von Karten und Luftbildern.

# Linux-Zertifizierung LPIC-1 / LPIC-2

## Mit Ingo Wichmann

- Lernen Sie mit LPI-zertifizierten Trainern und Dozenten!
- 100% abgestimmt auf die originalen Lehrpläne des LPI!
- Bereiten Sie sich optimal auf die LPIC-1- und LPIC-2-Prüfungen vor!

linuxhotel

CMG  
IT-ACADEMY



**IT-Onlinetrainings**  
Mit Experten lernen.

**LPIC-Prüfungsvorbereitung**  
mit Ingo Wichmann, Linuxhotel

LPIC-1 Kurs LPI 101  
**299 €**

LPIC-2 Kurs LPI 201  
**299 €**

LPIC-1 Kurs LPI 102  
**299 €**

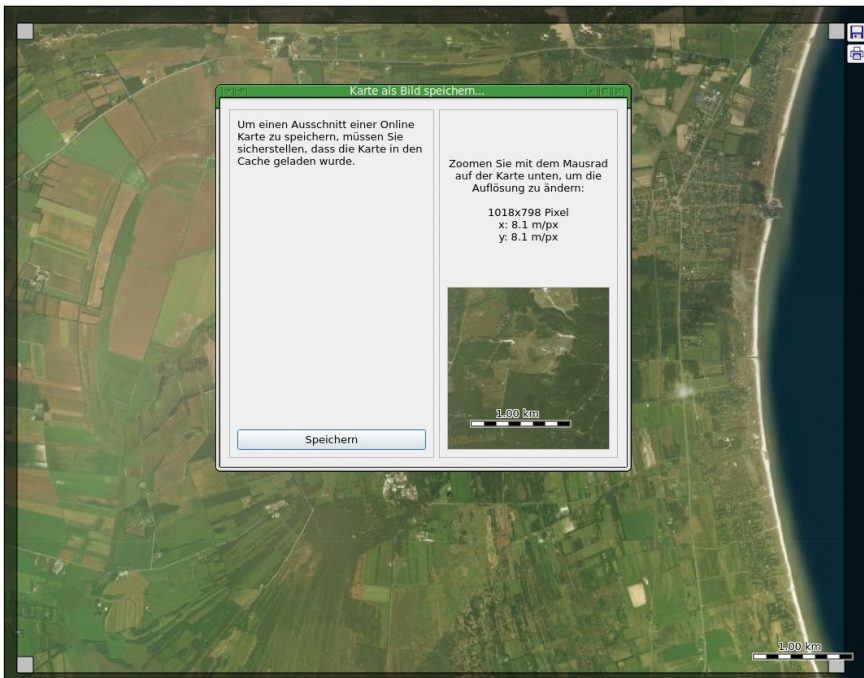
LPIC-2 Kurs LPI 202  
**299 €**

LPIC-1 Paket (101+102) **499 €** LPIC-2 Paket (201+202) **499 €**



/ComputeAcademy

[www.compute-academy.de](http://www.compute-academy.de)



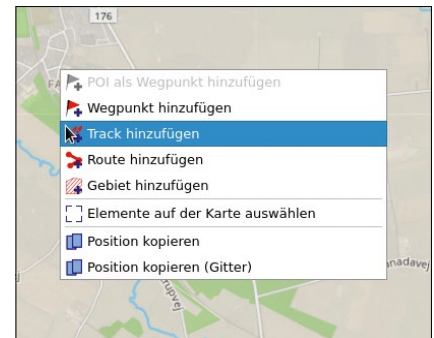
**7** Über die Druckfunktion bringen Sie die Karte nicht nur zu Papier, sondern speichern den gewünschten Ausschnitt auch als Bild ab.



**8** Bei der OpenStreetMap und OpenCycleMap stimmt das Ergebnis, bei übereinandergelegten Karten erzeugt QMapShack allerdings Fehler im Bild.

**Listing 4**

```
$ sudo apt install gcc make libc6-dev libz-dev libbz2-dev subversion
$ svn co http://routino.org/svn/trunk routino
$ cd routino && make && make install
```



**9** Mit einem Rechtsklick in die Karte öffnet sich ein Kontextmenü, über das Sie Tracks und Routen erstellen.

indem Sie den Routenverlauf mit der linken Maustaste anwählen und auf das Icon *Routenpunkte verschieben* klicken **10**.

### Routing mit Routino

Für zusätzliche Möglichkeiten beim Planen von Routen richten Sie den Open-Source-Router Routino ein (siehe [Kasten Routino installieren](#)). Das Programm berechnet auf Basis von topografischen Daten die Verbindung zwischen zwei Punkten auf der Karte. Allerdings benötigen Sie zusätzliche Daten (für Europa mehr als 5 GByte), um das Routing auf dem lokalen Rechner zu ermöglichen. Dann stehen Ihnen allerdings ähnliche Möglichkeiten zur Verfügung wie bei den Online-Diensten Naviki oder Openrouteservice.

Konfigurieren Sie zunächst Routino, indem Sie ein Verzeichnis mit den Routing-Daten angeben. Verwenden Sie eine Garmin-Karte, dann müssen Sie zusätzliche Dateien installieren: Die in der Gmapsuppl. img vorhandenen Routing-Daten kann Routino nicht nutzen. Die fraglichen Daten legen Sie im Konfigurationsverzeichnis unter Routino/ ab. Dieses Verzeichnis darf auch auf anderen Datenträgern liegen.

Um eine Route zu berechnen, setzen Sie zunächst den Startpunkt, aktivieren dann über den Button A das automatische Routing und erhalten dann, gemäß der Einstellungen im Routerdialog, entweder die kürzeste oder schnellste Strecke zur aktuellen Position des Mauszei-

gers. Routino bietet allerdings nur sehr begrenzte Einstellungsmöglichkeiten – Naviki oder Openrouteservice liefern in der Regel deutlich bessere Ergebnisse.

Auch die Integration von Routino funktioniert überraschend gut, wenn auch die Routing-Funktion viel zusätzlichen Speicherplatz benötigt. (cla) ■

## Fazit

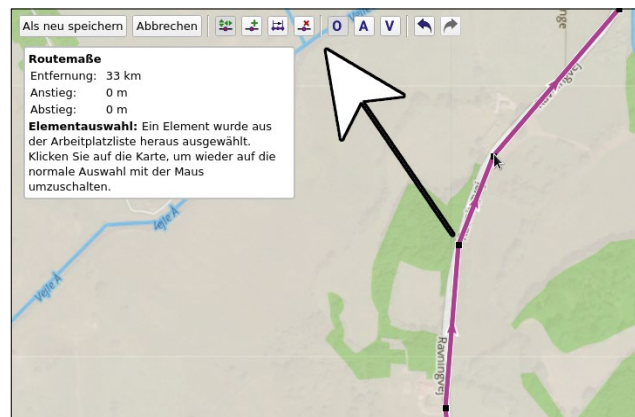
Mit QMapShack steht Outdoor- und Mapping-Enthusiasten ein interessantes und leistungsfähiges Programm zur Verfügung. Kleinigkeiten wie die nicht in allen Situationen gut funktionierenden Speicher- und Druckfunktionen trüben zwar ein wenig das Bild, stellen aber im Grunde ohnehin nur zusätzliche Features dar.

Die Darstellung von Online- und Offline-Karten funktioniert ausgesprochen gut, ebenso wie das Anlegen und Verwalten von Tracks und Routen. Nach einer gewissen Einarbeitungsphase lässt sich QMapShack ausgesprochen gut zum Planen und Dokumentieren von Touren nutzen.



Weitere Infos und interessante Links

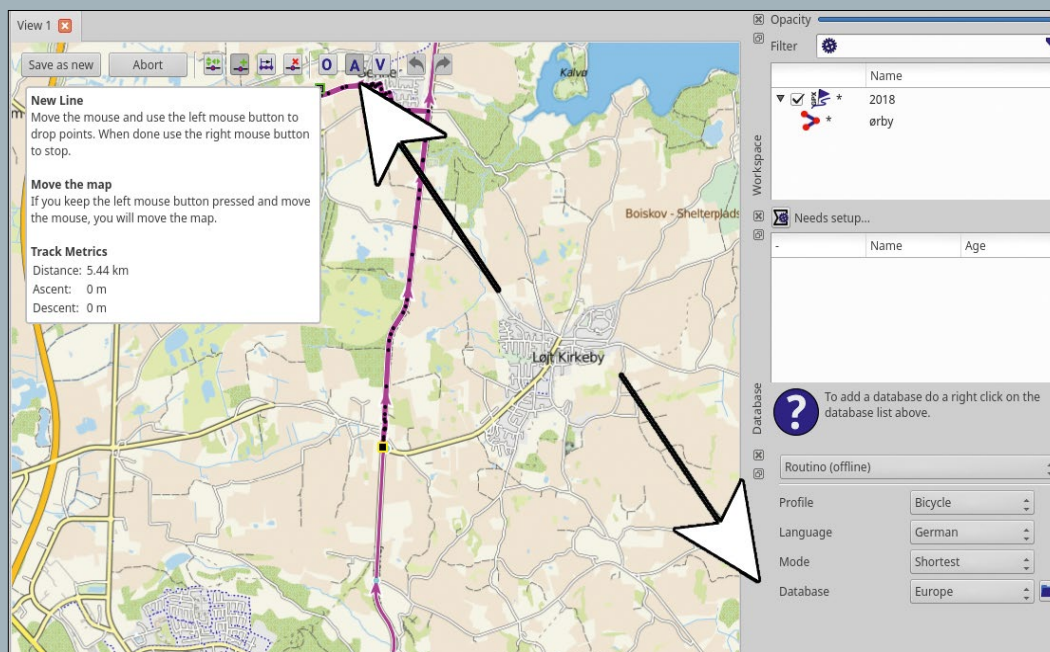
[www.linux-user.de/qr/41076](http://www.linux-user.de/qr/41076)



10 Der Routeneditor erlaubt die Stützpunkte bereits definierter Routen nachträglich anzupassen.

## Routino installieren

Routino ist der bei QMapShack bevorzugt eingesetzte (Offline-)Router. Das Programm findet sich in den Paketquellen vieler Distributionen. In manchen Umgebungen müssen Sie die Anwendung allerdings von Hand bauen, um die aktuellste Version zu nutzen. Listing 4 zeigt den Ablauf für Ubuntu. Die Routing-Daten laden Sie aus dem Internet [☞](#). Das entsprechende Archiv enthält vier Dateien mit einem Umfang von insgesamt mehr als 5 GByte, die Sie in das Verzeichnis `qmapshack-data/Routino/` entpacken. Zudem unterstützt Routino auch PFB-Files, wie sie beispielsweise die Geofabrik [☞](#) anbietet. Anschließend aktivieren Sie das Routing durch Auswahl des Routers, des Profils und der Datenbank [11](#).



11 Nach dem Setzen der Datenbank arbeitet Routino automatisch, wenn Sie den Router im Modus A erzeugen.