
OSMAND PLUGIN FÜR OWNTRACKS RECORDER

7. OKTOBER 2016
BJÖRN PRÖPPER

Inhalt

Einleitung.....	2
Systembeschreibung	3
Voraussetzungen	3
Installation.....	4
Konfiguration von OsmAnd	4
Tracking starten.....	12
Anhalten des Online Tracking.....	15
Anlagen.....	17
Skriptcode.....	17
Abbildungsverzeichnis:.....	18

Einleitung

Der Personentracker nutzt das Duo Mosquitto¹ und ot-recorder² als System zur Erfassung und zur Darstellung von Positionsdaten. Dazu gibt es die Möglichkeit die Daten mit einem Smartphone unter Nutzung der App Owntracks³ zu erfassen, oder von einem Rechner (z.B. Raspberry) mit GPS und Internetanbindung das Skript „PosUpload“ zu nutzen.

Eine unter Android beliebte App zur Navigation ist die Software OsmAnd⁴. Diese bietet Offline Kartendienste und die Möglichkeit Bewegungen als GPX-Track zu erfassen. Darüber hinaus kann sie als Navigationssoftware verwendet werden. Die Nutzung von OsmAnd als Navigationssoftware und von Owntracks zum Übermitteln der Positionsdaten erscheint unzweckmäßig. OsmAnd bietet eine Möglichkeit Daten zum Online Tracking zu versehen. Dabei gibt es zwei Möglichkeiten, einmal die Nutzung des OsMo Systems und zum anderen die Möglichkeit Positionsdaten online an einen frei zu definierenden Server zu senden. Der OsmAnd Client verschickt dabei die Daten über das http-Protokoll. Versuche ergaben, dass ein versenden über https ist nicht möglich. Darüber hinaus „versteht“ OsmAnd das MQTT Protokoll nicht.

Serverseitig versteht ot-recorder zwar http, aber erkennt das Datenformat von OsmAnd nicht nativ. Ot-recorder verfügt über ein Plug-In, das es ermöglicht über Lua Hooks Daten durch eine Vorverarbeitung zu schicken, bevor der Server die Daten verarbeitet. Da der vorliegende ot-recorder-Server aber hinter einem Webserver als Proxy arbeitet und Daten nur über https entgegennimmt, ist es notwendig eine andere Lösung zu finden, um die Daten von OsmAnd an ot-recorder zu senden.

Dazu wurde eine Lösung entwickelt, die einen eigenen Listener für OsmAnd auf dem Server installiert, die Daten umformt und dann an den MQTT Broker sendet.

Das Ergebnis eines erfolgreichen Test mit einem Übertragungsintervall von 90sec in OsmAnd zeigt Abbildung 1.

¹ <https://mosquitto.org/>

² <http://owntracks.org/booklet/guide/apps/> Download unter: <https://github.com/owntracks/recorder> oder <https://itunes.apple.com/de/app/owntracks/id692424691?mt=8>

³ <https://play.google.com/store/apps/details?id=org.owntracks.android&hl=de> oder

⁴ <http://OsmAnd.net/> und <https://play.google.com/store/apps/details?id=net.OsmAnd&hl=de>



Abbildung 1 Erfolgreicher Test OsmAnd und ot-recorder

Systembeschreibung

Die App OsmAnd auf einem Smartdevice/Smartphone sendet in regelmäßigen Abständen Positionsdaten an den Listener. Dieser lauscht im vorliegenden Fall auf Port 8001 auf dem Server. Er empfängt Daten von allen Clients die ihn erreichen können. Aus diesen Grund stellt er im Grunde genommen ein Sicherheitsrisiko für den Server als potentielltes Einfallstor für einen Angreifer dar. Dies ist aber zurzeit nicht zu vermeiden. Das Skript modifiziert die Daten, so dass sie als JSON Datensatz an den MQTT- Broker übergeben werden können. Der wiederum übergibt sie an ot-recorder Server, der dann die Daten speichert und dem Nutzer zur Verfügung stellt.

Voraussetzungen

Ein Server auf dem ein MQTT Broker auf Port 8883 aktiv ist. Dieser sollte sich auf dem gleichen „localhost“ befinden, kann sich aber auch auf einem anderen Server befinden.

Der Server muss über das Internet erreichbar sein, d.h. man muss ggf. eine Portweiterleitung im Router einrichten.

Das Plug-In ist in Python 2.7 geschrieben. In Python muss die Bibliothek „wsgi“⁵ installiert sein.

Darüber hinaus ist für die Kommunikation mit dem MQTT Server die Bibliothek Paho⁶ zu installieren.

Die anderen verwendeten Bibliotheken scheinen zum normalen Umfang einer Python-Dokumentation zu gehören und sind daher automatisch verfügbar.

⁵ <http://wsgi.readthedocs.io/en/latest/what.html>

⁶ <https://pypi.python.org/pypi/paho-mqtt/1.1>

Installation

Das Skript osmtrack.py läuft mit den Rechten eines nicht privilegierten Nutzers. Es kann zum einen über eine Autostart Routine geladen werden, oder direkt durch den Nutzer gestartet werden. Auf Grund der einfachen Installation bietet sich ein Start über Crontab an.

```
@reload /home/pi/osmtrack.py &
```

Das Skript muss ausführbar sein

```
Chmod 755 /home/pi/osmtrack.py
```

Das Skript muss mit Unix Zeichensatz eingespielt werden. Eine Kodierung für Windows funktioniert nicht!

Aus Sicherheitsgründen kann das Skript auch über einen Shell Zugriff nur bei Bedarf gestartet werden.

Das Skript läuft dann auf Port 8001, bis ein Fehler auftritt, oder es durch ein „kill“ beendet wird.

Innerhalb des Skriptes sind die entsprechenden Zugangsparameter für den MQTT Server zu setzen.

Das Skript kann keine Identifizierung des Clients vornehmen.

Konfiguration von OsmAnd

Auf dem Smartphone ist in der App OsmAnd das Plug-In Streckenaufzeichnung zu aktivieren. Unter Einstellungen ist für das entsprechende Profil unter dem Punkt Online Tracking Web Adresse die URL und der Port 8001 zu installieren.

```
http://Server-URL:8001?osmtrack.py?lat=\(0\)&lon=\(1\)&tst=\(2\)&alt=\(4\)&speed=\(5\)
```

Der Parameter „tst“ wird korrekt übertragen, obwohl OsmAnd und ot-recorder unterschiedliche Epoch Formate verwenden. Das Skript überträgt die Daten korrekt und wandelt die Formate (Long und Int) korrekt um.

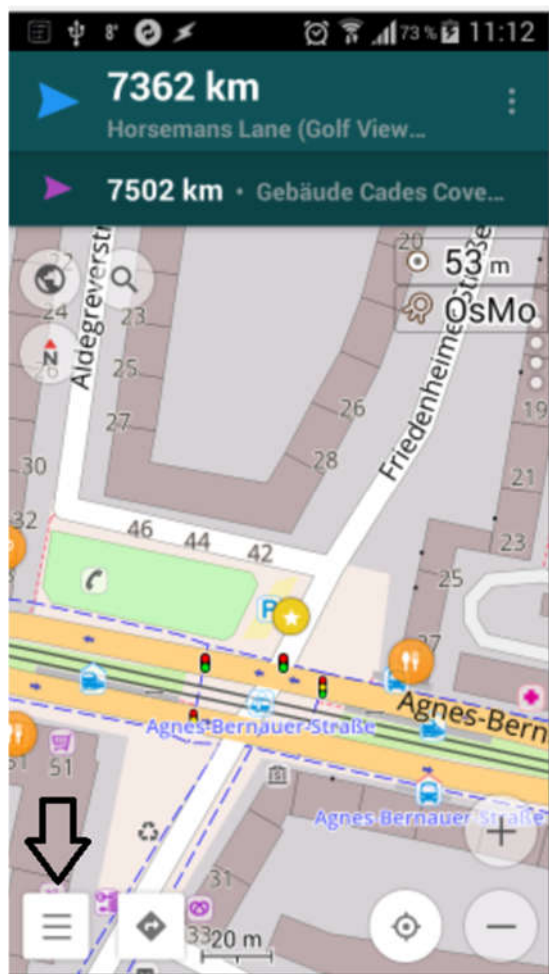


Abbildung 2 Startbildschirm von OsmAnd

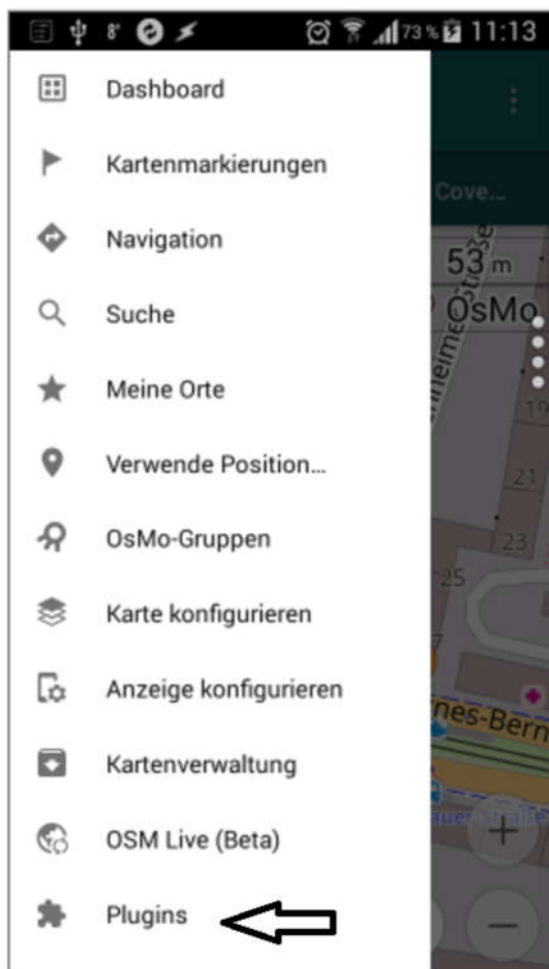


Abbildung 3 Plugins wählen

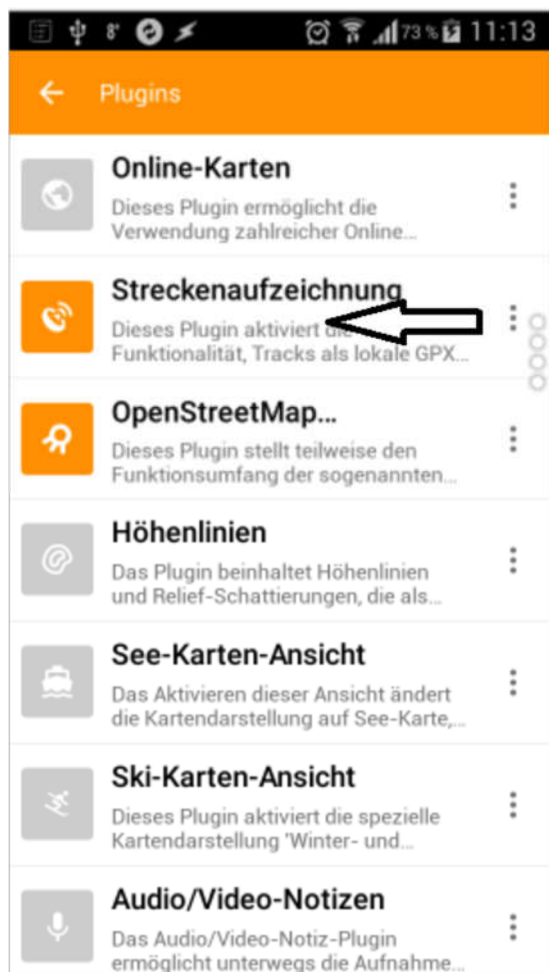


Abbildung 4 Streckenaufzeichnung wählen

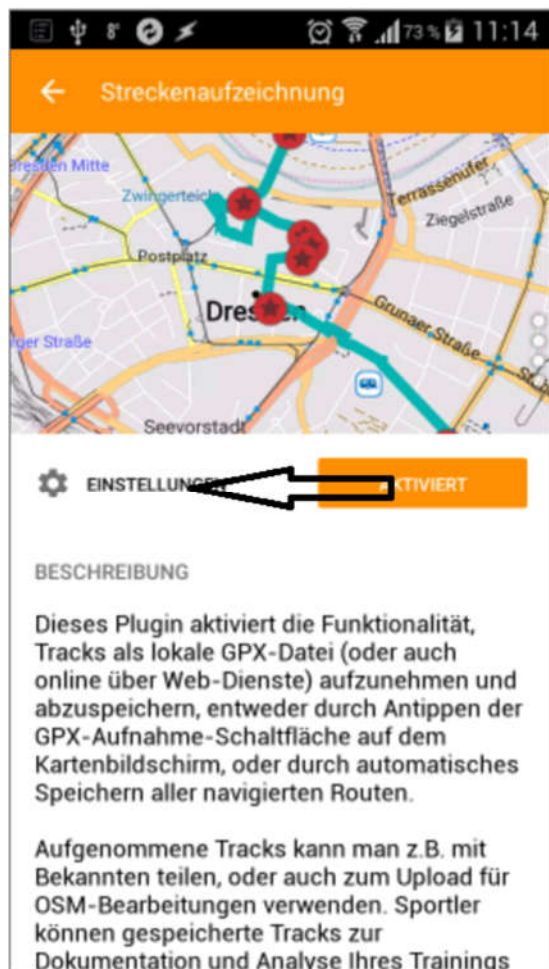


Abbildung 5 Einstellungen auswählen

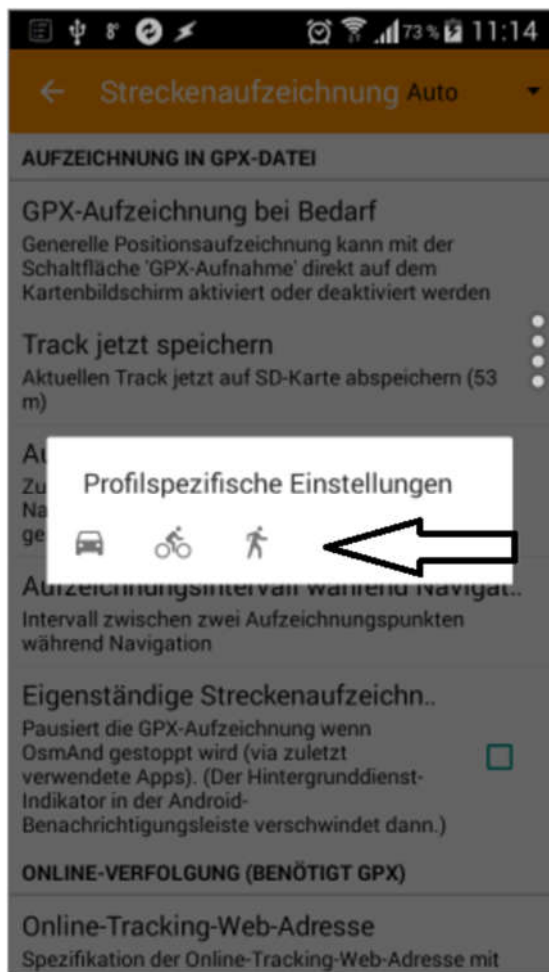


Abbildung 6 Modus auswählen - üblicherweise der Fahrzeugmodus

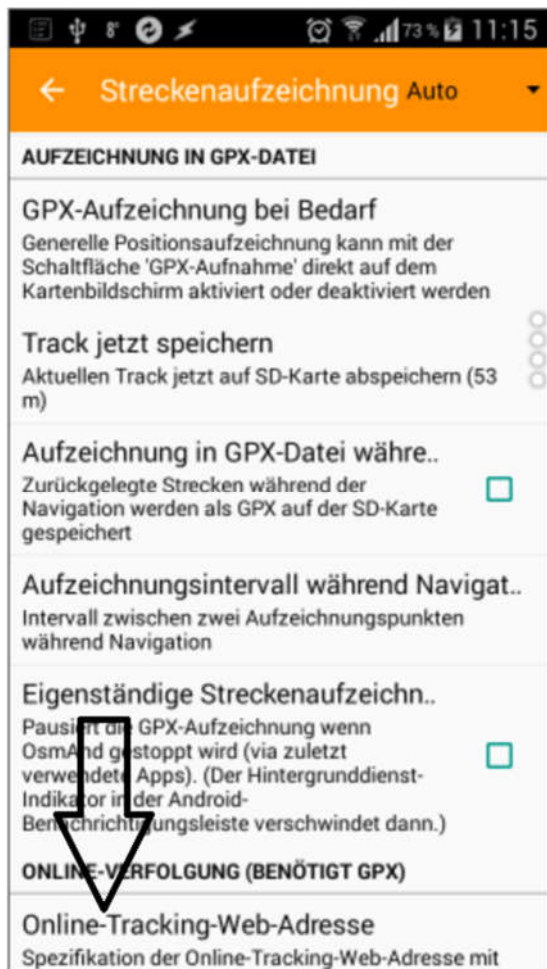


Abbildung 7 Online Tracking Adresse wählen, um den Server zu konfigurieren

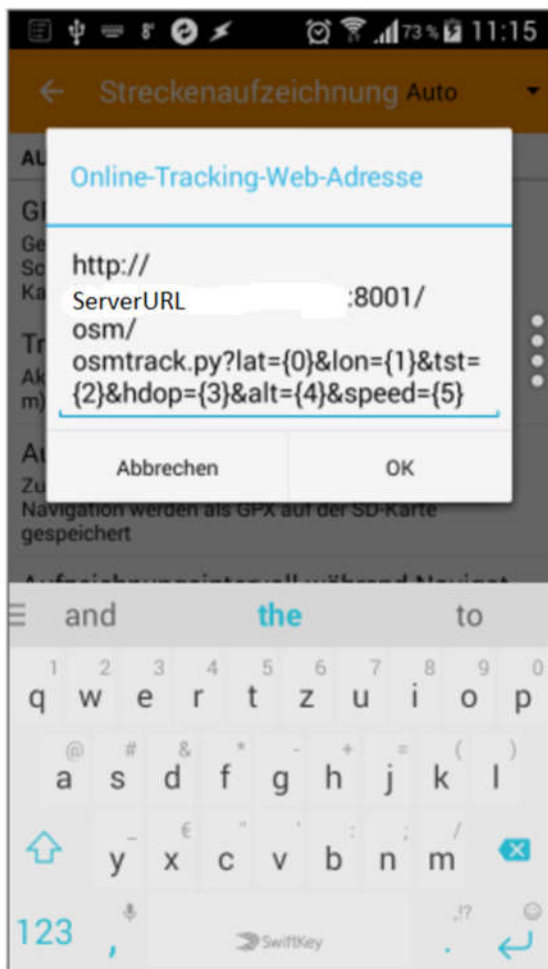


Abbildung 8 Serveradresse:8001 als Server angeben und „OK“ drücken

Tracking starten

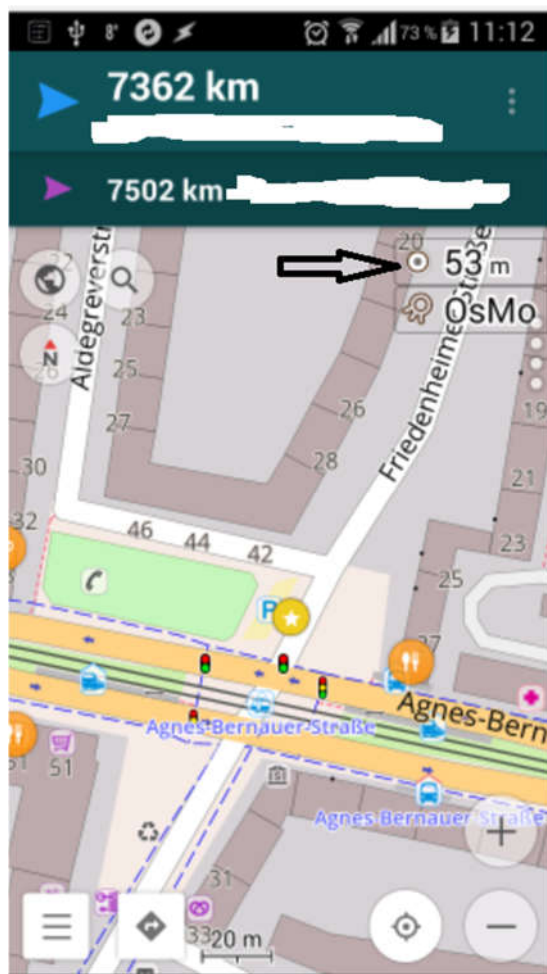


Abbildung 9 GPX-Tracking aufrufen

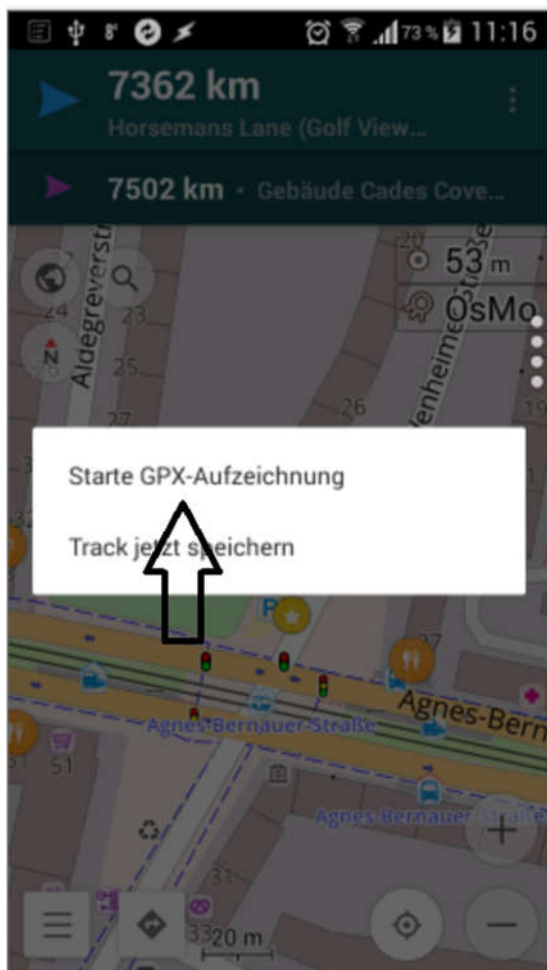


Abbildung 10 GPX Tracking starten

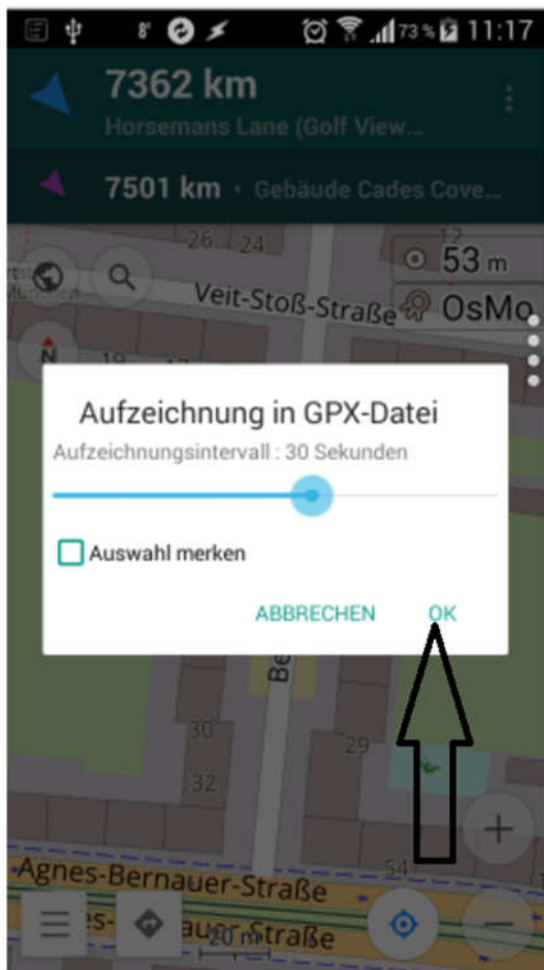


Abbildung 11 Intervall ändern oder bestätigen

Die Online Verfolgung startet sich dann automatisch mit der GPX Aufzeichnung.

Anhalten des Online Tracking

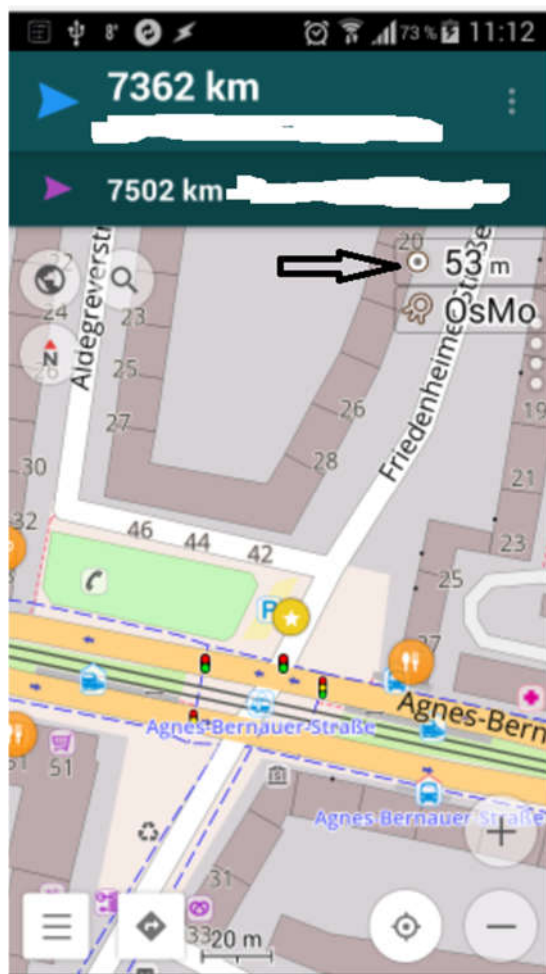


Abbildung 12 Aufrufen des GPX Menüs

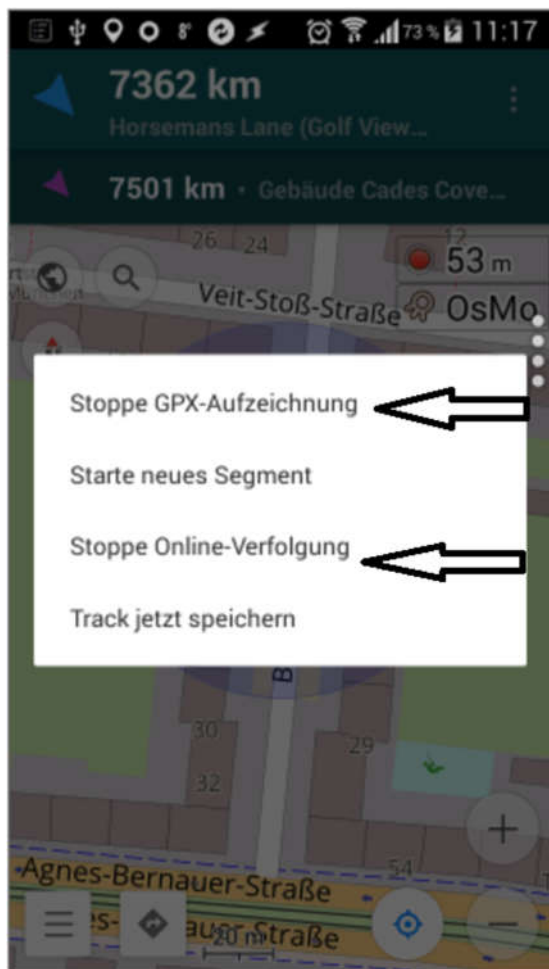


Abbildung 13 GPX Aufzeichnung stoppen, oder nur die Online Verfolgung stoppen

Anlagen

Skriptcode

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
.....
```

Created on Mon Oct 3 2016

Last Edit on Fri Oct 6 2016

```
@author: bjoern
.....
```

```
import paho.mqtt.publish as mqtt
import os, json, cgi, cgiib
import datetime
from wsgiref.simple_server import make_server
```

```
# osmupload ist der Handler der bei eingehenden Calls auf Port 8001 aufgerufen wird
def osmupload(env, start_response):
    userid={'username':"angeben",'password':"angeben"}
    topic="owntracks/bjoern/osmand"
    hostname="localhost"
    port=8883
    data={
        "_type" : "location",
        "alt" : 0,
        "lat" : 34.533333,
        "lon" : 69.166666,
        "tid" : "bp",
        "tst" : 0
    }
    #Zuweisung der aktuellen Daten
    htmldata = cgi.FieldStorage(environ=env)
    data['lat']=htmldata.getvalue('lat')
    data['lon']=htmldata.getvalue('lon')
    data['alt']=htmldata.getvalue('alt')
    data['tst']=int(long((htmldata.getvalue('tst')))/1000)-
int(datetime.datetime.utcnow().timestamp(0).strftime("%s"))
    if (data['lat']!=None) and (data['lon']!=None):
        #Json als String formatieren
        datastr=json.dumps(data)
        print datastr
        mqtt.single(topic,payload=datastr,auth=userid,port=port,hostname=hostname)
        status = '200 OK' # HTTP Status
        headers = [('Content-type', 'text/plain; charset=utf-8')] # HTTP Headers
        start_response(status, headers)
        return []
    else:
        status = '500' # HTTP Status
        headers = [('Content-type', 'text/plain; charset=utf-8')] # HTTP Headers
        start_response(status, headers)
        return []

#hier wird der Server erzeugt und der Handler angegeben
httpd=make_server("",8001,osmupload)
print("Serving on port 8001...")
#enables CGI Debugging
cgiib.enable()
# Serve until process is killed
httpd.serve_forever()
```

Abbildungsverzeichnis:

Abbildung 1 Erfolgreicher Test OsmAnd und ot-recorder	3
Abbildung 2 Startbildschirm von OsmAnd	5
Abbildung 3 Plugins wählen	6
Abbildung 4 Streckenaufzeichnung wählen	7
Abbildung 5 Einstellungen auswählen	8
Abbildung 6 Modus auswählen - üblicherweise der Fahrzeugmodus	9
Abbildung 7 Online Tracking Adresse wählen, um den Server zu konfigurieren	10
Abbildung 8 Serveradresse:8001 als Server angeben und „OK“ drücken	11
Abbildung 9 GPX-Tracking aufrufen	12
Abbildung 10 GPX Tracking starten.....	13
Abbildung 11 Intervall ändern oder bestätigen	14
Abbildung 12 Aufrufen des GPX Menüs	15
Abbildung 13 GPX Aufzeichnung stoppen, oder nur die Online Verfolgung stoppen.....	16