

IMT 4951 - Applied Digital Workflow  
«Routes and schedules of public transport»

Project

Nosov Konstantin

091310

19 November 2009

## 1 Planning

### 1. The main problem

Quite a lot of people every year come to Gjøvik [1]. It may be students, tourists, people who have moved due to change jobs, etc. The first time is difficult to navigate in the new location.

It would be nice to have handy multimedia guide of the city that shows the routes and schedules of public transport, with stops in places, with a useful or interesting places of the city.

### 2. The goal of this project

The aim of this project is to write a open source solution that automates generating of KML [2] files based on geotagged files with embedded metadata, with a further displaying of the buses routes of Gjøvik on the map. A well used to consolidate the knowledge gained in previous course Digital workflow fundamentals.

For publication of geographic data will be used:

- Site <http://www.stud.hig.no/> for uploading of KML files and photos
- Account on Google Webmaster Tools [3] for connection with <http://www.stud.hig.no/>

### 3. Methods for solving the problem

The following operations need for solving the problem:

- Collection of data
- Description of data
- Encoding / transcoding of data
- Transportation of data
- Presentation of data
- Writing a script

### 4. Tools for solving the project

- GPS tracker
- Camera

- UNIX shell [4]
- Open source software
- Web service Google Webmaster Tools
- Web service Google-Maps [5]

#### 5. The end result

In the end, I hope, will be tested, easy and understandable to the user a well-documented bash shell-script.

#### 6. Legal implications of the data

Collected data will be consistent for License "Attribution - Noncommercial - No Derivatives"

This license is allow the free circulation of products. This license is often called the license "free advertising" because it allows others to download your works and share them with others, until they mention you and link back to you, but they can not in any way to change your work and use it commercial purposes [6].

Gjøvik University College may use my work under this license in the future.

**Project plan - «Routes and schedules of public transport»**

**Last version: 4 November 2009**

**Project manager: Konstantin Nosov**

	Milestones	Activities	Persons	Resources	Risk	43	44	45	46	47	48	49	Total hours
1	Project plan	Writing a paper	Me	-	-								5
2	Operations with data	Collecting of coordinates of bus stops	Me	Camera, GPS tracker	I need GPS tracker!!! Bad weather Technical problems								8
		Working with GPX and JPG	Me	Laptop, software	Technical problems								8
3	Coding	Writing a script	Me	Laptop, software	Technical problems								8
4	Final report	Writing a paper	Me	Previous data Laptop, software	Technical problems								16

## 2 Implementing

### 1. Collecting of data

According to the purpose of my project, was collected information about bus route, schedule and stops on this route. Was chosen bus route number 53 as, I think, the most convenient for international students living in Sørbyen studenthjem. Was used smartphone Nokia 5800 [7] with built-in GPS tracker for collecting geo coordinates of the route.

The following table shows the workout summary:

Activity	Walking
Start time	10/11/2009 19:11:17
Duration	01:48:56
Distance	9.02 km
Steps	11941
Energy	483.6 kCal

During the "walk" the temperature changed from 7 °C to 2.6 °C. Wind speed max 10,6 m/s, min 3,6 m/s (Mobile weather service "1881" [8]).

Also for the collection of geo coordinates of stops used photo camera of smartphone that records the coordinates of the position.

Timetable has been taken from the site [www.opplandstrafikk.no](http://www.opplandstrafikk.no)

### 2. Describing of data

For convenience of further automation, information about each stop was recorded in the appropriate tags using Exiftool [9]. In the tag "Location" was recorded the name of a certain bus stop, in the tag "Description" was recorded the interval of the bus.

### 3. Transformation of data

At the GPS device coordinates of the route recorded in the GPX file. For further processing and extraction of coordinates GPX file has been recoded in the KML file using open source software GPSBabel [10].

### 4. Automation of the generation of KML file

According to the objectives of this project was written bash-script to automatically generate a KML file.

Bash-script performs the following steps:

- asks the input parameters: name of the route, the input GPX file, the output KML file
- reads the input parameters
- using GPSTools GPX file is converted to KML file
- inserting special KML tags in the file stops.kml

Then for all JPEG files in this folder:

- With Exiftool reading information from the tags GPSLatitude, GPSLongitude, Location, Description and write to file stops.kml
- With Exiftool command "-KmlDocumentFolderFolderPlacemarkLineStringCoordinates" reads the coordinates of the path of output.kml and recorded in 111.txt
- using Unix command "sed" [11] removed extra characters "." from 111.txt
- 111.txt written in stops.kml
- inserts closing KML tags
- Using the Unix utilities ZIP [12], creates archive output.kmz

#### 5. Moving of data

Using the Unix command "scp [13]" stops.kml, output.kmz and final bash-script were sent to the web folder on [www.stud.hig.no/~091310](http://www.stud.hig.no/~091310) and can be accessed using wget [14].

#### 6. Presenting using Mashup [15]

Final KML file enables to user to see on the Google maps the route and bus stop of bus number 53. Clicking on any stop user available information about the interval of motion of the bus and the arrival time at this stop.

### 3 End product

The final product consists of a bash-script which automatically generates the KML file.

The code of bash-script:

```
#!/bin/bash
echo "Enter the name of route"
read rou
echo "Enter the input GPX file"
read g
echo "Enter the output KML file"
read k
```

```
gpsbabel -i gpx -f $g.gpx -o kml -F $k.kml
```

```
echo '<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2" xmlns:gx="http://www.google.com/kml/ext/2.2"
xmlns:kml="http://www.opengis.net/kml/2.2" xmlns:atom="http://www.w3.org/2005/Atom">
<Document>
<name>Bus_Stops</name>
<StyleMap id="msn_bus">
<Pair>
<key>normal</key>
<styleUrl>#sn_bus</styleUrl>
</Pair>
<Pair>
<key>highlight</key>
<styleUrl>#sh_bus</styleUrl>
</Pair>
</StyleMap>
<Style id="sh_bus">
<IconStyle>
<scale>0.709091</scale>
<Icon>
<href>http://maps.google.com/mapfiles/kml/shapes/bus.png</href>
</Icon>
<hotSpot x="0.5" y="0" xunits="fraction" yunits="fraction"/>
</IconStyle>
<ListStyle>
</ListStyle>
</Style>
<Style id="sn_bus">
<IconStyle>
<scale>0.5</scale>
<Icon>
<href>http://maps.google.com/mapfiles/kml/shapes/bus.png</href>
</Icon>
<hotSpot x="0.5" y="0" xunits="fraction" yunits="fraction"/>
</IconStyle>
<ListStyle>
</ListStyle>
</Style>' > stops.kml |
```

```
for image in *.jpg
do
```

```
longitude=`exiftool -GPSLongitude -n -S -S -S $image`
latitude=`exiftool -GPSLatitude -n -S -S -S $image`
name=`exiftool -Location -S -S $image`
d=`exiftool -Description -S -S $image`
```

```
echo "<Placemark>
<name>${name}</name>
<description>
<![CDATA[
<Table width=100%>
<tr><b>${rou}</b></tr>
<tr><td>Days</td><td>Time</td><td>Interval</td></tr>
<tr><td>Monday-Friday</td><td>6.15-23.15</td><td>30 min</td></tr>
<tr><td>Saturday</td><td>8.15-23.15</td><td>30 min</td></tr>
]]>
</description>
</Placemark>"
```

```

<tr><td>Sunday</td><td>10.15-23.15</td><td>1 hour</td></tr>
<tr><td colspan=2><small>$d</small></td></tr>
<tr><td colspan=2><small>f-Forward b-Backward S-Sunday</small></td></tr>
<tr><td colspan=2><a
href="http://www.opplandstrafikk.no/samferdsel/prog/ruter/ruter.asp?reg=region050">Rutetabell for
bybuss Gjøvik</a></td></tr>
<tr><td colspan=2><a href="www.opplandstrafikk.no">www.opplandstrafikk.no</a></td></tr>
</Table>

```

```

]]>
</description>
<styleUrl>#msn_bus</styleUrl>
<Point>
<coordinates>$longitude,$latitude</coordinates>
</Point>
</Placemark>" >> stops.kml |
wait
done

```

```

exiftool -KmlDocumentFolderFolderPlacemarkLineStringCoordinates -S -S -S -S $k.kml > 111.txt
ttt=`sed -e 's/. //g' 111.txt`

```

```

echo "<Placemark>
<name>Path</name>
<styleUrl> lineStyle</styleUrl>
<LineString>
<tessellate>1</tessellate>
<coordinates>" >> stops.kml

```

```

echo $ttt >> stops.kml

```

```

echo "</coordinates>
</LineString>
</Placemark>" >> stops.kml

```

```

echo "</Document>
</kml>" >> stops.kml
echo "Enter the output KMZ file"
read kmz
zip $kmz.kmz stops.kml

```

```

exit

```

For automatic generation of KML file user should have:

1. GPX file with the route
2. photo with Exif-information about geographical position, location and description
3. bash-script



## 4 Conclusion

Expected result coincides with the end result. But I didn't use Google Webmaster Tools and work with <http://www.stud.hig.no/> directly.

Bash-script is clear and requires no special skills.

During the execution of the script, shows a slight warning, do not affect the result of the operation.

**References:**

- 1 <http://en.wikipedia.org/wiki/Gjøvik>
- 2 <http://code.google.com/apis/kml/documentation/>
- 3 <http://www.google.com/webmasters/>
- 4 [http://en.wikipedia.org/wiki/Unix\\_shell](http://en.wikipedia.org/wiki/Unix_shell)
- 5 <http://maps.google.com/>
- 6 Translation from: [http://ru.wikipedia.org/wiki/Creative\\_Commons](http://ru.wikipedia.org/wiki/Creative_Commons)
- 7 <http://europe.nokia.com/find-products/devices/nokia-5800-xpressmusic>
- 8 <http://www.1881.no/>
- 9 <http://www.sno.phy.queensu.ca/~phil/exiftool/>
- 10 <http://www.gpsbabel.org/>
- 11 <http://en.wikipedia.org/wiki/Sed>
- 12 [http://linux.about.com/od/commands/l/blcmdl1\\_zip.htm](http://linux.about.com/od/commands/l/blcmdl1_zip.htm)
- 13 [http://en.wikipedia.org/wiki/Secure\\_copy](http://en.wikipedia.org/wiki/Secure_copy)
- 14 <http://en.wikipedia.org/wiki/Wget>
- 15 [http://en.wikipedia.org/wiki/Mashup\\_\(web\\_application\\_hybrid\)](http://en.wikipedia.org/wiki/Mashup_(web_application_hybrid))