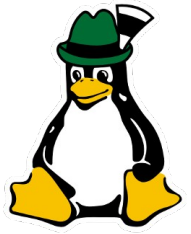


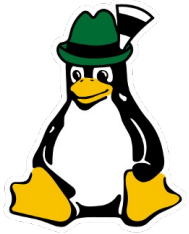
Grazer Linuxtage 2016

Präzise GNSS-Positionierung mit dem Raspberry Pi und RTKLIB

Dr. Franz Fasching

April 2016

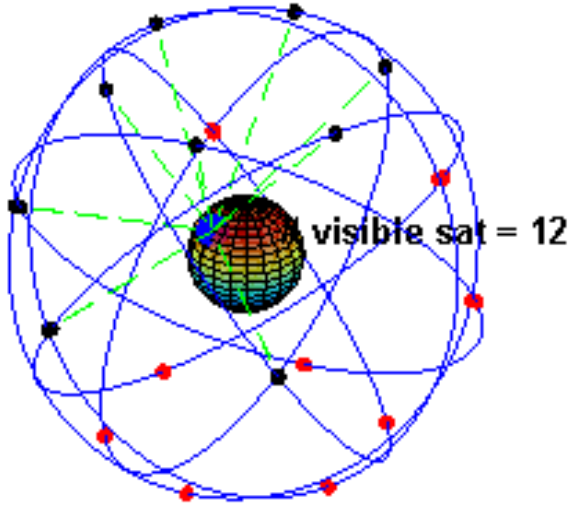
Überblick

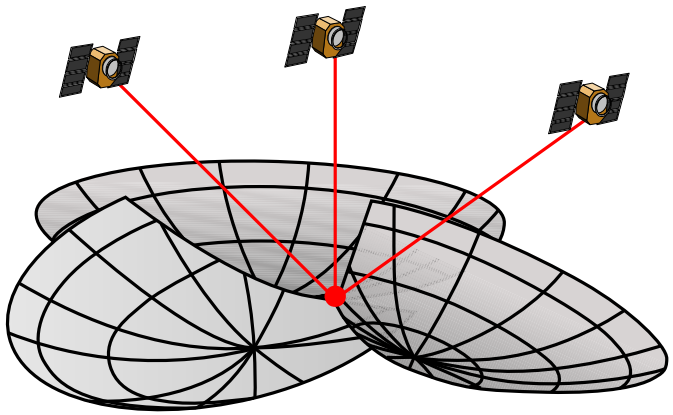


- Wie funktioniert Satellitennavigation?
- Hardwarevoraussetzungen
  - Raspberry Pi
  - GNSS-Empfänger
  - Antenne
- Softwarevoraussetzungen
  - Raspbian
  - RTKLIB
- Precise Point Positioning vs. Real-Time Kinematics
- Anwendungsfälle
- Fazit

GNSS-Satellitennavigation

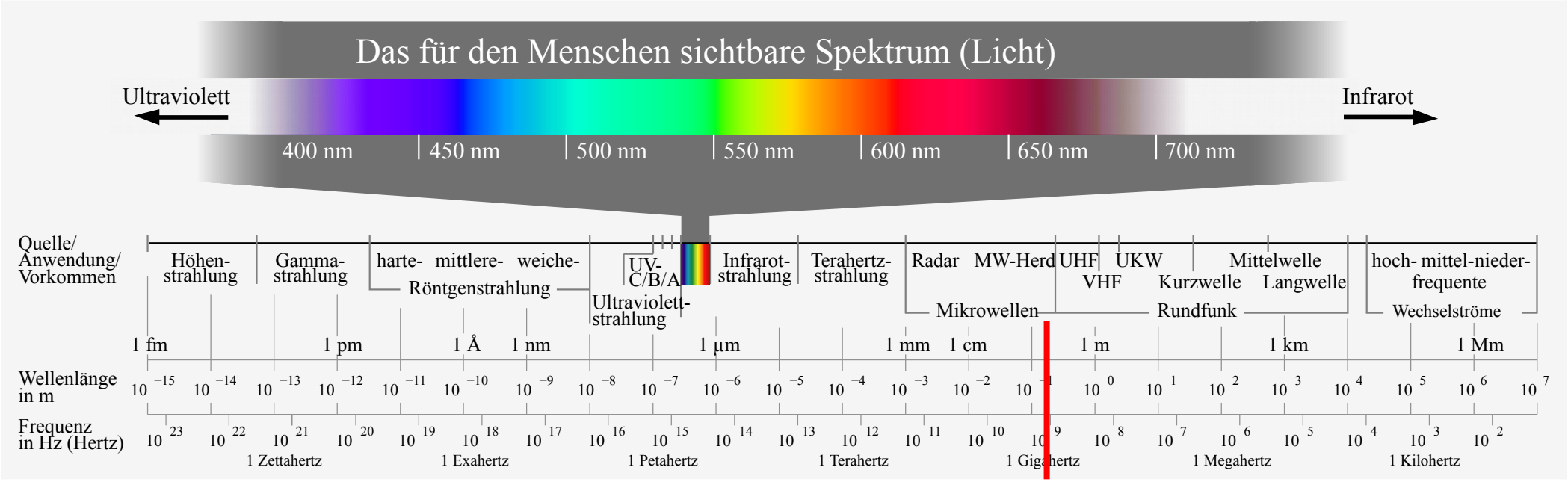
- GNSS (Global Navigation Satellite System) subsummiert GPS (Global Positioning System; NAVSTAR), GLONASS, GALILEO, BEIDOU, QZSS, IRNSS, ...
- Orbit 25.000 km, 6 km/s, 227 Satelliten
- GNSS-Satelliten besitzen 3-4 äußerst präzise Atomuhren (Cs, Rb)
- Senden ihre aktuelle Position und genaue GPS-Zeit auf definierter Frequenz
- Empfänger misst Signallaufzeiten (C/A-Code), bestimmt daraus Distanzen der Satelliten
- 3D-Positionsbestimmung durch Bestimmung des Schnittpunktes von 4 Kugeloberflächen





Trex2001 - Eigenes Werk, CC BY-SA 3.0

GNSS-Frequenzen

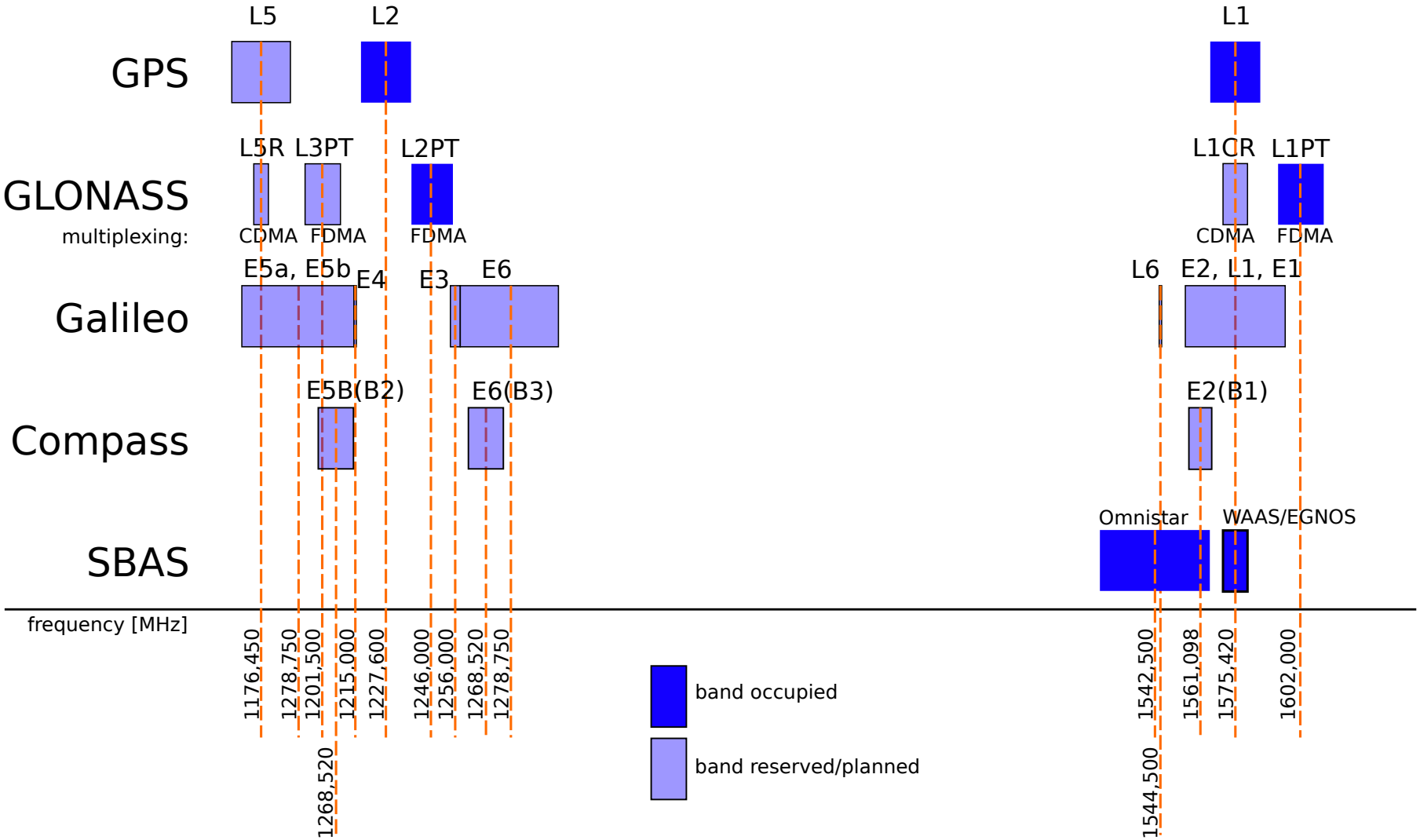


Horst Frank / Phrood / Anony - Horst Frank, Jailbird and Phrood, CC BY-SA 3.0

$$Wellenlänge\ \lambda = \frac{c}{f}$$

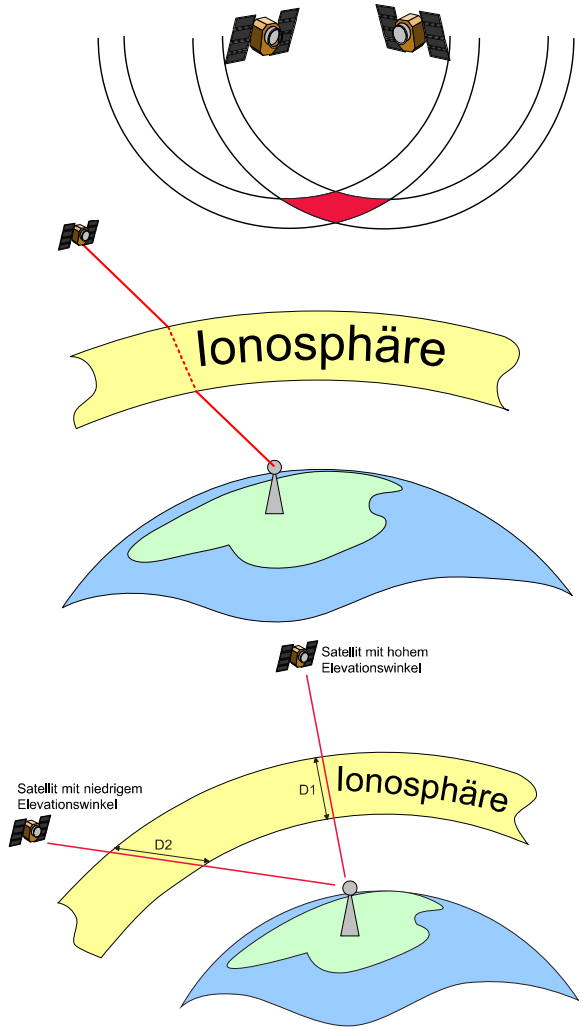
GPS-Frequenz 1575,42 MHz → Wellenlänge 0,19 m

GNSS-Frequenzbänder

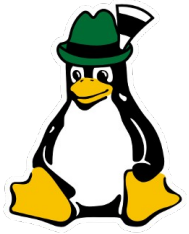


Fehlerquellen

- Numerisch: Verringerung der Positionsgenauigkeit durch schleifende Schnitte (Dilution of Precision, DOP) der Kugeloberflächen
- Satellit:
  - Präzision der Atomuhren (Relativitätstheorie!)
  - Präzision der Umlaufbahnen (Ephemeriden)
- Atmosphäre:
  - Ionosphäre verändert Signallaufzeit frequenz-abhängig (schwankende Anzahl freier Elektronen) durch Refraktion und Eintrittswinkel
  - Troposphäre: frequenzunabhängiger Einfluss
- Dopplereffekt: Bewegung des Satelliten auf den Beobachter zu oder vom Beobachter weg
- Multipath-Effekte: urban canyons, tree canopy
- Empfänger: Präzision der Zeitmessung (TCXO)



User Range Errors (URE)



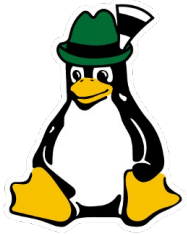
- Summe aller Zeit- und Ortsfehler bei unkorrigierten Meßwerten
  - Satellitenbedingt: SIS-URE (Signal-in-Space User Range Errors)
  - Signalausbreitung: UERE (User Equipment Range Errors)

Quelle	Zeitfehler	Ortsfehler
Satellitenposition	6-60 ns	1-10 m
Zeitdrift	0-9 ns	0-1,5 m
Ionosphäre	0-180 ns	0-30 m
Troposphäre	0-60 ns	0-10 m
Mehrwegeffekte	0-6 ns	0-1 m

Quelle: Wikipedia - Globales Navigationssatellitensystem

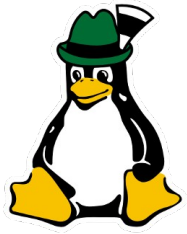
- → Ionosphäre hat größten Einfluss
- Genauigkeitsverbesserung: **überbestimmte Ortung**, **Referenz-messungen** (DGNSS), **Phasenauswertung** und **Korrekturdaten**

Genauigkeitsverbesserung



- **Überbestimmte Ortung**: Wird von praktisch allen Empfängern verwendet. Bei „Überangebot“ an sichtbaren Satelliten, Verwendung der „besseren“, Ausscheiden der „schlechteren“ (Elevation, SNR)
- **Differential-GNSS**: Verwendung einer Referenzmessung (base station), Eliminierung gleicher Fehler → RTK (real-time kinematics)
- **Phasenauswertung**: Zusätzlich zur Standardlösung (Code-Solution) wird die Phasenlage des Satellitenträgersignals (GPS: 1575,42 MHz) ausgewertet. Voraussetzung: Empfänger liefert Rohdaten des Satellitensignals
- **Korrekturdaten** zu Satelliten-Clocks, -Ephemeriden, Ionosphäre, Troposphäre aus eigenen Referenzstationen
  - NASA-CDDIS (Crustal Dynamics Data Information System)
  - IGS (International GNSS Service)
  - Diverse kommerzielle Anbieter

Real-Time Kinematics



- Referenzstation (base station) und Empfänger (rover) mit Funkverbindung (schmalbandig)
- Fehler heben sich bei kleinem Abstand (bis zu 10km) auf
- Messung der Phase des Trägersignals, deshalb Rohdatenausgabe des GNSS-Empfängers erforderlich

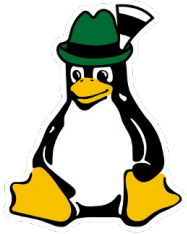
Raw Measurements (the data provided are for one channel of the receiver[1])

6	27	INT8U	Signal Type [2]
7	28	INT8U	Satellite Number
8	29	INT8U	A carrier number for GLONASS
9	30	INT8U	Signal-to-Noise Ratio, dB-Hz
10	31	FP64	Carrier Phase, cycles
11	39	FP64	Pseudo Range, ms
12	47	FP64	Doppler Frequency, Hz
13	55	INT8U	Raw Data Flags [3]
14	56	INT8U	Reserved

NVS BINR Protocol Specification V1.3, Raw Data

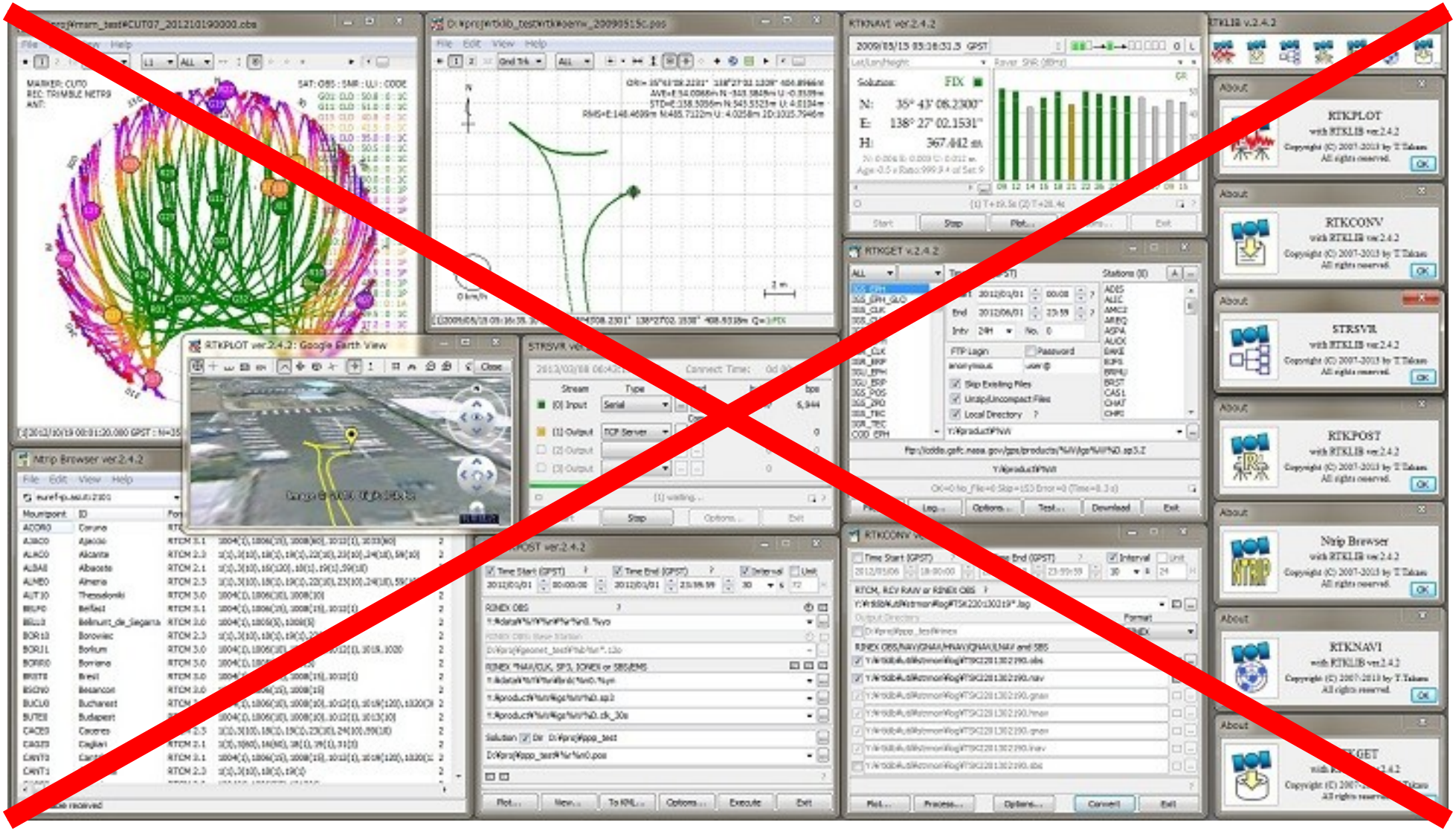
- Problem: *carrier phase ambiguity* da nicht bekannt ist, wieviele Perioden lang das Signal nun ist → Fehler ein Vielfaches der Wellenlänge (19cm)
- *Integer ambiguity resolution* durch statistische Methoden und Vergleich mit der Codelösung

RTKLIB

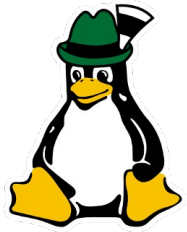


- FOSS-Software zur präzisen Berechnung von GNSS-Positionen
- Autor Tomoji Takasu (Tokyo University of Marine Science and Technology)
- Homepage www.rtklib.com
- Code gehostet auf GitHub: github.com/tomojitakasu/RTKLIB
- Lizenz: BSD 2-clause (opensource.org/licenses/BSD-2-Clause)
- Besteht aus einer Library, einem Windows-GUI (läuft unter WINE) und Commandline-Tools (ANSI C)
- Benötigt einen Empfänger, der Rohphasendaten liefern kann
  - Professionelle Hersteller: NovAtel, Hemisphere, JAVAD, Furuno
  - Low-Cost: u-blox, SkyTraq, NVS
- Beherrscht alle zuvor genannten Genauigkeitsverbesserungen

RTKLIB Windows-GUI

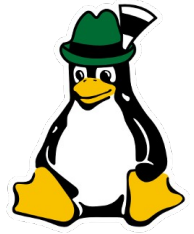


RTKLIB Commandline Tools



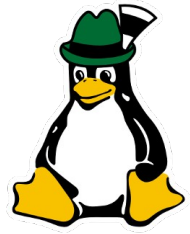
- Real-Time:
  - STR2STR – Stream-to-Stream Converter
    - Konvertiert GNSS-Rohdatenstreams „on-the-fly“
  - RTKRCV – Real-Time Positionsberechnung
    - Bis zu 3 Input-Streams (Base, Rover, Korrekturdaten)
- Postprocessing:
  - CONVBIN – Convert binary
    - Konvertiert binäre Rohdaten ins RINEX Format
  - RNX2RTKP – RINEX to RTK Position solution
    - Berechnet eine präzise PPP oder RTK-Lösung
  - POS2KML – Konvertiert POS-Dateien ins KML-Format

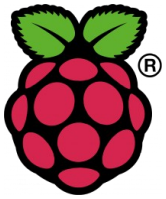
Betriebsmodi und damit erzielbare Genauigkeiten



- Single
  - „Normale“ GPS Codelösung
  - Genauigkeit etwa 2,5m 1-σ 2D
- PPP (Precise Point Positioning) mit Korrekturdaten, Postprocessing
  - Phasenlösung mit CDDIS/IGS Korrekturdaten
  - Genauigkeit bis zu 20cm 1-σ 2D
- RTK short baseline oder RTK long baseline mit Korrekturdaten
  - Konfiguration Rover mit Base Station
  - Genauigkeit etwa 3cm 1-σ 3D
- Long-term static RTK short baseline, Postprocessing
  - Genauigkeit bis zu 4mm 1-σ 3D (!)

Hardware





- Raspberry Pi (beliebiges Modell)
- Rohdatenfähiger GNSS-Empfänger:
  - Modul für den RPi Expansion Port: RaspiGNSS „Aldebaran“ drfasching.com/products/gnss/raspignss
  - USB-Anbindung: optimalsystem.de, onetalent-gnss.com
- GNSS-Antenne guter Qualität (z.B. Tallysman TW-2410 für Base Station, TW-4421 für Rover) drfasching.com/products/gnss/antennas

- Achtung: Antenne muss zum Empfänger passen (LNA-Versorgung, Frequenzen L1/L2, GPS/GLONASS, etc.)
- Achtung: Gute Sichtbedingungen („sky view“) Voraussetzung, großer Sichtwinkel, keine Abschattungen, keine Reflexionen

Hardware Beispiele

RasPiGNSS „Aldebaran“ auf Raspberry Pi A+
NVS NV08C-CSM GPS/GLONASS receiver

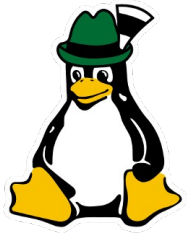
Manufactured in Austria by

KEYTRONIX

Tallysman TW-2410 GPS+GLONASS

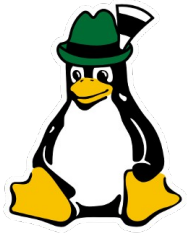
OptimalSystem NV08C-RTK-DSPP
USB, Bluetooth

Software



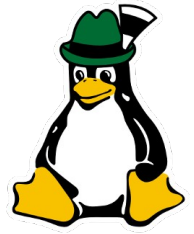
- Raspbian „Jessie“ Lite (die Debian-Variante am Raspberry Pi)
- RasPiGNSS-Tools (rpgtools, drfasching.com/downloads)
- RTKLIB CUI Tools, kompiliert für Raspberry Pi (rtklib, drfasching.com/downloads)
- Optional: Auswertesoftware, welche die gespeicherten Tracks aufbereiten und anzeigen kann, z.B. QGIS
- Vorbereitung Raspbian:
  - Freimachen von `/dev/ttyAMA0`: `console=serial0,115200` aus `/boot/cmdline.txt` entfernen
  - Takt der seriellen Schnittstelle erhöhen: `init_uart_clock=6000000` in `/boot/config.txt` einfügen, sonst können keine 230400 Baud erreicht werden

Konfiguration



- Base Station (str2str)
  - `str2str -in serial://ttyAMA0:230400:8:o:1:off -out tcpsvr://:1234`
- Rover (rtkrcv): Konfiguration der `rtkrcv.conf`
  - `inpstr1-type =serial`
    `inpstr1-path =ttyAMA0:230400:8:o:1:off`
    `inpstr1-format =nvs`
  - `inpstr2-type =tcpcli`
    `inpstr2-path =`*`basestation`*`:1234`
    `inpstr2-format =nvs`
  - `outstr1-type =tcpsvr`
    `outstr1-path =:2947`
    `outstr1-format =nmea`
  - `pos1-posmode =kinematic`
  - `pos2-armode =fix-and-hold`

Ergebnis



- Monitoring mit `status 2` in `rtkrcv` nach etwa 2 Minuten Laufzeit:

```
…
solution status                 : fix
# of input data rover           :
obs(680),nav(6),gnav(5),ion(2),sbs(137),pos(0),dgps(0),err(0)
# of input data base            :
obs(711),nav(0),gnav(0),ion(2),sbs(142),pos(0),dgps(0),err(0)
…
pos xyz float std (m) rover : 0.032,0.015,0.040
pos xyz fixed std (m) rover : 0.031,0.014,0.038
baseline length float (m)   : 9.606
baseline length fixed (m)   : 9.616
```

- → Positionsgenauigkeit etwa 3 cm bei einer Baselinelänge von 10m

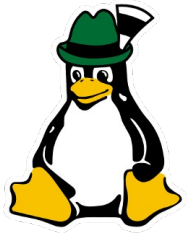
Anwendungsbeispiele

- **Langzeit-RTK im Bergbau (Australien)**
  - Vorspannungsmessung im Straßenbau
  - Messung horizontaler und vertikaler Verschiebungen an Tagbau- und Tiefbauminen
  - Messung der Oberflächeneffekte von Tiefbauminen
- Langzeitaufzeichnung und periodische Übermittlung von Rohdaten, RTK-Postprocessing mit Korrekturdaten samt statistischer Auswertung um präzise minimalste Bewegungen zu registrieren.
- *„We find the 24 hour averaged continuous static data to be easily and consistently sub 10mm....if not better. The low power draw is ideal for very long occupation. We are absolutely ecstatic with the technology.“*

AEROSPATIAL

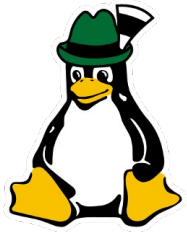
GNSS Monitoring

Anwendungsbeispiele (2)



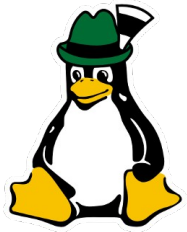
- **Realtime-RTK - „Precision Farming“**
  Autonomes Fahren im Agrarbereich zur präzisen Feldbearbeitung bzw. Dünger-/Pestizidausbringung.
  Ein Raspberry Pi / RasPiGNSS als fixe Base Station in der Umgebung, ein zweiter am Traktor/Arbeitsgerät verbunden mit einer Autopilot-Software. Funkstrecke zur Korrekturdatenüberittlung über ISM-Band (2.4 GHz)-Module (z.B. XBee Pro).
- **PPP Postprocessing**
  Verifikation von Mautstrecken für GPS-basierte Mautsysteme. Aufzeichnung Rohdaten mit Raspberry Pi / RasPiGNSS und PPP-Postprocessing mit CDDIS-Korrekturdaten. Nach der Messfahrt Hochladen der Rohdaten auf Server, der ständig CDDIS-Korrekturdaten sammelt, die Rohdaten im Batchbetrieb auswertet (rnx2rtkp) und das Ergebnis zum Download ablegt.

Fazit

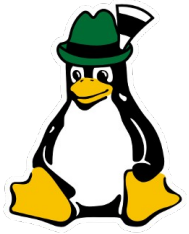


- Ein Raspberry Pi kann mit einem Low-Cost L1 GNSS-Empfänger in Kombination mit RTKLIB unter bestimmten Voraussetzungen zur präzisen GNSS-Positionsbestimmung genutzt werden
- Zeitdauer für einen Fix relativ lang, da im Gegensatz zu professionellem L1+L2-Equipment nur die L1-Frequenz benutzt wird → untauglich für On-the-Spot-Messungen im Vermessungswesen
- Preis für einen Rover aber sehr gering im Gegensatz zu professionellem Equipment
- Einsatz daher bevorzugt in Bereichen, wo viele Rover über lange Zeit verwendet werden sollen:
  - Langzeit-Monitoring in Geodäsie, Tagbau, Hochbau
  - Agrarbereich, autonomes Fahren in definierten Gebieten
  - Fahrzeugflotte zur Verifikation von Mautstrecken

Quellenverzeichnis



- Wikipedia GPS: de.wikipedia.org/wiki/GPS
- Wikipedia GNSS: de.wikipedia.org/wiki/Globales_Navigationssatellitensystem
- Wikipedia GPS Signals: en.wikipedia.org/wiki/GPS_signals
- Wikipedia Real Time Kinematic: en.wikipedia.org/wiki/Real_Time_Kinematic
- Navipedia RTK: www.navipedia.net/index.php/RTK_Fundamentals
- RTKLIB Homepage: www.rtklib.com
- Code auf GitHub: github.com/tomojitakasu/RTKLIB
- RasPiGNSS: drfasching.com/products/gnss/raspignss
- RasPiGNSS Software: drfasching.com/downloads



Vielen Dank für Ihre Aufmerksamkeit!

Sie können den Vortrag unter
http://glt16-programm.linuxtage.at/
bewerten.