ION GNSS+ 2014

Session C4: Software Receivers

GNSS-SDRLIB: An Open-Source and Real-Time GNSS Software Defined Radio Library

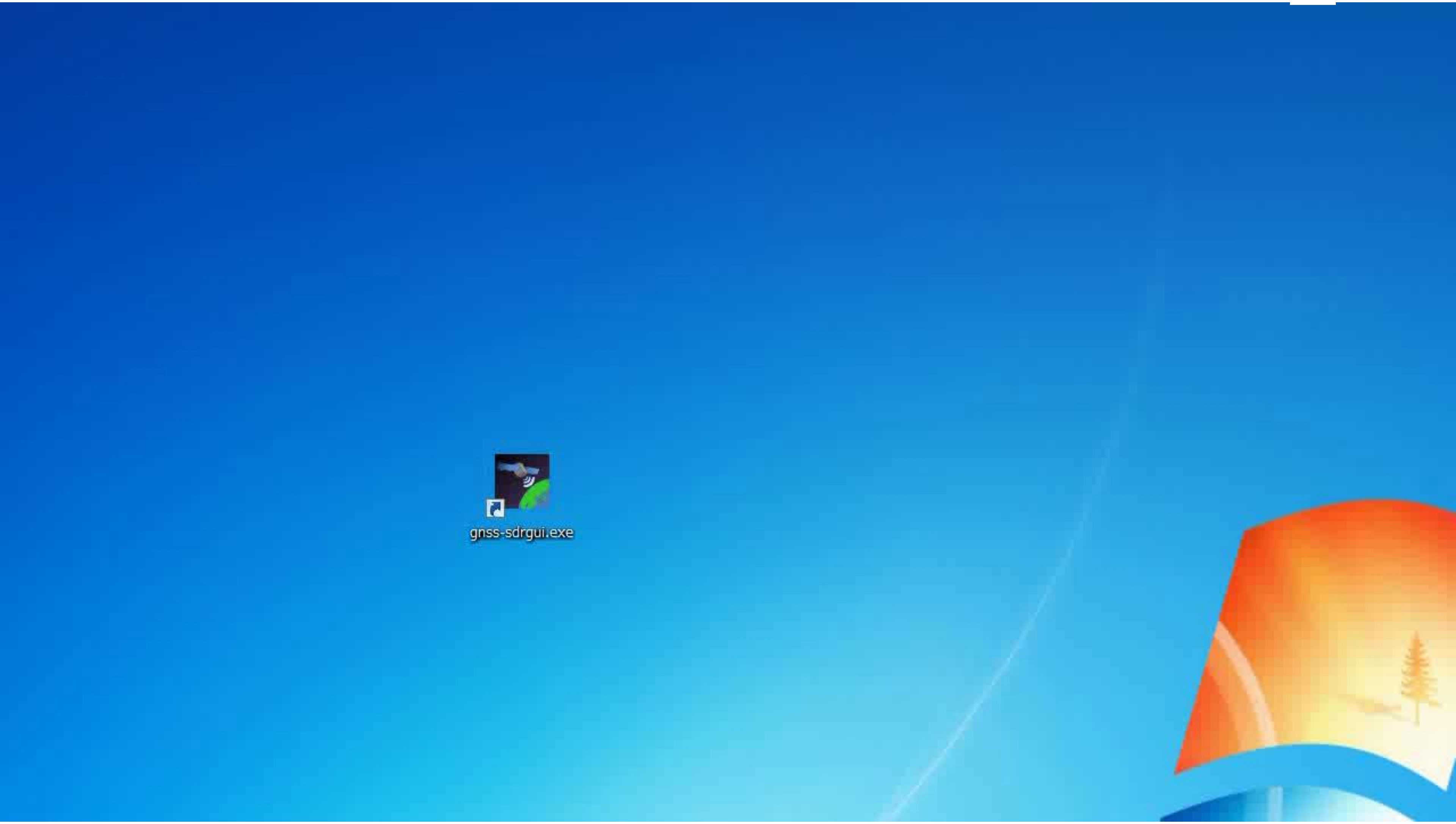
Taro Suzuki, Nobuaki Kubo

gnsssdrlib@gmail.com

Tokyo University Marine Science and Technology

GNSS-SDRLIB

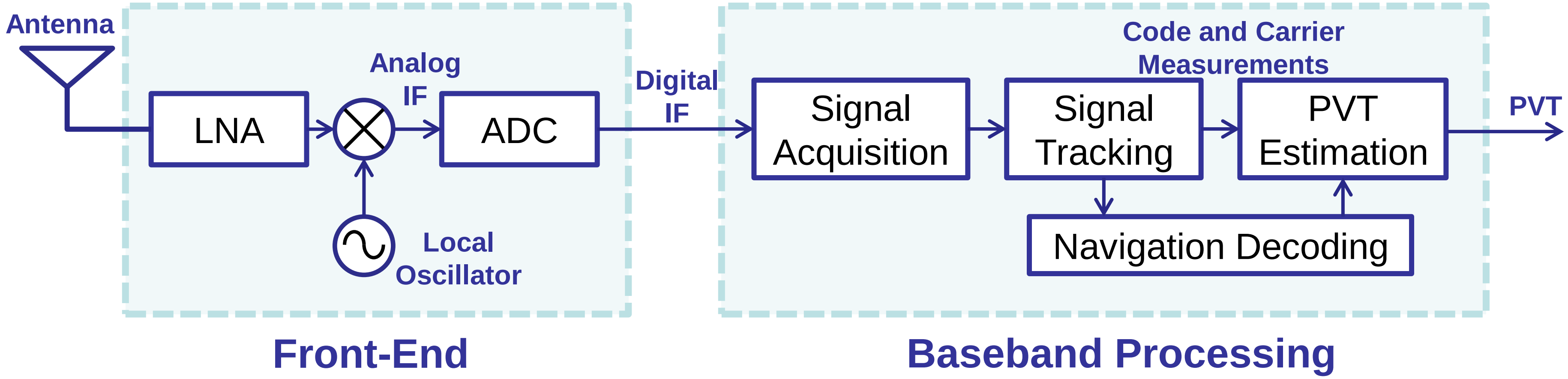
GNSS-SDRLIB http://www.taroz.net/gnsssdrlib_e.html
https://github.com/taroz/GNSS-SDRLIB



YouTube: **http://youtu.be/vVHOFs93vlU**

Why Software Receiver?

SDR（Software Defined Radio）



Advantage: Flexibility

◆Re-programmable and re-configurable

◆Developing new algorithms

◆Tracking new signal, Multipath mitigation method…

Software receiver is essential for cutting-edge research!

Open-Source GNSS Software Receivers

SoftGNSS
http://ccar.colorado.edu/gnss/
- MATLAB source codes, only for GPS L1 and post processing.

Fast GPS
http://sourceforge.net/projects/fastgps/
C++, only for GPS L1 and post processing

OpenSourceGPS
http://sourceforge.net/projects/osgps/
C++, only for GPS L1, real-time processing

GPS-SDR
https://github.com/gps-sdr
- C++, only for GPS L1, real-time processing

GNSS-SDR
https://code.google.com/p/gnsssdr/
- SCILAB, only for post processing, Multi-GNSS support (GPS, GLONASS, BeiDou L1)

GNSS-SDR
http://gnss-sdr.org/
- C++, Real-time Processing and Multi-GNSS support (GPS, Galileo L1)

Motivation

Motivation

- ◆ Study of GNSS receiver
- ◆ Developing new ultra-tightly coupled integration

↓

Developing my own software GNSS receiver from scratch!

Differentiation strategy

- ◆ **Cooperation with RTKLIB (http://www.rtklib.com/)**
  - Using RTKLIB functions as much as possible
- ◆ **Output of observation data**
  - Outputting RINEX, RTCM files as with hardware receivers
- ◆ **Real-time and Multi-GNSS support**
  - Support many consumer front-ends and all existing satellites

Feature of GNSS-SDRLIB

GNSS-SDRLIB

http://www.taroz.net/gnsssdrlib_e.html
https://github.com/taroz/GNSS-SDRLIB

Version 1.0 Beta, 2013 March
Version 1.0 , 2013 June
Version 2.0 Beta, 2014 June

- **GNSS signal processing functions written in C**
  - Code generation of all existing satellites
  - Signal acquisition / tracking functions
  - Decoding navigation messages
  - Pseudo-range / carrier phase measurements
- **GUI application written in C++/CLI**
- **Real-time positioning with RTKLIB**
- **Observation data can be outputted in RINEX or RTCM format**
- **Support multi-GNSS signals**
  - GPS, GLONASS, Galileo, BeiDou, QZSS L1 signals
  - Decoding QZSS SAIF/LEX message and SBAS message
- **Support RF binary file for post processing**
- **Support commercial front-ends for real-time positioning**

Support GNSS Front-end

<u>DVB-T dongle (RTL-2832U)</u>

・$10, Frequency: 24M-1.7GHz, Sampling: 2.56MHz

・Poor clock accuracy

<u>Nuand BladeRF (LMS6002D)</u>

・$420, Frequency: 300Hz~3.8GHz, Sampling: ~40Msps

・Tx function (transmitter)

<u>SiGe GN3S sampler V2/V3 (SiGe4120)</u>

・$450, Frequency: 1575.42MHz, Sampling: 4MHz

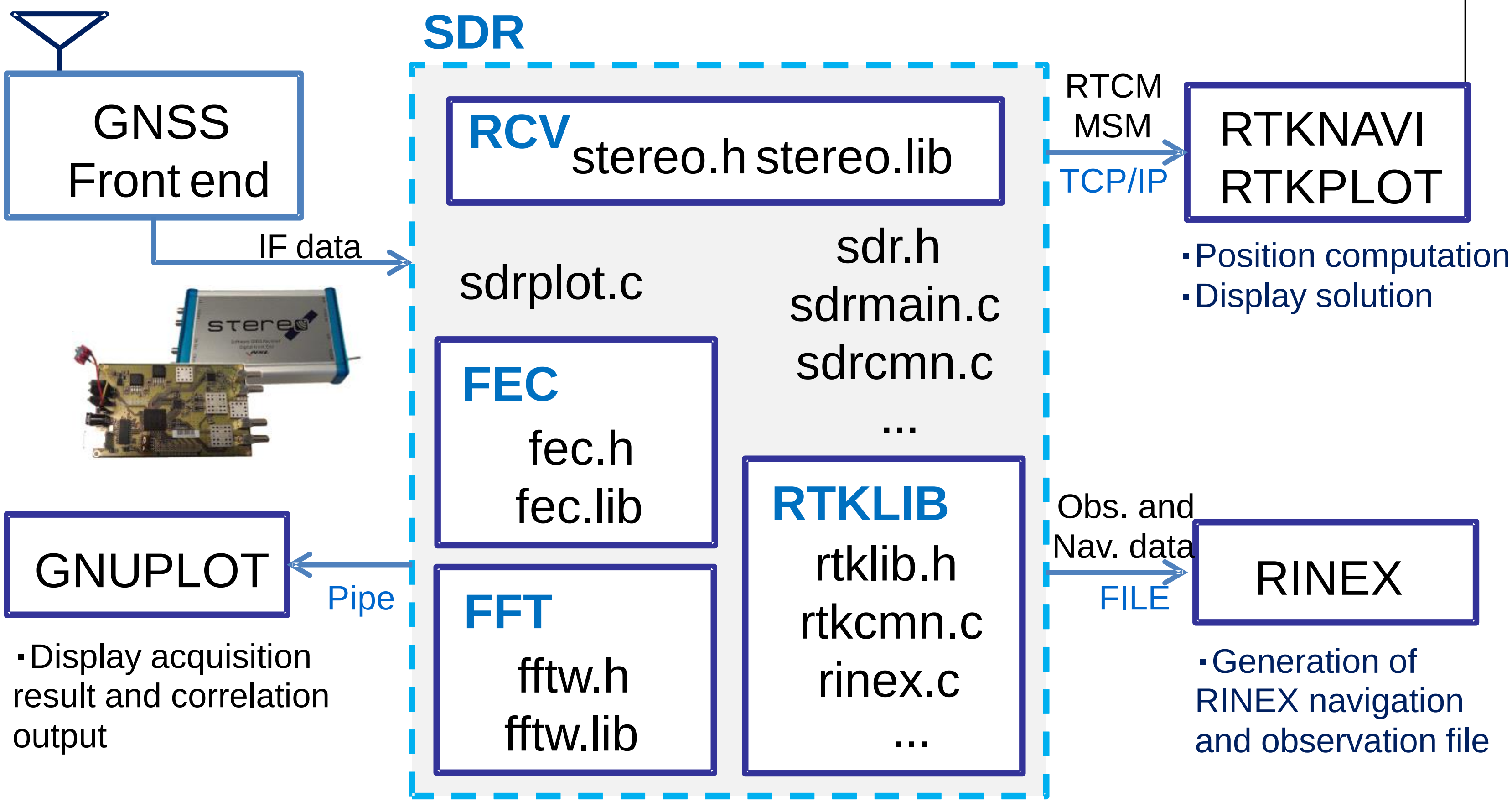
・For only GPS L1 signal

<u>NSL STEREO (MAX2769b+MAX2112)</u>

・$850, Frequency: 300Hz~3.8GHz, Sampling: ~40MHz

・Two front-ends

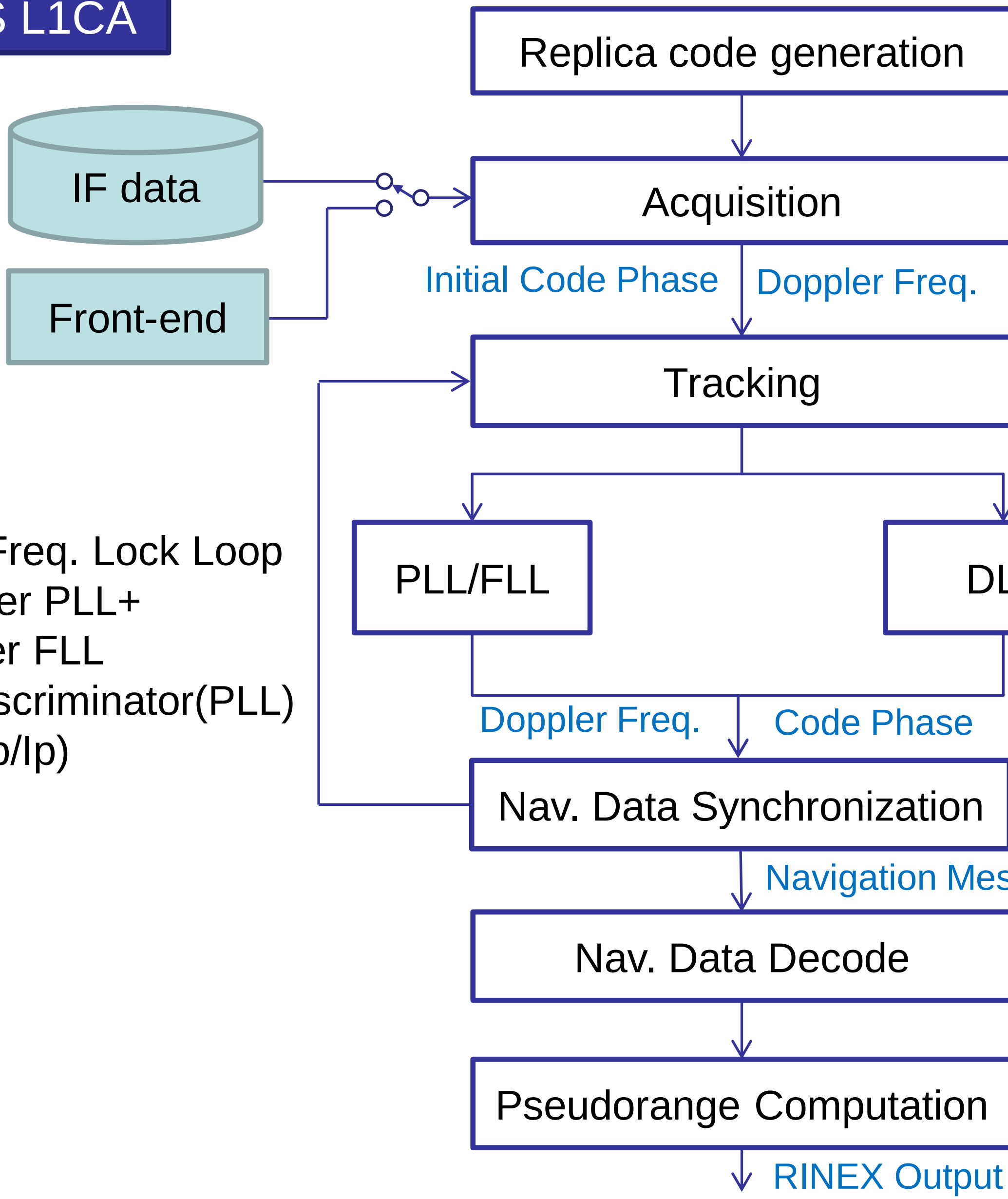
GNSS-SDRLIB Configuration



- Using FFT (Fast Fourier Transform) library
  - FFTW: http://www.fftw.org/)
- Using FEC (Forward Error Correction) library
  - http://www.ka9q.net/code/fec/
- Using RTKLIB
  - http://www.rtklib.com/

Acquisition and Tracking GNSS Signals

GPS L1CA



- Circular correlation by FFT
- Non-coherent integration
- Doppler search step: 200 Hz
- Doppler search range: -7~+7kHz

- Phase/Freq. Lock Loop
- 2nd order PLL+
  1st order FLL
- Error discriminator(PLL)
  atan(Qp/Ip)

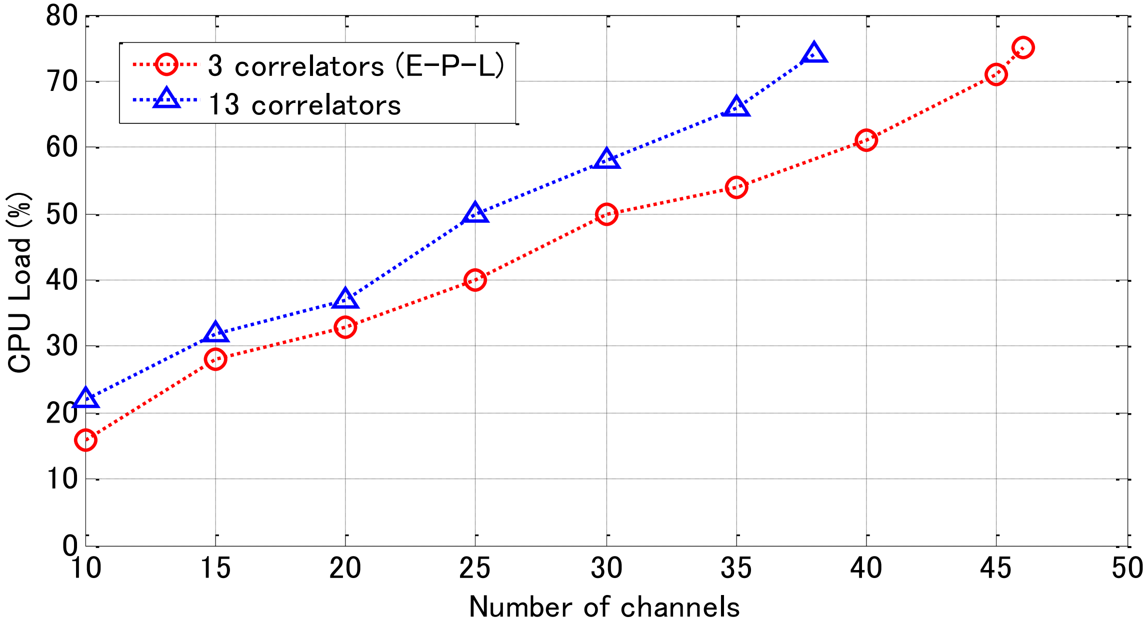
- Delay Lock Loop
- Carrier aid 2nd order DLL
- Error discriminator
  (E-L)/(E+L)/2

Real-time Processing Performance

How to speed up?

- **Signal Acquisition**
  - Dependent on FFT library
  - Currently, FFTW3.3.3, 64bit Single precision is used
- **Signal Tracking**
  - Using SIMD (Single Instruction Multiple Data) for correlation

Real-time Processing Performance（20Msps, Core i7-3770, SSE2）



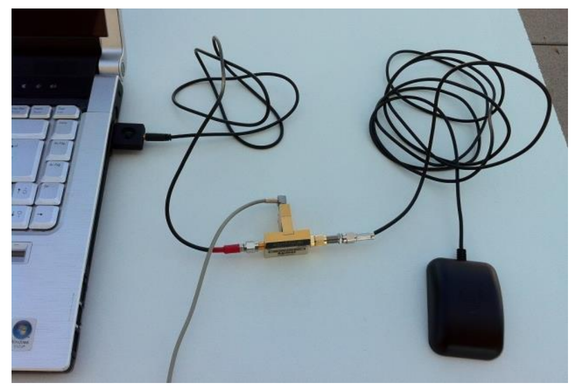
In the case of E-P-L correlator, it works with 45 channels!

Demonstration 1
Positioning with USB TV tuner Dongle ($10!)

RTL-SDR

- RTL-SDR
  - Most famous SDR front-end device
  - Using Elonics E4000 turnerchip
  - Using Realtek RTL2832U ADC
- Cheap (about $10~$20)
- Large community
  - http://sdr.osmocom.org/trac/wiki/rtl-sdr
- Active antenna cannot be used in default

Using GPS signal splitter and another GPS receiver



Using a bias-T network

We have to modify RTL-SDR Dongle to supply voltage into the antenna connecter

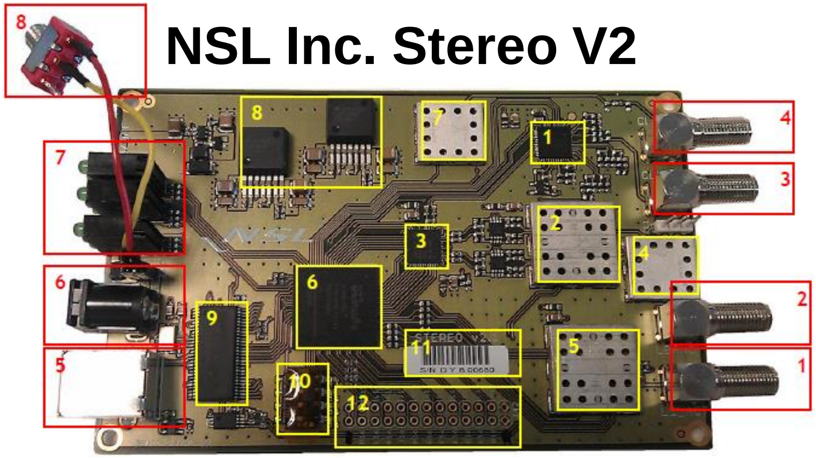
Positioning RTL-SDR Dongle

YouTube: http://youtu.be/Wv1h6MhwnBQ

Demonstration 2
Positioning with Multi-GNSS

Front-end for Multi-GNSS Constellation

NSL Inc. Stereo V2



£650

£800

http://www.nsl.eu.com/primo.html

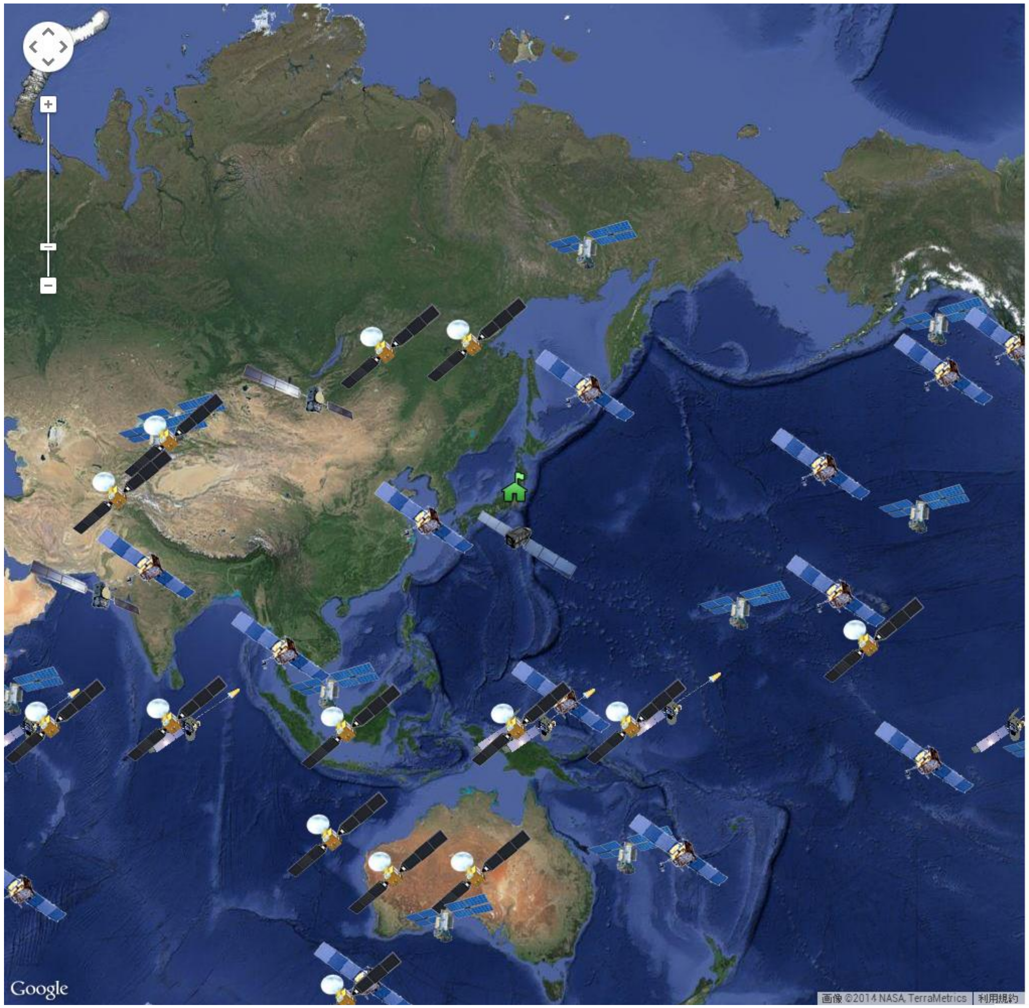
1. LMK03033C clock distribution chip
2. Maxim/Dallas MAX2112 RF front-end
   ⇒ For L Band (925 - 2175MHz)
3. Maxim/Dallas MAX19506 dual 8-bits ADC
4. MMIC amplifiers
5. Maxim/Dallas MAX2769B RF front-end
   ⇒ For L1 GNSS(1550 - 1610MHz)
6. Xilinx Spartan-6 FPGA
7. TXC 26MHz TCXO oscillator

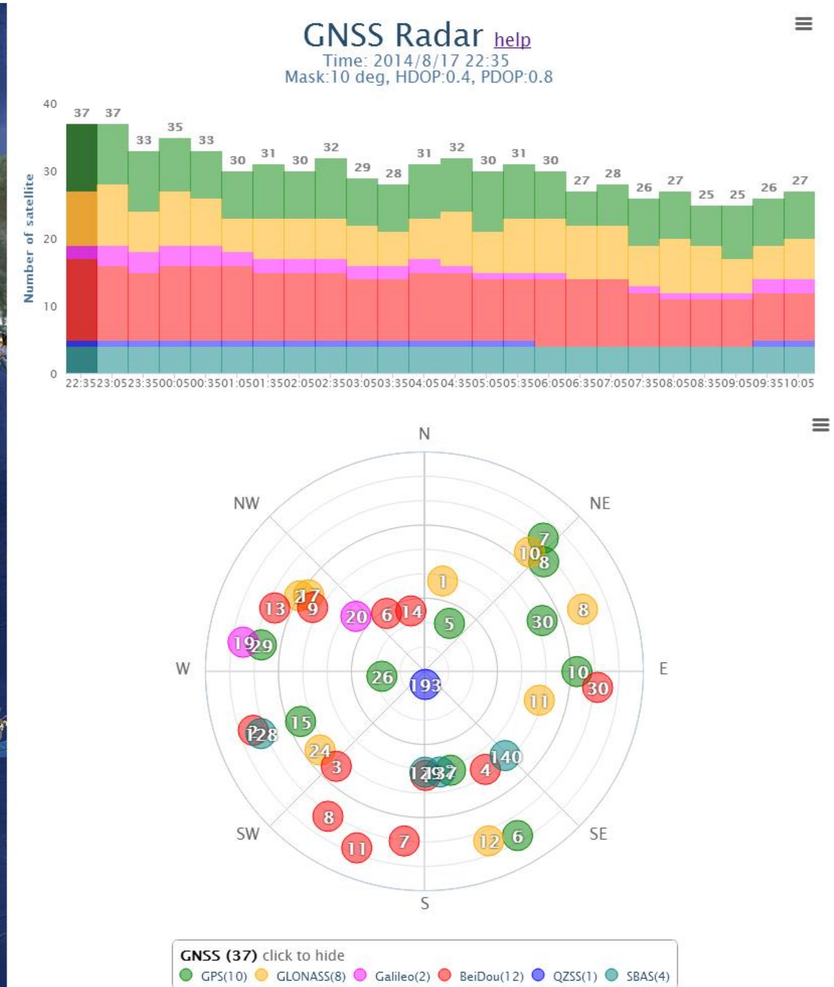
- USB 2.0 interface
- Simultaneously recording two front-end data
- All front-end setting is configurable
  - Center frequency
  - Bandwidth
  - Sample rate (8MHz~40MHz)

➡ Multi-GNSS signal tracking by two front-ends

How to Check Multi-GNSS Constellation?

GNSS-Radar: http://www.taroz.net/GNSS-Radar.html





Android version

https://play.google.com/store/apps/details?id=taroz.net.GNSS_Radar

Search “GNSS” in google play!

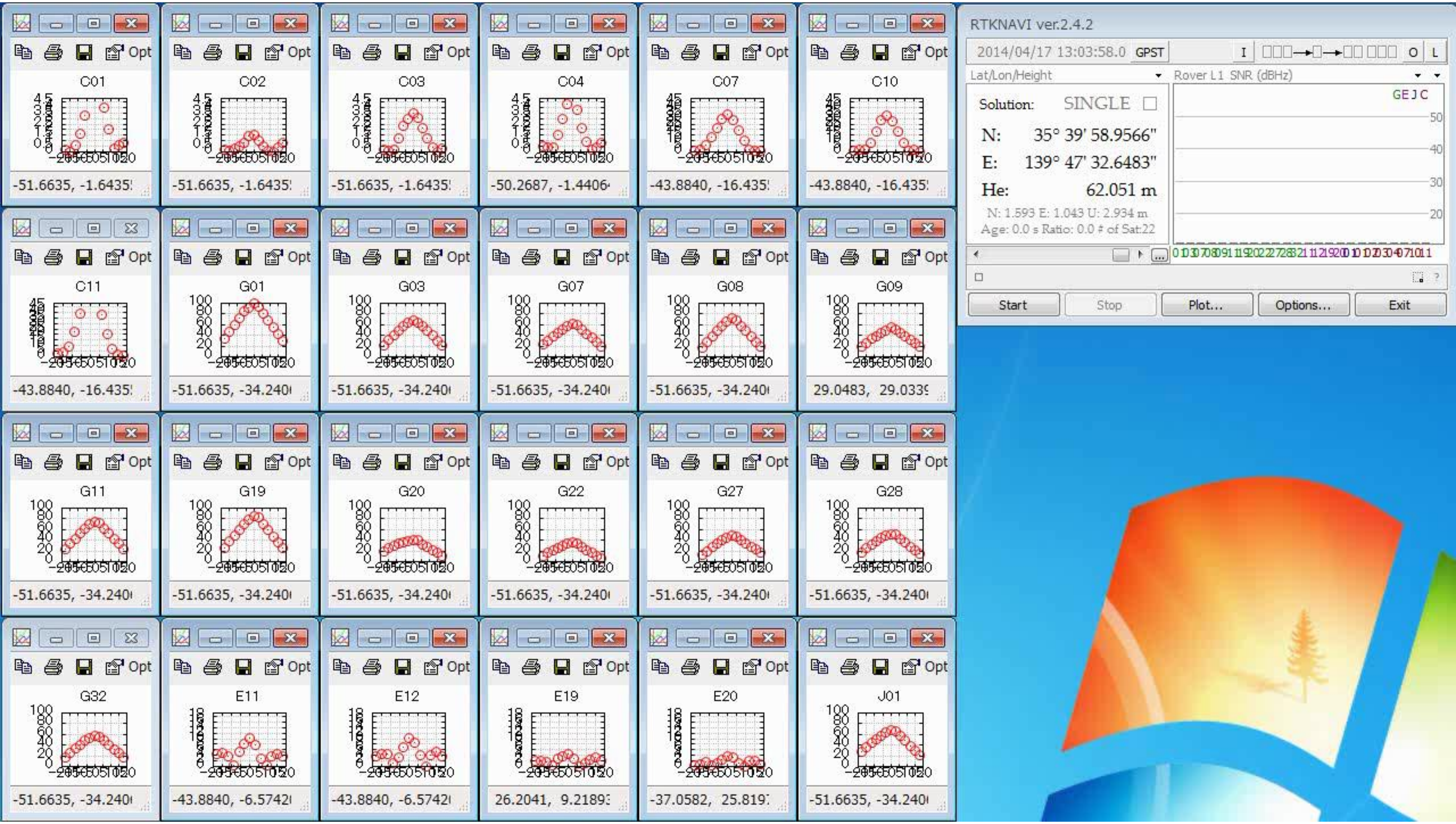
iOS version

https://itunes.apple.com/us/app/gnss-radar/id901597709

Search “GNSS” in iTunes Store!

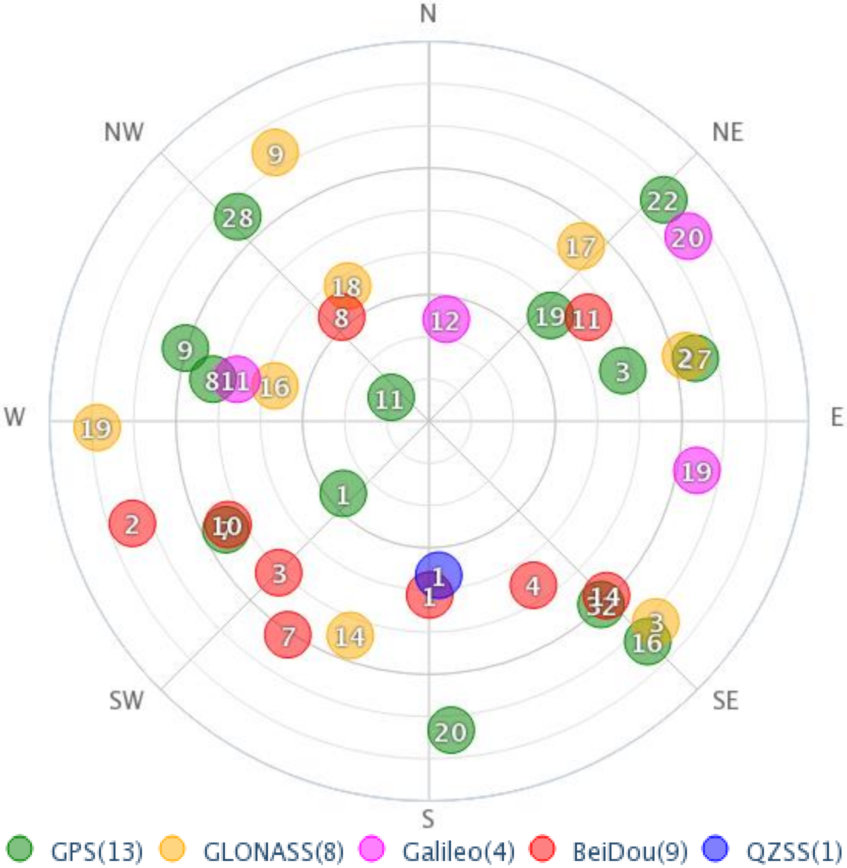
Demonstration of Multi-GNSS

2014/4/17, TUMSAT, GPS/QZS/Galileo/BeiDou (L1+B1), 20Msps

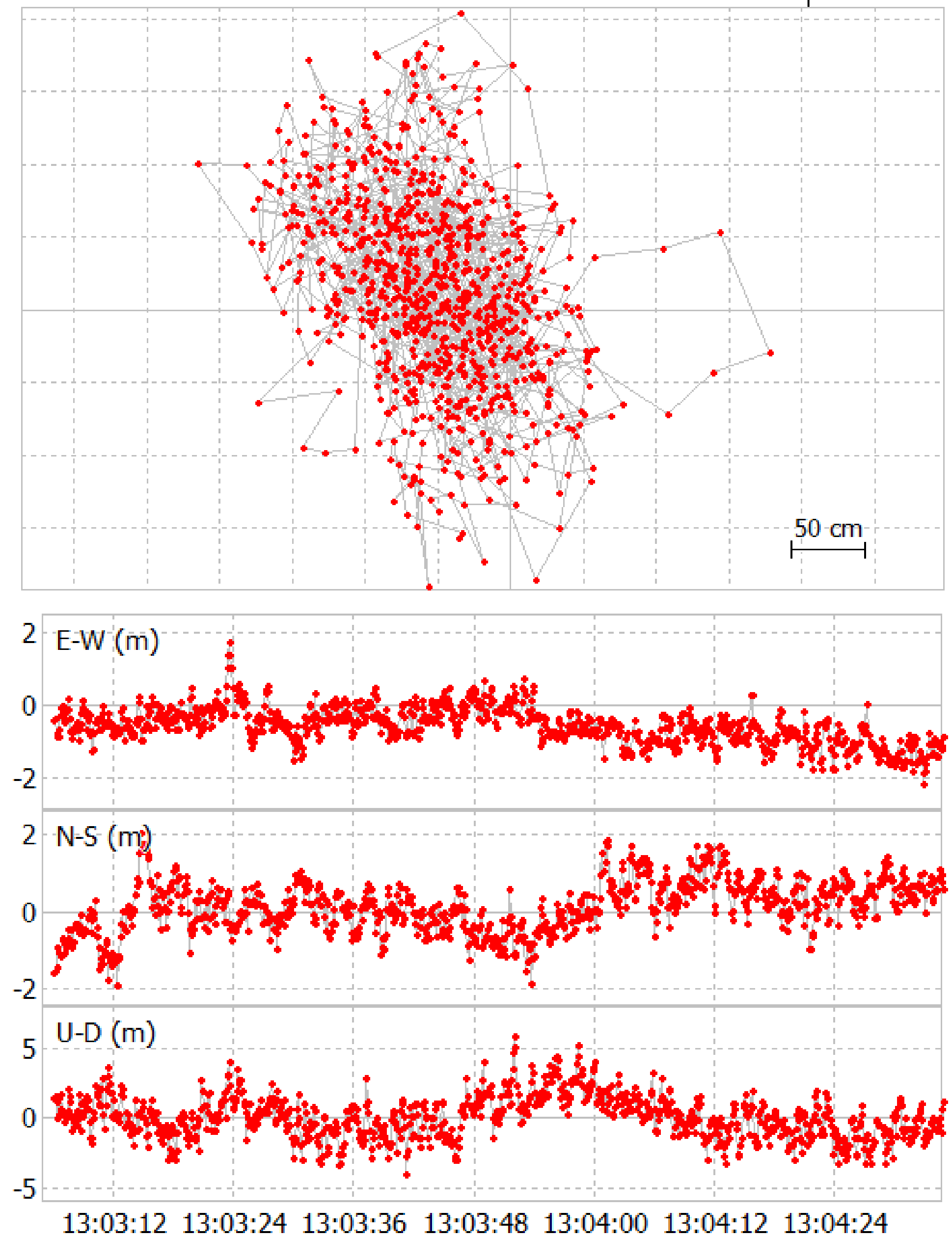


YouTube: http://youtu.be/N5tScnlQzkI

Demonstration of Multi-GNSS



NSAT: 25
HDOP: 0.7
X STD (1σ): 0.51m
Y STD (1σ): 0.66m
Z STD (1σ): 0.52m



Demonstration 3
PPP by QZSS LEX Message

QZSS LEX Message

LEX data structure

- 1 message = 2000 bits / 250 symbols, 2 kbps
- Length of overlaid code is 4 ms and it represents 1 symbol
- Using code shift keying (CSK) modulation
- Reed-Solomon error correction

2000 bits / 1s

Header (49 bits)	Data Part (1695 bits)	Reed-Solomon Code (256 bits)

MADOCA-LEX data structure

- Multi-GNSS real-time orbit/clock data (Currently, GPS/GLONASS/QZSS)
- Message format is based on RTCM SSR

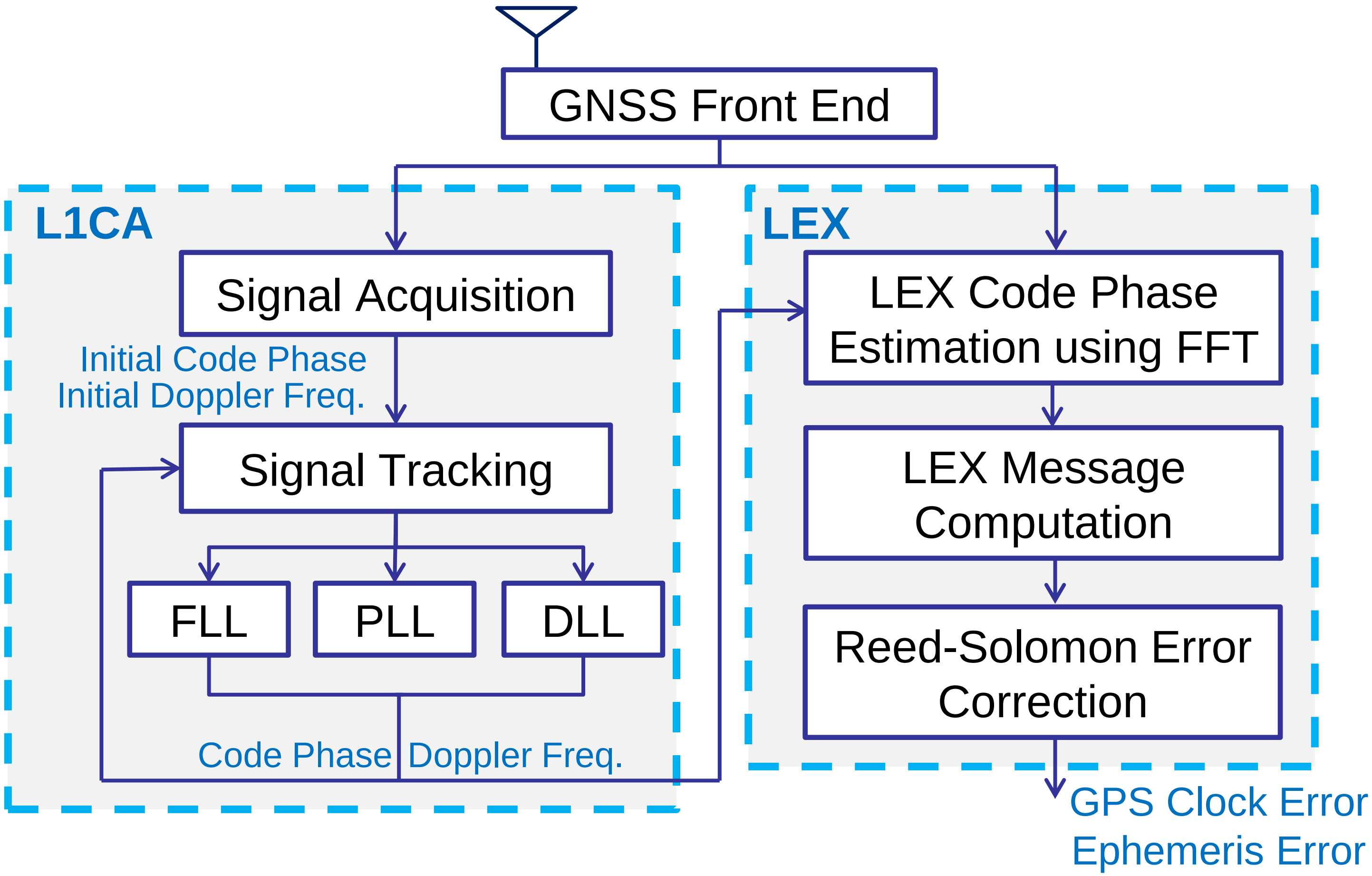
	SSR Message #				Update
	GPS	GLONASS	QZSS	Galileo	
Orbit	1057	1063	1246	1240	30 s
High rate clock	1062	1068	1251	1245	2 s

⇒ Decoding CSK modulated message using software GNSS receiver!

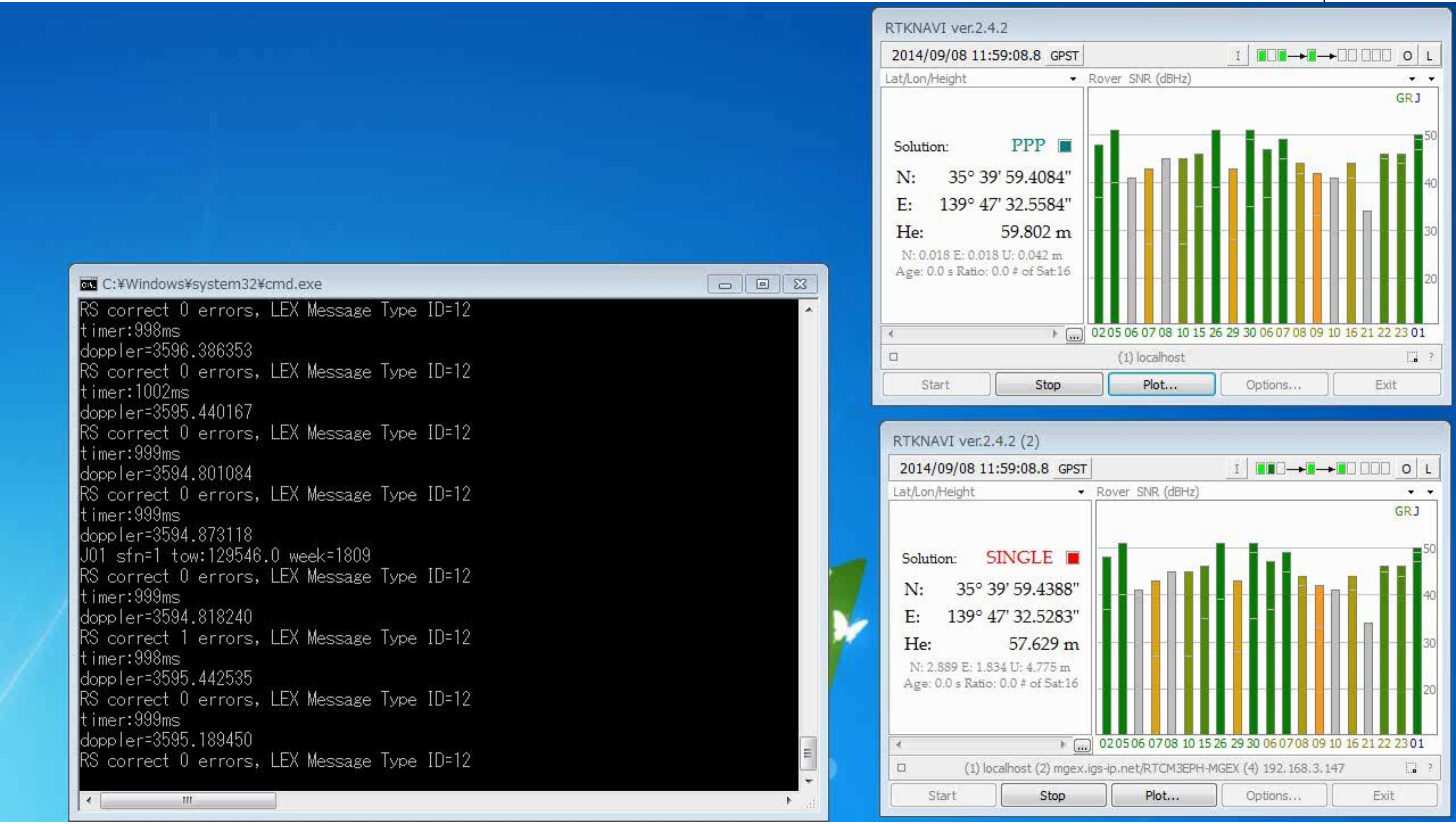
LEX Decoding Method

Approach

- LEX signal uses code shift keying (CSK) modulation
- L1CA Code phase and Doppler assistance for decoding LEX
- FFT based CSK modulation decoding

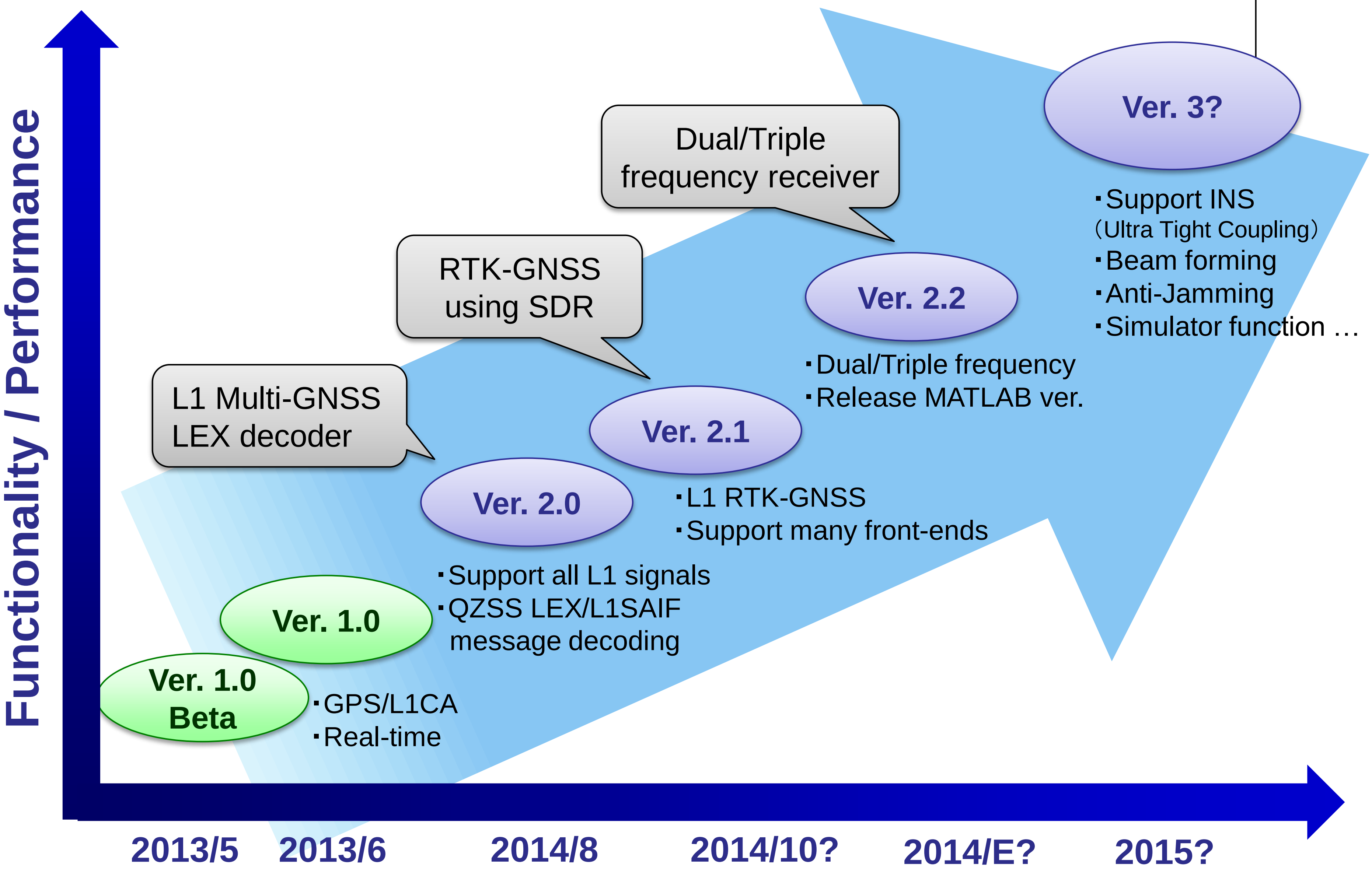


Real-time PPP using QZSS MADOCA-LEX



YouTube: http://youtu.be/VPaiF9s7cDs

GNSS-SDRLIB Roadmap



Summary

- Software GNSS receiver is essential for cutting-edge research

Developing my own software GNSS receiver from scratch!

Taro Suzuki, gnsssdrlib@gmail.com

Open-Source and Real-Time GNSS Software Defined Radio Library

GNSS-SDRLIB: **http://www.taroz.net/gnsssdrlib_e.html**
https://github.com/taroz/GNSS-SDRLIB

Tool for monitoring current GNSS consternation

GNSS-Radar: **http://www.taroz.net/GNSS-Radar.html**

Android version
https://play.google.com/store/apps/details?id=taroz.net.GNSS_Radar
Search "GNSS" in google play!

iOS version
https://itunes.apple.com/us/app/gnss-radar/id901597709
Search "GNSS" in iTunes Store!

Multi-GNSS Signals (1)

Around **L1 frequency** (1575.42 MHz)

GNSS	GPS/QZSS	QZSS		GALILEO		GLONASS	BeiDou
Service Name	C/A	L1C		E1		C/A (G1)	B1I
Center Freq.	1575.42 MHz	1575.42 MHz		1575.42 MHz		1602+ 0.5625K MHz	1561.098 MHz
Signal Component	Data	L1CD Data	L1CP Pilot	E1B Data	E1C Pilot	Data	Data
I/Q	Q	I	Q	I	Q	I	I
Band Width	2.046 MHz	4.096 MHz		24.552 MHz		1.002 MHz	2.046 MHz
Modulation	BPSK(1)	BOC(1,1)		CBOC(6,1,1/11)		BPSK	QPSK
Code Freq.	1.023 MHz	1.023 MHz		1.023 MHz		0.511 MHz	2.046 MHz
Code Chips	1023	10230		4092		511	2046
Code Length	1ms	10 ms	10 ms	4 ms	4 ms	1 ms	1 ms
Nav. Data	NAV	CNAV-2	-	I/NAV	-	NAV	D1/D2 NAV
Min. Received Power	-158.5 dBW	-163.0 dBW	-158.25 dBW	-163.0 dBW	-158.25 dBW	-161.0 dBW	-163.0 dBW

Multi-GNSS Signals (2)

Around **L2 frequency** (1227.60 MHz)

GNSS	GPS/QZSS		GLONASS
Service Name	L2C		C/A (G2)
Center Freq.	1227.60 MHz		1246+ 0.4375K MHz
Signal Component	L2CM Data	L2CL Pilot	Data
I/Q	I		I
Band Width	2.046 MHz		1.022 MHz
Modulation	BPSK		BPSK
Code Freq.	0.5115 MHz		0.511 MHz
Code Chips	10230	767250	511
Code Length	20 ms	1.5 s	1 ms
Nav. Data	CNAV	-	NAV
Min. Received Power	-160.0 dBW		-167.0 dBW

Multi-GNSS Signals (3)

Around **L5 frequency** (1176.45 MHz)

GNSS	GPS/QZSS		GALILEO				BeiDou
Service Name	L5		E5a		E5b		B2I
Center Freq.	1176.45MHz		1176.45MHz		1207.14MHz		1207.14 MHz
Signal Component	L5I Data	L5Q Pilot	E5aI Data	E5aQ Pilot	E5bI Data	E5bQ Pilot	B2I Data
I/Q	I	Q	I	Q	I	Q	I
Band Width	20.46 MHz		20.46 MHz		20.46 MHz		24.0 MHz
Modulation	BPSK(10)		BPSK(10)		BPSK(10)		BPSK(10)
Code Freq.	10.23 MHz		10.23 MHz		10.23 MHz		10.23 MHz
Code Chips	10230		10230		10230		10230
Code Length	1 ms	1 ms	1 ms	1 ms	1 ms	1 ms	1 ms
Nav. Data	CNAV	-	F/NAV	-	I/NAV	-	D1/D2 NAV
Min. Received Power	-157.9 dBW	-157.9 dBW	-155.0 dBW	-155.0 dBW	-155.0 dBW	-155.0 dBW	-163 dBW

Multi-GNSS Signals (4)

Navigation Message

Band	System	Signal	Nav. Type	Rate	Error Detection / Correction	Preamble bits	Secondary Code
L1	GPS/ QZS	L1CA	NAV	50 bps, 300 bits, 6 sec.	Hamming Code	8bit	-
		L1C	CNAV-2	100 bps, 1800 bits, 18 sec.	BCH+LDPC+Interleving	None	1800 bits
	GALILEO	E1	I/NAV	125 bps, 250 bits, 2 sec.	½Convolusion+Interleaving+CRC	10bit	25 bits (E1C)
	GLONASS	G1	NAV	50 bps, 100 bits, 2 sec.	Hamming Code	30bit	-
	BeiDou (MEO)	B1I	D1 NAV	50 bps, 300 bits, 6 sec.	BCH+Interleaving	11bit	NH20
	BeiDou (GEO)	B1I	D2 NAV	500 bps, 300 bits, 0.6 sec.	BCH+Interleaving	11bit	-
	SBAS	L1	SBAS	250 bps, 250 bits, 1 sec.	½Convolusion	(8x3) bit Encoded	-
L2	GPS/QZS	L2C	CNAV	25 bps, 300 bits, 12 sec.	½Convolusion	8bit	-
	GLONASS	G2	NAV	50 bps, 100 bits, 2 sec.	Hamming Code	30bit	-
L5	GPS/ QZS	L5	CNAV	50 bps, 300 bits, 6 sec.	½Convolusion	8bit	NH10 (L5I), NH20 (L5Q)
	GALILEO	E5a	F/NAV	25 bps, 250 bits, 10 sec.	½Convolusion+Interleaving+CRC	10bit	20 bits (E5aI) 100 bits (E5aQ)
	GALILEO	E5b	I/NAV	125 bps, 250 bits, 2 sec.	½Convolusion+Interleaving+CRC	10bit	4 bits (E5bI) 100 bits (E5aQ)
	BeiDou (MEO)	B1I	D1 NAV	50 bps, 300 bits, 6 sec.	BCH+Interleaving	11bit	NH20
	BeiDou (GEO)	B1I	D2 NAV	500 bps, 300 bits, 0.6 sec.	BCH+Interleaving	11bit	-

Strategy for Using Multi-GNSS Signals (1)

L1 Frequency Signals（G1, E1, B1）

- Do acquisition and tracking as with the GPS L1CA
- Differences are …
  - Chip rate and chip length
    - ➡ Difference is only replica code generation
  - Modulation type (E1:BOC)
    - ➡ We need to change a part of tracking method
  - Navigation Message
    - ➡ Read ICD and implement it!

⬇

Not so difficult except for decoding navigation message!

Strategy for Using Multi-GNSS Signals (2)

Other Frequency Signals (G2, E5ab, B2I, L5, LEX…)

- If L1 code is tracked, the Doppler and code phase computation is aided by L1 information
  - Doppler2 = Freq2/Freq1*Doppler1
  - Cphase2 = Cphase1 + (DCB)

  ➡ **Acquisition is not necessary**

- No need to decode navigation data
  - But time information is useful
  - Additional information in another navigation message

⬇

Only tracking loop is needed to generate pseudorange and carrier phase

How to Use GNSS-Radar

Source Code: https://github.com/taroz/GNSS-Radar

Options:

Set the observer location by latitude and longitude (the unit is degree)

ULR+?lat=xxx&lon=xxx (default: lat=35.7&lon=139.8 (Tokyo))

e.g. http://www.taroz.net/GNSS-Radar.html?lat=-37.8&lon=145

Set the elevation mask angle when computing the sky plot (the unit is degree)

ULR+?elemask=xxx (default: elemask=10)

e.g. http://www.taroz.net/GNSS-Radar.html?elemask=45

Set the time offset when computing the sky plot (the unit is hour)

ULR+?offhr=xxx (default: offhr=0)

e.g. http://www.taroz.net/GNSS-Radar.html?offhr=12

Set the time interval when computing the sky plot (the unit is minutes)

ULR+?tint=xxx (default: tint=30)

e.g. http://www.taroz.net/GNSS-Radar.html?tint=5

Set the number of times when computing the sky plot

ULR+?ntimes=xxx (default: tint=24, 24*30min=12hour)

e.g. http://www.taroz.net/GNSS-Radar.html?ntimes=48

Android version

https://play.google.com/store/apps/details?id=taroz.net.GNSS_Radar

Search “GNSS” in google play!

iOS version

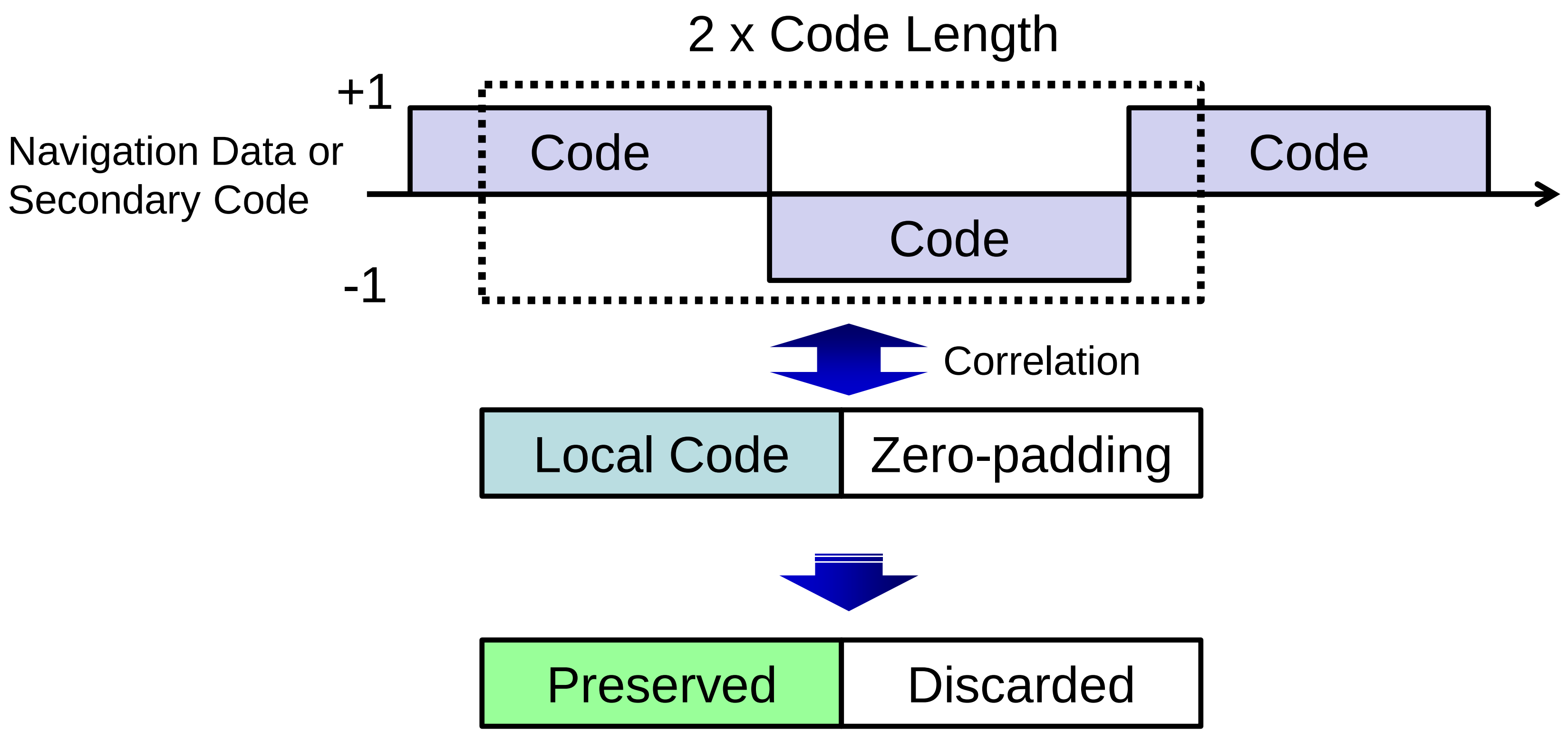
https://itunes.apple.com/us/app/gnss-radar/id901597709

Search “GNSS” in iTunes Store!

Acquisition Method

Circular correlation by Zero-padding FFT

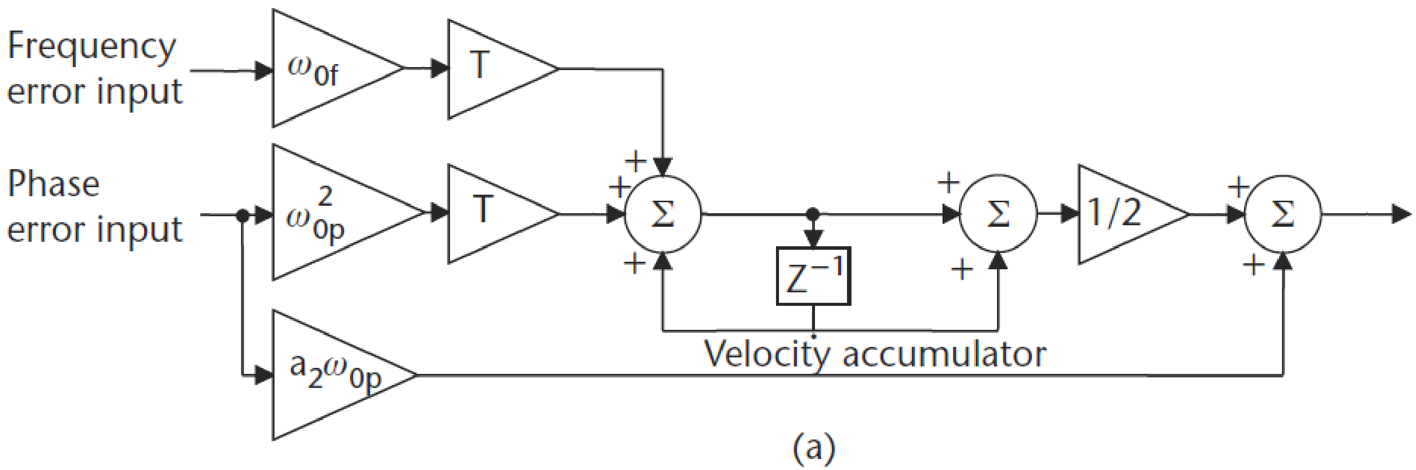
Ziedan, N. I., and Garrison, J. L. "Unaided acquisition of weak GPS signals using circular correlation or double-block zero padding"



- Perfect correlation can be obtained when navigation bit is changed
- Computational cost is doubled

Tracking Method

- 2^{nd} order PLL with 1^{st} order FLL



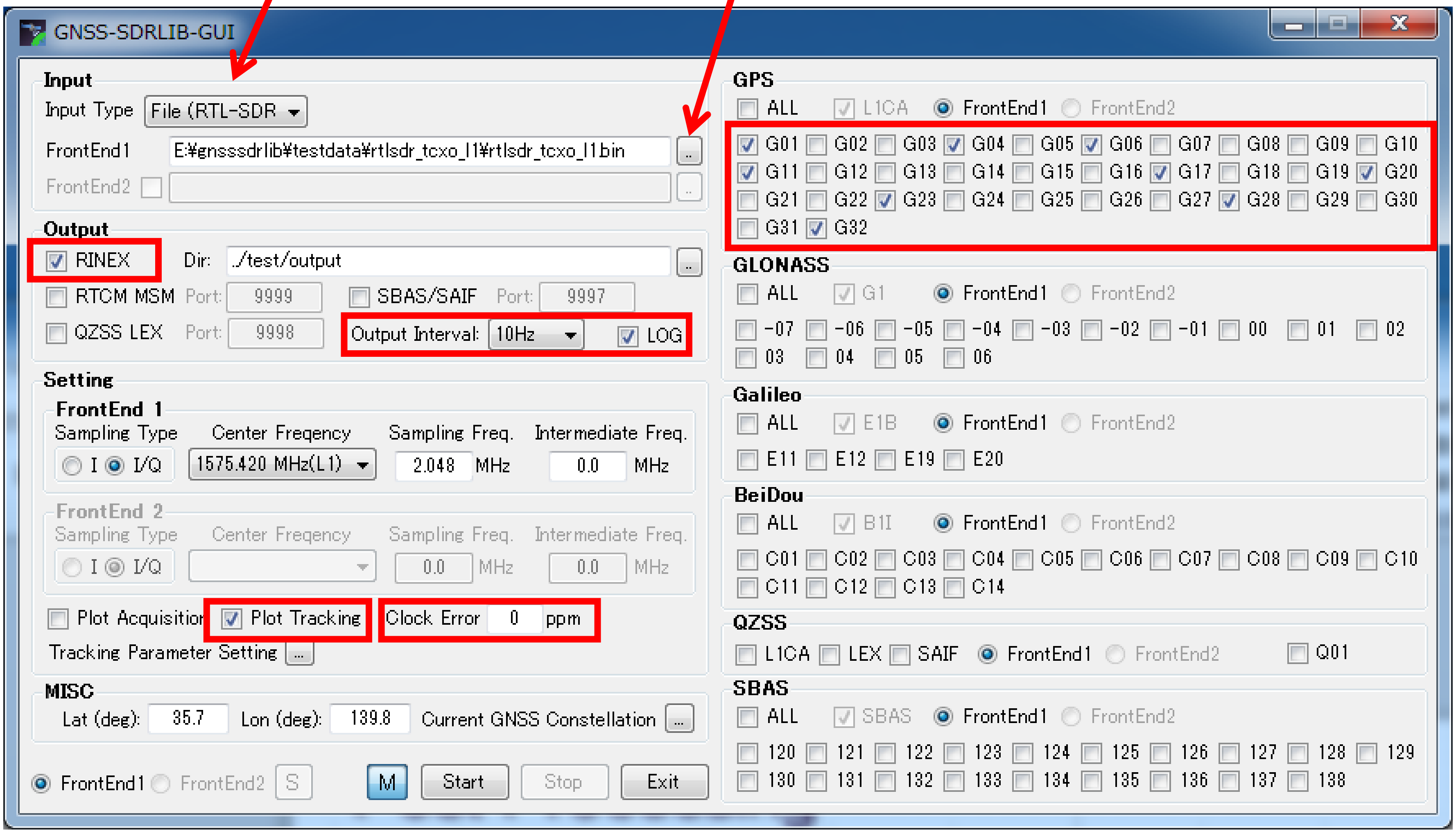
Elliott D. Kaplan, Christopher Hegarty , Understanding GPS: Principles And Applications

- Carrier aided 2^{nd} order DLL
- E-P-L correlator / Multi correlator
- Integration correlation result (4ms~20ms)
- Display correlation output in real-time

GUI Application of GNSS-SDRLIB

Select Input Tyepe

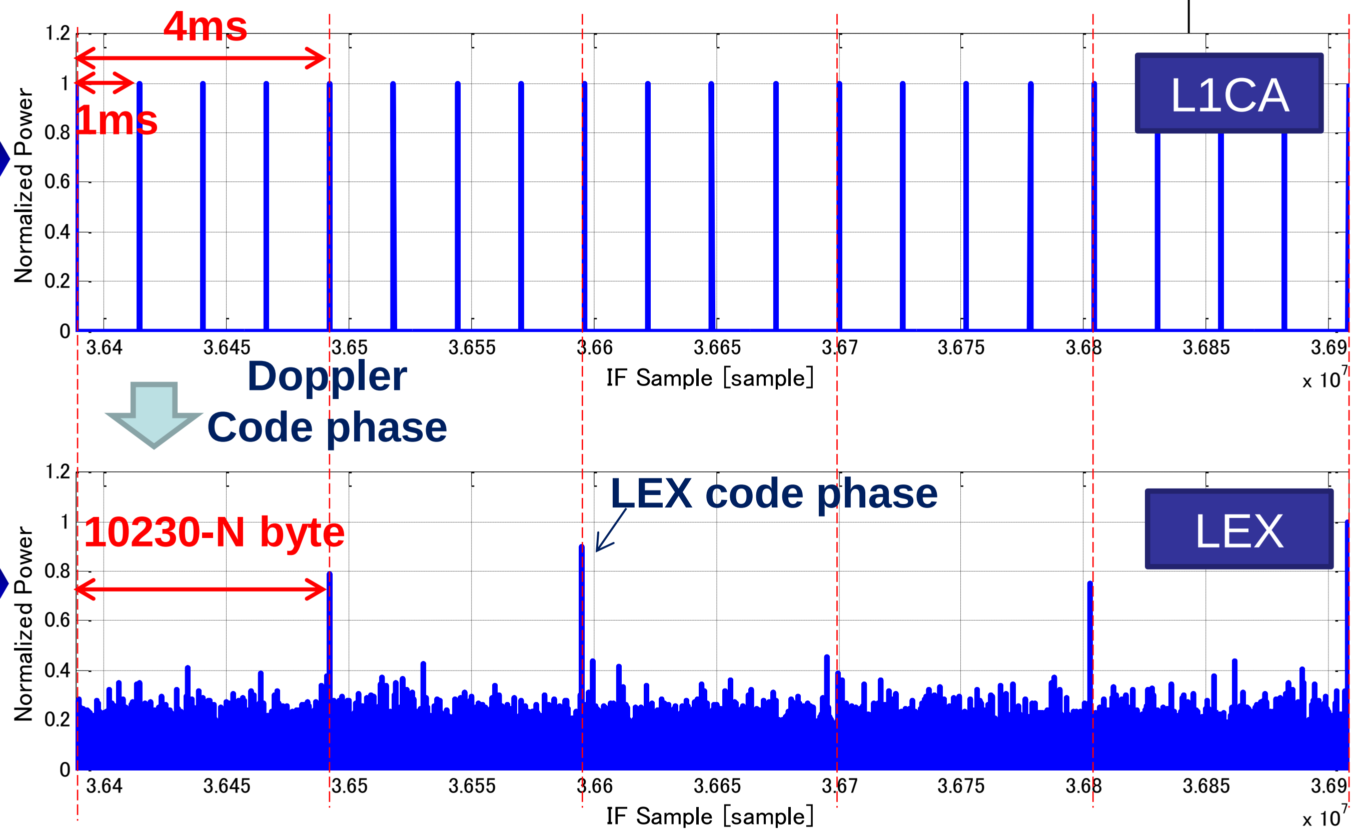
Select IF data for post processing



LEX Decoding Example

The estimated L1CA code phases

Computing the LEX message symbol by taking the difference between the L1CA code phase and LEX code phase

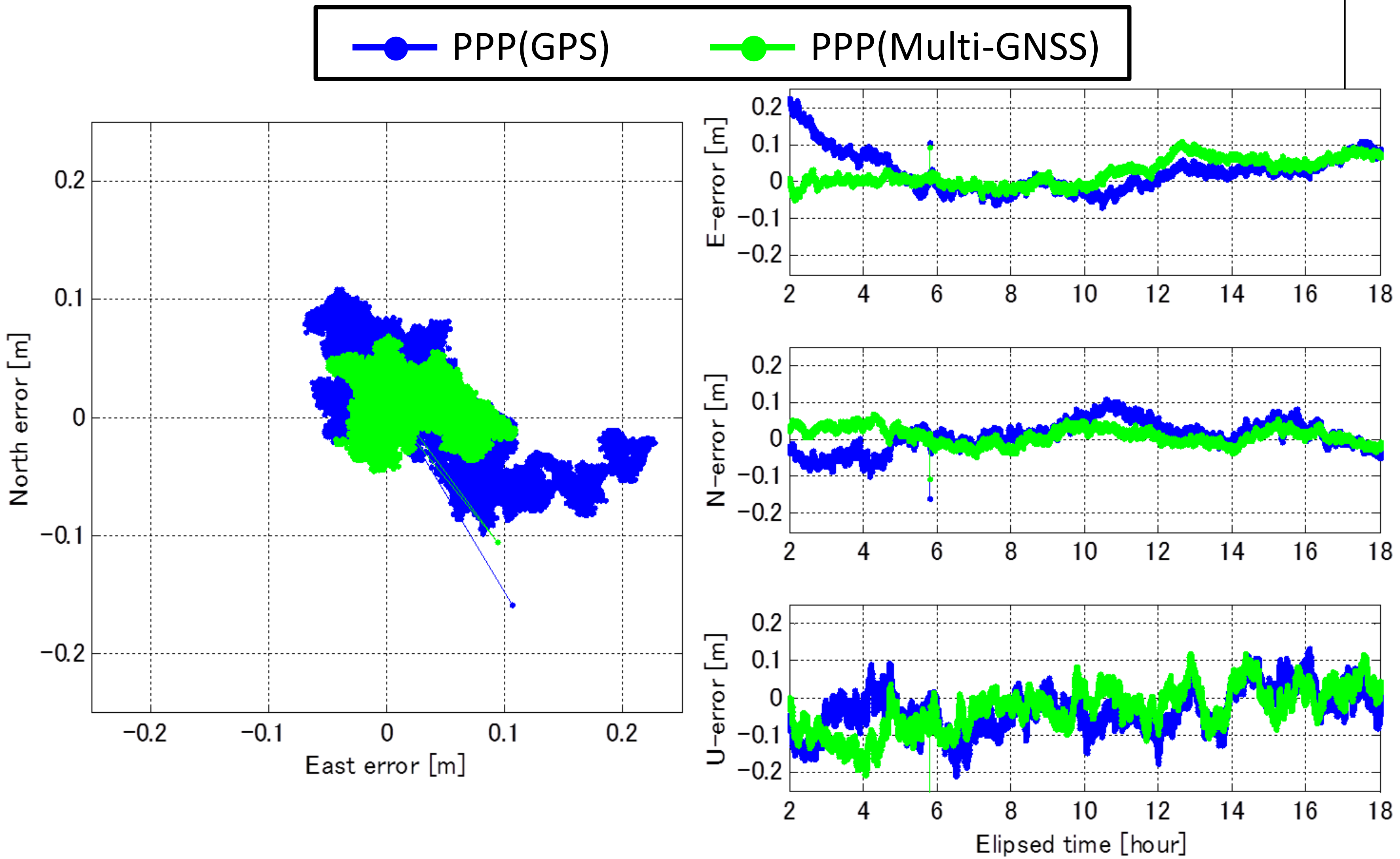


○ No need to track LEX ranging code (no DLL, PLL, etc.)

○ Easy implementation

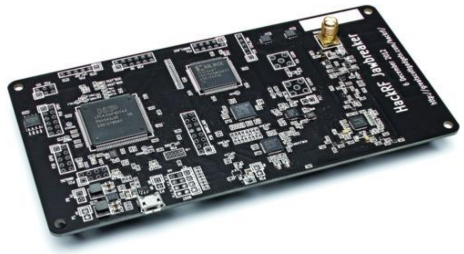
△ Computational cost

MADOCA-LEX PPP Result



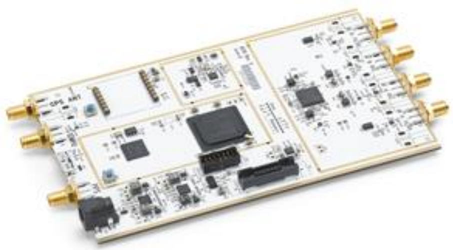
Horizontal error (18H)	SD	RMS	MAX.
GPS	3.9 cm	7.1 cm	22.7 cm
GPS+QZS+GLO	2.2 cm	5.0 cm	14.1 cm

GNSS Front-ends



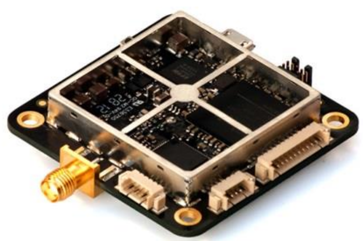
HackRF (LMS6002D)

- $300, Frequency: 30M-6GHz, Sampling: 20MHz
- Kick Starter project



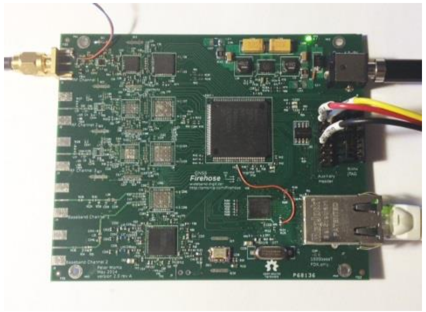
Ettus USRP (AD9361)

- $1100, Frequency: 300~3.8GHz, Sampling: 40Msps
- Two front-ends
- Tx function (transmitter)



SwiftNav Piksi (MAX2769)

- $525, Frequency: 1575.42MHz, Sampling: 16Msps
- For only GPS L1 signal
- RTK GPS enable? (FPGA based)



GNSS Firehose (MAX2112)

- $?, Frequency: 300Hz~3.8GHz, Sampling: ~40MHz
- Three front-ends
- Open Source Project https://github.com/pmonta/GNSS_Firehose

Which is Best?