

測位航法学会 2021年度 次世代高精度衛星測位研究委員会

An Open Source GNSS SDR: Development and Applications



Tokyo Univ. of Marine Science and Technology (TUMSAT)

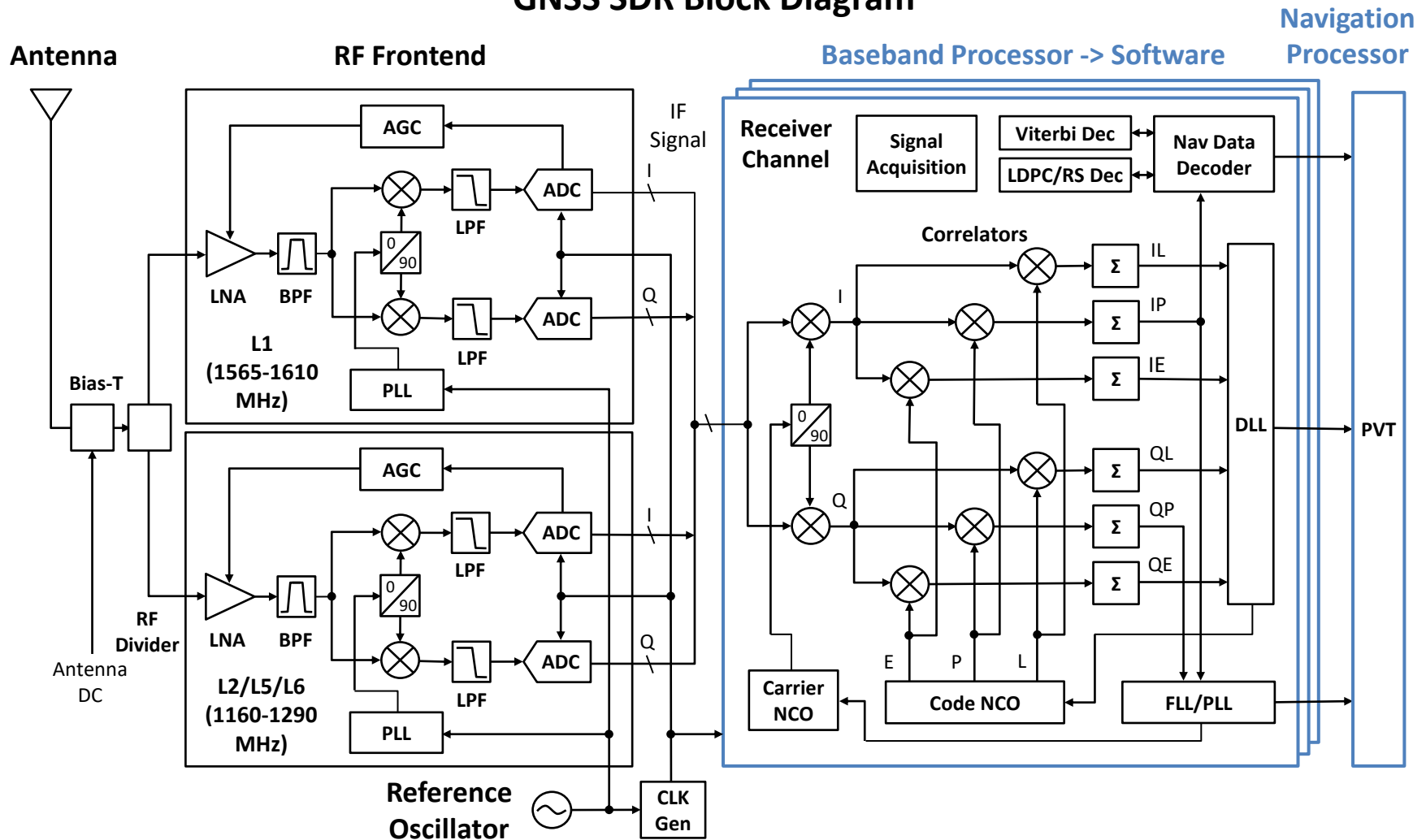
Tomoji TAKASU

2022-02-21 @Online, Japan

Pocket SDR

GNSS Software Defined Receiver

GNSS SDR Block Diagram



GNSS SDR RF Frontends

RF Frontend	Vendor	# of CH	RF Band	ADC	RF LSI	FPGA	Host I/F
USRP N210 ^[1]	Ettus Research	2CH (RX) 2CH (TX)	DC ~ 6 GHz	14 bit ~ 100 Msps	Custom Daughterboard	Spartan 3A- DSP	Giga- Ethernet
SiGe GN3S Sampler v3 ^[2]	GNSS Lab Univ Colorado	1CH (RX)	L1	2 bit ~ 16 Msps	SiGe4120	-	USB 2.0
Stereo ^[3]	NSL	2CH (RX)	L1 + L2/L5/L6	2/3 bit 26 Msps	MAX2769B + MAX2112	Spartan-6	USB 2.0
Blade RF ^[4]	NuBand	2CH (RX) 2CH (TX)	300 MHz ~ 3.8 GHz	12 bit ~ 40 Msps	LMS6002D	Cyclone IV E	USB 3.0
Lime SDR ^[5]	Lime microsystems	2CH (RX) 2CH (TX)	100 kHz ~ 3.8 GHz	12 bit ~ 61 Msps	LMS7002M	Cyclone IV EP4CE40F23	USB 3.0
NUT4NT ^[6]	Amungo Navigation	4CH (RX)	L1/L2/ L5/L6	2 bit ~ 99 Msps	NT1065	Lattice ECP5	USB 3.0
Pocket SDR	-	2CH (RX)	L1 + L2/L5/L6	2 bit ~ 24 Msps	MAX2771 x 2	-	USB 2.0



USRP N210



GN3S Sampler



Stereo



Blade RF



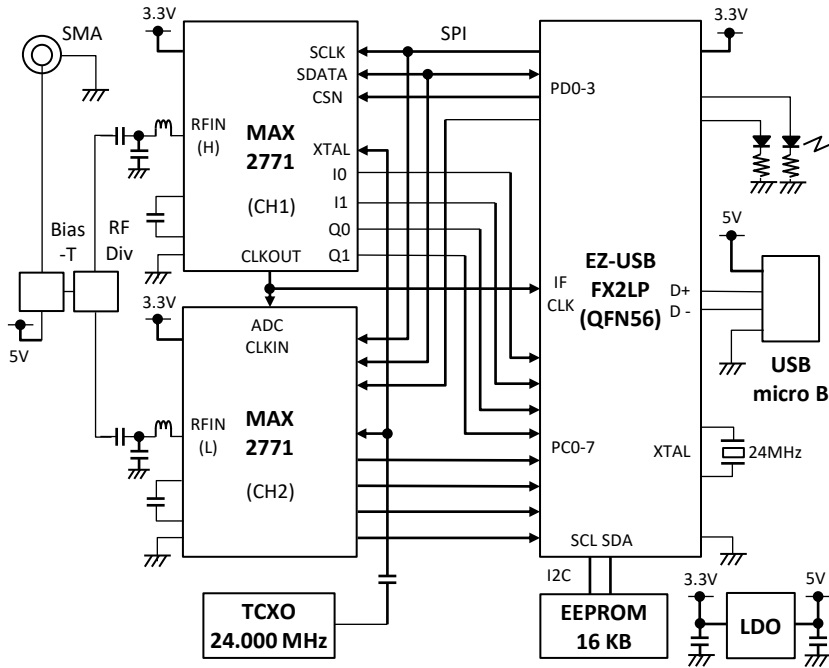
Lime SDR



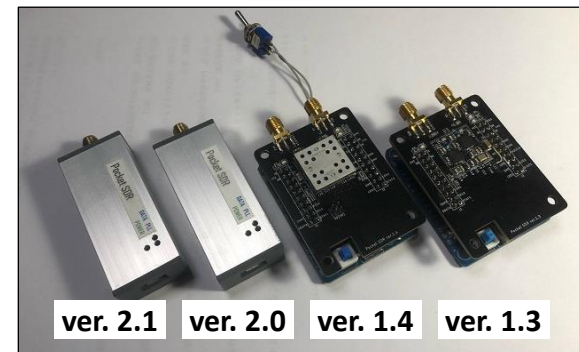
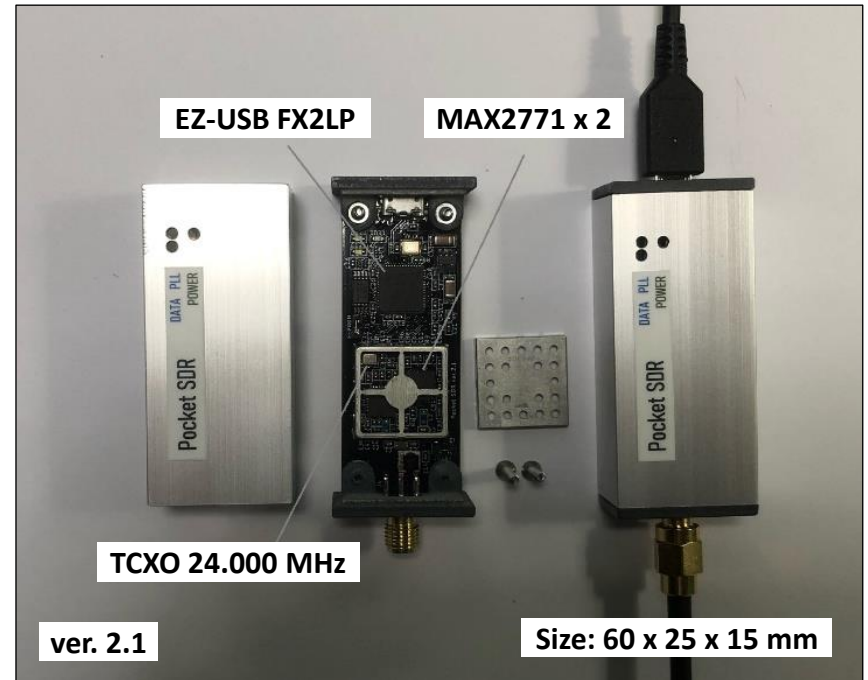
NUT4NT

[1] <https://www.ettus.com/all-products/un210-kit/>, [2] <https://ccar.colorado.edu/gnss/>, [3] <https://gmvnsl.com/advanced-gnss-hw-sw/>,
 [4] <https://www.nuand.com/bladerf-1/>, [5] <https://limemicro.com/products/boards/limesdr/>, [6] <https://www.amungo-navigation.com/>

Pocket SDR RF Frontend



LO Frequency : CH1 1525 ~ 1610 MHz (L1 band)
 : CH2 1160 ~ 1290 MHz (L2/L5/L6 band)
IF Bandwidth : 2 ~ 20 MHz
Sampling Rate : 4, 6, 8, 12, 16 or 24 Msps
Sampling Type : I or I/Q, 2 bits
Host I/F : USB 2.0, micro B
Power : 5V 140 mA, USB bus power
Parts Cost : ~ \$50



GNSS SDR Software

Software		GNSS-SDR	GNSS-SDRLIB	Pocket SDR
URL		https://gnss-sdr.org	https://github.com/taroz/GNSS-SDRLIB	https://github.com/tomojitakasu/PocketSDR
Current Release		v0.0.15 (Aug 2021)	v2.0 Beta (Dec 2014)	ver. 0.7 (Feb 2022)
GitHub Stars		☆ 966	☆ 299	☆ 48
Processing Mode		Post Processing, Real-time	Post Processing, Real-time	Post Processing
GNSS Signals	GPS	L1C/A, L2C, L5	L1C/A	L1C/A, L1C, L2C, L5
	GLONASS	L1C/A, L2C/A	L1C/A	L1C/A, L2C/A, L3OC
	Galileo	E1B/C, E5a	E1B	E1B/C, E5a, E5b, E6B/C
	QZSS	-	L1C/A, L1SAIF, LEX	L1C/A, L1C/B, L1C, L1S, L2C, L5, L5S, L6D/E
	BDS	B1I, B3I	B1I	B1I, B1C, B2I, B2a, B2b, B3I
	NavIC	-	-	L5-SPS
	SBAS	-	L1C/A	L1C/A, L5
Supported RF Frontends		USRP, UmTRX, HackRF One, Blade RF, RTL-SDR	SiGe GN3S sampler v2/v3, Stereo, Blade RF, RTL-SDR	Pocket SDR RF Frontend
OS		Windows, Linux, macOS, others	Windows, Linux	Windows, Linux
Language		C++, C	C, C++/CLI	Python 3, C
External Library		GNU Radio, Boost, FFTW3, VOLK, Armadillo, LAPACK/BLAS, glog, gflags, matio, pugiXML, Protocol Buffers, OpenSSL, RTKLIB	FFTW3, LIBFEC, RTKLIB, libusb	FFTW3, LIBFEC, LDPC-codes, RTKLIB, CyAPI or libusb-1.0

Pocket SDR Software

RF Frontend Utilities

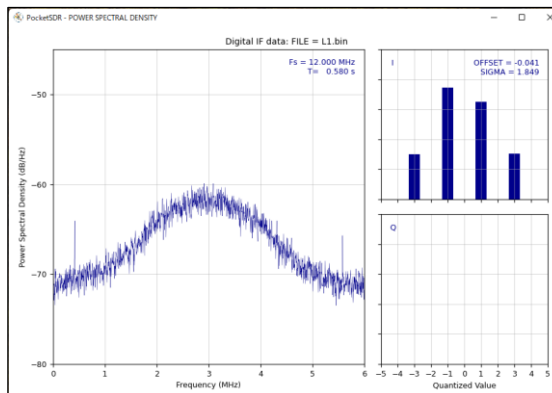
- `pocket_conf` : Device Configurator for RF Frontend
- `pocket_dump` : IF Data Sampler for RF Frontend

GNSS SDR Software

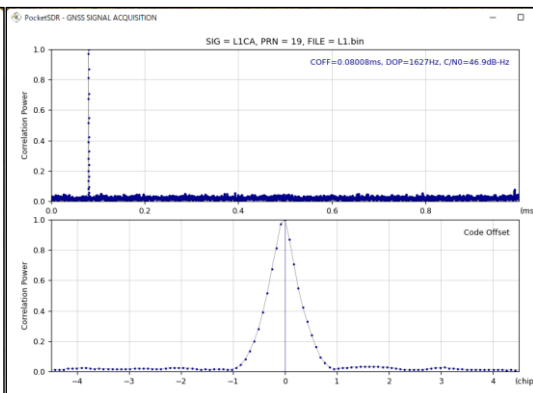
- `pocket_psd.py` : PSD Analysis of Sampling Data
- `pocket_acq.py` : GNSS Signal Search and Acquisition
- `pocket_trk.py` : GNSS Signal Tracking and Navigation Data Decoding
- `pocket_snap.py` : Snapshot Positioning with Sampling Data

Environment

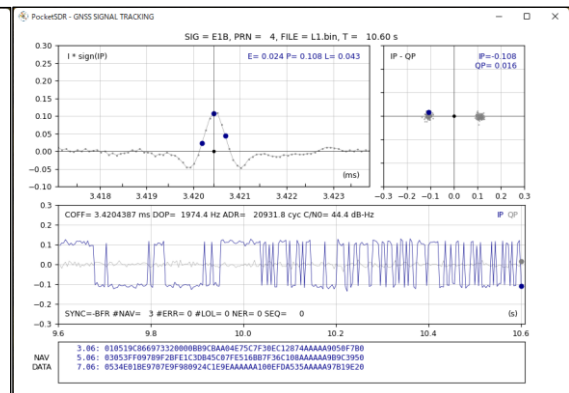
- Windows or Linux, Python 3 (Numpy, Scipy and matplotlib)



`pocket_psd.py`



`pocket_acq.py`



`pocket_trk.py`

Supported GNSS Signals

System	Freq. (MHz)	Signal	Modulation	Spreading Code		Nav Data
				Primary (chips)	Overlay (chips)	
GPS	1575.42	L1C/A	BPSK(1)	1023	-	LNAV
		L1C-D	BOC(1,1)	10230	-	CNAV-2
		L1C-P	TMBOC(6,1,4/33)	10230	1800	-
	1227.6	L2C-M	BPSK(1), TDM	10230	-	CNAV
		L2C-L		767250	-	-
	1176.45	L5-I	BPSK(10)	10230	10 (NH)	CNAV
		L5-Q	BPSK(10)	10230	20 (NH)	-
GLONASS	1602.0 + 0.5625K	L1C/A	BPSK(0.5)	511	10	GLO-STR
		L2C/A	BPSK(0.5)	511	10	GLO-STR
	1202.025	L3OCd	BPSK(10)	10230	10 (NH)	GLO-STR
		L3OCp	BPSK(10)	10230	5 (BC)	-
	1575.42	E1-B	CBOC(6,1,1/11)	4092	-	I/NAV
		E1-C	CBOC(6,1,1/11)	4092	25	-
Galileo	1176.45	E5a-I	BPSK(10)	10230	20	F/NAV
		E5a-Q	BPSK(10)	10230	100	-
	1207.14	E5b-I	BPSK(10)	10230	4	I/NAV
		E5b-Q	BPSK(10)	10230	100	-
	1278.75	E6-B	BPSK(5)	5115	-	C/NAV
		E6-C	BPSK(5)	5115	100	-
QZSS	1575.42	L1C/A	BPSK(1)	1023	-	LNAV
		L1C/B	BOC(1,1)	1023	-	LNAV
		L1C-D	BOC(1,1)	10230	-	CNAV-2
		L1C-P	TMBOC(6,1,4/33)	10230	1800	-
		L1S	BPSK(1)	1023	-	L1S
System	Freq. (MHz)	Signal	Modulation	Spreading Code		Nav Data
				Primary (chips)	Overlay (chips)	
QZSS (cont.)	1227.6	L2C-M	BPSK(1), TDM	10230	-	CNAV
		L2C-L		767250	-	-
	1176.45	L5-I	BPSK(10)	10230	10 (NH)	CNAV
		L5-Q	BPSK(10)	10230	20 (NH)	-
		L5S-I	BPSK(10)	10230	10 (NH)	L5S
		L5S-Q	BPSK(10)	10230	20 (NH)	-
	1278.75	L6L	BPSK(5), TDM	1048575	-	-
		L6D		10230	-	CLAS
		L6E		10230	-	MADOCA
BDS	1561.098	B1I	BPSK(2)	2046	20 (NH)	D1
					-	D2
	1575.42	B1C-D	BOC(1,1)	10230	-	B-CNAV1
		B1C-P	QMBOC(6,1,1/11)	10230	1800	-
	1207.14	B2I	BPSK(2)	2046	20 (NH)	D1
					-	D2
	1176.45	B2a-D	BPSK(10)	10230	5	B-CNAV2
		B2a-P	BPSK(10)	10230	100	-
	1207.14	B2b-I	BPSK(10)	10230	-	B-CNAV3
					-	B2b-PPP
NavIC	1176.45	L5-SPS	BPSK(10)	10230	-	IRN-NAV
	2492.028	S-SPS	BPSK(10)	10230	-	IRN-NAV
SBAS	1575.42	L1C/A	BPSK(1)	1023	-	SBAS
	1176.45	L5-I	BPSK(10)	10230	10 (NH)	L5 SBAS
		L5-Q	BPSK(10)	10230	20 (NH)	-

Red: supported by pocket_acq.py and pocket_trk.py

GitHub

The screenshot shows the GitHub repository page for `tomojitakasu/PocketSDR`. The repository is public and has 48 stars and 15 forks. The main branch is `master`. The repository contains a file tree with folders like `FW`, `HW`, `bin`, `conf`, `doc`, `driver/Win10/x64`, `image`, `lib`, `python`, `sample`, `src`, and files like `LICENSE.txt` and `README.md`. The `README.md` file is selected, showing the title **PocketSDR - An Open-Source GNSS SDR, ver. 0.6** and an **Overview** section. The overview describes PocketSDR as an open-source GNSS receiver based on SDR technology, supporting various GNSS signals. It also mentions the RF front-end device components: 2 CH Maxim MAX2771 GNSS RF front-end IC, LNA, mixer, filter, ADC, frequency synthesizer, and Cypress EZ-USB FX2LP USB 2.0 controller.

The screenshot shows the `README.md` file for the `tomojitakasu/PocketSDR` repository. The file contains the following text:

The RF front-end device consists of 2 CH Maxim MAX2771 GNSS RF front-end IC (LNA, mixer, filter, ADC, frequency synthesizer) and Cypress EZ-USB FX2LP USB 2.0 controller to connect to host PCs. The front-end CH1 is dedicated for GNSS L1 band (1525 - 1610 MHz) and CH2 is for GNSS L2/L5/L6 band (1160 - 1290 MHz). The frequency of the reference oscillator (TCXO) is 24.000 MHz and ADC sampling frequency can be configured up to 24 MHz.

PocketSDR contains some utility programs for the RF front-end device to configure the device, capture and dump the digitized IF (inter-frequency) data. These supports Windows, Linux and other environments.

PocketSDR also provides GNSS-SDR APs to show the PSD (power spectrum density) of captured IF data, search the GNSS signals, track these signals and decode navigation data in them. The supported GNSS signals are as follows. These APs are written in Python by very compact way. They are easily modified by users to add user's unique algorithms.

GPS: L1C/A, L1CP, L1CD, L2CM, L5I, L5Q, GLONASS: L1C/A, L2C/A, Galileo: E1B, E1C, E5aI, E5aQ, E5bI, E5bQ, E6B, E6C, QZSS: L1C/A, L1C/B, L1CP, L1CD, L1S, L2CM, L5I, L5Q, L5S1, L5S2, L6D, L6E, BeiDou: B1I, B1CP, B1CD, B2I, B2aD, B2aP, B2bI, B3I, SBAS: L1C/A, L5I, L5Q

The image shows the hardware components of the PocketSDR, including the EZ-USB FX2LP, MAX2771 v2, and TCXO 24.000 MHz.

Package Structure

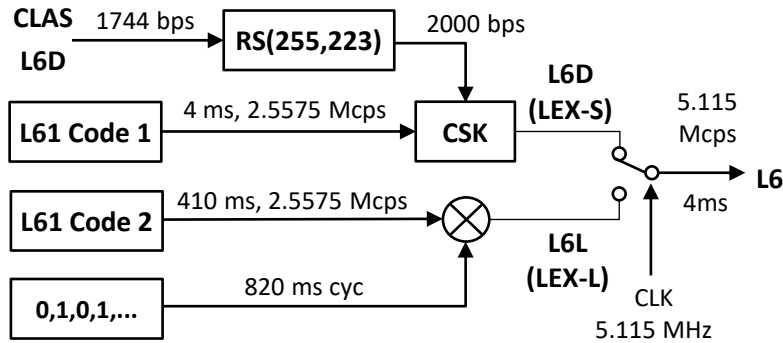
```
PocketSDR --+-- bin    PocketSDR utility binary programs for Windows
              +-+ src    PocketSDR utility source programs
              +-+ python PocketSDR utility Python scripts
              +-+ lib     External shared library for Python scripts
              +-+ conf     Configuration files for Python scripts
              +-+ driver   Windows driver for EZ-USB FX2LP/FX3 (cyusb3.sys) ([4])
              +-+ doc      Documents (ref [1], [2])
              +-+ FW       Firmware source programs and images
              |
              +-+ cypress  Cypress libraries for EZ-USB firmware development
```

<https://github.com/tomojitakasu/PocketSDR>

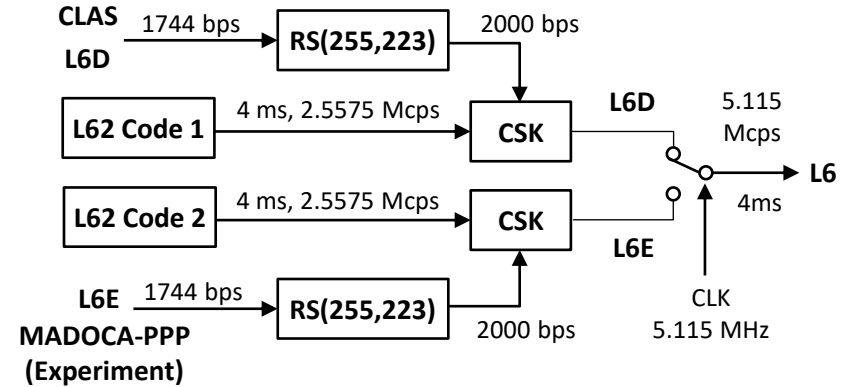
Applications of Pocket SDR

QZSS L6 Signal

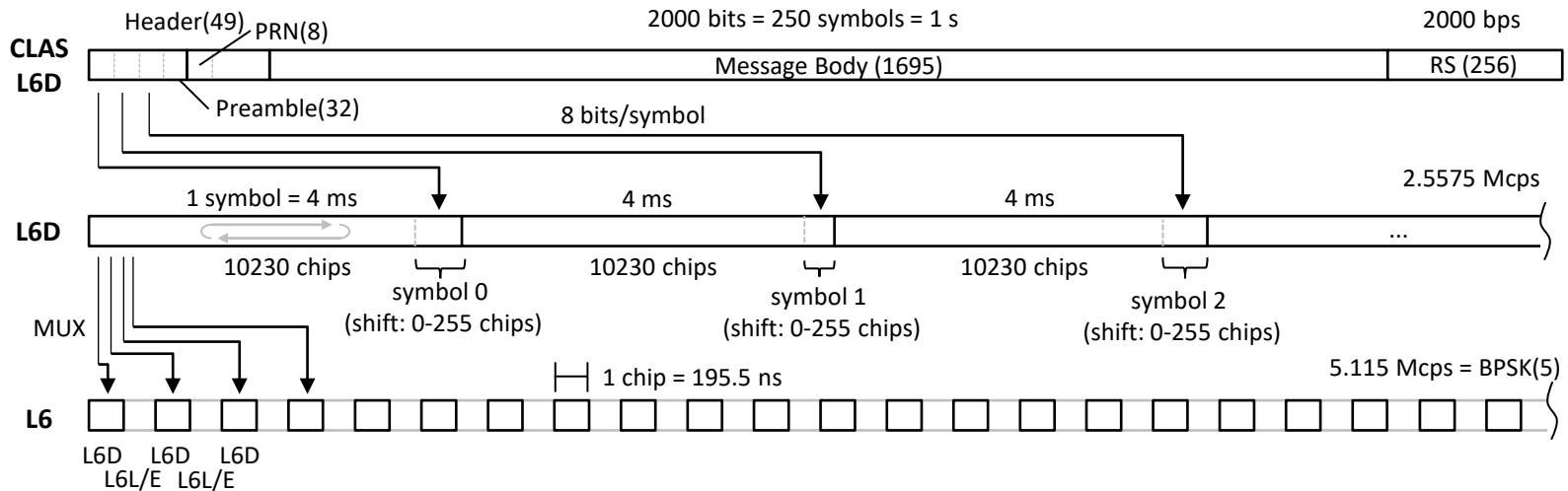
QZS-1



QZS-2, 3, 4, 1R, ...



**MADOCA-PPP
(Experiment)**



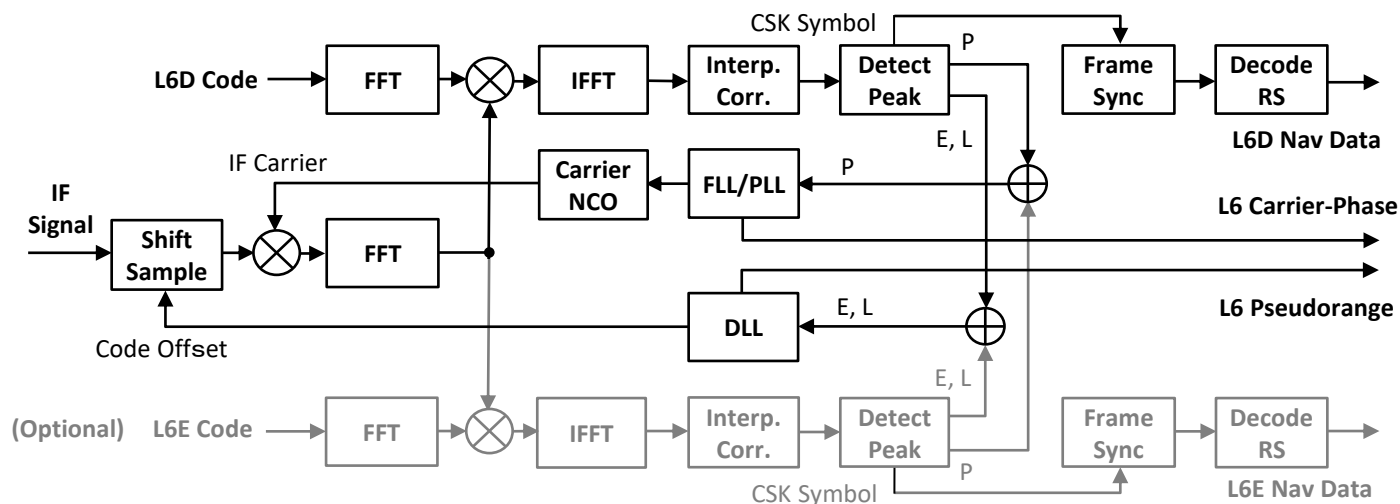
L6 CSK Receiver (1/2)

Existing Technique to Decode L6 CSK

- Decoding CSK referenced to L6L pilot (only for QZS-1)
- Decoding CSK referenced to L1 or L2C pilot (need dual freq. or wide BW)
- No Tracking L6D/E directly (no pseudorange or carrier-phase w/o DLL or PLL)
- NovAtel (only track QZS-1), u-blox D9C (L2C pilot), Allystar HD9310 (L1 pilot ?)

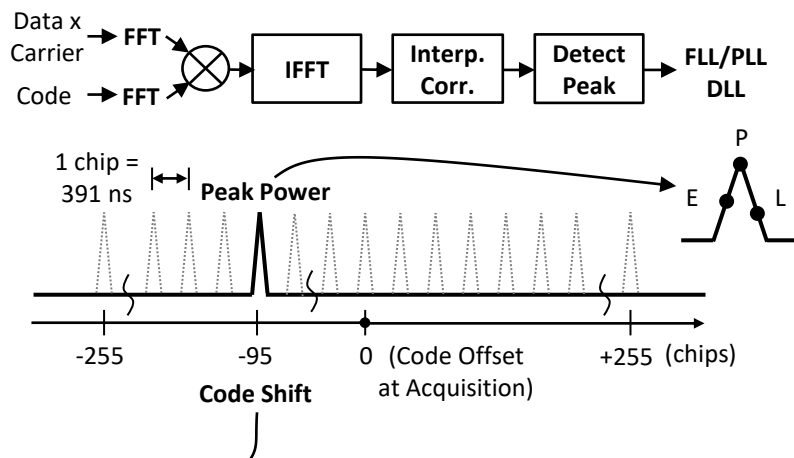
L6 CSK Receiver w/o pilot signal by **Pocket SDR**

- pocket_trk.py (ver. 0.6 ~)
- Narrow BW (10 MHz) IF to track BPSK(5)
- L6 Pseudorange and carrier-phase available for PVT

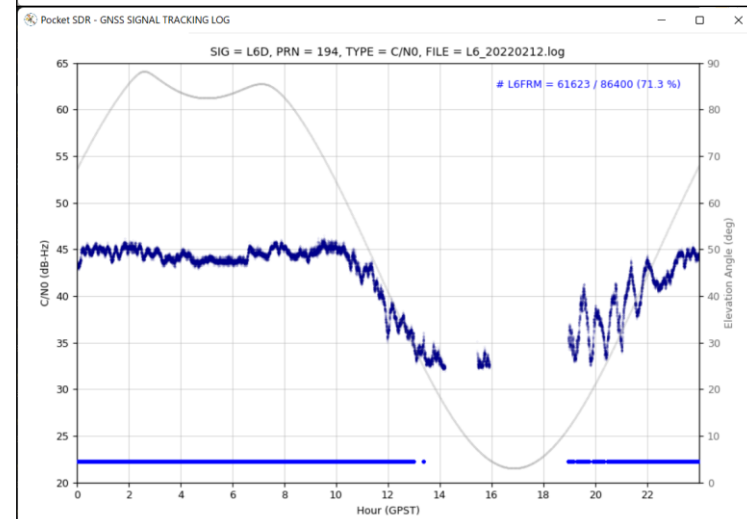
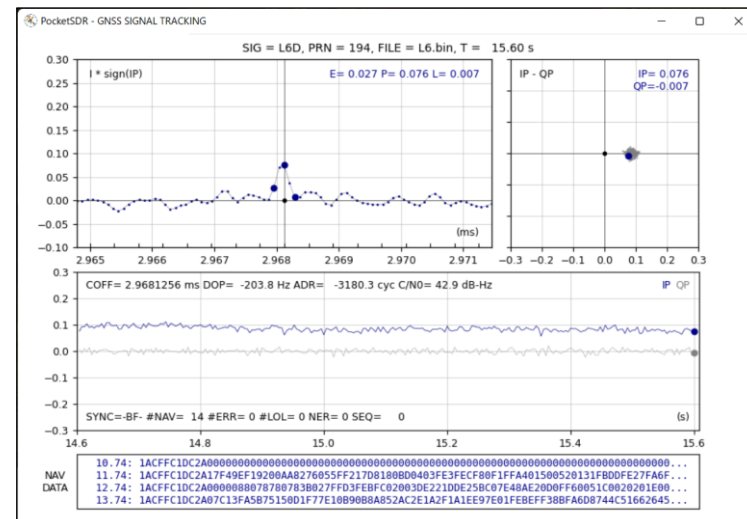
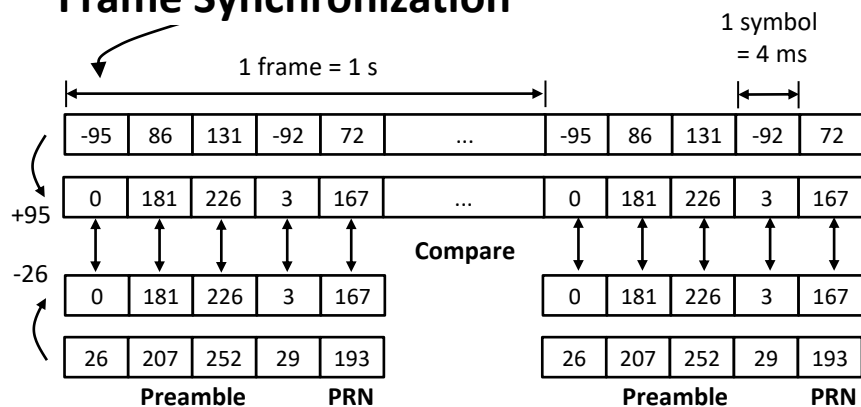


L6 CSK Receiver (2/2)

Generate Correlator Outputs EPL



Frame Synchronization



24H Real-time Tracking Status by pocket trk.py

Snapshot Positioning (1/2)

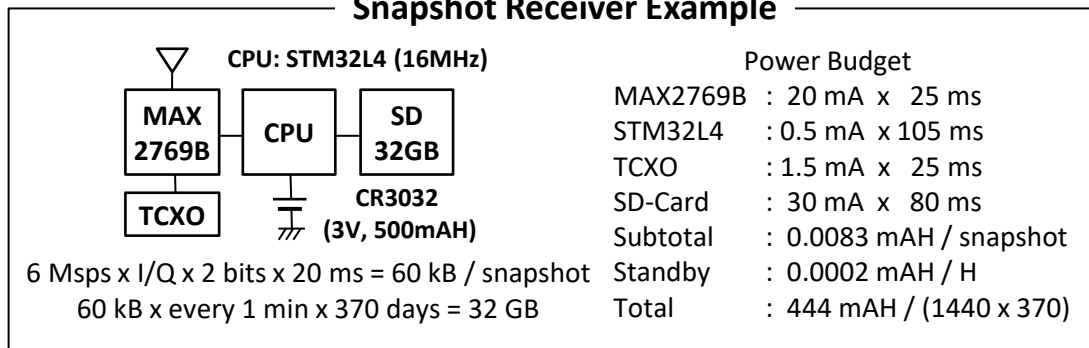
Snapshot Positioning

- Recode only DIF data in < 100 ms
- PVT by post-processing
- NAV data provided by external sources
- Very low power consumption by duty-cycling

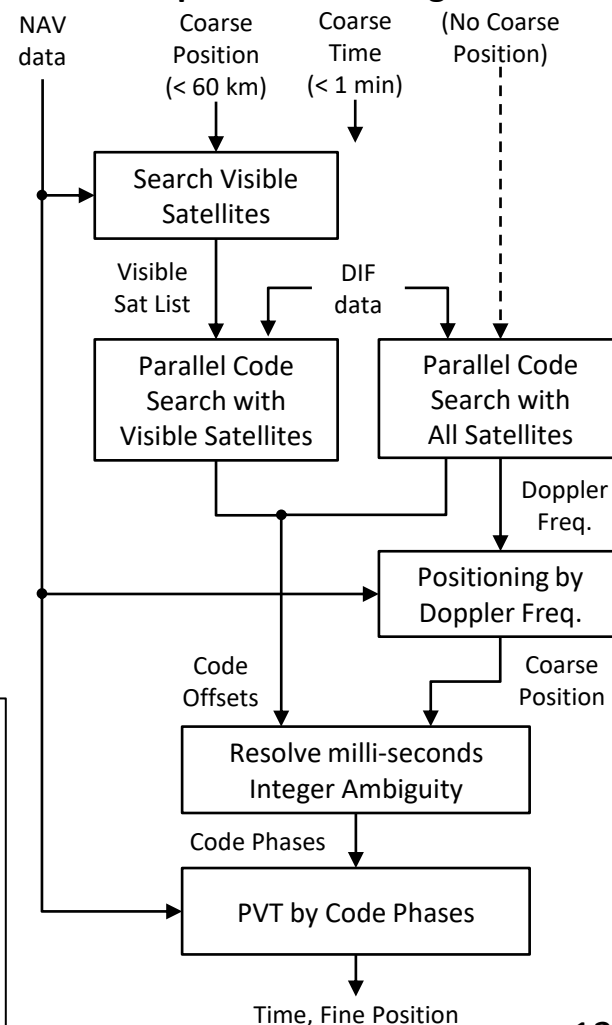
Snapshot Positioning by Pocket SDR

- pocket_snap.py (ver. 0.7 ~)
- L1 band (1575.42 MHz)
- GPS L1C/A, Galileo E1C, QZSS L1CP, BDS B1CP
- 4 or 6 Msps x I/Q x 2bits x 8 ~ 20 ms DIF
- Input RINEX NAV for Ephemerides

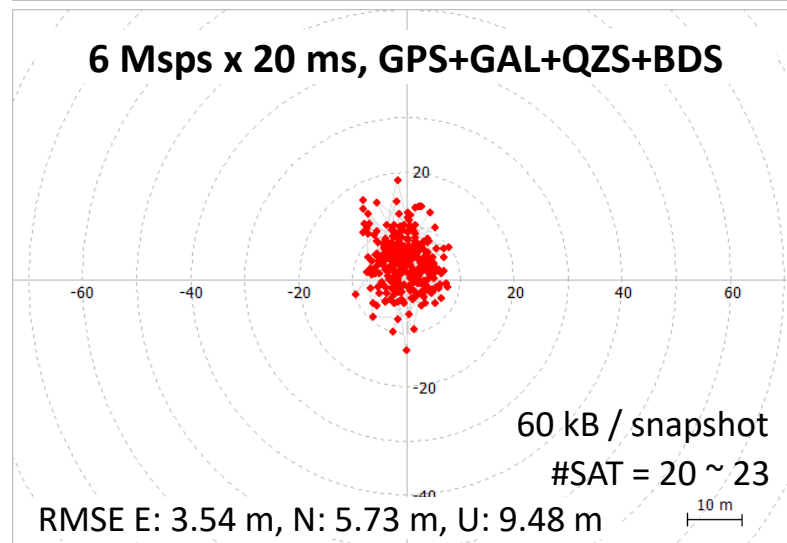
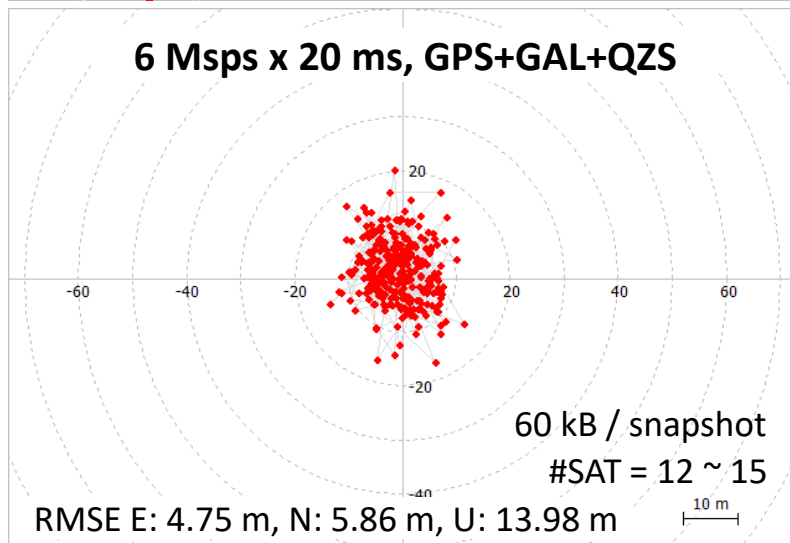
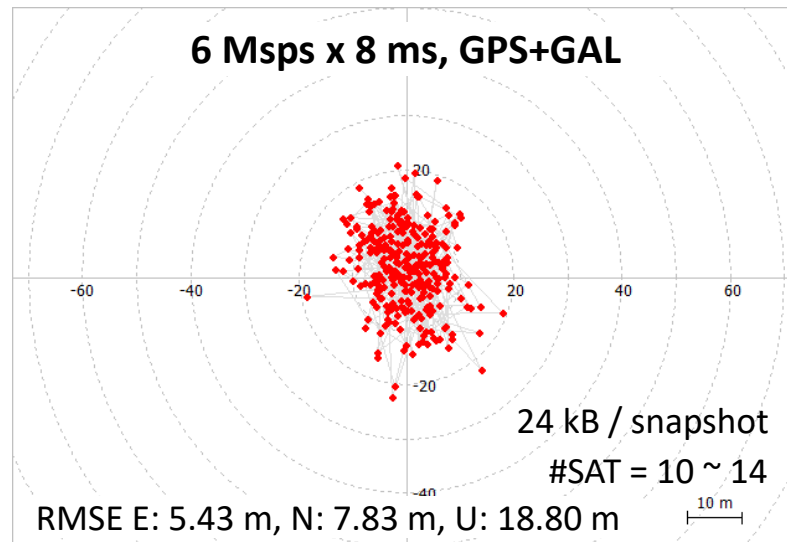
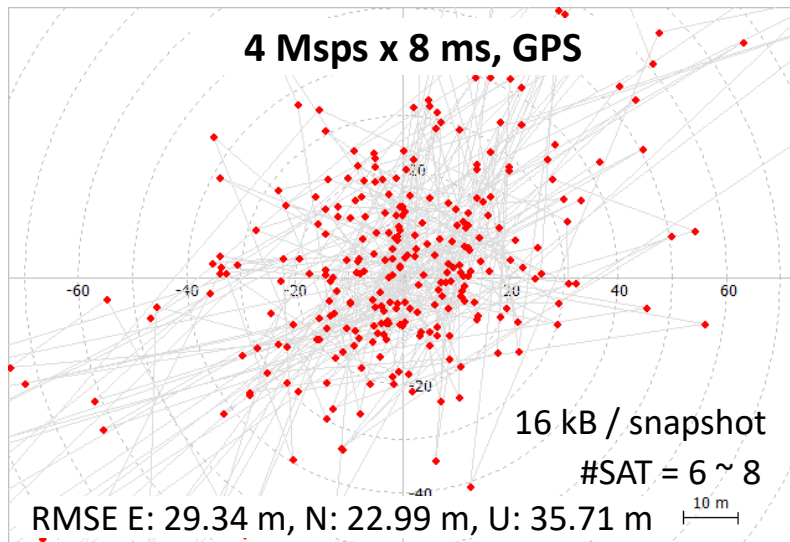
Snapshot Receiver Example



Snapshot Positioning Flow



Snapshot Positioning (2/2)



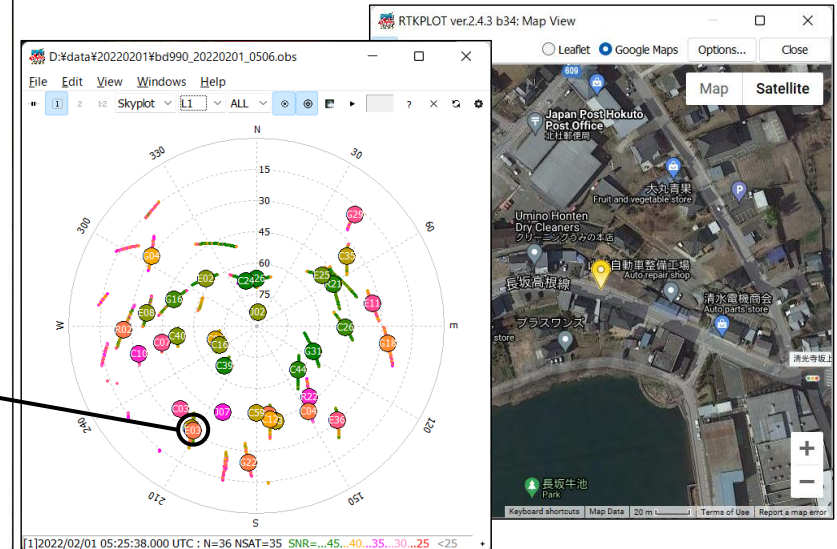
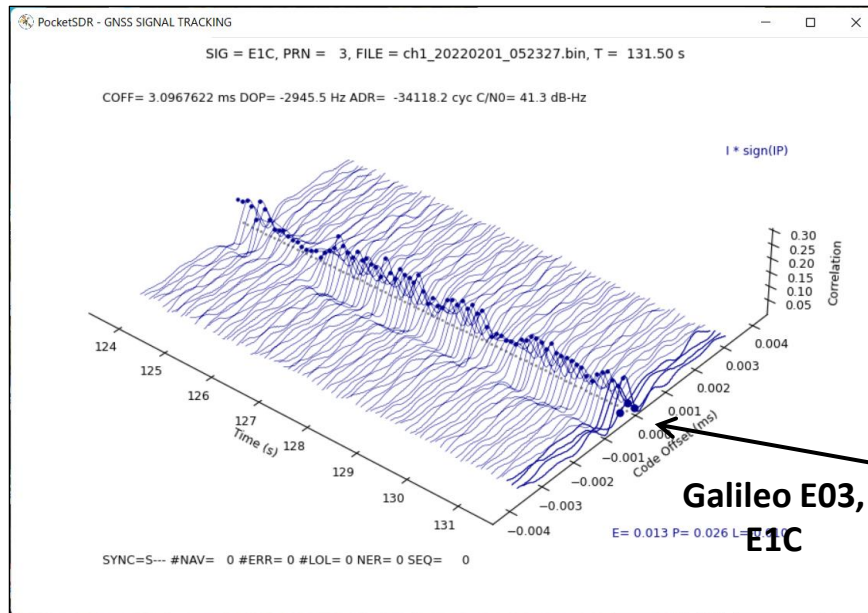
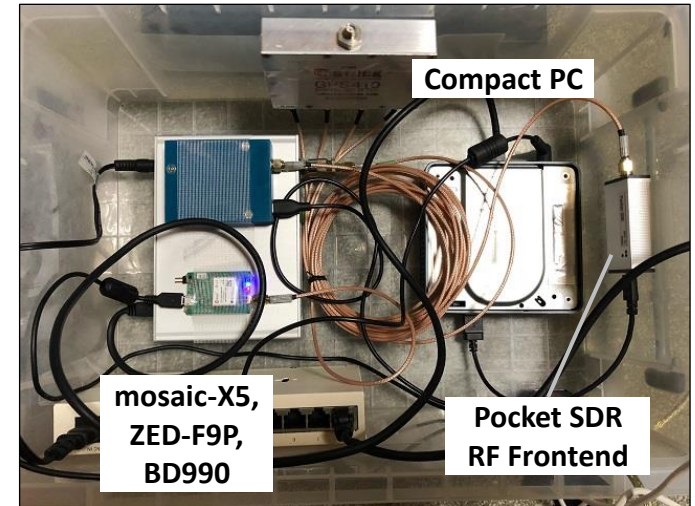
Horizontal Scatter Plots of 300 Snapshots, Post-processed by pocket_snap.py

GNSS Signal Recorder

GNSS Signal Recorder by Pocket SDR

- 2 CH x 24 Msps x I/Q x 2 bits (L1 + L5)
- Compact PC (Core i5 7260U, 2C/4T, RAM 8GB)
+ Ubuntu 18.04 LTS (Linux kernel 4.15)
- Max Capture Time: ~ 4.5 H / M.2 SSD 512 GB

Signal Analysis by pocket_trk.py



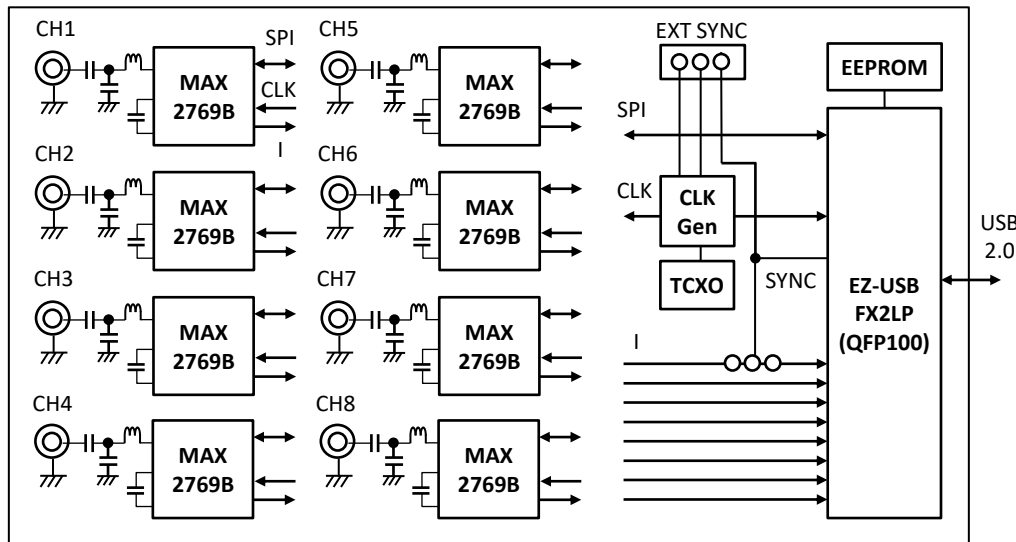
Post-processed by pocket_trk.py

Future Work

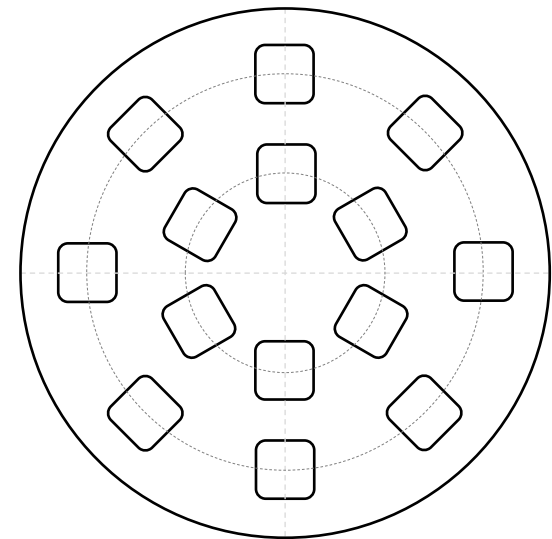
14 CH GNSS SDR RF Frontend

- GPS L1C/A, Galileo E1B/C, QZSS L1C/A and BDS B1C
- MAX2769B x 8 + EZ-USB FX2LP (QFP100, 16 bit Slave FIFO) / board
- 8 CH x 6 Msps x I/Q x 2 bits / board
- 14 CH x 6 Msps x I/Q x 2 bits / 2 boards (each 1 CH for board-synchronization)
- Total Parts Cost ~ \$150 / 2 boards

For Multipath-rejection, Anti-jamming, Anti-spoofing, ...



8 CH GNSS SDR RF Frontend Board



14 Elements GNSS Antenna