
How to make RTK mobile device with LTE

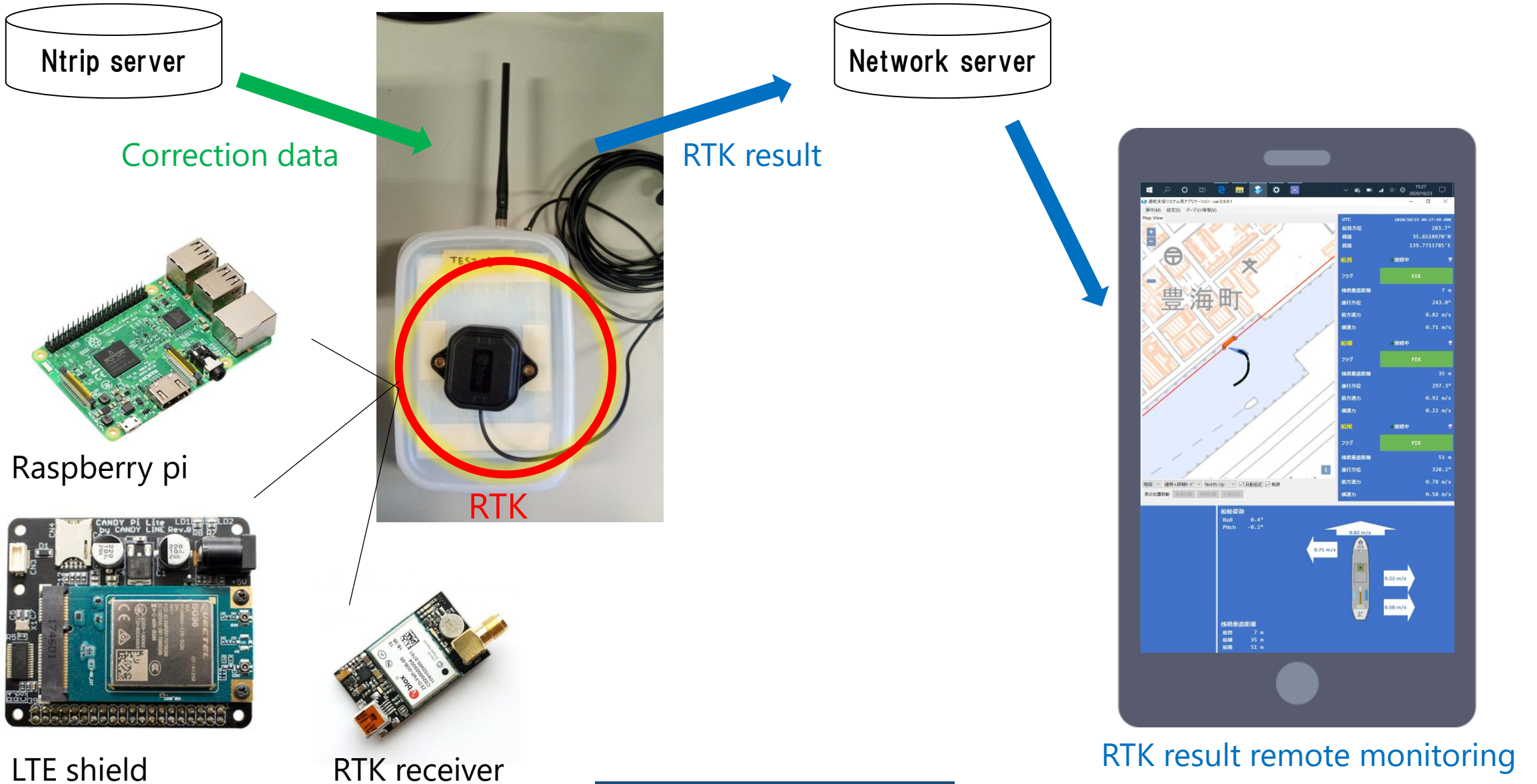
2021/03/15

1. System overview

Feature

- ◆ RTK on GNSS module
- ◆ Run by mobile power bank
- ◆ GNSS data logging
- ◆ Correction data logging
- ◆ RTK result up stream to network server (Ntrip, TCP, UDP)
- ◆ Low power consumption by LTE-M use

1. System overview

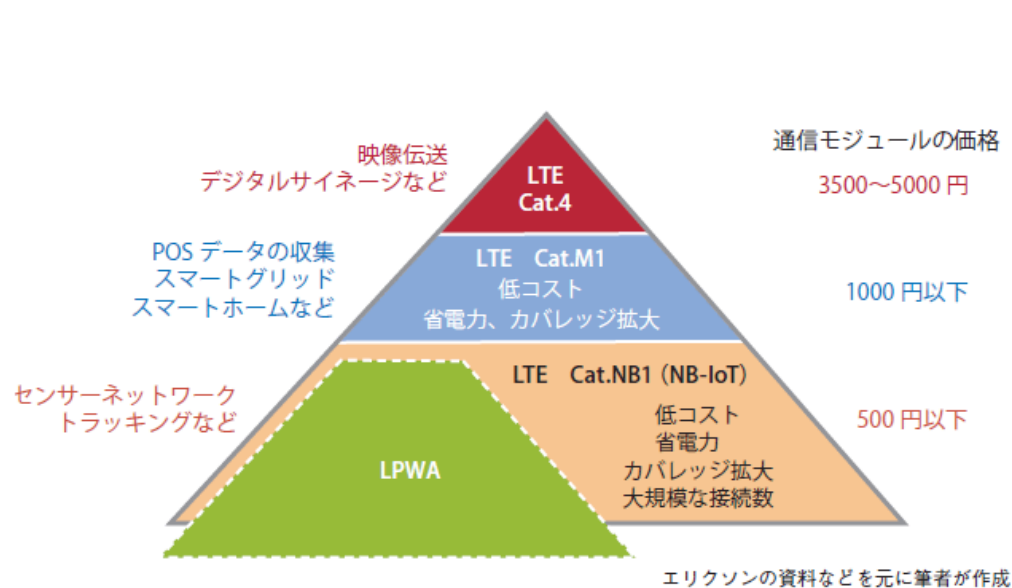


1. System overview

About LTE-M

One of LTE category.

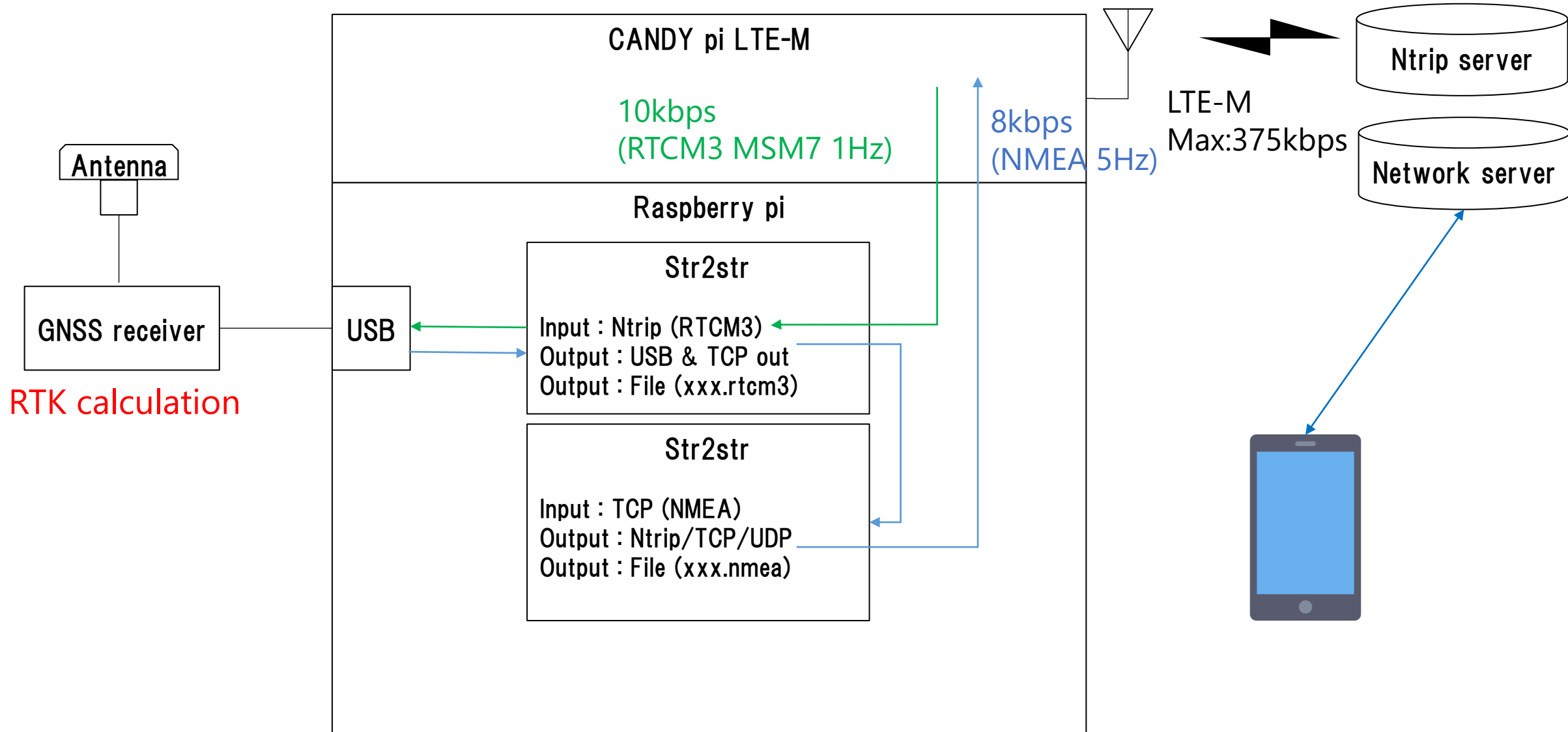
Low power consumption, low module cost but low data rate. Suitable for IoT device.
However, this data rate is enough for RTK correction receiving and NMEA output.



	Release 8 CAT 4	Release 8 CAT 1	Release 13 LTE-M	Release 13 NB-IoT
Downlink speed	150 Mbps	10 Mbps	384 kbps - 1 Mbps	170 - 250 kbps
Uplink speed	50 Mbps	5 Mbps	384kbps - 1 Mbps	50 - 144 kbps
Number of antennas	2	2	1	1
Duplex mode	Full duplex	Full duplex	Full or Half duplex	Half duplex
Receive bandwidth	20 MHz	20 MHz	1.4 MHz	200 kHz
Transmit power	23 dBm	23 dBm	20 dBm	23 dBm
Modem complexity	100%	80%	20%	<15%
Voice	Yes	Yes*	Yes*	No
Mobility	Full	Full	Limited	Fixed - Idle Mode Only

<https://businessnetwork.jp/Detail/tabid/65/artid/4775/Default.aspx>

1. System overview



2. Preparation

• Multi-band GNSS antenna	\$70
• ublox F9P	\$270
• Raspberry pi 3 Model B	\$50
• CANDY pi LTE-M	\$110
• LTE Data sim	-
• Micro SD card 16GB	\$7
• Mobile power bank 10000mAh	\$20

	\$527 + sim cost

*CANDY pi supports Mini PCIe LTE module by Quectel.
You should check what module is available in your region area.

For japan, BG96 is mounted.

<https://www.quectel.com/product/bg96.htm>



3. Set up

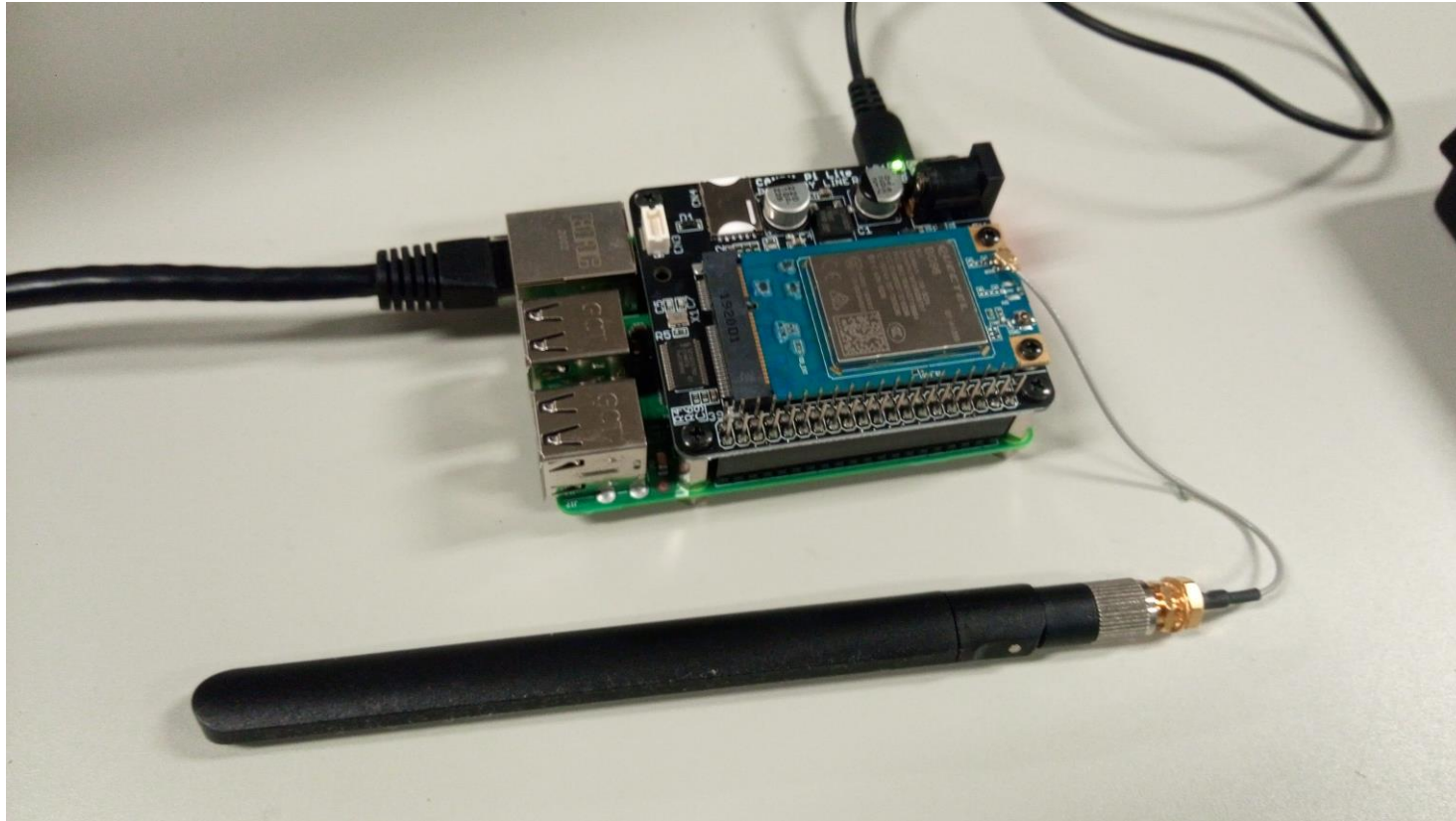
1. Assemble Raspberry pi and CANDY pi
2. Install rasbian OS on micro SD card
3. Install candy-pi-lite-service software
4. APN setting
5. Build str2str from RTKLIB
6. Make bash file to run str2str
7. Register bash file as auto start program

*For convenient use

Enable SSH, Enable VNC of Raspberry pi

3. Set up

1. Assemble Raspberry pi and CANDY pi



3. Set up

2. Install rasbian OS on micro SD card

Download OS writer

<https://www.raspberrypi.org/downloads/>



3. Set up

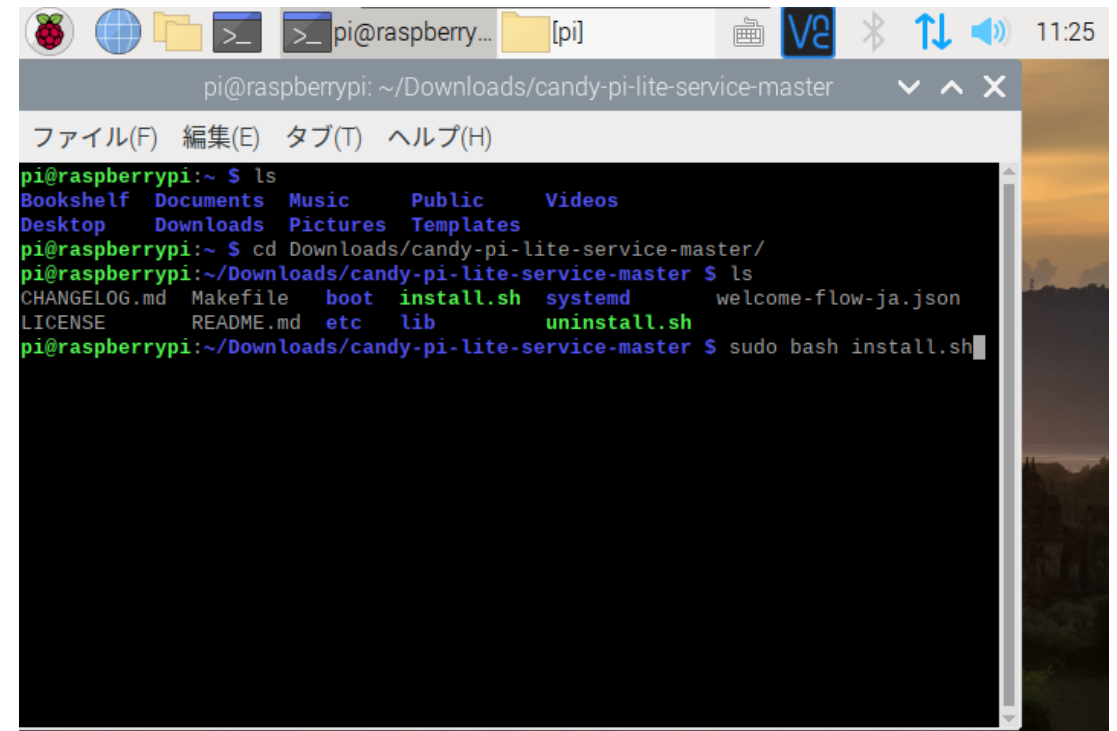
3. Install candy-pi-lite-service software

Clone from github

<https://github.com/CANDY-LINE/candy-pi-lite-service>

Type on terminal

“sudo bash install.sh”



```
pi@raspberrypi: ~/Downloads/candy-pi-lite-service-master
ファイル(F) 編集(E)  タブ(T)  ヘルプ(H)
pi@raspberrypi:~ $ ls
Bookshelf  Documents  Music      Public     Videos
Desktop    Downloads  Pictures   Templates
pi@raspberrypi:~ $ cd Downloads/candy-pi-lite-service-master/
pi@raspberrypi:~/Downloads/candy-pi-lite-service-master $ ls
CHANGELOG.md  Makefile  boot  install.sh  systemd  welcome-flow-ja.json
LICENSE        README.md  etc   lib         uninstall.sh
pi@raspberrypi:~/Downloads/candy-pi-lite-service-master $ sudo bash install.sh
```

3. Set up

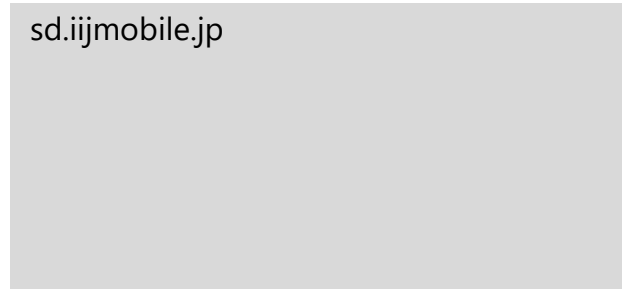
4. APN setting

Change apn name on /opt/candy-line/candy-pi-lite/apn.

Example is when we used IJ type D sim card.

After save "apn", reboot raspberry pi.

```
pi@raspberrypi:~/Downloads/candy-pi-lite-service-master $ cd /opt/candy-line/candy-pi-lite/
pi@raspberrypi:/opt/candy-line/candy-pi-lite $ ls
__modem_serial_port      boot-ip.windows.json      modem_reset.sh
__pppd_running           boot-ip.windows.usb0.json server_main.py
_common.sh               boot-ip.windows.wlan0.json start_pppd.sh
_pin_settings.sh        connection_resume.sh      start_systemd.sh
apn                      connection_status.sh      stop_systemd.sh
apn-list.json           connection_suspend.sh     uninstall.sh
boot-ip.macos.bridge.json environment
boot-ip.macos.json      fallback_apn
pi@raspberrypi:/opt/candy-line/candy-pi-lite $ sudo vi apn
pi@raspberrypi:/opt/candy-line/candy-pi-lite $
```



apn

3. Set up

5. Build str2str from RTKLIB

Clone RTKLIB from github

https://github.com/tomojitakasu/RTKLIB/tree/rtklib_2.4.3

Move to RTKLIB/app/str2str/gcc/ and

“sudo make”

Copy executable file “str2str” to your workspace.

3. Set up

6. Make bash file to run str2str

Example : Receive correction data from Ntrip mountpoint ECJ27 and up stream RTK result to Ntrip mountpoint TEST01

More detail about command please refer to RTKLIB manual p.99~ about command.

http://www.rtklib.com/prog/manual_2.4.2.pdf

```
#!/bin/bash
sleep 10s
pgrep -f str2str | xargs kill -9
sudo chmod 666 /dev/ttyACM0
cd /home/pi/strsvr_RTK
sudo chmod 777 str2str
lxtterminal -e ./str2str -in ntrip://[user ID]:[password]@[IP address]:2101/ECJ27 -out serial://ttyACM0:115200:8:n:1:off#10001 -
out file://data/Base_%y%m%d%H%M%S.rtc
sleep 2s
lxtterminal -e ./str2str -in tcpcli://127.0.0.1:10001 -out ntrips://:[password] @[IP address]:2101/TEST01 -out
file://data/Rover_%y%m%d%H%M%S.ubx
```

ntrip-serial.sh

3. Set up

7. Register bash file as auto start program

<https://qiita.com/tonosamart/items/f59daa481f90c85a8a99>

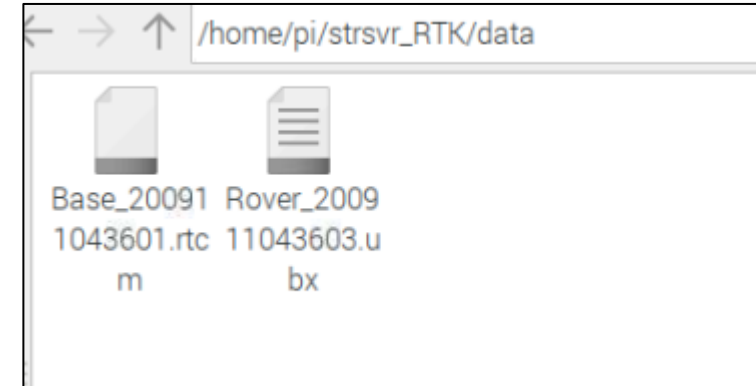
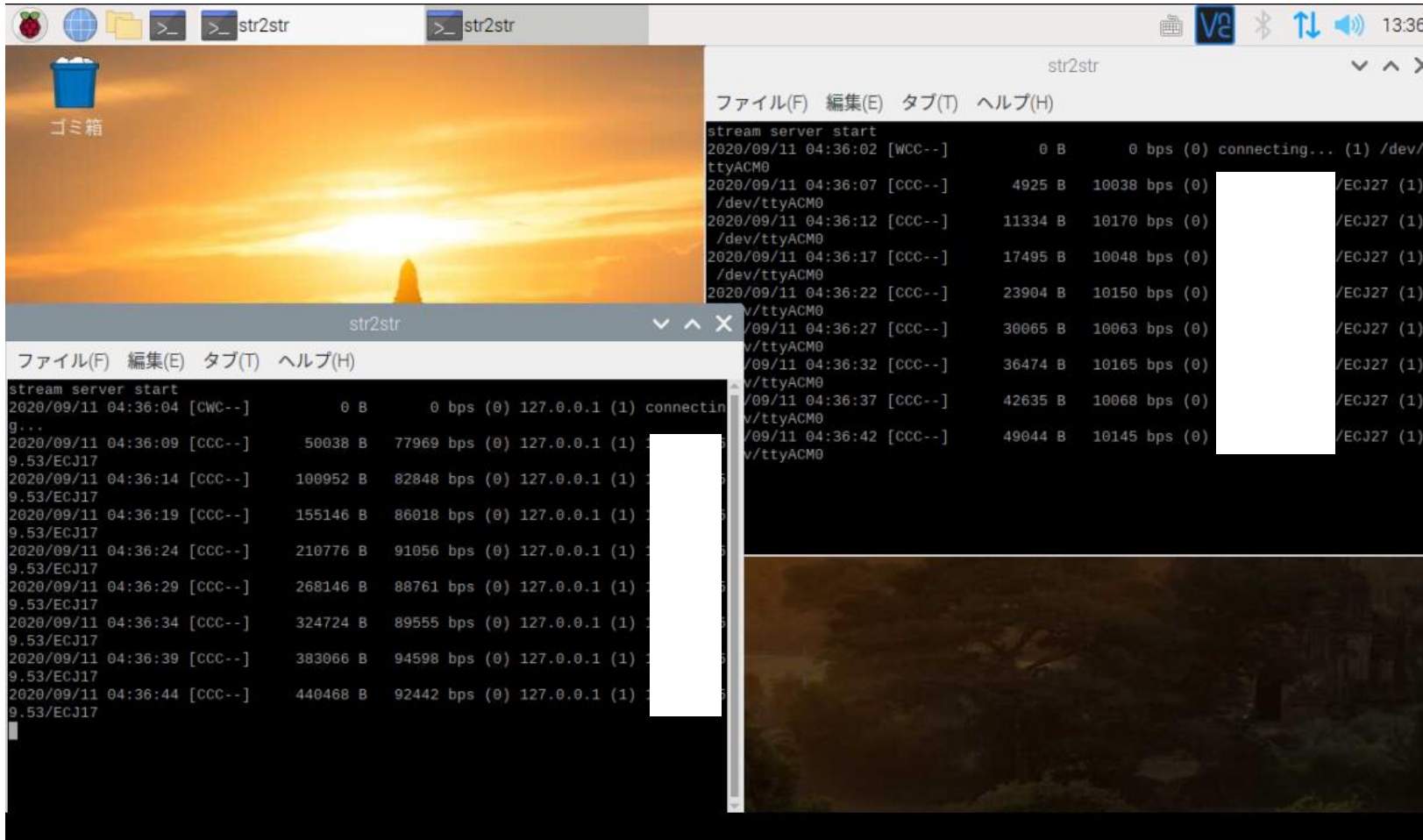
```
cd .config
mkdir autostart
cd autostart
vi autostart.desktop
sudo chmod 777 /home/pi/strsvr_RTK/ntrip-serial.sh
```

```
[Desktop Entry]
Exec=lxterminal -e /home/pi/strsvr_RTK/ntrip-serial.sh
Type=Application
Name=strsvr_rtk
Terminal=true
```

autostart.desktop

5. Test result

Boot screen of Raspberry pi
2 str2str software is running.

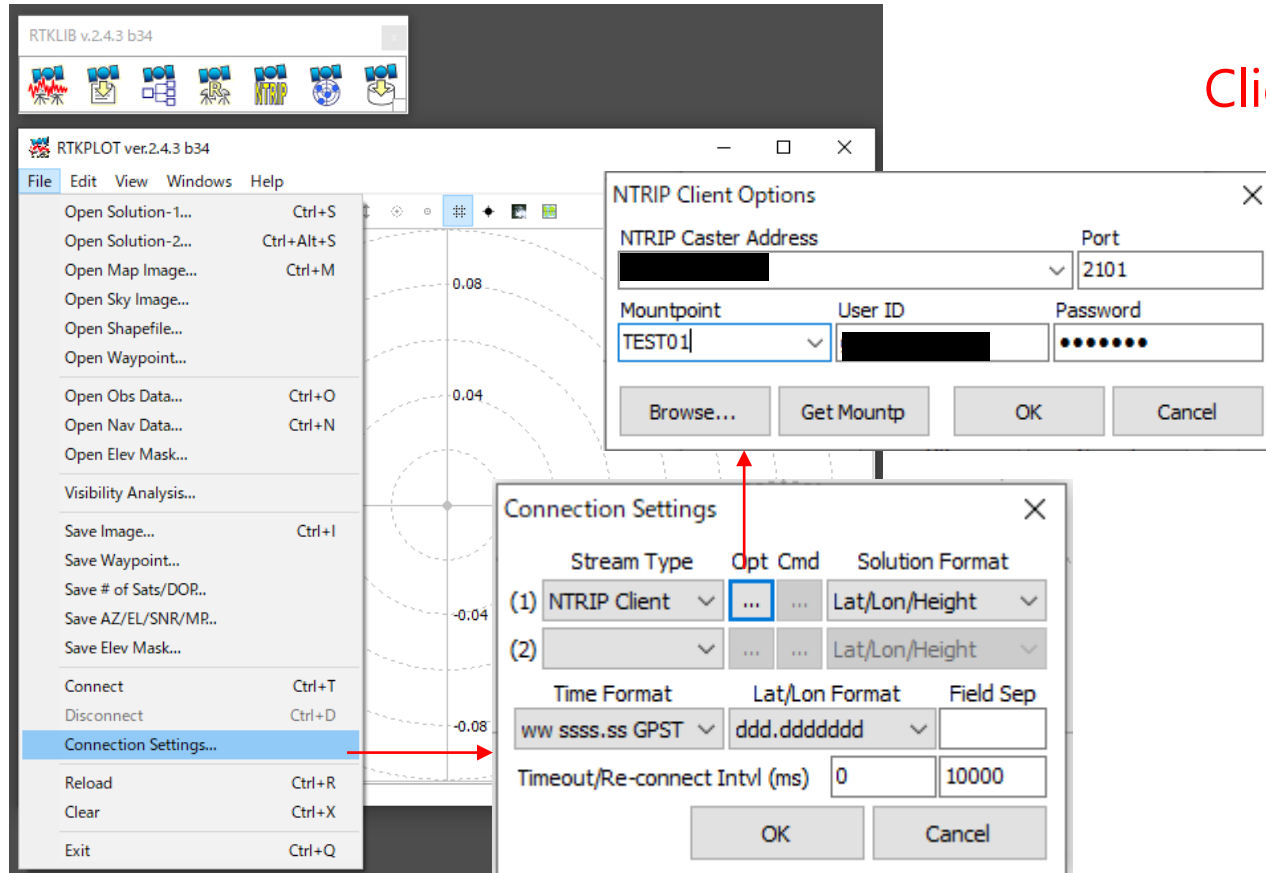


Internal log file

4. Test result

Monitoring result via Ntrip

Battery runtime : 18 hour with 20000mAh power bank



Click →

