

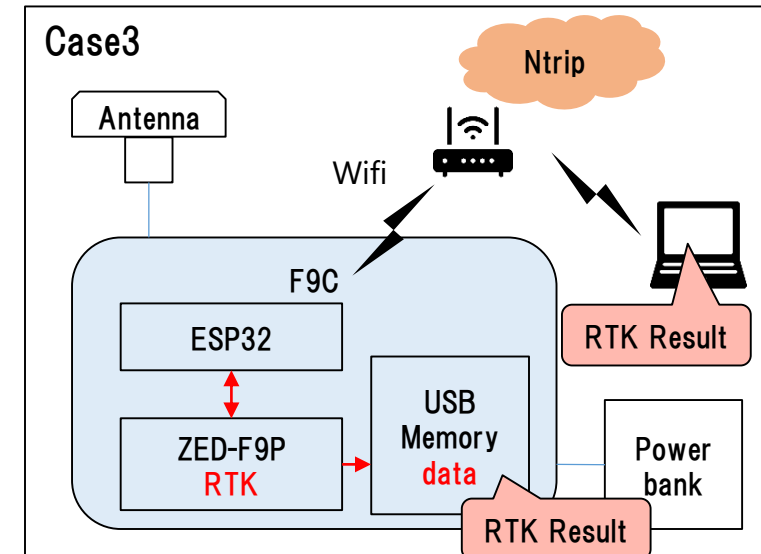
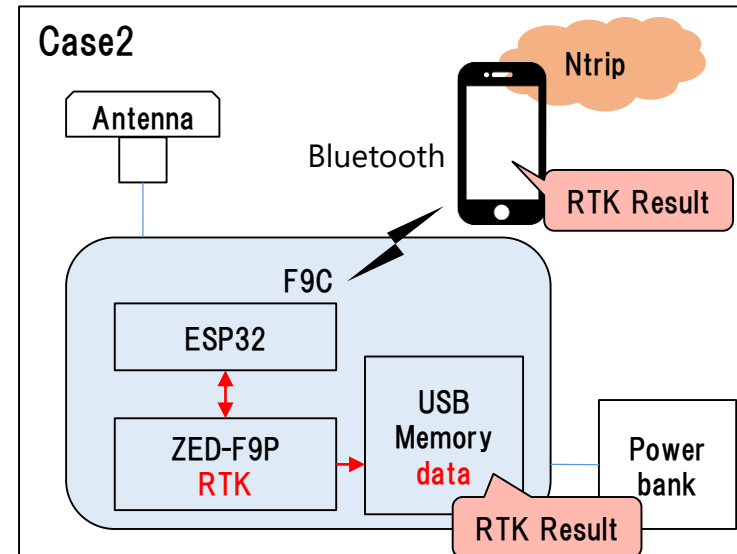
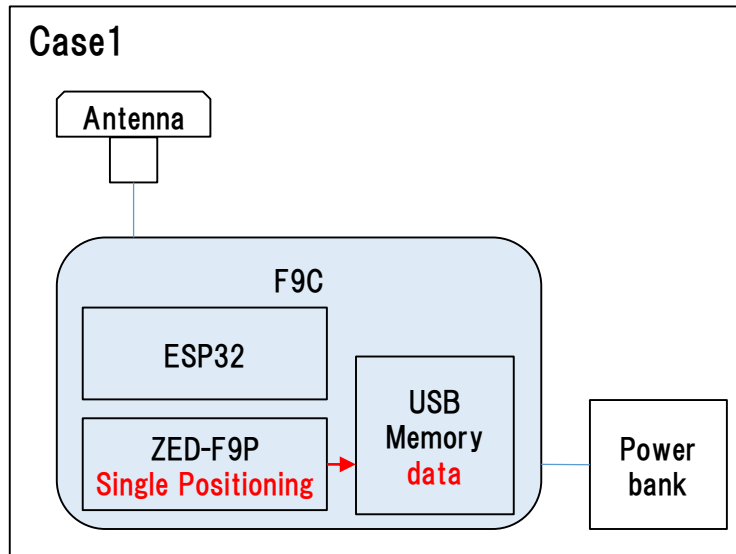
---

# RTK logger F9C setting manual

2020/04/20

# 1. About F9C

- ◆ F9C is RTK data logger powered by ublox ZED-F9P.
- ◆ Contact "Sensorcomm" to purchase.  
<http://www.sensorcomm.co.jp/>
- ◆ ESP32 WiFi and Bluetooth module is included.  
(Case 2 and Case 3 are alternative)



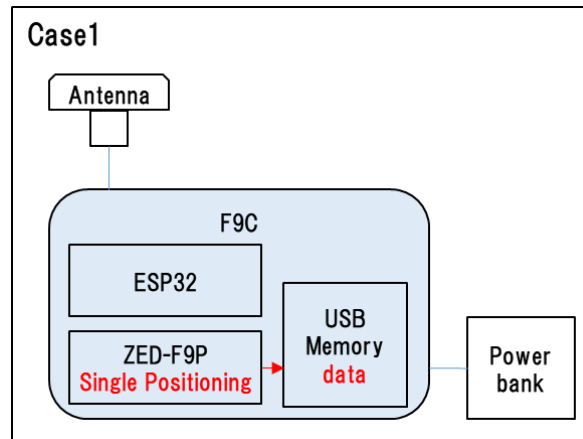
# 1. About F9C

## ◆ Power consumption

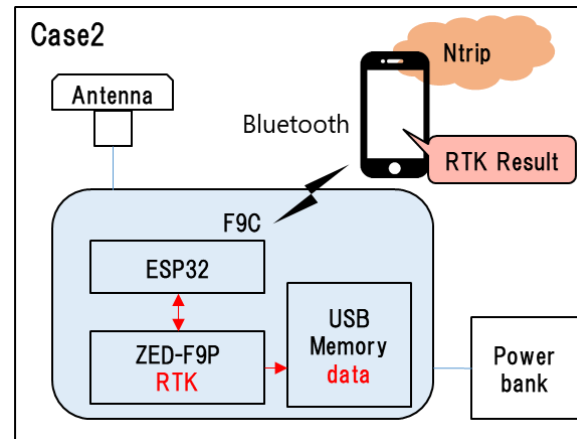
Tested with 10000mAh power bank.

(Bluetooth and WiFi power consumption depend on the signal environment)

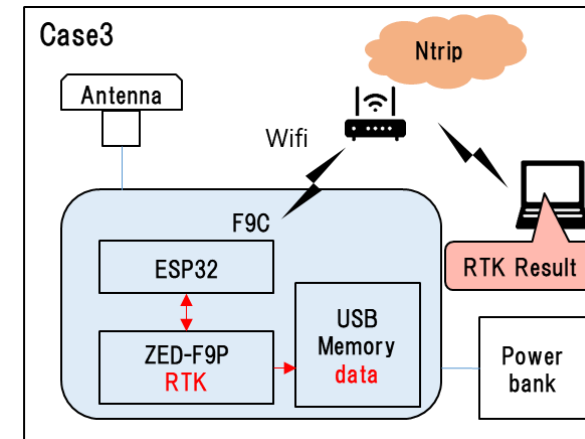
Running time 24 h >



20 h >



16 h

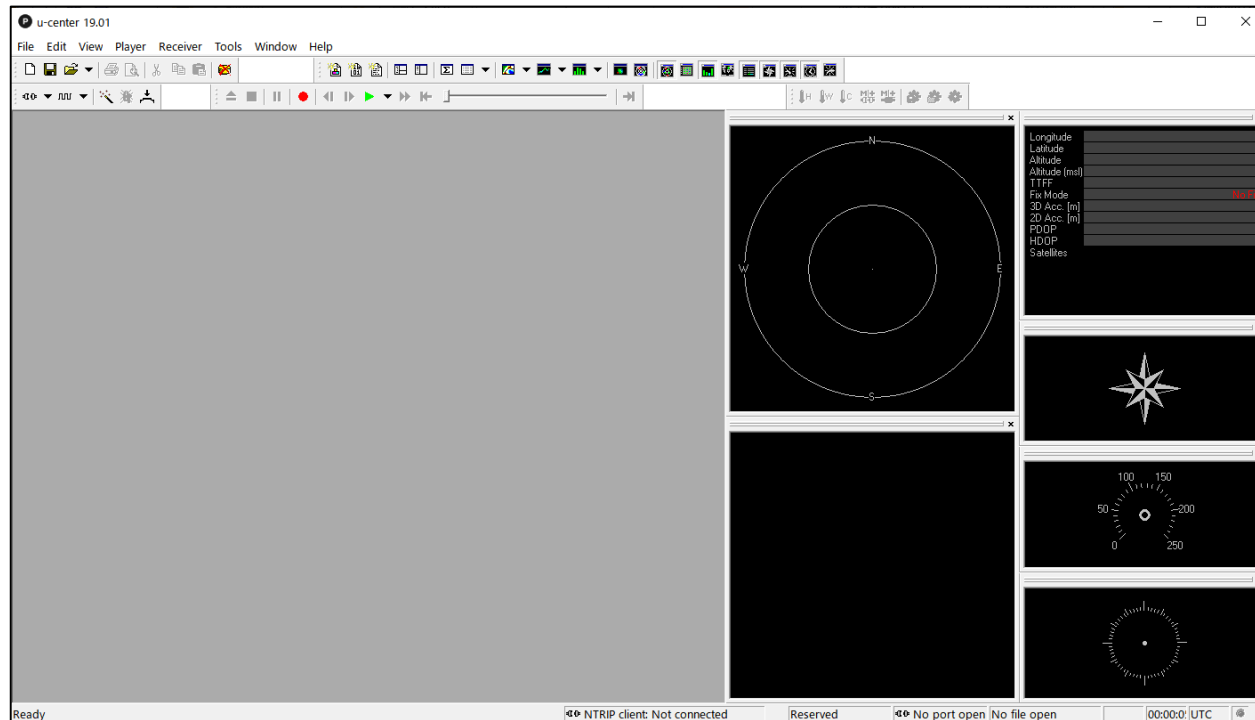


## 2. F9P setting

F9C can't log data in factory default. You have to set it up yourself.

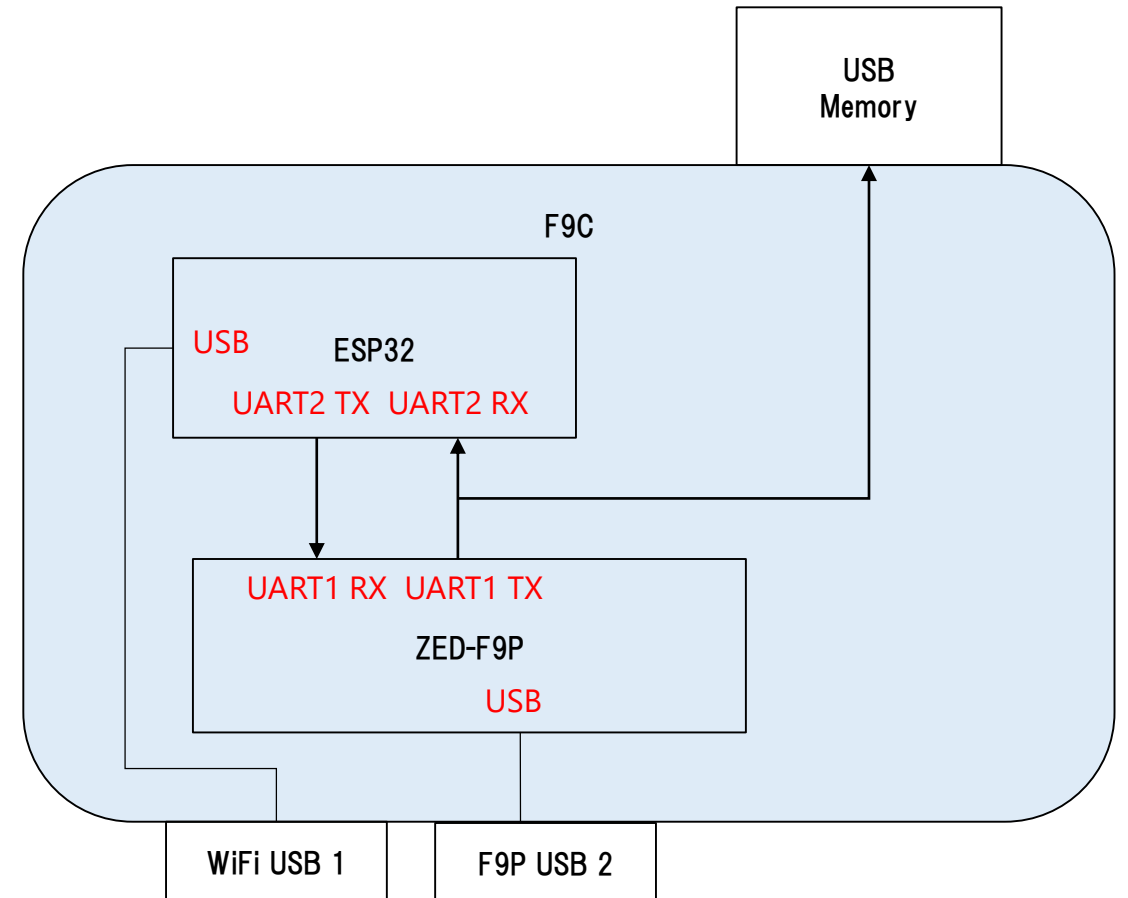
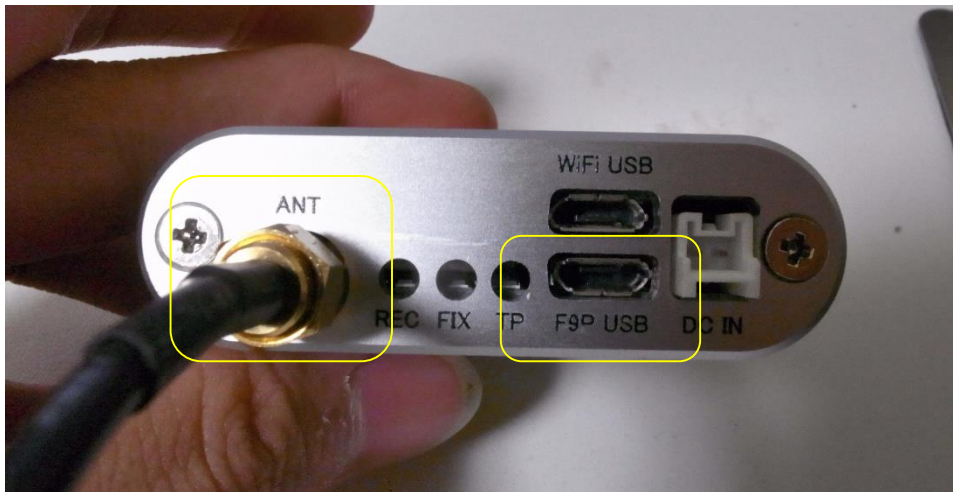
1. Install u-center, ublox GNSS software in your PC.

<https://www.u-blox.com/en/product/u-center>



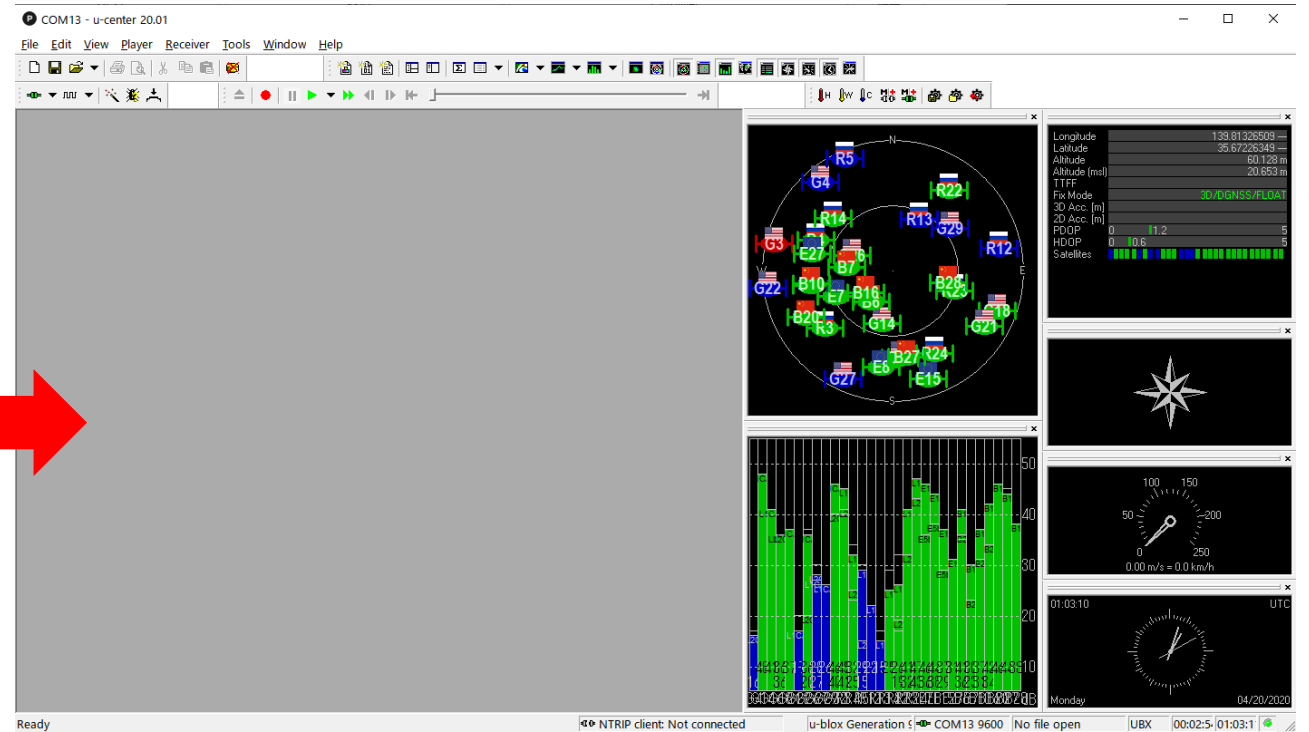
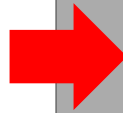
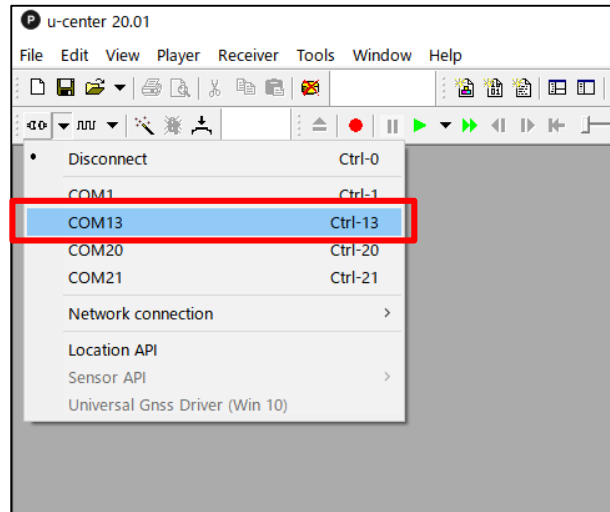
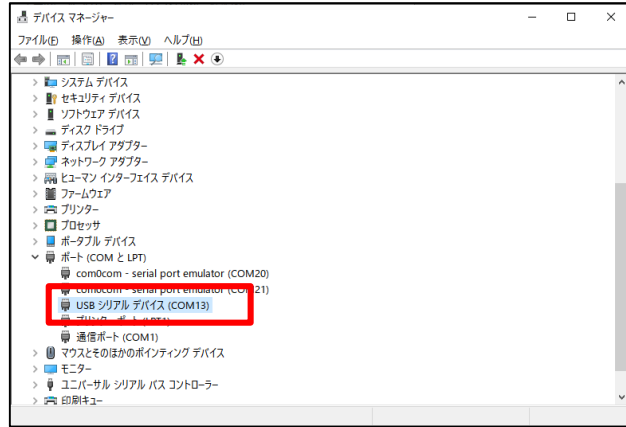
## 2. F9P setting

2. Connect GNSS Antenna.
3. Connect your PC and F9C by USB cable.



# 2. F9P setting

## 4. Select COM port and connect to F9P.



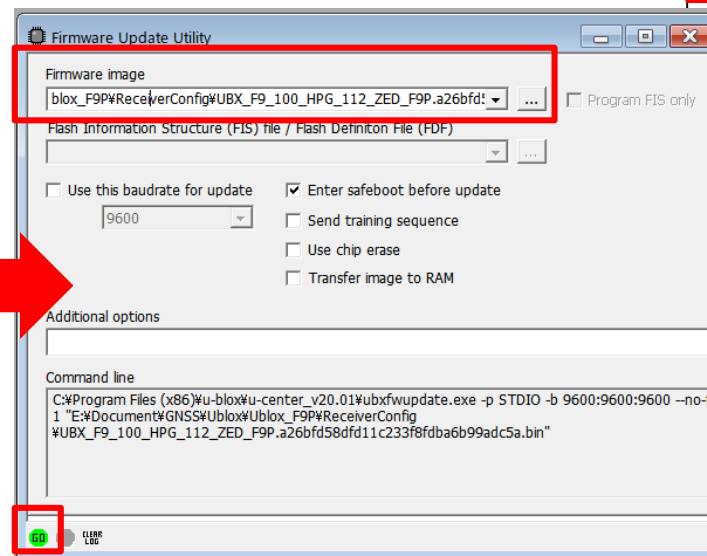
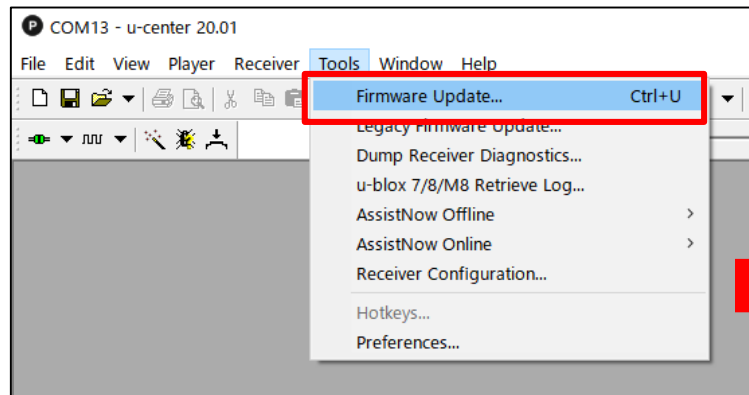
## 2. F9P setting

5. Download newest Firmware from u-blox web site.

<https://www.u-blox.com/en/product/zed-f9p-module#tab-documentation-resources>

6. Tools>Firmware Update

Select image file you downloaded and click "GO".



Product Change Note		
F9 and D9 Boot Loader Update	19-Dec-2019	133 KB PDF Download
System/software design		
ZED-F9P Product Change	12-Jul-2019	105 KB PDF Download
Production Deployment		
Conformity Declaration		
ZED-F9P Declaration of Conformity (RED)	16-Jul-2019	216 KB PDF Download
Certification		
Firmware Update		
Firmware UBX_F9_100_HPG_112_ZED_F9P	12-Jul-2019	1.02 MB LINK Download
System/software design Deployment		
Overview		
blox Package Reference	03-Apr-2020	3.51 MB PDF Download
Deployment		
GNSS product line card	18-Feb-2020	356 KB PDF Download
Product evaluation		
blox GNSS product overview	15-May-2019	548 KB PDF Download

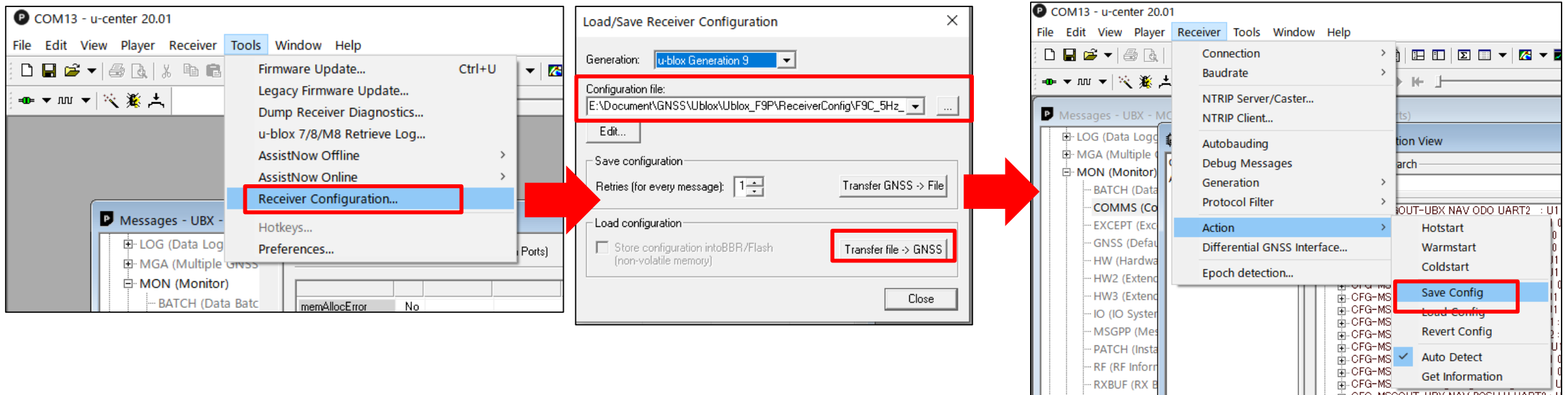
## 2. F9P setting

7. Download configuration sample file for F9C.

[http://www.denshi.e.kaiyodai.ac.jp/gnss\\_tutor/pdf/F9C/F9C\\_5Hz\\_NMEA.txt](http://www.denshi.e.kaiyodai.ac.jp/gnss_tutor/pdf/F9C/F9C_5Hz_NMEA.txt)

Load configuration file from Tools>Receiver Configuration.

After loading configuration, don't forget to do "save config" before power off the receiver.





# 2. F9P setting

7. Note: This sample configuration file output following sentence to USB memory and ESP32.

- NMEA...GGA, RMC, VTG...5Hz

If you want other sentences, please customize yourself.

- 2. Change value and "Set in RAM"  
Value 1= output enable  
Value 0=output disable

Example : output UBX-NAV-PVT

1. Select Message "xxx\_UART1"

2. Change value and "Set in RAM"  
Value 1= output enable  
Value 0=output disable

3. After queue was set click "Send config change"

4. Change UART1 "Protocol out" as suitable as your output message at message window.

### 3. Data logging

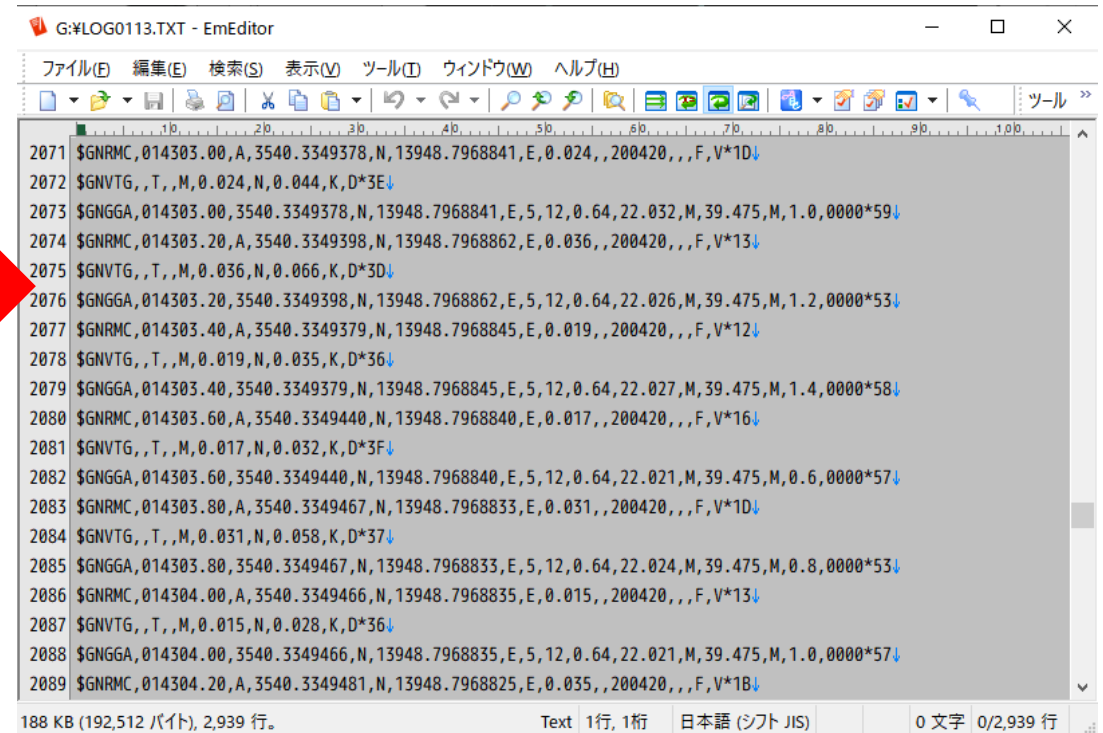
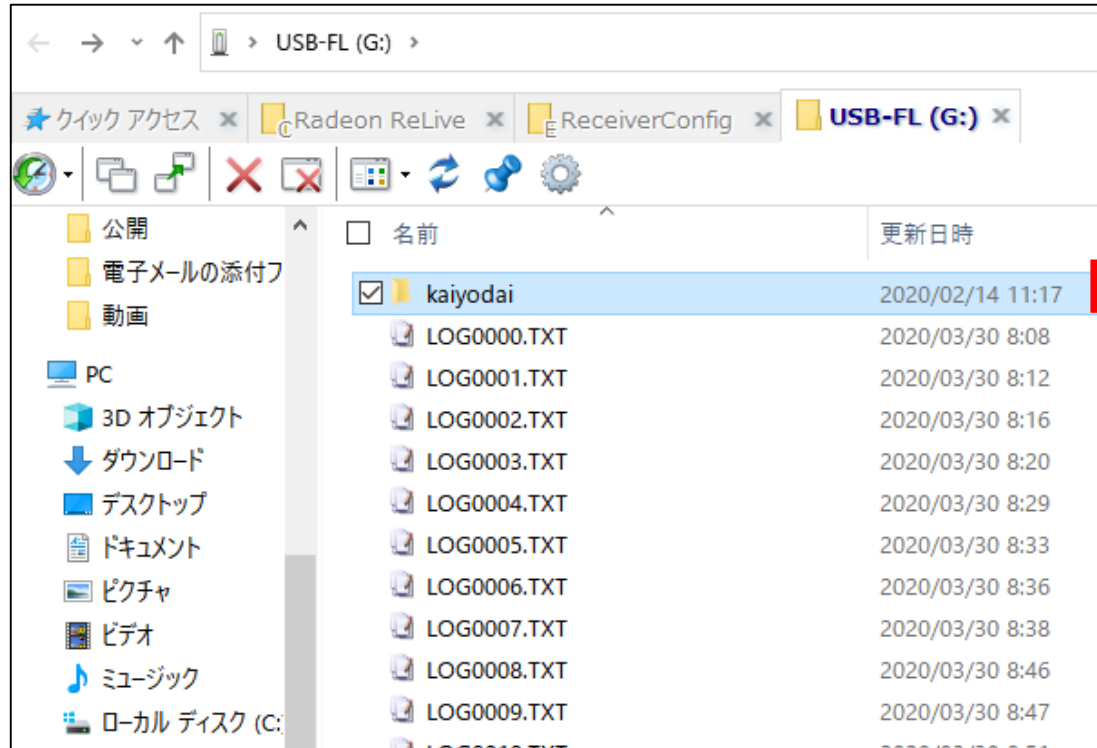
USB memory data logging will automatically start when you power on F9C.

And disconnect power USB, it stop logging.



# 3. Data logging

You can salvage data from USB memory. (USB memory data format should be FAT32)



## 4. Bluetooth setting

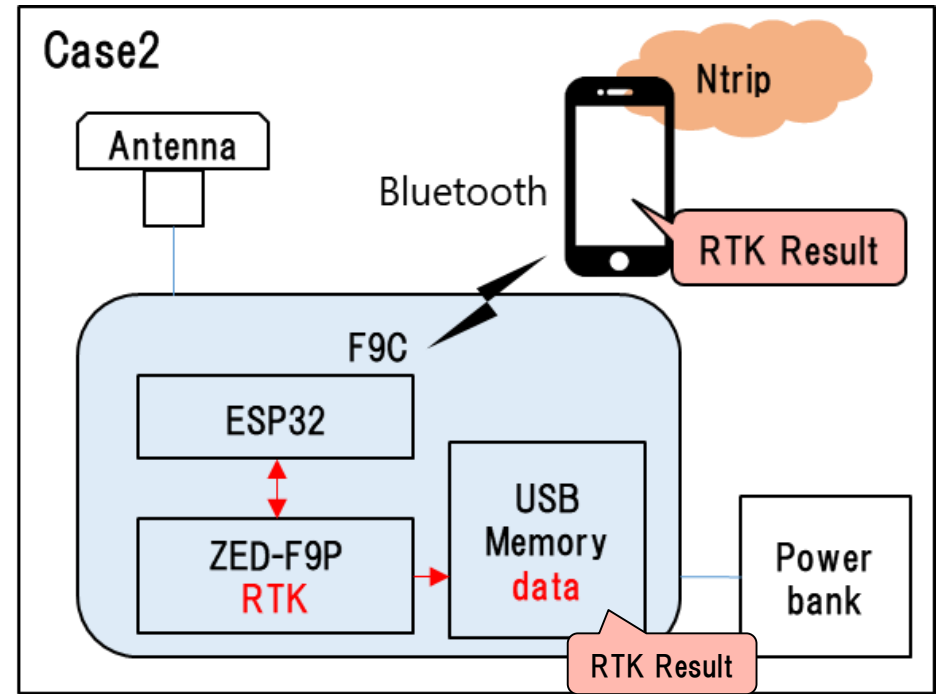
You can get RTK position with your smartphone (Android).

1. Install Arduino IDE in your PC.

<https://www.arduino.cc/en/main/software>

2. Download sketch for ESP32.

[http://www.denshi.e.kaiyodai.ac.jp/gnss\\_tutor/pdf/F9C/ESP32\\_BT\\_F9C\\_v2.zip](http://www.denshi.e.kaiyodai.ac.jp/gnss_tutor/pdf/F9C/ESP32_BT_F9C_v2.zip)

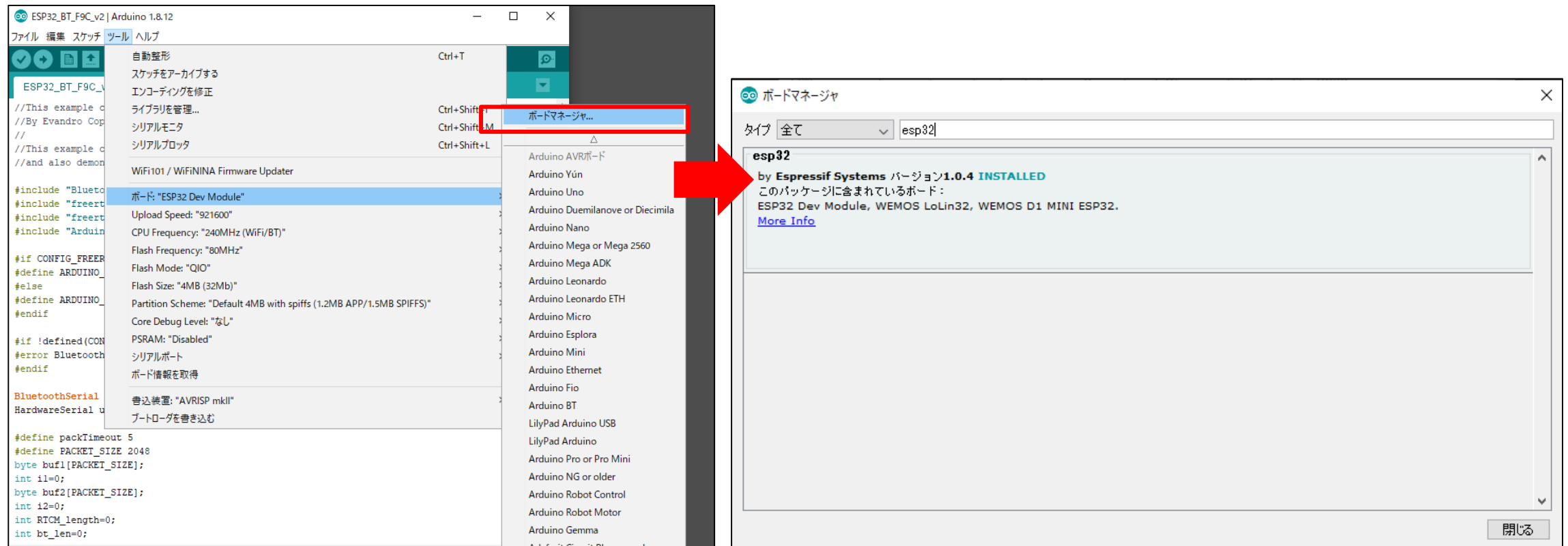


# 4. Bluetooth setting

3. Open downloaded sketch "ESP32\_BT\_F9C\_v2.ino"

4. Install the library for ESP32.

Tool > Board > Board Manager. Then search by "esp32" and install the library.



# 4. Bluetooth setting

5. Connect F9C with PC by USB cable.



6. Select board type as "ESP32 Dev Module" and COM port.

```
ESP32_BT_F9C_v2 | Arduino 1.8.12
ファイル 編集 スケッチ ツール ヘルプ
自動整形 Ctrl+T
スケッチをアーカイブする
エンコーディングを修正
ライブラリを管理... Ctrl+Shift+I
シリアルモニタ Ctrl+Shift+M
シリアルプロッタ Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ボード: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "なし"
PSRAM: "Disabled"
シリアルポート: "COM15"
ボード情報を取得
通信装置: "AVRISP mkII"
ブートローダを書き込む
#define packTimeout 5
#define PACKET_SIZE 2048
byte buf1[PACKET_SIZE];
int i1=0;
byte buf2[PACKET_SIZE];
int i2=0;
int RTCM_length=0;
int bt_len=0;
int bufsize=0;
void setup() {
```

# 4. Bluetooth setting

7. Change Bluetooth device name as you like.
8. Write sketch to the board.  
After writing completed you can disconnect USB from PC.

```
ESP32_BT_F9C_v2 | Arduino 1.8.12
ファイル 編集 スケッチ ツール ヘルプ
ESP32_BT_F9C_v2
HardwareSerial uart(2); // UART2を使う->F9P

#define packTimeout 5
#define PACKET_SIZE 2048
byte buf1[PACKET_SIZE];
int i1=0;
byte buf2[PACKET_SIZE];
int i2=0;
int RTCM_length=0;
int bt_len=0;
int bufsize=0;

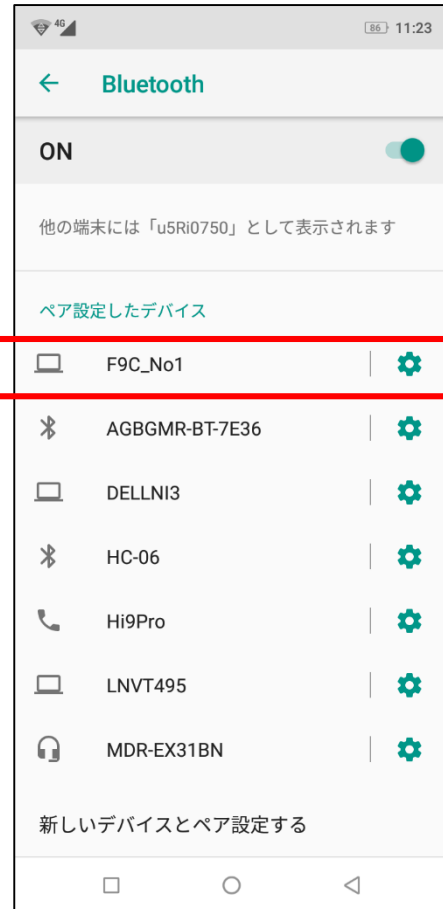
void setup() {
  Serial.begin(115200);
  uart.begin(115200);
  SerialBT.begin("F9C_No1"); //Bluetooth device name
  Serial.println("The device started, now you can pair it with bluetooth!");
  xTaskCreatePinnedToCore(loop2, "loop2", 8192, NULL, 1, NULL, ARDUINO_RUNNING_CORE);
}

void loop2(void *pvParameters){
  while(1){
    if (uart.available()) {
      buf2[i2] = uart.read(); // read char from UART
      if(i2<PACKET_SIZE-1) {
        i2++;
      }
    }else{
      SerialBT.write(buf2,i2);
      i2=0;
    }
  }
}
}

ボードへの書き込みが完了しました。
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 768.0 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
36 COM15のESP32 Dev Module
```

# 4. Bluetooth setting

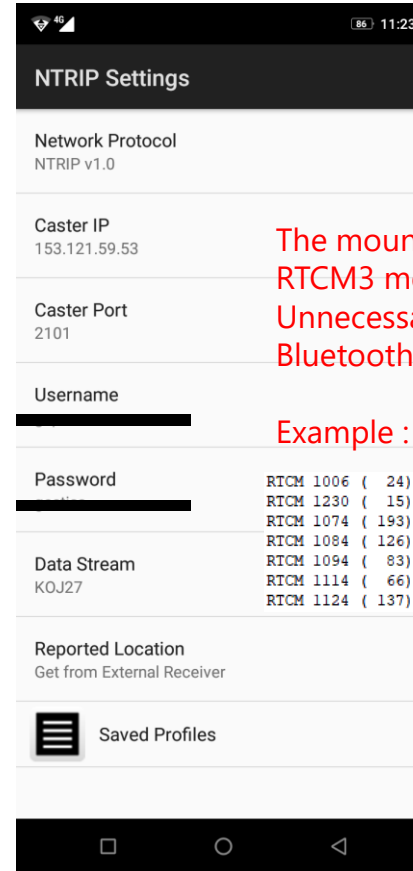
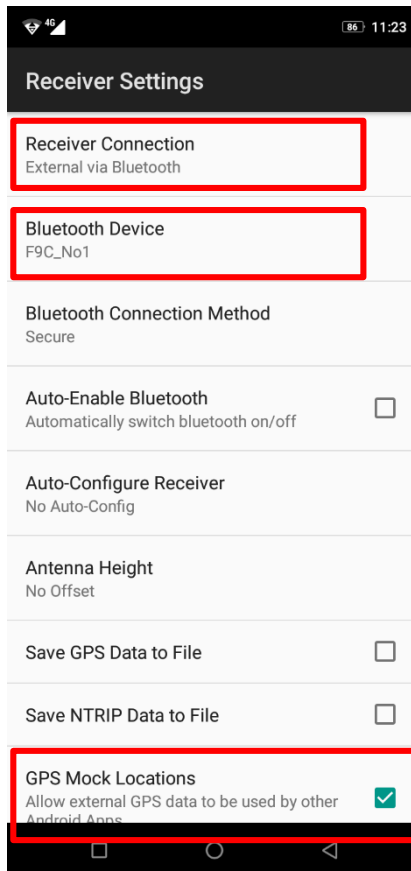
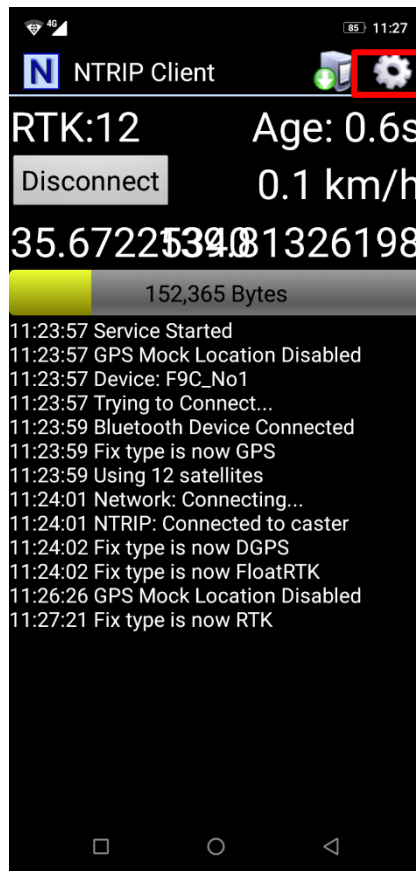
- 9. Power supply to F9C.
- 10. Pairing F9C with your smartphone.





# 4. Bluetooth setting

11. Install "Lefebure NTRIP Client" from Google play store on your Android.  
Set receiver as F9C and configure NTRIP base station, Then click "Connect".  
You can use RTK position on your smartphone app if you enable "GPS Mock Locations".



The mount point should broadcast minimal RTCM3 messages. Unnecessary messages will cause a stack of Bluetooth communication.

Example :

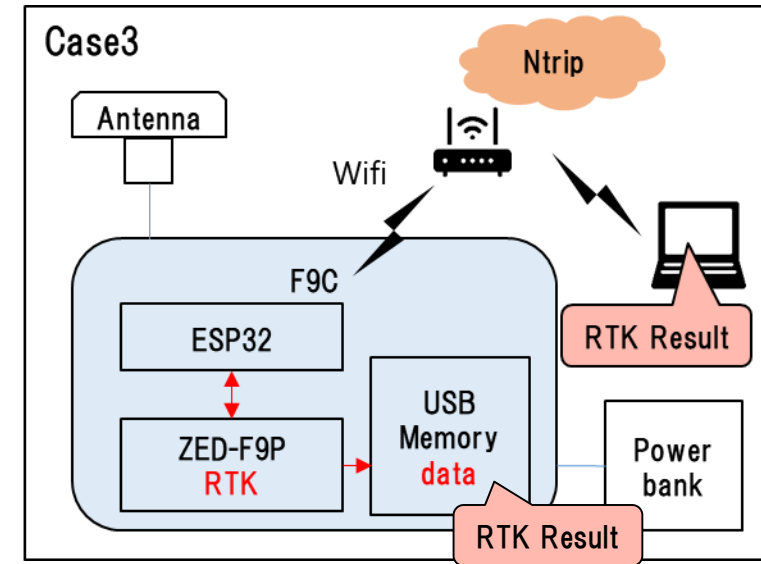
```
RTCM 1006 ( 24): staid= 0 pos=35.55193529 139.64700590 94.419 anth=0.000
RTCM 1230 ( 15):
RTCM 1074 ( 193): staid= 0 2020/04/20 02:43:47.00 nsat= 9 nsig= 4 iod= 0 ncell=24 sync=1
RTCM 1084 ( 126): staid= 0 2020/04/20 02:43:47.00 nsat= 7 nsig= 2 iod= 0 ncell=14 sync=1
RTCM 1094 ( 83): staid= 0 2020/04/20 02:43:47.00 nsat= 4 nsig= 2 iod= 0 ncell= 8 sync=1
RTCM 1114 ( 66): staid= 0 2020/04/20 02:43:47.00 nsat= 2 nsig= 3 iod= 0 ncell= 6 sync=1
RTCM 1124 ( 137): staid= 0 2020/04/20 02:43:47.00 nsat= 9 nsig= 2 iod= 0 ncell=15 sync=0
```

# 5. Wifi setting

F9C logger can get RTK position and store it in wifi coverage.  
The other device in the same LAN can monitor F9C status.



With mobile router



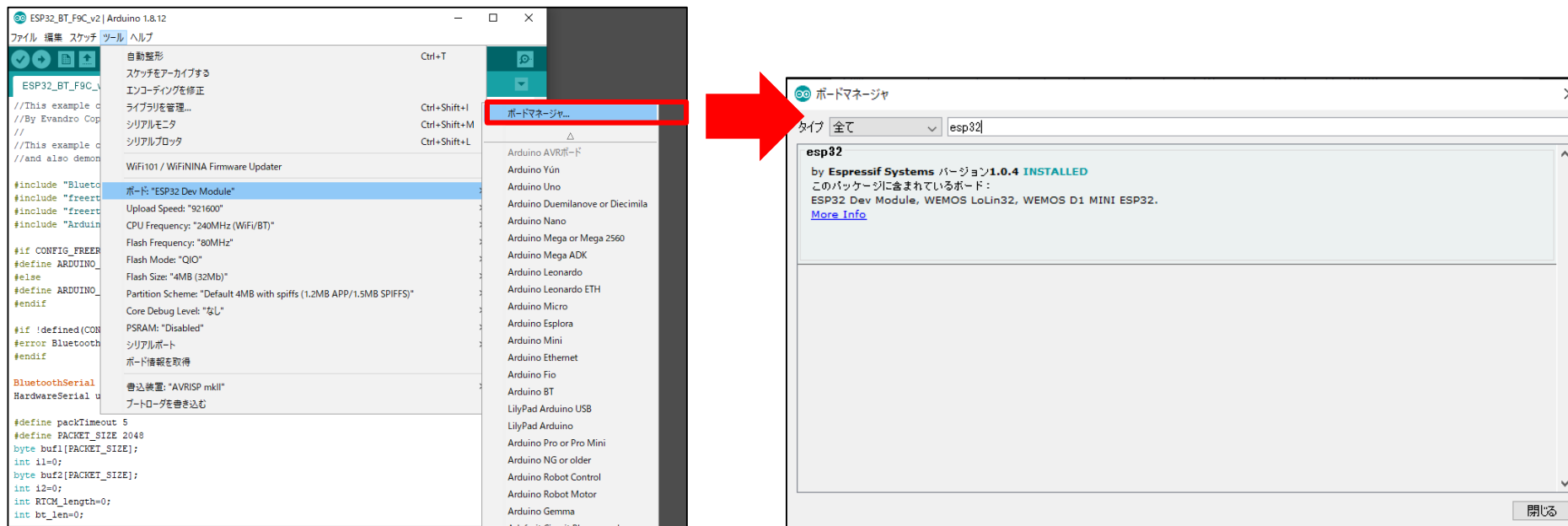
OR



RTK in Wifi coverage field

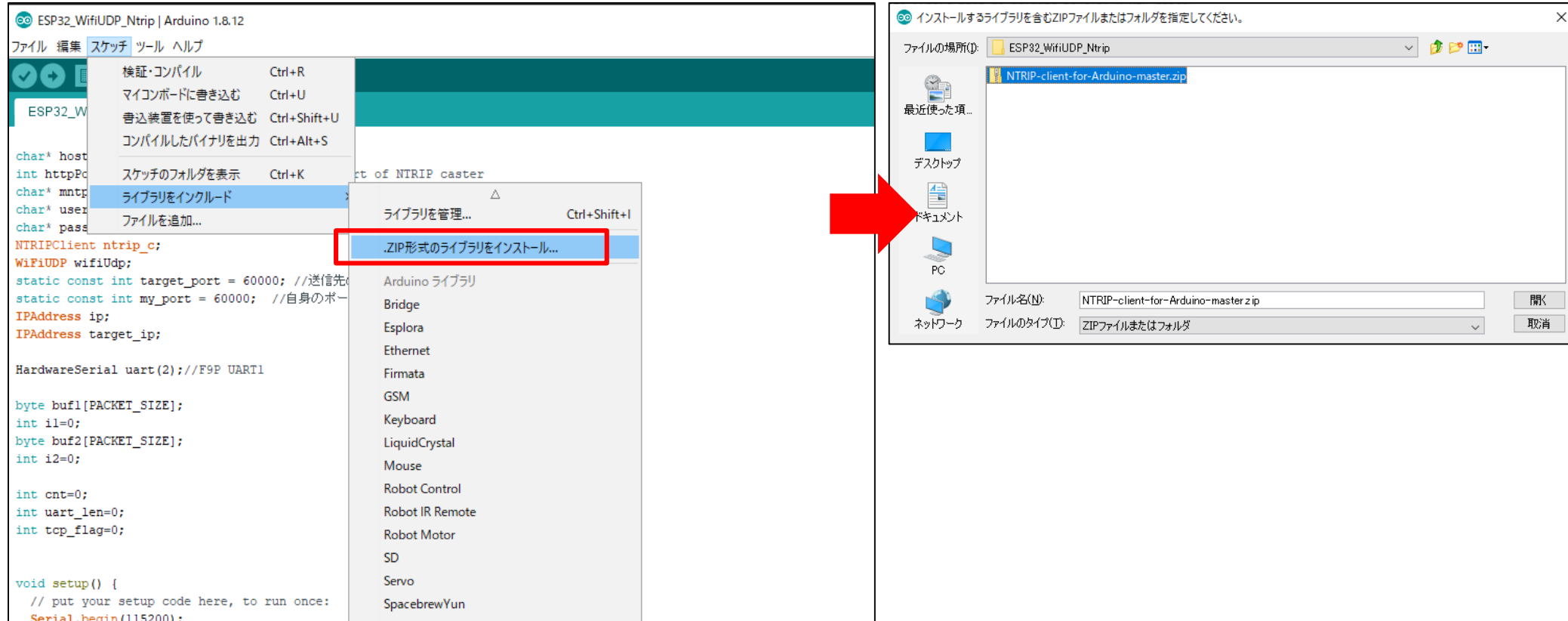
# 5. Wifi setting

1. Install Arduino IDE in your PC.  
<https://www.arduino.cc/en/main/software>
2. Download the sketch for ESP32 and open the sketch "ESP32\_WifiUDP\_Ntrip.ino" .  
[http://www.denshi.e.kaiyodai.ac.jp/gnss\\_tutor/pdf/F9C/ESP32\\_WifiUDP\\_Ntrip.zip](http://www.denshi.e.kaiyodai.ac.jp/gnss_tutor/pdf/F9C/ESP32_WifiUDP_Ntrip.zip)
3. Install the library for ESP32.  
Tool>Board>Board Manager. Then search by "esp32" and install the library.



# 5. Wifi setting

4. Install the library "Ntrip-clilent-for-Arduino-master.zip" from zip file. (In the download folder)



# 5. Wifi setting

- 5. Connect F9C with PC by USB cable.  
Select board type as "ESP32 Dev Module"  
and COM port.

A screenshot of the Arduino IDE interface. The top menu bar shows 'ファイル', '編集', 'スケッチ', 'ツール', and 'ヘルプ'. The 'ツール' menu is open, showing various options. Two options are highlighted with red boxes: 'ボード: "ESP32 Dev Module"' and 'シリアルポート: "COM15"'. The background shows the Arduino IDE code editor with C++ code for an ESP32 module. The code includes headers for Bluetooth and Arduino, and defines various parameters like board type, upload speed, and serial port.

```
ESP32_BT_F9C_v2 | Arduino 1.8.12
ファイル 編集 スケッチ ツール ヘルプ
自動整形 Ctrl+T
スケッチをアーカイブする
エンコーディングを修正
ライブラリを管理... Ctrl+Shift+I
シリアルモニタ Ctrl+Shift+M
シリアルプロッタ Ctrl+Shift+L
WiFi101 / WiFiNINA Firmware Updater
ボード: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "なし"
PSRAM: "Disabled"
シリアルポート: "COM15"
ボード情報を取得
通信装置: "AVRISP mkII"
ブートローダを書き込む
BluetoothSerial
HardwareSerial u
#define packTimeout 5
#define PACKET_SIZE 2048
byte buf1[PACKET_SIZE];
int i1=0;
byte buf2[PACKET_SIZE];
int i2=0;
int RTCM_length=0;
int bt_len=0;
int bufsize=0;
void setup() {
```

# 5. Wifi setting

6. Change following line in the sketch.

- WiFi SSID (2.4GHz)
- WiFi password
- Ntrip Server IP address
- Ntrip Mount Point
- Ntrip User name and password  
(Set blank it when you use Mount Point which don't require user name and password)

```
ESP32_WifiUDP_Ntrip | Arduino 1.8.12
ファイル 編集 スケッチ ツール ヘルプ

ESP32_WifiUDP_Ntrip $
#include <NTRIPClient.h>
#include <WiFi.h>
#include <WiFiUDP.h>
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "Arduino.h"

#if CONFIG_FREERTOS_UNICORE
#define ARDUINO_RUNNING_CORE 0
#else
#define ARDUINO_RUNNING_CORE 1
#endif

#define packTimeout 5
#define PACKET_SIZE 4096

const char* ssid = "*****"; //2.4GHz Wifi SSID
const char* password = "*****"; //Wifi password

char* host = "153.121.59.53"; //Ntrip server xxx.xxx.xxx.xxx
int httpPort = 2101; //port 2101 is default port of NTRIP caster
char* mntpnt = "*****"; //Mount Point Name
char* user = "*****"; //User Name
char* passwd = "*****"; //Password

NTRIPClient ntrip_c;
WiFiUDP wifiUdp;
static const int target_port = 60000; //送信先のポート
static const int my_port = 60000; //自身のポート
IPAddress ip;
IPAddress target_ip;

HardwareSerial uart(2); //F9P UART1

byte buf1[PACKET_SIZE];
int i1=0;

「NTRIP-client-for-Arduino-master」というライブラリはすでに存在します。 エラーメッセージをコピーする

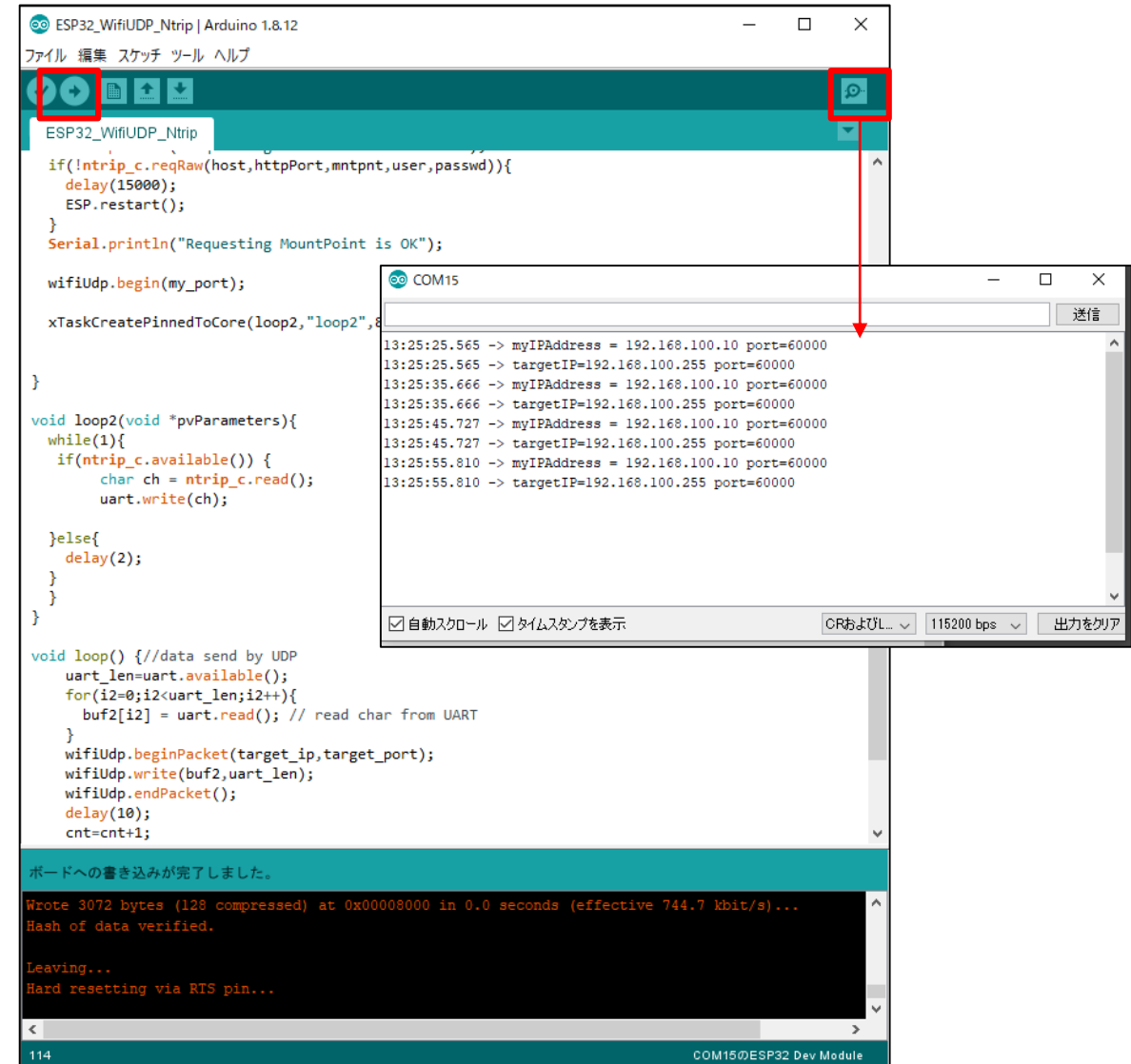
5 COM15のESP32 Dev Module
```

# 5. Wifi setting

## 7. Write the sketch to F9C.

You can find the IP address of F9C on Arduino IDE serial monitor after writing the sketch.

When F9C is power supplied in the WiFi coverage, it automatically starts RTK and stores its result.



# 5. Wifi setting

8. If you want to monitor RTK result, listen UDP 60000 port in the same LAN as F9C. Here, we show the example using RTKLIB and u-center on PC. (F9C will broadcast data to xxx.xxx.xxx.255 of your LAN. Some firewall setting to allow UDP broadcasting may be required on your WiFi router or PC)

The image displays the STRSVR software interface with several windows open. A WiFi router icon is shown above the main window. The main window shows the following data:

Stream	Type	Opt	Cmd	Conv	Bytes	Bps
(0) Input	UDP Server	...	...	...	656,611	0
(1) Output	TCP Server	...	...	...	717,646	0
(2) Output		...	...	...	0	0
(3) Output		...	...	...	0	0

The 'UDP Server Options' dialog box is open, showing:

- Server Address: [Empty]
- Port: 60000
- Mountpoint: [Empty]
- User-ID: [Empty]
- Password: [Masked]
- String: [Empty]

The 'TCP Server Options' dialog box is also open, showing:

- Server Address: [Empty]
- Port: 10001
- Mountpoint: [Empty]
- User-ID: [Empty]
- Password: [Empty]
- String: [Empty]

The 'Packet Console' window shows the following log entries:

```
04:54:57 R -> NMEA GNVTG, Size 35, 'Course Over Ground and Ground Speed'
04:54:57 R -> NMEA GNGGA, Size 89, 'Global Positioning System Fix Data'
04:54:58 R -> NMEA GNRMC, Size 74, 'Recommended Minimum Specific GNSS Data'
04:54:58 R -> NMEA GNVTG, Size 35, 'Course Over Ground and Ground Speed'
04:54:58 R -> NMEA GNGGA, Size 89, 'Global Positioning System Fix Data'
04:54:58 R -> NMEA GNRMC, Size 74, 'Recommended Minimum Specific GNSS Data'
04:54:58 R -> NMEA GNVTG, Size 35, 'Course Over Ground and Ground Speed'
04:54:58 R -> NMEA GNGGA, Size 89, 'Global Positioning System Fix Data'
04:54:58 R -> NMEA GNRMC, Size 74, 'Recommended Minimum Specific GNSS Data'
04:54:58 R -> NMEA GNVTG, Size 35, 'Course Over Ground and Ground Speed'
04:54:58 R -> NMEA GNGGA, Size 89, 'Global Positioning System Fix Data'
04:54:58 R -> NMEA GNRMC, Size 74, 'Recommended Minimum Specific GNSS Data'
04:54:58 R -> NMEA GNVTG, Size 35, 'Course Over Ground and Ground Speed'
04:54:58 R -> NMEA GNGGA, Size 89, 'Global Positioning System Fix Data'
04:54:58 R -> NMEA GNRMC, Size 74, 'Recommended Minimum Specific GNSS Data'
04:54:59 R -> NMEA GNRMC, Size 74, 'Recommended Minimum Specific GNSS Data'
```

The 'Deviation Map' window shows a circular plot with a green dot at the center, representing the deviation of the receiver. The plot has concentric circles labeled 0.5m and 1m. The 'u-center 20.01' window shows a deviation map, a compass, and a speedometer. The speedometer shows a speed of 0.00 m/s = 0.0 km/h. The status bar at the bottom shows 'Ready', 'NTRIP client: Not connected', 'u-blox Generation 4', 'tcp://127.0.0.1 No file open', 'NMEA 00:27:5:04:55:0', and '04/20/2020'.



## 6. Erase the sketch of ESP32

How to put back F9C to not use WiFi or Bluetooth

1. Connect F9C to PC by USB cable and open Arduino IDE.
2. Create new sketch from File>New File.
3. Write the null sketch to ESP32.

