

# センサーのデータをELTRESアドオンボード経由で CLIPサーバに送信

# 更新履歴

| 版数  | 更新内容        | 更新日        |
|-----|-------------|------------|
| 0.2 | Refsを追加した。  | 2023/01/11 |
| 0.1 | 初版を0.1版とする。 | 2022/05/25 |
|     |             |            |
|     |             |            |
|     |             |            |
|     |             |            |
|     |             |            |
|     |             |            |

# 目次

1. ELTRESアドオンボード概要
2. CLIP Viewer Lite コンテンツを確認する
3. 取扱説明書を確認する
4. CLIP Viewer Lite にデバイスIDを確認する
5. Cresco社提供したサンプルプログラムを確認する
6. こちらが作成したサンプルのご紹介
  1. Spresense + ELTRESアドオンボード + BMP280 + CLIP View Liteシステム構成図
  2. サンプル概要及び動作確認
  3. 取得したデータをCLIP Viewer Liteで確認する
7. 注意点
  1. GPSペイロードを確認する
  2. 温度・湿度・CO2 ペイロードを確認する
8. 注意点纏め

# ELTRESアドオンボード概要

ELTRES Add-onボードは、LPWAとGNSSアンテナコネクタを実装したSpresense用ELTRES通信Add-onボードです。Spresenseと組み合わせてエッジAI+IoT環境を構築できます。最大4ヶ月のELTRES通信とクラウド型データ解析ツール CLIP Viewer Liteが付属します。

※ ELTRES通信回線と契約が必要です。



[Refs]

<https://www.cresco-dt.co.jp/service/iot/iot-poc/eltres/>

# CLIP Viewer Lite コンテンツを確認する

※事前にELTRESアドオンボードを購入＆回線開通が必要です。

1. CLIP Viewer Lite : 操作マニュアル
  1. [CLIPViewerLite\\_Ver1.6.x\\_操作マニュアル.pdf](#)
2. CLIP Viewer Lite : API仕様書
  1. [CLIPViewerLite\\_Ver1.6.x\\_API仕様書.pdf](#)
3. ペイロードデータフォーマット仕様説明書
  1. [ペイロードデータフォーマット仕様説明書\\_Ver1.6.x.pdf](#)
4. ELTRESアドオンボード : 取扱説明書
  1. [ELTRESアドオンボード\\_取扱説明書.pdf](#)
5. ELTRESアドオンボード : ライブラリ説明書
  1. <Spresense SDK版>
  2. [ELTRESアドオンボード用ライブラリ\\_v1.1.x\\_説明書\(SpresenseSDK版\).pdf](#)
  3. <Arduino IDE版>
  4. [ELTRESアドオンボード用ライブラリ\\_v1.1.x\\_説明書\(ArduinoIDE版\).pdf](#)
6. ELTRESアドオンボード : ライブラリ
  1. <Spresense SDK版>
  2. [ELTRESアドオンボード用ライブラリ\\_v1.1.0\(SpresenseSDK版\).zip](#)
  3. <Arduino IDE版>
  4. [ELTRESアドオンボード用ライブラリ\\_v1.1.0\(ArduinoIDE版\).zip](#)

1. データをCLIPサーバーへUploadしてから確認するために参照します。
2. CLIP Viewer Liteを使わずに、独自のアプリでデータを表示したい場合参照します。
3. データをCLIPサーバーへUploadために、ペイロードの設定方法を参照します。
4. ELTRESアドオンボードを入手して、ハードウェアの接続方法を参照します。
5. デバイス側でアプリを作成するときに使うライブラリ説明資料です。
6. 5.で説明対象となるライブラリとなります。

# CLIP Viewer Lite コンテンツを確認する

## 1. ELTRESアドオンボード：サンプルプログラム

- <Spresense SDK版>
  1. ELTRESアドオンボード用サンプルプログラム\_ELTTRES送信編 (SpresenseSDK版).zip
  2. ELTRESアドオンボード用サンプルプログラム\_緯度経度BCD変換編 (SpresenseSDK版).zip
- <Arduino IDE版>
  1. ELTRESアドオンボード用サンプルプログラム\_ELTTRES送信編(ArduinoIDE版).zip
  2. ELTRESアドオンボード用サンプルプログラム\_緯度経度BCD変換編 (ArduinoIDE版).zip

```
12  * @brief ペイロードへ緯度経度を格納
13  * @param gga_info ELTRESアドオンボード用ライブラリから取得できるGGA情報
14  * @param payload_data ペイロード
15  */
16  void setup_lat_lon_into_payload(eltres_board_gga_info gga_info, uint8_t payload_data[16]) {
17      String lat_string = String((char*)gga_info.m_lat);
18      String lon_string = String((char*)gga_info.m_lon);
19      int index;
20
21      // 緯度設定
22      index = 0;
23      payload_data[1] = (uint8_t)((((lat_string.substring(index, index+1).toInt() << 4)
24      + lat_string.substring(index+1, index+2).toInt()) & 0xff));
25      index += 2;
26      payload_data[2] = (uint8_t)((((lat_string.substring(index, index+1).toInt() << 4)
27      + lat_string.substring(index+1, index+2).toInt()) & 0xff));
28      index += 2;
29      index += 1; // skip "."
30      payload_data[3] = (uint8_t)((((lat_string.substring(index, index+1).toInt() << 4)
31      + lat_string.substring(index+1, index+2).toInt()) & 0xff));
32      index += 2;
33      payload_data[4] = (uint8_t)((((lat_string.substring(index, index+1).toInt() << 4)
34      + lat_string.substring(index+1, index+2).toInt()) & 0xff));
```

1. ELTRESアドオンボード用サンプルプログラム  
\_ELTTRES送信編
  1. カウント数字1~5順番で1分ごとに  
CLIPサーバーに送信するサンプルで  
す。
2. ELTRESアドオンボード用サンプルプログラム  
\_緯度経度BCD変換編
  1. 緯度経度をBCD変換して、ペイロー  
ドに設定するサンプルです。
  2. ※BCD(Binary-coded decimal)と  
は、10進数の1桁を、0から9までを表  
す2進数の4桁で表現したものです。

# CLIP Viewer Lite コンテンツを確認する

- センサ系：サンプルプログラム

1. SCD41搭載CO2/温度/湿度センサ

- ・ <Spresense SDK版>
- ・ [センサ系サンプルプログラム\\_SCD41搭載CO2温度湿度センサ\(SpresenseSDK版\).zip](#)
- ・ <Arduino IDE版>
- ・ ※Githubにあるライブラリ・説明書
- ・ <https://github.com/Sensirion/arduino-i2c-scd4x>※Githubにあるサンプルプログラム
- ・ <https://github.com/Sensirion/arduino-i2c-scd4x/blob/master/examples/exampleUsage/exampleUsage.ino>
- ・ 【購入先リンク】
- ・ <https://nextstep.official.ec/items/57108630>

2. VL53L0X搭載Time-of-Flight距離センサ

- ・ <Spresense SDK版>
- ・ [センサ系サンプルプログラム\\_VL53L0X搭載Time-of-Flight距離センサ\(SpresenseSDK版\).zip](#)
- ・ <Arduino IDE版>
- ・ ※Githubにあるライブラリ・説明書
- ・ <https://github.com/pololu/vl53l0x-arduino>※Githubにあるサンプルプログラム
- ・ <https://github.com/pololu/vl53l0x-arduino/blob/master/examples/Continuous/Continuous.ino>
- ・ 【購入先リンク】
- ・ <https://www.switch-science.com/catalog/2894/>

# CLIP Viewer Lite コンテンツを確認する

- センサ系：サンプルプログラム
  - BM1383AGLV搭載気圧センサ
    - <Spresense SDK版>
    - ※Githubにあるサンプルプログラム
    - <https://github.com/sonydevworld/spresense/tree/master/examples/press>
    - <Arduino IDE版>
    - ※Githubにあるライブラリ・説明書
    - <https://github.com/RohmSemiconductor/Arduino> ※Githubにあるサンプルプログラム
    - <https://github.com/RohmSemiconductor/Arduino/blob/master/BM1383AGLV/examples/BM1383AGLV/BM1383AGLV.ino>
    - 【購入先リンク】
    - [https://www.chip1stop.com/view/searchResUpt/InventryInfo?unqPartId=ROHM\\*0085974&unqPartNm=SPRESENSE-SENSOR-EVK-701&partId=ROHM-0170579](https://www.chip1stop.com/view/searchResUpt/InventryInfo?unqPartId=ROHM*0085974&unqPartNm=SPRESENSE-SENSOR-EVK-701&partId=ROHM-0170579)
  - KX126搭載加速度センサ
    - <Spresense SDK版>
    - センサ系サンプルプログラム\_KX126搭載加速度センサ(SpresenseSDK版).zip
    - <Arduino IDE版>
    - ※Githubにあるライブラリ・説明書
    - <https://github.com/RohmSemiconductor/Arduino> ※Githubにあるサンプルプログラム
    - <https://github.com/RohmSemiconductor/Arduino/blob/master/KX126/examples/KX126/KX126.ino>
    - 【購入先リンク】
    - [https://www.chip1stop.com/view/searchResUpt/InventryInfo?unqPartId=ROHM\\*0085974&unqPartNm=SPRESENSE-SENSOR-EVK-701&partId=ROHM-0170579](https://www.chip1stop.com/view/searchResUpt/InventryInfo?unqPartId=ROHM*0085974&unqPartNm=SPRESENSE-SENSOR-EVK-701&partId=ROHM-0170579)



# CLIP Viewer Lite コンテンツを確認する

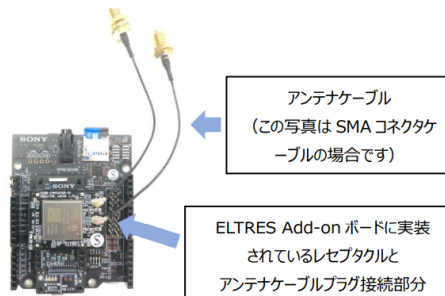
- センサ系：サンプルプログラム
  - RPR-0521RS搭載照度・近接一体型センサ
    - <Spresense SDK版>
    - ※Githubにあるサンプルプログラム
    - ・照度センサ
    - <https://github.com/sonydevworld/spresense/tree/master/examples/light>・近接センサ
    - <https://github.com/sonydevworld/spresense/tree/master/examples/proximity> <Arduino IDE版>
    - ※Githubにあるライブラリ・説明書
    - <https://github.com/RohmSemiconductor/Arduino>※Githubにあるサンプルプログラム
    - <https://github.com/RohmSemiconductor/Arduino/blob/master/RPR-0521RS/examples/RPR-0521RS/RPR-0521RS.ino>
    - 【購入先リンク】
    - [https://www.chip1stop.com/view/searchResUpt/InventoryInfo?unqPartId=ROHM\\*0085227&unqPartNm=RPR-0521RS-EVK-001&partId=ROHM-0169376&zaikoFlg=true](https://www.chip1stop.com/view/searchResUpt/InventoryInfo?unqPartId=ROHM*0085227&unqPartNm=RPR-0521RS-EVK-001&partId=ROHM-0169376&zaikoFlg=true)
  - LSM6DSO搭載ジャイロセンサ
    - <Spresense SDK版>
    - ※Githubにあるサンプルプログラム
    - センサ系サンプルプログラム\_LSM6DSO搭載ジャイロセンサ(SpresenseSDK版).zip
    - <Arduino IDE版>
    - ※Githubにあるライブラリ・説明書
    - [https://github.com/sparkfun/SparkFun\\_Qwiic\\_6DoF\\_LSM6DSO\\_Arduino\\_Library/](https://github.com/sparkfun/SparkFun_Qwiic_6DoF_LSM6DSO_Arduino_Library/)※Githubにあるサンプルプログラム
    - [https://github.com/sparkfun/SparkFun\\_Qwiic\\_6DoF\\_LSM6DSO\\_Arduino\\_Library/blob/main/examples/Basic\\_Readings/Basic\\_Readings.ino](https://github.com/sparkfun/SparkFun_Qwiic_6DoF_LSM6DSO_Arduino_Library/blob/main/examples/Basic_Readings/Basic_Readings.ino)
    - 【購入先リンク】
    - <https://www.switch-science.com/catalog/7182/>

# 取扱説明書を確認する

- ELTRESアドオンボードをSpresenseに搭載する際にまずは取扱説明書を確認が必要です。ELTRESアドオンボードとSpresenseとの接続方法を参照します。
- 実際にこちらで確認した際に、注意点がありませんでした。
- まずはハードウェアに関して、2セットのアンテナがありました(**注意点1**)。
  - SMAコネクタケーブル用アンテナ
  - GNSS 用受信アンテナ、LPWA 用アンテナ
- 2セットのどちらを使うか、取扱説明書に明確に記載していないため、初めて使う場合迷いました。実際動作確認して、2セット両方とも使えることを確認しました。
- GNSS,LPWA それぞれのアンテナケーブルプラグは同じなので、逆にしないように注意が必要です(**注意点2**)。

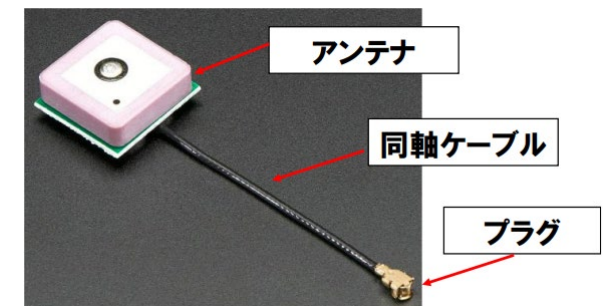
セット1

GNSSとLPWA アンテナ

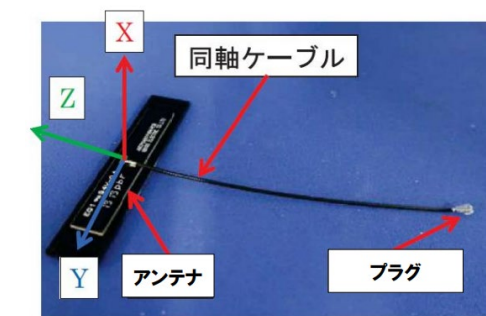


セット2

GNSS 用受信アンテナ



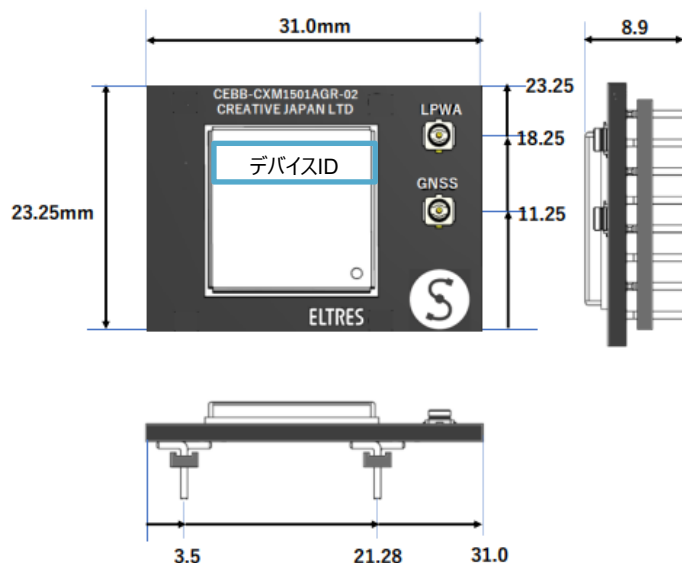
LPWA 用アンテナ



ANT2309-231B/U-W-L-100

# CLIP Viewer Lite にデバイスIDを確認する

※ELTRES回線を開通したら、CLIP Viewer Liteに登録するためのアカウントとパスワードを入力できる



1. CLIP Viewer Liteに複数のデバイスがある場合、自分が使っているデバイスがどれか迷いました(**注意点3**)。
2. 左記はCLIP Viewer LiteとELTRESアドオンボードに記載されているデバイスIDです。
3. ELTRESアドオンボードにデバイスIDが記載されています。それをCLIP Viewer Liteに表示されているデバイスIDと一致するものを選択して、該当のデバイスから取得したデータを表示できます。

# ELTRESアドオンボード：Cresco社提供したサンプルプログラムを確認する

## 1. 環境

- PC
  - Ubuntu 18.04
  - Arduino IDE:v1.8.13
- Spresense Arduino:v2.6.0
- Spresense Main Board
- Spresense 用 BMP280 Add-on ボード
- ELTRES Add-onボード

## 2. セットアップ

1. [Spresense Arduino スタートガイド](#)に記載の手順に従って環境を構築する  
※Spresense Arduino環境インストール済みの場合は実施不要

## 3. ビルド方法

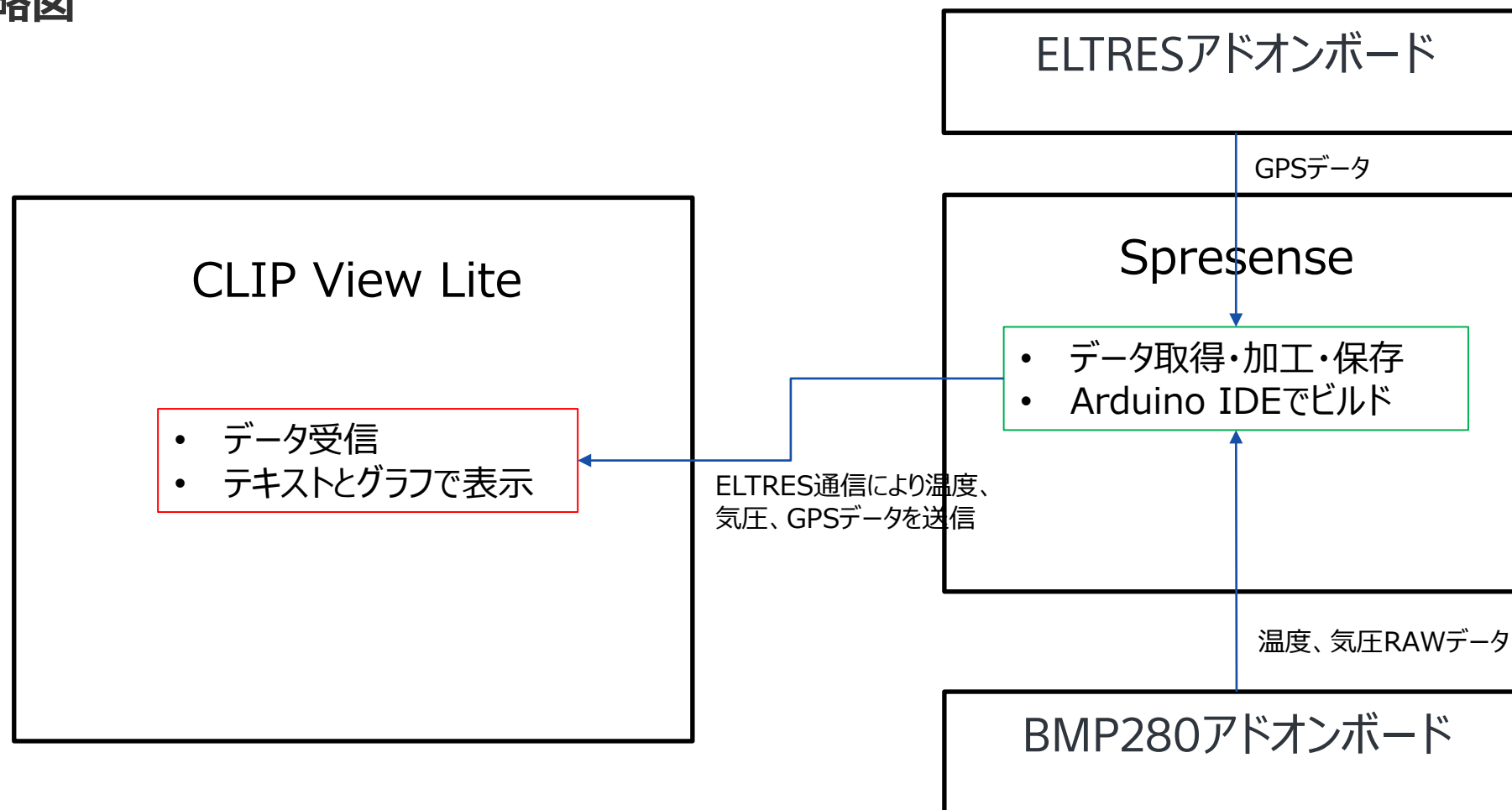
1. [Arduinoソースコードビルド方法](#)を参照して、  
サンプルプログラムをDown Loadして、Arduino IDEで開いてマイコンボードに書き込む ボタンをクリックして、スケッチのコンパイルと書き込みを行います。
2. スケッチの書き込みが完了するまで待ちます。
3. スケッチの書き込みが完了すると自動的にリセットがかかってプログラムが起動されます。

※事前に「**ELTRESアドオンボード：ライブラリ説明書**」を確認が必要です。

| III 項目    ≡ フィルター   |          |                                  |            |        |      |
|---------------------|----------|----------------------------------|------------|--------|------|
| 送信時間 ↑              | ペイロードタイプ | ペイロード                            | デバイスID     | 通信回線   | RSSI |
| 2022-04-26 10:20:57 |          | 01010101010101010101010101010101 | 000101497b | ELTRES | 3    |
| 2022-04-26 10:21:57 |          | 02020202020202020202020202020202 | 000101497b | ELTRES | 1    |
| 2022-04-26 10:22:57 |          | 03030303030303030303030303030303 | 000101497b | ELTRES | 2    |
| 2022-04-26 10:23:57 |          | 04040404040404040404040404040404 | 000101497b | ELTRES | 2    |
| 2022-04-26 10:24:57 |          | 05050505050505050505050505050505 | 000101497b | ELTRES | 2    |

# Spresense + ELTRESアドオンボード + BMP280 + CLIP View Liteシステム構成図

## 1. 概略図



# サンプル概要及び動作確認：

温度、気圧、GPSデータを取得して、1分ずつ気圧→GPS→温度データを繰り返して5回CLIP Viewer LiteにUploadする

## 1. 環境

•PC

Ubuntu 18.04

Arduino IDE:v1.8.13

•Spresense Arduino:v2.6.0

•Spresense Main Board

•Spresense 用 BMP280 Add-on ボード

•ELTRES Add-onボード

## 2. セットアップ

### 1. Spresense Arduino スタートガイド

に記載の手順に従って環境を構築する

※Spresense Arduino環境インストール済みの場合は実施不要

## 3. ビルド方法

### 1. Arduinoソースコードビルド方法

を参照して、[ELTRESのExample](#)をDownloadして、Arduino IDEで開いてマイコンボードに書き込む ボタンをクリックして、スケッチのコンパイルと書き込みを行います。

2. スケッチの書き込みが完了するまで待ちます。

3. スケッチの書き込みが完了すると自動的にリセットがかかってプログラムが起動されます。

```
File Edit Sketch Tools Help
eltres_sample sensor_app.cpp sensor_app.h

1 /*
2  * eltres_sample.ino - 5回ペイロードを送信して待止するサンプルプログラム
3  * Copyright 2021 CREATIVE JAPAN, LTD.
4  */
5 /**
6  * サンプルプログラムは、下記のようにLEDを用いて状態を表現します。
7  * LED0: [プログラム状態] 起動中: 点灯、正常・異常終了: 消灯
8  * LED1: [GNSS電波状態] 受信中: 点灯、未受信: 消灯
9  * LED2: [ELTRES状態] 送信中: 点灯、待機中: 消灯
10 * LED3: [エラー状態] GNSS受信エラー: 点滅、ELTRESエラー: 点灯、エラー無し: 消灯
11 * サンプルプログラムは、以下のように動作します。
12 * ELTRES送信を5回実施して、正常終了状態でループします。
13 * ペイロードには送信回数を格納します。
14 * ※GNSS受信エラー発生時はGNSS電波受信するまで待ち続けます。(LED0は点灯、LED3は点滅)
15 * ※ELTRESエラー発生時は異常終了状態でループします。(LED0は消灯、LED3は点灯)
16 */
17 #include <EltresAddonBoard.h>
18 #include <SSCI_BME280.h>
19 #include <Wire.h>
20 #include "sensor_app.h"
21
22 // PIN定義: LED(プログラム状態)
23 #define LED_RUN_PIN_LED0
24 // PIN定義: LED(GNSS電波状態)
25 #define LED_GNSS_PIN_LED1
26 // PIN定義: LED(ELTRES状態)
27 #define LED_SND_PIN_LED2
28 // PIN定義: LED(エラー状態)
29 #define LED_ERR_PIN_LED3
30
31 // プログラム内部状態: 初期状態
32 #define PROGRAM_STS_INIT (0)
33 // プログラム内部状態: 起動中
34 #define PROGRAM_STS_RUNNING (1)
35 // プログラム内部状態: 終了処理中
36 #define PROGRAM_STS_STOPPING (2)
37 // プログラム内部状態: 終了
38 #define PROGRAM_STS_STOPPED (3)
39
40 // プログラム内部状態
41 int program_sts = PROGRAM_STS_INIT;
42 // GNSS電波受信タイムアウト (GNSS受信エラー) 発生フラグ
43 bool gss_receive_timeout = false;

Done uploading.
Restarting the board ...
Reboot

15 Spresense on /dev/ttyUSB0
```

```
[gga]utc: 060508.00
lat: N3528.0664, lon: E13925.7163
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060509.00
lat: N3528.0667, lon: E13925.7164
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060510.00
lat: N3528.0672, lon: E13925.7163
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060511.00
lat: N3528.0676, lon: E13925.7166
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060512.00
lat: N3528.0675, lon: E13925.7167
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060513.00
lat: N3528.0668, lon: E13925.7166
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060514.00
lat: N3528.0665, lon: E13925.7166
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060515.00
lat: N3528.0666, lon: E13925.7166
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060516.00
lat: N3528.0664, lon: E13925.7168
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060517.00
lat: N3528.0664, lon: E13925.7173
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060518.00
lat: N3528.0664, lon: E13925.7173
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060519.00
lat: N3528.0660, lon: E13925.7171
pos_status: 1, sat_used: 3
hdop: 2.10, height: 40.00 m, geoid: 38.80 m
[gga]utc: 060520.00
lat: N3528.0659, lon: E13925.7167
pos_status: 1, sat_used: 4
hdop: 2.60, height: 68.00 m, geoid: 38.80 m
```

[Refs]

[https://github.com/SonySemiconductorSolutions/ssup-spresense/tree/main/Arduino/CEBB-CXM1501GR-02/eltres\\_sample](https://github.com/SonySemiconductorSolutions/ssup-spresense/tree/main/Arduino/CEBB-CXM1501GR-02/eltres_sample)



# 取得したデータをCLIP Viewer Liteで確認する:テキスト

| 項目                  |            | フィルター |      |   |                    |        |  |
|---------------------|------------|-------|------|---|--------------------|--------|--|
| 送信時間                | ペイロードタイプ   | RSSI  | 電波品質 | GPS                                     | 温度(°C)             | 圧力(Pa) |  |
| 2022-05-24 15:19:57 | 温湿度CO2情報   | 0     |      |   | 31.719999313354492 |        |  |
| 2022-05-24 15:18:57 | GPS情報      | 2     | 1    | <a href="#">35 28.0581 0139 25.7189</a> |                    |        |  |
| 2022-05-24 15:17:57 | 気圧圧力照度距離情報 | 1     |      |   |                    | 0      |  |
| 2022-05-24 15:16:57 | 温湿度CO2情報   | 1     |      |   | 32.15999984741211  |        |  |
| 2022-05-24 15:15:57 | GPS情報      | 2     | 1    | <a href="#">35 28.0572 0139 25.7193</a> |                    |        |  |
| 2022-05-24 15:14:57 | 気圧圧力照度距離情報 | 2     |      |   |                    | 0      |  |
| 2022-05-24 15:13:57 | 温湿度CO2情報   | 1     |      |   | 31.8700008392334   |        |  |
| 2022-05-24 15:12:57 | GPS情報      | 0     | 1    | <a href="#">35 28.0563 0139 25.7187</a> |                    |        |  |
| 2022-05-24 15:11:57 | 気圧圧力照度距離情報 | 1     |      |   |                    | 0      |  |
| 2022-05-24 15:10:57 | 温湿度CO2情報   | 2     |      |   | 32.029998779296875 |        |  |
| 2022-05-24 15:09:57 | GPS情報      | 2     | 1    | <a href="#">35 28.0562 0139 25.7185</a> |                    |        |  |
| 2022-05-24 15:08:57 | 気圧圧力照度距離情報 | 1     |      |   |                    | 0      |  |
| 2022-05-24 15:07:57 | 温湿度CO2情報   | 1     |      |   | 32.040000915527344 |        |  |
| 2022-05-24 15:06:57 | GPS情報      | 1     | 1    | <a href="#">35 28.0544 0139 25.7137</a> |                    |        |  |
| 2022-05-24 15:05:57 | 気圧圧力照度距離情報 | 2     |      |   |                    | 0      |  |

< 1 >

# 取得したデータをCLIP Viewer Liteで確認する:GPSグラフ

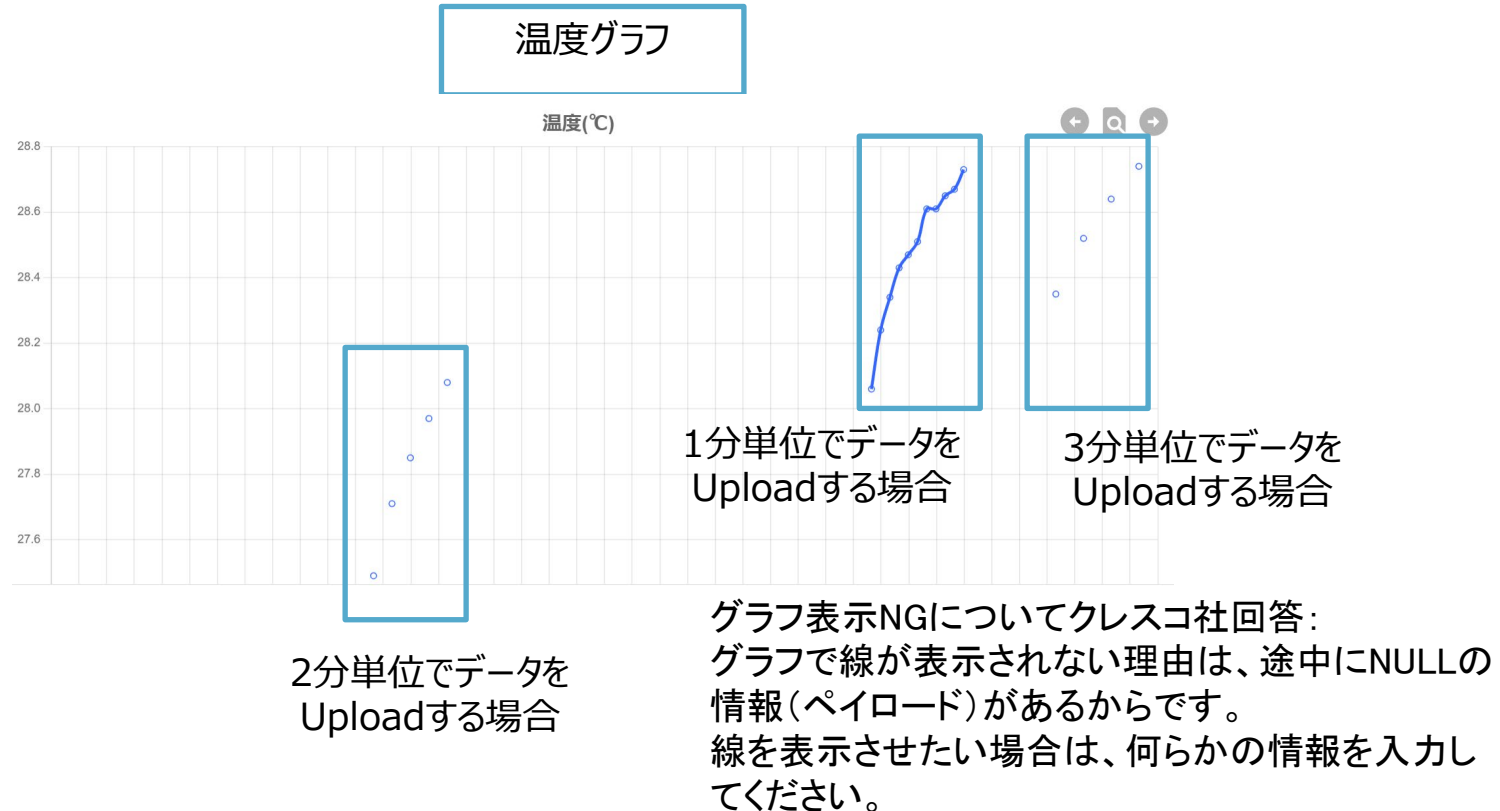




# 取得したデータをCLIP Viewer Liteで確認する:温度グラフ(気圧グラフも同じ)

(注意点7)温度グラフを確認する際に下記のようなグラフが正しく表示できない場合があります(ELTRES契約は1分単位データ送信)。

1. 1分単位で温度データをUploadする場合：グラフ表示OK
2. 2分単位で温度データをUploadする場合：グラフ表示NG  
※プログラムで2分間中に1分単位で温度データを送信する、次の1分単位でGPSデータを送信する
3. 3分単位で温度データをUploadする場合：グラフ表示NG  
※プログラムで3分間中に1分単位で温度データを送信する、次の1分単位でGPSデータを送信する、そしてもう一つの1分単位で気圧データを送信する



# 注意点：GPSペイロードを確認する

## 1. 位置情報

1. 位置情報をBCDに変換する必要があります。変更方法は「[ELTRESアドオンボード用サンプルプログラム\\_緯度経度BCD変換編](#)」を参照できます。
2. ※ (注意点4)提供したサンプルには位置情報はコールバック関数中のローカル変数に格納されているため、ペイロードに設定するには工夫する必要があります(例：グローバル変数を利用する)。

## 2. EPOCH秒

1. ライブラリを利用してEPOCH秒を取得するAPIがあります。
2. ※ (注意点5)提供したサンにはEPOCH秒をそのままペイロードに設定すると、1分遅れてしまいます。
  1. 原因：EPOCH秒取得はペイロードに設定後に取得したため、タイミングが次の送信タイミングになる。
  2. 対策：EPOCH秒取得処理をペイロードに設定する前に変更する。

## 3. 電波品質

1. 取得したGPS情報中に電波品質を利用できます。

## 2.2 No.1：GPSペイロード

| No. | ペイロード名       | ペイロードデータ(16byte) |                  |                  |                                  |                                  |                           |                            |                  |                                  |                                  |                 |        |        |                 |
|-----|--------------|------------------|------------------|------------------|----------------------------------|----------------------------------|---------------------------|----------------------------|------------------|----------------------------------|----------------------------------|-----------------|--------|--------|-----------------|
|     |              | 1                | 2                | 3                | 4                                | 5                                | 6                         | 7                          | 8                | 9                                | 10                               | 11              | 12     | 13     | 14              |
| 1   | GPS<br>ペイロード | 識別<br>コード        | GPS<br>(緯度)<br>度 | GPS<br>(緯度)<br>分 | GPS<br>(緯度)<br>分<br>小数点<br>1位,2位 | GPS<br>(緯度)<br>分<br>小数点<br>3位,4位 | GPS<br>(経度)<br>度<br>100の位 | GPS<br>(経度)<br>度<br>10,1の位 | GPS<br>(経度)<br>分 | GPS<br>(経度)<br>分<br>小数点<br>1位,2位 | GPS<br>(経度)<br>分<br>小数点<br>3位,4位 | EPOCH秒<br>(MSB) | EPOCH秒 | EPOCH秒 | EPOCH秒<br>(LSB) |
|     |              |                  |                  |                  |                                  |                                  |                           |                            |                  |                                  |                                  |                 |        |        | 拡張用             |
|     |              |                  |                  |                  |                                  |                                  |                           |                            |                  |                                  |                                  |                 |        |        | 電波品質            |

### ■ 識別コード

「0x81」

### ■ GPS (GNSS)

位置情報 (BCD)

ELTRES アドオンボード用ライブラリから取得した位置情報 (緯度・経度) を BCD へ変換した値

※詳しくは「[ELTRES アドオンボード用サンプルプログラム\\_緯度経度 BCD 変換編](#)」を参照

### ■ EPOCH 秒

時間情報を EPOCH 秒 (4Byte) へ変換した値

### ■ 拡張用

「0x00」

※固定値入力

### ■ 電波品質

「0x00」：非測位状態

「0x01」：測位状態

「0x02」：測位状態、かつ DGPS による補正が有効

「0x06」：DR (Dead Reckoning)による自律測位

# 注意点：温度・湿度・CO2 ペイロードを確認する

- 1. 温度・湿度・CO2は全部Float型なので、一つの温度データで確認しました。
- 2. ペイロードを確認すると、温度は4byteのuint8\_t temperature[4]型なので、取得したFloat値(例：25.27)はどうやってペイロードの4byteに格納するか迷いました。
  - 1. 最初はGPSのBCD変換方法を参照して確認してみました。結果はCLIP Viewer Liteに正しく表示できなくて、NGでした。
  - 2. そのあとはuint8\_t temperature[4]型をFloat型にキャストしてみましたが、また結果はCLIP Viewer Liteに正しく表示できなくて、NGでした。
  - 3. 最後は試行錯誤した結果OKでした。原因はSpresense側がLSBでデータを保存していますが、CLIP Viewer Liteサーバー側がMSBでデータを処理して表示しています(注意点6、詳細は次のページを参照してください)。

## 2.3 No.2：温度・湿度・CO2 ペイロード

| No. | ペイロード名             | ペイロードデータ(16byte) |    |    |    |    |    |    |    |    |     |     |     |     |     |
|-----|--------------------|------------------|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|
|     |                    | 1                | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  | 11  | 12  | 13  | 14  |
| 2   | 温度・湿度・CO2<br>ペイロード | 識別<br>コード        | 温度 | 湿度 | 湿度 | 湿度 | 湿度 | 湿度 | 湿度 | 湿度 | CO2 | CO2 | CO2 | CO2 | 拡張用 |

- 識別コード  
「0x82」
- 温度  
温度情報 (Float)
- 湿度  
湿度情報 (Float)
- CO2  
CO2 情報 (Float)

4

© CRESCO DIGITAL TECHNOLOGIES, LTD.

- 拡張用  
「0x00」  
※固定値入力

# ELTRESアドオンボード：温度・湿度・CO2 ペイロードを確認する

1. 下記は[こちらのサイト](#)の情報を参照しています。
2. 対策：
  1. まずはuint8\_t temperature[4]型をFloat型にキャストして、リトルエンディアンのデータを取得します。
  2. 次はリトルエンディアンのデータをビッグエンディアンに変換します。

## Spresense側

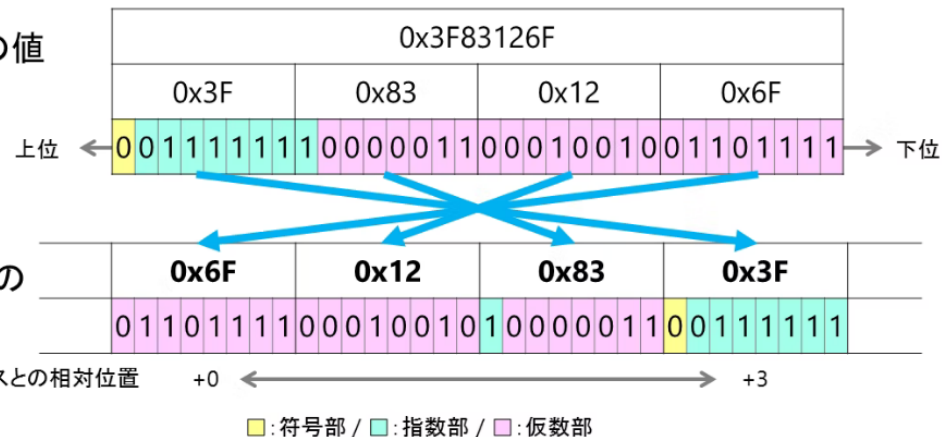


### 浮動小数点数値のメモリ上での並び(リトルエンディアン)

元の値(float型)

1.024

内部表現の値  
(int型)



## CLIP Viewer Liteサーバー側

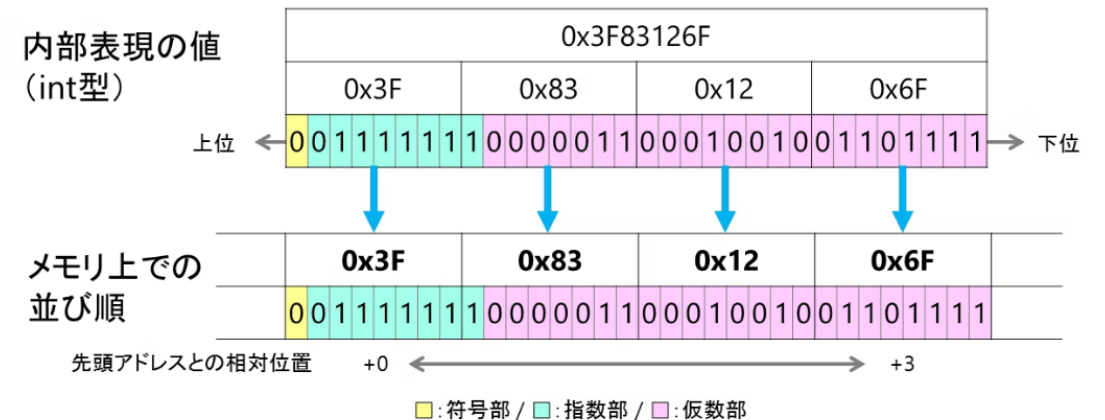


### 浮動小数点数値のメモリ上での並び(ビッグエンディアン)

元の値(float型)

1.024

内部表現の値  
(int型)



[Refs]

[https://qiita.com/nia\\_tn1012/items/340a1f0ad71bf6085f7f](https://qiita.com/nia_tn1012/items/340a1f0ad71bf6085f7f)

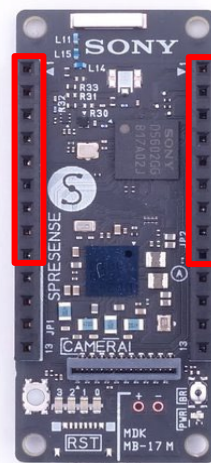
## 注意点8： ELTRESアドオンボードとToF(MM-S50MV) ボードと同時に使用できない

距離データをCLIPサーバーに表示(ToFとELTRES(クリエイティブジャパン)ボードが物理接続衝突のため、確認不可)



**MM-S50MV**  
GND  
UART2\_TX  
UART2\_RX  
UART2\_RTS  
UART2\_CTS  
NC  
NC  
SPI5\_CS\_X  
SPI5\_BCK

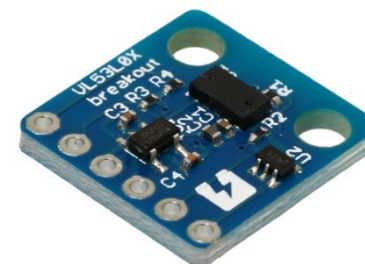
**CXD-5602**  
GND  
UART2\_TX  
UART2\_RX  
UART2\_RTS  
UART2\_CTS  
I2S0\_BCK  
I2S0\_LRCK  
SPI5\_CS\_X  
SPI5\_SCK



**CXD-5602**  
XRST\_PIN\_1.8V  
1.8V  
EXT\_VDD  
EMMC\_DATA3  
EMMC\_DATA2  
I2S0\_DATA\_IN  
I2S0\_DATA\_OUT  
T  
SPI5\_MISO  
SPI5\_MOSI

**MM-S50MV**  
XRST  
1.8V  
3.7V(4V)  
NC  
NC  
NC  
NC  
SPI5\_MISO  
SPI5\_MOSI

※下記のVL53L0X搭載Time-of-Flight距離センサとELTRES(クリエイティブジャパン)基板は同時に使用可能



ELTRES Add-on ボード



空いたスペースにセンサ系  
Add-on ボードを搭載可能  
センサと組み合わせることで  
IoT デバイスとして機能します

Spresense

左 1



[Refs]

<https://developer.sony.com/ja/develop/spresense/specifications>  
<https://shop.sunhayato.co.jp/products/mm-s50mv>  
<https://prtimes.jp/main/html/rd/p/000000067.000064534.html>  
<https://www.switch-science.com/products/2894/>

## 注意点纏め

1. 2セットのアンテナがありますので、取り扱い際にどちらか一つセットで使ってください。
  - SMAコネクタケーブル用アンテナ
  - GNSS 用受信アンテナ、LPWA 用アンテナ
2. GNSS,LPWA それぞれのアンテナケーブルプラグは同じなので、逆にしないように注意が必要です。
3. CLIP Viewer Liteに複数のデバイスがある場合、自分が使っているデバイスがどれかを確認が必要です。
4. 提供したサンプルには位置情報はコールバック関数中のローカル変数に格納されているため、ペイロードに設定するには工夫する必要があります(例：グローバル変数を利用する)。
5. 提供したサンプルにはペイロードの設定処理はEPOCH秒の取得処理により先に実行するので、実際EPOCH秒をペイロードに設定する際に順番を変える必要があります。
6. Spresense側がLSBでデータを保存していますが、CLIP Viewer Liteサーバー側がMSBでデータを処理して表示しています。
7. 温度グラフを確認する際に下記のようなグラフが正しく表示できない場合があります。
8. ELTRESアドオンボードとToF(MM-S50MV) ボードと同時に使用できない。