

CO2センサーボード(SCD41)

更新履歴

版数	更新内容	更新日
0.2	Refsを追加した。	2023/01/12
0.1	初版を0.1版とする。	2022/06/13

目次

1. Spresense 用 CO2センサー Addon ボード概要

1. CO2センサー Addon ボード紹介
2. CO2センサーはどんなデータ取得できるか

2. arduino-i2c-scd4xを動作確認する(Sensirion社提供したサンプル)

1. 動作確認
2. CO2濃度変化の確認
3. 温度キャリブレーション方法について

3. SSUP提供したサンプルについて

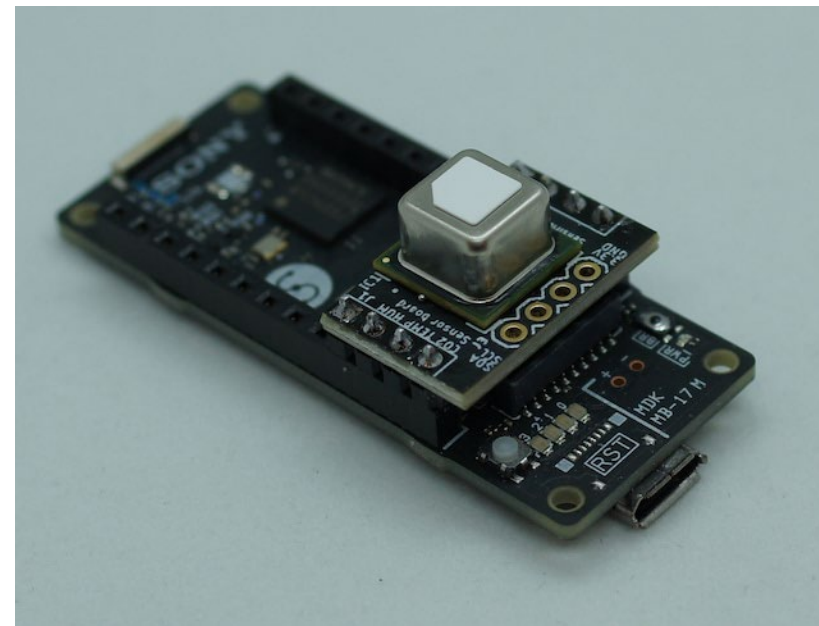
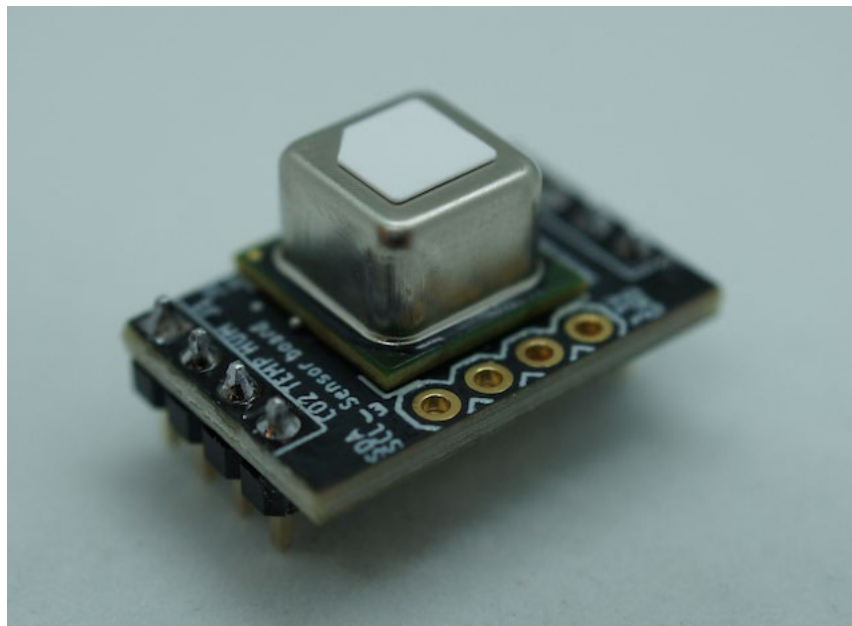
1. CO2(SCD41)データ保存のサンプルプログラムを動作確認する(SSUP提供したサンプル①)
2. CO2(SCD41)データをELTRES通信によりCLIPサーバーに確認する(SSUP提供したサンプル②)

4. ELTRESとCO2との組み合わせでMulticore対応について

5. 注意点纏め

Spresense 用 CO2センサー Addon ボード概要

1. Spresense 用のSensirion社の SCD41 を使ったCO2センサー拡張ボードです。OLED接続用のI2Cソケットを付けると小型のCO2モニターを作ることができます。
1. 本資料はCO2センサーから取得したデータについての資料となります。CO2モニターを作ることとは対象外です。
2. 本体のカメラ接続用のコネクタを隠す形になりますので、カメラも必要な場合はご注意ください(注意点1)。
3. 詳細は <https://nextstep.official.ec/items/57108630>



CO2センサーはどんなデータ取得できるか

- CO2濃度
 - 単位 : ppm
 - 標準精度 : $\pm 40\text{ppm} + 5\% \text{ M.V.}$ (読み値)
 - 範囲 : 400 – 5000
 - 反応時間 ($\tau_{63\%}$)※1 : 60s
- 温度
 - 単位 : $^{\circ}\text{C}$
 - 標準精度 : ± 0.8
 - 範囲 : -10 – 60
 - 反応時間 ($\tau_{63\%}$) : 120s
- 湿度
 - 単位 : %RH(相対湿度)
 - 標準精度 : ± 6
 - 範囲 : 0 – 95
 - 反応時間 ($\tau_{63\%}$) : 120s

詳細は下記を参照してください。
<https://sensirion.com/jp/products/product-catalog/SCD41/>

家や部屋における二酸化炭素濃度の基準

空気中の二酸化炭素濃度は通常**410ppm**とされており、室内の二酸化炭素濃度の基準は**1,000ppm以下**です。

この二酸化炭素濃度は室内の空気汚染を判断する1つの指針となり、この基準値を超えると室内環境が悪い・換気不足と判断されます。

新型コロナウイルス対策でも、二酸化炭素濃度は1,000ppm以下を基準としており、これを保てるような換気が必要です。

詳細は下記を参照してください。

<https://www.njkk.co.jp/blog/?itemid=81&dispmid=764#:~:text=%E7%A9%BA%E6%B0%97%E4%B8%AD%E3%81%AE%E4%BA%8C%E9%85%B8%E5%8C%96%E7%82%AD%E7%B4%A0,%E4%B8%8D%E8%B6%B3%E3%81%A8%E5%88%A4%E6%96%AD%E3%81%95%E3%82%8C%E3%81%BE%E3%81%99%E3%80%82>

※1反応時間 ($\tau_{63\%}$)について
環境は激変する場合、センサー取得した値は真値の63%になるまでかかる時間となります。

arduino-i2c-scd4xを動作確認する(Sensirion社提供したサンプル)

1. 概要

1. CO2(SCD41)センサーからCO2、温湿度データを5秒間隔でデータ取得して表示するサンプルとなります。

2. 環境

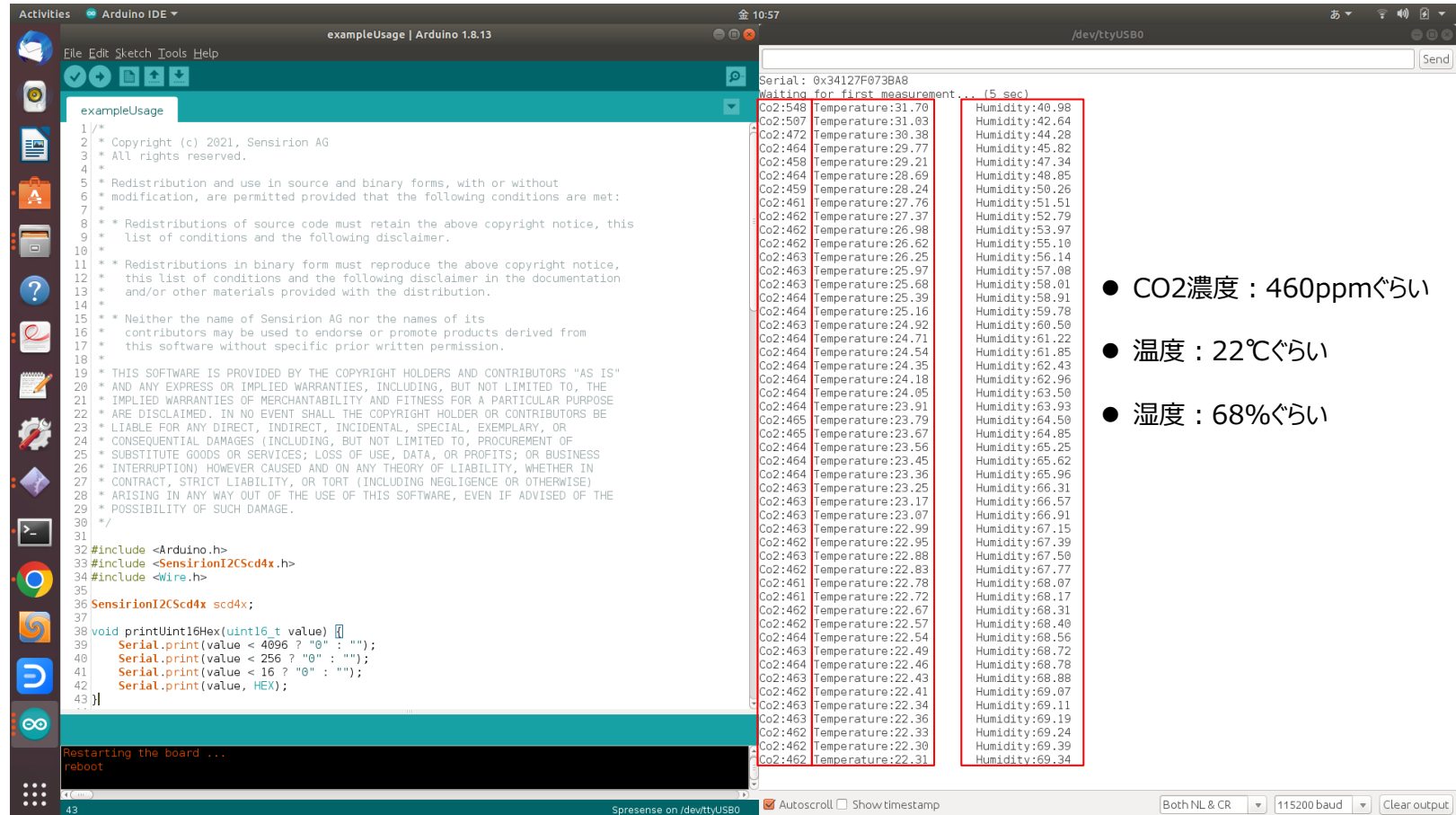
1. PC
 - Ubuntu 18.04
 - Arduino IDE:v1.8.13
2. Spresense Arduino:v2.6.0
3. Spresense Main Board
4. Spresense 用 CO2センサー Addon ボード

3. 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

4. ビルド方法

1. [Arduinoソースコードビルド方法](#)を参照して、[arduino-i2c-scd4x](#)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。



```
exampleUsage | Arduino 1.8.13
File Edit Sketch Tools Help
exampleUsage
1 /*
2  * Copyright (c) 2021, Sensirion AG
3  * All rights reserved.
4  *
5  * Redistribution and use in source and binary forms, with or without
6  * modification, are permitted provided that the following conditions are met:
7  *
8  * * Redistributions of source code must retain the above copyright notice, this
9  *   list of conditions and the following disclaimer.
10 *
11 * * Redistributions in binary form must reproduce the above copyright notice,
12 *   this list of conditions and the following disclaimer in the documentation
13 *   and/or other materials provided with the distribution.
14 *
15 * * Neither the name of Sensirion AG nor the names of its
16 *   contributors may be used to endorse or promote products derived from
17 *   this software without specific prior written permission.
18 *
19 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
20 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
21 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
22 * ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE
23 * LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
24 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
25 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
26 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
27 * CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
28 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
29 * POSSIBILITY OF SUCH DAMAGE.
30 */
31
32 #include <Arduino.h>
33 #include <SensirionI2CScd4x.h>
34 #include <Wire.h>
35
36 SensirionI2CScd4x scd4x;
37
38 void printUInt16Hex(uint16_t value) {
39   Serial.print(value < 4096 ? "0" : "");
40   Serial.print(value < 256 ? "0" : "");
41   Serial.print(value < 16 ? "0" : "");
42   Serial.print(value, HEX);
43 }
44
45 void setup() {
46   Serial.begin(115200);
47   while (!Serial) continue;
48   scd4x.begin();
49   delay(1000);
50   Serial.println("Waiting for first measurement... (5 sec)");
51 }
52
53 void loop() {
54   float co2, temperature, humidity;
55   scd4x.read();
56   co2 = scd4x.getCO2();
57   temperature = scd4x.getTemperature();
58   humidity = scd4x.getHumidity();
59   Serial.print("Co2:");
60   Serial.print(co2);
61   Serial.print(" Temperature:");
62   Serial.print(temperature);
63   Serial.print(" Humidity:");
64   Serial.print(humidity);
65   Serial.println();
66   delay(5000);
67 }
```

Serial: 0x34127F0739A8
Waiting for first measurement... (5 sec)
Co2:548 Temperature:31.70 Humidity:40.98
Co2:507 Temperature:31.03 Humidity:42.64
Co2:472 Temperature:30.38 Humidity:44.28
Co2:464 Temperature:29.77 Humidity:45.82
Co2:458 Temperature:29.21 Humidity:47.34
Co2:464 Temperature:28.69 Humidity:48.85
Co2:459 Temperature:28.24 Humidity:50.26
Co2:461 Temperature:27.76 Humidity:51.51
Co2:462 Temperature:27.37 Humidity:52.79
Co2:462 Temperature:26.98 Humidity:53.97
Co2:462 Temperature:26.62 Humidity:55.10
Co2:463 Temperature:26.25 Humidity:56.14
Co2:463 Temperature:25.97 Humidity:57.08
Co2:463 Temperature:25.68 Humidity:58.91
Co2:464 Temperature:25.39 Humidity:59.78
Co2:464 Temperature:24.92 Humidity:60.50
Co2:464 Temperature:24.71 Humidity:61.22
Co2:464 Temperature:24.54 Humidity:61.85
Co2:464 Temperature:24.35 Humidity:62.43
Co2:464 Temperature:24.18 Humidity:62.96
Co2:464 Temperature:23.91 Humidity:63.50
Co2:465 Temperature:23.79 Humidity:63.93
Co2:465 Temperature:23.67 Humidity:64.50
Co2:464 Temperature:23.56 Humidity:65.25
Co2:464 Temperature:23.45 Humidity:65.62
Co2:464 Temperature:23.36 Humidity:65.96
Co2:463 Temperature:23.25 Humidity:66.31
Co2:463 Temperature:23.17 Humidity:66.57
Co2:463 Temperature:23.07 Humidity:66.91
Co2:463 Temperature:22.99 Humidity:67.15
Co2:462 Temperature:22.95 Humidity:67.39
Co2:463 Temperature:22.88 Humidity:67.58
Co2:462 Temperature:22.83 Humidity:67.77
Co2:461 Temperature:22.78 Humidity:68.07
Co2:461 Temperature:22.72 Humidity:68.17
Co2:462 Temperature:22.67 Humidity:68.31
Co2:462 Temperature:22.57 Humidity:68.40
Co2:464 Temperature:22.54 Humidity:68.56
Co2:463 Temperature:22.49 Humidity:68.72
Co2:464 Temperature:22.46 Humidity:68.78
Co2:463 Temperature:22.43 Humidity:68.88
Co2:462 Temperature:22.41 Humidity:69.07
Co2:463 Temperature:22.34 Humidity:69.11
Co2:463 Temperature:22.36 Humidity:69.19
Co2:462 Temperature:22.33 Humidity:69.24
Co2:462 Temperature:22.30 Humidity:69.39
Co2:462 Temperature:22.31 Humidity:69.34

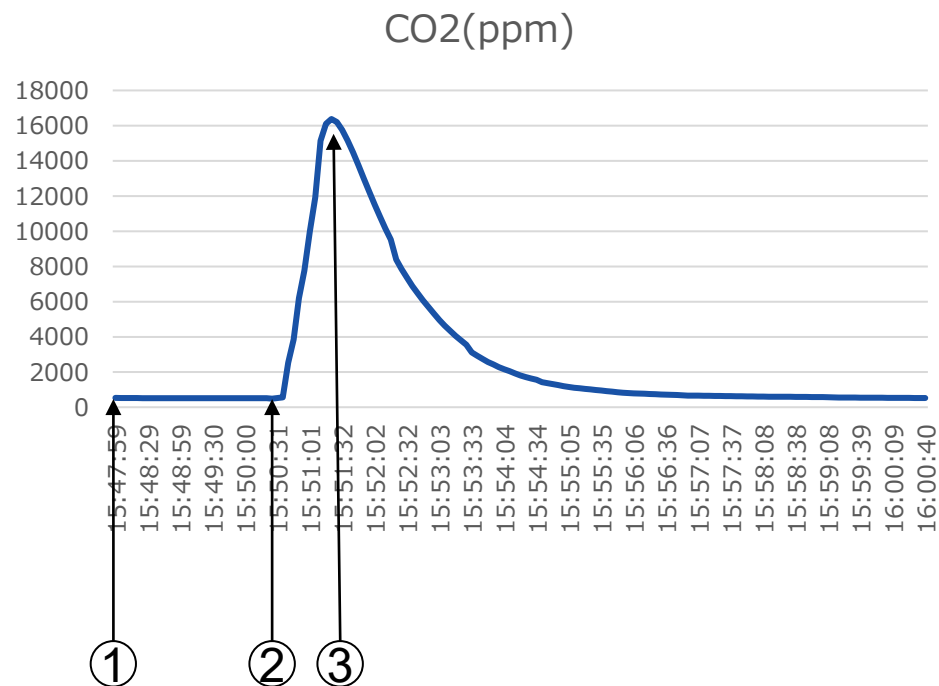
- CO2濃度 : 460ppmぐらい
- 温度 : 22℃ぐらい
- 湿度 : 68%ぐらい

[Sourceコード]

<https://github.com/Sensirion/arduino-i2c-scd4x>

CO2濃度変化の確認

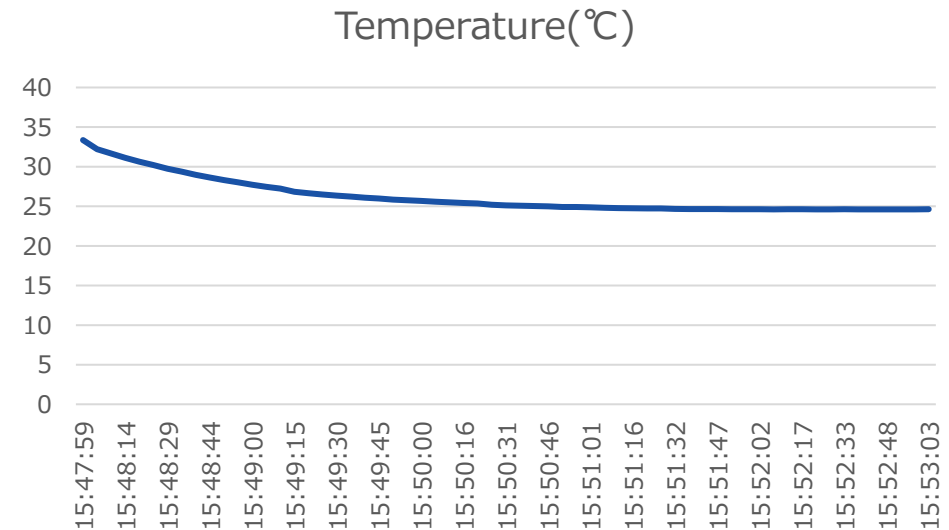
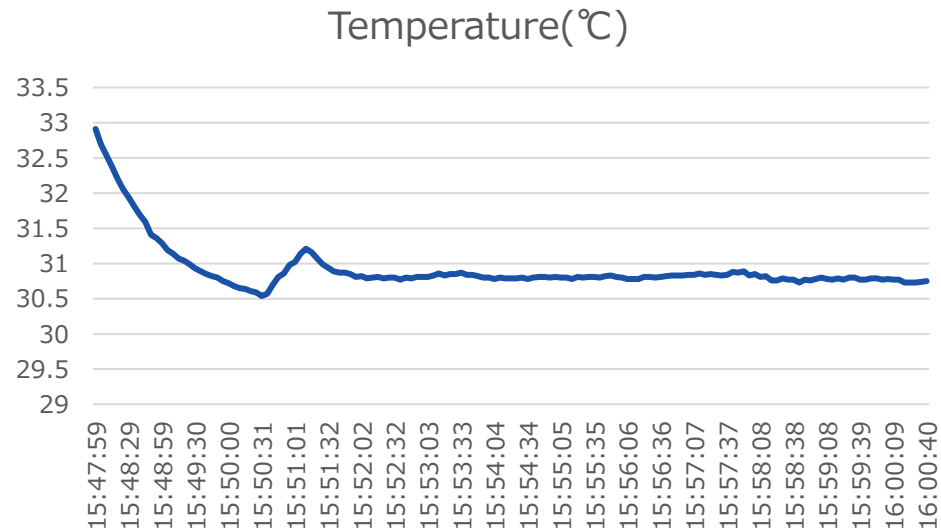
- CO2濃度変化の確認
 - CO2(SCD41)データ保存のサンプルプログラムを起動て、下記①②③の操作をする。
 - 取得したCO2濃度(ppm)は右記のグラフとなります。特に問題ありません。



- ① CO2センサーを部屋に放置する
- ② CO2センサーをビニール袋に入れて、口から空気を吹いて、ビニール袋を密閉する
- ③ CO2センサーをビニール袋から出して、部屋に放置する

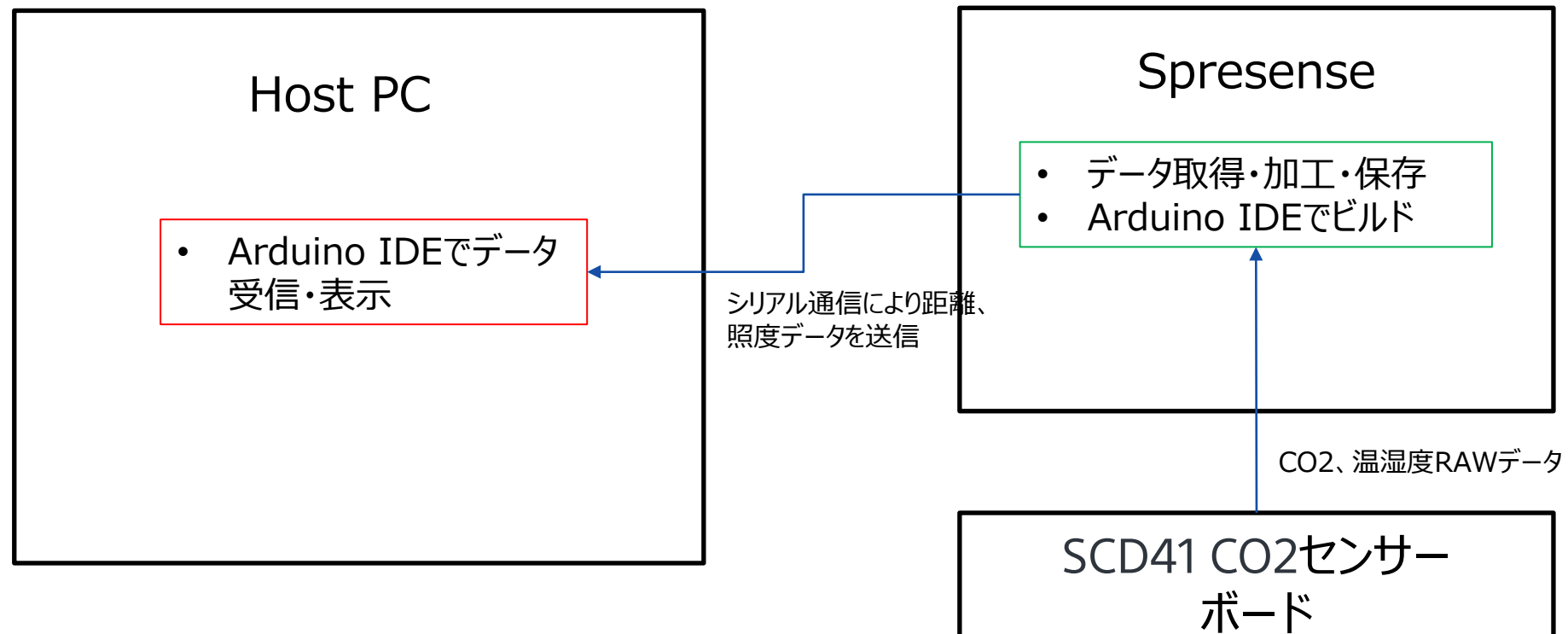
温度キャリブレーション方法について

- CO2センサーを使用している実際の周辺機器、環境などにより温度調整する必要があります(注意点3)。
 - こちらで動作確認したときに、実際温度が24℃ぐらいなので、キャリブレーションデータを修正して確認しました。
 - ※温度キャリブレーションはCO2濃度に影響しません。
- Default設定(−4℃)の場合
 - 温度は最初32℃ぐらいで、時間がたつと大体30℃ぐらいで落ち着いてきます。
- 温度キャリブレーション設定(−10℃)の場合
 - 温度は最初32℃ぐらいで、時間がたつと大体24℃ぐらいで落ち着いてきます。



Spresense + CO2センサーボード + Host PC システム構成図(SSUP提供したサンプル①)

1. 概略図



CO2(SCD41)データ保存のサンプルプログラムを動作確認する(SSUP提供したサンプル①)

1. 概要

1. CO2(SCD41)センサーからCO2濃度、温湿度のデータを取得して、FlashメモリやSDカードにcsvの形式で保存するサンプルプログラムとなります。データ取得間隔は秒単位で設定できます。

※データが準備できるまではおよそ5秒かかるため、取得間隔は5秒以上に設定してください(注意点2)。

2. 環境

1. PC
 - Ubuntu 18.04
 - Arduino IDE:v1.8.13
2. Spresense Arduino:v2.6.0
3. Spresense Main Board
4. Spresense Extension Board
5. Spresense 用 CO2センサー Addon ボード
6. microSD カード

3. 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

4. ビルド方法

1. [Arduinoソースコードビルド方法](#)を参照して、[CO2\(SCD41\)データ保存のサンプル](#)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。

[Sourceコード]

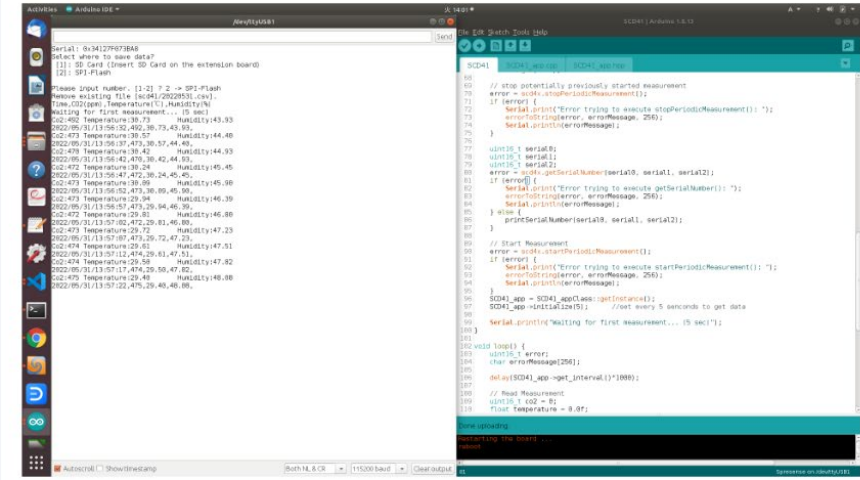
<https://github.com/SonySemiconductorSolutions/ssup-spresense/tree/main/Arduino/SCD41/SCD41>

動作例

シリアルモニタを開く



保存場所を選択する

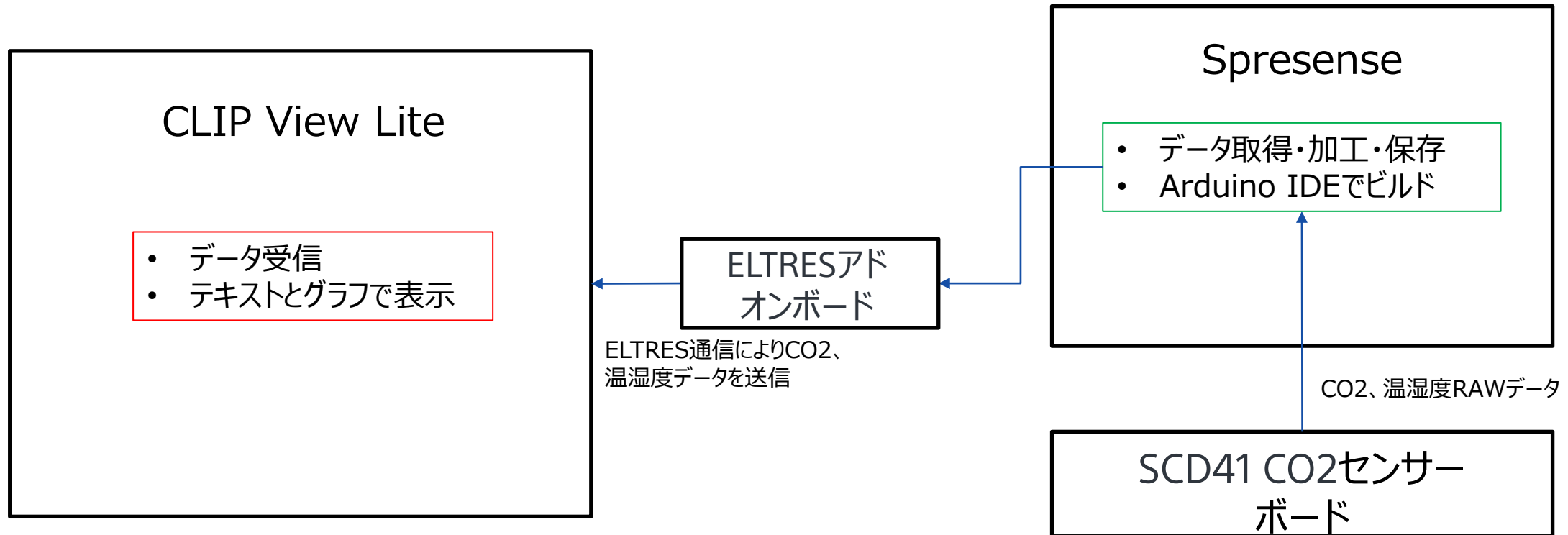


csvファイルを確認する

Time	CO2(ppm)	Temperature(°C)	Humidity(%)
2022/05/31/13:56:33	514	30.53	44.66
2022/05/31/13:56:39	515	30.37	45.14
2022/05/31/13:56:45	506	30.06	46.06
2022/05/31/13:56:51	504	29.92	46.49
2022/05/31/13:56:57	504	29.79	46.88
2022/05/31/13:57:03	505	29.57	47.51
2022/05/31/13:57:09	504	29.48	47.83
2022/05/31/13:57:15	503	29.39	48.1
2022/05/31/13:57:21	503	29.3	48.34
2022/05/31/13:57:27	502	29.14	48.8

Spresense + ELTRESアドオンボード + CO2センサーボード + CLIP View Liteシステム構成図(SSUP提供したサンプル②)

1. 概略図



CO2(SCD41)データをELTRES通信によりCLIPサーバーに確認する(SSUP提供したサンプル②)

1. 概要

1. CO2(SCD41)センサーからCO2濃度、温湿度のデータを取得して、トータル5回で、1分ごとにCLIP Viewer Lite サーバーに送信する必要があります。
※本サンプルはMulticoreで対応しています。

2. 環境

1. PC
 - Ubuntu 18.04
 - Arduino IDE:v1.8.13
2. Spresense Arduino:v2.6.0
3. Spresense Main Board
4. Spresense 用 CO2センサー Addon ボード
5. ELTRESアドオンボード

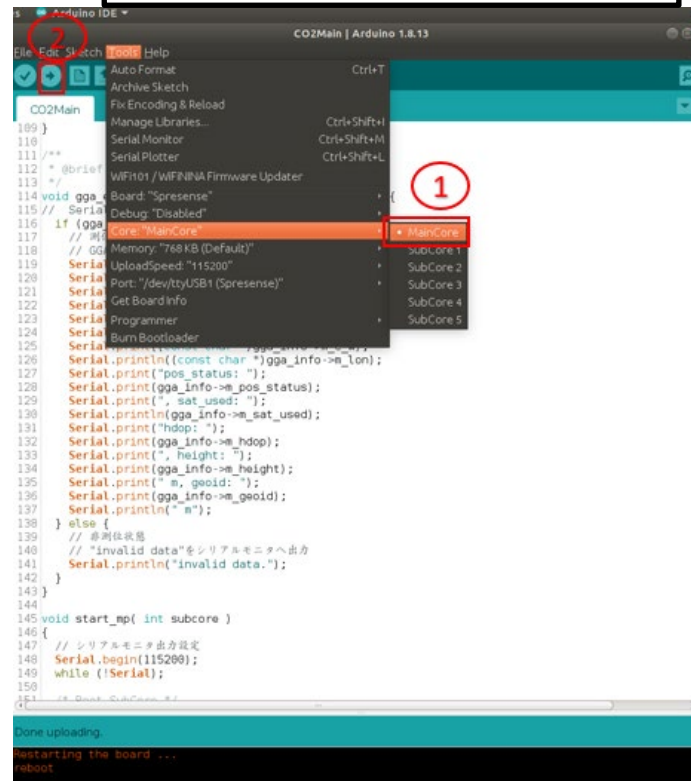
3. セットアップ

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

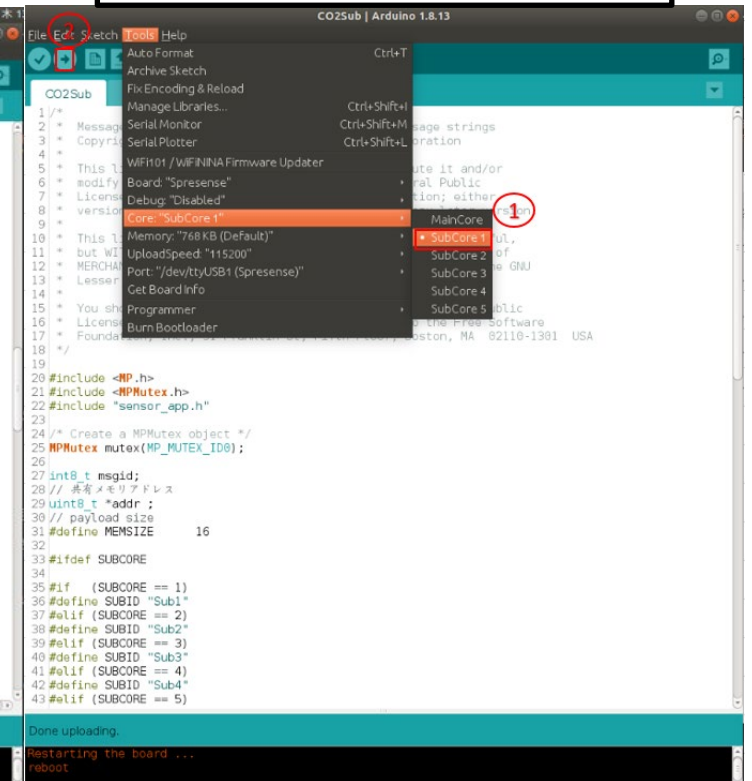
4. ビルド方法

1. [Arduinoソースコードビルド方法](#)を参照して、[CO2MainCoreソースコード](#)と[CO2SubCoreソースコード](#)をそれぞれ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。スケッチの書き込みが完了するまで待ちます。
2. 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。

MainCoreに書き込む方法



SubCoreに書き込む方法




[Sourceコード]

<https://github.com/SonySemiconductorSolutions/ssup-spresense/tree/main/Arduino/CEBB-CXM1501GR-02/CO2>

CO2(SCD41)データをELTRES通信によりCLIPサーバーに確認する(SSUP提供したサンプル②)

CO2濃度、温湿度のデータをCLIP Viewer Lite サーバーに表示するテキストデータです。

 **CLIP Viewer Lite**

ダッシュボード

グラフ

三

日時指定

2022-06-07 08:17

～

検索

CSVダウンロード

自動更新

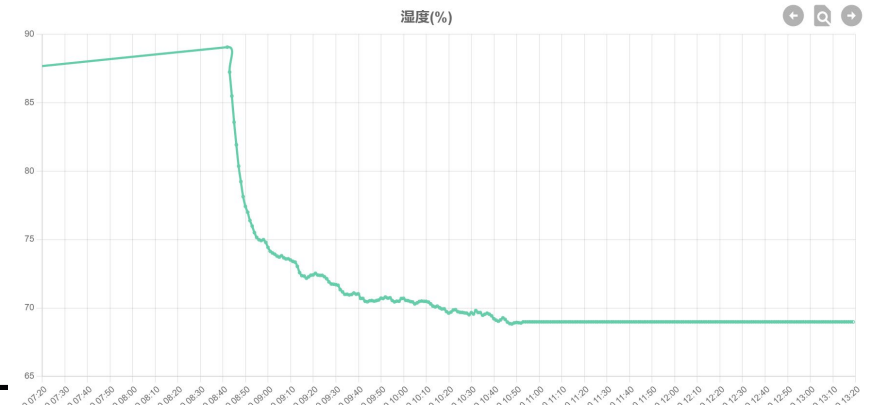
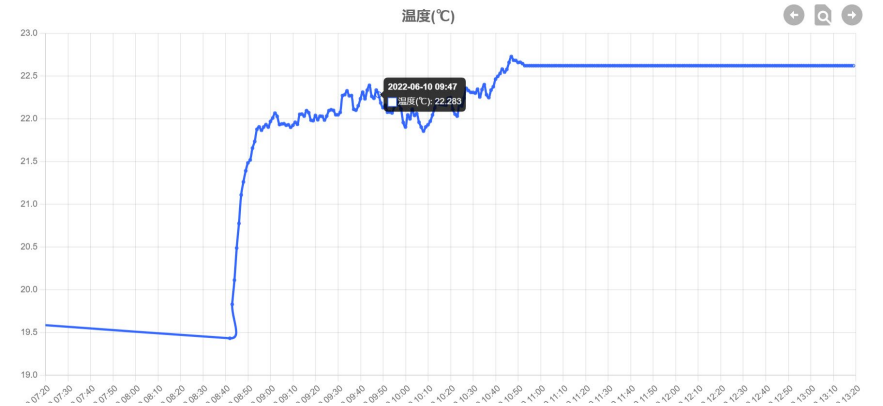
項目

フィルター

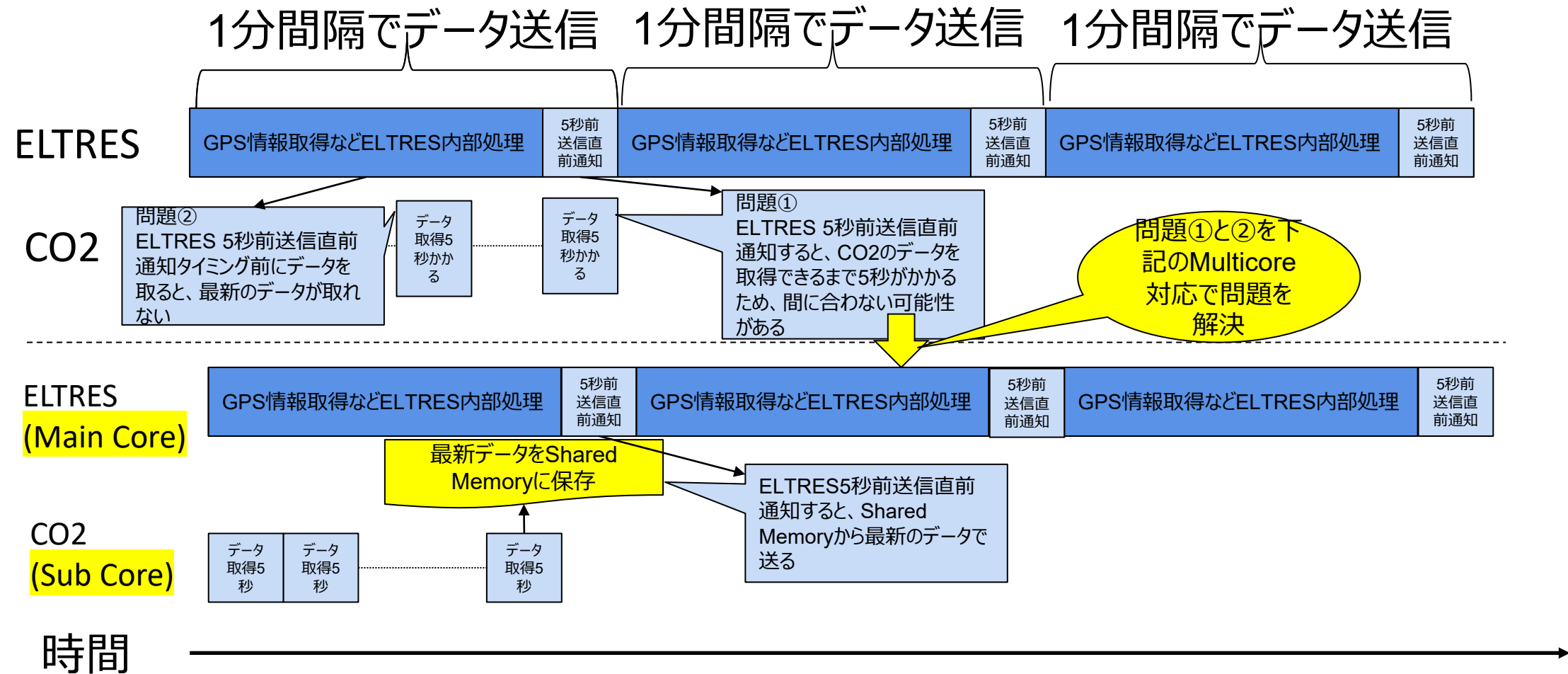
送信時間	ペイロードタイプ	ペイロード	RSSI	温度(℃)	湿度(%)	CO2濃度(ppm)
2022-06-10 13:26:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:25:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:24:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	2	22.62237548828125	68.98651123046875	480
2022-06-10 13:23:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:22:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:21:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:20:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:19:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:18:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:17:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:16:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:15:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:14:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:13:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:12:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:11:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:10:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	0	22.62237548828125	68.98651123046875	480
2022-06-10 13:09:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	1	22.62237548828125	68.98651123046875	480
2022-06-10 13:08:57	温湿度CO2情報	8241b4faa04289f91843f00000000000	2	22.62237548828125	68.98651123046875	480

CO2(SCD41)データをELTRES通信によりCLIPサーバーに確認する(SSUP提供したサンプル②)

CO2濃度、温湿度のデータをCLIP Viewer Lite サーバーに表示するグラフです。



ELTRESとCO2との組み合わせでMulticore対応について



注意点纏め：

1. CO2センサーはSpresense本体のカメラ接続用のコネクタを隠す形になりますので、カメラも必要な場合はご注意ください。
2. CO2(SCD41)データ保存のサンプルを動作確認(SSUP提供したサンプル)する際に、データが準備できるまではおおよそ5秒かかるため、取得間隔は5秒以上に設定してください。
3. CO2センサーを使用している実際の周辺機器、環境などにより温度調整する必要があります。