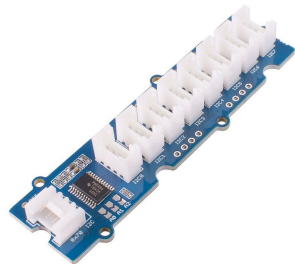


IoTデバイス活用マニュアル



センサ種類: 8チャンネルI2CHub(マルチプレクサ)

型番: [Grove - 8 Channel I2C Multiplexer/I2C Hub \(TCA9548A\)](#)

ストーリー: WioNodeに8チャンネルI2CHubを介して、複数の同種のI2Cセンサを接続する



改訂記録:

2023・04・14 初版 作成 陣内





0. 概要

なぜI2CHub(マルチプレクサ)が必要か



なぜI2CHub(マルチプレクサ)が必要か

- 通常のGrove I2CHub(下図)は、各線を単に並列に接続しているだけです。I2Cセンサそのものの機能で、異なるチャンネル番号をもつI2Cセンサを複数接続できるものです。

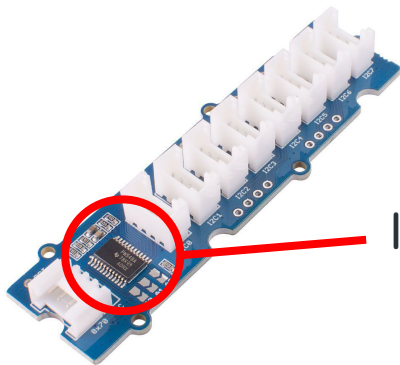
→ チャンネル番号を変更できない同種のI2Cセンサを複数接続したい場合、この種のI2CHubは使えません。



なぜI2CHub(マルチプレクサ)が必要か

- ・ TCA9548A I2Cマルチプレクサチップ搭載、最大8つの同じアドレスI2Cデバイスを同じGrove I2Cシステムに接続できます。時分割多重化技術 (time-division multiplexing technology) を採用しているため、同じコントローラーが同じアドレスを持つ8つのI2Cデバイスを制御できます。

出典：[Seeedstudio](https://www.seeedstudio.com/)



I2Cマルチプレクサチップ



1. Arduinoのプログラムの変更

ライブラリの読み込みとコードの変更箇所

プログラム

以下では、例として [GROVE - VL53L0X搭載 Time of Flight測距センサ](#) をWioNodeに 8Channel I2C Hub を介して2個接続するプログラムを扱います。

・**ライブラリ**: 以下2つのライブラリをそれぞれ、ダウンロードし、「zip形式のライブラリをインストール」してください。

- ・ [Grove 8 Channel I2C Hub Library](#) (zip)
- ・ [Grove-Ranging-sensor-VL53L0X](#) (GitHub)

・**サンプルプログラム**: [MultiChannel_VL53L0X.ino](#)

修正箇所: 以下の該当箇所を利用環境に応じて修正する。

- ・ WiFi の SSID、パスワード
- ・ Azure IoT Hub で発行されたデバイスの接続文字列

プログラムの解説 1/3

以下、要所をピックアップします。必ずしも、サンプルプログラム通りではありません。

//マルチチャンネルI2C Hubの宣言

```
#include "TCA9548A.h"
#include <Wire.h>
TCA9548A<TwoWire> TCA;
#define WIRE Wire
#define MUX_ADDRESS
TCA9548_DEFAULT_ADDRESS
```

//複数のセンサーの宣言

//(VL53L0X TOF距離センサーの場合)

//VL53L0X TOF距離センサー

#define SERIAL Serial **センサー個数分の配列にする**

```
#include "Seeed_vl53l0x.h"
```

```
const int Cnt_VL53L0X=2;
```

```
Seeed_vl53l0x VL53L0X[Cnt_VL53L0X];
```

```
int i;
```

```
int RangeMilliMeter[Cnt_VL53L0X];
```

ファイル冒頭の宣言部

プログラムの解説 2/3

//マルチチャンネル I2CHub初期化

```
#include "TCA9548A.h"
```

```
TCA.begin(WIRE);
```

```
TCA.openChannel(TCA_CHANNEL_0);
```

```
TCA.openChannel(TCA_CHANNEL_1);
```

void setup() 内

//チャンネル数分 TOFセンサー初期化

```
VL53L0X_Error Status = VL53L0X_ERROR_NONE;
```

```
for(i=0; i<Cnt_VL53L0X; i++){
```

```
    switchChannel(i);
```

```
    ...
```

チャンネルの切り換え

//データ読み込み チャンネル数分繰り返し

```
for(i=0; i<Cnt_VL53L0X; i++){
```

```
    switchChannel(i);
```

void loop() 内

プログラムの解説 3/3

// Azureへ送るデータの用意、送信

```
DataElement a = DataElement();  
a.setValue("Sensor", "TOF");  
a.setValue("EspValue", RangeMilliMeter[0]);  
a.setValue("EspValue2", RangeMilliMeter[1] );  
a.setValue("duration", (int)durationtime);  
Azure.push(&a);
```

//チャンネル切り換え

```
void switchChannel(int channel) {  
    WIRE.beginTransaction(MUX_ADDRESS);  
    WIRE.write(1 << channel);  
    WIRE.endTransmission();  
}
```

void loop() 内

この例①では、EspValue2を追加して、送信しているが、他に以下のような手法も考えられる、

②チャンネル番号の項目を追加して、チャンネル数分、このブロックを実行する方法

```
a.setValue("channel", i);
```

③sensorの項にチャンネル番号も含め、チャンネル数分、このブロックを実行する方法。

```
a.setValue("Sensor", "TOF1");
```

プログラムファイル下の方



2. Stream Analyticsのクエリ変更



StreamAnalyticsクエリの変更

SELECT

Dev as device,

DATEADD(hour, 9, EventEnqueuedUtcTime) as time,

EventEnqueuedUtcTime as utctime,

params.sensor as sensor,

params.espvalue as value,

params.espvalue2 as value2,

params.duration as duration,

INTO

outputpowerbi

FROM

inputiothub

この例①では、必要数 espvalueを追加する。
チャンネル番号を追加する方法②なら、
params.channel as channel,



3. PowerBIでのレポートの作成



レポートの変更

Power BI 00FSTA

ホーム 作成 参照 Data ハブ メトリック アプリ 詳細 ワークスペース 00FSTA

ファイル ビュー 閲覧表示 モバイルレイアウト

value の合計 value2 の合計 value3 の合計

value の合計 value2 の合計 value3 の合計

time

Apr 14, 6PM Apr 15, 12AM Apr 15, 6AM Apr 15, 12PM

注: Value、value2、value3 と凡例に出るので、凡例に deviceは使えなくなる。

②value2、value3と必要数、y軸に追加する。

①凡例の欄は空欄にする。

このビジュアルでのフィルター...

device (すべて) です

フィルターの種類 ①

基本フィルター

すべて選択

92jinnouchi01 59048

92jinnouchi06T... 18139

94hamanaka01-... 59926

668

536

1 つを選択する

time (すべて) です

value の合計 (すべて) です

視覚化 ビジュアルのビルド

検索

00FSTAtable

device

Σ duration

sensor

status

time

utctime

Σ value

Σ value2

Σ value3

X 軸

time

Y 軸

value の合計

value2 の合計

value3 の合計

第 2 Y 軸

ここにデータ フィールドを...

凡例

ここにデータ フィールドを...

ページ 1