# RL78/G23

## 簡単センサデータクラウド可視化-Ambient-サンプルスケッチ (Arduino™ スケッチ)

## 要旨

本アプリケーションノートでは、RL78/G23-128p Fast Prototyping Board(RL78/G23 FPB)と超低消費電力 Wi-Fi + Bluetooth Low Energy コンボ Pmod ボード(DA16x00)を使用して、CO2・温度・湿度センサ SCD40 から取得した CO2、温度、湿度データを Wi-Fi ネットワーク経由で IoT クラウドサービス(Ambient)に送信し、グラフ表示する方法を説明します。説明は 3 ステップに分け、ステップ 1 で SCD40 センサから取得した CO2、温度、湿度データをシリアルモニタに表示して確認し、ステップ 2 でダミーデータを Wi-Fi ネットワーク経由で IoT クラウドに送信してグラフ表示、ステップ 3 でステップ 1 と2を組み合わせて、CO2、温度、湿度データを IoT クラウドサービスに送信してグラフ表示します。

## 動作確認デバイス

RL78/G23-128p Fast Prototyping Board RTK7RLG230CSN000BJ
超低消費電力 Wi-Fi + Bluetooth Low Energy コンボ Pmod ボード US159-DA16600EVZ
「SCD40 を使用した CO2 センサモジュール」 AE-SCD40-BO

## 商標・他社 TM

Arduino は Arduino SA の商標です。

目次

1. システム概要 ..... 3

2. 動作確認環境 ..... 4

3. 開発環境構築 ..... 4

3.1 ボードの接続 ..... 4
3.2 使用端子一覧 ..... 6
3.3 ARDUINO™ IDE のセットアップ ..... 6
3.3.1 Arduino™ IDE を起動 ..... 6
3.3.2 ボードマネージャを起動し、RL78/G23-128p のボード情報をインストール ..... 6
3.3.3 ボードを PC に接続し、Arduino™ IDE でボードを選択 ..... 8

4. ソフトウェアの説明 ..... 10

4.1 ステップ1 SCD40 センサの動作確認 ..... 10
4.1.1 SCD40 Arduino ライブラリのインストール ..... 10
4.1.2 ステップ1の動作確認 ..... 12
4.2 ステップ2 IoT クラウドサービス「AMBIENT」へのデータ送信 ..... 15
4.2.1 Ambient ライブラリのインストール ..... 15
4.2.2 ステップ2のサンプルコード概要 ..... 17
4.2.3 Ambient ライブラリの API 関数 ..... 18
4.2.4 Ambient のユーザー登録とチャネル生成 ..... 20
4.2.5 ステップ2のプログラム概要 ..... 21
4.2.6 ステップ 2 の動作確認 ..... 22
4.3 ステップ3 センサデータの AMBIENT への送信 ..... 26
4.3.1 ステップ3の動作確認 ..... 26

5. 注意事項 ..... 28

5.1 ビルドが始まらない ..... 28
5.2 書き込みがエラーになる ..... 28

## 1.　システム概要

本システムは、RL78/G23-128p Fast Prototyping Board と超低消費電力 Wi-Fi + Bluetooth Low Energy コンボ Pmod ボード（DA16x00）、CO2・温度・湿度センサ SCD40 を搭載した CO2 センサモジュール、IoT クラウドサービス（Ambient）で構成されています。プログラムの作成と書き込みには Arduino™ IDE を使用します。SCD40 センサから取得した CO2、温度、湿度データを Wi-Fi ネットワークを介して IoT クラウドサービス（Ambient）に送信し、グラフ表示して確認します。

本アプリケーションノートは、説明を 3 ステップに分け、ステップ 1 で SCD40 センサから取得した CO2、温度、湿度データをシリアルモニタに表示して確認し、ステップ 2 でダミーデータを Wi-Fi ネットワーク経由で IoT クラウドに送信してグラフ表示し、ステップ 3 でステップ 1 と 2 を組み合わせて、CO2、温度、湿度データを IoT クラウドサービスに送信してグラフ表示します。最終形となるステップ 3 のスケッチ（プログラム）のブロック構成を以下に示します。

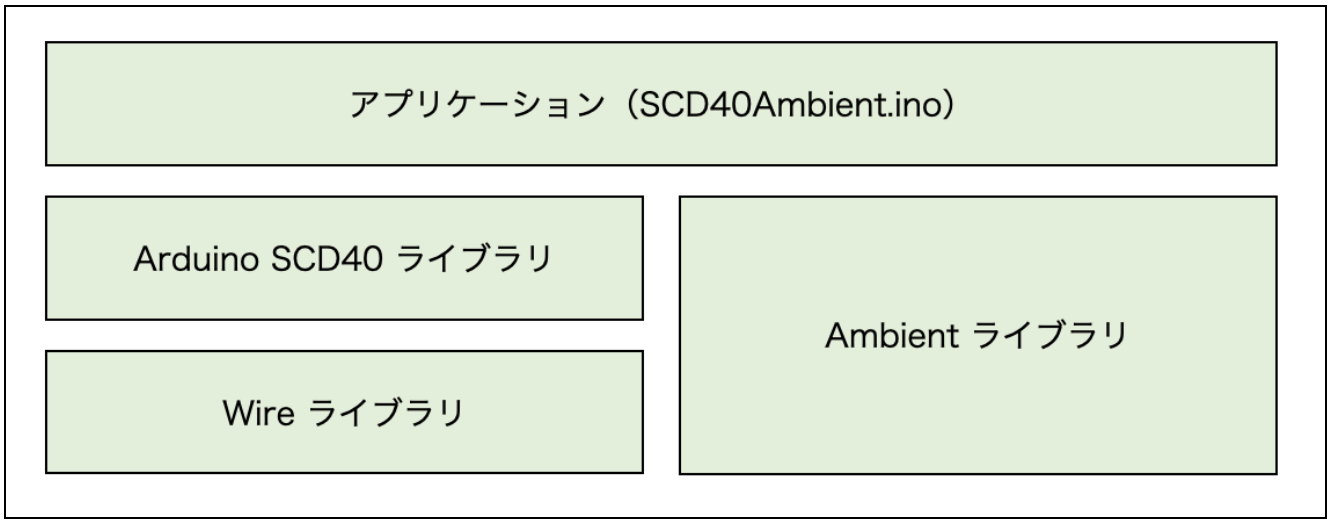


図 1-1 ステップ3のソフトウェアブロック図

本システム構成の概略図を以下に示します。

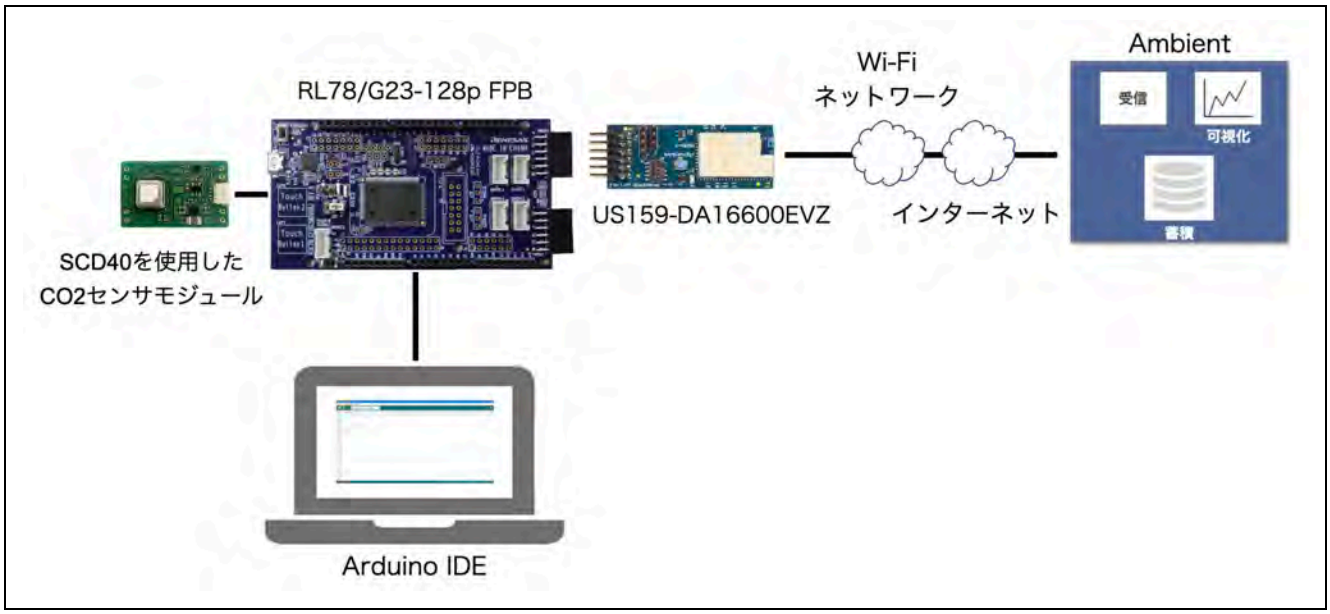


図 1－2　システム構成図

## 2.　動作確認環境

本システムの動作確認環境は、以下のとおりです。

表 2-1　動作確認環境（ハードウェア）

| 項目 | 内容 |
|---|---|
| 評価ボード | RL78/G23-128p Fast Prototyping Board RTK7RLG230CSN000BJ |
| 通信モジュール | US159-DA16600EVZ |
| センサボード | Sensirion 社 SCD40 を使用した CO2 センサモジュール AE-SCD40-BO（秋月電子通商） |
| 動作電圧 | 5V |

表 2-2　動作確認環境（ソフトウェア）

| 項目 | 内容 | バージョン |
|---|---|---|
| OS | Windows 11 | - |
| 統合開発環境（IDE） | Arduino™ IDE | 2.3.6 |
| 標準ライブラリ | RL78/G23-128p FPB ライブラリ | 1.2.0 |
| | Sensirion I2C SCD4x | 1.0.0 |

## 3.　開発環境構築

ボードの接続方法と Arduino™ IDE のセットアップを説明します。

本システムでは Arduino™ IDE 2.3.6 を使用しています。Arduino™ IDE 2.3.6 以降をインストールしていない場合は、インストールしてください。
https://www.arduino.cc/en/software

### 3.1　ボードの接続

図 3-1 のように PC と評価ボードを USB ケーブルで、評価ボードと通信モジュールを PMOD1 コネクタで接続します。また、図 3-2 のように評価ボードとセンサボードをジャンパワイヤで接続します。図が煩雑になるのを避けるために 2 つに分けて説明していますが、PC、評価ボード、通信モジュール、センサボードを合わせて接続してください。

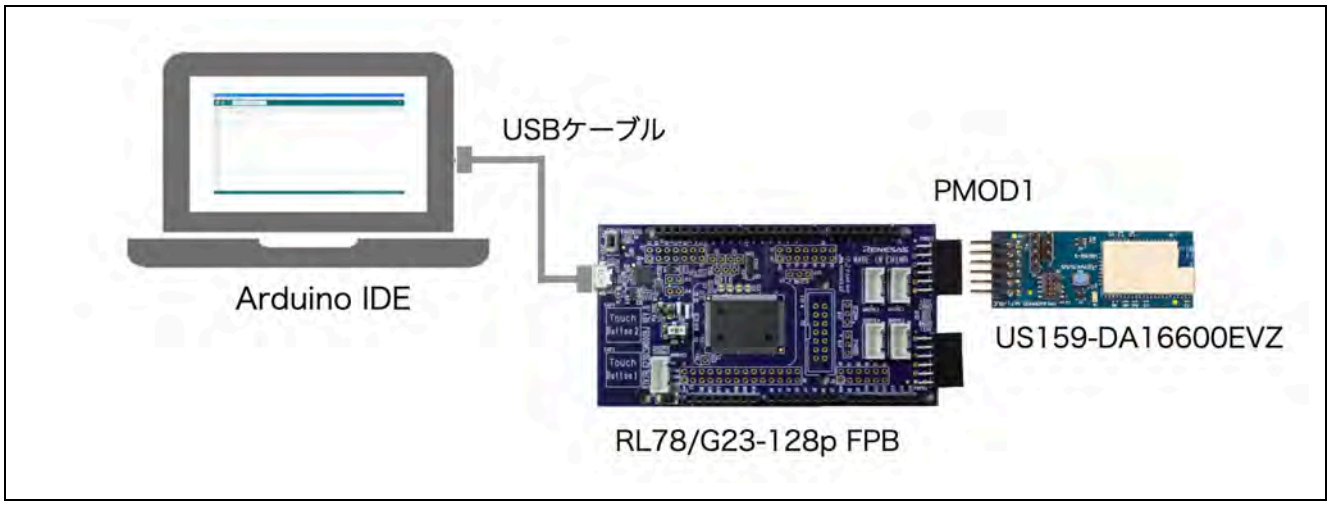


図３－１　PC と評価ボード、通信モジュールの接続

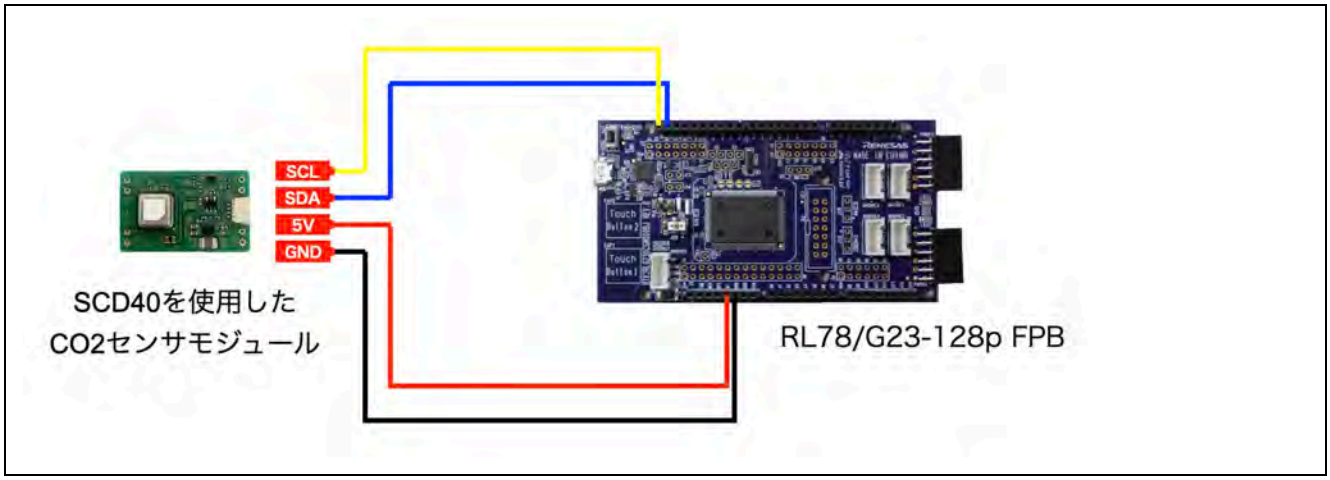


図３－２　評価ボードとセンサボードの接続

本システムでは、評価ボードへの電源供給は USB を使用します。評価ボードの回路を確認し、必要に応じてジャンパを設定してください。

本システムでは、評価ボードのジャンパを以下のように設定します。

表 3-1　評価ボードのジャンパ設定

| ジャンパ | 設定 | 機能 |
|---|---|---|
| J20 | 1-2 ショート | マイコンへの 5V 電源供給<br>出荷時初期設定 |

## 3.2　使用端子一覧

本システムはセンサボードとジャンパワイヤで接続します。使用端子を以下に示します。

表 3-2 本システムの使用端子一覧（センサボード）

| 項目 | Arduino™ 信号名 | 端子 |
|---|---|---|
| I2C | SDA | SDA |
| | SCL | SCL |
| VDD | 5V | 5V |
| GND | GND | GND |

各ボードの詳細なピンの説明は、以下のマニュアルを参照してください。

- RL78/G23-128p Fast Prototyping Board ユーザーズマニュアル（R20UT4870JJ0110）
- 「SCD40 を使用した CO2 センサモジュール」

## 3.3　Arduino™ IDE のセットアップ

本章で Arduino™ IDE のセットアップ手順を説明します。

備考. セットアップ手順は、クイックスタートガイド · renesas/Arduino Wiki · GitHub に記載されている手順と同様です。また上記サイトでは、LED を点滅させるサンプルスケッチが記載されています。必要に応じて、参照してください。

### 3.3.1　Arduino™ IDE を起動

Arduino™ IDE を起動します。

### 3.3.2　ボードマネージャを起動し、RL78/G23-128p のボード情報をインストール

左メニューの「ボードマネージャ」アイコンをクリックして、ボードマネージャを起動します。[タイプ] は “全て” を選択し、検索欄に “RL78” と入力し、表示された [RL78/G23-128p Fast Prototyping Board] の [インストール] をクリックします。本システムでは、バージョン 1.2.0 を使用しました。

インストールが終わったら、再度「ボードマネージャ」アイコンをクリックして、ボードマネージャ欄を閉じます。

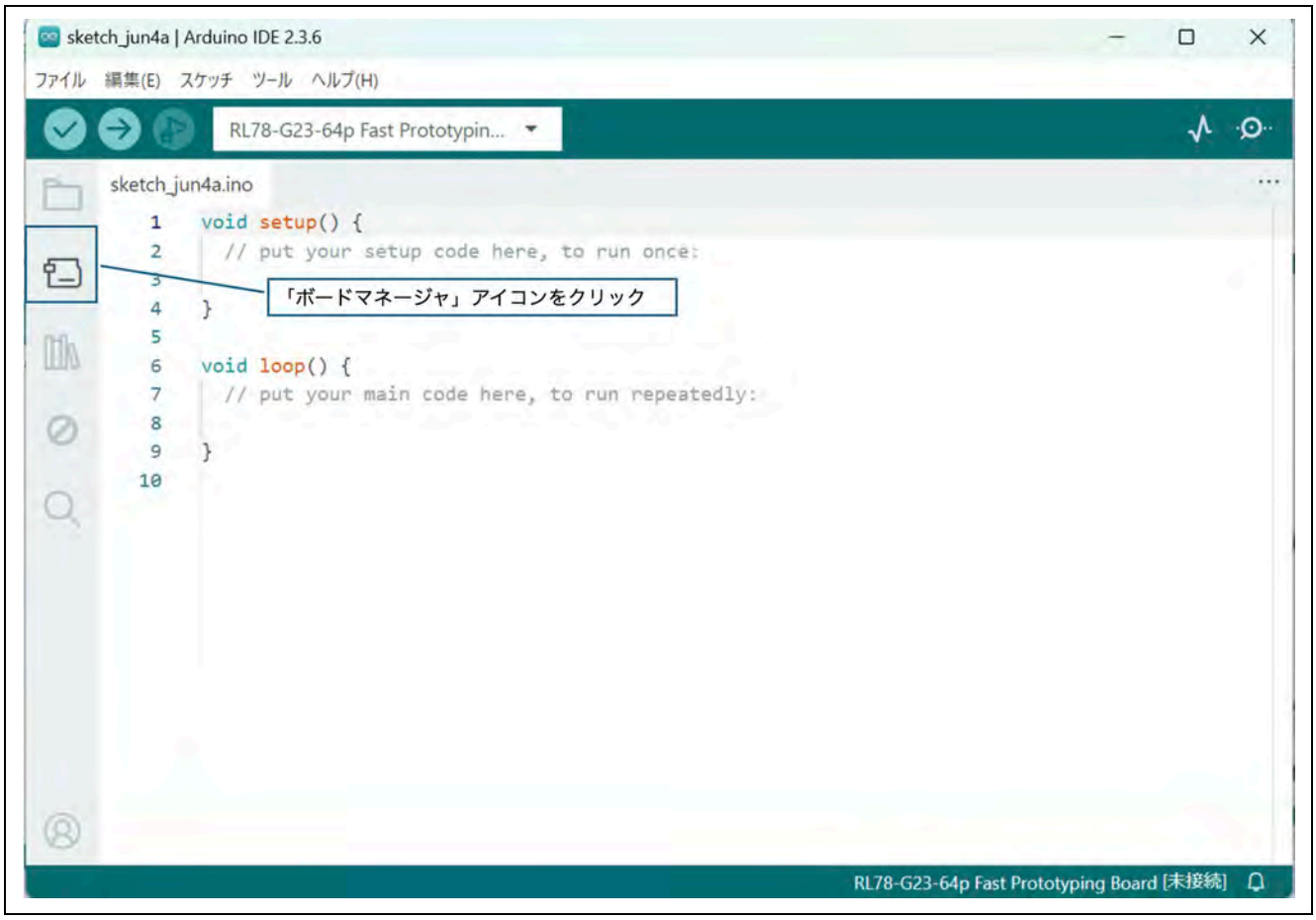


図 3-3 ボードマネージャの起動

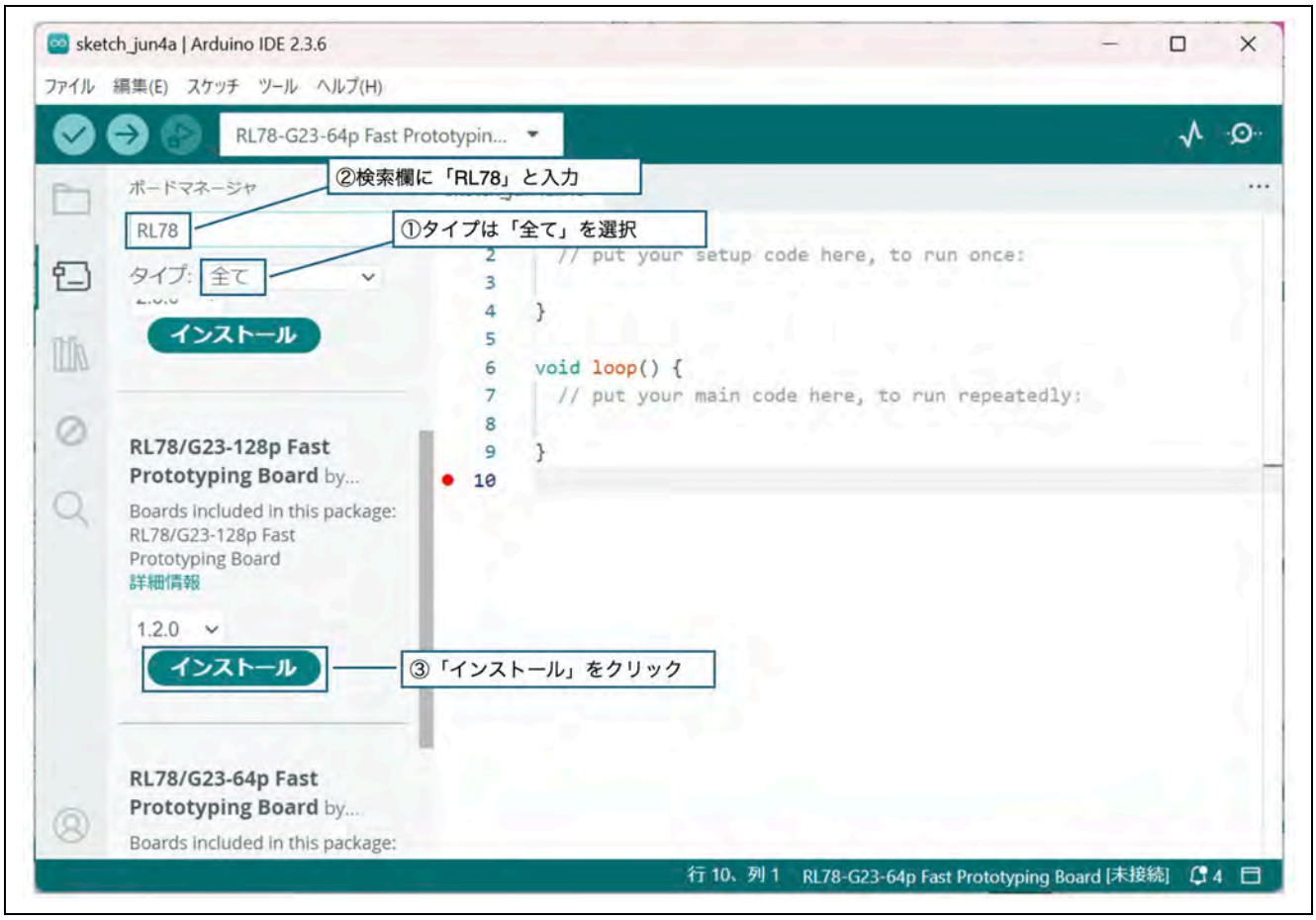


図 3-4 ボード情報のインストール

### 3.3.3　ボードを PC に接続し、Arduino™ IDE でボードを選択

PC の USB ポートに USB ケーブルで評価ボードを接続します。Arduino™ IDE のツールバーの「ボードを選択」欄をクリックし、評価ボードに割り当てられたシリアルポートを選択します。COM ポート番号は、Windows のデバイス マネージャから確認できます。

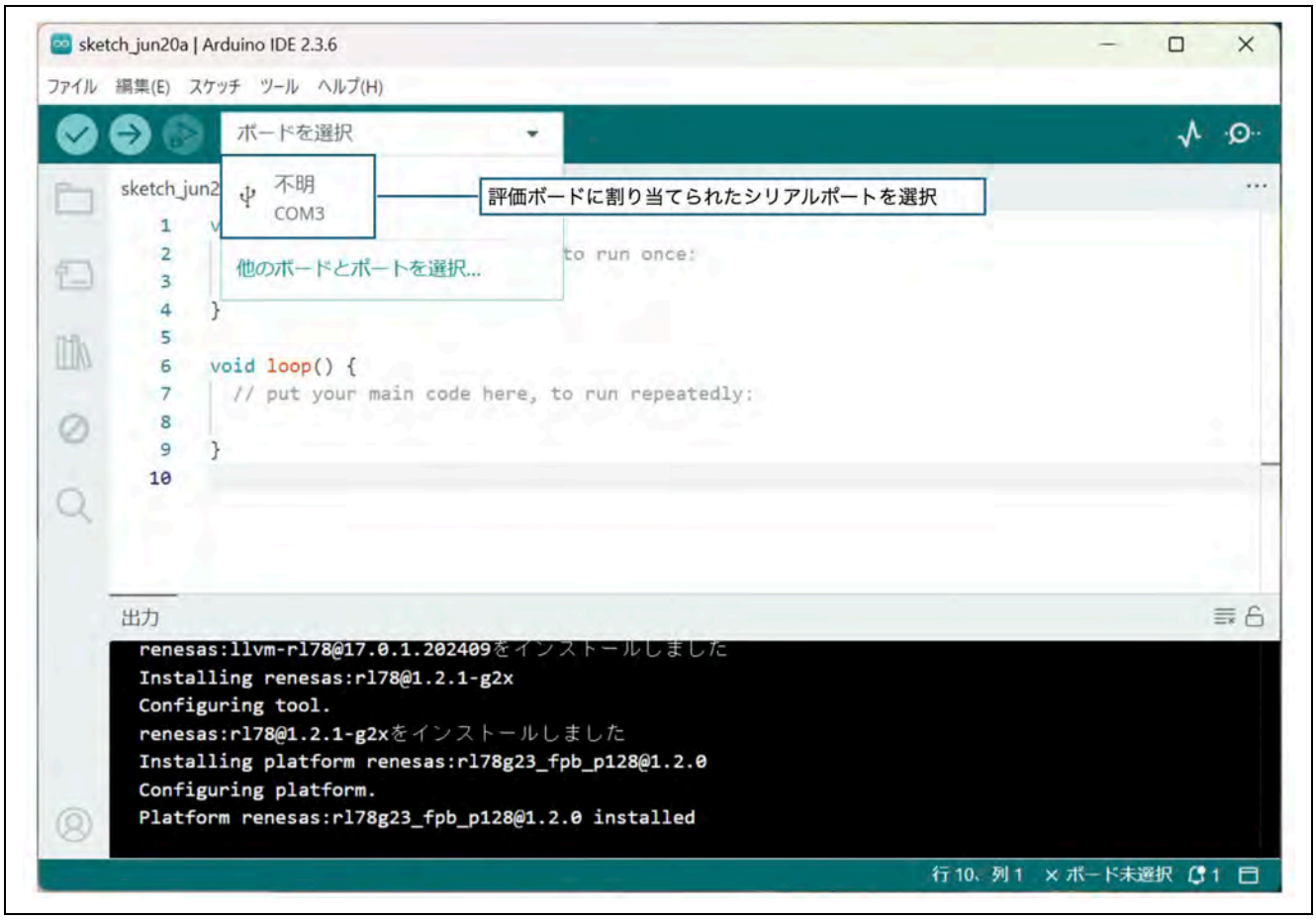


図 3-5 シリアルポートの選択

「他のボードとポートを選択」画面のボードの検索欄に「RL78」と入力し、「RL78-G23-128p Fast Prototyping Board」を選択し、「OK」をクリックします。

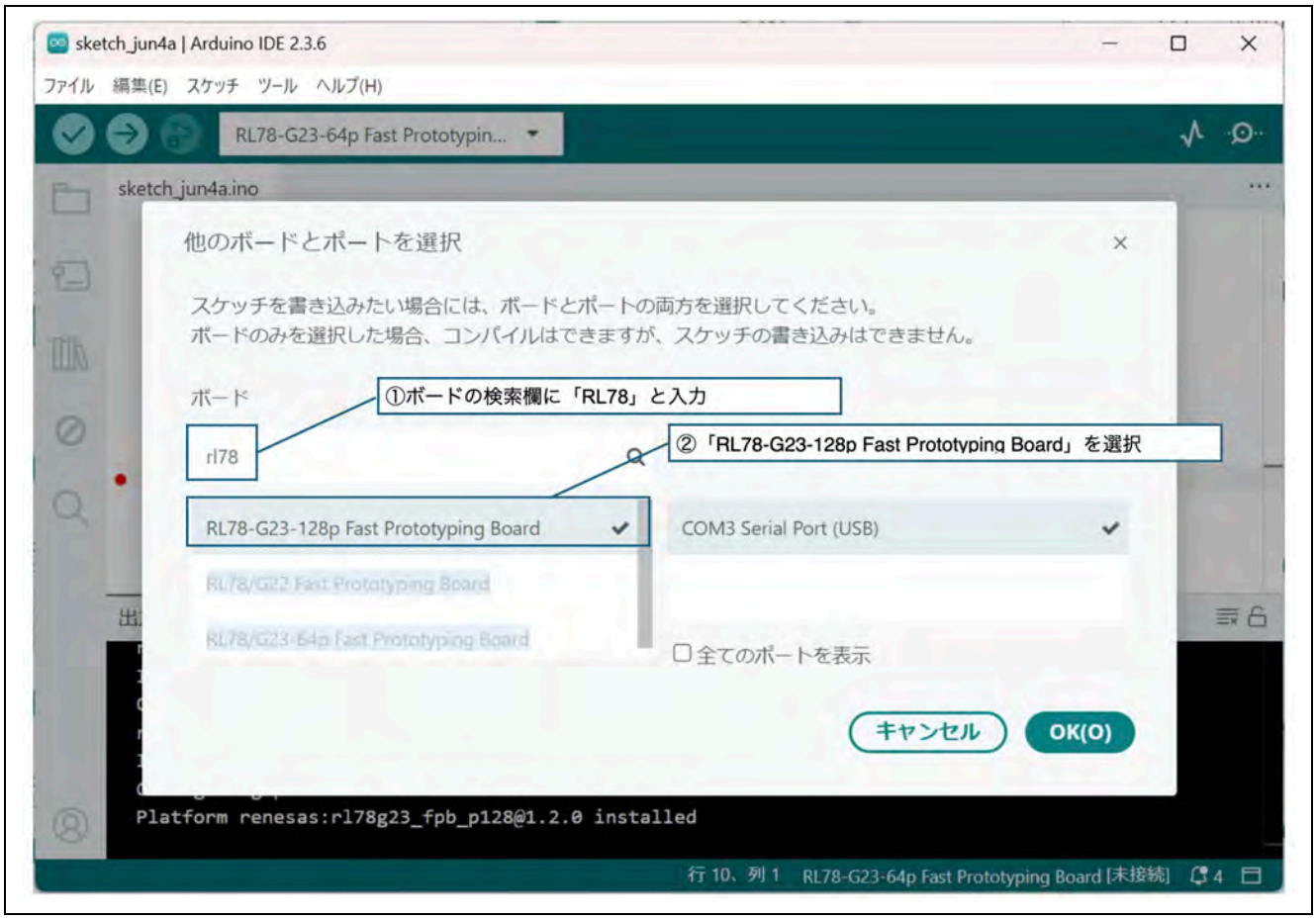


図 3-6 ボードの選択

## 4. ソフトウェアの説明

本アプリケーションノートでは、次の 3 ステップで動作確認をします。

ステップ 1：SCD40 センサから CO2、温度、湿度データを取得し、シリアルモニタに表示して確認
ステップ 2：ダミーデータを Wi-Fi ネットワーク経由で IoT クラウドサービス Ambient に送信してグラフ表示
ステップ 3：ステップ 1 と2を組み合わせて、CO2、温度、湿度データを Ambient に送信してグラフ表示

### 4.1 ステップ 1　SCD40 センサの動作確認

SCD40 センサは Sensirion 社が開発したセンサです。Sensirion 社は SCD40 にアクセスするための Arduino ライブラリとスケッチ例を提供しています。ステップ 1 の SCD40 センサの動作確認はこの Sensirion 社のスケッチ例を使っておこないます。

#### 4.1.1 SCD40 Arduino ライブラリのインストール

Arduino™ IDE で SCD40 Arduino ライブラリをインストールします。Arduino™ IDE の左メニューの[ライブラリマネージャ]アイコンをクリックしてライブラリマネージャを起動し、検索欄に「sensirion i2c scd4x」と入力します。表示された「Sensirion I2C SCD4x」のインストールボタンをクリックし、ライブラリをインストールします。本システムではバージョン 1.0.0 を使用しました。

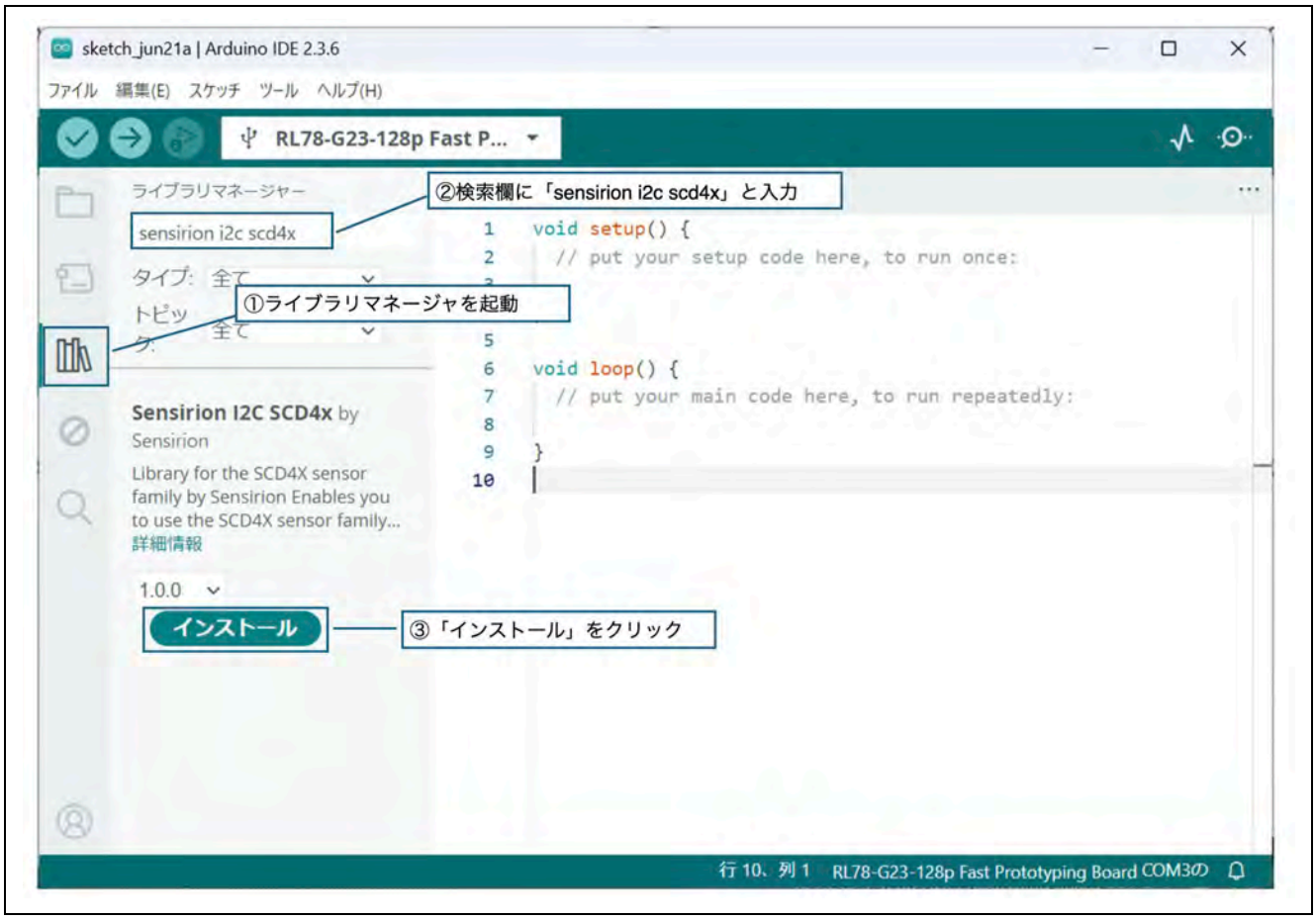


図 4-1 SCD40 ライブラリのインストール

図 4-2 のように「不足している依存関係をインストールしますか？」と聞かれたら、「全てをインストール」をクリックして、必要なライブラリをインストールしてください。

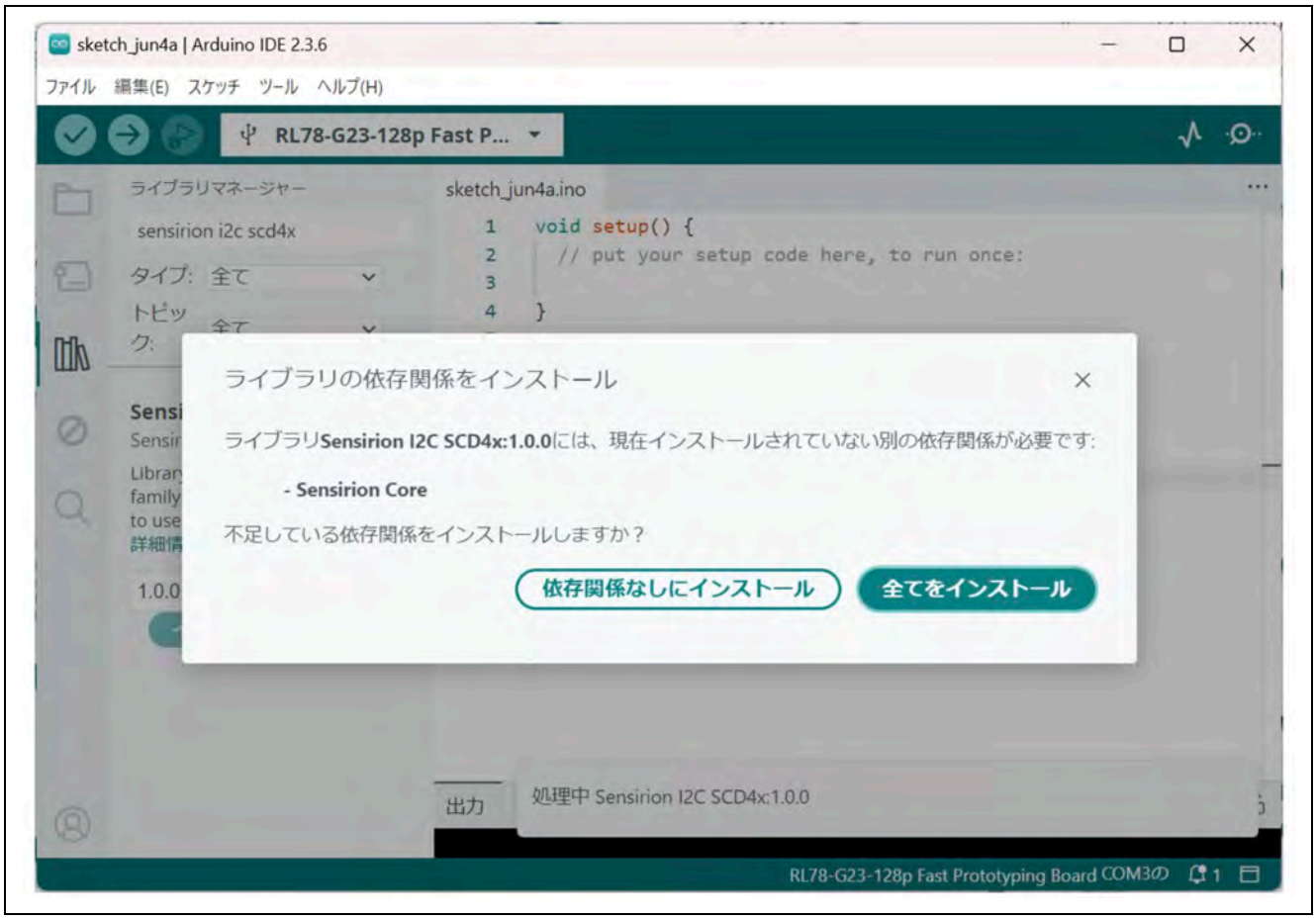


図 4-2 不足している依存関係のインストール

### 4.1.2　ステップ１の動作確認

SCD40 ライブラリをインストールするとスケッチ例も自動的にインストールされます。Arduino™ IDE の[ファイル]メニュー→[スケッチ例]→[Sensirion I2C SCD4x]とたどり、exampleUsage を選択してください。スケッチ例「exampleUsage.ino」が開きます。

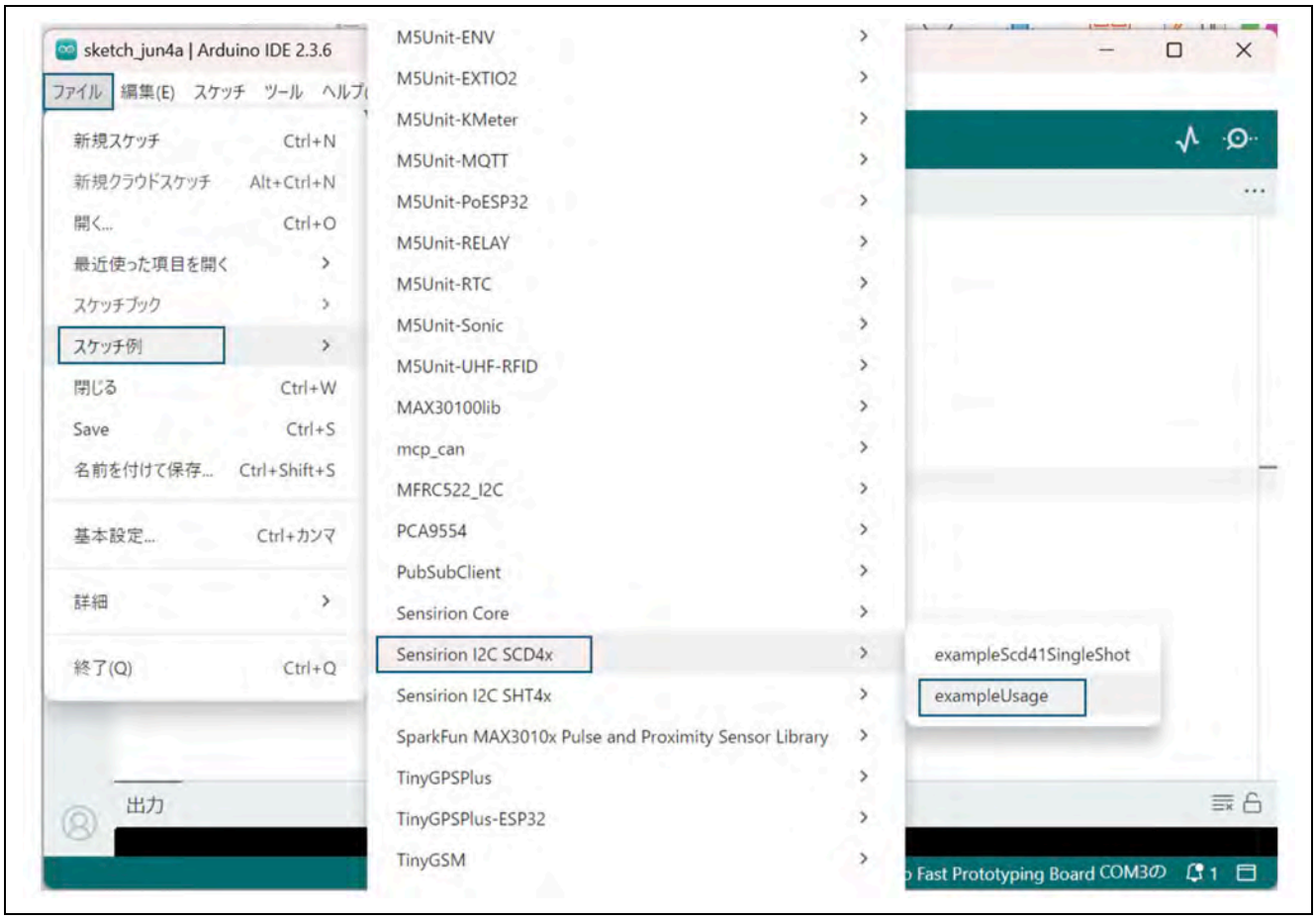


図 4-3 SCD40 のスケッチ例を選択

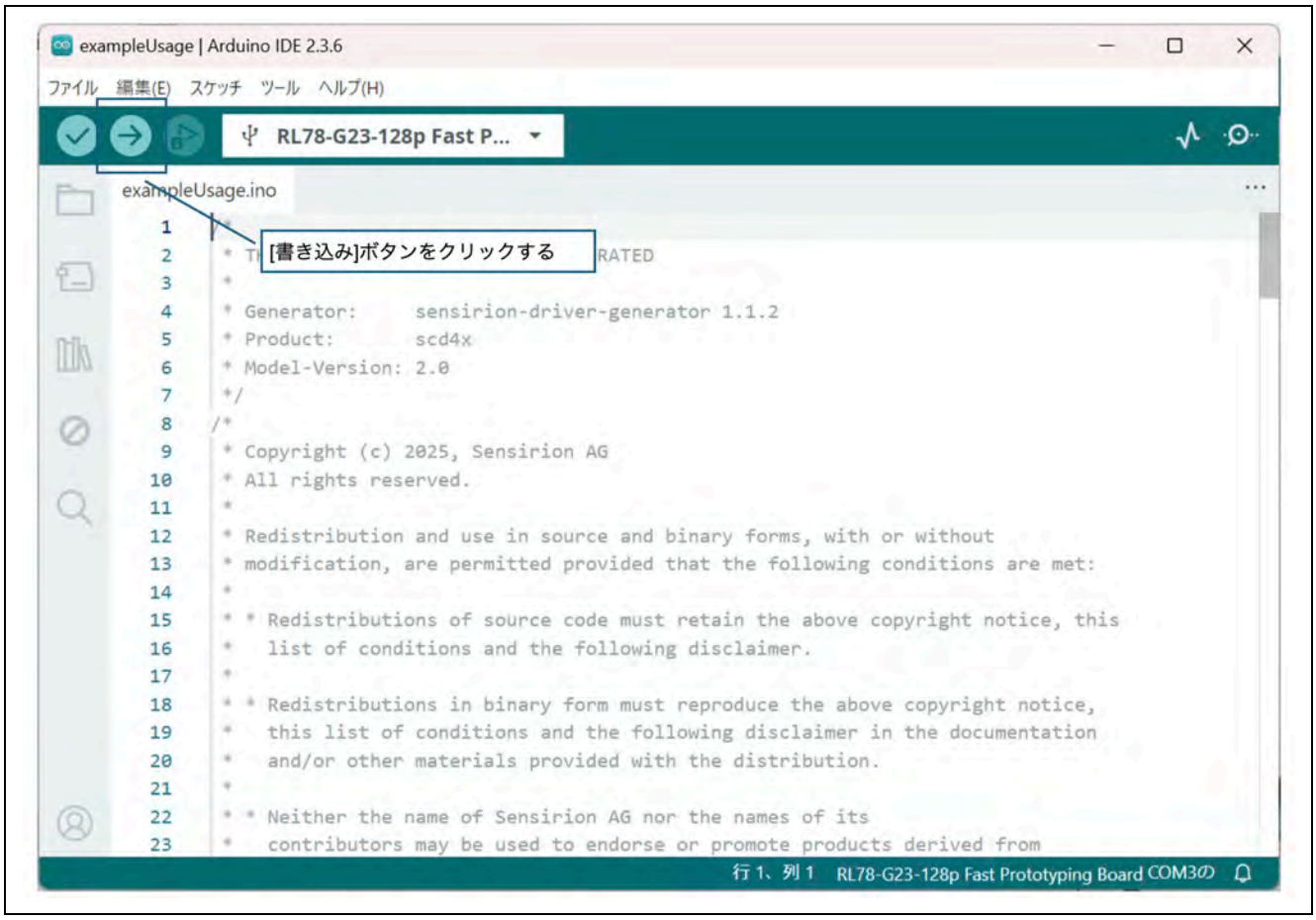


図 4-4 SCD40 のスケッチ例

「exampleUsage.ino」が開いたら、Arduino™ IDE の上部のツールバーの[書き込み]ボタンをクリックして、スケッチ例をビルドし、ボードに書き込みます。

書き込みが終了したら、Arduino™ IDE 右上の[シリアルモニタ]ボタンをクリックすると、画面下部にシリアルモニタが表示されます。最初にセンサのシリアル番号が表示され、次に 5 秒ごとに CO2 濃度、温度、湿度が表示されます。

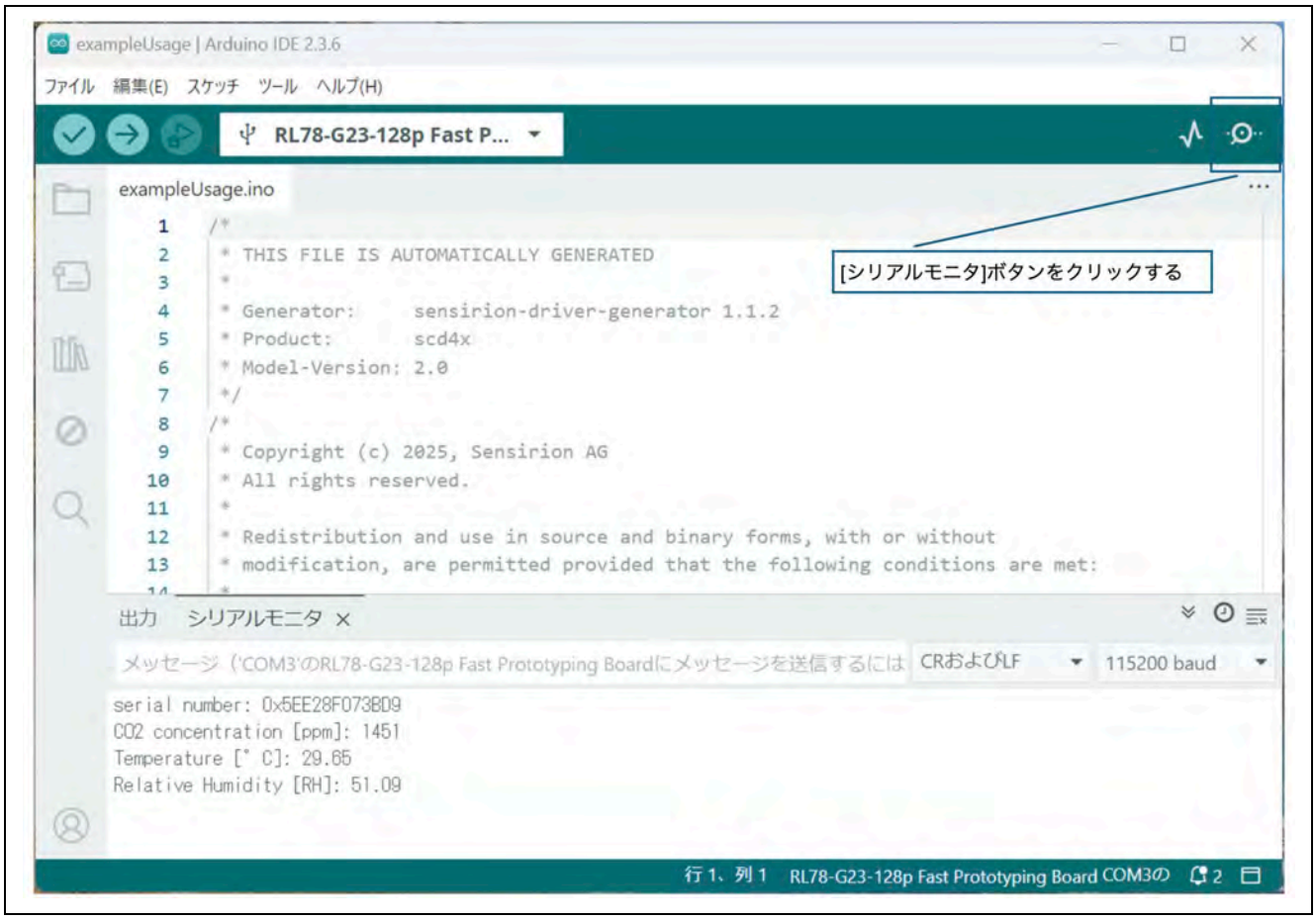


図 4-5 SCD40 の動作確認

## 4.2　ステップ２　IoT クラウドサービス「Ambient」へのデータ送信

ステップ 2 では、ダミーデータを Wi-Fi ネットワーク経由で IoT クラウドサービス Ambient に送信してグラフ表示します。

### 4.2.1　Ambient ライブラリのインストール

IoT クラウドサービス Ambient にデータを送信するには RL78/G23 Ambient ライブラリを使います。RL78/G23-128p FPB 製品ページに移動し、ドキュメントもしくはサンプルコードにて「Arduino」で検索をおこなうことでサンプルコードが表示されます。サンプルコードを PC の適当なフォルダにダウンロードします。

Arduino™ IDE の[スケッチ]メニュー→[ライブラリをインクルード]→[.ZIP 形式のライブラリをインストール]を選択し、ダウンロードした.ZIP 形式のライブラリ（RL78-G23-128p-Ambient.zip）を開きます。

図 4-6 Ambient ライブラリのインストール 1

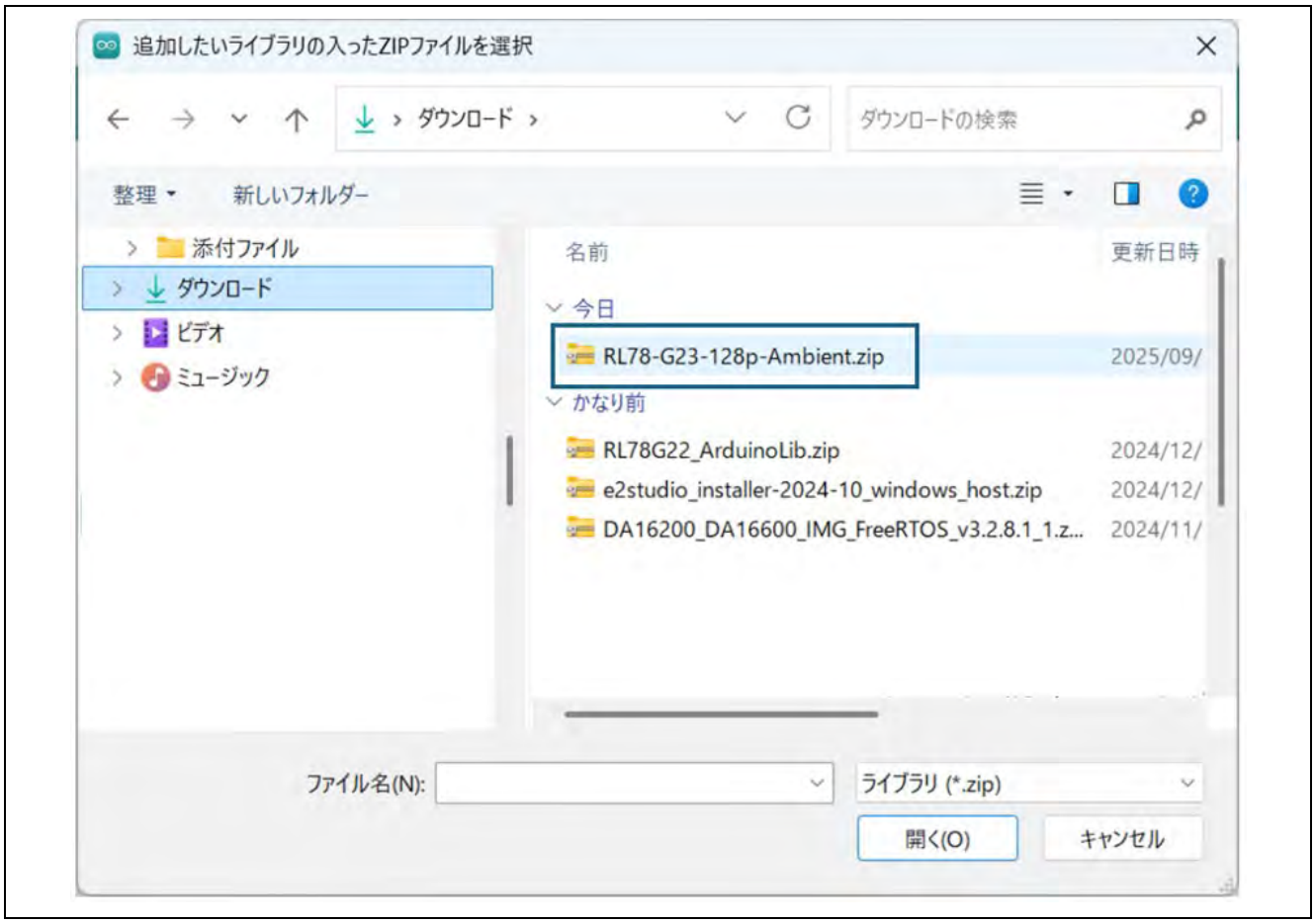


図 4-7 Ambient ライブラリのインストール 2

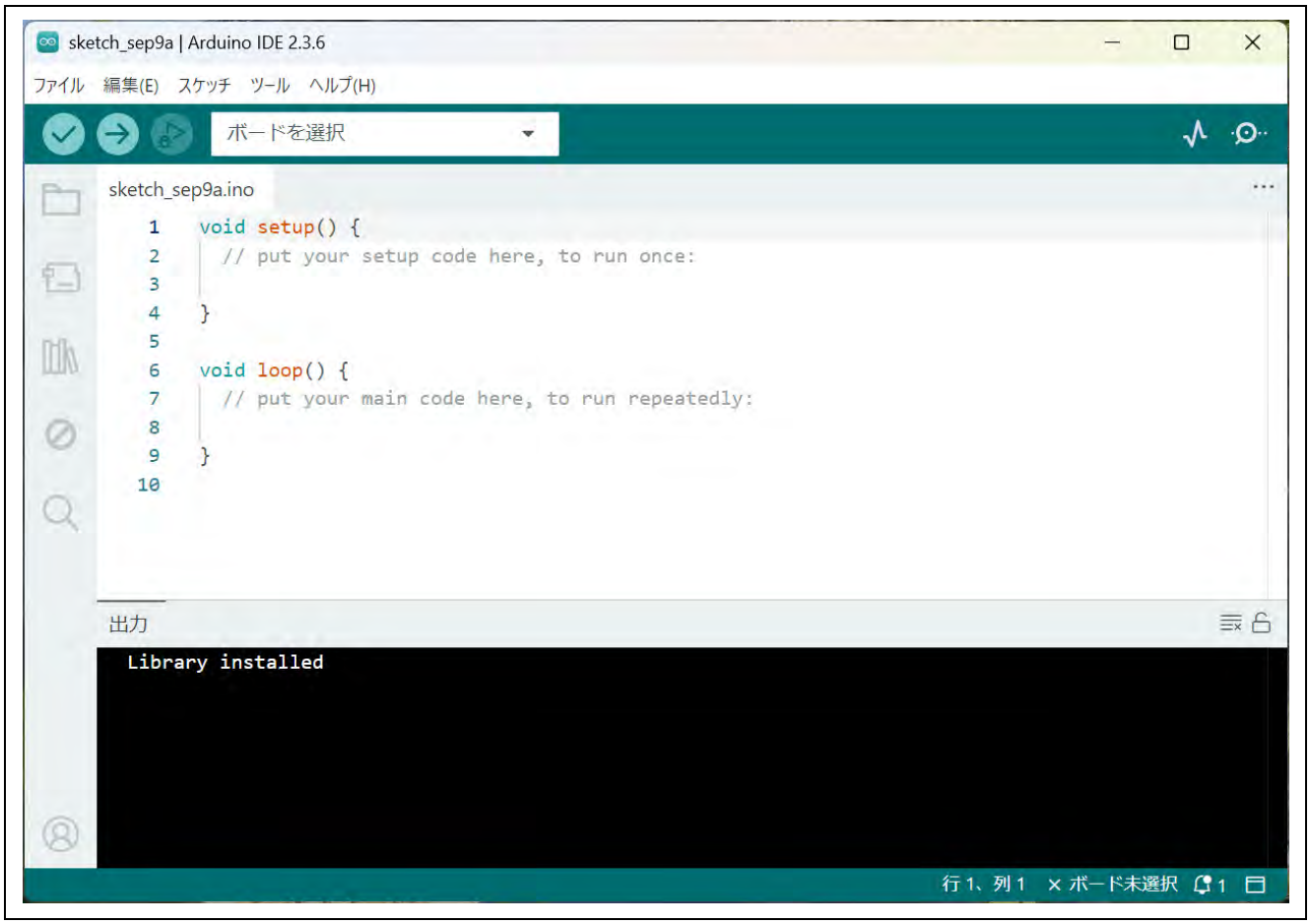


図 4-8 Ambient ライブラリのインストール 3

Ambient ライブラリがインストールされると「Library installed」と表示されます。

### 4.2.2　ステップ２のサンプルコード概要

本サンプルコードはステップ 2 のダミーデータをクラウドサービス Ambient に送信するプログラム（AmbientTest）とステップ 3 の CO2・温湿度センサ SCD40 から取得した CO2、温度、湿度データを Ambient に送信するプログラム（SCD40Ambient）が含まれます。ファイル構成を以下に示します。

```
RL78-G23-128p-Ambient
 ├ examples
 |    ├ AmbientTest
 |    |    └ AmbientTest.ino ←サンプルスケッチ
 |    └ SCD40Ambient
 |         └ SCD40Ambient.io ←サンプルスケッチ
 ├ src
 |    ├ Ambient.cpp ←AmbientライブラリのAPI関数を記述するソースファイル
 |    ├ Ambient.h ←Ambientライブラリを定義するヘッダファイル
 |    ├ DA16x00.cpp ←WiFiインタフェースDA16x00のAPI関数を記述するソースファイル
 |    ├ DA16x00.h ←WiFiインタフェースDA16x00を定義するヘッダファイル
 |    ├ WiFi.cpp ←WiFiモジュールのAPI関数を記述するソースファイル
 |    └ WiFi.h ←WiFiモジュールを定義するヘッダファイル
 ├ keywords.txt
 └ library.properties
```

図 4-9 サンプルコードのファイル構成

### 4.2.3 Ambient ライブラリの API 関数

Ambient ライブラリに実装する API 関数一覧を以下に示します。

| API 関数名 | 機能 |
|---|---|
| bool begin(channelId, writeKey, readKey) | Ambient 呼び出し準備 |
| bool set(field, data) | 送信データのセット |
| bool send() | Ambient へのデータ送信 |
| bool read(buf, len, n) | Ambient からのデータ受信 |
| int status | Ambient 呼び出し処理結果 |

Ambient ライブラリの関数仕様を以下に示します。

bool begin(channelId, writeKey, readKey)

| | | |
|---|---|---|
| 概要 | Ambient 呼び出し準備 | |
| 引数 | unsigned long channelId | チャネル ID |
| | const char *writeKey | ライトキー |
| | const char *readKey | リードキー |
| 戻り値 | 戻り値項目 | 呼び出し準備処理結果 |
| | 戻り値 | true: 正常終了 |
| | | false: エラー |
| | データ型 | bool |

bool set(field, data)

| | | |
|---|---|---|
| 概要 | 送信データのセット | |
| 引数 | int field | フィールド |
| | int data | データ |
| 戻り値 | 戻り値項目 | データセット処理結果 |
| | 戻り値 | true: 正常終了 |
| | | false: エラー |
| | データ型 | bool |

bool send()

| | | |
|---|---|---|
| 概要 | Ambient へのデータ送信 | |
| 引数 | なし | |
| 戻り値 | 戻り値項目 | データ送信処理結果 |
| | 戻り値 | true: 正常終了 |
| | | false: エラー |
| | データ型 | bool |

bool read(buf, len, n)

| | | |
|---|---|---|
| 概要 | Ambient からのデータ受信 | |
| 引数 | char *buf | バッファへのポインタ |
| | int len | バッファサイズ |
| | int n | 読み込むデータ個数 |
| 戻り値 | 戻り値項目 | データ受信処理結果 |
| | 戻り値 | true: 正常終了 |
| | | false: エラー |
| | データ型 | bool |

int status

| | | |
|---|---|---|
| 概要 | Ambient 呼び出し処理結果 | |
| 戻り値 | 戻り値項目 | Ambient 呼び出し処理結果 |
| | 戻り値 | HTTP ステータスコード |
| | | 200: 正常終了 |
| | データ型 | int |

### 4.2.4　Ambient のユーザー登録とチャネル生成

Ambient を使うには、まず無料のユーザー登録が必要です。ブラウザで Ambient サイト（https://ambidata.io）をアクセスし、「ユーザー登録（無料）」ボタンをクリックします。「Ambient 利用規約」を確認し、問題がなければ、ユーザー登録画面でメールアドレスとパスワードを入力し、「ユーザー登録」ボタンをクリックすると、入力したメールアドレスに確認メールが送られてきます。メール中の URL をクリックするとユーザー登録は完了です。

Ambient ではマイコンから送信するデーターを「チャネル」という単位で管理します。 データーを送る際は「チャネル」を指定して送ります。 ユーザー登録して Ambient にログインすると、チャネル一覧ページが表示されます。最初はチャネルがないので、「チャネルを作る」ボタンがあるだけです。「チャネルを作る」ボタンをクリックするとチャネルが作られ、チャネル ID、このチャネルからデータを受信する時に必要なリードキー、データを送信する時に必要なライトキーが割り当てられます。リードキー、ライトキーは英数字で構成された文字列です。

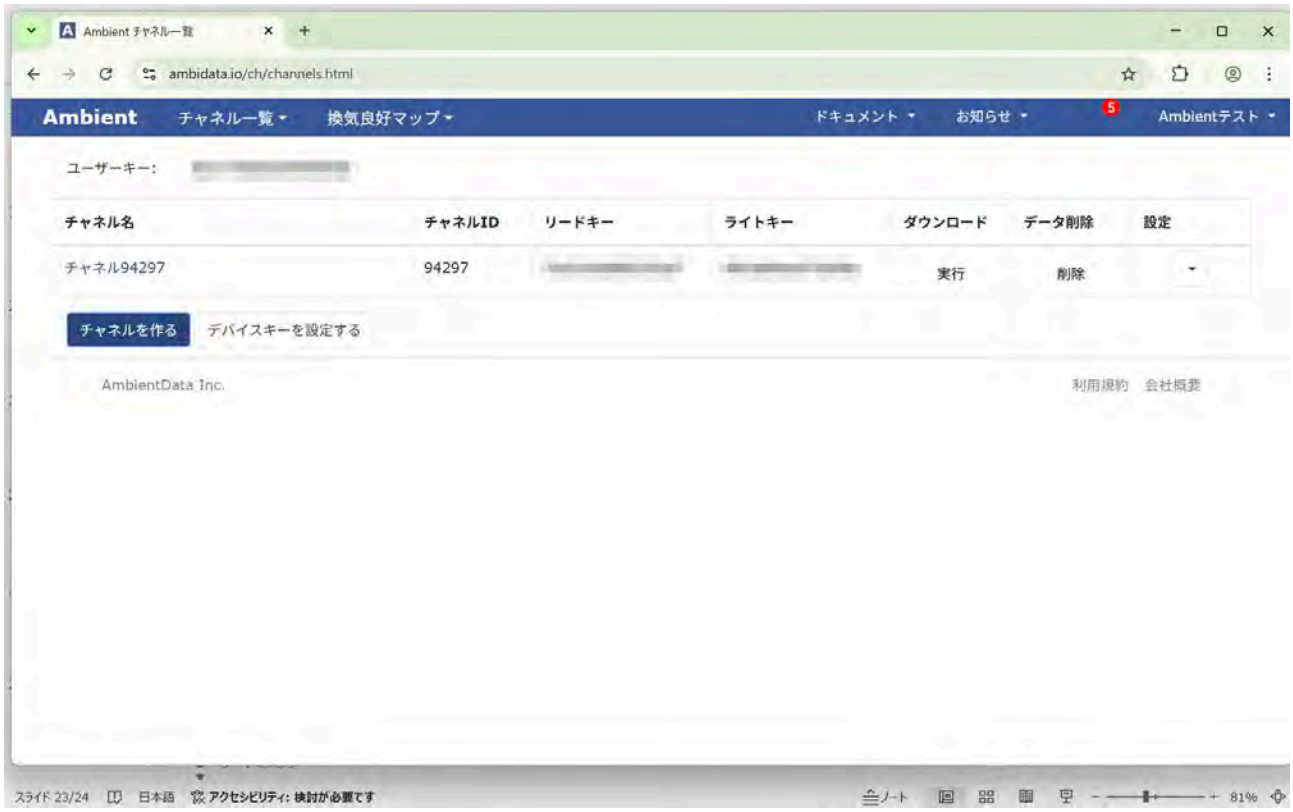


図 4-10 チャネル生成

### 4.2.5　ステップ２のプログラム概要

ステップ 2 のプログラム概要を以下に示します。

ステップ 2 のプログラム（AmbientTest.ino）は最初に 1 回実行される setup()関数と、繰り返し実行される loop()関数から出来ています。

setup()関数では、da16x00.begin()関数で WiFi インタフェース DA16x00 の初期設定をおこない、WiFi.begin()関数で ssid と password を指定して WiFi に接続要求を出します。WiFi.status()関数の値が WL_CONNECTED（接続完了）になるまで待ち、WiFi への接続が完了したら am.begin()関数で Ambient の呼び出し準備をおこないます。なお、シリアルモニタに文字を出力する Serial 関数の説明は省略しています。

```
void setup() {
  const char* ssid = "ssid";   // SSID
  const char* password = "password";   // password

  unsigned long channelId = 100; // your Ambient channel ID
  const char* writeKey = "writeKey"; // writeKey

  da16x00.begin();   // Initialize DA16x00 WiFi module

  WiFi.begin(ssid, password); // send connect request to WiFi
  while (WiFi.status() != WL_CONNECTED) {   // wait for connection
    delay(500);
    Serial.print('.');
  }

  am.begin(channelId, writeKey); // Initialize Ambient
}
```

loop 関数では count というダミーデータを用い、am.set()関数で 1 番目のデータとして count の値を、2 番目のデータとして count の 2 乗の値をセットし、am.send()関数でこれらのデータを Ambient に送信しています。送信が終わったら count の値を 1 増やしています。Ambient への最短送信間隔が 5 秒なので、5 秒間待って loop 関数を終わります。

```
uint32_t loopTime = 5000;
int count = 0;

void loop() {
  am.set(1, count);   // set dummy data 1 to Ambient data 1
  am.set(2, count * count);   // set dummy data 2 to Ambient data 2
  am.send();   // send data to Ambient

  count++;
  delay(loopTime);
}
```

### 4.2.6　ステップ 2 の動作確認

ステップ 2 の動作確認手順を以下に示します。

Arduino™ IDE の[ファイル]メニュー→[スケッチ例]→[RL78 G23 128p Ambient]→[AmbientTest]をクリックし、サンプルスケッチ AmbientTest.ino を開きます。

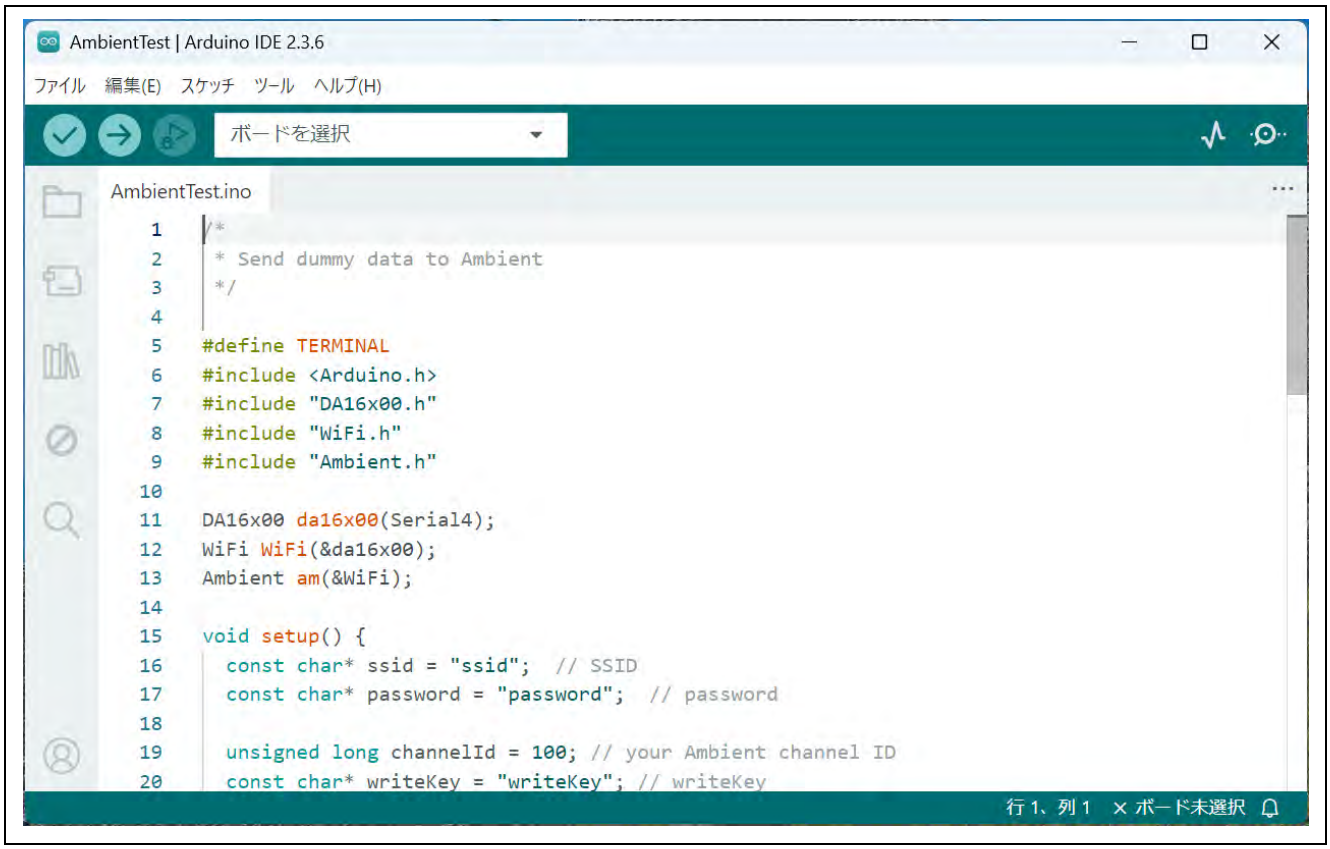


図 4-11 ステップ 2 サンプルスケッチを開く

16 行目の”ssid”と 17 行目の”password”を、接続する WiFi ルータのものに書き直します。DA16X00 は 2.4GHz で通信するので、ssid も 2.4GHz のものを選択します。19 行目の channelId と 20 行目の writeKey は 4.2.4 Ambient のユーザー登録とチャネル生成で作ったチャネルの channelId と writeKey に書き直します。

Arduino™ IDE のツールバーに「ボードを選択」と表示されていたら、その部分をクリックし、RL78/G23 128p ボードが接続されているポート番号を選択します。「他のボードとポートを選択」画面の検索欄に「rl78」と入力すると「RL78-G23-128p Fast Prototyping Board」が表示されるので、それを選択します。

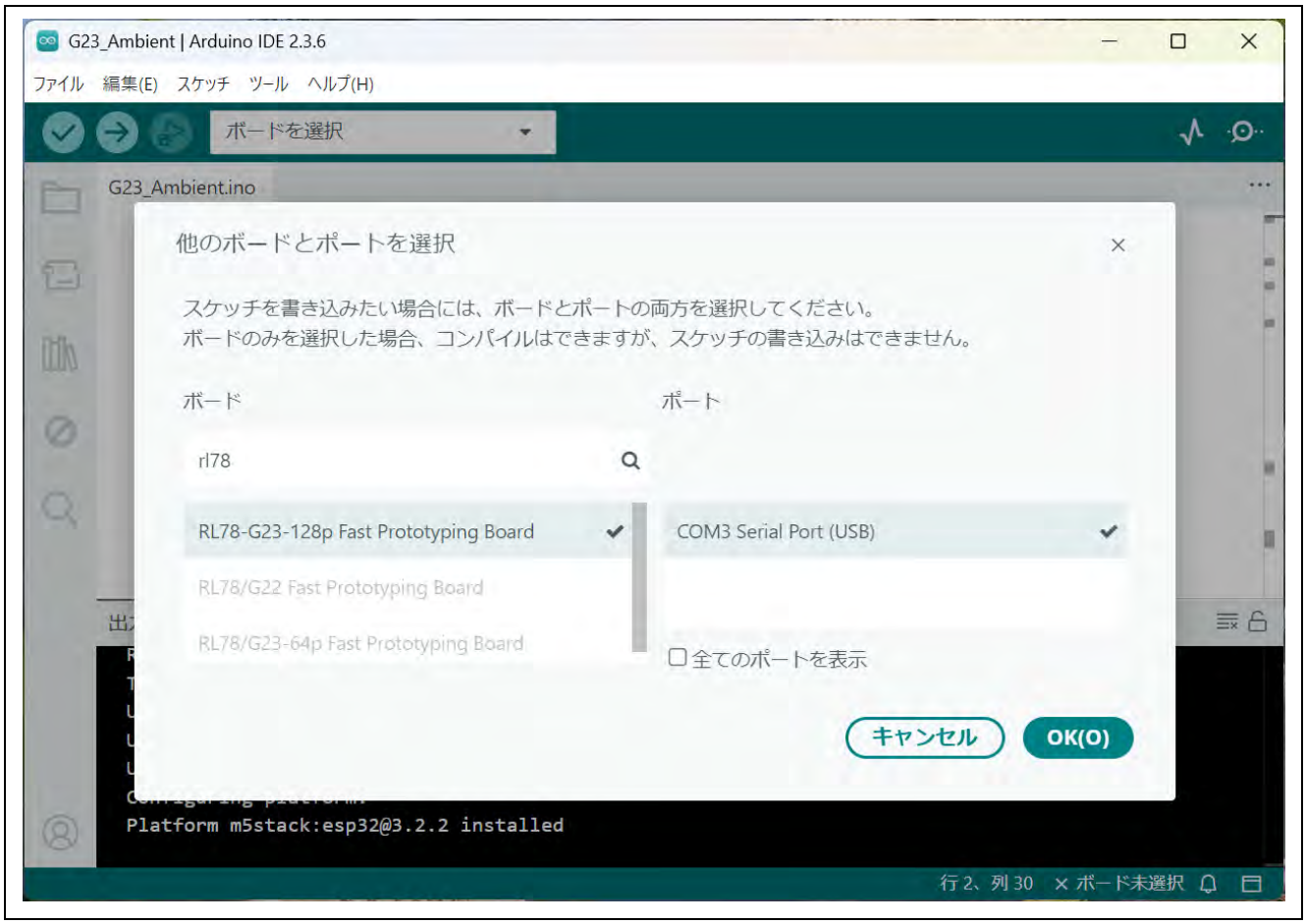


図 4-12 ポートとボードの選択

Ssid、password、channelId、writeKey を書き直し、ポートとボードを選択したら、Arduino™ IDE の[書き込み]ボタンをクリックしてプログラムをビルドし、ボードに書き込みます。

書き込みが終わったら、Arduino™ IDE の[シリアルモニタ]ボタンをクリックしてシリアルモニタを表示します。プログラムが動作し、シリアルモニタに図 4-12 のようにダミーデータが 5 秒おきに表示されることを確認します。

ブラウザで Ambient サイトにログインし、チャネル一覧画面からデータを送信したチャネルをクリックすると、図 4-13 のように Ambient に送信されたデータがグラフ化されて表示されることが確認できます。

図 4-13 ステップ 2 のプログラム起動

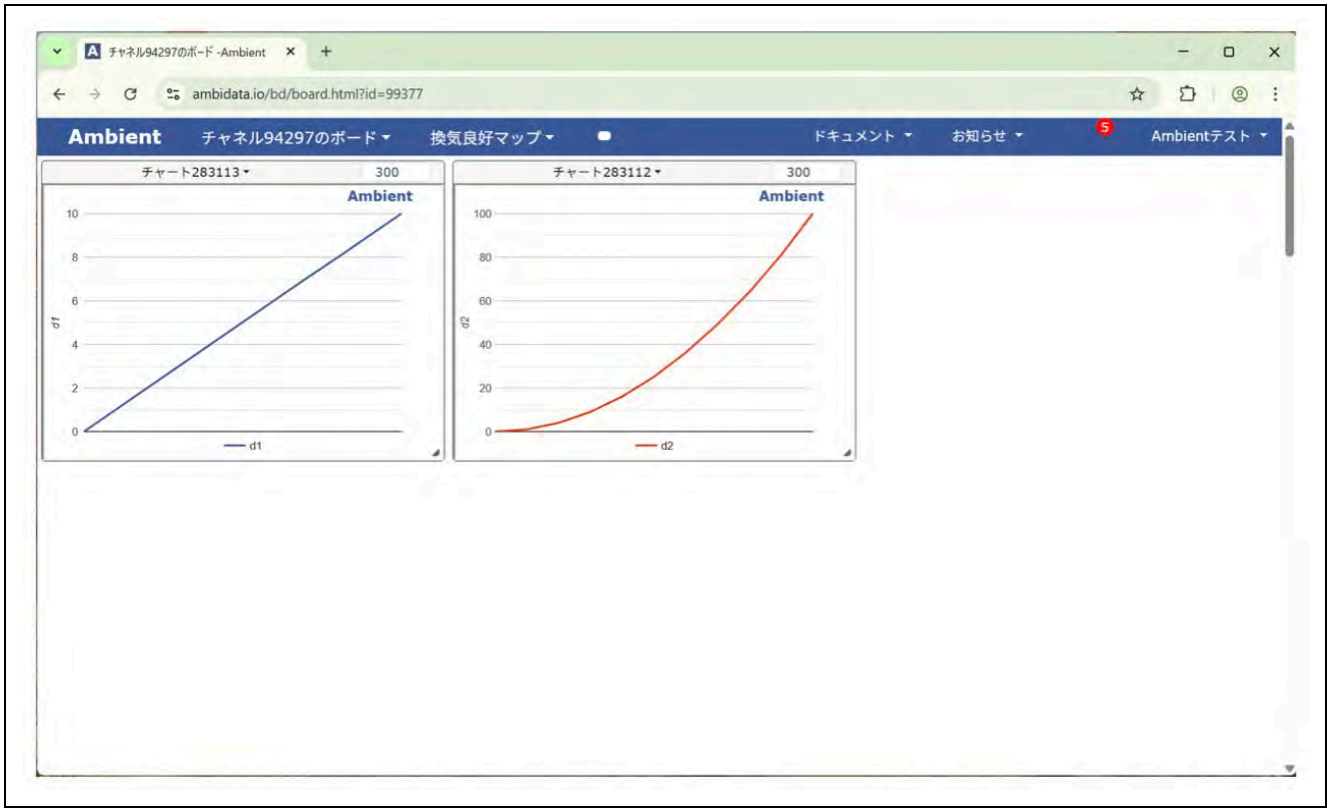


図 4-14 Ambient でのデータ表示

## 4.3　ステップ３　センサデータの Ambient への送信

ステップ３では、ステップ１とステップ２のプログラムを組み合わせ、CO2、温湿度センサ SCD40 からデータを取得し、そのデータを Ambient に送信し、Ambient でグラフ化して確認します。

### 4.3.1　ステップ３の動作確認

ステップ３の CO2・温湿度センサ SCD40 から取得した CO2、温度、湿度データを Ambient に送信するスケッチ（SCD40Ambient）は Ambient ライブラリに含まれています。

Arduino™ IDE の[ファイル]メニュー→[スケッチ例]→[RL78 G23 128p Ambient]→[SCD40Ambient]をクリックし、サンプルスケッチ SCD40Ambient.ino を開きます。

サンプルスケッチが開いたら、ステップ２と同様に ssid と password、channelId と writeKey をお使いの環境に合わせて書き直します。

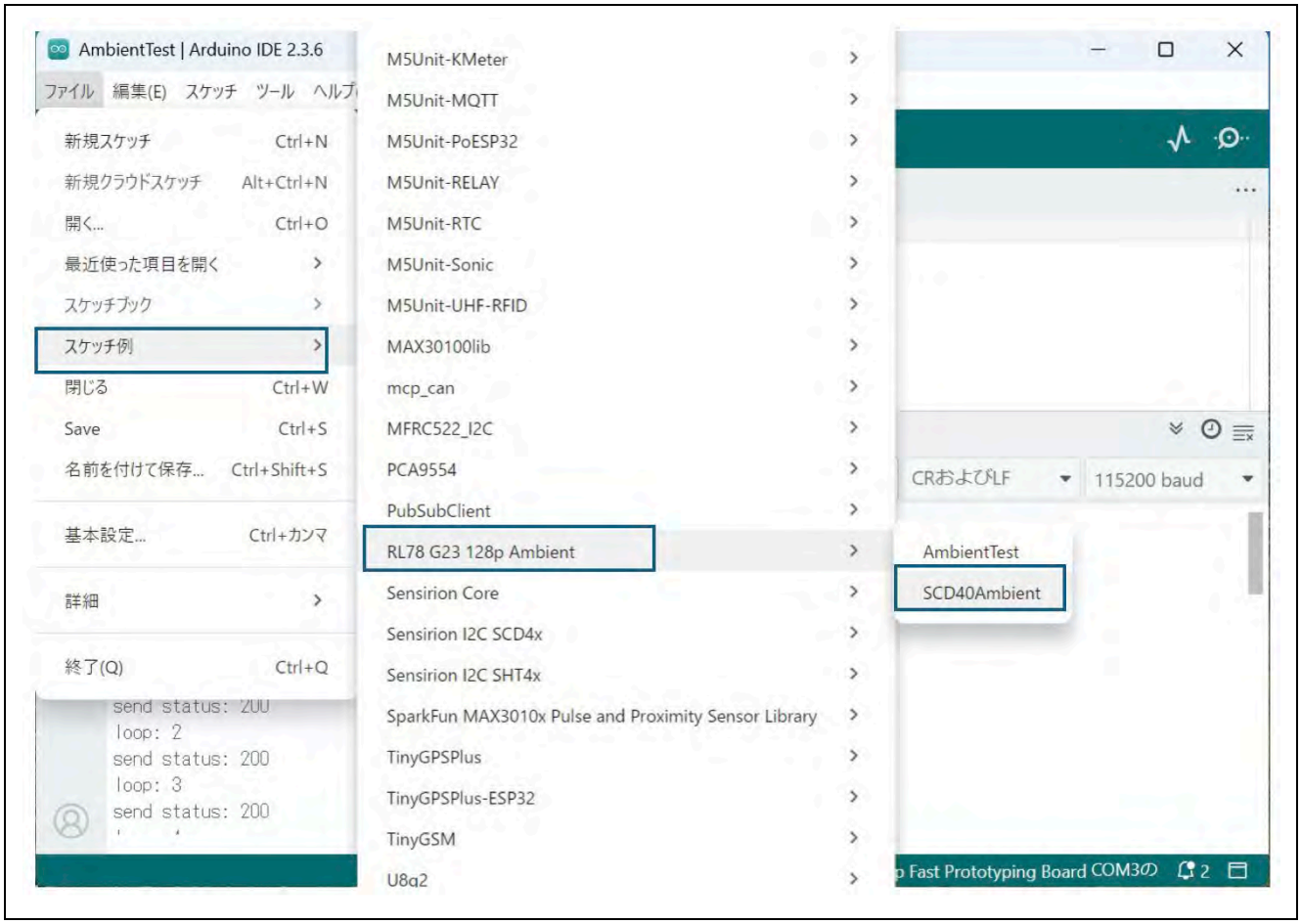


図 4-15 ステップ３のスケッチを開く

Arduino™ IDE の[書き込み]ボタンをクリックし、プログラムをビルドしてボードに書き込みます。書き込みが完了したら[シリアルモニタ]ボタンをクリックしてシリアルモニタを表示すると、図 4-15 のように「start」の文字列の次に SCD40 センサのシリアル番号が表示され、DA16x00 が起動され、WiFi に接続さ

れ、Ambient の初期設定がおこなわれたメッセージが表示されます。次に 30 秒ごとに SCD40 センサから取得した CO2 濃度、温度、湿度が表示され、Ambient への送信結果が成功していれば 200 が表示されます。

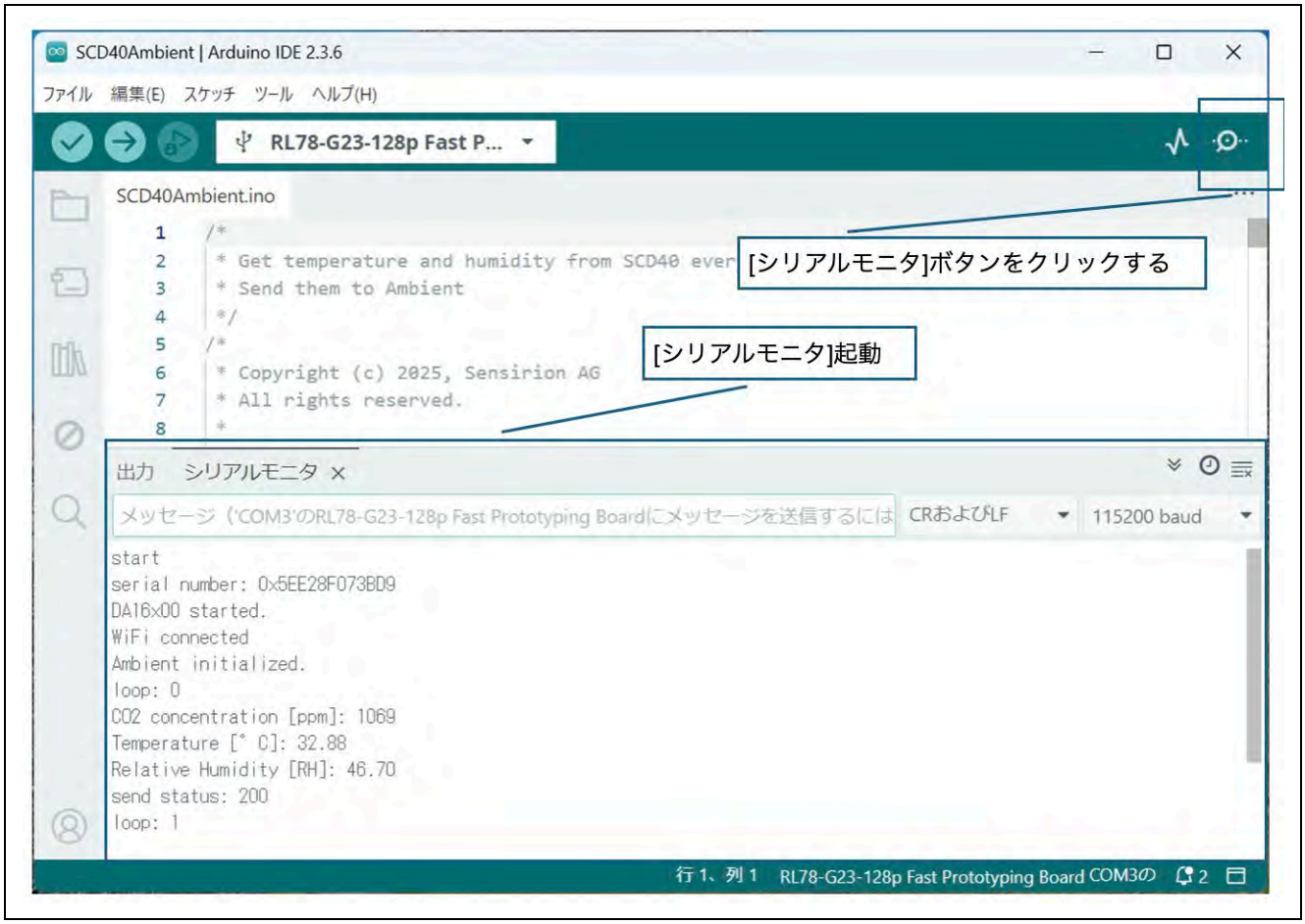


図 4-16 ステップ 3 のプログラム起動

ブラウザで Ambient サイトのデータを送信したチャネル画面を見ると温度、湿度、CO2 の値がグラフ表示されていることが確認できます。

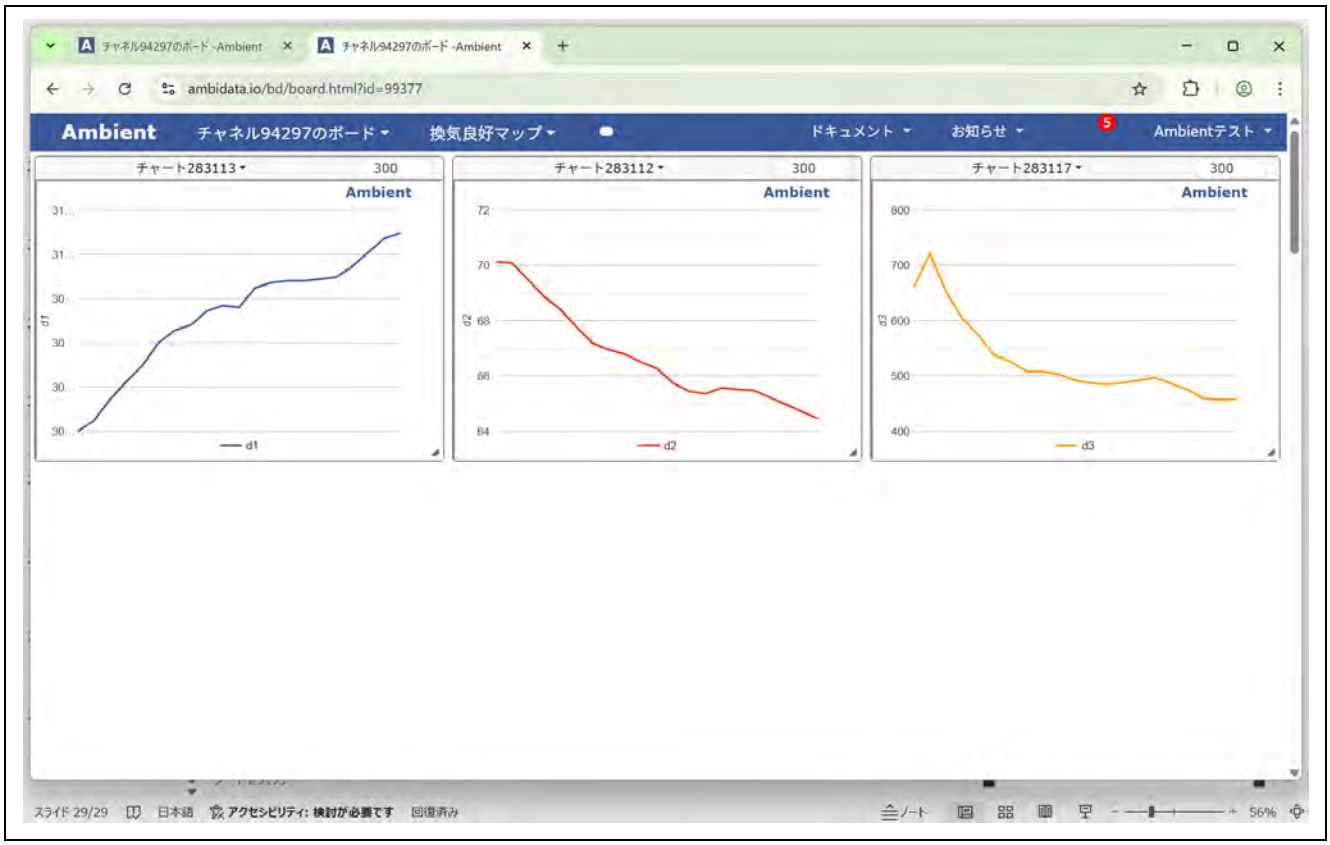


図 4-17 Ambient でのデータ表示

# 5. 注意事項

## 5.1 ビルドが始まらない

ボードを選択していない状態（Arduino™ IDE のツールバーに「ボードを選択」と表示されている状態）では[書き込み]ボタンを押してもビルドが始まりません。3.3.4 の手順に従ってボードとポートを選択してから書き込みをおこなってください。

## 5.2 書き込みがエラーになる

シリアルモニタを起動している状態で[書き込み]を行うと、ビルドは実行されますが、その後、図 5-1 のように「Failed uploading: uploading error: exit status 4098」と表示され、書き込みが失敗します。「シリアルモニタ」の表示の右側の「x」印をクリックしてシリアルモニタを閉じてから書き込みをおこなってください。

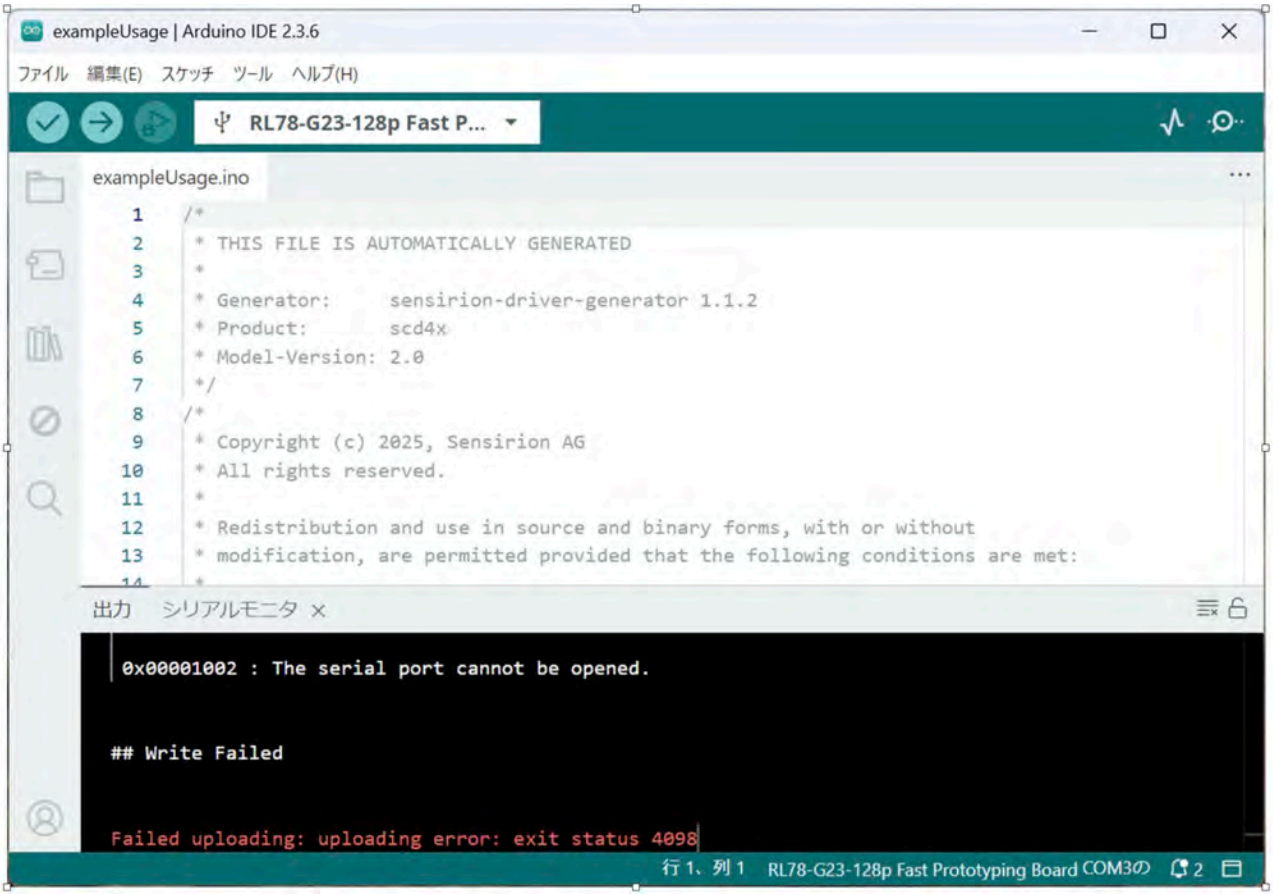


図 5-1 シリアルモニタ起動中は書き込みエラーになる

## 改訂記録

| Rev. | 発行日 | 改訂内容 | |
|---|---|---|---|
| | | ページ | ポイント |
| 1.00 | Apr. 28. 26 | - | 初版 |
| | | | |

# 製品ご使用上の注意事項

ここでは、マイコン製品全体に適用する「使用上の注意事項」について説明します。個別の使用上の注意事項については、本ドキュメントおよびテクニカルアップデートを参照してください。

1. 静電気対策

   CMOS 製品の取り扱いの際は静電気防止を心がけてください。CMOS 製品は強い静電気によってゲート絶縁破壊を生じることがあります。運搬や保存の際には、当社が出荷梱包に使用している導電性のトレーやマガジンケース、導電性の緩衝材、金属ケースなどを利用し、組み立て工程にはアースを施してください。プラスチック板上に放置したり、端子を触ったりしないでください。また、CMOS 製品を実装したボードについても同様の扱いをしてください。

2. 電源投入時の処置

   電源投入時は、製品の状態は不定です。電源投入時には、LSI の内部回路の状態は不確定であり、レジスタの設定や各端子の状態は不定です。外部リセット端子でリセットする製品の場合、電源投入からリセットが有効になるまでの期間、端子の状態は保証できません。同様に、内蔵パワーオンリセット機能を使用してリセットする製品の場合、電源投入からリセットのかかる一定電圧に達するまでの期間、端子の状態は保証できません。

3. 電源オフ時における入力信号

   当該製品の電源がオフ状態のときに、入力信号や入出力プルアップ電源を入れないでください。入力信号や入出力プルアップ電源からの電流注入により、誤動作を引き起こしたり、異常電流が流れ内部素子を劣化させたりする場合があります。資料中に「電源オフ時における入力信号」についての記載のある製品は、その内容を守ってください。

4. 未使用端子の処理

   未使用端子は、「未使用端子の処理」に従って処理してください。CMOS 製品の入力端子のインピーダンスは、一般に、ハイインピーダンスとなっています。未使用端子を開放状態で動作させると、誘導現象により、LSI 周辺のノイズが印加され、LSI 内部で貫通電流が流れたり、入力信号と認識されて誤動作を起こす恐れがあります。

5. クロックについて

   リセット時は、クロックが安定した後、リセットを解除してください。プログラム実行中のクロック切り替え時は、切り替え先クロックが安定した後に切り替えてください。リセット時、外部発振子（または外部発振回路）を用いたクロックで動作を開始するシステムでは、クロックが十分安定した後、リセットを解除してください。また、プログラムの途中で外部発振子（または外部発振回路）を用いたクロックに切り替える場合は、切り替え先のクロックが十分安定してから切り替えてください。

6. 入力端子の印加波形

   入力ノイズや反射波による波形歪みは誤動作の原因になりますので注意してください。CMOS 製品の入力がノイズなどに起因して、$V_{IL}$（Max.）から $V_{IH}$（Min.）までの領域にとどまるような場合は、誤動作を引き起こす恐れがあります。入力レベルが固定の場合はもちろん、$V_{IL}$（Max.）から $V_{IH}$（Min.）までの領域を通過する遷移期間中にチャタリングノイズなどが入らないように使用してください。

7. リザーブアドレス（予約領域）のアクセス禁止

   リザーブアドレス（予約領域）のアクセスを禁止します。アドレス領域には、将来の拡張機能用に割り付けられている リザーブアドレス（予約領域）があります。これらのアドレスをアクセスしたときの動作については、保証できませんので、アクセスしないようにしてください。

8. 製品間の相違について

   型名の異なる製品に変更する場合は、製品型名ごとにシステム評価試験を実施してください。同じグループのマイコンでも型名が違うと、フラッシュメモリ、レイアウトパターンの相違などにより、電気的特性の範囲で、特性値、動作マージン、ノイズ耐量、ノイズ輻射量などが異なる場合があります。型名が違う製品に変更する場合は、個々の製品ごとにシステム評価試験を実施してください。

# ご注意書き

1. 本資料に記載された回路、ソフトウェアおよびこれらに関連する情報は、半導体製品の動作例、応用例を説明するものです。回路、ソフトウェアおよびこれらに関連する情報を使用する場合、お客様の責任において、お客様の機器・システムを設計ください。これらの使用に起因して生じた損害（お客様または第三者いずれに生じた損害も含みます。以下同じです。）に関し、当社は、一切その責任を負いません。
2. 当社製品または本資料に記載された製品データ、図、表、プログラム、アルゴリズム、応用回路例等の情報の使用に起因して発生した第三者の特許権、著作権その他の知的財産権に対する侵害またはこれらに関する紛争について、当社は、何らの保証を行うものではなく、また責任を負うものではありません。
3. 当社は、本資料に基づき当社または第三者の特許権、著作権その他の知的財産権を何ら許諾するものではありません。
4. 当社製品を組み込んだ製品の輸出入、製造、販売、利用、配布その他の行為を行うにあたり、第三者保有の技術の利用に関するライセンスが必要となる場合、当該ライセンス取得の判断および取得はお客様の責任において行ってください。
5. 当社製品を、全部または一部を問わず、改造、改変、複製、リバースエンジニアリング、その他、不適切に使用しないでください。かかる改造、改変、複製、リバースエンジニアリング等により生じた損害に関し、当社は、一切その責任を負いません。
6. 当社は、当社製品の品質水準を「標準水準」および「高品質水準」に分類しており、各品質水準は、以下に示す用途に製品が使用されることを意図しております。

   標準水準：　コンピュータ、OA 機器、通信機器、計測機器、AV 機器、家電、工作機械、パーソナル機器、産業用ロボット等

   高品質水準：輸送機器（自動車、電車、船舶等）、交通制御（信号）、大規模通信機器、金融端末基幹システム、各種安全制御装置等

   当社製品は、データシート等により高信頼性、Harsh environment 向け製品と定義しているものを除き、直接生命・身体に危害を及ぼす可能性のある機器・システム（生命維持装置、人体に埋め込み使用するもの等）、もしくは多大な物的損害を発生させるおそれのある機器・システム（宇宙機器と、海底中継器、原子力制御システム、航空機制御システム、プラント基幹システム、軍事機器等）に使用されることを意図しておらず、これらの用途に使用することは想定していません。たとえ、当社が想定していない用途に当社製品を使用したことにより損害が生じても、当社は一切その責任を負いません。
7. あらゆる半導体製品は、外部攻撃からの安全性を 100%保証されているわけではありません。当社ハードウェア／ソフトウェア製品にはセキュリティ対策が組み込まれているものもありますが、これによって、当社は、セキュリティ脆弱性または侵害（当社製品または当社製品が使用されているシステムに対する不正アクセス・不正使用を含みますが、これに限りません。）から生じる責任を負うものではありません。当社は、当社製品または当社製品が使用されたあらゆるシステムが、不正な改変、攻撃、ウイルス、干渉、ハッキング、データの破壊または窃盗その他の不正な侵入行為（「脆弱性問題」といいます。）によって影響を受けないことを保証しません。当社は、脆弱性問題に起因しまたはこれに関連して生じた損害について、一切責任を負いません。また、法令において認められる限りにおいて、本資料および当社ハードウェア／ソフトウェア製品について、商品性および特定目的との合致に関する保証ならびに第三者の権利を侵害しないことの保証を含め、明示または黙示のいかなる保証も行いません。
8. 当社製品をご使用の際は、最新の製品情報（データシート、ユーザーズマニュアル、アプリケーションノート、信頼性ハンドブックに記載の「半導体デバイスの使用上の一般的な注意事項」等）をご確認の上、当社が指定する最大定格、動作電源電圧範囲、放熱特性、実装条件その他指定条件の範囲内でご使用ください。指定条件の範囲を超えて当社製品をご使用された場合の故障、誤動作の不具合および事故につきましては、当社は、一切その責任を負いません。
9. 当社は、当社製品の品質および信頼性の向上に努めていますが、半導体製品はある確率で故障が発生したり、使用条件によっては誤動作したりする場合があります。また、当社製品は、データシート等において高信頼性、Harsh environment 向け製品と定義しているものを除き、耐放射線設計を行っておりません。仮に当社製品の故障または誤動作が生じた場合であっても、人身事故、火災事故その他社会的損害等を生じさせないよう、お客様の責任において、冗長設計、延焼対策設計、誤動作防止設計等の安全設計およびエージング処理等、お客様の機器・システムとしての出荷保証を行ってください。特に、マイコンソフトウェアは、単独での検証は困難なため、お客様の機器・システムとしての安全検証をお客様の責任で行ってください。
10. 当社製品の環境適合性等の詳細につきましては、製品個別に必ず当社営業窓口までお問合せください。ご使用に際しては、特定の物質の含有・使用を規制する RoHS 指令等、適用される環境関連法令を十分調査のうえ、かかる法令に適合するようご使用ください。かかる法令を遵守しないことにより生じた損害に関して、当社は、一切その責任を負いません。
11. 当社製品および技術を国内外の法令および規則により製造・使用・販売を禁止されている機器・システムに使用することはできません。当社製品および技術を輸出、販売または移転等する場合は、「外国為替及び外国貿易法」その他日本国および適用される外国の輸出管理関連法規を遵守し、それらの定めるところに従い必要な手続きを行ってください。
12. お客様が当社製品を第三者に転売等される場合には、事前に当該第三者に対して、本ご注意書き記載の諸条件を通知する責任を負うものといたします。
13. 本資料の全部または一部を当社の文書による事前の承諾を得ることなく転載または複製することを禁じます。
14. 本資料に記載されている内容または当社製品についてご不明な点がございましたら、当社の営業担当者までお問合せください。

注 1. 本資料において使用されている「当社」とは、ルネサス エレクトロニクス株式会社およびルネサス エレクトロニクス株式会社が直接的、間接的に支配する会社をいいます。

注 2. 本資料において使用されている「当社製品」とは、注 1 において定義された当社の開発、製造製品をいいます。

（Rev.5.0-1　2020.10）

## 本社所在地

〒135-0061 東京都江東区豊洲 3-2-24（豊洲フォレシア）

www.renesas.com

## お問合せ窓口

弊社の製品や技術、ドキュメントの最新情報、最寄の営業お問合せ窓口に関する情報などは、弊社ウェブサイトをご覧ください。

www.renesas.com/contact/

## 商標について

ルネサスおよびルネサスロゴはルネサス エレクトロニクス株式会社の商標です。すべての商標および登録商標は、それぞれの所有者に帰属します。

© 2026 Renesas Electronics Corporation. All rights reserved.