

RL78/G23

Simple Sensor Data Cloud Visualization - Ambient - Sample Sketch

(Arduino™ Sketch)

Introduction

This application note describes how to use the RL78/G23-128p Fast Prototyping Board (RL78/G23 FPB) and the ultra-low-power Wi-Fi + Bluetooth Low Energy combo Pmod board (DA16x00) to transmit CO₂, temperature, and humidity data acquired from the CO₂, temperature, and humidity sensor SCD40 via a Wi-Fi network to the IoT cloud service (Ambient) and display it as graphs. The explanation is divided into three steps: Step 1: Display and verify the CO₂, temperature, and humidity data acquired from the SCD40 sensor on the serial monitor. Step 2: Send dummy data to the IoT cloud via a Wi-Fi network and display it as a graph. Step 3: Combine steps 1 and 2 to send the CO₂, temperature, and humidity data to the IoT cloud service and display it as a graph.

Target Device

RL78/G23-128p Fast Prototyping Board RTK7RLG230CSN000BJ
Ultra-Low Power Wi-Fi + Bluetooth Low Energy Combo Pmod Board US159-DA16600EVZ
"CO₂ Sensor Module Using SCD40" AE-SCD40-BO

Trademarks

Arduino is a trademark of Arduino SA.

Contents

1. System overview	3
2. Operation confirmation environment	4
3. Build development environment.....	4
3.1 Board connection.....	5
3.2 List of pins used.....	6
3.3 Setup of Arduino™ IDE	6
3.3.1 Starting the Arduino™ IDE	6
3.3.2 Starting the Board Manager and installing the RL78/G23-128p Board information.....	6
3.3.3 Connection the board to the PC and selection the board in the Arduino™ IDE	8
4. Software	9
4.1 Step 1: Verifying SCD40 Sensor Operation	9
4.1.1 Installing the SCD40 Arduino Library	9
4.1.2 Verification of Step 1 Operation	11
4.2 Step 2: Sending Data to the IoT Cloud Service “Ambient”	13
4.2.1 Installing the Ambient Library	13
4.2.2 Step 2 sample code overview	15
4.2.3 Ambient library API functions	16
4.2.4 Ambient user registration and channel creation.....	18
4.2.5 Step 2 Program overview	21
4.2.6 Verifying Step 2 Operation	22
4.3 Step 3: Sending sensor data to Ambient.....	26
4.3.1 Verifying Step 3 Operation	26
5. Notes	28
5.1 Build does not start.....	28
5.2 Failed uploading	28
Revision History	30

1. System overview

This system consists of an RL78/G23-128p Fast Prototyping Board, an ultra-low-power Wi-Fi + Bluetooth Low Energy combo Pmod board (DA16x00), a CO2 sensor module equipped with an SCD40 (CO2, temperature, and humidity sensor), and an IoT cloud service (Ambient). Arduino™ IDE is used for creating and uploading programs. CO2, temperature, and humidity data acquired from the SCD40 sensor is transmitted via a Wi-Fi network to the IoT cloud service (Ambient), where it is displayed graphically for verification.

This application note divides the explanation into three steps: Step 1 displays and verifies CO2, temperature, and humidity data acquired from the SCD40 sensor on the serial monitor; Step 2 sends dummy data to the IoT cloud via a Wi-Fi network for graph display; Step 3 combines Steps 1 and 2 to send CO2, temperature, and humidity data to the IoT cloud service for graph display. The block diagram of the final Step 3 sketch (program) is shown below.

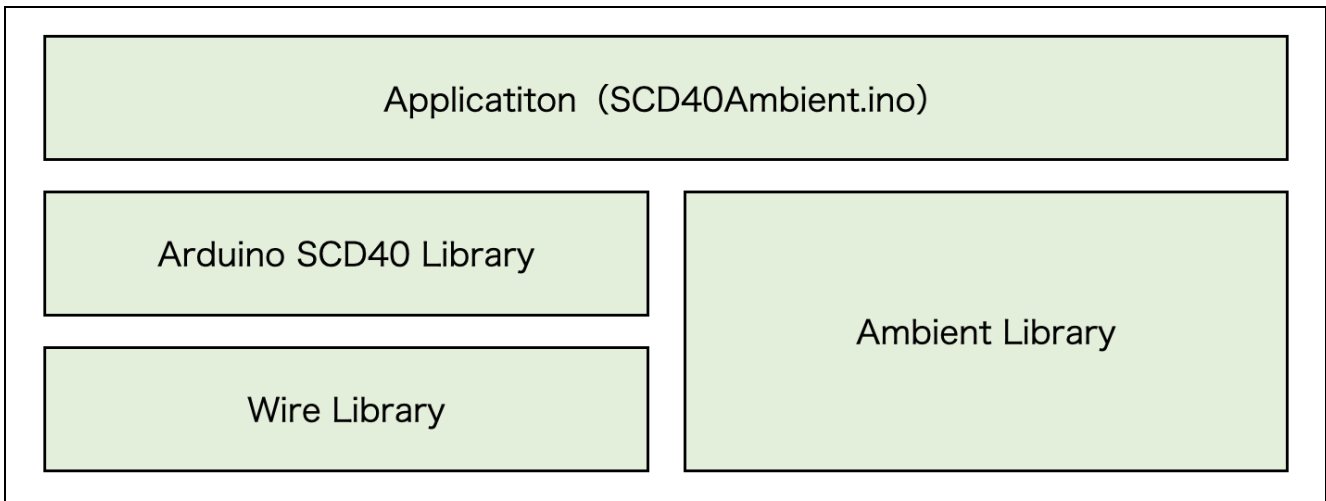


Figure 1-1 Block configuration of software for Step 3

The simple diagram of this system configuration is shown below.

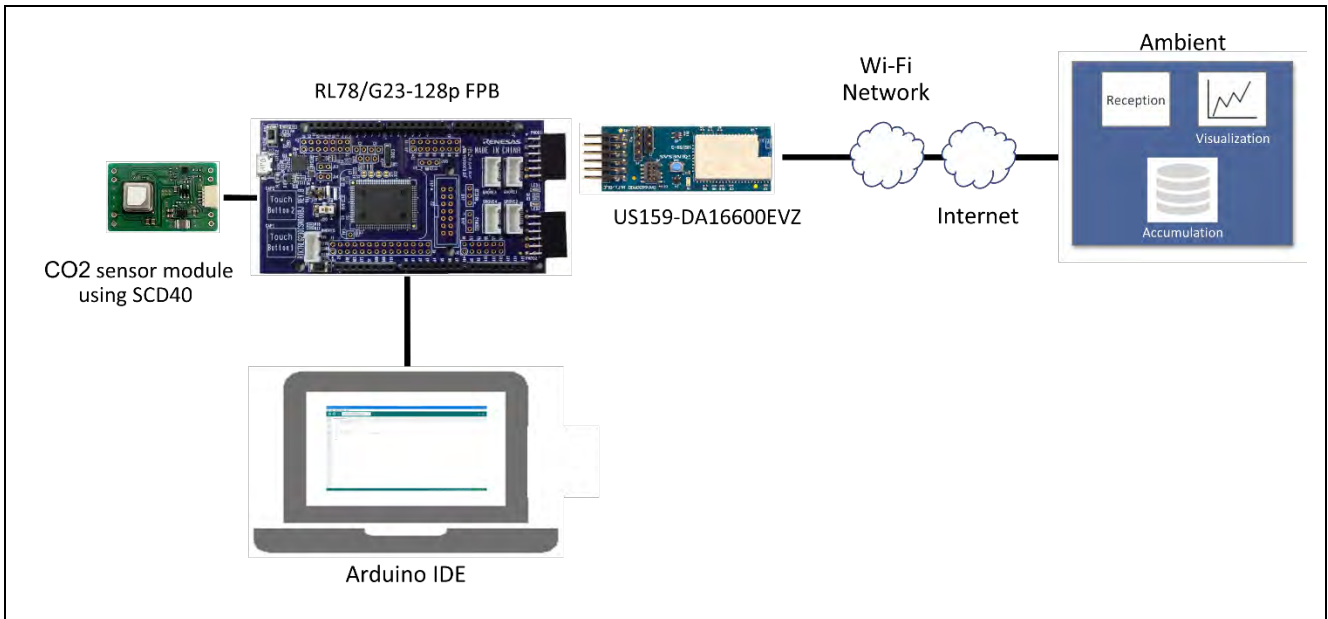


Figure 1-2 System configuration

2. Operation confirmation environment

The operating environment for this system is as follows:

Table 2-1 Operation confirmation environments (Hardware)

Item	Description
Evaluation board	RL78/G23-128p Fast Prototyping Board RTK7RLG230CSN000BJ
Communication module	US159-DA16600EVZ
Sensor board	CO2 Sensor Module AE-SCD40-BO Using SCD40 (Akizuki Denshi Tsusho)
Operating voltage	5V

Table 2-2 Operation confirmation environments (Software)

Item	Description	Version
OS	Windows 11	-
Integrated development environment (IDE)	Arduino™ IDE	2.3.6
Library	RL78/G23-128p FPB Library	1.2.0
	Sensirion I2C SCD4x	1.0.0

3. Build development environment

How to connect boards and how to set up the Arduino™ IDE are explained.

The Arduino™ IDE 2.3.6 is used in this system. Installation of the Arduino™ IDE 2.3.6 or later is necessary if it is not installed.

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

3.1 Board connection

A PC and evaluation board are connected via a USB cable, and the evaluation board and communication module are connected via the PMOD1 connector, as shown in Figure 3-1. Also, the evaluation board and sensor board are connected via jumper wires, as shown in Figure 3-2. To avoid cluttering the diagram, the connections are shown separately, but please connect the PC, evaluation board, communication module, and sensor board together.

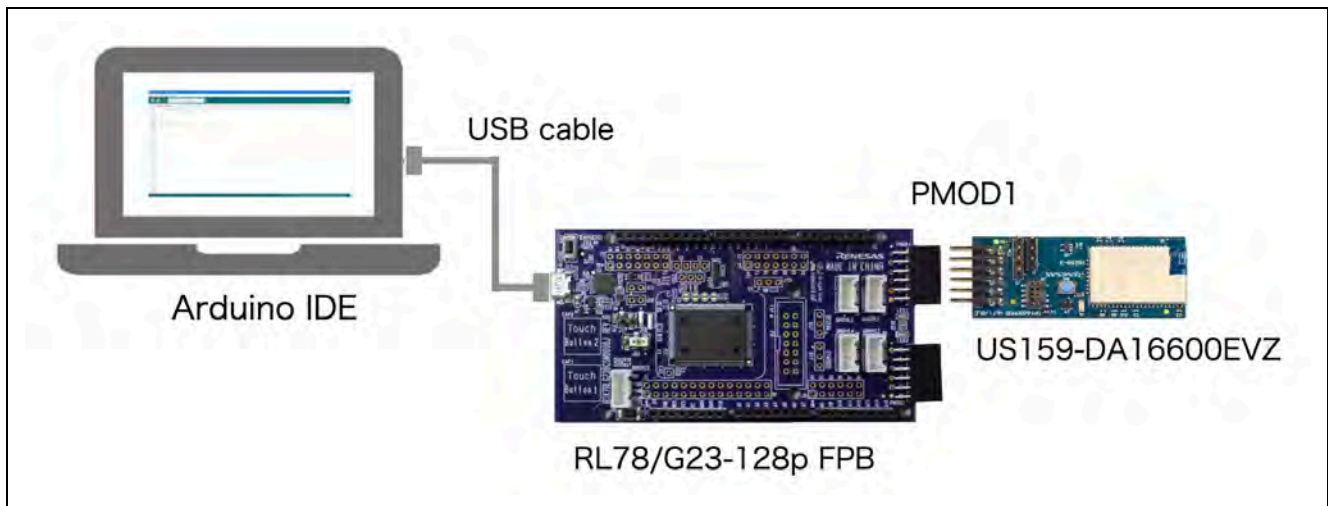


Figure 3-1 Connection of PC, evaluation board, and communication module

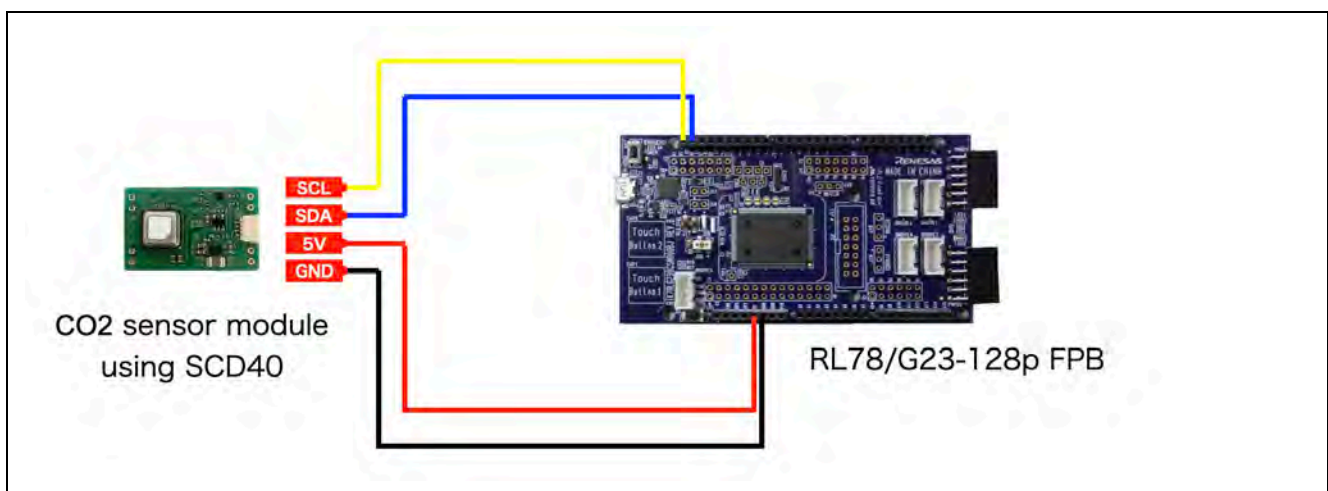


Figure 3-2 Connection of the evaluation board and sensor board

This system uses USB to supply power to the evaluation board. Please check the evaluation board's circuitry and set the jumpers as necessary.

In this system, set the evaluation board jumpers as follows:

Table 3-1 Jumper pins setting of evaluation board

Jumper pin	Setting	Function
J20	1-2 short	5V power supply to the microcontroller Factory default settings

3.2 List of pins used

In this system, the sensor board is connected using jumper wires. The pins used in this system are shown below.

Table 3-2 List of pins used in this system (sensor board)

Item	Arduino™ signal name	Pin
I2C	SDA	SDA
	SCL	SCL
VDD	5V	5V
GND	GND	GND

For detailed pin descriptions of each board, refer to the following manuals.

- RL78/G23-128p Fast Prototyping Board User's Manual (R20UT4870EJ0110)
- "CO2 Sensor Module Using SCD40"

3.3 Setup of Arduino™ IDE

The setup procedure of Arduino™ IDE is explained.

Remark. The setup procedure is almost the same as the procedure explained on the [Quick Start Guide : renesas/Arduino Wiki · GitHub](#). The sample sketch to flash LED is described on the above site. Refer to it if required.

3.3.1 Starting the Arduino™ IDE

Start the Arduino™ IDE.

3.3.2 Starting the Board Manager and installing the RL78/G23-128p Board information

Click the "Board Manager" icon in the left menu to start the Board Manager. Select "All" at the [Type] and enter "RL78" in the search field. Then, "RL78/G23-128p Fast Prototyping Board" is displayed. Next, click the [INSTALL]. Version 1.2.0 was used in this system.

After installation completes, click the "Board Manager" icon again to close the Board Manager window.

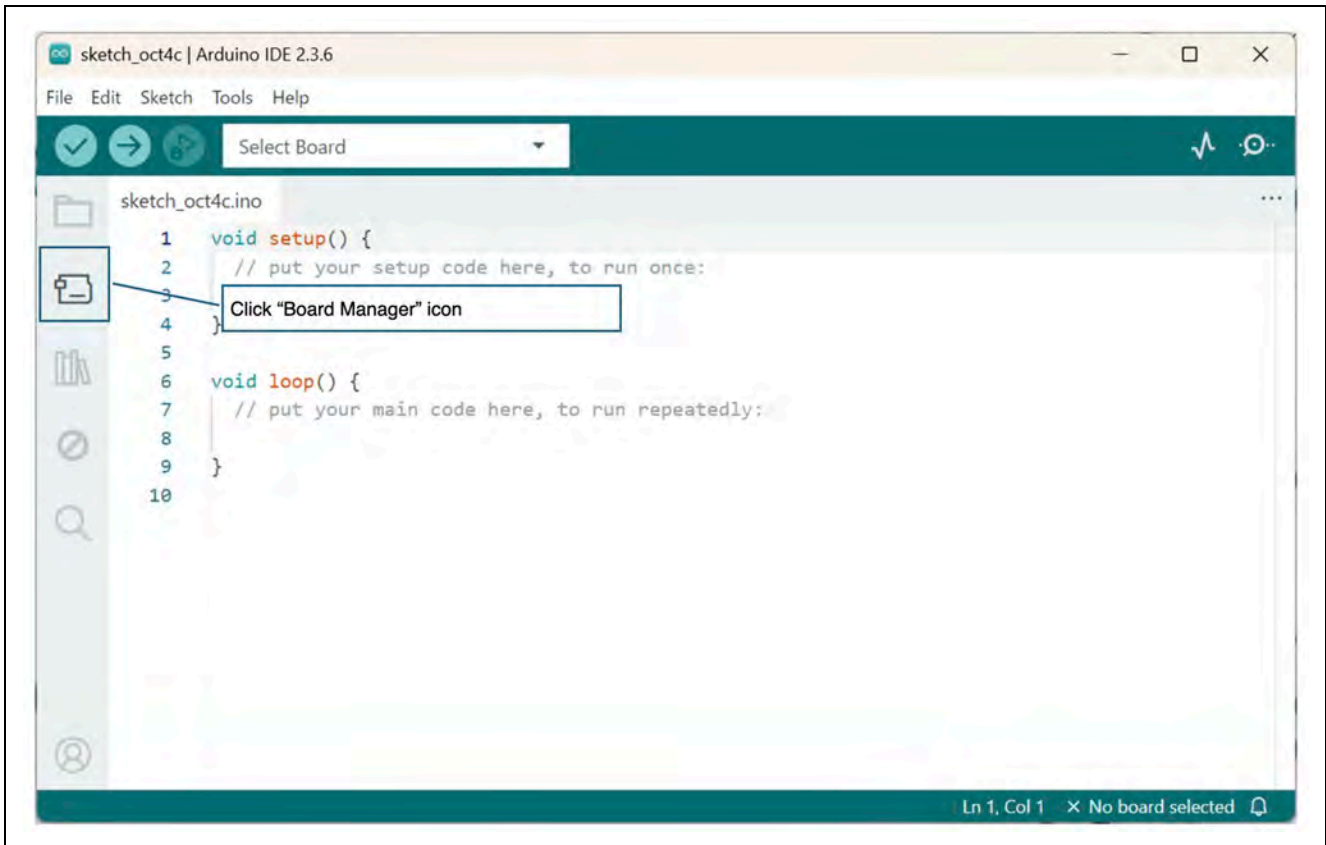


Figure 3-3 Starting the Board Manager

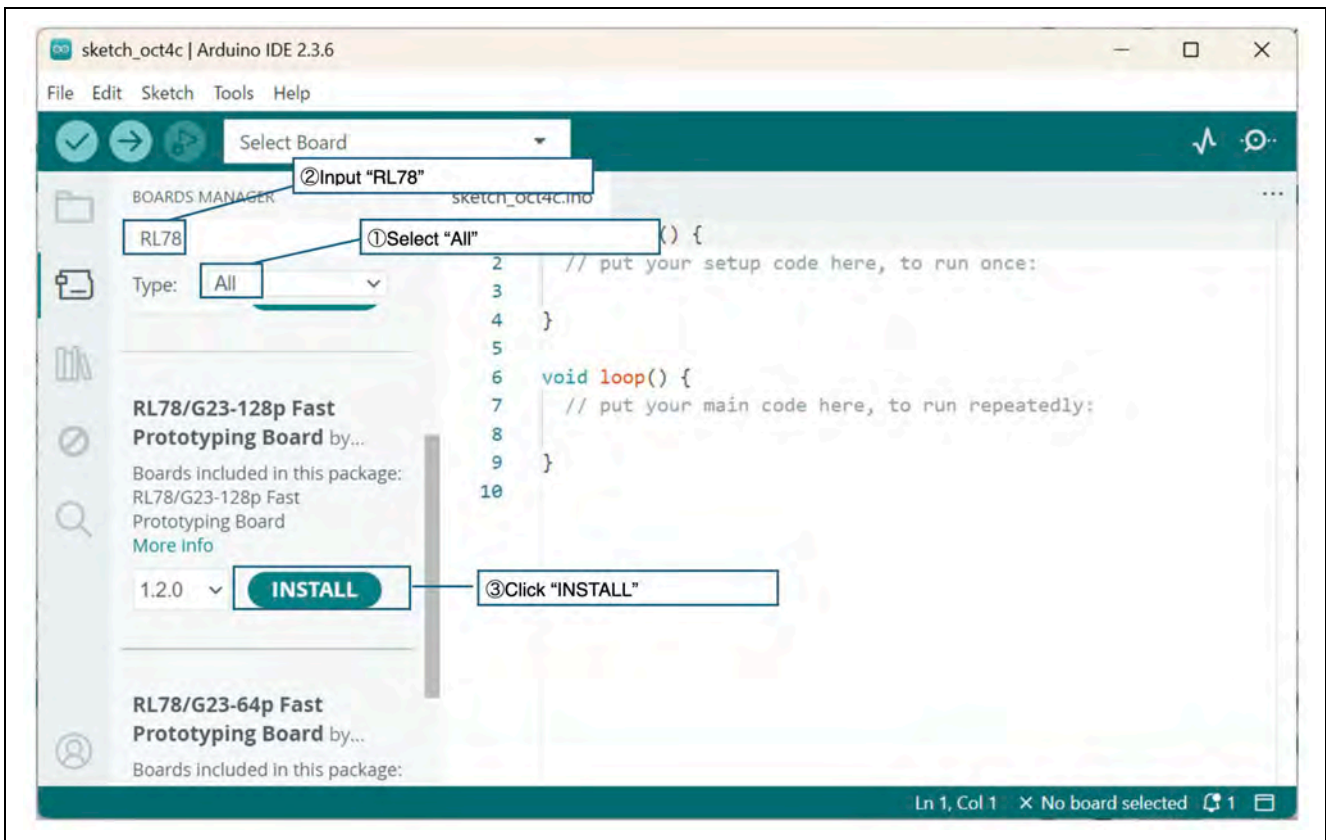


Figure 3-4 Installing the Board information

3.3.3 Connection the board to the PC and selection the board in the Arduino™ IDE

Connect the evaluation board to a PC's USB port using a USB cable. Click the “Select Board” field in the Arduino™ IDE toolbar and select the serial port assigned to the evaluation board. The COM port number can be confirmed by Windows Device Manager.

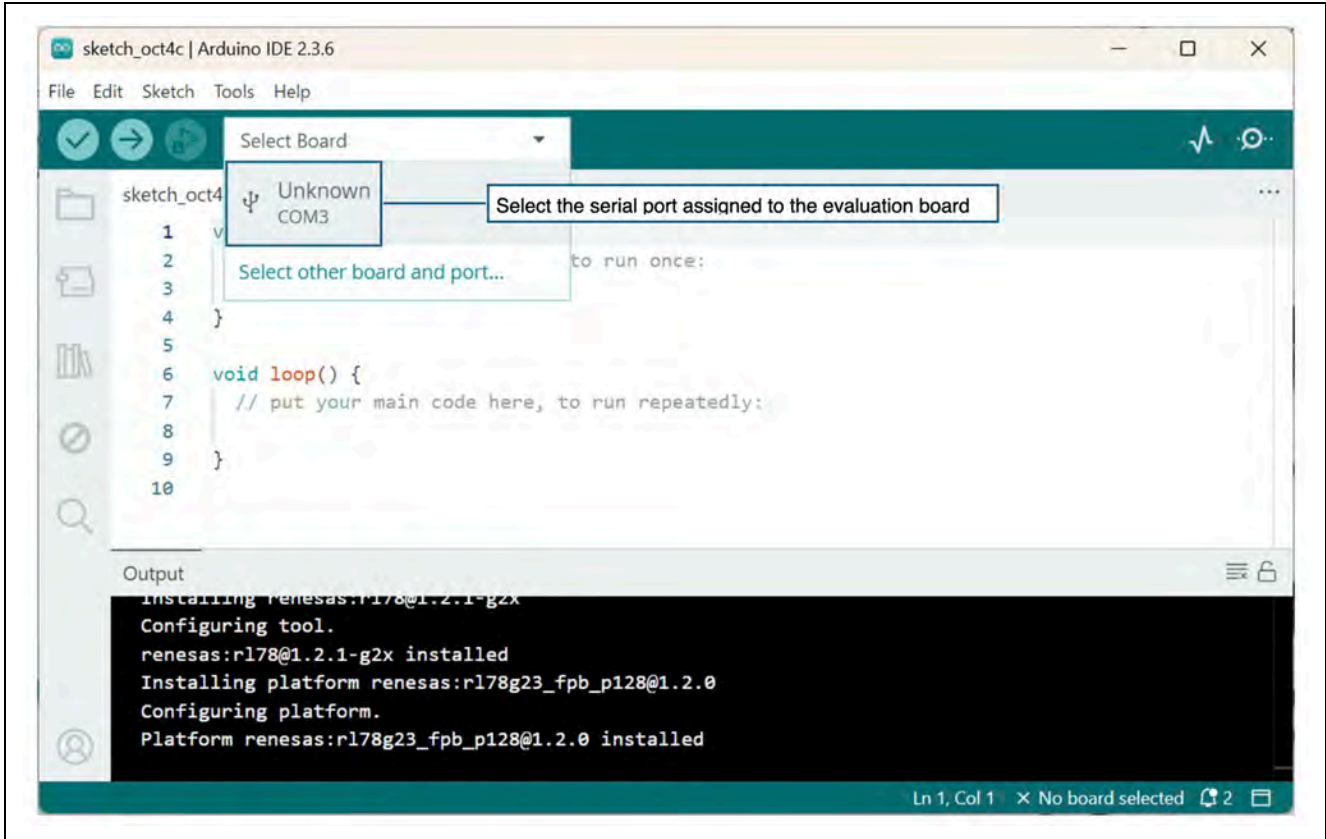


Figure 3-5 Selection of serial port

In the “Select other board and port...” screen, input “RL78” in the search field, select “RL78-G23-128p Fast Prototyping Board”, and click ‘OK’.

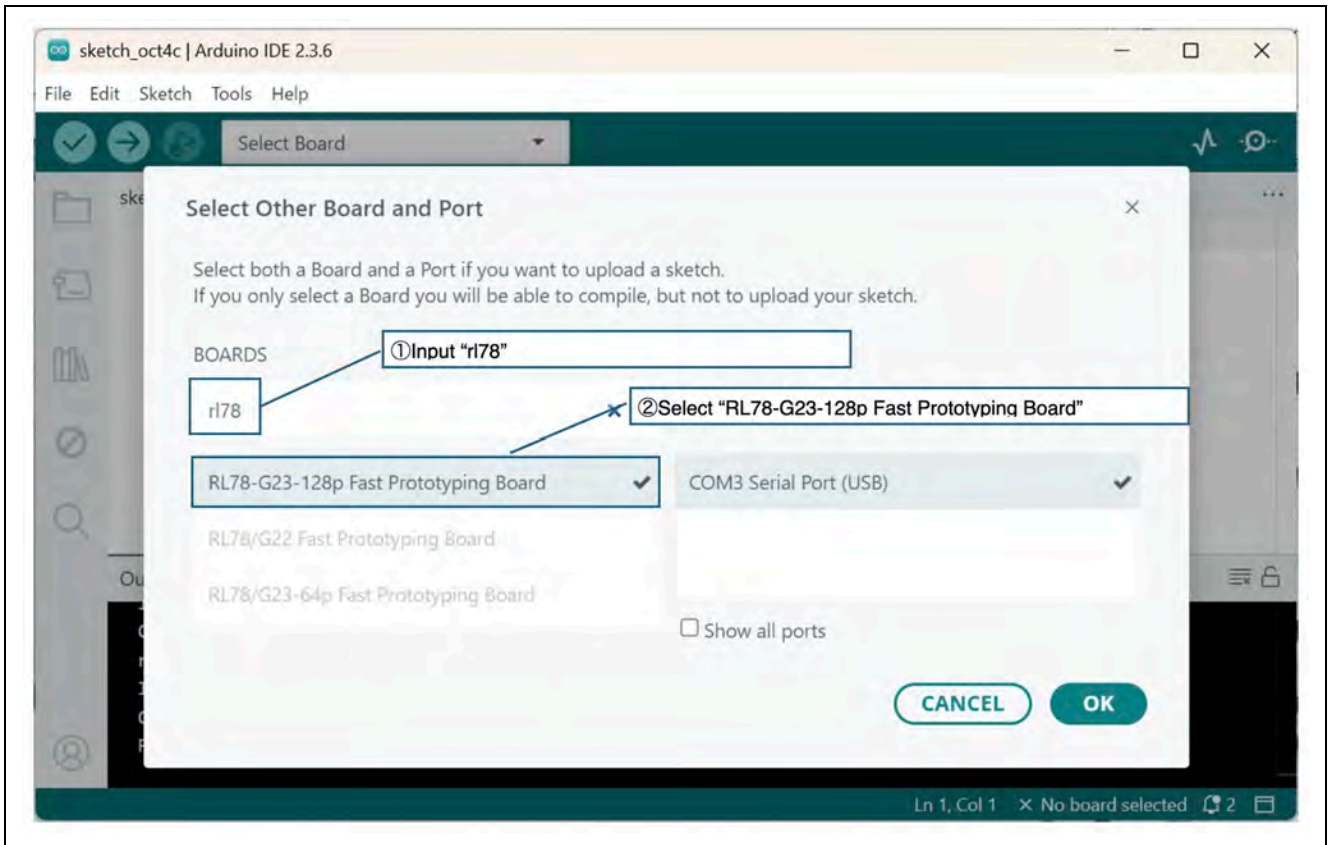


Figure 3-6 Selection of board

4. Software

This application note verifies operation in the following 3 steps:

Step 1: Acquire CO₂, temperature, and humidity data from the SCD40 sensor, display it on the serial monitor for verification

Step 2: Send dummy data to the IoT cloud service Ambient via the Wi-Fi network and display it as a graph

Step 3: Combine Steps 1 and 2 to send CO₂, temperature, and humidity data to Ambient and display it as a graph

4.1 Step 1: Verifying SCD40 Sensor Operation

The SCD40 sensor is developed by Sensirion. Sensirion provides an Arduino library and example sketches for accessing the SCD40. The SCD40 sensor operation verification in Step 1 will be performed using this Sensirion example sketch.

4.1.1 Installing the SCD40 Arduino Library

Install the SCD40 Arduino library in the Arduino™ IDE. Click the [Library Manager] icon in the left menu of the Arduino™ IDE to launch the Library Manager, then input “sensirion i2c scd4x” in the search field. Click the INSTALL button for the displayed “Sensirion I2C SCD4x” library to install it. Version 1.0.0 was used in this system.

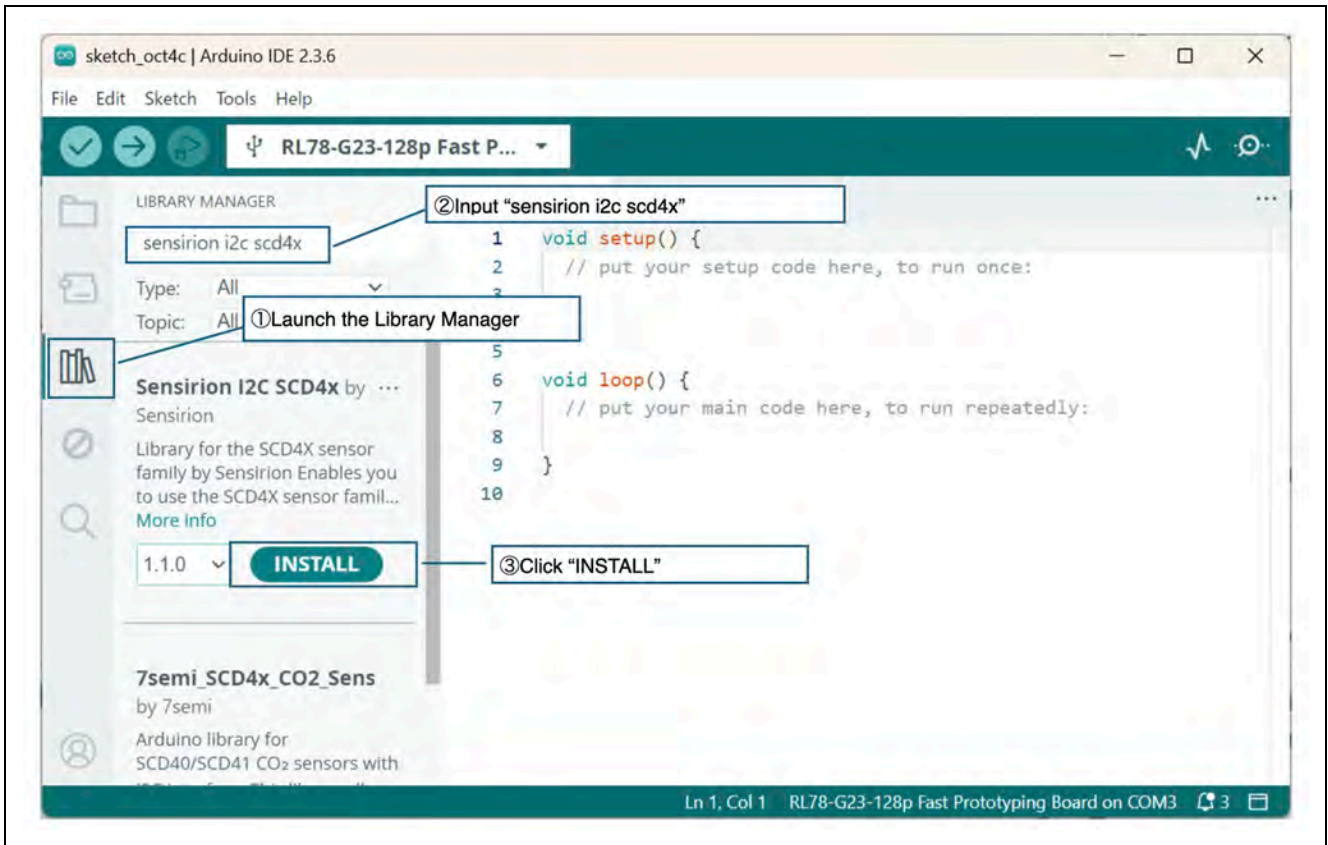


Figure 4-1 Installing the SCD40 Library

If asked “Would you like to install the missing dependency?” as shown in Figure 4-2, click “INSTALL ALL” to install the necessary libraries.

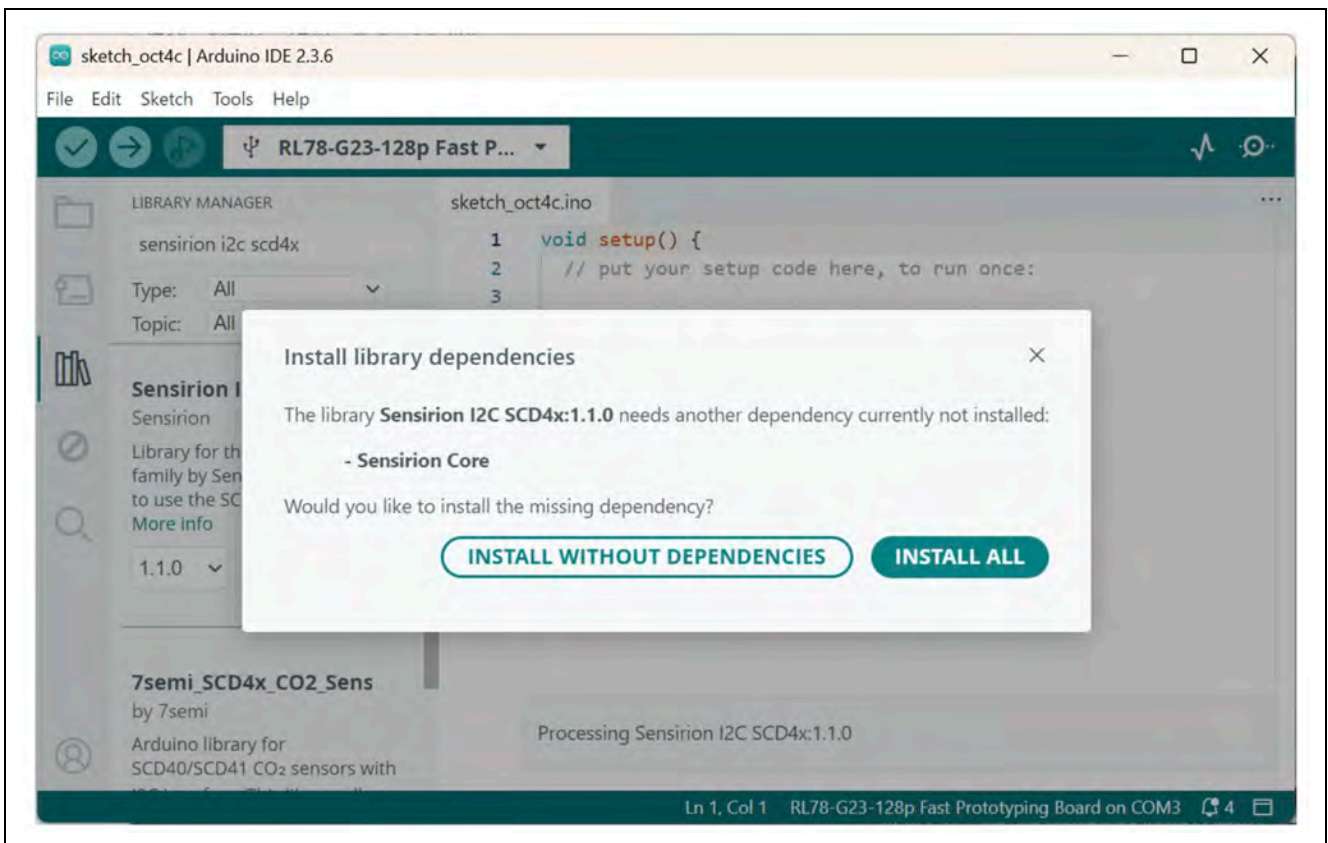


Figure 4-2 Installing missing dependencies

4.1.2 Verification of Step 1 Operation

Installing the SCD40 library automatically installs the example sketch. Select [File] menu → [Examples] → [Sensirion I2C SCD4x] in the Arduino™ IDE, then select “exampleUsage”. The example sketch “exampleUsage.ino” will open.

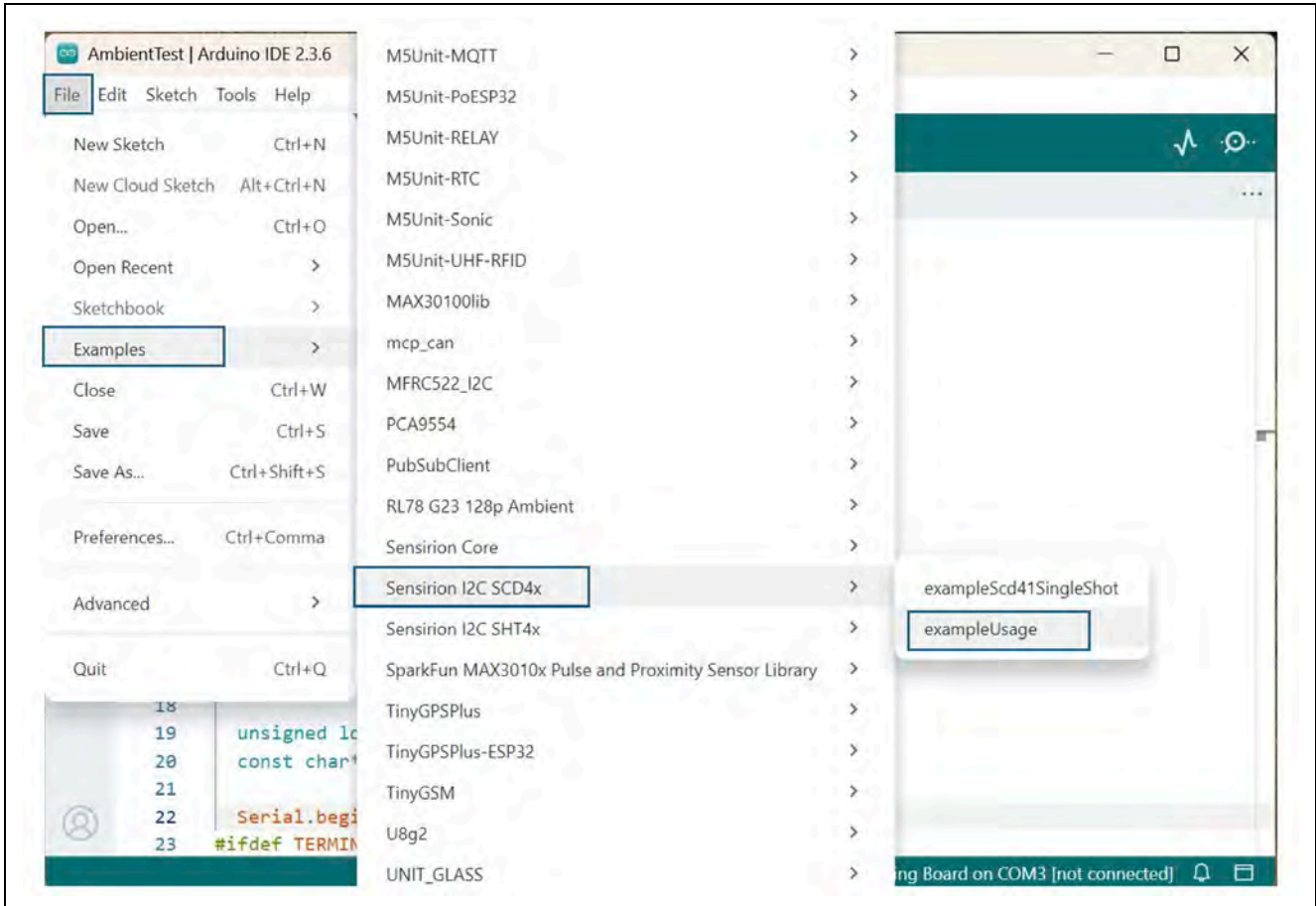


Figure 4-3 Select SCD40 sketch example

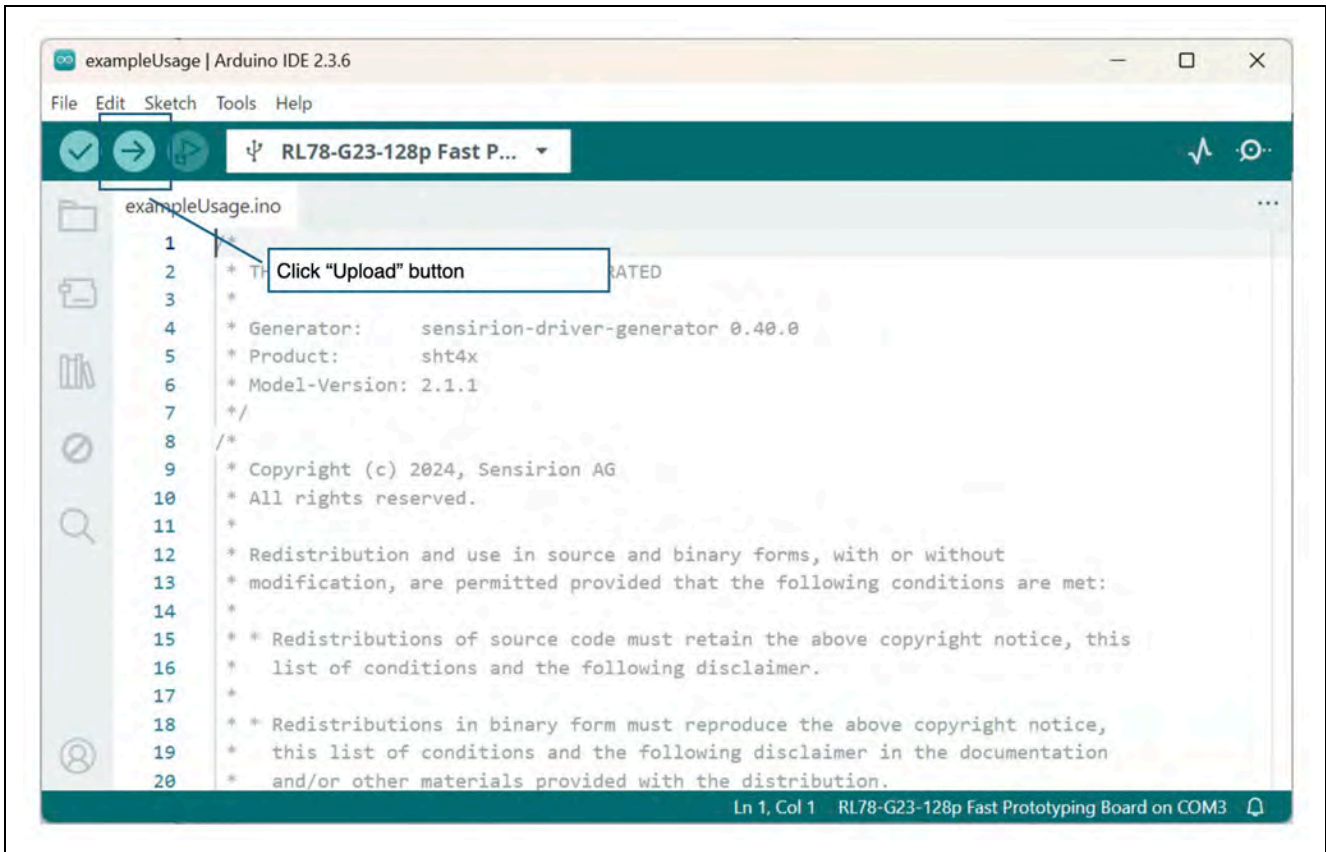


Figure 4-4 SCD40 Sketch Example

Once "exampleUsage.ino" is open, click the [Upload] button in the Arduino™ IDE's top toolbar to build the example sketch and upload it to the board.

Once the upload is complete, click the [Serial Monitor] button in the upper-right corner of the Arduino™ IDE to display the serial monitor at the bottom of the screen. The sensor's serial number will appear first, followed by the CO2 concentration, temperature, and humidity displayed every 5 seconds.

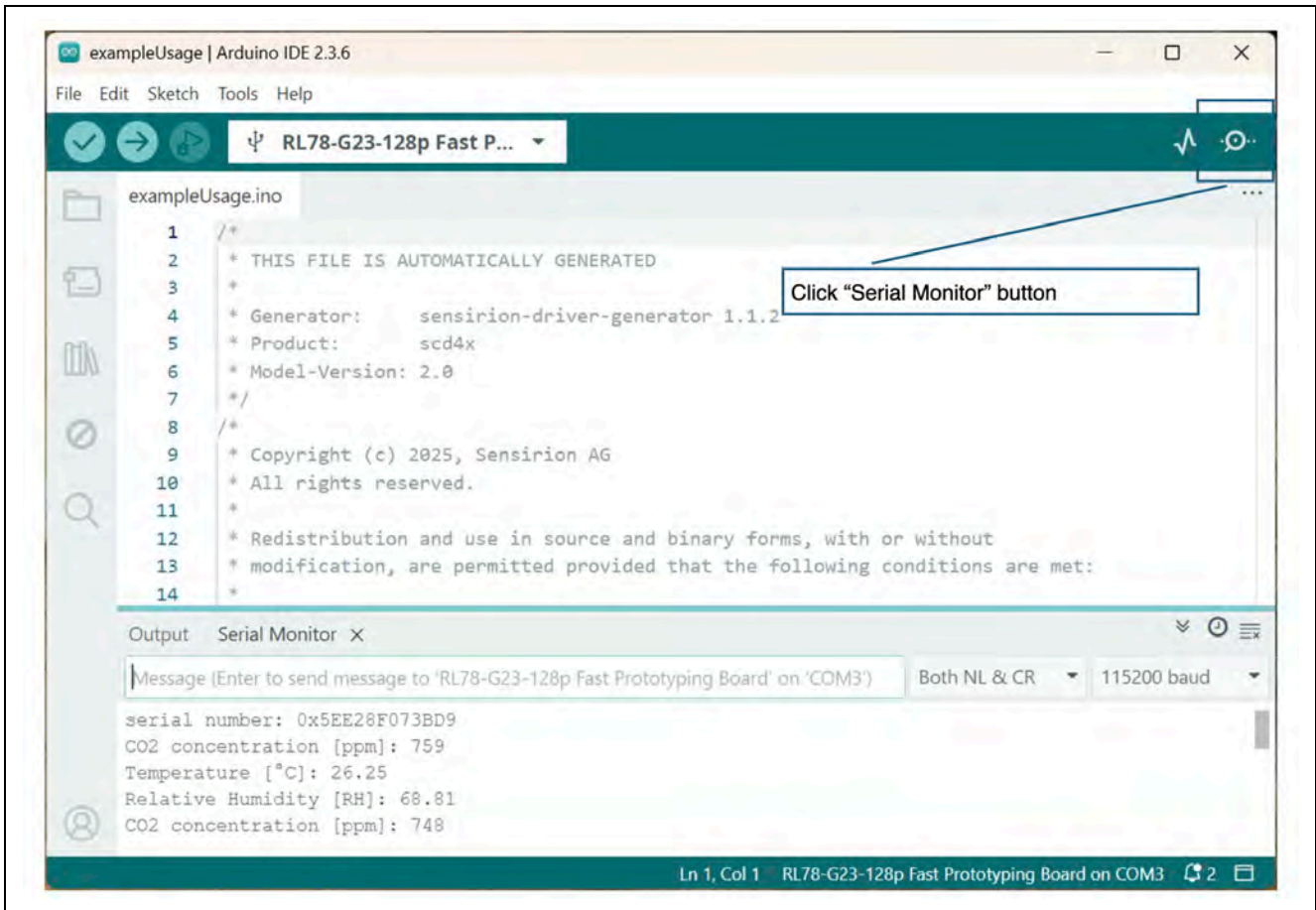


Figure 4-5 Verification of SCD40 Operation

4.2 Step 2: Sending Data to the IoT Cloud Service “Ambient”

In Step 2, the dummy data is sent to the IoT cloud service Ambient via the Wi-Fi network and displayed as a graph.

4.2.1 Installing the Ambient Library

To send data to the IoT cloud service Ambient, use the RL78/G23 Ambient library. Go to the [RL78/G23-128p FPB product page](#) and search for “Arduino” in the documentation or sample code to display the sample code. Download the sample code to an appropriate folder on the PC.

Select [Sketch] menu → [Include Library] → [Add.ZIP Library...] in the Arduino™ IDE, then open the downloaded .ZIP library file (RL78-G23-128p-Ambient.zip).

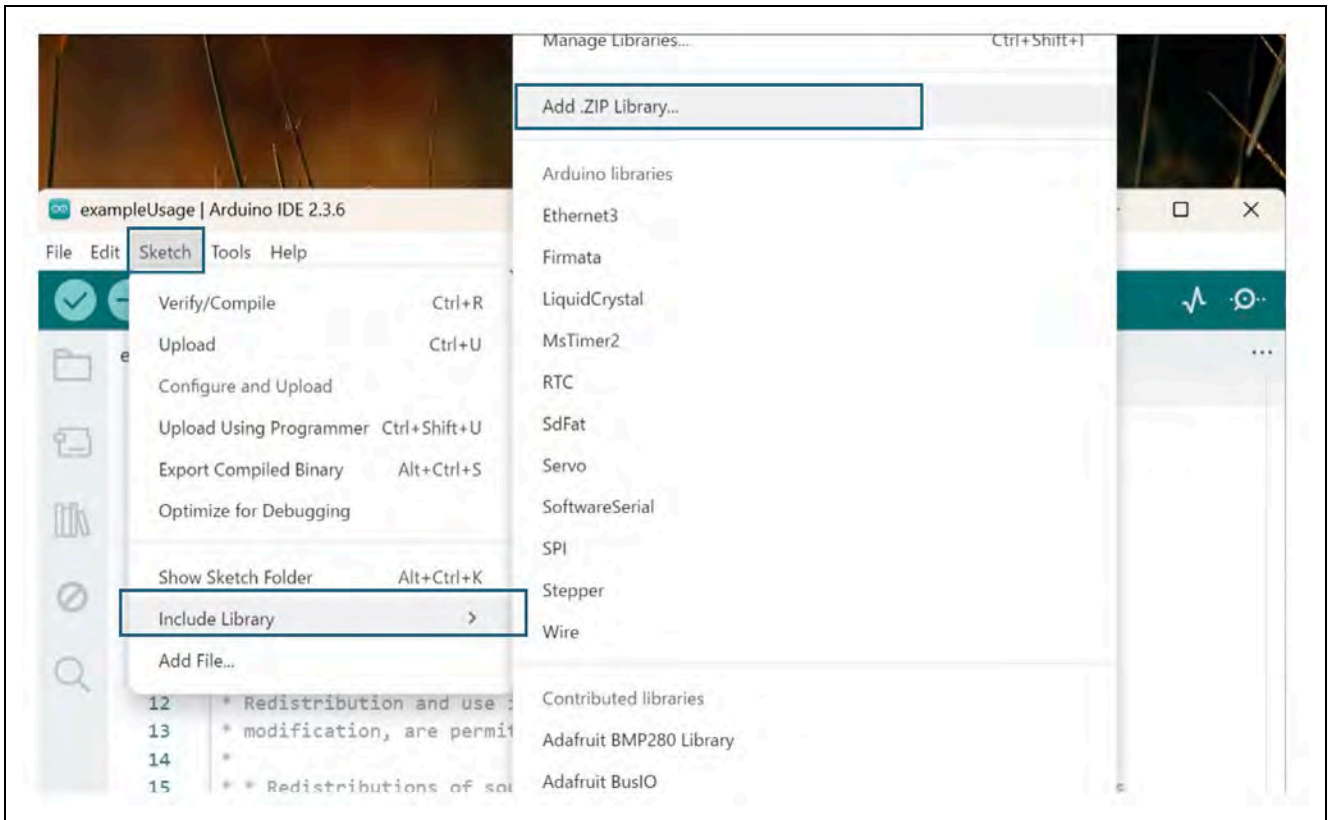


Figure 4-6 Installing the Ambient Library 1

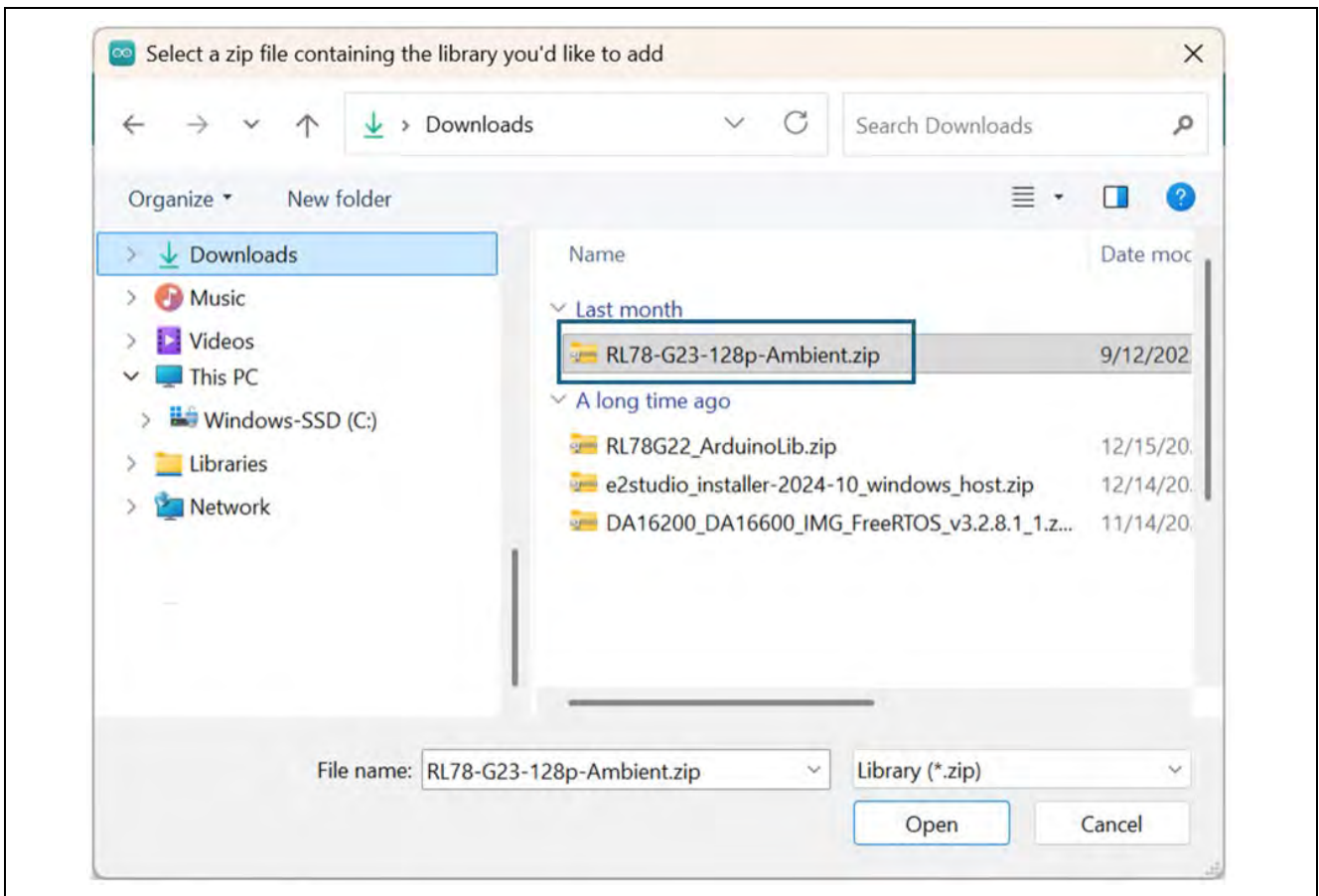


Figure 4-7 Installing the Ambient Library 2

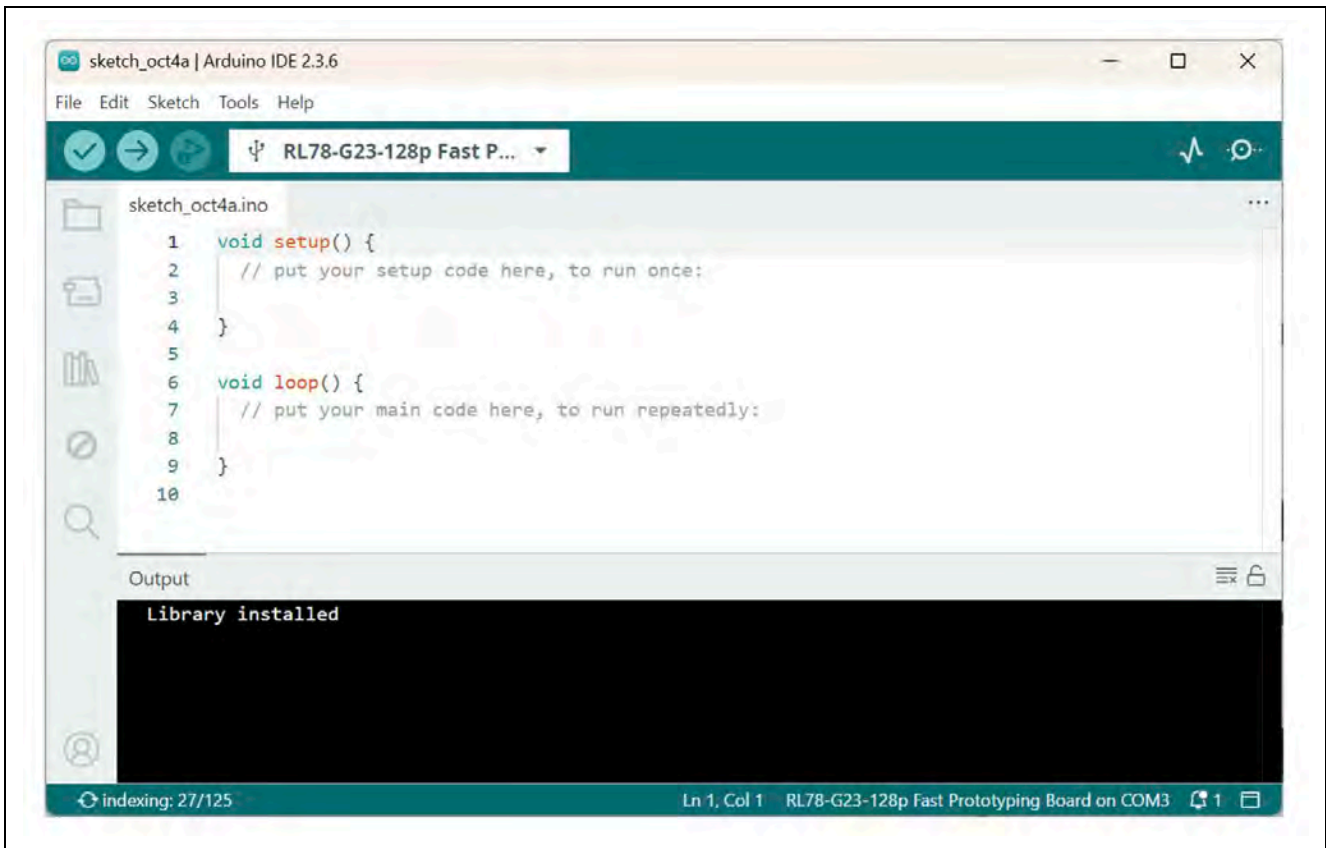


Figure 4-8 Installing the Ambient Library 3

When the Ambient library is installed, “Library installed” will be displayed.

4.2.2 Step 2 sample code overview

This sample code includes a program (AmbientTest) that sends dummy data from Step 2 to the Ambient cloud service, and a program (SCD40Ambient) that sends CO₂, temperature, and humidity data acquired from the CO₂/temperature/humidity sensor SCD40 in Step 3 to Ambient. The file structure is shown below.

```

RL78-G23-128p-Ambient
├─ examples
│   └─ AmbientTest
│       └─ AmbientTest.ino ← Sample Sketch
│       └─ SCD40Ambient
│           └─ SCD40Ambient.io ← Sample Sketch
├─ src
│   └─ Ambient.cpp ← Source file of Ambient library API function
│   └─ Ambient.h ← Header file of Ambient library
│   └─ DA16x00.cpp ← Source file of WiFi interface DA16x00 API function
│   └─ DA16x00.h ← Header file of WiFi interface DA16x00
│   └─ WiFi.cpp ← Source file of WiFi module API function
│       └─ WiFi.h ← Header file of WiFi module
├─ keywords.txt
└─ library.properties

```

Figure 4-9 File structure of the sample code

4.2.3 Ambient library API functions

The list of API functions implemented in the Ambient library is shown below.

API function name	Feature
bool begin(channelId, writeKey, readKey)	Ambient call preparation
bool set(field, data)	Sending data set
bool send()	Sending data to Ambient
bool read(buf, len, n)	Receiving data from Ambient
int status	Ambient call processing status

The function specifications for the Ambient library are shown below.

`bool begin(channelId, writeKey, readKey)`

Outline Ambient call preparation

Argument unsigned long channelId Channel ID

 const char *writeKey Write Key

 const char *readKey Read Key

Return Value Description Call preparation processing result

 Return Value true: Normal end

 false: Error

 Data type bool

`bool set(field, data)`

Outline Sending data set

Argument int field Field

 int data Data

Return Value Description Data set processing result

 Return Value true: Normal end

 false: Error

 Data type bool

`bool send()`

Outline Sending data to Ambient

Argument None

Return Value Description Data sending processing result

 Return Value true: Normal end

 false: Error

 Data type bool

bool read(buf, len, n)

Outline Receiving data from Ambient

Argument	char *buf	Pointer to buffer
	int len	Buffer size
	int n	Number of data items to read
Return Value	Description	Data receiving processing result
	Return Value	true: Normal end
		false: Error
	Data type	bool

int status

Outline Ambient call processing status

Return Value	Description	Ambient call processing status
	Return Value	HTTP status code
		200: Normal end
	Data type	int

4.2.4 Ambient user registration and channel creation

To use Ambient, you first need to register for a free account. Access the Ambient website (<https://ambidata.io>) in your browser and use your browser's translation feature to translate the page into English (from this point onward, translate each page using your browser's translation feature). On the Ambient website, click the "User registration (free)" button.

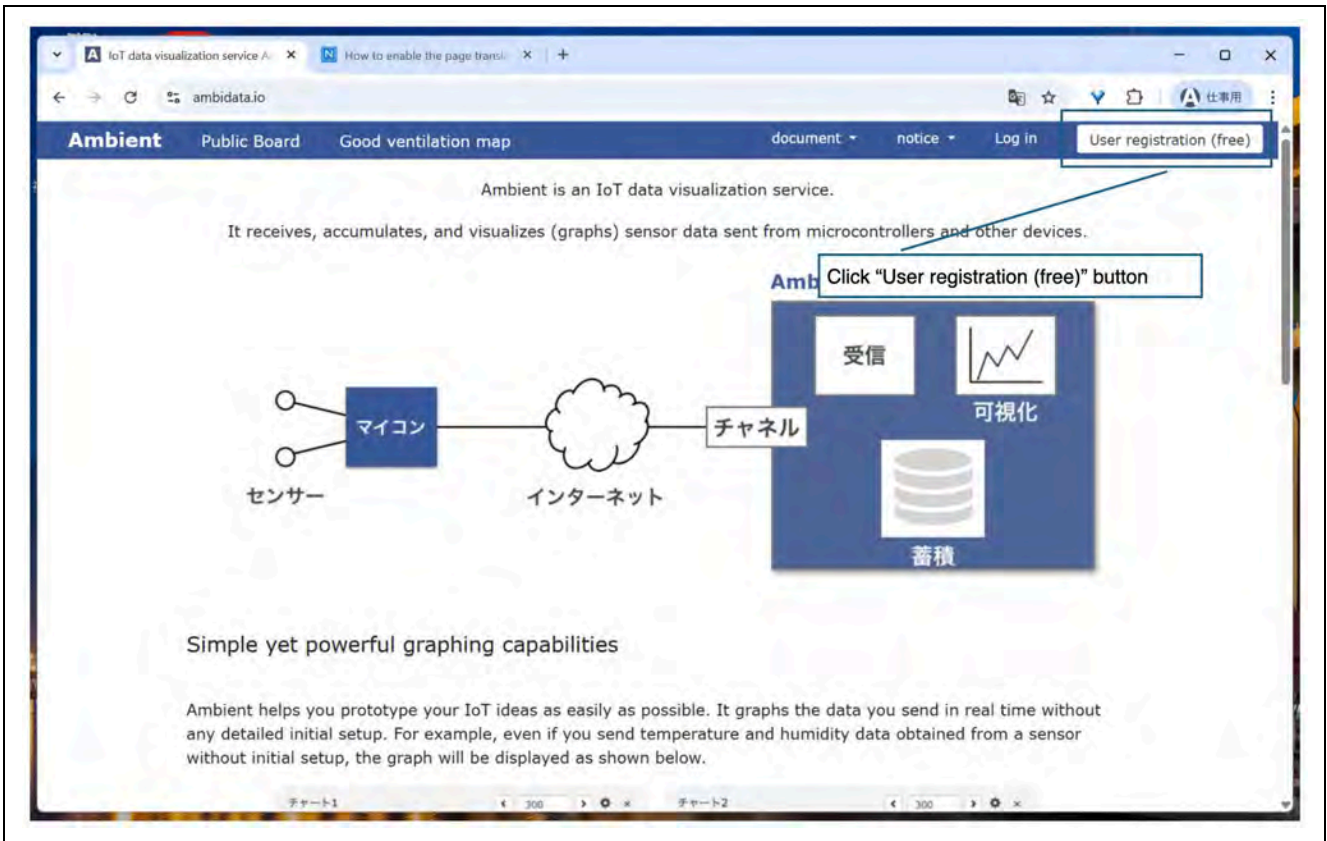


Figure 4-10 Ambient website

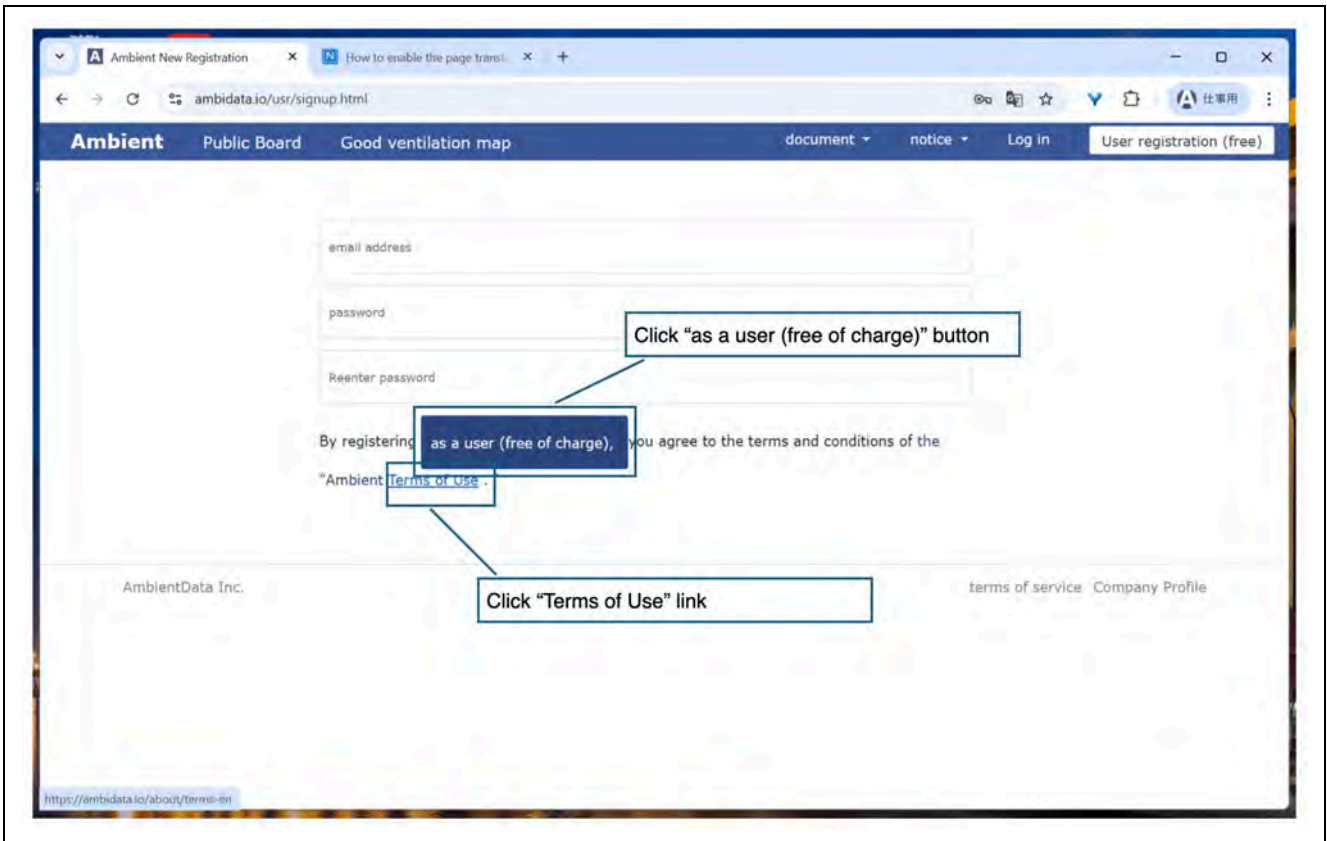


Figure 4-11 User registration page

Click the “Terms of Use” link to review the “Ambient Terms of Use (<http://ambidata.io/about/term-en/>)”. If you agree, enter your email address and password on the user registration page, then click the “as a user (free of charge)” button. A confirmation email will be sent to the entered email address. Click the URL in the email to complete your user registration.

In Ambient, data sent from the microcontroller is managed in units called “channels.” When sending data, you specify the ‘channel’ to send it to. After registering and logging into Ambient, the channel list page appears. Initially, there are no channels, so only the “Create a channel” button is present.

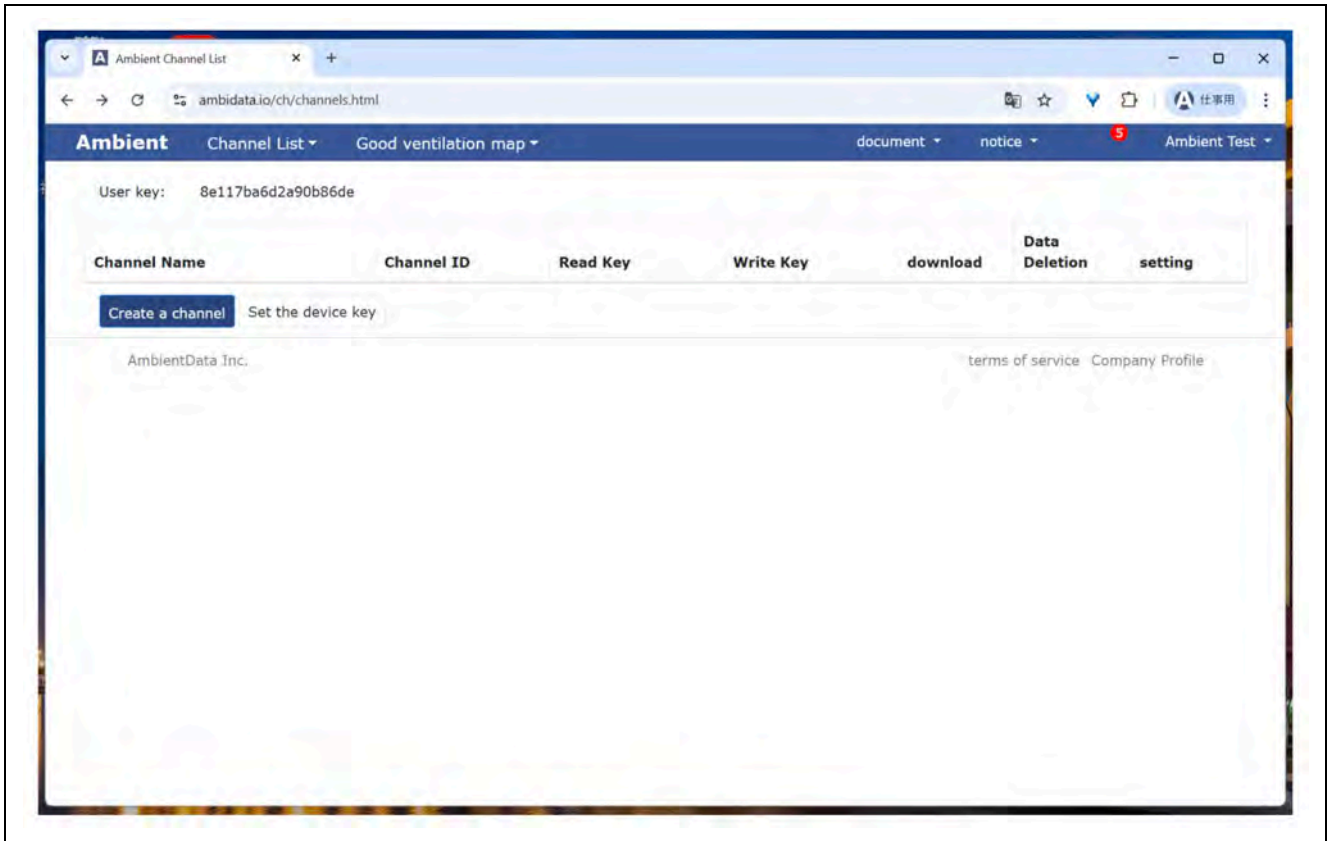


Figure 4-12 Channel list page

Clicking the “Create a channel” button creates a channel and assigns a channel ID, a read key required to receive data from this channel, and a write key required to send data. The read key and write key are strings composed of alphanumeric characters.

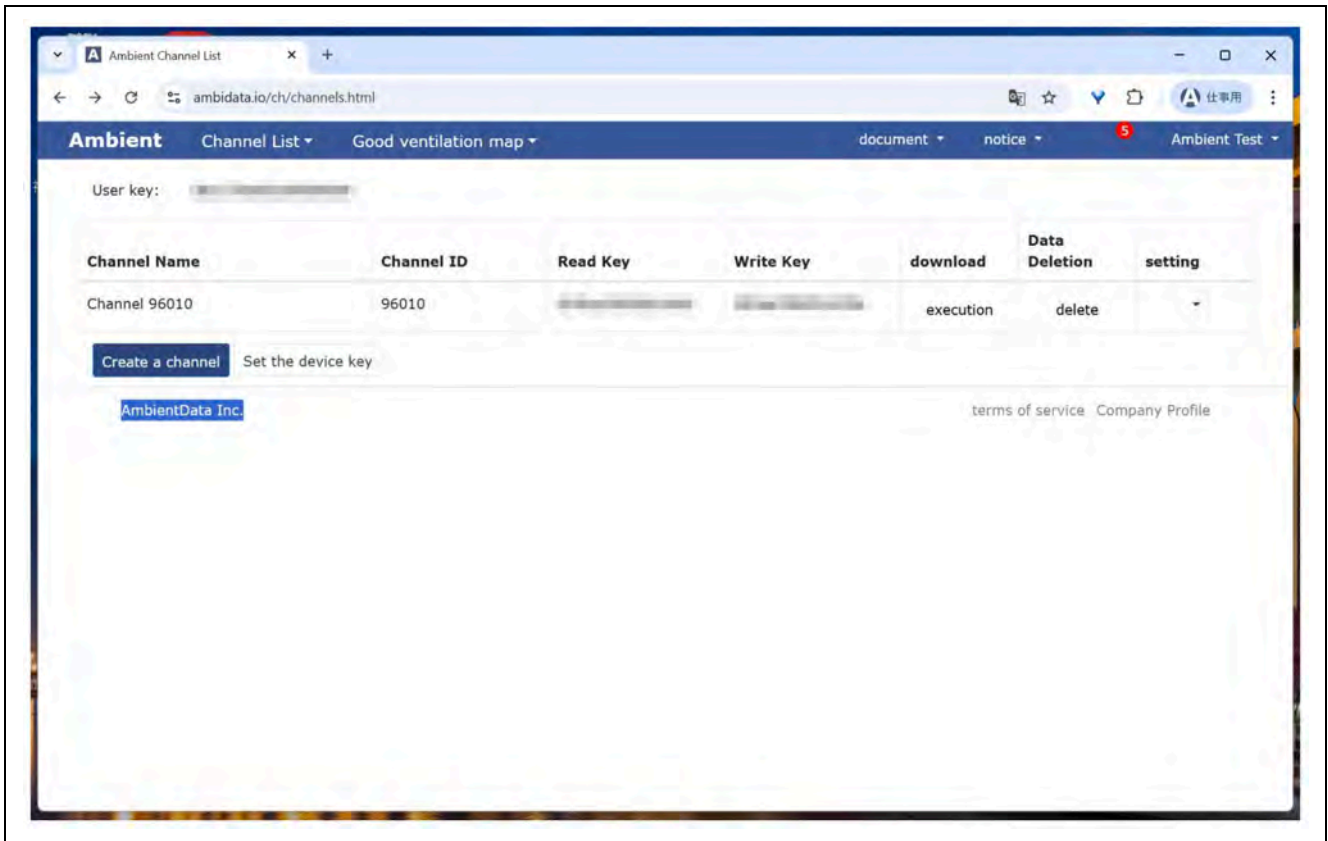


Figure 4-13 Channel creation

4.2.5 Step 2 Program overview

The program overview for Step 2 is shown below.

The program for Step 2 (AmbientTest.ino) consists of the `setup()` function, which runs once at the beginning, and the `loop()` function, which runs repeatedly.

In the `setup()` function, the `da16x00.begin()` function initializes the WiFi interface DA16x00. The `WiFi.begin()` function then sends a connection request to the WiFi network, specifying the SSID and password. It waits until the `WiFi.status()` function returns `WL_CONNECTED` (connection complete). Once the WiFi connection is established, it prepares to call Ambient using the `am.begin()` function. Note that the explanation of the Serial function, which outputs characters to the serial monitor, has been omitted.

```
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
}
```

The loop function uses a dummy variable named count. It sets the value of count as the first data and the square of count as the second data using the am.set() function. It then sends this data to the Ambient device using the am.send() function. After sending, it increments the value of count by 1. Since the minimum sending interval to the Ambient device is 5 seconds, it waits for 5 seconds before exiting the loop function.

```
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 Verifying Step 2 Operation

The procedure for verifying Step 2 operation is shown below.

Click [File] menu → [Examples] → [RL78 G23 128p Ambient] → [AmbientTest] in the Arduino™ IDE to open the sample sketch AmbientTest.ino.

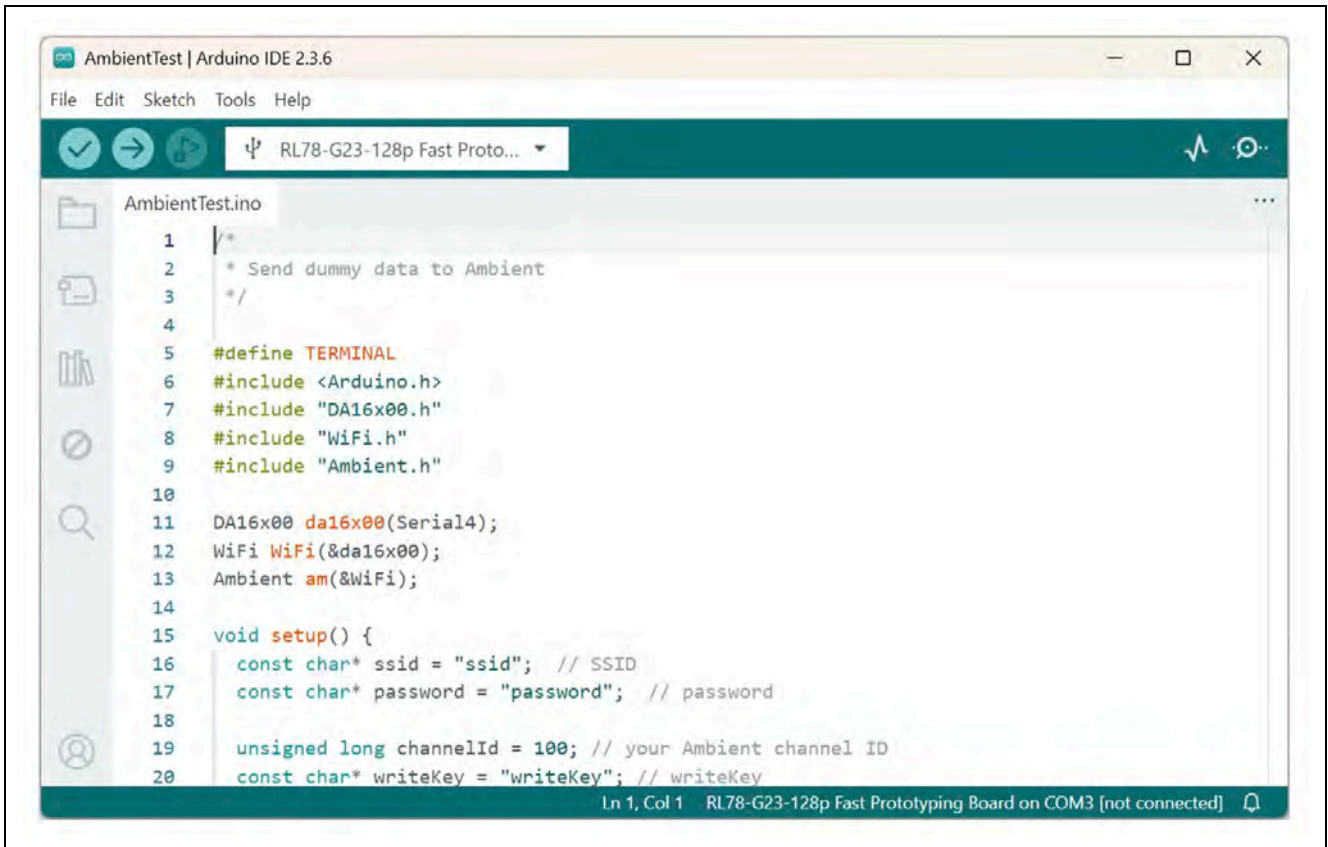


Figure 4-14 Step 2: Open the sample sketch

On line 16, "ssid" and on line 17, "password" should be replaced with the values for the WiFi router to be connected to. Since the DA16X00 communicates at 2.4GHz, select a 2.4GHz SSID. On line 19, channelId and on line 20, writeKey should be replaced with the channelId and writeKey of the channel created in "4.2.4 Ambient user registration and channel creation".

If "Select Board" appears in the Arduino™ IDE toolbar, click that section and select the port number to which the RL78/G23 128p board is connected. Enter 'rl78' in the search field of the "Select Other Board and Port" screen and "RL78-G23-128p Fast Prototyping Board" will appear; select it.

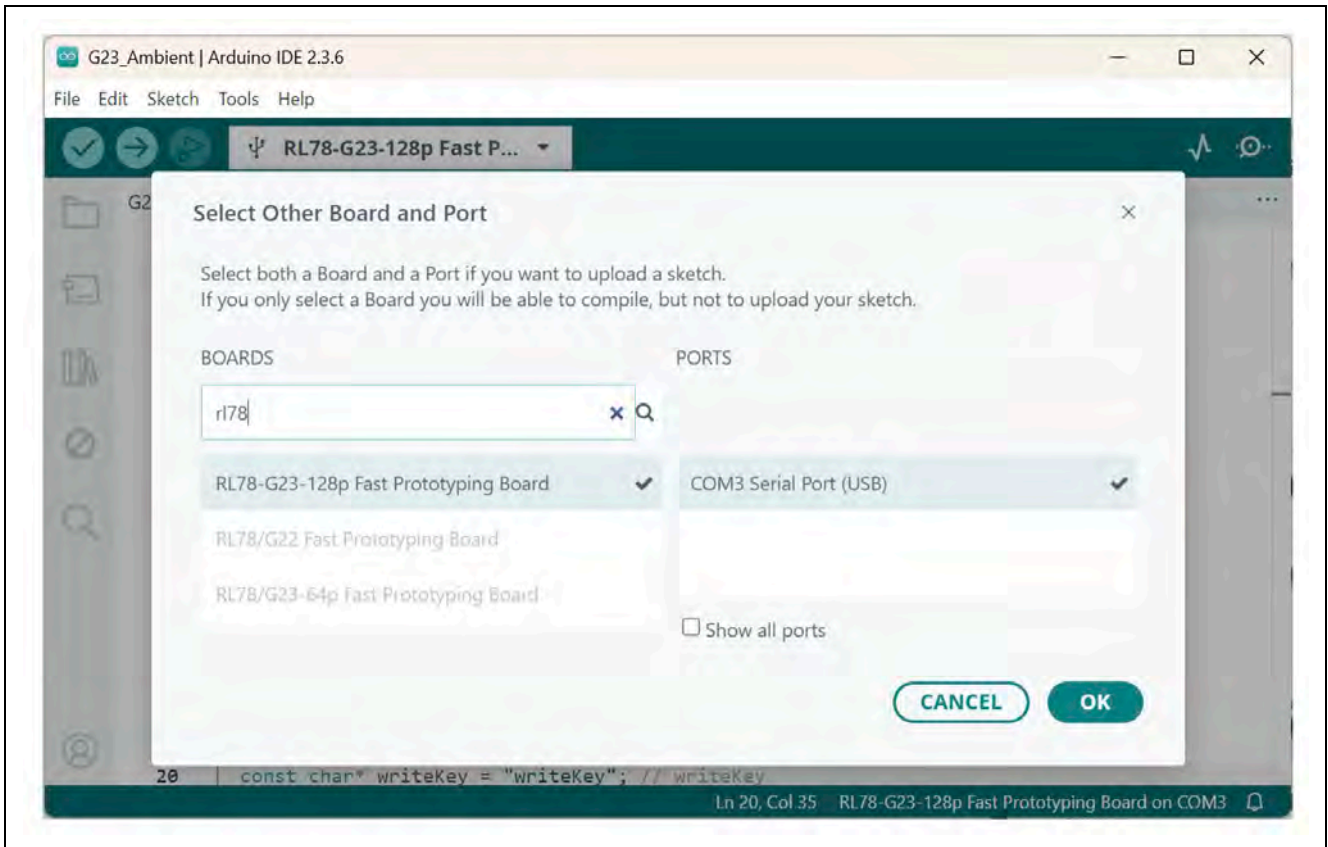


Figure 4-15 Port and Board Selection

Rewrite the SSID, password, channel ID, and write key, then select the port and board. Click the [Upload] button in the Arduino™ IDE to build the program and upload it to the board.

After writing the code, click the [Serial Monitor] button in the Arduino™ IDE to display the serial monitor. Verify that the program is running and dummy data is displayed every 5 seconds in the serial monitor, as shown in Figure 4-16.

Log in to the Ambient site in your browser, then click the channel from which you sent data on the Channel List screen. You can confirm that the data sent to Ambient is displayed graphically, as shown in Figure 4-17.

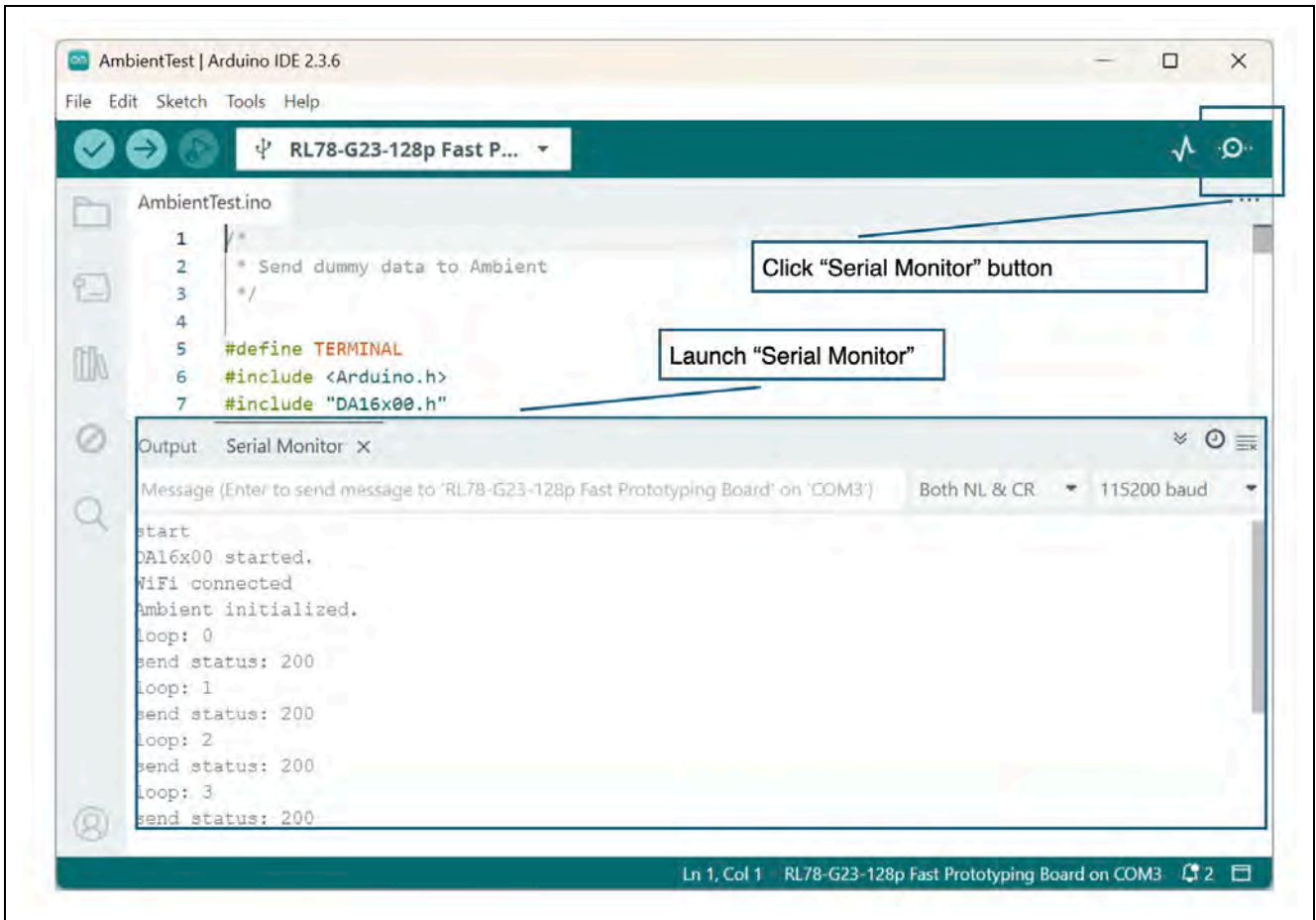


Figure 4-16 Starting the program in Step 2

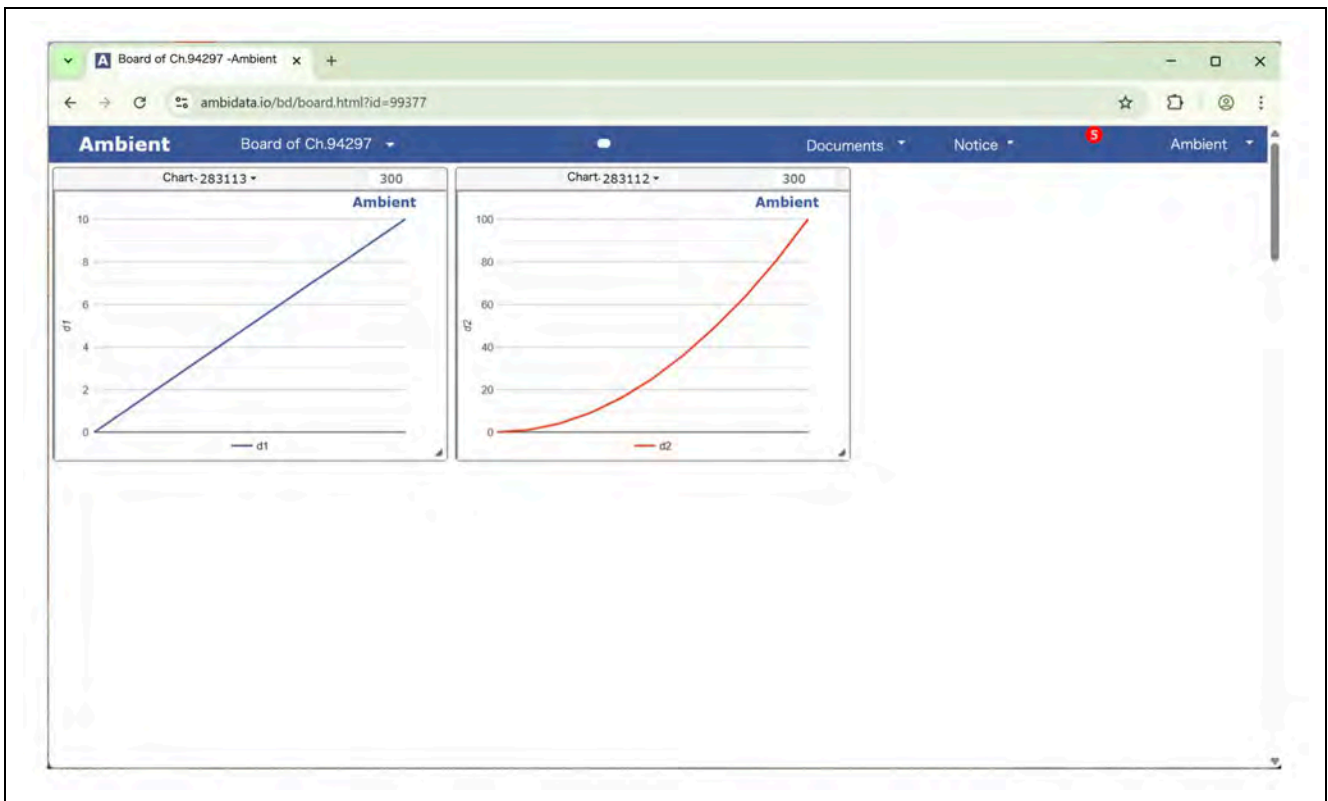


Figure 4-17 Data display in Ambient

4.3 Step 3: Sending sensor data to Ambient

In Step 3, the program from Step 1 and Step 2 is combined to acquire data from the CO2 and temperature/humidity sensor SCD40. This data is then sent to Ambient, where it is graphed for verification.

4.3.1 Verifying Step 3 Operation

The sketch (SCD40Ambient) that sends CO2, temperature, and humidity data acquired from the CO2/temperature/humidity sensor “SCD40” in Step 3 to Ambient is included in the Ambient library.

Click [File] menu → [Examples] → [RL78 G23 128p Ambient] → [SCD40Ambient] in the Arduino™ IDE to open the sample sketch SCD40Ambient.ino.

Once the sample sketch opens, rewrite the ssid, password, channelId, and writeKey to match your environment, just as in Step 2.

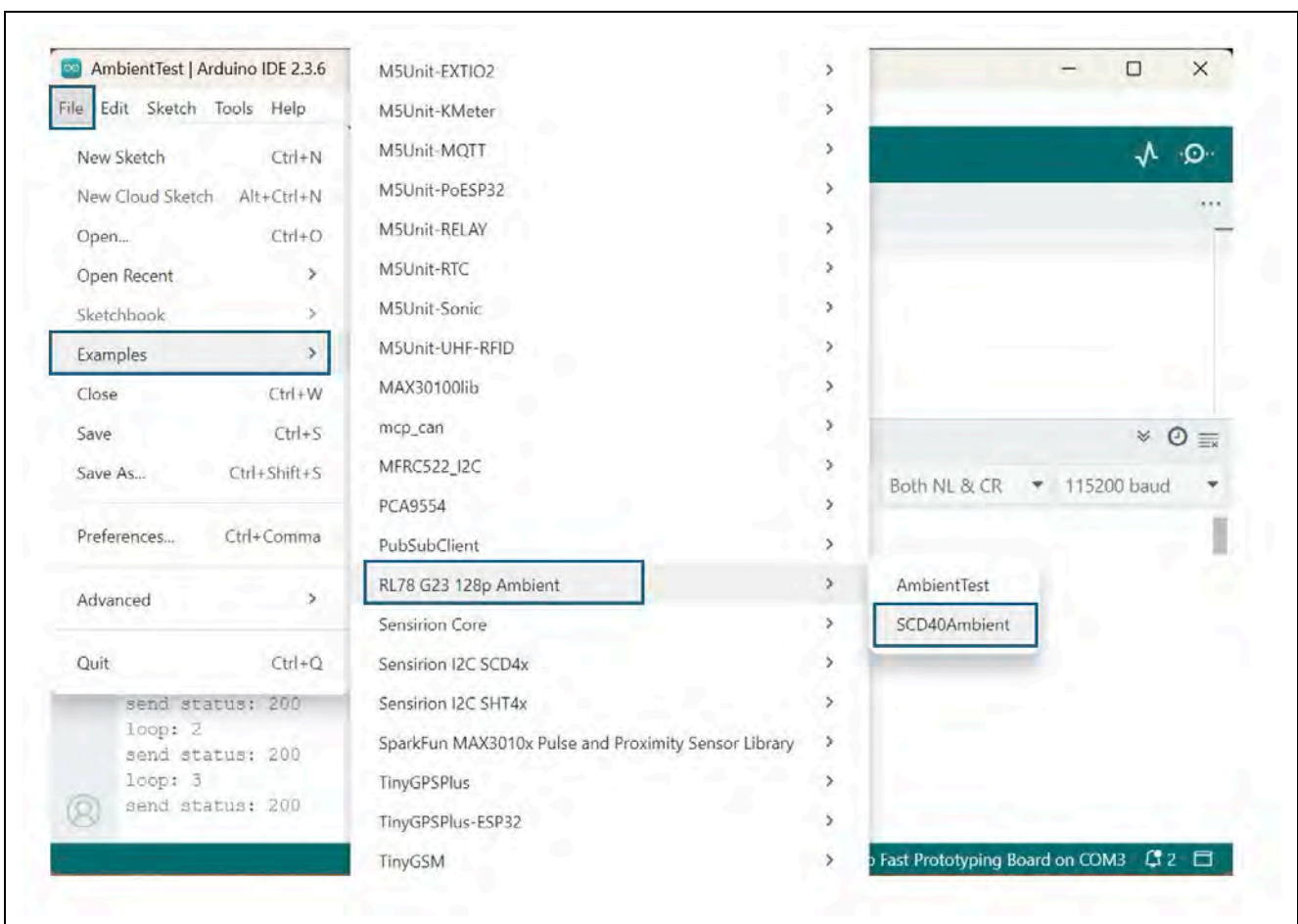


Figure 4-18 Open the sketch for Step 3

Click the [Upload] button in the Arduino™ IDE to build the program and upload it to the board. Once the upload completes, click the [Serial Monitor] button to display the serial monitor. As shown in Figure 4-19, the string “start” will appear followed by the SCD40 sensor’s serial number. Messages will then display indicating that the DA16x00 has started, connected to WiFi, and completed the initial setup for Ambient. Next, the CO2

concentration, temperature, and humidity obtained from the SCD40 sensor every 30 seconds will be displayed. If the results of sending to Ambient are successful, 200 will be displayed.

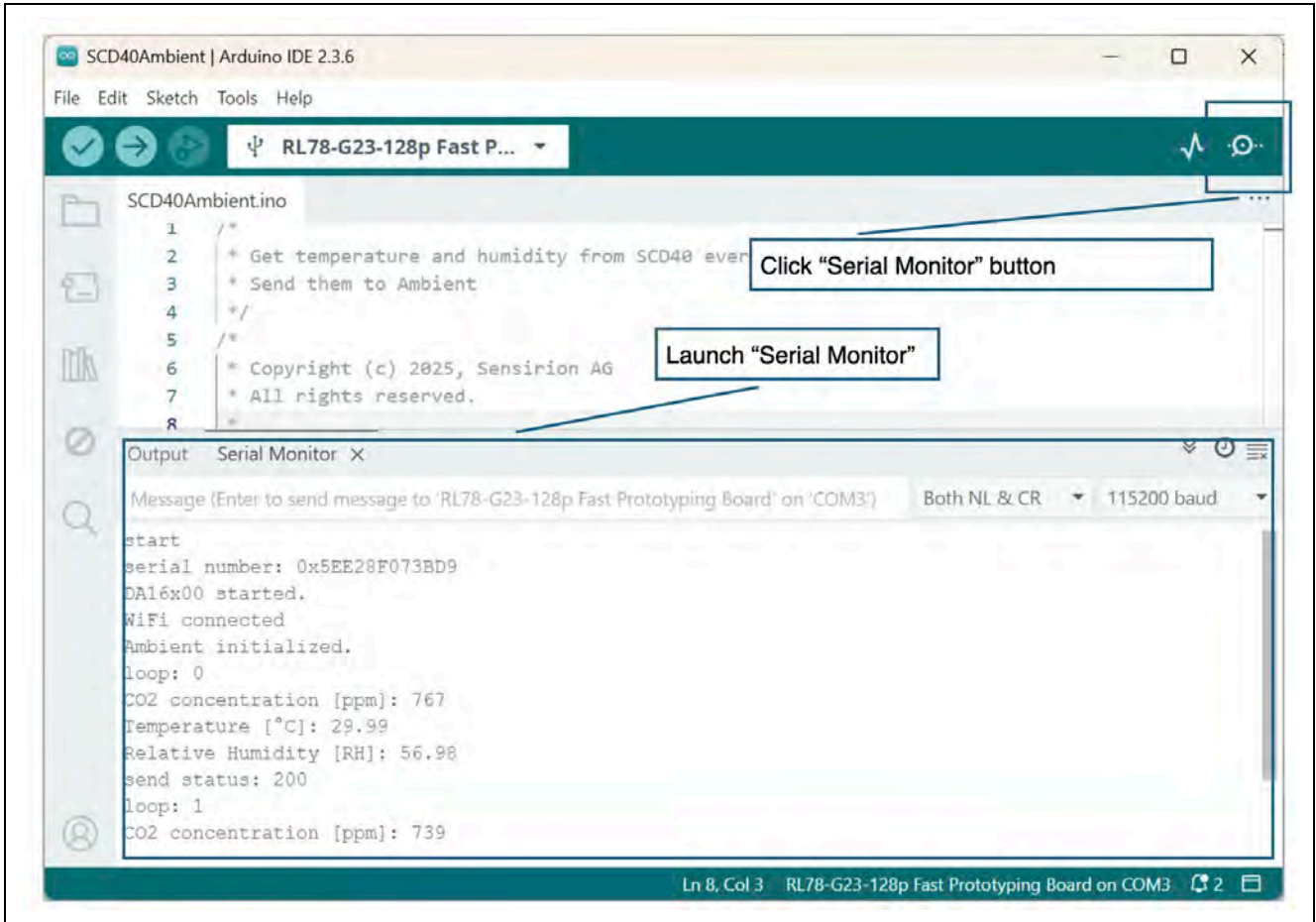


Figure 4-19 Program launch for Step 3

When viewing the channel screen displaying data sent from the Ambient site in your browser, you can confirm that temperature, humidity, and CO2 values are displayed as graphs.

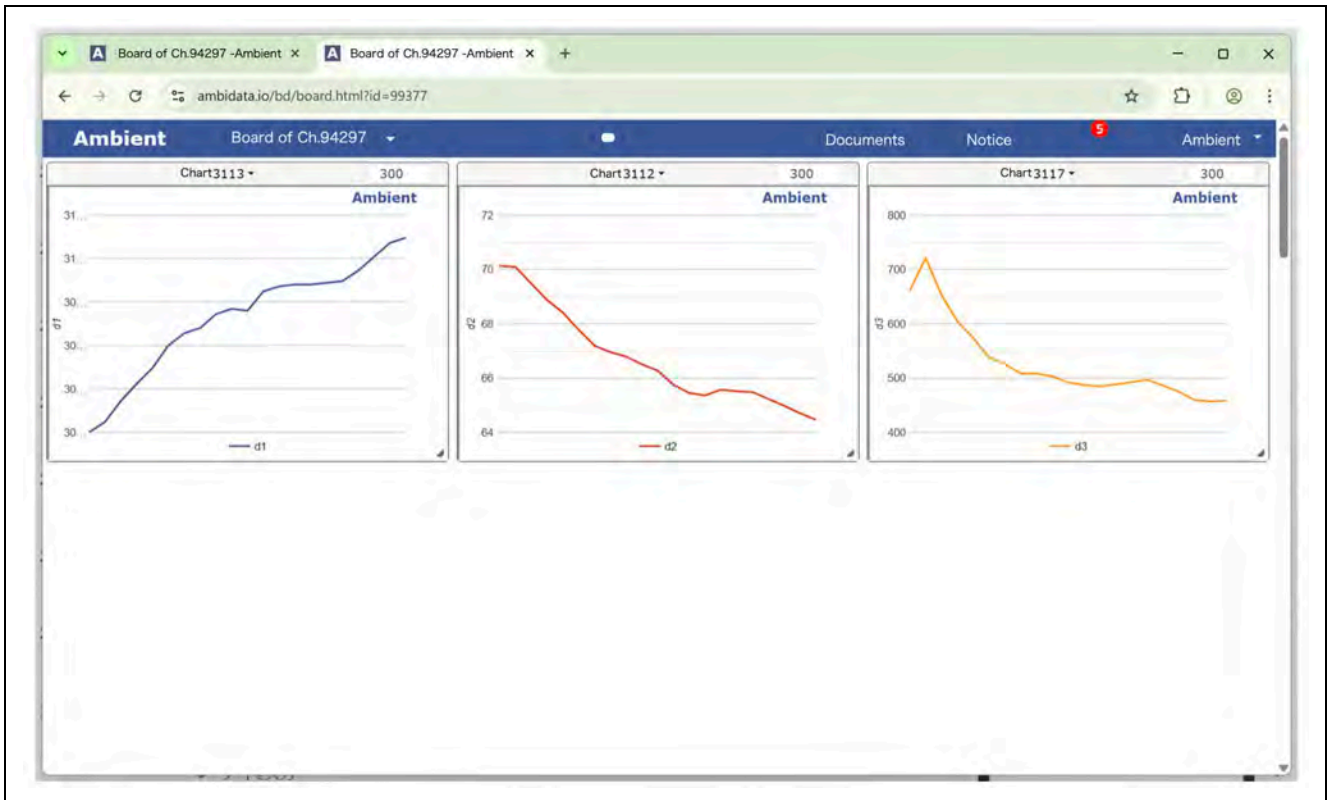


Figure 4-20 Data display in Ambient

5. Notes

5.1 Build does not start

When no board is selected (as indicated by the "Select Board" message in the Arduino™ IDE toolbar), pressing the [Upload] button will not start the build. Please select a board and port following the steps in section 3.3.4 before uploading.

5.2 Failed uploading

When [Upload] is performed while the Serial Monitor is running, the build will execute but then the upload will fail, displaying "Failed uploading: uploading error: exit status 4098" as shown in Figure 5-1. Close the Serial Monitor by clicking the "x" mark on the right side of its display before uploading.

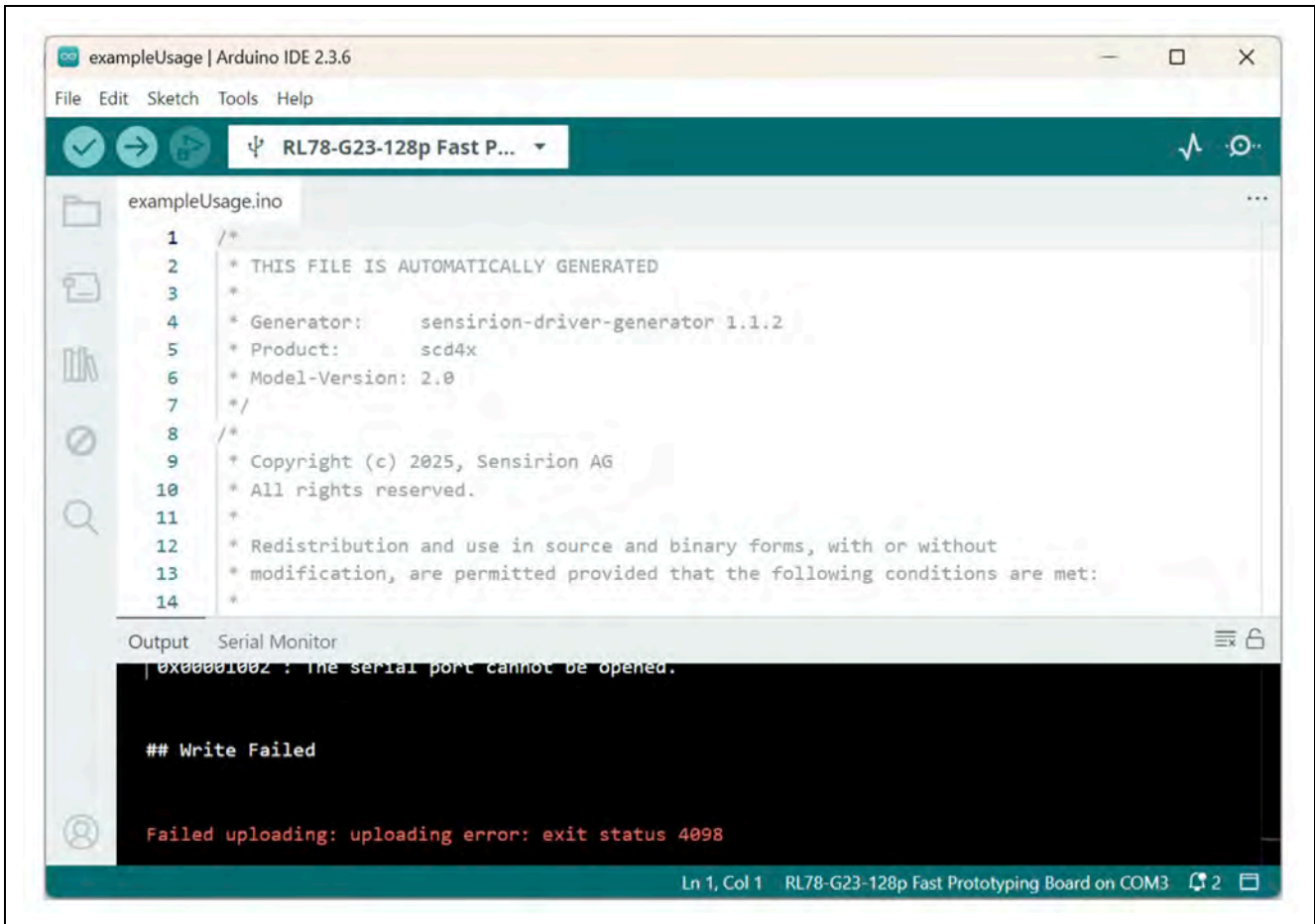


Figure 5-1 Upload errors occur while the serial monitor is running.

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Apr. 28. 26	-	First edition

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).

7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.

"Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.

"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:

www.renesas.com/contact/.