

GR-ROSE, HS3001

温湿度の取得、 Azure IoT Central接続

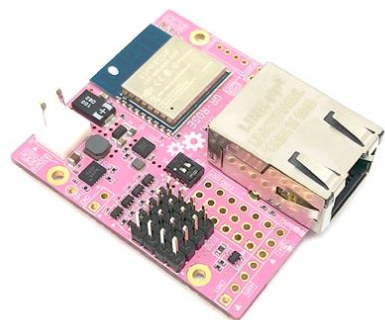
2021年9月11日 (REV. 1)

GADGET RENESAS



準備 ハードウェア

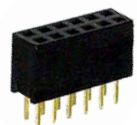
ルネサスからの提供物



GR-ROSE



PMOD HS3001
(温湿度)



6 x 2 ピンソケット

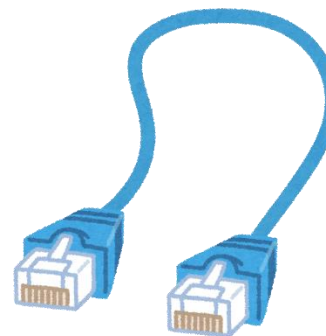


PMODケーブル

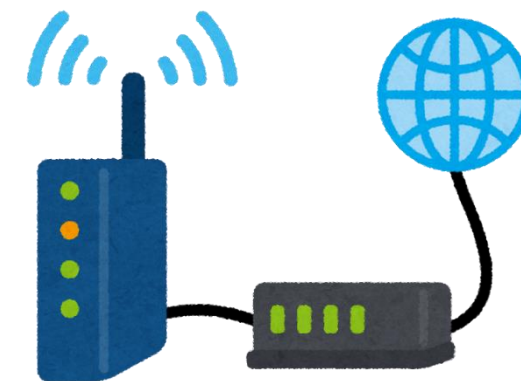
参加者が準備



USB micro-B

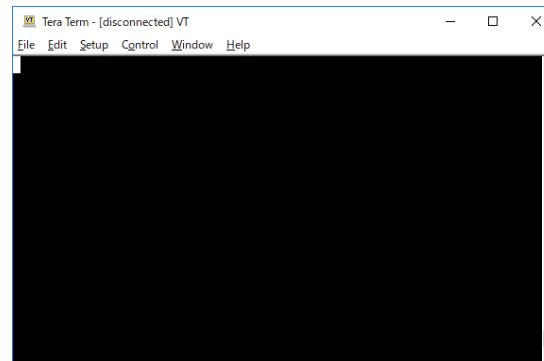


LANケーブル



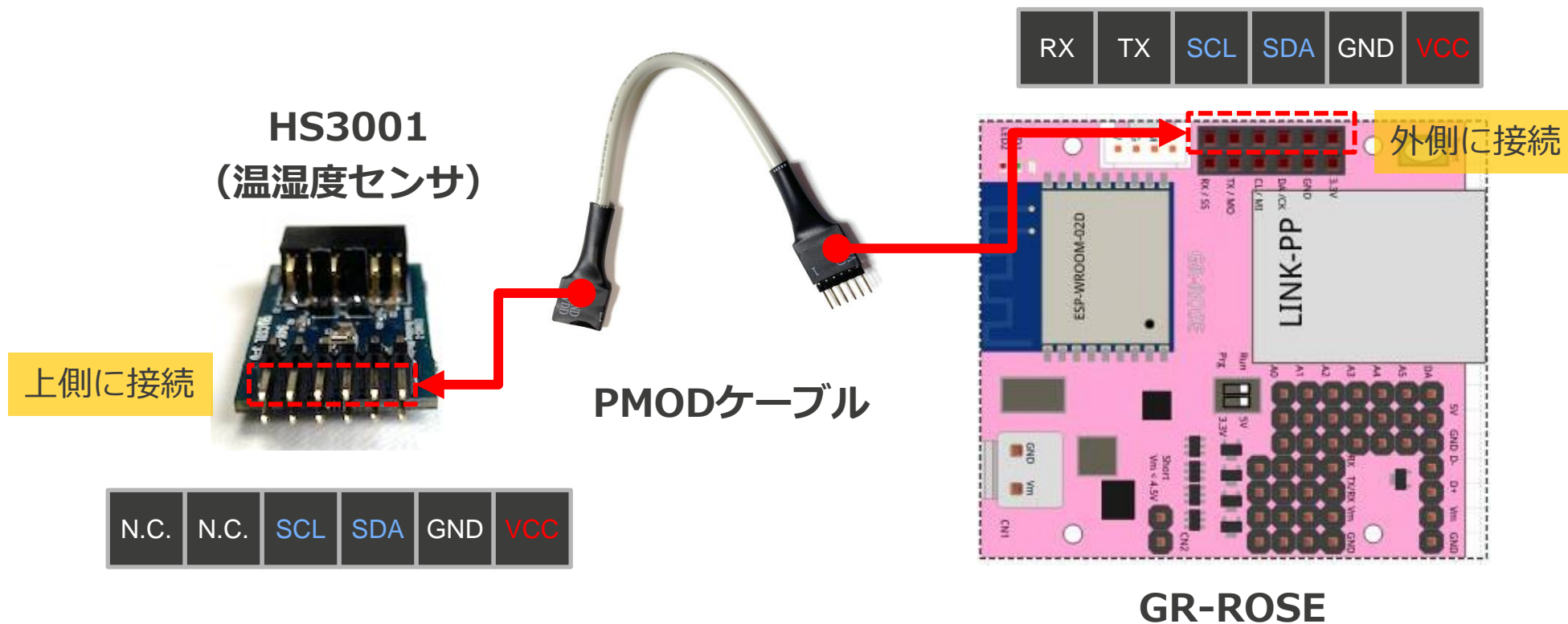
準備 ソフトウェア

シリアルモニタ (例 : Tera Term)



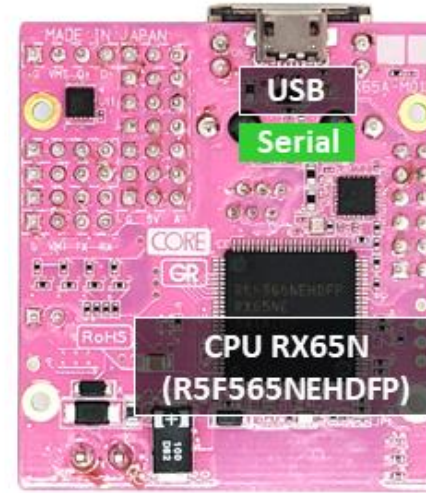
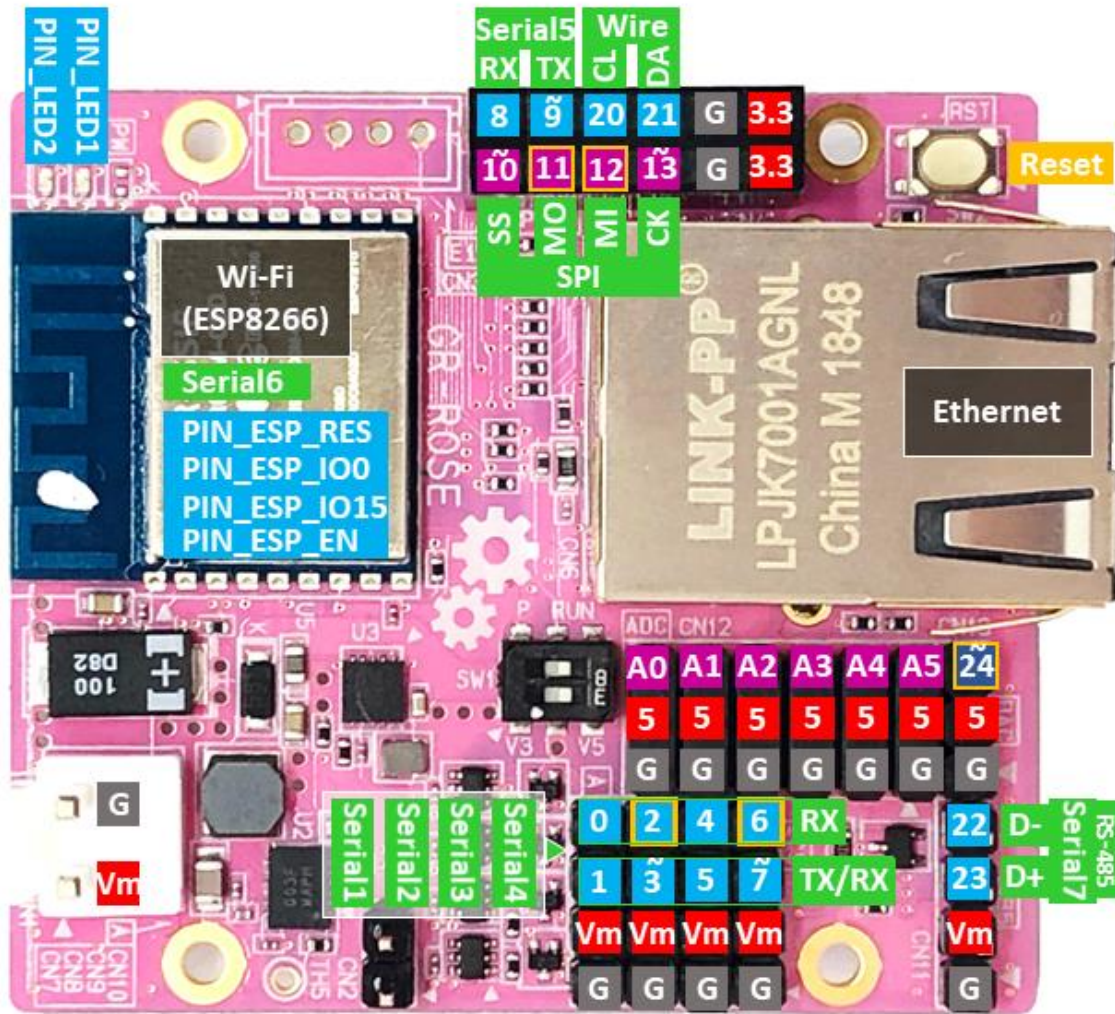
接続 GR-ROSE, HS3001

- 「GR-ROSE」と「HS3001」を接続します。信号の順序が合うように接続してください。



GR-ROSE Pin Map

Rev1.2



Main Clock	: 120MHz (12MHz x 10)
Sub Clock	: 32.768kHz
ROM	: 2MB(*1)
RAM	: 640KB
EEPROM	: 32KB

*1: 960KB available when writing a binary with USB storage

- D** Digital pin **D~** PWM by analogWrite **A** Analog & Digital pin
- A~** D/A output by analogWrite, specified 0 to 0xFFFF(3.3V).
- F** Communication pin **I** External Interrupt pin
- Vm** 4.5V to 18V power supply for motors.
- V** Supply voltage at described value **G** GND pin

Notes: In case of using motors 3V range, set SW1 to V3 and short CN2. Then Vm and 5 in red can be supplied from external like LiPo.



動作確認

WEBコンパイラ



WEBコンパイラ がじえるね検索

- 「がじえるね」でGoogle検索し、「Gadget Renesas」のWebサイトを開きます。

Google Translate × がじえるね - Google Search × +

← → ↻ google.co.jp/search?q=がじえるね&sxsrf=AOaemvLYykk4yR1d2H2EdmbYsWqwZaNvhA%3A1631289063825&source=hp&ei=...

アプリ がじえ Google Rulz Colabo Arduino TW FB Portal Outlook SCTC RX PMM TS

Google がじえるね 「がじえるね」でGoogle検索 × 🔊 🔍

🔍 All 📍 Maps 📺 Videos 📰 News 🛒 Shopping ⋮ More Tools

About 170,000,000 results (0.57 seconds)

<https://www.renesas.com/products> Translate this page

Gadget Renesas 「Gadget Renesas」を開く

がじえっとるねさす第一弾の投色のボード。32ビットのマイコンRX63Nを搭載。Arduino UNOに互換性があります。IDE for ...

<http://tool-cloud.renesas.com/atelier> Translate this page

がじえっとるねさす | ルネサス エレクトロニクス - Renesas ...

『がじえるね工房』では、ルネサスからお願いした“ルネサスMVP”による作品記事を一拳公開しています！苦労点や、工夫点、センサの制御、モーター制御などに関する情報など、読者に...



WEBコンパイラ ログイン

- 「WEBコンパイラ・ログイン」を押します。

RENESAS
BIG IDEAS FOR EVERY SPACE

Smart 検索

製品情報 アプリケーション デザイン/サポート ご購入/サンプル請求 会社情報

デザイン/サポート > Gadget Renesas Maker Resources

Gadget Renesas

WEBコンパイラ・ログイン WEBコンパイラ・ゲストログイン 新規登録 (MY RENESAS)

「WEBコンパイラ・ログイン」を押す

Gadget Renesas Connecting Ideas and Electronics

アイデアを高速にプロトタイプできるアイテムの提供、コミュニティの場作りや商品化支援等のイベントを通じて「創りたい！から造れるへ」夢ある楽しいものづくりを支援いたします。

Login

Email address *

yuuki.okamiya.yn@renesas.com

Password *

.....

[Forgot password?](#)

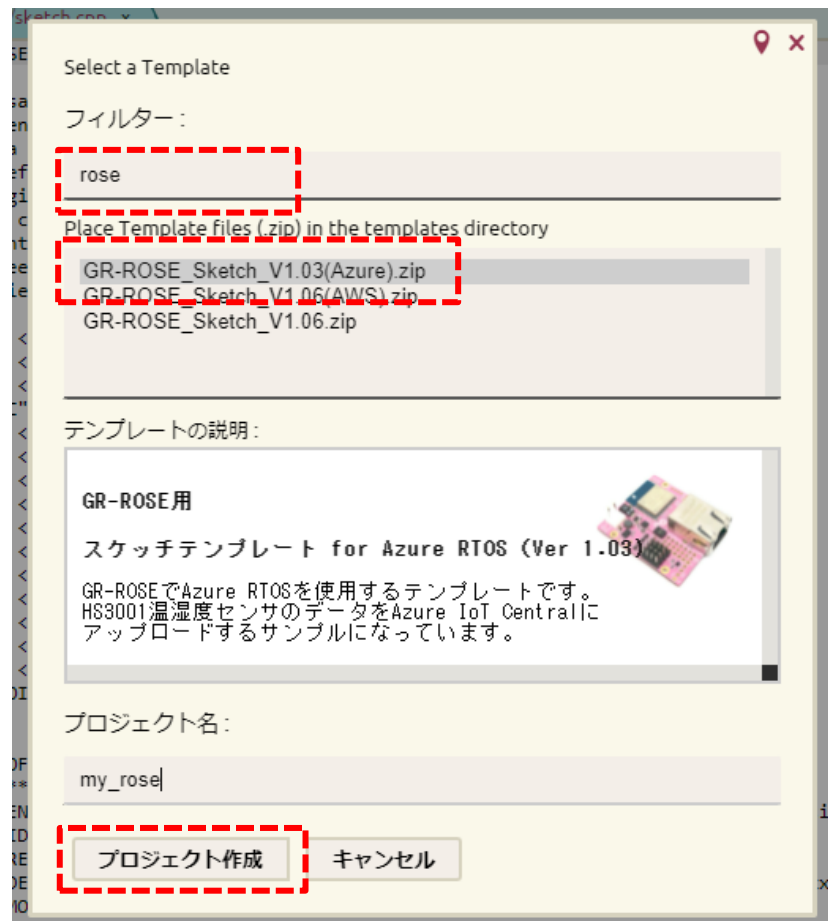
LOG IN or [Register](#)

MyRenesasのアカウント入力し、
「LOG IN」します。



WEBコンパイラ 新規プロジェクト

- 新規プロジェクトを作ります。「GR-ROSE_Sketch_V1.xx(Azure).zip」を選択して作成ボタンを押します。



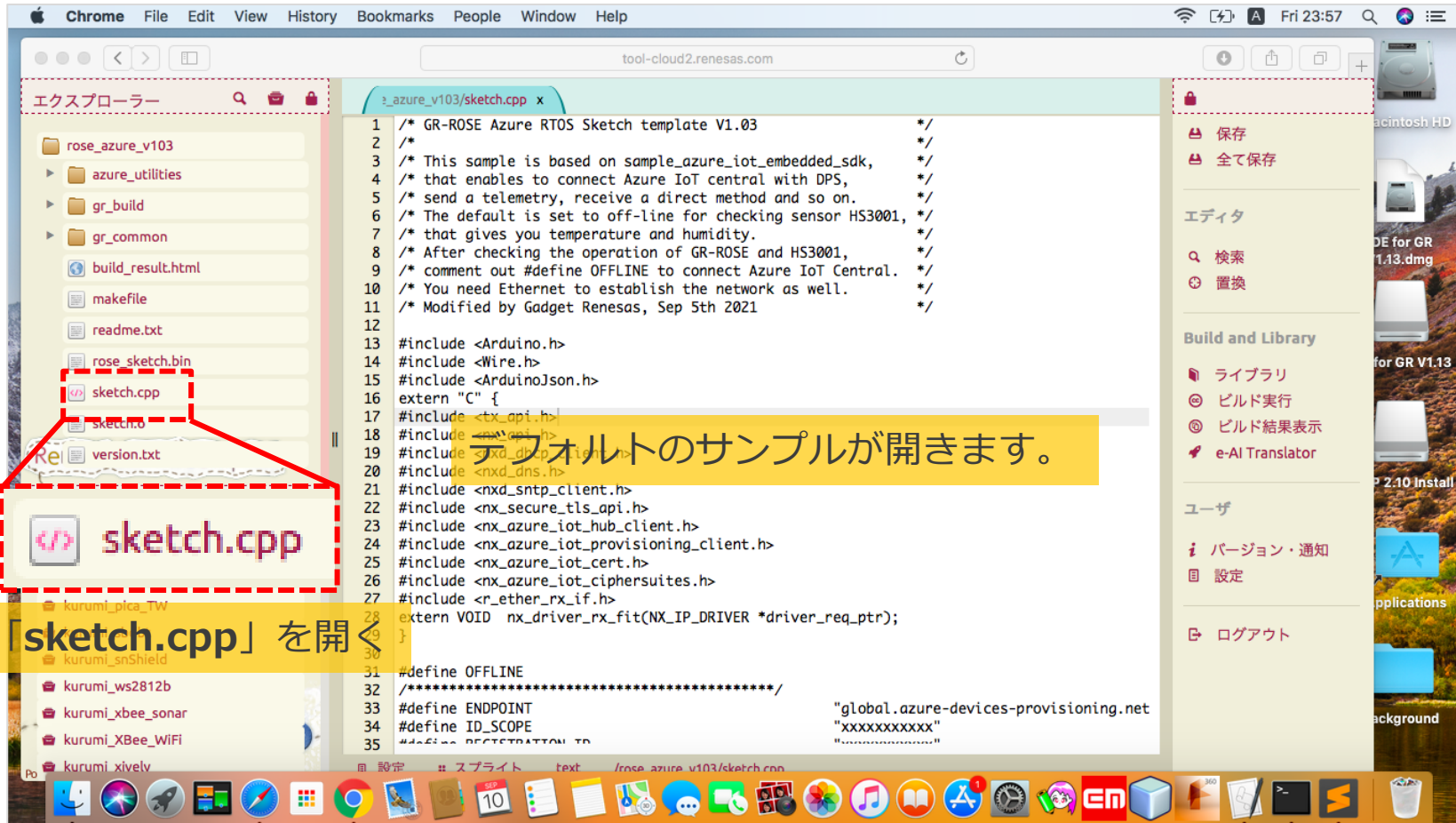
「フィルター」に「rose」と入力

「GR-ROSE_Sketch_V1.xx(Azure).zip」を選択

「プロジェクト作成」を押す

WEBコンパイラ プログラムの表示

- 「**sketch.cpp**」をダブルクリックすると、サンプルプログラムが開きます。



WEBコンパイラ ビルド、ダウンロード

- 「ビルド実行」を押すと、「rose_sketch.bin」が生成されます。

ビルド後に生成された「rose_sketch.bin」を「ダウンロード」をする

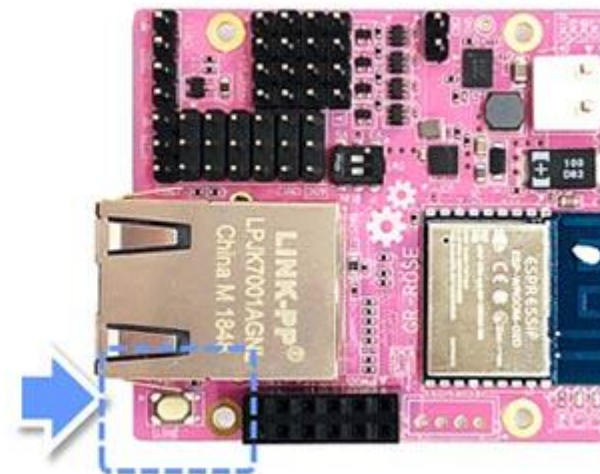
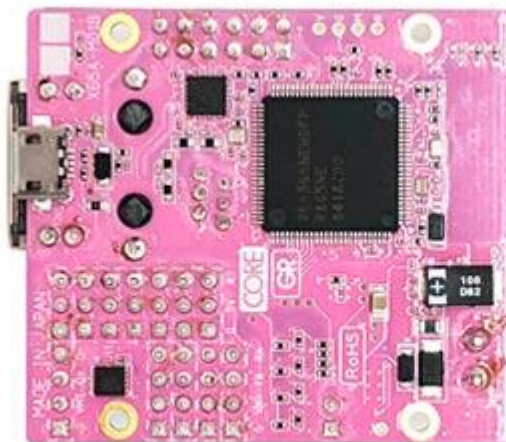
「ビルド実行」を押す

プログラム書き込み USB接続、リセットボタン

- GR-ROSEにUSBケーブルを接続し、リセットボタンを押します。GR-ROSEがUSBメモリとして認識されます。



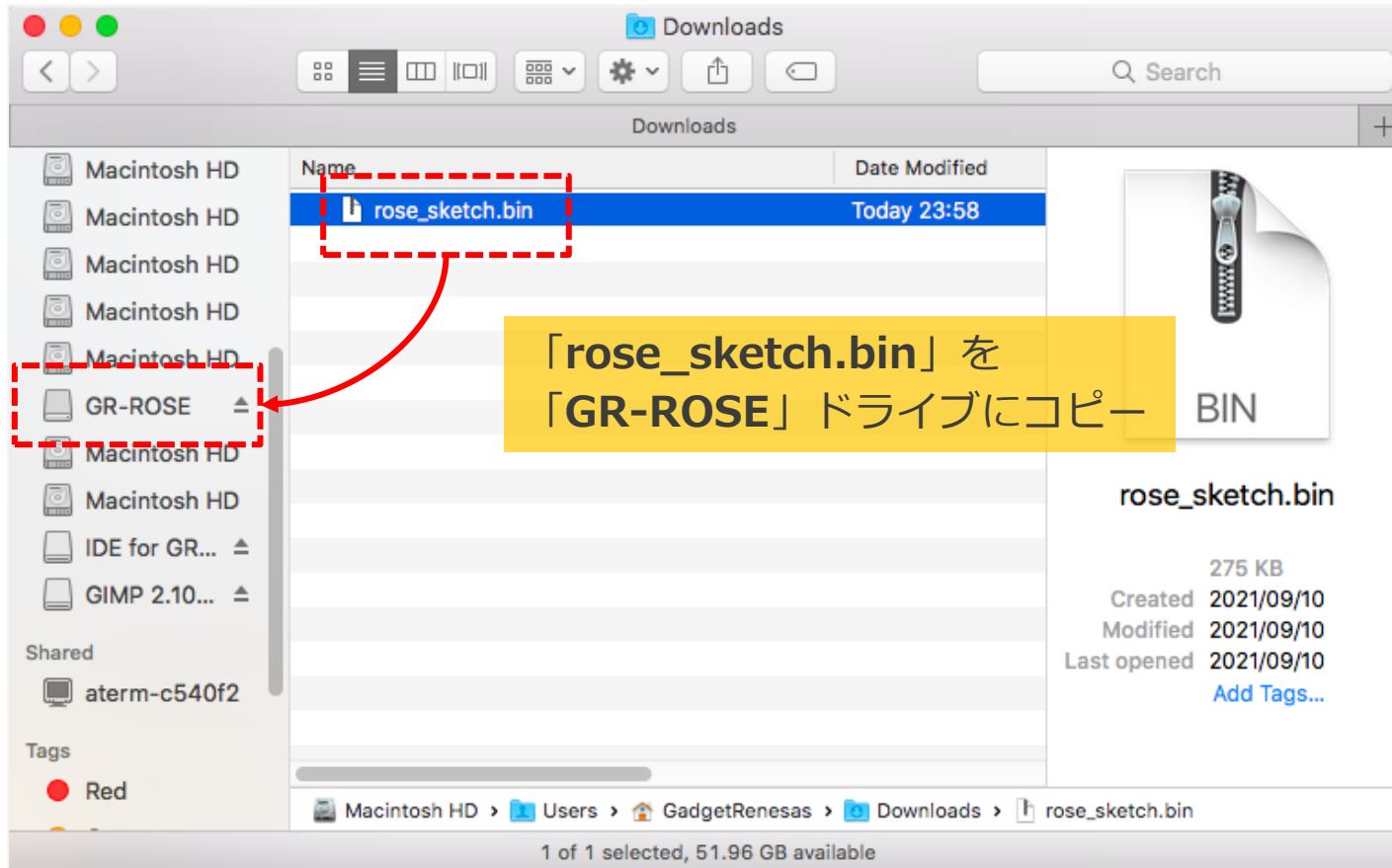
USBケーブルを接続



リセットボタンを押す

プログラム書き込み BINファイルのコピー

- 「rose_sketch.bin」を「GR-ROSE」にコピーします。LEDが光った後にプログラムが実行されます。





動作確認 シリアルモニタでの確認

- Tera Termなどのモニタを開きUSBポートを開きます。温湿度が表示されることを確認します。

```
Tera Term - [disconnected] VT
File Edit Setup Control Window Help
["Temperature":30.83, "Humidity":63.89]
["Temperature":30.81, "Humidity":63.86]
["Temperature":30.79, "Humidity":63.79]
["Temperature":30.78, "Humidity":63.77]
["Temperature":30.81, "Humidity":63.86]
["Temperature":30.82, "Humidity":63.88]
["Temperature":30.79, "Humidity":63.79]
["Temperature":30.75, "Humidity":63.71]
["Temperature":30.75, "Humidity":63.71]
["Temperature":30.74, "Humidity":63.69]
["Temperature":30.26, "Humidity":64.10]
["Temperature":30.82, "Humidity":63.88]
["Temperature":30.77, "Humidity":63.76]
["Temperature":30.76, "Humidity":63.74]
["Temperature":30.76, "Humidity":63.72]
["Temperature":30.79, "Humidity":63.80]
["Temperature":30.26, "Humidity":64.09]
```

シリアルモニタを表示（画面例はTera Term）。
温湿度が表示される。

デフォルトサンプルはオフラインでの確認用。
Azure IoT Centralにテレメトリとして送信されるデータが、
JSON形式で正しく作成されているか確認できる。

動作確認

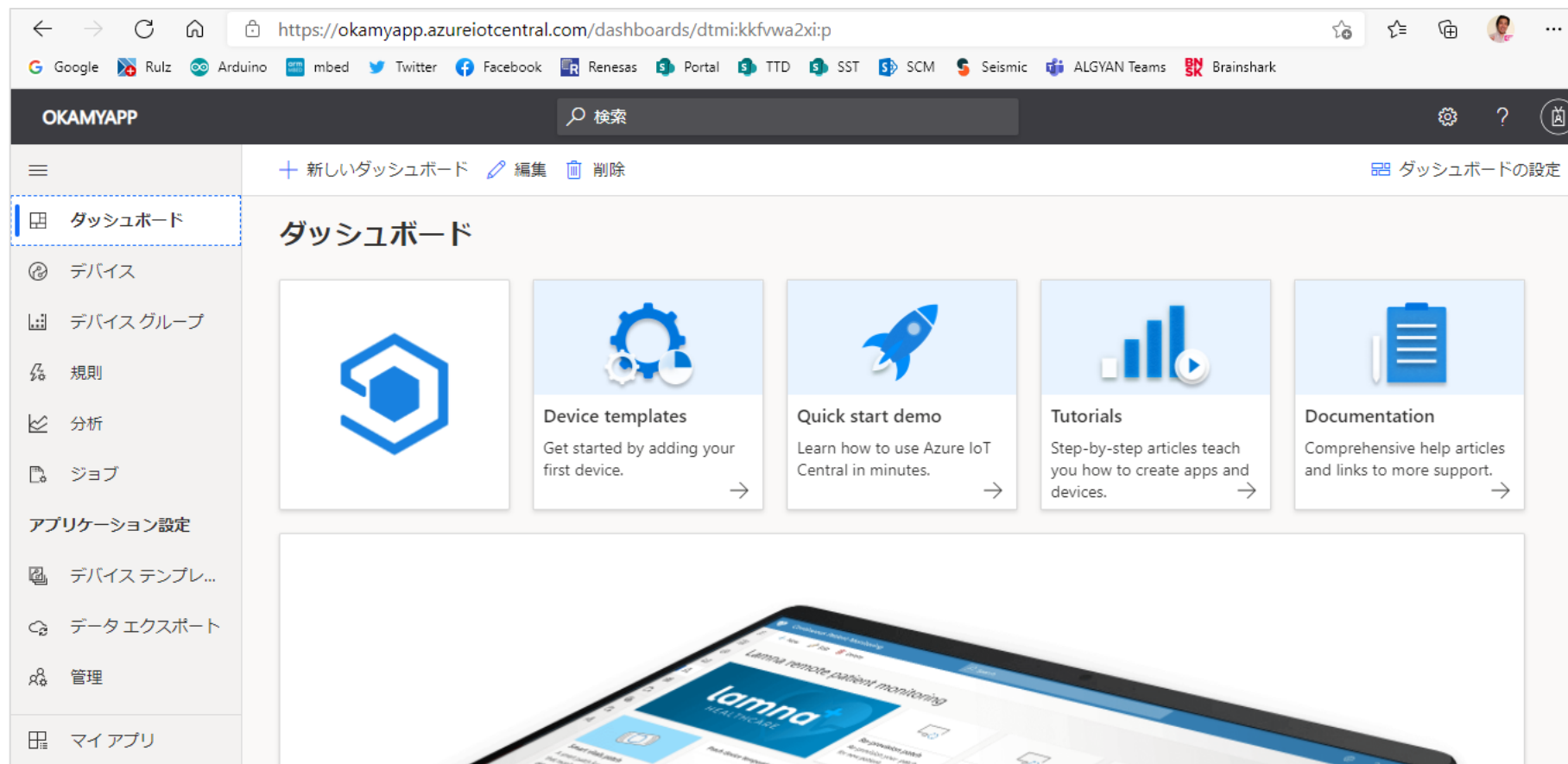
Azure IoT Central





準備 Azure IoT Central ログイン

- Azure IoT Centralにログインしてください。ここではAzureのアカウント作成方法などは省略します。





準備 デバイステンプレート (JSON)

- がじえっとるねさすのコミュニティサイトから「デバイステンプレート」をダウンロードしておきます。

RENESAS Search RenesasRulz.com Join or sign in

がじえっとるねさすコミュニティ > GR-ROSE IoTシステム開発コンテスト > Meet up用

- GR-COTTON
- GR-CITRUS
- GR-PEACH
- GR-KAEDE
- GR-ADZUKI
- GR-LYCHEE
- GR-ROSE
- GR-MANGO(*)
- SNShield
- Web Compiler
- IDE for GR

Azure IoT Central用

- デバイステンプレート

デフォルトサンプル用のテンプレート (JSON) です。テレメトリとして温湿度を取得、コマンドとしてLED、サーボを制御するようになっています。Azure IoT Centralでデバイスを新規に追加する際、インポートしてください。

```

1 [
2 {
3   "@id": "dtmi:okamyapp:grRoseSampleTemplateTelcs;1",
4   "@type": "Interface",
5   "contents": [
6     {
7       "@id": "dtmi:okamyapp:grRoseSampleTemplateTelcs;1",
8       "@type": [
9         "Telemetry",
10        "Temperature"
11      ],
12      "displayName": {
13        "en": "Temperature"
14      },
15      "name": "Temperature",
16      "schema": "double"
17    },

```

Download gr-rose-sample-template.json

gr-rose-sample-template.json

Not Answered

0 replies

2 subscribers

14 views

0 members are here

GR-ROSE

More

Related Tags

コミュニティサイトから「デバイステンプレート」をダウンロード

https://japan.renesasrulz.com/cfs-filesystemfile/_key/communityserver-discussions-components-files/178/gr_2D00_rose_2D00_sample_2D00_template.json?_=637664993372473157

AZURE デバイステンプレート作成 新規

- 「デバイステンプレート」を新規に作成します。その後に「IoTデバイス」を選択します。

OKAMYPAPP 検索

+ 新規

ダッシュボード

デバイス

デバイグループ

規則

分析

ジョブ

アプリケーション設定

デバイステンプレ...

データエクスポート

マイアプリ

デバイステンプレートを選択後、「+新規」を押す

デバイステンプレートの作成

デバイステンプレートは、デバイスの特性と動作を定義するブループリントです。 [詳細情報](#)

カスタム デバイステンプレートの作成

IoT デバイス

機能モデルをインポートするか、初めから機能を作成します。

新規ボタンを押した後、「IoTデバイス」を選択して次へ



AZURE デバイステンプレート作成 名前入力

- デバイステンプレート名を「gr-rose-sample-template」として次へをクリックします。

OKAMYAPP 検索

種類を選択
カスタマイズ
確認

デバイステンプレート > 新規作成

カスタマイズ

デバイステンプレート名*

gr-rose-sample-template

これはゲートウェイデバイスです

テンプレート名を適当に入力し次へ。
例では「gr-rose-sample-template」と入力。

[前へ]をクリックします。 次: レビュー



AZURE デバイステンプレート作成 インポート

- 「モデルのインポート」を選択し、ダウンロードしたJSONファイルを指定します。

The screenshot shows the OKAMYAPP interface. The top navigation bar includes the OKAMYAPP logo, a search bar, and settings/help icons. The left sidebar lists various application settings, with 'デバイス テンプレ...' (Device Template...) selected. The main content area displays the 'gr-rose-sample-template' page, which includes a breadcrumb 'デバイス テンプレート > gr-rose-sample-template' and a title 'gr-rose-sample-template'. Below the title, there is a section titled 'モデルの作成' (Model Creation) with the instruction 'カスタム モデルを最初から作成するか、既存のモデルをインポートします。' (Create a custom model from scratch or import an existing model). Two options are presented: 'カスタム モデル' (Custom Model) and 'モデルのインポート' (Import Model). The 'モデルのインポート' option is highlighted with a red dashed box, and a yellow callout box points to it with the text: 「モデルのインポート」を選択し、ダウンロードしたJSONファイルを指定 (Select 'Import Model' and specify the downloaded JSON file).



AZURE デバイステンプレート作成 公開

- インポート後「公開」ボタンをクリックします。

OKAMYAPP 検索

バージョン テストデバイスの管理 **公開** 名前の変更 削除

デバイス テンプレート > gr-rose-sample-template

「公開」をクリック

gr-rose-sample-template

アプリケーションが更新されました: Never インターフェイスが公開されました: Never

モデル

- gr-rose-sample-template
- クラウドのプロパティ
- 生データ
- カスタマイズ
- ビュー

gr-rose-sample-template ルート ドラフト

このデバイス モデルに固有の機能を追加します。 [詳細情報](#)

保存 + 機能の追加 ID の編集 エクスポート 削除 ... DTDL の編集

表示名	名前 *	機能の種類 * ⓘ	セマンティック... ⓘ		
Temperature	Temperature	Telemetry	Temperature	×	▽
Humidity	Humidity	Telemetry	Humidity	×	▽
Angle	Angle	Command		×	▽



AZURE デバイス作成 新規

OKAMYAPP 検索

デバイス < + 新規 ← インポート

テンプレートをフィルタリングして...

すべてのデバイス

gr-rose-algyan

sample-template

「デバイス」を選択

「+ 新規」を押す

デバイス名	デバイス ID
gr-rose-sample1	1q9mqt9cdvq



AZURE デバイス作成 名前、テンプレート指定

- 「デバイス名」を入力し、「デバイステンプレート」に先ほど公開したテンプレートを選択します。

新しいデバイスを作成します

デバイス名 * ⓘ
my_rose 「デバイス名」を適当に入力

デバイス ID * ⓘ
3vqydyzb0f

組織 * ⓘ
OKAMYAPP

デバイステンプレート *
gr-rose-sample-template 「デバイステンプレート」に公開したテンプレートを選択

作成 キャンセル



AZURE デバイス 表示

- 作成した「デバイス」をクリックします。

The screenshot shows the OKAMYAPP interface. The left sidebar contains a menu with items like 'ダッシュボード', 'デバイス', 'デバイスグループ', '規則', '分析', 'ジョブ', 'アプリケーション設定', 'デバイステンプレ...', 'データエクスポート', '管理', and 'マイアプリ'. The main content area is titled 'デバイス' and shows a list of devices. The first device, 'my_rose', is highlighted with a red dashed box. A yellow callout box with the text '「デバイス」をクリック' points to this device. The table below shows the device details.

デバイス名	デバイス ID	デバイスの状態	テ
my_rose		登録済み	g



AZURE デバイス 接続

- デバイスの画面で「**接続**」をクリックします。

The screenshot shows the OKAMYAPP interface. The top navigation bar includes the OKAMYAPP logo, a search bar, and icons for settings, help, and user profile. The left sidebar contains a menu with options: ダッシュボード, デバイス (selected), デバイスグループ, 規則, 分析, ジョブ, アプリケーション設定, デバイス テンプレ..., データ エクスポート, 管理, and マイ アプリ. The main content area shows a device named 'my_rose' with a '接続' (Connect) button highlighted by a red dashed box. A yellow callout box with the text '「接続」をクリック' points to this button. Below the device name, there is a 'コマンド' (Command) section with a '生データ' (Live Data) option. The main display area shows a template named 'gr-rose-sample-template / Angle' with three input fields labeled X, Y, and Z.

AZURE デバイス 接続情報の確認

- 「IDスコープ」「デバイスID」「主キー」はWEBコンパイラのプログラムで必要になるため、コピーしておきます。

The screenshot shows the '接続の種類' (Authentication Type) section of the IoT Central console. The 'IDスコープ' (ID Scope) is 'One0030852F', the 'デバイスID' (Device ID) is 'n6y3k8u4lx', and the '主キー' (Main Key) is 'HBONI/xWU7nDxcVeTsej70a1yVfdQ1AdudXGyP+Ptcs='.

Callouts from the image:

- 「IDスコープ」 → 「ID_SCOPE」
- 「デバイスID」 → 「REGISTRATION_ID」
- 「主キー」 → 「DEVICE_SYMMETRIC_KEY」



WEBコンパイラ ONLINE化、接続情報の入力

- WEBコンパイラのプログラムで「**#define OFFLINE**」のコメントアウト、「**接続情報**」の入力を行います（以下参照）。

```

1 azure_v104/sketch.cpp x
21 #include <nxd_sntp_client.h>
22 #include <nx_secure_tls_api.h>
23 #include <nx_azure_iot_hub_client.h>
24 #include <nx_azure_iot_provisioning_client.h>
25 #include <nx_azure_iot_cert.h>
26 #include <nx_azure_iot_ciphersuites.h>
27 #include <r_ether_rx_if.h>
28 extern VOID nx_driver_rx_fit(NX_IP_DRIVER *driver_req_ptr);
29 }
30
31 // #define OFFLINE
32 /*****
33 #define ENDPOINT "global.azure-devices-provisioning.net"
34 #define ID_SCOPE "0ne0030852F"
35 #define REGISTRATION_ID "n6y3k8u4lx"
36 #define DEVICE_SYMMETRIC_KEY "HBoNI/xWU7nDxcVeTsej70a1yVfdQ1AdudXGyP+Ptcs="
37 #define MODULE_ID ""
38 /*****
39 #define NX_AZURE_IOT_STACK_SIZE (2048)
40 #define NX_AZURE_IOT_THREAD_PRIORITY (4)
41 #define SAMPLE_MAX_BUFFER (256)
42 #define SAMPLE_STACK_SIZE (2048)
43 #define SAMPLE_THREAD_PRIORITY (16)
44 #define MAX_PROPERTY_COUNT 2
45 static NX_SECURE_X509_CERT root_ca_cert;
46 static UCHAR nx_azure_iot_tls_metadata_buffer[NX_AZURE_IOT_TLS_METADATA_BUFFER_SIZE];
47 static ULONG nx_azure_iot_thread_stack[NX_AZURE_IOT_STACK_SIZE / sizeof(ULONG)];
48 static NX_AZURE_IOT nx_azure_iot;

```

「**#define OFFLINE**」をコメントアウト（行頭に//を入れる）

「**ID_SCOPE**」
「**REGISTRATION_ID**」
「**DEVICE_SYMMETRIC_KEY**」
をコピーした接続情報に書き換える



動作確認 シリアルモニター

- 前述の手順でWEBコンパイラでビルド、binファイルを書き込むと、シリアルモニタに結果が表示されます。

```
COM21 - Tera Term VT
File Edit Setup Control Window Help
DHCP In Progress...Done
IP address: 192.168.0.11
Mask: 255.255.255.0
Gateway: 192.168.0.1
DNS Server address: 192.168.0.1
SNTP Time Sync...
SNTP Time Sync successfully.
Start Provisioning Client...
Registered Device Successfully.
IoT Hub Host Name: iotc-89f29fbf-7abe-49f4-b273-b7eff9a27d48.azure-devi
ces.net; Device ID: n6y3k8u4lx.
Connected to IoT Hub.
Receive twin properties :{"desired":{"$version":1},"reported":{"$versi
on":1}}
Telemetry message send: {"Temperature":30.77, "Humidity":62.18}.
Telemetry message send: {"Temperature":31.40, "Humidity":62.18}.
Telemetry message send: {"Temperature":31.38, "Humidity":62.12}.
Telemetry message send: {"Temperature":31.37, "Humidity":62.09}.
```



動作確認 AZURE 生データ

- AZURE側で「生データ」をクリックすると、テレメトリの受信を確認できます。

OKAMYAPP 検索

接続 Manage template デバイスの管理

デバイス > gr-rose-sample-template > my_rose

my_rose
Connected | データの最終受信日: 2021/9/11 3:39:03 | 状態: プロビジョニング済み | 組織: OKAMYAPP

コマンド **生データ**

「生データ」をクリックすると
テレメトリの受信を確認できる。

タイムスタンプ ↓	メッセージの種類	イベントの作成時刻	Humidity
> 2021/9/11 3:39:06	テレメトリ		62.09
> 2021/9/11 3:39:05	テレメトリ		62.1
> 2021/9/11 3:39:04	テレメトリ		62.1
> 2021/9/11 3:39:03	テレメトリ		62.12
> 2021/9/11 3:39:02	テレメトリ		62.16
> 2021/9/11 3:39:01	テレメトリ		62.18

https://okamyapp.azureiotcentral.com/devices/details/n6v3k8u4lx/rawdata



AZURE デバイステンプレート ビューの設定

OKAMYAPP 検索

メニュー

- ダッシュボード
- デバイス
- デバイスグループ
- 規則
- 分析
- ジョブ
- アプリケーション設定
- デバイステンプレート
- 管理
- マイアプリ

モデル

- gr-rose-sample-template
- クラウドのプロパティ
- 生データ
- カスタマイズ
- ビュー
- About

デバイスとクラウドのデータの編集
このビューを使用して、機能モデルとソリューション モデルで定義されているプロパティを編集および表示するためのフォームを作成します

デバイスの視覚化
このビューで、数々のグラフ、ゲージ、メトリック タイルを使用した、表現力豊かな機能モデルのダッシュボードを作成します

既定のビューの生成
直感的なダッシュボード操作でデバイス情報の表示をすばやく開始するために、既定のデバイス ビューを生成します

「デバイス」をクリック

「ビュー」をクリック

「既定のビューの生成」をクリック

「デバイステンプレート」をクリック



AZURE デバイステンプレート ビューの編集

- 「**Overview**」をクリックするとビューの編集ができます。温湿度のグラフが表示されるようになります。

OKAMYAPP 検索

戻る 保存 削除 プレビュー デバイスの構成

ビューの編集

▼ ビューの設定

ビュー名 * ⓘ

Overview

タイトルの追加

[ビジュアルから始める](#) ...

タイトルに表示するビジュアルの種類を選択し、[タイトルの追加] をクリックします (または、キャンバスにドラッグアンドドロップするだけです)。新しいタイトルの [設定] アイコンをクリックして、コンテンツまたはデバイス データを追加します。

主要業績評価指標 (KPI)

Temperature, Humidity

Temperature

Humidity

「Overview」をクリックするとビューの編集ができる。



AZURE デバイステンプレート 公開

- デバイステンプレートを「公開」します。ビューの設定がデバイスに反映されます。

OKAMYAPP 検索

このデバイス テンプレートは公開されています。公開されている機能を編集すると、ダッシュボード、ジョブ、ルール、またはデータ エクスポートで破壊的変更が発生するおそれがあります。
[詳細情報](#)

バージョン テストデバイスの管理 **公開** 名前の変更 削除

デバイス テンプレート > gr-rose-sample-template

gr-rose-sample-template
 アプリケーションが更新されました: 34 分前 インターフェイスが公開されました: 34 分前

テンプレートの初期画面に戻り、「公開」をクリック

gr-rose-sample-template ルート 発行済み
 このデバイス モデルに固有の機能を追加します。 [詳細情報](#)

保存 + 機能の追加 ID の編集 エクスポート 削除 ... DTDL の編集

表示名	名前 *	機能の種類 * ⓘ	セマンティック... ⓘ		
Temperature	Temperature	Telemetry	Temperature	X	∨
Humidity	Humidity	Telemetry	Humidity	X	∨



動作確認 AZURE 温湿度のグラフ表示

- 「デバイス」を選択し、「Overview」をクリックすると温湿度がグラフ表示されることが確認できます。

OKAMYAPP 検索

接続 Manage template デバイスの管理

About Overview コメント

「Overview」をクリック

「デバイス」を選択

Temperature, Humidity

Temperature Humidity

30.94

平均, 過去 12 時間

Humidity

62.28

平均, 過去 12 時間

03:23 午前 03:54 午前

https://okamyapp.azureiotcentral.com/devices/details/n6y3k8u4lx/dtmi:solutionModel:modelDefinition:q2rsqszqjw41zguyw:sud2cfqy



動作確認 AZURE コマンドの実行

- 「コマンド」をクリックし、LED1の設定を「1」にして「実行」すると、GR-ROSEのLED1が光ります。
- そのほかLED2の制御、Angleとしてx, y, zの値を送信するコマンドをサンプルで用意しています。

OKAMYAPP 検索

接続 Manage template デバイスの管理

デバイス > gr-rose-sample-template > my_rose

my_rose
Connected | データの最終受信日: 2021/9/11 3:57:04

About Overview **コマンド** 生データ

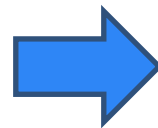
gr-rose-samp **「コマンド」をクリック**

▼ Angle

X

Y

Z



gr-rose-sample-template / LED1

LED1

1 LED1に「1」を入力

実行 「実行」をクリック

応答を確認するには、[コマンド履歴](#)をご確認ください。

LEDが点灯

サンプルプログラム解説





解説 テレメトリ

Arduino loop()

```
136     /// Send telemetry via message queue
137     UINT buffer_length;
138     buffer_length = (UINT)snprintf(buf_telemetry, sizeof(buf_telemetry), "{\"Temperature\":%0.2f, \"Humidity\"
139 #ifndef OFFLINE
140     tx_queue_send(&rose_telemetry, buf_telemetry, TX_WAIT_FOREVER);
141 #else
142     Serial.println(buf_telemetry); // for offline
143 #endif
144
145     delay(1000);
```

テレメトリ生成部分

キューを送信

およそ1秒ごとにループ

テレメトリ タスク部分

```
196     tx_queue_receive(&rose_telemetry, buf_telemetry, TX_WAIT_FOREVER);
197     if (nx_azure_iot_hub_client_telemetry_send(&iot_hub_client, packet_ptr,
198     (UCHAR *)buf_telemetry, sizeof(buf_telemetry),
```

キューを受信

テレメトリ送信



解説 ダイレクトメッセージ (コマンド)

ダイレクトメッセージ タスク部分

```
280 char method[10] = {0}; // for method name
281 strncpy(method, (CHAR *)method_name_ptr, method_name_length);
282
283 if(strcmp("Angle", method)==0){
284     StaticJsonDocument<200> doc;
285     DeserializationError error = deserializeJson(doc, packet_ptr -> nx_packet_prepend_ptr);
286     if (error) {
287         Serial.print(F("deserializeJson() failed: "));
288         Serial.println(error.c_str());
289     }
290     int x = doc["x"];
291     int y = doc["y"];
292     int z = doc["z"];
293     printf("x = %d, y = %d, z = %d\r\n", x, y, z);
294 } else {
295     if(strcmp("LED1", method)==0){
296         char packet[10] = {0};
297         strncpy(packet, (CHAR *)packet_ptr -> nx_packet_prepend_ptr, (INT)(packet_ptr -> nx_
298         int value = atoi(packet);
299         digitalWrite(PIN_LED1, value);
```

メソッド抽出

Angleの場合、ArduinoJSONでパース

パースしたx, y, zを表示するのみ。
(実際は何かにご活用ください)

LED1の場合、atoiで数値化し、
digitalWriteでLED制御



KNOWN ISSUE

- LANケーブルの活線挿抜には対応していません。
最初から最後まで接続されていることが前提になっています。
- DHCP失敗時の再接続がうまくいかないときは、マイコンをリブートしています。
- C2D、デバイスツインの使い方がよくわからなかったなので、解説に入れていません。
ただ、サンプルではタスク生成しています。詳しい方は是非試してみてください。



改訂履歴

Rev.	改定日	内容
1	2021年9月11日	新規