

予知保全のための STM32 MPU OpenSTLinux 拡張パックの使用方法

概要

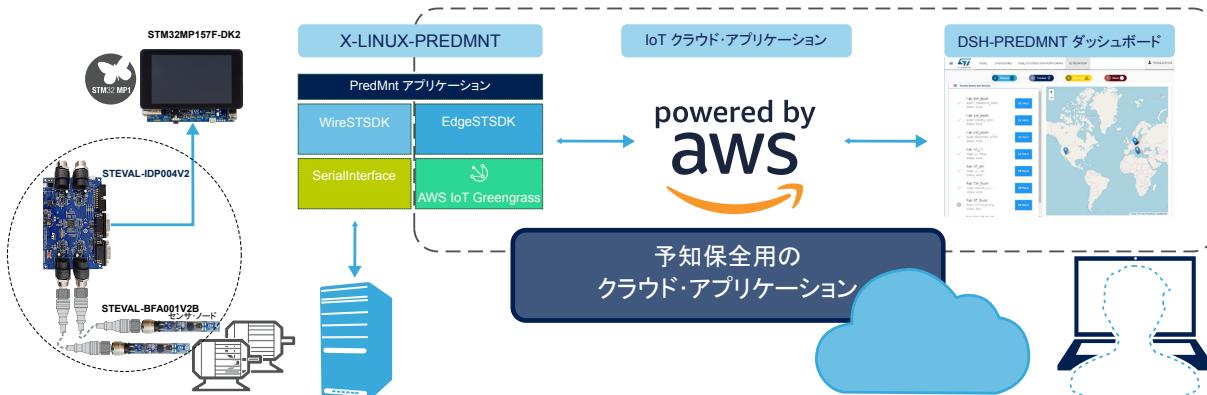
予知保全用の STM32 MPU OpenSTLinux 拡張パックを使用すると、エッジ処理アプリケーションを開発できます。対応するハードウェアのあるエンドツーエンドのソリューションを形成し、産業用機器から IoT アプリケーションへの環境データおよび慣性データの送信を可能にします。IoT アプリケーションには、データ分析と、メンテナンスが直ちに、または将来的に必要になる可能性のある状態を特定するための、専用のダッシュボードがあります。

このアプリケーションは、リモート・デバイスの登録、ゲートウェイの設定、IoT クラウドサービスへの接続など、リモート IoT センサ・ノードを使用した効果的な状態監視に必要な多くの重要側面を管理するのに役立ちます。特に、Amazon AWS IoT クラウドと接続し、ゲートウェイ上で AWS IoT Greengrass エッジ・コンピューティング・サービスを使用することで、インターネット接続が断続的になる条件下でも、ローカル・ロジックを実行し、データをシームレスに送信します。

このアプリケーションの開発ハードウェアには、モータ付き振動キット、慣性測定ユニットを搭載した STEVAL-BFA001V2B IO-Link センサ・ボード、センサ・データの計算および管理用のオンボード・マイクロコントローラを搭載したさまざまな環境センサ、ならびに STEVAL-IDP004V2 IO-Link マスター・ボードが含まれています。ゲートウェイ・ノードは、さまざまなワイヤードおよびワイヤレス・コネクティビティ・ソリューション、SD カード・データ・ストレージ、LCD タッチスクリーン・インターフェース、および適切な高性能 STM32MP1 シリーズのマイクロプロセッサを搭載した STM32MP157F-DK2 ディスカバリ・キットでセットアップされます。

エッジ・ゲートウェイは、加速度センサから環境データと FFT データを収集し、これらを AWS インフラストラクチャに基づいて MQTT over Ethernet または Wi-Fi を介してダッシュボードに送信します。

図 1. 状態監視とエッジ・クラウド: センサからゲートウェイ、そしてクラウドまでのダッシュボード



1 エッジ処理アプリケーションの概要

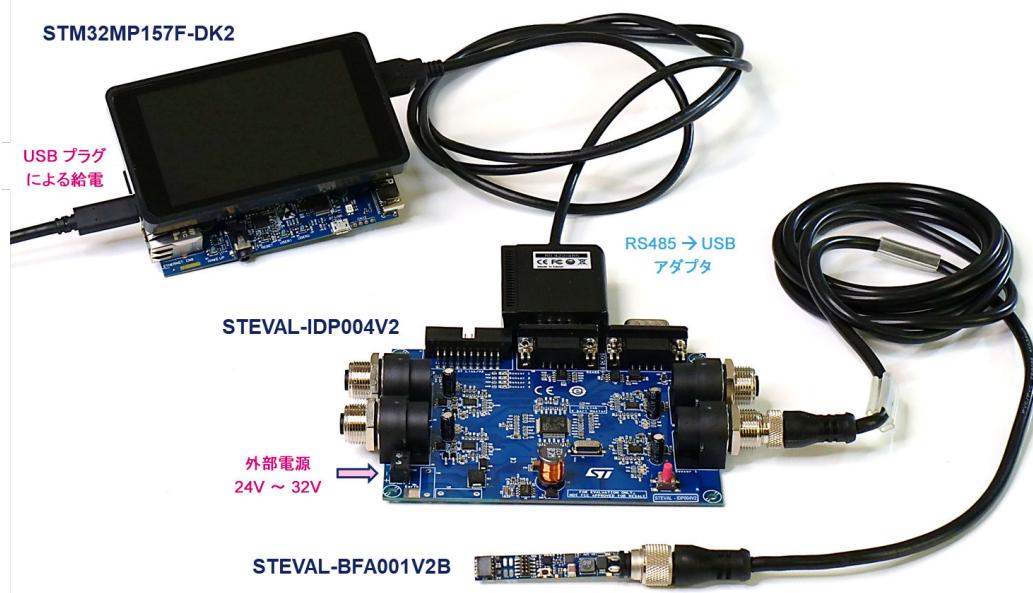
エッジ処理アプリケーションをセットアップするには、次の要素が必要です。

- ゲートウェイ・ノード:
 - STM32MP157F-DK2 ディスカバリ・キット(4GB 以上の SD カード)
- スマート・センサとマスタ・ハードウェア:
 - マスタ・ボード: STEVAL-IDP004V2
 - センサ・ボード: STEVAL-BFA001V2B キット
- エッジ処理アプリケーション・ダッシュボード用の myST アカウント。
- STM32 ST-LINK 用 ユーティリティを搭載した PC および ST-LINK プログラマ(を搭載しスタンドアロンまたは STM32 Nucleo ボードに統合されたもの)。
- プロキシやファイアウォールなしのインターネット接続。

下の図に、上記のすべてのボードに加えて、マスタと MP1 を接続する RS-485-to-USB アダプタを使用して、ソリューションを正しく構築する方法を示します。

このソリューションでは、有線または Wi-Fi コネクティビティを使用して、MP1 ゲートウェイのクラウドへの接続を開始できます。

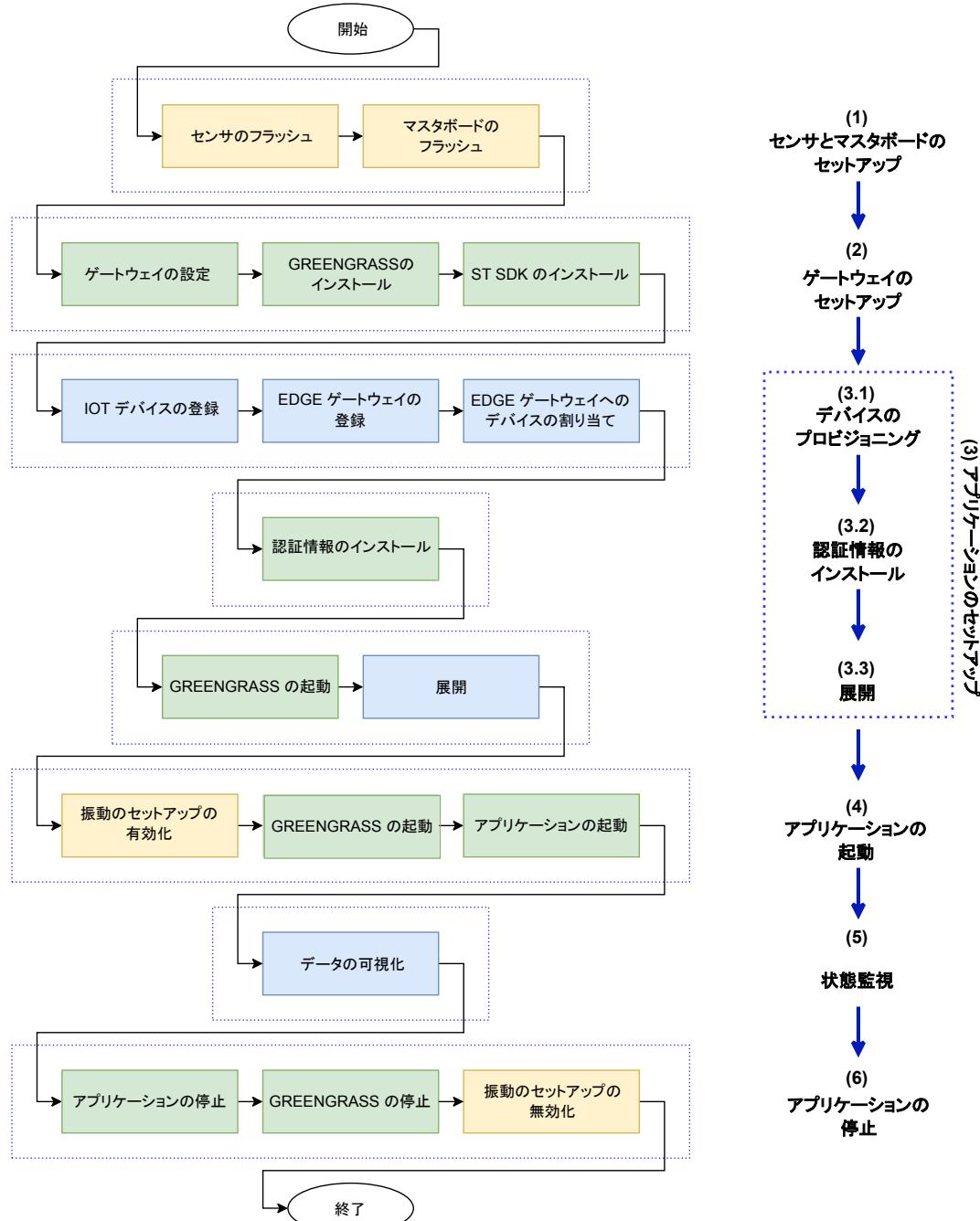
図 2. SL-PREDMNT-E2C ソリューションのセットアップ



2 エッジ処理アプリケーションのセットアップと動作

次のフロー・チャートに、アプリケーションのセットアップと実行に関する手順の概要を示します。

図 3. セットアップと操作



2.1 センサ・ボードとマスター・ボードのセットアップ

次の手順に従って、センサ・ボードおよびマスター・ボードに最新のファームウェアをフラッシュします。

Step 1. ST ウェブサイトの次の場所から、センサ・ボードおよびマスタ・ボードのバイナリ・ファイルをダウンロードします。

- センサ・ノード: STSW-BFA001V2 で入手可能な STSW-BFA2PREDMNT
- マスタ・ボード: STSW-IDP4PREDMNT

Step 2. 関連するバイナリを STEVAL-IDP004V2 IO-Link マスタ・ボードと STEVAL-BFA001V2B IO-Link センサ・ボードにフラッシュします。

STM32 ST-LINK (STSW-LINK004) ユーティリティおよび適切な ST-LINK プログラマ/デバッガ・デバイス(スタンダロンまたは STM32 Nucleo ボードに統合)を使用します。

2.2

ゲートウェイのセットアップ

現在のソリューションは、以下のソフトウェアに基づきます。

- STM32MP15-Ecosystem-v2.1.0 (openstlinux-5.4-dunfell-mp1-20-11-12)
- Amazon AWS IoT Greengrass Core Software v1.11.0
- Python 3.8.2

この章で説明する手順はこれらに類似しており、STM32MP15-Ecosystem-v2.1.0 のより新しいディストリビューションでも有効です。

STM32MP157F-DK2 ディスカバリ・キットを予知保全プラットフォームのエッジ・ゲートウェイとして設定するには、次のいずれかを行うことができます。

- ST によって事前設定されたイメージをフラッシュ(書き込み)します。
 - X-LINUX-PREDMNT
- カスタム・イメージを作成してフラッシュ(書き込み)します。

図 4. STM32MP157F-DK2 ディスカバリ・キット

デュアルコア Cortex-A7 CPU、800 MHz GPU、および Cortex-M4 マイクロコントローラを搭載した STM32 マイクロコントローラセキュア・ブートと暗号化、LCD、Wi-Fi、Bluetooth low energy technology



重要

マイクロスイッチは、プログラム書き込み前に OFF、書き込み後に ON に設定します。

—— 関連リンク ——

この無料ツールを使用して、バイナリイメージを SD カードにマスタリングすることができます。

STM32 MPU の wiki ページで、STM32 MPU に関するガイドとリソースをご覧いただけます。

2.2.1 STM32MP1 ディスカバリ・キット用イメージの作成方法

次の手順に従って、STM32MP157F-DK2 ディスカバリ・キットを Linux ゲートウェイとして設定します。

注 この手順は、STM32MP1 DK2 C01 および C2 リリースにのみ適用されます。

Step 1. STM32 MPU wiki ページの PC の前提条件ページに従って、ホスト環境をセットアップします。

https://wiki.st.com/stm32mpu/wiki/PC_prerequisites

Step 2. ディスカバリ・ボードを初めて使用する場合は、スターターパッケージをインストールし、空の SD カードにイメージをフラッシュして、すべてのパーティションを作成します。

https://wiki.st.com/stm32mpu/wiki/STM32MP15_Discovery_kits_-_Starter_Package

重要 マイクロスイッチは、プログラム書き込み前に OFF、書き込み後に ON に設定します。

Step 3. 次の wiki ページの説明に従って、ディストリビューションパッケージを取得します。

https://wiki.st.com/stm32mpu/wiki/STM32MP1_Distribution_Package

Step 4. 最低でもイメージをビルドするためのコマンドまでを含む命令に従います。

```
bitbake st-image-weston
```

注 「bitbake」命令を実行すると、常に次のようなエラーが発生する可能性があります。

```
ERROR: No space left on device or exceeds fs.inotify.max_user_watches?
ERROR: To check max_user_watches: sysctl -n fs.inotify.max_user_watches.
ERROR: To modify max_user_watches: sysctl -n -w fs.inotify.max_user_watches=<value>.
ERROR: Root privilege is required to modify max_user_watches.
```

この場合、次のコマンドを実行します。

```
sudo sysctl -n -w fs.inotify.max_user_watches=32768
```

引き続きエラーが発生する場合は、値を「2」のべき乗で増やします。

再起動して「bitbake」命令を繰り返します。引き続きエラーが発生する場合は、回避策を繰り返します。
イメージ・ファイルはディストリビューションパッケージに含まれています。

Step 5. <layers> ディレクトリに移動して、以下の URL から入手可能な meta-predmnt レイヤをインストールします。

<https://github.com/STMicroelectronics/meta-predmnt>

このレイヤには、予知保全アプリケーション、エッジ・コンピューティング用 Amazon AWS IoT Greengrass サービス(<https://aws.amazon.com/ja/greengrass/>)、およびその他の必要な Python パッケージのインストール方法が含まれています(詳細については、meta-predmnt/conf/layer.conf ファイルを参照)。

```
cd <path-to>/openstlinux-<version>/layers
```

```
git clone https://github.com/STMicroelectronics/meta-predmnt
```

```
cd <path-to>/openstlinux-<version>/build-openstlinuxweston-stm32mp1/conf/
```

Step 6. 次の行を bblayers.conf ファイルにコピーして BBLAYERS トークン定義の直前に貼り付け、BBLAYERS トークンに FRAMEWORKLAYERS トークンが含まれていることを確認し、含まれていない場合はこれを追加します。

```
FRAMEWORKLAYERS += "${@'${OEROOT}/layers/meta-predmnt' if os.path.isfile('${OEROOT}/layers/meta-predmnt/conf/layer.conf') else ''}"
```

Step 7. <openstlinux> ディレクトリに移動し、カーネルの USB サポートを有効にして、次をビルドします。

```
cd <path-to>/openstlinux-<version>
```

```
bitbake virtual/kernel -c menuconfig
```

Step 7a. [Device Drivers] と [USB support] に移動します。

Step 7b. 次のエントリにチェックを入れます。アスタリスク(*)文字が表示されるまで [スペース] キーを押してから、Enter キーを押します。

```
USB Serial Converter support
```

Step 7c. 次のエントリにチェックを入れます。アスタリスク(*)文字が表示されるまで [スペース] キーを押してから、Enter キーを押します。

```
USB Serial Console device support
```

```
USB Generic Serial Driver
```

```
USB Serial Simple Driver
```

```
USB FTDI Single Port Serial Driver
```

Step 7d. 保存して終了します。

```
bitbake virtual/kernel
```

Step 8. <openstlinux> ディレクトリに移動して、イメージをビルドします(ディストリビューションパッケージのページで説明している命令と同じです)。

```
cd <path-to>/openstlinux-<version>
```

```
DISTRO=openstlinux-weston MACHINE=stm32mp1 source layers/meta-st/scripts/envsetup.sh
```

```
bitbake st-image-weston
```

Step 9. <image> ディレクトリに移動し、イメージをフラッシュします(スタータパッケージのページで説明している命令と同じです)。

重要 マイクロスイッチは、プログラム書き込み前に OFF、書き込み後に ON に設定します。

```
cd <path-to>/openstlinux-<version>/build-openstlinuxweston-stm32mp1/tmp-glibc/deploy/images/stm32mp1
```

```
STM32_Programmer_CLI -c port=usb1 -w flashlayout_st-image-weston/FlashLayout_sdcard_stm32mp157c-dk2+trusted.tsv
```

2.2.2 ゲートウェイの設定方法

重要 マイクロスイッチは、ゲートウェイを起動にする前にオンに設定します。

Step 1. ゲートウェイを起動し、メイン GUI が表示されるまで待ちます。

図 5. ゲートウェイのメイン GUI



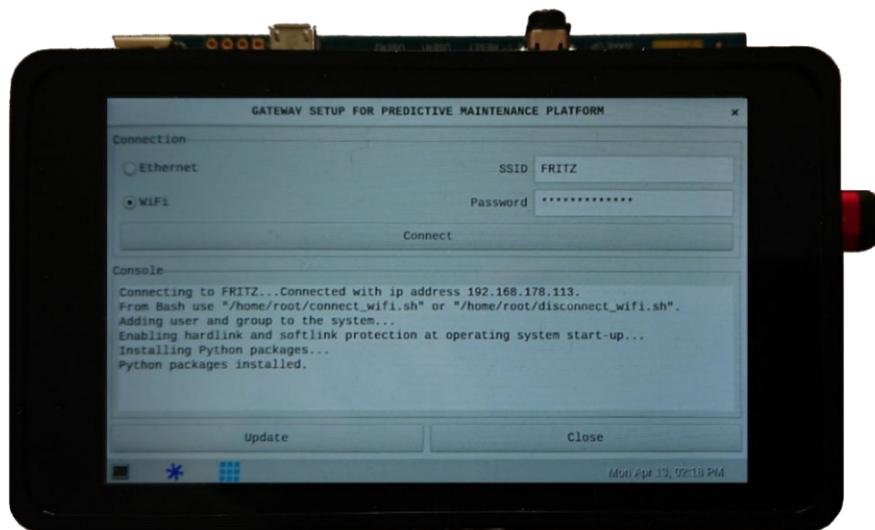
メイン GUI は、次のコマンドで起動することもできます。

```
/usr/local/predmnt/start.sh --gui
```

Step 2. [SETUP GATEWAY] ボタンをクリックして、ゲートウェイの設定を完了します。

次のパネルが表示されます。ゲートウェイを設定して、Wi-Fi ネットワークまたは Ethernet ケーブルを介してインターネットに接続できます。

図 6. ゲートウェイ GUI のセットアップ



Step 3. [Update] ボタンをクリックして、次のライブラリをインストールまたは更新します。

- Amazon AWS IoT Python SDK: <https://github.com/aws/aws-iot-device-sdk-python>
- WireST-SDK: https://github.com/STMicroelectronics/WireSTSDK_Python
- EdgeST-SDK: https://github.com/STMicroelectronics/EdgeSTSDK_Python
- その他の Python パッケージ(pyserial)

2.2.3 SSH 経由でゲートウェイに接続する方法

ゲートウェイがネットワークに接続したら、次の手順を実行します。

- Step 1.** 次のコマンドを使用して、ゲートウェイの端末から IP アドレスを取得します。

```
ifconfig
```

- Step 2.** 次のコマンドを実行して、SSH 経由で別のデバイスからゲートウェイに接続します（「openstlinux」ディストリビューションでは、ユーザに「root」を使用する必要があります）。

```
ssh <user>@<IP-ADDRESS>
```

2.2.4 再起動してゲートウェイをオフにする方法

- Step 1.** ゲートウェイを再起動するには、次のコマンドを実行します。

```
reboot
```

- Step 2.** ゲートウェイを安全にオフにするには、次のコマンドを実行します。

```
shutdown 0
```

2.2.5 ゲートウェイを STEVAL-IDP004V2 に接続する方法

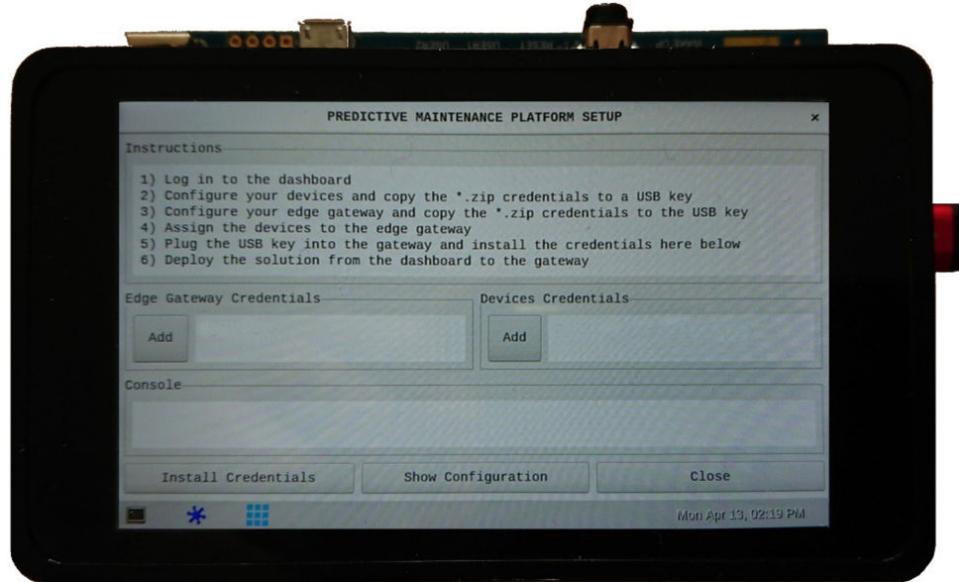
システムを正しく実行するには、ゲートウェイを STEVAL-IDP004V2 IO-Link マスタ・ボードに接続します。

ドングル USB を RS485 に接続します。ボード側は DB9 コネクタ(CN1)に、ゲートウェイ側は USB ポートに接続します。DB9 コネクタは EX-1303 ドングルと互換性があります。異なるドングルを使用する場合は、ピンの互換性に注意してください(詳細については、STEVAL-IDP004V2 回路図)。

2.3 アプリケーションのセットアップ

メイン GUI で、中央のボタン（「セットアップ・アプリケーション」）をクリックして設定手順を開始し、ゲートウェイおよびダッシュボードで行う手順に従ってください。

図 7. GUI アプリケーションのセットアップ



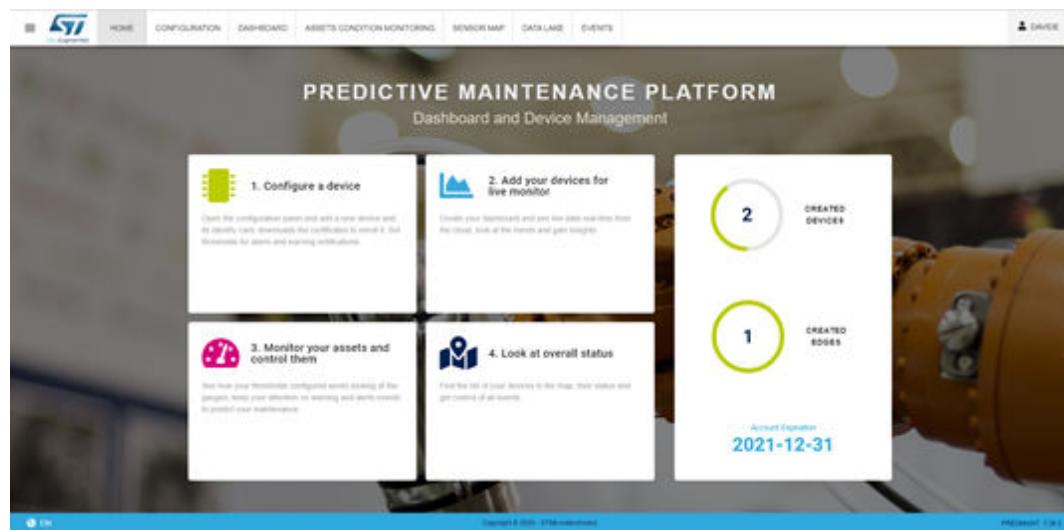
- Step 1.** ダッシュボードにログインします。
- Step 2.** デバイスを設定し、*.zip の認証情報を USB キーにコピーします。
- Step 3.** エッジ・ゲートウェイを設定し、*.zip の認証情報を USB キーにコピーします。
- Step 4.** デバイスをエッジ・ゲートウェイに割り当てます。
- Step 5.** USB キーをゲートウェイに接続して、認証情報をインストールします。
- Step 6.** ダッシュボードからゲートウェイにソリューションを展開します。

2.3.1 ダッシュボードへのログイン

重要 ダッシュボードを起動してデバイスを登録するには、有効な myST アカウントが必要です。

- Step 1.** ダッシュボードを開いて、myST の認証情報でログインします。
– <https://dsh-predmnt.st.com/>

図 8. 予知保全プラットフォームのメイン画面



2.3.2 デバイスの設定方法

- Step 1.** [Register a new device] をクリックし、[DEVICES] パネルをクリックします。
- Step 2.** [Add Device] をクリックし、2 つのデバイスを適切な名前とオプション情報を使用して登録します。

図 9. [Add Device] ページ

The screenshot shows the 'Add Device' registration form. The title bar says 'Add Device' with a close button. The form fields are as follows:

- Name Format: A radio button labeled 'Free' is selected.
- Device ID: An input field containing 'TSDemo1'.
- Asset Name: An input field containing 'Pump1'.
- Fab Name: An input field containing 'F13'.
- Coordinates: Two input fields for latitude and longitude, both showing '45.571445' and '9.362989' respectively, with location icons.
- Buttons: 'CANCEL' and 'REGISTER' buttons at the bottom right.

- Step 3.** ポップアップ・ウィンドウを閉じる前に、対応する zip ファイル(証明書と鍵を含む)をダウンロードし、USB キーのルートにコピーします。

図 10. 設定ファイルのダウンロード

Configuratlon Files For: TSDEMO1

Download the Zip file with the files necessary to connect the device with the cloud



I confirm that I have downloaded the Zip file
 (after closing the file can no longer be downloaded)

IoT Endpoint:

alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.com

CLOSE

2.3.3 エッジ・ゲートウェイの設定方法

- Step 1.** [EDGE GATEWAYS] ダッシュボード・パネルに移動します。

- Step 2.** [Add Device] をクリックし、適切な名前でエッジ・ゲートウェイを登録します。

図 11. 新しいエッジの追加

Add New Edge

Edge Name ■
TSDEMO

CANCEL ADD GROUP

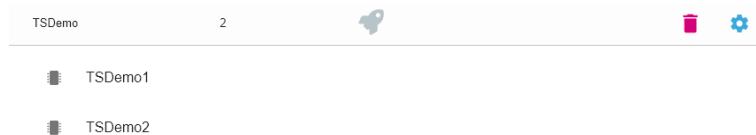
- Step 3.** ポップアップ・ウィンドウを閉じる前に、対応する zip ファイル(証明書および鍵の資格情報)をダウンロードし、以前に使用した USB キーのルートにコピーします。

2.3.4 エッジ・ゲートウェイにデバイスを割り当てる方法

- Step 1.** [EDGE GATEWAYS] ダッシュボード・パネルで、[Setting] 齒車アイコンをクリックしてから [Devices] をクリックします。

- Step 2.** ゲートウェイに追加するデバイスを選択して、[Save] をクリックします。

図 12. エッジ・ゲートウェイへのデバイスの割り当て



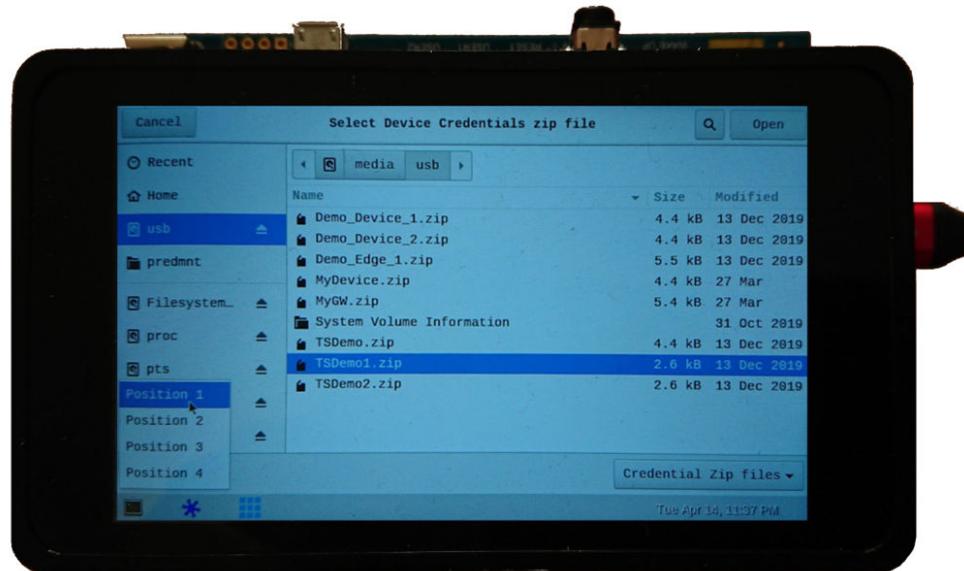
2.3.5

ゲートウェイに認証情報をインストールする方法

Step 1. 以前使用した USB キーをゲートウェイに接続し、対応する [Add] ボタンをクリックしてゲートウェイの認証情報と最大 4 台のデバイスを追加します。

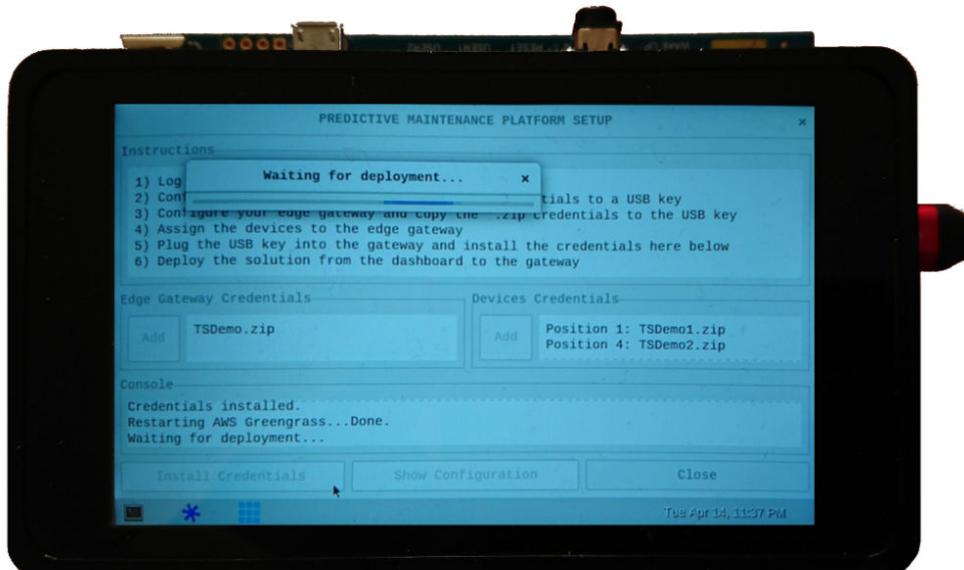
Step 2. マスター・ボード上のデバイスの位置に応じて、デバイスの位置を選択します(1 ~ 4)。

図 13. エッジ・ゲートウェイへのデバイスの割り当て



Step 3. [Install Credentials] ボタンをクリックして、ゲートウェイに承認情報をインストールします。
Greengrass デーモンが再起動され、ゲートウェイはダッシュボードからの展開を待ちます。

図 14. 展開待ち

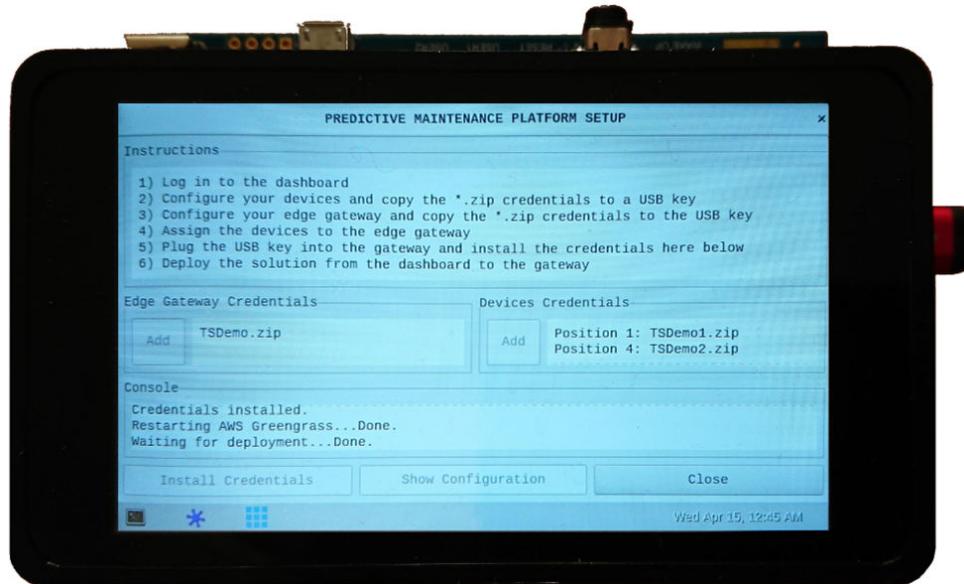


2.3.6 ソリューションの展開方法

Step 1. ダッシュボードの [EDGE GATEWAYS] パネルに移動して、[Setting] 齒車アイコンをクリックし、[Deploy] を選択してから [CONFIRM] を選択します。

接続が正しく動作している場合、展開は数秒後には正常に完了し、ゲートウェイが設定されます。

図 15. ゲートウェイでの正常な展開



Step 2. ウェブページをリロードして、「New Deployment」操作が正しく完了しているかどうか、ステータス(緑色のアイコン)を確認します。必要に応じて繰り返し行ってください。

図 16. ダッシュボードでの正常な展開



2.4 アプリケーションの開始

2.4.1 振動のセットアップを有効にする方法

Step 1. 振動のセットアップを USB ケーブルでゲートウェイに接続します。

Step 2. 振動のセットアップの電源を入れます。

Step 3. 振動のセットアップの速度を任意の値に設定します。

2.4.2 アプリケーションの起動方法

Step 1. 以下のようにアプリケーションを起動します。

- メイン GUI で、[RUN APPLICATION] ボタンをクリックしてアプリケーションを起動します。新しいウインドウがポップアップ表示され、基礎となるアプリケーションの出力が表示されます。
- または、ターミナルから次のスクリプトを実行します。

```
/usr/local/predmnt/start.sh
```

- または、次のコマンドを起動します。

- /greengrass/ggc/core/greengrassd restart

- export PYTHONPATH=\$PYTHONPATH:/usr/local/predmnt/

- python3 /usr/local/predmnt/pmp.py -c /usr/local/predmnt/pmp.json

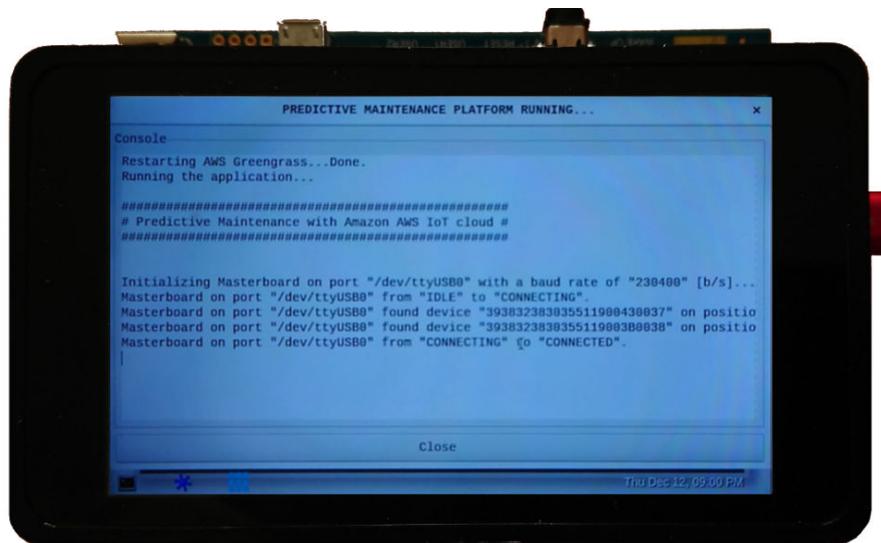
2.5 状態監視

2.5.1 ゲートウェイでのデータの可視化

アプリケーションの起動時に、ゲートウェイ上で次の手順を実行します。

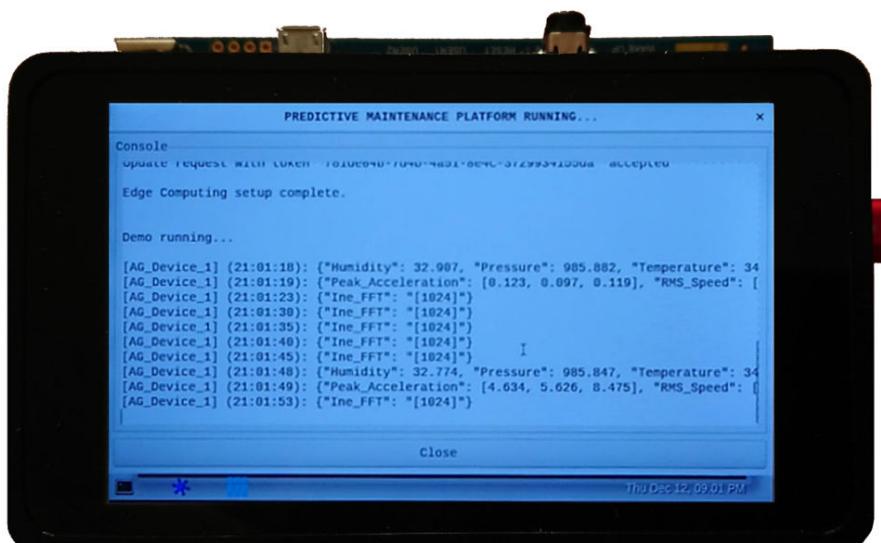
1. Greengrass デーモンが起動します。
2. WireST-SDK がマスター・ボードとセンサ・デバイスを初期化します。

図 17. Greengrass の起動とデバイスの初期化



3. Greengrass デーモンは、EdgeST-SDK 経由でクラウド上の Greengrass コアに接続します。
4. デバイスとクラウド上の相手が通信を開始します。
5. アプリケーションは、センサからデータを受信して、クラウドに送信します。

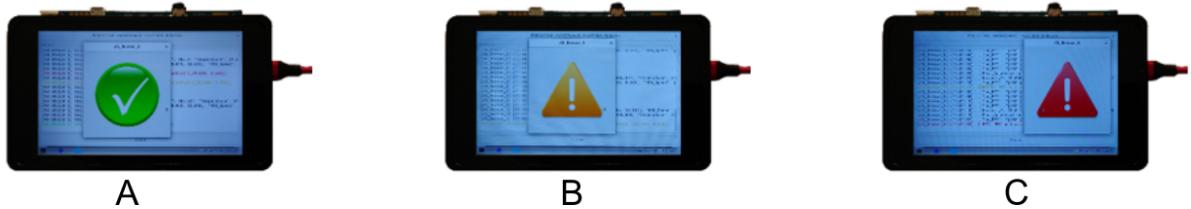
図 18. ゲートウェイがセンサからのデータを受信



6. オプション: ダッシュボードから展開されたソリューションにデータをローカルで処理する Lambda 関数(ダッシュボードの実装による)が含まれている場合(通常は振動データの高速フーリエ変換(FFT:Fast Fourier Transform)のフィルタ)、トリガされたアラームがゲートウェイに表示されます。
さまざまなアラーム・ウィンドウが画面にポップアップ表示され、接続しているスピーカーのオーディオ・ジャックから音響アラームが再生されます。

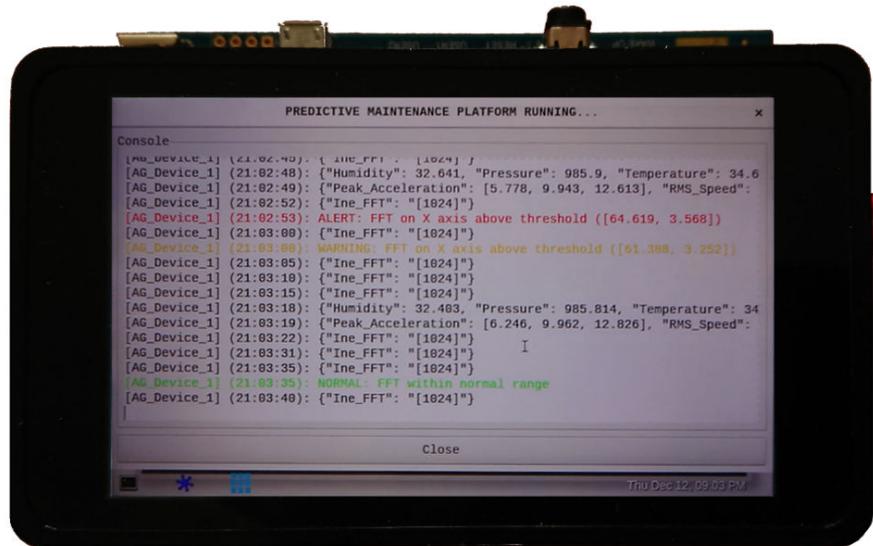
図 19. アラーム・ポップアップ・ウィンドウ

- A. 正常: FFT データが通常範囲内にある場合
B. 警告: FFT データが FFT の警告閾値を超える場合
C. 重要: FFT データが FFT の臨界閾値を超えた場合



さらに、データとアラームの色付きログが表示されます。

図 20. アラーム・ログ



アラーム信号は、今後使用するためにクラウド・ダッシュボードにも送信されます。

2.5.2

ダッシュボードでのデータの可視化

次の手順に従って、ダッシュボードにセンサ・データを表示します。

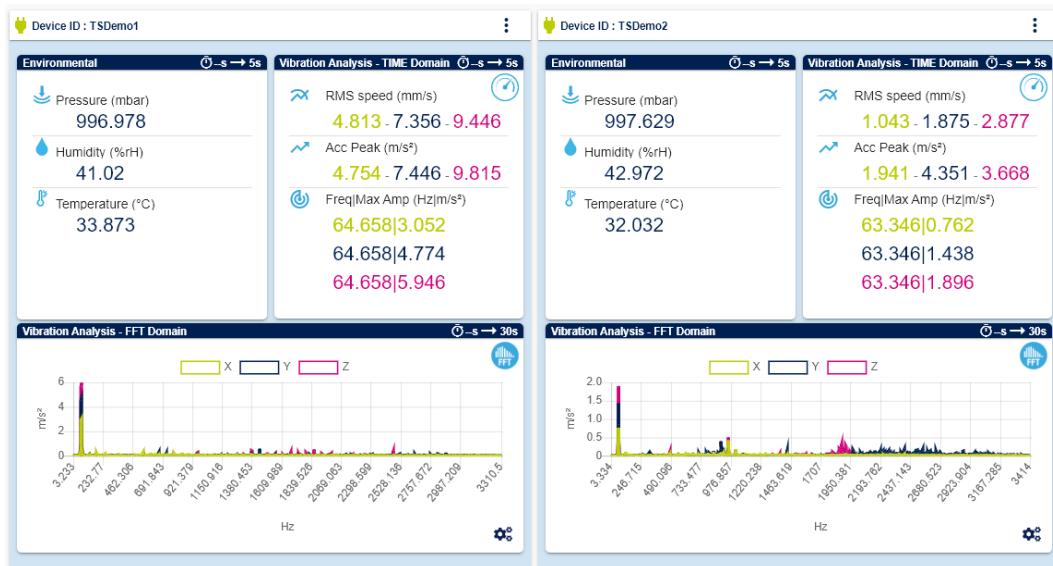
Step 1. ダッシュボードを開いて、myST の認証情報でログインします。

<https://dsh-predmnt.st.com/>

Step 2. [DASHBOARD] をクリックしてから [Add Device] をクリックし、監視対象のデバイスを選択します。

選択した各デバイスのパネルが開き、エクスポートされたデータ値が表示されます。

図 21. センサ・データとパラメータを表示したデバイスのパネル



Step 3. 必要に応じて、デバイスによってエクスポートされた機能に対していくつかの閾値を設定することができます。これはゲージ・パネルでデバイスの動作状態を簡単に確認する場合に便利です。メイン・ページで [Register a new device] を選択し、デバイスを選択し、[Settings] 歯車アイコンをクリックし、[Configure Thresholds] をクリックして、ドロップダウン・メニューから目的の閾値を設定するための機能を選択します。

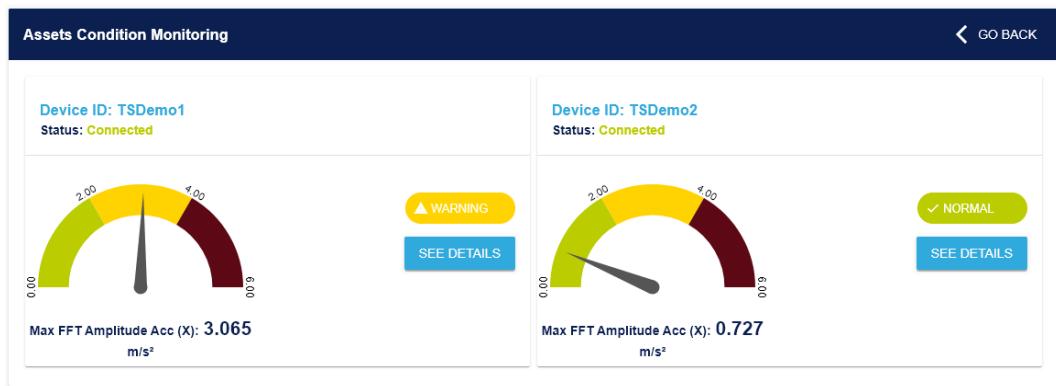
図 22. デバイスごとの閾値設定

Level	Min	Max
NORMAL	0	2
WARNING	2	4
ALERT	4	6

Level	Min	Max
NORMAL	0	2
WARNING	2	4
ALERT	4	6

- Step 4.** [ASSETS CONDITION MONITORING] をクリックします。
選択したデバイスごとのゲージ・パネルが表示され、最後に設定した機能に応じた動作状態が表示されます。

図 23. デバイスごとのゲージ・パネル



2.6 アプリケーションの停止

2.6.1 アプリケーションの停止方法

- Step 1.** アプリケーションを停止するには、次の手順に従います。
- 「RUN APPLICATION」の GUI で、[Close] ボタンをクリックします。
 - または、アプリケーションを起動した端末で [CTRL+C] を押します。
 - または、別の端末から、停止スクリプトを実行します。

```
/usr/local/predmnt/stop.sh
```

2.6.2 振動のセットアップを無効にする方法

- Step 1.** 振動のセットアップの速度をゼロに設定します。
- Step 2.** 電力振動のセットアップをオフに設定します。

3 ゲートウェイ・アプリケーション設定に関する追加情報

エッジ・ゲートウェイの認証情報、Greengrass 設定、ゲートウェイ上のアプリケーションの設定などの追加情報は、アプリケーションの開発時や問題のデバッグに役立つことがあります。

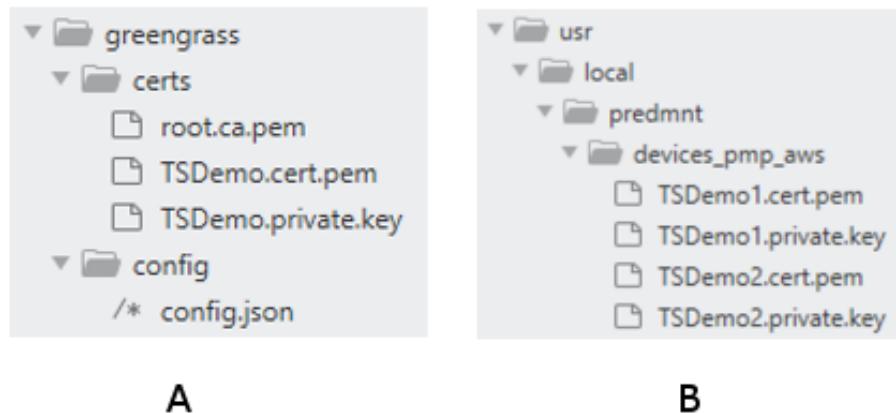
3.1 認証情報

「/greengrass」フォルダには、エッジ・ゲートウェイの認証情報、ルート認証局の証明書ファイル(root.ca.pem)、および Greengrass 設定ファイル(config.json)が含まれています。

「/usr/local/predmnt/devices_pmp_aws」フォルダには、デバイスの認証情報が含まれています。

図 24. ゲートウェイ・ファイルシステムの認証情報

- A. エッジ・ゲートウェイの認証情報
- B. デバイスの認証情報



A

B

3.2 Greengrass の設定

Greengrass デーモンが起動すると直ちに、「config.json」ファイルによって、クラウド・アプリケーションの Greengrass コアへの接続方法に関する情報を取得できます。

以下のコードは、設定ファイルの一例です。

```
{
  "coreThing" : {
    "caPath" : "root.ca.pem",
    "certPath" : "TSDemo.cert.pem",
    "keyPath" : "TSDemo.private.key",
    "thingArn" : "arn:aws:iot:eu-west-1:982787141379:thing/TSDemo_Core",
    "iotHost" : "alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.com",
    "ggHost" : "greengrass-ats.iot.eu-west-1.amazonaws.com",
    "keepAlive" : 600
  },
  "runtime" : {
    "cgroup" : {
      "useSystemd" : "yes"
    }
  },
  "managedRespawn" : false,
  "crypto" : {
    "principals" : {
      "SecretsManager" : {
        "privateKeyPath": "file:///greengrass/certs/TSDemo.private.key"
      },
      "IoTCertificate" : {
        "privateKeyPath": "file:///greengrass/certs/TSDemo.private.key",
        "certificatePath": "file:///greengrass/certs/TSDemo.cert.pem"
      }
    },
    "caPath" : "file:///greengrass/certs/root.ca.pem"
  }
}
```

3.3 アプリケーション設定

アプリケーション・ユーザ設定は、設定ファイル「/usr/local/predmnt/pmp.json」に保存されます。

```
{  
    "serial_port": {  
        "name": "/dev/ttyUSB0",  
        "baudrate_bits_per_second": 230400  
    },  
    "setup": {  
        "use_sensors": true,  
        "use_cloud": true,  
        "use_threads_for_polling_sensors": true,  
        "device_certificates_path": "/usr/local/predmnt/devices_pmp_aws",  
        "devices": [  
            {  
                "name": "TSDemo1",  
                "position": 1  
            },  
            {  
                "name": "TSDemo2",  
                "position": 4  
            }  
        ]  
    },  
    "dump": {  
        "env_samples": 0,  
        "tdm_samples": 0,  
        "fdm_samples": 0  
    }  
}
```

上記のコードは、ゲートウェイ上のアプリケーション設定ファイルの例を示しており、次のパラメータが含まれています。

- `serial_port`: 使用しているシリアル・ポートの名前とボーレート
- `use_sensors`: センサ・デバイスの実際のセットアップを使用するか、ランダム・データをシミュレートする
- `use_cloud`: データをクラウドに送信するかどうか
- `use_threads_for_polling_sensors`: 各センサ・デバイスのデータ・ドメイン(Environmental, Inertial_TDM, Inertial_FDM)ごとに個別のスレッドを実行するか、または連続してポーリングする
- `devices`: マスタ・ボード上で設定されたデバイスの名前と位置を表示する
- `dump`: 提供されたサンプル数までドメイン・データ(Environmental, Inertial_TDM, Inertial_FDM)をダンプします。

注 必要に応じて設定ファイルを編集することによって、コマンドラインからこれらのパラメータを変更できます。

4 トラブルシューティング

4.1 正しい動作

次のコードに、アプリケーション実行時の正しい動作の例を示します。

```
Greengrass successfully started with PID: 1933

#####
# Predictive Maintenance with Amazon AWS IoT cloud #
#####

Initializing Masterboard on port "/dev/ttyUSB0" with a baud rate of "230400" [b/s]
...
Masterboard on port "/dev/ttyUSB0" from "IDLE" to "CONNECTING".
Masterboard on port "/dev/ttyUSB0" found device "393832383035511900430037" on position "1".
Masterboard on port "/dev/ttyUSB0" found device "3938323830355119003B0038" on position "4".
Masterboard on port "/dev/ttyUSB0" from "CONNECTING" to "CONNECTED".

Initializing IO-Link Devices...
Device "TSDemo1" on position "1" initialized.
Device "TSDemo2" on position "4" initialized.

IO-Link setup complete.

Initializing Edge Computing...

AWS Greengrass service with endpoint "alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.com" from "IDLE" to "DISCOVERING_CORE".
AWS Greengrass service with endpoint "alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.com" from "DISCOVERING_CORE" to "CORE_DISCOVERED".
Client "TSDemo1" from "IDLE" to "CONNECTING".
Client "TSDemo1" from "CONNECTING" to "CONNECTED".
Client "TSDemo2" from "IDLE" to "CONNECTING".
Client "TSDemo2" from "CONNECTING" to "CONNECTED".

Sending handshake information...

[TSDemo1] (15:28:16.829431): {"state": {"reported": {"Device_Type": "STEVAL-IPD005 V1", "Features": ["Environmental", "Inertial_TDM", "Inertial_FDM"], "Firmware": "Firmware Ver. 1"}}}
Update request with token "aed4fb7e-1004-4130-a0d6-fada6bbfa295" accepted
[TSDemo2] (15:28:21.059531): {"state": {"reported": {"Device_Type": "STEVAL-IPD005 V1", "Features": ["Environmental", "Inertial_TDM", "Inertial_FDM"], "Firmware": "Firmware Ver. 1"}}}
Update request with token "91d11010-e2f6-4273-92dd-89684859e81f" accepted

Edge Computing setup complete.

Demo running...

[TSDemo1] (15:28:24.389429): {"Humidity": 39.976, "Pressure": 990.625, "Temperature": 37.086}
[TSDemo2] (15:28:26.329349): {"Humidity": 39.548, "Pressure": 991.476, "Temperature": 36.55}
[TSDemo1] (15:28:27.689421): {"Peak_Acceleration": [0.147, 0.108, 0.13], "RMS_Speed": [0.023, 0.024, 0.043]}
```

```
[TSDemo2] (15:28:29.529874): {"Peak_Acceleration": [0.144, 0.119, 0.127], "RMS_Speed": [0.023, 0.017, 0.021]}
[TSDemo1] (15:28:33.260450): {"Ine_FFT": "[1024]"}
[TSDemo2] (15:28:37.081863): {"Ine_FFT": "[1024]"}
[...]
```

4.2 設定ファイルの喪失

アプリケーション設定ファイル「pmp.json」が見つからない場合は、次のエラー・メッセージが表示されます。

```
[...]
#####
# Predictive Maintenance with Amazon AWS IoT cloud #
#####

Configuration file "/usr/local/predmnt/pmp.json" not found.

Exiting...
```

Greengrass の設定ファイル「config.json」が見つからない場合は、次のエラー・メッセージが表示されます。

```
[...]
#####
# Predictive Maintenance with Amazon AWS IoT cloud #
#####

Configuration file "/greengrass/config/config.json" not found.

Exiting...
```

解決策: ファイルの場所が正しいかどうかを確認し、エラー・メッセージが表示されない場合は、アプリケーションを再実行します。

4.3 センサ・デバイスの喪失

設定ファイル「pmp.json」内で指定された 1 つ以上のデバイスを初期化できない場合、次のエラーメッセージが表示されることがあります。

```
[...]
Initializing Masterboard on port "/dev/ttyUSB0" with a baud rate of "230400" [b/s]
...
Masterboard on port "/dev/ttyUSB0" from "IDLE" to "CONNECTING".
Masterboard on port "/dev/ttyUSB0" found device "393832383035511900430037" on position "1".
Masterboard on port "/dev/ttyUSB0" from "CONNECTING" to "CONNECTED".

Initializing IO-Link Devices...
Device "TSDemo1" on position "1" initialized.

IO-Link setup incomplete.

Exiting...
```

解決策:

- 喪失しているセンサ・デバイスとマスタ・ボードの間の接続を確認します。
- アプリケーション設定ファイルを確認してください。
- エラー・メッセージが表示されない場合は、アプリケーションを再実行します。

4.4

デバイスの認証情報の喪失

設定されたデバイスの一部で証明書が見つからない場合は、次のエラー・メッセージが表示されます。

```
[...]
Initializing Edge Computing...

Invalid device certificate path: "/user/local/predmnt/devices_pmp_aws/TSDemo1.cert
.pem"

Exiting...
```

秘密鍵を失った場合、次のメッセージが表示されます。

```
[...]
Initializing Edge Computing...

Invalid device private key path: "/user/local/predmnt/devices_pmp_aws/TSDemo1.priv
ate.key"

Exiting...
```

解決策: 証明書の場所とファイル名、および設定したデバイスの秘密鍵を確認し、エラー・メッセージが表示されない場合は、アプリケーションを再実行します。

4.5

ダッシュボードの問題

ダッシュボードに問題がある場合や、展開が正しく行われていない場合、次のエラー・メッセージが表示されます。

```
[...]
Initializing Edge Computing...

AWS Greengrass service with endpoint "alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.c
om" from "IDLE" to "DISCOVERING_CORE".
Discovery of the core related to the client "TSDemo1", with certificate
"/usr/local/predmnt/devices_pmp_aws/TSDemo1.cert.pem" and key
"/usr/local/predmnt/devices_pmp_aws/TSDemo1.private.key", failed after 3 retries.

Exiting...
```

ソリューションを開くときには、Greengrass デーモンが実行されている必要があります。クラウドでの最後の操作のステータスが正常(展開またはリセットのいずれか)である必要があります。

「SETUP APPLICATION」の GUI を使用して展開すると、Greengrass が自動的に起動します。

コマンドラインから実行する場合は、代わりに以下を行ってください。

- 次のコマンドを実行します。

```
/greengrass/ggc/core/greengrassd restart
```

- 展開して再実行します。問題が解決しない場合は、ダッシュボード管理者に連絡してください。

4.6

クラウド・コネクティビティの問題

クラウド・コネクティビティに問題がある場合、次のエラー・メッセージが表示されます。

```
[...]
Initializing Edge Computing...

AWS Greengrass service with endpoint "alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.com" from "IDLE" to "DISCOVERING_CORE".
AWS Greengrass service with endpoint "alazohj3ky8ktj-ats.iot.eu-west-1.amazonaws.com" from "DISCOVERING_CORE" to "CORE_DISCOVERED".
Client "TSDemo2" from "IDLE" to "CONNECTING".
Client "TSDemo2" from "CONNECTING" to "UNREACHABLE".
Client "TSDemo2" cannot connect to core.
AWS setup incomplete.

Exiting...
```

解決策: インターネットへの接続を確認してから、アプリケーションを再実行します。

4.7

ミスアライメントの問題

WireST-SDK およびメインの PMP アプリケーションに対応していないファームウェアを実行している場合は、次のエラー・メッセージが表示されることがあります。

```
[...]
Edge Computing setup complete.

Demo running...

Read 1 elements. Discarded.
[...]
```

解決策: センサとマスタ・ボードのファームウェア、WireST-SDK、およびメイン・アプリケーションを最新リリースに更新します。その後、アプリケーションを再実行します。

4.8

アプリケーションの予期しないクラッシュ

グラフィック・アプリケーションが予期せずクラッシュした場合は、コマンドラインから再起動してください。

```
/usr/local/predmnt/start.sh --gui
```

改版履歴

表 1. 文書改版履歴

日付	版	変更内容
2019 年 11 月 27 日	1	初版発行
2020 年 2 月 26 日	2	文章の軽微な修正。
2020 年 9 月 24 日	3	新しい GUI リリースを反映するためにすべての内容を更新。
2022 年 1 月 25 日	4	<p>セクション 1 エッジ処理アプリケーションの概要、セクション 2.2 ゲートウェイのセットアップ、セクション 2.2.1 STM32MP1 ディスカバリ・キット用イメージの作成方法、セクション 2.2.2 ゲートウェイの設定方法、セクション 2.2.4 再起動してゲートウェイをオフにする方法、セクション 2.3 アプリケーションのセットアップ、セクション 2.3.1 ダッシュボードへのログイン、セクション 2.4.2 アプリケーションの起動方法、セクション 2.5.2 ダッシュボードでのデータの可視化、セクション 2.6.1 アプリケーションの停止方法、セクション 4.2 設定ファイルの喪失、セクション 4.5 ダッシュボードの問題、および セクション 4.8 アプリケーションの予期しないクラッシュを更新。</p> <p>セクション 2.2.5 ゲートウェイを STEVAL-IDP004V2 に接続する方法を追加。</p>

目次

1	エッジ処理アプリケーションの概要	2
2	エッジ処理アプリケーションのセットアップと動作	3
2.1	センサ・ボードとマスタ・ボードのセットアップ	3
2.2	ゲートウェイのセットアップ	4
2.2.1	STM32MP1 ディスカバリ・キット用イメージの作成方法	5
2.2.2	ゲートウェイの設定方法	6
2.2.3	SSH 経由でゲートウェイに接続する方法	8
2.2.4	再起動してゲートウェイをオフにする方法	8
2.2.5	ゲートウェイを STEVAL-IDP004V2 に接続する方法	8
2.3	アプリケーションのセットアップ	9
2.3.1	ダッシュボードへのログイン	9
2.3.2	デバイスの設定方法	10
2.3.3	エッジ・ゲートウェイの設定方法	11
2.3.4	エッジ・ゲートウェイにデバイスを割り当てる方法	11
2.3.5	ゲートウェイに認証情報をインストールする方法	12
2.3.6	ソリューションの展開方法	13
2.4	アプリケーションの開始	13
2.4.1	振動のセットアップを有効にする方法	13
2.4.2	アプリケーションの起動方法	13
2.5	状態監視	14
2.5.1	ゲートウェイでのデータの可視化	14
2.5.2	ダッシュボードでのデータの可視化	16
2.6	アプリケーションの停止	17
2.6.1	アプリケーションの停止方法	17
2.6.2	振動のセットアップを無効にする方法	17
3	ゲートウェイ・アプリケーション設定に関する追加情報	18
3.1	認証情報	18
3.2	Greengrass の設定	19
3.3	アプリケーション設定	20
4	トラブルシューティング	21
4.1	正しい動作	21
4.2	設定ファイルの喪失	22
4.3	センサ・デバイスの喪失	22
4.4	デバイスの認証情報の喪失	23
4.5	ダッシュボードの問題	23

4.6	クラウド・コネクティビティの問題	24
4.7	ミスアラインメントの問題	24
4.8	アプリケーションの予期しないクラッシュ	24
	改版履歴	25

図一覧

図 1.	状態監視とエッジ・クラウド: センサからゲートウェイ、そしてクラウドまでのダッシュボード.....	1
図 2.	SL-PREDMNT-E2C ソリューションのセットアップ	2
図 3.	セットアップと操作	3
図 4.	STM32MP157F-DK2 ディスカバリ・キット	4
図 5.	ゲートウェイのメイン GUI	7
図 6.	ゲートウェイ GUI のセットアップ	7
図 7.	GUI アプリケーションのセットアップ	9
図 8.	予知保全プラットフォームのメイン画面	10
図 9.	[Add Device] ページ	10
図 10.	設定ファイルのダウンロード	11
図 11.	新しいエッジの追加	11
図 12.	エッジ・ゲートウェイへのデバイスの割り当て	11
図 13.	エッジ・ゲートウェイへのデバイスの割り当て	12
図 14.	展開待ち	12
図 15.	ゲートウェイでの正常な展開	13
図 16.	ダッシュボードでの正常な展開	13
図 17.	Greengrass の起動とデバイスの初期化	14
図 18.	ゲートウェイがセンサからのデータを受信	14
図 19.	アラーム・ポップアップ・ウィンドウ	15
図 20.	アラーム・ログ	15
図 21.	センサ・データとパラメータを表示したデバイスのパネル	16
図 22.	デバイスごとの閾値設定	16
図 23.	デバイスごとのゲージ・パネル	17
図 24.	ゲートウェイ・ファイルシステムの認証情報	18

重要なお知らせ(よくお読み下さい)

STMicroelectronics NV およびその子会社(以下、ST)は、ST 製品及び本書の内容をいつでも予告なく変更、修正、改善、改定及び改良する権利を留保します。購入される方は、発注前に ST 製品に関する最新の関連情報を必ず入手してください。ST 製品は、注文請書発行時点で有効な ST の販売条件に従って販売されます。

ST 製品の選択並びに使用については購入される方が全ての責任を負うものとします。購入される方の製品上の操作や設計に関して ST は一切の責任を負いません。明示又は默示を問わず、ST は本書においていかなる知的財産権の実施権も許諾致しません。

本書で説明されている情報とは異なる条件で ST 製品が再販された場合、その製品について ST が与えたいかなる保証も無効となります。

ST および ST ロゴは STMicroelectronics の商標です。ST の登録商標については ST ウェブサイトをご覧ください。www.st.com/trademarks. その他の製品またはサービスの名称は、それぞれの所有者に帰属します。

本書の情報は本書の以前のバージョンで提供された全ての情報に優先し、これに代わるものです。

© 2022 STMicroelectronics – All rights reserved