演習 3: 分散制御アプリケーションの作成

この演習では、センサー データを MicroZed インテリジェント I/O モジュールから AWS クラウドへ送信し、アセット オーナーへのアラートを監視および受信します。

Greengrass コアの設定変更とデプロイ

前の演習では、AWS Greengrass の諸設定をデプロイしました。この演習では、この AWS Greengrass 設定を更新します。設定を更新すると、次のようになります。

- コア、デバイス、およびロギングの定義は変わりません。
- Lambda の定義が変わります。最後の演習で使用する Lambda がグループから削除されます。
- サブスクリプションの定義が変わります。hello world サブスクリプションが削除され、Zynq-7000 デバイスからクラウドへテレメトリを伝搬する新しいサブスクリプションが追加されます。

この場合、6 つの定義のうち 2 つだけが変わり、Greengrass グループ バージョンも変更されます。設定を更新後、新しいグループをデプロイします。

Greengrass コア経由での MicroZed センサー データの送信

MicroZed IIoT キットは、FPGA を経由して 3 つのセンサーを利用できます。各センサーとそのインターフェイスは、次のとおりです。

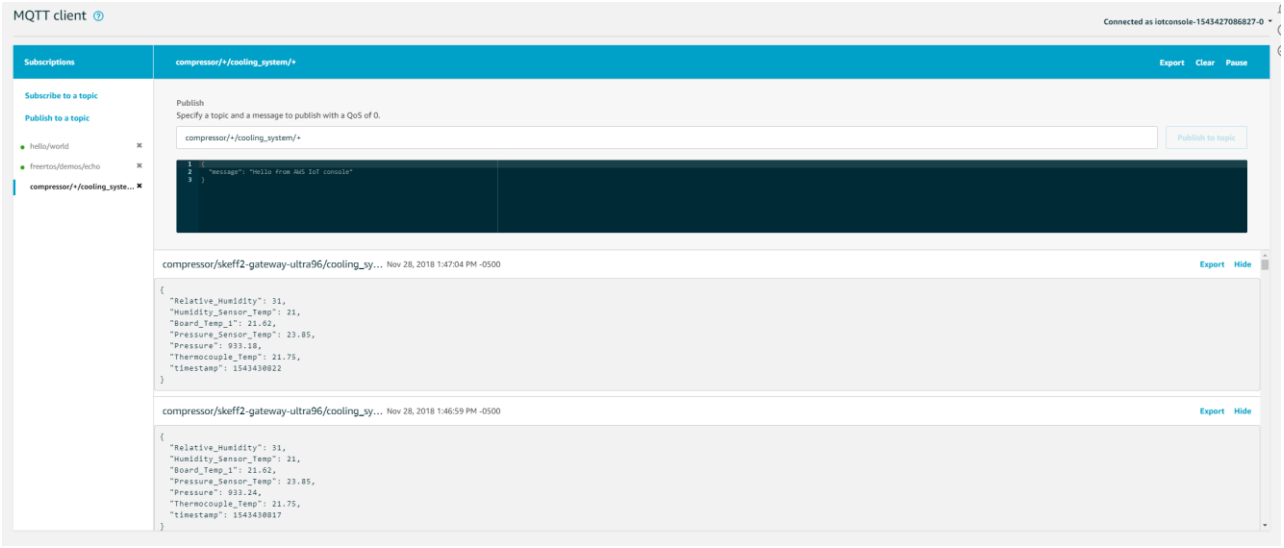
- Maxim MAX31855: 熱電対および IC 温度センサー (SPI)
- STMicroelectronics LPS22HB: 気圧および IC 温度センサー (I2C)
- STMicroelectronics HTS221: 湿度および IC 温度センサー (I2C)

Amazon FreeRTOS のリファレンス イメージは、これらのインターフェイスを一定のレートでポーリングし、取得したデータを Ultra96 デバイスで動作する AWS Greengrass に送信するようにプログラムされています。

Greengrass コアは、これらのメッセージを AWS クラウドに転送します。MicroZed 上で動作する Amazon FreeRTOS は、Greengrass Discovery 機能を使用して Greengrass コア エンドポイントを検出および設定します。次に、AWS IoT コンソールからセンサー MQTT トピックをサブスクライブし、クラウドへのデータフローを確認します。

1. AWS IoT コンソール ページにアクセスし、左側メニューの **[Test]** をクリックします。
2. **[Subscriptions]** ヘッダーの下にある **[Subscribe to a topic]** をクリックします。
3. **[Subscription topic]** に「`compressor/+/cooling_system/+`」と入力します。
4. **[Subscribe to topic]** をクリックします。

次の図に示すように、各センサーから取得したデータ値が AWS IoT コンソールに表示されます。分散制御アプリケーションでは、これらの値がユニット コントローラーと共有されます。MicroZed と Ultra96 上の AWS Greengrass コアの間の通信経路は、演習 1 で実行したエッジ設定スクリプトで設定したものです。



MicroZed のセンサー エラー

MicroZed の Amazon FreeRTOS 上でビルドしたアプリケーションには、センサーをチェックしてエラー条件を検出する機能が実装されています。また、それ自身および制御対象のアセットを保護するために、制御システムに警告を送信します。

このセクションでは、システムの動作中に MAX31855 ボードから熱電対を取りはずして、センサーのエラーをシミュレーションします。熱電対を取りはずすと、**開回路条件**が FPGA プラットフォームによって検出され、ユニット コントローラーおよび AWS クラウドに対してセンサー ヘルス メッセージが生成されます。

> この実装では、冷却システム モジュール (UltraZed) が AWS Greengrass コアのシャドウを直接変更し、問題が発生したことを知らせるアラートを送信します。この演習では、アクチュエーションを簡単に体験できるように、シンプルにした実装モデルを採用しています。より現実的な実装では、冷却システム モジュールはそれ自身のシャドウの **Reported State** を変更し、AWS Greengrass の AWS Lambda 関数で変更を読み出すようにします。そして、インテリジェント I/O モジュールのシャドウを利用してコンプレッサーのインテリジェント I/O モジュールの LED を点灯させるなど、さらにアクションを実行する必要があるかどうかを判断します。

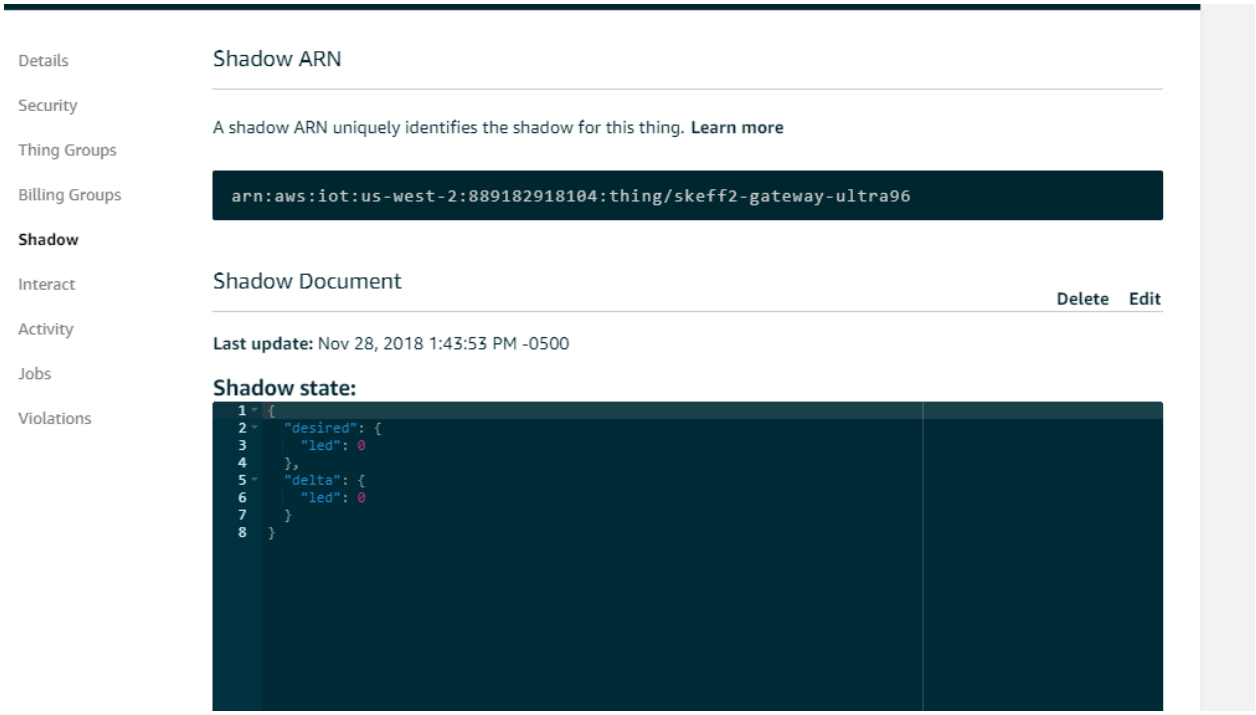
1. AWS IoT コンソールの左側メニューに **[Manage]** があります。

2. **[Manage]** のすぐ下にある **[Things]** をクリックします。

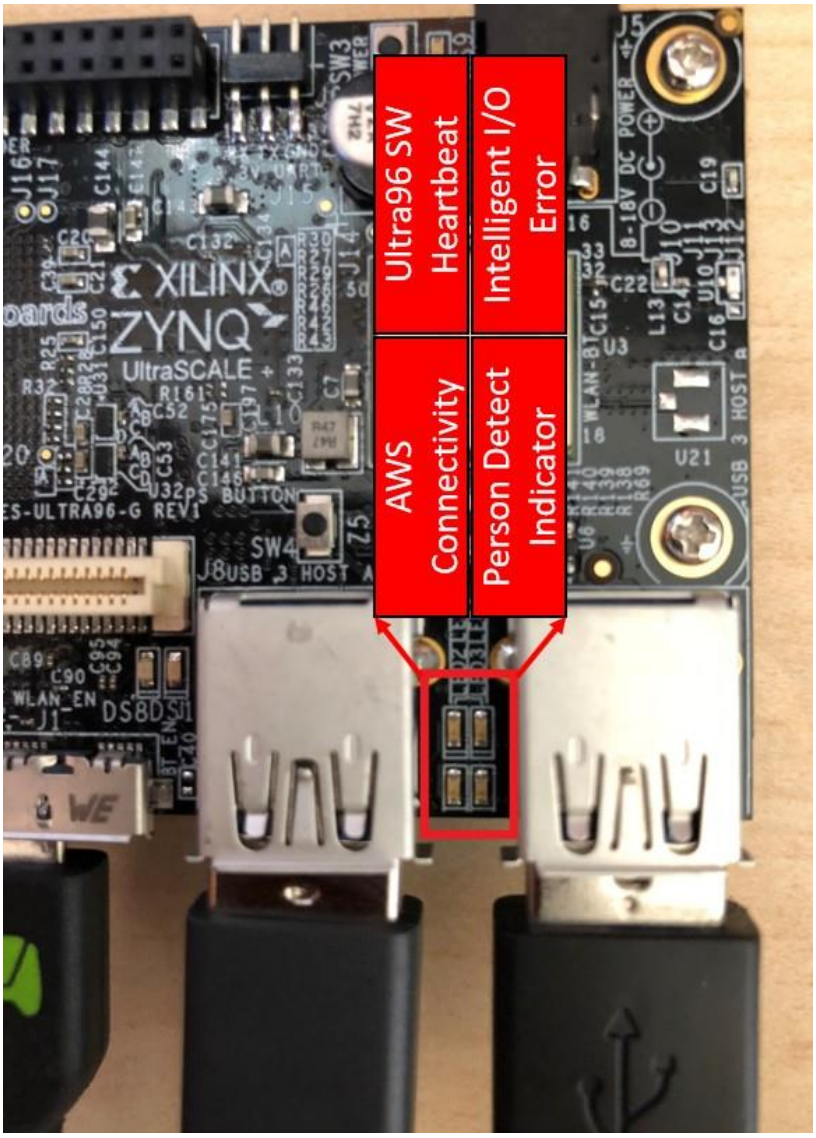
3. 目的のコアに該当する Thing (モノ) を見つけます。名前は `<prefix>-gateway-ultra96` です。

4. **[Shadow]** をクリックします。

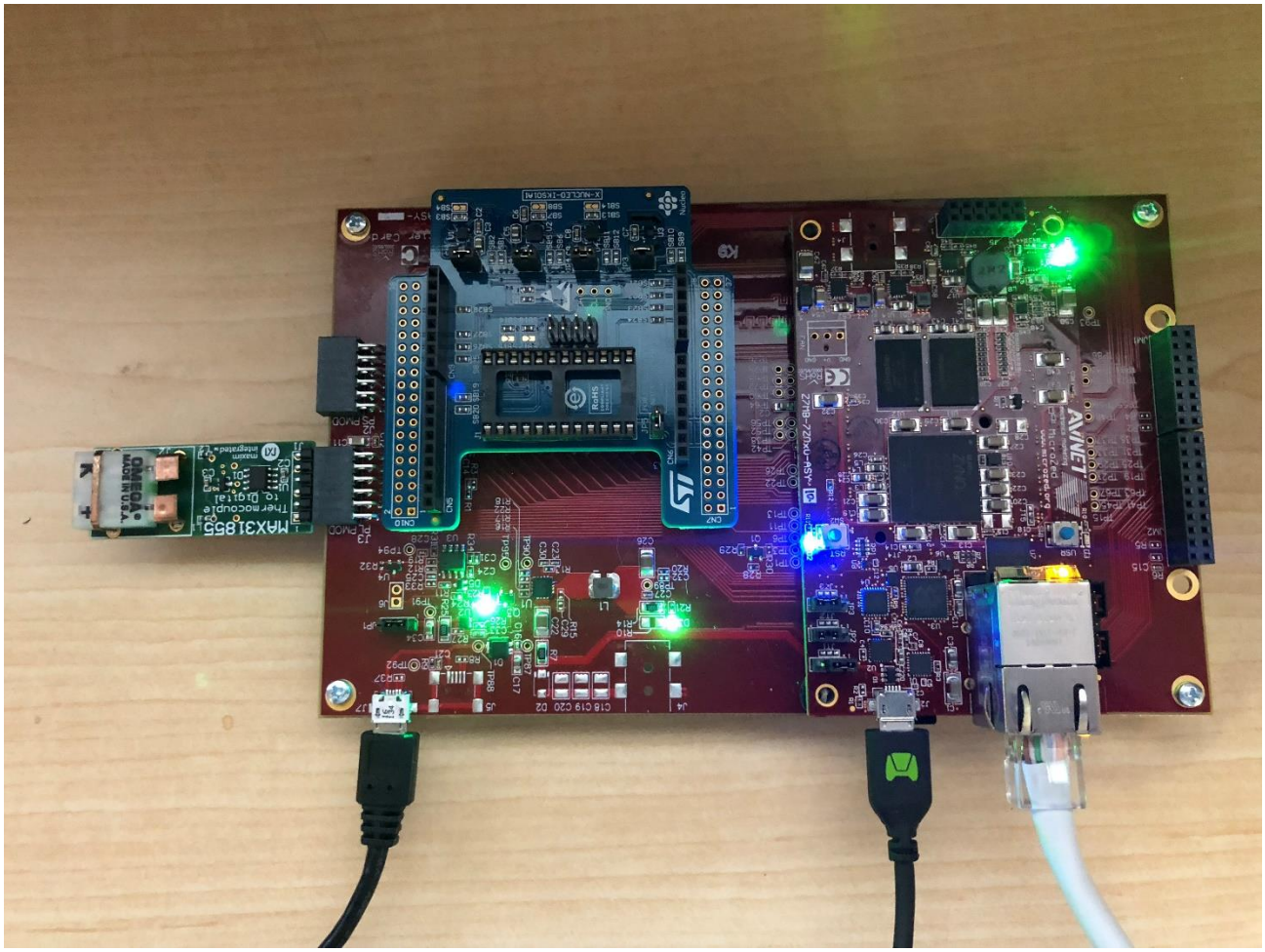
5. AWS IoT コンソールで、熱電対のステート メッセージにエラーが発生していないこと (「led」の値が 0) を確認します。



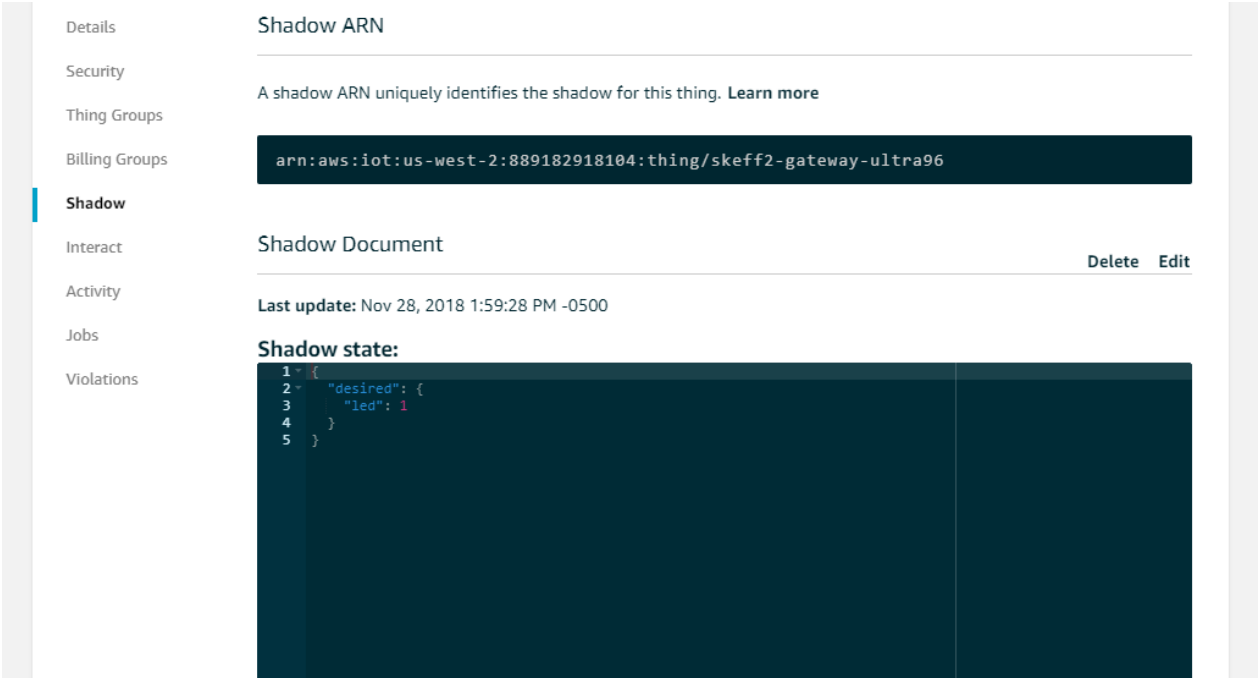
6. Ultra96 ボード (この制御アプリケーションのユニット コントローラー) を見て、インテリジェント I/O モジュールのエラー LED (ユーザー LED #1) が**消えて**いることを確認します。LED の定義は、次のとおりです。



7. MAX31855 のボードはそのままにして、熱電対のみを取りはずします。次の図を参考にしてください。



8. Ultra96 ボードを見て、インテリジェント I/O モジュールのエラー LED (ユーザー LED #1) が点灯していること、そしてデバイス シャドウがエラー ステート (「led」の値が 1) であることを確認します。



> **この動作の詳細**: アプリケーションが**開回路条件**を検出し、AWS Greengrass コアのデバイス シャドウを直接更新し、LED の **Desired State** を 1 にセットします。AWS Greengrass コアは、LED を点灯させたら自身のデバイス シャドウを **Reported State** で更新します。これを AWS IoT コアのコンソールで確認できるのはなぜか。それは、AWS Greengrass がエッジとクラウドの間でデバイス シャドウを自動的に同期しているためです。

このクラウド データ ポイントを使用して、アセット オーナーやフィールド エンジニアに電子メールやショート メッセージなどの通知を自動送信することもできます。

9. 熱電対モジュールを元に戻すとインテリジェント I/O モジュールのエラー LED が消えて、デバイス シャドウのヘルス ビットがクリアされます。

AWS IoT コアとの接続インジケーター

テレメトリを AWS クラウドに伝搬するには、AWS Greengrass コアから AWS IoT コアへの接続が必要です。必要な通信が確立されているかどうかを示すインジケーターが I/O コントローラーにあると便利です。

このセクションでは、Lifecycle Management と呼ばれる AWS IoT コアの機能を利用します。これは、2 つの特別なトピックを使用して接続情報を関心のあるコンシューマーに送信します。AWS Greengrass コアから AWS IoT コアへの接続の有無を検出できるように、いくつかの設定を行います。次に、接続が確立したらエッジ側のコアがデバイスの LED を点灯させるように、コアのデバイス シャドウを設定します。

一連のイベントは、次のとおりです。

1. AWS Greengrass が AWS IoT コアに接続します。
2. AWS IoT コアが、次の 2 つの Lifecycle Management トピックへのデータを生成します。
   - `$aws/events/presence/connected/+`
   - `$aws/events/presence/disconnected/+`
3. これらのトピックをリッスンして AWS Lambda 関数を呼び出す 2 つの AWS IoT ルールを作成します ([Actions] 欄に表示)。
4. AWS Lambda 関数が証明書 ID (プリンシパル) を利用して、関係する Thing (モノ) を「逆引き参照」します。
5. 接続または切断イベントに応じて AWS Lambda 関数がデバイス シャドウを適切に設定します。

この機能を実装するには、次の手順を実行します。

1. Ultra96 で、cloud/scripts ディレクトリに移動します。

```
cd $HOME/aws-cloud-and-xilinx-workshop/cloud/script
```

2. 次に、ライフサイクル設定をデプロイするスクリプトを実行します。

```
./deploy-lifecycle-handler.sh <prefix>
```

3. LED 機能をテストします。次のコマンドを実行して、AWS Greengrass を停止します。

```
sudo /greengrass/ggc/core/greengrassd stop
```

4. 次に、AWS Greengrass を起動します。

   `sudo /greengrass/ggc/core/greengrassd start` AWS IoT の接続を示す LED が点灯します。AWS Greengrass の停止/起動を繰り返して LED の状態を確認してください。

まとめ

この演習では、ライブ制御センサーの読み値を AWS クラウドに送信すると同時に、Amazon FreeRTOS の MQTT 通信メカニズムによってこれらのデータがダッシュボードに表示されるのを見てきました。次に、センサーにエラーを発生させると、MicroZed デバイス シャドウがエラー ステートに変化しました。そしてこれがユニット コントローラーと AWS クラウドによって自動検出され、アラートが生成されました。

次の演習

トップ ページ