

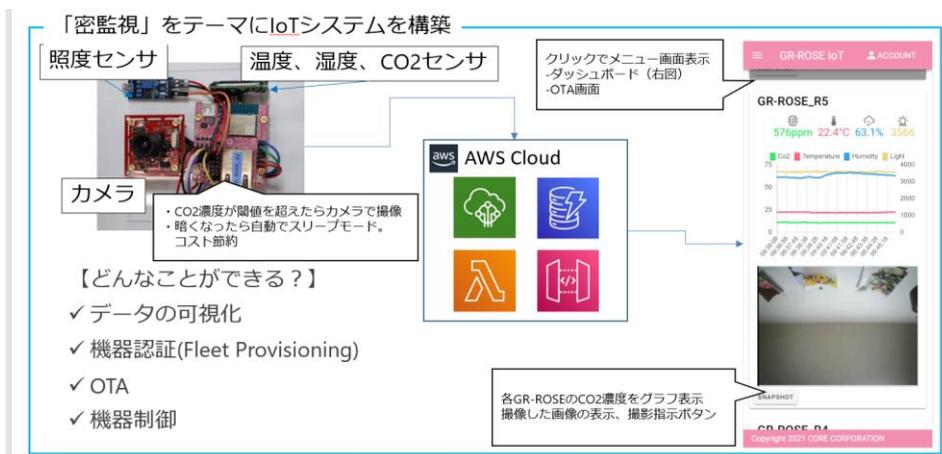
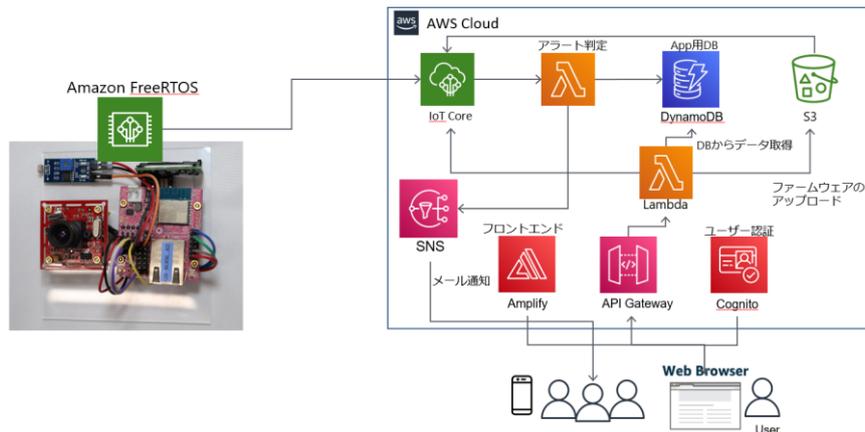
必見！AWS IoT 組み込み開発のノウハウを伝授します

株式会社コア

本記事では、株式会社コア製 [GR-ROSE](#) と、AWS IoT Core や Amazon S3、Amazon SNS、AWS Amplify、Amazon Cognito など様々な AWS サービスを組み合わせた完成度の高いデモを紹介します。このデモでは、複数台の IoT デバイスを管理するプロビジョニング手法に加えて、Wi-Fi 通信による AWS クラウドからの個々のデバイス制御や、異常検知アラート時のカメラ撮影機能、Amazon S3 を活用して後付けで機能更新/追加できる所謂ファームウェアアップデート(OTA)機能、管理者スマートフォンへのメール通知機能など、AWS IoT サービスを余すことなく活用した CO2 検知デモの実装事例です。

一般的に IoT デバイスには Raspberry Pi 等の Linux 搭載の組み込み用システムを活用しますが、昨今、組み込み機器市場からのリアルな声として、より廉価で、しかも堅牢かつセキュアな IoT デバイスが強く求められています。そこで産業・インフラ分野にも採用されるマイクロコントローラ(ルネサスエレクトロニクス製 RX65N)に AWS 製 リアルタイム OS である FreeRTOS を搭載し、AWS の各種サービスを簡単に利用することが可能な GR-ROSE をご紹介します。

本記事で紹介するデモの構築手順については、[こちら](#)をご覧ください



IoT デバイスのあるべき姿

巷のガジェット系 IoT 製品を使えば、誰でも気軽に、簡単に IoT を始められるようになりました。PoC を作る場合、それらの利用は選択肢として非常に魅力的であると思います。しかし、実際の製品開発を見据えた場合、品質要件を十分に満たすことができず、かといって0からデバイス開発をする余力もない、結果的に PoC 止まりで終わってしまう、という事態も起こりうるかと思います。

本記事で紹介する GR-ROSE+AWS のデモは、実際に IoT システムに必要なエッセンスをふんだんに盛り込んでいます。なかでも、

- ・フリートプロビジョニングによる機器認証
- ・証明書、秘密鍵のセキュアな管理
- ・FreeRTOS OTA によるソフトウェアのアップデート

は、デモの大きな特徴で、ガジェット系 IoT 製品にはない特徴かと思います。本記事では、これらのデモ実装例を公開し、さらにソースコードも公開しています。デモの構築手順を通して、本格的な IoT デバイスの開発フローを体感いただき、お客様の製品づくりに活かしていただければと思います。

デモで使用しているサービス

最初に、このデモで用いている代表的な AWS サービスをいくつか紹介します。

まず、最も特長的なのは AWS IoT Core で提供されている機能である『フリートプロビジョニング (Fleet Provisioning)』を使用している点です。この機能は、複数台ある個々の IoT デバイスに対して、工場出荷時に埋め込むアイデンティティを簡単に実装できる便利な機能です。

フリートプロビジョニングの詳細な説明は、こちらの [AWS ブログ](#) もご参照ください。

また、本デモの特徴として、AWS Amplify を使用してシステムを構築している点が挙げられます。AWS Amplify は、フロントエンドに提供される API 群、バックエンドのリソース作成、ホスティングなど、Web サービスを CLI から簡単に構築、デプロイすることができるサービスです。AWS Amplify を利用することで、AWS にバックエンドの構築を任せられることができるため、開発者はアプリケーションの開発に注力することができます。

その他、Amazon Cognito を使用してユーザ認証機能も実装しています。Amazon Cognito は、AWS Amplify と連携させることで、ユーザのサインアップ、サインイン機能など、あらかじめ用意された UI 画面を利用することができ、すぐにでも認証機能の実装可能です。また、AWS の各リソースに対するセキュアなアクセス (認証済みユーザのみが S3 などにアクセス可能とする仕組み) についても、簡単に実現する仕組みが提供されているため、開発者の負担を大幅に減らすことができます。

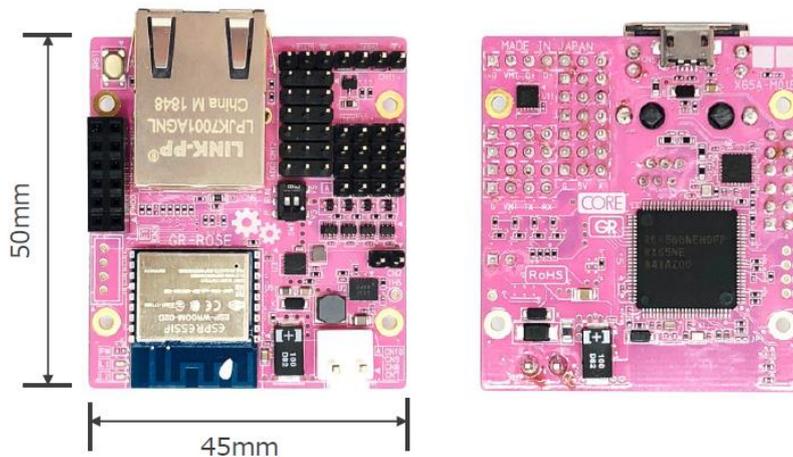
デモで使用しているサービスは以下表にまとめています。

デモで使用している AWS サービス

AWS IoT Core	端末を AWS に接続するためのサービス 端末(モノ)の登録や、証明書の管理をし、端末からのデータ受信をトリガに他のサービス(例えば lambda)を起動することが可能
AWS Lambda	サーバーレスのコンピューティングサービス デモでは、AWS IoT Core から受け取ったデータの Amazon Dynamo DB への格納、端末の接続、切断イベントを処理する用途で使用
Amazon SNS	メッセージ配信機能を提供したり、メッセージの受信者を管理したりするサービス。デモでは、CO2 センサが閾値を超えた場合に、Amazon SNS を通じてメールを配信している
Amazon Dynamo DB	NoSQL のデータベース。デモでは、センシングしたデータの格納や、端末の状態を管理するために使用
Amazon S3	ストレージサービス。デモでは、OTA 用のファームウェアや、カメラで撮像した jpeg 画像を保存している
AWS Amplify	バックエンドの自動構築機能、ホスティング、フロントエンドに提供される API で構成されるサービス。デモでは、S3 へのアクセス、デモとの連携、ホスティングで使用
Amazon API Gateway	API の作成、公開、保守、モニタリング、保護を行うことができるサービス。デモでは Amazon DynamoDB のデータにアクセスや、OTA ジョブの作成等で利用
Amazon Cognito	アプリケーションで利用されるユーザ管理機能を提供するサービス。デモでは、ユーザのサインイン、サインアウト機能で利用

GR-ROSE のご紹介

デモのエッジデバイスとして利用している GR-ROSE について、その特徴を簡単に説明します。



GR-ROSE は、ルネサスエレクトロニクス製 32bit MCU の RX65N 搭載した超小型 IoT プロトタイプングボードです。ネットワークのインターフェースとして Ethernet ,Wi-Fi に対応しているため、I2C,SPI,ADC などのインターフェースから取得したセンサ情報を AWS などのクラウドにアップロードすることができます。また、センシングだけでなく、シリアルサーボ制御用のインターフェースも持つため、ロボットの制御など、あらゆる用途にお使いいただける組み込みボードです。

「GR-ROSE に搭載しているマイコンは、ルネサス エレクトロニクス株式会社製 RX ファミリ [RX65N](#) です。このマイコンを選んだ理由としては、IoT 機器に必要な不可欠な機能や性能だけでなく、セキュリティ対策まで、全てこの 1 チップで満たせるためです。

最大のポイントは、絶対に外部に漏れてはいけな秘密鍵や証明書 (AWS クラウドに接続する際に必要) を安全に格納できる点です。

巷の市販 IoT キットや他の汎用マイコンを使うケースでは、万が一、秘密鍵の管理が不十分だと、漏洩のリスクがあります。一方で、RX65N は『[Trusted Secure IP \(以下、TSIP\)](#)』と言うセキュリティ H/W がチップに内蔵されているため、強固でかつ安全に秘密鍵を守ることができます。

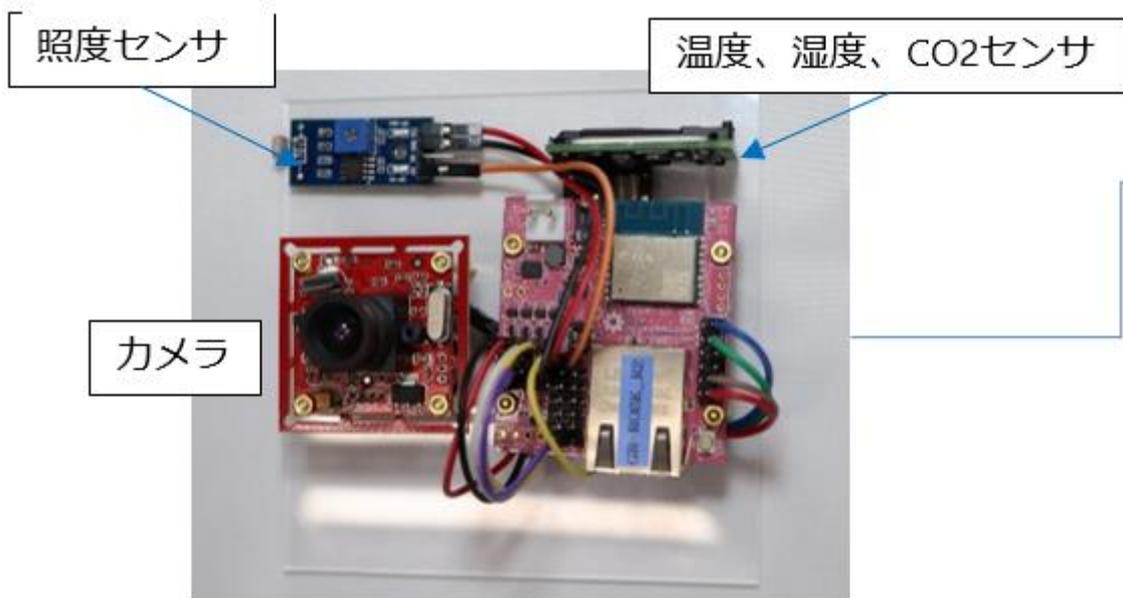
セキュリティ対策については、ハードルの高さや知識不足も相まってどうしても疎かになりがちですが、プロジェクトを企画する初期段階から最適なマイコンを選択することが、IoT デバイスを安全かつスムーズに運用するための成功のカギとなります。

ルネサス製 RX ファミリは、日本国内の採用実績が豊富な点も安心材料の1つですね。」

(株式会社コア エンベデッドソリューションカンパニー 営業統括部 ソリューション担当/利根川氏コメント)

デモのご紹介

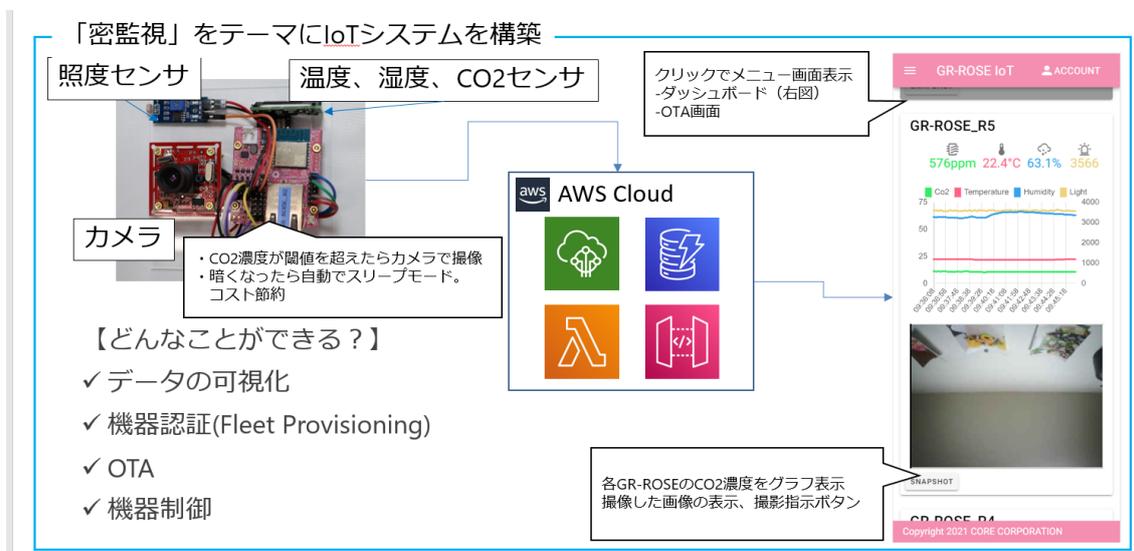
今回のデモでは、GR-ROSE と下記の温度・湿度・CO2・照度・カメラセンサを制御します。



デモで使用するセンサ類

搭載センサ	メーカー	仕様・機能	参照 URL
SCD 30	Sensirion	I2C で接続 温度、湿度、CO2 の計測	https://www.sensirion.com/jp/environmental-sensors/humidity-sensors/co2-sensor/
LM393 感光性センサ	HiLetgo	ADC で光量を計測	https://www.amazon.co.jp/gp/product/B010GX9DHQ/ref=ppx_yo_dt_b_asin_title_o03_s00?ie=UTF8&psc=1
Grove シリアルカメラ	Seeed Studio	UART で接続 撮像、JPEG データの出力	https://akizukidenshi.com/catalog/g/gM-09161/

これらのセンサ類を使用し、以下のデモを構築しました。



< デモ機で実現できること (IoT 製品への適用事例) >

- ① フリートプロビジョニングにより機器の認証
- ② CO2、温度、湿度のセンサデータを定期的に AWS にパブリッシュ
- ③ CO2 が閾値を超えた場合、「密」と判断して周囲の状況を撮影し、ブラウザ上に表示
- ④ CO2 が閾値を超えた場合、あらかじめ登録されたメールアドレスにアラーム通知メールを送信
- ⑤ センサデータのパブリッシュ周期、CO2 センサの閾値などをブラウザ上から設定
- ⑥ ブラウザ上の操作で、端末にカメラ撮影を指示し、撮影画像を表示
- ⑦ ブラウザ画面から指定した端末のソフトウェアアップデート実行

※デモ動画のリンク

<https://www.core.co.jp/service/iot/gr-rose/dl/gr-roseawssol>

これらのなかで、

- ・ フリートプロビジョニング
- ・ センサデータ(CO2、温度、湿度)のパブリッシュ
- ・ 端末の制御(Shadow)
- ・ OTA

について、[Getting Started with the GR ROSE](#) の構築手順を説明しています。ぜひお手元に GR-ROSE をご用意いただき、デモの構築にチャレンジしてください。

※センサ、カメラがご用意できない場合でも、デモを動作させることができます。

フリートプロビジョニング

デモでは、フリートプロビジョニングによる機器認証機能を実装しました。プロビジョニングとは、AWS IoT Core とデバイスを接続するために必要な作業で、デバイス・AWS 間のセキュアな通信を実現するための証明書、秘密鍵の作成、それらの AWS IoT Core への登録、リソースへのアクセス許可 (IoT ポリシー) の関連付け、を行います。これらの作業はデバイスの製造時に行う必要がありますが、多数のデバイスを量産する場合、それらを手動で作業するのはコスト的に大きな負担となってしまいます。

フリートプロビジョニングを利用することで、この問題を解決することができます。フリートプロビジョニングでは、デバイス製造時に、共通の証明書、秘密鍵を書き込みます。デバイスは AWS との初回接続時にシリアルナンバー、MAC などを送信し、事前に承認済みのシリアルナンバーであった場合は認証 OK とし、新たに生成した証明書、秘密鍵をデバイスに送信します。以降は、デバイスはその証明書、秘密鍵を用いて通信を行います。このように、製造時のプロビジョニングの作業コストを大幅に削減する事ができ、さらに不正なデバイスによるアクセス、認証も防止することができる非常に便利なサービスです。

TSIP(Trusted Secure IP)

デモでは、RX65N に搭載されている TSIP を使用して、鍵の管理、フリートプロビジョニングによる証明書、鍵データの更新を行います。フリートプロビジョニングでは、製造時に全デバイスで共通の証明書、秘密鍵を書き込むため、これらの漏洩が大きなセキュリティリスクとなります。

TSIP を使用することで、フリートプロビジョニングに使用する証明書、秘密鍵を暗号化した形で ROM に保存することができるため、デバイスに対して物理的にアクセスされた場合でも、証明書、鍵情報の漏洩を防ぐことができます。そのセキュリティ実装の強固さは NIST(米国標準技術研究所) FIPS140-2 CMVP という暗号モジュール評価試験においてレベル 3 を達成しており、一般的な IoT デバイスのセキュリティ要件を十分に満たす水準です。また、TSIP は RX65N に代表される RX ファミリー全般に内蔵されており、ワンチップでセキュリティ実装が完結でき、IoT デバイスの製造コスト低減にも寄与します。

TSIP の詳細については、

<https://www.renesas.com/jp/ja/document/apn/rx-family-tsip-trusted-secure-ip-module-firmware-integration-technology-binary-version?language=ja>

を参照してください。

まとめ

本記事では、GR-ROSE+AWS IoT のデモについて実装事例をご紹介します、[こちら](#)でデモの構築方法を説明いたしました。デモの構築を通して、センサデータのパブリッシュはもちろん、TSIP+フリートプロビジョニングによる証明書、鍵データのセキュアな管理、FreeRTOS OTA によるファームウェアのアップデート、Shadow による端末の制御など、AWS IoT の機能をフルに活用し、本格的な IoT 端末の実装、動作の確認ができたと思います。

ここでは、端末にフォーカスして環境の構築手順を説明しましたが、もちろん、IoT システムを構築するには、バックエンド、フロントエンドの開発も必要になります。この記事をご覧になり、クラウド側の開発についてご興味のある方、ご検討されている方はぜひとも弊社までお声がけいただければと思います。

コア X AWS IoT

弊社では、ここでご紹介したサービス、アーキテクチャを AWS IoT におけるリファレンス・デザインとして定義し、実際の開発案件におけるベースのデザインとして利用しています。

弊社は、もともと組込機器開発の分野を強みとしていますので、もちろん端末側の開発には精通してはいますが、今回ご紹介したデモ事例のように、AWS クラウドのシステム構築についてもノウハウを蓄積し、実際にいくつかのお客様のシステム開発・運用をしております。端末(HW/SW)+クラウド まで、IoT に必要な要素、すべてを一貫して手掛けることができるのが、弊社の強みです。IoT の導入をご検討のお客様、PoC→製品レベルへのステップアップをご検討のお客様、ぜひとも弊社にお声掛けいただければと思います。お問合せは、株式会社コア エンベデッドソリューションカンパニー営業統括部 鈴木 まで、お気軽にご連絡ください。

お問い合わせ先

株式会社コア エンベデッドソリューションカンパニー営業統括部 鈴木沙織

esc-biz@core.co.jp

参考 URL

[フリートプロビジョニングを用いて、IoT デバイスと AWS IoT Core の初期セットアップを自動化する方法 | Amazon Web Services ブログ](#)