Getting Started with the GR-ROSE

株式会社コア

本説明書では、GR-ROSEを使った AWSIoT デモ構築手順を説明します。デモでは、

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

の動作をご覧いただくことができます。

目次

1	デモ環境	2
1.1	GR-ROSE について	4
1.2	TSIP について	4
1.3	AWS アカウントについて	5
2	デモ環境構築手順	6
2.1	GR-ROSE ビルド環境	6
2.2	TSIP のサンプルプロジェクトの展開	7
2.3	ルート CA 証明書、クライアント証明書、鍵ペアのダウンロード、バイナリ化	8
2.4	秘密鍵の暗号化、バイナリ化	10
2.5	ビルド、プログラムの書き込み	12
2.6	フリートプロビジョニングの準備	12
2.7	AWS との接続	16
2.8	フリートプロビジョニング	18
2.9	センサデータのパブリッシュ	
2.10	Shadow	19
2.11	OTA	21

1 デモ環境

本デモを実行するには、以下のご準備が必要です。

必須

- WindowsPC (e²stduio 用)
- ・ WSL or Git Bash(TSIP 鍵情報生成用シェルスクリプト実行用)
- ・ Teraterm 等のターミナルソフト
- GR-ROSE
- ・ GR-ROSEとPC 接続用 USB ケーブル(Type-A/micro-B)
- 無線 LAN ルータ
- ・ AWS アカウント

デモの全機能を動かす場合に必要なもの

※センサをご用意いただかなくても、デモの動作を確認することができます

- ・ センサを接続するためのジャンパワイヤ(オス-メス)
- 2.54mm ピッチのピンヘッダ
 - ※GR-ROSEの以下赤枠部分に半田付けしてください。



以下のセンサ類

SCD 30	https://www.sensirion.com/jp/environmental-sensors/humidity-
	sensors/co2-sensor/
LM393 感光性センサ	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	https://akizukidenshi.com/catalog/g/gM-09161/



<u>こちら</u>に GR-ROSE のソースコードを公開していますので、ダウンロードして、以下手順どおりにデモ環境を 構築してください。

なお、デモに使用するセンサ類がお手元に用意できない場合でも、GR-ROSE 単体でデモの動作確認が 可能です(その場合は、ダミーデータを AWS にパブリッシュします)。

1.1 GR-ROSE について

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、その特長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 プロトタイピ 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。 回路図等、スペックに関する情報は、<u>こちら</u>をご参照ください。

1.2 TSIP 12017

本デモでは、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鍵 データの更新を行います。TSIPを使用することで、鍵データが暗号化された形で ROM に保存され、 デバイスに対して物理的にアクセスされた場合でも、鍵情報の漏洩を防ぐことができます。 TSIPの詳細については、

<u>https://www.renesas.com/jp/ja/document/apn/rx-family-tsip-trusted-secure-ip-module-</u> <u>firmware-integration-technology-binary-version?language=ja</u> を参照してください。

1.3 AWS **アカウントについて**

※AWS のアカウントをお持ちではない方は、AWS のドキュメントを参考にアカウントの作成を行ってください。

https://aws.amazon.com/jp/premiumsupport/knowledge-center/create-and-activateaws-account/

AWS のコンソールを利用する IAM ユーザーの作成については、AWS のドキュメントを参考に作成を 行ってください。

<u>https://docs.aws.amazon.com/ja_jp/IAM/latest/UserGuide/id_users_create.html</u> AWS IoT Core 及び FreeRTOS を利用するための権限として以下の 2 つの権限が IAM ユーザー に必要となります。

- · AmazonFreeRTOSFullAccess
- · AWSIoTFullAccess

IAM ユーザーの作成及び、権限の付与については、FreeRTOS のドキュメントの「Setting up your AWS account and permissions」で紹介されておりますので、その手順を参考に進めてください。 <u>https://docs.aws.amazon.com/freertos/latest/userguide/freertos-prereqs.html</u> 注意)この資料に含まれる IoT Policy やセキュリティーの設定は開発用途向けですので、本番環 境での利用では「AWS IoT Core policy examples」や、「Security best practices in AWS IoT Core」を参考にしてください。

https://docs.aws.amazon.com/iot/latest/developerguide/example-iot-policies.html https://docs.aws.amazon.com/iot/latest/developerguide/security-best-practices.html

2 デモ環境構築手順

2.1 GR-ROSE ビルド環境

e2studio のインストール

https://www.renesas.com/jp/ja/software-tool/e-studio

の「統合開発環境 e² studio 2021-07 Windows 用インストーラ」をダウンロードしてインストール RX マイコンのツールチェインをインストールしてください

インストール手順詳細:

https://www.renesas.com/jp/ja/document/man/e-studio-integrated-developmentenvironment-users-manual-getting-started-guide-rx-rl78-rh850-family?r=1175496

インストール後、ダウンロードした GR-ROSE_demo/amazon-freertos.zip を任意の場所に展開して ください。上記「3.3 ワークスペースへの既存プロジェクトのインポート」を参照し、「(3) e2 studio の [ファイル(F)] → [インポート(I)...]を選びます。」で、ルートディレクトリの選択->amazonfreertos¥projects¥renesas¥rx65n-gr-rose-iot¥e2studio-gcc¥aws demos を選択してください。



インポート後、プロジェクト->プロジェクトのビルドで aws_demos をビルドしてください。 ※インストールしたツールチェインのバージョンによってはビルド時にエラーが発生することがあります。そ の場合は、プロジェクト->プロバティ-> C/C++ビルド->設定->Toolchain で、バージョンをインストー ルしたツールチェインのバージョンに設定してください



2.2 TSIP **のサンプルプロジェクトの展開**

ダウンロードした GR-ROSE_demo/r20an0548jj0112-lib-rx-tsip-security.zip を任意のフォルダに 展開してください。 2.3 ルート CA 証明書、クライアント証明書、鍵ペアのダウンロード、バイナリ化

AWS マネジメントコンソール上でのルート CA 証明書、クライアント証明書、鍵ペアのダウンロード保 方法について説明します。まず、AWS IoT Core のページを開き、下図赤枠の証明書をクリックして ください。

※本書で使用している AWS マネジメントコンソールのキャプチャは、2021/10/5 時点のものです。



作成ボタンを押すと、以下画面が表示されます。1-Click 証明書作成(推奨)の「証明書を作成」 をクリックしてください



モノの証明書(クライアント証明書)、パブリックキー、プライベートキー、AWS IoT のルート CA をダウ ンロードしてください AWS IoT のルート CA は、別ページに遷移するので、ページ内の RSA 2048 bit key: Amazon Root CA 1.

を開き、AmazonRootCA1.pemの名前で保存してください



ダウンロードしたルート CA 証明書(AmazonRootCA1.pem)は、

r20an0548jj0112-lib-rx-tsip-security

¥FITDemos¥rx_tsip_freertos_mbedtls_sample¥key_crt_sig_generator¥ca

に格納してください

ダウンロードしたクライアント証明書(xxxxx-certificate.pem.crt)、パブリックキー(xxxxx-

public.pem.key)、プライベートキー(xxxxx-private.pem.key)は、

r20an0548jj0112-lib-rx-tsip-security

¥FITDemos¥rx_tsip_freertos_mbedtls_sample¥key_crt_sig_generator¥client-rsa2048 に格納してください

最後に、有効化を押し、完了を押してください。

次に、クライアント証明書を DER 形式に変換します

r20an0548jj0112-lib-rx-tsip-security

¥FITDemos¥rx_tsip_freertos_mbedtls_sample¥key_crt_sig_generator ¥key_crt_sig_generator フォルダ内の 1_rsa2048_convertCrt.sh をWSL もしくは Git Bash で実行してください。

output フォルダに生成された

- ✓ client_rsa2048_crt_array.txt
- ✓ AmazonRootCA1_crt_array.txt
- ✓ AmazonRootCA1_sig_array.txt

を

amazon-freertos¥vendors¥renesas¥boards¥rx65n-gr-rose-iotgcc¥aws_demos¥application_code

に格納してください(上書きで OK)

2.4 秘密鍵の暗号化、バイナリ化

key_crt_sig_generator フォルダ内の 3_showkeyValues.sh を実行してください。コンソールに、

Root CA signature RSA-2048bit Public:↩
ADR9EAFEF54741FCD43331398A217548800010001_004
Client RSA-2048bit All:↔

D16E760EFC7752A45A9F296EE0E3879D2BA5A80716298A0FEE862CEC8376372A558C4D21-e e

のように表示されます。

r20an0548jj0112-lib-rx-tsip-security¥tool¥ Renesas Secure Flash Programmer.exe を実行 してください。GR-ROSE に埋め込む鍵情報の C 言語ソースファイルを作成します

- ① Key Wrap タブを選択し、Select MCU で TSIP(DF Memory 32KB)を選択してください
- ② Key Type で「RSA-2048bit Public」を選択し、コンソールの上記赤枠部分をコピーし、Key Data に貼り付けてください。ただし、末尾の-e は不要です。Register を押下してください
- ③ Key Type で「RSA-2048bit All」を選択し、コンソールの上記青枠部分をコピーし、Key Data に貼り付けてください。ただし、末尾の-e は不要です。Register を押下してください
- ④ Key Type で「AES-128bit」を選択し、KeyData に
 11111112222222233333334444444 を入力してください。。Register を押下してください
- ⑤ Key Type で「AES-128bit」を選択し、KeyData に
 5555555666666666777777788888888 を入力してください。。Register を押下してください
- ⑥ Key Type で「Update Key Ring」を選択し、KeyData に
 11111111222222223333333444444555555556666666667777777888888888 を入力し
 てください。Register を押下してください
- ⑦ provisioning key の provisioning key File Path に amazon-freertos¥vendors¥renesas¥boards¥rx65n-gr-rose-iotgcc¥aws_demos¥application_code¥ provisioning.key を選択してください
- ⑧ provisioning key の encrypted provisioning key File Path に amazon-freertos¥vendors¥renesas¥boards¥rx65n-gr-rose-iotgcc¥aws_demos¥application_code¥provisioning.key_enc.key を選択してください
- ④ Generate Key Files を押下してください。key_data.c, key_data.h の保存先を amazon-freertos¥vendors¥renesas¥boards¥rx65n-gr-rose-iotgcc¥aws_demos¥application_code としてください(上書きOK)。

visioning kev Key Wra	P Firm Upda	ate				
MCU	i in opa					
Select MCU TSIP(DF Memory 3	2KB) 🗸				
(ey Setting						
Key Type Update Key	Ring 🗸	Key Data 11111111222222233333334444444455555555666666666777	77777888888888		Register	
Кеу Туре	Key Da	ata			Delete	
AES-128bit	11111	112222222333333344444444				
AES-128bit DSA-2049bit Dublic	555555 DE977	155666666667777777888888888 1000560D169900D03799599D0666D696609500D00959C0045909955C4	ACE40007004EEEE0	FOAD		
NOM-2040DIL FUDIIC	D00//	IBUUBUDDHUBBUMDMA/DAEOZUDMUUDUBUUMAEDBDUUBEBOUBHF0BZUZEFOH	9UE93ZZZ009EE3833	E 0 4 Z		
RSA-2048bit Public	EDA4E	C98610E6DA43EA87E1A7EA133D76D3A22E575D4BAEBDAD52523378429	7583C2C0167C56B	F5F4		
RSA-2048bit Public RSA-2048bit Private	EDA4E EDA4E	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378423 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378423	97583C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA48 EDA48 11111	3C98610E6DA49EA87F1A7EA133D76D3A22F575D4BAEBDAD5252337842 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD5252337842 112222222333333334444444555555556666666667777777888888888	97583C2C0167C56B 97583C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA48 EDA48 11111	3C98610E6DA49EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378425 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378425 112222222233333333444444455555555566666666677777777888888888	97583C2C0167C56B 97583C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA48 EDA48 11111	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378425 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378425 112222222233333333444444455555555566666666677777777888888888	97583C2C0167C56B 97583C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA48 EDA48 11111	3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222233333333444444455555555566666666677777777888888888	97589C2C0167C56B 97589C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA4E EDA4E 11111	3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222233333333444444455555555566666666677777777888888888	97583C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA4E EDA4E 11111	3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222233333333444444455555555566666666677777777888888888	97589C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring	EDA4E EDA4E 111111	3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3098610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222233333333444444455555555566666666677777777888888888	97583C2C0167C56B 97583C2C0167C56B	F5E4 F5E4		
RSA-2048bit Public RSA-2048bit Private Update Key Ring < provisioning key provisioning key File Pa	EDA4E EDA4E 11111	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 11222222223333333344444445555555566666666677777777888888888	97589C2C0167C56B 97589C2C0167C56B Browse	>		
RSA-2048bit Public RSA-2048bit Private Update Key Ring vovisioning key provisioning key File Pa encrypted provisioning key	EDA4E EDA4E 111111 th ey File Path	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 11222222223333333344444445555555566666666677777777888888888 x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key_enckby	97589C2C0167C56B 97589C2C0167C56B Browse Browse	>		
RSA-2048bit Public RSA-2048bit Private Update Key Ring « rovisioning key rovisioning key File Pa incrypted provisioning k	EDA4E EDA4E 111111 th ey File Path	C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222333333334444444555555556666666667777777788888888 x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key_ gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key_enc kby [gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key_enc kby]	97583C2C0167C56B 97583C2C0167C56B Browse Browse	>		
RSA-2048bit Public RSA-2048bit Private Update Key Ring « rovisioning key rovisioning key File Pa Incrypted provisioning key V (16 byte hex / 32 cha	EDA4E EDA4E 111111 th ey File Path aracters)	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222333333334444444555555556666666667777777788888888 *x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key_enc kby [Random)	97583C2C0167C56B 97583C2C0167C56B Browse Browse	>		
RSA-2048bit Public RSA-2048bit Private Update Key Ring vovisioning key provisioning key File Pa encrypted provisioning key V (16 byte hex / 32 cha	EDA4E EDA4E 111111 th ey File Path aracters)	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222333333334444444555555556666666667777777788888888 *x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning.key gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning.key_enc.kby (Random)	97583C2C0167C56B 97583C2C0167C56B Browse Browse	>		
RSA-2048bit Public RSA-2048bit Private Update Key Ring vovisioning key provisioning key encrypted provisioning key V (16 byte hex / 32 cha	EDA4E EDA4E 111111 th ey File Path aracters)	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222333333334444444555555556666666667777777788888888 x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning.key gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning.key_enc.kby (Random)	97583C2C0167C56B 97583C2C0167C56B Browse Browse	F5E4 F5E4	Key Files	
RSA-2048bit Public RSA-2048bit Private Update Key Ring vovisioning key rovisioning key rovisioning key File Pa encrypted provisioning k V (16 byte hex / 32 cha	EDA4E EDA4E 111111 th .ey File Path aracters)	C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222333333334444444555555556666666667777777788888888 x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key_enc kþy (Random)	97583C2C0167C56B 97583C2C0167C56B Browse Browse	F5E4 F5E4 >	Key Files	
RSA-2048bit Public RSA-2048bit Private Update Key Ring orovisioning key orovisioning key orovisioning key File Pa encrypted provisioning k V (16 byte hex / 32 cha	EDA4E EDA4E 111111 th .ey File Path aracters)	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 112222222333333344444445555555566666666667777777788888888 x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key (Random)	97583C2C0167C56B 97583C2C0167C56B Browse Browse Ge	F5E4 F5E4 >	Key Files.	
RSA-2048bit Public RSA-2048bit Private Update Key Ring vovisioning key rovisioning key File Pa morypted provisioning k V (16 byte hex / 32 cha enerate succeeded.	EDA48 EDA48 111111 th .ey File Path aracters)	3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 3C98610E6DA43EA87F1A7EA133D76D3A22F575D4BAEBDAD52523378429 1122222223333333444444455555555666666666677777777888888888 *x65n-gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key gr-rose-iot-gcc¥aws_demos¥application_code¥provisioning key (Random)	77583C2C0167C56B 77583C2C0167C56B Browse Browse Ge	F5E4 F5E4 >	Key Files.	

<u>※手順⑦、⑧で指定する provisioning.key, provisioning.key_enc.key は、本来ユーザ自身でご</u> <u>用意いただく必要があります。本環境では構築の手順を簡単にするため、サンプルプロジェクトに追</u> <u>加しておりますが、ご自身で provisionig.key、provisioning.key_enc.key を用意される場合は、</u> <u>r20an0548jj0112-lib-rx-tsip-security¥reference_documents¥ja¥r01an5792jj0101-rx-tsip.pdf</u> <u>をご参照ください</u>

2.5 ビルド、プログラムの書き込み

e2studio で aws_demos プロジェクトのビルドを実行してください。ビルドに成功すると、amazonfreertos¥projects¥renesas¥rx65n-gr-rose-iot¥e2studio-gcc¥aws_demos¥ReleaseBin フォル ダに、aws_demos.bin が生成されます。PC と GR-ROSE を USB で接続し、GR-ROSE のリセットス イッチ(下図赤枠)を押してください。



しばらくすると LED の赤、緑が点灯し、PC 上でストレージ: GR-ROSE としてアクセスすることができます。aws_demos.bin をドラッグ&ドロップで転送してください。プログラムの更新が始まります。LED の点滅->消灯したらアップデート完了です。

2.6 フリートプロビジョニングの準備

まず、ポリシーを作成します。AWS マネジメントコンソール上で、以下のように安全性 -> ポリシーを 開き、作成を押してください。

AWS IOT \times	(① 教人 があったコンソールエクスペリエンスのご知う コンソールエクスペリエンスの実践的れています。計画はこちらつ のいハエクスペリエンスを知道にいたます。計画はこちらつ のいハエクスペリエンスを知道にいただき、ご <u>意見を知識がいください。</u> 新しハエクスペリエンスはナビゲーションメニューが与知識にできます。	×
モニタリング アクティビティ ▼ Connect	avistor > ポリシー ポリシー	ňa
Get started Fleet provisioning templates ▼ 管理	17/5-08# Q	
徴要 モノ タイプ ≂ (∽ゲリーブ	0 EN	
モンのシルーン 請求グループ ジョブ ジョブテンプレート		
トンネル 保持されたメッセージ フリートメトリクス (1508)		
 フリートハブ Greengrass 		•••
 > ワイヤレス接続 ▼ 安全性 証明書 		
ポリシー み ロールエイリアス		

ポリシーの名前を aws-iot-demo とし、ポリシーを以下のように入力してください。アドバンスモードを クリックすることで、ポリシードキュメントを直接入力することができます。入力後、作成を押します。 ※<your-account-id>は AWS のアカウント ID に変更してください





次に、フリートプロビジョニングのテンプレートを作成します。

AWS マネジメントコンソール上で、以下のように Connect -> Fleet provisioning templates を選択 して、作成を押してください。

AWS IoT ×	⑥ 新しい AWS ht コンソールエクスペリエンスのご紹介 コンソールエクスペリエンスが受新されています。 詳細はこちら [2] 新しいエクスペリエンスをお試しいただき、ご意見をお聞かせください。新しいエクスペリエンスはナビグーションメニューから無効にできます。	×
モニタリング アクティビティ	AWS 167 〉 フリートのプロビジョニングテンプレート	_
 Connect Get started 	フリートのプロビジョニングテンプレート	
Fleet provisioning templates	テンプレートを検索 Q	
 マリートハブ 	各前 ステータス	
▶ Greengrass	有効 ***	
▶ ワイヤレス接続	有効 ***	
▶ 安全性	有功 ***	
▶ 防御	有効 ***	
ACT	有効 ***	
PTAN	有功 ***	

開始方法をクリックし、以下のように入力して、次へを押してください。

テンプレート名			
fleet-prov-template-grrose-tsip			
説明 (オプション)			
テンプレートを説明			
プロビジョニングロール このロールは、AWS IoT がユーザーに代わってリソース	へに対するアクセス許可を提供します。 詳細はこちら。		
fleet-provisioning-20210120-role 管理ポリシーが	アタッチされています 🗸	ロールの作成	選択
プロビジョニング前のフック (オプション) .ambda 関数を実行して、プロビジョニング前にデバイ	(スパラメータを検証します。詳細はこちら。	新しい Lambda	関数を作用
Lambda 関数が選択されていません			選択
オプション設定			
✓ AWS IoT レジストリを使用してデバイスフリートを フリートのデバイスは、AWS IoT レジストリでモノによ	注管理する よって表され、タイプ、グループ、その他の属性によって	分類されます。	

テンプレート名は、fleet-prov-template-grrose-tsip としてください。

プロビジョニングロールについては、「ロールの作成」を押し、作成してください。名前の指定はありません。



既存の AWS IoT ポリシーを使用する、を選択した状態で、先程作成した aws-iot-demo を選択し てください。ここで選択したポリシーがフリートプロビジョニング後にモノにアタッチされます。 次へを押し、以下のようにモノの情報を定義します。モノの名前プレフィックスを「GR-ROSE_」とし、そ の他は未入力でかまいません。

^{フリートのプロピジョニング} AWS IoT レジストリ設定を定義する
これらの設定は、プロビジョニングされた各デバイスで使用されます。
モノの名前プレフィックス モノの名前が自動的に作成され、最初の文字がこのプレフィックスで始まります。 GR-ROSE_
モノのタイプを適用する タイプは、同じ風性を共有するモノを識別するのに役立ちます。 モノのタイプ タイプが選択されていません - タイプの作成
グルレープを選択する モノのグループを使用すると、複数のデバイスを一度に管理できます。グループを使用して、ポリシーをアタッチし、ログ記録オプションを設定 し、ジョブを作成することができます。 モノのグループ
グループ / グループ の作成 変更
検索可能なモノの属性を定義する (オプション) レジストリ内のモノを検索できるように、これらの属性 (褒歎可) に値を入力してください。
居性キー 属性キー (メーカーなど)を指定します 値 属性値 (Acme-Corporation など)を指定します。 がのものを追加
キャンセル 戻る テンプレートを作成

テンプレートを作成、をクリックし、

2.3. ルート CA 証明書、クライアント証明書、鍵ペアのダウンロード、バイナリ化 で作成したクライアント証明書を選択してください。選択後、「ポリシーをアタッチ」を押してください。 最後に、テンプレートを有効化する、を押して完了です

^{フリートのプロピジョニング} デバイスをプロビジョニングするアクセス許可を証明書またはユーザーに付与する	
プロビジョニング証明書を持つデバイスに自身をオンボードする権限を与えることができます。または、IAM ロールを使用して、ユーザーにデバ イスをオンボードする権限を与えることもできます。詳細はこちら	C
プロビジョニングクレームを使用	
• 証明書を検索する • 証明書を検索する • □ <t< th=""><th>L .</th></t<>	L .

2.7 AWS との接続

teraterm 等、ターミナルソフトで GR-ROSE が接続されているポートを開いてください。USB-CDC 通信のため、ボーレート等の設定は不要です。

ターミナルにログが表示されたら、「ctrl+s」を入力し、setup mode(デモアプリケーションを起動せず、 コンソールからの入力のみを受けつける)へ移行してください。setup mode で起動すると、コンソール に以下のようなログが表示されます。



この状態で、

>default

と入力してください。RX65Nのデータフラッシュ領域がすべて初期化されます。

次に、以下を入力し、WiFiの設定を行います。

> param,set,socket,1,ssid,{接続する WifiAP の SSID}

> param,set,socket,1,password,{WifiAP のパスワード}

※SSID,パスワードは環境に合わせて入力してください

入力に成功すると、以下のようなログが表示されます。

\geq	param,set,socket,1,ssid,CORE
S	ocket[1] Gr=4 id=0x20148000(group:4 size:164 offset:0)
	SSID=CORE id=0x20040014(group:4 size:32 offset:20)
\geq	param,set,socket,1,password,CORE
S	ocket[1] Gr=4 id=0x20148000(group:4 size:164 offset:0)
	Password=CORE id=0x20040034(group:4 size:32 offset:52)

次に、端末のシリアル番号を入力します。以下コマンドを入力してください

> param,set,mqtt,serial_no,{シリアル番号(任意の文字列)}

ここで指定したシリアル番号と、フリートプロビジョニングテンプレートで指定した prefix =GR-ROSE_

からモノの名前およびクライアント ID がフリートプロビジョニング時に決定されます。

モノの名前

GR-ROSE_{シリアル番号}

最後に AWSIoTCore のエンドポイントを設定します。

>param,set,socket,1,url,XXXXXX-ats.iot.ap-northeast-1.amazonaws.com

IoTCoreのエンドポイントは、AWS マネジメントコンソール上の設定画面にデバイスデータエンドポイ

ントとして表示されます。

いドポイント

デバイスデータエンドポイント 情報 デバイスは、アカウントのデバイスデータエンドポイントを使用して AWS に接続できます。 С

各モノには、このエンドポイントで使用可能な REST API があります。MQTT クライアントと AWS IoT デバイス SDK 🖸 もこのエンドポイントを使用します。

-ats.iot.ap-northeast-1.amazonaws.com

SCD30などのセンサが用意できない場合

センサが用意できない場合は、以下コマンドでセンサを無効化することで、ダミーデータを送信することができます。

>param,set,sensor,3,connect,false SCD30の無効化 >param,set,application.camera.false カメラの無効化

Ctrl + dを押下後、端末が再起動し、Wifiに接続成功するとAWSに接続し、フリートプロビジョニン グを実行します。 2.8 フリートプロビジョニング

2.7 で端末を再起動すると、設定された Wifi のアクセスポイントに接続し、その後 AWS と接続します。 正常に接続されると、フリートプロビジョニングを行います。フリートプロビジョニングが成功すると、以下 のようなログが端末のターミナルに出力されます。



その後、再起動を行い、フリートプロビジョニングで AWS から取得したクライアント証明書、秘密鍵をも とに、AWS との接続を行います。接続後、センサデータのパブリッシュを開始します。

2.9 センサデータのパブリッシュ

フリートプロビジョニングで AWS から通知されたトピック名 = data/sensor/GR-ROSE_{シリアル番号} にセンサデータをパブリッシュします。

AWS マネジメントコンソール上、IoTCore->テスト->MQTT テストクライアントのトピックをサブスクライ ブする、で「トピックあのフィルター」に以下のようにトピックを入力し、サブスクライブを押してください。セ ンサデータがパブリッシュされると、図のように json 形式でセンサデータが表示されます。

MOTT テストクライアントを使用して、AV	
更やイベントを通知します。MQTT テスト	VS アカウントで渡される MQTT メッセージをモニタリングできます。デバイスは、トビックによって識別された クライアントを使用して MQTT メッセージトビックにサブスクライブし、トビックに MQTT メッセージを発行で
トビックをサブスクライブする	トビックに公開する
ト ビックのフィルター 情報 トピックフィルタは、サブスクライブするトビ	シックを認述します。トビックフィルタには、MQIT ワイルドカード文字を含めることができます。
data/sensor/GR-ROSE_1	
▶ 追加設定	
サブスクライブ	
サブスクリプション	data/sensor/GR-ROSE_1
data/sensor/GR-ROSE_1 ♥×	
• • •	
• •	▼ data/sensor/GR-ROSE_1

2.10 Shadow

本デモでは、Shadow の動作も確認いただけます。AWS マネジメントコンソール上、下図のように管理 ->モノ->Device Shadow-> shadowROSE を選択してください。

AWS IoT ×	⑥ 単しいやいコンワームとダイルリンスなど的 コンツームジタイルリンスが多たれてはす、前期にしておいたまたが、「日本のシーム」を見たいためで、単したセクスペリエンスはナビターションメニューから開始にできます。
モニタリング アクティビティ	ANS bit 〉世说 〉 E/ 〉 GH405L,1
► Connect	◎ MFをキュリティを含まりティエロスナフです mm MS MF Device Defender からのフレードに対する年目のチェックを有効にすることで、セキュリティ酸素を目的にします。転素では、ロとアクセスポリシーのチェックを知らとするセキュリティのペストプラのティスに基づいてい「放在を所図します。料金素を用る」と mMはごちらび
問題	GR-ROSE_1 ==
モノのグループ 請求グループ	モノの詳細
ジョブ ジョブテンプレート	名用 タイプ GAROSE_1 ・
トンネル 保持されたメッセージ フリートメトリクス (155)	Alter 語言のによる nothead Selection Sel
▶ フリートパプ ▶ Greengrass	歴史 延期書 モノルウガループ Decket Shadlow EXA(前 アウア・ビアィ フ ジョブ アラーム Defender Xトリウス
▶ ワイヤレス接続	Device Shadow (1) ms
▶ 安全性	Devel Balance Carl, Biller Development, Barl 2000 7 8 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
▶ ACT	Q. Doise Studier 10.7 cl. /915/J
▼ テスト	 ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・
▶ デバイスアドバイザー MQTT デストクライアント	0 shadew8058 SawUhirupuGR-R055L_10/ballow/sume/hadsw805g Oxtober 05, 2021, 14:31:47 (JTIC-0900)

本デモでは、Shadow の動作も確認いただけます。AWS マネジメントコンソール上、下図のように管理 ->モノ->Device Shadow-> shadowROSE を選択してください。編集を押し、

	Device shades ドキュメント MQTTトビック	
7		
5クループ 14 - プ	Device Shadow ドキュメント mm	likst.
v-2	Devis Stades ドキュメントには、デバイスの状態の構成体構成、指導体、デルタ時の自まれています。USEMPは、ことでまたはプログラムで構成できます。AMS-int Claid している構成、デバイスの状態を利用できます。	
テンプレート	Parales Chadey (Child)	
L	Preurice analysis of the second s	
れたメッセージ	states (
トメトリクス 📧	"destruct" (
トハブ	"240_749" 0, "140_green" 1,	
	"temp_threshold": 30,	
jan an a	"hwisity_threshold": 40, "oresput treshold": 1810.	
レス線統	"publish_interval": 60,	
	"Co2_Extreshold": 3900, "liet threadstof": 0	
	h	
	"reporter" ("tad red": e.	
	"led_green": 1,	
	"teng_threshold": 30, "handlift transhold": 60.	
マドバイザー	"pressure_threshold": 1090,	
テストクライアント	"publish_interval": 40, "col threshold": 5806.	
	"light_threshold": 0	
227		

Device Shadow の状態

1	d
2	"state": {
3	"desired": {
4	"led_red": 0,
5	"led_green": 1,
6	"temp_threshold": 30,
7	"humidity_threshold": 60,
8	"pressure_threshold": 1050,
9	"publish_interval": 60,
10	"co2_threshold": 3500,
11	"light_threshold": 0
12	},
13	"reported": {
14	"led_red": 0,
15	"led_green": 1,
16	"temp_threshold": 30,
17	"humidity_threshold": 60,
18	"pressure_threshold": 1050,
19	"publish_interval": 60,
20	"co2_threshold": 3500,
21	"light_threshold": 0
22	}
23	}
24	3

"state": { "desired": { の "led_red"、もしくは"led_green"を 0 もしくは 1 に変更してください。
0 の場合は LED が消灯し、1 の場合は LED が点灯します。更新を押すと shadow の変更が端末に
通知され、設定した LED 状態が GR-ROSE に反映されます。



2.11 OTA

GR-ROSE は OTA にも対応しています。IoTCore のコンソール画面で OTA 更新ジョブを作成することで、GR-ROSE のファームウェアをアップデートすることができます。

OTA を実行するためには、GR-ROSE の bootloader を更新する必要があります。まず、

<u>https://www.renesas.com/jp/ja/software-tool/renesas-flash-programmer-programming-gui</u> より、RenesasFlashProgrammer をダウンロード、インストールしてください。※本デモ作成時は、 V3.06 を使用しました。

インストール後、RenesasFlashProgrammerを起動してください。ファイル->プロジェクトを開く、

 $amazon-freer tos {\tt \ \ } vendors {\tt \ \ } renes as {\tt \ \ } bootloader {\tt \ \ } gr-rose {\tt \ } gr-rose.rpj$

を選択してください。その後、プログラムファイルの参照から、

 $amazon - freer tos \\ \texttt{V} endors \\ \texttt{F} renes as \\ \texttt{F} boot loader \\ \texttt{F} rose_us \\ bfw.mot$

を選択してください。

下図赤枠のようにディップスイッチを設定し、PCとGR-ROSEをUSBで接続してください。



接続設定タブから、COM port ->ツール詳細で、GR-ROSE のポートを選択してください ※RX USB Boot(CDC)と表示されます

操作タブでスタートを押すと、Bootloader が更新されます。アプリケーションプログラムがすべて消去されてしまうので、再度 aws_demos.bin を書き込み、2.7 にしたがって、WiFi の設定、シリアル番号をしてください。

※本節で説明する OTA 機能は、本デモ用に作成した USB ストレージとしてのプログラムアップデート 機能と併用可能なブートローダを使用しているため、ここで紹介されているブートローダに求められる 要件を満たしていません。

- 2.11.1 Amazon S3 バケットの作成
 - a. AWS マネジメントコンソールから Amazon S3 アクセスしてください。
 - b. 「バケット ->「バケットの作成」を選択します。
 - c. バケット名(gr-rose-aws-demo-XXX)を入力、バケットのバージョニングを「有効」に 設定し、「バケットを作成」を選択します。(XXX は任意の文字)
 ※以降の説明では、バケット名=gr-rose-aws-demo としています

=	バケットのバージョニング バージョニングは、オブジェクトの連動のバリアントを同じパケット内に (3時すす手段です。バージョニングを使用すると、Amason SJ バ ウットに装飾スTいらすべてのオブシェクトのすべてのパーションを整件、物単、販売できます。パージョニングを使用すると、意図しな いユーザーアクションと意図しないアブリケーション得受の違力から影響に(回日できます、JMMI2C56) []	
<	バケットのバージョニング 	
	タグ (0) - オブション パケットにタグ付けすることで、ストレージユストやその他の基準を追逐します。 詳細 🖸	

2.11.2 OTA Update サービスロールの作成

- a. AWS マネジメントコンソールで、AWS IAM にアクセスします。
- b. 「ロール」->「ロールの作成」を選択します。
- c. 「信頼されたエンティティの種類を選択」で「AWS サービス」を選択します。
- d. 「サービスを選択してユースケースを表示」で「IoT」を選択し、「ユースケースの選択」 で「IoT」を選択し、「次のステップ」を選択します。
- e. そのまま「次のステップ」を選択します。
- f. そのまま「次のステップ」を選択します。
- g. ロール名 (ここでは test_ota_role)を入力し、「ロールの作成」を選択します。

2.11.3 OTA Update サービスロールに OTA Update アクセス許可を付与

- a. 「ロール」 > 検索ボックスに作成したロール名(test_ota_role)を入力 > 作成し たロールを選択 > 「ポリシーをアタッチします」を選択します。
- b. "amazonfreertos"でフィルタをかけ、「AmazonFreeRTSOTAUpdate」を選択し、「ポリシーのアタッチ」を選択します。

2.11.4 OTA Update サービスロールに AWS IAM アクセス許可を付与

a.「ロール」> 検索ボックスに作成したロール名(test_ota_role)を入力 > 作成し たロールを選択 > 「インラインポリシーの追加」を選択します。



※<your_account_id>は AWS アカウント ID(12 桁)で、コンソール右上のアカウントタブ

(下図赤部分)を参照してください。



c. ポリシー名(ここでは test_ota_iam_policy)を入力し、「ポリシーの作成」を選択します。

2.11.5 OTA Update サービスロールに Amazon S3 アクセス許可を付与



2.11.6 OTA ユーザーポリシーの作成

- a. 「ユーザー」> ユーザー選択 > アクセス権限 > 「アクセス権限の追加」を 選択します。
- b. 「既存のポリシーを直接アタッチ」>「ポリシーの作成」を選択します。

c. 「JSON」タブに切り替え、以下を入力し、「ポリシーの確認」を選択します。



※<your-account-id>は AWS アカウント ID(12 桁)を入力します。

- d. ポリシー名を test-ota-policy と入力し、「ポリシーの作成」を選択します。
- e. アクセス権限の追加->作成したポリシー(test-ota-policy)を検索・選択し、「次のス テップ」>「アクセス権限の追加」を選択します。

2.11.7コード署名証明書の作成

- a. OpenSSL をインストールしてください。https://www.openssl.org/
- b. AWS CLI をインストールしてください。
 <u>https://awscli.amazonaws.com/AWSCLIV2.msi</u> (Windows)
- c. AWS CLI の設定をします。
- AWS IAM >「ユーザー」> ユーザー選択 >「認証情報」タグ >「アクセ スキーの作成」を選択します。

- ii. CSV ファイルをダウンロードしてください。 ※CSV ファイルは厳重に保管する。
- iii. コマンドプロンプトを開き、"aws configure"を入力します。

C:¥ >aws configure	
AWS Access Key ID []: 1)
AWS Secret Access Key [l]: ②
Default region name [ap-northeast-1]:	3
Default output format [text]: ④	0

- ① CSV ファイルにある"Access key ID"を入力します。
- CSV ファイルにある"Secret access key"を入力します。
- ③ リージョンを入力します(東京の場合:ap-northeast-1)。
- ④ "JSON"を入力します。
- d. 作業用ディレクトリで、以下の内容の"cert_config.txt"ファイルを作成します。

	[req] prompt = no distinguished_name = my_dn
	[my_dn] commonName = test_signer@amazon.com
	[my_exts] keyUsage = digitalSignature extendedKeyUsage = codeSigning
۰,	,

※test signer@amazon.com は自身のメールアドレスに変更します。

e. 以下のコマンドを用いてコード署名プライベートキー、コード署名証明書を作成しま す。

[コード署名プライベートキー]

openssl genpkey -algorithm EC -pkeyopt ec_paramgen_curve:P-256 -pkeyopt

ec_param_enc:named_curve -outform PEM -out ecdsasigner.key

[コード署名証明書]

openssl req -new -x509 -config cert_config.txt -extensions my_exts -nodes -days 365 -key ecdsasigner.key -out ecdsasigner.crt

.....

.....

f. コード署名証明書、プライベートキー、および証明書チェーンを AWS Certificate Manager にインポートします。

aws acm import-certificate --certificate fileb://ecdsasigner.crt --private-key fileb://ecdsasigner.key

インポートに成功すると、CertificateArn が表示されるのでメモします

g. 作成した証明書(ecdsasigner.crt)をデバイスのソースコードに記述します。 amazon-

freertos¥vendors¥renesas¥rose_sdk¥iotgr_sdk¥include¥iotgr_param_config.h



- 2.11.8 Code Signing for AWS IoT に IAM ユーザーアカウントのアクセス許可を付与
 - a. AWS マネジメントコンソールで、AWS IAM にアクセスします。
 - b.「ポリシー」>「ポリシーの作成」>「JSON」タブで以下を入力 >「ポリシーの 確認」

{	
"Version": "2012-10-17",	
"Statement": [
{	
"Effect": "Allow",	
"Action": [
"signer:*"	
],	
"Resource": "*"	
}	
]	
}	
· · · · · · · · · · · · · · · · · · ·	

- c. ポリシー名(code_signing_policy)を入力し、「ポリシーの作成」を選択します。
- d. 「ユーザー」> ユーザー選択 >「アクセス権限の追加」を押します。
- e. 「既存のポリシーを直接アタッチ」> 作成したポリシー(code_signing_policy)を選
 択 >「次のステップ」を選択します。
- f. 「アクセス権限の追加」を選択します。

2.11.9 OTA 更新ジョブの作成

- a. AWS マネジメントコンソールで、AWS IoT コンソールにアクセスします。
- b. 「管理」>「ジョブ」>「ジョブを作成」>「FreeRTOS OTA 更新ジョブを作成」

を選択します。

- c. ジョブ名を入力し、次へを押します。
- d. 更新するデバイス(GR-ROSE_XX)を選択して「次へ」を選択します。
- e. ファームウェアイメージ転送プロトコルを MQTT とします。
- f. コード署名プロファイルを作成します。

OTA ファイル設定	この OTA 更新ジョブは、MQTT または HTTP 経由で、FreeRTOS ベースのモノや漫訳したモノのグループにファイルを安全に送信します。
Step 3 OTA ジョブ設定	更新するデバイス モノおよび/またはモノのグループを選択する ▼ GR-ROSE_1 X
	ファイル転送のプロトコルを通知します。 デバイスが2011-トするプロトコルを通知します。 ■ MITP ■ MITP
	ファイル ***
	ファイルに基金して進行する。 コード考れにより、使用いされる通知にようて使用いされて、コード考点のはサンドイスで使用され、得る後にコードのありがサインジがあります。 ● 新しいファイルに 重合します。 ● 新しいファイルに 重合します。 ● 「新しいファイルに す。」 ● 「おしいファイルに す。」 ● 「おしいファイルと選択した アイルを選択した す。」 ● カスタム集合ファ イルを使用しま す。
	コード署名プロファイルL このプロファイルには、コード時谷ショブの内式に必要に実施が点まれています。このプロファイルは、 デバイスのトレージェアプラド・ショーム・Anno Settifule Manager からの気目形、あよびデバイスと のコード発行工時からいこの場所を設定します。
	既存のユード署名プロファイル 既存のユード署名プロファイルを選択 ▼ 新しいプロファイルの作成
	77-1L
	 新しいファイルをアップロードします。

- g. プロファイル名(profile_ota_grrose)を入力します。
- h. デバイスハードウェアプラットフォーム(ESP32-DevKitC)を選択します。
- i. 「既存の証明書の選択」を選択し、先程インポートした証明書を選択します。
 ※先程メモした CertificateArn を選択してください
- j. 「デバイスのコード署名証明書のパス名」に certs/ecdsasigner.crt を入力します。

プロファイル名			
profile_ota_prose			
スペースを使用しない一葉の名前を入力します。 約 です。	物な文字は、3~2、4~2、0~9、_(アンダ	-23	
デバイスのハードウェアブラットフォーム			
ESP32-DevKitC		*	
AWS Certificate Manager (ACH) は、SSL/TLS 証明 応します。ACH を使用して、ACH 証明書を作成し3 ボートしたりできます。コードに裏名するには、こ	800%活、管理、インボートに開手る機能な とり、第名に使用するサードパーティの証明 の証明得が必要となります。	信職に対 勝をイン	
AMS Compliant Manager (ACM は、SU/US EEM CLIST、ACM を使用して、ACM EEMBERING ポートしたりてきます、コードに基合するには、こ の新しいコード集合証明書のインポ ート	800%活、発達、インボートに開する使用な たり、蓄毛に使用するサードパーティの経営 の証明者が必要となります。	組織に対	
AND Cartificate Manager (AND IS - SNUT) E EFE CULT - AND EXERTIC - AND HERBERING ボートしたりできます。コードに厳らするによ こ 新しいコード集合証明書のインボ ート ancowes.acm.ap-northeast-	800m紀、智徳、インボートに用する現象が 20、第6に世界するサードバーティの証明 の証明的な差となります。 の証明的な差となります。 の正明的な差になります。 の正明的な差になります。 の正明的な差になります。 の正明的な差になります。 の正明的などのであります。 の正明的などのでのであります。 の正明的などのであります。 の正明的などのであります。 の正明的などのであります。 の正明的などのでのであります。 の正明ののでのであります。 の正明ののでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでのでので	加速に対極をイン	
Nako Gartikaan Minagen (APO) E: SAL/1012度97 (APT)、APA とないこと、APA 12時後年5月10日、 APT-10-102 (2017)、SP-1-12番(2017)、APA 12 (APT)、APA 12 (2017)、APA 12 (2017)、APA 12 (APA 12 (2017)、APA 12 (2017)、APA 12 (2017) APA 12	たのめ、現象、インホートに至する現在と つ。最に定時するトレドレティーの逆形 の証明者から着となります。 の 数件の経営者の編択 ertificate/s▼ イメーン面も時間を取けてきために使用す	組織に対 線をイン こ こ 記利	

- k. 「作成」を押します。
- I. 更新用ファームウェアを準備します

amazon-freertos¥demos¥include¥ aws_application_version.h の以下赤枠 部分を"3"に変更し、ビルドを実行してください。

● * FreeRTOS V202012.00.	
<pre> #ifndef _AWS_APPLICATION_VE #define _AWS_APPLICATION_VE </pre>	ERSION_H_ ERSION_H_
<pre>#include "iot appversion32.</pre>	.h"
extern const AppVersion32_t	t xAppFirmwareVersion;
#define APP_VERSION_MAJOR	0
#define APP VERSION MINOR	9
<pre>#define APP_VERSION_BUILD</pre>	2
#endif	

m. 2.11.1.で作成したバケットに更新用ファームウェアをアップロードします。

「新しいファイルをアップロードします」を選択した状態で、k で作成した aws_demos.binを選択してください。

ファイルに運名して選択する コード発名により、回聴された危機者によって発行されたコードのみがデバイスで実行され、兼名後にコ ード授至使また経験用していないことが保証されます。コード著名には3つのオプションがあります。			
 新しいファイルに 署名します。 	 以前に署名したフ アイルを選択しま す。 	 カスタム署名ファ イルを使用しま す。 	
のプロファイルには、コード署:	各ジョブの作成に必要な情報が含ま	れています。このプロファイルは、	
ジバイスのハードウェアブラット シコード著名証明書のバスの場所 5年のコード要名プロファイ	フォーム、AWS Certificate Manage を指定します。 ル	rからの駐明書、およびデバイス上	
デバイスのハードウェアブラッド ウコード署名証明書のバスの場所 5月のコード署名プロファイ 既存のコード署名プロファ・	フォーム、AWS Certificate Manage を推定します。 ル イルを選択 ・	rからの証明書、あよびデバイス上 新しいプロファイルの作	
デバイスのハードウェアブラット DDコード兼名証明書のバスの場所 既存のコード署名プロファイ 既存のコード署名プロファ・ ファイル	フォーム、AWS Certificate Manage を推定します。 ル イルを選択 マ	からの証明書、およびデバイス上 新しいプロファイルの作用	
バイスのハードウェアブラッド コード者名証明書のバスの場所 群のコード署名プロファイ 既存のコード署名プロファイ ファイル の 新しいファイルをアッ: ます。	フォーム. AVIS Certificate Manage を用意します。 パル イルを選択 マ プロードし 〇 既存の	からの証明書、およびデバイス上 新しいプロファイルの作用 ンファイルを選択します。	
FI(イスの)(-ドウェアブラット) コード電話(新聞)(スの場所 数件のコード署名プロファイ 町 年のコード署名プロファイ ファイル の新しいファイルをアッ) ます。 マップロードするファイル	フォーム. WS Certificate Minage 使用でします。 ル プロードし ○ 気体の	からの対照表、およびデバイス上 新しいプロファイルの作 ジフィイルを選択します。	

- n. 「デバイス上のファイルパス名」に、certs/ecdsasigner.crt 入力します。
- o. IAM ロールで test_ota_role を選択し「次へ」を選択します。
- p. ジョブの作成を押します

2.11.10 OTA Update の実行

OTA ジョブ作成後、OTA のステータスは、ジョブ画面で参照することができます。

AWS IoT \times	 ⑥ 新しい AWS INT コンソールエクスペリエンスのご紹介 コンソールエクスペリエンスが更新されています。計測はこちらび新しいエクスペリエンスをお 	減しいたださ、 <u>ご意見をお聞かせください。</u> 新しい	エクスペリエンスはナビゲーションメニューから無
モニタリング アクティビティ	AWS lot 〉 管理 〉 ジョブ		
▶ Connect	ジョブ (200+) 🗰		C iiii
▼ 管理	ジョブは、AWS IoT に投続されているデバイスに送信して実行するリモートオペレーションを定義します。		
和要	Q ジョブをフィルタリング	すべてのステータス値	 すべてのタイプ
モノ			
タイプ	□ 名前	タイプ ステータス	作成日
モノのグループ	AFR 0TA etc. job. 20230912002100	7+072-0+02013	17 2021 00-71-06 (UTC) 00-00
請求グループ		X79793 8447000	August 15, 2021, 00:51:06 (01040500)
ジョブ	AFR_0TA-ota_job_20210811133210	スナップショ ② 充了	August 11, 2021, 13:32:14 (UTC+0900)
ジョブテンプレート	AFR_OTA-ota_job_20210811133208	スナップショ 〇 完了	August 11, 2021, 13:32:13 (UTC+0900)
トンネル			
保持されたメッセージ	AFR_OTA-ota_job_20210811133207	スナップショ ② 完了	August 11, 2021, 13:32:12 (UTC+0900)
フリートメトリクス 💷	AFR_0TA-ota_job_20210811133203	スナップショ 〇 完了	August 11, 2021, 13:32:09 (UTC+0900)

2.11.9.で入力したジョブ名をクリックすると、ジョブのステータスを確認することができます

<u>9aプレ教行</u> 9aプレキュメント 9aプターヴット タヴ			
ジョブの実行(1) === このジョブを見る影響しているダバイス。			C +42/62 III
Q、ジョブ実行を検索		すべてのジョブの実行(1 ▼	< 1 > @
 モノの名前 	· 一般转更新日	v 360+a-638M	v 27−92 ▲
GR-ROSE_1	August 11, 2021, 13:43:29 (UTC+09	00) August 11, 2021, 13:32:14 (UTC+0900)	⊘成功しました