AN12404

EdgeLockTM SE05x for secure connection to AWS IoT CoreRev. 1.6 — 15 January 2021Application note535216535216

Document information

Information	Content
Keywords	EdgeLock SE05x, AWS IoT Core, Secure cloud onboarding
Abstract	This application note describes how to leverage the EdgeLock SE05x for secure cloud onboarding to the AWS IoT Core IoT Hub cloud platform. It provides detailed instructions to run the software example provided as part of the support package using an OM-SE050ARD and an FRDM-K64F board



Revision history

Revision hi	story	
Revision number	Date	Description
1.0	2019-07-17	First release
1.1	2019-11-26	Update referring to MW v02.12 KSDK
1.2	2020-02-27	Updated to MW v02.12.03 and added appendix for Ease of Use configuration
1.3	2020-04-30	Fixed AWS thing subscription topic
1.4	2020-11-11	Updated document to include AWS IoT Core Multi-Account Registration feature
1.5	2020-12-07	Updated to latest template and fixed broken URLs
1.6	2021-01-15	Updated Section 4.1.3

1 EdgeLock SE05x ease of use configuration

The IoT device identity should be unique, verifiable and trustworthy so that device registration attempts and any data uploaded to AWS IoT Core can be trusted by the OEM. AWS IoT Core verifies the device identity using PKI cryptography. This authentication scheme requires that the associated private key remains secret and hidden from users, software or malicious attackers during the product's lifecycle.

The EdgeLock SE05x security IC is designed to provide a tamper-resistant platform to safely store keys and credentials needed for device authentication and device onboarding to cloud service platforms such as AWS IoT Core. Using the EdgeLock SE05x security IC, OEMs can safely connect their devices to AWS IoT Core without writing security code or exposing credentials or keys.

However, key generation and injection into security ICs can introduce vulnerabilities if not done properly. Manual provisioning can lead to errors and is difficult to scale when more devices are needed. Also, to ensure keys are kept safe, injection should take place in a trusted environment, in a facility with security features like tightly controlled access, careful personnel screening, and secure IT systems that protect against cyberattacks and theft of credentials, among others.

In order to allow OEMs to get rid of the complexity of key management and to offload the cost of ownership of a PKI infrastructure, the EdgeLock SE05x is offered pre-provisioned for ease of use. This means that OEMs are not required to program additional credentials and can leverage the EdgeLock SE05x ease of use configuration for most of the use cases, including for secure cloud onboarding of their devices to AWS IoT Core.

Note: NXP is offering the EdgeLock 2GO service, which supports different options for provisioning your devices and onboarding your devices to AWS IoT Core. You can find more details about EdgeLock 2GO at www.nxp.com/edgelock2go.

2 Leveraging EdgeLock SE05x for AWS IoT Core device onboarding

The security architecture of the AWS IoT Core uses X.509 certificates and TLS authentication for device onboarding. AWS IoT Core implements a feature called *Multi-Account Registration*, which simplifies the device registration process and makes it possible to onboard devices without requiring the registration of a CA certificate in AWS IoT Core.

The EdgeLock SE05x is pre-qualified to work with AWS IoT Core *Multi-Account Registration* feature, meaning that the pre-provisioned credentials in EdgeLock SE05x are qualified to connect to AWS IoT Core by default,. This way, devices can connect to AWS IoT Core by just registering the device certificate stored in EdgeLock SE05x.

<u>Figure 1</u> illustrates the device registration flow using EdgeLock SE05x ease of use configuration:

- 1. NXP delivers a quantity of EdgeLock SE05x ICs based on a purchase order to the OEM's manufacturing facility.
- 2. The OEM's device manufacturer assembles the EdgeLock SE05x ICs and deploys the software into the final IoT devices. It also needs to take care to read out the device certificate from the EdgeLock SE05x samples.
- 3. The OEM, as the system operator, manages the AWS IoT Core account and registers on it every device by registering its device certificate.
- 4. IoT devices boot up and automatically connect to AWS IoT Core service using the pre-provisioned credentials inside EdgeLock SE05x.



3 Running AWS IoT Core device onboarding project example

The AWS IoT Core project example showcases how to leverage EdgeLock SE05x security IC to set up trusted connections to AWS IoT Core cloud. This section explains how to run the AWS project example included as part of the EdgeLock SE05x support package.

Note: The AWS device onboarding procedure described in this section and the AWS demo example are provided only for evaluation purposes. Therefore, the subsequent procedure must be adapted and adjusted accordingly for a commercial deployment.

3.1 Hardware required

This guide provides detailed instructions to the AWS IoT Core project example using the hardware described below. However, you could use other MCU / MPU boards supported by EdgeLock SE05x Plug & Trust Middleware for this purpose as well.

1. OM-SE050ARD development kit:

Table 1. OM-SE050ARD development kit details

Part number	12NC	Content	Picture
OM-SE050ARD	935383282598	EdgeLock SE050 development board	

2. FRDM-K64F board:

Table 2. FRDM-K64F details

Part number	12NC	Content	Picture
FRDM-64F	935326293598	Freedom development platform for Kinetis K64, K63 and K24 MCUs	

3.2 Sign up for an AWS IoT Core account

Amazon offers 12 months of free tier access. To create an AWS IoT Core account:

1. Go to <u>https://aws.amazon.com/iot-core/</u> and click *Get started for free* button as shown in <u>Figure 2</u>:



2. If you already have an account with AWS, you will be prompted to log in. If you do not have an account yet, click on *Create a new AWS account* as shown in Figure 3:

	aws	
	Sign in e	
	Email address of your AWS account Or to sign in as an IAM user, enter your account ID or account alias instead.	
	Next	
	New to AWS? Create a new AWS account	
Figure 3. Sign in or c	create a new AWS IoT Core account	

Email address	
Password	
Confirm password	
AWS account name ()	
Continue	
Sign in to an existing AWS account	
© 2019 Amazon Web Services, Inc. or its affiliates. All rights reserved.	

3. Next, fill in the form with your email, password, AWS account name and click *Continue* as shown in <u>Figure 4</u>:

Figure 4. Create an AWS IoT Core account

4. Select the account type, complete the fields with your contact details and click *Create Account and Continue* as shown in <u>Figure 5</u>:

Contac		All fields are required.
Please select	the account type and complete the fields below with you	r contact details.
	Account type 1 Professional Personal Full name	
	Company name	
	Phone number	
	Country/Region	
	Address	
	City	
	State / Province or region	
	Postal code	
	Check here to indicate that you have read and agree to the terms of the AWS Customer Agreement	
	Create Account and Continue	

5. Supply a valid credit or debit card and click *Secure Submit* as shown in <u>Figure 6</u>. AWS will use it to verify your identity (i.e.there might be a record for a \$1 transaction on your bank statement that will be automatically returned). AWS will not charge you

unless your usage exceeds the AWS Free Tier Limits. You can check the limits in <u>Free</u> <u>Tier Limits</u>.

Please type yo you unless you questions for n	our payment information so we can verify your identity. We will not charge our usage exceeds the AWS Free Tier Limits. Review frequently asked nore information.
	Credit/Debit card number
	Expiration date
	Cardholder's name
	Billing address Use my contact address
	Secure Submit

6. Verify your phone number to confirm your identity as shown in <u>Figure 7</u>. When you continue, the AWS automated system will contact you with a verification code.

Before you can continue, the AV	use your AWS account, you must verify your phone number. When you NS automated system will contact you with a verification code.
	How should we send you the verification code?
	Text message (SMS) Ovice call
	Country or region code
	·
	Phone number
	Security check
	<u>`n72ep5</u> [•] [∗] ≎
	n72ep5
	Contact me

7. Enter the verification code that was sent to your cell phone and click *Verify code* as shown in <u>Figure 8</u>:



8. Your identity is validated. Choose *Basic* to obtain a free subscription as show in Figure 9



9. In a few minutes, you will receive via email the account creation confirmation. On the welcome page, choose *Get started* as shown in <u>Figure 10</u>:



10.To get started with AWS IoT Core, click *Services* and then *IoT Core* from the AWS Management Console as shown in Figure 11:

🖷 🖷 🥚 AWS Management Con	* × + ~			壒	
← → ♡ ŵ A 1 Services → History Console Home Billing	https://us-east-2.console.aws.amazon.com/console/ Resource Groups	/home?nd	:2=h_ct®ion=us-east-2&src=header-	-signin#	
	EC2 Lightsail [2] ECR ECS EKS Lambda Batch Elastic Beanstalk Serverless Application Repository		AWS RoboMaker Blockchain Amazon Managed Blockchain Satellite Ground Station		Athena EMR CloudSearch Elasticsearch Serv Kinesis QuickSight (2* Data Pipeline AWS Glue MSK
Figure 11 AWS M	Storage S3 EFS FSx S3 Glacier Storage Cataway Annagement Console - 4	ا ا	Management & Governance AWS Organizations CloudWatch AWS Auto Scaling CloudFormation CloudTrail		Security, Identity IAM Resource Access Cognito Secrets Manager GuardDuty

3.3 Create an AWS IoT thing

An AWS IoT thing is a representation of your physical device in the cloud. The AWS IoT thing is an entry in the registry that contains attributes that describe a device. Any

AN12404	
Application not	е

physical device needs a thing record in order to work with AWS IoT. To create an AWS IoT thing, follow these steps:

1. From the AWS IoT Core dashboard, go to **Manage**, go to **Things** and click on the **Register a thing** button as shown in <u>Figure 12</u>:

aws Services ~	Resource Groups 🗸 🔭
AWS IOT	
Monitor	Tata -
Onboard	
Manage	
Things 1	
Thing Groups	
Billing Groups	You don't have any things yet
Jobs	A uning is the representation of a device in the could.
Secure	Learn more Register a thing
Defend	
Act	
Test	
ure 12 Go to th	AWS INT Things manu

2. A new menu called *Creating AWS IoT Things* will be opened. Click on the **Create a** single thing option as shown in Figure 13

aws	5 Services 🗸 Resource Groups 🗸 🍾	
	Creating AWS IoT things	
	An IoT thing is a representation and record of your phylicical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. Learn more.	
	Register a single AWS IoT thing Create a thing in your registry	Create a single thing
	Bulk register many AWS IoT things Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.	Create many things
	Cancel	Create a single thing
Figure '	13. Select <i>Create a single thing</i> option	

3. A new form called **Add your device to the thing registry** will be opened. For the purpose of this demo, you only need to fill in the AWS IoT Thing **name** and click **Next** as shown in Figure 14

aws	Services	∽ Resource Groups ∽ 🏌	
		Add your device to the thing registry	P
		This step creates an entry in the thing registry and a thing shadow for your device. Name my_thing	
		Apply a type to this thing Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with common set of attributes, which describe the identity and capabilities of your device, and a description. Thing Type No type selected	a
		Add this thing to a group Adding your thing to a group allows you to manage devices remotely using Jobs. Thing Group	
		Groups / Create group Change Set searchable thing attributes (optional) Enter a value for one or more of these attributes so that you can search for your things in the registry.	•
		Attribute key Value Provide an attribute key, e.g. Manufacturer Provide an attribute value, e.g. Acme-Corporation Add another Show thing shadow •	·
		Cancel Back Next	
Figure 14.	Add y	our device to the thing registry	

 Click on Create a thing without certificate to complete the AWS IoT Thing creation as shown in <u>Figure 15</u>. The certificate for your IoT Thing will be added later on in this tutorial.

CREATE ATTRING Add a certificate for your thing	STEP 2/3
A certificate is used to authenticate your device's connection to AWS IoT.	
One-click certificate creation (recommended) This will generate a certificate, public key, and private key using AWS IoT's certificate authority.	Create certificate
Create with CSR Upload your own certificate signing request (CSR) based on a private key you own.	Create with CSR
Use my certificate Register your CA certificate and use your own certificates for one or many devices.	Get started
Skip certificate and create thing You will need to add a certificate to your thing later before your device can connect to AWS IoT.	Create thing without certificate

5. Now, your AWS IoT Thing should be created and visible in your AWS IoT Core dashboard as shown in Figure 16

aws Service	s 🗸 Resource Groups 🗸 🏷	
🖗 AWS ΙοΤ	Things	Create 🖓
Monitor	Search things Q Configure fleet Indexing ()	Card •
Onboard Manage Things	my_thing NO TYPE	
Types Thing Groups		
Billing Groups Jobs		
Secure Defend		
Act Test		
aure 16. Conf	irm AWS IoT Thing creation	

3.4 Create a policy

AWS IoT policies are used to authorize your device to perform AWS IoT operations, such as subscribing or publishing to MQTT topics. To allow your device to perform AWS IoT

AN12404

operations, you must create an AWS IoT policy and attach it to your device certificate. To create an AWS IoT policy, follow these steps:

1. From the AWS IoT Core dashboard, go to **Secure**, go to **Policies** and click on the **Create a policy** button as shown in <u>Figure 17</u>:

aws Services ~	Resource Groups 🗸 🐐
AWS IOT	
Monitor Onboard Manage Secure	
Certificates Policies 1 CAs Role Aliases Authorizers Defend	You don't have any policies yet AWS IoT policies give things permission to access AWS IoT resources (like other things, MQTT topics, or thing shadows). Learn more Create a policy 2
Act Test	
Figure 17. Go to the	AWS IoT policies menu

2. A new menu called *Creating a policy* will be opened. Fill in a **name** for your AWS IoT policy and click on **Advanced mode** option as shown in <u>Figure 18</u>

aws Services - Resource Groups - 1	
Create a policy	
Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page. Name my_policy 1	
Add statements Policy statements define the types of actions that can be performed by a resource.	
Action Please use commas to seperate actions. e.g. lot:Publish; lot:Subscribe Resource ARN	
Specific resources could include client ID ARN, topic ARN, or topic filter ARN. Effect	
Allow Deny Remove	
Add statement	
Cruste	
Figure 18. Create a policy name and go to Advanced mode.	

3. Use the text box to personalize your policy with the following text:

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

Click on **Create** button as shown in Figure 19.

aws	Services 🗸 Resource Groups 🖌 🏷	
	Create a policy	
	Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the AWS IoT Policies documentation page. Name my_policy	
	Add statements Basic mode	
	Create	
Figure 19.	Personalize your AWS IoT Core policy	

4. Now, your AWS IoT policy should be created and visible in your AWS IoT Core dashboard as shown in Figure 20

aws	Services	Resource	Groups 🗸	*		
		Policies			Create) Т
		Search policies			Card 🗸 🖉	9
Monitor				1		
Onboard		my_policy	***			
Manage						
Secure						
Certificates						
Policies						
CAs						
Role Aliases						
Authorizers						
Defend						
igure 20. Co	onfirm A	VS loT poli	icy creati	on		

3.5 Extracting credentials from EdgeLock SE05x

We will be using ECC credentials in this example, corresponding to key ID $0 \times F0000000$ and certificate ID $0 \times F0000001$. You can use any of the available certificates that are preprovisioned in your EdgeLock SE05x. Please refer to <u>AN12436 - SE050 Configurations</u> for a list of available key and certificate IDs.

3.5.1 Download EdgeLock SE05x Plug & Trust Middleware

Follow these steps to download the EdgeLock SE05x Plug & Trust Middleware in your local machine:

1. Download EdgeLock SE05x Plug & Trust Middleware from the NXP website.

• 🔶 • 🛧 🍆 > Th	iis PC » Local Disk (C:)			
Terres 🛛 🖈 ^	Name	Date modified	Туре	Size
🔈 Music	Intel	2/25/2019 4:12 AM	File folder	
Projects		3/7/2019 1:28 AM	File folder	
📓 Videos	PEMicro	3/7/2019 1:34 AM	File folder	
ConeDrive	PerfLogs	4/11/2018 4:38 PM	File folder	
	Program Files	3/11/2019 4:05 AM	File folder	
狊 This PC	Program Files (x86)	3/11/2019 3:28 AM	File folder	
🔓 3D Objects	Projects	3/11/2019 6:17 AM	File folder	
E Desktop	Python27	3/11/2019 4:53 AM	File folder	
E Documents	se050_middleware	3/11/2019 6:27 AM	File folder	
Downloads	Users	2/25/2019 5:06 AM	File folder	
Music	Windows	3/11/2019 3:38 AM	File folder	
Pictures	Recovery	2/25/2019 12:46 PM	Text Document	01
Videos				
Local Disk (C:)				
🥪 USB DISK (E:)				

2. Create a folder called **se050_middleware** in C: directory as shown in Figure 21:

 Unzip the EdgeLock SE05x Plug & Trust Middleware inside the se050_middleware folder. After unzipping, you will see a folder called simw-top created. The contents of the simw-top directory should look as shown in Figure 22:

🗧 🔶 👻 🛧 📙 > This PC >	OS (C:) > se050_middleware > simw-top	(Search simw-top	م
1 Quidenne	Name	Date modified	Туре	Size		
Y QUICK access	akm	7/18/2019 5:26 PM	File folder			
MobileKnowledge	binaries	7/18/2019 5:26 PM	File folder			
This DC	demos	7/18/2019 5:26 PM	File folder			
This PC	doc	7/18/2019 5:27 PM	File folder			
SB Drive (D:)	ext	7/18/2019 5:27 PM	File folder			
A Naturali	hostlib	7/18/2019 5:27 PM	File folder			
- Network	projects	7/18/2019 5:27 PM	File folder			
	pycli	7/18/2019 5:27 PM	File folder			
	scripts	7/18/2019 5:27 PM	File folder			
	sss	7/18/2019 5:27 PM	File folder			
	tools	7/18/2019 5:27 PM	File folder			
	CMakeLists.bt	7/18/2019 5:26 PM	Text Document	3 KB		
	🔒 EULA.pdf	7/18/2019 5:26 PM	Adobe Acrobat D	134 KB		
	PlugAndTrustMW.pdf	7/18/2019 5:26 PM	Adobe Acrobat D	11,252 KB		
	README.First.txt	7/18/2019 5:26 PM	Text Document	1 KB		
	Third Party License.pdf	7/18/2019 5:26 PM	Adobe Acrobat D	369 KB		
	version_info.txt	7/18/2019 5:26 PM	Text Document	1 KB		

Note: It is recommended to keep *se050_middleware* with the <u>shortest</u> path possible and <u>without spaces</u> in it. This avoids some issues that could appear when building the middleware if the path contains spaces.

3.5.2 Flash FRDM-K64F with VCOM software

The VCOM software allows the FRDM-K64F board to be used as a bridge between the Windows machine and the EdgeLock SE05x and enables the execution of the EdgeLock SE05x ssscli tool and other utilities from the laptop. To flash the VCOM software into the FRDM-K64F, follow these steps:

1. Unplug and plug again the USB cable to the openSDA USB port as shown in <u>Figure 23</u>:



2. When you plug the board, your laptop should recognize the board as an external drive as shown in Figure 24:

← → ~ ↑ 🖬 > FRDM-K6	54FD (E:)			
★ Quick access MobileKnowledge	Name DETAILS.TXT PRODINFO.HTM	Date modified 3/22/2016 4:30 PM 3/22/2016 4:30 PM	Type Text Document HTM File	Size 1 KB 1 KB
Image: This PC Image:				
🛶 FRDM-K64FD (E:)				

 Flash the VCOM software to FRDM-K64F. The VCOM software binary can be found in the EdgeLock SE05x Plug & Trust Middleware package, inside the simw-top \binaries folder as shown in Figure 25:



4. Drag and drop or copy and paste the a7x_vcom-T1oI2C-frdmk64f-SE050x.bin file into the FRDM-K64F drive from your computer file explorer as shown in Figure 26:

Date modified Type Size 3/22/2016 4:30 PM Text Document 1 KB 3/22/2016 4:30 PM HTM File 1 KB Text Document 1 KB 1/22/2016 4:30 PM HTM File 1 KB Text Document 1 KB 3/22/2016 4:30 PM HTM File 1 KB

- 5. The serial and VCOM ports should be recognized by your Device Manager. To check that the ports are recognized, follow the steps indicated in Figure 27:
 - a. Unplug the USB cable from the OpenSDA USB port.
 - b. Plug the USB cable to the OpenSDA USB port.
 - c. Check that the serial port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as USB Serial Device (COM7) but this naming might change depending on your computer. Therefore, it is important that you

identify which device is recognized at the moment you plug the SDA USB port to the computer.

- d. Plug the USB cable to the K64F USB port.
- e. Check that the VCOM port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as *Virtual Com Port (COM8)* but this naming might change depending on your computer (e.g. It could also appear named as *USB Serial Device*). Therefore, it is important that you identify which device is recognized at the moment you plug the K64F USB port to the computer.



Note: Please note that it is possible that either of the two COM ports is not detected when using low-quality or charge-only USB cables.

3.5.3 Read device certificate from EdgeLock SE05x

To read the device certificate from EdgeLock SE05x storage, follow these steps:

1. First, open a command prompt and navigate to C:\se050_middleware\simw-top \binaries\pySSSCLI.

Send >cd C:\se050_middleware\simw-top\binaries\pySSSCLI.

 Connect to the EdgeLock SE05x using the executable ssscli.exe. You need to indicate the VCOM port number corresponding to the K64 USB port of your board (See <u>Section 3.5.2</u>).

Send >ssscli connect se050 vcom COM9.

3. We shall first make sure that the chosen keys we will be using are indeed available in the EdgeLock SE05x. To do this, we will fetch a list of available keys. Send >ssscli se05x readidlist. As shown in Figure 28, we can confirm that the desired keys are available.

C:\Windows\System32\	cmd.exe			-	\times
C:\se050_middleware	\simw-top\binar	ies\pySSSCLI>sssc	li connect se050 vcom COM9		
C:\se050 middleware	simw-top\binar	ies\pvSSSCIT>sssc	li se05x readidlist		
Opening COM Port '\	.\COM9'	100 (0) 00000000000000000000000000000000	II SCOSA (Cadialise		
sss :INFO :atr (Le					
00 00 0A 00	03 96 04 03	E8 00 FE 02	0B 03 E8 08		
01 00 00 00	00 64 00 00				
<pre>Key-Id: 0X7da00013</pre>	NIST-P	(Public Key)	Size(Bits): 256		
Key-Id: 0X7da00003	NIST-P	(Public Key)	Size(Bits): 256		
<pre>(ey-Id: 0X7da00012</pre>	AES		Size(Bits): 128		
(ey-Id: 0X7da00002	AES		Size(Bits): 128		
<pre>(ey-Id: 0X7da00011</pre>	USER-ID				
<pre>Key-Id: 0X7da00001</pre>	USER-ID				
Key-Id: 0X7fff0205	USER-ID				
Key-Id: 0X7fff0209	COUNT		Size(Bits): 32		
(ey-Id: 0Xf0000030	AES		Size(Bits): 128		
<pre>Key-Id: 0Xf0000003</pre>	BINARY		Size(Bits): 3760		
(ey-Id: 0X+0000001	BINARY		Size(Bits): 3760		
(ey-Id: 0X+0000002	NIST-P	(Key Pair)	Size(Bits): 256		
(ey-Id: 0X+0000000	NIST-P	(Key Pair)	Size(Bits): 256		
(ey-1d: 0X+0000012	NIST-P	(Key Pair)	Size(Bits): 256		
	NIST-P	(Public Key)	Size(Bits): 256		
(ey-10: 0X/TTT0204	NIST-P	(Public Key)	Size(Bits): 250		
(ey-10: 0X/1110202	NIST-P	(Key Pair)	Size(Bits): 250		
Key-IU. 0X/TTT0201		(key Pair)	Size(Bits), 250		
key-10. 0x/1110206	DIWART		S120(B1(S), 144		

Figure 28. Connect to the EdgeLock SE05x using ssscli and read certificate ID list

- 4. We will now retrieve the device certificate from the EdgeLock SE05x. Send >ssscli get cert 0xF000001 device_cert.cer. As shown in Figure 29, the certificate has been written to a file in the current path.
- 5. Finally, disconnect the communications to the EdgeLock SE05x. If the channel is not closed properly, you won't be able to establish a new connection until this command is executed.

Send >ssscli disconnect.

C:\s	e050_middleware\simw-top\binar	ies\pySSSCLI>ss	scli get	cert	0xF0000001 devic	e_cert.cer		
lett:	ing Certificate from KeyID = 0	xF0000001						
)pen	ing COM Port '\\.\COM9'							
	00 A0 00 00 03 96 04 03	E8 00 FE 02	0B 03	E8 08				
	54 50 4F			20 41				
	:WARN :!!!Not recommended to	r production us						
Retr	ieved Certificate from KeyID =	0xF0000001						
C•\e	050 middleware\simw_ton\hinar	ies\nv\$\$\$CLT>						

Figure 29. Get the certificate from the EdgeLock SE05x using ssscli

3.6 Registering device certificate in AWS IoT Core

The next step is to register the device certificate in AWS IoT Core. For that, follow these steps:

 From the AWS IoT Core dashboard, go to Secure, go to Certificates and click on the Create a certificate button as shown in <u>Figure 30</u>:

aws Service	es 🗸 Resource Groups 🗸 🐪
💮 AWS ІОТ	
Monitor	
Onboard	
Manage	
Secure	
Certificates	
Policies	You don't have any certificates yet
CAs	Certificates help things establish a secure connection.
Role Aliases	
Authorizers	Learn more Create a certificate
Defend	
Act	
Test	
Figure 30. Go to t	he AWS IoT Certificates menu

2. A new menu called *Create a certificate* will be opened. Click on the **Get started** option as shown in Figure 31:

Create	a certificate	
A certificate	is used to authenticate your device's connection to AWS IoT.	
One-click This will gen authority.	certificate creation (recommended) erate a certificate, public key, and private key using AWS IoT's certificate	Create certificate
Create wit Upload your	th CSR own certificate signing request (CSR) based on a private key you own.	Create with CSR
Use my ce Register you	ertificate Ir CA certificate and use your own certificates for one or many devices.	Get started

A new menu called *Select a CA* will be opened. In this menu, choose *Next* as shown in <u>Figure 32</u>:

aws Services ▼						¢	jordi.jofre 🔻	Frankfurt 🔻	Support 🔻
AWS IoT	× Aws	IoT > Certificates > 0	Create a certificate >	Select a CA					Î
Monitor Activity		Select a CA							
 Onboard Manage Things 		Select or register the CA ce registered CA, just select N	ertificate used to sign yo ext. Learn more.	ur device certificates. To I	use device certificat	tes tha	it are not signed	d by a	1
Types Thing groups Billing groups Jobs		Registered CAs	te						
Tunnels Greengrass 		No match found There is no CA ce	rtificate in your account						
 Secure Certificates Policies CAs 		Register CA							
Role Aliases Authorizers Defend		Cancel					•	Next	
Figure 32.	Select CA	menu							

3. On *Register existing device certificates*, choose *Select certificates*, and select the certificate exported from EdgeLock SE05x in <u>Section 3.5</u>, as shown in <u>Figure 33</u>:

aws Services ▼	🗘 jordi.jofre 🔻 Frankfurt	 Support
WS IoT ×	VS IoT > Certificates > Create a certificate > Select a CA > Register existing device certificates	
Ionitor ctivity	Register existing device certificates	
anage lings	You can upload up to 10 device certificates at one time. If you selected a CA, make sure you upload only certificates signed by th CA. Learn more. Existing certificates	hat
rpes ning groups	You have not selected any device certificates to upload yet.	
lling groups	Select certificates	
innels	se600_middleware → simultes → p\$555CU v δ	
reengrass	IE2 ↓ Iftey ^ Name ^ Dite modified Type Size	_
ecure	By materieneningspel 21 dep=2323 MI Potent Ensemin Bytemstepel 21 dep=2323 MI Potent Ensemin Done Constant Done Done	
licies	July 0 July 1 July 1 July 2 July 2	
ls Aliasas	D Constraint 11 Marco 20 10819M Security Confidence Sin Norgato 1 Luit 21 Sope 2022 DIM Application estimation 31 Sin Norgato 2, Luit 21 Sope 2022 DIM Application estimation 32	
ithorizers	Browycie 21% - brz-br 22.01 MI Compressed Stops 6.6 Brods-11, ddl 21% - brz-br 22.01 MI Application enter 1 Ell Provisión, MVS.ex 21% - brz 22.01 MI Application 1	

4. After closing the file dialog box, select *Activate all* and then click on *Register certificates*, as shown in Figure 34:

AWS INT X		
	AWS IOT > Certificates > Create a certificate > Select a CA > Register existing device certificates	
Monitor		
Activity	Register existing device certificates	
Onboard		
Manage	You can upload up to 10 device certificates at one time. If you selected a CA, make sure you upload only certificates signed by that	
hings	CA. Learn more.	
ypes	Existing certificates	
hing groups	Activate all Unselect all	
Billing groups	device_cert.cer	
obs	Remove	
lunnels	▲ Select certificates	
Greengrass		
Secure		
Certificates	Cancel Done Done	
olicies		
AS		
uthorizers		
Defend		
verenu		

The device certificates that are registered successfully appear in the list of certificates.

3.7 Attach AWS Thing and policy to the certificate

Finally, we only need to attach the thing and policy we created back in <u>Section 3.3</u> and <u>Section 3.4</u>, respectively, to the newly registered device certificate. Go to the AWS IoT Core administration console and follow the steps:

- 1. Attach your thing to the certificate following the instructions shown in Figure 35.
 - a. Click on the top right corner to go to the device certificate options.
 - b. Click on Attach a thing.
 - c. Select the AWS IoT Thing you created in <u>Section 3.3</u>. In this example, it was called **my_thing**.
 - d. Click on the Attach button.

AWS IoT X	AWS IOT > Certificates	
Aonitor	Certificates	Create
Dnboard	Search certificates Q	
1anage	Name Status	
ypes hing groups	0704c90b4637c035d35def2efb1fb254eb58e944f181452cc5c2201655756d29e Active	a Activate
lling groups ibs	Attach things to certificate(s)	Deactivate Revoke Accept transfer
nnels reengrass	Things will be attached to the following certificate/ol: 704c90b4637c035d35def2efb1fb254eb58e944f181452cc5c2201655756d29e	Reject transfer Revoke transfer
ecure ertificates	Choose one or more things	Start transfer Attach policy Attach thing
olicies As	c wy_thing	Download Delete
ole Aliases uthorizers	1 thing selected T	
efend		

- 2. Attach your policy to the certificate as shown in Figure 36.
 - a. Click on the top right corner to go to the device certificate options.
 - b. Click on Attach a policy.
 - c. Select the AWS IoT Policy created in <u>Section 3.4</u>. In this example, it was called **my_policy**.
 - d. Click on the Attach button.

AWS IoT	×	AWS IOT > Certificates			
Monitor		Certificates			Create
Activity					
Onboard		Search certificates Q			
Manage		Name	Status		
Things				_	
Types		704c90b4637c035d35def2efb1fb254eb58e944f181452cc5c2201655756d29e	Active	a	••••
Thing groups			1	Activate	
Billing groups		Attach policies to certificate(s)		Deactivate	
lobs				Accept trans	sfer
Tunnels		Policies will be attached to the following certificate(s):		Reject trans	fer
Greengrass		704c90b4637c035d35def2efb1fb254eb58e944f181452cc5c2201655756d29e		Revoke tran	sfer
Cocuro.		Choose one or more policies		Start transfe	er
Cortificator				b Attach polic	У
Policios		Q Search policies		Attach thing	1
-oncies		C my_policy View		Delete	
As Aliacos					
Authorizers		1 policy selected			
Defend	-		_		
edback English (US) 🔻		© 2008 - 2020. Amazon Web Services. Inc. or its	affiliates. Al	I rights reserved. Priva	cy Policy Terms of

Figure 36. Attach a policy to your device certificate

3.8 AWS IoT Core project configuration

To run the AWS project example using the FRDM-K64F board, we need to:

- Download and install the FRDM-K64F SDK
- Import AWS IoT Core example project
- <u>Configure AWS IoT Core project account settings</u>
- Execute AWS IoT Core example project

Note: Before running the AWS IoT Core demo example, you need to have installed MCUXpresso IDE and FRDM-K64F SDK in your local environment and imported the AWS IoT Core project example. Check <u>AN12396- Quick start guide to Kinetis K64</u> for detailed instructions on:

- How to install MCUXpresso
- How to obtain FRDM-K64F SDK
- How to import FRDM-K64F project examples, including AWS IoT Core project example.

3.8.1 Download and install the FRDM-K64F SDK

The AWS IoT Core device onboarding project example is included as part of the FRDM-K64F SDK. Install it to your MCUXpresso workspace as shown in <u>Figure 37</u>:

1. Download the FRDM-K64F SDK, publicly available from the NXP website.

AN12404	All information provided in this document is subject to legal disclaimers.	© NXP B.V. 2021. All rights reserved.
Application note	Rev. 1.6 — 15 January 2021	

535216

- 2. Drag and drop the FRDM-K64F SDK zip file in the *Installed SDKs* section in the bottom part of the MCUXpresso IDE.
- 3. Check that the FRDM-K64F SDK is installed successfully.

workspace - Welcome page - MCUXpresso IDE	SDK Import Wizard	
File Edit Navigate Search Project ConfigTools Run Analysie		
Image: A state of the state	8 Please select a board	
Project 22 📴 Periphe III Registers 🎋 Faults 📟 🗖	Board and/or Device selection page	
	SDK MCUs Available boards	
	MCUs from installed SDKs Please select an available board for your project.	
	NXP MIMXRT1052xxxxB Supported boards for device: MIMXRT1052xxxxB	-
	> K6x	
	MARCHINGS MARCHINGSCHEE SECON SECON	🍘 Installed SDKs 🙁 🛅 Properties 🖹 Problems 🖸 1
U Qui 🛛 (X)= Vari 📴 Outl 🗣 Bre (V)= Glo 🐃 🗖		🕅 Installed SDKs
_		To install an SDK, simply drag and drap an SDK (sin file/fol
MCUXpresso IDE - Quickstart Panel		Name
· Grante er import a project	Salacted Devices SDVz for calacter	
· create or import a project	Tarnet Core: cm7 Name	# SDK 2.x ERDM-K64E
Markey project Markey project Markey project (s) from file system.	Description: MIMXRT1052: LMX & MIMXRT1052: 600MHz, 512KB SRAM Microcontrollers (MCUs) based on ARM & Cortex & -M7 Core	BDK_2.x_LPCXpresso55569
▼ Build your project		
Clean	0	
Figure 37. Import FRDI	M-K64F SDK	

Note: For more detailed instructions on how to install it the FRDM-K64F SDK into our MCUXpresso workspace, refer to <u>AN12396 - Quick start guide with FRDM-K64F</u>.

3.8.2 Import AWS IoT Core example project

The FRDM-K64F SDK includes a project example called se_SE050x_cloud_aws. Import it to your MCUXpresso workspace as shown in Figure 38:

- 1. Click *Import SDK examples* from the MCUXpresso IDE quick start panel.
- 2. Select se SE050x cloud aws project example and click the Finish button.
- 3. Check that the project is now visible in your MCUXpresso workspace

Note: For detailed instructions on how to import project examples from FRDM-K64F SDK, check <u>AN12396</u> - <u>Quick start guide with Kinetis K64F</u>

AN12404

AN12404

EdgeLock[™] SE05x for secure connection to AWS IoT Core

workspace - Welcome page - MCUXpresso IDE	SDK Import Wizard	🔀 workspace - Welcome page - MCUXpresso IDE
File Edit Navigate Search Project ConfigTools Run Analysi:		File Edit Navigate Search Project ConfigTools Run Analysi
➡ ₩ ₩ ₩ ▼ % ▼ 20 ♥ 0 ▼ 1	The source from the SDK will be copied into the workspace. If you want to use linked files, please unzip the 'SDK_2x_FRDM-K64F' SDK.	Image: Second secon
Project X 🛃 Periphe IIII Registers 🎋 Faults 🗖 🗖	Import projects	Project X Periphe III Registers 🎋 Faults 🗖 🗖
	Project name prefix: frdmk64f	Gef frdmk64f_se_SE05x_cloud_aws < Debug> Section Settings
	Use default location	> 🗿 Includes > 🙉 CMSIS
	Location: C:\Users\Jordi Jofre\Documents\MCUXpressolDE_11.0.0_2516\workspac	> 🙆 amazon-freertos
	Project Type	> 🔁 board
	C Project C++ Project C Static Library C++ Static Library	> 🙆 device
		> 😫 drivers
		> 😂 libs
		> 😂 lwip
	Examples	> 😕 mbedtls
	type to filter	> 🐸 mmcau
	type to litter	> 😕 se_hostlib
	Name	> 🔁 source
	> 🗌 💆 demo_apps	> A startup
	✓ ■ Se_hostlib_examples	> 🗁 dor
	Se SEDSx cloud azure	7 B 400
	Se SE05x cloud gcp	
	se_SE05x_cloud_ibm	
	□ ≡ se_SE05x_ex_ecc	(1) Out M (V-Vari P= Out) 9- Bra M-Gla P
O Qui 23 (W- Vari 02 Outi 40 Bre W- Glo	se_SE05x_ex_hkdf	
	se_SE05x_ex_md	
MCIIVersees IDE Quisksteet Panal	se_SEO5x_ex_rsa	MCIIV proceso IDE - Quicketart Papel
No project relected	□ = se_seuox_ex_symmetric	Project: frdmk64f se SE05x cloud aws [Debug]
Greate or import a project	se_hostlib_se05x_ex_i2cMaster	Create or import a project
	Se_hostlib_seUox_get_into	
New project	□ = se_hostib_seox_minimal	New project
Market SDK example(s)	□ ≡ se_hostlib_vcomSE050	Import SDK example(s) Import project(s) from file system
→ Build your project		▼ Build your project
💽 🐔 Build		Real 🚳 Build
Clean		Clean
v	(?)	×
		frdmk64f_se_SE05x_cloud_aws
	L	L
Figure 38. Import AWS p	roject in the workspace	
gale eepoit/ite p		

3.8.3 Configure AWS IoT Core project account settings

We need to change the AWS Rest API Endpoint in the MCUXpresso demo project with the one in your AWS IoT Core account settings. Follow these steps:

1. From the AWS IoT Core dashboard, go to *Manage*, then go to *Things* and click on your AWS IoT Thing as shown in Figure <u>39</u>:

aws Service	s 👻 Resource Groups 👻 🛠		
AWS IOT	Things		Create
Monitor	Search things	Q Configure fleet indexing ③	Card 👻
Onboard	my_thing NO TYPE		
Manage			
Things			
Types			
Thing Groups			
Billing Groups			
Jobs			
Secure			
Defend			
Act			
Test			
Figure 39. Go to	your AWS loT Thing		

2. On the left hand side menu, (1) go to **Interact**. Inside this menu, you will find your (2) Rest API Endpoint as indicated in <u>Figure 40</u>. Copy this URL.

aws	Services 🗸 Re	source Groups 🗸	*			Δ	jordi.jofre 🗸	Ohio 👻	Support 👻
< label{eq:constraint}	Things > my_thing								ņ
_	THING								?
	my_thing								(\mathcal{D})
	NO TYPE						Act	ons -	
	Details	This thing alread	y appears to be connec	ted.			Connect a c	evice	
	Security	HTTPS							
	Billing Groups	-							
	Shadow	Update your Thir	ng Shadow using this Re	est API Endpoint. Learn mo	re				
	Interact	anarynuiagy	TT-ats.10t.us-eas	st-z.amazonaws.com					2
	Activity	MQTT							
	Jobs	Use topics to ena	ble applications and th	nings to get, update, or dele	te the state information f	or a Th	ning (Thing Shad	ow)	
	Defender metrics	Learn more							
	new	Update to this th	ning shadow	A SHORE					
		\$aws/things	:/my_thing/shadow,	/update				_	
		Update to this t	ning shadow was accep	oted				_	
		\$aws/things	/my_thing/shadow,	/update/accepted					
Figure 4	40. Find y	our Res		point					

3. Go to the AWS demo in your MCUXpresso workspace. Navigate to the aws_clientcredential.h file located in frdmk64f_se_SE05x_cloud_aws \source folder. Replace the clientcredentialMQTT BROKER ENDPOINT

All information provided in this document is subject to legal disclaimers.	© N

variable with the Rest API Endpoint of your AWS account obtained in the previous step, as well as your thing name as created in <u>Section 3.3</u>. Check <u>Figure 41</u> for reference.

🔇 workspace - frdmk64f_se_SE05x_cloud_aws/source/aws_clientcre	lential.h - MCUXpresso IDE -	
Edit Source Refactor Navigate Search Project Confi	JTools Run Analysis FreeRTOS Window Help	
	■ N 3 5 2 平 7 18 11 ■ 3 5 2 6 · 2 2 2 1 5 5 5 5 • 0 • 0 • 0 • 4 2 1 3 1 5 • 5 • 5 • 5 • 5 • 5 • 5 • 5 • 5 • 5	⇔ -
		-
	Quick Access	: B. 1 💘 1
Project 🕺 🐕 Periphe 👭 Registers 🏘 Faults 🔗 🗖	Welcome aws_clientcredential.h 🕴	- 6
- 🔶 m A. 🕅 🗸 🔻	49 #endif	^
frdmk6/f ra SE05v cloud aver (Dahun)	50⊝ /*	
Project Settings	51 MQTT Broker endpoint.	
) includes 2	static const char clientscedentialMOTT REGKER ENDPOINT[] : "approved] Professor int usearet-2 amazonawar com";	
> CMSIS	54	
> 🐸 amazon-freertos	550/" Use of a "define" and not a "static const" here to be able to	
> 😂 board 📃 🧧	56 "use pre-comple concatenation on the string,"/	
> 😂 component	Se der the Cirener edentiario_Infind_novic ing_ciring	
> 😕 device	59⊕ /*	
> 😂 drivers	60 * Port number the MQTT broker is using.	
> 😂 libs	61 */	
> 😂 lwip	63	
> 🙆 mbedtls	64⊖ /*	
> 🛃 mmcau	65 * Port number the Green Grass Discovery use for JSON retrieval from cloud is using.	
> 🛃 se_hostlib	00 -/ ientcredentialGRFFNGRASS DISCOVERY PORT 8443	
Source	68	
> in app.n	69® /*	
> B aws_bunepoor_comg.n	70 * WIFI network to join.	
h avs clientcredential.h	72 #define clientcredentialWIFI SSID "Paste WiFi SSID here."	
> In aws demo.h	73	~
N aws int confin h	<	>
Qui 💱 🕪 - Vari 📴 Outl 🗣 Bre 🕪 - Glo 🔍 🗖	🍈 Installed SDKs 📅 Properties 🖹 Problems 🖸 Console 💱 🧬 Terminal 🔐 Image Info 🔋 Memory: 40- Heap and Stack Usage 😭 Debugger Console 🗹 📮 • No consoles to display at this time.	• 📑 • • •
Project: frdmk64f_se_SE05x_cloud_aws [Debug]		
 Create or import a project 		
Import SDK example(s) Import project(s) from file system		
 Build your project 		
Suild Clean		
/frdmk64f se SE05x cloud aws/source/aws clientcredential.h	(i) NYD MK6dFN1M0vvv12 ffre	

Figure 41. Set the Rest API Endpoint and thing name in your project settings

4. On the same **Interact** menu, you will find MQTT topics that enable applications and things to get, update, or delete the state information for an AWS thing. For instance, copy the MQTT **update** topic as shown in <u>Figure 42</u>:

aws	Services 🗸 Res	source Groups 🗸	*			۵	jordi.jofre 👻	Ohio 👻	Support 👻
÷	Things > my_thing								Q
-	THING my_thing								© &
	NO TYPE						Acti	ons -	
	Details	This thing already	appears to be connected.				Connect a d	evice	
	Security Thing Groups	HTTPS							
	Billing Groups	Update your Thing	g Shadow using this Rest A	API Endpoint. Learn more					
	Shadow	an9ryhd19gvf	fr-ats.iot.us-east-	2.amazonaws.com					
	Activity	MQTT							
	Jobs Violations	Use topics to enab	ole applications and things	s to get, update, or delete the state	e information fo	or a Th	ning (Thing Shad	ow)	
	Defender metrics	Update to this thi	ing shadow						
		\$aws/things/	/my_thing/shadow/up	date					
		Update to this thi	ing shadow was accepted						
		\$aws/things/	/my_thing/shadow/up	date/accepted					
Figure	42. Find y	our MQT	T <i>Update</i> t	opic					

5. Go to the AWS demo in your MCUXpresso workspace. Navigate to the aws_jitr_task_lwip.c file located in frdmk64f_se_SE05x_cloud_aws
\source folder. Replace the #define PUB_TOPIC variable with the MQTT topic
you obtained in Figure 41 as shown in Figure 42.

workspace - frdmk64f_se_SE05x_cloud_aws/source/aws_jitr_tas	k_lvip.c - MCUXpresso IDE -	
Edit Source Refactor Navigate Search Project Co	nfigTools Run Analysis FreeRTOS Window Help	
💌 🔜 🐚 🕶 🗞 🕶 🛗 🗇 🏷 🕥 🕶 🛄 🔌	● ● パ ふ つ 点 三 愛 ● ● 風 巻 成 ● マ 🕹 🖋 📕 🍢 🎋 🌴 🗸 🖉 マ 🖉 🖉 マ 🚺 🖬 🐘 気 巻 成 ● マ ウ -	Ŧ
	Ouick Accord	-
	Concerned 1	10
Project 🛛 😤 Periphe 🔢 Registers 🏇 Faults 📟	O Welcome Aus_clientcredential.h C aws_ittr_task_lwip.c	•
- 🔶 m & M -	2 82 sss sscn session t "sscn session:	
	83 sscp_context_t_sscp;	
	84	
> amazon-meerios	850/***	
> Co board	00 UPTINITIONS	
> Component	88 #define SUB TOPIC "Saws/things/" clientcredentialIOT THING NAME "/shadow/update/accepted"	
> CO device	2 #define PUB TOPIC "Saws/things/my thing/shadow/update"	
OB Ehr	90	
CO huiz	91 /* Used to avoid warnings in case of unused parameters in function pointers */	
> CB mhadtle	92 #define 101_UNUSED(x) (Vold)(x)	
> Contractions	94 * types	
off as heatlik	95 ******	
S Se_nosino	96	
Source	970/************************************	
> a app.n	98 * Prototypes	
> a avs_burrerpool_coning.n	100 int HandleReceivedMsg(char "sJsonString, uint16 t len);	
> aws_clientcredential_keys.n	101 void awsPubSub(const UB ax_uid[A71CH_MODULE_UNIQUE_ID_LEN]);	
> avs_clientcredential.n	102	
> a aws_deno.n	103 static MQTTBool_t prvMQTTCallback(void "pvUserData, const MQTTPublishData_t "const pxCallbackParams);	
> aws_ite_comigat		
> c) ave lite task being	106 * Variables	
h ave mott agent configh	v <	>
torm 23 we varian BE Outlan 40 brean We Glow	🔟 Installed SUKS 🔛 Properties 🤮 Problems 😂 Console 23 🕉 Terminal 📷 Image Into 🕛 Memory 💷 Heap and Stack Usage 🙀 Debugger Console 🗂 单 🕛	3
	No consoles to display at this time.	
	n	
MCUXpresso IDE - Quickstart Panel		
Project: frdmkb4f_se_SE05x_cloud_aws [Debug]		
Create or import a project		
New project		
Import SDK example(s)		
Import project(s) from file system		
Build your project		
O Build		
😪 🧹 Clean	v	
	Su Writable Smart Isert 1:1 U NXP MK64FN1M0xx12 (frdmk.	d aws)

6. Finally, we need to tell the FRDM-K64F board which credentials to use. Recall from <u>Section 3.5</u> that we are using key ID 0xF0000000 and certificate ID 0xF0000001. Therefore, navigate to the aws iot config.h file located in

frdmk64f_se_SE05x_cloud_aws\source folder and set the #define lines
accordingly as shown in Figure 44.

vorkspace4 - frdmkb4t_seUbx_cloud_aws/source/aws_iot_co	onfig.h - MCUXpresso IDE	- 0
Edit Source Refactor Navigate Search Project	ConfigTools Run Analysis FreeRTOS Window Help	
• 🔛 🕼 🛞 • 🗞 • 🛗 🞺 🐃 🚸 • 💽 •	• 💁 + 🦢 🖉 + 📝 🖉 🗑 👖 💷 🔌 10- 10 🖷 12 N (2, 10- 10) 🗮 3, 10 A (2, 10) 👘 3, 10 A (2, 10) N (2, 10) A (2,	▼ 🛉 ▼ 🗘 🌣 ↔ · ↔ ·
		Q 🔡
mia 💱 🖓 Darin IIII Panir 🗰 Eaultr 🖓 🗖	Rauer client credential lear h Rauer jitt tack luin c Rauer jet confict 2 Rauer jet co	_
	en e	
E 🔊 A A . A 🕅 🔺 8	22 #ifndef_ANS_JITR_IOT_CONFIG_H	
🐸 freertos 🔨 🔨	23 HOFTINE_PARS_JIN_IO_CONFIG	
IDS IDS	25 #if defined(SSS_USE_FTR_FILE)	
C mbadtle	26 #include "fsl_sss_ftr.h"	
rinceaus maio	27 meise 28 minclude "fsl sss ftr default h"	
A mmcau	29 Wendif	
A phy	30	
😕 se_hostlib	310// Get from console	
29 source	33 #define AWS IOT MOTT CLIENT ID "c-sdk-client-id2" ///< MOTT client ID should be unique for every device	
> h aws_client_credential_keys.h	34	
> h aws_demo.h	35 /* doc start meskewids */	
> h aws_iot_config.h	30 #define SSS_KEPPAR_INDEX_LIENT_PRIVATE 0xf00000000	
> c aws_jitr_demo_Main_lwip.c	38 Martine 535 Levin terret new Criterin an addaded	
a nur jitr tark huin c		
> Fel aux lin rank unbre	39	
> h FreeRTOSConfig.h	40 // =====	
bright data minute bright data minut	39 40 //	
h FreeRDSConfig.h h fsl_sss_ftr.h h ist_config.common.h	40 //	
A freeROSConfig.h A fst_sss_ftr.h A iot_config_common.h E semihost_hardfault.c	<pre>//</pre>	a publish is done. Th
	 // //	a publish is done. Th
	19 J/ /	a publish is done. Th bigger than this buffn
Comparison of the second	39 //	a publish is done. Th bigger than this buff > ↓ @ ➡ □ ▾ ➡ ♥ ▾ =
) @ FerdDSConfigh) @ fold_sig_temp) @ fold_sig_temp) @ tet_config_commentA) @ tempet_temp #ranp > @ tamp_mk6412c ↓ (cstart_ % 10> Variables �e_Breakpoints	A // // TTT PubSub // Try PubSub // // Try PubSub // // Complete a message is sent out through the MQTT layer. The message is capied into this buffer anytime // Complete a message is sent out through the MQTT layer. The message is capied into this buffer anytime // Complete a message is capied into the device should be less than this buffer size. If a received message is // Complete a message is a message is a message is capied into the device should be less than this buffer size. If a received message is // Complete a message is a message into the device should be less than this buffer size. If a message is // Complete a message is a message into the device should be less than this buffer size. If a message is // Complete a message is a message into the device should be less than this buffer size. If a message is // Complete a message is // Complete a message into the device should be less than this buffer size. If a message is // Complete a message is a message into the device should be less than this buffer size. If a message is // Complete a message is // Complete a message into the device should be less than this buffer size. If a message is // Complete a message into the device should be less than this buffer size. If a message is // Complete a message is // Complete a message into the device should be less than the device should be less	a publish is done. Th bigger than this buff
characteristic of the second		a publish is done. Th bigger than this buff
DirentDSConfig.	30 // ***********************************	a publish is done. Th bigger than this buff. > @ [e] • • • • • •
MCUXpresso IDE - Quickstart Panel	39 // // ********************************	a publish is done. Th bigger than this buff
B Treat/DSConfig.h B Stars, bh B	1 // TTT PubSub 31 // TTT PubSub 32 /// Any Tty_BUF_LEN \ 43 Staffine Ask JOT YOT TX_BUF_LEN \ 44 Staffine Ask JOT YOT TX_BUF_LEN \ 45 Staffine Ask JOT YOT TX_BUF_LEN \ 46 Staffine Ask JOT YOT TX_BUF_LEN \ 46 Staffine Ask JOT YOT TX_BUF_LEN \ 47 Staffine Ask JOT YOT TX_BUF_LEN \ 47 Staffine Ask JOT YOT TX_BUF_LEN \ 48 Staffine Ask JOT YOT TX_BUF_LEN \ 48 Staffine Ask JOT YOT TX_BUF_LEN \ 49 Staffine Ask JOT YOT TX_BUF_LEN \ 49 Staffine Ask JOT YOT TX_BUF_LEN \ 49 Staffine Ask JOT YOT TX_BUF_LEN \ 40 Staffine Ask JOT YOT TX_BUF_LEN \ 41 Staffine Ask JOT YOT TX_BUF_LEN \ 42 Staffine Ask JOT YOT TX_BUF_LEN \ 43 Staffine Ask JOT YOT TX_BUF_LEN \ 43 Staffine Ask JOT YOT TX_BUF_LEN \ 44 Staffine Ask JOT YOT TX_BUF_LEN \ 44 Staffine Ask JOT YOT TX_BUF_LEN \ 45 Staffine Ask JOT YOT	a publish is done. Th bigger than this buff
Construction of the second seco	39 // // ********************************	a publish is done. Th bigger than this buff
B TreatTOSConfig.s. B Idays, Shuther Schemer	<pre>4 // // TTT PubSub 4 // // TTT PubSub 5 // // ATT PubSub 5 // ATT PubSub 5</pre>	a publish is done. Th bigger than this bufff
TreefT05Config. T	39 // // ********************************	a publish is done. Th bigger than this buff
B Treff03Configh B T	1 //	a publish is done. Th bigger than this buff
Construction of the set of	35 // // ********************************	a publish is done. Th bigger than this buff Ø ☎ ⓑ ▾ ☶ ▾ ≕]
<pre>> @ Treff05Confg.h >> [d tass,th.h >> [d tass,th.h] >> [d tass,th.h] CUXpresso IDE - Quickstart Panel Project: TamkBf #0%_cloudnk([dbug] set or Import 30% cample(t) >> [d theory 50% cample(t) >> [d moot spicet.]</pre>	32 // //	a publish is done. Th bigger than this buff 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Constraints Constrai	30 // ***********************************	a publish is done. Th bigger than this buff Ø ☎ 🖬 ▾ ㎡ ▾ ≕]
> @ TerritOScongs. >> Id Jass that >> id Jass that >> id Jass that >> id semiled Jasfault & # semiled Jasfault & # startup >> id startup.mik4H2c > id startup.m	32 // //	a publish is done. Th bigger than this buff 00 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Constraints Constrai	3 // //	a publish is done. Th bigger than this buff Ø ₫ ♥ + ঊ + =]

Figure 44. Setting the credential IDs

Everything is now fully ready to run the demo on the FRDM-K64F board. Please jump back to <u>Section 3.9</u> to execute the demo project and verify that everything is running as expected.

3.9 AWS IoT Core project execution

Now we are fully ready to run the project on the FRDM-K64F. To start the AWS IoT Core project example, follow these steps:

- 1. Subscribe to the MQTT topic from <u>Figure 87</u>. Go to the AWS IoT Core dashboard and follow the steps indicated in <u>Figure 45</u>:
 - a. Go to Test.
 - b. Go to Subscribe to a topic.
 - c. Write the MQTT topic name in the Subscription topic field.
 - d. Click on *Subscribe to topic* button.

aws Ser	rvices 🗸 Resource Groups 🗸	★
AWS IOT	MQTT client @	⑦ Connected as iotconsole-1563286728072-2 ▼ ⑦
Monitor	Subscriptions	
Onboard Manage Greengrass Secure Defend Act Test a Software Settings Learn	Subscribe to a topic Publish to a topic	Subscribe Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages. Subscription topic Subscription topic Too Max message capture ? 100 Quality of Service ? 0 - This client will not acknowledge to the Device Gateway that messages are received 1 - This client will acknowledge to the Device Gateway that messages are received 1 - This client will acknowledge to the Device Gateway that messages are received 1 - This client will acknowledge to the Device Gateway that messages are received 2 Auto-format JSON payloads (improves readability) 2 Display payloads as strings (more accurate) 2 Display raw payloads (in hexadecimal)
Figure 45.	Subscribe to t	the MQTT topic

2. The MQTT topic you subscribed will now appear in the *Subscriptions* section as shown in Figure 46:

aws Service	is 🗸 Resource Groups 🗸 🕻	Å jordi.jofre → Ohio → Support →
AWS IOT	MQTT client ③	Connected as iotconsole-1563286728072-2 •
Monitor	Subscriptions	\$aws/things/my_thing/shadow/update Export Clear Pause
Manage Greengrass Secure Defend Act Test	Subscribe to a topic Publish to a topic Saws/things/my_thing/sha *	Publish Specify a topic and a message to publish with a QoS of 0. Saws/things/my_thing/shadow/update Publish to topic
Figure 46. C	Check MQTT topic s	subscription

- Laptop

 Image: Contract FRDM-K64F board
- 3. Connect FRDM-K64F OpenSDA port, K64F port and Ethernet interface to your laptop as shown in Figure 47:

4. Open TeraTerm, go to Setup > Serial Port and choose the one corresponding to the OpenSDA port of the board, 115200 baud rate, 8 data bits, no parity and 1 stop bit and click OK as shown below.

Tera Term: New co	nnection		×	Port:	COM7 ~	
O TCP/IP	Host:	myhost.example.com	✓	Baud rate:	9600	0
	0 !	History TCP port#: 22		Data:	110	Car
	Service;	SSH SSH version: SSH2	2 ~	Parity:	600 1200	
		O Other Protocol: UNSP	FC ×	Stop:	2400	He
		0101		Flow control:	9600	
Serial	Port:	COM3: Intel(R) Active Managemen	t Te 🗸	Transmit delay	19200	
	OK	COM3: Intel[R] Active Managemen COM7: USB Serial Device (COM7)	t Technology - SOL (COM3)	0 msec/	57600	msec/line
	UK	COM9: Virtual Com Port (COM9)			230400	
			¥		921600	

 Go to the MCUXpresso Quickstart Panel and click *Debug* button, wait a few seconds until the project executes and click on *Resume* to allow the software to continue its execution as shown in <u>Figure 49</u>:



535216

- 6. Your device should now be connected to AWS. Check that your device is connected by:
 - a. Checking the TeraTerm logs as shown in Figure 50.

🔟 COM5 - Tera Term	VT	-	×
File Edit Setup Co	ntrol Window Help		
App:INFO sss:INFO sss:WARN sss:WARN	PlugAndTrust_02.11.03_20191107 atr (Len-35) 06 40 60 60 60 61 63 E8 60 FE 62 6B 63 E8 68 11 60 60 60 60 64 60 60 60 40 43 4F 50 34 20 41 4 50 4F Communication channel is Plain. i!!Not recommended for production use.!!!		^
App:INFO	-HWS SUBSCFIDE DUDIES EXAMPLE :Connecting to network		
Initializing PHY App:INFO	Getting IP address from DHCP		
App:INFO IPv4 Address			
App:INFO :			
App:INFO App:INFO	MQTT attempting to connect MQTT attempting to connect to 'c-sdk-client-id9'		
sss:INFO	Group id found - MBEDTLS_ECP_DP_SECP256R1		
App:INF0	MQTT Echo demo subscribed to \$aws/things/myTestThing/shadow/update/accepted		
App:INFO App:INFO	z—>sleep Echo successfully published		
App:INFO	Echo successfully published		
App:INFO App:INFO	:>sleep Echo successfully published		
App:INFO :	Echo successfully published		
App:INFO			
Figure 50	Device connection to AWS		

b. Checking the last time the device was seen in the AWS dashboard as shown in Figure 51.

IoT	Monitor		Sample period Time range One day Week
er ard	Successful connections		
d			•
	00.50 36.50 Jan 8 06.50 36.50 Jan 9 06.50	1600 Jan 10 08:00 16:00 Jan 11 08:00 16:00 Jan 12	98.59 16.99 Jan 12 98.59 16.59 Jan 14
	Messages		
	Protocol • erry • erry	Type * Game * Game * Game	Drector • Name • Adment
are gs	Messages published		
	• 1 1 1 1 1 1 1		

c. Checking the messages published in the subscribed MQTT topic as shown in Figure 52:

AN12404

EdgeLock[™] SE05x for secure connection to AWS IoT Core

aws Services -	Resource Groups 🗸 🔸	众 jordi.jofre → Ohio → Support →
	MQTT client ③	Connected as iotconsole-1563286728072-2 •
Monitor	Subscriptions	\$aws/things/my_thing/shadow/update Export Clear Pause
Onboard Manage Greengrass Secure Defend Act Test	Subscribe to a topic Publish to a topic \$aws/things/my_thing/sha ×	Publish Specify a topic and a message to publish with a QoS of 0. Saws/things/my_thing/shadow/update Publish to topic 1 ("message": "Hello from ANS IOT console" 3)
		\$aws/things/my_thing/stadow/update Jul 16, 2019 3:29:26 PM +0200 Export Hide
	-) \$ sws/things/my_thing/shadow/update Jul 16, 2019 3:29:26 PM +0200 Export Hide { "***********************************
		י \$aws/things/my_thing/shadow/update או 16, 2019 3:29:24 PM +0:200 Export Hide
		3 \$aws/things/my_thing/shadow/update Jul 16, 2019 3:29:24 PM +0200 Export Hide
Figure 52. De MQTT topic	evice connection	to AWS - Published messages in the <i>update</i>

4 Appendix: Registering a CA certificate for just-in-time registration

Alternatively to the procedure explained in <u>Section 3</u>, you can configure a CA certificate to enable device certificates it has signed to register with AWS IoT automatically the first time the device connects to AWS IoT. To register device certificates when a client connects to AWS IoT for the first time, you must enable the CA certificate for automatic registration and configure the first connection by the device to provide the required certificates.

This section generates an injects your own credentials in EdgeLock SE05x using the provisioning scripts included as part of EdgeLock SE05x Plug & Trust Middleware. Please use this procedure only if you prefer to generate your own keys instead of leveraging the EdgeLock SE05x ease of use configuration used in <u>Section 3</u>.

Note: The key generation and injection procedure described in this section is only applicable for **evaluation** or **testing** purposes. In a commercial deployment, key provisioning must take place in a trusted environment, in a facility with security features such as tightly controlled access, careful personnel screening, and secure IT systems that protect against cyberattacks and theft of credentials.

4.1 Running AWS IoT Core key provisioning scripts

This section explains how to generate the credentials for the EdgeLock SE05x using the key provisioning scripts included in EdgeLock SE05x Plug & Trust Middleware and a FRDM-K64F board as a host platform. These credentials are required for the device onboarding into AWS IoT Core.

Note: Check <u>AN12396- Quick start guide to Kinetis K64</u> for detailed instructions on how to bring up the FRDM-K64F board.

4.1.1 Download EdgeLock SE05x Plug & Trust Middleware

Follow these steps to download the EdgeLock SE05x Plug & Trust Middleware in your local machine:

1. Download EdgeLock SE05x Plug & Trust Middleware from the NXP website.

AN12404

• 🔿 • 🛧 🏷 > Th	iis PC > Local Disk (C:)			
Terres 🛛 🖈 ^	Name	Date modified	Туре	Size
🜗 Music	Intel	2/25/2019 4:12 AM	File folder	
Projects		3/7/2019 1:28 AM	File folder	
📓 Videos	PEMicro	3/7/2019 1:34 AM	File folder	
	PerfLogs	4/11/2018 4:38 PM	File folder	
Chebrive	Program Files	3/11/2019 4:05 AM	File folder	
🧢 This PC	Program Files (x86)	3/11/2019 3:28 AM	File folder	
3D Objects	Projects	3/11/2019 6:17 AM	File folder	
늘 Desktop	Python27	3/11/2019 4:53 AM	File folder	
Documents	se050_middleware	3/11/2019 6:27 AM	File folder	
Downloads	Users	2/25/2019 5:06 AM	File folder	
Music	Windows	3/11/2019 3:38 AM	File folder	
Pictures	Recovery	2/25/2019 12:46 PM	Text Document	01
Videos				
Local Disk (C:)				
🥪 USB DISK (E:)				

2. Create a folder called **se050_middleware** in C: directory as shown in Figure 53:

 Unzip the EdgeLock SE05x Plug & Trust Middleware inside the se050_middleware folder. After unzipping, you will see a folder called simw-top created. The contents of the simw-top directory should look as shown in Figure 54:

🕂 🔿 👻 🕇 📙 > This PC >	OS (C:) > se050_middleware > simw-top -				Search simw-top	۶
	Name	Date modified	Туре	Size		
A Quick access	akm	7/18/2019 5:26 PM	File folder			
MobileKnowledge	binaries	7/18/2019 5:26 PM	File folder			
T1: 00	demos	7/18/2019 5:26 PM	File folder			
	doc	7/18/2019 5:27 PM	File folder			
🕳 USB Drive (D:)	ext	7/18/2019 5:27 PM	File folder			
A Matural	hostlib	7/18/2019 5:27 PM	File folder			
VINETWORK	projects	7/18/2019 5:27 PM	File folder			
Network	pycli	7/18/2019 5:27 PM	File folder			
	scripts	7/18/2019 5:27 PM	File folder			
	sss	7/18/2019 5:27 PM	File folder			
	tools	7/18/2019 5:27 PM	File folder			
	CMakeLists.txt	7/18/2019 5:26 PM	Text Document	3 KB		
	EULA.pdf	7/18/2019 5:26 PM	Adobe Acrobat D	134 KB		
	PlugAndTrustMW.pdf	7/18/2019 5:26 PM	Adobe Acrobat D	11,252 KB		
	README.First.bt	7/18/2019 5:26 PM	Text Document	1 KB		
	🛃 Third Party License.pdf	7/18/2019 5:26 PM	Adobe Acrobat D	369 KB		
	version_info.txt	7/18/2019 5:26 PM	Text Document	1 KB		

Note: It is recommended to keep *se050_middleware* with the <u>shortest</u> path possible and <u>without spaces</u> in it. This avoids some issues that could appear when building the middleware if the path contains spaces.

4.1.2 Flash FRDM-K64F with VCOM software

The VCOM software allows the FRDM-K64F board to be used as a bridge between the Windows machine and the EdgeLock SE05x and enables the execution of the EdgeLock SE05x ssscli tool and other utilities from the laptop. To flash the VCOM software into the FRDM-K64F, follow these steps:

1. Unplug and plug again the USB cable to the openSDA USB port as shown in <u>Figure 55</u>:



2. When you plug the board, your laptop should recognize the board as an external drive as shown in <u>Figure 56</u>:

File Home Share Vi	ew Manage				
$\leftrightarrow \rightarrow \checkmark \uparrow \blacksquare \Rightarrow FRDM-K6$	4FD (E:)				
	^ Name	Date modified	Туре	Size	
A Quick access	DETAILS.TXT	3/22/2016 4:30 PM	Text Document	1 KB	
MobileKnowledge	PRODINFO.HTM	3/22/2016 4:30 PM	HTM File	1 KB	
This PC					
> 🗊 3D Objects					
> Desktop					
> 🗄 Documents					
> 🕹 Downloads					
> 👌 Music					
> 📰 Pictures					
> 📑 Videos					
> 🏥 OS (C:)					
> 🔜 USB Drive (D:)					
> 🛖 FRDM-K64FD (E:)					
🕳 FRDM-K64FD (E:)					

 Flash the VCOM software to FRDM-K64F. The VCOM software binary can be found in the EdgeLock SE05x Plug & Trust Middleware package, inside the simw-top \binaries folder as shown in Figure 57:



4. Drag and drop or copy and paste the a7x_vcom-T1oI2C-frdmk64f-SE050x.bin file into the FRDM-K64F drive from your computer file explorer as shown in Figure 58:

← → · · · ↑ 🛋 > FRDM-K64FD (E:)				
	Date modified 3/22/2016 4:30 PM 3/22/2016 4:30 PM	Type Text Document HTM File	Size 1 KB 1 KB	

- 5. The serial and VCOM ports should be recognized by your Device Manager. To check that the ports are recognized, follow the steps indicated in Figure 59:
 - a. Unplug the USB cable from the OpenSDA USB port.
 - b. Plug the USB cable to the OpenSDA USB port.
 - c. Check that the serial port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as USB Serial Device (COM7) but this naming might change depending on your computer. Therefore, it is important that you

identify which device is recognized at the moment you plug the SDA USB port to the computer.

- d. Plug the USB cable to the K64F USB port.
- e. Check that the VCOM port is recognized in the category **Ports (COM & LTP)**. In this document, it is recognized as *Virtual Com Port (COM8)* but this naming might change depending on your computer (e.g. It could also appear named as *USB Serial Device*). Therefore, it is important that you identify which device is recognized at the moment you plug the K64F USB port to the computer.



Note: Please note that it is possible that either of the two COM ports is not detected when using low-quality or charge-only USB cables.

4.1.3 Key and certificate configuration for use with AWS IoT Core

The EdgeLock SE05x Plug & Trust Middleware includes an executable file that allows you to easily generate some sample credentials and inject them into the EdgeLock SE05x for their use with this AWS IoT Core demo.

On the other hand, it is also possible to use the pre-provisioned credentials that are already in the EdgeLock SE05x for this purpose thanks to the Ease of Use configuration. However, this method requires an AWS feature called 'Multi-Account Registration'. If you wish to configure the credentials with the Ease of Use configuration, please skip ahead to <u>Section 3.5</u>.

To externally generate the keys and inject them into the EdgeLock SE05x, follow these steps:

AN12404

1. Mount OM-SE050ARD on top of the FRDM-K64F. Then, connect FRDM-K64F OpenSDA port and K64F port to your laptop as shown in <u>Figure 60</u>



2. Go to simw-top\binaries\pySSSCLI folder and locate the Provision_AWS.exe file as shown in Figure 61:

→ * ↑ → This PC >	OS (C:) \rightarrow se050_middleware \rightarrow simw-top \rightarrow binaries	> pySSSCLI		ڻ ~	Search pySSSCLI	
This PC	^ Name ^	Date modified	Туре	Size		
3D Objects	cryptography.hazmat.bindingsconstant	20/02/2020 09:29	Python Extension	19 KB		
Desktop	cryptography.hazmat.bindingsopenssl	20/02/2020 09:29	Python Extension	2,225 KB		
Documents	cryptography.hazmat.bindingspadding	20/02/2020 09:29	Python Extension	20 KB		
Downloads	error_log	20/02/2020 09:33	Text Document	1 KB		
Musia	libcrypto-1_1.dll	20/02/2020 09:29	Application exten	1,866 KB		
	libgcc_s_dw2-1.dll	20/02/2020 09:29	Application exten	917 KB		
Pictures	library	20/02/2020 09:29	Compressed (zipp	6,655 KB		
Videos	libssl-1_1.dll	20/02/2020 09:29	Application exten	397 KB		
🗎 OS (C:)	output_log	20/02/2020 09:33	Text Document	0 KB		
USB Drive (D:)	Provision_AWS	20/02/2020 09:29	Application	33 KB		
FRDM-K64FD (E:)	Provision_AZURE	20/02/2020 09:29	Application	32 KB		
	Provision_GCP	20/02/2020 09:29	Application	32 KB		
FRUM-R04FU (E:)	Provision_IBM	20/02/2020 09:29	Application	32 KB		
USB Drive (D:)	📄 pyexpat	20/02/2020 09:29	Python Extension	163 KB		
10 - EMVCo deck	python37.dll	20/02/2020 09:29	Application exten	3,515 KB		
lcons	select	20/02/2020 09:29	Python Extension	23 KB		
MK101	sssapisw.dll	20/02/2020 09:29	Application exten	298 KB		
others						

Figure 61. Find Provision_AWS.exe file in your EdgeLock SE05x Plug & Trust Middleware package

3. Open a command prompt

- 4. Use the Provision_AWS.exe executable to generate and inject keys into your EdgeLock SE05x. You cal follow these steps shown in Figure 62:
 - a. Go to the folder simw-top\binaries\pySSCLI and run >cd C:
 \se050_middleware\simw-top\binaries\pySSSCLI
 - b. Run the executable Provision_AWS.exe <K64_COM_port_number>. For that, you also need to indicate the VCOM port number corresponding to the K64 USB port of your board (See <u>here</u>).
 Sond > Drawing ion_AWS.exe_COM8

Send > Provision_AWS.exe COM8

- c. Check that the keys are generated and injected.
- d. Check that the program execution completes successfully.

🖾 Command Prompt Microsoft Windows [Version 10.0.17134.885]			– 🗆 X
(c) 2018 Microsoft Corporation. All rights	reserved.		
C:\Users\Jordi Jofre>cd C:\se050_middlewar	e\simw-top\binaries\pyS	sscli	
C:\se050_middleware\simw-top\binaries\pySS ===================================	SCLI>Provision_AWS.exe		
# SUBSYSTEM : se050 # CONNECTION_TYPE : vcom # CONNECTION_PARAMETER : COM8 #		 ✓ Ports (COM & LP) ♀ Intel(R) Active ♀ USB Serial Dev ♀ Virtual Com P 	T) Management Technology - SOL (COM3) <u>rice (COM7)</u> ort (COM8)
Opening COM Port '\\.\COM8'			
00 A0 00 00 01 00 00 00 54 50 4F	03 96 04 03 00 64 00 00	E8 00 FE 02 0A 4A 43 4F	08 03 E8 08 50 34 20 41
SSS:WARN :!!!!Not recommended for pr			

#			
SUBSYSTEM : se050 CONNECTION_TYPE : vcom CONNECTION PARAMETER : COM8			
ŧ			
//////////////////////////////////////	*****		
sss:INFO :atr (Len=35)			
			0B 03 E8 08
01 00 00 00			50 34 20 41
54 50 4F sss:WARN :Communication channel is			
001e41d4004230ec953			
ey pair file: C:\se050_middleware\simw-tc njecting ECC key pair at key ID: 0x201810 wccessfully Injected ECC key pair.	p\binaries\pySSSCLI\aws 05	\37782420273824681104417	9_device_key.pem
ertificate file: C:\se050_middleware\simw	-top\binaries\pySSSCLI\	aws\37782420273824681104	4179_device_certificate.cer
Injecting certificate at key ID: 0x2018100	7		
Creating ECC Reference key from key ID: 0x	20181005		С
riting to file in pem format			
buccessfully Created reference key at: C:\ a reference key nem	se050_middleware\simw-to	op\binaries\pySSSCLI\aws	\377824202738246811044179_devic
Program Completed Successfully	#		d
F I M M M M M M M M M M M M M M M M M M M	# ####################################		
:\se050_middleware\simw-top\binaries\pySS	SCLI>		
Igure 62. Run Provisio	n AWS.exe ex	ecutable	

5. Go to simw-topbinariespysscllaws folder and check that the keys appear inside the folder as shown in Figure 63:

$\vdash \rightarrow \uparrow \uparrow \square \rightarrow \text{This PC} \rightarrow 0$	JS (C:) > seUSU_middleware > simw-top > binaries > p	ySSSCLI > aws		V O Search a
💻 This PC	Name ^	Date modified	Туре	Size
🧊 3D Objects	377822231953814008977749_device_certificate	20/02/2020 09:33	Security Certificate	1 KB
E Desktop	377822231953814008977749_device_key	20/02/2020 09:33	CMS (S/MIME) File	1 KB
Documents	377822231953814008977749_device_reference_key	20/02/2020 09:33	CMS (S/MIME) File	1 KB
🕹 Downloads	▲ prime256v1	20/02/2020 09:33	CMS (S/MIME) File	1 KB
h Music	a rootCA_certificate	20/02/2020 09:33	Security Certificate	
Dictures	rootCA_key	20/02/2020 09:33	CMS (S/MIME) File	KB
Videos	v3_ext_device.cnf	20/02/2020 09:29	CNF File	1 KB
🛀 OS (C:)				
USB Drive (D:)				

After injecting the credentials, go to Section 4.2

4.2 Register root certificate authority (CA)

This section describes how to register the root CA certificate with AWS IoT Management Console. For the sake of simplicity, this application note only uses the AWS IoT Management Console web interface. For details on how to perform any of these steps using other tools provided by AWS, refer to AWS Core IoT documentation.

Note: The AWS IoT Core account preparation procedure is the same independently of the MCU / MPU platform you choose for evaluation purposes.

4.2.1 Get registration code from AWS

AWS IoT Core requires the registration of a CA certificate used to sign and issue your device certificates. This CA certificate is used for authentication of devices attempting to connect to the platform thereafter. As part of the CA certificate registration process, AWS IoT Core performs a proof-of-possession verification. This proof-of-possession mechanism ensures that the uploader of the CA certificate also knows the associated private key. The proof-of-possession mechanism consists of generating a verification certificate using:

• The CA certificate

- The CA private key
- A registration code given by AWS.

To generate the verification certificate, the AWS IoT Core registration code needs to be set in the Common Name field of the verification certificate signed by the CA certificate private key.

As a first step, we need to obtain the AWS registration code assigned to our account. For the sake of simplicity, this application note only uses the AWS IoT Management Console web interface. For details on how to perform any of these steps using other tools

AN12404	All information provided in this document is subject to legal disclaimers.
Application note	Rev. 1.6 — 15 January 2021
	535216

provided by AWS, refer to <u>AWS Core IoT documentation</u>. To obtain AWS registration code follow these steps:

1. On the menu of the left hand side of the AWS IoT Core dashboard, go to **Secure**, select **Certificates** and click **Create a certificate** as shown in Figure 64:

aws Services - Resource (Broups 🗸 🔥
AWS IOT	
Monitor	
Onboard	
Manage	
Secure	
Certificates	
Policies	
CAs	
Role Aliases	
Authorizers	You don't have any certificates yet
Defend	
Act	Certificates help things establish a secure connection.
Test	Learn more Create a certificate
Figure 64. AWS IoT certi	ficates menu

AWs IoT Core supports three options:

- **One-click certificate creation**: AWS IoT Core generates an individual certificate and their associated public and private keys for a device.
- **Create a CSR:** OEM generates a device key pair and generates a certificate signing request (CSR). This CSR is signed by AWS IoT Core certificate authority (CA).
- **Use my certificate:** OEM generates the device certificates. In this option, the OEM needs to register the CA certificate that signed and issued the device certificates. This document describes the *Use my certificate* option.

2. Select the Use my certificate option as shown in Figure 65:



3. Click *Register CA button* as shown in Figure 66:

orde	er to use your own certificates for one or many devices	you must use a registered CA	certificate.	
jiste	ered CAs			
Q	Search CA certificate			
	No match found There is no CA certificate in your account.			
	Register CA			

4. You will see a form with instructions to register a CA certificate. Go to step 2, and copy the **registration code** as shown in <u>Figure 67</u> for later use.



4.2.2 Generate AWS verification certificate

The EdgeLock SE05x Plug & Trust Middleware includes a Python script called verification_certificate.py that generates the AWS verification certificate. This scripts needs three arguments:

- The path to the root CA certificate
- The path to the root CA private key
- The AWS registration code

To generate the AWS verification certificate, follow these steps:

 Go to C:\se050_middleware\simw-top\pycli\Provisioning folder and find the file verification certificate.py as shown in Figure 68:

ightarrow 🕆 📙 > This PC	> OS (C:) > se050_middleware > simw-top >	pycli > Provisioning		ٽ ~	Search Provisioning	
This PC	^ Name ^	Date modified	Туре	Size		
3D Objects	aws	20/02/2020 09:29	File folder			
Desktop	azure	20/02/2020 09:29	File folder			
Documents	error_log	20/02/2020 09:39	Text Document	1 KB		
	GenerateAWSCredentials	20/02/2020 09:29	PY File	2 KB		
Music	GenerateAZURECredentials	20/02/2020 09:29	PY File	2 KB		
	GenerateGCPCredentials	20/02/2020 09:29	PY File	2 KB		
Pictures	GeneratelBMCredentials	20/02/2020 09:29	PY File	2 KB		
Videos	output_log	20/02/2020 09:39	Text Document	0 KB		
🛄 OS (C:)	readme.rst	20/02/2020 09:29	RST File	5 KB		
SB Drive (D:)	ResetAndUpdate_AWS	20/02/2020 09:29	PY File	2 KB		
- FRDM-K64FD (E:)	ResetAndUpdate_AZURE	20/02/2020 09:29	PY File	2 KB		
	ResetAndUpdate_GCP	20/02/2020 09:29	PY File	2 KB		
FRDM-K64FD (E:)	ResetAndUpdate_IBM	20/02/2020 09:29	PY File	2 KB		
USB Drive (D:)	verification_certificate	20/02/2020 09:29	PY File	3 KB		
10 - EMVCo deck						
lcons						
MK101						

2. Open a Command prompt

3. Go to C:\se050_middleware\simw-top\pycli\Provisioning folder as shown in Figure 69.

Send >cd C:\se050_middleware\simw-top\pycli\Provisioning

as Command Prompt	-	×
Microsoft Windows [Version 10.0.17134.885] (c) 2018 Microsoft Corporation. All rights reserved.		^
C:\Users\Jordi Jofre>cd C:\se050_middleware\simw-top\pycli\Provisioning		
C:\se050_middleware\simw-top\pycli\Provisioning>		
Figure 69. Go to Provisioning folder		

4. Execute the verification_certificate.py Python script. (Figure 70) Send > Python verification certificate.py

```
certificate.pycertificate.cer> (i.e rootCA_certificate.cer)
certificate.cer> (i.e rootCA_key.pem)
<aws registration code>
```



5. Check that the verification certificate is successfully generated in the **Provisioning** folder in your file system as shown in Figure 71:

→ * ↑ → This PC >	OS (C:) > se050_middleware > simw-top > p	oycli > Provisioning		v ⊙	Search Provisioning	
Quick access	Name	Date modified	Туре	Size		
A Quick access	aws	20/02/2020 09:29	File folder			
MobileKnowledge	azure	20/02/2020 09:29	File folder			
This DC	error_log	20/02/2020 09:39	Text Document	1 KB		
THIS P.C.	GenerateAWSCredentials	20/02/2020 09:29	PY File	2 KB		
Network	GenerateAZURECredentials	20/02/2020 09:29	PY File	2 KB		
	GenerateGCPCredentials	20/02/2020 09:29	PY File	2 KB		
	GeneratelBMCredentials	20/02/2020 09:29	PY File	2 KB		
	output_log	20/02/2020 09:39	Text Document	0 KB		
	readme.rst	20/02/2020 09:29	RST File	5 KB		
	ResetAndUpdate_AWS	20/02/2020 09:29	PY File	2 KB		
	ResetAndUpdate_AZURE	20/02/2020 09:29	PY File	2 KB		
	ResetAndUpdate_GCP	20/02/2020 09:29	PY File	2 KB		
	ResetAndUpdate_IBM	20/02/2020 09:29	PY File	2 KB		
	verification_certificate	20/02/2020 09:29	PY File	3 KB		
	verifyCert	20/02/2020 09:39	Security Certificate	1 KB		

4.2.3 Upload root CA and AWS verification certificate

The registration of the root CA is completed after uploading it together with the AWS verification certificate. To register your root CA, follow these steps indicated in Figure 72:

- 1. Select from your file system the root CA certificate in .cer format (1)
- 2. Select from your file system the verification certificate in .cer format (2)
- 3. Check the boxes Activate CA certificateand Enable auto-registration of device certificates (3)
- 4. Click Register CA Certificate (4)

AWS Services v Resource Groups v 🖈
Register a CA certificate
To use your own X.509 certificates, you must register a CA certificate with AWS IoT. You must prove you own the private key associated with the CA certificate by creating a private key verification certificate. The CA certificate can then be used to sign device certificates. You can register up to 10 CA certificates with the same subject field and public key per AWS account. This allows you to have more than one CA sign your device certificates.
Step 1: Generate a key pair for the private key verification certificate
openssl genrsa -out verificationCert.key 2048
Step 2: Copy this registration code
92bca51e2ac7ec7bec272f4a659bb763bd307c10669003c47c4a92b4190ffbb8
Step 3: Create a CSR with this registration code
openssl req -new -key verificationCert.key -out verificationCert.csr
Put the registration code in the Common Name field
Country Name (2 letter code) [AU]: State or Province Name (full name) [Some-State]: Locality Name (eg, tity) []: Organization Name (eg, company) [Internet Widgits Pty Ltd]: Organizational Unit Name (eg, section 0]]: Common Name (e.g. server FQDN or YOUR name) []: 92bc351e2ac7ec7bec272f4a659bb763bd307c10669003c47c4a9 Email Address []: <
Step 4: Use the CSR that was signed with the CA private key to create a private key verification certificate
openssl x509 -req -in verificationCert.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out veri
Step 5: Upload the CA certificate (rootCA.pem) rootCA_certificate.cer
Step 6: Upload the verification certificate (verificationCert.crt) verifyCert.cer
Activate CA certificate Enable auto-registration of device certificates
Cancel Register CA certificate
igure 72. Upload your root CA and AWS verification certificate

5. When your root CA is registered successfully, it should now be visible in AWS dashboard and appear as *Active* as shown in Figure 73:

aws ser	vices 🗸 Resource Groups 🗸 🛠	
🏠 AWS IOT	Certificate Authorities	Register 🖓
Monitor	Search CAs by CA certificate id Q	Card 👻 🐼
Onboard	77d6dr3ed9drbcba22	
Manage	ACTIVE	
Secure		
Certificates		
Policies		
CAs		
Role Aliases		
Authorizers		
Defend		
Figure 73. Check	k that your root CA is registered	

4.3 Register device certificate

AWS IoT Core uses client certificates for device authentication. Any device that does not have a valid certificate signed by the registered root CA is denied access and cannot communicate with AWS IoT Core servers. To register client certificates to AWS IoT core, follow these steps:

1. From the AWS IoT Core dashboard, go to **Secure**, go to **Certificates** and click on the **Create a certificate** button as shown in <u>Figure 74</u>:

aws Service	es 🗸 Resource Groups 🗸 🕻
💮 AWS ІОТ	
Monitor Onboard Manage Secure	
Certificates	
Policies	You don't have any certificates yet
CAs	Certificates help things establish a secure connection.
Role Aliases	
Authorizers	Learn more Create a certificate
Defend	
Act	
Test	
Figure 74. Go to t	he AWS IoT Certificates menu

2. A new menu called *Create a certificate* will be opened. Click on the **Get started** option as shown in Figure 75:

aws	Services 🗸 Resource Groups 🖌 🏠
	Create a certificate
	A certificate is used to authenticate your device's connection to AWS IoT.
	One-click certificate creation (recommended) This will generate a certificate, public key, and private key using AWS IoT's certificate authority. Create certificate
	Create with CSR Upload your own certificate signing request (CSR) based on a private key you own.
	Use my certificate Register your CA certificate and use your own certificates for one or many devices. Get started
Figure 7	5. Create a certificate

A new menu called Select a CA will be opened. In this menu, the root CA you
registered in Section 4.2 should appear. Select your root CA certificate and click on
Register certificates button as shown in Figure 75

aws	Services 🗸 Resource Groups 🥆 🏌
	Select a CA
	In order to use your own certificates for one or many devices, you must use a registered CA certificate.
	Registered CAs
	Q Search CA certificate
1	77d6dc3ed9dcbcba22703a1f3f1ca709b6693254137d6edfdd88c75acab42c23 View
	Register CA
	Cancel 2 Register certificates
Figure 7	6. Select a CA to register a client certificate

 Click on the button Select certificates (1), select from your file system the device certificate to be uploaded (2), and click Open button as shown in Figure 77 (3). This device certificate was generated in <u>Section 4.1.3</u>.

	Register existing device certificates								
					_				
	You can register device contificator signed by your CA contificato	Note that you must fi	ret vanistar vour CA sartificato bafar	o unloading de	ico.				
	certificates. You can unload up to 10 device certificates at a time	Open							_
		← → * ↑ ↓ Th	is PC > QS (C:) > se050 middleware > simw-tr	2vn < binaries > nvS	SSCI1 > aws	× či	Search aws		
	Existing certificates	S 7 1 1 1 1	ore - os(e) - scosonada entre - sinni e	op - onnines - pys		+ 0		Per	
	You have not selected any device certificates to upload yet.	Organize • New folde						8:: •	
		MK F&A - Jordi . ^	Name	Date modified	Туре	Size			
1		2 🔁	377824202738246811044179_device_certif	7/17/2019 11:09 AM	Security Certificate	1	KB		
	Select certificates	topos	8 37/824202/382468110441/9_device_key.p 8 377824202738246811044179_device_refer	7/17/2019 11:09 AM	CMS (S/MIME) File	1	KB		
		This PC	interCA_certificate.cer	7/17/2019 11:09 AM	Security Certificate	1	KB		
		3D Objects	6 interCA_key.pem	7/17/2019 11:09 AM	CMS (S/MIME) File	1	KB		
		Desktop	8 prime256v1.pem	7/17/2019 10:52 AM	CMS (S/MIME) File	1	KB		
		Documents	rootCA_certificate.cer	7/17/2019 11:09 AM	Security Certificate	1	KB		
	Cancel	Downloads	rootCA_key.pem	7/17/2019 10:52 AM	CMS (S/MIME) File	1	KB		
	Concer	h Music	V3_ext_device.cnf	7/16/2019 3:05 PM	CNF File	1	KB		
		E Pictures	v3_ext_intermca.cnf	7/16/2019 3:05 PM	CNF File	1	KB		
		Videos							
		🔛 OS (C:)							
		USB Drive (D:)							
		FRDM-K64FD (E:							
		~							
		File n	ame			~ ~	All files (*)		
							-		

5. Select the option *Activate all* (1) and click on the button **Register certificates** (2) as shown in Figure 78.

aws	Services 🗸 Resource Groups 🗸 🛠
	Register existing device certificates
	You can register device certificates signed by your CA certificate. Note that you must first register your CA certificate before uploading device certificates. You can upload up to 10 device certificates at a time. Existing certificates
	Deactivate all Revoke all 377824202738246811044179_device_certificate.cer
	Select certificates
	Cancel 2 Register certificates Done
Figure 78.	Register and activate device certificate

6. Your device certificate is now registered and visible in your AWS IoT Core dashboard as shown in Figure 79.

aws s	ervices 🗸 Resource Groups 🖌 🖌	🗘 jordi.jofre 🕶 Ohio 🕶 Support 👻
🕸 AWS ІоТ	Certificates	Create
Monitor	Search certificates Q	Card 👻
Onboard		
Manage	fc23a14c031fb9e090f	
Secure	ACTIVE	
Certificates		
CAs		
Role Aliases		
Authorizers		
Defend		
Act		
Test		
Figure 79.	Confirm device certificate registration	

4.4 Attach thing and policy to certificate

AWS IoT Core uses client certificates for device authentication. Any device that does not have a valid certificate signed by the registered root CA is denied access and cannot communicate with AWS IoT Core servers. To register client certificates to AWS IoT core, follow these steps:

- 1. Attach a thing to your certificate following the instructions shown in Figure 80.
 - a. Click on the top right corner to go to the device certificate options.
 - b. Click on Attach a thing.
 - c. Select the AWS IoT Thing you created in <u>Section 3.3</u>. In this example, it was called **my_thing**.
 - d. Click on the Attach button.

aws s	ervices 👻 Resource Group	os 🗸 🐐	众 jordi.jofre ∗ Ohio × Support ∗
🖗 AWS IOT	Certificate	S	Create Q.
Monitor	Search certifica	tes	Q Card •
Onboard			
Manage Secure Certificates	fc23a14c03 ACTIVE	Activate Deactivate	a Attach things to certificate(s)
Policies CAs Role Aliases		Revoke Accept transfer Reject transfer	Things will be attached to the following certificate(s): 2afb805e4cf79b9301bc995806eba862c3cddd8546c4cad23f9e181b9153d9d6
Authorizers		Revoke transfer Start transfer	Choose one or more things
Act		Attach policy	Q. Search things
Test	D	Download	C my_thing
		Delete	1 thing select d Attach
Figure 80.	Attach a thin	g to your d	evice certificate

- 2. Attach a policy to your certificate
 - a. Click on the top right corner to go to the device certificate options.
 - b. Click on Attach a policy.
 - c. Select the AWS IoT Policy created in <u>Section 3.4</u>. In this example, it was called **my_policy**.
 - d. Click on the Attach button.

AWS IOT	Certificates	Create 🖓
Monitor	Search certificates	Q Card •
Onboard Manage Secure Certificates	fc23a14c02 ACTIVE Deactivate Deactivate	a Attach policies to certificate(s)
Policies CAs Role Aliases Authorizers	Accept transfer Reject transfer Revoke transfer	Policies will be attached to the following certificate(s): 2afb805e4cf79b9301bc995806eba862c3cddd8546c4cad23f9e181b9153d9d6
Defend	b Start transfer Attach policy Attach thing	Choose one or more policies
Test	Download Delete	C View
		0 policies selected Attach

58 / 67

© NXP B.V. 2021. All rights reserved.

4.5 AWS IoT Core project configuration

To run the AWS project example using the FRDM-K64F board, we need to:

- Download and install the FRDM-K64F SDK.
- Import AWS IoT Core example project
- <u>Configure AWS IoT Core project account settings</u>
- Execute AWS IoT Core example project

Note: Before running the AWS IoT Core demo example, you need to have installed MCUXpresso IDE and FRDM-K64F SDK in your local environment and imported the AWS IoT Core project example. Check <u>AN12396- Quick start guide to Kinetis K64</u> for detailed instructions on:

- How to install MCUXpresso
- How to obtain FRDM-K64F SDK
- How to import FRDM-K64F project examples, including AWS IoT Core project example.

4.5.1 Download and install the FRDM-K64F SDK

The AWS IoT Core device onboarding project example is included as part of the FRDM-K64F SDK. Install it to your MCUXpresso workspace as shown in <u>Figure 82</u>:

- 1. Download the FRDM-K64F SDK, publicly available from the NXP website.
- 2. Drag and drop the FRDM-K64F SDK zip file in the *Installed SDKs* section in the bottom part of the MCUXpresso IDE.
- 3. Check that the FRDM-K64F SDK is installed successfully.

🗙 workspace - Welcome page - MCUXpresso IDE	SDK Import Wizard			
ile Edit Navigate Search Project ConfigTools Run Analysi:				
	8 Please select a board			
Project 😒 🛃 Periphe IIII Registers 🏘 Faults " 🗖	Board and/or Device s	election page		
■ State	✓ SDK MCUs	Available boards		
	MCUS from installed SDKs	Please select an available board for your project.		
	NXP MIMXRT1052xxxxB	Supported boards for device: MIMXRT1052xxxxB		
) ► PC55xx ► MIRASET1052 MIRASET1052xxxx8		SEO 50 (SDK)	⑦ Installed SDKs ⋈ □ Properties № Problems
) Qui ☆ (x)= Vari 음:: Outl ♥₀ Bre (x)= Glo 📟 🗖		-		Installed SDKs
				To install an SDK, simply drag and drop an SDK (zip file/
No project selected				Name
✓ Create or import a project	Selected Device:	S	DKs for selected I	SDK_2.x_EVKB-IMXRT1050
New project	Target Core: cm7 Description: MIMXRT1052: i.M (MCUs) based o	NX® MIMXRT1052 600MHz, \$12KB SRAM Microcontrollers ARM® Cortex®-M7 Core	Name	Image: SDK_2.x_FRDM-K64F 3 Image: SDK_2.x_LPCXpresso55569 3
 Build your project 				
Suild Clean				
×	(?)			

Note: For more detailed instructions on how to install it the FRDM-K64F SDK into our MCUXpresso workspace, refer to <u>AN12396 - Quick start guide with FRDM-K64F</u>.

4.5.2 Import AWS IoT Core example project

The FRDM-K64F SDK includes a project example called se_SE050x_cloud_aws. Import it to your MCUXpresso workspace as shown in Figure 83:

- 1. Click *Import SDK examples* from the MCUXpresso IDE quick start panel.
- 2. Select se_SE050x_cloud_aws project example and click the Finish button.
- 3. Check that the project is now visible in your MCUXpresso workspace

Note: For detailed instructions on how to import project examples from FRDM-K64F SDK, check <u>AN12396 - Quick start guide with Kinetis K64F</u>

🔀 workspace - Welcome page - MCUXpresso IDE	SDK Import Wizard	🔀 workspace - Welcome page - MCUXpresso IDE
ile Edit Navigate Search Project ConfigTools Run Analysi:		File Edit Navigate Search Project ConfigTools Run Anal
" ▪ (\$ \$ • • \$ • ! \$ \$ \$ \$ • ! \$ • ! \$ • ! \$ • ! \$ • ! \$ • ! \$ • ! \$ • ! \$ • ! \$ • ! \$	The source from the SDK will be copied into the workspace. If you want to use linked files, please unzip the 'SDK_2.x_FRDM-K64F' SDK.	Image: A state of the state
Project 🛛 🚼 Periphe IIII Registers 🎋 Faults 🖳 🗖	Import projects	Project 🕺 🛃 Periphe 🔢 Registers 🎄 Faults 🐡 🗖
E 😵 🖽 🖗 📕 🔻 🗸	Project name prefix: frdmk64f	Image: state
	Use default location	> Project Settings >) Includes
	Location: C:\Users\Jordi Jofre\Documents\MCUXpressolDE_11.0.0_2516\workspac	> 29 CMSIS > 29 amazon-freertos
	Project Type	> 😕 board
	C Project C++ Project C Static Library C++ Static Library	> 28 device
		> 29 drivers
		> 😂 lwip
	Examples	> 😂 mbedtls
	type to filter	> 🐸 mmcau
		> 😂 se_hostlib
	Name	> Ca startun
	y ■ Se hostlib examples	> 😝 utilities
	≥ se_SE05x_cloud_aws	> 🗁 doc
	se_SE05x_cloud_azure	
	se_SE05x_cloud_gcp	
	Se SED5x ex ecc	
) Qui 💥 (*)= Vari 🗄 Outl 💁 Bre (*)= Glo 🗖 🗖	se_SE05x_ex_hkdf	() Qui ☆ (×)= Vari 🔄 Outl 💁 Bre (×)= Glo
	se_SE05x_ex_md	
MCIIVasaaa IDE Quiskatat Danal	se_SE05x_ex_rsa	MCIIV process IDE - Quicketart Banal
INCOAPresso IDE - QUICKStart Panel	□ = se_scost_ex_symmetric □ = se_hostlib main471CH	Project: frdmk64f se SE05x cloud aws [Debug]
ino project science	se_hostlib_se05x_ex_i2cMaster	
Create or import a project	□ ≡ se_hostlib_se05x_get_info	 Create or import a project
New project	se_hostlib_se05x_minimal	New project
Market SDK example(s)	se_hostlib_vcomA71CH	🏹 🔣 Import SDK example(s)
Import project(s) from file system		Import project(s) from file system
 Build your project 		▼ Build your project
😥 🐔 Build		Sea 🔨 Build
Clean		Clean
·	(?)	Generation of the set of the
	L	

4.5.3 Configure AWS IoT Core project account settings

We need to change the AWS Rest API Endpoint in the MCUXpresso demo project with the one in your AWS IoT Core account settings. Follow these steps:

1. From the AWS IoT Core dashboard, go to *Manage*, then go to *Things* and click on your AWS IoT Thing as shown in Figure 84:

AWS ΙΟΤ	Things	
		Create
Monitor Onboard Manage Things	Search things Q Configure fleet indexing ()	Card 💌 🛞
Types Thing Groups Billing Groups Jobs		
Secure Defend		
Act Test		

2. On the left hand side menu, (1) go to **Interact**. Inside this menu, you will find your (2) Rest API Endpoint as indicated in <u>Figure 85</u>. Copy this URL.

aws Services	✓ Resource Groups ✓ ★	Support 👻
← Things > my_t	thing	Q
THING		(?) (A)
MO TYPE	ning Actions •	0
Details	This thing already appears to be connected.	
Security	нтрс	
Thing Grou Billing Grou	ps	
Shadow	an9ryhd19gvfr-ats.iot.us-east-2.amazonaws.com	2
Activity	ΜΩΤΤ	
Jobs Violations	Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow)	
Defender m	Learn more netrics Update to this thing shadow	
_	<pre>\$aws/things/my_thing/shadow/update</pre>	
	Update to this thing shadow was accepted \$aws/things/my_thing/shadow/update/accepted	
Figure 85 Fi	ind your Rest API Endpoint	

3. Go to the AWS demo in your MCUXpresso workspace. Navigate to the aws_clientcredential.h file located in frdmk64f_se_SE05x_cloud_aws \source folder. Replace the clientcredentialMQTT BROKER ENDPOINT

AN12404	
Application note	

variable with the Rest API Endpoint of your AWS account obtained in the previous step, as well as you thing name as created in <u>Section 3.3</u>. Check <u>Figure 86</u> for reference.

workspace - frdmk64f_se_SE05x_cloud_aws/source/aws_clientcred	ential.h - MCUXpresso IDE	
Edit Source Refactor Navigate Search Project Confi	Tools Run Analysis FreeRTOS Window Help	
- 🕅 🐘 🎘 - 🔍 - 🐘 🥑 🕓 🕋 - 🖻 🔪 🍺	■ 2、3、2 二) -
	Quick Access	EX 🗙 3
Project 😢 🐕 Periphe 🚟 Registers 🂠 Faults 👘 🗖	Welcome aws_clientcredential.h 23	- t
- 🔶 m & M - 🗸	49 #endif	~
	50⊖ /*	
Project Settings	51 * MQTT Broker endpoint.	
) includes 2	static const char clientcredentialMOTT_REGKER_ENDPOINT[] + "ap@mubdl@mufr_atr_ict ur_mast_2 amazonawr_com";	
> CMSIS	54	
> 📴 amazon-freertos	550 /" Use of a "define" and not a "static const" here to be able to	
> 😂 board 🛛 🛃 🤰	55 "use pre-comple concatenation on the string. "/	
> 😂 component	s whethe cherchedencially_indiv_inv_ my_chang	
> 😕 device	590 /*	
> 😂 drivers	60 * Port number the MQTT broker is using.	
> 🙆 libs	61 */	
> 😕 hwip		
> 🙆 mbedtls	640 /*	
> 📴 mmcau	65 * Port number the Green Grass Discovery use for JSON retrieval from cloud is using.	
> 😂 se_hostlib	67 #define clientcredentialGREENGRASS DISCOVERY PORT 8443	
V B source	68	
> m app.n	690 /*	
> B aws_clientcredential keys h	/0 Will network to join.	
h avs clientcredential.h	72 #define clientcredentialWIFI SSID "Paste WiFi SSID here."	
> h aws_demo.h	73	~
N aws int config h	<	>
Qui, S2 (V> Vari 🗄 Outl • Bre (V> Glo • 🗖	🍘 Installed SDG: 🦷 Properties 🗽 Problems 🕒 Console 😒 🦉 Terminal 🚔 Image Info 👔 Memory: VM-Heap and Stack Utage 🖳 Debugger Console 🗹 🗎 🔻 No consoles to display at this time.	
Project: frdmk64f_se_SE05x_cloud_aws [Debug]		
create or import a project		
New project		
Import SDK example(s)		
Import project(s) from file system		
Build your project		
C & Ruild		
Clean V		
/frdmk64f se SE05x cloud aws/source/aws clientcredential.h	RD IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	ok d aws)

Figure 86. Set the Rest API Endpoint and thing name in your project settings

4. On the same **Interact** menu, you will find MQTT topics that enable applications and things to get, update, or delete the state information for an AWS thing. For instance, copy the MQTT **update** topic as shown in <u>Figure 87</u>:

aws	Services - Res	source Groups 🐱	*			4	jordi.jofre 👻	Ohio 🗸	Support 👻
← Thi	ings > my_thing								Q
	THING my_thing								? I
	NO TYPE						Acti	ons 👻	
	Details	This thing already	appears to be connecte	d.			Connect a d	evice	
	Security Thing Groups	HTTPS							
	Billing Groups	Update your Thin	g Shadow using this Rest	API Endpoint. Learn more					
	Shadow	an9ryhd19gv	fr-ats.iot.us-east	-2.amazonaws.com					
	Activity	MQTT							
	Jobs Violations	Use topics to enal	ble applications and thin	gs to get, update, or delete the sta	te information for	a Thi	ng (Thing Shade	ow)	
	Defender metrics	Update to this th	ing shadow						
		\$aws/things	/my_thing/shadow/u	pdate					
		Update to this th	ing shadow was accepte	d				_	
		\$aws/things	/my_thing/shadow/u	pdate/accepted					
Figure 87	7. Find y	our MQT	T Update	topic					

5. Go to the AWS demo in your MCUXpresso workspace. Navigate to the aws_jitr_task_lwip.c file located in frdmk64f_se_SE05x_cloud_aws
\source folder. Replace the #define PUB_TOPIC variable with the MQTT topic
you obtained in Figure 86 as shown in Figure 87.



AN12404

EdgeLock[™] SE05x for secure connection to AWS loT Core

5 Legal information

5.1 Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability - Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors. In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory. Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification. Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products. NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or

the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Evaluation products — This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer. In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages. Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US\$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

Translations — A non-English (translated) version of a document is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security - Customer understands that all NXP products may be subject to unidentified or documented vulnerabilities. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately. Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP. NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

5.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

AN12404

EdgeLock[™] SE05x for secure connection to AWS IoT Core

Tables

Tab. 1. OM-SE050ARD development kit details	vl-K64F details5
---	------------------

Rev. 1.6 — 15 January 2021 535216

Figures

Fig. 1. Fig. 2. Fig. 3.	AWS IoT Core device registration flow4 Get started with AWS IoT Core for free6 Sign in or create a new AWS IoT Core
F : 4	
⊢ıg. 4.	Create an AVVS IoT Core account
Fig. 5.	Create an AWS IoT Core account - Contact
	information
Fig. 6.	Create an AWS IoT Core account -
	Payment information9
Fig. 7.	Create an AWS IoT Core account - Confirm your identity
Fig. 8.	Create an AWS IoT Core account - Enter
0	verification code11
Fia. 9.	Create an AWS IoT Core account - Select
0	a support plan12
Fig. 10.	AWS Management Console - IoT landing
ge.	page 13
Fig. 11.	AWS Management Console - AWS IoT
	Core
Fig. 12.	Go to the AWS Io1 Things menu
Fig. 13.	Select Create a single thing option 14
Fig. 14.	Add your device to the thing registry15
Fig. 15.	Create a thing without certificate16
Fig. 16.	Confirm AWS IoT Thing creation16
Fig. 17.	Go to the AWS IoT policies menu17
Fig. 18.	Create a policy name and go to Advanced mode 17
Fia. 19.	Personalize your AWS IoT Core policy
Fig. 20.	Confirm AWS IoT policy creation 19
Fig. 21.	Create se050 middleware folder 20
Fig. 22.	Unzip se050 middleware 20
Fig. 23	Unplug and plug OpenSDA port 21
Fig. 24	FRDM-K64F drive 21
Fig. 25	VCOM binary folder 22
Fig. 26	Drag and drop VCOM binary 22
Fig. 20.	Check VCOM and serial ports 23
Fig. 28	Connect to the Edgel ock SE05x using
1 19. 20.	ssecti and read certificate ID list 24
Fig 29	Get the certificate from the Edgel ock
i ig. 23.	SE05y using esseli
Fig 30	Go to the AW/S IoT Certificates menu 25
Fig. 31	Pogistor a cortificato
Fig. 31.	Register a certificate
Fig. 32.	Degister evicting device cortificate
Fig. 33.	Activate device certificate
Fig. 34.	Activate device certificate
Fig. 35.	Attach a maliau to your device certificate
FIY. 30.	Attach a policy to your device certificate
гig. 37.	IIIIpuit FRUIVI-R04F SUK
гig. 38. Гіт. 20	Import Avvo project in the workspace
FIG. 39.	Go to your AvvS to Lining
FIG. 40.	Cattle Dest AD Endpoint
⊢lg. 41.	Set the Rest API Endpoint and thing name
-	in your project settings
⊢ıg. 42.	Find your MQ11 Update topic

Fig 13	Set the MOTT publish tonic	34
Fig. 40.	Setting the credential IDs	34
Fig. 44.	Subscribe to the MOTT topic	30 36
Fig. 45.	Check MOTT topic subscription	30
Fig. 40.	Connect EPDM K64E board	30
Fly. 47.	Configure TereTerm	37 27
Fig. 40.		،د مد
Fig. 49.		20
Fig. 50.	Device connection to AWS - dashboard	
Fig. 57.	Device connection to AWS - Dublished	
i ig. 52.	messages in the undate MOTT tonic	40
Eig 52	Create so050 middleware folder	40 12
Fig. 53.	Linzin co050 middlowaro	_4 12
Fig. 54.	Unplug and plug OpenSDA port	_42 13
Fig. 55.	ERDM_K64E drive	4 0 13
Fig. 50.	VCOM bipany folder	43
Fig. 57.	Drag and drop VCOM binary	44 11
Fig. 50.	Check VCOM and aprial parts	44
Fig. 59.	Connect boards	40
Fig. 60.	Eind Provision AWS ave file in your	40
FIQ. 01.	Edgel ook SE05x Dlug & Trust Middlewere	
	backage	46
Fia. 62.	Run Provision AWS.exe executable	
Fig. 63.	Generated AWS credentials	
Fig. 64.	AWS IoT certificates menu	49
Fig. 65.	AWS certificate creation options	50
Fig. 66.	Register a CA certificate	50
Fig. 67.	Get registration code	51
Fig. 68.	Locate verification certificate.py Python	
0	script	51
Fig. 69.	Go to Provisioning folder	52
Fig. 70.	Execute verification certificate.py script	52
Fig. 71.	Verify generation of the AWS verification	
0	certificate	52
Fig. 72.	Upload your root CA and AWS verification	
•	certificate	53
Fig. 73.	Check that your root CA is registered	54
Fig. 74.	Go to the AWS IoT Certificates menu	55
Fig. 75.	Create a certificate	55
Fig. 76.	Select a CA to register a client certificate	56
Fig. 77.	Select a client certificate	56
Fig. 78.	Register and activate device certificate	57
Fig. 79.	Confirm device certificate registration	57
Fig. 80.	Attach a thing to your device certificate	58
Fig. 81.	Attach a policy to your device certificate	58
Fig. 82.	Import FRDM-K64F SDK	59
Fig. 83.	Import AWS project in the workspace	60
Fig. 84.	Go to your AWS IoT Thing	61
Fig. 85.	Find your Rest API Endpoint	61
Fig. 86.	Set the Rest API Endpoint and thing name	
	in your project settings	62
Fig. 87.	Find your MQTT Update topic	62
Fig. 88.	Set the MQTT publish topic	63

Contents

1	EdgeLock SE05x ease of use	
	configuration	3
2	Leveraging EdgeLock SE05x for AWS IoT	
•	Core device onboarding	4
3	Running AWS IOT Core device onboarding	_
0 4	project example	5
3.1	Hardware required	5
3.2	Sign up for an AWS Io1 Core account	5
3.3	Create an AWS IoT thing	13
3.4	Create a policy	16
3.5	Extracting credentials from EdgeLock SE05x	19
3.5.1	Download EdgeLock SE05x Plug & Trust	
	Middleware	.19
3.5.2	Flash FRDM-K64F with VCOM software	21
3.5.3	Read device certificate from EdgeLock SE05x	23
3.6	Registering device certificate in AWS IoT	24
3.7	Attach AWS Thing and policy to the	. 27
~ ~		.27
3.8	AWS IOT Core project configuration	29
3.8.1	Download and install the FRDM-K64F SDK	29
3.8.2	Import AWS Io I Core example project	30
3.8.3	Configure AWS IoT Core project account	~ .
~ ~	settings	.31
3.9	AWS Io1 Core project execution	35
4	Appendix: Registering a CA certificate for	
	just-in-time registration	.41
4.1	Running AWS IoT Core key provisioning	
	scripts	.41
4.1.1	Download EdgeLock SE05x Plug & Trust Middleware	.41
4.1.2	Flash FRDM-K64F with VCOM software	43
4.1.3	Key and certificate configuration for use	45
4.0		.45
4.2	Register root certificate authority (CA)	.48
4.2.1	Get registration code from AWS	.48
4.2.2	Generate AWS verification certificate	51
4.2.3	Upload root CA and AWS verification certificate	.52
4.3	Register device certificate	54
4.4	Attach thing and policy to certificate	. 57
4.5	AWS IoT Core project configuration	59
4.5.1	Download and install the FRDM-K64F SDK	59
4.5.2	Import AWS IoT Core example project	60
4.5.3	Configure AWS IoT Core project account	
	settings	.60
5	Legal information	. 64

Please be aware that important notices concerning this document and the product(s) described herein, have been included in section 'Legal information'.

© NXP B.V. 2021.

All rights reserved.

For more information, please visit: http://www.nxp.com For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 15 January 2021 Document identifier: AN12404 Document number: 535216