

Connecting the NuMaker-IoT-M487 to AWS



Introduction #

In this tutorial we are going to connect the Nuvoton NuMaker-IoT-M487 to Amazon Web Service (AWS) IoT and run a simple "Hello World" demo.



For this tutorial you will need:

- Nuvoton NuMaker-IoT M487 Contact us for more info
- MicroUSB Cable
- Windows Computer
 - Keil Essential or higher or Keil Nuovoton Edition installed
 - Nuvoton Nu-Link Keil Driver installed
 - Nuvoton M480 device packs installed using the pack installer
 - Nuvoton Nu-Link Keil driver installer should prompt you to install these
- Wifi connection to the internet
- · AWS account and an IAM user with permission to access AWS IoT and FreeRTOS cloud services



Download Amazon FreeRTOS #

Let's start by getting the Amazon FreeRTOS source code. This can be downloaded from github or by cloning the git using the following command: git clone https://github.com/aws/amazon-freertos.git -b release --recurse-submodules

Note: this will clone the release branch along with the required submodule. The release branch is the tested release code, whereas the master branch is the development code.

Once done, we need to open our example project. In Keil, click *File* > *Open*, and in the bottom right corner of the Open window, ensure that *Project Files* is selected:

~	Source (*.c; *.cpp; *.h; *.a*; *.inc; *.src; *.s)
	Source (*.c; *.cpp; *.h; *.a*; *.inc; *.src; *.s) Text (*.txt) Listing (*.lst: *.m*: *.cod)
	C Source (*.c) C Header (*.h)
	Asm Source (*.cpp) Asm Source (*.a*; *.s) Asm Header (*.inc)
	Project Files (*.uvproj;*.uvprojx) Multi Project Workspaces (*.uvmpw; *.mpw) All Files (*.*)

Navigate to where you downloaded the Amazon FreeRTOS repository and open the following file:

amazon-freertosprojectsnuvotonnumaker_iot_m487_wifiuvisionaws_demosaws_demos.uvproj

Once opened we need to verify that Keil is setup correctly for our Nuvoton device. Right click on the project name in the project panel and selection *Options*:



Click on the Utilities tab and verify that Use Target Driver for Flash Programming is selected and that NULink Debugger is set as the target driver:

Options for Target 'aws_demos'
Device Target Output Listing User C/C++ Asm Linker Debug Utilities
Configure Flash Menu Command
NULink Debugger Settings
Init File: Edit
C Use External Tool for Rash Programming Command: Arguments: Run Independent
Configure Image File Processing (FCARM):
Output File: Add Output File to Group:
freertos_kernel/
Image Files Root Folder:
OK Cancel Defaults Help

Next click on the Debug tab and firstly ensure that NULink Debugger is selected at the top right:



Options for Target 'aws_demos'	×
Device Target Output Listing User C/C++ Asm	Linker Debug Utilities
C Use Simulator with restrictions Settings Using Limit Speed to Real-Time	Use: NULink Debugger Settings
Load Application at Startup Run to main() Initialization File: Edit	Load Application at Startup Run to main() Initialization File: Edit.
Restore Debug Session Settings Breakpoints I Toolbox Vatch Windows & Performance Analyzer Memory Display V System Viewer CPU DLL: Parameter:	Restore Debug Session Settings Image: Session Settings
SARMCM3.DLL Parameter: DARMCM1.DLL DARMCM1.DLL Manage Component Virginia Sector Secto	SARMCM3.DLL Dialog DLL: Parameter: TARMCM1.DLL Warn if outdated Executable is loaded swer Description Files
OK Car	ncel Defaults Help

Click the Settings button next to this drop-down box and in the window that appears verify that the Chip Type is set to M480:

New Hirely Defense Cate			~
Nu-Link Driver Set	q		^
Debug Trace			
Nu-Link		Chip Select	Supporting Forum
Driver Version:	7130r	Chip Type: M480 💌	EN: http://forum.nuvoton.com/
ICE Version:		Reset Options	SC: http://www.nuvoton-mcu.com/
Device Family:	Cortex-M	Connect: Normal 💌	
Device ID:		Reset: Autodetect 💌	
Port:	SW 💌	Download Options	
Max Clock:	1MHz 💌	Verify Memory Code	
Power Control			
10 Voltage	C 1.8v C	2.5v © 3.3v © 5v	
]
			OK Cancel

Click OK on both of these windows and back in the project panel, right click on the AWS demos project and select build target:



If all went well, we should be able to build successfully:

d Output
mpiling jsmn.c
mpiling cborparser_dup_string.c
mpiling cborparser.c
iking
ugram Size: Code=228168 RO-data=82876 RW-data=780 ZI-data=155684
<pre>mELF: creating hex file</pre>
bbj\aws_demos.hex: Warning: Q9931W: Your license for feature Keil will expire in 29 days
hished: 0 information, 1 warning and 0 error messages.
er Build - User command #1: fromelfbin "./obj/aws_demos.axf"output "./obj/aws_demos.bin"
cning: Q9931W: Your license for feature Keil will expire in 29 days
hished: 0 information, 1 warning and 0 error messages.
er Build - User command #2: fromelftext -c ".\obj\aws_demos.axf"output ".\obj\aws_demos.txt"
ming: Q9931W: Your license for feature Keil will expire in 29 days
iished: 0 information, 1 warning and 0 error messages.
<pre>\obj\aws_demos.axf" = 0 Error(s), 21 Warning(s).</pre>
ild Time Elapsed: 00:00:22

Register NuMaker-IoT Board with AWS IoT #

We now need to configure the demo code to correctly connect to our AWS cloud but before we can do that we must first register it with AWS IoT. For this to work we need the following:



- AWS IoT Policy
 - This grants your device permission to access AWS IoT resources.
- AWS IoT Thing
 - This is the configuration of your device that's shown in AWS IoT, every device you have connected to AWS IoT must have an associated thing.
- Private & Public Key Pair
 - These keys are used to authenticate your device with AWS IoT.

Create an AWS IoT Policy

Let's start by creating the policy that will decide what permissions your devices will have when accessing AWS IoT. We first need to get our AWS account number and AWS region.

In the AWS Management Console, select My Account from the account drop down menu in the top right corner of the page:



From there we can get our Account ID:

Account Settings

Account Id:	123456789012
Seller:	AWS EMEA SARL
Account Name:	Ineltek UK
Password:	****

Next go to the AWS IoT Console by typing IoT Core into the search bar at the top and selecting IoT Core:

	Q iot core	×
	Search results for 'iot core'	
Services (18)	Services	See all 18 results ►
Features (13) Documentation (181,664) Marketplace (28)	Connect Devices to the Cloud	
	😚 IoT Core	

In the top right hand corner of the page we can see the region we are currently using:





In my case I am connected to Europe (Ireland), but we need to note down the correct endpoint name: eu-west-1.

From the navigation pane on the left, click Secure > Policies and then click the Create button from the top right. Enter a name for your policy and in the *Add statements* section, click *advanced mode*.

Copy and paste the following JSON and replace aws-region and aws-account with your AWS Region and account ID:

```
{
      "Version": "2012-10-17",
"Statement": [
      {
            "Effect": "Allow",
"Action": "iot:Connect"
            "Resource":"arn:aws:iot:aws-region:aws-account-id:*"
      },
      {
            "Effect": "Allow",
"Action": "iot:Publish",
"Resource": "arn:aws:iot:aws-region:aws-account-id:*"
     },
{
             "Effect": "Allow",
"Action": "iot:Subscribe",
"Resource": "arn:aws:iot:aws-region:aws-account-id:*"
      },
{
             "Effect": "Allow",
"Action": "iot:Receive",
                   "Resource": "arn:aws:iot:aws-region:aws-account-id:*"
            }
            1
      }
```



For example: "Resource": "arn:aws:iot:eu-west-1:123456789012:*"

This policy grants the following permissions:

- iot:Connect
 - Grants your device the permission to connect to the AWS IoT message broker with any client ID.
- iot:Publish

• Grants your device the permission to publish an MQTT message on any MQTT topic.

- iot:Subscribe
 - Grants your device the permission to subscribe to any MQTT topic filter.
- iot:Receive
 - Grants your device the permission to receive messages from the AWS IoT message broker on any MQTT topic.

Once complete, click the Create button:

AWS IOT \times	AWS IoT > Policies > Create a policy	
lonitor	Constant and Harr	
tivity	Create a policy	
nboard		
anage	Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics more about IoT policies go to the AWS IoT Policies documentation page.	i, topic filters). To learn
reengrass	Name	
ireless connectivity	NuMaker-IoT	
ecure		
rtificates	Add statements	
licies	Policy statements define the types of actions that can be performed by a resource.	Basic mod
5		
e Aliases	1 (g 2 = "Varsion": "2012.10.17".	
thorizers	3 "Statement": [
fend	<pre>5 "Effect: 'Allow", 6 "Action": 'Iot:Connect", 7 "Resource": 'Arn:Bws:Iot:eu-west-1:123456789012:**</pre>	
t	8 }, 9 { 10 *#####*******************************	
st	11 "Action": "foir/bubish", 12 "#esource": "arriawsidd:eu-west-1:223456789012:*" 13 },	
iftware	14 15 "Effect": "Allow", 16 "Action": "LotsUsscribe", 17 "Resource": "arminesioficteu-west-1:123456789012:**	
tings	18 }) 19 { 10 }	
m	21 "Action": "iot:Receive", 22 "Personnews:iot:eve-west-1:123456789012:*"	
cumentation 🖸		
) New console experience Tell us what you think	Add statement	
		Create

Create an AWS IoT Thing

We now need to create a thing in the IoT Console that will allow our device to connect. In the navigation pane on the left select Manage > Things. If you don't have any existing devices, select *Register a thing*, otherwise click the *Create* button. Next select *Create a single thing* and then on the next page enter a name for your device and click Next:

itor		
	CREATE & THING	
ity	Add your device to the thing registr	v
oard		, ,
age	This step creater as approvide the thing register and a thing of	nation for non- device
gs -	This step creates an end y in the thing registry and a thing si	addw for your device.
	Name	
groups	NuMaker-IoT-1	
g groups		
	Apply a type to this thing	
ls	Apply a type to this thing	an consistant conistant data for things that share a tune. Tunes provide things will
Darate	common set of attributes, which describe the identity and ca	ing consistent registry data for things that share a type. Types provide things wi pabilities of your device, and a description.
ingrass	Thing Type	
less connectivity	No type selected	Create a type
re		
nd		
nd	Add this thing to a group	
nd	Add this thing to a group Adding your thing to a group allows you to manage devices	remotely using jobs.
nd	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group	remotely using jobs.
nd	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Groups /	remotely using jobs. Create group Char
vare	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Groups /	remotely using jobs. Create group Char
rare	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Groups /	remotely using jobs. Create group: Char
vare	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Groups / Set searchable thing attributes (ontional)	remotely using jobs. Create group Char
are 195 mentation 🖸	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Groups / Set searchable thing attributes (optional) Entry a value for one or more of these attributes so that you	remotely using jobs. Create group Char can search for your things in the resistary.
are 195 mentation [2]	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Group / Set searchable thing attributes (optional) Enter a value for one or more of these attributes so that you Artichate the	remotely using jobs. Create group Char can search for your things in the registry. Value
d are 55 sector C	Add this thing to a group Adding your thing to a group allows you to manage devices Thing Group Groups / Set searchable thing attributes (optional) Enter a value for one or more of these attributes so that you Attribute key	remotely using jobs. Create group Char can search for your things in the registry. Value
are gs nentation [2] Were console experience fill or what you think	Adding your thing to a group allows you to manage devices. Thing Group Groups / Sets searchable thing attributes (optional) Enter a value for one or more of these attributes so that you Attribute key Provide an attribute key, e.g. Manufacturer	remotely using jubs. Create group Char can search for your things in the registry. Valar Previde an attribute value, e.g. Acres Corporation.
are gs mentation [2] New conside experience [fit is what you think	Add this thing to a group allows you to manage devices Thing Group Groups / Set searchable thing attributes (optional) Enter a value for one or more of these attributes so that you Attribute key Provide an attribute key e.g. Manufacturer	remotely using jobs. Create group Char can search for your things in the registry. Value Previde an attribute value, e.g. Acme Corporation
nd vare rgs mentation [2] New conside experience Triel as what yes think	Adding your thing to a group allows you to manage devices Thing Group Groups / Set searchable thing attributes (optional) Enter a value for one or more of these attributes so that you Attribute key Prodid an attribute key e.g. Manafacturer Add searcher	remotely using jobs. Create group Class can search for your things in the registry. Value Provide on statistude value, e.g. Acres Corporation C

On the next page we need to choose the certificate we'll be using to authenticate this *thing*. For this tutorial, we are going to ask AWS to create one for us. Click the *Create certificate* button next to *One-click certificate creation (recommended)*:





On the next page, download the certificate and private key, in my case this is 7602ee04f8.cert.pem and 7602ee04f8.private.key:

Certificate crea	ted!		
Download these files and after you close this page.	save them in a safe place. Certificat	es can be retrieved at a	any time, but the private and public keys cannot be retrieved
In order to connect a dev	ice, you need to download the foll	owing:	
A certificate for this thing	7602ee04f8.cert.pem	Download	
A public key	7602ee04f8.public.key	Download	
A private key	7602ee04f8.private.key	Download	
You also need to downlo: A root CA for AWS IoTDow Activate	ad a root CA for AWS loT: vnload		
Cancel			Done Attach a policy

Once done click the *Attach a policy* button. Next we need to attach a policy to our certificate that grants our device access to AWS IoT operations. Select the policy we just created and click the *Register thing* button:

CREATE A THING Add a policy for your thing	STEP 3/3
Select a policy to attach to this certificate:	
☑ NuMaker-IoT	View
1 policy selected	Register Thing

Finally, we need to activate the certificate that was just created. In the navigation pane on the left select Secure > Certificates and next to the certificate that was just created, click on the 3 dots menu and select Activate:

VS IoT > Certificates		
ertificates		Create
Search certificates Q		
Name	Status	
a9f572f6a2e54262d535ce7466d0f08beb9c29cc093edc69df3d5f5de3f3f3ef	Inactive	
		Activate
		Deactivate
	1	Revoke

We can now move on to configure our demo code.

Configure Demo Code #

In order for our demo code to connect to AWS IoT and be correctly recongised and authenticated, we need the following information:

- Thing name
- AWS IoT endpoint

Our Thing name is what we called our thing in the previous step, in my case, I called it NuMaker-IoT-1:



	AWS IoT	×	AWS IoT > Things	
	Monitor Activity		Things	
Þ	Onboard		Search things Q Fleet Indexing Info	
Ŧ	Manage		Name	
	Things			
	Types		NuMaker-IoT-1	
	Thing groups			
	Billing groups			
	Jobs			
	There is a			

If I were to add multiple NuMaker-IoT boards to AWS, I could call each one NuMaker-IoT-2, NuMaker-IoT-3, and so on.

Our AWS IoT endpoint is the server address that our device will connect to, and we can find this by click on settings in the navigation pane on the left:

Act
Test
Software
Settings
Learn
Documentation 🛽
New console experience Tell us what you think

Once there, our AWS IoT endpoint should be at the top:

AWS IoT $\qquad \times$	AWS IOT > Settings	
Monitor	Settings	
Onboard	Custom endpoint ENALED	
 Manage 	This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to prost when using an MOTT client or the AWE IoT Device SDV.	
Greengrass	Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.	
Wireless connectivity	Endpoint	
Secure	albc2d34ef678g-ats.iot.eu-west-1.amazonaws.com	
 Defend 		
> Act	DISABLED	
Test	Logs You can enable AWS IoT to log helpful information to CloudWatch Logs. As messages from your devices pass through the message	

Let's make a note of both of these and head back into Keil. We only need to configure a single file for our demo to work:

/demos/include/aws_clientcredential.h

We need to specify our IoT endpoint and thing name:

29 [
30	* @brief MQTT Broker endpoint.
31	*
32	* @todo Set this to the fully-qualified DNS name of your MQTT broker.
33	- */
34	<pre>#define clientcredentialMQTT_BROKER_ENDPOINT "albc2d34ef678g-ats.iot.eu-west-l.amazonaws.com"</pre>
35	
36 [白/*
37	* @brief Host name.
38	*
39	* @todo Set this to the unique name of your IoT Thing.
40	* Please note that for convenience of demonstration only we
41	* are using a #define here. In production scenarios the thing
42	* name can be something unique to the device that can be read
43	* by software, such as a production serial number, rather
44	* than a hard coded constant.
45	- */
46	<pre>#define clientcredentialIOT_THING_NAME "NuMaker-IoT-1"</pre>
47	

And we need to specify our WiFi network name, passkey and security type:



58	· 白/*
59	* @brief Wi-Fi network to join.
60	1 ×
61	* @todo If you are using Wi-Fi, set this to your network name.
62	1 - */
63	#define clientcredentialWIFI_SSID "WiFi Name"
64	
65	i @/*
66	* @brief Password needed to join Wi-Fi network.
67	* @todo If you are using WPA, set this to your network password.
68	1 - */
69	#define clientcredentialWIFI_PASSWORD "WiFi Password"
70	
71	- P/*
72	* @brief Wi-Fi network security type.
73	i *
74	* @see WIFISecurity_t.
75	*
76	* @note Possible values are eWiFiSecurityOpen, eWiFiSecurityWEP, eWiFiSecurityWPA,
77	* eWiFiSecurityWFA2 (depending on the support of your device Wi-Fi radio).
78	- */
.79	#define clientcredentialWIFI_SECURITY eWIFiSecurityWPA2
80	
81	Fendir /* linderAWS_CLIENICREDENIIAL_H */
82	ha

Adding Security Certificate to Demo Code

The final step before we can connect is to add our security certificate that we donwloaded previously. In the Amazon FreeRTOS source code, open up:

tools/certificate_configuration/CertificateConfigurator.html

And select your certificate PEM file and private key PEM file that we downloaded earlier:

FreeRTOS Developer Demos Certific X +					
\leftrightarrow \rightarrow C \textcircled{a} \textcircled{b} file:///C:/amazon-freertos/tools/certificate_conf	iguration/CertificateConfigurator.html 🛛 🐨 🖂 🕇	<u>≯</u>			
Certificate Configuration Tool FreeRTOS Developer Demos					
Provide client certificate and private key PEM file	s downloaded from the AWS IoT Console.				
Certificate PEM file: Browse 7602ee04f8-certificate.pem.crt.txt					
Private Key PEM file: Browse 7602ee04f8-private.pem.key					
③ Generate and save aws_clientcredential_keys.h					
A Save the generated header file to the <i>demos/include</i> folder o Copyright (C) 20	f the demo project. 17 Amazon.com, inc. or its affiliates. All Rights Reserved.				

Once done click Generate and save aws_clientcredential_keys.h:

-						
	Opening aws_client	credential_keys.h	×			
r	You have chosen t	o open:				
	🧧 aws_clientcr	edential_keys.h				
I	which is: Tex	t File (3.9 kB)				
I	from: blob:					
l						
	What should Firefox do with this file?					
	Open with	Applications\sublime_text.exe (default) ~				
n	◯ <u>S</u> ave File					
	Do this <u>a</u> uto	matically for files like this from now on.				
7						
		OK Cancel				

Save this file to the demos/include directory and replace the file if it already exists. Head back into Keil and build our code again:



Project				Ļ
🖃 🎕 Projec	t: aws_d	emos		
🗄 😓 aw	/s demo	r		1
÷	fre 🔊	Options for Target 'aws_demos'	Alt+F7	
🛨 🧰	fre	Add Group		
÷	VE 🔔	Manage Project Items		
	fr 📕			
	fr	Open Map File		
		Open Build Log		a (analia
	Ve (##)	Rebuild all target files		os/applica
	de	neo una un target mes		
÷	de 🛄	Build Target	F7	
.	de 🗸	Show Include File Dependencies		1
÷	librarre	s/c_sak/standard/common/		1
· .	librarie	s/c_sdk/standard/common/include/		
<u> </u>	librarie	s/c_sdk/standard/common/include/types/		

Testing

Once done, we can then load our code into the NuMaker-IoT-M487. Start by plugging a microUSB cable into the on-board NuLink port:



Then in Keil, select Download from the Flash menu:

rojects\nuvoton\numaker_iot_m487_wifi\u

ct	Flas	h	Debug	Peripherals	Тос	ols
	LOAD	D	ownload		F8	18
5		Er	ase			
		c	onfigure F	lash Tools		Ľ
IS	_					

This should download our built code to the NuMaker-IoT-M487.

The on-board NuLink will enumerate as a serial port when we plug it into our computer:





This means we should be able to open a serial terminal to see what's happening on the device (baud rate is 115200):



Let's go back to AWS IoT and from the left navigation panel, select *Test*. We then can then subscribe to an MQTT topic and display the output. Type *ThingName*/example/topic (but replace *ThingName* with your thing's name), or type # to subscript to all topics:

AWS INT > Test MQTT client Info	c	onnected as iotconsole-1234567890123-1
Subscriptions		
Subscribe to a topic Publish to a topic	Solucifie Device publick MQTT messages on topics. You can use this client to subsorbe to a topic and readve these messages. Solucifying topic: Notate left 'I clearing/ortpgic Ara message capture: Mo 100 B	Subscribe to topic

Hit the RESET button on the NuMaker-IoT-M487 and we should see our incoming messages:

Subscriptions	NuMaker-IoT-1/example/topic	Export Clear Pause
Subscribe to a topic Publish to a topic NuMaker-IoT-1/example/to ×	Posiblish Specify a topic and a message to positish with a QoS of Q. Nuklisher-IsT-Tommpfor/topic	Publish to topic
	a g mesage's "wells from Ad lot conside" g g	
	NuMaker-IoT-1/example/topic February 09, 2021, 14:56:20 (UTC+0000)	Export Hide
	We cannot display the message as JSON, and are instead displaying it as UTF-8 String. Relie World!	
	NuMaker-IoT-1/example/topic February 09, 2021, 14:56:17 (UTC+0000)	Export Hide
	We cannot display the message as JSON, and are instead displaying it as UTF-8 String. Ecile World!	



We can see our device sending these messages in the serial terminal:



We can also click on the *Publish to topic* if we didn't subscribe to a generic topic (#) and see our MCU acknowledge an incoming message in the serial terminal:

[iot_thread] Incoming Publish Topic Name: NuMaker-IoT-1/example/topic matches subscribed topic.Incoming Publish Mess

Conclusion #

Obviously this is just demo code showing our device sending the same string of data to our AWS IoT service over and over, but I hope this gave you a good look at how to get started with the Nuvoton IoT devices.