



User Guide:
Azure IoT Hub Device
Update for the SAM E54
Xplained Pro Development
Board

Table of Contents

<i>Overview</i>	<i>3</i>
<i>Prepare Azure Resources.....</i>	<i>5</i>
Create an IoT Hub.....	5
Register a Device with the IoT Hub	7
Create Device Update Account and Instance	8
<i>Prepare the New Firmware</i>	<i>9</i>
Clone/Download the Example Project.....	9
Build New Firmware Image.....	9
Prepare New Firmware for the Simulated Leaf Device.....	11
Generate Import Manifest	11
Upload Firmware and Manifest.....	12
<i>Prepare the Device</i>	<i>14</i>
Revert to the Previous Firmware Version.....	15
Program and Run the Device.....	15
<i>Deploy New Firmware.....</i>	<i>17</i>
Add a Tag to Your Device.....	17
Deploy New Firmware Update	17
<i>Clean Up Resources.....</i>	<i>23</i>
<i>Next Steps.....</i>	<i>24</i>

Overview

The following steps detail how to configure, build, and execute the [Device Update for IoT Hub](#) example on the Microchip SAM E54 Xplained Pro. For this example, we will learn how to:

- Create Azure IoT Hub and Device Update for IoT Hub resources
- Prepare the firmware for the dev board and the manifest file
- Prepare the firmware for the simulated leaf device for [proxy update](#) and manifest file required for it
- Upload the files and create the group for the device in the Device Update for IoT Hub
- Deploy the firmware to the device and observe the results

The IDE that was used testing this example is [MPLAB X](#) version 6.05. You will also need MPLAB [XC32](#) compiler version 3.00 to build the example projects. Earlier versions of Microchip's MPLAB X IDE and XC compilers can be found in the [MPLAB Development Ecosystem Downloads Archive](#).

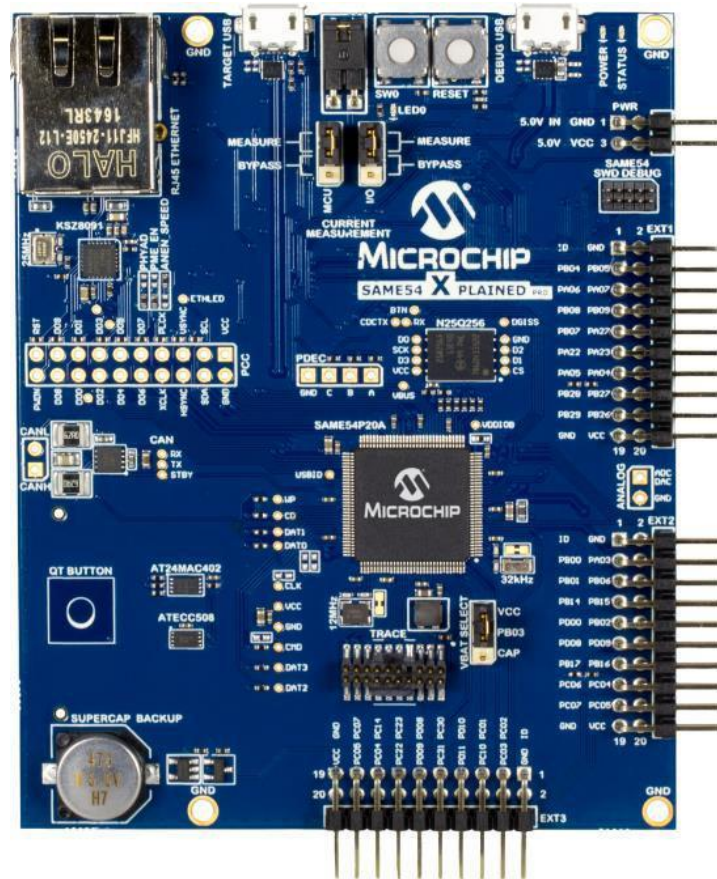


Figure 1 Microchip SAM E54 Xplained Pro

The example distribution file contains the following sub-folders:

Folder	Contents
<i>common_hardware_code</i>	Common code for SAM E54 board
<i>docs</i>	User guides
<i>netxduo</i>	NetX Duo source code
<i>sample_azure_iot_embedded_sdk_pnp</i>	Sample project to connect to Azure IoT Hub using Azure IoT Middleware for Azure RTOS
<i>same54_lib</i>	SAM E54 drivers
<i>threadx</i>	ThreadX source code
<i>filex</i>	FileX source code

Prepare Azure Resources

Create an IoT Hub

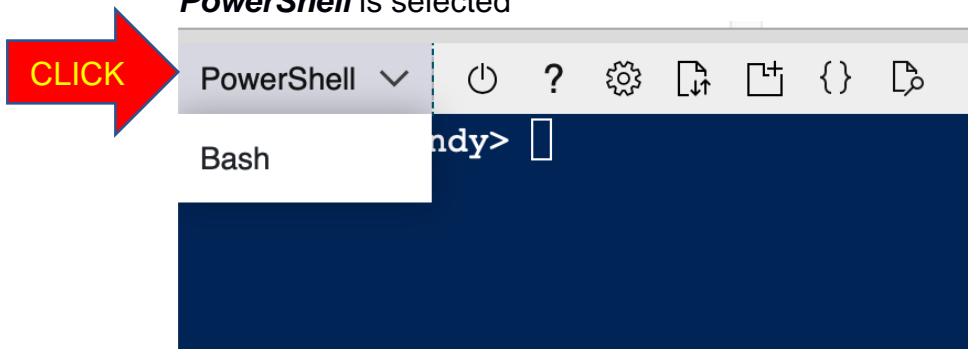
You can use the [Azure Command Line Interface \(CLI\)](#) to create an IoT hub that handles events and messaging for your device.

1. Log into your Azure [portal](#) account and click on the **Cloud Shell** icon located at the top of the web page (you may be prompted to create a storage account in order to use Cloud Shell)



Cloud Shell

2. In the upper left-hand corner of the **Cloud Shell** pane, confirm that **PowerShell** is selected



3. At the PS prompt, run the [az extension add](#) command to add the Microsoft Azure IoT Extension for Azure CLI to your CLI shell. The IoT Extension adds IoT Hub, IoT Edge, and IoT Device Provisioning Service (DPS) specific commands to the Azure CLI.

```
az extension add --name azure-iot
```

4. Run the [az account list](#) command to see the list of subscriptions accessible by your Azure account

```
az account list
```

5. Run the [az account set](#) command to select the specific subscription you want to use for creating the Azure resources

```
az account set --subscription "{MySubscriptionName}"
```

6. Run the 1 (or 2) command(s) required to create a resource group. For example, if your company's policy requires specific tags be added to a resource group, you can create a "tags" variable to point to a list of all the tag name/value pairs required before creating the resource group. Otherwise if no tags are needed, just run the **New-AzResourceGroup** command and without the -tag option. To set a specific location for your resource group, run [az account list-locations](#) to see a list of all available regions.

```
$tags=@{"name1"="val1";"name2"="val2";"name3"="val3";etc}
```

```
New-AzResourceGroup -Name {MyResourceGroup} -Location  
{MyLocation} [-tag $tags]
```

For example, the following 2 command lines will create a resource group named **ADU_TestGroup** with a location of **westus** along with 8 different tags which might be used for a company's policy for accounting purposes:

```
$tags=@{"BusinessOwner"="Rob Stein - C07707";  
"CostCenter"="SESE00"; "CreatedBy"="Randy Wu - C14166";  
"CreationDate"="20230421"; "Env"="DEV"; "Group"="N/A";  
"ManagedBy"="Randy Wu - C14166"; "SNOWTicket"="N/A"}
```

```
New-AzResourceGroup -Name ADU_TestGroup -Location westus  
-tag $tags
```

NOTE: If using Bash (UNIX) CLI, execute the following command line:

```
az group create --name {MyResourceGroup} --location  
{MyLocation} [--tags name1="value1" name2="value2"  
name3="value3" etc...]
```

...for example:

```
az group create --name ADU_TestGroup --location westus --  
tags BusinessOwner="Rob_Stein - C07707" CostCenter=SESE00  
CreatedBy="Randy Wu - C14166" CreationDate=20230421  
Env=DEV Group=N/A ManagedBy="Randy Wu - C14166"  
SNOWTicket=N/A
```

7. Run the [az iot hub create](#) command to create an IoT hub. It might take a few minutes for Azure to finalize the creation of the IoT hub...

Replace ***YourIotHubName*** below with the name you choose for your IoT Hub. An IoT Hub name must be globally unique in Azure.

```
az iot hub create --resource-group {MyResourceGroup} --  
name {YourIoTHubName}
```

... for example:

```
az iot hub create --resource-group "ADU-TestGroup" --name  
"ADU-IoTHub"
```

8. After the IoT hub is created, view the JSON output in the console, and copy the ***hostName*** value to a safe place. You will use this value in a later step. The ***hostName*** value will be in the following format:

```
{YourIoTHubName}.azure-devices.net
```

[TIP] You can look up the host name again by using the following command to list all the IoT Hub host names that are tied to your portal account:

```
az iot hub list --query  
"[].{hostname:properties.hostName}" --output table
```

Register a Device with the IoT Hub

In this section, you create a new device instance and register it with the IoT Hub you created. You will use the connection information for the newly registered device to securely connect your physical device in a later section.

1. In your console, run the [az iot hub device-identity create](#) command. This creates the simulated device identity.

YourIotHubName. Replace this placeholder below with the name you chose for your IoT hub.

MyDevKit. You can use this name directly for the device in CLI commands in this tutorial. Choose a meaningful name for your kit (e.g. ***WFI32-IoT***)

```
az iot hub device-identity create --device-id  
"{MyDevKit}" --hub-name "{YourIoTHubName}"
```

... for example:

```
az iot hub device-identity create --device-id "ATSAME54-  
XPRO" --hub-name "ADU-IoTHub"
```

2. After the device is created, view the JSON output in the console, copy and save the ***deviceId*** and ***primaryKey*** values to use in a future step.

Confirm that you have copied and saved the following values from the JSON output to use in the next section:

- ***hostName***
- ***deviceId***
- ***primaryKey***

TIP: You can get the ***primaryKey*** at any time by executing the following command in the Azure CLI:

```
az iot hub device-identity show --hub-name {YourIoTHubName} --  
device-id {MyDevKit}
```

... for example:

```
az iot hub device-identity show --hub-name "ADU-IoTHub" --  
device-id "ATSAME54-XPRO"
```

Create Device Update Account and Instance

Click on one of the following hyperlinks to create a Device Update account and instance using one of the following methods:

- Azure [portal](#)
- Azure [CLI](#)

Prepare the New Firmware

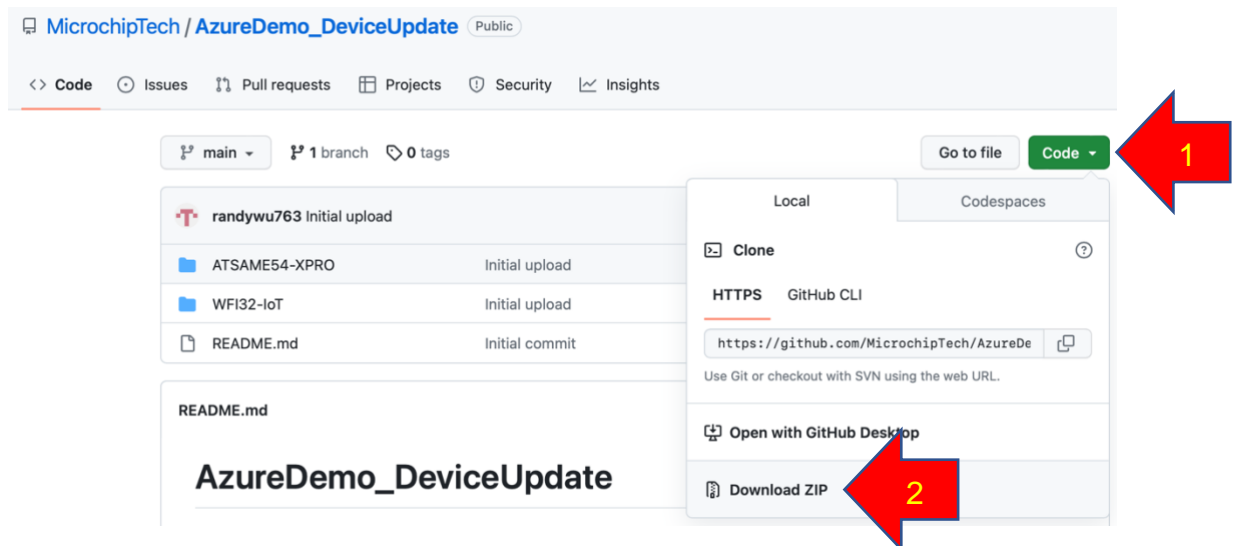
To connect the device to Azure, you'll modify a configuration file for Azure IoT settings, build and flash the image to the device.

Clone/Download the Example Project

Using the Git [command line](#) or [Desktop](#) tool, clone the repository located at the URL specified in the following command line:

```
git clone https://github.com/MicrochipTech/AzureDemo_DeviceUpdate
```

As an alternative, you can simply go to the specified URL and download a ZIP file of the repository:



Build New Firmware Image

1. Launch the MPLAB X IDE and select **File > Open Project** and select the following projects from **AzureDemo_DeviceUpdate > ATSAME54-XPRO > mplab**:
 - threadx
 - netxduo
 - same54_lib
 - sample_azure_iot_embedded_sdk_pnp
2. In the **Project** pane, right-click on the project **sample_azure_iot_embedded_sdk_pnp** and select **Set as main project**.

- Expand the **Header files** folder to open **sample_config.h** to set the Azure IoT device information constants to the values that you saved after you created Azure resources.

Constant name	Value
HOST_NAME	{Your IoT hub hostName value}
DEVICE_ID	{Your deviceId value}
DEVICE_SYMMETRIC_KEY	{Your primaryKey value}

- In the **Source files** folder, open **sample_azure_iot_embedded_sdk_adu.c** and modify the firmware version to mimic a new firmware version that will be deployed from Device Update.

```
#define SAMPLE_DEVICE_INSTALLED_CRITERIA "2.0.0"
```

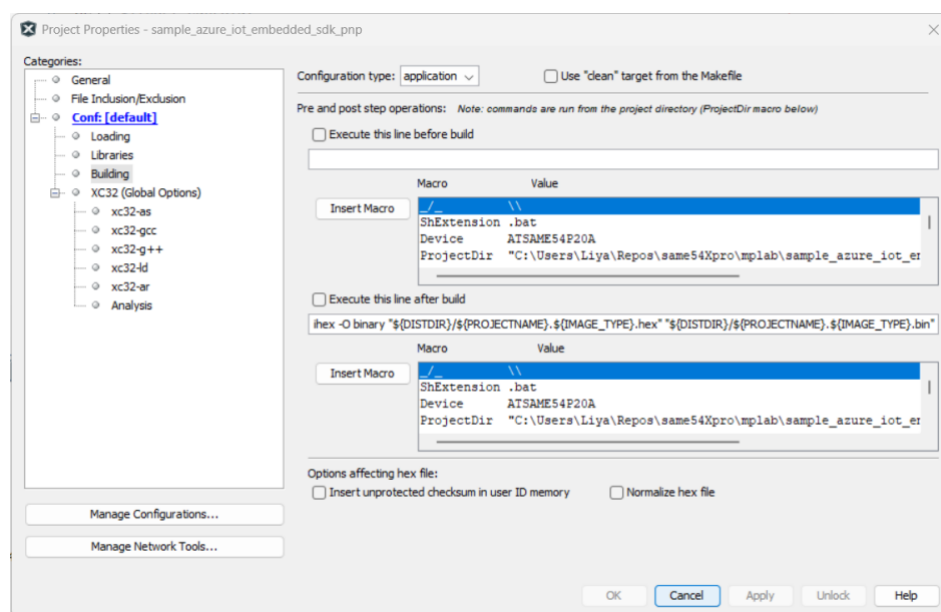
Originally 1.0.0

```
#define SAMPLE_LEAF_DEVICE_INSTALLED_CRITERIA "2.0.0"
```

Originally 1.0.0

- Make sure all of the sample project's dependent libraries (threadx, netxduo, same54_lib) are built by right-clicking on each project in the **Projects** pane and selecting **Clean and Build**.
- MPLAB X [does not create a *.bin file by default](#). Right click on the **sample_azure_iot_embedded_sdk_pnp** project, select **Properties > Building**, check the box "Execute this line after build", and confirm that the following command is in the text box:

```
${MP_CC_DIR}/xc32-objcopy -I ihex -O binary  
"${DISTDIR}/${PROJECTNAME}.${IMAGE_TYPE}.hex"  
"${DISTDIR}/${PROJECTNAME}.${IMAGE_TYPE}.bin"
```



7. In the **Projects** pane, right-click on the project **sample_azure_iot_embedded_sdk_pnp** and select the **Clean and Build** option. Wait for the project to finish building and confirm that the build was successful.
8. The binary firmware image will be generated in the sub-folder **AzureDemo_DeviceUpdate > ATSAME54-XPRO > mplab > sample_azure_iot_embedded_sdk_pnp > dist > default > production**. Copy the ***.bin** file to the **mplab > tools > AzureDeviceUpdateScripts** folder where the scripts for generating manifest files are also located. It is recommended to change the file name that matches with your firmware version (e.g. **firmware_2.0.0.bin**).

You have just prepared the new firmware (binary file) that will be uploaded to the “Device Update for IoT Hub” service. Make sure the size of the binary file does not exceed 900KB for this particular example for the ATSAME54-XPRO development board.

Prepare New Firmware for the Simulated Leaf Device

Even though we do not physically connect any leaf devices to the dev board we are using, it is a common scenario for your MCU device that you might have one or more leaf devices connect to it via UART or USB as peripherals or sensors. For this example, we just use a simulated leaf device and push an empty firmware to demonstrate the experience. To do that:

Create a file and name it **leaf_firmware_2.0.0.bin** and save in the same **mplab > tools > AzureDeviceUpdateScripts** folder with this PowerShell command:

```
New-Item "leaf_firmware_2.0.0.bin" -ItemType File -Value "Empty leaf firmware"
```

```
[ Directory: /Users/c14166/Azure/AzureDemo_DeviceUpdate/ATSAME54-XPRO/mplab/tools/AzureDeviceUpdateScripts
[
  UnixMode      User      Group      LastWriteTime      Size
  -----
  -rw-r--r-- c14166      5/6/2023 12:02      19
```

Generate Import Manifest

An [import manifest](#) is a JSON file that defines important information about the update that you are importing that is required by the Device Update for IoT Hub. You can learn the detailed steps about importing new update from [here](#). For this sample:

1. Ensure you have installed [PowerShell v7.0](#) or above.
2. In a PowerShell window, navigate to **AzureDemo_DeviceUpdate > ATSAME54-XPRO > mplab > tools > AzureDeviceUpdateScripts** directory and run:

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope Process
```

You may see the following result which is fine and can be ignored:

Set-ExecutionPolicy: Operation is not supported on this platform.

3. Run the following script and follow the prompt to generate the manifest files for the dev board and the simulated leaf device:

```
.\CreateSAME54Update.ps1
```

```
Supply values for the following parameters:
```

```
(Type !? for Help.)
```





```
Version: 2.0.0
```

```
HostPath: ./firmware_2.0.0.bin
```

```
LeafPath: ./leaf_firmware_2.0.0.bin
```

Replace the values with the actual path and version you have for the firmware. You can also type !? for the help.

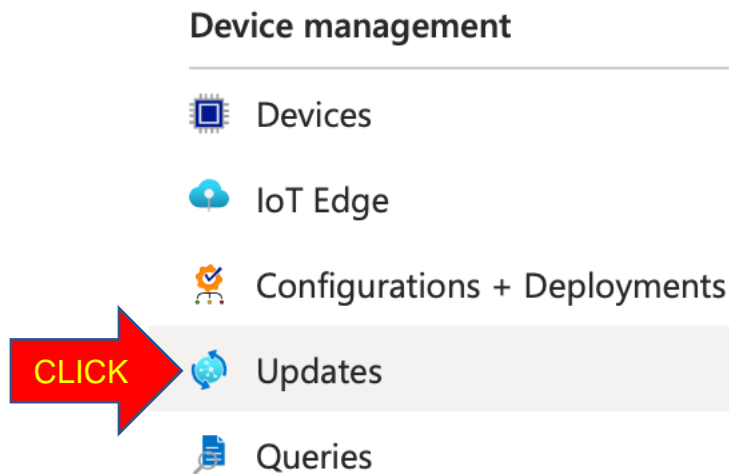
4. Upon successful completion of the script, all files required (to upload into the Device Update service) are generated in the sub-folder named **MICROCHIP.SAME54.2.0.0**

Name	
	firmware_2.0.0.bin
	leaf_firmware_2.0.0.bin
	MICROCHIP.SAME54-Leaf.2.0.0.importmanifest.json
	MICROCHIP.SAME54.2.0.0.importmanifest.json

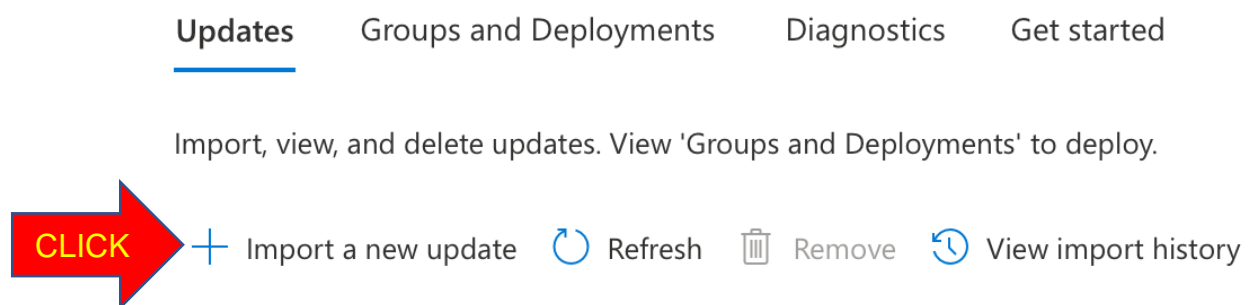
Upload Firmware and Manifest

1. Access your Azure [portal](#) account to find the IoT Hub resource which was created earlier. You may have to first access the resource group that contains it in order to find the IoT Hub. When the IoT Hub has been found in your Azure portal, click on the IoT Hub name to access its configuration details.

2. Using the left-hand navigation pane, select **Device Management > Updates**




3. Locate the **Updates** tab and then select **Import a new update**







4. Choose **+ Select from storage container**. If no storage accounts show up in the list, create a new one now by selecting **+ Storage account**.
5. From the list of Storage accounts, click on the name of the newly-created (or pre-existing) storage account that you would like to use for storing the manifest files and binary images required for the Device Update process.
6. You should now be on the **Containers** page. If no containers show up in the list, select **+ Container** and enter in a name for the new container. Click on the **Create** button.
7. In the list of existing Containers, click on the name of the container that you would like to use for storing the Device Update files.
8. Select **Upload** and browse to (or drag-and-drop) all of firmware image and manifest files required for the deployment from the **MICROCHIP.SAME54.2.0.0** folder.





9. Click on the **Upload** button. You should see the message “Successfully uploaded blob(s)” and then see a list of all the uploaded files (4 blobs total):

 **Successfully uploaded blob(s)**
Successfully uploaded 4 blob(s).

Current uploads

firmware_2.0.0.bin	 502.68 KiB / 502.68 KiB
leaf_firmware_2.0.0.bin	 19 B / 19 B
MICROCHIP.SAME54-Lea...	 826 B / 826 B
MICROCHIP.SAME54.2.0....	 1.01 KiB / 1.01 KiB

Name

<input type="checkbox"/>	 firmware_2.0.0.bin
<input type="checkbox"/>	 leaf_firmware_2.0.0.bin
<input type="checkbox"/>	 MICROCHIP.SAME54-Leaf.2.0.0.importmanifest.json
<input type="checkbox"/>	 MICROCHIP.SAME54.2.0.0.importmanifest.json

Now you have the new firmware images and manifest files uploaded into a container, which prepares you for creating a deployment in a future section. It's now time to turn our attention to the device and get it up and running to receive the new firmware.

Prepare the Device

We will build and run the same device application but for a previous version, so that later we can observe the new firmware being deployed to update to the newer version.

Revert to the Previous Firmware Version

1. In MPLAB, open **sample_azure_iot_embedded_sdk_adu.c**, modify version to an older one to mimic current firmware:

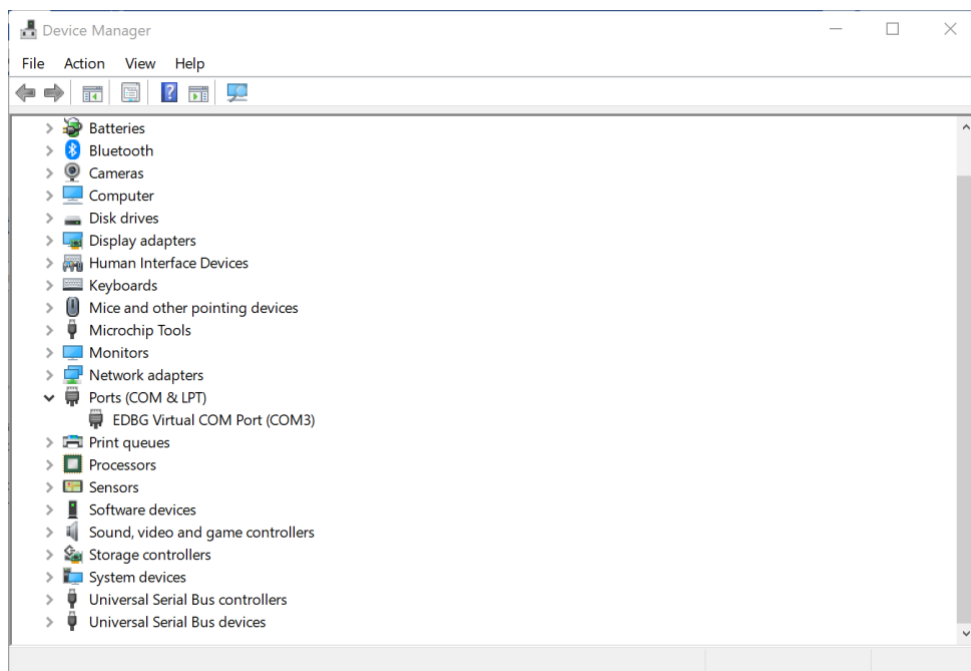
```
#define SAMPLE_DEVICE_INSTALLED_CRITERIA "1.0.0"
```

```
#define SAMPLE_LEAF_DEVICE_INSTALLED_CRITERIA "1.0.0"
```

2. Now rebuild the **sample_azure_iot_embedded_sdk_pnp** project.

Program and Run the Device

1. Use the micro USB cable to connect the **Debug USB** port on the Microchip ATSAME54-XPRO, and then connect it to your computer.
2. Use the Ethernet cable to connect the Microchip ATSAME54-XPRO to an Ethernet port.
3. Verify the serial port in your OS's device manager. It should show up as a Virtual COM port:

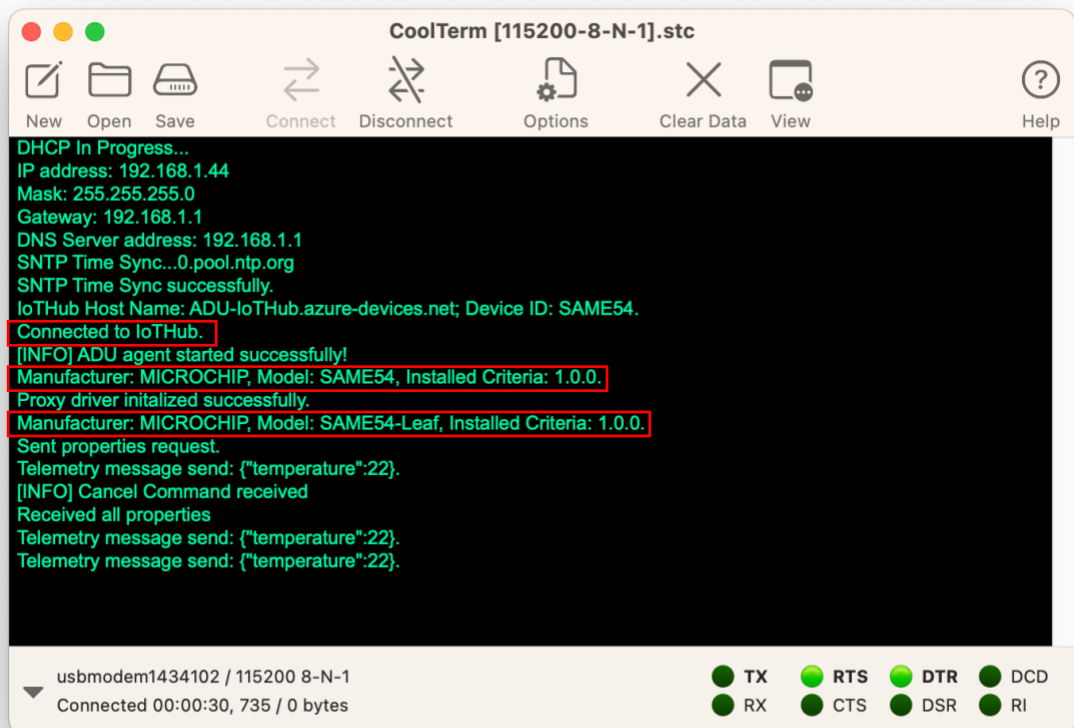


4. Open your favorite serial terminal program such as [Putty](#) or [Tera Term](#) and connect to the COM port discovered above at **115200** baud.
5. In MPLAB, navigate to the main toolbar and click the **Make and Program Device** icon.



6. Go to the terminal program window. As the demo project runs on the device, it continuously prints out status information to the terminal output window. Check the terminal output to verify that there are no error messages and that there is a message stating that the device has “Connected to IoT Hub”. Note the output message towards the end that displays the installed criteria version number.

NOTE: The terminal output content may vary depending on which example project you choose to build and run. Keep the terminal window open to monitor device output in subsequent steps.



The screenshot shows the CoolTerm application window titled "CoolTerm [115200-8-N-1].stc". The window has a menu bar with icons for New, Open, Save, Connect, Disconnect, Options, Clear Data, View, and Help. The main terminal area displays the following output:

```
DHCP In Progress...
IP address: 192.168.1.44
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS Server address: 192.168.1.1
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
IoT Hub Host Name: ADU-IoTHub.azure-devices.net; Device ID: SAME54.
Connected to IoT Hub.
[INFO] ADU agent started successfully!
Manufacturer: MICROCHIP, Model: SAME54, Installed Criteria: 1.0.0.
Proxy driver initialized successfully.
Manufacturer: MICROCHIP, Model: SAME54-Leaf, Installed Criteria: 1.0.0.
Sent properties request.
Telemetry message send: {"temperature":22}.
[INFO] Cancel Command received
Received all properties
Telemetry message send: {"temperature":22}.
Telemetry message send: {"temperature":22}.
```

At the bottom of the window, the connection status is shown as "usbmodem1434102 / 115200 8-N-1" and "Connected 00:00:30, 735 / 0 bytes". On the right side of the bottom bar, there are status indicators for TX, RX, RTS, CTS, DTR, DSR, DCD, and RI, all of which are currently green.

Deploy New Firmware

Add a Tag to Your Device

1. Confirm that the device application is still running from the previous step (i.e. normal execution was never interrupted).
2. Log into the Azure [portal](#) and navigate to the IoT Hub (you may need to access the resource group that's tied to the IoT Hub in order to view it).
3. Using the left-hand navigation pane, select **Device Management > Devices**.
4. Your device ID should show up in the list of devices; click directly on its name in the list.
5. Click on the **Device Twin** tab.
6. Create a deployment group by adding a tag to the Device Twin:

```
"tags": {  
  "ADUGroup": "{MyDeploymentGroupName}"  
}  
  
15   "modelId": "dtmi:com:example:Thermostat;3",  
16   "version": 6,  
17   "tags": {  
18     "ADUGroup": "TestDevices"  
19   },  
20   "properties": {  
21     "desired": {  
22       "$metadata": {  
23         "$lastUpdated": "2023-04-22T20:06:11.6280639Z"  
24       },  
25     }  
26   }  
27 }
```

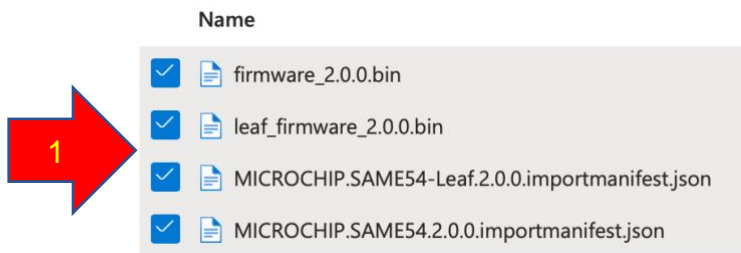
NOTE: To remove any tag in the future, the tag value can be set to 'null'

7. Click on the **Save** icon in the upper left-hand corner of the page, then click on the **X** in the upper right-hand corner of the page to close the **Device twin** view.

Deploy New Firmware Update

1. In the Azure [portal](#), navigate to the IoT Hub that you previously connected to your Device Update instance (you may need to navigate to the resource group which contains the IoT Hub in order to access it).
2. Using the left-hand navigation pane, select **Devices > Updates**

3. Select **+ Import a new update**.
4. Select **+ Select from storage container**.
5. Click on the name of the blob storage which contains the container that stores the manifest files and binary images.
6. Click on the container that stores the manifest files and binary images.
7. Check the boxes for all the files in the list and then click the **Select** button at the bottom of the page.



8. Confirm that the manifest and binaries show up on the resulting page. Click the **Import update** button at the bottom of the page.

✓ MICROCHIP | SAME54 | 2.0.0
MICROCHIP.SAME54.2.0.0.importmanifest.json
firmware_2.0.0.bin
✓ MICROCHIP | SAME54-Leaf | 2.0.0
MICROCHIP.SAME54-Leaf.2.0.0.importmanifest.json
leaf_firmware_2.0.0.bin



- You may get a warning message regarding the status of the import; ignore the message and proceed with the next step.

Import update

A problem occurred monitoring import of update:

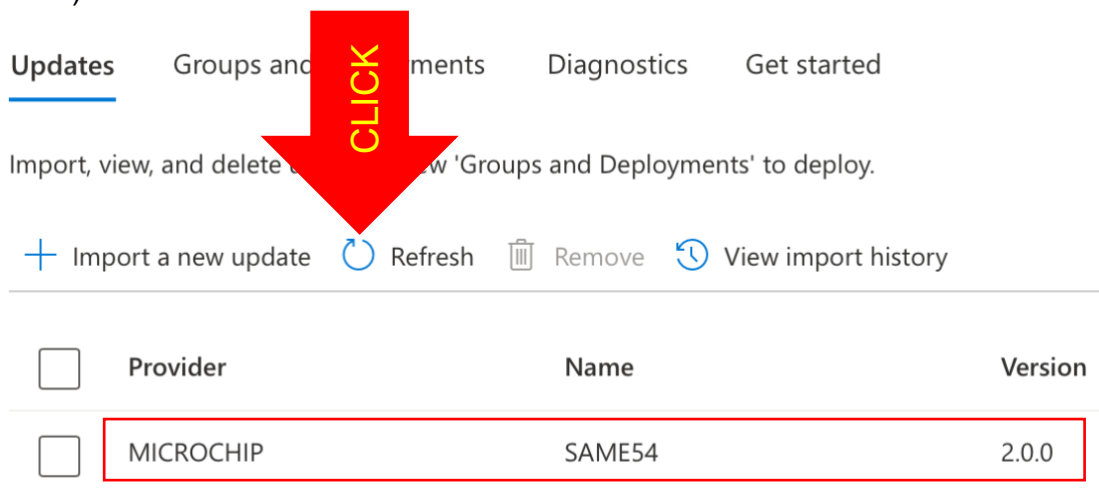
Provider: MICROCHIP

Name: SAME54

Version: 2.0.0

The update may still successfully import. You can monitor status by viewing import history under [IoT Hub Updates](#).

- Confirm that a new update was created showing the correct firmware version number. It may take up to a few minutes for the new update to appear in the list, so click on the **Refresh** icon periodically until the WFI32 update shows up in the list (now may be a good time for a break).



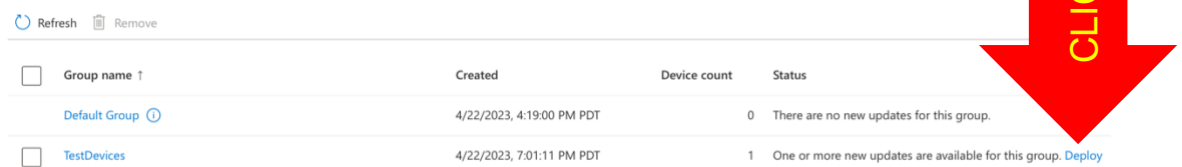
Updates Groups and Deployments Diagnostics Get started

Import, view, and delete updates. Click on 'Groups and Deployments' to deploy.

[+ Import a new update](#) [Refresh](#) [Remove](#) [View import history](#)

<input type="checkbox"/>	Provider	Name	Version
<input type="checkbox"/>	MICROCHIP	SAME54	2.0.0

- Click on the **Groups and Deployments** tab.
- The name of the deployment group that you just created (e.g. "TestDevices") should show up under the list of device groups. Click on the **Deploy** link which should show up under the **Status** column for the deployment group.

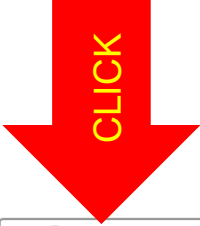


[Refresh](#) [Remove](#)

<input type="checkbox"/>	Group name ↑	Created	Device count	Status
<input type="checkbox"/>	Default Group ⓘ	4/22/2023, 4:19:00 PM PDT	0	There are no new updates for this group.
<input type="checkbox"/>	TestDevices	4/22/2023, 7:01:11 PM PDT	1	One or more new updates are available for this group. Deploy

- The **New Updates** pop-up window should appear, showing the correct firmware version for deployment. Click on the **→Deploy** button.

Provider	MICROCHIP
Name	SAME54
Version	2.0.0
More update details	
> Target devices	



Deploy

14. The **Create deployment** pop-up window should appear. Leave the default option as **Start immediately** and click on the **Create** button to start the deployment.

Create deployment

ADU-IoTHub

Create a new deployment targeting devices in this group.

Update properties

Provider	MICROCHIP
Name	SAME54
Version	2.0.0

Group name

TestDevices

Specify when this deployment should start. *

- ☒ Start immediately.
- ☐ Start at a scheduled date and time.

15. Go back to observing the terminal output window. You can see the update firmware is pushed from ADU to the device. After completing the upload, the device will reboot with the new firmware.

CoolTerm [115200-8-N-1].stc

New Open Save Connect Disconnect Options Clear Data View Help

```
DHCP in Progress...
IP address: 192.168.1.44
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS Server address: 192.168.1.1
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
IoT Hub Host Name: ADU-IoTHub.azure-devices.net; Device ID: SAME54.
Connected to IoT Hub.
[INFO] ADU agent started successfully!
Manufacturer: MICROCHIP, Model: SAME54, Installed Criteria: 1.0.0.
Proxy driver initialized successfully.
Manufacturer: MICROCHIP, Model: SAME54-Leaf, Installed Criteria: 1.0.0.
Sent properties request.
Telemetry message send: ("temperature":22).
Received all properties
Received new update: Provider: MICROCHIP; Name: SAME54, Version: 2.0.0
[INFO] Updating firmware...
[INFO] Manufacturer: MICROCHIP
[INFO] Model: SAME54-Leaf
[INFO] Firmware downloading...
Proxy driver flash erased successfully.
Proxy driver firmware writing...
[INFO] Getting download data... 0
Proxy driver firmware writing...
[INFO] Getting download data... 19
[INFO] Firmware downloaded
[INFO] Firmware installing...
Proxy driver firmware installed successfully.
[INFO] Firmware installed
[INFO] Updating firmware...
[INFO] Manufacturer: MICROCHIP
[INFO] Model: SAME54
[INFO] Firmware downloading...
[INFO] Getting download data... 705
[INFO] Getting download data... 2165
[INFO] Getting download data... 3625
[INFO] Getting download data... 5085
[INFO] Getting download data... 6545
[INFO] Getting download data... 8005
[INFO] Getting download data... 9465
[INFO] Getting download data... 10925
[INFO] Getting download data... 12385
[INFO] Getting download data... 13032
[INFO] Getting download data... 14492
[INFO] Getting download data... 15629
[INFO] Getting download data... 17089
[INFO] Getting download data... 18549
```

usbmodem1434102 / 115200 8-N-1
Connected 00:02:38, 2,650 / 0 bytes

TX RTS DTR DCD
RX CTS DSR RI



CoolTerm [115200-8-N-1].stc

New Open Save Connect Disconnect Options Clear Data View Help

```
[INFO] Getting download data... 491599
[INFO] Getting download data... 498554
[INFO] Getting download data... 500914
[INFO] Getting download data... 501474
[INFO] Getting download data... 502934
[INFO] Getting download data... 504394
[INFO] Getting download data... 505854
[INFO] Getting download data... 506176
[INFO] Getting download data... 507636
[INFO] Getting download data... 509096
[INFO] Getting download data... 510556
[INFO] Getting download data... 511694
[INFO] Getting download data... 513154
[INFO] Getting download data... 513476
[INFO] Getting download data... 514744
[INFO] Firmware downloaded
Telemetry message send: ("temperature":22).
[INFO] Firmware installing...
[INFO] Firmware installed
[INFO] Applying firmware...
[INFO] Manufacturer: MICROCHIP
[INFO] Model: SAME54-Leaf
[INFO] Firmware applying...
Proxy driver firmware apply successfully.
[INFO] Firmware applied

[INFO] Applying firmware...
[INFO] Manufacturer: MICROCHIP
[INFO] Model: SAME54
[INFO] Firmware applying...
DHCP in Progress...
IP address: 192.168.1.44
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS Server address: 192.168.1.1
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
IoT Hub Host Name: ADU-IoTHub.azure-devices.net; Device ID: SAME54.
Connected to IoT Hub.
[INFO] ADU agent started successfully!
Manufacturer: MICROCHIP, Model: SAME54, Installed Criteria: 2.0.0.
Proxy driver initialized successfully.
Manufacturer: MICROCHIP, Model: SAME54-Leaf, Installed Criteria: 2.0.0.
Sent properties request.
Telemetry message send: ("temperature":22).
Received all properties
Telemetry message send: ("temperature":22).
Telemetry message send: ("temperature":22).
Telemetry message send: ("temperature":22).
```

usbmodem1434102 / 115200 8-N-1
Connected 00:00:03, 45 / 0 bytes

TX RTS DTR DCD
RX CTS DSR RI

16. To cancel the deployment, click on the **Groups and Deployments** tab and then locate the deployment group (e.g. "TestDevices") under the list of device groups. Click on the **Deploy** link which should show up under the **Status** column for the deployment group.

[Refresh](#) [Remove](#)

<input type="checkbox"/> Group name ↑	Created	Device count	Status
Default Group ⓘ	4/22/2023, 4:19:00 PM PDT	0	There are no new updates for this group.
<input type="checkbox"/> TestDevices	4/22/2023, 7:01:11 PM PDT	1	One or more new updates are available for this group. Deploy

CLICK

17. Locate the name/version of the device update and click on its **View** button.

Provider

MICROCHIP

Name

SAME54

Version

2.0.0

More update details

> Target devices

Current update

View

CLICK

18. On the **Current updates** tab, click on the **Cancel deployment** button.

Group basics

Current updates

Deployment history

[Refresh](#)

Update

Provider

MICROCHIP

Name

SAME54

Version

2.0.0

[More update details](#)

Deployment details

Start time

5/6/2023, 12:40:04 PM PDT

Retried

No

Cloud rollback

No

Status

Active

Retry failed devices

Cancel deployment

[View available updates](#)

Target devices

Devices in this group with attributes:

manufacturer: MICROCHIP

model: SAME54

Contract Name: Device Update Model V1

Contract ID: dtmi:azure:iot:deviceUpdateContractModel;1

[Assign a descriptive label](#)

Device statistics

Total devices

1

In progress

0

Canceled

0

Succeeded

1

Failed

0

[View devices](#)

CLICK

Clean Up Resources

If you no longer need the Azure resources created in this tutorial, you can use the Azure CLI to delete the resource group and all the resources you created for this tutorial. Optionally, you can use Azure IoT Explorer to delete individual resources including devices and IoT hubs.

If you continue to another tutorial in this getting started guide, you can keep the resources you've already created and reuse them.

Important: Deleting a resource group is irreversible. The resource group and all the resources contained in it are permanently deleted. Make sure that you do not accidentally delete the wrong resource group or resources.

To delete a resource group by name:

1. Run the [az group delete](#) command. This removes the resource group, the IoT Hub, and the device registration you created.

```
az group delete --name MyResourceGroup
```

2. Run the [az group list](#) command to confirm the resource group is deleted.

```
az group list
```

Next Steps

In this tutorial you created an IoT Hub and added the Device Update resource to it. Then you prepared and deployed the new firmware for the dev board and the simulated leaf device.

To learn more about the APIs of the Device Update agent for Azure RTOS, or the Device Update for IoT Hub service, view <https://aka.ms/azrtos/adu>

To learn more about Azure RTOS and how it works with Azure IoT, view <https://azure.com/rtos>.