



Device Update Sample Microchip SAM E54 Xplained Pro using IAR User Guide

Published: November 2022

For the latest information, please see
azure.com/rtos

This document is provided “as-is”. Information and views expressed in this document, including URL and other Internet Web site references, may change without notice.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Azure RTOS provides OEMs with components to secure communication and to create code and data isolation using underlying MCU/MPU hardware protection mechanisms. It is ultimately the responsibility of the device builder to ensure the device fully meets the evolving security requirements associated with its specific use case.

© 2022 Microsoft. All rights reserved.

Microsoft Azure RTOS, Azure RTOS FileX, Azure RTOS GUIX, Azure RTOS GUIX Studio, Azure RTOS NetX, Azure RTOS NetX Duo, Azure RTOS ThreadX, Azure RTOS TraceX, Azure RTOS Trace, event-chaining, picokernel, and preemption-threshold are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

Revision 6.2

Table of Contents

<i>Overview</i>	<i>4</i>
<i>Prepare Azure Resources</i>	<i>7</i>
Create an IoT Hub and register a device.....	7
Create Device Update account.....	7
<i>Prepare the new firmware</i>	<i>8</i>
Build new firmware	8
Prepare new firmware for the simulated leaf device.....	9
Generate import manifest.....	9
Upload firmware and manifest.....	10
<i>Prepare the device</i>	<i>11</i>
Modify the configuration	11
Download and run the project.....	11
<i>Deploy new firmware.....</i>	<i>14</i>
Add a tag to your device.....	14
Deploy update	14
<i>Clean up resources</i>	<i>16</i>
<i>Next steps</i>	<i>17</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 sample, we will learn:

- Create Azure IoT Hub and Device Update for IoT Hub resources.
- Prepare the firmware for the dev board and the manifest file required for it.
- 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 result of it.

The tool used in the sample is IAR's EWARM 9.20.0 or later. A 30-day free trial of IAR's EWARM can be downloaded from this page:

<https://www.iar.com/iar-embedded-workbench/#!?architecture=Arm>

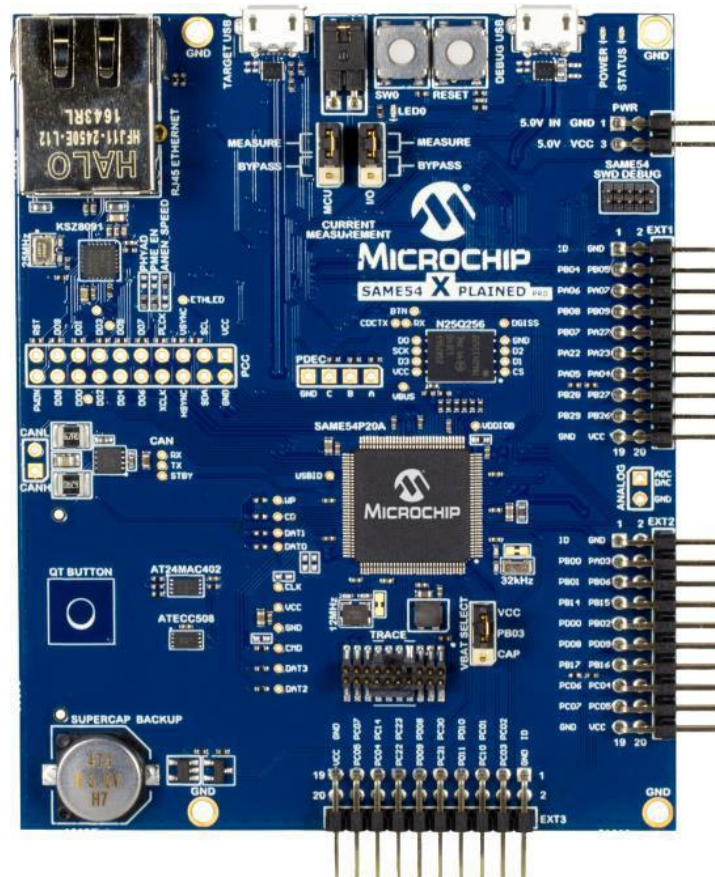


Figure 1 Microchip SAM E54 Xplained Pro

The sample distribution zip file contains the **azure_rtos.eww** workspace as well as following sub-folders:

Folder	Contents
<i>iar</i>	Contains the following sub-folders as well as the azure_rtos.eww workspace
<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 and register a device

Follow the same steps in *Azure_RTOS_ATSAME54-XPRO_Azure_IoT_Embedded_SDK_For_IAR.pdf*.

Create Device Update account

Follow this guide to create a device update account using Azure portal:
<https://learn.microsoft.com/azure/iot-hub-device-update/create-device-update-account>

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.

Build new firmware

1. Open the **azure_rtos.eww** EWARM Workspace in IAR from the extracted zip file.
2. Select the **sample_azure_iot_embedded_sdk_pnp** sample, right click on it on in the left **Workspace** pane and select **Set as active**.
3. Expand the sample 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}

4. In the same folder, open **sample_azure_iot_embedded_sdk_adu.c**, modify the firmware version to mimic it is a new firmware that will be deployed from Device Update.

```
#define SAMPLE_DEVICE_INSTALLED_CRITERIA "1.1.0"
```

```
#define SAMPLE_LEAF_DEVICE_INSTALLED_CRITERIA "1.1.0"
```

5. Select **Project > Batch Build** and choose **build_all** and select **Make** to build all projects. You will observe compilation and linking of the sample projects.
6. The binary firmware will be generated in **iar\sample_azure_iot_embedded_sdk_pnp\Debug\Exe**. Copy **sample.bin** to **iar\tools\AzureDeviceUpdateScripts** folder where the scripts for generating manifest files are within. It is optional but recommended also change the file name that matches with your firmware version such as **firmware_1.1.0.bin**.

Now you have prepared the new firmware for the dev board that will be uploaded to Device Update for IoT Hub.

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. But 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 sample, 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 as **leaf_firmware_1.1.0.bin** and save in the same **iar/tools/AzureDeviceUpdateScripts** folder with this PowerShell command:

```
New-Item "leaf_firmware_1.1.0.bin" -ItemType File -Value  
"Empty leaf firmware"
```

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 PowerShell, navigate to **iar/tools\ AzureDeviceUpdateScripts** directory and run:

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

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: 1.1.0  
HostPath: ./firmware_1.1.0.bin  
LeafPath: ./leaf_firmware_1.1.0.bin
```

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

4. All files required to upload will be generated in **MICROCHIP.SAME54.1.1.0** folder.

Upload firmware and manifest

1. Open the IoT Hub you created before with Device Update enabled from Azure portal.
2. Select **Device management > Updates**, and in **Updates** tab select **Import a new update**.
3. Choose **Select from storage container**. For the storage container, you can create a new or use existing storage container to host the firmware file.
4. Select **Upload** and choose all firmware and manifest files you will use for the deployment in **MICROCHIP.SAME54.1.1.0** folder.
5. You can go to **Import history** to see the progress of publishing the files. And once done, it will show in the **Available updates** tab.

Now you have the new firmware and manifest files uploaded for deployment. It's time to make the device be up and running to receive the new firmware.

Prepare the device

We will build and run the same device application but on an older version, so later we can observe the new firmware deployed to it.

Modify the configuration

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

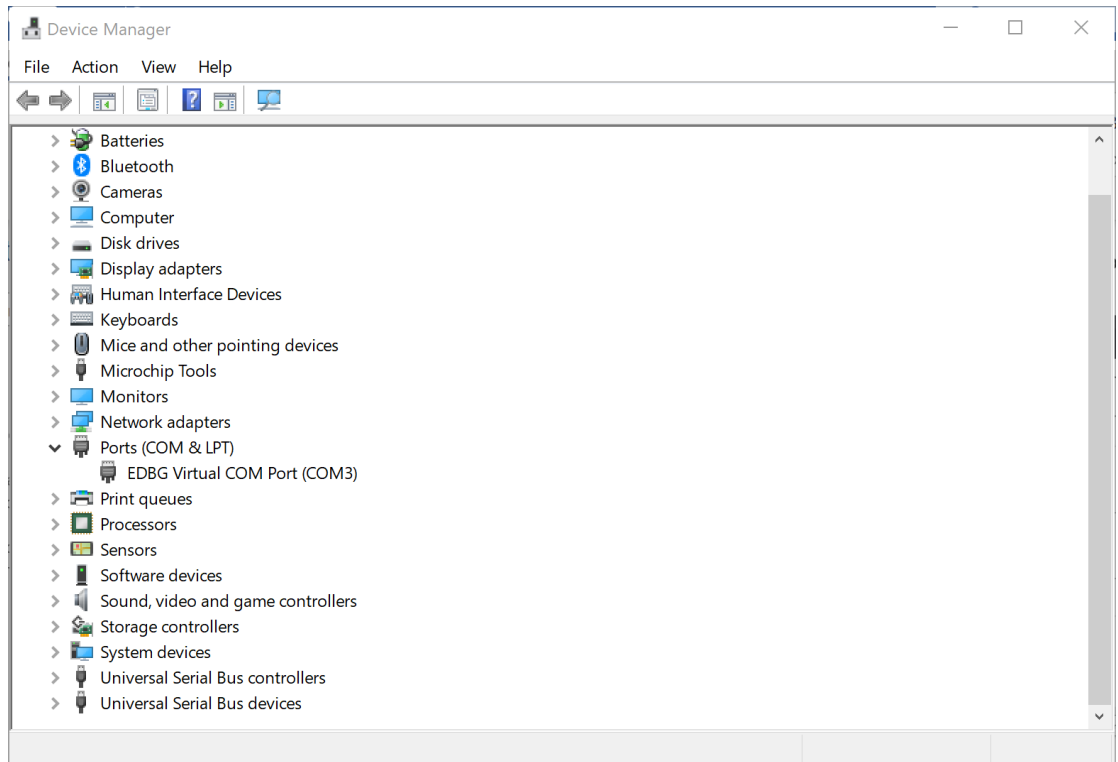
```
#define SAMPLE_DEVICE_INSTALLED_CRITERIA "1.0.0"
```

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

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

Download and run the project

1. Use the Micro USB cable to connect the ***Debug USB*** port on the Microchip SAM E54, and then connect it to your computer.
2. Use the Ethernet cable to connect the Microchip SAM E54 to an Ethernet port.
3. In IAR, press the green ***Download and Debug*** button in the toolbar to download the program and run it. Then press ***Go***.
4. Verify the serial port in your OS's device manager. It should show up as a COM port.



5. Open your favorite serial terminal program such as [Putty](#) or [Tera Term](#) and connect to the COM port discovered above. Configure the following values for the serial ports:

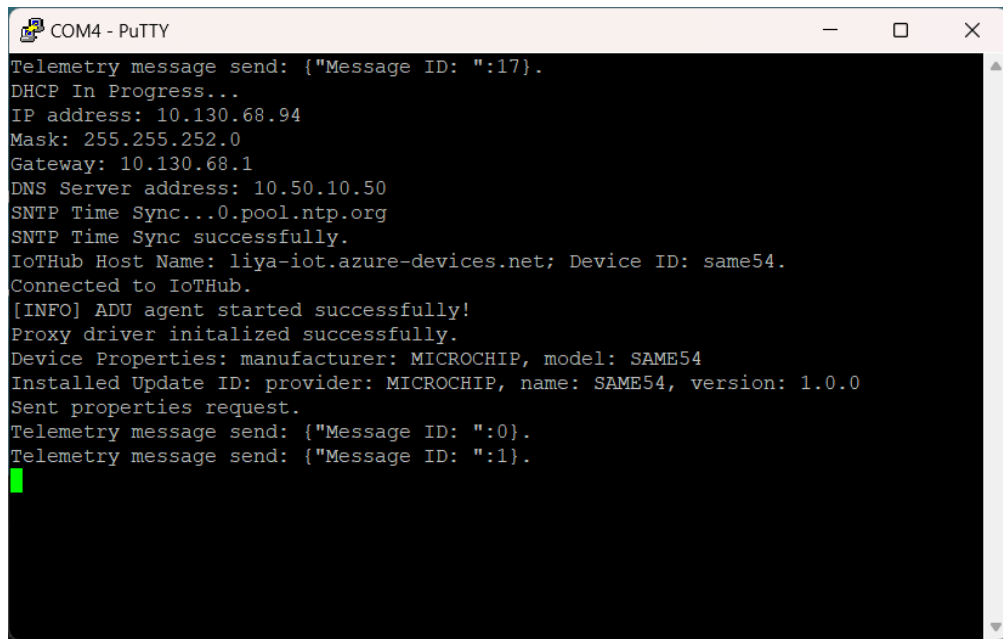
Baud rate: **115200**

Data bits: **8**

Stop bits: **1**

6. As the project runs, the demo prints out status information to the terminal output window. Check the terminal output to verify that messages have been successfully sent to the Azure IoT hub and the version number of the firmware is running.

NOTE: The terminal output content varies depending on which sample you choose to build and run.



```
COM4 - PuTTY
Telemetry message send: {"Message ID: ":17}.
DHCP In Progress...
IP address: 10.130.68.94
Mask: 255.255.252.0
Gateway: 10.130.68.1
DNS Server address: 10.50.10.50
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
IoT Hub Host Name: liya-iot.azure-devices.net; Device ID: same54.
Connected to IoT Hub.
[INFO] ADU agent started successfully!
Proxy driver initialized successfully.
Device Properties: manufacturer: MICROCHIP, model: SAME54
Installed Update ID: provider: MICROCHIP, name: SAME54, version: 1.0.0
Sent properties request.
Telemetry message send: {"Message ID: ":0}.
Telemetry message send: {"Message ID: ":1}.
```

Keep terminal window open to monitor device output in subsequent steps.

Deploy new firmware

Add a tag to your device

1. Keep the device application running from the previous step.
2. Log into Azure portal and navigate to the IoT Hub.
3. From **IoT Devices**, select the IoT device you use and navigate to **Device Twin** tab.
4. Delete any existing Device Update tag value by setting them to null.

```
"tags": {  
  "ADUGroup": null  
}
```

5. Add a new Device Update tag value:

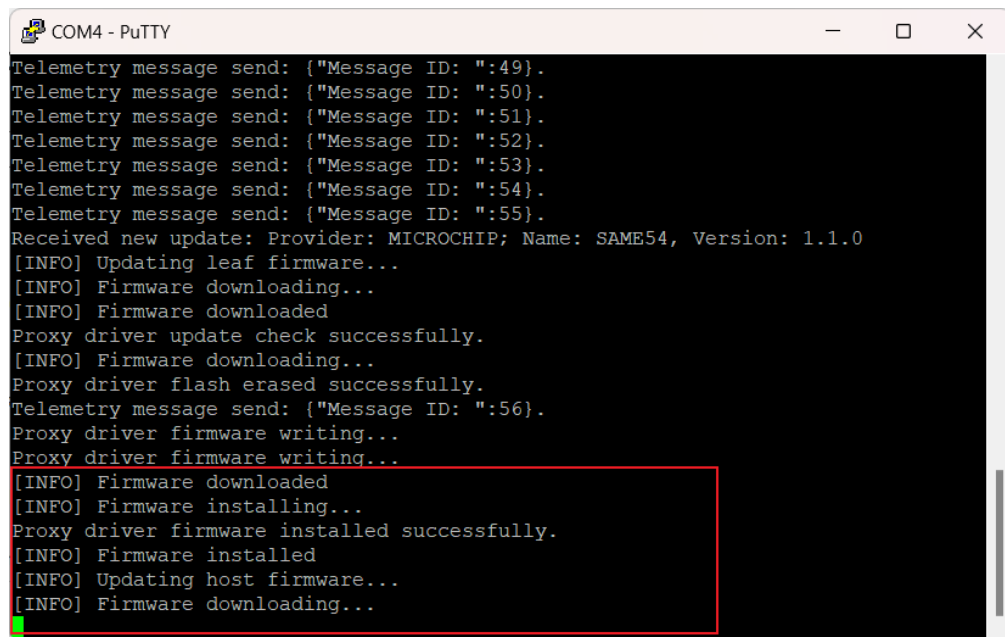
```
"tags": {  
  "ADUGroup": "<CustomTagValue>"  
}
```



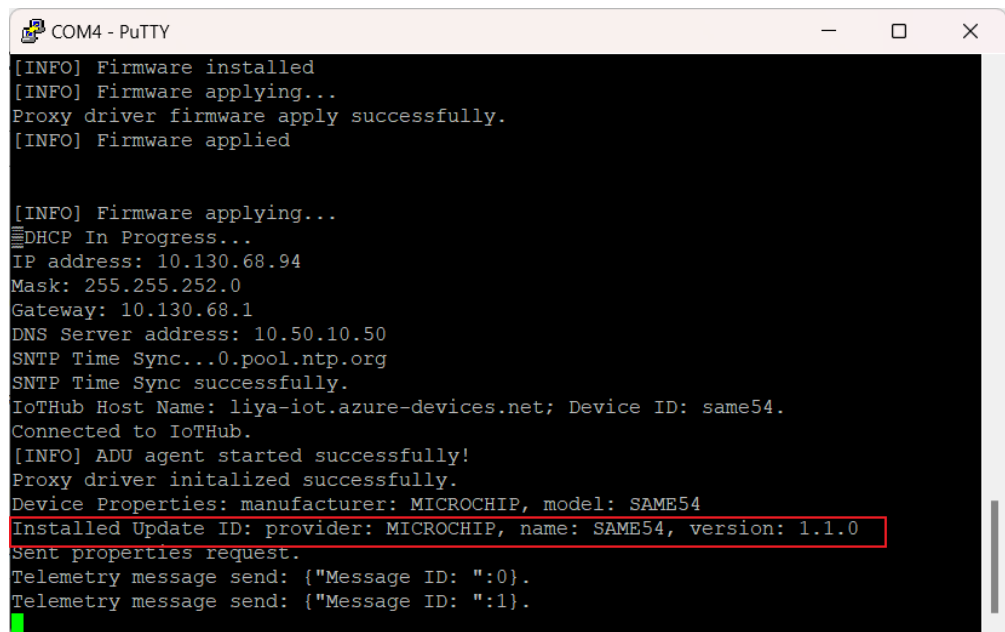
Deploy update

1. In the same **Groups and Deployments** tab, next to the group you just created, choose **Deploy**.
2. Leave the default option as **Start immediately** and choose **Create** to start a deployment.

3. Go back to terminal output window, you can see the update firmware is pushed from ADU to the device and after downloading it, the device will reboot with the new firmware:

A terminal window titled 'COM4 - PuTTY' showing the progress of a firmware update. The output includes several telemetry messages with IDs 49 through 55. It then reports receiving a new update from MICROCHIP (SAME54, Version: 1.1.0) and proceeds with updating the leaf firmware, downloading the proxy driver, and erasing the flash. A red box highlights the final steps: firmware installation and host firmware update.

```
COM4 - PuTTY
Telemetry message send: {"Message ID: ":49}.
Telemetry message send: {"Message ID: ":50}.
Telemetry message send: {"Message ID: ":51}.
Telemetry message send: {"Message ID: ":52}.
Telemetry message send: {"Message ID: ":53}.
Telemetry message send: {"Message ID: ":54}.
Telemetry message send: {"Message ID: ":55}.
Received new update: Provider: MICROCHIP; Name: SAME54, Version: 1.1.0
[INFO] Updating leaf firmware...
[INFO] Firmware downloading...
[INFO] Firmware downloaded
Proxy driver update check successfully.
[INFO] Firmware downloading...
Proxy driver flash erased successfully.
Telemetry message send: {"Message ID: ":56}.
Proxy driver firmware writing...
Proxy driver firmware writing...
[INFO] Firmware downloaded
[INFO] Firmware installing...
Proxy driver firmware installed successfully.
[INFO] Firmware installed
[INFO] Updating host firmware...
[INFO] Firmware downloading...
```

A terminal window titled 'COM4 - PuTTY' showing the device's initialization and connection to IoT Hub. It reports firmware installation, applying the proxy driver, and DHCP progress. Network details like IP address (10.130.68.94) and gateway (10.130.68.1) are shown. It then reports connecting to IoT Hub and starting the ADU agent. A red box highlights the device properties and the installed update information.

```
COM4 - PuTTY
[INFO] Firmware installed
[INFO] Firmware applying...
Proxy driver firmware apply successfully.
[INFO] Firmware applied

[INFO] Firmware applying...
DHCP In Progress...
IP address: 10.130.68.94
Mask: 255.255.252.0
Gateway: 10.130.68.1
DNS Server address: 10.50.10.50
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
IoT Hub Host Name: liya-iot.azure-devices.net; Device ID: same54.
Connected to IoT Hub.
[INFO] ADU agent started successfully!
Proxy driver initialized successfully.
Device Properties: manufacturer: MICROCHIP, model: SAME54
Installed Update ID: provider: MICROCHIP, name: SAME54, version: 1.1.0
Sent properties request.
Telemetry message send: {"Message ID: ":0}.
Telemetry message send: {"Message ID: ":1}.
```

Since the simulated leaf device just does nothing. The terminal output will only show that the empty firmware is deployed to the device and device will echo back to it.

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>.