User Guide:

Azure IoT Hub Device Update for the WFI32-IoT Development Board

Table of Contents

0

Overview

The following steps detail how to configure, build, and execute the [Device Update for IoT Hub](#) example on the Microchip WFI32-IoT Development Board. 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 required manifest file
- Prepare the firmware for the simulated leaf device for [proxy update](#) and the required manifest file
- 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 two installations of the MPLAB [XC32](#) compilers (specifically versions 3.01 & 4.10) 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](#).
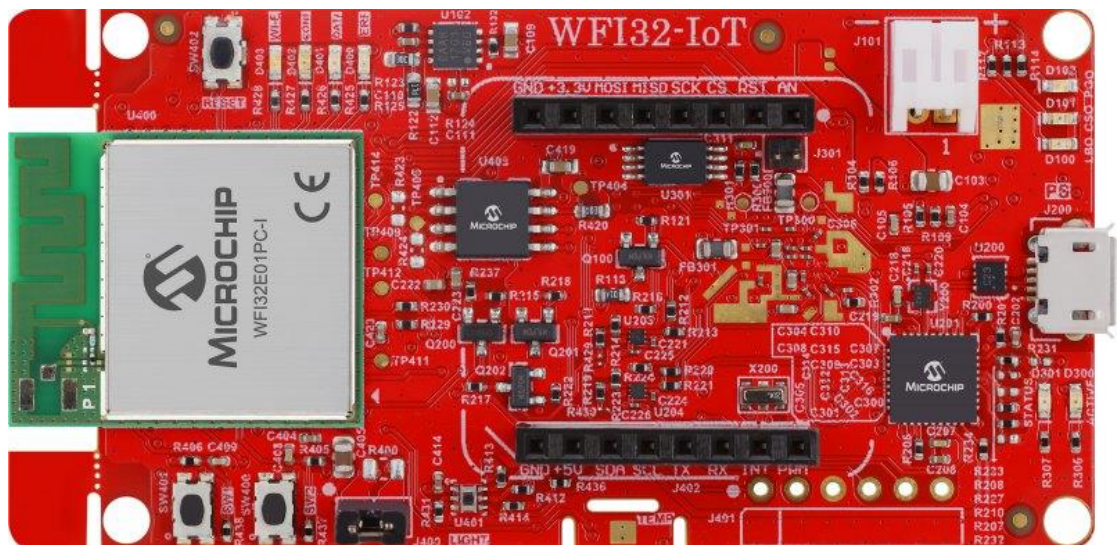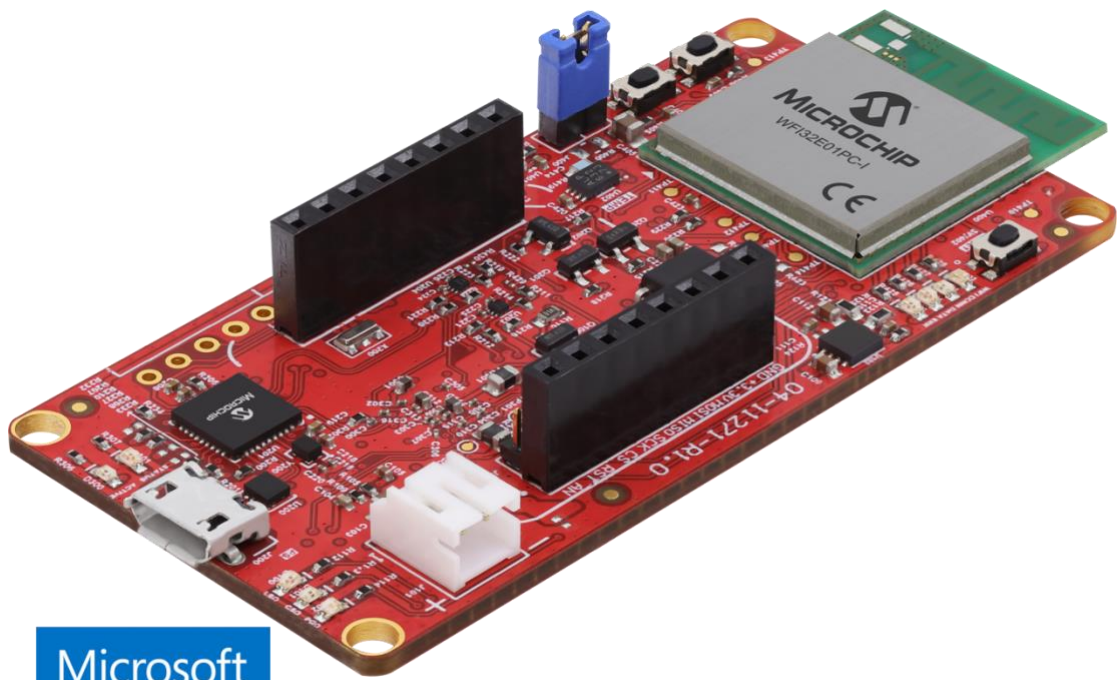
Figure 1 Microchip WFI32-IoT Development Board

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)

2. In the upper left-hand corner of the **Cloud Shell** pane, confirm that *PowerShell* is selected
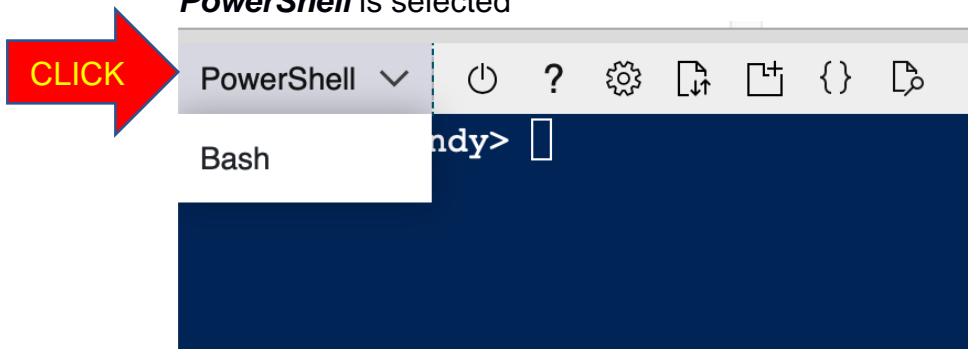
3. At the PS prompt, run the `az extension add` command to add the Microsoft Azure IoT Extension for Azure CLI to your CLI shell. The IoT Extension adds IoT Hub, IoT Edge, and IoT Device Provisioning Service (DPS) specific commands to the Azure CLI.

```
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}"
```

5

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 <u>az account list-locations</u> 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 <u>az iot hub create</u> 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 <u>az iot hub device-identity create</u> 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 "WFI32-IoT" --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 the 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 "WFI32-IoT"
```

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 New Firmware Image

To connect the device to Azure, you'll modify a configuration file for Azure IoT settings, build the example project, 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:
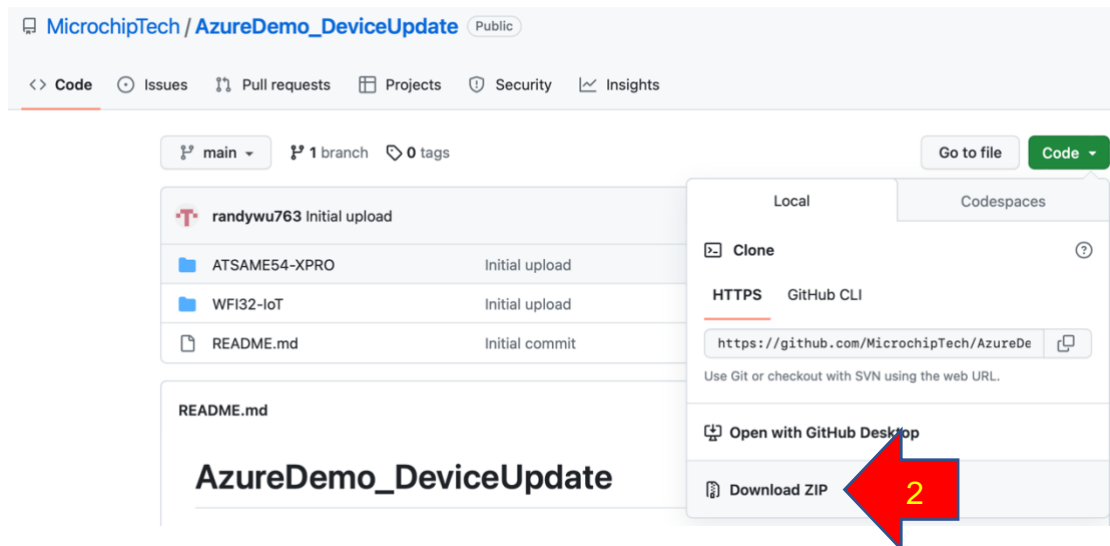
Build New Firmware

1. *Launch the MPLAB X IDE and select* ***File > Open Project*** *from the main toolbar. Navigate to the example project folder by browsing to* ***AzureDemo_DeviceUpdate > WFI32-IoT > AzureDemo_WFI32E01 > firmware > AzureDemo_WFI32-IoT.X***

2. In the ***Projects*** pane, right-click on ***AzureDemo_WFI32-IoT*** and select ***Set as Main Project***.

3. In the ***Projects*** pane, expand the main ***Header Files*** folder and navigate to the ***azure_rtos_demo > sample_azure_iot_embedded_sdk_pnp*** sub-folder.

4. Double-click the ***sample_config.h*** header file to set the Azure IoT device information constants to the values that you copied and saved after you created Azure resources in a previous section.
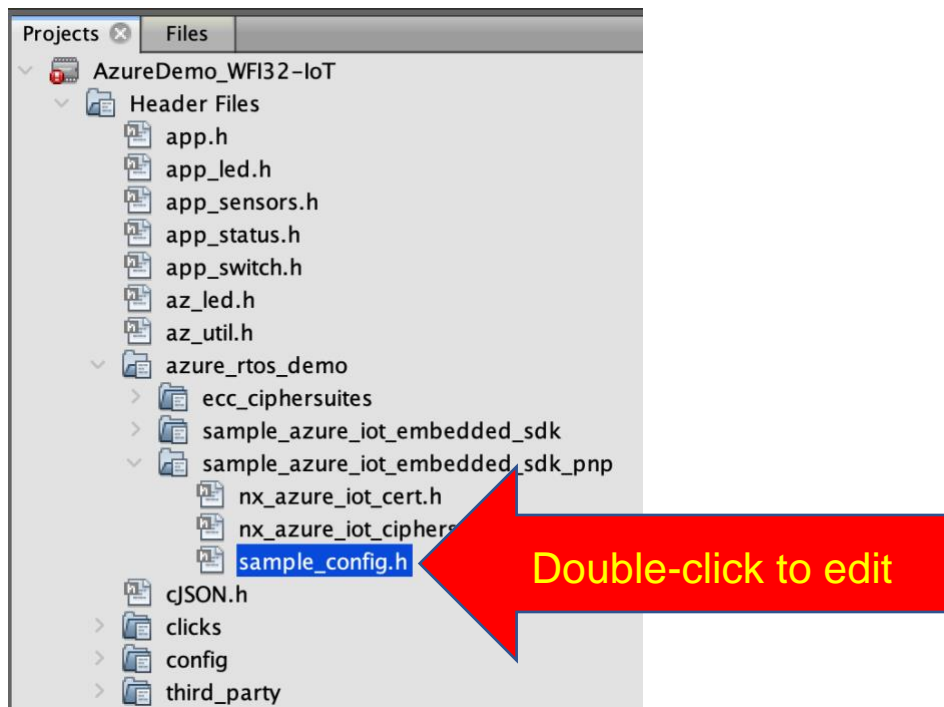
Constant name	Value
HOST_NAME	{Your IoT Hub **hostName** value}
DEVICE_ID	{Your **deviceID** value}
DEVICE_SYMMETRIC_KEY	{Your **primaryKey** value}

…for example:

```
67   ⊟   #ifndef HOST_NAME
68   │   #define HOST_NAME                              "ADU-IoTHub.azure-devices.net"
69   └   #endif /* HOST_NAME */
70
71   ⊟   #ifndef DEVICE_ID
72   │   #define DEVICE_ID                              "WFI32-IoT"
73   └   #endif /* DEVICE_ID */


92       /* Optional SYMMETRIC KEY. */
93   ⊟   #ifndef DEVICE_SYMMETRIC_KEY
94   │   #define DEVICE_SYMMETRIC_KEY                   "mWFZ695gwrBhWgV+I6Z/LaBbcAvdJMqMCaMv6j6aVKE="
95   └   #endif /* DEVICE_SYMMETRIC_KEY */
```

5. In the *Projects* pane, expand the **Source Files** folder and navigate to the *azure_rtos_demo > sample_azure_iot_embedded_sdk_pnp* sub-folder. Double-click on *sample_azure_iot_embedded_sdk_adu.c* to modify the firmware version (e.g. 1.0.0 to 1.1.0) to mimic the need for a new firmware deployment from Azure Device Update.

```
#define SAMPLE_DEVICE_INSTALLED_CRITERIA "1.1.0"
```
Originally 1.0.0

6. In the *Projects* pane, right-click on the *AzureDemo_WFI32-IoT* project and select the *Clean and Build* option. Wait for the project to finish building and confirm that the build was successful.

   NOTE: XC32 compiler version 4.10 is required for a successful build; higher versions may result in the program not running properly.

7. Open a command line window (Command Prompt, PowerShell, Bash) and navigate to the folder **AzureDemo_DeviceUpdate > WFI32-IoT > AzureDemo_WFI32E01 > tools > hex2bin**

8. Execute the following command line (for MacOS users only):

   > pip3 install colorama

9. Execute one of the following command lines based on your Operating System to convert the project's HEX (*.hex) file into a binary (*.bin) file

   MacOS

   > python3 hex2bin.py -I ../../firmware/AzureDemo_WFI32-

   IoT.X/dist/pic32mz_w1/production/AzureDemo_WFI32-

   IoT.X.production.unified.hex -o AzureDemo_WFI32-

   IoT.X.production.unified.bin

   ```
   hex2bin V1.2
   Generating AzureDemo_WFI32-IoT.X.production.unified.bin (10020000 ~ 100F5980 874880 bytes)

   "Digest":"4023254036659040bbaba2598b595e2339be2ce0e343ceedbbe134c5c4b97877"
   ```

   Windows/Linux

   > hex2bin.exe -i ../../firmware/AzureDemo_WFI32-

   IoT.X/dist/pic32mz_w1/production/AzureDemo_WFI32-

   IoT.X.production.unified.hex -o AzureDemo_WFI32-

   IoT.X.production.unified.bin

10. Rename the binary file to "firmware_<version>.bin" (e.g. "firmware_1.1.0.bin")

11. Copy the newly-named binary file to the folder
AzureDemo_DeviceUpdate > WFI32-IoT > AzureDemo_WFI32E01 > tools > AzureDeviceUpdateScripts

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 WFI32-IoT development board.

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 about the detailed steps of importing a new update here. For this example, perform the following steps:

1. Ensure you have installed PowerShell v7.0 or higher.

2. In a PowerShell window, navigate to the **AzureDemo_DeviceUpdate > WFI32-IoT > AzureDemo_WFI32E01 >** *tools >* *AzureDeviceUpdateScripts* directory and run the command:

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

3. Run the Python script and then enter info when prompted to generate the manifest files for the dev board and the simulated leaf device:

```
> .\CreateWFI32Update.ps1
```

```
Supply values for the following parameters:
(Type !? for Help.)
Version: 1.1.0
HostPath: ./firmware_1.1.0.bin
```

```
 Supply values for the following parameters:
 (Type !? for Help.)
[Version: 1.1.0
[HostPath: ./firmware_1.1.0.bin
 Preparing update MICROCHIP.WFI32.1.1.0 ...
 Preparing parent update MICROCHIP.WFI32.1.1.0 ...
 Generating an import manifest MICROCHIP.WFI32.1.1.0...
     Saving parent manifest file and payload(s) to .\MICROCHIP.WFI32.1.1.0...
```

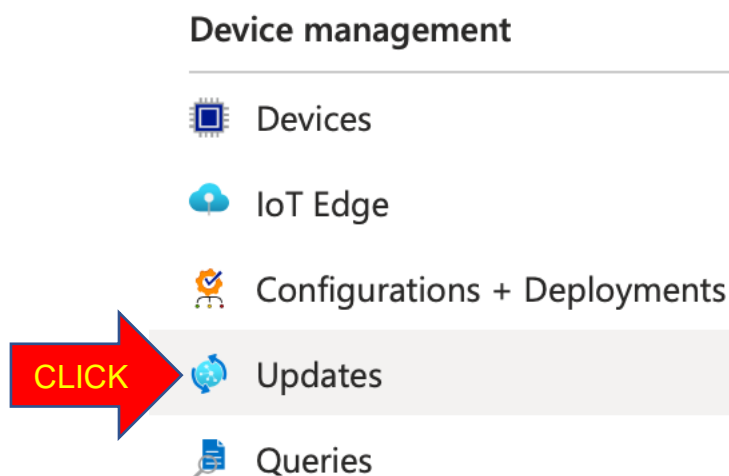NOTE: The script can also be run with just a single command line; e.g.

```
> .\CreateWFI32Update.ps1 -Version 1.1.0 -HostPath ./firmware_3.0.0.bin
```

4. Upon successful completion of the script, all files required (to upload into the Device Update service) is generated in a sub-folder named **MICROCHIP.WFI32.1.1.0**

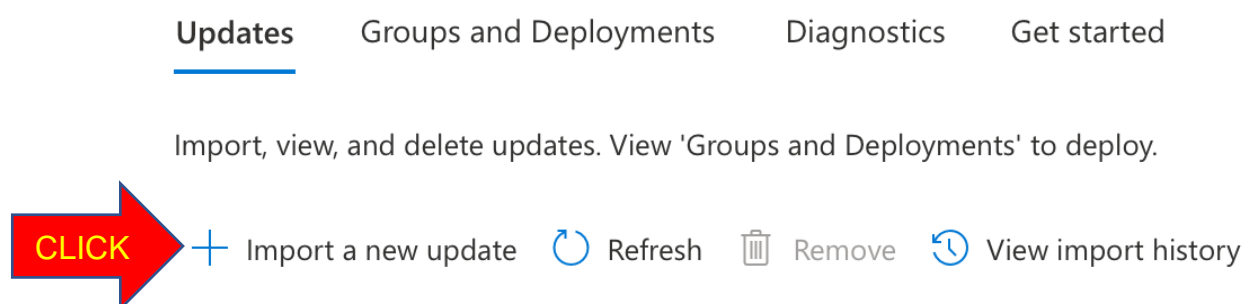| Name	^	Size
firmware_1.1.0.bin		875 KB
MICROCHIP.WFI32.1.1.0.importmanifest.json		807 bytes

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**

Device management

🔲 Devices

☁ IoT Edge

✅ Configurations + Deployments

◀ CLICK 🌐 Updates

📄 Queries

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

Updates	Groups and Deployments	Diagnostics	Get started

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

◀ CLICK ＋ Import a new update   ↻ Refresh   🗑 Remove   ⟲ View import history

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) the firmware and manifest file required for the deployment from the **MICROCHIP.WFI32.1.1.0** 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 (2 blobs total):

✅ **Successfully uploaded blob(s)**

Successfully uploaded 2 blob(s).

wfi32iot ⋯
Container

⤒ Upload   ↻ Refresh | ⧉ Give feedback

Authentication method: Access key (Switch to Azure AD User Account)
Location: wfi32iot

Search blobs by prefix (case-sensitive)

⊹ Add filter

	Name
☐	📄 firmware_1.1.0.bin
☐	📄 MICROCHIP.WFI32.1.1.0.importmanifest.json

Now you have the new firmware image and associated manifest file uploaded into a storage container, which prepares you for creating a deployment in a future section. These device update files are now being hosted by Microsoft Azure. 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.

Configure Wi-Fi Settings

1. Connect a micro-USB cable to the USB connector on the Microchip WFI32-IoT Development Board, and then connect the other end to your computer's USB port.

2. Using a File Explorer window (or Desktop view), confirm that a Mass Storage Device (most likely named *CURIOSITY*) appears.

3. Click here and use the web-based configurator tool to configure the WFI32-IoT Development Board's device settings with your wireless router's SSID, network type, and password. To be on the safe side, there should be no spaces used in the SSID and password for your network, and the Wi-Fi Access Point should be operating in the 2.4 GHz frequency band.

Wireless Network Connection

Wireless Network Login

My_SSID

Your Wi-Fi® information is not transmitted anywhere —the config file is generated in your browser.

Network Type

○ Open

● WPA/WPA2

○ WEP

●●●●●●●●●●●

Show password

Download Configuration

Revert to the Previous Firmware Version

1. In the *Projects* pane of the MPLAB X IDE, expand the *Source Files* folder and double-click on the source file *sample_azure_iot_embedded_sdk_adu.c* and modify the firmware version number back to the original:

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

2. In the *Projects* pane, right-click on the *AzureDemo_WFI32-IoT* project and select *Clean and Build*. Verify that the build process is successful before proceeding.

Program and Run the Device

1. Confirm that the WFI32-IoT Development Board is connected to the computer via the micro USB cable.

2. Click the *Make and Program Device* icon in the main toolbar.

3. Launch your OS's device manager and verify the serial port connection corresponding to the WFI32-IoT Development Board's USB connection – it should show up as a Virtual COM port:
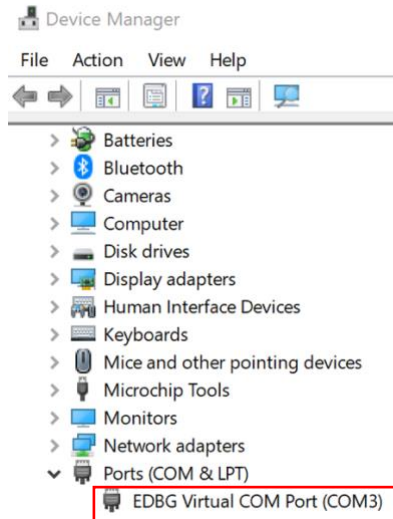
4. Open your favorite serial terminal program such as Putty or Tera Term and connect to the COM port number discovered in the device manager (use **115200-8-N-1**).

5. Press the **RESET** button on the WFI32-IoT Development Board.

6. 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 IoTHub". Note the output message towards the end that displays the installed criteria version number. At this point, the device is just waiting for the ADU to upload new firmware.

   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 messages in subsequent steps.

```
*********************************************************
BOOTLOADER Build Apr 22 2023 16:02:48
OTA VERSION 1.0.0
*********************************************************
status    :FF
version   :01
type      :02
boot_addr :B0021000

BOOTLOADER_TASK_CHECK_IMAGE
new image not found

BOOTLOADER_TASK_JUMP_TO_APP
status    :FF
version   :01
type      :02
boot_addr :B0021000

*********************************************************
DHCP In Progress...
Azure IoT Glue: Initialization Success: PIC32MZW1
Serial Number of the Device: 0123FDAC5A45B6CA01
Device Certificate Subject Common Name (used as a registration ID in DPS) is
 sn0123FDAC5A45B6CA01, length is 20

SSID:NETGEAR51 - PASSPHRASE:melodicship232 - AUTH:2
id_scope:0ne009F3487 - registration_id:PIC32MZW1 - primary_key:mujHRQMx8dUsZETtlWxSonGZ24++L69c8KjlvZDT+5M=
APP_WIFI Ready
Check credentials!
**********111****

****debug***
Connecting to Wi-Fi (NETGEAR51)
WiFi Connected
IP address: 192.168.1.43
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...1.pool.ntp.org
SNTP Time Sync...2.pool.ntp.org
SNTP Time Sync successfully.
IoTHub Host Name: ADU-IoTHub.azure-devices.net; Device ID: WFI32-IoT.
Connected to IoTHub.
SYS OTA : File Opened

***debug*** create_image_file ok
 [INFO] ADU agent started successfully!
Manufacturer: MICROCHIP, Model: WFI32, Installed Criteria: 1.0.0.
Sent properties request.
Received all properties
```

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              },
```

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 **Device Management > 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.

wfi32iot ...
Container

⤒ Upload ⟳ Refresh │ ⚆ Give feedback

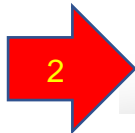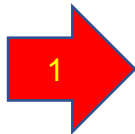**Authentication method:** Access key (Switch to Azure AD User Account)
Location: wfi32iot

Search blobs by prefix (case-sensitive)

⁺▽ Add filter

Name

☑ 📄 firmware_1.1.0.bin

☑ 📄 MICROCHIP.WFI32.1.1.0.importmanifest.json

1

2 Select

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

9. You may get a warning message regarding a potential problem of the import operation; ignore the message and proceed with the next step.

🛑 Import update

A problem occurred monitoring import of update:

Provider: MICROCHIP
Name: WFI32
Version: 1.1.0

The update may still successfully import. You can monitor status by viewing import history under IoT Hub Updates.

10. Confirm that a new update was created showing the correct firmware version number. It may take a minute or two for the update to appear in the list, so click on the **_Refresh_** icon periodically until the WFI32 update shows up in the list (this might be a good time to take a quick break ☺).

Updates Groups and CLICK ...yments Diagnostics Get started

Import, view, and delete ... ew 'Groups and Deployments' to deploy.

+ Import a new update ↻ Refresh 🗑 Remove ↺ View import history

	Provider	Name	Version
☐	MICROCHIP	WFI32	1.1.0

11. Click on the **Groups and Deployments** tab.

12. 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

	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.
☐	TestDevices	4/22/2023, 7:01:11 PM PDT	1	One or more new updates are available for this group. Deploy

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

New updates
ADU-IoTHub

View best available updates for devices in the group.

Provider	MICROCHIP
Name	WFI32
Version	1.1.0

More update details

> Target devices

⊕ Deploy

14. The **Create deployment** pop-up window should appear. For specifying when this deployment should start, leave the default option as **Start immediately** and click on the **Create** button to start the deployment.

15. Go back to observing the terminal output window. You can see the updated firmware is being pushed from your ADU to the device. After completing the upload, the device will reboot itself – showing that a new image has been found and ultimately the new firmware version.

```
*********************************************************
BOOTLOADER Build May  7 2023 09:22:54
OTA VERSION 1.0.0
*********************************************************
status   :FF
version  :01
type     :02
boot_addr :B0021000

BOOTLOADER_TASK_CHECK_IMAGE
new image not found

BOOTLOADER_TASK_JUMP_TO_APP
status   :FF
version  :01
type     :02
boot_addr :B0021000

*********************************************************
DHCP In Progress...
Azure IoT Glue: Initialization Success: PIC32MZW1
Serial Number of the Device: 012319AAC99CF42A01
Device Certificate Subject Common Name (used as a registration ID in DPS) is
 sn012319AAC99CF42A01, length is 20

SSID:WU_BasementBedroom - PASSPHRASE:Baobei71798 - AUTH:2
id_scope:0ne008AC844 - registration_id:PIC32MZW1 - primary_key:mujHRQMx8dUsZETtlWxSonGAPP_WIFI Ready
Check credentials!
**********111****

****debug***
Connecting to Wi-Fi (WU_BasementBedroom)
WiFi Connected
IP address: 192.168.0.195
Mask: 255.255.255.0
Gateway: 192.168.0.1
DNS Server address: 192.168.0.1
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync...1.pool.ntp.org
SNTP Time Sync successfully.
IoTHub Host Name: ADU-IoTHub.azure-devices.net; Device ID: WFI32-IoT.
Connected to IoTHub.
SYS OTA : File Opened

***debug*** create_image_file ok
 [INFO] ADU agent started successfully!
Manufacturer: MICROCHIP, Model: WFI32, Installed Criteria: 1.0.0.
Sent properties request.
Received new update: Provider: MICROCHIP; Name: WFI32, Version: 1.1.0
[INFO] Updating firmware...
[INFO] Manufacturer: MICROCHIP
[INF
 ***debug*** SYS_FS_FileWrite OK
[INFO] Getting download data... 694

 ***debug*** SYS_FS_FileWrite OK
[INFO] Getting download data... 2154
```

```
 ***debug*** SYS_FS_FileWrite OK
[INFO] Getting download data... 874121

 ***debug*** SYS_FS_FileWrite OK
[INFO] Getting download data... 875000
[INFO] Firmware downloaded
[INFO] Firmware installing...
[I.
********************************************************
BOOTLOADER Build May  7 2023 09:22:54
OTA VERSION 1.0.0
********************************************************
status    :FF
version   :01
type      :02
boot_addr :B0021000

BOOTLOADER_TASK_CHECK_IMAGE
new image found

BOOTLOADER_TASK_PROGRAM_IMAGE
ctx->copy_len : 875000
BOOTLOADER_TASK_PROGRAM_IMAGE success
BOOTLOADER_TASK_JUMP_TO_APP
status    :00
version   :00
type      :01
boot_addr :B0021000


********************************************************
DHCP In Progress...
Azure IoT Glue: Initialization Success: PIC32MZW1
Serial Number of the Device: 012319AAC99CF42A01
Device Certificate Subject Common Name (used as a registration ID in DPS) is
 sn012319AAC99CF42A01, length is 20

SSID:WU_BasementBedroom - PASSPHRASE:Baobei71798 - AUTH:2
id_scope:0ne008AC844 - registration_id:PIC32MZW1 - primary_key:mujHRQMx8dUsZETtlWxSonGAPP_WIFI Ready
Check credentials!
**********111****

****debug***
Connecting to Wi-Fi (WU_BasementBedroom)
WiFi Connected
IP address: 192.168.0.195
Mask: 255.255.255.0
Gateway: 192.168.0.1
DNS Server address: 192.168.0.1
SNTP Time Sync...0.pool.ntp.org
SNTP Time Sync successfully.
IoTHub Host Name: ADU-IoTHub.azure-devices.net; Device ID: WFI32-IoT.
Connected to IoTHub.
SYS OTA : File Opened

***debug*** create_image_file ok
 [INFO] ADU agent started successfully!
Manufacturer: MICROCHIP, Model: WFI32, Installed Criteria: 1.1.0.
Sent properties request.
Received all properties
|
```

RESET

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.

	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.
☐	TestDevices	4/22/2023, 7:01:11 PM PDT	1	One or more new updates are available for this group. Deploy

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

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

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.