

AMD KD240 Getting Started Guide for AWS IoT Greengrass V2

Table of Contents

Table of Contents.....	1
1. Document information.....	1
2. Overview.....	2
3. Hardware description.....	3
4. Set up your development environment.....	3
5. Set up device hardware.....	4
6. About AWS IoT Greengrass.....	7
7. Greengrass prerequisites.....	7
8. Install AWS IoT Greengrass.....	7
9. Installing the component.....	14
10. Debugging.....	16
11. Troubleshooting.....	18

1. Document information

Note that all instructions in this document have been written assuming a Kria KD240 Drives Starter Kit..

1.1 Document revision history

30-Dec-2023	Initial Draft
10-Apr-2024	Updated Document Format

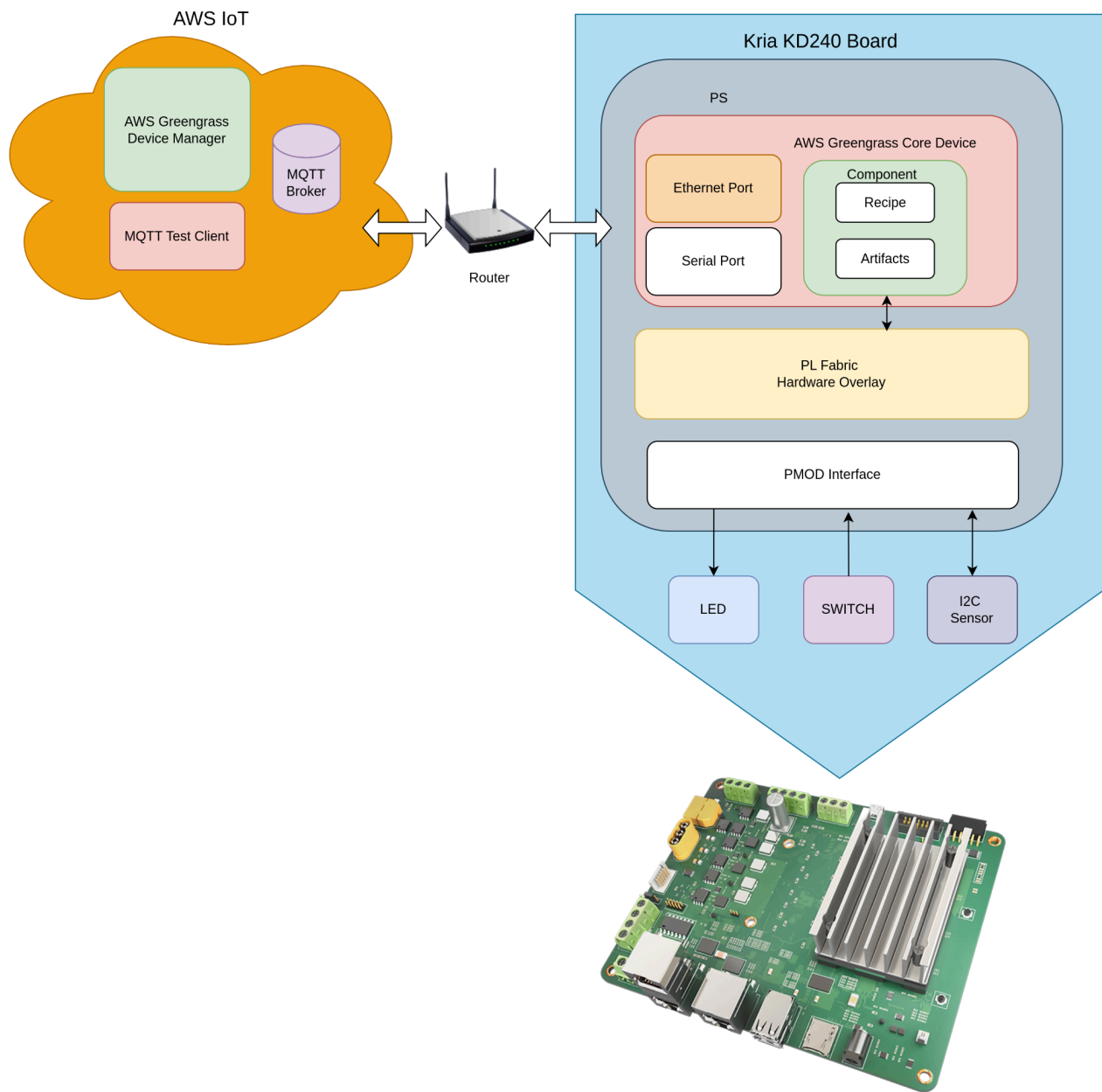
1.2 Applicable operating systems for this guide

For this guide host machine with any (Windows, Linux) OS can be used and for Edge Device Ubuntu 22.04 will be installed.

2. Overview

AMD KD240, Kria Drives Starter Kit is targeted for Industrial Motor Control and Drive applications. We can use industrial sensors to monitor, analyze and plan the industrial application in cloud platform. This board is also targeted for drive control and predictive maintenance applications.

This diagram shows the software and hardware architecture used in this tutorial. Kria KD240 board consists of Programmable Logic (PL) Fabric (FPGA) hardware overlay for interfacing LED, switch and I2C sensor. Further it runs AWS Greengrass Core Device Application which publish and subscribe message topics for actuating LED and monitoring sensors and switches. From AWS IoT MQTT Test Client KD240 LED will be controlled through subscribed topic and also publish Switch pressed event to AWS IoT cloud.



3. Hardware description

3.1 Datasheet

Refer to the Kria K24 SOM Product Brief and SOM Data Sheet at product documentation at the [K24 Documentation page](#).

For a view of the hardware portfolio, refer to this [Portfolio Comparison](#).

3.2 Standard kit contents

Details regarding the KD240 kit contents are provided [here](#).

Refer to the section What's Inside the Box in the online documentation [here](#).

3.3 User provided items

- Host PC

3.4 3rd party purchasable items

For “Hello world” component test in section 9.1, no additional hardware is needed.

While user has to procure following hardware for “MQTT-GPIO” component test in section 9.2:

- Generic Light Emitting Diode(LED)
- Push button switch
- Resistors (4.7 K and 100 ohms)
- Bread Board and connecting jumper wires

Above hardware is readily available at electronic hardware shops like [Amazon](#).

3.5 Additional References

Getting started guide for KD240 board is available [here](#).

Refer to the all the resources related to the Kria™ KD240 Drives Starter Kit on the [GitHub page](#)

Read the [Kria KD240 Drives Starter Kit User Guide](#) and [Data sheet](#)

4. Set up your development environment

4.1 Tools installation (IDEs, Toolchains, SDKs)

Not Applicable

5. Set up device hardware

The KD240 board boots off an SD card. To create this SD card, refer to the instructions at [Setting up the SD card image](#).

The instructions to set up and connect the board are available [here](#).

Follow the instructions [here](#) to boot and monitor your board.

5.1 Installing hardware overlay in KD240

Get the KD240 firmware folder. It contains:

- kd240-gpio-i2c.bit.bin
- kd240-gpio-i2c.dtbo
- shell.json

Copy these file to the KD240 board. For firmware to be loaded using xmutil (FPGA manager), one has to copy these file at “/lib/firmware/xilinx”.

For this create the folder at “kd240-gpio-i2c” at “/lib/firmware/xilinx” and copy the files in “kd240-gpio-i2c” folder.

```
cd /lib/firmware/xilinx
sudo mkdir kd240-gpio-i2c
Cd kd240-gpio-i2c
sudo cp <kd240-firmware directory>/kd240-gpio-i2c* ./
sudo cp <kd240-firmware directory>/shell.json ./
```

Next, check the available fpga firmware using `xmutil listapps` command. `kd240-gpio-i2c` will be available in the list.

```
ubuntu@kria:~$ sudo xmutil listapps
Accelerator      Accel_type      Base      Base_type      #slots(PL+AIE)      Active_slot
k24-starter-kits XRT_FLAT        k24-starter-kits XRT_FLAT        (0+0)                0,
kd240-gpio-i2c   XRT_FLAT        kd240-gpio-i2c   XRT_FLAT        (0+0)                -1
ubuntu@kria:~$
```

Next load the `kd240-gpio-i2c` firmware, which contains necessary hardwares (gpio) and interfaces. In our Greengrass Demo we will be using these gpio to trigger the publishing data to AWS Greengrass IoT cloud server and also actuate GPIO on the message received from AWS cloud.

```
sudo xmutil unloadapp
sudo xmutil loadapp kd240-gpio-i2c
```

```
ubuntu@kria:~$ sudo xmutil loadapp kd240-gpio-i2c
[ 827.076900] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/firmware-name
[ 827.087054] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/resets
[ 827.096939] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/afi0
[ 827.106454] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking0
[ 827.116398] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_0
[ 827.126422] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_1
[ 827.136450] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_gpio_0
[ 827.146477] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_iic_0
kd240-gpio-i2c: loaded to slot 0
ubuntu@kria:~$
```

Now to access GPIO in user application, we will be using `gpiod` library.

5.2 Installing gpiod packages

GPIOD packages are required to access the GPIO channels. It also provides python binding for accessing GPIO in python programming. Install the package using apt-get:

```
sudo apt update
sudo apt-get install gpiod python3-libgpiod
```

Now we can check the available gpio using gpiod applications:

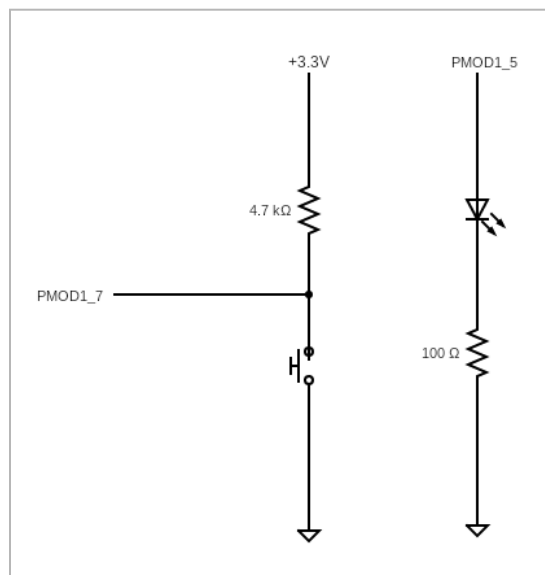
Using `gpiodetect` to get available gpio:

```
ubuntu@kria:~$ sudo gpiodetect
gpiochip0 [firmware:zynqmp-firmware:gpio] (4 lines)
gpiochip1 [zynqmp_gpio] (174 lines)
gpiochip2 [slg7xl45106] (8 lines)
gpiochip3 [800000000.gpio] (4 lines)
ubuntu@kria:~$
```

Here `gpiochip3` is the device corresponding to gpio in FPGA and it consists of 4 lines. Further these gpio lines are connected to PMOD 1 such that:

PMOD1-> 5 - gpiochip3 line 0

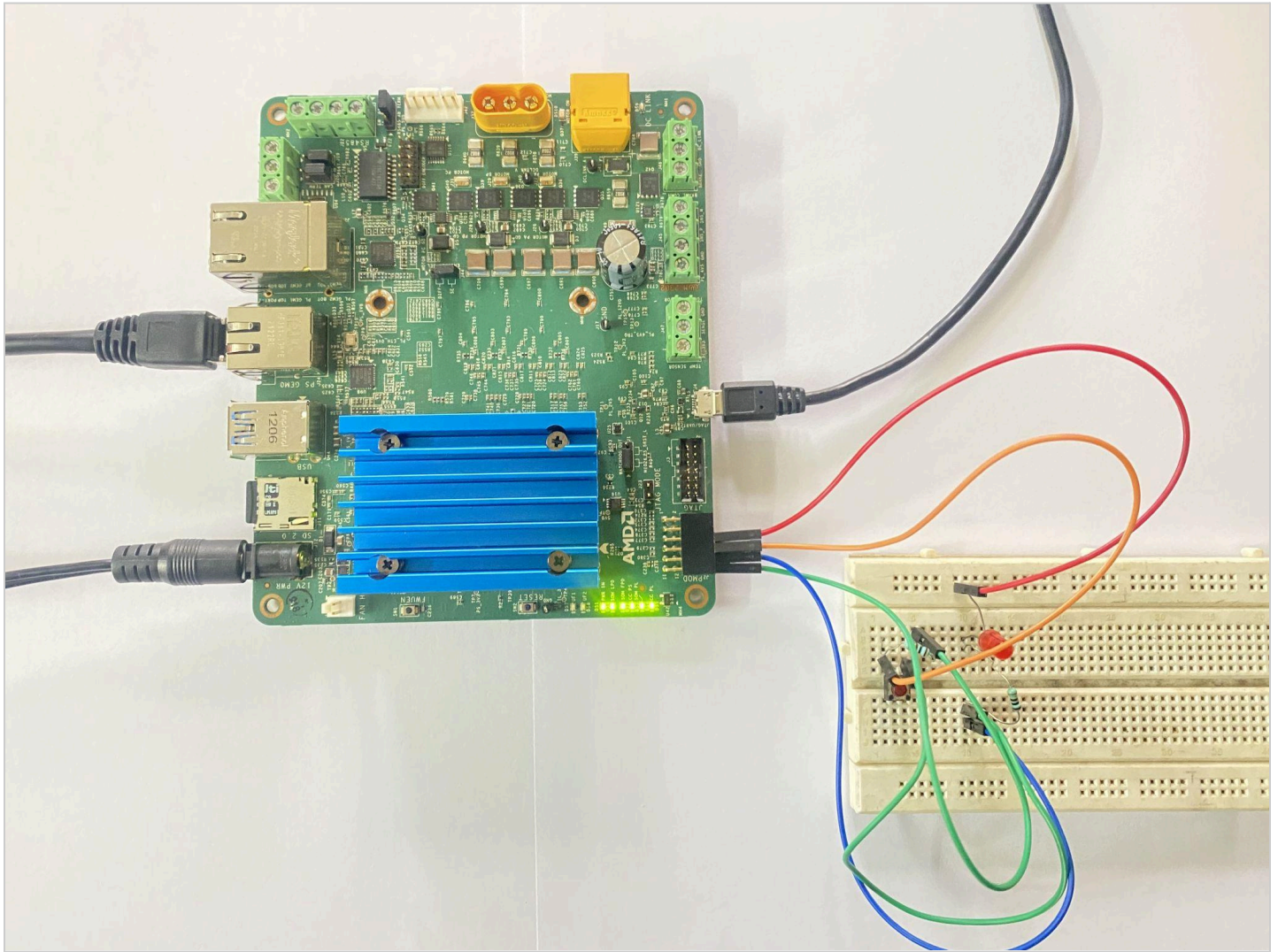
PMOD1-> 7 - gpiochip3 line 1



Schematic for LED and Switch Connection

11	9	7	5	3	1	PMOD UPPER
12	10	8	6	4	2	PMOD LOWER
V _{CC}	GND	I/O	I/O	I/O	I/O	

PMOD port numbering



KD240 Board Connected to switch and LED through PMOD

6. About AWS IoT Greengrass

To learn more about AWS IoT GreengrassV2, see [How AWS IoT Greengrass works](#) and [What's new in AWS IoT Greengrass Version 2](#).

7. Greengrass prerequisites

Refer to the online documentation detailing the [prerequisites](#) needed for AWS IoT Greengrass. Follow the instructions in the following sections:

- [Step 1: Set up an AWS account](#)
- [Step 2: Set up your environment](#)

8. Install AWS IoT Greengrass

Refer to the instructions in the following steps:

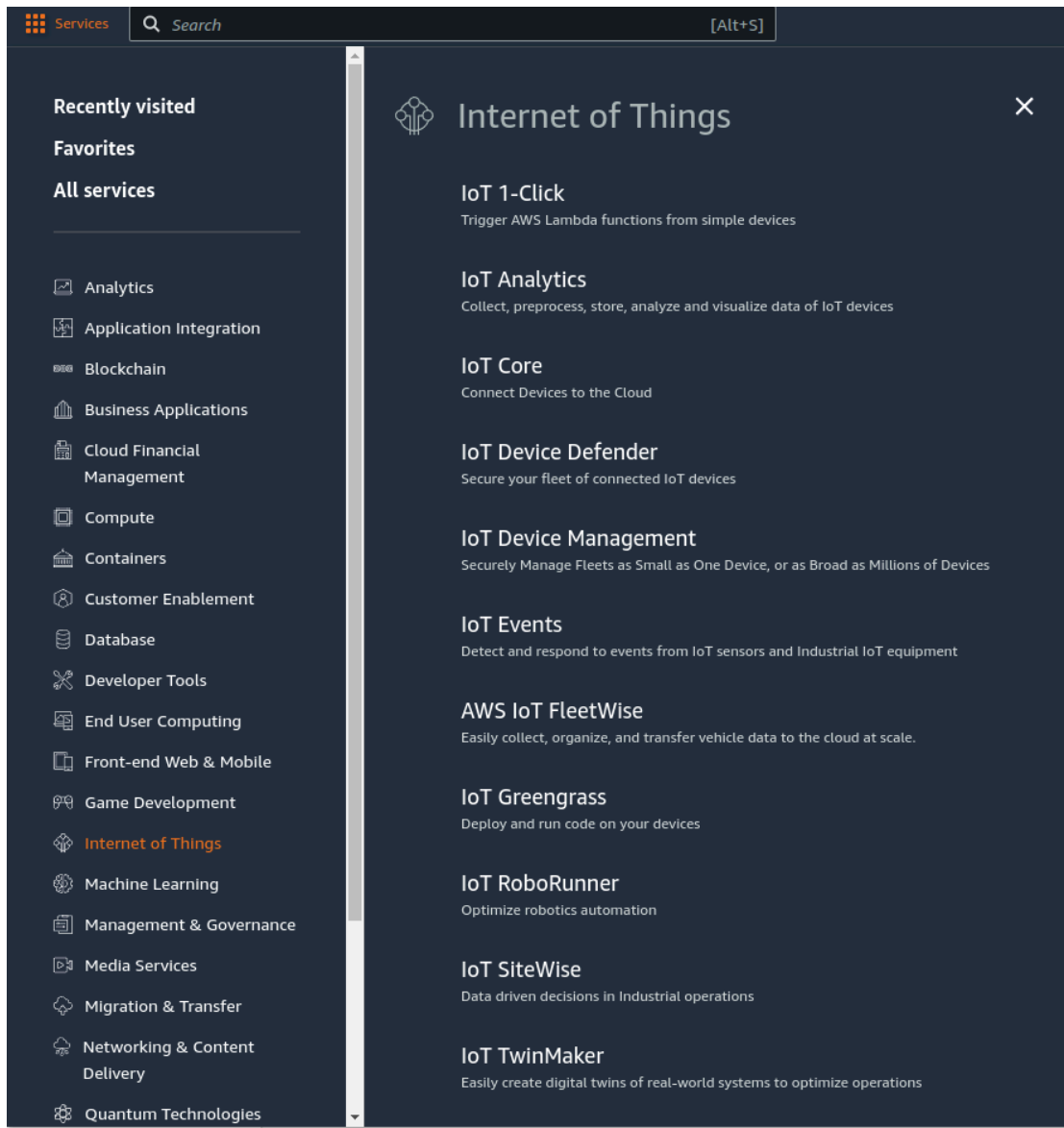
- [Set up the device environment](#)
- [Provide AWS credentials to the device](#). For development environments, you can use the option “Use long-term credentials from an IAM User”. An example of how to do this is shown below:

```
export AWS_ACCESS_KEY_ID=<the access key id for your user>
export AWS_SECRET_ACCESS_KEY=<the secret access key for your user>
```

- [Download the AWS IoT Greengrass Core software](#)
- [Install the AWS IoT Greengrass Core software \(console\)](#)
- [Install the AWS IoT Greengrass Core software](#)

8.1 Steps detail for Installing AWS IoT Greengrass on KD240

Steps and scripts for installing greengrass device is provided by AWS Greengrass dashboard in AWS web console. So first access the AWS Greengrass IoT page, go to AWS Services -> Internet of Things -> IoT Greengrass link



Get started with Greengrass

Set up one core device within minutes.

[Set up one core device](#)

Manage the components that deploy to your core devices.

[View deployments](#)

Pricing (US)

10,000 devices/month	\$0.16 per device
>10,000 devices/month	Contact us

[Learn more](#)

More resources

- [Documentation](#)
- [FAQ](#)
- [Greengrass forum](#)
- [Contact us](#)

How it works

Key terminology in Greengrass

Now click on “Set up one core device” button

This will open the Greengrass core device setup page:

Here you change the Core device name like `kd240-ubuntu-dev1`

And set Group for KD240 group device.

Set up one Greengrass core device

Step 1: Register a Greengrass core device

Greengrass core devices are AWS IoT things. Enter a thing name to be used to create a Greengrass core device.

Core device name

The name of the AWS IoT thing to create. We generated the following name for you.

kd240-ubuntu-dev1

The name can be up to 128 characters. Valid characters: a-z, A-Z, 0-9, underscore (_), and hyphen (-).

Step 2: Add to a thing group to apply a continuous deployment

Add your Greengrass core device to an AWS IoT thing group. If the thing group has an active Greengrass deployment, your new core device receives and applies the deployment when you finish the setup process. To deploy to only the core device, select No group.

Thing group

- ☒ Enter a new group name
- ☐ Select an existing group
- ☐ No group

Thing group name

The name of the AWS IoT thing group to create.

KD240UbuntuGroup

The name can be up to 128 characters. Valid characters: a-z, A-Z, 0-9, underscore (_), and hyphen (-).

Step 3: Install the Greengrass Core software

Operating System

☒ Linux

☐ Windows

Step 3.1: Install Java on the device

The AWS IoT Greengrass Core software runs on Java. Follow instructions to install the Java runtime on the device. [Learn](#)

Now in KD240 terminal console run following commands and scripts:

```
export AWS_ACCESS_KEY_ID=<AWS_ACCESS_KEY_ID>
export AWS_SECRET_ACCESS_KEY=<AWS_SECRET_ACCESS_KEY>
```

Greengrass CLI depends on Java. So to install the dependency run the following:

```
sudo apt install default-jre
sudo apt install default-jdk
```


```
sudo apt install unzip
```

Download and install Greengrass core software as instructed in AWS setup step 3.

```
curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip > greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d GreengrassInstaller
```

Step 3.3: Run the installer

AWS IoT Greengrass provides an installer that you can use to set up a Greengrass core device in a few minutes. The installer runs on the device and does the following:

1. Provisions the Greengrass core device as an AWS IoT thing with a device certificate and default permissions. [Learn more](#) 
2. Creates a system user and group, `ggc_user` and `ggc_group`, that the software uses to run components on the device.
3. Connects the device to AWS IoT.
4. Installs and runs the latest AWS IoT Greengrass Core software as a system service.

Download the installer

Run the following command on the device to download the AWS IoT Greengrass Core software.

```
curl -s https://d2s8p88vqu9w66.cloudfront.net/releases/greengrass-nucleus-latest.zip > greengrass-nucleus-latest.zip && unzip greengrass-nucleus-latest.zip -d GreengrassInstaller
```

✔ Command copied

 Copy

Next install the Greengrass core device as instructed in AWS setup step 3.:

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE -jar ./GreengrassInstaller/lib/Greengrass.jar --aws-region us-east-1 --thing-name kd240-ubuntu-dev1 --thing-group-name KD240UbuntuGroup --component-default-user ggc_user:ggc_group --provision true --setup-system-service true --deploy-dev-tools true
```

Run the installer

The AWS IoT Greengrass Core software is a JAR file that installs the software when you run it for the first time. Run the following command on the device.

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE -jar ./GreengrassInstaller/lib/Greengrass.jar --aws-region us-east-1 --thing-name kd240-ubuntu-dev1 --thing-group-name KD240UbuntuGroup --component-default-user ggc_user:ggc_group --provision true --setup-system-service true --deploy-dev-tools true
```

✔ Command copied

 Copy

Here is the console log after running above command:

```

Creating group ggc_group
ggc_group created
Added ggc_user to ggc_group
Provisioning AWS IoT Resources for the device with IoT Thing Name: [kd240-ubuntu-dev1]...
Found IoT policy "GreengrassV2IoTThingPolicy", reusing it
Creating keys and certificate...
Attaching policy to certificate...
Creating IoT Thing "kd240-ubuntu-dev1"...
Attaching certificate to IoT thing...
Successfully provisioned AWS IoT resources for the device with IoT Thing Name: [kd240-ubuntu-dev1]!
Adding IoT Thing [kd240-ubuntu-dev1] into Thing Group: [KD240UbuntuGroup]...
Successfully added Thing into Thing Group: [KD240UbuntuGroup]
Setting up resources for aws.greengrass.TokenExchangeService ...
Attaching TES role policy to IoT thing...
No managed IAM policy found, looking for user defined policy...
IAM policy named "GreengrassV2TokenExchangeRoleAccess" already exists. Please attach it to the IAM role if not already
Configuring Nucleus with provisioned resource details...
Downloading Root CA from "https://www.amazontrust.com/repository/AmazonRootCA1.pem"
Created device configuration
Successfully configured Nucleus with provisioned resource details!
Creating a deployment for Greengrass first party components to the thing group
Configured Nucleus to deploy aws.greengrass.Cli component
Successfully set up Nucleus as a system service
ubuntu@kria:~$ █

```

Now in Greengrass set up page, one can view the Greengrass core devices and find above `kd240-ubuntu-dev` in the list.

[AWS IoT](#) > [Greengrass](#) > Core devices

Greengrass core devices [Info](#)

Greengrass core devices (5)

Name	Status	Status reported
kr260-dev1	✓ Healthy	12 days ago
kr260-peta-dev1	✓ Healthy	12 days ago
kd240-ubuntu-dev1	✓ Healthy	1 minute ago
kr260-ubuntu-dev1	✓ Healthy	9 days ago
kd240-dev2	✗ Unhealthy	22 hours ago

In KD240 terminal one can get the device components by using `greengrass-cli`:

```
sudo /greengrass/v2/bin/greengrass-cli component list
```

```
ubuntu@kria:~$ sudo /greengrass/v2/bin/greengrass-cli component list
Components currently running in Greengrass:
Component Name: aws.greengrass.Nucleus
  Version: 2.12.1
  State: FINISHED
  Configuration: {"awsRegion":"us-east-1","componentStoreMaxSizeBytes":"100000000000","deploymentF
t":"","greengrassDataPlanePort":"8443","httpClient":{"},"iotCredEndpoint":"cluwyavs4wpvzg.credentials
ions":"-Dlog.store=FILE","logging":{"},"mqtt":{"spooler":{"}},"networkProxy":{"proxy":{"}},"platform0v
Component Name: DeploymentService
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: UpdateSystemPolicyService
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: FleetStatusService
  Version: null
  State: RUNNING
  Configuration: null
Component Name: TelemetryAgent
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: aws.greengrass.Cli
  Version: 2.12.1
  State: RUNNING
  Configuration: {"AuthorizedPosixGroups":null,"AuthorizedWindowsGroups":null}
ubuntu@kria:~$ █
```

Next we will be adding component to publish and subscribe the topic to the AWS cloud Broker.

9. Installing the component

Before installing the “mqtt-gpio” component, get familiar with the dummy “Hello World” component.

9.1 Create a “Hello World” component

9.1.1 Create the component on your edge device

Follow the instructions online under the section [Develop and test a component on your device](#) to create a simple component on your device.

9.1.2 Upload the “Hello World” component

Follow the instructions online at [Create your component in the AWS IoT Greengrass service](#) to upload your component to the cloud, where it can be deployed to other devices as needed.

9.1.3 Deploy your component

Follow the instructions online at [Deploy your component](#) to deploy and verify that your component is running.

9.2 Installing “mqtt-gpio” component

9.2.1 Get the component

Get the `components` folder and copy in the KD240 home directory.

It contains:

Artifacts

- com.example.mqtt
 - 1.0.0
 - mqtt.py (This python code published the data on button press and actuates gpio on receiving the data in subscribed topic)

Recipe

- com.example.mqtt-1.0.0.json

9.2.2 Deploy the component

To install the above component run the following in the KD240 terminal:

```
sudo /greengrass/v2/bin/greengrass-cli deployment create \  
--recipeDir ~/components/recipe \  
--artifactDir ~/components/artifacts \  
--merge "com.example.mqtt=1.0.0"
```

```
ubuntu@kria:~$ sudo /greengrass/v2/bin/greengrass-cli deployment create \  
--recipeDir ~/components/recipe \  
--artifactDir ~/components/artifacts \  
--merge "com.example.mqtt=1.0.0"  
  
Local deployment submitted! Deployment Id: 3e4cad17-9a79-4e76-bc20-58a3b5b0093f  
ubuntu@kria:~$ █
```

9.2.3 Upload the component

Follow the instructions online at [Create your component in the AWS IoT Greengrass service](#) to upload your component to the cloud, where it can be deployed to other devices as needed.

10. Debugging

Now check the installed component is in “running state”:

```
ubuntu@kria:~$ sudo /greengrass/v2/bin/greengrass-cli deployment create \
--recipeDir ~/components/recipe \
--artifactDir ~/components/artifacts \
--merge "com.example.mqtt=1.0.0"
Local deployment submitted! Deployment Id: 4835f786-9e25-4250-8b1c-c3acd3bc3de9
ubuntu@kria:~$ sudo /greengrass/v2/bin/greengrass-cli component list
Components currently running in Greengrass:
Component Name: aws.greengrass.Nucleus
  Version: 2.12.1
  State: FINISHED
  Configuration: {"awsRegion":"us-east-1","componentStoreMaxSizeBytes":"10000000000","deploymentPollingInterval":10,"greengrassDataPlanePort":"8443","httpClient":{"url":"https://greengrass.amazonaws.com/resources","method":"GET"},"iotCredEndpoint":"cluwyavs4wpvxg.credentials.io","logging":{"level":"INFO"},"mqtt":{"spooler":{"enabled":true},"topic":"greengrass"},"networkProxy":{"proxy":{"url":"https://greengrass.amazonaws.com/resources","method":"GET"},"enabled":true},"platformOverride":{"url":"https://greengrass.amazonaws.com/resources","method":"GET"},"enabled":true}}
Component Name: UpdateSystemPolicyService
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: DeploymentService
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: TelemetryAgent
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: FleetStatusService
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: com.example.mqtt
  Version: 1.0.0
  State: BROKEN
  Configuration: {"accessControl":{"aws.greengrass.ipc.mqttproxy":{"com.example.mqtt:mqttproxy:1":{"operations":["kd240/mqtt","kd240/button"]}}},"message":"hello"}
Component Name: aws.greengrass.Cli
  Version: 2.12.1
  State: RUNNING
  Configuration: {"AuthorizedPosixGroups":null,"AuthorizedWindowsGroups":null}
ubuntu@kria:~$
```

Now in aws IoT console, open “MQTT test client” and subscribe to “#”

Subscribe to a topic

Publish to a topic

Topic filter [Info](#)

The topic filter describes the topic(s) to which you want to subscribe. The topic filter can include MQTT wildcard characters.

#

▶ Additional configuration

Subscribe

Subscriptions

#

Pause Clear Export Edit

kd240/mqtt

#

You cannot publish messages to a wildcard topic.
Please select a different topic to publish messages to.

▼ kd240/button

December 27, 2023, 14:04:43 (UTC+0545)

```
{
  "button": "button pressed",
  "timemillis": 1703665181137
}
```

▶ Properties

You can see the “button pressed” message once the button is pressed.

Now to control the LED, publish the message to “kd240/mqtt” topic. Here is the screenshot of the message which switch on the LED.

Subscribe to a topic

Publish to a topic

Topic name

The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

Q kd240/mqtt

Message payload

```
{
  "ledon": true
}
```

▶ Additional configuration

Publish

Subscriptions

#

Pause Clear Export Edit

#

You cannot publish messages to a wildcard topic.
Please select a different topic to publish messages to.

▼ kd240/mqtt

December 27, 2023, 14:06:22 (UTC+0545)

```
{
  "ledon": true
}
```

Now to switch off the LED send “false” message in the “kd240/mqtt” topic.

Subscribe to a topic

Publish to a topic

Topic name

The topic name identifies the message. The message payload will be published to this topic with a Quality of Service (QoS) of 0.

Q kd240/mqtt X

Message payload

{
 "ledon": false
}

► Additional configuration

Publish

Subscriptions

#

Pause Clear Export Edit

#

♡ X

You cannot publish messages to a wildcard topic.
Please select a different topic to publish messages to.

▼ kd240/mqtt

December 27, 2023, 14:06:59 (UTC+0545)

{
 "ledon": false
}

► Properties

11. Troubleshooting

For more information, refer to the online documentation [Troubleshooting Greengrass v2](#).

You can also refer to [Logging and Monitoring](#) to learn how to log API calls, gather system health telemetry data, and check core device status.

If you need support on Kria SoM with Greengrass, then you can go for [AMD Kria SoMs forum](#).