

AMD KV260 Getting Started Guide

for AWS IoT Greengrass V2

Table of Contents

<i>1. Document information</i>	1
<i>2. Overview</i>	2
<i>3. Hardware description</i>	3
<i>4. Set up your development environment</i>	3
<i>5. Set up device hardware</i>	3
<i>6. About AWS IoT Greengrass</i>	6
<i>7. Greengrass prerequisites</i>	6
<i>8. Install AWS IoT Greengrass</i>	7
<i>9. Installing the components</i>	13
<i>10. Debugging</i>	15
<i>11. Troubleshooting</i>	17

1. Document information

Note that all instructions in this document have been written assuming a KV260 Ubuntu Edge device.

1.1 Document revision history

30-Dec-2023	Initial Draft
27-Feb-2024	Updated Document Format

1.2 Applicable operating systems for this guide

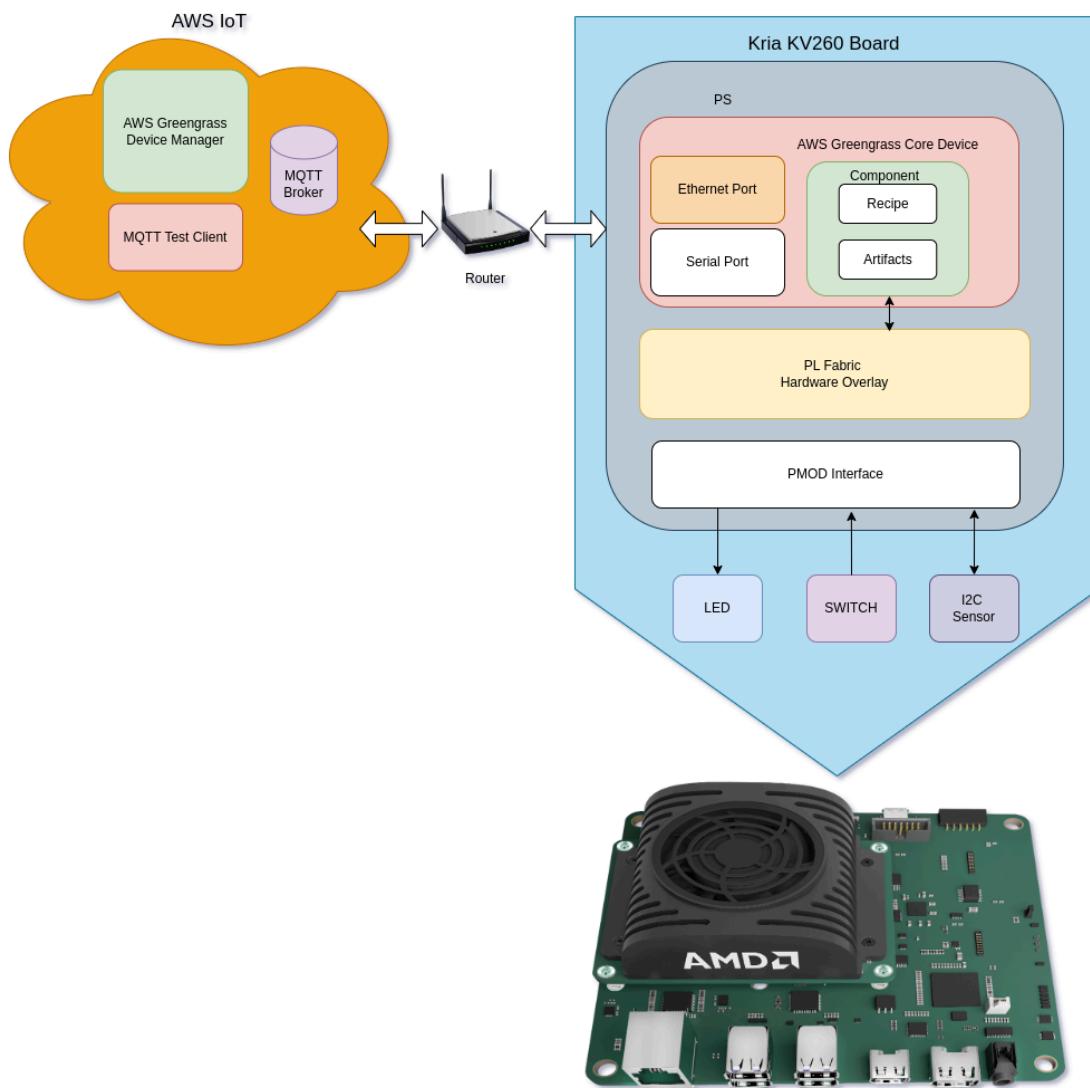
For this guide host machine with any (Windows, Linux) OS can be used and for Edge Device Ubuntu 22.04 will be installed.

2. Overview

AMD Kria SoM is Zynq® UltraScale+™ MPSoC architecture based FPGA platform, which is suitable for vision AI solutions for edge applications.

AMD KV260, Kria Vision AI Starter Kit is targeted for Vision Edge AI applications. By using different sensors and camera we can create Vision-IoT systems, which can be targeted for Smart cities, Industrial applications and Agricultural applications. We can perform machine learning inference in KV260 itself and send informations to cloud platform for further analysis and planning.

Following diagram shows the software and hardware architecture used in this tutorial. Kria KV260 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 KV260 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 K26 SOM Product Brief and SOM Data Sheet at product documentation at the [K26 Documentation page](#).

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

3.2 Standard kit contents

Details regarding the KV260 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 KV260 board is available [here](#).

Refer to the all the resources related to the Kria™ KV260 Vision AI Starter Kit on the [GitHub page](#)

Read the [Kria KV260 Robotics 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 KV260 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

Get the KV260 firmware folder. It contains:

- kv260-gpio-i2c.bit.bin
- kv260-gpio-i2ci2c.dtbo
- shell.json

Copy these file to the KV260 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 “kv260-gpio-i2c” at “/lib/firmware/xilinx” and copy the files in “kv260-gpio-i2c” folder.

```
cd /lib/firmware/xilinx
sudo mkdir kv260-gpio-i2c
sudo cp <kv260-firmware directory>/kv260-gpio-i2c* ./
sudo cp <kv260-firmware directory>/shell.json ./
```

Next, check the available fpga firmware using `xmutil listapps` command. `kv260-gpio-i2c` will be available in the list.

Accelerator	Accel_type	Base	Base_type	#slots(PL+AIE)	Active_slot
k26-starter-kits kv260-gpio-i2c	XRT_FLAT XRT_FLAT	k26-starter-kits kv260-gpio-i2c	XRT_FLAT XRT_FLAT	(0+0) (0+0)	0, -1

Next load the `kv260-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 kv260-gpio-i2c
```

```
ubuntu@kria:~$ sudo xmutil loadapp kv260-gpio-i2c
[ 1027.134932] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/firmware-name
[ 1027.145049] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/resets
[ 1027.154904] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/afio0
[ 1027.164406] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking0
[ 1027.174332] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking1
[ 1027.184267] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_0
[ 1027.194279] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_1
[ 1027.204304] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_gpio_0
[ 1027.214321] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_iic_0
kv260-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:

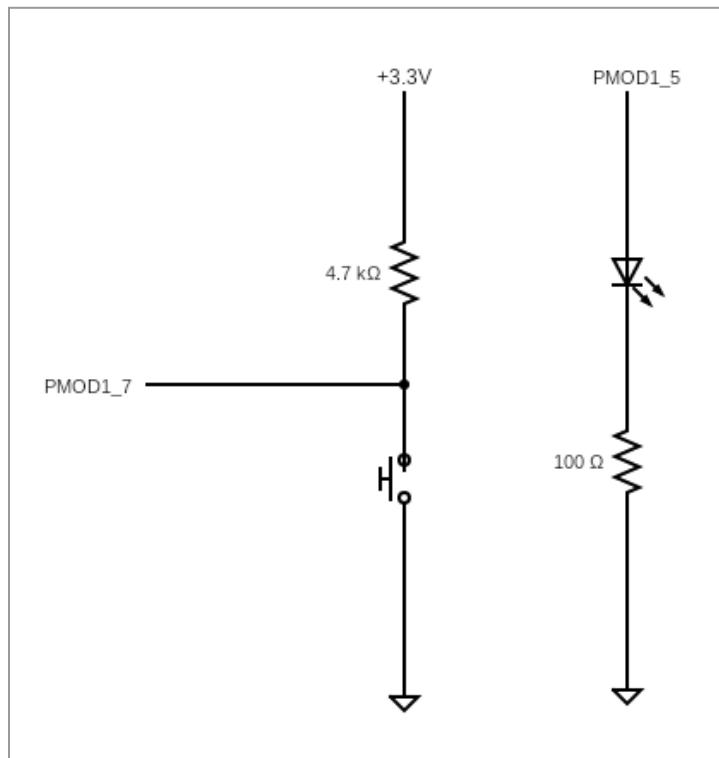
Using `gpiodetect` to get available gpio:

```
ubuntu@kria:~$ sudo gpiodetect
gpiochip0 [firmware:zynqmp-firmware:gpio] (4 lines)
gpiochip1 [zynqmp_gpio] (174 lines)
gpiochip2 [80020000 gpio] (4 lines)
ubuntu@kria:~$ █
```

Here `gpiochip2` 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 - gpiochip2 line 0

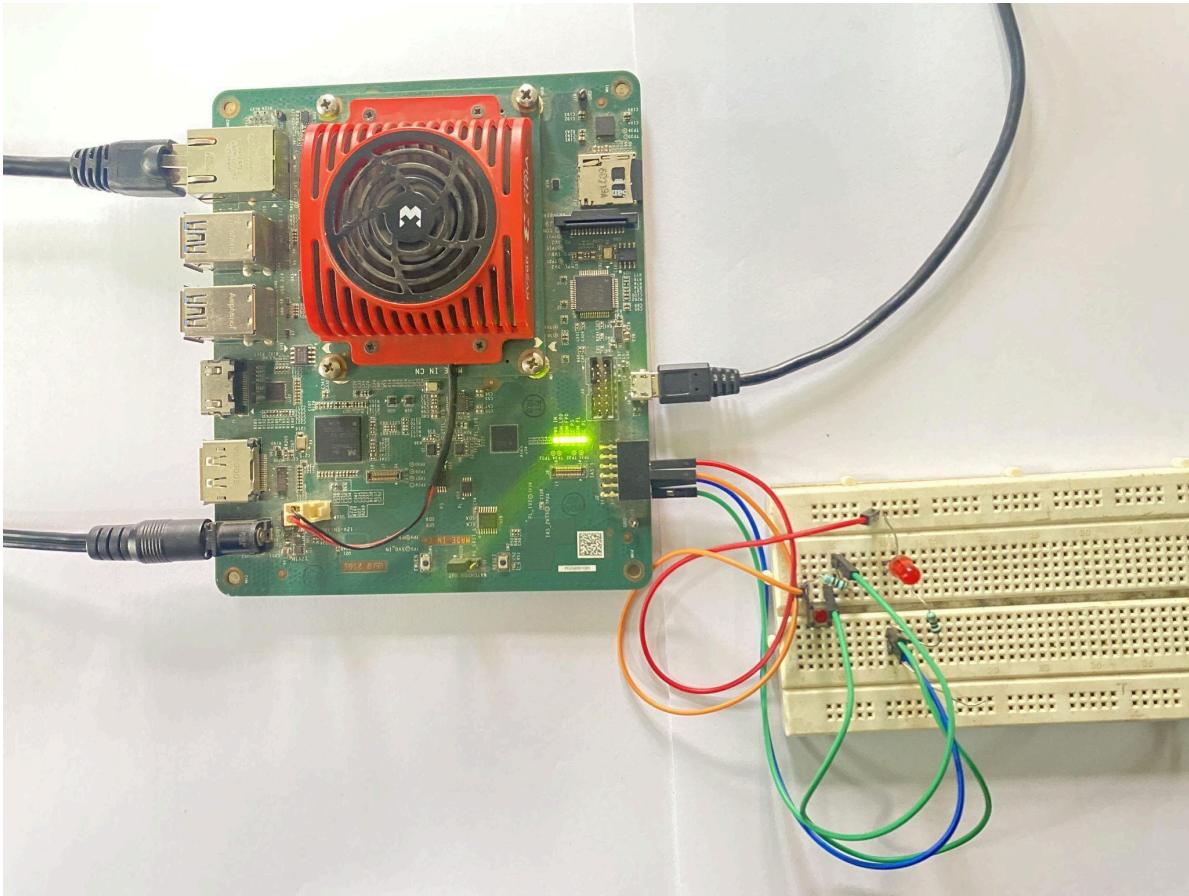
PMOD1-> 7 - gpiochip2 line 1



Schematic for LED and Switch Connection

11	9	7	5	3	1	PMOD UPPER
12	10	8	6	4	2	PMOD LOWER
Vcc	GND	I/O	I/O	I/O	I/O	

PMOD port numbering



KV260 Board Connected to switch and LED through PMOD

6. About AWS IoT Greengrass

To learn more about AWS IoT Greengrass V2, 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:

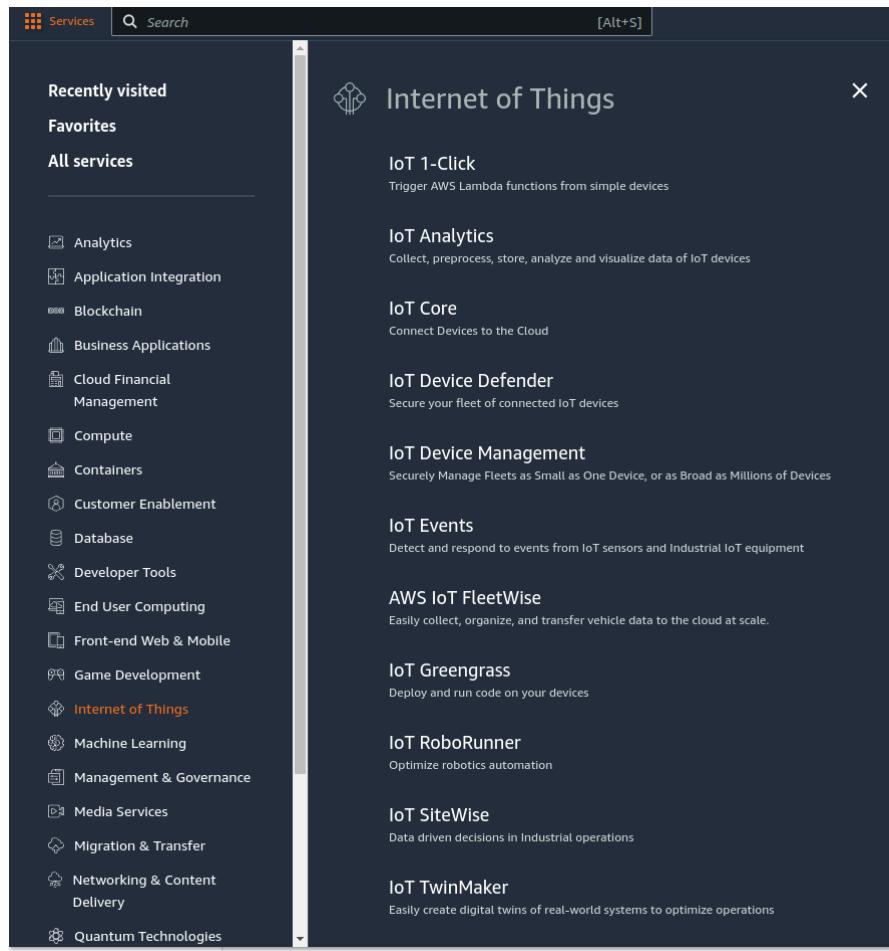
- [Set up the device environment](#)
- [Provide AWS credentials to the device](#). For development environments, you can use the option “Use long-term credentials from an IAM User”. An example of how to do this is shown below:

```
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 KV260

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



The screenshot shows the AWS IoT Greengrass landing page. The left sidebar contains navigation links for Monitor, Connect (with options for one or many devices), Test (Device Advisor, MQTT test client), and Manage (Greengrass devices, LPWAN devices, Software packages, Remote actions, Message routing, Retained messages, Security, Fleet Hub). The main content area features a large title "AWS IoT Greengrass" and subtitle "Build, deploy, and manage device software at-scale". It includes a callout "Bring local compute, messaging, data management, sync, and ML inference capabilities to edge devices." Below this is a diagram titled "How it works" showing the flow from AWS Cloud services like Lambda, API Gateway, and AWS IoT Core to a Greengrass core device, which then connects to an IoT Thing group. The right sidebar has sections for "Get started with Greengrass" (button "Set up one core device"), "View deployments", "Pricing (US)" (plans for 10,000 and >10,000 devices/month), "More resources" (links to Documentation, FAQ, Greengrass forum, and Contact us).

AWS IoT

Monitor

Connect

- Connect one device
- ▶ Connect many devices

Test

- ▶ Device Advisor
- MQTT test client
- Device Location New

Manage

- ▶ Greengrass devices
 - Core devices
 - Components
 - Deployments
 - Groups (V1)
- ▶ LPWAN devices
- Software packages New
- ▶ Remote actions
- ▶ Message routing
- Retained messages
- ▶ Security
- ▶ Fleet Hub

Internet of Things

AWS IoT Greengrass

Build, deploy, and manage device software at-scale

Bring local compute, messaging, data management, sync, and ML inference capabilities to edge devices.

How it works

Key terminology in Greengrass

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

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 `kv260-ubuntu-dev1`

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.

kv260-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.

kv260UbuntuGroup

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

Now in KV260 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
```

Download and install Greengrass core software as instructed in

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

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
```

 Copy

Next install the Greengrass core device:

```
sudo -E java -Droot="/greengrass/v2" -Dlog.store=FILE -jar  
./GreengrassInstaller/lib/Greengrass.jar --aws-region us-east-1 --thing-name  
kv260-ubuntu-dev1 --thing-group-name kv260UbuntuGroup --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 kv260-ubuntu-dev1 --thing-group-name kv260UbuntuGroup  
--component-default-user ggc_user:ggc_group --provision true --setup-system-service true --deploy  
-dev-tools true
```

 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: [kv260-ubuntu-dev1]...  
Found IoT policy "GreengrassV2IoTThingPolicy", reusing it  
Creating keys and certificate...  
Attaching policy to certificate...  
Creating IoT Thing "kv260-ubuntu-dev1"..."  
Attaching certificate to IoT thing...  
Successfully provisioned AWS IoT resources for the device with IoT Thing Name: [kv260-ubuntu-dev1]!  
Adding IoT Thing [kv260-ubuntu-dev1] into Thing Group: [kv260UbuntuGroup]...  
Successfully added Thing into Thing Group: [kv260UbuntuGroup]  
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 `kv260-ubuntu-dev` in the list.

Greengrass core devices

Greengrass core devices (7)

 Search by core device name

Name	Status	Status reported
kr260-dev1	 Healthy	13 days ago
kr260-peta-dev1	 Healthy	13 days ago
kd240-ubuntu-dev1	 Healthy	23 hours ago
kv260-peta-dev1	 Healthy	2 hours ago
kr260-ubuntu-dev1	 Healthy	10 days ago
kv260-ubuntu-dev1	 Healthy	7 seconds ago
kd240-dev2	 Unhealthy	2 days ago

In KV260 terminal one can get the device components by using `greengrass-cli`:

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

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

9. Installing the components

Before installing “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 KV260 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 KV260 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: 51255b6b-795d-43f7-a84a-d0b7a3d561fd  
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.

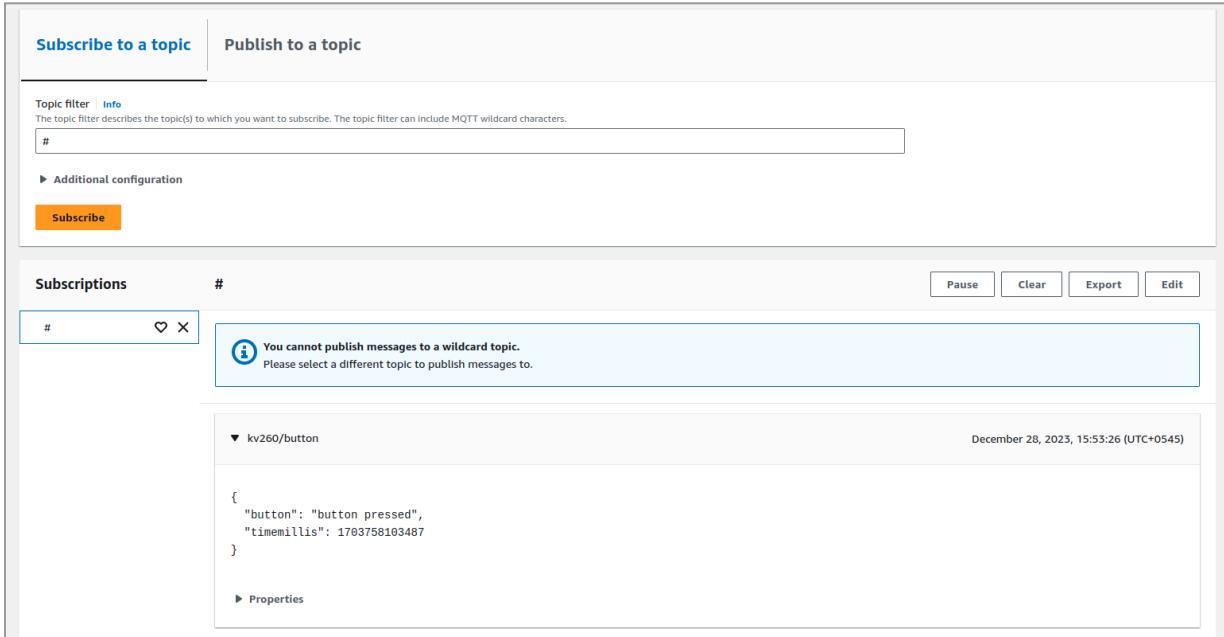
10. Debugging

Now check the installed component is in “running state”

```
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: com.example.mqtt
  Version: 1.0.0
  State: RUNNING
  Configuration: {"accessControl": {"aws.greengrass.ipc.mqttproxy": {"com.example.mqtt:mqttproxy:1": {"operations": ["kv260/mqtt", "kv260/button"]}}}, "message": "hello"}
Component Name: aws.greengrass.Nucleus
  Version: 2.12.1
  State: FINISHED
  Configuration: {"awsRegion": "us-east-1", "componentStoreMaxSizeBytes": "10000000000", "deploymentPollingFrequency": "", "greengrassDataPlanePort": "8443", "httpClient": {}, "iotCredEndpoint": "cluwyav54wpvxg.credentials.iot.us-east-1.amazonaws.com", "logLevel": "INFO", "logPath": "/var/log/greengrass", "logType": "File", "networkProxy": {"proxy": {}}, "platformOverride": {}}
Component Name: UpdateSystemPolicyService
  Version: 0.0.0
  State: RUNNING
  Configuration: null
Component Name: aws.greengrass.Cli
  Version: 2.12.1
  State: RUNNING
  Configuration: {"AuthorizedPosixGroups": null, "AuthorizedWindowsGroups": null}
Component Name: FleetStatusService
  Version: null
  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
ubuntu@kria:~$
```

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



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

Now to control the LED, publish the message to “kv260/mqtt” topic. Here is the screenshot of the message which switch on the LED.

The screenshot shows a publishing interface with the following details:

- Topic name:** kv260/mqtt
- Message payload:** `{ "ledon": true }`
- Publish button:** An orange button labeled "Publish".
- Subscriptions:** A list of topics with a note: "You cannot publish messages to a wildcard topic. Please select a different topic to publish messages to."
- Published message:** A log entry for topic kv260/mqtt at December 28, 2023, 15:54:58 (UTC+0545) with payload `{ "ledon": true }`.

Now to switch off the LED send “false” message in the “kv260/mqtt” topic.

The screenshot shows a publishing interface with the following details:

- Topic name:** kv260/mqtt
- Message payload:** `{ "ledon": false }`
- Publish button:** An orange button labeled "Publish".
- Subscriptions:** A list of topics with a note: "You cannot publish messages to a wildcard topic. Please select a different topic to publish messages to."
- Published message:** A log entry for topic kv260/mqtt at December 28, 2023, 15:55:08 (UTC+0545) with payload `{ "ledon": false }`. Below it, there is a "Properties" section.

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](#).