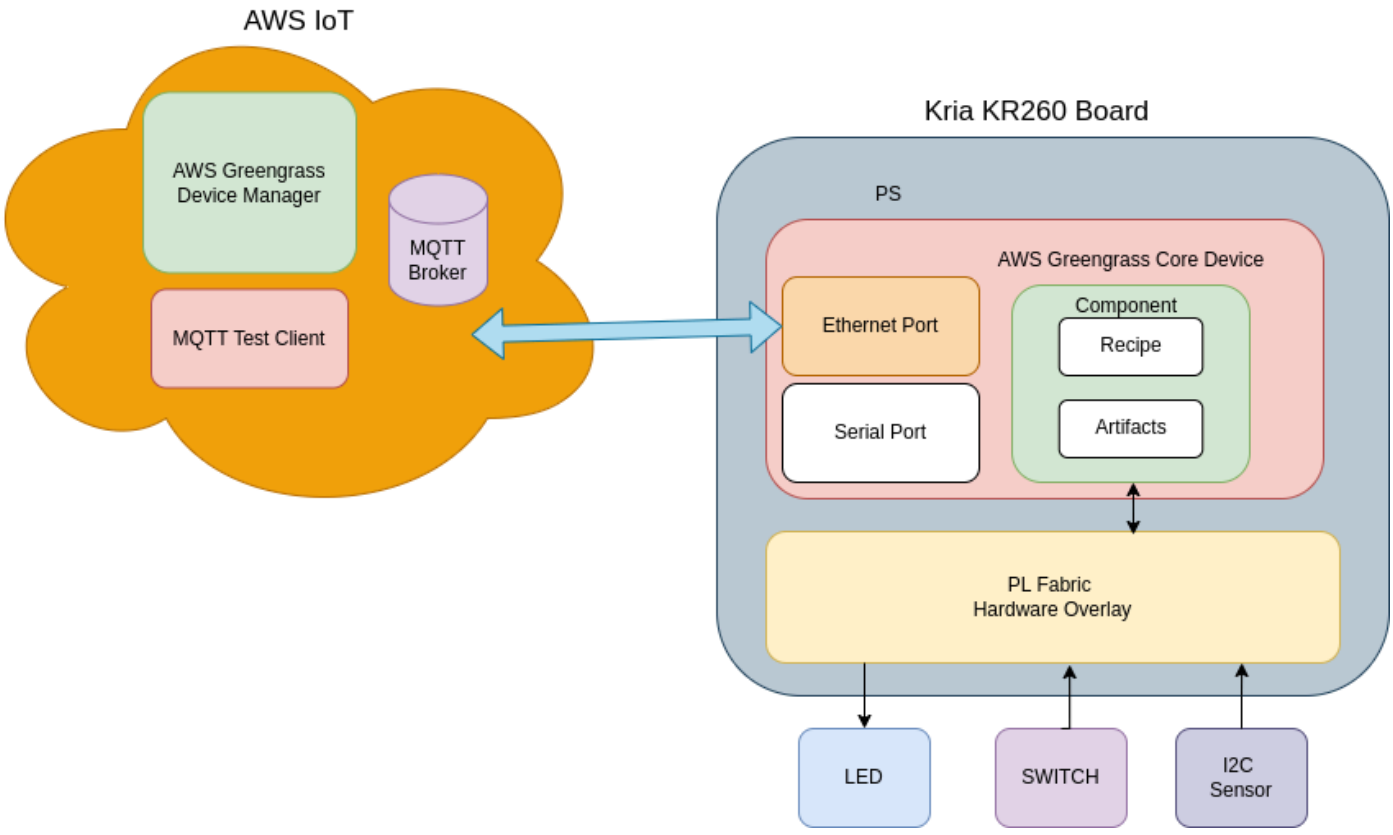
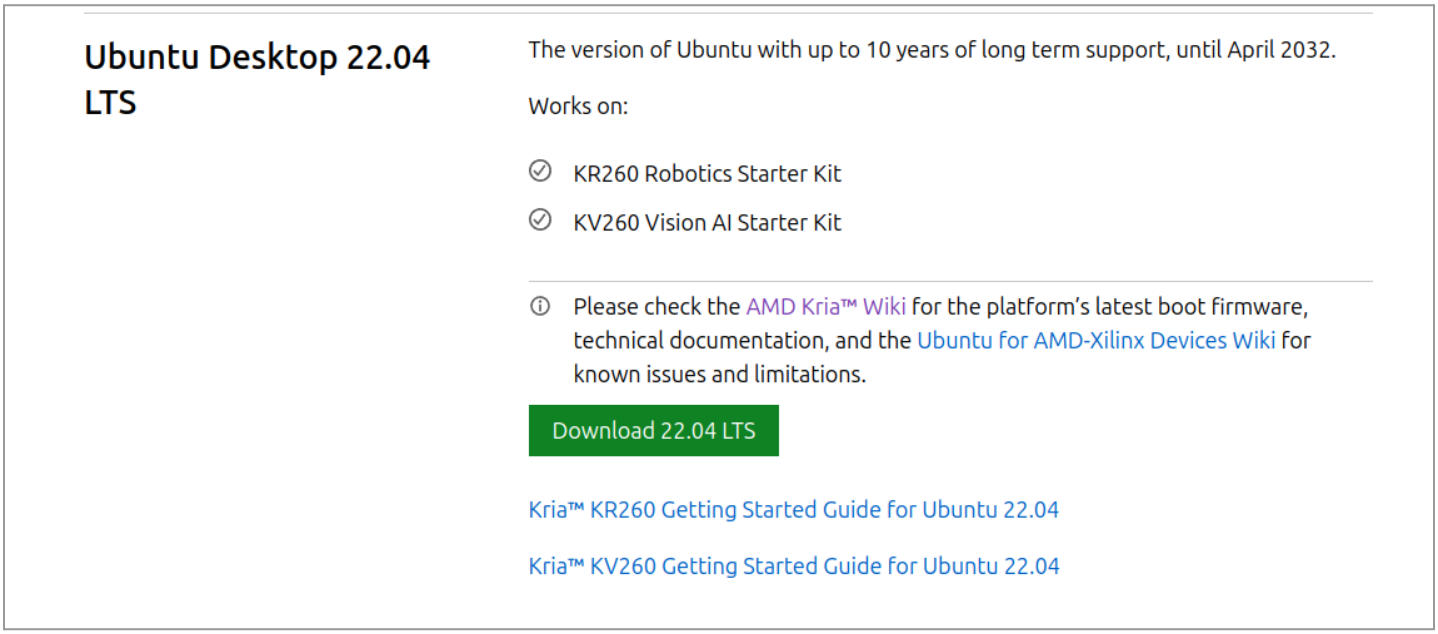
KV260 to AWS IoT Greengrass Architecture



This 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 BMP180 Test Client data collected from BMP180 sensor is published to the AWS IoT.

Preparing Ubuntu 22.04 OS for KRIA KV260 board

Download the Ubuntu 22.04 image from the download link



Next, prepare the SD card with the above downloaded Ubuntu image using burning tools like Balena Etcher.

Now boot the KV260 with the SD card with Ethernet and USB to Serial cable connected to board. We will be using Serial console for initial access and debugging and Ethernet network for accessing through SSH and KV260 connected to the internet.

For initial login here are the Login Details:
Username : ubuntu
Password: ubuntu
This will ask to change the password. So update the password and login the system.
After successful login, one can access the KV260 device console.

Installing hardware overlay

Get the KV260 firmware folder. It contains:

- kv260-gpio-i2c.bit.bin
- kv260-gpio-i2c.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

              kd240-gpio-i2c           XRT_FLAT          kd240-gpio-i2c           XRT_FLAT             (0+0)                 -1
          k24-starter-kits             XRT_FLAT      k24-starter-kits             XRT_FLAT             (0+0)                 0,
```

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
```

```
[  141.337484] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/firmware-name
[  141.347670] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /fpga-full/resets
[  141.357614] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/afi0
[  141.367136] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/clocking0
[  141.377081] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_0
[  141.387107] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_intc_1
[  141.397141] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_gpio_0
[  141.407176] OF: overlay: WARNING: memory leak will occur if overlay removed, property: /__symbols__/axi_iic_0
kd240-gpio-i2c: loaded to slot 0
```

Now, check the available i2c channels available in the system using `i2cdetect` i2c utility tool.

```
sudo i2cdetect -l
```

```
ubuntu@kria:~$ sudo i2cdetect -l
i2c-1   i2c             Cadence I2C at ff030000                 I2C adapter
i2c-2   i2c             ZynqMP DP AUX                           I2C adapter
i2c-3   i2c             xiic-i2c 80000000.i2c                   I2C adapter
ubuntu@kria:~$
```

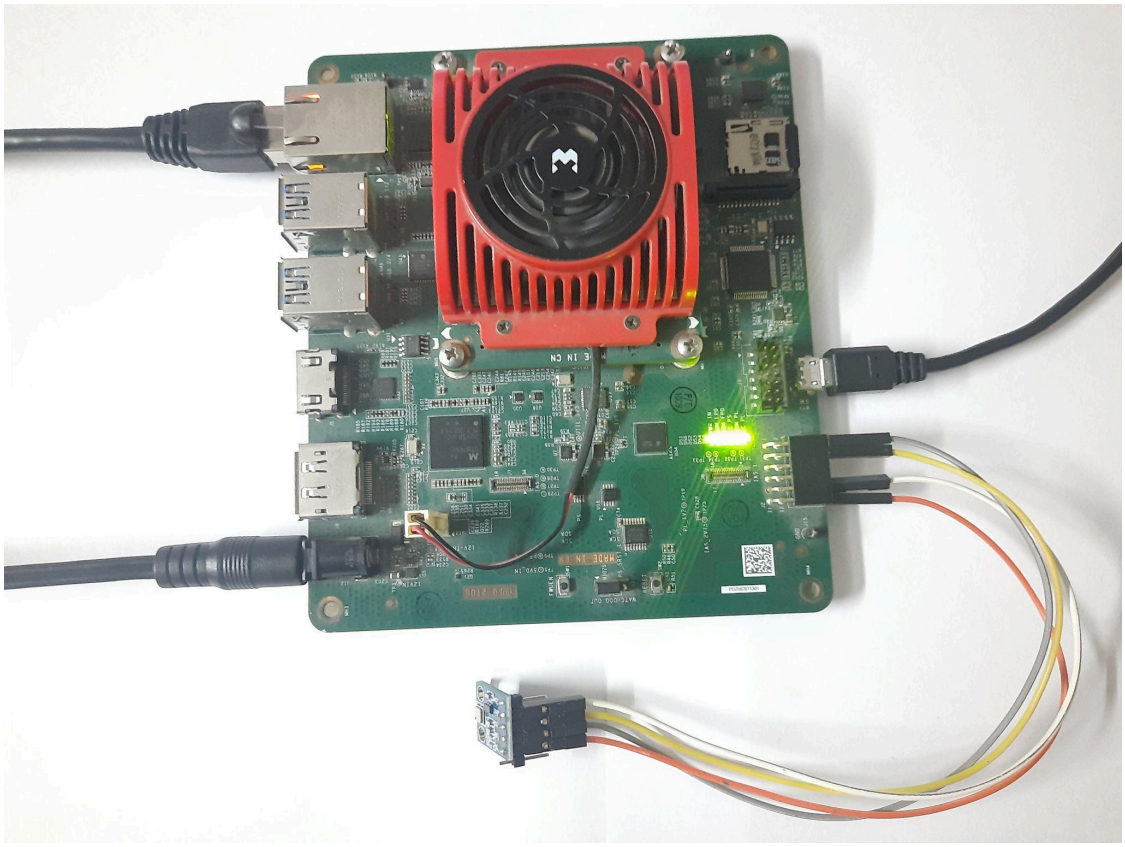
`i2c-3` channel will be used to connect to BMP180 sensor.

Connecting BMP180 to AXI I2C Bus

Connect BMP180 sensors, Vcc, GND, I2C SDA and I2C SCLK pins to PMOD as explained below:
PMOD1-> 3 - I2C SCLK
PMOD1-> 1 - I2C SDA
PMOD1-> GND - BMP180 GND
PMOD1->Vcc - BMP180 Vcc

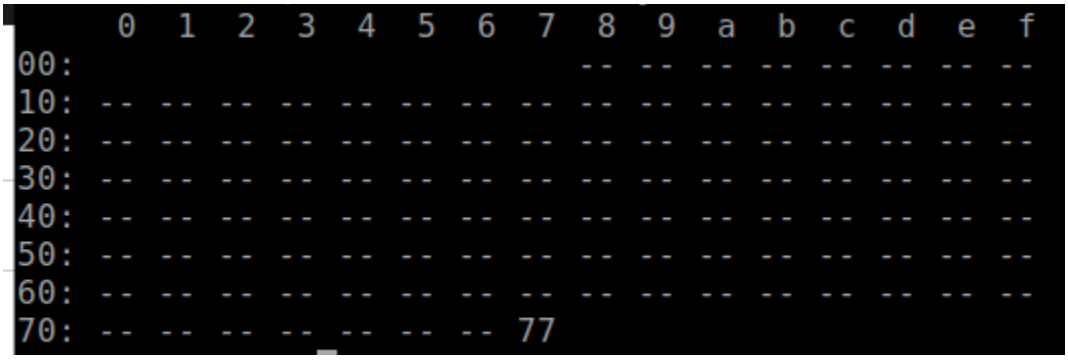
11	9	7	5	3	1	PMOD UPPER
12	10	8	6	4	2	PMOD LOWER
Vcc	GND	I/O	I/O	I/O	I/O	

PMOD port numbering



After connecting BMP180 sensor to KV260 PMOD port, use i2c utility tools to scan for the available devices in i2c-8 channel.

```
sudo i2cdetect -y 3
```



In i2c scan, we find a device is available at address '77', which corresponds to BMP180 i2c sensor.
Next we will add the component for publishing BMP180 sensor data to the AWS IoT cloud.

Setting up AWS IoT Greengrass Core Device

Follow these steps after installing AWS greengrass core device in the KV260 board as mentioned in [Kria connect to AWS IoT - GPIO document].

Installing the component

First install the BMP180 python library by running the following commands in KV260 terminal:

```
git clone https://github.com/m-rtijn/bmp180
cd bmp180
```

Update the ~/bmp180/bmp180/bmp180.py to use i2c-3 channel by changing following lines:

```
import smbus
import math
from time import sleep

class bmp180:
    # Global variables
    address = None
    bus = smbus.SMBus(3)
    mode = 1 # TODO: Add a way to change the mode

    # BMP180 registers
    CONTROL_REG = 0xF4
    DATA_REG = 0xF6

    # Calibration data registers
-- INSERT --                                          16,24         Top
```

Install the bmp180 module by running:

```
sudo python3 setup.py install
```

Get the `components` folder and copy in the KV260 home directory.
It contains:
artifacts

- com.example.bmp180
  - 1.0.0
    - mqtt.py (This python code publishes temperature data)

recipe

- com.example.bmp180-1.0.0.json

For installing the greengrass-cli core device, first bring the AWS_ACCESS_KEY_ID and AWS_SECRET_ACCESS_KEY for the device user to the terminal environment.

```
export AWS_ACCESS_KEY_ID=<AWS ACCESS KEY ID>
export AWS_SECRET_ACCESS_KEY=<AWS SECRET ACCESS KEY>
```

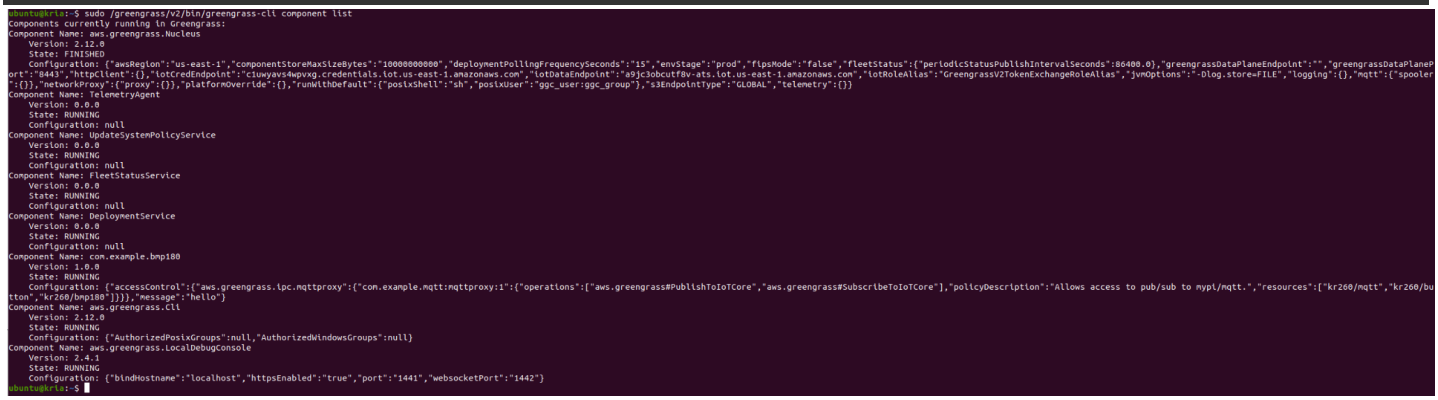
To install the above component run the following in the KV260 terminal:

```
sudo /greengrass/v2/bin/greengrass-cli deployment create \
--recipeDir ~/components/recipe \
--artifactDir ~/components/artifacts \
--merge "com.example.bmp180=1.0.0"
```

```
ubuntu@kria:~$ sudo /greengrass/v2/bin/greengrass-cli deployment create \
--recipeDir ~/components/recipe \
--artifactDir ~/components/artifacts \
--merge "com.example.bmp180=1.0.0"
Local deployment submitted! Deployment Id: d69603fe-b0cb-4caf-a1d3-928694a06502
ubuntu@kria:~$
```

Now check the installed component is in “running state”

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



Now in aws IoT console, open “MQTT test client” and subscribe to “#”
You will find the published message in the `kv260/bmp180` topic.