

## Grove Pi on Raspberry Pi 3 B+

### Hardware

#### *Raspberry Pi 3 B+*

The Raspberry Pi 3 Model B+ is a 1.4GHz, 64-bit quad-core processor with 1GB SDRAM.

Key Features include:

- 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
- Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)
- Extended 40-pin GPIO header

**Note:** This DevKit requires Ubuntu Linux 20.04 (64 bit)

### GrovePi+

GrovePi+ is an add-on board with 15 Grove 4-pin interfaces that allow you to attach Grove sensors to the Raspberry Pi. It is the latest version that is compatible with the Raspberry Pi model B/B+/A+/2/3.

The GrovePi+ starter kit integrates easily with Raspberry Pi and includes a GrovePi+ board and the following sensors/accessories:

- Temperature & Humidity Sensor
- Rotary Angle Sensor
- Light Sensor
- Sound Sensor
- LCD RGB Backlight
- Blue LED
- Red LED
- Green LED
- Buzzer
- Relay
- Button
- Ultrasonic Ranger

Order a kit from here:

RaspberryPi 3 Model B+ , Grove Pi with Sensors and BME680 Sensor

**Note:** The Temperature and Humidity Sensor available in the starter kit is not supported. Instead BME680 is used.

### Hardware Setup

The GrovePi+ board is mounted on the Raspberry Pi as shown below. This allows you to connect

Grove Sensors directly to the board. Make sure the pins are properly aligned.

Connect the Grove Sensors to the ports on the Grove Pi board as shown in the following table:

Port	Interface	Grove Sensor
D2	GPIO	Blue LED
D3	GPIO	Green LED
D4	GPIO	Button
D6	GPIO	Red LED
D7	GPIO	Relay
D8	GPIO	Buzzer
A0	AIO	Light Sensor
A1	AIO	Sound Sensor
A2	AIO	Rotary Angle Sensor
I2C-1	I2C	BME680 Sensor
I2C-2	I2C	RGB Backlit LCD

## Software

The EdgeX Device Grove Service was developed to control/communicate with Grove sensors connected on Grove Pi in an EdgeX deployment. The current version of the service supports the following sensors/functionality:

Sensor	Supported	Available Functionality
Temperature & Humidity Sensor (BME680)	Y	Get Temperature, Pressure and Humidity
Rotary Angle Sensor	Y	Get RotaryAngle & Rotary Voltage
Light Sensor	Y	Get Light Luminosity
Sound Sensor	Y	Get Sound Intensity in Vbiased
LCD RGB Backlight	Y	Write Text, starting at a particular row, column
Blue LED	Y	Turn On/Off Blue LED

Sensor	Supported	Available Functionality
Red LED	Y	Turn On/Off Red LED
Green LED	Y	Turn On/Off Green LED
Buzzer	Y	Turn On/Off Buzzer
Relay	Y	Turn On/Off Relay
Button	Y	Get Button State (Press/Release)
Ultrasonic Ranger	N	- N/A -

## Setup Instructions

### Prerequisite

Download Ubuntu 20.04 LTS - 64 bit available for Raspberry Pi 3 from <https://ubuntu.com/download/raspberry-pi>

- Extract the image ubuntu-20.04-preinstalled-server-arm64+raspi.img using the command:

```
- unxz ubuntu-20.04-preinstalled-server-arm64+raspi.img.xz
```

**Note:** This image was the latest available at the time of testing. You can use a later version if available.

- Flash the image to a FAT32 formatted SD card
- After flashing the SD card, mount it locally and modify the following files in system-boot/
  - config.txt - disable uart, that is, #enable\_uart=1
  - syscfg.txt - disable uart and spi, that is, #enable\_uart=1 and #dtparam=spi=on
- Insert SD card to Raspberry Pi and power on to boot.
- After bootup, upgrade the kernel, to get the latest update

```
cmd> sudo apt update
```

```
cmd> sudo apt dist-upgrade
```

### Software Setup

1. Install docker-ce and set the sudo permission :

- a) Install docker-ce

```
cmd> sudo apt update
```

```
cmd> sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common
```

```
cmd> curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

```
cmd> sudo add-apt-repository "deb [arch=arm64]
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

cmd> sudo apt update

cmd> sudo apt install docker-ce
```

Add user to the docker group

```
cmd> sudo usermod -aG docker $USER
```

Log out and log back in for the change to take effect.

b) Install docker-compose

```
cmd> sudo apt install docker-compose
```

2. Install Node-RED and Node-RED dashboard

a) Install Node-RED

```
cmd> sudo apt update

cmd> sudo apt install nodejs npm

cmd> sudo npm install -g --unsafe-perm node-red
```

b) Start Node-RED by typing

```
cmd> node-red
```

In the web browser, type <http://localhost:1880>

Install Node-RED dashboard and add the dashboard items to the palette as follows:

- Select the **Palette** tab of the User Settings dialog box in Node-RED
- Select the **Install** option from the **Manage Palette** menu
- Search for **node-red-dashboard**
- Install **node-red-dashboard**

**Note:** You must install Node-RED and the Node-RED dashboard to run the demo as described in Using Node-RED.

3. Start the device grove service, use the docker-compose file, available at *edgex\_examples/device-services/grove-c*

```
cmd> cd edgex_examples/device-services/grove-c

cmd> docker-compose -f docker-compose-demo-grove.yml up --detach
```

4. Start the Node-RED Dashboard. To do this, open a web browser and enter <http://localhost:1880/ui>

**Notes:**

1. The mqtt-broker needed to publish the data required by the Node-RED flow is configured in the docker-compose file.
2. edgex-device-grove image is built with the configuration described in the profile Grove\_Device.yml. This file can be found at */device-grove/res/Grove\_Device.yml* within the edgex-device-grove container.

## Testing of Grove Sensors

There are two ways to test the connected sensors on Grove Pi -

1. Using the curl commands

Set of the curl commands used to test the sensors - “device-grove curl commands” are as follows:

**Note:** These commands assume the following:

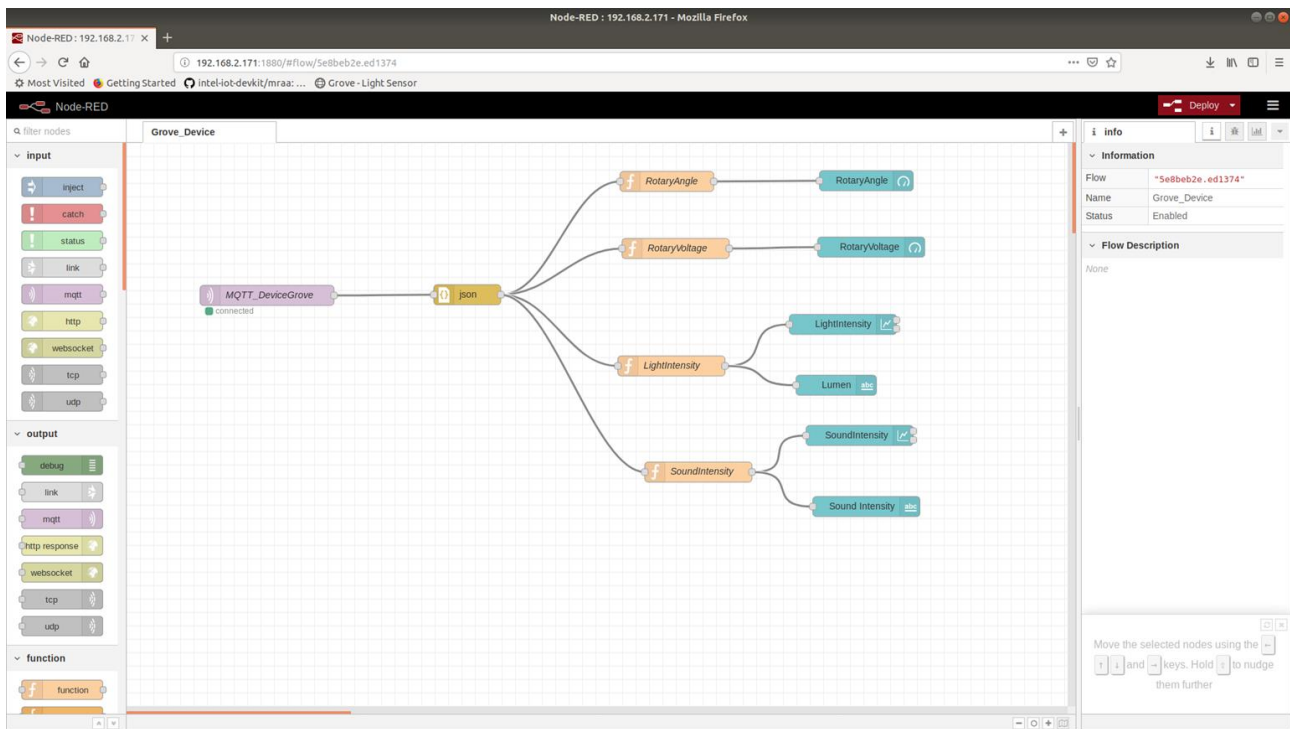
Set/ Get	Grove Sensor	Functionality	Curl Command
Set	Green LED	Turn On	curl -H "Content-Type: application/json" -d '{"Green-LED":"true"}' http://localhost:49992/api/v1/device/all/Set_GreenLed
		Turn Off	curl -H "Content-Type: application/json" -d '{"Green-LED":"false"}' http://localhost:49992/api/v1/device/all/Set_GreenLed
	Red LED	Turn On	curl -H "Content-Type: application/json" -d '{"Red-LED":"true"}' http://localhost:49992/api/v1/device/all/Set_RedLed
		Turn Off	curl -H "Content-Type: application/json" -d '{"Red-LED":"false"}' http://localhost:49992/api/v1/device/all/Set_RedLed
	Blue LED	Turn On	curl -H "Content-Type: application/json" -d '{"Blue-LED":"true"}' http://localhost:49992/api/v1/device/all/Set_BlueLed
		Turn Off	curl -H "Content-Type: application/json" -d '{"Blue-LED":"false"}' http://localhost:49992/api/v1/device/all/Set_BlueLed
	LCD	Write_String	curl -H "Content-Type: application/json" -d '{"Display-String":"Edgex Demo", "Row": "0", "Column": "0"}' http://localhost:49992/api/v1/device/all/Write_String
	Relay	Set	curl -H "Content-Type: application/json" -d '{"Grove-Relay":"true"}' http://localhost:49992/api/v1/device/all/Set_Relay

Set/ Get	Grove Sensor	Functionality	Curl Command
		Unset	curl -H "Content-Type: application/json" -d '{"Grove-Relay": "false"}' http://localhost:49992/api/v1/device/all/Set_Relay
	Buzzer	Turn On	curl -H "Content-Type: application/json" -d '{"Grove-Buzzer": "true"}' http://localhost:49992/api/v1/device/all/Set_Buzzer
		Turn Off	curl -H "Content-Type: application/json" -d '{"Grove-Buzzer": "false"}' http://localhost:49992/api/v1/device/all/Set_Buzzer
Get	Button	Get State	curl -H "Content-Type: application/json" http://localhost:49992/api/v1/device/all/Get_ButtonState
	Rotary Sensor	Get Rotary Sensor Measurements	curl -H "Content-Type: application/json" http://localhost:49992/api/v1/device/all/Get_RotarySensorMeasurements
	Sound Sensor	Get Sound Intensity	curl -H "Content-Type: application/json" http://localhost:49992/api/v1/device/all/Get_SoundIntensity
	Light Sensor	Get Light Intensity	curl -H "Content-Type: application/json" http://localhost:49992/api/v1/device/all/Get_LightIntensity
	BME680 Sensor	Get_TempPressHumidity	curl -H "Content-Type: application/json" http://localhost:49992/api/v1/device/all/Get_TempPressHumidity

- a) Curl commands are issued on the system where the device-grove service is running
- b) device-grove service configuration is set to use port 49992

## 2. Using Node-RED

Current version of the flow - *nodered\_flow.json*, available in the repo - reads the value from the sensors. The updates are displayed on the Node-RED dashboard as Line Charts & Gauge.



- a) In the browser, go to the Node-RED tab, from the menu select 'Import - Clipboard' and copy the contents from *nodered\_flow.json* and click on 'Import'. This creates a tab 'Grove\_Device' on Node-RED. The screen is similar to:
- b) Update the mqtt-server configuration on the 'MQTT\_DeviceGrove' node as follows:
  - Double-click 'MQTT\_DeviceGrove', to open mqtt node.
  - Click edit to update mqtt-broker node.
  - In the 'Server' tab, replace 'mqtt-broker' with the IP of the system on which the mqtt-broker is running.

**Note:** To get the ip-address of the mqtt-broker container running within the edgex-network, using the command:

```
cmd> docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
mqtt-broker
```

- Click 'Update' and 'Done' and 'Deploy' to start the flow.
- c) Test SoundSensor, Light Sensor and Rotary Sensor and monitor the output on the Node-RED dashboard. The following picture on the dashboard captures the variation of Light and Sound intensity captured over a period of one hour.

To test Grove Rotary Sensor, rotate the knob on the Rotary Sensor to observe the change

on the gauge that displays 'Rotary Angle' and 'Rotary Voltage'

