Getting started guide

Functionality

The R718E is identified as a LoRaWAN ClassA device with three-axis acceleration , temperature and compatible with LoRaWAN protocol.
When the device moves or vibrates over threshold value, it immediately reports the temperature ,acceleration and velocity of the X, Y, and Z axes.

Install CLI for AWS IoT Things Graph

```
## Install AWS CLI
https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-install.html

## Install preview Things Graph API models
https://docs.aws.amazon.com/cli/latest/reference/configure/add-model.html

aws configure add-model --service-name iotthingsgraph --service-model file://service-2.json

## Install jq
https://stedolan.github.io/jq/

## Verify preview model installed
aws iotthingsgraph map-property help

expected to see help output instead of "Invalid choice" error message
```

Find your desired device

1. Go to AWS device catalog page: https://devices.amazonaws.com
2. Locate your desired sensor.
3. For private beta, Things Graph is using API v1. For API v1, use the model file sensor_model.json (device manufacturer enters their model file name). For production, Things Graph will use API v2. For API v2, please use the S3 URL link that you see in the Device catalog page.

Using the device model

The following tests will show the different use cases and binary messages than be used with the model. You can use the binary test data to verify correct functionality of the model.

1. Test name: Report Data Command
    a. Mapping used: map_R718E_Uplink
    b. Use case tested: Report Data
    c. Payload format: port = 6

Bytes	1	1	1	Var(Fix=8 Bytes)
	Version	DeviceType	ReportType	NetvoxPayLoadData

Version:1 bytes – 0x01——the Version of NetvoxLoRaWAN Application Command Version
DeviceType: 1 byte – Device Type of Device
ReportType: 1 byte –the Presention of the NetvoxPayLoadData, according the devicetype
NetvoxPayLoadData: Fixed bytes (Fixed = 8bytes)

Device	DeviceType	ReportType	NetvoxPayLoadData				
R718E	0x1C	0x01	Battery (1Byte, unit:0.1V)	AccelerationX (Float16_2Bytes, m/s^2)	AccelerationY (Float16_2Bytes, m/s^2)	AccelerationZ (Float16_2Bytes, m/s^2)	Reserved (1Bytes, Fixed 0x00)
		0x02	VelocityX (Float16_2Bytes, mm/s)	VelocityY (Float16_2Bytes, mm/s)	VelocityZ (Float16_2Bytes, mm/s)	Temperature (Signed 2Bytes, unit:0.1°C)	

    d. Input binary data:
        i. 011C0124000A000E001400
        (DeviceType: 0x1C, ReportType: 0x01, Battery: $24_{hex} = 36_{dec}$, AccelerationX: $000A_{hex} = 10_{dec}$, AccelerationY: $000E_{hex} = 14_{dec}$, AccelerationZ: $00014_{hex} = 20_{dec}$)

        ii. 011C0200140019000500FD
        (DeviceType: 0x1C, ReportType: 0x02, VelocityX: $000A_{hex} = 10_{dec}$, VelocityY: $000E_{hex} = 14_{dec}$, VelocityZ: $00014_{hex} = 20_{dec}$, Temperature: $00FD_{hex} = 253_{dec}$)

    e. API call (V1):

```
aws iotthingsgraph map-property \
    --region us-east-1 \
    --endpoint-url "https://iotthingsgraph.us-east-1.amazonaws.com" \
    --namespace-snapshot file:///path/to/model/json/file \
    --property-value "011C0124000A000E001400" \
    --mapping-info '{"mappingId":"urn:tdm:us-east-
1/000000000000/default:mapping:map_R718E_Uplink","mappingDirection": "FORWARD",
"contextProvider":{"json": "{\"port\":6}"}}'
```

f. Expected results (V1)


```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "Version":"integer",
    "Battery":"aws/iot/thingsgraph/v1/types/Voltage_V",
    "Acceleration_X":"double",
    "ReportType":"integer",
    "Acceleration_Z":"double",
    "Acceleration_Y":"double"
  }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "Version":1,"Battery":3.6,
    "Acceleration_X":10,
    "ReportType":1,
    "Acceleration_Z":20,
    "Acceleration_Y":14
  }"
}
```


g. Expected results (Input data:011C0200140019000500FD)


```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "Temperature":"aws/iot/thingsgraph/v1/types/Temperature_Celsius",
    "Velocity_Z":"double",
    "Version":"integer",
    "Velocity_X":"double",
    "Velocity_Y":"double",
    "ReportType":"integer"
  }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "Temperature":25.3,
    "Velocity_Z":5,
    "Version":1,
    "Velocity_X":20,
    "Velocity_Y":25,
    "ReportType":2
  }"
}
```

2. Test name: Set and read report configuration
    a. Mapping used: map_R718E_Uplink
    b. Use case tested: Configure and read report parameters
    c. Payload format: port = 7

Bytes	1	1	Var(Fix =9 Bytes)
	CmdID	DeviceType	NetvoxPayLoadData

CmdID– 1 bytes
DeviceType– 1 byte – Device Type of Device
NetvoxPayLoadData– var bytes (Max=9 bytes)

Description	CmdID	Device Type	NetvoxPayLoadData				
Config ReportReq	0x01	0x1C	MinTime (2bytes Unit:s)	MaxTime (2bytes Unit:s)	Battery Change (1byte Unit:0.1v)	Acceleration Change (2Bytes, Unit:m/s^2)	Reserved (2Bytes, Fixed 0x00)
Config ReportRsp	0x81		Status (0x00:success)	Reserved (8 Bytes, Fixed 0x00)			
ReadConfig ReportReq	0x02		Reserved (9 Bytes, Fixed 0x00)				
ReadConfig ReportRsp	0x82		MinTime (2bytes Unit:s)	MaxTime (2bytes Unit:s)	Battery Change (1byte Unit:0.1v)	Acceleration Change (2Bytes, Unit:m/s^2)	Reserved (2Bytes, Fixed 0x00)
SetActive ThresholdReq	0x03		ActiveThreshold (2 Bytes)	InActiveThreshold (2 Bytes)	Reserved (5 Bytes, Fixed 0x00)		
SetActive ThresholdRsp	0x83		Status (0x00:success)	Reserved (8 Bytes, Fixed 0x00)			
GetActive ThresholdReq	0x04		Reserved (9 Bytes, Fixed 0x00)				
GetActive ThresholdRsp	0x84		ActiveThreshold (2 Bytes)	InActiveThreshold (2 Bytes)	Reserved (5 Bytes, Fixed 0x00)		

d. Input binary data:

i. **Set Report Configuration**

MinTime = 5min(300s), MaxTime = 15min(900s), BatteryChange = 0.1v, AccelerationChange = 1m/s^2

Downlink: 011C012C03840100010000 $012C_{hex} = 300_{dec}$, $0384_{hex} = 900_{dec}$, 0.1v(Unit:0.1v) => 0.1 ÷ 0.1 = 1, $01_{hex} = 1_{dec}$, $0001_{hex} = 1_{dec}$

Response: 811C000000000000000000 (Configuration success)
811C010000000000000000 (Configuration failure)

ii. **Read Report Configuration**

Downlink: 021C000000000000000000
Response: 821C012C03840100010000 (Current configuration)

iii. **Set Active Threshold**

Assuming that the Active Threshold is set to 10m/s^2, the value to be set is 10/9.8/0.0625=16.32,and the last value obtained is an integer and is configured as 16.

Assuming that the InActive Threshold is set to 8m/s^2, the value to be set is 8/9.8/0.0625=13.06, and the last value obtained is an integer and is configured as 13.

Configure device parameters ActiveThreshold = $16_{dec} = 0010_{hex}$, InActiveThreshold = $13_{dec} = 000D_{hex}$

Downlink: 031C0010000D0000000000
Response: 831C000000000000000000 (Configuration success)
831C010000000000000000 (Configuration failure)

iv. **Get Active Threshold**

Downlink: 041C000000000000000000
Response: 841C0010000D0000000000 (Current configuration)

e. API call (**Set report configuration**):

```
aws iotthingsgraph map-property \
    --region us-east-1 \
    --endpoint-url "https://iotthingsgraph.us-east-1.amazonaws.com" \
    --namespace-snapshot file:///path/to/model/json/file \
    --property-value "011C012C03840100010000" \
    --mapping-info '{"mappingId":"urn:tdm:us-east-
1/000000000000/default:mapping:map_R718E_Uplink","mappingDirection": "FORWARD",
"contextProvider":{"json": "{\"port\":7}"}}'
```

f. Expected results

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum",
    "Payload":"Netvox_R718E_Uplink/types/ConfigureCmdPayload"
    }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "CmdId":"ConfigReportReq",
    "Payload":{
      "AccelerationChange":1,
      "MaxTime":900,
      "MinTime":300,
      "BatteryChange":0.1
      }
  }"
}
```

g. Expected results (input data: 811C000000000000000000)

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum",
    "Payload":"Netvox_R718E_Uplink/types/ConfigureCmdPayload"
    }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "CmdId":"ConfigReportRsp",
    "Payload":{
      "Status":"Success"
    }
  }"
}
```

h. API call (**Read report configuration**):

```
aws iotthingsgraph map-property \
    --region us-east-1 \
    --endpoint-url "https://iotthingsgraph.us-east-1.amazonaws.com" \
    --namespace-snapshot file:///path/to/model/json/file \
    --property-value "021C000000000000000000" \
    --mapping-info '{"mappingId":"urn:tdm:us-east-
1/000000000000/default:mapping:map_R718E_Uplink","mappingDirection": "FORWARD",
"contextProvider":{"json": "{\"port\":7}"}}'
```

i. Expected results

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum"
    }",
  "propertyValue": "{
    "DeviceType":"R718E",
    "CmdId":"ReadConfigReportReq"
    }"
}
```

j. Expected results (input data: 821C012C03840100010000)

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum",
    "Payload":"Netvox_R718E_Uplink/types/ConfigureCmdPayload"
    }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "CmdId":"ReadConfigReportRsp",
    "Payload":{
      "AccelerationChange":1,
      "MaxTime":900,
      "MinTime":300,
      "BatteryChange":0.1
    }
  }"
}
```

k. API call (**Set Active Threshold**):

```
aws iotthingsgraph map-property \
    --region us-east-1 \
    --endpoint-url "https://iotthingsgraph.us-east-1.amazonaws.com" \
    --namespace-snapshot file:///path/to/model/json/file \
    --property-value "031C0010000D0000000000" \
    --mapping-info '{"mappingId":"urn:tdm:us-east-
1/000000000000/default:mapping:map_R718E_Uplink","mappingDirection": "FORWARD",
"contextProvider":{"json": "{\"port\":7}"}}'
```

l. Expected results

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum",
    "Payload":"Netvox_R718E_Uplink/types/ConfigureCmdPayload"
    }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "CmdId":"SetActiveThresholdReq",
    "Payload":{
      "ActiveThreshold":16,
      "InActiveThreshold":13
      }
  }"
}
```

m. Expected results (input data: 831C000000000000000000)

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum",
    "Payload":"Netvox_R718E_Uplink/types/ConfigureCmdPayload"
    }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "CmdId":"SetActiveThresholdRsp",
    "Payload":{
      "Status":"Success"
    }
  }"
}
```

n. API call (**Get Active Threshold**):

```
aws iotthingsgraph map-property \
    --region us-east-1 \
    --endpoint-url "https://iotthingsgraph.us-east-1.amazonaws.com" \
    --namespace-snapshot file:///path/to/model/json/file \
    --property-value "041C000000000000000000" \
    --mapping-info '{"mappingId":"urn:tdm:us-east-
1/000000000000/default:mapping:map_R718E_Uplink","mappingDirection": "FORWARD",
"contextProvider":{"json": "{\"port\":7}"}}'
```

o. Expected results

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum"
    }",
  "propertyValue": "{
    "DeviceType":"R718E",
    "CmdId":"GetActiveThresholdReq"
    }"
}
```

p. Expected results (input data: 841C0010000D0000000000)

```
{
  "propertyId":"{
    "DeviceType":"Netvox_R718E_Uplink/types/DeviceTypeEnum",
    "CmdId":"Netvox_R718E_Uplink/types/CmdIdEnum",
    "Payload":"Netvox_R718E_Uplink/types/ConfigureCmdPayload"
    }",
  "propertyValue":"{
    "DeviceType":"R718E",
    "CmdId":"GetActiveThresholdRsp",
    "Payload":{
      "ActiveThreshold":16,
      "InActiveThreshold":13
    }
  }"
}
```

Support

For questions on this model, please contact: support@netvox.com.tw