

Getting started guide

Functionality

R313LA is a Wireless Infrared Proximity Sensor for Netvox ClassA type devices based on the LoRaWAN open protocol.

It has an infrared proximity sensor that can detect if there is object existing within it detection range and transmits the detected data to Gateway for display via a wireless network.

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_R313LA_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 Presentation of the NetvoxPayloadData, according the devicetype

NetvoxPayloadData: Fixed bytes (Fixed = 8bytes)

Device	DeviceType	ReportType	NetvoxPayloadData			
R313LA	0xAA	0x01	Battery (1Byte, unit:0.1V)	Status (1Byte, 0:off, 1:on)	RawSensorData (2 Bytes)	Reserved (4 Bytes, fixed 0x00)

- d. Input binary data: 01AA011E010BB800000000
 (DeviceType: 0xAA, ReportType: 0x01, Battery: 1E_{hex} = 30_{dec}, Status: 0x01,
 RawSensorData: 0BB8_{hex} = 3000_{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 "01AA011E010BB800000000" \
  --mapping-info '{"mappingId": "urn:tdm:us-east-1/000000000000/default:mapping:map_R313LA_Uplink", "mappingDirection": "FORWARD", "contextProvider": {"json": "{\"port\":6}"}}'
```

- f. Expected results (V1)

```
{
  "propertyId": "Netvox_R313LA_Uplink/types/UplinkPort6Payload",
  "propertyValue": "{
    \"DeviceType\": \"R313LA\",
    \"Version\": 1,
    \"Battery\": 3,
    \"Status\": \"ON\",
    \"RawSensorData\": 3000,
    \"ReportType\": 1
  }"
```

- 2. Test name: Set configuration, read configuration, set distance threshold, read distance threshold
 - a. Mapping used: map_R313LA_Uplink
 - b. Use case tested:
 - (1). Configure and read report parameters
 - (2). Set and read distance threshold
 - 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	0xAA	MinTime (2bytes Unit:s)	MaxTime (2bytes Unit:s)	BatteryChange (1byte Unit:0.1v)	Reserved (4Bytes,Fixed 0x00)
Config ReportRsp	0x81		Status (0x00_success)	Reserved (8Bytes,Fixed 0x00)		
ReadConfig ReportReq	0x02		Reserved (9Bytes,Fixed 0x00)			
ReadConfig ReportRsp	0x82		MinTime (2bytes Unit:s)	MaxTime (2bytes Unit:s)	BatteryChange (1byte Unit:0.1v)	Reserved (4Bytes,Fixed 0x00)
SetOnDistance ThresholdReq	0x03		OnDistance Threshold (2bytes)	Reserved (7 Bytes, Fixed 0x00)		
SetOnDistance ThresholdRsp	0x83		Status (0x00_success)	Reserved (7 Bytes, Fixed 0x00)		
GetOnDistance ThresholdReq	0x04		Reserved (9Bytes,Fixed 0x00)			
GetOnDistance ThresholdRsp	0x84		OnDistance Threshold (2bytes)	Reserved (7 Bytes, Fixed 0x00)		

d. Input binary data:

i. **Set Report Configuration**

MinTime = 5min(300s) 、MaxTime = 15min(900s) 、BatteryChange = 0.1v

Downlink: 01AA012C03840100000000 012C_{hex} = 300_{dec}, 0384_{hex} = 900_{dec}
0.1v(Unit:0.1v) => 0.1 ÷ 0.1 = 1, 01_{hex} = 1_{dec}

Response: 81AA00000000000000000000 (Configuration success)
81AA01000000000000000000 (Configuration failure)

ii. **Read Report Configuration**

Downlink: 02AA00000000000000000000

Response: 82AA012C03840100000000 (Current configuration)

iii. **Set Distance Threshold**

*** Convert Threshold into little-endian

Threshold = 3000

Downlink: 03AAB80B0000000000000000 3000_{dec} = 0BB8_{hex} => B80B (little-endian)

Response: 83AA00000000000000000000 (Configuration success)
83AA01000000000000000000 (Configuration failure)

iv. **Read Distance Threshold**

Downlink: 04AA00000000000000000000

Response: 84AAB80B0000000000000000 (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 "01AA012C03840100000000" \  
  --mapping-info '{"mappingId":"urn:tmd:us-east-1/000000000000/default:mapping:map_R313LA_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"port":7}}}'
```

f. Expected results

```
{  
  "propertyId": "{  
    "DeviceType": "Netvox_R313LA_Uplink/types/DeviceTypeEnum",  
    "CmdId": "Netvox_R313LA_Uplink/types/CmdIdEnum",  
    "Payload": "Netvox_R313LA_Uplink/types/ConfigureCmdPayload"  
  }",  
  "propertyValue": "{  
    "DeviceType": "R313LA",  
    "CmdId": "ConfigReportReq",  
    "Payload": {  
      "MaxTime": 900,  
      "MinTime": 300,  
      "BatteryChange": 0.1  
    }  
  }"  
}
```

g. Expected results (input data: 81AA00000000000000000000)

```
{
  "propertyId": "{
    "DeviceType": "Netvox_R313LA_Uplink/types/DeviceTypeEnum",
    "CmdId": "Netvox_R313LA_Uplink/types/CmdIdEnum",
    "Payload": "Netvox_R313LA_Uplink/types/ConfigureCmdPayload"
  }",
  "propertyValue": "{
    "DeviceType": "R313LA",
    "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 "02AA00000000000000000000" \
  --mapping-info '{"mappingId": "urn:tdm:us-east-1/000000000000/default:mapping:map_R313LA_Uplink", "mappingDirection": "FORWARD", "contextProvider": {"json": {"port": 7}}}'
```

i. Expected results

```
{
  "propertyId": "{
    "DeviceType": "Netvox_R313LA_Uplink/types/DeviceTypeEnum",
    "CmdId": "Netvox_R313LA_Uplink/types/CmdIdEnum"
  }",
  "propertyValue": "{
    "DeviceType": "R313LA",
    "CmdId": "ReadConfigReportReq"
  }"
}
```

j. Expected results (input data: 82AA012C03840100000000)

```
{
  "propertyId": "{
    "DeviceType": "Netvox_R313LA_Uplink/types/DeviceTypeEnum",
    "CmdId": "Netvox_R313LA_Uplink/types/CmdIdEnum",
    "Payload": "Netvox_R313LA_Uplink/types/ConfigureCmdPayload"
  }",
  "propertyValue": "{
    "DeviceType": "R313LA",
    "CmdId": "ReadConfigReportRsp",
    "Payload": {
      "MaxTime": 900,
      "MinTime": 300,
      "BatteryChange": 0.1
    }
  }"
}
```

k. API call (Set Distance 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 "03AAB80B00000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R313LA_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"\"port\":7}}}'
```

l. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R313LA_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R313LA_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R313LA",  
    "CmdId":"SetOnDistanceThresholdReq",  
    "Payload":{"  
      "OnDistanceThreshold":3000  
    }  
  }  
}
```

m. Expected results (input data: 83AA000000000000000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R313LA_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R313LA_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R313LA",  
    "CmdId":"SetOnDistanceThresholdRsp",  
    "Payload":{"  
      "Status":"Success"  
    }  
  }  
}
```

n. API call (Read Distance 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 "04AA0000000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R313LA_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"port":7}}}'
```

o. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R313LA_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/CmdIdEnum"  
  }},  
  "propertyValue": "{  
    "DeviceType":"R313LA",  
    "CmdId":"GetOnDistanceThresholdReq"  
  }"  
}
```

p. Expected results (input data: 84AAB80B0000000000000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R313LA_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R313LA_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R313LA",  
    "CmdId":"GetOnDistanceThresholdRsp",  
    "Payload":{"  
      "OnDistanceThreshold":3000  
    }  
  }  
}
```

3. Test name: Sensor Calibrate

- a. Mapping used: map_R313LA_Uplink
- b. Use case tested: Sensor Calibrate Configuration
- c. Payload format: port = 14

Description	CmdID	Sensor Type	PayLoad (Fix = 9 bytes)				
SetGlobal CalibrateReq	0x01	0x36	Channel (1Byte) Channel1 : 0, Channel2 : 1, etc.	Multiplier (2bytes, Unsigned)	Divisor (2bytes, Unsigned)	DeltValue (2bytes, Signed)	Reserved (2Bytes, Fixed 0x00)
SetGlobal CalibrateRsp	0x81		Channel (1Byte) Channel1 : 0, Channel2 : 1, etc.	Status (1Byte, 0x00_success)	Reserved (7 Bytes, Fixed 0x00)		
GetGlobal CalibrateReq	0x02		Channel (1Byte) Channel1 : 0, Channel2 : 1, etc.	Reserved (8Bytes,Fixed 0x00)			
GetGlobal CalibrateRsp	0x82		Channel (1Byte) Channel1 : 0, Channel2 : 1, etc.	Multiplier (2bytes, Unsigned)	Divisor (2bytes, Unsigned)	DeltValue (2bytes, Signed)	Reserved (2Bytes, Fixed 0x00)

d. Input binary data:

i. **Set Global Calibrate**

Assuming that the current measured distance of R313LA is 3001, it should be corrected to 3000, and the difference is $3000-3001=-1$;
 Channel 1= 00, Multiplier = 0001, Divisor = 0000, DeltValue= -1 (0xFFFF)

Downlink: 01360000010000FFFF0000
 Response: 813600000000000000000000 (Configuration success)
 813601000000000000000000 (Configuration failure)

ii. **Get Global Calibrate**

Downlink: 023600000000000000000000
 Response: 82360000010000FFFF0000 (Current configuration)

e. API call(**Set Global Calibrate**):

```
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 "01360000010000FFFF0000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-  
1/000000000000/default:mapping:map_R313LA_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"\"port\":14}}}'
```

f. Expected results

```
{  
  "propertyId":{"  
    "SensorType":"Netvox_R313LA_Uplink/types/SensorTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/GlobalCalibrateCmdEnum",  
    "Channel":"integer",  
    "Multiplier":"integer",  
    "DeltValue":"integer",  
    "Divisor":"integer"  
  }},  
  "propertyValue":{"  
    "SensorType":"DistanceSensor",  
    "CmdId":"SetGlobalCalibrateReq",  
    "Channel":1,  
    "Multiplier":1,  
    "DeltValue":-1,  
    "Divisor":0  
  }  
}
```

g. Expected results (input data: 8136000000000000000000)

```
{  
  "propertyId":{"  
    "Status":"Netvox_R313LA_Uplink/types/ConfigReportRspStatus",  
    "SensorType":"Netvox_R313LA_Uplink/types/SensorTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/GlobalCalibrateCmdEnum",  
    "Channel":"integer"  
  }},  
  "propertyValue":{"  
    "Status":"Success",  
    "SensorType":"DistanceSensor",  
    "CmdId":"SetGlobalCalibrateRsp",  
    "Channel":1  
  }  
}
```

h. API call(Get Global Calibrate):

```
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 "02360000000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R313LA_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"\"port\":14}}}'
```

i. Expected results

```
{  
  "propertyId":{"  
    "SensorType":"Netvox_R313LA_Uplink/types/SensorTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/GlobalCalibrateCmdEnum",  
    "Channel":"integer"  
  }},  
  "propertyValue":{"  
    "SensorType":"DistanceSensor",  
    "CmdId":"GetGlobalCalibrateReq",  
    "Channel":1  
  }"  
}
```

j. Expected results (input data: 82360000010000FFFF0000)

```
{  
  "propertyId":{"  
    "SensorType":"Netvox_R313LA_Uplink/types/SensorTypeEnum",  
    "CmdId":"Netvox_R313LA_Uplink/types/GlobalCalibrateCmdEnum",  
    "Channel":"integer",  
    "Multiplier":"integer",  
    "DeltValue":"integer",  
    "Divisor":"integer"  
  }},  
  "propertyValue":{"  
    "SensorType":"DistanceSensor",  
    "CmdId":"GetGlobalCalibrateRsp",  
    "Channel":1,  
    "Multiplier":1,  
    "DeltValue":-1,  
    "Divisor":0  
  }"  
}
```

Support

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