

## Getting started guide

### Functionality

R718J is a ClassA type external dry contact device based on LoRaWAN open protocol of Netvox. It can externally connect various switches, buttons, relays and reed switch output. It can detect the closing or disconnecting signal of dry contact and is compatible with LoRaWAN protocol.

### 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\_R718J\_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		
R718J	0x21	0x01	Battery (1Byte, unit:0.1V)	Status (1 Byte, 0: OFF, 1: ON)	Reserved (6 Bytes, fixed 0x00)

- d. Input binary data: 012101240000000000000000  
(DeviceType: 0x21, ReportType: 0x01, Battery: 24<sub>hex</sub> = 36<sub>dec</sub>)

- 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 "012101240000000000000000" \
  --mapping-info '{"mappingId": "urn:tdm:us-east-1/000000000000/default:mapping:map_R718J_Uplink", "mappingDirection": "FORWARD", "contextProvider": {"json": {"port": 6}}}'
```

- f. Expected results (V1)

```
{
  "propertyId": "Netvox_R718J_Uplink/types/UplinkPort6Payload",
  "propertyValue": "{
    \"DeviceType\": \"R718J\",
    \"Status\": \"OFF\",
    \"Version\": 1,
    \"Battery\": 3.6,
    \"ReportType\": 1
  }"
```

- 2. Test name: Set and read report configuration
  - a. Mapping used: map\_R718J\_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	0x21	MinTime (2bytes Unit:s)	MaxTime (2bytes Unit:s)	BatteryChange (1byte Unit:0.1v)	Reserved (4 Bytes, 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 (4 Bytes, Fixed 0x00)

d. Input binary data:

i. **Set Report Configuration**

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

Downlink: 0121**012C**0384**01**00000000 **012C<sub>hex</sub> = 300<sub>dec</sub>**, **0384<sub>hex</sub> = 900<sub>dec</sub>**,  
 0.1v(Unit:0.1v) => 0.1 ÷ 0.1 = 1, **01<sub>hex</sub> = 1<sub>dec</sub>**

Response: 812100000000000000000000 (Configuration success)

812101000000000000000000 (Configuration failure)

ii. **Read Report Configuration**

Downlink: 022100000000000000000000

Response: 8221012C03840100000000 (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 "0121012C03840100000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R718J_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json":{"port":7}}}'
```

f. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718J_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718J_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718J_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718J",  
    "CmdId":"ConfigReportReq",  
    "Payload":{"  
      "MaxTime":900,  
      "MinTime":300,  
      "BatteryChange":0.1  
    }  
  }  
}
```

g. Expected results (input data: 8121000000000000000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718J_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718J_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718J_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718J",  
    "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 "02210000000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-  
1/000000000000/default:mapping:map_R718J_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"\"port\":7}}}'
```

#### i. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718J_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718J_Uplink/types/CmdIdEnum"  
  }},  
  "propertyValue": "{  
    "DeviceType":"R718J",  
    "CmdId":"ReadConfigReportReq"  
  }"  
}
```

#### j. Expected results (input data: 8221012C03840100000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718J_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718J_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718J_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718J",  
    "CmdId":"ReadConfigReportRsp",  
    "Payload":{"  
      "MaxTime":900,  
      "MinTime":300,  
      "BatteryChange":0.1  
    }  
  }  
}
```

## Support

For questions on this model, please contact: [support@netvox.com.tw](mailto:support@netvox.com.tw)