

Getting started guide

Functionality

R718CK2 (nickel-chromium-nickel silicon thermocouple): its use temperature is $-40 \sim +375^{\circ}\text{C}$, with good linearity, large thermoelectromotive force, high sensitivity, stability, can not be used directly at high temperature for sulfur, reducing or reducing. It is not recommended for use in weak oxidizing atmospheres in oxidizing alternating atmospheres and in vacuum.

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_R718CK2_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			
R718CK2	0x16	0x01	Battery (1Byte, unit:0.1V)	Temperature1 (2 Bytes, signed unit:0.1°C)	Temperature2 (2 Bytes, signed unit:0.1°C)	Reserved (3 Bytes, fixed 0x00)

- d. Input binary data: 0116012400FD0109000000
 (DeviceType: 0x16, ReportType: 0x01, Battery: 24_{hex} = 36_{dec}, Temperature1: 00FD_{hex} = 253_{dec},
 Temperature2: 0109_{hex} = 265_{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 "0116012400FD0109000000" \
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R718CK2_Uplink","mappingDirection": "FORWARD",
  "contextProvider":{"json": "{\\"port\\":6}"}}'
```

- f. Expected results (V1)

```
{
  "propertyId": "Netvox_R718CK2_Uplink/types/UplinkPort6Payload",
  "propertyValue": "{
    "DeviceType": "R718CK2",
    "Temperature1": 25.3,
    "Temperature2": 26.5,
    "Version": 1,
    "Battery": 3.6,
    "ReportType": 1
  }"
```

2. Test name: Set and read report configuration
 - a. Mapping used: map_R718CK2_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	0x16	MinTime (2bytes Unit:s)	MaxTime (2bytes Unit:s)	Battery Change (1byte Unit:0.1v)	Temperature Change (2Bytes, Unit:0.1°C)	Reserved (2Bytes, 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)	Battery Change (1byte Unit:0.1v)	Temperature Change (2Bytes, Unit:0.1°C)	Reserved (2Bytes, Fixed 0x00)	
SetThermocoupleTypeReq	0x03		ThermocoupleType (1Byte,0x00_Jtype,x01_Ktype, 0x02_Ttype,0x03_Rtype,0x04_Etype)			Reserved (8Bytes, Fixed 0x00)		
SetThermocoupleTypeRsp	0x83		Status (0x00:success)	Reserved (8Bytes, Fixed 0x00)				
GetThermocoupleTypeReq	0x04		Reserved (9Bytes, Fixed 0x00)					
GetThermocoupleTypeRsp	0x84		ThermocoupleType (1Byte,0x00_Jtype,x01_Ktype, 0x02_Ttype,0x03_Rtype,0x04_Etype)			Reserved (8Bytes, Fixed 0x00)		

d. Input binary data:

i. **Set Report Configuration**

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

TemperatureChange = 1°C

Downlink: 0116012C038401000A0000 012C_{hex} = 300_{dec}, 0384_{hex} = 900_{dec},

0.1v(Unit:0.1v) => $0.1 \div 0.1 = 1$, 01_{hex} = 1_{dec}

1°C(Unit:0.1°C) => $1 \div 0.1 = 10$, 000A_{hex} = 10_{dec}

Response: 81160000000000000000 (Configuration success)

81160100000000000000 (Configuration failure)

ii. **Read Report Configuration**

Downlink: 02160000000000000000

Response: 8216012C038401000A0000 (Current configuration)

iii. **Set Thermocouple Type**

Downlink: 03160100000000000000

Response: 83160000000000000000 (Configuration success)

83160100000000000000 (Configuration failure)

iv. **Get Thermocouple Type**

Downlink: 04160000000000000000

Response: 84160100000000000000 (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 "0116012C038401000A0000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R718CK2_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json":{"port":7}}}'
```

f. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718CK2_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718CK2",  
    "CmdId":"ConfigReportReq",  
    "Payload":{"  
      "TemperatureChange":1,  
      "MaxTime":900,  
      "MinTime":300,  
      "BatteryChange":0.1  
    }  
  }  
}"  
}
```

g. Expected results (input data: 811600000000000000000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718CK2_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718CK2",  
    "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 "02160000000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R718CK2_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"\"port\":7}}}'
```

i. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum"  
  }},  
  "propertyValue": "{  
    "DeviceType":"R718CK2",  
    "CmdId":"ReadConfigReportReq"  
  }"  
}
```

j. Expected results (input data: 8216012C038401000A0000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718CK2_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718CK2",  
    "CmdId":"ReadConfigReportRsp",  
    "Payload":{"  
      "TemperatureChange":1,  
      "MaxTime":900,  
      "MinTime":300,  
      "BatteryChange":0.1  
    }  
  }  
}
```

k. API call (Set Thermocouple Type):

```
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 "0316010000000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R718CK2_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"port":7}}}'
```

l. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718CK2_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718CK2",  
    "CmdId":"SetThermocoupleTypeReq",  
    "Payload":{"  
      "ThermocoupleType":"K Type"  
    }  
  }  
}
```

m. Expected results (input data: 8316000000000000000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718CK2_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718CK2",  
    "CmdId":"SetThermocoupleTypeRsp",  
    "Payload":{"  
      "Status":"Success"  
    }  
  }  
}
```

n. API call (Get Thermocouple Type):

```
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 "04160000000000000000" \  
  --mapping-info '{"mappingId":"urn:tdm:us-east-1/000000000000/default:mapping:map_R718CK2_Uplink","mappingDirection": "FORWARD",  
"contextProvider":{"json": {"\"port\":7}}}'
```

o. Expected results

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum"  
  }},  
  "propertyValue": "{  
    "DeviceType":"R718CK2",  
    "CmdId":"GetThermocoupleTypeReq"  
  }"  
}
```

p. Expected results (input data: 8416010000000000000000)

```
{  
  "propertyId":{"  
    "DeviceType":"Netvox_R718CK2_Uplink/types/DeviceTypeEnum",  
    "CmdId":"Netvox_R718CK2_Uplink/types/CmdIdEnum",  
    "Payload":"Netvox_R718CK2_Uplink/types/ConfigureCmdPayload"  
  }},  
  "propertyValue":{"  
    "DeviceType":"R718CK2",  
    "CmdId":"GetThermocoupleTypeRsp",  
    "Payload":{"  
      "ThermocoupleType":"K Type"  
    }  
  }  
}
```

[Support](#)

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