
Tech Note: IO-Link Node-Red Event Function

Last Updated
1/31/20

Technical Support

Contents

Testing Environment.....	2
Prerequisites	2
Background	3
Set-up Node Red Function	3

Testing Environment

Description	Name/Type	Version
Node-Red		V0.20.7
Node.js		V10.14.1
EtherNet/IP IO-Link Master	BNI006A (BNI EIP-508-105-Z105)	H_06 S_4.4
Multiple IO-Link Device	Varies	Varies

Prerequisites

- Be able to start a Node-Red application
- Be able to set-up a http request node, get Parsed JSON Data from the IO-Link Master at IP/ports.json



Edit http request node

Delete Cancel Done

Properties

Method GET

URL http://192.168.1.11/ports.json

☐ Append msg payload as query string parameters

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Use proxy

Return a parsed JSON object

Name Name

Tip: If the JSON parse fails the fetched string is returned as-is.

Background

When using Node-Red to utilize the JSON data, there is much more data available. One type of valuable data is Event data, this is the “extra” data that points out Errors, Warnings and Notification about the state of the IO-Link Device. This document will show an example Node-Red Function Node to extract the Event Data.

Example of Events from User Manual for IO-Link Power Supply:

Event Codes

Definition	Event-Codes IO-Link 1.1	Event-Codes IO-Link 1.0	Device status	Type
No malfunction	0x0000	0x0000	0	Notifica- tion
General mal- function. Unknown error	0x1000	0x1000	4	Error
Process variable range over-run. Process Data uncertain	0x8C10	0x8C10	2	Warning
Process variable range under-run. Process Data uncertain	0x8C30	0x8C10	2	Warning
PSU discon- nected from Clip	0xB000	0x1800	4	Error
Bad connec- tion between PSU and Clip	0xB001	0x1801	4	Error
PSU changed	0xB004	0x1804	4	Error
High stress level	0xB005	0x1805	4	Warning
Fast aging	0xB006	0x1806	4	Warning
Over Tempera- ture	0xB007	0x1807	4	Warning
Over load	0xB008	0x1808	4	Warning

Set-up Node Red Function

- Add http request node and connect function node

The screenshot shows a Node-RED workflow with three nodes: a 'timestamp' node, an 'http request' node, and a 'function' node. Blue arrows point from the 'http request' and 'function' nodes to their respective configuration panels.

Edit http request node

Properties

- Method: GET
- URL: http://192.168.1.11/ports.json
- ☐ Append msg.payload as query string parameters
- ☐ Enable secure (SSL/TLS) connection
- ☐ Use authentication
- ☐ Use proxy
- Return: a parsed JSON object
- Name: Name

Tip: If the JSON parse fails the fetched string is returned as-is.

Edit function node

Properties

- Name: Name

Function

```
1  
2 return msg;
```

Outputs: 1

Info Panel

Information

- Node: "741ed8fc-88a3f8"
- Type: function

Description

Node Help

A JavaScript function block to run against the messages being received by the node. The messages are passed in as a JavaScript object called `msg`. By convention it will have a `msg.payload` property containing the body of the message. The function is expected to return a message object (or multiple message objects), but can choose to return nothing in order to halt a flow.

Details

See the [online documentation](#) for more information on writing functions.

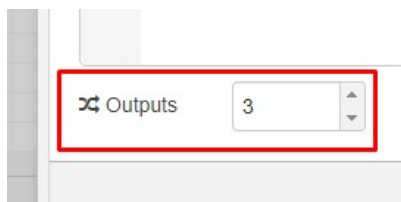
Sending messages

The function can either return the messages it wants to pass on to the next nodes in the flow, or can call `node.send(messages)`. It can return/send:

- a single message object - passed to

Hold down `ctrl` when you `click` on a node to add or remove it from the current selection

- Adjust the Outputs at the bottom of the function node to 3



- Notepad file



Global Event Function Javascript 1-30-20.txt

- Copy text from Notepad document into function node

Global Event Function Javascript 1-30-20.txt - Notepad

```
File Edit Format View Help
//Function evaluates EventFlag and Event JSON data
//Three Outputs: Event Mode (msg1), Event Type (msg2), Event Code (msg3)
msg1={};
msg2={};
msg3={};
//**** Set PORT value to JSON Port Number ****
var PORT = 0;
//Replace spaces with no spaces
var iOLEventFlag = (msg.payload.ports[PORT].eventFlag).replace(/\s/g, '');
//Collect High Byte from eventFlag and isolate bits 6 and 7
var iOLEventMode = (parseInt(iOLEventFlag.slice(2,4),16) & 192) >>> 6;
//Evaluate Event Mode to Match Correct Text
if (iOLEventMode === 0) {message1payload = "Reserved";
} else if (iOLEventMode == 1) {message1payload = "Event One Shot";
} else if (iOLEventMode == 2) {message1payload = "Event Disappears";
} else if (iOLEventMode == 3) {message1payload = "Event Appears";
} else {message1payload = "No Event";
}
//Collect High Byte from eventFlag and isolate bits 4 and 5
var iOLEventType = (parseInt(iOLEventFlag.slice(2,4),16) & 48) >>> 4;
//Evaluate Event Type to Match Correct Text
if (iOLEventType === 0) {message2payload = "Reserved";
} else if (iOLEventType == 1) {message2payload = "Notification";
} else if (iOLEventType == 2) {message2payload = "Warning";
} else if (iOLEventType == 3) {message2payload = "Error";
} else {message2payload = "No Event";
}
//Collect Event Code Data, left as text
var iOLEventCode = (msg.payload.ports[PORT].event);
//Evaluate Event Code to Match Correct Text
if (iOLEventCode == "0x0000") {message3payload = "No Malfunction";
} else if (iOLEventCode == "0x1000") {message3payload = "General Malfunction - unknow";
} else if (iOLEventCode == "0x4000") {message3payload = "Temperature Fault - Overload";
} else if (iOLEventCode == "0x4210") {message3payload = "Device Temperature Over-Run";
}
```

Edit function node

Delete Cancel Done

Properties

Name

Function

```
1 //Function evaluates EventFlag and Event JSON data
2 //Three Outputs: Event Mode (msg1), Event Type (msg2), Event Code (msg3)
3 msg1={};
4 msg2={};
5 msg3={};
6 //**** Set PORT value to JSON Port Number ****
7 var PORT = 0;
8 //Replace spaces with no spaces
9 var iOLEventFlag = (msg.payload.ports[PORT].eventFlag).replace(/\s/g, '');
10 //Collect High Byte from eventFlag and isolate bits 6 and 7
11 var iOLEventMode = (parseInt(iOLEventFlag.slice(2,4),16) & 192) >>> 6;
12 //Evaluate Event Mode to Match Correct Text
13 if (iOLEventMode === 0) {message1payload = "Reserved";
14 } else if (iOLEventMode == 1) {message1payload = "Event One Shot";
15 } else if (iOLEventMode == 2) {message1payload = "Event Disappears";
16 } else if (iOLEventMode == 3) {message1payload = "Event Appears";
17 } else {message1payload = "No Event";
18 }
19 //Collect High Byte from eventFlag and isolate bits 4 and 5
20 var iOLEventType = (parseInt(iOLEventFlag.slice(2,4),16) & 48) >>> 4;
21 //Evaluate Event Type to Match Correct Text
22 if (iOLEventType === 0) {message2payload = "Reserved";
23 } else if (iOLEventType == 1) {message2payload = "Notification";
24 } else if (iOLEventType == 2) {message2payload = "Warning";
25 } else if (iOLEventType == 3) {message2payload = "Error";
26 } else {message2payload = "No Event";
27 }
28 //Collect Event Code Data, left as text
29 var iOLEventCode = (msg.payload.ports[PORT].event);
30 //Evaluate Event Code to Match Correct Text
31 if (iOLEventCode == "0x0000") {message3payload = "No Malfunction";
32 } else if (iOLEventCode == "0x1000") {message3payload = "General Malfunction - unknow";
33 } else if (iOLEventCode == "0x4000") {message3payload = "Temperature Fault - Overload";
34 } else if (iOLEventCode == "0x4210") {message3payload = "Device Temperature Over-Run";
35 }
```

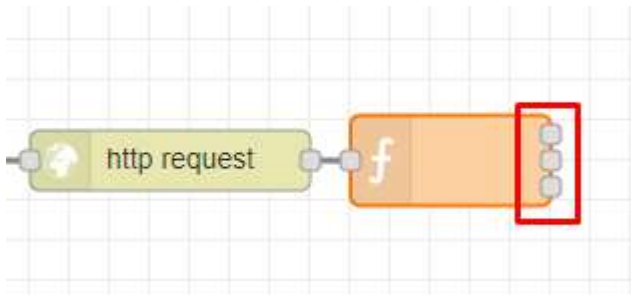
Outputs 3

- Set the PORT number that coincides with the IO-Link port based on the JSON port designation

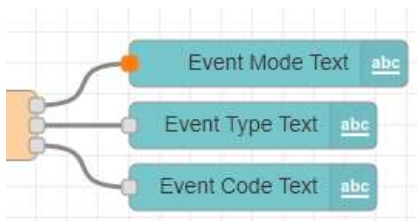
Function

```
1 //Function evaluates EventFlag and Event JSON data
2 //Three Outputs: Event Mode (msg1), Event Type (msg2), Event Code (msg3)
3 msg1={};
4 msg2={};
5 msg3={};
6 //**** Set PORT value to JSON Port Number ****
7 var PORT = 0;
8 //Replace spaces with no spaces
9 var iOLEventFlag = (msg.payload.ports[PORT].eventFlag).replace(/\s/g, '');
10 //Collect High Byte from eventFlag and isolate bits 6 and 7
```

- The final function will have 3 output connection points



- The 3 outputs connection contain the following text outputs
 - Output 1 – Event Mode
 - Event One Shot
 - Event Disappears
 - Event Appears
 - Output 2 – Event Type
 - Notification
 - Warning
 - Error
 - Output 3 – Event Code
 - Unique codes based on IO-Link Specification version 1.1.2
 - Also included are some Balluff vendor specific codes
 - Note: All codes may not be represented in this function
- Optional:
 - If using the Node-Red Dashboard function, you can tie text field nodes to the output points



Events

Event Mode
Event Appears

Event Type
Warning

Event Code

Process Variable Range Over-Run - Process Data Uncertain