

## Edge iO28 Specifications (Hardware V1.5)

<b>Power Options and Requirements:</b>	
Via Terminal	Power Supply: 24VAC +/- 3% or 24VDC +10%/-10% Consumption 100 mA
<b>Connectivity:</b>	
Ethernet	2x Ethernet (Switch) Ethernet 10/100base-T auto-selection
<b>Communication Options:</b>	
SSH over IP	Edge iO 28 gets IP address via DHCP via default, can be configured as Static
Edge Connect Module	Provides power and breakout for I2C, UART and GPIO add-on modules



Available IO		Edge Connect Module	
Universal Inputs	7	I2C Channels	2
Digital Inputs	7	UART Channels	2
Universal Outputs	7	GPIO	1
Digital Outputs	5	Power	5V DC wired or 24V serial connection
Relays	2		

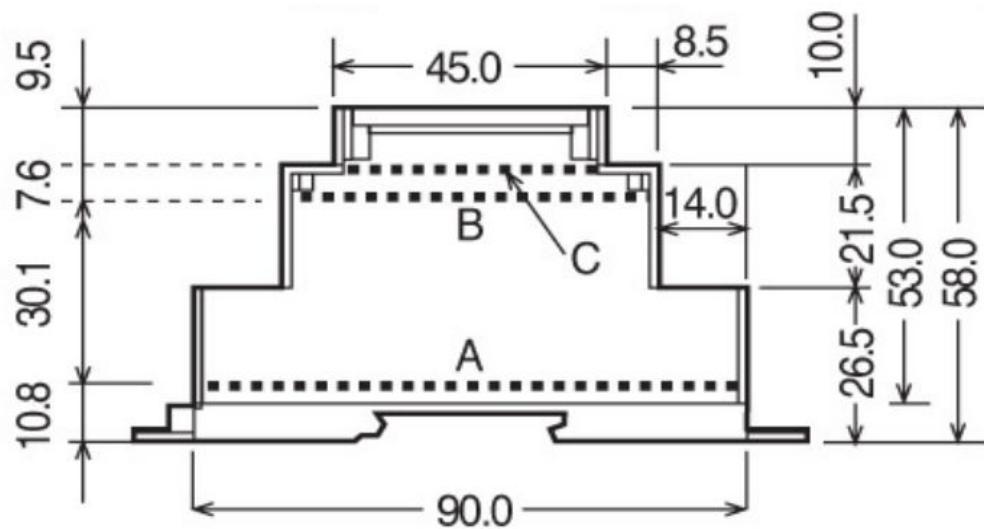
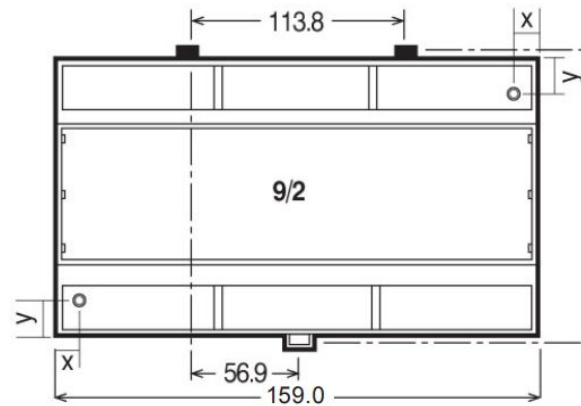
## Input/Output Details

Name	Tolerance	Usage
Digital Inputs	Dry Contact	Detecting the closing or opening of circuit
Universal Input	0-10V DC, 4-20ma, 10/20K Resistor	Measuring 0-10V Sensors, Voltage, Temperature and current
Universal Output	0-12V @ 50ma Max	Controlling voltage variable equipment. Can drive a slim-line relay
Digital Output	0V - 12V On-Off	Controlling relays, enabling equipment
Relays	Max 24V/2amp Capable of switching 3.3VDC to 48VDC and VAC	NO dry contact




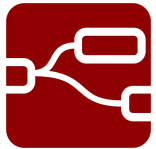


## Dimensions

<b>Length</b>	159 mm
<b>Width</b>	90 mm
<b>Height</b>	58 mm
<b>Material Type</b>	Plastic (Polycarbonate)
<b>UL Rating</b>	UL94-V0
<b>IP Rating</b>	IP20
<b>Mounting</b>	DIN mount
<b>Weight</b>	335 gms






## Computing and Programming

<b>Physical Specs</b>		Processor: AM335x 1GHz ARM® Cortex-A8 512MB DDR3 RAM 4GB 8-bit eMMC on-board flash storage NEON floating-point accelerator 2x PRU 32-bit microcontrollers USB client for power & communications USB host Optional storage of an SD card upto 64Gb
<b>OS Software</b>		Debian based OS
<b>Software</b>		JVM and node-js are default installed on the device
<b>Programming</b>		Local Flow Based Programming On-board API Node Red

## Universal Input Jumper Settings:

The Universal Inputs of the Edge are used to set the configuration from 10/20K Resistor, 0-10V DC and 4-20ma. There are three settings to configure universal input type, set with a 2-pin **jumper**.



Mode	Purpose	Setting
<b>10/20K Resistor</b>	Connect a thermistor to the input and common.	
<b>4-20 ma</b>	Using 4-20 ma sensitive sensors.	
<b>0-10 VDC</b>	Measuring 0-10 VDC.	



## IP Address

The default IP address is set to DHCP

## Edge-IO 28 Inbuilt API

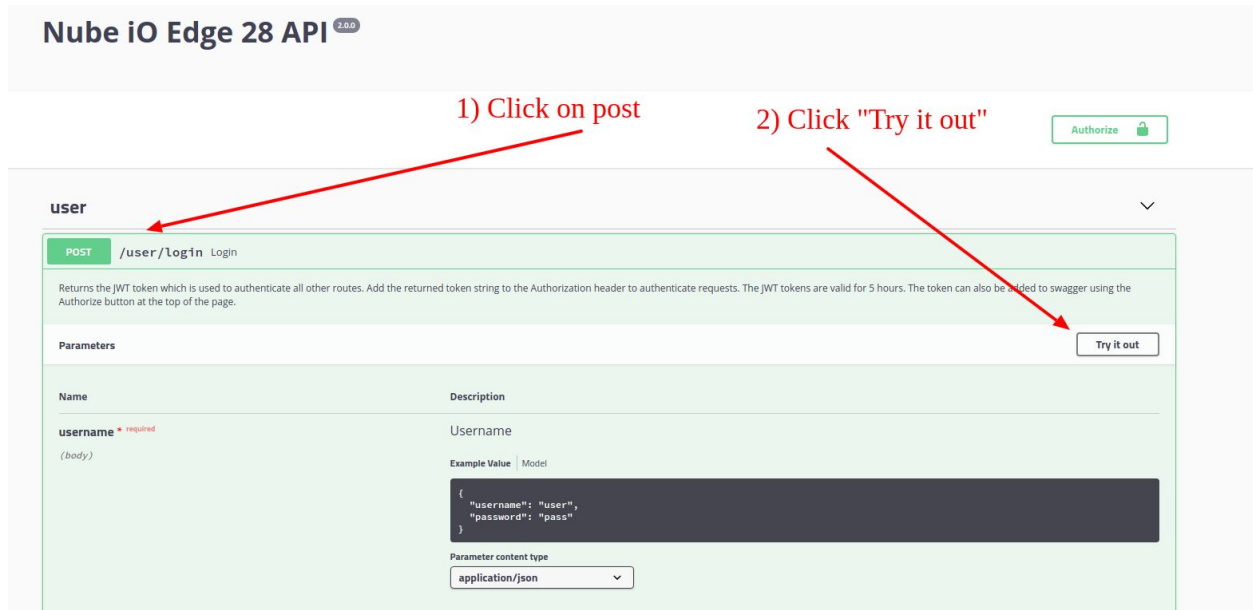
The inbuilt api is a http-rest server. The api lets give you full flex ability to add and update point objects and point values

- Schedules
- Points
- Alerts
- Tags
- Histories

## Swagger API

<http://192.168.15.151:4000/>

See how to test the API



The screenshot displays the Swagger UI for the Nube iO Edge 28 API. At the top, the title "Nube iO Edge 28 API" is followed by a version badge "3.8.0". In the top right corner, there is an "Authorize" button with a lock icon. The main content area shows the "user" namespace with a dropdown arrow. Under "user", the "POST /user/login Login" endpoint is selected. A red arrow labeled "1) Click on post" points to the "POST" method. The endpoint description states: "Returns the JWT token which is used to authenticate all other routes. Add the returned token string to the Authorization header to authenticate requests. The JWT tokens are valid for 5 hours. The token can also be added to swagger using the Authorize button at the top of the page." Below the description is a "Parameters" section. It includes a "body" parameter named "username" with a red asterisk indicating it is required. The "Description" column shows "Username". An "Example Value" is provided in a dark box: 

```
{  "username": "user",  "password": "pass"}
```

. Below the example, a "Parameter content type" dropdown is set to "application/json". A red arrow labeled "2) Click 'Try it out'" points to the "Try it out" button in the top right of the endpoint details.



user

POST /user/login Login

Returns the JWT token which is used to authenticate all other routes. Add the returned token string to the Authorization header to authenticate requests. The JWT tokens are valid for 5 hours. The token can also be added to swagger using the Authorize button at the top of the page.

Parameters

Name	Description
username <span>required</span> (body)	<div>Username</div> <div>Example Value   Model</div> <div> <pre>{   "username": "admin",   "password": "a725a5ceca59" }</pre> </div>

Cancel

Parameter content type  
application/json

Execute

3) Enter your username and password

4) Then run "Execute"

Execute

Clear

Responses

Response content type application/json

Curl

```
curl -X POST "http://144.139.37.173:4000/user/login" -H "accept: application/json" -H "Content-Type: application/json" -d '{"username": "admin", "password": "a725a5ceca59"}'
```

Request URL

http://144.139.37.173:4000/user/login

Server response

Code

Details


200

Response body

```
{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2x1IjoiYmRta41LC2IjOiJmFzZSI6ImFkbWUiLCJ1IjoiYW90IjozNTUxOTk0MDh1LCJleHAiOiJlNTIwMTIwMzV9.B81xSCDsEsFgs7_NK193Plb2QH1LmVjNz4sx3okFshw",
  "data": {
    "role": "admin",
    "username": "admin"
  }
}
```

Download

5) Copy your token

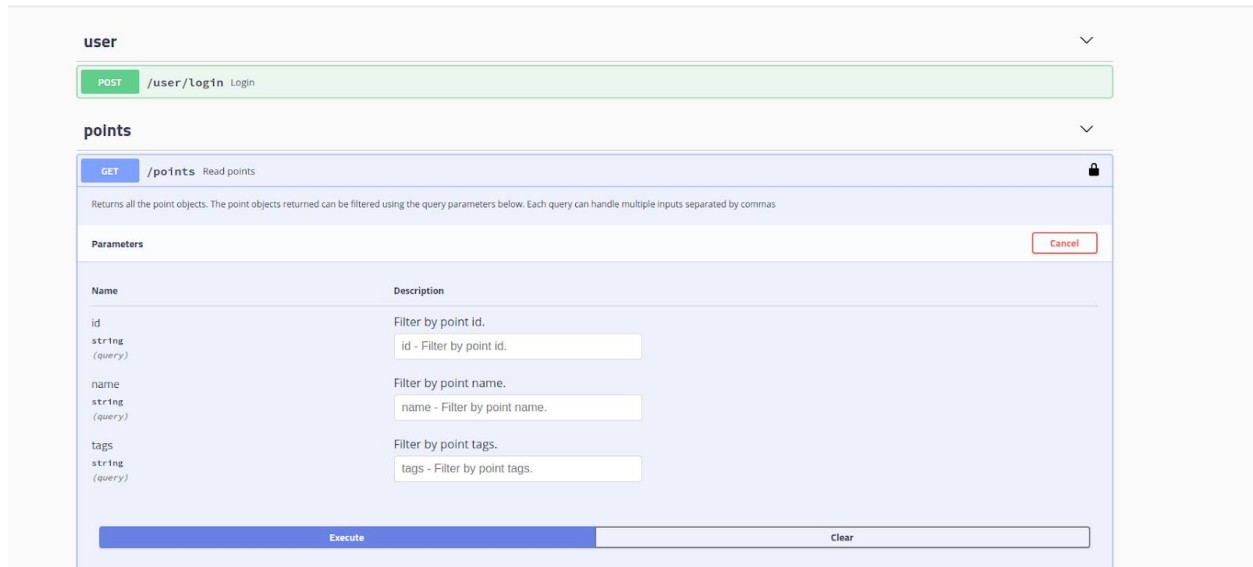


nube-io.com - Edge IO-28 User Guide. Hardware V1.5 , Nov 2018, V2.2 Page 7 of 20





Now you can interact with the device API



The screenshot shows the NUBE iO API interface. At the top, there is a 'user' section with a 'POST /user/login Login' button. Below it is the 'points' section, which is expanded. The 'points' section has a 'GET /points Read points' button and a lock icon. Below the button, there is a description: 'Returns all the point objects. The point objects returned can be filtered using the query parameters below. Each query can handle multiple inputs separated by commas'. Below the description is a 'Parameters' section with a 'Cancel' button. The 'Parameters' section contains a table with two columns: 'Name' and 'Description'. The table has three rows: 'id' (string, query) with a description 'Filter by point id.' and an input field 'id - Filter by point id.'; 'name' (string, query) with a description 'Filter by point name.' and an input field 'name - Filter by point name.'; and 'tags' (string, query) with a description 'Filter by point tags.' and an input field 'tags - Filter by point tags.'. At the bottom of the 'Parameters' section are two buttons: 'Execute' and 'Clear'.

## Rest-API Endpoint

Example to read all points

<http://127.0.0.1:4000/points>

Read a Point

<http://127.0.0.1:4000/points/UI1>

Read a Point details

<http://127.0.0.1:4000/points/UI1/kind>

Read Histoires

<http://127.0.0.1:4000/histories/data>

<http://127.0.0.1:4000/histories/UI1/data>



## Schedules

<http://127.0.0.1:4000/schedules>

## Alerts

<http://127.0.0.1:4000/alerts>



```
{
  - UI1: {
    name: "UI1",
    type: "0-10dc",
    scale: "0:10",
    precision: 3,
    offset: 0,
    value: 0,
    pinValue: 0,
    kind: "Number",
    unit: "null",
    tags: [ ],
    - historySettings: {
      type: "periodic",
      schedule: "0 */15 * * * *",
      size: 672
    }
  },
  - UI2: {
    name: "UI2",
    type: "0-10dc",
    scale: "0:10",
    precision: 3,
    offset: 0,
    value: 0,
    pinValue: 0,
    kind: "Number",
    unit: "null",
    tags: [ ],
    - historySettings: {
      type: "periodic",
      schedule: "0 */15 * * * *",
      size: 672
    }
  },
  - UI3: {
    name: "UI3",
    type: "0-10dc",
    scale: "0:10",
    precision: 3,
    offset: 0,
    value: 0,
    pinValue: 0,
    kind: "Number",
    unit: "null",
    tags: [ ],
    - historySettings: {
      type: "periodic",
      schedule: "0 */15 * * * *",
      size: 672
    }
  }
}
```



## Node-red

### Getting Started

Website

<https://nodered.org/>

Forum

<https://discourse.nodered.org/>

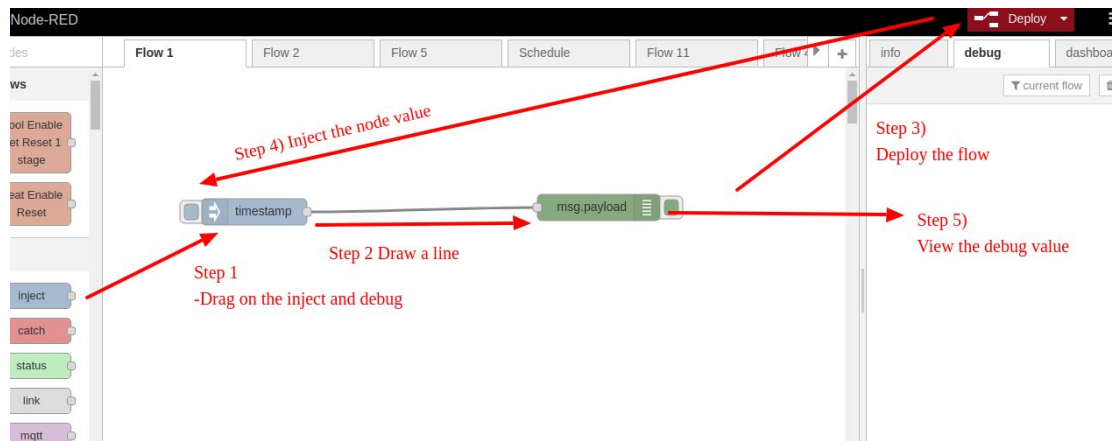
### Default Port for node-red

The default port is 1880, If you're using the USB host the open chrome and go to 192.168.7.2:1880

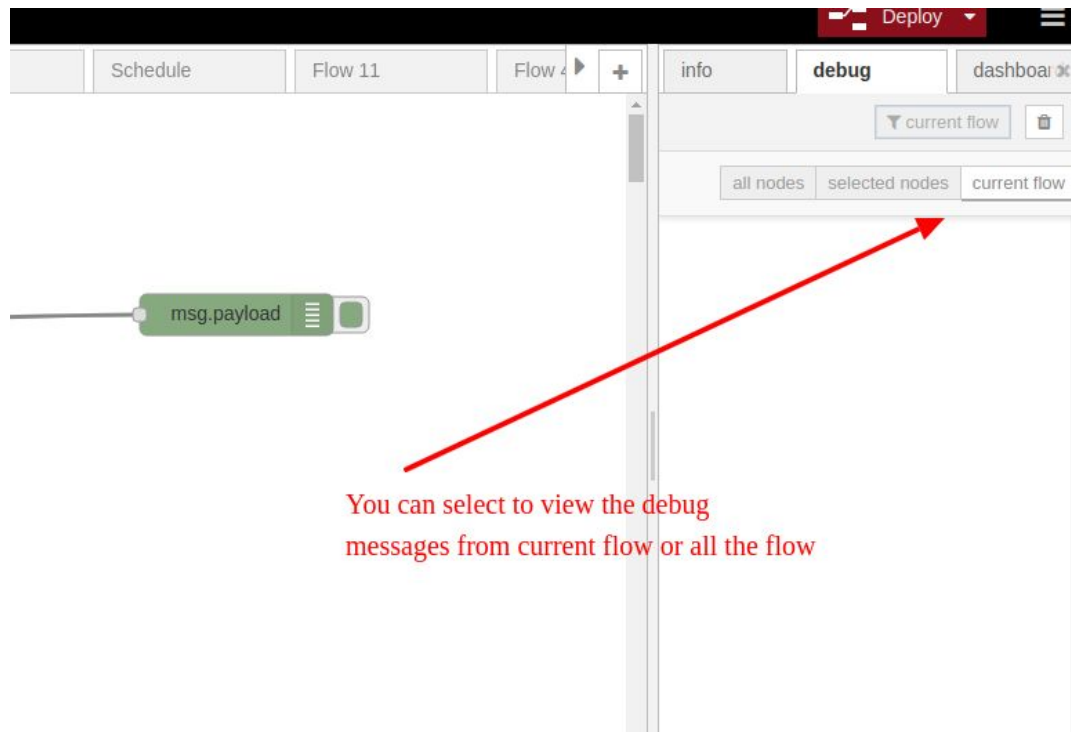


## Basics in node-red

### Create a flow



## View Debug Messages



## Deploy a flow



The screenshot shows the NUBE iO interface with a flow editor. At the top, there's a tab bar with 'Flow 1', 'Flow 2', 'Flow 5', 'Schedule', 'Flow 11', and 'Flow 4'. Below the tabs, a flow diagram is visible with a 'timestamp' node connected to a 'msg.payload' node. A 'Deploy' button is located in the top right corner. A dropdown menu is open, showing three options: 'Full', 'Modified Flows', and 'Modified Nodes'. Red arrows point from text annotations to these options.

You can select the flow deploy methods

The full deploy will restart all the flow runtime

- Full**  
Deploys everything in the workspace
- Modified Flows**  
Only deploys flows that contain changed nodes
- Modified Nodes**  
Only deploys nodes that have changed



## Node Red Function Code

### Boolean Functions

AND, OR, NOT

Logical Operators

Logical operators are used to determine the logic between variables or values.

Given that  $x = 6$  and  $y = 3$ , the table below explains the logical operators:

Operator	Description	Example
&&	and	$(x < 10 \ \&\& \ y > 1)$ is true
	or	$(x == 5 \    \ y == 5)$ is false
!	not	$!(x == y)$ is true

### Comperserson

[https://www.w3schools.com/js/js\\_comparisons.asp](https://www.w3schools.com/js/js_comparisons.asp)





Operator	Description	Comparing	Returns
==	equal to	x == 8	false
		x == 5	true
		x == "5"	true
===	equal value and equal type	x === 5	true
		x === "5"	false
!=	not equal	x != 8	true
!==	not equal value or not equal type	x !== 5	false
		x !== "5"	true
		x !== 8	true
>	greater than	x > 8	false
<	less than	x < 8	true
>=	greater than or equal to	x >= 8	false



<=	less than or equal to	x <= 8	true
----	-----------------------	--------	------

## Javascript Functions

```
function myFunction(p1, p2) {
  return p1 * p2;
}
msg.payload = myFunction(4, 4);

return msg;
```

## Arithmetic Operators

Operator	Description
+	Addition
-	Subtraction
*	Multiplication



/	Division
%	Modulus
++	Increment
--	Decrement

## Math Functions

[https://www.w3schools.com/js/js\\_math.asp](https://www.w3schools.com/js/js_math.asp)

### Math.random()

#### Example

- 1 is the start number
- 6 is the number of possible results (1 + start (6) - end (1))

```
msg.payload = (Math.floor(Math.random() * 6) + 1 ); //(1 + start (6) - end (1))  
  
return msg;
```



## Conversions

### Converting Strings to Numbers

Method	Description
parseFloat()	Parses a string and returns a floating point number
parseInt()	Parses a string and returns an integer

