



## CIM 10

**Devices (Modbus RTU, Modbus TCP IP, OPC UA and OPC DA)**

CIMCON DIGITAL (India) Pvt. Ltd.

## Contents

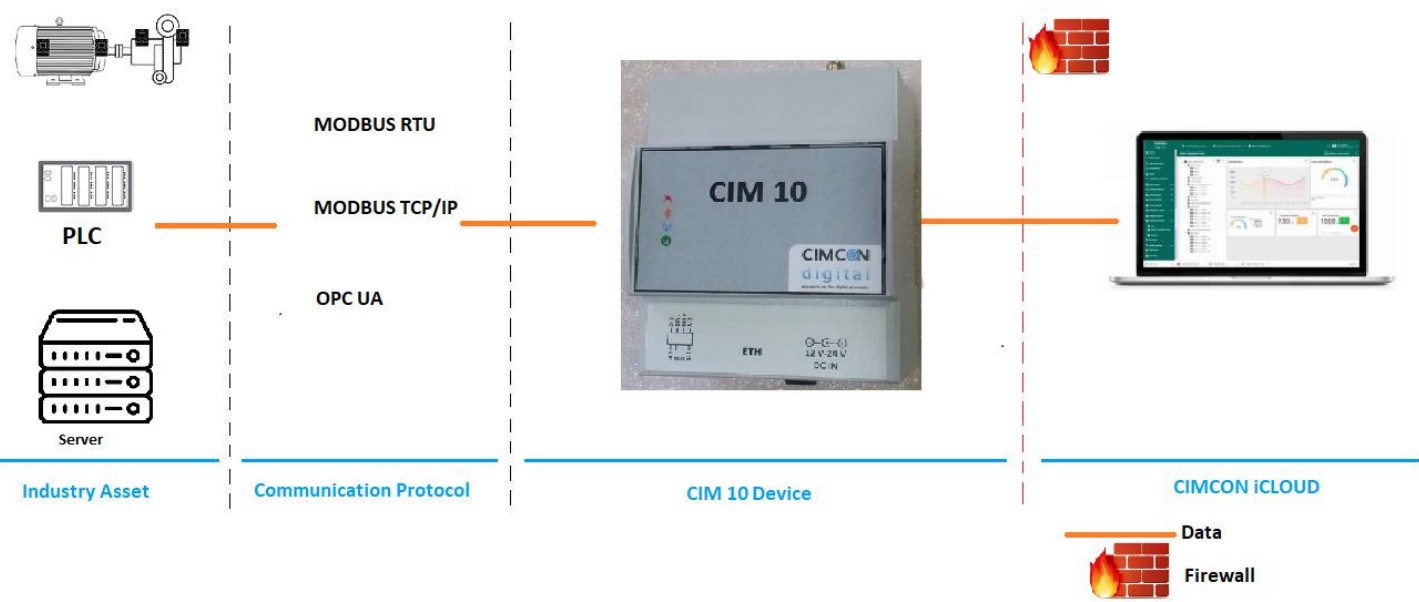
Block Diagram.....	1
1. Modbus RTU .....	2
Overview .....	2
MODBUS REGISTER MAP .....	2
Communication between MODBUS Devices.....	3
Basic Modbus Network.....	3
It's Application .....	3
CIM 10 device support MODBUS RTU Features .....	4
Pre-requisites .....	5
Example: Secure Meter .....	6
Connection Setup.....	6
Under Modbus RTU .....	9
Add Writable Tag .....	11
The details to configure a single query would be as below:.....	12
How to add Parameter under that query .....	13
2. Modbus TCP/IP .....	16
CIM 10 device support MODBUS TCP/IP Features .....	17
Pre-requisites: .....	18
Connection Setup: .....	18
Add Writable Tag .....	21
Configuration: .....	22
The details to configure a single query would be as below: Communication Protocol Modbus-TCP: .....	22
3. OPC UA .....	26
Overview:.....	26
CIM 10 device OPC UA Features.....	27
Pre-Requisite:.....	28

# CIM 10

- OPC UA Client/Server Configuration: ..... 29
  - Server Side ..... 29
  - Client-Side: ..... 30
  - Parameter Configuration:..... 34
  - Configuring a Variable or Object ..... 38
- 4. OPC DA ..... 41
  - Overview:..... 41
  - Pre-Requisite:..... 41
  - OPC DA Client/Server Configuration: ..... 41
    - Server Side ..... 41
    - Server Side ..... 41
    - Client-Side: ..... 43
  - Under OPC DA ..... 45
  - ADD Tags..... 47
  - CONTACT US..... 49

# CIM 10

## Block Diagram



## 1. Modbus RTU

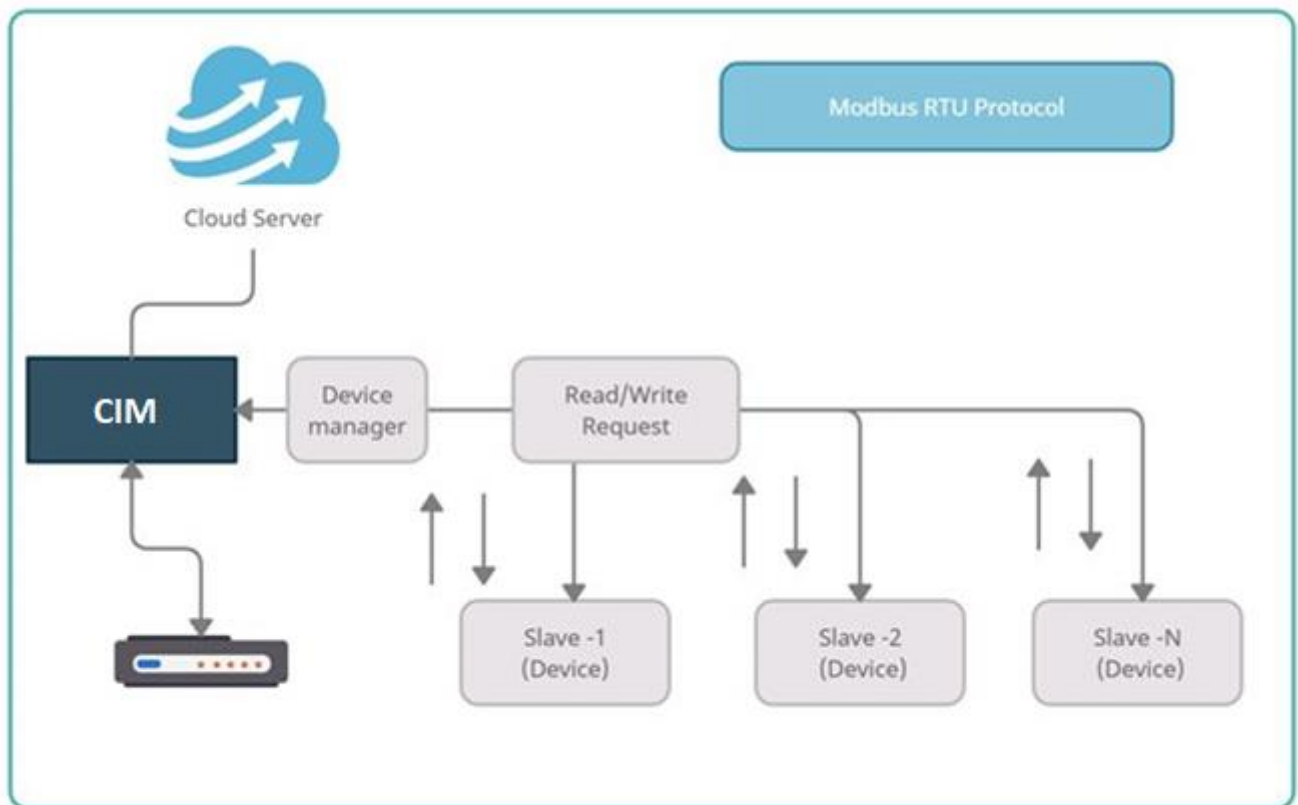
### Overview

MODBUS Protocol used to establish master-slave/client-server communication between intelligent devices. It is a de facto standard, truly open, and the most widely used network protocol in the industrial manufacturing environment. The MODBUS protocol provides an industry-standard method that MODBUS devices use for parsing messages.

### MODBUS REGISTER MAP

CIM 10 device are pre-configured to work with CIMCON iCLOUD. Reliability manager can install these devices easily and quickly using mobile application and can start monitoring equipment in just few hours.

Block Diagram



## Communication between MODBUS Devices

MODBUS devices communicate using a master-slave technique in which only one device (the master) can initiate transactions (called queries).

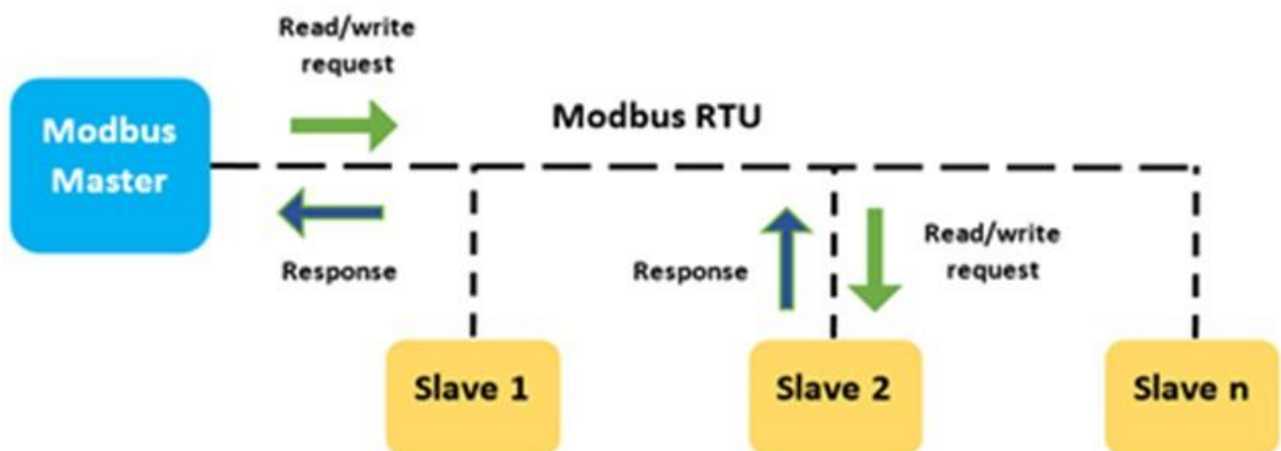
The other devices (slaves) respond by supplying the requested data to the master

A slave is any peripheral device (I/O transducer, valve, network, or other measuring devices), which processes information and sends its output to the master.

Masters can address individual slaves or can initiate a broadcast message to all slaves.

## Basic Modbus Network

Modbus serial protocol (the original version) is a master/slave protocol, e.g. one master that controls the Modbus data transactions with multiple slaves that respond to the master's requests to read from or write data to the slaves. Network architectures are shown in the Figure below.



## It's Application

- Used to establish master-slave/client-server communication between intelligent devices
- The machines you see in factories are commonly controlled by Programmable Logic Controllers (PLC), Industrial Control Systems (ICS), Distributed Control Systems (DCS), and/or Variable Frequency Drive (VFD).

- Modbus enables the communication between many (approximately 247) devices connected to the same network
- Programmable Logic Controller (PLC) and Programmable Automation Controller (PAC) are primarily industrial systems that are designed keeping in mind the manufacturing environment in a plant. These computers are highly reliable when it comes to gauging the ability to control and monitor industrial processes.
- MODBUS is used to monitor and program devices.
- MODBUS used to communicate intelligent devices with sensors and instruments
- MODBUS used to monitor field devices using PCs and HMIs
- MODBUS is also an ideal protocol for RTU applications where wireless communication is required.

## **CIM 10 device support MODBUS RTU Features**

- Supports multiple types of equipment via RTU
- Supports adjustable address base (0 or 1)
- Supports full address range (0-65535)
- Supports word and byte swapping (byte order):
  - MSW: Most significant word first
  - LSW: Least significant word first
  - MSB: Most significant byte first
  - LSB: Least significant byte first
- Supports equipment slave id full range (1-254)
- Supported functions:
  - read coil status (01);
  - read input status (02);
  - read holding registers (03);
  - read input registers (04);
  - force single-coil (05);
  - force multiple coils (15);



- preset multiple registers (16);
- Connection status
- Supported data types;
  - Boolean
  - Integer8
  - Integer16
  - Integer32
  - Unsigned8
  - Unsigned16
  - Unsigned32
  - Floating Point 32
- Support for reading/writing data that spans multiple contiguous registers with different sizes and byte orders.
- Adjustable polling request time per equipment.
- Adjustable minimum request interval per register.
- Adjustable pooling request timeout.
- Supports minimum channel silence, forcing a time between every request on the serial bus.
- Allows different communication options (baud rate, byte size, parity and stop bits) on the same serial bus

## **Pre-requisites**

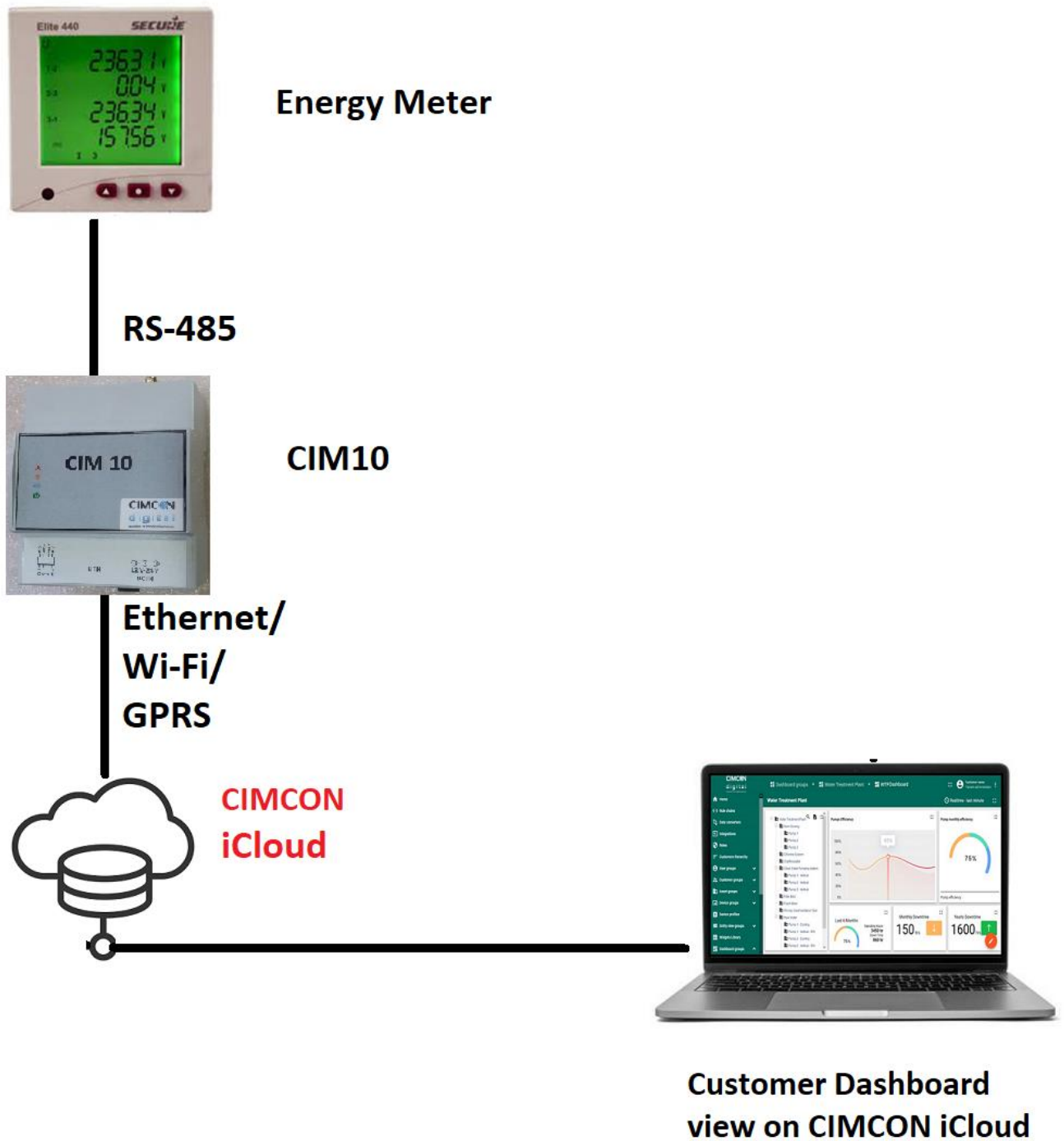
- Modbus RTU Interface
- CIM 10 device and Rs485 Devices
- Interface CIM 10 device and Energy Meter Device on RS485 Port of RTU.



## Example: Secure Meter

- Here we are using Modbus RTU Protocol to test the device data on Webui.

## Connection Setup



- Open webUI with provided URL by CIMCON support team (192.168.3.100) and webpage look like as below. Default configuration is as below.

**IP:** 192.168.3.100

**Subnet :**255.255.255.0

**Gateway:**192.168.3.1



- Login with Valid Username and Password.

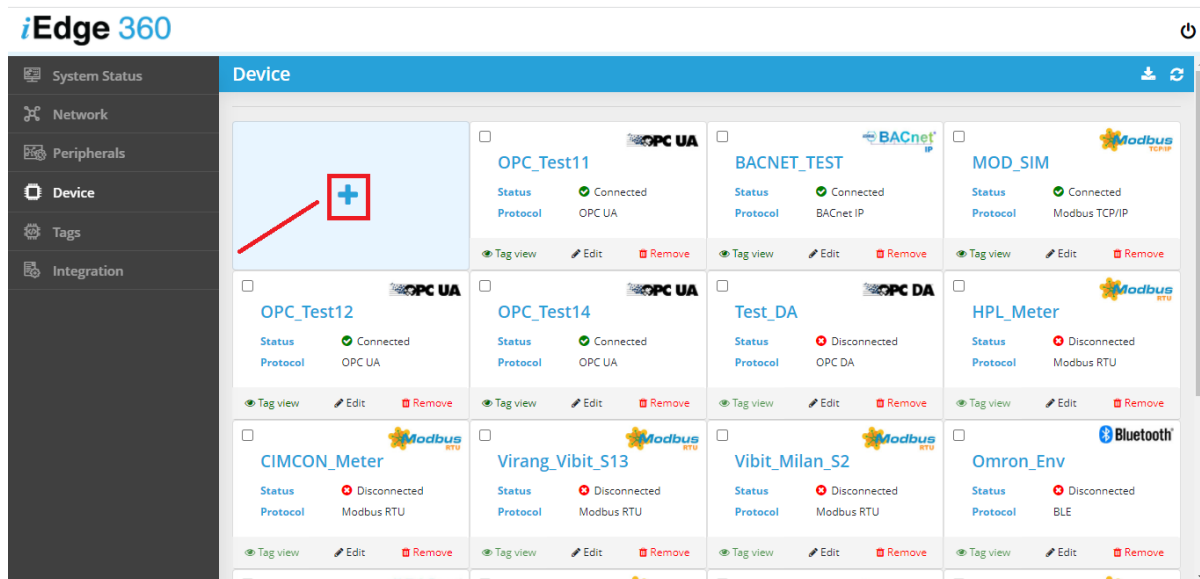
User Name: iEdgeAdmin

Password: iEA@12345

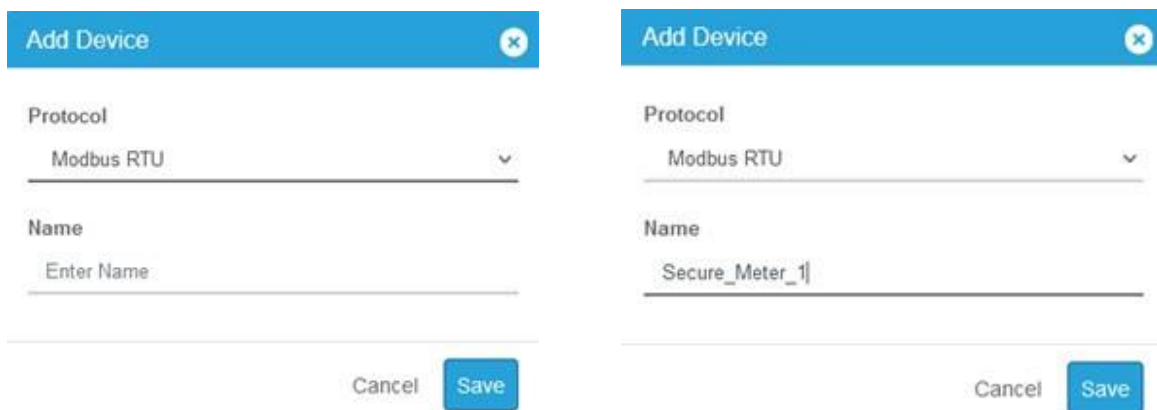


- Connection Setup on webUI

We can add the new device using + sign as like in the below screenshot.



- After click on the add device option, we will get a pop-up window on which we can select the protocol on which we want to create a device and give a proper name for that device. Then click on the "Save" button to see the device on the dashboard.



The device will be created like as below Image. It contains many details which as below

- Device Name. (Give a proper name to the device)
- Device Status. (Connected, Disconnected, Unknown)
- Tag View. (It is reflecting live data of the device after configuration)
- Edit. (Add configuration of the device)
- Remove. (To remove/delete the device from the dashboard)



- The below basic configuration would help in establishing a connection between Energy meters to Modbus RTU on the webUI.

Modbus RTU

Device

Protocol

Modbus RTU

Name

HPL\_Meter

Destination

CIM1

+ Add Writable Tag

Port

Port Select

☒ Serial
 ☐ TCP IP
 ☐ MAC

Port

RS485M

Parity

None

Stop bits

1

Baudrate

9600

Data bits

8

Port timeout (Sec)

3

Device Template

Please Select

Save

## Under Modbus RTU

### Device

**Protocol** - Already selected during device creation step.

**Name** – Already configured during device creation step.

**Destination** – Here user need to mention destination ex. "CIM1" where user can see the result of all parameter. In this case, this service sends data to CIMCON iCLOUD.

### Port

**Port Select** – There are two options for port selection. CIM 10 device supports one RS485 so based on wiring user can select an appropriate one. Need to select the "Serial" option for Modbus RTU protocol.

Port	RS485E	Parity	None	Stop bits	1
	RS485M				
Baudrate	19200	Data bits	8	Port Timeout	5

Port

Port Select

☒ Serial
 ☐ TCP IP
 ☐ MAC

Port

RS485E

▼

Parity

None

▼

Stop bits

1

▼

Baudrate

19200

▼

Data bits

8

▼

Port timeout (Sec)

5

▼

- webUI Support multiple baudrate to connect different peripheral devices.
- Below are port detail of CIM 10 device web portal i.e. webUI support and are configurable.

Parameter	Supported Values	Default Values
Baud Rate	4800	19200
	9600	
	19200	
	38400	
	57600	
	115200	
Data bits	8	8
Parity	EVEN = 0	None
	ODD = 1	
	NONE = 2	
Stop Bit	1	1

## Add Writable Tag

- webUI portal support to write a query on the physical device connected to CIM 10 device. Users can check the peripheral permission, to check writable option is allowed or not.

The screenshot shows a webUI form titled "Add Writable Tag". The form has a blue header bar with the title and a close button (X). Below the header, there are several input fields: "Slave ID" with the value "0", "Address" with the value "0", "Input" with a dropdown menu showing "Coil Input", "Name" with the text "Name", "Type" with a dropdown menu showing "Boolean", and "Value" with radio buttons for "Enable" (selected) and "Disable". At the bottom right of the form is a blue "Save" button.

- Below are detail need to be added for updating the writable tag.

Slave ID	1 to 254	Address	1 to 65535
Input	Coil input	Name	User have configurable name option
	Holding single register		
	Holding multiple register		
Type	Boolean	Value	Enable or Disable (For Coil input)
	Int16		
	Unit16		

## Configuration:

- We can configure the device using below details:
- First, the user needs to create a query on the webUI.

Modbus RTU

Port

Port Select

Serial

TCP IP

MAC

Port

RS485E

Parity

None

Stop bits

1

Baudrate

19200

Data bits

8

Port timeout (Sec)

5

Device Template

Please Select

+ Add Query

Query

+

Query

+

Query

+

Query

+

Save

The details to configure a single query would be as below:

- Communication Protocol Modbus-RTU:

Modbus RTU

Query

Remove Query

Slave ID

5

Interval (Sec)

30

Input

Holding Register

Peripheral ID

4210

Address

7

Peripheral Name

HPL\_Enegry\_Meter\_1

No. of Register

20

Query Interval(Ms)

5000

Parameters

Data type

Integer

Factor

0.1

Converter

Float

Name

HPL\_Meter\_Voltage

Remove

+ Add Parameter

Save

12 | CIMCON Digital India Pvt. Ltd.



Slave ID	1 to 254	Interval (Sec)	1 to 300
Input	Drop down menu 1. Coil Input 2. Input status 3. Holding register 4. Input register		
Address	1 to 65535	Peripheral ID	1 to 9999
No. of register	1 to 32	Peripheral Name	User configurable
		Query Interval	1 to 5000

\* Input need to select one input from dropdown menu

- The slave ID of the device will be unique.
- Each query will have a peripheral name assigned to it.
- **Peripheral ID Will be provided by the CIMCON Customer Success Team.**

## How to add Parameter under that query

- To add a parameter user need to click add a parameter. Users can add a maximum of 16 parameters.

The screenshot shows the 'Modbus RTU' configuration window. At the top, there's a 'Query' section with fields for Slave ID (5), Input (Holding Register), Address (7), No. of Register (20), Interval (Sec) (30), Peripheral ID (4210), Peripheral Name (HPL\_Energy\_Meter\_1), and Query Interval (Ms) (5000). A 'Remove Query' button is in the top right. Below this is a 'Parameters' section with a table. The table has columns: Data type, Factor, Converter, Name, and Remove. One parameter is listed: Data type 'Integer', Factor '0.1', Converter 'Float', Name 'HPL\_Meter\_Voltage'. An 'Add Parameter' button (green plus icon) is in the top right of the parameters section. A 'Save' button is at the bottom right.

Data type	Factor	Converter	Name	Remove
Integer	0.1	Float	HPL_Meter_Voltage	

- Users can configure parameters as per the below detail and selection available at the webUI.

Data type	Factor	Converter	Name
Float	Up to 9999999	Float	User can configure as per device manual
Long		Swapped Float	
Binary		Long	
Integer		Hex	
Character		None	
Decimal			

Modbus RTU

Slave ID

5

Interval (Sec)

30

Input

Holding Register

Peripheral ID

4210

Address

7

Peripheral Name

HPL\_Enegry\_Meter\_1

No. of Register

20

Query Interval(Ms)

5000

Remove Query

Parameters

Integer

0.1

Float

HPL\_Meter\_Voltage

Remove

Integer

0.1

Float

HPL\_Meter\_Phase\_Current

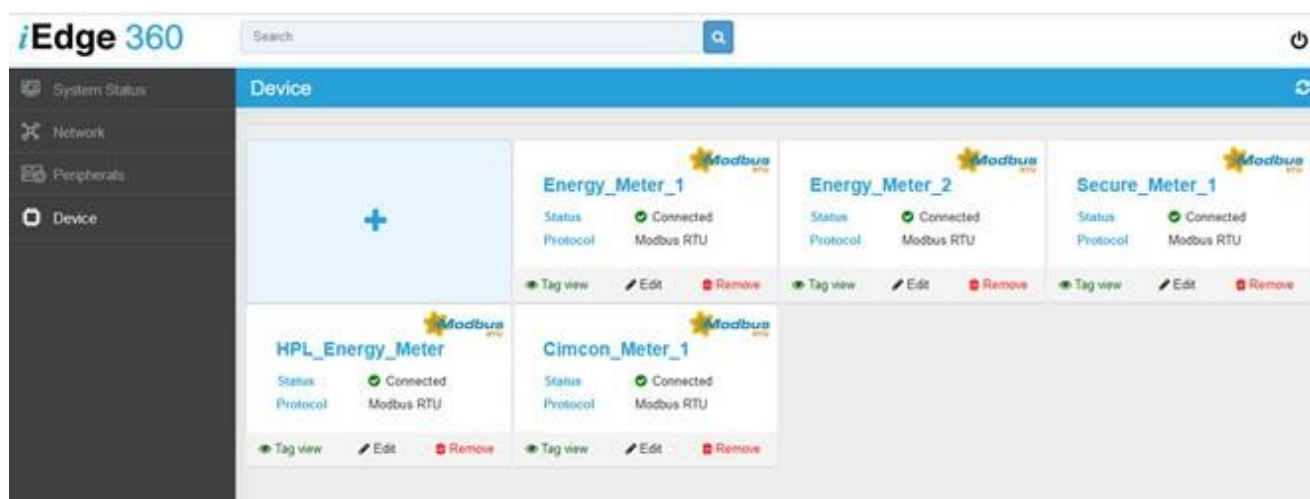
Remove

Add Parameter

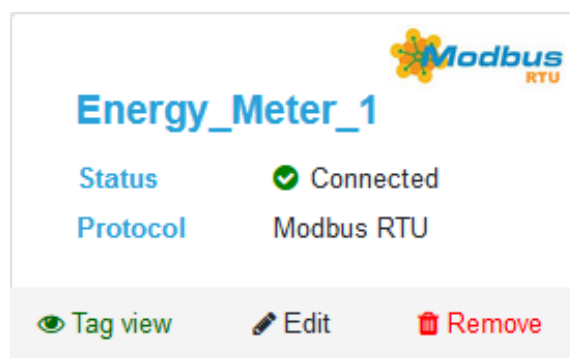
Save

## Note:

- We can add a maximum of 32 queries and 16 parameters each query-wise.
- Users can add different energy meter devices of different make supporting RTU Modbus to webUI
- After adding the Queries and parameters, the device will be created with connected status.



- Now user can able to see data on webUI, just need to click on Tag view



- Data Showing as per define query in device:-

Tags - Energy_Meter_1				
Cimcon_Energy_Meter1				
Name	Value	Type	Timestamp	Status
VRV	418.07	Float	2021/08/10 13:32:01	✓
VVB	418.14	Float	2021/08/10 13:32:01	✓
VBR	418.08	Float	2021/08/10 13:32:01	✓
IR	0.00	Float	2021/08/10 13:32:01	✓
IY	0.00	Float	2021/08/10 13:32:01	✓
IB	0.00	Float	2021/08/10 13:32:01	✓
Frequency	50.05	Float	2021/08/10 13:32:01	✓
Load	0.00	Float	2021/08/10 13:32:01	✓
PF	1.00	Float	2021/08/10 13:32:01	✓
Kwh	0.00	Float	2021/08/10 13:32:01	✓

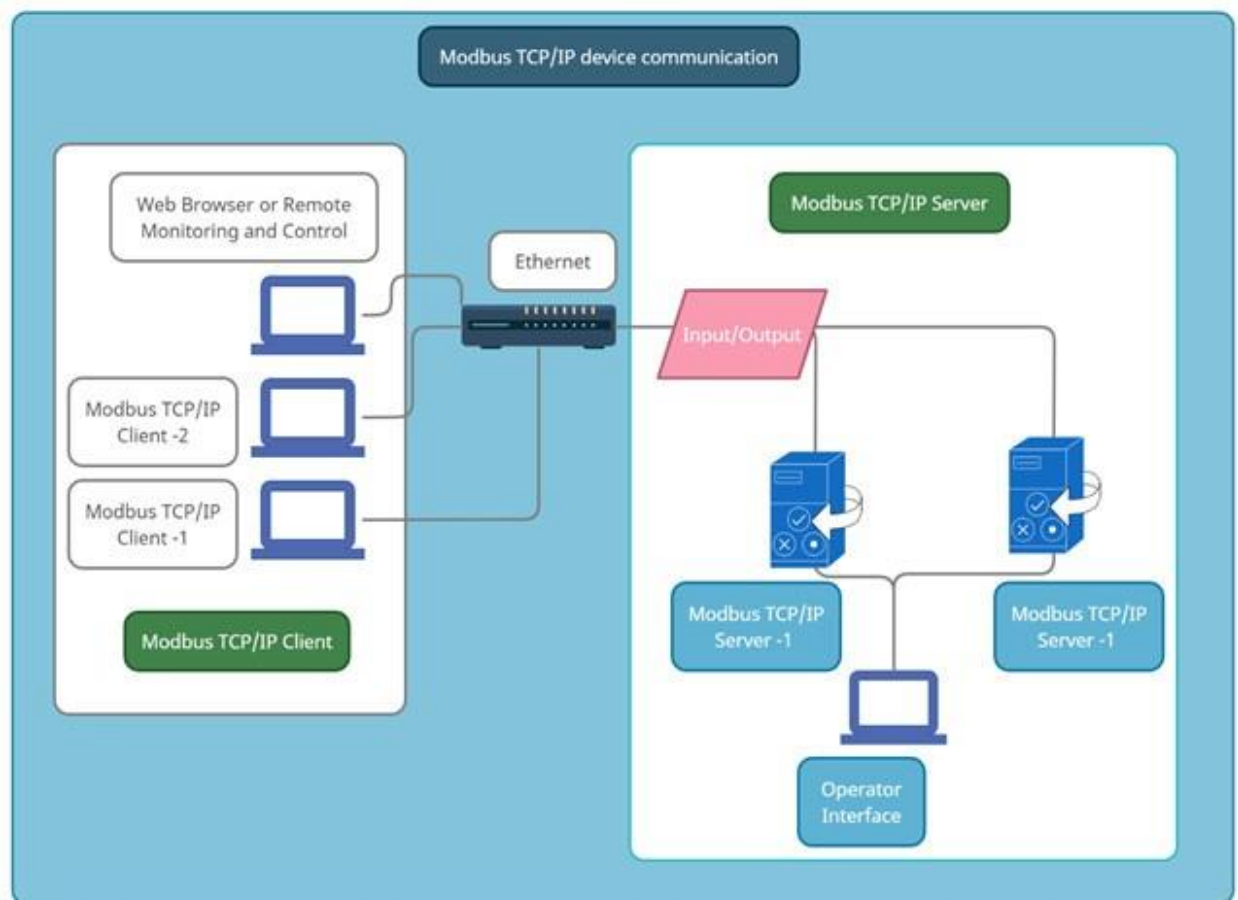
## 2. Modbus TCP/IP

### Overview:

Modbus TCP/IP (also Modbus-TCP) is simply the Modbus RTU protocol with a TCP interface that runs on Ethernet. That is, Modbus TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (Modbus as the application protocol).

TCP/IP refers to the Transmission Control Protocol and Internet Protocol, which provides the transmission medium for Modbus TCP/IP messaging.

TCP/IP allows blocks of binary data to be exchanged between computers. It is also a worldwide standard that serves as the foundation for the World Wide Web. The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed.



## CIM 10 device support MODBUS TCP/IP Features

- Supports multiple types of equipment via IP addressing.
- Supports adjustable address base (0 or 1)
- Supports full address range (0-65535)
- Supports word and byte swapping (byte order):

MSW: Most significant word first

LSW: Least significant word first

MSB: Most significant byte first

LSB: Least significant byte first

- Supports equipment slave id full range (1-254)

- Supported functions:

read coil status (01);

read input status (02);

read holding registers (03);

read input registers (04);

force single-coil (05);

force multiple coils (15);

preset multiple registers (16);

Connection status

- Supported data types;

Boolean

Integer8

Integer16

Integer32

Unsigned8

Unsigned16

Unsigned32

Floating Point 32

- Support for reading/writing data that spans multiple contiguous registers with different sizes and byte orders.
- Adjustable polling request time per equipment.
- Adjustable minimum request interval per register.
- Adjustable pooling request timeout.
- Supports minimum channel silence, forcing a time between every request on the serial bus.
- Allows different communication options (baud rate, byte size, parity and stop bits) on the same serial bus

## Pre-requisites:

- Modbus Ethernet TCP/IP Simulator or Modbus TCP/IP device
- Whitelist IP address and Port.

**Here we are using Modbus Ethernet Simulator to test the device data on webUI.**

## Connection Setup:

- Open CIM 10 device web page with provided URL by CIMCON support team (192.168.3.100) and webpage look likeas below. Default configuration is as below.

**IP:** 192.168.3.100

**Subnet :**255.255.255.0

**Gateway:**192.168.3.1



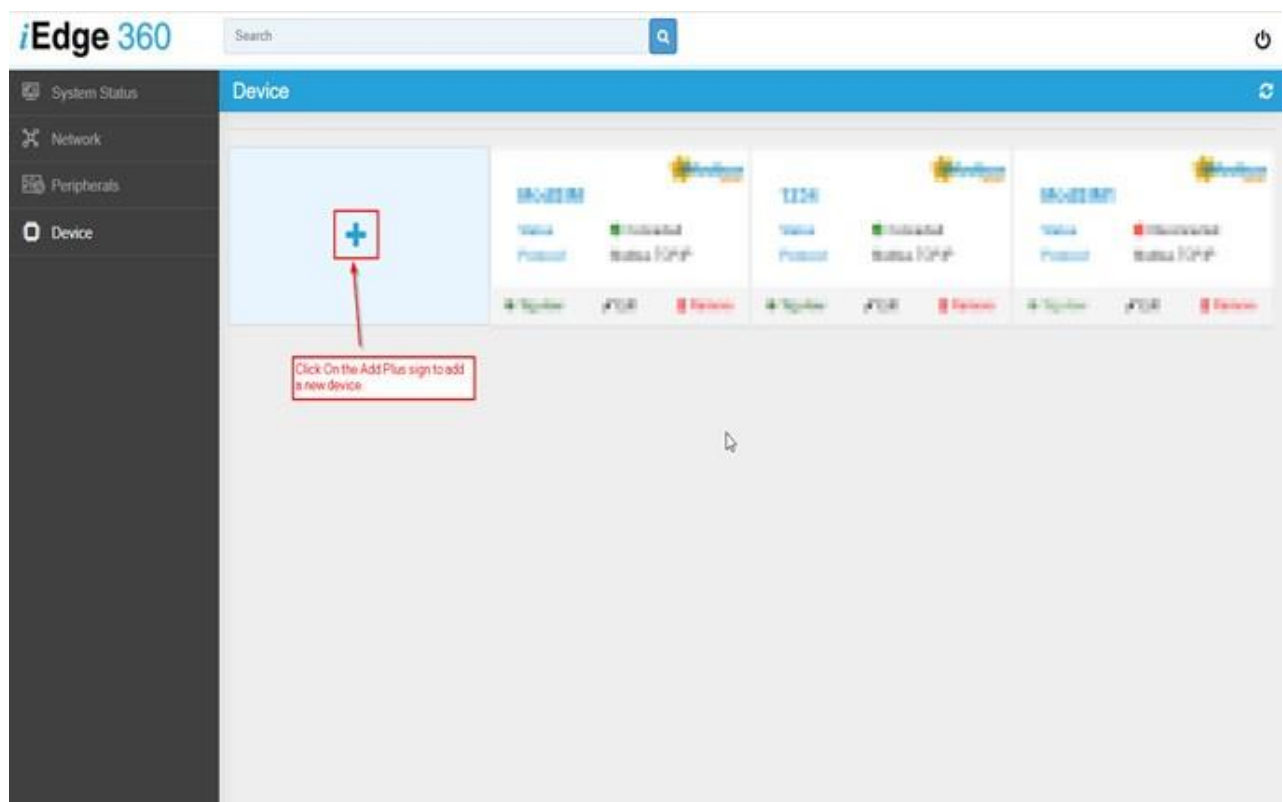
- Login with Valid Username and Password.

User Name: iEdgeAdmin

Password: iEA@12345.



- Connection Setup on webUI:
- We can add the new device using + sign as like in the below screenshot.





- After click on the add device option, we will get a pop-up window on which we can select the protocol on which we want to create a device and give a proper name for that device. Then click on the “Save” button to see the device on the dashboard.



**The device will be created like as below Image. It contains many details which as below:**

- Device Name. (Give a proper name to the device)
- Device Status. (Connected, Disconnected, Unknown)
- Tag View. (It is reflecting live data of the device after configuration)
- Edit. (Add configuration of the device)
- Remove. (To remove/delete the device from the dashboard)



**We use the “edit” option to configure a device. It requires below details:**

- IP Address of the Simulator or the Modbus TCP/IP server.
- Port.
- Port timeout in second.
- The Above basic configuration would help in establishing a connection between Modbus TCP clients to Modbus TCP server.

Modbus TCP/IP

Device

Protocol

Modbus TCP/IP

Name

MOD\_SIM

Destination

CIM1

+ Add Writable Tag

Port

Port Select

Serial

TCP IP

MAC

IP

199.199.50.151

Port

502

Port timeout (Sec)

3

Device Template

Please Select

+ Add Query

Save

Add Writable Tag

- webUI support to write a query on the physical device connected to CIM 10 device. Users can check the peripheral permission, to check writable option is allowed ornot.

Add Writable Tag

Slave ID

0

Address

0

Input

Coil Input

Name

Name

Type

Boolean

Value

Enable

Disable

Save

- Below are detail need to be added for updating the writable tag.

Slave ID	1 to 254	Address	1 to 65535
Input	Coil input	Name	User have configurable name option
	Holding single register		
	Holding multiple register		
Type	Boolean	Value	Enable or Disable (For Coil input)
	Int16		
	Unit16		

## Configuration:

We can configure the device using below details:

- First, we have to create a query on our device

The details to configure a single query would be as below:

### Communication Protocol Modbus-TCP:

- **Slave Id. (Device Id of the peripheral. (max 255))**
- **Input.**

Coil Input	Single bit	Read-Write	This type of data would alterable by an application.
Input Status	Single bit	Read-Only	This type of data would be provided by the I/O system only.
Input Register	16-bit word	Read-Only	This type of data would be provided by the I/O system only.
Holding Register	16-bit word	Read-Write	This type of data would alterable by an application.

- **Address Field. (Start Address - First register address to be read (01....F7 HEX))**
- **No. of Registers. (Number of registers (max 32) to be read)**
- **Interval. (Interval in second)**

- **Peripheral ID.** (Number mentioned on “CIMCON iCLOUD” would be connected and receive the data)
- **Peripheral Name.** (Display name of the peripheral on CIMCON iCLOUD application)
- **Query Interval.** (Ms)

**Modbus TCP/IP**

+ Add Query

**Query**

Remove Query

Slave ID: 1 Interval (Sec): 15

Input: Holding Register Peripheral ID: 4210

Address: 0 Peripheral Name: test

No. of Register: 10 Query Interval (Ms): 500

**Parameters**

+ Add Parameter

Data type	Factor	Converter	Name	Remove
-----------	--------	-----------	------	--------

Save

We can add the parameters as below Image:

- **Data type.**
  - o Float
  - o Long
  - o Integer
  - o Binary
  - o Character
  - o Decimal
- **Factor (Max 999999)**
- **Converter.**
  - o Float
  - o Swapped Float
  - o Long

- o Hex
- o None

- **Name (Parameter Name as per device manual)**

Modbus TCP/IP

Query

Remove Query

Slave ID

1

Interval (Sec)

15

Input

Holding Register

Peripheral ID

4210

Address

0

Peripheral Name

test

No. of Register

10

Query Interval(Ms)

500

Parameters

Add Parameter

Data type	Factor	Converter	Name	Remove
Float	1	Swapped Float	T1	

Save

**Note:**

We can add a maximum of 32 queries and 16 parameters each query-wise.

**Tag View:**

- It will display output as per configuration.

MOD\_SIM1

Status

Connected

Protocol

Modbus TCP/IP

Tag view

Edit

Remove

Tags - MOD\_SIM1

test

Name	Value	Type	Timestamp	Status
T1	45	Decimal	2022/03/07 06:17:01	✓
T2	68	Decimal	2022/03/07 06:17:01	✓
T3	0	Decimal	2022/03/07 06:17:01	✓
T4	0	Decimal	2022/03/07 06:17:01	✓
T5	126	Decimal	2022/03/07 06:17:01	✓

Close

## 3. OPC UA

### Overview:

- OPC UA (short for Open Platform Communications Unified Architecture) is a data exchange standard for industrial communication (machine-to-machine or PC-to-machine communication). This open interface standard is independent of the manufacturer or system supplier of the application, of the programming language in which the respective software was programmed, and of the operating system on which the application is running. OPC UA is a more secure, open, reliable mechanism for transferring information between Servers and Clients. It provides more open transports, better security, and a more complete information model. OPC UA provides a very flexible and adaptable mechanism for moving data between enterprise-type systems and the kinds of controls, monitoring devices, and sensors that interact with real-world data.
- OPC UA uses scalable platforms, multiple security models, multiple transport layers, and a sophisticated information model to allow the smallest dedicated controller to freely interact with complex, high-end server applications. OPC UA can communicate anything from simple downtime status to massive amounts of highly complex plant-wide information.



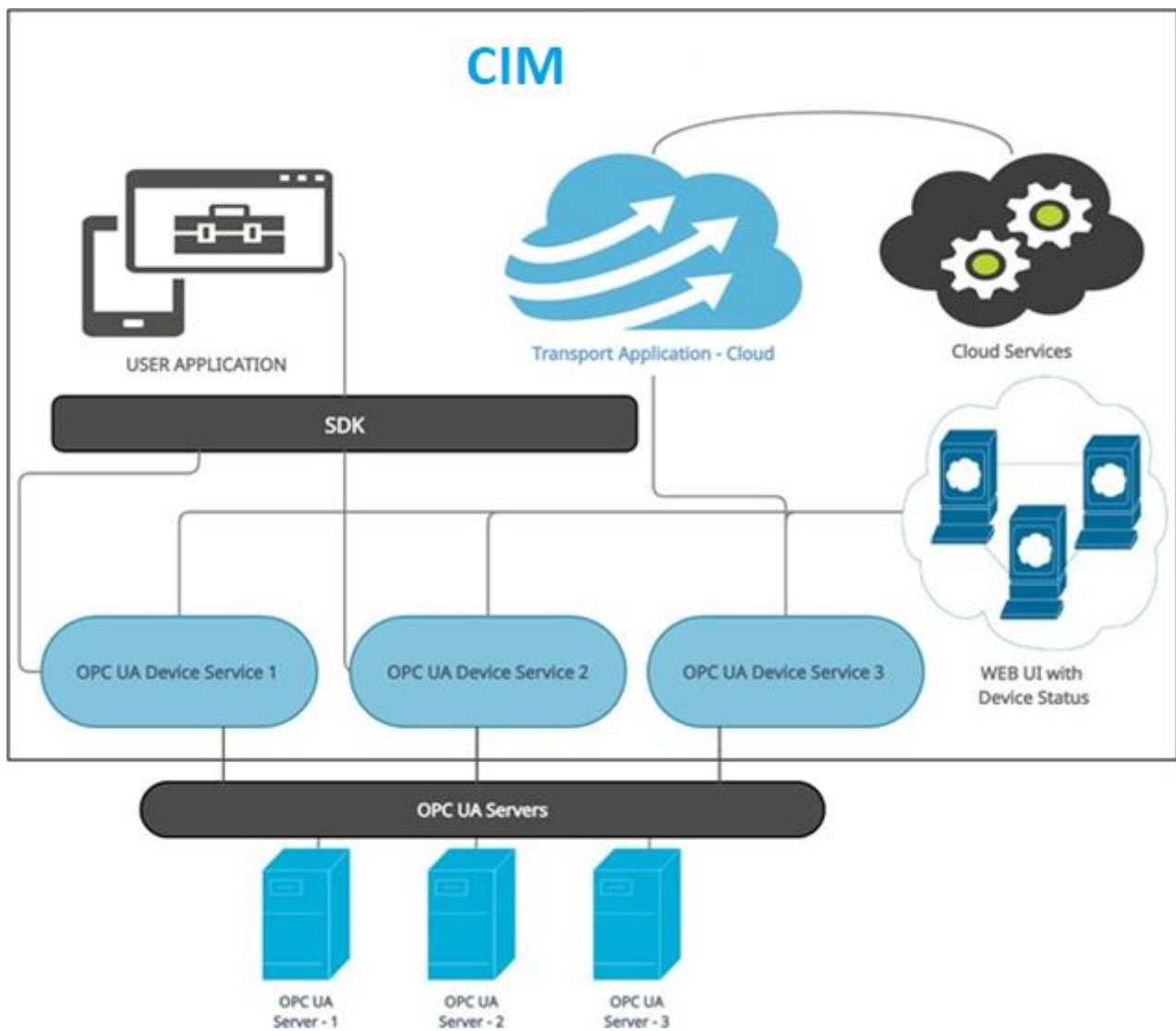


Figure 1: CIM10 device – OPC UA Block Diagram

## CIM 10 device OPC UA Features

- Support for the OPC UA protocol over TCP on VPNs, through firewalls, and across the internet, WAN, or LAN.
- Support for reading and writing OPC UA variable nodes by node ID.
- Support for multiple OPC UA server connections.
- Support for basic authentication.
- Support for authentication through x509 certificates.
- Support for data encryption via RSA Standards.
- Supports Read, Write, and Subscription functionality.
- Adjustable pooling read time.

- Supported data types:

BOOL (boolean)

SINT (int 8)

USINT (uint 8)

BYTE (uint 8)

INT (int 16)

UINT (uint 16)

WORD (uint 16)

DINT (int 32)

UDINT (uint 32)

DWORD (uint 32)

LINT (int 64)

ULINT (uint 64)

LWORD (uint 64)

REAL (float)

LREAL (double)

CHAR (char)

WCHAR (2-byte char)

STRING (utf-8)

## **Pre-Requisite:**

To verify the functionality of the OPC UA application, we need OPC UA Server and client.

1. OPC UA Server (Device) or OPC UA Simulation Server
2. OPC UA Client (CIM 10 device)
3. Internet Connectivity
4. Whitelist IP/Port

## OPC UA Client/Server Configuration:

(For OPC UA Client server configuration we have taken Prosys Simulator as an example)  
Here, we have provided complete details of the OPC UA server – Client configuration with Prosys OPC UA simulation server.

### Server Side

- First of all, we need to install the Prosys OPC UA Simulation Server into the PC, below is the link to download “Prosys OPC UA Simulation Server” for your ready reference.

**<https://downloads.prosysopc.com/opc-ua-simulation-server-downloads.php>**

***Note: The application is a “self-contained” package that contains a private JRE for running the application, so no Java install is required.***

- After successfully installation, the status View (Figure 1) displays short information about the current server status and available connection addresses. If everything is fine, Server Status displays Running. During startup, the status will show a message displaying the current startup phase. In case of errors, it may turn to error with additional information about the exact problem. The application can fail at startup, for example when another instance of the application is already running.

**This is below an example of a Prosys simulation server.**



Figure 1. The Status View displays some basic information about the server.

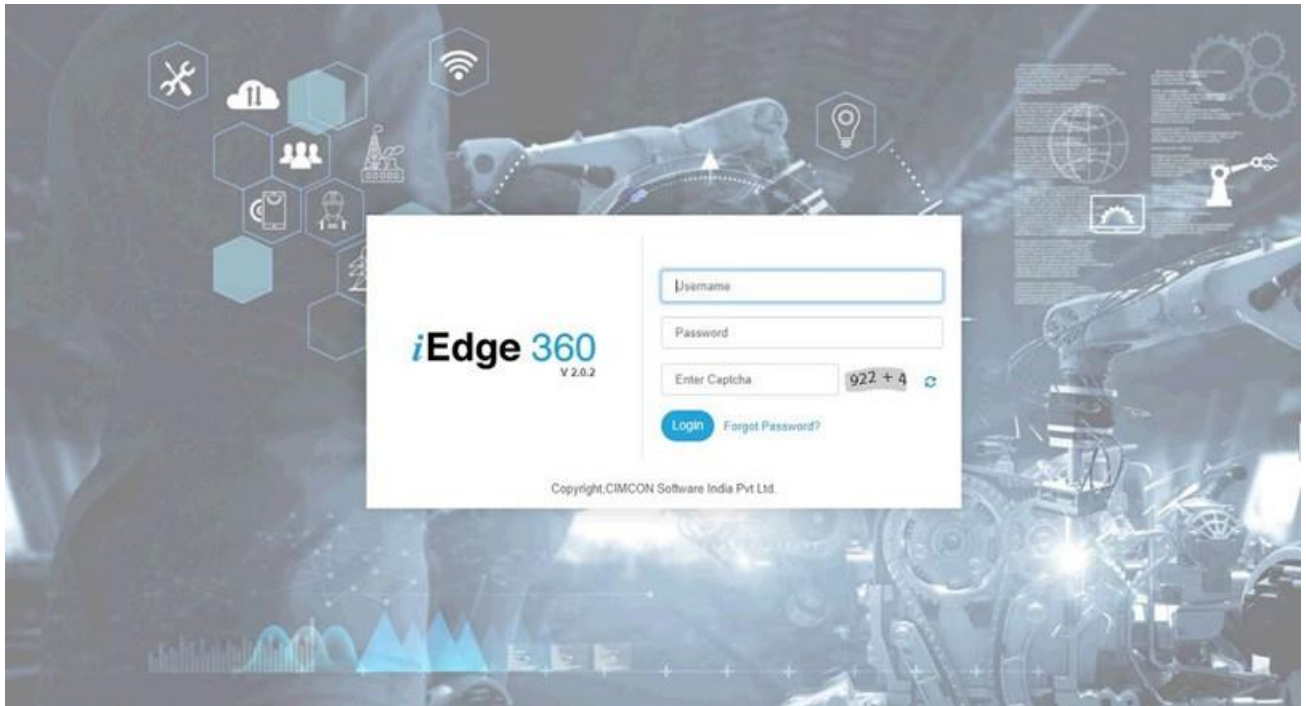
## Client-Side:

- Open webUI web page with provided URL by CIMCON support team (192.168.3.100) and webpage look like as below. Default configuration is as below.

**IP:** 192.168.3.100

**Subnet :**255.255.255.0

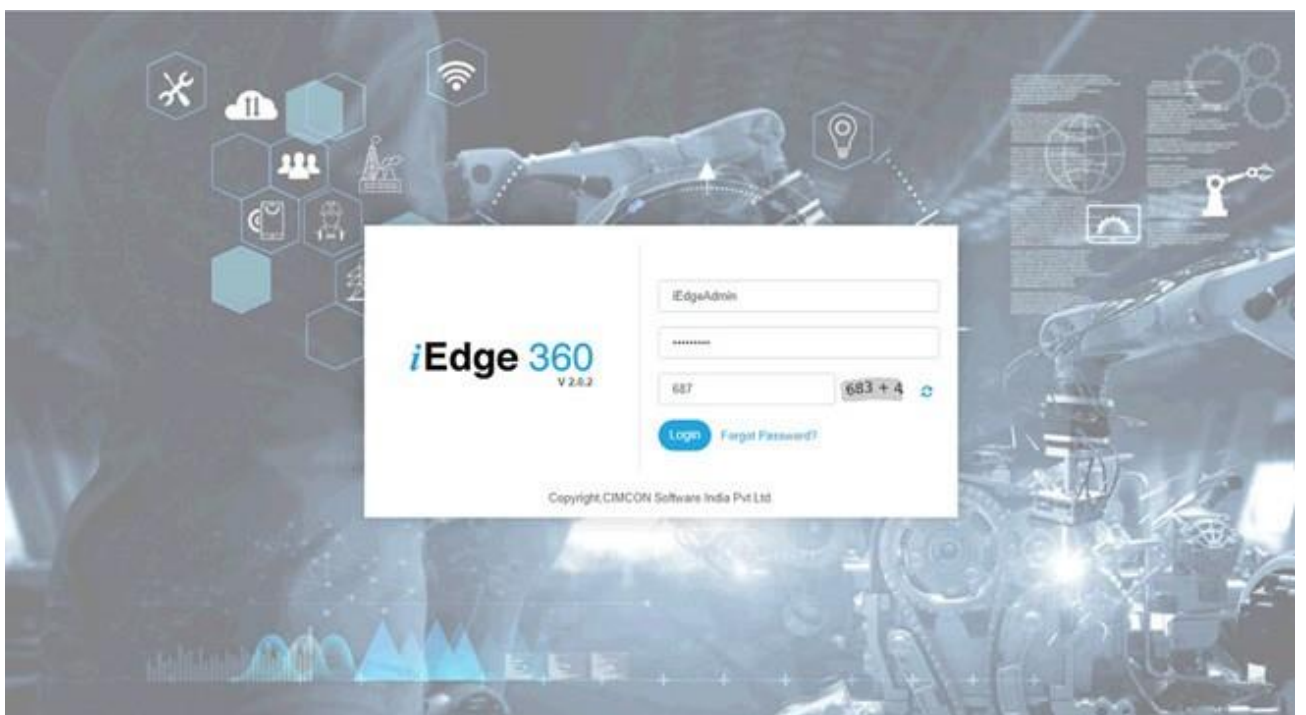
**Gateway:**192.168.3.1



- Login with Valid Username and Password.

User Name: iEdgeAdmin

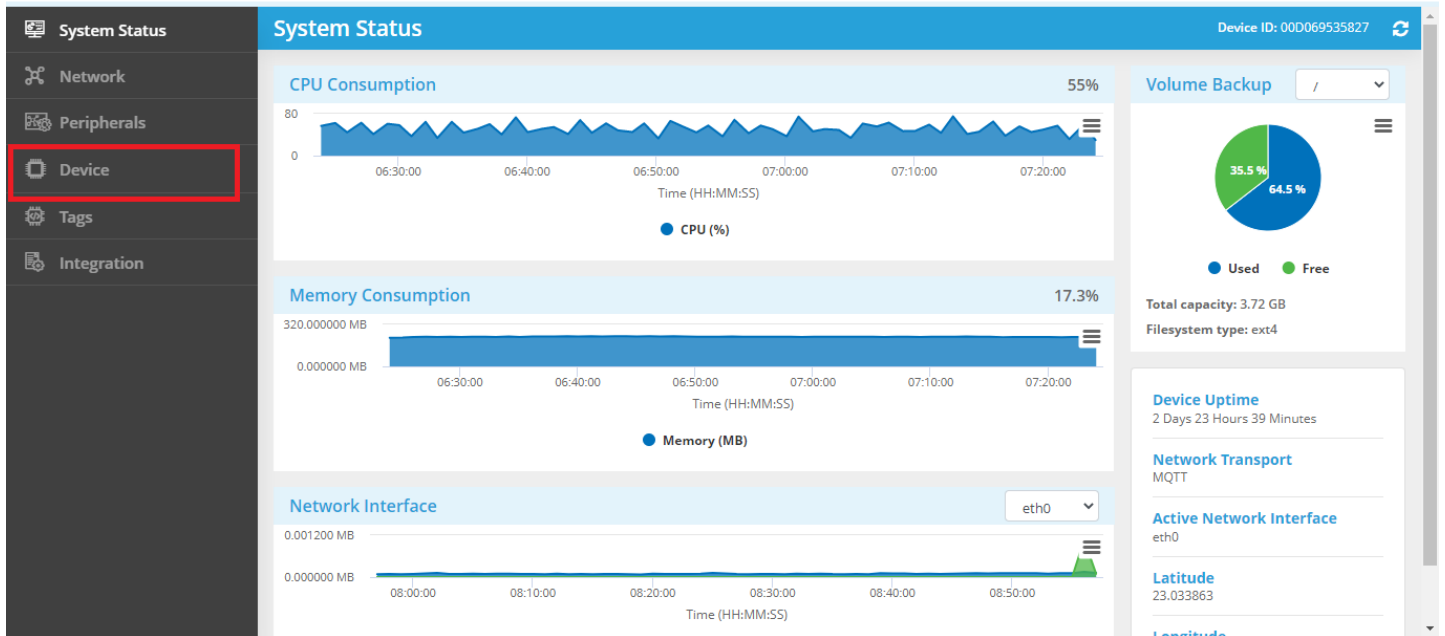
Password: iEA@12345



- After successfully login in, you can able see the status page of the application as below.

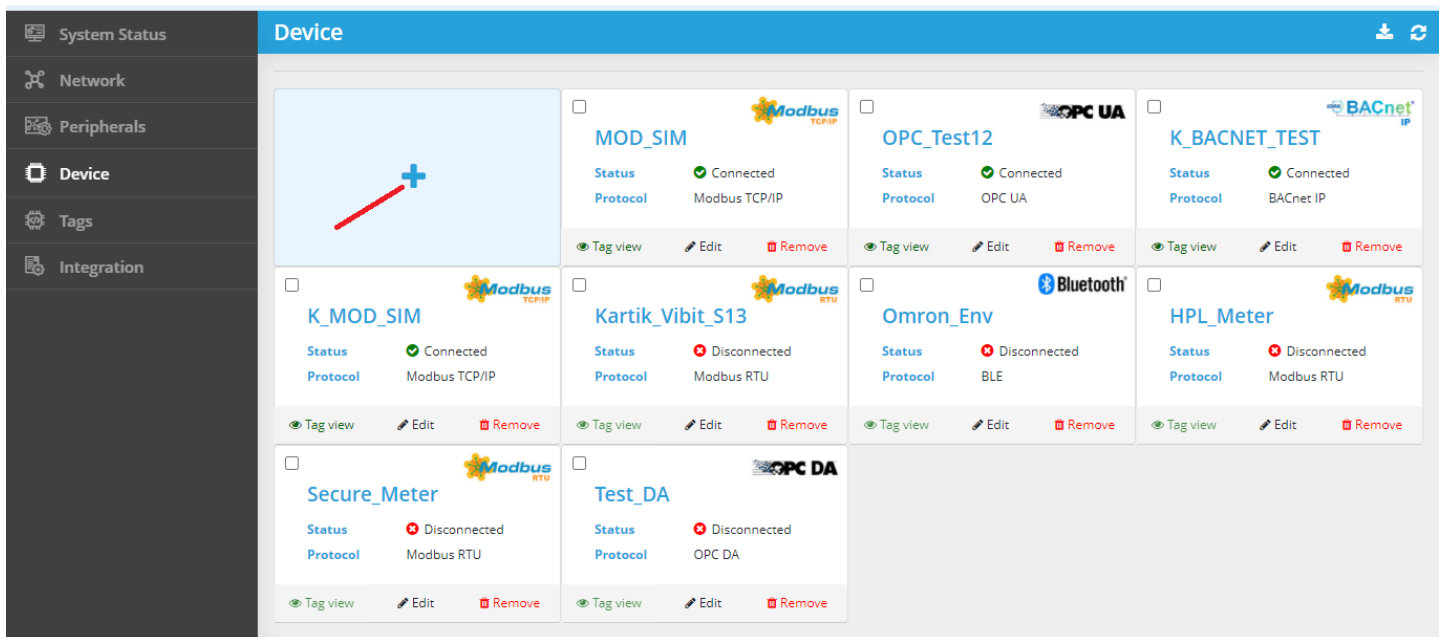
On the left side of the status page, there is an option “Device”, click on the “Device”.

## iEdge 360

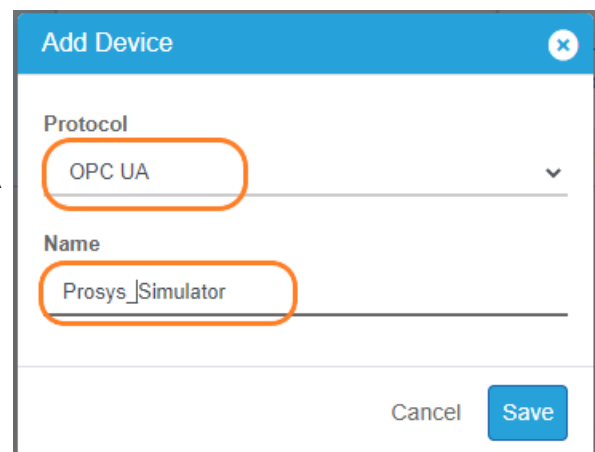


- You can able to see the below page once you clicked on “Device”. Now, add the device of “OPC UA” by clicking on “+”.

## iEdge 360

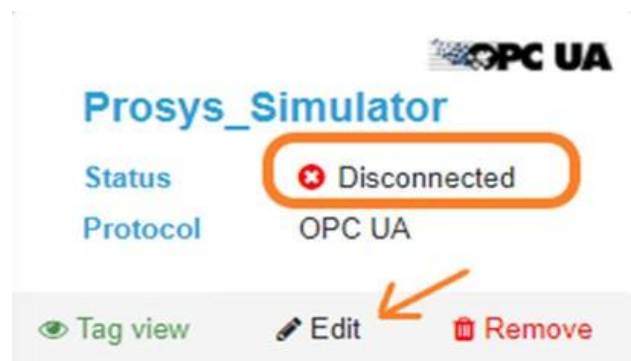


- Select "OPC UA" protocol to add an OPC UA device, configured the Name of the OPC UA device, and click on "Save" to create an OPC UA device.



The 'Add Device' dialog box shows the 'Protocol' dropdown menu set to 'OPC UA' and the 'Name' text field containing 'Prosys\_Simulator'. Both the dropdown and the text field are highlighted with orange rectangular boxes. At the bottom right, there are 'Cancel' and 'Save' buttons.

- Initially, created OPC UA device status shows "Disconnected" as shown below image.



- Click on "Edit" to configure the OPC UA Server URL on the client side. Now, you can see below the configuration page of OPC UA and here, configured "IP" of the PC where simulator server is installed and Prosys OPC UA Simulation Server Connection Address in URL and click on "Save" as shown in below image.

OPC UA

Device

Protocol

OPC UA

Name

OPC\_Yogesh

Sampling rate (Sec)

10

Destination

None selected

Port

Port Select

Serial

TCP IP

MAC

IP

199.199.50.151

Port

5252

Port timeout (Sec)

3

URL

opc.tcp://199.199.50.151:5252/OPCUA/SimulationServer

Save



- Now, device status has been changed to "Connected" as shown below image, which means OPC UA Server and Client connection set-up done successfully.



## Parameter Configuration:

- Starting with the configuration page of the OPC UA application can able to see the configuration page as below.

The image shows a configuration window titled 'OPC UA'. It contains several fields and sections:
 

- Protocol:** A dropdown menu showing 'OPC UA'.
- Name:** A text field containing 'OPC\_Yogesh'.
- Sampling rate (Sec):** A text field containing '10'.
- Destination:** A dropdown menu showing 'None selected'.
- Port Section:**
  - Port Select:** Radio buttons for 'Serial', 'TCP IP' (selected), and 'MAC'.
  - IP:** A text field containing '199.199.50.151'.
  - Port:** A text field containing '5252'.
  - Port timeout (Sec):** A text field containing '3'.
- URL:** A text field containing 'opc.tcp://199.199.50.151:5252/OPCUA/SimulationServer'.
- Security Mode:** A dropdown menu showing 'Sign & Encrypt'.
- Security Policy:** A dropdown menu showing 'Auto'.
- Save Button:** A blue button labeled 'Save' at the bottom right.

## Device

- **Protocol:** OPC UA which is already selected when created device.
- **Name:** Already given the relevant name of when created device.
- **Sampling rate (Sec):** user can define sampling rate in second (0 to 3600), at which rate the user requires information/data on the cloud.
- **Destination:** user can define/add the list of applications on which the user wants data.

## Port

- **Port Select:** Serial or TCP IP
- **IP & Port:** user needs to configure the IP and port of the PC/Server.

## URL

- **URL:** Need to configure URL of the OPC UA simulation server/Device which is an OPC UA server.

## “+ Add Writable Tag”

- Users can add a tag by providing the below details.

The screenshot displays the OPC UA configuration window. The 'Device' section is active, showing 'Protocol' as OPC UA, 'Name' as Prosys, 'Sampling rate (Sec)' as 10, and 'Destination' as service\_transport. A red box highlights the '+ Add Writable Tag' button in the top right corner. A modal dialog box titled 'Add Writable Tag' is open in the center, containing the following fields:

Field	Value
Name Space Index	0
Type	Boolean
Identifier	Identifier
Identifier Type	Numeric
Name	Name
Value	<input checked="" type="radio"/> Enable <input type="radio"/> Disable

At the bottom right of the dialog is a 'Save' button. The background window also shows a 'Port' section with 'Port Select' set to TCP and 'IP' field, and a 'URL' section with the address 'opc.tcp://199.199.51.43:53530/OPCUA/SimulationServer'. At the bottom, 'Security Mode' is set to 'Sign & Encrypt' and 'Security Policy' is set to 'Auto'. A 'Save' button is also present at the bottom right of the main window.

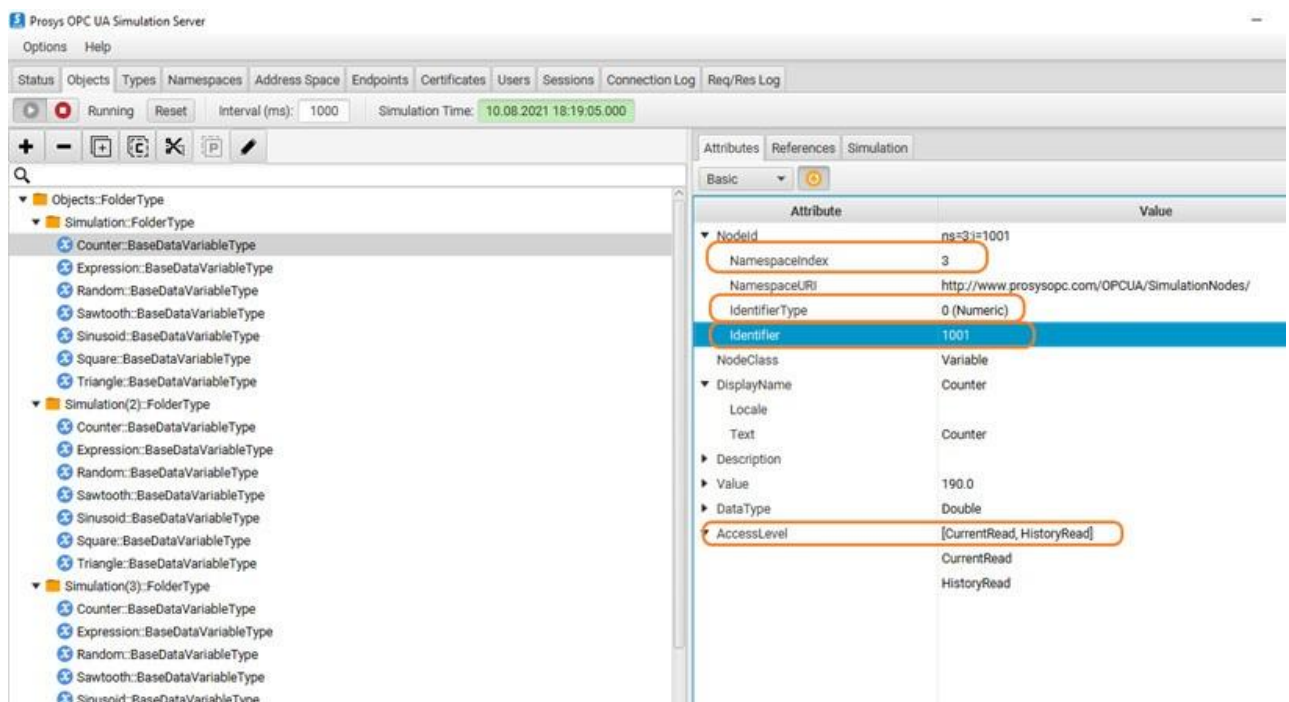
**When adding a new Variable or Object or when editing an existing one, you will be prompted to fill all or some of the following information**

- **Name Space Index** defines the namespace that the Node belongs to
  - o Name Space Index : **0 to 99 (numeric)**
- **Identifier Type** defines which Variable Type or Object Type the Node will implement. The type defines the structure of the Node.
  - o Identifier Type : **0 (Numeric) or 3 (string)**
- **Identifier** defines the name or number of that particular variable/node.
  - o Identifier: **name or number of Node**

- **Name:** defines the Display Name and the Browse Name of the Node
  - o User can give name as per requirement
- **Type:** defines the Datatypes.
  - o Users can select the data type from the drop-down list

**Note:** User can get the variable or object details from Prosys simulator server as shown in the below image.

Object -> Select Variable (Counter) -> Attribute -> Node ID



## Security

URL:

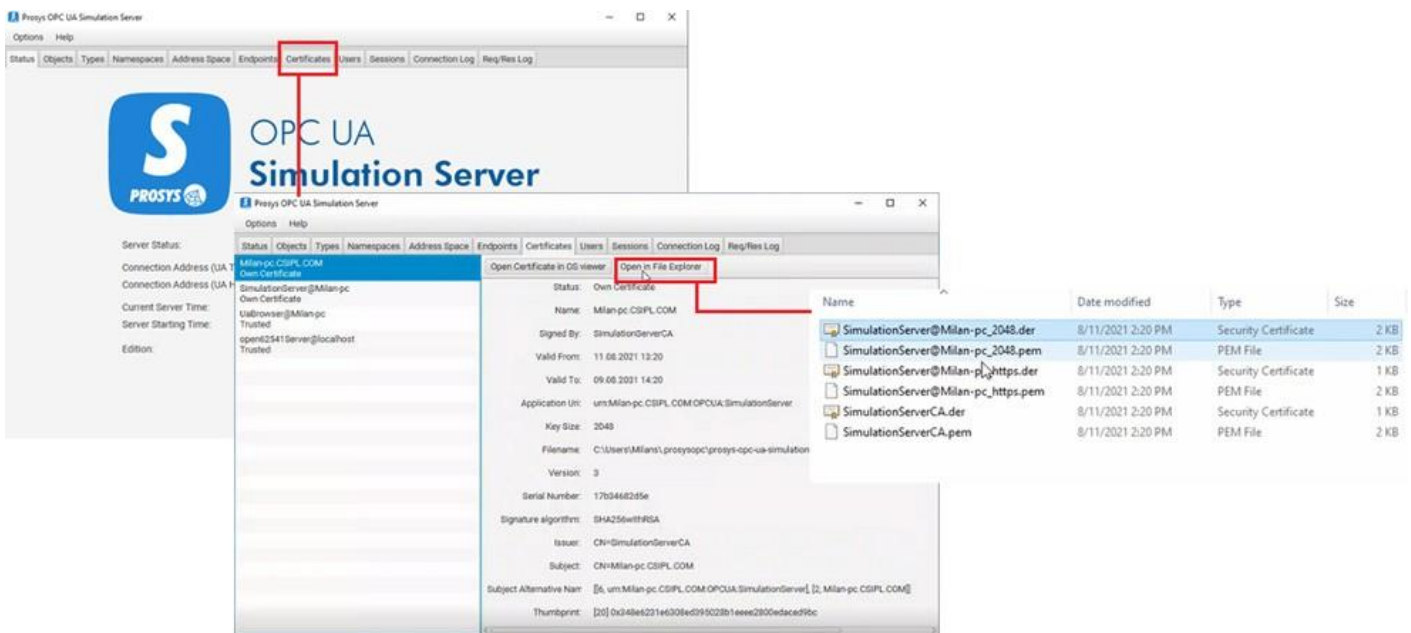
Security Mode:  Security Policy:

Authentication: ☒ Anonymous ☐ Username & Password ☒ Certificate

Server Certificate:  Own Certificate:  Own Key:

ProsysSimulationServer.der server\_cert.der server\_key.der

- **Security Mode:** User need to select security mode as per drop down mentioned
  - a) **None**
  - b) **Sign**
  - c) **Sign & Encrypt**
- **Security Policy:** User needs to select security policy as per drop down mentioned.
  - a) **Auto**
  - b) **Basic128Rsa15**
  - c) **Basic256**
  - d) **Basic256Sha256**
  - e) **Aes128Sha256RsaOaep**
- **Authentication** – User can choose the authentication mode
  - a) **Anonymous** (No login credential is required)
  - b) **Username & Password** (Login credential is required)
  - c) **Certificate**
- **Server Certificate:** we can get a certificate from the OPC server. Example as per below.



- **Own certificate and Own key:** The certificate and key are of the CIM 10 device. CIMCON CS team will provide the file.

## Configuring a Variable or Object

- Apart from this, the OPC UA application has main three parts in the configuration section as shown below image.
- **“Data Access” – read and write data/information based on access permissions.**
- **“Subscriber” – monitor data/information and report-by-exception when values change based on a client’s criteria.**
- **“Events” – notify important information based on the client’s criteria.**

The screenshot shows the OPC UA configuration window with three main sections: Data Access, Subscriber, and Event. Each section has a table with columns: Name Space Index, Identifier Type, Identifier, Name, Peripheral ID, and Remove. There are also buttons to add new entries for each section.

Name Space Index	Identifier Type	Identifier	Name	Peripheral ID	Remove
3	Numeric	1001	Counter	1	
3	Numeric	1002	Random	1	

Name Space Index	Identifier Type	Identifier	Name	Peripheral ID	Remove
3	Numeric	1005	Sqaure	1	

Name Space Index	Identifier Type	Identifier	Name	Peripheral ID	Remove
6	String	MyObjectsFolder	MyObjectsFolder	1	

Save

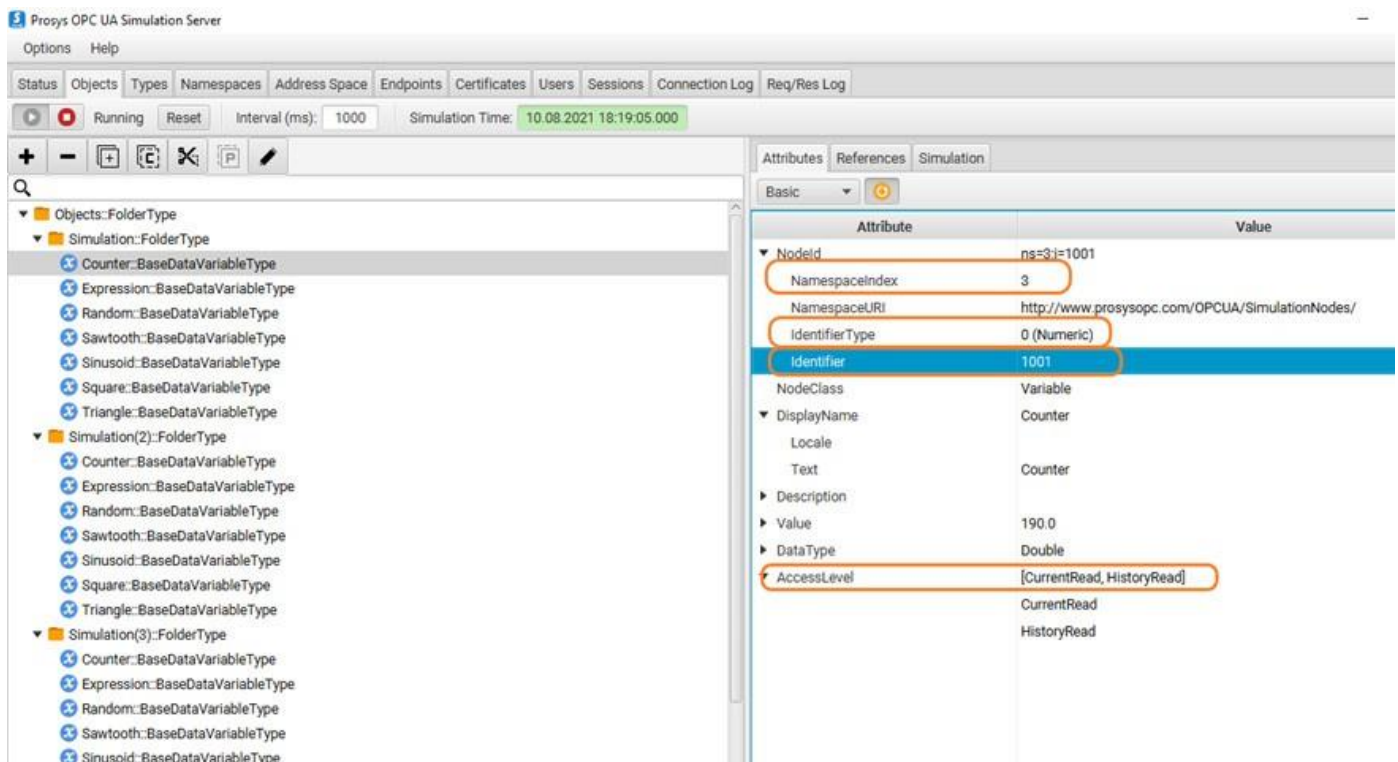
When adding a new Variable or Object or when editing an existing one, you will be prompted to fill all or some of the following information

- **Name Space Index defines the namespace that the Node belongs to**
  - o **Name Space Index** : 0 to 99 (numeric)
- **Identifier Type defines which Variable Type or Object Type the Node will implement. The type defines the structure of the Node.**
  - o **Identifier Type** : 0 (Numeric) or 3 (string)
- **Identifier defines the name or number of that particular variable/node.**
  - o **Identifier**: name or number of Node

- **Name:** defines the Display Name and the Browse Name of the Node
  - User can give name as per requirement
- **Peripheral ID:** defines the ID number through which you'll get the data on CIMCON iCLOUD.
  - Users can define peripheral ID as per requirement

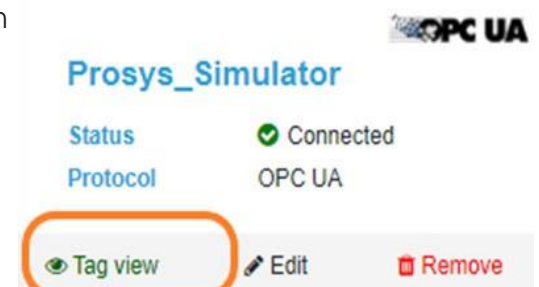
**Note:** User can get the variable or object details from Prosys simulator server as shown in the below image.

Object -> Select Variable (Counter) -> Attribute -> Node ID



**After configuring a variable or object on the configuration page of OPC UA, click on "Save".**

Tag View: To see the value of configured parameters on the configuration page, click on "Tag View" as shown below image and you'll get the value of the configured parameters along with its data type, timestamp, and status.



Tags - Prosys\_Simulator



## Data Access

Name	Value	Type	Timestamp	Status
Counter	60.0	Double	2021/08/10 13:00:40	✓
Random	59188350	UInt64	2021/08/10 13:00:40	✓

## Subscribe

Name	Value	Type	Timestamp	Status
Sqaure	1000	UInt32	2021/08/10 13:00:49	✓

## Event

Name	Value	Type	Timestamp	Status
MyObjectsFolder	Level exceeded	String	2021/08/10 13:00:13	✓

Close



## 4. OPC DA

### Overview:

- OPC XML DA is a SOAP Web Service based mechanism to communicate with OPC DA server running on Windows operating systems. This mechanism is a standard from the OPC Foundation and therefore helps a client process to communicate with a OPC DA server from any software vendor.
- A OPC XML DA client connects to a OPC XML DA gateway. The communication protocol is HTTP and the data is sent and received in SOAP XML packets. The SOAP XML packet structures in this case is defined by the OPC Foundation.
- The communication between a OPC XML DA gateway and a OPC DA server is through usual Windows DCOM components.
- For the purpose of CIM 10 device a UNIX daemon process shall act as a OPC XML DA client.

### Pre-Requisite:

To verify the functionality of the OPC DA application, we need OPC DA Server and client.

- OPC DA Server (Device) or OPC DA Simulation Server
- OPC DA Client (CIM 10 device)
- Installed gateway running during communication between client and server.
- Internet Connectivity
- Whitelist IP/Port

### OPC DA Client/Server Configuration:

#### Server Side

(For OPC DA Client server configuration we have taken MitrokonOPC simulator as an example) Here, we have provided complete details of the OPC UA server – Client configuration with MitrokonOPC simulator simulation server.

#### Server Side

- First of all, we need to install the Simulation Server into the PC i.e. MatrikonOPCSimulation.exe , below is the link to download “MatrikonOPCSimulation” for your ready reference. <https://www.matrikonopc.com/downloads/178/index.aspx>  
***Note: The application is a “self-contained” package that contains a private JRE for running the application, so no Java install is required.***
- After successfully installation, the status View displays short information about the current server status and available connection addresses.



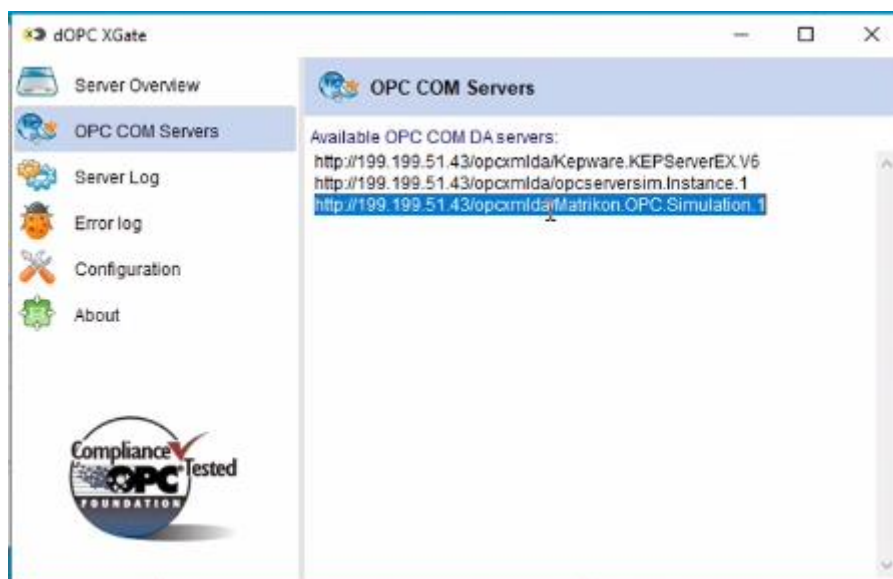
This is below an example of a MatrikonOPC simulation server.



Figure 1. The Status View displays some basic information about the server.

- Gateway installed must be running on server PC for communication. Below is the screen showing server IP address which will be as URL link during device creation.

<https://www.kassl.de/opc/download.shtml>



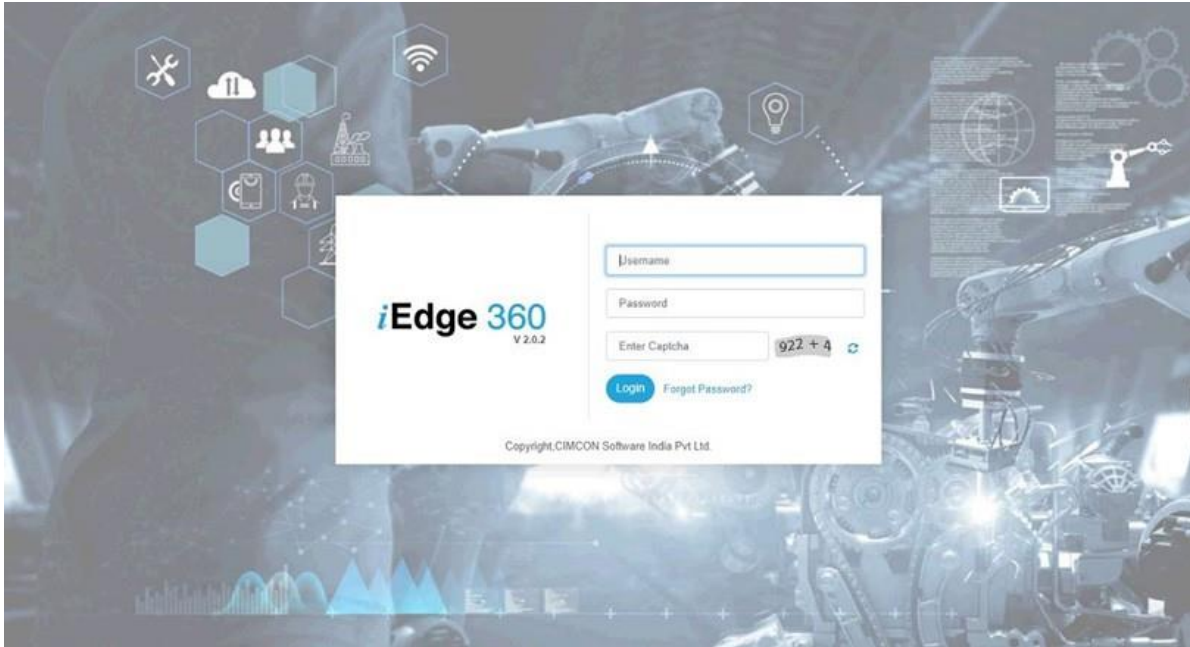
## Client-Side:

- Open webUI web page with provided URL by CIMCON support team (192.168.3.100) and webpage look like as below. Default configuration is as below.

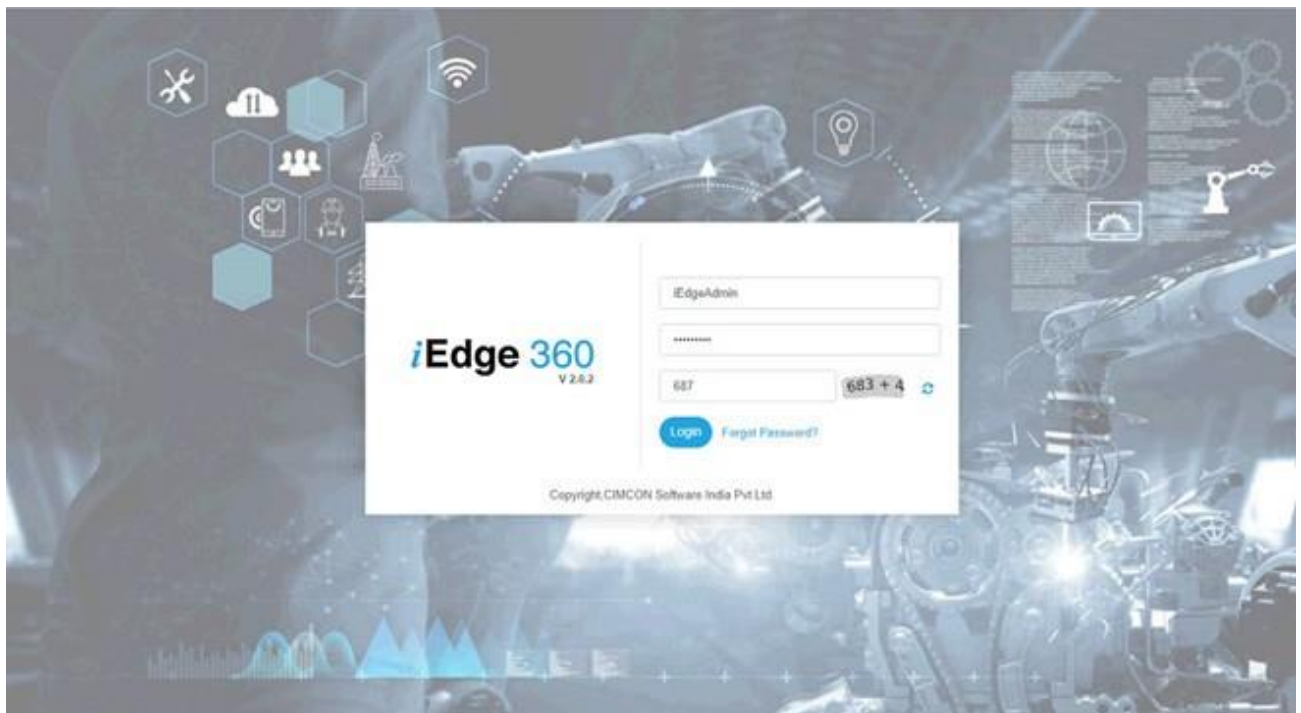
**IP:** 192.168.3.100

**Subnet :**255.255.255.0

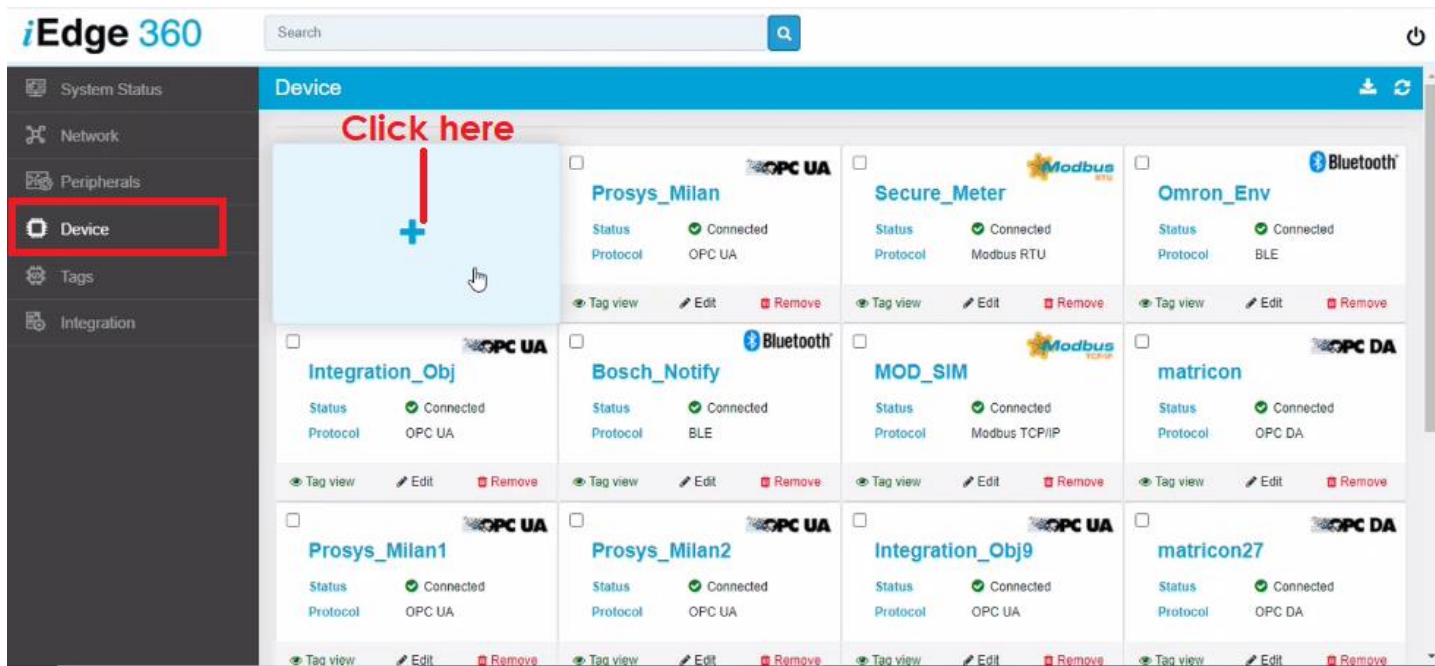
**Gateway:**192.168.3.1



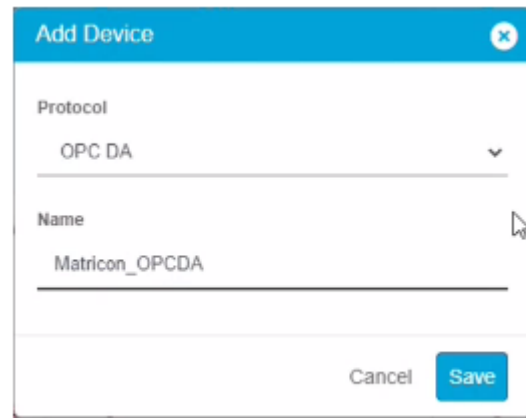
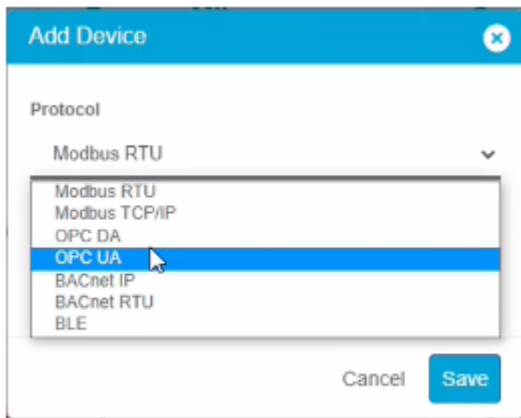
- Login with Valid Username and Password.  
User Name: iEdgeAdmin Password: iEA@12345



- User need to go to Device, click "+" sign to add new device.

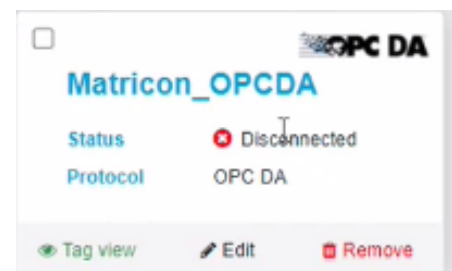


- After click on the add device option, we will get a pop-up window on which we can select the protocol on which we want to create a device and give a proper name for that device. Then click on the "Save" button to see the device on the dashboard.



**The device will be created like as below Image. It contains many details which as below**

- Device Name. (Give a proper name to the device)
- Device Status. (Connected, Disconnected, Unknown)
- Tag View. (It is reflecting live data of the device after configuration)
- Edit. (Add configuration of the device)
- Remove. (To remove/delete the device from the



dashboard)

- Initially, created OPC UA device status shows “Disconnected” as shown below image.



- Click on “Edit” to configure the OPC DA Server URL on the client side. Now, you can see below the configuration page of OPC DA and here, configured “IP” of the PC where simulator server is installed and MatriconOPC Simulation Server Connection Address in URL and click on “Save” as shown in below image

The image shows the 'OPC DA' configuration window. It has a blue header bar with the title 'OPC DA'. The main content area is divided into sections:
 

- Device**: Contains 'Protocol' (set to 'OPC DA'), 'Name' (set to 'Matricon\_OPDA'), 'Sampling rate (Min)' (set to '1'), and 'Destination' (set to 'Matricon OPC', with a dropdown menu showing options like 'xyz', 'test3', 'Test4', and 'Matricon\_OPC').
- Port**: Contains 'Port Select' (with radio buttons for 'Serial', 'TCP IP' (selected), and 'MAC'), 'IP' (set to '199.199.51.43', highlighted with a red box), 'Port' (set to '80'), and 'Port timeout (Sec)' (set to '5'). A red error message box states 'Port timeout must be between 1 to 99'.
- URL**: A text field containing 'http://199.199.51.43/opcxmlda/Matricon.OPC.Simulation.1', highlighted with a red box.
- Tags**: A section with a '+ Add Tags' button and a table with columns: 'Edge Name', 'OPC Name', 'Peripheral ID', and 'Remove'.

 A 'Save' button is located at the bottom right of the window.

## Under OPC DA

### Device

**Protocol** - Already selected during device creation step.

**Name** – Already configured during device creation step.

**Sampling rate (MIN)** – user can define sampling rate in minute, at which rate the user requires information/data on the cloud.

**Destination** – user can define/add the list of applications on which the user wants data. For

example “Matricon\_OPC” here user need to select the destination from available drop down list (list available are already created in Integration).

## Port

**Port Select** – There are three options for port selection. Need to select the “TCP/IP” option as OPC DA simulator is used.

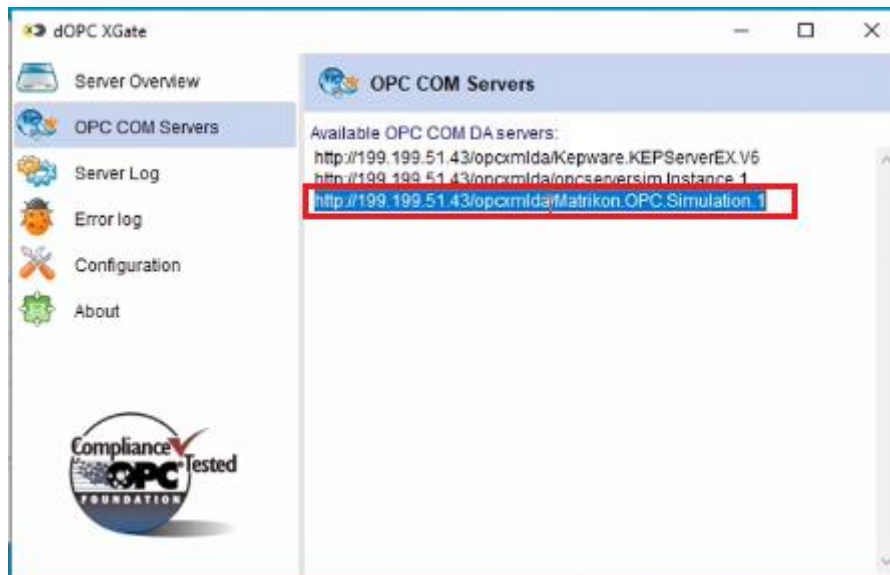
**IP** – Need to mention the IP address of OPC DA supported devices. Here in mention the IP address of OPC DA simulator is installed

**Port**- Port of PC/server

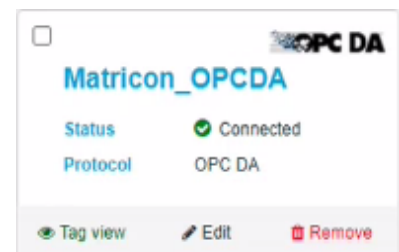
**Port timeout** - This is time at which Port of PC/server will automatically disconnected

## URL

- **URL:** Need to configure URL of the OPC DA Gateway as highlighted below.



- Now, device status has been changed to “Connected” as shown below image, which means OPC UA Server and Client connection set-up done successfully.





## ADD Tags

- User can add multiple tags for different parameters. Maximum 32 tags can be added.

The screenshot shows the 'OPC DA' configuration window. Under the 'Port' section, 'TCP/IP' is selected. The IP is '199.199.51.43', Port is '80', and Port timeout (Sec) is '5'. The URL is 'http://199.199.51.43/opcxmlda/Matrikon.OPC.Simulation.1'. In the 'Tags' section, there is a table with columns: Edge Name, OPC Name, Peripheral ID, and Remove. The first row contains 'Object1', 'Bucket Brigade Time', and '1234'. A red box highlights the '+ Add Tags' button with the text 'Click here to add tag'. A 'Save' button is at the bottom right.

Edge Name	OPC Name	Peripheral ID	Remove
Object1	Bucket Brigade Time	1234	

**Edge Name** : this is configurable option user can add Name as per there need

**OPC Name**: As defined in Simulators

**Peripheral ID** : as defined in CIMCON iCLOUD

## Tags defined at Simulation server

The first screenshot shows the 'Matrikon OPC Explorer' window. The 'Add Tags' button is highlighted with a red arrow. The second screenshot shows the 'Tag Entry' dialog box. The 'Item ID' is 'Bucket Brigade.LUnit2', 'Data Type' is 'Empty/Default', and 'Access Path' is 'Bucket Brigade.LUnit2'. The 'Available Items in Server' list includes 'Bucket Brigade', 'Random', 'Read Error', 'Sim-toothed Waves', and 'Square Waves'. The 'Available Tags' list includes 'ArrayOffset', 'Time', 'Unit1', 'Unit2', 'Unit3', and 'Unit4'. A red arrow points to the 'Save' button in the top right corner of the dialog box.

**Contents of Group0**

Item ID	Access Path	Value	Quality	Timestamp	Status
Bucket Brigade.ArrayOfReal8			Good, non-specific	12/20/2021...	Active
Bucket Brigade.ArrayOfString			Good, non-specific	12/20/2021...	Active
Bucket Brigade.Boolean	False		Good, non-specific	12/20/2021...	Active
Bucket Brigade.Int1	0		Good, non-specific	12/20/2021...	Active
Bucket Brigade.String			Good, non-specific	12/20/2021...	Active
Bucket Brigade.Time	12:00:00 AM		Good, non-specific	12/20/2021...	Active
Bucket Brigade.Int11	0		Good, non-specific	12/20/2021...	Active
Bucket Brigade.UInt2	0		Good, non-specific	12/20/2021...	Active

**OPC DA Configuration**

Port Select: ☐ Serial ☒ TCP/IP ☐ MAC

IP: 199.199.51.43 Port: 80 Port timeout (Sec): 5

URL: http://199.199.51.43/opcxmlda/Matrikon.OPC.Simulation.1

**Tags**

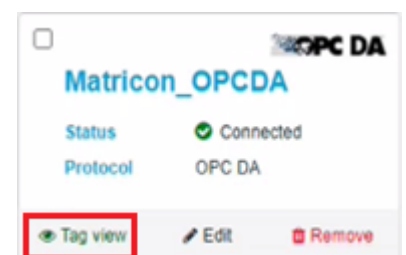
Edge Name	OPC Name	Peripheral ID	Remove
Object1	Bucket Brigade.Time	1234	

**Click here to add tag** [+ Add Tags](#)

**Save**

**After configuring a variable or object on the configuration page of OPC DA, click on “Save”.**

Tag View: To see the value of configured parameters on the configuration page, click on “Tag View” as shown below image and you’ll get the value of the configured parameters along with its Value, data type, timestamp, and status.



Tags - Matricon_OPCDA				
Matricon_OPCDA				
Name	Value	Type	Timestamp	Status
Object1	1899-12-30T00:00:00.000+05:30	dateTime	2021/12/23 06:14:53	✓
Random.ArrayOfRead8	14709	double	2021/12/23 06:14:53	✓
Random.ArrayOfString	control,software	ArrayOfString	2021/12/23 06:14:53	✓
Random.Boolean	false	boolean	2021/12/23 06:14:53	✓
Random.Int4	30351	int	2021/12/23 06:14:53	✓
Random.Money	21621	decimal	2021/12/23 06:14:53	✓
Random.Real8	450.80591085	double	2021/12/23 06:14:53	✓
Random.Time	2021-12-23T06:11:25.000+05:30	dateTime	2021/12/23 06:14:53	✓
Random.UInt1	59	int	2021/12/23 06:14:53	✓

Close

## CONTACT US

### U.S. OFFICE

319 Littleton Rd, Westford, MA 01886, USA

Email: [sales@cimcondigital.com](mailto:sales@cimcondigital.com)

Tel: (617) 741-9291

### ASIA PACIFIC OFFICE

**CIMCON DIGITAL (INDIA) Pvt. Ltd.**

802, SAKAR IV, EllisBridge, Ahmedabad-380 006, IndiaTel: +91-79-2657 8639

Fax: +91-79-2657-8659

**CIMCON**  
digital

solutions for the digital economy

SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE

MODBUS RTU,TCP IP, OPCUA and OPCDA

© CIMCON DIGITAL 2021