Introduction of CIM10

CIM10 serves as your gateway to seamless IoT edge connectivity and management. With its robust system monitoring, versatile network support, advanced security features, and seamless integration capabilities, CIM10 empowers you to unlock the full potential of your IoT ecosystem. From device integration to user application support, CIM10 offers a streamlined experience, ensuring scalability, flexibility, and security at every step.

Hardware Information

CIM10 is equipped with the following hardware specifications:

- The CIM10 has 1 Ethernet Port
- 1 The CIM10 has 1 Power Port (12v to 24v DC)
- The CIM10 has 1 Connector for the Modbus RS485, Digital Input 1 & 2, Analog Input 1 & 2, and Ground
- The CIM10 has 1 USB Mini Port for the power and backend tasks
- The CIM10 has 1 Micro HDMI Port
- The CIM10 has 1 Micro SD Card Slot
- The CIM10 has 1 SIM Card Slot
- The CIM10 has 1 GPS Antenna Slot
- The CIM10 has 1 GSM Antenna Slot
- The CIM10 has 4 LEDs (GPS LED, Bluetooth LED, LTE /GPRS LED, Power LED)

Application Interface

CIM10 comes with an integrated web UI that can be accessed through its default IP address.

- Default IP: 192.168.1.100
- Default Subnet: 255.255.255.0
- Default Gateway: 192.168.1.1

Steps to Open WebUI

- Open any web browser like Google Chrome, Firefox, Microsoft Edge, etc., on your PC or laptop.
- Open a new tab and enter the default IP address of CIM10 (e.g., 192.168.1.100) into the address bar and press Enter.
  Note: Ensure that your PC or laptop is in the same IP range.
- Once the webpage opens, it will prompt you to enter login credentials.
- Use the following credentials:
  Username: iEdgeAdmin
  Password: iEA@12345 or iEAcX#t6V)
  Note: Do not include the double quotes.
- After successful login, the system status page will appear.

WebUI Features

- System Status
    - In the System Status, users can view CPU consumption, memory consumption, network interface, volume backup, device uptime, network transport, active network interface, latitude, longitude, RTC, and time zone configuration.
    - Users can configure NTP servers manually or automatically from the RTC and Time Zone settings. By default, the Primary NTP Server is set to "time.google.com," and the Secondary NTP Server is set to "time1.google.com." To use these default NTP servers, simply check the "Use NTP Default Server" box.
    - Users can export CPU consumption, memory consumption, network interface, and volume backup data in various formats such as PNG, JPEG, PDF, SVG, ASV, and XLS.
- Network
    - In the "Network" section, there are four options available:

- Ethernet
    - In the Ethernet configuration, the following options are available:
      Enable/Disable Interface
      DHCP Server/Static IP Address
    - When DHCP Mode is selected, the CIM10 will automatically obtain an IP address from the DHCP server available on the LAN network. And when Static IP Address Mode is selected user needs to set a manual IP Address.
- GSM / LTE
    - In the GSM / LTE configuration, the following options are available:
      Enable/Disable Interface
      Serial Port (Should be the default)
      APN Name (Which will be different as per the used SIM)
      Modem Name
    - There are two options are there in the “Modem name”:
      2G and 4G
    - On the right-side GSM / LTE Status will be visible, there are below mentioned options are available:
      IP Address
      Gateway
      Signal Strength
      Registration
      Status
      Operator Name
      IMEI
- Port Forwarding
    - In this webUI section user will be able to forward Ethernet port and GSM SIM card-related settings

- Firewall Settings

- In this webUI section user will be able to configure and manage the Ethernet port and GSM SIM card-related settings

- Peripherals
  - In this webUI section user will be able to Digital Input, Analog Input, and GPS.
- Device
  - In this webUI section user will be able to create the devices that need to communicate with the CIM10.
  - Here Modbus RTU, Modbus TCP, OPC UA, BACnet IP, PROFINET, Ethernet/IP, and Vibit_BP Protocols are available for the communication.
- Tags
  - In this webUI section user will be able to access MQTT Topics of the individual tags for the specific devices.
- Integration
  - In this webUI section user will be able to assign and configure details for the MQTT Server / Broker. For the Transport, there are 4 options are there, CIM Cloud, MQTT, AWS, and Asure.
- User Application
  - CIM10 supports Python SDK as the user can write the Python script / Custom Application by using Python SDK.
- Package Manager
  - In this webUI section user will be able to see the current package versions and well user can update the packages.
- Service Manager
  - In this webUI section user will be able to see the status of the specific Services related to Network, Peripheral, Device, Integration, and User Application. Also, the user can Enable / Disable services, turn ON / OFF Debug logs, can Start / Restart / Stop services.
- Board Configuration
  - In this webUI section user will be able to Enable /Disable specific service and service configuration settings. Users can disable the specific Service/service configuration settings to decrease load.

CIM10 Analog Input Configuration steps

For Physical connection: CIM10 has onboard PERIPHERAL IO

There are Two Analog Inputs available in the CIM10 with 12-bit resolution and 0-10V or 4-20mA input.

IO expansion: Yes, Modbus IO (External on Modbus RS-485)

CIM10 has 8 Pin Connector and from that 8 Pins 3pins AI-1, AI-2, GND will be used,

Example: A Flow meter will have two wires for its 4-20mA analog output one wire will be for Signal and the other will be for Ground,

so, on CIM10, the signal wire from the flow meter will be connected to AI-1 or AI-2 Pin and the Ground wire will be connected to GND Pin

For Web page configuration

There will be below mentioned fields on the configuration page of CIM10

- Pin Number: to add Ai inputs for configuration
- Sampling rate (Sec): reads data on the configured frequency
- Destination: if the user wants data on the Cloud, then the user can select the desired cloud service (Configured in the integration section of CIM10)
- Name: The user can give the desired Analog input name
- Device ID: The user has to add the CIMCON Cloud Device ID
- Channel Type: The user will select Analog Input type Voltage or Current
- Engg. Scale Low: for current User will set 4, Voltage user will set 0
- Engg. Scale High: for current User will set 20, For Voltage user will set 10
- Scale Low: The user will set the desired low value for Scaling
- Scale High: The user will set the desired High value for Scaling

For example, the user has connected one Flow meter on an Analog input AI-1 pin and wants to configure it in CIM10, and 4-20mA will be scaled in 0-100 percent,

Then Below will be the Configuration settings in CIM10

- Pin Number: 1
- Sampling rate (Sec): 10
- Destination: CIMCON Cloud
- Name: Flow Meter
- Device ID: 1122334455667788
- Channel Type: Current
- Engg. Scale Low: 4
- Engg. Scale High: 20
- Scale Low: 0
- Scale High: 100

CIM10 Digital Input Configuration steps

For Physical connection: CIM10 has onboard PERIPHERAL IO

There are 2 Digital Inputs available in the CIM10 which are 12V DC or 24V DC operated.

CIM10 has 8 Pin Connector and from that 8 Pins 3 pins DI-1, DI-2, and GND will be used,

Example: A Pressure Sensor will have two wires for its Digital output one wire will be for Signal and the other will be for Ground,

so, on CIM10, the signal wire from the Pressure Sensor will be connected to DI-1 or DI-2 Pin and the Ground wire will be connected to GND Pin

At specific pressure thresholds, the pressure sensor detects and measures the pressure. Based on the predefined parameters, it generates a digital output signal, which is captured and interpreted by the CIM10.

For Web page configuration

There will be below mentioned fields on the configuration page of CIM10

- Pin Number: to add DI inputs for configuration
- Sampling rate (Sec): reads data on the configured frequency
- Destination: if the user wants data on the Cloud, then the user can select the desired cloud service (Configured in the integration section of CIM10)
- Pin Name: The user can give the desired Digital input name
- Device ID: The user has to add the CIMCON Cloud Device ID
- Status: The user can see the real-time status of the Digital Input from this field.

For example, the user has connected one Pressure Sensor to a Digital input AI-1 pin and wants to configure it in CIM10.

- Pin Number: 1
- Sampling rate (Sec): 10
- Destination: CIMCON Cloud
- Pin Name: Pressure Sensor
- Device ID: 1122334455667788

Modbus RS485 Protocol Overview

A Modbus RS485 Master gateway or any other Modbus Master is used to collect data from multiple Modbus RS485 Slave devices, such as sensors and controllers, within an industrial environment. The Modbus RS485 protocol enables reliable and long-distance communication between the master and slaves using differential signaling, ensuring robust performance even in noisy conditions. The master gateway reads the data from these slaves, leveraging the simple and efficient Modbus protocol. Once collected, the gateway converts this data into the MQTT protocol, a lightweight messaging protocol ideal for low-bandwidth, high-latency networks, and sends it to the cloud. This integration allows real-time monitoring and control of industrial processes through cloud-based applications, combining the reliability of Modbus RS485 with the flexibility and accessibility of MQTT for IoT solutions.

➢ Modbus RS485 Registers Overview

Modbus registers are memory locations used to store data within Modbus devices. These registers are essential for communication between Modbus Master and Slave devices, enabling the exchange of data for monitoring and control purposes.

Modbus registers are fundamental for data storage and communication in Modbus-based systems, with four main types: Coils Input, Input Status, Input Registers, and Holding Registers. These registers support various data types, including 16-bit integers, 32-bit integers, 32-bit floating points, Booleans, and many more enabling flexible data representation for industrial automation and control.

➢ Types of Modbus Registers

Modbus supports four primary types of registers:

1. Coils Input (Discrete Outputs)

The address range 00001-09999 is for 1-bit (Boolean) registers that can be read from the Slave and written by the master device. These registers are used to represent binary outputs.

2. Input Status (Discrete Inputs)

The address range 10001-19999 is for 1-bit (Boolean) registers that are read-only. These registers are used to represent binary inputs and can only be read by the master device.

3. Input Registers

The address range 30001-39999 is for 16-bit read-only registers. These registers store input data, such as measurements from sensors, and can only be read by the master device.

4. Holding Registers

The address range 40001-49999 consists of 16-bit read/write registers. These registers are used to store general data and parameters. They can be both read from the Slave and written to by the master device.

CIM10 CIM Modbus RS485 Configuration

CIM10 Supports Modbus RS485 and Modbus TCP Protocols, below are the steps given for the configuration:

➢ Steps for the Modbus RS485 or RTU Configuration with the CIM10:

1. Navigate to the "Device" section in the CIM10 WebUI.
2. Click the "+" button, select the protocol "Modbus RTU," and Give name to your Modbus RS485 slave.
3. Click on the "Edit" option on the created device.
4. Note that "Protocol" and "Name" are already assigned and cannot be edited.
5. **Destination:** Select the appropriate CIM10 "Integration" (e.g., CIM Cloud, Azure, MQTT, AWS)
6. **Port Select:** "Serial" is automatically selected as the protocol is "Modbus RTU."
7. **Port:** Select "RS485".
8. **Parity:** Choose one of the following options based on your Modbus RTU devices: None, Even, and Odd.
9. **Stop Bits:** Select either 1 or 2, as required by your Modbus RTU devices.
10. **Baud Rate:** Select the appropriate baud rate for connected Modbus RTU devices. The available baud rates in the CIM10 are 4800, 9600, 19200, 38400, 57600, and 115200.
11. **Data Bits:** Choose either 7 or 8, as specified by your Modbus RTU device.
12. **Port Timeout:** Enter the port timeout in milliseconds. The default is 3000ms.
13. **Device Template:** Select a pre-configured device template from the CIM10 database or save a custom template.
14. **Slave ID:** Enter the Slave ID of the connected Modbus RS485 device.
15. **Add Query:** Click on "Add Query" to define the Modbus registers to be fetched. Fill in the following details:

- **Input:** Select the appropriate Modbus register type. Available Register types are Coil Input, Input Status, Holding Register, and Input Register.
- **Interval (ms):** Set the interval for fetching data. The default is 60000ms.
- **Address:** The Modbus register number should be specified in this field, indicating which register number the CIM10 should use to fetch data from the Modbus slave.
- **Device ID:** Device ID is a specific ID obtained from the created device on the CIM Cloud.
- **No of Registers:** Add the numbers of Modbus Registers that will be used for this particular Query.
- **Peripheral Name:** Give the specific name to this Query.
- **Query Interval (ms):** It is a delay between one Modbus Poll, which can be modified per the requirement, and Modbus Slave response capacity. The Default Query interval is 100ms.

16. **Parameters:** In this section Data type, Factor, Converter, and Parameter's name need to be mentioned.

- **Datatype:** There are several Data types available like Character 1-Byte, Integer, Float, Double, Long, Unsigned Long, Unsigned Integer, Long Long, Unsigned Long Long, Character 2-Byte, and Decimal. The User can select the datatype that the connected Modbus RTU Slave supports.
- **Factor:** In this field that number can be inserted from which the original data need to be multiplied. The default Factor is 1.
- **Converter:** This field is used when the original data needs to be converted into another Datatype. Below are the available converters for the Specific Data Types:

  - **Integer:** BE Integer, LE Integer, Float, and Hex
  - **Float:** BE Float, LE Float, BE Swapped Float, and LE Swapped Float
  - **Double:** BE Double, LE Double, BE Swapped Double, and LE Swapped Double
  - **Long:** BE Long, LE Long, BE Swapped Long, LE Swapped Long, Float, and Hex
  - **Unsigned Long:** BE Long, LE Long, BE Swapped Long, LE Swapped Long, Float, and Hex
  - **Unsigned Integer:** BE Integer, LE Integer, Float, and Hex
  - **Long Long:** BE Long Long, LE Long Long, BE Swapped Long Long, LE Swapped Long Long
  - **Unsigned Long Long:** BE Long Long, LE Long Long, BE Swapped Long Long, LE Swapped Long Long
- **Name:** In the Name, Section enter the Name of that specific Modbus Register and which data it indicates.

Modbus TCP Protocol Overview

A Modbus TCP Master gateway or any other Modbus TCP Master device is used to collect data from multiple Modbus TCP Slave devices over an Ethernet network. Modbus TCP/IP extends the traditional Modbus protocol to operate over TCP/IP, allowing for high-speed, reliable communication in industrial settings. The gateway acts as a master, polling the Modbus TCP slaves for data, which might include various sensor readings, actuator states, or other operational parameters. Once the data is gathered, the gateway converts it into the MQTT protocol, a lightweight messaging protocol optimized for low-bandwidth, high-latency networks, and publishes it to the cloud. This integration allows real-time data monitoring and control from remote locations, leveraging the robustness of Modbus TCP for local data collection and the efficiency of MQTT for cloud communication, facilitating advanced IoT applications and seamless industrial automation.

Modbus TCP offers several advantages over Modbus RS485, making it a preferred choice for modern industrial automation. Utilizing Ethernet networks, Modbus TCP allows for higher data transfer rates, greater scalability, and easier installation and maintenance compared to the traditional RS485. It supports simultaneous connections between multiple masters and slaves, extends communication distances without degradation, and integrates seamlessly with existing IP-based systems, enhancing flexibility and interoperability. Additionally, Modbus TCP benefits from advanced security features and network management tools, providing robust, reliable, and secure communication ideal for today's interconnected industrial environments.

Using standard IPv4 addressing, a subnet can typically handle up to 254 devices, with larger networks possible through subnetting or IPv6. Practical limitations such as network bandwidth, latency, and the master device's processing capabilities also influence the effective number of connected slaves, but well-designed networks can comfortably support a large number of devices for industrial applications.

In the Modbus TCP protocol, port 502 is the default port used for communication between Modbus TCP master (client) and slave (server) devices over an Ethernet network.

CIM10 CIM Modbus TCP Configuration

Below are the steps given for the configuration:

➢ Steps for the Modbus TCP Configuration with the CIM10:

1. Navigate to the "Device" section in the CIM10 WebUI.
2. Click the "+" button, select the protocol "Modbus TCP," and give a name to your Modbus TCP slave.
3. Click on the "Edit" option on the created device.
4. Note that "Protocol" and "Name" are already assigned and cannot be edited.
5. **Destination:** Select the appropriate CIM10 "Integration" (e.g., CIM Cloud, Azure, MQTT, AWS)
6. **Port Select:** "TCP / IP" is automatically selected as the protocol is "Modbus TCP".
7. **IP:** Enter the IP address of the Modbus TCP device.
8. **Port:** Enter the port number for the Modbus TCP device (default is 502).
9. **Port Timeout:** Enter the port timeout in milliseconds. The default is 3000ms.
10. **Device Template:** Select a pre-configured device template from the CIM10 database or save a custom template.
11. **Slave ID:** Enter the Slave ID of the connected Modbus RS485 device.
12. **Add Query:** Click on "Add Query" to define the Modbus registers to be fetched. Fill in the following details:

- **Input:** Select the appropriate Modbus register type. Available Register types are Coil Input, Input Status, Holding Register, and Input Register.
- **Interval (ms):** Set the interval for fetching data. The default is 60000ms.
- **Address:** The Modbus register number should be specified in this field, indicating which register number the CIM10 should use to fetch data from the Modbus slave.
- **Device ID:** Device ID is a specific ID obtained from the created device on the CIM Cloud.
- **No of Registers:** Add the numbers of Modbus Registers that will be used for this particular Query.
- **Peripheral Name:** Give the specific name to this Query.
- **Query Interval (ms):** It is a delay between one Modbus Poll, which can be modified per the requirement, and Modbus Slave response capacity. The Default Query interval is 100ms.

13. **Parameters:** In this section Data type, Factor, Converter, and Parameter's name need to be mentioned.

- **Datatype:** There are several Data types available like Character 1-Byte, Integer, Float, Double, Long, Unsigned Long, Unsigned Integer, Long Long, Unsigned Long Long, Character 2-Byte, and Decimal. The User can select the datatype that the connected Modbus RTU Slave supports.
- **Factor:** In this field that number can be inserted from which the original data need to be multiplied. The default Factor is 1.
- **Converter:** This field is used when the original data needs to be converted into another Datatype. Below are the available converters for the Specific Data Types:
  - **Integer:** BE Integer, LE Integer, Float, and Hex
  - **Float:** BE Float, LE Float, BE Swapped Float, and LE Swapped Float
  - **Double:** BE Double, LE Double, BE Swapped Double, and LE Swapped Double
  - **Long:** BE Long, LE Long, BE Swapped Long, LE Swapped Long, Float, and Hex
  - **Unsigned Long:** BE Long, LE Long, BE Swapped Long, LE Swapped Long, Float, and Hex
  - **Unsigned Integer:** BE Integer, LE Integer, Float, and Hex

- **Long Long:** BE Long Long, LE Long Long, BE Swapped Long Long, LE Swapped Long Long
- **Unsigned Long Long:** BE Long Long, LE Long Long, BE Swapped Long Long, LE Swapped Long Long

- **Name:** In the Name, Section enter the Name of that specific Modbus Register and which data it indicates.

➢ Data Types Supported by Modbus Registers

While Modbus registers natively support 16-bit data, various data types can be represented using combinations of these registers:

1. 16-bit Integer:

Directly stored in a single register.

Examples: 30001, 40001

2. 32-bit Integer:

Stored across two consecutive registers.

Example: Registers 40001-40002, 30001-30002

3. 32-bit Floating Point:

Stored across two consecutive registers using IEEE 754 format.

Example: Registers 40001-40002, 30001-30002

4. Boolean:

Single-bit data is stored in Coils or Discrete Inputs.

Example: Coil 00001, Discrete Input 10001

Complex data structures can be mapped to multiple registers, depending on the application and implementation requirements.

Note: Datatypes will remain same for the both Modbus RTU/RS485 and Modbus TCP Protocols.

Data Types Description:

- **Boolean:** Represents binary data (true/false, 1/0).
- **Unsigned Integer:** Ranges from 0 to 65,535.
- **Signed Integer:** Ranges from -32,768 to 32,767.
- **Floating Point:** IEEE 754 format, typically using two consecutive registers (32-bit).
- **Double Precision Float:** IEEE 754 format, using four consecutive registers (64-bit).
- **Signed Long (INT32):** Combining two 16-bit registers to form a 32-bit integer. Ranges from -2,147,483,648 to 2,147,483,647.
- **Unsigned Long (UINT32):** Combining two 16-bit registers to form a 32-bit unsigned integer. Ranges from 0 to 4,294,967,295.

Some are the custom conversations are also available in the Modbus that are as below:

- **Big Endian:**

Definition: In Big Endian format, the most significant byte (MSB) is stored at the first register address, and the least significant byte (LSB) is stored at the subsequent register address.

Example: Consider a 16-bit register value 0x1234. In Big Endian format, it would be stored as follows:

Register Address 0: 0x12 (MSB)

Register Address 1: 0x34 (LSB)

Usage: Big Endian format is commonly used in network communication (network byte order). When dealing with Modbus communication, some devices might default to this format.

- **Little Endian:**

Definition: In Little Endian format, the least significant byte (LSB) is stored at the first register address, and the most significant byte (MSB) is stored at the subsequent register address.

Example: Using the same 16-bit register value 0x1234, in Little Endian format, it would be stored as follows:

Register Address 0: 0x34 (LSB)

Register Address 1: 0x12 (MSB)

Usage: Little Endian format is often used by x86 and x86-64 architectures. Some Modbus devices may use this format to store multi-byte values.

CIM10 CIM Cloud Integration

➢ Integration Configuration Steps:

1. Navigate to the "Integration" section located on the left side of the CIM10 Web UI.
2. Click on the "+" icon within the Integration section.
3. Choose "CIM Cloud" from the list of options and provide an appropriate name for this Integration.
4. Click "Save" to create the CIM Cloud Integration.
5. Refresh the Integration by clicking the Refresh button located at the top right corner.
6. Click on "Edit" for the created Integration.
7. Note that the "Transport Name" and "Protocol" fields are not editable as they are determined by previous selections.
8. For any other options, please contact the CIMCON Support Team.

➢ Integration Options Definitions:

- **End Point:** Enter the CIM Cloud IP Address in this section.
- **HTTP Port:** Add HTTP Port **8080** in this section.
- **MQTT Port:** Add MQTT Port **1883** in this section.
- **MQTTs Port:** Add MQTTs Port **8883** in this section.
- **Username (Email):** Enter the Username of the CIM Cloud Account in this section.
- **Password:** Enter the appropriate Password of the Username (Email) in this section.
- **Authentication:** Define MQTT Authentications in this field by clicking on “Add Authentication” from the top right corner. For each Authentication, fill in the following details:
    1. **Device ID:** Enter the unique Device ID automatically defined for the device on the CIM Cloud.
    2. **Auth Type:** Choose between “MQTT X.509” or “MQTT Basic”.
    3. **Client ID:** Automatically generated after saving.
    4. **User:** Automatically generated after saving.
    5. **Password:** Automatically generated after saving.
    6. **Remove:** Delete specific Authentication if needed.
- **Incoming Message:** Subscribe to specific topics from the CIM Cloud using this option.
- **Outgoing Message:** Push device data or service data to the CIM Cloud using this option. Fill in the following details:
    1. **App Name:** Select the specific field containing device data or service data.
    2. **Type:** Choose between Data, Event, Response, and Attribute. Data is commonly selected.
    3. **Client ID:** Select the specific Device ID assigned for the device.
    4. **Topic:** Write the default MQTT Topic **“v1/devices/me/telemetry”** for sending data to the CIM Cloud.
- Click the “Save” button to save the Integration Configuration.