

MQTT SparkplugB Driver

Summary Information

Communication Driver Name: MQTTspB

Implementation DLL: T.ProtocolDriver.MQTTspB.dll

Interface: TCPIP

Protocol: MQTT (Message Queuing Telemetry Transport) is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. "Sparkplug provides an open and freely available specification for how Edge of Network (EoN) gateways or native MQTT enabled end devices and MQTT Applications communicate bi-directionally within an MQTT Infrastructure." ([Sparkplug Specification](#))

PC Hardware requirements: Ethernet board

Implemented Methods: Connect, Disconnect, Subscribe, and Publish

Channel Configuration

Protocol Options

Type: It defines how a channel works when it connects to the MQTT Broker

- **Application Node:** A MQTT client that consumes real-time messages, performs analytical calculations, saves history data, etc. Subscription is allowed. Publishing is not allowed
- **Scada IIoT Host:** A primary MQTT client that consumes real-time messages and publishes commands (messages) to **EoN Node** nodes and devices
- **EoN Node:** A MQTT client that mainly functions to publish messages read from field equipment. Also, it can receive commands from a **Scada IIoT Host**

ClientID: Indicates the client identification used to connect to the MQTT Broker

TagProperties: Defines the tag properties besides the tag Value. The names of the properties should be separated by a ',' (comma)

TimePublishRate: Indicates the time to publish in ms. The messages that are published to the MQTT Broker are grouped and sent using this period

Node Configuration

Station Configuration

Station syntax: <BrokerURL>;<Port>;[Username];[Password];[X509Certificate];[SslProtocol];[WebSocket];[ScadaIIoTHostID]

Where:

- **<BrokerURL>** = A MQTT Broker (Server) URL
- **<Port>** = A MQTT Broker port. It must be the same port that is configured in the Broker as the listening port. Default value is 1883
- **[Username]**, optional = A Username defined in the MQTT Broker. This will be requested if the Broker needs this configuration
- **[Password]**, optional = A Password defined in the MQTT Broker. This will be requested if the Broker needs this configuration
- **[X509Certificate]**, optional = The Path for the X509 Client Certificate. If using a server-side only certificate, this field does not need to be configured. This must be the X509 certificate's whole in the client computer. The certificate must be installed on the computer



Note

One of the easiest ways to install the certificate on the client computer is to use the wizard to import the certificates through the "Internet Options" and to import the certificate in the "Trusted Root Certification Authorities". You need to import the certificate in .DER format.

- **[SslProtocol]**, optional = If the MQTT Broker works by using a certificate, this field is mandatory and must be configured the same as it was configured in the MQTT Broker. Options: None, Ssl2, Ssl3, Tls, Tls11, and Tls12
- **[WebSocket]**, optional = A Flag indicating if the connection to the MQTT Broker should be via WebSocket. Check how the MQTT Broker works before checking it

- **[ScadalIoTHostID]**, optional = This field identifies what MQTT Broker must be used if the channel works as a EoN Node, the Scada IIoT Host is active, and other MQTT Brokers are present in the network (redundancy, scalability, etc). Scada IIoT Host publishes your ID when you are connected to the MQTT Broker

**Note**

All nodes of the same channel share all device points regardless of which node they have been configured to. For MQTT Brokers redundancy, more than one node should be used for the same channel.

Point Configuration

Address

The syntax for the MQTT communication points is: <GroupId>;<EdgeNodeId>;<DeviceId>;<QoS>

Where:

- **<GroupId>** provides a logical grouping of Edge Node's. Wildcards are not supported
- **<EdgeNodeId>** identifies the Edge Node's ID. Wildcards are not supported.
- **<DeviceId>** identifies the Device's ID from the Edge Node. This field can be empty while accessing the main Edge Node. Wildcards are not supported
- **<QoS>** Quality of Service (QoS) is an agreement between a message's sender and receiver regarding the message's guarantees of delivery. There are 3 QoS levels: AtMostOnce, AtLeastOnce, and ExactlyOnce

Troubleshoot

The status of the driver's execution can be observed through the diagnostic tools, which are:

- Trace window
- Property Watch
- Module Information