MQTT SparkplugB Client Communication Driver

This document has the specific information related to this driver configuration. For a generic explanation on Device Module, Channels, Nodes and Points configuration, please refer to reference guide.

Contents

Section 1 – 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.*" (https://s3.amazonaws.com/cirrus-link-com/Sparkplug+Topic+Namespace+and+State+ManagementV2.1+Apendix++Payload+B+format.pdf)

PC Hardware requirements: Ethernet board

Implemented Methods: Connect, Disconnect, Subscribe and Publish.

Section 2 – Channel Configuration

Protocol Options

Type: Defines how the channel works when connecting to MQTT Broker.
- **Application Node**: It is a MQTT client that consumes the real-time messages and performs some analytical calculations, save history data, etc. Subscription is allowed but publishing is not allowed.
- **Scada IIoT Host**: It a primary MQTT client that consumes the real-time messages but also publishes commands (messages) to **EoN Node** nodes and devices.
- **EoN Node**: It is a MQTT client that has as its main function to publish messages read from field equipment. It also can receive commands from **Scada IIoT Host**.

ClientID: Indicates the client identification used for connecting to the MQTT Broker.

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

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

Section 3 – Node Configuration

Station Configuration

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

Where:

 *< **BrokerURL** > =* MQTT Broker (Server) URL.

*< **Port** > =* MQTT Broker port. It must be the same *port* is configured in the Broker to be listening. Default value is 1883.

*[**Username**], optional =* Username defined on MQTT Broker. This will be requested if the Broker needs this configuration.

*[**Password**], optional =* Password defined on MQTT Broker. This will be requested if the Broker needs this configuration.

*[**X509Certificate**], optional =* Path of X509 Client Certificate. If using server-side only certificate this field does not need to be configured. This must be the complete path of X509 certificate in the client computer. The certificate must be installed in the computer.
Note: One of the easiest ways to install the certificate on client computer is to use the wizard to import the certificates is through the "Internet Options" to import the certificate in the "Trusted Root Certification Authorities". You need to import the certificate in DER format.

*[**SslProtocol**], optional =* If MQTT Broker works using certificate this field is mandatory and must be the same that was configured in MQTT Broker. Options: None, Ssl2, Ssl3, Tls, Tls11 and Tls12.

*[**WebSocket**], optional =* Flag indicating whether connection to MQTT Broker should be via WebSocket. Check how MQTT Broker works before checking it.

*[**ScadaIIoTHostID**], optional =* If channel works as **EoN Node**, some **Scada IIoT Host** is active and other MQTT Brokers are also present in the network (redundancy, scalability, etc) then this field is used to know what is MQTT Broker must be used. **Scada IIoT Host** publishes your ID when connecting to MQTT Broker.

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

Section 4 – Point Configuration

Address
The syntax for the MQTT communication points are:

- **<GroupId>;<EdgeNodeId>;<DeviceId>;<QoS>**

< GroupId > provides a logical grouping of Edge Node's. Wildcards are not supported.

< EdgeNodeId > identifies ID of Edge Node. Wildcards are not supported.

< DeviceId > identifies ID of Device from Edge Node. This field can be empty while accessing main Egde Node. Wildcards are not supported.

<QoS> Quality of Service (*QoS*) is an agreement between sender and receiver of a message regarding the guarantees of delivering a message. There are 3 QoS levels:

- **AtMostOnce**
- **AtLeastOnce**
- **ExactlyOnce**

Section 5 – Troubleshoot

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

- Trace window

- Property Watch

- Module Information

Revision History

Revision	Version	Description	Date
A	1.0.0.1	Initial Revision	*October* 2019