Irisys MQTT Specification

Introduction

Irisys Vector devices can be configured to deliver event-based data to a 'broker' service using the MQTT protocol.

This document describes the implementation of MQTT on the Vector device, and the format Irisys use to contain the data.

NOTE: this document assume you have a working knowledge of MQTT, the publish/subscribe model, topics, and the configuration of your own choice of broker.

Configuration Options

The following configuration options can be set in the Vector device's configuration tool "RIFT" (accessible by browsing to the device IP address):

Address

This is the IP address or hostname of the MQTT broker. If using a hostname, please ensure that the device's IP settings are configured with a valid and accessible DNS server.

Port

MQTT is on port 1883 by default, but this can be configured to a different port, should this be required.

Keep Alive (seconds)

This configuration option dictates how regularly the device sends a keep alive message to the broker. This is part of the MQTT protocol.

Max History (hours)

In the event that the device cannot communicate with the broker, it will retain any messages not yet sent. To limit storage requirements, any messages exceeding the max history shall be discarded.

Max Size (bytes)

This configuration option limits the amount of the data that the device can send to the MQTT broker in a single transmission. Any data not sent in a transmission due to this limit will be retained in the history and will be the priority for sending in the next transmission. Irisys recommend that the limit is set to no more than 256Kb. The user may wish to adjust this lower to manage message size limits.

Count Topic Address

MQTT messages are sent to the broker by 'topic'. This configuration option allows the user to set the topic string that count register changes are published to.

Count Topic Period (seconds)

The device will send count register changes on an interval controlled by the count topic period.

Status Topic Address

Connection and disconnection events (regarding the device's connection to the broker) are sent over this topic.

Targets Topic Address

Live targets are streamed over this topic at 15 frames per second.

MQTT Message Format

There are three different message types: "Counts", "Status" and "Targets", all published on separate topics.

Counts

Whenever a count register value changes, an event message is queued containing the register name, the change in value and the timestamp. This will be sent out along with any other pending messages on the next publication interval (configured in Rift as "period").

When the publication interval comes around, any pending messages are sent out in a JSON array as exemplified below:

```
[{
        "register": "MainEntranceIn",
        "value": 1,
        "t": "2019-02-91T12:30:00+0000"
}, {
        "register": "MainEntranceOut",
        "value": 1,
        "t": "2019-02-91T12:30:12+0000"
}, {
        "register": "MainEntranceIn",
        "value": 1,
        "t": "2019-02-91T12:30:14+0000"
}]
```

In the example above there have been count increments on two registers- two on the one named as "MainEntranceIn" and one on the register named as "MainEntranceOut". The timestamps indicate the time of each event occurrence.

If no count events occur in the interval, an empty array will be sent out as below:

```
[]
```

Status

Status event messages are published on connection and (ungraceful) disconnection from the broker.

Connection Message

This is of the following format:

{"status": "connected"}

Disconnection Message

This is sent up to the broker on connection as its "Last Will and Testament". The broker will then push this out to subscribers in the event of the device's (ungraceful) disconnection. This is of the following format:

{"status": "disconnected"}

Targets

The Vector4D device supports sending the co-ordinate data for each tracked target in real-time over MQTT. This allows the end-user to build rich applications beyond people counting, and includes Vector 4D features such as multi-unit target tracking and staff identification.

Data Format

The target data is delivered on a per-frame basis, regardless of whether any targets are currently being tracked. Data is delivered from the device to the broker at 15 frames per second.

The target data is delivered as little-endian binary data with no packing. The first byte is the target data format byte. This byte will always be present. If packet type is >=1, the next 8 bytes are timestamp (type 0 contained no timestamp). This is the number of milliseconds that have elapsed since 1 Jan 1970 (Unix Epoch time).

DataType	**Description**
uint8	Message type
uint64	timestamp (milliseconds), present since message type 1

The packet ends here if no targets are present for this frame. If targets are present, the data continues with 15 bytes per target, packed as follows:

DataType	Description
uint16	target ID
uint8	status bits (0x20: tagged)
float32	x position (m)
float32	y position (m)
float32	target height (m)