

## Application Note: Telematic Data Collector (TDC) supported Protocols

### Introduction

SICK Telematic Data Collector is essential component in modern IoT ecosystems, enabling seamless communication between various devices and systems. TDC-E support multiple communication protocols, each suited for specific applications and requirements. This application note provides an overview of four commonly used out-of-the-box protocols: MQTT, REST API, Modbus TCP and WebSocket. Understanding these protocols will help in selecting the appropriate one for your application, ensuring optimal performance and compatibility. Please note, TDC-E is a programmable device which means that it can support other protocols if implemented.

Please, follow our TDC\_E developers' documentation via here; [GitHub - SICKAG/sick\\_tdc-e-developers-documentation](#)

Our application engineering team can support you in specific use case.

### Protocols Overview

## 1. MQTT (Message Queuing Telemetry Transport)

### *Description*

MQTT is a lightweight, publish-subscribe network protocol that transports messages between devices. It is designed for connections with remote locations where a small code footprint is required and network bandwidth is at a premium.

### *When to use it with TDC-E*

- **IoT Applications:** Ideal for low-bandwidth, high-latency, or unreliable networks.
- **Home Automation:** Enables efficient communication between sensors and controllers.
- **Real-time Data Monitoring:** Facilitates real-time data collection and monitoring in industrial applications.

### *Key Features*

- **Lightweight:** Minimal overhead makes it suitable for constrained environments.
- **Scalable:** Can handle a large number of devices with minimal resource consumption.
- **QoS Levels:** Supports three Quality of Service levels for message delivery:

- QoS 0: At most once (no confirmation).
- QoS 1: At least once (confirmation required).
- QoS 2: Exactly once (two-phase confirmation).

### *Implementation*

MQTT requires an MQTT broker (e.g., Mosquitto, HiveMQ) to manage message distribution. Devices publish messages to topics, and subscribers receive messages from these topics as per their subscriptions.

## **2. REST API (Representational State Transfer Application Programming Interface)**

### *Description*

REST API is an architectural style for designing networked applications. It relies on a stateless, client-server communication model and uses standard HTTP methods (GET, POST, PUT, DELETE) to perform operations on resources represented by URIs (Uniform Resource Identifiers).

### *When to use it with TDC-E*

- **Web Services:** Widely used for web services and web applications.
- **Resource Management:** Suitable for CRUD (Create, Read, Update, Delete) operations on resources.
- **Interoperability:** Facilitates communication between different systems and applications.

### *Key Features*

- **Stateless:** Each request from a client contains all the information needed to process the request.
- **Scalable:** Handles large numbers of requests efficiently.
- **Interoperable:** Uses standard HTTP methods, making it easy to integrate with various clients and servers.

### *Implementation*

RESTful services are typically implemented using web servers (e.g., Apache, Nginx) and frameworks (e.g., Flask, Spring). Clients communicate with the server using HTTP requests, and the server responds with HTTP status codes and data, usually in JSON or XML format.

### 3. Modbus TCP

#### *Description*

Modbus TCP is an open, standardized protocol widely used in industrial automation systems for communication over TCP/IP networks. It allows the integration of Modbus devices with Ethernet-based networks.

#### *When to use it with TDC-E*

- **Industrial Automation:** Commonly used in SCADA (Supervisory Control and Data Acquisition) systems and PLCs (Programmable Logic Controllers).
- **Process Control:** Facilitates real-time control and monitoring of industrial processes.
- **Device Integration:** Enables integration of various Modbus devices in a networked environment.

#### *Key Features*

- **Simplicity:** Easy to implement and understand.
- **Compatibility:** Widely supported by industrial devices and systems.
- **Real-time Communication:** Suitable for real-time data exchange and control.

#### *Implementation*

Modbus TCP communication involves a client-server architecture where Modbus masters (clients) send requests to Modbus slaves (servers) over TCP/IP. The servers respond with the requested data or status information.

### 4. WebSocket

#### *Description*

WebSocket is a communication protocol that provides full-duplex communication channels over a single TCP connection. It is designed for real-time, event-driven applications.

#### *When to use it with TDC-E*

- **Real-time Applications:** Ideal for applications requiring real-time updates
- **Interactive Web Applications:** Enhances user experience by enabling real-time interactions.
- **IoT Applications:** Supports real-time data streaming from IoT devices to the cloud.

### *Key Features*

- **Full-Duplex Communication:** Allows simultaneous bidirectional data exchange between client and server.
- **Low Latency:** Reduces latency compared to traditional HTTP requests.
- **Persistent Connection:** Maintains a persistent connection, eliminating the need for repeated handshakes.

### *Implementation*

WebSocket communication starts with an HTTP handshake, after which the connection is upgraded to a WebSocket connection. WebSocket servers (e.g., Node.js with ws library) handle connections and data exchange with clients, which can be web browsers or other WebSocket-enabled applications.

### *Conclusion*

Selecting the appropriate communication protocol for your TDC-E device depends on the specific requirements of your application. MQTT is ideal for low-bandwidth IoT applications, REST API is suitable for web services and resource management, Modbus TCP is widely used in industrial automation, and WebSocket is perfect for real-time, interactive applications. Understanding the strengths and limitations of each protocol will help you design efficient and reliable communication systems for your wireless gateway.

For further assistance or specific implementation details, please contact our technical support team.

---

### **Contact Information:**

[SICK Mobilisis]  
Technical Support Team  
Email: [zzm-mo-ae-tdc@sick.com](mailto:zzm-mo-ae-tdc@sick.com)