



Azure IoT Hub Client

With the “Azure IoT Hub Client” library, CODESYS controllers can exchange messages with the “Azure IoT Hub” cloud service from Microsoft.

The library “Azure IoT Hub Client” is now part of the product [IIoT Libraries SL](#) and is no longer available as single product.

Product description

The “Azure IoT Hub” cloud service from Microsoft directly links IoT devices. (For more detailed information, refer to <https://azure.microsoft.com/en-us/services/iot-hub/>). The “Azure IoT Hub Client” library provides function blocks for sending and receiving messages. A sample project demonstrates how to use the library.

The library contains separate function blocks for communication via HTTPS and MQTT. The library supports the following functions:

- Send “Device to Cloud (D2C)” messages (telemetry data)
- Receive “Cloud to Device (C2D)” messages
- Read the device twin
- Update the device twin (desired properties only)
- Subscribe Device Twin (desired properties, MQTT only)
- Direct method call (Cloud -> Device, MQTT only)

The sample project “AzureExample.project” demonstrates how to use the corresponding function blocks.

Getting started

1. Setting up the Azure IoT hub

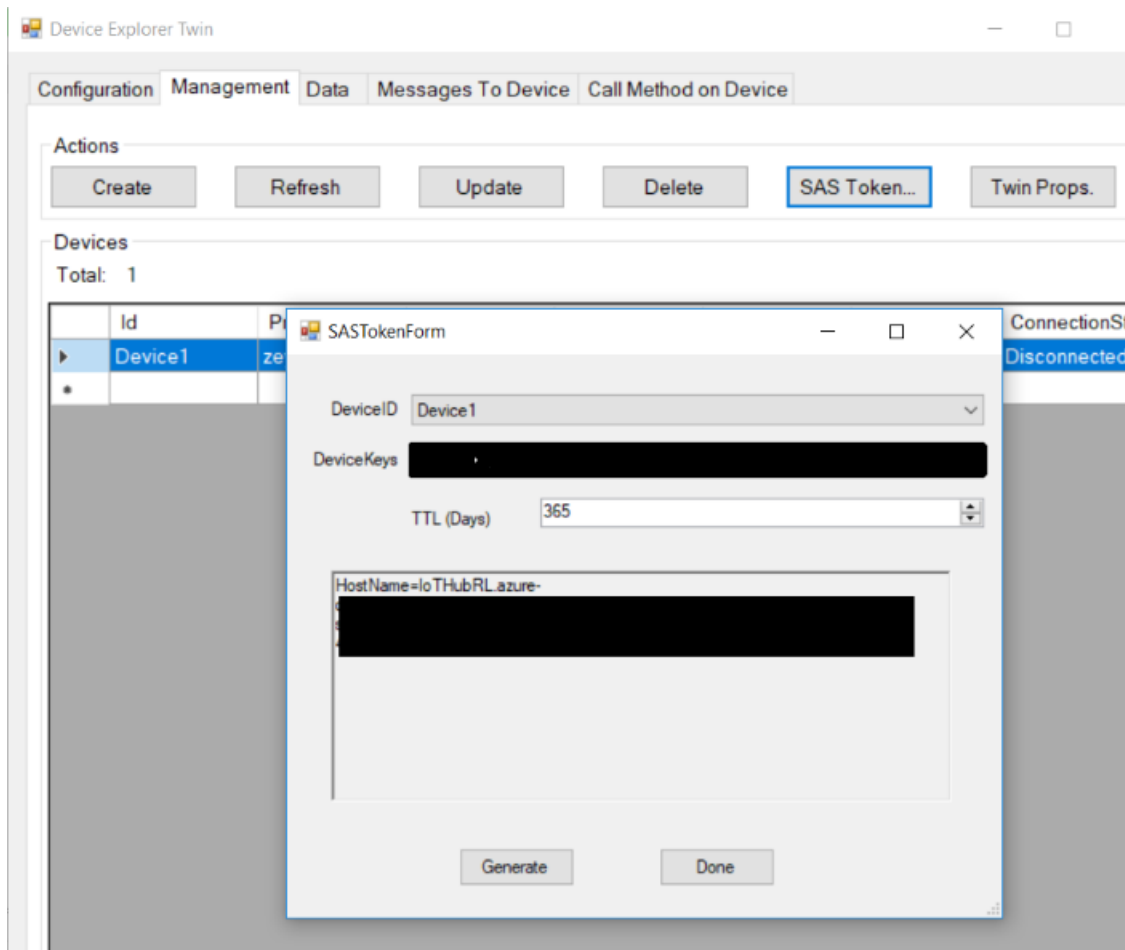
See <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-create-through-portal>

2. Creating devices in the IoT hub

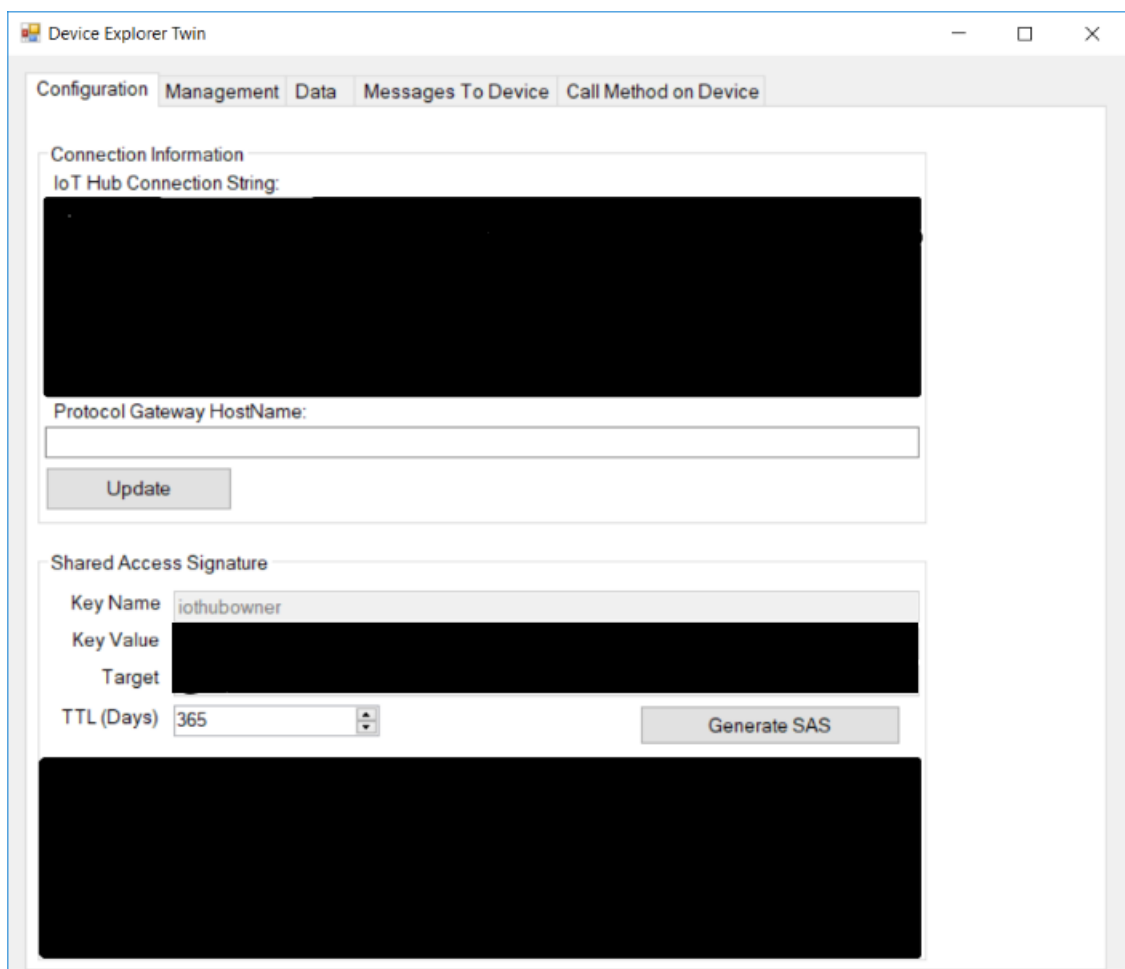
See <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-get-started-physical>

3. Generating shared access signatures (SaS token) with the “Device Explorer” tool

See <https://github.com/Azure/azure-iot-sdk-csharp/tree/master/tools/DeviceExplorer>



Device Explorer: Generate an SaS token for the device



Device Explorer: Generate an SaS token for the IoT hub owner (for read/write of the device twin)

4. Setting the name of the IoT hub, device ID, and SaS token in the sample project

sSubDomainName: Name of the Azure IoT Hubs (see 1.)

sDeviceId: Device ID (see 2.)

wsDeviceSaS: SaS token of the device (D2C, C2D messages) (see 3.)

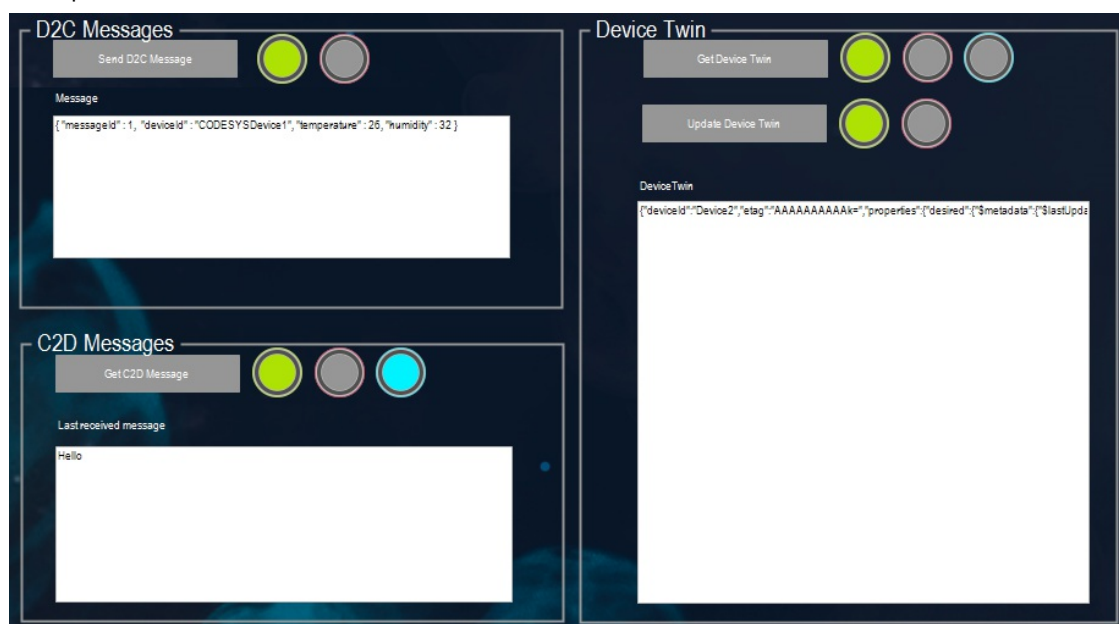
wsIoTHubOwnerSaS: SaS token of the IoT Hub owner (read/write of device twin) (see 3.)

5. Downloading the project to the controller and starting it

Application AzureHTTPEdemo

The following functions can be executed from the visualization:

- Send D2C message
- Get C2D message
- Get device twin
- Update device twin



Visualization of the HTTPS sample project

Applikation AzureMQTTDemo

The following functions can be executed from the visualization:

- Connect to IoT Hub via MQTT
- Send D2C message
- Get C2D message
- Get device twin
- Update device twin
- Direct method call
- Subscribe device twin (desired properties)

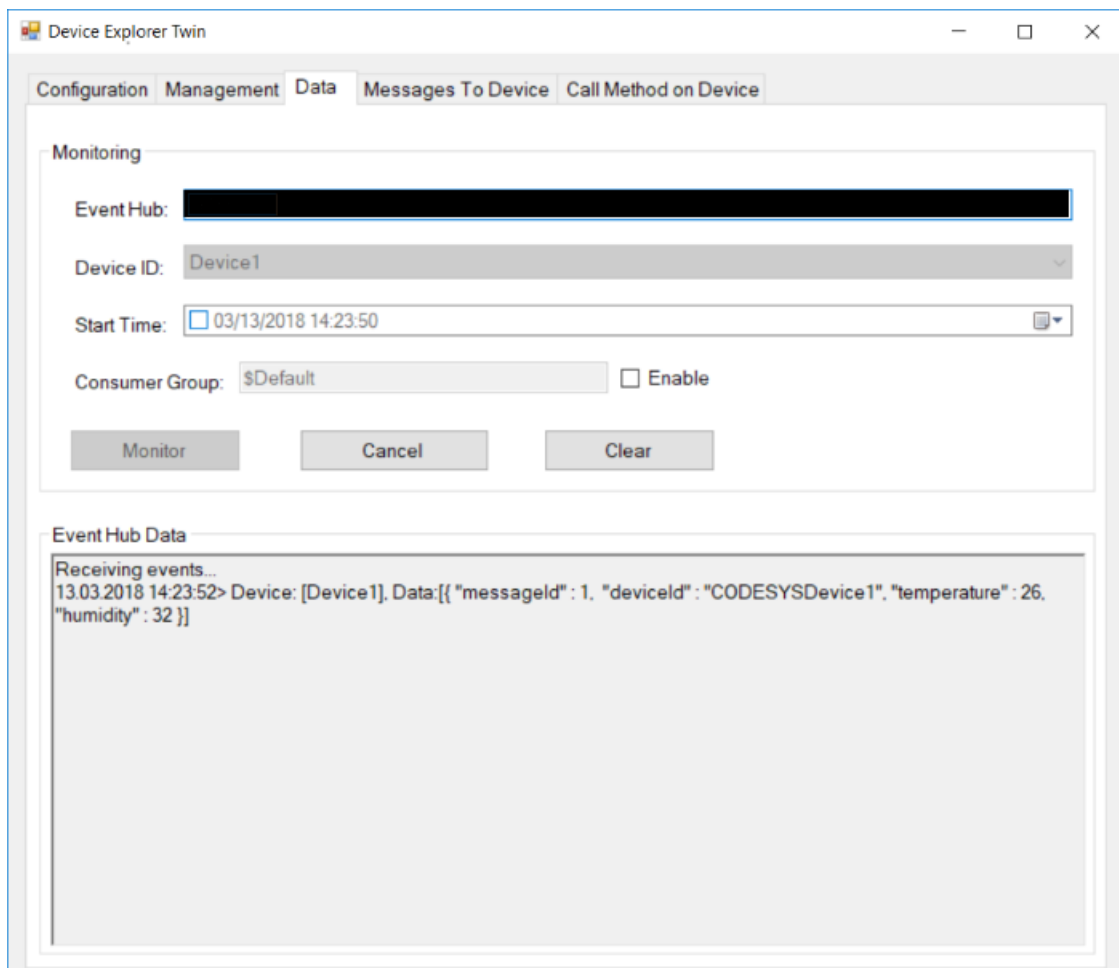
The screenshot displays the Azure IoT Hub Client interface with the following sections:

- Connection Settings:** Includes input fields for Domain name (IoT Hub URL), Device Id (Device2), and SaS Token (IoT Hub URL), along with power and status icons.
- D2C Messages:** Features a "Send D2C Message" button and a text area containing a JSON message: `{ "messageId": 1, "deviceId": "CODESYSDevice1", "temperature": 26, "humidity": 32 }`.
- Get Device Twin:** Includes a "Get Device Twin" button and a text area showing a JSON response for the device twin: `{ "desired": { "version": 9 }, "reported": { "temperature": 24, "version": 82 } }`.
- C2D Messages:** Shows a "Last received message" section with the text "Hello..".
- Update Device Twin:** Includes an "Update Device Twin" button and a text area for "DeviceTwin Properties" containing `{ "temperature": 24 }`.
- Direct method call:** Displays a large yellow circular button with a grey border.
- Subscribe Device Twin (desired):** Shows a "Last received message" section with an empty text area.

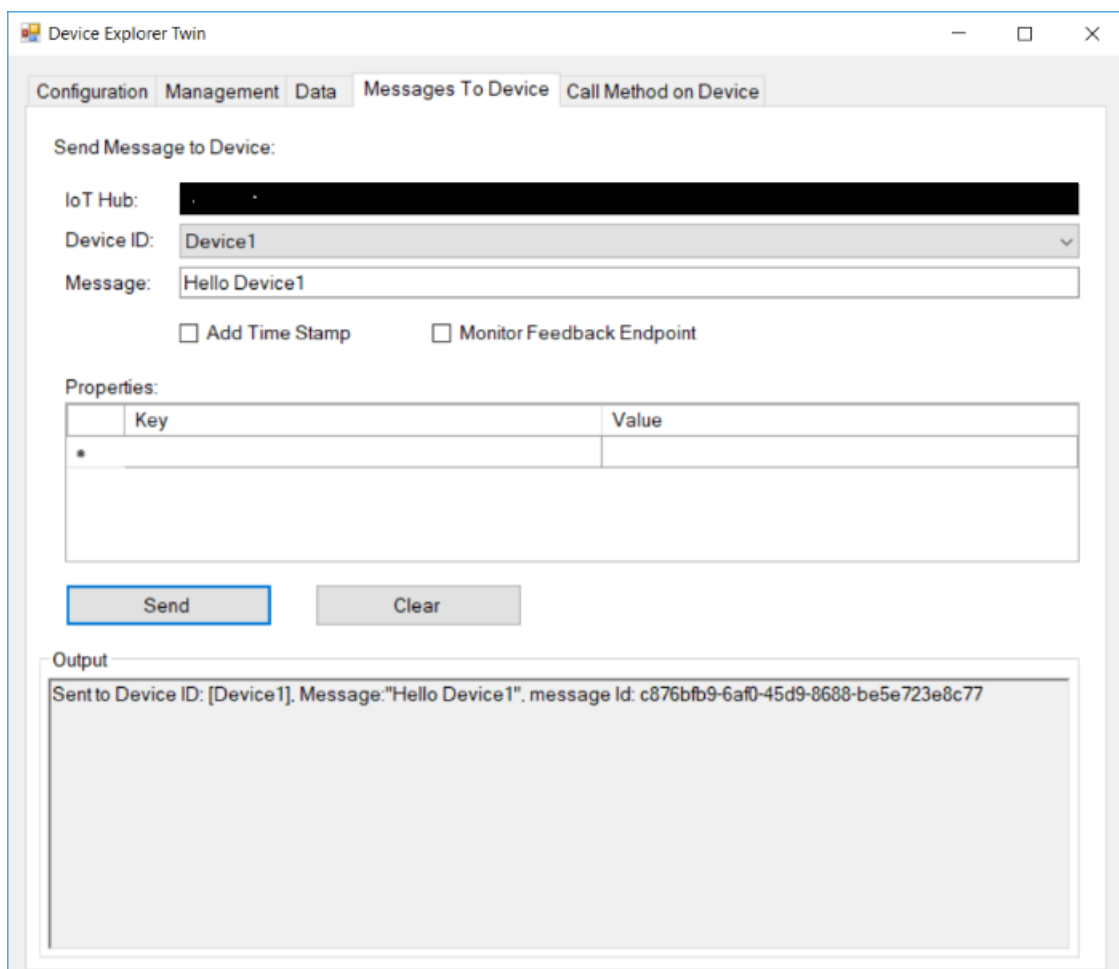
Visualization of the MQTT sample project

6. Sending and receiving messages with the “Device Explorer” tool

Messages can be displayed and sent with the “Device Explorer” tool.



Device Explorer: Receiving D2C messages



Device Explorer: Sending C2D messages

Options

In the Azure environment, messages are usually sent in JSON format. The CODESYS library “JSON Utilities” is ideal for parsing and generating JSON data.

General information

Supplier:

CODESYS GmbH
 Memminger Strasse 151
 87439 Kempten
 Germany

Support:

<https://support.codesys.com>

Item:

Azure IoT Hub Client

Item number:

2111000026

Sales:

CODESYS Store

<https://store.codesys.com>

Included in delivery:

CODESYS package with library and sample project

System requirements and restrictions

Programming system	CODESYS Development System V3.5.13.0 or later
Runtime system	CODESYS Control V3.5.13.0 or later
Supported platforms and devices	Note: Use the "Device Reader" project for locating the functions supported by the PLC. The "Device Reader" project is available in the CODESYS Store free of charge.
Additional requirements	Microsoft Azure account with Azure IoT Hub service; Device Explorer
Restrictions	•
Licensing	Workstation license
Required accessories	CODESYS Key for CODESYS < 3.5.14.0

Note: Not all CODESYS features are available in all territories. For more information on geographic restrictions, please contact sales@codesys.com.

Note: Technical specifications are subject to change. Errors and omissions excepted. The content of the current online version of this document applies.