

Connecting OpcDbGateway to MQTT broker to monitor a home / building automation system.

Vladimir Palacka

SAE - Automation, s.r.o. (Slovakia). Trenčianska 19, NOVA DUBNICA 01851, email:

vladimir_palacka@saeautom.sk

Abstract

The topic of the paper is connecting home/building automation system over OPC client /server to MQTT broker using software OpcDbGateway providing configurable data exchange and processing between different data sources using event of the type “start external application”. The external console application is parameterised by values of memory operands to that data points from different data sources are mapped. This method can be used not only for MQTT but also for outbound communication to other data targets e.g. with REST interface in IoT and IIoT applications. Main advantage is easy configuring of message payloads – none programming for that is needed.

Introduction

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, extremely simple and lightweight messaging protocol, designed for constrained devices and low-bandwidth, high-latency or unreliable networks. The design principles are to minimise network bandwidth and device resource requirements whilst also attempting to ensure reliability and some degree of assurance of delivery. These principles also turn out to make the protocol ideal of the “**machine-to-machine**” (M2M) or “**Internet of Things**” world of connected devices, and for **mobile applications** where bandwidth and battery power are at a premium¹. Security is supported in V3.1 of the protocol by passing user name and password with an MQTT packet. Encryption across the network can be handled with SSL, independently of the MQTT protocol itself. TCP/IP port 1883 is reserved with IANA for use with MQTT. TCP/IP port 8883 is also registered, for using MQTT over SSL. MQTT v3.1.1 is an OASIS Standard.

MQTT communication **can be used to transfer data through firewalls**. Data from a home / building automation system can be published from home to MQTT broker placed in cloud. Different client applications can subscribe for data relevant to them and to be notified when data are changed.

A commonly used way of connecting home / building automation system is OPC DA /UA standard. A communication hub that has an OPC DA client can connect **different types** of home automation systems because commercial OPC servers are on the market for most of them. OpcDbGateway² is such hub and it has also many different functionalities for subsequent processing of collected data gained from OPC DA client or a few other interfaces.

¹ <http://mqtt.org/faq>

² <http://www.saeautom.sk/sk/products/opcdbgateway/>, OpcDbGateway and SAEAUT Universal OPC Server User's guide - http://www.saeautom.sk/download/help/opcdbgateway_en.pdf

Configuring data transfer from HAS /BAS to MQTT broker.

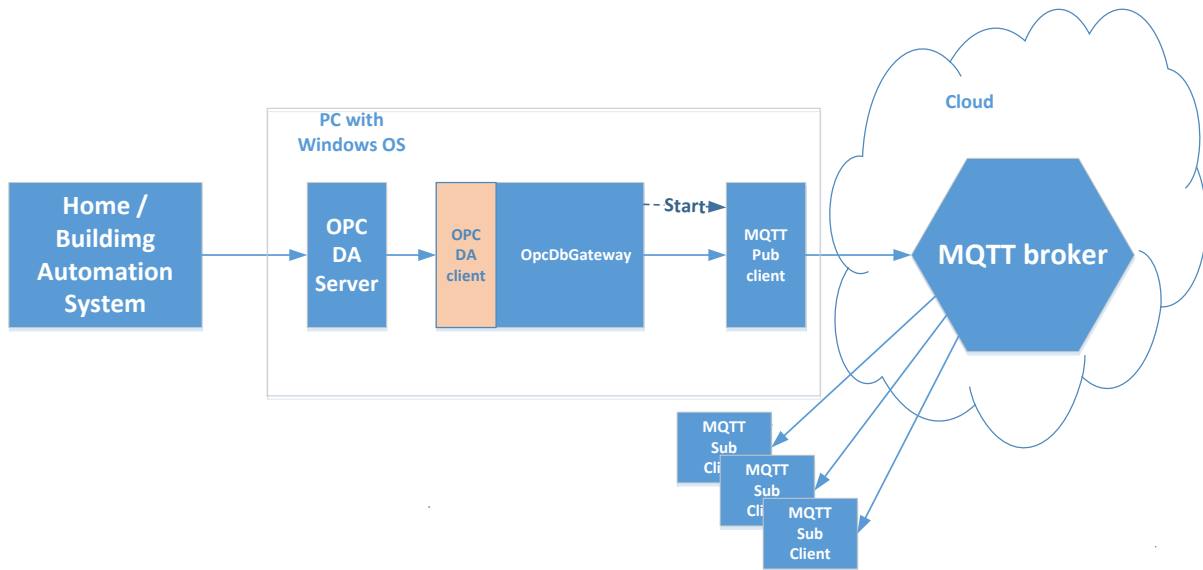


Figure 1 Communication chain to transfer data from Home/Building Automation system to MQTT Broker

The product OpcDbGateway consists of 2 applications for Windows operating systems – a **runtime application (RA)** and a **configuring application (CA)**. The main functionality of RA is transferring data from/to different sources to /from internal memory area of RA named Process Image Memory (PIM). The data in this part of memory are variables with defined data types, quality and timestamps that are named memory operands (MO). OpcDbGateway transfers data between different MOs and this way enables mutual connection different external data sources / destinations.

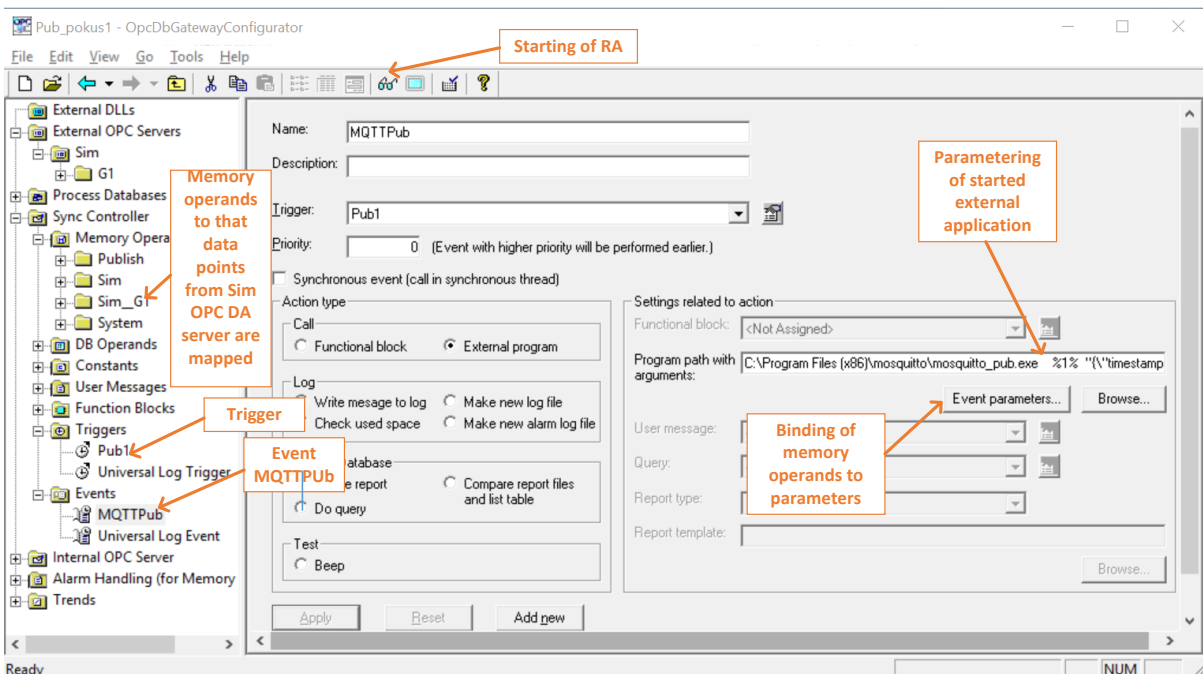


Figure 2 Configuring event to start external application using OpcDbGateway CA

In complicated integrated applications number of data points from different sources can be very high. To work effectively, it is necessary to have CA enabling easy mapping of data points to MO. Most commercial OPC servers are configured using own CA. OpcDbGateway CA uses a browsing interface of OPC to map information about data points from commercial OPC servers' configuration for automatic configuring of data points / MO mapping.

There are two ways in OpcDbGateway to connect with MQTT broker:

- using its own MQTT client enhancing software module for OpcDbGateway,
- using a console external MQTT client application.

Within this paper the second possibility is described. We will use a known MQTT product called Mosquitto³.

OpcDbGateway enables execution **by CA configurable events** that are started according **configurable triggers**. A trigger is a condition (of the type TIME, VALUE or TIME & VALUE) to execute an event. One of the configurable event types is **starting of external program**. This feature enables easy and effective way of creating / integrating applications that are often used in Linux. It is based on the linking of console applications through standard input /output. These applications can be started from a shell script or command line with **different command line parameters**, to put data to the *stdout* and to pipe them through *stdin* to a next application. Because of the possibility to start external applications using configurable commands that are parameterised by names of MOs, this kind of application integration can be provided also by Windows OS application OpcDbGateway⁴. In our case, it will start console application **mosquitto_pub** - an MQTT version 3.1/3.1.1 client for publishing simple messages. It will publish just a single message on a topic and exit. Using periodic trigger to start the event can be provided continuous stream of messages.

The configurable command for publishing to MQTT broker can contain application name with path and command line parameters of the application itself e. g.:

```
C:\Program Files (x86)\mosquitto\mosquitto_pub.exe -h sae33 -t topic2 -m"ttt mmm"
```

where:

-h or *--host* - the host to connect to. Defaults to localhost.

-t or *--topic* - the MQTT topic on which to publish the message.

-m or *--message* – the published message

Instead of *-m* also *-f* can be used to publish a message contained in a file.

In case that message or other parameters contain spaces, they have to be in quotation marks. Special characters have to be preceded with backslash.

Mostly we do not want to transfer repeatedly the same message as in previous example. Instead, we want **to publish actual data from a home automation system to MQTT broker**. Because of this, we need to parameterize the configurable command with names of MO (to that data points are mapped) and from that actual values will be extracted by every execution of *mosquitto_pub.exe*. Let's suppose that we want to transfer message with JSON payload containing one variable for *timestamp* and another for value of *power* used by a device like this:

```
{"timestamp": 10/27/2016 2:23:13 PM, "power": 28669.1785}
```

³ https://mosquitto.org/man/mosquitto_pub-1.html

⁴ However, there can be a problem with catching of *stdout* for using by MOs in OpcDbGateway RA which can be solved by means that are not described in this article.

We can try to test it using own configuration⁵ for OpcDbGateway CA and to use following command for that

```
C:\Program Files (x86)\mosquitto\mosquitto_pub.exe %1% "{\timestamp\": %2%,
\"power\": %3%}"
```

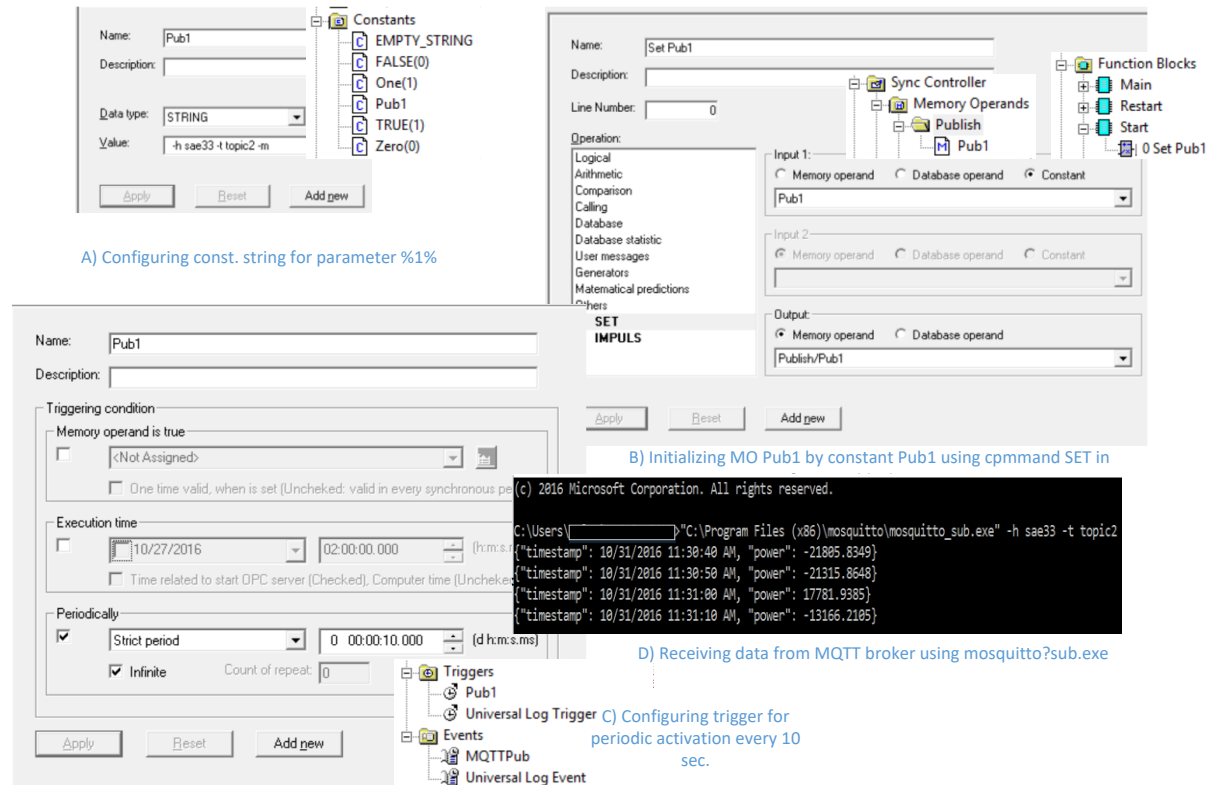


Figure 3 Configuring constant, function block START, trigger and watching running application using mosquitto_sub.

By executing, the parameter %1% is substituted by constant string `-h sae33 -t topic2` that is by initialization (within functional blocks START and RESTART) moved from configurable constant with name *Pub1* to MO with the same name, the parameter %2% is substituted by system time from MO named *PlcSystemTime* and %3% is substituted from MO named *type Double*. Of course, when creating a test application, the naming of variables is up to you. The testing configuration can also be prepared and tested without a commercial OPC server for a home automation system as the principle of configuration / mapping of data points to MO is very similar to using with OpcDbGateway delivered simulation OPC server SAEAutomation.OPCSimDA.3.

The preparation of test configuration consists of following steps (please see details to every step in OpcDbGateway documentation):

1. Create new configuration (see User's guide)
2. Create new External OPC server using SAEAutomation.OPCSimDA.3 (see User's guide)
3. Create new OPC group (see User's guide)
4. Make **automatic** mapping of OPC items from SAEAutomation.OPCSimDA.3 (see User's guide)
5. Create new periodic trigger (with period e.g. 10 s - Figure 3 C)

⁵ OpcDbgateway configuration for the example can be downloaded from http://home.gts.sk/saeautomation//examples/OpcToMqtt/Pub_pokus1.ODG

6. Create a constant with the type "string" with content `-h sae33 -t topic2` (Figure 3)
7. Within function blocks START and RESTART make initialization of a MO with constant from step 6 using configurable command SET (Figure 3 B).
8. Create new event of the type **Start external application** which will use trigger from the step 5, command `C:\Program Files (x86)\mosquitto\mosquitto_pub.exe %1% "{ \"timestamp\": %2%, \"power\": %3%}"` and MOs for actualising parameters `%1%`, `%2%` and `%3%` (Figure 2).
9. Start OpcDbGateway RA from OpcDbGateway CA using icon with glasses (Figure 2).ⁱ
10. To watch running application start MQTT subscription client in console using command `"C:\Program Files (x86)\mosquitto\mosquitto_sub.exe" -h sae33 -t topic2` (Figure 3 D)
11. In case of problems see the view of OPC client in OpcDbGateway CA and view with log files.

Conclusions

The paper explains by example how can be easily provided and configured outbound communication from OpcDbGateway using a specific external communication driver. Within that, data coming to OpcDbGateway PIM from other data sources can be transferred using communication drivers implemented as external console applications. It uses feature of the communication driver that destination as well as message payload can be defined within command line parameters of the communication driver itself. These parameters can be set within the 2nd level of parameterisation – from OpcDbGateway. OpcDbGateway parameters contain names of MOs. Actual values of MOs have to be properly set and transfer as part of command line parameter for message payload and so to be delivered to the destination. In this case, we can just configure (not program) variable communication payloads.

In case that we want to use alike method using event for starting of external communication driver for inbound communication (from source to MOs), we need to find a way to get data from external communication driver process to OpcDbGateway RA process. We can do it e.g. by piping of data from `stdout` through named pipe to OpcDbGateway RA. However, in this case we need specific enhancing software module (DLL) for implementing of the named pipes. It is more complicated to implement it for variable types of message payloads comparing with outbound communication in case that we want completely avoid programming and use only configuring.

i