

Quick Start Guide



Nexto Xpress Compact PLC

SUEN00399 - MQTT communication



Beijer
ELECTRONICS

1 Function and area of use

This document provides guidelines when working with

2 About this Startup document

This Startup document should not be considered as a complete manual. It is an aid to be able to Startup a normal application quickly and easily.

Copyright © Beijer Electronics, 2020

This documentation (below referred to as 'the material') is the property of Beijer Electronics. The holder or user has a non-exclusive right to use the material.

The holder is not allowed to distribute the material to anyone outside his/ her organization except in cases where the material is part of a system that is supplied by the holder to his/ her customer.

The material may only be used with products or software supplied by Beijer Electronics.

Beijer Electronics assumes no responsibility for any defects in the material, or for any consequences that might arise from the use of the material.

It is the responsibility of the holder to ensure that any systems, for whatever applications, which is based on or includes the material (whether in its entirety or in parts), meets the expected properties or functional requirements.

Beijer Electronics has no obligation to supply the holder with updated versions.

Use the following hardware, software, drivers and utilities in order to obtain a stable application:

In this document we have used following software and hardware

- BCS Tools 3.30
- Nexto Xpress

For further information refer to

- Nexto Xpress CPU User's manual ([Click here to download](#))
- [Beijer Electronics knowledge database, HelpOnline](#)

This document and other Startup documents can be obtained from our homepage.

Please use the address support.europe@beijerelectronics.com for feedback about our Quick Start documents.

3 Table of Contents

- 1 Function and area of use.....2
- 2 About this Startup document.....2
- 3 Table of Contents.....3
- 4 Introduction4
- 5 Add MQTTT function your project5
- 6 Test of the MQTT client7
- 7 Check the connection9
- 8 About Beijer Electronics 10
 - 8.1 Contact us 10

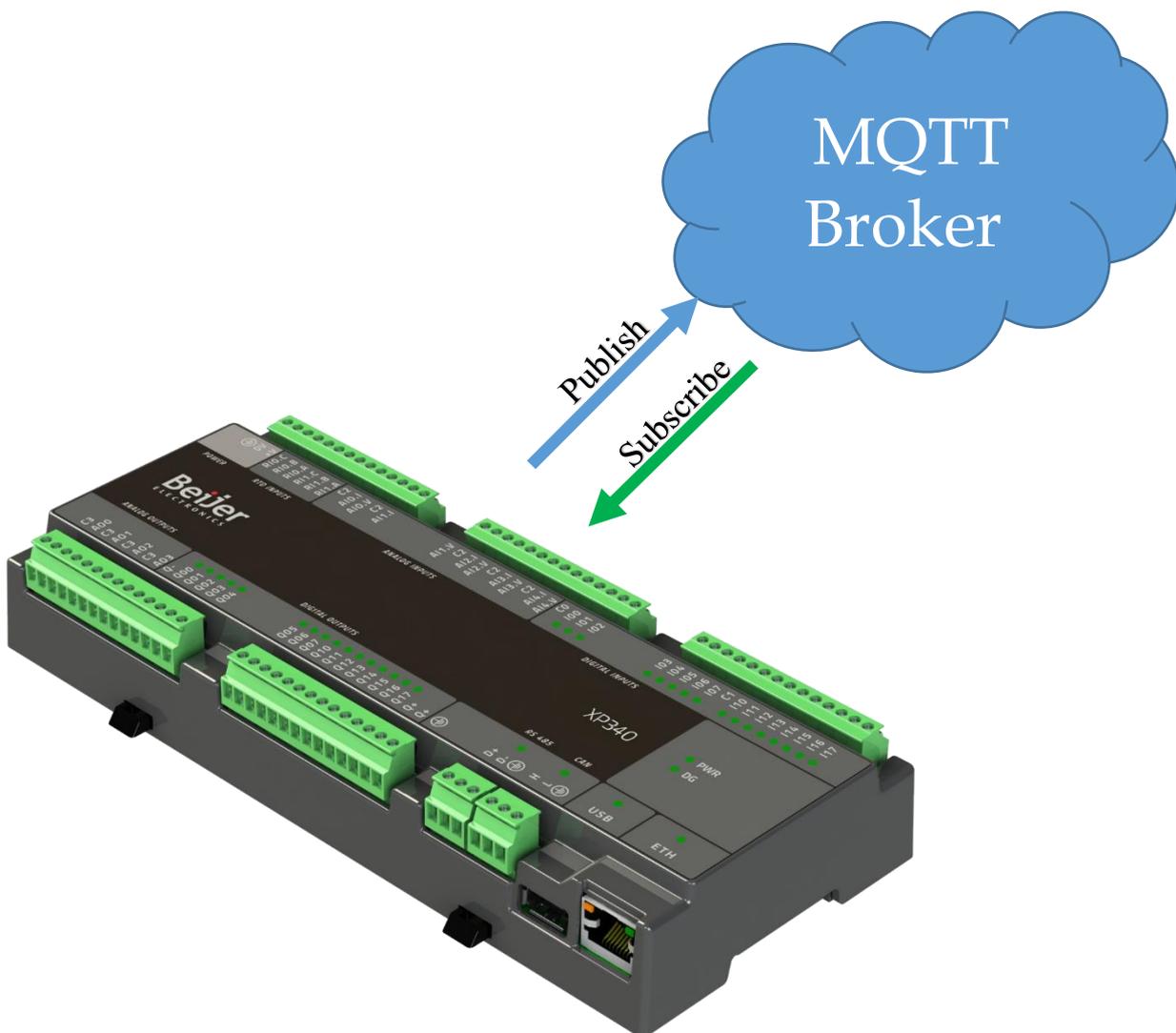
4 Introduction

The Nexto XPress series from Beijer Electronics includes a library (LibMQTT) for communication via 'Message Queuing Telemetry Transport' (MQTT). This is an event-controlled protocol (asynchronous), based on message publishing/ subscription.

Communication takes place between a client (eg Nexto XP325) and a server service (Broker). Once a client has published data to Broker, other clients can subscribe to it.

MQTT is a lightweight protocol that requires little bandwidth. It can also be encrypted and is therefore well suited for communications against cloud services (IIOT).

The code can easily be adapted to the larger PLS system in the Nexto series. It is also similarly easy to convert from other Codesys based PLC systems to Nexto XPress.



5 Add MQTT function your project

Start BCS Tools and open the project where you want to add MQTT communication. In my example 'HelloWorld' project from the basic get started document (KIE00100).

The MQTT functionality is in the library 'LibMQTT'. Add this library via 'Add Library' to 'Library Manager'. Here is a function block, 'MQTT_CLIENT' and associated object structure. It is this function block that establishes communication against brokers. Broker can be online (cloud) or installed on a local PC.

In my example I added 'MQTT_CLIENT' to 'UserPrg', associated object structure as well as variables in 'GVL_MQTT' and required settings in 'PrepareMQTT'

The screenshot displays the BCS Tools interface for a project named 'HelloWorld_XP325'. The 'Library Manager' on the left shows the 'GVL_MQTT' library selected. The main workspace is divided into several panes:

- Configuration (NP):** Shows the 'MQTT_CLIENT' function block with its object structure:
 - ENABLE: bEnable
 - CONNECTION_CONFIG: CONN
 - SUBSCRIBE: SUBS
 - CONNECTED: CONNECTED
 - STATE: STATE
 - ERROR: ERROR
- Code Editor:** Shows the 'PrepareMQTT' program with the following code:


```

1 PROGRAM PrepareMQTT
2
3   VAR
4     Puls_3s_Edge: TON;
5     MyMachineTemperature: INT;
6   END_VAR
7
8   // Setter nødvendige parametre
9   CONN.sClientId := 'MyID';
10  CONN.sHostname := '10.0.11.76';
11
12  SUBS[1].sSubscribeTopic := 'MyMachineID1234';
13
14  SUBS[1].pbPayloadBuffer := ADR(RcvBuffer);
15  SUBS[1].uiMaxPayloadSize := SIZEOF(RcvBuffer);
16
17  PUBS[1].bEnablePublish := PubEnable;
18  PUBS[1].sPublishTopic := 'MyMachineID1234';
19  PUBS[1].pbPublishPayload := ADR(PubMessage);
20  PUBS[1].uiPublishPayloadSize := SIZEOF(PubMessage);
21
22  // Simuler noen verdier hvert 3 sec
23  Puls_3s_Edge(IN := NOT Puls_3s_Edge.Q, PT := T#3S);
24
25  IF Puls_3s_Edge.Q THEN
26    MyMachineTemperature := OSCAT_BASIC.RIM2(last := MyMachineTemperature, low:=37, high :=89);
27    PubMessage := 'MyMachineTemperature=';
28    PubMessage := CONCAT(PubMessage, INT_TO_STRING(MyMachineTemperature));
29  END_IF
            
```
- Global Variables:** Shows the 'GVL_MQTT' library with the following global variables:
 - CONN: MQTT_CONN_CONFIG;
 - SUBS: ARRAY [1..gc_uiMaxSubs] OF MQTT_SUBSCRIBE;
 - PUBS: ARRAY [1..gc_uiMaxPubs] OF MQTT_PUBLISH;
 - Rcv: ARRAY [1..gc_uiMaxSubs] OF MQTT_RECEIVED;
 - RcvBuffer: STRING := ''; // This can be of any type (e.g. STRING, DWORD, BYTE...)
 - PubEnable: BOOL := FALSE;
 - PubMessage: STRING := 'MyTestData';
 - MyMachineData: MyMachineData;
- Library Parameters:** Shows the parameters for the MQTT library:

Name	Type	Value (editable)	Comment
gc_uiMaxSubs	UINT	2	Maximum number of subscribers
gc_uiMaxPubs	UINT	2	Maximum number of publishers

The MQTT client is controlled by five objects. I have only configured what is necessary. The complete manual for the MQTT client (MU214606) contains a list of all variables and possibilities with the MQTT client, Including encryption.

‘MQTT_CONN_CONFIG’:

Contains parameters and variables for connection to broker. In my example, variable 'CONN' is defined against this object. I have changed two variables; the rest are default values.

```
CONN.sClientId := 'MyID';           // ID identifikasjon at broker
CONN.sHostname := '10.0.11.76';    // Hostnavn or IP adresse for broker (my PC)
```

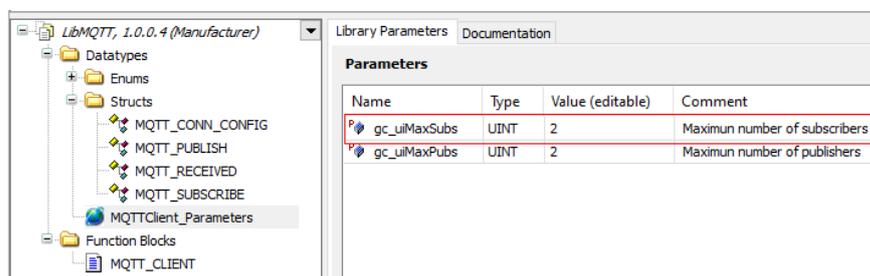
‘MQTT_SUBSCRIBE’:

Contains parameters and variables to subscribe to data from a broker. In my example, variable 'SUBS' is defined against this object. I use Index 1 (subscription number 1). I've changed three variables, the rest are default values

```
SUBS[1].sSubscribeTopic := 'MyMachineID1234';           // Name of variable I want to subscribe
to
SUBS[1].pbPayloadBuffer := ADR(RcvBuffer);             // Point to buffer for reception
SUBS[1].uiMaxPayloadSize := SIZEOF(RcvBuffer);         // Max receive size
```

The number of concurrent subscriptions is controlled by a constant. By default, this is set to 2, but this parameter can be changed in the library. NB! Remember that memory is allocated for each instance, so do not set it higher than needed.

ARRAY [1..ui_gcMaxSubs] OF MQTT_SUBSCRIBE

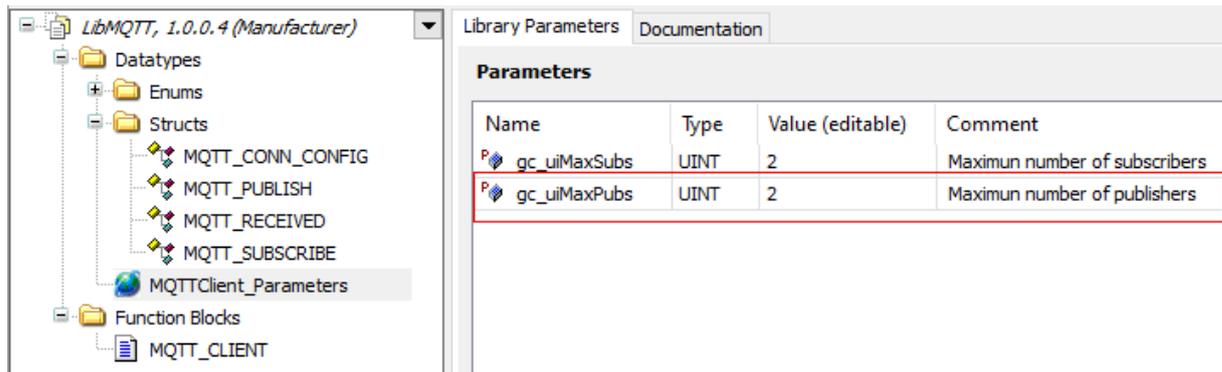
**‘MQTT_PUBLISH’:**

Contains parameters and variables for publishing data to Broker. In my example, variable 'PUBS' is defined against this object. I use Index 1 (publication number 1). I've changed four variables; the rest are default values.

```
PUBS[1].bEnablePublish := PubEnable;           // Message is sent to broker on positive flank
PUBS[1].sPublishTopic := 'MyMachineID1234';    // Name of variable I want to send
PUBS[1].pbPublishPayload := ADR(PubMessage);   // Point to variable I want to send
PUBS[1].uiPublishPayloadSize := SIZEOF(PubMessage); // Size of publish message
```

The number of simultaneous publications is controlled by a constant. By default, this is set to 2, but this parameter can be changed in the library. NB! Remember that memory is allocated for each instance, so do not set it higher than needed.

ARRAY [1..ui_gcMaxPubs] OF MQTT_PUBLISH

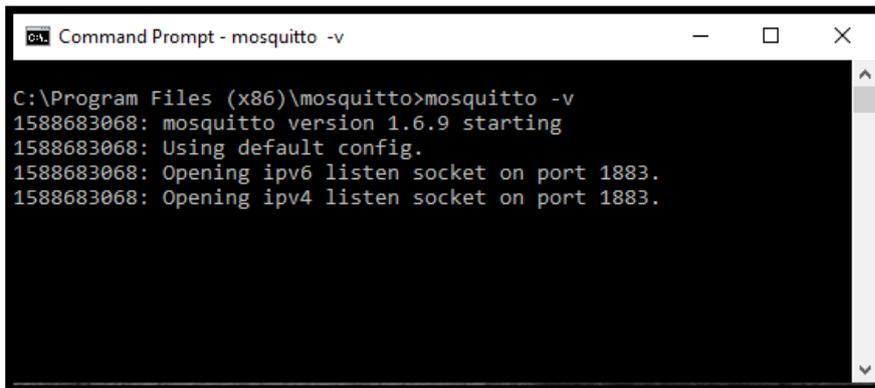


‘MQTT_RECEIVED’:

Contains variables for data received from a broker as a result of a subscription. In my example, variable 'RCV' is defined against this object. No variables are changed from default values.

6 Test of the MQTT client

To test the example, use a local installation of Mosquitto from the Eclipse Foundation as a broker. The program can be downloaded from either Eclipse or GitHub. After installation, open a command window and launch Mosquitto with command 'mosquitto -v'.



This starts the broker with default configuration. Launch the program with the command 'mosquitto -help' for further information.

Mosquitto also includes a publishing program (mosquitto_pub) and a subscription program (mosquitto_sub). The 'mosquitto_sub', I use as an extra client to better visualize the test.

```

Command Prompt - mosquitto_sub -h 10.0.11.76 -p 1883 -i "MyID2" -t "MyMachineID1234"
C:\Program Files (x86)\mosquitto>mosquitto_sub -h 10.0.11.76 -p 1883 -i "MyID2" -t "MyMachineID1234"
    
```

Now the broker, as well as the test client are online, and we can download and start our pls project.

The connection from PLC to broker is established by setting 'bEnable'. In the log we see the connection created from IP 10.0.11.64 with 'MyID' with the subscription to variable 'MyMachineID1234' (Item 1).

The connection from the 'mosquitto_sub' client is then logged with 'MyId2' and the same subscription (point 2).

'PubEnable' is triggered every 5 seconds in PLC and publishes a value to the 'MyMachineID1234' variable. Broker publishes this to the clients who have it subscribed (point 3).

Expression	Value
CONN	MQTT_CLIENT
SUBS	MyMachineID1234
SUBS[1]	MyMachineID1234
SUBS[2]	MyMachineID1234
PUBS	MyMachineID1234
PUBS[1]	MyMachineID1234
PUBS[2]	MyMachineID1234
RCV	MyMachineID1234
RcvBuffer	MyMachineTemperature = 51
PubEnable	TRUE
PubMessage	MyMachineTemperature = 51
RcvMessage	

```

mosquitto_Broker
1588766482: mosquitto version 1.6.9 starting
1588766482: Using default config.
1588766482: Opening ipv6 listen socket on port 1883.
1588766482: Opening ipv4 listen socket on port 1883.
1588766501: New connection from 10.0.11.64 on port 1883.
1588766501: New client connected from 10.0.11.64 as MyID (p1, c1, k60).
1588766501: No will message specified.
1588766501: Sending CONNACK to MyID (0, 0)
1588766501: Received SUBSCRIBE from MyID
1588766501:   MyMachineID1234 (QoS 0)
1588766501: MyID 0 MyMachineID1234
1588766501: Sending SUBACK to MyID
1588766511: New connection from 10.0.11.76 on port 1883.
1588766511: New client connected from 10.0.11.76 as MyID2 (p2, c1, k60).
1588766511: No will message specified.
1588766511: Sending CONNACK to MyID2 (0, 0)
1588766511: Received SUBSCRIBE from MyID2
1588766511:   MyMachineID1234 (QoS 0)
1588766511: MyID2 0 MyMachineID1234
1588766511: Sending SUBACK to MyID2
1588766518: Received PUBLISH from MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588766518: Sending PUBLISH to MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588766518: Sending PUBLISH to MyID2 (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
    
```

```

C:\Program Files (x86)\mosquitto>mosquitto_sub -h 10.0.11.76 -p 1883 -i "MyID2" -t "MyMachineID1234"
MyMachineTemperature = 51

```

7 Check the connection

Clients who subscribe to data periodically send a 'PINGREQ' message to the broker to check that the connection is in order. Broker responds to this with a 'PINGRESP' message.

```

mosquitto_Broker
1588770619: Sending CONNACK to MyID2 (0, 0)
1588770619: Received SUBSCRIBE from MyID2
1588770619:   MyMachineID1234 (QoS 0)
1588770619: MyID2 0 MyMachineID1234
1588770619: Sending SUBACK to MyID2
1588770629: Received PUBLISH from MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770629: Sending PUBLISH to MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770629: Sending PUBLISH to MyID2 (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770634: Received PUBLISH from MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770634: Sending PUBLISH to MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770634: Sending PUBLISH to MyID2 (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770639: Received PUBLISH from MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770639: Sending PUBLISH to MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770639: Sending PUBLISH to MyID2 (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770644: Received PUBLISH from MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770644: Sending PUBLISH to MyID (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770644: Sending PUBLISH to MyID2 (d0, q0, r0, m0, 'MyMachineID1234', ... (27 bytes))
1588770679: Received PINGREQ from MyID2
1588770679: Sending PINGRESP to MyID2
1588770703: Received PINGREQ from MyID
1588770703: Sending PINGRESP to MyID
1588770739: Received PINGREQ from MyID2
1588770739: Sending PINGRESP to MyID2
1588770763: Received PINGREQ from MyID
1588770763: Sending PINGRESP to MyID
1588770799: Received PINGREQ from MyID2
1588770799: Sending PINGRESP to MyID2
1588770824: Received PINGREQ from MyID
1588770824: Sending PINGRESP to MyID

```

8 About Beijer Electronics

Beijer Electronics is a multinational, cross-industry innovator that connects people and technologies to optimize processes for business-critical applications. Our offer includes operator communication, automation solutions, digitalization, display solutions and support. As experts in user-friendly software, hardware

and services for the Industrial Internet of Things, we empower you to meet your challenges through leading-edge solutions.

Beijer Electronics is a Beijer Group company. Beijer Group has a sale over 1.6 billion SEK in 2019 and is listed on the NASDAQ OMX Nordic Stockholm Small Cap list under the ticker BELE. www.bejergroup.com

China

Shanghai

NORWAY

Drammen

TAIWAN

Taipei

DENMARK

Roskilde

SOUTH KOREA

Seoul

TURKEY

Istanbul

FRANCE

Paris

SWEDEN

Göteborg
Malmö
Stockholm

UNITED KINGDOM

Nottingham

GERMANY

Nürtingen

USA

Salt Lake City

8.1 Contact us

[Global offices and distributors](#)

Beijer Electronics AB – a *Beijer Electronics Group* company

Head Office

Beijer Electronics AB
P.O. Box 426, Stora Varvsgatan 13a
SE-201 24 Malmö, SWEDEN
Telephone +46 40 35 86 00

Subsidiaries

[Click here for details](#)