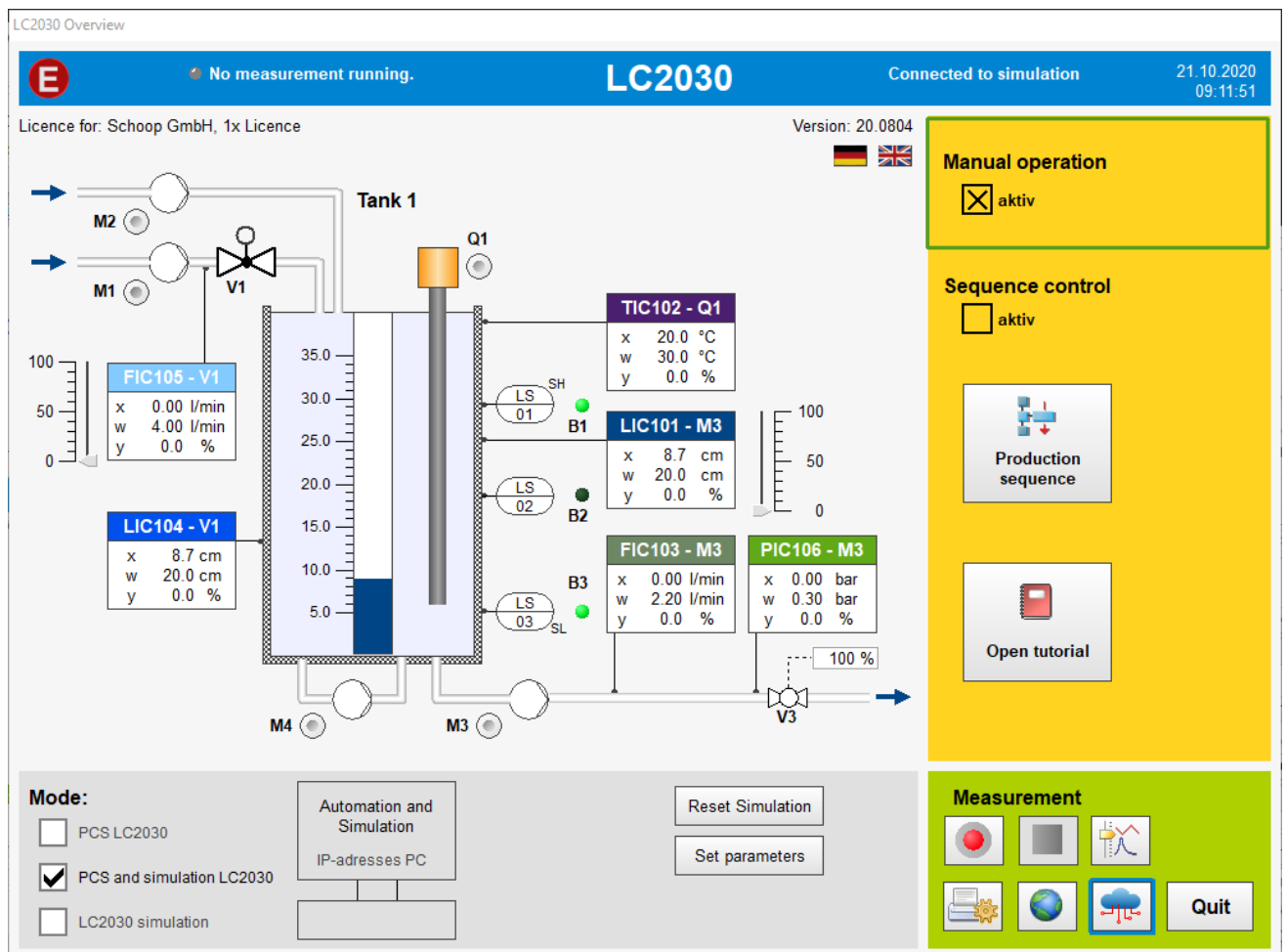


TASKS FOR THE USAGE OF LC2030 TRAINING WITH MQTT



Content

1	INTRODUCTION.....	3
2	TASKS WITH THE MQTT EXPLORER.....	3
3	TASKS FOR CONNECTION SECURITY	4
4	TASKS WITH APP	7
4.1	TASKS FOR MANUAL OPERATION VIA MQTT	7
4.2	TASKS WITH SEQUENCE CONTROL	9

This document is protected by copyright. All rights reserved, including the translation, reprinting and duplication of the work or parts thereof. No part of this publication may be reproduced, duplicated or distributed in any form without the written permission of Ingenieurbüro Dr.-Ing. Schoop GmbH

1 INTRODUCTION

In the LC2030 Training an MQTT Client is activated via cloud symbol.



The overview site of MQTT usage offers the choice between manual operation and sequence control. In manual operation it is possible to start and configure all pumps, heating rod and control values. In sequence control mode a production sequence can be started and monitored.

The Broker with which the program connects is set up in the background window called WRP-Serf. A detailed description is in the document MQTT_activation.pdf

For the following tasks a local mosquitto broker was installed. For a first testing the program MQTT explorer and later the App IoTMQTTPanel were used.

2 TASKS WITH THE MQTT EXPLORER

TASK 1.1 Set up your local broker and test if it's possible to establish a connection. Establish the connection also with the MQTT explorer and check if you can see level, flow, temperature and pressure.

TASK 1.2 Insert a diagram for flow and level.

SOLUTION

To insert a diagram you have to mark the according topic and click the wave symbol in the right column.

```

- "value": 6.708870018220803e-38,
- "ts": 1602150667150
~+ "value": 5.492775512917979e-38,
~+ "ts": 1602150667251

```

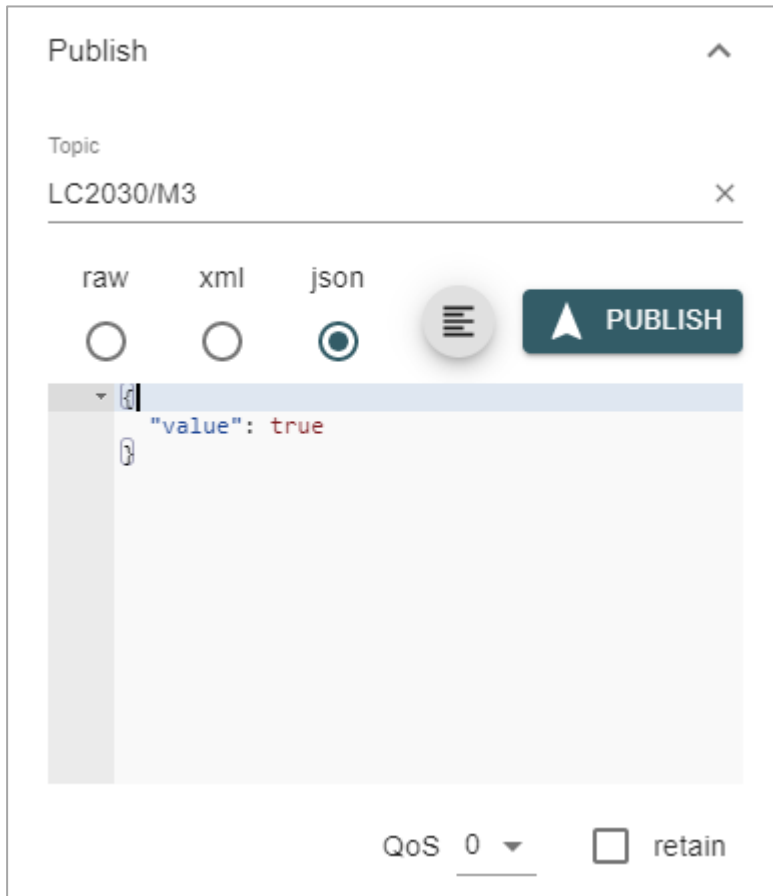


TASK 1.3 Switch on a pump with the MQTT Explorer.

SOLUTION

Go to the right column below *Publish*. Insert the desired topic and the json code which sets the signal to true.

The code is: { "value": true }.



Publish

Topic
LC2030/M3

raw xml json

☐ ☐ ☒

```
{ "value": true }
```

QoS 0 ☐ retain

3 TASKS FOR CONNECTION SECURITY

TASK 3.1 Explain why it's reasonable to secure the connection.

SOLUTION

With a not secured connection you can't be sure that:

- No one reads your data,
- Your data hasn't been changed before you read it,
- The connection was established to the correct server.

TASK 3.2 Which security mechanisms are possible with MQTT protocol?

SOLUTION

- SSL/TLS Description: Uses TCP/IP protocol not MQTT, clients have to support this protocol.
- User Authentication: There are three possibilities
 - Client IDs, which are set up in the broker, it's also possible to specify special user access.

- User name and password
- The safest method is client certificates. This is very time consuming, because all client certificates have to be managed manually.
- Message encryption: messages are *end to end* encrypted by the client. Therefore it's independent from the broker.
-

TASK 3.3 Setup a user with password on your local mosquitto broker.

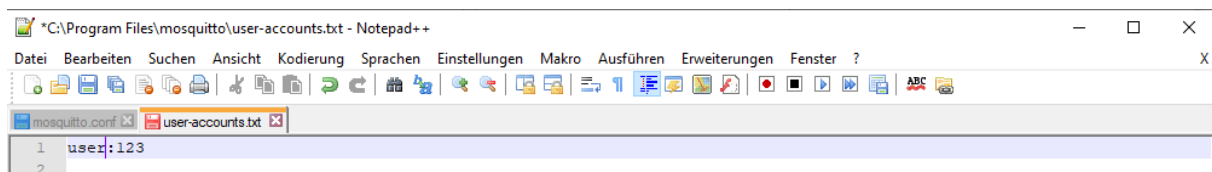
SOLUTION

Create a text file in you mosquitto folder (user-accounts.txt) with following content:

User 1:Password 1

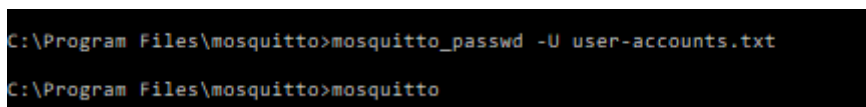
User2:Password 2

Etc.

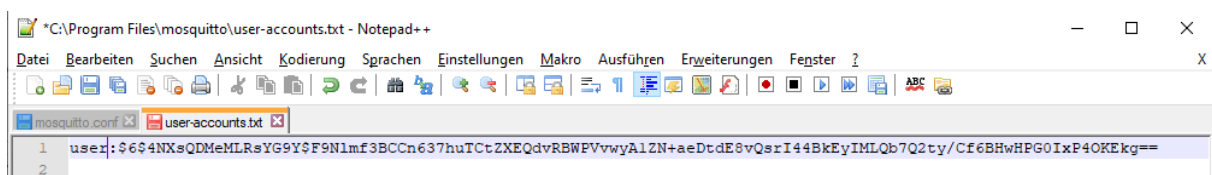


To encrypt this file open the terminal with administrator rights. Change directory to mosquitto folder and insert following line:

Mosquitto_passwd -U user-accounts.txt



Check if the file was encrypted.



The user-account.txt file has to be inserted to the mosquitto configuration file. Open *mosquitto.conf* and set the option *allow_anonymous* to *false* and enter user-accounts.txt behind *password_file*.

```

646 # Defaults to true if no other security options are set. If `password_file` or
647 # `psk_file` is set, or if an authentication plugin is loaded which implements
648 # username/password or TLS-PSK checks, then `allow_anonymous` defaults to
649 # false.
650 #
651 allow_anonymous false
652 #
653 # -----
654 # Default authentication and topic access control
655 # -----
656 #
657 # Control access to the broker using a password file. This file can be
658 # generated using the mosquitto_passwd utility. If TLS support is not compiled
659 # into mosquitto (it is recommended that TLS support should be included) then
660 # plain text passwords are used, in which case the file should be a text file
661 # with lines in the format:
662 # username:password
663 # The password (and colon) may be omitted if desired, although this
664 # offers very little in the way of security.
665 #
666 # See the TLS client require_certificate and use_identity_as_username options
667 # for alternative authentication options. If an auth_plugin is used as well as
668 # password_file, the auth_plugin check will be made first.
669 password_file C:\Program Files\mosquitto\user-accounts.txt

```

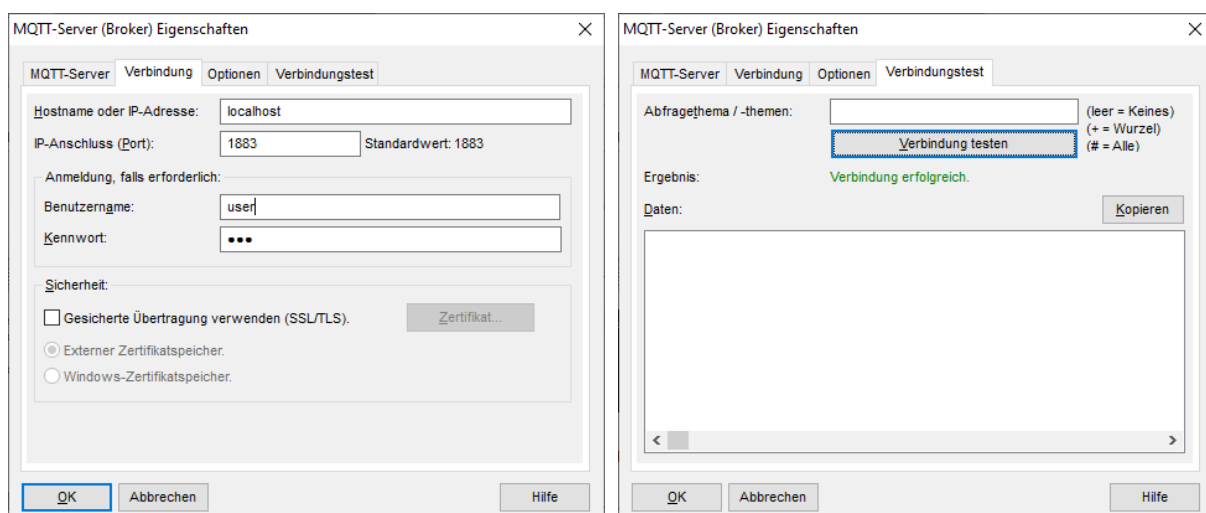
When starting mosquitto broker again load the configuration with:
Mosquitto -v -c mosquitto.conf

```

C:\Program Files\mosquitto>mosquitto -v -c mosquitto.conf
1602159466: mosquitto version 1.6.12 starting
1602159466: Config loaded from mosquitto.conf.
1602159466: Opening ipv6 listen socket on port 1883.
1602159466: Opening ipv4 listen socket on port 1883.
1602159466: mosquitto version 1.6.12 running

```

You can check the settings in the background window WRTSP-Serv. Go to the broker configuration (Settings->Options and Settings, last tab is MQTT client). Check the broker with defined user. Also test a wrong password, if no user is set up in the broker there will be no error.



Now log on in the MQTT explorer.

4 TASKS WITH APP

There are many apps to read and publish MQTT messages. The following tasks were performed with the android app *IoT MQTT Panel*. Via WLAN the mobile device is in the same network as the PC with running mosquitto broker.

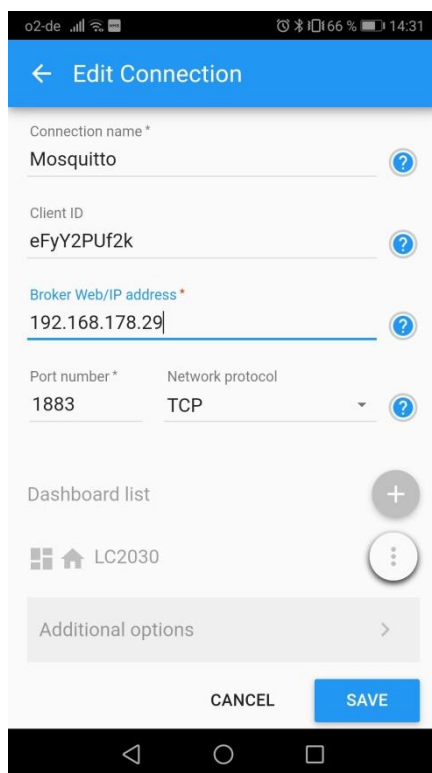
4.1 TASKS FOR MANUAL OPERATION VIA MQTT

For following tasks change the mode to manual mode.

Manual operation



TASK 4.1.1 Establish a connection between app and broker.



SOLUTION

Add a new connection. The displayed input mask opens.
Under *Broker Web/IP address* insert IP address of PC on which the broker is running.
Under *Additional options* insert user and password.

TASK 4.1.2 Set up three diagrams

- Level (LI101)
- Temperature (TI102)
- Flow rate (FI103 und FI105)

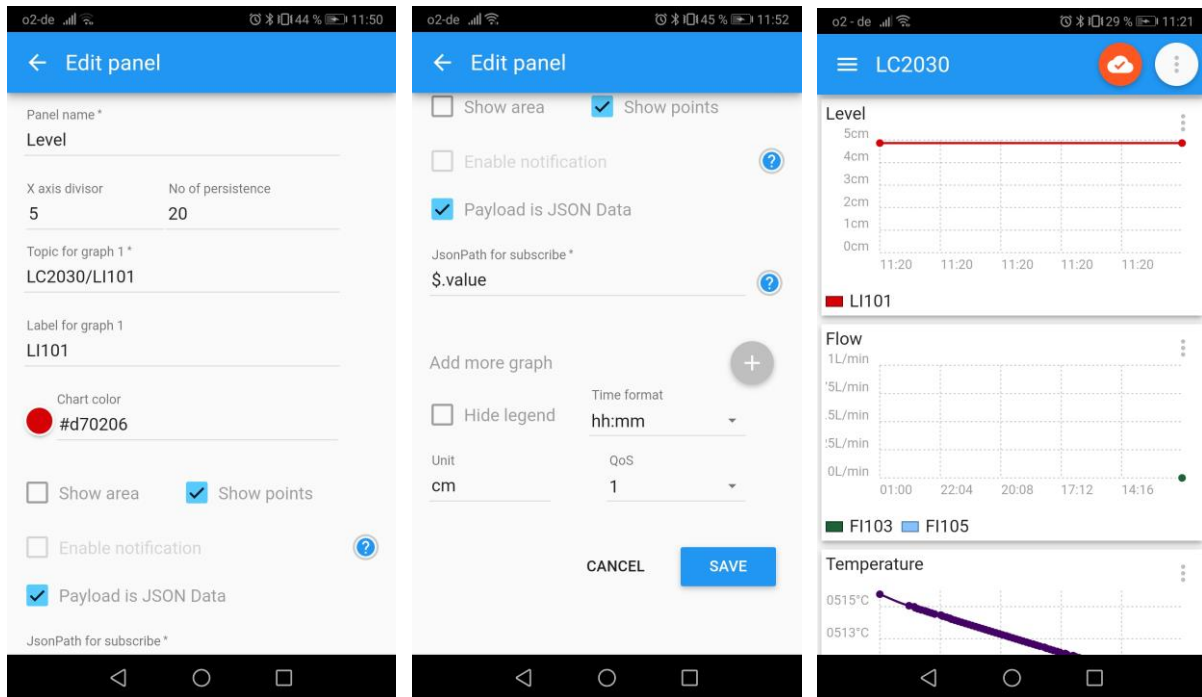
SOLUTION

Below you can see the input mask for the level diagram. Name, label and colour can be chosen freely.

The topics are:

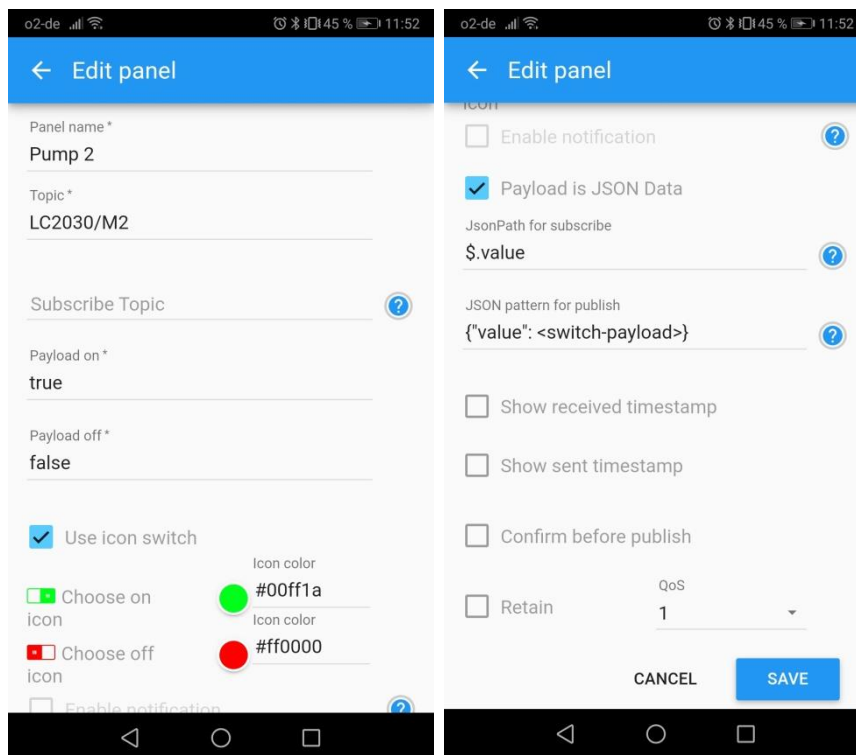
- LC2030/LI101
- LC2030/TI102
- LC2030/FI103 and LC2030/FI105

It's important to check *Payload is json data* and the *Json Path \$.value*.



TASK 4.3 Insert switches for the pumps and sliders for control value of valve and pump.

SOLUTION



The figures show the setup for pump switches.

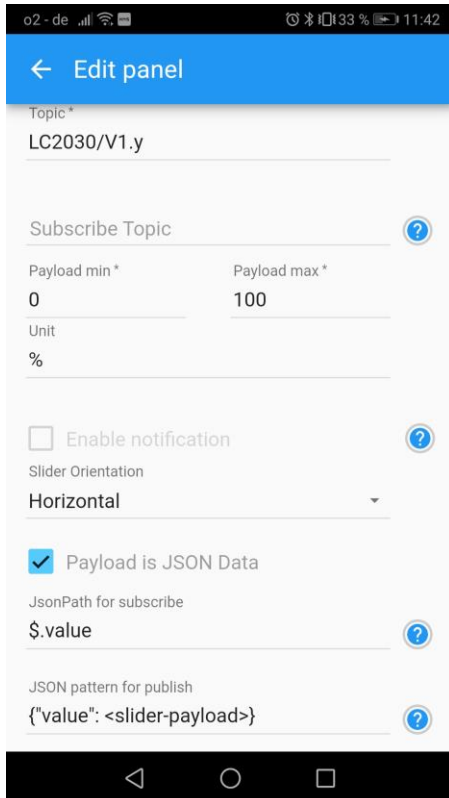
The *topics* are

- LC2030/M1
- LC2030/M2
- LC2030/M3

Payload on is the message value which is sent when switching on. It has to be true. *Payload off* has to be false accordingly.

The *json Path* is *\$.value*.

The *json pattern for publish* is *{\"value\": <switch-payload>}*.



o2 - de 11:42

← Edit panel

Topic*

LC2030/V1.y

Subscribe Topic ?

Payload min *

0

Payload max *

100

Unit

%

☐ Enable notification ?

Slider Orientation

Horizontal

☒ Payload is JSON Data

JsonPath for subscribe

\$.value ?

JSON pattern for publish

{"value": <slider-payload>} ?

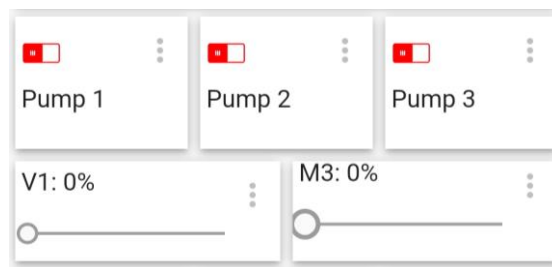
The slider setup is displayed in the figure.

The topics are:

- LC2030/P3.y
- LC2030/V1.y

The json Path is \$.value and the json pattern for publish is {"value": <slider-payload>}.

The result could look like:



4.2 TASKS WITH SEQUENCE CONTROL

A production sequence was realised with GRAFCET. It can be viewed with the button *Production sequence*. It's activated when the sequence control mode is active.

Sequence control

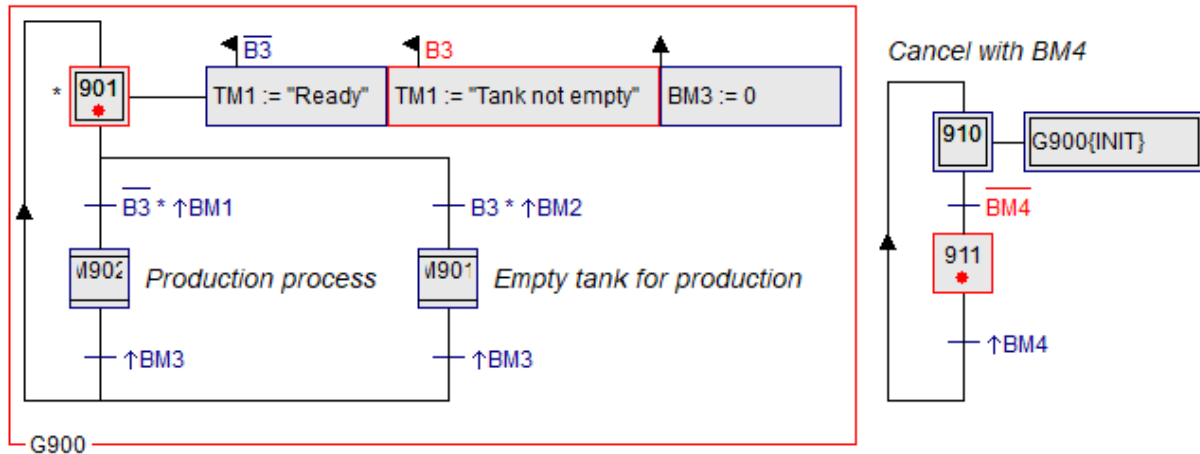


The production sequence is built with two macros which are started by a subordinated sequence. One macro is the production and the other is the preparatory emptying of the tank.

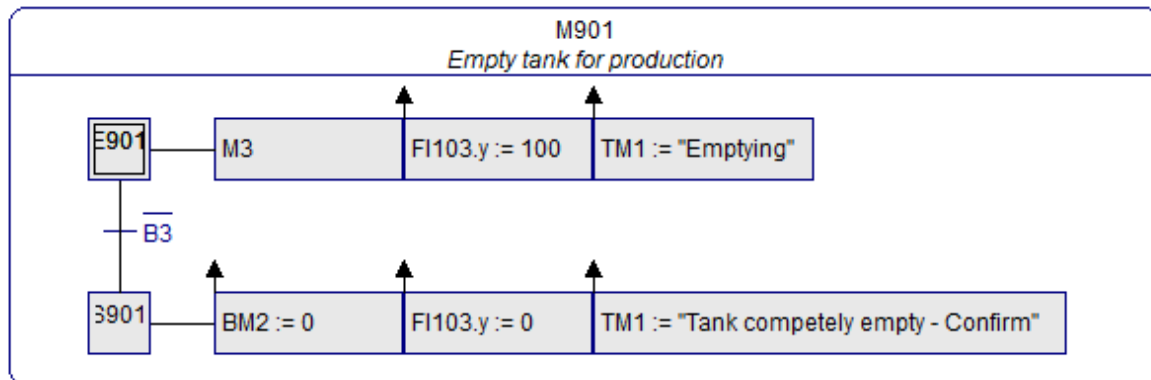
To start, confirm and cancel binary flags are used (BM1, BM2, BM3, BM4), which can be set in the app and which are reset in the GRAFCET diagram. A text signal (TM1) is used as a status signal.

To cancel the sequence binary flag B4 is used. With a positive flag of this signal the subordinated sequence is set to initial step. Only when BM4 is not active the sequence can be started.

In the initial step the status signal is set to „Ready“ or „Tank not empty“ according to level switch B3. From here the production can be started with BM1 or the tank can be emptied with BM2. When the selected macro finished the control waits for confirmation BM3.



The emptying sequence has two steps. First pump M3 is switched on and meanwhile the control signal of the pump is set to 100%. Status is „Emptying“. When level switch B3 is open BM2 and control signal are reset. Pump M3 is not active. Status is „Tank empty - Confirm“. The control waits for confirmation.



The production sequence has six steps. First pump M2 is switched on, the status signal is „Filling fluid 1“. When level switch B2 (centre) is reached the control goes to the next step.

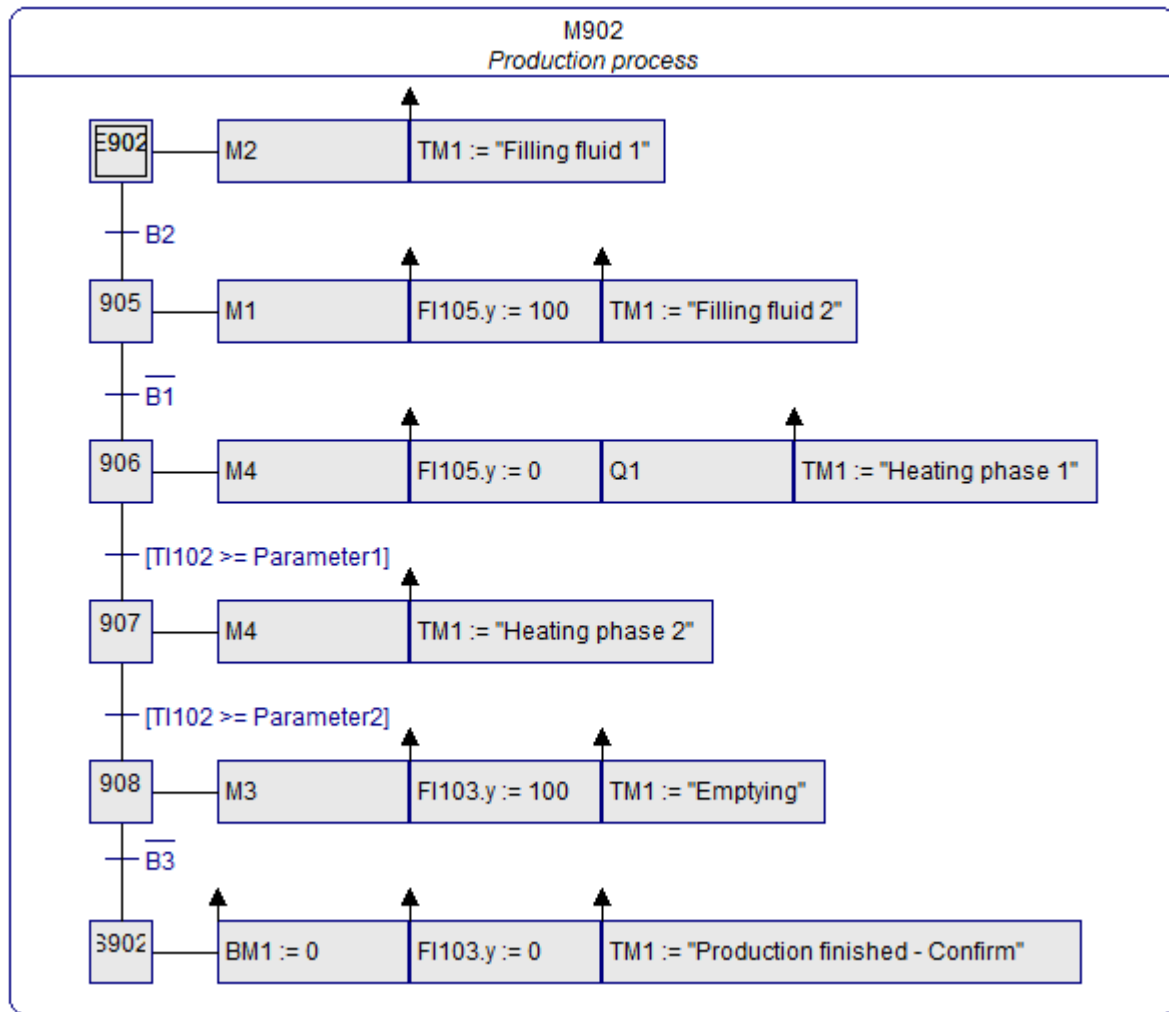
Next pump M1 is activated and meanwhile the valve V1 is opened 100%. The status is “Filling fluid 2“. When the tank is filled up (level switch B1), the control goes to the next step.

In the next step is valve is closed. Meanwhile the heating rod and the circulation pump M4 are switched on. The status is „Heating phase 1“. The step is finished, once a pre-defined temperature is reached. The temperature can be set via Parameter 1 which has the topic *LC2030/Par1*.

Then the heating rod is not active anymore, but because of it’s residual heat the temperature continues increasing, until another temperature limit (*LC2030/Par2*) is reached. The status is „Heating phase 2“.

In the next step pump M3 is active and it’s power is set to 100%. The Status is “Emptying“. When the tank is empty (level switch B3) the control goes to the final step.

Last flag BM1 und pump power are reset. Status is „Production finished - Confirm“. The control waits for confirmation.



Following signals are used in this control.

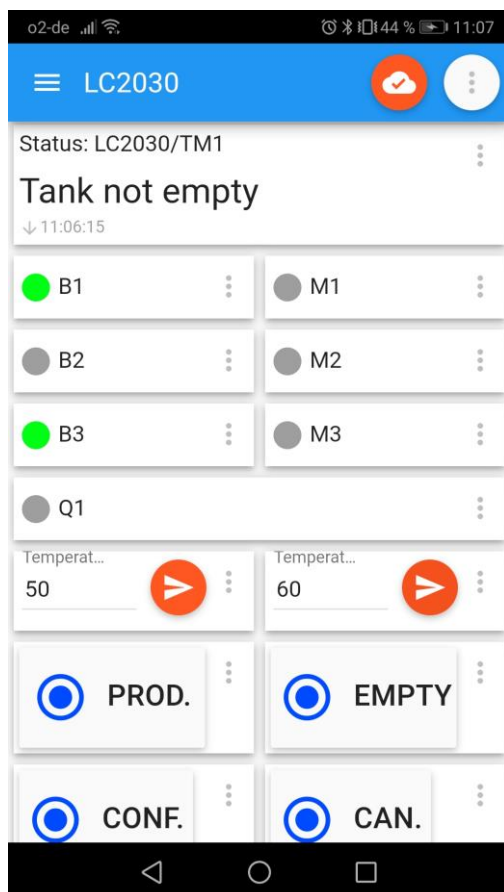
Signal name	Description	Type	Topic
H.M1	Pump M1 on/off	Status	LC2030/M1
H.M2	Pump M2 on/off	Status	LC2030/M2
H.M3	Pump M3 on/off	Status	LC2030/M3
B1	Level switch top	Status	LC2030/B1
B2	Level switch centre	Status	LC2030/B2
B3	Level switch bottom	Status	LC2030/B3
Q1	Heating rod on/off	Status	LC2030/Q1
BM1	Start production	Input	LC2030/BM1
BM2	Start empty tank	Input	LC2030/BM2
BM3	Confirmation	Input	LC2030/BM3
BM4	Cancel	Input	LC2030/BM4
TM1	Status as Text	Input	LC2030/TM1
Parameter1	Temperature limit 1	Input	LC2030/Par1
Parameter2	Temperature limit 2	Input	LC2030/Par2

TASK 4.2.1

Create a dashboard with following functions:

- Display:
 - Status as text
 - Pump status (M1, M2, M3, M4) with LED (yellow)
 - Heating rod status (Q1) with LED (pink)
 - Level switch status (B1, B2, B3) with LED (green)
- Button:
 - Start production
 - Start empty tank
 - Confirmation
 - Cancelation
- Input field
 - Temperature limit (Par1, Par2)

SOLUTION



A dashboard could look like displayed.

Following elements were used

- Button
- Text Input
- Text Log
- LED Indicator

Text Log

o2-de 45% 11:10

← Edit panel

Panel name *
Status

Topic *
LC2030/TM1

Additional options >

QoS
1

☐ Enable notification

☒ Payload is JSON Data

JsonPath for subscribe *
\$.value

CANCEL SAVE

LED Indicator

o2-de 45% 11:10

← Edit panel

Panel name *
Q1

Topic *
LC2030/Q1

Payload on *
true

Payload off *
false

LED color
#ff00b2

LED color
#9E9E9E

LED on icon

LED off icon

☐ Enable notification

☒ Payload is JSON Data

JsonPath for subscribe *
\$.value

Text Input

o2-de 45% 11:11

← Edit panel

Panel name *
Temperature 1

Topic *
LC2030/Par1

☐ Show sent timestamp

☐ Confirm before publish

☒ Payload is JSON Data

JSON pattern for publish *
{"value": <text-payload>}

☐ Clear text on publish

Retain QoS
1

CANCEL SAVE

Button

o2-de 45% 11:11

← Edit panel

Panel name *
Prod.

Topic *
LC2030/BM1

Payload *
true

☐ No payload

☒ Use icons for button

Icon color
#004bff

☒ Choose icon

☐ Hide label

☒ Payload is JSON Data

JSON pattern for publish *
{"value": <button-payload>}

Information regarding errors, inaccuracies and expansion options are highly appreciated!

Ingenieurbüro Dr.-Ing. Schoop GmbH
Riechelmannweg 4
D-21109 Hamburg
Tel.: 040 / 754 922 30
www.schoop.de
Email: info@schoop.de