

# GATEWAY SS10680 CONFIGURATION GUIDE

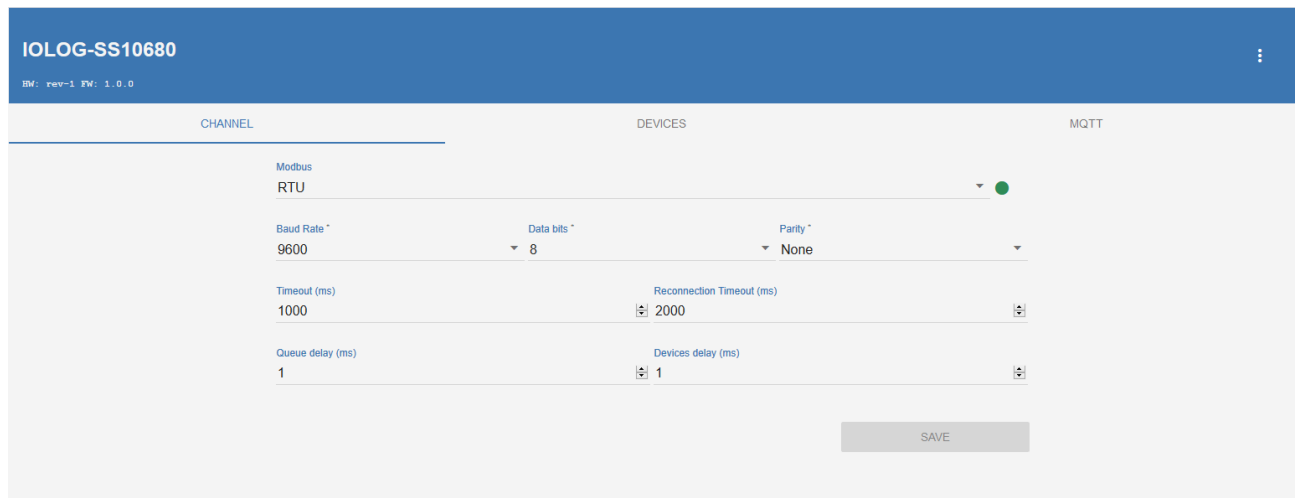
## Prerequisites

- 1) Have an account on Bluemix
- 2) Have created an Internet of Things Platform on Bluemix
- 3) Have created a device on Internet of Things Platform, and have its Device ID and its Authentication Token

## Procedure

- 1) Connect the gateway SS10680 to the power supply and to Internet.
- 2) Open the browser, and digit in index bar 192.168.1.100
- 3) Enter username and password. The default ones are:
  - Username: admin
  - Password: password
- 4) Click on 'CHANNEL' tab, and fill the requested fields for Modbus channel. For this example, we will use the following parameters:
  - Modbus: select 'RTU'
  - Baud Rate: select '9600'
  - Data bits: select '8'
  - Parity: select 'None'
  - Timeout (ms): enter '1000'
  - Reconnection Timeout (ms): enter '2000'
  - Queue Delay (ms): enter '1'
  - Devices Delay (ms): enter '1'

Click on 'SAVE' button. If everything went well, you get the following result



The screenshot shows the configuration interface for the IOLOG-SS10680 gateway. The title bar is blue and contains the text 'IOLOG-SS10680' and 'BM: rev-1 FW: 1.0.0'. Below the title bar, there are three tabs: 'CHANNEL', 'DEVICES', and 'MQTT'. The 'CHANNEL' tab is selected and highlighted. The configuration form is divided into two columns. The left column contains: 'Modbus' (RTU), 'Baud Rate \*' (9600), 'Timeout (ms)' (1000), and 'Queue delay (ms)' (1). The right column contains: 'Data bits \*' (8), 'Parity \*' (None), 'Reconnection Timeout (ms)' (2000), and 'Devices delay (ms)' (1). A green status indicator is visible next to the Modbus setting. At the bottom right of the form, there is a 'SAVE' button.

- 5) Click on 'DEVICES' tab. For this example, we will use the following two devices:
- IOLOG SS10014, configured as in the following picture

The screenshot displays the configuration page for device IOLOG SS10014. At the top, the device name and address (1) are shown. Below this, four input channels are listed, each with a read-only (RO) status and a Pt100 sensor type. The channels are IN 0 (3:14), IN 1 (3:15), IN 2 (3:16), and IN 3 (3:17). Each channel has a Topic (PUB) set to 'iot-2/type/SS10680/id/gateway/evt/input1/fmt/json', a QoS of 0, and a Retain checkbox that is unchecked. The Publish event is set to 'On value change'. The Threshold (inclusive) is 0.1 for IN 0 and IN 1, and 0 for IN 2 and IN 3. At the bottom of the configuration area, there are 'REMOVE' and 'SAVE' buttons.

Channel	Address	Topic (PUB)	QoS	Retain	Publish	Threshold (inclusive)
IN 0	3:14	iot-2/type/SS10680/id/gateway/evt/input1/fmt/json	0	<input type="checkbox"/>	On value change	0.1
IN 1	3:15	iot-2/type/SS10680/id/gateway/evt/input1/fmt/json	0	<input type="checkbox"/>	On value change	0.1
IN 2	3:16	iot-2/type/SS10680/id/gateway/evt/input1/fmt/json	0	<input type="checkbox"/>	On value change	0
IN 3	3:17	iot-2/type/SS10680/id/gateway/evt/input1/fmt/json	0	<input type="checkbox"/>	On value change	0

At the end, click on 'SAVE' button.

- IOLOG SS10130, configuring the inputs as below

IOLOG SS10130 ×

---

Address\* 2 Name\* IOLOG SS10130

---

RO **DI 0**  
1:0

Topic (PUB) QoS  
iot-2/type/SS10680/id/gateway/evt/input1/fmt/json 0  Retain

Publish Threshold (inclusive)  
On value change 0.1

---

RO **DI 1**  
1:1

Topic (PUB) QoS  
iot-2/type/SS10680/id/gateway/evt/fan1/fmt/json 0  Retain

Publish Threshold (inclusive)  
On value change 0.1

---

RO **DI 2**  
1:2

Topic (PUB) QoS  
iot-2/type/SS10680/id/gateway/evt/fan2/fmt/json 0  Retain

Publish Threshold (inclusive)  
On value change 0.1

---

RO **DI 3**  
1:3

Topic (PUB) QoS  
iot-2/type/SS10680/id/gateway/evt/bulb1/fmt/json 0  Retain

Publish Threshold (inclusive)  
On value change 0.1

---

...

REMOVE SAVE

Now, scroll down and configure the outputs as below

The screenshot shows the IOLOG SS10130 configuration interface. It displays four digital outputs (DO 0 to DO 3) with their respective publish and subscribe topics and settings. Each output is configured with a publish topic and a subscribe topic, both using the syntax `iot-2/type/SS10680/id/gateway/evt/alarm_*/fmt/json` for publish and `iot-2/type/SS10680/id/gateway/cmd/activateAlarm_*/fmt/json` for subscribe. The publish topics are set to "On value change" with a threshold of 0. The subscribe topics are set to "On value change" with a threshold of 0. The interface includes a "REMOVE" button and a "SAVE" button.

Output	Topic (PUB)	CoS	Topic (SUB)	CoS
DO 0 (1:16)	iot-2/type/SS10680/id/gateway/evt/alarm_input1/fmt/json	0	iot-2/type/SS10680/id/gateway/cmd/activateAlarm_input1/fmt/json	0
DO 1 (1:17)	iot-2/type/SS10680/id/gateway/evt/alarm_fan1/fmt/json	0	iot-2/type/SS10680/id/gateway/cmd/activateAlarm_fan1/fmt/json	0
DO 2 (1:18)	iot-2/type/SS10680/id/gateway/evt/alarm_fan2/fmt/json	0	iot-2/type/SS10680/id/gateway/cmd/activateAlarm_fan2/fmt/json	0
DO 3 (1:19)	iot-2/type/SS10680/id/gateway/evt/alarm_bulb1/fmt/json	0	iot-2/type/SS10680/id/gateway/cmd/activateAlarm_bulb1/fmt/json	0

When you have finished, click on 'SAVE' button.

N.B. The topics used in this example have a precise form, due at the Watson IoT Platform that require a special syntax.

For publish a topic to the Watson IoT Platform, the following syntax is used

`iot-2/type/type_id/id/device_id/evt/event_id/fmt/format_string`

where

- *type\_id*, is the ID of the type of device added on the Watson IoT Platform
- *device\_id*, is the ID of the device added on the Watson IoT Platform
- *event\_id*, is the event's ID. The ID can be any valid string for MQTT.
- *format\_string*, is a string that define the content type of the message payload. For this example, we will use *json* format.

For subscribe to a topic from Watson IoT Platform, the following syntax is used

`iot-2/type/type_id/id/device_id/cmd/command_id/fmt/format_string`

where

- *type\_id*, is the ID of the type of device added on the Watson IoT Platform
- *device\_id*, is the ID of the device added on the Watson IoT Platform
- *command\_id*, is the event's ID. The ID can be any valid string for MQTT.
- *format\_string*, is a string that define the content type of the message payload. For this example, we will use *json* format.

6) Click on 'MQTT' tab, and fill the form. For this example, we will use the following parameters:

- Broker Address: address used by Watson IoT Platform for send/receive topics. The address must be as below

`orgId.messaging.internetofthings.ibmcloud.com`

where *orgId* is the unique organization ID that was generated when you registered the service instance

- Broker Port: enter '1883'
- ClientId: must be in the following format '**g:orgId:DeviceType:DeviceId**', where:
  - *g*, due the fact that we defined the device as *gateway* in Watson IoT Platform
  - *orgId*, is the unique organization ID that was generated when you registered the service instance
  - *DeviceType*, is the ID of the type of device added on the Watson IoT Platform
  - *DeviceId*, is the ID of the device added on the Watson IoT Platform
- Keep Alive (s): enter '60'
- Clean Session: check the checkbox
- Username: enter the string '*use-token-auth*'
- Password: enter the Authentication Token that you saved when you created the device on Watson IoT Platform.

On 'Payload Structure' section, select the checkboxes relative to 'device' and 'datetime'.

At the end, click on 'SAVE' button. If everything went well, you get the following result

**IOLOG-SS10680**  
RM: rev-1 FM: 1.0.0

CHANNEL DEVICES **MQTT**

Broker Address \*  
2ruxma.messaging.internetofthings.ibmcloud.com

Broker Port \*  
1883

ClientId \*  
g.2ruxma.SS10680.gateway

Keep Alive (s) \*  
60

Clean session

Username  
use-token-auth

Password  
.....

Enable TLS/SSL

**Payload structure**

value - Sampled value  
 valid - Sampled value validity  
 device - Device name  
 timestamp - Unix time  
 datetime - Date in ISO-8601 format  
 communicationKO - Communication KO

Non-valid value  
null

string null

```
{
  "value": 68.35,
  "datetime": "2017-12-20T09:30:56.965Z",
  "device": "MyDevice"
}
```

**SAVE**