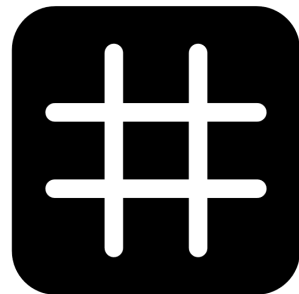


MQTT dingz



dingz

MQTT topics

V1.01 - May 2022



Content

mDNS Discovery	3
MQTT Topics	4
Lights	6
Front LED	7
Motors (Blind/Roller/Shutter/Shade/Window/Door)	8
Input	10
Button	11
MQTT Command tree	13
Availability & Announces	13
Exemples	14

mDNS Discovery

dingz devices announce a HTTP service on port 80 via mDNS. Hostname is always in the form of dingz-<model>-<mac>.

MQTT Config

- Set broker IP and port
- Set security and certificate
- If authentication is needed, define user and password
- Enable MQTT in the settings

cURL:

To configure MQTT client use:

```
curl -i -d '{"mqtt":{"uri":"mqtt://user:password@192.168.99.115","enable":true}}'  
http://192.168.99.111/api/v1/services\_config
```

dingz config file: (services)

```
},  
  "mqtt": {  
    "uri": "mqtt://192.168.99.115",  
    "enable": true,  
    "server.crt": null  
  }  
}
```

If needed do also define port and path.

Can also be used in mqttts mode

MQTT Topics

MQTT Topic	Description	Retained
Light:		
dingz/<ID>/<model>/state/light/<n>	dingz publishes JSON if a parameter changes	true
dingz/<ID>/<model>/power/light/<n>	dingz publishes power if it changes > 3W (max every 10s)	true
dingz/<ID>/<model>/energy/light/<n>	dingz publishes energy if it changes (max every 10min)	true
dingz/<ID>/<model>/command/light/<n>	dingz subscribes to command (used to set values on dingz)	false
Motor:		
dingz/<ID>/<model>/state/motor/<n>	dingz publishes JSON if a parameter changes	true
dingz/<ID>/<model>/power/motor/<n>	dingz publishes power if it changes $\geq 3W$ (max every 10s)	true
dingz/<ID>/<model>/energy/motor/<n>	dingz publishes energy if it changes (max every 10min)	true
dingz/<ID>/<model>/command/motor/<n>	dingz subscribes to command (used to set values on dingz)	false
Input:		
dingz/<ID>/<model>/state/input/<n>	dingz publishes the current input state as 0 or 1	true
Button:		
dingz/<ID>/<model>/event/button/<n>	dingz publishes button events: p=press, r=release, h=hold (repeats every 1s), m<cnt>=multiple clicks (e.g., double click = m2) <ul style="list-style-type: none"> dingz button: channel = 0...3 external button on input: channel = 4 	false
Motion Detector (PIR):		
dingz/<ID>/<model>/event/pir/<n>	dingz publishes pir event: s=somebody, ss=still somebody (publish every 3s while a person is present) n=nobody	false
Sensors:		
dingz/<ID>/<model>/sensor/temperature	dingz publishes temperature if it changes $\geq 0,5^{\circ}C$	true
dingz/<ID>/<model>/sensor/humidity	dingz publishes humidity if it changes $\geq 5\%$	true
dingz/<ID>/<model>/sensor/co2	dingz publishes CO2 level if it changes $\geq 100ppm$	true
Function:		
dingz/<ID>/<model>/state/function/thermostat	dingz publishes thermostat state as 0 or 1	true
dingz/<ID>/<model>/state/function/nightlight	dingz publishes nightlight state as 0 or 1	true
dingz/<ID>/<model>/state/function/pir	dingz publishes pir state as 0 or 1 <ul style="list-style-type: none"> 1 if person detected 0 if timer (manual or light) elapsed 	true
dingz/<ID>/<model>/command/function/...	dingz subscribes to command (used to control functions on dingz)	
Settings:		
dingz/<ID>/<model>/settings/dingz	dingz publishes JSON with general settings	true
dingz/<ID>/<model>/settings/light/<n>	dingz publishes JSON with settings of the light	true

dingz/<ID>/<model>/settings/motor/<n>	dingz publishes JSON with settings of the motor	true
dingz/<ID>/<model>/settings/input/<n>	dingz publishes JSON with settings of the input	true
dingz/<ID>/<model>/settings/button/<n>	dingz publishes JSON with settings of the button	true
dingz/<ID>/<model>/settings/pir/<n>	dingz publishes JSON with settings of the pir	true
dingz/<ID>/<model>/settings/function/pir/<n>	dingz publishes JSON with settings of the pir function	true
dingz/<ID>/<model>/settings/function/nightlight/<n>	dingz publishes JSON with settings of the nightlight function	true
dingz/<ID>/<model>/settings/function/thermostat/<n>	dingz publishes JSON with settings of the thermostat functions	true
dingz/<ID>/<model>/command/settings...	dingz subscribes to command (used to set values on dingz)	false

Lights

dingz publishes the state of the lights

If any value of the a light changes, because of an internal (button, timer,...) or external (REST, MQTT, Cloud,...) event, the new values are immediately published to the broker.

dingz/<ID>/<model>/state/light/<n>

- <n> is the light number [0...3]
- Payload JSON
 - {"brightness": 100, "fadetime": 20, "turn": "on", "exception": 0 }
 - brightness = 0...100% [0...100 in 1%]
 - fadetime = 0.0...10.0s [0...100 in 1/10s]
 - turn = "on" or "off" [string]
 - exception = 0 (ok), 1 (overload),

dingz subscribes to the light command topic

If a new value is published to the command topic, it is immediately overwritten on the dingz and the lights adapt to the new values. This can be used to control the lights of dingz by any other MQTT client.

dingz/<ID>/<model>/command/light/<n>

- <n> is the light number [0...3]
- Payload JSON
 - {"brightness": 100, "fadetime": 20, "turn": "on" }
 - brightness = 0...100% [0...100 in 1%]
 - fadetime = 0.0...10.0s [0...100 in 1/10s]
 - turn = "on" or "off" or "toggle" [string]

Front LED

dingz/<ID>/<model>/state/led

dingz/<ID>/<model>/command/led

it takes arguments: on=<0,1>, r=<0..255>, g=<0..255>, b=<0..255>

The event from led switching also should be send with 1 second delay from last change.

Motors

(Blind/Roller/Shutter/Shade/Window/Door)

dingz publishes the state of the motors

If any value of the a motor changes, because of an internal (button, timer,...) or external (REST, MQTT, Cloud,...) event, the new values are immediately published to the broker. If the motor is in motion the values are updated (published) every 2s.

dingz/<ID>/<model>/state/motor/<n>/

- Payload JSON
 - {"position": 100, "goal": 50, "lamella": 30, "motion": 2}
 - position = 0...100% [0...100 in 1%]] the current position of the motor
 - goal = 0...100% [0...100 in 1%]] the goal position the motor is running to
 - lamella = 0...100% [0...100 in 1%]] the current position of the blind lamellas
 - motion =
 - 0 = stopped: the motor is not in motion
 - 1 = opening: the motor is running towards 100%
 - 2 = closing: the motor is running towards 0%
 - 3 = calibrating: the motor is executing the calibration process

dingz subscribes to the motor command topic

If a new value is published to the command topic, it is immediately overwritten on the dingz and the motors adapt to the new values. This can be used to control the motors of dingz by any other MQTT client.

*The motor can be controlled by sending a "motion" or "position" command.
(0% is fully closed, 100% is fully open)*

dingz/<ID>/<model>/command/motor/<n>

- <n> is the motor number [0 or 1]
- Payload JSON to send the motor to a specific **position**
 - {"position": 100, "lamella": 30}
 - position = 0...100% [0...100 in 1%]] the current position of the motor
 - lamella = 0...100% [0...100 in 1%]] the position of the blind lamellas
- Payload JSON to move the motor to a specific **direction**
 - {"motion": 1}
 - motion =
 - 0 = stop: the motor stops at its current position
 - 1 = open: the motor will run towards 100%
 - 2 = close: the motor will run towards 0%
 - 3 = do calibration: the motor will execute the calibration process

Input

Dingz publishes the state of the inputs

If the state of an input changes, the new values are immediately published to the broker.

The input only publishes on that topic, if the input type is defined as "state". If the input type is "PIR" or "Button" then the changes are published to the topics: PIR or Button.

dingz/<ID>/<model>/state/input/<n>

- <n> is the input number [0] (current hardware only supports one input, so it is always 0)
- Payload
 - 0: no voltage applied to the input
 - 1: voltage is applied to the input

In preparation for Q2 2022

The following Input types must be added to the dingz fw and webUi: pulse button, toggle button, pir, state

Button

Dingz publishes the button events

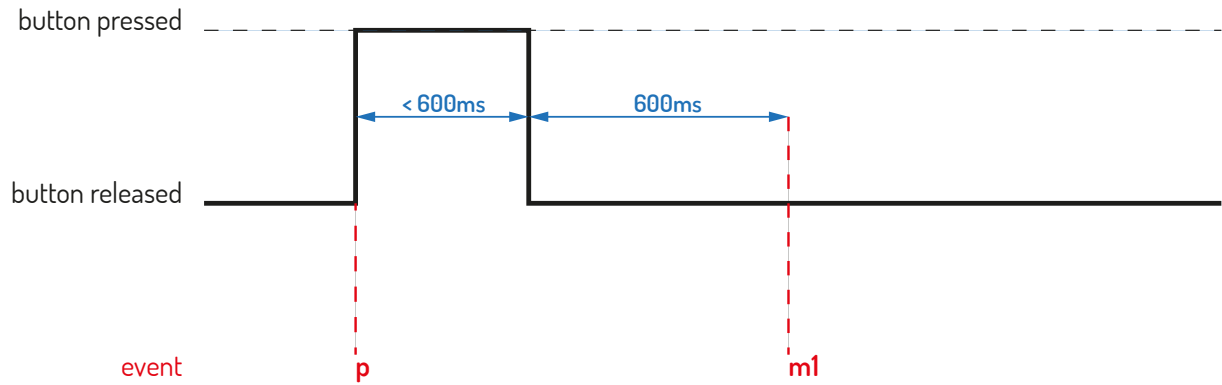
If a button event occurs, it is immediately published to the broker.

Events from an external button connected to the dingz input are also published as button (4) events.

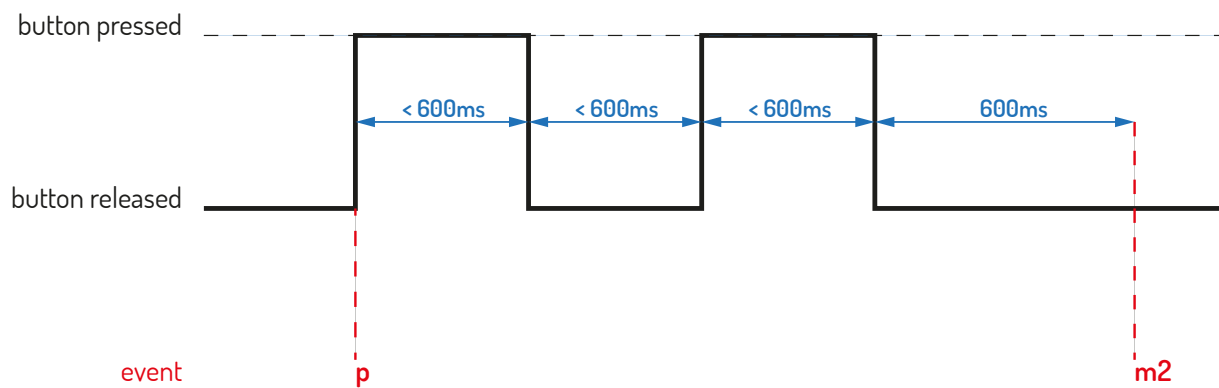
dingz/<ID>/<model>/event/button/<n>

- <n> is the button number [0...3] internal dingz buttons and [4] external button connected to the dingz input.
- Payload:
 - p=press (immediately if button pressed)
 - h=hold (if button is pressed for more than 600ms) (repeats every 1s) (not available for a toggle button connected to the input)
 - r=release (only if button is released after hold)
 - m<cnt>=multiple clicks (e.g., double click = m2) (if button is clicked multiple times and all never held for more than 600ms.

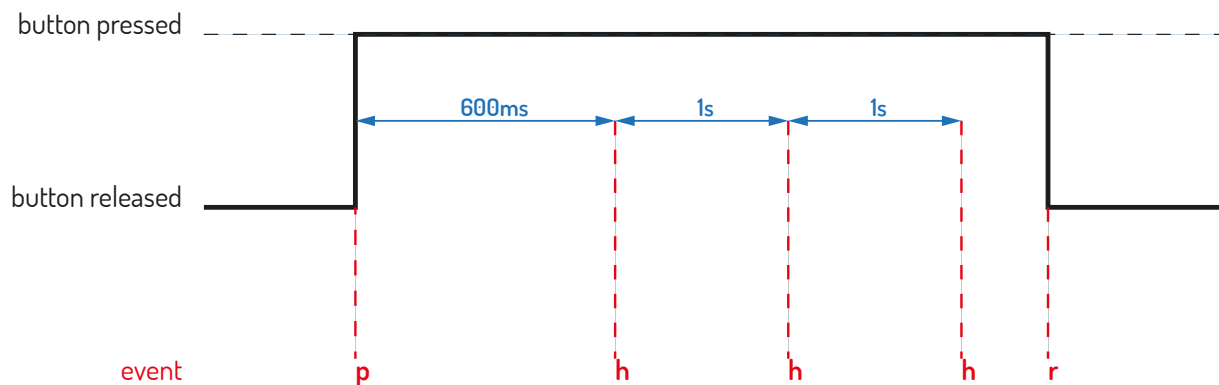
Click Event



Double Click Event



Long Press Event



MQTT Command tree

dingz/<ID>/<model>/command

- Address an individual dingz

dingz/command

- Address all dingz on the broker

dingz/groupe/<groupe ID>/command

- Address all dingz in one groupe on the broker

Availability & Announces

dingz/<ID>/announce

The message is JSON-formatted and contains a list of attributes:

- model (mac)
- IP
- fw_version (contains the current firmware version)
- hw_version.

dingz/<ID>/online

- True if dingz is available
mqtt_update_period under /settings (default = 30s). 0 = automatic update disabled

Last Will: online=false

Exemples

Subscribe to MQTT on local host example:

```
mosquitto_sub -v -t '#'
```

received events examples:

```
dingz/B4E62DF19989/announce {"model":"DZ1F-PIR","ip":"192.168.99.111","hw_version":"1.0","fw_version":"1.3.9"}
```

```
dingz/B4E62DF19989/DZ1F-PIR/state/light/1 {"turn":"off","brightness":0}
```

```
dingz/B4E62DF19989/DZ1F-PIR/state/light/0 {"turn":"off","brightness":0}
```

```
dingz/B4E62DF19989/DZ1F-PIR/state/motor/1 {"state":"stop","position":50,"lamella":100}
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 97.00
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/temperature 27.21
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 250.00
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 210.00
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 164.00
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
```

```
dingz/B4E62DF19989/DZ1F-PIR/state/input/0 1
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/1 s
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
```

```
dingz/B4E62DF19989/DZ1F-PIR/state/input/0 0
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 154.00
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/temperature 28.15
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/1 n
```

```
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 147.00
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
```

```
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 p
dingz/B4E62DF19989/DZ1F-PIR/state/input/0 1
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 t
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 123.00
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 h
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 h
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 h
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 h
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 h
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 r
dingz/B4E62DF19989/DZ1F-PIR/state/input/0 0
dingz/B4E62DF19989/DZ1F-PIR/event/button/4 t
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
dingz/B4E62DF19989/DZ1F-PIR/sensor/light 122.00
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 s
dingz/B4E62DF19989/DZ1F-PIR/event/pir/0 n
```

Send command to dingz examples:

```
sj:-$ mosquitto_pub -t 'dingz/B4E62DF19989/DZ1F-PIR/command/light/1' -m '{"turn":"on","brightness":10}'
sj:-$ mosquitto_pub -t 'dingz/B4E62DF19989/DZ1F-PIR/command/light/1' -m '{"turn":"toggle"}'
sj:-$ mosquitto_pub -t 'dingz/B4E62DF19989/DZ1F-PIR/command/motor/0' -m '{"state":"open"}'
sj:-$ mosquitto_pub -t 'dingz/B4E62DF19989/DZ1F-PIR/command/motor/0' -m '{"state":"stop"}'
sj:-$ mosquitto_pub -t 'dingz/B4E62DF19989/DZ1F-PIR/command/motor/0' -m
'{"position":50,"lamella":100}'
```



dingz

dingz by iolo AG - Pra Pury 7d - 3280 Morat

hello@iolo.ch - 026 674 60 00