MQTT dingz

dingz

MQTT topics

FW 2.1.58

V1.02 - Feb 2024

Content

mDNS Discovery

dingz devices announce a HTTP local TXT service on port 80 over mDNS. Hostname is always in the form of DINGZ-<model>-<MAC>.

Eg.: DINGZ-dz1f-4b-30C6F72355C4.local TXT

MQTT Config

- Set broker IP and port
- If authentication is needed, define user and password
- Enable MQTT in the settings over dingz API request
- Optionally, set security and certificate as an PEM-encoded string (beta)

To configure and enable the MQTT client on dingz you can use curl as follows:

- curl -i -d '{"mqtt":{"uri":"mqtt://user:password@192.168.0.123"","enable":true}}'

  http://192.168.99.111/api/v1/services_config

where 192.168.0.123 in the payload must be replaced by the address of your MQTT broker and 192.168.0.100 in the endpoint URL must be replaced by the address of your dingz

You find the settings in the JSON config file you can dump from the dingz web ui, in section “services”

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

If your MQTT broker listens to a port which is not the default 1883, you need to also define port, like:

- mqtt://192.168.99.115:<port_number>

MQTT Topics

The MQTT client on dingz publishes topics formatted tipically like this:

- dingz/<id>/<model>/[state | power | energy | command | event]/[light | motor | ...] /<n>

where the first part containing dingz/<id>/<model> is common between all topics, with exception of announces, and are defined as follows:

- <id> is the dingz ID value you can find in the webui in the System section. By default, dingz ID is set to the device's mac address in lowercase
- <model> is the dingz model name in lowercase
- <n> is an index always starting from 0 and goes up to 3 or 4 depending what is indexed

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	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	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 (beta)		
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) • dingz button: channel = 0...3 • 1 for external PIR	false

MQTT Topic	Description	Retained
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. The index n assumes: • always 0 for internal PIR • always 1 for external PIR note that dingz without PIR can have an external PIR and its events will be published with n=1	false
Sensors:		
dingz/<id>/<model>/sensor/temperature	dingz publishes temperature if it changes ≥ 0.5°C	true
dingz/<id>/<model>/sensor/light	dingz publishes changes inlux readings	true
Function:		
dingz/<id>/<model>/state/thermostat	Dingz publishes state changes as JSON eg.: {"status":"heating","mode":"heating","target":22.0} the “mode” can be “heating” or “cooling”	true
dingz/<id>/<model>/command/function/...	dingz subscribes to command (used to control functions on dingz)	
Availability and announces:		
dingz/<id>/online	dingz publishes online state	true
dingz/<id>/last_alive	dingz publishes last_alive as a timestamp	true
dingz/<id>/announce	dingz publishes its model, ip, hw_ and fw_version in JSON	true

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]
- JSON payload, eg.: {"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]
- JSON payload, eg.: {"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 publishes the state of the led

if the led switches on or off, or color changes dingz plublishes he new on/off status.

dingz/<id>/<model>/state/led

dingz subscribes to the led command topic

dingz/<id>/<model>/command/led

- Payload: {"on":0,"r":255,"g":255,"b":255}
    - on=<0,1>, r=<0..255>, g=<0..255>, b=<0..255>

The event from led switching should be sent 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>

- JSON payload, eg.: {"motion":2", 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 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]
- JSON payload 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
- JSON payload 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 execut the calibration process

Input (beta)

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

Button

Dingz publishes the button events

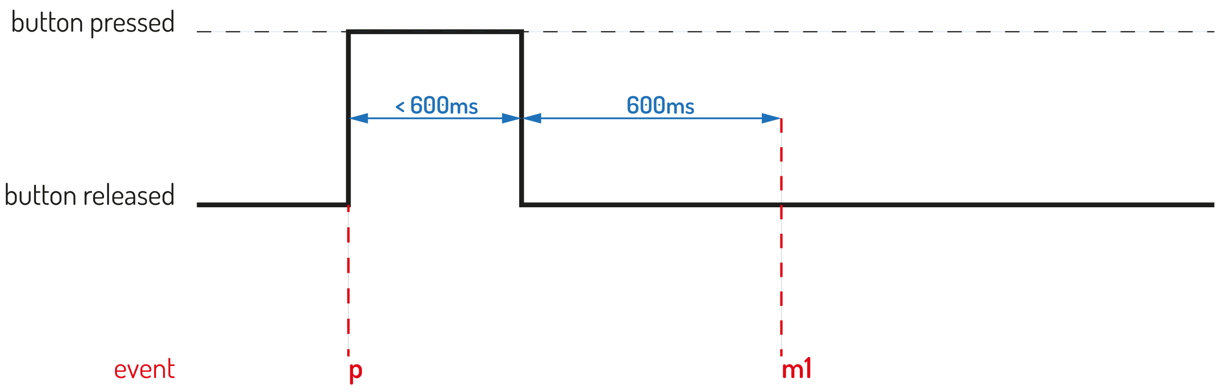
If a button event occurs, it is immediately published to the broker.

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

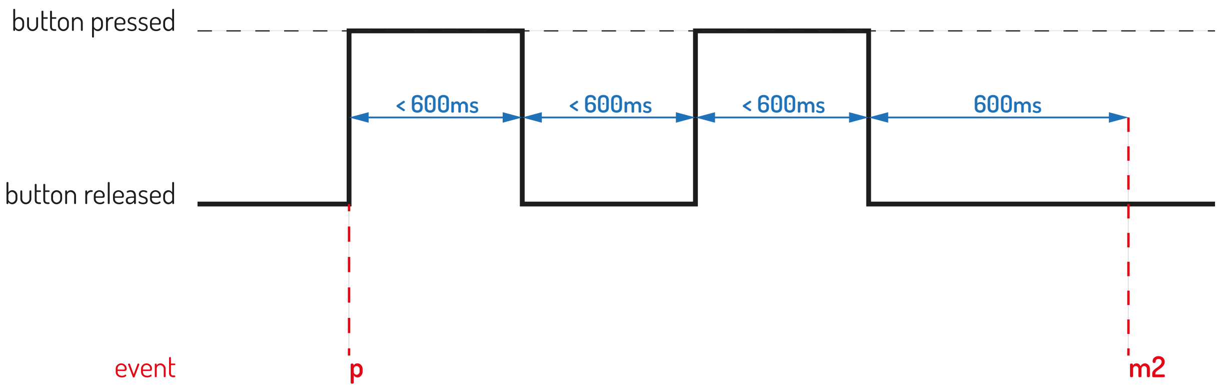
dingz/<id>/<model>/event/button/<n>

- <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.

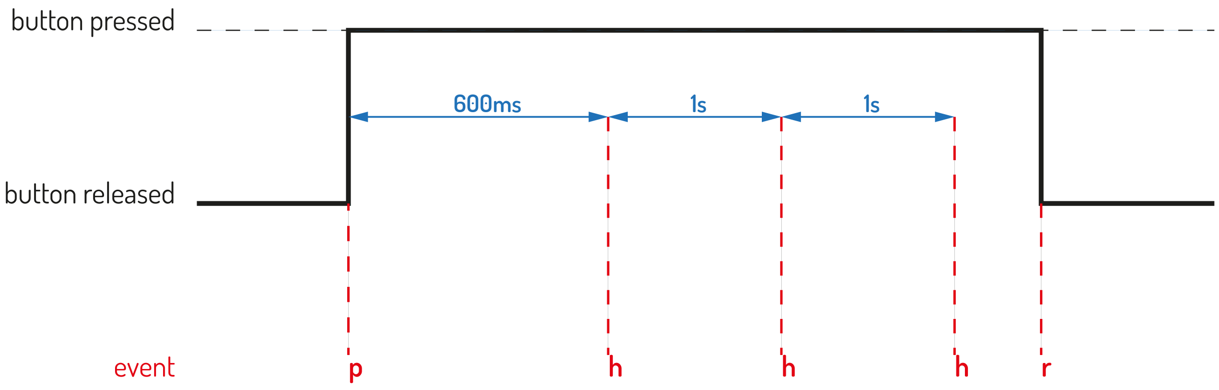
Click Event



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

Examples

Subscribe to MQTT on local host

mosquitto_sub -v -t '#'

received events examples:

```
dingz/mytestdingzid/announce {"model":"dz1f-pir","ip":"192.168.99.111","hw_version":"1.0","fw_version":"1.3.9"}

dingz/mytestdingzid/dz1f-pir/state/light/1 {"turn":"off","brightness":0}
dingz/mytestdingzid/dz1f-pir/state/light/0 {"turn":"off","brightness":0}
dingz/mytestdingzid/dz1f-pir/state/motor/1
{"state":"stop","position":50,"lamella":100}
dingz/mytestdingzid/dz1f-pir/event/pir/0 n
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/sensor/light 97.00
dingz/mytestdingzid/dz1f-pir/event/pir/0 n
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/sensor/temperature 27.21
dingz/mytestdingzid/dz1f-pir/sensor/light 250.00
dingz/mytestdingzid/dz1f-pir/sensor/light 210.00
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/event/pir/0 n
dingz/mytestdingzid/dz1f-pir/sensor/light 164.00
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/state/input/0 1
dingz/mytestdingzid/dz1f-pir/event/pir/1 s
dingz/mytestdingzid/dz1f-pir/event/pir/0 n
dingz/mytestdingzid/dz1f-pir/state/input/0 0
dingz/mytestdingzid/dz1f-pir/sensor/light 154.00
dingz/mytestdingzid/dz1f-pir/sensor/temperature 28.15
dingz/mytestdingzid/dz1f-pir/event/pir/1 n
dingz/mytestdingzid/dz1f-pir/sensor/light 147.00
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/event/pir/0 n
dingz/mytestdingzid/dz1f-pir/event/button/4 p
dingz/mytestdingzid/dz1f-pir/state/input/0 1
dingz/mytestdingzid/dz1f-pir/event/button/4 t
dingz/mytestdingzid/dz1f-pir/sensor/light 123.00
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/event/button/4 h
dingz/mytestdingzid/dz1f-pir/event/button/4 h
dingz/mytestdingzid/dz1f-pir/event/button/4 h
dingz/mytestdingzid/dz1f-pir/event/button/4 h
dingz/mytestdingzid/dz1f-pir/event/button/4 h
dingz/mytestdingzid/dz1f-pir/event/button/4 r
```

```
dingz/mytestdingzid/dz1f-pir/state/input/0 0
dingz/mytestdingzid/dz1f-pir/event/button/4 t
dingz/mytestdingzid/dz1f-pir/event/pir/0 n
dingz/mytestdingzid/dz1f-pir/sensor/light 122.00
dingz/mytestdingzid/dz1f-pir/event/pir/0 s
dingz/mytestdingzid/dz1f-pir/event/pir/0 n

dingz/mytestdingzid/dz1f-pir/state/led {"on":1,"r":255,"g":255,"b":255}

dingz/mytestdingzid/dz1f-pir/state/light/0 {"turn":"on","brightness":100,"exception":0}
```

Send command to dingz

```
sj:~$ mosquitto_pub -t 'dingz/mytestdingzid/dz1f-pir/command/light/1' -m
'{"turn":"on","brightness":10}'

sj:~$ mosquitto_pub -t 'dingz/mytestdingzid/dz1f-pir/command/light/1' -m
'{"turn":"toggle"}'

sj:~$ mosquitto_pub -t 'dingz/mytestdingzid/dz1f-pir/command/motor/0' -m
'{"state":"open"}'

sj:~$ mosquitto_pub -t 'dingz/mytestdingzid/dz1f-pir/command/motor/0' -m
'{"state":"stop"}'

sj:~$ mosquitto_pub -t 'dingz/mytestdingzid/dz1f-pir/command/motor/0' -m
'{"position":50,"lamella":100}'

sj:~$ mosquitto_pub -t 'dingz/mytestdingzid/dz1f-pir/command/led' -m '{"on":1}'
```

dingz

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

hello@iolo.ch - 026 674 60 00