

KC868-D16 dimmer controller http MQTT TCP command

A.HTTP command

Postpwd is http command password, set by webpage in browser.

Brightness range (0-100) 0:OFF 100:MAX brightness

1. Read dimmer' state

`http://192.168.1.200/dimmer_ctl.cgi?DimmerFF=RD&postpwd=12345`

if successful will feedback:

```
DimmerCallback({"Dimmer1":"10","Dimmer2":"2","Dimmer3":"8","Dimmer4":"68","Dimmer5":"48","Dimmer6":"48","Dimmer7":"51","Dimmer8":"50","Dimmer9":"0","Dimmer10":"0","Dimmer11":"0","Dimmer12":"0","Dimmer13":"0","Dimmer14":"0","Dimmer15":"0","Dimmer16":"0","post_pwd":"12345"});
```

if failed will feedback: Error parameter or password!

2. Change every channel output brightness:

```
http://192.168.1.200/dimmer_ctl.cgi?Dimmer01=10&postpwd=12345 // dimmer output1=10% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer02=20&postpwd=12345 // dimmer output2=20% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer03=30&postpwd=12345 // dimmer output3=30% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer04=40&postpwd=12345 // dimmer output4=40% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer05=50&postpwd=12345 // dimmer output5=50% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer06=60&postpwd=12345 // dimmer output6=60% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer07=70&postpwd=12345 // dimmer output7=70% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer08=80&postpwd=12345 // dimmer output8=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer09=80&postpwd=12345 // dimmer output9=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer10=80&postpwd=12345 // dimmer output10=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer11=80&postpwd=12345 // dimmer output11=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer12=80&postpwd=12345 // dimmer output12=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer13=80&postpwd=12345 // dimmer output13=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer14=80&postpwd=12345 // dimmer output14=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer15=80&postpwd=12345 // dimmer output15=80% brightness
http://192.168.1.200/dimmer_ctl.cgi?Dimmer16=100&postpwd=12345 // dimmer output16=100% brightness
```

if successful will feedback:

```
DimmerCallback({"Dimmer1":"10","Dimmer2":"2","Dimmer3":"8","Dimmer4":"68","Dimmer5":"48","Dimmer6":"48","Dimmer7":"51","Dimmer8":"50","Dimmer9":"0","Dimmer10":"0","Dimmer11":"0","Dimmer12":"0","Dimmer13":"0","Dimmer14":"0","Dimmer15":"0","Dimmer16":"0","post_pwd":"12345"});
```

if failed will feedback: Error parameter or password!

B.MQTT command

command_topic: "dimmer/xxxxxxxxxxxxxxxxxxxxxxx/set" read state topic:

state_topic: "dimmer/xxxxxxxxxxxxxxxxxxxxxxx/state"

Note: xxxxxxxxxxxxxxxxxxxxxxxxxx is your KC868-D8's ID, it's 24bit. You can find in controller's config webpage.

Example:

set channel1 output is 10% brightness:

```
Payload= {"dimmer1":{"value":10}}
```

if successful will feedback:

```
{"dimmer1":{"value":10},"dimmer2":{"value":0},"dimmer3":{"value":0},"dimmer4":{"value":0},"dimmer5":{"value":0},"dimmer6":{"value":0},"dimmer7":{"value":0},"dimmer8":{"value":0}}
```

set channel2 output is 20% brightness:

```
Payload= {"dimmer2":{"value":20}}
```

if successful will feedback:

```
{"dimmer1":{"value":10},"dimmer2":{"value":20},"dimmer3":{"value":0},"dimmer4":{"value":0},"dimmer5":{"value":0},"dimmer6":{"value":0},"dimmer7":{"value":0},"dimmer8":{"value":0}}
```

Read dimmer's output state:

```
Payload= {"dimmer":{"read":all}}
```

if successful will feedback 8 channel output state:

```
{"dimmer1":{"value":10},"dimmer2":{"value":20},"dimmer3":{"value":0},"dimmer4":{"value":0},"dimmer5":{"value":0},"dimmer6":{"value":0},"dimmer7":{"value":0},"dimmer8":{"value":0}}
```

Integrate with home assistant configuration.yaml sample:

light:

```
- platform: mqtt
  name: dimmer1
  unique_id: dimmer1
  schema: template
  command_topic: "dimmer/xxxxxxxxxxxxxxxxxxxxx/set"
  state_topic: "dimmer/xxxxxxxxxxxxxxxxxxxxx/state"
  command_on_template: >
    {"dimmer1":{"value":{{(brightness / 255 * 100) | int - 1}}}
  command_off_template: '{"dimmer1":{"value": 0}}'
  state_template: >
    {%- if value_json.dimmer1.value == 0 -%}
      off
    {%- else -%}
      on
    {%- endif -%}
  brightness_template: '{{(value_json.dimmer1.value / 99 * 255)| int}}'
```

C.TCP command

Note: use TCP command you should set dimmer “work mode”=“TCP Server” on network setting webpage.

if you forgot the IP address of your dimmer controller. You can hold on the reset button beside RJ45 port for about 3 seconds, and the KC868-D8 controller will automatically return to the following work mode:

IP:192.168.1.200

Port: 4196

Work Mode: TCP Server

1. Read single channel output state:

Send: DIMMER-READ-1 number “1” is channel

Feedback: DIMMER-READ-1,53,OK

53% is brightness

if failure, feedback for ERROR

2. set single channel output is 10% brightness:

Send: DIMMER-SEND-1,53 number "1" is channel 1 and 53% is brightness

Feedback: Receive DIMMER-SEND-1,53,OK

if failure, feedback for ERROR

3. Read all output channel state by one command:

Send: DIMMER-READ-ALL

Feedback: DIMMER-READ-ALL,12,0,0,8,0,0,0,10,0,0,0,0,0,53,OK

16 numbers is 16 channel brightness status value if failure, feedback for ERROR

4. Set multi channel output by one command:

Send: DIMMER-SEND-ALL,50,51,52,53,54,55,56,57,58,50,51,52,53,54,55,56,57,58

16 numbers is 16 channel brightness value to output

Feedback: DIMMER-READ-ALL,50,51,52,53,54,55,56,57,58,50,51,52,53,54,55,56,OK

if failure, feedback for ERROR

5. RF433MHz remoter button decode:

Feedback: key:LC:D55D0335 such as: LC:xxxxxxx

LC:xxxxxxx is button keycode. Every button will have different keycode.

6. Set 1-16 buttons keycode for 1-16 channel dimmer:

Send: DIMMER-STUDY-ALL,LC:D55D0335,LC:D55D0C3E,,,,,,,,,,,,,

Feedback: Signature:LC:D55D0335,LC:D55D0C3E,,,,,,,,,,,,,

This is sample set for 2 buttons.

Every keycode use "," as separator, total will have 16 ","

7. Check keys that have learned:

Send: DIMMER-STUDY-CHECK

Feedback: Signature:LC:D55D3062,LC:D55D0335,LC:D55DCCFE,LC:D55D0C3E,,,,,,,,,,,,,

MAX will feedback 16 keycodes.

8. Clear keycodes:

Send: DIMMER-STUDY-CLEAR

Feedback: Signature:,,,,,,,,,,,,,

9. Controller reboot:

Send: DIMMER-RESTART-GO

Feedback: DIMMER-RESTART-SUCCESS

D.RS485 command

Note: use device address "RS485 Slave Addr" for RS485 in network config webpage firstly.

RS485 communication parameters: 9600,8,N,1

RS485 address range: 01-99

1. Read single channel output state:

Send: DIMMER01-READ-2 number "01" is RS485 address, number "2" is channel

Feedback: DIMMER01-READ-2,53,OK

53% is brightness

if failure, feedback for ERROR

2. set single channel output is 10% brightness:

Send: DIMMER01-SEND-2,53 number "01" is RS485 address , number "2" is channel 2 and 53% is brightness

Feedback: Receive DIMMER01-SEND-2,53,OK

if failure, feedback for ERROR

3. Read all output channel state by one command:

Send: DIMMER01-READ-ALL number "01" is RS485 address

Feedback: DIMMER01-READ-ALL,50,53,0,0,0,0,0,0,0,0,0,0,0,0,0,0,OK

16 numbers is 16 channel brightness status value if failure, feedback for ERROR

4. Set multi channel output by one command:

Send: DIMMER01-SEND-ALL,50,51,52,53,54,55,56,57,58,50,51,52,53,54,55,56,57,58 number "01" is RS485 address

16 numbers is 16 channel brightness value to output

Feedback: DIMMER01-SEND-ALL,50,51,52,53,54,55,56,57,58,50,51,52,53,54,55,56,OK

if failure, feedback for ERROR

5. Set all dimmer output with same value:

Send: DIMMER01-SEND-SAME,50 number "01" is RS485 address

16 channel dimmer output all set to "50" value at the same time.

Feedback: DIMMER01-SEND-ALL,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,50,OK

if failure, feedback for ERROR