IOT Relay Programing Manual

V1.9.8

1 Protocol:Dingtian string....3
1.1 Query status command/KeepAlive....3
1.2 Control command....4
1.2.1 ON/OFF example....4
1.2.2 Delay example....4
1.2.3 Jogging example....4
2 Protocol:Dingtian binary....6
2.1 default setting....6
2.2 command....6
2.2.1 read relay status....7
2.2.2 write relay....8
2.2.3 write relay with delay....9
2.2.4 write relay with jogging....10
2.2.5 relay keep alive....11
3 Protocol:HTTP GET CGI....12
3.1 load relay status....12
3.2 set relay....15
3.3 load input status....18
3.4 Session check....20
3.4.1 web config to Enable HTTP Session check....20
3.4.2 HTTP CGI test tool....21
3.4.3 full example(success):....22
3.4.4 full example(fail)....23
4 Protocol:Modbus-RTU/TCP/ASCII....24
4.1 Registers....25
4.2 Modbus-RTU + Modbus-RTU Over TCP/UDP....27
4.2.1 0x03:Read holding register....27
4.2.2 0x06:Write Single Register....28
4.2.3 0x10: Write Multiple Register....29
4.3 Modbus-TCP/UDP....33
4.3.1 0x03:Read holding register....33
4.3.2 0x06:Write Single Register....34
4.3.3 0x10: Write Multiple Register....35
4.4 Modbus-ASCII + Modbus-ASCII Over TCP/UDP....39
4.4.1 0x03:Read holding register....39
4.4.2 0x06:Write Single Register....40
4.4.3 0x10: Write Multiple Register....41
5 Protocol:MQTT....45
5.1 MQTT Topic Fast View....47
5.2 MQTT Topic(firmware version < V2.15.869)....48

5.3 MQTT Topic(firmware version >= V2.15.869)....49
5.4 MQTT Topic(firmware version >= V2.17.xx)....50
5.5 MQTT LWT topic....53
6 Protocol:CoAP....54
6.1 Compile libcoap....54
6.2 Get relay status....54
6.3 Control relay(simple)....55
6.4 Control relay....56

1 Protocol:Dingtian string

Suport TCP client, TCP server, UDP, CAN/RS485

1.1 Query status command/KeepAlive

command code	00(2 character)	Return character (0:OFF, 1:ON) Format [relay1][relay2]...[relay32]:[input1][input2]...[input32]:[channel count] example(2 channel): 00:11:2 example(4 channel): 0000:1111:4 example(8 channel): 00000000:11111111:8 example(16 channel): 0000000000000000:1111111111111111:16 example(24 channel): 000000000000000000000000:111111111111111111111111:24 example(32 channel): 00000000000000000000000000000000:11111111111111111111111111111111:32

Remarks

1 The command code is a text string and does not need to be followed by a return.

1.2 Control command

X	R	C	P
1(ON) 2(OFF)	1~32 relay X for all relay	null(ON/OFF) :(Delay) *(Jogging)	C parameter

1.2.1 ON/OFF example

11 #relay 1 ON
21 #relay 1 OFF

13 #relay 3 ON
23 #relay 3 OFF

132 #relay 32 ON
232 #relay 32 OFF

1X #relay all ON
2X #relay all OFF

1.2.2 Delay example

Delay ON/OFF some time and then OFF/ON
Delay second range 1-65535

11:30 #relay 1 ON,delay 30second OFF
21:30 #relay 1 OFF,delay 30second ON

13:30 #relay 3 ON,delay 30second OFF
23:30 #relay 3 OFF,delay 30second ON

132:30 #relay 32 ON,delay 30second OFF
232:30 #relay 32 OFF,delay 30second ON

1X:30 #relay all ON,delay 30second OFF
2X:30 #relay all OFF,delay 30second ON

1.2.3 Jogging example

Jogging ON/OFF little time(ms) and then OFF/ON
Jogging time is 500ms(can change from web page),

11* #relay 1 ON,Jogging 500ms OFF
13* #relay 3 ON,Jogging 500ms OFF
132* #relay 32 ON,Jogging 500ms OFF
1X* #all relay ON,Jogging 500ms OFF

2 Protocol:Dingtian binary

Support Different network segment communication(Mulitcast only support UDP)
Mulitcast addr: 224.0.2.11

Support password

2.1 default setting

IP	192.168.1.100
Netmask	255.255.255.0
Gateway	192.168.1.1
UDP Port	60000
Multicast addr	224.0.2.11

2.2 command

data bytes >=2byte store format is LSB
example:0x1234,store format is 0x34,0x12

format

filed	bytes	comment
command	1	0xFF: set relay 0x07: multicast set relay
result(xor 0xAA)	1	pc->device: 0 xor 0xAA device->pc: result xor 0xAA result=0 success result=other fail
session	1	0~255 device reply the same
relay command	1	0: read relay status 1:write relay 2:write relay with delay 3:write relay with jogging 4:relay keep alive
password	2	0~9999 0:no password Password incurrent device no reply
command data	x	

2.2.1 read relay status

pc send

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	0: read relay status
password	2	0~9999 0:no password

device reply

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	0: read relay status
Relay status	1(2/4/8CH) 2(16CH) 3(24CH) 4(32CH)	Bit0~31 map to relay relay1~32 Bit=1 relay on Bit=0 relay off

Example:

pc send:

FF AA 00 00 34 12 # password 0x1234

device reply:

FF AA 00 00 01 # relay 1 on

2.2.2 write relay

pc send

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	1:write relay
password	2	0~9999 0:no password
relay mask	1(2/4/8CH) 2(16CH) 3(24CH) 4(32CH)	Bit0~31 map to relay relay1~32 Bit=1,relay need update
relay set	1(2/4/8CH) 2(16CH) 3(24CH) 4(32CH)	Bit0~31 map to relay relay1~32 Bit=1,relay on Bit=0,relay off

device reply

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	1:write relay

Example:

pc send:

FF AA 00 01 34 12 05 01 # relay 1 on, rely 3 off

device reply:

FF AA 00 01

2.2.3 write relay with delay

pc send

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	2:write relay with delay
password	2	0~9999 0:no password
relay index and relay on/off	1	Bit0=1 relay on Bit0=0 relay off Bit1~bit7=relay index
Relay delay second	2	1~65535 second

device reply

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	2:write relay with delay

Example:

pc send:

FF AA 00 02 34 12 03 05 # relay 1 on, delay 5 second off

device reply:

FF AA 00 02

2.2.4 write relay with jogging

pc send

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	3:write relay with jogging
password	2	0~9999 0:no password
relay index and relay on/off	1	Bit0=1 relay on Bit0=0 relay off Bit1~bit7=relay index

device reply

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 device not change
relay command	1	3:write relay with jogging

Example:

pc send:

FF AA 00 03 34 12 05 05 # relay 2 on, jogging

device reply:

FF AA 00 03

2.2.5 relay keep alive

device send

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 pc not change
relay command	1	4: relay keep alive
device MAC	6	device MAC address
Relay status	1	Bit0~7 map to relay relay1~8 Bit=1 relay on Bit=0 relay off

pc reply

filed	bytes	comment
command	1	0xFF
result(xor 0xAA)	1	0 xor 0xAA=0xAA
session	1	0~255 pc not change
relay command	1	4: relay keep alive

Example:

device send:

FF AA 00 04 BC 34 88 12 34 56 00 # MAC BC:34:88:12:34:56 00:all relay off

pc reply:

FF AA 00 00

3 Protocol:HTTP GET CGI

Relay board as HTTP server, accept HTTP GET CGI request.
Support CGI relay on/off
Support CGI relay jogging
Support CGI relay delay
Support CGI password verification

Support CGI Session check(details: 3.4 Session check)

3.1 load relay status

HTTP GET request

parameter	filed	data	comment
1	CGI API	relay_cgi_load.cgi	cgi changeable suffix relay_cgi_load.cgi, relay_cgi_load.php, relay_cgi_load.cs is work ok

HTTP GET respond

parameter	filed	data	comment
1	result	0	0: ok other fail
2	relay count	2/4/8/16/24/32	
3	relay 1 status	0/1	0:off 1:on
4	relay 2 status	0/1	0:off 1:on
5	relay 3 status	0/1	0:off 1:on
6	relay 4 status	0/1	0:off 1:on
7	relay 5 status	0/1	0:off 1:on
8	relay 6 status	0/1	0:off 1:on
9	relay 7 status	0/1	0:off 1:on
10	relay 8 status	0/1	0:off 1:on
11	relay 9 status	0/1	0:off 1:on

12	relay 10 status	0/1	0:off 1:on
13	relay 11 status	0/1	0:off 1:on
14	relay 12 status	0/1	0:off 1:on
15	relay 13 status	0/1	0:off 1:on
16	relay 14 status	0/1	0:off 1:on
17	relay 15 status	0/1	0:off 1:on
18	relay 16 status	0/1	0:off 1:on
19	relay 17 status	0/1	0:off 1:on
20	relay 18 status	0/1	0:off 1:on
21	relay 19 status	0/1	0:off 1:on
22	relay 20 status	0/1	0:off 1:on
23	relay 21 status	0/1	0:off 1:on
24	relay 22 status	0/1	0:off 1:on
25	relay 23 status	0/1	0:off 1:on
26	relay 24 status	0/1	0:off 1:on
27	relay 25 status	0/1	0:off 1:on
28	relay 26 status	0/1	0:off 1:on
29	relay 27 status	0/1	0:off 1:on
30	relay 28 status	0/1	0:off 1:on
31	relay 29 status	0/1	0:off 1:on
32	relay 30 status	0/1	0:off 1:on
33	relay 31 status	0/1	0:off

			1:on
34	relay 32 status	0/1	0:off 1:on

example(4 channel relay):

HTTP GET request

http://192.168.1.100/relay_cgi_load.cgi # request relay board HTTP CGI API

HTTP GET respond

&0&4&1&0&1&0& # ok,4 relay,relay 1 on,relay 2 off,relay 3 on, relay 4 off

3.2 set relay

HTTP GET request

parameter	filed	data	comment
1	CGI API	relay_cgi.cgi	cgi suffix variable relay_cgi.cgi, relay_cgi.php, relay_cgi.cs is work ok
2	type	0/1/2	0:relay on/off 1:relay jogging 2:relay delay
3	relay	0~31	
4	on	0/1	0:off 1:on
5	time	0 1~255 1~65535	0:type 0:time 1:type 1~255:time(1=100ms) 2:type 1~65535:time(second)
6	pwd	0~9999	0~9999 Password incurrent device no respond

HTTP GET respond

parameter	filed	data	comment
1	result	0	0: ok other fail
2	type	0/1/2	0:relay on/off 1:relay jogging 2:relay delay
3	relay	0~31	0:relay 1 1:relay 2 ... 31:relay 32
4	on	0/1	0:off 1:on
5	time	0 1~255 1~65535	0:type 0:time

			1:type 1~255:time(1=100ms) 2:type 1~65535:time(second)

example 1(relay on):
HTTP GET request(request relay board HTTP CGI API, set relay 0 on ,time 0,password 0)
http://192.168.1.100/relay_cgi.cgi?type=0&relay=0&on=1&time=0&pwd=0&
HTTP GET respond
&0&0&0&1&0& # ok, type 0 on/off,relay 0 on,time 0

example 2(relay off):
HTTP GET request(request relay board HTTP CGI API, set relay 0 off ,time 0,password 0)
http://192.168.1.100/relay_cgi.cgi?type=0&relay=0&on=0&time=0&pwd=0&
HTTP GET respond
&0&0&0&0&0& # ok, type 0 on/off,relay 0 off,time 0

example 3(relay 1 jogging on):
HTTP GET request(request relay board HTTP CGI API, set relay 1 jogging on ,time 500ms,password 4660)
http://192.168.1.100/relay_cgi.cgi?type=1&relay=1&on=1&time=5&pwd=4660&
HTTP GET respond
&0&1&1&1&5& # ok, type 1 jogging,relay 1 on,time 5(500ms)

example 4(relay 1 jogging off):
HTTP GET request(request relay board HTTP CGI API, set relay 1 jogging off,time 500ms,password 4660)
http://192.168.1.100/relay_cgi.cgi?type=1&relay=1&on=0&time=5&pwd=4660&
HTTP GET respond
&0&1&1&0&5& # ok, type 1 jogging,relay 1 off,time 5(500ms)

example 5(relay 1 on delay 10 second off):
HTTP GET request(request relay board HTTP CGI API, set relay 1 on delay 10 second off ,time 5 second,password 4660)
http://192.168.1.100/relay_cgi.cgi?type=2&relay=1&on=1&time=10&pwd=4660&
HTTP GET respond
&0&2&1&1&10& # ok, type 2 delay,relay 1 on,time 10 second

example 6(relay 1 off delay 10 second on):
HTTP GET request(request relay board HTTP CGI API, set relay 1 off delay 10 second on ,time 5 second,password 4660)
http://192.168.1.100/relay_cgi.cgi?type=2&relay=1&on=0&time=10&pwd=4660&

HTTP GET respond

&0&2&1&0&10& # ok, type 2 delay,relay 1 off,time 10 second

3.3 load input status

HTTP GET request

parameter	filed	data	comment
1	CGI API	input.cgi	cgi changeable suffix input.cgi, input.php, input.cs is work ok

HTTP GET respond

parameter	filed	data	comment
1	result	0	0: ok other fail
2	Input start postion	0	default: 0
3	input count	2/4/8/16/24/32	
4	relay 1 status	0/1	0:low 1:high
5	relay 2 status	0/1	0:low 1:high
6	relay 3 status	0/1	0:low 1:high
7	relay 4 status	0/1	0:low 1:high
8	relay 5 status	0/1	0:low 1:high
9	relay 6 status	0/1	0:low 1:high
10	relay 7 status	0/1	0:low 1:high
11	relay 8 status	0/1	0:low 1:high
12	relay 9 status	0/1	0:low 1:high
13	relay 10 status	0/1	0:low 1:high
14	relay 11 status	0/1	0:low 1:high
15	relay 12 status	0/1	0:low 1:high
16	relay 13 status	0/1	0:low 1:high

17	relay 14 status	0/1	0:low 1:high
18	relay 15 status	0/1	0:low 1:high
19	relay 16 status	0/1	0:low 1:high
20	relay 17 status	0/1	0:low 1:high
21	relay 18 status	0/1	0:low 1:high
22	relay 19 status	0/1	0:low 1:high
23	relay 20 status	0/1	0:low 1:high
24	relay 21 status	0/1	0:low 1:high
25	relay 22 status	0/1	0:low 1:high
26	relay 23 status	0/1	0:low 1:high
27	relay 24 status	0/1	0:low 1:high
28	relay 25 status	0/1	0:low 1:high
29	relay 26 status	0/1	0:low 1:high
30	relay 27 status	0/1	0:low 1:high
31	relay 28 status	0/1	0:low 1:high
32	relay 29 status	0/1	0:low 1:high
33	relay 30 status	0/1	0:low 1:high
34	relay 31 status	0/1	0:low 1:high
35	relay 32 status	0/1	0:low 1:high

example(4 channel relay):

HTTP GET request

http://192.168.1.100/input.cgi # request relay board HTTP CGI API

HTTP GET respond

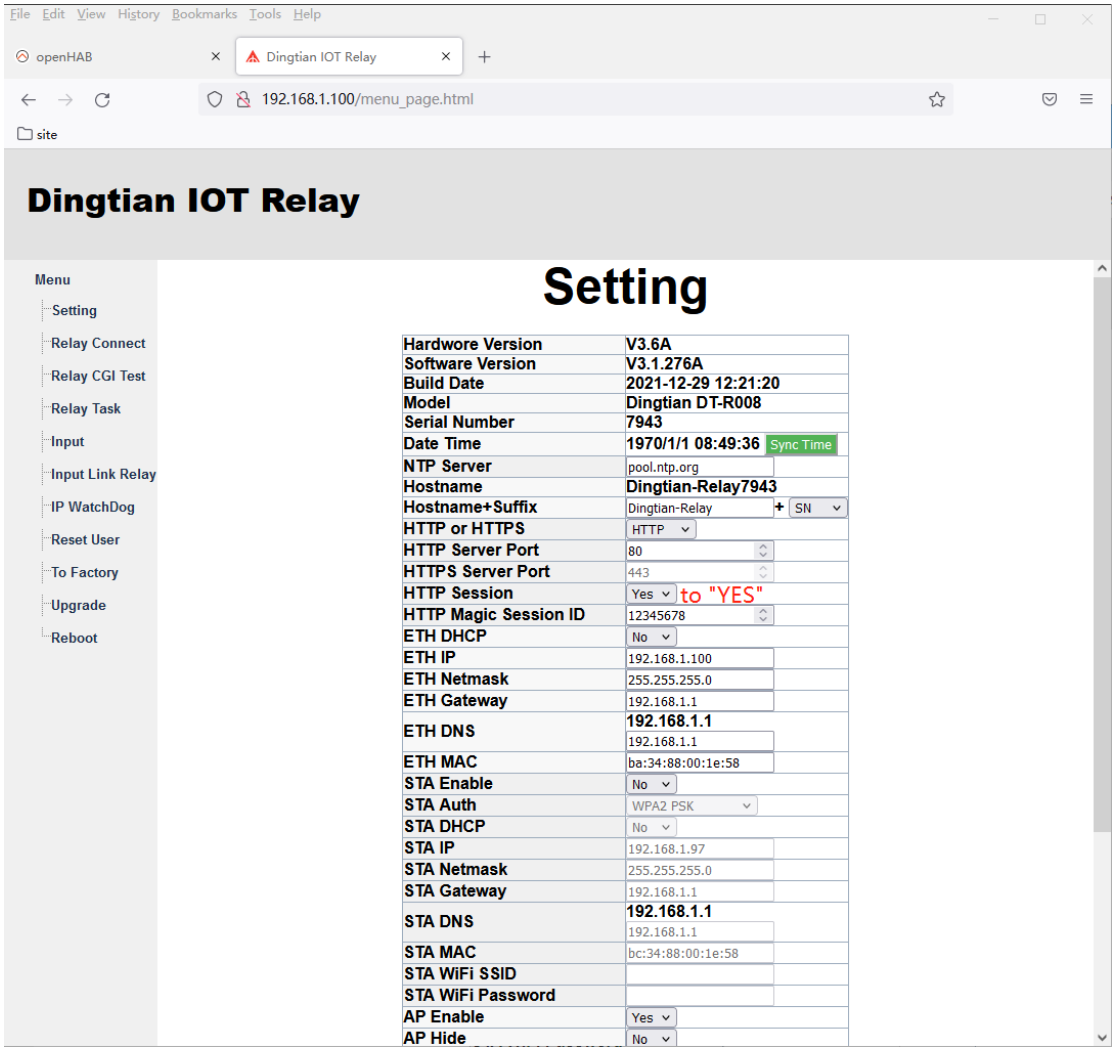
&0&0&4&1&0&1&0& # ok,reserve,4 input,input 1 high,relay 2 low,relay 3 high, relay 4 low

3.4 Session check

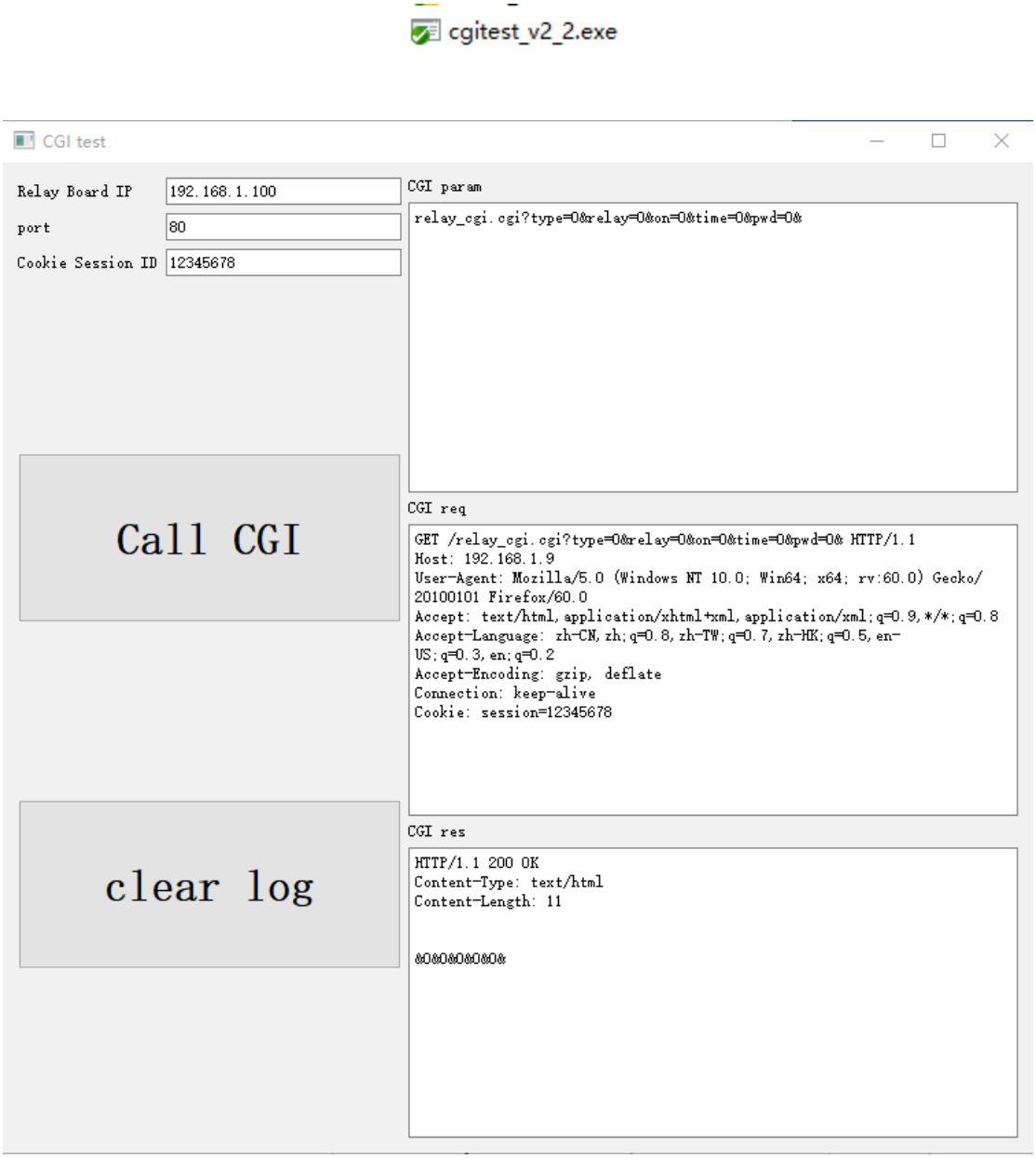
The HTTP CGI session check is implemented by adding a “Cookie” header

“Cookie” header example:
Cookie: session=12345678

3.4.1 web config to Enable HTTP Session check



3.4.2 HTTP CGI test tool



3.4.3 full example(success):

HTTP GET request:

GET /relay_cgi.cgi?type=0&relay=0&on=1&time=0&pwd=0& HTTP/1.1

Host: 192.168.1.9

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Connection: keep-alive

Cookie: session=12345678

HTTP GET response:

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 11

&0&0&0&1&0&

3.4.4 full example(fail)

HTTP GET request:

GET /relay_cgi.cgi?type=0&relay=0&on=0&time=0&pwd=0& HTTP/1.1

Host: 192.168.1.9

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2

Accept-Encoding: gzip, deflate

Connection: keep-alive

Cookie: session=1234567

HTTP GET response:

HTTP/1.1 200 OK

Content-Type: text/html

Content-Length: 7

&302&/&

4 Protocol:Modbus-RTU/TCP/ASCII

Support Modbus:

Modbus-RTU

Modbus-TCP/UDP

Modbus-ASCII

Modbus-RTU Over TCP/UDP

Modbus-ASCII Over TCP/UDP

Support Modbus Function:

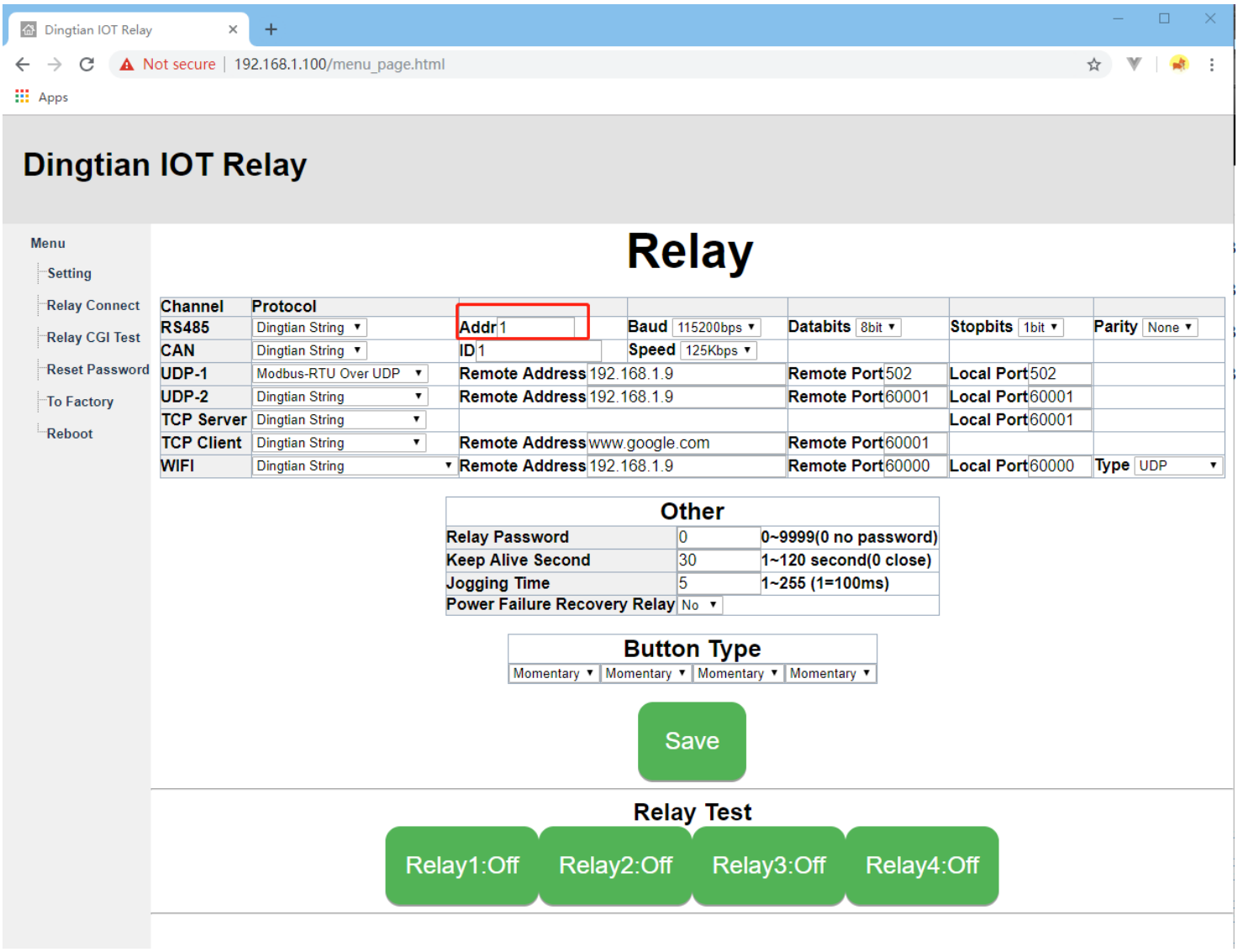
0x03read holding register

0x06Write Single register

0x10Wirte Multile register(CAN bus not support)

Notice:

Modbus-RTU Over UDP/TCP,Modbus-ASCII Over UDP/TCP use RS485 addr



4.1 Registers

Register	Name	0x03/0x06/0x10	Value
0x0000	Relay Count	0x03	2/4/8/16/32
0x0001	Relay Status	0x03	bit0~7 map to relay1~8
0x0002	Write Relay	0x06	bit0~7 new status of relay1~8(bit=1 ON,bit=0 OFF) bit8~15 map to relay1~8 need update(bit=1 Update)
0x0003	Advance Write Type	0x10	Bit0~5: 1:Write ON/OFF 2:Write with delay 3:Write with Jogging bit6~15:(only for Type:Write ON/OFF(1)) relay group:0~3 r1~8:G0 r9~16:G1 r17~24:G2 r25~32:G3
0x0004	Advance Write Password	0x10	Password 0~65535 when password in current do nothing
0x0005	Advance Write Relay	0x10	Type:Write ON/OFF(1) bit0~7 new status of relay1~8(bit=1 ON,bit=0 OFF) bit8~15 map to relay1~8 need update(bit=1 Update) Type:Write with delay(2) bit0: bit=1 ON,bit=0 OFF bit1~7:relay index 0~31 Type:Write with Jogging(3) bit0: bit=1 ON,bit=0 OFF bit1~7:relay index 0~31
0x0006	Advance Write Time	0x10	Type:Write ON/OFF(1) 0 Type:Write with delay(2) Number of Second need delay Type:Write with Jogging(3) Number of 100ms need jogging(1=100ms)
0x0007	Expand Write Status Group	0x10	relay1~16:G0 relay16~32:G1
0x0008	Expand Write Relay Mask	0x10	bit0~15 map to relay G0:R1~16 / G1:R17~32 need update(bit=1 Update)
0x0009	Expand Write Relay	0x10	bit0~15 map to relay G0:R1~16 / G1:R17~32
0x000A	Expand Input Status 1~16	0x03	input1~16
0x000B	Expand Input Status 17~32	0x03	input17~32
0x000C	Expand Input Status 33~48	0x03	input33~48
0x000D	Expand Input Status 49~64	0x03	input49~64
0x000E	Expand Relay Status 1~16	0x03	relay1~16

0x000F	Expand Relay Status 17~32	0x03	relay17~32
0x0010	Expand Relay Status 33~48	0x03	relay33~48
0x0011	Expand Relay Status 49~64	0x03	relay49~64
0x0012	Expand Write Relay Mask 1~16	0x10	relay1~16 mask bits; bit=1 need change,bit=0 no change
0x0013	Expand Write Relay mask 17~32	0x10	relay17~32 mask bits; bit=1 need change,bit=0 no change
0x0014	Expand Write Relay bits 1~16	0x10	relay1~16 relay bits; bit=1(ON),bit=0(OFF)
0x0015	Expand Write Relay bits 17~32	0x10	relay17~32 relay bits, bit=1(ON),bit=0(OFF)

Notice:

1、0x0003~6/0x0007~9/0x0012~15 is block， must written at the same time.

4.2 Modbus-RTU + Modbus-RTU Over TCP/UDP

4.2.1 0x03:Read holding register

Read all Relay Status
Send:
01 03 0000 0002 C40B
Recv:
01 03 04 0004 0000 BBF2

Read all Input Status(2/4/8 channel)
Send:
01 03 000A 0001 A408
Recv:
01 03 02 FFFF B9F4

Read all Input Status(support 16/24/32 channel)
Send:
01 03 000A 0002 E409
Recv:
01 03 04 FFFF 00FF BA57

Read all Relay Status(2/4/8 channel)
Send:
01 03 000E 0001 E5C9
Recv:
01 03 02 0000 B844

Read all Relay Status(support 16/24/32 channel)
Send:
01 03 000E 0002 A5C8
Recv:
01 03 04 0000 0080 FB93

4.2.2 0x06:Write Single Register

4 Relay All ON
Send:
01 06 0002 0f0f 6DFE
Recv:
01 06 0002 0f0f 6DFE

4 Relay All OFF
Send:
01 06 0002 0f00 2DFA
Recv:
01 06 0002 0f00 2DFA

Relay 1,4 ON; Relay 2,3 stay the same
Send:
01 06 0002 0909 EE5C
Recv:
01 06 0002 0909 EE5C

4.2.3 0x10: Write Multiple Register

1、ON/OFF

4 Relay All ON

Send:

01 10 0003 0004 08 0001 0000 0f0f 0000 91A9

Recv:

01 10 0003 0004 31 CA

4 Relay All OFF

Send:

01 10 0003 0004 08 0001 0000 0f00 0000 A1AA

Recv:

01 10 0003 0004 31 CA

Relay 2,3 ON; Relay 1,4 stay the same

Send:

01 10 0003 0004 08 0001 0000 0606 0000 4237

Recv:

01 10 0003 0004 31 CA

2、Delay

Relay 1 OFF Delay 5 Second ON

Send:

01 10 0003 0004 08 0002 0000 0000 0005 51BD

Recv:

01 10 0003 0004 31 CA

Relay 1 ON Delay 5 Second OFF

Send:

01 10 0003 0004 08 0002 0000 0001 0005 007D

Recv:

01 10 0003 0004 31 CA

Relay 2 ON Delay 5 Second OFF

Send:

01 10 0003 0004 08 0002 0000 0003 0005 A1BD

Recv:

01 10 0003 0004 31 CA

Relay 3 ON Delay 5 Second OFF

Send:

01 10 0003 0004 08 0002 0000 0005 0005 41BC

Recv:

01 10 0003 0004 31 CA

Relay 4 ON Delay 5 Second OFF

Send:

01 10 0003 0004 08 0002 0000 0007 0005 E07C

Recv:

01 10 0003 0004 31 CA

3、Jogging

Relay 4 ON Joging 500ms OFF,Password 0x1234

Send:

01 10 0003 0004 08 0003 1234 0007 0005 420A

Recv:

01 10 0003 0004 31 CA

Relay 1 OFF Joging 500ms ON

Send:

01 10 0003 0004 08 0003 0000 0000 0005 417D

Recv:

01 10 0003 0004 31 CA

Relay 1 ON Joging 500ms OFF

Send:

01 10 0003 0004 08 0003 0000 0001 0005 10BD

Recv:

01 10 0003 0004 31 CA

Relay 2 ON Joging 500ms OFF

Send:

01 10 0003 0004 08 0003 0000 0003 0005 B17D

Recv:

01 10 0003 0004 31 CA

Relay 3 ON Joging 500ms OFF

Send:

01 10 0003 0004 08 0003 0000 0005 0005 517C

Recv:

01 10 0003 0004 31 CA

Relay 4 ON Joging 500ms OFF

Send:

01 10 0003 0004 08 0003 0000 0007 0005 F0BC

Recv:

01 10 0003 0004 31 CA

4 16/24/32 channel write relay

8 channel all relay ON

Send:

01 10 0012 0004 08 00FF 0000 00FF 0000 B17D

Recv:

01 10 0012 0004 61CF

8 channel all relay OFF

Send:

01 10 0012 0004 08 00FF 0000 0000 0000 814D

Recv:

01 10 0012 0004 61CF

16 channel all relay ON

Send:

01 10 0012 0004 08 FFFF 0000 FFFF 0000 CE6D

Recv:

01 10 0012 0004 61CF

16 channel all relay OFF

Send:

01 10 0012 0004 08 FFFF 0000 0000 0000 CE49

Recv:

01 10 0012 0004 61CF

24 channel all relay ON

Send:

01 10 0012 0004 08 FFFF 00FF FFFF 00FF 9A39

Recv:

01 10 0012 0004 61CF

24 channel all relay OFF

Send:

01 10 0012 0004 08 FFFF 00FF 0000 0000 DA5D

Recv:

01 10 0012 0004 61CF

32 channel all relay ON

Send:

01 10 0012 0004 08 FFFF FFFF FFFF FFFF CFC6

Recv:

01 10 0012 0004 61CF

32 channel all relay OFF

Send:
01 10 0012 0004 08 FFFF FFFF 0000 0000 CE52
Recv:
01 10 0012 0004 61CF

4.3 Modbus-TCP/UDP

4.3.1 0x03:Read holding register

Read all Relay Status
Send:
0000 0000 0006 FF 03 0000 0002
Recv:
0000 0000 0007 FF 03 04 0004 000F

Read all Relay Status(Extend support 16/24/32 channel relay board)
Send:
0000 0000 0006 FF 03 0000 0002
Recv:
0000 0000 0007 FF 03 04 0004 000F

Read all Input Status(2/4/8 channel)
Send:
0000 0000 0006 FF 03 000A 0001
Recv:
0000 0000 0005 FF 03 02 FFFF

Read all Input Status(support 16/24/32 channel)
Send:
0000 0000 0006 FF 03 000A 0002
Recv:
0000 0000 0007 FF 03 04 FFFF 00FF

Read all Relay Status(2/4/8 channel)
Send:
0000 0000 0006 FF 03 000E 0001
Recv:
0000 0000 0005 FF 03 02 0000

Read all Relay Status(support 16/24/32 channel)
Send:
0000 0000 0006 FF 03 000E 0002
Recv:
0000 0000 0007 FF 03 04 0000 0000

4.3.2 0x06:Write Single Register

4 Relay All ON
Send:
0000 0000 0006 FF 06 0002 0f0f
Recv:
0000 0000 0006 FF 06 0002 0f0f

4 Relay All OFF
Send:
0000 0000 0006 FF 06 0002 0f00
Recv:
01 06 0002 0f00 2DFA

Relay 1,4 ON; Relay 2,3 stay the same
Send:
0000 0000 0006 FF 06 0002 0909
Recv:
0000 0000 0006 FF 06 0002 0909

4.3.3 0x10: Write Multiple Register

1 ON/OFF

4 Relay All ON

Send:

0001 0000 000F FF 10 0003 0004 08 0001 0000 0f0f 0000

Recv:

0001 0000 0006 FF 10 0003 0004

4 Relay All OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0001 0000 0f00 0000

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 2,3 ON; Relay 1,4 stay the same

Send:

0001 0000 000F FF 10 0003 0004 08 0001 0000 0606 0000

Recv:

0001 0000 0006 FF 10 0003 0004

2 Delay

Relay 1 OFF Delay 5 Second ON

Send:

0001 0000 000F FF 10 0003 0004 08 0002 0000 0000 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 1 ON Delay 5 Second OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0002 0000 0001 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 2 ON Delay 5 Second OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0002 0000 0003 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 3 ON Delay 5 Second OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0002 0000 0005 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 4 ON Delay 5 Second OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0002 0000 0007 0005

Recv:

0001 0000 0006 FF 10 0003 0004

3 Jogging

Relay 4 ON Joging 500ms OFF,Password 0x1234

Send:

0001 0000 000F FF 10 0003 0004 08 0003 1234 0007 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 1 OFF Joging 500ms ON

Send:

0001 0000 000F FF 10 0003 0004 08 0003 0000 0000 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 1 ON Joging 500ms OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0003 0000 0001 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 2 ON Joging 500ms OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0003 0000 0003 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 3 ON Joging 500ms OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0003 0000 0005 0005

Recv:

0001 0000 0006 FF 10 0003 0004

Relay 4 ON Joging 500ms OFF

Send:

0001 0000 000F FF 10 0003 0004 08 0003 0000 0007 0005

Recv:

0001 0000 0006 FF 10 0003 0004

4 16/24/32 channel write relay

8 channel all relay ON

Send:

0001 0000 000F FF 10 0012 0004 08 00FF 0000 00FF 0000

Recv:

0001 0000 0006 FF 10 0012 0004

8 channel all relay OFF

Send:

0001 0000 000F FF 10 0012 0004 08 00FF 0000 0000 0000

Recv:

0001 0000 0006 FF 10 0012 0004

16 channel all relay ON

Send:

0001 0000 000F FF 10 0012 0004 08 FFFF 0000 FFFF 0000

Recv:

0001 0000 0006 FF 10 0012 0004

16 channel all relay OFF

Send:

0001 0000 000F FF 10 0012 0004 08 FFFF 0000 0000 0000

Recv:

0001 0000 0006 FF 10 0012 0004

24 channel all relay ON

Send:

0001 0000 000F FF 10 0012 0004 08 FFFF 00FF FFFF 00FF

Recv:

0001 0000 0006 FF 10 0012 0004

24 channel all relay OFF

Send:

0001 0000 000F FF 10 0012 0004 08 FFFF 00FF 0000 0000

Recv:

0001 0000 0006 FF 10 0012 0004

32 channel all relay ON

Send:

0001 0000 000F FF 10 0012 0004 08 FFFF FFFF FFFF FFFF

Recv:

0001 0000 0006 FF 10 0012 0004

32 channel all relay OFF

Send:

0001 0000 000F FF 10 0012 0004 08 FFFF FFFF 0000 0000

Recv:

0001 0000 0006 FF 10 0012 0004

4.4 Modbus-ASCII + Modbus-ASCII Over TCP/UDP

4.4.1 0x03:Read holding register

Read all Relay Status
Send:
ASCII : 01 03 0000 0002 BA \r\n
HEX 3A 3031 3033 30303030 30303032 4241 0D0A
Recv:
ASCII : 01 03 04 0004 0000 54 \r\n
HEX 3A 3031 3033 3034 30303034 30303030 3534 0D0A

Read all Input Status(2/4/8 channel)
Send:
ASCII : 01 03 000A 0001 AA \r\n
HEX 3A 3031 3033 30303041 30303031 4141 0D0A
Recv:
ASCII : 01 03 02 FFFF C2 \r\n
HEX 3A 3031 3033 3032 46464646 4332 0D0A

Read all Input Status(support 16/24/32 channel)
Send:
ASCII : 01 03 000A 0002 A9 \r\n
HEX 3A 3031 3033 30303041 30303032 4139 0D0A
Recv:
ASCII : 01 03 04 FFFF 00FF D4 \r\n
HEX 3A 3031 3033 3034 46464646 30304646 4434 0D0A

Read all Relay Status(2/4/8 channel)
Send:
ASCII : 01 03 000E 0001 A6 \r\n
HEX 3A 3031 3033 30303045 30303031 4136 0D0A
Recv:
ASCII : 01 03 02 0000 1A \r\n
HEX 3A 3031 3033 3032 30303030 3141 0D0A

Read all Relay Status(support 16/24/32 channel)
Send:
ASCII : 01 03 000E 0002 A5 \r\n
HEX 3A 3031 3033 30303045 30303032 4135 0D0A
Recv:
ASCII : 01 03 04 0000 0080 FB93 \r\n
HEX 3A 3031 3033 3034 30303030 30303830 3530 0D0A

4.4.2 0x06:Write Single Register

4 Relay All ON

Send:

ASCII : 01 06 0002 0F0F 8B \r\n

HEX 3A 3031 3036 30303032 30463046 3842 0D0A

Recv:

ASCII : 01 06 0002 0F0F 8B \r\n

HEX 3A 3031 3036 30303032 30463046 3842 0D0A

4 Relay All OFF

Send:

ASCII : 01 06 0002 0F00 A1 \r\n

HEX 3A 3031 3036 30303032 30463030 4131 0D0A

Recv:

ASCII : 01 06 0002 0F00 A1 \r\n

HEX 3A 3031 3036 30303032 30463030 4131 0D0A

4.4.3 0x10: Write Multiple Register

1 ON/OFF

4 Relay All ON

Send:

ASCII :01 10 0003 0004 08 0001 0000 0F0F 0000 22 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303031 30303030 30463046 30303030 3232 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

4 Relay All OFF

Send:

ASCII :01 10 0003 0004 08 0001 0000 0F00 0000 38 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303031 30303030 30463030 30303030 3338 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

Relay 2,3 ON; Relay 1,4 stay the same

Send:

ASCII :01 10 0003 0004 08 0001 0000 0606 0000 42 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303031 30303030 30363036 30303030 3432 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

2 Delay

Relay 1 ON Delay 5 Second OFF

Send:

ASCII :01 10 0003 0004 08 0002 0000 0001 0005 47 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303032 30303030 30303031 30303035 3437 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

Relay 4 ON Delay 5 Second OFF

Send:

ASCII :01 10 0003 0004 08 0002 0000 0007 0005 41 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303032 30303030 30303037 30303035 3431 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

3 Jogging

Relay 4 ON Joging 500ms OFF,Password 0x1234

Send:

ASCII :01 10 0003 0004 08 0003 1234 0007 0005 36 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303033 31323334 30303037 30303035 3336 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

Relay 1 ON Joging 500ms OFF

Send:

ASCII :01 10 0003 0004 08 0003 0000 0001 0005 46 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303033 30303030 30303031 30303035 3436 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

Relay 4 ON Joging 500ms OFF

Send:

ASCII :01 10 0003 0004 08 0003 0000 0007 0005 40 \r\n

HEX 3A 3031 3130 30303033 30303034 3038 30303033 30303030 30303037 30303035 3430 0D0A

Recv:

ASCII :01 10 0003 0004 B7 \r\n

HEX 3A 3031 3130 30303033 30303034 4237 0D0A

4 16/24/32 channel write relay

8 channel all relay ON

Send:

ASCII :01 10 0012 0004 08 00FF 0000 00FF 0000 F7 \r\n

HEX 3A 3031 3130 30303132 30303034 3038 30304646 30303030 30304646 30303030 4637 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

8 channel all relay OFF

Send:

ASCII :01 10 0012 0004 08 00FF 0000 0000 0000 23 \r\n

HEX 3A 3031 3130 30303132 30303034 3038 30304646 30303030 30303030 30303030 3233 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

16 channel all relay ON

Send:

ASCII :01 10 0012 0004 08 FFFF 0000 FFFF 0000 9F \r\n

HEX 3A 3031 3130 30303132 30303034 3038 46464646 30303030 46464646 30303030 3946 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

16 channel all relay OFF

Send:

ASCII :01 10 0012 0004 08 FFFF 0000 0000 0000 F7 \r\n

HEX 3A 3031 3130 30303132 30303034 3038 46464646 30303030 30303030 30303030 4637 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

24 channel all relay ON

Send:

ASCII :01 10 0012 0004 08 FFFF 00FF FFFF 00FF \r\n

HEX 3A 3031 3130 30303132 30303034 3038 46464646 30304646 46464646 30304646 3437 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

24 channel all relay OFF

Send:

ASCII :01 10 0012 0004 08 FFFF 00FF 0000 0000 CB \r\n

HEX 3A 3031 3130 30303132 30303034 3038 46464646 30304646 30303030 30303030 4342 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

32 channel all relay ON

Send:

ASCII :01 10 0012 0004 08 FFFF FFFF FFFF FFFF CFC6 \r\n

HEX 3A 3031 3130 30303132 30303034 3038 46464646 46464646 46464646 46464646 4546 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

32 channel all relay OFF

Send:

ASCII :01 10 0012 0004 08 FFFF FFFF 0000 0000 CE52 \r\n

HEX 3A 3031 3130 30303132 30303034 3038 46464646 46464646 30303030 30303030 3946 0D0A

Recv:

ASCII :01 10 0012 0004 B7 \r\n

HEX 3A 3031 3130 30303132 30303034 4237 0D0A

5 Protocol:MQTT

MQTT version 3.1.1

Relay board as MQTT client,communcation with broker..

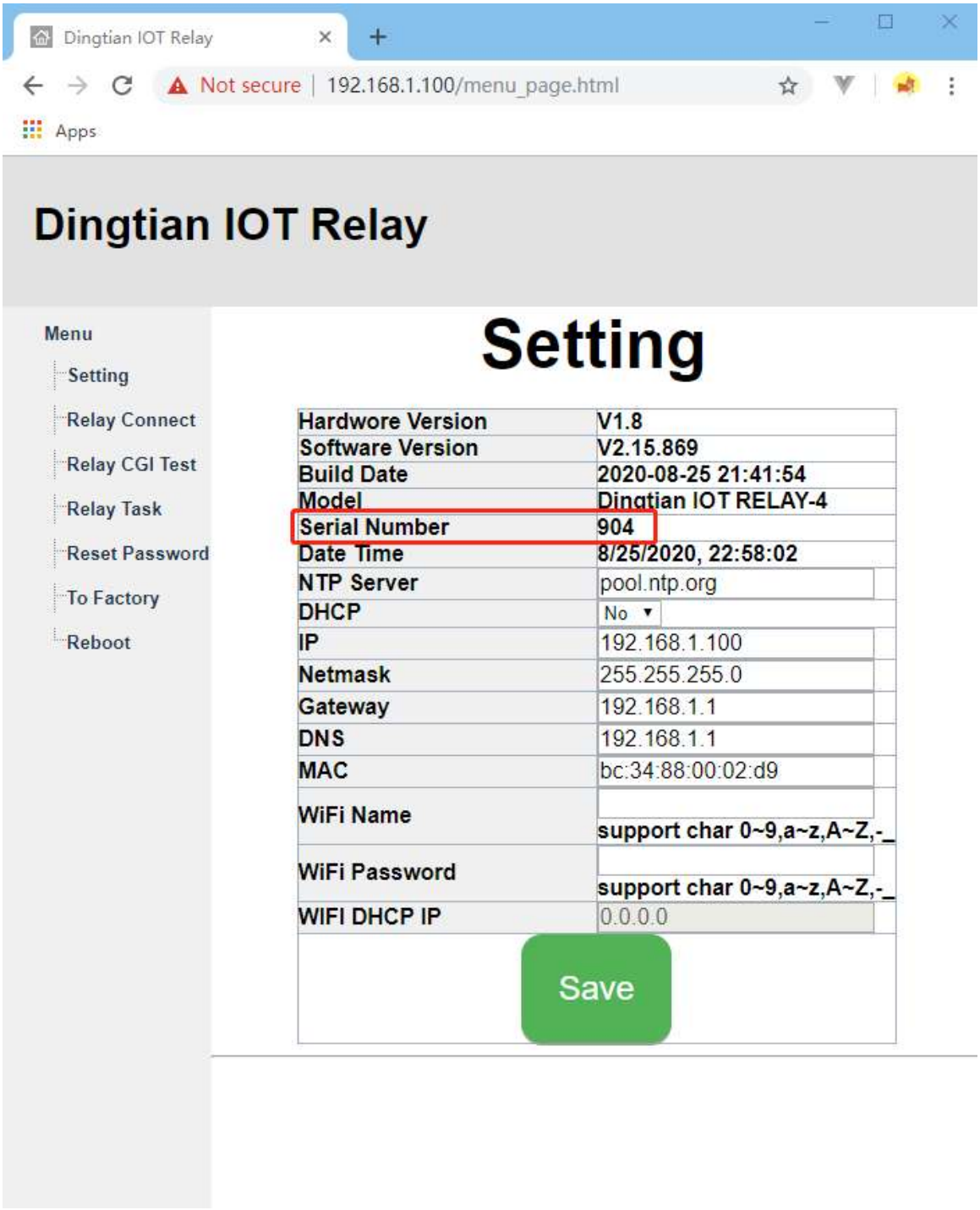
Support relay on/off

Support relay jogging

Support relay delay

Support password verification

Support LWT(details: 5.5 MQTT LWT topic)



Relay board Ethernet MQTT Client Id
dingtian-relay+SN
Relay board WiFi MQTT Client Id
dingtian-wrelay+SN

example:
below relay board “Serial Number” is 1868
so ETH MQTT client id is:dingtian-relay1868
so WiFI MQTT client id is:dingtian-wrelay1868

5.1 MQTT Topic Fast View

firmware version <V2.15.869

/dingtian/relay/in/control

/dingtian/relay/out/relayX

firmware version >=V2.15.869

/dingtian/relaySN/in/control

/dingtian/relaySN/out/relayX

firmware version >= V2.17.xx

ETH

/dingtian/relaySN/in/control

/dingtian/relaySN/in/rX

/dingtian/relaySN/out/rX

/dingtian/relaySN/out/iX

/dingtian/relaySN/out/relayX

/dingtian/relaySN/out/inputX

/dingtian/relaySN/out/ip

/dingtian/relaySN/out/sn

/dingtian/relaySN/out/mac

/dingtian/relaySN/out/input_cnt

/dingtian/relaySN/out/relay_cnt

WiFi

/dingtian/wrelaySN/in/control

/dingtian/wrelaySN/in/rX

/dingtian/wrelaySN/out/rX

/dingtian/wrelaySN/out/iX

/dingtian/wrelaySN/out/relayX

/dingtian/wrelaySN/out/inputX

/dingtian/wrelaySN/out/ip

/dingtian/wrelaySN/out/sn

/dingtian/wrelaySN/out/mac

/dingtian/wrelaySN/out/input_cnt

/dingtian/wrelaySN/out/relay_cnt

5.2 MQTT Topic(firmware version < V2.15.869)

<table>
<tr><td>topic</td><td>type</td><td colspan="3">value</td></tr>
<tr><td rowspan="7">/dingtian/relay/in/control</td><td rowspan="7">subscribe</td><td>parameter</td><td>filed</td><td>data</td></tr>
<tr><td>type</td><td>command type</td><td>ON/OFF
DELAY
JOGGING</td></tr>
<tr><td>idx</td><td>relay index</td><td>1~32</td></tr>
<tr><td>status</td><td>relay status</td><td>ON,OFF</td></tr>
<tr><td>time</td><td>time for type</td><td>ON/OFF:0
DELAY:1~65535second
JOGGING:1~255*100ms</td></tr>
<tr><td>pass</td><td>password</td><td>0~9999</td></tr>
<tr><td colspan="3">example:
{"type":"ON/OFF",'idx':'1',"status":"ON","time":"0","pass":"0"}
{"type":"DELAY",'idx':'2',"status":"ON","time":"5","pass":"0"}
{"type":"JOGGING",'idx':'3',"status":"ON","time":"5","pass":"0"}
{"type":"ON/OFF",'idx':'4',"status":"OFF","time":"0","pass":"0"}</td></tr>
<tr><td rowspan="4">/dingtian/relay/out/relayX

example:
/dingtian/relay/out/relay1
/dingtian/relay/out/relay2</td><td rowspan="4">publish</td><td>parameter</td><td>filed</td><td>data</td></tr>
<tr><td>idx</td><td>relay index</td><td>1~32</td></tr>
<tr><td>status</td><td>relay status</td><td>ON,OFF</td></tr>
<tr><td colspan="3">example:
{“idx”:”1”,”status”:”OFF”}</td></tr>
</table>

5.3 MQTT Topic(firmware version >= V2.15.869)

<table>
<tr><td>topic</td><td>type</td><td colspan="3">value</td></tr>
<tr><td rowspan="7">/dingtian/relaySN/in/control

example:
/dingtian/relay1868/in/control</td><td rowspan="7">subscribe</td><td>parameter</td><td>filed</td><td>data</td></tr>
<tr><td>type</td><td>command type</td><td>ON/OFF
DELAY
JOGGING</td></tr>
<tr><td>idx</td><td>relay index</td><td>1~32</td></tr>
<tr><td>status</td><td>relay status</td><td>ON,OFF</td></tr>
<tr><td>time</td><td>time for type</td><td>ON/OFF:0
DELAY:1~65535second
JOGGING:1~255*100ms</td></tr>
<tr><td>pass</td><td>password</td><td>0~9999</td></tr>
<tr><td colspan="3">example:
{"type":"ON/OFF",'idx':'1',"status":"ON","time":"0","pass":"0"}
{"type":"DELAY",'idx':'2',"status":"ON","time":"5","pass":"0"}
{"type":"JOGGING",'idx':'3',"status":"ON","time":"5","pass":"0"
}
{"type":"ON/OFF",'idx':'4',"status":"OFF","time":"0","pass":"0"}</td></tr>
<tr><td rowspan="4">/dingtian/relaySN/out/relayX
example:
/dingtian/relay1868/out/
relay1</td><td rowspan="4">publish</td><td>parameter</td><td>filed</td><td>data</td></tr>
<tr><td>idx</td><td>relay index</td><td>1~32</td></tr>
<tr><td>status</td><td>relay status</td><td>ON,OFF</td></tr>
<tr><td colspan="3">example:
{“idx”:”1”,”status”:”OFF”}</td></tr>
</table>

5.4 MQTT Topic(firmware version >= V2.17.xx)

ETH: firmware version >= V2.17.xx

WiFi: firmware version >= V1.0.xx

<table>
<tr><td>topic</td><td>type</td><td colspan="3">value</td></tr>
<tr><td rowspan="6">ETH
/dingtian/relaySN/in/control
WiFi
/dingtian/wrelaySN/in/control

example:
/dingtian/relay1868/in/control
/dingtian/wrelay1868/in/control</td><td rowspan="6">subscribe</td><td>parameter</td><td>filed</td><td>data</td></tr>
<tr><td>type</td><td>command type</td><td>ON/OFF
DELAY
JOGGING</td></tr>
<tr><td>idx</td><td>relay index</td><td>1~32</td></tr>
<tr><td>status</td><td>relay status</td><td>ON,OFF</td></tr>
<tr><td>time</td><td>time for type</td><td>ON/OFF:0
DELAY:1~65535second
JOGGING:1~255*100ms</td></tr>
<tr><td>pass</td><td>password</td><td>0~9999</td></tr>
<tr><td colspan="4">example:
{"type":"ON/OFF",'idx':'1',"status":"ON","time":"0","pass":"0"}
{"type":"DELAY",'idx':'2',"status":"ON","time":"5","pass":"0"}
{"type":"JOGGING",'idx':'3',"status":"ON","time":"5","pass":"0"}
{"type":"ON/OFF",'idx':'4',"status":"OFF","time":"0","pass":"0"}</td></tr>
<tr><td>ETH
/dingtian/relaySN/in/rX
WiFi
/dingtian/wrelaySN/in/rX

example:
/dingtian/relay1868/in/r1
/dingtian/relay1868/in/r2
/dingtian/wrelay1868/in/r1
/dingtian/wrelay1868/in/r2</td><td>subscribe</td><td colspan="3">X:1~32
value: ON,OFF</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td>ETH
/dingtian/relaySN/out/rX
WiFi
/dingtian/wrelaySN/out/rX

example:
/dingtian/relay1868/out/r1
/dingtian/relay1868/out/r2
/dingtian/wrelay1868/out/r1
/dingtian/wrelay1868/out/r2</td><td>publish</td><td colspan="3">X:1~32
value: ON,OFF</td></tr>
</table>

<table>
<tr><td>ETH
/dingtian/relaySN/out/iX
WiFi
/dingtian/wrelaySN/out/iX

example:
/dingtian/relay1868/out/i1
/dingtian/relay1868/out/i2
/dingtian/wrelay1868/out/i1
/dingtian/wrelay1868/out/i2</td><td>publish</td><td>X:1~32
value: ON,OFF</td></tr>
<tr><td>ETH
/dingtian/relaySN/out/relayX
WiFi
/dingtian/wrelaySN/out/relayX

example:
/dingtian/relay1868/out/relay1
/dingtian/relay1868/out/relay2
/dingtian/wrelay1868/out/relay1
/dingtian/wrelay1868/out/relay2</td><td>publish</td><td><table><tr><td>parameter</td><td>filed</td><td>data</td></tr><tr><td>idx</td><td>relay index</td><td>1~32</td></tr><tr><td>status</td><td>relay status</td><td>ON,OFF</td></tr></table>example:
{“idx”:”1”,”status”:”OFF”}</td></tr>
<tr><td>ETH
/dingtian/relaySN/out/inputX
WiFi
/dingtian/wrelaySN/out/inputX

example:
/dingtian/relay1868/out/input1
/dingtian/wrelay1868/out/input1</td><td>publish</td><td><table><tr><td>parameter</td><td>filed</td><td>data</td></tr><tr><td>idx</td><td>relay index</td><td>1~32</td></tr><tr><td>status</td><td>relay status</td><td>HIGH,LOW</td></tr></table>example:
{"idx":"1","status":"HIGH"}
{"idx":"1","status":"LOW"}</td></tr>
<tr><td>ETH
/dingtian/relaySN/out/ip
WiFi
/dingtian/wrelaySN/out/ip

example:
/dingtian/relay1868/out/ip
/dingtian/wrelay1868/out/ip</td><td>publish</td><td>example:
192.168.1.100</td></tr>
<tr><td>ETH
/dingtian/relaySN/out/sn
WiFi
/dingtian/wrelaySN/out/sn

example:
/dingtian/relay1868/out/sn</td><td>publish</td><td>example:
1868</td></tr>
</table>

/dingtian/relay1868/out/sn		
ETH /dingtian/relaySN/out/mac WiFi /dingtian/wrelaySN/out/mac example: /dingtian/relay1868/out/mac /dingtian/relay1868/out/mac	publish	example: bc:34:88:00:00:00
ETH /dingtian/relaySN/out/input_cnt WiFi /dingtian/wrelaySN/out/input_cnt example: /dingtian/relay1868/out/input_cnt /dingtian/wrelay1868/out/ input_cnt	publish	2,4,8,16,32
ETH /dingtian/relaySN/out/relay_cnt WiFi /dingtian/relaySN/out/relay_cnt example: /dingtian/relay1868/out/relay_cnt /dingtian/wrelay1868/out/relay_cnt	publish	2,4,8,16,32

5.5 MQTT LWT topic

ETH: firmware version >= V2.17.188

WiFi: firmware version >= V1.0.449

topic	type	value
ETH /dingtian/relaySN/out/lwt_availability WiFi /dingtian/wrelaySN/out/lwt_availability example /dingtian/relay1868/out/lwt_availability /dingtian/wrelay1868/out/ lwt_availability	publish	online,offline

6 Protocol:CoAP

Relay board as CoAP server, accept CoAP Client request.
Support relay on/off
Support relay jogging
Support relay delay
Support password verification

you need linux system to compile libcoap

6.1 Compile libcoap

git clone --recurse-submodules https://github.com/obgm/libcoap
./autogen.sh
./configure --disable-manpages --enable-examples --enable-tests
make

6.2 Get relay status

Relay Status(1:ON, 0:OFF)
./coap-client -m get coap://192.168.1.100/dingtian/r1
./coap-client -m get coap://192.168.1.100/dingtian/r2
./coap-client -m get coap://192.168.1.100/dingtian/r3
./coap-client -m get coap://192.168.1.100/dingtian/r4
./coap-client -m get coap://192.168.1.100/dingtian/r5
./coap-client -m get coap://192.168.1.100/dingtian/r6
./coap-client -m get coap://192.168.1.100/dingtian/r7
./coap-client -m get coap://192.168.1.100/dingtian/r8
…
./coap-client -m get coap://192.168.1.100/dingtian/r32

Input Status(1:High, 0:Low)
./coap-client -m get coap://192.168.1.100/dingtian/i1
./coap-client -m get coap://192.168.1.100/dingtian/i2
./coap-client -m get coap://192.168.1.100/dingtian/i3
./coap-client -m get coap://192.168.1.100/dingtian/i4
./coap-client -m get coap://192.168.1.100/dingtian/i5
./coap-client -m get coap://192.168.1.100/dingtian/i6
./coap-client -m get coap://192.168.1.100/dingtian/i7
./coap-client -m get coap://192.168.1.100/dingtian/i8
…
./coap-client -m get coap://192.168.1.100/dingtian/i32

6.3 Control relay(simple)

```
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r1     # relay1 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r1     # relay1 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r2     # relay2 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r2     # relay2 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r3     # relay3 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r3     # relay3 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r4     # relay4 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r4     # relay4 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r5     # relay5 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r5     # relay5 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r6     # relay6 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r6     # relay6 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r7     # relay7 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r7     # relay7 OFF
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r8     # relay8 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r8     # relay8 OFF
…
./coap-client -e "1" -m put coap://192.168.1.100/dingtian/r32    # relay32 ON
./coap-client -e "0" -m put coap://192.168.1.100/dingtian/r32    # relay32 OFF
```

6.4 Control relay

format:

status:type:time:password

parameter	filed	data	comment
status	relay status	0,1	
type	ON/OFF DELAY JOGGING		
time	time for type	ON/OFF:0 DELAY:1~65535second JOGGING:1~255*100ms	
password	password	0~9999	

example:

1:ON/OFF:0:4660

status:1

type:ON/OFF

time:0

password:4660

ON/OFF example:

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r1

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r2

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r3

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r4

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r5

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r6

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r7

./coap-client -e "1:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r8

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r1

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r2

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r3

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r4

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r5

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r6

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r7

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r8

…

./coap-client -e "0:ON/OFF:0:4660" -m put coap://192.168.1.100/dingtian/r32

DELAY example:

./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r1

./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r2
./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r3
./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r4
./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r5
./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r6
./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r7
./coap-client -e "1:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r8
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r1
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r2
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r3
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r4
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r5
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r6
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r7
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r8
…
./coap-client -e "0:DELAY:5:4660" -m put coap://192.168.1.100/dingtian/r32

JOGGING example:
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r1
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r2
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r3
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r4
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r5
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r6
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r7
./coap-client -e "1:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r8
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r1
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r2
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r3
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r4
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r5
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r6
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r7
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r8
…
./coap-client -e "0:JOGGING:5:4660" -m put coap://192.168.1.100/dingtian/r32