IOT Relay Programing Manual

V1.8

1 Product Overview

1.1 Overview

Support multiple channel relay, On/OFF/Jogging/Delay.
Support multiple interface RJ45/RS485/CAN/WIFI
Support HTTP GET CGI/UDP/TCP Server/TCP Client
10/100Mbps ethernet, Auto-MDIX,DHCP ip,Static IP
Local Button control(SelfLock/Jogging/Delay)
PC app config and control
WEB config and control
Support password.
Support Modbus-RTU/ASCII/TCP/UDP
Support Modbus-RTU Over TCP/UDP
Support Modbus-ASCII Over TCP/UDP
Support MQTT(Only Ethernet)
Support CoAP

Home Automation System Support:

Name	github
Domoticz	https://github.com/dtlzp/Domoticz-Dingtian-Relay-Plugin

SDK download address:
ftp://ftp.dingtian-tech.com/relay_sdk.zip

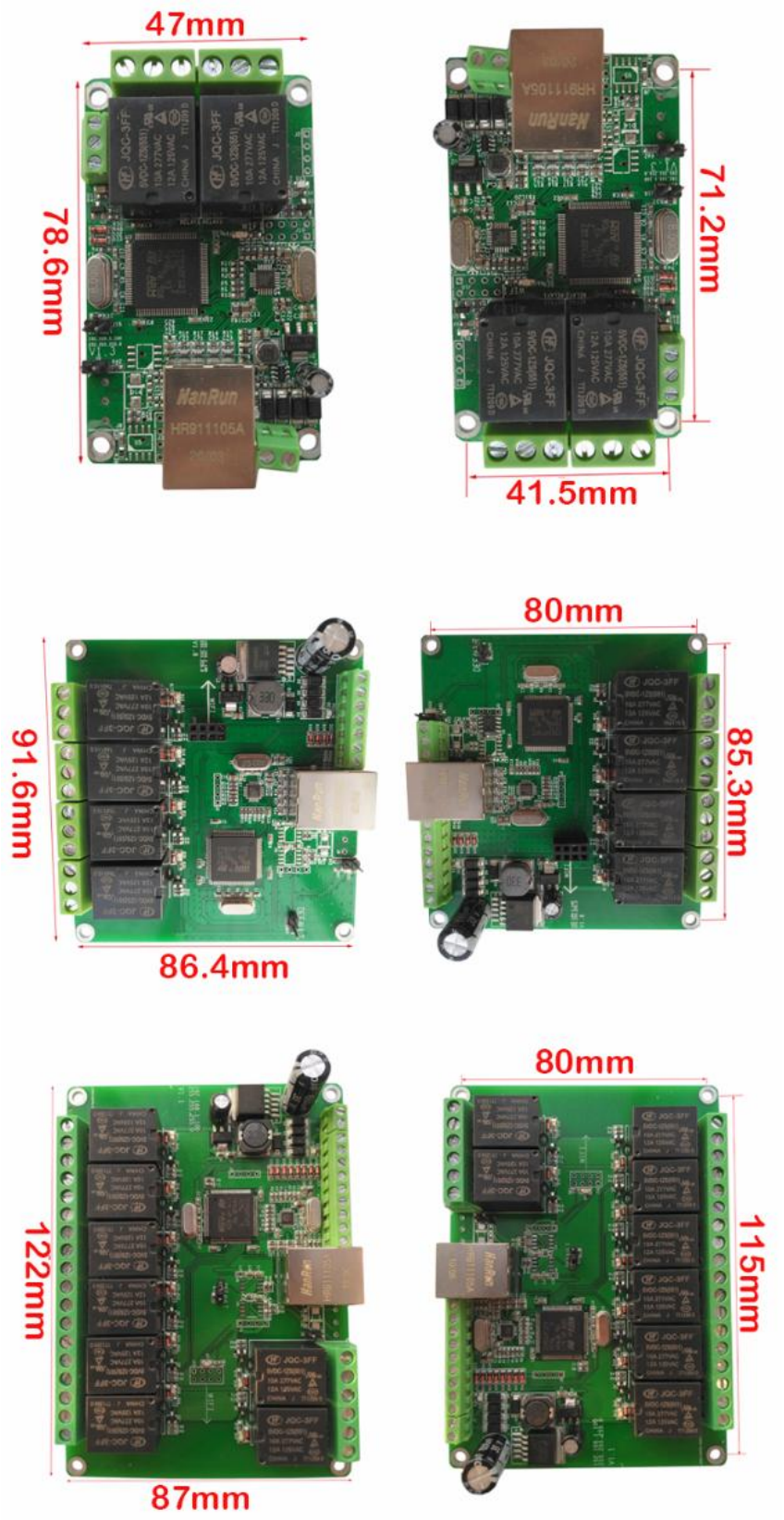
1.2 Technical Parameters

Network	Interface	RJ45/ RS485/CAN/WIFI
	Rate	100M/115200bps/125kbps/150Mbps
	Protocol	TCP server/client, UDP HTTP GET CGI, Modbus-RTU/ASCII/TCP/UDP Modbus-RTU Over TCP/UDP Modbus-ASCII Over TCP/UDP MQTT(only Ethernet) CoAP
Output	Relay Power	AC 250V/10A,DC 30V/10A
	Contacts	Normally Close Normally Open
	Delay	1~65535 seconds

	Jogging	Pull in 0.5 seconds, automatically release
Working environment	Operating temperature	0~+85°C
Power	Power Specifications	12V DC
	Current	500mA@12V DC At least 1A/12V adapter fill Voltage and current(Please satisfy)
	Power consumption	5W

2 Size

Hole size: 3.5mm



3 Interface Description

3.1 Indictor

wifi led	on: connect wireless route success off: can not connect wireless router
CH1-CH8 led	on: relay on off: relay off

3.2 Relay contact

Each set of relay outputs has three terminals: normally open contact, common terminal and normally closed contact. The contact capacity is AC 250V10A, DC 30V10A, and the output of controlling higher power requires external contactor.

• Normally open contact:

When the relay is released (or the module is powered off), the common terminal is disconnected from the normally open contact. After the suction is closed, the two contacts are closed.

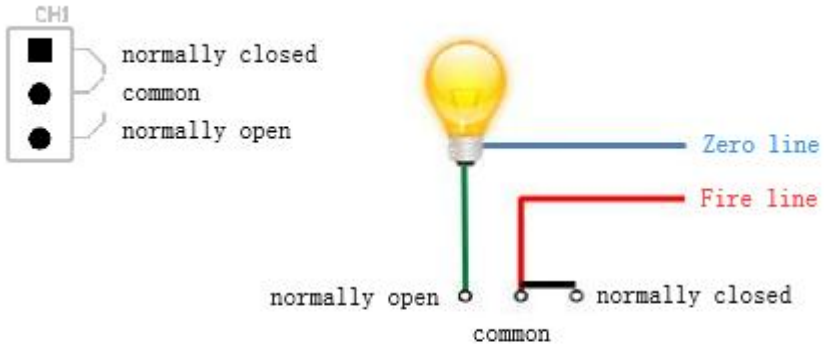
• Common:

Controlled power input

• Normally closed contact:

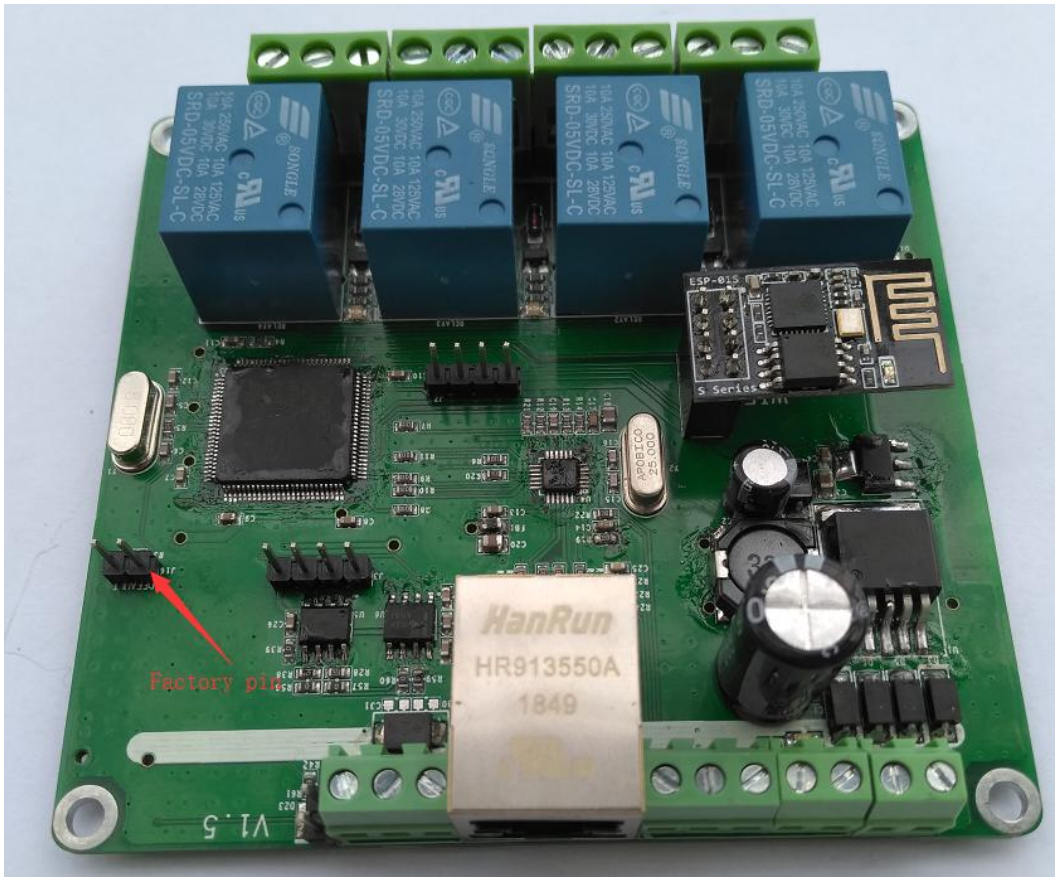
When the relay is released (or the module is powered down), the common and normally closed contacts are closed. After the pull-in, the two contacts are disconnected.

Connection example



3.3 Reset to factory

1 Short the 2 pin headers under the Default assembly with a jumper cap



2 Turn off the power of the network module, and then power on the module again.

3 Pull out the Default jumper cap

4 Tools

RelayTool	PC app for device
CGITool	PC test HTTP CGI API
relay.sh	unix/linux shell control relay

5 Protocol:Dingtian string

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

5.1 Query status command

command code	00(2 character)	return 8 character, Each character may be 0 or 1, representing a relay On or Off The state, such as the return value of 11000000, means that CH1 and CH2 are On, and the other channels are Off

Remarks

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

2 UDP mode does not support query instructions

5.2 Basic control command

CH1 On	11	The return value is the same as 5.1 Query status command
CH1 Off	21	
CH2 On	12	
CH2 Off	22	
CH3 On	13	
CH3 Off	23	
CH4 On	14	
CH4 Off	24	
CH5 On	15	
CH5 Off	25	
CH6 On	16	
CH6 Off	26	
CH7 On	17	
CH7 Off	27	
CH8 On	18	
CH8 Off	28	
All On	1X	
All Off	2X	

5.3 Delay command

The delay command consists of the basic command + ":" + delay seconds. The delay time range is 1-65535 seconds, which can be turned Off delay On or the delay is Off after On

E.g

status	Command code	result
CH1 is currently Off	11:30	CH1 On and Off automatically after 30

		seconds
CH2 is currently in On	22:30	CH2 Off, automatically On after 30 seconds
CH2 is currently Off	22:30	CH2 Off(no state change), automatically On after 30 seconds

5.4 Jogging command

The jogging command consists of the basic pull-in command + "*". The effect of the jogging is that the relay is automatically Off after 0.5 seconds of On

6 Protocol:Dingtian binary

Only support UDP
Support Different network segment communication
Mulitcast addr: 224.0.2.11

Support password

6.1 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

6.2 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	

6.2.1 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	Bit0~7 map to relay relay1~8 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

6.2.2 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	Bit0~7 map to relay relay1~8 Bit=1,relay need update
relay set	1	Bit0~7 map to relay relay1~8 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

6.2.3 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

6.2.4 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

6.2.5 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

7 Protocol:Dingtian-binary config

Only support UDP
Support Different network segment communication
Mulitcast addr: 224.0.2.11

7.1 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

7.2 config struct

```
#define WEB_USER_LEN            (16)
#define WEB_PASSWORD_LEN        (16)
#define MAC_LEN                 (6)
struct rs232_conf
{
    u32 baudrate;/* 1200/2400/4800/9600/19200/38400/57600/115200bps */
    u32 databits;/* 8:8bit1 */
    u32 stopbits;/* 1:1bits 2:2bits */
    u32 parity;/* 0:none 1:odd 2:even    */
};
struct can_id
{
    u32 id          :29;
    u32 res         :1;
    u32 remote      :1;
    u32 extid       :1;
};/* 4bytes */
enum CAN_BAUD
{
    CAN_BAUD_START,
    CAN_BAUD_5K = CAN_BAUD_START,
    CAN_BAUD_10K,
    CAN_BAUD_20K,
    CAN_BAUD_25K,
    CAN_BAUD_50K,
    CAN_BAUD_100K,
    CAN_BAUD_125K,
    CAN_BAUD_200K,
    CAN_BAUD_250K,
    CAN_BAUD_500K,
    CAN_BAUD_800K,
    CAN_BAUD_888K,
```

```
	CAN_BAUD_1000K,
};
struct can_conf
{
	u32 baudrate;/* enum CAN_BAUD */
	struct can_id id;
};
enum
{
	DNS_START,
	DNS_UDP = DNS_START,
	DNS_TCP,
	DNS_MQTT,

	DNS_END,
};
#define SERVER_URL_MAX    (32)
struct server_port
{
	char server[SERVER_URL_MAX];
	u32 port;
};
#define RELAY_MAX            (32)

#define ROUTER_SSID_LEN        (64)
#define ROUTER_PWD_LEN          (64)

#define KEEP_ALIVE_MAX          (120)
#define KEEP_ALIVE_DEF          (30)
#define KEEP_ALIVE_CLOSE        (0)

#define JOGGING_100MS_MAX          (255)
#define JOGGING_100MS_DEF          (5)
#define JOGGING_100MS_MIN          (1)

//#pragma pack(1)
struct param
{
	u32 sn;
	u32 sw_ver;
	u32 hw_ver;
	u32 model;

	char web_passwd[WEB_PASSWORD_LEN];
```

```
    u32 ip;
    u32 netmask;
    u32 gateway;
    u32 dns;
    b8 dhcp;
    u8 res;
    u8 mac[MAC_LEN];

    struct rs232_conf rs232;

    struct can_conf can;

    struct server_port sp[DNS_END];

    u16 relay_password;
    u8 power_failure_recovery;/* power failure recover relay status */
    u8 relay_cnt;
    u32 relay_status;/* 1bit per relay */
    u8 key_type[RELAY_MAX];

    char router_ssid[ROUTER_SSID_LEN];
    char router_pwd[ROUTER_PWD_LEN];
    u8 keep_alive_second;/* 1~120 second */
    u8 jogging_100ms;/* jogging 100ms count */
    u8 res2[2];
};/* 360 bytes */
```

7.3 command

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

notice:
multicast command reply to 224.0.2.11 forever

format

filed	bytes	comment
command	1	3: read info+config 4: write config 5: multicast read info+config 6: multicast write config
result(xor 0xAA)	1	pc->device: 0 xor 0xAA

		device->pc: result xor 0xAA result=0 success result=other fail
command data	xx	

7.3.1 read config

pc send

filed	bytes	comment
command	1	3: read info+config or 5: multicast read info+config
result(xor 0xAA)	1	0 xor 0xAA=0xAA

device reply

filed	bytes	comment
command	1	0x03
result(xor 0xAA)	1	0 xor 0xAA=0xAA
config	360	struct param

7.3.2 write config

pc send

filed	bytes	comment
command	1	3: write config or 5: multicast write config
result(xor 0xAA)	1	0 xor 0xAA=0xAA
config	360	struct param

device reply

filed	bytes	comment
command	1	0x03
result(xor 0xAA)	1	0 xor 0xAA=0xAA

8 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

8.1 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	
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

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

8.2 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	0~8	
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	0~7	0:relay 1 1:relay 2 ... 7:relay 8
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

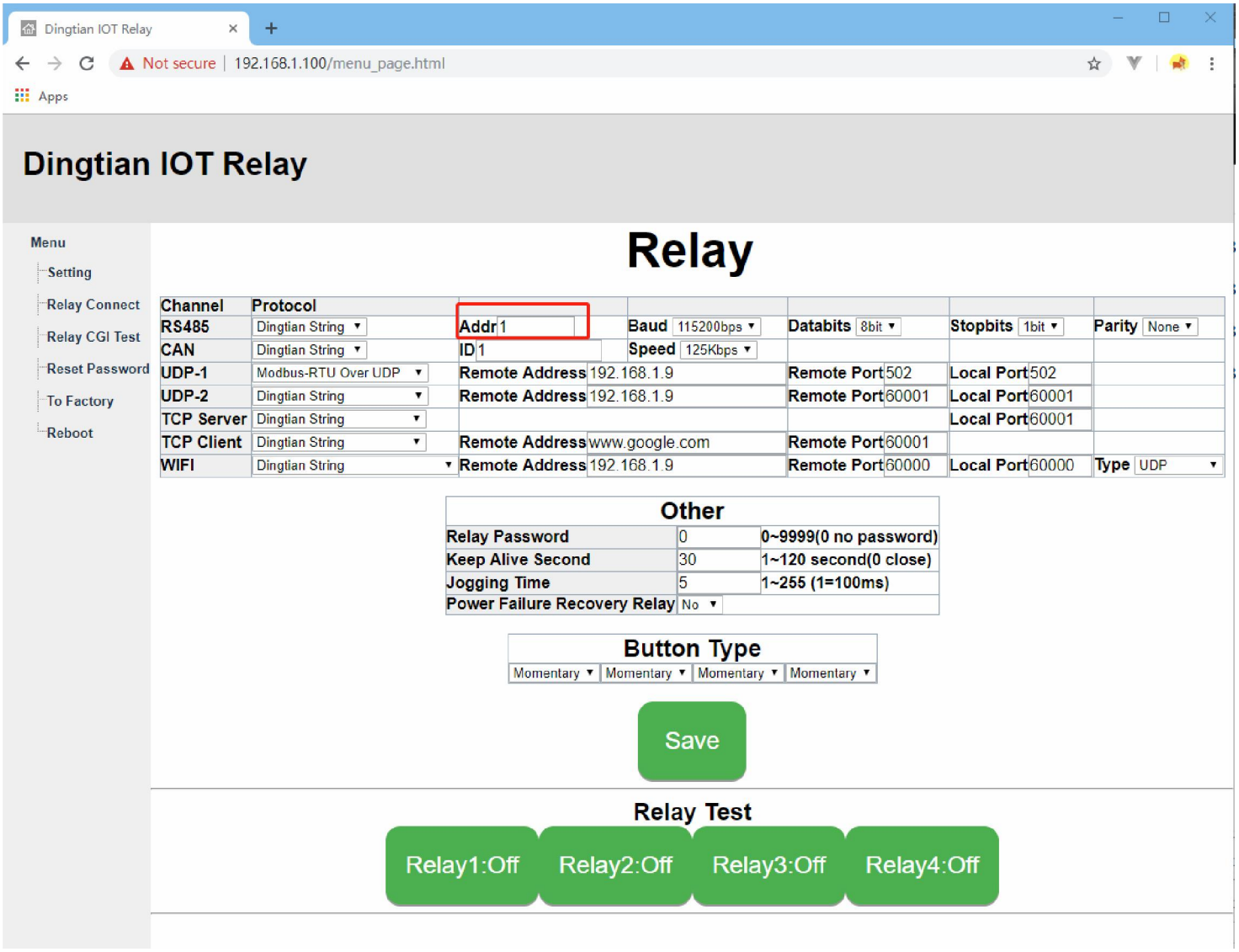
&0&2&1&0&10& # ok, type 2 delay,relay 1 off,time 10 second

9 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



9.1 Registers

Register	Name	0x03/0x06/0x10	Value
0x0000	Relay Count	0x03	2/4/8
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	1:Write ON/OFF 2:Write with delay 3:Write with Jogging
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 0~7 Type:Write with Jogging(3) bit0: bit=1 ON,bit=0 OFF bit1~7:relay index 0~7
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)

Notice:

1、0x0003~6 must is block， must written at the same time.

9.2 example

Notice:

All example is 4 channel relay

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

9.2.1.1 0x03:Read holding register

Read all Relay Status

Send:

01 03 0000 0002 C40B

Recv:

01 03 04 0004 0000 BBF2

9.2.1.2 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

9.2.1.3 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

2、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

9.3 Modbus-TCP/UDP

9.3.1.1 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

9.3.1.2 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

9.3.1.3 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

2、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

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

9.4.1.1 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

9.4.1.2 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

9.4.1.3 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

2、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

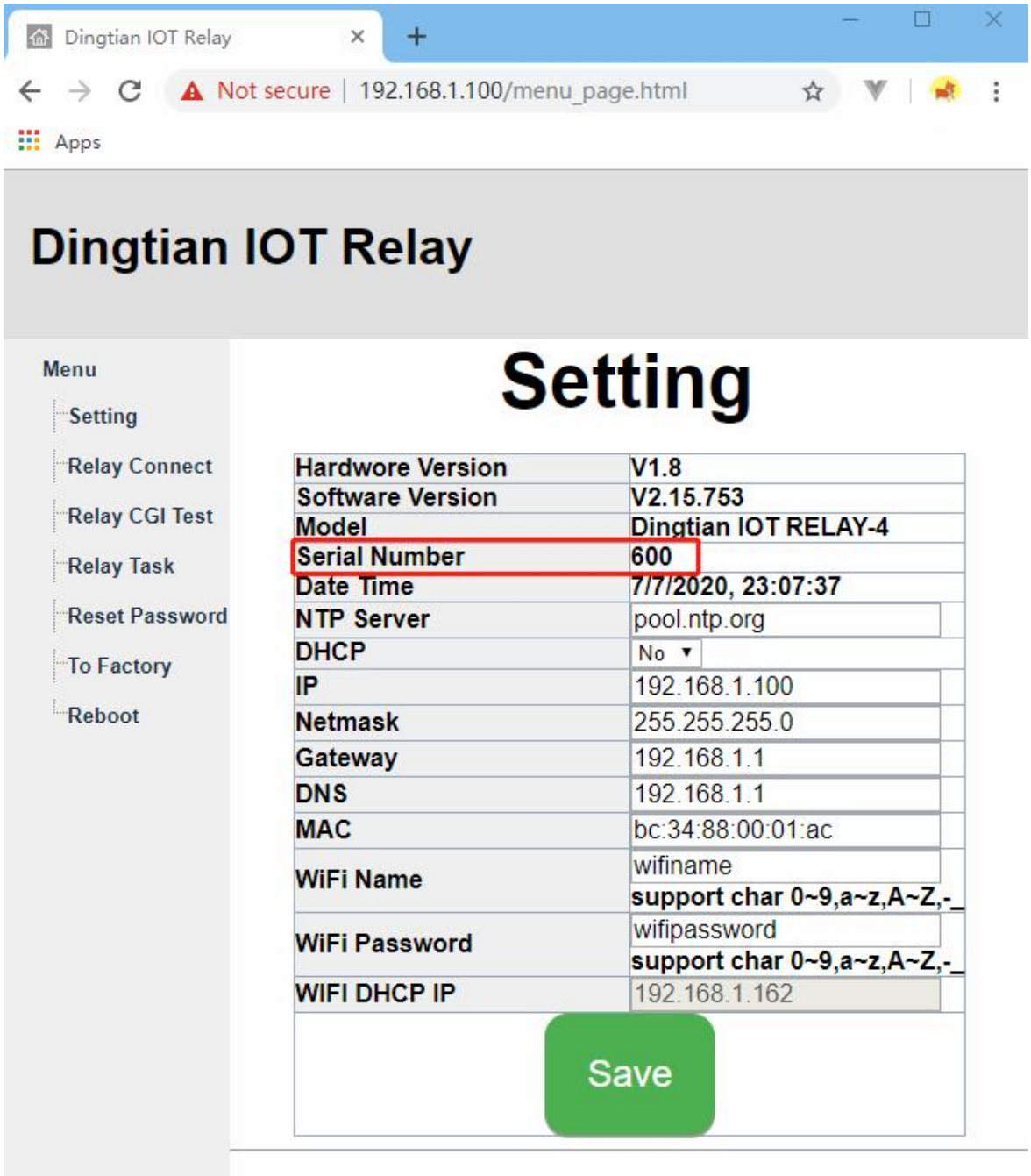
HEX 3A 3031 3130 30303033 30303034 4237 0D0A

10 Protocol:MQTT

Relay board as MQTT client,communcation with broker..

Support relay on/off
Support relay jogging
Support relay delay
Support password verification

relay board client id format: dingtian-relay+SN
example:
below relay board “Serial Number” is 600
so MQTT client id is:dingtian-relay600



10.1 Topic to subscribe

Relay board subscribe topic: /dingtian/relay/in/control

format:

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

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"}

10.2 Topic to publish

Relay board publish topic:

/dingtian/relay/out/relay1

/dingtian/relay/out/relay2

/dingtian/relay/out/relay3

/dingtian/relay/out/relay4

/dingtian/relay/out/relay5

/dingtian/relay/out/relay6

/dingtian/relay/out/relay7

/dingtian/relay/out/relay8

format

parameter	filed	data	comment
idx	relay index	0	
status	relay status	ON,OFF	

example:

{“idx”:”1”,”status”:”OFF”}

11 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

11.1 Compile libcoap

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

11.1 Get relay status

./coap-client -m get coap://192.168.1.100/dingtian-relay?r1
./coap-client -m get coap://192.168.1.100/dingtian-relay?r2
./coap-client -m get coap://192.168.1.100/dingtian-relay?r3
./coap-client -m get coap://192.168.1.100/dingtian-relay?r4
./coap-client -m get coap://192.168.1.100/dingtian-relay?r5
./coap-client -m get coap://192.168.1.100/dingtian-relay?r6
./coap-client -m get coap://192.168.1.100/dingtian-relay?r7
./coap-client -m get coap://192.168.1.100/dingtian-relay?r8

11.2 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

11.3 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

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

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