KC868-A series board protocol – HTTP get command

Note: This protocol document use for KinCony smart controller:

KC868-AM ASR A2 A4 A4S A6 A8 A8M A8S A16 A16S E16S A32 A32M A64 A128 AG AK AI AIO AP

Different board will have different channel of digital output, digital input , ADC, DAC, so the protocol is same , just according to the hardware resource to set channel number.

HTTP command format:

http://ip/ctrl.cgi?secret=aaa&cmd=bbb&id=ccc&value=ddd

secret: set "secret code" in board setting webpage
cmd: set command type
id: set device input or output ID
value: set value of device

The complete command consists of these four parameters.

Fox example: KC868-A64 board ip=192.168.1.200   secret=abcd

Will have this command:

http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=get_inputs&id=0&value=0

---

1. Read all digital input state

parameter	value
secret	abcd
cmd	get_inputs
id	0
value	0

send:
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=get_inputs&id=0&value=0

feedback:
```
{
	"inputs":  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
	"status":  "success",
    "code":   0
}
```

Feedback JSON format data， "inputs" means: 1-64 channel digital input state: 1 is trigger , 0 is not trigger.

For example: if digital input-1 is trigger will feedback:

```
{
    "inputs":  [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    "status":  "success",
    "code":   0
}
```

2. Read all digital output state

parameter	value
secret	abcd
cmd	get_outputs
id	0
value	0

send：

http://192.168.1.200**/ctrl.cgi?secret=**abcd**&cmd=get_outputs&id=0&value=0**

feedback：

```
{
    "outputs":     [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    "status":  "success",
    "code":   0
}
```

Feedback JSON format data， "outputs" means: 1-64 channel digital output state: 1 is ON , 0 is OFF.

For example: if digital output-1 is ON will feedback:

```
{
    "outputs":     [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    "status":  "success",
    "code":   0
}
```

3. Read all ADC (analog input) state

parameter	value
secret	abcd
cmd	get_adcs
id	0
value	0

send:
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=get_adcs&id=0&value=0

feedback:
```
{
	"adcs":	[31, 48, 50, 0],
	"status":	"success",
	"code":	0
}
```
Feedback JSON format data, "adcs" means: 1-4 channel ADC original acquisition value. Range: 0-4095

4. Read all DAC (analog output) state

parameter	value
secret	abcd
cmd	get_dacs
id	0
value	0

send:
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=get_dacs&id=0&value=0

feedback:
```
{
	"dacs":	[31, 48],
	"status":	"success",
	"code":	0
}
```
Feedback JSON format data, "dacs" means: 1-4 channel DAC value. Range: 0-255 for output DC 0-10v

5. Set ON/OFF one channel of digital output

parameter	value
secret	abcd
cmd	set_output
id	output channel number -- KC868-A64 is (1-64)
value	1: ON 0: OFF

send:
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=set_output&id=1&value=1
this means: turn ON output-1
feedback:
```
{
	"id": 1,
	"value":	1,
	"status":	"success",
    "code":  0
}
```

Feedback JSON format data， ”success” is control OK.

6. Set ON/OFF all channels of digital output

parameter	value
secret	abcd
cmd	set_outputs
id	0
value	HEX data for all output, KC868-A64 is (1000000000000000)

send：

http://192.168.1.200**/ctrl.cgi?secret=**abcd**&cmd=**set_outputs**&id=0&value=**0000000000000010

feedback：

```
{
    "outputs":    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
    "status":  "success",
    "code":  0
}
```

The parameter value is a hexadecimal string, with a total of 8 bytes. The example is 0000000000000010. From left to right, it is D7, D6, D5, D4, D3, D2, D1, D0. D0 represents channels 1-8, and D1 represents Channels 9-16. D0, from low binary bit to high binary bit means: bit0: channel1, bit1:channel2……bit7:channel8, so D0=0x10, converted into binary is 00010000, which means turn ON output-5, others 0 means all channels turn OFF.

Feedback JSON format data， ”success” is control OK. Feedback all state after output updated.

For example value=8070605040302010

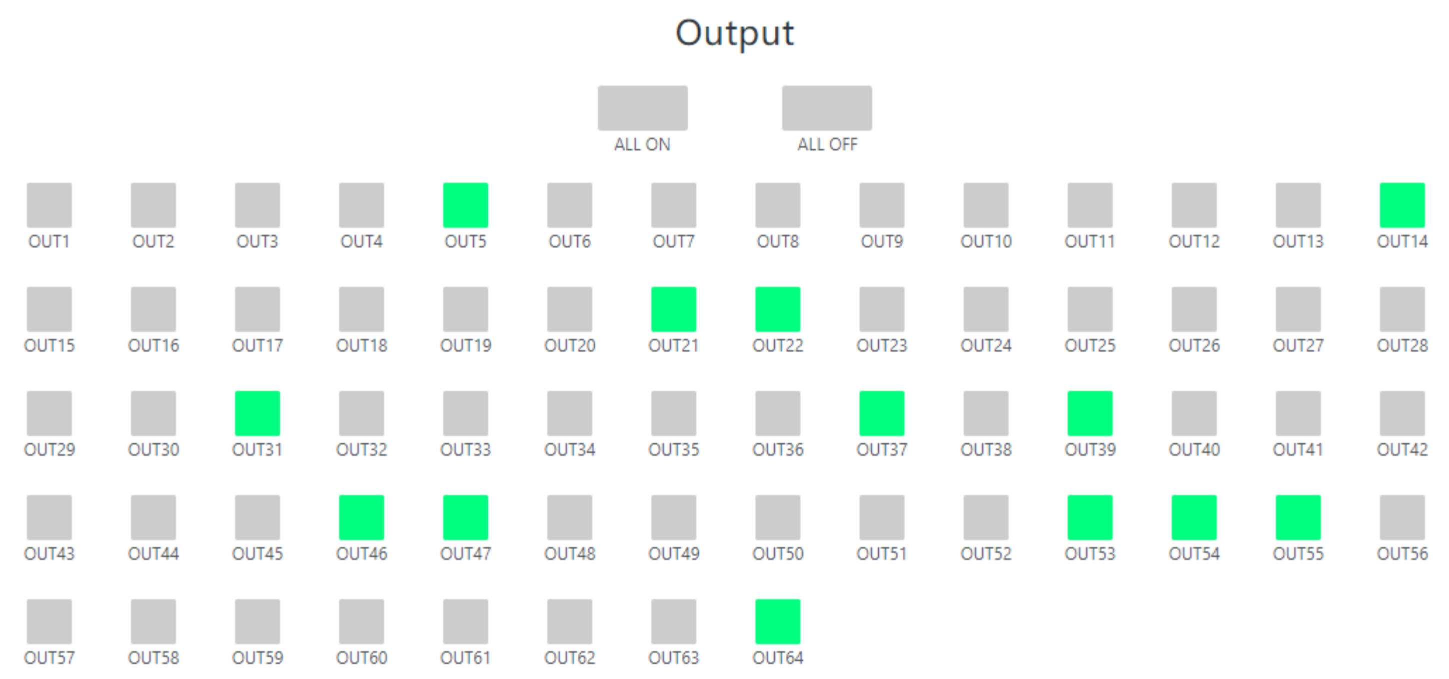
send：

http://192.168.1.200**/ctrl.cgi?secret=**abcd**&cmd=**set_outputs**&id=0&value=**8070605040302010

feedback：

```
{
    "outputs":    [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    "status":  "success",
    "code":  0
}
```

D7=0x80=(10000000)b	output (64-57)	output64:ON others: OFF
D6=0x70=(01110000)b	output (56-49)	output53,54,55:ON others: OFF
D5=0x60=(01100000)b	output (48-41)	output46,47:ON others: OFF
D4=0x50=(01010000)b	output (40-33)	output37,39:ON others: OFF
D3=0x40=(01000000)b	output (32-25)	output31:ON others: OFF
D2=0x30=(00110000)b	output (24-17)	output21,22:ON others: OFF
D1=0x20=(00100000)b	output (16-9)	output14:ON others: OFF
D0=0x10=(00010000)b	output (8-1)	output5:ON others: OFF



7. Set DAC

parameter	value
secret	abcd
cmd	set_dac
id	1
value	0-255

send:
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=set_dac&id=1&value=248

feedback:

```
{
        "id": 1,
        "value":   248,
        "status":  "success",
        "code":    0
}
```

Feedback JSON format data, "success" is control OK.

8. Read board all data

parameter	value
secret	abcd
cmd	get_all_datas
id	0
value	0

send:
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=get_all_datas&id=0&value=0

feedback:
```
{
        "inputs":  [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        "outputs":      [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
        "adcs":    [68, 134, 75, 0],
        "dacs":    [61, 34],
        "status":  "success",
        "code":    0
}
```

Feedback JSON format data, feedback all channels of digital input state, digital output state, ADC state and DAC state.

9. Send one IR signal have learned

parameter	value
secret	abcd
cmd	run_ir
id	IR id have learned (1-32)
value	0

send:
http://192.168.1.200/ctrl.cgi?cmd=run_ir&id=1&value=0&secret=abcd

this means: send IR signal that id=1
feedback:
{ "status": "success", "code": 0 }

10. Send one RF signal have learned

parameter	value
secret	abcd
cmd	run_rf
id	RF id have learned (1-32)
value	0

send:
http://192.168.1.200/ctrl.cgi?cmd=run_rf&id=1&value=0&secret=abcd

this means: send RF signal that id=1
feedback:
{ "status": "success", "code": 0 }

11. Set beep output

parameter	value
secret	abcd
cmd	set_beep
id	1
value	1: ON 0: OFF

send：
http://192.168.1.200/ctrl.cgi?secret=abcd&cmd=set_beep&id=1&value=1
this means: turn ON beep
feedback：
{ "status": "success", "code": 0 }

12. Read temperature & humidity sensor data
send:
http://192.168.1.200/ctrl.cgi?cmd=get_sensors&id=0&value=0&secret=abcd

feedback:

```
{
"sensors": [
{
"id": 1,
"type": 1,
"unit": 1,
  "temperature": 77.9000015258789,
"humity": -100
},
  { "id": 2,
"type": 0,
"unit": 0,
"temperature": -100,
"humity": -100
}
],
"status": "success",
"code": 0
}
```

Returns an array of sensors

Id: sensor ID
type: 0: disabled   1: ds18b20 2: DHT11   3: DHT22
unit   0: ℃   1: ℉
temperature: it’s float format data for temperature sensor
humity: data of humidity   -100: invalid