

질의 응답

ITS API는 API Console을 통해 관리되며 프로그램(Code)이나 코멘드라인인터프리터(CLI) 또는 URI로 테스트 및 제어가 가능하다.

명령문 구조

명령문은 배열(Array[])과 딕셔너리(Dictionary{})로 구성되는 JSON 형식이다.

- { "host":""," "port":""," "data":[{"Key:Value }] }
- Ex: [{"Key":{" ... }, "server":{" "host":""," "port":""," }, "keyCode":"sha256-hex", "debug":true}]
- Key
 - gpio, audio, system, camera, alarm, custom, server, keyCode, debug
- Value
 - Dictionary - 다음의 QnA 참조

GPIO - 테이블

Sensor	S01	S02	S03	S04	S05	S06	S07	S08
ID	io01	io02	io03	io04	io05	io06	io07	io08
Relay	R01	R02	R03	R04				
ID	io09	io10	io11	io12				

Q: GPIO 명령 - 기본

- status
 - '0:Off', '1:On', '2:Toggle', '3:Status', '7:Status Power', '8:Status Relay and Sensor', '9:Status All'
- id -> Table 참조
- hold (float)
- count (int)
- interval (float)

Q: Ex: 이벤트 발생시 릴레이 Off

- {"host":""," "port":""," "data":[{"gpio":{"status":"0","id":"io09","hold":"0"}}]}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"0","id":"io09","hold":"0"}}]

Q: Ex: 이벤트 발생시 릴레이 On

- {"host":""," "port":""," "data":[{"gpio":{"status":"1","id":"io09","hold":"0"}}]}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"1","id":"io09","hold":"0"}}]

Q: Ex: 이벤트 발생시 릴레이 Toggle

- {"host":"","port":"","data":[{"gpio":{"status":"2","id":"io09","hold":"0"}}]}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"io09","hold":"0"}}]

Q: 홀드(Sec) 기능 [hold]

- 홀드 값이 0이면 상태를 유지하며 0 이상이면 그 시간 이후 반전
- 홀드시간은 초단위로 0.1과 같은 무리수도 가능

Q: Ex: 이벤트 발생시 릴레이 반전, n초 후 복귀

- {"host":"","port":"","data":[{"gpio":{"status":"0","id":"io09","hold":"1","debug":false}}]}
- {"host":"","port":"","data":[{"gpio":{"status":"1","id":"io09","hold":"1","debug":false}}]}
- {"host":"","port":"","data":[{"gpio":{"status":"2","id":"io09","hold":"1","debug":false}}]}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"0","id":"io09","hold":"1","debug":true}}]
 - Return Value

```
{
  "category": "gpio",
  "ip": "192.168.0.80",
  "status": "0",
  "response": {
    "io09_desc": 0
  }
}
```

- http://my_ip_address/api.php?api=[{"gpio":{"status":"1","id":"io09","hold":"1","debug":true}}]
 - Return Value

```
{
  "category": "gpio",
  "ip": "192.168.0.80",
  "status": "1",
  "response": {
    "io09_desc": 1
  }
}
```

- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"io09","hold":"1","debug":true}}]

Q: GPIO 명령 - 고급명령

Q: 실행 결과 반환(Debug Mode) 기능 [debug]

- 요청 명령의 실행 결과의 반환 - "debug":true
- Console Log나 브라우저 또는 프로그램을 통해 반환값 전송
- 복합명령 요청시 콤마구분 없이 관련정보를 연속적으로 반환한다.

Q: Ex: 이벤트 발생시 릴레이 반전, n초 후 복귀, 결과확인(debug)

- {"host":"","port":"","data":[{"gpio":{"status":"2","id":"io09","hold":"1"},"debug":true]}}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"io09","hold":"1"},"debug":true}]

Q: Count 기능 - [count]

- 센서 관리 기능 - 기준주기: 횟수
- 카운터 설정값 이상일때 이벤트 발생
- 카운터 설정값을 주기로 반복됨

Q: Ex: Count

- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"pw01","hold":"0.1","count":"3"},"debug":true}]
- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"pw01","hold":"0.1","count":"7"},"debug":true}]

Q: Interval 기능 [interval]

- 센서 관리 기능 - 기준주기: 시간
- 최초 알람 후 인터벌 시간내 이벤트는 무시됨
- 수신측 이벤트 처리속도에 적절한 조절

Q: Ex: Interval

- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"pw01","hold":"0.1","interval":"1"},"debug":true}]
 - 설명: 최초의 이벤트 발생후 interval(1초)이내에 발생하는 이벤트는 무시 한다.
- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"pw01","hold":"0.1","interval":"4"},"debug":true}]
 - 설명: 최초의 이벤트 발생시 알람을 전송하고 interval(4초)이내에 발생하는 이벤트는 무시 한다.

Q: Count 와 Interval 조합 기능

- 센서 관리 기능 - 기준주기: 횟수 와 시간
- 인터벌 시간내 카운터 설정값 이상 감지시 알람발생
- 이때 알람은 종료시간 까지 연속적으로 발생한다.
- 인터벌가 종료 되면 카운터도 초기화됨

Q: Ex: Count & Interval

- `http://my_ip_address/api.php?api=[{"gpio": {"status": "2", "id": "pw01", "hold": "0.1", "count": "3", "interval": "1"}, "debug": true}]`
 ◦ 설명: 인터벌 시간 내에 이벤트발생 횟수가 카운트수를 넘는시점 부터 알람이 발생한다.
- `http://my_ip_address/api.php?api=[{"gpio": {"status": "2", "id": "pw01", "hold": "0.1", "count": "7", "interval": "2"}, "debug": true}]`
 ◦ 설명: interval(2초)이내에 count(7번)이상의 이벤트가 발생하면 알람을 발생한다.

Q: 복합명령(Complex Command)

- 하나 이상의 명령을 동시에 요청 한다.(예: 요청시 n초후 릴레이 작동)
- 단일명령을 콤마로 구분 및 배열, 조합수는 무제한
- `http://my_ip_address/api.php?api=[{"system":{"command":"sleep","value":"1.0"},"debug":true},{"gpio":{"status":"2","id":"io09","hold":"0"},"debug":true}]`

Q: Ex: 복합명령과 조합 가능한 Sleep 명령 : n초 대기

- `{"host":"","port":"","data":[{"system":{"command":"sleep","value":"1.0"},"debug":true}]}`
- `http://my_ip_address/api.php?api=[{"system":{"command":"sleep","value":"1.0"},"debug":true}]`
 ◦ Return Value

```
{
  "category": "system",
  "ip": "192.168.0.80",
  "command": "sleep",
  "msg": "sleep 1.0sec"
}
```

Q: Ex: 이벤트 발생시 릴레이 n초 대기 후 Toggle

- `{"host":"","port":"","data":[{"system":{"command":"sleep","value":"1.0"}},{"gpio":{"status":"2","id":"io09","hold":"0.1"}}]}`
- `http://my_ip_address/api.php?api=[{"system":{"command":"sleep","value":"1.0"}},{"gpio":{"status":"2","id":"io09","hold":"0.1"}}]`

Q: Ex: 이벤트 발생시 릴레이(R01), 릴레이(R02) 동시 반전, n초 후 복귀

- `{"host":"","port":"","data":[{"gpio":{"status":"2","id":"io09","hold":"1"}},{"gpio":{"status":"2","id":"io10","hold":"1"}}]}`
- `http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"io09","hold":"1"}},{"gpio":{"status":"2","id":"io10","hold":"1"}}]`

Q: Ex: 이벤트 발생시 릴레이(R01) 반전, n초 후 릴레이(R02) 반전

- {"host":"","port":"","data":[{"gpio":{"status":"2","id":"io09","hold":"0.1"}},{"system":{"command":"sleep","value":"1.0"}},{"gpio":{"status":"2","id":"io10","hold":"0.1"}}]}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"io09","hold":"0.1"},"debug":true},{"system":{"command":"sleep","value":"1.0"}},{"gpio":{"status":"2","id":"io10","hold":"0.1"},"debug":true}]
 - Return Value
 - {"category": "gpio", "ip": "192.168.0.80", "status": "2", "response": {"io09_desc": 0}}
 - {"category": "gpio", "ip": "192.168.0.80", "status": "2", "response": {"io10_desc": 1}}

Q: Ex: 단일 이벤트 릴레이(R01, R0, R03, R04) 전체 반전

- {"host":"","port":"","data":[{"gpio":{"status":"2","id":"io09","hold":"0.2"}},{"gpio":{"status":"2","id":"io10","hold":"0.4"}},{"gpio":{"status":"2","id":"io11","hold":"0.6"}},{"gpio":{"status":"2","id":"io12","hold":"0.8"}}]}
- http://my_ip_address/api.php?api=[{"gpio":{"status":"2","id":"io09","hold":"0.2"}},{"gpio":{"status":"2","id":"io10","hold":"0.4"}},{"gpio":{"status":"2","id":"io11","hold":"0.6"}},{"gpio":{"status":"2","id":"io12","hold":"0.8"}}]}

Q: System 명령

- help, sleep, set_name, get_name, set_time, get_time, stop_audio, list_audio, health_check, disable_sensor, enable_sensor, trigger_sensor, restart, reboot

Q: Ex: System

- http://my_ip_address/api.php?api=[{"system":{"command":"help","value":"","debug":true}}] - 예약
 - 도움말을 출력한다.
- http://my_ip_address/api.php?api=[{"system":{"command":"sleep","value":"1"},"debug":true}]
 - 복합명령 시 [sleep] 시간대기후 다음명령 수행
- http://my_ip_address/api.php?api=[{"system":{"command":"set_name","value":"타이틀명"},"debug":true}]
 - 타이틀명 설정.
- http://my_ip_address/api.php?api=[{"system":{"command":"get_name","value":"","debug":true}}]
 - ITS API 타이틀명 반환
 - Return Value

```
{
  "category": "system",
  "ip": "192.168.0.80",
  "command": "get_name",
  "msg": "Location"
}
```

- `http://my_ip_address/api.php?api=[{"system":{"command":"set_time","value":"2021-10-18 10:12:40"},"debug":true}]`
 - ITS API 시간 설정
 - Return Value

```
{
  "category": "system",
  "ip": "my_ip_address",
  "command": "set_time",
  "msg": "Success set_time Mon 18 Oct 10:12:40 KST 2021\n"
}
```

- `http://my_ip_address/api.php?api=[{"system":{"command":"get_time","value":""},"debug":true}]`
 - ITS API 시간 반환
- `http://my_ip_address/api.php?api=[{"system":{"command":"stop_audio","value":""},"debug":true}]`
 - 진행되는 방송 강제 중단
 - 방송중 경보가 발생된 경우 대비
 - Return Value

```
{
  "category": "system",
  "ip": "192.168.0.80",
  "command": "stop_audio",
  "msg": "Success stop_audio"
}
```

- `http://my_ip_address/api.php?api=[{"system":{"command":"list_audio","value":""},"debug":true}]`
 - 내부음원 목록 반환
 - Return Value

```
{
  "category": "system",
  "ip": "192.168.0.80",
  "command": "stop_audio",
  "msg": [
    "Air_Horn.mp3",
    "Fire_Truck.mp3",
    "Industrial.mp3",
    "Siren.mp3",
    "Smoke.mp3",
    "Whistle.mp3"
  ]
}
```

- `http://my_ip_address/api.php?api=[{"system":{"command":"health_check","value":""},"debug":true}]`

- ITS API 시스템 정보 반환
- Return Value

```
{
  "category": "system",
  "ip": "192.168.0.80",
  "command": "health_check",
  "msg": {
    "memUseKb": {
      "total": "924.2",
      "free": "365.9"
    },
    "diskGb": {
      "avail": "11G",
      "pcent": "22%",
      "used": "3.0G",
      "size": "15G"
    },
    "fixed": {
      "run": "API",
      "deviceModel": "Raspberry Pi 3 Model B Rev 1.2",
      "license": "e75a42f0a1d ... 64 Byte",
      "execTime": "0:00:00.280148",
      "liveTime": "311648.32",
      "noLicense": 2592000,
      "ipAddr": "192.168.0.80",
      "dateTime": "2021-10-21 21:11:31.762376",
      "diskSize": "15G",
      "serialKey": "000000008a845096",
      "systemTitle": "ECOS",
      "licenseStatus": "Approved",
      "ioBoard": "ITS STD",
      "lastStart": "2021-10-18 05:16:54"
    },
    "cpuPcent": {
      "idle": "93.7",
      "system": "5.1",
      "user": "1.3"
    },
    "cpuTemp": 48.312
  }
}
```

- [http://my_ip_address/api.php?api=\[{"system": {"command": "enable_sensor", "value": "io01"}, "debug": true}\]](http://my_ip_address/api.php?api=[{)
 - 센서 이벤트 발생에 따른 명령어 수행을 허용한다.
- [http://my_ip_address/api.php?api=\[{"system": {"command": "disable_sensor", "value": "io01"}, "debug": true}\]](http://my_ip_address/api.php?api=[{)
 - 센서 이벤트 발생에 따른 명령어 수행을 제한한다.

- `http://my_ip_address/api.php?api=[{"system": {"command": "trigger_sensor", "value": "io01"}, "debug": true}]`
 - Soft Sensing (Dry Contact)이 아닌 프로그램 이벤트)
 - 주의: 무한루프 오류
 - Return Value

```
{
  "category": "system",
  "ip": "192.168.0.80",
  "command": "trigger_sensor",
  "msg": "trigger io02"
}
```

- `http://my_ip_address/api.php?api=[{"system":{"command":"reboot","value":""},"debug":true}]` - 예약

Q: Audio 명령

- 방송중 연속적인 재생 요청은 무시된다.
- 제공되는 오디오 플레이어는 omxplayer와 mplayer이 있다.
- 플레이어 특성에 따라 볼륨레벨이 다를 수 있다.
- omxplayer 선택시 반복(loop)기능은 무시된다.
- mplayer는 omxplayer보다 재생 직전 초기화 시간이 길다.
- source: Internal(No/Full Path), External(URI)
- volume(%): 0 - 100
- loop: 0 ~ Int.
- Manager - `http://my_ip_address/theme/ecos-its_optex/utility/filemanager/api_audio_common.php`
- Stop Audio - `http://my_ip_address/api.php?api=[{"system": {"command": "stop_audio", "value": ""}, "debug": true}]`
- `http://my_ip_address/api.php?api=[{"system":{"command":"enable_audio","value":""},"debug":true}]`
 - 센서 이벤트 발생에 따른 명령어 수행을 허용한다.
- `http://my_ip_address/api.php?api=[{"system":{"command":"disable_audio","value":""},"debug":true}]`
 - 센서 이벤트 발생에 따른 명령어 수행을 제한한다.

Q: Ex: 내장 오디오

- 사전정의된 숫자로 재생 가능한 음원과 사용자 업로드후 재생 기능이 있다.
- 사용자 업로드 한 음원을 이웃하는 ITS API(**server 기능** 참조)로 재생시 네트워크 복사후 재생된다.
- `{"host":"","port":"","data":{"audio":{"source":"5","volume":"40","loop":"0"},"debug":true}}]`
- `http://my_ip_address/api.php?api=[{"audio":{"source":"5","volume":"40","loop":"0"},"debug":true}]`

Q: Ex: 웹 오디오

- 웹에서 다운로드 후 재생한다.
- `{"host":"","port":"","data":{"audio":{"source":"https://www.mfiles.co.uk/mp3-downloads/chopin-nocturne-op9-no2.mp3","volume":"40","loop":"1"},"debug":true}}]`
- `http://my_ip_address/api.php?api=[{"audio":{"source":"https://www.mfiles.co.uk/mp3-downloads/chopin-nocturne-op9-no2.mp3","volume":"40","loop":"1"},"debug":true}]`

◦ Return Value

```
{
  "category": "audio",
  "ip": "192.168.0.80",
  "response": {
    "sent": "https://www.mfiles.co.uk/mp3-downloads/chopin-nocturne-op9-
no2.mp3"
  }
}
```

Q: Ex: 오디오 응용

- {"host":"","port":"","data":[{"system":{"command":"sleep","value":"2"},"debug":true},{"system":{"command":"stop_audio","value":""},"debug":true},{"audio":{"source":"5","volume":"40","loop":"0"},"debug":true}]}
- http://my_ip_address/api.php?api=[{"system":{"command":"sleep","value":"2"},"debug":true},{"system":{"command":"stop_audio","value":""},"debug":true},{"audio":{"source":"5","volume":"40","loop":"0"},"debug":true}]
 - 재생되는 오디오를 n초후, 정지시키고 새 음원 재생
 - Sleep -> Stop Audio -> Play Audio
- {"host":"","port":"","data":[{"system":{"command":"stop_audio","value":""},"debug":true},{"audio":{"source":"5","volume":"40","loop":"0"},"debug":true}]}
- http://my_ip_address/api.php?api=[{"system":{"command":"stop_audio","value":""},"debug":true}, {"audio":{"source":"5","volume":"40","loop":"0"},"debug":true}]
 - 재생되는 오디오를 즉시 정지시키고 새 음원 재생(비상경보)
 - Stop Audio -> Play Audio

Q: Camera 명령 - 예약

- ITS mDVR 연동
- still_shot, motion_shot(mDVR), list_shot, download_shot, footprint(mDVR)
- Manager - http://my_ip_address/theme/ecos-its_optex/utility/filemanager/api_camera_common.php

Q: Ex: Camera

- {"camera":{"command":"still_shot","value":"","debug":true}} - 예약
- {"camera":{"command":"motion_shot","value":"","debug":true}} - 예약
- {"camera":{"command":"list_shot","value":"","debug":true}} - 예약
- {"camera":{"command":"download_shot","value":"","debug":true}} - 예약
- {"camera":{"command":"footprint","value":"","debug":true}} - (mDVR)
 - ITS microDVR과 연동되는 기능이다.
 - 이벤트 발생시점(Filename)을 기준으로 이전과 이후 영상을 저장하는 기능이다.
 - {"host":"","port":"","data":[{"camera":{"command":"footprint","value":"","debug":true}]}
 - http://my_ip_address/api.php?api=[{"camera":{"command":"footprint","value":"","debug":true}]

Q: KeyCode 기능

- 모든 명령문에 암호화(Encryption) 키 인증(Authentication) 기능
- 키(Keycode) 생성은 관리자 권한이며 생성후 모든 통신에 적용됨
- 전송시 명령어레벨에 [keyCode]항목에 첨부 (SHA256 - Hex)

Q: Ex: KeyCode

- `http://my_ip_address/api.php?api=[{"gpio": {"status": "2", "id": "io09", "hold": "0", "interval": "2"}, "keyCode": "e3b0c44298fc1c...", "debug": true}]`

Q: Server 기능

- Server 기능을 통해 각기다른 서버(ITS)로 명령 요청이 가능
- Node to Node의 IOT 기반의 수평적 제어구현
- Master와 Sub("server"명 소속)로 구분

Q: Ex: Server

- Console Test - `{"host": "", "port": "", "data": [{"gpio": {"status": "2", "id": "io12", "hold": "0.6", "interval": "4"}, "server": {"host": "A_ip_address", "port": "34001"}, "debug": true}, {"audio": {"source": "5", "volume": "40", "loop": "0"}, "server": {"host": "B_ip_address", "port": "34001"}, "debug": true}]}`
 - 콘솔은 ITS API의 28080포트를 통한 접속을 의미 한다.
 - 테스트는 센서버튼 클릭시 센서에 속한 명령 수행을 의미 한다.
- Browser Test - `http://my_ip_address/api.php?api=[{"gpio": {"status": "2", "id": "io12", "hold": "0.6", "interval": "4"}, "server": {"host": "S_ip_address", "port": "34001"}, "debug": true}, {"audio": {"source": "5", "volume": "40", "loop": "0"}, "server": {"host": "B_ip_address", "port": "34001"}, "debug": true}]}`
 - 브라우저에 ITS API URI를 통한 테스트 이다.
- CLI Test - `$ echo '{"gpio":{"status":"2","id":"io12","hold":"0.6","interval":"4"},"server":{"host":"A_ip_address","port":"34001"},"debug":true},{"audio":{"source":"5","volume":"40","loop":"0"},"server":{"host":"B_ip_address","port":"34001"},"debug":true}}' | nc my_ip_address 34001 -q 0`
 - 설명: 서버 A_ip_address에는 gpio 명령은 보내고 서버 B_ip_address에는 audio 명령을 실행하는 예 이다.

Q: Custom 명령 - 요약

- tcp_socket, http_get, http_post를 통한 네트워크상의 이기종 서버에 명령을 전송함
- 주위를 요한다. - 무한루프 가능성
- "tcp_socket"
 - 데이터 기본형식은 JSON이다. 평문인 경우 -> "isJson":false
 - `{"host": "", "port": "", "data": [{"custom": {"method": "tcp_socket", "isJson": true, "data": {"id_01": "name_01", "id_02": "name_02", "id_03": "name_03", "id_04": "name_04"}}, "server":`

- ```

{"host":"192.168.0.80","port":"33001"},"debug": true}}
 ■ ASC: {"data": {"id_04": "name_04", "id_03": "name_03", "id_02": "name_02", "id_01":
 "name_01"}}
 ○ {"host":"","port":"","data":{"custom":{"method":"tcp_socket", "isJson":false,
 "data":"id_01=name_01,id_02=name_02,id_03=name_03,id_04=name_04"},"server":
 {"host":"192.168.0.80","port":"33001"},"debug": true}}
 ■ ASC: id_01=name_01,id_02=name_02,id_03=name_03,id_04=name_04
 ○ {"host":"","port":"","data":{"custom":{"method":"tcp_socket", "isJson":false,
 "data":"id=g300t100_192_168_0_9_0012,name=R01,beep=1,shot=,latS=0.0,lngS=0.0,latE=0.0,lng
 E=0.0,count=1,block=0,status=1,msg=Active_Event"},"server":
 {"host":"192.168.0.91","port":"38087"},"debug": true}}
 ○ {"host":"","port":"","data":{"custom":{"method":"tcp_socket", "isJson":false,
 "data":"id=g400t300_192_168_0_20_0001,name=GKT
 Basic,beep=1,status=1,shot=192.168.0.90/mDVR/20211216_155806/039_Z008.png,video=,count
 =1,block=0,msg=Tailing,subzone=GIKENT"},"server":
 {"host":"192.168.0.91","port":"38087"},"debug": true}}
 ■ ASC:
 id=g300t100_192_168_0_9_0012,name=R01,beep=1,shot=,latS=0.0,lngS=0.0,latE=0.0,lng
 E=0.0,count=1,block=0,status=1,msg=Active_Event
 ■ ASC:
 id=g300t100_192_168_0_9_0012,name=R01,beep=0,shot=,latS=0.0,lngS=0.0,latE=0.0,lng
 E=0.0,count=0,block=0,status=2,msg=Idle_Event
 ■ ASC:
 id=g300t100_192_168_0_9_0012,name=R01,beep=1,shot=,latS=0.0,lngS=0.0,latE=0.0,lng
 E=0.0,count=2,block=0,status=9,msg=Error_Event
 ■ ASC: id=g400t300_192_168_0_20_0001,name=GKT
 Basic,beep=1,status=1,shot=192.168.0.90/mDVR/20211216_155806/039_Z008.png,video=,
 count=1,block=0,msg=Tailing,subzone=GIKENT
 • "http_post"
 ○ {"host":"","port":"","data":{"custom":{"method":"http_post", "data":
 {"id_01":"name_01","id_02":"name_02","id_03":"name_03","id_04":"name_04"},"server":
 {"url":"http://192.168.0.80/api.php"},"debug": true}}}
 • "http_get"
 ○ {"host":"","port":"","data":{"custom":{"method":"http_get", "data":
 {"id_01":"name_01","id_02":"name_02","id_03":"name_03","id_04":"name_04"},"server":
 {"url":"http://192.168.0.80/api.php"},"debug": true}}}

```

## Q: Alarm 명령 - 예약 (Crontab)

- 콘솔 윈도 내에 알람 설정을 실행 한다.
- 본 기능은 리눅스 Crontab 룰을 따른다.
- 실행명령(스크립트)은 기존 센서룰과 같다.



- Log

## Q: Database Query 명령 - 예약

- Database

# Programming Examples

---

- python Code Example - **pyCode.py**

```
특정 릴레이(io09)를 토글링 한후 1초후 되돌린다.
디버그 모드를 활성화("debug":True)해서 반환값을 받는다.

#!/usr/bin/env python
-*- coding: utf-8 -*-

import socket
import json

host = '192.168.0.80'
port = 34001
obj = [{"gpio":{"status":"2","id":"io09","hold":"1"},"debug":True}]

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.settimeout(1) # settimeout
try:
 s.connect((host,port))
 s.send(json.dumps(obj).encode('utf-8'))
 print(s.recv(1024))
except socket.error:
 print('socket.error')
except socket.timeout:
 print('socket.timeout')
finally:
 s.close()
```

- Python Code Test (Linux Terminal)

```
$
$ python pyCode.py
{"category": "gpio", "ip": "192.168.0.80", "status": "2", "response":
```

```
{"io09_desc": 0}}
$
```

- PHP Code Example - **phpCode.php**

```
// 특정 릴레이(io11)를 활성화(1:On) 한후 3초 후 비 활성화("hold":"3" -> Off) 시킨다.
// 디버그 모드를 활성화("debug":True)해서 반환값을 받는다.
```

```
<?php
$obj = '[{"gpio":{"status":"1","id":"io11","hold":"3"},"debug":true}]';

try {
 // socket_create
 $socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);
 if ($socket === false) {
 echo "socket_create() failed: " .
socket_strerror(socket_last_error());
 }
 // socket_connect
 $result = socket_connect($socket, "192.168.0.80", "34001");
 if ($result === false) {
 echo "socket_connect() failed. " .
socket_strerror(socket_last_error($socket));
 }
 // socket_write
 socket_write($socket, $obj, strlen($obj));
 // socket_read when "debug":true
 print socket_read($socket, 1024);
} finally {
 socket_close($socket);
}

?>
```

- PHP Code Test (Linux Terminal)

```
$ php phpCode.php
{"category": "gpio", "ip": "192.168.0.80", "status": "2", "response":
{"io11_desc": 1}}
$
```

[illegible]

```

/_/ daivoc@gmail.com /_/
/_/_/_/_/_/_/_/_/_/_/_/_/_/_/_/

```