

WIFI Audio HTTP API

**All Copyrights belong to iEAST
No disclosure without permission**

Confidential. No Disclosure

2014.07

Revision record

Date	Revision version	Description	Author
2014-03-26	0.1	Initially built	Shao Chunsuo
2014-4-10	0.2	Revised interface description, added multiroom interfaces	Shao Chunsuo
2014-6-11	0.3	Added favorites, recording and MCU communication	Shao Chunsuo
2014-6-25	0.4	Revised some interface description, song's meta information was changed to hex coding, strikethrough were added to the interfaces that are unavailable	Shao Chunsuo
2014-7-1	0.5	Added WPS server start interface	Shao Chunsuo
2014-12-07	0.6	Added line-in switch interface	Haiyong
2016-3-28	0.7	Added volume up/down interfaces to define the volume increase/decrease percentage	Ianming
2016-3-28	0.8	Added interfaces to get the preset program	Ianming

Confidential. No Disclosure

目 录

WIFI AUDIO	1
HTTP API	1
1. BRIEF	6
1.1. BRIEF	6
1.2. HTTP GET.....	6
1.3. ABOUT HEXED.....	6
2. API LIST	7
2.1. GET DEVICE INFORMATION	7
2.2 NETWORK.....	9
2.2.1 <i>Get the scan AP list</i>	9
2.2.2 <i>Connect wifi</i>	10
2.2.3 <i>connect hide wifi</i>	10
2.2.4 <i>Get the connect state</i>	11
2.3 PLAYBACK CONTROL	11
2.3.1 <i>Get playback state</i>	11
2.3.2 <i>playback</i>	13
2.3.3 <i>Pause / Resume</i>	14
2.3.4 <i>Pause or Resume</i>	14
2.3.5 <i>Previous</i>	14
2.3.6 <i>Next</i>	14
2.3.7 <i>Seek</i>	14
2.3.8 <i>stop</i>	14
2.3.9 <i>Set volume</i>	14
2.3.10 <i>Mute</i>	15
2.3.11 <i>Loop mode set</i>	15
2.3.12 <i>EQ set</i>	15
2.4 USB DISK PLAYBACK.....	16
2.4.1 <i>Get the playlist of USB</i>	16
2.4.2 <i>Get Music information in USB disk</i>	16
2.6 MULTIROOM	18
2.6.1 <i>Get slave list</i>	18
2.6.2 2.6.3 <i>kickout slave</i>	19
<i>Kickout one slave from group.</i>	19
2.6.4 <i>mask slave</i>	19
2.6.5 <i>un-mask slave</i>	19
2.6.6 <i>set slave volume</i>	19
2.6.7 <i>set master volume</i>	19
2.6.8 <i>mute slave</i>	19
2.6.9 <i>mute master</i>	19
2.6.10 <i>Slave playback channel set</i>	20
2.6.11 <i>Master playback channel set</i>	20

2.6.12 start WPS server	20
2.6.13 stop WPS server	20
2.6.14 start WPS Client	20
2.8 DEVICE CONTROL	21
2.8.1 Set SSID	21
2.8.2 Set AP password	21
2.8.3 Restore	21
2.8.4 Reboot	21
2.8.5 shutdown	21
2.8.6 Get the shutdown timer	22
2.8.7 Close WIFI	22
2.9 ONLINE UPGRADE.....	22
2.9.1 Check the new version.....	22
2.9.2 Start upgrade.....	22
http:// 10.10.10.254/httpapi.asp?command=getMvRemoteUpdateStart.....	22
It will download the firmware first and then burn it.	22
2.9.3 Check the upgrade status	22
2.9.6 Get the upgrade percent	23
2.10 ALARM CLOCK.....	23
2.10.1 time sync.....	23
2.10.2 alarm set.....	23
2.11 PLAYBACK SOURCE.....	24
2.12 GPIO SIMULATION.....	25
2.13 VOICE PROMPT.....	26
2.14 PRE-SET PROGRAM 1-6.....	26

Confidential. No Disclosure

1. Brief

1.1. brief

Wifi Audio AXX module is the SoC module for WiFi audio solutions, it support Smartlink, DLNA and Airplay.

It also support some http API for quick access.

1.2. HTTP get

You can send HTTP get request to the module, the response if in JSON.

Request format is `http://x.x.x.x/httpapi.asp?command=*****`

X.x.x.x is the IP address, (Below, assume the IP is 10.10.10.254)

***** is the command.

1.3. About Hexed

Some data should hexed before transfer it.

Here is the hex method (in C):

```
int hex2ascii(const char *pSrc, unsigned char *pDst, unsigned int nSrcLength, unsigned int nDstLength)
```

```
{
    int i, j = 0;
    memset(pDst, 0, nDstLength);
    for (i = 0; i < nSrcLength; i += 2)
    {
        char val1 = pSrc[i];
        char val2 = pSrc[i+1];
        if (val1 > 0x60) val1 -= 0x57;
        else if (val1 > 0x40) val1 -= 0x37;
        else val1 -= 0x30;
        if (val2 > 0x60) val2 -= 0x57;
        else if (val2 > 0x40) val2 -= 0x37;
        else val2 -= 0x30;
        if (val1 > 15 || val2 > 15 || val1 < 0 || val2 < 0)
            return 0;
        pDst[j] = val1 * 16 + val2;
        j++;
    }
    return j;
}
```

```
}

int ascii2hex(char* ascii_in, char* hex_out, int ascii_len, int hex_len)
{
    const char hex[16] = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
    int i = 0;
    int ret = 0;
    memset(hex_out, 0, hex_len);
    while( i < ascii_len )
    {
        int b= ascii_in[i] & 0x000000ff;
        hex_out[i*2] = hex[b/16] ;
        hex_out[i*2+1] = hex[b%16] ;
        ++i;
        ret += 2;
    }
    return ret;
}
```

2. API list

2.1. Get device information

[http:// 10.10.10.254/httpapi.asp?command=getStatusEx](http://10.10.10.254/httpapi.asp?command=getStatusEx)

- result:

```
{
  "language": "en_us",
  "ssid": "FA5100_a4dc",
  "firmware": "WIFIAudio.multil_room.1.2.20140324 ",
  "builddate": "release ",
  "Release": "20140324 ",
  "group": "",
  "expired": "0",
  "internet": "0",
  "uuid": "de cf 1d 2e b1 60 e4 38 ",
}
```

```

"netstat": "0",
"ssid": "",
"apcli0": "",
"eth2": "192.168.120.112",
"hardware": "WiiMu-A03 ",
}

```

language	language
ssid	SSID
firmware	version
builddate	Can be: release, debug, backup Release: means this is a release version Debug: means this is a debug version Backup means this is a backup version
Release	The build date
group	
expired	1 means the firmware has expired
internet	Internet access: 0: not ready; 1: ready
uuid	UUID
netstat	WIFI Station connect state: 0: no connection; 1: connecting; 2: connected
ssid	The router name connected. The name is hexed
apcli0	WIFI Station IP address
ra0	WiFi AP IP address, normally is 10.10.10.254
eth2	Ethernet IP address
hardware	The hardware version
project	The project name
VersionUpdate	Is there new version.
NewVer	If there is new version, this is the version number.
DeviceName	The device upnp and airplay friendly name
temp_uuid	Temp uuid, will change after boot up
capability	Bit: 0 if airplay support 1 if ethernet support 2 if USBdisk support 3 if WPS button support

	4 if battery support 5 if preset key support 6 if I2S in support
streams	Bit: 0 if airplay enabled 1 if DLNA enabled 2 if TTPod support 3 if TuneIn support 4 if Pandora support 5 if DoubanFM support
external	
preset_key	presets key number
plm_support	Bit: 0 LineIn (Aux support) 1 BT support 2 optical support
WifiChannel	WiFi channel
AP_clients	AP client number
RSSI	Router RSSI (0 ~ 100), 100 is the best
TxQuality	Tx quality,
RxQuality	Rx quality
battery	1 means battery is in charging
battery_percent	0 ~100
securemode	WiFi secure or not
psk	If WiFi is in secure, the password
usb_storagesize	Udisk storage size
usb_freesize	Udisk free size
part1_storage	User1 part in ROM, storage size
part1_free	User1 part in ROM, free size
part2_storage	User2 part in ROM, storage size
part2_free	User2 part in ROM, free size

2.2 Network

2.2.1 Get the scan AP list

[http:// 10.10.10.254/httpapi.asp?command=wlanGetApListEx](http://10.10.10.254/httpapi.asp?command=wlanGetApListEx)

```
{
  "res": "0",
  "aplist": [
    {
      "ssid": "wmmAudio_a7b8",
      "bssid": "00:22:6c:00:a7:b8",
      "rssi": "76",
      "channel": "1",
      "auth": "OPEN",
      "encry": "NONE",
      "extch": "1"
    },
    {
      "ssid": "WIIMU_Network",
      "bssid": "20:dc:e6:cb:7e:78",
      "rssi": "70",
      "channel": "11",
      "auth": "WPA2PSK",
      "encry": "AES",
      "extch": "0"
    },
    {
      "ssid": "WiFiAudio_29b0",
      "bssid": "00:22:6c:16:29:b0",
      "rssi": "65",
      "channel": "11",
      "auth": "OPEN",
      "encry": "NONE",
      "extch": "0"
    }
  ]
}
```

res	number
aplist	
ssid	Wifi SSID name (hide SSID is not include), hexed
bssid	MAC
rssi	RSSI (0~100)
channel	Wifi channel
auth	Authorization
encry	Encrypt
extch	

2.2.2 Connect wifi

<http://10.10.10.254/httpapi.asp?command=wlanConnectApEx:ssid=xxx:ch=n:auth=xxx:encry=xxx:pwd=xxx:chext=n>

Connect the device to a router. When connecting, the device AP connection may lost. This API has no return result, you can call wlanGetConnectState to get the connect state if the AP connection still not lost.

ssid	Router ssid, hexed
channel	Wifi channel
auth	Authorization
encry	encrypt
pwd	Passowrd, hexed , if no password, not input here
chext	1

2.2.3 connect hide wifi

<http://10.10.10.254/httpapi.asp?command=wlanConnectHideApEx:ssid:pwd>

Connect the device to a hide router. When connecting, the device AP connection may lost.

This API has no return result, you can call wlanGetConnectState to get the connect state if the AP connection still not lost.

Ssid and pwd must hexed, if no pwd, API is:

<http://10.10.10.254/httpapi.asp?command=wlanConnectHideApEx:ssid>

2.2.4 Get the connect state

<http://10.10.10.254/httpapi.asp?command=wlanGetConnectState>

Note the return result is not in json

- return:

返回值字段	字段说明
PROCESS	In progress
PAIRFAIL	Wrong password
FAIL	Connect fail
OK	connected

2.3 Playback control

Note, the 3rd party DLNA can't be controlled.

2.3.1 Get playback state

<http://10.10.10.254/httpapi.asp?command=getPlayerStatus>

```
{  
  "mainmode": "0",  
  "nodetype": "0",  
  "mode": "3",
```

```

"sw": "0",
"status": "play",
"curpos": "12900",
"totlen": "229000",
"Title": "736865",
"Artist": "47726f6f766520436f766572616765",
"Album": "xxxxxxxxxx",
"Year": "2005",
"Track": "7",
"Genre": "Dance",
"locallistflag": "1",
"locallistfile": "",
"plicount": "1",
"plicurr": "1",
"vol": "90",
"mute": "0",
"iuri": "xxxxxxxxxxxxxxxxxxxxxxxx",
"uri": "xxxxxxxxxxxxxxxxxxxxxxxx"
}

```

● 字段说明:

返回值字段	字段说明
type	0: master or standalone device 1: slave
ch	0 stereo, 1 left, 2 right
mode	0 none 1 Airplay 2 3 rd party DLNA 11 ~ 19 Wiimu playlist, 11 is the USB disk playlist 21 ~ 29 Http API playback, 21 is the USB disk playback 30 alarm 40 AUX 41 BT 42 external storage 43 optical 50 mirror

	60 voice mail 99 slave
loop	Loop mode: 0: sequence, no loop 1: single loop 2: shuffle all 3: loop all
eq	The Equalizer number
status	0 stop 1 play 2 load 3 pause
curpos	Position, in ms
totlen	Duration in ms
Title	Return hexed data, the music name or music URL
Artist	Return hexed data
Album	Return hexed data
udsikflag	Is there USB disk
plicount	The track number of playlist
plicurr	Current track index
vol	Current volume
mute	Current mute state

2.3.2 playback

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:play:uri>

Play the URI

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:playlist:uri:<index>>

Play the URI (URI is the m3u playlist, index is the start index)

http://10.10.10.254/httpapi.asp?command=setPlayerCmd:hex_playlist:uri:<index>

Play the URI (URI is the m3u playlist, index is the start index), here, uri is hexed

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:playLocalList:<index>>

Play the USB disk, index is the start index

2.3.3 Pause / Resume

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:pause>

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:resume>

2.3.4 Pause or Resume

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:onepause>

If the state is paused, resume it; or, pause it.

2.3.5 Previous

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:prev>

2.3.6 Next

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:next>

2.3.7 Seek

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:seek:position>

Position is from 0 to duration in second

2.3.8 stop

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:stop>

2.3.9 Set volume

Vol++

<http://X.X.X.X/httpapi.asp?command=setPlayerCmd:Vol%2B%2Bn>

Vol--

<http://X.X.X.X/httpapi.asp?command=setPlayerCmd:Vol--n>

to show the playing volume

<http://X.X.X.X/httpapi.asp?command=setPlayerCmd:Vol--n>

note: n is from 1-100, it stands each volume tuning will increase or decrease the volume by n%.

2.3.10 Mute

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:mute:n>

Mute: n=1

Unmute: n=0

the slave mute state will be set at the same time.

2.3.11 Loop mode set

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:loopmode:n>

n	
0	Sequence, no loop
1	Single loop
2	Shuffle loop
-1	Sequence loop

2.3.12 EQ set

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:equalizer:mode>

mode	
0	disable
1	Classic
2	Popular
3	Jazzy
4	Vocal

Not all projects support EQ.

2.3.13 EQ get

<http://10.10.10.254/httpapi.asp?command=getEqualizer>

2.4 USB disk playback

2.4.1 Get the playlist of USB

<http://10.10.10.254/httpapi.asp?command=getLocalPlayList>

```
{
  "num": "2",
  "loclist": [
    {
      "file": "/media/sda1/avril lavigne - tik tok.mp3"
    },
    {
      "file": "/media/sda1/avril lavigne - hush hush.mp3"
    }
  ]
}
```

num	Music file number in USB disk
loclist	
file	Music file, hexed

2.4.2 Get Music information in USB disk

<http://10.10.10.254/httpapi.asp?command=getFileInfo:index:range>

Get the music metadata, index is the sequence number in USBDisk playlist, range is the number

If range is large than 1, return multiple fileinfo

For example:

```
{“num”:”2”, “infolist”:[
```



```

{
  "filename": "/media/sda1/avril lavigne – hush hush.mp3",
  "totlen": "0",
  "Title": "unknown",
  "Artist": "unknown",
  "Album": "unknown"
},
{
  "filename": "/media/sda1/avril lavigne – hush hush2.mp3",
  "totlen": "0",
  "Title": "unknown",
  "Artist": "unknown",
  "Album": "unknown"
}}

```

Or return single metadata

```

{
  "filename": "/media/sda1/avril lavigne – hush hush.mp3",
  "totlen": "0",
  "Title": "unknown",
  "Artist": "unknown",
  "Album": "unknown"
}

```

● 字段说明:

返回值字段	字段说明
filename	Filename, hexed
totlen	
Title	Return unknown or hexed data
Artist	Return unknown or hexed data
Album	Return unknown or hexed data

2.4.3 Play the USBDisk

<http://10.10.10.254/httpapi.asp?command=setPlayerCmd:playLocalList:index>

2.6 Multiroom

2.6.1 Get slave list

<http://10.10.10.254/httpapi.asp?command=multiroom:getSlaveList>

```
{
  "slaves": "1",
  "slave_list": [
    {
      "name": "FA5100_a3f4",
      "mask": "0",
      "volume": "90",
      "mute": "0",
      "channel": "0",
      "ip": "10.10.10.100",
      "version": "WiFiAudio.1.2.2321"
    }
  ]
}
```

slaves	Slave number
slave_list	
name	Slave device name
mask	If slave is masked, it will act as a standalone device.
Volume	Slave volume
Mute	slave is mute or not
Channel	Slave channel: 0 stereo 1 left 2 right
ip	Slave IP address
version	Slave firmware version
Ssid	Slave ssid
Uuid	Slave uuid

2.6.2 2.6.3 kickout slave

<http://10.10.10.254/httpapi.asp?command=multiroom:SlaveKickout:ip>

Kickout one slave from group.

2.6.4 mask slave

<http://10.10.10.254/httpapi.asp?command=multiroom:SlaveMask:ip>

Mask one slave, If slave is masked, it will act as a standalone device.

2.6.5 un-mask slave

<http://10.10.10.254/httpapi.asp?command=multiroom:SlaveUnMask:ip>

2.6.6 set slave volume

- 接口 URL:

<http://10.10.10.254/httpapi.asp?command=multiroom:SlaveVolume:ip:volume>

2.6.7 set master volume

- 接口 URL:

http://10.10.10.254/httpapi.asp?command=setPlayerCmd:slave_vol:volume

Just set the master or standalone volume, not affect the slave volume.

2.6.8 mute slave

<http://10.10.10.254/httpapi.asp?command=multiroom:SlaveMute:ip:mute>

2.6.9 mute master

http://10.10.10.254/httpapi.asp?command=setPlayerCmd:slave_mute:mute

Just mute the master or standalone , not affect the slave .

2.6.10 Slave playback channel set

<http://10.10.10.254/httpapi.asp?command=multiroom:SlaveChannel:ip:channel>

Slave channel:

0 stereo

1 left

2 right

2.6.11 Master playback channel set

http://10.10.10.254/httpapi.asp?command=setPlayerCmd:slave_channel:channel

Just set the master or standalone playback channel , not affect the slave .

Slave channel:

0 stereo

1 left

2 right

2.6.12 start WPS server

<http://10.10.10.254/httpapi.asp?command=wpsservermode>

Start WPS server, wait for the slave connection

2.6.13 stop WPS server

<http://10.10.10.254/httpapi.asp?command=wpscancel>

2.6.14 start WPS Client

<http://10.10.10.254/httpapi.asp?command=wpsclientmode>

scan the WPS server and connect it

2.8 Device control

2.8.1 Set SSID

`http:// 10.10.10.254/httpapi.asp?command=setSSID:value`

Set the device SSID

SSID value should < 16 bytes without special charaters.

2.8.2 Set AP password

`http:// 10.10.10.254/httpapi.asp?command=setNetwork:1:password`

Password value should < 16 bytes without special charaters.

`http:// 10.10.10.254/httpapi.asp?command=setNetwork:0`

Remove AP passowrd

2.8.3 Restore

`http:// 10.10.10.254/httpapi.asp?command=restoreToDefault`

2.8.4 Reboot

`http:// 10.10.10.254/httpapi.asp?command=reboot`

2.8.5 shutdown

`http:// 10.10.10.254/httpapi.asp?command=setShutdown:sec`

Shutdown device in sec

sec:

0: shutdown immediately

-1: cancel the previous shutdown timer

2.8.6 Get the shutdown timer

`http:// 10.10.10.254/httpapi.asp?command=getShutdown`

Return the seconds

2.8.7 Close WIFI

`http:// 10.10.10.254/httpapi.asp?command=setPowerWifiDown`

Device is still active but Wifi is closed.

2.9 Online Upgrade

2.9.1 Check the new version

`http:// 10.10.10.254/httpapi.asp?command=getMvRemoteUpdateStartCheck`

2.9.2 Start upgrade

`http:// 10.10.10.254/httpapi.asp?command=getMvRemoteUpdateStart`

It will download the firmware first and then burn it.

2.9.3 Check the upgrade status

`http:// 10.10.10.254/httpapi.asp?command=getMvRemoteUpdateStatus`

return	
40	Find new version
10	checking
21	Check fail
22	Download fail
23	Verify firmware fail
25	downloading
27	Download finished, burning (for user part)
30	Download finished, burning
others	No new version

--	--

2.9.6 Get the upgrade percent

<http://10.10.10.254/httpapi.asp?command=getMvRomBurnPrecent>

{"status":"0","progress":"50"}	
status	0 in progress -1 failed
progress	0~100

2.10 Alarm clock

2.10.1 time sync

If the device has no internet access, you need to sync its time with:

<http://10.10.10.254/httpapi.asp?command=timeSync:YYYYMMDDHHMMSS>

YYYY is year (such as 2015), MM is month (01~12), DD is day (01~31), HH is hour (00~23), MM is minute (00~59), SS is second (00~59)

In UTC

2.10.2 alarm set

[http://10.10.10.254/httpapi.asp?command=setAlarmClock:n:trig:op:time\[:day\]\[:url\]](http://10.10.10.254/httpapi.asp?command=setAlarmClock:n:trig:op:time[:day][:url])

n: 0~2, currently support max 3 alarm

Trig: the alarm trigger:

0 cancel the alarm, (AXX+TMR+S010&)

1 once, day should be YYYYMMDD

2 every day, day is no need

3 every week, day should be 2 bytes (00" ~ "06"), means from Sunday to Saturday.

4 every week, day should be 2 bytes, the bit 0 to bit 6 means the effect, for example, "7F" means every day in week, "01" means only Sunday.

5 every month, day should be 2 bytes ("01" ~ "31")

op: the action

0 shell execute

- 1 playback or ring
- 2 stop playback

Time should be HHMMSS

url: the shell path or playback url, should less than 256 bytes

2.10.3 Get alarm

http://10.10.10.254/httpapi.asp?command=getAlarmClock:n

n: 0~2, currently support max 3 alarm

```
{"enable": "1",  
  "trigger": "%d",  
  "operation": "%d",  
  "date": "%02d:%02d:%02d", //if not a "every day" alarm, no this  
  "week_day": "%d", //if not a "every week" alarm, no this  
  "day": "%02d", //if not a "every month" alarm, no this  
  "time": "%02d:02d:%02d",  
  "path": "%s"}
```

2.10.4 Stop the current alarm

http://10.10.10.254/httpapi.asp?command=alarmStop

2.11 Playback source

2.11.1 switch playback source

http://10.10.10.254/httpapi.asp?command=setPlayerCmd:switchmode:%s

line-in:

bluetooth:

optical:

Above, device will capture I2S data and play it

udisk: if Udisk is connected to the device, it will playback the UDisk, or device will capture I2S data and play it

wifi:

2.12 GPIO simulation

2.12.1 Pull up GPIO

<http://10.10.10.254/httpapi.asp?command=IOSimuPullUp:%d>

00: GPIO_nReload

11: GPIO_nReady (only A02)

12: GPIO_DCD

14: GPIO_RIN

18: GPIO1 (only A11)

17: GPIO2 (only A11)

2.12.2 Pull down GPIO

<http://10.10.10.254/httpapi.asp?command=IOSimuPullUp:%d>

2.12.3 Read GPIO level

<http://10.10.10.254/httpapi.asp?command=IOSimuRead:%d>

2.12.4 Key simulation

<http://10.10.10.254/httpapi.asp?command=IOSimuKeyIn:%d>

Simulate the MCU+KEY+XXX command

<http://10.10.10.254/httpapi.asp?command=IOSimuKeyOut:%d>

Generate the AXX+KEY+XXX command to MCU

2.13 Voice prompt

<http://10.10.10.254/httpapi.asp?command=PromptEnable>

<http://10.10.10.254/httpapi.asp?command=PromptDisable>

2.14 pre-set program 1-6

<http://10.10.10.254/httpapi.asp?command=MCUKeyShortClick:n>

n is from 1-6

Confidential. No Disclosure