

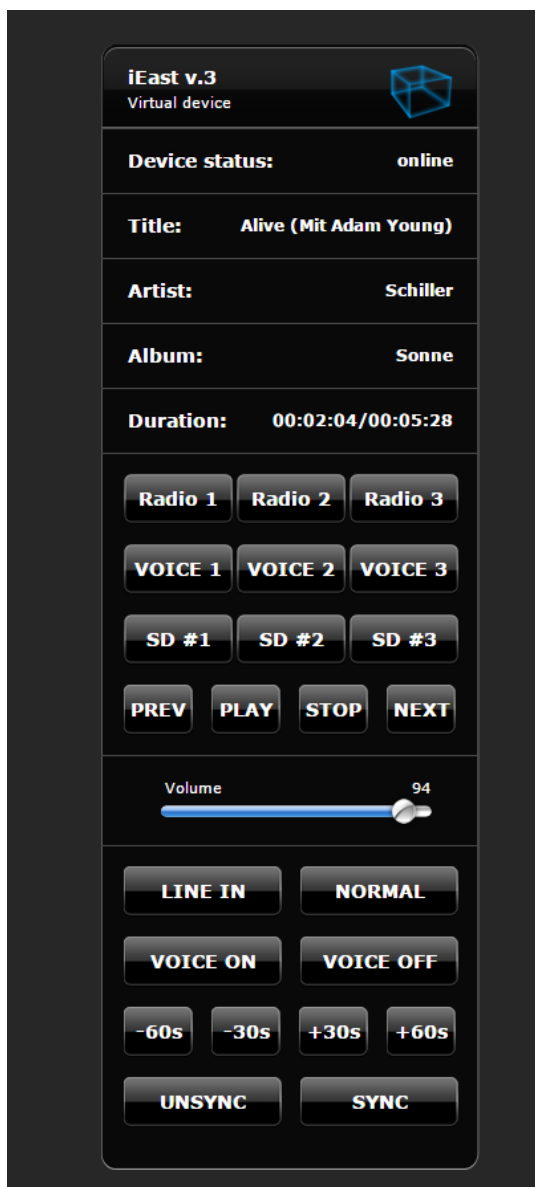
Introduction

This document will be explained how to use iEast SoundStreamer device with for Fibaro Home Center 2. We will talk about special plugin which allows to control it.

Primary functions

Home Controller Fibaro allows to create virtual devices (VD). This is custom UI modules consists of buttons, text fields and sliders. Every control element is able to handle click events and does some action. Current VD allows to connect to iEast module and sent HTTP requests to it.

On this screenshot provided default view of iEast Fibaro plugin. Users will be able to modify it (create own custom buttons, develop his own scripts and so on).

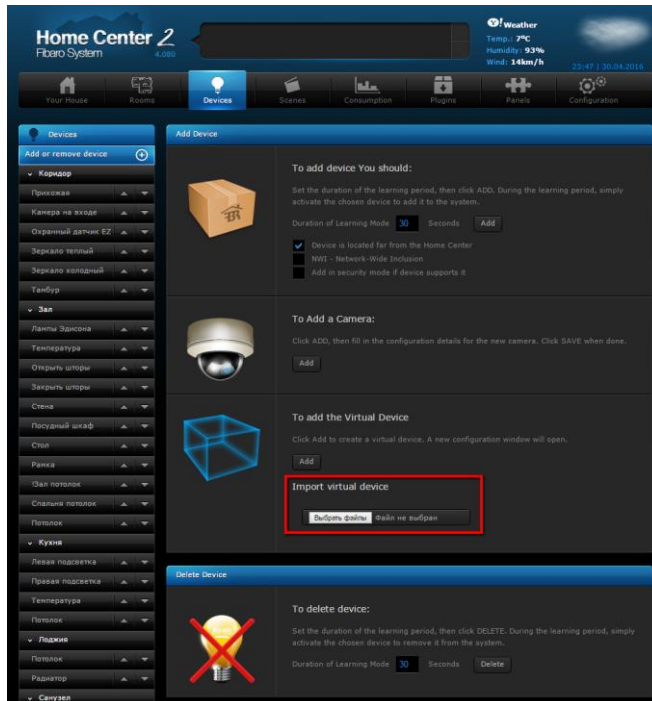


1. Device status allows to see if Fibaro HC2 is able to control iEast. It could be online or offline.
2. Title, Artist and Album for current song.
3. Duration: current time / total length.
4. Three example buttons allow to start playback radio station from web.
5. Three example buttons call online voice generator service. It should be any home event notification, weather forecast and so on.
6. Three example buttons allow to start playback from SD card.
7. Playback control (previous item, play, stop, next item).
8. Volume control.
9. Input selector.
10. Voice guide (turn on/off).
11. Seek playback
12. Multi room functionality.

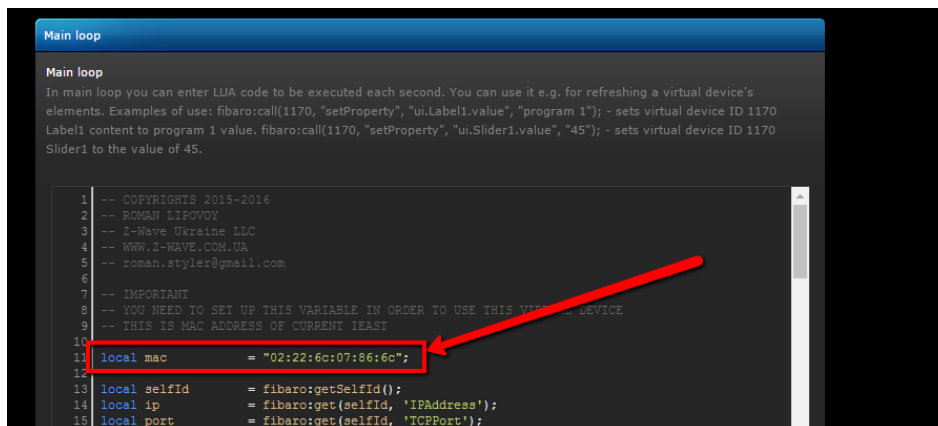
Installation

Installation procedure is quite simple. User need to do several steps:

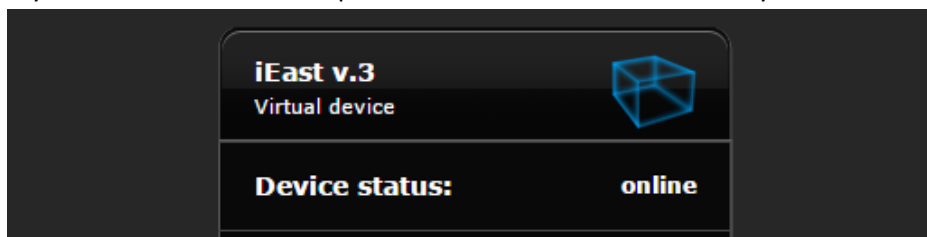
1. Ensure iEast device successfully connected to your home network.
2. Login into Fibaro HC2 and click import VD in interface.



3. Select file iEAST.vfib provided with this manual.
4. After several seconds plugin will be available in Fibaro interface.
5. Scroll down to the bottom of VD and set current MAC address of iEast



6. If you don't know to get MAC address of iEast, just use application like this: <http://www.advanced-ip-scanner.com/>
7. If you see state **online** in imported VD it means HC2 successfully found iEast.



How it works

Application consists of:

- main loop (code which executed every several seconds, it allows to get current song info and ping connection to iEast).
- small scripts for every button.

Play radio

```
-- This code allows to start playback of any web radio or file
-- Just modify url variable
local url                = "http://online.radiorecord.ru:8102/dc_320";

local selfId            = fibaro:getSelfId();
local ip                = fibaro:get(selfId, 'IPAddress');
local port              = fibaro:get(selfId, 'TCPPort');
local connection        = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection,
"httpapi.asp?command=setPlayerCmd:playlist:" .. url .. ":1");
```

Play web generated voice command

```
-- This code allows to play voice message generated from text
-- You can use service like this: http://www.voicerss.org/api/documentation.aspx
-- Just modify value of url variable
local url                = "http://translate.google.com/translate_tts?ie=UTF-
8&q=Text%20generated%20by%20Google%20Voice!&tl=en&total=1&idx=0&textlen=31&tk=917202.773248&clien
t=t&prev=input";

local selfId            = fibaro:getSelfId();
local ip                = fibaro:get(selfId, 'IPAddress');
local port              = fibaro:get(selfId, 'TCPPort');
local connection        = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection,
"httpapi.asp?command=setPlayerCmd:playlist:" .. url .. ":1");
```

Play file from SD Card

```
-- This code allows to play file from SD card by index
-- Just modify value of index variable to use it
local index              = 1;

local command           = "httpapi.asp?command=setPlayerCmd:playLocalList:" .. index;
local selfId            = fibaro:getSelfId();
local ip                = fibaro:get(selfId, 'IPAddress');
local port              = fibaro:get(selfId, 'TCPPort');
local connection        = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
```

Play previous item in playlist

```
-- Play previous item in playlist

local command           = "httpapi.asp?command=setPlayerCmd:prev";
```

```
local selfId          = fibaro:getSelfId();
local ip              = fibaro:get(selfId, 'IPAddress');
local port            = fibaro:get(selfId, 'TCPPort');
local connection      = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
```

Play

```
-- Resume playback
```

```
local command        = "httpapi.asp?command=setPlayerCmd:play";
local selfId         = fibaro:getSelfId();
local ip             = fibaro:get(selfId, 'IPAddress');
local port           = fibaro:get(selfId, 'TCPPort');
local connection      = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
```

Stop playback

```
-- Stop playback
```

```
local command        = "httpapi.asp?command=setPlayerCmd:stop";
local selfId         = fibaro:getSelfId();
local ip             = fibaro:get(selfId, 'IPAddress');
local port           = fibaro:get(selfId, 'TCPPort');
local connection      = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
```

Play next song in playlist

```
-- Play next item in playlist
```

```
local command        = "httpapi.asp?command=setPlayerCmd:next";
local selfId         = fibaro:getSelfId();
local ip             = fibaro:get(selfId, 'IPAddress');
local port           = fibaro:get(selfId, 'TCPPort');
local connection      = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
```

Adjust current volume

```
-- Set volume on change slider
```

```
local selfId          = fibaro:getSelfId();
local ip              = fibaro:get(selfId, 'IPAddress');
local port            = fibaro:get(selfId, 'TCPPort');
local connection      = Net.FHttp(ip, port);
local volume          = fibaro:getValue(selfId, "ui.volume.value");
local command         = "httpapi.asp?command=setPlayerCmd:vol:" .. volume;
local response, status, errorCode = connection.GET(connection, command);
local json = json.decode(response);
```

Select line input

```
-- Activate line input
```

```
local command      = "httpapi.asp?command=setPlayerCmd:switchmode:line-in";
local selfId       = fibaro:getSelfId();
local ip           = fibaro:get(selfId, 'IPAddress');
local port         = fibaro:get(selfId, 'TCPPort');
local connection   = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
local json         = json.decode(response);
```

Select local playback input

```
-- Activate line input
```

```
local command      = "httpapi.asp?command=setPlayerCmd:switchmode:line-in";
local selfId       = fibaro:getSelfId();
local ip           = fibaro:get(selfId, 'IPAddress');
local port         = fibaro:get(selfId, 'TCPPort');
local connection   = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
local json         = json.decode(response);
```

Turn on voice guide

```
-- Activate voice guide
```

```
local command      = "httpapi.asp?command=PromptEnable";
local selfId       = fibaro:getSelfId();
local ip           = fibaro:get(selfId, 'IPAddress');
local port         = fibaro:get(selfId, 'TCPPort');
local connection   = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
local json         = json.decode(response);
```

Turn off voice guide

```
-- Deactivate voice guide
```

```
local command      = "httpapi.asp?command=PromptDisable";
local selfId       = fibaro:getSelfId();
local ip           = fibaro:get(selfId, 'IPAddress');
local port         = fibaro:get(selfId, 'TCPPort');
local connection   = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection, command);
local json         = json.decode(response);
```

Seek playback

```
-- Seek 60 seconds backward
```

```
local selfId       = fibaro:getSelfId();
local ip           = fibaro:get(selfId, 'IPAddress');
local port         = fibaro:get(selfId, 'TCPPort');
local connection   = Net.FHttp(ip, port);

local response, status, errorCode = connection.GET(connection,
"httpapi.asp?command=getPlayerStatus");
```

```

local json                = json.decode(response);
local totlen              = tonumber(json.totlen);
local position            = tonumber(json.curpos);

position                  = math.floor((position - 60000) / 1000);
if(position < 0) then position = 0; end

local command             = "httpapi.asp?command=setPlayerCmd:seek:" .. position;
local response, status, errorCode = connection.GET(connection, command);

```

Unsync all slave iEast devices

```

-- Unsync all connected devices (if there are any)

local selfId              = fibaro:getSelfId();
local ip                  = fibaro:get(selfId, 'IPAddress');
local port                = fibaro:get(selfId, 'TCPPort');
local connection          = Net.FHttp(ip, port);

-- Get all slaves
local response, status, errorCode = connection.GET(connection,
"/httpapi.asp?command=multiroom:getSlaveList");
jsonTable = json.decode(response);

for k, v in pairs(jsonTable.slave_list) do
    fibaro:debug("Remove slave with ip: " .. v.ip);
    local response, status, errorCode = connection.GET(connection,
"/httpapi.asp?command=multiroom:SlaveKickout:" .. v.ip);
    fibaro:sleep(500);
end

```

Attach slave device

```

-- Sync other device to current
-- You just need to set up mac address of device
-- which you want to sync with current
local mac                  = "";

local selfId              = fibaro:getSelfId();
local ip                  = fibaro:get(selfId, 'IPAddress');
local port                = fibaro:get(selfId, 'TCPPort');
local connection          = Net.FHttp(ip, port);
local response, status, errorCode = connection.GET(connection,
"/httpapi.asp?command=getStatusEx");

-- This function allows to convert string to HEX
function string.tohex(str)
    return (str:gsub('.', function (c)
        return string.format('%02X', string.byte(c))
    end))
end

-- Get ssid of current device
local json                = json.decode(response);

```

```
ssid                                     = json.ssid:tohex();

-- Get IP of slave device and try to group
for j=1,4 do
    local response, status, errorCode = HC2.POST(HC2, "/api/networkDiscovery/arp", "{}");
    local jsonTable = json.decode(response)

    for i, name in ipairs(jsonTable) do
        if(name['mac'] == mac) then
            local connection = Net.FHttp(name['ip'], 80);
            local response, status, errorCode = connection.GET(connection,
"/httpapi.asp?command=ConnectMasterAp:ssid=" .. ssid ..
":ch=5:auth=OPEN:encry=NONE:pwd=null:chext=1");
            end
        end
    end
end
```