

# **Home Theater PC Media Control System**

Media Control System Overview .....	3
Amplifier Control Client .....	6
Amplifier Control Server .....	7
Universal Remote Client .....	9
Issues and Design Considerations .....	11
Third Party Products .....	12
Software Licensing .....	12
Software Download .....	12

# Media Control System Overview

What are the advantages of installing a Home Theater PC based media control system?

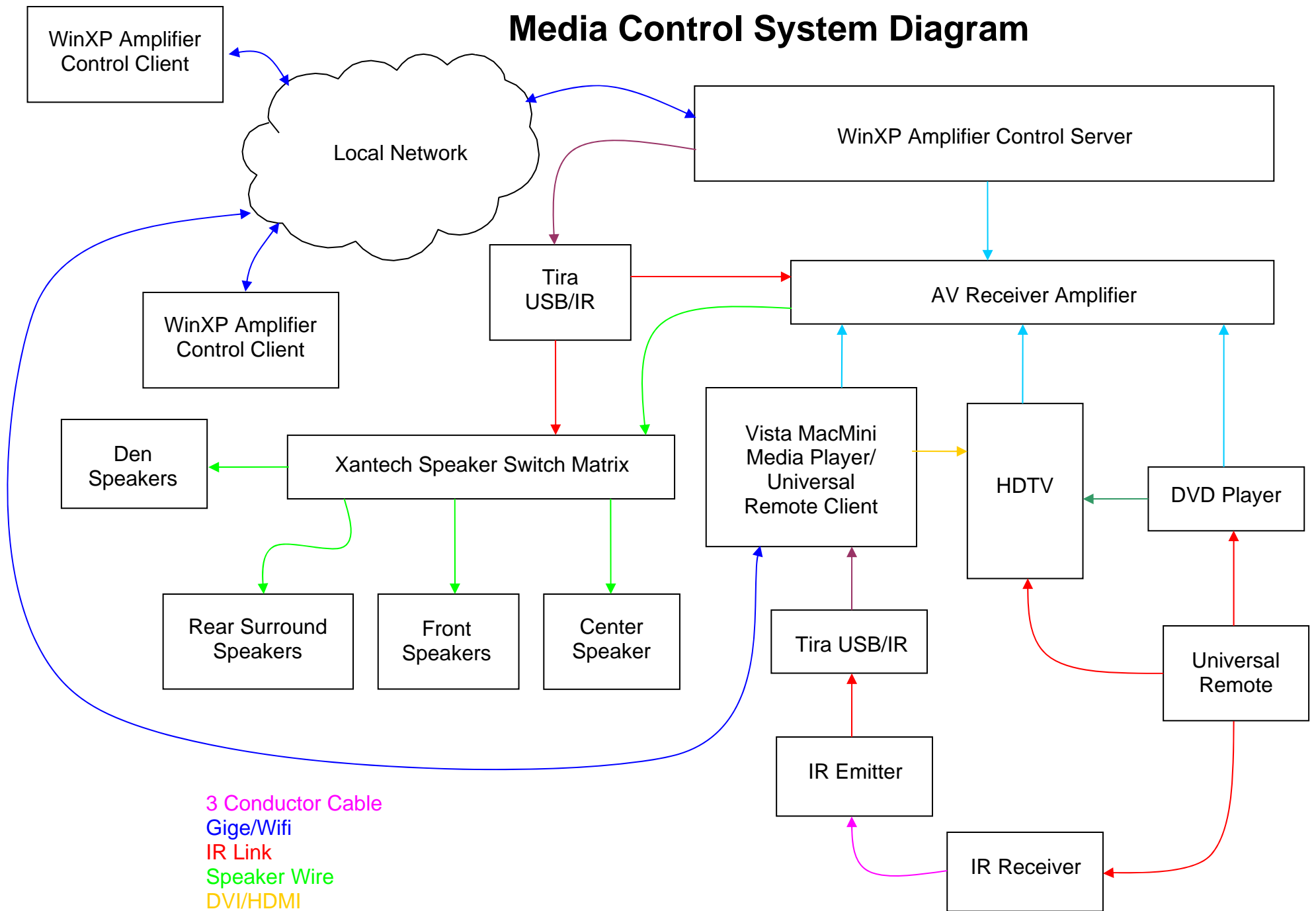
- Play media from any home computer using an existing home theater
- Control your home theater from any computer or using a single universal remote
- Simplify the user interface to play music, television and movies using your home theater

Media content is now available from a multitude of online sources: downloadable movies, Internet radio, mp3 music files, digital cameras, ripped DVDs, and a vast array of web based video content. Given the trend toward online entertainment content, integrating a home theater with the computers in your home will simultaneously broaden choice and enhance your home entertainment experience.

A speaker switch matrix is used for turning speakers on or off in a particular room. Surround sound is great for movies but not necessarily desirable for everyday television. There are occasions when you would like to enjoy music in the den but not in the computer room. For a party or working around the house, you may want to turn speakers on throughout your home. A myriad of combinations all managed from any computer anywhere in the house thanks to wifi networking.

The implementation example for this media control system was written to manage a Yamaha RX-V800 AV Receiver but the software was written to readily adapt to any home theater implementation with infrared remote control. Adapting the software to a particular receiver does require C and Java language programming knowledge. However, the software is generic enough to readily adapt to any system by only modifying the header files that contain the infrared codes. Any variety or arrangement of system components can be managed by simply defining the relevant remote control commands. The software for this system and all the development tools are freely available.

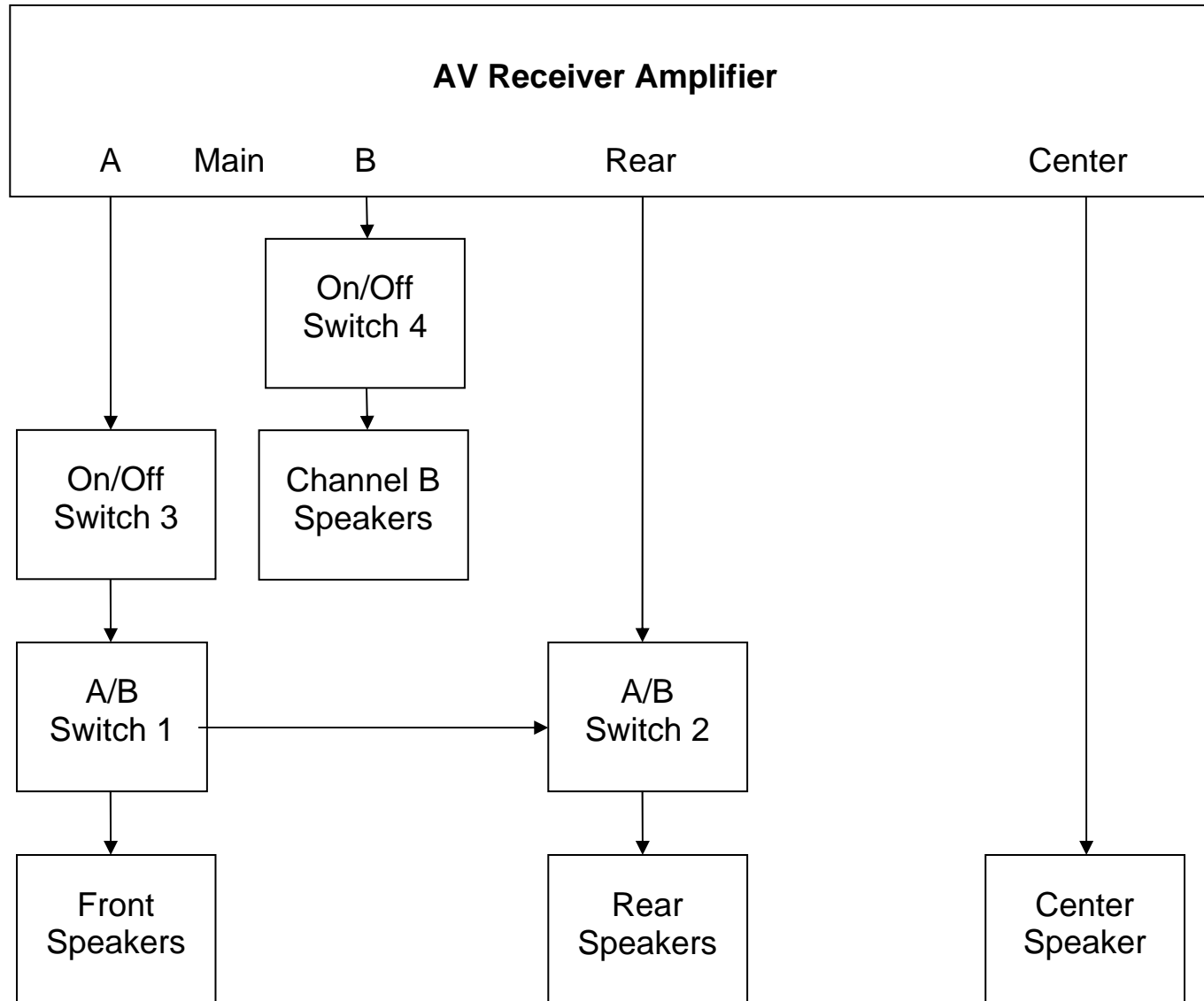
# Media Control System Diagram



3 Conductor Cable  
Gige/Wifi  
IR Link  
Speaker Wire  
DVI/HDMI  
Component Video  
Toslink/Coax S/PDIF  
USB

MacMini is managed remotely using a RealVNC session  
IR Receiver/Emitter extender is the Xantech model 48095DBKIT

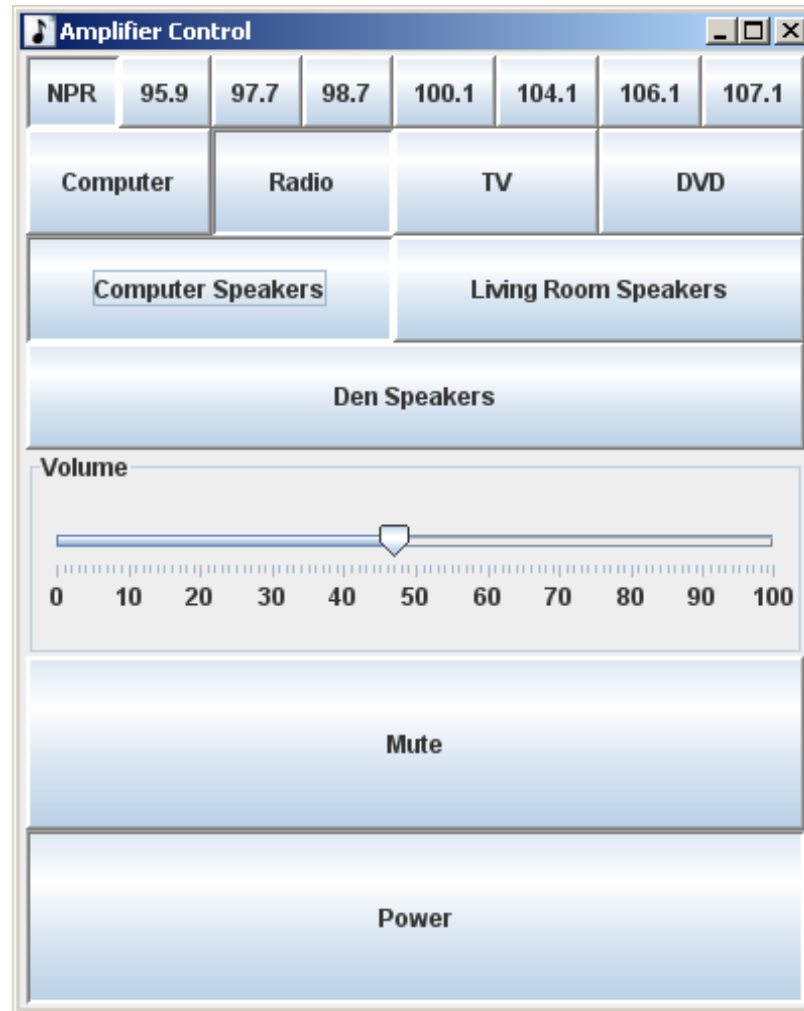
# Xantech Speaker Switch Matrix



On/Off Switch is the Xantech model CC12  
A/B Switch is the Xantech model SR21

# Amplifier Control Client

The following screen capture shows the Java based amplifier control client utility used for the installation example. The buttons shown can be easily reconfigured for a different installation. Pressing any button sends an IP message request to the amplifier control server. The amplifier control server translates the IP message and then transmits the corresponding IR command.



The java based client utility requires the JRE to execute and the SE JDK to build. Run the utility with:

```
javaw -jar acClient.jar
```

or, to see debug messages:

```
java -jar acClient.jar
```

Java is freely available from <http://java.sun.com/javase/downloads/index.jsp>

## Amplifier Control Server

Amplifier control clients send IP messages to a central amplifier control server process called acServer.exe. An IP message such as “power on” is translated using the “irMsg” array found in mcs/h/common.h. The irMsg array contains structures that associate IP message strings to IR commands. For example, the “power on” IP command is defined by:

```
{POWER_ON,          // unique integer
  SZ_POWER_ON,      // unique ip command string sent from a client
  &powerState,       // valid values POWER_ON or POWER_OFF
  sizeof(irPowerOn),
  irPowerOn},        // array of chars passed to the Tira device for transmitting an IR command
```

IR commands for an amplifier are transmitted by a library call to the Tira device with an array of chars argument. Generating the array of chars for desired IR commands is done using the mcs/demo/demo.exe utility. Simply connect a Tira device to a PC USB port, run demo.exe, turn on capture mode, point the device’s IR remote at the Tira receiver and press the remote button you wish to capture. The demo utility will output the IR codes as an array of chars. The following example shows the output generated for defining the irPowerOn char array:

```

> demo.exe 7
Library loaded
Tira activated on com port 7
>a
Playback capture activated
IR Code captured!
DataSize: 163
0x00, 0x00, 0x00, 0x00, 0x08, 0x00, 0x00, 0x00, 0x3E, 0x04,
0x00, 0x00, 0x3E, 0x02, 0x00, 0x00, 0x46, 0x00, 0x00, 0x00,
0xD0, 0x00, 0x00, 0x00, 0x2B, 0x13, 0x00, 0x00, 0x10, 0x01,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xDC, 0x82, 0xAC, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00,
0x00, 0x00, 0x0C, 0x00, 0x00, 0x00, 0x90, 0x65, 0x00, 0x00,
0x03, 0x00, 0x00, 0x00, 0x47, 0x00, 0x00, 0x00, 0xFF, 0xFF,
0xFF, 0xFF, 0x00, 0x01, 0x02, 0x02, 0x02, 0x03, 0x02, 0x02,
0x02, 0x03, 0x02, 0x03, 0x02, 0x03, 0x02, 0x03, 0x02, 0x02,
0x02, 0x03, 0x02, 0x02, 0x02, 0x03, 0x02, 0x02, 0x02, 0x02,
0x02, 0x02, 0x02, 0x02, 0x02, 0x03, 0x02, 0x03, 0x02, 0x02,
0x02, 0x02, 0x02, 0x03, 0x02, 0x03, 0x02, 0x03, 0x02, 0x04,
0x00, 0x05, 0x02
Capture deactivated

```



# Universal Remote Client

The Mac Mini runs a dedicated amplifier control client called rcClient.exe that translates IR commands coming from a universal remote into IP messages for the amplifier control server. For example, when the television is turned on via the remote, the audio amplifier turns on simultaneously. Similar IR commands such as volume and mute are translated and forwarded to the amplifier control server.

The rcClient utility receives IR commands from the Tira and encodes them as a six byte string. These strings and subsequent translation actions are printed as text output e.g.:

```
c:\utils\win32\startup
> rcClient.exe
Library loaded
Tira activated on com port 5
Callback capture activated
type 'q' to quit
>match found: remote tv power toggle
A28A808A0003
```

The six byte string is matched to an array of possible matches in mcs/remote/rcClient.h, i.e.:

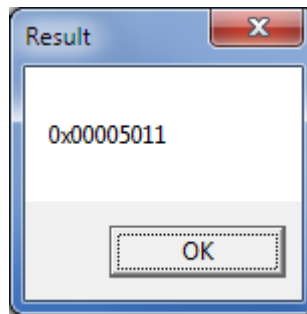
```
const char * irRemoteTvPowerToggle[] = {
    "A28A808A0003", // six byte encoding of an IR cmd from a universal remote
    "61B583B503BC" // other possible encoding for the same command
};
```

which is a member of a structure in the translation[] entry:

```
{SZ_REMOTE_TV_POWER_TOGGLE, // outgoing string for the amplifier control server
  REMOTE_TV_POWER_TOGGLE,
  0,
  sizeof(irRemoteTvPowerToggle)/sizeof(char *),
  irRemoteTvPowerToggle} // incoming encoded IR cmd from a universal remote
```

Translation array entries are used to associate a received IR message to a particular case statement. In this case, matching `irRemoteTvPowerToggle` executes the `REMOTE_TV_POWER_TOGGLE` case to send the corresponding IP message string to the amplifier control server to turn on the amplifier. Unknown IR commands are also printed to the debug console as a six byte encoded string to facilitate the creation of new translations.

The `rcClient.exe` process also accepts IR remote commands to synthesize a corresponding keyboard shortcut to control a running media player. Pressing remote buttons such as play, pause, stop, ff, rewind etc., synthesize pressing the corresponding media player keyboard shortcut. The included `vkcode.exe` (virtual key code) utility will generate the hex codes used to translate an IR message to a keyboard shortcut. For example, a `<CTRL>-P` will pop up a message box:



`0x5011` is added to the `rcClient.h` header as a `#define` becomes part of an IR command to keyboard shortcut translation[] entry:

```
#define WMC_PLAY_PAUSE 0x5011 // remote pause -> windows media center Ctrl-P

{SZ_REMOTE_TV_PAUSE,
  REMOTE_TV_PAUSE,
  WMC_PLAY_PAUSE, // outgoing virtual key code used to synthesize a keyboard shortcut
  sizeof(irRemoteTvPause)/sizeof(char *),
  irRemoteTvPause}, // incoming encoded IR cmd from a universal remote
```

# **Issues and Design Considerations**

## **Vista Monitor Dependency**

The 2007/2008 Mac Mini would not boot Vista connected to my HDTV display. I can only speculate there was some sort of PnP issue because Vista would boot fine connected to a VGA monitor. The 2009 Mac Mini did not have the problem.

## **Universal Remote Client/Amplifier Control Server Consolidation**

Consolidating the functions of the Universal Remote Client and Amplifier Control Server to use a single Tira receiver/transmitter is an optimal arrangement. Unfortunately, my installation precludes the possibility of colocating the amplifier and the Mac Mini. Nonetheless, the two modules should readily combine into a single executable by consolidating their main routines and linking the modules.

## Third Party Products

The following are several of the links to different companies that provided the products that made this project a reality:

<http://www.xantech.com> – Xantech IR controlled relays and extenders

<http://www.cygwin.com> – free GCC compiler for win32

<http://java.sun.com/javase/downloads/index.jsp> – free Java development kit and runtime environment software

<http://home-electro.com> – the Tira PC USB interface IR receiver/transmitter

<http://www.apple.com/macmini> - Mac Mini

<http://www.slysoft.com/en> – DVD ripper software

<http://realvnc.com> – free Virtual Network Computing (VNC) software

<http://www.videolan.org> – free media player software

<http://www.amazon.com> – search for an “HDMI-M to DVI-M Cable, 3 FT / 1 M”

## Software Licensing

Software user license is granted under the terms of the Gnu General Public License Version 3 (GPLv3):

<http://www.gnu.org/licenses/gpl-3.0.txt>

## Software Download

<https://drive.google.com/drive/folders/0B6HZtsotckwSQ3lzSVRMS3Zoa1k>