Home Theater PC
Media Control System

Media Control System Overview

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

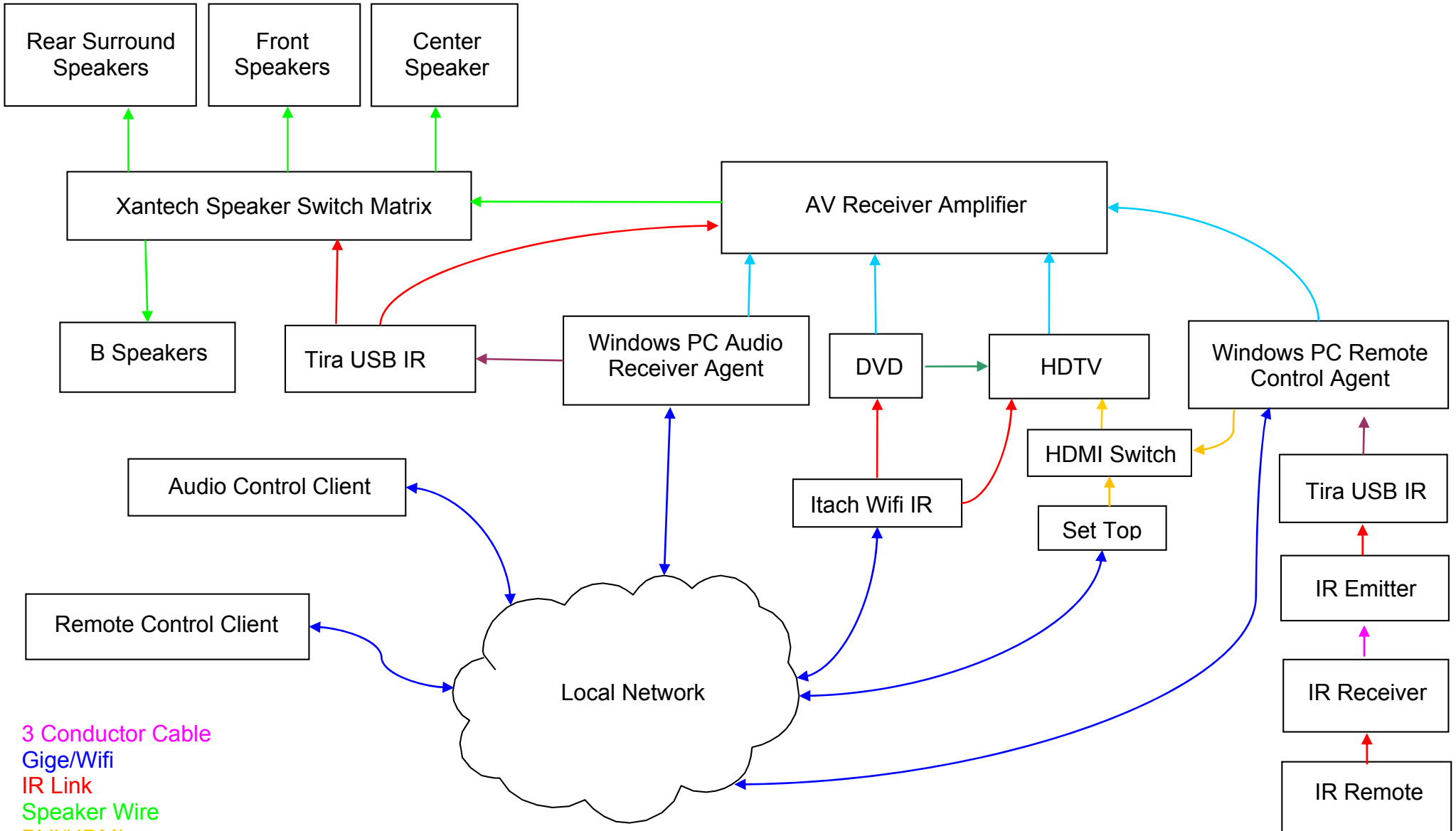
- Play media from any home computer using an existing home theater
- Control your home theater from any computer or using a single universal remote
- Simplify the user interface to play music, television and movies using your home theater

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.

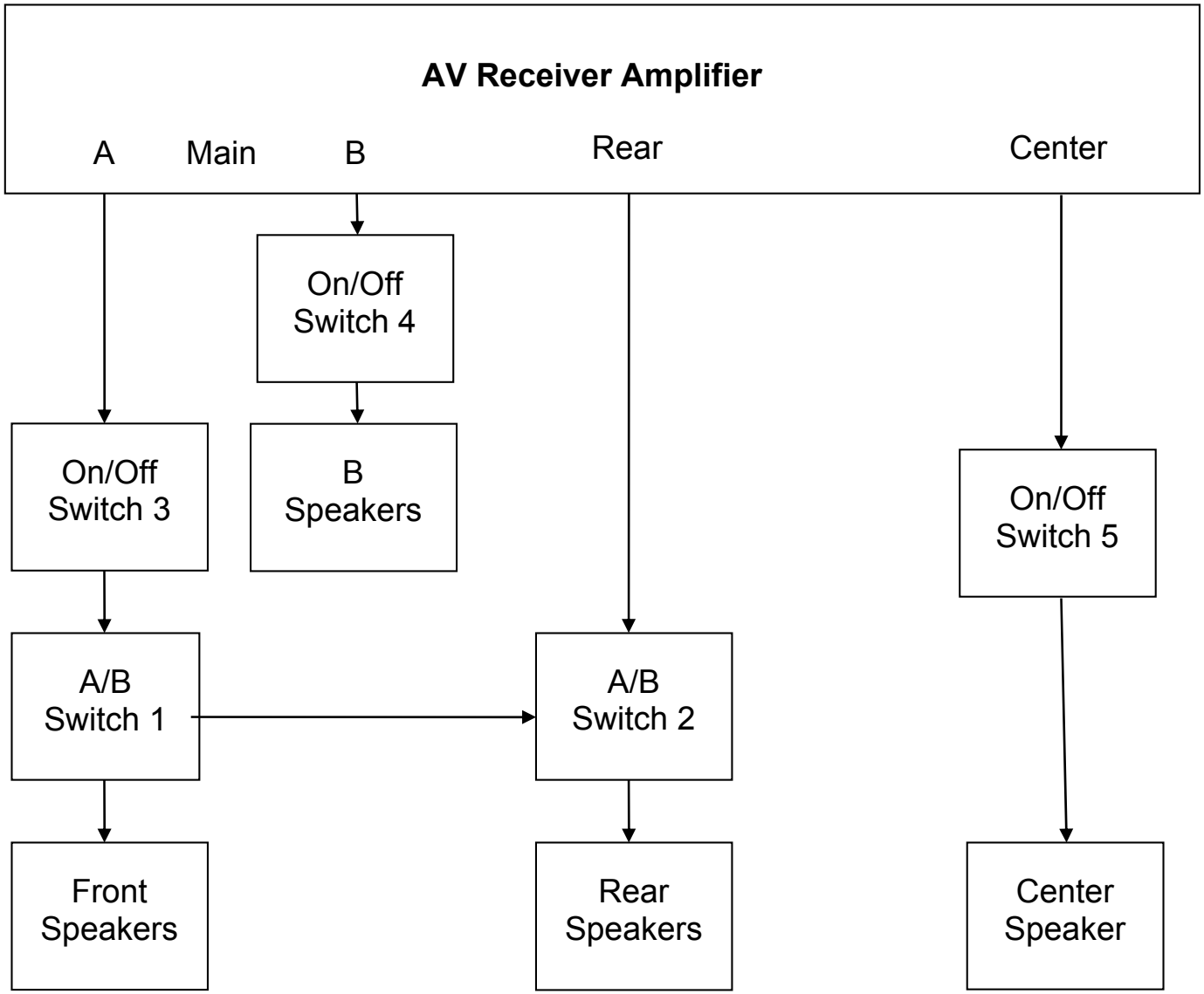
The implementation example for this media control system was written to manage a Yamaha RX-V800 AV Receiver but the software was written to readily adapt to any home theater implementation with infrared remote control. Adapting the software to a particular receiver does require C and Java language programming 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



IR Receiver/Emitter is the Xantech model 48095DBKIT
Itach Wifi IR is the Global Cache model WF2IR

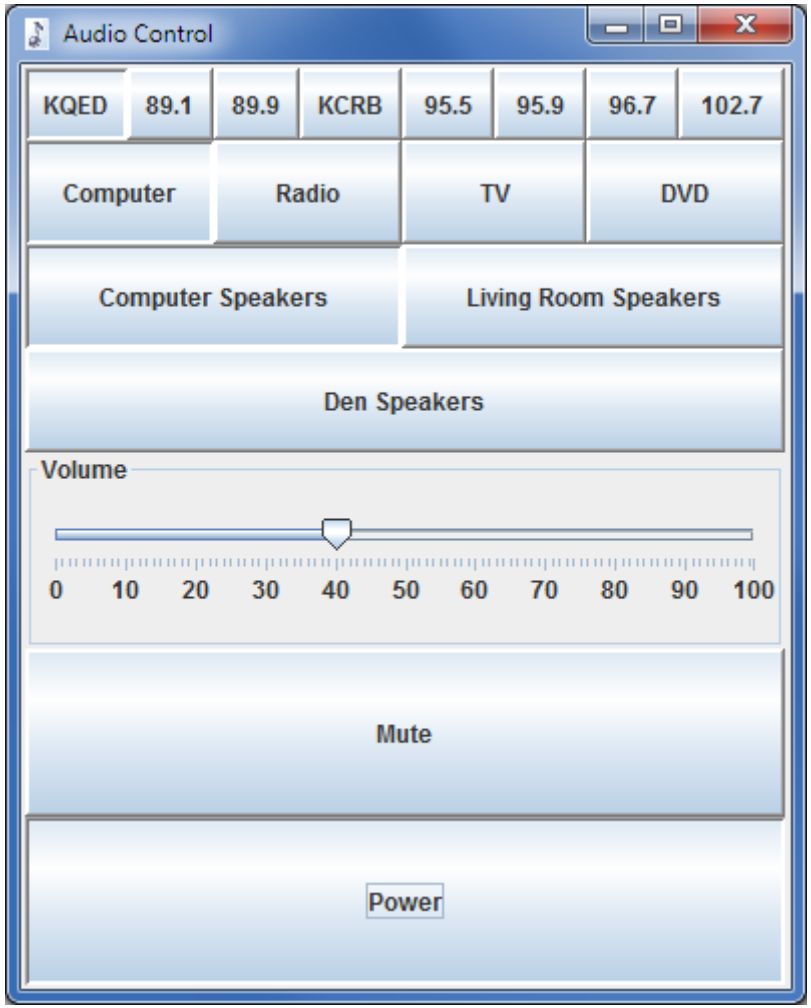
Xantech Speaker Switch Matrix



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

Audio Control Client

The following screen capture shows the Java based audio 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 audio receiver agent. The audio receiver agent 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

Audio Receiver Agent

Audio control clients send IP messages to a central audio receiver agent process called arAgent.exe. An IP message such as “power on” is translated using the “arCmd” array found in mcs/h/common.cpp. The arCmd 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/util/tira/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, 0x03, 0x02, 0x03, 0x02, 0x03, 0x02, 0x02, 0x02, 0x02,
0x02, 0x02, 0x02, 0x02, 0x02, 0x03, 0x02, 0x02, 0x02, 0x02,
0x02, 0x02, 0x02, 0x03, 0x02, 0x03, 0x02, 0x03, 0x02, 0x04,
0x00, 0x05, 0x02
Capture deactivated
```

Remote Control Agent

A Home Theater PC like a Mac Mini displays to a TV and runs a remote control agent called rcAgent.exe. The rcAgent.exe process accepts IR commands coming from an IR remote or text commands from a remote control client and translates them into audio receiver agent or iTach device commands. For example, when a user presses power on the universal remote, the television and the audio receiver turn on simultaneously.

The rcAgent utility receives IR commands from the Tira device 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
> rcAgentConsole.exe
Library loaded
Tira activated on com port 5
Callback capture activated
type 'q' to quit
>match found: ir tv power
A28A808A0003
```

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

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

which is contained in an irRx structure:

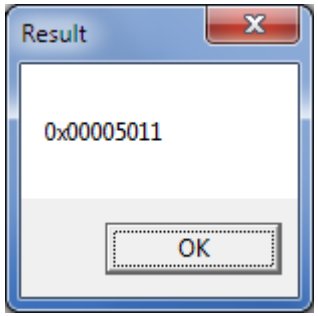
```
static struct irRx irRxTvPower = {
    sizeof(tvPower)/sizeof(char *),
    tvPower,
    SZ_IR_TV_POWER};  // command string and debug description
```

An rcCmd array of structures associates received commands to desired actions.

```
{
   IR_IGNORE,
   SZ_IR_IGNORE,
   &irRxTvPower,
   NULL,
   NULL
},
```

In this case, matching irRxTvPower code informs the Remote Control Client the TV has been successfully turned on by the iTach device. The iTach device translates an IP message into an IR command.

The remote control agent process also accepts commands to synthesize a corresponding keyboard shortcut. Commands such as play, pause, stop, ff, rewind etc., synthesize pressing a corresponding media player keyboard shortcut. The included vkcode.exe (virtual key code) utility will generate the hex codes used to define a keyboard shortcut. Running vkcode.exe and pressing <CTRL>-P will pop up a message box:



0x5011 is added to the rcClient.h header as a #define that becomes part of a `shortcut` structure:

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

```
static struct shortcut scPlayPause = {
   {
      KBD_SPACE,
      KBD_SPACE,
      KBD_CTRL_P,
      KBD_SPACE
   },
   {
      SZ_KBD_SPACE,
      SZ_KBD_SPACE,
      SZ_KBD_CTRL_P,
      SZ_KBD_SPACE
   }
};
```

which is also an element in the array of rcCmd structures:

```
   {
      PLAY,
      SZ_PLAY,
      &irRxVcrPlay,
      &scPlayPause,
      irTxPlay
   },
```

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.globalcache.com/products/itach/models2 - Wifi to IR control device

http://cygwin.com – free POSIX compatible GCC compiler

http://mingw.org – free windows compatible GCC compiler

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.hdhomerun.com – Prime TV Tuner used to record shows using Windows Media Center

http://www.howtogeek.com/258695/how-to-install-windows-media-center-on-windows-10 – WMC install instructions

http://shark007.net/ – codecs needed to run WMC on windows 8 and later (use Shark007 SUGGESTED settings)

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

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

http://www.monoprice.com – Blackbird 4k Pro 3x1 HDMI Switch with HDCP 2.2 Support

Compiler Setup

C++ compilers are used from Cygwin and MinGW.

Cygwin:
Download and run their 32 bit setup.exe utility. When you get to the “Select Packages” page, Search for g++ and add the appropriate packages. Repeat by doing a Search for gcc and adding relevant packages.

MinGW:
Download and run the mingw-get-setup.exe. Next, open a command-line shell, cd to the mingw bin directory and run:

mingw-get.exe install mingw32-base
mingw-get.exe install mingw32-gcc-g++

The Cygwin and MinGW bin directories must be added to the PATH environment variable.

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://github.com/jhmpub/mcs.git