

RELAY ATTENUATOR

USER MANUAL

Table of Contents

Introduction	3
Setup	3
Software.....	3
Installation	3
WiringPi.....	3
LIRC	4
r_attenu	5
Telnet	6
Modes of Control	6
Audio Player control	6
IR remote control.....	7
Push-button switch	8
LIRC config files	8
hardware.conf file.....	8
lircd.conf file	9
lircrc files.....	10

Introduction

Relay Attenuator (RA) is a hardware volume control. It takes input from the DAC shields and attenuates the output by relay switching.

Setup

Setup of RA involves SBC, DAC, Relay Attenuator and Amplifier shield.

- SBC can be RPi or Sparky boards.
- DAC shields can be Piano or Boss DAC shields.
- Amplifiers can be Volt AMP or Volt+ AMP.

RA gets sandwiched between DAC and Amplifier Shields.

Software

Relay Attenuator can be controlled using “r_attenu” application in three modes.

1. Audio Player control (using RA Client application, “r_attenuc”)
2. IR remote control
3. Push-button switch on RA shield

r_attenu software requires 2 packages to be installed; WiringPi & LIRC.

Installation

WiringPi

ON RASPBERRY PI

This package is mandatory for r_attenu to function. Check if wiringPi package is already installed, using the below command (should be V2.32 or greater)

```
$ sudo apt-cache policy wiringPi
```

Execute the below command to install wiringPi, if not installed.

```
$sudo apt-get install wiringpi
```

Enable i2c using raspi-config & test hardware using i2cdetect. Please refer this link for more details;

<http://www.raspberrypi-spy.co.uk/2014/11/enabling-the-i2c-interface-on-the-raspberry-pi/>

ON SPARKY

Download wiringSparky package from <https://github.com/sparkysbc/WiringSparky.git> and follow the instructions

```
$git clone https://github.com/sparkysbc/WiringSparky.git  
$cd WiringSparky/WiringSparky  
$sudo ./build
```

Add below line to /etc/modules

```
i2c_dev
```

LIRC

ON RASPBERRY PI

This package is required if IR remote control is necessary. Check if LIRC package is already installed, using the below command

```
$ sudo apt-cache policy lirc
```

Execute the below command to install LIRC, if not installed.

```
$sudo apt-get install lirc
```

In /boot/config.txt file, update as shown below

```
dtoverlay=lirc-rpi,gpio_in_pin=17
```

See sections “LIRC config files” for details on config files.

ON SPARKY

LIRC package is required if IR remote control is necessary. Check if LIRC package is already installed, using the below command

```
$ sudo apt-cache policy lirc
```

Execute the below command to install LIRC, if not installed.

```
$sudo apt-get install lirc
```

If “libasound2” dependence error is displayed, try installing libasound2 using

```
$sudo apt-get install libasound2
```

Add these lines to /etc/modules file;

```
lirc_dev
```

```
lirc_sparky
```

See sections “LIRC config files” for details on config files.

r_attenu

Download the corresponding tar file (r_attenu_rpi.tgz or r_attenu_sparky.tgz) from
https://github.com/sparkysbc/downloads/raw/master/r_attenu_rpi.tgz or

https://github.com/sparkysbc/downloads/raw/master/r_attenu_sparky.tgz and extract under “/”.

Tar file contains the r_attenu (server software), r_attenuc (Client software), lircd.conf, lircrc & hardware.conf files. (Note: these config file are written to work with squeezelite player).

Start r_attenu program with –d option. Make sure the r_attenu program is always running.

```
$ sudo r_attenu -d
```

OR

Edit /etc/rc.local file and added below line to start the program automatically on boot up.

```
sudo /usr/bin/r_attenu -d > /dev/null 2>&1
```

Syntax for r_attenu:

Usage: r_attenu [options] [config_file]

-d --daemon	Run in background
-h --help	Display usage summary
-v --version	Display version
-l --withoutLIRC	Program will work without IR control
-n --name=progname	Use this program name for lircrc matching
-c --lircdconfig=configfile	LIRCD config file

Telnet

LIRC works with telnet for Squeezelite. Install telnet using below cmd;

```
$sudo apt-get install expect telnet
```

Modes of Control

Audio Player control

Changing volume in the player GUI can control the Relay Attenuator, with the help of r_attenu application.

r_attenu communicates with r_attenu server, so make sure r_attenu is executed.

Available options are:

Usage: r_attenuc [options]

-h Display usage summary

-c Command to execute

Commands:

GET_VOLUME	Get volume
------------	------------

SET_VOLUME=[value]	Set volume. value = 0 to 63
--------------------	-----------------------------

GET_MUTE	Get mute status
----------	-----------------

SET_MUTE=[value]	Set mute. value = 0/1 (0=unmute 1=mute)
------------------	---

Example:

To set volume to level 50

```
$r_attenuc -c SET_VOLUME=50
```

To get current volume level

```
$r_attenuc -c GET_VOLUME
```

IR remote control

r_attenu program works with LIRC library to control the Relay attenuator with a remote control. Lircrc config provided, works for squeezelite player. Similar config can be written for other players.

Following are the pin details on the Relay Attenuator for connecting the IR senor (HS0038).

IR-pins	J7 16-pin connector
----------------	----------------------------

Pin 1 – OUT	Pin 8
-------------	-------

Pin 2 - GND	pin 16
-------------	--------

Pin 3 - VCC pin 2

For more details, refer to the technical manual.

Push-button switch

Relay Attenuator shield provides option to connect four push-button switches (not included with shield) which can be placed on the front panel of the casing. Switch functions are Volume increase, Volume decrease, Mute & Play/pause.

Push-button switches are connected to connector J10 on the relay attenuator shield. For more details on the connector, please refer to technical manual.

LIRC config files

hardware.conf file

Hardware.conf file defines the devices & modules used and the arguments required for lircd

Location: /etc/lirc/hardware.conf

ON RASPBERRY PI

```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS="--uinput --allow-simulate"

#Don't start lircmd even if there seems to be a good config file
#START_LIRCMD=false

#Don't start irexec, even if a good config file seems to exist.
#START_IREXEC=false

#Try to load appropriate kernel modules
LOAD_MODULES=true

# Run "lircd --driver=help" for a list of supported drivers.
DRIVER="default"
```

```
# usually /dev/lirc0 is the correct setting for systems using udev
#DEVICE="/usr/share/doc/lirc/README.Debian"
DEVICE="/dev/lirc0"
MODULES="lirc_rpi"

# Default configuration files for your hardware if any
LIRCD_CONF="/etc/lirc/lircd.conf"
LIRCMD_CONF=""
```

ON SPARKY

```
# /etc/lirc/hardware.conf
#
# Arguments which will be used when launching lircd
LIRCD_ARGS="--uinput --allow-simulate"
#LIRCD_ARGS=""
#Don't start lircmd even if there seems to be a good config file
#START_LIRCMD=false

#Don't start irexec, even if a good config file seems to exist.
#START_IRExec=false

#Try to load appropriate kernel modules
LOAD_MODULES=true

# Run "lircd --driver=help" for a list of supported drivers.
DRIVER="default"
# usually /dev/lirc0 is the correct setting for systems using udev
DEVICE="/dev/lirc0"
MODULES="lirc_sparky"

# Default configuration files for your hardware if any
LIRCD_CONF="/etc/lirc/lircd.conf"
LIRCMD_CONF=""
```

lircd.conf file

lircd.conf file can be downloaded from <http://lirc.sourceforge.net/remotes/> for individual remotes depending on the make & model. Make sure that copied file is renamed to “lircd.conf”.

OR

It can be generated for individual remotes using “irrecord” cmd. Example below shows the config file for Allo remote.

Location: /etc/lirc/lircd.conf

```

begin remote

name /etc/lirc/lircd.conf
bits 16
flags SPACE_ENC|CONST_LENGTH
eps 30
aeps 100

header 9087 4492
one 584 1686
zero 584 558
ptrail 597
repeat 9086 2221
pre_data_bits 16
pre_data 0xC77
gap 108399
toggle_bit_mask 0x0

begin codes
KEY_PREVIOUS 0x609F
KEY_NEXT 0xE01F
KEY_PLAYPAUSE 0x50AF
KEY_VOLUMEUP 0x10EF
KEY_VOLUMEDOWN 0x906F
end codes

end remote

```

lircrc files

This file defines the actions for each key defined in lircd.config & push-button switches of relay attenuator (BUTTON1, BUTTON2, BUTTON3 and BUTTON4). For more details about the field in lircrc file, please refer to <http://www.lirc.org/html/configure.html>. Along with the standard implementation of fields, “config” field can handle “hardware_control” option, which instructs r_attenu to handle the corresponding button’s action. The 4 push-buttons (keys) on the relay attenuator is mapped as follows:

Push-buttons	Function
BUTTON1	Mute

BUTTON2	Play/Pause
BUTTON3	Volume down
BUTTON4	Volume up

Location: /etc/lirc/lircrc

```

begin
    button = KEY_POWER
    prog = r_attenu
    config = sudo echo "power" | telnet 127.0.0.1 9090
end

begin
    button = BUTTON1
    prog = r_attenu
    config = hardware_control
    #config = sudo echo "mute" | telnet 127.0.0.1 9090
end

begin
    button = BUTTON2
    prog = r_attenu
    config = KEY_PAUSE
    #config = sudo echo "pause" | telnet 127.0.0.1 9090
end

begin
    button = BUTTON3
    prog = r_attenu
    config = hardware_control
    #config = sudo echo "mixer volume -1" | telnet 127.0.0.1 9090
end

begin
    button = BUTTON4
    prog = r_attenu
    config = hardware_control
    #config = sudo echo "mixer volume +1" | telnet 127.0.0.1 9090
end

```