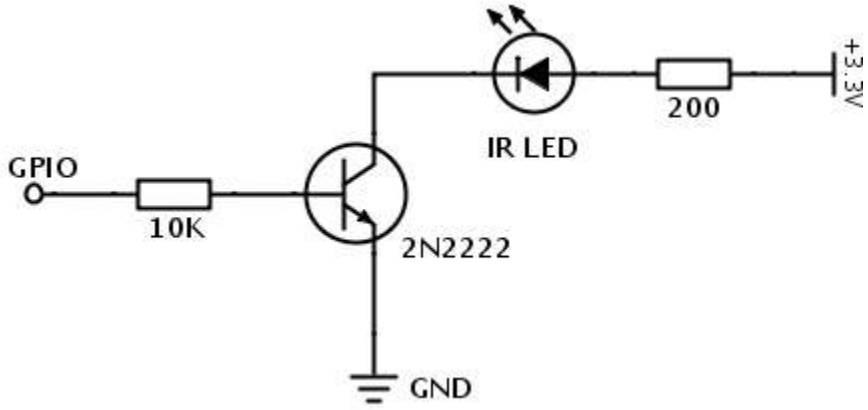
Lirc Setup

This setup was performed on a Raspberry PI 3 running the 11 / 25 / 2016 Jessie Lite version of Raspbian.

```
Linux xxxxxx 4.4.34-v7+ #930 SMP Wed Nov 23 15:20:41 GMT 2016 armv7l
GNU/Linux
```

lirc was setup for sending ir commands to a JVC receiver. Receiving ir commands was not required. The version in the repo was several revisions old and therefore I chose to build lirc from source.

The ir led was setup using a npn 2222 transitory wired to 5v, G and GPIO 22 with a 10K and a 220 ohm ohm resistors to complete the circuit.



Circuit diagram by Dmitri Popov: http://www.raspberry-pi-geek.com/Archive/2015/10/Raspberry-Pi-IR-remote

Install lircd using apt-get install or if a newer version is desired build from source.

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

To build from source:

Download the source. As of 12 / 23 / 2016 the source is found on Sourceforge under files in the lirc project area. The source file is lirc-0.9.4c.tar.gz.

Install dependancies:

```
sudo apt-get install autoconf automake autopoint autotools-dev bsd-mailx
dctrl-tools debhelper    debian-keyring debootstrap devscripts dh-autoreconf
dh-python    dh-strip-nondeterminism diffstat distro-info-data dput equivs
exim4-base    exim4-config exim4-daemon-light gettext gettext-base hardening-
includes    intltool-debian libapt-pkg-perl libarchive-zip-perl libasprintf-
dev    libasprintf0c2 libclass-accessor-perl libclass-inspector-perl
libclone-perl    libcommon-sense-perl libconvert-binhex-perl libcroco3
libcrypt-ssleay-perl    libdigest-hmac-perl libdistro-info-perl libemail-
valid-perl liberror-perl    libexporter-lite-perl libfile-
stripnondeterminism-perl libgettextpo-dev    libgettextpo0 libio-pty-perl
libio-sessiondata-perl libio-socket-inet6-perl    libio-string-perl libio-
stringy-perl libipc-run-perl libjson-perl    libjson-xs-perl liblist-
moreutils-perl liblockfile-bin liblockfile1    libltdl-dev libltdl7 libmail-
```

```
sendmail-perl libmime-tools-perl libmpdec2    libnet-dns-perl libnet-domain-
tld-perl libnet-ip-perl libossp-uuid-perl    libossp-uuid16 libparse-
debcontrol-perl libparse-debianchangelog-perl    libperlio-gzip-perl
libpython3-stdlib libpython3.4-minimal    libpython3.4-stdlib libsigsegv2
libsoap-lite-perl libsocket6-perl    libsub-name-perl libsys-hostname-long-
perl libtask-weaken-perl    libtext-levenshtein-perl libtool libunistring0
libxmlrpc-lite-perl lintian    m4 patchutils pbuilder po-debconf python3
python3-apt python3-chardet    python3-debian python3-magic python3-minimal
python3-pkg-resources    python3-six python3.4 python3.4-minimal t1utils
wdiff
```

Run ./configure as shown below. This will configure the install the same way the armhf Debian package is.

```
./configure --build=arm-linux-gnueabihf --prefix=/usr --
includedir=\${prefix}/include --mandir=\${prefix}/share/man --
infodir=\${prefix}/share/info --sysconfdir=/etc --localstatedir=/var --
disable-silent-rules --libdir=\${prefix}/lib/arm-linux-gnueabihf --
libexecdir=\${prefix}/lib/arm-linux-gnueabihf --disable-maintainer-mode --
disable-dependency-tracking HAVE_UINPUT=1
```

Run make and sudo make install.

```
make
sudo make install.
```

The lirc build is complete.

<u>lirc Setup:</u>

Ensure that the following lines are in your /boot/config.txt and not commented out. The debug=on is optional but helps with debugging.

```
# Uncomment this to enable the lirc-rpi module
dtoverlay=lirc-rpi,gpio_out_pin=22,gpio_in_pin=23,debug=on
```

Add lirc_rpi.conf to /etc/modprobe.d:

```
options lirc_rpi gpio_in_pin=23 gpio_out_pin=22
```

Add lirc.conf to /etc/modules-load.d:

```
lirc_dev
lirc_rpi
```

Update the configuration files in /etc/lirc and add jbc_rm-sr309u.conf to /etc/lirc/lircd.conf.d.

Remove lirc from /etc/init.d and enable system scripts:

```
sudo systemctl enable lirc-setup
sudo systemctl enable lircd.services
sudo systemctl enable lircd.socket
```

For a more secure setup lirch should run under a user other than root. Setup lircd user

```
sudo useradd -M --shell /bin/false lircd
sudo usermod -L lircd
sudo adduser lircd video
```

Ensure that in the lircd_options.conf file the effective_user is set to the user created above.

If you wish to provide tcp/ip access to lirc you will need to make sure that the listen option is set in the lircd_options.conf file.

Reboot and check the following for errors:

```
dmesg
/var/log/syslog
sudo systemctl status lirc-setup
sudo systemctl status lirc.service
sudo systemctl status lirc.socket
```

Testing

Test JCV remote with the command below. The **–count=2 is critical** for JVC as if you just send the command once it will be ignored.

```
irsend --count=2 SEND_ONCE RM-SR3 09U power_audio
```

Troubleshooting

To run lircd from the command line use the command below:

```
sudo lircd --nodaemon --driver=de fault --device=/dev/lirc0 --uinput
```

If you run into problems with your driver not being loadable check that your plugins directory is set right. This is done on the command line using the –plugindir option or can be set in the lirc_options.conf file. Using the ./configure command string above will put the plugins in /usr/lib/arm-linux-gnueabihf/lirc/plugins.

Sources

Raspberry Pi IR remote by Dmitri Popov: http://www.raspberry-pi-geek.com/Archive/2015/10/Raspberry-Pi-IR-remote

Setting Up LIRC on the RaspberryPi by Alex Bain: http://alexba.in/blog/2013/01/06/setting-up-lirc-on-the-raspberrypi/

Lirc.org website: http://www.lirc.org/

Appendix Files

cat hardware.conf

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

# 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_rpi"

# Default configuration files for your hardware if any
LIRCD_CONF=""
LIRCMD_CONF=""
########################################################
```

cat lircd.conf

```
# Populated config files can be found at http://sf.net/p/lirc-remotes. The
# irdb-get(1) and lirc-setup(1) tools can be used to search and download
# config files.
#
# From 0.9.2 config files could just be dropped as-is in the lircd.conf.d
# directory and be included by this file.

include "lircd.conf.d/*.conf"
```

cat lircmd.conf

```
# Populated config files can be http://sf.net/p/lirc-remotes. The
# irdb-get(1) and lirc-setup(1) tools can be used to search and download
# config files.
```

cat lirc_options.conf

```
# These are the default options to lircd, if installed as
# /etc/lirc/lirc_options.conf. See the lircd(8) and lircmd(8)
# manpages for info on the different options.
```

```
[lircd]
nodaemon        = False
driver          = default
device          = /dev/lirc0
output          = /var/run/lirc/lircd
pidfile         = /var/run/lirc/lircd.pid
plugindir       = /usr/lib/arm-linux-gnueabihf/lirc/plugins
permission      = 666
allow-simulate  = No
repeat-max      = 600
effective-user  = lircd
listen          = 8765
#connect        = host[:port]
#debug          = 6
#uinput         = ...
#release        = ...
#logfile        = ...

[lircmd]
uinput          = False
nodaemon        = False

[modprobe]
#modules        = [lircd_dev, lirc_sir...]
```

cat jvc_rm-sr309u.conf

```
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
#
# this config file was automatically generated
# using lirc-0.9.0-pre1(default) on Thu Dec 24 21:34:43 2015
#
# contributed by
#
# brand:                       jvc.codes
# model no. of remote control:
# devices being controlled by this remote:
#

begin remote

  name  RM-SR309U
  bits           16
  flags SPACE_ENC
  eps            30
  aeps          100

  one           582  1524
  zero          582   470
  ptrail        582
  gap          20485
  toggle_bit_mask 0x0
```

```
    begin codes
        power_audio             0xC5E8
        power_vcr               0xC2D0
        power_tv                0xC0E8
        source_phono            0xC53C
        source_cd               0xC5BC
        source_tape1            0xC5FC
        source_tape2            0xC5E0
        source_fm               0xC510
        source_am               0xC590
        vcr_play                0xC230
        vcr_stop                0xC2C0
        vcr_pause               0xC2B0
        vcr_rewind              0xC2E0
        vcr_forward             0xC260
        tape_play               0xC130
        tape_prev               0xC118
        tape_next               0xC198
        tape_stop               0xC1C0
        tape_pause              0xC1B0
        tape_rewind             0xC1E0
        tape_forward            0xC160
        cd_play                 0xCD9E
        cd_prev                 0xCD5E
        cd_next                 0xCDDE
        cd_stop                 0xCD3E
        cd_pause                0xCDBE
        cd_rewind               0xCD7E
        cd_forward              0xCD1E
        preset_minus            0xC518
        preset_plus             0xC598
        fade                    0xC538
        volume_down             0xC5F8
        volume_up              0xC578
    end codes

end remote
```