LIRC

LIRC is the program that encode and decode the IR signals of the common infrared remote (TV, HI-FI, AV-receiver, etc...), while lirc_web is a web interface written by alexbain that allows you to send IR command without dealing with a terminal.

In d-diot an IR receiver and IR led are connected directly to GPIO pin 22 and 12 of the Raspberry Pi.

Configure LIRC: d-diot images v.1.0 and v.1.1

With Raspbian Stretch a lot has been changed in the configuration of LIRC, but here you can find an updated and detailed guide that explains how to use LIRC with in a Raspberry Pi.

To configure LIRC in d-diot first make a backup of the /etc/modules file and then edit it.

```
pi@d-diot:~ $ cd /etc
pi@d-diot:~ $ sudo cp modules  modules.bak
pi@d-diot:~ $ sudo nano /etc/modules
```

Add the following lines at the bottom of the file

/etc/modules

```
lirc_dev
lirc_rpi gpio_in_pin=22 gpio_out_pin=12
```

Exit (CTRL+x) and save the changes (y).

A fresh install of LIRC does not contain a hardware configuration file so you have to create it. Create the file '/etc/lirc/hardware.conf'

```
pi@d-diot:~ $ sudo nano /etc/lirc/hardware.conf
```

with the following content

/etc/lirc/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=""
############################################################
```

Exit (CTRL+x) and save the changes (y).

Next we need to update the '/etc/lirc/lirc_options.conf' file, but as always, make a backup first, just in case.

```
pi@d-diot:~ $ cd /etc/lirc
pi@d-diot:~ $ sudo cp lirc_options.conf lirc_options.conf.bak
```

Update line 11 of lirc_options.conf from driver = devinput to **driver=default**

```
pi@d-diot:~ $ sudo nano lirc_options.conf
```

Exit (CTRL+x) and save the changes (y).

/etc/lirc/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.
#
# Some tools including mode2 and irw uses values such as
# driver, device, plugindir and loglevel as fallback values
# in not defined elsewhere.

[lircd]
nodaemon        = False
driver          = default
device          = auto
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 =
```

```
#listen         = [address:]port
#connect        = host[:port]
#loglevel       = 6
#uinput         = ...
#release        = ...
#logfile        = ...

[lircmd]
uinput          = False
nodaemon        = False

# [modinit]
# code = /usr/sbin/modprobe lirc_serial
# code1 = /usr/bin/setfacl -m g:lirc:rw /dev/uinput
# code2 = ...
```

Reboot your Pi

```
pi@d-diot:~ $ sudo reboot
```

Configure LIRC: d-diot image v.1.2

With Raspbian Buster the configuration and the installation of LIRC are different (see here). Now we have to create tow instances of the LIRC daemon, one of the receiver and one for the transmitter.

Before digging into the configuration details of each instance edit the file /etc/udev/rules.d/71-lirc.rules:

```
pi@d-diot:~ $ sudo nano /etc/udev/rules.d/71-lirc.rules
```

Add the following lines to get stable /dev/lirc-rx and /dev/lirc-tx device names:

/etc/udev/rules.d/71-lirc.rules

```
ACTION=="add", SUBSYSTEM=="lirc", DRIVERS=="gpio_ir_recv",
SYMLINK+="lirc-rx"
ACTION=="add", SUBSYSTEM=="lirc", DRIVERS=="gpio-ir-tx",
SYMLINK+="lirc-tx"
```

Exit (CTRL+x) and save the changes (y).

Receiver: /dev/lirc-rx

Edit the lirc_options.conf file:

```
pi@d-diot:~ $ sudo nano /etc/lirc/lirc_options.conf
```

Change the device and listening address.

/etc/lirc/lirc_options.conf

```
device          = /dev/lirc-rx
listen          = 0.0.0.0:8766
```

Exit (CTRL+x) and save the changes (y).

/etc/lirc/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.
#
# Some tools including mode2 and irw uses values such as
# driver, device, plugindir and loglevel as fallback values
# in not defined elsewhere.

[lircd]
nodaemon        = False
driver          = default
device          = /dev/lirc-rx
listen          = 0.0.0.0:8766
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 =
#listen         = [address:]port
#connect        = host[:port]
#loglevel       = 6
#release        = true
#release_suffix = _EVUP
#logfile        = ...
#driver-options = ...

[lircmd]
uinput          = False
nodaemon        = False

# [modinit]
# code = /usr/sbin/modprobe lirc_serial
```

```
# code1 = /usr/bin/setfacl -m g:lirc:rw /dev/uinput
# code2 = ...


# [lircd-uinput]
# add-release-events = False
# release-timeout    = 200
# release-suffix     = _EVUP
```

Transmitter: /dev/lirc-tx

Copy lirc_options.conf to lirc_tx_options.conf

```
pi@d-diot:~ $ sudo cp /etc/lirc/lirc_options.conf
/etc/lirc/lirc_tx_options.conf
```

Edit the lirc_tx_options.conf file:

```
pi@d-diot:~ $ sudo nano /etc/lirc/lirc_tx_options.conf
```

Change the following lines:

/etc/lirc/lirc_tx_options.conf

```
device          = /dev/lirc-tx
output          = /var/run/lirc/lircd-tx
pidfile         = /var/run/lirc/lircd-tx.pid
listen          = 0.0.0.0:8765
connect         = 127.0.0.1:8766
```

Exit (CTRL+x) and save the changes (y).

/etc/lirc/lirc_tx_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.
#
# Some tools including mode2 and irw uses values such as
# driver, device, plugindir and loglevel as fallback values
# in not defined elsewhere.

[lircd]
nodaemon        = False
driver          = default
device          = /dev/lirc-tx
```

```
listen          = 0.0.0.0:8765
connect         = 127.0.0.1:8766
output          = /var/run/lirc/lircd-tx
pidfile         = /var/run/lirc/lircd-tx.pid
plugindir       = /usr/lib/arm-linux-gnueabihf/lirc/plugins
permission      = 666
allow-simulate  = No
repeat-max      = 600
#effective-user =
#listen         = [address:]port
#connect        = host[:port]
#loglevel       = 6
#release        = true
#release_suffix = _EVUP
#logfile        = ...
#driver-options = ...

[lircmd]
uinput          = False
nodaemon        = False

# [modinit]
# code = /usr/sbin/modprobe lirc_serial
# code1 = /usr/bin/setfacl -m g:lirc:rw /dev/uinput
# code2 = ...


# [lircd-uinput]
# add-release-events = False
# release-timeout    = 200
# release-suffix     = _EVUP
```

Create /etc/systemd/system/lircd-tx.service:

```
pi@d-diot:~ $ sudo nano /etc/systemd/system/lircd-tx.service
```

and edit it to be:

/etc/systemd/system/lircd-tx.service

```
[Unit]
Documentation=man:lircd(8)
Documentation=http://lirc.org/html/configure.html
Description=Flexible IR remote input/output application support
Wants=lircd-setup.service
After=network.target lircd-setup.service lircd.service
```

```
[Service]
Type=simple
ExecStart=/usr/sbin/lircd --nodaemon --options-file
/etc/lirc/lirc_tx_options.conf
; User=lirc
; Group=lirc

; Hardening opts, see systemd.exec(5). Doesn't add much unless
; not running as root.
;
; # Required for dropping privileges in --effective-user.
; CapabilityBoundingSet=CAP_SETEUID
; MemoryDenyWriteExecute=true
; NoNewPrivileges=true
; PrivateTmp=true
; ProtectHome=true
; ProtectSystem=full

[Install]
WantedBy=multi-user.target
```

Exit (CTRL+x) and save the changes (y).

The lircd-tx.service is created on the basis of the original lircd service, visible with the following command:

```
pi@d-diot:~ $ systemctl cat lircd
```

Create a new socket for the transmitter daemon:

```
pi@d-diot:~ $ sudo nano /etc/systemd/system/lircd-tx.socket
```

and edit it:

/etc/systemd/system/lircd-tx.socket

```
[Socket]
ListenStream=/run/lirc/lircd-tx

[Install]
WantedBy=sockets.target
Also=lircd-tx.service
```

Exit (CTRL+x) and save the changes (y).

The lircd-tx socket is created on the basis of the original lircd socket, visible with the following command:

```
pi@d-diot:~ $ systemctl cat lircd.socket
```

Enable the lircd-tx daemon and autostart it at boot:

```
pi@d-diot:~ $ sudo systemctl daemon-reload
pi@d-diot:~ $ sudo systemctl enable lircd-tx
```

Create the file /usr/local/bin/irsend:

```
pi@d-diot:~ $ sudo nano /usr/local/bin/irsend
```

Add the following lines:

/usr/local/bin/irsend

```
#! /bin/sh
exec /usr/bin/irsend --device=/var/run/lirc/lircd-tx "$@"
```

Exit (CTRL+x) and save the changes (y). Make the file executable:

```
pi@d-diot:~ $ sudo chmod 755 /usr/local/bin/irsend
```

Reboot your Pi

```
pi@d-diot:~ $ sudo reboot
```

Test LIRC

Quick test of the IR receiver

```
pi@d-diot:~ $ sudo systemctl stop lircd
pi@d-diot:~ $ mode2 -d /dev/lirc0
```

Point a remote to the IR receiver and press some buttons. If all it's OK you should see a series of numbers in the terminal.

```
pi@d-diot:~ $ sudo systemctl start lircd
```

Quick test of the IR send

Downlaod a remote config file from the LIRC database and save it as test.conf

```
pi@d-diot:~ $ cd /etc/lirc/lircd.conf.d/
pi@d-diot:/etc/lirc/lircd.conf.d $ sudo wget -O test.conf
http://lirc.sourceforge.net/remotes/samsung/BN59-00940A
```

```
pi@d-diot:~ $ sudo systemctl restart lircd
pi@d-diot:~ $ irsend SEND_START Samsung_BN59-00940A KEY_1
```

If everything is OK you should see the IR led led blinking. If you don't see anything with your naked eye, try with the camera of your smartphone.
To stop sending IR signals

```
pi@d-diot:~ $ irsend SEND_STOP Samsung_BN59-00940A KEY_1
```

Remove the remote file once finished.

```
pi@d-diot:~ $ sudo rm /etc/lirc/lircd.conf.d/test.conf
pi@d-diot:~ $ sudo systemctl restart lircd
```

Clone an IR signal

To clone the signal of a remote that is not present in the LIRC database you can use the irrecord program, following the on-screen instructions. Name your remote “test” to create a file named “test.lircd.conf”

```
pi@d-diot:~ $ sudo systemctl stop lircd
pi@d-diot:~ $ irrecord
pi@d-diot:~ $ sudo cp test.lircd.conf /etc/lirc/lircd.conf.d/test.lircd.conf
pi@d-diot:~ $ sudo systemctl restart lircd
```

To Check that the cloned signal is well recognized by LIRC run the irw utility.

```
pi@d-diot:~ $ irw
```

If OK you should see the remote and key name previously recorder every time the corresponding key is pressed on the remote. To send the cloned code continuosly:

```
pi@d-diot:~ $ irsend SEND_START test KEY_UP
```

To stop sending

```
pi@d-diot:~ $ irsend SEND_STOP test KEY_UP
```

Remove the test file once finished.

```
pi@d-diot:~ $ sudo rm /etc/lirc/lircd.conf.d/test.lircd.conf
pi@d-diot:~ $ sudo systemctl restart lircd
```

Install lirc_web

Install lirc_web via npm

```
pi@d-diot:~ $ sudo npm install -g lirc_web
```

To make your lirc web installation working, according to this, you have to modify the file "lirc_node.js":

```
pi@d-diot:~ $ find / -name lirc_node.js
```

Usually it is at the following path "/usr/local/lib/node_modules/lirc_web/node_modules/lirc_node/lib/lirc_node.js", so:

```
pi@d-diot:~ $ sudo nano
/usr/local/lib/node_modules/lirc_web/node_modules/lirc_node/lib/lirc_node.js
```

and then:

- change line 32 and 53 `stderr.split('\n');` to `stdout.split('\n');`
- change line 37 from `element.match(/\s(.*)$/);` to `element.match(/\b(.*)$/);`
- change line 56 to `element.match(/\b.*\s(.*)$/);`

Exit (CTRL+x) and save the changes (y).

/usr/local/lib/node_modules/lirc_web/node_modules/lirc_node/lib/lirc_node.js

```
exports.version = '0.0.2';
exports.IRSend = require('./irsend');
exports.irsend = new exports.IRSend();
exports.remotes = {};

exports.IRReceive = require('./irreceive');
var irreceive = new exports.IRReceive();
exports.addListener = irreceive.addListener.bind(irreceive);
exports.on = exports.addListener;
exports.removeListener = irreceive.removeListener.bind(irreceive);

// In some cases the default lirc socket does not work
// More info at
http://wiki.openelec.tv/index.php?title=Guide_to_Lirc_IR_Blasting
exports.setSocket = function(socket) {
  exports.irsend.setSocket(socket);
}

exports.init = function(callback) {
  exports.irsend.list('', '', irsendCallback);

  function irsendCallback(error, stdout, stderr) {
    exports._populateRemotes(error, stdout, stderr);
    exports._populateCommands();
    if (callback) callback();
```

```
  }

  return true;
};

// Private
exports._populateRemotes = function(error, stdout, stderr) {
  var remotes = stdout.split('\n');

  exports.remotes = {};

  remotes.forEach(function(element, index, array) {
    var remoteName = element.match(/\b(.*)$/);
    if (remoteName) exports.remotes[remoteName[1]] = [];
  });
};

exports._populateCommands = function() {
  for (var remote in exports.remotes) {
    (function(remote) {
      exports.irsend.list(remote, '', function(error, stdout, stderr) {
        exports._populateRemoteCommands(remote, error, stdout, stderr);
      });
    })(remote);
  }
};

exports._populateRemoteCommands = function(remote, error, stdout,
stderr) {
  var commands = stdout.split('\n');

  commands.forEach(function(element, index, array) {
    var commandName = element.match(/\b.*\s(.*)$/);
    if (commandName && commandName[1])
exports.remotes[remote].push(commandName[1]);
  });
};
```

Start lirc_web:

```
pi@d-diot:~ $ lirc_web
```

To access the web interface, put the following url in a web browser http://YOUR_RASPI_IP:3000, or if your network supports avahi / zeroconf, http://d-diot.local:3000

In the next steps we will integrate lirc_web in the Home Assistant home page with a Panel iFrame.

To terminate lirc_web press CTRL+c

Autostart lirc_web at boot

Edit the /etc/rc.local file

```
pi@d-diot:~ $ sudo nano /etc/rc.local
```

Add the following lines before the “exit 0” string

/etc/rc.local

```
# Launch lirc_web
su - homeassistant -c lirc_web &
```

SSL encryption

Follow this guide.

Usage

See the how to section.