

LIRC Setup for GPIO based IR RX on Ubuntu 18.04

Configuring LIRC for GPIO based IR RX

Please follow these 4 steps.

1. Install LIRC Package
2. Modify hardware.conf
3. Register lircd.conf
4. Restart lircd service

[Step 1] Install LIRC Package

target

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

[Step 2] Modify lirc_options.conf

First, you need to set up lirc_options.conf under /etc/lirc/.

This includes driver, port and other important configuration and is generated by default when lirc is installed.

target

```
$ sudo vi /etc/lirc/lirc_options.conf
```

The following parts are needed to be modified.

(1) driver and device

```
driver      = default
device     = /dev/lirc0
```

(2) module init

Let's suppose to connect IR output of your extra IR receiver module to Pin '11' of expansion connectors (J2) of ODROID-C2.

(Expansion net name : GPIOX.BIT19, GPIO number : 247)

```
[modinit]
code = /sbin/modprobe gpio-ir-recv
# gpio_nr must be adjusted based on port connections
code1 = /sbin/modprobe gpioplug-ir-recv gpio_nr=247 active_low=1
```

(3) input event

```
[lircd-uinput]
add-release-events = True
release-timeout    = 50
release-suffix     = _EVUP
```

Please note that some parts in lirc_options.conf depend on the toolchain version, so select the correct conf file among the following references based on your board.

Reference lirc_options.conf

ODROID-C4

For the example on ODROID-C4, Pin#11 (GPIOX.3, GPIO479) is considered.

[lirc_options.conf](#)

```
[lircd]
nodaemon          = False
driver            = default
device           = /dev/lirc0
output           = /var/run/lirc/lircd
pidfile          = /var/run/lirc/lircd.pid
plugindir        = /usr/lib/aarch64-linux-gnu/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 = /sbin/modprobe gpio-ir-recv
# gpio_nr must be adjusted based on port connections.
code1 = /sbin/modprobe gpioplug-ir-recv gpio_nr=479 active_low=1

[lircd-uinput]
add-release-events = True
release-timeout    = 50
release-suffix     = _EVUP
```

ODROID-N2

For the example on ODROID-N2, Pin#11 (GPIOX.3, GPIO479) is considered.

[lirc_options.conf](#)

```
[lircd]
nodaemon          = False
driver            = default
device            = /dev/lirc0
output            = /var/run/lirc/lircd
pidfile           = /var/run/lirc/lircd.pid
plugindir         = /usr/lib/aarch64-linux-gnu/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 = /sbin/modprobe gpio-ir-recv
# gpio_nr must be adjusted based on port connections.
code1 = /sbin/modprobe gpioplug-ir-recv gpio_nr=479 active_low=1

[lircd-uinput]
```

```
add-release-events = True
release-timeout    = 50
release-suffix     = _EVUP
```

ODROID-C2

For the example on ODROID-C2, Pin#11 (GPIOX.BIT19, GPIO247) is considered.

[lirc_options.conf](#)

```
[lircd]
nodaemon      = False
driver        = default
device        = /dev/lirc0
output        = /var/run/lirc/lircd
pidfile       = /var/run/lirc/lircd.pid
plugindir     = /usr/lib/aarch64-linux-gnu/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 = /sbin/modprobe gpio-ir-recv
# gpio_nr must be adjusted based on port connections.
code1 = /sbin/modprobe gpioplug-ir-recv gpio_nr=247 active_low=1

[lircd-uinput]
add-release-events = True
release-timeout    = 50
release-suffix     = _EVUP
```

ODROID-XU4

For the example on ODROID-XU4, Pin#13 (GPX1.5, GPIO21) is considered.

[lirc_options.conf](#)

```
[lircd]
nodaemon      = False
driver        = default
device        = /dev/lirc0
output        = /var/run/lirc/lircd
pidfile       = /var/run/lirc/lircd.pid
plugindir     = /usr/lib/arm-linux-gnueabi/hf/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 = /sbin/modprobe gpio-ir-recv
# gpio_nr must be adjusted based on port connections.
code1 = /sbin/modprobe gpioplug-ir-recv gpio_nr=21 active_low=1

[lircd-uinput]
add-release-events = True
release-timeout    = 50
release-suffix     = _EVUP
```

ODROID-C1

For the example on ODROID-C1, Pin#7 (GPIOY.3, GPIO83) is considered.

[lirc_options.conf](#)

```
[lircd]
nodaemon      = False
```

```
driver          = default
device          = /dev/lirc0
output          = /var/run/lirc/lircd
pidfile         = /var/run/lirc/lircd.pid
plugindir      = /usr/lib/arm-linux-gnueabi/hf/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 = /sbin/modprobe gpio-ir-recv
# gpio_nr must be adjusted based on port connections.
code1 = /sbin/modprobe gpioplug-ir-recv gpio_nr=83 active_low=1

[lircd-uinput]
add-release-events = True
release-timeout    = 50
release-suffix     = _EVUP
```

[Step 3] Register lircd.conf

lircd.conf files under /etc/lirc/lircd.conf.d/ include signal timing information and provide mappings from button presses to key symbols.

You need to add a new lircd.conf based on your custom remote controller.

target

```
$ sudo vi /etc/lirc/lircd.conf.d/hk.lircd.conf
```

reference lircd.conf

This reference hk.lircd.conf is to support Hardkernel IR remote controller.

<https://www.hardkernel.com/shop/ir-remote-controller/>

[hk.lircd.conf](#)

```
begin remote

name odroid
bits 16
flags SPACE_ENC|CONST_LENGTH
eps 30
aeps 100

header 9000 4500
one 563 1688
zero 563 564
ptrail 563
pre_data_bits 16
pre_data 0x4DB2
repeat 9000 2250
gap 100000
toggle_bit_mask 0x0
  begin codes
    KEY_LEFT 0x9966
    KEY_RIGHT 0x837C
    KEY_UP 0x53AC
    KEY_DOWN 0x4BB4
    KEY_ENTER 0x738C
    KEY_HOME 0x41BE
    KEY_MUTE 0x11EE
    KEY_MENU 0xA35C
    KEY_BACK 0x59A6
    KEY_VOLUMEDOWN 0x817E
    KEY_VOLUMEUP 0x01FE
    KEY_POWER 0x3BC4
  end codes
end remote
```

[Step 4] Restart lircd service

[target](#)

```
$ sudo service lircd restart
$ sudo service lircd-uinput restart
```

To confirm if the LIRC daemon is running based on gpio-ir-recv, check device nodes and protocol name of ir-keytable.

target

```
$ ls /dev/lirc*  
/dev/lirc0
```



If you have the warning **No such device** during **modprobe gpioplug-ir-recv**, check if the gpio_nr is already assigned to another purpose and change to another pins.

[Step 5] How to check if it works

https://wiki.odroid.com/odroid-c2/application_note/lirc/lirc_ubuntu18.04#how_to_test

From:
<https://wiki.odroid.com/> - **ODROID Wiki**

Permanent link:
https://wiki.odroid.com/common/application_note/gpio/ir/lirc_gpioir_ubuntu18.04

Last update: **2020/04/23 15:36**

