A technical report on the Device Drivers Course Project: Controlling a projector using Raspberry Pi

Prashanthi S.K

22/11/2017

Contents

1 Abstract

This is a technical paper on the course project for Device Driver Development.The objective of the project is to be able to control the projector using a Raspberry Pi. Sub-problems that are addressed here are as follows:
1)To connect an IR receiver to the Pi and decode the signals sent by a remote
2)To connect an IR emitter to the pi and send codes to the projector to turn it off/on
3)To be able to automatically detect when the HDMI cable is unplugged from the pi and thereafter turn off the projector by sending the appropriate codes.

2 Questions put forth in the assignment

1. How do you distinguish an IR emitter from a receiver?
Going by appearance alone,the transmitter is usually a two-legged LED whereas the receiver used in most TSOP modules is a phototransistor and therefore has three terminals.In some cases,a two-legged photodiode may be used as a receiver but it is usually painted black and can easily be detected.Another way to detect is to supply a voltage to the LED and use a phone camera to see if it is emitting IR.

2.We can't see IR. How do you check if emitter is working?
IR has a wavelength that is longer than the visible spectrum and is therefore not perceived by the human eye[1][2].On the other hand,a phone camera has a much larger range and can therefore sense IR as well.Therefore,a phone camera can be used to check if the IR emitter is working.Some phone cameras,however,come with IR filters and cannot be used for this purpose.

3. What is the frequency of the IR receiver?
IR frequency is typically between 33 and 40kHz or 50 and 60kHz[3].NEC,which is the most commonly used protocol,specifies a frequency of 38kHz,whereas Phillips RC-5 uses 36kHz.Most IR receivers are designed for a frequency of 38kHz.

4. Why is the IR receiver built to a specific frequency?
IR sources are all around us.Basically anything that emits heat,like light bulbs,the sun,are all sources of IR.[4] If the IR receiver is not built for a particular frequency,all the ambient IR will get detected causing a lot of false positives.Since this is undesirable,usually IR receivers are built to 38kHz.

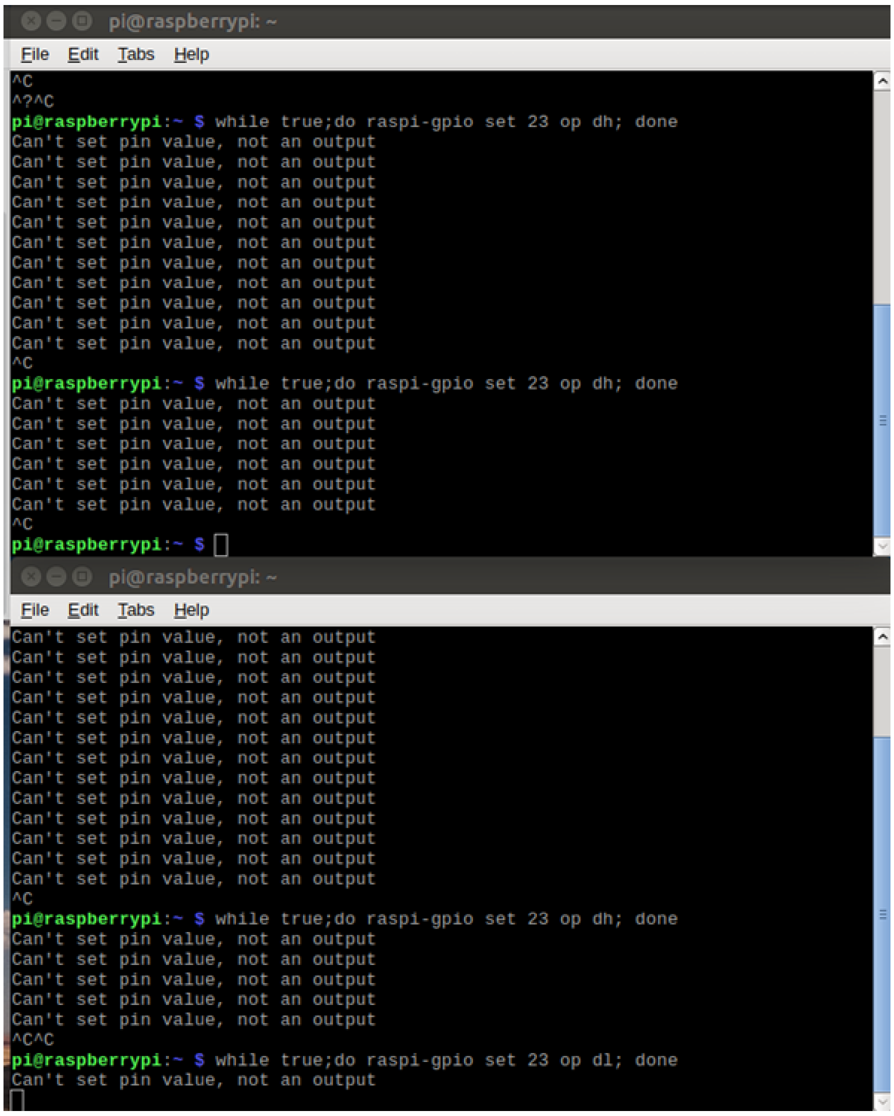
5.Why is IR communication based on gating a high frequency pulse instead of a direct on/off pulse?
A PWM based modulation with a carrier of 38kHz is used in IR systems.The carrier is pulsed using PWM for two reasons:so that the led can gain sufficient time to cool off and also because receivers are built to a particular frequency.Also,pulsing reduces the effects of ambient lighting.[5]

3 Approach,Experiments and Observations

When I initially read the assignment,I honestly had no clarity on what is to be done.All I understood was the problem statement. I kept re-reading the question and with time,it seemed clearer.One thing that especially confused me was the mention of two lircs–the driver and the userland package.It brought up the question of whether we had to use the userland package and its sub-utilities or to modify the lirc driver itself.

3.1 GPIO

I first started by reading up on lirc[6][7].The lwn article required a couple of readings,but it was very well-written and gave me a much clearer picture than the lirc homepage itself.Since this project involved both hardware and software,it seemed to me that a reasonable approach would be to get one of them working at first and then proceed to the other. I connected the IR led(transmitter) and photodiode(receiver) to the Pi and started by conducting a few simple experiments on GPIO. I tried using both raspi-gpio and the sysfs methods. The equivalent of a Hello world program in hardware is a blink LED and that was my first experiment. Since IR lies outside the spectrum of human vision, I used a phone camera(without IR filters) to check if the program was working. Out of curiosity,I wondered what would happen if two processes tried to write to a GPIO pin at the same time.I was sure there would be some kind of a locking mechanism,but I wanted to see it.As expected,there was an error.But the error wasnt what Id expected–it said not an output.I pulled up the source code of raspi-gpio and looked through it.This error was thrown only if the pin state was not an output.I concluded that the state must have changed to something else but did not investigate further as this seemed tangential to the objectives.



```
^C
^?^C
pi@raspberrypi:~ $ while true;do raspi-gpio set 23 op dh; done
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
^C
pi@raspberrypi:~ $ while true;do raspi-gpio set 23 op dh; done
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
^C
pi@raspberrypi:~ $ 
```

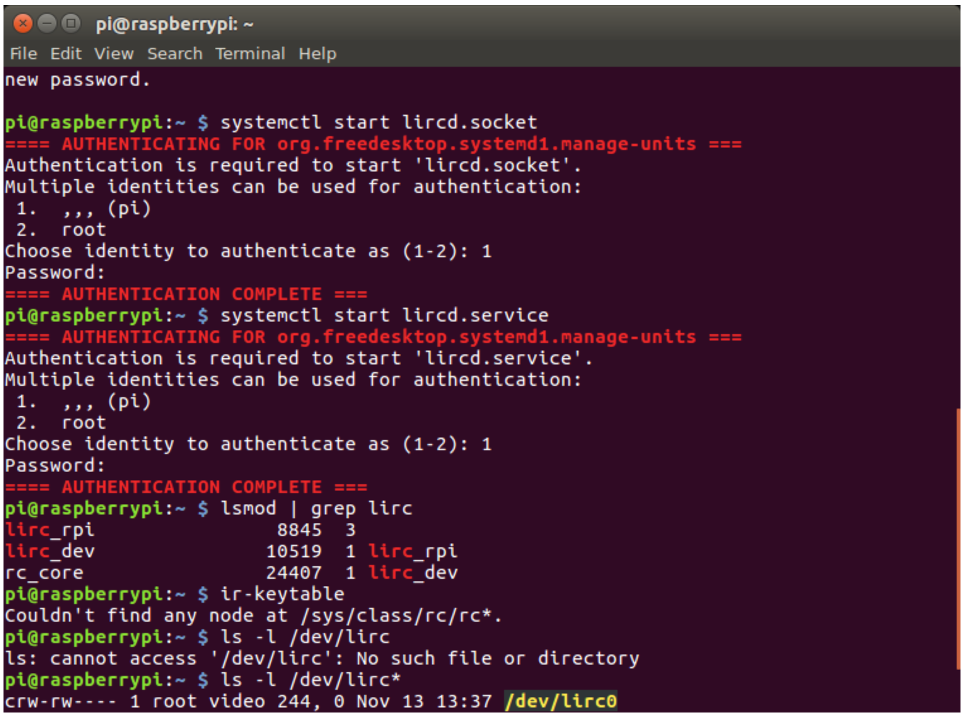
```
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
^C
pi@raspberrypi:~ $ while true;do raspi-gpio set 23 op dh; done
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
^C^C
pi@raspberrypi:~ $ while true;do raspi-gpio set 23 op dl; done
Can't set pin value, not an output
```

Figure 1: When two processes try to write to a pin at the same time

3.2 LIRC–userland package and its subutilities

After the gpio experiments,I decided to look into lirc.I thought of first using the userspace utility to receive and decode the signals that the remote sends to the projector and then send these same signals back to the projector using the utility.Once this small loop worked,I could proceed to writing a driver that did the same.As suggested in the assignment,I began by installing and configuring lirc and testing if it was running[6][8].

The next logical step would be to try and receive pulses from an IR remote.I connected a photodiode to the Pi and used mode2 for this purpose.It



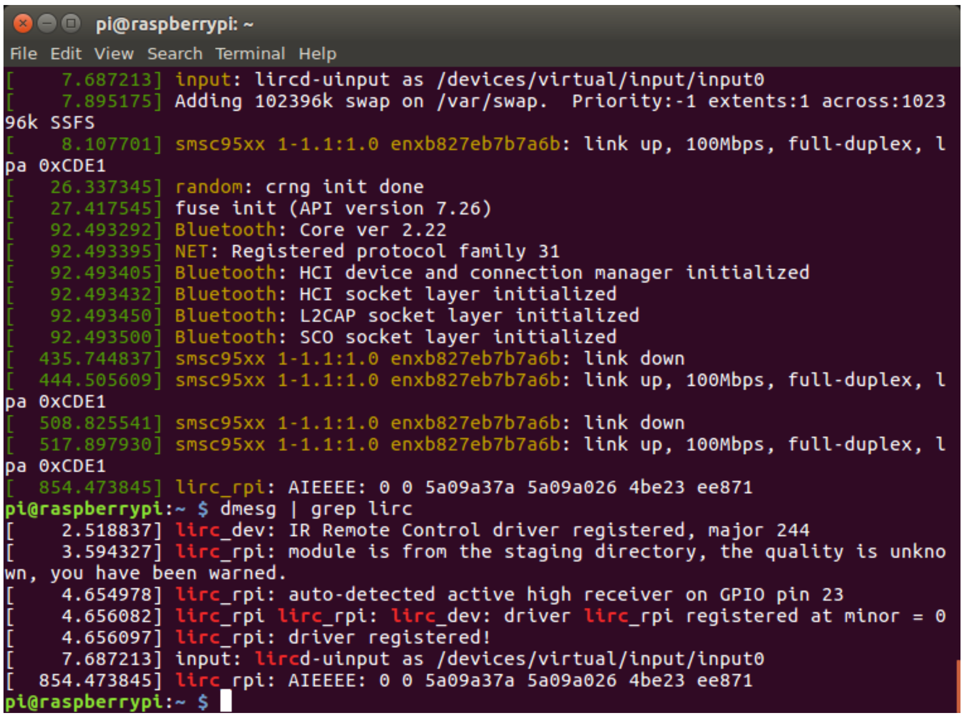
```
pi@raspberrypi: ~
File  Edit  View  Search  Terminal  Help
new password.

pi@raspberrypi:~ $ systemctl start lircd.socket
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to start 'lircd.socket'.
Multiple identities can be used for authentication:
 1.  ,,, (pi)
 2.  root
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ===
pi@raspberrypi:~ $ systemctl start lircd.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to start 'lircd.service'.
Multiple identities can be used for authentication:
 1.  ,,, (pi)
 2.  root
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ===
pi@raspberrypi:~ $ lsmod | grep lirc
lirc_rpi                8845  3
lirc_dev               10519  1 lirc_rpi
rc_core                24407  1 lirc_dev
pi@raspberrypi:~ $ ir-keytable
Couldn't find any node at /sys/class/rc/rc*.
pi@raspberrypi:~ $ ls -l /dev/lirc
ls: cannot access '/dev/lirc': No such file or directory
pi@raspberrypi:~ $ ls -l /dev/lirc*
crw-rw---- 1 root video 244, 0 Nov 13 13:37 /dev/lirc0
```

Figure 2: Checking if LIRC is loaded

didnt work as expected and I got something called a partial read.To debug this,I connected it a CRO so that I could see the signal at a hardware level itself.But I was not able to get a proper signal. I looked it up on the net and read that photodiodes do not have built-in amplifiers and therefore would not be able to sense the emitted radiation in a lot of cases and that it would be easier to use a module that is built with a phototransistor.[9] Im pretty sure that I could have gotten the photodiode itself to work with a bit of additional circuitry,but it seemed to me at the time that there were a lot more things to be done in the project,so I didnt debug the issue further and instead took a GPIO board that came with a TSOP(Thin Small Outline Package) IR sensor.

Now when I tried mode2 and pressed a few buttons on the remote,the expected output in the form of spaces and pulses was seen on the terminal.One puzzling observation was that the same key seemed to produce slightly different outputs.I looked it up and read that the some remotes use a number of outputs for the same key–they alternate between one of these every time the same key is pressed.(Although,in my case,I dont think this was the issue.I just saw slight differences:say 1045 once and 1047 the next time.I think this is due to measuring errors or accuracy limitations of the hardware).

The next step was to test a loop–generate a configuration file for the remote using irrecord,transmit the codes using irsend and receive them back

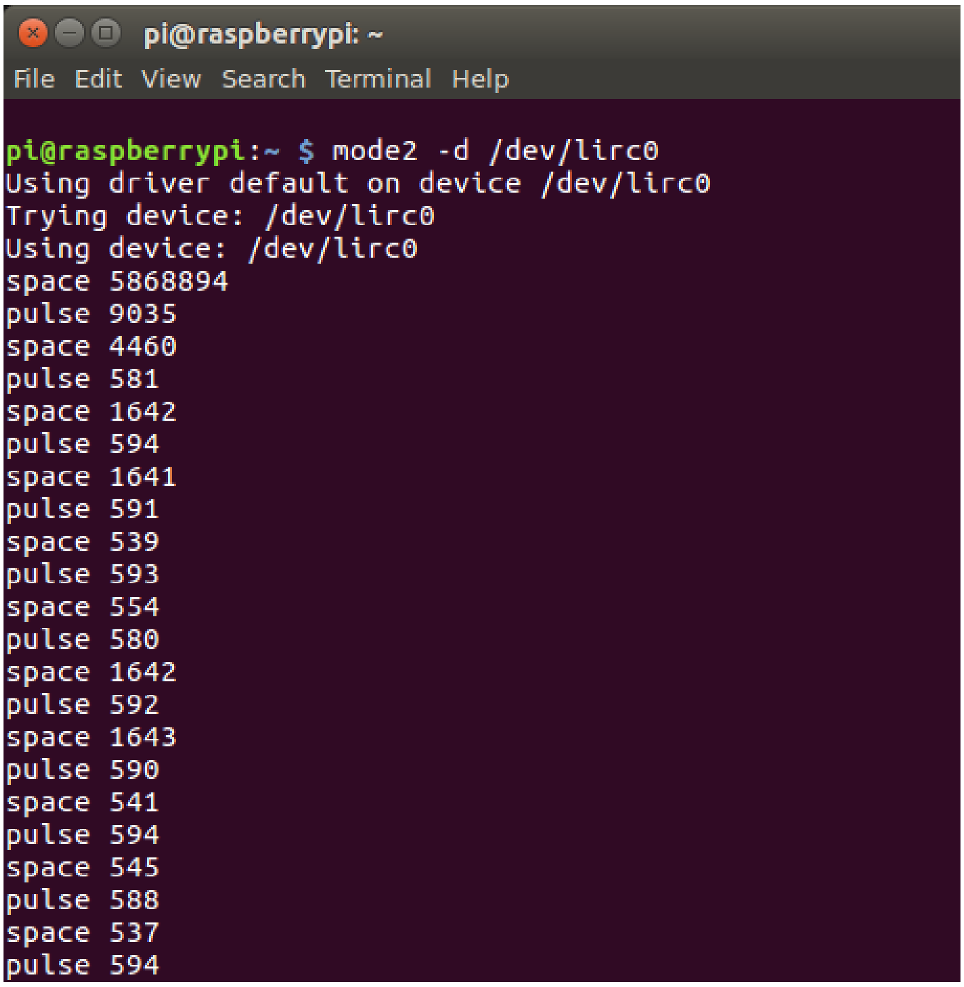


```
[    7.687213] input: lircd-uinput as /devices/virtual/input/input0
[    7.895175] Adding 102396k swap on /var/swap.  Priority:-1 extents:1 across:1023
96k SSFS
[    8.107701] smsc95xx 1-1.1:1.0 enxb827eb7b7a6b: link up, 100Mbps, full-duplex, l
pa 0xCDE1
[   26.337345] random: crng init done
[   27.417545] fuse init (API version 7.26)
[   92.493292] Bluetooth: Core ver 2.22
[   92.493395] NET: Registered protocol family 31
[   92.493405] Bluetooth: HCI device and connection manager initialized
[   92.493432] Bluetooth: HCI socket layer initialized
[   92.493450] Bluetooth: L2CAP socket layer initialized
[   92.493500] Bluetooth: SCO socket layer initialized
[  435.744837] smsc95xx 1-1.1:1.0 enxb827eb7b7a6b: link down
[  444.505609] smsc95xx 1-1.1:1.0 enxb827eb7b7a6b: link up, 100Mbps, full-duplex, l
pa 0xCDE1
[  508.825541] smsc95xx 1-1.1:1.0 enxb827eb7b7a6b: link down
[  517.897930] smsc95xx 1-1.1:1.0 enxb827eb7b7a6b: link up, 100Mbps, full-duplex, l
pa 0xCDE1
[  854.473845] lirc_rpi: AIEEEE: 0 0 5a09a37a 5a09a026 4be23 ee871
pi@raspberrypi:~ $ dmesg | grep lirc
[    2.518837] lirc_dev: IR Remote Control driver registered, major 244
[    3.594327] lirc_rpi: module is from the staging directory, the quality is unkno
wn, you have been warned.
[    4.654978] lirc_rpi: auto-detected active high receiver on GPIO pin 23
[    4.656082] lirc_rpi lirc_rpi: lirc_dev: driver lirc_rpi registered at minor = 0
[    4.656097] lirc_rpi: driver registered!
[    7.687213] input: lircd-uinput as /devices/virtual/input/input0
[  854.473845] lirc_rpi: AIEEEE: 0 0 5a09a37a 5a09a026 4be23 ee871
pi@raspberrypi:~ $
```

Figure 3: grep dmesg for LIRC

using irw.I did the irrecord part this without facing any difficulty.Irsend also worked but to my chagrin irw did not receive anything.I re-did the whole recording,suspecting a faulty config file but to no avail.I opened up both the config files generated and compared them.Initially,I looked only at the key codes and saw the same ones in both files.After I recorded for a third time,I looked at the files more closely and saw that they differed in the parts above the codes–header,one,zero,trail etc.This was reminiscent of the same behavior I saw with mode2:slight differences.But obviously, these slight differences were causing a huge problem because they were not recognised by irw.

I searched for this problem but couldnt find a solution online.Intuitively,this seemed to be a recording malfunction and this is what seemed the logical reason as to why.If the program made a slight mistake in recognising the value for a one and a zero in terms of spaces and pulses,the entire code generated for all the keys would use these to denote 1s and 0s thereby not recognising the key presses.I was not very sure about this deduction except that it seemed logical at the time and I headed over to the man page of irrecord[10].There,I came across an argument called force with the instruction that it should be used when recording fails and that it would generate a raw file.I tried it out and had a look at the raw file,where each key press was denoted by numbers themselves rather than a binary representation.

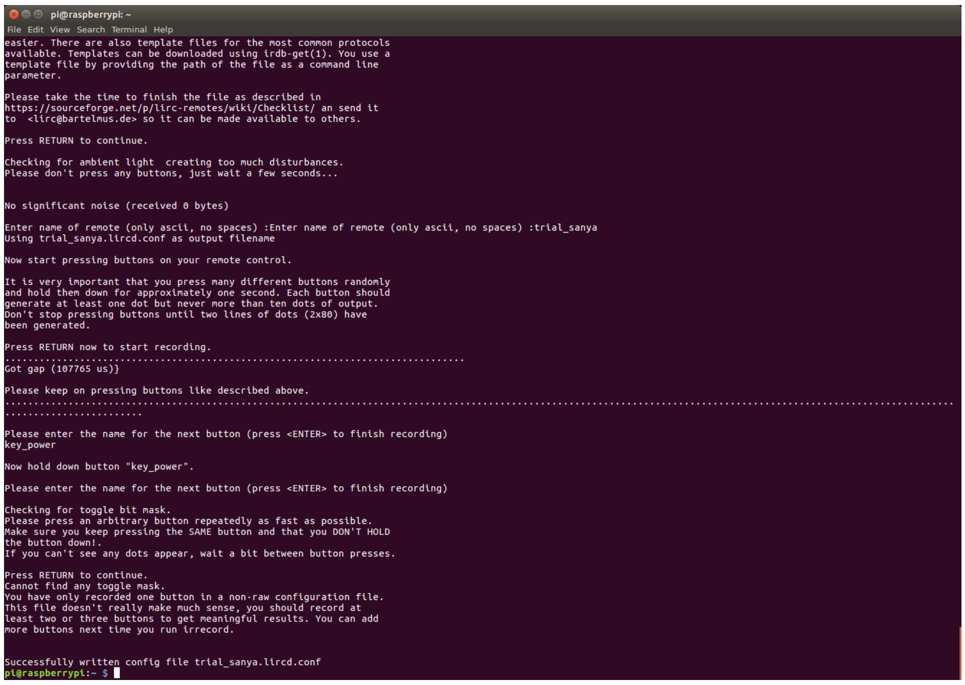


```
pi@raspberrypi:~ $ mode2 -d /dev/lirc0
Using driver default on device /dev/lirc0
Trying device: /dev/lirc0
Using device: /dev/lirc0
space 5868894
pulse 9035
space 4460
pulse 581
space 1642
pulse 594
space 1641
pulse 591
space 539
pulse 593
space 554
pulse 580
space 1642
pulse 592
space 1643
pulse 590
space 541
pulse 594
space 545
pulse 588
space 537
pulse 594
```

Figure 4: Ouput of mode2

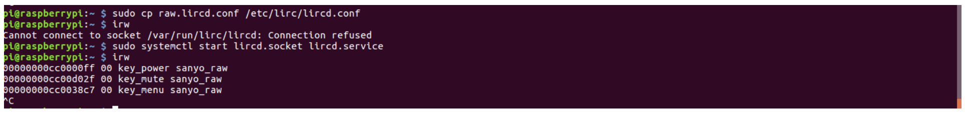
I moved this file to the appropriate location,fairly sure that it would solve my problem.However,to my surprise,irw was still blank. After struggling for what seemed like eons,I finally figured it out.There were two locations:One was the default config file that lirc used lircd.conf.The other was a directory that contained a number of config files:/etc/lirc/lircd.conf.d.As per what I had read,lirc was supposed to scan through and include all the config files in this directory and that the old method of putting the config file in lircd.conf was deprecated.However,this clearly wasnt happening and I copied the newly generated file to replace lircd.conf itself.Finally,irw showed some results and all my efforts fell into perspective.

I quickly went on to test the loop that I had initially envisioned and for once,that worked without any glitches.I tested the same for about three remotes,all of them worked with irsend and irw.



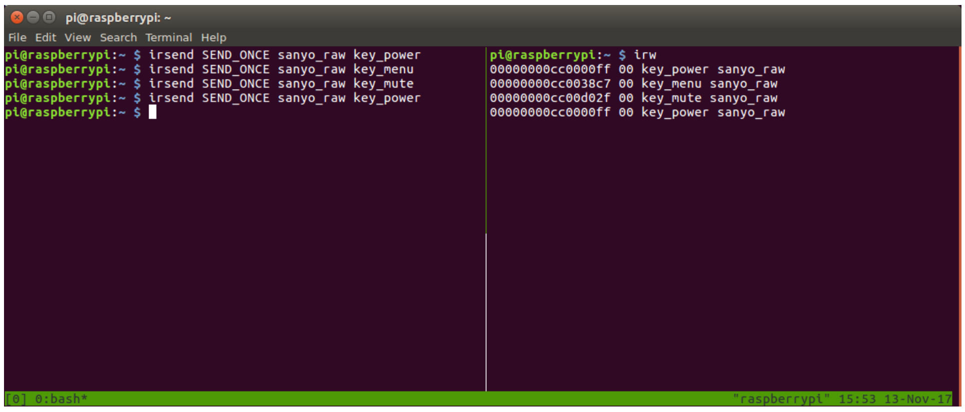
```
easier. There are also template files for the most common protocols
available. Templates can be downloaded using irdb-get(1). You use a
template file by providing the path of the file as a command line
parameter.

Please take the time to finish the file as described in
https://sourceforge.net/p/lirc-remotes/wiki/Checklist/ an send it
to  <lirc@bartelmus.de> so it can be made available to others.

Press RETURN to continue.

Checking for ambient light  creating too much disturbances.
Please don't press any buttons, just wait a few seconds...


No significant noise (received 0 bytes)

Enter name of remote (only ascii, no spaces) :Enter name of remote (only ascii, no spaces) :trial_sanya
Using trial_sanya.lircd.conf as output filename

Now start pressing buttons on your remote control.

It is very important that you press many different buttons randomly
and hold them down for approximately one second. Each button should
generate at least one dot but never more than ten dots of output.
Don't stop pressing buttons until two lines of dots (2x80) have
been generated.

Press RETURN now to start recording.
................................................................................
Got gap (107765 us)}

Please keep on pressing buttons like described above.
................................................................................................................................................................................
........................

Please enter the name for the next button (press <ENTER> to finish recording)
key_power

Now hold down button "key_power".

Please enter the name for the next button (press <ENTER> to finish recording)

Checking for toggle bit mask.
Please press an arbitrary button repeatedly as fast as possible.
Make sure you keep pressing the SAME button and that you DON'T HOLD
the button down!.
If you can't see any dots appear, wait a bit between button presses.

Press RETURN to continue.
Cannot find any toggle mask.
You have only recorded one button in a non-raw configuration file.
This file doesn't really make much sense, you should record at
least two or three buttons to get meaningful results. You can add
more buttons next time you run irrecord.


Successfully written config file trial_sanya.lircd.conf
pi@raspberrypi:~ $
```

Figure 5: Ouput of irrecord



```
pi@raspberrypi:~ $ sudo cp raw.lircd.conf /etc/lirc/lircd.conf
pi@raspberrypi:~ $ irw
Cannot connect to socket /var/run/lirc/lircd: Connection refused
pi@raspberrypi:~ $ sudo systemctl start lircd.socket lircd.service
pi@raspberrypi:~ $ irw
00000000cc0000ff 00 key_power sanyo_raw
00000000cc00d02f 00 key_mute sanyo_raw
00000000cc0038c7 00 key_menu sanyo_raw
^C
```

Figure 6: Trying irsend



```
pi@raspberrypi:~ $ irsend SEND_ONCE sanyo_raw key_power
pi@raspberrypi:~ $ irsend SEND_ONCE sanyo_raw key_menu
pi@raspberrypi:~ $ irsend SEND_ONCE sanyo_raw key_mute
pi@raspberrypi:~ $ irsend SEND_ONCE sanyo_raw key_power
pi@raspberrypi:~ $
```

```
pi@raspberrypi:~ $ irw
00000000cc0000ff 00 key_power sanyo_raw
00000000cc0038c7 00 key_menu sanyo_raw
00000000cc00d02f 00 key_mute sanyo_raw
00000000cc0000ff 00 key_power sanyo_raw
```

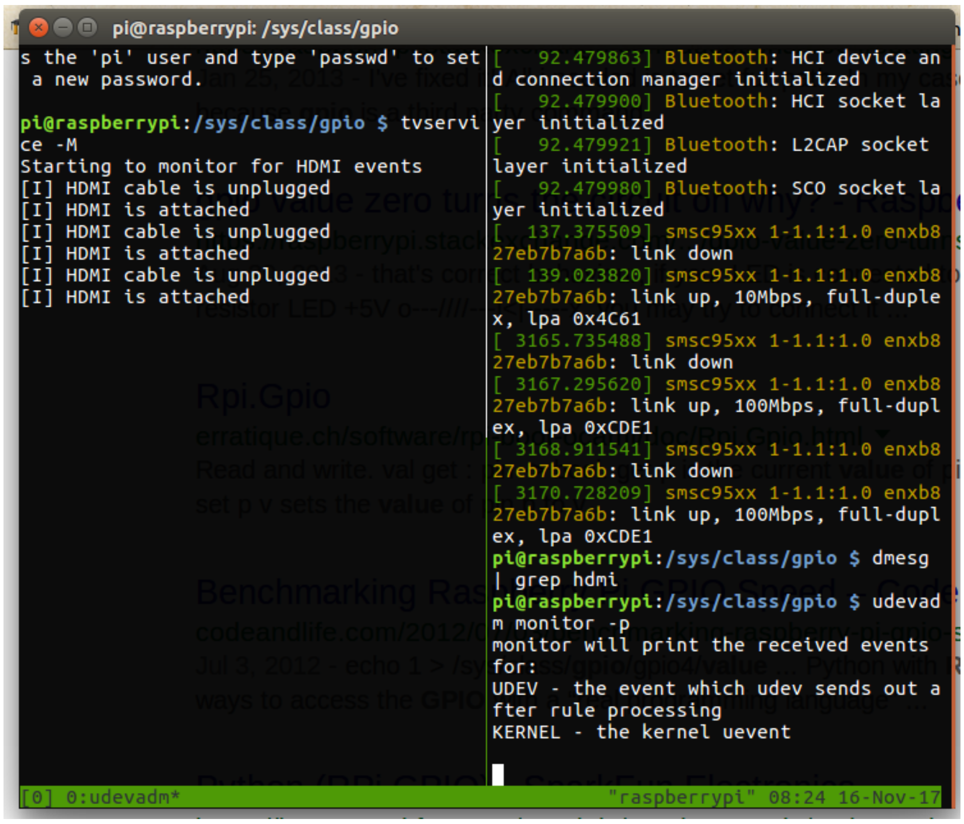
```
[0] 0:bash*                                    "raspberrypi" 15:53 13-Nov-17
```

Figure 7: The loop–irsend and irw

But when I tried to control the projector using irsend,it did not work for one projector(worked with the other two remotes).One observation was that the projector was placed at a significant height in this class,so it could be the power was insufficient.

3.3 Tvservice,Uevents and Mailbox

I started out with the tvservice utility.I used the -s and -M flags and observed the change in outputs when the hdmi cable was connected and disconnected. Tvservice -n gave the name of the device as well.Although the assignment made it quite clear that the hdmi connect event would not appear as a uevent,I tried it out just to verify.But no uevents were generated.When I tried the same on my laptop,uevents were generated as expected.



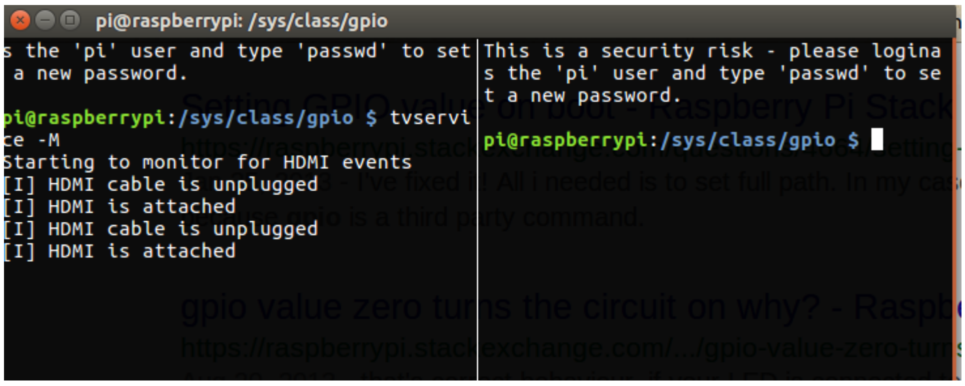
```
pi@raspberrypi: /sys/class/gpio
s the 'pi' user and type 'passwd' to set
 a new password.

pi@raspberrypi:/sys/class/gpio $ tvservi
ce -M
Starting to monitor for HDMI events
[I] HDMI cable is unplugged
[I] HDMI is attached
[I] HDMI cable is unplugged
[I] HDMI is attached
[I] HDMI cable is unplugged
[I] HDMI is attached
```

```
[   92.479863] Bluetooth: HCI device an
d connection manager initialized
[   92.479900] Bluetooth: HCI socket la
yer initialized
[   92.479921] Bluetooth: L2CAP socket
layer initialized
[   92.479980] Bluetooth: SCO socket la
yer initialized
[  137.375509] smsc95xx 1-1.1:1.0 enxb8
27eb7b7a6b: link down
[  139.023820] smsc95xx 1-1.1:1.0 enxb8
27eb7b7a6b: link up, 10Mbps, full-duple
x, lpa 0x4C61
[ 3165.735488] smsc95xx 1-1.1:1.0 enxb8
27eb7b7a6b: link down
[ 3167.295620] smsc95xx 1-1.1:1.0 enxb8
27eb7b7a6b: link up, 100Mbps, full-dupl
ex, lpa 0xCDE1
[ 3168.911541] smsc95xx 1-1.1:1.0 enxb8
27eb7b7a6b: link down
[ 3170.728209] smsc95xx 1-1.1:1.0 enxb8
27eb7b7a6b: link up, 100Mbps, full-dupl
ex, lpa 0xCDE1
pi@raspberrypi:/sys/class/gpio $ dmesg
| grep hdmi
pi@raspberrypi:/sys/class/gpio $ udevad
m monitor -p
monitor will print the received events
for:
UDEV - the event which udev sends out a
fter rule processing
KERNEL - the kernel uevent
```

```
[0] 0:udevadm*                          "raspberrypi" 08:24 16-Nov-17
```

Figure 8: No uevent on connection of HDMI on the Pi

At this point,I did not have clarity on how to proceed.There were two points mentioned in the assignment:tvservice.c and mailboxes.I decided to read up on both of them.[10][11][12][13] These two exercises took up a lot of time,especially tvservice.c.Since hdmi events were managed by the gpu,underneath the function calls tvservice had to be using the mailbox read-write calls–therefore I had to delve deeper into the code till I found the final APIs reading from the mailbox.I found a callback function tvser-



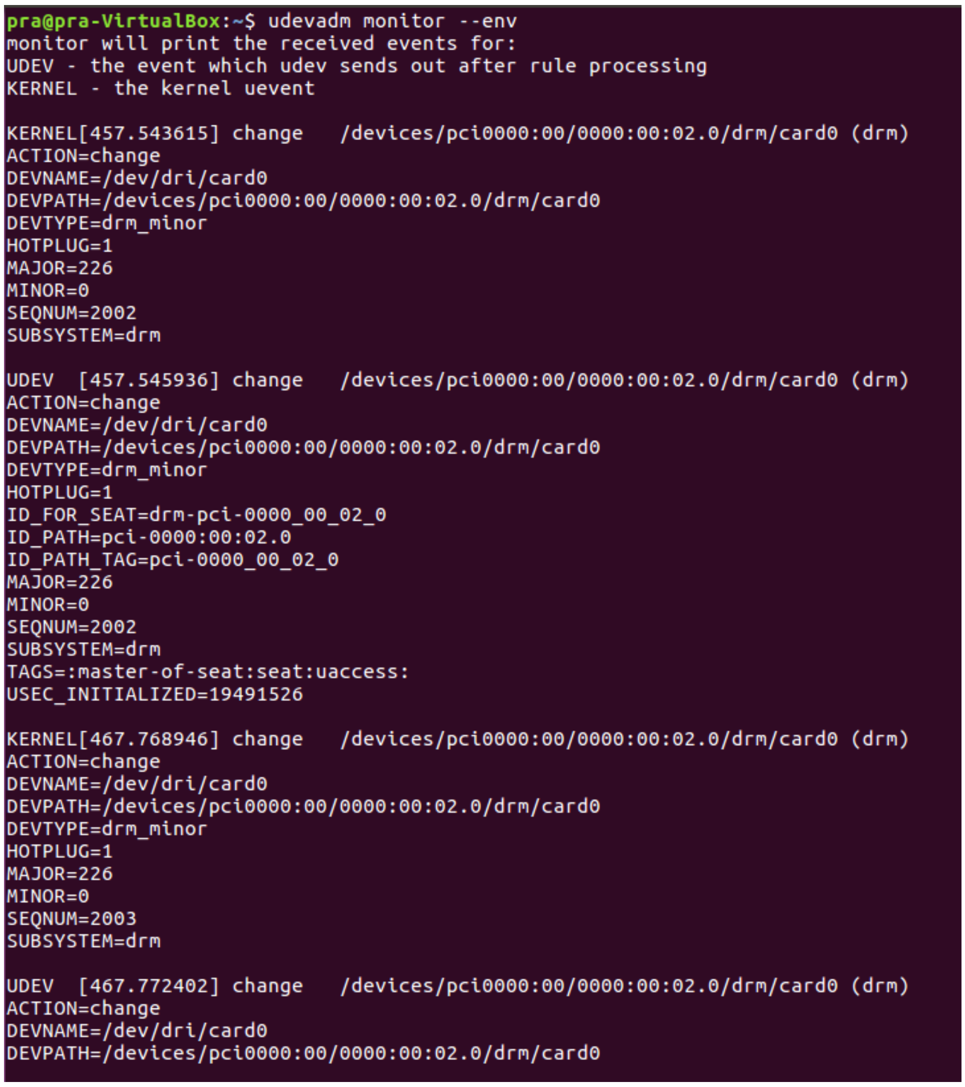
```
pi@raspberrypi: /sys/class/gpio
s the 'pi' user and type 'passwd' to set | This is a security risk - please logina
 a new password.                         | s the 'pi' user and type 'passwd' to se
                                         | t a new password.
pi@raspberrypi:/sys/class/gpio $ tvservi |
ce -M                                    | pi@raspberrypi:/sys/class/gpio $ 
Starting to monitor for HDMI events
[I] HDMI cable is unplugged
[I] HDMI is attached
[I] HDMI cable is unplugged
[I] HDMI is attached
```

Figure 9: Output of tvservice -M

vice_callback() which was the one actually reporting the hdmi states. Obviously,this function had to be traced down.I used a grep -ri on userland(elixir is a really useful tool,wish it was there for userland as well) to look for which files contained particular functions and reached vc_tvservice_defs.h and read through this as well.I went through all the includes but did not find any direct reference to mailboxes.I did see a few usages of message queues and also a few ioctl calls.This seemed a rather arduous process so I decided to directly search for the bcm mailbox read and write functions[14].Before that,I decided to use the utility vcmailbox available at opt/vc/bin[15].

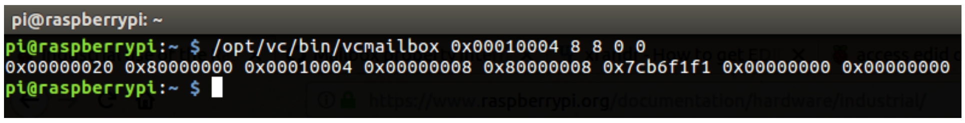
After reading the github mailbox property interface,I understood that there were a number of channels depending on what information one desired.[11] For a request by ARM and response by the videocore(VC),channel 8 was the appropriate one and a number of tags existed to further narrow down the query.Although mailbox is an elegant method of communication between the cores,in my opinion,it is rather poorly documented and can be understood only by reading the source code.Or it could be that the other parts are usually accompanied by examples rather than just an explanation of the internal structures.The github page was exhaustive in its explanation about the tags and I tried a simple one using the utility–get board serial.[16]This one worked and I was able to read the serial number after carefully analysing the response returned.I went through the source code of this utility expecting to see a direct include of the bcm mailbox file.To my surprise,all I saw was ioctl calls to/dev/vcio.It then struck me that the ioctl calls I had seen in tvservice could be related to mailboxes as well.

I came across an EDID tag [17] in the property interface page which seemed to be the appropriate tag to send.I tried sending this request using vcmailbox both when the hdmi cable was disconnected and connected but the output was not exactly decipherable.As I later discovered,this was due to an error in one of the inputs–the number of bytes to be queried for.At the time,I just used tvservice -d and wrote the edid to a file which I then read



```
pra@pra-VirtualBox:~$ udevadm monitor --env
monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel uevent

KERNEL[457.543615] change   /devices/pci0000:00/0000:00:02.0/drm/card0 (drm)
ACTION=change
DEVNAME=/dev/dri/card0
DEVPATH=/devices/pci0000:00/0000:00:02.0/drm/card0
DEVTYPE=drm_minor
HOTPLUG=1
MAJOR=226
MINOR=0
SEQNUM=2002
SUBSYSTEM=drm

UDEV  [457.545936] change   /devices/pci0000:00/0000:00:02.0/drm/card0 (drm)
ACTION=change
DEVNAME=/dev/dri/card0
DEVPATH=/devices/pci0000:00/0000:00:02.0/drm/card0
DEVTYPE=drm_minor
HOTPLUG=1
ID_FOR_SEAT=drm-pci-0000_00_02_0
ID_PATH=pci-0000:00:02.0
ID_PATH_TAG=pci-0000_00_02_0
MAJOR=226
MINOR=0
SEQNUM=2002
SUBSYSTEM=drm
TAGS=:master-of-seat:seat:uaccess:
USEC_INITIALIZED=19491526

KERNEL[467.768946] change   /devices/pci0000:00/0000:00:02.0/drm/card0 (drm)
ACTION=change
DEVNAME=/dev/dri/card0
DEVPATH=/devices/pci0000:00/0000:00:02.0/drm/card0
DEVTYPE=drm_minor
HOTPLUG=1
MAJOR=226
MINOR=0
SEQNUM=2003
SUBSYSTEM=drm

UDEV  [467.772402] change   /devices/pci0000:00/0000:00:02.0/drm/card0 (drm)
ACTION=change
DEVNAME=/dev/dri/card0
DEVPATH=/devices/pci0000:00/0000:00:02.0/drm/card0
```

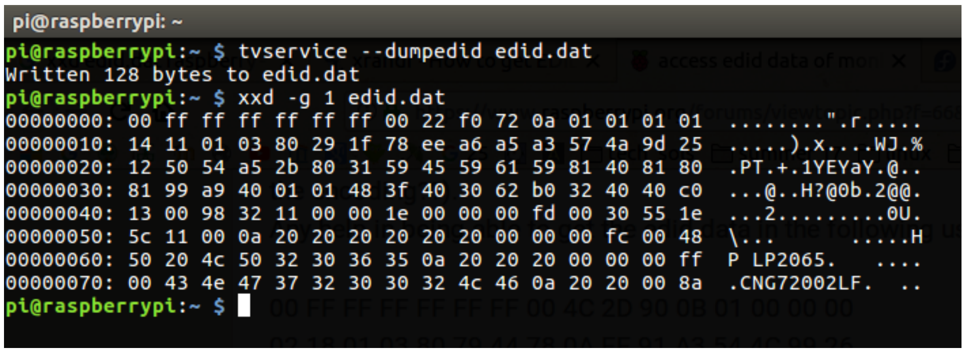
Figure 10: Uevent on connection of HDMI on the laptop



```
pi@raspberrypi: ~
pi@raspberrypi:~ $ /opt/vc/bin/vcmailbox 0x00010004 8 8 0 0
0x00000020 0x80000000 0x00010004 0x00000008 0x80000008 0x7cb6f1f1 0x00000000 0x00000000
pi@raspberrypi:~ $
```

Figure 11: Reading serial number using vcmailbox utility

in a hexadecimal format using xxd[18].From this,I could read the name of the device which matched the output of tvservice -n(which internally parses the edid to obtain the name).This could be used to address the problem of detecting whether it is a monitor that was connected to the hdmi cable or a projector.



```
pi@raspberrypi: ~
pi@raspberrypi:~ $ tvservice --dumpedid edid.dat
Written 128 bytes to edid.dat
pi@raspberrypi:~ $ xxd -g 1 edid.dat
00000000: 00 ff ff ff ff ff ff 00 22 f0 72 0a 01 01 01 01  ........".r.....
00000010: 14 11 01 03 80 29 1f 78 ee a6 a5 a3 57 4a 9d 25  .....).x....WJ.%
00000020: 12 50 54 a5 2b 80 31 59 45 59 61 59 81 40 81 80  .PT.+.1YEYaY.@..
00000030: 81 99 a9 40 01 01 48 3f 40 30 62 b0 32 40 40 c0  ...@..H?@0b.2@@.
00000040: 13 00 98 32 11 00 00 1e 00 00 00 fd 00 30 55 1e  ...2.........0U.
00000050: 5c 11 00 0a 20 20 20 20 20 20 00 00 00 fc 00 48  \...      .....H
00000060: 50 20 4c 50 32 30 36 35 0a 20 20 20 00 00 00 ff  P LP2065.   ....
00000070: 00 43 4e 47 37 32 30 30 32 4c 46 0a 20 20 00 8a  .CNG72002LF.  ..
pi@raspberrypi:~ $ █
```

Figure 12: EDID read using tvservice

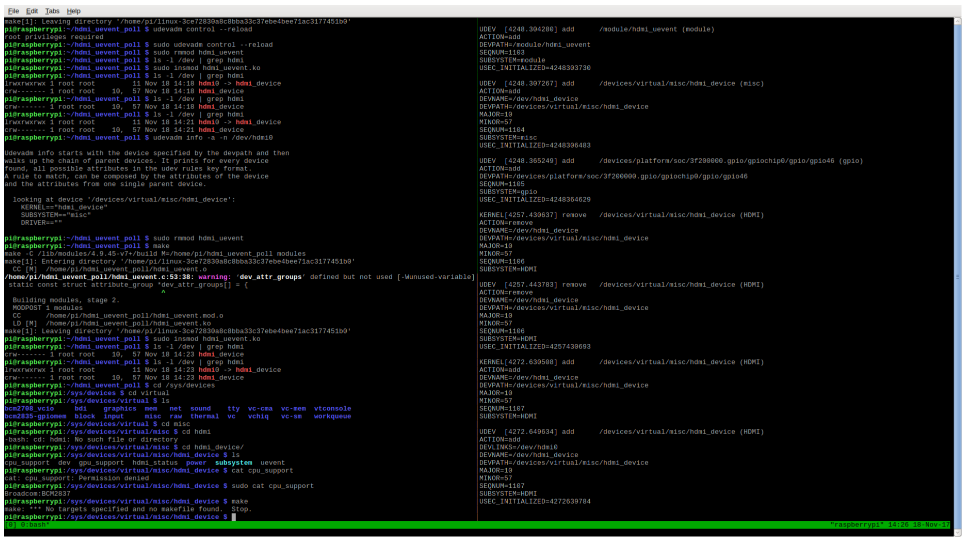
I also observed that there were quite a few utilities in /opt/vc.However,the documentation was limited and I didnt get time to explore all of them thoroughly.

Also,by this time,we had looked at the device tree file,and since we were using a Pi2,it turned out that pin 46 could be used to monitor the state of the hdmi cable.Thus,my explorations with mailbox were put to a halt to be continued later,if necessary.

Next,I proceeded to read about uevents and udev rules.[19, 20, 21, 22]The next small step was to generate a custom uevent using a kernel module. Initially,this module seemed problematic,but it was later found that the call to kobject_uevent_env was failing because the kset was NULL and it threw EINVAL[23].
Once this was fixed,a uevent was generated with parameters that were user-defined(before which a misc_device was created and registered) and a rules file was written to match and execute some trivial action upon a match.One observation here is that regardless of the values of the envp parameters(which were sysfs attributes) passed to kobject_uevent_env,the rules matched depending on the values of sysfs attributes and not these envp values.Basically,the rule succeeded even if the attribute value passed in envp was different from the one in the rule as long as the sysfs value matched.I tried a few things like creating a symlink on match etc.

3.4 Proposed solution

Having gone through all the components of the project,it was time to put them all together to come up with the solution.Two loadable kernel modules were envisioned for the same:One which would monitor pin 46 for a change(basically hdmi disconnect) and generate a uevent.Another module that was a customised version of lirc_rpi.c which would send the codes for power off.A rule would be written to match the uevent generated by the first



```
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
pi@raspberrypi:~/hdmi_uevent_poll $ udevadm control --reload
root privileges required
pi@raspberrypi:~/hdmi_uevent_poll $ sudo udevadm control --reload
pi@raspberrypi:~/hdmi_uevent_poll $ sudo rmmod hdmi_uevent
pi@raspberrypi:~/hdmi_uevent_poll $ ls -l /dev | grep hdmi
pi@raspberrypi:~/hdmi_uevent_poll $ sudo insmod hdmi_uevent.ko
pi@raspberrypi:~/hdmi_uevent_poll $ ls -l /dev | grep hdmi
lrwxrwxrwx 1 root root         11 Nov 18 14:18 hdmi0 -> hdmi_device
crw------- 1 root root    10,  57 Nov 18 14:18 hdmi_device
pi@raspberrypi:~/hdmi_uevent_poll $ ls -l /dev | grep hdmi
crw------- 1 root root    10,  57 Nov 18 14:18 hdmi_device
pi@raspberrypi:~/hdmi_uevent_poll $ ls -l /dev | grep hdmi
lrwxrwxrwx 1 root root         11 Nov 18 14:21 hdmi0 -> hdmi_device
crw------- 1 root root    10,  57 Nov 18 14:21 hdmi_device
pi@raspberrypi:~/hdmi_uevent_poll $ udevadm info -a -n /dev/hdmi0

Udevadm info starts with the device specified by the devpath and then
walks up the chain of parent devices. It prints for every device
found, all possible attributes in the udev rules key format.
A rule to match, can be composed by the attributes of the device
and the attributes from one single parent device.

  looking at device '/devices/virtual/misc/hdmi_device':
    KERNEL=="hdmi_device"
    SUBSYSTEM=="misc"
    DRIVER==""

pi@raspberrypi:~/hdmi_uevent_poll $ sudo rmmod hdmi_uevent
pi@raspberrypi:~/hdmi_uevent_poll $ make
make -C /lib/modules/4.9.45-v7+/build M=/home/pi/hdmi_uevent_poll modules
make[1]: Entering directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
  CC [M]  /home/pi/hdmi_uevent_poll/hdmi_uevent.o
/home/pi/hdmi_uevent_poll/hdmi_uevent.c:53:38: warning: ‘dev_attr_groups’ defined but not used [-Wunused-variable]
 static const struct attribute_group *dev_attr_groups[] = {
                                      ^
  Building modules, stage 2.
  MODPOST 1 modules
  CC      /home/pi/hdmi_uevent_poll/hdmi_uevent.mod.o
  LD [M]  /home/pi/hdmi_uevent_poll/hdmi_uevent.ko
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
pi@raspberrypi:~/hdmi_uevent_poll $ sudo insmod hdmi_uevent.ko
pi@raspberrypi:~/hdmi_uevent_poll $ ls -l /dev | grep hdmi
crw------- 1 root root    10,  57 Nov 18 14:23 hdmi_device
pi@raspberrypi:~/hdmi_uevent_poll $ ls -l /dev | grep hdmi
lrwxrwxrwx 1 root root         11 Nov 18 14:23 hdmi0 -> hdmi_device
crw------- 1 root root    10,  57 Nov 18 14:23 hdmi_device
pi@raspberrypi:~/hdmi_uevent_poll $ cd /sys/devices
pi@raspberrypi:/sys/devices $ cd virtual
pi@raspberrypi:/sys/devices/virtual $ ls
bcm2708_vcio     bdi    graphics  mem   net  sound    tty  vc-cma  vc-mem  vtconsole
bcm2835-gpiomem  block  input     misc  raw  thermal  vc   vchiq   vc-sm   workqueue
pi@raspberrypi:/sys/devices/virtual $ cd misc
pi@raspberrypi:/sys/devices/virtual/misc $ cd hdmi
-bash: cd: hdmi: No such file or directory
pi@raspberrypi:/sys/devices/virtual/misc $ cd hdmi_device/
pi@raspberrypi:/sys/devices/virtual/misc/hdmi_device $ ls
cpu_support  dev  gpu_support  hdmi_status  power  subsystem  uevent
pi@raspberrypi:/sys/devices/virtual/misc/hdmi_device $ cat cpu_support
cat: cpu_support: Permission denied
pi@raspberrypi:/sys/devices/virtual/misc/hdmi_device $ sudo cat cpu_support
Broadcom:BCM2837
pi@raspberrypi:/sys/devices/virtual/misc/hdmi_device $ make
make: *** No targets specified and no makefile found.  Stop.
pi@raspberrypi:/sys/devices/virtual/misc/hdmi_device $
[0] 0:bash*
```

```
UDEV  [4248.304280] add      /module/hdmi_uevent (module)
ACTION=add
DEVPATH=/module/hdmi_uevent
SEQNUM=1103
SUBSYSTEM=module
USEC_INITIALIZED=4248303730

UDEV  [4248.307267] add      /devices/virtual/misc/hdmi_device (misc)
ACTION=add
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
SEQNUM=1104
SUBSYSTEM=misc
USEC_INITIALIZED=4248306483

UDEV  [4248.365249] add      /devices/platform/soc/3f200000.gpio/gpiochip0/gpio/gpio46 (gpio)
ACTION=add
DEVPATH=/devices/platform/soc/3f200000.gpio/gpiochip0/gpio/gpio46
SEQNUM=1105
SUBSYSTEM=gpio
USEC_INITIALIZED=4248364629

KERNEL[4257.430637] remove   /devices/virtual/misc/hdmi_device (HDMI)
ACTION=remove
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
SEQNUM=1106
SUBSYSTEM=HDMI

UDEV  [4257.443783] remove   /devices/virtual/misc/hdmi_device (HDMI)
ACTION=remove
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
SEQNUM=1106
SUBSYSTEM=HDMI
USEC_INITIALIZED=4257430693

KERNEL[4272.630508] add      /devices/virtual/misc/hdmi_device (HDMI)
ACTION=add
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
SEQNUM=1107
SUBSYSTEM=HDMI

UDEV  [4272.649634] add      /devices/virtual/misc/hdmi_device (HDMI)
ACTION=add
DEVLINKS=/dev/hdmi0
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
SEQNUM=1107
SUBSYSTEM=HDMI
USEC_INITIALIZED=4272639784

"raspberrypi" 14:26 18-Nov-17
```

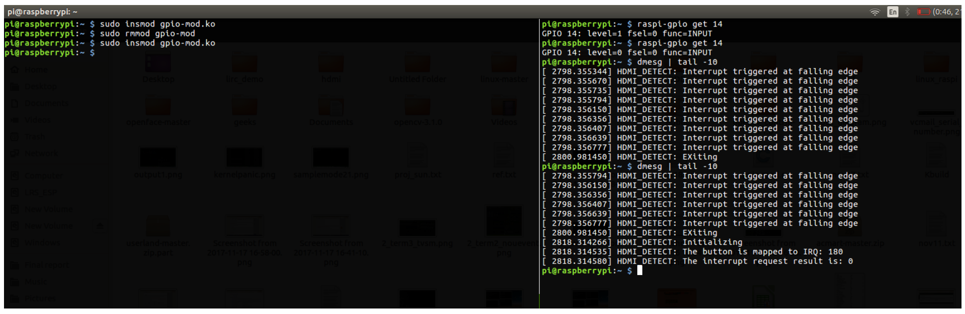
Figure 13: Uevent generation

module upon which the second module would be loaded.

3.4.1 LKM 1: hdmi_uevent.ko

The device tree revealed that pin 46 could be used instead of the much more complicated mailbox to monitor hdmi.A simple raspi-gpio get on 46 revealed that the pin was active low–it had a value of 1 when the hdmi cable was disconnected and a value of 0 when connected.

The purpose of this module was to detect a change in GPIO pin 46–basically a rising transition in the pin.I wrote an interrupt based module that would trigger a function at the rising edge.[24]It worked for other pins but sadly not 46.I remember investigating why this did not work by reading up the reference manuals[25][26] but I do not remember the exact reason as to why this happened.I did not document this at the time and now, I cannot recollect much except that this approach did not work.



```
pi@raspberrypi:~ $ sudo insmod gpio-mod.ko
pi@raspberrypi:~ $ sudo rmmod gpio-mod
pi@raspberrypi:~ $ sudo insmod gpio-mod.ko
pi@raspberrypi:~ $
```

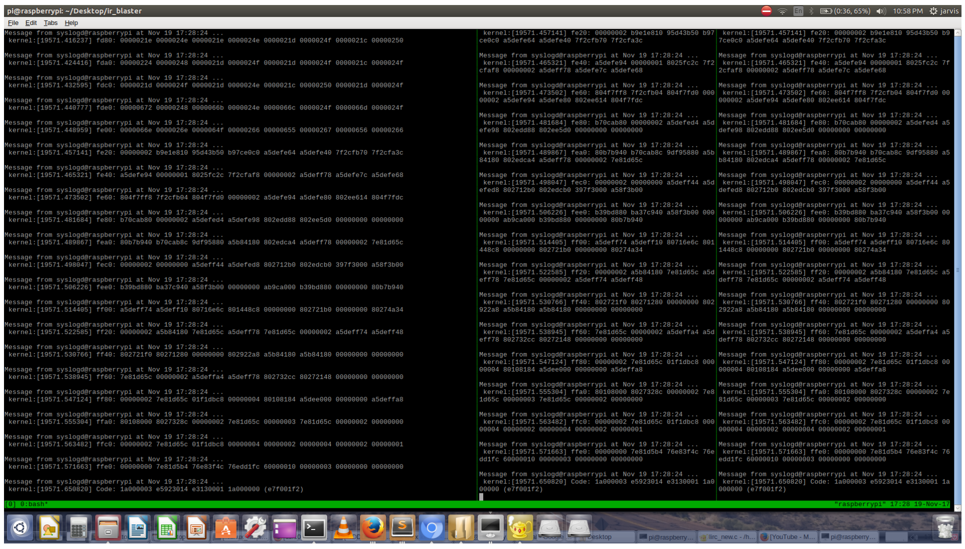
```
pi@raspberrypi:~ $ raspi-gpio get 14
GPIO 14: level=1 fsel=0 func=INPUT
pi@raspberrypi:~ $ raspi-gpio get 14
GPIO 14: level=0 fsel=0 func=INPUT
pi@raspberrypi:~ $ dmesg | tail -10
[ 2798.355344] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.355670] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.355735] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.355794] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356150] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356356] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356407] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356639] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356777] HDMI_DETECT: Interrupt triggered at falling edge
[ 2800.981450] HDMI_DETECT: EXiting
pi@raspberrypi:~ $ dmesg | tail -10
[ 2798.355794] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356150] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356356] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356407] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356639] HDMI_DETECT: Interrupt triggered at falling edge
[ 2798.356777] HDMI_DETECT: Interrupt triggered at falling edge
[ 2800.981450] HDMI_DETECT: EXiting
[ 2818.314266] HDMI_DETECT: Initializing
[ 2818.314535] HDMI_DETECT: The button is mapped to IRQ: 180
[ 2818.314580] HDMI_DETECT: The interrupt request result is: 0
pi@raspberrypi:~ $
```

Figure 14: Interrupt generation on GPIO 14

The second approach was to poll the pin,which in this case sufficed,because the transition wasnt all that frequent/fast anyway.This module internally used a timer which would expire every second and call its handler function which would then generate a uevent.A couple of interesting problems here:Initially,when the module was compiled,it threw up an indecipherable error from a header file.Later,on looking at the class example for the timer,it was found that the timer function took a dummy argument.Upon including this,voila,the error went away.Also,I forgot to delete the timer in the exit function and inserted the module,after which the Pi froze.Once all this was taken care of,the module worked as expected.

3.4.2 LKM 2: lirc_new.ko

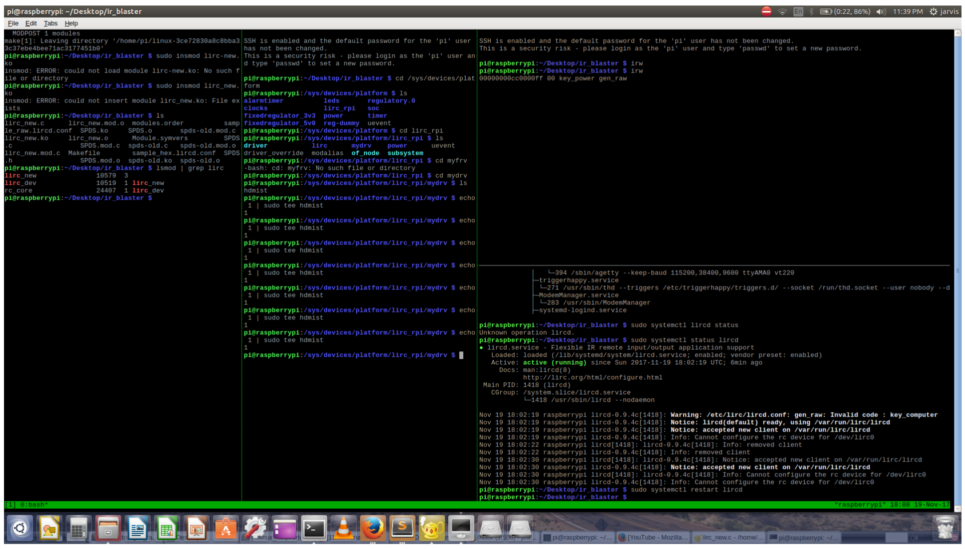
This was an attempt to write a custom lirc driver which sends the power sequence.After going through the source code of lirc_rpi[27] which seemed rather incomprehensible at first,an initial naive attempt was made to write a module from scratch.The module compiled and could be inserted,but when the pulses and spaces it generated were monitored using mode2,they did not make any sense at all.This necessitated going back to the source and finding out how this naive module differed from the actual lirc_rpi.The differences were humongous and in a way,it made me appreciate how well-written the original one was.This time,instead of reinventing the wheel,I chose to use what was already available to simplify my task.During this process,I once did a kfree wrongly.After inserting the module,all the tmux terminals at once spouted out numbers saying message from syslog.Not only that,even the host terminal that was used to ssh into the pi was filled with numbers.The Pi stopped responding.I instantly freaked out, wondering if this was the infamous kernel panic.My first thought was that my code was not version controlled or backed up anywhere and that I would lose it all if the Pi refused to boot.It thankfully did,and I thanked my stars and the people who wrote Linux and made it so robust.Also,I immediately put up the code on github.



```
Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.416237] fd80: 0000021e 0000024e 0000021e 0000024e 0000021d 0000024f 0000021c 00000250

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.424416] fda0: 00000224 00000248 0000021d 0000024f 0000021d 0000024f 0000021c 0000024f

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.432595] fdc0: 0000021d 0000024f 0000021d 0000024e 0000021c 00000250 0000021d 0000024f

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.440777] fde0: 00000672 00000248 0000066b 0000024e 0000066c 0000024f 0000066d 0000024f

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.448959] fe00: 0000066e 0000026e 0000064f 00000266 00000655 00000267 00000656 00000266

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.457141] fe20: 00000002 b9e1e810 95d43b50 b97ce0c0 a5defe64 a5defe40 7f2cfb70 7f2cfa3c

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.465321] fe40: a5defe94 00000001 8025fc2c 7f2cfaf8 00000002 a5deff78 a5defe7c a5defe68

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.473502] fe60: 804f7ff8 7f2cfb04 804f7fd0 00000002 a5defe94 a5defe80 802ee614 804f7fdc

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.481684] fe80: b70cab80 00000002 a5defed4 a5defe98 802edd88 802ee5d0 00000000 00000000

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.489867] fea0: 80b7b940 b70cab8c 9df95880 a5b84180 802edca4 a5deff78 00000002 7e81d65c

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.498047] fec0: 00000002 00000000 a5deff44 a5defed8 802712b0 802edcb0 397f3000 a58f3b00

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.506226] fee0: b39bd880 ba37c940 a58f3b00 00000000 ab9ca000 b39bd880 00000000 80b7b940

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.514405] ff00: a5deff74 a5deff10 80716e6c 801448c8 00000000 802721b0 00000000 80274a34

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.522585] ff20: 00000002 a5b84180 7e81d65c a5deff78 7e81d65c 00000002 a5deff74 a5deff48

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.530766] ff40: 802721f0 80271280 00000000 802922a8 a5b84180 a5b84180 00000000 00000000

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.538945] ff60: 7e81d65c 00000002 a5deffa4 a5deff78 802732cc 80272148 00000000 00000000

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.547124] ff80: 00000002 7e81d65c 01f1dbc8 00000004 80108184 a5dee000 00000000 a5deffa8

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.555304] ffa0: 80108000 8027328c 00000002 7e81d65c 00000003 7e81d65c 00000002 00000000

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.563482] ffc0: 00000002 7e81d65c 01f1dbc8 00000004 00000002 00000004 00000002 00000001

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.571663] ffe0: 00000000 7e81d5b4 76e83f4c 76edd1fc 60000010 00000003 00000000 00000000

Message from syslogd@raspberrypi at Nov 19 17:28:24 ...
 kernel:[19571.650820] Code: 1a000003 e5923014 e3130001 1a000000 (e7f001f2)
[0] 0:bash*
```

Figure 15: Close to panic!

Once the module was complete,I checked the outputs in the raw form using mode2 and as a hex code in irw.Both were successful.



```
  MODPOST 1 modules
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba3
3c37ebe4bee71ac3177451b0'
pi@raspberrypi:~/Desktop/ir_blaster $ sudo insmod lirc-new.
ko
insmod: ERROR: could not load module lirc-new.ko: No such f
ile or directory
pi@raspberrypi:~/Desktop/ir_blaster $ sudo insmod lirc_new.
ko
insmod: ERROR: could not insert module lirc_new.ko: File ex
ists
pi@raspberrypi:~/Desktop/ir_blaster $ ls
lirc_new.c      lirc_new.mod.o  modules.order         samp
le_raw.lircd.conf  SPDS.ko     SPDS.o       spds-old.mod.c
lirc_new.ko     lirc_new.o      Module.symvers        SPDS
.c                 SPDS.mod.c  spds-old.c   spds-old.mod.o
lirc_new.mod.c  Makefile        sample_hex.lircd.conf  SPDS
.h                 SPDS.mod.o  spds-old.ko  spds-old.o
pi@raspberrypi:~/Desktop/ir_blaster $ lsmod | grep lirc
lirc_new               10579  3
lirc_dev               10519  1 lirc_new
rc_core                24407  1 lirc_dev
pi@raspberrypi:~/Desktop/ir_blaster $
```

```
SSH is enabled and the default password for the 'pi' user
has not been changed.
This is a security risk - please login as the 'pi' user an
d type 'passwd' to set a new password.

pi@raspberrypi:~/Desktop/ir_blaster $ cd /sys/devices/plat
form
pi@raspberrypi:/sys/devices/platform $ ls
alarmtimer          leds       regulatory.0
clocks              lirc_rpi   soc
fixedregulator_3v3  power      timer
fixedregulator_5v0  reg-dummy  uevent
pi@raspberrypi:/sys/devices/platform $ cd lirc_rpi
pi@raspberrypi:/sys/devices/platform/lirc_rpi $ ls
driver           lirc      mydrv    power      uevent
driver_override  modalias  of_node  subsystem
pi@raspberrypi:/sys/devices/platform/lirc_rpi $ cd myfrv
-bash: cd: myfrv: No such file or directory
pi@raspberrypi:/sys/devices/platform/lirc_rpi $ cd mydrv
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ ls
hdmist
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ echo
 1 | sudo tee hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $
```

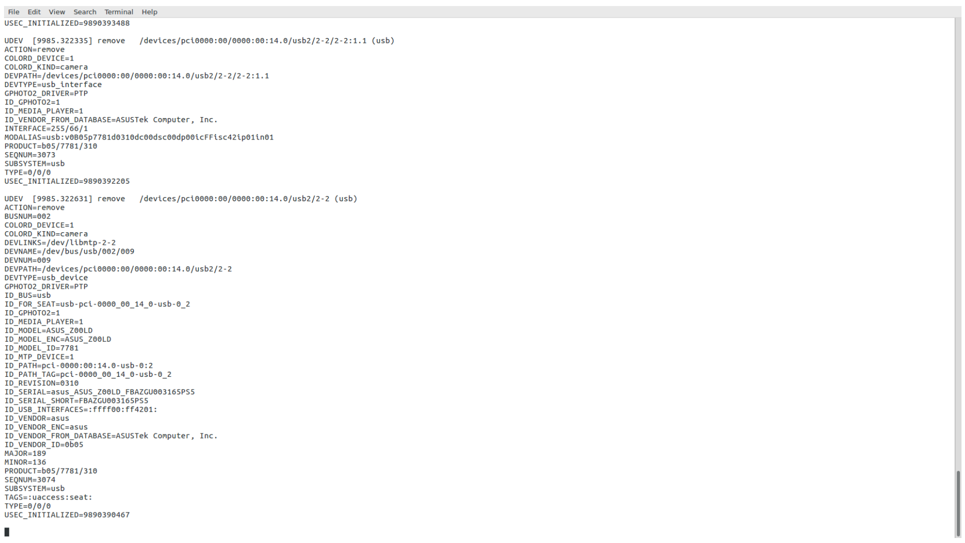
```
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set a new password.

pi@raspberrypi:~/Desktop/ir_blaster $ irw
pi@raspberrypi:~/Desktop/ir_blaster $ irw
00000000cc0000ff 00 key_power gen_raw
```

```
           │  └─394 /sbin/agetty --keep-baud 115200,38400,9600 ttyAMA0 vt220
           ├─triggerhappy.service
           │ └─271 /usr/sbin/thd --triggers /etc/triggerhappy/triggers.d/ --socket /run/thd.socket --user nobody --d
           ├─ModemManager.service
           │ └─283 /usr/sbin/ModemManager
           ├─systemd-logind.service

pi@raspberrypi:~/Desktop/ir_blaster $ sudo systemctl lircd status
Unknown operation lircd.
pi@raspberrypi:~/Desktop/ir_blaster $ sudo systemctl status lircd
● lircd.service - Flexible IR remote input/output application support
   Loaded: loaded (/lib/systemd/system/lircd.service; enabled; vendor preset: enabled)
   Active: active (running) since Sun 2017-11-19 18:02:19 UTC; 6min ago
     Docs: man:lircd(8)
           http://lirc.org/html/configure.html
 Main PID: 1418 (lircd)
   CGroup: /system.slice/lircd.service
           └─1418 /usr/sbin/lircd --nodaemon

Nov 19 18:02:19 raspberrypi lircd-0.9.4c[1418]: Warning: /etc/lirc/lircd.conf: gen_raw: Invalid code : key_computer
Nov 19 18:02:19 raspberrypi lircd-0.9.4c[1418]: Notice: lircd(default) ready, using /var/run/lirc/lircd
Nov 19 18:02:19 raspberrypi lircd-0.9.4c[1418]: Notice: accepted new client on /var/run/lirc/lircd
Nov 19 18:02:19 raspberrypi lircd-0.9.4c[1418]: Info: Cannot configure the rc device for /dev/lirc0
Nov 19 18:02:22 raspberrypi lircd[1418]: lircd-0.9.4c[1418]: Info: removed client
Nov 19 18:02:22 raspberrypi lircd-0.9.4c[1418]: Info: removed client
Nov 19 18:02:30 raspberrypi lircd[1418]: lircd-0.9.4c[1418]: Notice: accepted new client on /var/run/lirc/lircd
Nov 19 18:02:30 raspberrypi lircd-0.9.4c[1418]: Notice: accepted new client on /var/run/lirc/lircd
Nov 19 18:02:30 raspberrypi lircd[1418]: lircd-0.9.4c[1418]: Info: Cannot configure the rc device for /dev/lirc0
Nov 19 18:02:30 raspberrypi lircd-0.9.4c[1418]: Info: Cannot configure the rc device for /dev/lirc0
pi@raspberrypi:~/Desktop/ir_blaster $ sudo systemctl restart lircd
pi@raspberrypi:~/Desktop/ir_blaster $
[1] 0:bash*                                                                    "raspberrypi" 18:08 19-Nov-17
```

Figure 16: Functioning of lirc_new.ko demonstrated using a send-receive loop

3.4.3 Integration

Now that both modules were ready,all that was left was to integrate them.As mentioned in the proposed solution,when the first module generated a uevent,the second module had to be loaded. I think the origin of this idea is worth mentioning here.When a phone was connected to the usb port of the laptop,it was logical that its driver had to be loaded.When udevadm monitor was run,it showed up uevents that had a field called modalias.[28]



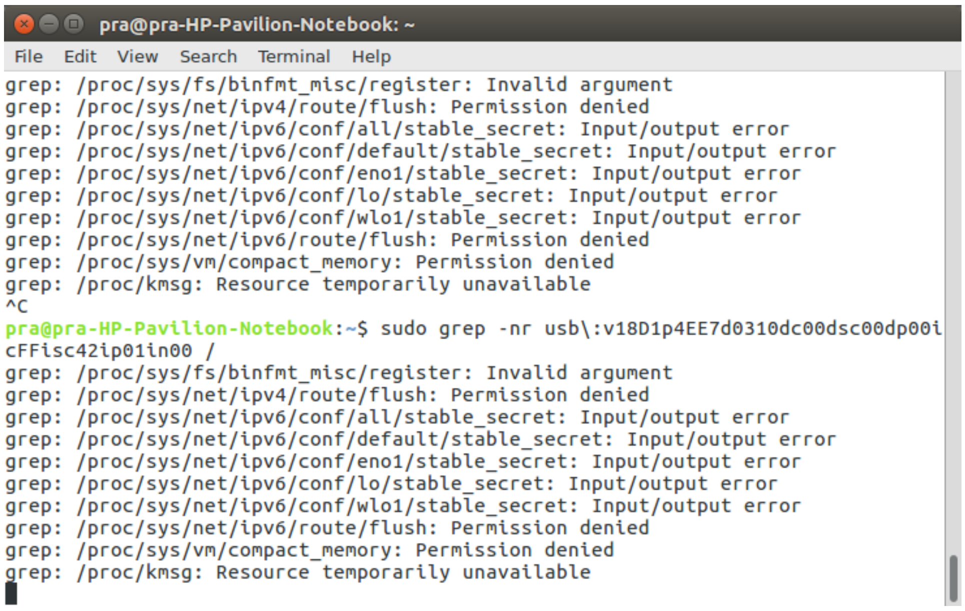
```
USEC_INITIALIZED=9890393488

UDEV  [9985.322335] remove   /devices/pci0000:00/0000:00:14.0/usb2/2-2/2-2:1.1 (usb)
ACTION=remove
COLORD_DEVICE=1
COLORD_KIND=camera
DEVPATH=/devices/pci0000:00/0000:00:14.0/usb2/2-2/2-2:1.1
DEVTYPE=usb_interface
GPHOTO2_DRIVER=PTP
ID_GPHOTO2=1
ID_MEDIA_PLAYER=1
ID_VENDOR_FROM_DATABASE=ASUSTek Computer, Inc.
INTERFACE=255/66/1
MODALIAS=usb:v0B05p7781d0310dc00dsc00dp00icFFisc42ip01in01
PRODUCT=b05/7781/310
SEQNUM=3073
SUBSYSTEM=usb
TYPE=0/0/0
USEC_INITIALIZED=9890392205

UDEV  [9985.322631] remove   /devices/pci0000:00/0000:00:14.0/usb2/2-2 (usb)
ACTION=remove
BUSNUM=002
COLORD_DEVICE=1
COLORD_KIND=camera
DEVLINKS=/dev/libmtp-2-2
DEVNAME=/dev/bus/usb/002/009
DEVNUM=009
DEVPATH=/devices/pci0000:00/0000:00:14.0/usb2/2-2
DEVTYPE=usb_device
GPHOTO2_DRIVER=PTP
ID_BUS=usb
ID_FOR_SEAT=usb-pci-0000_00_14_0-usb-0_2
ID_GPHOTO2=1
ID_MEDIA_PLAYER=1
ID_MODEL=ASUS_Z00LD
ID_MODEL_ENC=ASUS_Z00LD
ID_MODEL_ID=7781
ID_MTP_DEVICE=1
ID_PATH=pci-0000:00:14.0-usb-0:2
ID_PATH_TAG=pci-0000_00_14_0-usb-0_2
ID_REVISION=0310
ID_SERIAL=asus_ASUS_Z00LD_FBAZGU003165PS5
ID_SERIAL_SHORT=FBAZGU003165PS5
ID_USB_INTERFACES=:ffff00:ff4201:
ID_VENDOR=asus
ID_VENDOR_ENC=asus
ID_VENDOR_FROM_DATABASE=ASUSTek Computer, Inc.
ID_VENDOR_ID=0b05
MAJOR=189
MINOR=136
PRODUCT=b05/7781/310
SEQNUM=3074
SUBSYSTEM=usb
TAGS=:uaccess:seat:
TYPE=0/0/0
USEC_INITIALIZED=9890390467
```

Figure 17: Uevent generated on laptop when phoen is plugged in

Upon further reading,it came to be known that this is a rather important and useful parameter.When depmod is run,it accumulates all of these into a file called modules.alias and this is the key to loading modules in a very simple manner using udev rules:kmod load $envMODALIAS.For this purpose,bhalu was taken to be the sample module and given a modalias and installed under /lib/modules/$(uname -r)/kernel using the INSTALL_MOD_DIR flag.Thereafter,depmod was run to populate all the appropriate files.Finally,when the other module generated a uevent that was picked up by udev and matched with the rules,bhalu was loaded using this idea.

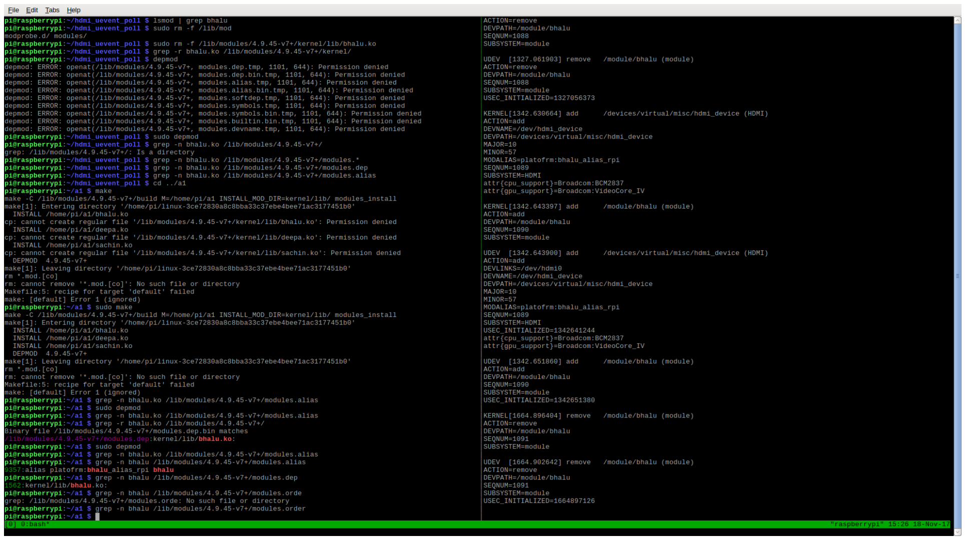
Now,all that remained was to do the same using the 2 custom modules.Easier said than done.There problem faced is outlined below:

So,the idea was to load the lirc_new module upon hdmi disconnect and unload it soon after sending the power codes.Therefore,the module_exit() function was called from within the power_off() function.However,this did not completely remove the module and it was still present when checked using lsmod.Upon further doing an rmmod it resulted in a segmentation fault.



```
grep: /proc/sys/fs/binfmt_misc/register: Invalid argument
grep: /proc/sys/net/ipv4/route/flush: Permission denied
grep: /proc/sys/net/ipv6/conf/all/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/default/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/eno1/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/lo/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/wlo1/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/route/flush: Permission denied
grep: /proc/sys/vm/compact_memory: Permission denied
grep: /proc/kmsg: Resource temporarily unavailable
^C
pra@pra-HP-Pavilion-Notebook:~$ sudo grep -nr usb\:v18D1p4EE7d0310dc00dsc00dp00i
cFFisc42ip01in00 /
grep: /proc/sys/fs/binfmt_misc/register: Invalid argument
grep: /proc/sys/net/ipv4/route/flush: Permission denied
grep: /proc/sys/net/ipv6/conf/all/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/default/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/eno1/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/lo/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/conf/wlo1/stable_secret: Input/output error
grep: /proc/sys/net/ipv6/route/flush: Permission denied
grep: /proc/sys/vm/compact_memory: Permission denied
grep: /proc/kmsg: Resource temporarily unavailable
```

Figure 18: Searching for the modalias

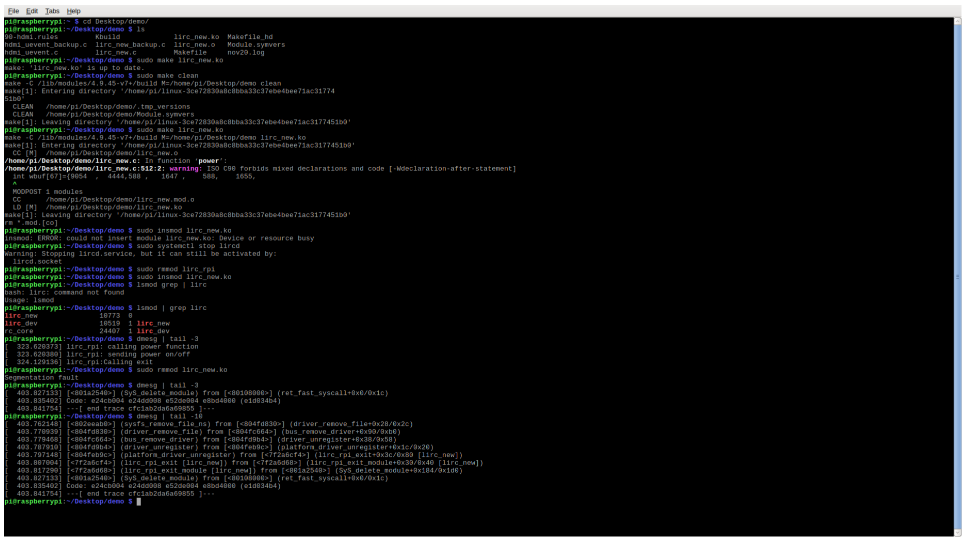


```
pi@raspberrypi:~/hdmi_uevent_poll $ lsmod | grep bhalu
pi@raspberrypi:~/hdmi_uevent_poll $ sudo rm -f /lib/mod
modprobe.d/ modules/
pi@raspberrypi:~/hdmi_uevent_poll $ sudo rm -f /lib/modules/4.9.45-v7+/kernel/lib/bhalu.ko
pi@raspberrypi:~/hdmi_uevent_poll $ grep -r bhalu.ko /lib/modules/4.9.45-v7+/kernel/
pi@raspberrypi:~/hdmi_uevent_poll $ depmod
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.dep.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.dep.bin.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.alias.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.alias.bin.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.softdep.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.symbols.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.symbols.bin.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.builtin.bin.tmp, 1101, 644): Permission denied
depmod: ERROR: openat(/lib/modules/4.9.45-v7+, modules.devname.tmp, 1101, 644): Permission denied
pi@raspberrypi:~/hdmi_uevent_poll $ sudo depmod
pi@raspberrypi:~/hdmi_uevent_poll $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/
grep: /lib/modules/4.9.45-v7+/: Is a directory
pi@raspberrypi:~/hdmi_uevent_poll $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/modules.*
pi@raspberrypi:~/hdmi_uevent_poll $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/modules.dep
pi@raspberrypi:~/hdmi_uevent_poll $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/modules.alias
pi@raspberrypi:~/hdmi_uevent_poll $ cd ../a1
pi@raspberrypi:~/a1 $ make
make -C /lib/modules/4.9.45-v7+/build M=/home/pi/a1 INSTALL_MOD_DIR=kernel/lib/ modules_install
make[1]: Entering directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
  INSTALL /home/pi/a1/bhalu.ko
cp: cannot create regular file '/lib/modules/4.9.45-v7+/kernel/lib/bhalu.ko': Permission denied
  INSTALL /home/pi/a1/deepa.ko
cp: cannot create regular file '/lib/modules/4.9.45-v7+/kernel/lib/deepa.ko': Permission denied
  INSTALL /home/pi/a1/sachin.ko
cp: cannot create regular file '/lib/modules/4.9.45-v7+/kernel/lib/sachin.ko': Permission denied
  DEPMOD  4.9.45-v7+
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
rm *.mod.[co]
rm: cannot remove '*.mod.[co]': No such file or directory
Makefile:5: recipe for target 'default' failed
make: [default] Error 1 (ignored)
pi@raspberrypi:~/a1 $ sudo make
make -C /lib/modules/4.9.45-v7+/build M=/home/pi/a1 INSTALL_MOD_DIR=kernel/lib/ modules_install
make[1]: Entering directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
  INSTALL /home/pi/a1/bhalu.ko
  INSTALL /home/pi/a1/deepa.ko
  INSTALL /home/pi/a1/sachin.ko
  DEPMOD  4.9.45-v7+
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
rm *.mod.[co]
rm: cannot remove '*.mod.[co]': No such file or directory
Makefile:5: recipe for target 'default' failed
make: [default] Error 1 (ignored)
pi@raspberrypi:~/a1 $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/modules.alias
pi@raspberrypi:~/a1 $ sudo depmod
pi@raspberrypi:~/a1 $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/modules.alias
pi@raspberrypi:~/a1 $ grep -r bhalu.ko /lib/modules/4.9.45-v7+/
Binary file /lib/modules/4.9.45-v7+/modules.dep.bin matches
/lib/modules/4.9.45-v7+/modules.dep:kernel/lib/bhalu.ko:
pi@raspberrypi:~/a1 $ sudo depmod
pi@raspberrypi:~/a1 $ grep -n bhalu.ko /lib/modules/4.9.45-v7+/modules.alias
pi@raspberrypi:~/a1 $ grep -n bhalu /lib/modules/4.9.45-v7+/modules.alias
9357:alias platofrm:bhalu_alias_rpi bhalu
pi@raspberrypi:~/a1 $ grep -n bhalu /lib/modules/4.9.45-v7+/modules.dep
1562:kernel/lib/bhalu.ko:
pi@raspberrypi:~/a1 $ grep -n bhalu /lib/modules/4.9.45-v7+/modules.orde
grep: /lib/modules/4.9.45-v7+/modules.orde: No such file or directory
pi@raspberrypi:~/a1 $ grep -n bhalu /lib/modules/4.9.45-v7+/modules.order
pi@raspberrypi:~/a1 $
[0] 0:bash*
```

```
ACTION=remove
DEVPATH=/module/bhalu
SEQNUM=1088
SUBSYSTEM=module

UDEV  [1327.061903] remove   /module/bhalu (module)
ACTION=remove
DEVPATH=/module/bhalu
SEQNUM=1088
SUBSYSTEM=module
USEC_INITIALIZED=1327056373

KERNEL[1342.630664] add      /devices/virtual/misc/hdmi_device (HDMI)
ACTION=add
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
MODALIAS=platofrm:bhalu_alias_rpi
SEQNUM=1089
SUBSYSTEM=HDMI
attr{cpu_support}=Broadcom:BCM2837
attr{gpu_support}=Broadcom:VideoCore_IV

KERNEL[1342.643397] add      /module/bhalu (module)
ACTION=add
DEVPATH=/module/bhalu
SEQNUM=1090
SUBSYSTEM=module

UDEV  [1342.643900] add      /devices/virtual/misc/hdmi_device (HDMI)
ACTION=add
DEVLINKS=/dev/hdmi0
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
MODALIAS=platofrm:bhalu_alias_rpi
SEQNUM=1089
SUBSYSTEM=HDMI
USEC_INITIALIZED=1342641244
attr{cpu_support}=Broadcom:BCM2837
attr{gpu_support}=Broadcom:VideoCore_IV

UDEV  [1342.651860] add      /module/bhalu (module)
ACTION=add
DEVPATH=/module/bhalu
SEQNUM=1090
SUBSYSTEM=module
USEC_INITIALIZED=1342651380

KERNEL[1664.896404] remove   /module/bhalu (module)
ACTION=remove
DEVPATH=/module/bhalu
SEQNUM=1091
SUBSYSTEM=module

UDEV  [1664.902642] remove   /module/bhalu (module)
ACTION=remove
DEVPATH=/module/bhalu
SEQNUM=1091
SUBSYSTEM=module
USEC_INITIALIZED=1664897126

"raspberrypi" 15:26 18-Nov-17
```

Figure 19: using modalias to load a module upon uevent generation

I then removed the module_exit() call from the power function and found that I could insert and remove it without any issues.Also,the module no longer showed up in lsmod.Im not sure if this line of reasoning is correct,but it seemed that modprobe -r was doing something more than just calling the exit function.I remember reading something about modprobe taking care of reference counts,but that link went unbookmarked as well and I cant cite



```
pi@raspberrypi:~ $ cd Desktop/demo/
pi@raspberrypi:~/Desktop/demo $ ls
90-hdmi.rules         Kbuild              lirc_new.ko  Makefile_hd
hdmi_uevent_backup.c  lirc_new_backup.c  lirc_new.o   Module.symvers
hdmi_uevent.c         lirc_new.c          Makefile     nov20.log
pi@raspberrypi:~/Desktop/demo $ sudo make lirc_new.ko
make: 'lirc_new.ko' is up to date.
pi@raspberrypi:~/Desktop/demo $ sudo make clean
make -C /lib/modules/4.9.45-v7+/build M=/home/pi/Desktop/demo clean
make[1]: Entering directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac31774
51b0'
  CLEAN   /home/pi/Desktop/demo/.tmp_versions
  CLEAN   /home/pi/Desktop/demo/Module.symvers
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
pi@raspberrypi:~/Desktop/demo $ sudo make lirc_new.ko
make -C /lib/modules/4.9.45-v7+/build M=/home/pi/Desktop/demo lirc_new.ko
make[1]: Entering directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
  CC [M]  /home/pi/Desktop/demo/lirc_new.o
/home/pi/Desktop/demo/lirc_new.c: In function ‘power’:
/home/pi/Desktop/demo/lirc_new.c:512:2: warning: ISO C90 forbids mixed declarations and code [-Wdeclaration-after-statement]
  int wbuf[67]={9054  ,  4444,588 ,   1647 ,    588,    1655,
  ^
  MODPOST 1 modules
  CC      /home/pi/Desktop/demo/lirc_new.mod.o
  LD [M]  /home/pi/Desktop/demo/lirc_new.ko
make[1]: Leaving directory '/home/pi/linux-3ce72830a8c8bba33c37ebe4bee71ac3177451b0'
rm *.mod.[co]
pi@raspberrypi:~/Desktop/demo $ sudo insmod lirc_new.ko
insmod: ERROR: could not insert module lirc_new.ko: Device or resource busy
pi@raspberrypi:~/Desktop/demo $ sudo systemctl stop lircd
Warning: Stopping lircd.service, but it can still be activated by:
  lircd.socket
pi@raspberrypi:~/Desktop/demo $ sudo rmmod lirc_rpi
pi@raspberrypi:~/Desktop/demo $ sudo insmod lirc_new.ko
pi@raspberrypi:~/Desktop/demo $ lsmod grep | lirc
bash: lirc: command not found
Usage: lsmod
pi@raspberrypi:~/Desktop/demo $ lsmod | grep lirc
lirc_new               10773  0
lirc_dev               10519  1 lirc_new
rc_core                24407  1 lirc_dev
pi@raspberrypi:~/Desktop/demo $ dmesg | tail -3
[  323.620373] lirc_rpi: calling power function
[  323.620380] lirc_rpi: sending power on/off
[  324.129136] lirc_rpi:Calling exit
pi@raspberrypi:~/Desktop/demo $ sudo rmmod lirc_new.ko
Segmentation fault
pi@raspberrypi:~/Desktop/demo $ dmesg | tail -3
[  403.827133] [<801a2540>] (SyS_delete_module) from [<80108000>] (ret_fast_syscall+0x0/0x1c)
[  403.835402] Code: e24cb004 e24dd008 e52de004 e8bd4000 (e1d034b4)
[  403.841754] ---[ end trace cfc1ab2da6a69855 ]---
pi@raspberrypi:~/Desktop/demo $ dmesg | tail -10
[  403.762148] [<802eeab0>] (sysfs_remove_file_ns) from [<804fd830>] (driver_remove_file+0x28/0x2c)
[  403.770939] [<804fd830>] (driver_remove_file) from [<804fc664>] (bus_remove_driver+0x90/0xb0)
[  403.779468] [<804fc664>] (bus_remove_driver) from [<804fd9b4>] (driver_unregister+0x38/0x58)
[  403.787910] [<804fd9b4>] (driver_unregister) from [<804feb9c>] (platform_driver_unregister+0x1c/0x20)
[  403.797148] [<804feb9c>] (platform_driver_unregister) from [<7f2a6cf4>] (lirc_rpi_exit+0x3c/0x80 [lirc_new])
[  403.807004] [<7f2a6cf4>] (lirc_rpi_exit [lirc_new]) from [<7f2a6d68>] (lirc_rpi_exit_module+0x30/0x40 [lirc_new])
[  403.817290] [<7f2a6d68>] (lirc_rpi_exit_module [lirc_new]) from [<801a2540>] (SyS_delete_module+0x184/0x1d0)
[  403.827133] [<801a2540>] (SyS_delete_module) from [<80108000>] (ret_fast_syscall+0x0/0x1c)
[  403.835402] Code: e24cb004 e24dd008 e52de004 e8bd4000 (e1d034b4)
[  403.841754] ---[ end trace cfc1ab2da6a69855 ]---
pi@raspberrypi:~/Desktop/demo $
```

Figure 20: Problem with unloading using exit

the source now.I am fairly sure that messed up reference counts is the reason why this is happening.

As a result of this,the approach had to be modified.

3.5 Modified Solution

An easy solution was to avoid loading and unloading the module on uevent generation and instead just write to one of its attributes whose store function could then called the power_off() function.The rest of the approach remains unmodified.The source code of the project is hosted on github at `https://github.com/amateurcoder10/lirc_demo`

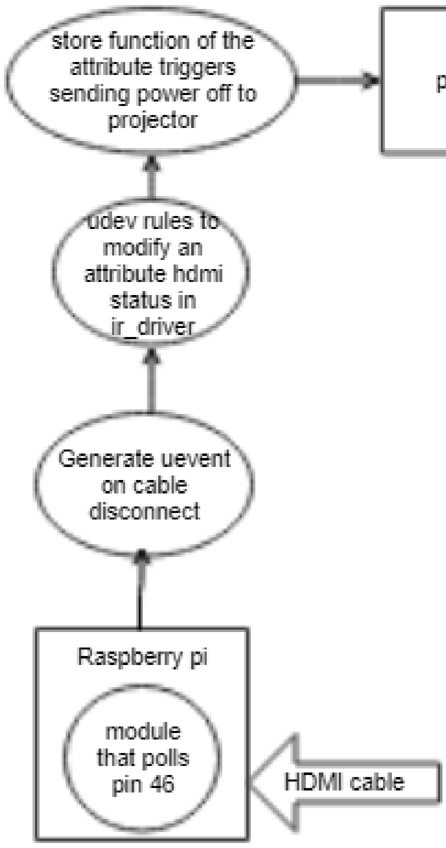

store function of the attribute triggers sending power off to projector
projector
udev rules to modify an attribute hdmi status in ir_driver
Generate uevent on cable disconnect
Raspberry pi
module that polls pin 46
HDMI cable

Figure 21: A flowchart depicting the solution

The details of the hardware connections are as shown below:pins 22 and 23 were used as the transmitter and receiver respectively.

This approach worked.

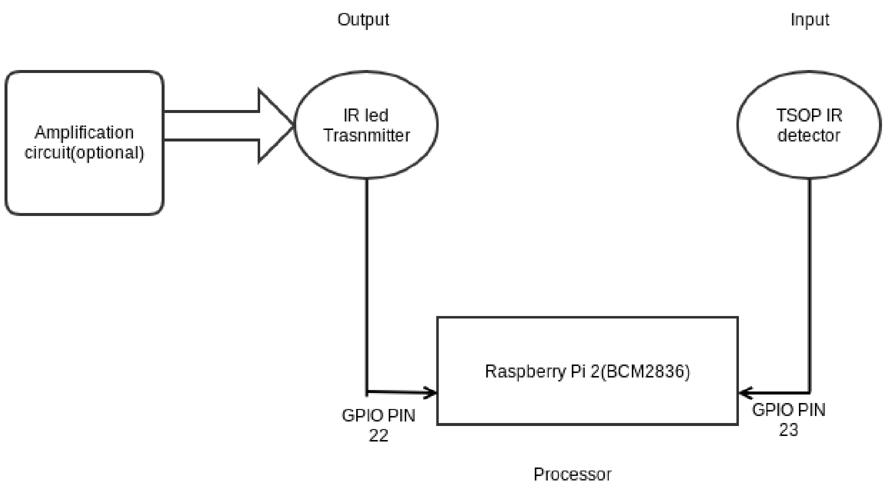
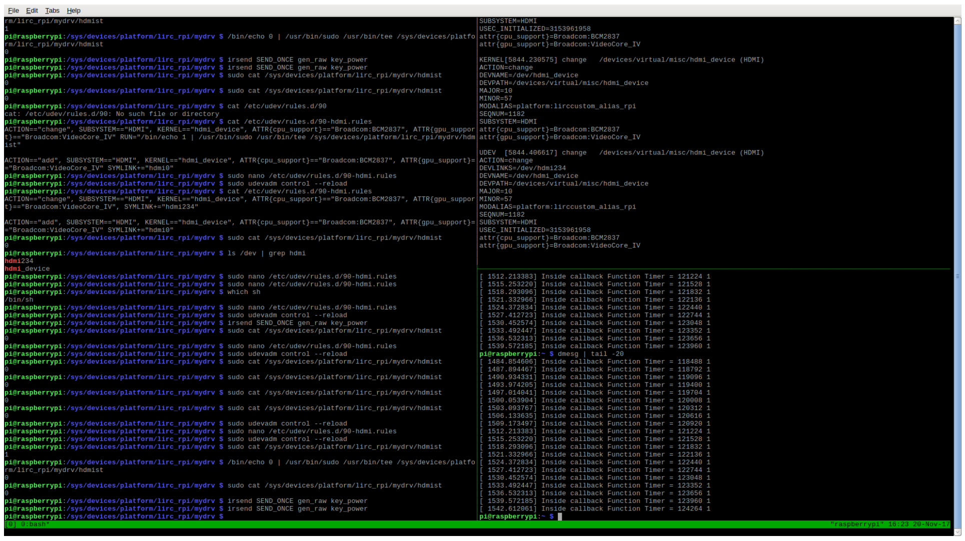


Figure 22: Block Diagram



```
rm/lirc_rpi/mydrv/hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ /bin/echo 0 | /usr/bin/sudo /usr/bin/tee /sys/devices/platfo
rm/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ irsend SEND_ONCE gen_raw key_power
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ irsend SEND_ONCE gen_raw key_power
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ cat /etc/udev/rules.d/90
cat: /etc/udev/rules.d/90: No such file or directory
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ cat /etc/udev/rules.d/90-hdmi.rules
ACTION=="change", SUBSYSTEM=="HDMI", KERNEL=="hdmi_device", ATTR{cpu_support}=="Broadcom:BCM2837", ATTR{gpu_suppor
t}=="Broadcom:VideoCore_IV" RUN="/bin/echo 1 | /usr/bin/sudo /usr/bin/tee /sys/devices/platform/lirc_rpi/mydrv/hdm
ist"

ACTION=="add", SUBSYSTEM=="HDMI", KERNEL=="hdmi_device", ATTR{cpu_support}=="Broadcom:BCM2837", ATTR{gpu_support}=
="Broadcom:VideoCore_IV" SYMLINK+="hdmi0"
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo nano /etc/udev/rules.d/90-hdmi.rules
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo udevadm control --reload
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ cat /etc/udev/rules.d/90-hdmi.rules
ACTION=="change", SUBSYSTEM=="HDMI", KERNEL=="hdmi_device", ATTR{cpu_support}=="Broadcom:BCM2837", ATTR{gpu_suppor
t}=="Broadcom:VideoCore_IV", SYMLINK+="hdmi234"

ACTION=="add", SUBSYSTEM=="HDMI", KERNEL=="hdmi_device", ATTR{cpu_support}=="Broadcom:BCM2837", ATTR{gpu_support}=
="Broadcom:VideoCore_IV" SYMLINK+="hdmi0"
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ ls /dev | grep hdmi
hdmi234
hdmi_device
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo nano /etc/udev/rules.d/90-hdmi.rules
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo nano /etc/udev/rules.d/90-hdmi.rules
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ which sh
/bin/sh
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo nano /etc/udev/rules.d/90-hdmi.rules
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo udevadm control --reload
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ irsend SEND_ONCE gen_raw key_power
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo nano /etc/udev/rules.d/90-hdmi.rules
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo udevadm control --reload
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo udevadm control --reload
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo nano /etc/udev/rules.d/90-hdmi.rules
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo udevadm control --reload
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
1
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ /bin/echo 0 | /usr/bin/sudo /usr/bin/tee /sys/devices/platfo
rm/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ sudo cat /sys/devices/platform/lirc_rpi/mydrv/hdmist
0
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ irsend SEND_ONCE gen_raw key_power
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $ irsend SEND_ONCE gen_raw key_power
pi@raspberrypi:/sys/devices/platform/lirc_rpi/mydrv $
[0] 0:bash*
```

```
SUBSYSTEM=HDMI
USEC_INITIALIZED=3153961958
attr{cpu_support}=Broadcom:BCM2837
attr{gpu_support}=Broadcom:VideoCore_IV

KERNEL[5844.230575] change   /devices/virtual/misc/hdmi_device (HDMI)
ACTION=change
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
MODALIAS=platform:lirccustom_alias_rpi
SEQNUM=1182
SUBSYSTEM=HDMI
attr{cpu_support}=Broadcom:BCM2837
attr{gpu_support}=Broadcom:VideoCore_IV

UDEV  [5844.406617] change   /devices/virtual/misc/hdmi_device (HDMI)
ACTION=change
DEVLINKS=/dev/hdmi234
DEVNAME=/dev/hdmi_device
DEVPATH=/devices/virtual/misc/hdmi_device
MAJOR=10
MINOR=57
MODALIAS=platform:lirccustom_alias_rpi
SEQNUM=1182
SUBSYSTEM=HDMI
USEC_INITIALIZED=3153961958
attr{cpu_support}=Broadcom:BCM2837
attr{gpu_support}=Broadcom:VideoCore_IV
```

```
[ 1512.213383] Inside callback Function Timer = 121224 1
[ 1515.253220] Inside callback Function Timer = 121528 1
[ 1518.293096] Inside callback Function Timer = 121832 1
[ 1521.332966] Inside callback Function Timer = 122136 1
[ 1524.372834] Inside callback Function Timer = 122440 1
[ 1527.412723] Inside callback Function Timer = 122744 1
[ 1530.452574] Inside callback Function Timer = 123048 1
[ 1533.492447] Inside callback Function Timer = 123352 1
[ 1536.532313] Inside callback Function Timer = 123656 1
[ 1539.572185] Inside callback Function Timer = 123960 1
pi@raspberrypi:~ $ dmesg | tail -20
[ 1484.854606] Inside callback Function Timer = 118488 1
[ 1487.894467] Inside callback Function Timer = 118792 1
[ 1490.934331] Inside callback Function Timer = 119096 1
[ 1493.974205] Inside callback Function Timer = 119400 1
[ 1497.014041] Inside callback Function Timer = 119704 1
[ 1500.053904] Inside callback Function Timer = 120008 1
[ 1503.093767] Inside callback Function Timer = 120312 1
[ 1506.133635] Inside callback Function Timer = 120616 1
[ 1509.173497] Inside callback Function Timer = 120920 1
[ 1512.213383] Inside callback Function Timer = 121224 1
[ 1515.253220] Inside callback Function Timer = 121528 1
[ 1518.293096] Inside callback Function Timer = 121832 1
[ 1521.332966] Inside callback Function Timer = 122136 1
[ 1524.372834] Inside callback Function Timer = 122440 1
[ 1527.412723] Inside callback Function Timer = 122744 1
[ 1530.452574] Inside callback Function Timer = 123048 1
[ 1533.492447] Inside callback Function Timer = 123352 1
[ 1536.532313] Inside callback Function Timer = 123656 1
[ 1539.572185] Inside callback Function Timer = 123960 1
[ 1542.612061] Inside callback Function Timer = 124264 1
pi@raspberrypi:~ $
                                        "raspberrypi" 16:23 20-Nov-17
```

Figure 23: The final output

4 Future work

There remains one curious glitch.The lirc_new module works as expected if preceded by an irsend first(can be observed in the output picture above).Else,mode2 and irw outputs are gibberish.It just needs the initial irsend.Thereafter,it works perfectly any number of times.

This is not a major glitch,but it is interesting entirely due to its uniqueness.This was a rather fortuitous catch and an entirely lucky observation–I had spent close to ten hours in despair trying to debug the gibberish mode2 was spewing out before I noticed this.Obviously,this means that irsend is doing something internally that the module isnt.There was a buffer involved that was clumsily statically allocated in the custom module and I was sure it was this.But even after properly allocating it,the bug still remains.I have gone insane trying to figure this out,but to no avail.My idea is to find out what irsend does in the beginning that my module doesnt.Im sure the reason lies there in some tiny line of code,but whether Ill get there remains to be seen.Reminds me of the etymology of the word bug!

Due to permission issues,this was only tested in room 204 with a casio projector.Field testing in R105 is yet to be done.

Detecting whether the connected device is a monitor or projector was also not possible because of time constraints.The same is to be said for an in-depth exploration of mailboxes.

5 Conclusion/Experience

This was a commendable exercise that involved attaining quite a few objectives along the way,which are in themselves significant.While it obviously improved my abilities to write device drivers,it brought about other things worth mentioning.The ability to read source code without documentation.Before this,had I been asked to go through a 1000 line long file such as lirc_rpi.c or tvservice.c and thereafter modify it,I would have blanched at the thought.Today,it seems doable–not trivial,yet possible.The realisation that writing kernel code is much more different from userspace programming and involves a lot more caution.The myth that drivers is a very stable and static area–with kernel releases in every month and changes in internal mechanisms,this is anything but true.In fact,the changes are so rapid that its hard to keep up sometimes.

6 Acknowledgements

This project would have been incomplete,if not impossible,without all of my teammates.One aspect that I really appreciate about this team is that no two members share the same skillset.They were all very diverse;therefore their individual contributions were significantly valuable and varied.It was a wonderful experience in terms of people management,group work,leadership and coordination.

However,I would be lying if I said the journey was always smooth.Many a time,I was reminded of lines from The Mythical Man Month.Nevertheless,these are problems that are inherent in the real world as well,and dealing with such vagaries will definitely come in use.

I would also like to acknowledge my other classmates,who shared their valuable insights and results of their own experiments with me.This was crucial in re-affirming or contradicting my own conclusions.

Last but not the least,the course material,by which I primarily refer to the examples like bhalu,blinker and digiout, was extremely relevant and helpful.They contained pieces of the puzzle,so to speak.

To conclude,it has been a great learning experience,on so many levels.

7 Footnotes

NB:All experiments were performed as a team.The observations and deductions are,however,my own.I have narrated in the first person throughout purely for the sake of consistency and this not imply that I have worked individually.

As a team leader,I did my best to emphasize the importance of properly documenting results and drawing conclusions from them.However,I am not sure if my words had the desired impact on all of my teammates.Writing,in itself,is abhorrent to some;and when coupled with detailed explanations,it comes across as too arduous to them.As the saying goes,one can only lead the horse to the water.I do not see what more I can do.

I did not follow the template of standard IEEE papers while typesetting because a two column format restricted the width of the screenshots and rendered them unreadable.

Although the assignment mentioned that the report should be written like a technical paper,I have gone into detail much more than I would if I were writing an actual paper.The reason is that I believe the journey to the solution–the challenges,the debugging and the fixes–are more important than the solution itself.If the report is too verbose,please accept my apologies.

References

[1] Dragos Mitrica, "Infrared light can be detected by the human eye after all," https://www.zmescience.com/science/physics/infrared-light-human-eye-detection-06455/, [Online; accessed Nov 2017].

[2] stackoverflow.com, "If infrared not visible, why the red leds?," https://electronics.stackexchange.com/questions/321780/if-infrared-not-visible-why-the-red-leds, [Online; accessed Nov 2017].

[3] wikipedia.com, "Consumer ir," https://en.wikipedia.org/wiki/Consumer_IR, [Online; accessed Nov 2017].

[4] sparkfun.com, "Ir communication," https://learn.sparkfun.com/tutorials/ir-communication, [Online; accessed Nov 2017].

[5] adafruit.com, "Ir remote signals," https://learn.adafruit.com/ir-sensor/ir-remote-signals, [Online; accessed Nov 2017].

[6] lirc.org, "Lirc 0.10.0rc1 manual," http://www.lirc.org/html/, [Online; accessed Nov 2017].

[7] Jonathan Corbet, "Kernel support for infrared receivers," https://lwn.net/Articles/364515/, [Online; accessed Nov 2017].

[8] Prasanth J, "Getting lirc to work with raspberry pi 3 (raspbian stretch)," https://gist.github.com/prasanthj/c15a5298eb682bde34961c322c95378b, [Online; accessed Nov 2017].

[9] stackexchange.com, "Tsop 38khz reciever vs. ir phototransistor," https://electronics.stackexchange.com/questions/213999/tsop-38khz-reciever-vs-ir-phototransistor, [Online; accessed Nov 2017].

[10] github.com, "source code-tvservice.c," https://github.com/raspberrypi/userland/blob/master/host_applications/linux/apps/tvservice/tvservice.c, [Online; accessed Nov 2017].

[11] github.com, "Mailbox property interface," https://github.com/raspberrypi/firmware/wiki/Mailbox-property-interface, [Online; accessed Nov 2017].

[12] Mahesh Sreekandath, "Raspberry pi 2 arm-gpu ipc," https://msreekan.com/2016/09/13/raspberry-pi2-arm-gpu-ipc/, [Online; accessed Nov 2017].

[13] Dieter Oelofse, “Raspberry pi frame buffer,” `http://magicsmoke.co.za/?p=284`, [Online; accessed Nov 2017].

[14] github.com, “Source code–mailbox,” `https://github.com/raspberrypi/linux/blob/rpi-4.9.y/drivers/mailbox/bcm2835-mailbox.c`, [Online; accessed Nov 2017].

[15] github.com, “source code-vcmailbox.c,” `https://github.com/raspberrypi/userland/blob/master/host_applications/linux/apps/vcmailbox/vcmailbox.c`, [Online; accessed Nov 2017].

[16] raspberryi.org, “Industrial use of the raspberry pi,” `https://www.raspberrypi.org/documentation/hardware/industrial/`, [Online; accessed Nov 2017].

[17] wikipedia.org, “Extended display identification data,” `https://en.wikipedia.org/wiki/Extended_Display_Identification_Data`, [Online; accessed Nov 2017].

[18] man pages, “man(1) xxd,” `https://linux.die.net/man/1/xxd`, [Online; accessed Nov 2017].

[19] man pages, “Man(8) udev,” `https://linux.die.net/man/8/udev`, [Online; accessed Nov 2017].

[20] doc.opensuse.org, “Dynamic kernel device management with udev,” `https://doc.opensuse.org/documentation/leap/reference/html/book.opensuse.reference/cha.udev.html`, [Online; accessed Nov 2017].

[21] Daniel Drake, “Writing udev rules,” `http://reactivated.net/writing_udev_rules.html`, [Online; accessed Nov 2017].

[22] man pages, “Man udevadm,” `https://www.freedesktop.org/software/systemd/man/udevadm.html`, [Online; accessed Nov 2017].

[23] elixir.free electrons.com, “Cross reference to kobject uevent,” `https://elixir.free-electrons.com/linux/v3.2/source/lib/kobject\_uevent.c\#L328`, [Online; accessed Nov 2017].

[24] Derek Molly, “Writing a linux kernel module part 3: Buttons and leds,” `http://derekmolloy.ie/kernel-gpio-programming-buttons-and-leds/`, [Online; accessed Nov 2017].

[25] Broadcom, “Bcm2836 arm-local peripherals,” `https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2836/QA7_rev3.4.pdf`, [Online; accessed Nov 2017].

[26] ARM, “Arm cortex a7 technical reference manual,” `http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0464f/index.html`, [Online; accessed Nov 2017].

[27] github.com, “source code–lirc_rpi.c,” `https://github.com/bengtmartensson/lirc_rpi/blob/master/lirc_rpi.c`, [Online; accessed Nov 2017].

[28] wiki.archlinux.org, “Modalias,” `https://wiki.archlinux.org/index.php/Modalias`, [Online; accessed Nov 2017].