Controlling a projector using Raspberry Pi

Submitted by:Prashanthi S.K(MT2016520)

Abstract

This is a technical report 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 plugged/unplugged to the pi and thereafter turn on/off the projector as is the case. [To be modified to include implementation details,tentative abstract]

Introduction

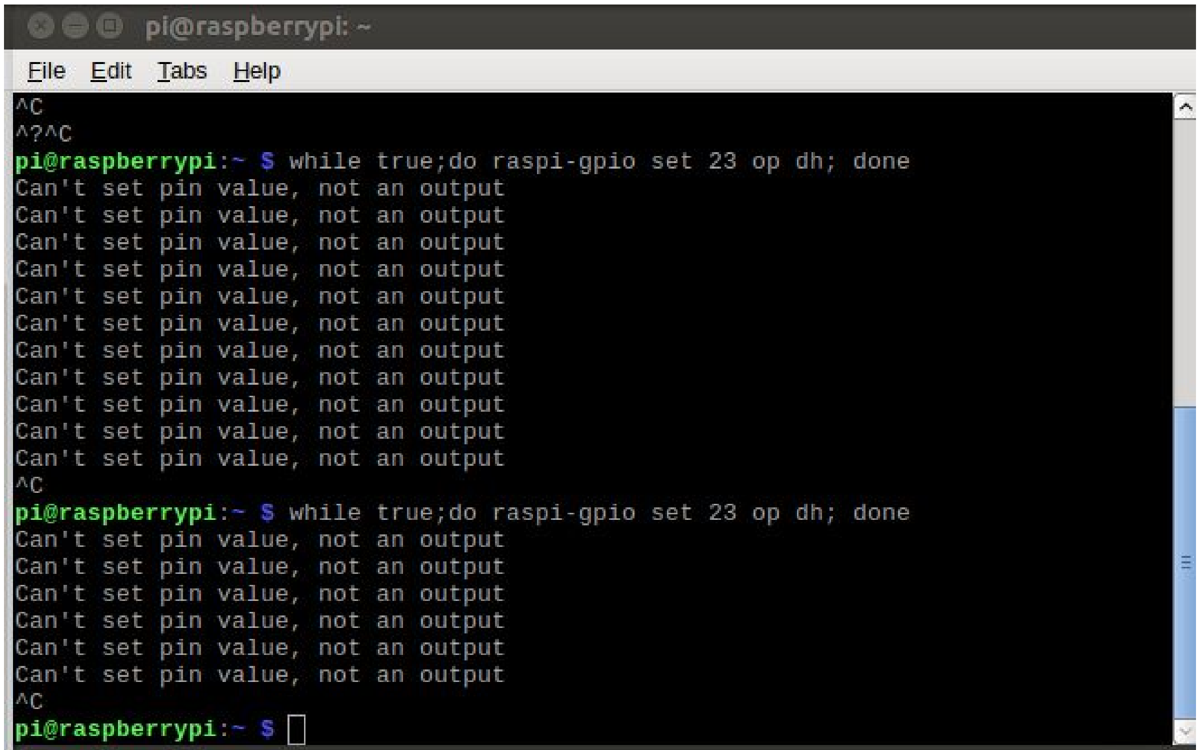
[to be written]

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.

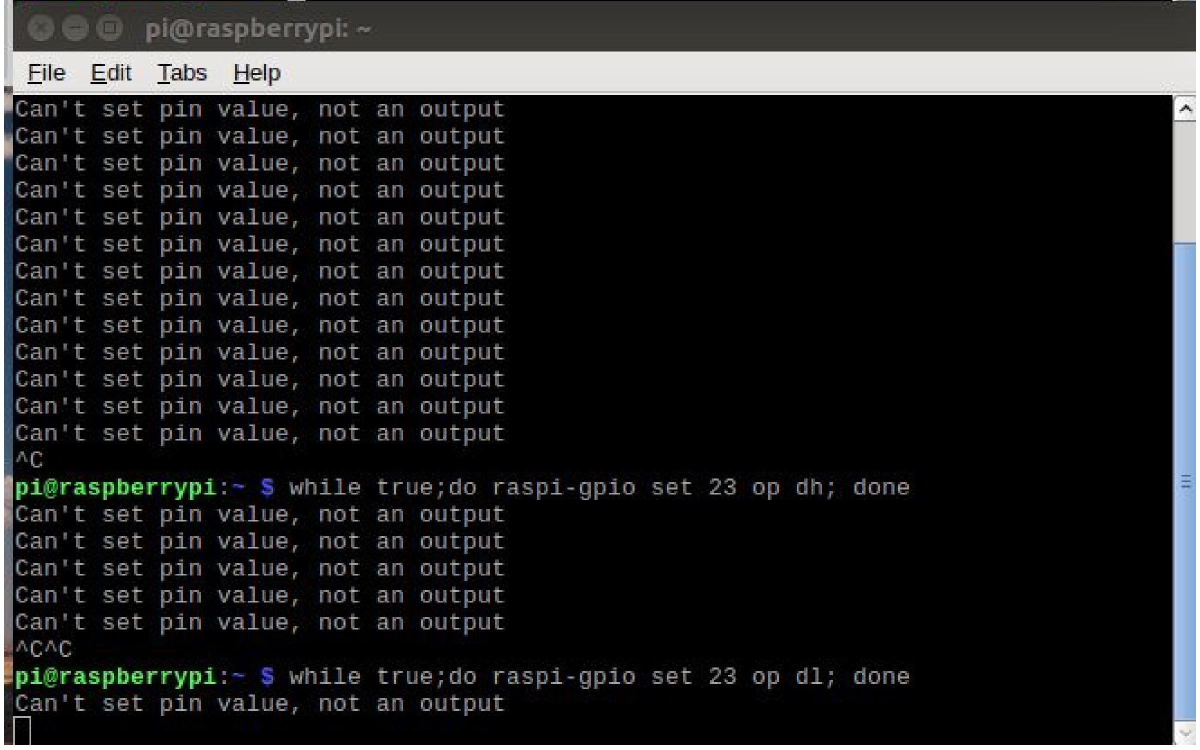
GPIO

I first started by reading up on lirc[1][3].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 wasn't what I'd 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.



File Edit Tabs Help

```
^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:~ $
```

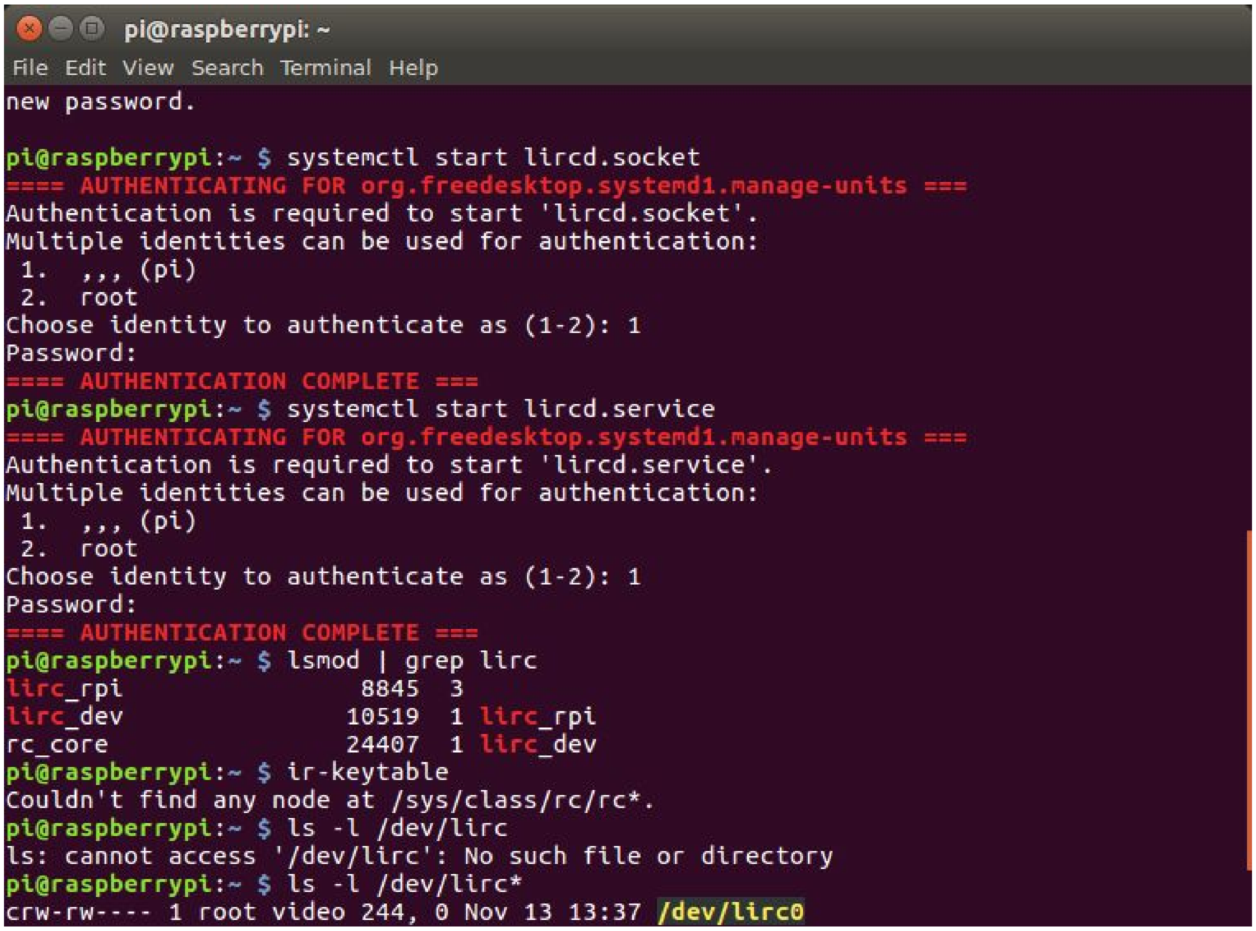


File Edit Tabs Help

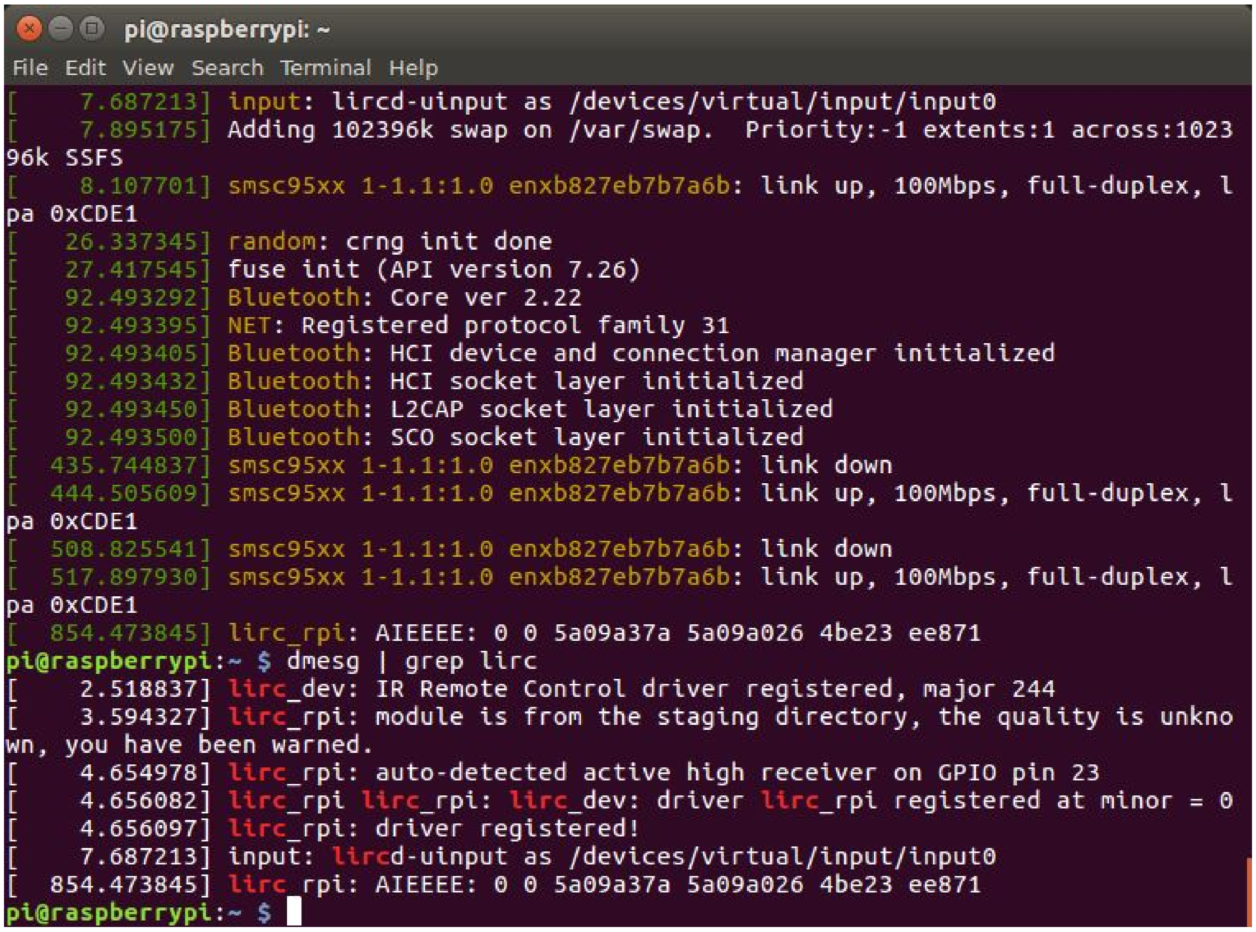
```
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, not an output
Can't set pin value, 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
```

LIRC

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[1][2].



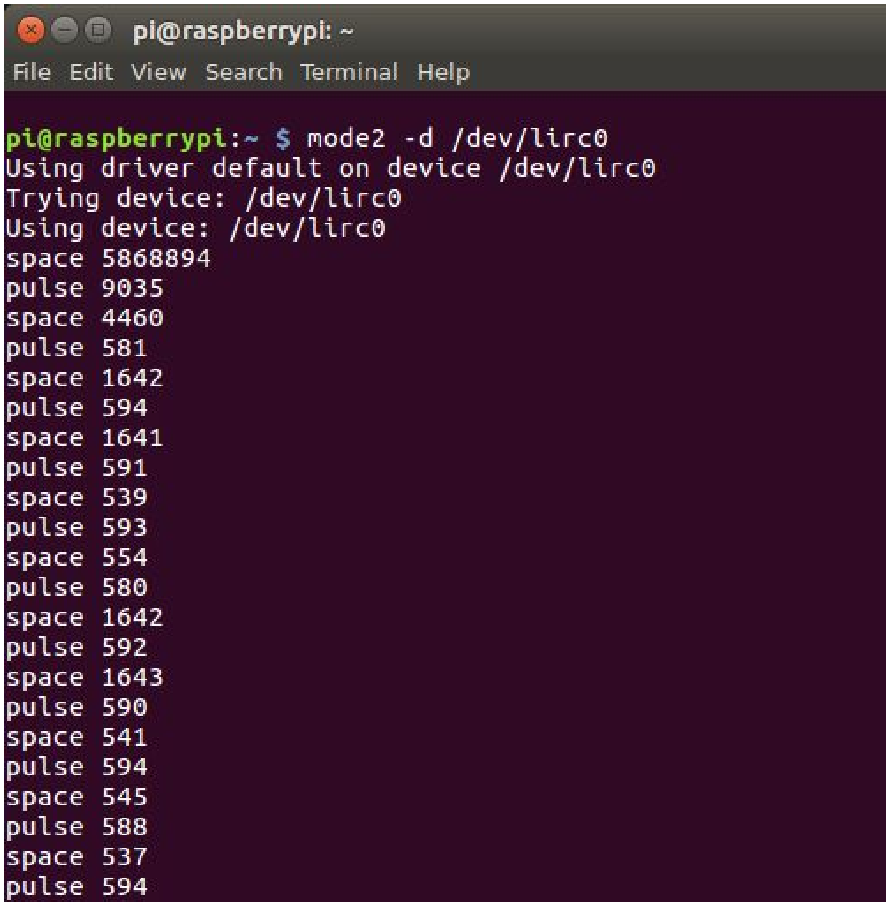
```
new password.

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
```



```
[    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:~ $
```

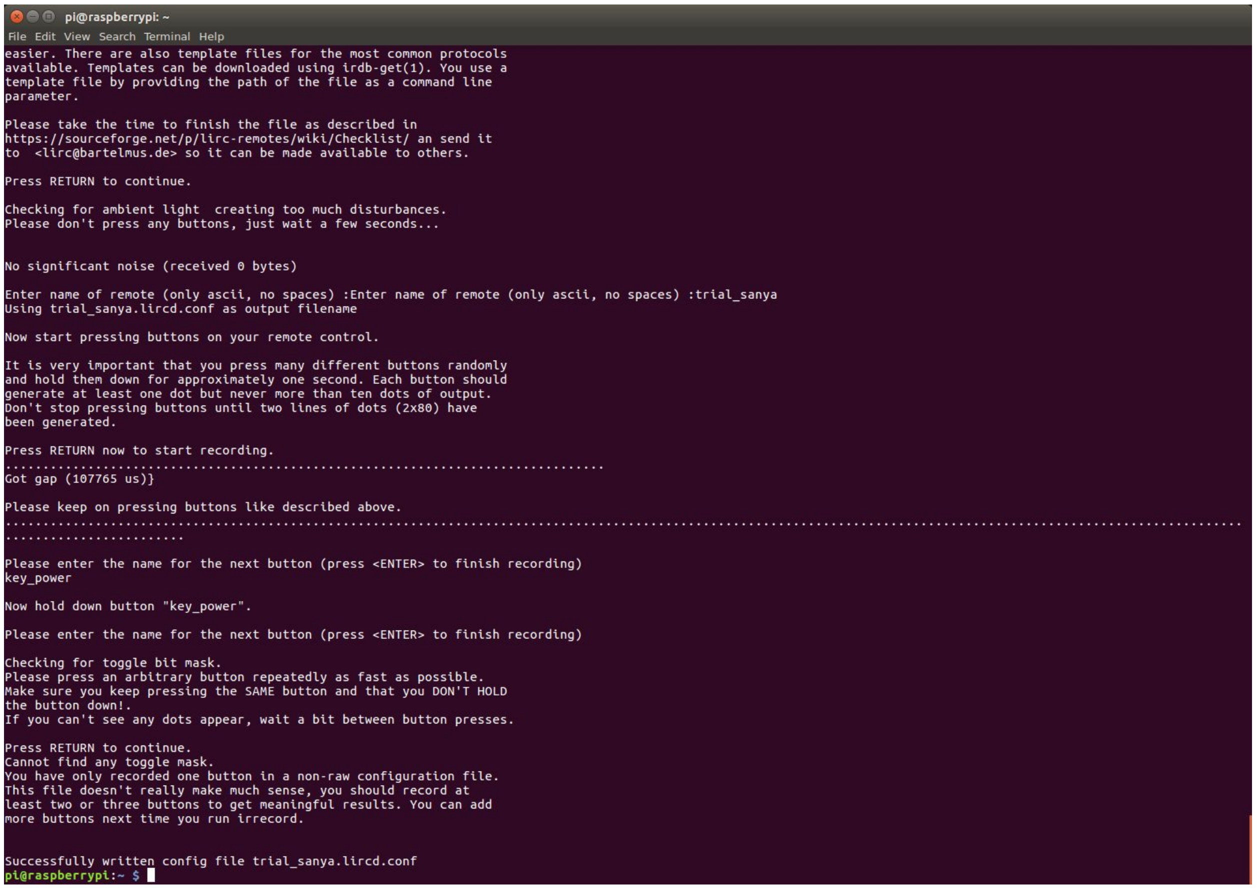
The next logical step would be to try and receive pulses from an IR remote.I connected a photodiode to the Pi and used mode2 for this purpose.It didn't work as expected and I got something called a partial read.To debug this,I connected it a CRO so that I could see the signal at a hardware level itself.But 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] I'm 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 didn't 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 don't 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).



```
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
```

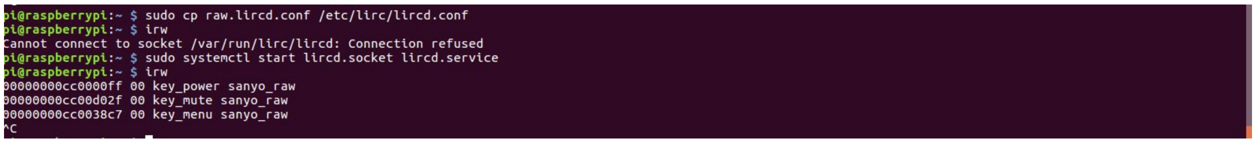
The next step was to test a loop--generate a configuration file for the remote using irrecord,transmit the codes using irsend and receive them back using irw.I did the irrecord part this without facing any difficulty.Irsend also worked but to my chagrin irw did not receive anything.I re-did the whole 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.



```
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:~ $
```

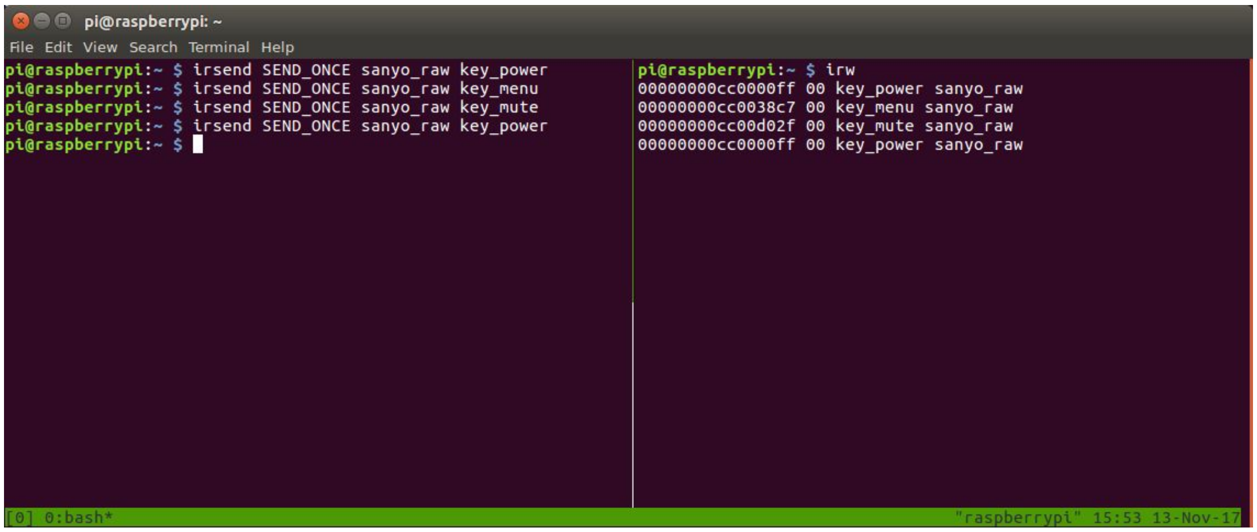
I searched for this problem but couldn’t find a solution online.Intuitively,this seemed to be a recording malfunction and this is what seemed the logical reason as to why.If the program made a slight mistake in recognising the value for a one and a zero in terms of spaces and pulses,the entire code generated 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.

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 wasn’t 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.



```
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
```

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.



```
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
```

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.I haven’t debugged this issue fully yet.

I then went on to read lirc_rpi.c so that I could modify this to come up with my own driver.But there were two components to this project and it looked like I was spending far too much time on one,so I decided to move on to the other part i.e hdmi.(Yesterday’s mail reinforced this conclusion.In fact,I felt guilty about spending too much time on lirc.However,as the saying goes,the devil is in the details--attention to detail is something I consider important.Time constraints,however,bring an implicit choice between depth and breadth.Unless I move on to hdmi,there won’t be enough time to explore it in detail.)

HDMI

I started out with the tvservice utility.I used the -s flag and observed the change in outputs when the hdmi cable was connected and disconnected. 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.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.[10-13] These two exercises took up a lot of time,especially tvservice.c.I found a callback function tvservice_callback() which was the one actually reporting the hdmi states.Obviously,this function had to be traced down.I used a grep -ri to look for which files contained particular functions and terms but am yet to trace it fully.

Challenges faced

I am more used to looking at one or two code files at a time and this set of seemingly circular dependencies are overwhelming and making it hard for me to get to the right point.For some files,the documentation available is also meagre and hard to comprehend,which leaves no choice but to read the source code.

Another point this brought up is that without a proper outline to dictate the flow of work,more time is spent figuring out what is to be done next rather than completing a sub-task.This situation is more pertinent to the real world,where there is no solution already outlined and just a problem statement.

Team-work is another aspect this required.Initially,as a team leader,I wrestled with the dilemma of whether all of us should work parallely on similar sub-tasks and share our results or take up different sub-problems and focus on that alone.While the latter implies more productivity in a lesser time,it does not give the individual people an idea of the bigger picture and I lose out on valuable insights of others just due to lack of knowledge.Therefore,I went with the former approach,which is rather slow,yet promising;because it gives me a chance to look at different ways to get to a solution.

I realise that there is quite a lot of work left over for the coming week,I will put in extra effort and time to complete the outlined objectives.Having worked with my teammates for a while now,I have figured out what they each are good at.Therefore,I will utilise each of their skills to the fullest to come up with a solution.

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.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[7].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.[8]

To be done

Thorough code review of tvservice--trace the calls down
Detect a hdmi plug/unplug event and use a uevent to trigger sending the power on/off
Use the IEEE format for references
IEEE format for websites:Author Initial. Author Surname, 'Title', Year Published. [Online]. Available: http://Website URL. [Accessed: 10- Oct- 2013].

Conclusion

[To be filled in]

References

To be formatted

1)lirc.org,'LIRC 0.10.0rc1 Manual',1999[Online],Available:http://www.lirc.org/html/

2)Prashanth J,'Getting lirc to work with Raspberry Pi 3 (Raspbian Stretch)',Oct 2017--https://gist.github.com/prasanthj/c15a5298eb682bde34961c322c95378b

3)Kernel support for infrared receivers--https://lwn.net/Articles/364515/

4)https://learn.sparkfun.com/tutorials/ir-communication

5)https://electronics.stackexchange.com/questions/321780/if-infrared-not-visible-why-the-red-leds

6)https://www.zmescience.com/science/physics/infrared-light-human-eye-detection-06455/

7)https://en.wikipedia.org/wiki/Consumer_IR

8)https://learn.adafruit.com/ir-sensor/ir-remote-signals

9)https://electronics.stackexchange.com/questions/213999/tsop-38khz-reciever-vs-ir-phototransistor

10)https://github.com/raspberrypi/userland/blob/master/host_applications/linux/apps/tvservice/tvservice.c

11)https://github.com/raspberrypi/firmware/wiki/Mailbox-property-interface

12)https://msreekan.com/2016/09/13/raspberry-pi2-arm-gpu-ipc/

13)http://magicsmoke.co.za/?p=284

NB:Most of the experiments were performed as a group.The observations and deductions are,however,my own.I have narrated in the first person throughout purely for the sake of consistency.Also,I'm still in the process of learning typesetting,and I wanted to get all the content into the draft report before I began formatting,therefore this draft is not typeset.

I,like most of my other classmates,mistook draft report to mean something like a midphase report detailing progress.Therefore,I went about preparing a proper report containing whatever I have done so far. The mail stated otherwise;but I already written this by then and I don't see the point of going from a proper report to an outline backwards,so I am submitting this.My report is more rambling than concise;my apologies for the same--it was my misunderstanding.I will write the final one like a proper technical paper.