IR PROTOCOLS AND GENERATION OF A/C REMOTE CONFIGURATION FILE

In electronics, a **remote control** is a component of an electronic device used to operate the device wirelessly from a distance. For example, in consumer electronics, a remote control can be used to operate devices such as a television set, DVD player, or other home appliance, from a short distance.

Technology:

The main technology used in home remote controls is infrared (IR) light. The signal between a remote control handset and the device it controls consists of pulses of infrared light, which is invisible to the human eye, but can be seen through a **digital camera,** video camera or a phone camera. The transmitter in the remote control handset sends out a stream of pulses of infrared light when the user presses a button on the handset. A transmitter is often a light emitting diode (LED) which is built into the pointing end of the remote control handset. The infrared light pulses form a pattern unique to that button. The receiver in the device recognizes the pattern and causes the device to respond accordingly.

Today, IR remote controls almost always use a pulse width modulated code, encoded and decoded by digital computer: a command from a remote control consists of a short train of pulses of carrier-present and carrier-not-present of varying widths.

Consumer electronics infrared protocols

Different manufacturers of infrared remote controls use different protocols to transmit the infrared commands. The **RC-5** protocol that has its origins within **Philips**, uses, for instance, a total of **14 bits** for each button press. The bit pattern is modulated onto a carrier frequency that, again, can be different for different manufacturers and standards, in the case of RC-5, the carrier is 36 kHz. Other consumer infrared protocols include the various versions of SIRCS used by Sony, the RC-6 from Philips, the Ruwido R-Step, and the NEC TC101 protocol.Some Electronic manufactures like Samsung ,Lg ,Daikin use their own protocols where each IR command data length varies from **16bits to 52bits.**

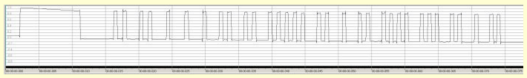
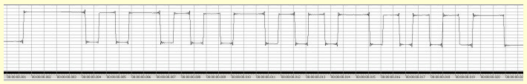
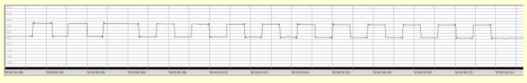
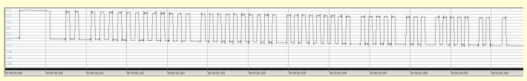
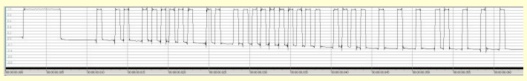
Since the Consumer IR protocols are for the most part not standardised, computers and **universal remotes** often memorise a bit stream, possibly with compression and possibly without determining the actual bit rate, and play it back. Similarities between remotes are often largely the accidental result of the finite selection of infrared encoder/decoder chips (though now microcontrollers are also used) and IR receiver modules or imitation of the older chips rather than by design. Manufacturers of consumer appliances often reuse the same protocol on many similar devices, though for each manufacturer and device type there are usually multiple protocols in use. The code listings inform about for any universal remote.

Sony use the **SIRC** protocol for remote controls. SIRC is developed in three different versions: 12 bit, 15 bit and 20 bit. After 12 bits have been received, the receiver waits to see if there are more falling edges to know if the SIRC protocol is **15 bit or 20 bit** coded.

The **RECS-80** and **RC-5** codes developed by **Philips** have been casually referred to as international standards. However, the RECS-80 protocol was prone to interference and was quickly replaced by the RC-5 protocol. Although it appears that they were proprietary protocols developed by Philips, they were also adopted by various other manufacturers, specifically European- and US-based ones. This allowed interoperability between the remote handsets and equipment of various brands. The RC-5 code was, and still is, used by many US- and European-based manufacturers of specialty audio/video equipment. Unfortunately, documentation of the standard commands were not widely distributed. Therefore, there are some brands of equipment that use non-standard commands, causing interference with other equipment also using the RC-5 protocol.

The rapidly expanding requirements for newer categories of electronics products since that time (e.g., DVD players, cable boxes, DVR's, et cetera) has led Philips to replace the RC-5 protocol with the newer RC-6 protocol that has both an expanded set of devices (256 versus 32) and commands per device (256 versus 64 in RC-5 and 128 in RC-5x). Again, information on the RC-6 protocol is not readily available from Philips.

In contrast, the major Japanese consumer electronics manufacturers almost universally adopted a protocol that was developed and administered by NEC. In the NEC protocol, each manufacturer is assigned a unique code that is contained in the transmitted command, avoiding the possibility of false triggering by other remote handsets.

# - protocol	Total waveform samples:
2-NEC	
4-SIRCS	
5-RC5	
7-JAPAN	
8-SAMSUNG	

Waveforms of IR Signals

Daikin Protocol

The Daikin protocol, is split into three 'messages' these messages are split by a 33-34 millisecond gap. The first message, in the case of my air conditioner) doesn't change and is 8 bytes long. The second message holds the current time and is 8 bytes long. Finally the third message holds everything else and is 19 bytes long.

RECS-80 uses **pulse position modulation** and RC-5 uses **bi-phase**. Early dedicated-purpose chips were offered by Philips Semiconductors to allow for the easy use of RECS-80 and RC-5 protocols. The SAA3004, SAA3007, and SAA3008 encoder chips used RECS-80, and the SAA3006 and SAA3010 encoder chips used RC-5. The SAA3049A decoder chip decoded either type. (Note that the Philips Semiconductors division is now NXP). All of these chips have been discontinued. However, these transmission protocols are easily created and/or decoded with general-purpose 8-bit micro controllers, such as those offered by Microchip Technology and Atmel.

Transmission of the IR commands requires only a micro controller and an infrared LED, available from a wide variety of sources. Reception of the modulated commands for RC-5, RC-6, and the NEC protocols is easily accomplished with specialised IR receivers, most readily available from Sharp Corporation and Vishay Inter technology. These receivers include a photo-diode, an automatic gain control (AGC) circuit, and a demodulator. The demodulated signal is then decoded with a micro controller

GENERATION REMOTE IR REMOTE CONFIGURATION FILE USING LIRC

LIRC is a mature and stable open source library that provides the ability to send and receive IR commands.

Setting up LIRC on the RaspberryPi

Entire process is followed as Explained by alexa's blog

The method explained in above blog works well for TV,DVD ,Music Systems but they fail when used for Air Conditioning Systems.

This document addresses this problem on how to use LIRC for controlling AC units as per the steps below.

One thing to know is that usual remotes controls, for TV, HI-FI, ... send a signal for each key pressed (often in loops while the key is pressed). An air conditioning remote often displays information about the parameters selected. But of course parameters can be changed on the remote while the unit is out of reach, which could lead to synchronisation problems between the display and the unit in some cases if it worked like the TV remotes. This implies that when such a remote sends a signal, it sends the whole parameters set.

The A/C remote shows multiple options to transmit:

- The target temperature to achieve

- The "mode" ("cool","heat","dry", "auto")

- The air flow swing (5 possible positions, and one automatic mode)

- The fan speed

- A powerful or quiet option (which impacts the fan speed (and more?))

- 2 timers: one to turn off the unit after a certain amount of time, one to bring it back on

Procedure to generate Remote configuration File for A/C

a) First step is to test that the system is able to receive the IR remote data.

b) Second step is to to generate a configuration file. Before generating a configuration file one need to verify if there already exists a configuration file in which case no need to generate.

c) Third step is to use this config file to control the AC.

Setting up Lirc

Lirc stands for “Linux Infrared Remote Control”. It is a package that will allow us to decode and to save IR signals for later use. Three of the tools it contains will be most useful to us :

Mode2:

outputs the pulse/space length of infrared signals to the console

irrecord :

used in order to record IR signals for later usage with lirc.

Ir send :

send IR signals from the command line. Signals can be sent once or repeatedly during two calls.

Install lirc steps

Before Installing LIRC we need to update and upgrade the Raspbian OS

sudo apt-get update

sudo apt-get upgrade

- `apt-get update` updates the list of available packages and their versions, but it does not install or upgrade any packages.
- `apt-get upgrade` actually installs newer versions of the packages you have. After updating the lists, the package manager knows about available updates for the software you have installed.

Install LIRC

sudo apt-get install lirc

Next We have to manually configure the driver, setting the GPIO pins which we use to receive and transmit the signals.

Edit /etc/modules

sudo nano /etc/modules

and add at the end of the file

lirc_dev

lirc_rpi gpio_in_pin=23 gpio_out_pin=22

Add this at the end of the file holding your system configuration parameters /boot/config.txt: .

dtoverlay=lirc-rpi,gpio_in_pin=23,gpio_out_pin=22

Next edit lirc hardware's configuration file /etc/lirc/hardware.conf

/etc/lirc/hardware.conf
 # Arguments which will be used when launching lircd LIRCD_ARGS="--uinput" # Don't start lircmd even if there seems to be a good config file # START_LIRCMD=false

 # Don't start irexec, even if a good config file seems to exist. # START_IREXEC=false

 # Try to load appropriate kernel modules LOAD_MODULES=true # Run "lircd --driver=help" for a list of supported drivers. DRIVER="default"
 # usually /dev/lirc0 is the correct setting for systems using udev DEVICE="/dev/lirc0" MODULES="lirc_rpi" # Default configuration files for your hardware if any LIRCD_CONF=""
LIRCMD_CONF="

Reboot your Rasbperry Pi

Registering new IR signals

We are going to make sure that our circuit and lirc are working properly by using mode2 to output the characteristics of an IR signal. Start by stopping lirc and launching mode2.

sudo /etc/init.d/lirc stop

mode2 -d /dev/lirc0

pi is now listening to IR inputs. Point a remote to the receiver and press any button. The program should output something similar to this, indicating that the reception side of things is all set.

space 4734
pulse 274
space 884
pulse 403
space 532
pulse 635
space 649
pulse 440

Procedure for Generating Configuration file for a remote using Samsung as test case

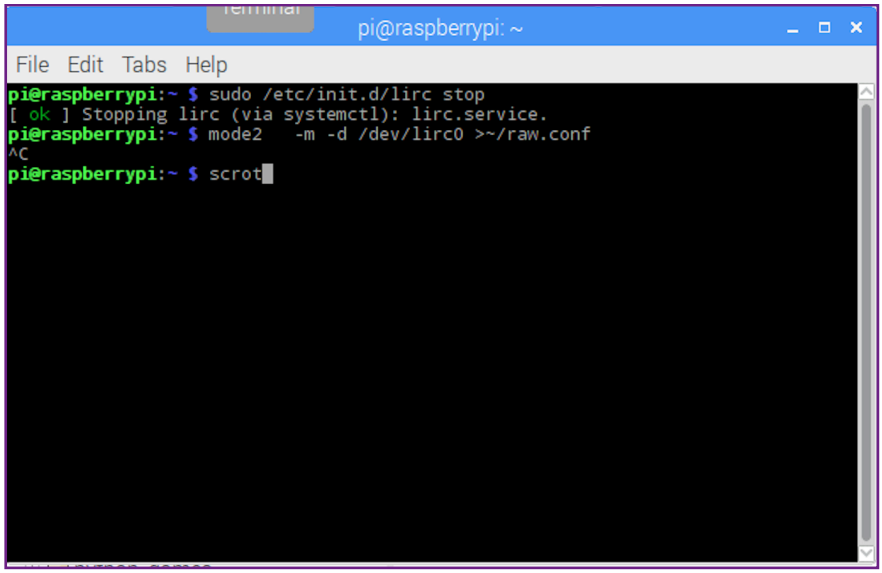
Generating a configuration File for Samsung A/C

To generate a remote configuration file, first stop lirc so that /dev/lirc0 is available for irrecord to use

sudo /etc/init.d/lirc/stop

To register a new remote control using mode2 tool,press any button facing the IR Receiver, the raw data from the the remote will be saved in decimal format in raw.conf file.Launch the command below and press any button in remote and stop recording by pressing CTRL + C .

mode2 -m —d /dev/lirc0 >> ~/raw.conf
CTRL^C



The raw.conf file generated for power on is

```
1667402

  4380     4389      469     1702      470      623
   469     1712      464     1703      469      595
   496      596      496     1676      496      597
   518      576      495     1686      491      596
   499      592      496     1675      497     1676
   496      597      494     1683      518     1661
   495      596      495     1680      496     1678
   494     1677      495     1676      496     1676
   496     1677      520      575      496     1676
   495      596      496      595      496      597
   495      596      495      596      496      598
   517      576      496      596      495      596
   497     1676      496      595      498      593
   496      596      496      598      517     1661
   495     1677      495     1677      495      595
   497     1658      517     1678      495     1677
   494     1677      549     5215     4410     4353
   495     1681      491      596      495     1682
   495     1676      496      596      495      596
   496     1676      498      596      518      575
   496     1680      496      595      497      595
   496     1677      495     1677      495      596
   496     1682      518     1660      496      596
   495     1681      495     1676      496     1676
   496     1676      497     1675      496     1678
   498      596      497     1675      495      595
   497      595      496      595      496      596
   496      595      496      602      514      575
   496      596      496      596      495     1677
   495      595      497      595      496      596
   495      598      520     1659      495     1702
   470     1676      496      594      497     1707
   473     1698      470     1702      479     1697
   469
```

In order to get the data into to correct format, remove the first row in the raw.conf file you just created, which is just the 'space' between when the recording started and you started to press the button.

After this insert the above raw.conf file into Original lircd file template which is as shown below

```
# Please make this file available to others
# by sending it to <lirc@bartelmus.de>
# this config file was automatically generated
# using lirc-0.9.0-pre1(default) on Tue Dec  3 09:00:19 2013
# contributed by
# brand:                    /home/pi/lircd.conf
# model no. of remote control:
# devices being controlled by this remote:

begin remote

  name  REMOTE_NAME
  flags RAW_CODES
  eps            30
  aeps           100


      begin raw_codes
          name KEY_POWER
              <<copy decimal values from raw.conf file to here>>
          name KEY_1
              <<copy decimal values from raw.conf file to here>>

      end raw_codes
end remote
```

Follow the above steps to record any number of keys.

The Final configuration file is

```
# Please make this file available to others

# by sending it to <lirc@bartelmus.de>

# this config file was automatically generated

# using lirc-0.9.0-pre1(default) on Thu Jun  8 11:27:14 2017

# contributed by

# brand:                    /home/pi/lircd.conf

# model no. of remote control:

# devices being controlled by this remote:
```

```
#
begin remote
  name  SAMSUNG
  bits          58
  flags RAW_CODES
  eps           30
  aeps          100
  gap           8970
      begin raw_codes
          name KEY_POWER
                    569   17895   3029    8954    481   517   485   1507    515   482   513   483
  517   480   513   483   513   483   513   483   488   508   514   1480  505
  492   512   488   509   1509  452   1537  457   539   458   1535  485   1507
  514   1479  491   1501  499   1494  491   509   485   508   495   502   487
  509   513   487   510   509   491   505   456   541   455   547   451   540
  481   515   484   513   511   483   516   482   513   483   513   482   493
  505   514   482   513   484   513   482   514   483   488   510   487   512
  509   510   486   510   456   541   460   536   457   538   482   516   482
  514   488   508   513   483   514   483   514   482   487   1506  517   1475
  514   2986  2993  8964  487   1507  511   514   484   508   456   541   456
  540   457   539   480   517   484   512   486   510   514   1479  522   479
  508   483   513   1480  513   484   513   1480  486   1508  511   1511  482
  1506  457   1535  481   1512  485   511   514   482   514   483   514   487
  509   483   512   484   513   484   511   484   513   484   512   485   512
  485   511   488   509   509   487   514   453   539   457   541   456   540
  482   514   485   511   513   484   512   484   512   484   514   483   513
  487   510   482   513   483   495   503   512   484   513   485   511   488
  509   509   487   509   456   541   456   540   457   539   483   513   513
  2965  3015  8964  513   1479  513   484   513   483   513   483   514   483
  513   484   503   493   513   484   512   485   511   1506  456   545   451
  541   456   540   478   519   483   513   512   484   513   480   514   1479
  512   1484  510   1479  513   483   512   1483  510   486   512   1505  456
  1541  452   540   458   539   481   515   512   1480  513   1480  514   1480
  512   482   514   483   513   484   513   484   512   484   513   1482  511
  509   486   510   457   1540  452   1537  482   514   511   1482  513   1479
  513   484   513   487   509   483   512   485   512   484   513   483   513
  484   513   486   510   510   487   510   456   1541  452   1536  512
          name KEY_1
                          564   17957   3006    8965    519   505   461   1530    487   510   485   511   486   511   485   510   461   564   457   512   515   508   434   1563    428   563   457   540
  458   1534    486   511   485   485   486   1507    486   1534    483   1507    486   1507    486   1509    472   549   489   507   432   564   440   557   433   564   457   513   484   537   461   536   485
  511   486   511   485   510   486   512   483   512   491   506   486   509   462   535   484   510   488   538   460   535   489   508   433   563   434   537   459   563   463   534   459   537   486   511
  483   486   487   536   460   535   487   511   484   512   485   511   485   1512    481   1508    485   1508    466   1552    486   2992    2986    8970    478   1537    458   537   485   508   464   535   486
  510   487   511   485   511   486   510   485   511   486   1506    462   534   461   516   488   556   489   508   432   1560    433   1558    459   538   484   1483    486   1530    487   1515    477   513
  467   1524    486   510   487   1506    462   1569    447   539   456   553   418   566   438   1554    456   1535    486   1482    486   536   485   512   485   510   485   511   492   506   485   1507    485
  511   485   1506    487   1534    454   1537    440   1553    457   540   457   1509    487   1533    460   536   485   486   485   536   490   506   485   511   487   510   486   509   487   1506    487   1533
  491   1476    464   1529    485
          name KEY_2
                          539   17975   2998    9010    459   544   452   1535    459   538   486   510   486   510   486   510   486   510   487   509   487   509   487   1512    455   536   461   534   461   1559
  459   536   434   563   434   1559    457   1536    459   1533    486   1505    487   1506    487   509   488   509   486   515   481   510   462   535   460   535   461   563   458   538   459   537   433   564
  433   563   434   563   457   543   454   537   486   512   485   510   487   510   485   511   486   510   486   510   486   509   462   535   461   541   455   536   486   511   485   538   458   538   433
  563   457   538   458   539   458   539   459   537   486   1511    482   1506    487   1504    488   1506    486   3018    2960    8989    486   1507    487   537   458   538   492   505   431   565   432   564
  433   563   457   539   459   537   486   1507    486   510   485   511   493   1499    487   509   461   1532    460   1532    460   564   437   1555    456   1536    458   1541    452   538   458   1534    487
  510   486   1506    486   1507    486   509   486   511   486   510   460   1533    460   1559    458   1535    432   563   456   540   458   546   451   537   460   512   486   536   485   511   486   1506
  486   1507    487   1510    481   1506    474   521   484   1537    459   1533    433   564   434   562   457   546   451   539   458   538   485   511   486   510   486   1506    487   1506    487   1510    455
  1532    461
          name KEY_3
                          543   17981   2992    8951    484   536   484   1507    485   512   487   508   487   510   485   491   505   539   432   563   493   509   429   1559    457   539   459   537
  460   1531    486   511   460   510   487   1543    476   1507    493   1499    485   1505    462   1562    432   535   492   519   476   539   429   562   458   538   457   538   460   536   460   536   464
  536   459   534   485   484   488   539   458   534   469   527   486   511   487   507   494   506   487   544   434   557   493   505   455   513   456   569   452   539   457   539   461   535   486   511
  459   536   487   483   488   538   463   530   486   510   486   514   483   1505    487   1488    505   1505    488   1507    493   3008    2989    8969    480   1507    488   535   482   515   460   537   485
  511   464   532   460   536   485   511   486   510   487   1505    486   513   484   538   458   538   458   538   459   1534    433   1532    486   537   485   1508    486   1507    471   1525    457   535
  460   1531    487   510   487   1507    486   1532    434   563   496   474   483   539   457   1536    458   1534    487   1506    460   536   464   533   464   531   471   526   461   535   486   510   487
  1505    488   1485    507   1505    524   1470    486   1533    461   536   486   1505    487   1480    488   536   464   505   488   535   461   535   486   510   461   534   463   533   463   1559    433   1559
  494   1498    460   1533    486
          name KEY_4
```

```
                539  17859  2998  8958  483  540  459  1534  486  510  485  511  486  511  460  511  484  537  460  536  465  532  486  1506  486  510  486  512
459  1558  455  543  432  563  434  1533  484  1509  486  1531  461  1531  462  1532  485  511  485  511  485  516  480  511  486  511  486  538  433  535  515  509  433
563  433  563  456  541  456  541  458  543  454  536  461  536  461  509  486  536  461  535  467  529  485  511  485  512  486  510  485  516  481  512  460  562  489  507
455  516  458  563  433  564  458  539  459  537  460  536  472  1525  456  1507  485  1532  486  1507  483  2990  3018  8965  484  1534  433  563  491  504  434  562  434
563  457  514  484  538  486  511  460  509  487  1531  462  536  484  485  487  536  461  1531  491  505  487  1505  487  536  458  1537  455  1537  432  1533  491  1527
486  1481  487  1532  461  1532  485  1506  491  505  487  510  461  535  485  1535  458  1534  457  1507  462  563  464  533  458  538  460  509  487  536  485  1480  487
537  460  1533  460  1532  486  1504  462  1532  461  562  460  1533  434  1560  460  509  484  512  486  537  460  509  487  536  460  537  460  536  460  1532  486  1512
481  1506  460  1532  485
```

name KEY_5

```
                597  17954  3015  8962  485  513  486  1506  515  482  514  481  513  482  516  482  515  482  514  482  514  483  515  1478  513  481  494  530
462  1530  434  568  451  541  479  1512  512  1481  514  1478  515  1479  515  1477  514  482  527  470  513  483  514  483  491  532  487  487  510  508  488  509  452
544  457  539  481  515  485  512  513  483  514  482  515  482  515  481  515  481  515  481  514  482  515  483  514  482  514  483  513  482  514  509  488  509  434  562
455  541  456  541  482  520  479  511  514  483  515  482  514  1477  514  1480  515  1476  515  1484  454  3039  2940  9017  489  1531  486  510  432  565  455  540  479
517  484  512  512  486  513  484  513  482  514  1478  515  483  519  476  515  1478  514  1479  512  484  513  1506  487  510  487  1511  451  1535  482  1510  514  1480
514  1478  515  1483  510  1476  516  1479  514  482  514  481  515  509  487  1505  433  1560  456  1537  483  513  486  510  515  482  514  482  513  483  515  481  521
476  514  1479  514  1479  512  1479  489  1531  487  509  493  1500  456  1536  483  514  485  511  514  482  515  482  514  482  515  481  520  477  513  1479  515  1479
513  1479  513  1505  484
```

end raw_codes

end remote

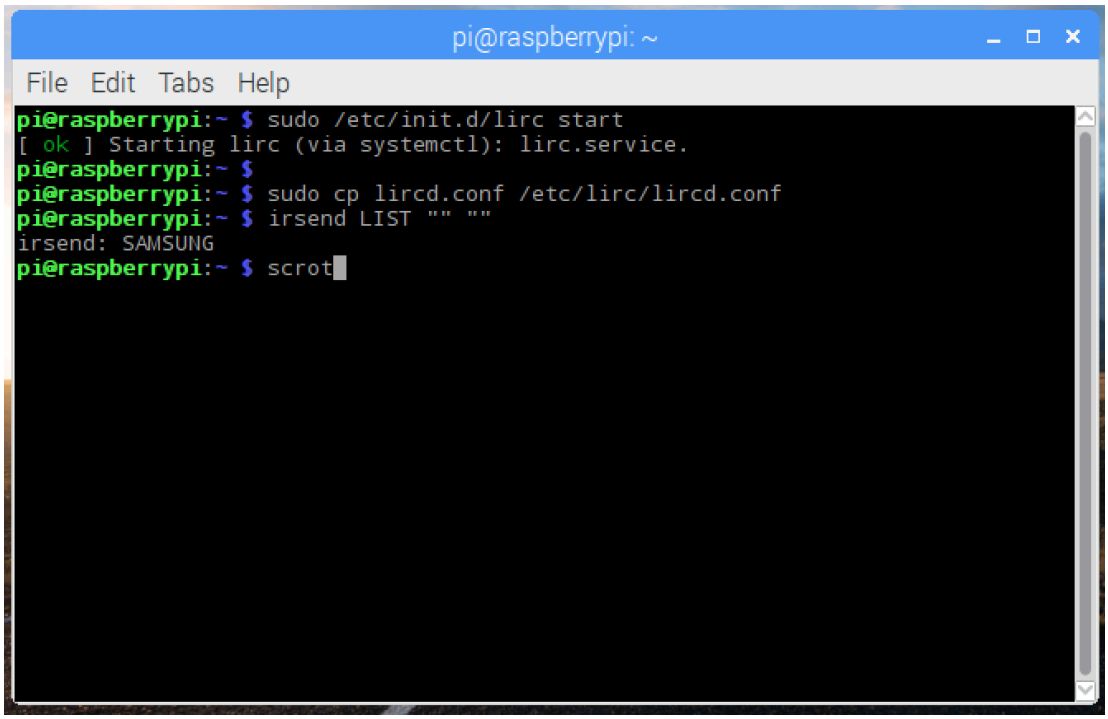
Testing the Generated Configuration File

Start lirc to send the the signal to the ac from the IR LED

sudo /etc/init.d/lirc start

Copy the generated file the to lirc conf file

sudo cp lircd.conf /etc.lirc/lircd.conf



By using irsend list command displays all available remotes

irsend LIST “” “”

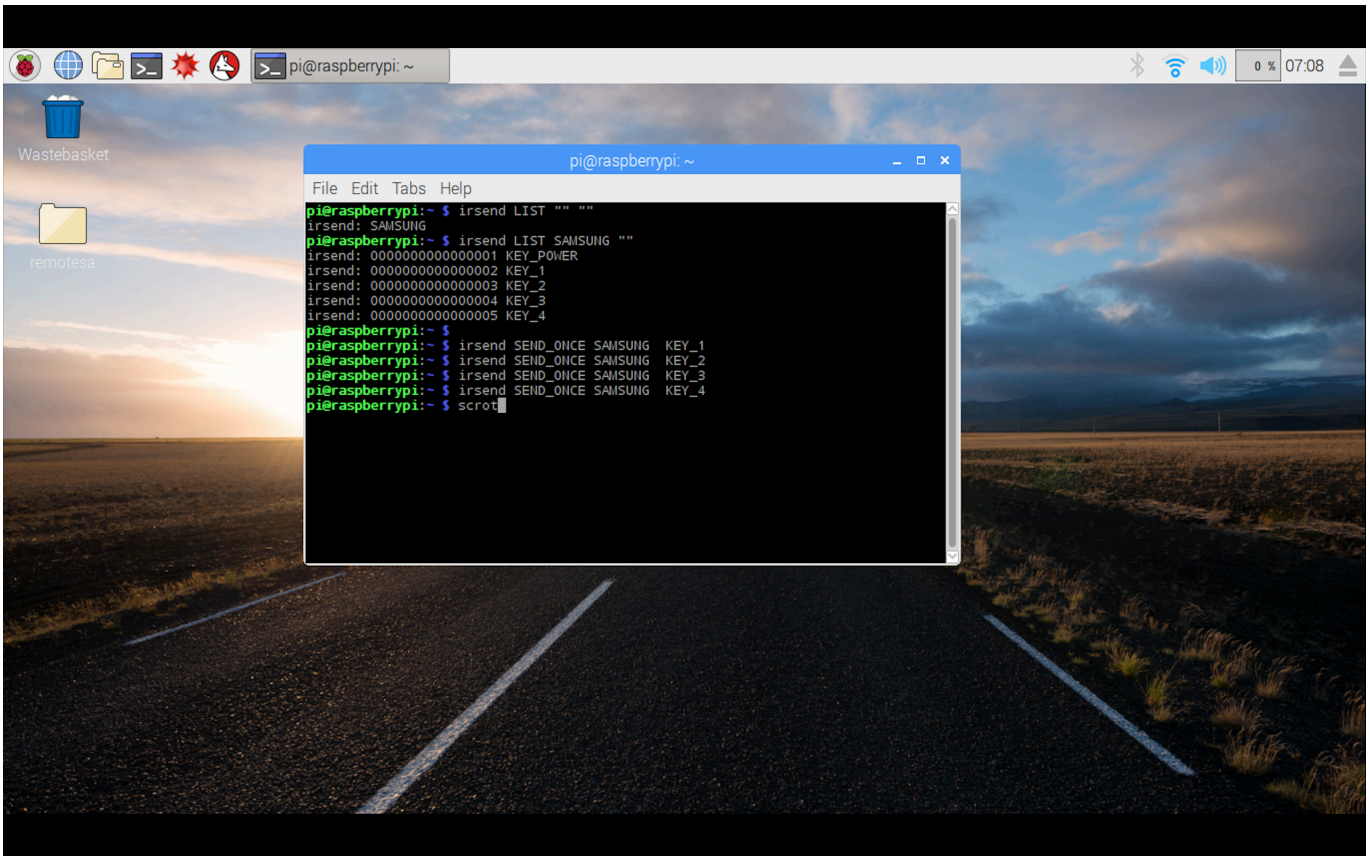
By using irsend list remote name displays all available commands for the remote

irsend LIST REMOTE_NAME “”

SENDING SIGNAL TO THE AC

IRSEND tool is used to send the IR signals to the AC

irsend SEND_ONCE REMOTE_NAME KEY_NAME



POSSIBLE ERRORS:

UNKNOW REMOTE:

- This is caused when a wrong configuration file is generated or if there are any mistakes
- Wrong remote name
- wrong configuration file is copied to etc/lirc/lircd.conf path .Check the copied config file using editor sudo nano /etc/lirc/lircd.conf
- Try reloading lirc and again copying the config file to lirc folder

UNABLE TO CONNECT TO SOCKET:

- LIRC is stopped try starting it by using sudo /etc/init.d/lirc start
- Reload LIRC by using sudo /etc/init.d/lirc start

References:

- lirc.org
- Reverse Engineering Air Conditioner IR remote protocols
- Setting up LIRC on Raspberry pi by Alex
- Popular IR protocols
- Waveforms of IR samples

-Documented by

Baswaraj mamidgi

baswarajmamidgi10@gmail.com

9989478011