# Download Base Image

In general base Rpi images can be found here:

<https://www.raspberrypi.org/downloads/raspbian/>

We will use RASPBIAN STRETCH WITH DESKTOP

<https://downloads.raspberrypi.org/raspbian_latest>

* Need to make sure the SD card is not expanded to use all the space.
* 6 GB is allocated to the root partition.
* 16 GB is allocated for video storage and is formatted using a XFS file system
* See this link for more information on how this was done. <https://www.raspberrypi.org/forums/viewtopic.php?p=1114520>
* A private copy of was made of raspi-config and modified to allocate 6GB to the root partition.

# sudo fdisk -l

Disk /dev/mmcblk0: 29.7 GiB, 31914983424 bytes, 62333952 sectors

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0x020c3677

Device Boot Start End Sectors Size Id Type

/dev/mmcblk0p1 8192 93814 85623 41.8M c W95 FAT32 (LBA)

/dev/mmcblk0p2 94208 12677119 12582912 6G 83 Linux

/dev/mmcblk0p3 12677120 46231551 33554432 16G 83 Linux

# Create Users

The table specifies the default users and passwords that will be part of the image.

|  |  |  |
| --- | --- | --- |
| User | Password | Sudo Priviledges |
| root | r00t | yes |
| pi | pi | yes |
| camera | camera | yes |

The following link gives an overview of how to manage users:

<https://www.raspberrypi.org/documentation/linux/usage/users.md>

1) Change the password for root:

sudo passwd root

2) Change the password for pi:

sudo passwd pi

3) Create the new user camera:

sudo adduser camera

sudo visudo (to add camera as super user)

|  |
| --- |
| # User privilege specification  root ALL=(ALL:ALL) ALL  camera ALL = NOPASSWD: ALL |

sudo usermod -a -G pi,adm,dialout,cdrom,audio,video,plugdev,games,users,input,netdev,gpio,i2c,spi camera

# Set Default Editor to VI

sudo update-alternatives --set editor /usr/bin/vim.tiny

To change the default command-line text editor, follow these steps:

1. Log in to your account using SSH.
2. Open the .bashrc file in your preferred text editor.
3. Add the following lines to the .bashrc file. Replace both occurrences of program with the editor you want to set as the default editor:

|  |
| --- |
| export EDITOR='program'  export VISUAL='program'  set -o ‘program’ |

To set vi as the default text editor, replace program with vi.

To set nano as the default text editor, replace program with nano

export EDITOR='vi'

export VISUAL='vi'

set -o 'vi'

1. Save the changes to the .bashrc file and exit the text editor.
2. To make the new default text editor settings take effect, log out of your account and then log back in.

This is setup in the default /etc/skel/.bashrc file

This is done for root under /root/.bashrc.

This is done for the pi user in the home directory

This is done for the camera user in the home directory

# Upgrade Raspbian with the latest

Check for Raspian OS and software updates with the following commands:

sudo apt-get update

sudo apt-get dist-upgrade

Check for firmware updates, Raspian UI modifications and clean your installation with the following commands:

sudo rpi-update

sudo apt-get install raspberrypi-ui-mods

sudo apt-get clean

The Raspberry Pi firmware allows the hardware to communicate with the software. On a full size PC, the firmware normally resides on an integrated circuit on the motherboard. On the Raspberry Pi, the firmware resides on an area (partition) of the SD card. This configuration allows firmware updates to be simple and reduces hardware costs.

At the terminal prompt, type the following command to update the Raspberry Pi firmware:

sudo rpi-update

# **Raspian OS and software update**

After a firmware update, update the Raspian OS and software using the commands below:

sudo apt-get update

The apt-get update option connects the Raspberry Pi to the online software repository database and compares available software to installed software. From this comparison, apt develops a list of packages that need an update. It may take up to a minute or two for the update to finish.

After the repository updates, the Raspberry Pi can now determine what Raspian OS and software updates are available. Updates often contain security enhancements and software feature additions. It is always good practice to use this command before you use upgrade commands.

To identify and apply the updates, use the command below:

sudo apt-get dist-upgrade

The Raspberry Pi will display a list of the updates.

If you only want to update software and not the Raspian OS, you can use the commands below:

sudo apt-get update

sudo apt-get upgrade

I recommend using the dist-upgrade option. To me, it just makes sense to update both the OS and all software on the Raspberry Pi.

# **Raspian user interface update**

Occasionally, the Raspian OS receives user interface updates that are not included in the base updates. Check for and install these updates using the commands below:

sudo apt-get update

sudo apt-get install raspberrypi-ui-mods

# **Clean up**

After upgrades, it’s good practice to execute the apt clean option. This will recover additional space on the SD card. During the installation process, aptdownloads .deb files. These are installation files that are no longer needed once the application is installed. Removing them will free space on your SD card. This is especially valuable if you use a 4Gb SD card.

Use the command below to clean the Raspberry Pi SD card:

sudo apt-get clean

# Customize Desktop

- Remove apps off of the top bar.

- Color scheme choice

# Raspberry Pi Configuration

Can be done from the GUI or rapsi-config:

- Set hostname

- Enable camera

- Enable ssh

- set GPU Memory = 256

- Set Localisation to English and USA.

- Set Timezeone, keyboard, and Wifi country.

# Configure HDMI defaults

Edit /boot/config.txt

|  |
| --- |
| disable\_overscan=1  hdmi\_force\_hotplug=1  hdmi\_group=1  hdmi\_mode=16 |

# Enabling the Serial Port for Wiring GPS

# sudo raspi-config

Option P6 - Enable/Disable shell and kernel messages on the serial connection

Set to NO - Would you like a login shell to be accessible over serial?

Set to YES - Would you like the serial port hardware to be enabled?

# Auto-login As Camera User

Edit /etc/lightdm/lightdm.conf

Change the entry for autologin-user

autologin-user=camera

#

# Not sure if the following is required

#

Edit /etc/systemd/system/autologin@.service

Change the entry for the autologin to be the camera user

# Setup NTP

1. Uninstall the existing Raspbian Stretch TimeSync service.

sudo systemctl stop systemd-timesyncd

sudo systemctl daemon-reload

sudo systemctl disable systemd-timesyncd

2. Instructions for setting up the control.local processor as the NTP server.

* <http://www.satsignal.eu/ntp/Raspberry-Pi-NTP.html>
* <https://developers.redhat.com/blog/2017/02/22/how-to-build-a-stratum-1-ntp-server-using-a-raspberry-pi/>
* http://unixwiz.net/techtips/raspberry-pi3-gps-time.html

3. Install NTP Software on the Clients (See above for compiling NTP for Server)

sudo apt-get install ntp

sudo apt-get install ntpdate

4. Make sure you configure the ntp.conf file as follows:

/etc/ntpconf file on NTP Server processor (control.local)

|  |
| --- |
| # /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help  driftfile /var/lib/ntp/ntp.drift  # Enable this if you want statistics to be logged.  # statsdir /var/log/ntpstats/    statistics loopstats peerstats clockstats  filegen loopstats file loopstats type day enable  filegen peerstats file peerstats type day enable  filegen clockstats file clockstats type day enable  # PPS(0), gpsd: /dev/pps0: Kernel-mode PPS ref-clock for the precise seconds  # http://doc.ntp.org/current-stable/drivers/driver22.html  server 127.127.22.0 minpoll 4 maxpoll 4 true  fudge 127.127.22.0 refid PPS  # SHM(0), gpsd: Server from shared memory provided by gpsd  # # http://doc.ntp.org/current-stable/drivers/driver28.html  server 127.127.28.0 prefer  fudge 127.127.28.0 refid GPS time1 +0.500 stratum 2  # pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will  # pick a different set every time it starts up. Please consider joining the  # pool: <http://www.pool.ntp.org/join.html>  server 0.us.pool.ntp.org  server 1.us.pool.ntp.org  server 2.us.pool.ntp.org  server 3.us.pool.ntp.org  # Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for  # details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>  # might also be helpful.  #  # Note that "restrict" applies to both servers and clients, so a configuration  # that might be intended to block requests from certain clients could also end  # up blocking replies from your own upstream servers.  # By default, exchange time with everybody, but don't allow configuration.  restrict -4 default kod notrap nomodify nopeer noquery limited  restrict -6 default kod notrap nomodify nopeer noquery limited  # Local users may interrogate the ntp server more closely.  restrict 127.0.0.1  restrict ::1  # Needed for adding pool entries  # restrict source notrap nomodify noquery  # Clients from this (example!) subnet have unlimited access, but only if  # cryptographically authenticated.  # restrict 192.168.123.0 mask 255.255.255.0 notrust  # Give machines on the local network access to query for time  restrict 169.0.0.0 mask 255.0.0.0 nomodify notrap  restrict 10.0.0.0 mask 255.0.0.0 nomodify notrap  # If you want to provide time to your local subnet, change the next line.  # (Again, the address is an example only.)  # broadcast 192.168.123.255  # If you want to listen to time broadcasts on your local subnet, de-comment the  # next lines. Please do this only if you trust everybody on the network!  # disable auth  # broadcastclient  #  # Fix false tickers  tos mindist 0.5 |

/etc/ntp.conf file on clients (left.local, front.local, and rear.local)

|  |
| --- |
| # /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help  driftfile /var/lib/ntp/ntp.drift  # Enable this if you want statistics to be logged.  #statsdir /var/log/ntpstats/  statistics loopstats peerstats clockstats  filegen loopstats file loopstats type day enable  filegen peerstats file peerstats type day enable  filegen clockstats file clockstats type day enable  # You do need to talk to an NTP server or two (or three).  server control.local prefer  server 0.us.pool.ntp.org  server 1.us.pool.ntp.org  server 2.us.pool.ntp.org  server 3.us.pool.ntp.org  # pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will  # pick a different set every time it starts up. Please consider joining the  # pool: <http://www.pool.ntp.org/join.html>  # pool 0.debian.pool.ntp.org iburst  # pool 1.debian.pool.ntp.org iburst  # pool 2.debian.pool.ntp.org iburst  # pool 3.debian.pool.ntp.org iburst  # Access control configuration; see /usr/share/doc/ntp-doc/html/accopt.html for  # details. The web page <http://support.ntp.org/bin/view/Support/AccessRestrictions>  # might also be helpful.  #  # Note that "restrict" applies to both servers and clients, so a configuration  # that might be intended to block requests from certain clients could also end  # up blocking replies from your own upstream servers.  # By default, exchange time with everybody, but don't allow configuration.  restrict -4 default kod notrap nomodify nopeer noquery limited  restrict -6 default kod notrap nomodify nopeer noquery limited  # Local users may interrogate the ntp server more closely.  restrict 127.0.0.1  restrict ::1  # Needed for adding pool entries  restrict source notrap nomodify noquery  # Clients from this (example!) subnet have unlimited access, but only if  # cryptographically authenticated.  # restrict 169.254.129.0 mask 255.255.255.0 nomodify notrap noquery  # If you want to provide time to your local subnet, change the next line.  # (Again, the address is an example only.)  # broadcast 169.254.129.0  # If you want to listen to time broadcasts on your local subnet, de-comment the  # next lines. Please do this only if you trust everybody on the network!  #disable auth |

5. Restarting the ntp service

sudo service ntp stop

sudo ntpdate -u control.local

sudo service ntp start

6. To see that status of ntp

ntpq -p -n

You will see output something like this:

camera@left:~ $ ntpq -p -n

remote refid st t when poll reach delay offset jitter

==============================================================================

\*169.254.139.188 .PPS. 1 u 647 1024 377 0.628 1.097 0.633

+96.126.105.86 132.246.11.231 2 u 941 1024 377 37.896 -3.717 1.154

+216.228.47.167 64.62.153.210 3 u 932 1024 375 78.501 1.144 1.082

+2604:a880:2:d0: 216.218.254.202 2 u 905 1024 377 72.679 -0.754 5.729

+69.89.207.99 .PPS. 1 u 972 1024 377 36.817 4.476 2.635

The astericks shows where the clock is getting it’s sync.

# Set up browser settings for Chromium

This section to be completed later.

# Install VLC software

# # Not needed anymore.

sudo apt-get install vlc

# Install GPAC software

sudo apt-get install gpac

# Copy VTE software to camera user

Go to $HOME directory of camera user

git clone <https://github.com/raether/vte.git>

This will create a directory called VTE with the software.

# Install GPS Daemon software

sudo apt-get install gpsd gpsd-clients python-gps

sudo systemctl stop gpsd.socket

sudo systemctl disable gpsd.socket

sudo service gpsd start

Setup listener for GPSD on all IP addresses, not just the local loopback.

/lib/systemd/system/gpsd.socket file:

|  |
| --- |
| [Unit]  Description=GPS (Global Positioning System) Daemon Sockets  [Socket]  ListenStream=/var/run/gpsd.sock  ListenStream=[::1]:2947  ListenStream=0.0.0.0:2947  SocketMode=0600  [Install]  WantedBy=sockets.target |

/etc/default/gpsd file:

|  |
| --- |
| # Default settings for the gpsd init script and the hotplug wrapper.  # Start the gpsd daemon automatically at boot time  START\_DAEMON="true"  # Use USB hotplugging to add new USB devices automatically to the daemon  USBAUTO="false"  # Devices gpsd should collect to at boot time.  # They need to be read/writeable, either by user gpsd or the group dialout.  DEVICES=""  # Other options you want to pass to gpsd  GPSD\_OPTIONS="-D2 -S2947 -n -G /dev/ttyS0 /dev/pps0"  GPSD\_SOCKET="/var/run/gpsd.sock" |

The ListenStream needs to get change to 0.0.0.2947 so the gps3 python library can connect. Not exactly sure why this is needed.

/lib/systemd/system/gpsd.socket

|  |
| --- |
| [Unit]  Description=GPS (Global Positioning System) Daemon Sockets  [Socket]  ListenStream=/var/run/gpsd.sock  ListenStream=[::1]:2947  ListenStream=0.0.0.0:2947  SocketMode=0600  [Install]  WantedBy=sockets.target |

# Enabling the Serial Port for Wiring GPS

# sudo raspi-config

Option P6 - Enable/Disable shell and kernel messages on the serial connection

Set to NO - Would you like a login shell to be accessible over serial?

Set to YES - Would you like the serial port hardware to be enabled?

# Install Navit software

#

# Not needed anymore

#

sudo apt-get install navit

install illinois maps

* Copy Illinois.bin under $HOME/vte/maps

install navit.xml configuration under $HOME/.navit

# How to set up SSH keys

Steps to setup secure ssh keys:

1. Create the key pair using ssh-keygen command.
2. Copy and install the public key using ssh-copy-id command.

# Setting up autostart for vte.sh

copy autostart from $HOME/vte/config to $HOME/.config/lxsession/LXDE-pi

# Install Bluetooth Dongle

We will disable the native Rpi bluetooth and use a USB dongle for bluetooth instead for audio playback and recording.

1. Remove bluez

# sudo apt-get purge bluealsa -y

2. Install pulseaudio

# sudo apt-get install pulseaudio pulseaudio-module-bluetooth  
# dpkg -l pulseaudio pulseaudio-module-bluetooth

3. To turn off built-in Bluetooth controller (BCM43438), blacklist it:

# sudo vi /etc/modprobe.d/raspi-blacklist.conf

Add lines:  
blacklist btbcm  
blacklist hci\_uart

4. Install the firmware for the USB dongle which is located in the ../vte/config directory.

# sudo cp vte/config/BCM20702A1-0b05-17cb.hcd /lib/firmware/brcm/

# reboot

5. Add the following lines to /etc/pulse/default.pa

# sudo vi /etc/pulse/default.pa

# Load Bluetooth automatically

load-module module-switch-on-connect

Reference Links:

<http://youness.net/raspberry-pi/bluetooth-headset-raspberry-pi-3-ad2p-hsp>

<http://youness.net/raspberry-pi/how-to-connect-bluetooth-headset-or-speaker-to-raspberry-pi-3>

<https://scribles.net/enabling-hands-free-profile-on-raspberry-pi-raspbian-stretch-by-using-pulseaudio/>

# Install Kivy

1. Install the dependencies:

sudo apt-get update

sudo apt-get install libsdl2-dev libsdl2-image-dev libsdl2-mixer-dev libsdl2-ttf-dev \

pkg-config libgl1-mesa-dev libgles2-mesa-dev \

python-setuptools libgstreamer1.0-dev git-core \

gstreamer1.0-plugins-{bad,base,good,ugly} \

gstreamer1.0-{omx,alsa} python-dev libmtdev-dev \

xclip xsel

2. Install a new enough version of Cython:

sudo pip install -U Cython==0.28.2

3. Install Kvy globally on the system:

sudo pip install git+https://github.com/kivy/kivy.git@master

4. The kivy configuration file is stored in the “.kivy” directory under the home directory. Add the following lines to the end of the config.ini file.

[modules]

cursor=1

touchring = show\_cursor=true

5. The user running kivy needs the right group permissions to support accessing the devices, or the mouse cursor might not show up.

References:

The kivy directory with samples is stored under:

/usr/local/share/kivy-examples

<https://kivy.org/doc/stable/installation/installation-rpi.html>

# Install Infra-Red Control

1. Install LIRC:

sudo apt-get install lirc

2. Add to your /etc/modules file by entering the command below:

sudo cat >> /etc/modules <<EOF

lirc\_dev

lirc\_rpi gpio\_in\_pin=23 gpio\_out\_pin=24

EOF

3, /etc/lirc/hardware.conf file should contain the following

########################################################

# /etc/lirc/hardware.conf

#

# Arguments which will be used when launching lircd

#

LIRCD\_ARGS="--uinput --listen"

#

# 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=""

########################################################

4. Edit your /boot/config.txt by entering the command below:

cat >> /boot/config.txt <<EOF

dtoverlay=lirc-rpi,gpio\_in\_pin=23,gpio\_out\_pin=24

EOF

5. Add the IR codes to the /etc/lirc/lircd.conf file

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

6. Update the following lines in /etc/lirc/lirc\_options.conf

driver = default

device = /dev/lirc0

6. Now restart lircd so it picks up these changes:

sudo /etc/init.d/lirc stop

sudo /etc/init.d/lirc start

References:

<https://www.hackster.io/austin-stanton/creating-a-raspberry-pi-universal-remote-with-lirc-2fd581>

https://gist.github.com/prasanthj/c15a5298eb682bde34961c322c95378b