RaspiCam Documentation

This document describes the use of the three Raspberry Pi camera applications as of November 15th 2013.

There are three applications provided, raspistill, raspivid and raspistillyuv. raspistill and raspistillyuv are very similar and are intended for capturing images, raspivid is for capturing video.

All the applications are command line driven, written to take advantage of the mmal API which runs over OpenMAX. The mmal API provides an easier to use system than that presented by OpenMAX. Note that mmal is a Broadcom specific API used only on Videocore 4 systems.

The applications use up to four OpenMAX(mmal) components - camera, preview, encoder and null_sink. All applications use the camera component, raspistill uses the Image Encode component, raspivid uses the Video Encode component and raspistillyuv does not use an encoder, and sends its YUV or RGB output direct from camera component to file.

The preview display is optional, but can be used full screen or directed to a specific rectangular area on the display. If preview is disabled, the null_sink component is used to 'absorb' the preview frames. It is necessary for the camera to produce preview frames even if not required for display, as they are used for calculating exposure and white balance settings.

In addition it is possible to omit the filename option, in which case the preview is displayed but no file is written, or to redirect all output to stdout.

Command line help is available by typing just the application name in on the command line.

Setting up the Camera hardware

Warning. Cameras are static sensitive. Earth yourself prior to handling the PCB, a sink tap/faucet or similar should suffice if you don't have an earthing strap.

The camera board attaches to the Raspberry Pi via a 15 way ribbon cable. There are only two connections to make, the ribbon cable need to be attached to the camera PCB and the Raspberry Pi itself. You need to get it the right way round or the camera will not work. On the camera PCB, the blue backing on the cable should be away from the PCB, and on the Raspberry Pi it should be towards the Ethernet connection (or where the Ethernet connector would be if you are using a model A).

Although the connectors on the PCB and the Pi are different, they work in a similar way. On the

Raspberry Pi, pull up the tabs on each end of the connector. It should slide up easily, and be able to pivot around slightly. Fully insert the ribbon cable into the slot, ensuring it is straight, then gently press down the tabs to clip it into place. The camera PCB itself also requires you to pull the tabs away from the board, gently insert the cable, then push the tabs back. The PCB connector is a little more awkward than the one on the Pi itself.

Setting up the Camera software

Execute the following instructions on the command line to download and install the latest kernel, GPU firmware and applications. You will need a internet connection for this to work correctly.

```
sudo apt-get update
sudo apt-get upgrade
```

Now you need to enable camera support using the raspi-config program you will have used when you first set up your Raspberry Pi.

```
sudo raspi-config
```

Use the cursor keys to move to the camera option and select enable. On exiting raspi-config it will ask to reboot. The enable option will ensure that on reboot the correct GPU firmware will be running (with the camera driver and tuning), and the GPU memory split is sufficient to allow the camera to acquire enough memory to run correctly.

To test that the system is installed and working, try the following command :

```
raspistill -v -o test.jpg
```

The display should show a 5 second preview from the camera and then take a picture, saved to the file test.jpg, whilst display various informational messages.

Troubleshooting

If the camera is not working correctly, there are number of things to try.

- Are the ribbon connectors all firmly seated and the right way round? They must be straight in their sockets.

- Is the camera module connector firmly attached to the camera PCB? This is the connection from the smaller black camera module itself to the camera PCB. Sometimes this connection can come loose. Using a fingernail, flip up the connector on the PCB, then reseat it with gentle pressure, it engages with a very slight click.
- Have sudo apt-get update, sudo apt-get upgrade been run?
- Has raspi-config been run and the camera enabled?
- Is your power supply sufficient? The camera adds about 200-250mA to the power requirements of the Raspberry Pi.

So, if things are still not working, try the following:

Error : raspistill/raspivid not found. This probably means your update/upgrade failed in some way. Try it again.

Error : ENOMEM displayed. Camera is not starting up. Check all connections again.

Error : ENOSPC displayed. Camera is probably running out of GPU memory. Check config.txt in the /boot/ folder. The gpu_mem option should be at least 128.

If after all the above, the camera is still not working, it may be defective. Try posting on the Raspberry Pi forum (Camera section) to see if there is any more help available there.

Common Command line Options

Preview Window

    --preview,   -p   Preview window settings <'x,y,w,h'>

Allows the user to define the size and location on the screen that the preview window will be placed. Note this will be superimposed over the top of any other windows/graphics.

    --fullscreen,   -f   Fullscreen preview mode

Forces the preview window to use the whole screen. Note that the aspect ratio of the incoming image will be retained, so there may be bars on some edges.

    --nopreview,   -n,   Do not display a preview window

Disables the preview window completely. Note that even though the preview is disabled, the camera will still be producing frames, so will be using power.

--opacity, -op Set preview window opacity

Sets the opacity of the preview windows. 0 = invisible, 255 = fully opaque.

Camera Control Options

--sharpness, -sh Set image sharpness (-100 to 100)

Set the sharpness of the image, 0 is the default.

--contrast, -co Set image contrast (-100 to 100)

Set the contrast of the image, 0 is the default

--brightness, -br Set image brightness (0 to 100)

Set the brightness of the image, 50 is the default. 0 is black, 100 is white.

--saturation, -sa Set image saturation (-100 to 100)

set the colour saturation of the image. 0 is the default.

--ISO, -ISO Set capture ISO

Sets the ISO to be used for captures. Range is 100 to 800.

--vstab, -vs Turn on video stabilisation

In video mode only, turn on video stabilisation.

--ev, -ev Set EV compensation

Set the EV compensation of the image. Range is -10 to +10, default is 0.

--exposure, -ex Set exposure mode

Possible options are:

auto	Use automatic exposure mode	
night	Select setting for night shooting	
nightpreview		
backlight	Select setting for back lit subject	
spotlight		
sports	Select setting for sports (fast shutter etc)	
snow	Select setting optimised for snowy scenery	
beach	Select setting optimised for beach	
verylong	Select setting for long exposures	
fixedfps,	Constrain fps to a fixed value	
antishake	Antishake mode	
fireworks	Select settings	

Note that not all of these settings may be implemented, depending on camera tuning.

--awb,   -awb   Set Automatic White Balance (AWB) mode

off	Turn off white balance calculation
auto	Automatic mode (default)
sun	Sunny mode
cloud	Cloudy mode
shade	Shaded mode
tungsten	Tungsten lighting mode
fluorescent	Fluorescent lighting mode
incandescent	Incandescent lighting mode
flash	Flash mode
horizon	Horizon mode

--imxfx,   -ifx   Set image effect

Set an effect to be applied to the image

none	NO effect (default)
negative	Negate the image
solarise	Solarise the image
posterize	Posterise the image
whiteboard	Whiteboard effect
blackboard	Blackboard effect
sketch	Sketch style effect
denoise	Denoise the image
emboss	Emboss the image
oilpaint	Apply an oil paint style effect

hatch	Hatch sketch style
gpen	
pastel	A pastel style effect
watercolour	A watercolour style effect
film	Film grain style effect
blur	Blur the image
saturation	Colour saturate the image
colourswap	Not fully implemented
washedout	Not fully implemented
posterise	Not fully implemented
colourpoint	Not fully implemented
colourbalance	Not fully implemented
cartoon	Not fully implemented

--colfx,   -cfx   Set colour effect <U:V>

The supplied U and V parameters (range 0 to 255) are applied to the U and Y channels of the image. For example, --colfx 128:128 should result in a monochrome image.

--metering,   -mm   Set metering mode

Specify the metering mode used for the preview and capture

average	Average the whole frame for metering.
spot	Spot metering
backlit	Assume a backlit image
matrix	Matrix metering

--rotation,   -rot Set image rotation (0-359)

Sets the rotation of the image in viewfinder and resulting image. This can take any value from 0 upwards, but due to hardware constraints only 0, 90, 180 and 270 degree rotations are supported.

--hflip,   -hf   Set horizontal flip

Flips the preview and saved image horizontally.

--vflip,   -vf   Set vertical flip

Flips the preview and saved image vertically.

--roi,	-roi	Set sensor region of interest

Allows the specification of the area of the sensor to be used as the source for the preview and capture. This is defined as x,y for the top left corner, and a width and height, all values in normalised coordinates (0.0-1.0). So to set a ROI at half way across and down the sensor, and an width and height of a quarter of the sensor use :

-roi 0.5,0.5,0.25,0.25

--shutter,	-ss	Set shutter speed

Set the shutter speed to the specified value (in microseconds). There is currently an upper limit of approximately 330000us (330ms, 0.33s) past which operation is undefined. This is being investigated.

Application specific settings

raspistill

--width,	-w	Set image width <size>
--height,	-h	Set image height <size>
--quality,	-q	Set jpeg quality <0 to 100>

Quality 100 is almost completely uncompressed. 75 is a good all round value

--raw,	-r	Add raw bayer data to jpeg metadata

This option inserts the raw Bayer data from the camera in to the JPEG metadata

--output	-o	Output filename <filename>.

Specify the output filename. If not specified, no file is saved. If the filename is '-', then all output is sent to stdout.

--latest	-l	Link latest frame to filename <filename>

Make a file system link under this name to the latest frame.

--verbose,	-v	Output verbose information during run

Outputs debugging/information messages during the program run.

--timeout,	-t	Time before takes picture and shuts down.

The program will run for this length of time, then take the capture (if output is specified). If not specified, this is set to 5 seconds.

--timelapse,	-tl	Timelapse mode.

The specific value is the time between shots in milliseconds. Note you should specify %04d at the point in the filename where you want a frame count number to appear. e.g.

        -t 30000 -tl 2000 -o image%04d.jpg

will produce a capture every 2 seconds, over a total period of 30s, named image1.jpg, image0002.jpg..image0015.jpg. Note that the %04d indicates a 4 digit number with leading zero's added to pad to the required number of digits. So, for example, %08d would result in an 8 digit number.

If a timelapse value of 0 is entered, the application will take pictures as fast as possible. Note there is an minimum enforced pause of 30ms between captures to ensure that exposure calculations can be made.

--thumb,	-th	Set thumbnail parameters (x:y:quality)

Allows specification of the thumbnail image inserted in to the JPEG file. If not specified, defaults are a size of 64x48 at quality 35.

if '--thumb none' is specified, no thumbnail information will be placed in the file. This reduces the file size slightly.

--demo,	-d	Run a demo mode <milliseconds>

This options cycles through range of camera options, no capture is done, the demo will end at the end of the timeout period, irrespective of whether all the options have been cycled. The time between cycles should be specified as a millisecond value.

--encoding,	-e	Encoding to use for output file

Valid options are jpg, bmp, gif and png. Note that unaccelerated image types (gif, png, bmp) will take much longer to save than JPG which is hardware accelerated. Also note that the filename suffix is completely ignored when deciding the encoding of a file.

--exif,   -x   EXIF tag to apply to captures (format as 'key=value')

Allows the insertion of specific exif tags in to the JPEG image. You can have up to 32 exif tge entries. This is useful for things like adding GPS metadata. For example, to set the Longitude

--exif GPS.GPSLongitude=5/1,10/1,15/100

would set the Longitude to 5degs, 10 minutes, 15 seconds. See exif documentation for more details on the range of tags available; the supported tags are as follows.

IFD0.< or
IFD1.<
ImageWidth, ImageLength, BitsPerSample, Compression, PhotometricInterpretation, ImageDescription, Make, Model, StripOffsets, Orientation, SamplesPerPixel, RowsPerString, StripByteCounts, Xresolution, Yresolution, PlanarConfiguration, ResolutionUnit, TransferFunction, Software, DateTime, Artist, WhitePoint, PrimaryChromaticities, JPEGInterchangeFormat, JPEGInterchangeFormatLength, YcbCrCoefficients, YcbCrSubSampling, YcbCrPositioning, ReferenceBlackWhite, Copyright>

EXIF.<
ExposureTime, FNumber, ExposureProgram, SpectralSensitivity,   a
ISOSpeedRatings, OECF, ExifVersion, DateTimeOriginal, DateTimeDigitized, ComponentsConfiguration, CompressedBitsPerPixel, ShutterSpeedValue, ApertureValue, BrightnessValue, ExposureBiasValue, MaxApertureValue, SubjectDistance, MeteringMode, LightSource, Flash, FocalLength, SubjectArea, MakerNote, UserComment, SubSecTime, SubSecTimeOriginal, SubSecTimeDigitized, FlashpixVersion, ColorSpace, PixelXDimension, PixelYDimension, RelatedSoundFile, FlashEnergy, SpacialFrequencyResponse, FocalPlaneXResolution, FocalPlaneYResolution, FocalPlaneResolutionUnit, SubjectLocation, ExposureIndex, SensingMethod, FileSource, SceneType, CFAPattern, CustomRendered, ExposureMode,
WhiteBalance, DigitalZoomRatio, FocalLengthIn35mmFilm, SceneCaptureType, GainControl, Contrast, Saturation, Sharpness, DeviceSettingDescription, SubjectDistanceRange, ImageUniqueID>

GPS.<
GPSVersionID, GPSLatitudeRef, GPSLatitude, GPSLongitudeRef, GPSLongitude, GPSAltitudeRef, GPSAltitude, GPSTimeStamp, GPSSatellites, GPSStatus, GPSMeasureMode, GPSDOP, GPSSpeedRef, GPSSpeed, GPSTrackRef, GPSTrack, GPSImgDirectionRef, GPSImgDirection, GPSMapDatum, GPSDestLatitudeRef, GPSDestLatitude, GPSDestLongitudeRef, GPSDestLongitude, GPSDestBearingRef, GPSDestBearing, GPSDestDistanceRef, GPSDestDistance, GPSProcessingMethod, GPSAreaInformation, GPSDateStamp, GPSDifferential>

EINT.<
InteroperabilityIndex, InteroperabilityVersion, RelatedImageFileFormat, RelatedImageWidth,
RelatedImageLength>

Note that a small subset of these tags will be set automatically by the camera system, but will be overridden by any exif options on the command line.

Setting '--exif none' will prevent any EXIF information being stored in the file. This reduces the file size slightly.

    --fullpreview,   -fp   Full Preview mode

This runs the preview windows using the full resolution capture mode. Maximum frames per second in this mode is 15fps and the preview will have the same field of view as the capture. Captures should happen more quickly as no mode change should be required. This feature is currently under development.

    --keypress   -k   Keypress mode

The camera is run for the requested time (-t), and a captures can be initiated throughout that by pressing the Enter key. Press X then Enter will exit the application before the timeout is reached. If the timeout is set to 0, the camera will run indefinitely until X then Enter is typed. Using the verbose option (-v) will display a prompt asking for user input, otherwise no prompt is displayed.

    --signal   -s   Signal mode

The camera is run for the requested time (-t), and a captures can be initiated throughout that time by sending a USR1 signal to the camera process. This can be done using the kill command. You can find the camera process ID using the 'pgrep raspistill' command.

        kill -USR1 <process id of raspistill>

raspistillyuv

Many of the options for raspistillyuv are the same as those for raspistill. This section shows the differences.

Unsupported Options:

--exif, --encoding, --thumb, --raw, --quality

Extra Options :

      --rgb,   -rgb   Save uncompressed data as RGB888

This option forces the image to be saved as RGB data with 8 bits per channel, rather than YUV420.

Note that the image buffers saved in raspistillyuv are padded to a horizontal size divisible by 16 (so there may be unused bytes at the end of each line to make the width divisible by 16). Buffers are also padded vertically to be divisible by 16, and in the YUV mode, each plane of Y,U,V is padded in this way.

raspivid

      --width,   -w   Set image width <size>

Width of resulting video. This should be between 64 and 1920.

      --height,   -h   Set image height <size>

Height of resulting video. This should be between 64 and 1080.

      --bitrate,   -b   Set bitrate.

Use bits per second, so 10MBits/s would be -b 10000000. For H264, 1080p30 a high quality bitrate would be 15Mbits/s or more. Maximum bitrate is 25Mbits/s (-b 25000000), but much over 17Mbits/s will not show noticeable improvement at 1080p30.

      --output   -o   Output filename <filename>.

Specify the output filename. If not specified, no file is saved. If the filename is '-', then all output is sent to stdout.

      --verbose,   -v   Output verbose information during run

Outputs debugging/information messages during the program run.

      --timeout,   -t   Time before takes picture and shuts down.

The program will run for this length of time, then take the capture (if output is specified). If not specified, this is set to 5seconds. Setting 0 will mean the application will run continuously until stopped with Ctrl-C.

--demo,   -d   Run a demo mode <milliseconds>

This options cycles through range of camera options, no capture is done, the demo will end at the end of the timeout period, irrespective of whether all the options have been cycled. The time between cycles should be specified as a millisecond value.

--framerate,   -fps   Specify the frames per second to record

At present, the minimum frame rate allowed is 2fps, the maximum is 30fps. This is likely to change in the future.

--penc,   -e   Display preview image *after* encoding

Switch on an option to display the preview after compression. This will show any compression artefacts in the preview window. In normal operation, the preview will show the camera output prior to being compressed. This option is not guaranteed to work in future releases.

--intra,   -g   Specify the intra refresh period (key frame rate/GoP)

Sets the intra refresh period (GoP) rate for the recorded video. H264 video uses a complete frame (I-frame) every intra refresh period from which subsequent frames are based. This options specifies the numbers of frames between each I-frame. Larger numbers here will reduce the size of the resulting video, smaller numbers make the stream more robust to error.

--qp,   -qp   Set quantisation parameter

Sets the initial quantisation parameter for the stream. Varies from approximately 10 to 40, and will greatly effect the quality of the recording. Higher values reduce quality and decrease file size. Combine this setting with bitrate of 0 to set a completely variable bitrate.

--profile,   -pf   Specify H264 profile to use for encoding

Sets the H264 profile to be used for the encoding. Options are : baseline, main, high.

--inline,   -ih   Insert PPS, SPS headers

Forces the stream to include PPS and SPS headers on every I-frame. Needed for certain streaming cases. e.g. Apple HLS. These headers are small, so do not greatly increase file size.

--timed   -td   Do timed switches between capture and pause

This options allows the video capture to be paused and restarted at particular time intervals. Two values are required, the On time and the Off time. On time is the amount of time the video is captured, off time is the amount it is paused. The total time of the recording is defined by the timeout option. Note the recording may take slightly over the timeout setting depending on the On and Off times.

For example

        raspivid -o test.h264 -t 25000 -timed 2500,5000

will record for a period of 25 seconds. The recording will be over a timeframe consisting of 2500ms (2.5s) segments with 5000ms (5s) gaps, repeating over the 20s. So the entire recording will actually be only 10s long, since 4 segments of 2.5s = 10s separated by 5s gaps.

        2.5 record – 5 pause - 2.5 record – 5 pause -2.5 record – 5 pause – 2.5 record

gives a total recording period of 25s, but only 10s of actual recorded footage

        --keypress   -k   Toggle between record and pause on ENTER key pressed

On each press of the ENTER key the recording will be paused or restarted. Pressing X then ENTER will stop recording and close the application. Note that the timeout value will be used to signal end of recording, but is only checked after each ENTER keypress, so if the system is waiting for a keypress, even if the timeout has expired, it will still wait for the keypress before exiting.

        --signal   -s   Toggle between record and pause according to SIGUSR1

Sending a USR1 signal to the raspivid process will toggle between recording and paused. This can be done using the kill command. You can find the raspivid process ID using 'pgrep raspivid'.

        kill -USR1 <process id of raspistill>

Note that the timeout value will be used to indicate the end of recording, but is only checked after each receipt of the SIGUSR1 signal, so if the system is waiting for a signal, even if the timeout has expired, it will still wait for the signal before exiting.

        --initial   -i   Define initial state on startup.

Define whether the camera will start paused or will immediately start recording. Options are 'record'

or 'pause'. Note that if you are using a simple timeout, and initial is set to 'pause', no output will be recorded.

> --segment   -sg   Segment the stream in to multiple files

Rather than creating a single file, the file is split up in to segments of approximately the numer of milliseconds specified. In order to provide different filesnames, you should add %04d or similar at the point in the filename where you want a segment count number to appear. e.g.

> --segment 3000 -o video%04d.h264

will produce video clips of approximately 3000ms (3s) long, named video0001.hs64, video0002.h264 etc. The clips should be seamless (no frame drops between clips), but the accuracy of each clip length will depend on the intraframe period, as the segments will always start on an I-frame. They will therefore always be equal or longer to the specified period.

> --wrap   -wr   Set the maximum value for segment number.

When outputting segments, this is the maximum the segment number can reach before it is reset to 1, giving the ability to keep recording segments, but overwriting the oldest one. So if set to four, in the segment example above, the files produced will be video0001.hs64, video0002.h264, video0003.hs64, video0004.h264. Once video0004.hs64 is recorded, the count will reset to 1, and the video0001.hs64 will be overwritten.

Examples

Still captures

By default, captures are done at the highest resolution supported by the sensor. This can be changed using the -w and -h command line options.

Taking a default capture after 2s (note times are specified in milliseconds) on viewfinder, saving in image.jpg

> raspistill -t 2000 -o image.jpg

Take a capture at a different resolution.

> raspistill -t 2000 -o image.jpg -w 640 -h 480

Now reduce the quality considerably to reduce file size

```
raspistill -t 2000 -o image.jpg -q 5
```

Force the preview to appear at coordinate 100,100, with width 300 and height 200 pixels.

```
raspistill -t 2000 -o image.jpg -p 100,100,300,200
```

Disable preview entirely

```
raspistill -t 2000 -o image.jpg -n
```

Save the image as a png file (lossless compression, but slower than JPEG). Note that the filename suffix is ignored when choosing the image encoding.

```
raspistill -t 2000 -o image.png –e png
```

Add some EXIF information to the JPEG. This sets the Artist tag name to Boris, and the GPS altitude to 123.5m. Note that if setting GPS tags you should set as a minimum GPSLatitude, GPSLatitudeRef, GPSLongitude, GPSLongitudeRef, GPSAltitude and GPSAltitudeRef.

```
raspistill -t 2000 -o image.jpg -x IFDO.Artist=Boris -x GPS.GPSAltitude=1235/10
```

Set an emboss style image effect

```
raspistill -t 2000 -o image.jpg -ifx emboss
```

Set the U and V channels of the YUV image to specific values (128:128 produces a greyscale image)

```
raspistill -t 2000 -o image.jpg -cfx 128:128
```

Run preview ONLY for 2s, no saved image.

```
raspistill -t 2000
```

Take timelapse picture, one every 10 seconds for 10 minutes (10 minutes = 600000ms), named image_num_001_today.jpg, image_num_002_today.jpg onwards, with the latest picture also available under the name latest.jpg.

```
raspistill -t 600000 -tl 10000 -o image_num_%03d_today.jpg -l latest.jpg
```

Take a picture and send image data to stdout

```
raspistill -t 2000 -o -
```

Take a picture and send image data to file

```
raspistill -t 2000 -o - > my_file.jpg
```

Run camera forever, taking a picture when Enter is pressed

```
raspistill -t 0 -k -o my_pics%02d.jpg
```

Video Captures

Image size and preview settings are the same as for stills capture. Default size for video recording is 1080p (1920x1080)

Record a 5s clip with default settings (1080p30)

```
raspivid -t 5000 -o video.h264
```

Record a 5s clip at a specified bitrate (3.5MBits/s)

```
raspivid -t 5000 -o video.h264 -b 3500000
```

Record a 5s clip at a specified framerate (5fps)

```
raspivid -t 5000 -o video.h264 -f 5
```

Encode a 5s camera stream and send image data to stdout

```
raspivid -t 5000 -o -
```

Encode a 5s camera stream and send image data to file

```
raspivid -t 5000 -o - > my_file.h264
```

Shell Error Codes

The applications described here will return a standard error code to the shell on completion. Possible error codes are :

EX_OK	0	Application ran successfully.
EX_USAGE	64	Bad command line parameter
EX_SOFTWARE	70	Software or camera error
	130	Application terminated by ctrl-C.