RaspiCam Documentation

July 2013

This document describes the use of the three Raspberry Pi camera applications as of July 2013.

There are three applications provided: `raspistill`, `raspivid` and `raspistillyuv`. Both `raspistill` and `raspistillyuv` are very similar and are intended for capturing images, while `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

Please note that camera modules 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 facing away from the PCB, and on the Raspberry Pi it should be facing 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. You can watch a video showing you how to attach the connectors at www.raspberrypi.org/archives/3890 (scroll down for the video).

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 an 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, while displaying 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` and `sudo apt-get upgrade` been run?
- Has `raspi-config` been run and the camera enabled?

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 have a defect (most likely because it has suffered static shock). Try posting on the Raspberry Pi forum in the camera board 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 stabilization

In video mode only, turn on video stabilization.

`--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:

off	
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 optimized for snowy scenery
beach	Select setting optimized for beach
verylong	Select setting for long exposures
fixedfps	Constrain fps to a fixed value
antishake	Antishake mode
fireworks	Select setting optimized for fireworks

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

`--awb, -awb` Set automatic white balance (AWB)

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

`--imxfx, -ifx` Set image effect

none	No effect
negative	Produces a negative image
solarise	Solarise the image
whiteboard	Whiteboard effect
blackboard	Blackboard effect
sketch	Sketch-style effect
denoise	Denoise the image
emboss	Embossed effect
oilpaint	Oil paint-style effect
hatch	Cross-hatch sketch style
gpen	Graphite sketch style
pastel	Pastel effect
watercolour	Watercolour effect
film	Grainy film 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
```

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.

`--verbose, -v`	Output verbose information during run

Outputs debugging/information messages during the program run.

`--timeout, -t`	Time before capture and shut 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. For example:

```
-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 four-digit number with leading zeros added to pad to the required number of digits. So, for example, %08d would result in an eight-digit number.

`--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.

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

This options cycles through range of camera options, and 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 as 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 encoding a file.

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

Allows the insertion of specific EXIF tags into 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, aISOSpeedRatings, 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.

`--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.

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 made 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, 1080p a high quality bitrate would be 15Mbits/s or more.

`--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 capture and shut 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 five seconds. Setting 0 will mean the application will run continuously until stopped with Ctrl-C.

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

This option 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. H.264 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.

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 two seconds (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 Mooncake, 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=Mooncake -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 two seconds, no saved image.

```
raspistill -t 2000
```

Take timelapse picture, one every 10 seconds for 10 minutes (10 minutes = 600000ms), named image_number_1_today.jpg, image_number_2_today.jpg onwards.

```
raspistill -t 600000 -tl 10000 -o
image_num_%d_today.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
```

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
```