RPi Cam

Intro

This document describes RPiCam_Web_Interface software. It was originally developed by sylvan Melchior (https://github.com/silvanmelchior/RPi_Cam_Web_Interface) This references a large thread on the raspberry forums and a wiki.

It was popular as it was easy to install and gave high quality recordings exploiting the capabilities of the native HD Raspberry Pi camera.

I have forked the original code and extended it in a number of ways. The code is available at https://github.com/roberttidey/RPi_Cam_Web_Interface

Installation instructions will not be repeated here as they are identical to the original as described on the wiki.

Extended functionality

- **Expanded preview / download screen with thumbnails and selective commands.**
- Automated scheduling of modes of operation and capture commands referenced to sun-rise and sun-set times.
- Access to motion settings.
- Splitting of time lapse recordings from image and video.
- Persisting changes to raspimjpeg settings made from the web interface.

Preview
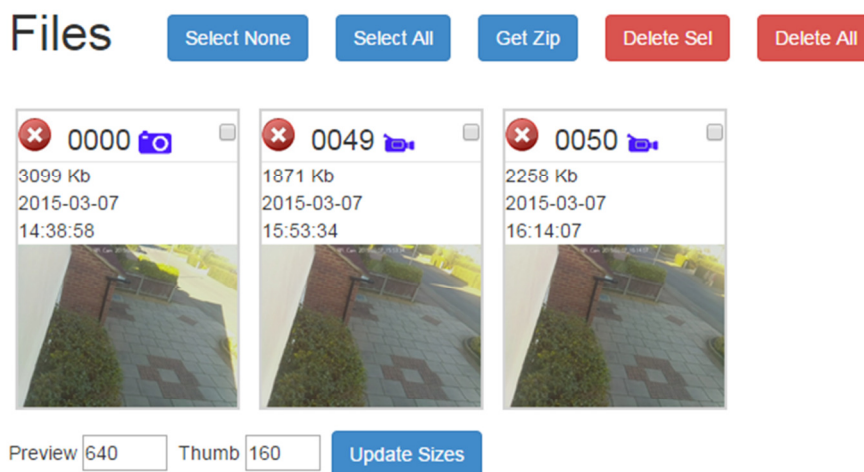
Preview is now a set of thumbnails giving basic characteristics of each recording.

Clicking on an image brings it up in a larger size and videos can be played.

Each image has a select checkbox together with Select All/None. This allows captures to be quickly deleted, or all selections may be downloaded in 1 zip.

Size of the main preview and the size of the thumbnail can be updated via the controls at the bottom. These take immediate effect even for previous captures.

Time lapse recordings show up as a single icon with a thumbnail from first image. Deleting the time lapse deletes all images associated with it.

If a timelapse is clicked to bring up a preview then a convert button is shown and this may be then processed to video with the command shown. The command may be edited with different parameters and these will be saved for nest usage. Do not change the input selector i_%05d.jpg. The conversion can take a long time but runs in the background.

Scheduling

The same program file schedule.php is used both as the web facing settings interface and the run-time daemon which does the automation. All the settings used are contained within the schedule.json config file.

Parameter	Value
scheduleLog	scheduleLog.txt
Fifo_In	/var/www/FIFO1
Fifo_Out	/var/www/FIFO
Cmd_Poll	0.04
Mode_Poll	10
Max_Capture	30
Latitude	51.75
Longtitude	-0.36
GMTOffset	0
DawnStart_Minutes	-180
DayStart_Minutes	0
DayEnd_Minutes	0
DuskEnd_Minutes	180
AllDay	Use just day time settings ☐

Time Offset: 0	Sunrise: 06:04	Sunset: 18:14	Current: 16:33	Period: Day
Type	**Night**	**Dawn**	**Day**	**Dusk**
Commands_On			ca 1	
Commands_Off			ca 0	
Modes	md 0;em night	md 0;em night	md 0;em auto;md 1	md 0;em night

Command reference

The Scheduler will normally be started on boot up and can be left running all the time. A Stop button shows when the background scheduling program is running and turns into a start button if it is not. . Normally it should just be left running. Note that if it is stopped then motion start stop triggers will not be actioned as they pass through the scheduler.

Save Settings saves the settings to schedule.json. Changes will be passed to the background program and take immediate effect. These settings may be backed up and restored.

- scheduleLog is the file where scheduling activity is recorded. The contents are displayed when the Show Log button is used and the log may be cleared from there.
- Fifo_In defines the named pipe that schedule monitors. It should be the same as where motion sends its commands. Fifo_Out is where scheduler sends its commands to raspimjpg Do not change these without good reason. Other configs are relying on these and would need to be changed as well.
- Cmd_Poll is how often the scheduler checks the Fifo_In for incoming commands. It should be kept quite short to avoid unnecessary delays.
- Mode_Poll is how often the scheduler checks for changes between the 4 main daily periods. Setting it to 10 means there might be a 10 second delay in determining when day starts for example.
- Max_Capture determines the maximum capture period. If motion sends a start command and doesn't send a stop command then the scheduler will automatically stop the capture after this interval. This can be used to make all recordings this length by configuring motion never to send stop commands. The scheduler will then always time out and stop the recording.
- Latitude, and Longtitude define where the camera is and allow the sunrise and sunset times to be calculated throughout the year.
- GMTOffset adjusts for sunrise / sunset calculation. Set in hours or TimeZone string
- DawnStart_Minutes through to DuskEnd_Minutes divide the day up into 4 periods based on sunrise and sunset. Dawn starts at sunrise + DawnStart_Minutes so would normally be negative to make Dawn start before sunrise. Similarly for the other 3 numbers. Night is from DuskEnd to DawnStart the next day.
- AllDay – If unticked then commands on /off and mode changes between periods will vary according to the daily schedule set up below. If ticked then the Day settings determine the operation for the whole 24 hour period and the other columns have no effect.

The period table allows mode changes and start and end commands to be differen tin each of the 4 daily periods.

- Commands_On are used to start captures. If left blank (e.g. at night) then no capture happens.
- Commands_Off are used to end captures.
- Modes is used to send in commands at the start of each daily period. So, for example they may be used to control motion detection and change camera settings. Changing to night mode for example extends the usefulness of the camera in dusk and dawn periods.

Note the scheduler is calculating the day periods based on local time conditions. The raspberry pi should have the appropriate time zone set in raspi-config. This may be checked by issuing a date +%R command line which should show local time. It is also displayed on the schedule settings page.

A commands reference table is included.

Motion

| | Show All | Save Settings | Backup | Restore |

framerate	2
minimum_frame_time	0
netcam_url	http://localhost:6655/cam_pic.php
netcam_userpass	password
switchfilter	off
threshold	1500
threshold_tune	off
noise_level	80
noise_tune	on
despeckle	EedDl
area_detect	(null)
mask_file	/home/pi/RPi_Cam_Web_Interface/motionmask.pgm
smart_mask_speed	0
lightswitch	0
minimum_motion_frames	1
gap	3
on_event_start	echo -n '1' >/var/www/FIFO1
on_event_end	echo -n '0' >/var/www/FIFO1
on_motion_detected	(null)
on_area_detected	(null)

The motion screen just gives access to the motion config settings. Not all settings are relevant here. A filtered list is shown and the full list can be accessed if required.

Settings can be saved backed up and restored. Saving does tell motion to start using the new settings.

Time Lapse
There is no separate screen for this but the behaviour is a bit different from the original.

First there is an additional config item in the raspimjpeg settings that allows the destination and naming of the time lapse recordings to be set differently to image and video recordings. So if time lapses are needed in a different folder then this is now possible.

The naming of time lapse recordings has an additional field after the normal image count and before the date. This is the count within a single lapse sequence. It restarts at 0 for each separate time lapse operation.

If time lapses are left being stored in the media folder then they won't show up in the current preview screen.

A thumbnail config parameter allows for control of thumbnail generation separately for images, video and timelapses. By default it is on for images, videos and timelapses. If turned on for time-lapse then one thumbnail will be generated at the time the lapse sequence is started.

Raspimjpeg settings
There are a few new raspimjpeg config values.

- lapse_path controls where time lapse images are stored and named (/var/www/media/tl_%i_%t_%Y%M%D_%h%m%s.jpg)
- thumb_gen controls whether video(v), image(i), and lapse(t) thumbnails are generated. (vit)
- user_config controls where to look for a secondary user config file. The format of this file is identical and can contain any values. (/var/www/uconfig)
- media_path sets the root to hold thumbnails and captures. Captures can be put in subfolders below this if defined in their naming rules (/var/www/media)
- subdir_char this character is used to flatten the subfolder path for thumbnail storage. (@)

Architecture

The overall functionality is quite complex but centres around the raspimjpeg process which accesses the camera data. The following diagram shows the major components. Blue lines indicate data flows. Red lines indicate control.
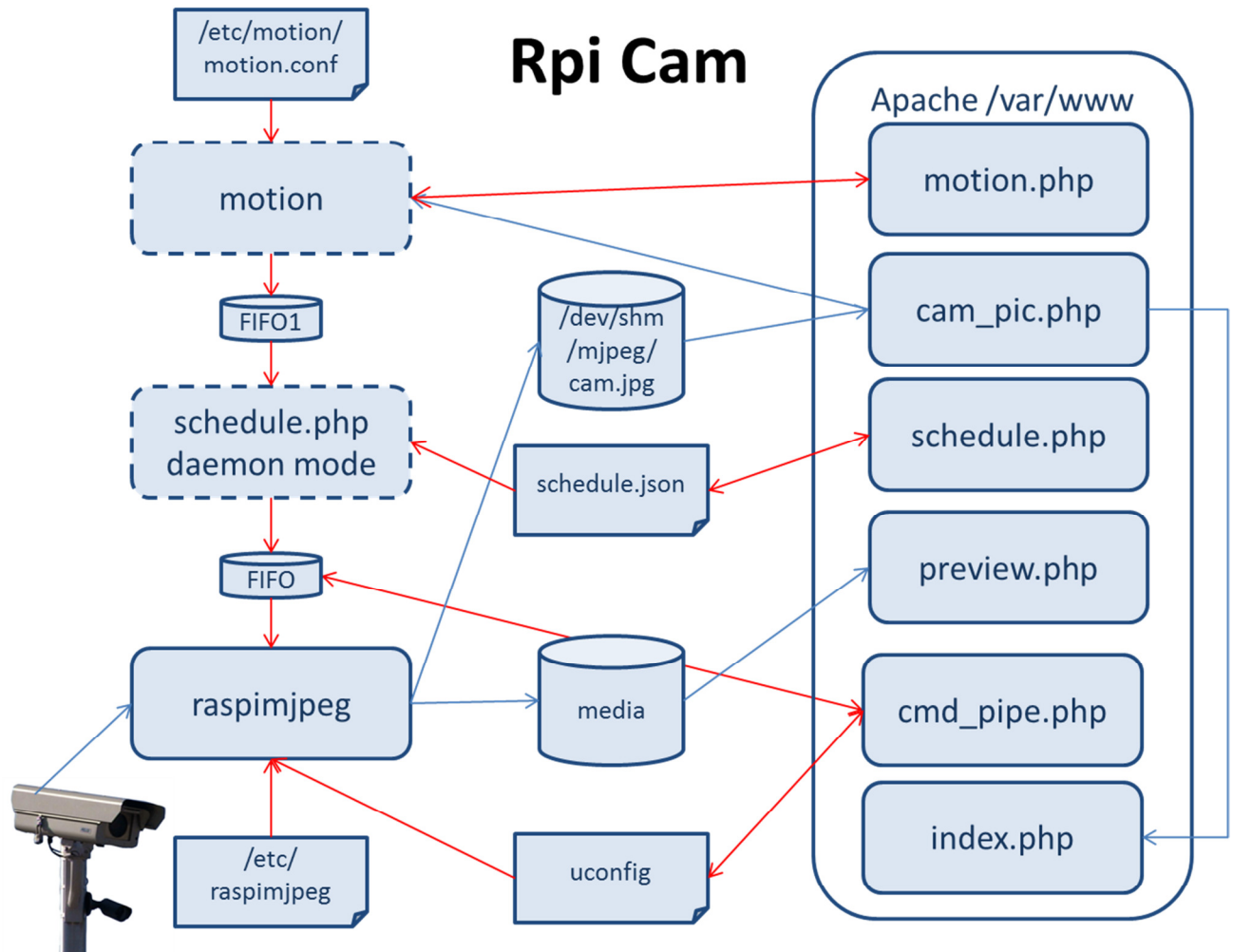
Data flow

In normal monitoring mode raspimjpeg makes a connection to the camera (MMAL) and generates a continuous stream of preview jpeg captures in the /dev/shm/mjpeg directory all with the same name cam.jpg. This folder is RAM memory based so does not strain the SD card memory.

These captures may be accessed via URL which uses the cam_pic.php process on the Apache Web server to return the latest one. Two processes use this. First, if a browser has logged into the web server then the main page (index.php) will continuously refresh a window to give effectively a moving representation of the camera output. Second, if motion detection is active then the motion process also accesses these images in order to analyse differences between successive frames and detect motion.

If raspimjpeg is put into a capture mode (described below) then the flow of preview images is maintained but an extra recording is made of either a single image, a time lapse sequence of single images, or a full video recording which can be at any format the camera can support including HD

normal frame rates. This data is stored in the media folder. For the images and tile lapse images these are stored directly in jpg format. The video is initially stored as a raw h264 stream from the camera but can be automatically formatted into mp4 when the recording ends. The boxing mode (MP4Box) controls this. Three options are provided; false leaves the files in raw h264, true converts them inline but no further recording is possible till this completes, background spawns a separate process to do the boxing operation.

When raspimjpeg initiates any of these sequences it also grabs the current cam.jpg from the preview stream and stores it in the media folder with a thumbnail name tied to the captured data. These thumbnails are used by the preview_php process to give a representation of each capture when the download button is pressed.

Control flow

Raspimjpeg can be controlled by sending in commands into a named FIFO pipe in the /var/www folder.

The commands give access to most of the camera settings plus stopping and starting the capture processes. Commands can come from effectively 3 sources but one is directed through a secondary process for scheduling purposes.

Commands can come from the web browser via the cmd_pipe.php web page. This is used to allow changing the camera settings and manually starting and stopping captures.

Commands can come from the scheduling process (scheduler.php – daemon) which can be used to change various modes, camera settings at different times based on sunrise and sunset.

Commands can also come from motion and these are used to start and stop captures based on motion detection. In the original scheme these commands went straight to raspimjpeg. In the modified scheme they are sent through the scheduling process daemon so that it may change the nature of stop and start based on the daily periods. TO achieve this motion now sends its commands to a secondary named FIFO pipe (FIFO1). The scheduler is monitoring this for motion start / stops and then sends in translated commands to the main raspimjpeg process.

Configuration scheme

A number of configuration files control how the overall system operates. The web pages give browser access to some of these.

raspimjpeg file in /etc is read whenever the raspimjpeg process starts up including if it is stopped and started from the browser. It contains basic paths and information to allow raspimjpeg to do its job plus the camera settings that will be used by default. This file is not accessible from the browser. It should be regarded like a factory default file. Some settings could be changed to suit particular circumstances but this must be dome manually by editing the /etc/raspimjpeg file itself.

uconfig file in the /var/www folder is used to hold any changes from the factory defaults applied from the web browser. To start with it doesn't exist but new values will be added if changes are made from the browser. Changes apply directly because the cmd_ipipe process has fed those into

the raspimjpeg process, but they are also now written to the uconfig file. When raspimjpeg starts it reads the factory defaults file first and then overwrites any settings that are in the uconfig file. You can also clear the uconfig file from the web browser to effectively return to factory settings. This doesn't hold real-time commands like capture start/end commands or motion enable.

motion.conf in the /etc/motion folder is read by motion to determine its operating characteristics. As motion is being used here in a failry simple mode of motion detection then many of its parameters are irrelevant. The primary ones of interest are those setting the motion detection characteristics like mask files, thresholds, noise levels. motion provides a web api to view and edit these settings and this is used by the motion.php page to show and allow altering the settings.

schedule.json in the /var/www folder is used by the scheduling process to determine the characteristics of the automation. It is read and edited using the browser based schedule.php. It is also read by the same file running in command line mode as a background daemon. If changes are made then the scheduler daemon must be started and stopped via theweb page to allow it to see the new settings.

Custom Annotation and flickering

RPiCam supports V2 and V3 annotation. V2 provides basic support and allows either transparent or black background to the annotation. V3 has more flexibility allowing text size, colour and background colour to be set.

Annotation version :	Select option... ▾
Annotation size v3 (0-60):	[ ] OK
Custom text color v3:	Disabled ▾ y:u:v = [ ]:[ ]:[ ] OK
Custom background color v3:	Disabled ▾ y:u:v = [ ]:[ ]:[ ] OK

Some systems have suffered from flickering annotation size. This is a problem with the camera drivers contained in the /boot/start_x.elf file. A fixed version is available at https://github.com/6by9/RPiTest/tree/master/annotate

This also contains a working v3 support system. Do not use v3 set up without updating the start_x.elf. This will become incorporated into the standard raspbian at some point.

raspimjpeg naming

Currently raspimjpeg uses a name for captures and annotation that are controlled by the settings in the config files. These have a number of %X substitution parameters which will be filled in as follows. X must be one of the the following characters. They can be put in any order and repeated if required for a maximum total of 16 substitutions. Any other characters are passed through, but do not use any 'illegal' filename characters or the subdir_char (@). Extra path separators may be put in below the media folder level.

%%	Single %
%Y	4 digit year
%y	2 digit year
%M	2 digit month

%D	2 digit day
%h	2 digit hour
%m	2 digit minute
%s	2 digit second
%v	4 digit video recording IndexNumber
%i	4 digit image recording IndexNumber (also used for time lapse batch)
%t	4 digit LapseIndex (starts at 1 for each capture)

So a config image_path of /var/www/media/im_%i_%Y%M%D_%h%m%s.jpg

Will give a name like im_0005_20150309_093057.jpg

Lapse Index starts at 1 for a particular Time Lapse set and increments.

Thumbnails are named with the base capture name appended with ' . [vit]IndexNumber.th.jpg' where [vit] is a single character for video, image and lapse recordings. This allows cross referencing back to the real capture files independent of the actual base name.
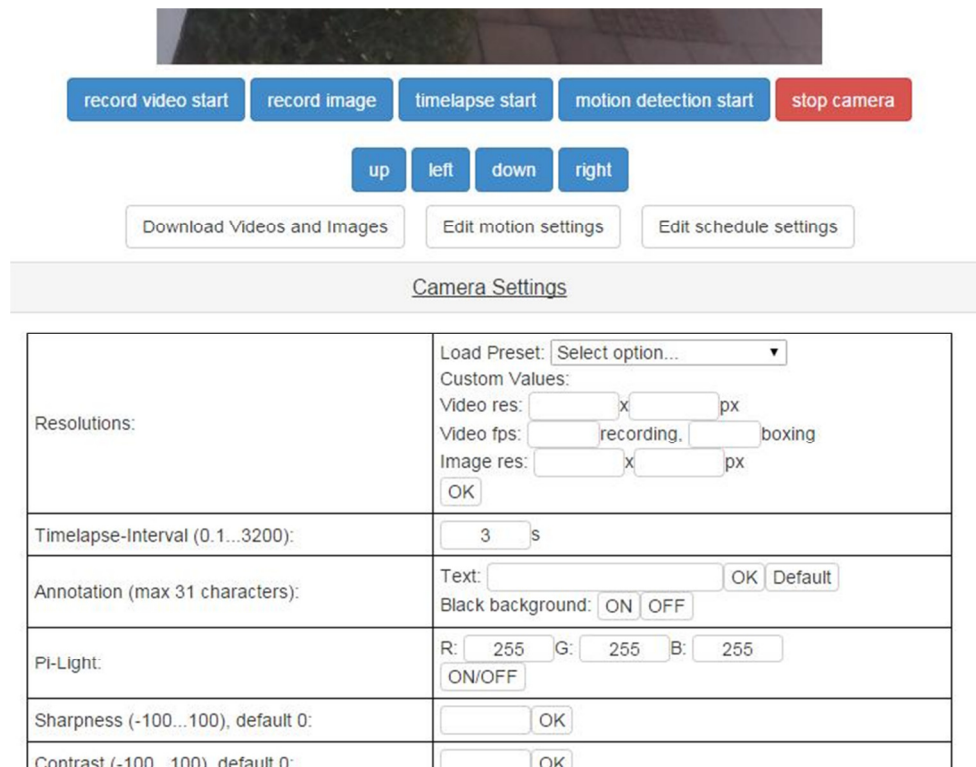
The %i IndexNumber is shared between thumbnails and images. Vidoes have their own IndexNumber %v. The starting IndexNumbers used to be fed in via command line switches after a start script calculated them. This has been replaced by raspimjpeg doing a scan itself at start up and looking for the highest number in each category.

Style override selector

All the pages try to load a style sheet called extrastyle.css from the css folder. This allows particular styles to be overridden, ranging from small adjustments through to complete colour scheme changes. A selector in the system settings of the main page can move preset versions of css override styles into extrastyle. The preset should be named with a es_ StyleName.css and will then show up in the selector.

Pipan / Pilight support

The web pages do have the basic code for Pipan and Pilight controls built in but by default they are not visible. To enable Pipan controls to show up a file called pipan_on must be present in the www folder. Similarly for Pilight a file called pilight_on must be present. It doesn't matter what is in these files. The install includes two files called pipan_off and pilight_off so these just need to be renamed according to requirements. If they are both on then the front screen has extra controls and camera settings like.

NOTE: This is only the web control side. Refer to the wiki for details on how to install the device control side.

raspimjpeg Pipe command reference

The following commands are supported by raspimjpeg when received from the pipe. Many are equivalents of values in the config file. Others are associated with several parameters.

They are sent in as a serial stream as a 2 character command, space, and space separated parameters. There must only be 1 space and the length of parameters must be as shown

cmd	parameters	Description
an	text	set annotation
ab	0/1	annotation background
px	AAAA BBBB CC DD EEEE FFFF	set video+img resolution video = AxB px, C fps, boxed with D fps, image = ExF px)
av	number	Set annotation version 2 or 3
as	number	Set text size 0-99
at	E YYY UUU VVV	Set Text colour E (0/1 enable) Colour as Y:U:V
ac	E YYY UUU VVV	Set background colour E (0/1 enable) Colour as Y:U:V
sh	number	set sharpness (range: [-100;100]; default: 0)
co	number	set contrast (range: [-100;100]; default: 0)
br	number	set brightness (range: [0;100]; default: 50)
sa	number	set saturation (range: [-100;100]; default: 0)
is	number	set ISO (range: [100;800]; default: 0=auto)
vs	number	0/1 turn off/on video stabilisation
ec	number	set exposure compensation (range: [-10;10]; default: 0)
em	keyword	set exposure mode (range: [off/auto/night/nightpreview/backlight/spotlight/sports/snow/beach/verylong/fixedfps/antishake/fireworks]; default: auto)
wb	keyword	set white balance (range: [off/auto/sun/cloudy/shade/tungsten/fluorescent/incandescent/flash/horizon]; default: auto)
mm	keyword	set metering mode (range: [average/spot/backlit/matrix]; default: average)
ie	keyword	set image effect (range: [none/negative/solarise/posterize/whiteboard/blackboard/sketch/denoise/emboss/oilpaint/hatch/gpen/pastel/watercolour/film/blur/saturation/colourswap/washedout/posterise/colourpoint/colourbalance/cartoon]; default: none)
ce	A BB CC	set colour effect (A=enable/disable, effect = B:C)
ro	number	set rotation (range: [0/90/180/270]; default: 0)
fl	number	set flip (range: [0;3]; default: 0)
ri	AAAAA BBBBB CCCCC DDDDD	set sensor region (AAAAA BBBBB CCCCC DDDDD, x=A, y=B, w=C, h=D)
ss	number	Set shutter speed
qu	number	set output image quality (range: [0;100]; default: 85)
bi	number	set output video bitrate (range: [0;25000000]; default: 17000000)
bo	number	Set mp4 boxing mode 0=off,1=inline,2=background
rl	0/1	0/1 disable / enable raw layer
ru	0/1	0/1 halt/restart RaspiMJPEG and release camera
md	0/1	0/1 stop/start motion detection
ca	0/1	0/1 stop/start video capture
im	0/1	capture image
tl	0/n	start timelapse, parameter is time between images n * 1/10 seconds.0 is stop