Final Graduate Project:

Capture Module for IP Elphel 353 cameras. Evaluation of performance.

Studies: Graduate in Audiovisual Systems Engineering.

Autor: Ángel Torres Moreira

Director: Verónica Vilaplana Biesler.

Advisors: Albert Gil Moreno, Josep Ramon Casas, Josep Pujal.

Escola de Enginyeria de Terrassa, 2013

ABSTRACT

This project's main objective is to evaluate the performance of the capture made by Elphel 353 camera. The assessment will serve as a starting point for the integration of these cameras in the new Smart room built in the Department of Signal Theory and Communications ,UPC (Universitat Politècnica de Catalunya).

First the most important properties of the camera and the tools provided by the camera are described, Next, we study how to use these tools to get images in two capture modes, online and offline. Once we know how to get images, we define the methods used for evaluation, finally the capture is evaluated and the results and conclusions are presented.

CONTENT TABLE

1. LIST OF FIGURES

2. LIST OF TABLES

3. INTRODUCTION.

After the work made in building a new Smart Room in the Universitat Politècnica de Catalunya (UPC), the need of updating hardware became necessary in order to improve quality of the research made by the Image Processing Group (GPI).

The research group's plan is to place the cameras in the new Smart room in order to improve the work developed in the current room: Multi-View/Multi-Sensor scene capture, analysis and representation, gesture recognition, classification, etc.

In answer to these requirements, after an in-depth study of the market and current resources, the group decided to use the following type of cameras:

Elphel model 353 (Toolkit NC353L-369-5C, +PoE (Power Over Ethernet) Power Supply, +Live USB, +Camera Case - NC353L) [1]

The main goal of this work is to analyze the performance of this camera in the main usage scenarios:

- Capturing a sequence of images from a server and connecting it to a processing application running in real time.
- Recording a sequence of images (save to disk).

Assessing the performance of these working modes, requires an analysis of the performance related to image acquisition at different resolution modes. In particular:

- Drop frames.
- CPU usage in host.
- Network load.
- Hard drive writing capacity usage.

Major objectives and possible problems are described as part of this introduction.

3.1. OBJECTIVES.

- Study of camera's resources.

Initially, a consistent study of camera's properties and capabilities must be done. The manufacturer provides a wiki where the documentation of the camera is discussed, Some projects use the camera in movie production with documentation useful to read.

- Implementation of a software capable of manage needed camera's settings.

The software should be able to change all parameters of the camera.

Configuration parameters such as frames per second, resolution, color mode acquisition, color gains should be available to the user in order to check and change them at will.

Knowing tools inside the camera will help to decide the best way to implement this target.

- Software development to record appropriately from 1 camera

A client application should be developed for recording in online mode and offline mode. The applications can be independent or mixed in an user's choice application.

This software will be the base of the experiments to determine the next objective.

- Definition of the best way to get video from the camera.

According to the researchers requirements two different modes of recording are needed: online and offline..

Offline mode refers to a scenario where the researchers, making a request of capture, receive a sequence of images and these images are saved directly to disk (dumping) without including any kind of processing.

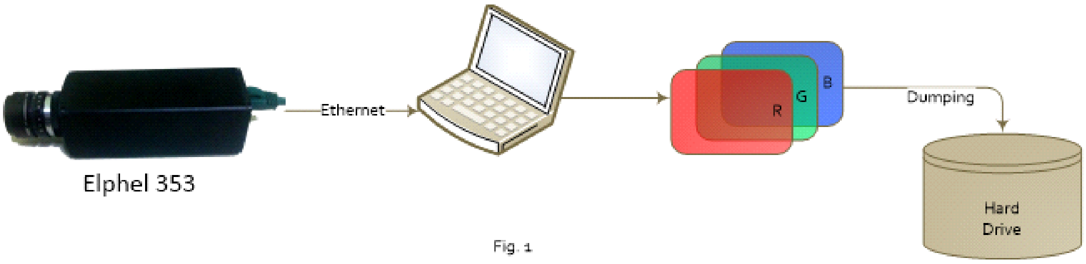


Figure 1: Basic offline recording model

In the offline mode (figure 1), CPU power shouldn't be a problem since any image processing is applied in a second stage, non simultaneously to acquisition. Here the balance between resolution, network load, CPU usage and drop frames must be done given that network load (100 Mbps) is the limitation in this capture mode.

The online scenario means that the researchers ask for a capture and images received are directly available for image processing applications. the result of the acquisition process is a sequence of RGB images contained in a data structure known by the researchers.

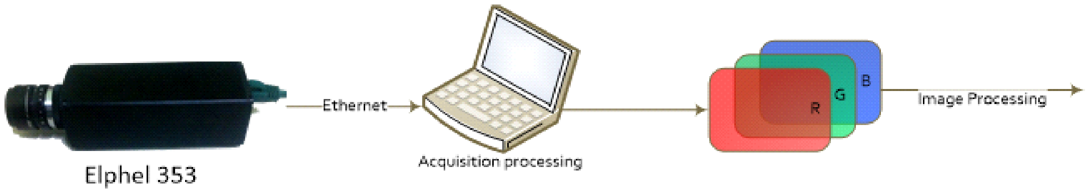


Figure 2: Basic online recording model

In online mode (figure 2), camera's stream should be available to be used in another real time image processing module (face recognition, gesture recognition, classifiers, etc) running simultaneously to acquisition. Getting appropriate output from the camera is necessary, that's why a balance between resolution, network load, CPU usage and drop frames should be done, focusing in obtaining the less CPU use, as this

CPU power should be consumed by image processing, not by the acquisition stage.

4. REVIEW OF THE LITERATURE

Once the goals have been specified, a basic study of camera's properties, ways to change these features and how they affect the outcome is made.

The following subsections describe the stages through which the image captured by the sensor passes up to the transmission. The hardware architecture of the camera expose the path followed by an image, going from capture to transmission and what are the tools to access this information in the stages described.

After knowing the acquisition properties of the camera, it is necessary to know the tools provided to establish a streaming and what their properties are.

Using these tools imply that one should know the methods used by them: compression, standards, new terminology, that's why this section includes a brief description too.

4.1. A GENERAL OVERVIEW OF ELPHEL 353 CAMERA

4.1.1. ELPHEL DESCRIPTION.

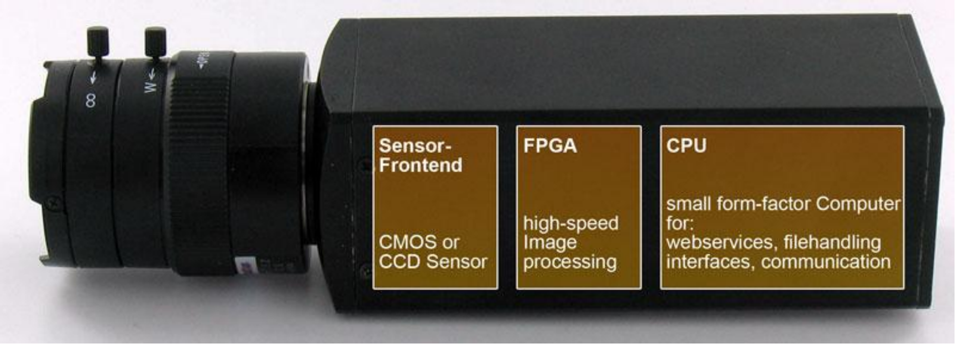

Sensor-Frontend
CMOS or CCD Sensor
FPGA
high-speed Image processing
CPU
small form-factor Computer for:
webservices, filehandling
interfaces, communication

Figure 3: Elphel 353 basic block diagram

Elphel 353 is a highly customizable camera, with an IO-board. A serial port provides access to the root console that simplifies firmware development and a USB port allows connecting additional peripherals. The IO-board also features the external sync/trigger port.

There's a Linux distribution inside de camera, on which tools are developed for ease of setup and the transmission of information through a physical medium connected directly to the camera, or the Ethernet channel.

One of these tools is a php server, which handle configurations that can control all the acquisition parameters of the sensor, as well as the properties of the FPGA used for image processing.

4.1.1.1. Power

- 48V DC with midspan power supply/injector
- or optionally a 12-36V with power injector (non IEEE802.3af compliant)
- or single 3.3V source
- Typical power consumption between 2.4W - 5.8W depending on operation and load.

4.1.2. SENSOR

Elphel 353 model uses Aptina MT9P031: 1/2.5-Inch 5Mp as its CCD sensor [2].

This CMOS active pixel digital image sensor has an active imaging pixel array of 2592H x 1944V. It incorporates sophisticated camera functions on-chip such as windowing, column and row skip mode, and snapshot mode.

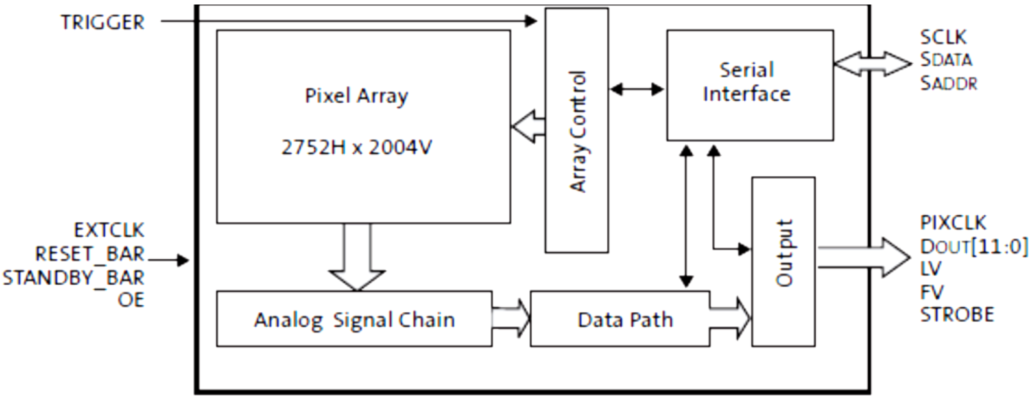


Figure 4: Block diagram of sensor operation

The MT9P031 is a progressive-scan sensor that generates a stream of pixel data at a constant frame rate. It uses an on-chip, phase-locked loop (PLL) to generate all internal clocks from a single master input clock running between 6 and 27 MHz The maximum pixel rate is 96 Mp/s, corresponding to a clock rate of 96 MHz. Figure 4 illustrates a block diagram of the sensor.

MT9P031 pixel array consists of a 2752-column by 2004-row matrix of pixels addressed by column and row. The address (column 0, row 0) represents the upper-right corner of the entire array, looking at the sensor, as shown in Figure 4: Block diagram of sensor operation

The array consists of a 2592-column by 1944-row active region in the center representing the default output image, surrounded by a boundary region (also active), surrounded by a border of dark pixels. The boundary region can be used to avoid edge effects when doing color processing to achieve a 2592 x 1944 result image (figure 5), while the optically black column and rows can be used to monitor the black level.

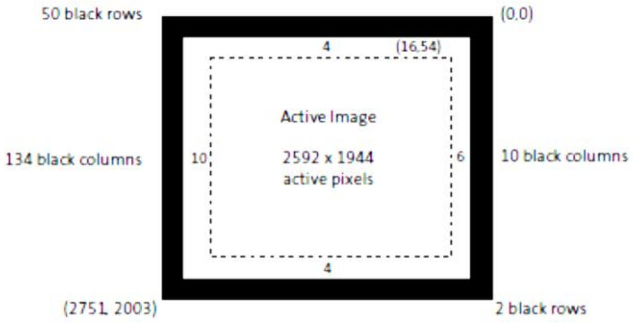


Figure 5: Effective region used by the sensor

4.1.3. CAMERA HARDWARE ARCHITECTURE

The main system board is essentially a universal computer running GNU/Linux [3] with addition of the FPGA and a dedicated video buffer memory. FPGA is used for image processing and compression; it is also used for interfacing sensors and daughter boards. Block diagram of a typical Elphel camera includes camera processor board (10353), CMOS sensor front end (10338), optional I/O extension board (10369) with adapter boards (103691, 103692). Either universal I/O extension board or just smaller sync boards can be used to lock (synchronize) multiple cameras for stereo or panoramic applications.

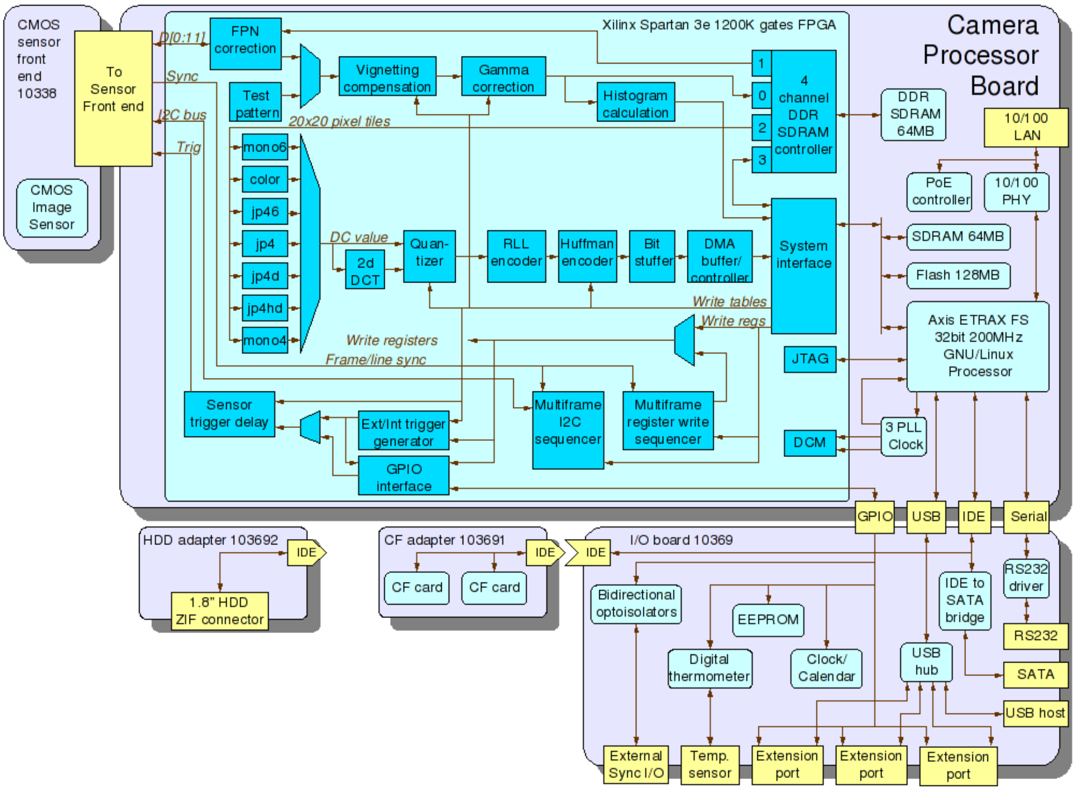


Figure 6: Camera hardware architecture

The figure 9 shows camera main hardware components, I/O ports and internal connections between the boards. It also includes details about the major processing modules implemented in the FPGA.

There are four ways to code in this camera [4].

- Verilog HDL code in the FPGA performs most computationally-intensive tasks in the camera, it handles all of the image processing/compression operations providing compressed bit stream ready to be sent over the network or recorded on the storage media.

- Kernel drivers make up the next overall software layer in the camera (and the lowest one of the code that is running in the CPU). These drivers supplement the standard ones (i.e. network, IDE, USB) and provide software interface to the FPGA modules (i.e. gamma correction, histograms, color converter, compressor) and external devices (like image sensor) connected to the FPGA. As the main camera function is to stream or record video that can run at high frame rate, the related drivers are designed to ease the real time requirements to the application layer software.

- Application layer, consists of standard programs like web, ftp, telnet and SSH servers. There are also camera-specific applications, such as:

  str - unicast/multicast RTSP video server capable of streaming full sensor frame video (plain RTP has a limit of 2040 pixels for image width or height), the stream can be rendered with such media players as MPlayer or VLC.

- Web applications and scripts: This is the highest coding level in the camera, in this layer lays a web server (lighttpd) which is responsible of running PHP efficiently. This technology allows several copies of the PHP to be running ready to serve HTTP requests, interpreter (: 2MB in the camera) does not needed to be re-started for each new one.

There are multiple scripts installed in the camera that are accessible through the lighttpd web server, including those designed to be part of AJAX applications, others just provide access to the camera hardware (like I2C bus to clock/calendar, identification EEPROM, thermometer). Many of the scripts are intended for development - from creating control interfaces to fine-tuning of the driver parameters, monitoring internal data structures and profiling the interrupt service routine. This layer allows the user to create and upload its own applications in both modes: to the camera RAM-disk to safely try them or to the camera flash memory - to stay longer survive power cycles.

4.1.4. COLOR PROCESSING

First the most important features of JPEG encoding which is the core technology in the color treatment of the camera, are explained.

Then this section will outline the ways the system handles color information of the images and which are the options provided by Elphel to make a balance between image quality, in-camera color processing and host-processing

4.1.4.1. JPEG

JPEG [5] is a commonly used method of lossy compression for digital (image). The degree of compression can be adjusted, allowing a selectable tradeoff between storage size and image quality. JPEG typically achieves 10:1 compression with little perceptible loss in image quality.

JPEG uses a lossy form of compression based on the discrete cosine transform (DCT). This mathematical operation converts each frame/field of the video source from the spatial (2D) domain into the frequency domain (aka transform domain.) A perceptual model based loosely on the human psycho visual system discards high-frequency information, i.e. sharp transitions in intensity, and color hue. In the transform domain, the process of reducing information is called quantization. Quantization is a method for optimally reducing a large number scale (with different occurrences of each number) into a smaller one, and the transform-domain is a convenient representation of the image because the high-frequency coefficients, which contribute less to the over picture than other coefficients, are characteristically small-values with high compressibility. The quantized coefficients are then sequenced and losslessly packed into the output bit stream. Nearly all software implementations of JPEG permit user control over the compression-ratio (as well as other optional parameters), allowing the user to trade off picture-quality for smaller file size.

4.1.4.1.1. A COMMON JPEG CODING SCENARIO

The encoding process consists of several steps (figure 10):

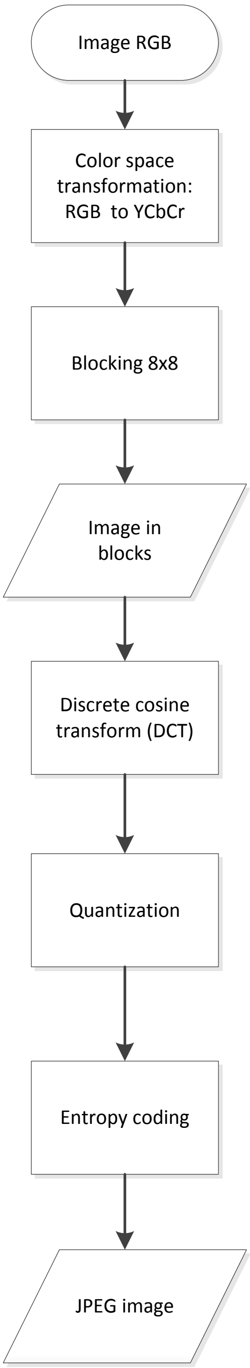


Figure 7: Common JPEG coding.

1. The representation of the colors in the image is converted from RGB to Y’CBCR, consisting of one luma component (Y'), representing brightness, and two chroma components, (CB and CR), representing color.

2. The resolution of the chroma data is reduced, usually by a factor of 2. This reflects the fact that the eye is less sensitive to fine color details than to fine brightness details.

3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y, CB, and CR data undergoes the Discrete Cosine Transform (DCT). A DCT is similar to a Fourier transform in the sense that it produces a kind of spatial frequency spectrum.

4. The amplitudes of the frequency components are quantized. Human vision is much more sensitive to small variations in color or brightness over large areas than to the strength of high-frequency brightness variations. Therefore, the magnitudes of the high-frequency components are stored with a lower accuracy than the low-frequency components. The quality setting of the encoder affects to what extent the resolution of each frequency component is reduced. If an excessively low quality setting is used, the high-frequency components are discarded altogether.

5. The resulting data for all 8×8 blocks is further compressed with a lossless algorithm, a variant of Huffman encoding.

The decoding process reverses these steps, except the quantization because it is irreversible.

4.1.4.1.2. SOME PROPERTIES OF JPEG STANDARD.

- Achieve rate and reconstructed image quality “ with image fidelity classifiable as “very good” to “excellent”
- Be useful for compressing any continuous tone still image, including both gray-scale and color, any color space, and most image sizes
- Have complexity that would allow software implementations on many computing platforms and affordable hardware implementations.
- Adjustable degree of compression, allowing a selectable trade-off between image quality and storage size
- Typical 10 to 1 compression ratio without appreciable loss of quality
- Appropriated for real color images and realistic paintings with smooth variations of tone and color
- Not appropriated for images with sharp contrasts between adjacent pixels

4.1.4.2. ELPHEL COLOR TREATMENT.

The color information of most of images sensors is obtained placing over the pixel sensors of an image sensor, a Bayer filter.

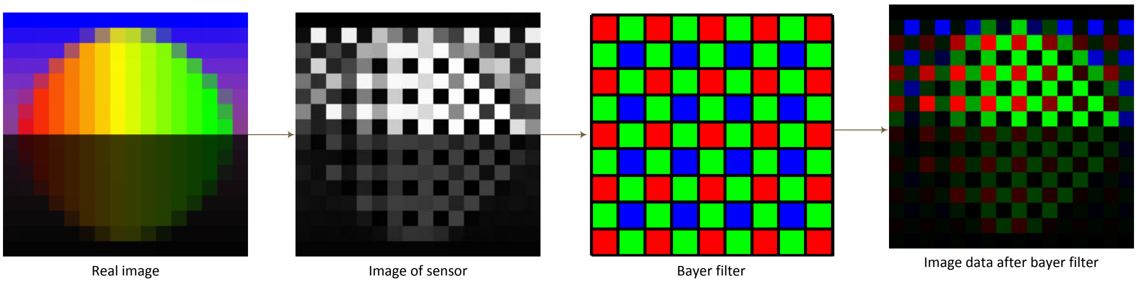

Real image

Image of sensor

Bayer filter

Image data after bayer filter

Figure 8: Image from the sensor to the Bayer filter.

This filtered image can be reconstructed with modern algorithms that can restore missing color components precisely. As this algorithm requires high processing as well as data transfer between FPGA and external buffer memory.

Elphel defines two working color modes:

4.1.4.2.1. COLOR MODE

Elphel camera transforms the image from the Bayer filter to YCbCr (Figure 9: Bayer image to YCbCr 4:2:0) format in order to be ready to Jpeg compression. This transformation split the image in 4 luma and 2 color components with a 4:2:0 chroma subsampling [6]. The Bayer pattern interpolation and RGB-: YCbCr conversion is done in a single step, result values for Y, Cb and Cr are calculated directly from the input Bayer with a 3x3 bilinear interpolation code.

Color is represented as Cb (difference between blue and green) and Cr (difference between red and green), each having half spatial resolution in both vertical and horizontal directions compared to the original pixels.

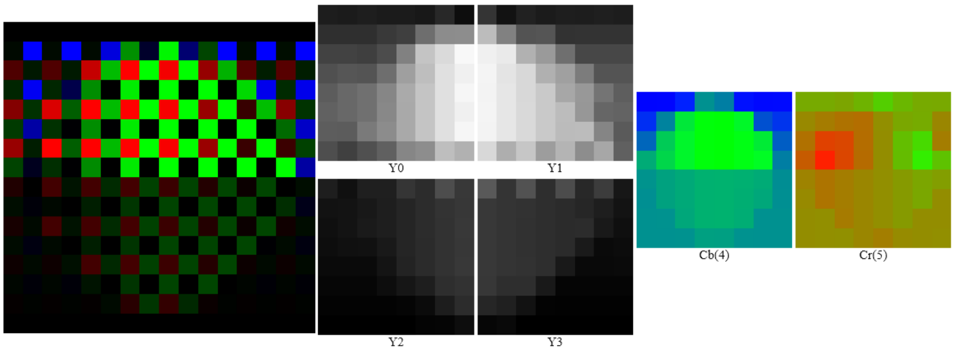
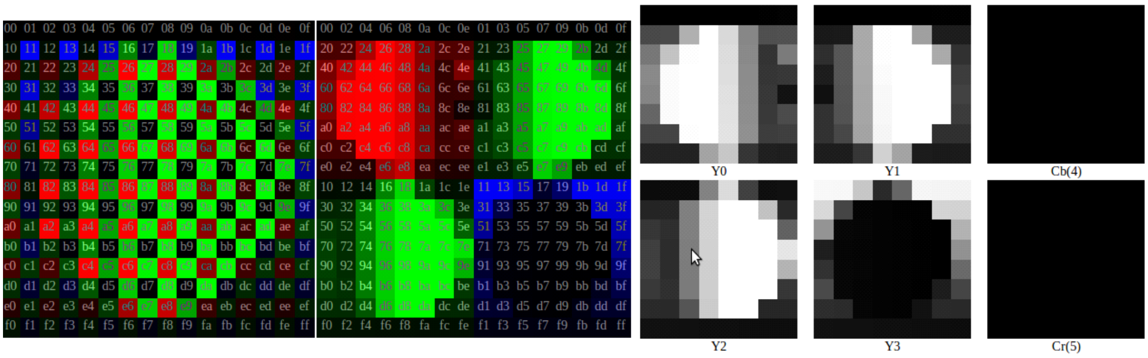


Figure 9: Bayer image to YCbCr 4:2:0

4.1.4.2.2. JP46 MODE

JP46 is a recording mode provided by Elphel, which bypasses the demosaic in the FPGA (bilinear interpolation) provides an image with pixels in each 16x16 macroblock that are rearranged to separate Bayer colors in individual 8x8 blocks.

In Figure 10: Bayer Image rearranged and transformed to YCbCr 4:2:0 , it can be seen the Bayer sensor image re-arranged at first in color components RGGB blocking 16x16 so each color component pixels get into a separate 8x8 pixel block, then transformed to Y, Cb, Cr components for an efficient Jpeg compression.



00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
10	11	12	13	14	15	16	17	18	19	1a	1b	1c	1d	1e	1f
20	21	22	23	24	25	26	27	28	29	2a	2b	2c	2d	2e	2f
30	31	32	33	34	35	36	37	38	39	3a	3b	3c	3d	3e	3f
40	41	42	43	44	45	46	47	48	49	4a	4b	4c	4d	4e	4f
50	51	52	53	54	55	56	57	58	59	5a	5b	5c	5d	5e	5f
60	61	62	63	64	65	66	67	68	69	6a	6b	6c	6d	6e	6f
70	71	72	73	74	75	76	77	78	79	7a	7b	7c	7d	7e	7f
80	81	82	83	84	85	86	87	88	89	8a	8b	8c	8d	8e	8f
90	91	92	93	94	95	96	97	98	99	9a	9b	9c	9d	9e	9f
a0	a1	a2	a3	a4	a5	a6	a7	a8	a9	aa	ab	ac	ad	ae	af
b0	b1	b2	b3	b4	b5	b6	b7	b8	b9	ba	bb	bc	bd	be	bf
c0	c1	c2	c3	c4	c5	c6	c7	c8	c9	ca	cb	cc	cd	ce	cf
d0	d1	d2	d3	d4	d5	d6	d7	d8	d9	da	db	dc	dd	de	df
e0	e1	e2	e3	e4	e5	e6	e7	e8	e9	ea	eb	ec	ed	ee	ef
f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	fa	fb	fc	fd	fe	ff

00	02	04	06	08	0a	0c	0e	01	03	05	07	09	0b	0d	0f
20	22	24	26	28	2a	2c	2e	21	23	25	27	29	2b	2d	2f
40	42	44	46	48	4a	4c	4e	41	43	45	47	49	4b	4d	4f
60	62	64	66	68	6a	6c	6e	61	63	65	67	69	6b	6d	6f
80	82	84	86	88	8a	8c	8e	81	83	85	87	89	8b	8d	8f
a0	a2	a4	a6	a8	aa	ac	ae	a1	a3	a5	a7	a9	ab	ad	af
c0	c2	c4	c6	c8	ca	cc	ce	c1	c3	c5	c7	c9	cb	cd	cf
e0	e2	e4	e6	e8	ea	ec	ee	e1	e3	e5	e7	e9	eb	ed	ef
10	12	14	16	18	1a	1c	1e	11	13	15	17	19	1b	1d	1f
30	32	34	36	38	3a	3c	3e	31	33	35	37	39	3b	3d	3f
50	52	54	56	58	5a	5c	5e	51	53	55	57	59	5b	5d	5f
70	72	74	76	78	7a	7c	7e	71	73	75	77	79	7b	7d	7f
90	92	94	96	98	9a	9c	9e	91	93	95	97	99	9b	9d	9f
b0	b2	b4	b6	b8	ba	bc	be	b1	b3	b5	b7	b9	bb	bd	bf
d0	d2	d4	d6	d8	da	dc	de	d1	d3	d5	d7	d9	db	dd	df
f0	f2	f4	f6	f8	fa	fc	fe	f1	f3	f5	f7	f9	fb	fd	ff

Figure 10: Bayer Image rearranged and transformed to YCbCr 4:2:0

Recording with this mode, optimizes JPEG encoding because rearranging macroblocks per color retain more information of high frequencies even if there are no sharp gradients in the image. For a colored object there will be a repetitive pattern with maximal frequency - odd and even pixels will have different values.

4.1.5. RESOLUTION CHANGING MODES

Aptina MT9P031: 1/2.5-Inch 5Mp (CMOS inside the camera) [2] is able to change the resolution of acquisition by two methods: Windowing and Skipping/binning.

4.1.5.1. Windowing.

A window of interest (WOI) from total image sensor can be specified. This window of interest has 4 necessary values:

Window start width
Window start height
Window width
Window height

This values define where exactly the full sensor image is trimmed, defining a starting point and a width and height value.
The resolution changed in this way could be exploited if only one region of the image is needed instead of all sensor visual field.

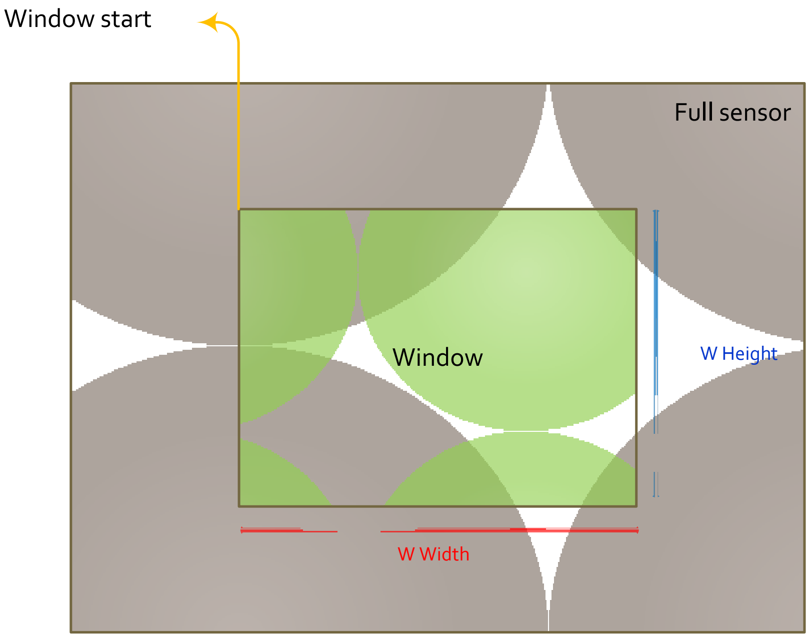
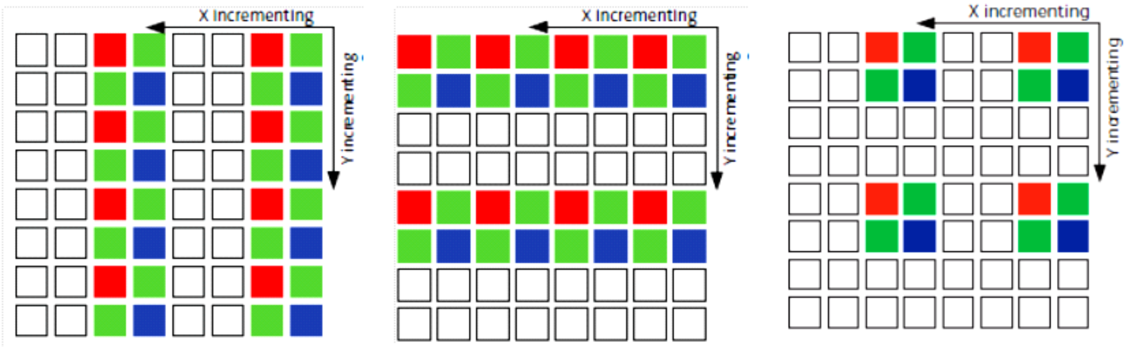


Figure 11: Windowing scheme

4.1.5.2. Skipping

Also called "Downsampling", means that a certain amount of photosites is not read out but skipped (horizontally, vertically or in both axes). This reduces resolution of the resulting image but introduces downsampling artifacts. This is an on-sensor-feature.

Skipping reduces resolution by using only selected pixels from the FOV(Field of Vision) in the output image. In skip mode, entire rows and columns of pixels are not sampled, resulting in a lower resolution output image. A skip 2X mode skips one Bayer pair of pixels for every pair output. Skip 3X skips two pairs for each one pair output. Rows and columns are always read out in pairs. If skip 2X mode is enabled with otherwise default sensor settings, the columns in the output image correspond to the pixel array columns 16, 17,20, 21, 24, 25...



Column skip 2x

Row Skip 2x

Column skip 2x, Row Skip 2x

Figure 12: Skipping working models

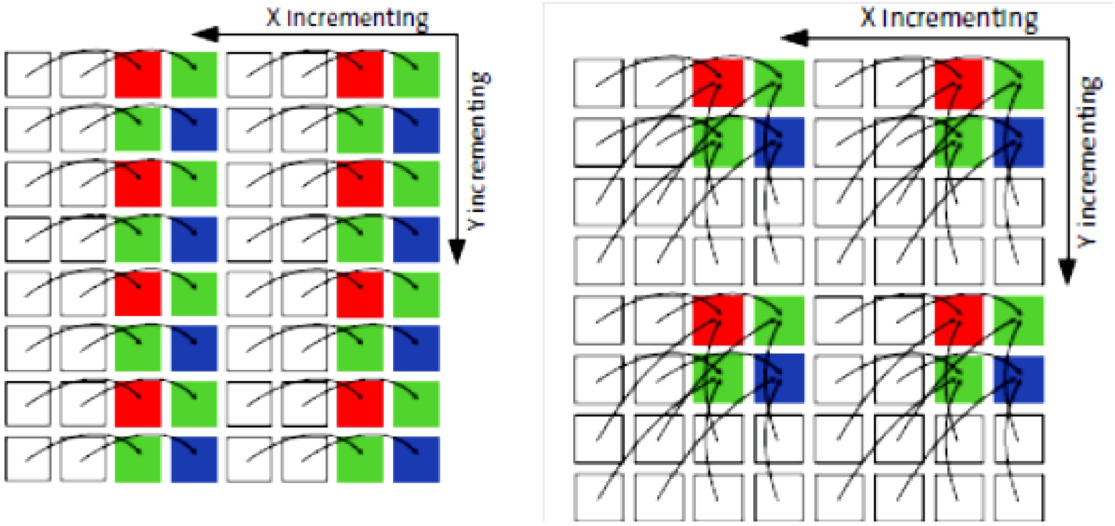
4.1.5.3. Binning

Binning reduces resolution by combining adjacent same-color imager pixels to produce one output pixel. All of the pixels in the FOV contribute to the output image in binning mode.

This can result in a more pleasing output image with reduced downsampling artifacts. It also improves low-light performance. For columns, the combination step can be either an averaging or summing operation. Depending on lighting conditions, one or the other may be desirable. In low-light conditions, summing produces a gain roughly equivalent to the column bin factor. Column summing may be enabled by setting Column_Sum.

Binning works in conjunction with skipping. Pixels that would be skipped because of the Column_Skip and Row_Skip settings can be averaged instead by setting Column_Bin and Row_Bin to the number of neighbor pixels to be averaged with each output pixel.

For example, to set bin 2x mode, set Column_Skip and Column_Bin to 1. Additionally, Column_Start must be a multiple of (2 * (Column_Bin + 1)) and Row_Start must be a multiple of (2 * (Row_Bin + 1)).
Only certain combinations of binning and skipping are allowed.



Column 2x bin

Column Bin 2X, Row Bin 2X)

Figure 13: Binning model scheme.

4.1.5.4. Pixel Downsampling and Binning in Elphel Cameras

Elphel includes a tool to modify downsampling, from 1-8, i.e.: If the defined window is 2592x1944 and downsampling property is 2, the resolution of the acquired image will be 1296x972.

For Binning, two methods are defined:

The "normal" Binning mode averages the charges from photosites thus reducing noise in the image but not altering light sensitivity.

There is an alternative mode that sums up the charges and therefore makes the binned pixel more light sensitive.

When using Downsampling/binning one can:

- Lower the resolution of an image while using the same sensor area
- Achieve higher FPS for the same resulting resolution
- Decrease image noise (averaging) or increase light sensitivity (summing) .

4.2. OBTAINING VIDEO FROM THE CAMERA

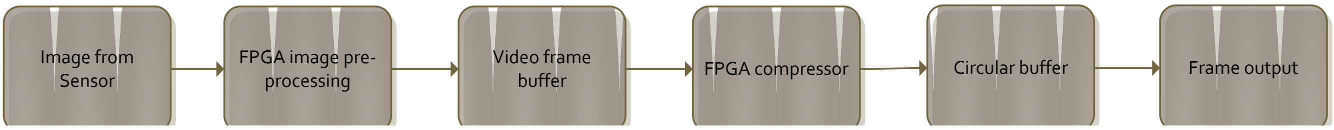


Figure 14: Pipeline of image acquisition

Summarizing the previous section, the path followed by the image can be reduced to the pipeline in figure 14.

The image captured and processed by the camera (frame output) has to be sent by the transmission channels mentioned (Sata / IDE interface, 10/100 Mbps LAN interface). This block presents the possible options to transfer the image.

Before defining the tools provided by the camera, one should know the technology related to obtaining video from the camera.

4.2.1. MOTION JPEG (MJPEG)

The video format used by the camera to stream video, is a video format where each video frame or interlaced field of a digital video sequence is compressed separately as a JPEG image. MJPEG is widely used in digital cameras, ip cameras, and webcams.

MJPEG has only intra-frame compression unlike most of modern inter-frame video formats (MPEG-4, H.264). This becomes an advantage because MJPEG needs lower processing and memory requirements on hardware devices.

Advantages

- Simple implementation.
- Good response in high motion in video stream against high quality loses in inter-frame compression.
- Wide market acceptance, as it is an “old” standard.
- Hardware requirements are minimal.

Disadvantages.

- JPEG is inefficient in terms of storage. Better entropy coding is applied in new formats like H.264/Mpeg-4.
- Not defined a unique recognized form of “MotionJpeg” . Incompatibility.

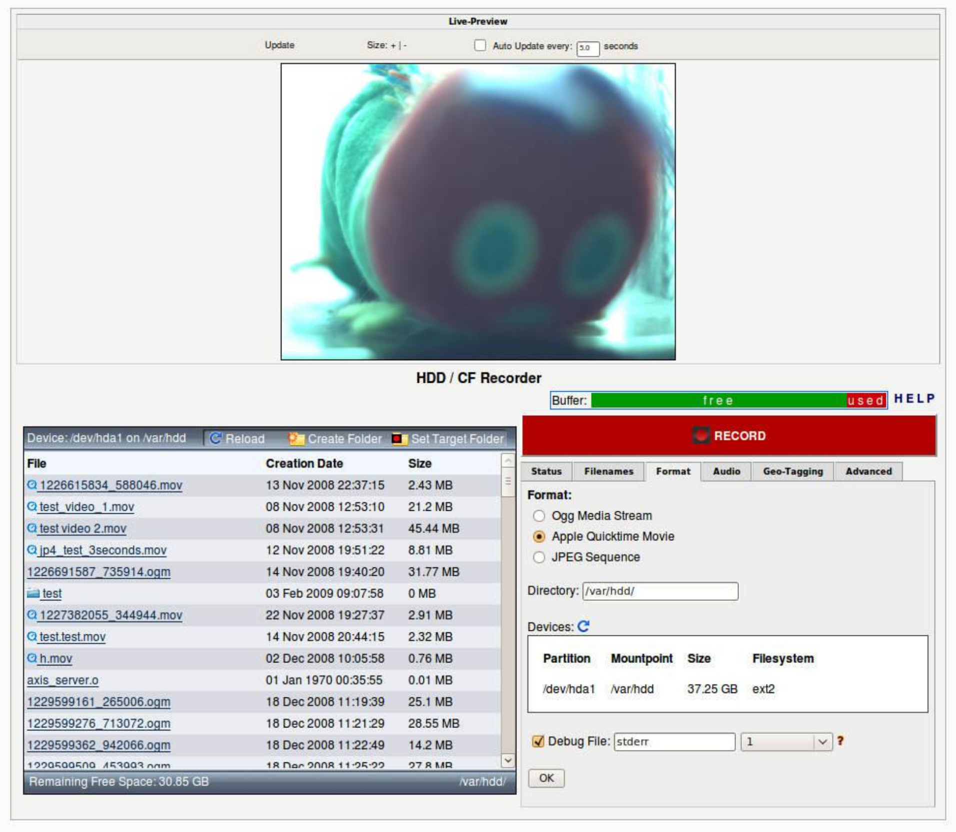
4.2.2. SAVING IMAGE TO AN EXTERNAL HARD DRIVE.

One application developed by Elphel called Camogm can perform this procedure, this program allows recording of the video/images acquired by Elphel 353/363 series cameras to the storage media. It is developed to use such media as hard disk drives, compact flash cards or USB storage devices (with reduced data rate as ETRAX FS processor currently supports only USB 1.1, USB 2).

Camogm is designed to run in the background and accept commands through a named pipe. It writes JPEG-encoded frames from the camera circbuf-circular video buffer in any of the 3 formats:

- ogm - MJPEG video in Xiph Ogg container
- jpeg - series of the individual JPEG files (1 file per frame)
- mov - MJPEG video in Apple QuickTime(R) container.

The wiki of the camera [1] provides details for using this application commands, so there's a php application installed in the camera which integrates in a GUI(Graphical User Interface) the commands of Camogm.



Device: /dev/hda1 on /var/hdd

File	Creation Date	Size
1226615834_588046.mov	13 Nov 2008 22:37:15	2.43 MB
test_video_1.mov	08 Nov 2008 12:53:10	21.2 MB
test video 2.mov	08 Nov 2008 12:53:31	45.44 MB
jp4_test_3seconds.mov	12 Nov 2008 19:51:22	8.81 MB
1226691587_735914.ogm	14 Nov 2008 19:40:20	31.77 MB
test	03 Feb 2009 09:07:58	0 MB
1227382055_344944.mov	22 Nov 2008 19:27:37	2.91 MB
test.test.mov	14 Nov 2008 20:44:15	2.32 MB
h.mov	02 Dec 2008 10:05:58	0.76 MB
axis_server.o	01 Jan 1970 00:35:55	0.01 MB
1229599161_265006.ogm	18 Dec 2008 11:19:39	25.1 MB
1229599276_713072.ogm	18 Dec 2008 11:21:29	28.55 MB
1229599362_942066.ogm	18 Dec 2008 11:22:49	14.2 MB
1229599509_453993.ogm	18 Dec 2008 11:25:22	27.8 MB

Remaining Free Space: 30.85 GB /var/hdd/

Format:
○ Ogg Media Stream
● Apple Quicktime Movie
○ JPEG Sequence

Directory: /var/hdd/

Devices:

Partition	Mountpoint	Size	Filesystem
/dev/hda1	/var/hdd	37.25 GB	ext2

☑ Debug File: stderr 1 ?

Figure 15: GUI of Camogm: Elphel's application to save image to an external Hard drive

4.2.3. STREAMING

The video generated by the FPGA compressor is sent over LAN interface using the RTSP protocol. In this section, a brief introduction to this protocol is made; we next describe the process of creating streaming from acquisition to delivery via LAN interface.

4.2.3.1. RTSP

The Real Time Streaming Protocol [6], or RTSP, is an application-level protocol for control over the delivery of data with real-time properties. RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data, such as audio and video. Sources of data can include both live data feeds and stored clips. This protocol is intended to control multiple data delivery sessions, provide a means for choosing delivery channels such as UDP, multicast UDP and TCP, and provide a means for choosing delivery mechanisms based upon RTP.

The Real-Time Streaming Protocol (RTSP) establishes and controls either a single or several time-synchronized streams of continuous media such as audio and video. It does not typically deliver the continuous streams itself, although interleaving of the continuous media stream with the control stream is possible.
In other words, RTSP acts as a "network remote control" for multimedia servers.

RTSP has the following properties:

Extendable:
New methods and parameters can be easily added to RTSP.

Easy to parse:
RTSP can be parsed by standard HTTP or MIME parsers.

Secure:
RTSP re-uses web security mechanisms. All HTTP authentication mechanisms such as basic and digest authentication are directly applicable.
One may also reuse transport or network layer security mechanisms.

Transport-independent:

RTSP may use either an unreliable datagram protocol (UDP), a reliable datagram protocol (RDP) or a reliable stream protocol such as TCP as it implements application-level reliability.

Multi-server capable

Each media stream within a presentation can reside on a different server. The client automatically establishes several concurrent control sessions with the different media servers. Media synchronization is performed at the transport level.

Control of recording devices:

The protocol can control both recording and playback devices, as well as devices that can alternate between the two modes ("VCR").

Live555 has developed an RTSP server [7] that runs inside the camera adding all frame related metadata to build a RTSP stream, which can be accessed with any RTSP client capable of decode MJPEG streaming .

This server called "str" runs when the camera is turned on and remains waiting for commands.

4.2.3.2. BUILDING THE STREAM

The information sent by the streamer (Figure 16: Basic streamer operation)is extracted accessing the circular buffer [1] which is a file containing JPEG encoded bit stream delivered by the FPGA compressor through DMA(direct memory access) channel implemented in the CPU, additionally this circular buffer contains besides the JPEG compressed image, all metadata needed to build RTP packets such as timestamps, width and height of the image, JPEG quality.

Live Networks developed an RTSP server application [8] which is running in the camera and accessing this buffer, create and send the packets according to the standard RTSP.

The streamer is a standard RTSP streamer (on the standard 554 RTSP port) and supports unicast or multicast streaming of video with audio. The camera will automatically be setup for unicast streams by default.

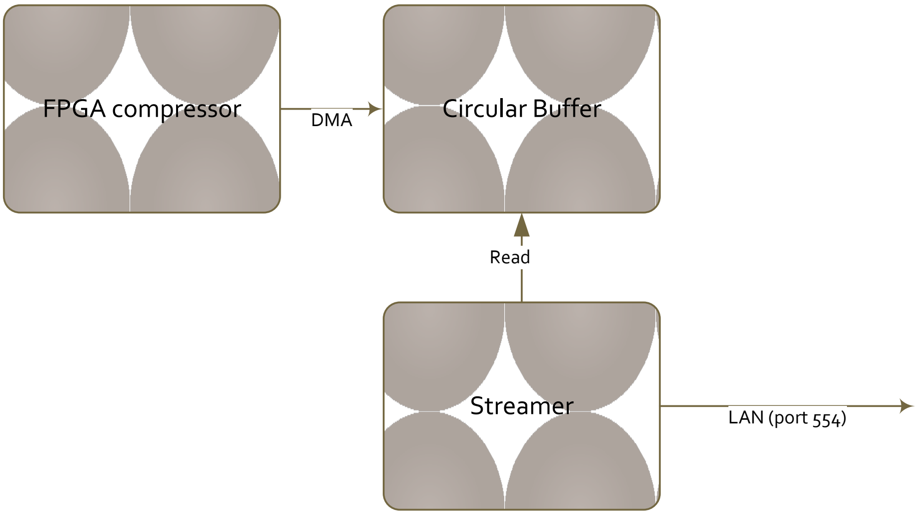


Figure 16: Basic streamer operation

5. PRESETS.

This section defines the settings applied on the host, external software and additional libraries used in the development.

5.1. HOST CONFIGURATION

The equipment used is an Acer Travelmate 5760G, Intel Core i3 2310 CPU dual core 2.10 GHz with 6 GB of Ram 300 GB HDD, with a dual boot OS Windows 8 premium and Ubuntu Linux 12.04.

Elphel provides some instructions to control the camera appropriately in Ubuntu Linux, thus Ubuntu Linux 12.04 was the O.S. used. Once in Linux, we should install the software to control the camera stored in a ppa, which can be downloaded writing in a terminal:

```
sudo add-apt-repository ppa:elphel/ppa
sudo apt-get update
```

The packages needed for developing can be installed by running the following commands:

```
sudo apt-get install cvs build-essential autoconf flex byacc bison libglib2.0-dev tcl gettext libncurses5-dev patch zlib1g-dev nfs-kernel-server bash xutils-dev
```

Other optional packages to install are set up writing in a terminal:

```
sudo apt-get install kinfocenter minicom firefox graphviz doxygen ctags cervisia php5 php5-cli xchat ssh kompare git-core
```

There is a set of software that can be installed optionally:

- Special C/C++ compiler for the processor working inside the camera
- Source code of Linux distro inside the camera
- Source code of the FPGA

This software is not essential in the development of this project but can be useful in another intended uses of the camera.

Besides installing camera's software, the tools for developing need to be set up too. Programming language used to develop is C++ [8] and the IDE picked was Kdevelop.

5.1.1. DEVELOPMENT ENVIRONMENT.

The development environment selected was Kdevelop; the main reason to use this IDE was because Elphel provides one Kdevelop project connected with camera's source code, so a core modification of the camera should be easier to accomplish.

Kdevelop [8] is a free, open source IDE (Integrated Development Environment) for Linux, Solaris, and FreeBSD.

Installation is done by running in a Linux terminal the next instructions:

```
sudo add-apt-repository ppa:kubuntu-ppa/backports
sudo apt-get update
sudo apt-get install kdevelop
sudo apt-get install cmake
```

Kdevelop uses a special tool to build the program called Cmake .

Cmake [9] is a family of tools designed to build, test and package software: It's used to control the software compilation process using simple platform and compiler independent configuration files. For this project Cmake 2.8.10 was the version installed.

5.1.2. ADDITIONAL LIBRARIES

These additional packages also have to be installed; its attributes are described in the following subsections. These libraries will be used to capture the host image using as source RTSP streaming generated by the camera.

5.1.2.1. GSTREAMER

GStreamer [10] is a powerful and versatile framework designed to create streaming media applications. Many of the virtues of this framework come from its modularity: GStreamer can seamlessly incorporate new plug-in modules.

From a technical point of view, the GStreamer framework was designed to make it easy to write applications that handle audio and video, giving the developer the ability of processing any kind of flow. Its main advantages are that the pluggable components can be mixed and matched into arbitrary pipelines so that it’s possible to write a full-fledged video or audio editing application.

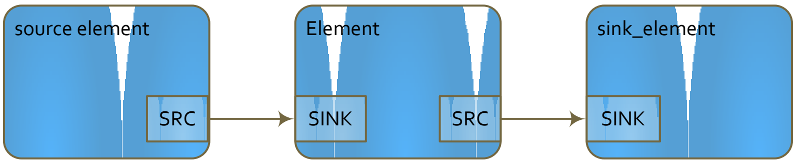


Figure 17: Basic Gstreamer's pipeline

GStreamer was one of the options to get the camera's stream. Taking advantage of its pipe structure (figure 17).

Ubuntu 12.04 comes with a version of GStreamer already installed: 0.10. For the project, is necessary to add only some plug-ins for decoding incoming image from the camera.

This command line instruction can record from mjpeg Elphel streaming and show it in a window (figure 18):

```
gst-launch-0.10 rtspsrc location=rtsp://IP*CAMERA:554 ! rtpjpegdepay ! jpegdec ! queue ! ffmpegcolorspace ! xvimagesink sync=false
```

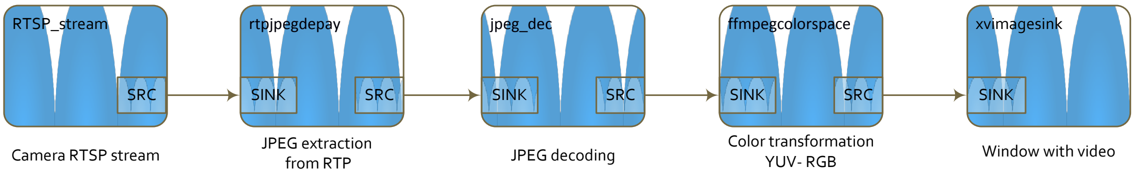


Figure 18: Gstreamer's pipeline for Elphel 353 stream

5.1.2.2. OPEN CV

(OPEN Source Computer Vision) [10] [11] A library of real-time computer vision routines from Intel. First released in 2000, Open CV code is used in applications such as object, face and gesture recognition, lip reading and motion tracking.

The library is cross-platform. It focuses mainly on real-time image processing. If the library finds Intel's Integrated Performance Primitives on the system, it will use these proprietary optimized routines to accelerate itself.

Features:

- Image data manipulation (allocation, release, copying, setting, conversion).
- Image and video I/O (file and camera based input, image/video file output).
- Matrix and vector manipulation and linear algebra routines (products, solvers, eigenvalues, SVD).
- Various dynamic data structures (lists, queues, sets, trees, graphs).
- Basic image processing (filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids).
- Structural analysis (connected components, contour processing, distance transform, various moments, template matching, Hough transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation).
- Camera calibration (finding and tracking calibration patterns, calibration, fundamental matrix estimation, homography estimation, stereo correspondence).
- Motion analysis (optical flow, motion segmentation, tracking).
- Object recognition (Eigen-methods, HMM).
- Basic GUI (display image/video, keyboard and mouse handling, scroll-bars).
- Image labeling (line, conic, polygon, text drawing).

In the experiments, the tools provided by openCV to capture RTSP streaming (class VideoCapture) will be used, once extracted the image from the stream, the data structure for managing matrices (Mat) will be served to processing coming next.

5.1.2.3. LIVE555 STREAMING MEDIA

This code forms a set of C++ libraries for multimedia streaming, using open standard protocols (RTP/RTCP, RTSP, and SIP). These libraries - which can be compiled for UNIX (including Linux and Mac OS X), Windows, and QNX (and other POSIX-compliant systems) - can be used to build streaming applications. The libraries grant access to the source code and examples in order to an easy integration with a proprietary C++ code

Live555 libraries [7] provide some enhanced tools for capturing online and offline an RTSP streaming; one of these implementations is "openRTSP".

"openRTSP" is a command-line program that can be used to open, stream, receive, and record media streams that are specified by a RTSP URL - i.e., an URL that begins with RTSP://
Developed by Live Networks, Inc., this tool is designed to control an RTSP camera's stream specifying some parameters[1].

Specifically for this project, only - m , -b, -Q options are used for development.

- -m output each incoming frame into a separate file.
- -b change the output file buffer size.
- -Q output 'QOS' Quality measures statistics about the data stream.

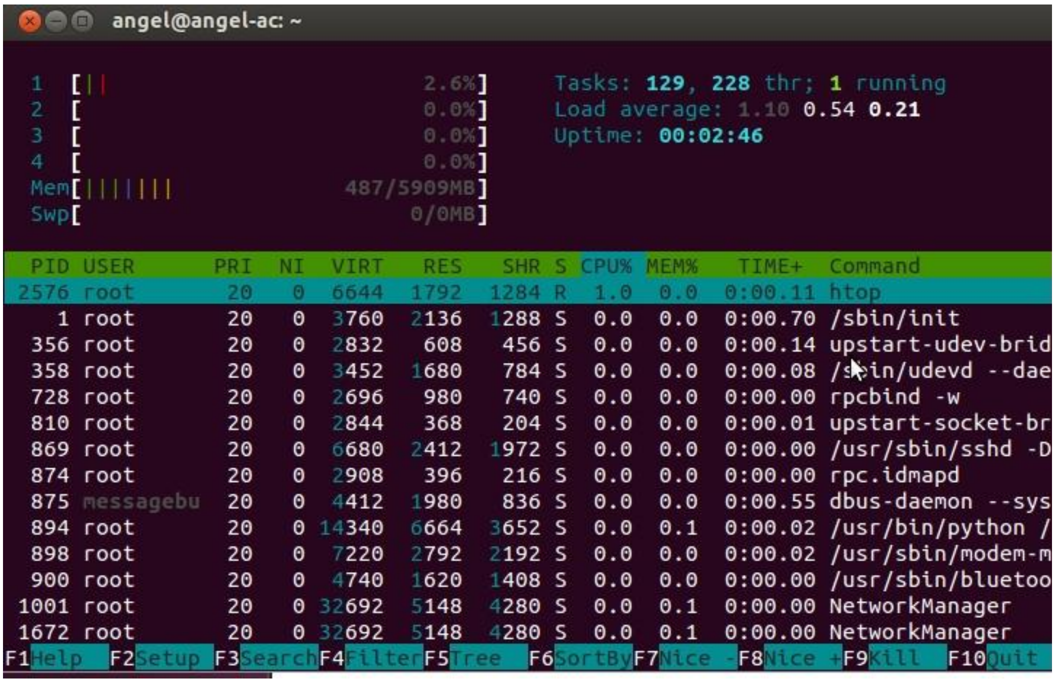
[1] total list of parameters in openRTSP application are included in Annexes

5.2. EVALUATION TOOLS

One of the most important objectives is to evaluate the CPU usage (%), disk usage (MB/ sec) and network load for each of the modes. This section lists the tools used to monitor the resources mentioned.

5.2.1. HTOP

This Linux command line application shows an interactive window with a description of the running process in the host.



angel@angel-ac: ~

```
  1  [||                        2.6%]     Tasks: 129, 228 thr; 1 running
  2  [                          0.0%]     Load average: 1.10 0.54 0.21
  3  [                          0.0%]     Uptime: 00:02:46
  4  [                          0.0%]
  Mem[|||||||            487/5909MB]
  Swp[                        0/0MB]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
 2576 root       20   0  6644  1792  1284 R  1.0  0.0  0:00.11 htop
    1 root       20   0  3760  2136  1288 S  0.0  0.0  0:00.70 /sbin/init
  356 root       20   0  2832   608   456 S  0.0  0.0  0:00.14 upstart-udev-brid
  358 root       20   0  3452  1680   784 S  0.0  0.0  0:00.08 /sbin/udevd --dae
  728 root       20   0  2696   980   740 S  0.0  0.0  0:00.00 rpcbind -w
  810 root       20   0  2844   368   204 S  0.0  0.0  0:00.01 upstart-socket-br
  869 root       20   0  6680  2412  1972 S  0.0  0.0  0:00.00 /usr/sbin/sshd -D
  874 root       20   0  2908   396   216 S  0.0  0.0  0:00.00 rpc.idmapd
  875 messagebu  20   0  4412  1980   836 S  0.0  0.0  0:00.55 dbus-daemon --sys
  894 root       20   0 14340  6664  3652 S  0.0  0.1  0:00.02 /usr/bin/python /
  898 root       20   0  7220  2792  2192 S  0.0  0.0  0:00.02 /usr/sbin/modem-m
  900 root       20   0  4740  1620  1408 S  0.0  0.0  0:00.00 /usr/sbin/bluetoo
 1001 root       20   0 32692  5148  4280 S  0.0  0.1  0:00.00 NetworkManager
 1672 root       20   0 32692  5148  4280 S  0.0  0.1  0:00.00 NetworkManager
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

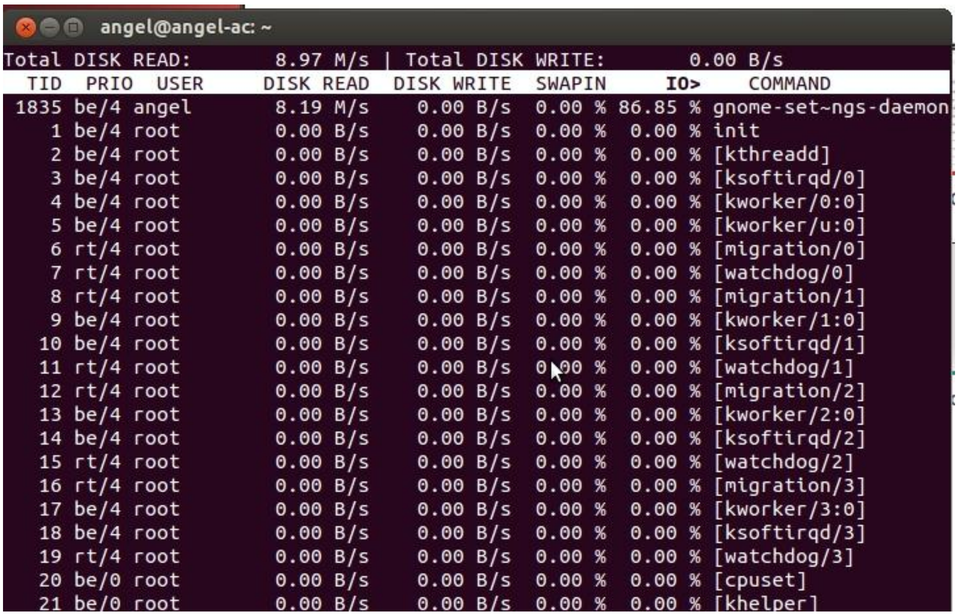
Figure 19: htop appearance

There are three notable sections in the application: the first summarizes the use of CPU, RAM and swap memory with horizontal bars. Additionally top right it can be found a brief statistical.

The second section describes a list of process currently running processes throughout the system sorted by CPU usage.

5.2.2. IOTOP

This tool, similar in appearance to htop shows the status of all input devices, i.e. hard drive (all partitions).

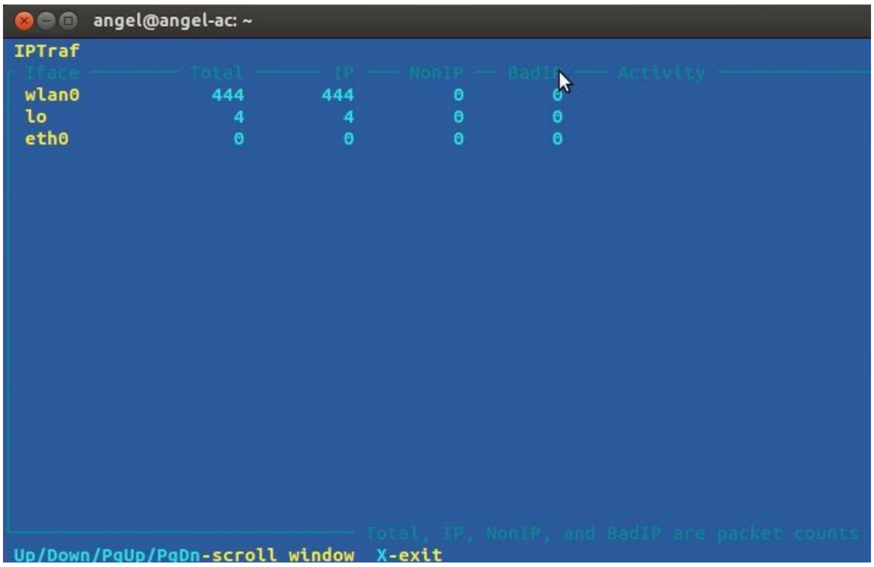


```
angel@angel-ac: ~
Total DISK READ:       8.97 M/s | Total DISK WRITE:       0.00 B/s
  TID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN     IO>    COMMAND
 1835 be/4 angel       8.19 M/s    0.00 B/s  0.00 % 86.85 % gnome-set~ngs-daemon
    1 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % init
    2 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [kthreadd]
    3 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [ksoftirqd/0]
    4 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [kworker/0:0]
    5 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [kworker/u:0]
    6 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [migration/0]
    7 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [watchdog/0]
    8 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [migration/1]
    9 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [kworker/1:0]
   10 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [ksoftirqd/1]
   11 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [watchdog/1]
   12 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [migration/2]
   13 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [kworker/2:0]
   14 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [ksoftirqd/2]
   15 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [watchdog/2]
   16 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [migration/3]
   17 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [kworker/3:0]
   18 be/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [ksoftirqd/3]
   19 rt/4 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [watchdog/3]
   20 be/0 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [cpuset]
   21 be/0 root        0.00 B/s    0.00 B/s  0.00 %  0.00 % [khelper]
```

Figure 20: IOtop appearance

5.2.3. IPTRAF

Iptraf [14] is a command line application that shows the network statistics. It shows informs such as TCP connection packet and byte counts, interface statistics and activity indicators, TCP/UDP traffic breakdowns, and LAN station packet and byte counts.



```
angel@angel-ac: ~
IPTraf
┌ Iface ────────── Total ───────── IP ─── NonIP ── BadIP ─── Activity ──────────
  wlan0               444          444        0        0
  lo                    4            4        0        0
  eth0                  0            0        0        0

└──────────────────────────────── Total, IP, NonIP, and BadIP are packet counts
Up/Down/PgUp/PgDn-scroll window  X-exit
```

Figure 21: Iptraf appearance

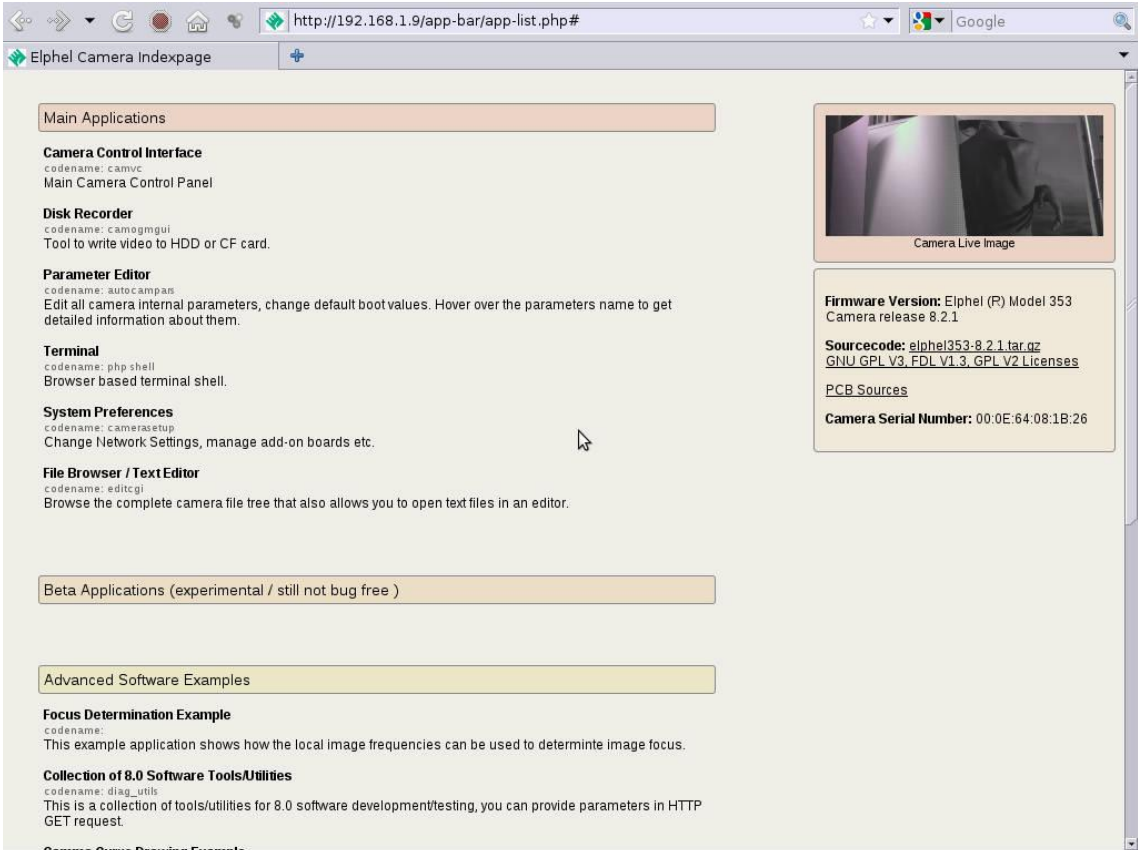
6. METHODS

In this chapter, the methods used to reach the objectives: change camera settings, assessing capture offline and online will be described.

6.1. CHANGIN CAMERA PARAMETERS

There are 2 methods for changing camera's parameters, The easiest is using camera GUI, where it can be changed almost every parameter in the camera like FPS, color gains, resolutions, types of binning, etc.

In a browser, navigate to http://IP_CAMERA, the result is this



Main Applications

Camera Control Interface
codename: camvc
Main Camera Control Panel

Disk Recorder
codename: camogmgui
Tool to write video to HDD or CF card.

Parameter Editor
codename: autocampars
Edit all camera internal parameters, change default boot values. Hover over the parameters name to get detailed information about them.

Terminal
codename: php shell
Browser based terminal shell.

System Preferences
codename: camerasetup
Change Network Settings, manage add-on boards etc.

File Browser / Text Editor
codename: editcgi
Browse the complete camera file tree that also allows you to open text files in an editor.

Beta Applications (experimental / still not bug free)

Advanced Software Examples

Focus Determination Example
codename:
This example application shows how the local image frequencies can be used to determinte image focus.

Collection of 8.0 Software Tools/Utilities
codename: diag_utils
This is a collection of tools/utilities for 8.0 software development/testing, you can provide parameters in HTTP GET request.

Firmware Version: Elphel (R) Model 353 Camera release 8.2.1

Sourcecode: elphel353-8.2.1.tar.gz
GNU GPL V3, FDL V1.3, GPL V2 Licenses

PCB Sources

Camera Serial Number: 00:0E:64:08:1B:26

Figure 22: Elphel's camera homepage

An html page the camera loads when start-up, this page includes some tools to modify all camera's parameters.

Some of these tools help us to exploit the resources of the camera such as:

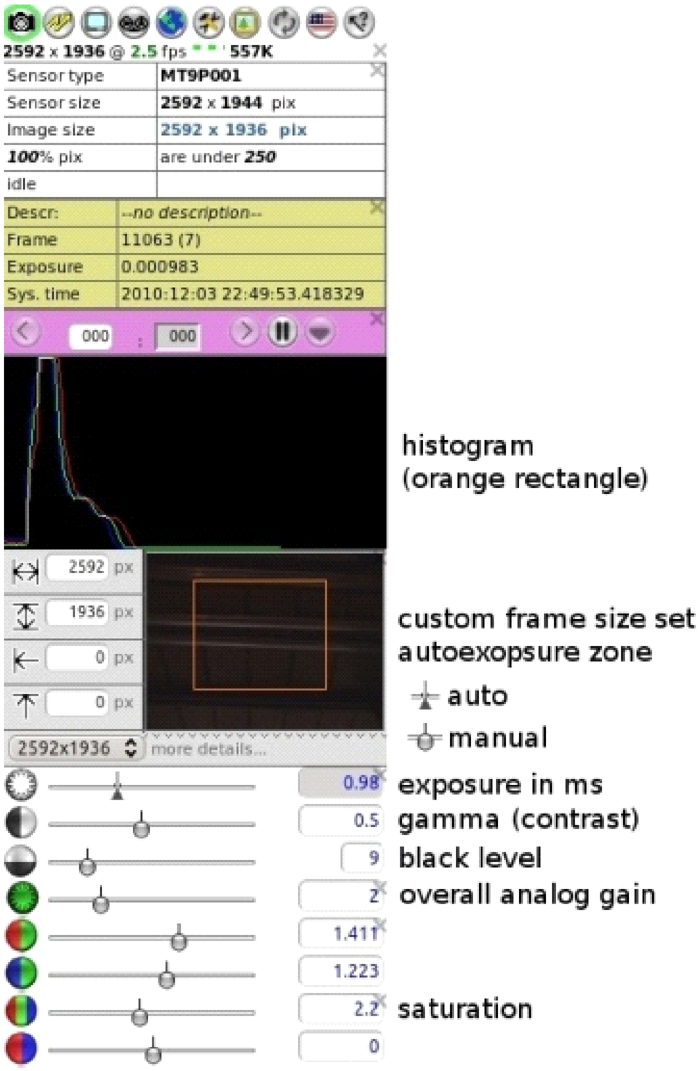
- Save images to a mounted Sata hard drive.

- Modify network's parameters (IP, Subnet Mask...)
- Use Linux distribution installed in the camera by a command line application.

Also, there are some sample applications which will serve to verify that the equipment works properly:

- Use the camera as a webcam.
- Save a single RAW frame.
- Gamma correction tables, updated in real time.
- Focus determination example.

If you choose the option Camera Control Interface, it will be displayed the following control panel.



2592 x 1936 @ 2.5 fps 557K

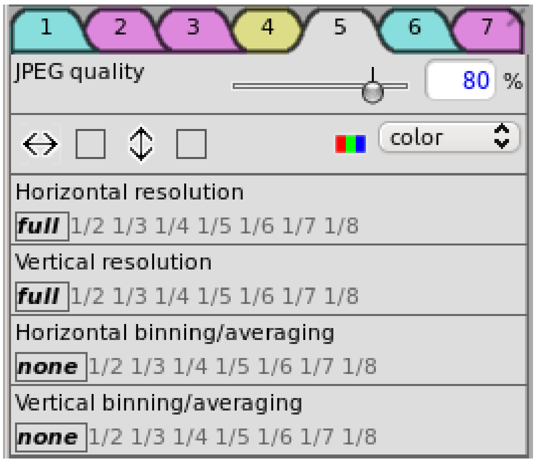
Sensor type	MT9P001
Sensor size	2592 x 1944 pix
Image size	2592 x 1936 pix
100% pix	are under *250*
idle	
Descr:	*--no description--*
Frame	11063 (7)
Exposure	0.000983
Sys. time	2010:12:03 22:49:53.418329

000 : 000

2592 px
1936 px
0 px
0 px
2592x1936
more details...

0.98
0.5
9
2
1.411
1.223
2.2
0

Figure 23: GUI to change camera's parameters



JPEG quality 80 %

color

Horizontal resolution
full 1/2 1/3 1/4 1/5 1/6 1/7 1/8

Vertical resolution
full 1/2 1/3 1/4 1/5 1/6 1/7 1/8

Horizontal binning/averaging
none 1/2 1/3 1/4 1/5 1/6 1/7 1/8

Vertical binning/averaging
none 1/2 1/3 1/4 1/5 1/6 1/7 1/8

Figure 24: GUI to change image's resolution

This way to control the parameters has the benefit that is a user-friendly interface but also has a lot of features that are not necessary in this project.

So, according to the requirements is not an acceptable option to modify the parameters using this application.

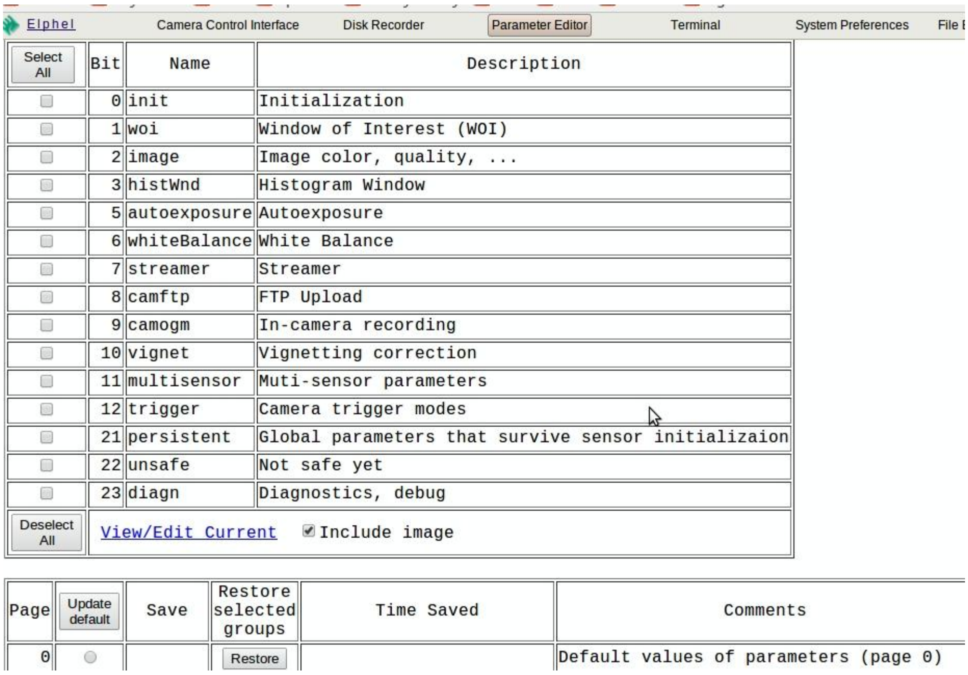
The second option is, Elphel's tool, developed in php, to change parameters: parsedit.php.

6.1.1. PARSEDIT.PHP

This software is designed to provide access to the camera acquisition related parameters through the web browser. It is easy to create customized parameter view/edit form with specially configured URL.

This application has two modes of using:

- Opening the browser in http://IP_CAMPERA/parsedit.php where each parameter may be just a name or a name=value pair. When the page opens, it includes multiple input fields and several control buttons, pressing button "apply" submits the form to the same parsedit.php in POST mode - this is when the new parameter values are applied to the camera. Most controls and parameter names have "tooltips" - short help messages that open when you hold mouse pointer over those page elements.



Select All	Bit	Name	Description
☐	0	init	Initialization
☐	1	woi	Window of Interest (WOI)
☐	2	image	Image color, quality, ...
☐	3	histWnd	Histogram Window
☐	5	autoexposure	Autoexposure
☐	6	whiteBalance	White Balance
☐	7	streamer	Streamer
☐	8	camftp	FTP Upload
☐	9	camogm	In-camera recording
☐	10	vignet	Vignetting correction
☐	11	multisensor	Muti-sensor parameters
☐	12	trigger	Camera trigger modes
☐	21	persistent	Global parameters that survive sensor initializaion
☐	22	unsafe	Not safe yet
☐	23	diagn	Diagnostics, debug
Deselect All	View/Edit Current ☑ Include image		

Page	Update default	Save	Restore selected groups	Time Saved	Comments
0	○		Restore		Default values of parameters (page 0)

Figure 25: parsedit.php appereance

This php tool has an easy way to read and write information of camera's settings.

Read:

http://192.168.0.9/parsedit.php?immediate&PAR1&PAR2

Change:

http://192.168.0.9/parsedit.php?immediate&PAR1=VAL1&PAR2=VAL2

The application developed is a bash script which executes the http command using the "wget" function. The function The function interprets and executes instructions as if it were a browser but without showing any results on screen.

6.1.1.1. CHANGING THE DIFFERENT SKIPPING/BINNING MODES

It can be set writing in a browser:
http://*cameraIP*/parsedit.php?embed=0.18&SENSOR_REGS32

The default value is "0x40" which means averaging binning mode.
If one set it to "0x60" you switch to summing binning mode and will get increased light sensitivity.

Although, one may control the binning and decimation of the camera sensor using the camera control interface (camvc). These for parameters: DCM_HOR, DCM_VERT, BIN_HOR and BIN_VERT

There is also available parsedit program or the custom PHP script running in a browser or with the command line application developed before:

http://*cameraIP*/parsedit.php?embed=0.18&DCM_HOR&DCM_VERT&BIN_HOR&BIN_VERT[2]

[2] Correct values to define in the variables DCM_HOR, DCM_VERT, BIN_HOR, BIN_VERT can be seen in the parsedit.php manual in the annexes.

6.1.1.2. CHANGING THE FRAME REFRESH RATE (IN FRAMES/SEC)

The frames per second are directly dependant of the trigger signals emitter. With this panorama, frames per second can be modified using 3 methods:

- Using automatic trigger mode. The trigger is controlled by the sensor itself.
- FPGA trigger.
- Reading external trigger signal.

The provided tool parsedit.php can switch between this 3 methods, changing some values in fields:

TRIG, TRIG_PERIOD, TRIG_DELAY, EARLY_TIMESTAMP, TRIG_CONDITION, TRIG_OUT

1. 1.Set TRIG_CONDITION = 0
2. 2.Set TRIG = 4
3. 3. Set the TRIG_PERIOD according to your desired FPS (96.000,000 / FPS). For our example of 25 FPS that will be 96.000,000 / 25 = 3840000
4. 4.Write in a browser:

http://*cameraIP*/parsedit.php?embed=0.18&TRIG_CONDITION=0&TRIG=4
&TRIG_PERIOD=3840000

6.2. OFFLINE MODE

According to the objectives, in this mode take the video stream of the camera and dump into a hard disk is the main task; to reach that goal, an analysis of which are the "bottlenecks", needs to be done.

It is supposed that in this working mode the only limitation is the image the camera can produce. Network speed limitation (100 Mbps) can handle with any problem from highest resolution with low frames per second to less resolution with higher frame rates as the stream has only JPEG compressed images. In the host there isn't any processing, decoding o scaling when frames are recorded. The only high usage of resources in the host should be in writing to disk.

Live555 offers a tool to capture a RTSP stream and dumping to a hard drive without any processing applied besides reading RTSP messages.

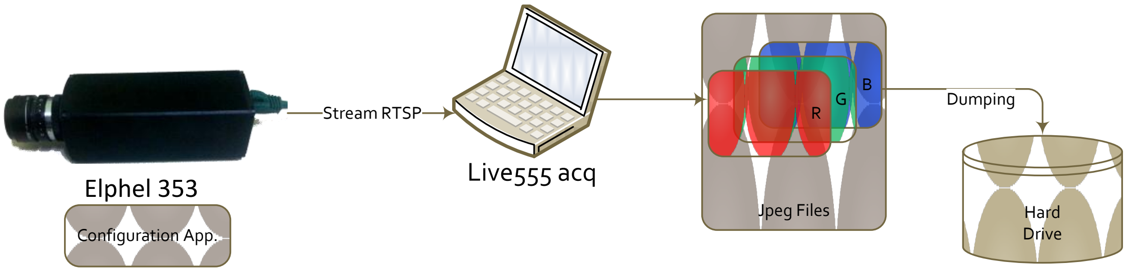


Figure 26: Basic Offline operation mode

Command line instruction for offline capture:

open RTSP -m -b 50000 rtsp://IPCAMERA

With this command line instruction, launched on the folder we want to save images from the camera in the disk, communication and decoding RTSP is established.

Option -m indicates 1 file per frame is needed, as the stream is MJPEG, it will save one jpeg image for each frame streamed by the camera.

The name of the file created will have this format:

TYPEOFSTREAMING-CODEC-NSESSION-TIMESTAMP

An example of recorded file from Elphel camera is: video-JPEG-1-3455936759.604702.

Next step in this operation mode is transforming the acquired frames in a format known by the researchers. This format requires a .png frame with a format name like seq00001-seq00000n and a creation of a seq.index file which will have the correspondences of every frame with its timestamp.

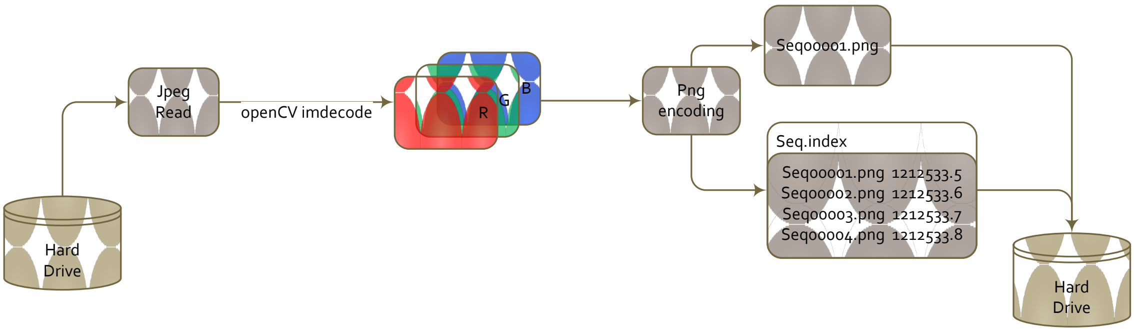


Figure 27: JPEG to PNG operation mode

Png encoding is made with native openCV libraries and seq.index with basic C++ file management functions.

An annex to this operation mode is the development of a viewer tool that read files dumped before, in a viewer to see results.

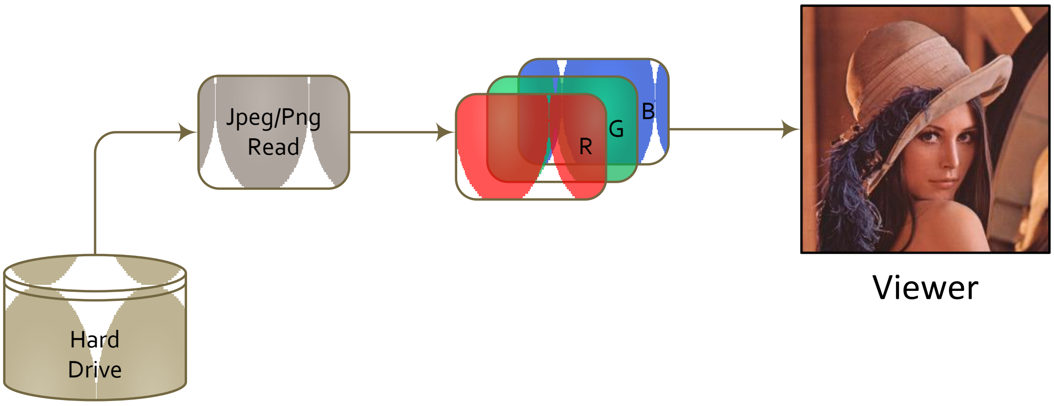


Figure 28: Offline viewer

Results in this mode can be improved placing the recording mode of the camera in JP46.

6.2.1. JP46 OFFLINE MODE

With this recording mode in the camera, the quantity of frames per second is achieved as it free some load of CPU's camera.

Configuring the camera in this recording mode will place images encoded with JP46 algorithm in the stream jpeg, without applying demosaic. As it can be seen in introduction, JP46 applies a 16x16 blocking in the jpeg compressed image and send it in grey scale.

Using this mode, a de-block/debayer processing is needed in the host for the purpose of having a stream or record RGB images.

In order to get a RGB image the process inverted has to be applied.

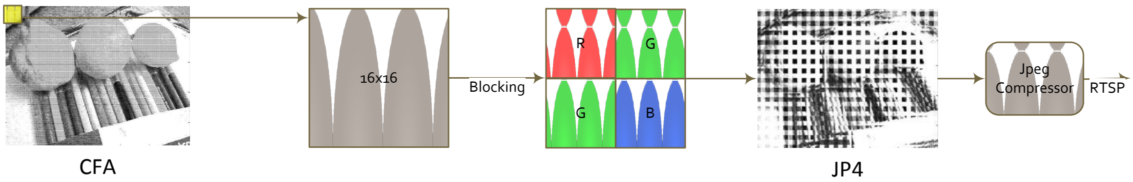


Figure 29: JP4 encoding, example

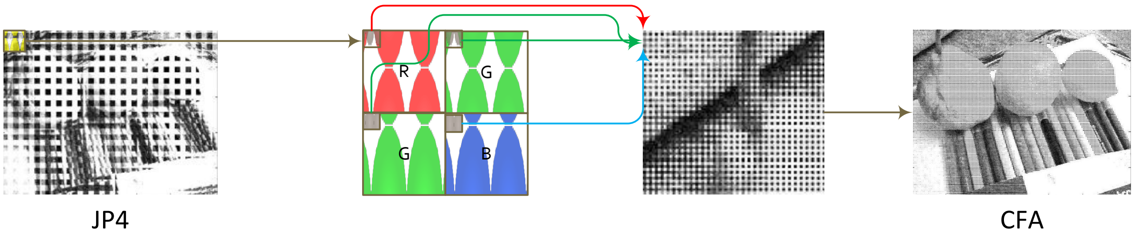


Figure 30: Deblock 16x16. JP4 to CFA (color filter array)

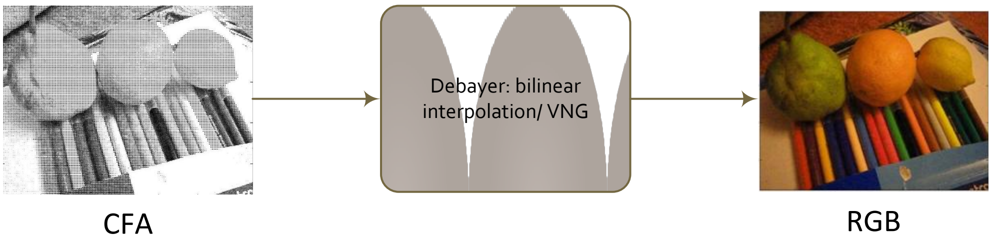


Figure 31: Debayering

The process of debayering can be done with some algorithms: variable number of gradients (VNG), pixel grouping, Adaptive homogeneity-directed interpolation.

Bilinear interpolation is used by Elphel's camera.

Later, images stored in this format have to be processing in order to get an RGB image. This processing called deblocking + demosaic (Figure 32: Offline later processing) can be made any time after capture.

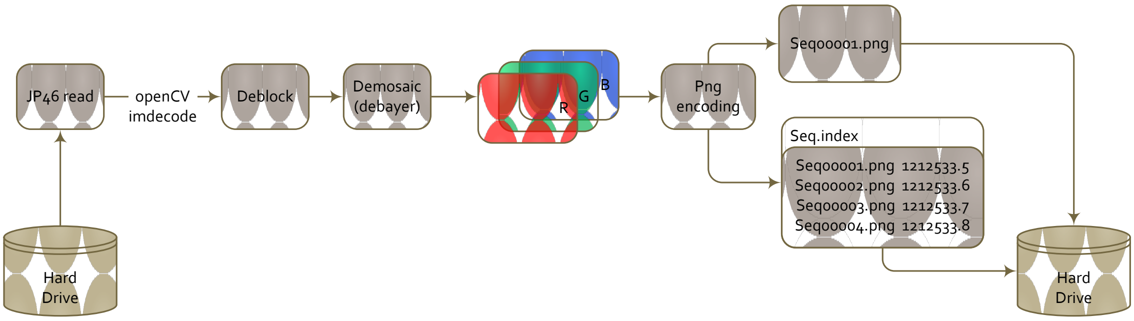


Figure 32: Offline later processing

6.3. ONLINE MODE

The main goal was unload CPU usage from the host client. The only way to obtain this, is transmitting from the camera, uncompressed data requiring minimum processing to get a RGB image.

With this panorama, we had 4 options:

1. Modify source code of streamer in the camera: RTSP server.
2. Modify source code of FPGA in order to get non compressed image.
3. Get data over the network reading a buffer placed in Linux filesystem: "/dev/ccam_img".
4. Placing in the host a GPU unit to dedicated jpeg decompressing.

The first two options presented a difficulty that is not in the scope of this project, the inconvenient of the next one is that the acquisition needs to be stopped, that lead us to lower the frame rate or the resolution of the image to values not too practical.

For better results, option 4 should be considered for future applications.

The online mode needs to have an RGB image in its output in a format known by the researchers, which is an openCV Mat [3] structure.

Assuming the host has any problems in the JPEG decoding, there will be evaluated two tools for extracting images from a RTSP:

- openCV VideoCapture class (Figure 33: Online mode with openCV).
- Live555 testopenRTSP model (Figure 34: Online mode with Live555).

[3] openCV C++ matrix class

6.3.1. OPENCV VIDEOCAPTURE

```
  cv::VideoCapture vcap;
   cv::Mat image;
   const std::string videoStreamAddress = "rtsp://192.168.0.9:554?tcp";
   /* it may be an address of an mjpeg stream,
   e.g. "http://user:pass@cam_address:8081/cgi/mjpg/mjpg.cgi?.mjpg"
*/

   //open the video stream and make sure it's opened
if(!vcap.open(videoStreamAddress)) {
       std::cout    "Error opening video stream or file"    std::endl;
       return -1;
   }

     for(;;) {
    //read image
     if(!vcap.read(image)) {
          std::cout    "No frame"    std::endl;
```

Class for video capturing from video files or cameras, The class provides C++ video capturing API. The code in C++ for extracting the image from the stream is described above.

6.3.2. LIVE555 TESTOPENRTSP

Sample program where it is described the process to extract a sequence of images from a RTSP stream. The application developed receives the media contained in a int pointer, then create an openCV Mat object including this information, creating the Mat object implies a JPEG decoding too.

In the following code block, it will be described the actions taken to create the Mat object.

```
void DummySink::afterGettingFrame(unsigned frameSize, unsigned numTruncatedBytes,
                                                    struct timeval presentationTime, unsigned
/*durationInMicroseconds*/) {
  // We've just received a frame of data. :

              cv::Mat imagen;
              cv::Mat imageRGB;
std::vector uint8_t:  buffer (fReceiveBuffer,fReceiveBuffer+frameSize);
 cv::Mat fram = cv::imdecode(cv::Mat(buffer),CV_LOAD_IMAGE_COLOR);

   // Then continue, to request the next frame of data:
  continuePlaying();
}

Boolean DummySink::continuePlaying() {
  if (fSource == NULL) return False; // sanity check (should not happen)

  // Request the next frame of data from our input source.  "afterGettingFrame()" will get called
later, when it arrives:
  fSource-: getNextFrame(fReceiveBuffer, DUMMY_SINK_RECEIVE_BUFFER_SIZE,
                   afterGettingFrame, this,
                   onSourceClosure, this);

  return True;
}
```

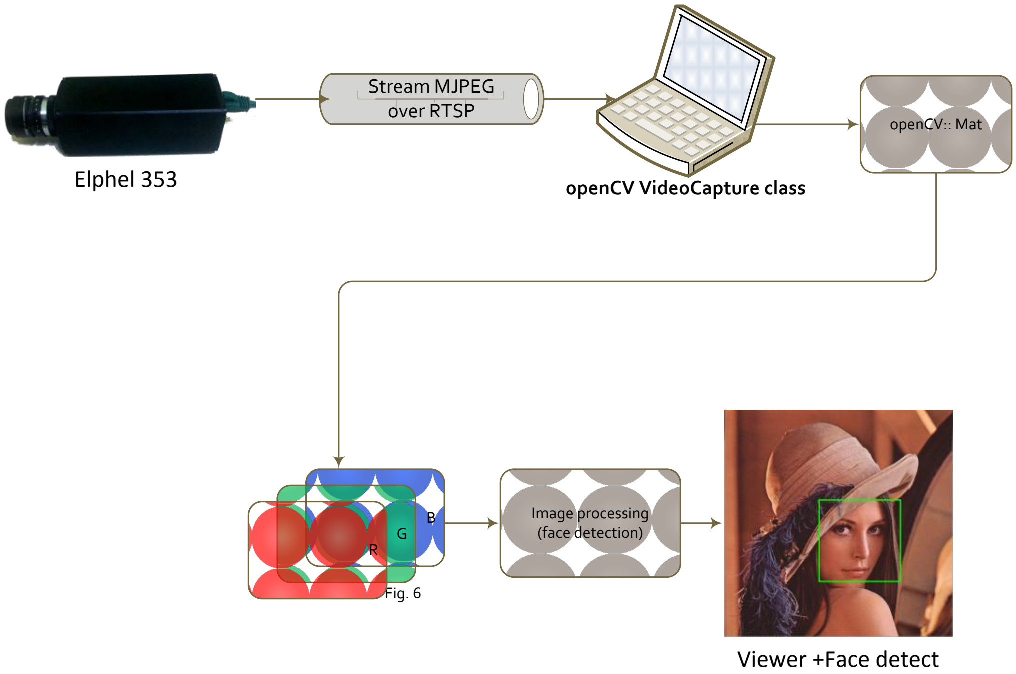


Figure 33: Online mode with openCV

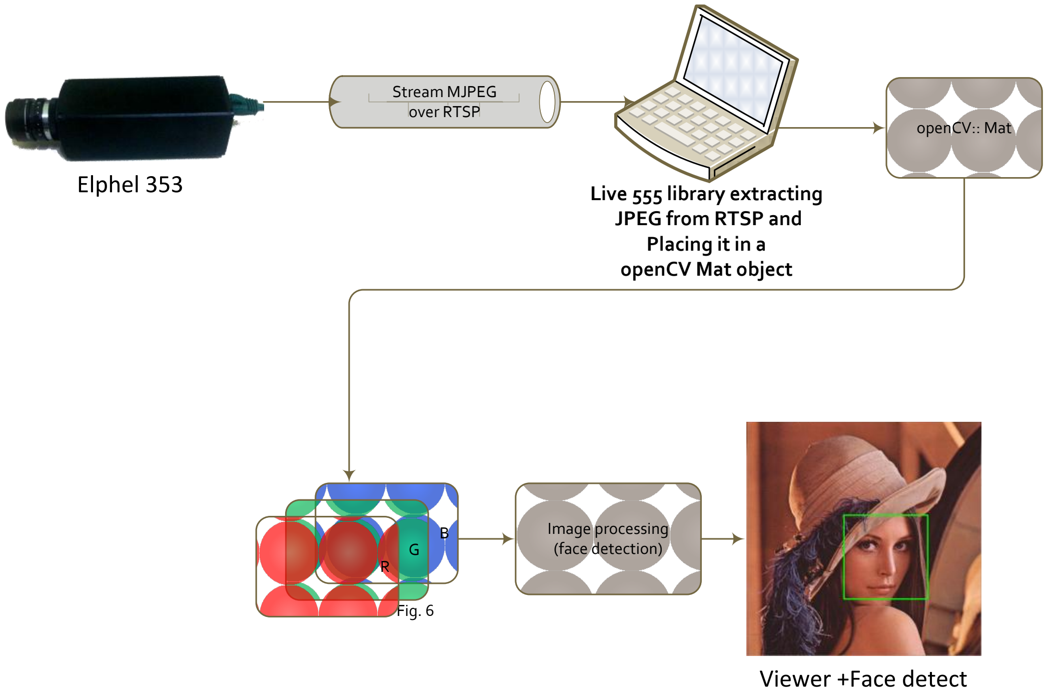

Stream MJPEG over RTSP
Elphel 353
Live 555 library extracting JPEG from RTSP and Placing it in a openCV Mat object
openCV:: Mat
B
G
R
Fig. 6
Image processing (face detection)
Viewer +Face detect

Figure 34: Online mode with Live555

6.4. EXPERIMENTAL PARAMETERS

In order to evaluate the performance of the capture some tests should be made, a few changes in capture's parameters need to be applied in each test:

- Resolutions to make the tests:

At full resolution: 2592x1936

Half resolution: 1292x960

Half of it: 640x480.

- Four different frames per second: 30, 25,20,15,10.
- Length of the capture: 2000 frames.

The experiments consist of measuring, for each proposed configuration using monitoring tools described in previous sections:

- CPU load (%) (htop)
- A measure of disk write usage (MB / s) (iotop)
- The load on the network (Mbps) (iptraf)
- Drop frames count (%), quantity of dropped frames in 2000 captured.

7. RESULTS

Presenting the tables filled with information gathered according to experimental parameters will be served to assess online and offline possible capture methods.

7.1. ONLINE CAPTURE

Resolution	FPS	CPU (%)	HD (MB/s)	Network (kbps)	Dropframes (%)
2592x1536	30	-	-	-	-
1296x960	30	68	0	30680	2
640x480	30	56	0	8290,4	0,2
2592x1536	25	-	-	-	-
1296x960	25	65	0	25460,6	1
640x480	25	45	0	6526,3	0,1
2592x1536	20	-	-	-	-
1296x960	20	55	0	18599,2	0,1
640x480	20	45	0	4522,3	0,1
2592x1536	15	-	-	-	-
1296x960	15	50	0	13974,2	2/15, from 1000 frames
640x480	15	45	0	3087,4	0,1
2592x1536	10	50	0	36893,2	0,5
1296x960	10	52	0	9229	1/10 from 1000 frames
640x480	10	50	0	2669,8	0,1

Table 1: Evaluation of Live555 online mode

Resolution	FPS	CPU (%)	HD (MB/s)	Network (kbps)	Dropframes (%)
2592x1536	30	-	-	-	-
1296x960	30	55	0	30680	0,1
640x480	30	20	0	8290,4	0
2592x1536	25	-	-	-	-
1296x960	25	40	0	25460,6	0
640x480	25	20	0	6526,3	0
2592x1536	20	-	-	-	-
1296x960	20	35	0	18599,2	0
640x480	20	20	0	4522,3	0
2592x1536	15	-	-	-	-
1296x960	15	30	0	13974,2	0
640x480	15	18	0	4087,4	0
2592x1536	10	-	0	-	-
1296x960	10	28	0	9229	0
640x480	10	13	0	2669,8	0

Table 2: Evaluation of openCV online mode

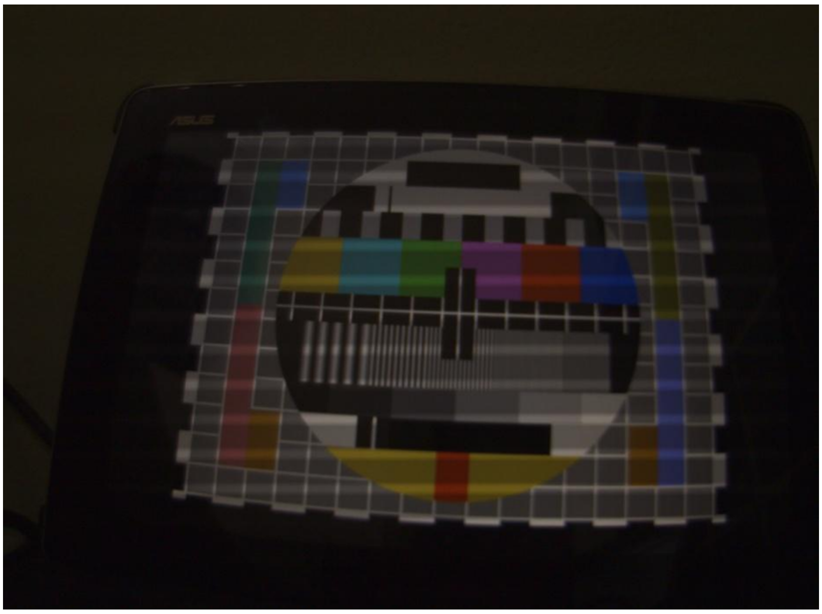


Figure 35: online snapshot.

- Online capture, with openCV implementation, is superior in terms of dropframes % and CPU usage. Having a greater number of dropped frames with live555 library tell that the code trying to read the image of the stream, is not the most optimal for the creation of a Mat structure, unlike VideoCapture (openCV) which makes it automatically.

- The CPU usage has an average of 60% for live555 capture taking into account all possible resolutions and frames per second, whereas with openCV is 50 %, and is reduced as you reduce the frame rate or image resolution. That makes me infer that the code used in the capture live555 keeps unnecessary CPU usage, which is reflected in the results of the drop frames too.

- It can be seen that there is any problem in fitting in the Ethernet channel (100 Mbps), because if one looks at the highest resolution with the highest number of frames per second (2592x1536 10 fps), it only reaches at 36893.2 kbps, value that can be transmitted smoothly through the LAN interface. The network load is the same in both cases(live555 and openCV), logical fact because it only changes the way you read this information but not the transmission, and

- As expected, there is no use of writing to disk in this mode of capture.

7.2. OFFLINE CAPTURE

Resolution	FPS	CPU (%)	HD (MB/s)	Network (kbps)	Dropframes (%)
2592x1536	30	-	-	-	-
1296x960	30	5	3,8	30680	0
640x480	30	5	2,2	8290,4	0
2592x1536	25	-	-	-	-
1296x960	25	5	3,2	25460,6	0
640x480	25	5	1,7	6526,3	0
2592x1536	20	-	-	-	-
1296x960	20	5	2,6	18599,2	0
640x480	20	5	1,3	4522,3	0
2592x1536	15	-	-	-	-
1296x960	15	5	1,9	13974,2	0
640x480	15	5	1	4087,4	0
2592x1536	10	5	4,8	-	0
1296x960	10	5	1,3	9229	0
640x480	10	5	0,6	2669,8	0

Table 3: Evaluation of offline mode

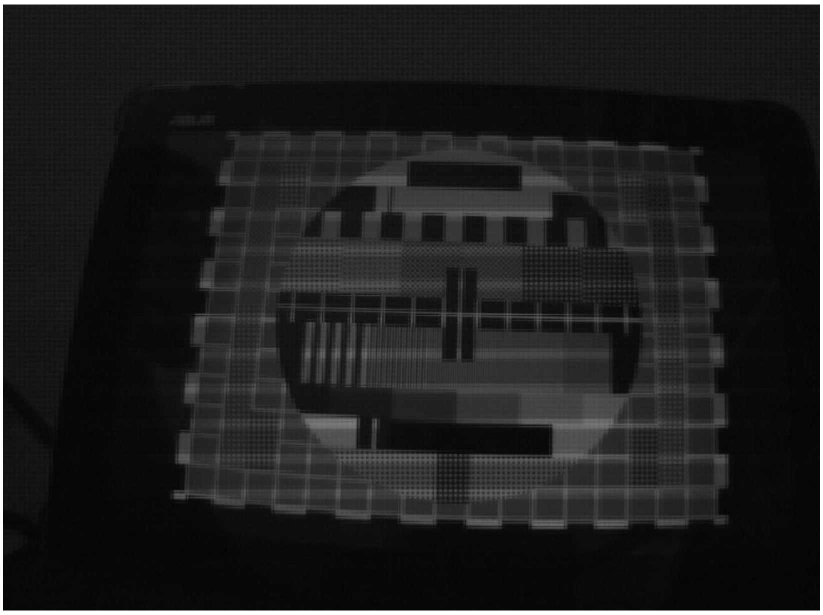


Figure 36: Offline snapshot

- One limitation that could exist in this mode of capture was the network load, is solved sending compressed images that, as mentioned in the analysis of online mode, has no problems to be transmitted over the LAN interface.

- Knowing the limit of disk writing is 60 MB/s, the disk usage is not critical, even at highest resolution.

8. CONCLUSIONS

At the beginning of the document the objectives of the work were defined, in this section they are reviewed and some conclusions are drawn.

- Objectives achieved and limitations

- Throughout the document the most important properties of the camera were described. The experiments performed and the applications developed have been the result of a careful study of the camera resources.

- The software developed for changing the parameters as explained in the methods section, is based on Elphel's parsedit.php tool. It is a command line application written in bash script that receives as parameters the values of the camera's parameters to change.The application executes an "wget" instruction which can be modified using the tags defined in the manual parsedit.php, i.e. in a future work, we can define a list of the most common parameters to modify in the acquisition and based on the application developed in this document, customize camera settings at will.

At the moment, the script can only modify the parameters that are necessary for the execution of the experiments: mode color, FPS (frames per second) and image resolution with skipping/binning mode.

- A capture client was developed, which draws and decode the image from the camera RTSP streaming based in openCV VideoCapture class.

In the online case, the decoded image is used in other image processing modules. There is also the option to save the image to disk, but this option will be very inefficient as the computer would not be able to perform both operations (decode/save to disk) in real-time thus frames would be dropped.

For the offline mode operation, the" Live media networks" tool called openRTSP has proven to be a viable option to extract image from an RTSP stream and save to disk without any processing in between.

The evaluation of performance done in chapter 7 confirms that the highest resolutions and frames per second obtaining an optimal drop frames value (under 0.1 %), with a complete FOV (field of vision) and an acceptable CPU usage, are:

2592x1536 at 10 fps in an offline capture.
1296x960 at 30 fps in an online capture.

The provisions made in this document will serve as a reference in the work necessary for integration into the smart room. The next step in development is the extension of current capture client in order to manage a multiple camera capture, the study of this scenario will present new challenges such as synchronization.

9. REFERENCES

[1] Elphel, «http://wiki.elphel.com/,» [En línea].

[2] A. I. Corporation, «http://www.aptina.com/,» [En línea].

[3] Axis, «http://developer.axis.com/wiki/doku.php?id=axis:sdk,,» [En línea].

[4] A. N. Filippov, «http://docs.elphel.com/linuxdevices/AT0000000000.html,» [En línea].

[5] Wikipedia, «http://en.wikipedia.org/,» [En línea].

[6] C. Poynton, «Chroma Subsampling Notation,» 2008.

[7] IETF, «Real Time Streaming Protocol (RTSP),» 1998.

[8] L. N. «http://www.live555.com/liveMedia/,» [En línea].

[9] Cplusplus, «http://www.cplusplus.com/,» [En línea].

[10] Kdevelop, «http://kdevelop.org/,» [En línea].

[11] Cmake, «http://www.cmake.org/,» [En línea].

[12] E. Walthinsen, «http://docs.gstreamer.com/display/GstSDK/Home,» [En línea].

[13] A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library.

[14] WillowGarage, «http://opencv.willowgarage.com/wiki/,» [En línea].

[15] Iptraf, «http://iptraf.seul.org/about.html,» [En línea].

[16] qazav_szaszak, «http://szaszak.wordpress.com/linux/elphel-as-a-digital-cinema-camera/,» [En línea].

[17] Elphel, «http://blog.elphel.com/,» [En línea].

[18] Elphel, «http://www.mail-archive.com/support-list@support.elphel.com/,» [En línea].

[19] Ubicast, «https://code.google.com/p/gst-plugins-elphel/,» [En línea].

[20] Live555, «http://blog.gmane.org/gmane.comp.multimedia.live555.devel,» [En línea].

[21] A. G. Moreno, «Sistema de gestió de vídeo off-line,» UPC, 2007.

[22] Ubuntu, «http://www.ubuntu.com/,» [En línea].

[23] J. C. Ferret, «Adquisició de senyal de vídeo multicàmera per a una smartroom.,» UPC, 2004.

[24] N. Kohl, «http://en.cppreference.com/w/,» [En línea].

[25] S.-Y. Huang and J.-S. Wang, "A low-cost desktop videoconferencing codec: an adaptive Motion-JPEG design," Consumer Electronics, IEEE Transactions, 1994.

[26] F. Harris y D. Wright, «The JPEG Algorithm for Image Compression: A Software Implementation and some Test Results,» de *Signals, Systems and Computers, 1990 Conference Record Twenty-Fourth Asilomar Conference on*, 1990.

[27] Apertus, «https://www.apertus.org/elphelcamera,» [En línea].

[28] K. Telenoika, «http://www.kinoraw.net/en,» [En línea].

[29] Live555, «http://www.live555.com/liveMedia/,» [En línea].

10.ANNEXES

10.1. OPTIONSFOR OPENRTSP

-4	output a '.mp4'-format file (to 'stdout')
-a	play only the audio stream
-A CODEC-NUMBER:	specify the static RTP payload format number of the audio codec to request from the server ("PLAYSIP" ONLY)
-b BUFFER-SIZE:	change the output file buffer size
-B BUFFER-SIZE:	change the input network socket buffer size
-c	play continuously
-M MIME-SUBTYPE:	specify the MIME subtype of a dynamic RTP payload format for the audio codec to request from the server ("PLAYSIP" ONLY)
-d DURATION:	specify an explicit duration
-D MAXIMUM-INTER-PACKET-GAP:	specify a maximum period of inactivity to wait before exiting
-f FRAME-RATE:	specify the video frame rate (used only with "-q", "-4", or "-i")
-F FILENAME-PREFIX:	specify a prefix for each output file name
-h HEIGHT:	specify the video image height (used only with "-q", "-4", or "-i")
-H	output a QuickTime 'hint track' for each audio/video track (used only with "-q" or "-4")
-i	output a '.avi'-format file (to 'stdout')
-l	try to compensate for packet losses (used only with "-q", "-4", or "-i")

-m output each incoming frame into a separate file

-n be notified when RTP data packets start arriving

-o request the server's command options, without sending "DESCRIBE" ("OPENRTSP" ONLY)

-O don't request the server's command options; just send "DESCRIBE" ("OPENRTSP" ONLY)

-p STARTINGPORTNUMBER: specify the client port number(s)

-Q output 'QOS' statistics about the data stream (when the program exits)

-q output a QuickTime '.mov'-format file (to 'stdout')

-r play the RTP streams, but don't receive them yourself

-s INITIAL-SEEK-TIME: request that the server seek to the specified time (in seconds) before streaming

-S BYTE-OFFSET: assume a simple RTP payload format (skipping over a special header of the specified size)

-t stream RTP/RTCP data over TCP, rather than (the usual) UDP. ("OPENRTSP" ONLY)

-T HTTP-PORT-NUMBER: like "-t", except using RTSP-over-HTTP tunneling. ("OPENRTSP" ONLY)

-u USERNAME: PASSWORD: specify a user name and password for digest authentication

-V print less verbose diagnostic output

-v play only the video stream

-w WIDTH: specify the video image width (used only with "-q", "-4", or "-i")

-y try to synchronize the audio and video tracks (used only with "-q" or "-

	4")
-z SCALE:	request that the server scale the stream (fast-forward, slow, or reverse play)

10.2. TYPICAL ERRORS.

Action	Error	Solution
Open RTSP connection	Server is busy	Change resolution of the camera.
Changing FPS	Low frame rate	Lower Auto exposure parameter.

Table 4: List of typical errors and solutions founded in developing

10.3. PARSEDIT.PHP TAGS, POSIBLE VALUES AND DESCRIPTION.

SENSOR: "Sensor ID number is determined by driver. Writing 0 to this location will re-start sensor identification" /SENSOR:

SENSOR_RUN: "Sensor can be stopped (0), acquire a single frame (1) or run continuously (2)" /SENSOR_RUN:

SENSOR_SINGLE: "Pseudo-register to write SENSOR_RUN_SINGLE here. Same as SENSOR_RUN, just the command will not propagate to the next frames" /SENSOR_SINGLE:

ACTUAL_WIDTH: "Actual image size (appropriately divided when decimation is used) - readonly" /ACTUAL_WIDTH:

ACTUAL_HEIGHT: "Actual image height (appropriately divided when decimation is used) - readonly" /ACTUAL_HEIGHT:

BAYER: "Bayer mosaic shift 0..3 (+1 - swap horisontally,+2 - swap vertically)" /BAYER:

PERIOD: "Frame period in pixel clock cycles - readonly" /PERIOD:

FP1000SLIM: "FPS limit as integer number of frames per 1000 seconds" /FP1000SLIM:

FRAME: "Absolute frame number, counted by the sensor frame sync pulses. Includes the frames that are not compressed and never appeared in the circbuf." /FRAME:

CLK_FPGA: "Sensor clock in HZ (so 96MHz is 96000000)" /CLK_FPGA:

CLK_SENSOR: "FPGA compressor/memory clock in HZ (so 1600Hz is 160000000)" /CLK_SENSOR:

FPGA_XTRA: "Extra clock cycles compressor needs to compress a frame in addition to the number of compressed pixels (for non-jp4 images each sensor pixel needs 3 FPGA clock cycles, for some of the jp4 modes - 2 clocks/pixel" /FPGA_XTRA:

TRIG: "Trigger mode. currently 0 - free running, 4 - triggered by external signal or FPGA timing generator." /TRIG:

EXPOS: "Exposure time in microseconds. Sensor driver modifies the value of this parameter to be multiple of sensor scan line times (see VEXPOS)" /EXPOS:

BGFRAME: /BGFRAME:

IMGSZMEM: /IMGSZMEM:

PAGE_ACQ: /PAGE_ACQ:

PAGE_READ: /PAGE_READ:

OVERLAP: /OVERLAP:

VIRT_KEEP: "Preserve \"virtual window\" size when resizing the window of interest (WOI) if non-zero. That will preserve the same FPS when resizing WOI" /VIRT_KEEP:

VIRT_WIDTH: "Width of the virtual window defines the time of reading out 1 sensor scan line. Normally this parameter is determined by the driver automatically, but may be manually modified." /VIRT_WIDTH:

VIRT_HEIGHT: "Height of the virtual window defines the frame duration in scan lines. Readout period (in free-running, not externally triggered mode) is equal to the product of VIRT_WIDTH * VIRT_HEIGHT. Normally this parameter is determined by the driver automatically, but may be manually modified." /VIRT_HEIGHT:

WOI_LEFT: "Window of interest left margin. Should be even number" /WOI_LEFT:

WOI_TOP: "Window of interest top margin. Should be even number" /WOI_TOP:

WOI_WIDTH: "Window of interest width. Should be multiple of 16 (divided by decimation if any). This parameter is modified by the driver according to the sensor capabilities, so if you put 10000 this value will be reduced to the full sensor width." /WOI_WIDTH:

WOI_HEIGHT: "Window of interest height. Should be multiple of 16 (divided by decimation if any). This parameter is modified by the driver according to the sensor capabilities, so if you put 10000 this value will be reduced to the full sensor width." /WOI_HEIGHT:

FLIPH: "Mirroring (flipping) the image horizontally. Driver is aware of the sensor orientation in Elphel cameras so value 0 is used for normal image orientation when captured by the camera (contrary to the previously released software)" /FLIPH:

FLIPV: "Mirroring (flipping) the image vertically. Driver is aware of the sensor orientation in Elphel cameras so value 0 is used for normal image orientation when captured by the camera (contrary to the previously released software)" /FLIPV:

FPSFLAGS: "FPS limit mode - bit 0 - limit fps (not higher than), bit 1 - maintain fps (not lower than)" /FPSFLAGS:

DCM_HOR: "Horizontal decimation of the image (as supported by the sensor). Actual number of pixels read from the senor will is divided (from the WOI size) by this value (0 considered to be the same as 1)" /DCM_HOR:

DCM_VERT: "Vertical decimation of the image (as supported by the sensor). Actual number of pixels read from the senor will is divided (from the WOI size) by this value (0 considered to be the same as 1)" /DCM_VERT:

BIN_HOR: "Horizontal binning (adding/averaging) of several adjacent pixels of the same color (so odd and even pixels are processed separately) as supported by the sensor." /BIN_HOR:

BIN_VERT: "Vertical binning (adding/averaging) of several adjacent pixels of the same color (so odd and even pixels are processed separately) as supported by the sensor." /BIN_VERT:

FPGATEST: "Replace the image from the sensor with the internally generated test pattern. Currently only two values are supported: 0 - npormal image, 1 horizontal gradient. /FPGATEST:

TESTSENSOR: "Sensor internal test modes: 0x10000 - enable, lower bits - test mode value" /TESTSENSOR:

COLOR: "Compressor modes (only modes 0..2 can be processed with standard libjpeg):`

0 - mono6, monochrome (color YCbCr 4:2:0 with zeroed out color componets)`

1 - color, YCbCr 4:2:0, 3x3 pixels`

2 - jp46 - original jp4, encoded as 4:2:0 with zeroed color components`

3 - jp46dc, modified jp46 so each color component uses individual DC diffenential encoding`

4 - reserved for color with 5x5 conversion (not yet implemented)`

5 - jp4 with ommitted color components (4:0:0)`

6 - jp4dc, similar to jp46dc encoded as 4:0:0`

7 - jp4diff, differential where (R-G), G, (G2-G) and (B-G) components are encoded as 4:0:0`

8 - jp4hdr, (R-G), G, G2,(B-G) are encoded so G2 can be used with high gain`

9 - jp4fiff2, (R-G)/2, G,(G2-G)/2, (B-G)/2 to avoid possible overflow in compressed values`

10 - jp4hdr2, (R-G)/2, G,G2,(B-G)/2`

14 - mono, monochrome with ommitted color components (4:0:0)" /COLOR:

FRAMESYNC_DLY: "not used, should be 0" /FRAMESYNC_DLY:

PF_HEIGHT: "Height of the strip in photofinish mode (not functional)" /PF_HEIGHT:

BITS: "data width stored from the sensor - can be either 8 or 16. 16 bit mode bypasses gamma-conversion, but it is not supported by the compressor" /BITS:

SHIFTL: "not used, should be 0" /SHIFTL:

FPNS: "FPN correction subtract scale - not yet supported by current software" /FPNS:

FPNM: "FPN correction multiply scale - not yet supported by current software" /FPNM:

VEXPOS: "Exposure measured in sensor native units - number of scan lines, it can be any integer number, while the EXPOS measured in microseconds is modified by the driver to make it multiple of scan lines. Both VEXPOS and EXPOS can be used to specify exposure." /VEXPOS:

VIRTTRIG: "Not used, should be 0" /VIRTTRIG:

PERIOD_MIN: Minimal frame period (in pixel clock cycles) calculated by the driver from the user and hardware limitations (readonly) /PERIOD_MIN:

PERIOD_MAX: Maximal frame period (in pixel clock cycles) calculated by the driver from the user and hardware limitations (readonly) /PERIOD_MAX:

SENSOR_PIXH: Pixels to be read from the sensor, horizontal,incliding margins, excluding embedded timestamps (readonly) /SENSOR_PIXH:

SENSOR_PIXV: Pixels to be read from the sensor, vertical, incliding margins (readonly) /SENSOR_PIXV:

GAINR: "Red channel sensor overall (analog and digital) gain multiplied by 0x10000, so 0x10000 corresponds to x1.0 gain. If ANA_GAIN_ENABLE is enabled this overall gain is split between the sensor analog gain and digital scalining. Digital scaling is needed to fill the gaps in between large analog gain steps." /GAINR:

GAING: "Green channel sensor overall (analog and digital) gain multiplied by 0x10000, so 0x10000 corresponds to x1.0 gain. If ANA_GAIN_ENABLE is enabled this overall gain is split between the sensor analog gain and digital scalining. Digital scaling is needed to fill the gaps in between large analog gain steps. Green channel is used in automatic exposure adjustment and as reference to differencial color gains. When changing the value of GAING other gains will be changed proportionally if their ratios to green are preserved (see RSCALE, GSCALE, BSCALE" /GAING:

GAINGB: "Second green (GB - green in blue line) channel sensor overall (analog and digital) gain multiplied by 0x10000, so 0x10000 corresponds to x1.0 gain. Normally the second green channel is programmed to have the same gain as the first green, but can be used separately for HDR applications. If ANA_GAIN_ENABLE is enabled this overall gain is split between the sensor analog gain and digital scalining. Digital scaling is needed to fill the gaps in between large analog gain steps." /GAINGB:

GAINB: "Blue channel sensor overall (analog and digital) gain multiplied by 0x10000, so 0x10000 corresponds to x1.0 gain. If ANA_GAIN_ENABLE is enabled this overall gain is split between the sensor analog gain and digital scalining. Digital scaling is needed to fill the gaps in between large analog gain steps." /GAINB:

RSCALE_ALL: "Combines RSCALE and RSCALE_CTL data" /RSCALE_ALL:

GSCALE_ALL: "Combines GSCALE and GSCALE_CTL data" /GSCALE_ALL:

BSCALE_ALL: "Combines BSCALE and BSCALE_CTL data" /BSCALE_ALL:

RSCALE: "Ratio of gains in Red and Green (base) colors, multiplied by 0x10000. This value is connected to individual gains: GAINR and GAING, when you change RSCALE it will cause GAINR to be updated also (if RSCALE is not disabled in RSCALE_CTL). When GAINR is changed, this RSCALE value may also change (or not - depending on the RSCALE_CTL)" /RSCALE:

GSCALE: "Ratio of gains in Green2 and Green (base) colors, multiplied by 0x10000. This value is connected to individual gains: GAINGB and GAING, when you change GSCALE it will cause GAINGB to be updated also (if GSCALE is not disabled in GSCALE_CTL). When GAINGB is changed, this GSCALE value may also change (or not - depending on the GSCALE_CTL). This second green scale should normally have the value 0x10000 (1.0) - it may be different only in some HDR modes." /GSCALE:

BSCALE: "Ratio of gains in Blue and Green (base) colors, multiplied by 0x10000. This value is connected to individual gains: GAINB and GAING, when you change BSCALE it will cause GAINB to be updated also (if BSCALE is not disabled in BSCALE_CTL). When GAINB is changed, this BSCALE value may also change (or not - depending on the BSCALE_CTL)" /BSCALE:

RSCALE_CTL: "A 2-bit RSCALE control. The following constants are defined:`

ELPHEL_CONST_CSCALES_CTL_NORMAL - use RSCALE to update GAINR and be updated when GAINR is changed`

ELPHEL_CONST_CSCALES_CTL_RECALC - recalculate RSCALE from GAINR/GAING once, then driver will modify the RSCALE_CTL value to ELPHEL_CONST_CSCALES_CTL_NORMAL`

ELPHEL_CONST_CSCALES_CTL_FOLLOW - update RSCALE from GAINR/GAING, but ignore any (external to the driver) changes to RSCALE itself`

ELPHEL_CONST_CSCALES_CTL_DISABLE - completely disable RSCALE - do not update it from GAINR and ignore any external changes to RSCALE`" /RSCALE_CTL:

GSCALE_CTL: "A 2-bit GSCALE control. The following constants are defined:`

ELPHEL_CONST_CSCALES_CTL_NORMAL - use GSCALE to update GAINGB and be updated when GAINGB is changed`

ELPHEL_CONST_CSCALES_CTL_RECALC - recalculate GSCALE from GAINGB/GAING once, then driver will modify the GRSCALE_CTL value to ELPHEL_CONST_CSCALES_CTL_NORMAL`

ELPHEL_CONST_CSCALES_CTL_FOLLOW - update GSCALE from GAINGB/GAING, but ignore any (external to the driver) changes to GSCALE itself`

ELPHEL_CONST_CSCALES_CTL_DISABLE - completely disable GSCALE - do not update it from GAING and ignore any external changes to GSCALE`" /GSCALE_CTL:

BSCALE_CTL: "A 2-bit BSCALE control. The following constants are defined:`

ELPHEL_CONST_CSCALES_CTL_NORMAL - use BSCALE to update GAINB and be updated when GAINB is changed`

ELPHEL_CONST_CSCALES_CTL_RECALC - recalculate BSCALE from GAINB/GAING once, then driver will modify the BSCALE_CTL value to ELPHEL_CONST_CSCALES_CTL_NORMAL`

ELPHEL_CONST_CSCALES_CTL_FOLLOW - update BSCALE from GAINB/GAING, but ignore any (external to the driver) changes to BSCALE itself`

ELPHEL_CONST_CSCALES_CTL_DISABLE - completely disable BSCALE - do not update it from GAINB and ignore any external changes to BSCALE`" /BSCALE_CTL:

FATZERO: "not used" /FATZERO:

QUALITY: "JPEG compression quality in percents. Supports individual setting of the Y and C quantization tables and quality values when the second byte is non-zero. Table zero is used always used for Y components (and all JP4/JP46 ones), table for C components is determined by the bit 15. When this bit is 0 the color quantization table (table 1) is used, when it is one - Y quantization table (table 0). If bits 8..14 are all zero, the same quality is used for both Y and C (with Y and C tables, respectively). Bit 7 determins if the standard JPEG table is used (bit7==0) or transposed one for portrait mode (bit7=1)." /QUALITY:

PORTRAIT: "JPEG quantization tables are optimezed for human perception when the scan lines are horizontal. If the value of PORTRAIT parameter is odd, these tables are transposed to be optimal for vertical scan lines. 0 - landscape with first line on top, 1 first line oin right, 2 - fierst line on the bottom and 3 - first line on the left. " /PORTRAIT:

CORING_INDEX: "Combined coring index for Y and C components - MSW - for C, LSW - for Y. Each is in the range 0f 0..99, default is 0x50005 (5/5). Highter values reduce noise (and file size) but can cause compression artifacts. The optimal values depend on compression quality, the higher the quality the larger the coring idexes are needed. Index 10 corresponds to 1.0 in quantized DCT coefficients, coefficients below index/10 are effectively zeroed out." /CORING_INDEX:

FP1000S: "Current sensor frame rate measured in frames per 1000 seconds" /FP1000S:

SENSOR_WIDTH: Sensor width in pixels (readonly) /SENSOR_WIDTH:

SENSOR_HEIGHT: Sensor height in pixels (readonly) /SENSOR_HEIGHT:

COLOR_SATURATION_BLUE: "Saturation of blue color (B-G) in percents. This scale value is used in the Bayer-to-YCbCr converter that feeds the JPEG compressor. Normally the saturation should be more than 100 to compensate the color washout when the gamma correction value is less than 1.0, because the gamma correction (which is applied to the raw Bayer pixel data) decrease relative (divided by the full value) difference between color components" /COLOR_SATURATION_BLUE:

COLOR_SATURATION_RED: "Saturation of red color (R-G) in percents. This scale value is used in the Bayer-to-YCbCr converter that feeds the JPEG compressor. Normally the saturation should be more than 100 to compensate the color washout when the gamma correction value is less than 1.0, because the gamma correction (which is applied to the raw Bayer pixel data) decrease relative (divided by the full value) difference between color components" /COLOR_SATURATION_RED:

VIGNET_AX: "AX in AX*X^2+BX*X+AY*Y^2+BY*Y+C" /VIGNET_AX:

VIGNET_AY: "AY in AX*X^2+BX*X+AY*Y^2+BY*Y+C" /VIGNET_AY:

VIGNET_BX: "BX in AX*X^2+BX*X+AY*Y^2+BY*Y+C" /VIGNET_BX:

VIGNET_BY: "BY in AX*X^2+BX*X+AY*Y^2+BY*Y+C" /VIGNET_BY:

VIGNET_C: "C in AX*X^2+BX*X+AY*Y^2+BY*Y+C" /VIGNET_C:

VIGNET_SHL: "Additional shift left of the vignetting correction multiplied by digital gain. Default 1" /VIGNET_SHL:

SCALE_ZERO_IN: "Will be subtracted from the 16-bit unsigned scaled sensor data before multiplying by vignetting correction and color balancing scale. It is a 17-bit signed data" /SCALE_ZERO_IN:

SCALE_ZERO_OUT: "Will be added to the result of multiplication of the 16-bit sennsor data (with optionally subtracted SCALE_ZERO_IN) by color correction coefficient/vignetting correction coefficient" /SCALE_ZERO_OUT:

DGAINR: "&quot;Digital gain&quot; for the red color channel - 17 bit unsigned value. Default value is 0x8000 fro 1.0, so up to 4X gain boost is available before saturation" /DGAINR:

DGAING: "&quot;Digital gain&quot; for the green color channel - 17 bit unsigned value. Default value is 0x8000 fro 1.0, so up to 4X gain boost is available before saturation" /DGAING:

DGAINGB: "&quot;Digital gain&quot; for second green color channel - 17 bit unsigned value. Default value is 0x8000 fro 1.0, so up to 4X gain boost is available before saturation" /DGAINGB:

DGAINB: "&quot;Digital gain&quot; for the blue color channel - 17 bit unsigned value. Default value is 0x8000 fro 1.0, so up to 4X gain boost is available before saturation" /DGAINB:

CORING_PAGE: "Number of coring LUT page number. Current software programs only page 0 (of 8) using CORING_INDEX parameter." /CORING_PAGE:

TILES: Number of 16x16 (20x20) tiles in a compressed frame (readonly) /TILES:

SENSOR_PHASE: "Sensor phase adjusment, packed, low 16 bit - signed fine phase, bits [18:17] - 90-degrees shift" /SENSOR_PHASE:

TEMPERATURE_PERIOD: "Period of sesnor temperature measurements, ms" /TEMPERATURE_PERIOD:

AUTOEXP_ON: "1 - autoexposure enabled when, 0 - autoexpousre disabled. Autoexposure can still be off if the bit responsible for autoexposure daemon in DAEMON_EN is turned off - in the latter case the whole autoexposure daemon will be disabled, including white balancing and hdr mode also." /AUTOEXP_ON:

HISTWND_RWIDTH: "Histogram (used for autoexposure, white balancing and just histograms display) window width, relative to the window (WOI) width. It is defined as a fraction of 65536(0x10000), so 0x8000 is 50%" /HISTWND_RWIDTH:

HISTWND_RHEIGHT: "Histogram (used for autoexposure, white balancing and just histograms display) window height, relative to the window (WOI) height. It is defined as a fraction of 65536(0x10000), so 0x8000 is 50%" /HISTWND_RHEIGHT:

HISTWND_RLEFT: "Histogram (used for autoexposure, white balancing and just histograms display) window left position, relative to the window (WOI) remaining (after HISTWND_RWIDTH). It is defined as a fraction of 65536(0x10000), so when HISTWND_RLEFT=0x8000 and HISTWND_RWIDTH=0x8000 that will put histogram window in the center 50% leaving 25% from each of the left and right WOI limits" /HISTWND_RLEFT:

HISTWND_RTOP: "Histogram (used for autoexposure, white balancing and just histograms display) window top position, relative to the window (WOI) remaining (after HISTWND_RHEIGHT). It is defined as a fraction of 65536(0x10000), so when HISTWND_RTOP=0x8000 and HISTWND_RHEIGHT=0x8000 that will put histogram window vertically in the center 50% leaving 25% from each of the top and bottom WOI limits" /HISTWND_RTOP:

AUTOEXP_EXP_MAX: "Maximal exposure time allowed to autoexposure daemon (in microseconds)" /AUTOEXP_EXP_MAX:

AUTOEXP_OVEREXP_MAX: "not used" /AUTOEXP_OVEREXP_MAX:

AUTOEXP_S_PERCENT: "not used" /AUTOEXP_S_PERCENT:

AUTOEXP_S_INDEX: "not used" /AUTOEXP_S_INDEX:

AUTOEXP_EXP: "not used" /AUTOEXP_EXP:

AUTOEXP_SKIP_PMIN: "not used" /AUTOEXP_SKIP_PMIN:

AUTOEXP_SKIP_PMAX: "not used" /AUTOEXP_SKIP_PMAX:

AUTOEXP_SKIP_T: "not used" /AUTOEXP_SKIP_T:

HISTWND_WIDTH: "Histogram (used for autoexposure, white balancing and just histograms display) window width in pixels (readonly)" /HISTWND_WIDTH:

HISTWND_HEIGHT: "Histogram (used for autoexposure, white balancing and just histograms display) window height in pixels (readonly)" /HISTWND_HEIGHT:

HISTWND_TOP: "Histogram (used for autoexposure, white balancing and just histograms display) window top position in pixels (readonly)" /HISTWND_TOP:

HISTWND_LEFT: "Histogram (used for autoexposure, white balancing and just histograms display) window left position in pixels (readonly)" /HISTWND_LEFT:

FOCUS_SHOW: "Show focus information instead of/combined with the image: 0 - regular image, 1 - block focus instead of Y DC (AC=0), 2 - image Y DC combined all frame, 3 combined in WOI" /FOCUS_SHOW:

FOCUS_SHOW1: "Additional parameter that modifies visualization mode. Currently just a single bit (how much to add)" /FOCUS_SHOW1:

RFOCUS_LEFT: "init" /RFOCUS_LEFT:

RFOCUS_WIDTH: "init" /RFOCUS_WIDTH:

RFOCUS_TOP: "init" /RFOCUS_TOP:

RFOCUS_HEIGHT: "init" /RFOCUS_HEIGHT:

FOCUS_LEFT: "Focus WOI left margin, in pixels, inclusive (3 LSB will be zeroed as it should be multiple of 8x8 block width)" /FOCUS_LEFT:

FOCUS_WIDTH: "Focus WOI width (3 LSB will be zeroed as it should be multiple of 8x8 block width)" /FOCUS_WIDTH:

FOCUS_TOP: "focus WOI top margin, inclusive (3 LSB will be zeroed as it should be multiple of 8x8 block height)" /FOCUS_TOP:

FOCUS_HEIGHT: "Focus WOI height (3 LSB will be zeroed as it should be multiple of 8x8 block height)" /FOCUS_HEIGHT:

FOCUS_TOTWIDTH: "Total width of the image frame in pixels (readonly)" /FOCUS_TOTWIDTH:

FOCUS_FILTER: "Select 8x8 filter used for the focus calculation (same order as quantization coefficients), 0..14" /FOCUS_FILTER:

TRIG_CONDITION: "FPGA trigger sequencer trigger condition, 0 - internal, else dibits ((use&lt;&lt;1) | level) for each GPIO[11:0] pin). Example:0x200000 - input from external connector (J15 - http://wiki.elphel.com/index.php?title=10369#J15_-_SYNC_.28external.29), 0x20000 - input from internal (J13/J14 - http://wiki.elphel.com/index.php?title=10369#J13_-_SYNC_.28internal.2C_slave.29)" /TRIG_CONDITION:

TRIG_DELAY: "FPGA trigger sequencer trigger delay, 32 bits in pixel clocks" /TRIG_DELAY:

TRIG_OUT: "FPGA trigger sequencer trigger output to GPIO, dibits ((use &lt;&lt; 1) | level_when_active). Bit 24 - test mode, when GPIO[11:10] are controlled by other internal signals. Example: 0x800000 - output to external (J15 - http://wiki.elphel.com/index.php?title=10369#J15_-_SYNC_.28external.29) connector, 0x80000 - to internal (J12 - http://wiki.elphel.com/index.php?title=10369#J12_-_SYNC_.28internal.2C_master.29)" /TRIG_OUT:

TRIG_PERIOD: "FPGA trigger sequencer output sync period (32 bits, in pixel clocks). 0- stop. 1 - single, : =256 repetitive with specified period" /TRIG_PERIOD:

TRIG_BITLENGTH: "Bit length minus 1 (in pixel clock cycles) when transmitting/receiving timestamps, without timestamps the output pulse width is 8*(TRIG_BITLENGTH+1). Legal values 2..255." /TRIG_BITLENGTH:

EXTERN_TIMESTAMP: "When 1 camera will use external timestamp (received over inter-camera synchronization cable) if it is available (no action when external syncronization is not connected), when 0 - local timestamp will be used" /EXTERN_TIMESTAMP:

XMIT_TIMESTAMP: "Specify output signal sent through internal/external connector (defined by TRIG_OUT). 0 - transmit just sync pulse (8*(TRIG_BITLENGTH+1) pixel clock periods long), 1 - pulse+timestamp 64*(TRIG_BITLENGTH+1) pixel clock periods long" /XMIT_TIMESTAMP:

SKIP_FRAMES: "Changes parameter latencies tables pages for each of the trigger modes separately (0/1), currently should be 0" /SKIP_FRAMES:

I2C_QPERIOD: "Number of system clock periods in 1/4 of i2c SCL period to the sensor/sensor board, set by the driver" /I2C_QPERIOD:

I2C_BYTES: "Number of bytes in hardware i2c write (after slave addr) -0/1/2, set by the driver" /I2C_BYTES:

IRQ_SMART: "IRQ mode (3 bits) to combine interrupts from the sensor frame sync and compressor when it is running: +1 - wait for VACT in early compressor_done, +2 - wait for dma fifo ready. Current software assumes both bits are set (value=3), set up by the driver". Currently bit 2 (+4) needs to be set to 1 when bit 0 is 0 - otherwise the latest frame will not have parameters copied to. So instead of IRQ_SMART=2 it should be IRQ_SMART=6 /IRQ_SMART:

OVERSIZE: "0 - normal mode, 1 - ignore sensor dimensions, use absolute WOI_LEFT, WOI_TOP - needed to be able to read optically black pixels" /OVERSIZE:

GTAB_R: "Identifies Gamma-table for the red color. Camera can use either automatically generated tables using the provided black level and gamma (in percent) or arbitrary custom tables, in that case the top 16 bits are used as a 16-bit hash (user provided) to distinguish between different loaded tables. The lower 16 bits determine scale applied to the table (saturated to the full scale), so the value is (black_level&lt;&lt;24) | (gamma_in_percent &lt;&lt;16) | (scale_times_0x1000 &amp; 0xffff). In PHP (or PHP scripts) the individual fields of GTAB_R can be referenced with composite names like GTAB_R__0824 for black level, GTAB_R__0816 for gamma in percents and GTAB_R__1600 for scale. /GTAB_R:

GTAB_G: "Identifies Gamma-table for the green color. Camera can use either automatically generated tables using the provided black level and gamma (in percent) or arbitrary custom tables, in that case the top 16 bits are used as a 16-bit hash (user provided) to distinguish between different loaded tables. The lower 16 bits determine scale applied to the table (saturated to the full scale), so the value is (black_level&lt;&lt;24) | (gamma_in_percent &lt;&lt;16) | (scale_times_0x1000 &amp; 0xffff). In PHP (or PHP scripts) the individual fields of GTAB_G can be referenced with composite names like GTAB_G__0824 for black level, GTAB_G__0816 for gamma in percents and GTAB_G__1600 for scale. /GTAB_G:

GTAB_GB: "Identifies Gamma-table for the second green (in blue line) color. Camera can use either automatically generated tables using the provided black level and gamma (in percent) or arbitrary custom tables, in that case the top 16 bits are used as a 16-bit hash (user provided) to distinguish between different loaded tables. The lower 16 bits determine scale applied to the table (saturated to the full scale), so the value is (black_level&lt;&lt;24) | (gamma_in_percent &lt;&lt;16) | (scale_times_0x1000 &amp; 0xffff). In PHP (or PHP scripts) the individual fields of GTAB_GB can be referenced with composite names like GTAB_GB__0824 for black level, GTAB_GB__0816 for gamma in percents and GTAB_GB__1600 for scale. /GTAB_GB:

GTAB_B: "Identifies Gamma-table for the blue color. Camera can use either automatically generated tables using the provided black level and gamma (in percent) or arbitrary custom tables, in that case the top 16 bits are used as a 16-bit hash (user provided) to distinguish between different loaded tables. The lower 16 bits determine scale applied to the table (saturated to the full scale), so the value is (black_level&lt;&lt;24) | (gamma_in_percent &lt;&lt;16) | (scale_times_0x1000 &amp; 0xffff). In PHP (or PHP scripts) the individual fields of GTAB_B can be referenced with composite names like GTAB_B__0824 for black level, GTAB_B__0816 for gamma in percents and GTAB_B__1600 for scale. /GTAB_B:

SDRAM_CHN20: "Internal value used by the driver (memory controller register 0 channel 2)" /SDRAM_CHN20:

SDRAM_CHN21: "Internal value used by the driver (memory controller register 1 channel 2)" /SDRAM_CHN21:

SDRAM_CHN22: "Internal value used by the driver (memory controller register 2 channel 2)" /SDRAM_CHN22:

COMPRESSOR_RUN: "Compressor state: 0 - stopped, 1 - compress single frame, 2 - run continuously. Some applications (streamer, videorecorder) rely on this register to be set to 2" /COMPRESSOR_RUN:

COMPRESSOR_SINGLE: "Pseudo-register to write COMPERRSOR_RUN_SINGLE here. Same as COMPRESSOR_RUN, just the command will not propagate to the next frames" /COMPRESSOR_SINGLE:

COMPMOD_BYRSH: "Additional bayer shift in compressor only (to swap meanings of the colors), 0..3" /COMPMOD_BYRSH:

COMPMOD_TILSH: "Diagonal shift of the 16x16 pixel block in the 20x20 tile that compressor receives (0 - top left corner is (0,0), ..., 4 - top left corner is (4,4))" /COMPMOD_TILSH:

COMPMOD_DCSUB: "Subtract average block pixel value before DCT and add it back after" /COMPMOD_DCSUB:

COMPMOD_QTAB: "Quantization table bank number (set by the driver)" /COMPMOD_QTAB:

SENSOR_REGS: Sensor internal registers (sensor-specific). In PHP scripts it is possible to reference individual register/bit fields with composite names, i.e. SENSOR_REGS160__0403 in Micron MT9P031 sensor allows to edit test patter number - bits 3..6 of the sensor register 160 (0xa0). There is additional suffix availble in multi-sesnor cameras. Some parameters may have different values for different sensor, in that case __A (and __a) reference register of sensor 1, __B (__b) and __C (__c) - sensors 2 and 3. Parametes with upper case (__A, __B and __C) will reference the base parameter if individual is not defined, low case suffixes are strict and return error if the parameter does not have individual values for sensors. /SENSOR_REGS:

DAEMON_EN: "Controls running daemons (individually turns them on/off by setting/resetting the related bit). It is more convinient to control them as individual bits using defined composite parameters, like DAEMON_EN_AUTOEXPOSURE, DAEMON_EN_STREAMER, etc." /DAEMON_EN:

DAEMON_EN_AUTOEXPOSURE: "0 - turns autoexposure daemon off, 1 - on. When off - not just autoexposure, but also white balance and HDR are disabled" /DAEMON_EN_AUTOEXPOSURE:

DAEMON_EN_STREAMER: "0 - turns the videostreamer off, 1 - on." /DAEMON_EN_STREAMER:

DAEMON_EN_CCAMFTP: "0 - turns the FTP uploader off, 1 - on. (not yet implemented)" /DAEMON_EN_CCAMFTP:

DAEMON_EN_CAMOGM: "0 - turns videorecording off, 1 - on. (not yet implemented)" /DAEMON_EN_CAMOGM:

DAEMON_EN_TEMPERATURE: "0 - turns temperature logging off, 1 - on" /DAEMON_EN_TEMPERATURE:

DAEMON_EN_AUTOCAMPARS: "when set to 1 autocampars daemon will wake up, launch autocampars.php script (that will actually process the provided command of saving/restoring parameters from the file) and goes back to sleep by clearing this bit by itself." /DAEMON_EN_AUTOCAMPARS:

AEXP_FRACPIX: "Fraction of all pixels that should be below P_AEXP_LEVEL (16.16 - 0x10000 - all pixels)" /AEXP_FRACPIX:

AEXP_LEVEL: "Target output level: [P_AEXP_FRACPIX]/0x10000 of all pixels should have value below it (also 16.16 - 0x10000 - full output scale)" /AEXP_LEVEL:

HDR_DUR: "0 - HDR 0ff, : 1 - duration of same exposure (currently 1 or 2 - for free running)" /HDR_DUR:

HDR_VEXPOS: "Second exposure setting in alternating frames HDR mode. if it is less than 0x10000 - number of lines of exposure, : =10000 - relative to "normal" exposure" /HDR_VEXPOS:

EXP_AHEAD: "How many frames ahead of the current frame write exposure to the sensor" /EXP_AHEAD:

AE_THRESH: "Autoexposure error (logariphmic difference between calculated and current exposures) is integrated between frame and corrections are scaled when error is below thershold." /AE_THRESH:

WB_THRESH: "White balance error (logariphmic difference between calculated and current values) is integrated between frame and corrections are scaled when error is below thershold (not yet implemented)" /WB_THRESH:

AE_PERIOD: "Autoexposure period (will be increased if below the latency)" /AE_PERIOD:

WB_PERIOD: "White balance period (will be increased if below the latency)" /WB_PERIOD:

WB_CTRL: "Combines WB_CTRL and WB_EN fields" /WB_CTRL:

WB_MASK: "Bitmask - which colors to adjust (1 - adjust, 0 - keepe). Default on is 0xd - all colors but Green1" /WB_MASK:

WB_EN: "Enable (1) or disable (0) automatic white balance adjustment. When enabled each color is individually controlled by WB_MASK" /WB_EN:

WB_WHITELEV: "White balance level of white (16.16 - 0x10000 is full scale, 0xfae1 - 98%, default)" /WB_WHITELEV:

WB_WHITEFRAC: "White balance fraction (16.16) of all pixels that have level above [P_WB_WHITELEV] for the brightest color [P_WB_WHITELEV] will be decreased if needed to satisfy [P_WB_WHITELEV]. default is 1% (0x028f)" /WB_WHITEFRAC:

WB_MAXWHITE: "Maximal allowed white pixels fraction (16.16) to have level above [WB_WHITELEV] for the darkest color of all. If this limit is exceeded there will be no correction performed (waiting for autoexposure to decrease overall brightness)." /WB_MAXWHITE:

WB_SCALE_R: "Additional correction for red/green from calulated by white balance. 0x10000 - 1.0 (default)" /WB_SCALE_R:

WB_SCALE_GB: "Additional correction for green2/green from calulated by white balance. 0x10000 - 1.0 (default). May be used for the color HDR mode" /WB_SCALE_GB:

WB_SCALE_B: "Additional correction for blue/green from calulated by white balance. 0x10000 - 1.0 (default)" /WB_SCALE_B:

HISTRQ: "Single histogram calculation request address (used automatically)" /HISTRQ:

HISTRQ_Y: "Single histogram calculation request for Y (green1) histogram (used automatically)" /HISTRQ_Y:

HISTRQ_C: "Single histogram calculation request for C (red, green2, blue) histograms (used automatically)" /HISTRQ_C:

HISTRQ_YC: "Single histogram calculation request for Y and C (red, green, green2, blue) histograms (used automatically)" /HISTRQ_YC:

PROFILE: "index to access profiles as pastpars (i.e. from PHP ELPHEL_PROFILE1,PHP ELPHEL_PROFILE2)" /PROFILE:

GAIN_MIN: "Minimal sensor analog gain (0x10000 - 1.0) - used for white balancing. May be user limited from the hardware capabilities." /GAIN_MIN:

GAIN_MAX: "Maximal sensor analog gain (0x10000 - 1.0) - used for white balancing. May be user limited from the hardware capabilities." /GAIN_MAX:

GAIN_CTRL: "Analog gain control for white balance. Combines GAIN_STEP and ANA_GAIN_ENABLE" /GAIN_CTRL:

GAIN_STEP: "minimal correction to be applied to the analog gain (should be set larger that sensor actual gain step to prevent oscillations (0x100 - 1.0, 0x20 - 1/8)" /GAIN_STEP:

ANA_GAIN_ENABLE: "Enabling analog gain control in white balancing (it uses scaling in gamma tables for fine adjustments and may additionally adjust analog gains if this value is 1" /ANA_GAIN_ENABLE:

AUTOCAMPARS_CTRL: "Input patrameter for the autocampars daemon to execute when enabled: bits 0..24 - parameter groups to restore, bits 28..30: 1 - restore, 2 - save, 3 - set default 4 save as default 5 - init. Combines AUTOCAMPARS_GROUPS, AUTOCAMPARS_PAGE and AUTOCAMPARS_CMD" /AUTOCAMPARS_CTRL:

AUTOCAMPARS_GROUPS: "Input patrameter for the autocampars daemon to execute when enabled: each of the 24 bits enables restoration of the related parameter group" /AUTOCAMPARS_GROUPS:

AUTOCAMPARS_PAGE: Input patrameter for the autocampars daemon to execute when enabled - page number to use to save/restore parameters. 0..14 - absolute page number, 15 - default when reading, next after last saved - when writing (15 will be replaced by the particular number by autocampars, so that value can be read back /AUTOCAMPARS_PAGE:

AUTOCAMPARS_CMD: Commands for the autocampars daemon to execute (to use from PHP - add ELPHEL_CONST_ to the name):`

1 - AUTOCAMPARS_CMD_RESTORE - restore specified groups of parameters from the specified page`

2 - AUTOCAMPARS_CMD_SAVE - save all current parameters to the specified group (page 0 is write-protected)`

3 - AUTOCAMPARS_CMD_DFLT - make selected page the default one (used at startup)`

4 - AUTOCAMPARS_CMD_SAVEDFLT - save all current parameters to the specified group (page 0 is write-protected) and make it default (used at startup)`

5 - AUTOCAMPARS_CMD_INIT - reset sensor/sequencers, restore all parameters from the specified page /AUTOCAMPARS_CMD:

FTP_PERIOD: "Desired period of image upload to the remote FTP server (seconds)" /FTP_PERIOD:

FTP_TIMEOUT: "Maximal waiting time for the image to be uploaded to the remote server" /FTP_TIMEOUT:

FTP_UPDATE: "Maximal time between updates (camera wil re-read remote configuration file)" /FTP_UPDATE:

FTP_NEXT_TIME: "Sheduled time of the next FTP upload in seconds from epoch (G_ parameter)" /FTP_NEXT_TIME:

MAXAHEAD: "Maximal number of frames ahead of current to be programmed to hardware" /MAXAHEAD:

THIS_FRAME: "Current absolute frame number (G_ parameter, readonly)" /THIS_FRAME:

CIRCBUFSIZE: "Circular video buffer size in bytes (G_ parameter, readonly)" /CIRCBUFSIZE:

FREECIRCBUF: "Free space in the circular video buffer in bytes - only make sense when used with the global read pointer CIRCBUFRP (G_ parameter, readonly)" /FREECIRCBUF:

CIRCBUFWP: "Circular video buffer write pointer - where the next acquired frame will start(G_ parameter, readonly)" /CIRCBUFWP:

CIRCBUFRP: "Circular video buffer (global) read pointer. Used for synchronization between applications (i.e. reduce the streamer CPU load/fps if video recorder is not keeping up with the incoming data (G_ parameter)" /CIRCBUFRP:

FRAME_SIZE: "Size of the last compressed frame in bytes (w/o Exif and JFIF headers)" /FRAME_SIZE:

SECONDS: "Buffer used to read/write FPGA real time timer, seconds from epoch (G_ parameter)" /SECONDS:

MICROSECONDS: "Buffer used to read/write FPGA real time timer, microseconds (G_ parameter)" /MICROSECONDS:

CALLNASAP: "Bit mask of the internal tasks that can use FPGA sequencer - can be modified with parseq.php (G_ parameter)" /CALLNASAP:

CALLNEXT: "Four registers (CALLNEXT1..CALLNEXT4) that specify latencies of the internal tasks - can be modified with parseq.php (G_ parameters)" /CALLNEXT:

NEXT_AE_FRAME: "Next frame when autoexposure is scheduled (G_ parameter)" /NEXT_AE_FRAME:

NEXT_WB_FRAME: "Next frame when white balancing is scheduled (G_ parameter)" /NEXT_WB_FRAME:

HIST_DIM_01: "Zero levels (on 0xffff scale) for red and green1 color components for white balancing and autoexposure (G_ parameter)" /HIST_DIM_01:

HIST_DIM_23: "Zero levels (on 0xffff scale) for green2 and blue color components for white balancing and autoexposure (G_ parameter)" /HIST_DIM_23:

AE_INTEGERR: /AE_INTEGERR:

WB_INTEGERR: /WB_INTEGERR:

TASKLET_CTL: "Tasklet control, parent to HISTMODE_Y, HISTMODE_C and additionally:`

bit 0 (TASKLET_CTL_PGM) - disable programming parameters (should not be)`

Bit 1 (TASKLET_CTL_IGNPAST) - ignore overdue parameters`

Bit 2 (TASKLET_CTL_NOSAME) - do not try to process parameters immediately after being written. If 0, only non-ASAP will be processed`

Bit 3 (TASKLET_CTL_ENPROF) - enable profiling (saving timing of the interrupts/tasklets in pastpars) - can be controlled through PROFILING_EN (G_parameter)" /TASKLET_CTL:

GFOCUS_VALUE: "Sum of all blocks focus values inside focus WOI (G_ parameter,readonly)" /GFOCUS_VALUE:

HISTMODE_Y: "Controls when the Y (green1) histograms are calcuted:`

0 - TASKLET_HIST_ALL - calculate each frame`

1 - TASKLET_HIST_HALF calculate each even (0,2,4,6 frame of 8)`

2 - TASKLET_HIST_QUATER - calculate twice per 8 (0, 4)`

3 - TASKLET_HIST_ONCE - calculate once per 8 (0)`

4 - TASKLET_HIST_RQONLY - calculate only when specifically requested`

7 - TASKLET_HIST_NEVER - never calculate.`

NOTE: It is safer to allow all histograms at least once in 8 frames so applications will not be locked up waiting for the missed histogram (G_ parameter)" /HISTMODE_Y:

HISTMODE_C: "Controls when the C (red, green2, blue) histograms are calcuted:`

0 - TASKLET_HIST_ALL - calculate each frame`

1 - TASKLET_HIST_HALF calculate each even (0,2,4,6 frame of 8)`

2 - TASKLET_HIST_QUATER - calculate twice per 8 (0, 4)`

3 - TASKLET_HIST_ONCE - calculate once per 8 (0)`

4 - TASKLET_HIST_RQONLY - calculate only when specifically requested`

7 - TASKLET_HIST_NEVER - never calculate.`

NOTE: It is safer to allow all histograms at least once in 8 frames so applications will not be locked up waiting for the missed histogram (G_ parameter)" /HISTMODE_C:

SKIP_DIFF_FRAME: "Maximal number of frames of the different size streamer should skip before giving up - needed to allow acquisition of the full-frame images during streaming lower resolution ones(G_ parameter)" /SKIP_DIFF_FRAME:

HIST_LAST_INDEX: "Index of the last acquired histograms in the histogram cache (G_ parameter,readonly)" /HIST_LAST_INDEX:

HIST_Y_FRAME: "Frame number for which last Y (green1) histogram was calculated(G_ parameter,readonly)" /HIST_Y_FRAME:

HIST_C_FRAME: "Frame number for which last C (red, green2, blue) histogram was calculated(G_ parameter,readonly)" /HIST_C_FRAME:

DAEMON_ERR: "Bits from up to 32 daemons to report problems or requests (G_ parameter)" /DAEMON_ERR:

DAEMON_RETCODE: "32 locations - DAEMON_RETCODE0...DAEMON_RETCODE31 to get calues from the running daemons(G_ parameter)" /DAEMON_RETCODE:

PROFILING_EN: "Enable profiling (saving timing of the interrupts/tasklets in pastpars) - this is a single bit of the TASKLET_CTL parameter.(G_ parameter)" /PROFILING_EN:

STROP_MCAST_EN: "0 - disable videostreamer multicast, 1 - enable." /STROP_MCAST_EN:

STROP_MCAST_IP: "Videostream multicast IP. If == 0 - used addres 232.X.Y.Z, where X.Y.Z is last part of the camera IP" /STROP_MCAST_IP:

STROP_MCAST_PORT: "Videostream multicast port." /STROP_MCAST_PORT:

STROP_MCAST_TTL: "Videostream multicast TTL." /STROP_MCAST_TTL:

STROP_AUDIO_EN: "0 - disable videostream audio, 1 - enable." /STROP_AUDIO_EN:

STROP_AUDIO_RATE: "Videostream audio rate." /STROP_AUDIO_RATE:

STROP_AUDIO_CHANNEL: "Videostream audio channels: 1 - mono, 2 - stereo." /STROP_AUDIO_CHANNEL:

STROP_FRAMES_SKIP: "How many frames skip before next output; 0 - outpur each frame, 1 - output/skip = 1/1 for two frames, 2 - output frame and skip next 2 from 3 frames etc." /STROP_FRAMES_SKIP:

AUDIO_CAPTURE_VOLUME: "streamer and camogm2 audio capture volume. 0 == 0%, 65535 == 100%." /AUDIO_CAPTURE_VOLUME:

!--Parameters related to multi-sensor operations --:

MULTISENS_EN: "0 - single sensor, no 10359A, otherwise - bitmask of the sensors enabled (obeys G_SENS_AVAIL that should not be modified at runtime)." /MULTISENS_EN:

MULTI_PHASE_SDRAM: "Similar to SENSOR_PHASE, contols 10359 SDRAM. Adjusted automatically" /MULTI_PHASE_SDRAM:

MULTI_PHASE1: "Similar to SENSOR_PHASE, but for sensor1, connected to 10359" /MULTI_PHASE1:

MULTI_PHASE2: "Similar to SENSOR_PHASE, but for sensor2, connected to 10359" /MULTI_PHASE2:

MULTI_PHASE3: "Similar to SENSOR_PHASE, but for sensor3, connected to 10359" /MULTI_PHASE3:

MULTI_SEQUENCE: "Sensor sequence (bits 0,1 - first, 2,3 - second, 4,5 - third). 0 - disable. Will obey SENS_AVAIL and MULTISENS_EN" /MULTI_SEQUENCE:

MULTI_FLIPH: "additional per-sensor horizontal flip to global FLIPH, same bits as in G_SENS_AVAIL" /MULTI_FLIPH:

MULTI_FLIPV: "additional per-sensor vertical flip to global FLIPV, same bits as in G_SENS_AVAIL" /MULTI_FLIPV:

MULTI_MODE: "Mode 0 - single sensor (first in sequence), 1 - composite (only enabled in triggered mode - TRIG=4)" /MULTI_MODE:

MULTI_HBLANK: "Horizontal blanking for buffered frames (2,3) - not needed?" /MULTI_HBLANK:

MULTI_CWIDTH: "Composite frame width (stored while in single-sensor mode, copied to WOI_WIDTH)" /MULTI_CWIDTH:

MULTI_CHEIGHT: "Composite frame height (stored while in single-sensor mode)" /MULTI_CHEIGHT:

MULTI_CLEFT: "Composite frame left margin (stored while in single-sensor mode, copied to WOI_LEFT)" /MULTI_CLEFT:

MULTI_CTOP: "Composite frame top margin (stored while in single-sensor mode)" /MULTI_CTOP:

MULTI_CFLIPH: "Horizontal flip for composite image (stored while in single-sensor mode)" /MULTI_CFLIPH:

MULTI_CFLIPV: "Vertical flip for composite image (stored while in single-sensor mode)" /MULTI_CFLIPV:

MULTI_VBLANK: "Vertical blanking for buffered frames (2,3) BEFORE FRAME, not after" /MULTI_VBLANK:

MULTI_WOI: "Width of frame 1 (direct) // Same as next" /MULTI_WOI:

MULTI_WIDTH1: "Width of frame 1 (direct) // same as MULTI_WOI !!!!" /MULTI_WIDTH1:

MULTI_WIDTH2: "Width of frame 2 (first buffered)" /MULTI_WIDTH2:

MULTI_WIDTH3: "Width of frame 3 (second buffered)" /MULTI_WIDTH3:

MULTI_HEIGHT1: "Height of frame 1 (direct)" /MULTI_HEIGHT1:

MULTI_HEIGHT2: "Height of frame 2 (first buffered)" /MULTI_HEIGHT2:

MULTI_HEIGHT3: "Height of frame 3 (second buffered)" /MULTI_HEIGHT3:

MULTI_LEFT1: "Left margin of frame 1 (direct) " /MULTI_LEFT1:

MULTI_LEFT2: "Left margin of frame 2 (first buffered)" /MULTI_LEFT2:

MULTI_LEFT3: "Left margin of frame 3 (second buffered)" /MULTI_LEFT3:

MULTI_TOP1: "Top margin of frame 1 (direct)" /MULTI_TOP1:

MULTI_TOP2: "Top margin of frame 2 (first buffered)" /MULTI_TOP2:

MULTI_TOP3: "Top margin of frame 3 (second buffered)" /MULTI_TOP3:

MULTI_TOPSENSOR: "Number of sensor channel used in first (direct) frame: 0..2, internal parameter (window-: sensorin) - used internally" /MULTI_TOPSENSOR:

MULTI_SELECTED: "Number of sensor channel (1..3) used when composite mode is disabled" /MULTI_SELECTED:

M10359_REGS: 10359 board inrternal registers, total of 96. First 64 are 16-bit, next 32 - 32 bit wide (Register definitions in http://elphel.cvs.sourceforge.net/viewvc/elphel/elphel353-8.0/os/linux-2.6-tag--devboard-R2_10-4/arch/cris/arch-v32/drivers/elphel/multisensor.h?view=markup). /M10359_REGS:

!-- Global parameters , persistent through sensor (re-) detection TODO: move other ones here--:

SENS_AVAIL: "Bitmask of the sensors attached to 10359 (0 if no 10359 brd, multisensor operations disabled). It is automatically set during sensor detection." /SENS_AVAIL:

FPGA_TIM0: "FPGA timing parameter 0 - difference between DCLK pad and DCM input, signed (ps). Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /FPGA_TIM0:

FPGA_TIM1: "FPGA timing parameter 1. Persistent through initialization." /FPGA_TIM1:

DLY359_OUT: "Output delay in 10359 board (clock to out) in ps, signed. Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_OUT:

DLY359_P1: "Delay in 10359 board sensor port 1 (clock to sensor - clock to DCM) in ps, signed. Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_P1:

DLY359_P2: "Delay in 10359 board sensor port 2 (clock to sensor - clock to DCM) in ps, signed. Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_P2:

DLY359_P3: "Ddelay in 10359 board sensor port 3 (clock to sensor - clock to DCM) in ps, signed. Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_P3:

DLY359_C1: "Cable delay in sensor port 1 in ps, Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_C1:

DLY359_C2: "Cable delay in sensor port 2 in ps, signed. Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_C2:

DLY359_C3: "Cable delay in sensor port 3 in ps, signed. Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /DLY359_C3:

MULTI_CFG: "Additional configuration options for 10359 board. Bit 0 - use 10353 system clock, not the local one (as on 10359 rev 0). Persistent through sensor detection/initialization, should be set prior to it (in startup script or modified before running &quot;/usr/html/autocampars.php --init&quot; from the command line)." /MULTI_CFG:

DEBUG: "Selectively enables debug outputs from differnt parts of the drivers. Can easily lock the system as some output goes from inside the interrupt service code or from the parts of the code where interrups are disabled. To us it safely you need to kill the klog daemon an redirect debug output to file with &quot;printk_mod &amp;&quot; command. After that the output will be available as http://camera_ip/var/klog.txt". The printk_mod also kills restart restart daemon so any normally restarted applications (like lighttpd, php, imgsrv) will not be restarted automatically (G_ parameter, not frame-related) /DEBUG:

TEMPERATURE01: "Temperature data from the 10359 board (if available, lower 16 bits) and the first sensor front end (high 16 bits). In each short word bit 12 (0x1000) is set for negative Celsius, lower 12 bits - absolute value of the Celsius, lower bit weight is 1/16 grad. C. This data is provided by the temperature daemon if it is enabled and running, data is embedded in the Exif MakerNote bytes 56-59" /TEMPERATURE01:

TEMPERATURE23: "Temperature data from the second sensor front end (if available, lower 16 bits) and the third sensor front end (high 16 bits). In each short word bit 12 (0x1000) is set for negative Celsius, lower 12 bits - absolute value of the Celsius, lower bit weight is 1/16 grad. C. This data is provided by the temperature daemon if it is enabled and running, data is embedded in the Exif MakerNote bytes 56-59" /TEMPERATURE23: