GHI electronics

ALCAM SoC Processor User Manual

February 10, 2015

THIS IS A PRELIMINARY MANUAL. ALL CONTENT IS SUBJECT TO CHANGE.

Table of Contents

1. Introduction....4
2. Example Applications....5
3. Key Features....6
4. Architecture....7
  Commands....7
  FAT File-System Engine....7
  Memory Card Access (SDHC, SD or MMC)....7
  Bootloader....8
5. Pinout....9
6. ALCAM Command Set....14
  Terminology and Syntax of Command Definitions:....14
  Command Classes....17
  V – Get Firmware Version....17
  I – Initialization Commands (Camera, USB, SD)....19
  D – Power Control....21
  P – Picture (still shot) Control....22
    C – capture image to storage media: P C[>filename]....22
    R – capture Picture: P R 23
    T – capture image and thumbnail: P T[>filename]....24
  F – File Operations....25
    W – write bytes to a file:....26
    R – open and read bytes from a file: F R>filename....27
    C – close a file: F C....27
    S – get the size of a file: F S>filename....28
    D – delete a file: F D>filename....28
    G – Get last default filename: F G....28
  M – Movie Operations....30
    N – Set movie mode to Normal: M N....31
    S – Set movie mode to Stop Motion: M S[>fps]....31
    T – Set movie mode to Time Lapse: M T[>delay[>fps]]....31
    B – Begin recording the movie: M B[>filename]....33
    C – Capture a frame: M C....33
    E – End the movie: M E....34
    I – Get information on a movie file: M I>filename....34
  T – Time and Date Operations....35
    B – Use battery backed real-time clock: T B....35
    I – Use internal clock: T I....35
    S – Set date and time: T S>timestamp....36
    D - Get date and time: T G....36
  Q – Format Media....38
  C – Configure Camera....39
    S - Set frame/image size: C S>size....40
    B - Set brightness level: C B>brightness....40
    O - Set contrast level: C O>contrast....41
    A - Manual or automatic white-balance: C A>white balance....41
    E - Special effect processing: C E>special effect....42
    H - Sharpness: C H>sharpness....42
    M - Mirror and/or flip the image: C M>mirroring orientation....43
    T - Saturation values: C T>saturation....43
    F - Set image data format: C F>image format....44
    W - Write current configuration to non-volatile storage: C W....44
    D - Restore factory defaults: C D....44
    Z - Zoom levels: C Z>zoom....45
  J – Read Status and Configuration....46
    J – Read ALCAM Status: J....46
    S - Get frame/image size: J S....47
    B - Brightness level of exposure: J B....48
    O - Contrast level: J O....49
    A - Automatic white-balance or explicit value: J A....49
    E - Special effect processing: J E....50
    H - Sharpness: J H....50
    M - Mirror and/or Orientation setting: J M....51
    T - Saturation value: J T....51
    Z - Zoom level: J Z....51
  K – Get Free Space....52
  B – Change Baud Rate....53
  No operation (NOP)....53

7. Standalone Mode....54
8. Booting ALCAM....55
Interface Mode Selection....55
Boot/Reset Protocol....56
UART Interface Mode....56
UART Configuration....57
SPI Interface Mode....57
SPI Bus Configurations....58
SPI Frames....58
I2C Interface Mode....60
I2C Bus Configuration....60
9. The Bootloader....62
General Description....62
Connecting and Controlling the Bootloader....62
Bootloader Commands....62
10. Hardware – Design considerations....64
Power Source....64
Main System Power....64
Camera Power....64
Camera Master Clock....64
Camera Connection....65
Status LEDs....66
Crystals....66
Main System Clock....66
Real Time Clock....66
SD Card Detect....67
Bootloader Access....67
Other Components....67
Electrical characteristics....68
USB....68
11. ALCAM OEM....69
ALCAM OEM Board....69
OEM Pinout....69
SD Connector Pinout....70
Error LED....71
Status LED....71
Push Button and 2.5mm Jack....71
12. Performance....72
Selecting the right storage media....72
File Access Speed....72
13. Result Codes....73
14. Revision History....75
15. Legal Notice....76
Disclaimer....76

1. Introduction

ALCAM is a System on Chip (SoC) which directly supports still shots, video, time-lapse video, and stop-motion video.

This drop-in SoC adds imaging and video to any project. ALCAM can be attached to a PC for software prototyping and experimenting.

The underlying processing of image data from cameras is encapsulated in an API that is both simple and powerful. The API uses standard photography/video terminology that is transmitted as compact, human readable, commands. For instance, to start taking a movie, the ASCII string “M B” (**M**ovie **B**egin) is sent to the module; subsequently, “M E” (**M**ovie **E**nd) is sent to to stop the recording and finalize processing of the captured video. Movies, still pictures, and data files can be read by the micro-controller (MCU) connected to ALCAM. Standard interface pins for SD card reader/writers and/or USB Client mass-storage devices (memory sticks, flash storage, etc.) are fully exposed.

GHI Electronics supplies the “ALCAM Explorer” software that runs on a PC. ALCAM Explorer provides both a GUI and command window for experimenting with ALCAM over a standard USB to UART cable (virtual COM port).

Hardware integration between ALCAM and a controlling processor can be UART, SPI, or I2C.

2. Example Applications

- Remote visual monitoring, real-time and/or time-lapse.
- Security systems.
- Automated machinery.
- Digital picture viewers, incorporating built-in loading of videos and stills.
- Point of sale terminal interfaces
- Measurement tools and testers.
- Autonomous multi-angle visual systems.
- Consumer products.
- Simplifies non-IR or distance based motion detectors.

3. Key Features

- Standard, Time-Lapse, and Stop Motion video modes.
- Still images.
- Resolutions: 720 HD, 1024x768, 800x600, 640x480, 320x240, 160x120.
- Serial interface choices for command and data: I2C, UART, and SPI.
- Configuration settings available for: brightness, contrast, white balance, special effects, sharpness, saturation, mirror, rotate, and the image file format.
- Simple human-readable command protocol.
- Supports OV2640 image sensors.
- 5V tolerant GPIOs.

4. Architecture

ALCAM's firmware and pins directly support control of USB and SD cards.

The MCU's connection to ALCAM can be UART, SPI, or I2C; whatever is the best fit for the project.

The circuit design is completed with the designer's choice of lens, supported sensor, image processor and storage media.

Commands

All communications between the MCU and ALCAM use a well defined serial protocol; which is compact and easy to implement.

Other than the raw data read or written to files, Commands and their results are in human-readable ASCII. This allows for easier development and troubleshooting.

The simplicity of commands can be seen with this example which begins recording a movie to a default file. Send ALCAM the command:

```
M B
```

ALCAM returns the string:

```
!00
```

the "**!00**" (the "final result code") indicates the command was processed with no errors. During development, this entire command sequence could be executed from a terminal program running on a PC.

FAT File-System Engine

The file-system engine for the storage media uses FAT file-system standards. It has been optimized for high performance and reliability.

Memory Card Access (SDHC, SD or MMC)

ALCAM includes a memory card driver that supports SD, SDHC and MMC cards. This gives ALCAM the ability to access a wide range of memory cards such as standard or high capacity

SD/MSD cards or multimedia cards. There is no limit on the card capacity.

Bootloader

The bootloader is a small embedded application that

- Initializes ALCAM.
- Verifies and launches the ALCAM firmware.
- Provides an interface for firmware updates.

The bootloader is accessed through the UART port. It uses XMODEM 1K to transfer the firmware file to ALCAM. The Bootloader chapter explains this in detail.

5. Pinout

The ALCAM SoC package is a standard 16x16mm LQFP100. The following table is a list of supported pins on ALCAM.

Pin	Name	Description / (Notes)
1	BUSY / UART_RTS	Indicates that the ALCAM is busy
2	DNC	Do Not Connect
3	DCMI D4	Camera *Data* Pin
4	DCMI D6	Camera *Data* Pin
5	DCMI D7	Camera *Data* Pin
6	VBAT	1.65V ~ 3.6V for RTC backup / **(typically 3.0V coin cell)**
7	DNC	Do Not Connect
8	OSC32 IN	Pin 1 of a 32.768 kHz crystal / **(Connect an 18pF capacitor to ground) (Optional)**
9	OSC32 OUT	Pin 2 of a 32.768 kHz crystal / **(Connect an 18pF capacitor to ground) (Optional)**
10	VSS1	Ground
11	VDD1	3.3V Power / **(Connect a .1µF capacitor to ground)**
12	OSC IN	Pin 1 of a 12 MHz crystal / **(Connect an 18pF capacitor to ground)**
13	OSC OUT	Pin 2 of a 12 MHz crystal / **(Connect an 18pF capacitor to ground)**
14	RESET#	ALCAM Reset / **(Pull-up to VCC (3.3V) w/10k resistor)**
15	DNC	Do Not Connect
16	DNC	Do Not Connect
17	DNC	Do Not Connect
18	DNC	Do Not Connect
19	VDD2	3.3V Power / **(Connect a .1µF capacitor to ground)**
20	VSSA	Ground
21	VREF+	3.3V Power

Pin	Name	Description / (Notes)
22	VDDA	3.3V Power / **(Connect a .1µF capacitor to ground)**
23	Function 12	TBD
24	Function 1	TBD
25	Function 2	TBD
26	LED2	ALCAM Status Indicator LED
27	VSS2	Ground
28	VDD3	3.3V Power / **(Connect a .1µF capacitor to ground)**
29	DCMI HSYNC	Camera *Horizontal Sync* Pin
30	SPI CLK	SPI *Clock* Interface to External MCU
31	DCMI PIXCK	Camera *Pixel Clock* Pin
32	SPI MOSI	SPI *Master Out Slave In* Interface to External MCU / **(Boot mode Select Pin)**
33	DNC	Do Not Connect
34	DNC	Do Not Connect
35	DNC	Do Not Connect
36	DNC	Do Not Connect
37	PB2	**(Connect to ground w/10k resistor)**
38	DNC	Do Not Connect
39	DNC	Do Not Connect
40	PWR DOWN	Camera *Power-down* Pin
41	DNC	Do Not Connect
42	LED1	ALCAM Error Code Indicator LED
43	DNC	Do Not Connect
44	Function 5	Standalone mode trigger
45	Function 6	TBD
46	Function 7	TBD
47	UART TX / I^{2}C SCL	UART *TX Pin* and I^{2}C *Clock Pin* Interface to External MCU
48	UART RX / I^{2}C SDA	UART *RX Pin* and I^{2}C *Data Pin* Interface to External MCU / **(Pull-up to VCC (3.3V) w/10k resistor)**

Pin	Name	Description / (Notes)
49	VCAP1	**(Connect a 2.2µF capacitor to ground)**
50	VDD4	3.3V Power / **(Connect a .1µF capacitor to ground)**
51	DNC	Do Not Connect
52	DNC	Do Not Connect
53	Function 8	TBD
54	Function 9	TBD
55	DNC	Do Not Connect
56	DNC	Do Not Connect
57	DNC	Do Not Connect
58	DNC	Do Not Connect
59	DNC	Do Not Connect
60	DNC	Do Not Connect
61	DNC	Do Not Connect
62	DNC	Do Not Connect
63	DNC	Do Not Connect
64	DNC	Do Not Connect
65	SD D0	SD Card *Data* Pin
66	SD D1	SD Card *Data* Pin
67	MCO1	24 MHz Signal Output / **(Connect to Camera Master Clock pin if not using external 24 MHz crystal for camera sensor)**
68	DCMI D0	Camera *Data* Pin
69	DCMI D1	Camera *Data* Pin / (**Pull-up to VCC (3.3V) w/10k resistor)**
70	Function 10	TBD / USB D- / **(Connect a 22Ω resistor to MCU D- pin)**
71	Function 11	TBD / USB D+ / **(Connect a 22Ω resistor to MCU D+ pin)**
72	Function 3	TBD
73	VCAP2	**(Connect a 2.2µF capacitor to ground)**
74	VSS3	Ground
75	VDD5	3.3V Power / **(Connect a .1µF capacitor to ground)**

Pin	Name	Description / (Notes)
76	Function 4	TBD
77	SPI SSEL	SPI *Slave Select* Interface to External MCU / **(Boot mode Select Pin)**
78	SD D2	SD Card *Data* Pin
79	SD D3	SD Card *Data* Pin / (**Pull-up to VCC (3.3V) w/10k resistor)**
80	SD CLK	SD Card *Clock* Pin
81	DNC	Do Not Connect
82	DNC	Do Not Connect
83	SD CMD	SD Card *Command* Pin
84	DNC	Do Not Connect
85	DNC	Do Not Connect
86	Camera RESET	Camera *Reset* Pin
87	SD CD	SD Card *Card Detect* Pin
88	DNC	Do Not Connect
89	DNC	Do Not Connect
90	SPI MISO/UART CTS	SPI *Master In Slave Out* Interface to External MCU and UART *CTS* signal to MCU (Connect to ground if not using UART handshaking).
91	PB5	**(Connect to VCC (3.3V) w/10k resistor)**
92	DCMI D5	Camera *Data* Pin
93	DCMI VSYNC	Camera *Vertical Sync* Pin
94	BOOT0	**(Connect to ground w/10k resistor)**
95	Camera I^{2}C SCL	Camera I^{2}C *Clock* Pin Interface / **(Pull-up to VCC (3.3V) w/2.2k resistor)**
96	Camera I^{2}C SDA	Camera I^{2}C *Data* Pin Interface / **(Pull-up to VCC (3.3V) w/2.2k resistor)**
97	DCMI D2	Camera *Data* Pin
98	DCMI D3	Camera *Data* Pin
99	VSS4	Ground
100	VDD6	3.3V Power / **(Connect a .1µF capacitor to ground)**

6. ALCAM Command Set

These specifications are subject to change; including, but not limited to, re-definition, additions, and/or deletions of commands, arguments, and results.

Terminology and Syntax of Command Definitions:

- **Examples and declarations are color coded.** Items going from the controlling MCU to the ALCAM are red. Items transmitted from ALCAM to the MCU are in green.
- **All commands sent from the MCU and all responses sent back by ALCAM are 8 bit, readable ASCII characters**. The only exception to this is when raw-data is sent to or from ALCAM as part of streaming images and/or file contents.
- **All commands and all responses are terminated by a newline ("\n", 0xA).** In this manual, if a command or response is transmitted with multiple, newline separated, strings, the examples will show them sequentially on separate lines. For example, when queried for the date and time, ALCAM would send back:

  "15:05:45\n2014-12-30\n!00\n"

  The examples in this manual will show this as:

  15:05:45

  2014-12-30

  !00

  ALCAM will accept carriage-return "\r" in place of "\n".
- **Every result returned by ALCAM includes a result code** of the form "!RR". The value "!00" signifies no error. The result code can be looked up in the Result Codes chapter. Most commands have two result codes, an initial result code that is immediately sent back to MCU after ALCAM has received a new command and a final result code that comes at the very end of all data transfers to and/or from the MCU and ALCAM.

  IMPORTANT: **If the *initial result code* represents an error, the transaction should be terminated.** Here is an example of a *successful* File Read operation; following that is the same example but *with an error*:

  F R>S:\test.txt

!00	initial result code. There were no errors; data bytes will follow...
$ssssssss!00	Exactly ssssssss bytes are sent followed by the final result code

In this case, after ALCAM transmits the final result code, it goes into an idle state waiting to receive the next command.

In the case of an initial result code that signals an error: ALCAM will go into the idle state immediately after transmitting the error:

F R>S:\test.txt

!03	Initial result code indicates an error. ALCAM does not send any bytes of data or the final result-code.

ALCAM goes into the idle state immediately after transmitting the error.

- **All numbers sent between the controller and ALCAM are written as strings of hexadecimal digits.** The digits are encoded using ASCII. For example, to send the decimal number 16 to ALCAM, the string transmitted would be “10” which is the byte value 0x31 (ASCII for 1) followed by 0x30 (ASCII for 0). Hexadecimal digits A to F must be upper case. Numbers sent to ALCAM can be variable length, for example the string shown above “10” can be sent as “0010” or “00000010”; on the other hand, ALCAM normally returns all numbers with a 0 (zero), left-padded, 8 character wide field. Most numbers returned by ALCAM will be proceeded by a dollar sign ($).
- **ALCAM is space sensitive.** For example, the Movie Begin command is:

  “`M B`”

  if the MCU sends:

  “`M  B`”

  an error will be returned.
- **ALCAM is case in-sensitive.** For example, if a command is defined as

  “M B”

  then

  “m B”

  will also be accepted.
- **FPS** – Frames per second refers to suggested playback speed; it does not change the actual frames in the movie file. It is added as part of the file's meta-data. Examples:

If FPS was set to 30 and ALCAM took a movie in *normal* video mode consisting of 30 frames then, when viewed, the movie should play in one second.

If FPS was set to 30 in *Stop Motion* mode and 90 frames were taken over the course of an hour, the playback should be 3 seconds.

- **Filenames** take the form: *device-specifier:\path*\name. Where the device-specifier is either "U" for USB or "S" for SD card. Path is a back-slash separated list of directory (folder) names. Examples:
    1. "S:\my_picture.jpg" This filename refers to the SD card. It will be in the *root* directory (\).
    2. "U:\my_picture.jpg" The filename refers to the USB drive, in the *root* (\) directory.
    3. "S:\rootfolder\subfolder\my_picture.jpg" The filename refers to the SD card, in the directory "\rootfolder\subfolder\"

- **Default Filenames** have the format in commands that have an optional filename argument

  YYYYMMDD_HHMMSS_xxx

  Where YYYY is year, MM month (01-12), DD day (01-31), HH hours (00-23), MM minutes (00-59), SS seconds (00:59), xxx is an internally generated number that is sequential (000-999).

  The name is suffixed appropriately for its contents (movies: ".avi", single shot: ".jpg" or ".bmp")

  If an SD card is connected, the file will be there; otherwise it will be on the USB drive.

  For Example: "20141230_130231_001.jpg". If there is both a USB drive and an SD card available, the filename refers to the SD card (S:\ 20141230_130231_001.jpg).

  The Get Last Filename command (F G) will return the last default name used.

IMPORTANT: Read the command references carefully as there are exceptions to the above rules!

Command Classes

Commands are grouped by class. Multiple commands in any class use the same "Command Set Letter" for the first character of the command and are usually followed by a sub-command letter and/or optional arguments.

Command Set Letter	Class or Function
M	Movie Operation Controls.
P	Picture Operation Controls.
C	Camera Configurations.
D	Power Control
T	Time and Date Operations.
F	File Operations.
I	Initializes Device Drivers.
J	Read Status Register and Configurations.
K	Get Available Storage on Media.
Q	Format media.
B	Change Baud Rate of UART Interface.
NOP*	Newline is a No Operation Command.
V	Get Version Number of Firmware.

*NOP is not the letter for the No Operation COMMAND, the command is sent as either a newline or a carriage-return. This is convenient for the case where a terminal program on a PC is being used over UART; in which case, a blank line (ENTER key) is the NOP and causes no-error.

V – Get Firmware Version

COMMAND SYNTAX	Command Synopsis
V	Return Current Version of ALCAM's Firmware.

Arguments:

None.

Results:

vn.n.n	The Firmware Version
!RR	Final result code

Notes:

The bootloader has its own version code. See the Bootloader Commands section.

The use of decimal here is an exception to ALCAM's rule of returning numbers in hexadecimal with a fixed field length.

Example:

```
V
v0.0.1
!00
```

I – Initialization Commands (Camera, USB, SD)

COMMAND SYNTAX	Command Synopsis
I C	Initialize the camera.
I S[>clock]	Initialize the SD card.
I U	Initialize the USB.

The initialize commands reset and establish a default state for devices. Under normal circumstances, initialization will occur automatically.

During start-up the camera is initialized; if present, SD Card and/or USB initialization occurs.

During active running, if an SD card or USB device is inserted, appropriate initialization takes place.

The time taken to initialize either the SD Card or USB device is variable based on the physical device. This time can be significant; on the order of multiple seconds.

Initialization of the camera takes approximately 500 milliseconds.

In general, there is no need to use initialization commands; they can be used to perform a "software reset."

If ALCAM cannot initialize a storage device after the media is inserted (for example, due to timeouts) the MCU can issue the initialize command explicitly and monitor the result with the Read Status command ("J"). If errors persist in initializing, the media may be bad.

All commands have the same result syntax:

Result:

!RR	Final result code

The final result code acknowledges that the command has been accepted and the action has been started; it does not indicate that initialization has completed. Initialization is a blocking operation, no other commands will be processed until completion. To check for completion, issue a command (NOP, J, V, etc.), then wait for the response.

The initialize SD card command take an optional argument, clock.

Clock allows the card driver to be adjusted to card readers that do not function well at some speeds.

Values for clock are:
0 - 24 MHz.
1 - 16 MHz.
2 - 12 MHz. **(the default value**)
3 - 9.6 MHz.
4 - 8 MHz.

ALCAM requires fast data transfer to SD cards. At a minimum, cards should be class 4; we recommend using cards with class 6 or 10, especially for large resolutions.

Example:

```
I S          initialize SD
!00
M B          begin a movie, by default it will be stored on the SD card.
!00          command succeeded, recording is underway
M E          end a movie
!00          success, post processing of movie begins
$00F00000 the movie size is 15,728,640 bytes
!00          movie file was successfully written.
I U          Initialize the USB storage system
!00
M B>U:\test.avi    begin, put the video file (test.avi) on the USB drive
!00          command succeeded, recording is underway
M E          end a movie
!00          success, post processing of movie begins
$00F00000 the movie size is 15,728,640 bytes
!00
```

D – Power Control

COMMAND SYNTAX	Command Synopsis
D 0	Power camera up
D 1	Power down the camera
D 2	Reboots ALCAM and stays in bootloader mode

By default, the camera is powered-up and initialized.

The power down command (D 1) can be used to turn the camera off.

Subsequently, D 0 is used to enable power.

When D 0 is used the camera is automatically initialized (the equivalent of the I C command).

Results:

!00	These commands always return "!00" (no error)

Note:

J – Read Status and Configuration can be used to determine the current power state of the camera.

When using the D 2, you do not need to set the boot mode pins. The device goes directly to the bootloader.

P – Picture (still shot) Control

COMMAND SYNTAX	Command Synopsis
P C[>**filename**]	Capture to file.
P R	Take Picture and send to MCU.
P T[>**filename**]	Capture to file. Produce a thumbnail picture.

Command Descriptions

C – capture image to storage *media: P C[>filename]*

Arguments:

Filename – optional. If filename is supplied it should have an appropriate extension.

Default: "YYYYMMDD_HHMMSS_*xxx.ext*"
Where YYYY is year, MM month (01-12), DD day (01-31), HH hours (00-23), MM minutes (00-59), SS seconds (00-59), *xxx* is an internally generated number (000-999). *ext* is either "jpg" or "bmp". Precedence of media selection: SD card then USB.

Results:

!RR	Initial result code If it is !00, the picture is being taken.
$HHHHHHHH	The size of the image written to the storage device
!RR	Final result code

R – capture Picture: P R

A still image is captured and then sent over the interface between ALCAM and the MCU.

Arguments:

None.

Results:

!RR	Initial result code.
$HHHHHHHH	Result size of the picture in bytes. Specified in hexadecimal.
data_stream!RR	The picture, exactly $HHHHHHHH bytes will be sent; followed by the Final result code.

Notes:

If the Final result code is an error, ALCAM will still transmit exactly $HHHHHHHH bytes; the data is undefined.

Example:

```
P C>U:\example.jpg        take a picture and put it in “example.jpg” on the USB drive
!00
$0001FC46                 picture file contains 103,118 bytes
!00                       the file was successfully written
P R                       take another picture and send it to the MCU.
!00
$0002FC66                 ALCAM will send 195,686 bytes to the MCU
data_stream!00            ALCAM sends 195,686 bytes to the MCU and signals success.
```

T – capture image and thumbnail: P T[>`filename`]

Arguments:

filename – optional. This is the name of the file containing the full image. It should have an appropriate extension.

The thumbnail name is "filename_Thumbnail.ext" with ext appropriate for the file format type. Any original extension on filename is removed before _Thumbnail.ext is appended.

Default:
"YYYYMMDD_HHMMSS_*xxx.ext*"

Where YYYY is year, MM month (01-12), DD day (01-31), HH hours (00-23), MM minutes (00-59), SS seconds (00:59), *xxx* is an internally generated number (000-999).

ext is either "jpg" or "bmp".

The thumbnail name will be the default name (as above), with ".ext" replaced by "_Thumbnail.ext."

The storage device is chosen by checking for the SD Card, if present, it is used; otherwise, the USB drive is used.

Results:

!RR	Initial result code
$HHHHHHHH	Size of image written to storage device.
!RR	Final result code

Notes:

This command can take longer than Picture Capture (P C) as the thumbnail file is produced by taken a small picture before the actual picture.

F – File Operations

COMMAND SYNTAX	Command Synopsis
F W>filename>length data_stream	Write stream of bytes to a file.
F C	Close the current open file, if any.
F D>**filename**	Delete the file.
F S>**filename**	Get file size.
F R>filename	Read a file.
F G	Get name of last file created by default.

The device containing the file must have been initialized, see the I – Initialization Commands (Camera, USB, SD).

Usage suggestions:

In addition to reading existing movies and stills, this command allows the MCU to store and/or read auxiliary information. Example uses would be time-stamps, information from other sensors attached to the MCU, data streams from other MCU communication links (Ethernet, UART, CAN, WiFi, etc).

Sub-command Descriptions:

W – write bytes to a file:

```
F W>filename>length
data_stream
```

Arguments:

filename - the file to contain the information

length – the size of the data that will be sent to ALCAM for the file (specified in hexadecimal)

data_stream - the data. NOTE: no newline is sent after data_stream.

Results:

!RR	Initial result code.
$HHHHHHHH	Actual amount of data written to the file.
!RR	Final result code.

If the initial result code is not “!00”, ALCAM will not send $HHHHHHHH or the final result code. The MCU must not send the data stream. ALCAM will idle, waiting for a command from the MCU.

Notes:

If the filename doesn't exist, it is created. Data is always appended.

If the file is not open (from a previous use of this command), it is opened.

For performance reasons, ALCAM buffers data. Before removing media, shutting down ALCAM, or resetting ALCAM, the MCU should wait until the final result code is received, then close the file (F C).

R – open and read bytes from a file: F R>filename

Arguments:

filename – the file that contains the desired information. The file's data is sent to the MCU. The data (data_stream) is a raw stream of bytes.

Results:

!RR	Initial result code.
$HHHHHHHH	Size of data that ALCAM will send to MCU.
data_stream!RR	HHHHHHHH bytes of data. Followed immediately by the final result code.

Notes:

If the Initial result code is not !00 , the command will be terminated; neither data_stream nor the final result code will be sent.

If the final-result-code is not !00, the data in the stream is undefined.

C – close a file: F C

Arguments:

None.

Results:

!RR	Final result code

S – get the size of a file: F S>`filename`

Arguments:

`filename` – the file to get the size of.

Results:

!RR	Initial result code.
$HHHHHHHH	The size of the file in bytes.
!RR	Final result code.

See also :

Get last default file name (F G).

D – delete a file: F D>`filename`

Arguments:

`filename` – the file to delete.

Results:

!RR	Final result code.

See also :

Get last default file name (F G).

G – Get last default filename: F G

Arguments:

None.

Results:

!RR	Initial result code if not !00, rest of data from ALCAM will not be sent
$HHHHHHHH	The length (bytes) of the filename in hexadecimal
Filename!RR	The file's name, exactly name_length characters. Immediately followed by a Final result code. If Final result code is not !00, Filename name is undefined.

If the initial result code is !58, it signifies that no default filename has yet been used.

It Useful when the MCU needs to use file commands that require a filename argument.

M – Movie Operations

COMMAND SYNTAX	Command Synopsis
M N	Configure movie to *Normal mode* (default).
M S[>fps]	Configure movie to *Stop Motion mode.*
M T[>ddd[>fps]]	Configure movie to *Time Lapse mode*.
M B[>filename]	Begin the movie.
M C	Take a frame (while in stop motion mode).
M E	End the movie.
M I>filename	Get information on movie file.

Sub-command Descriptions:

N – Set movie mode to Normal: M N

Normal mode is taking a movie with the fastest rate possible given the current configuration settings. For example, the size of a frame, number of colors in the scene being recorded, etc.

Arguments:

None.

Results:

!RR	Final result code.

S – Set movie mode to Stop Motion: M S[>fps]

Arguments:

fps – Frames per second, an optional hexadecimal number, that refers to suggested playback speed. It is embedded in the meta-data of the video file. It does not change the actual frames in the movie file.

Default: 16 FPS.

Results:

!RR	Final result code.

T – Set movie mode to Time Lapse: M T[>delay[>fps]]

Arguments:

delay – an integer, in hexadecimal, specifying the time between frame captures.
DEFAULT is 500ms

fps – Frames per second, an optional hexadecimal number, that refers to suggested playback speed. It is embedded in the meta-data of the video file. It does not change the actual frames in the movie file.

Default: 16 FPS

Results:

!RR	Final result code.

B – Begin recording the movie: M B[>`filename`]

Arguments:

`filename` – optional, the file to contain the information. If `filename` is supplied it should have an appropriate extension.

Default: "YYYYMMDD_HHMMSS_*xxx*.avi"
Where YYYY is year, MM month (01-12), DD day (01-31), HH hours (00-23), MM minutes (00-59), SS seconds (00:59), *xxx* is an internally generated number (000-999). Precedence of media usage: SD card then USB.

Results:

!RR	Final result code.

A storage device must be initialized, see the I – Initialization Commands (Camera, USB, SD) section.

C – Capture a frame: M C

This command is used after a stop-motion movie has begun. ALCAM takes a frame and adds it to the movie.

Arguments:

None.

Results:

!RR	Final result code.

E – End the movie: M E

Arguments:

None.

Results:

!00	Initial result code. Capturing has stopped; ALCAM begins post processing.
$HHHHHHHH	Size of the movie file in hexadecimal.
!RR	Final result code. If it is !00 then the post processing is finished file operations are complete.

Notes:

For performance reasons, ALCAM buffers data. Before removing media, shutting down ALCAM, or resetting ALCAM, the MCU should wait until the Final result code is received.

I – Get information on a movie file: M I>`filename`

Arguments:

`filename` – the file to get information on.

Results:

!00	Initial result code.
$HHHHHHHH	The requested number of frames per second (FPS) for playback.
$HHHHHHHH	This number is the total number of frames in the movie.
$HHHHHHHH	The size/dimension code as defined in the configure size ("C S>size_code") command (gives W x H).
!RR	Final result code.

T – Time and Date Operations

COMMAND SYNTAX	Command Synopsis
T B	Initialize and use battery backed RTC for time and date.
T I	Initialize and use internal system clock for time and date.
T S>timestamp	Set Date and Time.
T G	Get the date and time.

Sub-command Descriptions:

B – Use battery backed real-time clock: T B

Unless an external battery is wired to the VBAT pin, the Set Date and Time, Get Date, and Get Time commands will work identically as though the time was using system mode.

After issuing this command, the time and date will be preserved through resets.

Arguments:

None.

Results:

!RR	Final result code.

I – Use internal clock: T I

In this mode, the time and date commands use the internal system clock. Every time ALCAM is reset the clock starts at 00:00:00 1980-01-01.

Arguments:

None.

Results:

!RR	Final result code.

S – Set date and time: T S>timestamp

Arguments:

timestamp is a hexadecimal number representing 32 bits of encoded date and time, as follows:

Bits(s)	Field	Description
31-25	Year	Years since 1980.
24-21	Month	1-12.
20-16	Day	1-31.
15-11	Hour	0-23.
10-5	Minute	0-59.
4-0	Second	Seconds divided by 2 (0-30).

Results:

!RR	Final result code.

For example: 34212002 is 04:00:00 2006-01-01

D - Get date and time: T G

Arguments:

None.

Results:

HH:MM:SS	Hours (00-24) - minutes – seconds.
YYYY-MM-DD	Year, Month(00-12), Day(00-31).
!RR	Final result code.

Note:

The number representation for time and date as used above is in decimal not hexadecimal. The fields are fixed width and padded with zero (0).

Unlike most other commands there is no initial result code.

Q – Format Media

COMMAND SYNTAX	Command Synopsis
Q S	Format SD card.
Q U	Format USB storage.

The speed of formatting can vary greatly, size is one factor, another is the type and/or class of the media.

Warning: Formatting erases all data on the selected media. Use it with care.

The results are the same for both commands:

!00	Initial result code If it is !00, formatting has started.
$HHHHHHHHHHHHHHHH	Formatting is complete, this is the free size in bytes (64 bit long number in hexadecimal).
!RR	Final result code.

Note: the initial result code is an acknowledgment that ALCAM is processing the command. Due to extra processing this command can take longer than average. The initial result code allows the MCU to adjust any timeout values and similar activities.

C – Configure Camera

COMMAND SYNTAX	Command Synopsis
C S>size	Set frame/image size.
C B>brightness	Set brightness level.
C O>contrast	Set contrast level.
C A>white balance	Manual or automatic setting of white balance.
C E>special effect	Select from a variety of special effect processing.
C H>sharpness	Select from different levels of sharpness.
C M>mirroring orientation	Mirror and/or flip the image.
C T>saturation	Select different saturation values.
C F>image format	For still pictures choose image data format.
C W	Write current configurations to non-volatile storage (EEPROM).
C D	Restore Configuration to factory defaults.
C Z>zoom	Select zoom levels.

***Important notes:**

When configure commands are used, they are not stored; on the next reset (or factory reset command (C D)), they will be lost.

To store configuration for future resets, make the desired changes then use the Write Configuration (C W) command.

For each Set Configuration command (C) there is a corresponding Get Configuration command (J). For instance for “C A>white_balance” there is “J A.” There is one exception: to get the current file type (JPEG or BMP) of still images, the “J” Get Status Register is used (there is no “J F”).

The C D (restore configuration) does not change time commands.

Sub-command Descriptions:

S - Set frame/image size: C S>size

Arguments:

size as shown below.

Default: value is 1, 320x240, QVGA

Size value	Pixel dimensions	Common name
0	160x120	QQVGA
1	320x240	QVGA
2	640x480	VGA
3	800x600	SVGA
4	1024x768	XGA
5	1280x720	720 HD

Results:

!RR	Final result code

B - Set brightness level: C B>brightness

Arguments:

brightness. a value from 0 to A (10 decimal). Values above 0 increase the brightness.

Default: 0. Values above 0 increase the brightness.

Results:

!RR	Final result code.

O - Set contrast level: C O>contrast

Arguments:

contrast is a value from 0 to A (10 decimal). Each value above 0 increases the contrast.

Default: 0.

Results:

!RR	Final result code.

A - Manual or automatic white-balance: C A>white balance

Arguments:

white balance – if 0, then automatic white-balance is used; otherwise, a 6 byte hexadecimal number represents the white balance using Red, Green, and Blue: RRGGBB. The maximum value is FFFFFF

Default: 0

Examples:

```
C A>0
!00
C A>400511
!00
```

The first command set the white-balance to automatic.
The 2nd command sets it to red 64, green 5, and blue 17 (in decimal)

Results:

!RR	Final result code.

E - Special effect processing: C E>special effect

Arguments:

special effect – a number from the Value column

Default: 0

Special Effect Value	Applied Effect
0	None
1	Sepia
2	Blue tint
3	Green tint
4	Red tint
5	Yellow tint
6	Black and White
7	Negative

Results:

!RR	Final result code

H - Sharpness: C H>sharpness

Arguments:

sharpness - a value from 0 to 8. Numbers above 0 correspond to increasing levels of sharpness.

Default: 0.

Results:

!RR	Final result code.

M - Mirror and/or flip the image: C M>mirroring orientation

Arguments:

mirroring orientation – a number from the value column

Default: 0

Mirroring Orientation Value	Orientation
0	Normal (the Default)
1	Flip
2	Mirror
3	Mirror and flip

Results:

!RR	Final result code.

T - Saturation values: C T>saturation

Arguments:

saturation - a value from 0 to 8. Numbers above 0 correspond to increasing levels of saturation.

Default: 0.

Results:

!RR	Final result code.

F - Set image data format: C F>image format

Arguments:

image format – one of “J” or “B”. J for JPEG and B for BMAP.

Default: J

Results:

!RR	Final result code.

Note:

Most configuration commands have a corresponding Get Configuration command; in this case, the command would be “J F”. However, the current image format is part of the Get Status Register command (“J” with no sub-command letter).

W - Write current configuration to non-volatile storage: C W

Arguments:

None.

Results:

!RR	Final result code.

Note:

Configuration changes are temporary; to establish a configuration that will be used after reset, set the desired values, then use this command.

D - Restore factory defaults: C D

Arguments:

None.

Results:

!RR	Final result code.

Note:

This removes any changes that were stored with the Write Configuration command (C W).

Z - Zoom levels: C Z>zoom

Arguments:

zoom level	Enlargement
0	None (default)
1	1.25 (125%)
2	1.50 (150%)
3	1.75 (175%)

Support for zoom levels vary by dimension/size:

Zoom Supported	Pixel Dimensions	Common Name	Size Value (see C S>size)
Yes	160x120	QQVGA	0
Yes	320x240	QVGA	1
Yes	640x480	VGA	2
Yes	800x600	SVGA	3
No. 0 only	1024x768	XGA	4
No. 0 only	1280x720	720 HD	5

Results:

!RR	Final result code.

Note:

Zoom is performed algorithmically; consequently, larger levels of zoom, will have greater image degradation.

J – Read Status and Configuration

COMMAND SYNTAX	Command Synopsis
J	Query ALCAM for current settings and storage device status
J S	Get frame/image size.
J B	Get brightness level.
J O	Get contrast level.
J A	Get white balance setting.
J E	Get special effect setting.
J H	Get level of sharpness.
J M	Get mirror/flipping setting.
J T	Get saturation value.
J Z	Get zoom levels.

The “J” commands, return the *current* values in effect. For example, if the level of brightness held in stored configuration data was $00000001, and a reset or “C D” command occurred, then “J B” would return $00000001. If a subsequent “C B>0” command was issued then “J B” would return $00000000.

Sub-command Descriptions:

J – Read ALCAM Status: J

The J command returns bit-fields describing the current state of ALCAM.

Results: the high level format of the return is:

$HHHHHHHH	Hexadecimal representation of the bit-fields defined below.
!RR	Final result code.

Bit-Field definitions:

7	6	5	4	3	2	1	0
Reserved	**Current Image Format** 0:JPEG 1: BMAP	Reserved	Reserved	Reserved	**SD Card Detect** 0: not detected 1: detected	**SD Card Initialized** 0: no 1: yes	**Camera Initialized** 0: no 1: yes

15	14	13	12	11	10	9	8
Camera Power 0: Powered 1: Power Down Mode	**USB Initialized** 0: no 1: yes	**USB Detected** 0: no 1: yes	Reserved	Reserved	Reserved	**Time-Lapse** 0: no 1: yes	**Stop-Motion** 0: no 1: yes

23	22	21	20	19	18	17	16
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

31	30	29	28	27	26	25	24
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved

S - Get frame/image size: J S

Results:

image_size	See below.
!RR	Final result code.

Size value	Pixel dimensions	Common name
$00000000	160x120	QQVGA

$00000001	320x240	QVGA
$00000002	640x480	VGA
$00000003	800x600	SVGA
$00000004	1024x768	XGA
$00000005	1280x720	720 HD

B - Brightness level of exposure: J B

Results:

brightness	See below.
!RR	Final result code.

brightness a value from $00000000 to $0000000A (10 decimal). Values above $00000000 represent increased brightness.

O - Contrast level: J O

Results:

contrast	See below.
!RR	Final result code.

contrast is a value from $00000000 to $0000000A (10 decimal). Each value above $00000000 is a higher level of contrast.

A - Automatic white-balance or explicit value: J A

Results:

white_balance	See below.
!RR	Final result code.

white_balance – if $00000000, then automatic white-balance is in use; otherwise, a hexadecimal number represents the white balance using Red, Green, and Blue: $00RRGGBB where “00” is undefined. The maximum value returned is $00FFFFFF.

E - Special effect processing: J E

Results:

Special effect	See below.
!RR	Final result code.

Special Effect Value	Applied Effect
$00000000	none
$00000001	sepia
$00000002	blue-ish
$00000003	Green-ish
$00000004	Red-ish
$00000005	Yellow-ish
$00000006	Black and White
$00000007	Negative

H - Sharpness: J H

Results:

sharpness	See below.
!RR	Final result code.

sharpness - a value from $00000000 to $00000008. Numbers above $00000000 correspond to increasing levels of sharpness.

M - Mirror and/or Orientation setting: J M

Results:

mirroring orientation	See below.
!RR	Final result code.

Mirroring_orientation value	Orientation
$00000000	Normal (default)
$00000001	Flip
$00000002	Mirror
$00000003	Mirror and flip

T - Saturation value: J T

Results:

saturation	See below.
!RR	Final result code.

saturation a value from $00000000 to $00000008. Numbers above $00000000 correspond to increasing levels of saturation.

Z - Zoom level: J Z

Results:

zoom	See below.
!RR	Final result code.

zoom – see Configure Zoom (C Z).

K – Get Free Space

COMMAND SYNTAX	Command Synopsis
K S	Report available space on SD card.
K U	Report available space on USB storage.

The speed of this command can be longer than many of the other commands.

Arguments:

None.

Results: are the same format for both commands:

!RR	Initial result code.
$HHHHHHHHHHHHHHHH	64 bit value which is the number of free bytes on the media.
!RR	Final result code.

B – Change Baud Rate

COMMAND SYNTAX	Command Synopsis
B baudrate	Set the baudrate of ALCAM when using UART.

Arguments:

baudrate

Default: 1C200 (115200 decimal)

Results:

!RR	Initial result code If it is !00, ALCAM will change the UART baud rate. This Initial result code code is transmitted using current baud rate, not the new baudrate. If there is an error the baud rate will not be changed and the Final result code will not be sent.
!RR	Final result code transmitted using the new baudrate.

Note:

The default baud rate is re-established on reset.

No operation (NOP)

COMMAND SYNTAX	Command Synopsis
\n or \r	Sending a line-feed (0xA) or carriage-return (0xD) to ALCAM causes no change in the state of ALCAM.

Results:

!00	Final result code.

This command is useful when a terminal program on a PC is being used over UART to control ALCAM; in which case, the ENTER key is the NOP.

7. Standalone Mode

Standalone mode on the ALCAM allows you to trigger different actions by using a button, remote camera control, or other input source connected to the standalone pin (which is internally pulled low).

To trigger an action, bring the pin high for the duration specified in the following table:

Duration	Action
Less than 3 seconds	Take a picture
3 to 9 seconds	Record a movie
More than 9 seconds	Load the configuration

When taking a picture or recording a movie using this mode, the default name rules are used. See the default file names discussion in the command set section.

To stop recording a movie when in normal or time-lapse mode, bring the pin high for under one second.

When in stop-motion mode, after you begin recording a movie in standalone mode, further presses will add a frame to the movie. To stop recording the movie, you must bring the pin high for three seconds.

When recording a movie, you must wait for the error LED to illuminate after stopping the movie before removing the storage media. The device needs some time to flush the data.

The load configuration action is useful because standalone mode allows the device to operate without a host MCU. In order to configure the camera, you can generate a configuration file using ALCAM Explorer and save the resulting file to the root of the connected storage media. Make sure the configuration is named “ALCAM.cfg”.

When using standalone mode, you will not receive any output over the communication interface nor can standalone mode be controlled that way. For example, a movie begun in standalone mode can only be stopped using standalone mode. In fact, standalone mode is disabled when you begin a movie through the command interface.

8. Booting ALCAM

Interface Mode Selection

The micro-controller circuit can be connected to ALCAM using one of the following interfaces:

1. UART.
2. SPI.
3. I2C.

During initialization (after reset or power-up), ALCAM detects which interface to use by reading SPI_SSEL and SPI_MOSI:

SPI_SSEL	SPI_MOSI	Interface
low	low	UART
low	high	Bootloader UART
high	low	I2C
high	high	SPI

The bootloader is primarily used for firmware updates and can only be accessed using UART. The Bootloader chapter explains this in greater detail.

SPI_SSEL and SPI_MOSI have internal pull-down resistors.

Boot/Reset Protocol

1. The MCU holds ALCAM's RESET pin low until the power is stable.
2. The MCU sets SPI_SSEL and SPI_MOSI to indicate which bus is chosen (as shown in the table above).
3. After the interface selection pins are set, the MCU sets RESET high. **The RESET pin is NOT 5V tolerant.**
4. The MCU delays for a time allowing ALCAM to initialize (described below).
5. If the UART interface is selected, ALCAM samples samples SPI_SCK. If this pin is pulled low, the UART baud rate will be 9600; otherwise, it will be 115200 baud.
6. ALCAM sends the MCU a banner string; which should be read to synchronize with ALCAM.

The total time from reset until ALCAM sends its banner consists of the following:

- ALCAM configures itself: approximately 50µS.
- It initializes the camera (equivalent to the I C command): approximately 500 ms.
- If present, it initializes the USB (I B) and SD Card (I S): due to differences in external devices, this time is variable**.**

The total time could be anywhere from 2 to 10 seconds. The MCU can begin checking for ALCAM's banner after approximately 50µS.

UART Interface Mode

The UART interface uses four hardware signals:

1. UART_TX data sent from ALCAM to the controller.
2. UART_RX data sent from the controller to ALCAM.
3. UART_CTS pin for data for control. The MCU sets this pin low when it is ready to receive data.
4. UART_RTS pin (also known as the BUSY pin) indicator pin for data flow control. While high, no data should be sent from MCU to ALCAM.

UART Configuration

- The defaults are:
  - Baudrate: 115200.
  - Data bits: 8.
  - Parity: None.
  - Stop bits: 1.
- At boot time the SPI_SCK pin is internally pulled high. If this pin is pulled low during reset or power-up, the UART baud rate will be 9600 instead of 115200 baud.
- The baud rate can be changed at run-time using the B – Change Baud Rate command.

Due to interrupt handling, ALCAM may need to pause data exchange with the MCU; this is indicated with a BUSY pin. Because the interrupts are related to loading data into or out-of a 4KB buffer, the checks must occur at 4KB boundaries in the data. The 4KB number only applies to file data read/written on an SD Card or USB drive. Command bytes used by the MCU are not counted.

Checks are not necessary if the data totals no more than 4KB.

After 4KB has been transferred (read or written), the MCU samples the BUSY pin; it must pause while the pin is high.

SPI Interface Mode

With the SPI interface ALCAM is a slave device. It uses five hardware signals:

1. SPI_SSEL: slave select.
2. SPI_MOSI: mosi.
3. SPI_MISO: miso.
4. SPI_SCK: clock.
5. BUSY: This can be monitored while sending data to ALCAM. When it is high, no more data should be exchanged with ALCAM until it goes low.

SPI Bus Configurations

- The maximum SPI clock is 24 MHz.
- SPI clock idle state is low.
- Sampling is on the rising edge.
- Slave select is active low.
- Multibyte transfers are supported.

SPI Frames

When using SPI, the data exchanged between the MCU and ALCAM is broken up into *frames*.

- Frames are classified as either:
  1. *Write* - MCU sending data to ALCAM (MOSI).
  2. *Read* **-** ALCAM sending data to MCU (MISO).
- All frames start with a header:
  1. The first byte indicates whether it is write (1) or read (2).
  2. The next 2 bytes are the data count. Count is a 16 bit number, with the Least Significant Byte (LSB) sent first.

Frame sizes can be adjusted to the designer's needs using the data count.

Here is an example of SPI frames used to perform the Get Firmware Version command, “V”. The SPI framing chosen here is one frame for the command and one frame for each result. If the version of the firmware is “1.0.0”, the high level (serial based) dialog looks like this:

```
V   MCU sends command
v1.0.0 MCU reads version sent by ALCAM
!00  MCU reads Result-code sent by ALCAM
```

The SPI frames are:

(1) Write Frame:

MOSI	1	2	0	'V'	'\n'
MISO	U	1	1	1	1

(2) Read Frame, first line of result:

MOSI	2	7	0	0	0	0	0	0	0	0
MISO	U	7	0	'v'	'1'	'.'	'0'	'.'	'0'	'\n'

(3) Read Frame, second line of result:

MOSI	2	4	0	0	0	0	0
MISO	U	4	0	'!'	'0'	'0'	'\n'

For a second example, the software designer uses one frame for all the results (perhaps for efficiency); in this case, the frames look like this:

(1) Write Frame:

MOSI	1	2	0	'V'	'\n'
MISO	U	1	1	1	1

(2) Read Frame:

MOSI	2	0xB	0	0	0	0	0	0	0	0	0	0	0	0
MISO	U	0xB	0	'v'	'1'	'.'	'0'	'.'	'0'	'\n'	'!'	'0'	'0'	'\n'

Because ALCAM buffers MISO bytes based on the last data transfer, the initial MISO byte of a new frame is designated as undefined. Thereafter, in a frame
where a byte from ALCAM has no significance, ALCAM transmits a 1. In the above examples, undefined is shown as U. Defined, insignificant bytes are shown as 1.

A final note on read frames: the 2nd and 3rd bytes of the MISO side of the SPI transaction is the **actual** count of data that ALCAM will return, this may not be the same as the size that was requested in the Read Frame header. For example look at the header of a read request for 10 bytes:

MOSI	2	0xA	0
MISO	U	0x8	0

ALCAM indicates it only has 8 bytes to return. The MCU must only read 8 bytes. So the entire transaction would look like:

MOSI	2	0xA	0	0	0	0	0	0	0	0	0
MISO	U	0x8	0	'1'	'2'	'3'	'4'	'5'	'6'	'7'	'8'

Due to interrupt handling, ALCAM may need to pause data exchange with the MCU; this is indicated with a BUSY pin. Because the interrupts are related to loading data into or out-of a 4KB buffer, the checks must occur at 4KB boundaries in the data.

The 4KB number only applies to the file data read/written on an SD Card or USB drive. The 4KB number is independent of the 3 header bytes; it is only the data bytes that matter.

Checks are not necessary if the the data count is less than or equal to 4KB.

After 4KB has been transferred (read or written), the MCU samples the BUSY pin; it must pause while the pin is high.

I2C Interface Mode

I2C interface uses three hardware signals:

1. I2C_SCL: clock.
2. I2C_SDA: data.
3. BUSY: While sending data to ALCAM, if this is high then data transmission should pause until it goes low.

I2C Bus Configuration

- ALCAM I2C slave address is 0x52 (7 bit). This is a fixed address and can not be changed.
- Bit zero is the RW bit, 0=Write 1= Read.
- The maximum allowed I2C clock is 400KHz.
- The circuit must provide pull-up resistors, usually they are 2.2K, on the bus as specified in the I2C specifications.

Transmitting and receiving data to and from ALCAM is preformed through standard I2C transactions.

- Transmitting: ALCAM's I2C address (with R/W bit set to zero) is sent; followed by the stream of bytes. ALCAM processes the assembled payload bytes sequentially.
- Receiving: The MCU starts by transmitting ALCAM's I2C-address with (R/W bit set to

one) followed by reading of one or more bytes. When a read request is sent to ALCAM and no data is available, a *No Data* token (0x00) is returned. This presents a problem when raw data (such as file contents) contains 0x00; in this case, a two byte sequence is used. Actual data bytes of 0x00 are transmitted by preceding them with an *escape* byte, 0xFF. To resolve the problem of differentiating actual data values of 0xFF versus its use as the *escape* byte, it is followed by 0xFF.

Actual Data coming from File	Data transmitted by ALCAM
0x00	0xFF followed by 0x00
0xFF	0xFF followed by 0xFF
0x01 … 0xFE	Data is sent as-is

Important:

- At the I2C level, the number of bytes transmitted may not match the actual size of the file contents. For example, If a file has one byte and this byte value is zero, ALCAM will actually send 2 bytes, a 0xFF followed by 0x00. The I2C interface processing code in the MCU should remove the escape bytes so higher level application code does not have to be aware of this.

9. The Bootloader

General Description

The bootloader is the software that runs when ALCAM is powered up or reset. It has two primary functions:

1. Boots and initializes the state of the hardware; verifies and starts execution of the ALCAM firmware.
2. GHI Electronics periodically releases new firmware; the bootloader provides the interface to install firmware on ALCAM.

Connecting and Controlling the Bootloader

- The bootloader is entered by resetting ALCAM with SPI_SSEL low and SPI_MOSI high.
- Firmware transfer uses the XMODEM protocol with 1K packets and 16bit CRC over UART.

Bootloader Commands

Command	Description
R	Run: Exit loader; run firmware
X	Transfer: Erase current firmware and flash new firmware
V	Version. Print the bootloader version number. The format is “vn.n.n”

- Commands and their result codes are readable ASCII characters.
- Successful commands return “OK.” while unsuccessful commands return an error message. Both are followed by a carriage return (0x0D) **and** line feed (0x0A).
- Commands are case-insensitive and are followed by carriage return **or** line feed.
- A PC terminal console program that supports XMODEM will work for updating.
- Another way to perform firmware updates is to add XMODEM software to the controlling MCU. With this method the controller can perform in-field firmware updates.

- The transfer and run commands will ask for confirmation before executing. Send 'Y' followed by a carriage return or line feed to confirm.
- After you confirm the transfer command, C's will be printed to the terminal to signal that the device is ready to receive the firmware.

10. Hardware – Design Considerations

WARNING: this is preliminary set of instructions. DO NOT DESIGN hardware based on this chapter.

Power Source

Main System Power

Use a clean and stable power supply of 3.3V. We recommend adding 0.1µF capacitors as close as possible to each of the power (VDD) pins.

Camera Power

A typical camera sensor uses two or three different power supplies depending on which camera sensor is being used. Please review your camera sensor's datasheet for more information on which power supplies are needed. The required power supplies found on the ALCAM OEM are as followed:

- 2.8V Digital
- 2.8V Analog

Some cameras may also require a 1.8V Digital power supply.

It is also recommended to add a .1µF capacitor as close as possible to each of the different power pins of the camera interface.

Isolating the analog ground from the digital ground is also recommended to have cleaner (less noisy) images. However, to keep the ground plane a single common ground you can use a 0 Ohm resistor by connecting the analog ground with the digital ground, or create a small channel between the digital ground and analog ground with a small 10 mil trace or even through a via. For a visual reference, please see the ALCAM OEM schematic and PCB layout.

Camera Master Clock

A typical camera sensor requires a 24 MHz crystal for proper functionality. There are two options available:

1. Provide your own external crystal, or
2. Connect pin 67 (MCO1) of ALCAM to the master clock pin of the camera sensor

Please note that some camera sensors may already provide an oscillator crystal and would not require either option above. Please review your camera sensor's datasheet for more information regarding crystals.

Camera Connection

The following is an example of a typical 24 pin camera interface as found on the ALCAM OEM module:

ALCAM Pin	ALCAM Name	Camera Pin	Camera Name
–	–	1	NC
–	–	2	Analog Ground
96	Camera I^{2}C SDA	3	I^{2}C SDA
–	–	4	Analog 2.8V
95	Camera I^{2}C SCL	5	I^{2}C SCL
86	Camera Reset	6	Camera Reset
93	DCMI VSYNC	7	VSYNC
40	PWR DOWN	8	Power Down
29	DCMI HSYNC	9	HSYNC
–	–	10	Digital 1.8V
–	–	11	Digital 2.8V
5	DCMI D7	12	D9 ↔ D7
67	MCO1	13	Master Clock
4	DCMI D6	14	D8 ↔ D6
–	–	15	Digital Ground
92	DCMI D5	16	D7 ↔ D5
31	DCMI PIXCK	17	Pixel Clock
3	DCMI D4	18	D6 ↔ D4
68	DCMI D0	19	D2 ↔ D0
98	DCMI D3	20	D5 ↔ D3
69	DCMI D1	21	D3 ↔ D1
97	DCMI D2	22	D4 ↔ D2
–	–	23	D1

ALCAM Pin	ALCAM Name	Camera Pin	Camera Name
–	–	24	D0

Status LEDs

ALCAM communicates to the user visually with two indicator LEDs. They are as followed:

- Pin 26 – Status Indicator (LED2)
- Pin 42 – Error Code Indicator (LED1)

Crystals

Main System Clock

The main oscillator is to be connected to pin 12 (OSC IN) and pin 13 (OSC OUT). External crystal capacitors should be placed in parallel with the oscillator and the oscillator pins. **Please NOTE: It may be necessary to place a 0 Ohm resistor in series of pin 13 and the crystal.** Please see the ALCAM OEM schematics for reference.

ALCAM's Main Crystal Specifications

Frequency: 12 MHz

Frequency Tolerance (@ 25ºC): ± 30ppm

Load Capacitance (C_L): 18pF

External Input and Output Crystal Capacitors: 18pF

Real Time Clock

An optional external crystal is required for the use of keeping accurate time while ALCAM is powered-down or in a hibernation. There are two requirements for the RTC circuit:

1. A 32.768 kHz crystal, and
2. A power supply that outputs from 1.65V to 3.6V (commonly provided by a 3V coin cell battery)

The RTC crystal is to be connected to pin 8 (OSC32 IN) and pin 9 (OSC32 OUT). External crystal capacitors should be placed in parallel with the RTC crystal and the OSC32 pins. A positive power terminal needs to be connected to pin 6 (VBAT) and a negative power terminal to common ground.

ALCAM's RTC Crystal Specifications

Frequency: 32.768 kHz

Frequency Tolerance (@ 25ºC): ± 20ppm

Load Capacitance (C_L): 7pF

External Input and Output Crystal Capacitors: 18pF

SD Card Detect

Pin 87 (SD CD) of ALCAM should be connected to the card detect pin of the SD connector to utilize the card detect feature on ALCAM, otherwise, connect pin 87 to ground if the card detect feature is not implemented.

Bootloader Access

It is recommended to expose the following pins for bootloader access should GHI Electronics release any minor updates and/or major firmware upgrades:

- Pin 47 – UART TX / I^2C SDA
- Pin 48 – UART RX / I^2C SCL
- Pin 32 – SPI MOSI
- Pin 77 – SPI SSEL
- Pin 14 – RESET#

For more information, please review the following sections: Interface Mode Selection and The Bootloader

Other Components

There are a few extra components needed for ALCAM to function properly listed as followed:

Resistors

- Pin 14 (RESET#) -- Pull-up to VCC (3.3V) w/10k resistor

- Pin 37 (PB2) – Connect to ground w/10k resistor
- Pin 48 (Serial RX / I^{2}C SDA) – Pull-up to VCC (3.3V) w/10k resistor
- Pin 69 (DCMI D1) – Pull-up to VCC (3.3V) w/10k resistor
- Pin 79 (SD D3) – Pull-up to VCC (3.3V) w/10k resistor
- Pin 91 (PB5) – Pull-up to VCC (3.3V) w/10k resistor
- Pin 94 (BOOT0) – Connect to ground w/10k resistor
- Pin 95 (Camera I^{2}C SCL) – Pull-up to VCC (3.3V) w/2.2k resistor
- Pin 96 (Camera I^{2}C SDA) – Pull-up to VCC (3.3V) w/2.2k resistor

Capacitors

- Pin 49 (VCAP1) – Connect a 2.2μF capacitor to ground
- Pin 73 (VCAP2) – Connect a 2.2μF capacitor to ground
- Connect a 47μF capacitor from VCC (3.3V) to ground

Electrical characteristics

The ALCAM SoC is based on STM32F407VE . Consult the processor datasheet for electrical characteristics as needed.

USB

ALCAM supports USB drives along with SD cards for storage. Only full-speed (12Mbps) mode is supported. If you plan to use USB, you must add a 22 ohm resistor in series with each of the following pins:

- Pin 70 (Function 10) – USB D-
- Pin 71 (Function 11) – USB D+

11. ALCAM OEM

GHI Electronics offers the ALCAM OEM board that use the ALCAM SoC. This board exposes all of the needed signals to interface with ALCAM over UART, SPI or I2C and provide a convenient connector for SD. The board is easily mountable on existing or new products.

ALCAM OEM Board

The ALCAM OEM Board is designed to allow projects to incorporate the full capabilities of ALCAM SoC with a minimal number of components and the ability for direct human control.

Features of the ALCAM OEM board include:

- 3.3V regulator, input 5V.
- Micro SD card reader.
- Standard 2.5mm remote camera control interface and button for triggering standalone mode
- Boot mode select pins allowing for firmware updates.
- Standard FFC 24pin 0.5mm pitch sensor connector.
- 2 LEDs to convey camera state.
- Standard 3.3V IO pins.
- On-board crystal oscillator for RTC.
- Full speed USB

OEM Pinout

ALCAM OEM Interface Pin	Description
1	GND
2	BUSY
3	+5V
4	UART3_RX/I2C2.SDA
5	UART3_TX/I2C2.SCL
6	SPI.MISO
7	SPI.MOSI

ALCAM OEM Interface Pin	Description
8	SPI.CLK
9	SPI.SSEL
10	FUNCTION 1
11	FUNCTION 2
12	FUNCTION 3
13	FUNCTION 4
14	FUNCTION 5/STANDALONE MODE
15	FUNCTION 6
16	FUNCTION 7
17	FUNCTION 8
18	FUNCTION 9
19	FUNCTION 10/USB.D-
20	FUNCTION 11/USB.D+
21	VBAT
22	FUNCTION 12
23	RESET

SD Connector Pinout

ALCAM SD card Interface Pin	
1	Vcc
2	N/C
3	SD_CD
4	SD_D0
5	SD_D1
6	SD_CMD
7	SD_D2
8	SD_D3
9	SD_CLK
10	GND

Error LED

The error LED will blink a number of times when something has gone wrong when using standalone mode, as shown below. If the LED keeps blinking, then a movie is being recorded.

Number of blinks	Description
1	Capturing images using the standalone function.
3	The camera has not been initialized (I command)
4	There is no SD card or USB drive connected.
5	The SD card or USB drive has not been initialized (I command)
6	An error occurred during movie recording
7	Opening a file failed.
8	The operation failed.

Status LED

The status LED will be on if the ALCAM is ready to capture or record. If it is off, it means the camera module is bad or there is no SD card or USB drive connected or they are not initialized.

If the status LED is blinking quickly, it means the configuration was successfully loaded and applied. You should restart the ALCAM to use the new values. If it is blinking once per second, it means the configuration failed to update. The configuration will reset to the defaults once you restart the ALCAM.

Push Button and 2.5mm Jack

The ALCAM OEM board has a push button and 2.5mm remote camera trigger jack. The push button and a hard press of the remote trigger are both connected to function 5 (standalone mode). The light press of the remote trigger is connected to function 12 and is currently not implemented.

12. Performance

Selecting the right storage media

The current storage media market is flooded with low grade devices. These devices may work with a PC but that doesn't mean the device follows standards and will work with ALCAM. Also, other devices may have advanced features not suitable for embedded devices. For example, some USB memory drives have a built in USB hub. We make our best effort to support a wide range of storage media that follow published standards. But GHI Electronics does not guarantee that ALCAM will be able to access all storage media

For products using ALCAM SoC, it is important to test different media devices. Then maintain a list of supported media for a product. GHI Electronics does not recommend any specific brand but always recommends selecting a well known source. If media is not supported, a failure most often occurs at initialization (I command). Less often (rarely) the media will initialize but then have problems with reading and/or writing. If the media mounts with no errors, in most cases, it is safe to assume it will function normally.

Make sure the power source to the media is reliable and there is a large enough capacitor placed as close as possible to the media connector. We recommend adding 0.1uF and 22uF. Low power may result in slower performance.

SD cards are rated for speed; ALCAM can work with Class 4 cards, however, Class 6 or, better yet, Class 10, will perform the best.

File Access Speed

There are many factors that affect file access speeds. Some storage media devices have internal buffering, others have high speed rating. But even on the exact same media, speeds might differ between different tests. Factors such as fragmentation, media life and voltage may affect the speed on the same media.

Fragmented storage runs slower because the system needs to spend more time, or even read more sectors from the FAT table, to find the needed cluster. Formatting the media should take care of this fragmentation.

Also, storage access speed decreases when the storage get closer to the end of life. The time needed to erase sectors increases over usage time.

13. Result Codes

Result-codes are always strings with 2 characters.

CODE VALUE	STRING
00	SUCCESS
01	ERROR_UNKNOWN_CMD
02	ERROR_INCORRECT_PARAMETER
03	ERROR_OPERATION_FAILED
5	ERROR_MEDIA_FULL
10	ERROR_NO_OR_NOT_SUPPORT_CAMERA
11	ERROR_CAM_NOT_INIT
12	ERROR_INVALID_FRAME
13	ERROR_BMB_SIZE_TOO_LARGE
14	ERROR_INCORRECT_CFG_LEVEL_PARAMETER
16	ERROR_OV2640_NOT_SUPPORT
17	ERROR_AF_NOT_SUPPORT
18	ERROR_AF_NOT_LOADED
19	ERROR_AF_ERROR
20	ERROR_SD_NOT_DETECT
21	ERROR_SD_NOT_INIT
22	ERROR_OPEN_FILE_FAILED
23	ERROR_INVALID_FILENAME
30	ERROR_MOVIE_NOT_RECORDING
31	ERROR_MOVIE_RECORDING
32	ERROR_CMD_ONLY_IN_STOPMOTION_MODE
33	ERROR_TIMELAPSE_DELAY_VALUE_TOO_SMALL
34	ERROR_TIMELAPSE_DELAY_VALUE_TOO_LARGE
35	ERROR_MOVIE_NOT_ACCEPT_BMP_FORMAT
40	ERROR_USB0_NOT_CONNECT
41	ERROR_USB0_NOT_INIT
50	ERROR_FILE_OPEN_FAILED
51	ERROR_FILE_CLOSE_FAILED

52	ERROR_FILE_WRITE_FAILED
53	ERROR_FILE_READ_FAILED
54	ERROR_FILE_NOT_OPEN
55	ERROR_FILE_NOT_FOUND
56	ERROR_FILE_DELETE_FAILED_OR_FILE_NOT_FOUND
57	ERROR_USB_SD_NOT_INIT
58	ERROR_LAST_FILENAME_NOT_SAVED
1A	ERROR_ZOOM_LEVEL_NOT_SUPPORTED

14. Revision History

Rev No.	Date	Modification
0.02	02/10/15	Updated supported image resolutions.
0.01	02/06/15	Preliminary.

15. Legal Notice

Disclaimer

IN NO EVENT SHALL GHI ELECTRONICS, LLC. OR ITS PARTNERS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS PRODUCT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT ANY NOTICE. GHI ELECTRONICS, LLC LINE OF PRODUCTS, INCLUDING ALCAM, ARE NOT DESIGNED FOR LIFE SUPPORT APPLICATIONS.