OV528

Single Chip Camera to Serial Bridge

ShangHai OmniVision IC Design, Inc.

July 18, 2002

Preliminary

User Manual

Rev. 1.1

ShangHai OmniVision IC Design, Inc. reserves the right to make changes without further notice to any product herein to improve reliability, function or design. OmniVision does not assume any liability arising out of the application or use of any project, circuit described herein; neither does it convey any license under its patent nor the right of others.

This document contains information of a proprietary nature. None of this information shall be divulged to persons other than ShangHai OmniVision IC Design, Inc. employee authorized by the nature of their duties to receive such information, or individuals or organizations authorized by ShangHai OmniVision IC Design, Inc.

Table of Contents

1. Introduction 1

1.1 OV528....................2

1.2 Sensors 2

1.3 Boards 2

1.3.1 Evaluation Board.................... 2

1.3.2 Demo Board....................2

2. DownLoad / Update Firmware....................5

3. Software User Guide....................7

3.1 System Installation....................7

3.2 Initialization. 7

3.3 Functions Overview....................8

3.4 View Group. 9

3.5 Snap Shot Group....................10

3.6 DSC Group 11

3.7 Cursor & Mouse Messages....................12

4. Software Develop Guide....................13

4.1 Data Structure 13

4.1.1 Sensor Type....................13

4.1.2 Color Type....................13

4.1.3 JPEG Image Size....................13

4.1.4 Bitmap Image Size (Preview Size) 14

4.1.5 Preview or Snapshot Image Information (SELECT_PARAM) 14

4.1.6 Flash Memroy File Information(PFlash_FileTableItem) 14

4.1.7 Display Image Information (only in OV528Ap.exe) 15

4.2 Function Modules 15

4.2.1 Commands....................15

4.2.2 Download or Update Firmware....................20

4.2.3 System's Initialization, Start, Stop and Reset (Exit)....................22

4.2.4 Preview of Live Video....................26

4.2.5 SnapShot of Still JPEG Image....................27

4.2.6 DSC or Flash Memory Functions....................29

4.2.7 JPEG Image Decoding....................34

4.2.8 Other Functions....................36

4.2.9 Threads and Synchronization....................39

Index of functions....................40

References....................40

1. Introduction

This document aims at providing a guideline to the user or developer of OV528 system in case the system will be maintained and developed.

1.1 OV528

OV528 is the back-end chip for the Serial Bus Camera System that can be attached to a wireless or PDA host and performs as a video camera or a JPEG compressed still camera. It provides camera (sensor) interface, serial interface and JPEG image compression engine to act as a low cost and low powered single chip solution for the high-resolution serial bus PDA or cellular phone camera accessory applications (See Fig. 1-1).
For detailed information of OV528, please refer [1].

1.2 Sensors

So far, OV528 supports four kinds of image sensors: OV7620, OV7635, OV6630 and OV6640.
OV7620 and OV7635 are CMOS digital video camera sensors with VGA (640x480) or QVGA (320x240) resolution. The frequency of the output frame is up to 30 Hz (VGA). Different data formats, YCrCb 4:2:2, GRB 4:2:2 and RGB raw data, are supported. 16Bit or 8Bit video data, namely ZV Port, CCIR601 or CCIR656 are outputted.
OV6630 and OV6640 image sensors provide CIF (352x288) or QCIF (176x144) data output format. Maximum FPS (frame per second) can achieve to 60.
Some advanced and attractive functions are also integrated into the sensors, such as the exposure control, gamma correction, gain control, automatic white balance, blooming's drastically reduction, etc. All these functions or algorithms together guarantee the output images' quality to reach the satisfactory level.
For more information and specifications about the image sensors, please refer materials [2] – [5].
Note: The supplied powers for these four types of sensors are not the same. OV7620 needs 5VDC to support while the other three types need 3.0-3.6 VDC. Therefore, before a sensor is connected with a board, please make sure the power provided by board is correct.

1.3 Board

1.3.1 Evaluation Board

Jumpers on evaluation board must be correctly set before any field usage. Table 1-1 gives a reference of the jumper and button positions. Fig.1-2 shows the sketch map of jumpers and buttons on OV528 evaluation board.

1.3.2 Demo Board

User should remove R20 off and add R13 (10K) on demo board to down load firmware. If succeed, reposit them to run the system.

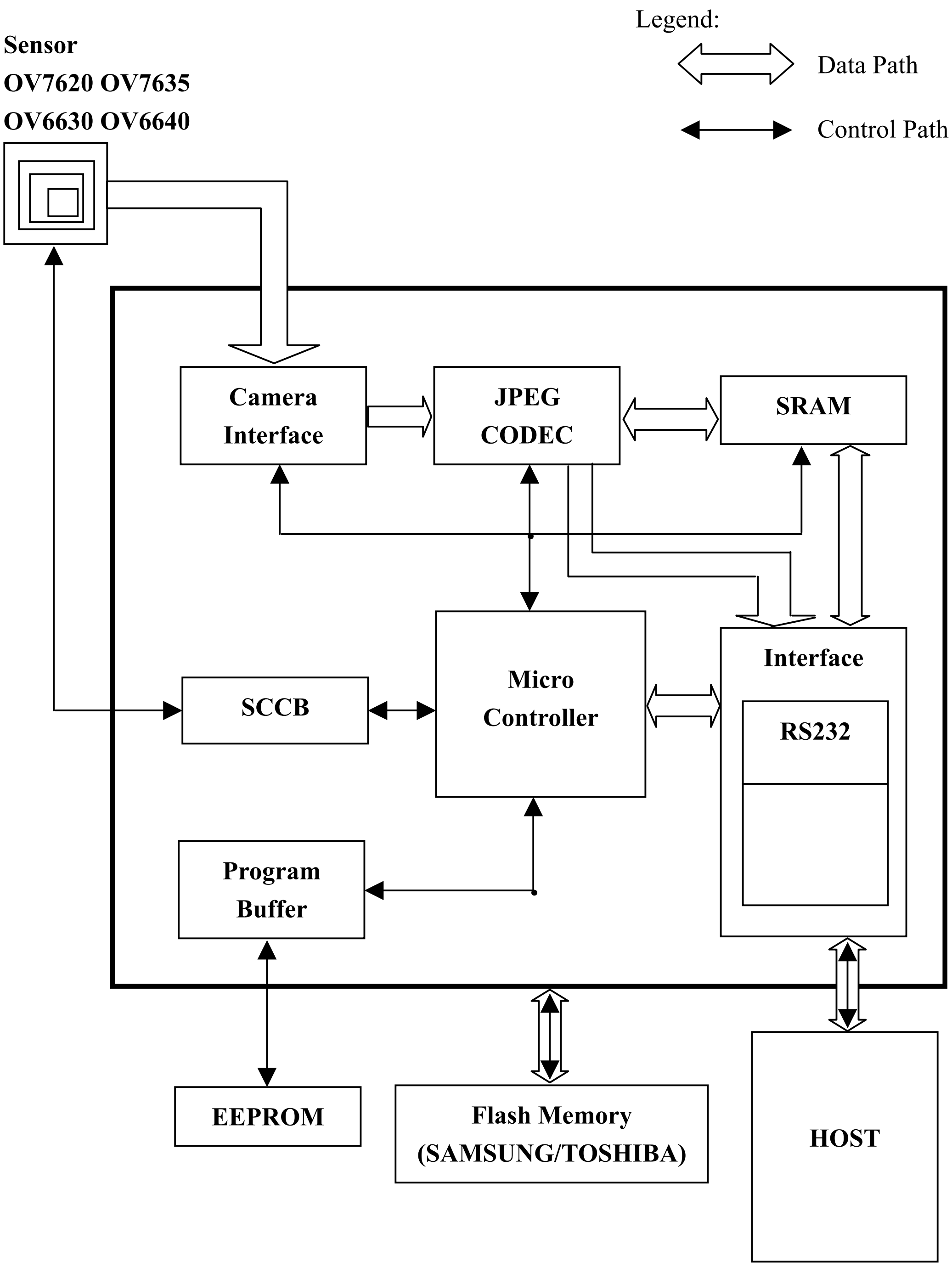
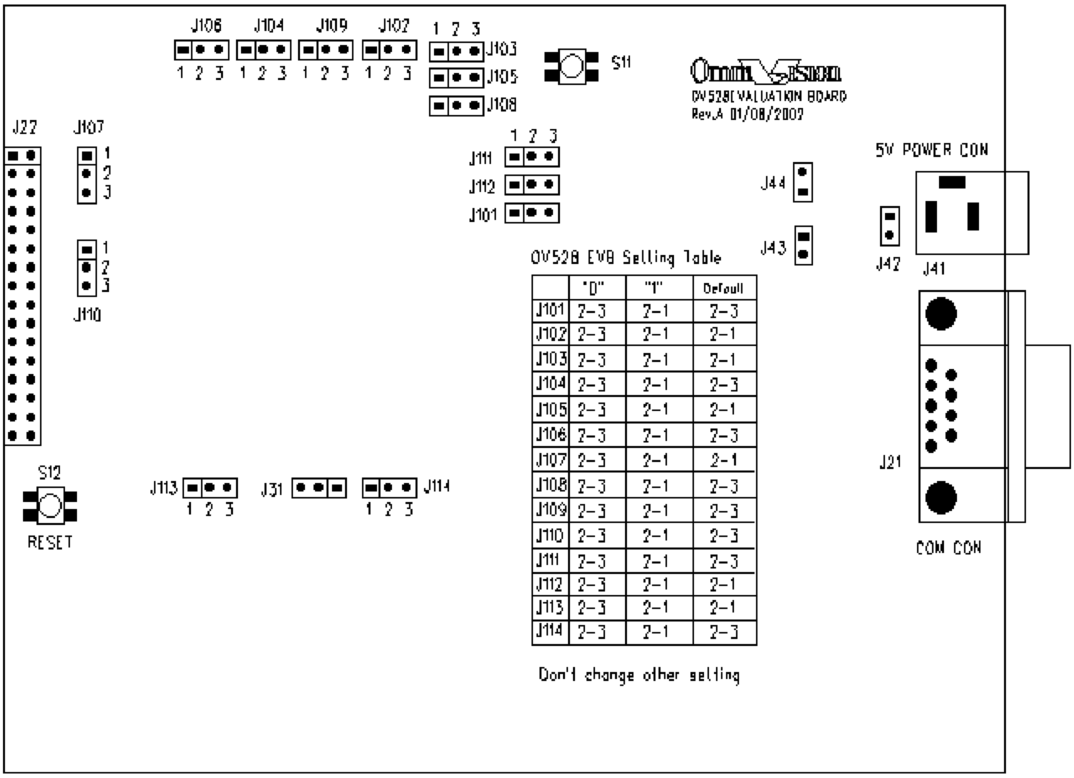


Fig. 1-1 OV528 Function Block Diagram



OV528 EVB Setting Table

	"0"	"1"	Default
J101	2-3	2-1	2-3
J102	2-3	2-1	2-1
J103	2-3	2-1	2-1
J104	2-3	2-1	2-3
J105	2-3	2-1	2-1
J106	2-3	2-1	2-3
J107	2-3	2-1	2-1
J108	2-3	2-1	2-3
J109	2-3	2-1	2-3
J110	2-3	2-1	2-3
J111	2-3	2-1	2-3
J112	2-3	2-1	2-1
J113	2-3	2-1	2-1
J114	2-3	2-1	2-3

Don't change other setting

Fig. 1-2 Sketch Map of Jumpers and Buttons on OV528 Evaluation Board

Table 1-1. Jumper and Button Positions of OV528 Evaluation Board

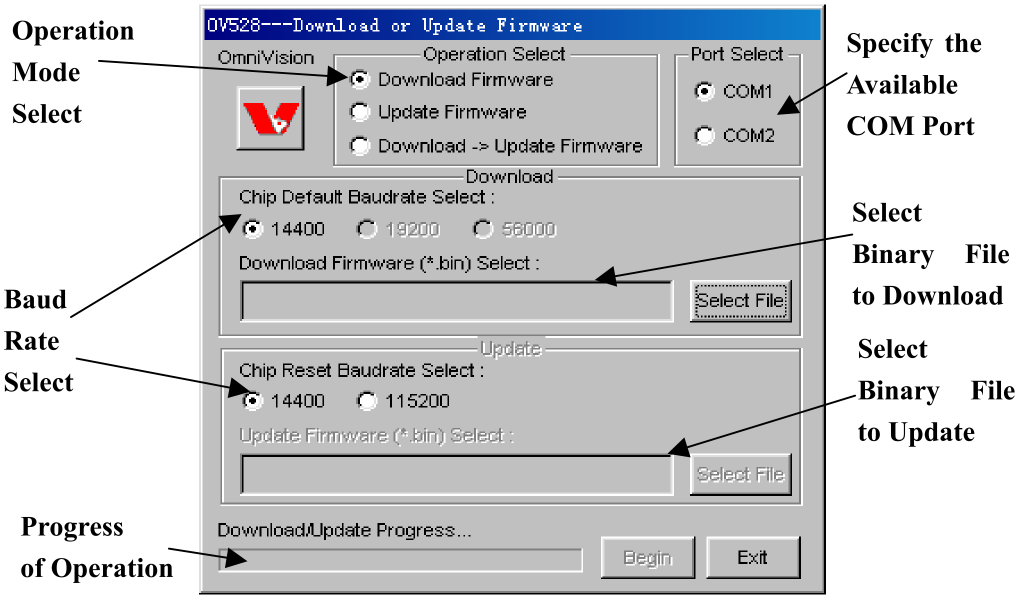
Description of Functions	Relevant Jumpers or Buttons	Remarks
All Functions	J 101->'0' J 102->'1' J 103->'1' J 104->'0' J 105->'1' J 106->'0' J 107->'1' J 111->'0' J 113->'1'	These 9 jumpers must be set on the given positions regardless the other conditions that the system works under.
Boot Up the System / Down Load Firmware	J 108->'0' / J108->'1'	Firmware is stored in EEPROM.
RS232 serial bus / 4 wires serial bus	J 109->'0' / J109->'1'	If use RS232, to short-circuit J1C pin 3 and pin 4
Internal Microprocessor / External Microprocessor	J 114->'0' / J114->'1'	
Reset	Button S12	Reset the whole system after power on
Snap Shot	Button S11	Get a still image and save it into flash memory
External Power Jet	J41	5VDC

2. DownLoad/Update Firmware

If it's the first time to use the OV528 system, firmware should be downloaded into the EEPROM at first. While, if an old version firmware has already been in the EEPROM and a newer one exists, user can update the old one by the new firmware (in the latter situation, downloading can also be effective but we don't recommend user to do so).

An application program named OV528DL.exe is used to undertake such work.

Double click the ICON of OV528DL.exe to run the program, the main interface will look like Fig.2-1:



OV528---Download or Update Firmware

Operation Select: ● Download Firmware ○ Update Firmware ○ Download -> Update Firmware

Port Select: ● COM1 ○ COM2

Download — Chip Default Baudrate Select : ● 14400 ○ 19200 ○ 56000

Download Firmware (*.bin) Select :

Update — Chip Reset Baudrate Select : ● 14400 ○ 115200

Update Firmware (*.bin) Select :

Download/Update Progress...

Fig. 2-1 Main Interface of DownLoading or Updating Firmware Program

Operation Mode:

1. **Download Firmware**: download firmware into OV528, if it's valid, save it into EEPROM.
2. **Update Firmware**: update the old firmware in EEPROM with a new one.
3. **DownLoad->Update Firmware**: download a firmware into OV528, then use it to save a different firmware file into EEPROM. In fact, except the source file (being downloaded) is different from the object file (being updated), mode 3 is the same with mode 1.

If it is downloading firmware, user should set Jumper J108 at position '1', otherwise, set it at '0' (refer Table 1-1 and Fig. 1-1).

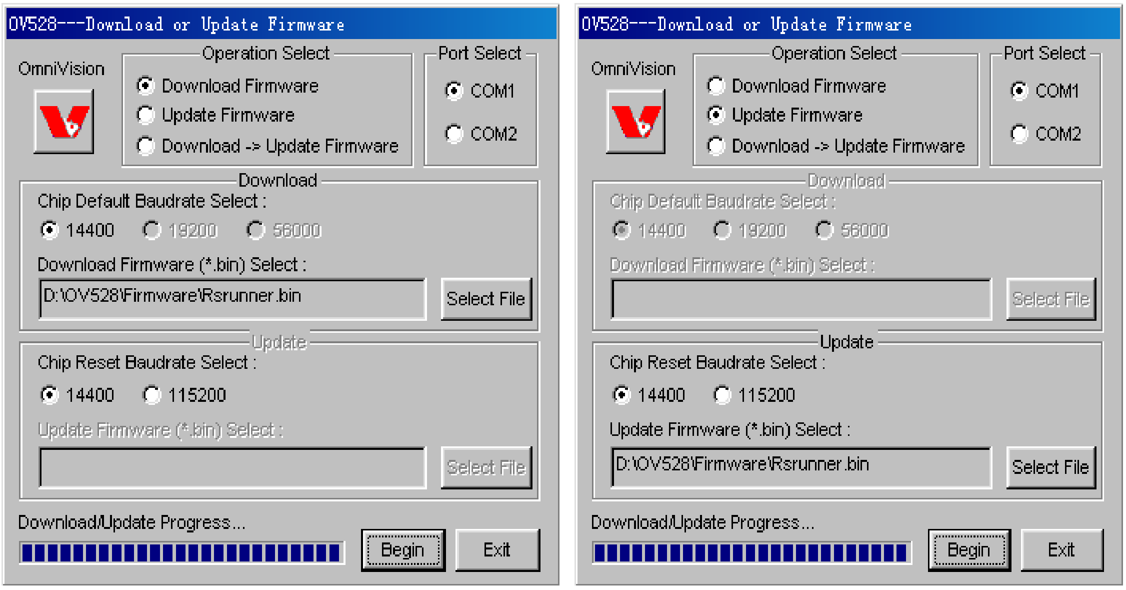
Besides this, user also needs to do some additional things before downloading or updating the firmware

1. select the binary firmware file with the name *.bin

   If the operation mode is '**Download->Update Firmware**', user needs to select two *.bin files, one is to be downloaded into OV528, the other is to be saved into EEPROM.
2. select an available COM (COM1 or COM2) port through which host can communicate with the firmware

3. select the supported baud rate, default one is 14400 for both downloading and updating.

Only after all the three requirements mentioned above are met, can the operation be carried out (See Fig. 2-2), otherwise, error message will appear (See Fig. 2-3).



OV528---Download or Update Firmware

OmniVision

Operation Select
● Download Firmware
○ Update Firmware
○ Download -> Update Firmware

Port Select
● COM1
○ COM2

Download
Chip Default Baudrate Select :
● 14400 ○ 19200 ○ 56000
Download Firmware (*.bin) Select :
D:\OV528\Firmware\Rsrunner.bin

Update
Chip Reset Baudrate Select :
● 14400 ○ 115200
Update Firmware (*.bin) Select :

Download/Update Progress...

OV528---Download or Update Firmware

OmniVision

Operation Select
○ Download Firmware
● Update Firmware
○ Download -> Update Firmware

Port Select
● COM1
○ COM2

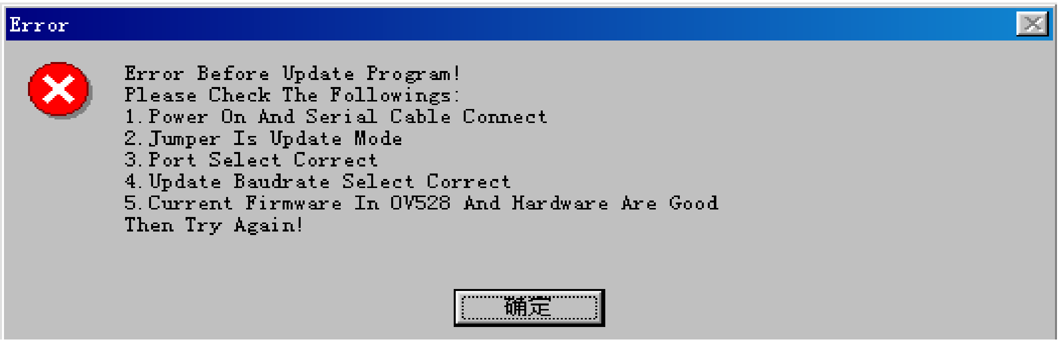
Download
Chip Default Baudrate Select :
● 14400 ○ 19200 ○ 56000
Download Firmware (*.bin) Select :

Update
Chip Reset Baudrate Select :
● 14400 ○ 115200
Update Firmware (*.bin) Select :
D:\OV528\Firmware\Rsrunner.bin

Download/Update Progress...

a. Download Firmware into OV528   b. Update Firmware in EEPROM

Fig. 2-2 Begin Download or Update Operation

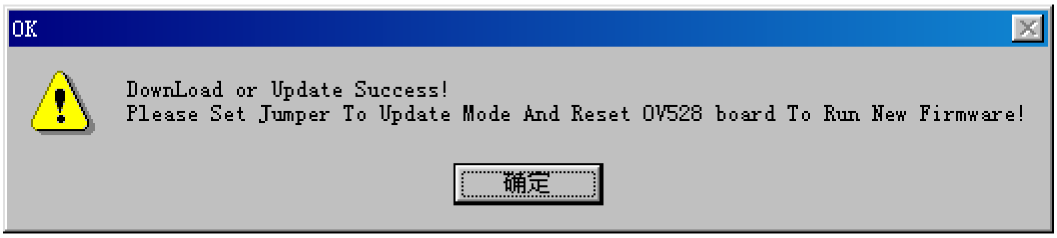


Error

```
Error Before Update Program!
Please Check The Followings:
1.Power On And Serial Cable Connect
2.Jumper Is Update Mode
3.Port Select Correct
4.Update Baudrate Select Correct
5.Current Firmware In OV528 And Hardware Are Good
Then Try Again!
```

Fig. 2-3 Error Occurs before Updating Firmware

If an error message box like Fig.2-3 is shown, user should follow the instructions to find out the problem. If not, Fig. 2-4 tells user that the operation has been finished and the system is all set.



OK

```
DownLoad or Update Success!
Please Set Jumper To Update Mode And Reset OV528 board To Run New Firmware!
```

Fig. 2-4 Download or Update Success Message

Note: After successfully downloading the firmware, J108 should be placed at position ‘0’ (refer Table 1-1). Then push S11 (Reset button) to reset the system before running the application program.

3. Software User Guide

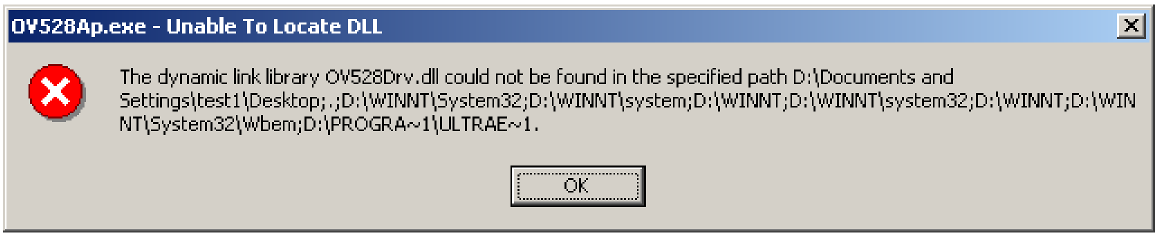
Software will automatically recognize which one is in use when the system works up. Considering the flash memory is only an optional function of the system, we will also check the system before showing the Main Interface to make sure whether it is supported or not.

3.1 System Installation

The software of the system contains three files:

1. OV528Ap.exe : application program
2. OV528Drv.dll : dynamic link library, driver for OV528Ap.exe
3. JPEGDecoder.dll: dynamic link library, decode JPEG image into Bitmap file.

Copy these three files to the same folder then double click OV528Ap.exe to run the program. The two DLLs can also be placed under the window's default directories. If OV528Ap.exe can't find any of the two DLLs, an error message similar with Fig. 3-1 will appear (under Windows 2000).



OV528Ap.exe - Unable To Locate DLL

The dynamic link library OV528Drv.dll could not be found in the specified path D:\Documents and Settings\test1\Desktop;.;D:\WINNT\System32;D:\WINNT\system;D:\WINNT;D:\WINNT\system32;D:\WINNT;D:\WINNT\System32\Wbem;D:\PROGRA~1\ULTRAE~1.

Fig. 3-1 Application Program Can't Find the DLL

3.2 Initialization

The first of all things that the software will do is to initialize the Serial Interface (COM or SPI) of the Host (PC, PDA, Cell-phone, etc.) in order to set up the communication way between host and OV528.

When searching the available communication port and the supported baud rate, a little time will be cost. Therefore, the driver will create a processing bar to show the status.

Unfortunately, sometimes the function may fail to initialize the system. In this case, a dialog box (see Fig.3-2) will show the error message.

If this happens, please push S12 (**Reset button**) then run OV528Ap.exe once again.

If the problem still exists, please make sure:

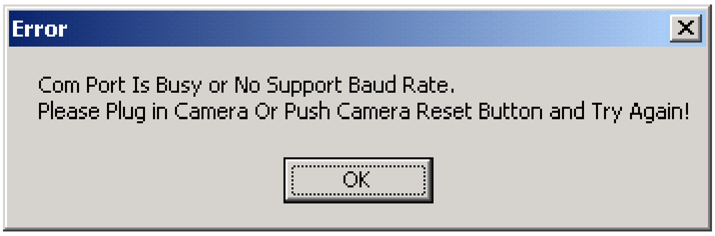
a. External power has been correctly connected with the board (5V DC)

b. The mounted sensor is compatible with the board (For OV7620, the sensor's input power is 5V DC, for others, the input power is 3.3V DC. Just because of this reason, there is a minor difference between the boards as for the different sensor type.)

c. RS-232 cable has been correctly established between the board and host

d. At last one of COM1 and COM2 ports is unoccupied and available.

Therefore, sometime user needs to check and reset the system for several more times.



Error

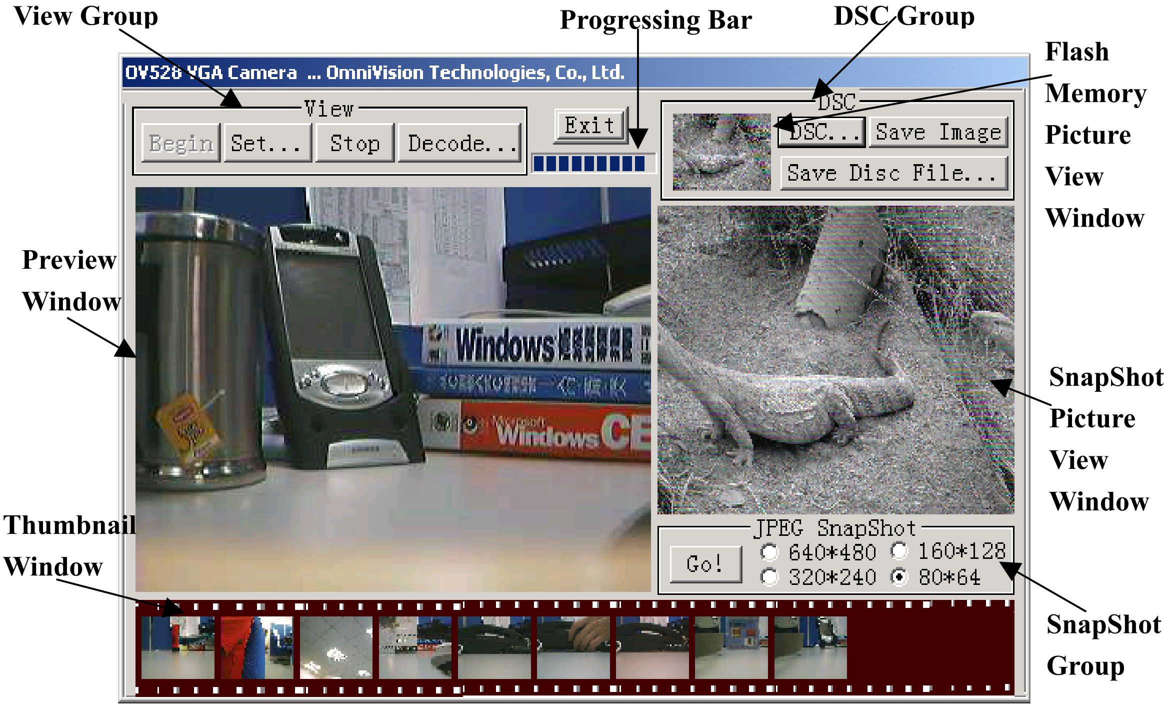
Com Port Is Busy or No Support Baud Rate.
Please Plug in Camera Or Push Camera Reset Button and Try Again!

Fig.3-2 Initialize Fail Message

3.3 Functions Overview:

If the hardware supports the flash memory and the sensor type is OV76#0 (VGA), the appearance of the user interface will look like Fig.3-3.

Images will be showing in the **Preview Window** frame by frame once the system has been booted up. It will differ in the frame rate when user chooses different image formats or sizes. The **progressing bar** indicates the progress of image's loading operation.



OV528 VGA Camera ... OmniVision Technologies, Co., Ltd.

JPEG SnapShot
○ 640*480 ○ 160*128
○ 320*240 ● 80*64

Fig.3-3 Main Interface of VGA Sensor (with Flash Memory Support)

A still image will be displayed in the **SnapShot Picture View Window** if user snapshots an image. Meanwhile, a copy of this image will appear in the **Thumbnail Window** in a smaller size. Totally, there can be 11 such small images being displayed within the bound of the Thumbnail Window at most. Which means that when more images are got, latter ones will overwrite the former images. For convenience, we give them the name **Thumbnail**. User can review any one of these 11 thumbnails by clicking it with the cursor. Then, the image interest you will appear in the **SnapShot Picture View Window** again.

If you want to save the image in any small thumbnail into a file, you can double click with your mouse's left button on it, then, a JPEG file will be generated, the file's name is shown to you in a message box.

The image in the **DSC** group is the one that you load from the flash memory. In **Flash Memory Picture View Window**, you will view the last picture you load from the flash memory in a minified size. By clicking it with the cursor, it can also be displayed in the **SnapShot Picture View Window** where you can see more details of it**.**

If the sensor type is OV66#0 (CIF) and the flash memory is supported, the Main Interface will be another one (see. Fig.3-4). Except the previewing and still images' sizes, all the other functions are the same with that of VGA system.

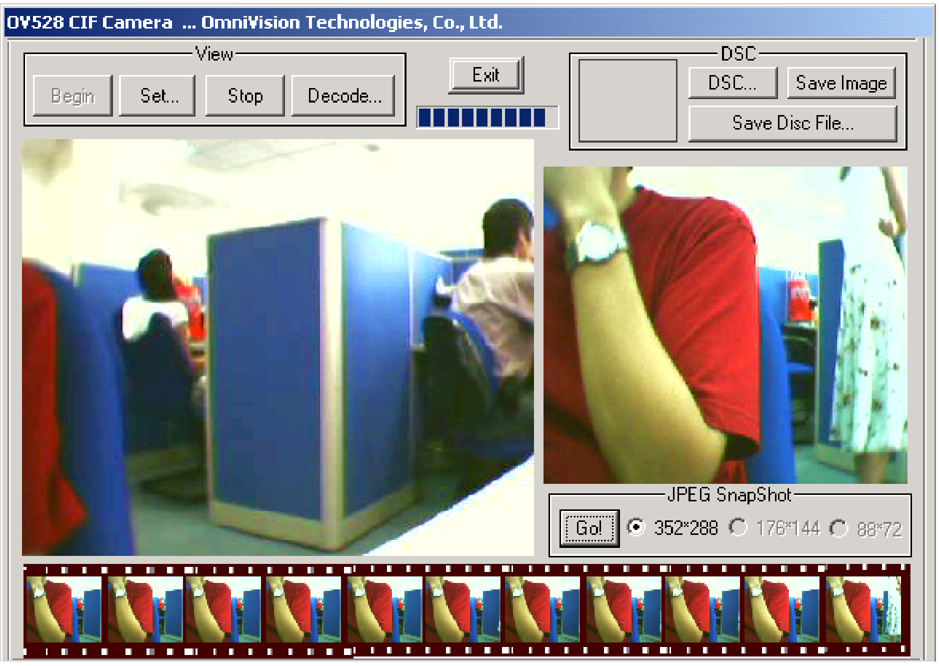


Fig.3-4 Main Interface of CIF Sensor (Flash Memory Support)

3.4 View Group

From the main dialog, it can be easily found that there are four function modules in this group. When the button **Begin** is pressed down, live images will appear within the preview window frame by frame. Limited by the window's area, some image's showing size (maximum is 320*240) may be different with its real size (maximum is 640*480). When the button **Set** is chosen, a dialog containing the data structures will be popup out, all the supporting parameters and color types are listed in it (See Fig.3-5). The '**SnapShot Image Resolution**' in this dialog is the resolution of images got by '**SNAP**' (button **S11** on evaluation or demo board).

The **Stop** button is to be used to stop the displaying operation immediately.

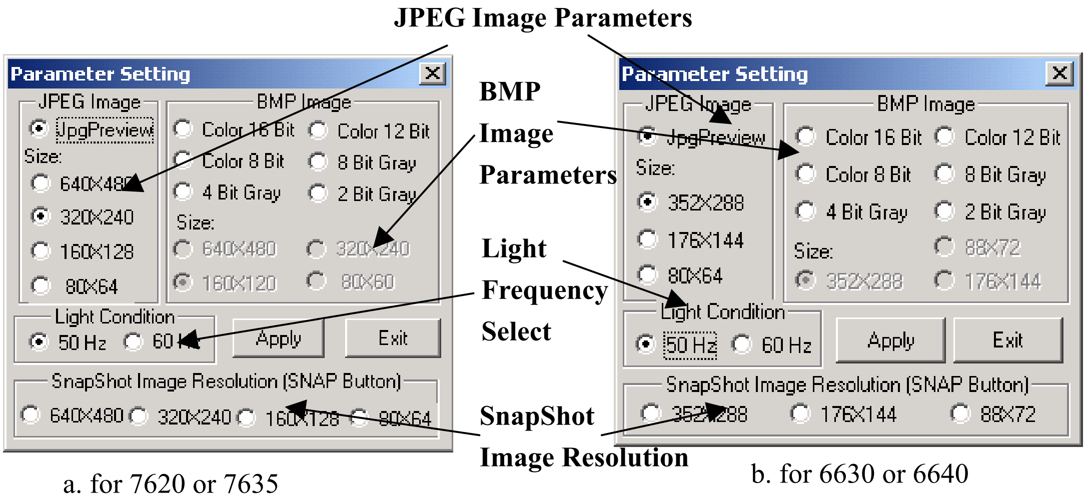
The last one, namely the **Decode** button corresponds to the function of decompressing some JPEG image into the Bitmap files through hardware.

As for the **Decode** function, more words are needed.

At first, user should select a JPEG file from the hard disc as the source image (Fig.3-6). Then the software will check several domains of the file header to make sure that file is a '**VALID**' one.

Note: word **'VALID'** mentioned above refers to the following situations:

1. Source file's size must be one of the following 4 kinds: 320*240, 640*480, 176*144, 35*288.
2. JPEG files must be those being created from OV528 system especially those generated by



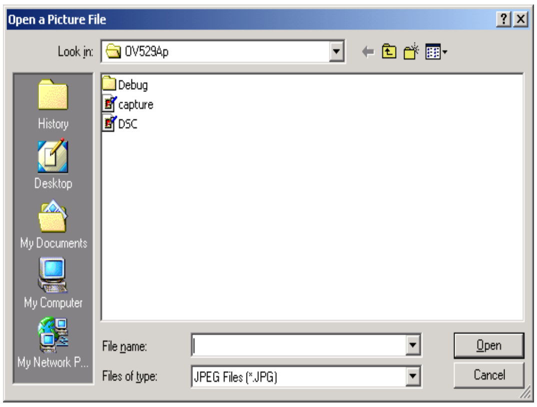
Parameter Setting
JPEG Image: ● JpgPreview
Size: ○ 640X480 ● 320X240 ○ 160X128 ○ 80X64
BMP Image: ○ Color 16 Bit ○ Color 12 Bit ○ Color 8 Bit ○ 8 Bit Gray ○ 4 Bit Gray ○ 2 Bit Gray
Size: ○ 640X480 ○ 320X240 ● 160X120 ○ 80X60
Light Condition: ● 50 Hz ○ 60 Hz
Apply Exit
SnapShot Image Resolution (SNAP Button): ○ 640X480 ○ 320X240 ○ 160X128 ○ 80X64

a. for 7620 or 7635

Parameter Setting
JPEG Image: ● JpgPreview
Size: ● 352X288 ○ 176X144 ○ 80X64
BMP Image: ○ Color 16 Bit ○ Color 12 Bit ○ Color 8 Bit ○ 8 Bit Gray ○ 4 Bit Gray ○ 2 Bit Gray
Size: ○ 88X72 ● 352X288 ○ 176X144
Light Condition: ● 50 Hz ○ 60 Hz
Apply Exit
SnapShot Image Resolution (SNAP Button): ○ 352X288 ○ 176X144 ○ 88X72

b. for 6630 or 6640

Fig 3-5. Sensor Parameters' Setting Dialog



Open a Picture File

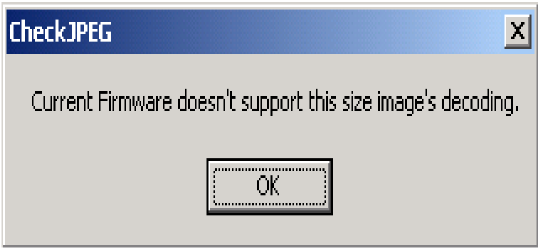
Look in: OV529Ap

Debug
capture
DSC

File name:

Files of type: JPEG Files (*.JPG)

Fig.3-6 Select a JPEG file



CheckJPEG

Current Firmware doesn't support this size image's decoding.

OK

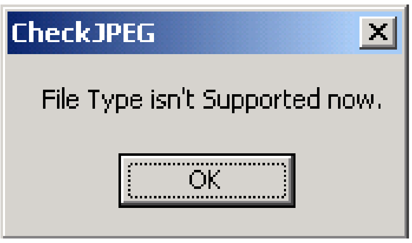
Fig.3-7 Invalid File Size Message

'SnapShot'. That means some other types even the standard JPEG files can't meet our demands at the moment. Otherwise, an error message will be posted and the function will return FALSE (Fig.3-7 and Fig.3-8).

If the file meets the standard, a dialog box (Fig.3-9) will be created to inform the user to select the destination Bmp file's properties. And, of course, the decoded file's size must be no larger than the source file's.

3.5 Snap Shot Group

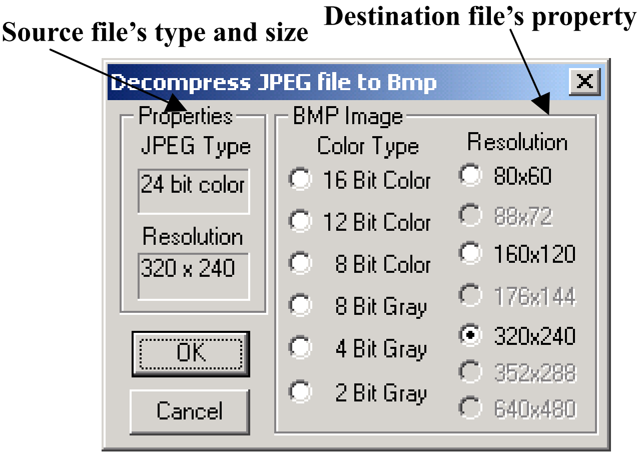
This group contains one button and several radios. Button **Go!** is the entry of functions to get



CheckJPEG

File Type isn't Supported now.

Fig.3-8 Invalid File Type message



Decompress JPEG file to Bmp

Properties
JPEG Type: 24 bit color
Resolution: 320 x 240

BMP Image
Color Type: ○ 16 Bit Color ○ 12 Bit Color ○ 8 Bit Color ○ 8 Bit Gray ○ 4 Bit Gray ○ 2 Bit Gray
Resolution: ○ 80x60 ○ 88x72 ○ 160x120 ○ 176x144 ● 320x240 ○ 352x288 ○ 640x480

Fig.3-9 Set the destination file properties

a compressed picture from the hardware and

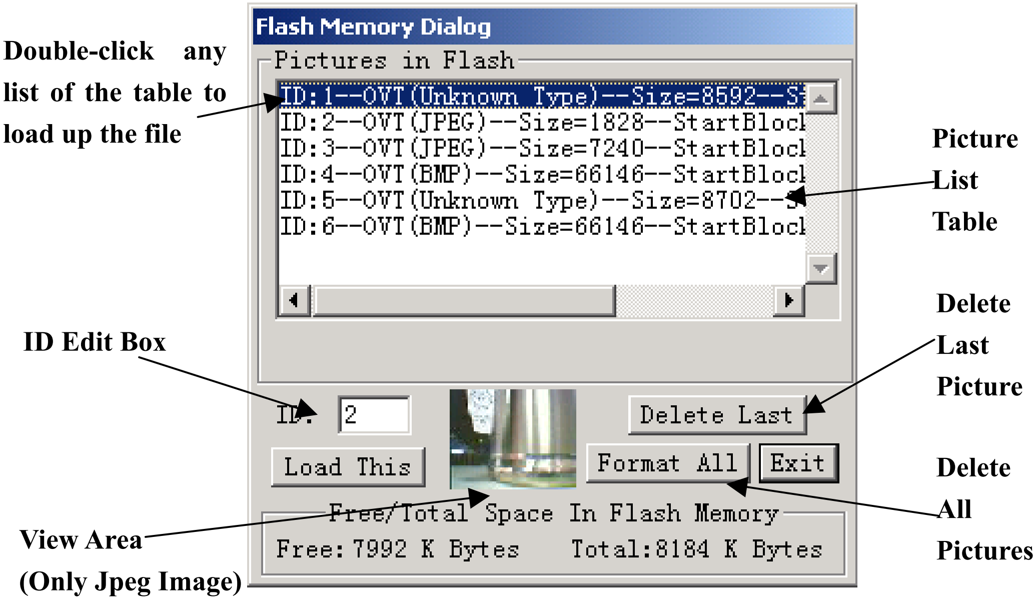
1. display the decompressed still Bmp image on the snap shot picture view window;
2. save the decompressed image data into file '**snapshot.jpg**'. The size of the file is determined beforehand by choosing one of the four radios (VGA mode) or one of the three radios (CIF mode).

Also don't forget to specify a size from the snapshot group before 'Go!'.

3.6 DSC Group

When **DSC** button is pressed down, Flash Memory Dialog will appear (See Fig.3-10). From the dialog, you can view a table that contains all the flash memory images' information, or you can delete all of them by pressing **Format ALL** button or the last one by **Delete Last** (of course, you must think twice befor3e doing so, though it's just a small deal).

It provides two ways to load an image from the flash memory.



Double-click any list of the table to load up the file
ID Edit Box
View Area (Only Jpeg Image)
Picture List Table
Delete Last Picture
Delete All Pictures

Flash Memory Dialog

Pictures in Flash

```
ID:1--OVT(Unknown Type)--Size=8592--S
ID:2--OVT(JPEG)--Size=1828--StartBlocl
ID:3--OVT(JPEG)--Size=7240--StartBlocl
ID:4--OVT(BMP)--Size=66146--StartBlocl
ID:5--OVT(Unknown Type)--Size=8702--S
ID:6--OVT(BMP)--Size=66146--StartBlocl
```

ID: 2

Free/Total Space In Flash Memory
Free: 7992 K Bytes Total:8184 K Bytes

Fig.3-10 Flash Memory Picture View Dialog

1. When you **double click** an item listed in the table, the ID of this item will appear in the **ID Edit Box,** at the same time, the real image (JPEG type) will show in the **View Area**. So far, the image

has been loaded out even you exit the dialog now.
2. You can also put the ID number into the **ID Edit Box** then by pressing the **Load This** button to load the image with the given ID number.
The foregoing two ways can produce the same effect.
The total memory of the flash is 8 Mega bytes and has been divided into 1000 blocks. So each lock's size is 8 K bytes.
Save Image button of DSC group corresponds to the function of saving the current live image into the flash memory. A message box will appear after this operation to inform whether it has been successfully finished or not.
Unlike **Save Image**, **Save Disc File** will save a disc file rather than a current displayed image into the flash memory. Thus, an object file must be specified by the user at first (See Fig.3-11).

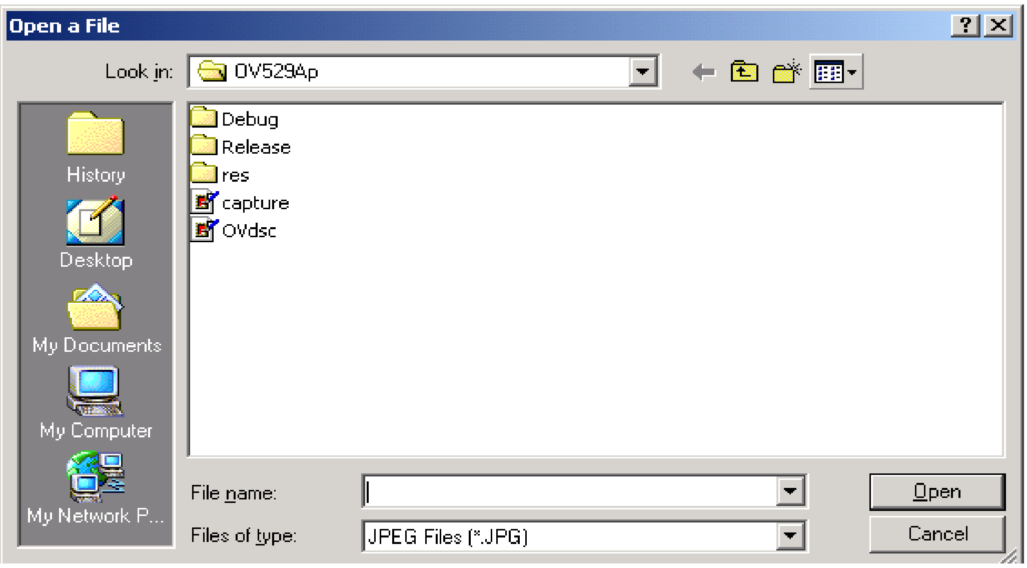

Open a File
Look in: OV529Ap
Debug
Release
res
capture
OVdsc
File name:
Files of type: JPEG Files (*.JPG)

Fig.3-11 Select a File to Down Load into Flash Memory

3.7 Cursor & Mouse Messages

As we know, when you click or double-click with your mouse on some given region, messages are created. Windows will capture the messages and send them to the program or to be more precisely, to the windows procedure.
Such regions that defined in our program and response to the messages together compose the '**Flash Memory Picture View Window**' and '**Thumbnail Window**'. Totally, there are 12 such small regions which we call 'thumbnail'.
If there is image within any of these 12 thumbnails you can click or double-click with your mouse on it. In the former case, the image will be shown in the **SnapShot Picture View Window** and in the latter case, the image date will be saved into a JPEG file with the name 'thumbnail*.jpg' (Thumbnail Window) or 'OVDsc.jpg'(Flash Memory Picture View Window) under current directory.

Note: In thumbnail*.jpg, * is replaced by number from 0 to 10.

4. Software Develop Guide

In this part, we will give a detailed description of OV528 software's develop guide. Data structures, driver exported functions, parameters, commands and protocols (only in OV528Drv.dll) are explained. Base on the guide, users can deeply understand OV528 software and easily develop their own software.

4.1 Data Structure

4.1.1 Sensor Type

Table 4-1 Sensor Type

Sensor Type	Definition
OV7620	0x00
OV7635	0x01
OV6630	0x02
OV6640	0x03

4.1.2 Color Type

Table 4-2 Color Type

Color Type	Definition
Bitmap 2 bit gray	0x01
Bitmap 4 bit gray	0x02
Bitmap 8 bit gray	0x03
Bitmap 8 bit color	0x04
Bitmap 12 bit color	0x05
Bitmap 16 bit color	0x06
JPEG	0x07

4.1.3 JPEG Image Size

Table 4-3 JPEG Image Size

JPEG Image Size	Definition	Remarks
80x64 (VGA)	0x01	For OV7620 and OV7635 (VGA/QVGA) The height of the JPEG image must be the multiples of 16. That is why there are 128 and 64 instead of 120 and 60.
160x128	0x03	
320x240	0x05	
640x480	0x07	
80x64 (CIF)	0x02	For OV6630 and OV6640 (CIF/QCIF) Because of the same reason as in OV76##, there is 80x64 instead of 88x72.
176x144	0x04	
352x288	0x06	

4.1.4 Bitmap Image Size (Preview Size)

Table 4-4 Bitmap Image Size

Bitmap Image Size	Definition	Remarks
80x60	0x01	For OV7620 and OV7635 (VGA/QVGA)
160x120	0x03	
320x240	0x05	
640x480	0x07	
88x72	0x02	For OV6630 and OV6640 (CIF/QCIF)
176x144	0x04	
352x288	0x06	

4.1.5 Preview or Snapshot Image Information (SELECT_PARAM)

Table 4-5 Preview or Snapshot Image Information

Parameters	Types	Remarks
Color Type	UINT	One of the types in Table 4-2
Bitmap Image Size		One of the types in Table 4-4
JPEG Image Size		One of the types in Table 4-3

4.1.6 Flash Memory File Information (PFlash_FileTableItem)

Table 4-6 Flash Memory File Information

Parameters	Types	Remarks
File ID	WORD	Start from 1 to 511
File Name	Char[2]	
Reserved 1	DWORD	Reserved
Reserved 2	WORD	
Property	BYTE	File types (JPEG, Bitmap or Others)
Low-Byte of Start Block		8k bytes per block (Samsung and Toshiba)
High-Byte of Start Block		
Low-Byte of File Size		Support flash memory size: 2M x 8Bit ~ 32M x 8Bit, Single file's max. size is 16M. (for Samsung and Toshiba, the flash memory size is 8M)
Middle-Byte of File Size		
High-Byte of File Size		

4.1.7 Display Image Information (only in OV528Ap.exe)

Table 4-7 Display Image Information

Parameters	Types	Remarks
Bitmap Image Data Buffer	BYTE	To display image on 'Flash Memory Picture View Window' or 'Thumbnail Window'
JPEG Image Data Buffer	BYTE	
Bitmap Image Data Width	UINT	
Bitmap Image Data Height	UINT	
JPEG Image Data Size	DWORD	

4.2 Function Modules

Just like user guide, we divide the functions provided by the driver into several modules. Generally speaking, they are:

1. Download or update firmware
2. System's initialization, start, reset and stop
3. Preview of Live Video
4. Snapshot of Still JPEG Image
5. DSC or flash memory
6. JPEG image's decoding
7. Others

4.2.1 Commands

There are totally15 own-defined commands in OV528 system. OV528Drv.dll talks with firmware (default name is Rsrunner.bin) by means of these commands. With them, software and firmware have their own 'language'. Only after driver got in touch with firmware through commands, can the whole system run up. What is more, only after driver got the affirmative answer (again, it is command) from firmware, can any operation be successful.

Any command is composed with eight bytes.

Starting with three bytes of 0xFF, the fourth byte is the ID number of the command, following them are four parameters. Command's ID number ranges from 1 (0x01) to 15 (0x0F). It is used by firmware to identify which is which. The last four bytes, or namely parameters, are the content of the command. Therefore, any command will look like this: 0xFFFFFF0?P1P2P3 (? is from 1 to F).

Table 4-8 lists all the commands' name, ID and parameters.

Table 4-8 Command List

Command	ID	Parameter1	Parameter2	Parameter3	Parameter4
INITIAL	0x01	0x00	Color Type	Preview Resolution	JPEG Resolution
DUMP	0x02	0x00	0x00	0x00	0x00
SET REGISTER	0x03	Address of Register	Register Type	Data	0x00
GET PICTURE	0x04	Picture Type	File ID Low Byte / Color Type/0x00	File ID High Byte / Preview Resolution/0x00	0x00 / JPEG Resolution
SNAP SHOT	0x05	Snap shot Type	Skip frame Low Byte	Skip frame High Byte	0x00
SAVE DATA	0x06	Destination	Length Byte 0 / JPEG Resolution/0x00	Length Byte 1 / 0x00	File Property / Length Byte 2 / 0x00
SET BAUDRATE	0x07	Fist Divider	Second Divider	0x00	0x00
RESET	0x08	Reset Type	0x00	0x00	0xXX*
POWER OFF	0x09	0x00	0x00	0x00	0x00
DATA	0x0A	Data Type	Length Byte 0	Length Byte 1	Package Status / Length Byte 2
GET REGISTER	0x0B	Address of Register	Register Type	0x00	0x00
DOWNLOAD PROGRAM	0x0C	Destination	Length Byte 0	Length Byte 1	Length Byte 2
SYNC	0x0D	0x00	0x00	0x00	0x00
ACK	0x0E	Command ID	ACK counter	0x00	0x00
NAK	0x0F	0x00	NAK counter	Error Code	0x00

Note: * If the parameter is 0xFF, the command is a special Reset command and the firmware responds to it immediately.

Table 4-9 gives out the explanations of these commands.

Table 4-9 Command Explanation

Command	Explanation
INITIAL	Host sends this command to configure the camera. After receiving this command, camera programs its internal settings based on its own uC program. **Color Type**: Table 4-2 **Preview Resolution**: Table 4-4 **JPEG Resolution**: Table 4-3
DUMP	Camera will set its registers with default values when receives this command.

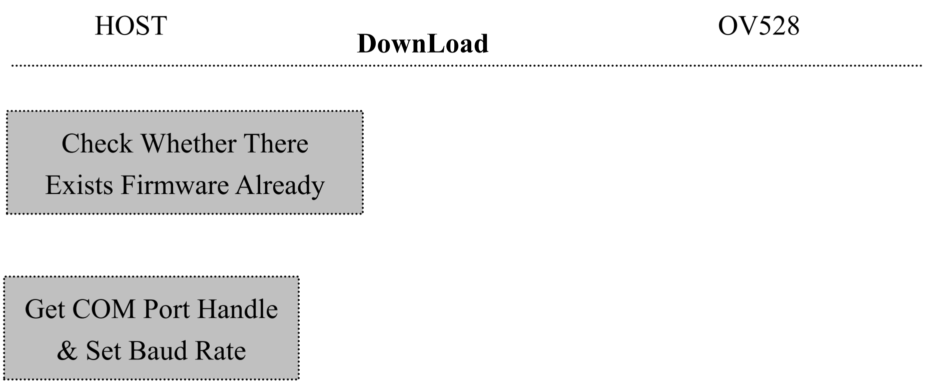
SET REGISTER	Host can control camera's internal register value by using this command. **Register Type**: OV528 Register: 0x00 Sensor Register: 0x01
GET PICTURE	Host can get picture from OV528 by using this command. **Picture Type**: Snap Shot Picture: 0x01 Preview Picture: 0x02 Flash Memory File: 0x04 JPEG Preview Picture: 0x05 Decode Picture: 0x06 Flash Memory Free Space: 0x08 **File ID Low Byte and File ID High Byte**: If it is getting pictures from Flash memory, these two parameters determine which one is the object. The ID number starts from 0x01. If the ID number is 0x00, firmware will send Flash File Table data back.
SNAP SHOT	Host sends this command to inform OV528 to keep a JPEG image in its memory. **Snap Shot Type**: Compressed Picture: 0x00 Uncompressed Picture: 0x01 (unused) **Picture skip count**: Host can define how many frame pictures to skip before compression. "0" means keep current picture. "1" means compress next frame. Est.
SAVE DATA	By using this command, host can save data to OV528 or Flash Memory, change snapshot resolution and delete Flash Memory Files. **Destination**: Serial bus to SRAM (buffer in OV528): 0x01 Serial bus to OV528 FIFO RAM: 0x02 (unused) Serial bus to EEPROM: 0x03 Serial bus to Flash Memory: 0x04 SRAM to EEPROM: 0x05 SRAM to Flash Memory: 0x06 Change JPEG Resolution for Snapshot Button: 0x07 Delete Flash Memory Last File: 0xF0 Format Flash Memory: 0xFF **Length Byte 0 / JPEG Resolution**: If Destination is 'Change JPEG Resolution for Snapshot Button', this byte is JPEG Resolution. If Destination is Flash Memory, this byte is 0x00. Otherwise, it represents the low byte of downloading program's length. **Length Byte 1**: If saving file into Flash Memory, it is 0x00. Otherwise, it represents the middle byte of downloading program's length.

	File Property / Length Byte 2: If saving file into Flash Memory, it represents the File Property (See Table 4-6). Otherwise, this is the highest byte of downloading program's length.
SET BAUDRATE	**Fist Divider and Second Divider**: Baud rate = 14.7456MHz / 2 x (Second Divider + 1) / 2 x (First Divider + 1)
RESET	Host can reset the system by using this command. **Reset Type**: "0x00" means reset the whole system. System will reboot and reset all registers and state machine. "0x01" means camera only reset state machine.
POWER OFF	System will go into sleep-mode when receives this command. To wake up the system, host need to send SYNC command several times till receives ACK command from firmware.
DATA	Host downloads or system sends data back by using this command. The unit for the length is byte and it doesn't include the length of command itself. **Data Type**: Register Data: 0x00 SnapShot Picture: 0x01 Preview Picture: 0x02 Program: 0x03 Flash Memory File: 0x04 JPEG Preview Picture: 0x05 Decode Picture: 0x06 Check Sum Data: 0x07 (unused) Flash Memory Free Space Data: 0x08 Flash Memory File Size: 0x09 **Length byte 1 and Length byte 0**: If the data is about Register, Snapshot, Preview picture, JPEG or Flash, these two bytes represent the length of those data. Otherwise, these two bytes are 0x0000. **Package Status / Length Byte 2**: If the data is Flash File, this byte represents the package status (last or not last). Otherwise, this byte is the highest length byte.

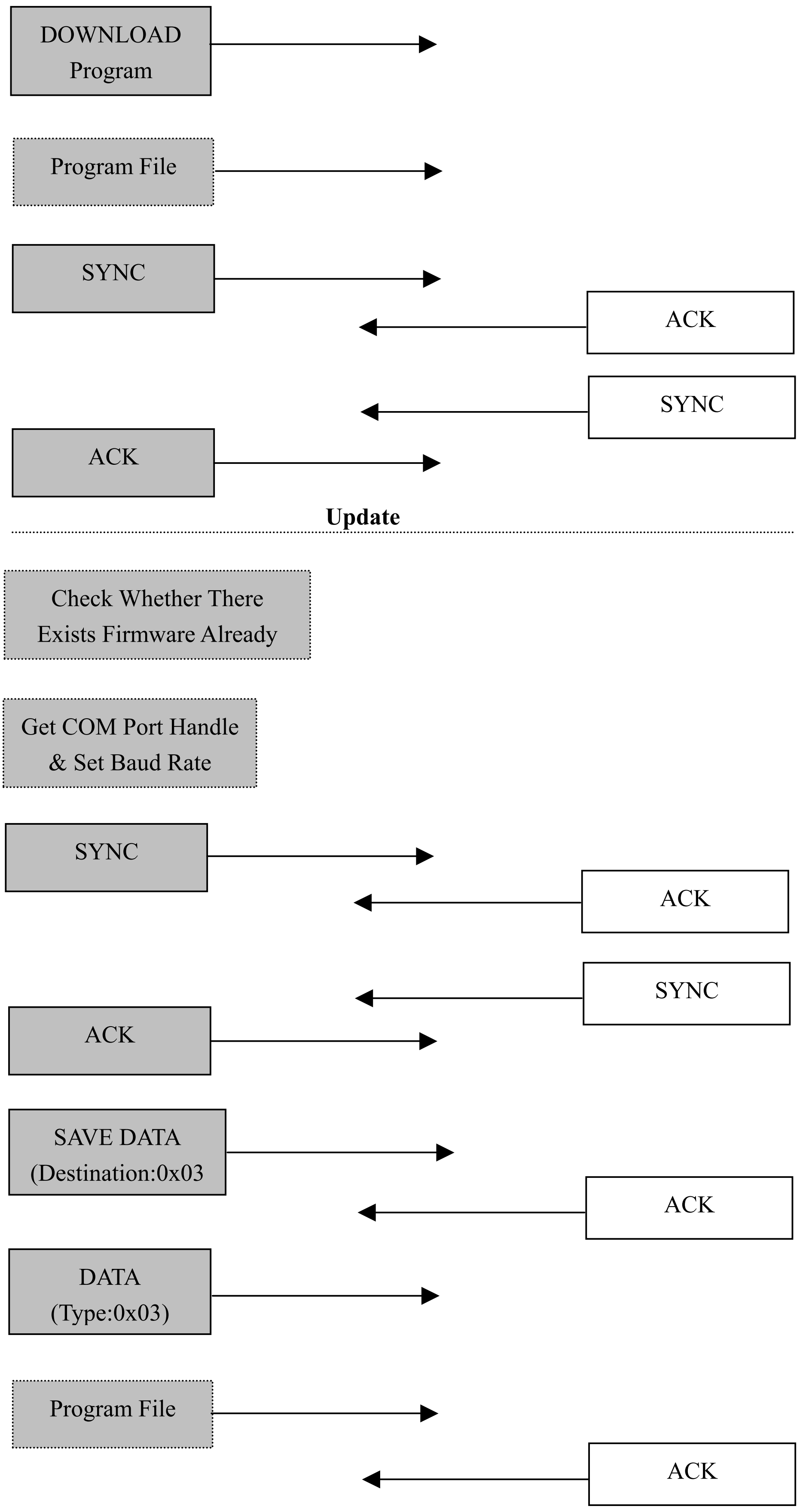
GET REGISTER	Host can read system's internal register value by using this command. System will send back the register value by using DATA command. **Register Type:** OV528 Register: 0x00 Sensor Register: 0x01
DOWNLOAD PROGRAM	Host can download program to OV528 by using this command. After host downloads program into system's program RAM, host needs to send SYNC command and wait ACK command from firmware. **Destination**: Program RAM: 0x05 **Length**: These three bytes represent the last two bytes of the download program's length.
SYNC	Either host or OV528 can send this command. ACK command is sent back after receiving this command.
ACK	This command is used to indicate the success of the last operation. After receiving any valid command, ACK command is sent out except downloading program or getting preview data. **Command ID**: Indicate the receiving command ID. **ACK Count**: No use.
NAK	This command is used to indicate the fail of the last communication or any unsupported feature. **NAK Count**: No use. **Error Code**: Error codes defined in firmware.

4.2.2 Download or Update Firmware

Table 4-10 Download or Update Functions and protocol

Functions	**BOOL DownloadProgramToOV528(HWND hWnd, LPBYTE lpBuffer, DWORD dwBufferSize, HWND hProcessBar)** **Function description:** If it's the first time for the system to be used and no firmware in it, this function will fulfill the order of downloading file. This function is used to write firmware file into OV528 and check its validity. If the file is valid, the function will return TURE and program should invoke function SaveToEEPROM to re-write the file into EEPROM and store the file there. **Parameters:** **hWnd**: handle of the main window **lpBuffer**: buffer for firmware file (Rsrunner.bin) **dwBufferSize**: size of firmware file **hProgressBar**: handle of the progressing bar which indicates the progress of the downloading or updating operation
	BOOL SaveToEEPROM(HWND hWnd, LPBYTE lpBuffer, DWORD dwBufferSize, HWND hProgressBar) **Function description:** This function will update the old firmware with a new version. In fact, it writes a new firmware into EEPROM. We recommend user to use this function instead of the first one as long as there is usable firmware in the system. **Parameters:** All the parameters are the same with that in function 'DownloadProgramToOV528'.
Protocol	 HOST DownLoad OV528 Check Whether There Exists Firmware Already Get COM Port Handle & Set Baud Rate

Protocol


DOWNLOAD Program
Program File
SYNC
ACK
SYNC
ACK
Update
Check Whether There Exists Firmware Already
Get COM Port Handle & Set Baud Rate
SYNC
ACK
SYNC
ACK
SAVE DATA (Destination:0x03
ACK
DATA (Type:0x03)
Program File
ACK

4.2.3 System's Initialization, Video Start, Stop and Reset (Exit)

Table 4-11 System's Initialization

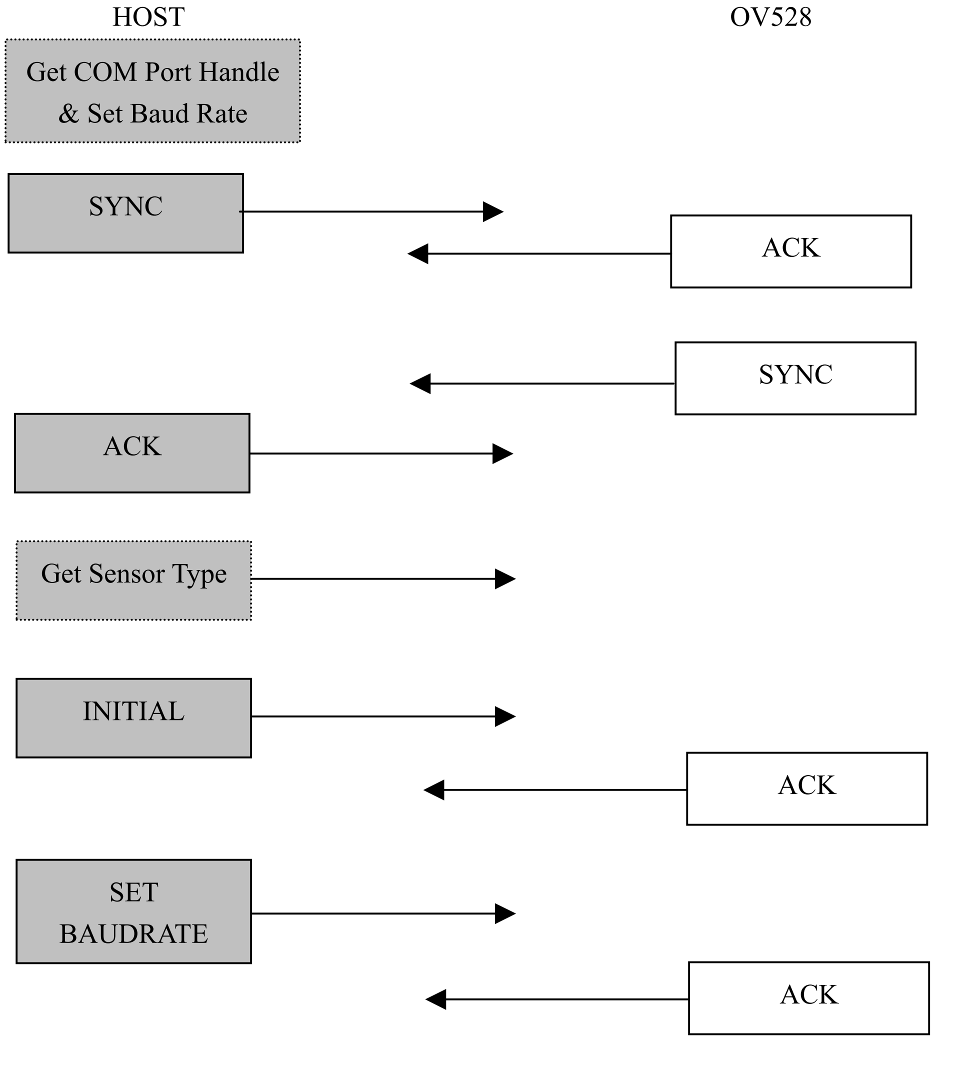
Function	**BOOL InitComPort (HINSTANCE hInstance, HWND hWnd, DWORD * pOVTSensorType)** **Function description**: This function will get the handle of an available COM port and set the correct baud rate. **Parameters**: **hInstance**: handle of the application program's instance **hWnd**: handle of main window **pOVTSensorType**: sensor type, see Table 4-1
Protocol	 HOST OV528 Get COM Port Handle & Set Baud Rate SYNC ACK SYNC ACK Get Sensor Type INITIAL ACK SET BAUDRATE ACK

Table 4-12 Video Start

Function	**BOOL VideoStart(SELECT_PARAM Select_Param, HWND GetImageProcess, int nLightFreq);** **Function Description:** Invoking this function will inform the driver to apply a lot of source and begin to get image data from the sensor. **Parameters**: **Select_Param**: Preview image information, see Table 4-5. **GetImageProcess**: Handle of the process bar which indicates the operation of reading a frame of image. Because the rate of image data's reading through RS-232 port is a little slow in a sense, the processing bar will inform the customer about the system's current state. **FreqLight**: It is the electronic frequency of the man-made light source. The value must be 50 or 60, otherwise, it will return FALSE. If the system works under the natural light condition, such as outdoors, this parameter is an invalid one. It must be uniform with the electronic frequency to guarantee the exposure time of any frame to be identical. Otherwise, many black and white strips will appear on the image.

Protocol

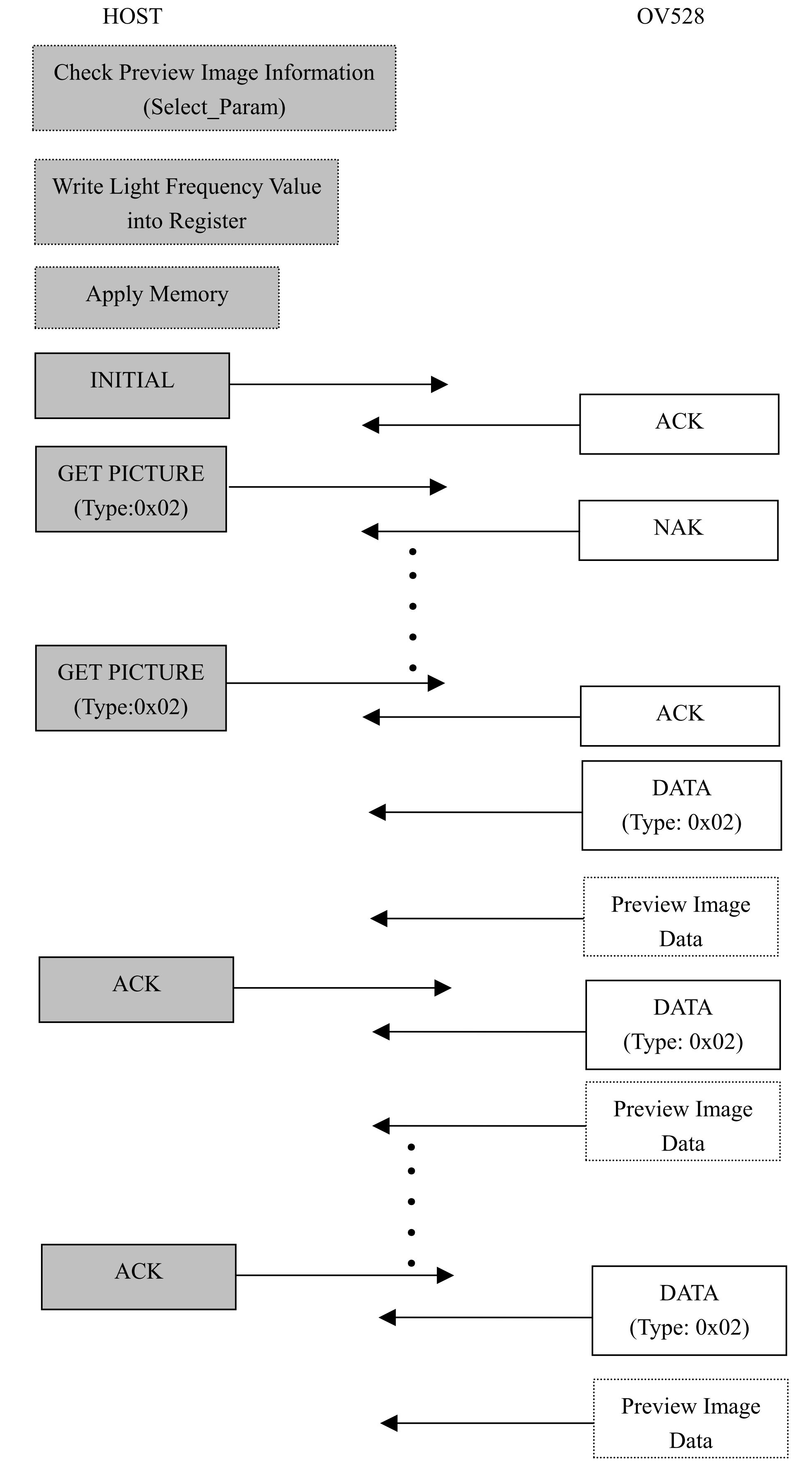

HOST
OV528
Check Preview Image Information (Select_Param)
Write Light Frequency Value into Register
Apply Memory
INITIAL
ACK
GET PICTURE (Type:0x02)
NAK
GET PICTURE (Type:0x02)
ACK
DATA (Type: 0x02)
Preview Image Data
ACK
DATA (Type: 0x02)
Preview Image Data
ACK
DATA (Type: 0x02)
Preview Image Data

Table 4-13 Stop

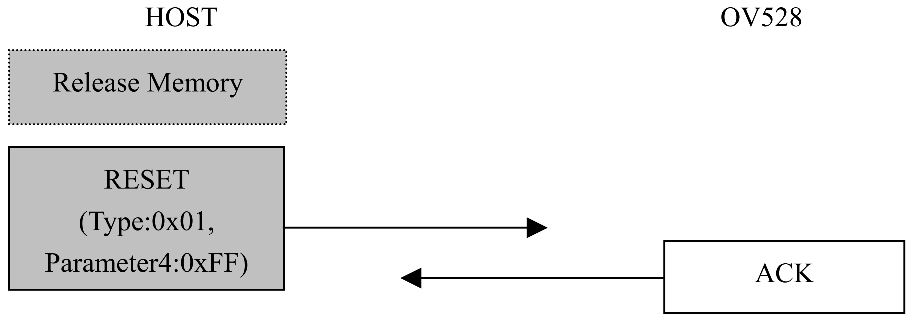
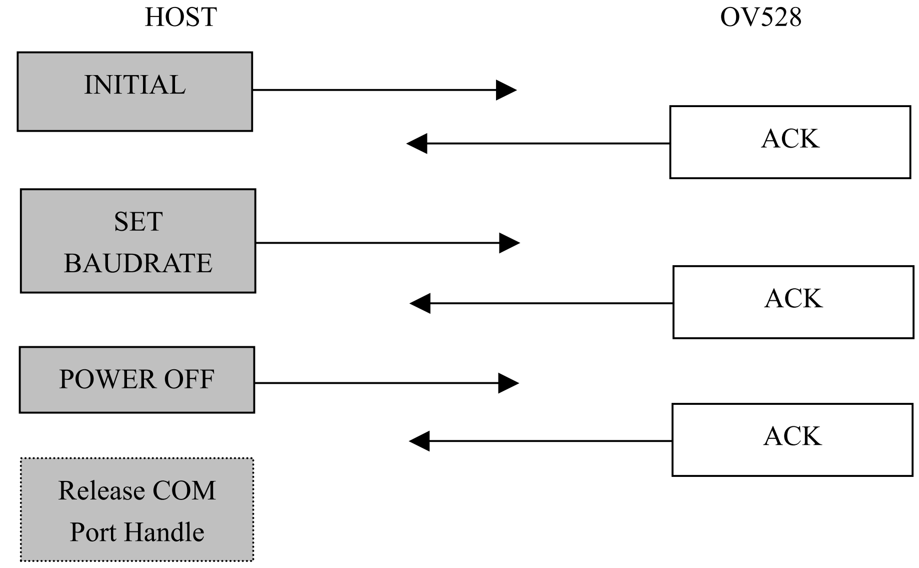
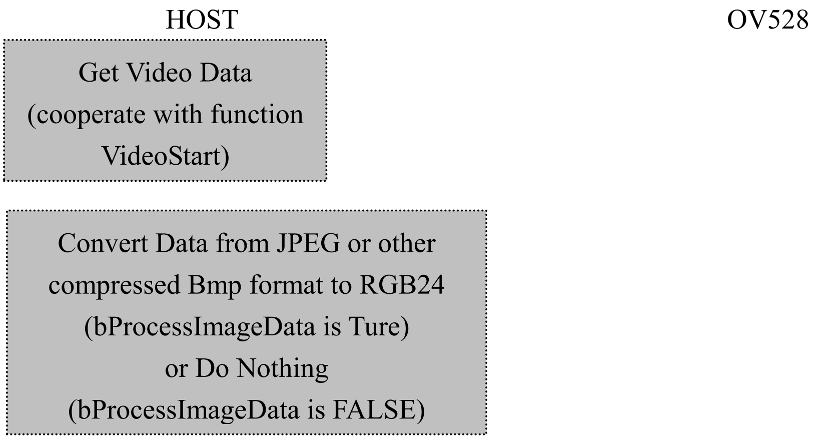
Functions	**BOOL VideoStop(void)** **Function Description:** When user needs to change the previewing image format or take snapshot still image, he should stop the previewing video. If the application exits, user should also call this function before invoking function ResetDevice().
Protocol	 HOST OV528 Release Memory RESET (Type:0x01, Parameter4:0xFF) ACK

Table 4-14 Reset (Exit)

Functions	**BOOL ResetDevice(void)** **Function Description:** Once the application program exit, it should reset the device and set the camera to power off status to save energy. The driver should also free the COM port handle which have been applied to program. (Before invoking this function, user needs to call VideoStop() at first in application program.)
Protocol	 HOST OV528 INITIAL ACK SET BAUDRATE ACK POWER OFF ACK Release COM Port Handle

4.2.4 Preview of Live Video

Table 4-15 Preview of Live Video

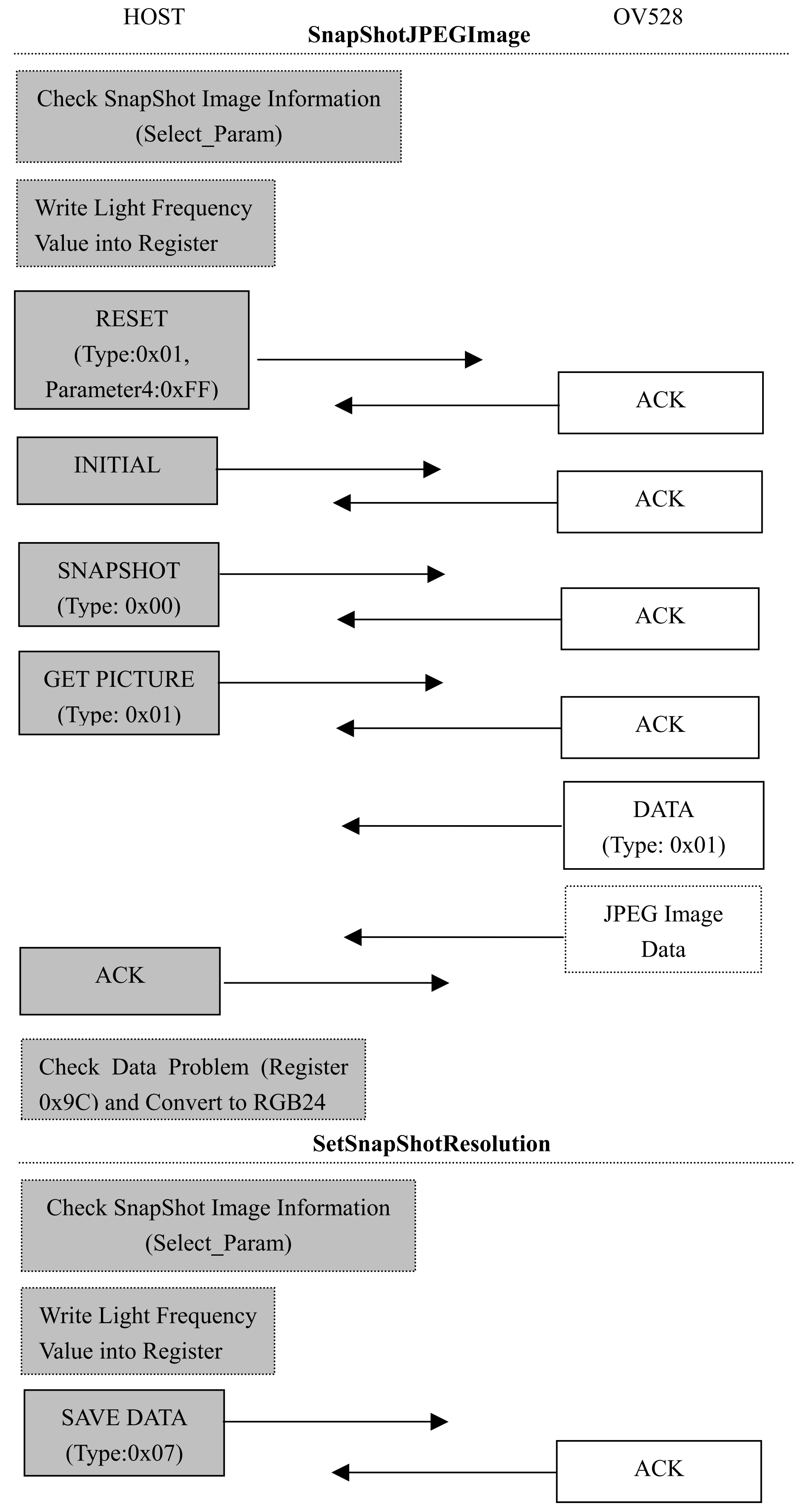
Functions	**BOOL ViewLiveVideo(BYTE *pVideoBuffer, DWORD dwBufferSize, BOOL bProcessImageData, DWORD *pRetImageSize, DWORD *pWidth, DWORD *pHeight)** **Function Description:** After calling VideoStart() function, the application program can get the image data from the driver one frame by one frame. If the driver is not ready for one frame image data, it will return FALSE, otherwise, it will return TRUE. And the image data will be stored in the buffer that applied by the application program. If the driver gets a whole frame of image data, driver will decode the data and convert them to RGB24 format according to the requirement from the application program, then return them to the latter to show on screen (Preview Window, see Fig.3-3) **Parameters**: **pVideoBuffer**: buffer allocated by application program beforehand for live video data. **dwBufferSize**: size of the video buffer. **bProcessImageData**: if it is TURE, driver will convert the image data (which are got in function VideoStart).to RGB24; otherwise, it will send out the data according to color type. **pRetImageSize**: it is the size of the returned image. **pWidth and pHeigh**: width and height of the frame.
Protocol	**Note:** no protocol (command) in this function, all the tasks are finished inside OV528Drv.dll.  HOST OV528 Get Video Data (cooperate with function VideoStart) Convert Data from JPEG or other compressed Bmp format to RGB24 (bProcessImageData is Ture) or Do Nothing (bProcessImageData is FALSE)

4.2.5 SnapShot of JPEG Still Image

Table 4-16 SnapShot of JPEG Still Image

Functions	**BOOL SnapShotJPEGImage(BYTE *pVideoBuffer, DWORD dwBufferSize, SELECT_PARAM Still_Select_Param, DWORD *pDataSize, DWORD *pWidth, DWORD *pHeight, int nLightFreq)** **Function Description:** This function is to get the JPEG image data from the hardware and save to memory buffer allocated by application program, then return the image data to the latter to show on screen (SnapShot Picture View Window, see Fig.3-3). **Parameters**: **pVideoBuffer**: buffer allocated by application program beforehand for still JPEG image data. **dwBufferSize**: size of the video buffer. **Select_Param**: Snapshot image information, see Table 4-5. **pDataSize**: JPEG image's real size. **pWidth and pHeigh**: width and height of the frame. **nLightFreq**: light frequency.
	BOOL SetSnapShotResolution(SELECT_PARAM SelectParam, int nLightFreq, BOOL bCompressed, BYTE byMediumType) **Function Description:** This function is to change the size of the JPEG image (see Table 4-3 for the size type). Here, the JPEG image is got by pressing button '**SNAP**' mounted on evaluation or demo board and saved directly into flash memory. It differs from that got by driver and send to application program to display (just as function SnapShotJPEGImage does). **Parameters**: **Select_Param**, **nLightFreq**: the same with that in function SnapShotJPEGImage. **bCompressed**: Is the snap shot picture in compressed format or not. In fact, uncompressed image's snapshot isn't supported by firmware now. **byMediumType**: the type of the storage medium, MT_PFLASH is the only choice in current system.

Protocol


HOST
SnapShotJPEGImage
OV528
Check SnapShot Image Information (Select_Param)
Write Light Frequency Value into Register
RESET (Type:0x01, Parameter4:0xFF)
ACK
INITIAL
ACK
SNAPSHOT (Type: 0x00)
ACK
GET PICTURE (Type: 0x01)
ACK
DATA (Type: 0x01)
JPEG Image Data
ACK
Check Data Problem (Register 0x9C) and Convert to RGB24
SetSnapShotResolution
Check SnapShot Image Information (Select_Param)
Write Light Frequency Value into Register
SAVE DATA (Type:0x07)
ACK

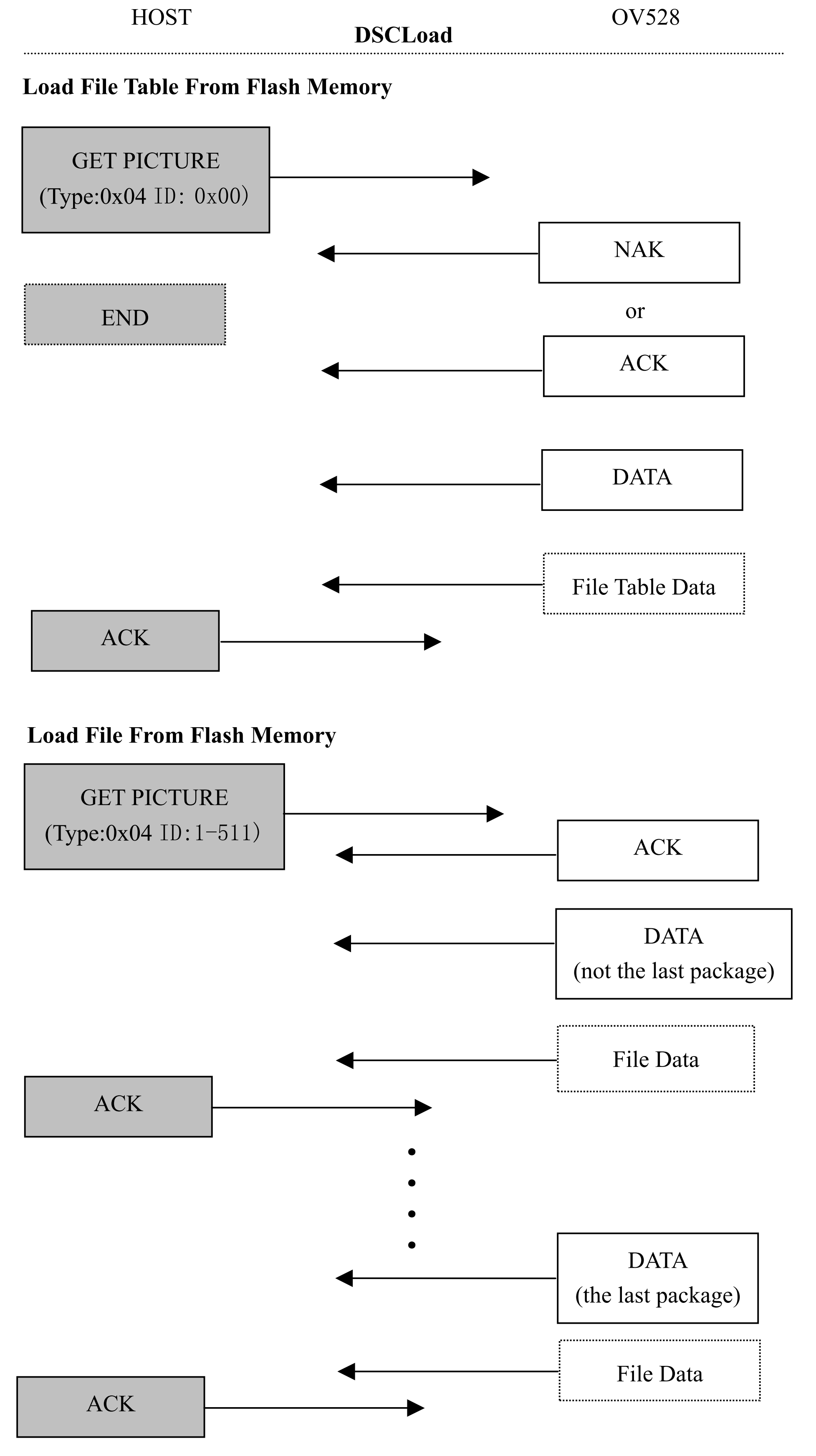
4.2.6 DSC or Flash Memory Funtions

Table 4-17 DSC or Flash Memory Functions

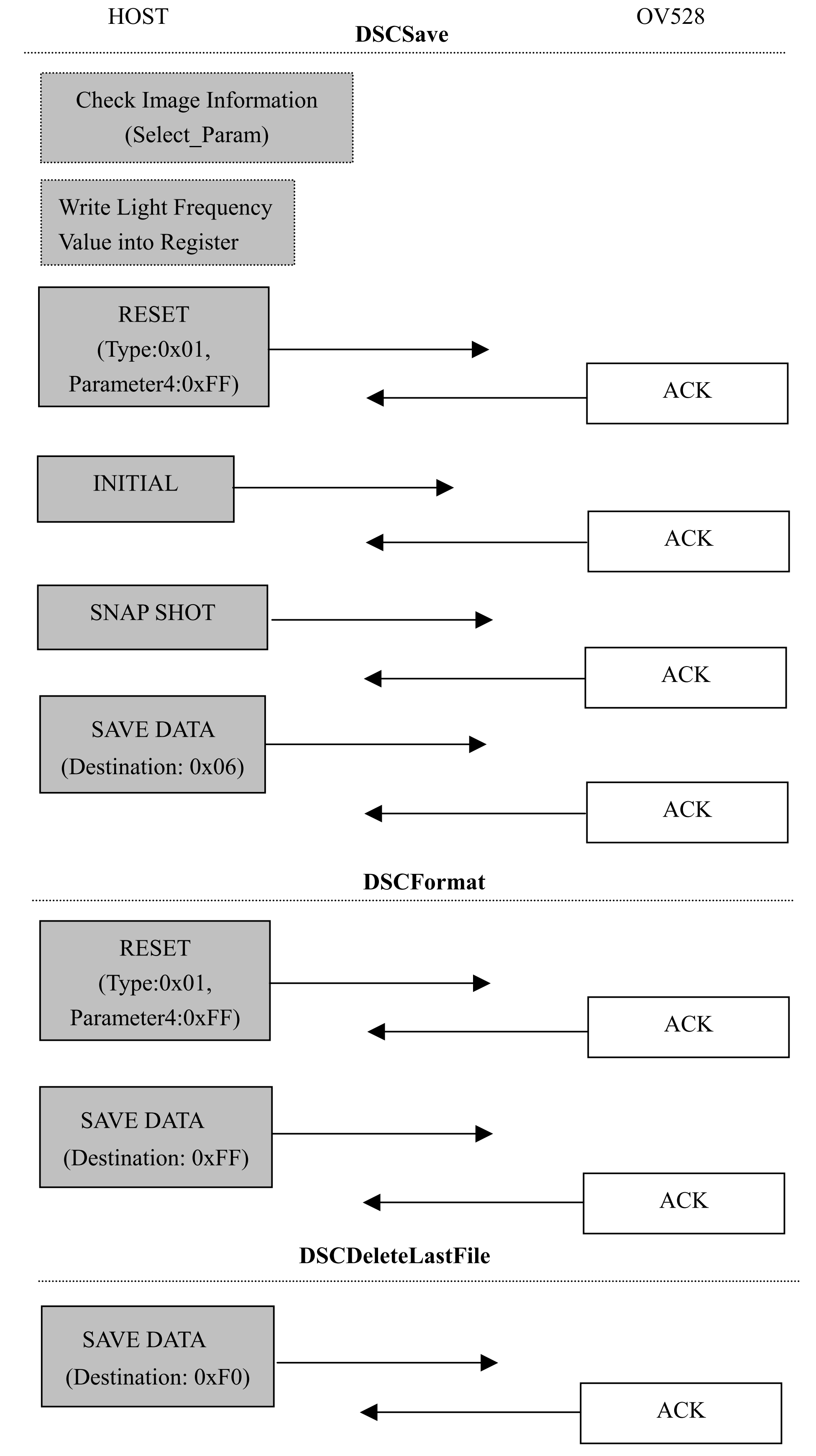
Functions	**BOOL DSCLoad(const PFlash_FileTableItem *pFileTableItem, BYTE *pFileBuffer, DWORD dwFileBufferSize, DWORD *pNumberOfBytesWritten, BYTE byMediumType)** **Function Description:** This functions will load either the file table or the specified file from the flash memory then save into a disc file on the hard disc. The file's extension name can be one of the two types : jpg and bmp or no extension name exists if the file is neither a JPEG nor a Bitmap file. **Parameters**: **PFlash_FileTableItem**: flash memory file information (see Table 4-6). If this parameter is NULL, driver will think that application program want to load the file table from the flash memory. Otherwise, driver consider it to be the information of the specified file in flash memory. **pFileBuffer**: buffer contain the file content. **dwFileBufferSize**: size of the file buffer. **pNumberOfBytesWritten**: total number of the bytes that read into the file buffer. **byMediumType**: the type of the storage medium, MT_PFLASH is the only choice in current system
	BOOL DSCSave(SELECT_PARAM SelectParam, int nLightFreq, WORD wSkipFrame, BOOL bCompressed, BYTE byMediumType) **Function Description:** It will save the current image in the RAM into the flash memory. The total number of the image is from 1 to 511. Single file's maximum size is less than 16 M bytes (limited by flash memory's size, such as Samsung and Toshiba, this number is only 8 M bytes in our current system). **Parameters**: **Select_Param**: Snapshot image information, see Table 4-5. **nLightFreq**: light frequency, 50Hz or 60Hz. **wSkipFrame**: save which image, current or not. If it is 0, current image will be saved. Otherwise, firmware will skip wSkipFrame frames of image and save the next one. **bCompressed**: Is the image saved in compressed format or not. In fact, uncompressed image's saving isn't supported by firmware now. **byMediumType**: the type of the storage medium, MT_PFLASH is the only choice in current system.

Functions	**BOOL DSCFormat(BYTE byMediumType)** **Function Description:** If this function is called, all the images in flash memory will be deleted without recoverability. Thus, application program should remind user of the aftermath before invoking this function. **Parameters**: **byMediumType**: the type of the storage medium, MT_PFLASH is the only choice in current system
	BOOL DSCDeleteLastFile(BYTE byMedeiumType) **Function Description:** This function will inform the firmware to delete the last file from the flash memory. **Parameters**: **byMediumType**: the type of the storage medium, MT_PFLASH is the only choice in current system
	BOOL DSCTransmitFile(const BYTE *pBuffer, DWORD dwBufferSize, BYTE bySaveDataType, BYTE byMediumType) **Function Description:** Unlike DSCSave, this function will save a disc image file rather than a currently displayed one into the flash memory. Thus, an object image file must be specified by the user at first. File of any type can be saved as long as the file's size is within the bound of limitation. **Parameters**: **pBuffer**: buffer that contains the file to be saved into flash memory. **dwBufferSize**: size of the file buffer. **bySaveDataType**: this parameter is used by application program to classify the files. Now, we roughly divided the files in flash memory into 3 types, JPEG, BMP and Others. According to this parameter, application program will apply different operation methods on them and save into different files. **byMediumType**: the type of the storage medium, MT_PFLASH is the only choice in current system
	BOOL DSCGetFreeSpace(DWORD *pFreeSpaceSize, BYTE byMediumType) **Function Description:** To get the remained space in flash memory. **Parameters**: **pFreeSpaceSize**: quantity (in bytes) of the free space.

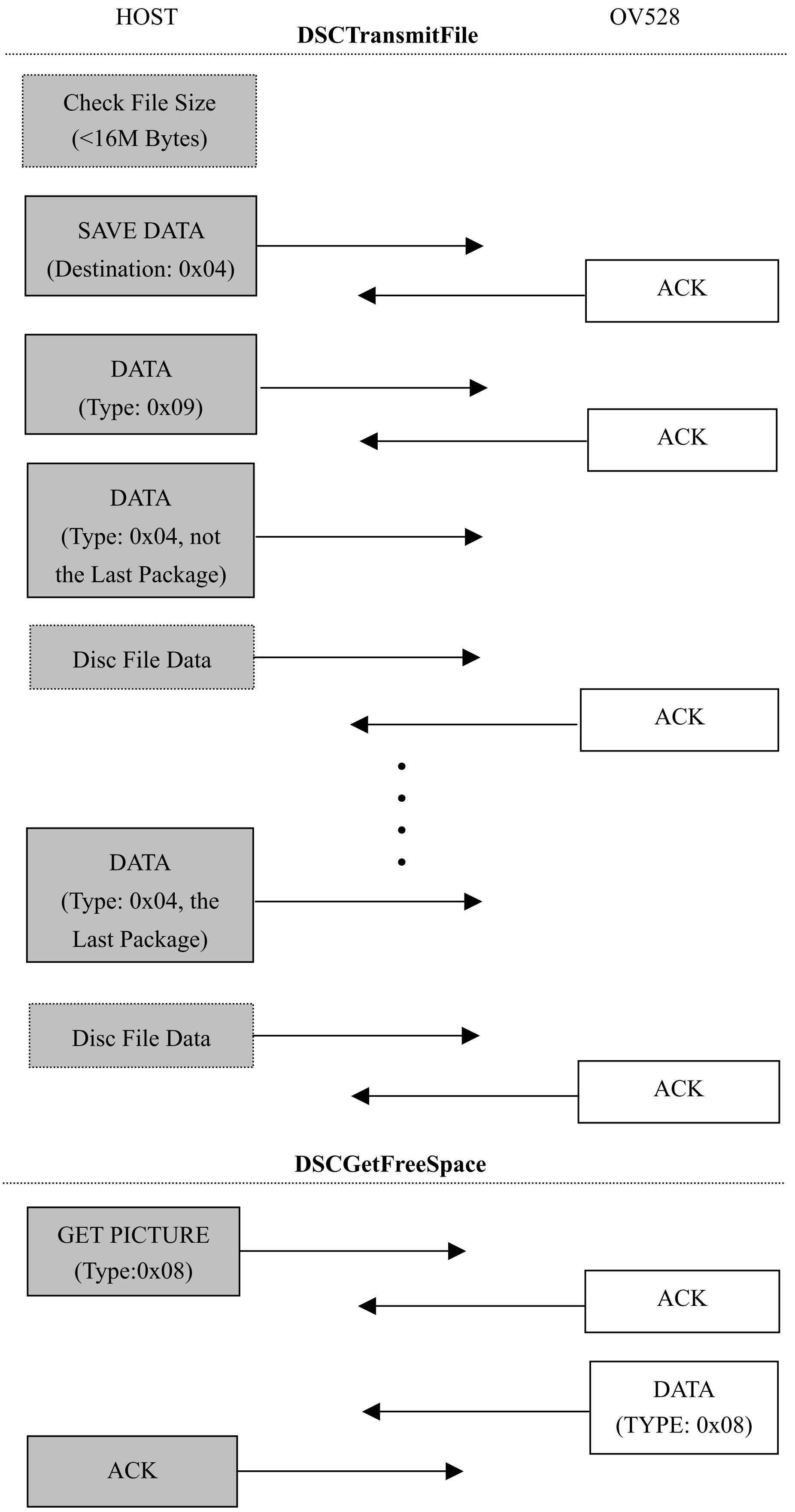
Protocol


HOST
DSCLoad
OV528
Load File Table From Flash Memory
GET PICTURE
(Type:0x04 ID: 0x00)
NAK
END
or
ACK
DATA
File Table Data
ACK
Load File From Flash Memory
GET PICTURE
(Type:0x04 ID:1-511)
ACK
DATA
(not the last package)
File Data
ACK
DATA
(the last package)
File Data
ACK

Protocol


HOST
OV528
DSCSave
Check Image Information (Select_Param)
Write Light Frequency Value into Register
RESET (Type:0x01, Parameter4:0xFF)
ACK
INITIAL
ACK
SNAP SHOT
ACK
SAVE DATA (Destination: 0x06)
ACK
DSCFormat
RESET (Type:0x01, Parameter4:0xFF)
ACK
SAVE DATA (Destination: 0xFF)
ACK
DSCDeleteLastFile
SAVE DATA (Destination: 0xF0)
ACK

Protocol


HOST
OV528
DSCTransmitFile
Check File Size
(<16M Bytes)
SAVE DATA
(Destination: 0x04)
ACK
DATA
(Type: 0x09)
ACK
DATA
(Type: 0x04, not
the Last Package)
Disc File Data
ACK
DATA
(Type: 0x04, the
Last Package)
Disc File Data
ACK
DSCGetFreeSpace
GET PICTURE
(Type:0x08)
ACK
DATA
(TYPE: 0x08)
ACK

4.2.7 JPEG Image Decoding

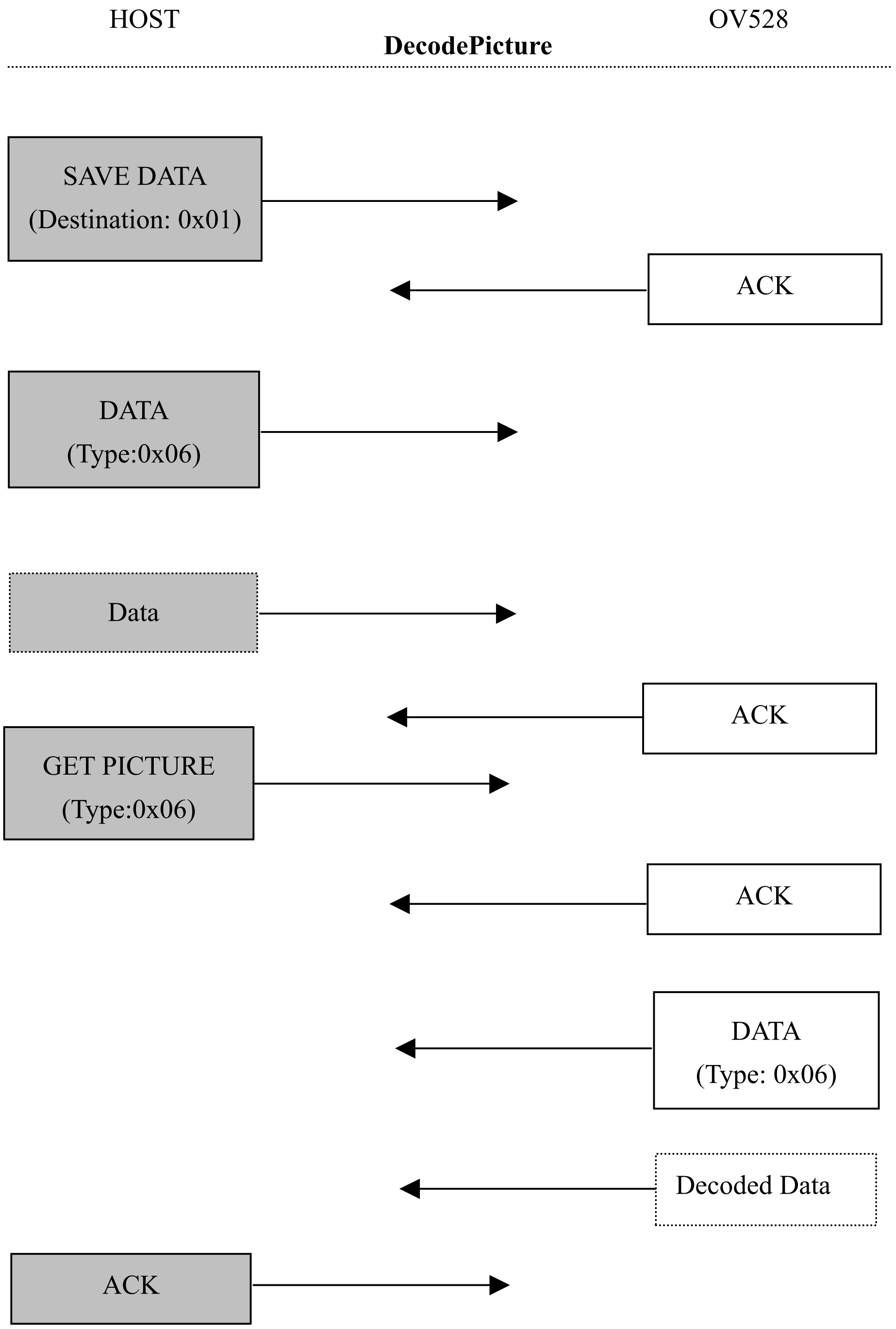
Table 4-18 JPEG Image Decoding

Functions	**BOOL JPGDecoder(unsigned char *ucSourceBuffer, unsigned char *ucDestBuffer)** **Function Description:** This function will decode a JPEG image file into a 24-bit Bitmap file. JPEG file's size, width and height will be read out inside this function. **Parameters**: **ucSourceBuffer**: source buffer for JPEG image data. **ucDestBuffer**: destination buffer for Bitmap image data.
	BOOL DecodePicture(SELECT_PARAM DecodeParam, const BYTE *pPictureBuffer, DWORD dwBufferSize, BYTE *pBMPBuffer); **Function Description:** This function can also decode a JPEG file into a Bitmap one. Unlike function JPGDecoder, this one decodes the JPEG file by means of hardware (OV528) but not by software. And, there are some limitations on this function's usage. The limitations are: 1. Source file's size must be one of the following 4 kinds: 320*240, 640*480, 176*144, 352*288. 2. JPEG files must be those being created from OV528 system especially those generated by operation 'SnapShot'. **Parameters**: **Select_Param**: source image file's information, see Table 4-5. **pPictureBuffer**: source image file's buffer. **dwBufferSize**: size of source image file's buffer **pBMPBuffer**: object image file's buffer

<table>
<tr>
<td><b>Protocol</b></td>
<td>

JPGDecoder

This is defined and implemented in JPEGDecoder.dll, no command has been come down to.



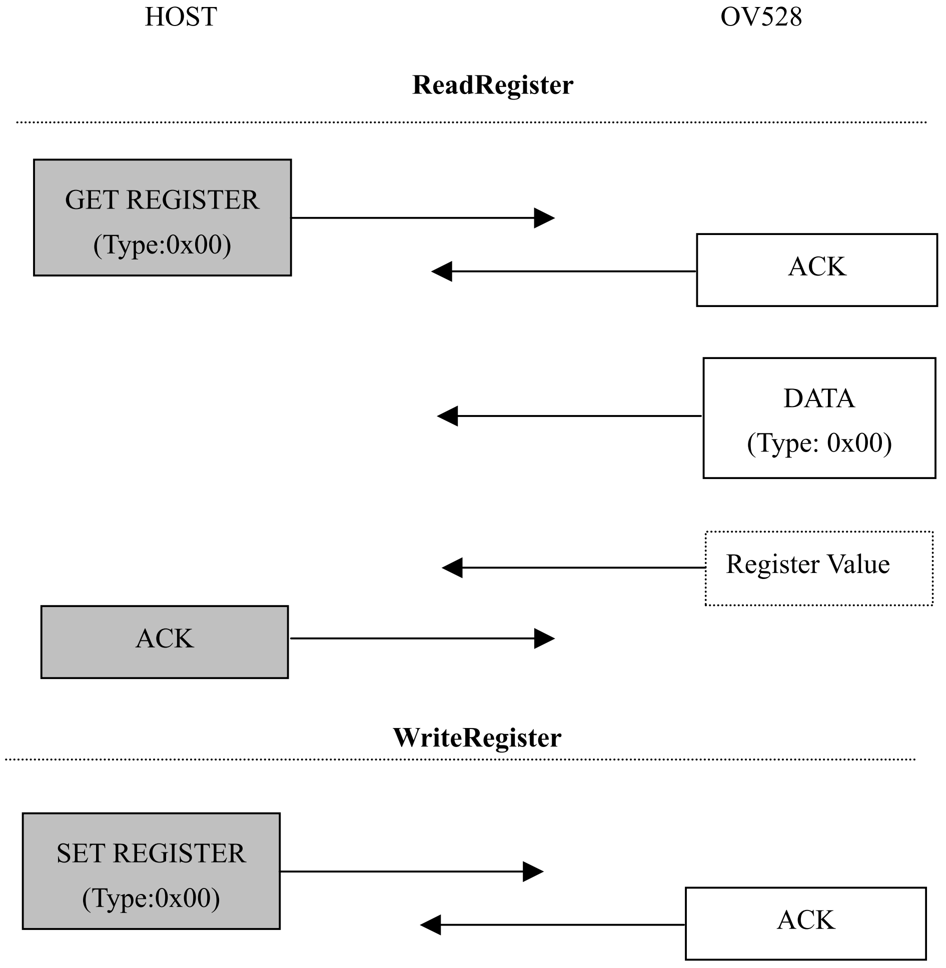
DecodePicture

</td>
</tr>
</table>

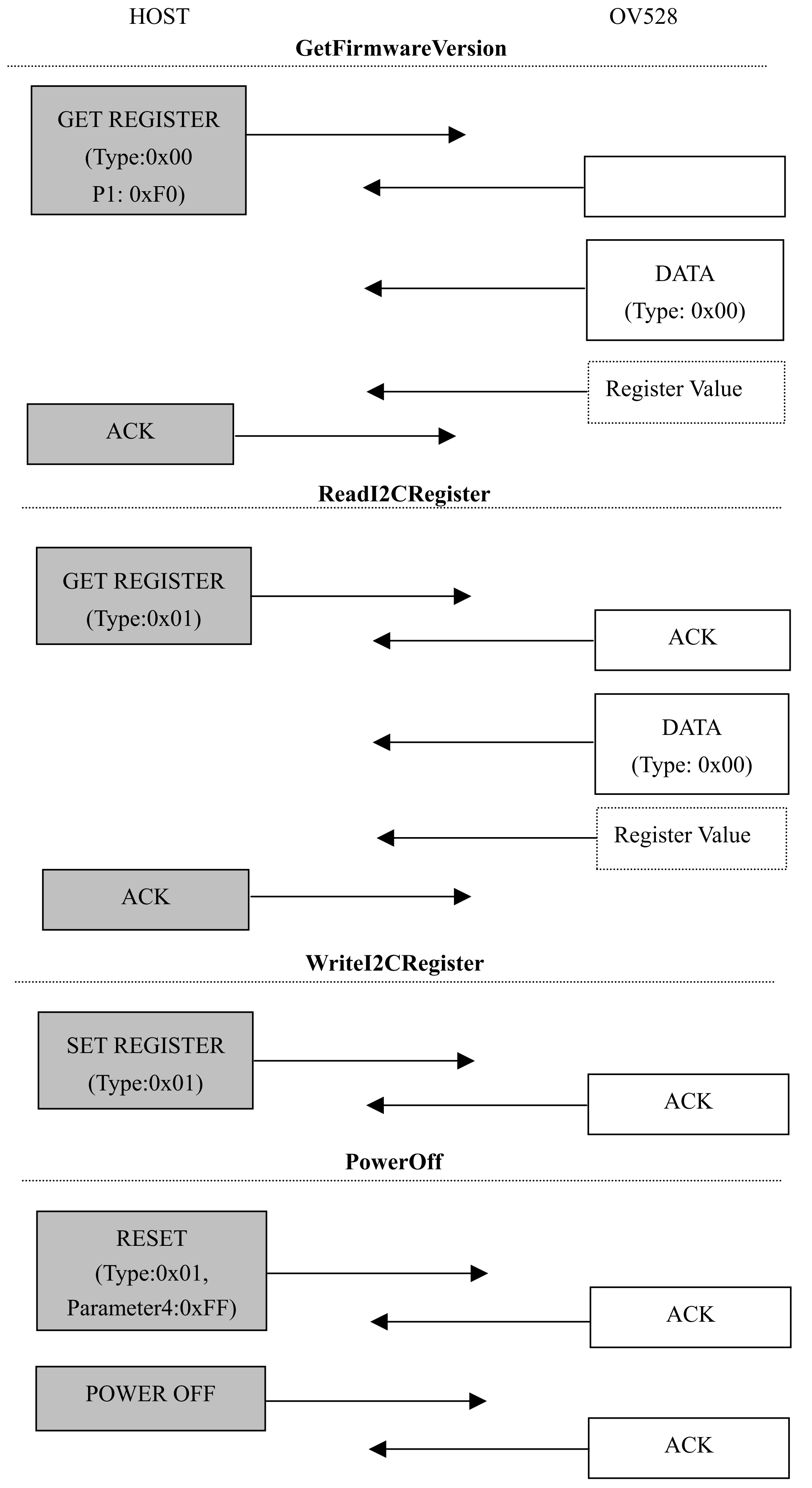
4.2.8 Other Functions

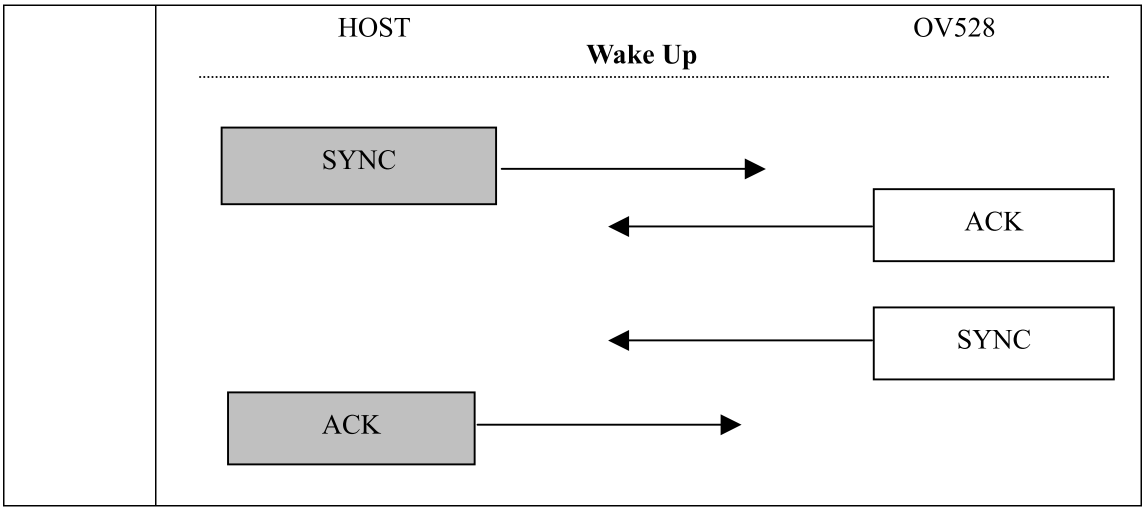
Table 4-19 Other Functions

Functions	**BOOL ReadRegister(int nRegisterAdd, int *pRegisterValue)** **Function Description:** This function is designed to read the value of OV528 registers. **Parameters**: **nRegisterAdd**: address of the object register **pRegisterValue**: address where the value read out from register is saved.
	BOOL WriteRegister(int nRegisterAdd, int nRegisterValue) **Function Description:** This function is designed to write a number into OV528 registers. **Parameters**: **nRegisterAdd**: address of the object register **pRegisterValue**: value that will be written into the object register.
	BOOL ReadI2CRegister(int nRegisterAdd, int *pRegisterValue) **Function Description:** This function is designed to read the value of sensor registers. **Parameters**: **nRegisterAdd**: address of the object register **pRegisterValue**: address where the value read out from register is saved.
	BOOL WriteI2CRegister(int nRegisterAdd, int nRegisterValue) **Function Description:** This function is designed to write a number into sensor registers. **Parameters**: **nRegisterAdd**: address of the object register **pRegisterValue**: value that will be written into the object register.
	BOOL GetFirmwareVersion(BYTE *pMajorVersion, BYTE *pMinorVersion) **Function Description:** This function will read the firmware version from a given register. **Parameters**: **pMajorVersion** and **pMinorVersion**: version value.

Functions	**DWORD OVGetLastError(void)** **Function Description:** This function will return the error codes defined in firmware to the application program or the driver.
	BOOL PowerOff(void); **Function Description:** Set the system into sleep mode in order to save energy.
	BOOL WakeUp(void); **Function Description:** Wake up the system who is sleeping to work.
Protocol	 HOST OV528 ReadRegister GET REGISTER (Type:0x00) ACK DATA (Type: 0x00) Register Value ACK WriteRegister SET REGISTER (Type:0x00) ACK **OVGetLastError** As for functions **OVGetLastError**, no commands are used in them.

Protocol


HOST
OV528
GetFirmwareVersion
GET REGISTER (Type:0x00 P1: 0xF0)
DATA (Type: 0x00)
Register Value
ACK
ReadI2CRegister
GET REGISTER (Type:0x01)
ACK
DATA (Type: 0x00)
Register Value
ACK
WriteI2CRegister
SET REGISTER (Type:0x01)
ACK
PowerOff
RESET (Type:0x01, Parameter4:0xFF)
ACK
POWER OFF
ACK



4.2.9 Thread and Synchronization

In function VideoStart, a thread is created to read in image data from hardware. Because this thread has been regarded as a global variable, we give name ‘**g_hReadImageThreadHandle’** to the handle of this thread. Once the thread is executed, image data will be continually read in until an error engenders or the operation is stopped by user.

In windows, several synchronization objects are provided to permit the threads to synchronize their actions. Therefore, depending on them, we can inform the thread to stop reading the image data. Here, we use ‘Event’ which is created in function VideoStart to take on the work, and it is initialized to be ‘manual-reset’ and ‘unsignaled’.
In function VideoStop, the Event Object (the name of its handle is ‘**g_hStopReadEvent**’) will be manually reset to be ‘signaled’. So, the thread function (**ReadImageThread**) needs to check the state of g_hStopReadEvent at proper positions to make sure whether the reading operation should be continued or not. To be specific, if the Event Object is ‘signaled’, function should stop immediately. Otherwise, continue.

Beyond reading image data from hardware, OV528Drv.dll still needs to send the data to the application program. For the sake of not interrupting both of the operations, we allocate two image data buffers (**g_Img_Data_Buf**) to receive the image data alternately. Each buffer has its owe flag which has four states now and can be used to indicate the current state of the buffer. The four states are :
0 – no data, no operation; 1 – writing data into; 2 – reading data out; 3 – has data, no operation. However, only state 0 and state 3 have been utilized now.
Another synchronization object concerning with g_Img_Data_Buf is Semaphore, it indicates the status of the source (image buffers). ‘Signaled Semaphore Object’ means there is available source. With Semaphore Object, the image buffers are guaranteed not to be ‘illegally occupied’.

(In fact, the functions of Semaphore Object and image buffer’s flag are overlapped to some extend, user can develop their own synchronization mechanism.)

Index of Functions

Totally, there are 24 exported functions from OV528Drv.dll, all of them are described in this manual. For convenience, we give out the index of each function.

Table Function Index

Function Name	Page	Table
DownloadProgramToOV528	20-21	4-10
SaveToEEPROM	20-21	4-10
InitComPort	21-22	4-11
VideoStart	23-24	4-11
VideoStop	25	4-13
ResetDevice	25	4-14
ViewLiveVideo	26	4-15
SnapShotJPEGImage	27	4-16
DSCLoad	29，32	4-17
DSCSave	29，33	4-17
DSCFormat	30，32	4-17
DSCDeleteLastFile	30，32	4-17
DSCTransmitFile	30，33	4-17
DSCGetFreeSpace	30，33	4-17
JPGDecoder	34-35	4-18
DecodePicture	34-35	4-18
ReadRegister	36-37	4-19
WriteRegister	36, 38	4-19
ReadI2CRegister	36, 38	4-19
WriteI2CRegister	36, 38	4-19
GetFirmwareVersion	36, 38	4-19
OVGetLastError	37-38	4-19
PowerOff	37-38	4-19
WakeUp	37-39	4-19

References

[1]. Chip OV528 Datasheet

[2]. OV7620 Product Specifications –Rev.1.3

[3]. OV7635 Advanced Information Preliminary

[4]. OV6630 Advanced Information Preliminary

[5]. OV6640 Advanced Information Preliminary