VIMOS SOFTWARE FAMILY

Using VIMOS on ADSP camera

15 March 2005

Thank you for your interest in our Vision Inspection and Optical Measurement System (VIMOS). In this manual you will find out how to install VIMOS system on ADSP camera.

Before going on reading the manual, we kindly ask you to read the following

DISCLAIMER

This documentation is provided for reference purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this documentation, this documentation is provided “as is” without any warranty whatsoever and to the maximum extent permitted, atto-Systems Ltd. and Wolf Systeme AG disclaim all implied warranties, including without limitation the implied warranties of merchantability, non-infringement and fitness for a particular purpose, with respect to the same. Neither atto-Systems Ltd. nor Wolf Systeme AG shall be responsible for any damages, including without limitation, direct, indirect, consequential or incidental damages, arising out of the use of, or otherwise related to, this documentation or any other documentation. Notwithstanding anything to the contrary, nothing contained in this documentation or any other documentation is intended to, nor shall have the effect of, creating any warranties or representations from atto-Systems Ltd., Wolf Systeme AG or any of their suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of this software.
The described software is provided 'as is', without any warranty expressed or implied. No guaranty is given that the software is suitable for any given purpose.

COPYRIGHT

Atto-Systems Ltd.
Sofia, Bulgaria

Wolf Systeme AG
Karlsbad, Germany

TRADEMARKS

CONVENTIONS USED IN THIS MANUAL

***INFORMATION**. This sign marks section in the manual, which is for information only. You can decide to read or skip this section.*

***ATTENTION**. This sign marks section of the manual, which is particularly important for the general understanding of VIMOS. Please, make sure to read this section before proceeding with reading the manual.*

***TIPS & TRICKS**. This sign marks a Tips & Tricks section. Here you can find some practical advises on using the system or get a more detailed explanation of some features. Reading this section may help you in solving a particular problem or give you some ideas but is not vital for understanding VIMOS.*

***PREMISE**. This sign marks a section, which requires you to do something before proceeding with reading the manual. Usually this is a demo program, you have to run or something similar.*

File	*Menu item*
File > Open	*Sub-menu item or dialog control*
"1.1 About"	*Section reference. If the section is within the current manual no manual name is specified. When the section is within external manual the name of the respective manual is also included.*
`Ctrl+E`	*Hot-key combination. The first part of the combination specifies which system key to use. Possible values are: Ctrl, Alt, Shift. The second part specifies the normal key to be used in the combination. The plus sign means that you should press these keys simultaneously.*

CONTENTS

1. Introduction

VIMOS is a machine-vision system – it uses digital technology to perform visual tasks such as inspection, measurement, quality assurance and control of visual quality parameters. VIMOS runs on intelligent cameras manufactured by Vision Components, Germany. The cameras are based on DSP processors (CPUs) from Analog Devices (**ADSP cameras**) and Texas Instruments (**TI cameras**).

This manual provides detailed information on how to install and register VIMOS on ADSP camera with video-output (**VC38, VC61**). ADSP cameras have flash memory, which replaces the hard disk of the PC. The flash memory is a non-volatile memory, which means its data is preserved when camera power is switched off. The installation procedure loads files from PC to camera flash.

The manual also contains information about camera resources, flash files and various useful tips when working with VIMOS on ADSP camera.

***ATTENTION**. A description how to work with VIMOS kernel – the system module, which actually does the image-processing work, and which is a platform-independent module, can be found in the manual "Using the VIMOS Kernel".*

***INFORMATION**. The more powerful TI cameras currently replace the ADSP cameras. TI cameras are the recommended choice for new VIMOS projects, which require camera with video-output.*

2. VIMOS installation

The installation of VIMOS on ADSP camera includes two steps:

- Load VMALL.MSF file into flash memory (MSF is a special file format accepted by the cameras).
- Register VIMOS (enable execution of VIMOS kernel on camera).

ADSP cameras are usually equipped with 2 Mbytes flash (32 sectors x 64Kbytes). One or two sectors are reserved by the operating system of the camera. The remaining free space is used for user files.

Due to hardware constraints of ADSP cameras, VIMOS is composed of multiple executable modules, which call each other in a subroutine manner. All modules are contained in one merged file VMALL.MSF.

Read section “2.2. How to load files to camera” to learn how to load MSF files to camera.

2.1. VIMOS distribution file

A VIMOS system for ADSP camera is usually distributed in a ZIP file with name **V**xyys_mm.ZIP, where:

Xyy	VIMOS version x.yy
S	Optional VIMOS sub-version letter
Mm	Camera model (38, 61, …)

Examples:

V251_38.ZIP VIMOS version 2.51 for VC38

V251_61.ZIP VIMOS version 2.51 for VC61

Typical contents of a VIMOS distribution ZIP file for ADSP camera:

File	Description
VMALL.MSF	Merged MSF file with all system modules.
VMAUTO.MSF	VIMOS autoexec file, used to start VIMOS kernel on camera power on.
VCTIO.MSF	VCTIO module used by the Simulator to get JPEG pictures with specified shutter and quality (see the “Camera Files Functions” dialog).

The installation is fairly simple – just load VMALL.MSF to camera flash. Load the file **VMAUTO.MSF** when you are using VIMOS in stand-alone mode in a factory environment and VIMOS kernel should be started automatically on camera power on. Load **VCTIO.MSF** if you want to get camera pictures by Simulator with shutter and quality, which a different from the default ones.

2.2. How to load files to camera

Section “2.2.1. Using the Simulator” describes how to load MSF files by the Simulator. Currently the Simulator provides all means needed to install and work with VIMOS on camera, so you may skip next sections.

Sections “2.2.2. Using the VCT terminal” and “2.2.3. Using a general-purpose terminal” describe how to load files to camera via a terminal and present brief information on how to work with terminals. Read

these sections if you want to learn more about the camera operating system and how to load and run other camera programs.

***INFORMATION**. Simulator 2.60 and higher versions have built-in terminal via which you may load MSF files to camera and enter shell commands.*

2.2.1. Using the Simulator

Start Simulator, select **Camera > Camera Files Functions…** or press the toolbar button  to open the **Camera Files Functions** dialog. Setup for loading files by the following operations:

- Select a COM port (you should know which COM port of the PC is connected to camera, the default is COM1). Select baud rate (default = 9600).
- Press the **Find Camera** button to check PC-camera connection. Remember that camera baud-rate is set to 9600 after power-on. The simulator must find the camera without error messages and display a list of flash files (the list may be empty). In case of problems switch camera off, then switch on, wait several seconds for camera booting and try again. If the camera has AUTOEXEC file, which starts a program on power-on, press the "**Wait boot…**" button while switching the camera on.
- To ensure enough space for VIMOS kernel, erase all files in the camera flash by the **Clean Camera** button. The file list should become empty.
- Set baud rate = 115200 to enable fastest transfer speed. If you have problems with this speed, set lower baud rate.

Now you are ready to load MSF files to camera by the **Send MSF** button. Press the button and select the MSF file you want to load.

To load the all system modules:

- Load **VMALL.MSF** to flash.

To load the autoexec file:

- Load **VMAUTO.MSF** to flash.

Now you are ready to register VIMOS (see "3. VIMOS registration") and to start VIMOS kernel on ADSP camera (see "4. Starting VIMOS kernel on camera"). Refer to "Using the Simulator" manual for more information about the "Camera Files Functions" dialog and other Simulator functions.

2.2.2. Using the VCT terminal

VCT is a DOS terminal program developed by atto-Systems for work with Vision Components cameras and VIMOS. It is more convenient than general-purpose terminals especially when loading of MSF files to camera, writing or reading other files to/from camera and changing baud-rate. You may use VCT to work with VIMOS kernel on camera or to load and run other application programs. The terminal works with all camera models.

This section describes how to load MSF files using the VCT terminal. You should follow the installation procedures described above and load respective MSF files. We recommend erasing the standard flash by the **er** shell command before you start loading the MSF files.

Enter **Alt-P** to set correct serial port settings – COM port (usually COM1), no parity, 8-bit word, 1 stop bit, baud rate 9600. Check communication by pressing Enter - you should receive the prompt **'$**' each time.

Enter **Alt-L** command to open the **Load MSF file** window. This window is used to load MSF files to camera flash. Set the following window options:

- **F1** = BINARY (transfer mode, binary is faster).

- **F2** = 115200 (baud rate used during transfer). The terminal returns to current baud rate when the transfer is over.
- **F3** = None.

Finally enter name of MSF file and press Enter to start transfer. During the transfer you will see a progress bar on the bottom terminal line.

2.2.3. Using a general-purpose terminal

You can load VIMOS MSF files (or other MSF files) using a general-purpose terminal like the Windows program **HyperTerminal** or the DOS program **PROCOMM**. A file-loading procedure is described below:

- Set correct serial port settings – COM port (usually COM1), no parity, 8-bit word, 1 stop bit, baud rate 9600. Check communication by pressing Enter - you should receive the prompt '**$**' each time.
- If the camera has AUTOEXEC file, which starts a program on power-on, you may not see the shell prompt. Switch camera off, press **Esc** key and switch camera on. Hold the key down until you see the camera boot message.
- Erase flash by the **er** shell command. Remember that the Enter key should terminate each shell command.
- Upload MSF files by entering the "**lo**" command and follow the instructions for file uploading of the respective terminal (use ASCII protocol, no hardware handshake). When the uploading operation is finished, you must see multiple new files by the **dir** command.
- Register VIMOS (see next chapter). After VIMOS registration the module **ent_key** is no longer needed and you may delete it by the **del ent_key** command.

2.3. Installing custom logo on camera

On cameras with video output, VIMOS displays a system logo in the upper right corner of the overlay picture. You can replace the default logo by a custom logo in the following manner:

- Create a monochrome bitmap file with a custom logo picture. The width of the logo image in pixels should be a multiple of 16 for ADSP cameras.
- Run the batch file GENLOGO.BAT, which is contained in the software package:

  `GENLOGO {-cam|-ticam}` *`bmp_file`*

  where:

`-cam`	Generate output file for ADSP camera
`-ticam`	Generate output file for TI camera
`bmp_file`	*N*ame of the monochrome bitmap file.

  The batch file generates the file LOGOA.MSF. The batch file executes several DOS programs from the software package – BMP2BIN.EXE, BCONV.EXE and SCVT.EXE. Run the batch file in a folder, where you have access to these programs.
- Load LOGOA.MSF to the camera flash by the Send MSF function of the Camera Files Functions dialog.
- Start VIMOS Kernel on camera. The custom logo will replace the default logo in the overlay screen.

You should install a custom logo after the standard VIMOS installation on the camera.

3. VIMOS registration

The registration operation enables the execution of the VIMOS kernel on the camera. It should be done after the installation of the VIMOS system files on the camera. Send the serial number of your camera to the software supplier and receive a VIMOS key. Remember that you are able to register VIMOS when the **ent_key** module is present in the flash.

Start Simulator, open the **Camera Files Functions** dialog and perform the following operations:

- Paste the VIMOS key in the **Vimos Key** field.
- Press the **Set key** button.
- Delete **ent_key** module from flash to free flash space. Select **ent_key** in the file list by left click and then press **Delete selected** button.
- Wait for the end of the delete operation and then press the **Pack Flash** button to pack flash.

***ATTENTION**. If you erase camera flash and reload kernel files, you must re-enter the registration code.*

Refer to "Using the Simulator" manual for more details about the "Camera Files Functions" dialog. Another possibility for VIMOS registration is a direct invocation of the **ent_key** module using a terminal.

4. Starting VIMOS kernel on camera

Start Simulator and select **Camera > Start Camera Vimos** or press the toolbar button  to start VIMOS kernel on camera. The mouse cursor will disappear from the PC monitor, but mouse commands are captured and sent to camera. You can switch between “Capture Mouse” mode and Windows GUI mode (visible mouse cursor) by **`Alt+M`**.

While the system is loading you will see a system logo displayed on the camera monitor and several running messages about:

- How many modules have been decompressed and loaded in percents.
- DRAM usage in percents.
- Flash usage in percents.

***INFORMATION**. You can start VIMOS kernel using the VCT terminal. Start the terminal in full DOS window and:*

- *Set 1200 bauds by **Alt-F1**.*
- *Capture mouse to send mouse commands to camera by **Alt-M**.*
- *Enter **`vm`** and press Enter key.*

***ATTENTION**. To ignore current I/O settings, which disable mouse usage (other I/O device), and regain control over the system by mouse commands, you should:*

- *Start VIMOS by Simulator or VCT terminal.*
- *While the system logo is displayed on the camera monitor move the mouse (not too much) to force recognition of the mouse.*

On entry VIMOS sets camera baud rate to 1200 for work with mouse. On exit VIMOS restores default baud rate of 9600.

5. ADSP camera environment

This chapter presents information about ADSP camera environment – the operating system, the file system, hardware features and some limitations, which concern VIMOS operation on ADSP-based cameras.

5.1. File system of ADSP camera

This section presents some general information about the flash memory and the file system of the ADSP camera. Flash file names have maximum length of 8 characters. Each file has type, which is seen by the "**dir**" shell command or in the file list, shown in the "Camera Files Functions" dialog of the Simulator. The file types are:

- 0 : executable
- 1 : text (ASCII)
- 2 : data (binary)
- 3 : JPEG file

5.1.1. Flash capacity, checking available flash space

The available flash space for user files on ADSP camera is 30 or 31 sectors (each sector is 64 Kbytes). After finding the camera, the "Camera Files Functions" dialog of the Simulator shows current free space of the flash in the "**Free Flash**" field. Using a terminal, you can see the free flash space by the "**mem -f**" shell command.

5.1.2. Flash packing

The flash memory of the ADSP camera has some peculiarities, compared to the hard disk of the PC. The file system of the camera rewrites existing files in the free area of the flash. The area, where a previous copy of the file resides, is not freed and it can't be used until a pack operation of the flash is performed. The same rule is valid when deleting files. Thus, after continuous work with the flash, a moment comes when the flash overflows.

***INFORMATION**. This problem is caused by the electrical properties of the flash PROM chip. The flash memory is organized in sectors of 64K bytes. Files are stored sequentially in the flash memory without gaps and several files may reside in one sector. To overwrite one byte in the flash memory, it is necessary to erase the whole sector the byte belongs to.*

When working with VIMOS on ADSP camera, you should perform regular packing of the flash. This can be done in one of the following ways:

- Before exit, VIMOS kernel prompts for a flash packing operation. If the flash is about to be full (for example less than 64K bytes remain free), exit VIMOS by packing.
- Another possibility is to enable automatic flash packing on VIMOS exit by the "General-purpose configuration" dialog.
- You may do dynamic packing during the execution of the user-program by the "Delete and pack" tool. This method should be used if VIMOS kernel works for a long time without interruption.
- The flash can be packed in non-VIMOS environment by the Simulator (see the "Camera Files Functions" dialog) or by the "**pk**" shell command of the camera when using a terminal program.

5.2. PLC input and outputs of ADSP camera

The ADSP camera has 4 input digital lines (IN0-IN3) and for 4 output digital lines (OUT0-OUT3), called **PLC** inputs/outputs. Each line can hold one digital value: 0=OFF (inactive) or 1=ON (active). VIMOS uses these lines to receive information from the outer world (trigger signals for example) and to pass control signals as a result of its operation. VIMOS has tools, which read input PLC lines and write output PLC lines.

The table below shows the electrical properties of the PLC lines of the ADSP camera:

PLC line	ON / OFF line voltage
IN0-IN3	0=no voltage 1=voltage present
OUT0-OUT3	0=no voltage 1=voltage present

5.3. VIMOS performance on ADSP camera

The best VIMOS performance on ADSP cameras is reached in "Live" run mode. This mode disables automatic image taking in the main system loop and you should use the “Take image” tool in your user-program.

5.3.1. Module reloading system overhead

The ADSP CPU of the camera has 16K program words and that is the maximum size of one executable module. VIMOS is composed of multiple separate executable modules, each with size less than 16K. When different tools are executed, different modules are loaded and discarded from the program memory, which increases the system overhead time – about 5-6 ms for each tool.

The approximate time for module reloading when VIMOS executes external EXEC module is several milliseconds.

5.4. VIMOS limitations on ADSP camera

This section describes are some VIMOS limitations due to insufficient ADSP hardware resources.

5.4.1. Limitations of “Find BLOBs” tool in connected mode

The maximum number of objects that can be processed is limited to **500**. The tool returns error 9005 in case of object overflow.

The maximum width of the work area is about **540** pixels (no limitations for the height of the work area). The tool returns error 2 (memory allocation error) in case of too large width.

5.4.2. No subroutines

Due to lack of resources you can’t use subroutines on all ADSP-based cameras like VC38, VC61, VCM40 and VCM50.