VIMOS SOFTWARE FAMILY

Using VIMOS on sensor camera

15 February 2006

atto
Systems

Wolf
Vision Systems

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 sensor 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

Under the copyright laws, neither the documentation nor the software may be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine-readable form, in whole or in part, without the prior written consent of atto-Systems Ltd or Wolf Systeme AG, except in the manner described in the documentation or the applicable licensing agreement governing the use of the software. All rights are reserved. Do not reverse-engineer. Do not modify or distribute without all of the documentation.
© Copyright atto-Systems Ltd. and Wolf Systeme AG

Atto-Systems Ltd.
Sofia, Bulgaria

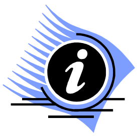
Wolf Systeme AG
Karlsbad, Germany

All rights reserved.

TRADEMARKS

All trademarks and copyrights mentioned within the documentation are respected. They are the property of their respective owners.

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 **sensor** ADSP cameras without video-output (**VCM40, VCM50**). These cameras provide the cheapest platform for the VIMOS system. The sensors 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 sensor 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”.*

2. VIMOS installation

The installation of VIMOS on a sensor camera includes two steps:

- Loading of MSF files into flash memory (MSF is a special file format accepted by the cameras).
- Registration, which permits execution of VIMOS kernel on camera.

The installation does not need to be performed in that order and registration can be done first.

Sensor cameras are usually equipped with 512 Kbytes flash (8 sectors x 64Kbytes). The first sector is reserved by the operating system of the camera. The remaining free space of 7 sectors is used for user files.

Due to hardware constraints of sensor cameras, VIMOS is composed of multiple executable modules, which call each other in a subroutine manner. VIMOS is a big system and all VIMOS modules can't be stored simultaneously in the free flash space of 7 sectors. Therefore VIMOS modules are grouped into base system modules (contained in VMALL.MSF) which are always loaded into flash, and plug-in modules, which are loaded optionally. There is approximately 1 sector, which remains free after the installation of the base modules. This space can be used for one or more plug-in modules, but not for all.

Sorry, but this is VIMOS limitation, imposed by the camera hardware. It may disappear when the camera manufacturer begins to yield sensor cameras with greater flash size.

2.1. VIMOS distribution file

A VIMOS system for sensor 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 (40 or 50)

Examples:

V251_40.ZIP	VIMOS version 2.51 for VCM40.
V251_50.ZIP	VIMOS version 2.51 for VCM50.

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

File	Description
VMALL.MSF	Merged MSF file with base system modules.
OCR.MSF BARCODE.MSF NCOR.MSF JPEG.MSF VMCMPPL.MSF VMUPX13.MSF VMUPX14.MSF	Plug-in MSF files…

ENT_KEY.MSF	Registration module
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, lights and quality (see the “Camera Files Functions” dialog).
M40_USER.TXT	Optional text file with latest installation information.

Further in this manual you will learn which files should be installed on camera and how. Some files like the plug-ins are optional and do not need to be used in the first installation. For example the file **VMAUTO.MSF** should be loaded to flash 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, lights and quality, which are different from the default ones.

2.2. Installation of base system modules

Upload **VMALL.MSF** to sensor camera. This is enough to run the base system configuration without the plug-in modules. Read section “2.4. How to load files to camera” to learn how to load MSF files to camera.

2.3. Installation of plug-in modules

If you are going to use tools from plug-in modules, load respective MSF files to flash (see section “*2.4. How to load files to camera*” for details). The plug-in modules overwrite dummy modules with the same names. Some plug-in modules need other plug-in modules - see the "Needs" column of the table.

Remember that you can’t install all plug-ins simultaneously in 512 Kbytes flash. Probably you will be able to install just one or two modules depending on the module size. Refer to section “*6.1.1. Flash capacity, checking available flash space*” to learn how to see the size of the free flash.

***ATTENTION**. New models of sensor cameras may have 2Mbytes flash. On these cameras you may load all plug-ins simultaneously.*

The system aborts execution with error 0x0008 (red LEDs = 0000 0000 0000 1000) when a plug-in tool is executed, but its respective plug-in module is not loaded (see chapter “*5. System errors*”)

The table below contains a list with all VIMOS plug-in modules. All tools not present in the table are contained in the base system modules.

Plug-in table:

Module	Size	Contains tools or utility modules	Needs
OCR	30720	Merged OCR utility modules vmcocr, vmrot, vmbin	
BARCODE	35161	Merged bar-code utility modules vmbcs, vmbc39, vmbc25	
NCOR	19244	Merged normalized correlation utility modules pminit, pmlrn, pmfind	
JPEG	5543	JPEG utility module	
VMCMPPL	16620	Compare point-lists	
VMUPX13	20548	Find BLOBs	
VMUPX14	18995	Point-list transform	

		Point-list sort Point-list filter Save point-list Load point-list	
VMUPX15	18997	OCR Bar-code reader Correlation init Correlation exec	OCR BARCODE NCOR NCOR
VMUPX16	18980	Point-list fast compare Point-list angle Match filter Point-list calculator	
VMUPX17	21092	Send point-list Receive point-list Send image (Camera only) Receive image (PC only) Send string Receive string Send result Receive result	 JPEG
VMUPX18	18148	Point-list best circle Point-list edge-detection Point-list median filter Point-list generate line Point-list generate circle	
VMUPX19	17057	Take CMOS image (M40 & M50 only) Color pixel counter	
VMUPX20	17434	Calibrate tool	
ENT_KEY	24252	VIMOS registration module	

Note: Sizes of modules may vary a bit for different camera models and different VIMOS versions.

2.4. How to load files to camera

Section "*2.4.1. Using the Simulator*" describes how to load MSF files by the Simulator. Currently the "Camera Files Functions" of the Simulator provides all means to install VIMOS on the camera. To communicate with the VIMOS shell (see "*4.2. VIMOS shell on sensor camera*") you should use a terminal.

Sections "*2.4.2. Using the VCT terminal*" and "*2.4.3. 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 a built-in terminal via which you may load MSF files to camera and enter shell commands.*

2.4.1. 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 base system modules (obligatory):

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

To load the registration module (obligatory, recommended to delete after registration, see "*3. VIMOS registration*"):

- Load **ENT_KEY.MSF** to flash.

Load plug-in module(s) if needed. Remember that some plug-ins require other plug-ins (see the plug-in table in section "*2.3. Installation of plug-in modules*"):

After you have done the registration you are ready to start VIMOS kernel on the sensor camera (see "*4. Using VIMOS kernel on camera*"). Refer to "Using the Simulator" manual for more information about the "Camera Files Functions" dialog and other Simulator functions.

2.4.2. 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.

2.4.3. 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.

3. VIMOS registration

The registration operation enables the execution of the VIMOS kernel on the camera. It can be done before (recommended) or 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 ENT_KEY.MSF has been loaded to flash.

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

- Load **ENT_KEY.MSF** to camera by the “**Send MSF**” button. You should see the **ent_key** module in the file list when the loading is finished.
- Paste the VIMOS key in the **Vimos Key** field.
- Press the **Set key** button.
- Delete **ent_key** module from flash. 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 VIMOS 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. Using 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`.

ATTENTION. *VIMOS on sensor cameras has no GUI interface and is not able to receive mouse commands (see "4.1. I/O defaults"). The serial interface is used to send and receive data via the I/O VIMOS tools. We strongly recommend disabling the "Capture Mouse" mode by* ***Alt+M*** *immediately after you have started VIMOS.*

INFORMATION. *You can start VIMOS from a terminal. Start the terminal and check that you have connection with the camera (press Enter to receive "$"). Enter "**vm**" and press the Enter key.*

During the system loading and initialization you will see a binary counter displayed by the green LEDs of the sensor camera. The counter shows the number of the module, which currently is being decompressed and loaded from flash to DRAM. In case of error, the operation finishes with endless error-display loop (see chapter "*5. System errors*").

4.1. I/O defaults

The sensor VIMOS does not support mouse. The system does not change the baud rate to 1200 and uses current camera baud rate when started. The default I/O settings are:

I/O Device = Other I/O device

I/O Baud Rate = 9600

VIMOS exits without restoring the baud rate to 9600. Note that executed user program may have changed the baud rate.

4.2. VIMOS shell on sensor camera

Sensor cameras have no video-output. Sensor VIMOS has no GUI interface and can't receive mouse commands for system configuration. Therefore VIMOS has special mode of operation in which you are able to communicate with the so-called VIMOS shell. This mode replaces the edit mode of VIMOS on cameras with video-output. The shell is invoked by a continuous pressing of the sensor camera button (at least one VIMOS cycle).

Use a general-purpose or VCT terminal to enter VIMOS shell commands (similar to entering of camera shell commands):

- The shell prompt character is **'>**' (different from the prompt of the camera shell).
- Press Enter key to execute the entered command.

Here is a brief description of the shell commands:

Command	Description
L *file*	Loads user program file.
R	Exits shell and continues user-program execution.

I *mode*	Sets illumination mode for automatic image taking (mode 1): *mode* = 0 : illumination and mark projector off *mode* = 1 : mark projector on *mode* = 2 : illumination on
M *mode*	Sets run mode type: *mode* = 0 : no shoot (no automatic image taking) *mode* = 1 : shoot (automatic image taking in each system cycle)
G *p1 p2 p3 p4*	Sets sensor gain (VCM40 only). The parameters *p1-p4* are 8-bit values in the range [0,255]: *p1* = erec = even row, even column gain (default = 150) *p2* = eroc = even row, odd column gain (default = 150) *p3* = orec = odd row, even column gain (default = 150) *p4* = oroc = odd row, odd column gain (default = 150)
X	Saves current system settings into the configuration file *ini.vm and* exits VIMOS. When started next time, VIMOS will run the last user-program loaded by the L command.
H	Prints this help (currently N/A)

Note: The VIMOS shell commands may be entered with upper-case or lower-case letters.

4.3. Taking pictures on sensor camera

One of the most essential features of the sensor camera is the lack of video-output hardware - you can't connect a monitor and see what pictures are taken and processed by the camera. You also can't see tool drawings in the second overlay picture. So the usual way to develop a new application on such camera is:

- Create a user-program by the Editor.
- Get some pictures using the "Camera Files Functions" dialog of the Simulator.
- Test the user-program in the Simulator with the obtained pictures.
- Finally run the program on the camera.

Remember that you may receive different pictures in Simulator and on the camera if you do not ensure the same conditions (shutter, lights, sensor gain) when taking the pictures. The VIMOS kernel takes pictures in 3 modes:

- In auto mode (one of the SHOOT run-modes), which takes picture automatically in each system cycle without the need of picture-taking tool(s).
- By the "Take image" tool, usually used to take pictures on non-sensor cameras with video-output.
- By the "Take CMOS image" tool, which takes images on cameras without video-output like M40, M50 and FA45. This tool supports special features like setting sensor gain and lights control.

Remember to load VCTIO.MSF to flash if you want to control JPEG quality and lights from Simulator. Remember that JPEG compression is not a lossless one - the JPEG picture is not a pixel-to-pixel equivalent with the original camera picture. You may receive different results on camera and PC with visually equivalent pictures.

We recommend checking the quality of the taken pictures on cameras without video-output in the test and/or the production stage of your application. Send pictures from camera to PC for example by putting "Take CMOS image" and "Send image area" tools in the camera program. On PC you can receive the pictures by the VIMOS recorder or by the Simulator, running a program with a “Receive image area” tool.

5. System errors

System abort error codes are displayed in endless loop by red LEDs 3-0. The 16-bit error code is displayed as 4 hex digits, starting from the most-significant digit. Green markers (all 6 LEDs on) separate the hex digits. Each hex digit is displayed for about 0.7 sec. A wait time of about 1.5 sec occurs between last and first hex digit of the error code. Press continuously the M40/M50 button to break the endless error-display loop and exit VIMOS.

INFORMATION*. Initialization errors, which occur before starting the main system loop, are printed on the terminal as well.*

For example the system error **`910 = 0x038E`** (QX_RC_MOD_NOTFOUND = module not found) is displayed consecutively by red LEDs 3-0 as shown below:

`0000` (first quad is displayed after wait time of about 1.5 sec)

`0011`

`1000`

`1110`

where: LED on = 1, LED off = 0

6. Sensor camera environment

This chapter presents information about sensor camera environment – the operating system, the file system and some hardware features, which concern VIMOS operation.

6.1. File system of sensor camera

This section presents some general information about the flash memory and the file system of the sensor 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

6.1.1. Flash capacity, checking available flash space

The available flash space for user files on sensor camera is 7 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.

6.1.2. Flash packing

The flash memory of the sensor 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 sensor camera, you should perform regular packing of the flash. This can be done in one of the following ways:

- 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.

6.2. PLC input and outputs of sensor camera

The sensor camera has 2 input digital lines (IN0, IN1) 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 sensor camera:

PLC line	ON / OFF line voltage
IN0-IN1	0=no voltage 1=voltage present
OUT0-OUT3	0=no voltage 1=voltage present

6.3. VIMOS performance on sensor camera

There is no special camera mode, which gives best performance of VIMOS on sensor cameras.

6.3.1. 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.

Similar overhead is valid when executing external EXEC modules.

6.4. VIMOS limitations on sensor camera

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

6.4.1. 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.

6.4.2. No subroutines

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