

VIMOS SOFTWARE FAMILY

Using the Simulator

28 February 2006



Thank you for your interest in our Vision Inspection and Optical Measurement System (VIMOS). In this manual you will find information about the VIMOS Simulator – a powerful and convenient tool for developing/debugging VIMOS user-programs using PC in the office or on the road. You will also find information about the ways to interact with the camera and transfer files between camera and PC.

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.....	6
1.1. REGISTER THE SIMULATOR	6
1.2. SIMULATOR WINDOW	6
1.3. PROJECTS	7
2. SIMULATION FUNCTIONS.....	9
2.1. SETTING WORKSPACE.....	9
2.2. CONFIGURING SIMULATION OPTIONS.....	10
2.3. IMAGE SOURCES	11
2.3.1. <i>Image files</i>	12
2.3.2. <i>Video file</i>	13
2.3.3. <i>Video capture</i>	13
2.3.4. <i>VC camera capture</i>	14
2.4. <i>STARTING VIMOS KERNEL SIMULATION</i>	15
2.4.1. <i>Starting Kernel simulation from Editor</i>	16
2.4.2. <i>Simulation of PLC lines</i>	16
2.5. IMPORT USER-PROGRAM.....	16
2.6. EXPORT USER PROGRAM	17
2.7. ENCRYPT USER PROGRAM	17
2.8. UPDATE IN EDITOR.....	17
2.9. AUTOMODE	18
3. GENERAL-PURPOSE CAMERA CONTROL FUNCTIONS.....	19
3.1. USING THE “CAMERA FILES FUNCTION” DIALOG	19
3.1.1. <i>Setting connection options</i>	20
3.1.2. <i>Changing the camera drive</i>	21
3.1.3. <i>Refreshing the file list</i>	21
3.1.4. <i>Automatic refreshing of the file list</i>	21
3.1.5. <i>Deleting flash files</i>	21
3.1.6. <i>Packing the flash memory</i>	21
3.1.7. <i>Erasing the flash memory</i>	22
3.1.8. <i>Sending MSF files</i>	22
3.1.9. <i>Sending VIMOS data files to camera</i>	22
3.1.10. <i>Reading VIMOS data files from camera</i>	23
3.1.11. <i>Reading pictures from camera</i>	23
3.1.12. <i>Changing camera baud rate</i>	24
3.1.13. <i>Entering VIMOS registration key</i>	24
3.1.14. <i>Waiting for camera boot</i>	24
3.1.15. <i>Aborting camera file operations</i>	25
3.1.16. <i>Entering the terminal</i>	25
3.1.17. <i>Exiting the “Camera Files Function” dialog</i>	25
3.2. SHOW LIVE PICTURE.....	26
3.3. SHOW LIVE VC PICTURE.....	26
3.4. TERMINAL.....	26
4. WORKING WITH VIMOS KERNEL ON CAMERA.....	28
4.1. INSTALLING VIMOS KERNEL ON CAMERA	28
4.2. COMMUNICATION OPTIONS	28
4.3. STARTING VIMOS KERNEL ON CAMERA	28
4.4. MOUSE SIMULATION	29
4.5. VIRTUAL SCREEN	29
5. VIMOS FILES	31

- 5.1. KERNEL FILES 31
 - 5.1.1. Kernel file names on PC and camera..... 31
- 5.2. USER-PROGRAM FILE FORMATS 32
- 5.3. OVERVIEW OF USER-PROGRAM EXCHANGE OPERATIONS 32
 - 5.3.1. Export user-program from Editor to Simulator 32
 - 5.3.2. Export user-program from Editor to camera..... 32
 - 5.3.3. Export user-program from Simulator to camera..... 32
 - 5.3.4. Export user-program from camera to Simulator..... 33
 - 5.3.5. Export user-program from Simulator to Editor 33
- 5.4. HOW TO RUN DEMO USER-PROGRAMS 33
- 6. GUI REFERENCE 34**
 - 6.1. FILE..... 34
 - 6.1.1. Change Workspace... 34
 - 6.1.2. Import MPR..... 34
 - 6.1.3. Export MPR... 34
 - 6.1.4. Encrypt... 34
 - 6.1.5. Update in Editor..... 34
 - 6.1.6. Exit 34
 - 6.2. EDIT 34
 - 6.2.1. Copy Image 34
 - 6.3. CAMERA..... 34
 - 6.3.1. Camera Simulation 34
 - 6.3.2. Simulation Options... 35
 - 6.3.3. Start Camera Vimos 35
 - 6.3.4. Mouse Simulation..... 35
 - 6.3.5. Camera Files Functions... 35
 - 6.3.6. Terminal... 35
 - 6.3.7. Communication Options... 35
 - 6.3.8. Image Source... 35
 - 6.3.9. Color mode..... 35
 - 6.3.10. Live Picture..... 35
 - 6.3.11. Live VC Picture 35
 - 6.3.12. Virtual screen..... 35
 - 6.3.13. PLC Inputs 36
 - 6.4. CAPTURE MOUSE/RELEASE MOUSE 36
 - 6.5. HELP 36
 - 6.5.1. Register..... 36
 - 6.5.2. About VIMOS Simulator... 36

1. Introduction

The VIMOS Simulator is a PC program, which simulates operation of VIMOS Kernel on VC cameras and provides additional functions for camera control. The Simulator functions are divided into three major groups:

- **Simulation functions.** Start Kernel simulator to create and run user-programs exactly in the way, they are created and run by VIMOS Kernel on VC camera. Export and import user-programs to/from the Editor. Use image files (BMP, JPEG,...), AVI-movies or PC cameras as image source.
- **General-purpose camera control functions.** Transfer various types of files – executable (MSF), user-programs, images, etc. between PC and VC camera. Show live pictures from PC cameras to check image source without starting Kernel simulation. Use general-purpose functions to load Kernel distribution file to camera, to register Kernel on camera and to maintain camera projects.
- **Working with VIMOS Kernel on camera.** Start VIMOS Kernel on VC camera. Use PC mouse to work with Kernel's graphical user interface (GUI).

The Simulator is the basic tool, which provides functions for data transfer between different VIMOS software components and camera control.

1.1. Register the Simulator

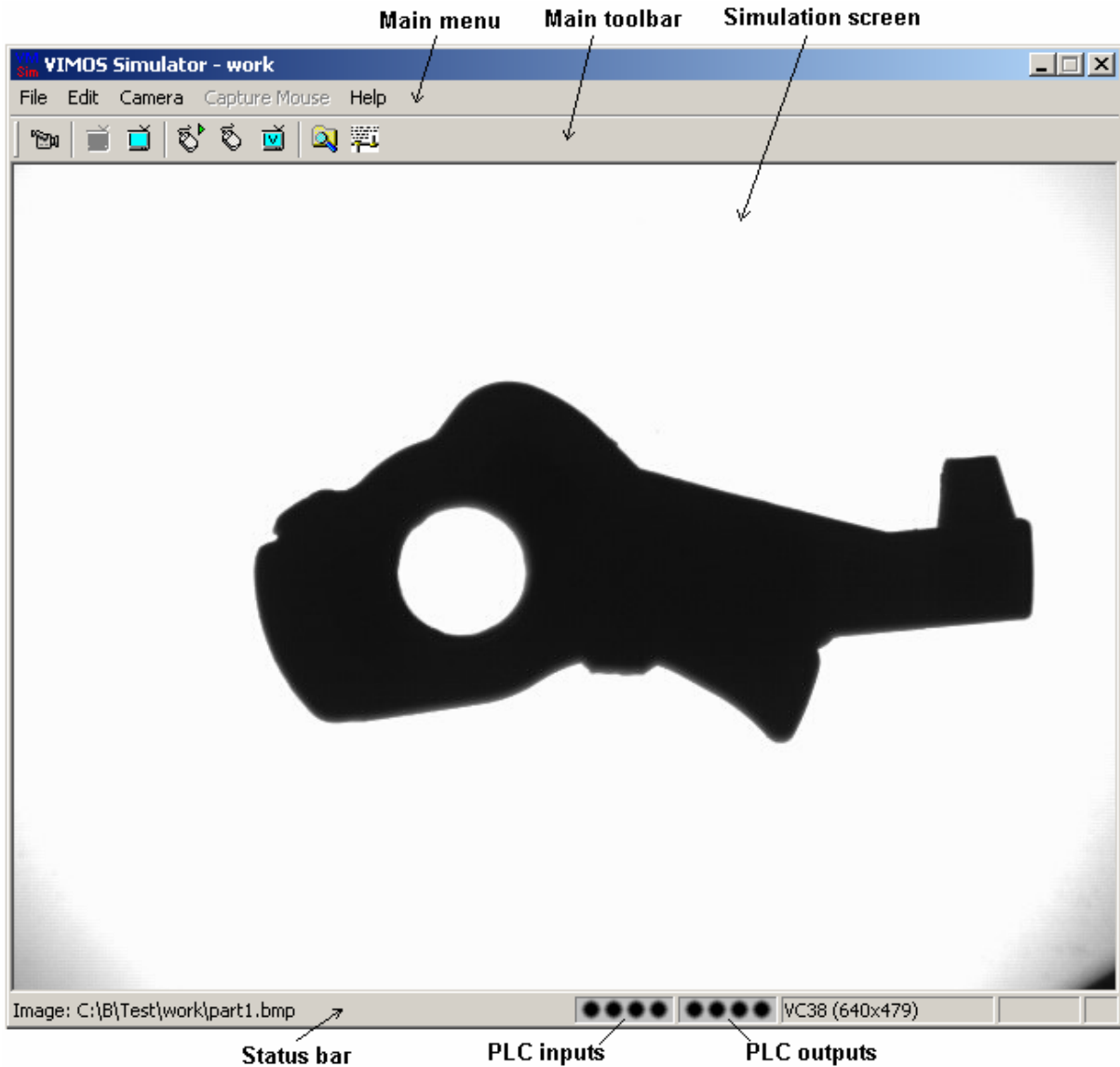
The VIMOS software family is distributed usually with a free 30-days trial period. After the trial period has expired, you will not be able to use it. When you buy the software the limitation is removed by the following registration procedure:

- Select **Help > Register** to open the **Register VIMOS Simulator** dialog. Send the **Product ID** number to the software supplier.
- The software supplier will send you back a registration key. Paste the key into the **Registration key** field of the dialog and press the **Register** button. The Simulator will inform you about a successful or failed registration.

1.2. Simulator window

The Simulator has a standard GUI window, seen in most Windows applications. The main elements of the Simulator window are:

- **Main menu** – provides access to all functions.
- **Main toolbar** – buttons for quick execution of frequently used functions.
- **Simulation screen** – simulates the video output of a VIMOS Kernel, running on a camera (screen of the camera monitor). Here you will see a gray picture, processed by the Kernel simulator, and a second single-color overlay picture of the Kernel's GUI.
- **Status bar** – displays current status or description of selected command. Includes 8 circle indicators for camera PLC lines (4 inputs and 4 outputs).



1.3. Projects

To implement a given image-processing algorithm on VIMOS kernel, you need to create a set of files called project. The project consists of user-program files and various data files. You may create a project entirely using the VIMOS Kernel on camera, but we recommend using the Editor and the Simulator for this purpose.

The process of creating and running a VIMOS Kernel project includes the following stages:

1. **User-program creation stage.** Create one or more user-programs using the Editor and/or the Simulator. Note that one user-program may load and execute other user-programs. A complex project may consist of a root user-program, which performs setup and control functions, and several second-level user-programs, which are called by the root program. Refer to manuals "Using the Editor" and "VIMOS Programming" for details.
2. **Simulation stage.** Start Kernel simulation and execute created user-programs (see section "2. Simulation functions"). Use this stage to debug user programs and generate data files. Store all files into the current workspace folder. Data files are created in two ways:
 - By the user program during its execution.
 - By interactive work with VIMOS Kernel's graphical interface (GUI). Such file for example is the calibration data file **clb1.vm**.

Use Kernel GUI to save user-programs, loaded dynamically from Editor to Simulator, into kernel-compatible files (see “5.2. *User-program file formats*” and “5.3. *Overview of user-program exchange operations*”).

3. **Transfer stage.** Transfer Kernel files from current workspace folder to camera (see section “3. *General-purpose camera control functions*”). Description of Kernel files can be found in section “5.1. *Kernel files*”.
4. **Execution stage.** Start VIMOS Kernel on camera (see “4. *Working with VIMOS Kernel on camera*”) and execute user-programs. This is the target production stage, which performs the machine-vision algorithms, coded in the user-program(s).
5. **Project maintenance stages.** Use the file transfer functions of the Simulator to save kernel files from camera to PC and to keep camera project on PC. This stage is especially important when you have made changes in user-programs and data files during Kernel execution on camera. Restore back kernel files from PC to camera when necessary. See “3. *General-purpose camera control functions*” and “5.1. *Kernel files*”.

2. Simulation functions

Use Kernel simulation functions to create and test user-programs in convenient way on PC. The Simulator allows development to be done not at the place where the camera is installed (usually a dirty and noisy factory environment) but in a comfortable office instead. The user-programs, developed by Simulator, are fully functional and compatible with VIMOS kernel on camera.

Before you start Kernel simulation, you should perform several setup operations:

1. Specify workspace folder. Use workspace folders for different camera projects (see “2.1. *Setting workspace*”).
2. Specify camera model (see “2.2. *Configuring simulation options*”).
3. Specify image source. The Kernel simulation works with images, read from one of the following image sources (see “2.3. *Image sources*”):
 - Sequence of image files (BMP, JPEG,...)
 - Video file (AVI)
 - Video for Windows (VfW) compatible video sources, such as cheap USB-cameras or more sophisticated video-capture cards
 - Vision Components (VC) cameras, connected to PC

The setup settings are saved in current project folder. Once done, you don't need to repeat setup operations unless you change project, camera model or image sources.

After you have done all the necessary setup, you are ready to begin Kernel simulation (see “2.4. *Starting VIMOS kernel simulation*”). You can create, modify and run user-programs. You can import and export user programs between Simulator and Editor.

2.1. Setting workspace

Use projects to organize your work. Each project includes a set of files, needed by the VIMOS Kernel to implement a given algorithm. The files are stored in a folder, called “*workspace*” folder. All user-program and data files, used by the Kernel simulator, are read from or written to current workspace folder.

The necessity of the projects comes from the fact that VIMOS Kernel uses fixed number of predefined file names due to the limitations of the embedded environment on the camera (see “5.1. *Kernel files*”). It is necessary to use equivalent names of user-program and data files in different projects. The solution of the problem is to keep project files in different folders and to load or save project files to/from camera when needed.

Remember to specify a new workspace folder for each new project, otherwise you may overwrite and lose user-program or data files. Editor user-programs files (AEF files) are not loaded directly into camera. They are first transferred from Editor to Simulator.




ATTENTION All files used by the VIMOS Kernel on the camera are stored in the flash memory, which plays the role of the PC disk. There are no folders in the camera flash. Be careful when working with several projects on camera because you can use limited number of user-program and data files with fixed names.


We recommend keeping files of different camera projects in different workspace folders on PC. Keep one project on camera at a time. Change camera projects by:

- Delete old-project camera files (user-program and data files).
- Pack the flash to increase free flash space.
- Copy new-project files from PC workspace folder to camera flash.

Select a workspace folder by:


- Make sure that simulation is turned off (**Camera Simulation** button  on the toolbar is not pressed).
- Select menu **File > Change Workspace...** to open the **Choose Workspace** dialog. Find existing or open a new workspace folder.



TIPS & TRICKS. To create a new workspace folder press the **Create New Folder** button  in the **Choose Workspace** dialog.

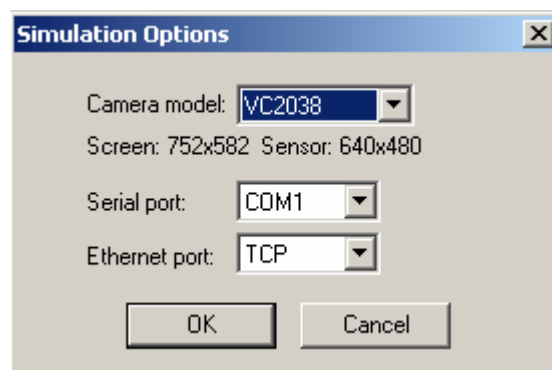
The Simulator stores an initialization file `simulator.ini` in each workspace folder. Press **Open** button to select workspace folder and close the dialog. Do not change the **File name**. The name of the current workspace folder is displayed in the Simulator window title.



ATTENTION. When you start the Simulator from the Editor, the folder of the current editor file becomes default Simulator workspace. Save your work into current workspace folder before running the Editor program in Simulator by pressing the Editor's **Run** button .

2.2. Configuring simulation options

Select menu **Camera > Simulation Options...** to open the **Simulation Options** dialog.



Use this function before you start Kernel simulation. The **Camera model** combo box allows you to select simulated camera model. VC cameras are similar in functions and resources, but usually have different sizes of processed pictures. To ensure correct results we strongly recommend specifying in Editor and Simulator the actual camera model, you are going to use in production. Note that TI cameras may have different dimensions of sensor and screen picture.

You can select a **Serial port** on your computer to be used by the I/O tools in serial mode (e.g. Send Result, Receive Result, etc.) during simulation. The "none" option disables the serial I/O transfers.

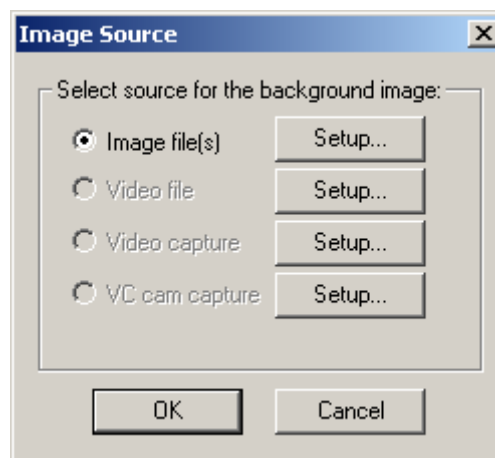
The **Ethernet port** combo box enables the execution of the I/O tools in TCP/IP mode. TCP devices are initialized (created or connected) by the I/O tools “Create server device” and “Client device connect”. The “none” option disables the TCP I/O transfers.

Note: An I/O device, which is not initialized as TCP device or as a special RS232/RS422 serial device by the “Open serial device” tool, is used as **default** serial device – please refer to chapter “8. I/O Tools” in the “Tools Description” manual for more information. The **Serial port** combo-box enables I/O transfer for the **default** serial device. You can open a RS232/RS422 serial device only on cameras like VC40xx and FA45, which have both Ethernet and serial interface.

2.3. Image sources

Before you start Kernel simulation you need to specify image(s), the user-program will work on. The currently processed image is displayed in the **Simulation screen** area of the Simulator window.

Select menu **Camera > Image Source...** to open the **Image Source** dialog.



Press a **Setup...** button to configure the corresponding image source. Refer to the following subsections for description of each image source.

Select one of the four available image sources:

- Image file(s)
- Video file
- Video capture
- VC camera capture

Note: Some of these options could be disabled when you open the **Image Source** dialog. The reason usually is that they are not configured yet. Use the respective **Setup...** button to do that and the option should become available. If even the **Setup...** button is disabled, the respective option is not supported by your current hardware/software environment.

Finally press **OK** button to accept specified image source. Press **Cancel** button to ignore changes and exit the dialog.

Note: Canceling this dialog does not cancel changes you have made in any configuration dialogs opened by the **Setup** buttons.



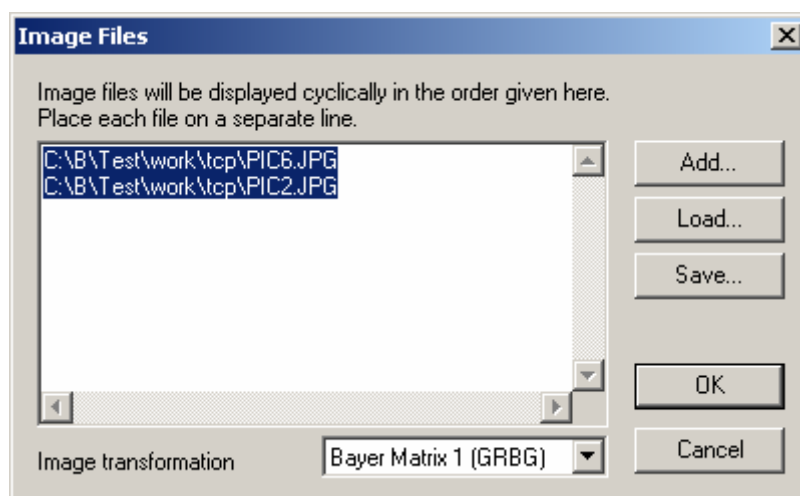
ATTENTION. Press the **OK** button in “Image Source” dialog to save setup information into current workspace folder. Once you have selected image source for current workspace, you don’t need to specify it any more before you start Kernel simulation. Each workspace supports a separate image source configuration.

2.3.1. Image files

Open the **Image Source** dialog. Press the **Setup...** button, which corresponds to the **Image file(s)** option and open the **Image Files** dialog. Here you can review the sequence of image files that your user-program will work on. The list is processed cyclically by the user-program, i.e. after reaching the end processing goes back to the beginning.

Use the following buttons to edit the image list:

- **Add** button. Select a folder, which contains image files. We recommend using the workspace folder or its subfolder for image files, needed by the current project. Select one or more image files. The selected file(s) will be inserted before the current line (the line containing the cursor) in the list, each file on a separate line. When you insert multiple files at once, they will be inserted in alphabetical order.
- **Load** button. Select a text file, which contains a list with image files. The current list will be replaced by the contents of the selected text file.
- **Save** button. Save current list in a text file. This file could be used later with the **Load** function.



TIPS & TRICKS. Use *Shift* or *Ctrl* key to select multiple files in the standard file-selection dialog.

The image list is displayed in a standard text box, so you can easily edit its contents. For example to clear the list simply delete all the text. An easy way to do this is to press the right mouse button inside the text box and to choose **Select All** from its context menu. Then hit **Del** button to delete all the text.

The **Image transformation** combo-box specifies how the image file is displayed in the simulation screen.

The first option "**None**" displays a gray-scale picture.

The next 4 options of the combo-box display a color image, coded in a grayscale image (1 byte per pixel) in one of four available Bayer matrix formats. Each 4 pixels in 2 successive rows and columns of the image specify one color pixel. These formats correspond to different color cameras as shown below.

- **Bayer matrix 1.** The top-left corner of the image is (M40, M50):

```
G(green 1) R(red)
B(blue)    G(green 2)
```

- **Bayer matrix 2.** The top-left corner of the image is (FA45):

```
R(red)     G(green 1)
G(green 2) B(blue)
```

- **Bayer matrix 3.** The top-left corner of the image is:

```
G(green 2) B(blue)
R(red)      G(green 1)
```

- **Bayer matrix 4.** The top-left corner of the image is:

```
B(blue)      G(green 2)
G(green 1)  R(red)
```

The way an image is displayed does not affect the work of the image processing tools. They will still process the actual grayscale image with 1 byte per pixel. To toggle this option during simulation, use **Camera > Color** mode menu or the **Alt+B** shortcut.

Press the **OK** button to accept the list, close the **Image Files** dialog and return to the **Image Source** dialog. Click on **Image file(s)** option to enable image files as image source. Exit the dialog by the **OK** button. After closing the **Image Source** dialog the first image from the list is displayed as background image in the simulation screen. The images are processed according to their order in the list from top to bottom. When the last image is processed the system returns to the first image.

New images are loaded:

- By each “Take image” or “Take CMOS image” tool in the user-program.
- Before each user-program iteration when automatic image acquisition is specified - some of the “shoot” run-modes.

It is a good practice to copy all image files needed to test your user program in the workspace project folder.

2.3.2. Video file

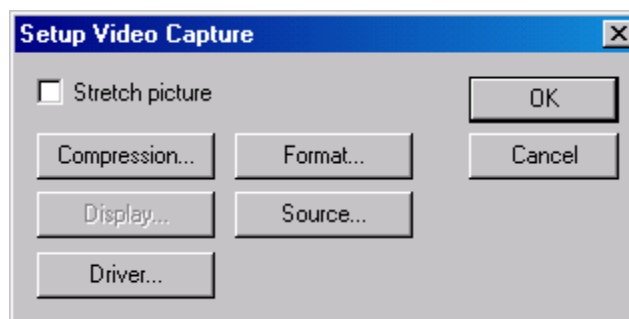
Open the **Image Source** dialog and press the **Setup...** button next to **Video file** option. Select a video (AVI) file and return to **Image Source** dialog. Select the **Video file** option to enable video file as image source and close the dialog by the **OK** button.

Next frame is taken:

- By each “Take image” tool in the user-program.
- Before each user-program iteration when automatic image acquisition is specified (one of the “shoot” run-modes).

2.3.3. Video capture

This feature works only with PC cameras compatible with *Video for Windows*. Most USB cameras meet this requirement. Open the **Image Source** dialog and press the **Setup...** button, next to **Video capture** option. Use the **Setup Video Capture** dialog to adjust various properties of the video capture process.



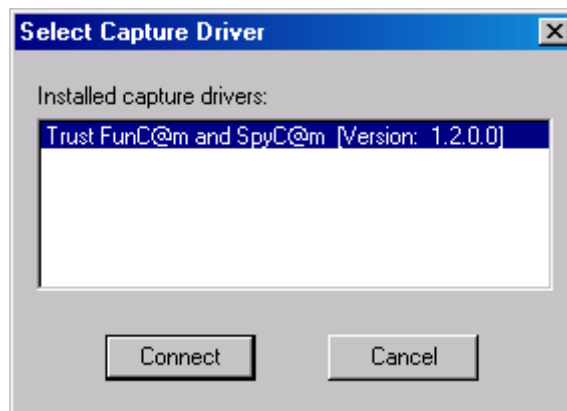
Dialog controls:

- **Stretch picture** – if checked, the captured frame will be resized to fit the size of the simulation screen. If not checked, the captured frame will be displayed with its original dimensions in the upper left corner of the simulation screen.
- **Compression** - displays a dialog box in which you can select a compressor to use during the capture process. The list of available compressors can vary with the video format selected in the capture driver's **Video Format** dialog box.
- **Format** - displays a dialog box in which you can select the video format. The **Video Format** dialog box might be used to select image dimensions, bit depth, and hardware compression options.
- **Display** - displays a dialog box, in which you can set or adjust the video, output. This dialog box might contain controls that affect the hue, contrast, and brightness of the displayed image, as well as key color alignment.
- **Source** - displays a dialog box, in which you can control the video source. The **Video Source** dialog box might contain controls that select input sources; alter the hue, contrast and brightness of the image; and modify the video quality before digitizing the images into the frame buffer.
- **Driver** – displays **Select Capture Driver** dialog box where you can choose a capture driver to use.

Most dialog controls remain disabled until you select a capture driver from **Select Capture Driver** dialog box.

Video Format, **Video Source** and **Video Display** dialog boxes are provided by the selected capture driver, so their contents may vary. If the selected driver does not provide some of them, the respective buttons will be disabled.

The **Select Capture Driver** dialog box lists all capture drivers installed on your system. Here you can choose which one to use.



Close the **Setup Video Capture** dialog by the **OK** button. Select **Video capture** radio-button in the **Image Source** dialog to enable video capture as image source. Close the dialog by the **OK** button.

2.3.4. VC camera capture

Open the **Image Source** dialog and press the **Setup...** button next to the **VC cam capture** option. You will open a **Communication Options** dialog. Set **COM port** and **Baud rate** for serial VC camera. Select **COM port = TCP** for Ethernet camera VC20xxE. Change the default IP address and port if necessary – for example when you connect to remote camera via Internet. Select name of temporary image file in **Temp file** by pressing the **Browse** button.

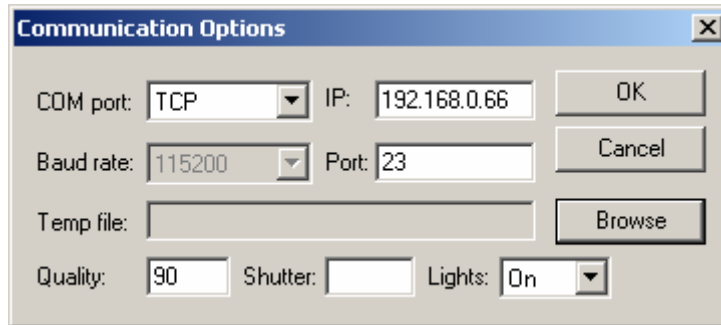
Images are transferred from camera to PC in JPEG format. **Quality** specifies JPEG quality factor (compression degree) in the range [1,100]. Greater quality factor means greater size and better quality of JPEG image but lower transfer speed. **Shutter** specifies camera sensor exposure time in microseconds (default = 10000). **Lights** controls the illumination on cameras without video-output

(sensor cameras VCM40, VCM50 and TI-based Fa45) - illumination off, mark projector on or illumination on.




ATTENTION. The **Quality** and **Lights** options require the **VCTIO** program to be present in the flash.

Press **OK** to return to **Image Source** dialog. Select **VC cam capture** option to enable VC camera as image source and close the dialog by the **OK** button.



ATTENTION. This way of getting pictures is slow when images are transferred via the serial port. Faster solution is to use a TCP camera or to save several images into image files by the file transfer dialog and then to use a sequence of image files.

2.4. Starting VIMOS kernel simulation

Select **Camera > Camera simulation** or press the **Camera Simulation** button  to start simulation of VIMOS Kernel. The mouse cursor disappears and you will begin to work with Kernel single-color graphical interface, displayed in the **simulation screen** area of the Simulator window. Switch between VIMOS Kernel GUI and normal Windows interface when the mouse cursor is visible by **Alt+M**. For example you can exit temporarily Kernel simulation by **Alt+M**, start another Windows application, then select Simulator window and return to Kernel simulation by second **Alt+M**.



ATTENTION. The Simulator captures mouse during Kernel simulation to achieve exact camera simulation. The Simulator needs to receive all mouse signals exclusively and it cannot share the mouse with Windows.




TIPS & TRICKS. While the mouse is captured by the Simulator you can still switch to other applications with **Alt+Tab**. This will release the mouse automatically. You can also use Simulator menus by holding down **Alt** and pressing the respective underlined letter. For example **Alt+F** will open the **File** menu and show the mouse pointer. The mouse pointer will be automatically hidden when the menu is closed.

The VIMOS Kernel has two modes of operation - run and edit. Execute user-programs in run mode. Create/modify user-programs and configure the system in edit mode. When started, the Kernel enters automatically run mode and executes the last user-program (if any) in endless loop.

Open the **Run main menu** by right mouse click (the run mode is not stopped). The menu will be displayed after finishing the current loop cycle. Terminate user-program execution, close the menu and enter edit mode by left mouse click on the **Stop run-mode** option. Open the **Edit main menu** by right mouse click. It contains options to **Start run-mode** and to **Exit** the Kernel (stop simulation). Terminate Kernel simulation and return to Windows interface (visible mouse cursor) by left click on the **Exit** option.

Abort Kernel simulation by:

- **Alt+M** to enable Windows interface.
- Click on  button to abort Kernel simulation.







ATTENTION. If you abort the simulation, you may lose changes in current user-program and configuration data. We recommend using **Edit main menu > Exit** command to stop simulation.

Refer to manual “Using the VIMOS Kernel” for detailed description of Kernel usage.

2.4.1. Starting Kernel simulation from Editor

You can start Kernel simulation in Editor by:

- Start Editor and open a user-program file. Press the **Run program** button  in the Editor toolbar. The Editor starts the Simulator (if not started already), starts Kernel simulation, loads the Editor program into Kernel and enters run-mode to start execution of the program.

Start Editor and open a user-program file (AEF file). Open a tool’s context menu by right click on its icon. Select **Edit in Simulator** option. The editor program is loaded into Simulator and the Kernel is started in edit mode. Move or configure the selected tool. Editor and simulator versions 2.80 and higher support interactive move/resize/rotate operations by the mouse using the buttons ,  and  respectively. The **M** button is used to move tools with 2 or more point arguments. Terminate editing of current tool and return to parent “Configure program element” menu by right mouse click. Configure other tools if necessary. Press **F6** key to update the Editor program. You can change tool arguments and link results in this mode, but you can’t add or delete user-program tools.

2.4.2. Simulation of PLC lines

The circle indicators, placed on the status bar of the simulator window, simulate the PLC inputs and outputs of the camera (see “1.2.Simulator window”). The first (left) 4 indicators represent the PLC inputs. Click on a given indicator to toggle the corresponding PLC input (from left to right – IN0, IN1, IN2, IN3). You can also use **Alt+1**, **Alt+2**, **Alt+3**, **Alt+4** to do this. The right 4 indicators represent the PLC outputs. They are set or reset by your user-program during execution.

Here is an example:



This figure represents the following line states:

Input line (green)	State	Output line (red)	State
IN0	1	OUT0	1
IN1	1	OUT1	0
IN2	0	OUT2	1
IN3	0	OUT3	0

2.5. Import user-program

The easiest method to import user-programs from Editor is to use **Run in Simulator** and **Edit in Simulator** Editor’s functions. They are described in section “2.4.1 Starting Kernel simulation from Editor”. This method will allow you to reflect any changes made in Simulator back in Editor (see section “2.8 Update in Editor”).

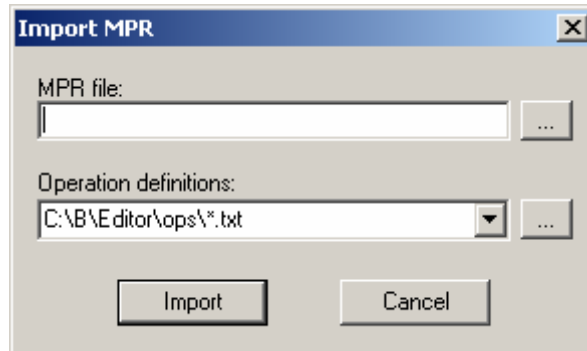
Another method to exchange programs between Editor and Simulator is to store user-programs into meta-program files (MPR). MPR files are created by:

- Editor. Use **File > Export** menu option to save current program into MPR file.
- Simulator. Use **File > Export MPR** menu command. See “2.6. Export user program”.

Start simulation of VIMOS Kernel and enter edit mode. Release the mouse by **Alt+M**, select **File > Import MPR...** and open the **Import MPR** dialog (or directly press **Alt+F** followed by **M** key without **Alt+M**).

Enter name of **MPR file** or select a file by the “...” button. The simulator supplies default folder for **Operation definitions**. Press **Import** button to load the user-program from the selected **MPR file**. The simulator informs about successful or failed import operation.

Now you can edit the program, start it or save it into a Kernel-compatible file with predefined name (*up0.vm, up1.vm,...*).



2.6. Export user program

Start simulation of VIMOS Kernel and enter edit mode. Load a user-program, which should be saved into MPR file (use **Edit main menu > Load user-program from file...** if necessary). Enable Windows interface by **Alt+M**, select **File > Export MPR...** and open the **Export MPR** dialog. Another way to open the dialog is to press **Alt+F** followed by the **E** key. Enter name of **MPR file** or select a file by the “...” button. The simulator supplies default folder for **Operation definitions**. Press **Export** button to save current user-program into specified MPR file. The simulator informs about successful or failed export operation.

2.7. Encrypt user program

This option encrypts current user-program. Start simulation of VIMOS Kernel and enter edit mode. Load a user-program, which should be encrypted (use **Edit main menu > Load user-program from file...** if necessary). Enable Windows interface by **Alt+M**, select **File > Encrypt...** and open the **Encrypt user-program** dialog. Select minimum and maximum serial numbers of cameras, which should run the encrypted program. Enter name of encrypted user-program file and press OK.



ATTENTION. Encrypted user-programs can be run on TI cameras only. The simulator is not able to load and run such programs.

2.8. Update in Editor

When you edit in Simulator a user-program that has been transferred by the Editor (through **Run program** or **Edit in Simulator** commands), you can save changes back in the Editor by **File > Update in Editor** menu command or **F6** key. See section “2.4.1 Starting Kernel simulation from Editor”.

2.9. AutoMode

Sometimes it is necessary to start executing a user-program when the Simulator is launched. To achieve this, pass `/AutoMode` command line option to `vimos.exe`. This will start the Simulator in the last used workspace as usual and will automatically enter simulation mode, starting the last used user-program. In this mode the user is allowed to use only the VIMOS kernel user interface, i.e. all Simulator menus and buttons are disabled. The user cannot even close the Simulator.

The user interface is enabled again when the simulation is terminated. But you can prevent even this by using the GUI Setup tool in your user-program. In this way you will be sure that the user-program will run and the operator will not be able to do something wrong by accident.

3. General-purpose camera control functions


This group of simulator functions performs general-purpose camera control. Here are all functions, which work directly with the camera operating system – file upload/download, getting *images* from camera to PC, erasing and packing camera flash, etc.

A special case is the registration of the VIMOS Kernel on camera, which is performed by the **Camera Files Functions** dialog.



ATTENTION. Use these functions to install VIMOS Kernel on camera before starting to work with the Kernel. See “Installation Guide” for details.

3.1. Using the “Camera Files Function” dialog

Select **File > Camera Files Functions** or press the **Camera Files Functions** toolbar button  to open the **Camera Files Functions** dialog. The dialog is used to:

- Work with VC camera flash - view file directory, delete file(s), pack/erase flash memory, change VC20xx/VC40xx camera drives (fd: or md:).
- Send PC files to camera and receive files from camera to PC.
- Get images from camera.
- Set VIMOS registration key on camera.

The first thing you should do when you enter this dialog is to specify connection options – see “3.1.1. Setting connection options”.

The **Camera Files Functions** dialog:

Camera Files Functions

Camera Info

Model VC2038 Serial Number 5900107 DC 10/16/03

Num Files 15 Free Flash 492947 Baud 9600

Drive Flash EPROM (fd:) Path fd:/user/

Name	Type	Start	Len
#IP	001		00000026
key.vm	002		00000012
vctio	000		00004B09
vmgui	002		0002DFF6
vmgimd	002		00018EC2
vmlogo	002		000003C4
logoa	002		000003A0
logob	002		000003A0
vmlogol	002		00003F44
vmimes	000		000075F1
vm	001		0000000F
tcpkey	002		00000010
up0.vm	002		0000280D
tb	000		00007F97
ini.vm	002		0000005C

Camera is connected to

TCP 192.168.0.66 Port 23

Files

Binary send (faster) Auto refresh files list

Refresh Files Send MSF Delete selected

Pictures

Freeze Buffer Picture Shutter: 10000 usec

Set as Background

Lights: Off Qual: 80 Take Pic

Data files

Get Send

Up/Download files as: Binary data

Vimos Key

Set key

Other

Pack Flash Clean Flash

Camera found... 0:02

Find Camera Wait boot... Terminal Close



ATTENTION. You are able to use camera files functions when VIMOS Kernel is not started on camera and the simulator program communicates with the shell of the camera operating system. See “3.1.14. Wait camera boot”.

3.1.1. Setting connection options

Before you gain access to camera you must specify connection options (if different from the default ones) in the group “**Camera is connected to**”. Select COM port and baud rate for serial camera (ADSP cameras **VC38**, **VC61**, sensor cameras **VCM40**, **VCM50** and TI cameras **VC20xx**). Select **TCP** option, IP address and TCP port for Ethernet camera **VC20xxE**.

Press the **Find Camera** button to establish connection with the camera. If the connection fails you will see the message “Camera not found!”. In case of successful connection you will see a “Camera found” message. The **Camera Info** group will show camera parameters – model, serial number, free flash space, etc., and a list of flash files (the list may be empty).

If you don't know the baud rate of a serial camera, you can:

- Check **Auto find** and press **Find Camera** to find automatically the baud rate of the camera. Note that you can't find automatically the COM port.
- Set default camera baud rate (9600) by power reset.

3.1.2. Changing the camera drive

Some TI cameras VC20xx have two file drives:

- Flash EPROM (**fd:**) – standard non-volatile memory present in all VC cameras
- Multi-media card (**md:**) – abbreviated also as MMC. This drive is used to access also the SD card on VC40xx cameras.

Use Drive dropdown to select the current drive. The file list is updated to show the files on the current drive.



ATTENTION. Functions that work with files like **Send MSF**, **Delete**, **Get** and **Send** use the current drive.

The **VCTIO** module should be loaded in camera flash (the **fd:** drive) for file transfers to/from MMC. If you try to send or receive a file from MMC and VCTIO could not be found, you will get an error message.

3.1.3. Refreshing the file list

Press the **Refresh Files** button in the **Files** group to re-read directory of current camera drive and update the file list. Use this function after execution of dialog operations, which change the file list.

3.1.4. Automatic refreshing of the file list

Check **Auto refresh files list** to perform automatic update of file list after each file operation, which changes the file list.

3.1.5. Deleting flash files

Select one or more items in the file list by:

- Click on a list item to select a single file name
- **Ctrl**+click to select a group of non-sequential files
- **Shift**+click to select a group of sequential files

Press **Delete selected** button in the **Files** group to delete selected file(s) from flash. Press **Refresh Files** to see updated directory of flash files or use **Auto refresh files list** for automatic update.



ATTENTION. The free flash space is not increased automatically when you delete files (see the **Camera Info** group). Press **Pack Flash** button to regain unused memory.

3.1.6. Packing the flash memory

The flash memory plays the role of the PC hard disk, but has some peculiarities. The file system of the camera places new files into the free area above the existing files. The areas occupied by previous copies of existing files or deleted files are marked as unused, but can't be overwritten by newer files. In other words the free flash space is decreased each time we store a file into flash and does not increase when we delete files. Regular packing regularly of flash memory can solve this problem.



INFORMATION. This problem is caused by the electrical properties of the flash PROM chip. The flash memory is organized in sectors of size 64K bytes. To overwrite one flash byte, a whole sector must be erased.

You don't need to pack the MMC drive.

Press the **Pack Flash** button from the **Other** group to pack flash files. Wait until the "Flash packed" message appears on the status bar. Press **Refresh Files** to see updated **Free Flash** space in the **Camera Info** group or check **Auto refresh files list** for automatic update.

3.1.7. Erasing the flash memory

This operation resembles the formatting of a hard disk. All files in the Flash EPROM (fd:) are erased (and lost). Erasing of flash memory is recommended before the installation of the VIMOS Kernel on the camera.

Press **Clean Camera** button from the **Other** group to erase all flash files. Wait until the “Flash erased” message appears on the status bar. Press **Refresh Files** to see updated **Free Flash** space in **Camera Info** group, or check **Auto refresh files list** for automatic update. The file list should become empty.

3.1.8. Sending MSF files

The program development system of the camera generates PC files in the so-called S-record format (MSF files). This simulator function is used to transfer MSF files from PC to camera, the Kernel distribution file VMALL.MSF for example.



INFORMATION. MSF files contain different types of camera files – executable, text, binary and JPEG.

Press the **Send MSF** button from the **Files group** to enter the **Open** dialog. Select a name of MSF file. Press the **Open** button or hit the **Enter** key to start transfer of selected file to camera. Wait while transfer-progress information is displayed on the status bar. Press **Refresh Files** to see updated file list or check **Auto refresh files list** for automatic update.

On serial cameras you may change the transfer speed by selecting a new value in the baud rate combo box (in the **Camera is connected to** group). Use higher transfer speeds for big MSF files (see “[3.1.12. Changing camera baud rate](#)”). Check **Binary send (faster)** to set binary transfer of MSF files, which is about two times faster than ASCII transfer. The binary transfer is currently not supported for TI cameras.

Note: Sending of MSF files directly to the MD drive requires the **VCTIO** program to be present in the camera flash.

3.1.9. Sending VIMOS data files to camera

Use this simulator function to transfer VIMOS data files from PC to camera.



INFORMATION. VIMOS data files are:

- User-programs created and/or saved by the Kernel simulator in edit mode.
- Data files generated by the Kernel simulator during execution of user-programs – image files, OCR files, etc.
- User-programs and data files created by the VIMOS Kernel on camera, which have been transferred from camera to PC.

Select file type by **Up/Download files as** combo box. Press the **Send** button from the **Data files** group to bring the **Open** dialog. Select name of a PC file. Press the **Open** button or hit the **Enter** key to open the **Camera File Name** dialog. Enter name of a camera file or accept the default name (8 chars max).



ATTENTION. Camera file names are maximum 8 characters long, including the file extension. Section “5.1. Kernel files” contains information about default names of VIMOS data files. Enter a predefined file name if you are going to use the file with camera VIMOS.

Kernel data files have identical formats on PC and on camera with one exception – image files **area...** The VIMOS Kernel on PC uses image files in standard bitmap (BMP) format. The VIMOS Kernel on camera uses image files in special internal bitmap-like format. A conversion between the two formats is done on the fly when you transfer image files.

Press **OK** to start transfer. Wait while transfer-progress information is displayed on the status bar. Press **Refresh Files** to see updated file list or check **Auto refresh files list** for automatic update.

Change transfer speed by selecting a new value in the baud rate combo box (**Camera is connected to group**).

Note: proper version of **VCTIO** tool should be present on camera flash to send data files to MMC (md:).

3.1.10. Reading VIMOS data files from camera

Select file type by **Up/Download files as** combo box. Click on a file list item to specify camera file. Press **Get** button in **Data files** group to enter the **Open** dialog. Specify name of PC file. We recommend reading the data files into the workspace folder of the current project. Press **Save** button or hit **Enter** key to start file transfer from camera to PC. Wait while transfer-progress information is displayed on the status bar.

For serial camera you can change the transfer speed by selecting a new value in the baud rate combo box.

Note: proper version of **VCTIO** tool should be present on camera flash to get data files from MMC (md: drive).

3.1.11. Reading pictures from camera

Press the **Take Pic** button in **Pictures** group to open the **Save as** dialog. Specify name of JPEG file the picture will be saved into. Press the **Save** button to start transfer of camera picture to PC file. Wait while transfer-progress information is displayed on the status bar.

Check **Freeze Buffer Picture** to get constant picture from camera frame buffer, otherwise get live camera picture. Both options read picture from frame buffer, but the second option stores new picture into the buffer before the transfer.

Check **Set as Background** to use the taken picture as background image, which is shown in the simulation screen area of the main simulator window. The picture will be used as image source during Kernel simulation. This picture is ignored if other image source has been specified for Kernel simulation (see “2.3. Image sources”).

Shutter specifies camera sensor exposure time in microseconds (default = 10000).

Images are transferred from camera to PC in JPEG format. **Qual** specifies JPEG quality factor (compression degree) in the range [1,100]. Greater quality factor means greater size and better quality of the received JPEG file but lower transfer speed. **Lights** specifies illumination on cameras without video-output (sensors M40, M50 and FA45) - illumination off, mark projector on (M40/M50 only) or illumination on.



ATTENTION. **Qual** and **Lights** need the **VCTIO** program.

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.

3.1.12. Changing camera baud rate

The file-transfer functions **Send MSF**, **Send** and **Get** and **Take Pic** use the current baud rate on serial camera. Change camera baud rate by selecting a new value in the baud rate combo box. Successful baud-rate change ends with the message “Baud changed”. Press **Refresh Files** to read file list and check serial connection.

Set higher baud rates (max 115200) to achieve higher transfer speed, especially for MSF files and pictures.

3.1.13. Entering VIMOS registration key

Use this function to register VIMOS Kernel on camera. Send the serial number of the camera to the software supplier. Paste the VIMOS key you have received from the software supplier into **Vimos Key** field and press the **Set key** button.



ATTENTION. *This option is available when the Kernel distribution files are loaded into camera flash (VMALL.MSF for ADSP camera, VMSYS.MSF for TI camera, ENT_KEY.MSF for sensor camera).*

3.1.14. Waiting for camera boot

Use this function to gain control over camera, which has been running VIMOS Kernel in stand-alone mode. A flash file with name “autoexec” starts programs on camera power on. The autoexec file, which starts VIMOS, contains the following line:

```
vm
```

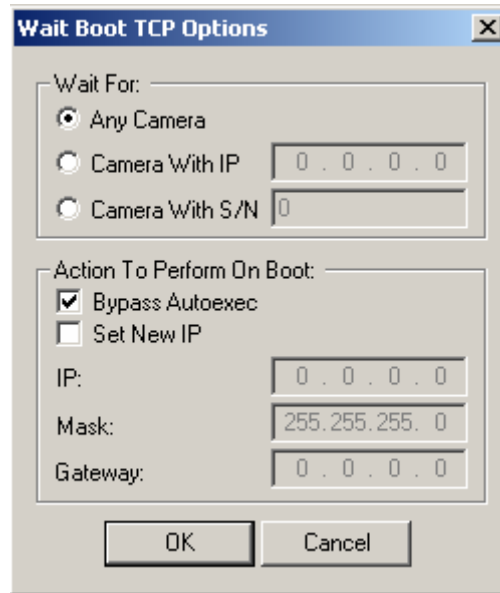
To bypass the execution of the autoexec file, switch the camera off and press the **“Wait boot...”** button. Your next operations depend on the type of the connected camera – serial camera (ADSP-based VCxx or TI-based VC20xx) or Ethernet camera (VC20xxE, VC40xxE, FA45):

Serial camera

Switch the camera on. You will see a window with a boot message of the camera’s operating system and a window with the results of a hardware test. Press **OK** to close the windows. In case of successful operation the simulator will update the camera parameters in the “Camera Info” group and will display a “Camera boot OK” message on the status bar.

Ethernet camera

The simulator will open the following dialog:



You can select two dialog operations:

- Check “Bypass Autoexec” if you want to skip the execution of the autoexec file.
- Check “Set New IP” if you want to change the current IP address of a camera.

The two operations can be enabled separately or simultaneously. The “Wait For” radio-button menu selects a camera for which the enabled operations are performed:

- **Any camera.** Performs selected operation(s) on the camera, which boots first (if several cameras are switched on simultaneously).
- **Camera with IP.** Performs selected operation (s) on the camera with the specified IP address.
- **Camera with S/N.** Performs selected operation (s) on the camera with the specified serial number.

Press the **OK** button to perform selected operation(s). The simulator will close the “Wait Boot TCP Options” dialog you will see the message “Waiting for camera to boot...” displayed on the status bar. Switch the camera on. After a successful “Bypass Autoexec” operation the simulator will update the camera parameters in the “Camera Info” group and will display a “Camera found...” message. After a successful “Set New IP” operation the simulator ends with a “New IP set” message.

3.1.15. Aborting camera file operations

The text of the button, which starts some camera file operation, is usually changed to **Abort**. Press the button to abort the operation if the camera hangs and you don’t see changing progress information for a long period of time.

3.1.16. Entering the terminal

You can enter the terminal directly from the “Camera Files Functions” by pressing the **Terminal** button (see “3.4. Terminal”). This operation makes automatically connection with the camera shell – press **Enter** several times to see the shell prompt ‘\$’.


3.1.17. Exiting the “Camera Files Function” dialog

Press **Close** button to exit **Camera Files Function** dialog and return to main Simulator window. This operation sets default baud rate of camera (9600).

3.2. Show live picture

Use this function to view live picture from video-capture device on the simulation screen when the Kernel simulation is off. The video-capture function works with cameras compatible with *Video for Windows*. Most USB cameras meet this requirement.

Configure video-capture device from **Camera > Image Source**. Open the **Image Source** dialog and press the video-capture **Setup...** button. Enter the **Setup Video Capture** dialog and specify video capture parameters (refer to “2.3.3. *Video capture*” for more details).

Select **Camera > Live Picture** or press **Live Picture** toolbar button . Live pictures are taken continuously from video-capture device and displayed on the simulation screen. Press same toolbar button to terminate the operation.



ATTENTION. This function is disabled when no video-capture driver is installed on your computer.

3.3. Show live VC picture

Use this function to view live picture from VC camera on the simulation screen when the Kernel simulation is off.


Configure VC-camera video-capturing by **Camera > Image Source** and open the **Image Source** dialog. Press the **Setup...** button next to the **VC cam capture** text to open the **Communication Options** dialog. Specify COM port and baud rate for serial camera or IP address and TCP port for Ethernet cameras VC20xxE/V40xxE. Select temporary image file and close the dialog (refer to “2.3.4. *VC camera capture*” for more details). You don't need to check the **VC cam capture** radio-button in the **Image Source** dialog.

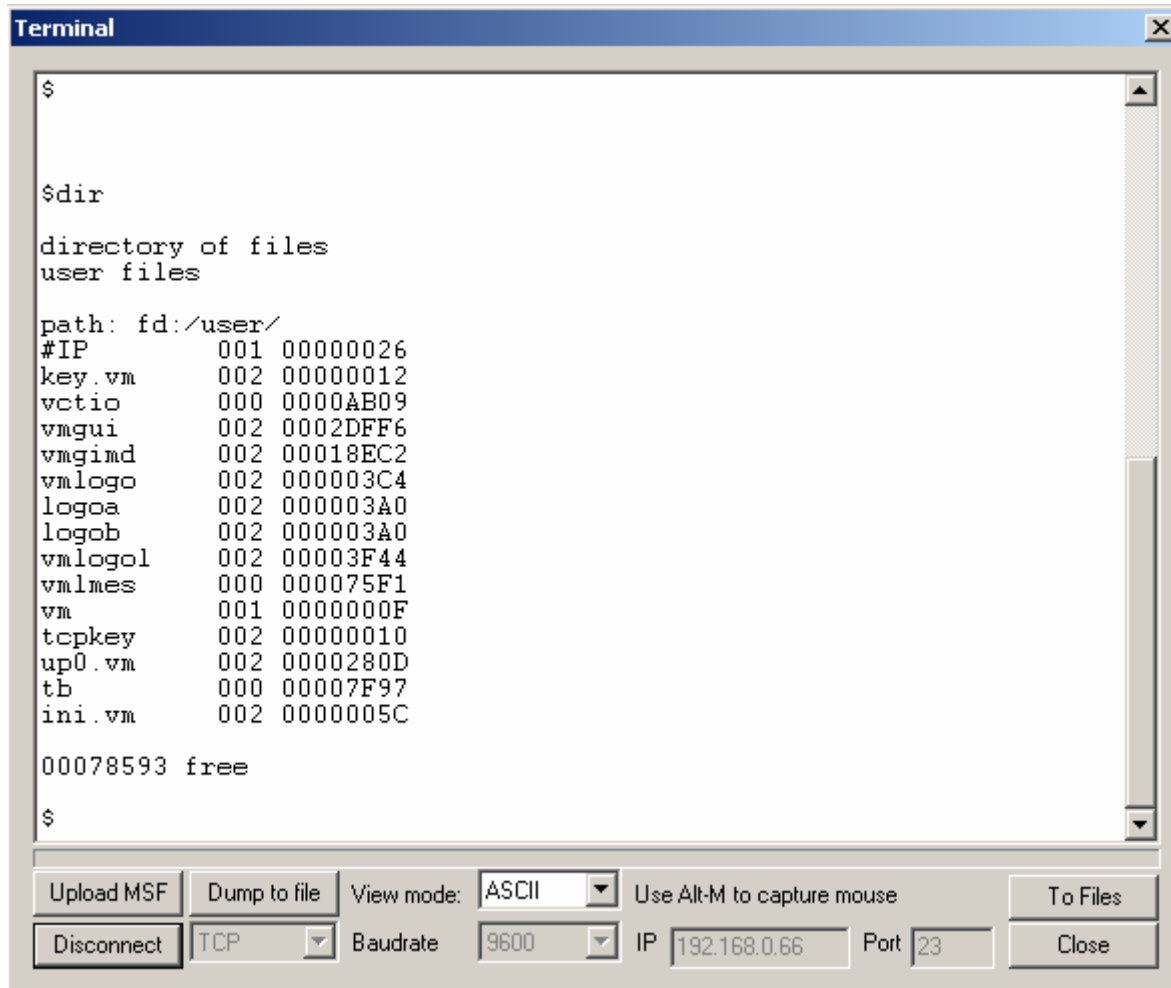
Select **Camera > Live VC Picture** or press **Live VC Picture** toolbar button . Live pictures are taken continuously from VC camera and displayed on the simulation screen. Press the same toolbar button to terminate the operation.



INFORMATION. Live pictures are read from VC camera in JPEG format. To achieve best transfer speed on serial cameras, you should set maximum baud rate (115200) from **Communication Options**.

3.4. Terminal

The simulator supports a general-purpose terminal, which can be used to communicate with the shell of the camera's operating system. Select **Camera > Terminal** or press the  toolbar button to open the following terminal window:



Specify a COM port and a baud rate for serial camera or an IP address and TCP port for Ethernet camera VC20xxE/VC40xxE or FA45. Press the **Connect** button to establish connection with the camera. After successful connection the text of the button is changed to **Disconnect**. Hit several times the Enter key to see the shell command prompt '\$'.

You may enter camera shell commands like:

```

dir    to see file directory
?      to print command help

```

The **Upoad MSF** button loads MSF files to the camera. The **Dump to file** button captures the output of the camera and saves it to a file. The **View mode** combo-box sets the type of the displayed/captured camera output – ASCII, mixed ASCII/hex or hex only. This option may be very useful when tracing non-ASCII camera output. Enter **Alt+M** to capture mouse and send mouse commands to camera. The next **Alt+M** command releases the mouse.

Press the **Disconnect** button to disconnect the camera.

Press the **To Files** button to close the terminal window, enter the “Camera Files Functions” dialog and make connection with the camera.

Press the **Close** button to disconnect and close the terminal.

4. Working with VIMOS Kernel on camera

Use this group of simulator functions to start VIMOS Kernel on VC camera and work with Kernel GUI using the PC mouse. These functions are disabled during simulation of VIMOS Kernel. Don't forget to set communication options before starting VIMOS Kernel on the camera. Once set for a given project, the communication options are saved and you don't need to set them each time before you start VIMOS on the camera.

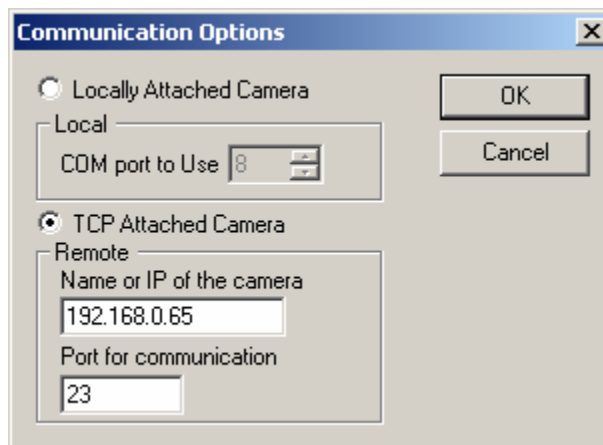
4.1. Installing VIMOS Kernel on camera

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**). Refer to the manual "**Installation Guide**" to learn how to install VIMOS on different camera models.


4.2. Communication options

PC mouse commands are captured and sent to camera when working with VIMOS Kernel on camera. You must configure mouse capturing by **Camera > Communication Options...**, which opens the **Communication Options** dialog.

If you are working with serial camera, enable **Locally Attached Camera** and COM port in use. If you are working with Ethernet camera VC20xxE, enable **TCP Attached Camera** and enter IP address and port if you need to change the default values.



4.3. Starting VIMOS kernel on camera

Select **Camera > Start Camera Vimos** or press 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. Switch between "Capture Mouse" mode and Windows GUI mode (visible mouse cursor) by **Alt+M**.



ATTENTION. Remember that before you start VIMOS on camera, you must configure mouse capturing by **Camera > Communication Options...**



You will see a system logo displayed on the camera monitor while the Kernel is booting. Don't move the mouse or press mouse buttons while system logo is displayed. The Kernel will recognize the PC mouse as a Microsoft Intelli Mouse.

The VIMOS Kernel enters run-mode by executing the current user-program (if any). The Kernel GUI is displayed on the camera monitor. Camera and Simulator mouse commands don't differ. Just to recall some basic commands, described for Kernel simulation:

- Open/close **Run main menu** by right mouse clicks (run mode is not stopped). The menu will be displayed after the end of the current user-program pass. Enter edit mode by left mouse click on **Stop run-mode** option (the edit main menu is not displayed automatically – see below).
- Open/close **Edit main menu** by right mouse clicks. Start run mode by left mouse click on **Start run-mode** option. Exit VIMOS Kernel by left mouse click on the **Exit** option.


Refer to the manual “*Using the VIMOS Kernel*” for detailed description of Kernel usage.






ATTENTION. The button  will remain pressed and mouse captured after exit from the camera Kernel, because there is no feedback from camera to Simulator while sending mouse commands. Return to Windows GUI by **Alt+M** and click on  to release the button.

4.4. Mouse simulation

Use this function to resume communication with VIMOS Kernel on camera, which is already started.



Select **Camera > Mouse Simulation** or press the toolbar button  to capture mouse and send mouse commands to camera. The response of your commands will be seen on the camera monitor. The mouse cursor will disappear from the PC monitor. Switch between “Capture Mouse” mode and Windows GUI mode (visible mouse cursor) by **Alt+M**. The **Capture Mouse/Release Mouse** option in the main Simulator menu shows current capture mode – enabled or disabled.



ATTENTION. To achieve exact camera simulation, the Simulator captures mouse and hides mouse cursor during Kernel simulation and mouse simulation (toolbar buttons ,  and ). This is necessary because the Simulator needs to receive all mouse signals exclusively and it cannot share the mouse with Windows.

Press **Alt+M** to release temporarily the mouse and use it in Windows. Press **Alt+M** again to recapture mouse and use it in camera simulation.

Example of VIMOS usage:







- Start VIMOS Kernel on camera by . Start user-program for inspection of machine parts, which runs twenty-four hours a day.
- Exit Simulator, leave the camera work in stand-alone mode.
- You need to gain control on VIMOS Kernel again and change some system settings. Start Simulator and press the mouse simulation button . PC mouse commands are redirected to camera. Stop run mode, enter edit mode and reconfigure the system.

4.5. Virtual screen

This option can be very useful on Ethernet TI-based cameras without video-output like FA45. You can't connect a monitor to such cameras and you can't see the captured gray-level picture and overlay drawings of your VIMOS program. The virtual screen helps to get rid (to a certain extent) of this drawback of the cheapest TI-based cameras. The virtual screen is slower than the real monitor.



INFORMATION. You can also use the virtual screen on Ethernet TI-based cameras VC20xxE/VC40xxE, which have video-output hardware.

Check **Camera > Virtual Screen** or press the “**Virtual screen**” button  to enable transfer of virtual screen from camera to PC. Then start VIMOS kernel on Ethernet camera by the  button. When VIMOS kernel starts, you will see a picture in the simulator window, which usually is displayed on a monitor, connected to the camera. You can enter for example edit mode by right mouse click or work with GUI tools and mouse cursor in run mode. Once you have started VIMOS on the camera with enabled virtual screen, the mouse simulation button  starts and stops the virtual screen. To stop receiving of the virtual screen, enter **Alt+M** and then click on  to stop the mouse simulation. Press  to restart the virtual screen. Enabling/disabling of mouse capture mode by **Alt+M** does not affect the virtual screen. Press  once again to dis



ATTENTION. Remember that the virtual screen increases the overhead time by 20-25 ms in each system cycle, which sends a screen picture to PC. The average transfer speed is about 8-10 screens per second for 640x480 pictures. If your program runs for example with 60 ms/cycle, approximately each second pass it will run for 80-85 ms. The actual time needed to transfer one screen from camera to PC is about 100-125 ms. The transfer is done in multi-tasking mode on the camera, in parallel with the running VIMOS kernel. If your program cycle time is greater than 100-125 ms, you will receive less than 8-10 pictures per second, i.e. your virtual screen will become slower.

5. VIMOS files

This section describes the files used by the VIMOS software components – Editor, Simulator and VIMOS Kernel on camera.

5.1. Kernel files

The VIMOS Kernel uses fixed number of predefined file names due to the limitations of the embedded environment on camera and the lack of keyboard connected to camera. File names are entered by dialog toggle controls with fixed number of states. Each toggle state specifies a file name. There are two general types of kernel files:

1. User-program files: **up0.vm, up1.vm, ..., up29.vm**. These are Kernel-compatible files (see “5.2. User-program file formats”).
2. Data files:
 - **st0.vm, st1.vm, ..., st29.vm** – statistics files with counter values.
 - **clb1.vm** – calibration file with calibration set data (all calibration sets).
 - **pl0.vm, pl1.vm, ..., pl999.vm** – point-list files.
 - **sb0.vm, sb1.vm, ..., sb999.vm** – string buffer files.
 - **ch0.vm, ch1.vm, ...** – OCR files.
 - **areaa0, areaa1, ..., areab0, areab1, ...** - image files.

The VIMOS software components, which create kernel user-program files, are VIMOS Kernel on Simulator and VIMOS Kernel on camera. Data files are created during execution of user-programs or in response to Kernel GUI commands.

Use **Send** and **Get** functions of “Camera Files Function” dialog to transfer kernel data files between PC and camera. The Kernel files have identical formats on PC and on camera with one exception – the image files **area...** The VIMOS Kernel on Simulator reads and writes image files in standard bitmap (BMP) format. The VIMOS Kernel on camera reads and writes image files in special internal bitmap-like format. Conversion between the two formats is done on the fly by the file-transfer functions **Send** and **Get** when you select file type **Image file** (see “3.1.9. Sending VIMOS data files to camera” and “3.1.10. Reading VIMOS data files from camera”).



ATTENTION. Don't mix up image files **area...**, generated by Kernel, with image files, used by the Simulator as image sources during Kernel simulation. For example you may get files from camera in JPEG format by the “Take Pic” function when camera-Kernel is not started.

5.1.1. Kernel file names on PC and camera

When you save kernel files from camera to PC, you may enter arbitrary long names of PC files (remember that maximum length of camera file names is 8 characters). We recommend however to use the same names due to the following considerations:

- You may save project files from camera to PC and later store files back to camera. You must restore the short names of the kernel files on camera. Keeping the same names will protect you from possible file mismatch errors.
- You may save project files from camera to PC and start Kernel simulation of project programs. The VIMOS Kernel on Simulator will use the same short file names, used by VIMOS kernel on camera.

5.2. User-program file formats

Different components of VIMOS software use different formats of user-program files. User-programs are exchanged between VIMOS components by the following methods:

- **Dynamic program loading.** This method exchanges user-programs between Editor and Simulator without intermediate files. Section “2.4.1. Starting Kernel simulation from Editor” describes how to load editor program in Simulator. Section “2.7. Update in Editor” describes how to update editor program from Simulator.
- **MPR file.** This file format is used to exchange user-programs between Editor and Simulator. See “2.5. Import user program” and “2.6. Export user program”.
- **Kernel-compatible file.** This file format is used to exchange user-programs between VIMOS Kernel on Simulator and on camera. You can use fixed number of predefined names of user-program files due to the limitations of the Kernel GUI - *up0.vm*, *up1.vm*, ..., *up29.vm*. Create Kernel user-program files by the **Save user-program to file ...** option in the edit main menu. Load Kernel user-program files by the **Load user-program from file ...** option in the edit main menu. Refer to section “6.2. Edit main menu” in the manual “Using the VIMOS Kernel” for details.

Convert editor user-program into a Kernel user-program file by the following Simulator operations:


- Load the user-program from Editor to Simulator, start Kernel simulation and enter edit mode.
- Save current user-program into Kernel-compatible file with one of the predefined names described above.

5.3. Overview of user-program exchange operations

Next sections present brief overview of operations for user-programs exchange between VIMOS system components - Editor, Simulator and Kernel.

5.3.1. Export user-program from Editor to Simulator


Use one of the following three methods:

- Press Editor button  to load current Editor user-program in Simulator and start VIMOS Kernel simulation in run mode.
- Right-click on a tool in the Editor to open its context menu. Select **Edit in simulator** option to load the program in Simulator and start VIMOS Kernel simulation in edit mode. Note that tools can't be inserted/deleted in this mode, only arguments can be modified. Hit **F6** to update the program in Editor.
- Select **File > Export** in main Editor menu to save current program into MPR file. Start Simulator and import MPR file from **File > Import MPR**.

5.3.2. Export user-program from Editor to camera


There is no direct link between Editor and VIMOS Kernel on camera. Export user-program from Editor to Simulator (see previous section). Export user-program from Simulator to camera (see next section).

5.3.3. Export user-program from Simulator to camera

Start Kernel simulation and enter edit mode. Save current user-program into a Kernel-compatible file, *up0.vm* for example. Exit Kernel simulation and press  to open the file-transfer dialog. Send *up0.vm* to the camera (see “3.1.9. Sending VIMOS data files to camera”). Start VIMOS Kernel on the

camera, enter edit mode and load **up0.vm** by **Load user-program from file ...** option in the **Edit main menu**. Start run mode to execute the program.


5.3.4. Export user-program from camera to Simulator

Start VIMOS Kernel on camera, enter edit mode and save current user-program into file **up0.vm** for example. Exit VIMOS Kernel to allow file transfer operations with camera. Start Simulator and open the file-transfer dialog by the  button. Read the flash file **up0.vm** to a PC file with the same name by the **Get** function (see “3.1.10. Reading VIMOS data files from camera”). Exit the file-transfer dialog and start simulation of VIMOS Kernel. Enter edit mode and load **up0.vm**. Start run mode to execute the program.

5.3.5. Export user-program from Simulator to Editor

Start simulation of VIMOS Kernel, enter edit mode and load user-program. Export the program to MPR file by **Alt-M, File > Export MPR...** (see “2.6. Export user program”). Start Editor and import the MPR file by the **File > Import...** option.

5.4. How to run demo user-programs

Start Editor and open the demo user-program file. Press toolbar button  to load and run the user-program in Simulator (see “2.4.1. Starting Kernel simulation from Editor”).

6. GUI reference

This section presents an overview of the main menu Simulator functions.

6.1. File

6.1.1. Change Workspace...

Set workspace folder for current project (see “2.1. *Setting workspace*”).

6.1.2. Import MPR...

Import user-program from MPR file (see “2.5. *Import user program*”).

6.1.3. Export MPR...

Export user-program to MPR file (see “2.6. *Export user program*”).

6.1.4. Encrypt...

Encrypt user-program (see “2.7. *Encrypt user program*”).

6.1.5. Update in Editor

Update user-program in Editor (see “2.8. *Update in editor*”). Enabled only when the current user-program has been dynamically loaded from the Editor.

6.1.6. Exit

Exit the Simulator program.

6.2. Edit

6.2.1. Copy Image

Copy to clipboard the current image (background + overlay) displayed in the simulation screen of the Simulator window. You can also use `Ctrl+C` shortcut.

6.3. Camera

6.3.1. Camera Simulation

Start simulation of VIMOS Kernel (see “2.4. *Starting VIMOS kernel simulation*”).

6.3.2. Simulation Options...

Specify simulation options – camera model, port for I/O tools (see “2.2. *Configuring simulation options*”).

6.3.3. Start Camera Vimos

Start VIMOS Kernel on camera and send PC mouse commands to camera (see “4.3. *Starting VIMOS Kernel on camera*”).

6.3.4. Mouse Simulation

Capture mouse and send PC mouse commands to VIMOS Kernel on camera (see “4.4. *Mouse simulation*”).

6.3.5. Camera Files Functions...

Open the Camera Files Functions dialog for camera file-transfer functions (see “3.1. *Using the “Camera Files Function” dialog*”).

6.3.6. Terminal...

Open the terminal window (see “3.4. *Terminal*”).

6.3.7. Communication Options...

Set communication options (COM port) before starting to work with VIMOS Kernel on camera (see “4.2. *Communication options*”).

6.3.8. Image Source...

Specify image source for Kernel simulation (see “2.3. *Image sources*”).

6.3.9. Color mode

Toggle Bayer matrix color image display (see “2.3.1 *Image files*”).

6.3.10. Live Picture

Show live picture from video-capture device on simulation screen (see “3.2. *Show live picture*”).

6.3.11. Live VC Picture

Show live picture from VC camera on simulation screen (see “3.3. *Show live VC picture*”).

6.3.12. Virtual screen

Enables or disables the virtual screen (see “4.5. [Virtual screen](#)”).

6.3.13. PLC Inputs

A submenu with items that toggle each PLC input line. See “2.4.2 Simulation of PLC lines”.

6.4. Capture Mouse/Release Mouse

Shows the state of the PC mouse – captured (mouse cursor off) or released (mouse cursor on). Select **Release Mouse** or hit **Alt+M** to release captured mouse, which is used to control VIMOS Kernel on Simulator and on camera.

6.5. Help

View the online VIMOS manuals.

6.5.1. Register...

Register Simulator program (see “1.1. Register the Simulator”).

6.5.2. About VIMOS Simulator...

Open “**About VIMOS Simulator**” dialog with program and version information.