

# V-REP

---

QUICK USER manual



version 1.4  
MAY 2012

---

## Documentation Author

Julien Tharin  
K-Team S.A.  
Z.I. Plans-Praz  
1337 Vallorbe  
Switzerland

Email: [info@k-team.com](mailto:info@k-team.com)

Url: [www.k-team.com](http://www.k-team.com)

## Documentation version

Version	Date	Author	Description
1.0	03.06.2010	J. Tharin	First draft
1.1	14.06.2010	J. Tharin	Updated
1.2	16.12.2010	J. Tharin	Updated floating license dongle; added Khepera 3 Lua parser
1.3	05.07.2011	J. Tharin	Some corrections
1.4	16.05.2012	J. Tharin	Updated license email address, versions for Apple and Linux

## Trademark Acknowledgements:

**IBM PC:** International Business Machines Corp.

**Macintosh:** Apple Corp.

**SUN Sparc-Station:** SUN Microsystems Corp.

**LabVIEW:** National Instruments Corp.

**Matlab:** MathWorks Corp.

**Khepera:** K-Team and LAMI

**Logitech:** Logitech Int. SA

**V-REP:** Dr. Marc Andreas Freese

## LEGAL NOTICE:

- The contents of this manual are subject to change without notice.
- All efforts have been made to ensure the accuracy of the content of this manual. However, should any error be detected, please inform K-Team.
- The above notwithstanding, K-Team can assume no responsibility for any error in this manual.

# TABLE OF CONTENTS

---

<b>1. INTRODUCTION .....</b>	<b>1</b>
1.1 V-REP .....	1
1.2 SPECIFICATIONS .....	1
1.3 REQUIRED HARDWARE/ SOFTWARE .....	3
1.4 HOW TO USE THIS HANDBOOK .....	3
<b>2. INSTALLATION .....</b>	<b>4</b>
2.1 THE SINGLE FIXED LICENSE .....	4
2.2 THE SINGLE DONGLE LICENSE.....	4
2.3 THE FLOATING LICENSE .....	4
<b>3. USAGE .....</b>	<b>6</b>
3.1 TUTORIAL.....	6
3.2 LOADING INTEGRATED SCENES.....	6
3.3 PROGRAMMING THE KHEPERA III ROBOT .....	7
3.3.1 <i>On V-REP simulator.....</i>	<i>7</i>
3.3.2 <i>Applying Lua script on the real Khepera 3 robot .....</i>	<i>8</i>



# 1. INTRODUCTION

---

## 1.1 V-REP

Thank you for buying the V-REP software. A powerful 3D robot simulator, which features several versatile calculation modules (inverse kinematics, Physics/dynamics, collision detections, minimum distance calculations, path planning, etc.), a distributed control architecture (unlimited number of control scripts, threaded or non threaded), and several extension mechanisms (plug-ins, custom client application, etc.).

V-REP is the perfect tool for fast prototyping and verification, remote monitoring, fast algorithm development, robotics related education, and simulation of factory automation systems. A V-REP plug-in was specifically developed to allow an intuitive, easy and realistic Khepera robot simulation.

This document introduces the instructions for the installation and the first steps in V-REP. For more information on how to use V-REP, please refer to the help file that ships with V-REP.

## 1.2 Specifications

- Full [collision detection module](#), allowing fast interference checking between any geometries or collection of geometries, with optional collision contour calculation.
- [Minimum distance calculation module](#), allowing fast minimum distance calculation between any geometries (convex, concave, open, closed) or collection of geometries.
- Full [forward/inverse kinematics calculation module](#), allowing handling any type of mechanism (branched, closed, redundant, containing nested loops, etc.)
- [Geometric constraint solver module](#), allowing to intuitively solve/actuate/interact with mechanisms
- [Path planning module](#), allowing holonomic and non holonomic path planning tasks
- Powerful, customizable and fast [dynamics calculation module](#), allowing to simulate near to real-world object interactions.
- Powerful, realistic and exact [proximity sensor simulation](#), fully customizable
- [Simulation of camera-like sensors](#) with image processing capabilities, fully customizable
- [Simulation of force and torque sensors](#), with a “breakable” option.
- [Simulation of cutting operations using mill objects](#)
- Various types of [joints](#) (revolute-, prismatic-, screw- or spherical-type)
- [Path or trajectory functionality](#) for elaborate kinematic movements

- [Script-driven](#), fully customizable behavior, also at low level.
- Unlimited number of scripts operating at the same time, non-threaded or threaded (threaded scripts operate flawlessly also on single-core CPUs).
- Minimal or no programming required for [simple simulations](#).
- Support for [extension modules](#) (i.e. plugins).
- Fully customizable [client application loop](#).
- Fully [customizable user interface elements](#) with [integrated edit-mode](#)
- More than [250 clean and fine-grained API functions](#), available from the C/C++ and Lua side (wrappers can make them available to other languages too)
- Fully integrated edit mode for robot/mechanism/scene edition ([triangle edit mode](#), [vertex edit mode](#), [edge edit mode](#), [custom user-interface edit mode](#), [path/trajectory edit mode](#), etc.) [4 in-parallel running simulator instances](#) sharing a same copy buffer.
- Full interactivity during simulation
- Full copy/paste functionality, not just for objects, but also for their associated behavior (programs), also during simulation
- Model [self-duplication](#) or [self-destruction](#) capability (from within the model itself), together with associated calculation objects and scripts.
- Exhaustive and comprehensive documentation
- Compact scene and model files (one single compressed binary file containing all what is needed (objects and programs))
- Powerful [data recording, visualization and export functionality](#)
- Exhaustive [CAD-data import and export](#) file-format support (DXF, 3DS, OBJ, STL)
- Built-in [AVI recorder](#)
- Powerful and flexible [wireless communication simulation and visualization](#)
- [Customized drawing objects](#) (points, lines, triangles, etc.). Allows simulation of paint, welding seams, etc.
- [Static and dynamic texturing](#)
- Multi-level undo/redo
- Objects and attached functionalities are fully scalable (resizable), also during simulation
- Simulation speed control also during simulation
- Compact and lightweight, with no dependency installations required, ideal candidate for embedding
- Possibility to lock scenes from further edition/modification, script content viewing or resource export
- Free V-REP PLAYER VERSION that allows you to distribute and play created scenes and models anywhere
- Very high level of in-house technology development and/or implementation

## 1.3 Required Hardware/ Software

- Required hardware :
  - PC, Linux or Apple computer
  - Minimal configuration:
    - CPU: >300 ([Passmark CPU Rating](#))
    - RAM: >=512 MB
    - Storage capacity: >50 MB
    - Video card: >200 ([Passmark G3D Rating](#)) (a good video card (i.e. hardware accelerated) is probably the most critical element of V-REP)
    - Monitor resolution: >=1024x768
- Required software :
  - OS:
    - Microsoft Windows Operating Systems: XP with SP3/Vista/Seven
    - Linux 2.6: Ubuntu, or another distribution
    - MacOS: from version 10.6
  - V-REP setup: “V-REP Vx.y.z Setup.exe”, where x,y,z are the current version of the software from <http://ftp.k-team.com/V-REP/>
  - optional: floating license type of V-REP: file “v\_repFloatingLicenseServer.exe”

Remark: Please go often check <http://ftp.k-team.com/V-REP/> to see if an update is available, as this software is constantly improving. You won't need to request any new license for updating.

## 1.4 How to use this handbook

This handbook introduces the V-REP software and how to start using it. For more information on how to use V-REP, please refer to the help file that ships with V-REP.

If this handbook does not answer one of the problems you are confronted with, please consult the K-Team web site ([www.k-team.com](http://www.k-team.com)) and, especially the Forum and the FAQs.

- **Installation:** instruction for installation of the software and licensing.
- **Usage:** quick description of the usage of the V-REP software

## 2. INSTALLATION

---

There are different versions of license corresponding to different versions of installations (see: <http://www.k-team.com/mobile-robotics-products/v-rep/versions>)

Depending of which one you bought, select the corresponding instruction.

### 2.1 The single fixed license

This license permits the installation of the software on only one computer.

Install from the “V-REP Vx.y.z Setup.exe” file. When you launch the program for the first time, you have to register the software. It will be then locked to your current computer as it is a single fixed license. Request and buy a license by sending an email at [info@k-team.com](mailto:info@k-team.com) (with information of the license type, your Hardware ID). Your Hardware ID (a, Fig 2.1) and a License key (for b) will be sent back.

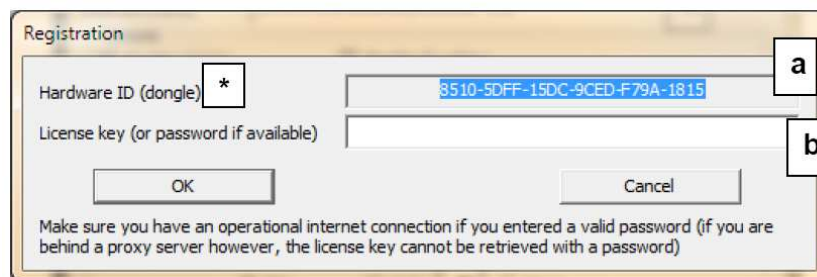


Fig 2.1 registration window

### 2.2 The single dongle license

With this license, you can use the software on multiple computers but only one at a time.

You will receive by post mail a dongle key and by email the corresponding license key to the dongle. The dongle has the shape of an USB memory key.

Install the V-REP software from the “V-REP Vx.y.z Setup.exe” file. Then insert your dongle key into your computer and launch the program. Insert the license key.

### 2.3 The floating license

This license is suited for a park of computers. You will receive the license server and the corresponding license keys to a dongle key by email, and by post mail the dongle key. The dongle has the shape of an USB memory key.

1. Insert your dongle key into your computer.
2. Start the license server “v\_repFloatingLicenseServer.exe” on the PC that should act as the floating license server (fig. 2.2).



```

Press ESCAPE to terminate program
2010/6/3 11:21:5 >
#####
#####
Logging started
V-REP floating license server built date is: May 8 2010
V-REP floating license server version: 1.1
PC hardware ID: XXXX- XXXX - XXXX - XXXX - XXXX - XXXX (dongle)
Reading settings...
Using port 20249.
Reading licenses...
1: invalid input.
2: invalid input.
3: invalid input.
4: invalid input.
5: invalid input.
6: invalid input.
ERROR: no valid licenses found.
Leaving here (press a key).

```

Fig 2.2: floating license server

3. When you received the license keys, put them in a text file named “floatingLicenses.txt”, one by line, in the same directory of the license server. Edit also the “floatingLicenseSettings.txt” file, where the port on which the server listens to can be set. All text output is automatically saved to a log file (“floatingLicenseServerLog.txt”). Make sure that Windows allows the server to listen to that port (Firewall). Then launch the server.
4. On each client PCs that should run V-REP, install V-REP “V-REP Vx.y.z Setup.exe” file then edit the “V-REP FULL\system\usrset.txt” file. Make sure following values are added and correct:
  - a. “floatingLicenseEnabled = true”
  - b. “floatingLicenseServer = xxx.xxx.xxx.xxx” (specify the IP address of the floating license server PC)
  - c. “floatingLicensePort = 20249” (make sure the port number is same as the one specified on the floating license server PC)
  - d. “floatingLicenseLight = false” (if false, that PC will search for a V-REP FULL floating license, otherwise it will search for a V-REP LIGHT floating license)
5. Save the file.
6. Start V-REP. If the floating license server cannot be contacted, or if no appropriate (or available) floating license could be found, an error message is displayed and V-REP ends. Otherwise V-REP starts and automatically uses the settings of the floating license that it received from the server (options, FULL/LIGHT, serial number). Every 4 minutes, V-REP will actualize the license (reserve it for another 8 minutes). If it failed at actualizing the license, an error message appears every 10-15 seconds.
7. When ending V-REP, V-REP will check-out the license that it borrowed, so that other PCs can use it. If V-REP crashed (or failed at checking a license out because it could not connect to the server), then its license will automatically be freed (checked-out) after 8 minutes.

## 3. Usage

### 3.1 Tutorial

After having installed and registered the software, you can start using it.

The best way is to start with the tutorials located in the “V-REP User Manual”. You can access it by the link on your desktop or from the V-REP program: “Help” menu, “Help Topics” sub-menu, into “V-REP => Contents => “V-Rep” => “Tutorials” => “BubbleRob tutorial”.

### 3.2 Loading integrated scenes

You can load created scenes distributed with the software.

In the top Menu under *File/Open Scene*, choose the file *khepera3.ttt* in the dialog box. When you press the "Start/Resume" green button, the simulation will start (Fig. 3.1).

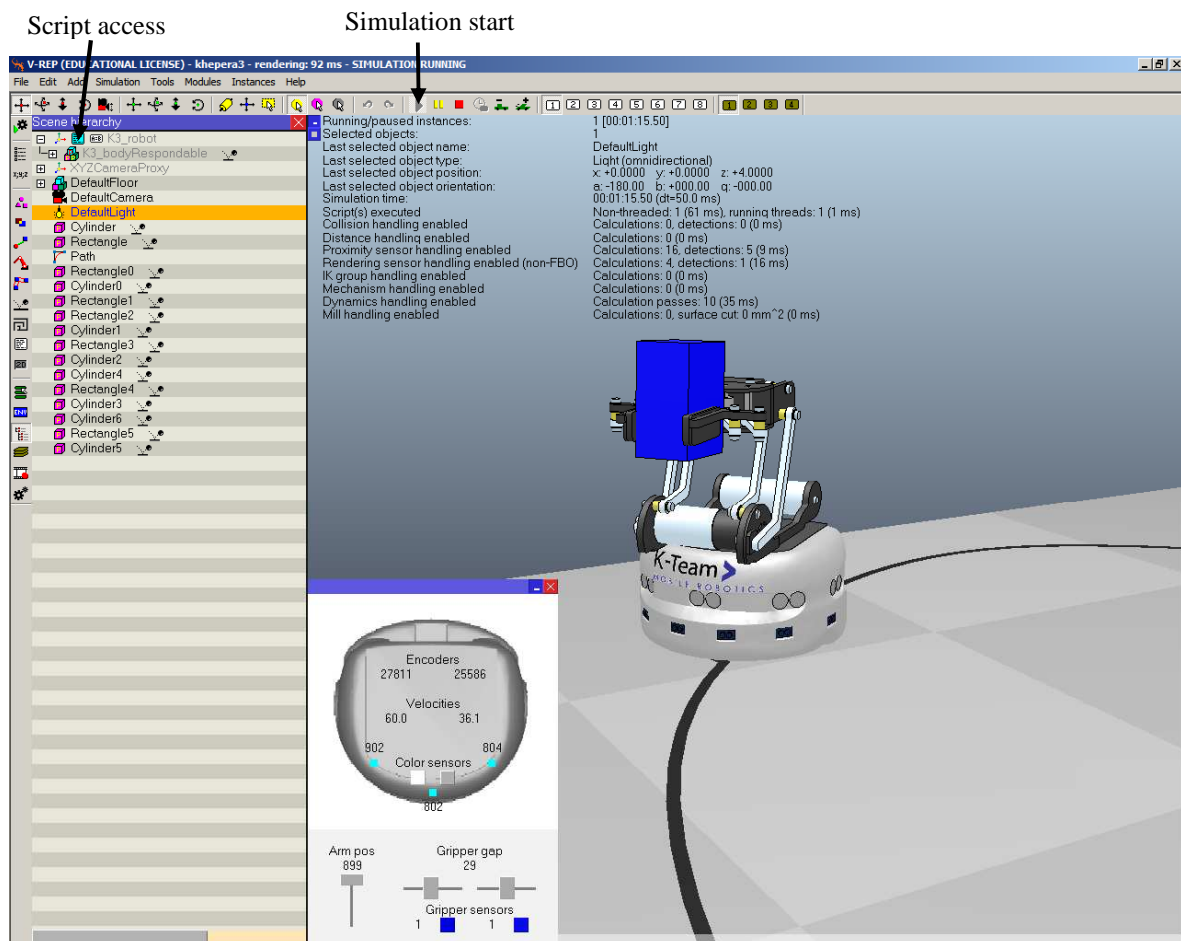


Fig. 3.1 V-REP with Khepera III and gripper

## 3.3 Programming the Khepera III robot

### 3.3.1 On V-REP simulator

The programming language used in V-REP is LUA. It is a script language. You can have more information on that language there: <http://www.lua.org/docs.html> and in the User manual in the software for specific commands to the simulator.

You can start modifying the script associated with the Khepera III simulation (fig 3.1). Basically it follows the black line and grasps an object (fig 3.2).

```
...
maxVel=2*math.pi
while (simGetSimulationState()~=sim_simulation_advancing_abouttostop) do -- while simulation running...

    if (simExtK3_getLineSensor(0)>0.75) then
        velLeft=maxVel      -- left sensor intensity > 0.75
    else
        velLeft=maxVel*0.5  -- left sensor intensity <= 0.75
    end

    if (simExtK3_getLineSensor(1)>0.75) then
        velRight=maxVel     -- right sensor intensity > 0.75
    else
        velRight=maxVel*0.5 -- right sensor intensity <= 0.75
    end

    simExtK3_setVelocity(velLeft,velRight) -- Set desired left and right motor velocities

    if (simExtK3_getGripperProxSensor(0)<0.015)or(simExtK3_getGripperProxSensor(1)<0.015) then
        -- Something was detected between the fingers
        simExtK3_setGripperGap(0) -- close the fingers
        simExtK3_setArmPosition(900) -- move the arm up
    end

end
...
```

Fig. 3.2 V-REP Khepera III main loop

#### Implemented LUA commands for the Khepera 3:

The following LUA script commands and their parameters for the Khepera 3 robot and its gripper are described below:

***distance = simExtK3\_getInfrared(index)***

return the distance measure in [m] of the infrared sensor index [0..8]

***distance\_inMeters=simExtK3\_getUltrasonic(index)***

return the distance measure in [m] of the ultrasonic sensor index [0..4]

***intensity=simExtK3\_getLineSensor(index)***

return the intensity [0 to 1] of the ground infrared sensor index [0 or 1]

***distance =simExtK3\_getGripperProxSensor(index)***

return the distance in [m] of the gripper sensor index [0 or 1]

- simExtK3\_setVelocity(velocity\_Left , velocity\_Right)*  
 set the velocities of the left and right wheels in [rad/s]
- simExtK3\_setArmPosition(position)*  
 set position of the gripper's arm; range [300 (down) .. 900 (up)]
- simExtK3\_setGripperGap(gap)*  
 set aperture position of the gripper; range [0 (closed) .. 170 (open)]

### 3.3.2 Applying Lua script on the real Khepera 3 robot

With the Lua parser compiled for the robot, you will be able to run on the Khepera 3 robot a script developed on V-REP.

You will need the following hardware and software:

- Hardware:
- a Khepera 3 robot mounted with its Korebot II Linux board, optionally the Khepera 3 gripper
  - KoreConnect converter and serial cable
  - a PC with serial connection or with an USB to serial converter
- Software:
- a terminal (Hyperterminal or TeraTerm) for communicating with the robot through the serial port
  - file *kh3luaparser*, the Lua parser for Khepera 3: available from: [http://ftp.k-team.com/KheperaIII/V-REP\\_lua/binaries/](http://ftp.k-team.com/KheperaIII/V-REP_lua/binaries/)
  - file *libstdc++6\_4.1.2-r10\_armv5te.ipk*, the C++ library for Korebot II: available from [http://ftp.k-team.com/KheperaIII/V-REP\\_lua/binaries/](http://ftp.k-team.com/KheperaIII/V-REP_lua/binaries/)
  - file *liblua-5.1.so*, the Lua library: available from [http://ftp.k-team.com/KheperaIII/V-REP\\_lua/binaries/](http://ftp.k-team.com/KheperaIII/V-REP_lua/binaries/)
  - source files of the parser [http://ftp.k-team.com/KheperaIII/V-REP\\_lua/sources/kh3LuaParser\\_VERSION-DATE.tar.bz2](http://ftp.k-team.com/KheperaIII/V-REP_lua/sources/kh3LuaParser_VERSION-DATE.tar.bz2) where \_VERSION-DATE is the release version.
  - Lua examples: [http://ftp.k-team.com/KheperaIII/V-REP\\_lua/lua\\_examples/](http://ftp.k-team.com/KheperaIII/V-REP_lua/lua_examples/)

Follow the instructions below for the Installation and Usage:

#### Installation:

- connect the Khepera 3 robot and open the terminal (see Khepera 3 user manual, chapter “5.2 Testing the secondary serial link”: <http://ftp.k-team.com/KheperaIII/UserManual/> )
- transfer the file *libstdc++6\_4.1.2-r10\_armv5te.ipk* to the robot and install it with the command:  

```
ipkg install libstdc++6_4.1.2-r10_armv5te.ipk
```
- transfer the file *liblua-5.1.so* and move it to the directory */usr/lib*:  

```
mv liblua-5.1.so /usr/lib
```

- transfer the file *kh3luaparser* to the robot
- change to executable mode the parser with the command:  
***chmod +x kh3luaparser***
- copy the Lua script from V-REP Khepera 3 robot (fig. 3.2) to a file named *script.lua* and transfer this file to the robot

Usage:

- run the parser with the Lua script file as argument (exit with keys CTRL-C):  
***./kh3luaparser script.lua***
- you can modify the script with the *vi* editor, directly on the robot  
***vi script.lua***

The *vi* commands are:

Key *x*: delete the letter under the cursor

Key *a*: enter in edit (append) mode

Key *i*: enter in edit (insert) mode

***Escape*** key then **:** : exit mode: write ***wq!*** for saving and exiting  
write ***q!*** for exiting without saving

If you have several consoles connected to the robot via network (see Korebot II user manual), you can launch the script into one console; and edit it in another one. The parser program is checking if the script file is modified and saved. It will reload and launch the newer if it is the case.

Look at the Lua script examples in the subfolder *lua\_examples* in the source files of the parser or at [http://ftp.k-team.com/KheperaIII/V-REP\\_lua/lua\\_examples/](http://ftp.k-team.com/KheperaIII/V-REP_lua/lua_examples/)

Implemented commands for the Lua parser running on the Khepera3 robot:

The common Lua commands can be used. The implemented LUA commands for the Khepera 3 are described in the chapter 3.3.1 above. And these following specific commands of the simulator can be used too:

- ***simWait(time in [s])*** : wait a time in [s], floating point is allowed.
- ***simGetSimulationTime()***: return the time in [s] since the simulation started.
- ***simStopSimulation()***: stop the simulation.







K-Team S.A.  
Z.I. PLANS-PRAZ  
1337 VALLORBE  
SWITZERLAND

---