

The RRC Project Manual de Usuario

Arakyd Software

25 de mayo de 2014

Índice

1. Introducción	4
2. Instalación y funcionamiento en Ubuntu	4
2.1. Instalación QT 5.2.1	4
2.2. Configuración del PATH	5
2.3. Preparar Makefile	5
2.4. Compilar y linkear	5
2.5. Ejecutar	6
2.6. Futuras configuraciones	6
2.6.1. Uso avanzado	6
3. Instalación en Windows	7
3.1. Instalar Visual Studio 2012	7
3.2. Descargar e instalar Qt 5.2.1.	7
3.3. Configuración del entorno	8
3.3.1. Configurar archivo Makefile	8
3.3.2. Configurar el PATH	11
3.4. Compilar y linkear	11
3.5. Ejecutar	12
4. Instalación de V-Rep	13
4.1. Descarga de V-Rep	13
4.2. Instalación en Windows 7.1	13
4.3. Instalación en OSX - Mavericks 10.9.2	17
4.4. Instalación en Ubuntu 12.04 LTS x64/x32	18
4.4.1. Instalación gráfica	18
4.4.2. Instalación mediante consola	19
5. Manual de la GUI	21
5.1. Modos: Demo y Control	21
5.1.1. Demo	21
5.1.2. Control	21
5.1.3. Log	22
5.2. Parte superior	22
5.2.1. Text Inputs: IP y Puerto	22
5.2.2. Botón: Iniciar Sim.	23
5.2.3. Botones: Pausar y Parar Sim.	23
5.2.4. Botón: Ejecutar	23
5.3. Parte inferior izquierda	23
5.3.1. Botones Desplegables	24
5.3.2. Botones: Funciones	24
6. Parte inferior derecha	24

7. Código del Usuario	25
7.1. Control.cpp por defecto	25
7.2. KheperaSimGui.pro	27
7.3. Demo	27

1. Introducción

El objetivo del proyecto es un sistema distribuido para la implementación y prueba de algoritmos de control sobre instancias físicas y simuladas del robot Khepera III. En el Host Controlador se ejecutará un conjunto de programas que permitirá al usuario ejecutar sus algoritmos de manera remota en los Khepera físicos o el Host Simulador, que simula las instancias virtuales del robot. El Software y entorno de simulación empleado será V-REP.

Este documento pretende guiar al usuario en la instalación y uso del software Khepera-SimGui en diferentes plataformas.

Aspectos que se cubren:

- Instalación y configuración en SO Linux.
- Instalación y configuración en SO Windows.
- Puesta en marcha del entorno de simulación V-Rep.
- Creación del código del usuario.
- Manual de la Interfaz gráfica de usuario.

2. Instalación y funcionamiento en Ubuntu

El presente capítulo cubre los pasos a seguir para la configuración del software en una máquina con sistema operativo Ubuntu.



2.1. Instalación QT 5.2.1

Será necesario instalar la biblioteca Qt 5.2.1 y se hará uso de su sistema “qmake” para generar el archivo makefile. Para ello escribimos en el terminal:

```
$ sudo apt-get install qt-sdk
```



2.2. Configuración del PATH

Añadiremos qmake al PATH:

Se hará export de la carpeta /bin en el directorio de instalación de Qt 5.2.1, si se ha dejado por defecto escribiremos en terminal:

```
$ export PATH=$HOME/Qt5.2.1/5.2.1/gcc/bin:$PATH
```

Nota: el nombre /gcc del directorio anterior cambiará según la versión sea de 32 ó 64 bits.

El export presenta el inconveniente de que solamente se podrá invocar qmake en el terminal donde lo hemos hecho, sin embargo, una vez realizado queda configurado el archivo Makefile que se utilizará en las siguientes modificaciones.

2.3. Preparar Makefile

Una vez configurado el path, Nos moveremos al directorio del programa en el terminal y llamaremos la función qmake.

```
$ qmake
```

Se creará el Makefile.

Problemas conocidos: En una instalación limpia de Ubuntu faltará la librería IGL, que debería estar incluida, esto es un problema de Ubuntu que podemos resolver instalando dicha librería, una forma de hacerlo sería escribir en el terminal:

```
$ sudo apt-get install libgl1-mesa-dev
```

2.4. Compilar y linkear

Debemos estar en el directorio del programa en el terminal.

```
$ make
```

Se creará el archivo ejecutable KheperaSimGui.

Nota: Previamente, el usuario deberá haber preparado su programa (Control.cpp). Ver sección 7.

2.5. Ejecutar

Debemos estar en el directorio del programa en el terminal.

```
$ ./KheperaSimGui
```

Ejecutamos el fichero de la forma anterior en terminal ó haciendo doble click en la carpeta del software, ésto nos mostrará la interfaz gráfica de usuario.

Para conocer en detalle las funciones de la interfaz gráfica de usuario (ó GUI), ver sección 5.

2.6. Futuras configuraciones

Una vez generado el archivo makefile haciendo uso de la biblioteca Qt, esto es, realizados los puntos 2.1, 2.2 y 2.3, tras modificar el código fuente (Código de Usuario) sólo será necesario ejecutar make y abrir el ejecutable generado, KheperaSimGui; es decir, seguir los apartados 2.4 y 2.5.

2.6.1. Uso avanzado

Si el usuario desea manipular algunos items aparte de “Control.cpp”, podría ser necesario tener que ejecutar qmake para generar un nuevo Makefile (ver apartado 2.3), si ésto se va a dar más de una vez sería conveniente hacer un enlace en usr/bin a qmake, para no tener que hacer el export cada vez que desee emplear qmake:

```
$ sudo ln -s RUTAQMAKE/qmake /usr/bin/qmake
```

Donde en el caso de instalación por defecto de QT:

```
$ sudo ln -s $HOME/Qt5.2.1/5.2.1/gcc/bin/qmake /usr/bin/qmake
```

Para la modificación del código de control de usuario ver sección 7.

3. Instalación en Windows

En esta guía se cubre, paso por paso, la instalación y una primera ejecución de KheperaSim GUI en Windows 8.

3.1. Instalar Visual Studio 2012

Para llevar a cabo la instalación y ejecución de KheperaSim GUI, debemos tener instalado Visual Studio 2012 o superior en nuestra computadora. La instalación de Visual Studio no se explica en este manual.

3.2. Descargar e instalar Qt 5.2.1.

Qt es una biblioteca multiplataforma para desarrollo de diferentes aplicaciones con interfaz gráfica de usuario, así como también para el desarrollo de programas sin interfaz gráfica, como herramientas para la línea de comandos y consolas para servidores. Ha sido desarrollada como un software libre y de código abierto por Qt Project (<http://qt-project.org>).

La descarga se puede realizar desde el siguiente enlace, como se aprecia en la figura 1:

http://download.qt-project.org/official_releases/qt/5.2/5.2.1/single/qt-everywhere-opensource-src-5.2.1.zip

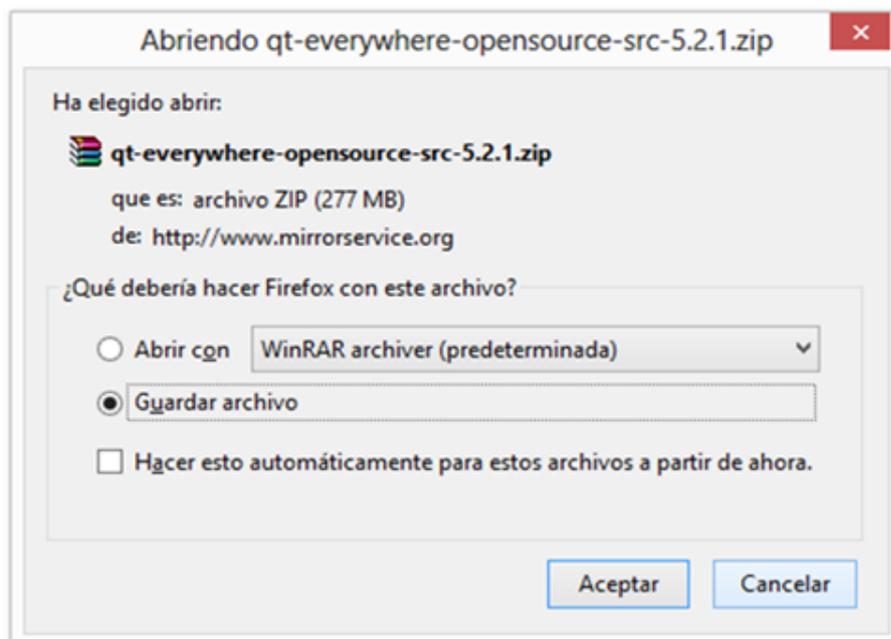


Figura 1: Descarga Qt 5.2.1.

Una vez completada la descarga de Qt 5.2.1, debemos descomprimir el archivo zip en nuestra computadora, teniendo en cuenta el directorio en el que se hace.

3.3. Configuración del entorno

3.3.1. Configurar archivo Makefile

Para llevar a cabo el paso 3.3.1., ofrecemos 3 opciones diferentes.

OPCIÓN 1:

- Abrir ventana de comandos de Visual Studio 2012, el cual debe haber sido instalado previamente, según la sección 3.1.
- Cambiar directorio de trabajo al directorio donde previamente se ha descomprimido la carpeta Qt 5.2.1., sección 3.2 de ese manual.

Nota: Para movernos por el terminal, utilizaremos los siguientes comandos, como se puede observar en la figura 2:

'cd ..': cambiar directorio actual por el inmediatamente superior.

'dir': listado de directorios dependientes del directorio actual.

'cd *directorio*': cambiar directorio actual por *directorio*, que debe ser dependiente del directorio actual.

```

C:\Program Files (x86)\Microsoft Visual Studio 11.0>cd ..
C:\Program Files (x86)>cd ..
C:\>dir
El volumen de la unidad C es OS
El número de serie del volumen es: 681A-AB90

Directorio de C:\
24/02/2014  20:51  <DIR>          asusUibeData
28/04/2014  22:55  <DIR>          Equipo Programación
24/02/2014  20:49  <DIR>          eSupport
29/04/2014  00:57  <DIR>          Program Files
03/05/2014  17:30  <DIR>          Program Files (x86)
28/04/2014  21:40  <DIR>          qt-everywhere-opensource-src-5.2.1
29/04/2014  19:53  <DIR>          sources
16/04/2014  19:02  <DIR>          Users
29/04/2014  19:53  <DIR>          Windows
                0 archivos          0 bytes
                9 dirs 130.718.040.256 bytes libres
C:\>cd qt-everywhere-opensource-src-5.2.1

```

Figura 2: Términoal de Visual Studio 2012.

- Una vez situados en el directorio en el que hemos descomprimido Qt 5.2.1., ejecutar los siguiente comandos:

```
configure -platform win32-msvc2012
```

```
nmake (Puede tardar varias horas.)
```

(Para reducir el tiempo puede usar jom: <http://qt-project.org/wiki/jom>)

OPCIÓN 2:

- Configurar el PATH para incluir la carpeta `/VC/bin/` de su instalación de Visual Studio 2012:

PATH es la variable del sistema que utiliza el sistema operativo para buscar los ejecutables necesarios desde la línea de comandos o la ventana Terminal.

La variable del sistema PATH se puede establecer utilizando la utilidad de sistema en el panel de control de Windows o en el archivo de inicio del shell en Linux y Solaris.

Configuración PATH (Windows 8):

Según se indica en la la figura 3:

Arrastre el cursor a la esquina inferior derecha de la pantalla y acceda a la pestaña de configuración.

Vaya al Panel de control —Sistema —Configuración avanzada del sistema.

Una vez en la ventana de Configuración avanzada del sistema, haga clic en variables de entorno.

En la ventana de variables de entorno, pinche en Variables del sistema, busque PATH y haga clic en él.

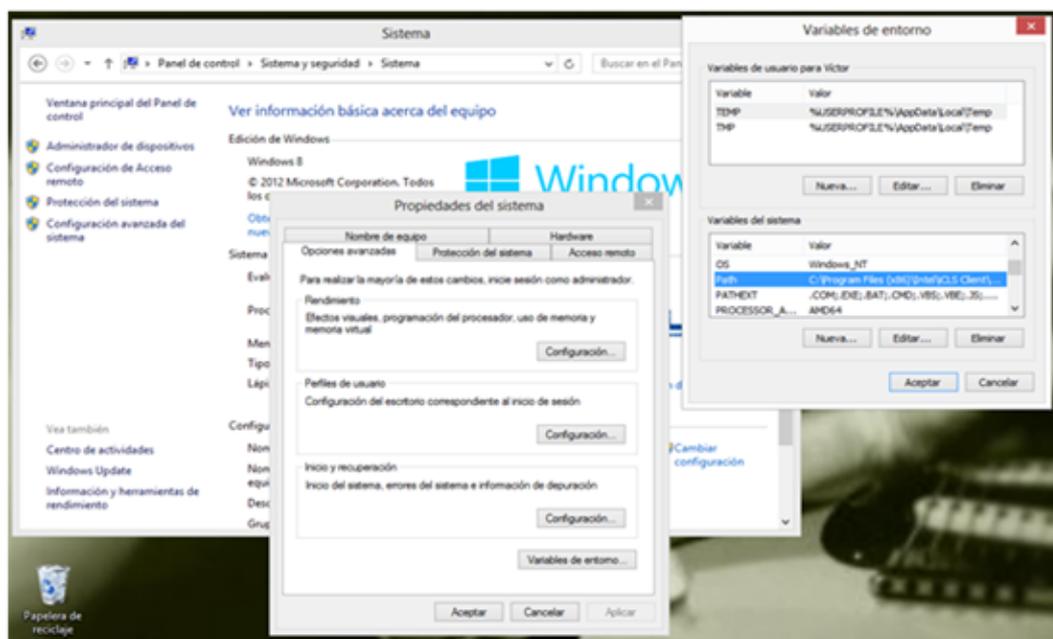


Figura 3: Acceder a la configuración de la variable PATH.

En la ventana Editar, modifique PATH agregando la ubicación de la clase. Si no dispone del elemento PATH, puede optar por agregar una nueva variable y agregar PATH como el nombre y la ubicación de la clase como valor.

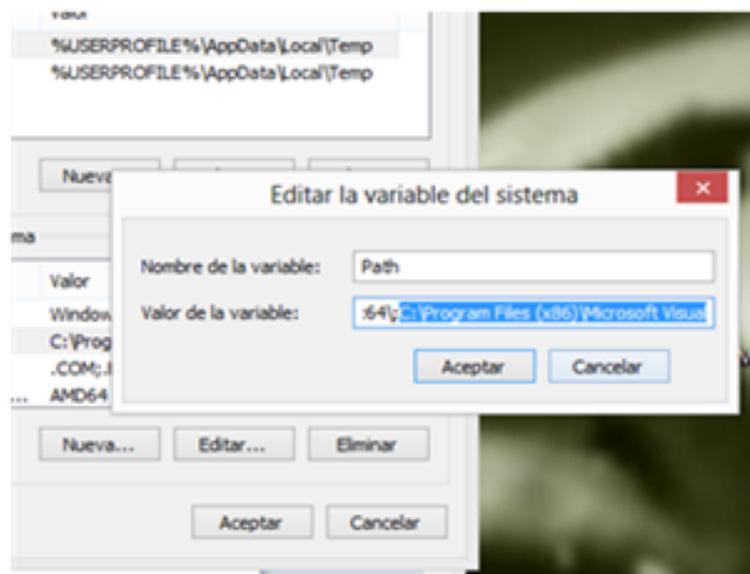


Figura 4: Edición de la variable PATH.

Finalmente, incluya la ruta completa en la variable PATH, *C:/Program Files (x86)/Microsoft Visual Studio 11.0/VC/bin* en nuestro caso, como se muestra en la figura 4.

Una vez configurado el PATH:

- Entrar en carpeta de código fuente de Qt con una ventana de comandos (*cmd.exe*).
- Ejecutar los siguientes comandos:

```
configure -platform win32-msvc2012
```

```
nmake (Puede tardar varias horas.)
```

(Para reducir el tiempo puede usar jom: <http://qt-project.org/wiki/jom>)

OPCIÓN 3:

- Abrir ventana de comandos (*cmd.exe* figura 5) y ejecutar *vsvars32.bat* (localización: *Common7/Tools/* de su instalación de VS2012).

Nota: Se puede copiar a otra carpeta por comodidad. Debe volver a ejecutarse cada vez que se ejecuta *cmd.exe* para que funcione.

- Situarse en el directorio donde descomprimos la carpeta de código fuente de Qt. Utilizar los comandos según se explicó en OPCIÓN 1, sección 3.3.1.
- Ejecutar (sin comillas):

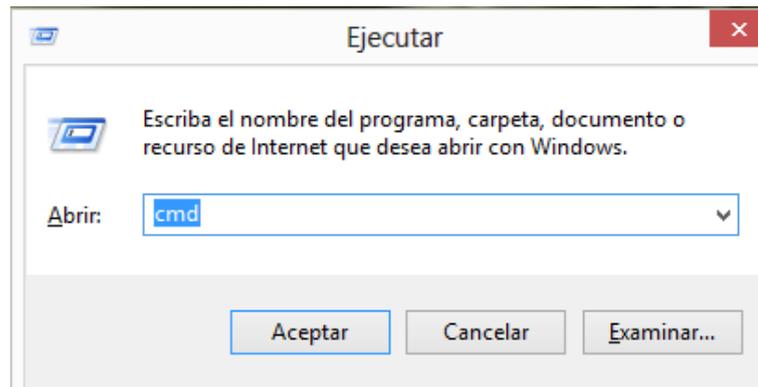


Figura 5: Ejecutar *cmd.exe*.

```
configure -platform win32-msvc2012
```

nmake (Puede tardar varias horas.)

(Para reducir el tiempo puede usar jom: <http://qt-project.org/wiki/jom>)

3.3.2. Configurar el PATH

- Configurar el PATH para incluir la carpeta */qtbase/bin/* de su instalación de Qt.

Nota: En la OPCIÓN 2 del apartado 3.3.1 se explica como configurar el PATH.

Si no se quiere configurar el PATH, copiar el ejecutable *qmake.exe* en la carpeta de código fuente *KheperaSimGui_codigo_fuente*.

3.4. Compilar y linkear

Nota: Hasta aquí la instalación de la aplicación.

Los siguientes pasos, hasta la ejecución de *KheperaSimGui.exe*, se han de realizar solo si se hace algún cambio en el código de usuario *Control.cpp* 7.

- Abra una ventana de comandos.
- Situese en la carpeta de código fuente *KheperaSimGui_codigo_fuente*.
- Ejecutar los siguiente comandos:

```
qmake
```

```
nmake -f MakeFile.Release
```

3.5. Ejecutar

- Ejecutar *KheperaSimGui.exe* ubicado según la ruta:
KheperaSimGUI_codigofuente_win_release/release/KheperaSimGui.exe

En la figura 6 se muestra la interfaz gráfica de usuario que se abre al ejecutar *KheperaSimGUI.exe*

IMPORTANTE: Antes de ejecutar *KheperaSimGUI.exe*, abrir V-REP en el host simulador y cargar cualquier escena con el modelo *kh3_noplugin*, el modelo *khepera* incluido de serie en V-Rep NO FUNCIONA SIN MODIFICARLO.

(ejemplo de modelo V-REP *demo2khepera.ttt* incluida con la aplicación)

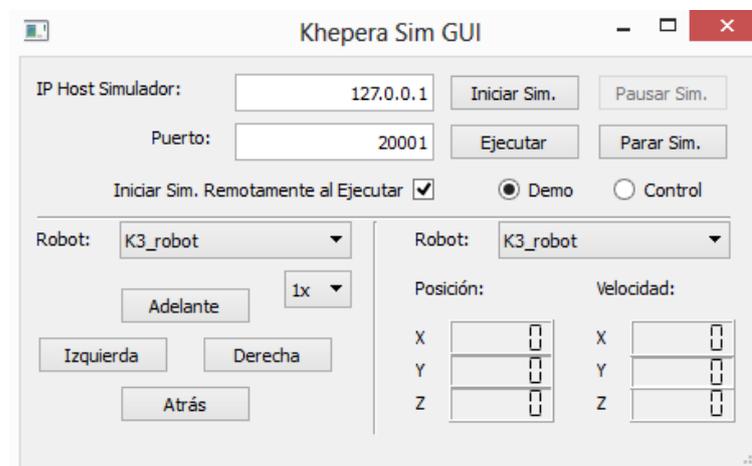


Figura 6: Interfaz gráfica de usuario.

Para conocer en detalle las funciones de la interfaz gráfica de usuario (GUI), ver sección 5.

Nota: Si no se configuró el PATH en la sección 3.3.1 OPCIÓN 2, deberán incluirse una copia de los siguientes archivos de la instalación de Qt 5.2.1. en la carpeta *KheperaSimGUI_codigofuente_win_release/release*:

Qt5Core.dll

Qt5Gui.dll

Qt5Widgets.dll

libEGL.dll

libGLESw2.dll

platforms/qwindows.dll

msvcpr120.dll (sólo si se lleva a un ordenador sin VS2012, ubicado en la carpeta de instalación de VS2012)

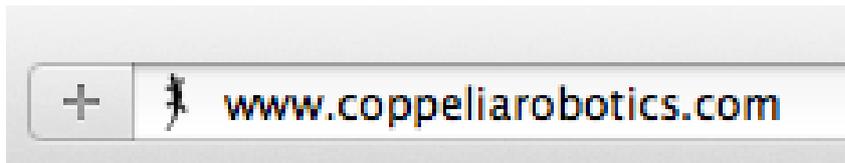
msvcr120.dll (sólo si se lleva a un ordenador sin VS2012, ubicado en la carpeta de instalación de VS2012)

4. Instalación de V-Rep

Este documento servirá de apoyo para la descarga e instalación del programa de simulación V-Rep, necesario para la ejecución del sistema, por parte del usuario final.

4.1. Descarga de V-Rep

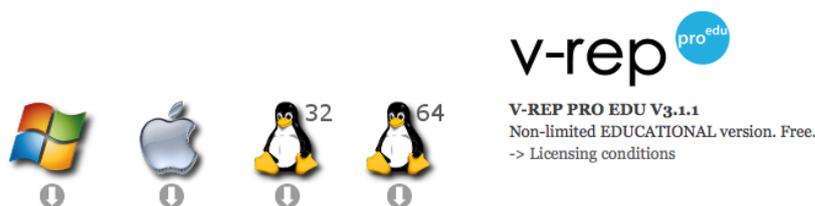
1.- Acceder a la página web del desarrollado.



2.- Ir a la sección de descargas (*Downloads*)

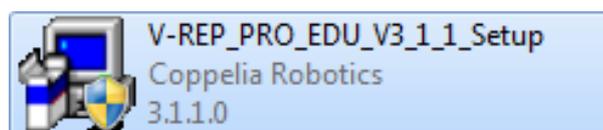


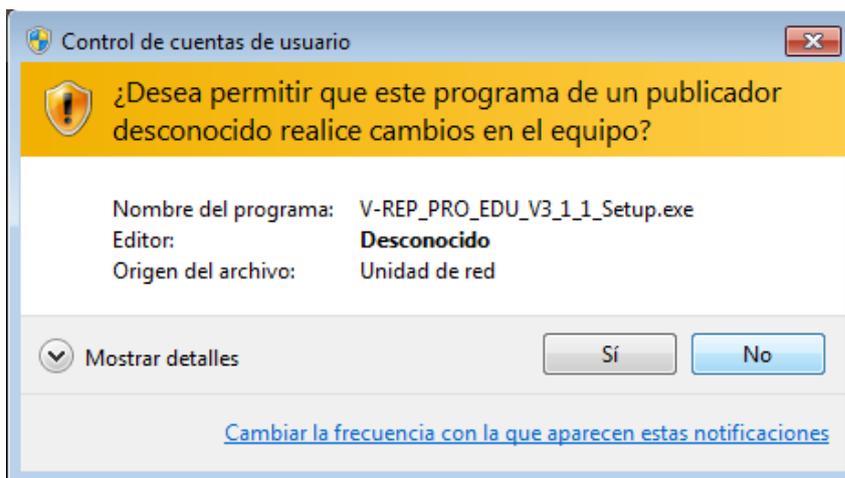
3.- Descargaremos la versión que se ajuste a nuestras necesidades. En este caso: V-rep *pro^{edu}*.



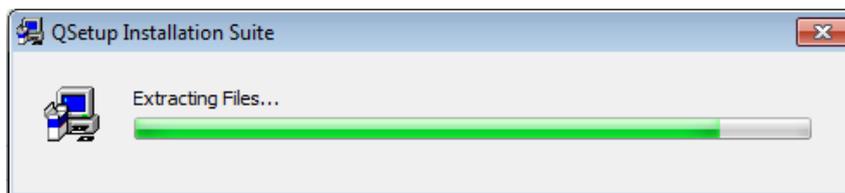
4.2. Instalación en Windows 7.1

1.- Una vez que hallamos descargado el instalador, debemos hacer doble clic sobre él y permite que realice cambios en el ordenador.





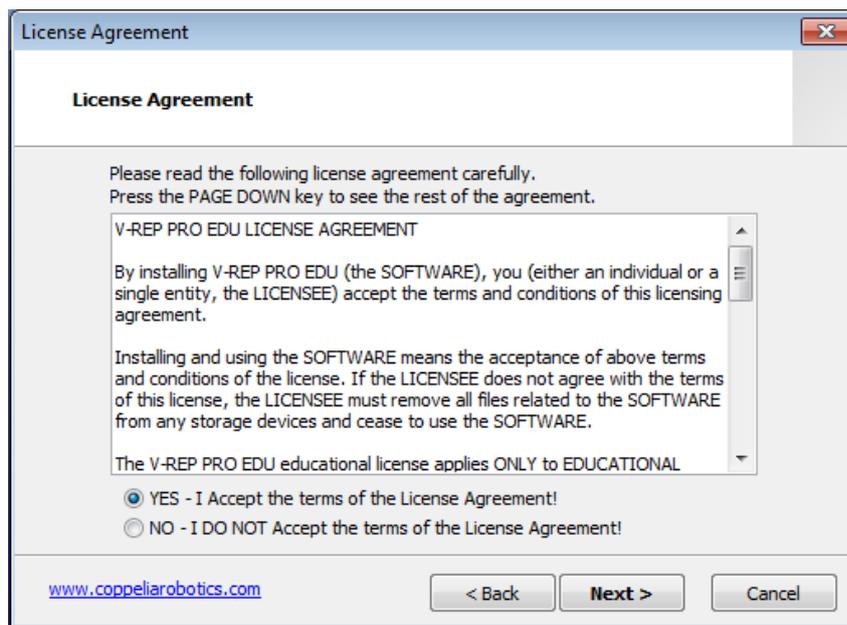
2.- A continuación se extraerán los archivos necesarios para continuar con la instalación.



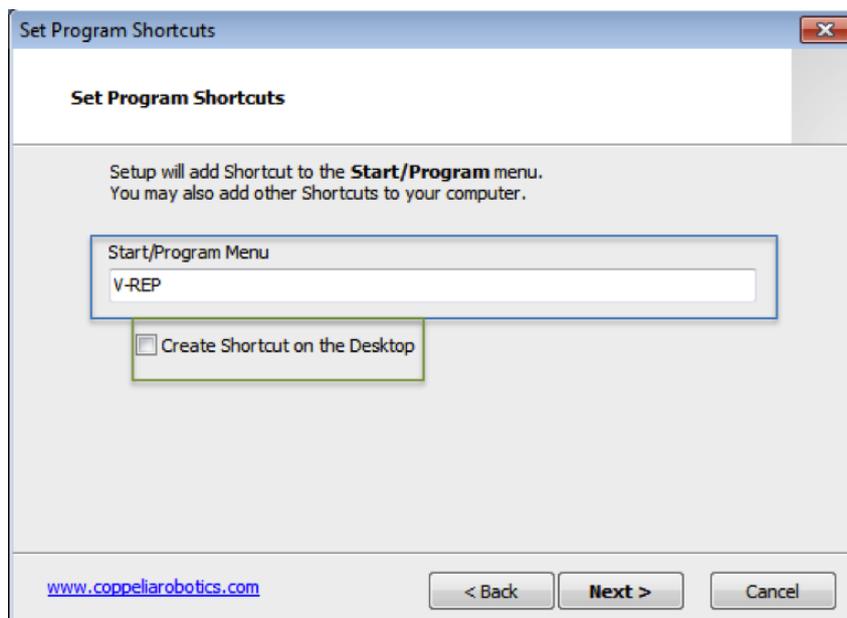
3.- En la primera ventana que aparece nos explica como se realizará la instalación.



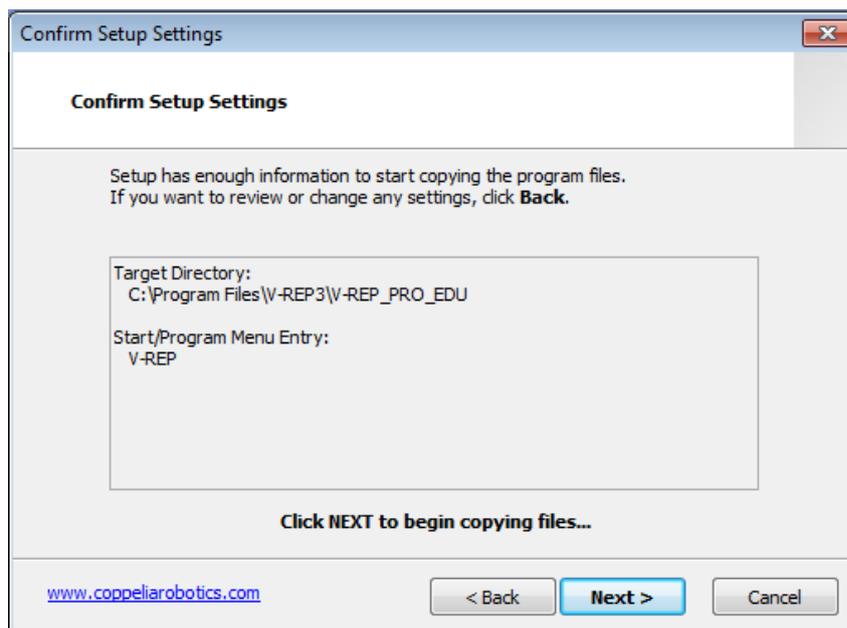
4.- La siguiente ventana es la licencia de uso, se debe leer y aceptar para poder continuar.



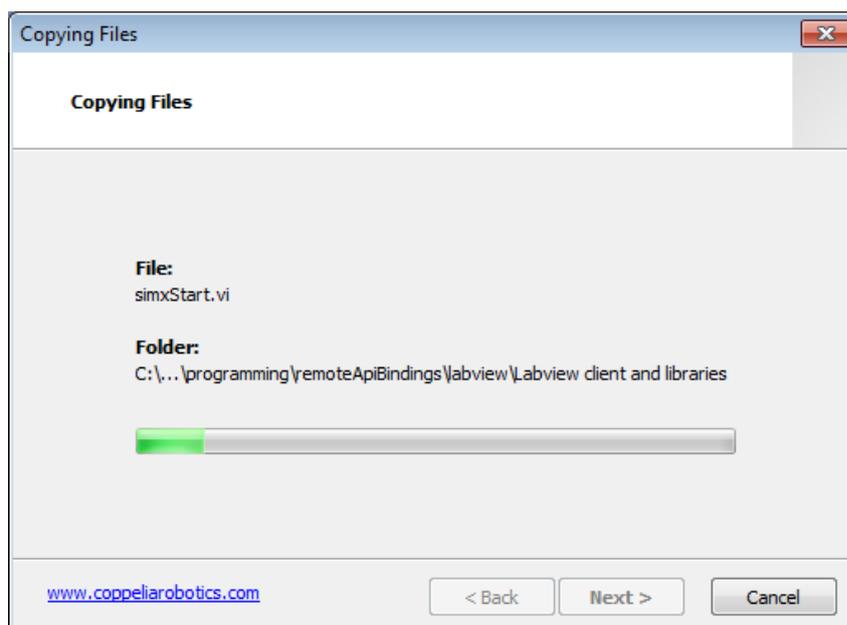
5.- Una vez aceptados las condiciones nos pedirá como queremos que sea el **directorio en el menú inicio** y si queremos un **icono en el escritorio**.



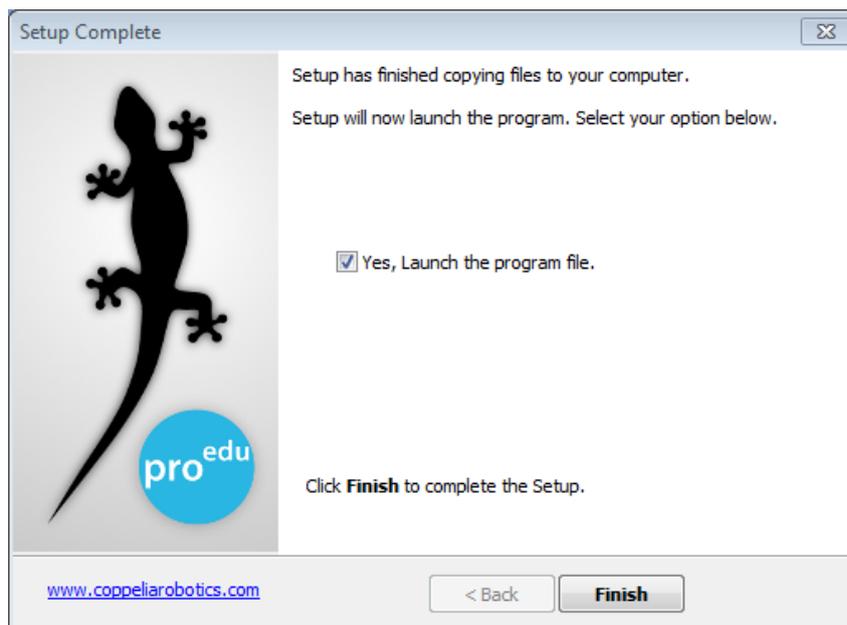
6.- El último paso es confirmar las opciones que hemos elegido.



7.- A continuación comienza la instalación.



8.- Una vez terminada la instalación nos pregunta si deseamos iniciar el programa.

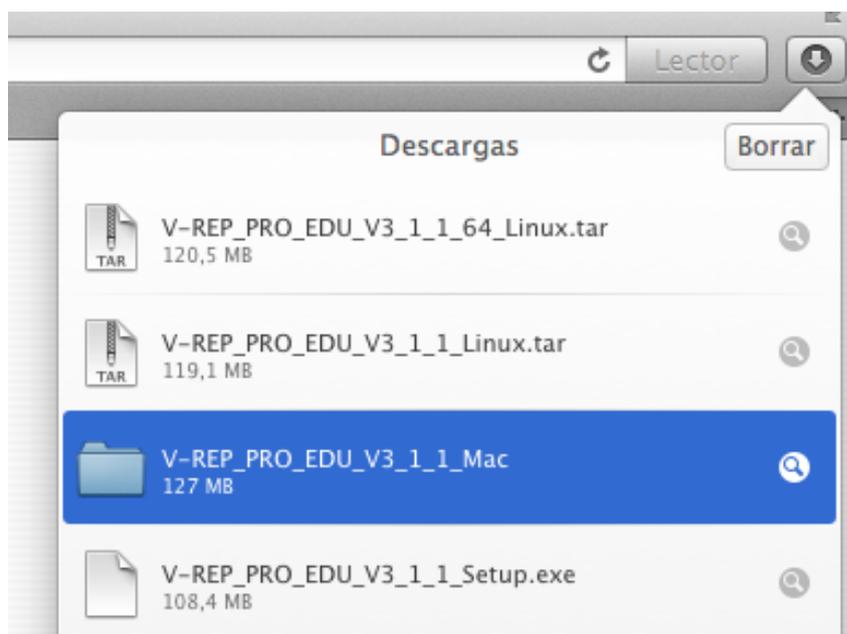


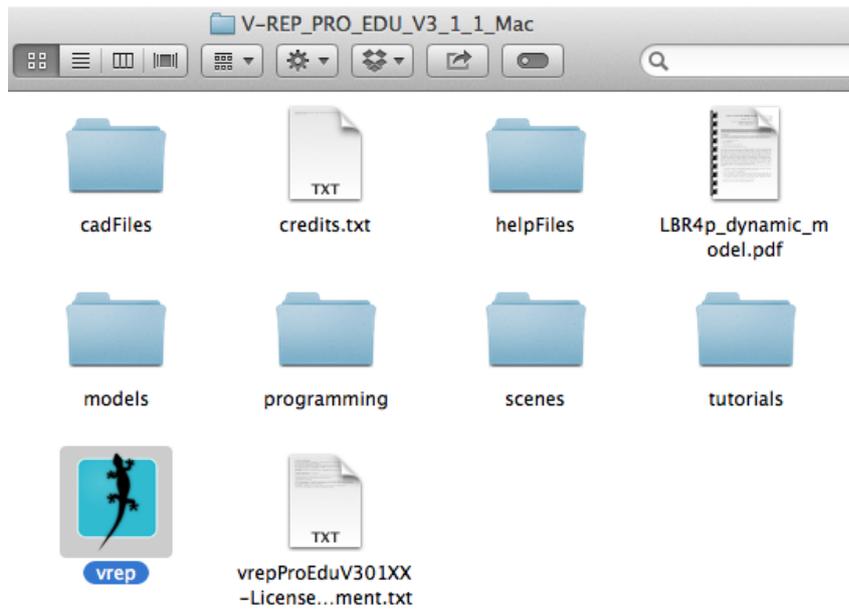
4.3. Instalación en OSX - Mavericks 10.9.2

Para simplificar el proceso de instalación en *OSX* se recomienda descargar el archivo con el navegador *Safari*.

De esta manera, una vez que se termine la descarga, el sistema descomprimirá directamente la carpeta.

Una vez que hemos accedido a la carpeta solo es necesario hacer doble clic sobre el icono de *V-Rep*.





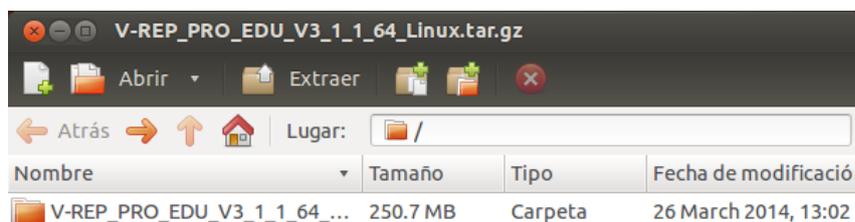
4.4. Instalación en Ubuntu 12.04 LTS x64/x32

4.4.1. Instalación gráfica

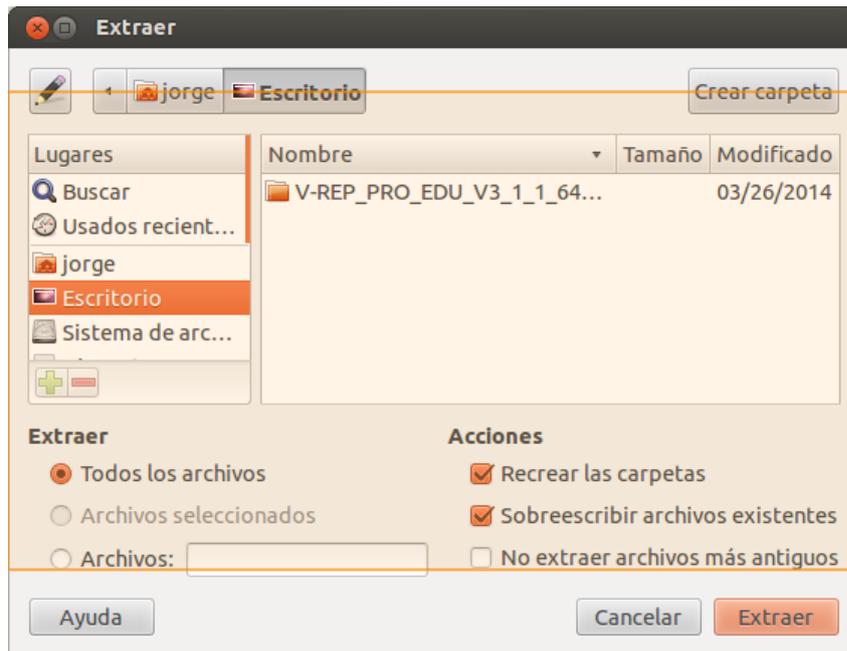
1.- Una vez que hemos descargado el correspondiente archivo .tar hacemos doble clic sobre él para abrirlo con el gestor.



2- A continuación pulsamos sobre el botón *Extraer?*.



3.- Seleccionamos el directorio de extracción y esperamos a que finalice la extracción.

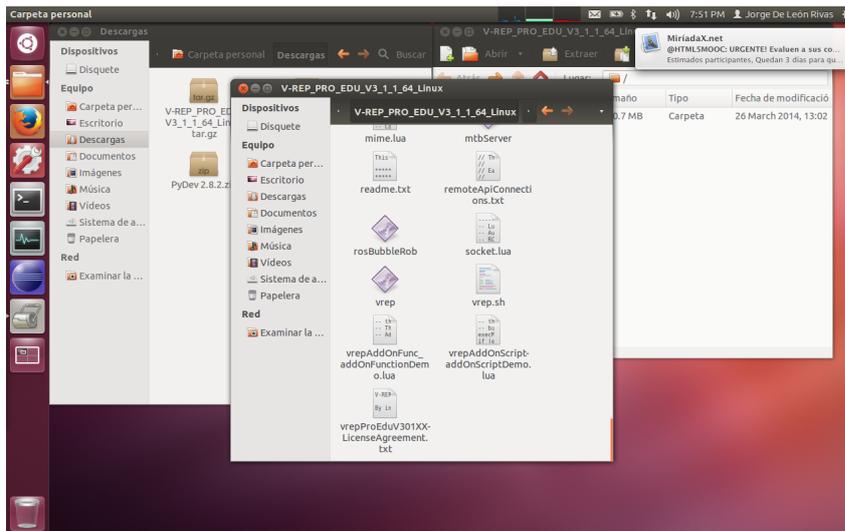


4.- Por último hacemos clic sobre *Mostrar los archivos?* y buscamos el archivo que se llama *V-rep*.

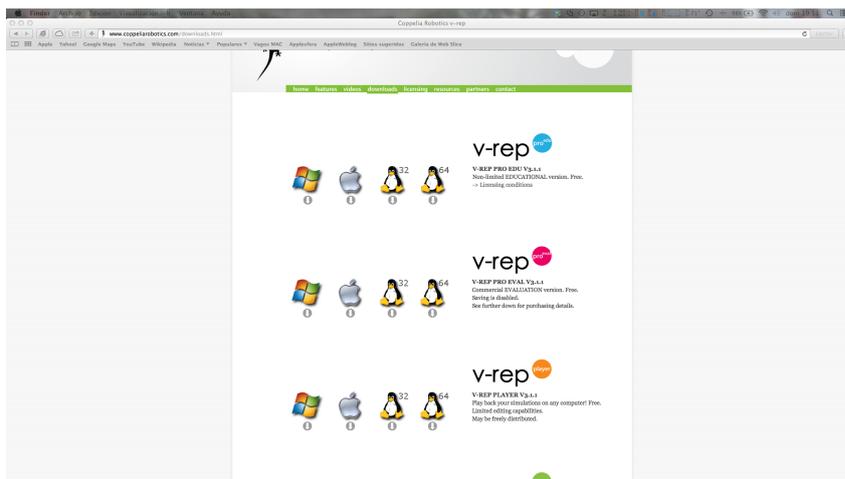


4.4.2. Instalación mediante consola

- 1.- Una vez que tenemos el archivo *.tar.gz* en el directorio que deseamos, abrimos la *Terminal*.
- 2.- Lo segundo es navegar hasta el directorio en cuestión. Por noma general, si no lo hemos movido estará en la carpeta *"Descargas"*.
- 3.- A continuación ejecutamos el siguiente comando: `$ tar zxvf V-REP_PRO_EDU_V3_1_1_64_Linux.tar.gz`. Con este comando se realiza la extracción de los ficheros en el directorio en el que nos encontramos.



4.- Por último solo debemos entrar en el nuevo sub-directorio y ejecutar `$./vrep.sh`. De esta forma ejecutaremos el programa.



5. Manual de la GUI

Esta sección muestra información sobre la interfaz gráfica de usuario del software KheperaSimGui. Podemos ver el aspecto de la misma en la figura 7.



Figura 7: Interfaz Gráfica de Usuario

5.1. Modos: Demo y Control

La GUI permite elegir entre dos modos de operación mediante los radio buttons **Demo** y **Control**.



Figura 8: Radiobuttons para selección de modo

5.1.1. Demo

El equipo de programación ha incluido el modo de funcionamiento **Demo**, que configura 5 de los botones de la interfaz gráfica con 5 funciones ya programadas que nos permiten mover los Kheperas en el entorno de simulación. Para más detalle ver apartado 7.3

5.1.2. Control

Los 5 botones quedarían configurados con las funciones que el usuario haya programado en la clase Control (Ver sección 7).



Figura 9: Botones de funciones

5.1.3. Log

Se ha implementado un historial de mensajes del sistema que podemos sacar pulsando en la barra de menú sobre Opciones → Mostrar Output.

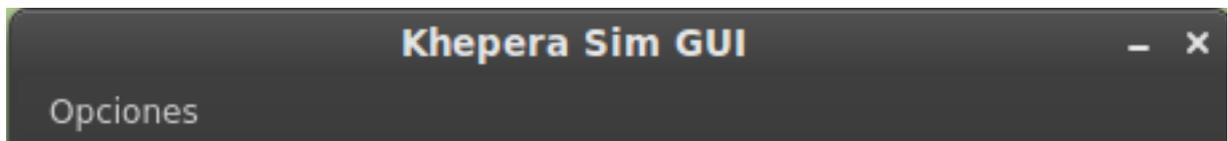


Figura 10: Barra de Menú

5.2. Parte superior

5.2.1. Text Inputs: IP y Puerto

Tenemos dos text inputs para introducir la dirección IP y el puerto del host simulador, servidor en el que tenemos el simulador Vrep en funcionamiento.

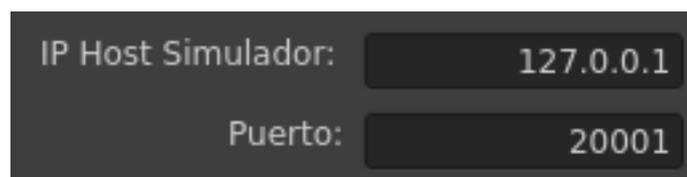


Figura 11: Text Inputs

5.2.2. Botón: Iniciar Sim.

Establece la conexión con Vrep en la máquina configurada en las text boxes e inicia la simulación de forma remota. Si se quiere modificar el puerto es obligatorio cambiarlo también en el script lua asociado al modelo `kh3_noplugin.ttm`, que es el único modelo válido para su uso con KheperaSimGui.



Figura 12: Botones Zona Superior

5.2.3. Botones: Pausar y Parar Sim.

Se puede pausar o detener la simulación con estos botones.

5.2.4. Botón: Ejecutar

Es el primero de los 5 botones de funciones mencionados anteriormente.

En el caso del modo **Demo**, dos robots comenzarán a moverse hacia delante, será necesario cargar en Vrep el escenario `demo2khepera.ttt`, alternativamente, podemos cargar 2 robots (modelo `kh3_noplugin.ttm`) con los nombres `K3_robot` y `K3_robot#0`, nombre por defecto de los dos primeros robots.

En el modo **Control** este botón ejecutará el código programado por el usuario. Ver sección 7.

5.3. Parte inferior izquierda

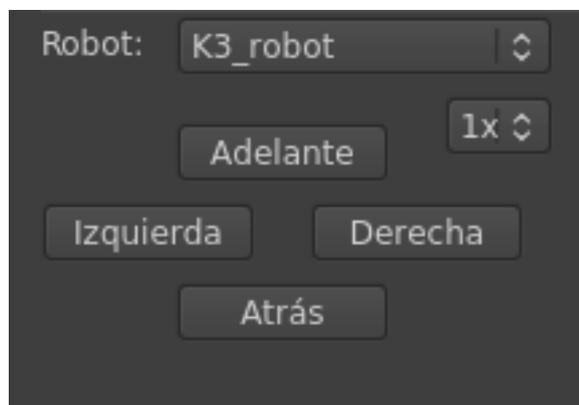


Figura 13: Selección de modo Demo: Botones desplegable

5.3.1. Botones Desplegables

Habilitados solamente en el modo **Demo**, el primero de ellos (el mayor) sirve para seleccionar el robot que queremos controlar, el segundo nos permite seleccionar la velocidad.

5.3.2. Botones: Funciones

Tenemos aquí los 4 botones de interrupción restantes.

En el modo **Demo** estos botones harán al robot seleccionado en el primer desplegable realizar movimientos de giros a izquierda, a derecha, avanzar y retroceder a la velocidad seleccionada en el segundo botón.

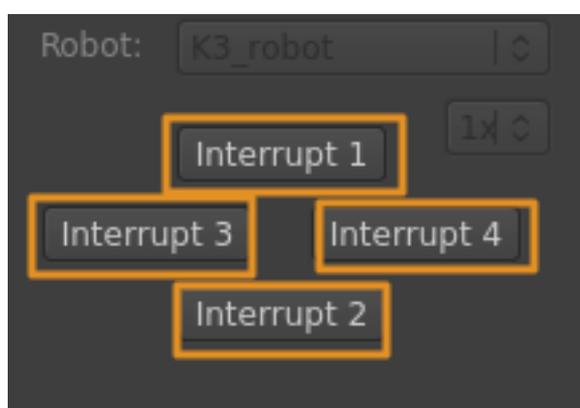


Figura 14: Zona Inferior Izquierda en modo Control

En el modo **Control** los 4 botones quedarán configurados con las funciones programadas por el usuario.

6. Parte inferior derecha



Figura 15: Zona Inferior Derecha

Consta de un panel con un botón desplegable para seleccionar un robot y 6 áreas de texto que muestran información sobre dicho robot, la posición y la velocidad en los 3 ejes.

7. Código del Usuario

Para un funcionamiento con un código propio, el usuario únicamente debe crear una clase, llamada *Control*. Esta clase debe hacer referencia a la clase abstracta *LControl* incluida. *LControl* es una clase abstracta con 5 métodos virtuales puros, y por tanto equivale a una interfaz.

La clase *Control* creada por el usuario por tanto deberá hacer referencia a los 5 métodos de *LControl*, que son invocados por la GUI.

Si el usuario pulsa el botón *Ejecutar* invocará a *Control* y ejecutará el código de control principal.

Si el usuario pulsa algún botón *Interrupt X* se activará una interrupción que ejecutará una acción de código programada por el usuario.

Cabe destacar que cada nueva llamada a una función (por ejemplo cada pulsación de un botón) termina la conexión existente y abre una nueva.

La clase *Control* debe ser definida por el usuario siguiendo las convenciones de *C++*: un archivo *control.h* de declaración y un archivo *control.cpp* de implementación.

7.1. Control.cpp por defecto

Para una mejor experiencia del usuario han incluido ambos archivos en la distribución, con el contenido mínimo para implementar *LControl*, por lo que el usuario solo deberá añadir los atributos y métodos que considere oportunos, ver figura 16.

```
1  #include <control.h>
2  extern "C" {
3  #include "extApi.h"
4  }
5
6
7
8
9  int Control::control(int clientID) {
10     //LLAMADA DESDE EJECUTAR (CONTROL PRINCIPAL)
11     return 0;
12 }
13
14 int Control::interrupt_1(int clientID){
15     //LLAMADA DESDE BOTON DE INTERRUPCION 1
16     return 0;
17 }
18
19 int Control::interrupt_2(int clientID){
20     //LLAMADA DESDE BOTON DE INTERRUPCION 2
21     return 0;
22 }
23
24 int Control::interrupt_3(int clientID){
25     //LLAMADA DESDE BOTON DE INTERRUPCION 3
26     return 0;
27 }
28
29 int Control::interrupt_4(int clientID){
30     //LLAMADA DESDE BOTON DE INTERRUPCION 4
31     return 0;
32 }
33
```

Figura 16: Control.cpp por defecto

Nota:

El código de control del usuario, que podrá emplear la API *remota1* de VREP completa para el control de los kheperas virtuales y de la simulación en sí, se desvincula del resto del código de la aplicación (la GUI) empleando el concepto de herencia e interfaz de los lenguajes de programación orientada a objetos, particularmente el lenguaje *C++*. De esta forma el usuario no necesita conocer el funcionamiento del código de la aplicación y puede dedicarse exclusivamente a programar sus algoritmos de control.

7.2. KheperaSimGui.pro

Si el usuario quisiera añadir otras clases a su código para, por ejemplo, incluirlas en la declaración de Control, podría hacerlo simplemente añadiendo al archivo KheperaSimGui.pro los archivos .h y .cpp que requiera en las secciones SOURCES y HEADERS:

```

15 SOURCES += main.cpp\
16           kheperasimgui.cpp \
17           demo.cpp \
18           control.cpp \
19           remoteApi/extApiPlatform.c \
20           remoteApi/extApiCustom.c \
21           remoteApi/extApi.c \
22           check.cpp
23
24 HEADERS += kheperasimgui.h \
25           i_control.h \
26           include/v_repTypes.h \
27           include/v_repLib.h \
28           include/v_repConst.h \
29           include/extApiCustomConst.h \
30           remoteApi/extApiPlatform.h \
31           remoteApi/extApiInternal.h \
32           remoteApi/extApiCustom.h \
33           demo.h \
34           control.h \
35           check.h
36
37           remoteApi/extApi.h
38

```

Figura 17: Modificación de KheperaSimGui.pro

En la figura 17 se han añadido los archivos check.h y check.cpp al archivo .pro, que en convención C++ definirían una clase Check. Estos archivos se compilarán junto con el resto del programa (tras una nueva ejecución de qmake para actualizar el Makefile).

7.3. Demo

Como ejemplo de lo que se puede hacer con esta estructura, la aplicación tiene un modo *Demo*, que accede a la clase *Demo*.

La clase *Demo* es un ejemplo de todo el procedimiento que se ha descrito para la creación de la clase *Control*. Este modo es muy simple (desplaza dos Kheperas en línea recta, uno al doble de velocidad que el otro, para lo que se incluye la escena *demo2khepera.ttt*) pero sirve como demostración de la aplicación. Además ofrece interrupciones para que al usuario interactúe: marcha adelante/atrás y giro izquierda/derecha, cada uno con 5 niveles de velocidad aunque para niveles superiores a 3 no se garantiza la estabilidad del Khepera.

El resto del código de la aplicación incluye comentarios explicándola para que si el usuario quisiera hacer cambios a la GUI pudiera realizarlos.



Figura 18: Interfaz del modo Demo

Índice de figuras

1.	Descarga Qt 5.2.1.	7
2.	Términoal de Visual Studio 2012.	8
3.	Acceder a la configuración de la variable PATH.	9
4.	Edición de la variable PATH.	10
5.	Ejecutar <i>cmd.exe</i>	11
6.	Interfaz gráfica de usuario.	12
7.	Interfaz Gráfica de Usuario	21
8.	Radiobuttons para selección de modo	21
9.	Botones de funciones	22
10.	Barra de Menú	22
11.	Text Inputs	22
12.	Botones Zona Superior	23
13.	Selección de modo Demo: Botones despleables	23
14.	Zona Inferior Izquierda en modo Control	24
15.	Zona Inferior Derecha	24
16.	Control.cpp por defecto	26
17.	Modificación de KheperaSimGui.pro	27
18.	Interfaz del modo Demo	28