Player/Stage Manual en línea: http://playerstage.sourceforge.net/doc/Player-1.6.5/player-html/index.php

#### Introducción.

Cada fabricante de robots móviles, los dota de un conjunto de dispositivos sensores y actuadores con sus propios comandos para su operación, por lo que la programación realizada para un determinado robot, debe reescribirse para ser empleada en un robot de otra marca.

Player/Stage es una plataforma general para la programación de robots móviles. Consta de dos productos: El Player (la interfaz) y el Stage (el simulador). Su plataforma original fué la familia ActivMedia Pioneer 2.

Player es una interfaz de los dispositivos de un robot. En vez de manipular directamente a los dispositivos, Player proporciona una serie de comandos generales que facilitan trabajar con diferentes marcas de robots y dispositivos y Stage es el simulador donde podemos realizar pruebas virtuales que posteriormente podemos ejecutar directamente sobre el robot real.

Los programas desarrollados en Player, pueden ejecutarse sobre el simulador o sobre el propio robot.

# El simulador Stage.



Stage es un simulador para robots móviles. Se emplea como un plugin con Player o como una librería de C. En este último caso, ya no se usarían las interfaces del Player, sino a través de nuestros propios programas en C.

Para instalar el simulador Stage como plugin de Player, se utilizan dos archivos:

- un archivo de configuración, con extensión "cfg", que contiene la descripción para el control de los dispositivos. Por ejemplo, si se emplea una cámara, contiene el nombre del "driver" y del puerto de entrada empleado.
- otro archivo, (que se referencía en el de configuración) contiene la descripción del mundo (entorno) donde va a desplazarse el robot, el tamaño de la ventana visual en pixeles. También se incluyen las referencias a los archivos de descripción de los dispositivos empleados (incluyendo al propio robot).



Un ejemplo de archivo de configuración, lo encontramos en la instalación del Stage:

```
/usr/local/share/stage/worlds/simple.cfg
```

```
# Desc: Player sample configuration file for controlling Stage devices
# Author: Richard Vaughan
# Date: 1 December 2004
# CVS: $Id: simple.cfg,v 1.30.2.1 2006/07/13 17:59:10 gerkey Exp $
# load the Stage plugin simulation driver
driver
(
 name "stage"
 provides ["simulation:0"]
 plugin "libstageplugin"
 # load the named file into the simulator
 worldfile "simple.world" # enlaza el archivo simple.world
)
driver
(
 name "stage"
 provides ["map:0"]
 model "cave"
)
# Create a Stage driver and attach position2d and laser interfaces
# to the model "robot1"
driver
(
 name "stage"
 provides ["position2d:0" "laser:0" "sonar:0"]
 model "robot1"
)
# Demonstrates use of a Player "abstract driver": one that doesn't
# interface directly with hardware, but only with other Player devices.
# The VFH driver attempts to drive to commanded positions without
# bumping into obstacles.
driver
(
 name "vfh"
 provides ["position2d:1"]
 requires ["position2d:0" "laser:0" ]
)
```

Este enlaza con el archivo /usr/local/share/stage/worlds/simple.world que incluye las referencias a los archivos de descripción de los robots y sus dispositivos.

# Desc: 1 pioneer robot with laser # CVS: \$Id: simple.world,v 1.63 2006/03/22 00:22:44 rtv Exp \$

*#* defines Pioneer-like robots; por ejemplo, sus dimensiones, el número y la posición de los sonares. include "pioneer.inc"

# defines 'map' object used for floorplans
include "map.inc"

# defines sick laser
include "sick.inc"

# size of the world in meters
size [16 16]

*#* set the resolution of the underlying raytrace model in meters resolution 0.02

# update the screen every 10ms (we need fast update for the stest demo)
gui\_interval 20

```
# configure the GUI window
window
(
 size [ 591.000 638.000 ]
 center [-0.010 -0.040]
 scale 0.028
)
# load an environment bitmap
map
(
 bitmap "bitmaps/cave.png"
 size [16 16]
 name "cave"
)
# create a robot
pioneer2dx
 name "robot1"
 color "red"
 #pose [-6.5 -6.5 45]
 pose [-5.25 6.2 359]
 sick laser( samples 361 laser sample skip 4 )
)
```

### La interfaz de Player

Player proporciona interfaces abstractas al robot y sus dispositivos. Y estas intefaces abstractas a su vez se comunican con los "drivers" del robot y dispositivos.

1) Estructura del player/Stage



Las interfaces del Player son estándar para diferentes "drivers" de diferentes dispositivos. Por ejemplo, la interfaz "**position**" permite manejar comandos de desplazamiento de un robot abstracto. En los archivos de configuración se estable el "driver" que corresponda al robots. Igualmente la interfaz "laser" que puede corresponder a un SICK LMS-200 o algún otro.

El acceso a los dispositivos en Player es a través de sockets TCP (Transmission Control Protocol) que son dispositivos de comunicación de dos vías, lo que permite programar el control de sensores y actuadores en varios robots en paralelo.

Los comandos del player, corresponden a los controles y sensores disponibles. Se presentan en dos librerías disponibles: libplayerc y libplayerc++

## Usando la libplayerc.

Un programa básico para mover al robot usando la libplayerc, es /usr/local/share/player/examples/libplayerc/simple.c

```
#include <stdio.h>
#include <libplayerc/playerc.h>
int
main(int argc, const char **argv)
{
 int i;
 playerc client t *client; // la estructura client
 playerc position2d t *position2d; // la estructura position2d
 // Create a client and connect it to the server.
 client = playerc client create(NULL, "localhost", 6665);
 if (0 != playerc client connect(client))
  return -1;
 // Create and subscribe to a position2d device. // corresponde al ROBOT
 position2d = playerc position2d create(client, 0);
 if (playerc position2d subscribe(position2d, PLAYER OPEN MODE))
  return -1:
 // Make the robot spin! // pone al ROBOT a girar
 if (0 != playerc position2d set cmd vel(position2d, 0, 0, DTOR(40.0), 1))
  return -1;
 for (i = 0; i < 200; i++)
 {
  // Wait for new data from server
  playerc client read(client);
  // Print current robot pose // ejes X, Y y ángulo en radianes
  printf("position2d : %f %f %f\n",
       position2d->px, position2d->py, position2d->pa);
 }
 // Shutdown
 playerc position2d unsubscribe(position2d);
 playerc position2d destroy(position2d);
 playerc client disconnect(client);
 playerc_client_destroy(client);
 return 0;
}
Para compilar:
gcc -o simple `pkg-config --cflags playerc` simple.c `pkg-config --libs playerc`
```

### Usando la libplayerc++.

Un programa básico para mover al robot usando la libplayerc++, es /usr/local/share/player/examples/libplayerc++/example0.cc

```
#include <iostream>
#include <libplayerc++/playerc++.h>
int main(int argc, char *argv[])
{
 using namespace PlayerCc;
 PlayerClient robot("localhost");
 SonarProxy
                 sp(&robot,0);
                                                             Front
 Position2dProxy pp(&robot,0);
                                                       5
                                                                   3
                                                             Δ
                                                  6
 for(;;)
 {
                                                           -10<sup>0</sup>
                                                                   100
  double turnrate, speed;
                                                    -300
                                                                          30<sup>0</sup>
                                                                                50<sup>0</sup>
                                              -500
  // read from the proxies
  robot.Read();
                                                              Top
                                                                                 900
                                            - 90
  // print out sonars for fun
  std::cout << sp << std::endl;</pre>
  // do simple collision avoidance
                                              Figure 12. ActivMedia robot sonar array
  if((sp[0] + sp[1]) < (sp[6] + sp[7]))
    turnrate = dtor(-20); // turn 20 degrees per second
  else
    turnrate = dtor(20);
  if(sp[3] < 0.500)
    speed = 0;
  else
    speed = 0.100;
  // command the motors
  pp.SetSpeed(speed, turnrate);
 }
}
```

Compilación:

gcc -o example0 `pkg-config --cflags playerc++` example0.cc `pkg-config --libs playerc++`

Algunos tips para desarrollo de videos a partir de la captura de la pantalla del Simulador Stage.

- Seleccionar File->ScreenShot para activar la captura de imágenes de la ventana del simulador, (indicar "ScreenShot interval" y "sequence of frames")
- Utilizar Stopmotion <u>http://developer.skolelinux.no/info/studentgrupper/2005-hig-</u> <u>stopmotion/index.php</u> para armar el video y guardar en el formato conveniente.



#### Notas de instalación del Player/Stage

Sitio: <u>http://playerstage.sourceforge.net/</u>

Towards Collabora	tive Robot	atrolbot mobile rob	ots - B a Sourc	eForge.net: Files	
	w/vision,Prec www.dwfritz.co	ision.Robotics m	Now! www.AntivirusPower.info	DB www.visual-pa	aradium, com
F.net » Projects » The Pl	layer Project » Files				Eilos Search advance
i në Player Pro	oject	Candada Davi	Table .		Draiest Web Site 5
Project - Tracker -	Mailing Lists - Code	- Services - Down			Project web Site L
					- Enter Here to Research Featured Solutions -
About The Player P	roject				
(including Linux and C	os x) s				
Package	Release	Date	Notes / Monitor	Downloads	VistBooltem
Gazebo	gazebo-0.8-pre1	2007	📫 - 🖂	Download	Up One Level Ctrl+Shift+Up
ibRTK (deprecated)	2.3.0	May 31, 2004	💼 - 🖂	Download	A Moves one level up in the folder tree.
Mezzanine	0.00	May 9, 2002 December 13.		Download	Go Online
Player	2.1.0rc1	2007	·····	Download	
Player/Stage (deprecated)	1.4rc1	July 16, 2003	💼 - 🖂	Download	
pmap (deprecated)	0.1.0	December 12, 2004	👫 - 🖂	Download	
Stage	2.1.0rc1	December 13, 2007	ita - 🖂	Download	
File Releases Below is a list of all fil	es released by this	project. Before down	loading, you may wan	to read Release Note	es. The current release for each package is shown.

Una vez que se baja el archivo a tu cuenta de usuario, se descomprime en un subdirectorio.

Location Edit View Go Bookmarks Tools Settings Window Help										
Cocation: S /home/apolinar/Player Stage/player-2.1.0rc1										
👞 🏠 Home Folder 🛛 🗖	Name		Size	Modifie						
🚬 - 🔄 aaa	aplayer-2, 1, 0rc1	Folder	4.0 KB	13/12/0						
2 - Agosto Dic2007	÷ S client libs	Folder	4.0 KB	13/12/0						
🔚 - 😫 Algoritmos	+ S config	Folder	4.0 KB	13/12/0						
🚊 - 😂 amsn_received	+- Adoc	Folder	4.0 KB	13/12/0						
🖳 🗄 😂 build	+ Sexamples	Folder	4.0 KB	13/12/0						
🚊 – 👌 Desktop	+-S libplavercore	Folder	4.0 KB	13/12/0						
🖹 🖮 😂 DrSucar	-Silbplaveripeg	Folder	4.0 KB	13/12/0						
🟠 🗄 😂 install_flash_player_9_linux	t-S libplayersd	Folder	4.0 KB	13/12/0						
🛄 🗄 🔯 install_flash_player_9_linux.tar.gz	- Sibplayertcp	Folder	4.0 KB	13/12/0						
兰 - 😫 Linux	Bibplaverxdr	Folder	4.0 KB	13/12/0						
inux-2.6-2.6.18.dfsg.1	+- S replace	Folder	4.0 KB	13/12/0						
늘 🗄 🕲 linux-2.6_2.6.18.dfsg.1.orig.tar.gz	e Strtk2	Folder	4.0 KB	13/12/0						
💐 🗄 😂 Machine Learning	+ Server	Folder	4.0 KB	13/12/0						
🔥 🗄 🕲 Markov_chain	+ Cautils	Folder	4.0 KB	13/12/0						
🚔 🗄 😂 musica	- DINEWS	Empty Document	0 8	07/08/0						
🛋 🗄 🍓 O'ReillyLearning_Debian_GNU_Linux.:	- 🚯 Makefile.am	Makefile	508 B	13/12/0						
👘 🗄 🙆 O'Reilly - Pthreads_Program.zip	- Makefile.in	Makefile	41.0 KB	13/12/0						
🗄 😂 Pioneer Arm	- Di config.h.in	Objective-C Source File	10.7 KB	11/12/0						
🗄 🍓 Player_Stage	-Sacinclude.m4	Plain Text Document	18.4 KB	11/12/0						
⊕ 🕲 boost_1_34_0.tar.gz	-Saclocal.m4	Plain Text Document	261.4 KB	11/12/0						
🕂 🔯 cap07.zip	AUTHORS	Plain Text Document	61 B	27/02/0						
🕂 🔯 gazebo-0.7.0.tar.gz		Plain Text Document	597.8 KB	13/12/0						
gdk-pixbuf-0.22.0	-Si config.p2os	Plain Text Document	968 B	31/05/0						
👳 🔯 gdk-pixbuf_0.22.0.orig.tar.gz	-Si config.simulation	Plain Text Document	808 B	31/05/0						
🕂 😳 IEEEtranBST.tar.gz	-Si configure.ac	Plain Text Document	27.7 KB	10/12/0						
🗄 🔯 IEEEtran.tar.gz		Plain Text Document	17.6 KB	25/09/0						
🕂 🔞 MPlayer-1.0rc1.tar.bz2		Plain Text Document	25.8 KB	13/12/0						
e Sopency-1.0.0	-SI INSTALL	Plain Text Document	9.3 KB	21/08/0						
⊕ 🔯 openc∨-1.0.0.tar.gz		Plain Text Document	586 B	27/02/0						
player-2.0.4	TODO	Plain Text Document	1.1 KB	24/01/0						
🕂 🔯 player-2.0.4.tar	🗐 compile	Shell Script	3.6 KB	25/10/0						
player-2.0.4.tar.bz2	- Config.guess	Shell Script	42.5 KB	25/10/0						
e aplayer-2.1.0rc1	- 🗐 config.sub	Shell Script	31.0 KB	25/10/0						
⇒⊜player-2.1.0rc1	- Configure	Shell Script	1.8 MB	11/12/0						
player-2.1.0rc1.tar.bz2	- Depcomp	Shell Script	15.6 KB	25/10/0						
# Sqc-usb-0.6.6	install-sh	Shell Script	9.0 KB	25/10/0						
README (586 B) Plain Text Document										
Image: Weight of the second secon	a 🛇 🥑 🖬 🤜 🌠 🔏 📼	18 : 42 ·								

#### 1er Paso: Configurar el paquete: ./configure

Revisa paquetes indispensables para la instalación apropiada del Player:

Si sale este mensaje:

- Support for plugin drivers will NOT be included.
- You need plugin support to use Stage
- To add plugin support, install libltdl which is part of GNU libtool
- multithreading NOT included; Install BOOST (<u>www.boost.org</u>) libraries to enable multithreaded and/or signaling in libplayerc++

lista de drivers que se instalarán y los que no se instalarán

Player will be installed in: /usr/local/ (en /usr/local/bin/player y en /usr/local/share/player)

2) Cambiar a Superusuario y Teclear: make install

#### Instalación de Stage.

Bajar el archivo a un directorio y descomprimir:

- bzip2-d Stage...bz2
- tar -xvf Stage...tar

Instalar GTK+ <u>http://www.gtk.org</u> herramienta para crear interfaces gráficas de usuario. o librería libgtkmm-2.4

1er Paso: Configurar el paquete: ./configure 20. paso: Cambiar a Superusuario y Teclear: make install

Stage queda instalado en /usr/local (/usr/local/share/stage/worlds, con archivos de configuración y del ambiente )

Probando el simulador Stage:

En /usr/local/share/stage/worlds/player simple.cfg

Probando la interfaz Player:

En /usr/local/share/player/examples/libplayerc/simple En /usr/local/share/player/examples/libplayerc++/example0

Para comentarios, dudas:

Apolinar Ramírez, <u>apolinar\_r@ccc.inaoep.mx</u> Cubo 1114 -INAOE