

Pilotage des modèles de Virtual Universe avec la dll Vudll.dll

Langage C

Python

Java

1- dll

Le fichier vudll.dll (ou vudll64.dll) se trouve dans le sous-répertoire dll du répertoire d'installation de Virtual Universe (version 1.113 ou supérieure).

La dll vudll.dll (ou vudll64.dll) exporte 3 fonctions génériques :

int VuConnect(void)

Connexion à Virtual Universe et lancement de l'exécution. En retour : 0 si connexion réalisée, -1 autrement. Virtual Universe doit être lancé avant l'appel de cette fonction. Le projet à simuler doit également être ouvert. Si une simulation est déjà en cours, celle-ci est réinitialisée.

float VuRead(char *item)

Retourne la valeur du comportement dont le nom est donné par <item>.

int VuWrite(char *item,float value)

Ecrit la valeur du comportement dont le nom est donné par <item> avec la valeur <value>.

2- langage C

Exemple de programme (Ce fichier est l'exemple se trouvant dans le sous-répertoire dll\c du répertoire d'installation de Virtual Universe) :

```
// Exemple de pilorage du robot aspirateur de Virtual Universe avec la dll  
vudll.dll
```

```
#include "stdio.h"  
#include "conio.h"  
#include "windows.h"  
  
int (_stdcall *VuConnect)(void);  
float (_stdcall *VuRead)(char *);  
int (_stdcall *VuWrite)(char *,float);  
  
int main(int argc, char * argv[])  
{  
    int stop=0;  
    HMODULE h;  
    h=LoadLibrary(L"vudll.dll"); // use vudll64.dll windows 64 bits  
    if(!h)  
    {  
        return -2;  
    }  
    VuConnect=(int (_stdcall *) (void))GetProcAddress(h, "VuConnect");  
    VuRead=(float (_stdcall *) (char *))GetProcAddress(h, "VuRead");  
    VuWrite=(int (_stdcall *) (char *,float))GetProcAddress(h, "VuWrite");  
  
    if(VuConnect()!=0)  
    {  
        printf("Can't connect to Virtual Universe\n");  
        return -1;  
    }  
do  
    {  
        printf("\r\nGo straight (press ESC to stop program)...");  
        VuWrite("moteurdroitavant",1);  
        VuWrite("moteurgaucheavant",1);  
        while(VuRead("detecteurcollision")==0&&VuRead("detecteurvide")==0)  
        {  
            if(_kbhit()&&_getch()==27)  
            {  
                stop=1;  
                break;  
            }  
            if(stop) break;  
            printf("\r\nGo back...");  
            VuWrite("moteurdroitavant",0);  
            VuWrite("moteurgaucheavant",0);  
            VuWrite("moteurdroitARRIERE",1);  
            VuWrite("moteurgaucheARRIERE",1);  
            Sleep(4000);  
  
            printf("\r\nRotating...");  
            VuWrite("moteurdroitavant",1);  
            VuWrite("moteurdroitARRIERE",0);  
            Sleep(2000);  
  
            VuWrite("moteurgaucheARRIERE",0);  
        }  
    }  
while(1);  
  
return 0;  
}
```

3- Python

Le fichier vu.py permet de un accès simple aux fonctions de la dll vudll.dll

Ce fichier et l'exemple se trouvant dans le sous-répertoire dll\python du répertoire d'installation de Virtual Universe :

```
#Exemple de pilotage de Virtual Universe à partir de Python

from vu import *

import time

if VuConnect()==0:

    while 1:

        print "Avancer"

        VuWrite("moteurdroitavant",1)

        VuWrite("moteurgaucheavant",1)

        #attendre qu'un capteur soit détecté

        while VuRead("detecteurcollision")==0 and VuRead("detecteurvide")==0:

            time.sleep(0.01)

        print "Reculer"

        VuWrite("moteurdroitavant",0)

        VuWrite("moteurgaucheavant",0)

        VuWrite("moteurdroitARRIERE",1)

        VuWrite("moteurgaucheARRIERE",1)

        time.sleep(4)

        print "Tourner"

        VuWrite("moteurdroitavant",1);

        VuWrite("moteurdroitARRIERE",0);

        time.sleep(2)

        VuWrite("moteurgaucheARRIERE",0);
```

4- Java

Ce fichier est l'exemple se trouvant dans le sous-répertoire dll\java du répertoire d'installation de Virtual Universe :

```
package CInterface;

import com.sun.jna.Library;
import com.sun.jna.Native;
import com.sun.jna.Platform;

public class hello
{
    public static void main(String[] args)
    {
        // Loading dynamically the library

        CInterface demo = (CInterface) Native.loadLibrary("C:/Program Files/Virtual
Universe/dll/vudll164.dll", CInterface.class);

        System.out.print("Connecting...");

        if(demo.VuConnect()==0)
        {
            System.out.println("Connected");

            do
            {
                System.out.println("Go straight...");

                demo.VuWrite("moteurdroitavant",1);

                demo.VuWrite("moteurgaucheavant",1);

                while(demo.VuRead("detecteurcollision")==0&&demo.VuRead("detecteurvide")==0);

                System.out.println("Go back...");

                demo.VuWrite("moteurdroitavant",0);

                demo.VuWrite("moteurgaucheavant",0);

                demo.VuWrite("moteurdroitARRIERE",1);

                demo.VuWrite("moteurgauchearriere",1);

                long ms=System.currentTimeMillis();

                while(System.currentTimeMillis()-ms<4000);

                System.out.println("Rotating...");

                demo.VuWrite("moteurdroitavant",1);

                demo.VuWrite("moteurdroitARRIERE",0);
```

```
ms=System.currentTimeMillis();

while(System.currentTimeMillis()-ms<2000);

demo.VuWrite("moteurgauchearriere",0);

}

while(1==1);

}

else

{

    System.out.println("Can't connect, please launch Virtual Universe and open the
project to simulate");

}

}

}
```