

Devices-Controler

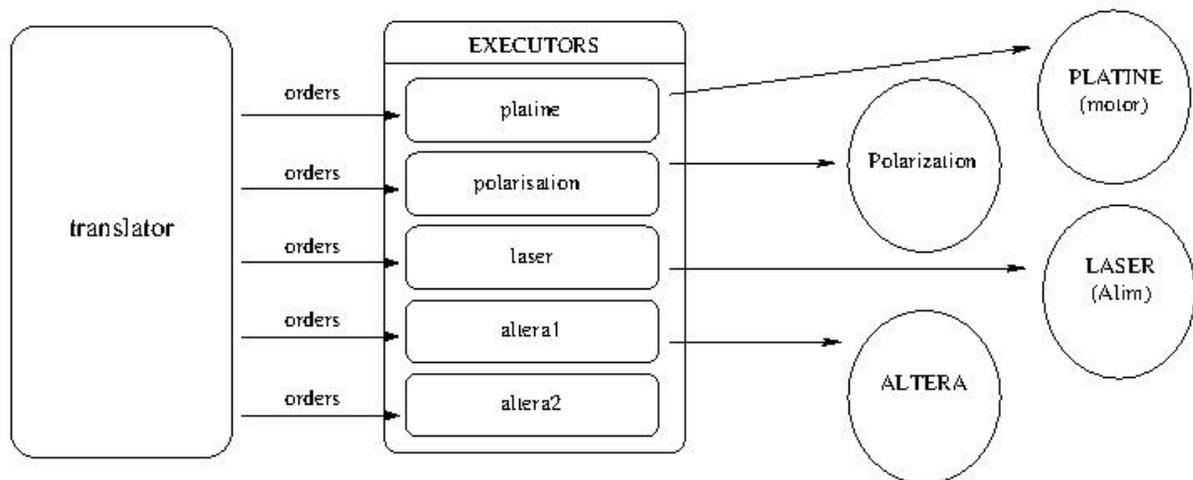
Contents

1	Introduction	1
2	Diagramme de flux	1
2.1	Niveau n+1	1
2.2	Événements	1
3	Implémentation	2
4	Générateur de squelette	2
4.1	Installation	2
4.2	Configuration	2
4.3	Make	2
5	Prise en main du nouveau driver	3
6	Edition des liens	3
6.1	Appel à une bibliothèque dynamique	3
6.2	Linkage avec un fichier objet C	4
6.3	Linkage avec un fichier objet C++	4

1 Introduction

Cette page décrit la sous activité '**executors**' faisant partie de l'activité '**devices-controler**'. Il s'agit d'une abstraction du code communiquant avec les différents modules. C'est ici que sera localisé le code spécifique d'accès au périphériques.

2 Diagramme de flux



2.1 Niveau n+1

L'activité se divise 4 traitements :

- platine
- polarisation
- laser
- altera

2.2 Événements

L'activité réagit via les évènements suivants :

- *platine-executor-orders* en provenance du traitement translator et a destination du traitement platine
- *polarisation-executor-orders* en provenance du traitement translator et a destination du traitement polarization
- *laser-executor-orders* en provenance du traitement translator et a destination du traitement laser
- *altera1-executor-orders* en provenance du traitement translator et a destination du traitement toAltera1
- *altera2-executor-orders* en provenance du traitement translator et a destination du traitement toAltera2

3 Implémentation

Il n'y a pas de code à ce niveau. Néanmoins, chacun des traitements inclus dans cette activité sont codés de la même manière. Aussi, nous exposons ci-dessous les grandes lignes de l'implémentation de ces traitements.

4 Générateur de squelette

4.1 Installation

Instructions officielles :

```
# su - enx
$ cd ~/svn/enx/EnxTools/Driver_Generator

fichier /.bash_profile

# needed for driver generator's configuration
export ENX_SRC=~/svn/enx/Trunk/

# needed for ENX to find it's plugins (the drivers)
export LD_LIBRARY_PATH=~/ENX-2.4/i686/Drivers
```

4.2 Configuration

```
$ ./configure --enable-C --with-name=monDriver
```

- You must define the driver name (--with-name=...).

```
$ ./configure --with-name=monDriver
```

- You should look for environment variable ENX_SRC

```
$ export ENX_SRC=/home/enx/svn/enx/Trunk/
```

4.3 Make

```
$ make
```

```
Creating stubs for monDriver driver
  Creating directory build/monDriver
  Copying files
DONE.
```

```
$ cp -fr build/monDriver ~/.
```

```
$ cd ~/monDriver
```

```
$ make
```

```
$ make install
```

5 Prise en main du nouveau driver

- charger le nouveau **driver** en mémoire :

```
$ ~/ENX-2.4/i686/console
```

```
ENX > !driver_add monDriver libmondriver.so
```

```
ENX > !list driver
```

```
- monDriver
- DRIVER_DEBUG
```

- création d'un nouveau **device** utilisant le nouveau **drivers** :

```
ENX > !create monDevice 0 0xFF monDriver
```

```
ENX > !list device
```

```
- monDevice
```

- utilisation :

```
ENX > md monDevice 0
```

```
249 INFO [main_task_0810D0D0] ENX.Console - 0x00000000 -> 0x0C000000
```

```
ENX > mm monDevice 0 0x1234
```

```
256 INFO [main_task_0810D0D0] ENX.Console - OK
```

```
ENX > md monDevice 0
```

```
268 INFO [main_task_0810D0D0] ENX.Console - 0x00000000 -> 0x00000000
```

```
ENX > special monDevice test
```

```
315 INFO [main_task_0810D0D0] ENX.Console - SUPER TEST !!!
```

```
ENX > help monDevice
```

```
329 INFO [main_task_0810D0D0] ENX.Console - SUPER HELP !!!
```

6 Edition des liens

6.1 Appel à une bibliothèque dynamique

Fichier */monDriver/libmondriver.gpr* :

```
for Library_Options use ("../altera2ada.o", "-L/usr/local/lib", "-lftd2xx");
```

Fichier */monDriver/libmondriver.gpr* :

```
...
#include "ftd2xx.h"
...
char* special (void *struct_c, char *command)
{
    char *return_value=(char*)0;

    FT_STATUS l_ftStatus;
    DWORD l_luNumDevs;

    // create the device information list
    l_ftStatus = FT_CreateDeviceInfoList(&l_luNumDevs);
    if (l_ftStatus == FT_OK) {
        printf("Number of devices is %lu\n",l_luNumDevs);
    }

    return return_value;
}
...
```

6.2 Linkage avec un fichier objet C

Fichier */monDriver/libmondriver.gpr* :

```
for Library_Options use ("../altera2ada.o", "../toto.o");
```

Fichier */monDriver/libmondriver.gpr* :

```
...
extern int tata();
...
char* special (void *struct_c, char *command)
{
    char *return_value=(char*)0;

    printf("printf: toto = %i\n", tata());

    return return_value;
}
...
```

6.3 Linkage avec un fichier objet C++

Fichier */monDriver/libmondriver.gpr* :

```
for Library_Options use ("../altera2ada.o", "../toto.o");
```

Fichier */monDriver/libmondriver.gpr* :

```
...
extern int tata();
...
char* special (void *struct_c, char *command)
{
    char *return_value=(char*)0;

    printf("printf: toto = %i\n", tata());

    return return_value;
}
...
```

Fichier */monDriver/toto.cc* :

```
extern "C" int tata()
{
    static int bof=42;
    return ++bof;
}
```