dans le cas d'un composant i2c 5V d'utiliser des convertisseurs de niveau ou, plus simplement une version 1,8 - 3,3 V dudit composant si elle existe.

3 En parlant d'espace utilisateur

En ayant appliqué toutes ces modifications dans le code et en ayant pris soin de configurer le noyau en activant le support i2c et i2c-dev, il nous suffit de recompiler l'ensemble. Profitons-en pour sensiblement modifier une partie du reste du système. Je pense en particulier aux entrées dans /dev. Nous devons, en effet, disposer des pseudo-fichiers correspondants aux bus i2c pour pouvoir les utiliser depuis l'espace utilisateur. L'outil utilisé pour générer l'image du rootfs offre une facilité dans ce sens. Il nous suffit de créer des fichiers vides dans notre /rootfs/dev ainsi:

```
% touch rootfs/dev/@f2c-0.c.89.8
% touch rootfs/dev/@12c-1,c,89,1
% touch rootfs/dev/@i2c-2,c.89,2
```

La syntaxe utilisée est simple : « @nom,type,majeur,mineur », exactement comme on le ferait avec mknod. Enfin, pour que nous puissions tester simplement le bon fonctionnement de l'ensemble, nous allons récupérer les sources des i2c-tools du projet lm-sensors sur http://www.lm-sensors.org/wiki/ 12CTools. La cross-compilation nécessite simplement de spécifier la racine du compilateur en modifiant le Makefile :

```
CROSS_COMPILE ?= arm-uclinuxeabi-
                 = $(CROSS_COMPILE)as
AS
                 = $(CROSS_COMPILE)gcc
CC
                = $(CC) -nostdlib
= $(CC) -E
LD
CPP
AR
                 = $(CROSS_COMPILE)ar
                 = $(CROSS_COMPILE)nm
NM
STRIP
                 = $(CROSS_COMPILE)strip
                 = $(CROSS_COMPILE)objcopy
OBJCOPY
OBJDUMP
                 = $(CROSS_COMPILE)objdump
```

Après compilation, d'un simple make, nous trouverons l'outil qui nous intéresse dans le sous-répertoire tools : i2cdetect. Nous copierons ce binaire dans rootfs/usr/ bin/ (là ou se trouve déjà fbstest) et reconstruirons l'ensemble du système avant de l'inscrire dans la flash du devkit avec OpenOCD.

Et voici le résultat :

```
12c /dev entries driver
  12c-2 at 40005c00,irg=72
 # cat /proc/interrupts
          CPUS
                      NVIC STM32 TEX10/PAG Button
```

```
NVIC STM32 TEXI3/PC3 Button
NVIC STM32 Kernel Time Tick
NVIC STM32 USART Port
                            NVIC STM32 USART Port
NVIC 12c_stm32.2
NVIC 12c_stm32.2
- # i2cdetect -1
12c-2 12c 12c stm32.2 I2C adapter
~ # 12cdetect 2
MARNING! This program can confuse your I2C bus,
  will probe file /dev/i2c-2
I will probe address range 0x03-0x77.
Continue? [Y/n]
88-
```

Notre bus i2c est bel et bien présent et utilisable. Notez qu'il se présente sous la forme d'une entrée /dev/i2c-2 (on compte à partir de 0 alors que le bus est, pour la plateforme, le 12C3). La commande permettant de scanner le bus nous affiche bien un composant à l'adresse 0x41, c'est notre STMPE811! Ceci est totalement indépendant de notre éventuelle dernière modification dans le code du noyau où nous n'avons fait que signaler qu'un composant était présent. Mais celui-ci ne disposant pas de pilote l'ayant réclamé, cette déclaration est inutile. Ce que nous avons là sous les yeux est un scan ou en d'autres termes le résultat une tentative d'accès systématique à chaque adresse du bus afin de voir « si quelque chose répond ». Et c'est le cas, notre bus est totalement fonctionnel.

4 Conclusion et devoirs de vacances

Ou'avons-nous obtenu jusqu'à présent ? Un système basé sur uClinux, disposant d'un framebuffer, accessible via une liaison série et avec, maintenant, un bus i2c sur lequel brancher tout ce qu'on souhaite. Pour le prix du STM32F429IDISCO, moins de 20 euros (soit le prix d'un Arduino), je trouve que cela est absolument fantastique et grisant. Nous avons là une plateforme extrêmement puissante et complète. Les 2 Mo de flash sont encore largement sous utilisées. En effet, nous avons 128 Ko pour le bootloader, 1024 Ko pour le noyau. Il nous reste donc 896 Ko pour le rootfs et à ce stade, avec fbtest et i2cdetect inclus, nous n'occupons que 425 Ko. De quoi encore largement nous en donner à cœur joie!