La chaîne de compilation sera GCC 3.4 en cross-compilation (type `arm-linux-gcc`).

Les sources du noyau ainsi que la chaîne de développement sont généralement fournies dans les kits de développement, si bien que la première chose à faire sera de reconstruire le noyau… ou du moins les drivers dont nous avons besoin en tant que modules :

- `touchscreen`,
- `I2C`,
- `FrameBuffer` si celui-ci n'est pas déjà sélectionné dans votre distribution.

Tapez donc la commande `make xconfig` (Figure 1) à la racine de vos sources et sélectionnez le support du bus I2C et le driver de chip PCF8574 en tant que module :

Faites de même pour la gestion de l'écran tactile et vérifiez avant compilation que la section `Graphics support` supporte la gestion du `Frame Buffer` (Figure 2).

Afin de recompiler le noyau si besoin et/ou de compiler les drivers en tant que modules, tapez simplement la commande `make`. Le calcul de dépendances, ainsi que la compilation du noyau et des modules seront automatiquement effectués par celle-ci.

Les modules seront présents sous *drivers/input/touchscreen/* pour le module de gestion d'écran tactile et sous *drivers/i2c/chips* pour celui de la gestion du PCF8574. Les fichiers modules qui pourront être linkés dynamiquement au noyau porte l'extension *.ko* pour les distinguer des fichiers *.o* destinés au linker (la confusion était courante sur le noyau 2.4).

Si un nouveau noyau (zImage en format compressé) est disponible, vous aurez besoin de le flasher sur la carte. Aucunement besoin de sonde de programmation si votre noyau est stocké dans la flash NAND : demandez au noyau en cours de fonctionnement de reflasher la carte pour vous ! Transférez le fichier zImage sous */var* et faites :

```
cp /var/zImage /dev/mtdblock/1
```

si la seconde partition de la flash NAND est bien celle de votre noyau (c'est souvent le cas, la première étant destinée au BIOS). Quant aux modules, vous pourrez les transférer sur la cible puis les charger dynamiquement à l'aide de la commande `insmod xxx.ko`.

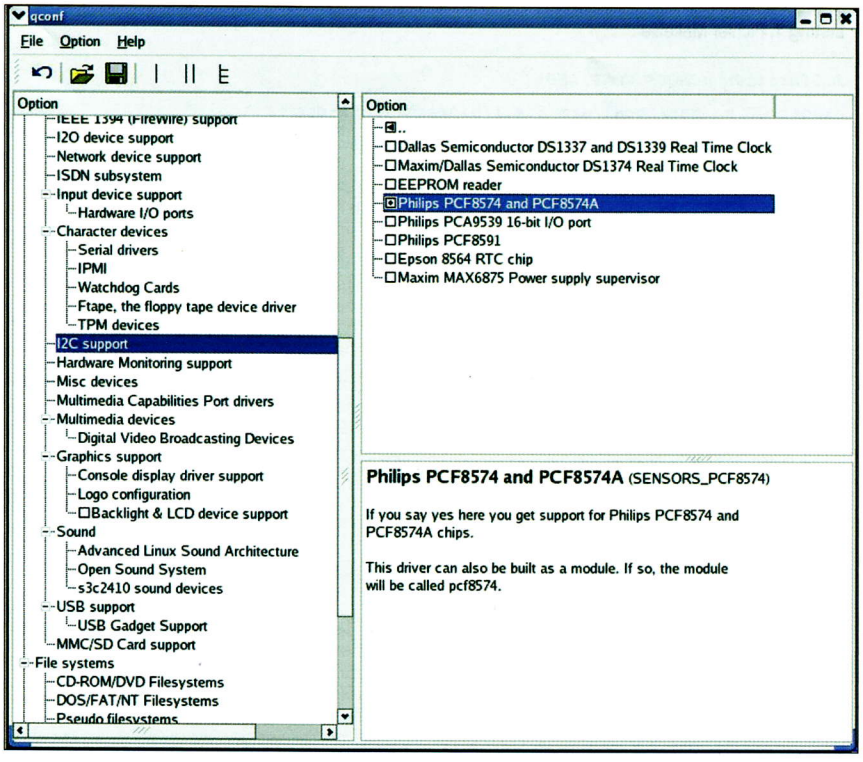

Figure 1. Make xconfig de Linux 2.6

Utilisation du LCD

Notre carte de développement bénéficie d'un écran LCD TFT 320x240 et d'un

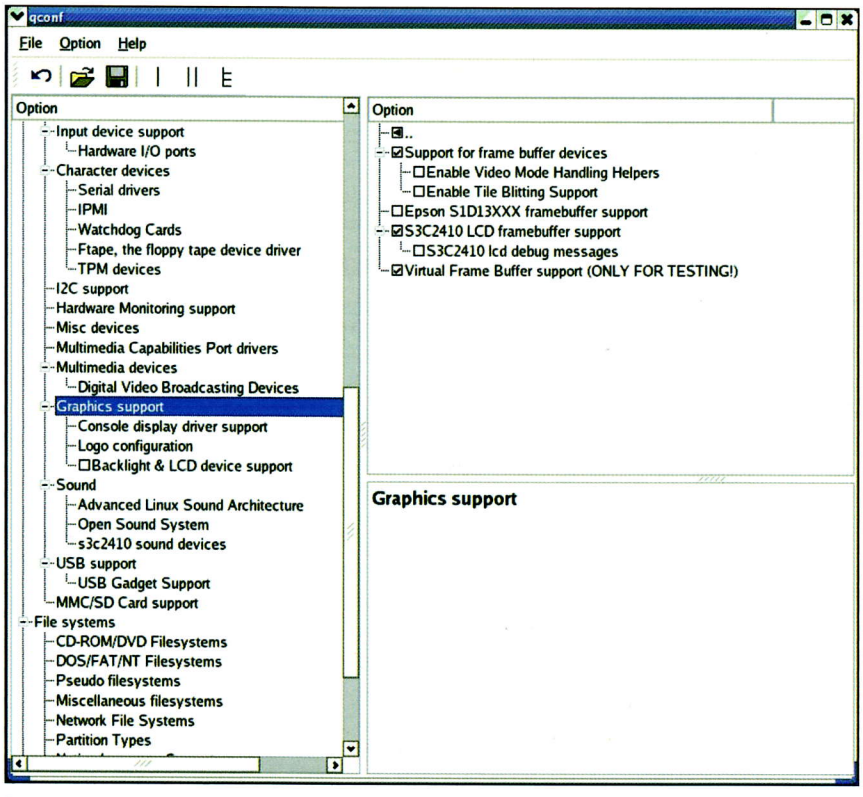

Figure 2. FrameBuffer_xconfig

Ce qu'il faut savoir...

Les cartes de développement d'aujourd'hui permettent de réaliser des applications pour de véritables PC embarqués. Tout comme dans le cas d'un PC, ils disposent d'une vidéo sur LCD, d'une gestion d'entrées/sorties type clavier/souris/écran tactile, mais sont en plus connectés au monde extérieur : cartes d'extension sur bus I2C ou SPI, convertisseur analogique/numérique, PWM…