

```
# 13: 0x00284000 (0x4000 16kB) not protected
# 14: 0x00288000 (0x4000 16kB) not protected
# 15: 0x0028c000 (0x4000 16kB) not protected
# 16: 0x00290000 (0x10000 64kB) not protected
# 17: 0x002a0000 (0x20000 128kB) not protected
# 18: 0x002c0000 (0x20000 128kB) not protected
# 19: 0x002e0000 (0x20000 128kB) not protected
# 20: 0x00300000 (0x20000 128kB) not protected
# 21: 0x00320000 (0x20000 128kB) not protected
# 22: 0x00340000 (0x20000 128kB) not protected
# 23: 0x00360000 (0x20000 128kB) not protected
stm32f4x - Rev: unknown
shutdown command invoked
```

Ici, cette version Debian 0.7.0-2 liste les blocs de flash avec un « vide » (*gap*) entre les blocs 11 et 12 alors que les banques de mémoire sont normalement adjacentes. Les tentatives d'accès à une adresse > **0x08280000** provoqueront une erreur, ce qui est le cas pour l'image du rootfs (**romfs.bin**) débutant à **0x08120000**. La sortie affichée par la version 0.8.0 en cours de développement est très différente :

```
#0 : stm32f4x at 0x08000000, size 0x00200000, buswidth 0, chipwidth 0
# 0: 0x00000000 (0x4000 16kB) not protected
# 1: 0x00004000 (0x4000 16kB) not protected
# 2: 0x00008000 (0x4000 16kB) not protected
# 3: 0x0000c000 (0x4000 16kB) not protected
# 4: 0x00010000 (0x10000 64kB) not protected
# 5: 0x00020000 (0x20000 128kB) not protected
# 6: 0x00040000 (0x20000 128kB) not protected
# 7: 0x00060000 (0x20000 128kB) not protected
# 8: 0x00080000 (0x20000 128kB) not protected
# 9: 0x000a0000 (0x20000 128kB) not protected
# 10: 0x000c0000 (0x20000 128kB) not protected
# 11: 0x000e0000 (0x20000 128kB) not protected
# 12: 0x00100000 (0x4000 16kB) not protected
# 13: 0x00104000 (0x4000 16kB) not protected
# 14: 0x00108000 (0x4000 16kB) not protected
# 15: 0x0010c000 (0x4000 16kB) not protected
# 16: 0x00110000 (0x10000 64kB) not protected
# 17: 0x00120000 (0x20000 128kB) not protected
# 18: 0x00140000 (0x20000 128kB) not protected
# 19: 0x00160000 (0x20000 128kB) not protected
# 20: 0x00180000 (0x20000 128kB) not protected
# 21: 0x001a0000 (0x20000 128kB) not protected
# 22: 0x001c0000 (0x20000 128kB) not protected
# 23: 0x001e0000 (0x20000 128kB) not protected
```

Notez également que pour cette version il n'est plus utile de spécifier l'interface et la plateforme avec **-f interface/stlink-v2.cfg -f target/stm32f4x_stlink.cfg** puisqu'un profil de *board* existe : un simple **-f board/stm32f429discovery.cfg** suffit donc.

Note

NDLR : OpenOCD 0.8-stable est maintenant disponible et règle tous ces problèmes, mais n'est certainement pas encore packagé pour votre distribution GNU/Linux

On pourra éventuellement modifier le fichier **mk/flash.mak** pour affiner les réglages mais un simple **make install** fera l'affaire en invoquant OpenOCD ainsi :

```
FLASH_CMD := openocd \
-f interface/stlink-v2.cfg \
-f target/stm32f4x_stlink.cfg \
-c "init" \
-c "reset init" \
-c "flash probe 0" \
-c "flash info 0" \
-c "flash write_image erase $(uboot_target) 0x00000000" \
-c "flash write_image erase $(kernel_target) 0x08020000" \
-c "flash write_image erase $(rootfs_target) 0x08120000" \
-c "reset run" -c shutdown
```

Comme vous pouvez le voir, les différentes images issues de la compilation sont inscrites en flash, à commencer par le bootloader en tout début de mémoire. Suite à cette opération et grâce au **reset run** demandé à OpenOCD, le devkit va redémarrer et, normalement, vous fournir toutes les informations de démarrages jusqu'à l'invite du shell via les connecteurs PC10 et PC11 correspondants respectivement aux lignes TX et RX du troisième port série de la carte (USART3). Il vous suffira donc d'un simple convertisseur série/USB pour accéder au système. Attention cependant, les niveaux de tension utilisés sont 0-3,3V. Vous devrez donc veiller à utiliser un module compatible, le devkit n'est PAS tolérant au 5V !

Le démarrage s'accompagne également de l'initialisation du périphérique framebuffer de Linux et donc de l'affichage sur l'écran d'un mignon petit Tux. Dans la version initiale du portage, la console en framebuffer était activée et affichait donc les messages de démarrage. Ceci a depuis été retiré étant donné qu'aucun périphérique de saisi n'était disponible. Dans les développements annoncés par certains participants au fil de discussion initial sur le forum ST figurait le support de l'USB hôte permettant ainsi la connexion d'un périphérique HID comme un clavier. Il semblerait cependant que ceci ne soit pas encore publiquement implémenté à la date de rédaction de cet article.

Pour vous assurer du bon fonctionnement du framebuffer, un petit programme appelé **fbtest** est inclus au firmware. Celui-ci affiche un cube 3D en rotation sur l'écran LCD du devkit avec une fluidité tout à fait acceptable.

3 Petits changements

L'exploration des sources, suite à cet essai fructueux mais sans grand intérêt technique, se fera de préférence dans l'ordre de chargement et d'exécution du système. Ceci débute donc avec U-Boot et en particulier le fichier **include/configs/stm32429-disco.h**. On y apprend, par exemple, qu'il n'y a pas d'environnement pour le bootloader :