

```
#define CONFIG_ENV_IS_NOWHERE
```

Ceci implique que le binaire U-Boot intègre un environnement par défaut qu'il ne nous est pas possible de changer sans compilation. Toutefois, il pourrait être intéressant d'accéder à U-Boot avant le chargement du noyau pour éventuellement changer les arguments à des fins de test. Pour cela, nous devons changer le délai d'attente par défaut à zéro seconde :

```
#define CONFIG_BOOTDELAY 0
```

Glisser ici un **5** et réinstaller le bootloader nous donnera davantage de libertés d'expérimentations. Toujours dans ce même fichier, on retrouve la configuration par défaut utilisée :

```
#define CONFIG_BOOTARGS "stm32_platform=stm32429-disco mem=7M \"\n\
                        \"console=ttyS2,115200n8 consoleblank=0 \"\n\
                        \"root=/dev/mtdblock0 rdinit=/sbin/init \"\n\
                        \"video=vfb:enable,fbmem:0x90700000,fbsize:0x1000000\"
#define CONFIG_BOOTCOMMAND "run flashboot"

[...]

#define CONFIG_EXTRA_ENV_SETTINGS \
    "loadaddr=0x90000000\0" \
    "addip=setenv bootargs ${bootargs}\0" \
    "flashaddr=08020000\0" \
    "flashboot=run addip;bootm ${flashaddr}\0" \
    "image=uImage\0" \
    "stdin=serial\0" \
    "stdout=serial\0" \
    "stderr=serial\0"
```

Enfin, la liaison série fait également partie de la configuration :

```
#define CONFIG_STM32_USART_CONSOLE
#define CONFIG_STM32_USART_PORT 3 /* USART3 */
#define CONFIG_STM32_USART_TX_IO_PORT 2 /* PORTC */
#define CONFIG_STM32_USART_RX_IO_PORT 2 /* PORTC */
#define CONFIG_STM32_USART_TX_IO_PIN 10 /* GPIO10 */
#define CONFIG_STM32_USART_RX_IO_PIN 11 /* GPIO11 */

#define CONFIG_BAUDRATE 115200
#define CONFIG_SYS_BAUDRATE_TABLE \
    { 9600, 19200, 38400, 57600, 115200 }
```

L'une des premières tâches qu'on peut s'attribuer est de basculer la console série sur **ttyS0** en lieu et place de **ttyS2** qui semble un peu atypique. Pour ce faire on changera le contenu de **include/configs/stm32429-disco.h** en :

```
#define CONFIG_STM32_USART_PORT 1 // USART1
#define CONFIG_STM32_USART_TX_IO_PORT 0 // PORTA
#define CONFIG_STM32_USART_RX_IO_PORT 0 // PORTA
#define CONFIG_STM32_USART_TX_IO_PIN 10 // GPIO10
#define CONFIG_STM32_USART_RX_IO_PIN 9 // GPIO9
```

Ce qui nous permettra de connecter notre adaptateur série/USB aux broches PA10 (TX) et PA9 (RX) du dev-kit. U-boot initialise le port et configure la vitesse, ce qui explique qu'en activant les autres ports, ils ne soient pas utilisables immédiatement en 115200. N'oubliez pas également de changer le **bootargs** et en particulier la chaîne **console=ttyS2,115200n8** de manière à la faire correspondre au **ttyS0**. Il faut ensuite nous pencher sur la configuration du noyau en utilisant **ARCH=arm make menuconfig** dans le répertoire **uclinux** pour activer le port série qui nous intéresse. Chargez la configuration depuis **configs/kernel\_config**, procédez à vos modifications de la configuration et réenregistrez celle-ci dans ce même fichier afin qu'il puisse être appliqué lors de la prochaine compilation.

Mais ce n'est pas tout, pour la plateforme **PLATFORM\_STM32\_STM32429\_DISCO**, Robutest n'a configuré que le port **UART3** au niveau des E/S. Vous devrez éditer **arch/arm/mach-stm32/iomux.c** dans les sources du noyau pour ajouter :

```
#if defined(CONFIG_STM32_USART1)
    gpio_dsc.port = 0;
    gpio_dsc.pin = 10;
    stm32f2_gpio_config(&gpio_dsc, STM32F2_GPIO_ROLE_USART1);

    gpio_dsc.port = 0;
    gpio_dsc.pin = 9;
    stm32f2_gpio_config(&gpio_dsc, STM32F2_GPIO_ROLE_USART1);
#endif
```

Ceci se fait dans la définition de la fonction **stm32\_iomux\_init(void)**, dans le **switch/case, case PLATFORM\_STM32\_STM32429\_DISCO** : ajouté par Tnk/Robutest. Vous y trouverez là la configuration conditionnée par un **#if defined(CONFIG\_STM32\_USART3)**.

Enfin, c'est vers le système de fichiers racine qu'il faudra vous tourner et plus exactement vers le répertoire **rootfs/** contenant les éléments qui seront inclus, en plus du binaire BusyBox et de ses liens symboliques, au **romfs.bin**. C'est le fichier **rootfs/etc/inittab** qui nous intéresse, contenant :

```
ttyS2::respawn:/bin/login -f root
```

Vous devrez, bien entendu, remplacer **ttyS2** par **ttyS0**. Nous venons de changer des éléments du bootloader, du noyau et du système de fichiers racine. Vous êtes bon pour un **make clean && make...**

Tant que nous y sommes, nous pouvons corriger quelques imperfections dans la configuration initiale. Je pense en particulier au fichier **rootfs/etc/start** lancé via **inittab**. On y trouve, par exemple, le montage de sysfs dans **/sys** mais ce répertoire n'est pas dans **rootfs/**.