

MANUEL D'UTILISATEUR

SOMMAIRE

Note préliminaire.....	2
Configurer Linux du host.....	2
Installer gtkterm	2
Installation serveur TFTP	3
Brancher la carte APF28Dev	3
Paramétrer le réseau de la carte	4
Installer armadeus	4
Parameter le bus CAN.....	6
Créer le makefile.....	6
Coder, compiler et exécuter	7
Exécuter notre application au démarrage	8

NOTE PRÉLIMINAIRE

Dans les exemples suivants, lorsque l'ont a « \$ » en préfixe, cela signifie que l'ont exécute cette commande a partir d'un terminal du PC. Lorsque l'ont a « # » en préfixe, cela signifie que l'ont exécute cette commande a partir du terminal dans gtkterm. Et lorsque l'ont a « BIOS> » en préfixe, cela signifie que l'ont exécute cette commande dans le bios de la carte également dans gtkterm.

CONFIGURER LINUX DU HOST

Tout d'abord, il faut récupérer l'ensemble des librairies et patches nécessaire à l'ensemble des programmes utilisé par les outils de développement:

```
$ sudo apt-get install -y build-essential gcc g++ autoconf automake libtool
bison flex gettext
$ sudo apt-get install -y patch subversion texinfo wget git-core
$ sudo apt-get install -y libncurses5 libncurses5-dev
$ sudo apt-get install -y zlib1g-dev liblz2-2 liblz2-dev
$ sudo apt-get install -y libacl1 libacl1-dev gawk cvs curl lzma
$ sudo apt-get install -y uuid-dev mercurial
$ sudo apt-get install -y python-serial python-usb
$ sudo apt-get install -y libglib2.0-dev
$ sudo apt-get install -y libnetpbm10-dev
$ sudo apt-get install -y python-xcbgen
$ sudo apt-get install -y xutils-dev
```

INSTALLER GTKTERM

Pour avoir accès au terminal de la carte, il faut installer un logiciel permettant de communiquer via le port USB. Nous avons choisit d'utiliser gtkterm dans notre projet.

Installer gtkterm :

```
$ sudo apt-get install gtkterm
```

Il est recommander d'utiliser & pour lancer gtkterm afin de conserver le terminal pour une autre utilisation :

```
$ sudo gtkterm &
```

Une fois gtkterm aller sur Configuration –Port et sélection les paramètres suivants :

```
Port: /dev/ttyACM0      Speed:115200      Parity: None
Bits: 8                Stop bit: 1      Flow control: None
```

Le port peut être soit /dev/ttyUSB0 soit, dans notre cas, /dev/ttyACM0.

Sauvegarder cette configuration dans Configuration-Sauvegarder une Configuration sous le nom « Armadeus ».

Dorénavant, il suffira de charger la configuration « Armadeus » au lancement de gtkterm avec Configuration-Charger une Configuration.

Lorsque la carte est branchée, on obtiendra soit « BIOS> », soit « # »

Lorsque l'ont sur « BIOS> », le linux embarqué n'est pas démarrer. Pour démarrer le linux, entrer la commande suivante :

```
BIOS>boot
```

Lorsque l'ont est sur « # », le linux embarqué est démarrer. Pour revenir au BIOS, entrer la commande suivante :

```
# reboot
```

INSTALLATION SERVEUR TFTP

Récupérer la librairie nécessaire :

```
$ sudo apt-get install tftpd xinetd
```

Créer le dossier partagé a l'adresse voulue (ici /home/salleil/Armadeus) :

```
$ sudo mkdir /home/salleil/Armadeus/tftpboot  
$ sudo chmod 777 /home/salleil/Armadeus/tftpboot
```

Créer un fichier de configuration /etc/xinetd.d/tftp et la compléter avec le contenu suivant :

```
# default: off  
# description: The tftp server serves files using the trivial file transfer  
#               protocol. The tftp protocol is often used to boot diskless  
#               workstations, download configuration files to network-aware printers,  
#               and to start the installation process for some operating systems.  
service tftp  
{  
    socket_type          = dgram  
    protocol            = udp  
    wait                = yes  
    user                = root  
    server               = /usr/sbin/in.tftpd  
    server_args         = -s /home/salleil/Armadeus/tftpboot  
    #                   = yes  
}
```

Relancer le service xinetd :

```
$ sudo killall -HUP xinetd
```

BRANCHER LA CARTE APF28DEV

Pour communiquer avec la carte, on doit se brancher le port debug et on doit connecter la carte avec le pc via un câble Ethernet afin de pouvoir envoyer des fichiers sur la carte avec le serveur TFTP (cf. Figure 1).

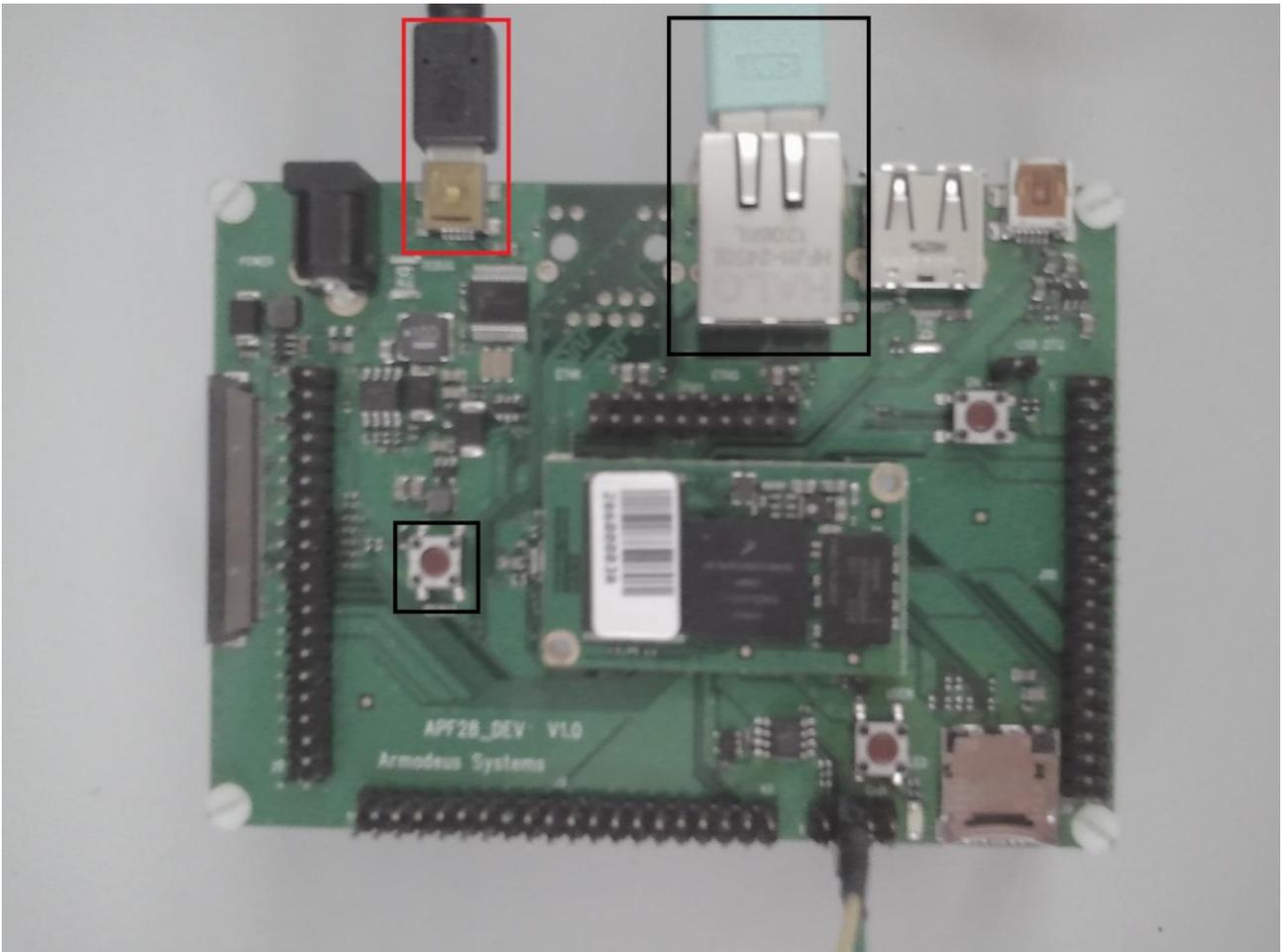


Figure 1: Branchement de la carte APF28Dev

Il n'est pas nécessaire d'alimenter la carte lorsque le port USB est branché.

PARAMETRER LE RESEAU DE LA CARTE

Il faut paramétrer les adresses réseau afin de pouvoir communiquer avec la carte via le port Ethernet. Dans notre cas, nous avons sélectionnés les variables suivantes :

```
BIOS> setenv netmask 255.255.255.0 # (masque réseau)
BIOS> setenv ipaddr 192.168.13.101 # (adresse IP de la cible)
BIOS> setenv serverip 130.120.13.13 # (adresse IP du serveur = votre PC)
```

La commande « printenv » permet d'afficher la plupart des informations de la carte.

INSTALLER ARMADEUS

Le kit de développement est disponible sur le lien suivant :

<http://sourceforge.net/projects/armadeus/files/armadeus/armadeus-5.3/armadeus-5.3.tar.bz2/download>

On obtiendra un dossier contenant l'ensemble des outils nécessaire pour le développement sur la carte.

Il faut ensuite se déplacer dans ce dossier et compiler

```
$ cd /home/salleil/Armadeus/armadeus-5.3/
$ make
```

La première compilation peut prendre un certain temps, environ 1 heure.

Une fois cette compilation terminée, exécuter :

```
$ make apf28_devconfig
```

Afin d'avoir accès au buildroot qui permettra de modifier les paramètres par défaut de la carte.

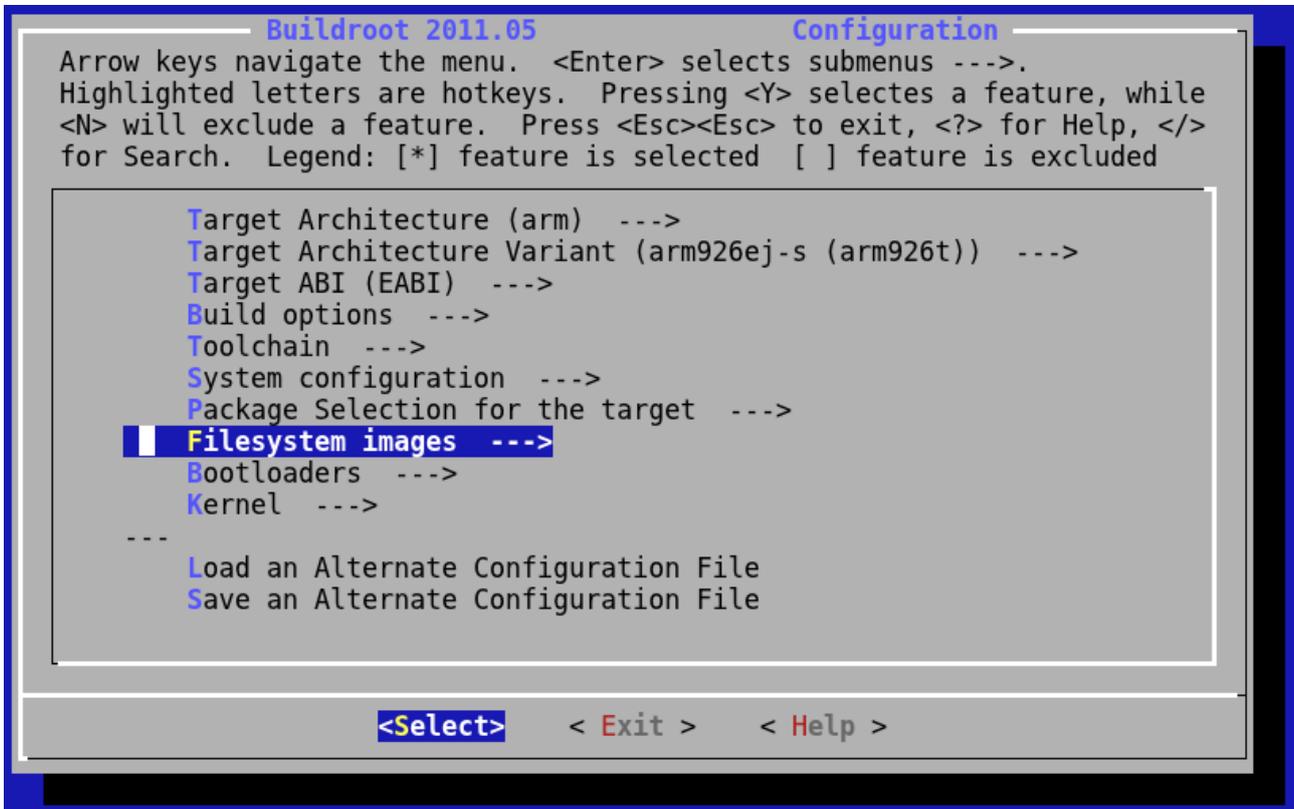


Figure 2: Menu du buildroot

On cherche :

```
System configuration --->
[*] Armadeus Device Support
```

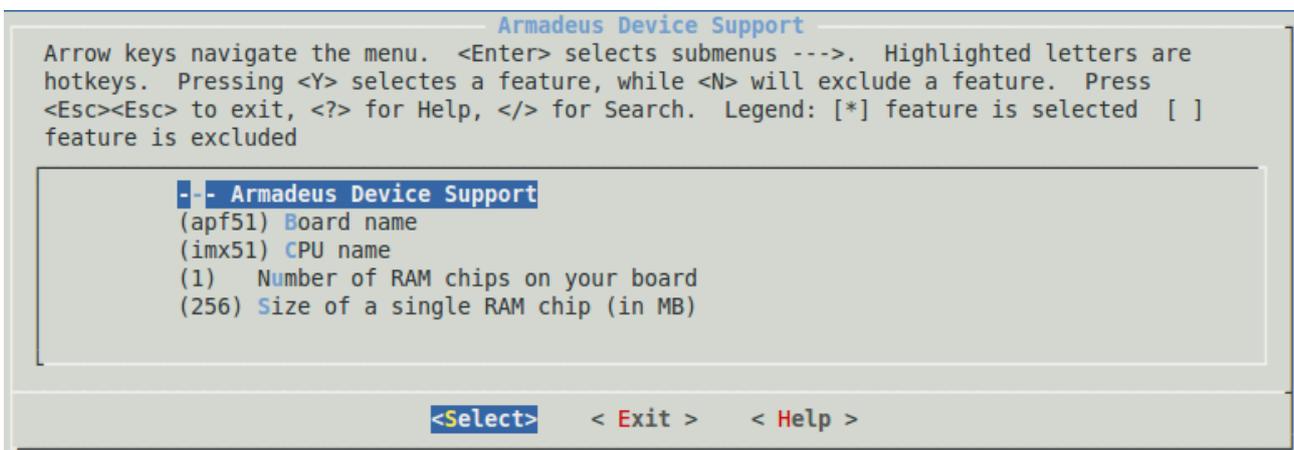


Figure 3: Paramètres à sélectionner

On vérifie la configuration donnée par la figure ci-dessus, ensuite, sauvegarder et quitter.

A chaque que l'ont fait des modifications sur ces paramètre, il est nécessaire de copier les fichiers produit sur le serveur TFTP :

```
$ cp /home/salleil/Armadeus/armadeus-5.3/buildroot/output/images/*  
/home/salleil/Armadeus/tftpboot/
```

Puis sur le BIOS de la carte, charger ces fichiers avec la commande suivante:

```
BIOS> run update_all
```

PARAMETER LE BUS CAN

Retourner sur le menu de buildroot avec la commande suivante:

```
$ make linux-menuconfig
```

Puis vérifier si les paramètres sont bien dans le mode suivant:

```
Networking support --->  
  <M>  CAN bus subsystem support --->  
      --- CAN bus subsystem support  
      <M>  Raw CAN Protocol (raw access with CAN-ID filtering)  
      <M>  Broadcast Manager CAN Protocol (with content filtering)  
      CAN Device Drivers --->  
          <M> Virtual Local CAN Interface (vcan)  
          <M> Platform CAN drivers with Netlink support  
          [*]  CAN bit-timing calculation  
          <M> Microchip 251x series SPI CAN Controller
```

Sauvegarder et quitter, puis compiler:

```
$ make linux  
$ make
```

Charger ces paramètres sur la carte.

Pour utiliser le CAN, il est nécessaire d'exécuter les commandes suivantes :

```
# modprobe can  
# modprobe can-dev  
# modprobe can-raw  
# modprobe flexcan
```

On peut fixer la vitesse de communication (ici 500kbps):

```
# set can0 up type can bitrate 500000
```

Pour démarrer le CAN, il faut exécuter la commande suivante:

```
# ifconfig can0 up
```

Pour l'arrêter, on fait :

```
# ifconfig can0 down
```

CREER LE MAKEFILE

Créer le makefile dans le dossier voulut et le compléter avec le contenu suivant :

```

#chemin du compilateur
CC=/home/salleil/Armadeus/armadeus-5.3/buildroot/output/host/usr/bin/arm-linux-
gcc
CFLAGS=-W -Wall
LDFLAGS=

EXEC=hello
#nom de l'exécutable
SRC=$(wildcard *.c)
OBJ=$(SRC:.c=.o)

all: $(EXEC)

$(EXEC): $(OBJ)

    $(CC) -o $@ $^ $(LDFLAGS)

%.o: %.c

    $(CC) -o $@ -c $< $(CFLAGS)

.PHONY: clean install

clean:

    rm -rf *.o
    rm -f $(EXEC)

install: all

    cp -f $(EXEC) /home/salleil/Armadeus/tftpboot/
#chemin du dossier partager par le serveur TFTP

```

Ce makefile compilera tout les fichiers du dossier dans lequel il est, créera un exécutable et le copiera dans le dossier partager par le serveur TFTP. Veiller a bien copier les librairie nécessaire pour utiliser les fonction de la carte Armadeus (GPIO, etc...).

Si vous copier ce makefile, faire attention aux tabulations.

Faire attention de ne pas avoir plusieurs main dans ce dossier.

CODER, COMPILER ET EXÉCUTER

Pour compiler, lancer la commande suivante :

```
$ make clean install
```

Puis sur le shell de la carte, récupérer l'exécutable créer :

```
# tftp -g -r hello -l /usr/bin/hello 130.120.13.13
# chmod a+x /usr/bin/hello
```

Et enfin exécuter le programme :

```
# /usr/bin/hello
```

EXÉCUTER NOTRE APPLICATION AU DÉMARRAGE

Créer un script nommé S99hello contenant les lignes suivantes :

```
/usr/bin/hello &  
exit 0
```

Ne pas oublier le '&', sinon la carte ne continuera pas de démarrer tant que l'ont n'aura pas terminé le programme.

La valeur 99 est l'ordre dans lequel les scripts s'exécuteront, ici, le script s'exécutera en dernier. Certains sont déjà existants sur la carte. Il est recommandé d'utiliser les valeurs les plus élevées afin de ne pas avoir de conflit avec d'autres scripts plus importants.

Copier ce script sur le serveur, puis récupérer le sur la carte:

```
# tftp -g -r S99hello -l /etc/init.d/S99hello 130.120.13.13  
# chmod a+x /etc/init.d/S99hello
```

Tester ce script :

```
# /etc/init.d/S99hello
```

Si tout fonctionne normalement, vous pouvez relancer la carte soit par la commande « reboot », soit en appuyant sur le bouton « reset » de la carte.