TP 2 Linux carte Fox Board LX832

http://www.acmesystems.it

Version 1.0 du 6 février 2008

sebphi@voila.fr

Table des matières

1 Introduction

L'objet de ce document est de permettre à toute personne désireuse de se familiariser avec un environnement Linux et en particuliers un noyau Linux embarqué sur une carte électronique de programmer ses fonctionnalités avancées en y ajoutant l'extension SX18.

Cette deuxième partie montre comment utiliser les périphériques USB pouvant être connecter. A titre d'exemple nous verrons l'utilisation d'une webcam, d'un périphérique Bluetooth et Wifi. Nous verrons également comment utiliser les fonctionnalités de la carte SX18 par plusieurs exemples. Parmi ceux qui seront traités: gestion des led, des boutons, d'un affichage LCD, de l'horloge temps réelle, et montage d'une carte MMC/SD.

2 Présentation

a - Objet

La finalité de ce document est d'apprendre à programmer sur une cible embarquée de type Fox Board LX 832 fabriquée par la société Acme System (http://www.acmesystems.it/).

il a été réalisé en grande partie grâce aux nombreuses articles et HowTo disponibles sur les sites suivants http://www.acmesystems.it/ et http://www.areasx.com/.Une grande partie du travail a consisté en la traduction des articles présents sur le site en y apportant un fil conducteur pour pouvoir l'utiliser sous la forme d'un TP. La finalité étant d'apporter l'expérience et la pratique qui ont permis de mieux interpréter et comprendre le mécanisme de cet environnement.

b - Présentation de la carte LX832

La carte Fox Board embarque un réel système d'exploitation Linux sur un processeur ETRAX 100LX. Ce processeur est basé sur une architecture RISC 100 MIPS fabriqué par la société AXIS (http://developer.axis.com).

La carte FOX Board possède 2 champs d'applications principaux:

* Comme un micro serveur Web ou tout autre application réseau comme un proxy, un router, un firewall, etc...
* Comme une interface intelligente destinée à remplacée un microprocesseur sur une carte électronique.

Elle est équipée de ports USB 1.1 qui pourront être connectée à des périphériques comme une clé mémoire USB, un disque dur, une webcam, un modem, un adaptateur WI-FI ou Bluetooth, un adaptateur ADSL, un convertisseur USB série, etc...

A travers une connection réseau 10/100 il est également possible d'accéder au serveur Web interne, serveur FTP, SSH, telnet et toute la pile TCP/IP.

2 connecteurs de 40 pins sont disponible pour fixer la carte sur n'importe quelle carte électronique spécifique ou pour connecter une carte d'interface spécifique et disponible chez Acme System. Sur ces connecteurs sont disponibles toutes sortes de signaux comme expliqué ci-dessous.

Un grand nombre de périphériques peuvent être connectés à la Fox Board en utilisant les signaux et adaptateurs hardware suivants:

* Jusqu'à 2 ports parallèles ou 1 port parallèle étendu (*)

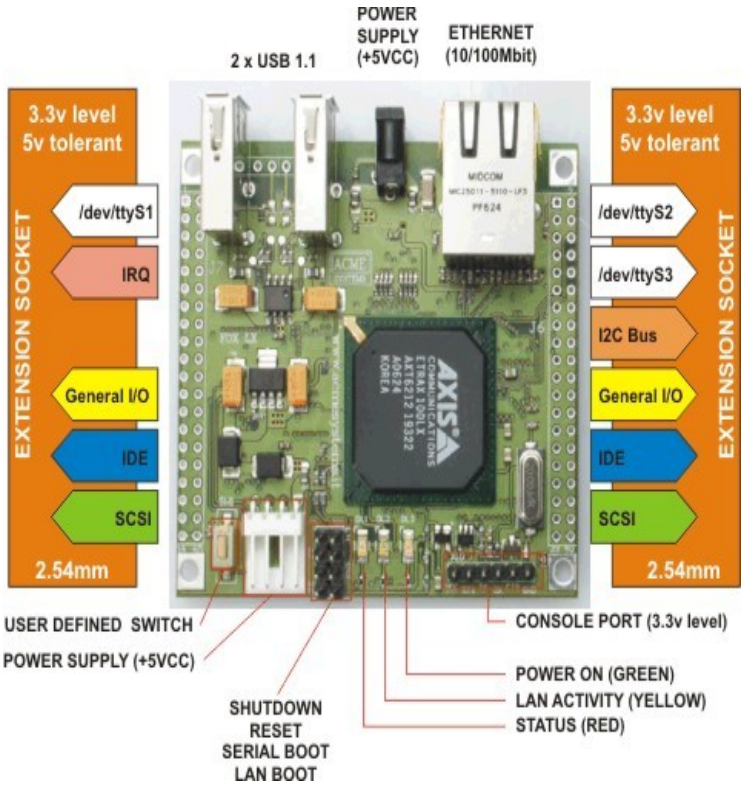
* Jusqu'à 4 ports IDE ports (8 périphériques)(*)

* Jusqu'à 2 ports SCSI ou 1 port SCSI étendu (*)

* Gestion du bus I2C

* Jusqu'à 48 lignes I/O (*)

* Jusqu'à 4 ports série asynchrones (*) (**)



Notes: Tous les signaux disponibles sur les connecteurs d'extension ont besoin d'une interface mécanique et/ou électrique pour être utilisés.

(*) Tous les signaux sur les connecteurs d'extension ne peuvent être utilisés en même temps. La configuration des périphériques signaux dépend du setup du noyau et des lignes assignées.

(**) Il existe 4 ports sur le processeur AXIS : 1 port asynchrone déjà utilisé pour la console (/dev/ttyS0) qui n'est pas accessible sur les connecteurs d'extension mais sur le connecteur J10; 1 autre port (/dev/ttyS1) qui est commun avec le port USB1, les 2 autres ports sont disponibles sur les connecteurs d'extension (/dev/tty2 et /dev/tty3). Un de ces ports peut être configuré aussi comme un port série RS485 ou un port série synchrone.

3 Installation du Software Developpement tool Kit

a - Matériel nécessaire

Se reporter au TP n°1.

b - Sous Linux Ubuntu

Se reporter au TP n°1.

c - Sous Windows XP

Se reporter au TP n°1.

4 Utilisation des ports USB 1 et 2 de la carte

La Fox Board possède deux connecteurs USB 1.1 ou il est possible de connecter des clés moires , des caméras numériques, des lecteurs mp3, des disques durs et tout sorte d'appareils USB fonctionnant comme une clé mémoire USB.
Nous allons voir dans ce chapitre comment en utiliser certains d'entre eux.

a Utilisation d'une clé mémoire USB

Pour vérifier et voir ce qu'il se passe lorsque l'on connecte une clé mémoire USB, taper la commande suivante dans une session telnet:

```
# tail -f /var/log/messages
```

De cette manière il est possible de voir les messages enregistrés par le noyau et vérifier donc que tout fonctionne correctement. Les messages qui devraient apparaître sont les suivants:

```
kernel: drivers/usb/host/hc_crisv10.c: USB controller in host mode.
kernel: drivers/usb/host/hc_crisv10.c: USB controller started.
kernel: drivers/usb/host/hc_crisv10.c: USB controller in host mode.
kernel: drivers/usb/host/hc_crisv10.c: USB controller started.
kernel: drivers/usb/host/hc_crisv10.c: USB controller running.
kernel: usb 1-2: new full speed USB device using ETRAX 100LX and address 2
kernel: scsi0 : SCSI emulation for USB Mass Storage devices
kernel:   Vendor: USB Mass  Model:  Storage Drive 2  Rev: 002
kernel:   Type:   Direct-Access                      ANSI SCSI revision: 02
kernel: SCSI device sda: 503808 512-byte hdwr sectors (258 MB)
kernel: sda: Write Protect is off
kernel: sda: assuming drive cache: write through
kernel: SCSI device sda: 503808 512-byte hdwr sectors (258 MB)
kernel: sda: Write Protect is off
kernel: sda: assuming drive cache: write through
kernel:  sda: sda1
kernel: Attached scsi removable disk sda at scsi0, channel 0, id 0, lun 0
```

A partir de ce message on peut voir que la clé mémoire USB a été correctement identifié par le noyau et assigné à un lien physique **/dev/sda**.

Pour pouvoir avoir un accès aux fichiers de cette mémoire , il est nécessaire de monter (**mount)** la partition **/dev/sda1** sur le répertoire **/mnt/1** en tapant ce qui suit:

```
# mount -t vfat /dev/sda1 /mnt/1
```

Il est maintenant possible d'avoir accès au répertoire du système de fichiers de la clé mémoire USB:

```
# cd /mnt/1
# ls
...
```

Pour voir quel espace libre est accessible sur la mémoire USB, taper:

```
# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/flash3               5032      5032         0 100% /
/dev/flash2                640       240       400  38% /mnt/flash
tmpfs                     6984        72      6912   1% /var
/dev/sda1                253156         0    253156   0% /mnt/1
```

Dans le cas présent, une clé mémoire de 256 MB est utilisée, pour l'enlever, il suffit de sortir du répertoire dus système de fichiers et taper:

```
 # cd /
# umount /mnt/1

# df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/flash3               5032      5032         0 100% /
/dev/flash2                640       240       400  38% /mnt/flash
tmpfs                     6984        72      6912   1% /var
```

Monter automatiquement une clé mémoire USB au démarrage du système:

Un répertoire spécial existe dans **/etc/init.d/boottime** ou l'on peut sauvegarder des scripts bash qui démarre au démarrage du système. Pour créer un script de montage automatique, saisir les commandes suivantes:

```
# cd /etc/init.d/boottime
# echo "mount –t vfat /dev/sda1 /mnt/1" > usbmount.sh
# cat usbmount.sh
mount –t vfat /dev/sda1 /mnt/1
# chmod +x usbmount.sh
```

Insérer maintenant une clé mémoire USB et démarrer le système.

Malheureusement la méthode **fstab** ne marche pas car au démarrage elle est gérée avant la reconnaissance des périphériques USB.

b Utilisation d'une clé USB Bluetooth (Gabriele Giottoli et Sergio Tanzilli)

Bluetooth est un protocole de communication radio fréquence. Cette section va expliquer comment utiliser un adaptateur USB Bluetooth avec la carte Fox Board et comment implémenter des services comme la recherche de périphériques ou l'envoi d'objets.

« Bluetooth est une spécification industrielle pour les réseaux sans fils personnels. Bluetooth apporte une manière de connecter et d'échanger des informations entre différents appareils, comme des téléphones mobiles, des ordinateurs portables ou de bureau, des imprimantes, des appareils photos numériques, ou

encore des consoles de jeux vidéo par l'intermédiaire d'une connection à courte distance radio fréquence sécurisée et sans licence d'exploitation » d'après une interprétation de la définition de (Wikipedia Bluetooth définition).

Démarrage rapide:

Pour essayer rapidement le protocole Bluetooth il est nécessaire de posséder:

- Un adaptateur Bluetooth USB (normalement tous les types sont supportés).
- Un appareil fonctionnant avec Bluetooth comme un téléphone mobile, un PDA, etc...

Télécharger une image prête a être utilisée pour le modèle de Fox Board possédé:

- LX816 fimage
- LX832 fimage

Puis reprogrammer la carte Fox bard (se référer au TP 1 section comment reprogrammer la carte).

Se connecter à la carte Fox via une connection Telnet et taper:

```
# cdBluez-start
Using /lib/modules/2.6.15/kernel/net/bluetooth/bluetooth.ko
Using /lib/modules/2.6.15/kernel/net/bluetooth/l2cap.ko
Using /lib/modules/2.6.15/kernel/net/bluetooth/rfcomm/rfcomm.ko
Using /lib/modules/2.6.15/kernel/drivers/bluetooth/hci_usb.ko
```

Grâce à cette commande, les modules Bluetooth vont se charger. Allumer un appareil fonctionnant avec Bluetooth et activer une recherche de périphériques en tapant:

```
# hcitool scan
```

Si tout fonctionne correctement, une liste d'appareils équipés du Bluetooth devrait apparaître.

```
Scanning ...
        00:16:20:8F:BA:60       Z520i
        00:12:56:C2:88:17       Sergio
        08:00:46:EA:CA:12       NOME-YOMKTFIKO9
```

Pour trouver des appareils qui autorisent le service OPUSH (envoi de données) taper:

```
# sdptool search OPUSH
Inquiring ...
Searching for OPUSH on 00:16:20:8F:BA:60 ...
Service Name: OBEX Object Push
Service RecHandle: 0x10005
Service Class ID List:
  "OBEX Object Push" (0x1105)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 5
  "OBEX" (0x0008)
Profile Descriptor List:
  "OBEX Object Push" (0x1105)
    Version: 0x0100
```

Essayer maintenant d'envoyer une image à l'appareil Bluetooth sélectionné. Par exemple envoyer l'image du pingouin Linux TUX.

Si la Fox Board est connectée à internet, il est possible de récupérer directement l'image à partir du site d'Acme Systems en tapant:

```
# cd /mnt/flash
# wget http://80.241.169.67/articles/00112/tuxcase.jpg
Connecting to 80.241.169.67[80.241.169.67]:80
tuxcase.jpg          100% |*****************************|  4710        00:00 ETA
```

Puis pour envoyer l'image sur un téléphone portable par exemple taper:

```
# ussp-push 00:16:20:8F:BA:60@5 tuxcase.jpg tuxcase.jpg
name=tuxcase.jpg, size=4710
Local device 00:02:72:81:4A:EC
Remote device 00:16:20:8F:BA:60 (5)
Connection established
```

Si tout se passe bien l'image du TUX devrait être stockée dans la mémoire du téléphone:

Le format de la commande ussp-push est la suivante:

```
# ussp-push addressdevice@channel localfile remotefile
```

La librairie BlueZ

La pile Bluetooth implémente plusieurs couches:

1. **La couche de communication Radio**. Implémentées pour les couches physiques bas niveau.
2. **La couche de management de la liaison**, est utilisée pour connecter physiquement les appareils entre eux.
3. **La couche HCI** (Host Controller Interface), est utilisée pour relier physiquement l'appareil au système d'exploitation.
4. **La couche L2Cap**. Cette couche de contrôle de lien logique et d'adaptation de protocole relie numériquement les appareils.
5. **La couche application**. Cette couche est utilisée pour communiquer avec les applications en utilisant différents profiles.

Quelques implémentation de la cinquième couche sont:

- **RFComm**: Émulation d'une communication série.
- **OBEX**: Transfert et gestion des fichiers.
- **SDP**: Connection aux périphériques et demande d'informations sur ceux-ci.
- **BNEP** (Bluetooth Network Encapsulation Protocol): Pour encapsuler la couche TCP/IP à travers le Bluetooth.

La pile BlueZ est le protocole Bluetooth officielle pour Linux (www.bluez.org). Elle apporte des librairies pour interfacer les application avec les services et utilitaires Bluetooth .

OpenOBEX

« OpenOBEX est logiciel libre pour l'implémentation du protocole *(OBEX). OBEX est un protocole de session et la meilleure description peut être un protocole HHTP binaire. OBEX est optimisé pour les connections sans fils ad-hoc et peut être utilisé pour échanger toutes sortes d'objets comme des fichiers, des images, des entrées pour le calendrier (vCal) , et des cartes professionnelles (vCard)*" (définition provenant du site http://openobex.triq.net)

ussp-push

Ussp-push est un objet logiciel d'envoi pour Linux utilisant la pile Bluetooth BlueZ. Avec cet utilitaire il est possible d'envoyer un fichier à distance à un appareil Bluetooth.

Comment paramétrer le support du Bluetooth grâce au SDK ?

A l'intérieur du répertoire principal du SDK taper:

```
# make menuconfig
```

Puis autoriser le support du Bluetooth comme montré ci-dessous:

```
Driver settings
  Bluetooth BlueZ libraries and tools
                [*] OpenObex
                [*] ussp-push
```

Puis taper save, exit et taper:

```
# ./configure
...
# make
```

Finalement reprogrammer la carte avec la nouvelle image (fimage).

c Utilisation d'une webcam USB

Cette section va expliquer comment connecter une webcam à la Fox Board et pouvoir ainsi visualiser le flux vidéo à distance sur un ordinateur Windows ou Linux via une connection ethernet.
La webcam utilisée est une Labtec Pro USB qui est capable de gérer un flux vidéo de 30 images secondes.

Note: Il reste toujours un problème non résolu sur le driver WIFI ou USB qui cause un mauvais fonctionnement lorsque le flux de la webcam passe à travers un adaptateur WIFI D-Link USB.

Démarrage rapide:

Pour démarrer rapidement il suffit de reprogrammer la carte Fox avec l'image suivante:

- fimageLX832.zip

Brancher la webcam sur le port USB de la carte Fox et se connecter à la carte grâce à une session Telnet à

l'adresse par défaut **192.168.0.90**.

Après la connection taper:

```
[root@axis /root]106# spcaload
Using /lib/modules/2.4.31/kernel/drivers/media/video/videodev.o
Using /lib/modules/2.4.31/kernel/drivers/usb/spca5xx/spca5xx.o
```

Puis taper:

```
[root@axis /root]106# dmesg
...
...
Linux video capture interface: v1.00
usb.c: registered new driver spca5xx
spca_core.c: USB SPCA5XX camera found. Type Labtec Webcam Pro Zc0302 + Hdcs2020
spca_core.c: spca5xx driver 00.57.06LE registered
```

La LED erte de la webcam devrait maintenant s'allumer. Cela indique qu'elle est prête à être utilisée.

Démarrer le serveur de flux vidéo en tapant:

```
[root@axis /root]106# servfox -d /dev/video0 -s 640x480 -w 7070
servfox version: 1.1.3 date: 11:12:2005 (C) mxhaard@magic.fr
wrong spca5xx device
Waiting .... for connection. CTrl_c to stop !!!!
```

Le message "wrong spca5xx device" apparaît mais la Fox Board est maintenant prête à envoyer le flux vidéo sur un ou plusieurs client sur le port 7070.

Démarrer le client de lecture du flux vidéo:

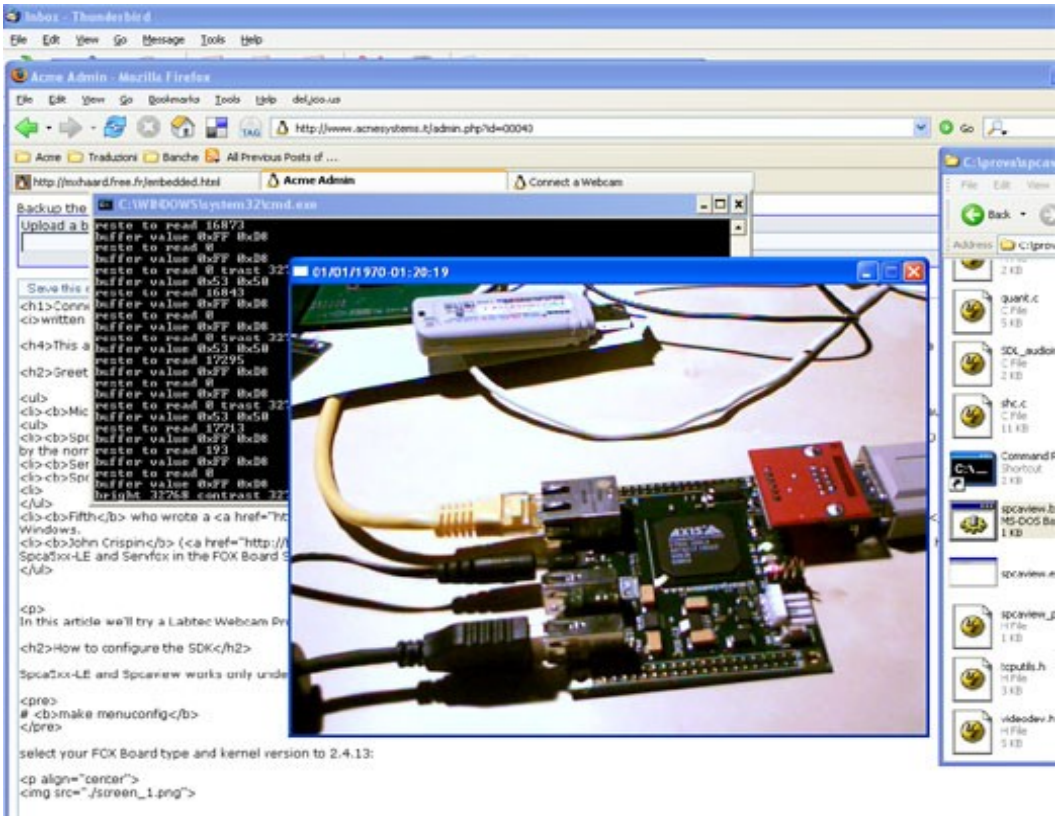
Spcaview est un client de lecture de flux vidéo disponible pour Windows et Linux. Pour les utilisateurs de Windows, il est possible de télécharger le code source et le fichier binaire à partir de ce lien:

- Windows version of Spcaview

Décompresser le fichier archive et éditer le fichier **spcaview.bat** comme indiqué ci-dessous:

```
spcaview -w 192.168.0.90:7070 -s 640x480
```

Double cliquer sur le fichier .bat et le flux vidéo devrait arriver de la Fox Board.



Comment paramétrer le support de la webcam grâce au SDK ?

A l'intérieur du répertoire principal du SDK taper:

```
# make menuconfig
```

Sélectionner l'élément **Driver settings** --->et valider les éléments suivants:

```
[*] Enable Webcam support
[*]  Webcam tools - Michel Xhaard (servfox & co)
```

Cliquer sur Save, exit et taper:

```
# ./configure
...
# make
...
```

Finalement reprogrammer la carte avec la nouvelle image (fimage).

d Utilisation d'une clé WIFI USB

Cette section va expliquer comment connecter une clé Wifi à la Fox Board et pouvoir ainsi communiquer à distance sur un ordinateur Windows, Linux ou un boîtier ADSL via une connection ethernet sans fils.

DWL-G122 Product info

Note: Malheureusement le driver USB AXIS présent sur la version 2.6.15 du noyau et la version C1 du driver Wifi ne peuvent fonctionner ensemble avec des flux importants de données. Ainsi par exemple, il est impossible pour le moment d'envoyer le flux vidéo d'une webcam Labtec (voir session précédente).

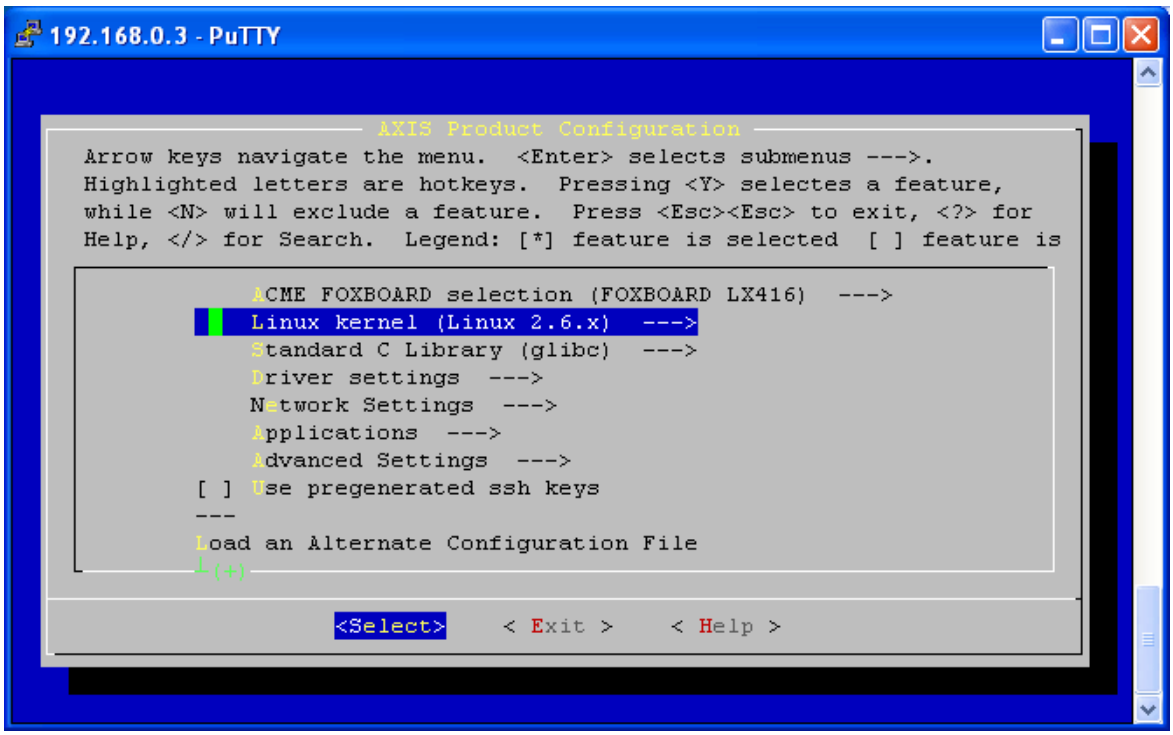
Les adaptateurs Wifi B1 semblent fonctionner correctement. Pour les utilisateurs de la version C1 de l'adaptateur Wifi, *Sergio Tanzilli* travaille sur une solution basée sur le nouveau SDK de AXIS version 2.10. Tout aide sera appréciée par ce dernier.

Comment générer une image supportant le Wifi avec le SDK ?

A l'intérieur du répertoire principal du SDK taper:

```
/home/fox/devboard-R2_01# make menuconfig
```

Sélectionner le type de carte utilisée, ensuite sélectionner la version du noyau 2.6.15:



Sélectionner l'élément **Driver settings** ---> et sélectionner :

```
[*] Enable WLAN support
```

Sélectionner le nouvel élément **Wireless networking (WLAN)** ---> et sélectionner:

```
[*] wireless tools
```

Sélectionner maintenant l'onglet **DLink DWL-G122 (None)** ---> et autoriser l'adaptateur B1 ou C1 (Se référer à la version du firmware indiqué sur l'adaptateur).

Sauvegarder, sortir et taper:

```
/home/fox/devboard-R2_01# ./configure
...
/home/fox/devboard-R2_01# make
...
```

Maintenant il est possible de reprogrammer la carte avec la nouvelle image 'fimage'. Voir le TP n°1 pour connaître les différents mode reprogrammation.

Comment test la fonctionnalité WiFi?

Connecter la carte Fox sur le réseau par le port ethernet et se connecter sur l'adresse ip par défaut **192.168.0.90**.
Une fois connecté pour voir les messages du système taper:

```
# tail -f /var/log/message
```

Insérer ensuite l'adaptateur Wifi. Les messages suivants devraient apparaître:

```
crisv10_irq dbg: ctr_status_irq, controller status: host_mode started
crisv10_irq dbg: ctr_status_irq, controller status: host_mode started running
usb 1-2: new full speed USB device using hc-crisv10 and address 5
idVendor = 0x7d1, idProduct = 0x3c03
```

Pour sortir de la commande **tail** appuyer sur **CTRL-C** et taper:

```
# /etc/init.d/wireless start
 * Bringing up rausb0...
    + IP: 192.168.10.90
    + WEP: off
    + Channel: 1
    + ESSID: FOX
    + Mode: ad-hoc
```

L'adaptateur Wifi va maintenant fonctionner avec l'adresse IP par défaut **192.168.10.90**.
Il est possible de voir la nouvelle configuration TCP/IP en tapant:

```
# ifconfig rausb0
rausb0    Link encap:Ethernet  HWaddr 00:17:9A:D0:12:EE
          inet addr:192.168.10.90  Bcast:192.168.10.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:102 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:8034 (7.8 KiB)
```

Pour voir les paramètres de la configuration sans fils:

```
# iwconfig
Warning: Driver for device rausb0 has been compiled with version 19
of Wireless Extension, while this program supports up to version 17.
Some things may be broken...

rausb0    RT73 WLAN  ESSID:""
          Mode:Ad-Hoc  Frequency=1 MHz  Cell: D6:48:6D:35:23:32
          Bit Rate=11 Mb/s
          RTS thr:off   Fragment thr:off
          Encryption key:off
          Link Quality=0/100  Signal level:-121 dBm  Noise level:-115 dBm
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Comment configurer l'adaptateur Wifi?

Le fichier de configuration d'origine de l'adaptateur chargé au démarrage est **/etc/conf.d/net. wireless**.
Voyons son contenu:

```
DEV="rausb0"
```

rausb est le nom du périphérique pour l'adaptateur WiFi.

```
IP="192.168.10.90"
```

```
SUBNET="255.255.255.0"
```

IP est l'adresse attribuée par défaut à l'adaptateur.
SUBNET est le masque de sous-réseau du réseau internet.

```
ESSID="FOX"
```

ESSID pour **E**xtended **S**ervice **S**et **ID.** Cette variable identifie le nom du réseau sans fils. En spécifiant cette variable, il devient plus commode de se connecter à son réseau et non à celui de ses voisins par erreur.

```
WIFI_DEFAULT="0"
WIFI_GATEWAY=""
```

WIFI-DEFAULT indique si l'adaptateur Wifi est la gateway par défaut, et **WIFI_GATEWAY** indique l'adresse IP de la gateway du réseau.

```
CHANNEL="1"
```

CHANNEL représente le canal qui est utilisé.

```
MODE="ad-hoc"
```

Positionner **MODE="ad-hoc"** pour contacter un autre hôte (par exemple une autre carte Fox Board) sans utiliser de point d'accès.
Positionner **MODE="managed"** pour relier une carte FOX Board à un point d'accès.

```
KEY="off"
```

Ce paramètre correspond à la clé d'encryptage WEP. Quand il est égal à "off" aucun encryptage n'est utilisé.

Une fois les paramètres modifiés, il est nécessaire de redémarrer l'interface sans fils en exécutant les commandes suivantes:

```
# /etc/init.d/wireless stop
# /etc/init.d/wireless start
```

L'interface sans fil redémarrera avec les nouveaux paramètres.

e Utilisation d'un convertisseur USB vers Série RS232

Dans cette partie sera expliqué comment connecter et configurer une communication asynchrone sur le port série RS232 de la carte Fox Board.

Ports séries asynchrones de la carte Fox.

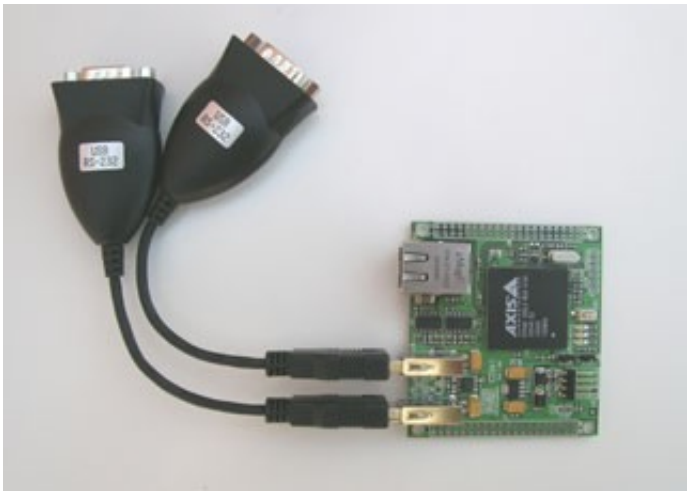
Le composant ETRAX LX100 contient un total de quatre ports séries asynchrones mappés dans le noyau de la manière suivante:

- **/dev/ttyS0** est le port configuré par défaut comme le port console extérieure. Ce port est physiquement accessible sur le connecteur J10.
- **/dev/ttyS1** Les signaux de ce port sont partagés avec les signaux du port USB1. Pour utiliser ce port série, il est obligatoire de désactiver le support de l'USB1 dans le noyau et de le recompiler. Ces signaux sont physiquement accessibles sur le connecteur d'extension J7.
- **/dev/ttyS2** Ce port est complètement accessible. Les signaux sont accessibles sur le connecteur d'extension J6. Comme tous le signaux de contrôle de ce port sont accessibles, il est le plus adapté pour se connecter à un modem.
- **/dev/ttyS3** Ce port est complètement accessible. Les signaux sont accessibles sur le connecteur d'extension J6.

Tous ces signaux fonctionne avec un niveau de tension de **3.3Volt** mais ils tolèrent également des niveaux de tension de **5Volt**.

Port séries asynchrones externes:

Grâce à la présence des deux ports USB (mode hôte) il est possible d'étendre les ports séries internes de la carte à l'aide de deux convertisseurs USB/RS232 à bas coût. A cela on peut rajouter 2 Hubs USB avec chacun 7 ports, et l'on obtient un total de 14 ports série accessible sur la carte Fox.



L'image du noyau délivrée par défaut sur la Fox supporte tous les convertisseurs commerciaux USB/RS232 qui utilisent les composants **FTDI** (http://www.ftdichip.com) ou PROLIFIC 2303.

Quand un convertisseur est inséré sur un des 2 ports USB, le noyau reconnaît le port comme **/dev/ttyUSBx** où x est nombre incrémenté automatiquement de 0 à n.

Pour vérifier si le convertisseur a bien été reconnu il suffit taper la commande suivant:

```
tail –f /var/log/messages
```

Au moment ou le convertisseur est inséré, le message suivant apparaît:

```
Mar 16 12:00:24 Fox kernel: usb-host.c: USB controller running.
Mar 16 12:00:24 Fox kernel: hub.c: new USB device ETRAX 100LX-1, assigned address
3
Mar 16 12:00:24 Fox kernel: usbserial.c: FTDI 8U232AM Compatible converter
detected
Mar 16 12:00:24 Fox kernel: usbserial.c: FTDI 8U232AM Compatible converter
now attached to ttyUSB0 (or usb/tts/0 for devfs)
```

Dans le dernier message est assigné le nom du périphérique correspondant au convertisseur. Dans le cas présent c'est **ttyUSB0**.

Si un deuxième convertisseur est inséré sur le deuxième port USB de la carte Fox, le message suivant apparaîtra:

```
Mar 16 12:02:34 Fox kernel: hub.c: new USB device ETRAX 100LX-2, assigned address
4
Mar 16 12:02:34 Fox kernel: usbserial.c: FTDI 8U232AM Compatible converter
detected
Mar 16 12:02:34 Fox kernel: usbserial.c: FTDI 8U232AM Compatible converter
now attached to ttyUSB1 (or usb/tts/1 for devfs)
```

Lorsque la carte Fox est allumée avec les convertisseurs USB/série déjà connectés, l'association entre les périphériques et les convertisseurs se fera grâce à la liste suivante:

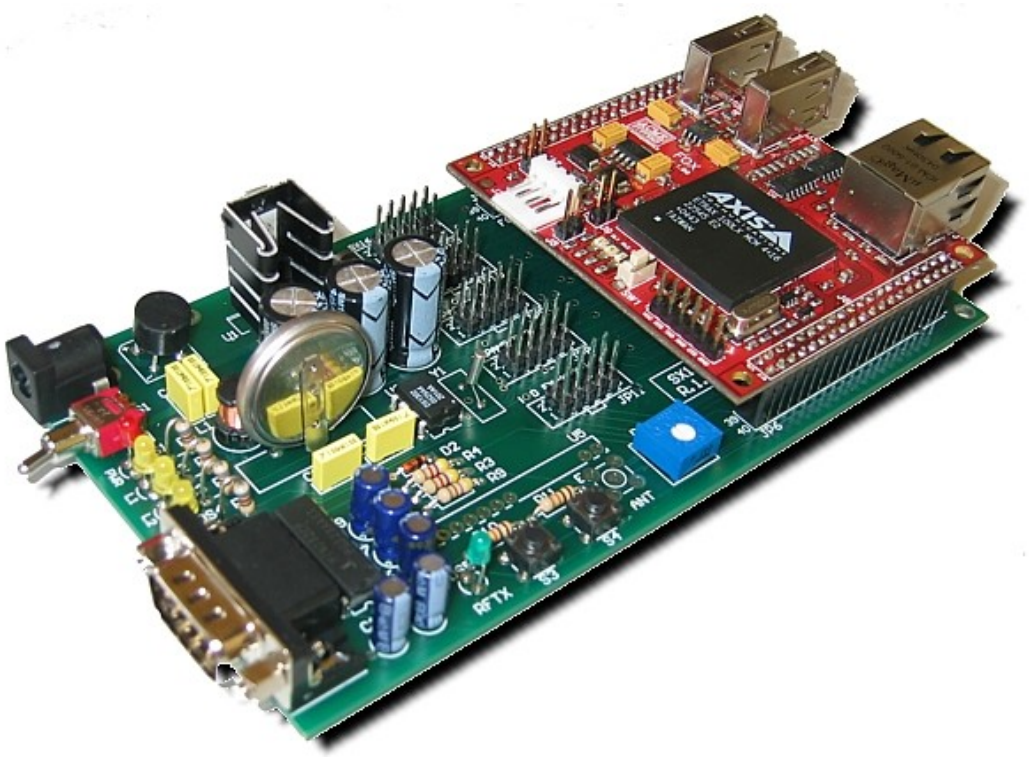
- **/dev/ttyUSB0** assigné au convertisseur connecté sur USB1.
- **/dev/ttyUSB1** assigné au convertisseur connecté sur USB2.

5 Utilisation des fonctions de la carte sx18

Dans cette section sera expliqué comment se servir de la carte SX18 et de ses multiples ressources grâce aux schémas fournis sur le site (http://www.areasx.com) ainsi que le code source fourni à chaque nouvelle partie.

a Présentation de la carte sx18

La carte SX18 est spécialement conçue pour être utilisée avec la carte Fox, et n'a besoin d'aucun éléments extérieurs pour l'alimentation des périphériques ou de la carte Fox.



Avec des dimensions de 150 x 80 mm cette carte comprend:

- 1 port série RS232 placé sur la face de connection
- 1 connecteur pour un module radio ER400TRS radio (Pour plus d'informations voir: Trasmissioni in Radiofrequenza facili con Easy Radio)
- 1 composant DS1302 avec une batterie de sauvegarde au Lithium pour obtenir une horloge temps réel.
- 2 boutons poussoirs.
- 3 led pour une utilisation générale.
- 1 led indiquant la présence de la tension d'alimentation.
- 3 connecteurs d'extension 5+5 pour une utilisation générale.
- 1 connecteur d'extension utilisable pour les entrées/sorties de la carte SX16 (Pour plus d'information:

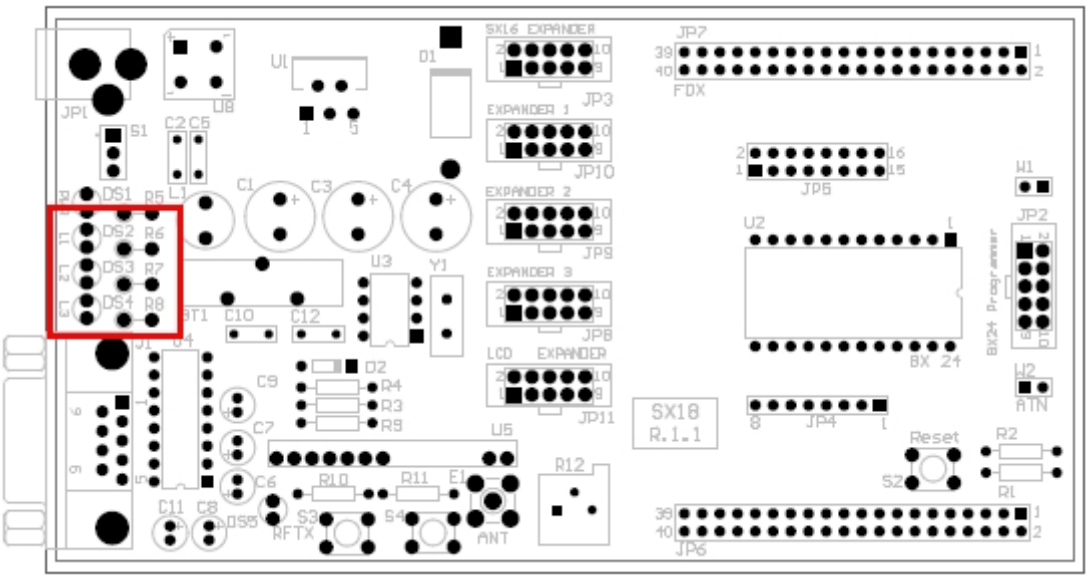
SX16B - IN/OUT expansion board)

- 1 connecteur d'extension destiné uniquement à la carte SX13 (Pour plus d'information: Multifunctions dimmer).
- 1 connecteur d'extension destiné à contrôler un afficheur LCD.

Pour visualiser le schéma électrique, se rendre à la section suivante: SX18_Schematic.pdf

b Test des led

Sur la carte SX18 se trouvent 3 Led libre d'utilisation: DS2, DS3 et DS4 sont connectées respectivement sur les pins 13 (OG25), 21 (OG0) et 38(PA0). Ces pins se trouvent sur le connecteur J7.



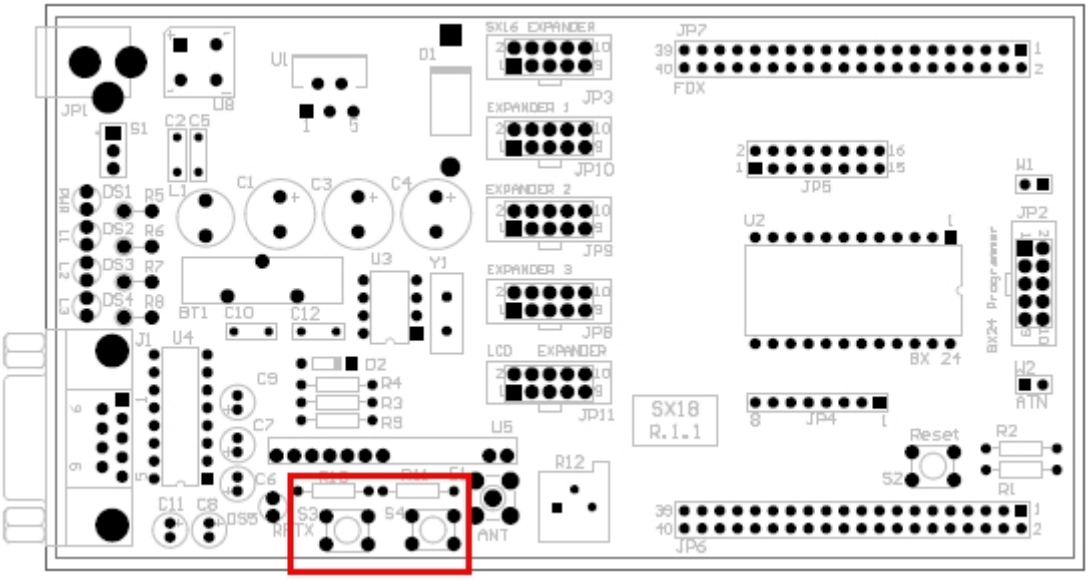
Ci-dessous se trouve un code source simple pour les tester: Initialisation des ports puis clignotement des Led une après l'autre.

```
#include "stdio.h"
#include "unistd.h"
#include "sys/ioctl.h"
#include "fcntl.h"
#include "time.h"
#include "asm/etraxgpio.h"

#define LED_1  1<<25
#define LED_2  1<<24
#define LED_3  1<<0
#define IO_SETGET_OUTPUT  0x13
```

```
int main(void) {
 int fda;
 int fdg;
 int iomask_a;
 int iomask_b;
 int iomask_g;
 int i;

 if ((fda = open("/dev/gpioa", O_RDWR))<0) {
  printf("ERROR opening /dev/gpioa\n");
  return 1;
 }
 if ((fdg = open("/dev/gpiog", O_RDWR))<0) {
   printf("ERROR opening /dev/gpiog\n");
   return 1;
 }
 iomask_a = LED_3;
 ioctl(fda, _IO(ETRAXGPIO_IOCTYPE, IO_SETGET_OUTPUT),&iomask_a);
 iomask_g = LED_1 | LED_2;
 ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_SETGET_OUTPUT),&iomask_g);

while (1) {
   printf("Test LED\n");
   ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_SETBITS),LED_1);  //LED 1 ON
   sleep(1);
   ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_CLRBITS),LED_1);  //LED 1 OFF
   sleep(1);
   ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_SETBITS),LED_2);
   sleep(1);
   ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_CLRBITS),LED_2);
   sleep(1);
   ioctl(fda, _IO(ETRAXGPIO_IOCTYPE, IO_SETBITS),LED_3);
   sleep(1);
   ioctl(fda, _IO(ETRAXGPIO_IOCTYPE, IO_CLRBITS),LED_3);
   sleep(1);
 }
 close(fda);
 close(fdg);
 return 0;
}
```

c Test des boutons poussoirs

Deux boutons poussoirs S3 et S4 sont connectés sur les pins 29 (IG2) et 28 (IG3) disponibles sur le connecteur J6.



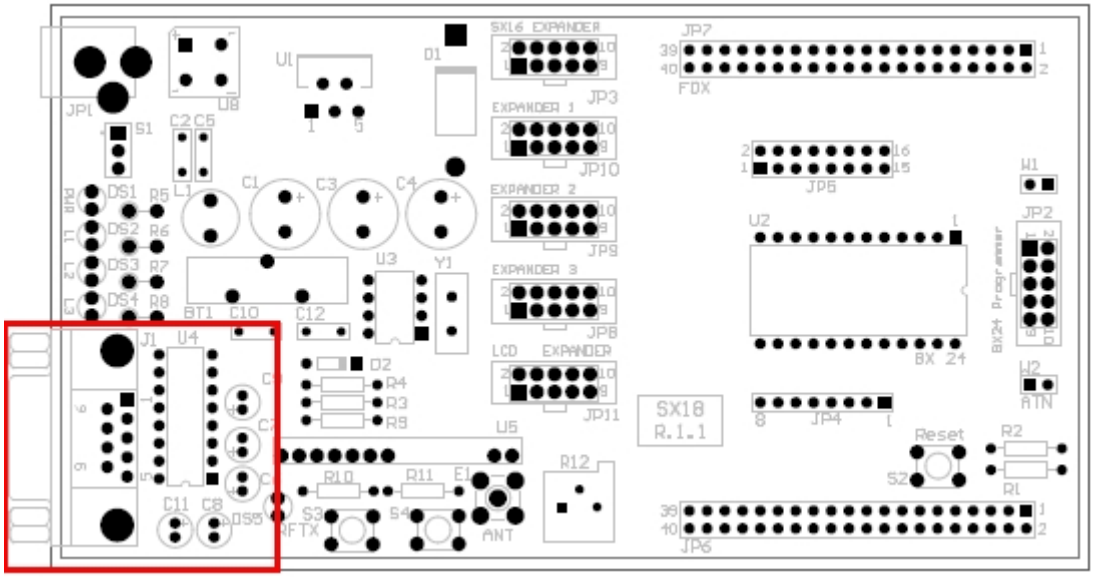
Ci-dessous se trouve un code source simple pour les tester: A chaque fois qu'un des deux boutons est pressé, un message sera affiché sur la console indiquant le nom du bouton poussoir pressé.

```
#include "stdio.h"
#include "unistd.h"
#include "sys/ioctl.h"
#include "fcntl.h"
#include "time.h"
#include "asm/etraxgpio.h"

#define BUTTON_1  1<<2
#define BUTTON_2  1<<3
#define IO_SETGET_INPUT   0x12

int main(void) {
  int fdg;
  int iomask_g;
  unsigned char value = 0;

  if ((fdg = open("/dev/gpiog", O_RDWR))<0) {
    printf("ERROR opening /dev/gpiog\n");
    return 1;
  }
```

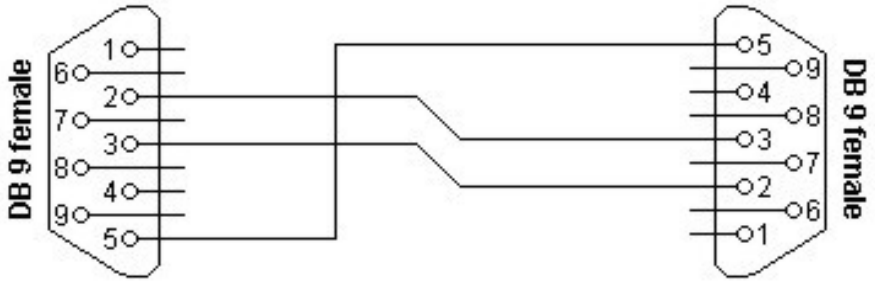
```
 iomask_g = BUTTON_1 | BUTTON_2;
 ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_SETGET_INPUT),&iomask_g);

 while(1) {
   value=ioctl(fdg, _IO(ETRAXGPIO_IOCTYPE, IO_READBITS));
   if ((value&(BUTTON_2))==0)
     printf("Button 1 pressed\n");
   if ((value&(BUTTON_1))==0)
     printf("Button 2 pressed\n");
 }
}
```

d Test du port série

Sur la carte SX18, est implanté une section spéciale pour la communication RS232 avec un composant MAX232 (composant intégrant les niveaux de tension pour la transformation TTL/CMOS vers RS232) et un connecteur mâle 9 points. Mise à part les signaux traditionnel RX et TX, sur le connecteur sont aussi disponibles les signaux de contrôle de flux RTS et CTS. La section RS232 est connectée sur le connecteur J6 pins 3 (RTS), 4 (RX), 5 (TX) et 6 (CTS) de la Fox Board et est identifiée par le noyau comme **/dev/ttyS3**.



Pour lancer les tests sur le port série, il est nécessaire de connecter la carte SX18 à l'ordinateur grâce à un câble série croisé. Pour réaliser ce câble, il est possible de suivre le schéma ci-dessous:



Ce type de câble est nécessaire car la paire de carte FOX-SX18 est configurée comme un DTE (Data Terminal Equipment), ce type de communication est souvent celle utilisée lors d'une connection série.

L'exemple de programme présenté ci-dessous est simple ECHO qui retourne exactement ce qui lui a été transmis. Le code source présenté est un exemple qui explique comment établir une communication simple RS232 avec les ports séries de la carte Fox. Dans la cas présent c'est ttyS3.

```
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <termios.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <signal.h>
#include <stdlib.h>

#define BAUDRATE B19200
#define DEVICE "/dev/ttyS3"

int main (int argc, char * argv[]) {
 int fd, res;
 struct termios oldtio,newtio;
 char buf[200];
 printf("Opening Port COM\n");
 fd = open(DEVICE, O_RDWR | O_NOCTTY);
 if (fd < 0 ) {
  printf("Device %s unavailable on this system\n\n", DEVICE);
  exit(-1);
 }
 printf("Serial communication initialization\n");

 tcgetattr(fd,&oldtio); /* Save previous settings */

 bzero(&newtio, sizeof(newtio));
 newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;
 newtio.c_iflag = IGNPAR;
```

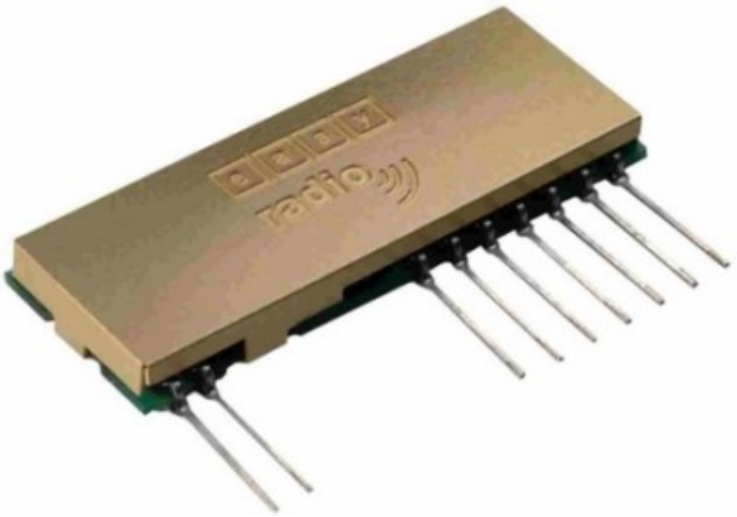
```
 newtio.c_oflag = 0;
 newtio.c_lflag = 0;
 newtio.c_cc[VTIME] = 0;
 newtio.c_cc[VMIN] = 1;

 tcflush(fd, TCIFLUSH);
 tcsetattr(fd,TCSANOW,&newtio);
 //Serial device initialization finished
 sprintf(buf, "Hello, I'm FOX board equipped with SX18 :)\n\r Write, I'm listening (Type Q to quit)\n\r");
 if (write(fd, buf, strlen(buf)) < 0) {
  printf("DATA SENDING ON SERIAL PORT FAILED\n");
 }
 while(1){
  res = read(fd,buf,sizeof(buf));
  buf[res]=0;
  if (write(fd, buf, strlen(buf)) < 0) {
   printf("DATA SENDING ON SERIAL PORT FAILED\n");
  }
  if (buf[0]=='Q') break;
 }
 printf("\nClosing serial commuenication...\n");
 tcsetattr(fd,TCSANOW,&oldtio);
 printf("Bye :)\n");
 close(fd);
 exit(0);
}
```

Afin de tester la communication il est nécessaire de: connecter l'ensemble FOX SX18 à l'ordinateur en utilisant un câble croisé (comme décrit ci-dessus); Ouvrir une console série sur l'ordinateur (par exemple Hyperterminal ou Minicom) et une autre console SSH sur la FOX pour lancer l'application TEST_SERIALE.C. OUT (fichier obtenu par la compilation du source ci-dessus) précédemment télécharger dans la mémoire de la Fox.

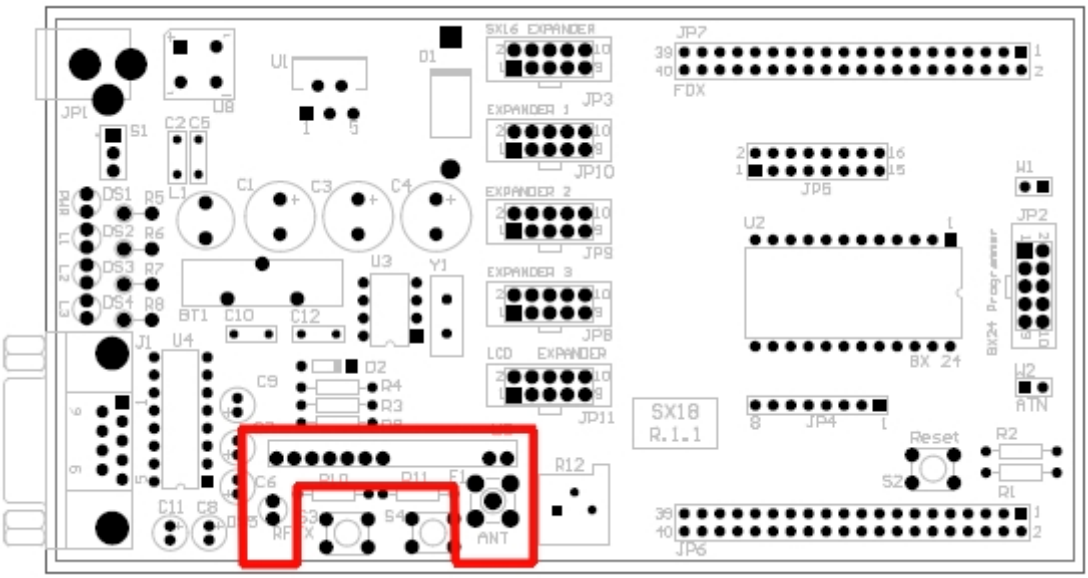
Maintenant toutes les données tapées dans la console série apparaissent dans celle de la Fox et inversement.

e Test de l'interface RF

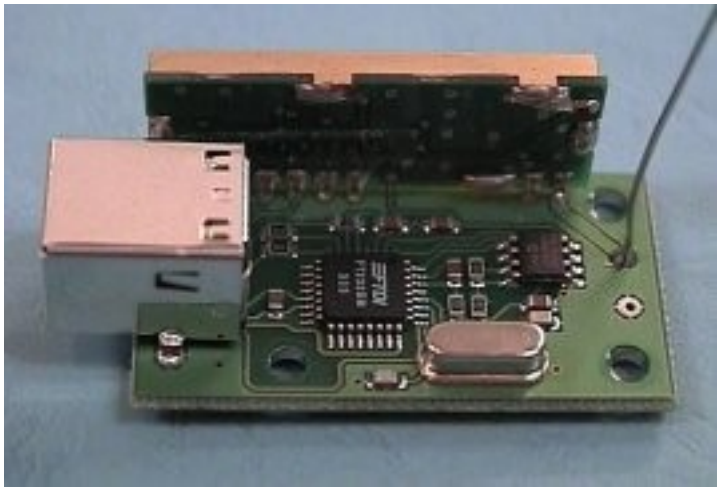
La carte SX18 peut être équipée par un module radio ER400TRS optionnel pour établir une liaison série radio. Ce module permet d'établir une communication série sans fils. Le module ER400TRS possède en interne toutes les fonctions nécessaires aux transmissions radio: codes de contrôle d'erreur, gestion de la transmission, gestion du contrôle de la tension d'alimentation, etc...



La partie communication série radio-fréquence est connectée sur le connecteur J6 pins 8 (RXD) et 9 (TXD) et est identifié comme étant **/dev/ttyS2**.



Le module série sans fil RF peut-être utilisé pour piloter la carte SX16-RF. Cette carte est une extension d'entrées/sorties communiquant par ondes radio directement grâce au module RF de la carte. Il est également possible de réaliser la SX18 Board à un ordinateur équipé avec le module USBRF04.



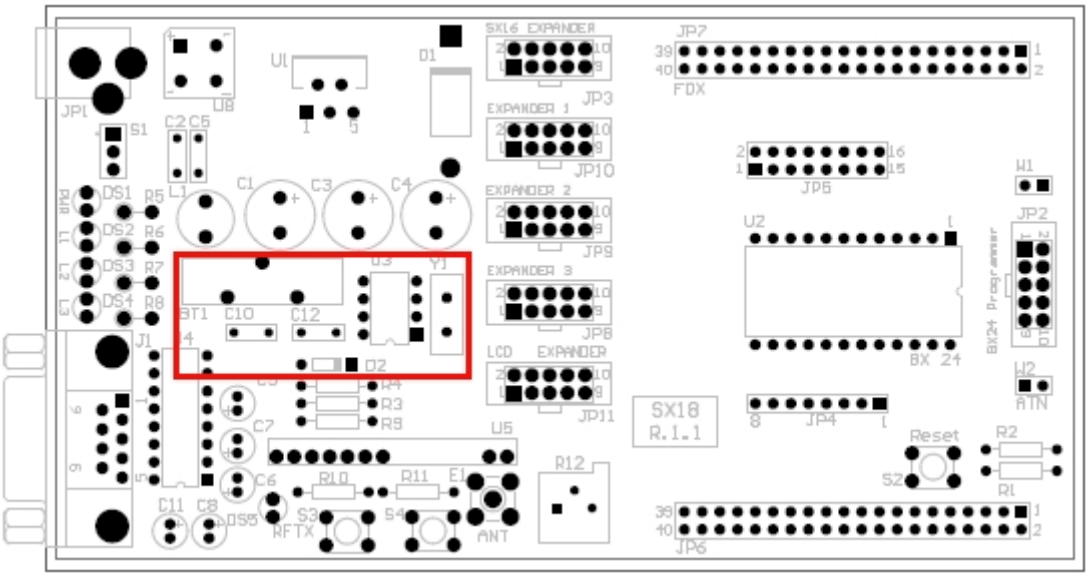
Le code source pour tester le périphérique SX18 RF est le même qu celui pour tester la liaison RS232, il est juste de nécessaire de modifier le **define Device** de cette manière:

#define DEVICE "/dev/ttyS2"

Lorsque les données séries passent au travers de la liaison RF la Led RFTX clignotera.

f Test de l'horloge temps réel

La carte SX18 est équipé d'un composant Dallas RTC DS1302 avec sa propre batterie de sauvegarde complètement supporté par le noyau de la Fox. L'horloge temps réelle est nécessaire pour stocker l'heure et la date exacte sur le système Fox, afin de la conserver même en cas d'extinction de l'alimentation.



Pour mettre à jour l'heure et la date, lancer la commande **date** avec la syntaxe suivante:

date MMDDHHmmYYYY

Où:

MM est le mois courant entre 01 et 12
DD est le jour courant entre 01 et 31
HH est l'heure courante entre 00 et 24
mm sont les minutes courantes entre 00et 59
YYYY est l'année courante

par exemple**:**

date 051813502005

Afin de sauvegarder l'heure de manière permanente dans le composant RTC et permettre au système de la recharger à chaque démarrage, il est nécessaire de lancer la commande **hwclock -w**

Ensuite pour visualiser la date sauvegardée, il est simplement nécessaire de lancer à partir d'une console la commande :

date

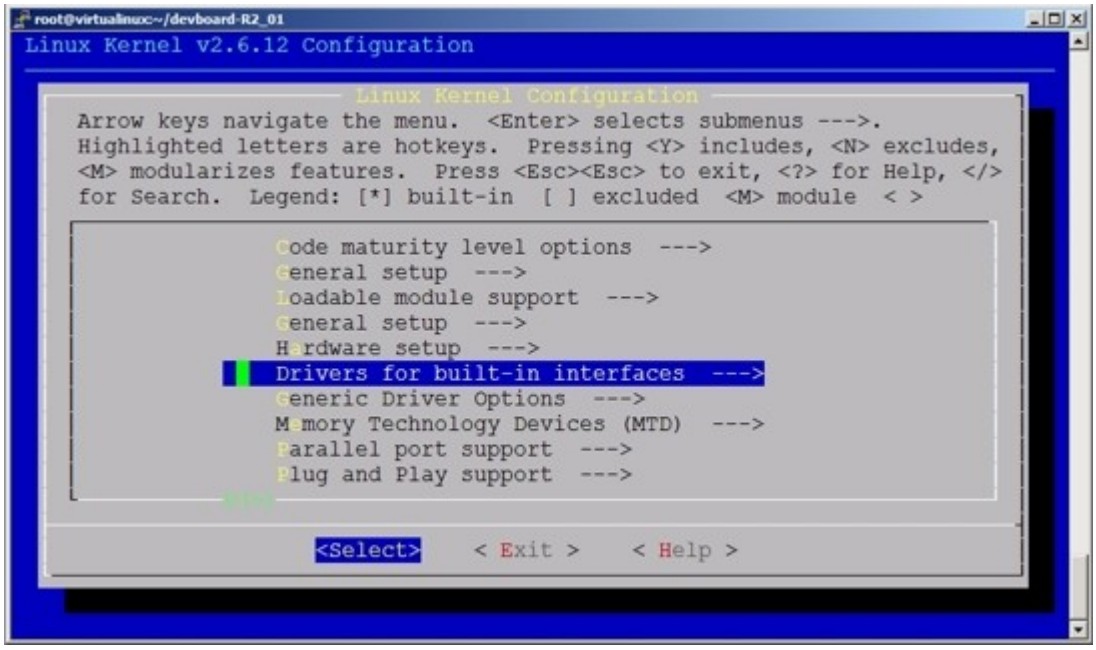
Comment autoriser la RTC à partir du noyau de la Fox ?

Dans cette partie nous allons voir comment configure l'utilisation de l'horloge temps réelle dans le SDK de la carte Fox Board/

Pour pouvoir utiliser le composant DS1302, il est nécessaire d'indiquer au noyau avec quels signaux il doit communiquer. Le composant DS1302 communique avec la carte Fox Board grâce au protocole I2C.

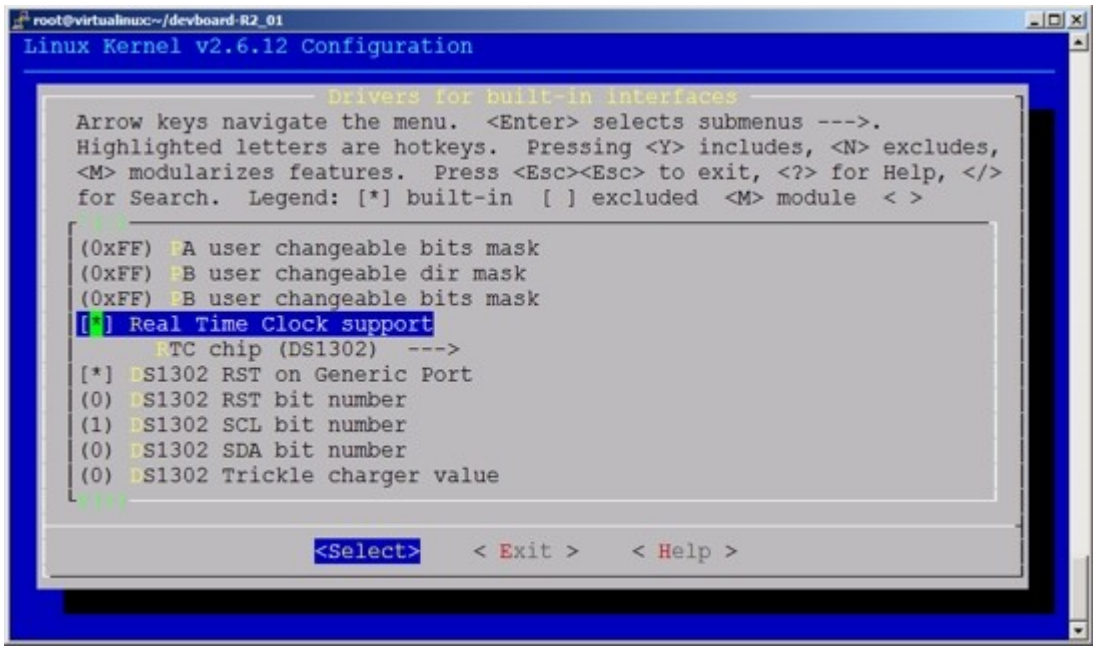
Dans une console positionnée sur le répertoire /devboard-R2_01 taper les commandes suivantes:

Autoriser l'utilisation des variable d'environnement: **. ./init_env**

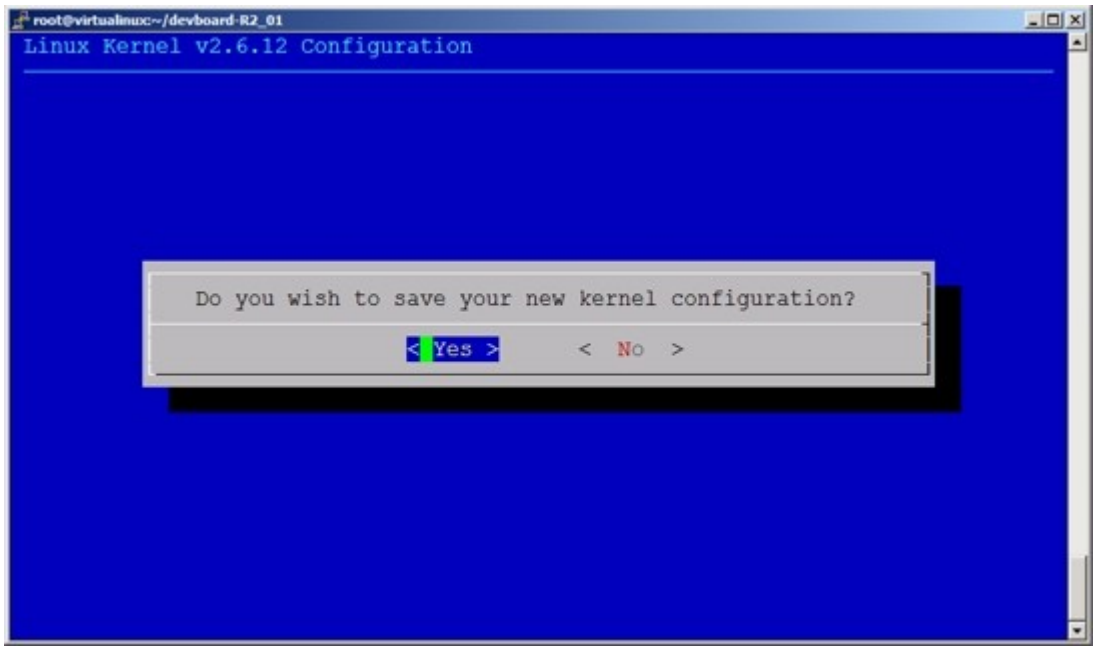


Lancer l'interface de configuration du noyau 2.6.15: **make menuconfig**

Choisir l'onglet **"Drivers for built-in interfaces --->"**



Sélectionner l'onglet "**Real Time Clock support**" et "**DS1302 RTS on Generic Port**" et positionner les bits de configuration du DS1302 comme indiqué sur la figure ci-dessus:



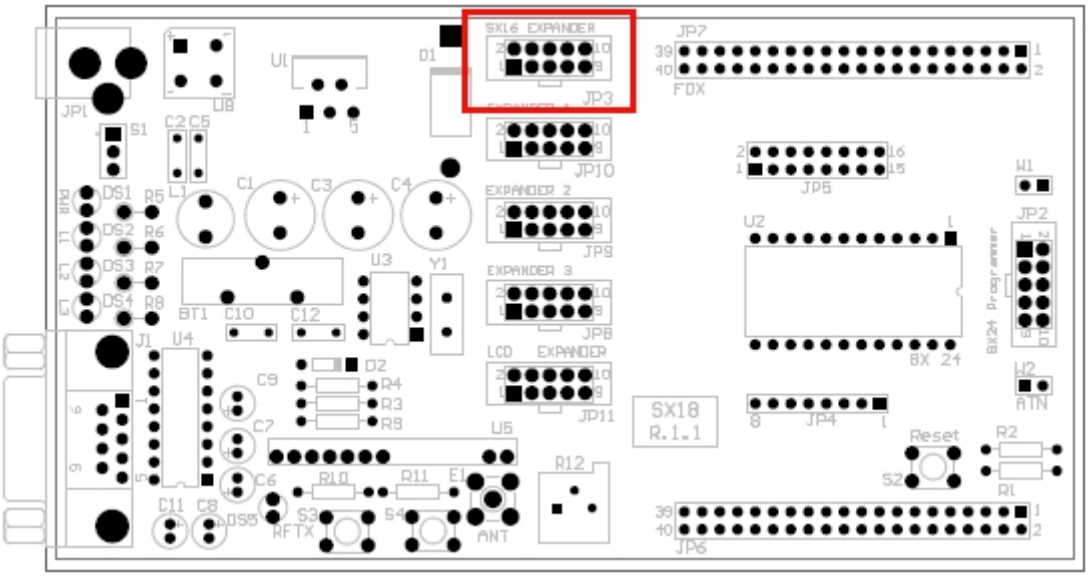
Sauvegarder la configuration et sortir.

Compiler la nouvelle image en tapant: **make**

Reprogrammer ensuite la mémoire flash de la carte Fox comme expliqué dans la TP n°1

g Test de l'extension SX16

Le connecteur d'extension 10 pins de la SX18 nommé "SX16 EXPANDER" a été dimensionné pour permettre une connection entre la carte SX16-BASE et la carte FOX Board.



Configuration matérielle:

- 24 entrées de nature différentes détaillé ci-dessous:
  - 8 entrées directes en niveau de tension TTL (0V-5V)
  - 8 entrées filtrées (filtre CLC) de niveau de tension TTL (0V-5V)
  - 8 entrées opto-couplées pour des tensions d'entrées jusqu'à 24V DC, éventuellement configurables comme des tensions d'entrées TTL (0V-5V)
- 6 sorties à relais qui autorise un pilotage de tensions jusqu'à 125V (sur une charge de 30W)
- 1 capteur de température DS1621 avec une précision de ½ C°et une plage de -55 a + 128°C

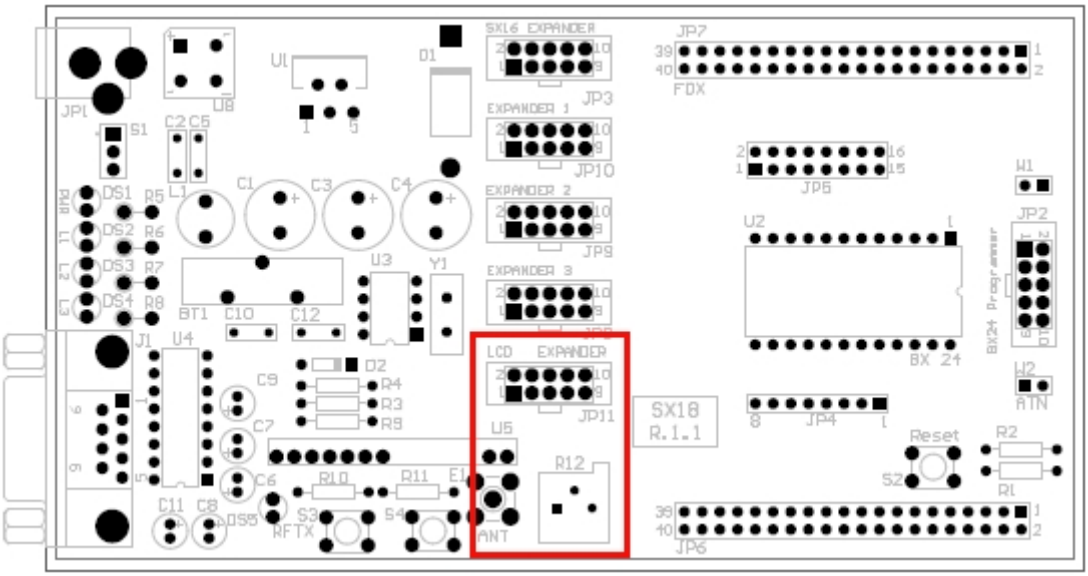
La carte d'extension SX16 est connectées à la carte Fox en suivant le tableau de correspondance des pins ci-dessous:

FOX	Extension SX16 de la SX18		Connecteur JP2 sur SX16-BASE
PB6 (pin 38-J6)	pin 1	<->	SDA (pin 1)
PB7 (pin 37-J6)	pin 2	-->	SCL (pin 2)
OG3 (pin 26-J6)	pin 3	-->	ICL (pin 3)
IG1 (pin 24-J6)	pin 4	<--	IDA (pin 4)
OG5 (pin 22-J6)	pin 5	-->	IPL (pin 5)
OG1 (pin 23-J6)	pin 6	-->	OCL (pin 6)
OG4 (pin 25-J6)	pin 7	-->	STR (pin 7)
OG2 (pin 21-J6)	pin 8	-->	ODA (pin 8)
VCC +5V	pin 9	---	VCC +5V (pin 9)
GND	pin 10	---	GND (pin 10)

Le code source pour tester cette carte SX16 est décrit dans l'article suivant: **Controllare la SX16-BASE dalla FOX board**

h Test de l'extension pour l'afficheurLCD

Le connecteur d'extension appelé "LCD EXPANDER" a été étudié spécialement pour pouvoir connecter rapidement un afficheur LCD sur la base du composant Hitachi HD44780 à la carte FOX.



A coté du connecteur 10 points JP11 se trouve la résistance ajustable (R12) nécessaire pour contrôler le contraste de l'afficheur.

Les pins du connecteur d'extension LCD sont décrits dans le tableau ci-dessous:

FOX	ESPANDER LCD on SX18	LCD
IOG8 (pin 14-J6)	pin 1	D0
IOG9 (pin 13-J6)	pin 2	D1
IOG10 (pin 16-J6)	pin 3	D2
IOG11 (pin 15-J6)	pin 4	D3
IOG12 (pin 18-J6)	pin 5	RS
IOG13 (pin 17-J6)	pin 6	EN
IOG14 (pin 20-J6)	pin 7	BL
Trimmer R12	pin 8	
VCC +5V	pin 9	VCC
GND	pin 10	GND

Des exemples de code source pour contrôler l'afficheur sont disponibles sur le lien suivant:
http://www.acmesystems.it/?id=8021

i Description de l'extension Entrées/sorties

La carte SX18 possède 3 connecteur d'extensions 10 points (5+5) nommés respectivement JP8, JP9 et JP10. il sont disponibles pour permettre à l'utilisateur de disposer d'un certain nombre de signaux de la carte Fox, comme décrits dans les 3 tableaux ci-dessous:

Extension JP8

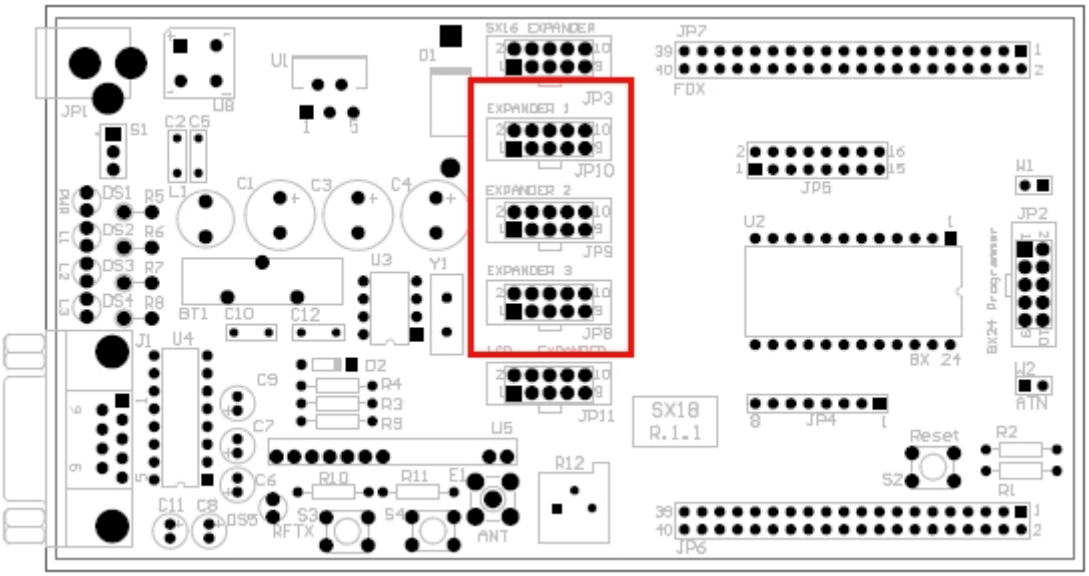
FOX	EXPANDER JP8 on SX18
IOG8 (pin 14-J6)	pin 1
IOG9 (pin 13-J6)	pin 2
IOG10 (pin 16-J6)	pin 3
IOG11 (pin 15-J6)	pin 4
IOG12 (pin 18-J6)	pin 5
IOG13 (pin 17-J6)	pin 6
IOG14 (pin 20-J6)	pin 7
IOG15 (pin 19-J6)	pin 8
VCC +5V	pin 9
GND	pin 10

Extension JP9

FOX	EXPANDER JP9 on SX18
IOG16 (pin 9-J7)	pin 1
IOG17 (pin 10-J7)	pin 2
IOG18 (pin 7-J7)	pin 3
IOG19 (pin 8-J7)	pin 4
IOG20 (pin 5-J7)	pin 5
IOG21 (pin 6-J7)	pin 6
IOG22 (pin 3-J7)	pin 7
IOG23 (pin 4-J7)	pin 8
VCC +5V	pin 9
GND	pin 10

Extension JP10

FOX	EXPANDER JP9 on SX18
TXD (pin 9-J6)	pin 1
RXD (pin 10-J9)	pin 2
IG2 (pin 29-J6)	pin 3
IG3 (pin 28-J6)	pin 4
IG4 (pin 27-J6)	pin 5
IG5 (pin 30-J6)	pin 6
IRQ (pin 30-J7)	pin 7
NMI (pin 11-J6)	pin 8
VCC +5V	pin 9
GND	pin 10

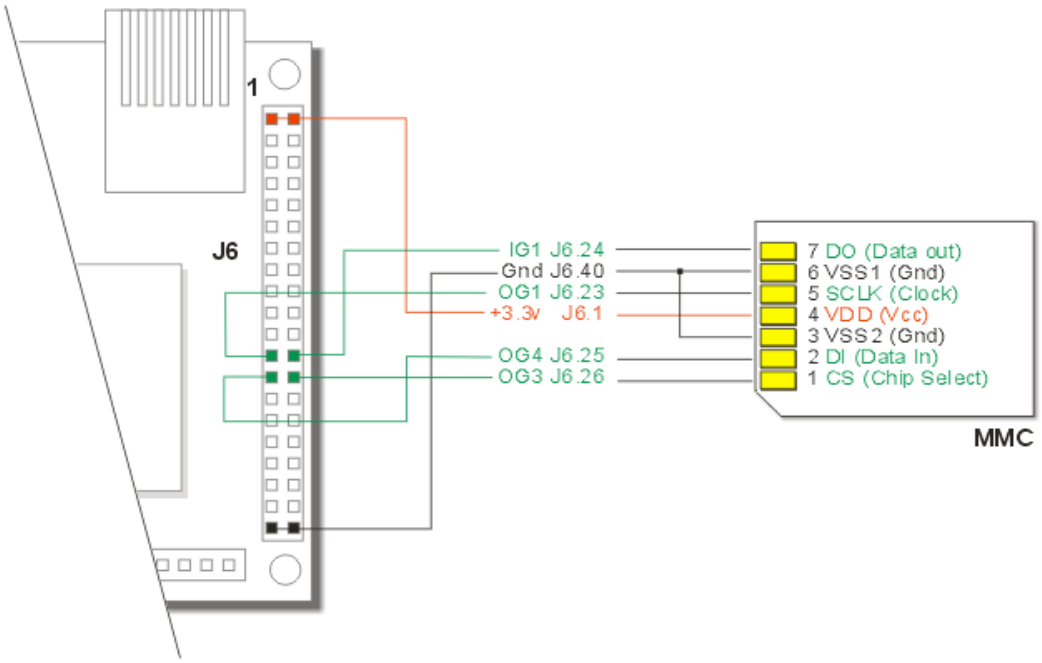


j Accès à la carte Flash SD/MMC

Connecter une carte mémoire MMC ou SD

écrit en anglais par John Crispin et Sergio Tanzilli

Cet article explique comment connecter une carte de type MMC ou SD en mode SPI sur la carte Fox Board.



Connections (à titre indicatif si l'on possède déjà la carte SX18):

Il est simplement nécessaire de relier 4 lignes de signaux plus 2 lignes d'alimentation pour connecter la carte mémoire à la carte Fox.

Comment configurer le noyau de la Fox avec le SDK pour connecter les cartes mémoire MMC/SD ?

A partir du répertoire devboard taper:

```
# make menuconfig
```

Autoriser l'option suivante:

```
Driver settings --> [*] Enable MMC Support
```

sauvegarder, sortir et générer l'image fimage, puis taper:

```
# ./configure
...
```

```
# make
```

Reprogrammer la mémoire de la carte Fox comme expliquer dans le TP n°1 et se logger avec **root** et taper le mot de passe **pass**.

Comment monter la carte mémoire?

insérer la carte mémoire MMC/SD à l'intérieur du connecteur spécial et la monter sur le système de fichiers en tapant:

```
# mount /dev/mmc0 /mnt/0 -t vfat -o noatime,sync
```

Maintenant il est possible de voir son contenu et de le modifier dans le répertoire /mnt/0 en tapant:

```
# ls -l /mnt/0
```