La mise à jour du firmware du FA-SDR

Le but de ce petit tutoriel est de décrire la procédure de mise à jour du firmware du FA-SDR, qui est un clone du SoftRock. J'ai réalisé cette mise à jour car je n'étais pas tout à fait satisfait du fonctionnement du SDR, en effet dans le logiciel PowerSDR-IQ ou "PowerSDR v2.4.4 for homemade rigs", le pas du synthétiseur était au minimum de 50 Hz alors que le Si570 est capable de s'accorder au Hertz près. Le mieux est parfois l'ennemi du bien me direz vous mais avec le logiciel HDSDR, le changement de fréquence du synthétiseur génèrait des craquements qui rendaient désagréable l'exploration de la bande. La mise à jour a résolu ces deux problèmes.

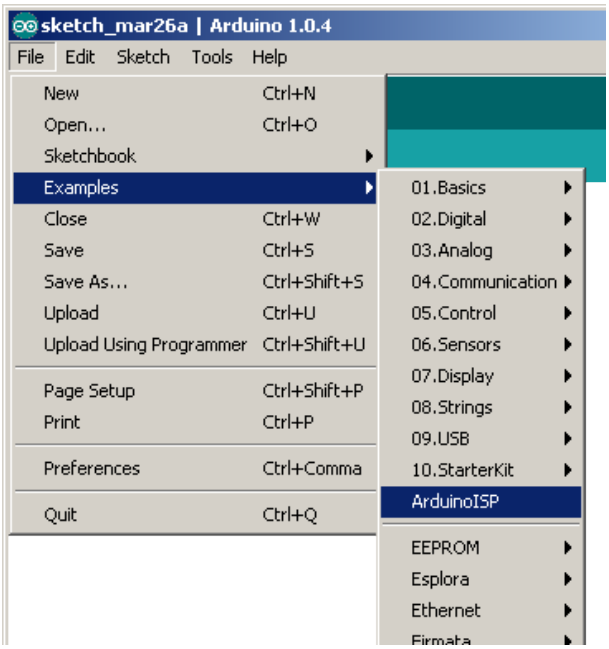
Toute "l'intelligence" du FA-SDR se trouve dans le logiciel qui pilote le synthétiseur Si570. L'accord à la fréquence d'écoute sur votre logiciel SDR préféré est réalisé par l'intermédiaire une couche d'abstraction matérielle, la fameuse DLL sous windows, qui fournit au logiciel SDR les fonctions permettant de donner ses ordres au synthétiseur. Dans le microcontroleur qui gère le synthétiseur se trouve le firmware (micrologiciel) . Le logiciel SDR demande la fréquence d'accord par l'intermédiaire de la DLL, qui envoie l'ordre via la liaison USB, et c'est le firmware du microcontrôleur ATMEL AVR qui calcule la valeur des registres à envoyer au Si570 par l'intermédiaire du bus i2c.

Par défaut le firmware installé sur le FA-SDR est celui de Thomas Baier DG8SAQ, la version v1.4 originelle qui date de 2008. Le firmware a été repris et amélioré par Fred Krom PE0FKO jusqu'à la version actuelle qui est la v15.15. En ce qui concerne le microcontrôleur, il faut savoir que le firmware v1.4 tient sur un AVR ATTiny45 (4ko de mémoire flash) et que la version v15.15 doit être flashée sur un ATTiny85 (8ko)

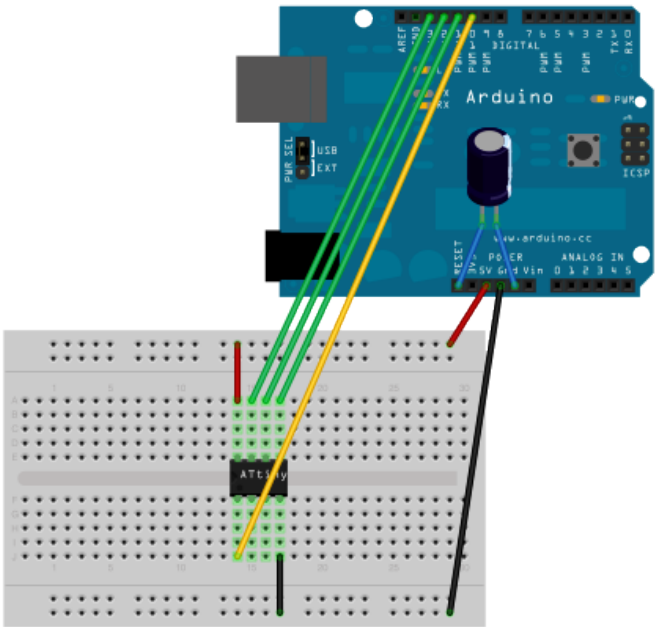
Pour les notes de version consulter
http://pe0fko.nl/SR-V9-Si570/#upgrade

La manipulation que je propose est faite à partir d'un arduino uno qui sert de programmateur d'eeprom flash. Le matériel nécessaire est le suivant.
* Un PC sous Windows
* Un arduino uno avec son câble USB
* Un platine à trous-trous avec quelques chutes de fil rigide pas trop fin
* Un ou plusieurs microcontrôleur ATMEL AVR ATTiny85

La première chose à faire est de transformer l'arduino en programmateur d'eeprom. Pour cela
Il faut charger le programme "arduino ISP" dans l'environnement de développement intégré (EDI) arduino (disponible ici : http://arduino.cc/en/main/software), et le flasher dans l'arduino.



Ensuite il faut connecter le programmateur arduino à la cible ATTiny85, à l'aide de la platine à trous comme sur le dessin suivant. Le condensateur chimique entre les broches RESET et GND de l'arduino n'est pas nécessaire ici.



Pour la partie logicielle, il faut installer WinAVR disponible ici :
http://sourceforge.net/projects/winavr/
Il faut veiller à bien cocher la case qui permet d'ajouter le dossier d'installation à la variable d'environnement PATH. Cet ensemble d'outils contient un programme nommé avrdude.exe, qui va nous servir pour flasher le firmware.

Ensuite il faut télécharger le firmware à cette adresse :
http://pe0fko.nl/SR-V9-Si570/AVR-FirmwareV15.15-ATtiny85.hex

Dans le dossier où se trouve le fichier HEX, il faut créer un fichier texte et le nommer en "upload.bat"



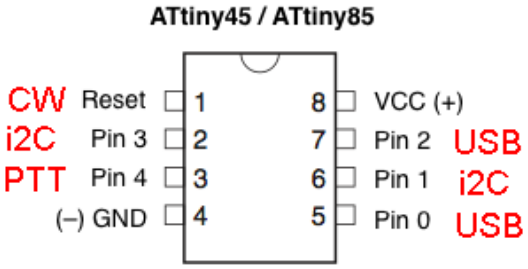
Ce fichier batch doit contenir le code suivant:
DEBUT

```
@rem modification F4GSW 2013/03/26
@rem PE0FKO 2009/02/01
@rem Fuse 1 = Disabled
@rem Fuse 0 = Enabled
@rem Fuse bit information:
@rem Fuse high byte:
@rem 0xdd = 1 1 0 1   1 1 0 1     RSTDISBL disabled (SPI programming can be done)
@rem 0x5d = 0 1 0 1   1 1 0 1     RSTDISBL enabled (PB5 can be used as I/O pin)
@rem        ^ ^ ^ ^   ^ \-+-/
@rem        | | | |   |   +------ BODLEVEL 2..0 (brownout trigger level -> 2.7V)
@rem        | | | |   +---------- EESAVE (preserve EEPROM on Chip Erase -> not preserved)
@rem        | | | +-------------- WDTON (watchdog timer always on -> disable)
@rem        | | +---------------- SPIEN (enable serial programming -> enabled)
@rem        | +------------------ DWEN (debug wire enable)
@rem        +-------------------- RSTDISBL (disable external reset -> disabled)
@rem
@rem Fuse low byte:
@rem 0xe1 = 1 1 1 0   0 0 0 1
@rem        ^ ^ \+/   \--+--/
@rem        | |  |       +------- CKSEL 3..0 (clock selection -> HF PLL)
@rem        | |  +--------------- SUT 1..0 (BOD enabled, fast rising power)
@rem        | +------------------ CKOUT (clock output on CKOUT pin -> disabled)
@rem        +-------------------- CKDIV8 (divide clock by 8 -> don't divide)
avrdude.exe -p t85 -c arduino -P com4 -b 19200 -e ^
  -U flash:w:AVR-FirmwareV15.15-ATtiny85.hex:i ^
  -U lfuse:w:0xE1:m -U hfuse:w:0x5D:m  -v
pause
```

FIN

bien sûr il faut remplacer le port com4 par le port sur lequel est connecté l'arduino.
pour connaître le rôle des différentes options, c'est ici :
http://www.nongnu.org/avrdude/user-manual/avrdude_4.html#Option-Descriptions

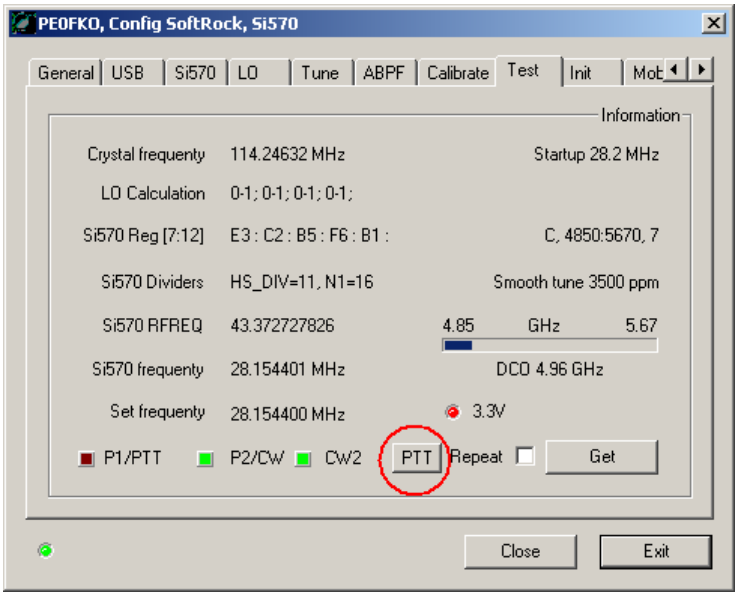
Il faut savoir qu'avec cette commande la programmation n'est possible qu'une seule fois, car cette configuration désactive la programmation par bus SPI. en effet la broche RESET du ATTiny85,qui sert à le programmer, est utilisée pour la commande CW dans le FA-SDR. si vous voulez pouvoir tester avant la programmation définitive, il faut remplacer "hfuse:w:0x5D:m" par "hfuse:w:0xDD:m".



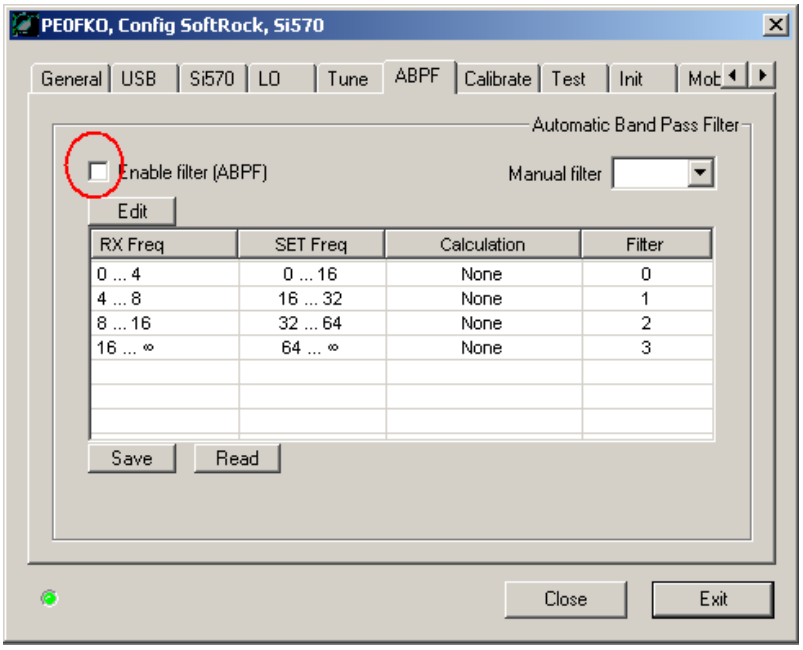
En cas de problème avec la programmation d'un ATTiny85, il est possible de réinitialiser les "fuses" (bits de configuration permanents) grâce à cette procédure que j'ai testé pour vous :
https://sites.google.com/site/wayneholder/attiny-fuse-reset

Voilà, on peut lancer l'exécution du fichier upload.bat et la programmation du microcontrôleur démarre. Si tout c'est bien passé vous pouvez remplacer le ATTiny45 du FA-SDR par le ATTiny85 fraîchement flashé.

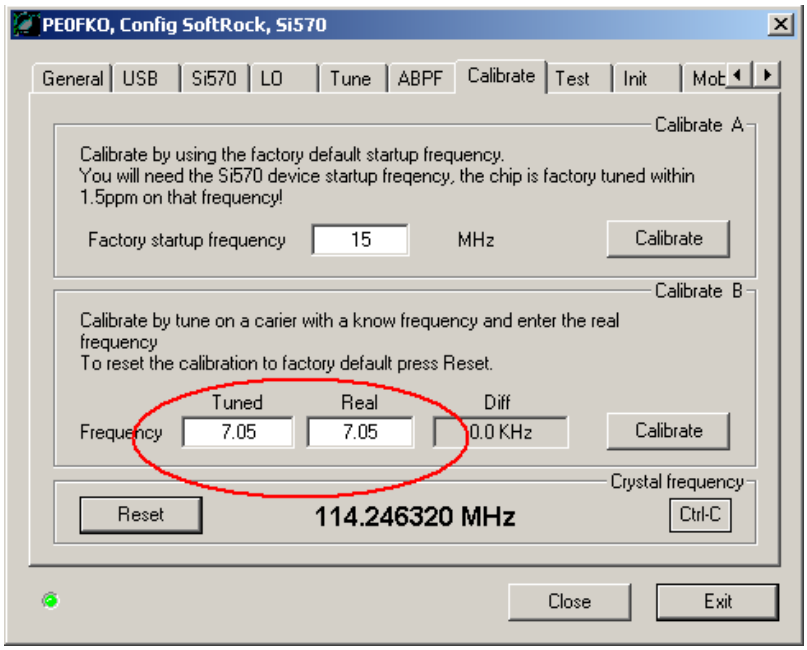
Lors de la première mise sous tension avec le nouveau firmware, un nouveau matériel USB est détecté par Windows. Il faut lui indiquer le driver de PE0FKO disponible ici : http://pe0fko.nl/SR-V9-Si570/PE0FKO-USB-Driver-1.2.6.0.zip sur la page http://pe0fko.nl/SR-V9-Si570/index.shtml#install
J'ai préfèré l'installation manuelle des pilotes. On remarquera que lors de la mise sous tension le PTT est enclenché. Pour le désactiver il suffit de lancer l'outil de configuration CFGSR disponible ici :
http://pe0fko.nl/CFGSR/Install_CFGSR_V2.6.msi sur la page http://pe0fko.nl/CFGSR/



Pour pouvoir utiliser la commande CW du FA-SDR il faut désactiver l'ABPF (automatic band pass filter), et couper puis remettre l'alimentation (reboot du firmware).



Et voilà il n'y a plus qu'à lancer PowerSDR pour trouver un étalon de fréquence (signaux horaires ou autre), et effectuer la calibration avec CFGSR



Par Alban F4GSW, le 27 mars 2013.

photos de F4GSW Alban