tricot-machine, ftdi, fabrication, brother-KH940, em

Fabrication d'un cable USB Série pour machine à tricoter

Pour envoyer des motifs à une machine à tricoter Brother Electroknit KH940, on utilise une communication série sur USB, afin de simuler un lecteur de disquette.

Ce câble doit respecter un certain protocole de communication série. Pour le fabriquer on part d'un câble USB série FTDI. Ce type de câble intégre un circuit intégré programmable dans le connecteur USB, il sera nécessaire de modifier la configuration de ce circuit en le reprogrammant avec le logiciel adapté.

Pour fabriquer ce câble, nous nous sommes largement basés sur le tutoriel de Daviworks (lien ci-dessous).


Cable FTDI USB-Série. Photo de FTDI, DR

Matériel nécessaire

Composants

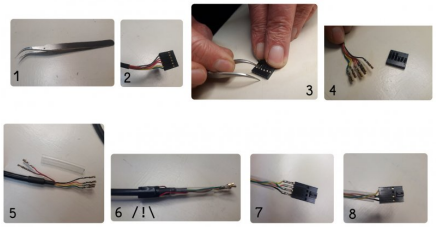
- Cable FTDI 5V USB série : https://fr.rs-online.com/web/p/products/6877770/
- Connecteur Molex C-GRID III 2×4 : https://fr.rs-online.com/web/p/products/3606235/
- gaine thermorétractable (c'est mieux, mais pas fondamental)
- scotch d'électricien

Outils

- brucelles fines (*tweezers* pour les anglophones)
- ordinateur pour la programmation du câble

Fabrication du connecteur

Il faut défaire le connecteur existant, sans endommager le sertissage des fils, et les réinsérer dans le connecteur C-GRID adapté à la prise de la machine.



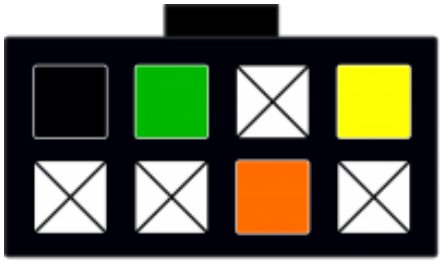
télécharger en pdf

1. défaire le connecteur

En s'aidant des brucelles, soulever chaque picot de plastique pour libérer l'extrémité sertie du câble.

2. insérer les fils dans le nouveau connecteur

On insère 4 fils aux bonnes positions : jaune, orange, vert, noir. Les fils rouge et marron sont repliés le long du câble et scotchés **sans se toucher**! (Attention il faut absolument que ces deux fils soient isolés l'un de l'autre.)



connecteur C-GRID vu de face

C'est assez délicat, il faut que les fils soient enfoncés à la bonne profondeur pour rester bloqué.

Programmation du câble FTDI avec Linux

Pour utiliser le cable, il est nécessaire de l'adapter au protocole de communication de la machine à tricoter. Pour cela, il faut inverser les niveaux logiques sur les broches RXD, TXD et RTS. Pour le programmer, on peut utiliser un logiciel fourni par FTDI, fabricant du câble, malheureusement il ne fonctionne que sur windows.

Nous avons cherché des moyens de le faire sur Linux, c'est complexe mais possible! D'abord, nous avons testé ftx-prog de Richard Meadows et FT232R de Mark Lord, sans succès.

Finalement, la solution consiste à utiliser la libFTDI

Installation de la libFTDI

```
# Debian Stretch 9.5 64bits @ Kirin / 20200317
# télécharger depuis le site
# décompresser dans un répertoire
cd /home/emoc/libftdi1-1.4
sudo apt-get install build-essential gcc make libftdi-dev libusb-dev libconfuse-dev
sudo apt-get install git-core cmake doxygen swig python-dev libboost-all-dev libusb-1.0-0-dev
cd libftdi
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX="/usr" ../
make
sudo make install
```

Utilisation

```
cd /home/emoc/libftdi1-1.4/examples     # pour tester la lecture
./eeprom
```

Je teste avec deux câbles différents (et la commande `./eeprom`), le premier a déjà été programmé (sur windows), les informations renvoyées sont correctes (= les bits TXD, RXD et RTS sont bien inversés) :

eeprom cable 1 (cliquer pour afficher le code)

```
Chip type 3 ftdi_eeprom_size: 128
0x000: 00 40 03 04 01 60 00 06  a0 2d 08 07 00 00 98 0a .@...`.. .-......
0x010: a2 02 a2 12 23 10 05 00  0a 03 46 00 54 00 44 00 ....#... ..F.T.D.
0x020: 49 00 12 03 46 00 54 00  32 00 4a 00 53 00 57 00 I...F.T. 2.J.S.W.
0x030: 59 00 4b 00 00 00 00 00  00 00 00 00 00 00 00 00 Y.K..... ........
```

```
0x040: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
0x050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
0x060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
0x070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 f0 32 ........ .......2
0x080: 17 04 e8 fb 00 00 cf f8  9c 58 42 00 00 00 00 00 ........ .XB.....
0x090: 00 00 00 00 00 00 00 00  4c 41 35 59 42 4b 5a 42 ........ LA5YBKZB
VID:     0x0403
PID:     0x6001
Release: 0x0600
Bus Powered:  90 mA USB Remote Wake Up
Manufacturer: FTDI
Product:
Serial:       FT2JSWYK
Checksum      : 32f0
Internal EEPROM
Oscillator: Internal
Enable Remote Wake Up
PNP: 1
Channel A has Mode UART VCP
Inverted bits: TXD RXD RTS
C0 Function: TXLED
C1 Function: RXLED
C2 Function: TXDEN
C3 Function: PWREN
C4 Function: SLEEP
```

Pour le second, dans son état tout frais du fournisseur, les bits TXD, RXD, RTS ne sont pas inversés! C'est la modification à effectuer pour que le câble soit utilisable avec la machine à tricoter.

eeprom cable 2 (cliquer pour afficher le code)

```
Chip type 3 ftdi_eeprom_size: 128
0x000: 00 40 03 04 01 60 00 06  a0 2d 08 00 00 00 98 0a .@...`.. .-......
0x010: a2 12 b4 12 23 10 05 00  0a 03 46 00 54 00 44 00 ....#... ..F.T.D.
0x020: 49 00 12 03 46 00 54 00  34 00 5a 00 33 00 4f 00 I...F.T. 4.Z.3.O.
0x030: 43 00 4e 00 12 03 46 00  54 00 35 00 46 00 54 00 C.N...F. T.5.F.T.
0x040: 56 00 33 00 4a 00 00 00  00 00 00 00 00 00 00 00 V.3.J... ........
0x050: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
0x060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 ........ ........
0x070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 86 77 ........ .......w
0x080: 21 04 de fb 00 00 8d fa  c4 38 42 00 00 00 00 00 !....... .8B.....
0x090: 00 00 00 00 00 00 00 00  4a 41 49 59 42 33 4e 30 ........ JAIYB3N0
VID:     0x0403
PID:     0x6001
Release: 0x0600
Bus Powered:  90 mA USB Remote Wake Up
Manufacturer: FTDI
Product:      FT4Z3OCN
Serial:       FT5FTV3J
Checksum      : 7786
Internal EEPROM
Oscillator: Internal
Enable Remote Wake Up
PNP: 1
Channel A has Mode UART VCP
C0 Function: TXLED
C1 Function: RXLED
C2 Function: TXDEN
C3 Function: PWREN
C4 Function: SLEEP
```

Flasher le câble FTDI pour machine à tricoter

En deux temps : d'abord, il faut créer le fichier de configuration :

ftdi.conf (cliquer pour afficher le code)

ftdi.conf

```
# Version modifiée de https://waterpigs.co.uk/articles/ftdi-configure-mac-linux/
# Vendor ID et Product ID permettent d'identifier le chipset
# sur linux on peut les trouver avec lsusb
# ex: Bus 001 Device 010: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC
# on peut trouver toutes les options de configuration dans le fichier libftdi1-1.4/ftdi_eeprom/main.c à partir de la ligne 430

vendor_id=0x403
product_id=0x6001

# The rest of the fields are settings which can be written to the
# FT2xx with the --flash-eeprom option.

# Max. power consumption: value * 2 mA. Use 0 if self_powered = true.
max_power=500
```

```
###########
# Strings #
###########
manufacturer="FTDI"
product="FT2JSWYK"
serial=""

###########
# Options #
###########
self_powered=false  # Turn this off for bus powered
remote_wakeup=false # Turn this on for remote wakeup feature
use_serial=true     # Use the serial number string

# Normally out don't have to change one of these flags
in_is_isochronous=false     # In Endpoint is Isochronous
out_is_isochronous=false    # Out Endpoint is Isochronous
suspend_pull_downs=false    # Enable suspend pull downs for lower power
change_usb_version=false    # Change USB Version
usb_version=0x0200          # Only used when change_usb_version is enabled

invert_rxd=true
invert_txd=true
invert_rts=true

cha_vcp=false
cha_type=FIFO
chb_type=UART

eeprom_type=0x46

########
# Misc #
########

# This is the relative filename that EEPROM contents will either be
# read from or written to, depending on whether ftdi_eeprom is run
# with the --read-eeprom or --flash-eeprom option.
filename="eeprom.bin"
```

Puis le flasher avec ftdi_eeprom

```
ftdi_eeprom --flash-eeprom ./ftdi.conf
```

Cela crée aussi un fichier binaire du contenu de l'EEPROM, qu'on peut afficher avec

```
hd eeprom.bin
```

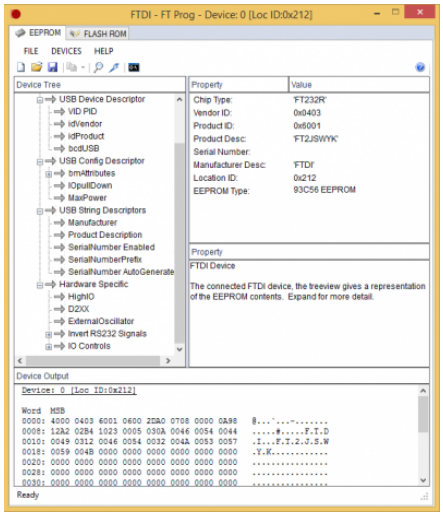
On peut vérifier que cela a bien fonctionné avec

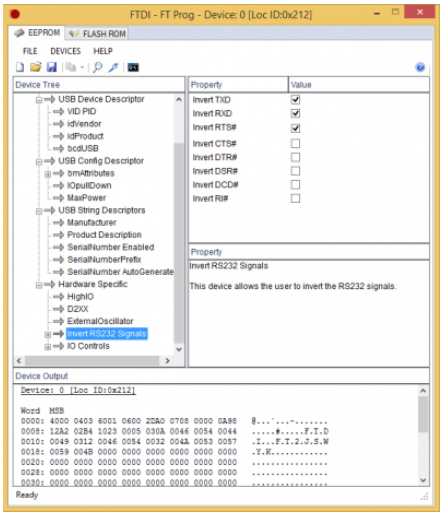
```
./eeprom # (eeprom est une application du dossier libftdi1-1.4/examples )
```

Programmation du câble FTDI avec Windows

Testé sur Windows 8.
Télécharger et installer l'application depuis de site de FTDI : https://www.ftdichip.com/Support/Utilities.htm#FT_PROG
Puis, brancher le câble et

- démarrer FT_PROG
- dans le menu devices : Scan & Parse (F5)
- dans le device tree à gauche, choisir “Hardware Specific / Invert RS232 Signals”
- cocher “Invert TXD”, “Invert RXD”, “Invert RTS#' (image de droite)
- dans le menu devices : Program
- cliquer Program





Sources & ressources

- datasheets du câble FTDI USB-Série : doc cable USB Série FTDI
- tutoriel pour le câble que l'on a suivi https://daviworks.com/knitting/cable_tutorial.html
- câbles pour Brother KH930 https://learn.adafruit.com/electroknit/cable
- doc de la commande ftdi_eeprom https://manpages.debian.org/testing/ftdi-eeprom/ftdi_eeprom.1.en.html
- puces FTDI “brickés” : https://tech.scargill.net/ftdi-bricked-chips-fix/
- https://waterpigs.co.uk/articles/ftdi-configure-mac-linux/
- https://github.com/mozilla-sensorweb/sensorweb-wiki/wiki/Flashing-the-FTDI-EEPROM-under-linux
- bibliothèque python pour puces FTDI (pas grand chose pour l'EEPROM, mais en dev.) : https://github.com/eblot/pyftdi
- exemple de fichier de config pour ftdi_eeprom : https://github.com/RandomReaper/ft2tcp/blob/master/eeprom-config/ft232h.conf

Article extrait de : http://lesporteslogiques.net/wiki/ - **WIKI Les Portes Logiques**
Adresse :
http://lesporteslogiques.net/wiki/recherche/tricot_machine/fabrication_cable_usb-serie?rev=1584528990
Article mis à jour: **2020/03/18 11:56**