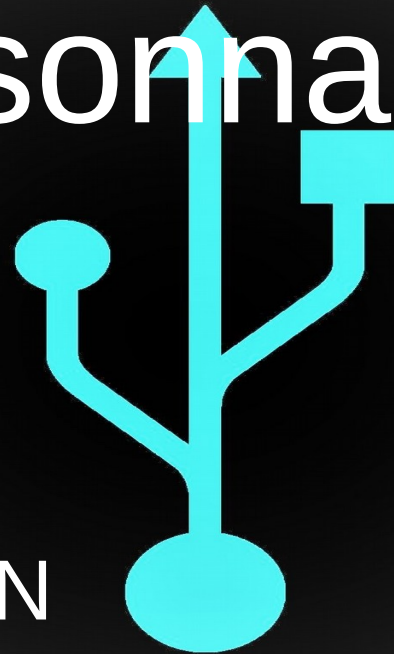


Périphériques USB personnalisés

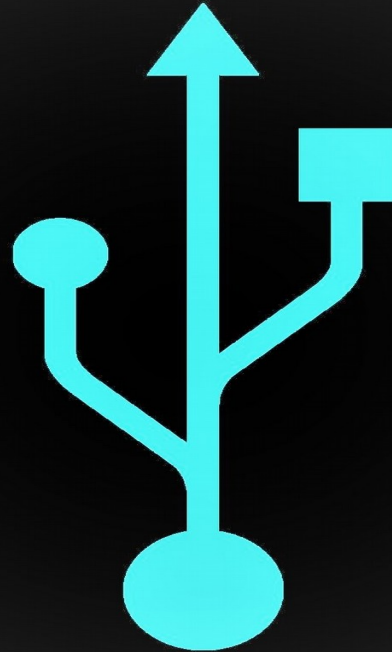


Rémi GUILLOMON
Maxime DUQUESNE
Quentin DELPECH
Baptiste JEAN-LOUIS

IMA3
Promo 2021

Sommaire

- Introduction
- Protocole USB
- Bibliothèque LUFA
 - Présentation
 - AT90USBKEY en stockage de masse
 - Configuration de LUFA
- Prototypes Arduino
 - Bibliothèques utilisées
 - Hardware utilisé
 - Prototype de la Clé USB
 - Prototype du Clavier
- Conclusion



Introduction

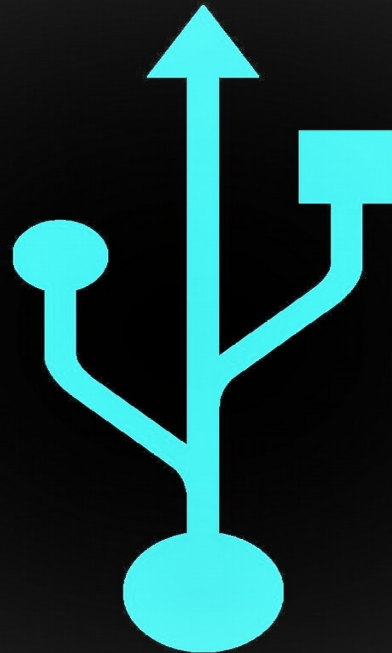
En quoi consiste notre projet ?

Un clavier

Ressemble à un clavier classique

Fonctionne comme un clavier classique

+ enregistre toutes les entrées clavier !

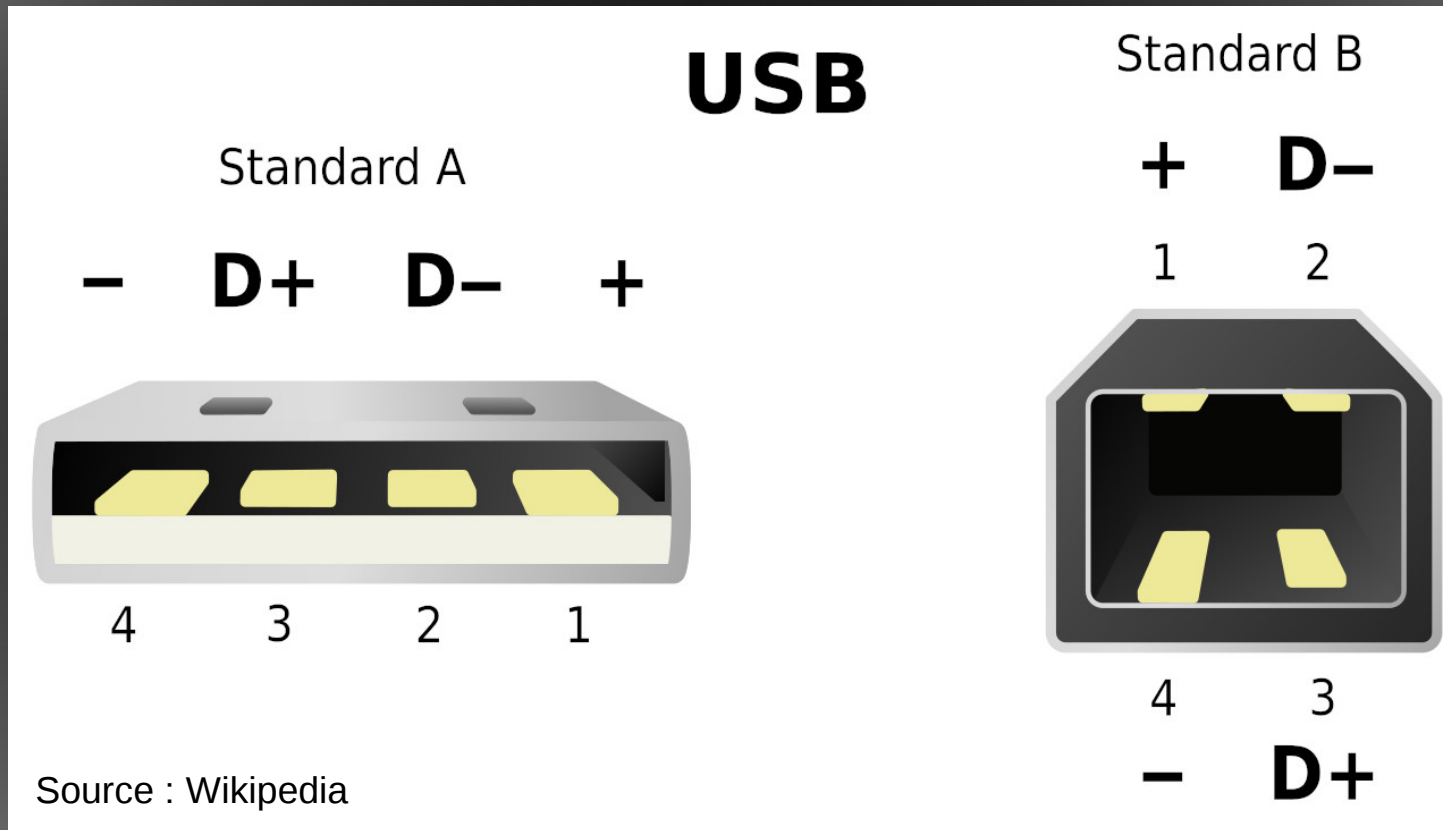


Une clé USB

Stocke vos fichiers comme une clé classique

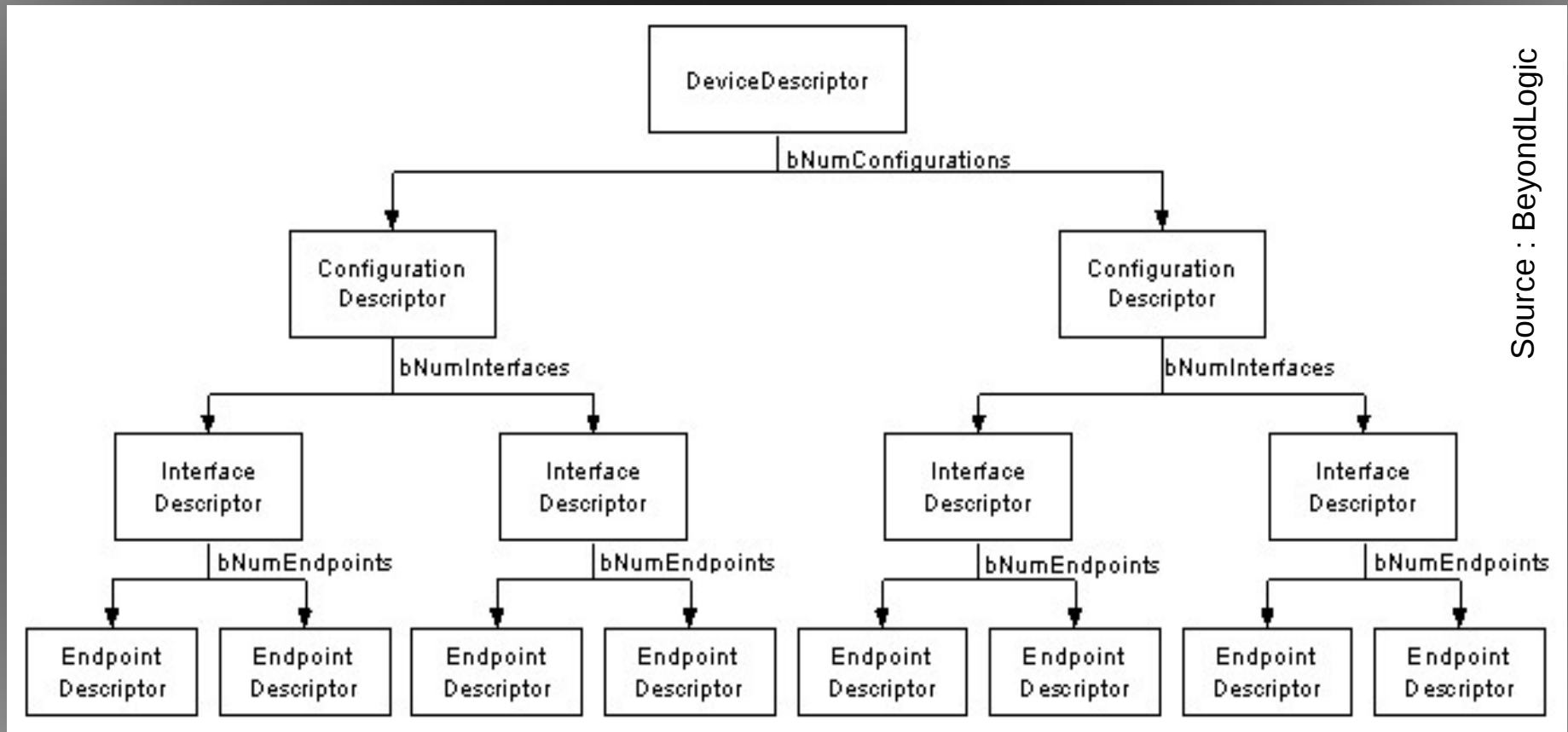
Vous vous absentez ?
Elle télécharge un logiciel espion !

Protocole USB



Broches USB

Protocole USB



Source : BeyondLogic

Descripteurs USB

Bibliothèque LUFA



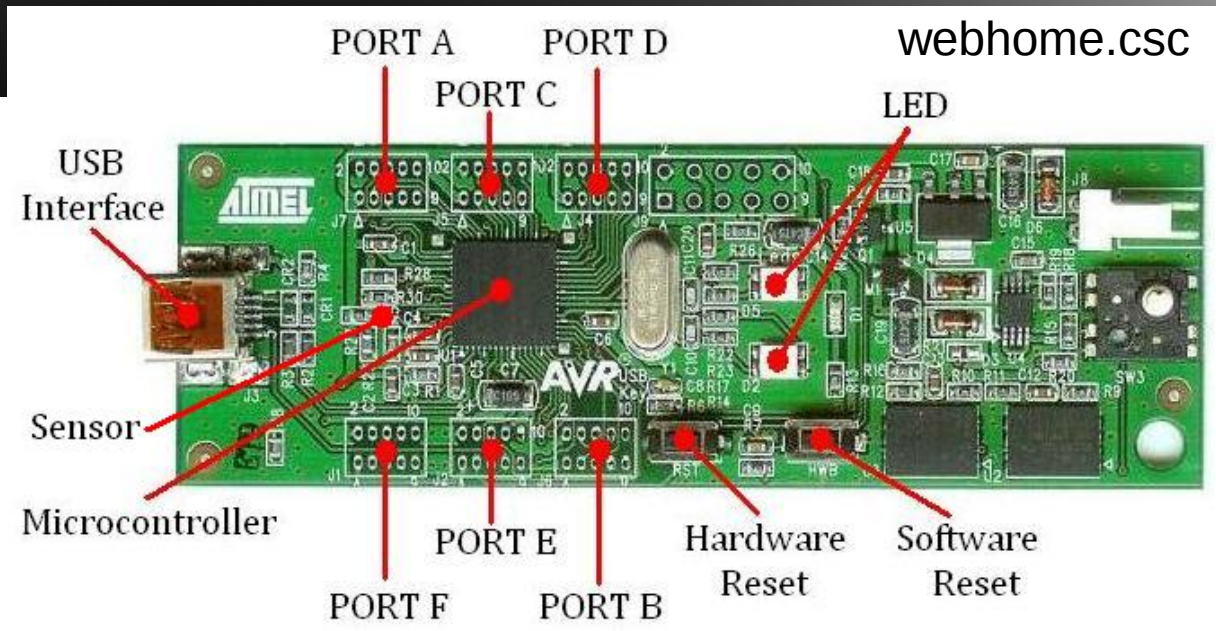
Lightweight USB
Framework for AVR's

USB Class	Device Mode	Host Mode
Android Open Accessory	No	Yes
Audio 1.0	Yes	Yes
CDC-ACM	Yes	Yes
HID	Yes	Yes
MIDI	Yes	Yes
Mass Storage	Yes	Yes
Printer	Yes	Yes
RNDIS	Yes	Yes
Still Image	No	Yes

Classes USB compatibles avec LUFA

AT90USBKEY en stockage de masse

```
##  
# LUFA Library  
# Copyright (C) Dean Camera, 2017.  
# dean [at] fourwalledcubicle [dot] com  
# www.lufa-lib.org  
# -----  
# LUFA Project Makefile.  
# -----  
# Run "make help" for target help.  
  
MCU = at90usb1287  
ARCH = AVR8  
BOARD = USBKEY  
F_CPU = 8000000  
F_USB = $(F_CPU)  
OPTIMIZATION = s  
TARGET = MassStorage  
SRC = $(TARGET).c Descriptors.c Lib/DataflashManager.c  
Lib/SCSI.c $(LUFA_SRC_USB) $(LUFA_SRC_USBCLASS)  
LUFA_PATH = ../../../../LUFA  
CC_FLAGS = -DUSE_LUFA_CONFIG_HEADER -IConfig/  
LD_FLAGS =
```



Carte AT90USBKEY

Makefile

AT90USBKEY en stockage de masse

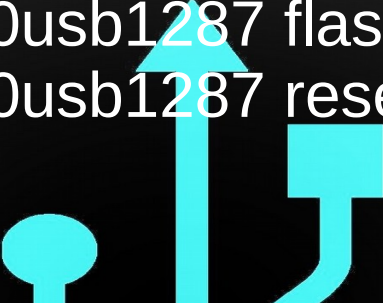
Commandes dans le terminal :

```
sudo dfu-programmer at90usb1287 erase
```

```
sudo dfu-programmer at90usb1287 flash MassStorage.hex
```

```
sudo dfu-programmer at90usb1287 reset
```

```
lsusb
```



```
Bus 002 Device 004: ID 8087:07da Intel Corp.  
Bus 002 Device 003: ID 04f2:b330 Chicony Electronics Co., Ltd Asus 720p CMOS webcam  
Bus 002 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub  
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 003: ID 0bda:0139 Realtek Semiconductor Corp. RTS5139 Card Reader Controller  
Bus 001 Device 002: ID 8087:0024 Intel Corp. Integrated Rate Matching Hub  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 004 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub  
Bus 003 Device 002: ID 046d:c05b Logitech, Inc. M-U0004 810-001317 [B110 Optical USB Mouse]  
Bus 003 Device 010: ID 03eb:2045 Atmel Corp. LUFA Mass Storage Demo Application  
Bus 003 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Carte AT90USBKEY reconnue comme stockage de masse

AT90USBKEY en stockage de masse

```
Bus 003 Device 010: ID 03eb:2045 Atmel Corp. LUFA Mass Storage Demo Application  
Couldn't open device, some information will be missing
```

```
Device Descriptor:
```

```
bLength          18  
bDescriptorType  1  
bcdUSB          1.10  
bDeviceClass     0 (Defined at Interface level)  
bDeviceSubClass  0  
bDeviceProtocol  0  
bMaxPacketSize0 8  
idVendor        0x03eb Atmel Corp.  
idProduct       0x2045 LUFA Mass Storage Demo Application  
bcdDevice       0.01  
iManufacturer   1  
iProduct        2  
iSerial         220  
bNumConfigurations 1
```

```
Configuration Descriptor:
```

```
bLength          9  
bDescriptorType  2  
wTotalLength    32  
bNumInterfaces  1  
bConfigurationValue 1  
iConfiguration  0  
bmAttributes    0x80  
  (Bus Powered)
```

```
MaxPower        100mA
```

```
Interface Descriptor:
```

```
bLength          9  
bDescriptorType  4  
bInterfaceNumber 0  
bAlternateSetting 0  
bNumEndpoints   2  
bInterfaceClass  8 Mass Storage  
bInterfaceSubClass 6 SCSI  
bInterfaceProtocol 80 Bulk-Only  
iInterface       0
```

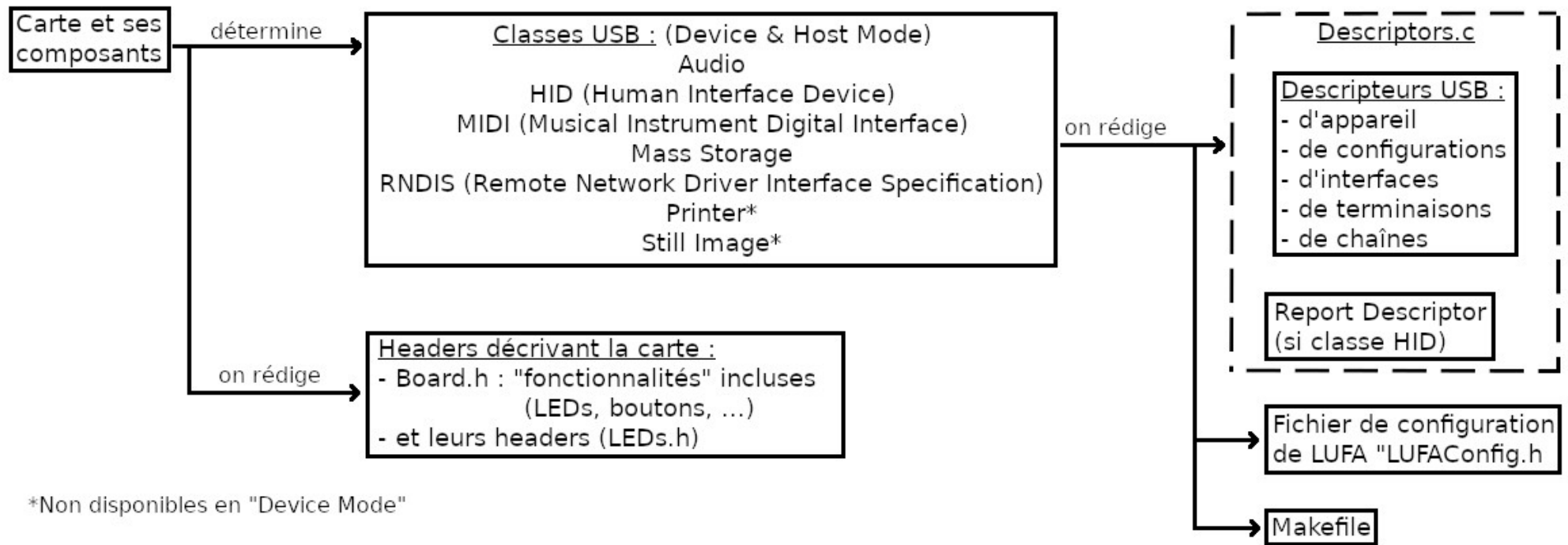
```
Endpoint Descriptor:
```

lsusb -v

Détail des descripteurs
Carte AT90USBKEY
en stockage de masse

Configuration de LUFA

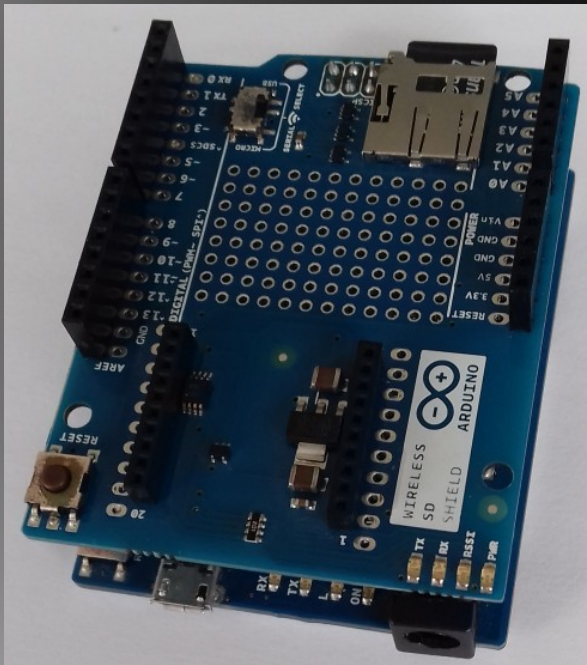
Comment configurer LUFA pour une carte personnalisée ?



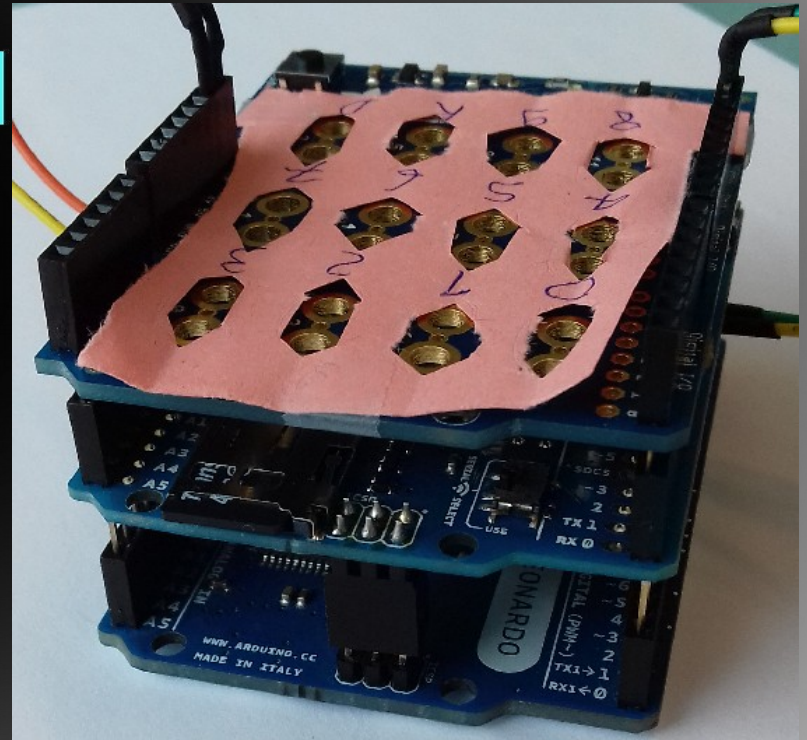
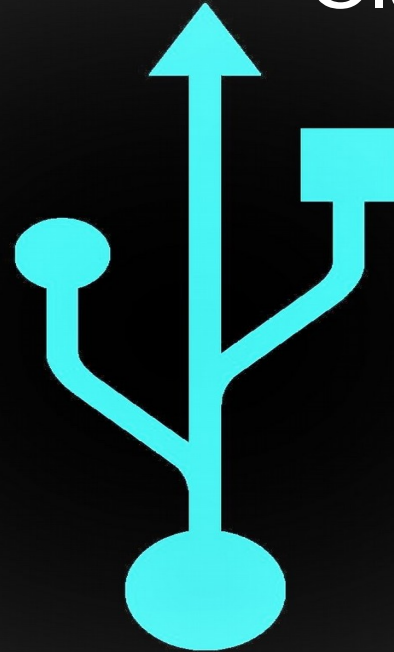
Schématisation de la configuration

Prototypes Arduino

- Clé USB



- Clavier



Bibliothèques

- SD et File

Elles permettent de gérer une carte SD et son contenu

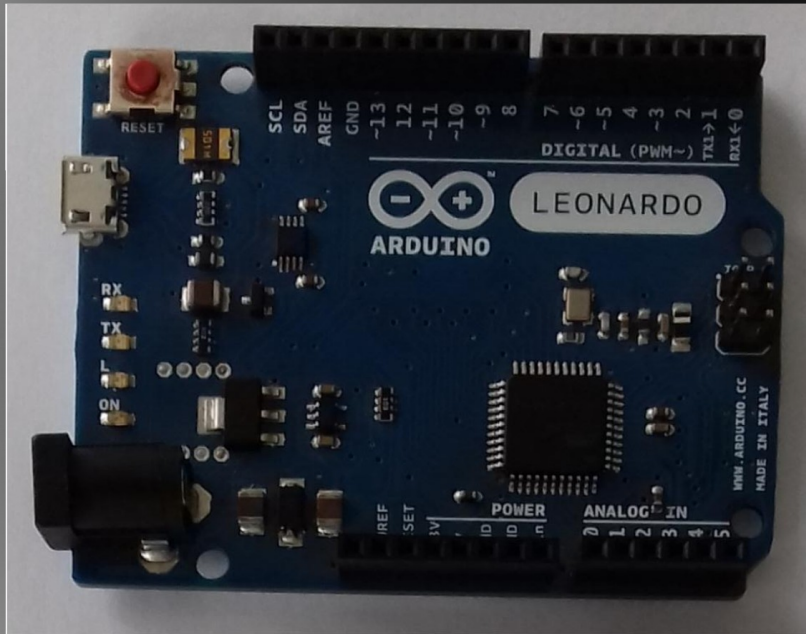
- Keyboard

Elle permet de faire passer la carte Arduino pour un clavier du point de vue de l'ordinateur.

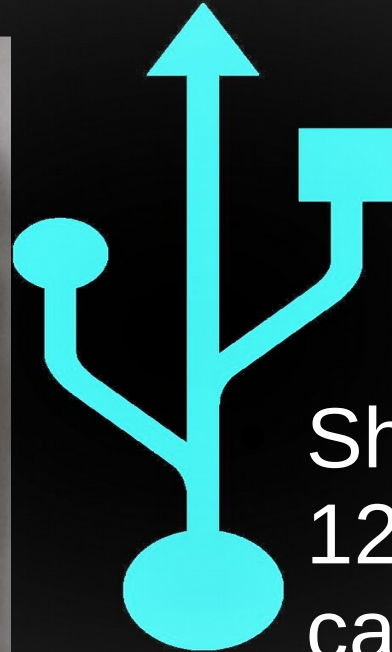
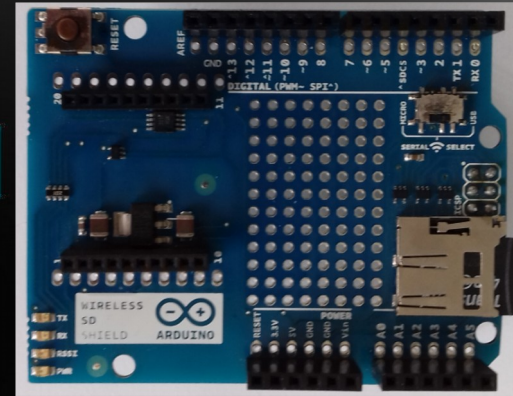


Hardware

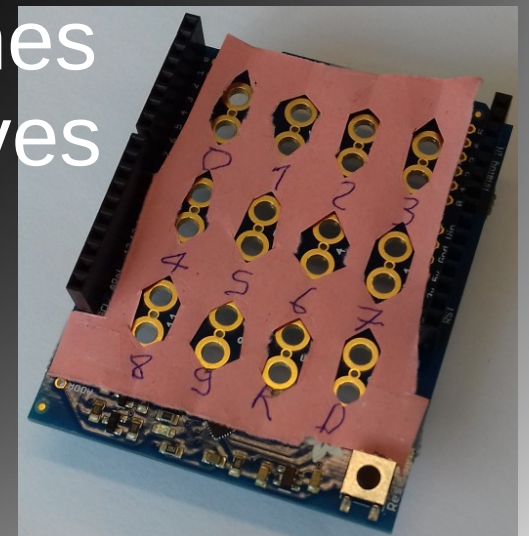
- Arduino Leonardo



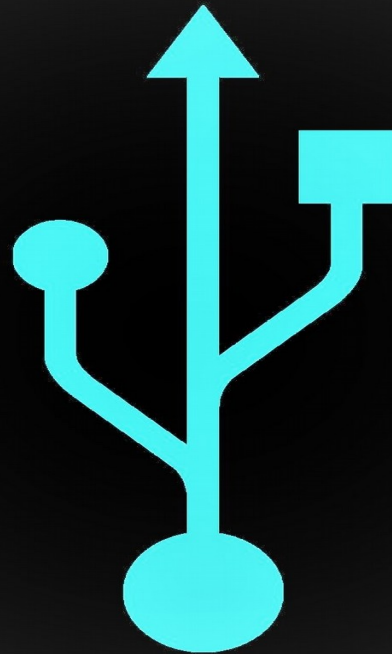
- Shield Arduino SD



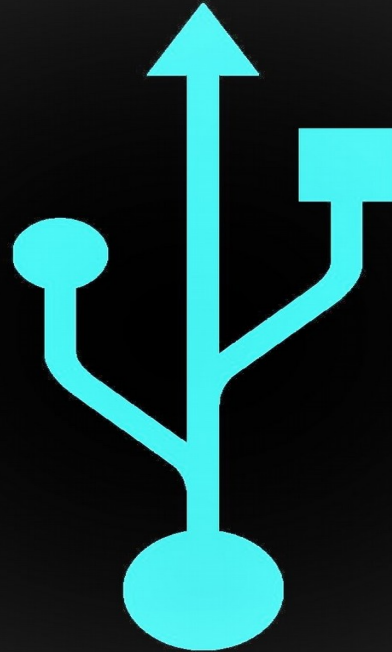
Shield Adafruit clavier
12 touches
capacitives



Conclusion



Merci pour votre écoute



Annexes

- Clef USB keyboardprint()
- Commandes Shell Windows
- Clavier - gestion des touches classiques
- Clavier - gestion de la touche R
- Clavier - gestion de la touche D

Clef USB :

Permet d'entrer directement une chaîne de caractère

```
void keyboardprint ( String texte ){
  int i = 0;
  while (texte[i]!='\0'){
    if(texte[i]=='@'){          /*
      Keyboard.write("");
    }
    else if(texte[i]=='""){
      Keyboard.write('@');    Prise en compte des caractères spéciaux
    }
    else if(texte[i]=='\\'){
      Keyboard.press(KEY_RIGHT_ALT);
      Keyboard.press(92);
    }                          */
    else {
      Keyboard.write(texte[i]);
    }
    delay(ti);                // Delay entre les inputs
    Keyboard.releaseAll();
    i++;
  }
}
```

Commandes Shell Windows

Ouvrir un power Shell en admin :

Presser touche Windows

Ecrire : Powershell

Presser : entrer + Shift + entrer

Presser : <-

Presser : Entre

Lancer un .exe dans un power Shell:

C:\Program Files\qBittorrent\qbittorrent.exe'
(exemple avec qbittorrent)

Télécharger un logiciel dans un power Shell

Invoke-WebRequest -Uri «adresse du fichier» -OutFile «adresse destination +
nom et type fichier»

«[https://www.mediacollege.com/audio/tone/files/440Hz_44100Hz_16bit_30sec.
mp3](https://www.mediacollege.com/audio/tone/files/440Hz_44100Hz_16bit_30sec.mp3)» -OutFile «./mp3.mp3»

Clavier : Gestion des touches classiques

```
for (uint8_t i=0; i<12; i++) {
    if(i==2 || i==5)i++; //ne pas traiter R et D
    // it if *is* touched and *wasnt* touched before, alert!
    if ((currouched & _BV(i)) && !(lasttouched & _BV(i)) ) {
        Serial.print(tab[i]); Serial.println(" touched");
        c=('0'+tab[i]);
        fichier2.write('<');
        fichier2.write(c);
        fichier2.write('>');
        Keyboard.press(c); }
    // if it *was* touched and now *isnt*, alert!
    if (!(currouched & _BV(i)) && (lasttouched & _BV(i)) ) {
        written=true;
        Serial.print(tab[i]); Serial.println(" released");
        c=('0'+tab[i]);
        fichier2.write('>');
        fichier2.write(c);
        fichier2.write('<');
        fichier.write(c);
        Keyboard.release(c);
    }
}
```

Clavier : Touche R

```
//récupération des entrées
if ((currouched & _BV(5)) && !(lasttouched & _BV(5))) {
  Serial.println("R");
  fichier.seek(0);
  while((c=fichier.read())!=-1){
    Keyboard.write(c);
  }
  Keyboard.write(KEY_RETURN);
  Keyboard.write(KEY_RETURN);

  fichier2.seek(0);
  while((c=fichier2.read())!=-1){
    Keyboard.write(c);
  }
}
```

Clavier : Touche D

```
//effacement de la mémoire
if ((currouched & _BV(2)) && !(lasttouched & _BV(2)) ) {
  Serial.println("D");
  fichier.close();
  SD.remove(filename);
  fichier=SD.open(filename,FILE_WRITE);
  fichier2.close();
  SD.remove(filename2);
  fichier=SD.open(filename2,FILE_WRITE);
}
```