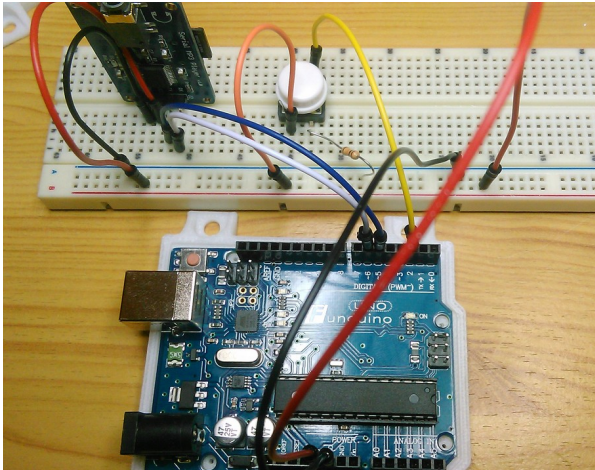




Test_MP3

La carte **Mp3 Serial Player** est une carte comprenant un circuit de décodage de fichiers MP3 pouvant être piloté par une liaison série sur deux fils, TX et RX, par une carte arduino.



Dans l'exemple qui suit on utilise les broche 5 et 6, respectivement RX et TX, pour echanger des ordres de commande avec la carte. les fichiers MP3 à lire et décodé seront stockés sur une carte microSD qui sera insérée dans l'emplacement prévu de la carte MP3 Sérial Player. Les fichiers devront être organisés d'une façon bien précise sur la carte mémoire.

Dans cet exemple on utilise un bouton poussoir connecté sur l'entrée 2 pour déclancher le changement de fichier.

```

/* ***** */
/* Demo pour la carte Serial MP3 Player de Catalex */
/* On utilise la bibliothèque standard SoftwareSerial pour dialoguer avec la carte */
/* ***** */

#include <SoftwareSerial.h>          // définition des commandes utilisées (doc constructeur)
#define CMD_SEL_DEV 0X09
#define DEV_TF 0X02
#define CMD_PLAY_W_INDEX 0X03

// définition des constantes
#define ARDUINO_RX 5                 // à connecter au TX de la carte MP3
#define ARDUINO_TX 6                 // à connecter au RX de la carte MP3

// Déclaration des constantes & variables
SoftwareSerial mySerial(ARDUINO_RX, ARDUINO_TX); // Créaltion d'une liason série
const int buttonPin = 2;             // Broche pour le bouton poussoir
const int ledPin = 13;              // Broche de la Led de la carte
int buttonState = 0;                 // variable pour mémoriser l'état du bouton poussoir
static int8_t Send_buf[8] = {0};    // Mémoire pour construire la commande à envoyer
static int16_t Track=0X1801;         // Premier fichier. Répertoire automatique

void setup()
{
  pinMode(ledPin, OUTPUT);           // initialisation the la broche de la LED en sortie
  pinMode(buttonPin, INPUT);         // initialisation de la broche du BP en entrée
  mySerial.begin(9600);              // ouverture du port série à 9600 bds
  delay(500);                        // attendre l'initialisation complète de la carte
  sendCommand(CMD_SEL_DEV, DEV_TF);  // commande de selection de la carte microSD
  delay(200);                        // attendre 200ms
  sendCommand(CMD_PLAY_W_VOL,0X1801); // lire le premier fichier au volume 15
}

void loop()
{
  buttonState = digitalRead(buttonPin); // lire et mémoriser l'état du BP
  if (buttonState == HIGH)              // si on a appuyé sur le bouton poussoir
  {
    Track+=1;                          // augmenter le n° de fichier
    sendCommand(CMD_PLAY_W_VOL, Track); // commande de lecture nouveau n° de fichier
    digitalWrite(ledPin, HIGH);        // Allumer la LED pendant 100ms
    delay (100);
  }
  else
  {
    digitalWrite(ledPin, LOW);         // Etteindre la LED
  }
}

```



```
// Construction et envoi de la commande à la carte MP3
// Programme d'exemple téléchargeable sur le site du constructeur

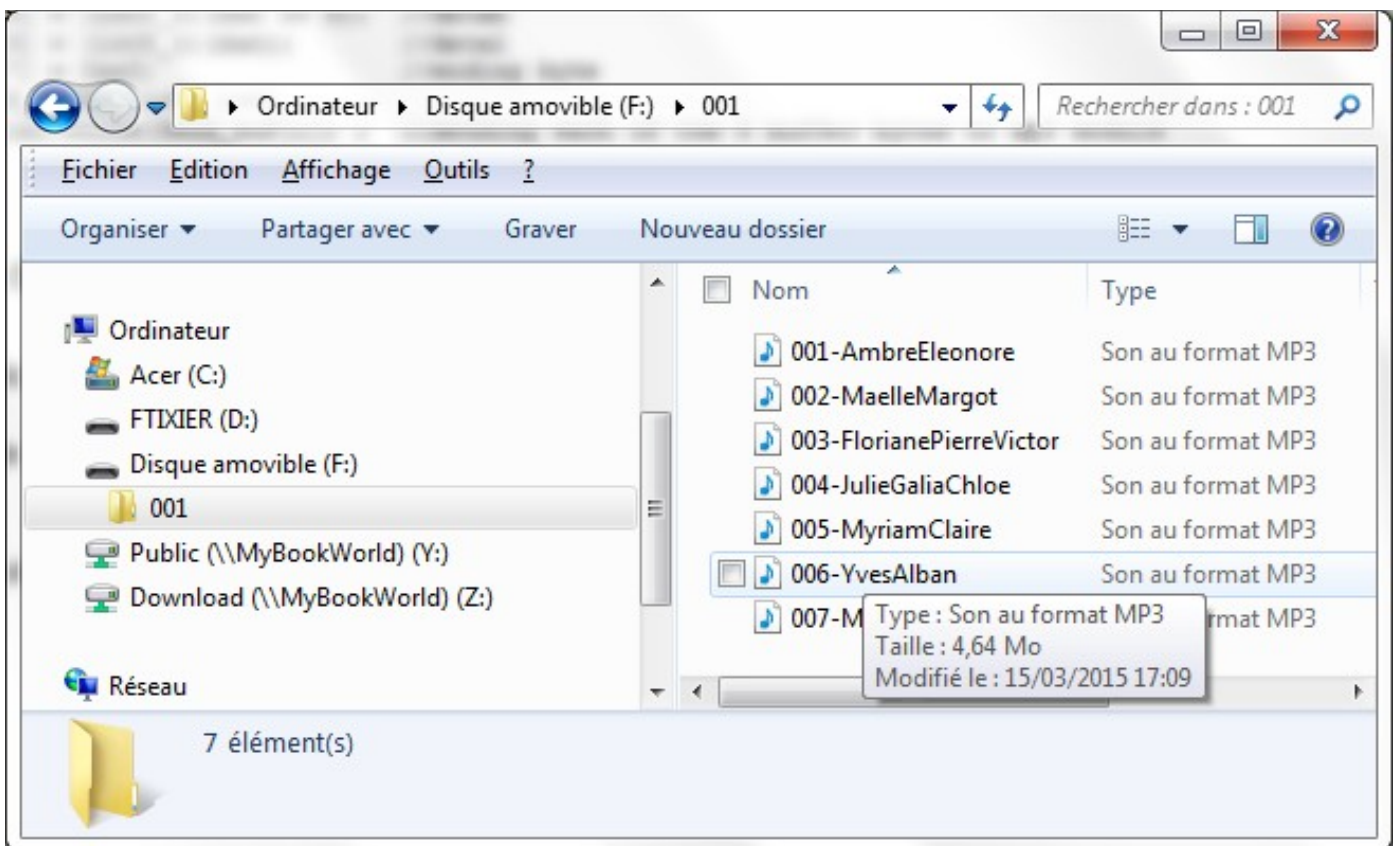
void sendCommand(int8_t command, int16_t dat)
{
    delay(20);
    Send_buf[0] = 0x7e; //starting byte
    Send_buf[1] = 0xff; //version
    Send_buf[2] = 0x06; //size (bytes) of the command without starting & ending byte
    Send_buf[3] = command;
    Send_buf[4] = 0x00; //0x00 = no feedback, 0x01 = feedback
    Send_buf[5] = (int8_t) (dat >> 8); //datah
    Send_buf[6] = (int8_t) (dat); //datal
    Send_buf[7] = 0xef; //ending byte
    for (uint8_t i=0; i<8; i++)
    {    mySerial.write(Send_buf[i]) ; //sending each of the 8 buffer bytes to mp3 module
    }
}
```

Stockage des fichiers sur la carte microSD

La carte microSD doit être formatée en FAT16 ou en FAT32 (ancien format windows).

Les noms de fichiers doivent impérativement commencer par un numéro sur trois chiffres décimaux.

Les répertoires seront nommés par des numéros : "001" par exemple.



Exemple d'arborescence de carte MicroSD (lecteur F:)