

Radial-V

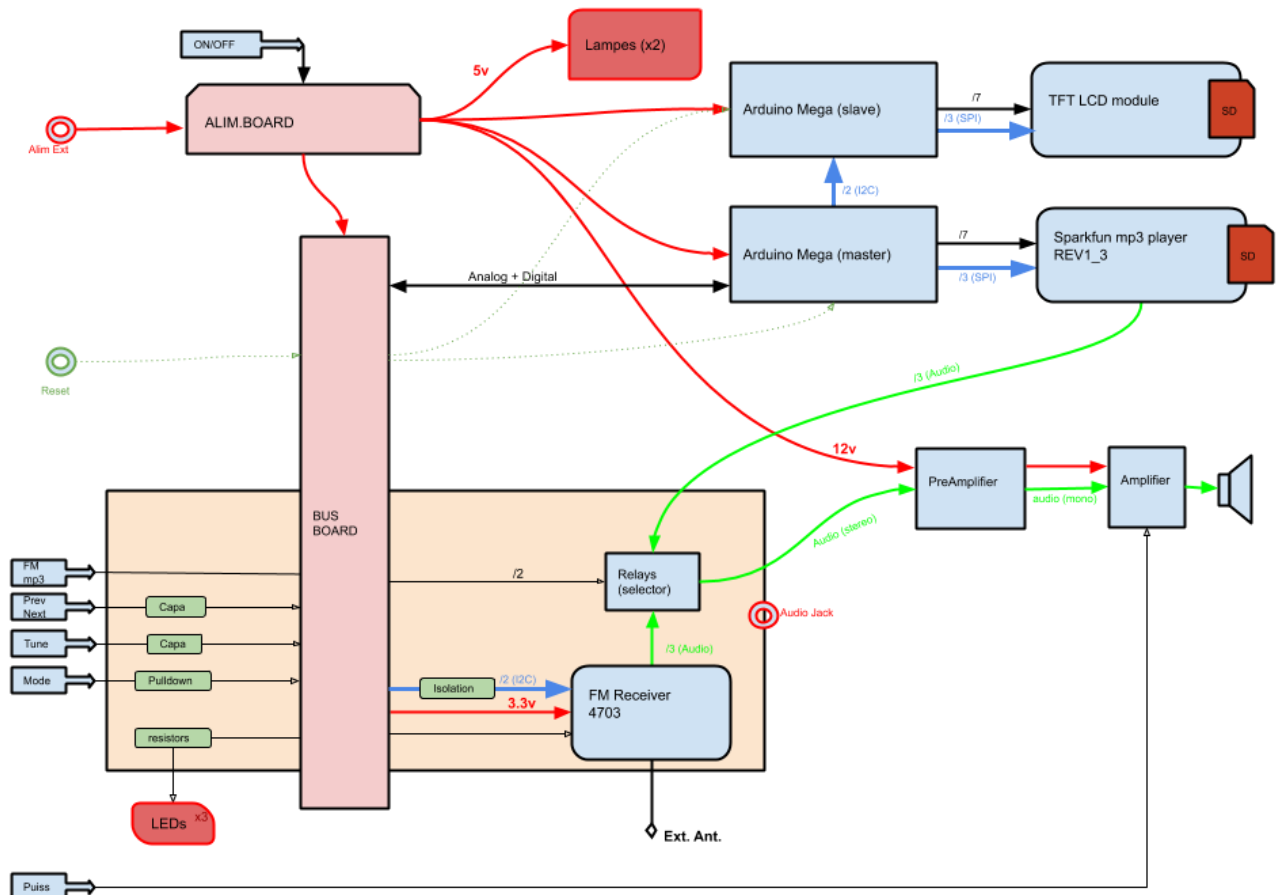
Date: 17 mai 2020

Table des matières

1. [Présentation](#)
2. [Protocoles des bus I2C et SPI](#)
 - 2.1. [Les commandes I2C](#)
3. [Câblage](#)
 - 3.1. [Câblage Ecran TFT](#)
 - 3.2. [Lexique Bus SPI](#)
 - 3.3. [Câblage bus SPI](#)
 - 3.4. [SelfReturn Button](#)
 - 3.5. [Adaptation de niveau pour bus I2C](#)
4. [Conception hardware](#)
 - 4.1. [Conception des schémas \(Eagle free edition\)](#)
 - 4.2. [Librairies et composants utiles](#)
 - 4.3. [Paramètres des circuits Eagle](#)
 - 4.4. [Script Eagle pour Export des Layouts](#)
5. [Cartes spécifiques](#)
 - 5.1. [Présentation](#)
 - 5.2. [Carte Alimentation Board](#)
 - 5.3. [Carte Bus Board](#)
 - 5.3. [Extension Board](#)
6. [Fabrication](#)
 - 6.1. [Réalisation des PCB](#)
 - 6.2. [Retour d'expérience](#)
7. [Utilisation de Radial-V](#)
 - 7.1. [Préparation des MP3](#)
8. [Annexes](#)

1. Présentation

Diagramme - Radial-V



Le système comporte deux cartes Arduino Mega: ARD1 (Master) et ARD2 (Slave).

- ARD1: utilise 19 I/O (dont 6 avec interruption) ce qui nécessite un Mega.
- ARD2: consomme 2 à 3 kB de RAM, ce qui nécessite un Mega.

Les deux Arduino communiquent en I2C.

Mega	16Mhz	16 analog	54 I/O	4 PWM	8 kB RAM	256 kB flash
Uno	16Mhz	6 analog	14 I/O	6 PWM	2 kB RAM	32 kB flash
Nano	16Mhz	8 analog	14 I/O	6 PWM	1 kB RAM	16 kB flash

2. Protocoles des bus I2C et SPI

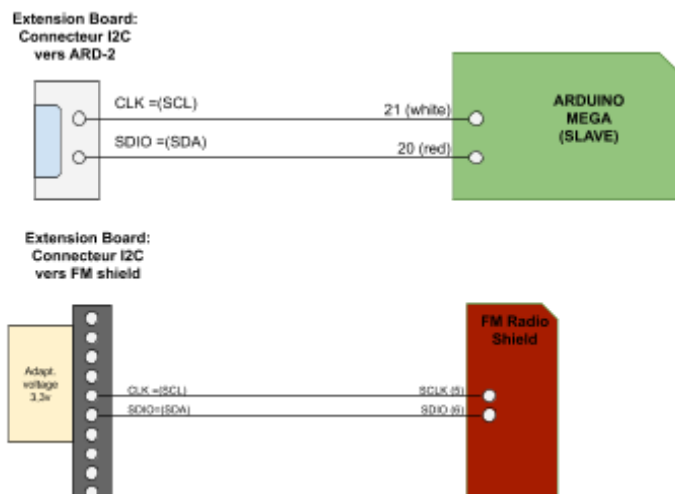
2.1. Les commandes I2C

Le bus I2C est utilisé par le Arduino 1 (Master) pour envoyer des commande d’affichage à l’Arduino 2 (Slave) pilotant un écran TFT, et pour piloter la carte FM RADIO.

Adresse TFT_SLAVE = **0x02**

Adresse FM-SHIELD = **0x10** (prédéfini dans le shield)

Label	Valeur	Paramètres	Action
C_BACKGROUNDIMAGE	2	néant	
C_CLEAR	3	néant	clear background
C_BL	4		TFT back light
C_ICON1	5	texte	display small icon (77x79)
C_ICON2	6	néant	display large icon “dance”
C_TITLE	12	texte	
C_ARTIST	13	texte	
C_ALBUM	14	texte	
C_YEAR	15		
C_GENRE	16	texte	
C_STARS	17		
C_LOG	18	texte	



3. Câblage

3.1. Câblage Ecran TFT

Arduino MEGA	Cable	Ecran TFT	Commentaire
GND	[=]	GND	
23	[0]	BL	<i>BackLight active LOW (normalement sur pin PWM)</i>
25	[1]	Reset	
27	[2]	D/C LD	<i>Data/Command (avec pullup)</i>
29	[3]	CS SD	<i>SPI SD Slave Select (avec pullup)</i>
31	[4]	CS LD	<i>SPI TFT Slave Select (avec pullup)</i>
51	[5]	MOSI	
52	[6]	SCK	
50	[7]	MISO	
5v	[+]	+5v	

3.2. Lexique Bus SPI

SPI = Serial Peripheral Interface

MOSI = Master Out - Slave In

MISO = Master In - Slave Out

SCK = Serial Clock (Master)

SS = Slave Select (Active Low). Doit être mis en Output+HIGH sur le Master.

3.3. Câblage bus SPI

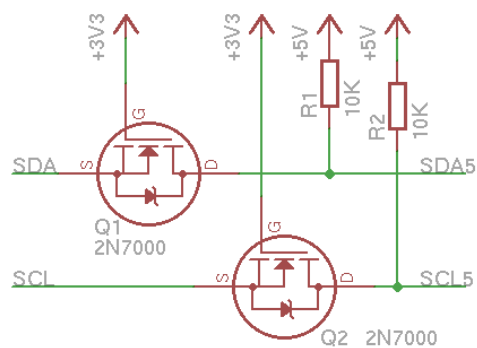
Arduino MEGA	Câble	MP3 Sparkfun Shield	Commentaire
2 (ISR0)		2	<i>MP3 DREQ</i>
NC	N/C	3	<i>MIDI-IN</i>
4		4	<i>GPIO1</i>
6	pullup	6	<i>MP3 CS</i>

7	pullup	7	MP3 DCS
8	pullup	8	MP3 RST
9	pullup	9	SD CS
51		11	MOSI
50		12	MISO
52		13	SCK
53	N/C Output LOW	N/C	N/C

3.4. SelfReturn Button

Arduino MEGA	Cable	Shield	Commentaire
3 (ISR1)	orange	1	Push
18 (ISR5)	rouge	2	Turn CW
GND	marron	3	Ground
19 (ISR4)	jaune	4	Turn CCW

3.5. Adaptation de niveau pour bus I2C



Sur le Bus I2C, nous avons un device slave en 5v (Arduino Mega) et un device slave en 3.3v (FM breakout). Cela nécessite une adaptation de niveau sur le bus I2C (shift level) avec deux transistors MosFet.

- 2N7000
- BS170

4. Conception hardware


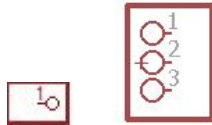




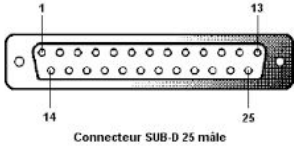
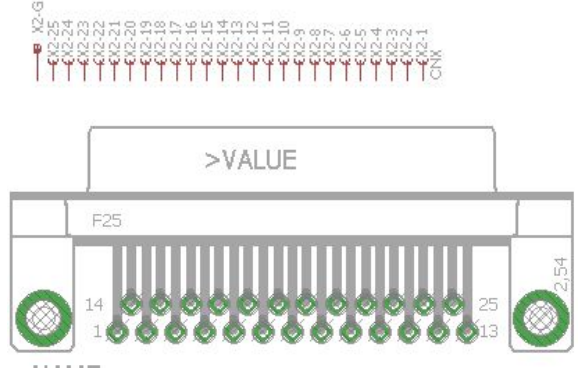
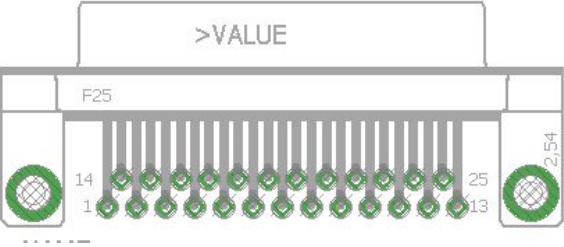

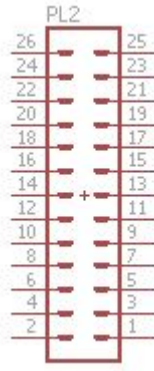
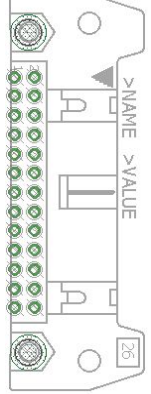
4.1. Logiciels

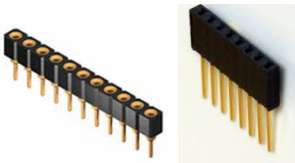
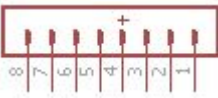





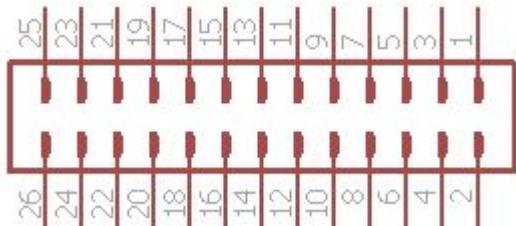

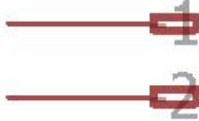
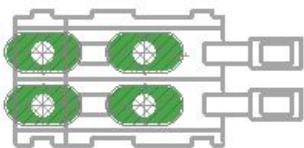


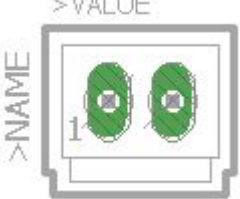

Logiciels utilisés pour la conception des circuits imprimés.

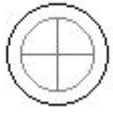

- 2015: Eagle (dans sa version gratuite: Eagle Free Edition)
- 2020 : KiCad (freeware open source)

4.2. Librairies et Composants utiles

Pour les connecteurs, on peut utiliser:

Composant (Photo)	Libellé Eagle (librairie)	Schéma	Layout
 Pin Header	PINHD-1x1 PINHD-1x3 (library pinheader)		
 Connecteurs Molex 254 KK series	22-27-2031-03 (library con-molex)		
 DB25 Sub-D connecteur	F25HP (library con-subd)		
 Connecteur H10-26 (library con-harting-h) Disponible à www.reboul.com			

<p>Female Header</p> 	<p>FE08-1 (library cons-1sta)</p>		
<p>Male header</p> 	<p>MA08-1 (library con-1stb)</p>		
<p>Dual Male Header</p> 	<p>MA13-2 (library con-1stb)</p>		
<p>Terminal Block Wago 234-102</p> 	<p>234-102 (librarycon-wago254)</p> <p>Non recommandé: les leviers sont trop durs.</p>		
<p>Screw Terminal Block Phoenix</p> 	<p>C-GRID-02-70543 (library con-molex)</p>		
<p>3x2 IDC connector</p> 			

	Holes		
	VCC, GND (library supply)		

4.3. Paramétrage pour l'édition des PCB

Note sur les unités:

Deux unités sont utilisées: les millimètres (**mm**) et les millièmes de inch (**mil**). Attention à ne pas confondre.

0.4mm = 16 mil

0.8mm = 32 mil

Width (largeur des pistes)

La largeur des pistes dépend de la façon dont les circuits seront fabriqués.

default : width=0.4 mm (réalisation artisanale) ou 0.3mm (réalisation industrielle)

power : width=0.8 mm (réalisation artisanale) ou 0.5mm (réalisation industrielle)

Réalisation artisanale: fabrication par UV et perchlorure.

Réalisation industrielle: fabrication par machine.

Paramétrage Eagle: Edit → Net Classes

Puis, éditer les *properties* des *wires* voulus dans le *schematic*, pour leur donner la *net class* "power".

Clearance (distance entre pistes):

réalisation artisanale:

réalisation industrielle: 2 mm

Drill (perçage pour les pins des composants):

drill = 32 mil = 0.8 mm (résistances, capa, transistors, CI)

drill = 35 mil = 0.9 mm (headers, molex, diodes)

Paramétrage Eagle: Edit → Net Classes

Paramétrage KiCad: Se fait pour chaque empreinte de composant (*footprint*)

Mounting Holes (trous de perçage pour les vis de fixation):

drill = 2.8mm = 110 mil = 0.110 inch

Vias:

perçage (drill) = 32 mil = 0.8 mm

largeur de la pastille (diamètre) = 0.07

Si on doit les souder manuellement: préférer les straps plutôt que les vias, positionner les vias éloignés des pistes et des autres vias (clearance).

4.4. Script Eagle pour Export des Layouts

Le script **Export-layers.scr** génère les trois images suivantes:

Components.png Image avec la disposition des composants (top view)

Layer-Top.png Image avec les pistes (top view)

Layer-Bottom.png Image avec les pistes (bottom view)

Les images sont générés dans: **Mes Documents\leagle**

Les images sont en négatif (fond noir). Avant de les imprimer, il faut inverser les couleurs avec la commande **InvertImage.bat**. (nécessite l'installation préalable de l'utilitaire **convert** de *ImageMagick*)

4.5. Debug

Pour la mise au point, il est possible de connecter:

- un câble USB pour debug à l'arduino Master
- un câble USB pour debug à l'arduino Slave
- et d'alimenter le l'Arduino Slave en 12v à partir du Master

5. Cartes spécifiques

5.1. Présentation

Alimentation Board

Cette carte est alimentée par l'alimentation générale. Elle répartit l'alimentation vers les différents composants:

- carte Arduino Master (5v)
- carte Arduino Slave (5v)
- Ampli (12v)
- Lampes de la face avant (5v)

Bus Board

La carte Bus collecte les différents E/S utiles de l'Arduino, et les envoie sur une nappe de 25 points vers la carte Extension. Cette carte se plugge directement sur l'Arduino Master.

Note: Trois connexions (3V3, D3 et RST) sont mal positionnées et gênent la carte MP3.

Extension Board

Cette carte comporte des connecteurs d'entrée/sortie pour les différents capteurs de l'Arduino (boutons, Leds, ...) avec leurs adaptations, et pilote les relais pour la sélection de l'audio (Audio issue de la carte MP3, ou de la carte Radio-FM).

Amélioration possible: Regrouper A12 et A14 sur un même connecteur (allant vers la capa à mesurer).

Audio Board

Cette carte fait passer le signal audio par le préamplificateur, le potentiomètre de volume et l'amplificateur.

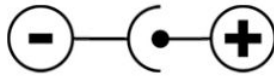
5.2. Carte Alimentation Board

Entrées:

- **POWER IN** (VCC 12v)
- **Bouton ON/OFF**
- **Jumper**: Permet d'alimenter les Arduino en VCC (12v) ou en +5v

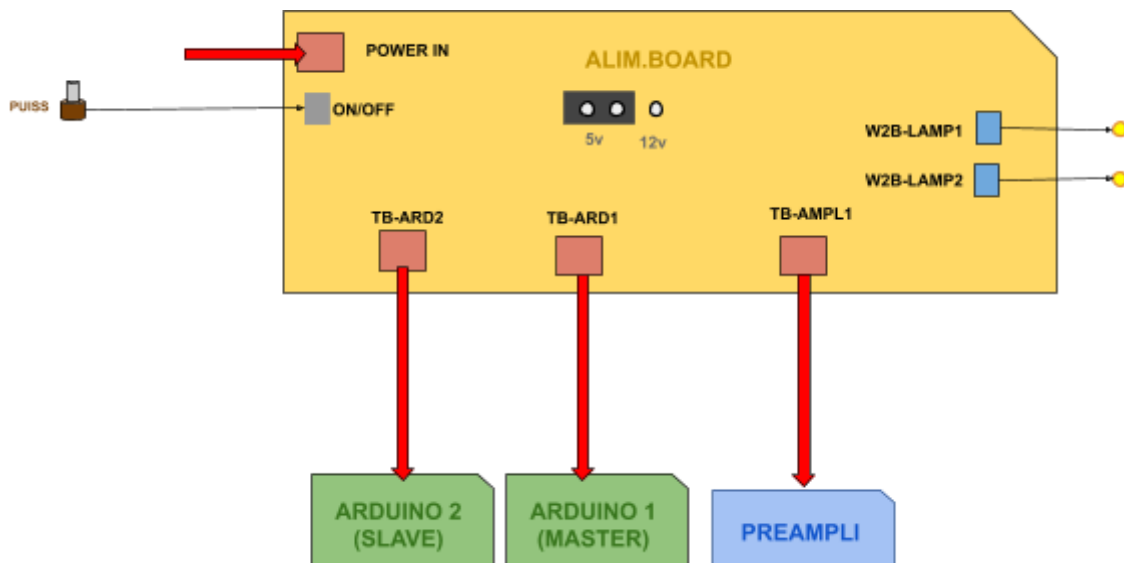
Sorties:

- **W2B-LAMP1** : Alimentation Lampe 1 (+5v)
- **W2B-LAMP2** : Alimentation Lampe 2 (+5v)
- **TB-AMPLI1** : Alimentation Ampli (12v)
- **TB-ARD1**: Alimentation pour carte Arduino 1
- **TB-ARD2**: Alimentation pour carte Arduino 2



Note: le culot des lampes est déjà à la masse (chassis).

W2B = Connecteur Wire-to-Board.



5.3. Carte Bus Board

Connexions:

- Plug de la carte Arduino Mega
- Plug de la carte MP3 Shield
- Connecteur Nappe 25 points vers Extension board
- Connecteur Alim 2 points vers Extension board
- Connecteur Reset (bouton poussoir)

Signaux Arduino:

- Entrées Analogiques A8-A15
- Entrées/Sorties Digitales: (inclut: D1-D53, le bus I2C, le bus SPI, et les entrées avec ISR)
- Lignes POWER: (inclut: GND et VCC 5v, 3v3 et RESET)

Sorties:

- Nappe 25 points vers la carte d'Extension (attention, le connecteur a 26 points)
- Connecteur Power pour les E/S carte d'Extension (5v - 0v). *Aurait dû être intégré à la nappe.*

Modifications effectuées en V2:

- Modification du câblage de RST* (Shield FM pin 8) et SEN* (Shield FM pin 7):
 - Avant:
 - [Arduino] D23 ----[Bus Board] X1-11 ----[Ext board] FM_SEN --- [Si7403] pin7
 - [Arduino] RST ---[Bus Board] X1-24 ----[Ext board] FM_RST --- [Si7403] pin8
 - Après:
 - **NC** ----[Bus Board] X1-11 ----[Ext board] FM_SEN --- [Si7403] pin7
 - **[Arduino] D23** ---[Bus Board] X1-24 ----[Ext board] FM_RST --- [Si7403] pin8
- Suppression du bus SPI de la nappe 25 points:
 - Avant:

■ [Arduino] MISO	[Bus Board] X1-14	[Ext board] MISO
■ [Arduino] MOSI	[Bus Board] X1-1	[Ext board] MOSI
■ [Arduino] SCLK	[Bus Board] X1-2	[Ext board] SCLK
 - Après:

■ [Arduino] A14	[Bus Board] X1-14	[Ext board] A14 (capa sensor)
■ [Arduino] GND	[Bus Board] X1-1	[Ext board] GND (capa sensor)
■ [Arduino] A12	[Bus Board] X1-2	[Ext board] A12 (capa sensor)
- Ajout du Reset:
 - Avant:

■ [Arduino] D37	[Bus Board] X1-20	[Ext board] JP1 (spare)
-----------------	-------------------	-------------------------
 - Après:

■ PushButton	[Bus Board] X1-20	[Ext board] JP1 (reset)
--------------	-------------------	-------------------------

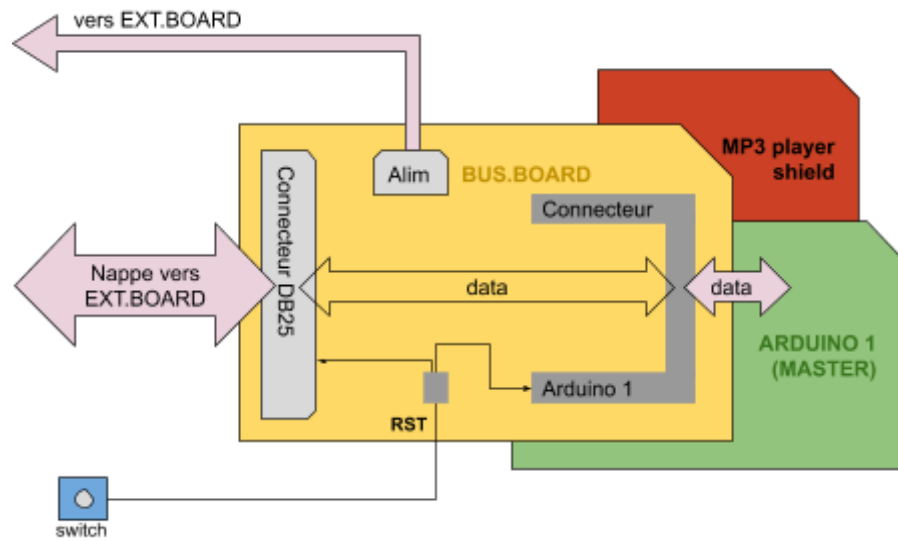
- Libération de pins:

- Avant:

- [Arduino] A14 [Bus Board] X1-16 [Ext board] TUNE
 - [Arduino] A12 [Bus Board] X1-17 [Ext board] TUNE

- Après:

- [Arduino] D49 [Bus Board] X1-16 [Ext board] Spare2
 - [Arduino] D37 [Bus Board] X1-17 [Ext board] Spare1



5.4. Extension Board

Interfaces:

- **X1**: Nappe 25 points DB25 vers la Carte Bus Board
- **CNX-ALIM**: +5v et GND
- Carte Sparkfun FM Shield Si7403

Bus d'Interfaces:

- **CNX-SPI** : Bus SPI (+ connecteur de test)
- **C-I2C** : Bus I2C (+ connecteur de test)

Entrées:

- **CMODE**: commutateur rotatif 4 positions (3 pour MODE, 1 pour FM/MP3)
- **C-FM-MP**: commutateur 3 positions (FM / MUTE / MP3)
- **CNX-PUSH**: commutateur avec retour et poussoir.
- **CNX-TUNE**: entrée analogique pour lecture Capa variable (schéma: C_TUNE1)
- **CNX-SPARE**: entrée analogique pour lecture Capa variable (Schéma: C_TUNE2)
- **C-AUDIO-IN**: entrée audio 2 voies (carte MP3).
- **FM-SCT** : info STC (Seek/Tune Completed) du shield FM Si7403 (D37 → JP1 → GPIO2)

Sorties:

- **LED1** : LED jaune (témoin d'activité carte SD)
- **LED2** : LED rouge (spare)
- **C-AUDIO-OUT**: Sortie audio 2 voies vers Ampli.
- **JP5** : Sortie Spare (pour une sortie de type LED) avec résistance.

Améliorations:

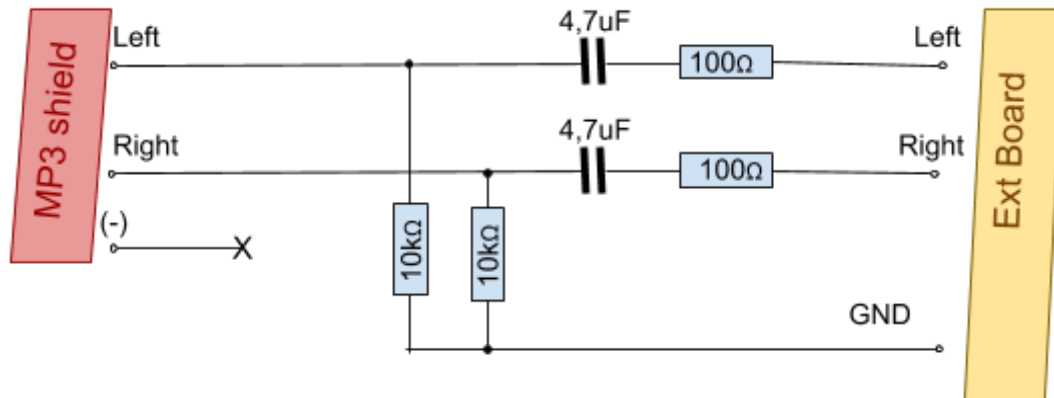
- Regrouper A12 et A14 sur un seul connecteur Molex-2pos (sans Vcc ni Gnd) pour la lecture de la capacité variable.
- Connecter *JP1* à la pin *GPIO2* du shield FM "Si7403". Cela sert pour les pulses STC utilisés par la librairie du shield Si7403. => *fait par un strap, sur la carte actuelle.*

Adaptateurs E/S:

Lors de l'assemblage, il s'est avéré nécessaire de réaliser différents adaptateurs d'entrée/sortie qui n'avaient pas été prévus lors de la conception. Le mieux aurait été de les intégrer directement sur la carte.

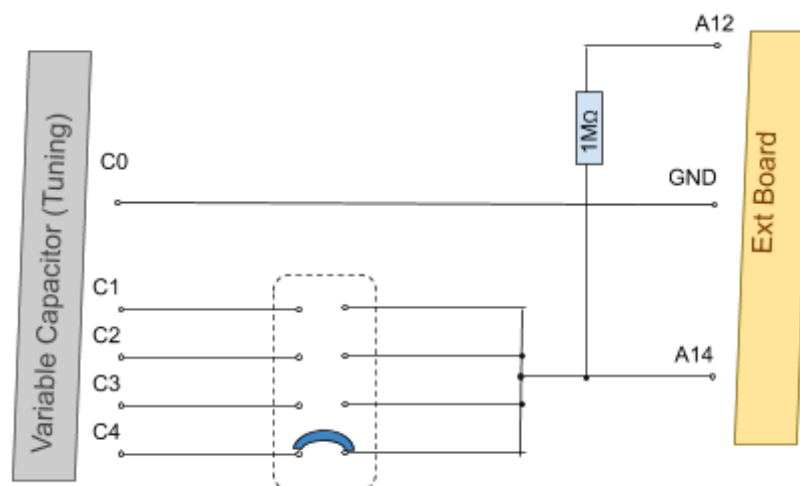
- **Adaptateur: Extension Board ◀▶ MP3 player shield**

Recommandé par la doc du shield Sparkfun.



- **Adaptateur: Extension Board ◀▶ Capa (Tune)**

Permet de sélectionner une (ou deux) capas parmi les 4 que comporte la capacité variable du Tuning.



5.5 Carte Audio board

Description

Cette carte envoie le signal audio entrant vers:

- un Préamplificateur stéréo
- un étage de fusion stéréo → mono
- un Potentiomètre de contrôle de volume (4.7 k Ω)
- un Amplificateur mono
- un Haut-Parleur 8 Ω

Remarque

L'étage de fusion **Stéréo** → **Mono** ne fonctionne pas: les signaux L et R s'annulent réciproquement. Après avoir essayé différentes valeurs de R1 et R2 (de 100 Ω à 10k Ω), seule la voie gauche est conservée.

5.6 Carte MP3 Player (Sparkfun)

Cette carte comporte un player MP3 et une carte SD. Elle est alimentée en 5v.
La carte est disponible chez [Sparkfun](#) (Produit [12660](#)).

Connectique :

Elle se plugge directement sur l'Arduino.

- Chip select SPI: module SD, module player MP3, MP3 data
- Bus SPI (SCLK, MISO et MOSI)
- Alimentation 5v

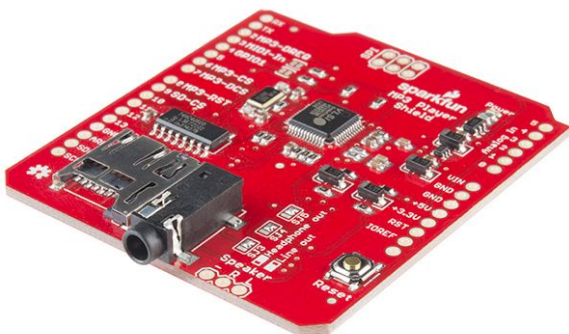
MISO	MOSI	SCLK	MP3 CS	MP3 data CS	SD CS	Data Request	MIDI IN	GPIO	RESET
Bus SPI	Bus SPI	Bus SPI	SPI select MP3 player	SPI: select data for MP3	SPI: select SD reader		N/U	N/U	Reset

Software:

Pour utiliser le shield, il faut ajouter au code:

- La librairie **SPI** (gestion du bus SPI)
- Les librairies **SDfat** et **SDfatUtil** (gestion du lecteur SD)
- La librairie **SFEMP3Shield** de William Greiman (gestion du chip player mp3)

Pour les tests, cette carte peut être simulée par *bouchon.h*



5.7 Carte LCD Screen (Arduino)

Cette carte comprend un écran LCD couleur 160x128, et une carte SD. Elle est alimentée en 5v. C'est une [carte native Arduino "Robot LCD"](#).

Connectique :

Cette carte comprend un connecteur 10 pins:

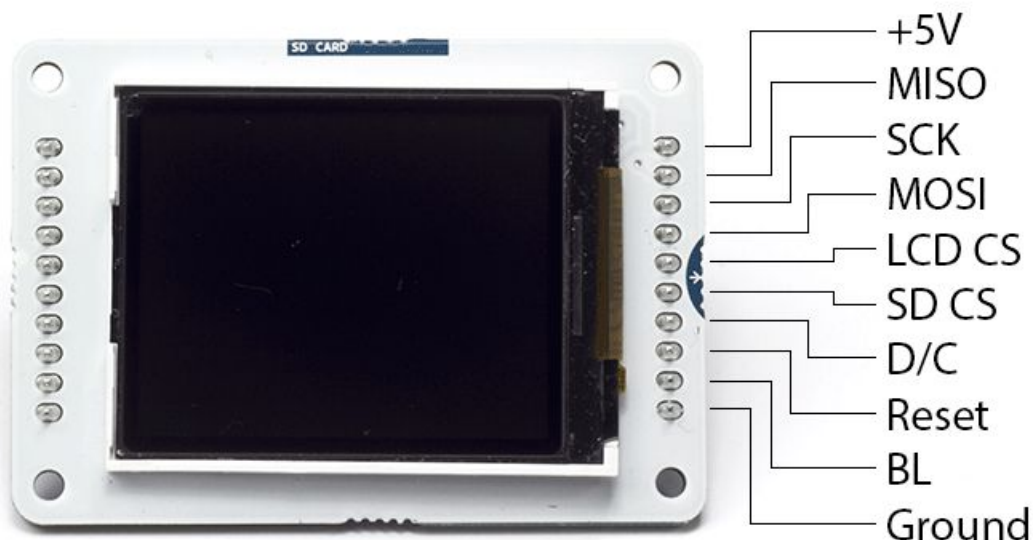
- Bus
- Chip Select : Écran LCD, Lecteur SD
- Rétro-éclairage
- Alimentation 5v.

+5v	MISO	SCK	MOSI	LCD CS	SD CS	D/C	Reset	BL	Ground
Power	Bus SPI	Bus SPI	Bus SPI	SPI: select LCD screen	SPI: select SD reader	Data or Com-mands	Reset	Backlight	Ground

Software:

Pour utiliser la carte, il faut ajouter dans le code:

- La librairie **SD** pour la gestion des fichiers de la carte SD.
- La [librairie TFT](#) pour l'affichage des images BMP



5.8 Carte Radio FM (Sparkfun Si4703)

- NON UTILISÉ - PAS RÉUSSI À LA FAIRE FONCTIONNER AVEC UN ARDUINO MEGA...

Cette carte comporte un tuner FM. Elle est alimentée en 3.3v.

La carte est disponible chez [Sparkfun](https://www.sparkfun.com/products/11083) (Produit [11083](https://www.sparkfun.com/products/11083)).

Connectique :

La carte comporte un connecteur 10+1 pins.

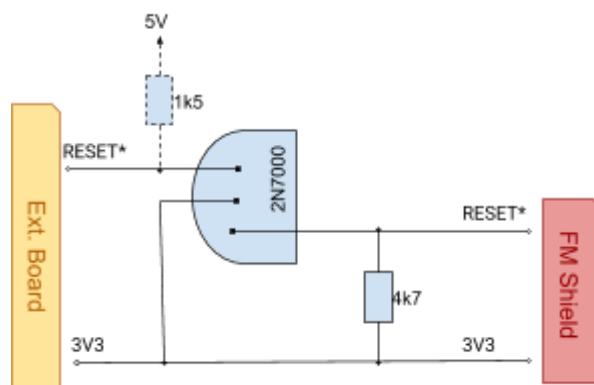
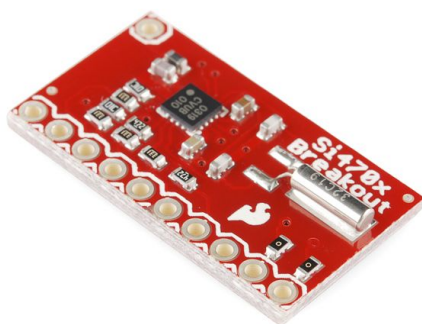
- Broche isolée = Antenne FM. Fil d'antenne nécessaire.
- Alimentation 3v3
- Bus I2C (adresse 0x10) + sorties Audio Stéréo + bits de contrôle.

L OUT	R OUT	GND	VCC	SDIO	SCLK	SEN*	RST*	GPIO1	GPIO2
audio	audio	ground	3,3 volts	bus I2C	bus I2C	N/U	Reset	N/U	info STC

Software:

On utilise les librairies:

- Librairie **Wire** pour le bus I2C
- Librairie **SparkFunSi4703** (modifiée).



Remarques:

Lors de la réalisation, il n'a pas été possible d'obtenir d'autre réponse que NACK à tout message envoyé via le bus I2C.

- Un convertisseur 5v→3v3 pour SDIO et SCLK existe sur la carte Ext Board.
- Un même convertisseur a été ajouté pour RESET et GPIO2 (cf ci-dessus): sans résultat. (A noter qu'il existe des composants LevelConvertor qui font ces conversions).
- Test unitaire avec une carte Arduino UNO: la carte FM fonctionne bien sans aucune adaptation de tension, mais retourne NACK si on en met. Malheureusement, avec Radial-V, on reçoit NACK même sans aucune adaptation.

=> Le pilotage de cette carte a été mis en Stand-by....

5.9 Carte PreAmplifier (Velleman K2572)

- NON UTILISÉ - GÉNÈRE UN SIGNAL TROP PUISSANT...

Cette carte est un étage préamplificateur audio stéréo, alimenté en 12v.
Le réglage du niveau de sortie se fait via deux résistances ajustables.
C'est un kit à monter soi-même de Velleman ([kit K2572](#))

Gain = 40db.

Impédance de sortie = 1 k Ω

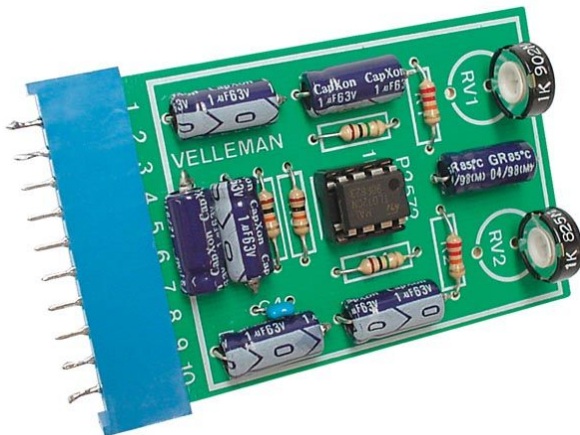
Connectique :

La carte comporte un connecteur 10 pins.

1	2	3	4	5	6	7	8	9	10
Ground OUT Left	Audio OUT Left	Ground IN Left	Audio IN Left	Ground Power Supply	+ Power Supply	Audio IN Right	Ground IN Right	Audio OUT Right	Ground OUT Right

Software:

Aucun.



5.10 Carte Amplificateur (Velleman)

Cette carte est un étage amplificateur audio mono, alimenté en 12v.
C'est un kit Velleman ([kit VM114](#))

Puissance de sortie RMS = 2W / 8Ω

Gain = 86dB

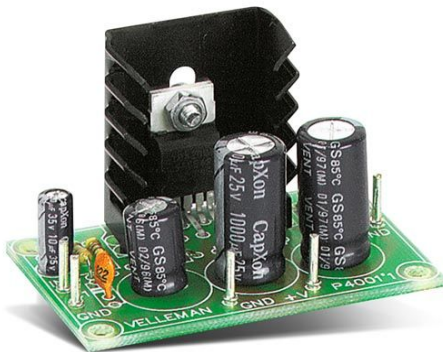
Connectique :

La carte comporte 6 connecteurs.

GND	+V	GND	IN	GND	LS
POWER 0v	POWER 12v	AUDIO IN Ground	AUDIO IN signal (Pot. volume)	AUDIO OUT Ground (Haut parleur)	AUDIO OUT signal (Haut parleur)

Software:

Aucun.



Remarque:

La qualité du son est très sensible à la qualité et à la stabilité de l'alimentation.
Les câbles audio doivent être courts et blindés.

5.11 Lecture de la valeur de la Capa variable

On utilise un algorithme basé sur la mesure du temps de décharge.
Deux pattes de la capacité sont reliées à 2 pins analogiques de l'Arduino.

Pin "Charge" : On envoie des créniaux de 5 volts sur cette pin.

Pin "Discharge":

Etat OUTPUT: qui revient à mettre une grosse résistance.

Etat INPUT: (faible résistance + capacité parasite stray).

On lit la valeur de la tension au bout de 0.1ms.

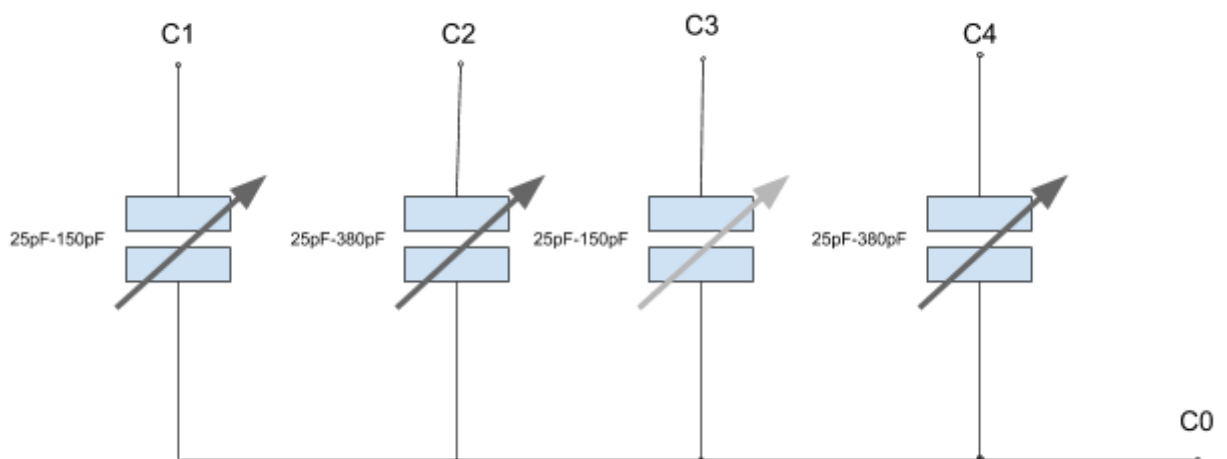
Les quatre capacités évoluent en même temps, de façon similaire, lorsque l'on tourne le bouton de Tuning.

On mesure la capacitance entre les broches C0 et C4.

On prévoit un circuit pour vider la capa: Relier momentanément les deux pins de la capa, via une résistance de 10 Ω . (non réalisé: trop complexe)

(Note: La valeur de C3 n'est pas très stable: on évite de l'utiliser).

Schéma de la capacité variable:



6. Fabrication

6.1. Réalisation des PCB

J'ai trouvé deux sites français qui proposent la réalisation de PCB.

<https://pcbshopper.com/> est un comparateur de fabricants.

- <http://bxtronics.free.fr>
e-tronic est limité à des PCB simple-face, en 35 et 70µm.
Ses tarifs sont de 14€/dm² + outillage (5€) + 100 trous (3.50€) + port (3.50€)
Perçage = 0.8mm par défaut.
Sérigraphie = non
Étamage = gratuit (à froid)
Découpe = gratuit
- http://www.util-pocket.com/electronique/circuit_imprime.htm
util-pocket fait simple-face et double face, en 35µm.
Ses tarifs sont de 14.50€/dm² + outillage (3.60€) + port (4.10€)
Perçage = 0.8mm par défaut.
Étamage = gratuit (à chaud).
Vias (trous) non métallisés.
- <https://jlcpcb.com>
Entreprise chinoise aux tarifs très compétitifs.
Réalise les PCB par 5 unités minimum.
Ses tarifs sont de 5€/dm² + port (8€)
Perçage inclus (fournir fichier drill).
Sérigraphie incluse.
Vias métallisés.

6.2. Retour d'expérience

PCB réalisés par util-pocket:

Délai : 4 semaines.

Qualité OK.

Pas de sérigraphie. Perçage à refaire. Vias à souder.

Tarifs: 36 € au total pour les 3 boards.

Fabrication à partir des fichiers .brd.

PCB réalisés par jlc-pcb

Délai : 6 semaines.

Qualité OK.

Sérigraphie. Perçage fait selon drill-map. via métallisés.

Tarifs: 8€ au total pour 2 boards.

Fabrication à partir des fichiers gerber et drill.

Le perçage (drills)

Util-pocket a percé tous les trous en 0.8mm. Certains nécessitent du 1mm (headers, connecteurs...), que j'ai dû repercer. C'est pas très grave.

JLC-PCB utilise le fichier .drill pour percer chaque trou avec le diamètre demandé.

les Vias:

Util-pocket les perce en 0.6mm : c'est OK.

Par contre la largeur de cuivre des pastilles que j'avais défini, était à 0.4mm : c'est trop petit. Il aurait fallu 0.6mm. Le métal se décolle si on chauffe trop. La soudure du via est difficile.

JLC-PCB les perce en 0.4mm: c'est OK car ils sont déjà métallisés et ne demandent aucun post-traitement (ni soudure).

Autres points:

- Ne pas mettre 2 vias trop proches. (par ex: 1mm c'est trop proche). Positionner **Clearance Via-Via** à 64mil (1.6mm), par exemple, dans Design Rules.
- Favoriser deux vias formant un "pont" où l'on pourra mettre un cable (strap).
- Ne pas faire de connexion "fil-carte" **à un seul point** avec un header: le fil se casse.
- Mettre un Test Point (header 2 pins) "+VCC/GND" sur chaque carte pour favoriser les vérifications.
- Eviter de mettre de la sérigraphie sur les pistes. C'est peu lisible.
- Sérigraphie: prendre des caractères de dimensions 1mm ou 1.2mm (la dimension par défaut de KiCad 0.8mm est trop petite).

7. Utilisation de Radial-V

7.1. Carte SD pour les MP3

Structure de la carte SD

Les fichiers MP3 sont sur une carte SD (16 GB) formatée en **XXX**.

Répertoire racine = fichier **Catalog.ndx**

Répertoire **Music** = fichiers mp3 (dont NOISE.MP3)

Contrainte du driver SD : les noms de fichiers doivent être en 8+3

Fichiers particuliers

Fichier **NOISE.MP3** = fichier de quelques secondes, joué lorsque l'application n'a pas réussi à trouver le fichier mp3 à jouer.

Fichier **STARTER.MP3** = premier fichier joué lors de la mise en route

Fichier **Catalog.ndx** = fichier index dont la préparation est décrite ci-après.

Structure du fichier Index

Le fichier est trié selon les années.

Le format de chaque ligne est le suivant:

YEAR[4];HASCHCODE[8];GENRE;RATING[1];

Exemple:

1927;E72C4A38;Ragtime;3;

Cas particuliers:

0000 : année inconnue

<ERROR> : le hash n'a pas pu être calculé

0 : rating non renseigné

Préparation de la carte SD

Les MP3 doivent être copiés sur une carte micro-SD accompagné d'un fichier index.

Un utilitaire permet de générer cet index, et de copier les fichiers MP3 sur la carte SD en les renommant en 8.3.

- Lancer l'utilitaire **mp3tag**
- Sélectionner l'ensemble des clips mp3 à mettre sur la carte SD (Ctrl-A).

- Cliquer sur **Export**, et choisir **radial_catalog**
- Ceci génère un fichier **catalog.txt** (contenant: Année, Filename, Genre et Rating)
- Cliquer sur **Export** et choisir **radial_transfert**
- Ceci génère un fichier **TransfertFiles.txt** (au format UTF-16).

Pour bien gérer les accents dans les filename, ce fichier de commande a besoin d'être converti.

- Exécuter **ConvertTransfertBatch.bat** qui génère **TransfertFiles.bat**.

Conversion en CP850 avec la commande:

```
iconv -f UTF-16 -t CP850 music\TransfertFiles.txt > TransfertFiles.bat
```

Pour copier les fichiers sur la carte SD:

- Exécuter **TransfertFiles.bat**:
- Ce script génère le catalogue final **Catalog.ndx**
 - Tri du catalog selon le champ Year
 - Copie les fichiers mp3 sur la carte SD (I:\Music) en les renommant.
 - Copie le catalogue sur la carte
 - Copie quelques bruitages sur la carte SD (NOISE.mp3, par exemple)

Note: le passage par utf-16 permet de convertir correctement les noms de fichiers qui comportent des caractères accentués. (le fichier batch devant comporter des caractères au format CP850).

Nécessite de disposer de l'utilitaire **iconv.exe**

Note sur l'extraction des Covers

L'utilitaire MP3TAG permet aussi d'extraire les images (*Art cover*) des fichiers MP3. Cela ne se fait pas par script, mais via l'IHM:

- Sélectionner des fichiers musicaux
- Menu Actions → Action (Quick) → Export Cover → *Donner le nom de fichier* → OK

Pour le nom de fichier, le format est le suivant:

- soit un nom fixe. Ex: "cover"
- soit un nom construit: \$validate(mypath\\$trim("%ARTIST%" - \$trim("%ALBUM%"))

Les paramètres utilisables sont:

- %ARTIST%
- %TITLE%
- %BAND%
- %YEAR%
- %ALBUM%
- mais malheureusement pas: %_crc%

8. Annexes: Utilitaires

8.1 Utilitaire CadSoft Eagle Schematic Design (Free Edition)

La Free Edition permet de designer des circuits électroniques de 10x10 cm. Complet et bien documenté sur le Net.

- Site web: <http://www.cadsoftusa.com/download-eagle/>
- Script **Export-Layers.scr**

8.2 Utilitaire ImageMagick

ImageMagick permet de traiter des fichiers image en ligne de commande (inversion de couleurs, resize, etc).

- Site web: <http://www.imagemagick.org/script/binary-releases.php>
- Script **InvertImage.bat**

8.3 Utilitaire Mp3Tag

MP3Tag permet de gérer les tags des mp3, et possède un système de scripting très poussé.

- Script **radial_catalog.mpe** (génère *Catalog.txt*)
- Script **radial_transfert.mpe** (génère *TransfertFiles.txt*)
- Site web: www.mp3tag.de/en/

8.4 Utilitaire iconv

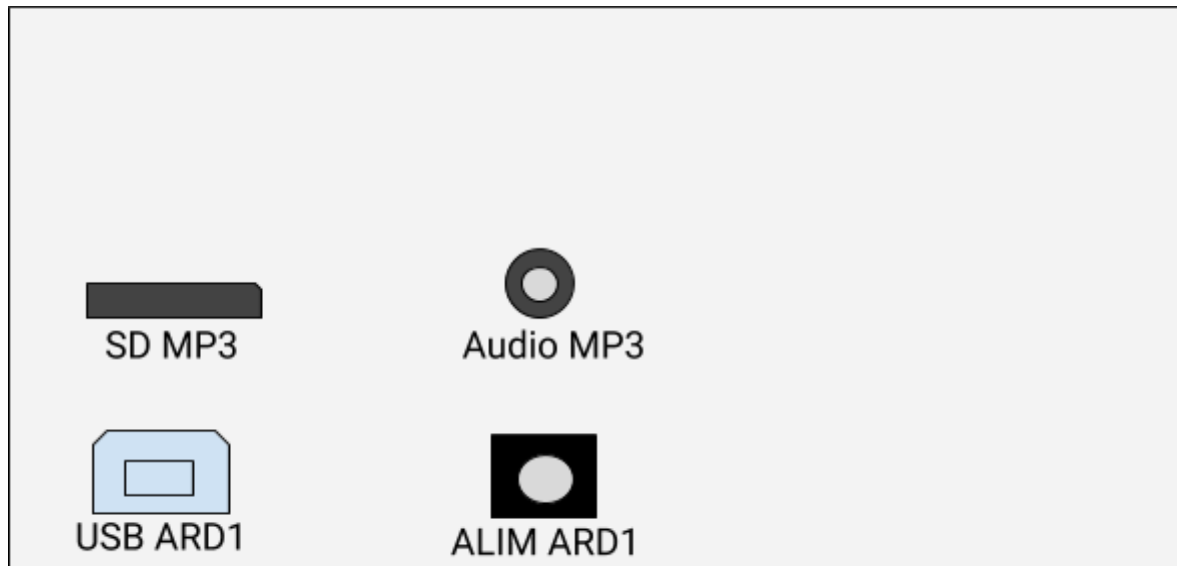
Cet utilitaire permet de convertir des fichiers ANSI en OEM, etc.

Utile pour gérer des caractères accentués dans des batch qui s'exécutent dans une fenêtre DOS en Code page 850..

Installer la commande *iconv* en téléchargeant *libiconv-1.9.2-1.exe* à partir de ce lien : <http://gnuwin32.sourceforge.net/downloads/libiconv.php>

- Script **ConvertTransfertBatch.bat**

9. Connectique



Port USB de debug ARDUINO 1 : Accès direct sur la carte en USB-C ♀ (115200 bauds).

Port USB de debug ARDUINO 2 : Câble de report USB-A ♂ (115200 bds).

Carte SD TFT: Accessible directement au niveau de l'écran TFT.

Ajustement de la couleur de background de l'écran TFT:

- se connecter sur le port de debug ARD2.
- Envoyer les caractères:
 - a - q : ajuste R
 - z - s : ajuste G
 - e - d : ajuste B
 - Entrée: envoi

10. Réalisation de la face avant (en plexi)

Réalisé par Negatif+ (17€ en promo).

Découpe du plexi: assez aisé avec la scie à chantourner. Mettre du scotch de protection pour avoir une découpe propre. Attention à ne pas mettre de scotch sur les parties coloriées: car cela enlève la peinture.

Ne pas mettre le tracé de découpe sur la réalisation. Cela bride la découpe à ce tracé et ne permet pas d'ajuster la découpe en fonction des besoins, si il y a besoin de s'adapter à un changement.

Note: Negatif+ a réalisé le plexi avec un changement d'échelle de environ 1,2 cm (6%). La raison probable est qu'ils ont légèrement redimensionné pour que le tracé de découpe ne dépasse pas, ce qui fut une erreur, car les marques de perçage ne correspondent plus aux position des boutons.