

CY IUT – GEII Neuville

NOTICE DE CONCEPTION

Orphéa

Document rédigé par : Lorenzo Salvatore & William Le Coënt
& Quentin Perbost

Ref : 030925_100_-_Notice_de_conception_Omphéa

ENREGISTREMENT DES MODIFICATIONS

Version	Date	Auteur	Modification
-	02/02/2026	Q.PERBOST	Création

APPROBATION

	Nom	Rôle	Signature	Date
Écrit par	Q. PERBOST	Technicien		
Vérifié par	L. SALVATORE	Technicien		
Approuvé par	W. LE-COËNT	Technicien		
Approuvé par le client si nécessaire	V. GAUTHIER	Client		

Avant-propos

Ce document porte sur la conception de la carte principale du projet Orphéa.
Il contient les solutions techniques retenues et leur justifications, ainsi que les notices de calculs des composants.

Documents de référence

Ci-dessous la liste des documents applicables à cette procédure.

ID	Référence
DOC_1	030925_06_-_Synoptique_SAE_S5.pdf

Table des matières

ENREGISTREMENT DES MODIFICATIONS.....	2
APPROBATION.....	2
Avant-propos.....	3
Documents de référence.....	3
Table des matières.....	4
1. Introduction.....	5
2. Choix des Micro-contrôleurs.....	6
3. Choix du Convertisseur Numérique Analogique I2S.....	8
4. Choix de l'amplificateur audio.....	9
5. Choix de l'étage d'alimentation.....	10
5.1 Contrainte d'utilisation.....	11
5.2 Consommation nominale.....	11
6. Choix des résistances de pull down pour les touches.....	11
7. Choix des valeurs de résistance pour les vérifications de tension.....	13
8. Choix des connectiques.....	14
9. Matrice de traçabilité des exigences.....	15

1. Introduction

La carte Orphéa est le centre névralgique du système, elle consiste à gérer la liaison entre tous les sous-systèmes du projet.

Elle gère les entrées/sorties, la génération du signal audio ainsi que son amplification.

cf : DOC_1

Afin de simplifier la gestion des différentes entrées et sorties, nous avons opté pour une architecture maître-esclave.

Le microcontrôleur esclave est chargé de scruter en permanence les 24 entrées. Dès qu'un changement d'état est détecté (passage à l'état haut d'une entrée), l'esclave transmet immédiatement l'information correspondante au maître via une liaison série, en utilisant le protocole MIDI.

Cette organisation permet d'associer directement chaque touche et chaque solénoïde au microprocesseur, ce qui réduit significativement la latence du système.

L'utilisation du protocole MIDI offre également une gestion plus structurée et plus fluide des événements, notamment pour le traitement de la polyphonie.

2. Choix des Micro-contrôleurs

Notre système nécessite un système de MAITRE - ESCLAVE

Cahier des charges :

Exigence	Rôle	Description	Résultat
$\mu\text{ESC}[1]$	ESCLAVE	24 Entrées	OK
$\mu\text{ESC}[2]$		24 Sorties	OK
$\mu\text{ESC}[3]$		1 Liaison Série RS232	OK
$\mu\text{ESC}[4]$		Tension d'alimentation comprise entre 3.3V et 12 V	OK
$\mu\text{ESC}[5]$		Programmable avec le framework Arduino	OK
$\mu\text{MAI}[6]$	Maître	Bus I2S	OK
$\mu\text{MAI}[7]$		BUS I2C	OK
$\mu\text{MAI}[8]$		5 Entrées Analogiques	OK
$\mu\text{MAI}[9]$		1 Liaison Série RS232	OK
$\mu\text{MAI}[10]$		WiFi	OK
$\mu\text{MAI}[11]$		Multi core	OK
$\mu\text{MAI}[12]$		RTOS	OK
$\mu\text{MAI}[13]$		Tension d'alimentation comprise entre 3.3V et 12 V	OK
$\mu\text{MAI}[14]$		Programmable avec le framework Arduino	OK

Pour répondre au besoin de la carte esclave, nous avons choisis la carte Arduino Mega Pro Mini, ci dessous les caractéristique principale :

- Microcontrôleur: ATMEL ATMeg2560
- Tension de fonctionnement: 5V
- Tension d'entrée sur la prise d'alimentation (ou broche VIN) recommandée : 7 - 12V
- Tension d'entrée sur la prise d'alimentation (ou broche VIN) limites : 6 - 20V
- Courant max par pin : 40 mA
- Courant continu pour les broches 3.3V : 50 mA
- Courant max total : 250 mA
- Mémoire Flash : 256 Ko dont 8 KB utilisés par bootloader
- SRAM : 8 KB
- EEPROM : 4 KB
- Fréquence d'horloge : 16 MHz
- 70 entrées/sorties numériques
- 4 UART hardware
- Framework Arduino

Pour répondre au besoin de la carte maître, nous avons choisis la carte ESP32-S3 DevKit C, ci dessous les caractéristique principale :

- Microcontrôleur: ESP-WROOM-32
- 2 interfaces I2C
- 2 interfaces I2C
- 3 interfaces UART
- Multicore : Core 0 / Core 1
- Tension d'alimentation : + 5 V
- Prise en compte de RTOS
- Programmable avec le framework Arduino
- 20 broches analogiques

Récapitulatif de la solution :

Microcontrôleur MAÎTRE : ESP32-S3 DevC-1-N16R8

Microcontrôleur ESCLAVE : Arduino Mega Pro Mini

3. Choix du Convertisseur Numérique Analogique I2S

Nous avons choisi d'utiliser un DAC fonctionnant en I2S afin d'améliorer la qualité de conversion numérique-analogique du signal audio. Contrairement à une sortie PWM (Pulse-Width-Modulation), le protocole I2S permet de transmettre un flux audio numérique synchronisé (données et horloges), garantissant une reproduction plus fidèle du signal réduisant le bruit et la distorsion.

Cette solution permet également de réduire la charge de calcul du microcontrôleur, puisque la génération précise du signal audio est prise en charge par le DAC, ce qui améliore la stabilité et la fiabilité globale du système.

Pour la génération des signaux sonores, nous avons besoin d'un DACI2S qui répond au besoin suivant:

Cahier des charges :

Exigence	Description	Résultat
DAC_I2S[1]	Résolution supérieur à 10 bits	OK
DAC_I2S[2]	Canaux Stéréo	OK
DAC_I2S[3]	Tension d'alimentation entre 3.3V et 5 V	OK
DAC_I2S[4]	Compatible avec ESP-32	OK
DAC_I2S[5]	Port jack pour sortie casque	OK
DAC_I2S[6]	Fréquence d'échantillonnage supérieur à 44100Hz	OK
DAC_I2S[7]	Faible distorsion harmonique	OK

Pour répondre au besoin, nous avons choisi pour le DAC I2S un UDA1334A, dont les caractéristiques principales sont :

- Type : DAC stéréo (conversion numérique → analogique) avec interface I2S
- Résolution : 24 bits
- Interface Audio : I2S (BCLK, LRCK, DIN)
- Sorties : Stéréo analogique (niveau ligne)
- Taux d'échantillonnage : Jusqu'à 96 kHz, certains modules indiquent jusqu'à 384 kHz
- SNR (rapport signal / bruit) : ~98 dB
- THD+N (distorsion harmonique + bruit) : < 0,01–0,02 %
- Alimentation : 3,3 V à 5 V
- Température d'exploitation : -40°C à +85°C
- Consommation : Très faible (~18 mW)
- PLL intégré : Oui, pas besoin de MCLK externe
- Contrôle Mute / De-emphasis : Oui
- Package : SSOP-16

4. Choix de l'amplificateur audio

En sortie du DAC I2S le signal n'est pas amplifié, il faut donc une amplification de ce dernier avant d'attaquer un haut-parleur. Notre solution devant être portable, nous avons fait le choix d'utiliser des haut-parleurs nécessitant peu de puissance.

Cahier des charges :

Exigence	Description	Résultat
AMP[1]	Compatible avec des Hauts-Parleurs entre 8 Ohms et 32 Ohms	OK
AMP[2]	Tension d'alimentation comprise entre 3.3 V et 12 V	OK
AMP[3]	Faible Consommation	OK
AMP[4]	Gain réglable	OK
AMP[5]	Taux de distorsion Harmonique faible	OK

Pour répondre au besoin nous avons choisi, pour l'amplification du signal audio, le LM386 dont les caractéristiques principales sont :

- Fonctionne de 4 V à 12 V (maximum 18 V).
- Gain interne de 200, ajustable de 20 à 200 avec des composants externes.
- 0,7 W à 8 Ω avec une alimentation de 9 V.
- 0,1 W à 0,2 W à 5 V pour des haut-parleurs de 8 Ω.
- Consommation en fonctionnement : ~4 mA à 9 V.
- Consommation en mode veille : < 1 µA.
- TDH < 1 % à 1 kHz à pleine puissance.
- Compatible avec des haut-parleurs de 8 Ω à 32 Ω
- Plage utile de 50 Hz à 100 kHz.
- Protection thermique interne pour éviter la surchauffe en cas de surcharge.

5. Choix de l'étage d'alimentation

Le système est connecté à une source d'alimentation de 24 Vdc.

Cahier des charges :

Exigence	Rôle	Description	Résultat
ALIM[1]	12 V	Abaïsser la tension de 24V vers 12V	OK
ALIM[2]		Intensité de sortie Supérieur à 1 A	OK
ALIM[3]	9 V	Régulateur de tension 9 V	OK
ALIM[4]		Intensité de sortie Supérieur à 500 mA	OK
ALIM[5]	5V	Régulateur de tension 5 V	OK
ALIM[6]		Intensité de sortie Supérieur à 300 mA	OK
ALIM[7]	3.3V	Régulateur de tension 3.3 V	OK
ALIM[8]		Intensité de sortie Supérieur à 300 mA	OK

*NOTA : Dissipation en W = (Vin – Vout) * Iout*

Pour abaisser la tension de 12 V vers 24 V, nous avons choisi d'utiliser un Buck-BOOST DC-DC, car la régulation de tension étant importante, la dissipation thermique sur un régulateur linéaire n'était pas suffisante.

Pour convertir le 12 V en 9 V, nous utiliserons un régulateur de tension linéaire de référence LM7809.

Pour convertir le 9 V en 5 V, nous utiliserons un régulateur de tension linéaire de référence LM7805.

Pour convertir le 5 V en 3.3 V, nous utiliserons un régulateur de tension linéaire de référence LM317.

Le choix des valeurs de condensateurs de découplage sont celles recommandées par le fabricant en page 22 figure 7 de la fiche technique applicable à la série des LM78XX.

Pour le LM317 le schéma et les valeurs des composants associés sont ceux indiqués par le fabricant page 9 figure 6 de la fiche technique applicable au LM317.

Les solénoïdes sont dimensionnés par le fabricant pour fonctionner avec une tension d'alimentation de 24 V et ne nécessitent pas d'étage d'alimentation dédié.

5.1 Contrainte d'utilisation

Suite à ces choix de conception adaptés à notre besoin, l'utilisation des alimentations est limitée en intensité aux valeurs suivantes :

Tension IN (en V)	Tension OUT (en V)	Intensité (en mA)	Puissance dissipé (en W)	Puissance Max (en W)
12	9	0,51	1,53	1,54
9	5	0,385	1,54	1,54
5	3,3	0,9	1,53	1,54

Ces valeurs sont données à titre indicatif, pour des régulateurs considérés parfaits et sans solutions adaptées de dissipation thermique.

5.2 Consommation nominale

Ci-dessous les consommations du système dans son état nominal.

SANS UTILISATION DES EFFETS AUDIOS

	Buck Boost	LM7809	LM7805	LM317
Arduino Mega		200 mA		
ESP 32			100 mA	
DAC I2S			20 mA	
SSD1306			50 mA	
24 Touches				16.8 mA
MAXIMUM	3A	510 mA	385 mA	900 mA
TOTAL	386.8 mA			

Calcul de l'échauffement : $Tj = Tambiente + (Pd * Rthja)$

$Tj = 25^\circ C$

Hypothèse courante = $Rthja = 65^\circ C/W$

	Buck Boost	LM7809	LM7805	LM317
Consommation mA	382	200	170	16.8
Rendement	90%	75%	55%	66%
Puissance à dissiper	0.458 W	0.6W	0.68W	0.204
Température	+22.9°C	+39.0°C	+44.2°C	+13.3°C

6. Choix des résistances de pull down pour les touches.

Nous avons besoin de piloter 24 touches, donc 24 pull down. Les boutons sont reliés au réseau 3.3V.

L'objectif des résistances de pull down est de limiter le courant en entrée de l'Arduino MEGA lorsqu'un bouton est appuyé.

Dans un contexte de réemploi de matériel, nous utilisons des réseaux de résistance pour limiter le nombres de composants, la valeur choisie est 4.7 kOhm

Soit l'intensité pour chaque bouton $I = \frac{3.3}{4700} = 0.7 \text{ mA}$

Dans le pire cas $I_{total} = 0.7 * 24 = 17 \text{ mA}$

7. Choix des valeurs de résistance pour les vérifications de tension

Nous avons mis en place 3 ponts diviseur de tension afin de mesurer les tensions à l'aide de l'ESP-32 avec une contrainte de conditionnement du signal à 3.3V pour ne pas endommager l'ESP-32.

$$\text{NOTA : } Ur2 = \frac{R2}{R2+R1} * Vin$$

Après calcul :

Tension In	R1 en (k ohm)	R2 en (k ohm)	Tension Out	Attendu ESP-32
12	27	10	3,243243243	3,3
9	18	10	3,214285714	3,3
5	5,1	10	3,311258278	3,3
3,3	--	--	3,3	3,3

Dans ce cas précis, il peut être utile de prendre des résistances +/- 1 % pour la fiabilité de la mesure.

8. Choix des connectiques

Afin de standardiser les connectiques, le choix s'est porté sur deux modèles.

D'un côté, le standard JST PH, choisi pour sa facilité d'utilisation, sa solidité et son bon maintien des connexions, et de l'autre, des connecteurs à broche standard pour les besoins plus importants en nombre de broches.

9. Matrice de traçabilité des exigences

Exigence	Description	Révision
$\mu\text{ESC}[1]$	24 Entrées	–
$\mu\text{ESC}[2]$	24 Sorties	–
$\mu\text{ESC}[3]$	1 Liaison Série RS232	–
$\mu\text{ESC}[4]$	Tension d'alimentation comprise entre 3.3V et 12 V	–
$\mu\text{ESC}[5]$	Programmable avec le framework Arduino	–
$\mu\text{MAI}[6]$	Bus I2S	–
$\mu\text{MAI}[7]$	BUS I2C	–
$\mu\text{MAI}[8]$	5 Entrées Analogiques	–
$\mu\text{MAI}[9]$	1 Liaison Série RS232	–
$\mu\text{MAI}[10]$	WiFi	–
$\mu\text{MAI}[11]$	Multi core	–
$\mu\text{MAI}[12]$	RTOS	–
$\mu\text{MAI}[13]$	Tension d'alimentation comprise entre 3.3V et 12 V	–
$\mu\text{MAI}[14]$	Programmable avec le framework Arduino	–
DAC_I2S[1]	Résolution supérieur à 10 bits	–
DAC_I2S[2]	Canaux Stéréo	–
DAC_I2S[3]	Tension d'alimentation entre 3.3V et 5 V	–
DAC_I2S[4]	Compatible avec ESP-32	–
DAC_I2S[5]	Port jack pour sortie casque	–
DAC_I2S[6]	Fréquence d'échantillonnage supérieur à 44100Hz	–
DAC_I2S[7]	Faible distorsion harmonique	–
AMP[1]	Compatible avec des Hauts-Parleurs entre 8 Ohms et 32 Ohms	–
AMP[2]	Tension d'alimentation comprise entre 3.3 V et 12 V	–
AMP[3]	Faible Consommation	–
AMP[4]	Gain réglable	–

AMP[5]	Taux de distorsion Harmonique faible	-
ALIM[1]	Abaïsser la tension de 24V vers 12V	-
ALIM[2]	Intensité de sortie Supérieur à 1 A	-
ALIM[3]	Régulateur de tension 9 V	-
ALIM[4]	Intensité de sortie Supérieur à 500 mA	-
ALIM[5]	Régulateur de tension 5 V	-
ALIM[6]	Intensité de sortie Supérieur à 300 mA	-
ALIM[7]	Régulateur de tension 3.3 V	-
ALIM[8]	Intensité de sortie Supérieur à 300 mA	-