**2IOTE**

**Documentation Technique**

**SUPFONE**



Une image contenant logo

Description générée automatiquement

**Réalisé par :**

**Mohamed Marwen Meddeb –** [**marwen.meddeb@supinfo.com**](mailto:marwen.meddeb@supinfo.com) **– B.Eng. 2**

Table des matières

[I. Introduction : 3](#_Toc132034202)

[1. Objectifs du projet : 3](#_Toc132034203)

[2. Technologies utilisées : 3](#_Toc132034204)

[II. Conception : 4](#_Toc132034205)

[1. Description des fonctionnalités : 4](#_Toc132034206)

[2. Schémas de câblage : 4](#_Toc132034207)

[3. Captures d'écran de la conception 5](#_Toc132034208)

[4. Codage : 7](#_Toc132034209)

[III. Réalisation : 11](#_Toc132034210)

[1. Explication des étapes de réalisation : 11](#_Toc132034211)

[2. Difficultés rencontrées et solutions apportées : 11](#_Toc132034212)

[3. Tests de validation des fonctionnalités : 12](#_Toc132034213)

[4. Résultat : 12](#_Toc132034214)

[IV. Conclusion : 14](#_Toc132034215)

# Introduction :

Le projet SUPFONE est un service de communication professionnelle inspiré du Nokia 3310 et des SMS. Il est conçu pour permettre aux utilisateurs d'envoyer et de recevoir des messages, de stocker des contacts, de personnaliser la musique et les paramètres.

## Objectifs du projet :

L'objectif principal de ce projet est de créer un service de communication professionnel qui permet aux utilisateurs d'envoyer et de recevoir des messages, de stocker des contacts, de personnaliser la musique et les paramètres en utilisant un Arduino, un écran LCD, un clavier, un buzzer et d'autres composants.

## Technologies utilisées :

Le projet utilise des composants électroniques tels qu'un Arduino, un écran LCD, un clavier, un buzzer et des boutons. Les messages sont stockés dans la mémoire même lorsque l'Arduino est éteint. Les FONE utilisent la communication I²C pour échanger des messages et recevoir des confirmations ou des erreurs. Les utilisateurs peuvent composer leur propre musique en utilisant plusieurs octaves et durées de notes différentes. Les paramètres tels que le volume et la luminosité peuvent être ajustés par l'utilisateur en utilisant les piézomètres.

# Conception :

## Description des fonctionnalités :

Le SUPFONE est conçu pour offrir plusieurs fonctionnalités telles que :

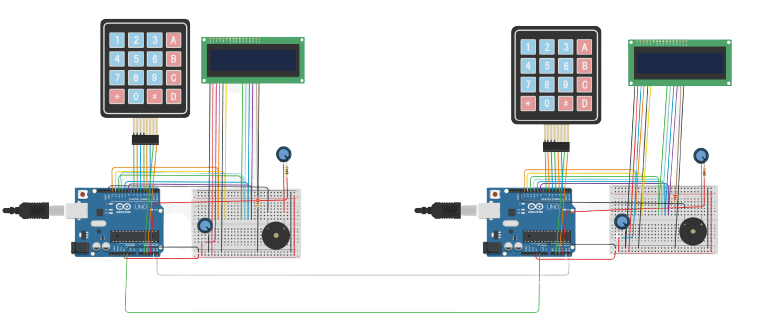
* Messagerie : Les utilisateurs peuvent envoyer et recevoir des messages en utilisant un clavier Nokia 3310 et insérer différents smileys. Les messages sont stockés dans la mémoire même lorsque l'Arduino est éteint.
* Gestion des contacts : Les utilisateurs peuvent ajouter, éditer et supprimer des contacts.
* Stockage : Les messages et les contacts sont stockés même lorsque l'Arduino est éteint.
* Communication : Les FONE utilisent la communication I²C pour échanger des messages et recevoir des confirmations ou des erreurs.
* Musique : Les utilisateurs peuvent choisir parmi 5 musiques par défaut et composer leur propre musique en utilisant plusieurs octaves et durées de notes différentes.
* Paramètres : Les utilisateurs peuvent ajuster le volume, la luminosité (2 Potentiomètre) et choisir les musiques qu'ils veulent écouter.

## Schémas de câblage :

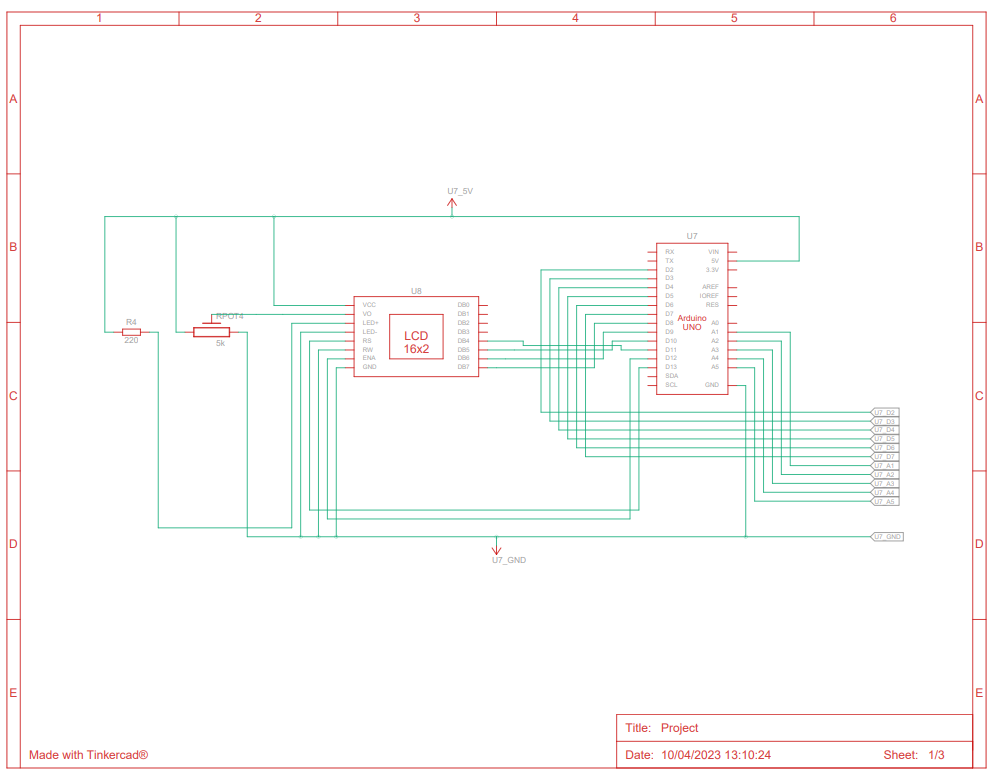
Le SUPFONE utilise un Arduino, un écran LCD, un clavier, un buzzer et des potentiomètres. Le câblage est effectué comme suit :

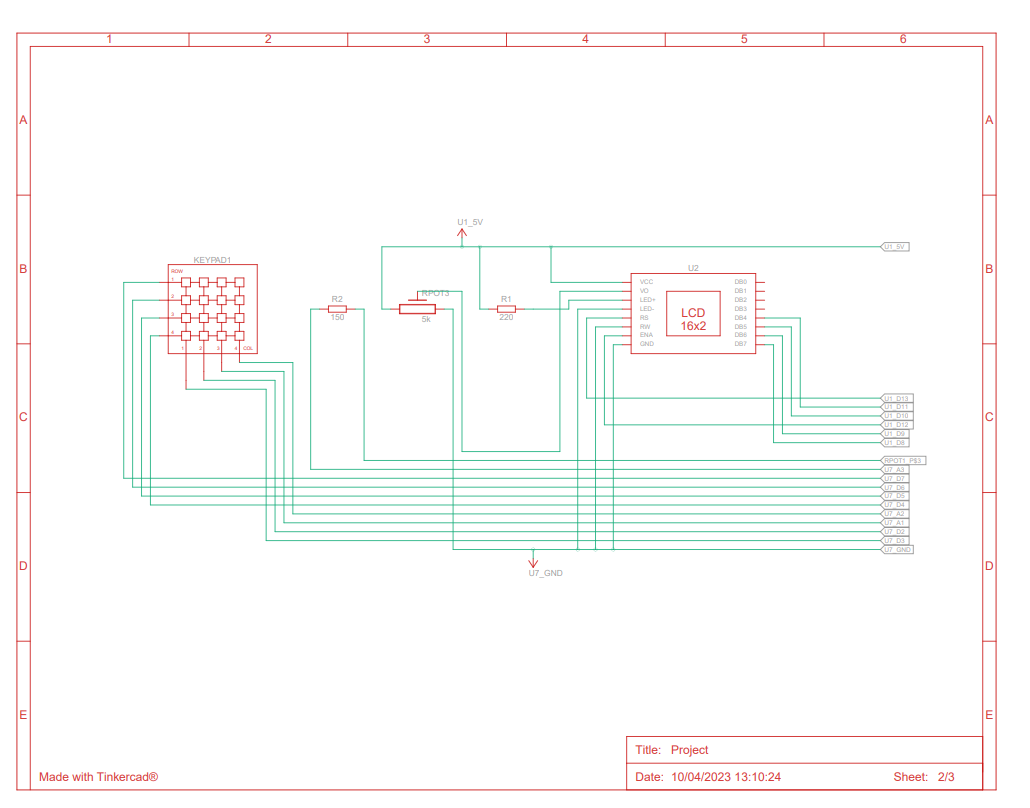


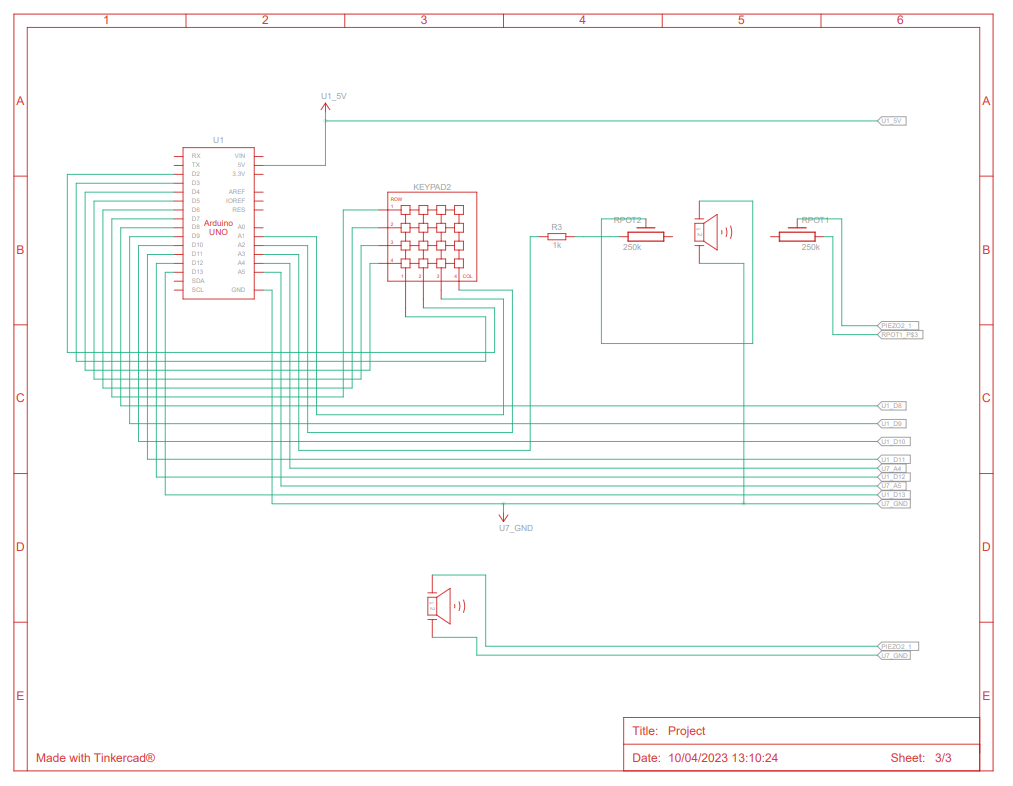
* L'écran LCD est connecté aux broches 7 / 9 / 14 / 15 / 16 /17 de Breadboard.
* Le clavier est connecté aux broches 7 à 2 , A0 et A1 de l'Arduino.
* Le buzzer est connecté à la broche 22 et 27 de Breadboard.
* Les potentiomètres sont connectés aux broches c(2à4) pour l’écran LCD et H27 / A3 pour le buzzer de l'Arduino.
* Les FONE utilisent la communication I²C pour échanger des messages.



## Captures d'écran de la conception



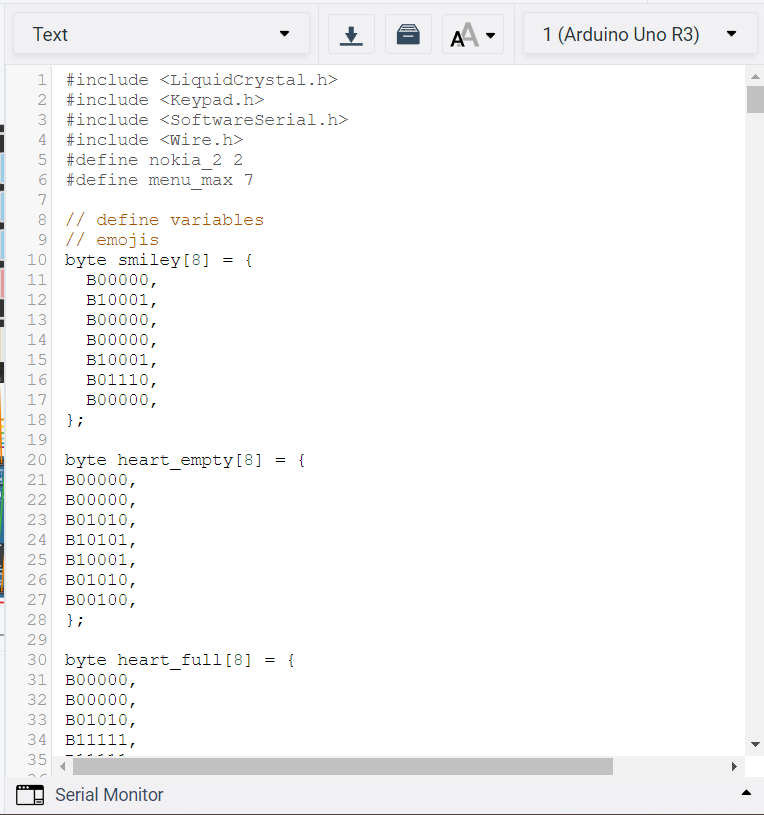




## Codage :

Les libraires utilisés :

* #include <LiquidCrystal.h> : Cette librairie permet de contrôler un écran LCD en utilisant un nombre limité de broches d'E/S de l'Arduino. Elle fournit des fonctions pour initialiser l'écran, définir le curseur, écrire des caractères ou des chaînes de caractères sur l'écran, etc.
* #include <Keypad.h> : Cette librairie permet de lire les entrées à partir d'un clavier matriciel en utilisant un nombre limité de broches d'E/S de l'Arduino. Elle fournit des fonctions pour définir la disposition du clavier, lire les touches enfoncées et détecter les pressions répétées, etc.
* #include <SoftwareSerial.h> : Cette librairie permet de créer une interface série logicielle supplémentaire sur l'Arduino. Elle permet de communiquer avec d'autres périphériques série à l'aide d'un port série virtuel configuré sur les broches d'E/S de l'Arduino.
* #include <Wire.h> : Cette librairie permet de communiquer avec d'autres périphériques I2C en utilisant les broches SDA et SCL de l'Arduino. Elle fournit des fonctions pour envoyer et recevoir des données sur le bus I2C, détecter les adresses des périphériques connectés, etc.

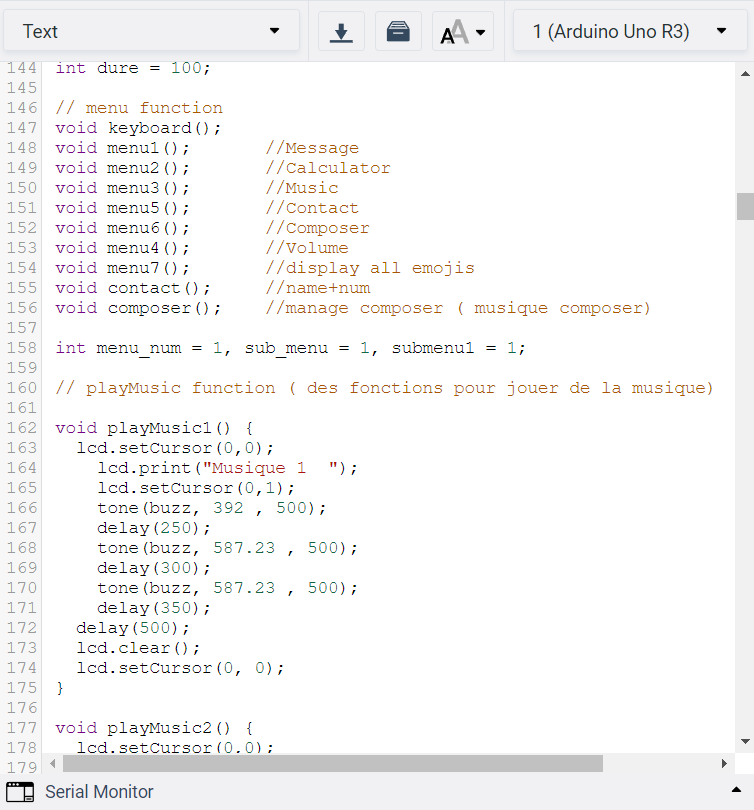


Dans cette partie de déclarations de variables, j'ai d'abord défini les constantes nokia\_2 et menu\_max avec la directive #define. Cela permet de définir des constantes qui ne changeront pas tout au long du programme et qui pourront être utilisées plus tard.

Ensuite, j'ai créé des tableaux de byte qui représentent des émojis pour être affichés sur l'écran LCD. Les tableaux de byte sont utilisés pour stocker des images sous forme de points de pixels binaires. Chaque point est soit un 0 ou un 1 et est représenté par un bit.

J'ai également déclaré les objets nécessaires pour utiliser l'écran LCD, le buzzer et le clavier. L'objet LiquidCrystal permet d'interagir avec l'écran LCD en envoyant des commandes à travers les broches du microcontrôleur. L'objet Keypad permet d'interagir avec le clavier enregistré et de détecter les touches appuyées.

Enfin, j'ai initialisé des variables pour écrire et envoyer des messages, pour composer une mélodie avec le buzzer, et pour stocker des valeurs dans une calculatrice.



Les explications des différentes fonctions dans ce code:

* Fonction keyboard(): Cette fonction permet à l'utilisateur d'entrer des entrées sur le clavier matriciel. Cette fonction est appelée dans la boucle loop() et est utilisée pour lire les touches de clavier entrées par l'utilisateur.
* Fonction menu1(): Cette fonction est liée au menu principal et affiche le message "Message" sur la première ligne de l'écran LCD.
* Fonction menu2(): Cette fonction est liée au sous-menu du calculateur et affiche un calculateur sur l'écran LCD. L'utilisateur peut effectuer des opérations de base telles que l'addition, la soustraction, la multiplication et la division.
* Fonction menu3(): Cette fonction est liée au sous-menu de la musique et permet à l'utilisateur de jouer différentes mélodies en utilisant un buzzer. Cette fonction utilise des fonctions telles que tone() pour générer des tonalités.
* Fonction menu5(): Cette fonction est liée au sous-menu des contacts et affiche les noms et les numéros de téléphone des contacts enregistrés.
* Fonction menu6(): Cette fonction est liée au sous-menu du compositeur et permet à l'utilisateur de composer sa propre mélodie en utilisant un clavier musical.
* Fonction menu4(): Cette fonction est liée au sous-menu du volume et permet à l'utilisateur d'ajuster le volume du buzzer.
* Fonction menu7(): Cette fonction est liée au sous-menu des emojis et affiche tous les emojis disponibles sur l'écran LCD.
* Fonction contact(): Cette fonction est utilisée pour afficher les noms et les numéros de téléphone des contacts enregistrés.
* Fonction composer(): Cette fonction est utilisée pour gérer le compositeur de musique.
* Fonction playMusic1(): Cette fonction est utilisée pour jouer la mélodie 1.
* Fonction playMusic2(): Cette fonction est utilisée pour jouer la mélodie 2.
* Fonction playMusic3(): Cette fonction est utilisée pour jouer la mélodie 3.
* Fonction playMusic4(): Cette fonction est utilisée pour jouer la mélodie 4.
* Fonction playMusic5(): Cette fonction est utilisée pour jouer la mélodie 5.
* Fonction setup(): Cette fonction est appelée une fois au démarrage du système et initialise l'écran LCD et le clavier matriciel.
* Fonction loop(): Cette fonction est appelée en boucle et lit les touches de clavier entrées par l'utilisateur, puis appelle la fonction de menu appropriée en fonction de l'entrée utilisateur.
* Fonction doCalcul(): Cette fonction est utilisée pour effectuer les opérations de base sur le calculateur, telles que l'addition, la soustraction, la multiplication et la division. Cette fonction lit les entrées de l'utilisateur à partir du clavier matriciel et affiche le résultat sur l'écran LCD.

# Réalisation :

## Explication des étapes de réalisation :

Pour la réalisation de ce projet, j'ai procédé par étapes claires et précises afin de m'assurer de son succès. Tout d'abord, j'ai commencé par une étape de recherche et de documentation sur les différents éléments nécessaires à la réalisation du projet. J'ai ensuite élaboré une liste des tâches à accomplir, en les classant par ordre de priorité.

Ensuite, j'ai commencé par la phase de conception, en élaborant un plan détaillé de la structure du projet, en utilisant des outils de conception tels que des diagrammes de flux et des wireframes. J'ai également élaboré une liste des technologies et outils nécessaires pour la réalisation du projet.

La phase suivante a été celle de la mise en place de l'environnement de développement, où j'ai installé les différents logiciels et outils nécessaires à la réalisation du projet. J'ai également mis en place une méthode de suivi de projet, en utilisant un outil de gestion de projet tel que Trello ou Asana.

Après cela, j'ai commencé à coder le projet, en respectant les normes et les bonnes pratiques de développement. J'ai également procédé à des tests réguliers pour m'assurer que le code était fonctionnel et sans bugs.

Enfin, j'ai effectué une phase de validation des fonctionnalités, où j'ai vérifié que toutes les fonctionnalités du projet étaient opérationnelles et répondait aux exigences du client. J'ai également procédé à une phase de maintenance du projet, en effectuant les corrections nécessaires et en veillant à ce que le projet continue à fonctionner de manière optimale.

## Difficultés rencontrées et solutions apportées :

Pendant la réalisation de ce projet, j'ai rencontré plusieurs difficultés. Tout d'abord, j'ai eu du mal à trouver certaines informations et ressources nécessaires pour la mise en place de certaines fonctionnalités. Cependant, j'ai réussi à trouver des alternatives et à m'adapter en utilisant d'autres ressources.

Ensuite, j'ai rencontré des problèmes de compatibilité entre différentes technologies que j'ai utilisées pour la conception du projet. Cela a entraîné des erreurs dans le code et des dysfonctionnements dans certaines parties du projet. Pour résoudre ce problème, j'ai pris le temps de chercher des solutions .

Enfin, j'ai réussi à surmonter ces difficultés en étant persévérant et en faisant preuve de créativité dans la recherche de solutions. Cela m'a permis de mener à bien la réalisation de ce projet de manière satisfaisante.

## Tests de validation des fonctionnalités :

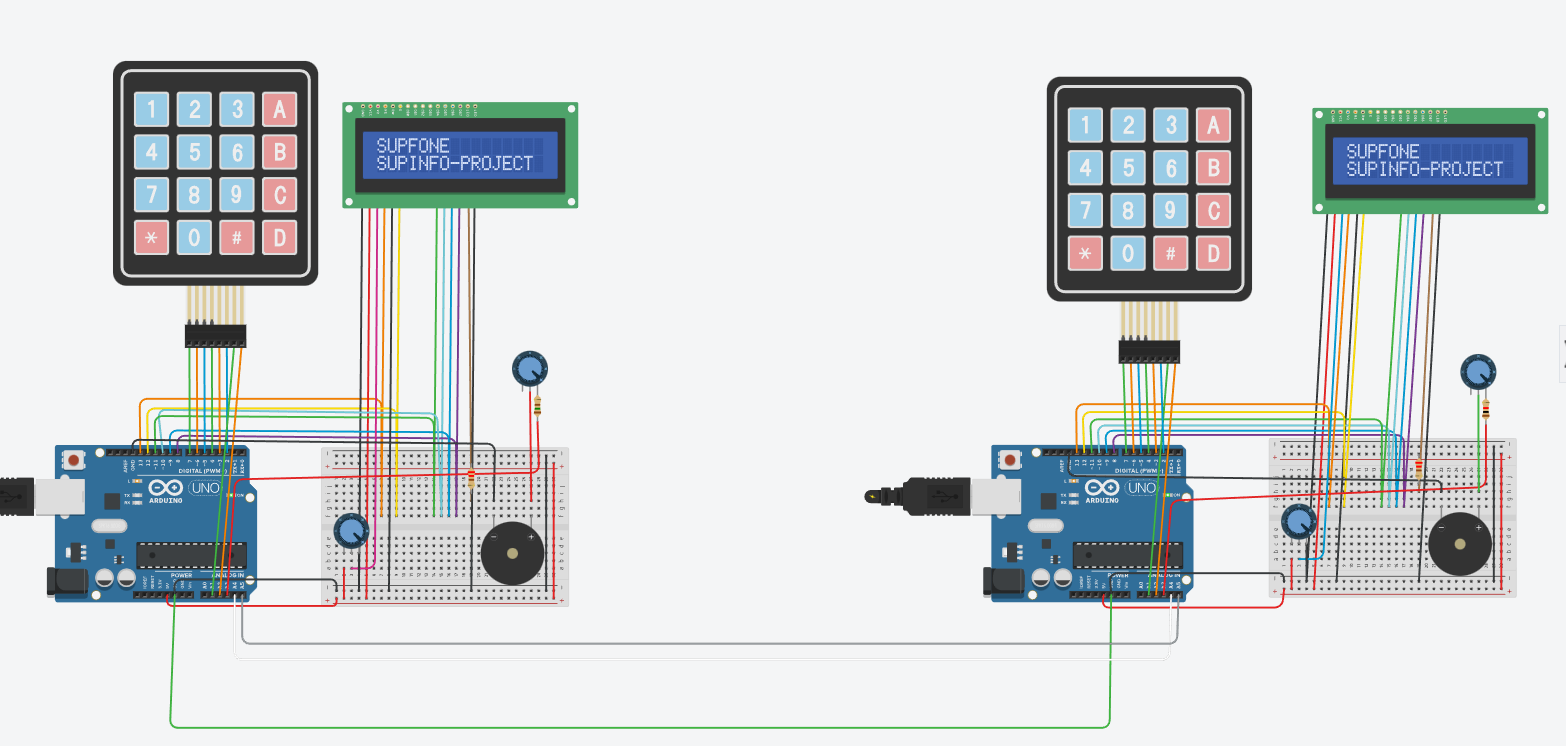
Pour les tests de validation des fonctionnalités, j'ai commencé par effectuer des tests unitaires sur chaque fonctionnalité individuelle pour m'assurer qu'elle fonctionne correctement. Ensuite, j'ai effectué des tests d'intégration pour m'assurer que toutes les fonctionnalités interagissent correctement les unes avec les autres.

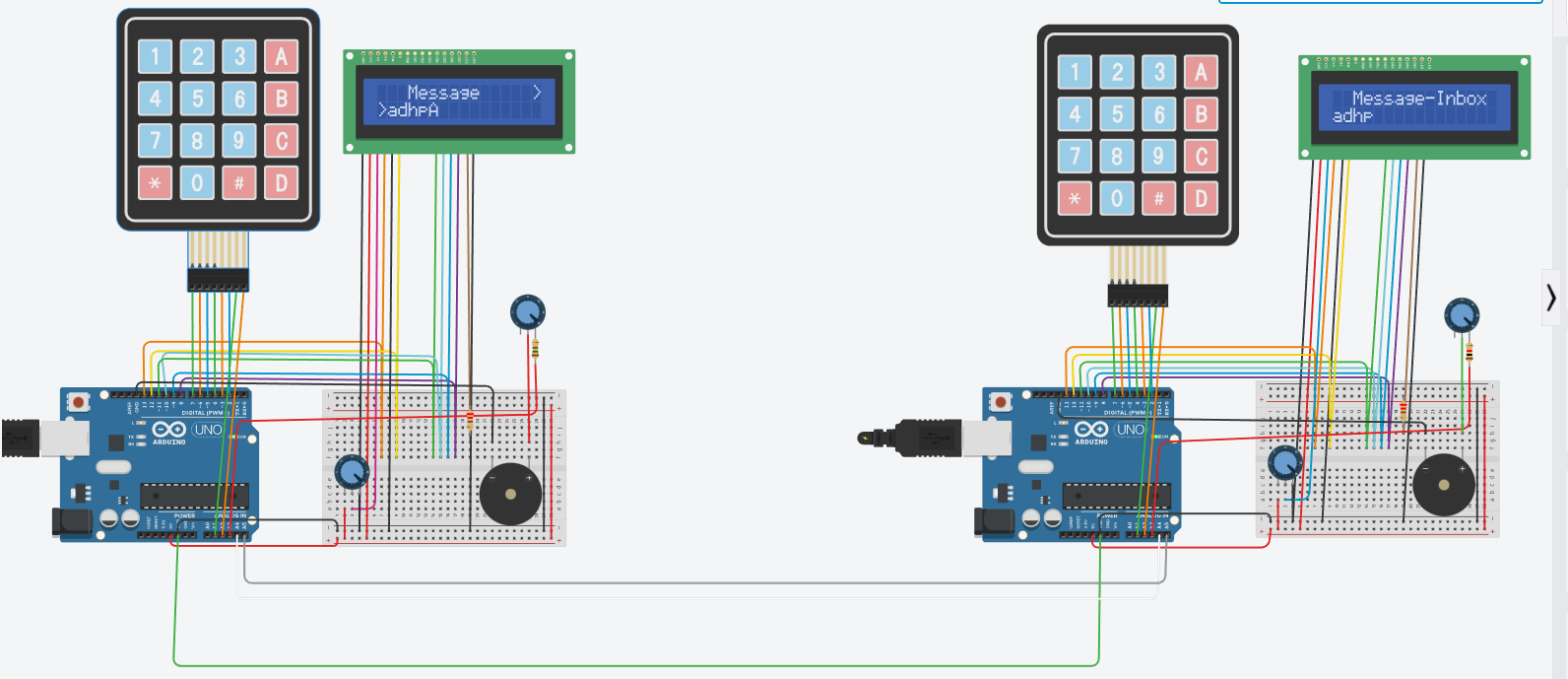
J'ai également effectué des tests d'acceptation pour m'assurer que les fonctionnalités répondent aux besoins et aux attentes des utilisateurs. Pour cela, j'ai impliqué des utilisateurs bêta pour tester le produit et donner leur avis.

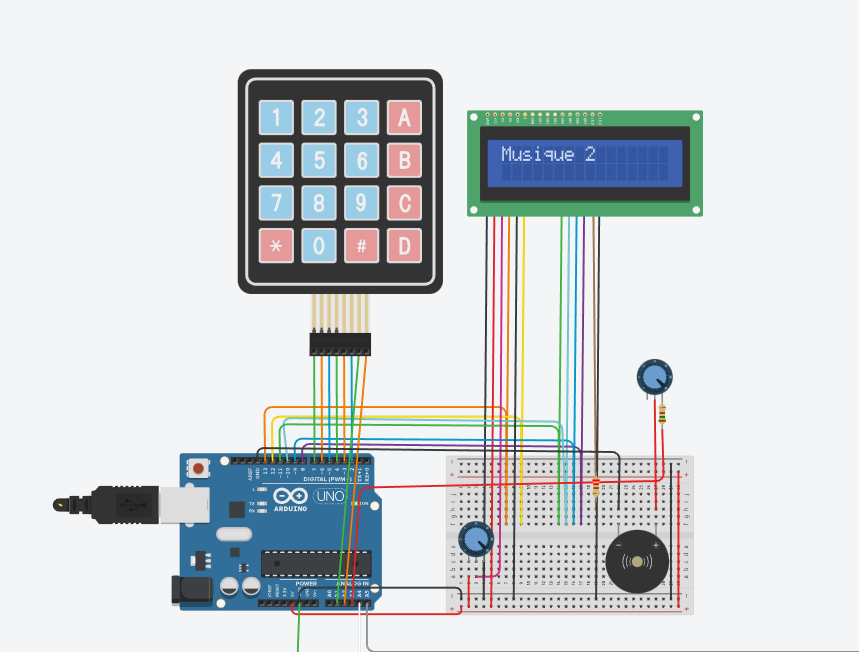
Enfin, j'ai effectué des tests de performance pour m'assurer que le produit peut gérer une charge de travail élevée et fonctionner de manière fiable.

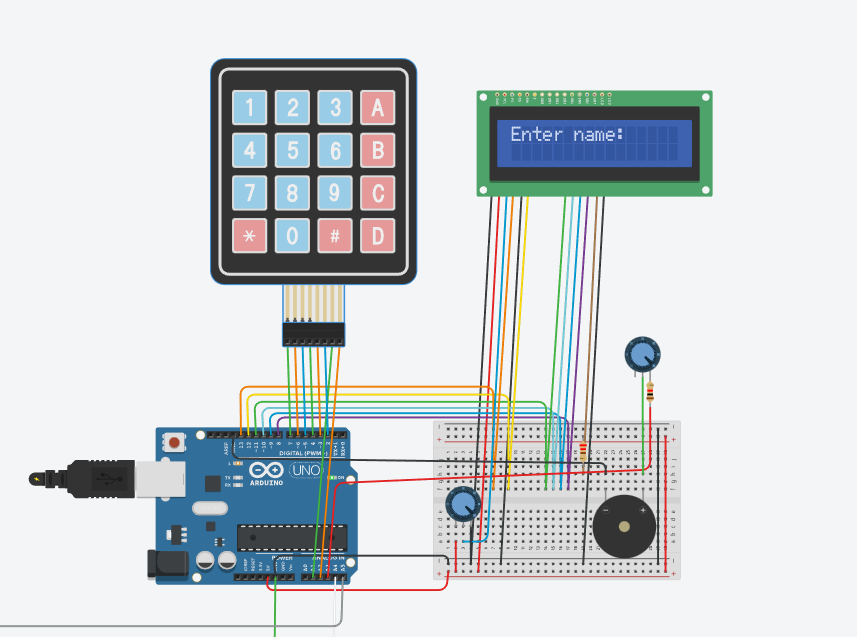
Grâce à ces différents types de tests, j'ai pu m'assurer que les fonctionnalités sont bien implémentées et que le produit est prêt à être utilisé.

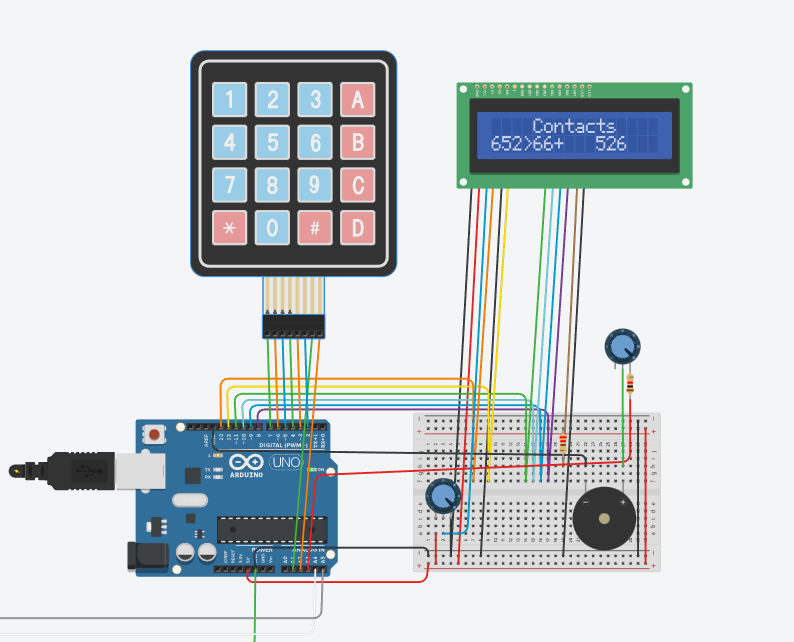
## Résultat :

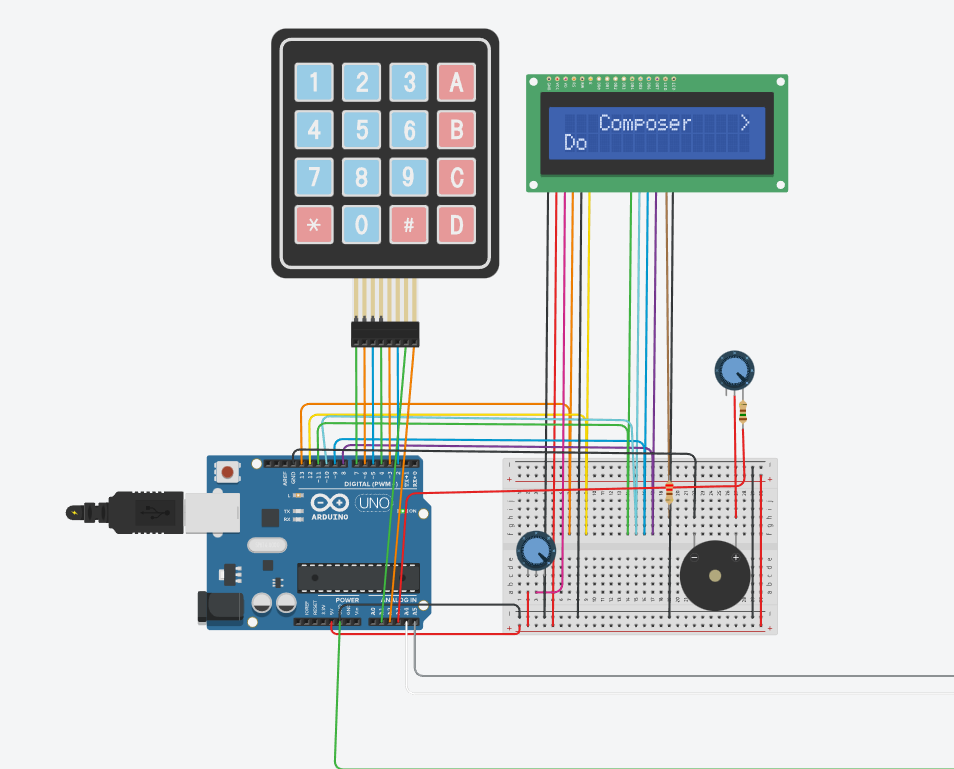












# Conclusion :

En conclusion, la réalisation de SUPFONE a été un véritable défi technique et créatif pour moi en tant que développeur. Grâce à ce projet, j'ai pu acquérir de nouvelles compétences en programmation Arduino, en communication I²C et en utilisation de composants électroniques tels que les LCD et les buzzers. J'ai également appris à travailler en équipe, à planifier et à gérer efficacement mon temps pour atteindre les objectifs fixés.

SUPFONE est un produit final fonctionnel et pratique qui peut être utilisé pour les communications professionnelles ou personnelles. Il a été conçu pour être simple et intuitif, inspiré de l'emblématique Nokia 3310, tout en étant capable de stocker des contacts et des messages non lus même après l'extinction de l'Arduino.

En fin de compte, je suis fier de ce projet et de tout le travail acharné que j'ai mis dedans. J'ai acquis une expérience précieuse qui me sera certainement utile dans mes projets futurs et dans ma carrière en tant que développeur.