

PROJET SYSTÈME EMBARQUÉ



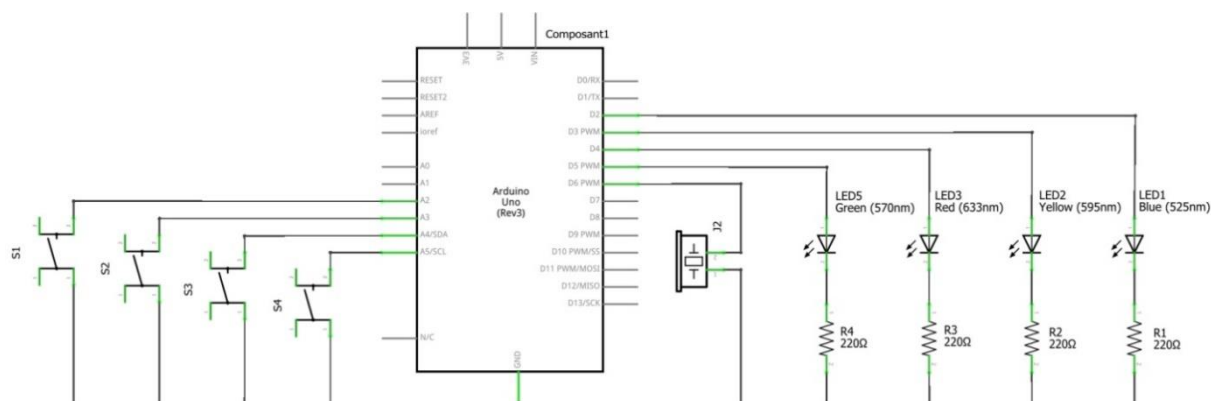
Arthur Bottemanne

CPNV

Table des matières :

1	Explication du schéma électrique	- 2 -
	Composant utiliser	- 2 -
	Explication	- 2 -
	Application	- 3 -
	Schéma d'implantation	- 3 -
2	Stratégie utilisée	- 4 -
	Parcours du travail	- 4 -
	Méthode du travail	- 4 -
3	Conclusions	- 5 -
	Points positifs	- 5 -
	Points négatifs	- 5 -
	Difficultés	- 5 -
	Suites possibles	- 5 -
4	Sources	- 6 -

1 Explication du schéma électrique



Composant utiliser

- Un Arduino Uno (Rev3)
- Un veroboard 20x20
- Une LED Bleue(524nm)
- Une LED Jaune(595nm)
- Une LED Rouge(633nm)
- Une LED Verte(570nm)
- Quatre Résistances 220 Ω
- Quatre Boutons
- Un Buzzer
- Deux barrettes mâles de connexion 8 broches
- Une barrette femelle de connexion 4 broches

Explication

Les boutons sont reliés aux pins analogues A2 (S1), A3 (S2), A4/SDA (S3) et 5/SCL (S4) sur l'Arduino Uno (Rev3), les LEDs au pins D2 (LED1), D3 PWM (LED2), D4 (LED3) et D5 PWM (LED5) et sont chacune suivie d'une résistance 220 Ω (R1, R2, R3 et R4 respectivement) et finalement le buzzer (J2) est connecté au pin D6 PWM.

Tous les composants sont finalement reliés à la terre de l'Arduino (GND).

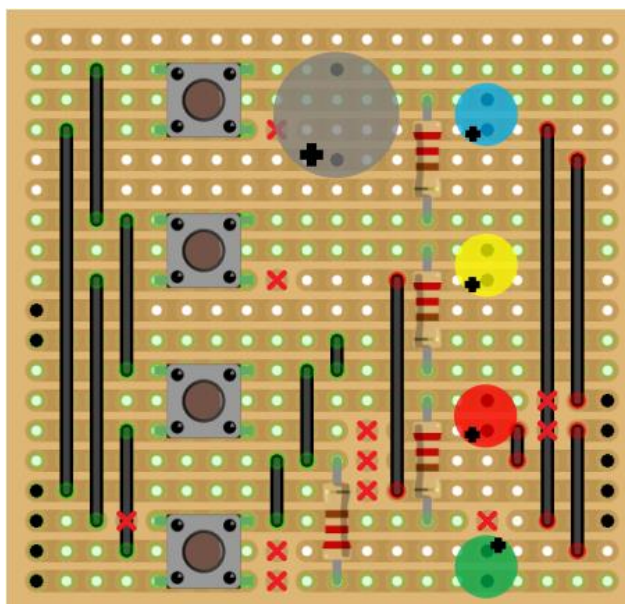
Application

Ce schéma a été utilisé pour créer un Shield compatible avec l'Arduino utilisé. Pour pouvoir connecter le Shield à l'Arduino, les barrettes ont été coupées pour faire 3 barrettes en tout, une à deux broches, à quatre broches et à cinq broches.

Les barrettes en question ont été soudées pour connecter le veroboard aux pins indiqués précédemment.

À la place de souder le buzzer directement au veroboard comparé aux autres composants, la barrette femelle a été installée pour pouvoir placer le buzzer et l'enlever sans encombre.

Schéma d'implantation



2 Stratégie utilisée

Parcours du travail

Dans un premier temps, un diagramme de flux basique pour les fonctionnalités obligatoires a été créé.

Avant d'avoir commencer le développement de mon programme, J'ai cherché des moyens pour comment changer d'IDE afin de ne pas utiliser Arduino mais Visual Studio Code.

Après avoir fait le changement sur VS code, j'ai fait une première version très basique du programme pour confirmer que l'Arduino, le Shield et que VS code fonctionne correctement qui m'a permis de créer un dépôt GitHub de mon projet.

Avant de continuer mon programme, j'ai recherché des manières de faire de la musique très précise en Arduino ou j'ai trouvé un site web qui converti les fichiers midi en code Arduino qui s'appelle [extramaster](#).

J'ai créé les fichiers midi à partir d'un programme nommé Waveform qui est un DAW gratuit pour faire de la musique.

La conversion des fichiers midi ayant été terminer, j'ai finalement fait une version basique du programme en rapport avec le diagramme de flux crée précédemment.

Avec le programme basique terminer, j'ai modifié le diagramme de flux pour implémenter les améliorations possibles qui ont été ajouté dans le cahier des charges.

Pour finir, j'ai fait comme pour l'a première version du programme ou je me suis appuyé sur le diagramme de flux pour programmer les améliorations finales.

Méthode du travail

En programmant, chaque fonctionnalité ajoutée au programme a été mis dans une fonction pour qu'il n'y ait aucune fonction faisant deux fonctionnalités différentes.

Les fonctions utilisées pour les jingles ont été mis dans un fichier appart pour leur taille et inclus dans le programme principal

Les commentaires ont été ajouté juste après avoir finit une fonction ou bien un bout de code.

Les jingles qui ont été généré par [extramaster](#) et ont été créditer correctement ainsi que les musique utilisé pour les jingles.

3 Conclusions

Points positifs

Grâce a ce projet, j'ai pu mieux découvrir ma méthode de travail en travaillant personnellement.

J'ai aussi eu l'occasion d'apprendre comment utiliser un DAW (Digital Audio Workstation) pour créer les fichiers midi à convertir en code Arduino.

Points négatifs

Le fait de travailler sur un jeu de société simple ne m'a pas vraiment inspiré. Je préfère développer des programmes qui offre plus de complexité.

Difficultés

J'ai eu des problèmes de me retrouver dans mon code par le simple fait que les fonctions était mises de manière aléatoire et sans aucune structure concrète.

Les premières utilisations du DAW Wavefrom ont été très intimidantes par le nombre gigantesque de fonctionnalité à disposition et par ce fait été compliqué de commencer de faire des fichiers midi.

Suites possibles

Faire un software en python communiquant avec l'Arduino affin d'offrir plus de possibilité come faire des meilleurs son et musique.

Implémenter d'autres mode de jeux pour les deux autres boutons ce qui rajouterai de la complexité au programme.

4 Sources

Extramaster: <https://extramaster.net/tools/midiToArduino/>

Wavefrom: <https://www.tracktion.com/products/waveform-free>

Dépôt GitHub: <https://github.com/WalvieCPNV/arduino>