

# Electronics Project

## TP Tech 1 : SMD et RGB

[contact@42chips.fr](mailto:contact@42chips.fr)

*Résumé: Musique électronique et vomit de licorne*

# Table des matières

# Chapitre I

## Préambule

Dans un saladier, mélanger la farine de maïs, le sel, le paprika, le curry et le poivre. Choisir la quantité d'épices selon vos goûts.

Versez l'huile au centre puis mélanger un peu. Verser ensuite l'eau tiède puis mélanger la pâte avec les mains.

Former une boule puis laisser la pâte reposer environ 30 min à température ambiante. A la fin du temps de repos, préchauffer votre four à 170°.

Prendre un tapis silicone ou une feuille de papier cuisson de la taille de votre plaque de four. Fariner le dessus. Couper la pâte en 2 parties.

Étaler finement la première partie sur le tapis ou la feuille, saupoudrer de paprika puis repasser le rouleau sur la pâte.

A l'aide d'une roulette à pizza, former un rectangle puis faire des traits verticaux et horizontaux pour former des carrés.

Passer ensuite la roulette en biais pour obtenir des triangles.

Ne pas séparer la pâte et faire cuire tel quel, les tortillas vont se séparer facilement. Faire la même chose avec le restant de pâte.

Enfourner la première plaque pendant 15 et 20 min. Faire attention à la coloration de la pâte, cela peut aller vite à la fin du temps de cuisson.

Laisser refroidir à température ambiante, faire cuire la seconde fournée de tortillas et détacher les triangles de la 1ère fournée.

Les tortillas doivent être fines et croustillantes !

# Chapitre II

## Consignes générales

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les TPs

- Le langage utilisé pour ce projet est le C.
- Il n'est pas nécessaire de coder à la norme de 42.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne prendrons en compte ni n'évaluerons un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Vos exercices seront évalués par des responsables de l'association 42Chips.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre répertoire lors de la peer-évaluation.
- Vous avez une question ? Demandez à votre voisin de droite ou de gauche. Vous pouvez demander sur le salon dédié dans le discord [42Chips](#) ou en dernier recours à un responsable 42Chips.
- Toutes les réponses à vos questions techniques se trouvent dans les **datasheets** ou sur Internet. A vous d'utiliser et d'abuser de ces sujets pour comprendre comment réaliser votre exercice.
- Vous devez utiliser la datasheet du microcontrôleur qui vous est fourni et commenter les parties importantes de votre programme en renseignant où vous avez trouvé les indices dans le document, et, si nécessaire, expliquer votre démarche. Ne faites pas des pavés non plus. Il faut que cela reste clair.
- Écoutez attentivement les encadrants lors des séances de TP, ils vous donneront des éléments essentiels sur le fonctionnement du microcontrôleur.

# Chapitre III

## Consignes pour les TPs Tech


Spécifiquement pour les TPs Tech vous devrez gérer votre temps entre la programmation embarquée et la soudure.

Comme les postes sont limités en nombre, il vous faudra partager la semaine avec vos camarades pour tous pouvoir réaliser la partie technique.

Nous vous encourageons à vous concerter pour ne pas avoir de soucis de dernière minute.

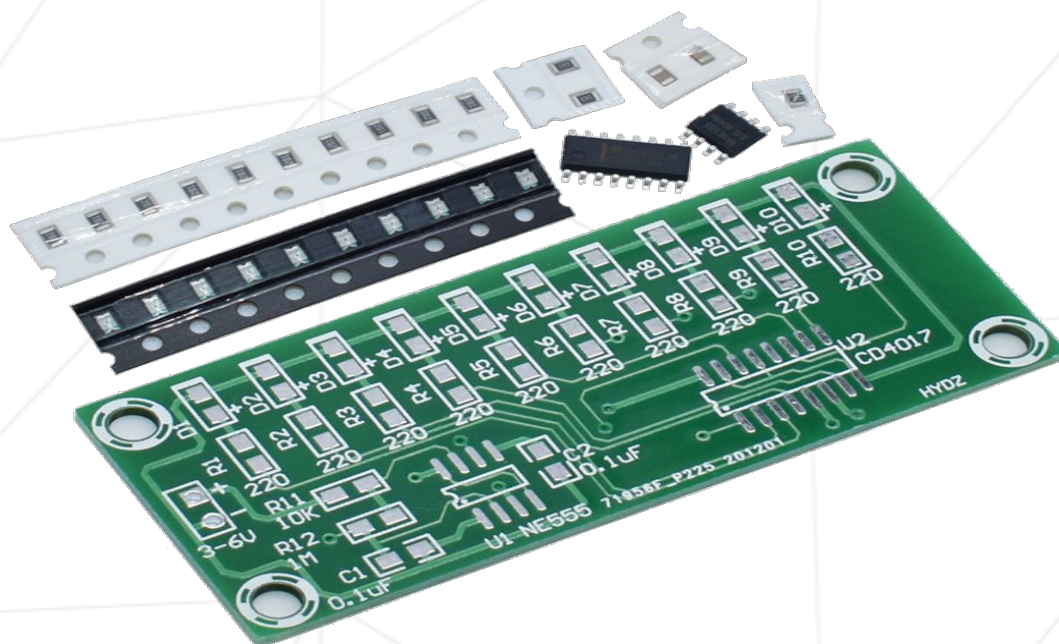
# Chapitre IV

## Soudure : Monté en surface

	Exercice : 00
Soudure	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre :	
Fonctions interdites : <b>Aucune</b>	


Maintenant que vous avez soudé en traversant, vous allez vous frotter à plus difficile et souder un kit de composants monté en surface.

C'est un exercice qui vous permet de vous familiariser avec de la soudure plus technique pour avoir moins de soucis lors de vos prochaines soudures.

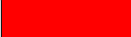








# Chapitre V

## Programation : PWM

	Exercice : 01
Rouge Vert Bleu d'Auvergne	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>rgb.c</b>	
Fonctions interdites : <b>Aucune</b>	

Écrivez un programme qui, à chaque fois que vous appuyez sur le bouton, Change la couleur de la LED RGB (PD6, PD5, PD3) avec la liste :

nom	R#	G#	B#	couleur
rouge	ff	0	0	
vert	0	ff	0	
bleu	0	0	ff	
yellow	ff	ff	0	
cyan	0	ff	ff	
magenta	ff	0	ff	
white	ff	ff	ff	






## Exercice : 02


Vomit de licorne

Dossier de rendu : *ex02/*Fichiers à rendre : **Makefile**, **rgb.c**Fonctions interdites : **Aucune**

Écrivez un programme qui, à chaque fois que vous appuyez sur le bouton, Change la couleur de la LED RGB (PD6, PD5, PD3) avec la liste :

nom	R#	G#	B#	couleur
fushia	f2	18	4f	
turquoise	19	e0	a1	
indigo	4b	00	82	



	Exercice : 03
dump	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <b>Makefile</b> , <b>dump.c</b>	
Fonctions interdites : <b>Aucune</b>	

Vous devez réaliser un programme qui lit l'entrée UART avec le format de code RGB : **#XXXXXX** où X est un digit hexadecimal et chaque composante va de **#00** à **#ff**.

Et qui doit maintenant changer la couleur de la LED RGB en fonction de la valeur du code.



Il est normal que les couleurs ne s'affiche pas exactement comme vous le voulez. Il est possible de corriger le problème avec un **filtre logiciel**. Mais il ne vous est pas demandé le corriger.