

Chaque chose en son temps

Penser, souder, coder



Introduction

“Chaque chose en son temps” est un projet d'acculturation au mouvement Maker dans lequel les participants vont se questionner, identifier un besoin dans leur quotidien et y répondre par la fabrication d'un objet DIY programmable. Il s'agit dans le cas présent d'un timer. Cet objet comprend un microprocesseur Arduino Nano, un buzzer pour un retour sonore, une LED RGB pour un retour lumineux et un bouton pour pouvoir interagir avec le dispositif. A noter que ce circuit peut aussi être utilisé comme plateforme d'apprentissage simple pour apprendre à programmer un bouton, une LED RGB et un buzzer.

Introduction	1
Fiche atelier	3
Déroulé de l'atelier	6
Préparation	6
Pré-requis	6
Installation de l'environnement mBlock 5	6
Préparation des composants et des outils nécessaires	8
Susciter un questionnement (10 minutes)	9
Réaliser le circuit électronique (50 minutes)	10
Notions clefs	10
Fonctionnement du circuit	12
Construire et Souder le circuit	15
Coder l'Arduino Nano pour faire fonctionner le circuit (1 heure)	24
Notions clefs	25
Proposition de déroulé pour débutant	26
Variantes et approfondissement (Dépend du niveaux du groupe)	26
Reconstruire le code	26
Utiliser des fonctions avancées et manipuler les variables	26

Fiche atelier

Titre de l'atelier : Chaque chose en son temps

Pour qui : Pour des participants de niveaux débutants à intermédiaire

Pour combien : 6

Combien de temps : 2h00 ou 2 x 1h00 (la durée peut varier selon les besoins)

Profil des intervenants : Une expérience avec arduino est préférable, une familiarisation avec cet objet et avec l'environnement de développement est nécessaire en amont (1 à 2h la première fois).

Objectifs pédagogiques :

- Identifier un besoin dans son quotidien et y répondre par un objet de type DIY
- Connaître la notion de senseur et d'effecteur
- Connaître quelques composants électroniques de base
- Se familiariser avec mblock 5 (la programmation Scratch)
- Comprendre, modifier et téléverser un programme Arduino relativement simple

Production : Un timer chronométrant une durée choisi par son utilisateur pour répondre à un besoin (ie : se brosser les dents, faire un oeuf dur, etc) capable de bipper (indicateur sonore) et changer de couleur (indicateur lumineux).

Materiel :

Liste du matériel en section 0.c : [Lien](#)

En annexe veuillez-trouver 3 suggestions de code correspondant à trois niveaux différents.

Veuillez également trouver le plan du circuit électronique au format jpg et le fichier Fritzing pour pouvoir l'éditer.

Histoire de ce projet : Objet initialement conçu pour faire du sport. Bien que beaucoup d'application mobile et d'objets permettent de chronométrer des durées précises, il ne s'est pas proposé de solution simple pour réaliser un enchaînement de 7 fois 40 secondes avec 1 minute 20 de pause entre chaque série de 7 et ce, 4 fois de suite. Cet objet, avec la programmation adéquate permet de répondre à cette problématique simplement, en se passant d'interagir avec son téléphone.

Plan de séquence et durée prévue :

Nom de l'atelier	Chaque chose en son temps
Temps de l'atelier	2h00 ou 2x1h00
Nombre de Personnes	6
Module 1 : Construire le circuit	
0. Préparation	
Avoir Mblock Installé sur tous les ordinateurs	Déjà sur tous les ordinateurs
Installer les ordinateurs, vérifier les batteries (brancher le cas échéant), les démarrer et ouvrir MBlock	20''
Sortir les câbles pour le téléversement	5'
Sortir les fers à souder, l'étain, les supports de fer à couder	
Selon le niveau et l'âge des participants, pré-souder quelques composants	15'-30'
1. Le questionnement	
Interroger les participants sur un besoin qu'ils auraient en terme de mesure du temps.	5'-10'
Chaque participant définit ce pourquoi le timer va être utile et le note quelque part	
2. Le Circuit	
Presenter arduino	5'-10'
Expliquer les notions de In and Out	
Présenter les composants senseur et effecteur : Boutton, Buzzer, LED	

Expliquer leurs rôles respectifs (succinctement)	
Expliquer le circuit	
Apprendre à souder	5'
Souder le circuit	40'
Verifier le circuit	5'
Module 2 : Le code	Cette partie est extrêmement modulaire. Le code étant déjà écrit, selon le temps restant, il est possible de juste changer la variable temps, ou bien d'explorer le code.
3. Le Code	
Presenter l'IDE choisi	
Présenter la structure du code Arduino "Loop and For"	10'
Expliquer le Code et les différentes fonction	
Explorer la fonction "attendre"	5'
Explorer la fonction "set digital output as"	5'
Explorer la fonction "Play Pin"	5'
Construire un code customisé	15'

Déroulé de l'atelier

0. Préparation

a. Pré-requis

Pour animer cet atelier pas besoin d'être un programmeur chevronné, en revanche il vous faudra être familier avec certaines notions de bases de la programmation, connaître l'environnement de développement mBlock 5, connaître quelques composants électroniques de base et savoir souder.

Voici quelques vidéos ressources pour combler ces lacunes si elles existaient :

Comment souder : <https://www.youtube.com/watch?v=8oGjG9uyYq8>

Prise en main de mBlock5 : <https://www.youtube.com/watch?v=9dB1ad63Y4M>

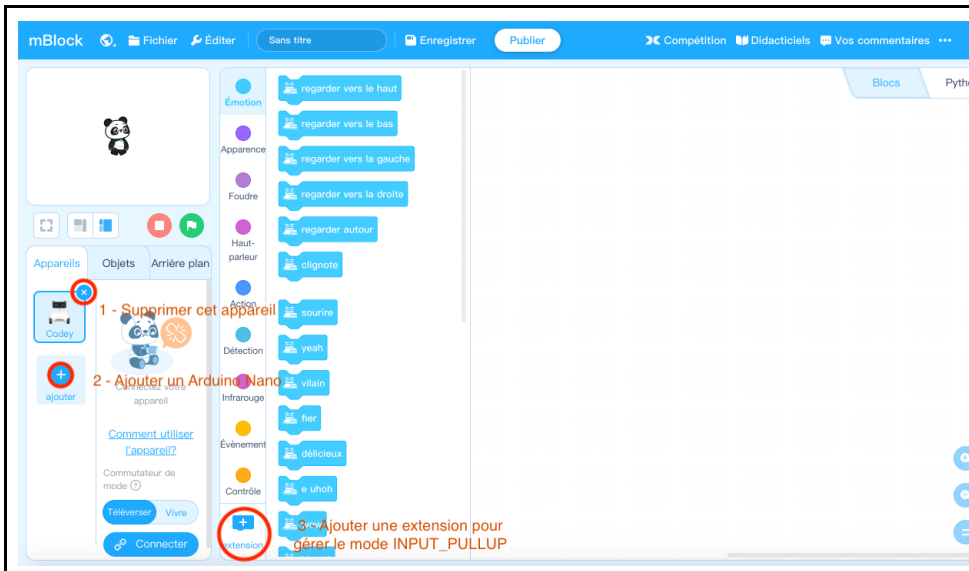
Prise en main d'un Arduino Uno : <https://www.youtube.com/watch?v=nL34zDTPkcs>

Pour aller un peu plus loin : https://www.youtube.com/watch?v=nbD_V4QtNvY

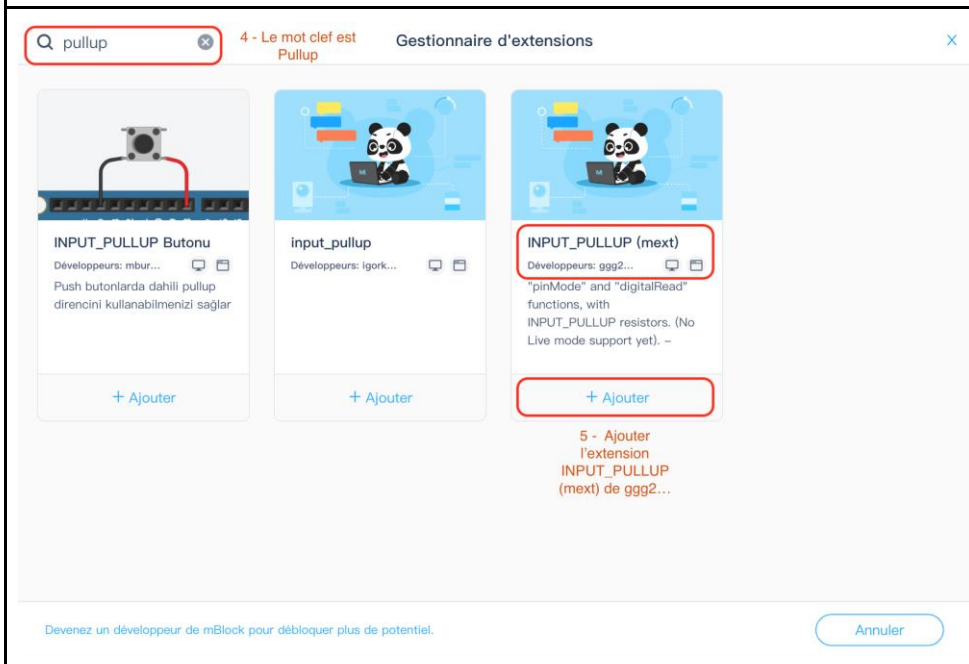
b. Installation de l'environnement mBlock 5

La première chose à faire est d'installer le logiciel mBlock 5 sur vos ordinateurs : [Download mBlock - Teach and learn coding | Designed for STEAM](#) (déjà installé sur les ordinateurs FabLab en Kit)

La seconde est d'ouvrir le fichier .mblock présent au côté de ce tutoriel. Normalement tout devrait déjà être en place. Si vous rencontrez des difficultés il vous faudra vérifier :



1) Que le projet est paramétré pour une carte Arduino Nano



2) Que l'extension INPUT_PULLUP(mext) est bien installée.

c. Préparation des composants et des outils nécessaires

Voici la liste du matériel nécessaire pour un participant :

Piece	QT	Qu'est ce que c'est
Arduino Nano + Broches mâles sortantes (2x15)	1	Carte de développement programmable en Arduino
proto board 4x6	1	Carte sur laquelle on construit le circuit électronique
Rangées de 15 Broches Femelle	2	Broche femelle que l'on va utiliser pour venir brancher l'Arduino Nano sur le circuit
LED RGB cathode commune	1	Une LED emettant du rouge, du vert et du bleu
Bouton 6x6x7	1	Un bouton poussoir pour interagir avec le circuit
Resistances 220	3	Résistances de 220 ohms pour protéger la LED
Resistance 110	1	Résistance de 110 ohms pour protéger le buzzer
5v Active Buzzer	1	Un buzzer actif, un petit speaker capable de jouer une note à la fois

Mettre ces composants dans un petit sachet ou dans une petite boîte pour pouvoir les distribuer facilement aux participants.

Il sera également nécessaire de sortir

Piece	QT	Pourquoi faire
Ordinateurs avec mBlock 5 de préinstaller	1	Pour coder et téléverser
Fer à souder	1	Pour souder
Pince coupante	1 pour 3	Pour couper les pattes des composants
De l'étain	2	Pour souder
Une pompe à dessouder	1 pour 4	En cas d'erreur pour dessouder
Des cable miniUsb-B	1	Pour téléverser les données dans l'Arduino Nano



1. Susciter un questionnement (10 minutes)

L'objectif de cet atelier n'est pas que purement technique. L'idée est de répondre à un besoin par un objet programmable.

A titre d'exemple, le "cahier des charges" initial de ce projet était :

- Avoir un objet qui ne soit pas un téléphone
- Avoir un objet avec lequel on n'interagit qu'une fois au début puis qui nous guide dans notre activité
- Avoir un bip toute les 40 secondes 7 fois de suite pour un enchaînement de sport
- Avoir deux bips d'affilé pour marquer le début et la fin d'une séance de sport
- Avoir un indicateur lumineux pour savoir où on en est dans l'effort

De la même manière le début de l'atelier est une petite séance de réflexion pour savoir à quelle problématique peut répondre cette objet. Libre à chacun d'avoir sa raison, en cas de panne d'idée en voici quelques une :

Discours : " Dans la vie de tous les jours, le temps est quelques chose d'important, d'ailleurs nous avons plusieurs outils qui nous permettent de le mesurer. Aujourd'hui vous allez en construire un pour répondre à un besoin spécifique que vous allez choisir. Dans votre vie de tous les jours, quand avez-vous besoin de mesurer le temps avec précision ?"

Idée :

- Un minuteur pour faire ses devoirs ou pour pratiquer un instrument, l'indicateur lumineux nous donne une idée de la progression sans interrompre la séance, contrairement à un téléphone qui est source de distraction.
- Un minuteur pour se brosser les dents, parce qu'on n'a pas envie d'emporter son téléphone dans la salle de bain ou qu'on en a marre d'avoir le téléphone à la main toute la journée.
- Un minuteur pour faire un œuf dur
- Un minuteur pour faire une bonne tasse de thé
- Le temps de cuisson d'un plat en particulier
- Le temps d'un programme d'une machine à laver pour se rappeler quand il est fini.

Augmentation : L'objet peut être transformé en métronome avec un potentiomètre, éventuellement avec éventuellement un écran, ou bien le bouton peut servir de "tap-tempo".

2. Réaliser le circuit électronique (50 minutes)

a. Notions clefs

Arduino Nano : la carte Arduino Nano est une carte de développement sur laquelle se trouve un micro-controller. Cette carte est programmable grâce au langage Arduino en utilisant soit Arduino IDE, mBlock 5 ou Visual Code Studio avec le plug-in Platformio.

Entré : désigne une broche par laquelle la carte va recevoir une information

Sortie : désigne une broche par laquelle la carte va envoyer une information

Buzzer : Buzzer électromagnétique produisant un son lorsque reçoit une oscillation. Le son produit dépend des caractéristiques de l'oscillation.

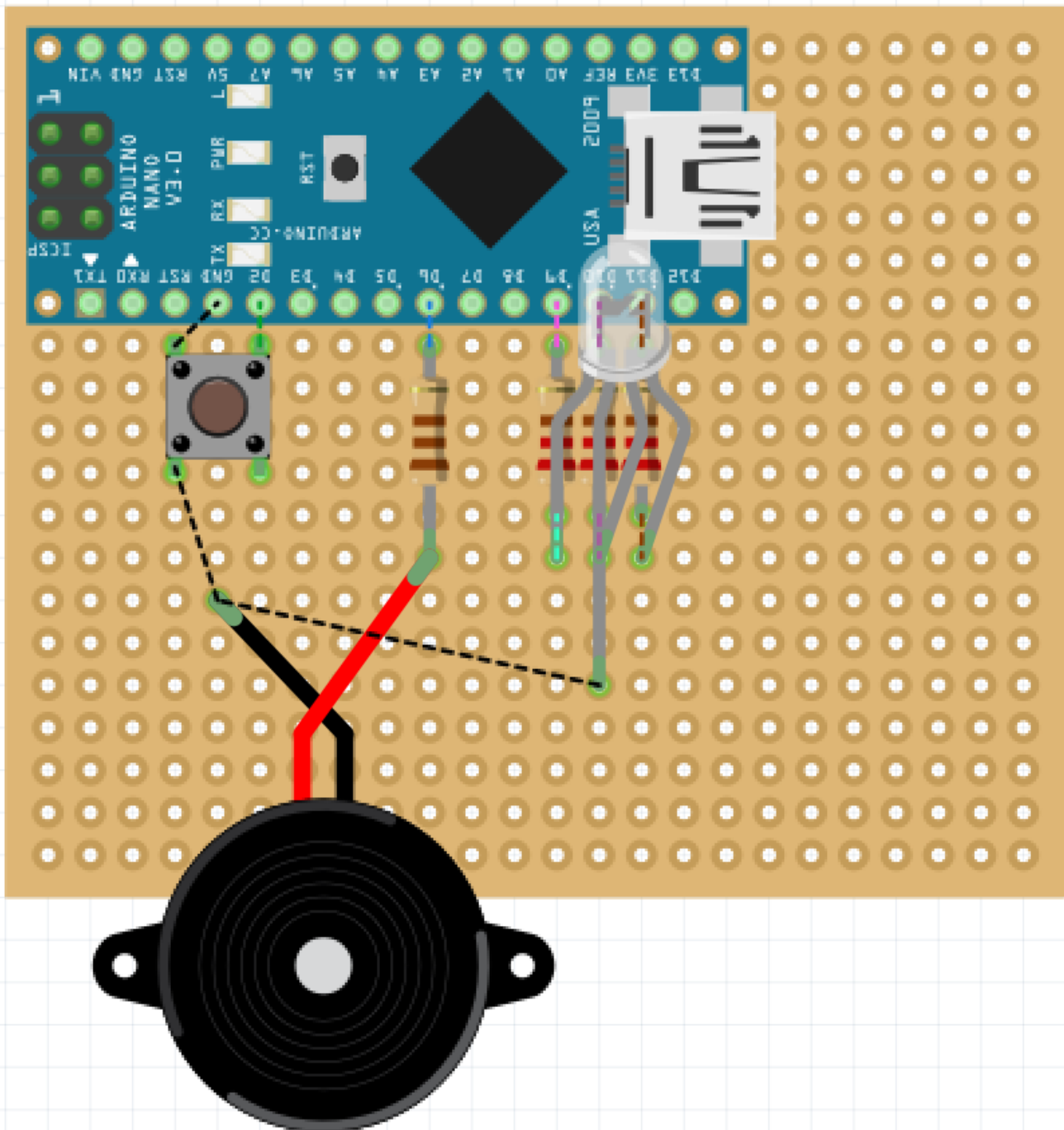
LED RGB cathode commune : désigne une LED à 4 broches

- La broche de gauche (1) contrôle la couleur rouge
- La plus longue (2) est la broche
- La broche à sa droite (3) contrôle la couleur verte
- La broche la plus à droite (4) contrôle la couleur bleue

Résistance : Les résistances “freinent” le courant électrique pour (dans notre cas) protéger les composants électroniques.

Bouton poussoir : Le bouton poussoir est une sorte d’interrupteur, lorsque l’utilisateur appuit dessus, il ferme le circuit, la carte arduino va percevoir ce changement.

b. Fonctionnement du circuit



Toutes les actions sont contrôlées par l'Arduino Nano, c'est le **cerveau de l'opération**.

1. Pour lancer le Timer, l'utilisateur appuie sur le bouton poussoir (**c'est un senseur**). L'Arduino perçoit cette interaction. En réponse à cette situation il initie une séquence de code particulière.
2. L'arduino envoie une oscillation dans le buzzer pour émettre un bip (**c'est un actionneur**), signalant le début du décompte.
3. L'arduino peut Allumer/Éteindre certaines couleurs des LEDS (**ce sont également des actionneurs**) pour donner un message lumineux
4. L'arduino attend un temps défini par l'utilisateur
5. A la fin de ce temps l'Arduino renvoie un signal au buzzer pour lui indiquer de bipper une seconde fois pour annoncer la fin.
6. L'arduino éteint les LEDS
7. L'arduino reste à l'écoute en attendant que quelqu'un appuie de nouveau sur le bouton pour relancer le programme.

Le circuit se compose d'une entrée d'information sur laquelle est branchée le senseur (bouton) et de quatre sorties de type PWM (Pulse Width Modulation) sur lesquelles sont branchées les actionneurs (le buzzer est les 3 broches de la LED RGB). Ci dessous, la liste des entrées/sorties et la broche correspondante.

Entré d'information :

- D2 : Le bouton poussoir

Sortie d'information :

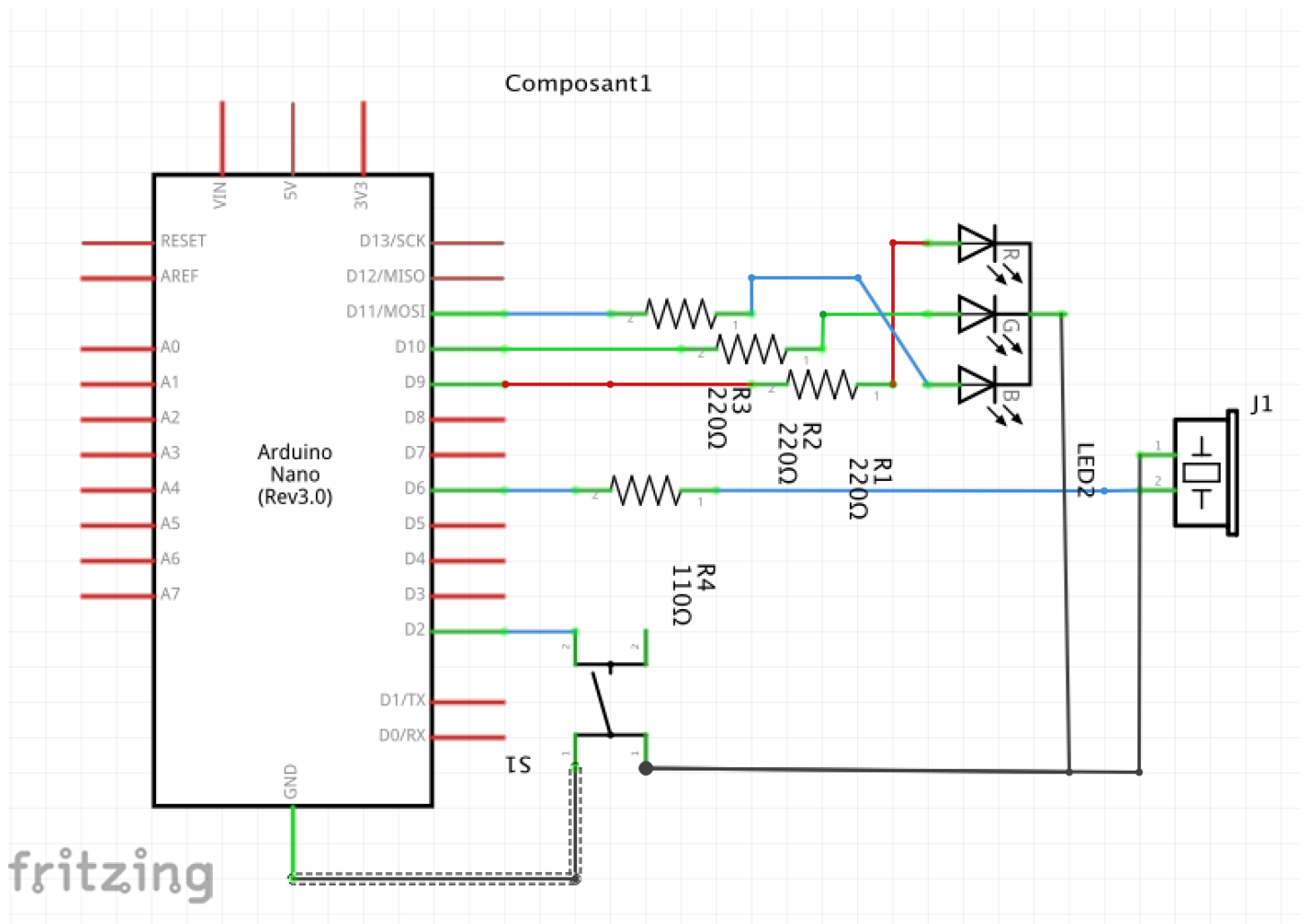
- D6 : Sortie PWM pour le buzzer
- D9 : Sortie PWM pour la broche "Rouge" de la LED
- D10 : Sortie PWM pour la broche "Verte" de la LED
- D11 : Sortie PWM pour la broche "Bleu" de la LED

Voici un très bon tutoriel de “La Fabrique DIY” pour programmer une LED RGB :

<http://www.lafabriquediy.com/tutoriel/led-rgb-arduino2-1-223/>

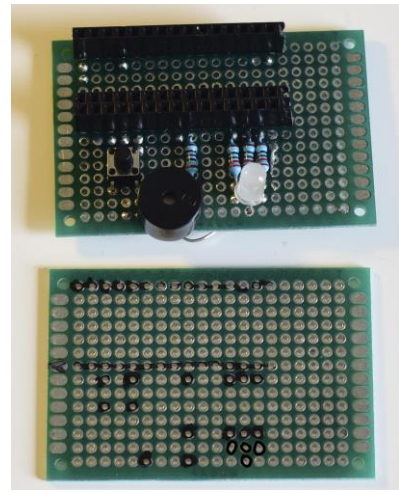
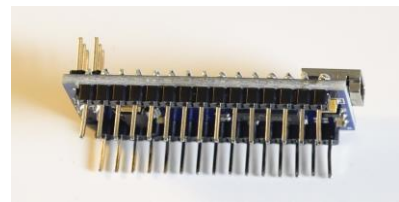
Pour en savoir plus sur les broche PWM : <https://www.locoduino.org/spip.php?article47>

c. Construire et souder le circuit

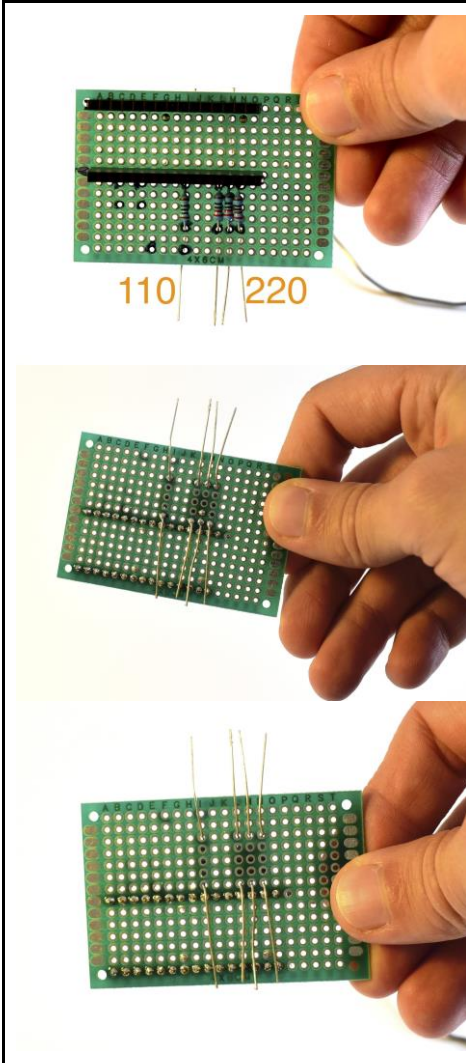


Ce circuit peut être soudé en 15/20 minutes dans les mains de quelqu'un d'expérimenté. Dans ce circuit il n'est pas utilisé de câbles ou autres, seules les "pattes" des composants sont utilisés pour faire toutes les connexions.

Une bonne pratique est de souder les composants du moins encombrant au plus encombrant.

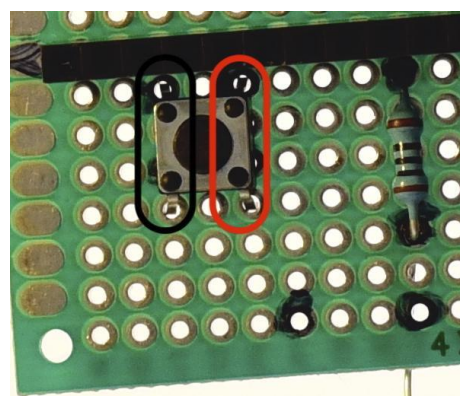
Photo	Etapes	
	<p>0. L'étape de base est de s'assurer que tous les participants sachent correctement souder. Il est temps de les laisser pratiquer un petit peu avant de commencer.</p>	<p>Tips :</p> <p>https://www.youtube.com/watch?v=8oGjG9uyYq8</p> <p>https://www.youtube.com/watch?v=FTYDSCUUDSQ</p>
	<p>1. Repérer sur la carte l'emplacement de tous les composants pour ne pas se tromper par la suite</p>	
	<p>2. Si l'arduino Nano n'a pas de broche mâle, les souder sur la carte. Ces broches servent à brancher l'arduino Nano sur une breadboard ou le cas échéant sur le circuit</p>	

	<p>3. Commencer par souder les deux rangées de broches femelles sur la carte.</p>	<ul style="list-style-type: none"> • Commencer par ne souder qu'une des broches sur la carte • Assurez-vous que la rangée est correctement positionnée • Souder le reste de la rangée
	<p>4. Placer et souder les résistances</p>	<ul style="list-style-type: none"> • Commencer par couder les résistances pour leur donner la forme d'une agrafe, ainsi elle rentreront facilement dans leurs emplacements • Il y a deux valeurs de résistance, ne vous trompez pas, les trois résistances de



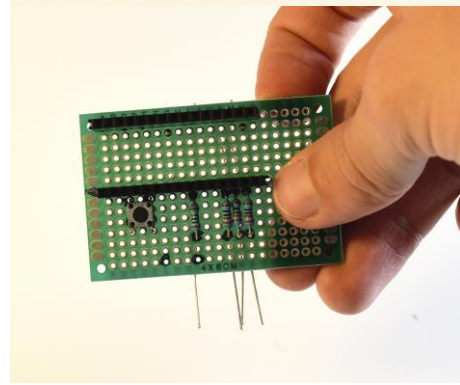
220 ohms vont sur les broche D9, D10 et D11.

- Les résistance n'ont pas de sens, souder les comme vous le souhaitez



**Les pattes du bouton
sont reliées deux par
deux**

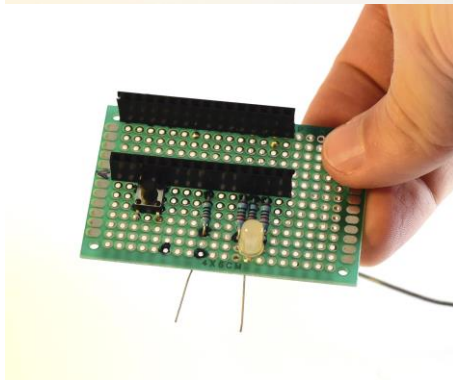
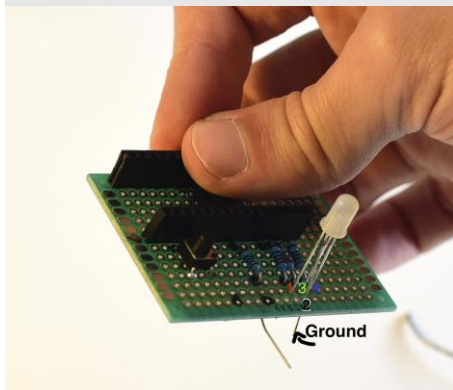
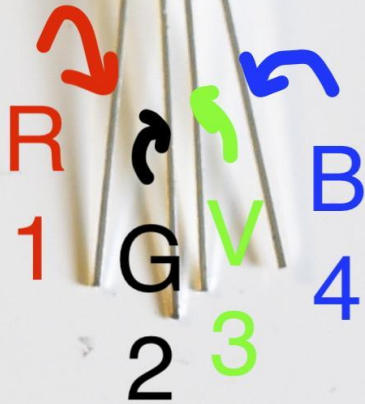
**Placer le bouton comme
indiqué ici**



5. Placer et souder le bouton poussoir

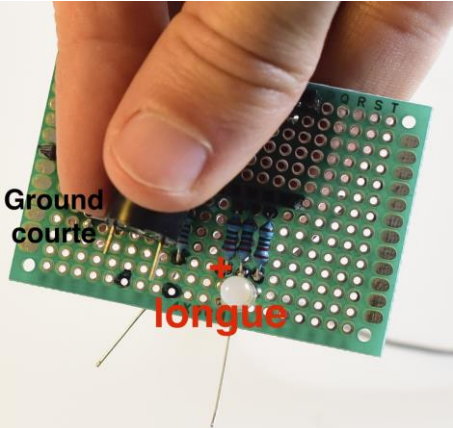
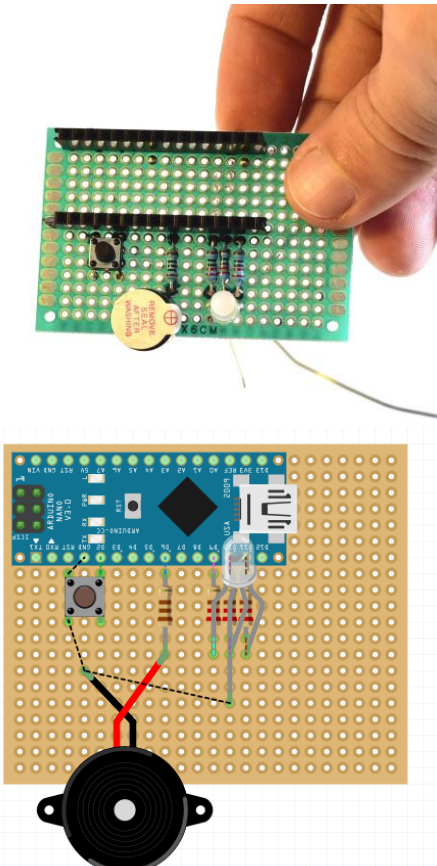
- Attention, le bouton poussoir a 4 broches. Elles sont reliées deux par deux. Lorsque l'on appuis les 4 broches sont en contact. Il y a donc un sens dans lequel le brancher.
- Attention les pattes se plient facilement, positionner le bouton sur la board avec précaution
- L'une des pattes doit être soudée au pied de la broche D2.

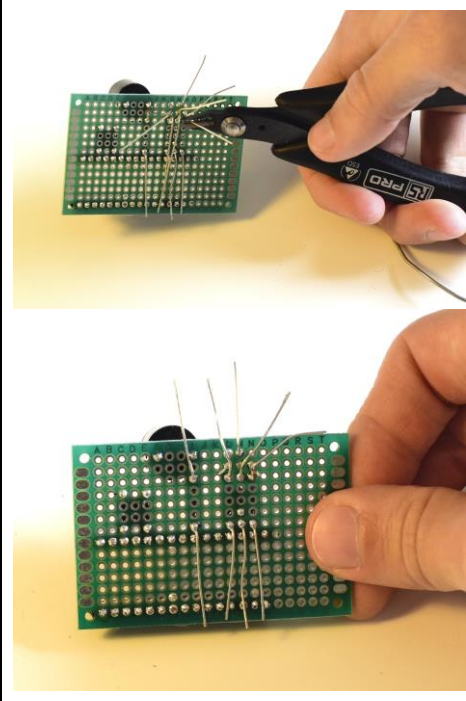
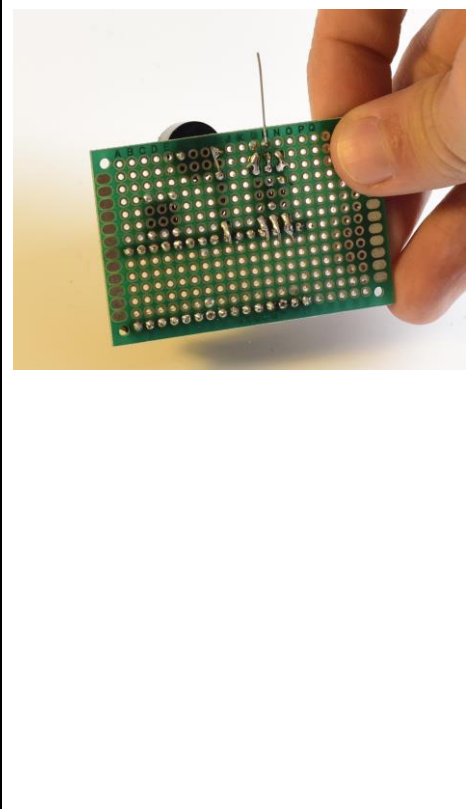
R = Rouge
V = Vert
B = Bleu
G = Ground

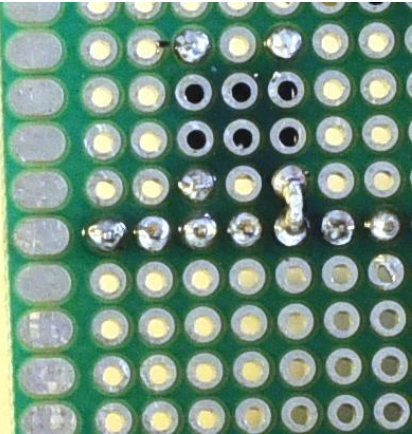
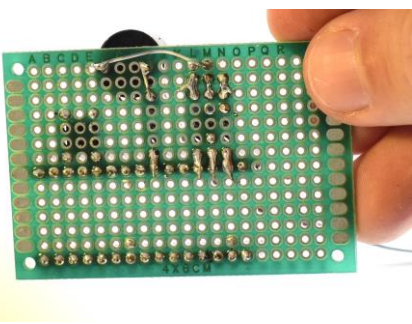
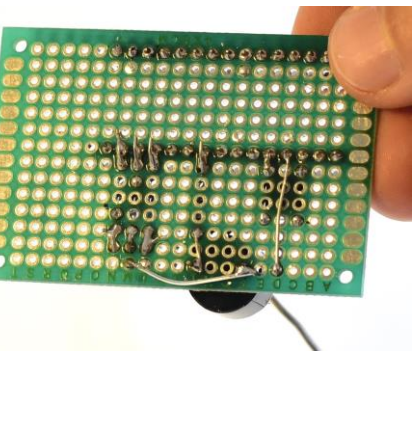


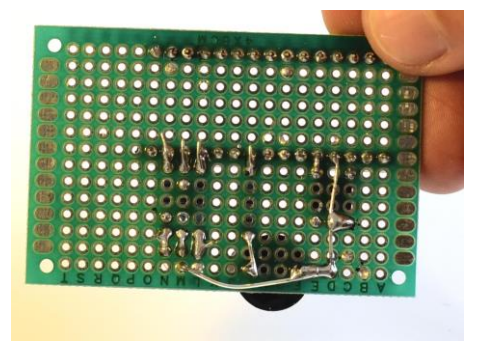
6. Placer et souder la LED

- Le point de référence est la broche la plus longue, c'est la broche que l'on doit brancher sur le ground, c'est la broche numéro 2
- La broche "Rouge" (1) se soude juste en dessous de la résistance alignée avec la broche 9
- La broche "Verte" (3) se soude juste en dessous de la résistance alignée avec la broche 10
- La broche "Bleue" (4) se soude juste en dessous de la résistance alignée avec la broche 11

	<p>7. Placer et souder le buzzer</p>	<ul style="list-style-type: none"> • Les buzzers ont un sens, la patte la plus longue correspond au + et doit se trouver dans le trou en dessous de la résistance menant à la broche 6 de l'arduino • La patte la plus courte correspond à la patte "Ground".
	<p>8. Faire un point avant de fermer le circuit</p>	<ul style="list-style-type: none"> • A ce stade voilà à quoi ressemble le circuit vu du dessus. • Toutes les connections sont à faire par le dessous • Sur le schéma ci joint, les connections à réaliser sur la face du dessous de la carte sont indiquées en pointillés

	<p>9. Élaguer quelques pattes pour y voir plus clair</p>	<ul style="list-style-type: none"> ● /!\ Cette étape demande de l'attention, il ne s'agirait pas de couper les mauvaises pattes ● Couper les pattes Rouge (1), Vert (3) et Bleu(4) de la LED. NE COUPER PAS LA PATTE GROUND (2) - Elle nous servira pour fermer le circuit et la souder avec le ground
	<p>10. Relier les actionneurs aux résistances puis les résistances aux broches de l'arduino</p>	<ul style="list-style-type: none"> ● Utiliser les pattes des résistances pour relier <ul style="list-style-type: none"> ○ D6 et le + du buzzer ○ D9 et le Rouge ○ D10 et le Vert ○ D11 et le Bleu ● Le fait d'utiliser la patte de la résistance rend l'opération très simple et rapide. Si jamais une patte avait été coupée par accident la liaison peut se faire juste avec de l'étain ● Une fois fait, vous pouvez couper les pattes des résistances ● Mettez de côté les pattes coupées ● /!\ NE COUPEZ PAS LA PATE GROUND DE LA LED

	<p>11. Relier le bouton poussoir à la broche 2</p>	<ul style="list-style-type: none"> • Utiliser de l'étain pour relier la patte du bouton la plus proche de D2 à D2
	<p>12. Relier le ground de la LED au ground du buzzer</p>	<ul style="list-style-type: none"> • Le Ground est commun à tous les composants • Souder la patte Ground de la LED à la patte Ground du buzzer
	<p>13. Faire un rail "Ground" à l'aide d'une des pattes coupés</p>	<ul style="list-style-type: none"> • Prenez la patte la plus longue que vous avez coupé. • Souder une extrémité à la la broche Ground de l'arduino et l'autre le plus proche possible de la patte Ground du buzzer • Faites attention à ne pas toucher d'autres composants avec

	<p>14. Relier tous les grounds au rail préalablement construit</p>	<ul style="list-style-type: none"> • A l'aide de l'étain, relier une des pattes du bouton poussoir, qui n'est pas du côté de D2, au rail Ground • Faites de même pour le Ground du Buzzer/LED
<p>Fin !</p>		

3. Coder l'Arduino Nano pour faire fonctionner le circuit (1 heure)

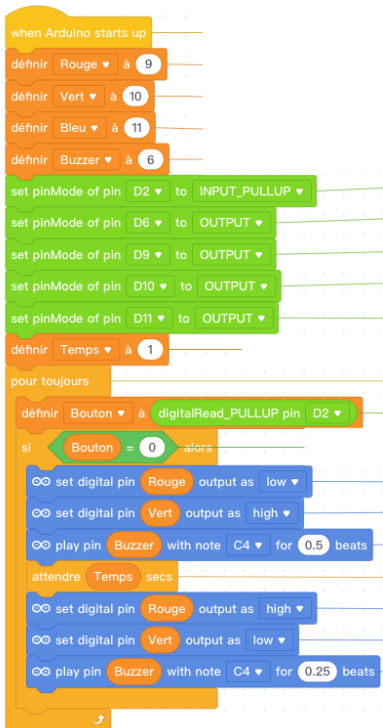
Cette partie est extrêmement modulaire :

- Les participants pourrait simplement ouvrir mBlock, changer la variable "Temps" pour indiquer le temps qu'ils aimeraient mesurer
- Les participant peuvent changer la variable temps et customiser le programmer pour qu'il fasse une mélodie au début et à la fin
- Les participants peuvent programmer les LEDs
- Les participants peuvent reconstruire le programme de zéro

Il est conseillé de se **former** au préalable sur le fonctionnement du logiciel mBlock et son **système de glisser déposer**. **D'explorer** en amont les différentes **sections** du logiciel, le système de **variable** et les **fonctions proposées**.

Vous trouverez en annexe 3 fichiers mBlock correspondant à un programme de niveau débutant, intermédiaire et avancé. En ce qui concerne le fichier débutant, chaque ligne est commentée pour en comprendre son fonctionnement.

a. Notions clefs



Le code est découpé en deux parties :

1. La partie "Setup" (ici "**When Arduino starts**"): Ne s'exécute qu'une seule fois au démarrage. Est utilisé pour définir un certain nombre de paramètres.
2. La partie "Loop" (ici "**pour toujours**"): s'exécute en boucle tant que l'arduino est allumé.

La notion de **variable** est primordiale. Il s'agit d'une entité à laquelle nous allons attribuer une valeur. X est une variable, nous pouvons définir X=2 ou X=4.

Fonctions :

pinMode = indique à l'arduino s'il s'agit d'une entrée ou d'une sortie

INPUT_PULLUP = C'est un mode particulier qui permet d'utiliser un bouton poussoir sans résistance. Dans ces conditions, quand le bouton n'est pas appuyé il renvoi une information = 1, lorsque quelqu'un appui sur le bouton, le bouton renvoi à l'arduino une valeurs égal à 0.

Attendre = Attend pour une durée déterminé

Set digital pin output as high/low = envoie ou non du courant dans une broche.

Play pin = Joue une note pour une durée donnée

Si... Alors = Si une condition est remplie alors exécute une portion de code.

Pour plus d'information, je vous invite à consulter le site locoduino.org.

b. Proposition de déroulé pour débutant

0. Tout le monde ouvre le fichier code
1. Expliquer la notions de variable et les différentes fonctions utilisées
2. Téléverser le programme dans l'arduino et voir ce qu'il se passe
3. Demander aux participants de changer la valeur de la variable temps et voir ce qu'il se passe. Il vaut mieux rester dans de petites valeurs pour ne pas attendre 10 minutes entre chaque essai
4. Demander aux participants de changer les Leds qui s'allument et s'éteignent, combiner les différentes couleurs
5. Demander aux participants de changer la note jouée par le buzzer et sa durée
6. Demander aux participants de composer une mélodie de fin pour la séquence
7. Demander aux participants de programmer leur timer comme ils le désirent

c. Variantes et approfondissement (Dépend du niveaux du groupe)

i. Reconstruire le code

1. Au lieu et à la place d'ouvrir le fichier code en entier, ne laisser que la partie "when arduino starts up".
2. Utiliser la fonction "attendre" et "set digital pin output as" pour faire clignoter une LED
3. Ajouter à cela le bip
4. Expliquer la fonction "Si... Alors"
5. Ajouter l'utilisation du bouton pour l'utiliser comme déclencheur
6. Ecrire le code final

ii. Utiliser des fonctions avancées et manipuler les variables

1. Ouvrir le code en vue de l'améliorer
2. Intégrer la fonction "for" et "Set Pwm output as" pour que les LEDs changent de couleur progressivement au fil du temps.
3. Ajouter des mélodies de début et de fin.
4. Construire un bloc indépendant capable de jouer une mélodie en utilisant les blocs customisables.