

Électronique pour manipulateur double contacts

F6HQP le 28/12/2023

Le cœur du montage est un microcontrôleur Arduino Nano que l'on trouve pour quelques euros sur le net. La programmation se fera non pas avec des lignes de code mais via des blocs qui seront automatiquement traduits en ligne de code. La programmation devient ainsi très facile et accessible pour des novices. Le programme a été développé pour l'éducation et une version francisée se trouve sur le net : ARDUBLOCK EDUCATION chez Technologie Services https://technologieservices.fr/ts_fr/ress-175345.html

Nota : il existe des keyeurs beaucoup plus sophistiqués avec de multiples paramètres que l'on peut changer directement en manipulant et avec des mémoires dynamiques.

On vise ici la simplicité et une réalisation entièrement « home made » incluant la programmation.

Caractéristiques du montage :

- alimentation secteur car utilisation au QRA
- branchement manip double contacts
- inversion possible points/traits
- branchement manip simple contact
- réglage vitesse
- moniteur avec réglage de la note et puissance
- jusqu'à 7 mémoires pré-enregistrées
- commande pour tune
- DEL de manipulation et sortie sur relais

A- Schéma électronique

le Nano a 30 connections mais toutes ne sont pas utilisées

1- alimentation

Une diode protège des inversions de polarité, un régulateur 5V alimente la broche #27 et le reste du montage. Ne pas oublier la masse avec les broches #29 et #4

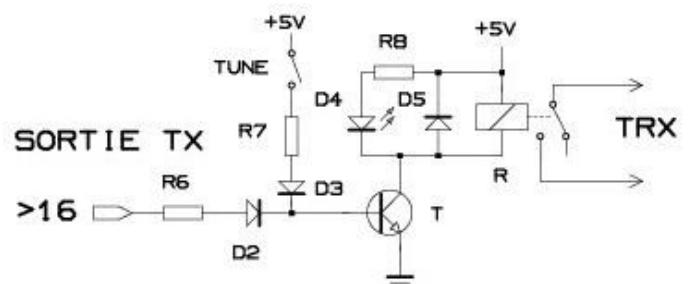
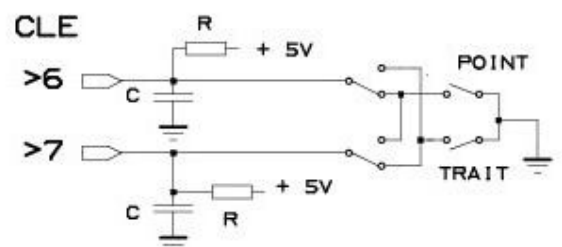
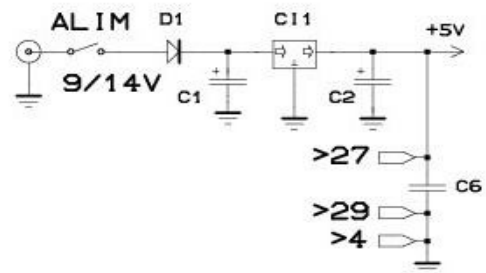
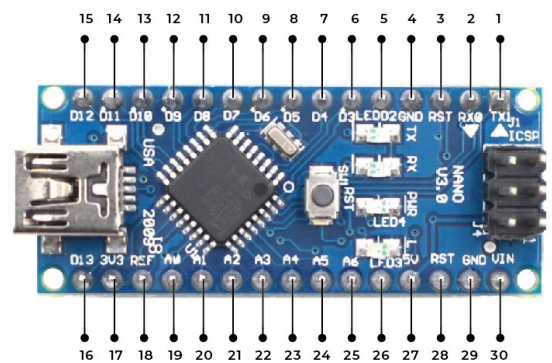
2 – les résistances mettent à l'état haut (1) les broches #6 et #7, Le manip mettra ces broches à la masse (0). Un inverseur permet de changer les points/traits

3 – Sortie TRX #16 : le transistor alimente le relais REED,

et aussi une DEL (facultatif)

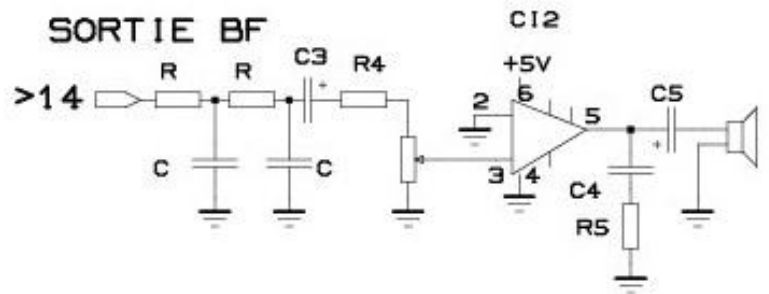
un inter « tune » permet de faire coller le relais pour un réglage du TRX (facultatif)

La sortie du relais est laissée flottante par rapport à la masse du montage. Sage précaution.

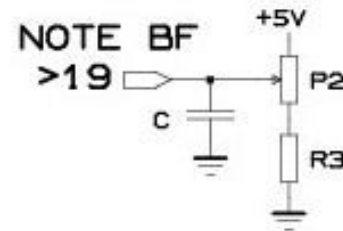


4 – Sortie BF (facultatif). Un filtre diminue les aigus, suivi d'un ampli LM386. R4 diminue le signal encore trop généreux. CI2 fonctionne très bien sous 5V.

*** une capa céramique (non représentée) de 10nF a été ajoutée entre la broche d'entrée 3 du CI et la masse pour éviter une perturbation HF, a mettre près du CI



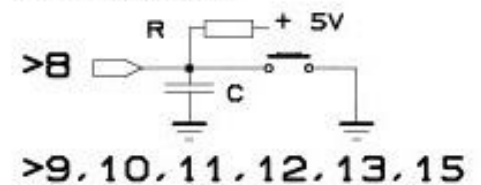
Le réglage de la note B se fait avec la broche #19. R3 évite les tonalités en dessous de 300 Hz



5 – Mémoires (facultatif)

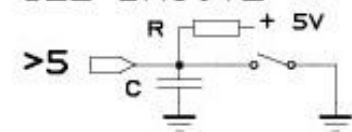
de 1 à 7 mémoires . Comme pour la clé, résistance qui met à l'état haut la broche, si 7 mémoires répliquer 7 fois le schéma

MEMOIRES



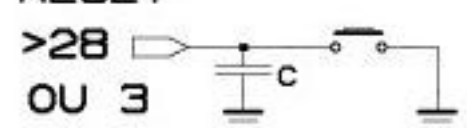
6 – Clé droite (facultatif): même principe que pour les mémoires

CLE DROITE



7 – bouton RST ou RAZ. Très commode si on veut interrompre le déroulé d'une mémoire. Broche #28 ou #3

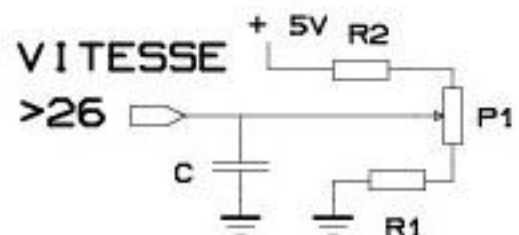
RESET



8 – Vitesse

sur la broche #26

R2 évite les vitesses trop lentes et R1 trop rapides ce qui permet d'avoir un réglage plus confortable

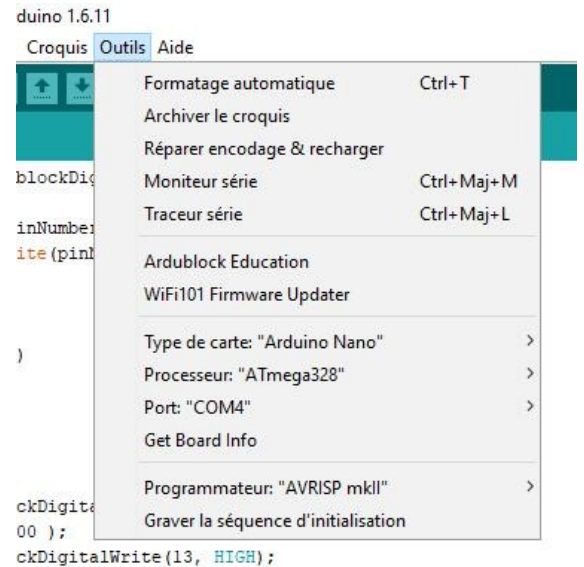


B programmation

Le programme ardublock lancé, mettre le câble USB sur la carte nano et le PC. Il faut aller dans « outils » pour choisir le type de carte « Arduino nano », puis le processeur « ATmega328 » puis le port COM

cliquer alors sur « Ardublock Education » pour avoir un nouvel écran avec le menu des blocs.

Pour tester dessiner un mini programme de clignotant

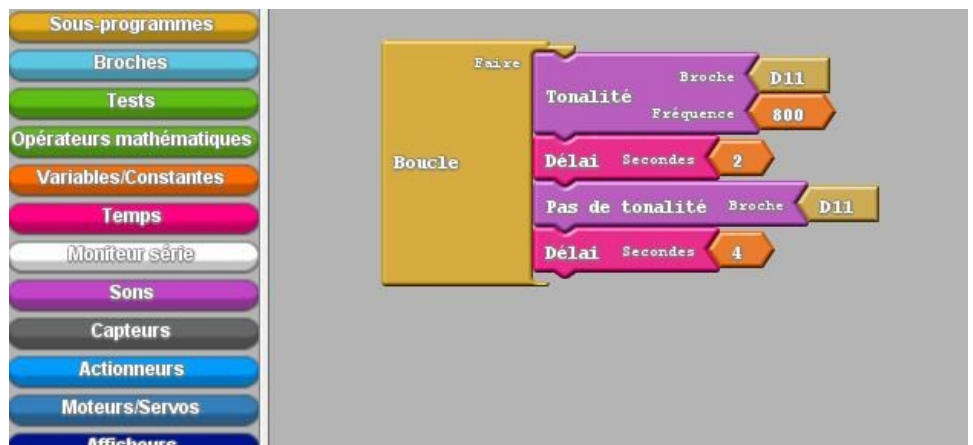


l'enregistrer puis cliquer sur « Téléverser vers l'Arduino » : on voit les lignes de code dans l'écran Arduino (cliquer en bas sur l'onglet Arduino) et un bandeau vert s'allonger. Cela doit clignoter.

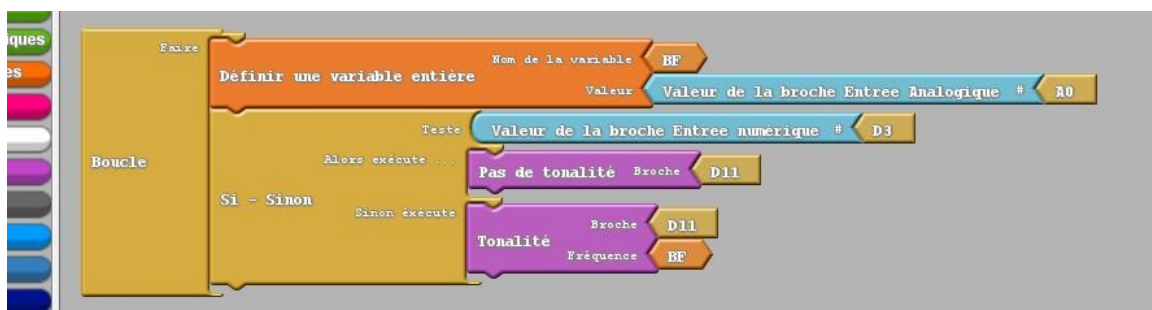
On peut commencer à programmer notre keyeur. Cela se fera progressivement par étapes.

1 – sortir un son

le son sortira de la broche #14 repérée D11 sur l'Arduino. La fréquence est ici fixée à 800Hz. 2 secondes de son puis 4 secondes de silence.

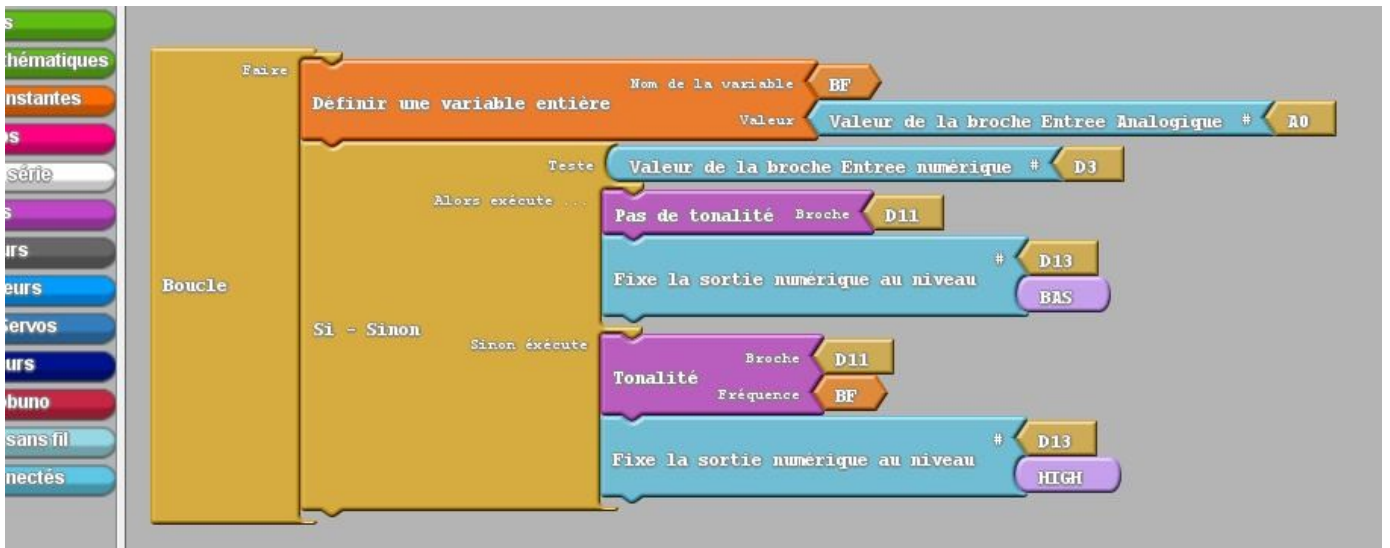


2 – son variable commandé par un coté du manip

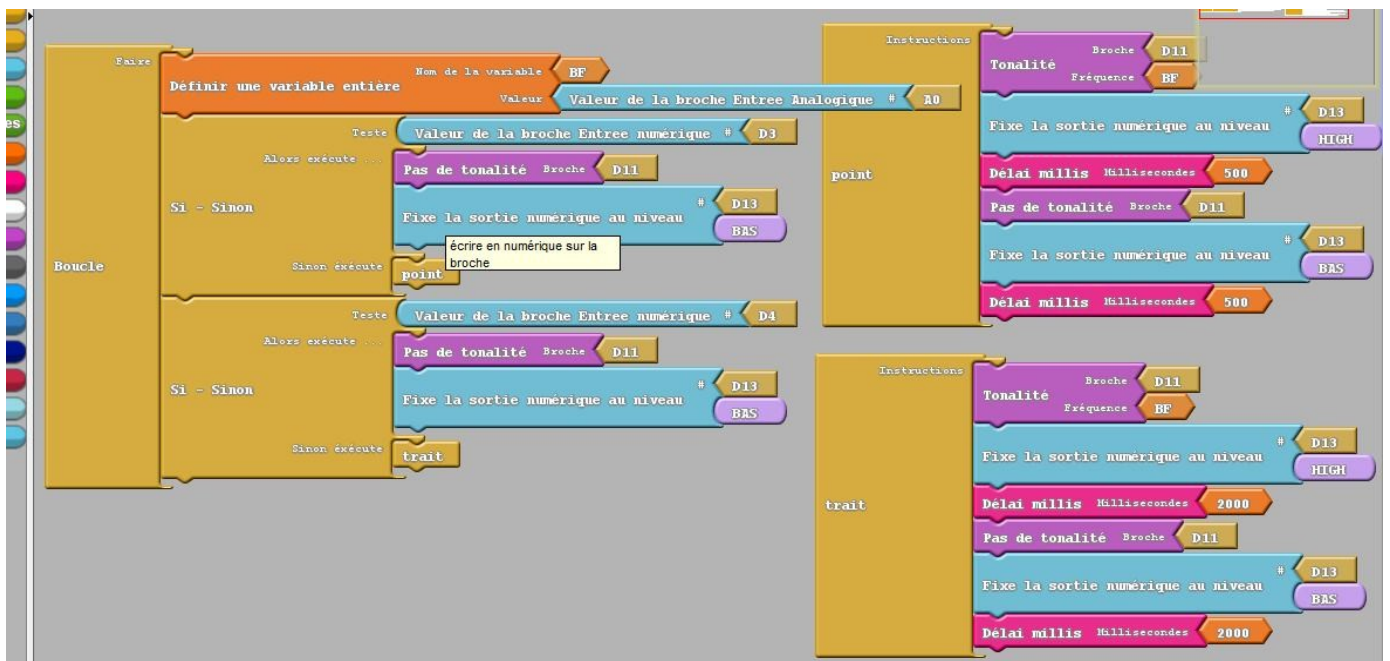


On définit une variable que l'on appelle BF. Sa valeur va dépendre de la position du potentiomètre sur la broche #19 appelée A0. Cette valeur varie de 0 à 1024 ce qui donnera directement la fréquence en Hz. Un appui sur un côté du manip met la broche #6 nommée D3 à la masse et cela sort un son. Rappel : les broches d'entrées ont une résistance qui les met à l'état haut (1) au repos.

3 – activation de la sortie TX sur la broche #16 nommée D13



4 – manipulateur double contacts

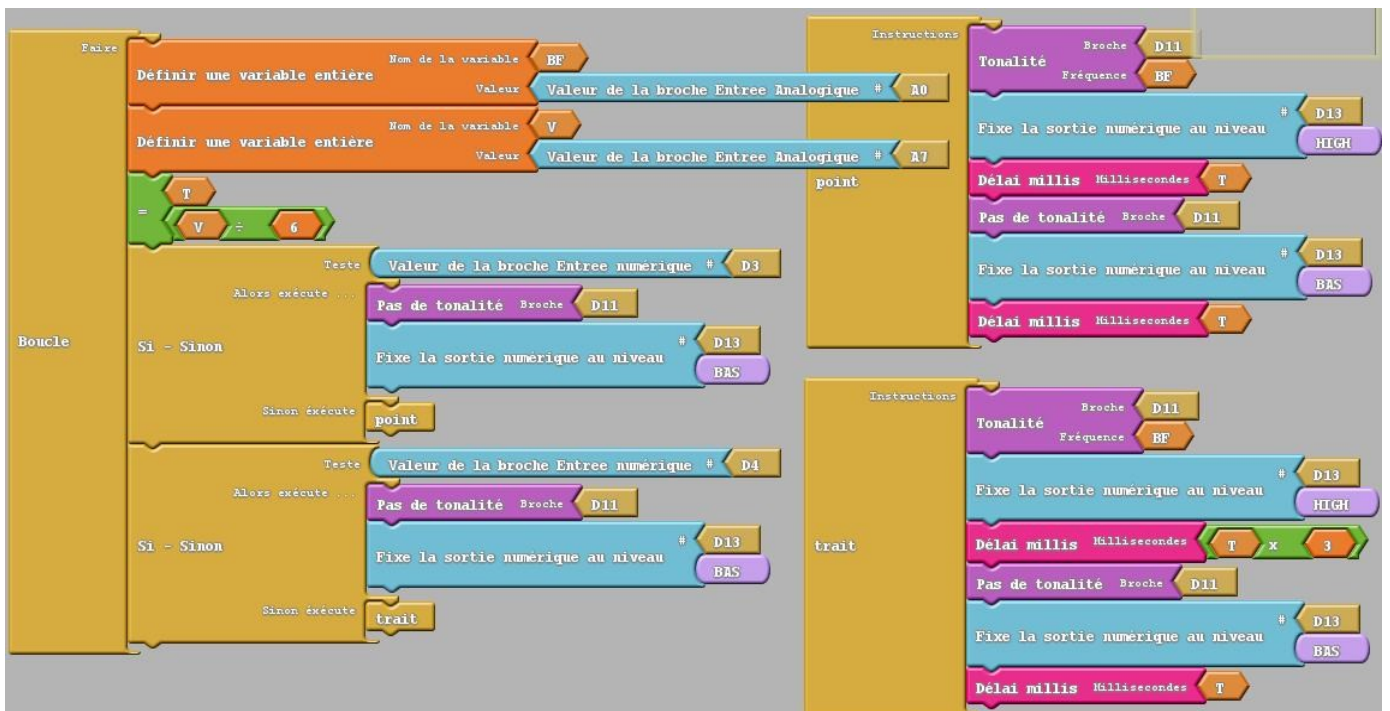


On va créer deux sous programmes que l'on va nommer « point » et « trait »

La mise à la masse de #6 D3 va appeler le sous programme « point », idem pour #7 D4 le sous programme « trait ».

pour le point : on active la sortie BF et la sortie TX pendant 1/2 seconde (arbitraire pour l'instant) puis on arrête tout pendant 1/2 seconde. Même principe pour le trait.

5 – variation de la vitesse de manipulation



Là il faut créer une deuxième variable *V* qui va dépendre du potentiomètre relié à la broche #26 A7. Mais on ne peut pas directement utiliser cette valeur car c'est trop lent sur une grande partie de la course du potentiomètre ! On va donc passer par une autre variable intermédiaire *T* qui va diviser par 6 la valeur de *V*. C'est cette valeur *T* qui sera utilisée pour les points puis multipliée par 3 pour les traits.

6 – mémoires



Pour chaque mémoire il faut ajouter un bloc si-sinon et faire un sous programme avec des points et des traits. Pour les délais entre les lettres il faut multiplier *T* par 2 car il y a déjà une valeur de *T* à la fin d'un point ou d'un trait. Entre deux mots il faut multiplier *T* par 4.

Il n'y a pas beaucoup de place dans la fenêtre pour placer les blocs mais on peut les superposer (sans les lier bien sûr!)

comme mémoire j'ai mis mon CALL, OK 5NN TU pour les expéditions, QTH, Prénom, CQ ... , 5NN73

7 – ajout clé droite



Il faut ajouter un bloc « tant que ... » qui sera commandé par la broche #5 D2. Mais comme celle-ci est au niveau haut il faut inverser la valeur du test par le bloc « non ». au repos le 1 devient 0, si l'on appuie sur la clé, le 0 devient 1.

Récapitulatif des différentes broches utilisées :

#26 A7 réglage de la vitesse
#19 A0 réglage fréquence BF

#16 D13 sortie TX
#14 D11 sortie BF

#6 D3 manip points
#7 D4 manip traits
#5 D2 clé droite

#28 ou #3 RST reset
#27 +5V alimentation
#4 et #29 masse

#8 D5 mémoire
#9 D6 mémoire
#10 D7 mémoire
#11 D8 mémoire
#12 D9 mémoire
#13 D10 mémoire
#15 D12 mémoire

Composants :

P1 P2 10 k linéaires
CI1 7805
CI2 LM386
Arduino Nano
T 2N2222 ou 2N2219

D1 D5 1N4007
D2 D3 1N914
D4 LED

1 inverseur double
2 interrupteurs
8 boutons poussoirs

C 10 nF céramique
C1 220uF
C2 4,7uF
C3 4,7 ou 10uF
C4 47nF
C5 220uF
C6 100 à 470nF
R 10k
R1 R8 1k5
R2 3k3
R3 2k2
R4 33k
R5 10
R6 R7 2k2