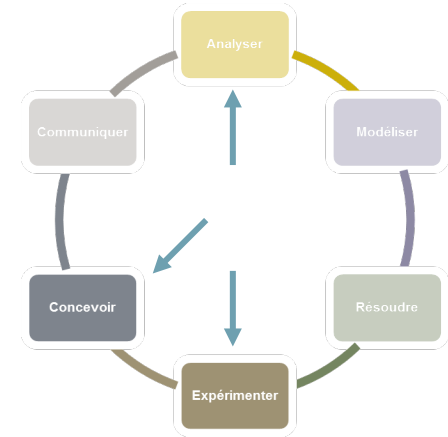


TP CPGE S2

INITIATION À MICROPYTHON 2 : COMMANDE VENTILATEUR



- Concevoir : Générer un programme et l'implanter dans un système cible
- Expérimenter : Etre capable d'effectuer un montage électronique contenant un hacheur et le tester
- *Prérequis : ce sujet fait suite au tp initiation Pico (feux tricolores)*

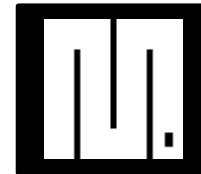
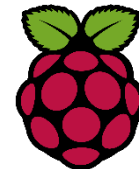
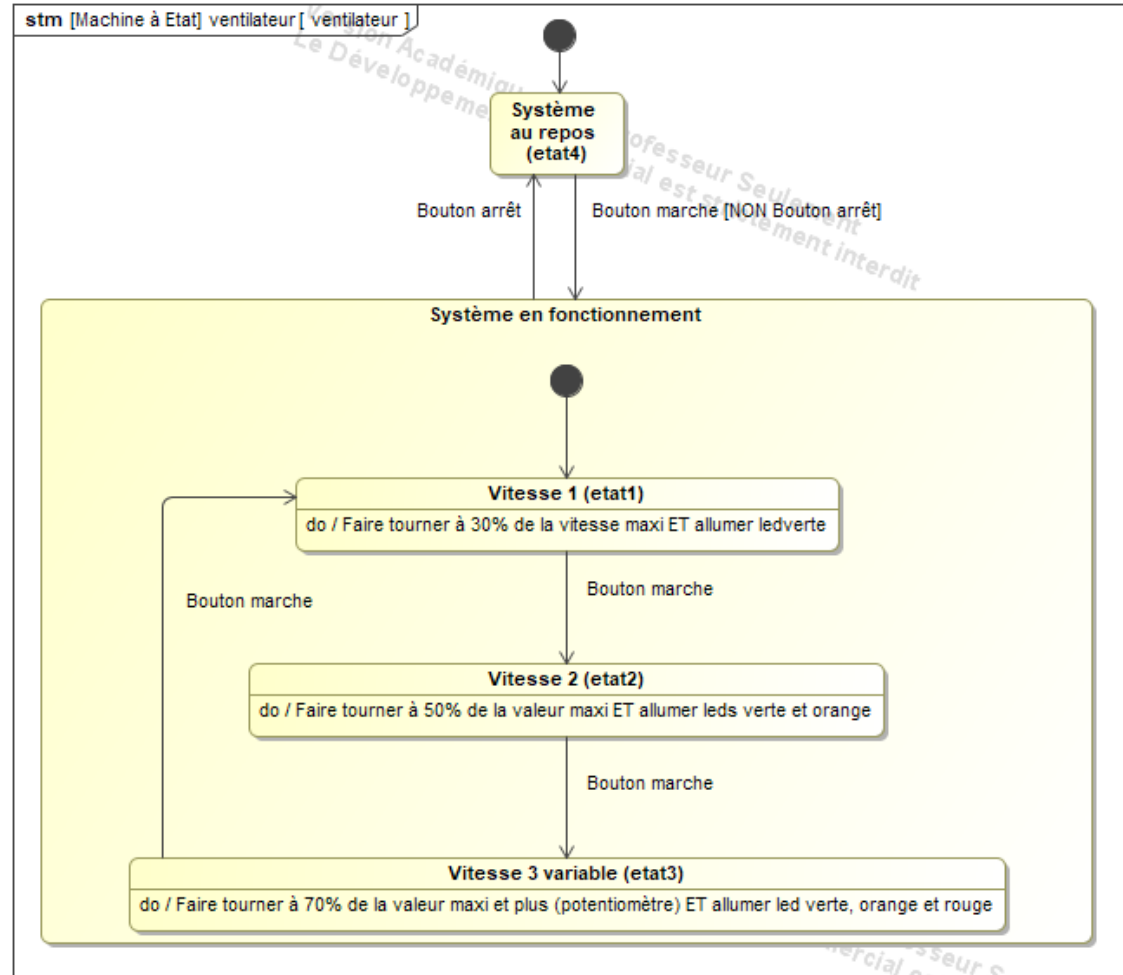


Diagramme d'états

L'objectif final est de réaliser la commande d'un ventilateur à l'aide d'une carte associée à un microcontrôleur en respectant ce diagramme d'états.

Lorsque le système est au repos, tout est arrêté (les leds sont éteintes).

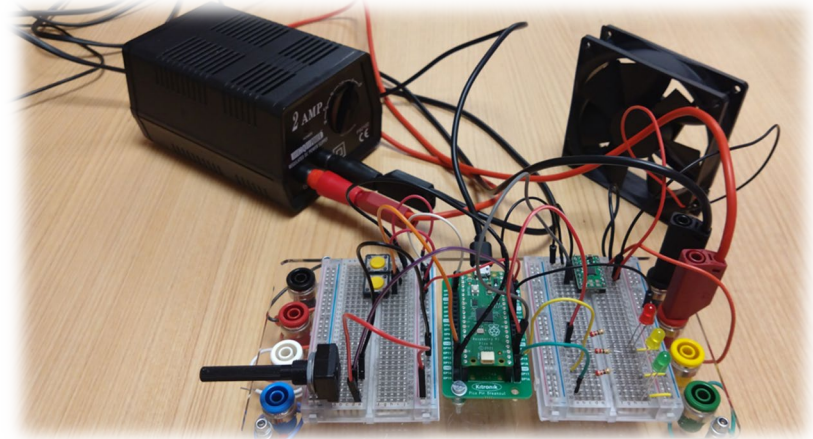
Le potentiomètre angulaire permet de passer de 70% à 100% de la vitesse maximale du ventilateur en mode vitesse 3.



Etapes du TP :

- Etape 1 : Inventaire
- Etape 2 : Commande MLI pour moteur à courant continu
- Etape 3 : câblage des boutons et leds. (voir TP feux tricolores)
- Etape 4 : Programmation Pico
- Etape 5 : Tests
- Etape 6 : Synthèse (description chaine fonctionnelle)
- Etape 7 : Mise en place de l'entrée analogique (potentiomètre) et tests finaux.

Exemple de montage en fin de TP.

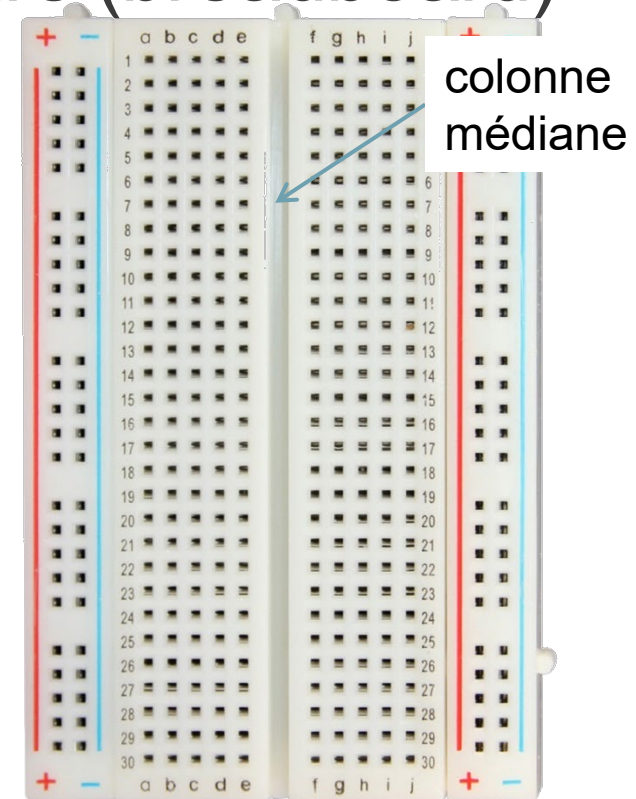


Etape 1 : Inventaire du matériel utile

- Carte Pico et un câble micro USB
- Plaque de prototypage rapide (breadboard)
- 3 leds de couleurs verte, orange (ou jaune) et rouge
- 3 résistances de $470\ \Omega$ environ
- Des fils...
- 2 boutons poussoir
- Un hacheur DRV 8833
- Un ventilateur (ou petit moteur MCC)
- Une alimentation externe 9V-12V (ou coupleur de piles compatible ventilateur)
- Un potentiomètre $10\ \text{k}\Omega$

Plaque montage sans soudure (breadboard)

La plaque permet de relier les composants sans soudure pour des tests.
La colonne + (rouge) sera branchée au 3.3V de la carte, et la colonne – (bleue ou noire) avec GND (ground).
Les composants à multiples entrées (comme le hacheur) doivent être placés à cheval sur la ligne médiane.



En fin de séance, vous laissez le hacheur et les 2 boutons poussoirs sur la plaque, merci.

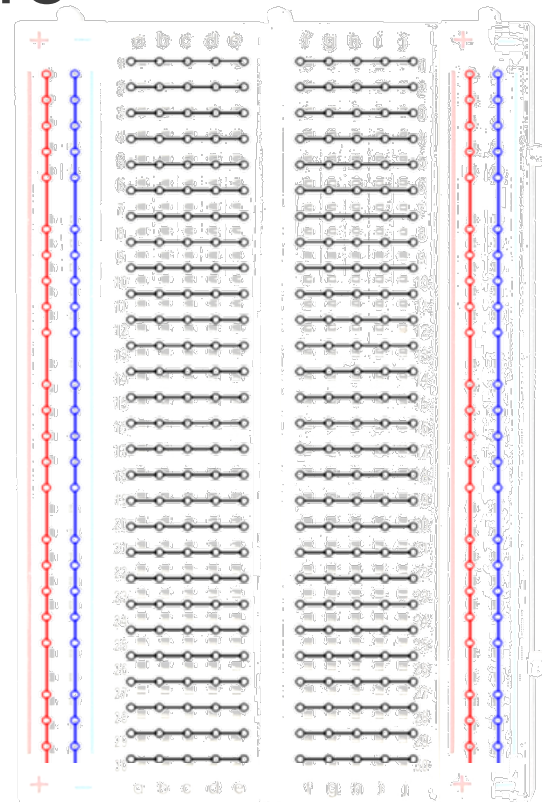
Cliquer sur l'image pour comprendre les connections

Plaque montage sans soudure

La plaque permet de relier les composants sans soudure pour des tests.

La colonne + (rouge) sera branchée au 3.3V de la carte, et la colonne – (bleue ou noire) avec GND (ground).

Les composants à multiples entrées (comme le hacheur) doivent être placés à cheval sur la ligne médiane.



En fin de séance, vous laissez le hacheur et les 2 boutons poussoirs sur la plaque, merci.

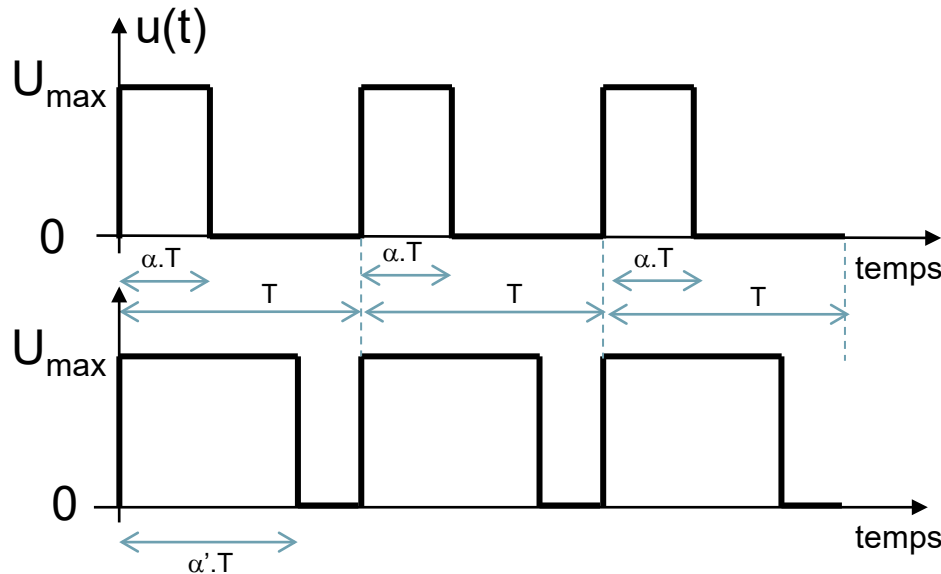


Etape 2 : Commande MLI pour moteur à courant continu

- Avant d'aborder l'ensemble du graphe, on propose de tester le moteur du ventilateur. Dans le ventilateur, une diode a été placée et impose un sens de courant dans le moteur et par conséquent un sens de rotation. Le câblage du moteur doit donc respecter les instructions.
- Le principe de la commande MLI (Modulation à Largeur d'Impulsions – PWM en anglais) exploite le comporte d'un composant appelé hacheur à placer entre la carte Pico et le moteur (fonction moduler)...
- Veuillez noter que chaque différents hacheurs vendus se câblent et se pilotent différemment. Ici on ne parle que du DRV8833 de Pololu.

Etape 2 : Signal PWM

- La carte microcontrôleur Pico est capable de fournir un signal modulé à largeur d'impulsions (PWM, Pulse Width Modulation) pour chaque sortie.
- Voir l'aide micropython (sortie PWM)



Signal PWM : le signal est de type créneau de période constante T . Il prend la valeur U_{\max} pendant $\alpha.T$, une portion de la période T . La tension U_{\max} est fournie par l'alimentation externe (9V ici). α est appelé **rapport cyclique**.

$$\langle u \rangle = \frac{1}{T} \int_0^T u(t) dt$$

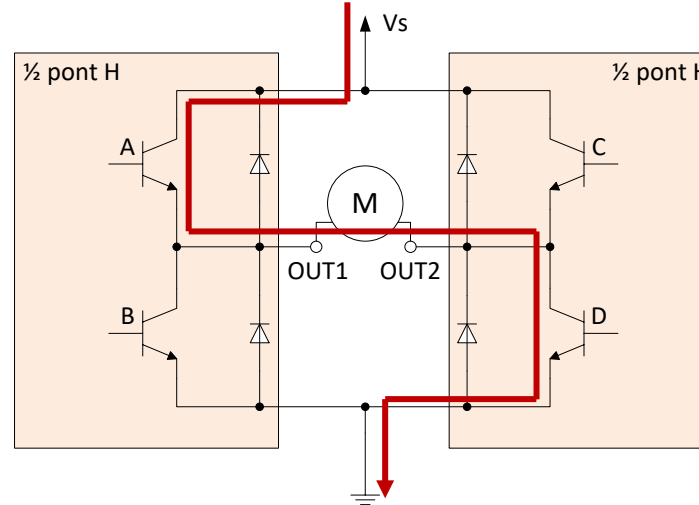
- Calculer $U_{\text{moy}} = \langle u \rangle$ en fonction de U_{\max} . En déduire l'intérêt de cette commande.



Étape 2 : Gestion du moteur (1/4)

- Le hacheur (driver en anglais) DRV8833 propose un mode de commande nommé « fast decay » : chute rapide du courant induit en cas d'arrêt du moteur. Augmente la durée de vie du hacheur.
- On choisit « Forward PWM, fast decay », donc l'entrée **AIN2** sera branchée sur la masse et le moteur sur **AOUT1** et **AOUT2**. (voir diapo suivante)

Les transistors A, B, C, D peuvent être modélisés par des interrupteurs.



V_s est la tension de puissance (9V pour le ventilateur)

— Courant Sens 1

Attention ici, une diode est placée aux bornes du moteur du ventilateur, ce qui impose de respecter les polarités à ses bornes et donc son sens de rotation.

Schéma d'un modèle de pont en H type DRV8833

Table 2. PWM Control of Motor Speed

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

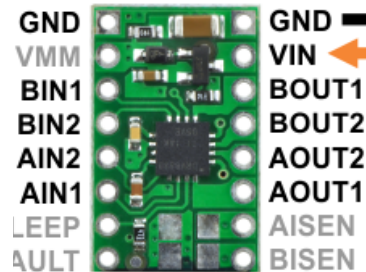
voir aide
branchement
DRV8833 et
page
suivante

Étape 2 : Gestion moteur (2/4)

La sortie PWM branchée sur AIN1 sera fixée à 1000 Hz.
(voir diapo suivante)

Tout câblage s'effectue Pico débranchée !

Appeler le professeur avant toute mise sous tension !

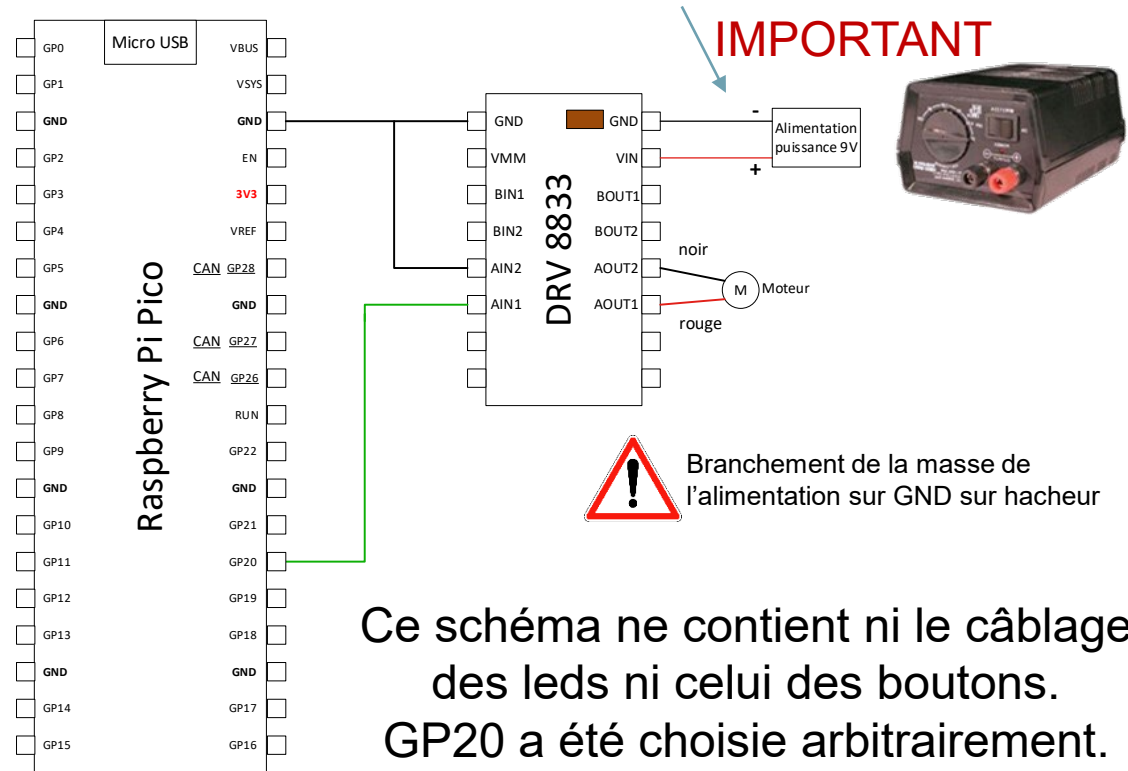


DRV8833

Plus de renseignements dans l'aide (sortie PWM).

L'alimentation dite de « puissance » provient de l'alimentation externe (9V).

Appeler l'enseignant pour vérification avant toute mise en tension



Ce schéma ne contient ni le câblage des leds ni celui des boutons.
GP20 a été choisie arbitrairement.

Ce driver 8833 contient sa propre régulation de tension, inutile de le brancher sur le 3,3V de la carte.

Toutes les masses sont communes, y compris celle de l'alimentation de puissance.

Étape 2 : Gestion moteur (3/4) ; Ecriture du code de test moteur

- Mettre dans Thonny :

```
import machine
from machine import Pin

mli=machine.PWM(Pin(20)) #commande MLI sur GP20
mli.freq(1000) #frequence MLI de 1000 Hz

mli.duty_u16(int(0.5*65535)) #tension moyenne moteur à 50% de Umax
```

- Tester en changeant le rapport cyclique et/ou la fréquence. Qu'observez-vous ?

Remarque, les hacheurs professionnels ont des fréquences autour de 25 kHz voire plus.

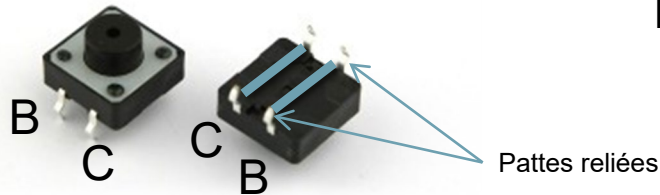
Etape 3 : câblage des boutons et leds.

- Reprenez le câblage du TP feux tricolores précédent :
 - 3 leds câblées et 2 boutons qui étaient nommés usr et trafic et qui deviennent marche et arrêt.
 - Vous pouvez aussi vous référer à l'aide micropython.

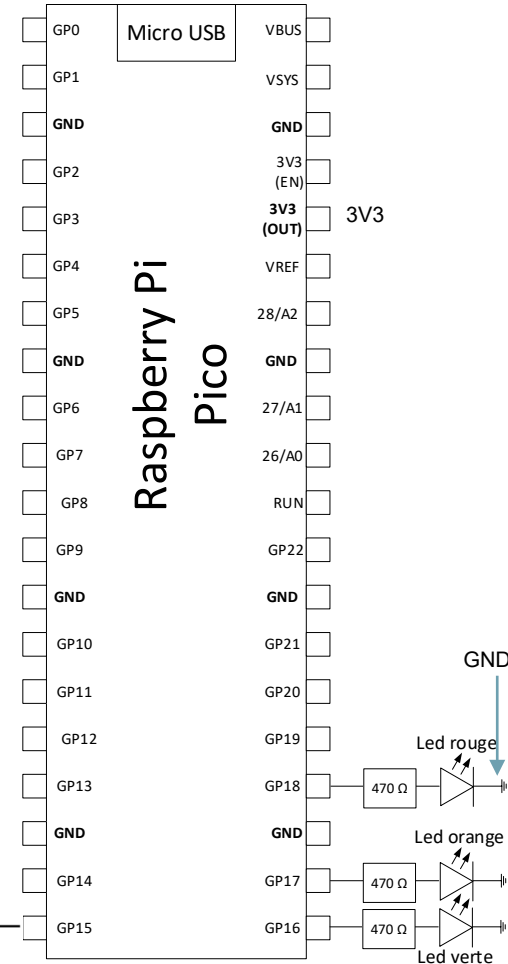
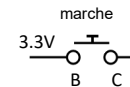
Les boutons poussoirs sont nommés (marche et arrêt) pour la compréhension du programme. **Bien respecter le branchement des pattes B et C.**

Utiliser les colonnes + (rouge) et – (bleue) de la plaque pour placer l'alimentation 3,3V (OUT) et la masse (GND).

Voir aide micropython « entrées externes ».



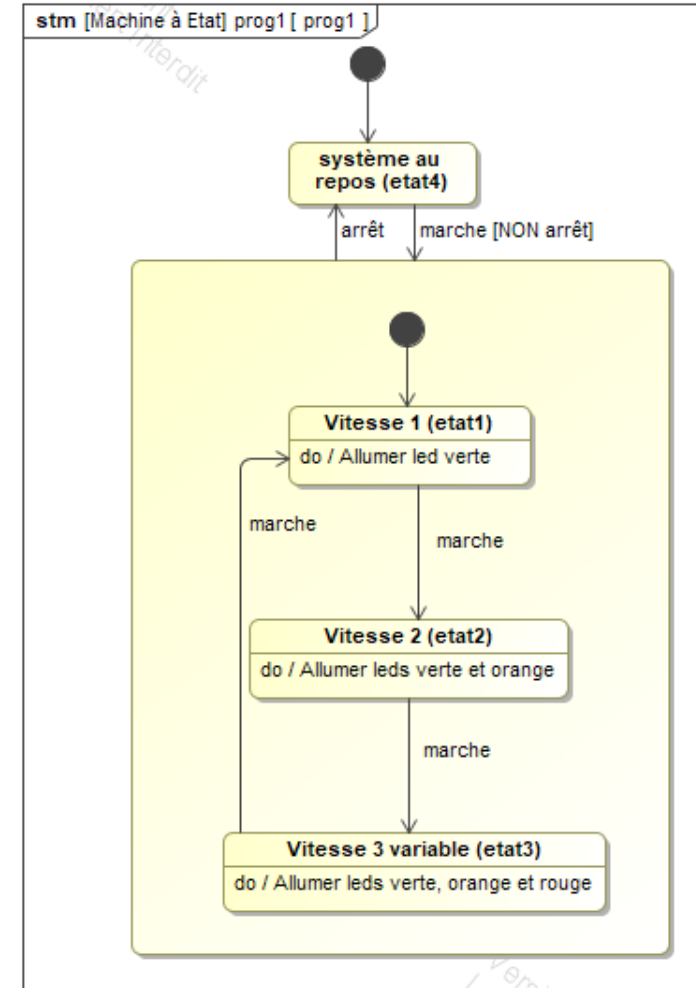
Pour ce type de bouton, il faut placer les fils coté méplat.



Etape 4 : Programmation

Prérequis : faire le tp SED feux tricolores

- Reprenez votre programme TP feux tricolores et modifier-le pour qu'il soit conforme au diagramme suivant.
- Complétez votre programme avec la gestion du bouton arrêt. Testez au fur et à mesure.
- Vous pouvez placer un « print(etat) » dans la boucle « while True:» pour visualiser sous le shell, l'état courant dans votre programme en fonctionnement.



Etape 4 « le rebond » si pb avec les boutons

- Vous pouvez constater que l'action sur le bouton marche peut engendrer un fonctionnement « aléatoire » : ceci est dû au phénomène appelé « rebond électrique » du bouton. En effet un appui sur le bouton peut être perçu comme plusieurs fronts montants par la carte (donc comme plusieurs événements). Pour éviter ce problème, il faut ajouter les lignes suivantes :

```
import time #à ajouter en début de programme
irq_time=time.ticks_ms() #irq_time à déclarer en début de programme (dans l'espace déclaration)
(time.ticks_ms() est une prise de date en milliseconde)
```

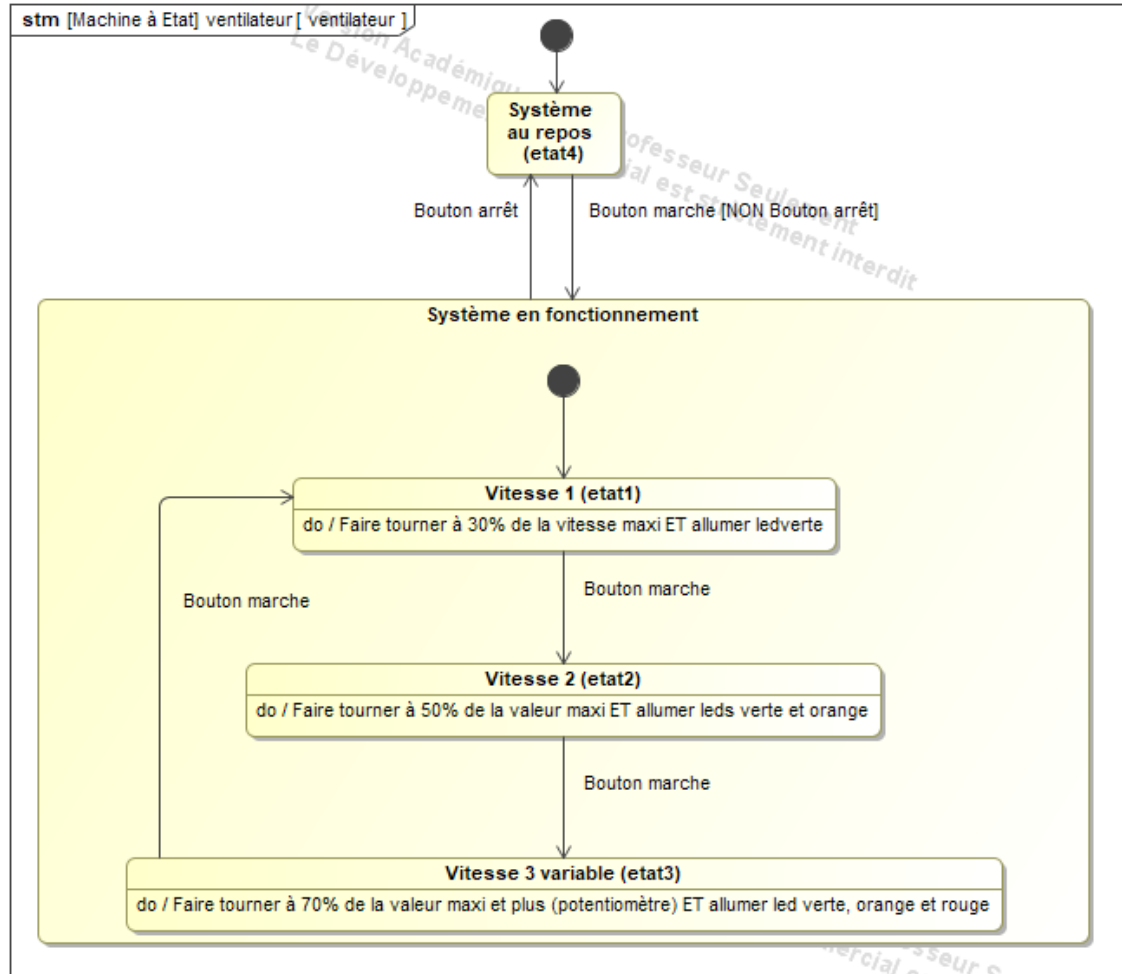
```
global irq_time
if time.ticks_diff(time.ticks_ms(), irq_time)>10:
    irq_time= time.ticks_ms()
...
```

A placer au début de chaque procédure associée à une interruption liée à un bouton (ex : « `def detect_marche(ref) :` »)

- Expliquez ce que réalisent ces lignes et implémenter–les.

Etape 5 : Tests

- A l'exception de la gestion du potentiomètre, complétez votre programme afin d'effectuer ce diagramme d'états.
- N'oubliez pas d'arrêter le moteur à l'état 4.



Etape 6 : Synthèse

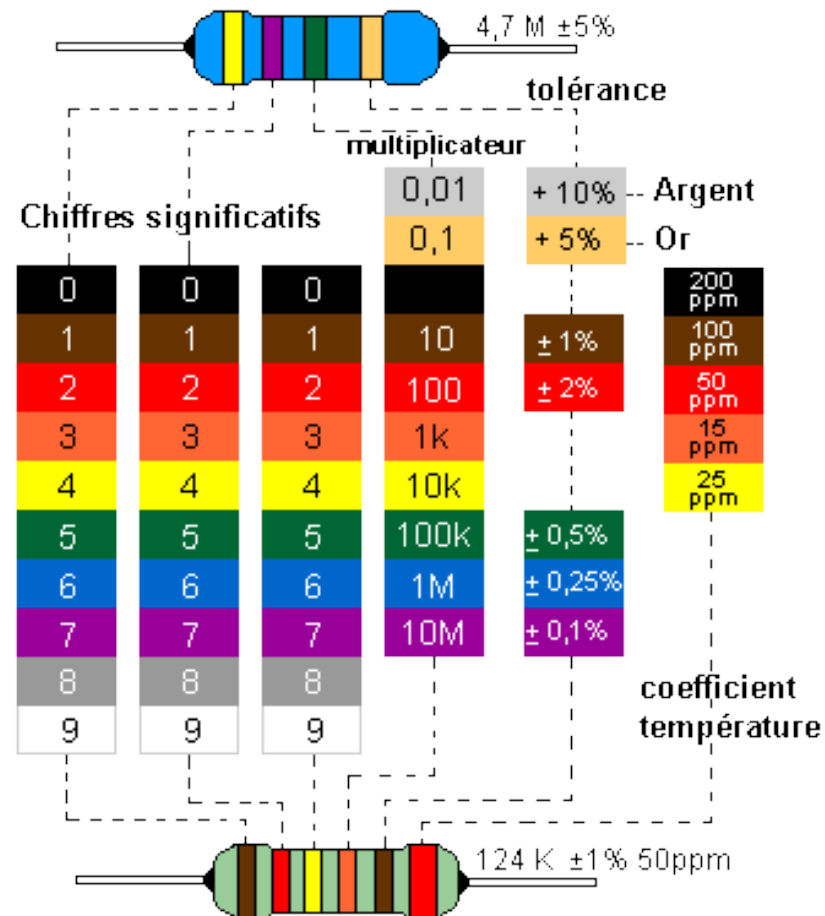
- Tracez les chaines d'information et de puissance pour ce montage en précisant les différents constituants.

Etape 7 : mise en place de l'entrée analogique (potentiomètre) et tests finaux.

- Débranchez la Pico.
- Placez un potentiomètre rotatif sur **une entrée ADC** de votre choix : GP26, 27 ou 28.
- **Voir aide Pico entrée analogique (fiche 4)**
- Les entrées analogiques sont numérisées sur 12 bits, mais le signal est exprimé en entier de 0 à 65535 (16 bits).
- Le potentiomètre est branché sur le 3,3V (OUT) de la carte.
- Vous pouvez afficher sur le shell les valeurs lues par l'entrée analogique avec la fonction print. (voir aide entrée analogique)
- Testez votre branchement avant de réaliser intégralement le diagramme d'états précédent.
- Tests finaux !

Valeurs dipôle résistant (résistor)

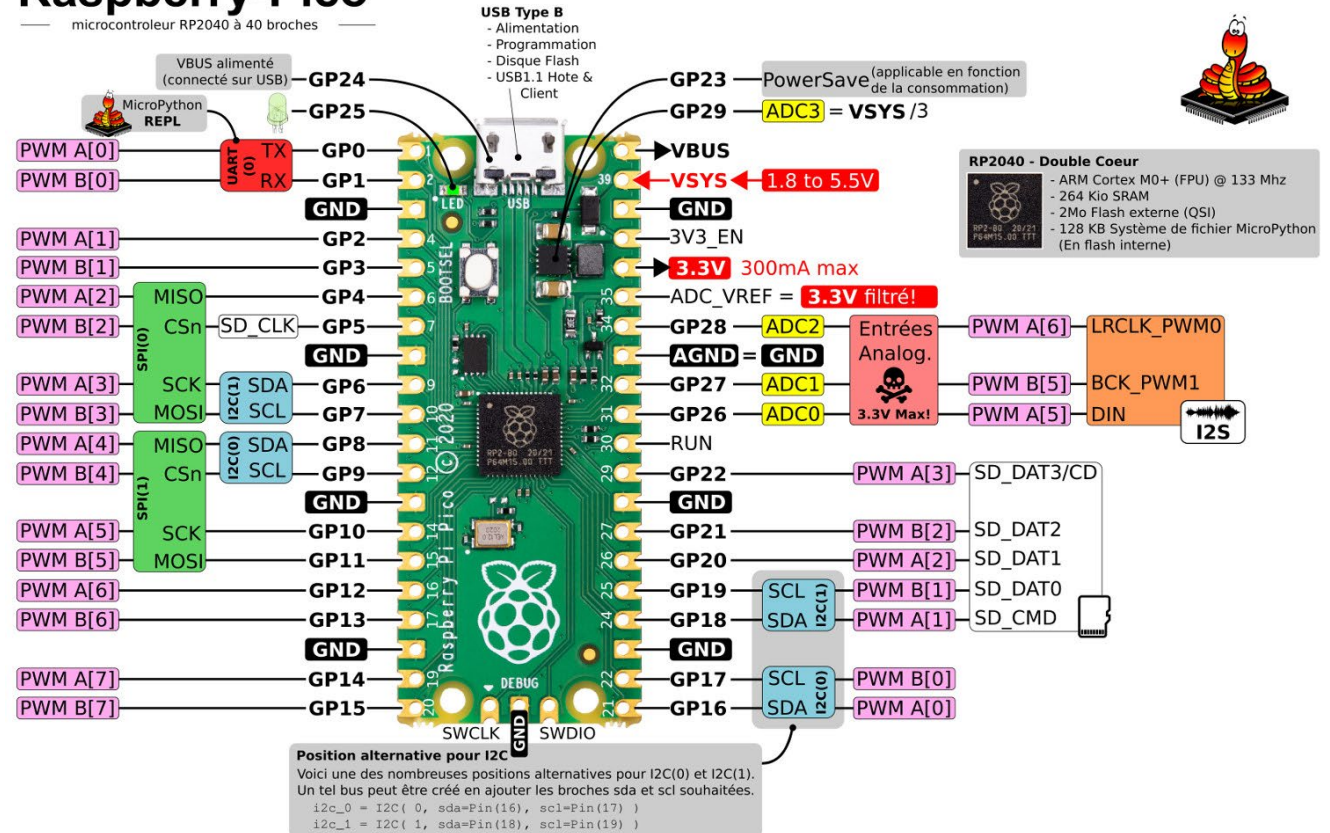
[Retour à la page
précédente](#)



Légendes des connecteurs Pico

Raspberry-Pico

microcontrôleur RP2040 à 40 broches



Connection hacheur DRV8833

1,2A en continu

Observer bien le sens du composant !

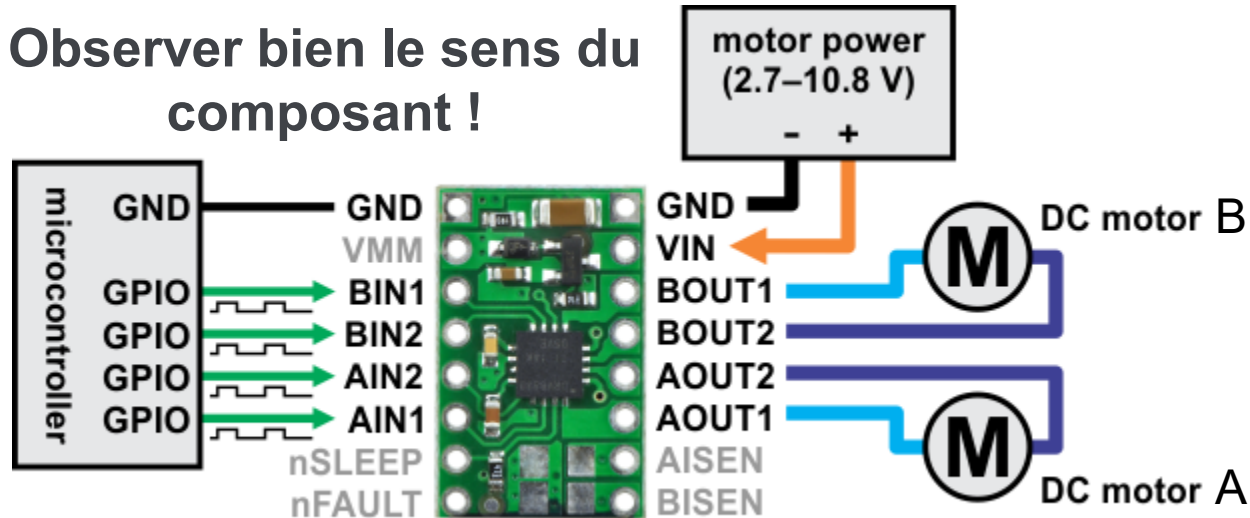


Table 2. PWM Control of Motor Speed

xIN1	xIN2	FUNCTION
PWM	0	Forward PWM, fast decay
1	PWM	Forward PWM, slow decay
0	PWM	Reverse PWM, fast decay
PWM	1	Reverse PWM, slow decay

Ce hacheur possède sa propre régulation de tension. Inutile donc de le relier au 3,3V de la carte Pico.