

# How To : Tester un moteur pas à pas avec A4988 et STM32 Nucleo-L476RG

## 1. Objectif

Expliquer comment connecter et contrôler un moteur pas à pas bipolaire à l'aide d'un driver A4988 et d'un microcontrôleur STM32 (Nucleo-L476RG).

## 2. Matériel nécessaire

- Carte STM32 Nucleo-L476RG
- Driver A4988
- Moteur pas à pas bipolaire (ex: 11HS18-0674S, 0.67 A)
- Alimentation externe pour le moteur (ex: 12V)
- Condensateur 100  $\mu$ F (entre VMOT et GND du driver)

## 3. Branchement

Voici les différentes documentations pour les branchements du driver et du moteur

Ce moteur bipolaire a deux bobines :

Bobine 1 :

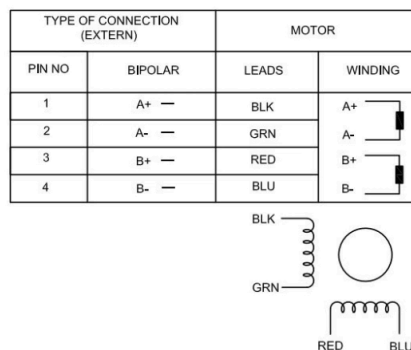
A+ → Fil noir (BLK)

A- → Fil vert (GRN)

Bobine 2 :

B+ → Fil rouge (RED)

B- → Fil bleu (BLU)



Ces bobines doivent être alimentées par le driver avec des signaux électriques appropriés pour faire tourner le moteur.

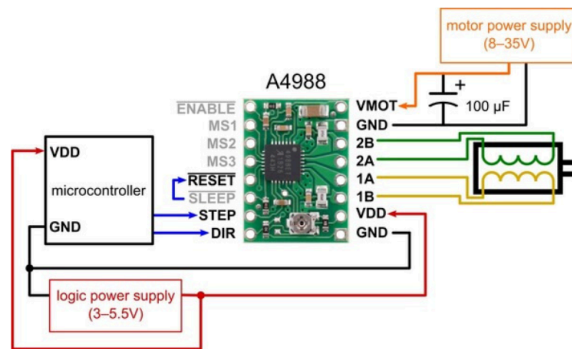


Schéma de câblage minimal pour connecter un microcontrôleur à un support de pilote de moteur pas à pas A4988 (mode pas complet).

Détails des branchements :

STEP : PA0

DIR : PA1

VMOT : +12V (alimentation moteur)

VDD : 3.3V (STM32)

Moteur : 4 fils connectés aux 1A/1B/2A/2B selon datasheet

Ajouter un condensateur 100 µF entre VMOT et GND du driver avant d'alimenter !

#### 4. Réglage du courant sur le driver A4988

Le moteur supporte 0.67 A → on ajuste le potentiomètre du A4988 pour  $V_{ref} \approx 0.54$  V

$$V_{ref} = I_{mot} \times 0.8 = 0.67 \times 0.8 \approx 0.54V$$

Ainsi, on alimente le driver (VMOT et VDD) et avec un multimètre (sonde noire sur GND, sonde rouge sur la vis du potentiomètre) on tourne jusqu'à 0.54 V.

#### 5. Code STM32

```
#define STEP_PIN GPIO_PIN_0
#define DIR_PIN  GPIO_PIN_1
#define STEP_PORT GPIOA

void step_motor(int steps, int delay_us) {
    for (int i = 0; i < steps; i++) {
        HAL_GPIO_WritePin(STEP_PORT, STEP_PIN, GPIO_PIN_SET); // Impulsion HIGH
        HAL_Delay(delay_us / 1000); // Attente (converti en ms si > 1000)
        HAL_GPIO_WritePin(STEP_PORT, STEP_PIN, GPIO_PIN_RESET); // Impulsion LOW
        HAL_Delay(delay_us / 1000);
    }
}
```

## **6. Conclusion**

En suivant cette procédure, le moteur pas à pas est contrôlé de manière fluide avec un timer hardware.

Le bon réglage du courant et la précision des signaux sont essentiels pour éviter les erreurs.

Le condensateur a un rôle important, il permet de stabiliser la tension et protège aussi le driver. Ainsi, il faut toujours l'installer avant d'alimenter un driver de moteur pas à pas !

Pour la génération d'un pwm avec un timer, celle-ci est expliquée dans un autre guide.