

Utiliser un Moteur pas à pas avec l'Arduino

Matériels :

- 1 arduino UNO
- 1 Moteur pas à pas Nema 17
- 1 Driver de moteur A4988
- 1 Potentiometre 10 Kohms
- 1 Breadboard , fils de liaison
- 1 alimentation 12V =

Fonctionnement normal

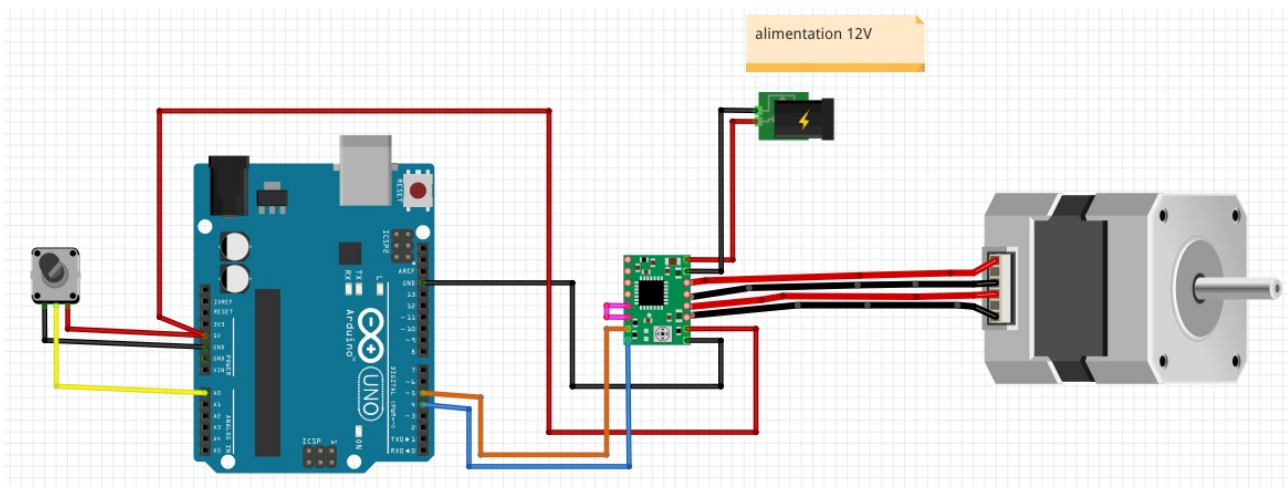
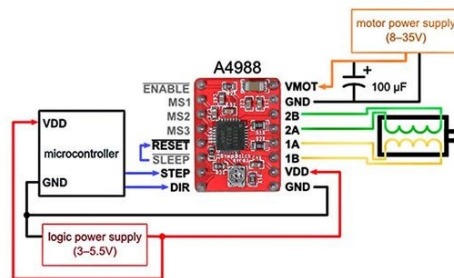
Le but de ce montage est de faire fonctionner le moteur pas à pas dans un sens et dans l'autre en variant le potentiomètre de la valeur maxi à la valeur mini

[doc1](#)

Et en envoyant les commandes avec la liaison série :

- « S » pour Stop on arrête le moteur
- « M » pour mise en marche du moteur
- « A » pour avance du moteur
- « R » pour arrière du moteur

Montage à Réaliser :



Programme PAP_A4988.ino

```
// Moteur pas à pas d'entraînement utilisant le pilote de
//moteur pas à pas A4988

const int DirPin = 4;    // Broche qui définit la direction avant ou arrière
const int StepPin = 5;   // Broche qui envoie PWM pour un tour du moteur
const int SPR = 200;     // Nombre de pas par tour
const int Pot1 = A0 ; ,  // Broche du potentiomètre
int ValPot1 = 0 ;        // valeur du potentiomètre
int ValPrecPot1 = 0 ;    // valeur précédente du potentiomètre

void setup(){
    pinMode(StepPin, OUTPUT);
    pinMode(DirPin, OUTPUT);
    pinMode(pot1, INPUT) ;
}
void loop(){
    ValPot1 = digitalRead(Pot1) ;
    if (ValPot1 == ValPrecPot1){
        Moteur_Arrêt() ;
    }
    if (ValPot1 > ValPrecPot1){
        Moteur_Avant() ;
    }
    if (ValPot1 < ValPrecPot1){
        Moteur_Arriere() ;
    }
    ValPrecPot1 = ValPot1 ;
}

void Moteur_Avant(){

    digitalWrite(DirPin, HIGH);    // définit la direction avant
    // PWM pour envoi de pas sur la broche StepPin
    for(int x = 0; x < SPR; x++){
        digitalWrite(StepPin, HIGH);
        delayMicroseconds(1000);
        digitalWrite(StepPin, LOW);
        delayMicroseconds(1000);
    }
    delay(1000);
}

void Moteur_Arriere(){
    // Définit la direction arrière
    digitalWrite(DirPin, LOW);

    // PWM pour envoi de pas sur la broche StepPin
    for(int x = 0; x < SPR; x++){
        digitalWrite(StepPin, HIGH);
        delayMicroseconds(1000);
        digitalWrite(StepPin, LOW);
        delayMicroseconds(1000);
    }
}
```

```
        delay(1000);  
    }  
    void Moteur_Arret(){  
        digitalWrite(StepPin, LOW);  
        delayMicroseconds(1000);  
    }
```