



Ateliers CREPP

Les moteurs pas-à-pas

Nicolas Le Guerroué

6 février 2022
21 pages

TABLE DES MATIÈRES

1	Préambule	3
I	Les moteurs pas-à-pas	4
2	Présentation	5
2.0.1	Constitution	5
2.0.2	Les types de moteur pas-à-pas	5
2.0.3	Principe	6
3	Commande des moteurs pas-à-pas	11
3.1	Moteurs unipolaires	11
3.2	Moteurs bipolaires	12
3.2.1	Avantages et inconvénients des moteurs pas-à-pas	14
3.2.2	Domaines d'application	14
3.3	Comment distinguer les différents types de moteurs ?	14
4	Exemples	15
4.1	Mise en pratique avec Arduino	15
4.1.1	Liste du matériel	16
4.1.2	Branchements	16
4.1.3	Code Arduino	17
4.2	Mise en pratique avec ESP8266	18
4.2.1	Liste du matériel	18
4.2.2	Branchements	19
4.2.3	Code ESP12	20

SECTION 1

PRÉAMBULE

- ▶ Document réalisé en L^AT_EX par Nicolas Le Guerroué pour le Club de Robotique et d'Electronique Programmable de Ploemeur (CREPP)
- ▶ Permission vous est donnée de copier, distribuer et/ou modifier ce document sous quelque forme et de quelque manière que ce soit.
- ▶ Version du 6 février 2022
- ▶ Taille de police : 11pt
- ☎ 06.20.88.75.12
- ✉ nicolasleguerroue@gmail.com
- ▶ **Dans la mesure du possible, évitez d'imprimer ce document si ce n'est pas nécessaire. Il est optimisé pour une visualisation sur un ordinateur et contient beaucoup d'images.**

Versions

octobre 2021	Fusion des supports d'ateliers
novembre 2021	Ajout de l'atelier sur les servomoteurs
décembre 2021	Ajout de l'atelier sur les moteurs pas-à-pas
janvier 2022	Ajout de l'annexe pour l'installation des bibliothèques ESP8266
février 2022	Ajout de l'annexe pour le serveur Web ESP8266 NodeMCU

Première partie

Les moteurs pas-à-pas



Théorie sur les moteurs pas-à-pas et applications pratiques avec Arduino et ESP8266

SECTION 2

PRÉSENTATION

Les moteur pas-à-pas sont utilisés lorsqu'on souhaite un asservissement en position d'un axe de rotation avec une précision inégalée par les servomoteurs

Constitution

Les moteur pas-à-pas sont constitués de :

- ▶ Plusieurs bobines (un pôle forme une paire de bobines)
- ▶ Un aimant qui sert d'axe de rotation

Les types de moteur pas-à-pas

- ▶ moteur pas-à-pas à phase bipolaire

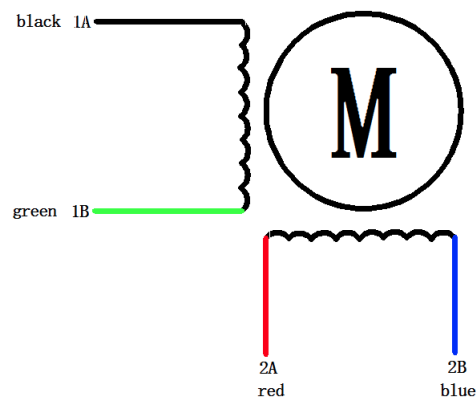


FIGURE 2.1 – moteur pas-à-pas bipolaire

- ▶ moteur pas-à-pas à phases unipolaires

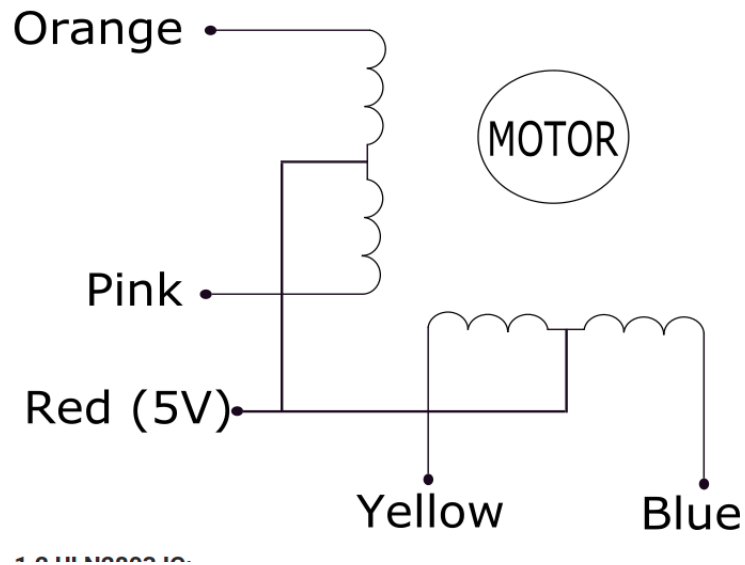


FIGURE 2.2 – moteur pas-à-pas unipolaires

- moteur pas-à-pas à reluctance variable (non abordés ici)

Principe

En faisant varier le champ électromagnétique des différentes phases, on peut faire varier la position angulaire de l'aimant.

Exemples avec phases bipolaires

En alimentant une paire de phases avec une tension positif, l'aimant se place dans l'alignement du champ électromagnétique formé par la paire de phase.

En alimentant la paire de phase avec une tension négative, l'aimant se place dans le sens contraire.

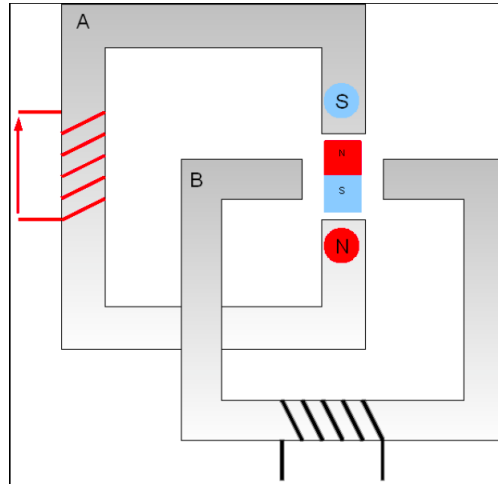


FIGURE 2.3 – Pas 1

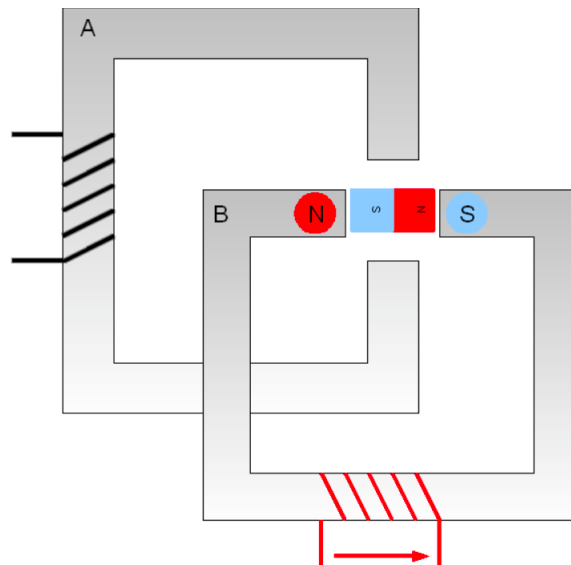


FIGURE 2.4 – Pas 2

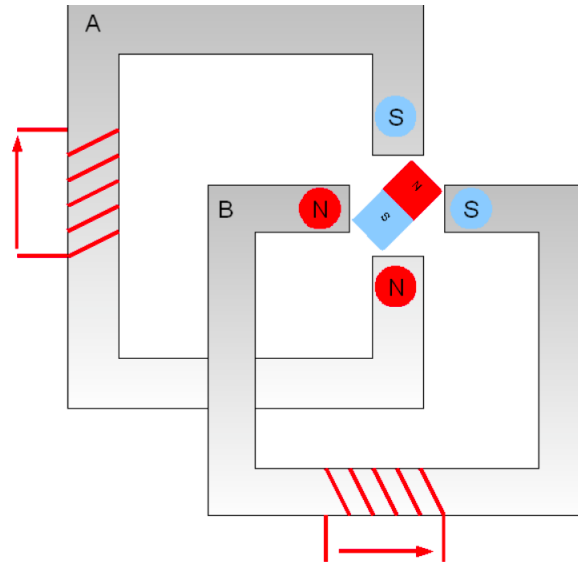


FIGURE 2.5 – Pas 3

Exemples avec phases unipolaires

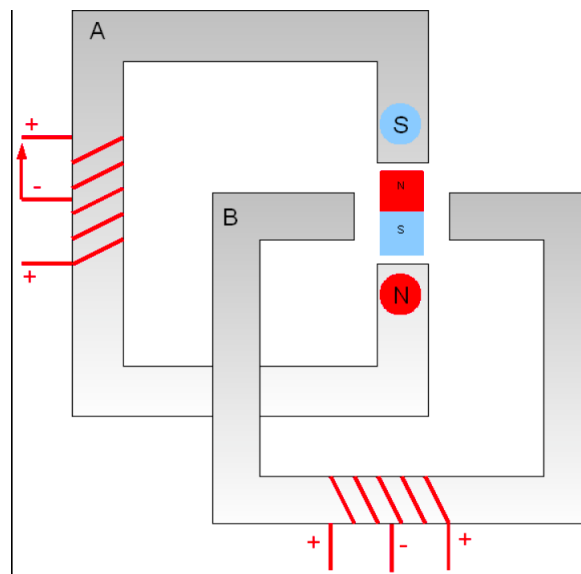


FIGURE 2.6 – Pas 1

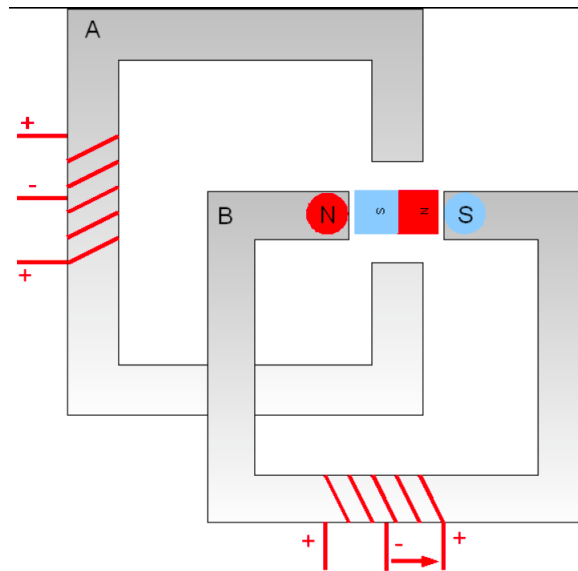


FIGURE 2.7 – Pas 2

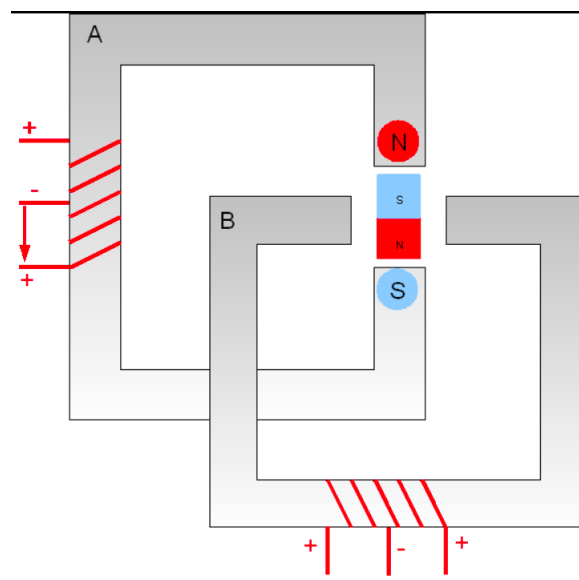


FIGURE 2.8 – Pas 3

Les moteurs possèdent plus de phases car un débattement de 45° est vite limité.

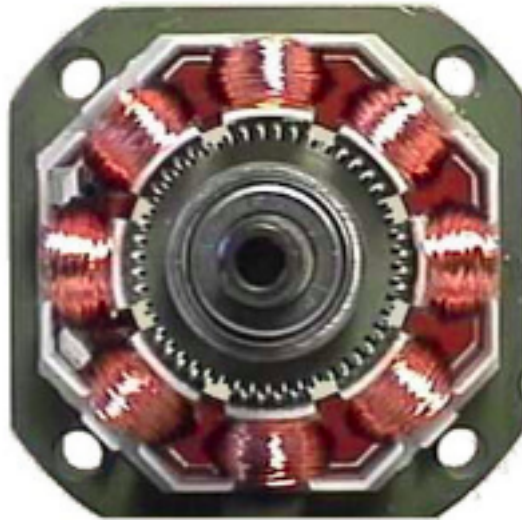


FIGURE 2.9 – Un intérieur de moteur

Les moteur pas-à-pas unipolaires présentent l'avantage de faire circuler un courant positif dans le circuit de commande. Ils sont donc plus simples à mettre en oeuvre mais ils nécessitent plus de bobinage.

SECTION 3

COMMANDE DES MOTEURS PAS-À-PAS

Moteurs unipolaires

Les moteur pas-à-pas unipolaires ont besoins d'être contrôlés via un circuit adaptés, le plus connu est le REF ULN2803



FIGURE 3.1 – Driver ULN2803

Pour les moteurs unipolaires, il faut mettre une des phases à la masse pour faire circuler le courant dans la phase¹

On constate sur la figure suivante un montage Darlington : deux transistors NPN forme un seul transistor dont le coefficient β est le produit de chacun des coefficients β de chaque transistor.

Cela permet de contrôler des charges avec très peu de courant de commande.²

-
1. Due au câble relié à l'alimentation positive du moteur
 2. Se référer à la partie **Circuits de puissance**

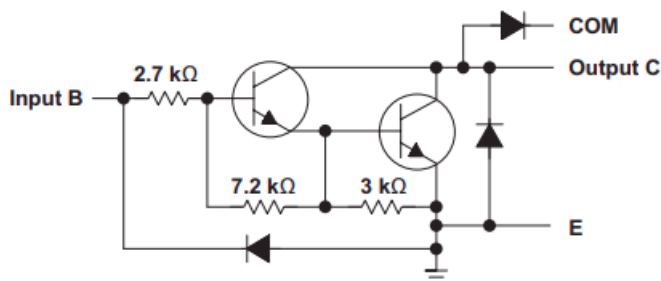


FIGURE 3.2 – Contrôle d'une phase

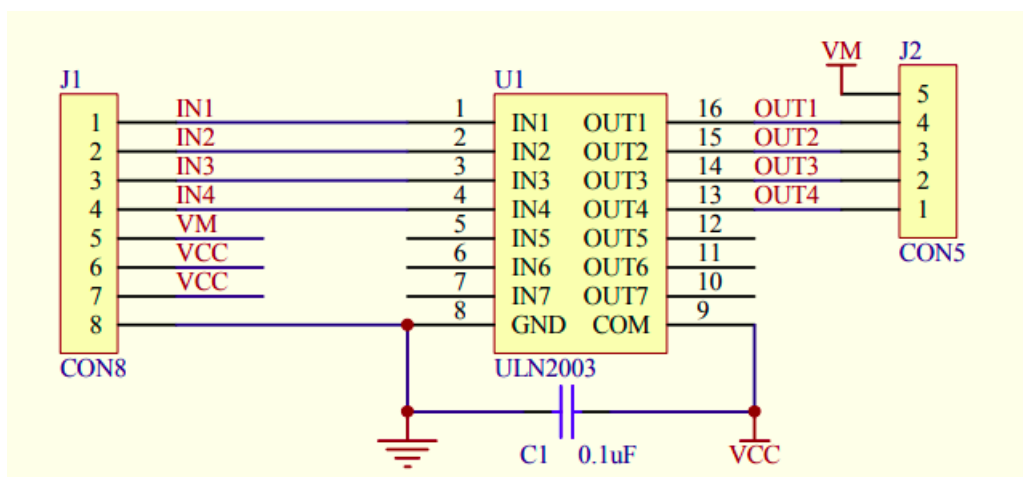


FIGURE 3.3 – Driver de controle

Moteurs bipolaires

On a vu qu'il fallait inverser la tension de commande au borne des bobines. Pour cela on peut utiliser le montage en pont en H.

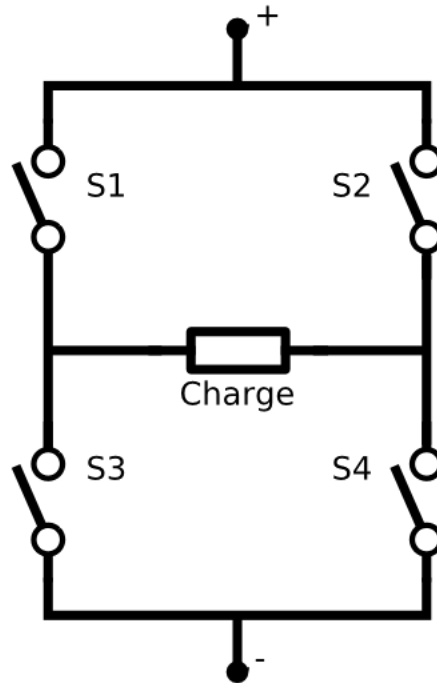


FIGURE 3.4 – Structure du pont en H

- En activant S1 et S4 (fermeture du circuit), la charge est parcourue par un courant allant de gauche à droite
- En activant S2 et S3 (fermeture du circuit), la charge est parcourue par un courant allant de droite à gauche

Et qui dit inversion de courant dit inversion de tension. Notre objectif est atteint, nous pouvons mettre des tensions positives et négatives aux bornes des phases de nos moteurs.

Ce principe est également utilisé pour contrôler les moteurs à courant continu
On peut utiliser le circuit REF L298

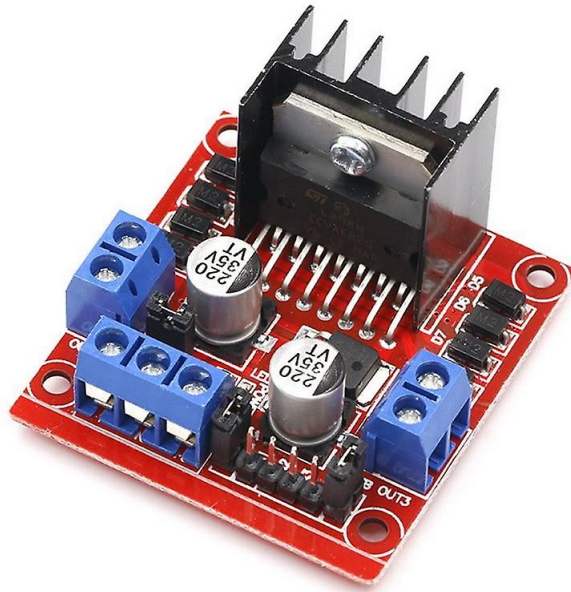


FIGURE 3.5 – Un pont en H intégré

Avantages et inconvénients des moteurs pas-à-pas

- Très grande précision en boucle ouverte³
- Couple élevé en bas régime
- Plus lent qu'un servomoteur
- Complexité de mise en oeuvre

Domaines d'application

- Imprimantes
- Machines CNC

Comment distinguer les différents types de moteurs ?

- 2 fils = moteur à courant continue
- 3 fils = servomoteurs
- 4 fils = moteur pas-à-pas bipolaire
- 5 fils = moteur pas-à-pas unipolaire

3. Contrôle sans asservissement, contrairement aux servomoteurs

SECTION 4

EXEMPLES

Mise en pratique avec Arduino

Nous utiliserons un moteur pas-à-pas **28BYJ-48** de type unipolaire.

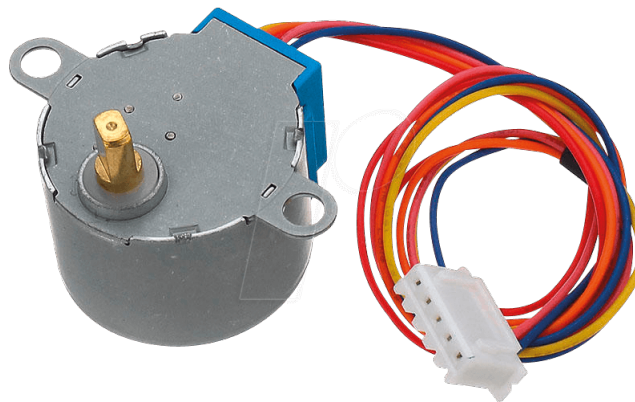


FIGURE 4.1 – Le moteur 28BYJ-48

Les caractéristiques sont les suivantes :

- Nombre de pas : 2048
- Tension d'alimentation : 5V

Pour augmenter le nombre de pas, on ajoute un train d'engrenage.

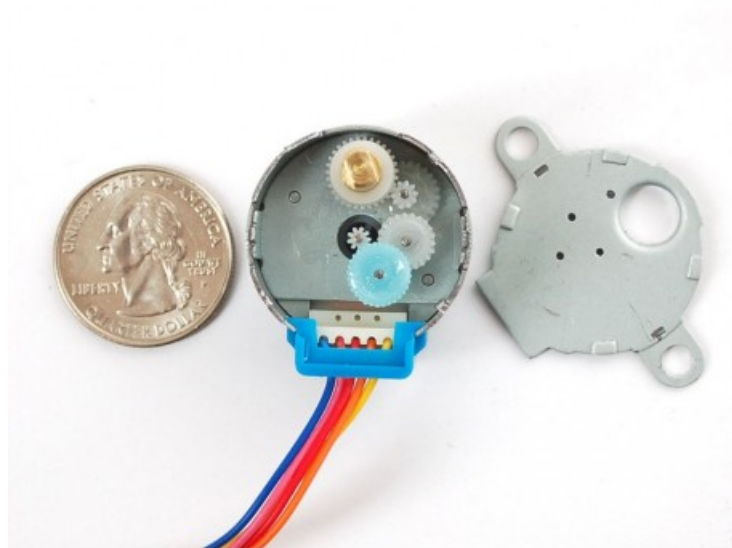


FIGURE 4.2 – Une augmentation du nombre de pas

Liste du matériel

- Carte Arduino Uno
- Driver ULN2803
- Moteur pas-à-pas 28BYJ-48 ou équivalent
- Câbles

Branchements

Nous utiliserons les broches 8, 9, 10 et 11 et l'alimentation 5V du moteur sera fournie par la broche +5V de l'Arduino

- D1 sur IN1
- D2 sur IN3
- D5 sur IN2
- D6 sur IN4
- Vin sur Vcc

- Gnd sur Gnd

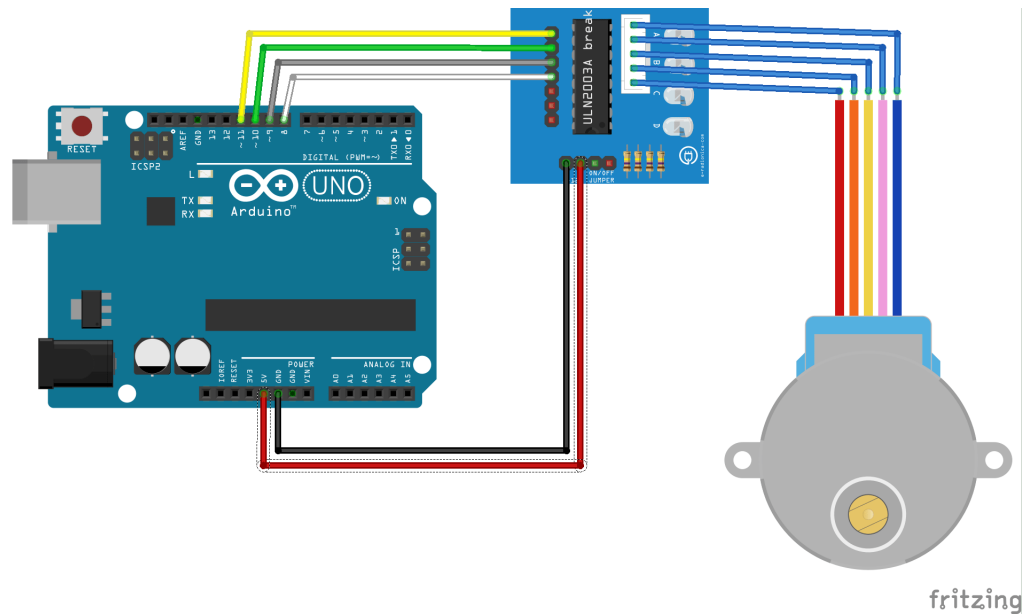


FIGURE 4.3 – Schéma Arduino

Code Arduino

Ce code fait tourner le moteur d'un tour, attend 2 secondes puis fait un tour dans l'autre sens avec un délai de 2s.

```
#include <Stepper.h> //Inclusion de la bibliothèque Stepper

int nbPas = 2048; //Nombre de pas pour le moteur 28BYJ-48

#define IN1 8 //Broche IN1
#define IN2 9 //Broche IN2
#define IN3 10 //Broche IN3
#define IN4 11 //Broche IN4

Stepper moteur(nbPas, IN1, IN3, IN2, IN4); //Création de l'objet moteur

void setup() {

    moteur.setSpeed(10); //On définit la vitesse à 10 tr/min

} //Fin setup

void loop() {
```

```
moteur.step(nbPas); //On avance de nbPas pas, c'est à dire un tour complet  
(sens horaire)  
delay(2000);        //pause de 2s  
moteur.step(-nbPas); //On avance de -nbPas pas, c'est à dire un tour complet  
(sens anti-horaire)  
delay(2000);        //pause de 2s  
  
} //Fin loop
```

Code minimaliste Arduino

Mise en pratique avec ESP8266

Nous utiliserons le même moteur pas-à-pas 28BYJ-48

Liste du matériel

- Carte ESP8266 NodeMCU (ESP-12)



FIGURE 4.4 – ESP12 NodeMCU

Cette carte fait partie de la famille des ESP8266 et se programme directement avec l'Editeur Arduino. L'installation des bibliothèques pour l'ESP12 est détaillée en annexe du document.

- Driver ULN2803
- Moteur pas-à-pas 28BYJ-48 ou équivalent (moteur pas-à-pas unipolaire)
- Câbles

Branchements

Les numéros des broches sont différents sur les cartes ESP812 (modèle NodeMCU). Voici les équivalences des broches entre le code et l'emplacement physique.

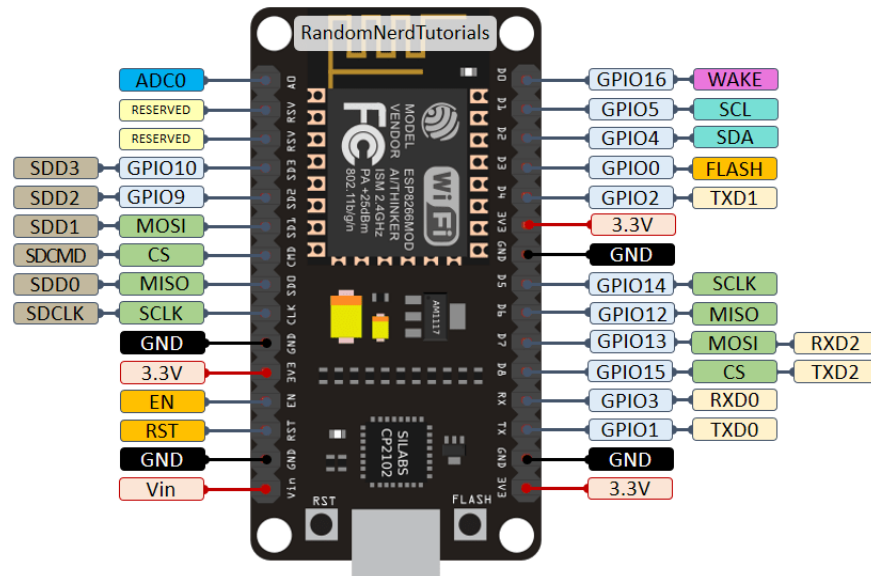


FIGURE 4.5 – Broches ESP12

Nous utiliserons les broches D1, D2, D5 et D6 et l'alimentation 5V du moteur sera fournie par la broche Vin de l'ESP12

- D1 sur IN1
- D2 sur IN3
- D5 sur IN2
- D6 sur IN4
- Vin sur Vcc
- Gnd sur Gnd

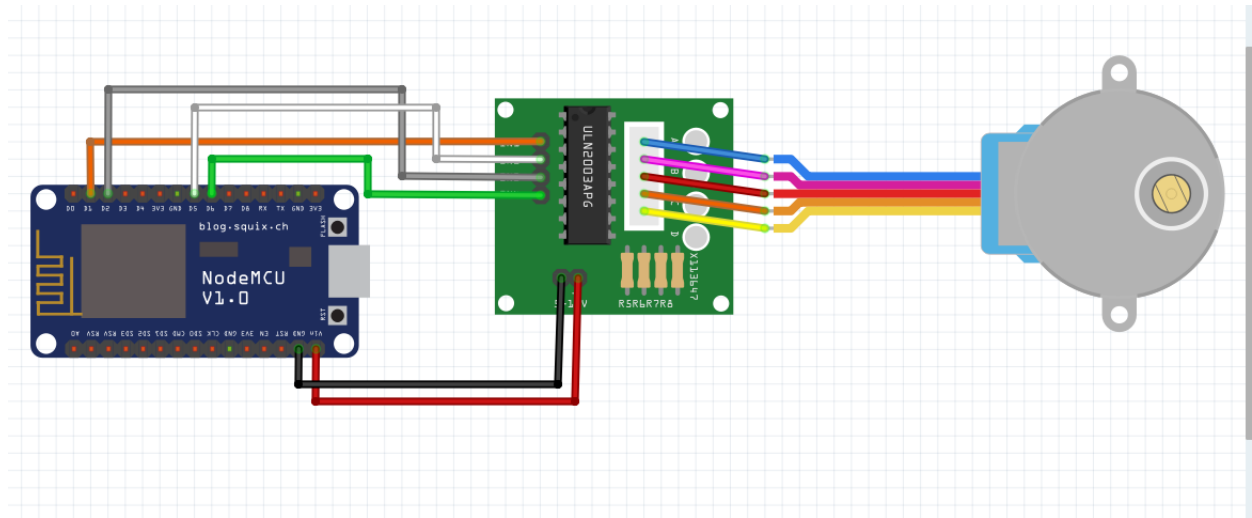


FIGURE 4.6 – Schéma ESP12

Code ESP12

Ce code fait tourner le moteur d'un tour, attend 2 secondes puis fait un tour dans l'autre sens avec un délai de 2s.

```
#include <Stepper.h> //Inclusion de la bibliothèque Stepper

int nbPas = 2048; //Nombre de pas pour le moteur 28BYJ-48

#define IN1 D1 //Broche IN1
#define IN2 D5 //Broche IN2
#define IN3 D2 //Broche IN3
#define IN4 D6 //Broche IN4

Stepper moteur(nbPas, IN1, IN2, IN3, IN4); //Création de l'objet moteur

void setup() {

    moteur.setSpeed(10); //On définit la vitesse à 10 tr/min

} //Fin setup

void loop() {

    moteur.step(nbPas); //On avance de nbPas pas, c'est à dire un tour
    complet (sens horaire)
    delay(2000); //pause de 2s
```

```
moteur.step(-nbPas); //On avance de -nbPas pas, c'est à dire un tour
complet (sens anti-horaire)
delay(2000);          //pause de 2s

} //Fin loop
```

Code minimaliste ESP12