step motor

branchements

### **Connexion avec un Arduino Mega**

#### **Branchement du moteur 28BYJ-48 à ULN2003 :**

* **Fil bleu (A)** : Connecté à **IN1** (broche 8 de l'Arduino Mega)
* **Fil rose (B)** : Connecté à **IN2** (broche 9 de l'Arduino Mega)
* **Fil jaune (C)** : Connecté à **IN3** (broche 10 de l'Arduino Mega)
* **Fil orange (D)** : Connecté à **IN4** (broche 11 de l'Arduino Mega)
* **Fil rouge (VCC)** : Connecté à **5V** (alimentation)
* **GND (du driver ULN2003)** : Connecté à **GND** (sur l'Arduino Mega)\*

2. **Connexion avec un ESP32**

#### **Branchement du moteur 28BYJ-48 à ULN2003 :**

Les fils du moteur sont les mêmes que précédemment :

* **Fil bleu (A)** : Connecté à **IN1** (broche 14 de l'ESP32)
* **Fil rose (B)** : Connecté à **IN2** (broche 12 de l'ESP32)
* **Fil jaune (C)** : Connecté à **IN3** (broche 13 de l'ESP32)
* **Fil orange (D)** : Connecté à **IN4** (broche 15 de l'ESP32)
* **Fil rouge (VCC)** : Connecté à **5V** (alimentation)
* **VCC (du driver ULN2003)** : Connecté à **5V** (sur l'ESP32)
* **GND (du driver ULN2003)** : Connecté à **GND** (sur l'ESP32)

#include <Stepper.h>

// Nombre de pas par tour du moteur 28BYJ-48 (2048 pas pour un tour complet)

#define STEPS 2048

// Définir les broches du moteur pour l'Arduino Mega

Stepper stepper(STEPS, 8, 10, 9, 11);

// Définir la broche du bouton poussoir

const int buttonPin = 6; // Changer la broche du bouton poussoir à 6

int buttonState = 0; // Variable pour stocker l'état du bouton

void setup() {

// Initialiser le bouton comme entrée avec résistance Pull-up interne

pinMode(buttonPin, INPUT\_PULLUP);

// Définir la vitesse du moteur

stepper.setSpeed(15); // Vitesse du moteur en tours par minute

}

void loop() {

// Lire l'état du bouton poussoir

buttonState = digitalRead(buttonPin);

// Si le bouton est appuyé (le bouton pousseur est connecté à GND, donc état LOW)

if (buttonState == LOW) {

stepper.step(STEPS / 4); // Effectuer un mouvement de 90° (1/4 de tour)

delay(7000); // Attendre 7 secondes (porte ouverte)

// Fermer la porte (mouvement de -90°)

stepper.step(-STEPS / 4); // Effectuer un mouvement de -90° (1/4 de tour dans l'autre sens)

}

}

le code permet une ouverture a 90 degrés et une fermeture au bout de 7 secondes

on va ajouter une serrure electronique grace au servo moteur towerpro sg90

#### **ervo moteur SG90 :**

* **Fil de signal** (jaune ou blanc) → broche **D3** de l'Arduino Mega (ou une autre broche PWM libre sur l'ESP32)
* **Fil de puissance (rouge)** → **5V**
* **Fil de masse (noir)** → **GND**

**#include <Stepper.h>**

**#include <Servo.h>**

**// Nombre de pas par tour du moteur 28BYJ-48 (2048 pas pour un tour complet)**

**#define STEPS 2048**

**// Définir les broches du moteur pour l'Arduino Mega**

**Stepper stepper(STEPS, 8, 10, 9, 11);**

**// Définir la broche du bouton poussoir**

**const int buttonPin = 6; // Broche du bouton poussoir**

**int buttonState = 0; // Variable pour stocker l'état du bouton**

**// Définir la broche du servo moteur SG90**

**Servo lockServo;**

**const int servoPin = 3; // Broche PWM pour le servo moteur**

**void setup() {**

**// Initialiser le bouton comme entrée avec résistance Pull-up interne**

**pinMode(buttonPin, INPUT\_PULLUP);**

**// Initialiser le servo moteur**

**lockServo.attach(servoPin); // Connecter le servo à la broche PWM**

**// Initialiser la position du servo pour verrouiller la serrure (0°)**

**lockServo.write(0);**

**// Définir la vitesse du moteur**

**stepper.setSpeed(15); // Vitesse du moteur en tours par minute**

**}**

**void loop() {**

**// Lire l'état du bouton poussoir**

**buttonState = digitalRead(buttonPin);**

**// Si le bouton est appuyé (le bouton pousseur est connecté à GND, donc état LOW)**

**if (buttonState == LOW) {**

**// Déverrouiller la serrure (tourner le servo à 90°)**

**lockServo.write(90);**

**delay(1000); // Attendre 1 seconde pour s'assurer que le servo est bien positionné**

**// Ouvrir la porte (tourner le moteur de 90°)**

**stepper.step(STEPS / 4); // Effectuer un mouvement de 90° (1/4 de tour)**

**delay(7000); // Attendre 7 secondes (porte ouverte)**

**// Fermer la porte (mouvement de -90°)**

**stepper.step(-STEPS / 4); // Effectuer un mouvement de -90° (1/4 de tour dans l'autre sens)**

**// Verrouiller la serrure (retourner le servo à 0°)**

**lockServo.write(0);**

**}**

**}**

la serrure va s ouvrir avant de laisser la porte s ouvrire a son tour puis se refermer automatiquement apres 7 secondes, une fois la porte fermée, la serrure va se verrouiller automatiquement

le delais des 7 secondes sera remplacé a l avenir par le détecteur de présence qui quand actionné commencera le processus de fermeture

**le bouton sera lui aussi remplacé par le dispositif de sécurité d’entrée (rfid et caméra de reconnaissance faciale)**

on utilisera ici le tower pro micro servo sg90 pour la serrure car plus rapide que le stepper motor 28BYJ-48 5VDC ,beaucoup plus adapté à la conception d une serrure électronique et beaucoup moins adapté à une ouverture de porte car trop rapide (risque de blessures, risque de casse) mais parfait pour refermer un veroux rapidement .