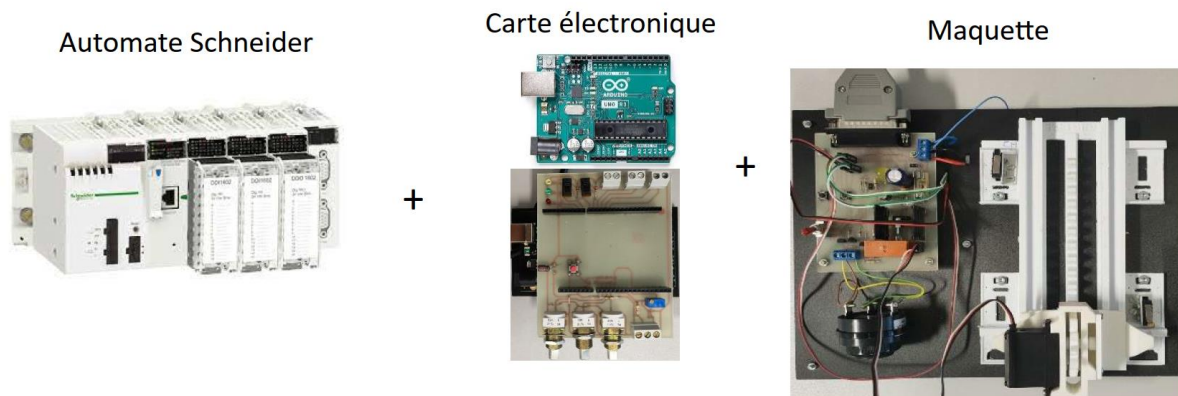


Compte Rendue SAE R2.06 AUTO – GUEYE HAMARI



Ce rapport présente le projet SAE de deuxième année du module R2.06 Automatisation, visant à automatiser une porte de garage. Le projet consiste en la conception, la réalisation et la mise en service d'un système de contrôle automatisé pour une maquette de porte de garage motorisée. En utilisant un automate programmable Schneider Electric et une interface Arduino, nous avons intégré divers composants électroniques et capteurs pour assurer un fonctionnement sécurisé et efficace de la porte. Le travail a été réalisé en binôme et a impliqué une combinaison d'études théoriques, de simulations pratiques et de tests expérimentaux pour répondre aux exigences spécifiées dans le cahier des charges.

A.Fonction Principale

La maquette utilise un servomoteur, qui est un moteur avec une électronique de contrôle pour gérer la vitesse et le sens de rotation. Le câblage inclut une masse (noir), une alimentation (rouge) et un signal de contrôle (vert). La sortie MO_ON active le relais pour l'alimentation. Le signal de contrôle doit avoir une forme PWM pour définir les vitesses et directions : maximale négative, nulle, et maximale positive. L'automate doit générer ces impulsions, ce qui nécessite de vérifier les caractéristiques du matériel disponible.

La valeur de la fréquence du signal de commande à délivrer par l'automate est de 100 Hz :

$$T = 10 \text{ ms} , f = \frac{1}{T} = 100\text{Hz}$$

- les caractéristiques des sorties de l'automate Siemens S7-1214 DC/DC/DC:

C'est un relais statique, 100Khz de Qa 0 à Qa 3 sont les fréquences de commutation ainsi que 20KHz pour Qa4 à Rb1.

Ensuite, les caractéristiques de la carte de sorties TOR de notre automate, DRA1605, ne possède pas de protection(24-48VCC ou 24-240VCA), compte 16 sorties TOR, c'est un relais et enfin, pour ce qui est de la fréquence c'est de 1 à 20 Hz relais en dc.

L'automate a une fréquence trop basse pour contrôler le servomoteur avec le matériel actuel

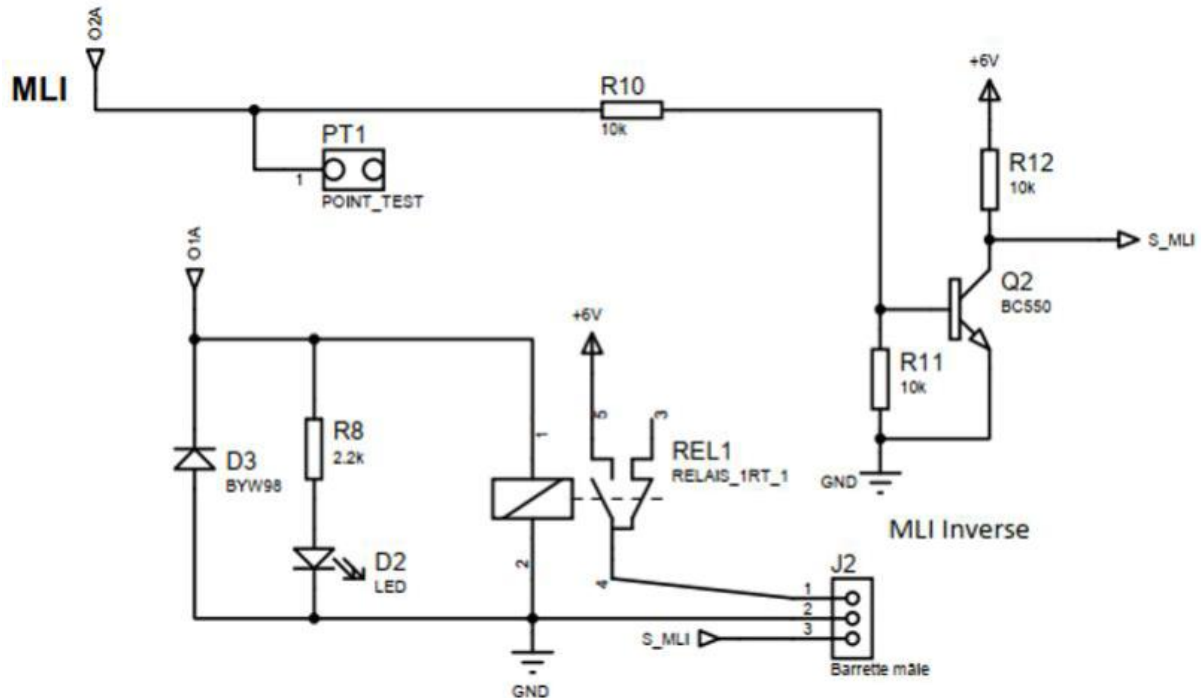
Conclusion sur la Commande du Servomoteur

Les sorties TOR de l'automate, en particulier si elles sont de type relais, ne sont pas adaptées pour générer des signaux PWM à la fréquence nécessaire pour le contrôle précis des servomoteurs. Les relais ne peuvent pas commuter suffisamment vite pour produire des signaux PWM fiables.

Solution Technique Proposée

Puisque les cartes utiles ne sont pas disponibles et sont coûteuses, une solution technique efficace est d'utiliser une sortie analogique de l'automate et de générer le signal de commande impulsif à l'aide d'une carte Arduino. L'Arduino est capable de produire des signaux PWM précis et adaptés aux besoins du servomoteur.

B. Raccordement Interface Arduino/Maquette



- Une sortie de l'interface Arduino devra envoyer les impulsions de commande (MLI) vers la carte de la maquette. Plusieurs sorties sont accessibles par les borniers B1 et B2 .
 - On utilise une des sorties PWM disponibles sur l'Arduino (par exemple D9) et on connecte à l'entrée de commande du servomoteur via le bornier B1.
 - On Relie la masse (GND) de l'Arduino à la masse de la maquette pour assurer une référence commune.
- **On fait le raccordement, sans oublier de relier les masses de chaque côté :**
 - Connection du pin D9 de l'Arduino au bornier B1 pour le signal de commande du servomoteur.
 - Connection du GND de l'Arduino au GND de la maquette.

C. Programmation et Envoi du Signal de Commande Servomoteur

Le bornier B4 de l'interface Arduino est destiné à recevoir le signal analogique provenant de l'automate, qui sera ensuite traité pour générer les impulsions de commande du servomoteur. L'objectif principal de la carte est de transformer un signal analogique en un signal impulsionnel de commande pour contrôler la vitesse du moteur.

Conditions Préalables

Pour que le servomoteur puisse tourner, il faut que :

- L'alimentation du servomoteur soit correctement fournie.
- Le signal de commande PWM soit correctement généré par l'Arduino.

Programmation de l'Arduino :

```
#define POTENTIOMETER_PIN A2 // Broche d'entrée analogique du potentiomètre
#define PWM_PIN 9 // Broche de sortie PWM vers le servomoteur

void setup() {

  pinMode(POTENTIOMETER_PIN, INPUT);

  pinMode(PWM_PIN, OUTPUT); }

void loop() {

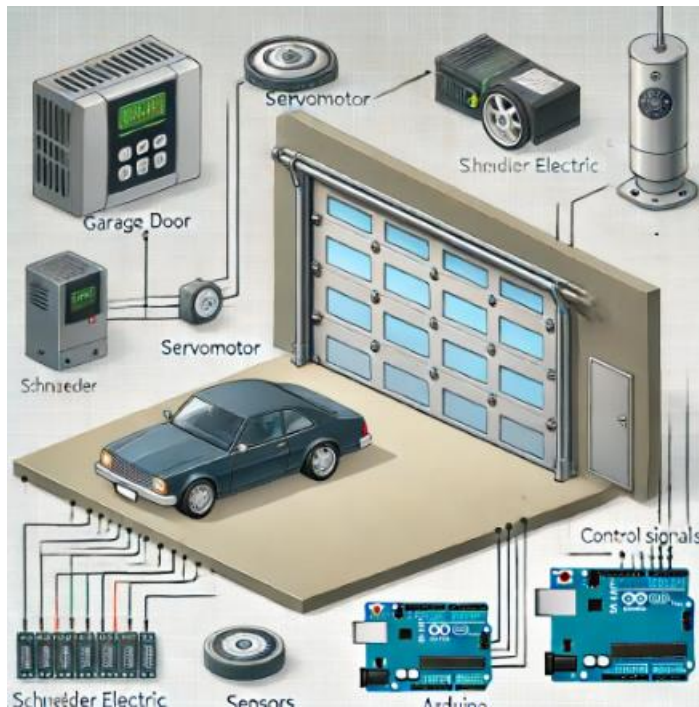
  // Lecture de la valeur analogique du potentiomètre

  int potValue = analogRead(POTENTIOMETER_PIN);           // Conversion de la valeur
  du potentiomètre (0-1023) en largeur d'impulsion (500-2000 microsecondes)

  int pulseWidth = map(potValue, 0, 1023, 500, 2000);      // Génération du signal
  PWM analogWrite(PWM_PIN, pulseWidth / 4);               // Diviser par 4 car analogWrite prend
  des valeurs de 0 à 255 // Attendre 5 ms avant de refaire le cycle

  delay(5); }
```

III. Liaison Automate-Interface Arduino



Il est crucial de s'assurer de la compatibilité des signaux entre l'automate Schneider et la carte Arduino.

Caractéristiques des Signaux

- **Référence de la carte analogique Schneider** : À extraire des documents techniques spécifiques fournis.
- **Tension maximale envoyée par la carte analogique Schneider** : Typiquement 0-10V.
- **Tension maximale acceptée par les entrées analogiques Arduino** : 0-5V.

Conclusion sur la Liaison Carte Analogique/Arduino

Le signal analogique devra être envoyé sur le bornier B4 et traité avant d'arriver sur les entrées A0 ou A4 de l'Arduino. Une mise à l'échelle est nécessaire si la tension maximale de la carte Schneider dépasse 5V.

Conditions de Conduction des Diodes

- **Diode D4 devient passante** : Lorsque la tension d'entrée A0 de l'Arduino dépasse la tension seuil de la diode D4, généralement autour de 0.7V pour une diode standard en silicium.

- **Diode D5 devient passante** : Lorsque la tension d'entrée A0 de l'Arduino dépasse la tension seuil de la diode D5.

Conséquences et Rôle du Montage

- **Ce qui se passe alors** : La diode D4 ou D5 protège l'Arduino contre les surtensions en devenant passante lorsque la tension dépasse le seuil.
- **Rôle du montage** : Protéger l'entrée analogique de l'Arduino contre les surtensions.

Montage Associant R8 et RV4

- **Nom typique** : Diviseur de tension.
- **Expression de la tension reçue en A4 par l'Arduino en fonction de celle arrivant en B4.A4** : $VA4 = VB4 \cdot A4 \cdot \frac{RV4}{R8+RV4}$
- **Utilité de ce montage** : Réduire la tension du signal analogique à une plage acceptable par l'Arduino.

Et enfin, pour la partie grafcet un document du projet en pdf effectué sur unity pro vous est envoyé avec toute les spéciations et paramètre inclus.

Conclusion du Projet SAE : Automatisation d'une Porte de Garage

Le projet SAE a consisté à automatiser une porte de garage en utilisant un automate Schneider Electric et une interface Arduino pour générer les signaux de commande nécessaires au contrôle du servomoteur. L'objectif principal était de concevoir et réaliser un système de contrôle automatisé pour une maquette de porte de garage, permettant un contrôle précis de la porte via l'automate et l'Arduino.

Les étapes ont inclus la définition des besoins, la conception du schéma de câblage, la sélection des composants, le raccordement et la configuration des équipements, ainsi que le développement des programmes nécessaires. Les tests et validations ont assuré le bon fonctionnement du système.



Le projet a permis de réaliser un fonctionnement précis et contrôlé de la porte de garage, démontrant une intégration efficace des technologies et une amélioration des compétences en programmation et en gestion des systèmes automatisés. En conclusion, ce projet a été une application réussie des concepts théoriques à une

réalisation pratique, enrichissant les compétences des participants et démontrant la flexibilité et la robustesse des solutions d'automatisation intégrées.