

Université de Toulon

IUT de Toulon

Département Génie Electrique et Informatique Industrielle (GEII)

# SAE

## Cahier des Charges

écrit le 5 janvier 2024

par

Bruno HANNA

*Encadrant universitaire :* Stephane PIGNOL

---



# Table des matières

<b>Introduction</b>	<b>1</b>
1 Objectifs . . . . .	1
2 Contraintes et Spécifications Techniques . . . . .	1
<b>1 Évaluation des Technologies</b>	<b>2</b>
1.1 LabVIEW . . . . .	2
1.2 Node-RED . . . . .	2
1.3 Arduino . . . . .	2
1.4 PlatformIO . . . . .	2
1.5 Rust . . . . .	3
1.6 Tableau de Comparatif . . . . .	3
1.7 Calendrier de conception sur 32 heures de cours . . . . .	3
1.7.1 Introduction au projet et planification (4h) . . . . .	3
1.7.2 Conception préliminaire (4h) . . . . .	3
1.7.3 Développement de l'interface utilisateur (6h) . . . . .	3
1.7.4 Programmation du microcontrôleur (6h) . . . . .	3
1.7.5 Conception du variateur de vitesse (6h) . . . . .	3
1.7.6 Tests et validation (4h) . . . . .	4
1.7.7 Révision et optimisation (2h) . . . . .	4
1.7.8 Tableau récapitulatif . . . . .	4

# Introduction

Le cahier des charges expose la conception et l'implémentation d'un variateur de vitesse pour un moteur à courant continu (MCC). Le projet souligne la maîtrise des compétences techniques essentielles telles que la conception, la vérification et la maintenance de systèmes électroniques avancés. L'objectif est de réaliser un variateur de vitesse commandable à distance par le biais d'une interface web, opérée via un microcontrôleur ESP8266.

## 1 Objectifs

- Confectionner un variateur adapté au moteur Maxon 406757, capable d'atteindre 6700 tr/min et conçu pour une paire de pôles.
- Instaurer un contrôle à distance par une interface web, animée par le microcontrôleur ESP8266.
- Incorporer l'encodeur AMT10E pour un retour précis de la position du moteur, opérant dans une plage de tension et avec une précision spécifiées.
- Configurer un système polyvalent et ergonomique, accessible par une machine dotée d'internet, et pourvu d'une interface utilisateur multilingue développée via LabVIEW, compatible avec les systèmes Windows.

## 2 Contraintes et Spécifications Techniques

La solution envisagée nécessitera de lancer un service web sur le port 80 ainsi qu'un service TCP sur un port spécifié, assurant une connectivité via WiFi pour la gestion à distance. Le moteur Maxon devra opérer à sa tension nominale de 12V avec une puissance de 20W, tandis que le variateur de vitesse offrira une gamme de tension de 0V à 24V avec une précision de 5%.

Le retour d'information sur la position du moteur sera assuré par l'encodeur AMT10E, qui devra présenter une modularité et une précision accrues pour une fiabilité dans la transmission des données.

# Évaluation des Technologies

## 1.1 LabVIEW

LabVIEW, environnement de développement intégré, est conçu pour la gestion d'applications complexes et de grande envergure. L'organisation et la gestion des fichiers sont essentielles, notamment pour des applications impliquant un grand nombre de VIs et de développeurs. Le Project Explorer offre une vue systémique des fichiers nécessaires, facilitant ainsi la gestion des applications LabVIEW.

## 1.2 Node-RED

Outil de programmation visuelle open-source, Node-RED repose sur Node.js et est idéal pour le développement IoT. Facilitant l'intégration de divers dispositifs, API, et services en ligne, Node-RED propose une bibliothèque étendue de nœuds pré-construits, réduisant significativement le temps de développement. Adapté à une variété de scénarios, Node-RED gère également le streaming et le traitement de données en temps réel.

## 1.3 Arduino

Plateforme électronique open-source, Arduino est conçu pour une approche économique et accessible au développement de projets utilisant des microcontrôleurs. Idéal pour les projets "réels", Arduino permet une initiation pratique à la programmation et à la construction de projets, offrant un moyen abordable d'acquérir des compétences en informatique.

## 1.4 PlatformIO

Outil de développement professionnel pour les systèmes embarqués, PlatformIO est multi-plateforme, multi-architecture et multi-framework. Fournissant des fonctionnalités supérieures à l'IDE Arduino, PlatformIO facilite la création et le débogage de code avec des outils comme IntelliSense et un débogueur intégré.

## 1.5 Rust

Langage de programmation conçu pour la sécurité et la haute performance, Rust est adapté aux systèmes embarqués. Offrant une analyse statique puissante et une gestion flexible de la mémoire, Rust garantit une utilisation sûre de la mémoire lors de la compilation et est soutenu par une communauté active.

## 1.6 Tableau de Comparatif

Critère	LabVIEW	Node-RED	Arduino	PlatformIO	Rust
Difficulté	++	+	++	++++	+++++
Flexibilité	++	+++	+++	+++++	+++
Coût	Payant	Gratuit	Gratuit	Gratuit	Gratuit
Intégration	Modules	Native	Native	Native	Mauvaise
Idéal pour	Industriel	IoT rapide	Éducatif	Performance	Sécurité

## 1.7 Calendrier de conception sur 32 heures de cours

### 1.7.1 Introduction au projet et planification (4h)

- Présentation des objectifs et du cahier des charges.
- Organisation des équipes et répartition des tâches.

### 1.7.2 Conception préliminaire (4h)

- Étude des composants (MCC, ESP8266, variateur de vitesse).
- Sélection des outils de développement (LabVIEW, Node-RED, etc.).

### 1.7.3 Développement de l'interface utilisateur (6h)

- Création de l'interface avec LabVIEW.
- Intégration de l'interface avec le serveur web ESP8266.

### 1.7.4 Programmation du microcontrôleur (6h)

- Codage des fonctionnalités de base avec Arduino/PlatformIO.
- Tests de communication avec l'ESP8266.

### 1.7.5 Conception du variateur de vitesse (6h)

- Assemblage du circuit électronique.

- Intégration et tests avec le moteur MCC.

### 1.7.6 Tests et validation (4h)

- Tests de contrôle à distance.
- Validation de la réponse du moteur et de l'encodeur.

### 1.7.7 Révision et optimisation (2h)

- Analyse des performances.
- Optimisation du code et des réglages du système.

### 1.7.8 Tableau récapitulatif

Étape	Description	Durée (heures)
1	Introduction et planification	4
2	Conception préliminaire	4
3	Développement de l'interface utilisateur	6
4	Programmation du microcontrôleur	6
5	Conception du variateur de vitesse	6
6	Tests et validation	4
7	Révision et optimisation	2

Table 1.1 – Répartition des tâches de conception