

HEIG-VD

Institut d’automatisation industrielle

Route de Cheseaux 1

1401 Yverdon-les-Bains

|  |
| --- |
| Travail de Bachelor - Spécification |
| **Contrôleur Brushless Ultra Compact** |

Rédigé par  
 HEIG-VD :  
 ***Joan Bommottet***

Distribution  
 HEIG-VD :  
 ***Yves Chevalier***

**Historique du document**

|  |  |
| --- | --- |
| **Date** | **Changements** |
| 02/03/2021 | Version initiale |
| 09/03/2021 | Mise en forme du document  Capteur SIN/COS & Hall analogiques rendus optionnels  Protocole CANopen simplifié en CAN sans application de la norme CiA402  Ajout d'un protocole UART  Ajout d'un interface PC  Ajout des contraintes de performances |

Table des matières

[1. Introduction 3](#_Toc65664402)

[1.1. Abréviations 3](#_Toc65664403)

[1.2. Contexte 3](#_Toc65664404)

[1.3. But du projet 3](#_Toc65664405)

[1.4. Approche suivie 3](#_Toc65664406)

[1.5. Structure du document 3](#_Toc65664407)

[2. Fonctionnalités 4](#_Toc65664408)

[2.1. Matériel compatible 4](#_Toc65664409)

[2.1.1. Moteur brushless 4](#_Toc65664410)

[2.1.2. Capteurs 5](#_Toc65664411)

[2.1.3. Alimentations 5](#_Toc65664412)

[2.2. Mesures internes 6](#_Toc65664413)

[2.2.1. Mesure de courant 6](#_Toc65664414)

[2.2.2. Mesure de tension 6](#_Toc65664415)

[2.3. Communications externes 7](#_Toc65664416)

[2.3.1. Messages transmis 7](#_Toc65664417)

[2.3.2. Protocoles de communications 7](#_Toc65664418)

[2.4. Interface PC 8](#_Toc65664419)

[2.4.1. Généralités 8](#_Toc65664420)

[2.4.2. Fonctionnalités en écriture 8](#_Toc65664421)

[2.4.3. Fonctionnalités en lecture 8](#_Toc65664422)

[Contraintes 9](#_Toc65664423)

[2.5. Contraintes Hardware 9](#_Toc65664424)

[2.6. Contraintes Software 9](#_Toc65664425)

[Performances 9](#_Toc65664426)

# Introduction

## Abréviations

***Aucunes abréviations pour le moment.***

## Contexte

Ce projet est réalisé dans le cadre du travail de Bachelor de la HEIG-VD.

Le développement qui en découle se base sur un produit déjà existant appartenant à l’institut IAI de l’école d’ingénieurs.

## But du projet

Ce projet a pour but de développer un contrôleur pour moteurs brushless qui soit à la fois compact et performant. Le but ultérieur est de pouvoir utiliser ce système dans des robots afin qu’ils puissent effectuer diverses tâches motorisées.

De plus, l’entièreté du développement sera déposée sur GitHub, afin qu’il devienne Open Source, dans le but de rendre cette technologie accessible.

## Approche suivie

Pour préparer cette spécification, un processus de dialogue avec mon professeur répondant est mis en place, de manière à s’assurer de la bonne compréhension mutuelle des exigences formulées. Le document de spécification s’insère par ailleurs dans ledit processus.

## Structure du document

Le chapitre 2 décrit les fonctionnalités du système dans sa globalité. Il peut s’agir d’une fonctionnalité déjà existante sur la carte électronique actuelle qu’il faut maintenir, d’une amélioration ou encore d’une nouvelle fonctionnalité.

Le chapitre 3 présente les contraintes liées aux fonctionnalités.

Le chapitre 4 détaille les exigences de performances.

# Fonctionnalités

Ce chapitre décrit les fonctionnalités du système dans sa globalité. Il peut s’agir d’une fonctionnalité déjà existante sur la carte électronique actuelle qu’il faut maintenir, d’une amélioration ou encore d’une nouvelle fonctionnalité.

## Matériel compatible

Cette section décrit les éléments extérieurs au système et leurs interactions.

### Moteur brushless

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Exigence | min | nom | max | Unité |
| 10.10.10 | Nombre de moteurs pouvant être piloté par le contrôleur | 1 | - | 2 | - |
| 10.10.20 | Puissance du moteur | 20 | - | 150 | W |
| 10.10.30 | Tension d'alimentation du moteur | 10 | 24 | 48 | V |
| 10.10.40 | Courant nominal du moteur | - | 6 | - | A |
| 10.10.41 | Amplitude du pic de courant d'un moteur supporté par le contrôleur | - | - | 15 | A |
| 10.10.42 | Durée du pic de courant | - | - | 5 | s |

#### Asservissement

|  |  |
| --- | --- |
| ID | Exigence |
| 10.10.50 | Le contrôleur doit permettre d’asservir le moteur avec une commande de couple. |
| 10.10.60 | Le contrôleur doit permettre d’asservir le moteur avec une commande en vitesse. |
| 10.10.70 | Le contrôleur doit permettre d’asservir le moteur avec une commande de position pour autant que le capteur utilisé soit en mesure d'atteindre la précision souhaitée par la commande. |
| 10.10.80 | Il doit être possible de choisir le type d’asservissement du moteur. |
| 10.10.81 | Il devrait être possible de pouvoir changer de type d’asservissement en cours de mouvement du moteur. Cette clause ne s’applique que pour l’asservissement de couple et de vitesse. |

### Capteurs

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Exigence | Hardware | Software |
| 10.20.10 | Le contrôleur doit être conçu de telle manière à pouvoir gérer l’acquisition de la position du rotor du moteur via différentes technologies de capteurs. | | |
| 10.20.20 | Aucun capteur (technologie "Sensorless") | x | Optionnel |
| 10.20.30 | 3 sondes de Hall digitales | x | x |
| 10.20.40 | 3 sondes de Hall analogiques | x | Optionnel |
| 10.20.50 | Codeur incrémental digital (Quadrature) | x | x |
| 10.20.60 | Codeur incrémental analogique (SIN/COS) | x | Optionnel |
| 10.20.70 | Codeur incrémental (TTL) | x | x |
| 10.20.80 | Codeur absolu analogique | x | Optionnel |

(x = implémenté)

### Alimentations

Lors d'une phase de freinage du moteur, celui-ci restitue de l'énergie. Ce sous-chapitre décrit comment le contrôleur doit gérer cette énergie.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Exigence | min | nom | max | Unité |
| 10.30.10 | Augmentation de la tension d'alimentation du moteur (10.10.30) autorisant de restituer l'énergie. | 0% | - | +20% | V |
| 10.30.11 | Si la tension d'alimentation du moteur augmente plus que le seul de la clause 10.30.10, alors le contrôleur doit dissiper cette énergie. | | | | |
| 10.30.12 | La valeur de seuil de la clause 10.30.10 doit être fixée par l'utilisateur. | | | | |
| 10.30.13 | Il doit être possible de paramétrer la valeur de seuil de la clause 10.30.10 sans avoir à reprogrammer le contrôleur. | | | | |
| 10.30.20 | Si le contrôleur est alimenté par une batterie, alors le contrôleur peut bypasser la clause 10.30.10 et directement restituer l'énergie. | | | | |
| 10.30.30 | Il doit être possible d'indiquer le type d'alimentation du contrôleur sans avoir à le reprogrammer. | | | | |

## Mesures internes

Cette section décrit les différentes mesures internes à la carte électronique du contrôleur.

### Mesure de courant

|  |  |
| --- | --- |
| ID | Exigence |
| 10.40.10 | Le contrôleur doit mesurer le courant circulant dans les trois phases du moteur. |

### Mesure de tension

|  |  |
| --- | --- |
| ID | Exigence |
| 10.50.10 | Le contrôleur doit mesurer la tension d’alimentation des trois phases du moteur. |
| 10.50.20 | Le contrôleur doit mesurer sa propre tension d'alimentation. |

## Communications externes

### Messages transmis

|  |  |
| --- | --- |
| ID | Exigence |
| 10.60.10 | Quel que soit le protocole de communication utilisé, il doit être possible de lire la valeur d'un paramètre du contrôleur.  Les paramètres sont (exemples non exhaustif) :   * Les spécifications du moteur nécessaire au bon fonctionnement du contrôleur. * Le type d'asservissement souhaité pour le moteur. * Le type de capteur connecté au contrôleur. * Le type d'alimentation du contrôleur. |
| 10.60.20 | Quel que soit le protocole de communication utilisé, il doit être possible de définir la valeur d'un paramètre du contrôleur (exemples semblables à la clause 10.60.10). |
| 10.60.30 | Quel que soit le protocole de communication utilisé, il doit être possible de demander des informations en live au contrôleur.  Les informations possibles sont les suivantes (exemples non exhaustif) :   * Est-ce que le moteur tourne ou non ? * La position instantanée du moteur. * La vitesse instantanée du moteur. * Le couple instantané du moteur. * Le courant dans les phases du moteur. * L'état du régulateur. |
| 10.60.40 | Quel que soit le protocole de communication utilisé, il doit être possible de définir une valeur de consigne pour le moteur. |
| 10.60.41 | La valeur de la consigne doit être traitée en fonction du type d'asservissement souhaité du moteur. |

### Protocoles de communications

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Exigence | Hardware | Software |
| 10.70.10 | Plusieurs protocoles de communication doivent être mis en place. | | |
| 10.70.11 | Controller Area Network (CAN) | x | x |
| 10.70.12 | Inter-Integrated Circuit (I2C) | x | x |
| 10.70.13 | Serial Peripheral Interface (SPI) | x | Optionnel |
| 10.70.14 | Serial Communications Interface (SCI / UART) | x | x |
| 10.70.20 | L’ensemble des messages décrit par les clauses 10.60.XX doit être gérés dans les protocoles de communication des clauses 10.70.1X. | | |

(x = implémenté)

## Interface PC

### Généralités

|  |  |
| --- | --- |
| ID | Exigence |
| 10.100.10 | Un logiciel PC permettant d'interagir avec le contrôleur doit être programmé. |
| 10.100.20 | Ce programme doit communiquer avec le contrôleur par le biais d'un port série RS232 (UART pour le contrôleur). |

### Fonctionnalités en écriture

|  |  |
| --- | --- |
| ID | Exigence |
| 10.110.10 | Le logiciel doit pouvoir envoyer une commande de consigne au contrôleur. |
| 10.110.20 | Ce programme doit permettre à l'utilisateur de paramétrer le contrôleur. |
| 10.110.30 | Il doit être possible de paramétrer différentes spécifications du moteur.  Voici une liste non exhaustive de paramètres possibles.   * Puissance nominale * Tension nominale * Courant nominal * Nombre de paires de pôles |
| 10.110.40 | Il doit être possible de paramétrer le type d'asservissement du moteur |
| 10.110.50 | Il doit être possible de paramétrer le type de capteur parmi ceux présentés aux clauses 10.20.XX. |
| 10.110.51 | Suivant le capteur choisi, il doit être possible de programmer différents paramètres, dépendant de celui-ci.  Voici une liste non exhaustive des paramètres possibles.   * Résolution * Plage de tension de sortie d'un capteur analogique. |
| 10.110.60 | Il doit être possible de paramétrer le type d'alimentation du contrôleur parmi les types d'alimentations présentés aux clauses 10.30.XX. |
| 10.110.70 | Il doit être possible de régler la fréquence de découpage du pont triphasé. |
| 10.110.80 | Il devrait être possible de régler les paramètres du régulateur. |

### Fonctionnalités en lecture

|  |  |
| --- | --- |
| ID | Exigence |
| 10.120.10 | Il doit être possible de visualiser les paramètres du contrôleur. |
| 10.120.11 | L'ensemble des paramètres décrits par les clauses 10.110.XX doit pouvoir être appliqué à l'affichage des paramètres lus dans la mémoire du contrôleur. |
| 10.120.20 | L'ensemble des informations décrites à la clause 10.60.30 doit pouvoir être visualisé en direct sur des graphiques adaptés. |

# Contraintes

## Contraintes Hardware

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Exigence | min | nom | max | Unité |
| 20.10.10 | Fréquence de découpage du pont triphasé | 20 | - | 100 | kHz |
| 20.10.20 | Longueur de la carte électronique | - | - | 100 | mm |
| 20.10.30 | Largeur de la carte électronique | - | - | 80 | mm |
| 20.10.40 | Le processeur du contrôleur doit être un DSP de la marque Texas Instrument et de la famille TMS320F28x. | | | | |

## Contraintes Software

|  |  |
| --- | --- |
| ID | Exigence |
| 20.20.10 | Le programme du contrôleur doit être écrit dans le langage C ou C++. |
| 20.20.11 | L'interface PC peut être codée dans le langage souhaité. |

# Performances

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Exigence | min | nom | max | Unité |
| 30.10.10 | Fréquence d'échantillonnage de la boucle de régulation. | 1 | - | - | kHz |
| 30.10.20 | Rendement global du contrôleur | 95 | - | - | % |
| 30.10.30 | Température de fonctionnement | -40 | - | 100 | °C |
| 30.10.40 | Le contrôleur doit être conforme aux directives CEM. | | | | |