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 |

Table des matières

[1. Introduction 3](#_Toc65585712)

[1.1. Abréviations 3](#_Toc65585713)

[1.2. Contexte 3](#_Toc65585714)

[1.3. But du projet 3](#_Toc65585715)

[1.4. Approche suivie 3](#_Toc65585716)

[1.5. Structure du document 3](#_Toc65585717)

[2. Fonctionnalités 4](#_Toc65585718)

[2.1. Matériel compatible 4](#_Toc65585719)

[2.1.1. Moteur brushless 4](#_Toc65585720)

[2.1.2. Capteurs 5](#_Toc65585721)

[2.1.3. Alimentations 5](#_Toc65585722)

[2.2. Mesures internes 6](#_Toc65585723)

[2.2.1. Mesure de courant 6](#_Toc65585724)

[2.2.2. Mesure de tension 6](#_Toc65585725)

[2.3. Communications externes 6](#_Toc65585726)

[2.3.1. Messages transmis 6](#_Toc65585727)

[2.3.2. Bus de données CAN 7](#_Toc65585728)

[2.3.3. Bus de données I2C 7](#_Toc65585729)

[2.3.4. Bus de données SPI 7](#_Toc65585730)

[2.3.5. Bus de données UART 7](#_Toc65585731)

[2.4. Interface PC 8](#_Toc65585732)

[2.4.1. Généralités 8](#_Toc65585733)

[2.4.2. Fonctionnalités en écriture 8](#_Toc65585734)

[2.4.3. Fonctionnalités en lecture 8](#_Toc65585735)

[3. Contraintes 9](#_Toc65585736)

[3.1. Contraintes Hardware 9](#_Toc65585737)

[3.2. Contraintes Software 9](#_Toc65585738)

[4. Performances 10](#_Toc65585739)

[4.1. Performances de régulation 10](#_Toc65585740)

# 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

|  |  |
| --- | --- |
| 10.10.10 | Le contrôleur doit être en mesure de piloter deux moteurs. |
| 10.10.20 | Le contrôleur doit permettre de piloter des moteurs brushless triphasés dont la puissance entre dans la plage de 20 à 150W. |
| 10.10.30 | La tension d’alimentation du moteur doit pouvoir varier entre 10 et 48V. |
| 10.10.40 | Le contrôleur doit être en mesure de délivrer un courant nominal de 6A pour le moteur. |
| 10.10.41 | Le contrôleur doit être en mesure de délivrer un courant pour le moteur allant jusqu’à 15A crête sur une durée maximale de 25 secondes. |

#### Asservissement

|  |  |
| --- | --- |
| 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

|  |  |
| --- | --- |
| 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. |
| 10.20.11 | Le contrôleur devrait être en mesure de piloter le moteur sans l’utilisation d’un quelconque capteur (technologie Sensorless). |
| 10.20.20 | Le contrôleur doit être en mesure de définir la position du rotor du moteur à l’aide de trois sondes de Hall digitales. |
| 10.20.21 | Le contrôleur doit être en mesure de définir la position du rotor du moteur à l’aide de trois sondes de Hall analogiques. |
| 10.20.30 | Le contrôleur doit être en mesure de définir la position du rotor du moteur à l’aide d’un codeur incrémental digital (Quadrature). |
| 10.20.31 | Le contrôleur doit être en mesure de définir la position du rotor du moteur à l’aide d’un codeur incrémental analogique (SIN/COS). |
| 10.20.40 | Le contrôleur doit être en mesure de définir la position du rotor du moteur à l’aide d’un codeur absolu digital dont la résolution n'excède pas 10 bits.. |
| 10.20.41 | Le contrôleur doit être en mesure de définir la position du rotor du moteur à l’aide d’un codeur absolu analogique. |

### Alimentations

|  |  |
| --- | --- |
| 10.30.10 | Le contrôleur doit être conçu de telle manière à pouvoir être alimenté par une alimentation de laboratoire. |
| 10.30.11 | Si le contrôleur est alimenté par une alimentation de laboratoire, il doit être en mesure de dissiper l’énergie générée lors d’un freinage du moteur. |
| 10.30.20 | Le contrôleur doit être conçu de telle manière à pouvoir être alimenté par une batterie. |
| 10.30.21 | Si le contrôleur est alimenté par une batterie, il doit être en mesure de récupérer l’énergie générée lors d’un freinage du moteur et la stocker dans la batterie. |
| 10.30.30 | Il doit être possible de définir le type d’alimentation du contrôleur sans avoir besoin de le reprogrammer. |

## Mesures internes

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

### Mesure de courant

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

### Mesure de tension

|  |  |
| --- | --- |
| 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

|  |  |
| --- | --- |
| 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. |

### Bus de données CAN

|  |  |
| --- | --- |
| 10.70.10 | Un protocole de communication CAN doit être mis en place. |
| 10.70.11 | La norme suivie pour la mise en place du protocole CAN doit être la suivante : "CANopen CiA 402". |
| 10.70.20 | L’ensemble des messages décrit par les clauses 10.60.XX doit être gérés dans ce protocole de communication. |

### Bus de données I2C

|  |  |
| --- | --- |
| 10.80.10 | Un protocole de communication I2C doit être mis en place. |
| 10.80.20 | La longueur du câble utilisé pour la communicationI2C ne soit pas excéder 20cm. |
| 10.80.20 | L’ensemble des messages décrit par les clauses 10.60.XX doivent être gérés dans ce protocole de communication. |

### Bus de données SPI

|  |  |
| --- | --- |
| 10.90.10 | Un protocole de communication SPI devrait être mis en place. |
| 10.90.20 | L’ensemble des messages décrit par les clauses 10.60.XX doivent être gérés dans ce protocole de communication dans le cas où la clause 10.90.10 est appliquée. |

### Bus de données UART

|  |  |
| --- | --- |
| 10.90.10 | Un protocole de communication UART doit être mis en place. |
| 10.90.20 | L’ensemble des messages décrit par les clauses 10.60.XX doivent être gérés dans ce protocole de communication. |

## Interface PC

### Généralités

|  |  |
| --- | --- |
| 10.100.10 | Un logiciel distant sur 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

|  |  |
| --- | --- |
| 10.110.10 | Le logiciel doit permettre d'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

|  |  |
| --- | --- |
| 10.120.10 | Il doit être possible de visualiser les paramètres mémorisées dans le 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

|  |  |
| --- | --- |
| 20.10.10 | Les dimensions de la carte électronique du contrôleur doivent être plus petite que les dimensions de la carte actuelle.  La carte mesure 100mm x 80mm. |
| 20.10.20 | Le processeur du contrôleur doit être un DSP de la marque Texas Instrument et de la famille TMS320F28x. |
| 20.10.30 | Le pont triphasé doit avoir une fréquence de découpage pouvant aller de 20 à 100kHz. |

## Contraintes Software

|  |  |
| --- | --- |
| 20.20.10 | Le programme doit être écrit dans le langage C pour toutes les fonctionnalités liées au processeur.   * Lecture des entrées analogiques / digitales * Ecriture des sorties analogiques / digitales / PWM * Gestion des timers * Lecture / Ecriture des bus de communications I2C et CAN |
| 20.20.11 | Le programme doit être écrit dans le langage C++ pour l'ensemble des fonctionnalités ne répondant pas à la clause 20.20.10. |
| 20.20.12 | L'interface PC peut être codée dans le langage souhaité. |

# Performances

## Performances de régulation

|  |  |
| --- | --- |
| 30.10.10 | La boucle de régulation du contrôleur doit avoir une fréquence d'échantillonnage d'au moins 1kHz. |