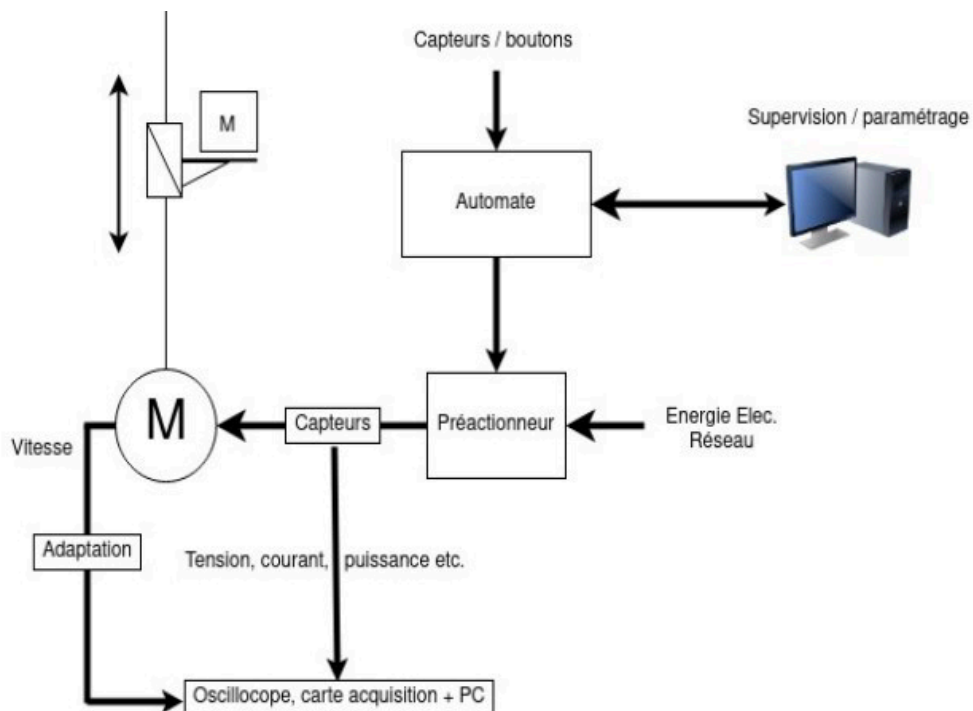


RAPPORT - PER

Glissière motorisée



Nom et prénom :

WAHBI Ahmed Amine

ABDEL KADER Omar

MARCHADOUR Eugene

Date : 11/06/2025

Module : PER

Encadrant : Mr Laurent PELT

Introduction	4
I. Présentation générale du système	5
I.1.Contexte	5
I.2.Objectifs du projet.....	5
I.3.Description fonctionnelle globale.....	5
II. Lecture encodeur et position	6
II.1.Branchement	6
II.2.Protection :	7
II.3.Configuration du compteur rapide (HSC).....	8
II.4.Conversion des impulsions	8
II.5.Affichage IHM	10
III. Commandes et logiques de fonctionnement.....	10
III.1.GEMMA:	10
III.2.GRAFCETs.....	11
III.2.1.GRAFCET F1 : Cycle automatique	11
III.2.2.GRAFCET F2 - Marche de préparation	12
III.2.3.GRAFCET F4 - Vérification manuelle	13
III.2.4.GRAFCET A5 - Réarmement après défaut.....	13
III.2.5.GRAFCET A6 - Retour à position initiale	14
IV. IHM	14
IV.1.IHM automate (API UniStream).....	14
IV.2.IHM PC (Qt/QML/Python).....	17
IV.2.1.Technologies	17
IV.3.Fonctionnement.....	17

IV.4.Tests réalisés pour l'IHM.....	20
V. Carte électronique de mesure de courant.....	21
Conclusion.....	26
Annexes	27

Introduction

Ce projet s'inscrit dans le cadre du module PER et porte sur la motorisation et l'instrumentation d'une glissière verticale. L'objectif est double : d'une part, concevoir un système capable de mesurer précisément la consommation énergétique d'un moteur à courant continu soumis à des charges variables ; d'autre part, évaluer ses performances dynamiques.

Le système repose sur l'utilisation d'un automate UniStream avec IHM intégrée pour piloter un moteur par commande PWM, lire un encodeur incrémental, et visualiser les données. Une carte électronique spécifique a également été conçue pour mesurer le courant moteur à l'aide d'une résistance de shunt.

Parallèlement, une IHM PC a été développée en Python et QML, permettant l'acquisition, l'affichage et la sauvegarde des paramètres moteur en temps réel via Modbus TCP.

Ce rapport présente l'ensemble du travail réalisé : configuration des équipements, développement logiciel, conception électronique, ainsi que les tests de validation.

I. Présentation générale du système

I.1.Contexte

Dans un contexte de transition énergétique et de recherche de sobriété, la maîtrise et l'optimisation des consommations électriques deviennent des enjeux essentiels. Pour évaluer la performance énergétique d'un moteur en conditions réelles, il est indispensable de mesurer précisément la consommation sous charge, en corrélation avec les profils de fonctionnement.

Le projet PER s'inscrit dans cette démarche. Il consiste à concevoir un banc de test instrumenté autour d'un système motorisé, capable de simuler des efforts mécaniques réalistes, tout en mesurant en temps réel les grandeurs physiques (position, vitesse, courant) nécessaires à l'analyse énergétique.

I.2.Objectifs du projet

Les objectifs pédagogiques et techniques du projet sont les suivants :

- Concevoir et câbler un système de déplacement vertical à l'aide d'un moteur à courant continu.
- Lire la position via un encodeur incrémental et convertir les impulsions en déplacement linéaire.
- Commander dynamiquement le moteur (vitesse variable) via PWM.
- Programmer un cycle de déplacement automatique basé sur un profil de vitesse paramétrable.
- Créer une interface IHM sur l'automate (UniStream) pour le contrôle local.
- Développer une IHM sur PC (en Python/QML) pour l'acquisition, le suivi graphique et la sauvegarde des données.
- Concevoir une carte électronique de mesure de courant adaptée à l'entrée analogique de l'automate.
- Analyser le comportement du système sous différentes charges et vitesses.

I.3.Description fonctionnelle globale

Le système repose sur le déplacement vertical d'une **masse de 10 kg** le long d'une glissière à vis, entraînée par un **moteur DC** commandé par PWM via l'automate UniStream.

La position de la masse est mesurée par un **encodeur incrémental** monté sur le moteur. L'automate convertit les impulsions de l'encodeur en millimètres et les affiche sur son IHM.

Un **cycle de fonctionnement automatique** est défini par l'utilisateur, avec des phases d'accélération, de vitesse constante, d'arrêt, puis de retour à la position initiale.

Ce cycle est visualisé et paramétré soit depuis l'IHM UniStream, soit depuis une **IHM PC** développée en QML, qui se connecte en **Modbus TCP** à l'automate.

Enfin, une **carte électronique dédiée** permet de mesurer le courant consommé par le moteur via une résistance de shunt, et d'envoyer la tension correspondante à l'entrée analogique de l'automate.

(voir **Annexe 1** et **Annexe 2** pour l'architecture matériel et schéma fonctionnel du système).

II. Lecture encodeur et position

II.1. Branchement

L'encodeur incrémental est connecté aux **entrées rapides (High Speed Inputs)** de l'automate Unitronics UniStream. Ces entrées sont conçues pour détecter les signaux impulsions à haute fréquence générés par les canaux A et B de l'encodeur (quadrature). Voici ci dessous le tableau de branchement de l'encodeur avec l'automates.

Fil encodeur	Branche sur l'automate
A (Phase A)	I10 (entrée rapide)
B (Phase B)	I11 (entrée rapide)
Vcc	Alimentation 18 V
GND	CM2 (commun des entrées rapides)

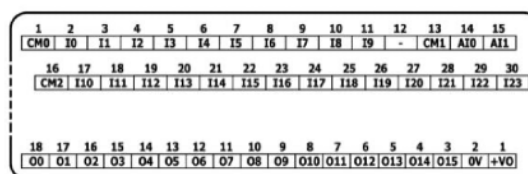
US5-Bx-T42

Wire colour

- Purple (1)
- Blue (2)
- Green (3)
- Brown (4)

Connection

- Hall Sensor B Vout
- Hall sensor A Vout
- Hall sensor ground
- Hall sensor Vcc



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CM0	I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	-	CM1	A10	A11
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
CM2	I10	I11	I12	I13	I14	I15	I16	I17	I18	I19	I20	I21	I22	I23
18	17	16	15	14	13	12	11	10	9	8	7	6	5	4
00	01	02	03	04	05	06	07	08	09	010	011	012	013	014
015	016	017	018	019	020	021	022	023	024	025	026	027	028	029

Figure 1 : Pinout encoder

Figure 2 : Pinout automate

Remarque : CM2 est relié au **0V** de l'alimentation encodeur.

Branchement encodeur incrémental vers automate

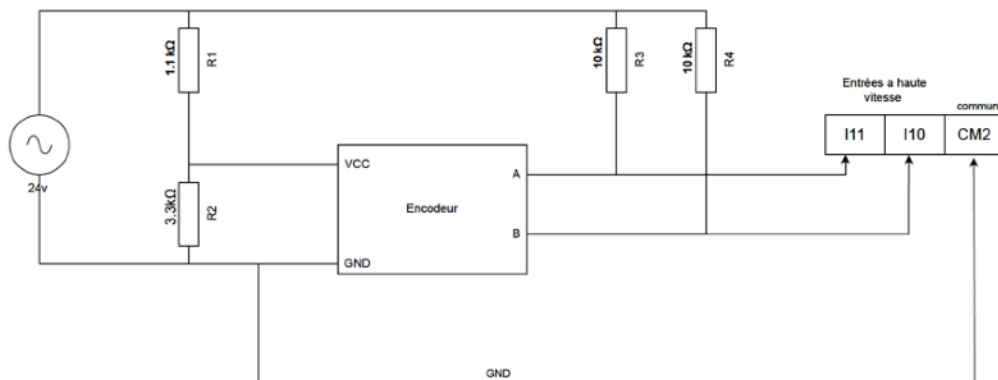


Figure 3 : Branchement de l'encodeur avec l'automate

II.2. Protection :

Les signaux A et B de l'encodeur incrémental sont souvent de type **collecteur ouvert** (ou **drain ouvert**), ce qui signifie qu'ils ne peuvent pas générer eux-mêmes un niveau logique haut. Pour garantir un signal exploitable par l'automate, on utilise donc des **résistances de pull-up** de **10 kΩ** reliées au **+24 V**. Celles-ci forcent le signal à remonter au niveau haut lorsqu'il n'est pas activé par l'encodeur, assurant ainsi une transition franche entre les états logiques. La valeur de 10 kΩ est un compromis standard : suffisamment faible pour garantir une détection fiable du niveau haut, tout en limitant le courant inutile consommé dans l'état actif.

- **Résistances de pull-up de 10 kΩ sur les signaux A et B**

Les signaux A et B de l'encodeur incrémental sont souvent de type **collecteur ouvert** (ou **drain ouvert**), ce qui signifie qu'ils ne peuvent pas générer eux-mêmes un niveau logique haut. Pour garantir un signal exploitable par l'automate, on utilise donc des **résistances de pull-up** de **10 kΩ** reliées au **+24 V**. Celles-ci forcent le signal à remonter au niveau haut lorsqu'il n'est pas activé par l'encodeur, assurant ainsi une transition franche entre les états logiques. La valeur de 10 kΩ est un compromis standard : suffisamment faible pour garantir une détection fiable du niveau haut, tout en limitant le courant inutile consommé dans l'état actif.

- **Diviseur de tension pour abaisser le +24 V à +18 V**

L'encodeur utilisé ne supporte pas une tension d'alimentation supérieure à **20 V**, alors que l'installation fonctionne généralement en **+24 V**. Pour rester dans la plage de fonctionnement sécurisée de l'encodeur et éviter tout risque de dégradation ou de destruction, on utilise un **diviseur de tension**. Ce montage passif, composé de deux résistances en série, permet d'abaisser la tension à une valeur compatible ici environ **18 V**. Le principe est simple : une partie de la tension est "prélevée" aux bornes de la seconde

résistance, selon la loi du pont diviseur. Cette solution est économique, facile à mettre en œuvre, et parfaitement adaptée lorsque le courant consommé par l'encodeur est faible. Dans ce cas, on utilise $R2 = 1,1 \text{ k}\Omega$ et $R1 = 3,3 \text{ k}\Omega$.

Wires are the colours from the actual cable.
The hall sensors accept voltages between 3.5v and 20v.
The outputs are open collector and require pull-ups to whatever signal level is required.
On the MD49 they are powered from 12v and pulled up to 5v for the signals.

Figure 4 : Extrait du datasheet de l'encodeur

II.3. Configuration du compteur rapide (HSC)

Dans UniLogic®, la configuration se fait en accédant à :

UniLogic > Hardware Configuration > High Speed I/O > HSC Input 0

Les paramètres suivants sont définis :

Paramètre	Valeur
Interface	Quadrature (A/B)
Counting Direction	Normal
Filter	5,5 μ s
Is Frequency Measured	Not in use

II.4. Conversion des impulsions

L'encodeur utilisé génère 980 impulsions pour une rotation complète du moteur. Ce moteur entraîne une vis qui convertit le mouvement rotatif en déplacement linéaire, avec une avance de 4 mm par tour.

On en déduit que chaque impulsion correspond à un déplacement linéaire de :

$$\frac{4 \text{ mm}}{980} = 0,00408 \text{ mm}$$

Ainsi, pour convertir le nombre brut d'impulsions comptées par l'automate en une position en millimètres, il suffit de multiplier cette valeur par le facteur de conversion 0,00408.

La formule utilisée dans le programme est donc :

$$Position(mm) = \text{Nombre d'impulsions} \times 0,00408$$

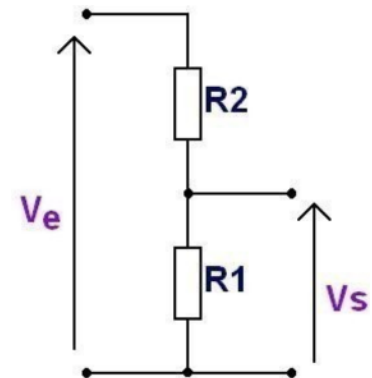


Figure 4 : Diviseur de tension


```

Calcul_Position_mm x f1 x
1 FUNCTION Calcul_Position_mm
2   VAR_EXTERNAL
3     On_board_IO: On_board_IO_Udt;
4     position_mm: REAL;
5   END_VAR
6   (* User code starts below this comment *)
7   Position_mm := TO_REAL(On_board_IO.HSC_0_Counter_Value) * 0.0408;
8   (* User code ends above this comment *)
9 END_FUNCTION
  
```

Figure 5 : Code de la fonction de calcul la position

Ce calcul est intégré dans une fonction en Structured Text, appelée à chaque cycle depuis le **Ladder principal**.

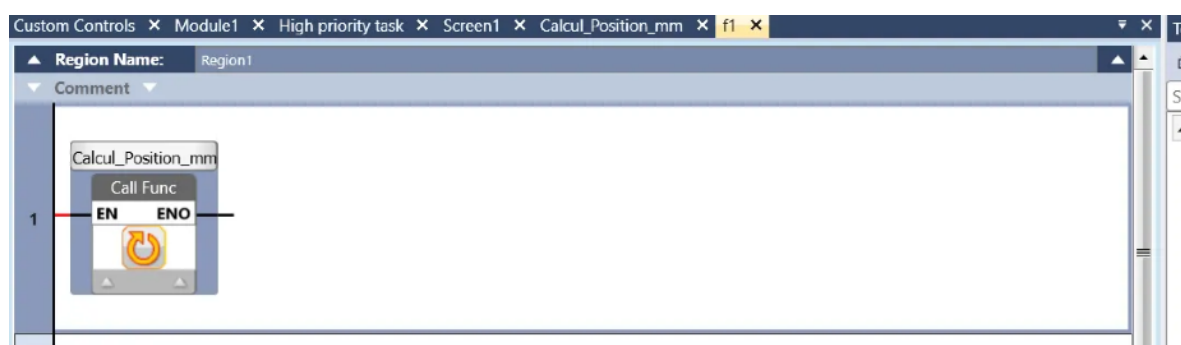
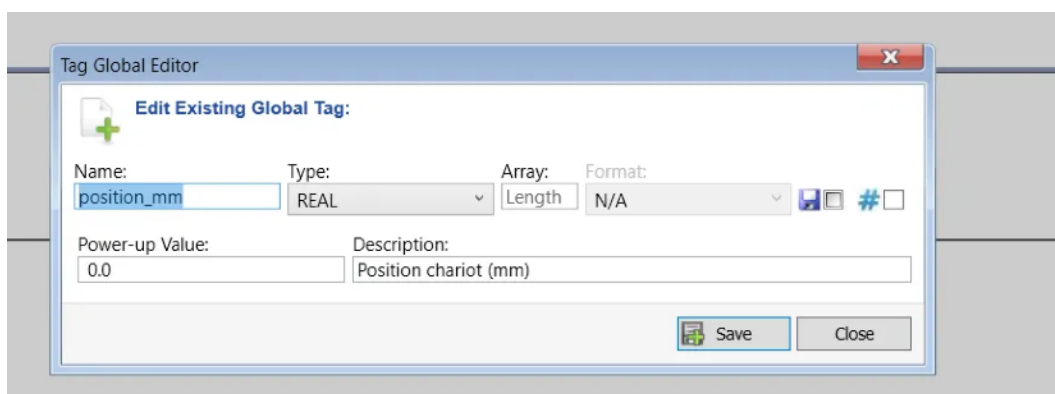


Figure 6 : La fonction est appelée depuis le **Ladder principal**

La variable globale `Position_mm`, de type `REAL`, sert à conserver et transmettre la position linéaire calculée du chariot. Cette variable est alimentée automatiquement à chaque cycle par la fonction `Calcul_Position_mm`, qui convertit le nombre



The screenshot shows the 'Tag Global Editor' dialog box with the title 'Edit Existing Global Tag:'. It contains the following fields:

- Name:** `position_mm`
- Type:** `REAL`
- Array:** `Length`
- Format:** `N/A`
- Power-up Value:** `0.0`
- Description:** `Position chariot (mm)`

At the bottom right, there are 'Save' and 'Close' buttons.

Figure 7 : Création de la variable qui indique la position

d'impulsions de l'encodeur en millimètres. Étant déclarée en global, elle peut être utilisée à la fois dans la logique automate et dans l'IHM pour affichage ou enregistrement. Elle constitue donc le lien entre la mesure physique (impulsions) et une information lisible directement par l'utilisateur ou par d'autres fonctions du programme.

II.5. Affichage IHM

Sur l'écran tactile de l'UniStream, deux valeurs principales sont affichées en temps réel

- Le nombre brut d'impulsions comptées (HSC_0_Counter_Value)
- La position convertie en millimètres (Position_mm)

Ces valeurs sont affichées via des **champs numériques (Numeric Box)**. Un bouton "ON" permet de commander manuellement le moteur pour observer le déplacement et vérifier que les valeurs évoluent correctement.

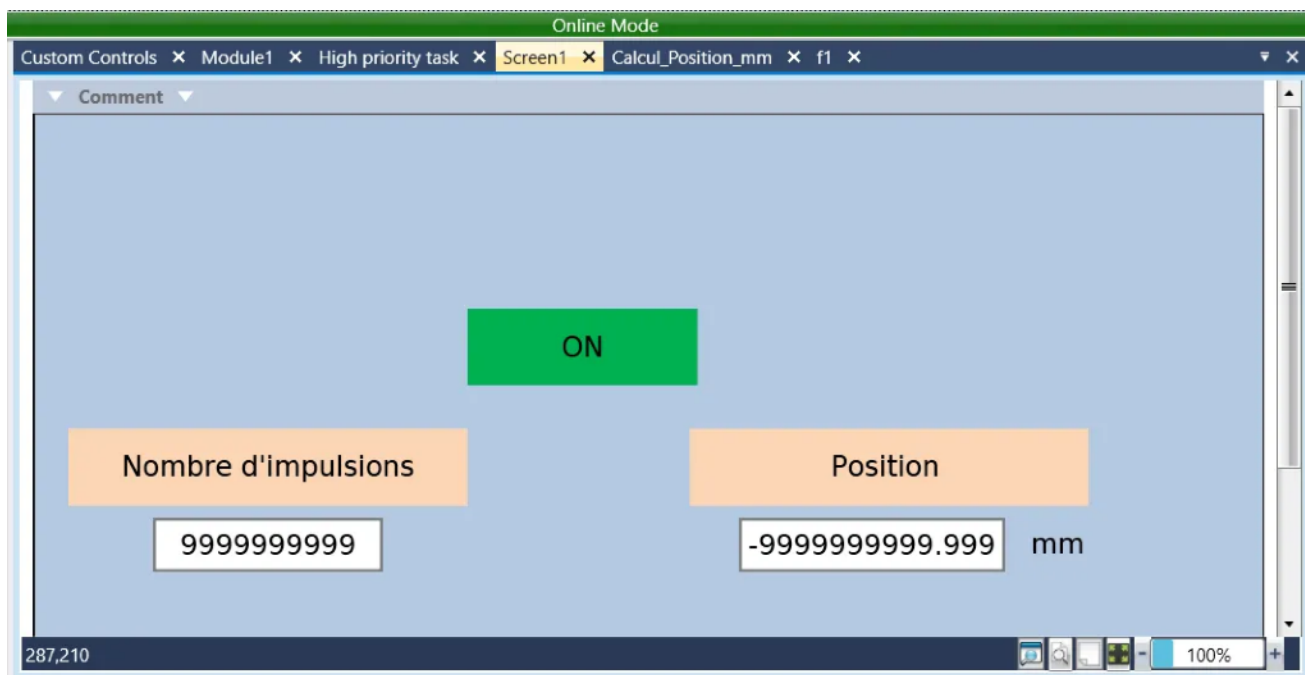


Figure 8 : Interface homme machine

III. Commandes et logiques de fonctionnement

III.1. GEMMA:

Le GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts) décrit tous les états du système : mise en service, production normale, tests, défauts. Il structure le comportement de l'automate à haut niveau. **(voir Annexe 3).**

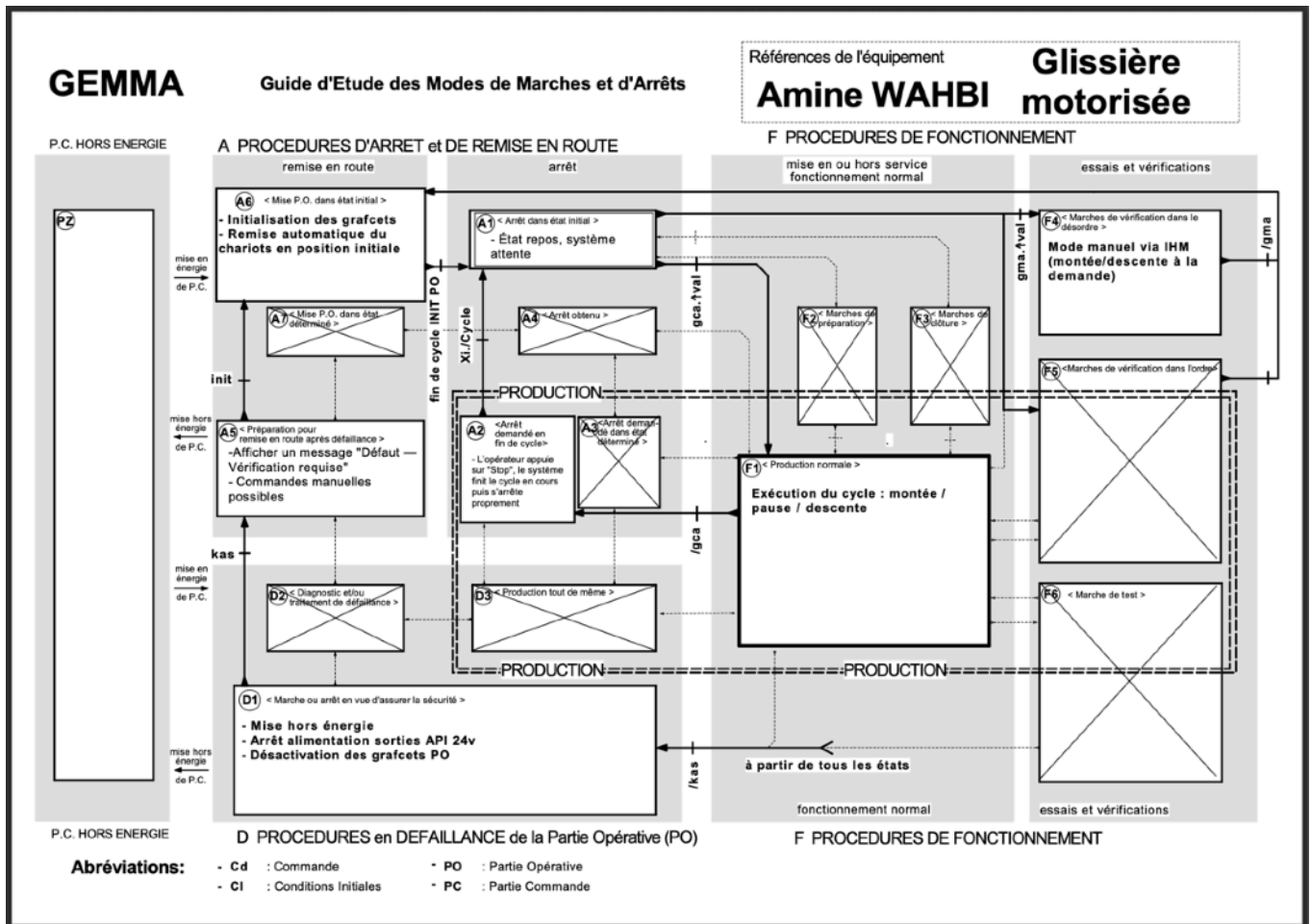


Figure 9 : Guide d'Étude des Modes de Marches et d'Arrêts - Glissière motorisée

III.2.GRAFCETs

Dans cette section, les différents GRAFCETs mis en œuvre pour piloter le système de la glissière motorisée sont présentés. Chaque GRAFCET correspond à une situation fonctionnelle spécifique (manuel, automatique, défaut, etc.). Chaque étape est numérotée dans une plage dédiée pour faciliter l'implémentation et la lecture dans UniLogic®.(voir annexes 4 , 5 , 6 et 7)

III.2.1.GRAFCET F1 : Cycle automatique

Ce GRAFCET pilote un **cycle automatique complet** : montée du chariot en 3 phases (rapide, constante, ralentie), pause haute, puis descente en 3 phases similaires. Le cycle est interrompu en cas de défaut ou sur appui arrêt.

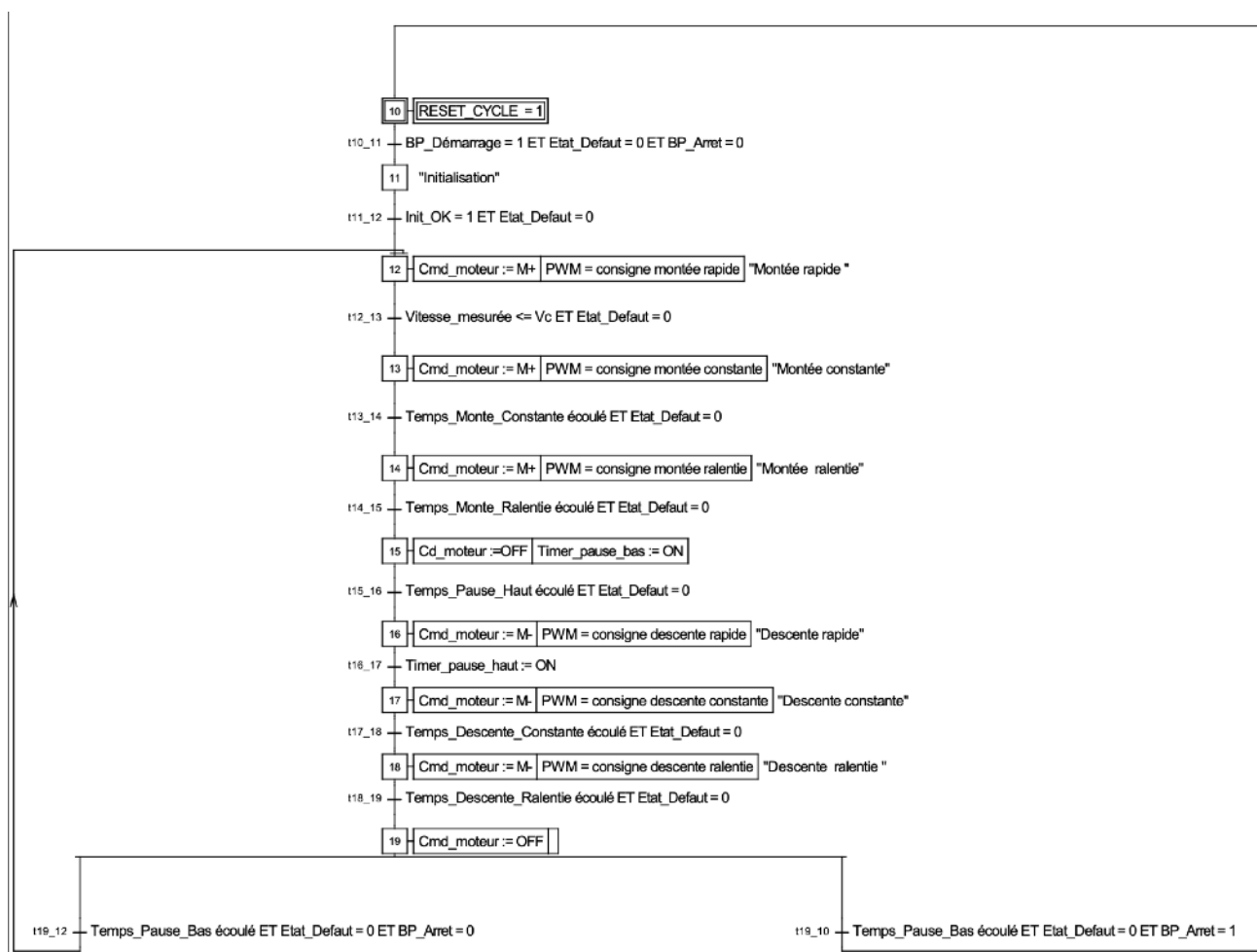


Figure 10 : GRAFCET F1 – Cycle automatique

III.2.2.GRAFCET F2 – Marche de préparation

Ce GRAFCET gère la phase de démarrage sécurisé : vérification des signaux de sécurité, initialisation des compteurs, timers et variables.

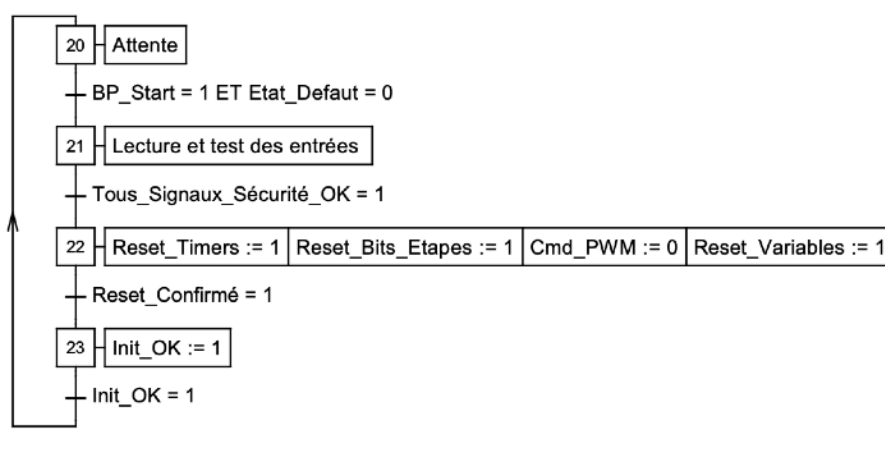


Figure 11 : GRAFCET F2 – Marche de préparation

III.2.3. GRAFCET F4 – Vérification manuelle

Ce mode permet à l'opérateur de tester les déplacements manuellement, pour les vérifications ou essais.

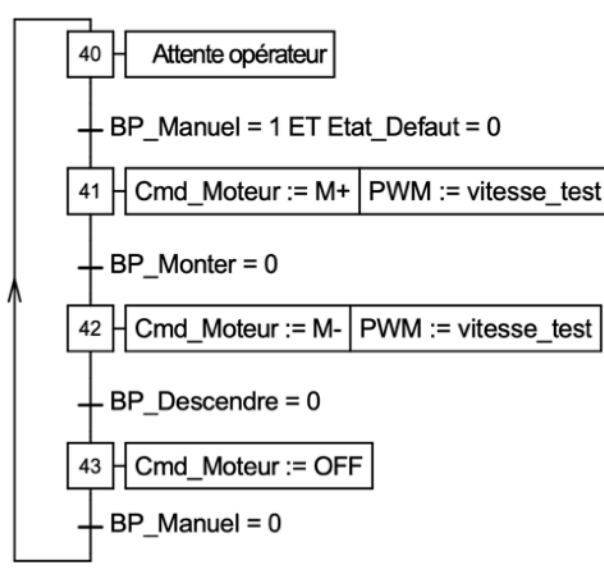


Figure 12 : GRAFCET F4 – Vérification manuelle

III.2.4. GRAFCET A5 – Réarmement après défaut

Ce GRAFCET est activé après un défaut ou un arrêt d'urgence. Il vérifie l'état de sécurité avant d'autoriser le redémarrage du système.

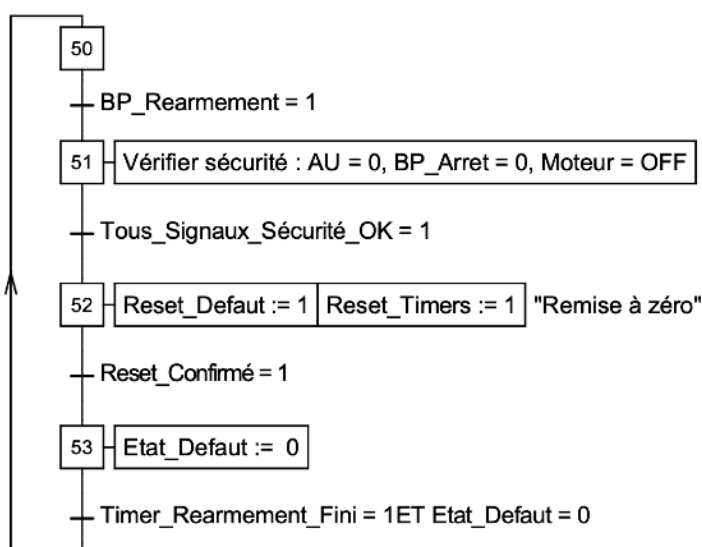


Figure 13 : GRAFCET A5 – Réarmement après défaut

III.2.5. GRAFCET A6 – Retour à position initiale

Le système déplace le chariot vers la position d'origine $x = 0$, en décidant dynamiquement de la direction à prendre selon la position actuelle.

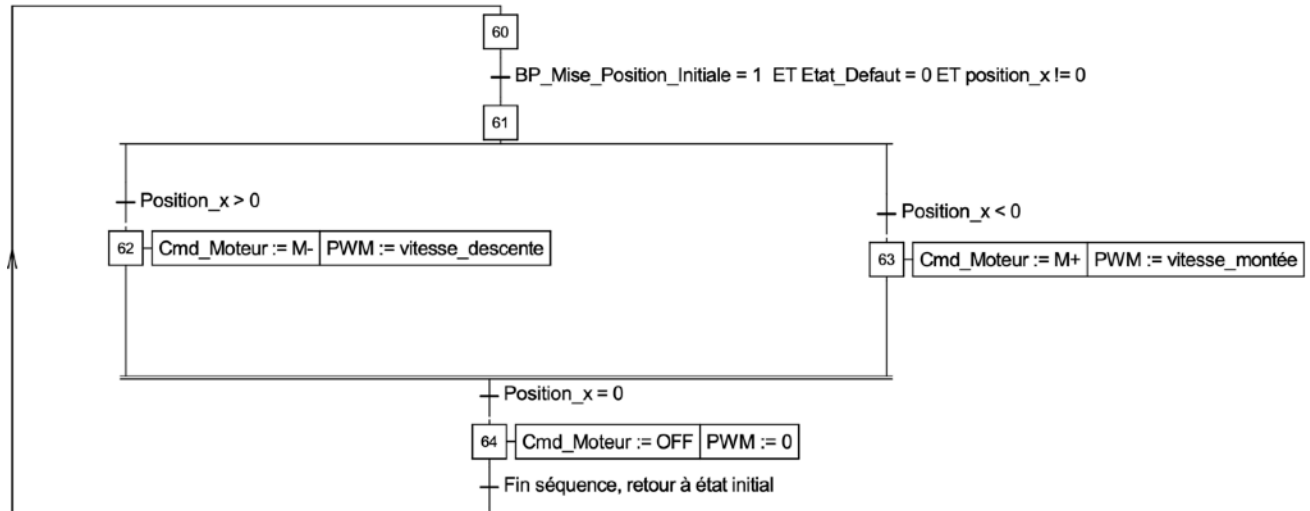


Figure 14 : GRAFCET A6 – Retour à position initiale

IV. IHM

IV.1.IHM automate (API UniStream)

Initialement, si le système est hors tension et que l'on allume l'automate, le premier écran à afficher sera celui de hors-tension .



Figure 15: IHM API, Hors tension

Sinon, on passe directement à l'écran du mode manuel en sécurité figure ci-dessous pour tester le bon fonctionnement du moteur. Dans ce mode, on peut déplacer manuellement la masse à la vitesse qu'on souhaite pour régler sa position avant de démarrer le cycle POM (prise d'origine mesure).

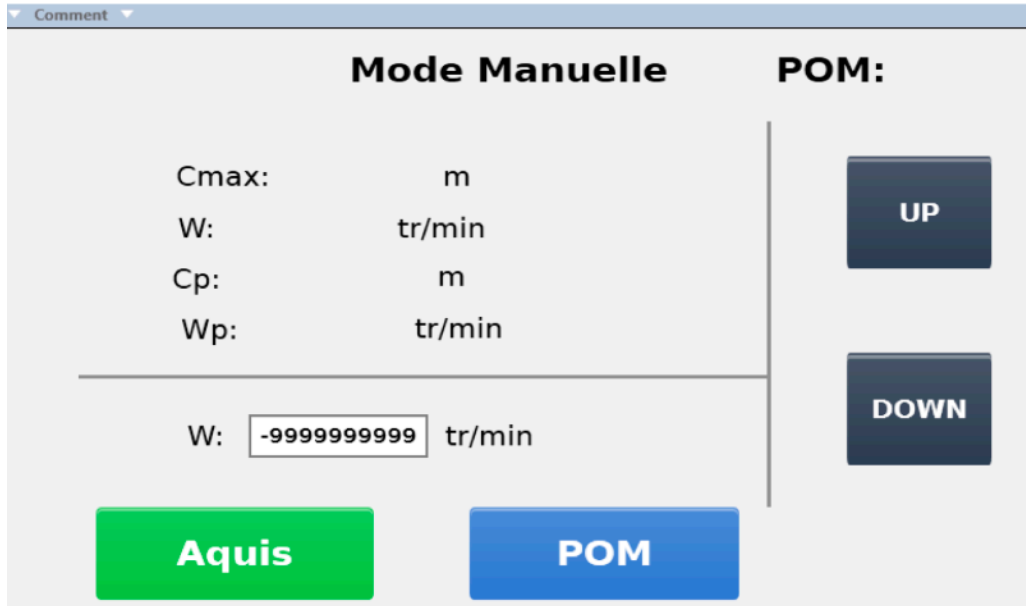


Figure 16: IHM API, mode manuelle sec

Une fois le cycle POM est terminer, on appuie sur le bouton « Acquis », on accède à l'écran d'initialisation du système .

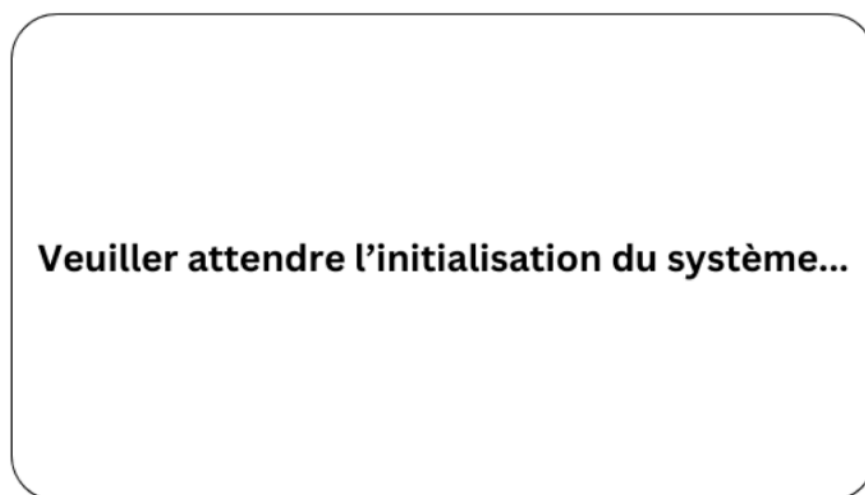


Figure 17 : IHM API, Initialisation

On attend que l'écran initial s'affiche.

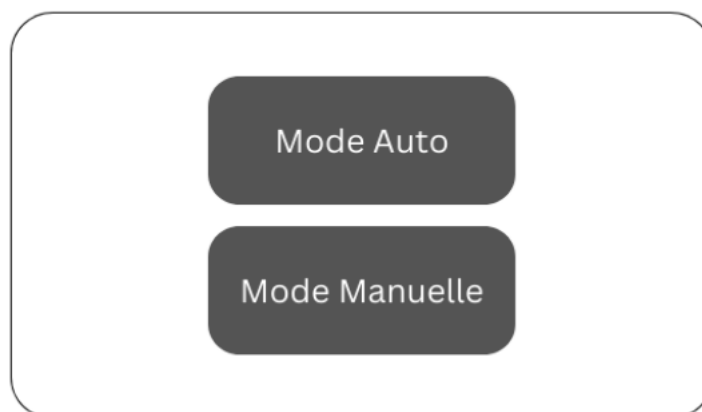


Figure 18 : IHM API , Choix du mode auto ou Manuel

À partir de là, on peut choisir soit le mode manuel (Figure 5), soit le mode automatique (Figure 6). Dans le mode manuel, on peut déplacer la masse manuellement avec la vitesse qu'on veut à la vitesse qu'on veut. Dans le mode automatique, la masse se déplace automatiquement tout en précisant l'accélération (p : pente de la vitesse), la durée que la masse va être à une vitesse constante ($T1$), la durée que la masse va être à une vitesse nulle ($T2$) et la consigne de vitesse (Wp).

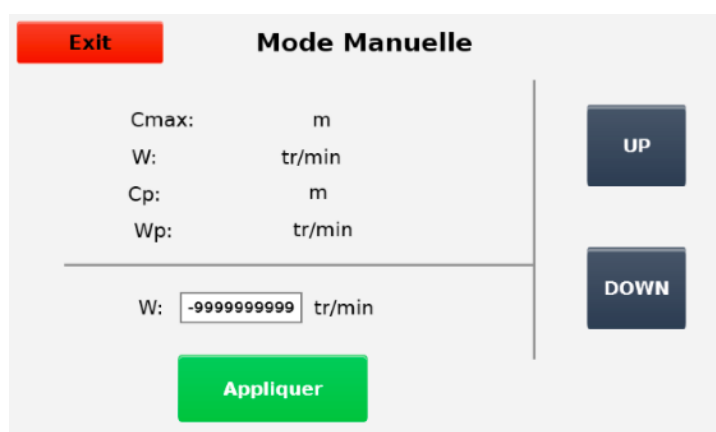


Figure 19 : IHM API , Mode manuel

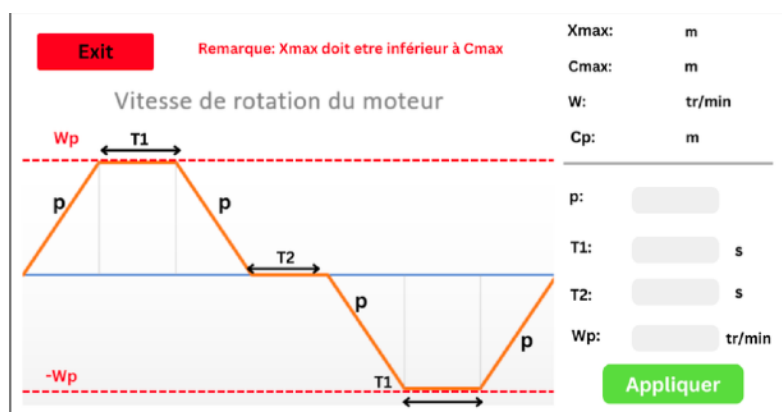


Figure 20 : IHM API , Affichage des données

POM : Pour lancer le cycle POM, il faut d'abord placer la masse au-dessus du capteur POM (celui en bleu). Une fois le cycle lancé, la masse doit descendre jusqu'à atteindre le capteur POM.

Ensuite, la distance entre le capteur POM et le capteur de début de course (celui en rouge) doit être mesurée au préalable pour programmer la descente automatique de la masse sur cette distance lorsqu'elle est positionnée sur le capteur POM.

IV.2.IHM PC (Qt/QML/Python)

IV.2.1.Technologies

L'IHM du PC a été développée en utilisant Python, Qt et QML via l'outil **Qt Design Studio**.

Qt Design Studio est un environnement de conception d'interfaces utilisateur (UI) basé sur **QML** (Qt Modeling Language). Il permet aux développeurs et designers de créer des interfaces graphiques interactives et animées pour des applications desktop, mobiles ou embarquées, sans avoir à écrire manuellement tout le code QML.

Un projet QML est structuré autour de plusieurs fichiers clés :

qmlproject : fichier principal de configuration du projet.

qmlproject.qtds : fichier complémentaire qui stocke des paramètres personnalisés spécifiques à Qt Design Studio.

Le projet contient également plusieurs fichiers QML spécialisés, tels que :

Constants.qml : centralise toutes les constantes utilisées dans l'application.

EventListModel.qml : définit un modèle de données pour la liste des événements.

EventListSimulator.qml : simule un flux d'événements dynamiques, utile pour les tests ou démonstrations.

Backend.qml : fait le lien entre l'interface QML et le backend écrit en Python ou C++.

Les fichiers **ui.qml** définissent les composants graphiques de l'interface utilisateur (Boutons, Text Box...). Leur syntaxe est proche de celle du CSS. En complément, les fichiers **QML** principaux contiennent la logique applicative, en reliant les composants graphiques aux actions via du code JavaScript. La partie **Python** est utilisée pour développer le backend de l'application, gérant notamment la logique métier, les données et la communication avec l'IHM.

IV.3.Fonctionnement

L'IHM a pour but de se connecter à l'automate via Modbus TCP, d'afficher en temps réel les données provenant de l'automate (Xmax, Cmax, Vitesse (W), Cp, p, T1, T2, Wp, Courant), puis d'afficher la vitesse du moteur, le courant du moteur et la consigne de vitesse (Wp) sur un graphe en temps réel, tout en sauvegardant l'ensemble des données en temps réel dans un fichier CSV.

L'interface affiche un graphique présentant la vitesse de rotation du moteur et le courant consommé ainsi la consigne de vitesse en fonction du temps. Elle comprend également :

- Un bouton « Commencer/Arrêter » pour démarrer ou interrompre l'acquisition des données. Un indicateur signalant si l'IHM est connectée ou non à l'automate.
- La sauvegarde doit commencer automatiquement en cliquant sur le bouton « Commencer » et s'arrête en cliquant sur « Arrêter ».
- Ainsi que le graph affiche les courbes en cliquant sur « Commencer » et s'arrête en cliquant sur « Arrêter ».
- À côté, une barre de menu propose quatre options:
- Paramètres de connexion : Ouvre une fenêtre pour configurer la connexion à l'automate.
- Paramètres du moteur : Affiche toutes les données relatives au moteur (vitesse, courant, etc.).
- Paramètres de sauvegarde : Permet de : Définir l'emplacement du fichier CSV enregistrant en temps réel la vitesse du moteur et les autres paramètres.

Si un fichier CSV existant est importé, les nouvelles données s'ajoutent à ce fichier, sinon, un nouveau fichier CSV est généré automatiquement.

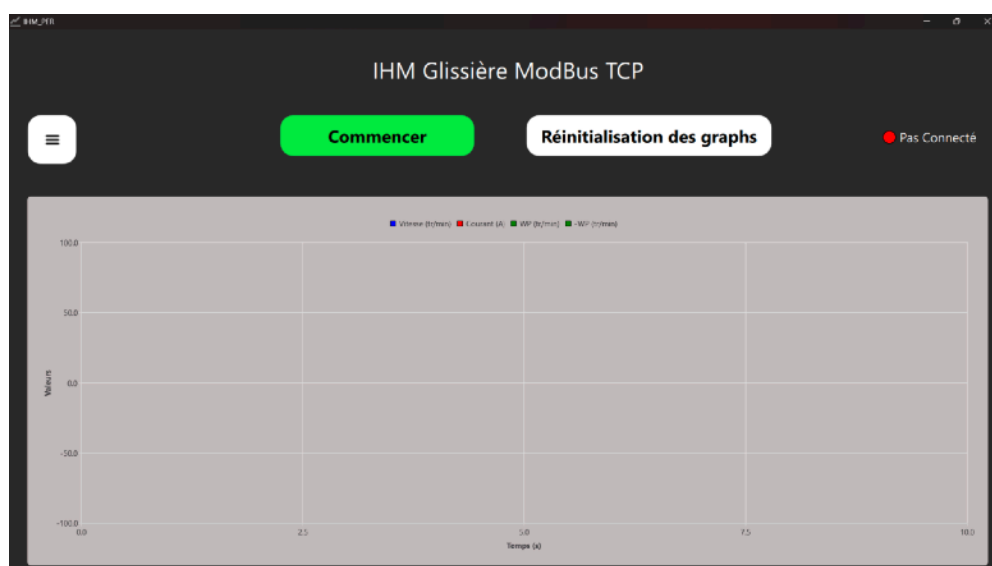
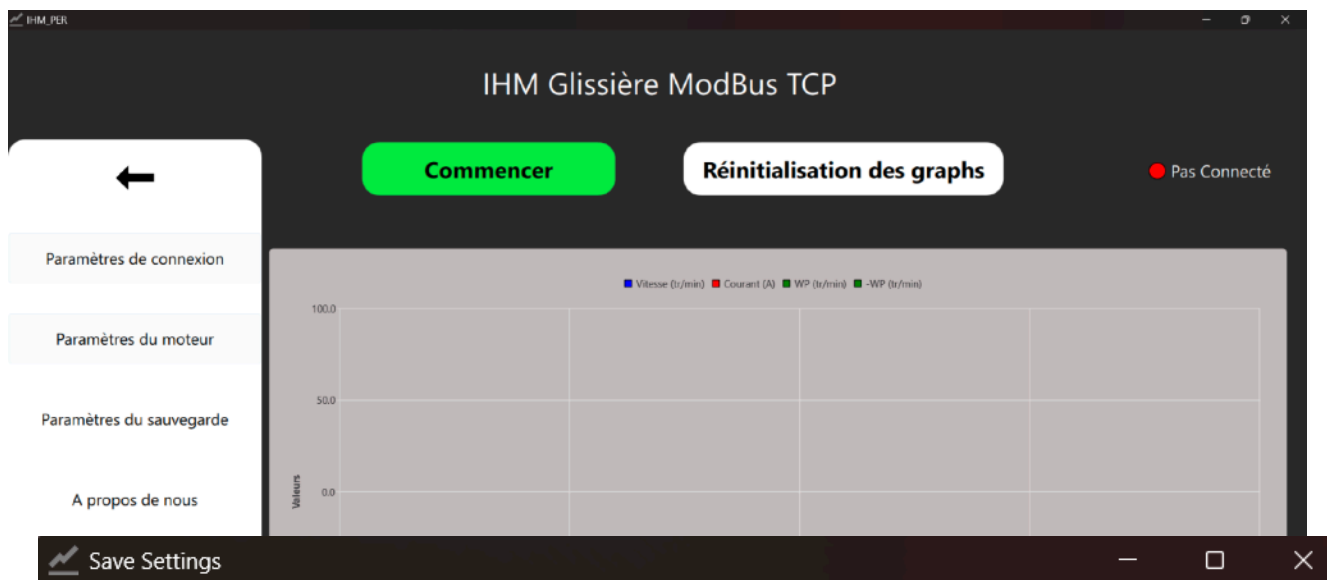


Figure 21 : IHM pc page initiale



Paramètres de sauvegarde

Chemin vers CSV



Appliquer

Paramètres du moteur

Figure 25 : IHM PC, paramètres de sauvegarde

Vitesse du moteur : tr/min

Courant du moteur : A

T1 : s

T2 : s

p :

Wp : tr/min

Cmax : m

Cp : m

Xmax : m

Figure 23 : IHM PC, paramètres du moteur

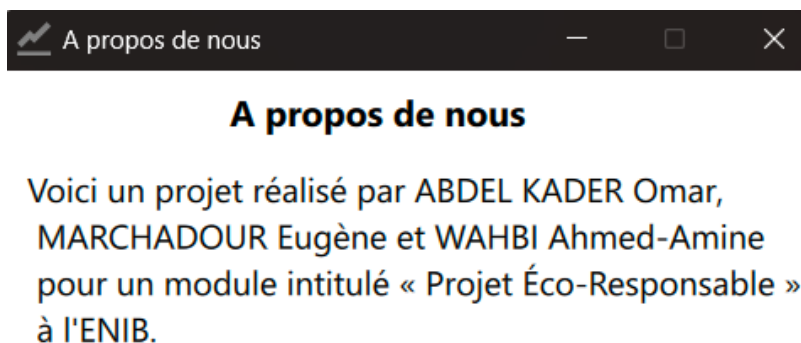


Figure 24 : IHM PC, à propos de nous

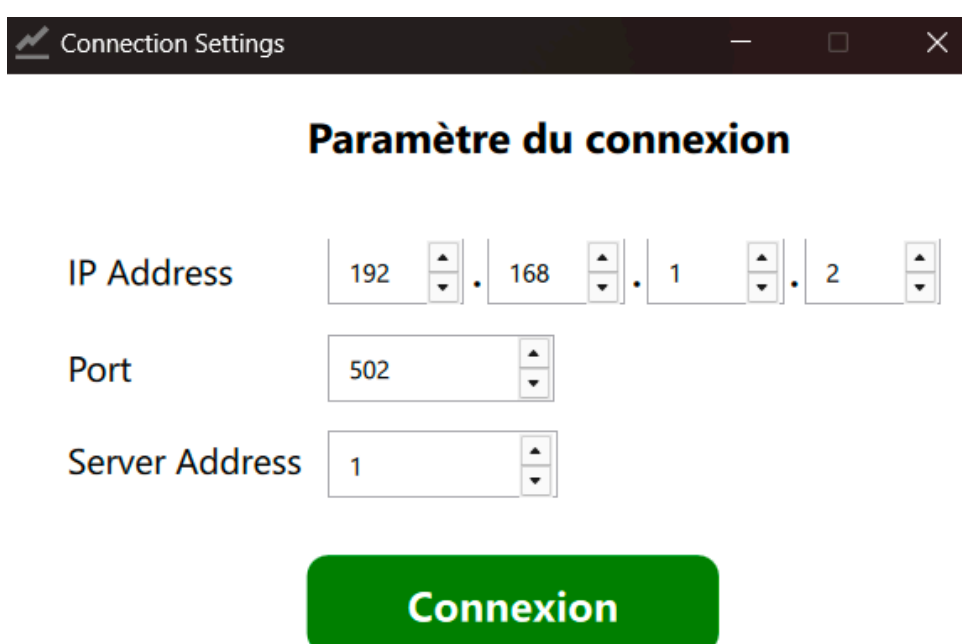


Figure 26 : IHM PC, paramètre de la connexion

IV.4.Tests réalisés pour l'IHM

Pour vérifier le bon fonctionnement de l'IHM du PC, j'ai créé une page sur l'automate regroupant tous les paramètres à envoyer normalement à l'interface (Figure 12).

Ensuite, je me suis connecté à l'automate et j'ai modifié les valeurs des paramètres via celui-ci pour vérifier si les changements se répercutaient également sur l'IHM du PC. Effectivement, les modifications ont été appliquées au niveau des paramètres du moteur, ainsi que dans le graphique et le fichier Excel.

Comment ▼			
Consigne vitesse :	-999999999 tr/min	Vitesse :	-999999999 tr/min
Consigne courant :	-999999999 A	Courant :	-999999999.99 A
Consigne Cp:	-999999999 m	Cp:	-999999999.999 m
Consigne T1 :	-999999999.999 s	T1 :	-999999999.999 s
Consigne T2 :	-999999999.999 s	T2 :	-999999999.999 s
Consigne p :	-999999999.999	p :	-999999999.999
Consigne Cmax :	-999999999.999 m	Cmax :	-999999999.999 m
Consigne Xmax :	-999999999.999 m	Xmax :	-999999999.999 m
Consigne Wp :	-999999999.999 m	Wp :	-999999999.999 m

Figure 27 : IHM API, écran de test

V. Carte électronique de mesure de courant

Je suis parti sur une solution basée sur la mesure tension aux bornes d'une résistance de shunt branchée en série avec le moteur.

Pour commencer j'ai fait les mesures des différences de tension au borne de la résistance pour la charge maximale de 10kg. Les tests ont été fait avec le moteur installé sur la glissière et tournant au maximum de sa vitesse (alimenté en 24V continu).

J'ai obtenu une différence de tension de 66 mA au maximum lors de la montée de la charge et une tension maximale de 30mA lors de la descente.

Grâce à ces tests, j'ai estimé que la tension maximale ne devrait pas dépasser 70 mA. Je me suis basé sur cette valeur pour la conception du système.

J'ai ensuite réalisé un schéma permettant d'adapter la tension aux bornes de la résistance de shunt en une tension interprétable par l'automate.. L'entrée analogique de l'automate accepte une plage de tension de 0 à +10V.

La première partie avec l'AD623 et la résistance de shunt est celle qui permet de récupérer le courant consommé pour le moteur. La résistance RG permet de calibrer le gain. J'ai pris une marge sur le gain dans le cas où la différence de tension aux bornes de la résistance de shunt serait plus importante que prévus. Le lien juste en dessous permet de voir la simulation réalisée et de calculer la résistance de gain en fonction du gain nécessaire:

<https://tools.analog.com/en/diamond/#difL=-0.14&difR=0.14&difSI=-0.14&gain=30&l=0&pr=AD623&r=0&sl=0&tab=1&ty=2&vn=-5&vp=5&vr=0>

Sur la deuxième partie du schéma, j'ai ajouté un montage réhausseur suivi d'un suiveur car la plage de sortie de l'AD623 avec ce montage est de -5V;5V alors que l'entrée de l'automate accepte une plage de 0;10V. J'ai mis à la suite une simulation Itspice permettant de vérifier cette partie du montage.

Voici le schéma du système en entier :

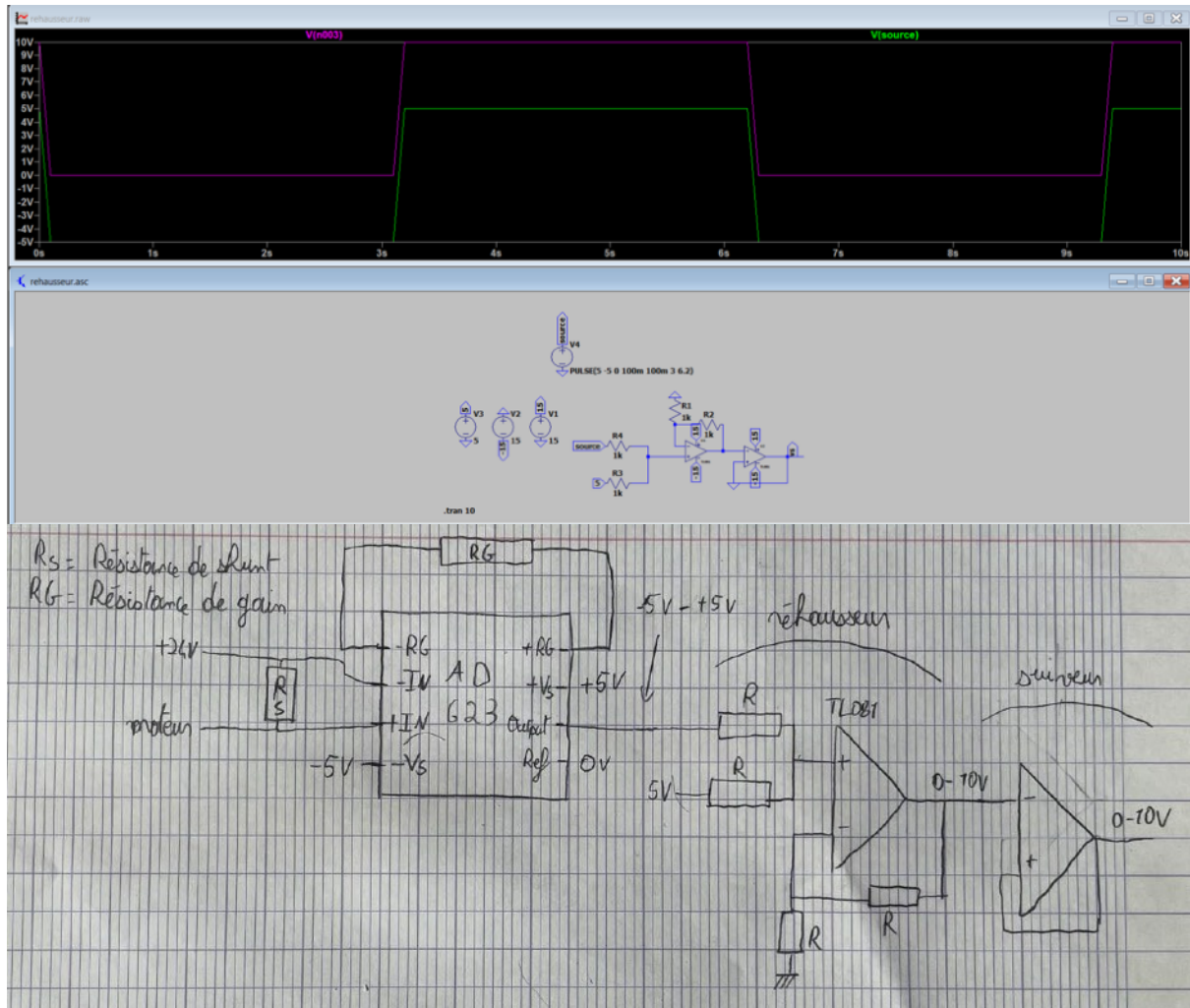


Figure 28 : Schéma du système

J'ai ensuite fait la CAO de la carte électronique grâce à EAGLE. Voici le schéma de câblage réalisé :

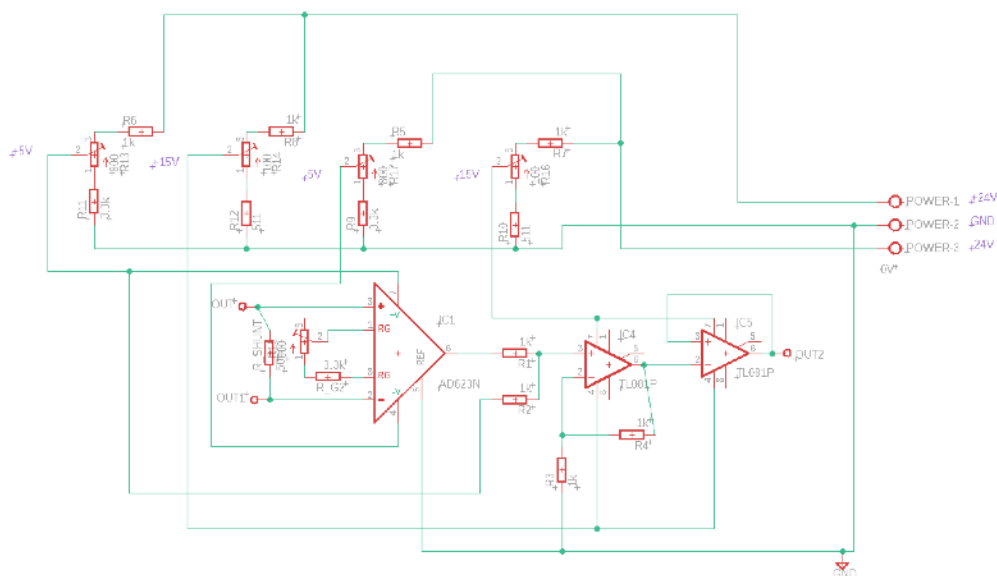


Figure 29 : CAO de la carte électronique

Voilà le schéma réalisé avec Eagle. Sur ce schéma, il y a besoin de tensions d'alimentation de +24V et -24V. J'ai réalisé 4 ponts diviseurs de tension afin d'abaisser les tensions + et - 15V pour alimenter l'AD623 et + et - 5V afin d'alimenter les deux TL081. Voilà le schéma de la carte imprimée :

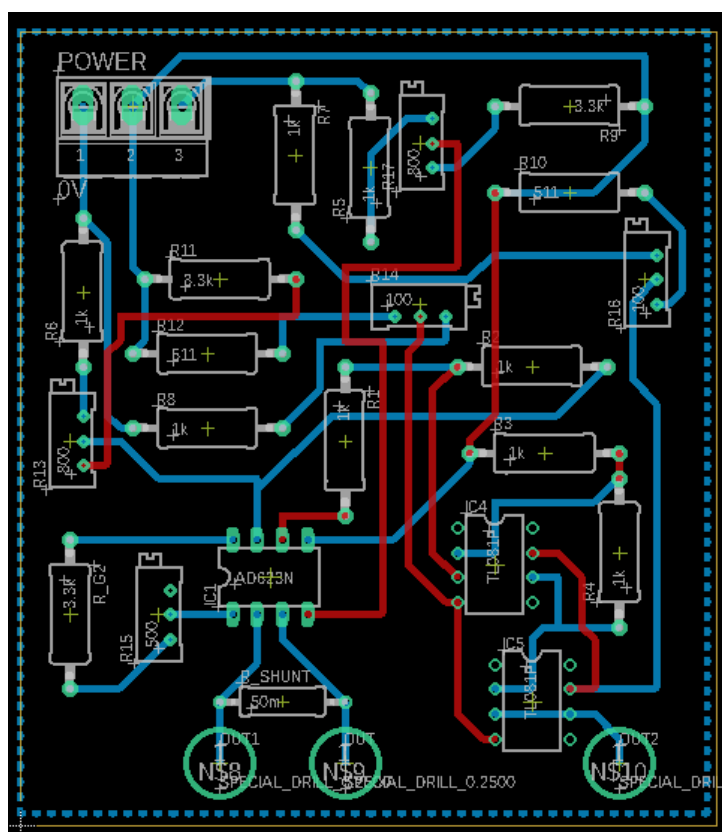


Figure 30 : Schéma de la carte imprimée

Voici la carte imprimée :

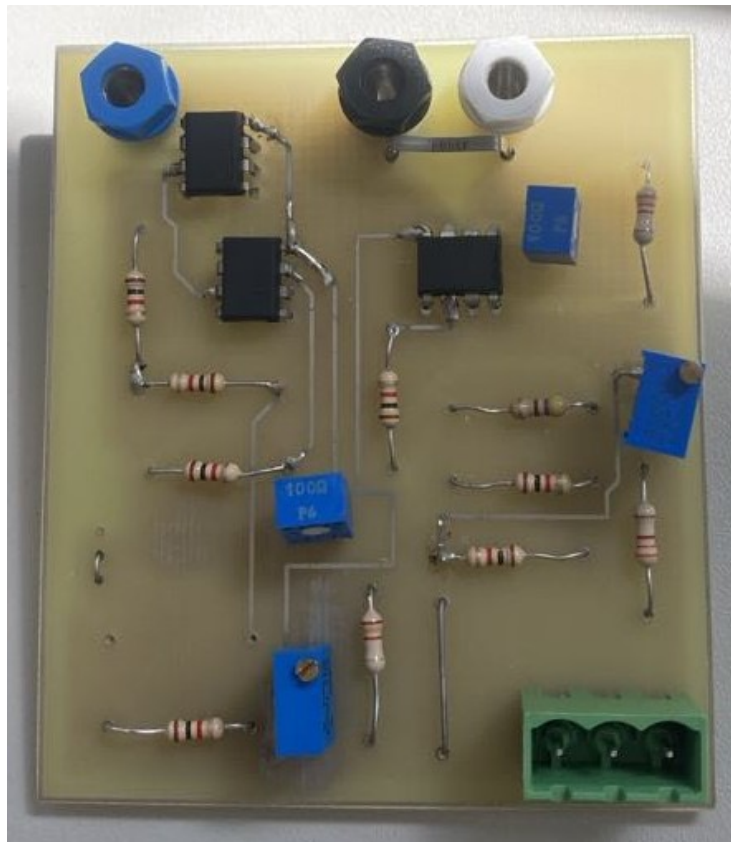


Figure 31 : Carte imprimée

Retour sur la carte imprimée :

Après l'avoir imprimée, je me suis rendu compte que nous n'aurions pas la possibilité d'avoir du -24V sur le montage, j'ai donc choisi d'ajouter un inverseur de tension au montage.

J'en ai trouvé deux qui étaient alimentées en +24V et délivrant en tension de -15V. Cela fait qu'un des deux ponts diviseurs de tensions est inutile et que les valeurs des résistances sur le deuxième sont à changer.

Voilà les liens et les données des deux convertisseurs de tensions qui ont été acheté :

<https://fr.rs-online.com/web/p/convertisseurs-dc-dc/0162997>

tension d'entrée : 18 → 32v

tension de sortie : -15v

courant maximum : 100mA

prix unité : 17.22 ht

<https://fr.rs-online.com/web/p/convertisseurs-dc-dc/9048688?gb=s>

tension d'entrée : 21.6 → 26.4v

tension de sortie : -15v

courant maximum : 33mA

prix unité : 10.48 ht

Les valeurs des résistances R6 et R11 sont interchangées.

Les valeurs des résistances R8 et R12 sont interchangées.

La valeur de R5 est maintenant de 500 ohm.

La résistance variable R17 est maintenant de 100 ohm.

La résistance R7 est remplacée par un fil, et la connexion entre la résistance R7 (changée par un fil) et les TL081 (AOP) est faite directement par un fil.

La résistance variable R16 et la résistance R10 sont supprimées.

Un dernier problème est qu'un des composants n'est pas fonctionnelle, il faut donc retester toute la carte.

Si la carte est à refaire, il faudrait modifier les 3 borniers du bas de la carte (les 3 en ronds) et il faudrait grossir les pistes entre la résistance de shunt et les 2 borniers qui lui sont directement connectés.

Autre chose partie électronique :

Il y a un risque de chute de tension de la partie lorsque la charge appliquée sur le moteur est importante et en refaisant le schéma électrique complet je me suis rendu compte que la partie commande et la partie puissance sont toutes les deux reliées. Une chute de tension comme celle-ci pourrait entraîner une casse de la partie commande, il faudrait donc prévoir une deuxième alimentation.

Conclusion

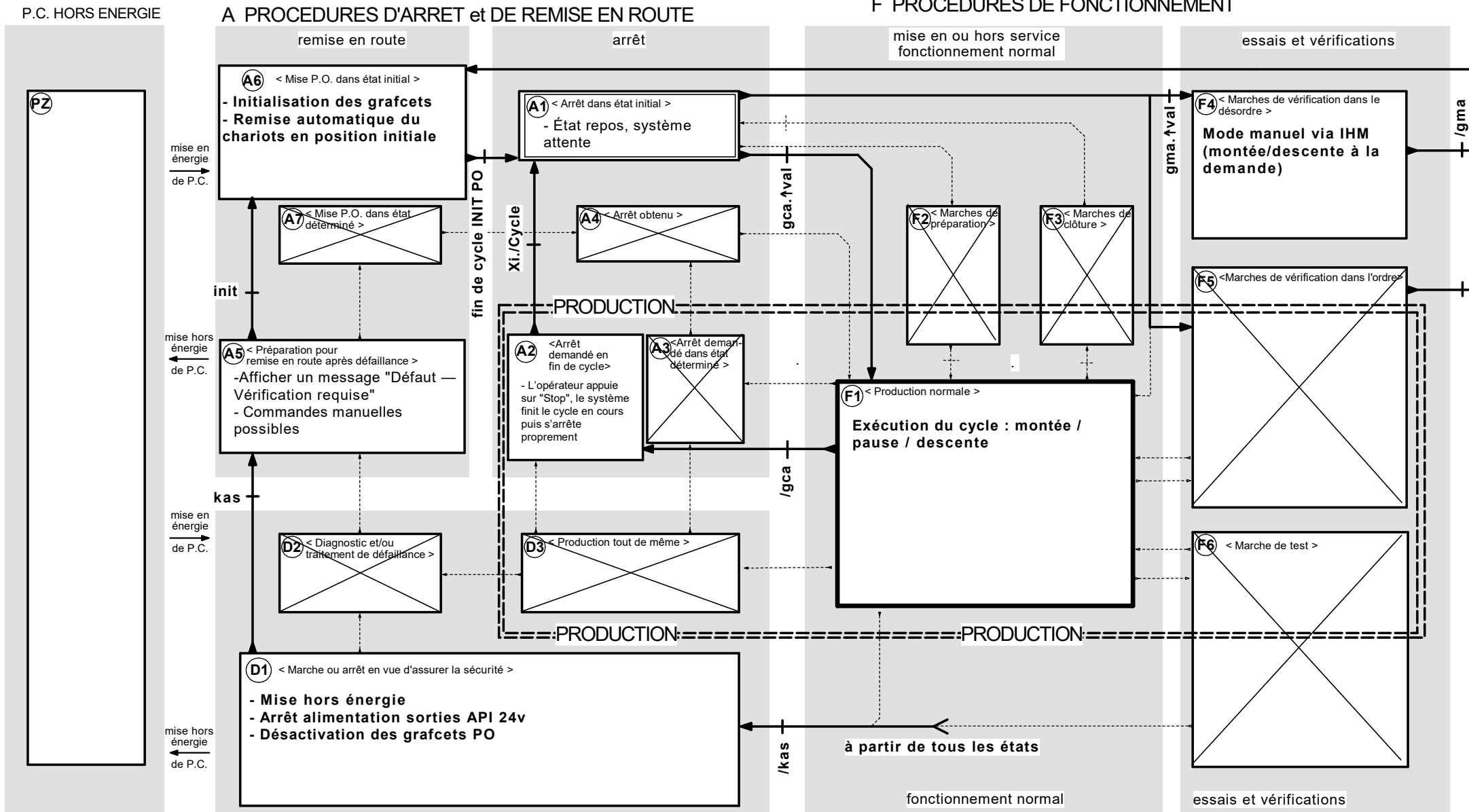
Le projet PER de glissière motorisée nous a permis de concevoir, assembler et programmer un système électromécanique complet, intégrant la commande moteur, la mesure de position et de courant, ainsi que la supervision via plusieurs interfaces IHM. Nous avons pu :

- Lire un encodeur incrémental et convertir les impulsions en position linéaire,
- Définir et programmer des cycles de fonctionnement avec GRAFCETs et GEMMA,
- Développer deux IHMs : une embarquée sur l'automate, l'autre sur PC (en Python/QML),
- Concevoir une carte électronique de mesure de courant, avec adaptation analogique et simulation,
- Réaliser et tester le banc complet avec une charge réelle.

Malgré quelques ajustements nécessaires en cours de route, notamment au niveau de l'alimentation et de la carte électronique, l'ensemble des fonctionnalités prévues ont été intégrées et validées sur prototype. Ce projet nous a permis d'aborder de manière concrète plusieurs disciplines techniques (automatisme, mesure, électronique, programmation) dans un cadre de travail collaboratif. Il constitue une base solide pour des études plus poussées sur l'optimisation énergétique des systèmes motorisés, le pilotage avancé et l'intégration industrielle.

Annexes

- https://www.unitronicsplc.com/Download/SoftwareHelp/!UniLogicWebhelp2024/Ladder/Structured_Text_Functions.htm
- <https://www.robot-electronics.co.uk/htm/emg49.htm>
- X_{max} : distance maximale qui va être calculée en fonction de p , $T1$, Wp
- C_{max} : C'est la distance entre la position initiale ($x=0$) et la position finale/maximale.
- W : C'est la vitesse de la palette en temps réelle.
- Cp : C'est la position de la masse par rapport à la position initiale en temps réelle
- P : C'est la pente de la vitesse de la rotation du moteur
- $T1$: C'est la durée que la palette reste à sa vitesse maximale.
- $T2$: C'est la durée que la palette reste à une vitesse nulle.
- Wp : Vitesse consigne à atteindre
- POM : prise d'origine mesure



P.C. HORS ENERGIE

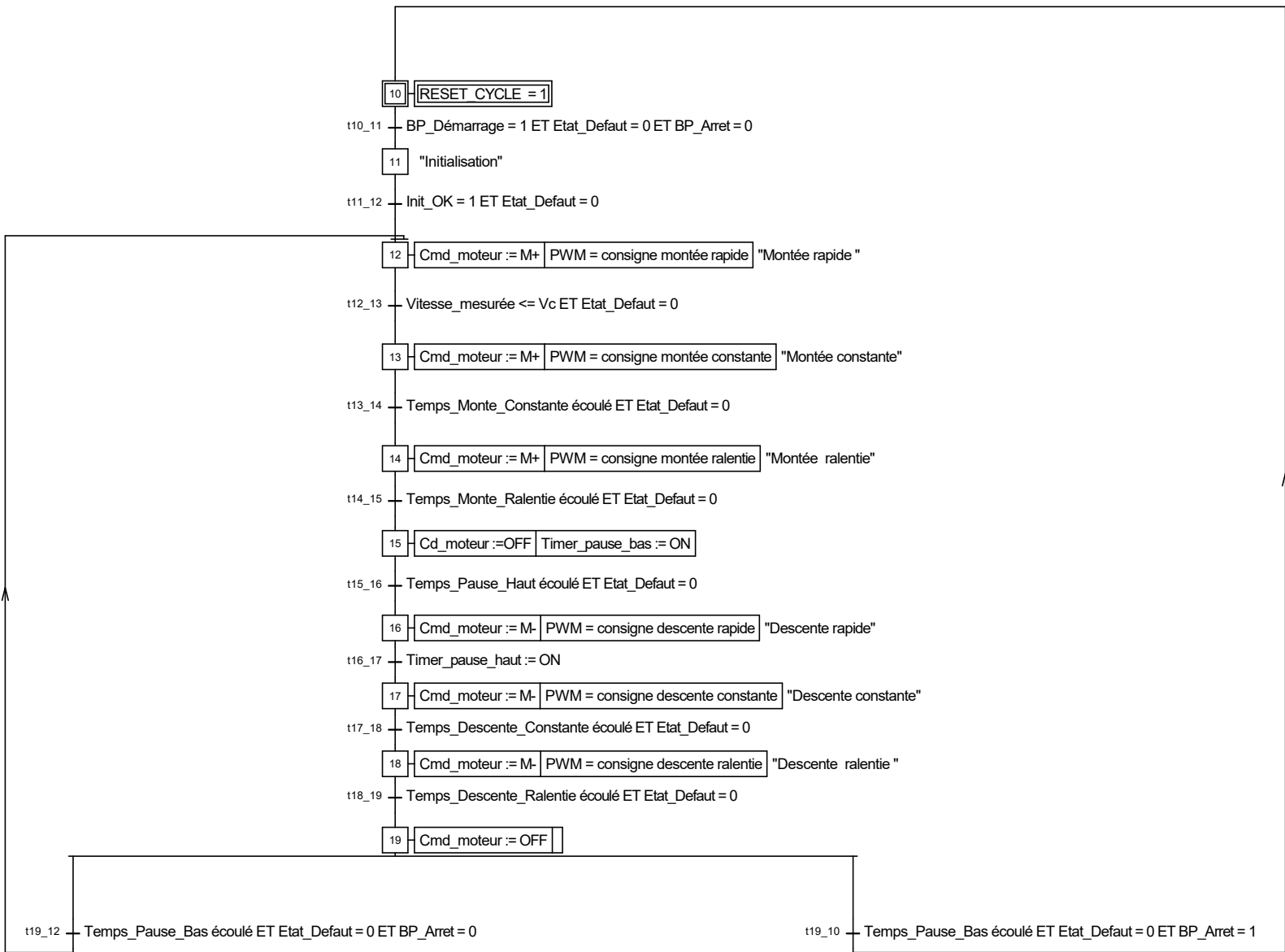
D PROCEDURES en DEFAILLANCE de la Partie Opérative (PO)

F PROCEDURES DE FONCTIONNEMENT

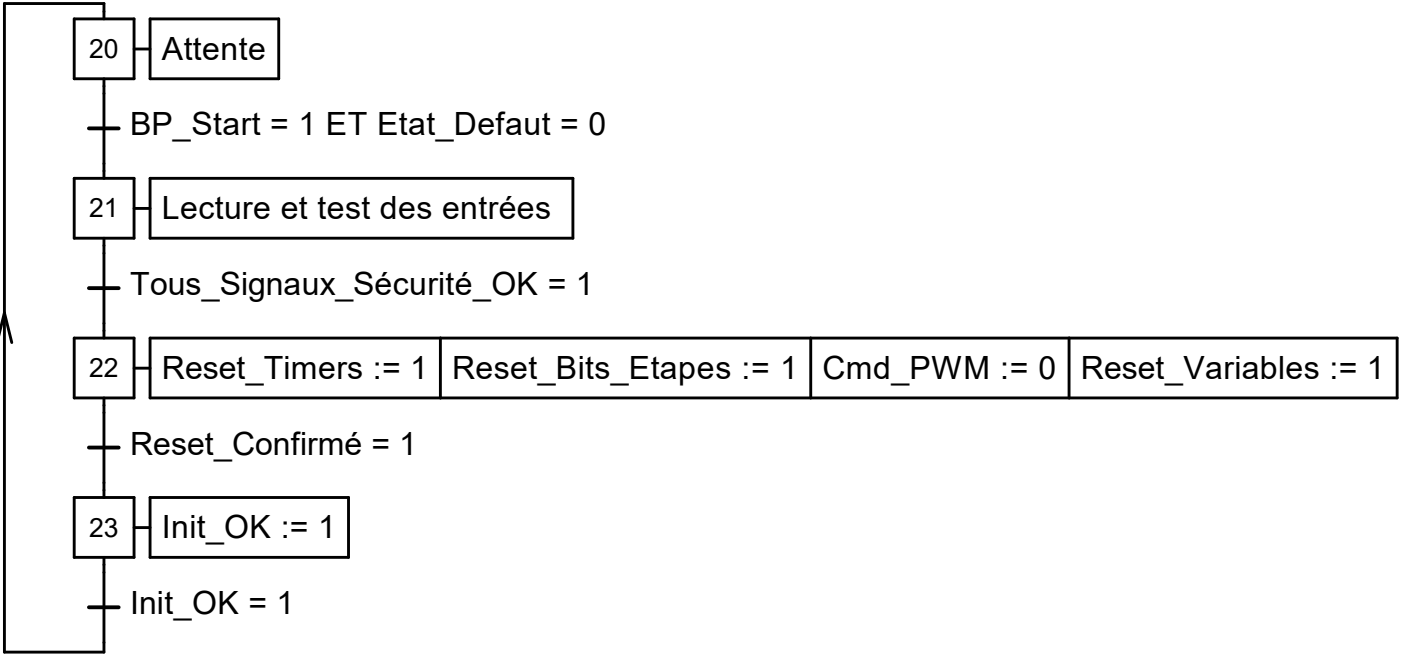
Abréviations:

- **Cd** : Commande
- **CI** : Conditions Initiales
- **PO** : Partie Opérative
- **PC** : Partie Commande

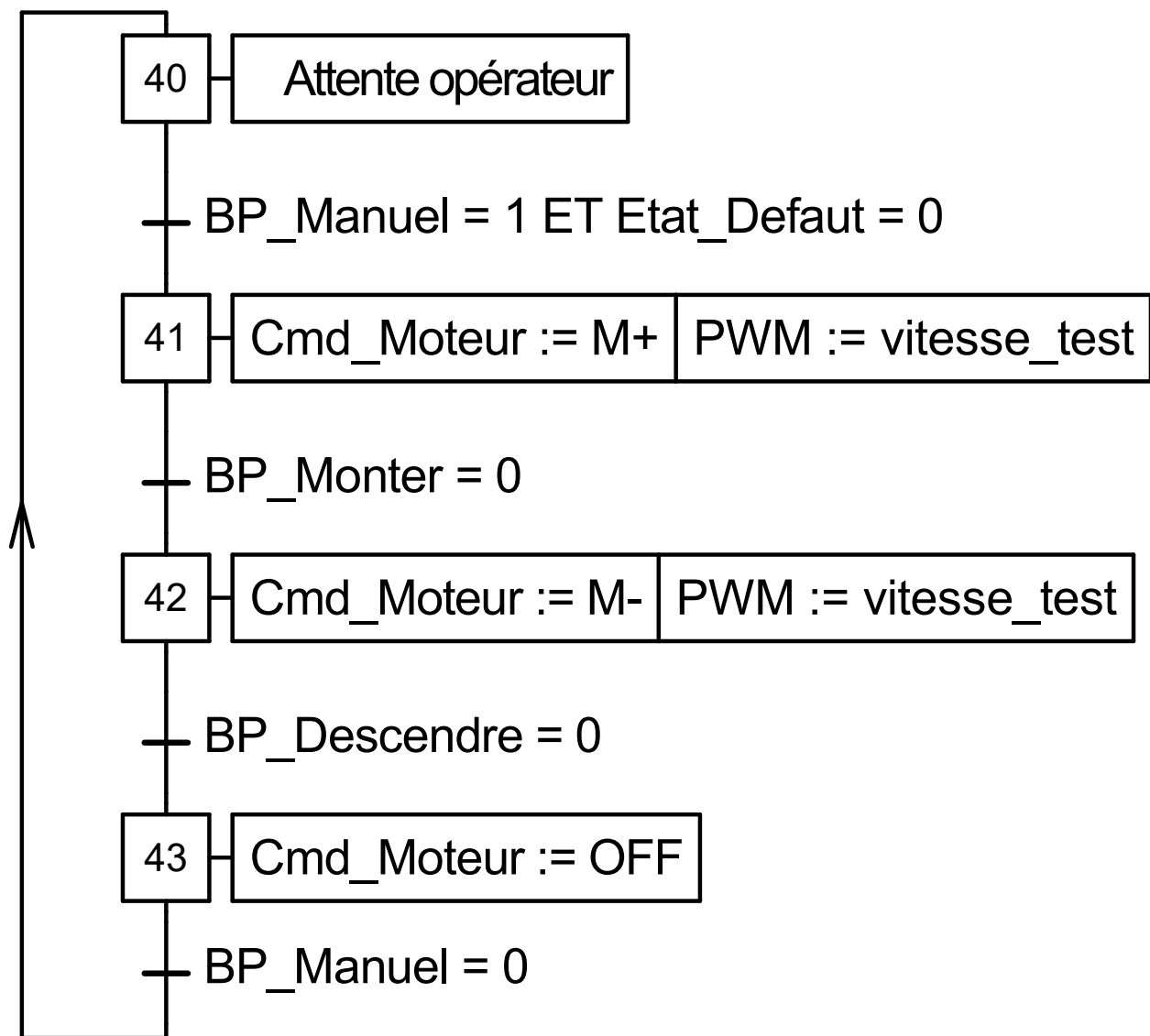
GRAFCET F1 : Cycle moteur automatique



GRAFCET F2 : Marche de préparation

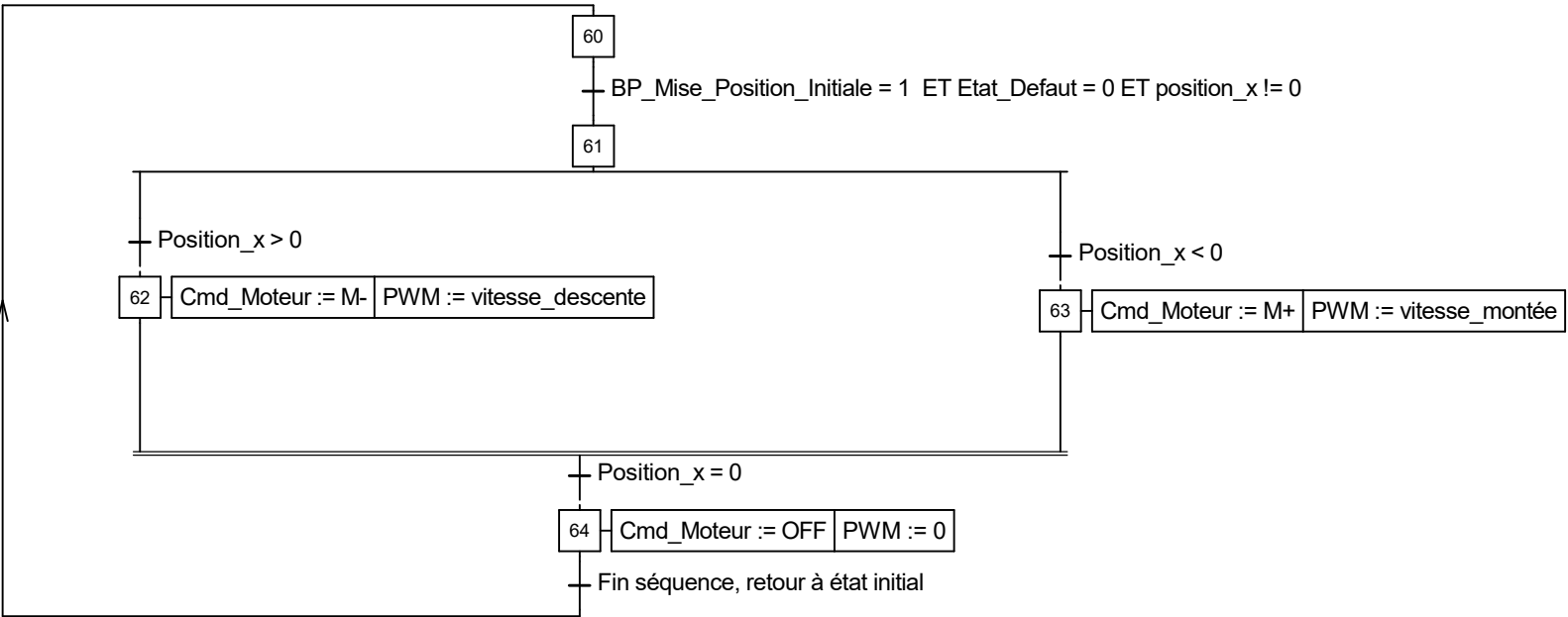


GRAFCET F4 : Vérification manuelle



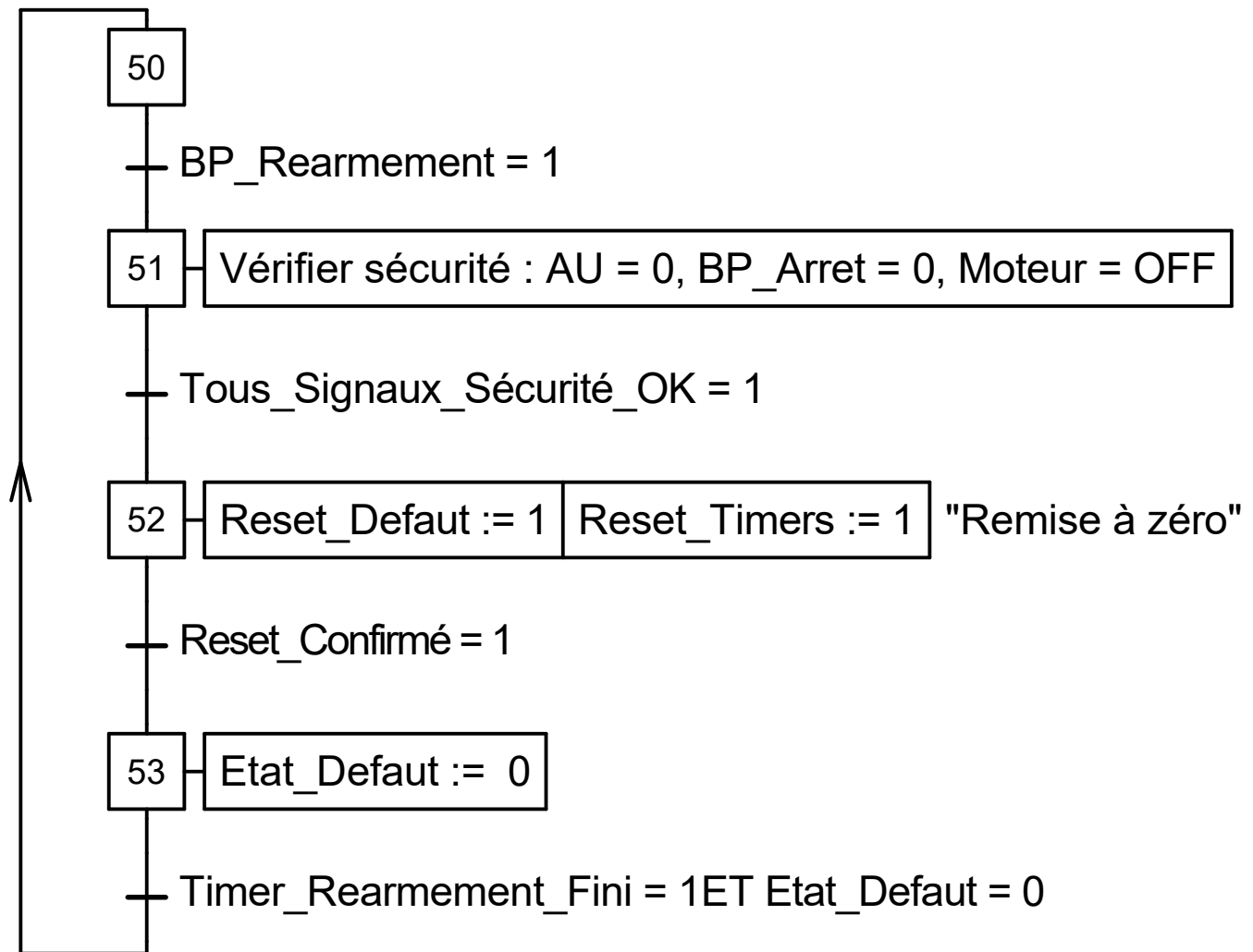
Annexe 5

GRAFCET A6 : Retour à position initiale

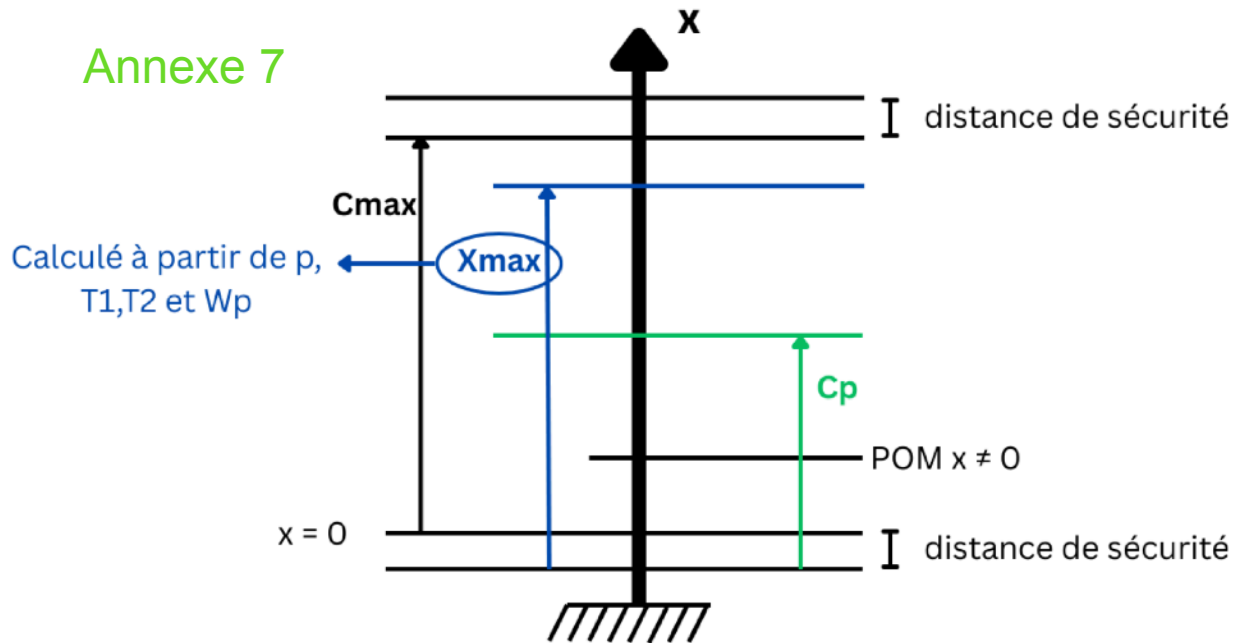


Annexe 6

GRAFCET A5 : Réarmement après défaut



Annexe 7



X_{max} : distance maximale qui va être calculée en fonction de p , T_1 , W_p

C_{max} : C'est la distance entre la position initiale ($x=0$) et la position finale/ maximale. W : C'est la vitesse de la palette en temps réelle.

C_p : C'est la position de la masse par rapport à la position initiale en temps réelle

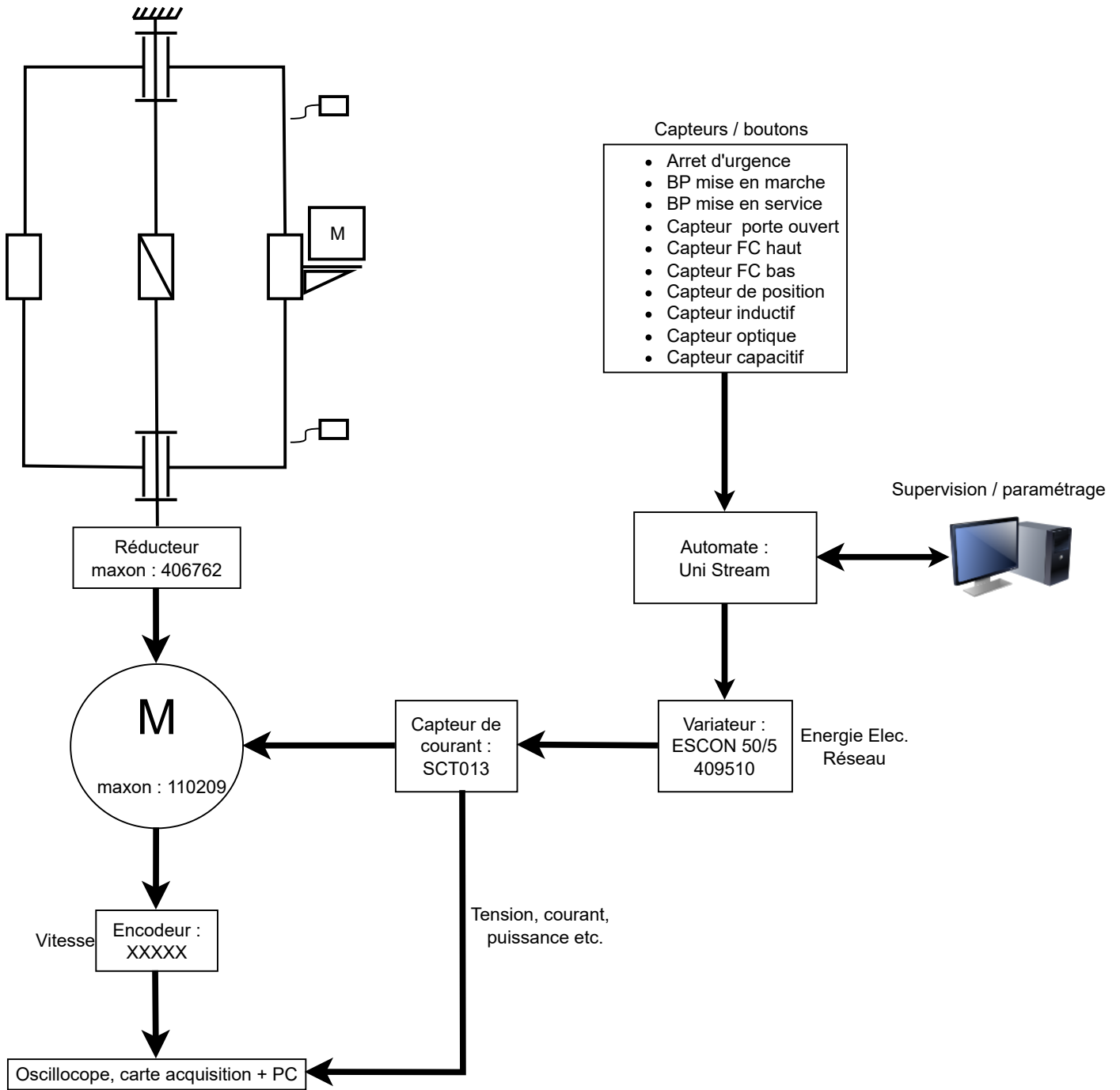
P : C'est la pente de la vitesse de la rotation du moteur

T_1 : C'est la durée que la palette reste à sa vitesse maximale.

T_2 : C'est la durée que la palette reste à une vitesse nulle.

W_p : Vitesse consigne à atteindre

POM : prise d'origine mesure



Annexe 9

IHM API, mode manuelle

Exit

Mode Manuelle

Cmax: m
W: tr/min
Cp: m
Wp: tr/min

W: tr/min

Appliquer

UP

DOWN

IHM API, mode automatique

Exit

Remarque: Xmax doit être inférieur à Cmax

Vitesse de rotation du moteur

The diagram illustrates the motor speed profile for automatic mode. It shows a sequence of phases: acceleration (p), constant speed (Wp) for duration T1, deceleration (p), dwell (T2), deceleration (p), dwell (T1), and acceleration (p) for negative speed (-Wp). The profile is bounded by dashed red lines at Wp and -Wp.

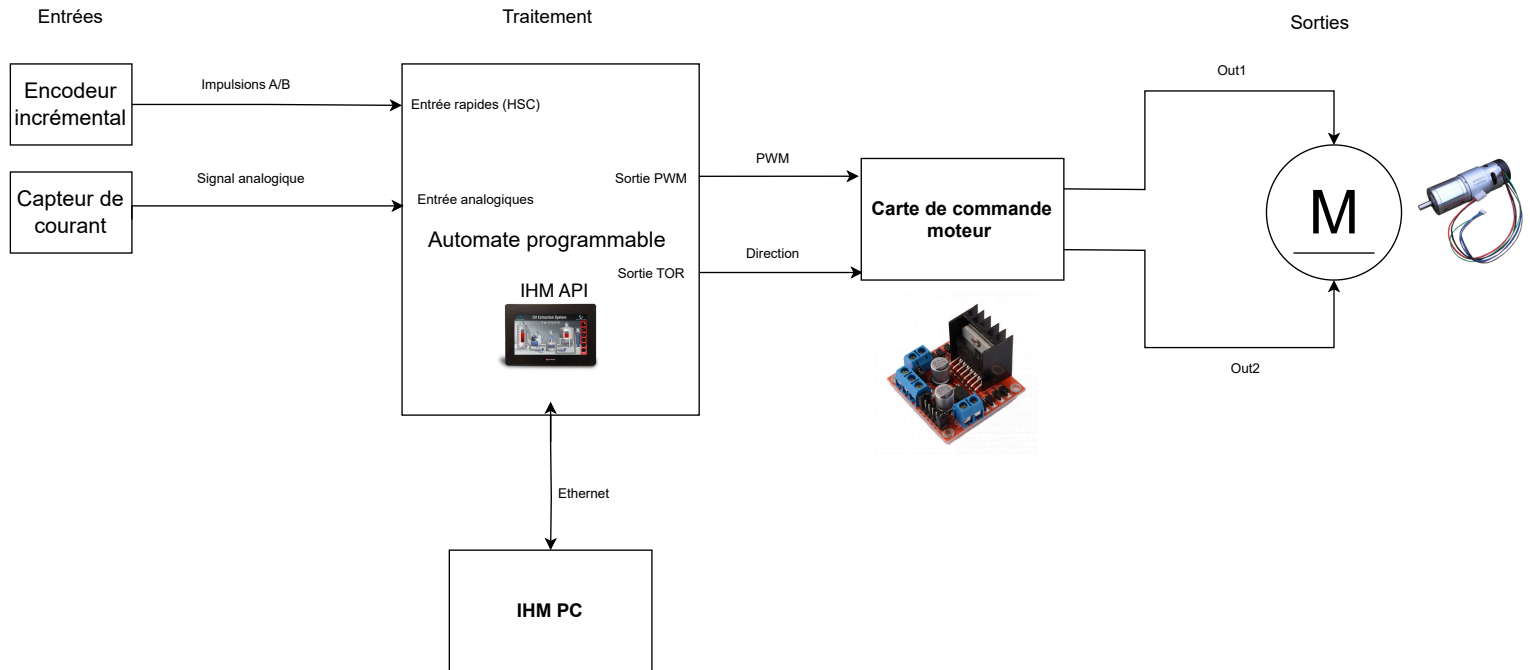
Xmax: m
Cmax: m
W: tr/min
Cp: m

p:
T1: s
T2: s
Wp: tr/min

Appliquer

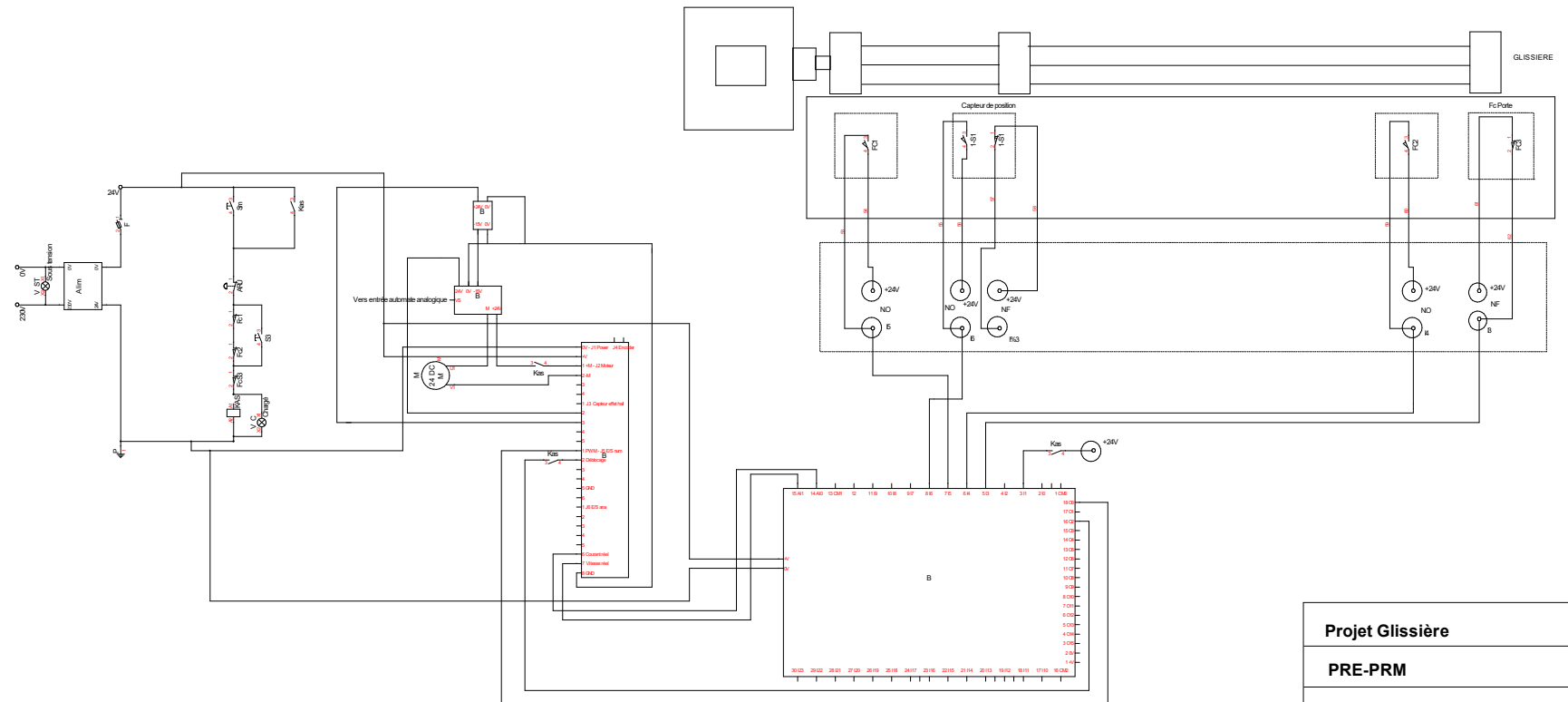
Annexe 10

Schéma fonctionnel global - Projet PER - Glissière motorisée



Annexe 11

Composante	Désignation	Commande en cours	En stock	Fournisseur	Rf
Sm	BP mise en charge				
FcS3	Captteur ouverture porte		X		
ARU	Arret d'urgence		X		
Fc1	Captteur fin de course droite glissière		X		
Fc2	Captteur fin de course gauche glissière		X		
1-S1	Captteur de position		X		
1-B2	Captteur Inductif		X		
1-B3	Captteur Capacitif				
1-B4	Captteur Optique				
S3	BP de remise en service de la glissière		X		
V ST	Voyant de mise sous tension du système		X		
V C	Voyant de mise en charge du système		X		
KAS	Relais de sécurité				
Kas	Contact relais de sécurité		X		
API	Automate programmables tout en un API+9HM+E/S	X		Unistream	US5-B10-T42
Module E/S analogiques				Unistream	UA-0402N
Variateur	Service constructeur pour moteur	X		Maxon Motor	409510
Encodeur	Codeur incrémentale	X		Maxon Motor	110520
Moteur	Moteur 24Vc 60W	X		Maxon Motor	353027
Alim	Alimentation 230V vers 24V	X		RS	400-7409



Projet Glissière

PRE-PRM

MARCHADOUR EUGENE