

**Jean-Rémy Dion**

***Rapport Personnel***

**DION Jean-Rémy**

**Tables des matières**

[I. Introduction 3](#_Toc128345971)

[A. Mise en place du poste de travail en fonction des tâches à effectuer 3](#_Toc128345972)

[B. Ressources nécessaires 3](#_Toc128345973)

[a. Logiciels : 3](#_Toc128345974)

[b. Matériel : 4](#_Toc128345975)

[II. Description des tâches à effectuer 4](#_Toc128345976)

[A. Décoder et interpréter les trames CAN (J1939) 4](#_Toc128345977)

[a. Utilisation de la norme J1939 pour les trames CAN 4](#_Toc128345978)

[B. Programmer des cycles de fonctionnement de l’émulateur en « Structured Text » 4](#_Toc128345979)

[C. Réaliser une interface WEB 4](#_Toc128345980)

[III. Spécifications techniques 4](#_Toc128345981)

[A. Langage de programmation et norme 4](#_Toc128345982)

[B. Fonctionnement de l’écran tactile FT 50 5](#_Toc128345983)

[IV. Analyse SysMl / UML 5](#_Toc128345984)

[A. Diagramme des exigences 5](#_Toc128345985)

[B. Diagramme de cas d’utilisation 7](#_Toc128345986)

# Introduction

## Mise en place du poste de travail en fonction des tâches à effectuer

Lors de la mise en place de notre projet, nous avons réalisé que nous avions besoin d'un poste de travail adapté aux tâches à effectuer pour travailler efficacement en équipe. Nous avons donc entamé une réflexion sur les différentes étapes du projet, les besoins spécifiques de chaque tâche et les outils et ressources dont nous avions besoin pour mener à bien notre projet.

Nous savions que nous allons travailler ensemble, mais que nous n’allions pas forcément avoir les mêmes équipements en fonctions de nos tâches.

Nous avons commencé par nous informer sur les différentes normes et standards utilisés dans le domaine de l'automatisation industrielle, tels que la norme J1939 pour les trames CAN. Nous avons également étudié les différents langages de programmation utilisés dans ce domaine, en particulier le Structured Text, qui est compatible avec le logiciel Automation Studio, et qui nous est demandé d’utiliser.

Une fois que nous avons identifié les différents outils et ressources nécessaires, nous avons mis en place notre poste de travail en conséquence. Nous avons ainsi pris des ordinateurs qui possédaient déjà le logiciel Automation Studio, puis nous avons installé des logiciels tel que Github Desktop, ou la librairie Office 365.

Au bout de deux à trois semaines où nous nous sommes principalement investit dans la recherche, documentation et rédaction des documents à faire, nous avons pris conscience, que notre poste de travail, n’était pas si bien adapté. Nous avons donc du tout enlever pour remettre tout en ordre pour qu’on puisse bien utiliser la maquette qui était seulement de mon côté. Nous avons donc du tout changer pour qu’elle se trouve au milieu.

Cette démarche nous a permis d'avoir un poste de travail adapté aux besoins spécifiques de notre projet et de travailler efficacement en équipe. Nous avons ainsi pu nous concentrer sur les tâches à effectuer en minimisant les problèmes techniques et en maximisant notre productivité.

## Ressources nécessaires

### Logiciels :

* Automation Studio : pour programmer l'émulateur et l'interface WEB en Structured Text
* Github Desktop : pour la gestion de version du code source
* Office 365 : pour la communication et la collaboration en équipe

### Matériel :

* Calculateur X90 : pour simuler le chariot élévateur
* Ecran tactile FT 50 : pour interagir avec l'émulateur du chariot élévateur
* IHM PP45 : pour la gestion de l'interface utilisateur (utilisé par Nelson Graveau)
* Gyrophare et feux arrière : pour ajouter des fonctionnalités intéressantes au projet.

# Description des tâches à effectuer

Réalisation : Une interface WEB associée au calculateur X90 pour le suivi, le contrôle et l’administration du système émulateur

## Décoder et interpréter les trames CAN (J1939)

Cette tâche consiste à comprendre les trames CAN norme J1939 pour le fonctionnement de l’émulateur, et analyser les trames et de comprendre les données qui y sont transmises.

## Utilisation de la norme J1939 pour les trames CAN

La norme J1939 est une norme utilisée pour les communications de données dans les systèmes de véhicules lourds, tels que les camions et les autobus. Elle est basée sur le protocole de communication CAN (Controller Area Network) et définit les messages, les paramètres et les protocoles de communication pour les différents composants du véhicule, tels que le moteur, le système de freinage et les systèmes de transmission. Dans notre projet, nous utilisons la norme J1939 pour les trames CAN pour communiquer avec l'émulateur du chariot élévateur.

## Programmer des cycles de fonctionnement de l’émulateur en « Structured Text »

Je dois programmer les cycles de fonctionnement de l’émulateur en utilisant le langage de programmation « Structured Text » répondant à la norme IEC 61131-3. Je dois également utiliser le logiciel de développement Automation Studio pour assurer la compatibilité.

## Réaliser une interface WEB

Je dois réaliser une interface WEB tactile pour interagir avec l’émulateur du chariot élévateur, programmer les pages WEB en utilisant les données transmises par les trames CAN que j’ai décryptées. Tout en assurant que l'interface est facile à utiliser et conviviale pour l'utilisateur final.

# Spécifications techniques

## Langage de programmation et norme

Le langage de programmation utilisé ici, et le Structured Text (ST) sous la norme IEC 61131-3.

Le ST est un langage de programmation utilisé dans les systèmes d'automatisation industrielle pour écrire des programmes qui contrôlent le fonctionnement des machines et des processus. Il est défini par la norme internationale IEC 61131-3 et est utilisé principalement dans les environnements de programmation de l'automatisation industrielle, tels qu’Automation Studio. Le langage est basé sur une syntaxe simple et compréhensible qui utilise des instructions structurées pour contrôler le flux de programme. Les instructions sont similaires à celles utilisées dans les langages de programmation comme le C. Le Structured Text est utilisé pour écrire des programmes qui gèrent des entrées/sorties, des temporisations, des boucles, des conditions et des fonctions mathématiques.

## Fonctionnement de l’écran tactile FT 50

Les écrans tactiles sont des périphériques d'entrée de données qui permettent aux utilisateurs d'interagir avec un appareil électronique en touchant l'écran. Leur fonctionnement repose sur deux technologies principales : résistive et capacitive.

Les écrans tactiles résistifs sont constitués de deux couches conductrices séparées par une fine couche d'air. Lorsqu'un utilisateur touche l'écran, les deux couches se touchent à cet endroit précis, créant ainsi un circuit électrique. Les coordonnées de l'endroit touché sont déterminées en mesurant la résistance électrique à l'aide d'un microcontrôleur, qui envoie ensuite ces informations à l'ordinateur ou au microcontrôleur de l'appareil électronique.

Les écrans tactiles capacitifs, quant à eux, sont constitués d'une couche conductrice transparente recouverte d'une fine couche isolante. Lorsqu'un utilisateur touche l'écran, son corps agit comme une charge électrique qui modifie le champ électrique de la couche conductrice. Cette modification est détectée par des capteurs disposés sur les bords de l'écran, qui envoient ensuite les coordonnées du point touché à l'ordinateur ou au microcontrôleur.

Les écrans tactiles modernes utilisent souvent des technologies capacitives projetées, qui fonctionnent en envoyant des signaux électriques à travers la surface de l'écran. Lorsque l'utilisateur touche l'écran, la charge électrique de son corps modifie les signaux électriques, permettant de déterminer la position du toucher.

En ce qui concerne l'écran tactile FT50, celui-ci est un écran tactile résistif qui fonctionne selon le même principe que les écrans tactiles résistifs classiques. La différence réside dans le fait que l'écran tactile FT50 est conçu pour être utilisé avec l'émulateur de chariot élévateur et dispose de fonctionnalités supplémentaires pour améliorer l'interaction avec l'émulateur. De plus, le FT50 est capable de transmettre les informations de position et de contact à l'émulateur via une connexion USB ou Ethernet, permettant ainsi de simuler le fonctionnement d'un écran tactile pour interagir avec l'interface utilisateur de l'émulateur.

# Tableau comparatif

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Désignation | Marque | Représentation | Avantage(s) | Inconvénient(s) |
|  |  |  | Caractéristiques | |
| Calculateur X90 | **B&R** |  | Processeur : ARM Cortex-A9-650 | 650MHZ RAM : 256 MB DDR3 SDRAM Application memory : 1 GB flash memory  Interfaces :   - 1x Ethernet  - USB  - 3x Bus CAN  - 1x POWERLINK | |
| Automates CompactLogix 5380 | **Rockwell Automation** | Vue de face de l'automate Allen-Bradley Compact GuardLogix SIL 3, catalogue 5069-L3100ERMS3 | 2x Ethernet  Haute vitesse de traitement  Plus large capacité de communication | Documentation difficile  Plus cher  Documentation qui n'est pas Open Source |
|  |  |  |  |  |
| Ecran tactile FT50 | **B&R** | Une image contenant texte, boîtier, cadre  Description générée automatiquement | Processeur : ARM Cortex-A9 4 cœurs | 800 MHz DRAM : 2GB Flash : 512 MB Ports :   - 1x Ethernet  - 1x USB 2.0 Diagonale : 15,6 Pouces Résolution : 1366 x 768 Couleurs : 16 millions Rétroéclairage : LED | |
| Basic Panel KTP700 | **Siemens** | SIMATIC Basic Panel Siemens KTP700 Basic PN - 6AV2123-2GB03-0AX0 | Moins cher  Plus grosse résolution d'écran  compatible avec plus de contrôleurs | Ecran tactile potentiellement moins précis  Mémoire moins importante (10Mo) |

# Analyse SysMl / UML

## Diagramme des exigences

A picture containing diagram

Description automatically generated

## Diagramme de cas d’utilisation

Diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Cas d'utilisation | Description | Acteur(s) concerné(s) | |
| **Responsable** | **Opérateurs** |
| Administrer le système | Le responsable technique est chargé d’effectuer les tests via un émulateur du chariot élévateur, permettant de vérifier le bon fonctionnement de l’IHM du Tableau de Bord de ce dernier. | ü |  |
| Superviser les données des tests | Le responsable technique pourra visualiser les résultats des tests (identification, temps, température, ...) sur :   * l’IHM du tableau de bord * l’interface WEB du banc de tests | ü |  |
| Paramétrer le système | Un responsable technique qualifié doit être en mesure de mettre en relation l’ordinateur de contrôle (Emulateur) avec le tableau de bord IHM du chariot. Il doit être en mesure de paramétrer les entrées/sorties du système pour répondre aux cycles de fonctionnement du système et à la sécurité des personnes sur le site. | ü |  |
| Calibrer les pupitres | Le responsable devra au préalable définir des configurations IHM en fonction de données telles que le type du chariot, la langue … et recalibrer l’IHM en cas de dysfonctionnement.  L’opérateur système vérifie le bon fonctionnement des indicateurs visuels de l’IHM du tableau de bord. | ü |  |
| Démarrer un cycle | Le responsable technique met en route le système grâce à l’interface de l’émulateur du chariot et visualise à l’aide d’un pupitre IHM, la mise en route du chariot.  Le système passe d’un état « repos » à un état « actif » et l’IHM renseigne l’opérateur de la situation par un message écrit  Ils pourront aussi choisir d’interrompre un processus s’ils jugent cela nécessaire, voir éventuellement d’y mettre fin. | ü | ü |
| Superviser le tableau de bord du chariot | Pendant un cycle, l’opérateur pourra suivre l’évolution des consignes en fonction des actions du chariot sur son tableau de bord (temps, température, masse à soulever, feux, clignotants, gyrophare, claxone, etc…) |  | ü |

# Conclusion

En conclusion, la mise en place d'un poste de travail adapté aux tâches à effectuer et la définition claire des spécifications techniques sont des éléments essentiels pour mener à bien un projet d'automatisation industrielle tel que le nôtre. En utilisant des normes et des langages de programmation standardisés, nous avons pu travailler efficacement en équipe et minimiser les problèmes techniques. Les ressources matérielles et logicielles nécessaires ont été identifiées et mises à disposition, ce qui nous a permis de programmer l'émulateur et l'interface WEB en Structured Text, de décoder et interpréter les trames CAN, et de réaliser une interface WEB tactile conviviale pour l'utilisateur final.