Une image contenant Police, texte, Graphique, logo

Description générée automatiquement Une image contenant Police, Graphique, graphisme, logo

Description générée automatiquement

**C.D.C : Cahier Des Charges**

**Projet Testeur Universel SNCF**

*Formation Informatique et Systèmes Intelligents Embarqués*

**Année 2024 – 2025**

Projet collectif 4A

Membres de l’équipe :

Paul HENRY – Chef de projet

Amandine LANGLAIS - Ingénieur qualité

Damien LORIGEON - Ingénieur électronique

Dorian BRISSON - Ingénieur développement

Bilel AAZZOUZ - Ingénieur développement

Client :

Frédéric CHAUVIN – Professeur de Polytech et employé de la SNCF

Auteur : Paul HENRY

Version 1.0 - 20/09/24

Version 1.1 - 27/09/24

Version 1.2 - 04/10/24

# Introduction :

Un train est composé de plusieurs automates, chacun pilotant divers éléments via un bus de données. Chaque automate possède des caractéristiques de communication spécifiques, telles que le protocole de communication, la vitesse de transmission, ou encore le type de bus (RS232, RS485), qui sont parfois inconnues.

Ce projet nous a été proposé par la SNCF, qui dispose déjà d'un outil permettant de scanner les bus de données et de lire les trames transcrites. Cet outil scanne les trames en connaissant toutes les caractéristiques de communication et affiche ces trames sous forme hexadécimale, un format qui n'est pas directement compréhensible.

# Problématique :

Ces données sont transmises sous forme de trames, qu'il peut être utile de décrire. Lors des dépannages, il devient difficile de déterminer la nature exacte des problèmes, car il est parfois impossible de connaître les caractéristiques de communication et, par conséquent, de décrypter les messages transitant sur le bus de données.

L’outil actuel permet de lire les trames de données en connaissant toutes les caractéristiques de la communication, telles que la vitesse ou le type de communication. Cependant, il ne peut pas les identifier automatiquement, ce qui serait utile lorsque ces informations ne sont pas connues à l’avance.

Bien que cet outil soit pratique, il affiche les trames lues en format hexadécimal, ce qui les rend difficiles à comprendre au premier coup d’œil.

# Objectif :

L’objectif de ce projet est de développer un outil capable d’analyser et de décrire les trames circulant sur un bus RS232 ou RS485, afin de les afficher de manière lisible, même sans connaître leur codage.

Cet outil devra offrir les fonctionnalités suivantes :

1. Analyse automatique des trames :

* Le logiciel analysera les trames sans nécessiter la connaissance préalable des caractéristiques de communication (protocole, vitesse, etc.).

1. Affichage des trames en clair :

* Les trames seront traduites et affichées sous forme de messages lisibles, interprétant les données transmises via le bus.

1. Personnalisation des paramètres :

* Si les paramètres de communication sont connus, ils pourront être configurés dans l'outil afin d'affiner l'analyse et la lecture des trames.

1. Affichage en temps réel des trames :

* Une **fenêtre principale** affichera en **temps réel** toutes les trames capturées, qu’elles soient normales ou qu’elles contiennent des erreurs.
* Si une trame ne peut pas être correctement lue ou analysée, un message d'erreur spécifique, tel que **"Lecture de la trame impossible"**, sera affiché à la place du contenu de la trame.

1. Fenêtre de diagnostic des messages :

* Une **seconde fenêtre dédiée** affichera uniquement les **messages analysés** avec un code couleur pour la compréhension des utilisateurs :
  + Les **messages de données normales** seront affichés en **vert**.
  + Les **messages d'erreurs spécifiques** (correspondant à des anomalies ou des alertes dans le système) seront affichés en **rouge**.

1. Support de plusieurs protocoles :

* Le logiciel devra être compatible avec divers protocoles de communication (Modbus, CAN, etc.) et être facilement extensible pour d'autres protocoles si nécessaire.

1. Historique et export des trames :

* L'outil devra permettre la sauvegarde des trames capturées et leur export dans des formats lisibles (CSV, fichier texte), pour analyse postérieure.

1. Gestion des erreurs de trame :

* En cas d'erreur dans la lecture d'une trame (trame incomplète, corrompue, etc.), un message explicatif sera affiché dans la fenêtre principale en temps réel, permettant ainsi aux utilisateurs de savoir pourquoi une trame n’a pas pu être interprétée.

1. Interface utilisateur intuitive :

* L'interface graphique du logiciel devra être simple et efficace. Elle permettra à l'utilisateur de visualiser en temps réel les trames, d'ajuster les paramètres de communication, de filtrer les trames affichées et d’accéder à une vue dédiée pour les diagnostics d'erreurs (messages en vert/rouge).

1. Affichage des courbes :

* Le logiciel doit pouvoir afficher les signaux logiques sous forme de **courbes graphiques**, représentant visuellement l’évolution des signaux capturés sur le bus.

1. Outil de recherche dans les trames :

* Une fonctionnalité de recherche doit permettre de **taper une adresse spécifique** pour filtrer les trames et **les masquer** selon des critères définis.
* Les trames valides seront **repérées avec un code couleur** (par exemple, vert pour les trames correctes) pour une meilleure lisibilité.

1. Création de nouveaux protocoles :

* L'outil devra inclure la capacité de **créer et intégrer de nouveaux protocoles** de communication en cas de besoin (dans le cas où de nombreux protocoles différents doivent être pris en charge).

# Contraintes :

* **Compatibilité** : Le logiciel devra être compatible avec Windows.
* **Fiabilité** : L'outil devra résister aux erreurs générées par des trames corrompues ou incomplètes.

# Livrables :

* Le logiciel fonctionnel avec toutes les fonctionnalités mentionnées ci-dessus.
* Le manuel d'utilisation pour les utilisateurs.
* Les sources du logiciel conçu.