



Table des matières

Objet	1
Prérequis	2
Paramétrage OpenRails.....	2
Fichier cabine	2
Arduino.....	2
Utilisation	3
Fenêtre « Connexion »	3
Fenêtre « Simulation »	4
Fenêtre « Trame série »	5
Fenêtre « Debug ».....	7
Fenêtre « Variables ».....	6
Fenêtre « Serial Monitor »	6

Objet

FS ORTS Link est un utilitaire permettant de lire les variables du jeu OpenRails Train Simulator (ORTS).

Pour plus d'information sur ORTS, rendez-vous sur : <https://www.openrails.org/>

FS ORTS Link est développé par FerroviSim dans le cadre de projets de simulateur, et mis à disposition gratuitement. Pour plus d'information sur FerroviSim, rendez-vous sur www.ferrovisim.fr.

FS ORTS Link est distribué sous la License GPL v3. Les termes du contrat de License sont disponibles dans le fichier dédié. En vertu de ces termes, ORTS Link est Open Source, et l'intégralité du code source est disponible sur le répertoire GitHub FerroviSim. Vous êtes libre d'apporter des modifications. Toute publication d'une version modifiée devra être Open Source et distribuée sous cette même licence.

Prérequis

Paramétrage OpenRails

FS ORTS Link Version 0.1 Beta a été conçu et testé avec Open Rails version 1.5.1.

Dans les options d'ORTS, il est nécessaire de choisir le port web qui sera utilisé pour la communication avec l'utilitaire. Par défaut, le port est « 2150 ». Il est tout à fait possible de garder ce port, et c'est ce qui sera fait pour la suite.

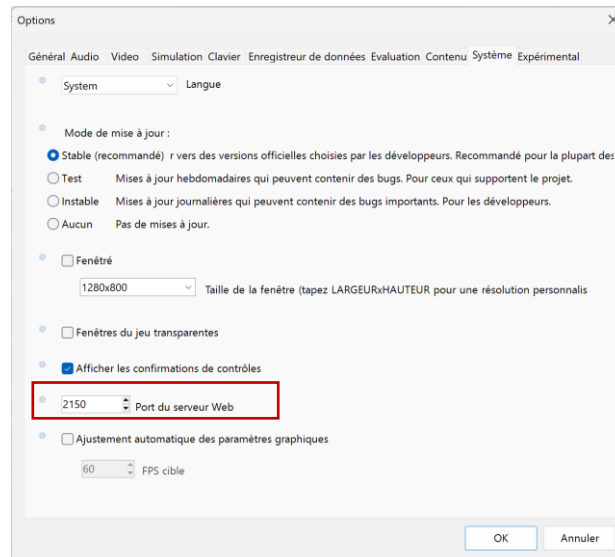


Figure 1 - Options d'Open Rails

Pour plus d'informations sur la fonction serveur web d'ORTS, rendez-vous sur <https://open-rails.readthedocs.io/en/unstable/software-platform.html#web-server>

Fichier cabine

L'API web d'Open Rails renvoie les variables qui sont renseignées dans le fichier cabview (.cvf) de l'engin utilisé dans la simulation. En principe, cela fonctionne avec n'importe quel fichier cabview, mais un fichier exemple optimisé pour l'utilisation avec ORST Link est fourni avec l'utilitaire : cabine.cvf

Ce fichier a été testé avec les BB26000 des Compagnons du Rail disponibles ici : <https://www.lescompagnonsdurail.fr/>

NB : ce fichier n'affiche aucune vue cabine, mais liste simplement des variables afin que celles-ci soit renvoyées par le Server Web d'ORTS.

Ce manuel suppose une connaissance des principes de fonctionnement d'Open Rails par l'utilisateur. En particulier, le rôle et contenu d'un fichier CVF.

Les ressources suivantes peuvent être utiles pour se documenter à ce sujet :

<https://open-rails.readthedocs.io/en/unstable/cabs.html>

<https://ajtrainsim.pierreg.org/cap3d/cabin/cabine.htm>

Arduino

Cet utilitaire a été testé pour fonctionner en liaison série avec une carte Arduino.

Important : Afin de ne pas surcharger la mémoire tampon du port série de la carte Arduino, les données ne sont pas envoyées en continu par l'utilitaire une fois la connexion établie. L'utilitaire n'envoie les données à la carte qu'après avoir reçu l'ordre de le faire. Cela se matérialise par la réception d'un caractère choisi. Par défaut, il s'agit du caractère « 7 ».

Dans le code Arduino, il est donc nécessaire d'insérer la ligne suivante : `Serial.print('7');` ;

Cette ligne devra être exécutée à chaque fois que la carte sera prête à recevoir une nouvelle trame de données (donc quand la trame précédemment reçue aura été traitée. Le choix du caractère se fait dans la fenêtre principale du programme (voir « Fenêtre connexion ci-après).

Pour la réception des données sur la carte Arduino, un article avait été rédigé. Le tout est disponible ici :

<https://ferrovisim.fr/2021/12/30/communication-serie-entre-python-et-arduino/>

Un programme Arduino prêt à l'emploi sera bientôt publié.

Utilisation

Fenêtre « Connexion »

Cette fenêtre permet de paramétrer d'une part la connexion avec ORTS et d'autre part la connexion avec la carte Arduino du simulateur.

Le port du serveur web renseigné doit être le même que dans ORTS.

Pour le choix du port série, tous les ports disponibles seront listés dans le menu déroulé. Il suffit de choisir celui de la carte Arduino connectée (ou tout autre périphérique série).

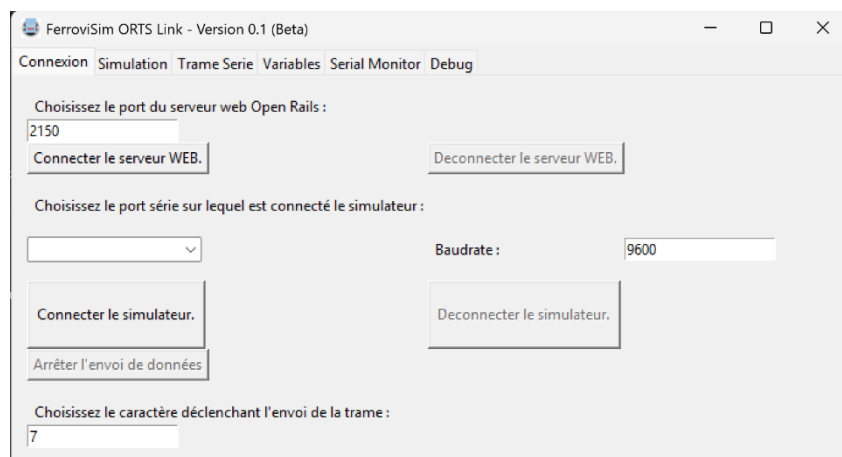


Figure 2 - Fenêtre connexion au lancement de l'utilitaire

Une fois le port du serveur web renseigné, cliquer sur « **Connecter le serveur WEB** ». Une fenêtre pop-up indique si la connexion a réussi ou non. Si la connexion échoue, assurez-vous qu'OpenRails est bien lancé avec le bon port. Le bouton « **Déconnecter le serveur WEB** » permet comme son nom l'indique d'interrompre cette connexion.

Une fois le port série de la carte Arduino renseigné, cliquer sur « **Connecter le simulateur** ». Une fenêtre pop-up indique si la connexion a réussi ou non. Si la connexion échoue, assurez-vous qu'une carte est bien connectée au PC, et que le port choisi est le bon. Vérifiez également que aucun autre programme (par exemple le moniteur série de l'IDE Arduino) n'utilise pas déjà la connexion série.

Une fois le port Série connecté, il est possible d'interrompre l'envoi de trames de données en utilisant le bouton « Arrêter l'envoi de données ». Cela n'interrompt pas pour autant la liaison série. Pour interrompre la liaison série, il est nécessaire d'utiliser le bouton « **Déconnecter le simulateur** ».

Le programme ne déclenche l'envoi de la trame qu'à la réception d'un caractère choisi sur le port série. Par défaut, la réception du caractère « 7 » déclenche cet envoi. Le caractère déclencheur peut être modifié dans la case correspondante.

Fenêtre « Simulation »

Cette fenêtre permet un affichage visuel d'un certain nombre de variables, au travers de jauges représentant de manière réaliste l'équipement unifié du matériel roulant français des années 60 à 1990.

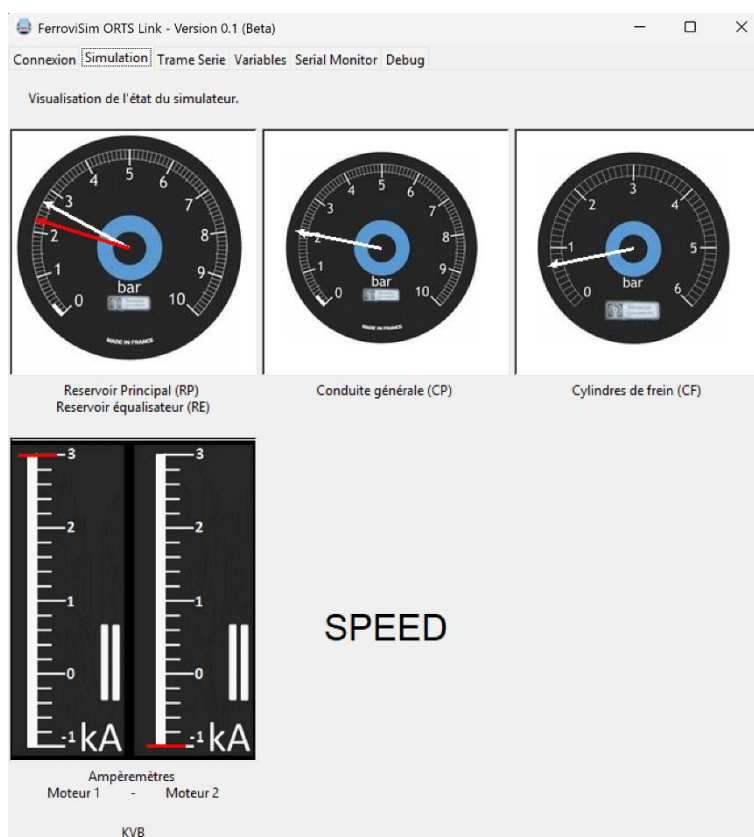


Figure 3 - Fenêtre simulation

Le bon fonctionnement de cette fenêtre est conditionné à la présence dans le fichier CVF des variables suivantes :

- BRAKE_CYL (cylindres de frein)
- BRAKE_PIPE (conduite générale)
- EQ_RES (réservoir égaliseur)
- MAIN_RES (réservoir principal/conduite principale)
- LOAD_METER (ampèremètre)
- SPEEDOMETER (indicateur de vitesse)
- ACCELEROMETER (accéléromètre)
- ORTS_TCS42 (pour l'affichage KVB, avec usage des scripts TCS France)
- ORTS_TCS43 (pour l'affichage KVB, avec usage des scripts TCS France)
- ORTS_TCS44 (pour l'affichage KVB, avec usage des scripts TCS France)
- ORTS_TCS45 (pour l'affichage KVB, avec usage des scripts TCS France)

Ces variables sont incluses dans le fichier CVF exemple fourni avec l'utilitaire.

Fenêtre « Trame série »

Cette fenêtre permet de définir les variables qui seront envoyées à la carte Arduino.

Le format de la trame envoyé à la carte est fixe : il s'agit d'une suite de variables, chaque variable étant identifiée par une lettre, sa valeur est un nombre entier, et les variables envoyées sont séparées par le caractère « / ».

La valeur numérique de la variable est obtenue par le serveur web qui lit les valeurs renvoyées par Open Rails. Afin de n'envoyer à l'Arduino que des valeurs numériques entières, il est proposé d'appliquer un ratio à la valeur décimale renvoyée par OpenRails. L'outil n'envoiera ensuite à l'Arduino que la partie entière de la valeur obtenue.

Par exemple, pour envoyer la pression dans la conduite générale avec une précision d'un décibar (0,1 bar), il sera nécessaire d'appliquer un ratio x10. Ainsi, une pression de 3,5 bars sera envoyée avec la valeur 35. En choisissant l'identificateur « G », cela donnera la trame « G35/ ».

Dans l'exemple ci-dessous, une trame a été constituée avec une douzaine de variables envoyées simultanément à l'Arduino.

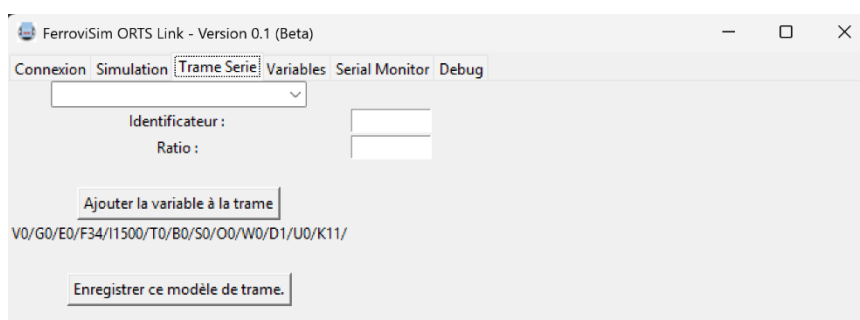


Figure 4 - Fenêtre Trame Série

Pour ajouter une variable à la trame, il suffit de la sélectionner dans le menu déroulant parmi les variables lues (c'est-à-dire celles qui sont définies dans le fichier CVF), de choisir un identificateur, un ratio, puis de cliquer sur « **Ajouter la variable à la trame** ». La liste des variables, avec les identificateurs et ratios attachés est sauvegardée, peut être enregistrée dans un fichier « trame.csv » qui s'enregistre automatiquement dans le même dossier que le programme, par appui sur le bouton « **Enregistrer ce modèle de trame** ».

Ci-dessous, un exemple de ce fichier avec les variables visibles sur la figure ci-dessus.

	A	B	C
1	SPEEDOMETER	V	1
2	BRAKE_PIPE	G	10
3	EQ_RES	E	10
4	BRAKE_CYL	F	10
5	AMMETER	I	1
6	THROTTLE	T	100
7	DYNAMIC_BRAKE	B	100
8	ORTS_TCS33	S	1
9	ORTS_TCS38	O	1
10	WHEELSLIP	W	1
11	ORTS_CIRCUIT_BREAKER_OPEN	D	1
12	LINE_VOLTAGE	U	1700

Figure 5 - Fichier csv d'enregistrement des paramètres de la trame

Notez que pour des besoins particuliers de votre système, le ratio peut ne pas être un multiple de 10.

Fenêtre « Variables »

Cette fenêtre permet d'avoir un aperçu brut et en temps réel de toutes les variables lues sur le serveur web d'Open Rails.

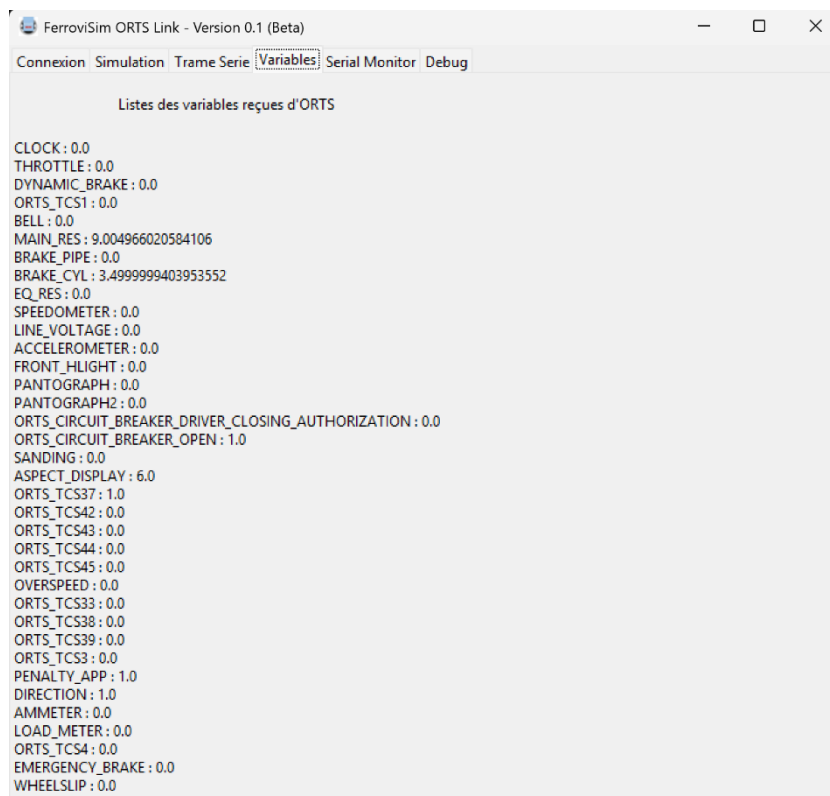


Figure 6 - Fenêtre Variables

Fenêtre « Serial Monitor »

Cette fenêtre fonctionne sur le même principe que le moniteur série de l'IDE Arduino. Elle permet d'envoyer ou de recevoir des messages série de la carte Arduino, sans avoir à interrompre la liaison série avec l'utilitaire et donc le flux de données entre Open Rails et la carte. Il est possible d'interrompre temporairement l'envoi de données automatique par le bouton « **Arrêter l'envoi de données** ».

La partie blanche (notée « 1 » sur la figure ci-dessous) affiche en temps réel les messages série reçus de la carte Arduino.

Pour envoyer un message, il faut le taper dans la bande blanche (notée « 2 ») et cliquer sur « **Envoi** ».

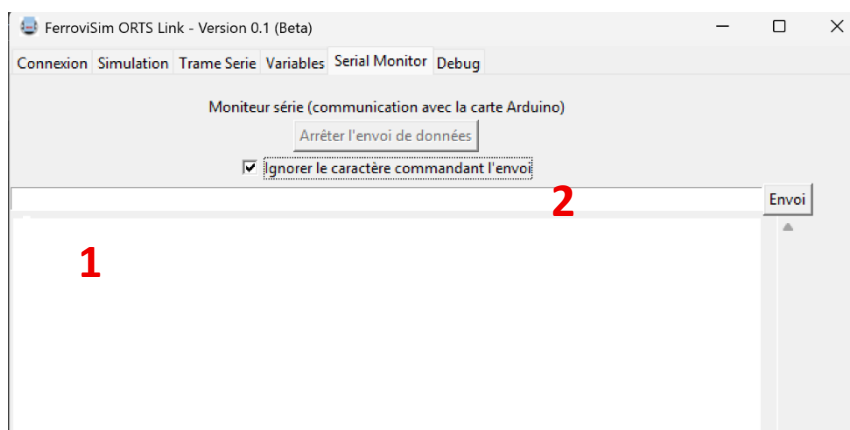


Figure 7 - Fenêtre Serial Monitor

Fenêtre « Debug »

Cet onglet est réservé à des ajouts futurs. Pour l'instant il permet simplement d'ouvrir une fenêtre de console permettant au développeur d'avoir plus d'information sur l'exécution du programme. Attention, une fois une console ouverte, sa fermeture par la croix rouge entraîne l'arrêt du programme. Pour fermer uniquement la console, il faut simplement décocher la case.