

Ce document produit par **Gérald Litzistorf en mai 2020** est destiné à un public restreint de **hobbyist** désireux de comprendre les limites et exigences de la programmation des trains à l'échelle H0; c'est-à-dire 87 fois plus petits que les modèles réels.

Dans l'idéal, un train réel ou miniature ne risque **aucune collision** s'il respecte la signalisation et que les aiguilles (**turn** par la suite) traversées sont dans la bonne position !

Ce document synthétise 5 ans de joies ... et de souffrances :

- Avril 2015 Découverte du projet Railuino → <https://wiki.mchobby.be/index.php?title=Railuino>
- Jan 2016 Utilisation de la carte Arduino Due
- Avril 2017 Publication du projet Raildue → http://gelit.ch/Train/Raildue_F.pdf
- Mai 2020 Publication de ce document qui prouve l'intérêt de parler directement aux locomotives & turns

§1 Contexte & Motivation

Lors de la période tristement célèbre du **Covid-19**, j'ai décidé de m'attaquer à la paroi Nord de l'Eiger ! Paroi suisse mythique qui me fait aussitôt penser à l'extraterrestre Ueli Steck, que je salue !!!

Comme fan de trains Märklin, qui a eu la chance d'avoir un père qui lui a acheté son premier train vers 10 ans, j'ai eu beaucoup de plaisir à développer du logiciel à partir du projet Railuino.

J'ai vite compris les limites du processeur 8 bit – 16 MHz de la carte Arduino Uno pour passer à BEAUCOUP plus costaud avec le processeur SAM3X (32 bit – 84 MHz) présent sur la carte Arduino Due.

De façon très simplifiée, la puissance de calcul est multipliée par un facteur 21 (84 MHz – 32 bit / 16 MHz – 8 bit)

Selon https://en.wikipedia.org/wiki/Instructions_per_second, le gain est de 25 !

Cet excellent processeur offre des performances voisines du premier Pentium (1994) que nous utilisions avec Win 95.

Pendant 2 ans, j'ai tenté de contrôler le plus de trains depuis mes logiciels et la **Gleisbox Märklin** !

- **Arduino Due** gère le contrôle de 1 à 12 trains, de 27 **décodeurs** de turn et de 67 détections S88 → <http://gelit.ch/Train/heli3.pdf>
- PC sous Windows 10 sert à "debugger" et à fournir un **écran** pour les explications données aux visiteurs
- La boîte de connexion (**Märklin Gleisbox**) convertit les commandes en signaux compréhensibles par les décodeurs présents dans chaque locomotive et turn



Windows 10



Arduino Due



BOX Märklin



Turn Decoder

Je ne vais pas tenté de décrire tous les trésors d'ingéniosité dont j'ai dû faire preuve pour parler avec cette boîte noire ! Cette Gleisbox est un module vendu par Märklin (comme Windows par Microsoft)

Vous payez mais ne connaissez pas précisément son mode de fonctionnement puisqu'il est gardé secret !

Heureusement, Märklin a produit ce document http://streaming.maerklin.de/public-media/cs2/cs2CAN-Protokoll-2_0.pdf, permettant à des éditeurs de logiciels de lui envoyer des commandes

Son défaut majeur est d'ignorer des commandes lorsque Madame travaille !

- Impossible d'activer rapidement les Leds de 10 locomotives → il faut attendre
- Impossible d'arrêter 10 trains rapidement lors d'un arrêt d'urgence → il faut attendre
- Impossible d'exécuter une commande après un ordre donné à un turn → il faut attendre

L'exercice est particulièrement critique lorsque mon programme donne l'ordre de positionner 4 turns pour le prochain trajet :

- Impossible de freiner ou stopper un autre train pendant $4 \times 250 \text{ ms} = 1 \text{ seconde}$

En conclusion : produire ma propre "Gleisbox" afin d'avoir un lien DIRECT (= une chaîne dont je maîtrise son ensemble) fut un rêve qui a pu se réaliser grâce au Covid-19

§2 Objectif

Mon idéal du train est très particulier !

Règle 1 **Aucun décor** ; j'en ai réalisé un magnifique lors de mon adolescence mais qu'elle tristesse de le voir terminé

Règle 2 5 trains en **mouvement simultané** c'est bien,
10 trains en **mouvement simultané** c'est mieux,
20 trains en **mouvement simultané** pour 2021, ça devient envisageable !

...
Sans collision et **avec des parcours aléatoires !!!**

Règle 3 **Montrer à mes invité-e-s** un affichage facilement compréhensible pour la personne qui n'a jamais travaillé au CFF

Le cahier des charges d'un **BON système de supervision des trains est donc de garantir** :

- Un comportement déterministe → Voir **§6** avec l'affichage de **métriques**
Aucun code caché → Voir **§6** avec l'affichage des temps de réponse de Windows 10
Aucun inter blocage possible car une seule boucle
Pas de pause avec `delay()`
- L'exécution des commandes (Speed – Function – Direction – Turn) dans un temps négligeable !

La solution proposée dans la démo, utilisant la technologie PWM (Pulse Width Modulation) disponible avec la carte Arduino Due exécute chaque commande dans **un temps maximum de 13 ms !!!**

Ce temps est en fait compris entre 1 et 13 ms, comme le montre le logiciel de démo ; ce qui conduit à une valeur moyenne de 6-7 ms chez moi avec 12 trains en circulation

QUI FAIT MIEUX ?

Je ne pense pas qu'il soit nécessaire d'utiliser 40 ordinateurs pour gérer 1000 trains comme certains le font

Pour info :

- Le logiciel Direct proposé comprend 367 lignes de code en C
- Mon logiciel actuel de supervision pour Arduino Due comprend 1590 lignes en C
- Son complément sous Windows 10 pour l'affichage comprend 460 lignes en XAML & 1423 lignes en C#

§3 Signal électrique

Dans la longue histoire Märklin, j'utilise leur technologie **MM2 (Motorola Märklin Protocol Version 2)**, introduite en **1994** et expliquerai ultérieurement ses avantages et inconvénients par rapport à MFX.

https://en.wikipedia.org/wiki/M%C3%A4rklin_Digital

Heureusement que je suis tombé sur cette excellente publication du prof. Andrea Scorzoni, que je remercie !!!

<http://spazioinwind.libero.it/scorzoni/motorola.htm>

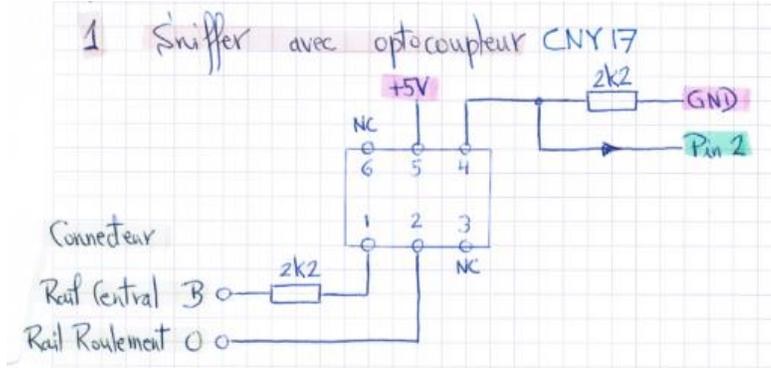
C'est elle qui m'a donné le courage de démarrer.

Dans un monde opaque, comme celui de la BlackBox Märklin, l'usage du **Reverse Engineering** fait partie des méthodes à utiliser lorsque toutes les autres ont échoué.

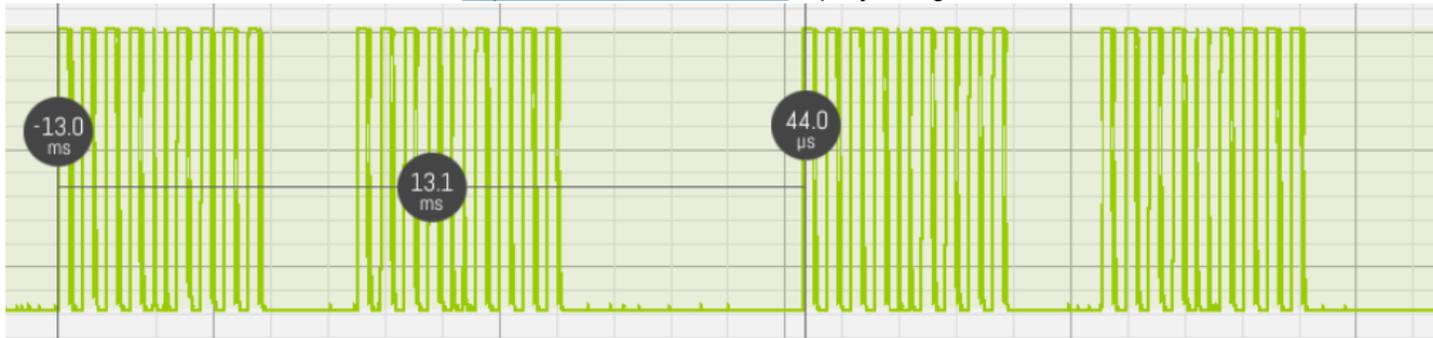
Ci-dessous le signal électrique produit par cette Gleisbox

IMPORTANT : le signal sur le rail alterne une différence de potentiel de +18- et -18+

Je préfère utiliser un optocoupleur et mesurer sur la Pin2 – Arduino par rapport au GND de l'Arduino



Mesure avec cet excellent oscillo USB <https://www.lab-nation.com/>, que j'ai largement amorti dans cette aventure !



Selon le descriptif du composant utilisé produit à l'origine par Motorola

<https://www.nxp.com/docs/en/data-sheet/MC145026.pdf>, le paquet de 18 bit doit être répété pour que le décodeur accepte la commande; d'où le terme de Doppel Packet

Détail des 13 ms mesurées ci-dessus :

- Premier paquet de 18 bit / Durée = 3.6 ms → voir ligne 316-336
Scorzoni parle plus précisément d'une durée égale à 208 μs avec un débit binaire = 38400 bit/s
- Pause de 1.7 ms entre les 2 paquets → voir ligne 337-340
- Second paquet, qui doit être identique au premier, pour que le décodeur prenne en compte la commande
- Pause de 4.1 ms avant l'émission du prochain paquet → voir ligne 293-316

Pour plus de détail de la partie PWM : code source à partir de la ligne 244

Ne pas oublier que ce signal, une fois redressé, fournira l'énergie nécessaire pour faire avancer une locomotive ou déplacer un turn.

Scorzoni parle d'état Idle. Il s'agit en fait du paquet 0 présent dans les lignes 110-113 du code

Il est important que l'adresse utilisée ne corresponde à aucun décodeur si bien que je l'ai baptisée "Virtual Decoder"

Sur le modèle de la BlackBox, je répète inlassablement la dernière commande et le paquet Idle.

Actuellement, ma version personnelle reproduit le même type de cycle pour 12 locomotives.

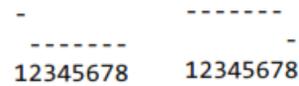
La puissance de cette technologie PWM garantit qu'une nouvelle commande sera prise en charge dans un temps maximum égal au 13 ms de la mesure ci-dessus.

Elle est de plus intégrée dans le chip SAM3X de la carte Arduino Due; ce qui ne pénalise en rien les performances du CPU !

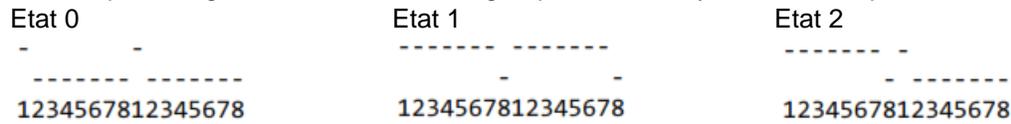
<https://store.arduino.cc/arduino-due>

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-11057-32-bit-Cortex-M3-Microcontroller-SAM3X-SAM3A_Datasheet.pdf

Dans son document, Scorzoni explique très bien comment produire l'état 0 ou l'état 1



La suite devient plus indigeste avec la notion de groupe de 2 bits ayant 3 valeurs possibles → voir Tri ligne 221



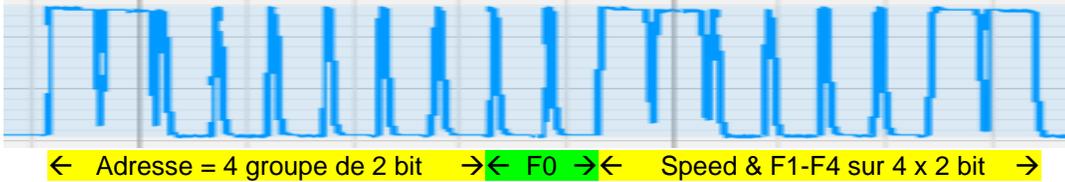
Cette méthode de codage est due au caractère **Simplex** de la transmission

Un décodeur ne fait que recevoir des commandes mais il ne parle jamais

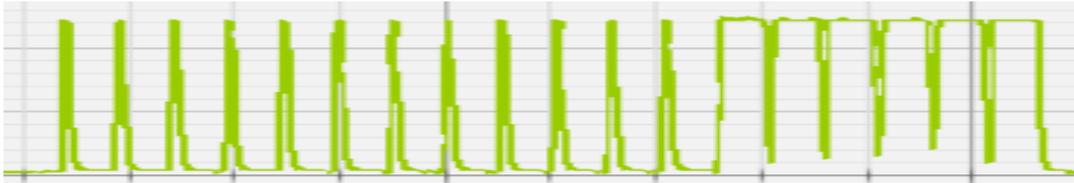
En cas de perturbation, il va ignorer toutes les combinaisons qui s'écartent de ces 3 valeurs

Ci-dessous un paquet destiné au décodeur 2

→ voir SetAdr ligne 229



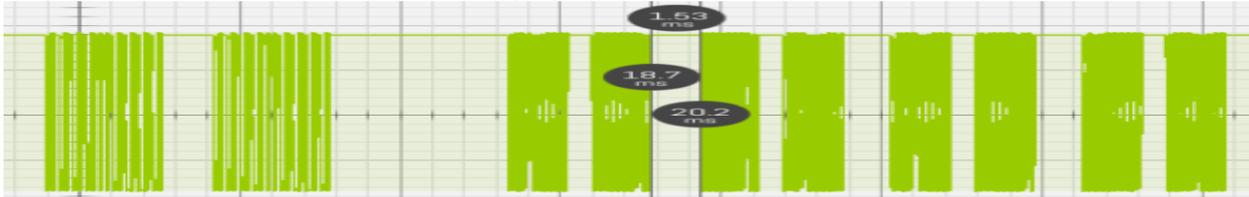
Voir le code source (177-218) pour comprendre la limite de la plage d'adresse fixée à 320



Le décodeur de turn fonctionne sur le même principe mais avec une durée de bit réduite à 104 µs

Il est intéressant de constater qu'il suffit d'ajuster un simple réseau RC sur les décodeurs 145027 ou 145028.

Paquet destiné à une locomotive suivi d'un paquet destiné à un turn



Ces mesures faites sur la BlackBox montrent qu'elle a tendance à en faire beaucoup plus que nécessaire

4 doubles paquets alors qu'un seul paquet double suffit pour commander un turn

J'ai même observé qu'elle envoie 10 doubles paquets identiques lorsque l'utilisateur appuie sur une touche de fonction

Un simple paquet double est suffisant ce que montre le logiciel de démo

§4 La bonne gestion des turns est critique

Parler à une locomotive pour lui allumer les Leds, lui donner l'ordre d'avancer, de ralentir, de s'arrêter ou de changer de direction n'est pas compliqué avec le logiciel de démo.

C'est déjà plus difficile avec la BlackBox; notamment pour le changement de direction ...

MAIS, les problèmes arrivent le plus souvent avec les turns !!!

Le monde du train miniature travaille souvent avec une tension de 18 Volt

Qui dit tension pense aussitôt au courant : **quel est le courant consommé par ma locomotive ou mon turn ?**

J'ai cherché sur internet mais je n'ai rien trouvé de précis !!!

... ma Mobile Station Märklin 60653 affiche bien une valeur en A (=Ampère ?) dans le menu Valeurs de Consommation – voie ppale ?!

Pour avoir commandé des relais dans ma jeunesse, j'ai appris à me méfier d'eux !

Le relais est un équipement électromécanique, très utilisé dans les centraux téléphoniques avant l'arrivée du transistor. Il est particulièrement important de permettre à l'énergie accumulée dans les bobines de pouvoir se dissiper (en chaleur) après la commutation.

Cette vidéo montre comment court-circuiter les diodes grillées qui avaient été mal dimensionnées

<https://www.youtube.com/watch?v=Pkt5h4b0WGE>

Durant la période 2016-2017, j'ai dû ramener au magasin 50% de moteur défectueux

http://www.maerklin.ch/fr/produits/details.html?art_nr=74491

Un moteur de turn modèle 74491 consomme énormément de courant !!!

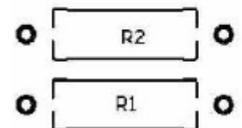


Vertical : max = **1.5 A**

Horizontal : 30 ms

Je conseille vivement d'utiliser un **booster** comme ce modèle <https://www.ldt-infocenter.com/dokuwiki/doku.php?id=de:db-4>

La mesure ci-dessus a été faite avec mon oscilloscope aux bornes des résistances R1-R2 qui sont mises en // pour produire : 0,225 V pour 4,5 A



Je tiens à remercier Peter Littfinski qui a toujours répondu de façon précise à mes questions !

Imaginons une séquence dans laquelle le logiciel doit commander 3 turns

Le courant total consommé pourrait atteindre $3 \times 1.5 = 4.5$ A

Il est donc vital d'éviter cette superposition en ajoutant une pause.

La BlackBox le fait très bien en ajoutant une pause de l'ordre de 200 ms MAIS aussi pour les autres commandes comme freiner un train !

Nous entrons là dans un vrai cauchemar :

- Impossible de stopper un train dans cette pause de 3 turn x 200 ms pour les novices
- Impossible de stopper un train dans cette pause de 200 ms pour les malins

Afin aussi de soulager les oreilles (= éviter l'effet mitrailleuse), mon logiciel personnel fixe cette pause à 500 ms.

Elle ne sert qu'à retarder la prochaine commande de turn et ne produit aucun retard pour les commandes de locomotives.

Je peux en tout temps ralentir ou freiner mes trains !

De plus les turns sont des équipements à la durée de vie limitée.

Mon logiciel personnel est capable de ne pas exécuter une commande sur le turn 9 lorsqu'il sait que la position du turn est la bonne.

Le PC offre un complément parfait à la carte Arduino, outre son écran que je peux facilement déporter sur une TV ou un écran géant, il me permet de sauvegarder la position des trains entre 2 séances de jeu.

§5 Machine d'états et gestion des exceptions

La programmation d'un train consiste à contrôler son déplacement de la position A à la position B.

Dans mon projet http://gelit.ch/Train/Raildue_F.pdf , j'explique comment créer des Cantons (segments)

La maquette actuelle comprend 54 Cantons ; certains servent à stationner des trains :

- 9 au niveau supérieur
- 4 par étage pour garer un maximum de 12 trains
- 3 dans l'hélicoïdal 1 et 6 dans l'hélicoïdal 2

Les autres, plus courts, garantissent la bonne circulation des trains entre les cantons de stationnement.

Algorithme du déplacement des trains :

```
If LocActive (User can active-deactivate each Loc)
  If LocStopped (Loc has reached position B)
    Search Next Path(s) → based on Canton Status
    If No → Wait
    If N Path → Choose randomly → next
    If 1 Path → Reserve Canton – Cmd to Turn(s) – Cmd Speed to Loc
```

La **machine d'états** (State Machine) est le mécanisme parfait pour faciliter cette programmation.

Elle est omniprésente dans la gestion des protocoles de communication

Voir cette gestion d'alarme avec les protocoles SMTP et NTP → http://gelit.ch/MKR/MKR_WIFI_ALARM.pdf

Les personnes chargées de la spécification des protocoles utilisent ou ont utilisé le **langage naturel** (texte en anglais, ...) compréhensible par tous.

Le monde internet regorge de ce type de mauvaise spécification.

La problématique devient évidente lorsque 2 développeurs comprennent et implémentent la même spécification **ambiguë** de 2 manières différentes !

Ci-dessous un exemple de précieux document utilisant un formalisme simple et puissant dès la page 18

https://www.etsi.org/deliver/etsi_i_ets/300100_300199/30010202/01_60/ets_30010202e01p.pdf

§6 Validation & critères de qualité

Quiconque a écrit du logiciel critique au temps (je déteste le terme temps réel), sait que les tests sont primordiaux.

Dans une démarche basée sur le cycle en V → <https://www.supinfo.com/articles/single/6278-cycle-v>, le processus de validation est permanent.

Alors que le monde professionnel travaille plutôt avec un modèle à 4 yeux (la personne qui développe ne teste pas); le *hobbyist*, que je suis, doit se débrouiller seul.

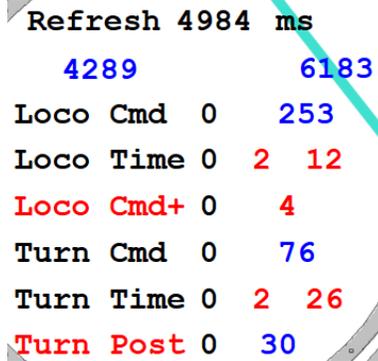
Les tests unitaires sur des parties de code à l'aide de `Serial.print` sont utiles mais pas suffisants

La qualité d'un logiciel se juge sur la durée car je suis incapable de pouvoir affirmer que ma nouvelle version de logiciel est exempte de bug.

Elle repose sur des **métriques** me permettant de juger la bonne santé de l'ensemble du système :

- Combien de commande ont été exécutées lors de la période précédente de mesure égale à 5 secondes ?
- Quel a été le temps moyen pour exécuter ces commandes. Est-il bien compris entre 1 – 13 ms ?
- Face à une montée en charge, mon système dispose d'un FIFO d'une profondeur égale à 45 commandes. Est-ce utile, surdimensionné, ... insuffisant ?
- Combien de commandes destinées aux locomotives sont exécutées alors que les commandes turn doivent attendre ?

La figure ci-dessous est le tableau de bord (du malade même s'il est en bonne santé) :



Refresh 4984 ms
4289 6183
Loco Cmd 0 253
Loco Time 0 2 12
Loco Cmd+ 0 4
Turn Cmd 0 76
Turn Time 0 2 26
Turn Post 0 30

- Un mécanisme Maître-Esclave assure le dialogue entre PC=Maître et Arduino=Esclave
Le Maître est sensé envoyé une commande chaque 5 seconde
Vous pouvez observer que la notion de temps sous Windows 10 est très médiocre puisqu'il fluctue entre 4.3 et 6.2 seconde !!! → **Je ne confierai jamais mes trains à Windows**

Dans la séquence de 15 minutes avec 10 locomotives actives :

- Loco Cmd → 253 commandes ont été envoyées aux locomotives
- Loco Time → Le temps d'exécution de ces commandes est compris entre 2 et 12 ms
- Loco Cmd+ → Parmi ces 243 commandes, 4 ont été possibles durant la pause d'un turn
- Turn Cmd → 76 commandes destinées au turn
- Turn Time → Temps d'exécution compris entre 2 et 26 ms étant donné qu'en cas de conflit entre commande Loc & Turn, la priorité est donnée à la conduite des locomotives et la commande turn doit être retardée d'un paquet de 13 ms
- Turn Postponed → Parmi les 76 commandes Turn, 30 ont dû être retardées

Avez-vous une telle remontée d'informations sur votre logiciel ?

§7 Utiliser aujourd'hui une technologie de 1994 est ringard alors que MFX existe !

Oui pour les utilisateurs non techniques incapables de changer l'adresse d'une locomotive
C'est du Plug&Play cher à Microsoft avec USB : tu poses ta locomotive sur le rail et ça marche; c'est magique !

Dans mon cas qui n'est pas intéressé de voir le cockpit de ma Gothard sur l'écran, **je suis limité aux fonctions F0-F4**
http://www.maerklin.ch/fr/produits/details.html?art_nr=39250
Heureusement, je peux activer le générateur de fumée, le bruit du moteur et le sifflet

Avec le modèle http://www.maerklin.ch/fr/produits/details.html?art_nr=39567, Märklin a été généreux en lui octroyant 2 adresses MM2; ce qui offre 4 fonctions supplémentaires que j'ai testées

Au besoin, je pourrais enregistrer un son produit par mes beaux wagons du Simplon Orient Express
http://www.maerklin.ch/fr/produits/details.html?art_nr=42790 puis les envoyer sur des haut-parleurs de qualité par mon PC.

Malheureusement tout est en allemand !

<https://static.maerklin.de/damcontent/a4/a1/a4a1dc145d4fc625742bbe81912d38f71584349661.pdf>

Il y a même de sympathique producteur de décodeur qui pense à moi et qui propose 16 fonctions pour cette superbe locomotive → <https://shop.zugkraft-stucki.ch/de/11812/bemo-rhb-ge-4-4-iii-651-glacier-on-tour-h0-normalspur-3l-ws-mit-esu-loksound-m4-v4-0-decoder>

... et demain, je peux être tenté de vaincre un autre sommet grâce à

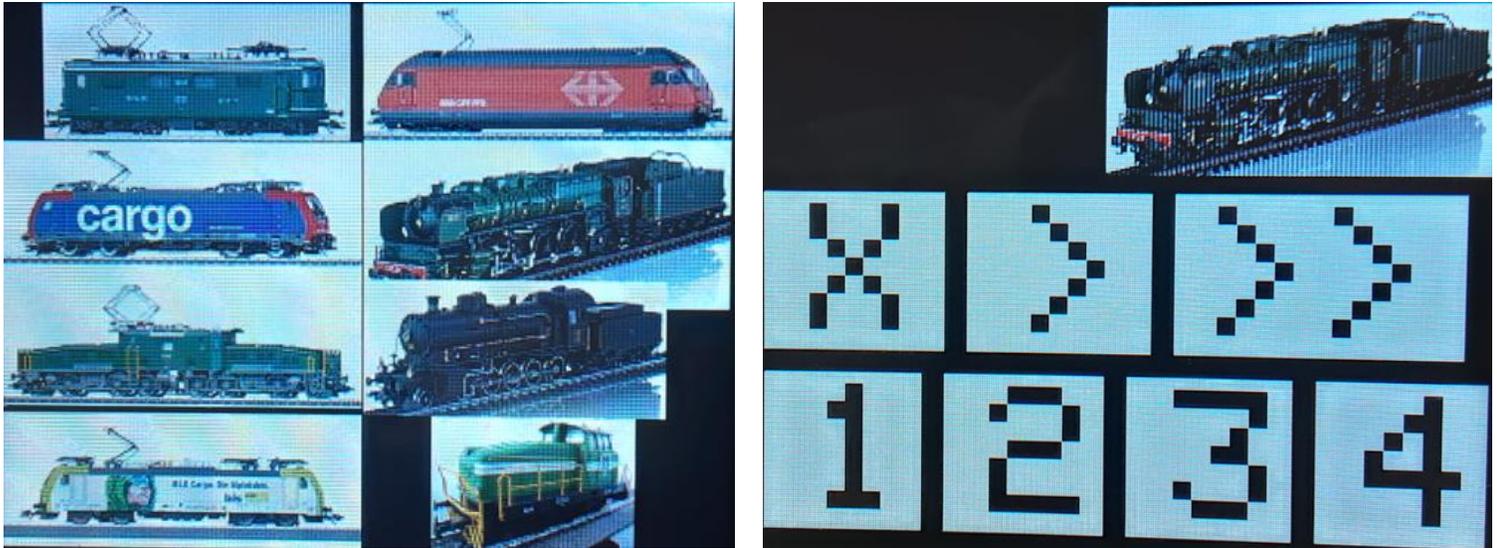
<http://www.skrauss.de/modellbahn/Schienenformat.pdf> Bravo à **Stefan Krauss** et à son équipe !!!

§8 Nouvelle génération

J'ai la chance d'avoir un petit-fils de 3½ ans qui ADORE les trains
Il est très curieux et me demandait souvent, avant le Covid-19, d'éviter les collisions.

Désormais je peux affirmer que la prochaine collision n'est pas prête d'arriver !!!

Ci-dessous sa commande basée sur un Arduino Due et un écran TFT-Touch



Il apprend ainsi à lire

§9 Conclusion & Remerciements

Ces 5 ans de bonheur m'ont rappelé l'époque du RNIS où j'ai eu l'occasion de développer le module ISDN de l'appareil Ares-C de Kudelski → https://www.radiomuseum.org/r/kudelski_portable_recorder_nagra_ares_c.html
Les étapes ont été nombreuses depuis le premier prototype wrappé jusqu'à l'adaptation du logiciel pour la Suisse, pour l'Europe avec la version EuroISDN puis les Etats-Unis pour la coupe du monde de football de 1994.

Cette réalisation illustre le terme d'émulation. Un **émulateur**, dans le jargon du prof. IT, est capable de faire croire à son interlocuteur (= le décodeur) qu'il reçoit des commandes de la BlackBox.

J'ai un peu réfléchi à une extension qui me permettrait de doubler le nombre de trains en mouvement

<http://gelit.ch/Train/sim4.pdf>

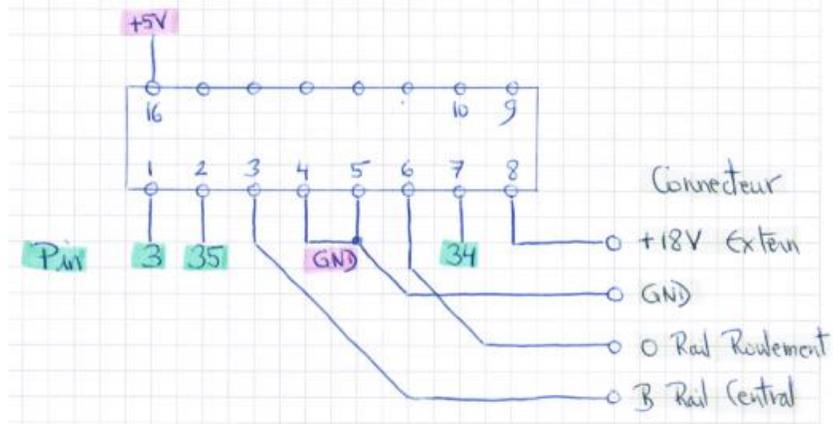
Les conditions sont aujourd'hui réunies pour m'y lancer; ce que je n'aurai pas aimé tenter via la Gleisbox

Je tiens à remercier :

- **Christian Abegg** qui a toujours une solution pertinente à mes questions comme http://gelit.ch/td/vmware/Abegg_P.pdf
- **Google Search**, qui m'accompagne chaque jour depuis très très très longtemps !
Je privilège souvent https://www.google.com/advanced_search

§10 H-Bridge

Il est nécessaire d'ajouter ce composant L293D à la carte Arduino Due



Les signaux mesurés sur la Pin 34 correspondent à ceux de ce document avec la tension limitée à 3.3 V par l'Arduino Due

H-Bridge → <https://www.distrelec.ch/fr/circuit-pour-pilote-de-moteur-dip-2a-st-l293d/p/17313224>

Possibilité d'utiliser l'adaptateur Märklin 230V-18V http://www.maerklin.ch/fr/produits/details.html?art_nr=66361 fourni dans les boîtes de démarrage comme http://www.maerklin.ch/fr/produits/details.html?art_nr=29484

§11 Schéma bloc

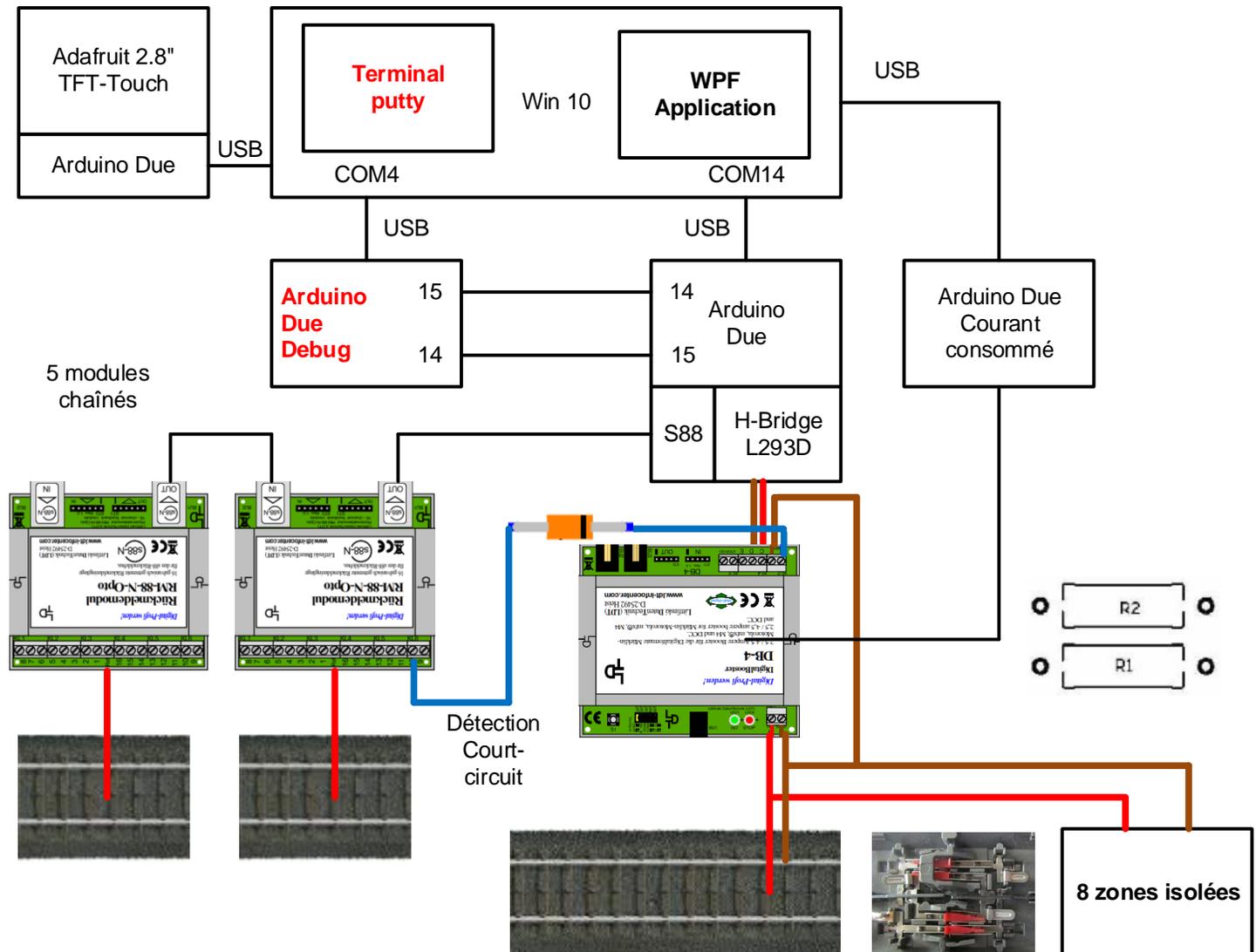
Ci-dessous la vue d'ensemble des composants matériels (hardware).

Séparer son réseau en zones isolées est une obligation car les nombreux contacts entre rails sont sources de chute de tension.

A titre d'exemple, je créé 1 zone pour 2 tours de mes hélicoïdaux; ce qui correspond à une distance linéaire d'environ 6 m

Arduino Due, de par son rapport prix/performance, peut servir à une multitude de fonctions :

- Commande pour le petit-fils mentionné au §8
- Mesure du courant consommé
- ... je vous laisse imaginer vos besoins
- **Terminal de debug**



Pour être franc, il a fallu que je sois dans une situation critique pendant 2 jours pour que je me décide à utiliser ce **terminal de debug** ; dont le retour sur investissement est phénoménal !

Les applications critiques au temps, comme le contrôle de 1 – 10 – 100 trains ou la gestion des protocoles de communication comme ISDN ou internet, exigent des **methodologies fiables de validation** :

- Le **processus de conformité** (validation) peut exiger la réussite d'une série de tests comme ceux produits par l'ETSI à l'époque pour le RNIS. J'ai réussi ces tests mais je savais qu'ils n'étaient pas complètement exhaustifs.
- **Votre voiture possède un tableau de bord; chaque véhicule habité (train, fusée, sous-marin, ...) aussi ! Avez-vous, pour vos très chers trains, le tableau de bord décrit au § 6 ?**
- **Il offre un confort correctif à celui qui développe et préventif à celui qui aime jouer avec ses trains !**