Le bus S88

Pour éviter d'avoir des grappes de fils qui partent de l'Arduino vers les capteurs il devient nécéssaire de concentrer les informations délivrées par les capteurs avec le moins de fils possible. Pour envoyer vers le PC les informations sur la détection des trains et autres détecteurs, je choisis le bus de rétro-signalisation **S88-N** pour sa simplicité qui garantit robustesse et fiabilité à l'utilisation. Adieu aux grappes de fils qui partent de l'Arduino, un simple câble RJ45 suffit comme pour Ethernet.

Pour les utilisateurs de Centrale DCC Z21 de Roco, il existe un convertisseur S88-N vers R-Bus sur **Digi-cz.** Vous trouverez des explications sur la réalisation de ce type d'interface sur le site de **Jindra Fucik** et un **schéma** de son montage. Actionnez le traducteur de vore navigateur pour lire ces pages.

Description

Les infos des capteurs seront rassemblées sur place par une carte de rétro-signalisation puis transmises par le câble RJ45 vers la carte suivante et ainsi de suite le long du réseau jusqu'à la Centrale DCC. Chaque bit lu sur ce bus représente l'état d'un capteur qui est connecté sur un module de rétro-signalisation (carte électronique) dont le role est de rassembler ces informations pour les transmettre vers la Centrale DCC. Le bus **S88** est un bus série qui fonctionne avec l'envoi d'une horloge asynchrone depuis un seul microprocesseur (maitre) et chaque module (esclave) connecté en série (daisy chain) lui renvoie un bit d'information à chaque transition positive de l'horloge. Dans ma Centrale DCC, l'Arduino Mega qui génère le DCC s'acquitera de cette tâche sans supplément. Un téléchargement de la nouvelle version **DCCppS88** suffit.

Ces modules **S88-N** (esclaves) ne nécessitent pas de microprocesseur car un simple registre à décalage s'acquitte de la tache. Ils sont donc

plus rapides et plus réactifs qu'un microprocesseur sans avoir besoin d'être programmés. Le schéma très simple a été établi depuis plusieurs décénnies et a fait ses preuves depuis longtemps. Ceux-ci sont télé-alimentés en 5VDC (ou 12VDC) donc pas besoin d'alimenter séparément les cartes de rétro-signalisation. Ils sont interconnectés par chainage série (Daisy chain) avec des câbles RJ45 à 8-pin, ce qui permet d'augmenter la distance entre les modules car l'immunité au bruit est bien meilleure.

Chaque entrée capteur sera connectée à la sortie des détecteurs de train décrits dans le menu "Les détecteurs". Cependant vous pouvez y connecter d'autres contacts, des ILS, des sorties de détecteurs infrarouge, etc..., mais en restant isolé du DCC. J'ai choisi de séparer physiquement la détection des trains et le module **S88-N** parce que la détection des trains doit se faire au plus proche des rails (courant fort) alors que la transmission d'informations (courant faible) vers le module **S88-N** esclave permet un éloignement plus important. De plus mes modules détecteurs de trains sont isolés du DCC, filtrent les mauvais contacts rails/roues et retardent la libération des cantons de 1 à 2 secondes.

Sur mon réseau il s'agit de renvoyer 96 bits plusieurs fois par seconde vers un PC, ce qui est infinitésimal pour ce dernier. La norme definit la limite à 512 capteurs pour un bus **S88-N**, soit 32 modules à 16 entrées. Lire 512 bits s'effectue par une lecture du bus en moins de 100 millisecondes, donc 10 lectures par seconde. Par comparaison, un train qui roule à 97 km/h (60 mph) à l'échelle 1/87e parcourt 3 cm (1.2") en 1/10e de seconde.

Après récupération des données par mon module maitre **S88**, il lui faut moins de 15 millisecondes pour transmettre un paquet de 512 bits vers un PC sur un bus USB à 115200 bauds ou par une liaison Ethernet. Depuis l'écran de contrôle, le module maitre sera paramétré en fonction du nombre de modules esclaves connectés sur les bus **S88** car il est inutile de lire plus de capteurs qu'il est nécessaire.

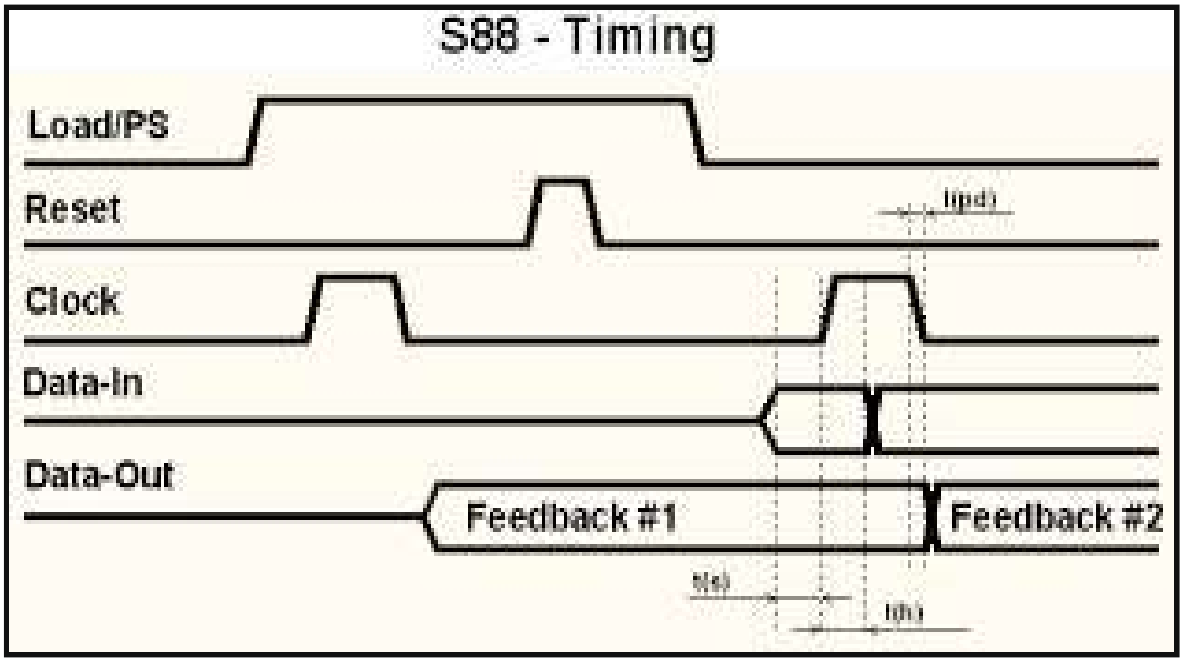
- Assignement des signaux du bus S88-N :

S88-N sur cable RJ45			
Pin	**Signal**	**T-568A**	**T-568B**

1	+5V/+12V	Blanc/vert	Blanc/orange
2	DATA IN/OUT	Vert	Orange
3	GND	Blanc/orange	Blanc/vert
4	CLK	Bleu	Bleu
5	GND	Blanc/bleu	Blanc/bleu
6	LOAD/PS	Orange	Vert
7	RESET	Blanc/marron	Blanc/marron
8	+5V/+12V	Marron	Marron

Je réutilise la pin 8 du câble RJ45 pour renforcer l'alimentation électrique du bus **S88-N** et diminuer les pertes occasionnées par le chainage des modules.

- Chronogramme des signaux du bus S88 :



Ce timing correspond parfaitement au fonctionnement de l'électronique des modules S88.

Ce timing collecte les données **S88** des dernières 100 millisecondes tout en filtrant les mauvais contacts rail/roues, si vous utilisez le design de la carte ci-dessous. Ajoutez 15 ms de temps de transfert USB vers le PC, plus le temps qu'il prendra pour traiter ces données. Avec Ethernet ou le WiFi il

faut ajouter les temps de latence du réseau.

Le Maitre S88-N

J'ai intégré le programme maitre **S88** dans la bibliothèque **DCCppS88** décrit dans le menu "Alimentation DCC", onglet "Le générateur DCCpp". En conséquence j'utilise un SEUL Arduino MEGA pour gérer tout mon réseau DCC : un seul et unique !
La rétro-signalisation est totalement integrée dans la bibliothèque **DCCppS88** sans ralentir l'Arduino. Une trame de 512 bits maxi est lue en permanence par l'Arduino sur le bus **S88-N** en 100 ms. Le PC qui contrôle le réseau reçoit jusqu'à 10 fois par seconde les paquets de data recueillies sur les 2 bus **S88** par l'Arduino qui lui transmet en moins de 15ms.

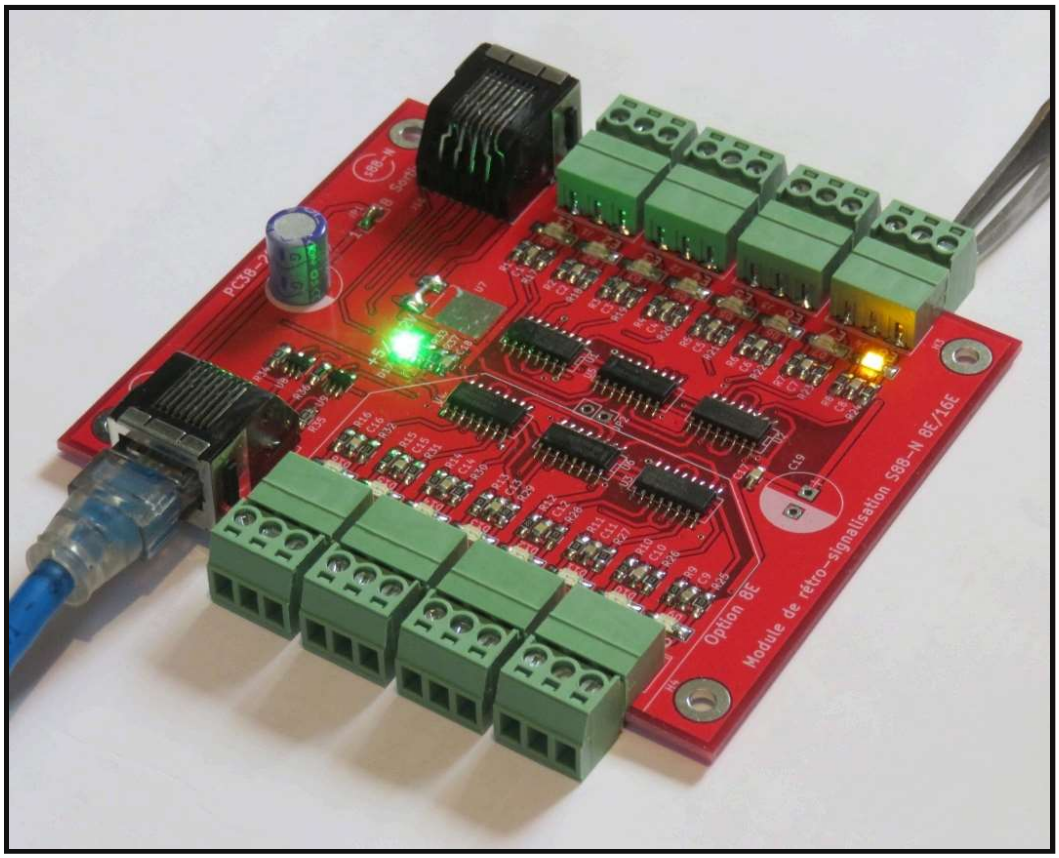
Cette fonction utilise 5 pins de l'Arduino : **Reset, Load/PS, Clock, DataL et DataR**. Cependant selon les besoins et le manque de pins libres, il est possible de libérer une pin en ne créant pas DataR, ce qui limitera le bus à 256 capteurs.

La commande DCC utilisée est :

```
‹Y N F› avec N = groupes de 8 capteurs à lire sans dépasser 64 (2x32x8=512)
(pour calculer le nombre N, prenez le plus grand nombre de groupes de 8
capteurs que vous avez connectés sur l'un des 2 connecteurs du bus S88 et
multipliez par 2. Exemple 3 à gauche et 1 à droite : N = 3 x 2 = 6)

F = 0 pour une réponse en binaire (msb first),
La réponse sera ‹y
0000101000010100011101000O.....................................00›

F = 1 pour une réponse en hexadecimal 3 fois plus rapide (MSB first ou LSB
first)
La réponse sera ‹y 0A0147405801CE..40›

F = 2 pour une réponse en hexadecimal pur, 7 fois plus rapide. (MSB first)
La réponse sera ‹y XXXXX......›

F = 3 pour une réponse du type SENSOR utilisé par JMRI.
La réponse sera ‹Q ID› si le capteur est actif (1), ‹q ID› si le capteur
est inactif (0). ID est le muméro du capteur.

En cas de commande non reconnue et selon le type d'erreur, les réponses
sont :
‹x Bad Argument count›
‹x Bad Argument value›
‹x S88 disconnected›

Note : si le bus S88 est totalement déconnecté, tous les bits lus sont à 1.
```

Note : si vous lisez des modules esclaves qui ne sont pas connectés à des capteurs, les bits correspondants seront à 0.

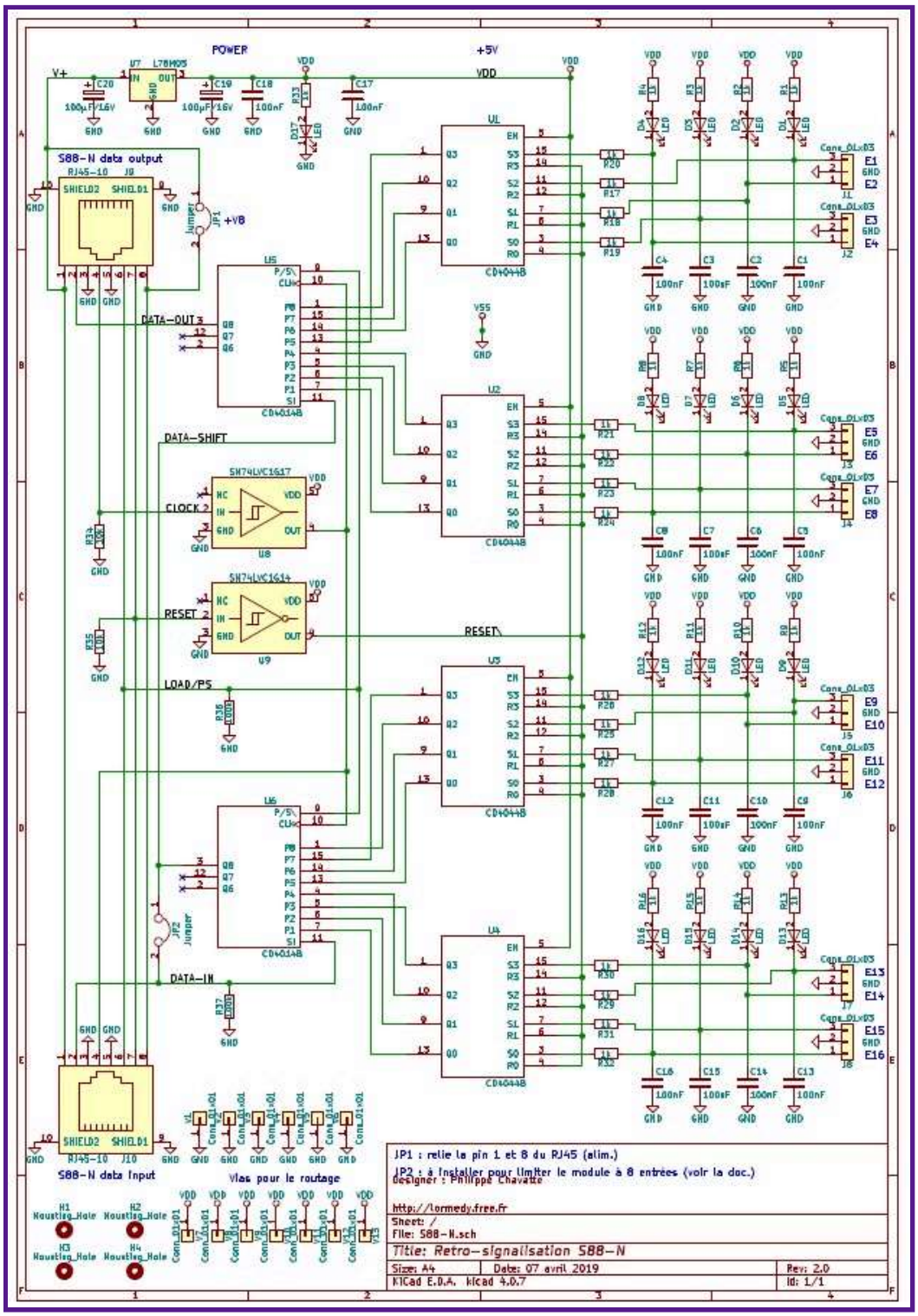
Mon module esclave S88-N 8E/16E



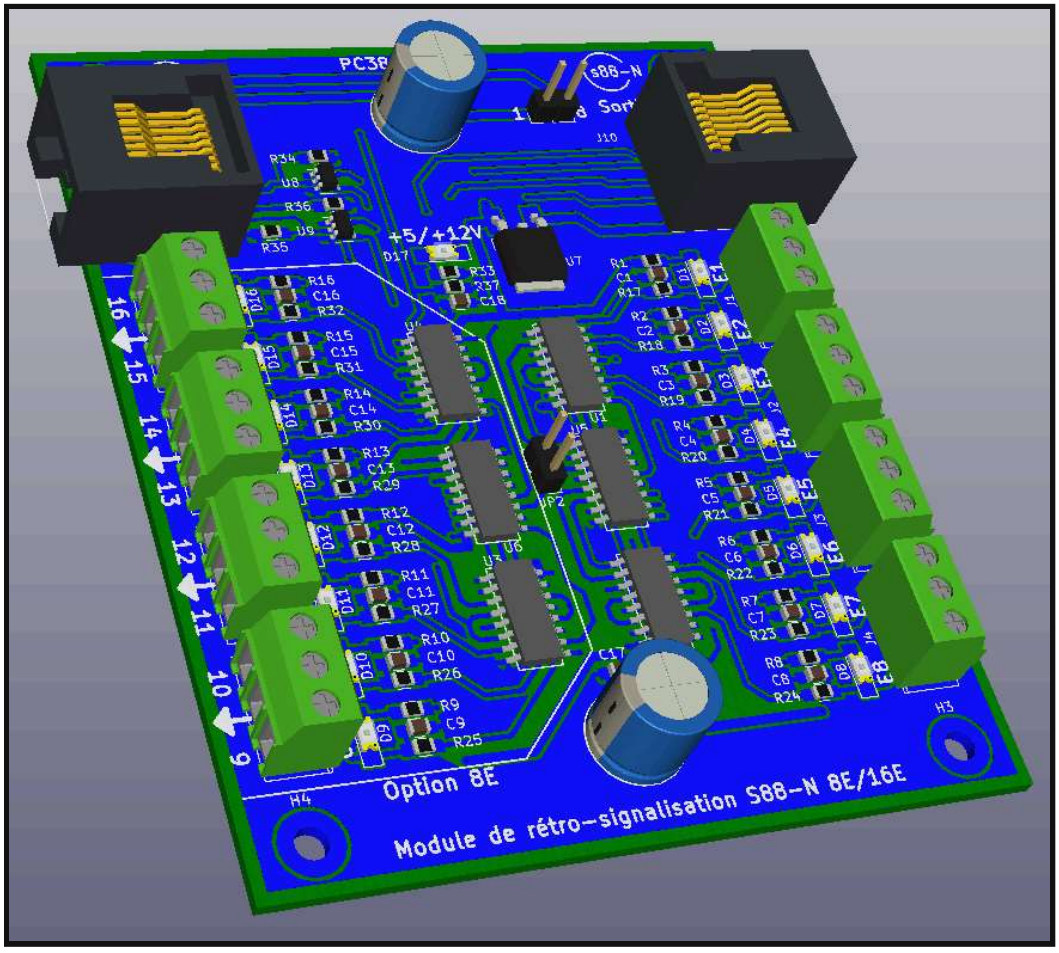
Carte soudée à la main

Ce module peux surveiller 16 entrées non isolées actives par contact à la masse. Pour les réseaux ferroviaires construits en longueur et utilisant peu d'entrée par module, celui-ci présente une option qui permet d'utiliser que 8 entrées en ne soudant pas les composants inutiles. Le circuit imprimé double face avec plan de masse utilise des composants CMS et mesure 80x97mm. Il s'alimente en +5V par le câble RJ45 dont on peut utiliser la pin 8 pour renforcer l'alimentation positive. Il est aussi possible de l'alimenter avec une tension supérieure comprise entre +7V et +12V car un régulateur U7 est implanté sur le module (en option). Je recommande cette option pour chainer un nombre important de modules si les derniers sont sous-alimentés à cause des pertes dans les câbles. J'ai ajouté un buffer d'horloge pour la même raison mais ce n'est pas nécessaire pour les data qui sont régénérées par chaque module. Une LED verte indique la présence du +5V sur la carte et l'état

des 16 entrées est visualisé avec des LED jaunes individuelles pour faciliter le repérage des actionneurs. Les entrées sont protégées contre presque tous les mauvais branchements. La sortie **S88-N** de l'Arduino MEGA se connecte sur "Entrée" du premier module **S88-N** et sa "Sortie" se connecte en série sur "Entrée" du module **S88-N** suivant. Et ainsi de suite pour former la chaine de modules **S88**.



Voici le plan dessiné avec le logiciel libre **KiCad** (avec lequel est réalisé le circuit imprimé ci-dessous).



En option, il est possible de réduire le module à 8 entrées en n'installant pas les composants situés dans la zone entourée d'une ligne blanche et en mettant un cavalier sur JP2.

Les composants sont suffisamment grands pour être soudé à la main par un modéliste soigneux. Seuls U8 et U9 réclament une attention plus précise.

Si vous utilisez une alimentation de +5V soudez un petit fils entre les pattes 1 et 3 de U7 sans installer U7 ni C20. Le régulateur U7 et le condensateur C20 ne sont soudés que pour ceux qui utilisent une alimentation supérieure à +5V. Ce module est télé-alimenté par le cable RJ45 et ne nécessite pas d'alimentation externe.

PC 28/03/2019

<u>Sélection de plusieurs modules S88-N avec 16 entrées :</u>

- Des modules développés par des amateurs (quelques €)

- TAMS Elektronik 44-01305-01-C S88-3 (25 € en kit, 16 entrées non optocouplées mais pas de RJ45)
- LDT LittfinskiDatenTechnik RM-88-N (30 € en kit, 16 entrées non optocouplées)
- LDT LittfinskiDatenTechnik RM-88-N-O (40 €, 8 entrées optocouplées), que je ne recommande pas.
- Digikeijs, Etc.

Ces modules fonctionnent avec une entrée de type pull-up qui est mise à la masse par un interrupteur, un contact, une sortie de type open-collector ou un signal 0-5V.

Voici un adaptateur qui s'adapte entre le bus S88 et le RBUS de ROCO (XpressNet) : R-Bus interface to S88-N

Le support du réseau La voie H0 Plan du réseau Les cantons Les détecteurs de trains Boucle de retournement Rétro-signalisation Le câblage Electronique DCC : Le DCC
Accessoires et aiguillages Décodeurs pour servo Décodeurs solénoïdes + signaux La centrale DCCppS88 USB La centrale DCCppS88 Wifi La centrale DCC32S88 Wifi
Logiciels DCC : Les logiciels Centrale DCC série CDT31 Centrale DCC WiFi WDD6 TCO WiFi DMC WiFi MAM_config MAM2 Annonces en gare> La gare Les signaux Liens utiles

---

Contact : Envoyer un e-mail *(Remplir le champ "adresse" avec le nom de la gare en minuscule sans guillemet) lormedy@free.fr*
