

# Parrot Awards 2014

## Projet RoseOnRails



*Noémie Bristol, Yann Le Guen, Gleison Storto, Valeh Valiollahpour Amiri*

*Sous la supervision d'Alexis Polti et Samuel Tardieu - cursus ROSE Télécom ParisTech*

<http://www.hackster.io/295/roseonrails>

### Pourquoi contrôler des trains électriques grâce au BLE ?

Contrairement à ce que l'on pourrait penser, le marché des trains électriques ne touche pas seulement les jouets pour enfants, mais en grande majorité un public d'adultes passionnés de modélisme, qui n'hésitent pas à acheter des trains de plus en plus fidèles à la réalité et sont à la recherche de nouvelles méthodes de contrôle.

Actuellement, il n'existe qu'une seule norme de contrôle des trains électriques : le DCC (Digital Command Control). Cette norme, bien qu'efficace, présente un certain nombre de contraintes physiques et fonctionnelles. Tout d'abord, le DCC est un protocole de contrôle filaire qui oblige l'utilisateur à se servir d'une télécommande reliée au circuit par un fil. D'autre part, le DCC est un protocole unidirectionnel, ce qui signifie que l'utilisateur peut envoyer des ordres aux éléments du circuit - trains, feux, aiguillages - mais que ces éléments ne peuvent pas envoyer de message à l'utilisateur.

RoseOnRails répond à ces contraintes en utilisant un protocole de communication encore jamais exploité pour le contrôle des trains électriques : le Bluetooth Low Energy (BLE). Le BLE est un protocole sans fil qui permet ainsi à l'utilisateur de se libérer des contraintes matérielles que constituent les fils. Par ailleurs, le BLE permet de faire communiquer l'utilisateur et les éléments du circuit de façon bidirectionnelle : l'utilisateur peut toujours envoyer des ordres aux trains, feux et aiguillages, mais ces éléments peuvent cette fois-ci lui répondre à leur tour en lui envoyant des informations utiles. Par exemple, un train peut envoyer à l'utilisateur sa position sur le circuit, un aiguillage son état, et un feu sa couleur. En outre, le BLE présente, par rapport au Bluetooth classique, l'avantage de permettre une communication avec un nombre illimité d'appareils. Là où nous aurions été limités à une communication avec seulement 7 éléments de circuit en utilisant le Bluetooth classique, nous pouvons communiquer avec autant de trains que l'on veut avec le BLE.

Afin d'adapter le circuit de train que nous avons au protocole BLE, nous avons conçu des circuits imprimés pour les locomotives, les aiguillages et les feux ainsi que le code qui les contrôle.

### Architecture et fonctionnalités matérielles

Le circuit sur lequel nous avons travaillé comportait 3 locomotives, 25 aiguillages et une cinquantaine de feux de circulation. Afin de pimenter un peu notre projet, nous avons décidé de remplacer les feux de circulation classiques du circuit par 600 LEDs placées sous la voie de train permettant à l'utilisateur d'adapter son circuit de trains au jeu s'il le désire.

# Parrot Awards 2014

## Projet RoseOnRails



Noémie Bristol, Yann Le Guen, Gleison Storto, Valeh Valiollahpour Amiri

Sous la supervision d'Alexis Polti et Samuel Tardieu - cursus ROSE Télécom ParisTech

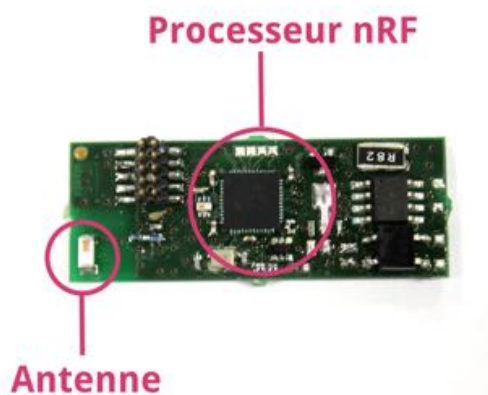
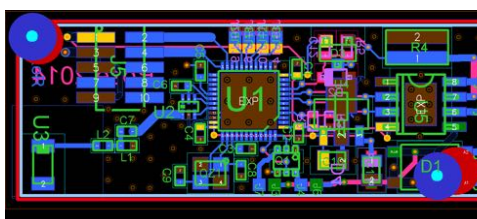
<http://www.hackster.io/295/roseonrails>

Au niveau matériel, nous avons dû concevoir et monter trois types de circuits imprimés pour les locomotives, les aiguillages et les LEDs qui étaient contrôlés en BLE par une station centrale (application qui s'exécute sur un ordinateur).

### • Locomotives

La première contrainte à laquelle nous avons dû faire face lors de la conception des PCBs<sup>1</sup> des locomotives était la taille. La cabine des locomotives étant de petite taille, nous avons dû concevoir un PCB qui était suffisamment petit pour entrer dans la locomotive et suffisamment grand pour pouvoir accueillir un certain nombre de composants, parmi lesquels :

- Une antenne permettant au train de communiquer en BLE avec la station centrale
- Un processeur nRF51822 comportant une pile BLE afin de pouvoir communiquer, et permettant également de contrôler le moteur de la locomotive en PWM (Pulse Width Modulation)
- Un pont en H permettant de contrôler la vitesse du moteur et le sens de parcours du train



### • Aiguillages

Lors de la conception du PCB des aiguillages, afin d'éviter l'achat de PCBs surnuméraires, nous avons décidé que chaque PCB d'aiguillage permettrait de contrôler plusieurs aiguillages (7 maximum). Comme pour les locomotives, ce circuit imprimé comporte notamment un processeur nRF51822 ainsi qu'une antenne.

---

<sup>1</sup> Printed Circuit Board (circuit imprimé)

# Parrot Awards 2014

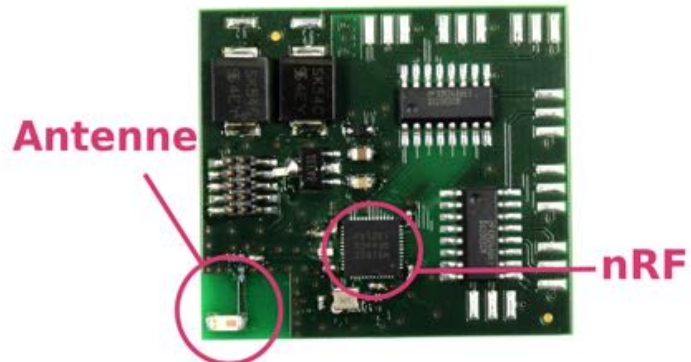
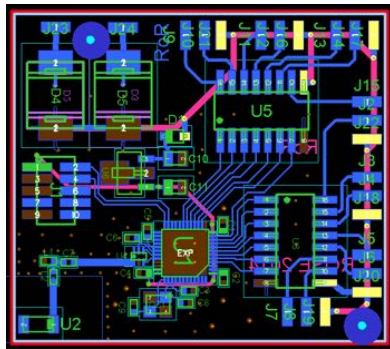
## Projet RoseOnRails



Noémie Bristol, Yann Le Guen, Gleison Storto, Valeh Valiollahpour Amiri

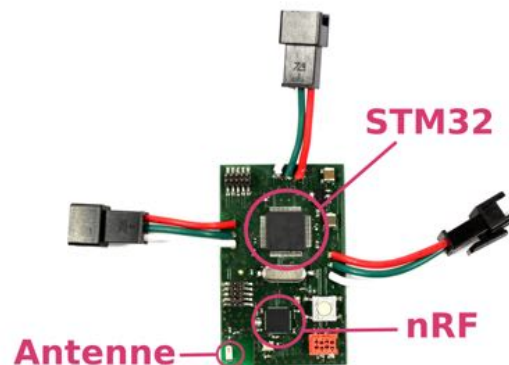
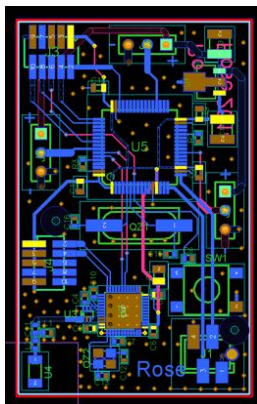
Sous la supervision d'Alexis Polti et Samuel Tardieu - cursus ROSE Télécom ParisTech

<http://www.hackster.io/295/roseonrails>



- Rubans de LEDs

À la différence du circuit imprimé précédent, le circuit imprimé des LEDs comporte en plus du nRF et de l'antenne, un processeur STM32F103RE permettant de faire des calculs un peu plus compliqués que le nRF. Nous avons, en effet, besoin d'une plus grande puissance de calcul pour contrôler de façon adéquate les rubans de LEDs qui requièrent un timing très précis dans l'envoi des commandes.



## Architecture logicielle

Il existe, dans notre système, trois entités principales : la locomotive, les aiguillages et les LEDs placées sous les rails. Chacun de ces appareils est muni d'une application lui permettant de communiquer en

# Parrot Awards 2014

## Projet RoseOnRails



*Noémie Bristol, Yann Le Guen, Gleison Storto, Valeh Valiollahpour Amiri*

*Sous la supervision d'Alexis Polti et Samuel Tardieu - cursus ROSE Télécom ParisTech*

<http://www.hackster.io/295/roseonrails>

BLE. La gestion de ces divers éléments est centralisée dans une seule application qui tourne sur la "Station Centrale" (un ordinateur). Cette application se scinde en deux grandes parties:

- Un module chargé de gérer les communications BLE avec les éléments du circuit de train : le *ble\_manager*
- Un module chargé de recevoir les commandes des utilisateurs et de prendre la décision des commandes à envoyer au circuit de train : le *gateway*

Ces deux modules communiquent en utilisant le protocole AMQP<sup>2</sup>. L'implémentation d'AMQP que nous avons utilisé est RabbitMQ.

Le *gateway* est une application codée en Python. Il contient le code pour recevoir les commandes des utilisateurs par websocket<sup>3</sup> et le code pour transférer ces commandes au *ble\_manager*.

Le *ble\_manager* est codé en Javascript. On y utilise la bibliothèque noble (module node.js permettant d'utiliser la pile BLE sous Linux) pour envoyer les commandes en BLE au circuit et recevoir les notifications venant de ce dernier (des trains). Le *ble\_manager* contient aussi une partie de code lui permettant de recevoir les commandes venant du *gateway* et de transmettre à ce dernier les notifications arrivant du circuit.

## Conclusion

L'utilisation du protocole BLE pour commander des circuits de train électrique est une nouveauté non encore explorée et qui permet d'améliorer diverses difficultés liées à l'usage du DCC. Ce progrès concerne non seulement le marché des jouets pour enfants mais aussi celui du modélisme ferroviaire.

De plus, notre API est conçue de sorte à pouvoir s'interfacer avec une application s'exécutant sur le téléphone mobile de l'utilisateur et qui pourrait envoyer les commandes en websocket à notre station centrale. Il est donc envisageable d'augmenter le nombre de trains et de concevoir des jeux multi-joueurs où chaque joueur peut contrôler sa locomotive à l'aide de son smartphone.

En résumé, le projet RoseOnRails est une solution pour remplacer la norme DCC avec un protocole de communication sans fil, sans limite au niveau du nombre de devices, simple d'utilisation, et qui se répand de plus en plus sur le marché des téléphones mobiles.

---

<sup>2</sup> Protocole de communication orienté message avec gestion de file

<sup>3</sup> Protocole de communication bidirectionnelle et half-duplex au-dessus de HTTP