Notice de configuration et d'utilisation

Réaliser un TCO « évolutif » avec un écran tactile

1. Présentation ____ 1

2. Dessiner le TCO ____ 3
- a. *Dimensions du dessin ... 3*
- b. *Format d'enregistrement et d'export ... 4*

3. Compiler le programme sous « Processing » ____ 4
- a. *Processing ... 4*
- b. *Compiler le projet ... 4*

4. Configurer le logiciel ____ 6
- a. *Paramétrage général ... 6*
- b. *Déterminer les ports COM utilisés ... 7*
- c. *Démarrage et initialisation des paramètres ... 7*
- d. *Paramétrage d'un TCO ... 8*
- e. *Exemple avec un circuit « test » ... 10*
- f. *Touches du TCO ... 12*
- g. *Icônes du TCO ... 13*

5. Cartes ARDUINO ____ 13
- a. *Décodeur DCC ... 14*
- b. *Souris XpressNet ... 14*
- c. *Clavier 16 touches ... 16*

6. Bonus – Autres possibilités… ____ 18
- a. *Pupitre ... 18*
- b. *Transformer le Raspberry Pi 3 en routeur Wifi ... 19*
- c. *Raspberry : Prise en main à distance ... 20*
- d. *Raspberry : Attribuer une IP fixe ... 22*
- e. *Raspberry : Installer d'autres applications ... 22*
- f. *Boitier GenLI pour le protocole XpressNet ... 23*
- g. *Raspberry : Installer l'application Python ... 23*
- h. *Raspberry : Installer les ports GPIO ... 23*
- i. *Raspberry : Installer des boutons Marche / Arrêt 23*
- j. *Raspberry : Configurer un écran LCD ... 25*
- k. *Raspberry : Bouton pour lancer une application . 28*
- l. *Application pour tablette et Smartphone ... 28*

1. Présentation

Le projet consiste à réaliser un TCO, « Tableau de Commande Optique » qui permet de visualiser la position des aiguillages ou l'état des accessoires et de les commander.

Le TCO a été réalisé pour une utilisation avec une centrale DCC utilisant le protocole « XpressNet ». Le projet pourra être adapté pour fonctionner en analogique.

Le projet est évolutif et se compose de plusieurs briques optionnelles.

Le projet met en œuvre un Raspberry Pi 3, des cartes Arduino dont les programmes et plans des cartes shield (ou boucliers) sont disponibles gratuitement.

L'objectif est de disposer d'un pupitre de commande évolutif en conservant la possibilité de commander les aiguillages avec les doigts et non une souris. Le projet permet également d'avoir une vision de plusieurs plans du réseau en fonction de la configuration mise en œuvre.

On entend par « accessoire » un équipement tel qu'un aiguillage, des barrières, des lumières, des lecteurs sonores, des animations diverses et variées, …

Le pupitre TCO met en œuvre :

- Un ***Écran LCD*** :
  - Si l'écran est tactile, on pourra piloter directement les accessoires avec les doigts ;
  - Si l'écran n'est pas tactile, l'écran servira à visualiser la position des accessoires.

- Un ***Ordinateur*** pour piloter l'écran LCD et traiter les informations :
  - Un ordinateur portable ou fixe équipé de ports USB (Windows, Mac ou Linux) ;
  - ou un ***Raspberry PI*** 3 de préférence avec une distribution Raspbian : **https://www.raspberrypi.org/downloads/raspbian/**

Sauf mention particulière, le terme générique Ordinateur concernera un équipement comme une unité centrale, un ordinateur portable, un nano ordinateur Raspberry Pi.

- Un ***Décodeur d'accessoires*** fonctionnant avec un Arduino :
  - Le décodeur reçoit les informations des accessoires pilotés et envoie les informations à l'ordinateur. Le décodeur est relié aux connecteurs « DCC » de la centrale.

- Une ***Souris XpressNet*** fonctionnant avec un Arduino :
  - La souris permet d'envoyer à la centrale DCC la commande des accessoires pilotés depuis l'écran tactile. La souris est reliée à la centrale via le connecteur « Xbus » « esclave » ou « slave ». Cette souris est optionnelle si on ne veut pas piloter d'accessoires via l'écran tactile ou le logiciel.

- Une ***Application*** compatible sous Windows, Macintosh et Linux :
  - L'application a été réalisée avec le logiciel gratuit « Processing » et permet une compilation pour rendre auto exécutable le logiciel sur plusieurs systèmes d'exploitation (Windows, Mac, Linux).

- Un ***Clavier 16 touches*** fonctionnant avec un Arduino :
  - L'ordinateur et l'écran tactile ne disposant pas de clavier, il a été ajouté un clavier 16 touches pour disposer de raccourcis pour accéder rapidement à des affichages du TCO. Le clavier est relié à un Arduino qui est connecté à l'ordinateur via un port USB pour simuler un clavier. *Ce clavier est optionnel.*

L'ensemble du projet (fichiers source) est disponible en téléchargement sur le site :

http://modelisme58.free.fr

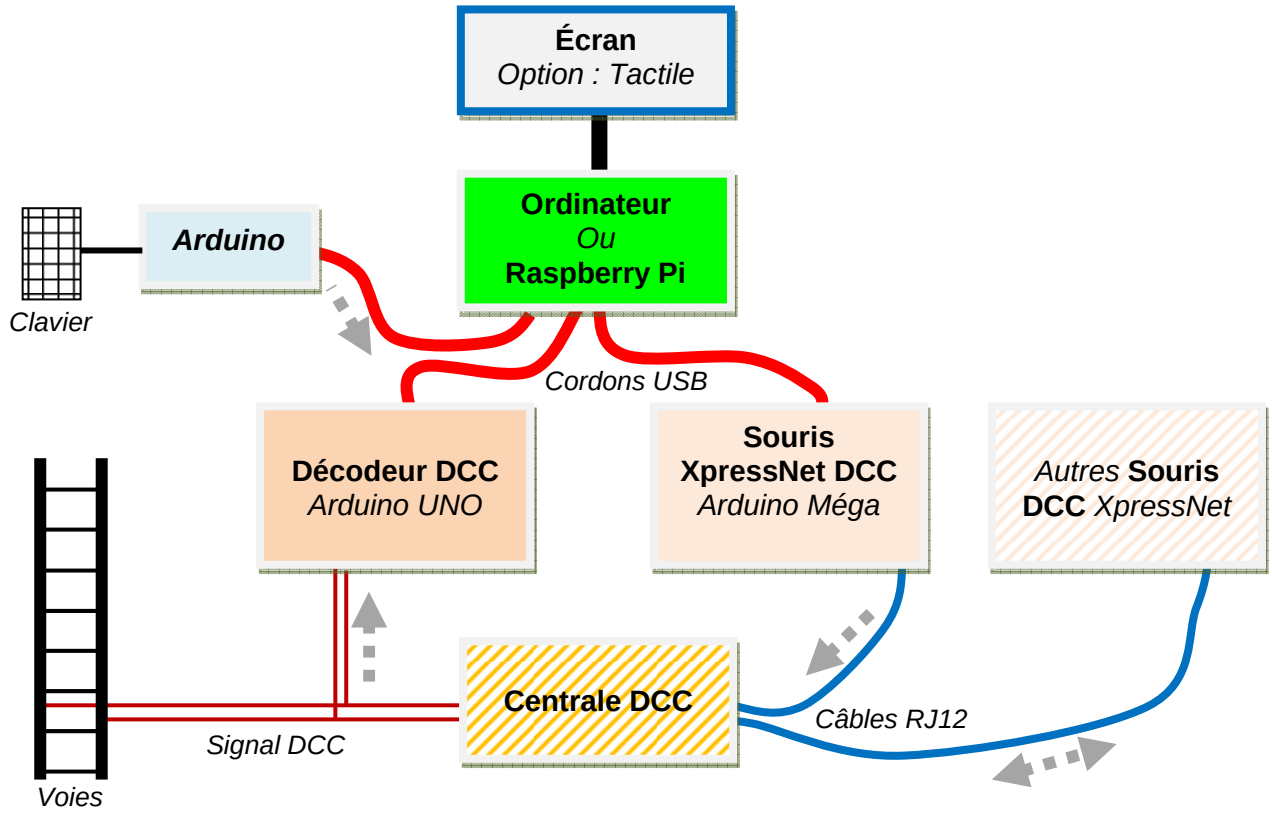


Schéma de principe

2. Dessiner le TCO

Il est nécessaire de réaliser le schéma du réseau : l'utilisation d'un logiciel de dessin vectoriel permet un redimensionnement sans perte de qualité.

Il est proposé d'utiliser « **Inkscape** » qui est un logiciel de dessin vectoriel qui fonctionne sous Windows, Mac OS X et GNU/Linux. Il est libre et gratuit.

Le logiciel est disponible à l'adresse : **https://inkscape.org**

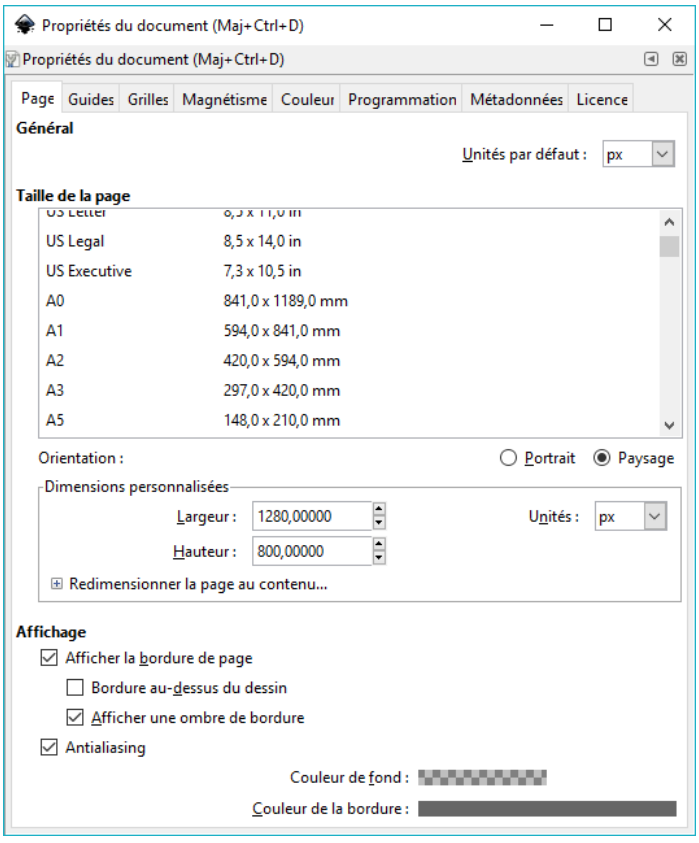
a. *Dimensions du dessin*

La dimension du dessin sera déterminée par la taille maximale configurée dans l'application « TCO ». Le paramétrage par défaut est « **1280 x 800** » avec un écran tactile de 10 pouces avec un Raspberry Pi

Dans le logiciel « **Inkscape** », cliquer sur « ***Fichier → Propriétés du document…*** ».

Dans la fenêtre qui s'ouvre, sélectionner l'onglet « ***Page*** » et indiquer les dimensions avec l'unité « ***px*** ».

Fermer la fenêtre à l'aide de la croix en haut à droite. Les paramètres sont pris en compte dans le logiciel.



Réaliser le tracer du plan ou du réseau ferroviaire. Des couleurs différentes peuvent être utilisées suivant les voies principales, de service,…

b. *Format d'enregistrement et d'export*

Afin de pouvoir modifier ultérieurement le dessin, il faut enregistrer le fichier au format « SVG ».

Pour utiliser le dessin, il faut le convertir dans un format « image ». Inkscape le permet avec la fonction « ***Fichier → Exporter une image PNG…*** ».

Le format « *PNG* » permet de gérer la transparence des fonds des fichiers image.

Indiquer le nom du fichier et son emplacement puis cliquer sur « Exporter ».

3. Compiler le programme sous « Processing »

a. *Processing*

L'application « TCO » a été réalisée avec le logiciel « Processing ».
« **Processing** » utilise un langage et un environnement de programmation open-source.

Basé sur la plate-forme Java, le logiciel fonctionne sous Macintosh, Windows et Linux.

Pour la mise en œuvre et utilisation du « TCO », aucune connaissance en programmation n'est exigée.

Le logiciel est téléchargeable à l'adresse : **https://processing.org**

L'extension « **.pde** » est utilisée par le logiciel Processing.

b. *Compiler le projet*

Ouvrir le fichier « **TCO.pde** » téléchargé au préalable.

L'application compilée sera utilisée en plein écran. Si besoin, changer la résolution de l'écran dans la fonction « ***SETUP*** ».

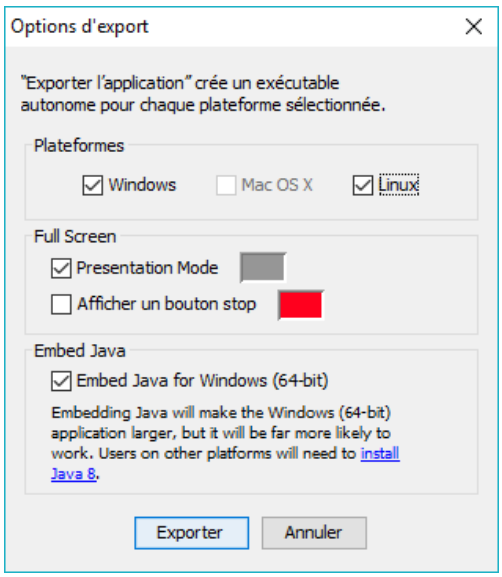
Aller à la ligne 100 :

```
void setup ()
{
  //size(displayWidth, displayHeight );
  size (1280, 800);
```

Enregistrer le projet si des modifications ont été réalisées.

Pour compiler le projet afin de réaliser le fichier exécutable, cliquer dans le menu sur :
« ***Fichier → Exporter*** »

Cocher la ou les cases des systèmes d'exploitation souhaités :



Cliquer sur le bouton « ***Exporter*** » et attendre que la compilation soit terminée.

Un répertoire est créé pour chaque système d'exploitation précédemment sélectionné.

Lancer l'application en fonction du système d'exploitation utilisé :

L'application s'ouvre en plein écran.

Pour quitter l'application, appuyer sur la touche « ***Echap*** » ou sur le bouton .

Remarques :

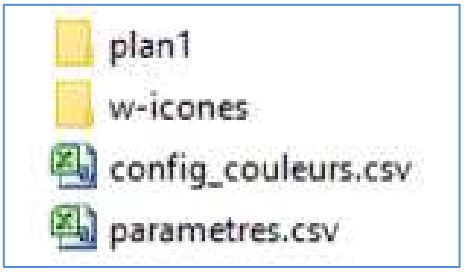
- Avec un Raspberry Pi, il faut utiliser l'application compilée dans le répertoire « application.linux-armv6hf ».
- Avec un Raspberry Pi, il est nécessaire de donner les droits d'accès « ***Exécuter = Tout le monde*** ». ***Faire un clic « Droit » sur le fichier exécutable, sélectionner « Propriétés » puis l'onglet « Droits d'accès*** ».

- **Il n'est pas nécessaire d'installer « Processing » pour utiliser le programme compilé avec un autre ordinateur, avec un Raspberry Pi.**

4. Configurer le logiciel

De nombreux affichages sont paramétrables. Les données sont stockées dans le répertoire « data » qui ne devra pas être renommé.

Le répertoire « data » comprend les 2 fichiers de configuration générale et le répertoire comprenant les icônes. Il sera créé autant de répertoires que l'on gère de TCO ou plan de circulation.



Les fichiers pourront être ouverts et modifiés avec un logiciel éditeur de texte comme le « Bloc-Notes » ou un tableur (Excel, LibreOffice,...).

- **Avec un Raspberry Pi, la modification des fichiers « CSV » sera réalisée uniquement avec le logiciel LibreOffice (Calc) - https://fr.libreoffice.org**
  Ne pas utiliser le logiciel « Text Editor ».

a. *Paramétrage général*

Le fichier « **config_couleurs.csv** » permet de configurer les couleurs qui seront utilisées.

La première ligne du fichier comprend les entêtes des colonnes. Les variables seront séparées par une virgule « , ».

La première colonne correspond à la couleur, la deuxième correspond à l'ordre et les 3 suivantes aux références RVB.

Le code couleur RVB se présente sous la forme de 3 couples de codes en chiffres compris entre 0 et 255.

Chaque couple représente le dosage nécessaire de chacune des couleurs primaires pour obtenir la couleur désirée. Les trois codes représentent respectivement le dosage du rouge, du vert et du bleu.

```
Couleur,Numero,R,V,B
Gris,0,150,150,150
Vert,1,0,255,0
Bleu,2,0,0,255
Jaune,3,255,255,0
Violet,4,128,0,128
Marron,5,128,64,0
Fushia,6,255,0,128
Bleu Clair,7,74,126,187
Noir,8,0,0,0
```

Il est nécessaire de redémarrer l'application « TCO » afin que ces modifications soient prises en compte.

Le fichier « **parametres.csv** » permet de configurer l'application :

```
Libelle,Valeur
XpressNet,9          -> Adresse de la souris XpressNet (entre 1 et 31)
Plan,1               -> Numéro du plan ouvert au démarrage du logiciel
Format,png           -> Extension des fichiers images (schéma TCO)
Port0,COM3           -> Numéro du port COM numéro 0
Port1,COM11          -> Numéro du port COM numéro 1
*** Fin paramètres ***
```

- Sous Windows, les ports COM sont nommés COM1, COM2, COM3,…
- Sous Linux, les ports COM commencent par : /dev/tty

b. *Déterminer les ports COM utilisés*

Pour déterminer les ports COM utilisés, relier au préalable les cartes ARDUINO avec un câble USB à l'ordinateur.

Lancer l'application « **TCO** ».

Appuyer sur la touche « **Tab** » ou « **|< >|** » et visualiser les ports utilisés : ils s'affichent dans une fenêtre. Les ports connectés sont indiqués à « Droite ».

Si les noms des ports COM sont correctement configurés, ils s'affichent à « Gauche ».

c. *Démarrage et initialisation des paramètres*

Lors du démarrage de l'application « TCO », le numéro du plan, les couleurs, les noms des ports COM, l'adresse de la souris XpressNet sont chargés via les données enregistrées dans le fichier « parametres.csv ».

Lors de cette phase d'initialisation, des échanges de données ont lieu entre l'application « TCO » et les cartes Arduino (enregistrement de l'adresse XpressNet si elle a été modifiée, …).

Lors de l'ouverture d'un plan (au démarrage ou à l'aide des touches Fn1 à Fn10 ou via les boutons « Loupe »), les paramètres du plan sont chargés automatiquement.

Un accessoire est représenté par un cercle : lorsqu'un accessoire est désactivé, son cercle est rempli en « Rouge ».

Les accessoires peuvent prendre 3 états :

- Activé ;
- Désactivé ;
- Indéterminé.

Quand un accessoire est à l'état indéterminé, sa couleur est celle configurée dans les paramètres de couleur du fond du plan.

Quand un accessoire est activé, sa couleur est celle configurée le fichier « adressesDDCX.csv ».

d. ***Paramétrage d'un TCO***

La configuration d'un plan est déterminée à l'aide de 5 fichiers :

- adressesDDCX.csv
- configPlanX.csv
- itinerairesX.csv
- planX.png
- configZoomX.csv

avec X le numéro du plan.

Les fichiers peuvent être ouverts et modifiés avec un logiciel éditeur de texte comme le « Bloc-Notes » ou un tableur (Excel, LibreOffice,...).

L'application a été conçue afin de permettre de s'adapter à de nombreux réseaux. Si le paramétrage peut paraître fastidieux, cette étape n'est à réaliser qu'une seule fois !

Le fichier « **planX.png** » ou « planX.jpg » est le dessin du réseau ferroviaire ou une partie du réseau.

Le fichier « **configPlanX.csv** » comprend un nombre déterminé de lignes correspondant au paramètre du plan :

Libelle,Valeur
Diametre,**15** -> Diamètre en pixels par défaut des boutons des accessoires
PlanX,**5** -> Origine X du plan (en haut à gauche)
PlanY,**50** -> Origine Y du plan (en haut à gauche)
Titre,**1 - Test des Accessoires** -> Nom du plan qui sera affiché
Police,**30** -> Taille de la police d'écriture du titre
TitreX,**450** -> Coordonnée X du texte du titre
TitreY,**90** -> Coordonnée Y du texte du titre
TitreCouleur,**6** -> Référence de la couleur du texte du titre
FondCouleur,**0** -> Référence de la couleur du fond (partie transparente)

Remarques :

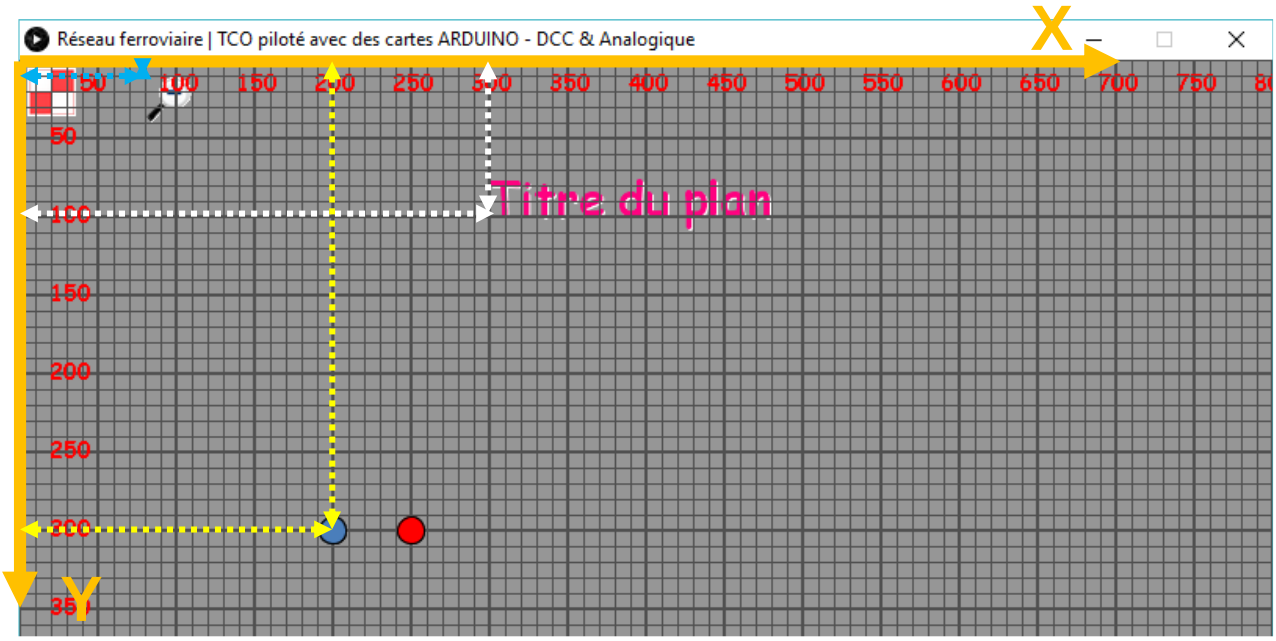
- Le texte du titre ne doit pas contenir de caractère accentué.
- Il est recommandé d'indiquer le numéro du plan dans le titre afin de faciliter la navigation entre les différentes vues du circuit ferroviaire.
- Lorsque l'on appelle un plan et que le répertoire ne contient pas les fichiers « adressesDDCX.csv », « itinerairesX.csv », « configZoomX.csv », ces derniers sont créés automatiquement. Un accessoire par défaut, est défini dans le fichier « adressesDDCX.csv » et positionné sur le plan.

Le fichier « **adressesDDCX.csv** » comprend un nombre indéterminé de lignes. Un accessoire sera représenté en général avec 2 lignes pour ses 2 états possibles.

Exemple :

Aiguillage,DCC,Sens,X,Y,Couleur,Diametre,Description
1,21,1,200,400,7,Aiguillage 1
1,21,2,250,400,7,Aiguillage 2

1 -> Numéro de l'accessoire (numéro utilisé pour se repéré, à ne pas confondre avec l'adresse DCC)
21 -> Adresse DCC de l'accessoire
1 ou 2 -> Accessoire activé ou désactivé
200 -> Coordonnée X du centre du cercle
200 -> Coordonnée Y du centre du cercle
7 -> Référence de la couleur de remplissage du cercle
Aiguillage 1 -> Description de l'accessoire (optionnel)



Repères et origine (X, Y) des objets

Le fichier « **configZoomX.csv** » comprend un nombre indéterminé de lignes. Il permet de positionner des images représentant une « loupe » afin de passer d'un plan à un autre. Lors du changement de plan, l'état des accessoires n'est pas changé.

Utiliser les loupes permet de visualiser certaines parties du réseau.

Exemple :

Plan,X,Y,Zoom,Description
plan2,80,10,0,Plan 2
plan3,80,100,1Plan 3

plan2 -> Indique le plan à afficher en cliquant sur le bouton
80 -> Coordonnée X de l'image (en haut à gauche)
10 -> Coordonnée Y de l'image (en haut à gauche)
0 ou 1 -> 0 = loupe avec le symbole (-) | 1 = loupe avec le symbole (+)
Plan 2 -> Description du plan (optionnel)

Remarque :

- S'il n'y a pas de loupe à afficher, l'ensemble du contenu du fichier pourra être effacé.

Le fichier « **itinerairesX.csv** » comprend un nombre indéterminé de lignes. Ce fichier va permettre de commander plusieurs accessoires lors de l'activation d'un accessoire : cette fonctionnalité est particulièrement utile pour commander des aiguillages dans le cadre de la réalisation d'un itinéraire.

Exemple :

DCC,Sens,DCC_2,Sens_2
11,1,16,2
12,1,11,1
15,1,13,2
13,2,15,1
15,2,12,2

11	-> Adresse DCC de l'accessoire piloté
1 ou 2	-> Accessoire activé ou désactivé
16	-> Adresse DCC de l'accessoire à piloter
1 ou 2	-> Accessoire activé ou désactivé

Lorsqu'un accessoire est piloté via l'application « TCO », les accessoires concernés sont également pilotés. Les ordres seront envoyés à la centrale les uns après les autres automatiquement ;

Remarque :

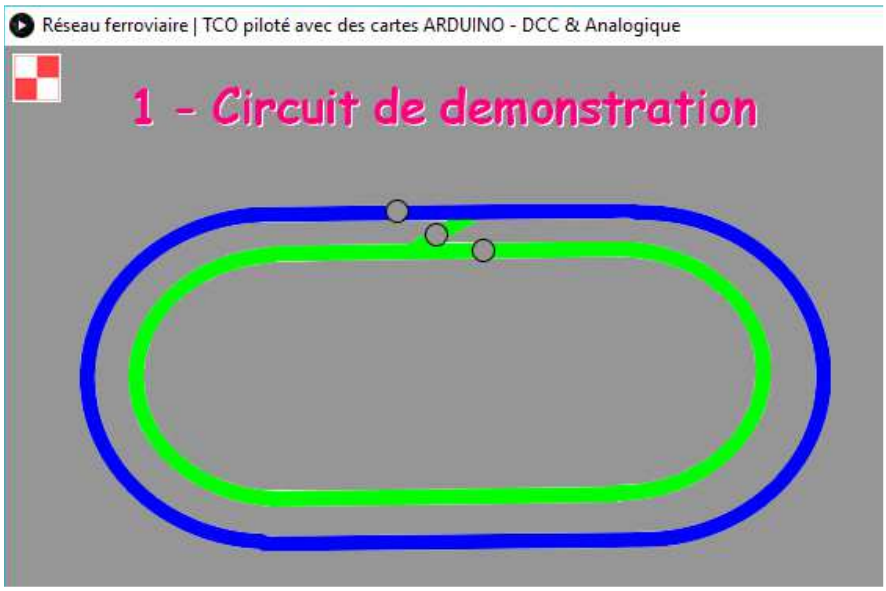
- S'il n'y a pas d'itinéraires à piloter, l'ensemble du contenu du fichier pourra être effacé.

e. *Exemple avec un circuit « test »*

Pour la démonstration, le réseau de démonstration comprend 1 ovale en double voie avec 2 aiguillages pilotées en digital (DCC).

Chaque aiguillage devrait comprendre 2 boutons de commandes : « Droit » et « Dévié ».

Comme les 2 aiguillages peuvent être pilotés en même temps, il est possible d'afficher un seul bouton pour la position déviée pour éviter une superposition de boutons.



Affichage du circuit à l'ouverture de l'application « TCO »

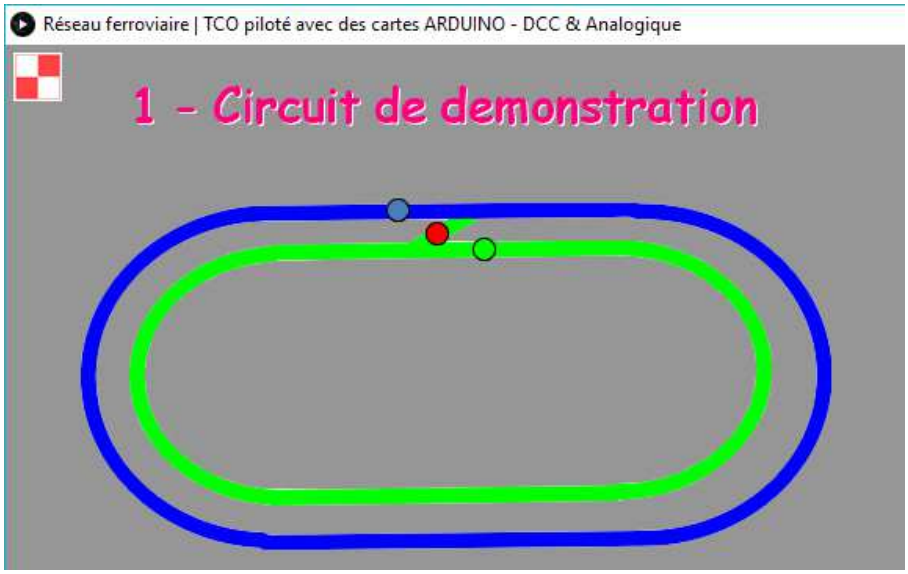
Les aiguillages sont configurés comme suit :

```
Aiguillage,DCC,Sens,X,Y,Couleur,Diametre,Description
1,21,1,305,130,1
2,21,1,250,105,7
2,21,2,275,120,3
```

Les aiguillages sont pilotés avec l'adresse « 21 ».

Les aiguillages « N°1 » et « N°2 » peuvent être pilotés avec la centrale DCC avec l'adresse 21 ou en cliquant sur un seul bouton sur le TCO.

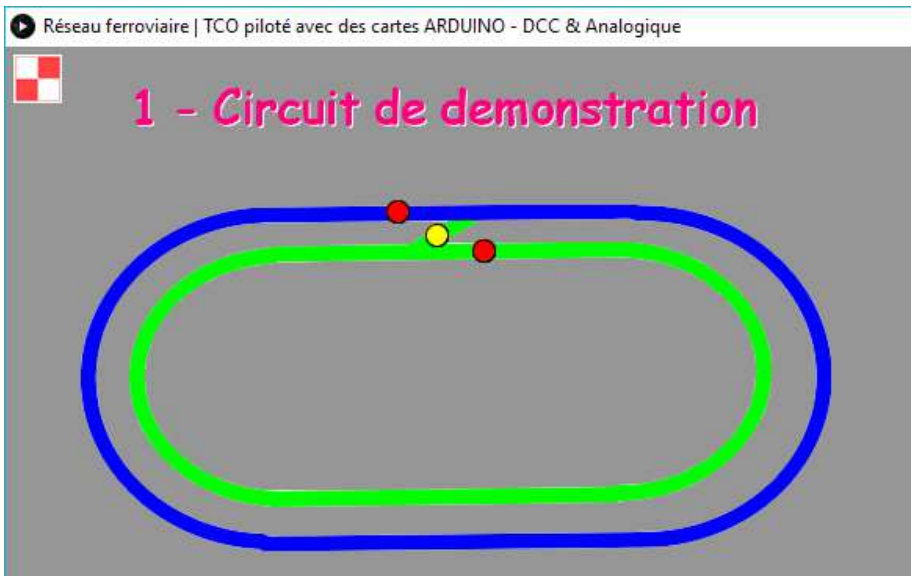
Les couleurs des autres boutons vont changer automatiquement.



Vue du circuit avec les aiguillages en position « Droite »

Le bouton « **Rouge** » indique que le franchissement des aiguillages en position déviée, est interdit.

Les 2 autres boutons ont la même couleur que les voies, soit des franchissements possibles de l'aiguillage en position « Droite ».



Vue du circuit avec les aiguillages en position « Déviée »

Les boutons « **Rouge** » indique que des itinéraires sont interdits.

Le bouton « **Jaune** » indique que les aiguillages sont en position déviée. La couleur est « **Jaune** » afin d'alerter l'utilisateur que l'on change de zone ou de sens de circulation. La couleur aurait pu être configurée en « Vert » ou « Bleu ».

f. Touches du TCO

À l'aide des touches du clavier de l'ordinateur, il est possible d'avoir accès à certaines fonctions :

- Touches « **Fn1** » à « **Fn10** » : Affiche et actualise les plans 1 à 10 ;
- Touche « **Supp** » : Réinitialise les plans, tous les accessoires sont à l'état « indéterminé ;
- Touche « **Tab** » : Visualise les ports paramétrés, les ports utilisés et les paramètres définis ;
- Touche « **Flèche Haut** » : Affiche sous l'icône « Carré » les coordonnées de la souris lors d'un clic ;
- Touche « **Flèche Bas** » : Affiche les caractéristiques de l'accessoire sous l'icône « Carré » lorsque la souris est positionnée dessus ;
- Touche « **Flèche Gauche** » : Masque les coordonnées de la souris ou les caractéristiques de l'accessoire ;
- Touche « **Flèche Droite** » : Affiche le quadrillage sur le plan.

g. *Icônes du TCO*

Le TCO a été conçu afin d'avoir un affichage épuré. Les icônes présents sont :

Bouton pour quitter l'application

Icône indiquant que le TCO est reconnue par la centrale DCC avec le protocole XpressNet

Icône indiquant que le TCO est reconnue par la centrale DCC avec le protocole XpressNet mais que la centrale est en court-circuit ou qu'un arrêt d'urgence a été enclenché

Bouton permettant de visualiser un autre plan

Bouton permettant de visualiser un autre plan

5. Cartes ARDUINO

Pour permettre au logiciel « TCO » de communiquer avec la centrale, 2 cartes Arduino sont à connecter à l'ordinateur :

- « Décodeur DCC » : la carte communique les informations de commandes des accessoires de la centrale DCC vers le logiciel « TCO » ;

- « Souris XpressNet » : la carte communique les informations du logiciel « TCO » vers centrale DCC.

On utilise 2 cartes Arduino car une solution fiable en utilisant 1 seule carte n'a pas été trouvée.

Remarques :

- Si on utilise uniquement la carte « Décodeur DCC », le logiciel fonctionnera avec toutes les centrales DCC utilisant la norme NMRA. Ceci permet d'avoir un tableau optique avec la position des aiguillages et l'état des accessoires ;

- « Souris XpressNet » : le protocole XpressNet est un bus de commande utilisé pour permettre aux souris de communiquer par les centrales Atlas, Lenz, Hornby, OpenDCC, Paco Canada, ZTC Controls, Viessmann, et Roco. Le bus XpressNet a été développé par Lenz. La souris « Souris XpressNet » esclave est compatible avec les centrales Roco 10761, 10764, Z21, Z21, Viessmann Commander,… : il faudra nécessairement disposer d'une souris maître Lockmaus ou Multimauss avec une centrale Roco 10761, 10764.

- On peut aussi développer un projet Arduino qui mette en œuvre la rétro signalisation avec le protocole « S88 ».

a. *Décodeur DCC*

Le « Décodeur DCC » est mis en œuvre avec une carte Arduino UNO ou Nano. Aucune librairie n'est à installer et aucune configuration n'est à réaliser. Il faudra veiller à ce que le débit de communication du port COM soit de « 38400 bauds ».

On peut télécharger le code source sur le site : **http://modelisme58.free.fr**

On ajoute un shield à la carte Arduino :

- Relier les 2 fils du signal DCC (signal DCC envoyer dans les voies) au bornier de la carte (JP6) ;
- Relier la carte Arduino à l'ordinateur à l'aide d'un câble USB. Ce même câble alimente en 5 Volts la carte Arduino UNO ou Nano.

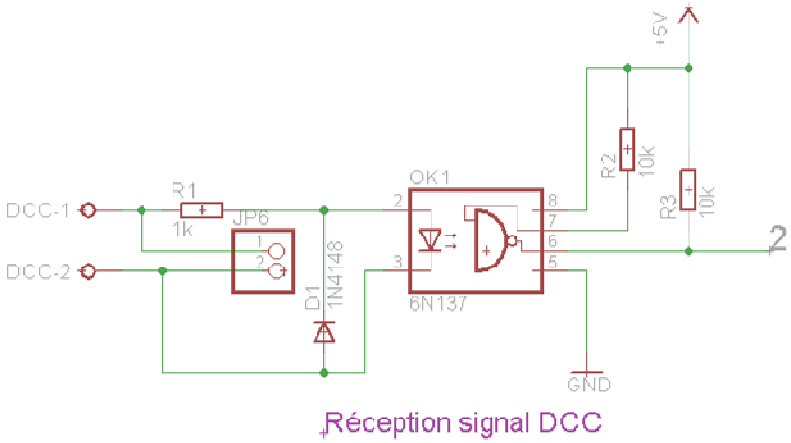


Schéma du décodeur

Le décodeur met en œuvre un optocoupleur 6N137.

La broche « **6** » du composant 6N137 est à relier à la borne « **2** » de l'Arduino.

b. *Souris XpressNet*

La « Souris XpressNet » est mise en œuvre avec une carte Arduino Méga.

On utilise les librairies EEPROM (gestion de la mémoire), SoftwareSerial (port Série) et XpressNet (protocole XpressNet de communication avec la centrale DCC).

La librairie « XpressNet » peut être téléchargée depuis le site :
http://sourceforge.net/projects/pgahtow/files/

Ensuite, il faut décompresser l'archive « XpressNet.zip » dans le répertoire où sont installées les librairies Arduino.

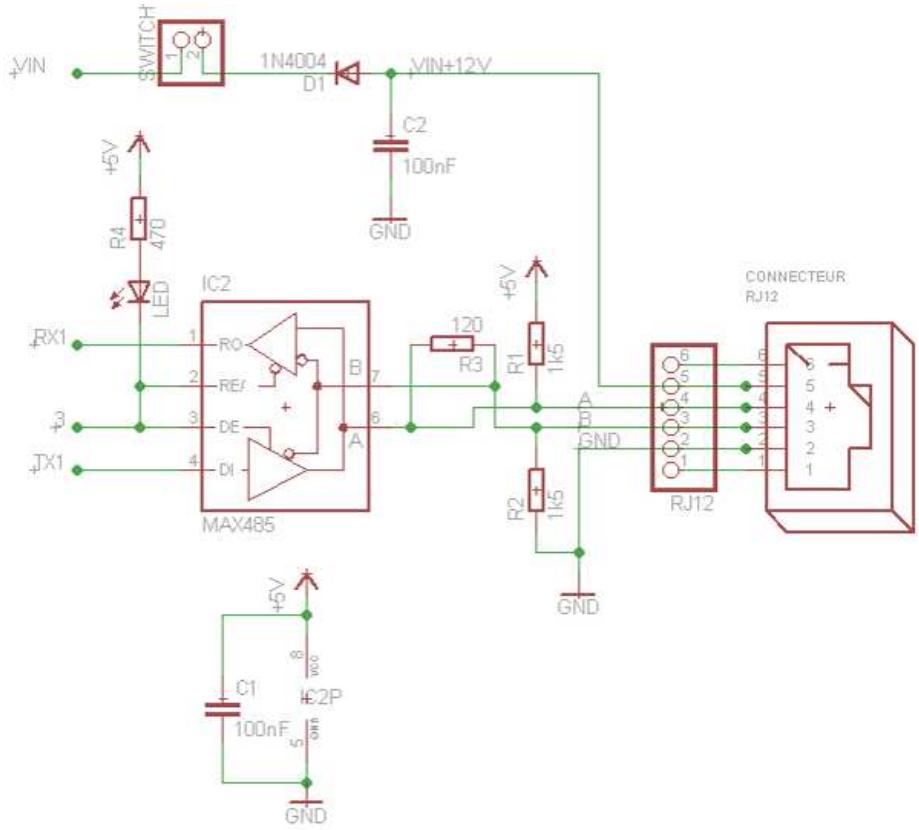


Schéma de la souris XpressNet

Le décodeur met en œuvre un composant électronique MAX485 qui est un transcepteur (transmetteur/récepteur ou transceiver).

La broche « **3** » du composant MAX485 est à relier à la borne « **3** » de l'Arduino Méga.
La broche « **1** » du composant MAX485 est à relier à la borne « **18** » de l'Arduino Méga.
La broche « **4** » du composant MAX485 est à relier à la borne « **19** » de l'Arduino Méga.

On peut télécharger le code source sur le site : **http://modelisme58.free.fr**

Il faudra veiller à ce que le débit de communication du port COM soit de « 38400 bauds ».

On ajoute un shield à la carte Arduino :

- Relier le connecteur RJ12 à la prise « Esclave » de la centrale DCC. Seuls les fils 2, 3, 4 et 5 sont utilisés.
- Relier la carte Arduino à l'ordinateur à l'aide d'un câble USB. Ce même câble alimente en 5 Volts la carte Arduino Méga.

La librairie « EEPROM » est utilisée afin d'enregistrer dans la mémoire de l'Arduino l'adresse de la souris Xpressnet. La valeur doit être comprise entre 1 et 31. La commande de l'enregistrement se fait via la configuration de l'application « TCO ».

Remarque :

- A défaut de connecteur RJ11 ou RJ12 disponible, les fils d'une rallonge téléphonique peuvent être directement soudés sur la carte.

c. *Clavier 16 touches*

Un Arduino équipé d'un ATMega32U4 permet d'être reconnu comme un périphérique USB. Cela signifie donc que l'on peut utiliser un Arduino pour simuler des touches d'un clavier USB.

Remarque :

- ATMega32U4 : le « U » signifie USB.

Sont compatibles les cartes Arduino « Leonardo », « Due » et « Micro Pro ».

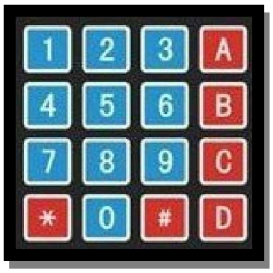
On utilise un clavier matriciel 4x4 afin de disposer de 16 touches.

Il faut installer au préalable 2 librairies pour compiler le programme :

- « Keyboard » pour simuler le clavier USB ;
- « Keypad » pour l'utilisation d'un clavier matriciel.

L'Arduino est relié à l'ordinateur avec le port USB.

Avec l'application « TCO », les touches ont les fonctions suivantes :

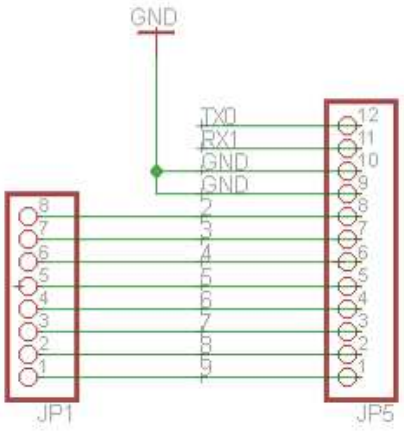


- Touches « **1** » à « **9** » : Affiche le plan 1 à 9 ;
- Touche « **0** » : Affiche le plan 10 ;
- Touche « ***** » : Réinitialise les plans, tous les accessoires sont à l'état « indéterminé ;
- Touche « **#** » : Visualise les ports configurés, les ports utilisés et les paramètres définis ;
- Touche « **A** » : Affiche les coordonnées de la souris lors d'un clic ;
- Touche « **B** » : Affiche les caractéristiques de l'accessoire lorsque la souris est positionnée dessus ;
- Touche « **C** » : Masque les coordonnées de la souris ou les caractéristiques de l'accessoire ;
- Touche « **D** » : Affiche le quadrillage sur le plan.

Le programme suivant est à compiler et à téléverser dans l'Arduino :

```
#include "Keyboard.h"          // Clavier USB
#include <Keypad.h>            // Touches
const byte Lignes   = 4;       // Nombre de lignes du clavier
const byte Colonnes = 4;       // Nombre de colonnes du clavier
char hexaTouches[Lignes][Colonnes] = {
  {194, 195, 196, 218},
  {197, 198, 199, 217},
  {200, 201, 202, 216},
  {212, 203, 179, 215} };
byte ColBroches[Colonnes] = {6, 7, 8, 9 };    // Clavier Colonne
byte LigBroches[Lignes]   = {2, 3, 4, 5 };    // Clavier Ligne
Keypad Clavier16Touches = Keypad( makeKeymap(hexaTouches), LigBroches, ColBroches,
Lignes, Colonnes);
void setup() {
delay (2000);
  Serial.begin(9600);
  Keyboard.begin();
}
```

```
void loop() {
  char Touche = Clavier16Touches.getKey();
  if (Touche)   {  Keyboard.write(Touche);  }
  delay (30);
}
```



Connexion du clavier matricielle à une carte Arduino

6. Bonus – Autres possibilités…

Sorti officiellement début 2012, le Raspberry Pi est développé par la Raspberry Pi Foundation. Il s'agit d'un ordinateur dont la taille est comparable à celle d'une carte de crédit.

Un Raspberry Pi est un mini-ordinateur qui permet à la fois de faire l'électronique ou de la robotique (grâce aux GPIO) mais aussi à installer un serveur voire de disposer d'un ordinateur d'appoint pour quelques dizaines d'€uros.

a. *Pupitre*

La mise en œuvre de ce pupitre est réalisée à l'aide de logiciels et d'applications gratuites :

- Logiciel « **LibreCAD** » pour dessiner les plans du pupitre en vue d'une découpe laser ;
- Logiciel « **FreeCAD** » pour vérifier l'assemblage en 3D des éléments du pupitre et pour dessiner les charnières, les cavaliers de fixation des câbles et pièces diverses ;
- Logiciel « **Processing** » pour réaliser l'application « TCO » ;
- Logiciel « **Inkscape** » pour dessiner les plans du réseau ferroviaire ;
- Logiciel « **Arduino** » pour programmer les cartes électroniques ;
  et
- Logiciel « **Eagle** » (utilisation de la version limitée) pour dessiner les schémas et les circuits électroniques.

Une partie du pupitre a été réalisé avec l'aide du FabLab d'Auxerre (89) – Atelier des Beaux Boulons (**http://www.beauxboulons.org**).

Les différents éléments en panneau de fibres de bois MDF (appellation anglophone pour medium density fiberboard) d'épaisseur de 3, 6 et 9 mm ont été découpés et gravés à l'aide d'une découpeuse laser.

Les pièces en 3D ont été imprimées à l'aide d'une imprimante 3D RepRap.

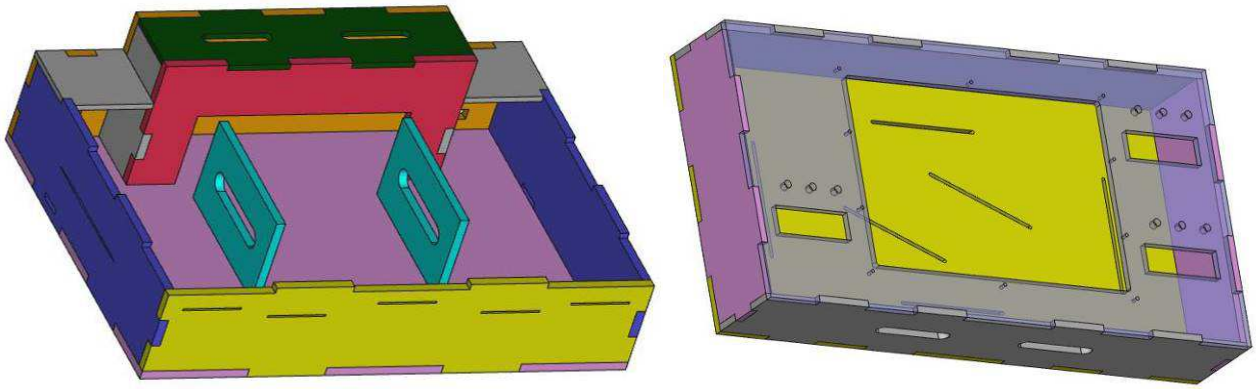
La mise en œuvre du pupitre est assez simple. Seulement 3 câbles sont à relier au pupitre qui se comporte comme une souris « Esclave » utilisant le protocole Xpressnet :

- **Alimentation 220 Volts** avec interrupteur général ;
- **Signal DCC** (signal envoyé à travers les voies) avec 2 fiches bananes ;
- **Câble XpressNet** (relié à la centrale DCC).

Le pupitre met en œuvre :

- **1 « Raspberry »** avec **écran tactile de 10 pouces** pour utiliser l'application « TCO » : la configuration utilise également un écran LCD 16x2 pour afficher les adresses IP, 3 boutons poussoirs (Arrêt, Scan IP / redémarrage, lancement de l'application « TCO ») et la fonction routeur Wifi ;
- **1 « Souris TCO »** avec une **carte Arduino Méga** pour piloter les accessoires depuis l'application « TCO » ;
- **1 « Décodeur DCC »** avec une **carte Arduino Nano** pour envoyer les ordres des commandes d'accessoires à l'application « TCO » ;
- **1 « Clavier 16 touches »** avec une **carte Arduino Micro Pro** pour piloter l'application « TCO ».

- **2 « Souris xPSE »** avec deux **cartes Arduino Méga** pour piloter les locomotives et éventuellement les accessoires : chaque carte utilise un lecteur de carte SD, un écran LCD 16x2, un encodeur rotatif, 2 boutons poussoirs et un bouton à levier (cf. le site Internet : **http://modelisme58.free.fr**) ;



Les 2 éléments du pupitre modélisés en 3D avec le logiciel FreeCAD

Pupitre assemblé

b. *Transformer le Raspberry Pi 3 en routeur Wifi*

On peut transformer le Raspberry Pi 3 en routeur Wifi afin de permettre d'utiliser des applications Wifi (Rocrail, Z21,…) via un Smartphone ou une tablette.

En plus de son rôle de TCO, le Raspberry Pi sera configuré comme un routeur Wifi.

Il faut installer l'application « Hostapd » sur le Raspberry Pi 3 qui dispose déjà d'un port Ethernet et d'une connexion Wi-Fi intégrée.

Dans le terminal, il faut commencer par mettre à jour le système d'exploitation :

sudo apt-get update
sudo apt-get upgrade

Il faut ensuite lancer la ligne de commande :

wget -q https://git.io/voEUQ -O /tmp/raspap && bash /tmp/raspap

Il faut répondre aux questions qui vous sont posées et l'installation de tous les composants va se faire toute seule.

Lorsque l'installation est terminée, le Raspberry Pi 3 redémarre, le réseau « raspi-webgui » apparait dans la liste des réseaux Wifi accessibles.

Le réseau sans fil est configuré comme un point d'accès comme suit :

Adresse IP : ***10.3.141.1***
Nom d'utilisateur : ***admin***
Mot de passe : ***secret***
Plage DHCP : ***10.3.141.50*** à ***10.3.141.255***
SSID : ***raspi-webgui***
Mot de passe : ***ChangeMe***

Plus d'informations sont disponibles sur la page github de l'application :
https://github.com/billz/raspap-webgui

Pour administrer le hotspot Raspberry Pi créé, il faut se connecter via le navigateur Internet à l'adresse « **10.3.141.1** ».

L'adresse « **10.3.141.1** » correspond à l'adresse qui devra être configurée dans les applications Wifi utilisées avec les Smartphones et tablettes.

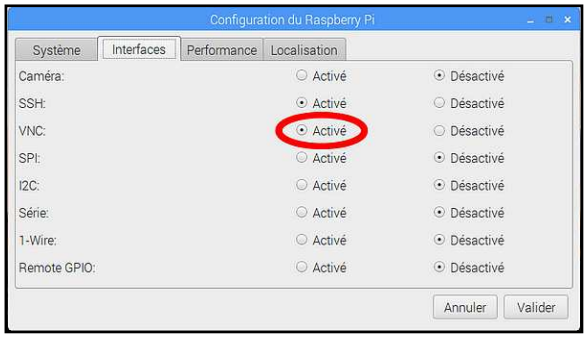
Il est conseillé de changer les mots de passe du login d'administration et le mot de passe de connexion au Wifi.

c. *Raspberry : Prise en main à distance*

Pour permettre la prise en main à distance du Raspberry Pi, il faut activer le service SSH.

Dans le terminal, il faut lancer la ligne de commande : **sudo raspi-config**

Dans le menu de configuration du Raspberry Pi et dans l'onglet Interfaces, activer le serveur VNC :



Redémarrer le Raspberry Pi.

Quand le hotspot est configuré, il ne faut pas installer l'application « Xrdp » pour la prise en main à distance.

Il faut installer « **TightVncServer** » sur le Raspberry Pi.

Saisir la ligne de commande dans le terminal : **sudo apt-get install tightvncserver**

Lancer l'application « **TightVncServer** » avec la commande **: vncserver :1**

Le serveur se lance.

Lors du premier lancement, il est demandé un mot de passe d'accès, qui ne doit pas comporter plus de 8 caractères.

Après confirmation du mot de passe, la question suivante apparait :

Would you like to enter a view-only password (y/n) ?

Taper 'n' et le mode 'view-only' sera inactif.

On peut configurer « TightVncServer » pour qu'il se lance automatiquement au démarrage du Raspberry Pi.

Dans le terminal de commande, saisir :

```
cd /home/pi
cd .config
mkdir autostart
cd autostart
sudo nano tightvncserver.desktop
```

Dans le fichier vide qui vient de s'ouvrir, saisir le texte suivant :

```
#!/bin/sh
[Desktop Entry]
Type=Application
Name=TightVNCServer
Exec=vncserver :1
StartupNotify=false
```

Sauvegarder le fichier avec les touches « ***Ctrl + o*** ».
Quitter l'éditeur de texte « ***Ctrl + x*** ».

Redémarrez le Raspberry Pi.

Remarque :

- Pour changer le mot de passe VNC, il faut utiliser la commande suivante dans le terminal : **vncpasswd**

Pour se connecter au Raspberry Pi depuis un ordinateur sous Windows, il faut installer un logiciel client gratuit. On peut utiliser PUTTY, VNC Viewer, RealVNC, …

Sous Linux, on peut installer « Vncviewer » grâce à la commande : **sudo apt-get install xvncviewer**

Connecter le Raspberry Pi et l'ordinateur au même réseau via le Wifi ou câbles RJ45.

Lancer l'application et entrer l'adresse IP du Raspberry Pi.

Si le hotspot a été installé, saisir l'adresse : « ***10.3.141.1:1*** »

D'une manière générale, il faut saisir l'adresse IP suivie de « ***:1*** ».

Saisir le mot de passe défini ci-dessus.

Le bureau apparait ce qui permet d'utiliser le Raspberry Pi, de paramétrer le logiciel « TCO » …

d. Raspberry : Attribuer une IP fixe

Dans certains cas, il peut être nécessaire d'attribuer une adresse IP fixe pour la connexion Ethernet.

Dans le terminal de commande, saisir :

sudo nano /etc/network/interfaces

Recherchez la ligne : « iface eth0 inet dhcp ».

Remplacer la ligne par:

iface eth0 inet static
address 192.168.1.x
netmask 255.255.255.0
network 192.168.1.0
gateway 192.168.1.1

Remarque :

- Les paramètres ci-dessus sont à modifier suivant le réseau !

Enregistrer et fermer le fichier : « ***Ctrl + o*** » (pour enregistrer) puis « ***Ctrl + x*** » (pour fermer)

Redémarrer le Raspberry Pi.

e. Raspberry : Installer d'autres applications

Pour le modélisme ferroviaire, on peut installer d'autres applications sur un Raspberry telles que :

- ARDUINO : **https://www.arduino.cc/en/Main/Software**
- JMRI : **http://jmri.sourceforge.net**
- Rocrail : **http://wiki.rocrail.net/doku.php?id=start-fr**
- Inkscape : **https://inkscape.org**
- LibreOffice : **https://fr.libreoffice.org**
- **…**

Pour l'**installation des applications**, il faudra télécharger les paquets **Linux ARM**.

f. Boitier GenLI pour le protocole XpressNet

Les logiciels JMRI et Rocrail pourront être installés en tant que souris et/ou serveur et on pourra se connecter via des applications mobiles via la connexion Wifi (*cf. Transformer le Raspberry en routeur Wifi*) ou RJ45.

Si on utilise JMRI ou Rocrail avec une centrale Roco utilisant le protocole XpressNet, il faut installer un boitier entre le Raspberry Pi et la centrale. Le boitier GenLI remplit cette fonction et peut être réalisé via les plans et codes source disponibles depuis le site Internet :
http://usuaris.tinet.cat/fmco/lokmaus_en.html

Ce boitier comprend un port COM à relier au Raspberry Pi via un cordon « USB-Port Série » et un port RJ12 (4 ou 6 fils) à relier à la centrale sur le port « Esclave » de la souris.

g. Raspberry : Installer l'application Python

Pour installer l'application Python, saisir les lignes de commande dans le terminal :

```
sudo apt-get update
sudo apt-get install python-dev
```

h. Raspberry : Installer les ports GPIO

Pour utiliser les ports GPIO, il faut installer le module RPi.GPIO à l'aide de la ligne de commande :

```
sudo apt-get install RPi.GPIO
```

i. Raspberry : Installer des boutons Marche / Arrêt

Le Raspberry Pi ne dispose pas de bouton pour l'arrêt du nano-ordinateur.

Il est possible d'ajouter 1 bouton poussoir connecté au GPIO4 et un programme pour réaliser cette fonction.

Le programme est développé en Python et lancé au démarrage du Raspberry Pi.

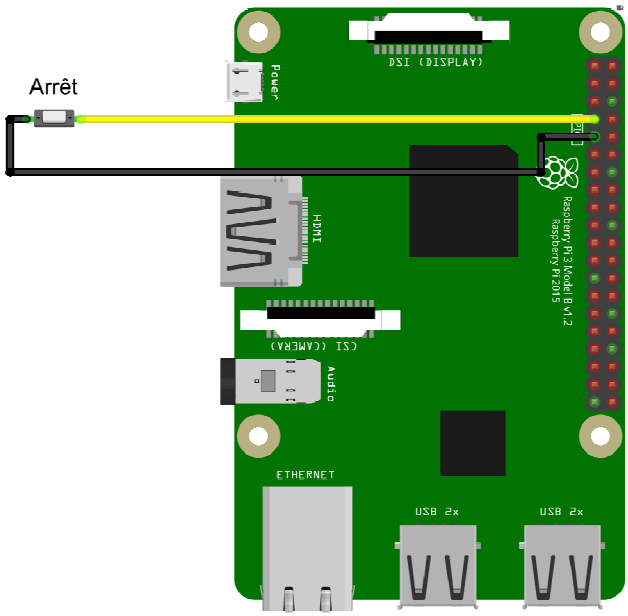


Schéma de câblage du bouton « Arrêt » pour un Raspberry Pi 2 ou 3

Pour éviter des arrêts par erreur, le bouton poussoir « **Arrêt** » devra être maintenu enfoncé au moins 4 secondes pour lancer le processus d'arrêt du Raspberry Pi.

Le script est enregistré dans un fichier nommé « **bouton_arret.py** ».

Dans le terminal de commandes, saisir :

```
sudo nano bouton_arret.py
```

Saisir le texte suivant :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import os, sys
if not os.getuid() == 0:
    sys.exit('Needs to be root for running this script.')
import RPi.GPIO as GPIO
import time
import subprocess
from time import sleep

# the button is connected on GPIO4 (pin 7 of header)
BTN_IO = 7
# we use the Broadcom numbering of the I/O
# (instead of the RasPi header pin numbering)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(BTN_IO, GPIO.IN, GPIO.PUD_UP)
print('Ecran demarre')
while True:
    pressed = (GPIO.input(BTN_IO) == 0)
    if pressed:
        time.sleep(4)
        pressed = (GPIO.input(BTN_IO) == 0)
        if pressed:
            break
    else:
```

```
        time.sleep(0.1)
    print('Bouton ARRET appuye - Le systeme va s arreter maintenant')
    subprocess.call('halt')
```

Dans le terminal de commande, saisir la commande pour déplacer le fichier créé :

sudo cp bouton_arret.py /usr/local/bin

Pour ajouter ce script au lancement dans la configuration du Raspberry Pi, il faut créer un fichier décrivant un « service ».

Dans le terminal de commande, saisir :

Sudo nano /etc/systemd/system/bt_arret.service

Dans le fichier vide qui vient de s'ouvrir, saisir le texte suivant :

```
[Unit]
Description=Bouton Arret

[Service]
Type=oneshot
ExecStart=/usr/bin/python3 /usr/local/bin/bouton_arret.py

[Install]
WantedBy=multi-user.target
```

Pour tester le bon fonctionnement de la configuration et du service, saisir dans le terminal de commandes :

sudo systemctl start bt_arret.service

Cliquer sur le bouton « Arrêt » pour arrêter le Raspberry Pi.

Pour lancer le service au démarrage du Raspberry Pi, saisir dans le terminal de commandes :

sudo systemctl enable bt_arret.service

j. Raspberry : Configurer un écran LCD

L'utilisation d'un écran LCD 16x2 permet d'afficher les adresses IP utilisées par le Raspberry Pi 3 (RJ45 et Wifi).

L'affichage permettra de connaitre rapidement l'adresse IP afin de se connecter à distance, d'utiliser avec le Wifi les applications Rocrail, JMRI, …

L'afficheur LCD est configuré en mode 4 bits.

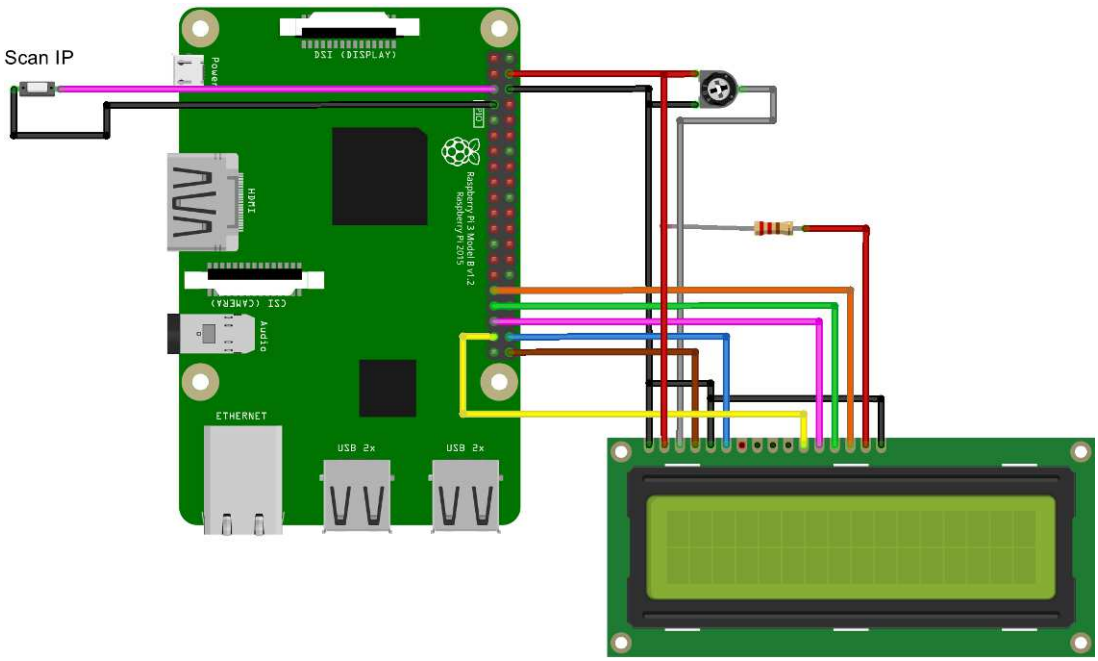


Schéma de câblage de l'écran LCD 16x2 avec un Raspberry Pi 2 ou 3

Le programme est développé en Python et lancé au démarrage du Raspberry Pi.

Le script est enregistré dans un fichier nommé « **LCDIP.py** ».
Dans le terminal de commandes, saisir :

```
sudo apt-get install netifaces*
sudo nano LCDIP.py
```

Saisir le texte suivant :

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import RPi.GPIO as GPIO
import netifaces
from time import sleep

class HD44780:

    def __init__(self, pin_rs=40, pin_e=38, pins_db=[37, 35, 33, 31]):

        self.pin_rs = pin_rs
        self.pin_e = pin_e
        self.pins_db = pins_db

        GPIO.setmode(GPIO.BOARD)
        GPIO.setup(self.pin_e, GPIO.OUT)
        GPIO.setup(self.pin_rs, GPIO.OUT)
        for pin in self.pins_db:
            GPIO.setup(pin, GPIO.OUT)

        self.clear()

    def clear(self):
        self.cmd(0x33) # init 8 bits
            self.cmd(0x32) # init 8 bits confirmation
```

```
            self.cmd(0x28) # 4 bits - 2 lignes
            self.cmd(0x0C) # Pas de curseur
            self.cmd(0x06) # IncrÃ©mentation curseur
            self.cmd(0x01) # Efface Ã©cran

        def cmd(self, bits, char_mode=False):
            sleep(0.005)
            bits=bin(bits)[2:].zfill(8)

            GPIO.output(self.pin_rs, char_mode)

            for pin in self.pins_db:
                GPIO.output(pin, False)

            for i in range(4):
                if bits[i] == "1":
                    GPIO.output(self.pins_db[::-1][i], True)

            GPIO.output(self.pin_e, True)
            GPIO.output(self.pin_e, False)

            for pin in self.pins_db:
                GPIO.output(pin, False)

            for i in range(4,8):
                if bits[i] == "1":
                    GPIO.output(self.pins_db[::-1][i-4], True)

            GPIO.output(self.pin_e, True)
            GPIO.output(self.pin_e, False)

        def message(self, text):
            for char in text:
                if char == '\n':
                    self.cmd(0xC0)
                else:
                    self.cmd(ord(char),True)

        def __del__(self):
            GPIO.cleanup()

    if __name__ == '__main__':
        lcd = HD44780()
        lcd.clear()
        lcd.message("Mon IP:\n%s" %
    netifaces.ifaddresses('eth0')[netifaces.AF_INET][0]['addr'])
```

Dans le terminal de commande, saisir la commande pour déplacer le fichier créé :

sudo cp LCDIP.py /usr/local/bin

Pour ajouter ce script au lancement dans la configuration du Raspberry Pi, il faut créer un fichier décrivant un « service ».

Dans le terminal de commande, saisir :

Sudo nano /etc/systemd/system/lcdip.service

Dans le fichier vide qui vient de s'ouvrir, saisir le texte suivant :

```
[Unit]
Wants=network-online.target
After=network.target network-online.target ssh.service

[Service]
Type=oneshot
ExecStart=/usr/bin/python3 /usr/local/bin/LCDIP.py

[Install]
WantedBy=multi-user.target
```

Pour tester le bon fonctionnement de la configuration et du service, saisir dans le terminal de commandes :

```
sudo systemctl start lcdip
```

Cliquer sur le bouton « ***Scan IP*** » connecté au GPIO3 du Raspberry Pi pour afficher les informations.

Pour lancer le service au démarrage du Raspberry Pi, saisir dans le terminal de commandes :

```
sudo systemctl enable lcdip
```

Remarque :

- Lorsque le Raspberry Pi a été arrêté avec le bouton poussoir « Arrêt », un appui sur le bouton « ***Scan IP*** » redémarre le Raspberry Pi.

k. Raspberry : Bouton pour lancer une application

Pour une question de facilité, on peut lancer l'application « TCO » directement à l'aide d'un bouton poussoir relié au Raspberry Pi.

l. Application pour tablette et Smartphone

Sur le site **http://sourceforge.net/projects/pgahtow/files/**, on peut aussi trouver des fonctions à rajouter au pupitre…