

Mise en route de LaBox

Le premier démarrage

En branchant une alimentation 12V, sur l'écran Oled de LaBox, on doit voir apparaître successivement les écrans suivants :



LaBox peut être commandée de différentes manières :

- JMRI sur le port USB
- Withrottle (application IOS) ou Engine Driver (application Android)
- Z21 (application sur IOS et Android)

On peut panacher des applications smartphone, jusqu'à 6 en même temps mais 3 maximum de type Withrottle et 3 maximum de type Z21.

A noter que ces applications ont été développées pour fonctionner avec JMRI donc leur comportement sera similaire.

Commençons par les applications sur smartphone, dans le même esprit que pour JMRI.

Je donne ici, pour commencer, deux exemples avec Withrottle Lite et Z21 Mobile, sur IOS (parce que j'ai un iPhone), mais il en existe aussi sur Android Engine Driver et Z21 Mobile aussi. On commencera donc par installer l'une de ces applications (gratuites) sur votre smartphone.

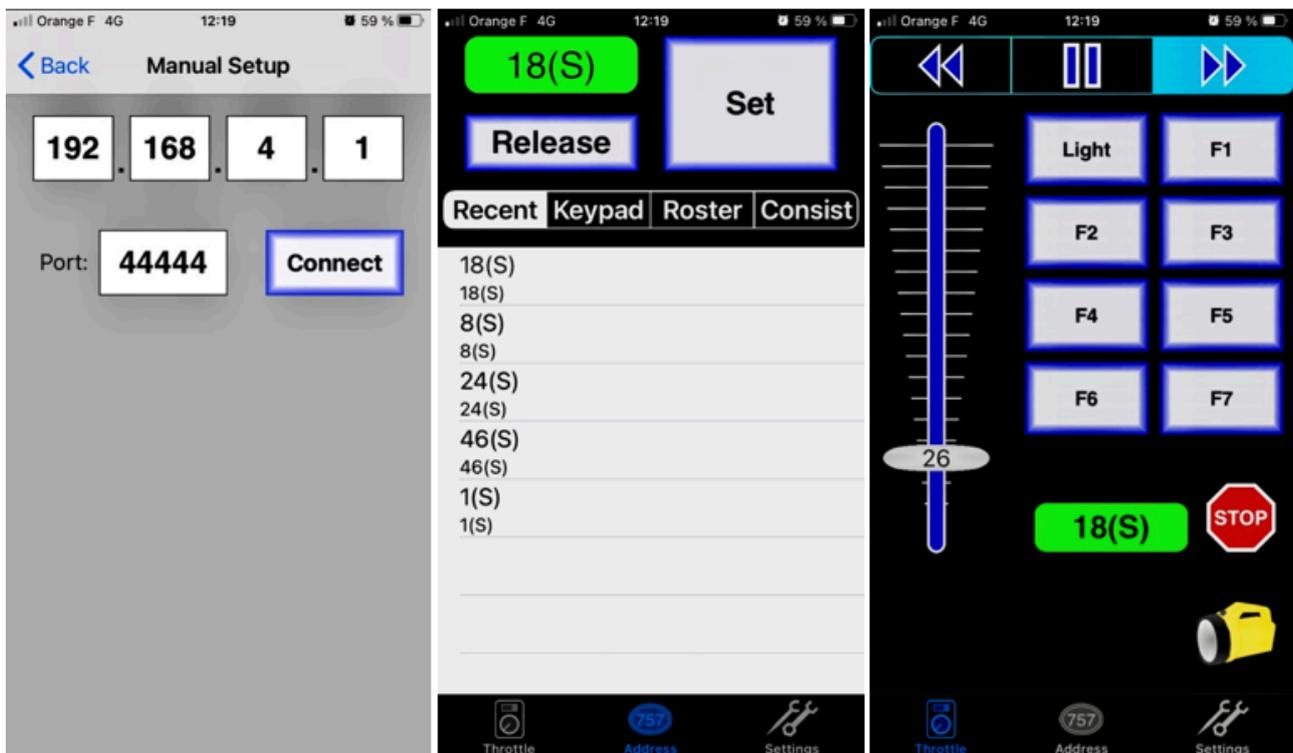
Pour commencer, on va connecter le smartphone sur le serveur WiFi que LaBox a créé : LoCoDuino LaBox, qui est un réseau sans mot de passe, mais sans risque pour votre réseau personnel à la maison.

Il faut donc changer de réseau WiFi avec les réglages de votre smartphone. Quand LaBox est branchée, le réseau LoCoDuino LaBox apparaît.



1) Exemple avec Withrottle Lite

Le Wifi de LaBox étant lancé et connecté en préalable, on peut lancer l'application Withrottle et cheminer dans les écrans (pour plus de détails on pourra se reporter à la documentation Withrottle)



Le premier écran montre les paramètres de le connexion Wifi.
L'adresse IP à configurer (écran de gauche) est **192.168.4.1**
Cette adresse est affichée sur l'écran de LaBox.
Le port est **44444**.

En cliquant sur **Connect**, remarquez que les Leds à coté du connecteur DCC s'allument toutes les deux : le DCC est présent sur les rails .
Le deuxième écran montre le choix de la locomotive (adresse DCC 18, soit présente dans l'historique, soit à saisir au clavier avec l'onglet Keypad - les autres onglets n'ont pas d'application pour le moment).
Le troisième écran montre le pilotage de la locomotive avec un curseur pratique, les fonctions et la direction.

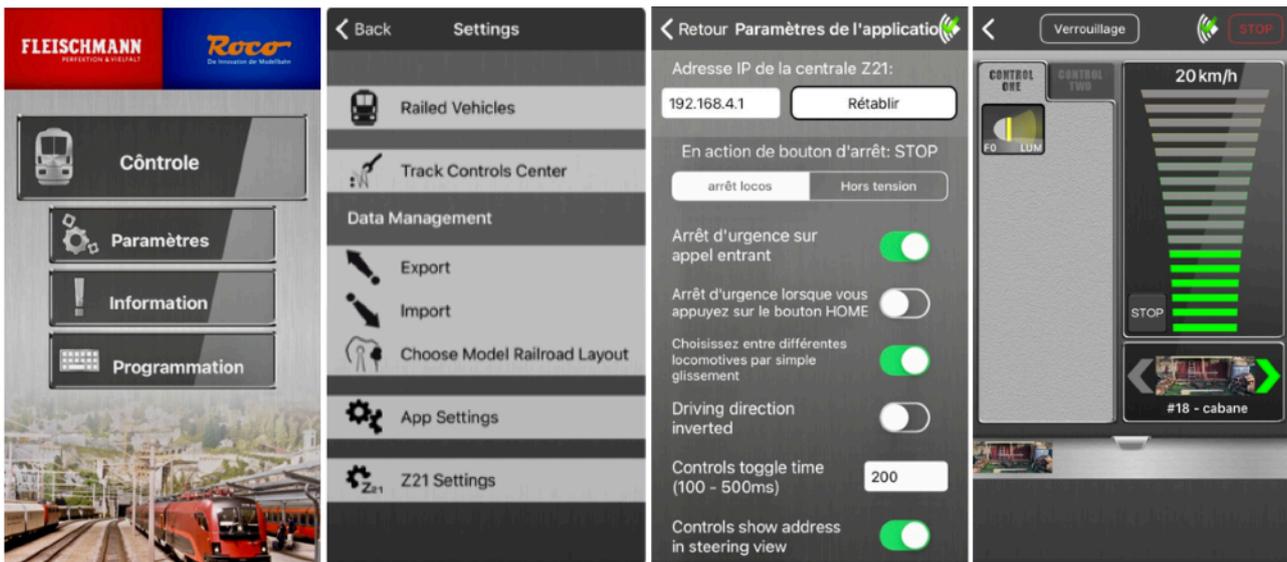
Quelques remarques : si le téléphone sonne et que vous prenez un appel ou si vous changez d'application en cours de jeu, votre locomotive va s'arrêter toute seule !!

Vous remarquerez que lors des connexions suivantes, l'application se souvient des réglages.

2) Exemple avec Z21 Mobile

Le Wifi doit être aussi lancé en préalable.

On trouvera aussi tous les détails sur cette application Z21-app



Le premier réglage est l'adresse IP de LaBox dans "App Settings » Adresse **192.168.4.1**, port **44444**

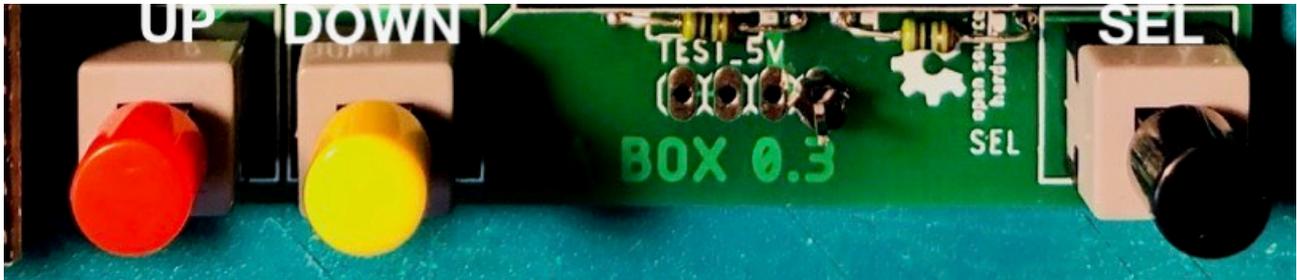
Puis le choix d'une loco dont on connaît l'adresse DCC (soit dans la liste préexistante, soit en la créant)
Enfin le pilotage selon le même principe avec le curseur de vitesse, la direction et les fonctions.

Quelques remarques : l'adresse IP de LaBox étant enregistrée dans l'application, aux lancements suivants l'application se connecte toute seule sans passer par un écran de connexion. Mais, contrairement à Withrottle, si on change d'application, la locomotive ne s'arrête pas tout de suite (il faut attendre que LaBox s'aperçoive de la perte de session avec l'application lorsqu'on la quitte). De même lorsque le téléphone sonne, mais il y a des réglages possibles à explorer...

3) Comment trouver l'adresse DCC d'une locomotive (dont on ne connaît pas l'adresse) ?

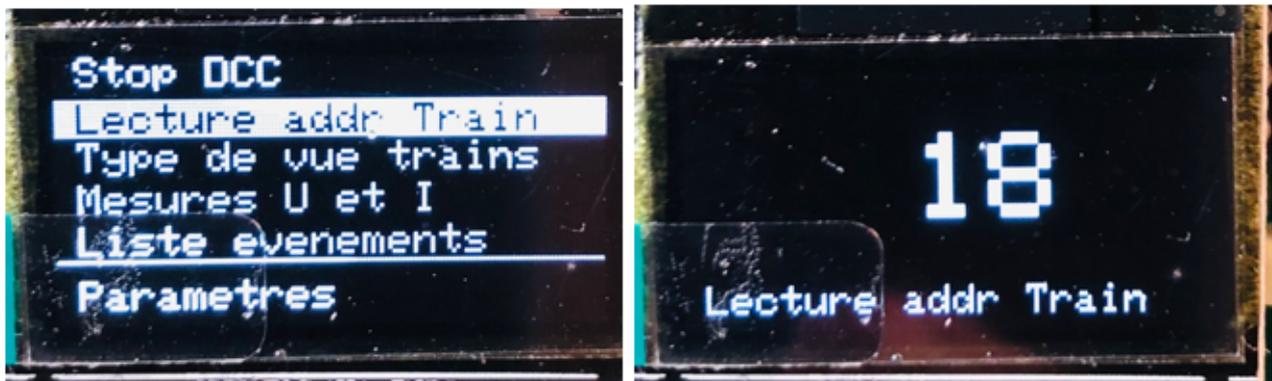
Et bien c'est là que LaBox est géniale :

On se sert des boutons de Labox :



Au préalable il faut que LaBox soit connectée sur les rails et que les voyants DCC soient allumés (en connectant l'application smartphone, même si on ne connaît pas -encore- l'adresse de la loco). Attention, une seule loco doit être posée sur les rails.

On premier appui sur le bouton "SEL" fait apparaître une liste de fonctions. Avec le bouton "DOWN" on sélectionne "Lecture addr Train". Puis on appuie sur SEL.



Au bout de quelques secondes, l'adresse DCC apparaît sur l'écran, après avoir vu la loco gigoter sur les rails (c'est le norme DCC décrite dans NMRA).

La lecture est répétée une 2ème et une 3ème fois pour une réussite plus certaine. En cas d'erreur (message "ERROR" affiché), on peut incriminer un mauvais contact entre les rails et la loco : je préconise de mettre un doigt léger sur la loco pour appuyer plus fermement les roues sur les rails. Mais ce sujet sera amplement décrit sur le forum.

Voilà vous avez son adresse, il ne reste plus qu'à la programmer dans l'application et piloter votre loco.

LaBox autorise plusieurs locomotives sur le même réseau donc plusieurs smartphones en même temps, ce qui permet de jouer à plusieurs avec sa propre manette associée à son propre train.

Dans ce cas, vous pouvez changer le "Type de vue trains" avec les boutons pour afficher 1, 2 ou 3 curseurs sur l'écran Oled. Cet écran permet de vérifier la conformité du comportement de LaBox avec votre application sur smartphone.

Alors ça vous plait ? 😊

le chargement du logiciel

1) Téléchargement du dossier "LaBox" sur le dépôt Git de Locoduino

Il est ici : <https://github.com/Locoduino/LaBox>

Ce dossier doit être placé dans le dossier "libraries" de l'IDE Arduino. C'est une bibliothèque mais il n'est pas (pour le moment) installable par l'IDE (menu Croquis/Inclure une Bibliothèque/Gérer les bibliothèques).

2) D'autres bibliothèques sont nécessaires pour réaliser une compilation avec succès :

"Adafruit_GFX"

"Adafruit_SSD1306"

"OneButton"

"ArduinoJson"

Ces dernières bibliothèques sont installables par l'IDE Arduino (menu Croquis/Inclure une Bibliothèque/Gérer les bibliothèques).

3) Installer l'environnement ESP32 dans l'IDE Arduino :



Dans les Préférences de l'IDE Arduino, ajouter l'URL de gestionnaire de cartes supplémentaires suivantes : https://dl.espressif.com/dl/package_esp32_index.json

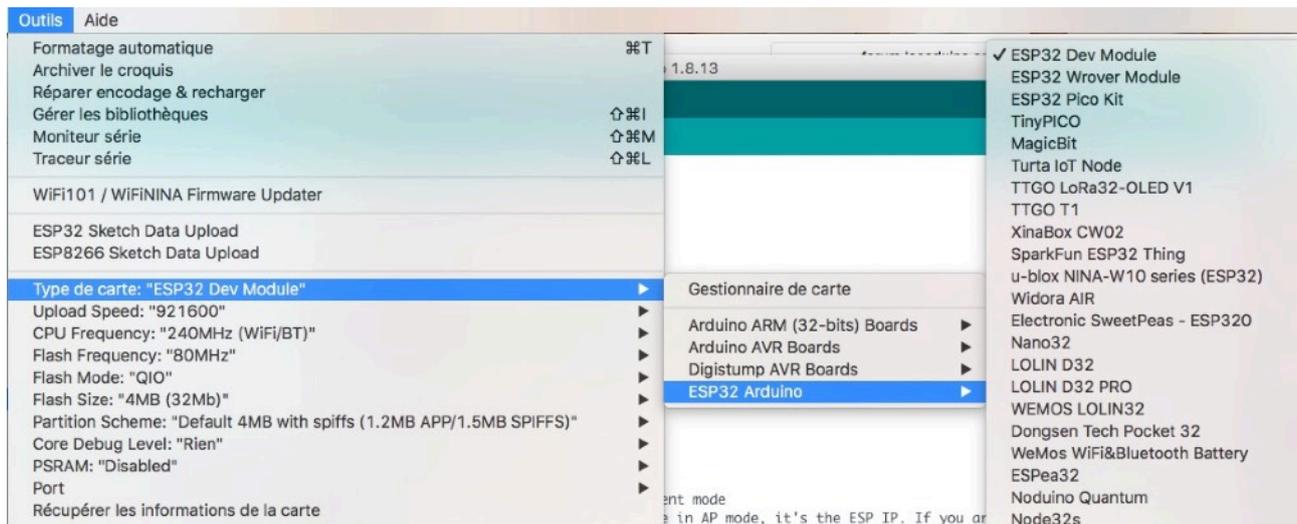
Puis redémarrez l'environnement IDE Arduino.

4) Choisissez le type de carte « ESP32 »

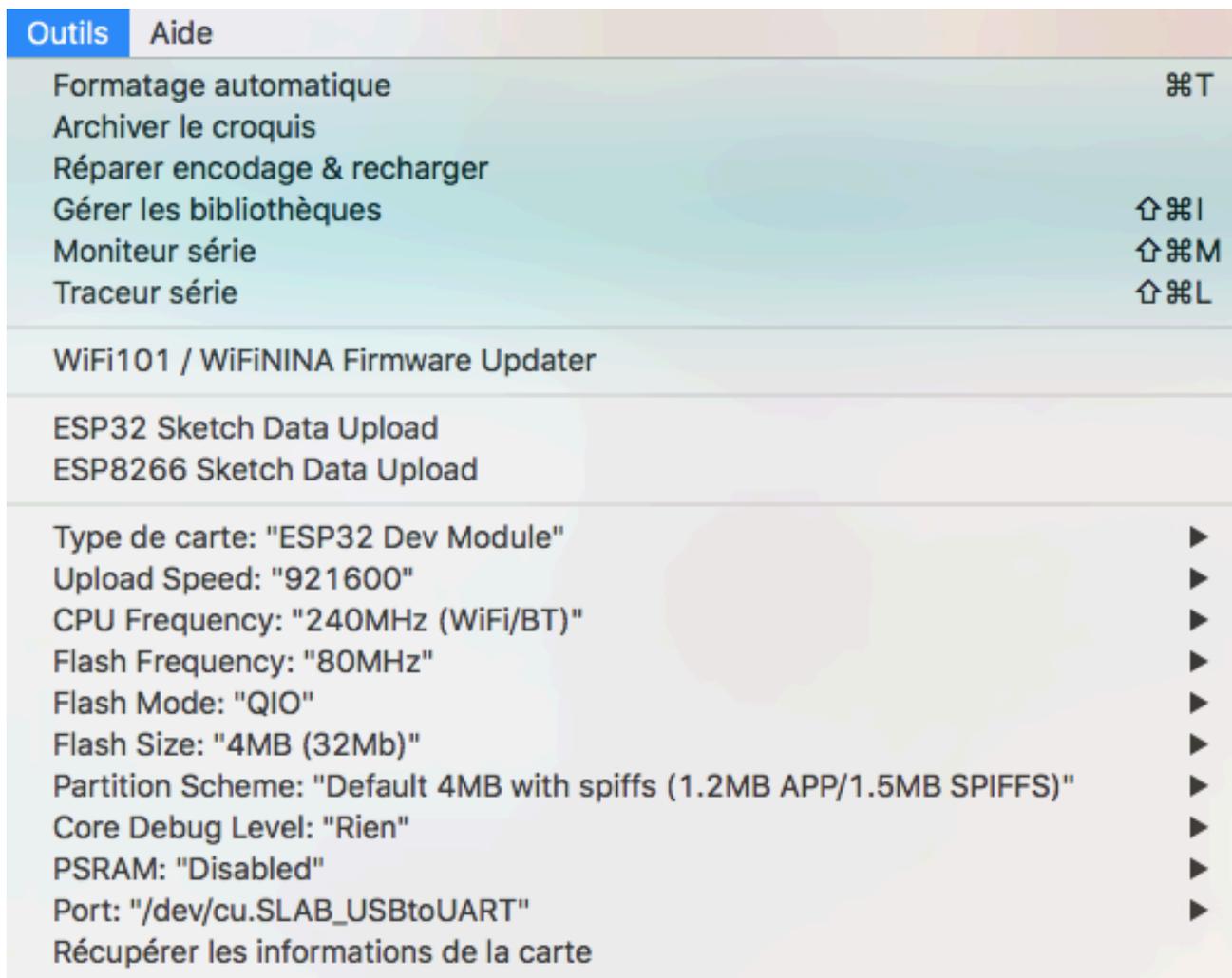


Dans le menu Outils/Type de cartes/Gestionnaire de cartes, cherchez et installez "esp32" de Espressif Systems (version 1.0.6 ou suivante).

Lorsque l'installation est terminée (cela peut prendre un peu de temps), sélectionnez la carte ESP32 Dev Module (la première dans la liste)



En ouvrant à nouveau le menu Outils/Type de cartes, on doit voir la description du module ESP32 Dev Module et on peut le relier à l'ordinateur par un câble USB et choisir le port correspondant :



Puis branchez un câble USB entre votre ordinateur et le port micro-USB de l'ESP32 de LaBox.

Sélectionnez le port série-USB correspondant à l'ESP32 de LaBox:

```
Ports série
/dev/cu.Bluetooth-Incoming-Port
/dev/cu.HC-05-DevB
/dev/cu.MacBookProdeDominique-B
✓ /dev/cu.SLAB_USBtoUART
```

4) Ouvrir l'exemple `LaBox.ino` qui se trouve dans le dossier `libraries/LaBox/examples/LaBox`

Si la compilation se passe bien, on obtient le texte suivant dans la fenêtre de compte-rendu de l'IDE :

Code: [Sélectionner]

Le croquis utilise 776454 octets (59%) de l'espace de stockage de programmes. Le maximum est de 1310720 octets.

Les variables globales utilisent 43588 octets (13%) de mémoire dynamique, ce qui laisse 284092 octets pour les variables locales. Le maximum est de 327680 octets.

On peut alors téléverser le logiciel dans l'ESP32

Au préalable, on peut ouvrir le moniteur série de l'IDE Arduino puis on lance le téléversement.

Dans la fenêtre en bas de l'IDE Arduino, on obtient ce message concluant :

Sur Mac, tout se déroule automatiquement !

Sur Windows il faut appuyer sur le bouton 'boot' de l'ESP32 pendant l'affichage des points d'attente pour que le téléversement démarre.

Code: [Sélectionner]

Le croquis utilise 776454 octets (59%) de l'espace de stockage de programmes. Le maximum est de 1310720 octets.

Les variables globales utilisent 43588 octets (13%) de mémoire dynamique, ce qui laisse 284092 octets pour les variables locales. Le maximum est de 327680 octets

`esptool.py v3.0-dev`

`Serial port /dev/cu.SLAB_USBtoUART`

`Connecting....._`

`Chip is ESP32-D0WDQ6 (revision 1)`

Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme
None
Crystal is 40MHz
MAC: 24:6f:28:7b:52:30
Uploading stub...
Running stub...

Stub running...
Changing baud rate to 921600
Changed.
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 8192 bytes to 47...
Writing at 0x0000e000... (100 %)
Wrote 8192 bytes (47 compressed) at 0x0000e000 in 0.0 seconds (effective 5120.4
kbit/s)...
Hash of data verified.
Compressed 18656 bytes to 12053...
Writing at 0x00001000... (100 %)
Wrote 18656 bytes (12053 compressed) at 0x00001000 in 0.2 seconds (effective
988.4 kbit/s)...
Hash of data verified.
Compressed 776576 bytes to 474600...
Writing at 0x00010000... (3 %)
Writing at 0x00014000... (6 %)
Writing at 0x00018000... (10 %)
Writing at 0x0001c000... (13 %)
Writing at 0x00020000... (17 %)
Writing at 0x00024000... (20 %)
Writing at 0x00028000... (24 %)
Writing at 0x0002c000... (27 %)
Writing at 0x00030000... (31 %)
Writing at 0x00034000... (34 %)
Writing at 0x00038000... (37 %)
Writing at 0x0003c000... (41 %)
Writing at 0x00040000... (44 %)
Writing at 0x00044000... (48 %)
Writing at 0x00048000... (51 %)
Writing at 0x0004c000... (55 %)
Writing at 0x00050000... (58 %)
Writing at 0x00054000... (62 %)
Writing at 0x00058000... (65 %)
Writing at 0x0005c000... (68 %)
Writing at 0x00060000... (72 %)
Writing at 0x00064000... (75 %)
Writing at 0x00068000... (79 %)
Writing at 0x0006c000... (82 %)
Writing at 0x00070000... (86 %)
Writing at 0x00074000... (89 %)
Writing at 0x00078000... (93 %)
Writing at 0x0007c000... (96 %)

```
Writing at 0x00080000... (100 %)
Wrote 776576 bytes (474600 compressed) at 0x00010000 in 7.3 seconds (effective
851.8 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 128...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (128 compressed) at 0x00008000 in 0.0 seconds (effective 1741.1
kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

Dans le moniteur série de l'IDE on obtient :

```
Server IP address: 192.168.4.1 (Locoduino LaBox) connected ! connectWifi
achieved.
Ino executing on core 1
Throttles command receivers executing on core 0
begin achieved
Signal started for Main track on pin 33
Main track DCC ESP32 started.
beginMain achived with pin 33
*** LaBox LIBRARY : 0.8.0
VERSION DCC++      : 2.0.0
COMPILED           : May 15 2021 17:13:23
  ENABLE(PWM): 32
  CURRENT: 36
Throttles -----
0 : Serial
1 : JMRI
2 : ThrottleWifi: Z21 - 1  WifiPort: 21105  WifiProtocol: UDP (Z21 converter : )
not connected
3 : ThrottleWifi: Z21 - 2  WifiPort: 21105  WifiProtocol: UDP (Z21 converter : )
not connected
4 : ThrottleWifi: Z21 - 3  WifiPort: 21105  WifiProtocol: UDP (Z21 converter : )
not connected
5 : ThrottleWifi: WiThrottle - 1  WifiPort: 44444  WifiProtocol: TCP (WiThrottle
converter)  start:60 end:62 not connected
6 : ThrottleWifi: WiThrottle - 2  WifiPort: 44444  WifiProtocol: TCP (WiThrottle
converter)  start:60 end:62 not connected
7 : ThrottleWifi: WiThrottle - 3  WifiPort: 44444  WifiProtocol: TCP (WiThrottle
converter)  start:60 end:62 not connected
```

Sur l'écran Oled de LaBox, on doit voir apparaître successivement les écrans suivants :



5) Réglage du "zéro" de la mesure de courant

En tournant le potentiomètre Pot de 500k à droite de l'écran Oled, cherchez à obtenir un affichage proche de 0 mA. Evitez principalement une valeur négative mais une valeur positive entre 2 et 5 mA conviendra parfaitement. La mesure de courant qui s'affiche sur l'Oled reste une valeur approximative qui donne une idée de la consommation de courant du matériel roulant sur la voie.

6) connexion d'un Throttle Withrottle (IOS)

Observez ce qui s'affiche dans la fenêtre moniteur de l'IDE

a- choix du réseau WiFi Locoduino LaBox:

```
dhcps: send_nak>>udp_sendto result 0
```

b- lancement de Withrottle et réglage sur le serveur LaBox (192.168.4.1:44444):

```
Converter : New client
5 -> VN2.0
5 -> RL0
5 -> PPA2
5 -> RCL0
5 -> PW12080n
DCCpp PowerOn
Throttles -----
0 : Serial
1 : JMRI
2 : ThrottleWifi: Z21 - 1 WifiPort: 21105 WifiProtocol: UDP (Z21 converter : )
not connected
3 : ThrottleWifi: Z21 - 2 WifiPort: 21105 WifiProtocol: UDP (Z21 converter : )
not connected
4 : ThrottleWifi: Z21 - 3 WifiPort: 21105 WifiProtocol: UDP (Z21 converter : )
not connected
5 : ThrottleWifi: WiThrottle - 1 WifiPort: 44444 WifiProtocol: TCP timeout:
60000 (WiThrottle converter) start:0 end:10 ip:192.168.4.2
6 : ThrottleWifi: WiThrottle - 2 WifiPort: 44444 WifiProtocol: TCP (WiThrottle
converter) start:60 end:62 not connected
```

7 : ThrottleWifi: WiThrottle - 3 WifiPort: 44444 WifiProtocol: TCP (WiThrottle converter) start:60 end:62 not connected
5 From Throttle : NiPhoneDB

c- reconnaissance de la connexion de Withrottle (iPhone):

Throttle 5 NiPhoneDB
5 -> <p1>*10
5 From Throttle : HUBDE241D3-07EA-4A60-BD6B-EE19
Throttle 5 HUBDE241D3-07EA-4A60-BD6B-EE19

d-choix de l'adresse DCC : 3 (ou reconnaissance automatique préalable et choix de l'adresse affichée sur l'écran Oled):

5 From Throttle : MT+S3<;>S3
Throttle 5 MT+S3<;>S3

5 -> MT+S3<;>
5 -> MTAS3<;>F00
5 -> MTAS3<;>F01
5 -> MTAS3<;>F02
5 -> MTAS3<;>F03
5 -> MTAS3<;>F04
5 -> MTAS3<;>F05
5 -> MTAS3<;>F06
5 -> MTAS3<;>F07
5 -> MTAS3<;>F08
5 -> MTAS3<;>F09
5 -> MTAS3<;>F010
5 -> MTAS3<;>F011
5 -> MTAS3<;>F012
5 -> MTAS3<;>F013
5 -> MTAS3<;>F014
5 -> MTAS3<;>F015
5 -> MTAS3<;>F016
5 -> MTAS3<;>F017
5 -> MTAS3<;>F018
5 -> MTAS3<;>F019
5 -> MTAS3<;>F020
5 -> MTAS3<;>F021
5 -> MTAS3<;>F022
5 -> MTAS3<;>F023
5 -> MTAS3<;>F024
5 -> MTAS3<;>F025
5 -> MTAS3<;>F026
5 -> MTAS3<;>F027
5 -> MTAS3<;>F028
5 -> MT+S3<;>V0
5 -> MT+S3<;>R1
5 -> MT+S3<;>s1

Locomotives -----

0 : Loco reg:1 id:3 max:128 +/-speed:0 functions:

e- variation de la vitesse avec le curseur de vitesse de Withrottle:

5 From Throttle : MTAS3<;>V3
Throttle 5 MTAS3<;>V3
DCCpp SetSpeed for loco 3 : 3/128)
5 From Throttle : MTAS3<;>V16
Throttle 5 MTAS3<;>V16
DCCpp SetSpeed for loco 3 : 16/128)
5 From Throttle : MTAS3<;>V26
Throttle 5 MTAS3<;>V26
DCCpp SetSpeed for loco 3 : 26/128)
5 From Throttle : MTAS3<;>V36
Throttle 5 MTAS3<;>V36
DCCpp SetSpeed for loco 3 : 36/128)
5 From Throttle : MTAS3<;>V39
Throttle 5 MTAS3<;>V39
DCCpp SetSpeed for loco 3 : 39/128)
5 From Throttle : MTAS3<;>V40
Throttle 5 MTAS3<;>V40
DCCpp SetSpeed for loco 3 : 40/128)
5 From Throttle : MTAS3<;>V41
Throttle 5 MTAS3<;>V41
DCCpp SetSpeed for loco 3 : 41/128)