

Documentation de l'ESP Smart Radio

Référence : Autre référence : Version : Date d'édition :	IMT/OLPS/OPENSERV/CONTENT/IAM.	<i>Vérifié par :</i>
Auteurs : Erwann Caroff		<i>Approuvé par :</i>
Résumé : Présentation et prise en main du boîtier ESP Smart Radio Mots clés : Live Objects, IoT, NodeMcu, objet connecté, smart Radio		

Le présent document contient des informations qui sont la propriété de la R&D d'Orange. L'acceptation de ce document par son destinataire implique, de la part de ce dernier, la reconnaissance du caractère confidentiel de son contenu et l'engagement de n'en faire aucune reproduction, aucune transmission à des tiers, aucune divulgation et aucune utilisation commerciale sans l'accord préalable écrit de la R&D d'Orange.

Documentation de l'ESP Smart Radio

I. Présentation

- **Veille technologique**

➤ Deux bases:

- Karadio: Développement Open source par: jp cocatrix

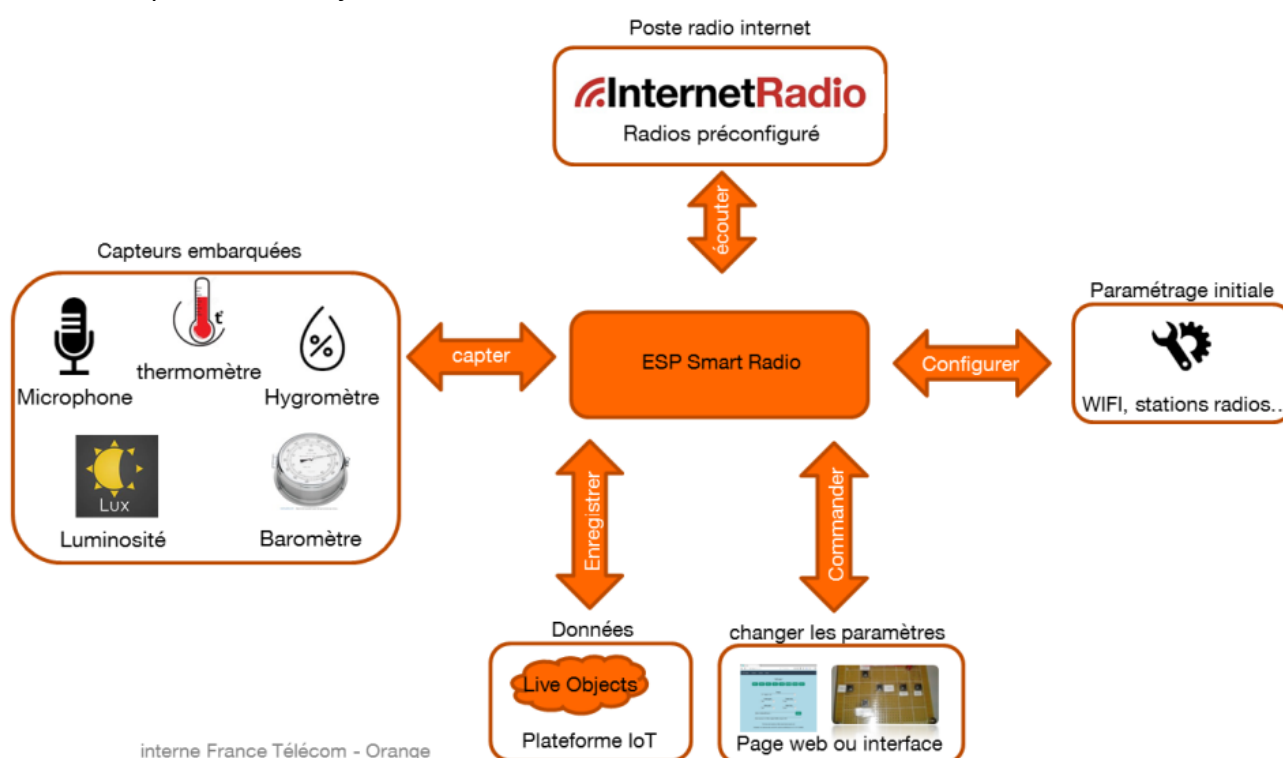


- EspRadio: Développement Open source par: Edzelf



- **Use case :**

Voici ce que réalise l'objet :



- **Capter et enregistrer les données sur LiveObjects**

Les données captées par l'objet sont envoyées sur le broker LiveObjects. On y retrouve deux trames de données envoyées à un intervalle de 30 secondes :

- ✓ Trame de données de la radio
- ✓ Trame de données sur l'état de la radio

Voici un exemple des données envoyées sur LiveObjects

06/06/2017 10:54:23	urn:lo:nsid:espWebRadio:3790567!StrucRadioEvent	{ "Station": "Radionova", "Song": "In onda TOP CLUB WAR" }
06/06/2017 10:54:23	urn:lo:nsid:espWebRadio:3790567!StateEvent	{ "Vol": 78, "State": "stoped", "Gateway": "Livebox-LB115", "Ip": "192.168.1.49" }

Les données peuvent être exploitée et générer des actions grâce au générateur d'évènements que possède LiveObjects

- **Commande de la radio :**

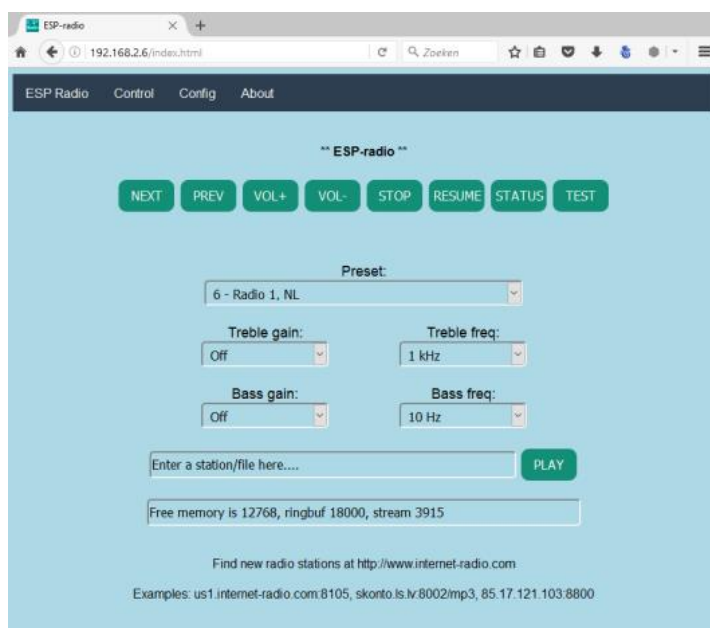
Il y a deux types de commandes sur la radio :

- ✓ Commande via une page web sur navigateur internet

Il faut se connecter via une interface numérique (pc, smartphone...) sur l'adresse IP de la radio avec un navigateur internet (s'assurer que la connexion wifi de la radio et de l'interface numérique soit les même).

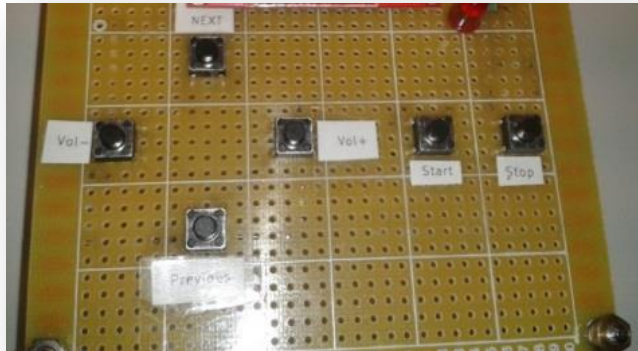
L'adresse IP est communiquée sur l'écran de la radio

Il est alors possible de commander en temps réel la radio.



- ✓ commande physique via une interface à bouton

La commande physique se réalise via des boutons classiques.



- **Ecran :**

L'écran affiche des données dans différentes phases :

- ✓ Phase d'initialisation (démarrage) :

La radio s'initialise et affiche à l'écran l'IP et la box de connexion :



- ✓ Phase de fonctionnement normal (la radio est lancée):

L'objet diffuse la radio sonore et l'écran affiche la station de musique ainsi que le titre de la chanson :



- ✓ Phase de changement de station :

L'objet se connecte à une station internet et l'afficheur affiche « stop »

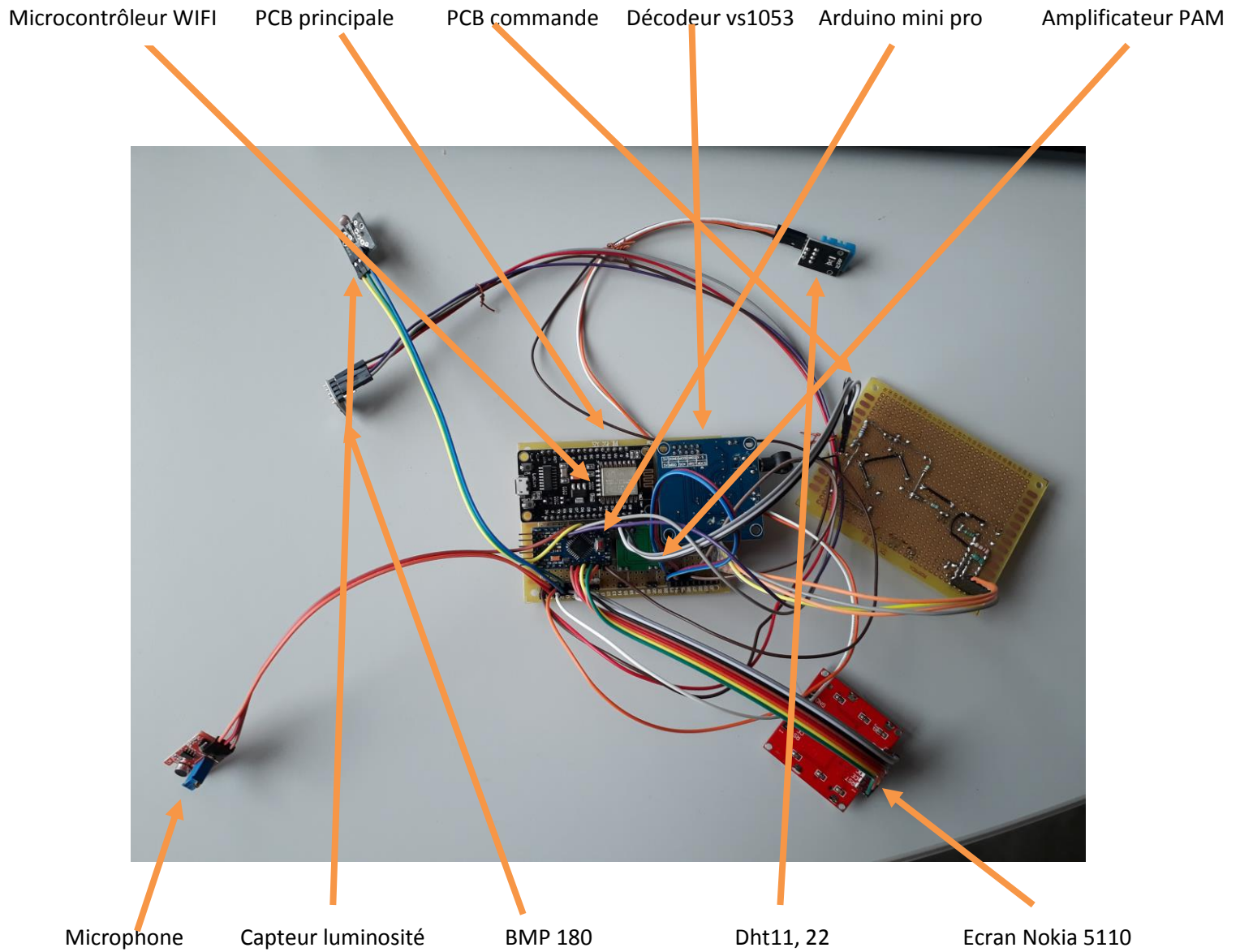
- ✓ Phase de commande :

L'objet est commandé par l'utilisateur et va donc afficher la commande réalisée.

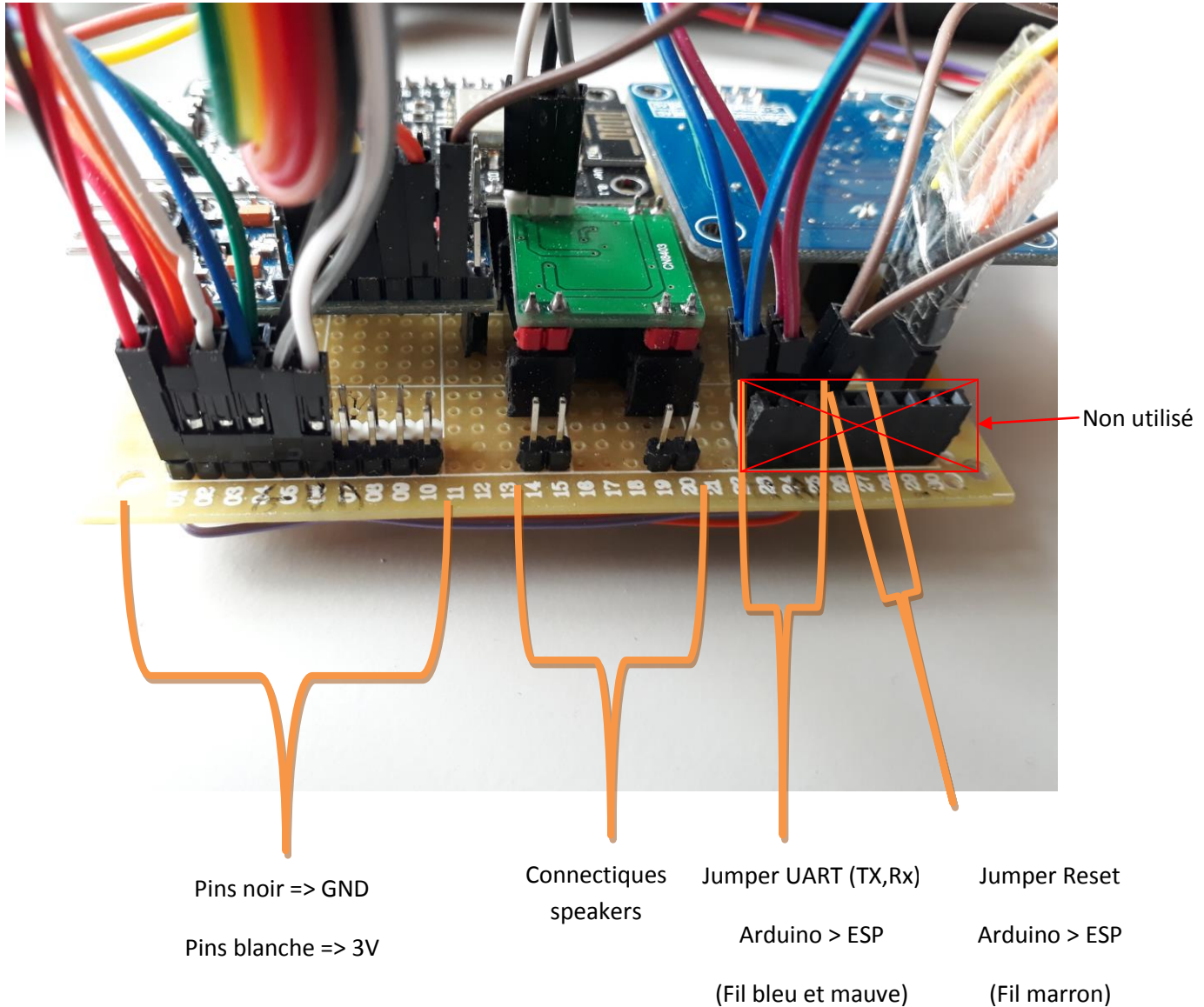
II. Branchements

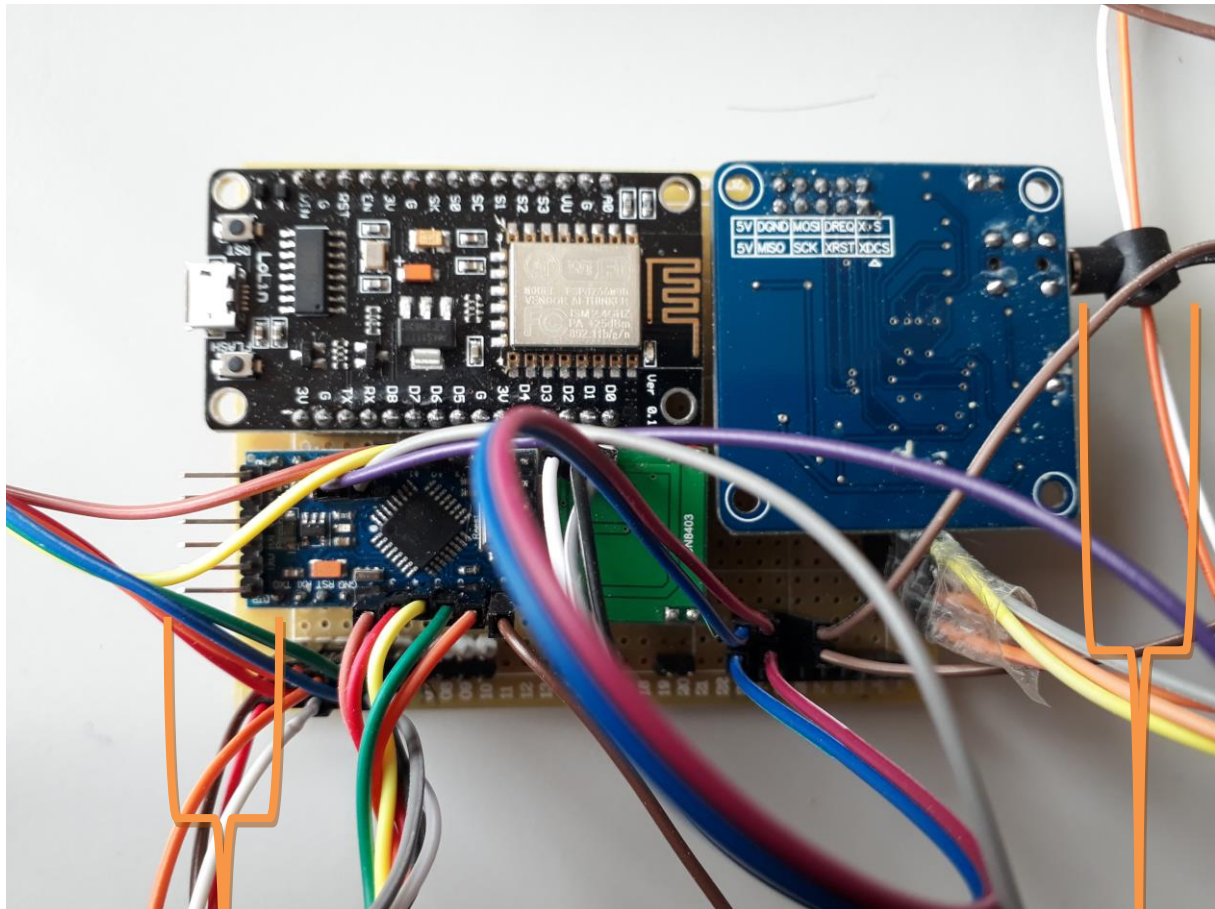
Composants ajoutés	Pins composant	Pins PCB principale	Pins PCB commande	Pins ESP	Pins Arduino mini pro	Pins Amplificateur PAM	Pin Décodeur vs1053
Capteur luminosité (photorésistance)	3V, GND, S	3V, GND			A0		
Capteur température/humidité (DHT11, 22)	3V, GND, S	3V, GND			8		
Capteur humidité/température/altitude (Bmp180)	3V, GND, SDA, A, SDL	3V, GND			bmp>Ard SDA > A4 SDL > A5		
Microphone	3V, GND, A0	3V, GND			A1		
Ecran Nokia 5110	1 à 8	Ecran > carte 8,7 > GND 6 > 3V			Ecran > Ard 1 > 8 2 > 7 3 > 3 4 > 4 5 > 7		
PCB principale > PCB commande		1>1 2>2 3>3 4>4	1>1 2>2 3>3 4>4				
Décodeur vs1003 > amplificateur PAM						1>1 2>2 3>3	jack
amplificateur PAM > speakers						1>1 2>2	

III. Descriptive visuel des composants



IV. Descriptive des connectiques du PCB principale



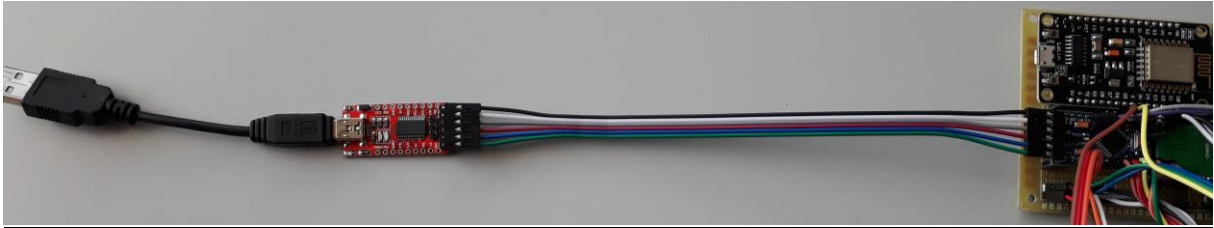


Port
programmation
Arduino Mini Pro

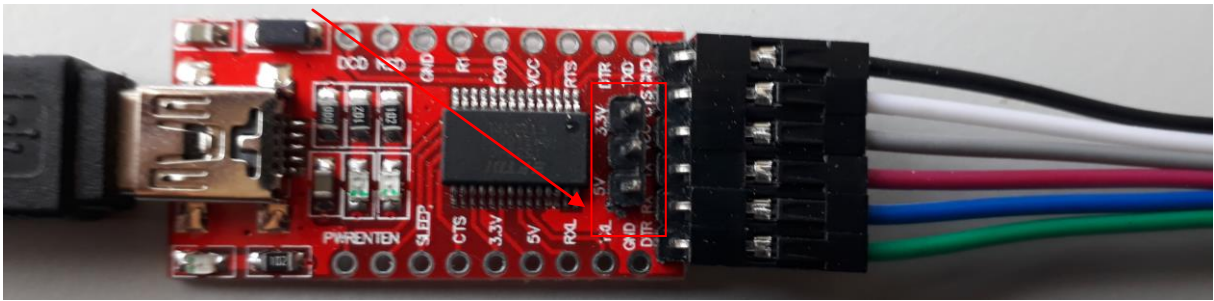
Sortie jack
(Signal son
analogique
vers amplificateur,
Isolé pour éviter les
bruits
perturbateurs)

V. Branchement du programmeur Arduino Mini Pro

- Convertisseur FTDI USB vers UART
- Branchement simple comme décrit dans la photo ci-dessous :



Attention : Ne pas oublier d'enlever le « jumper » d'alimentation si on alimente le PCB principale par le microcontrôleur NodeMcu (Deux alimentations en concurrence => alimentations grillées)



Si vous alimentez l'Arduino mini pro via le convertisseur FTDI, il faut choisir le mode 3.3V avec le jumper car l'Arduino mini pro fonctionne en 3.3V.



Configurez l'Arduino IDE (pour le convertisseur FTDI et Arduino mini pro) en choisissant dans l'onglet outils :

- Type de carte Arduino pro/pro mini
- Processeur : Atmega328(3.3V,8Mhz)
- Port : à choisir selon le port USB utilisé

Le choix de la carte et le processeur dépend de la carte à disposition.

Attention : Il est indispensable que l'Arduino fonctionne en 3.3V car l'esp8266 fonctionne exclusivement en 3.3V (à 5V, l'esp8266 sera endommagé via l'UART et le RESET, interaction électrique des deux cartes). Il est donc possible d'utiliser un processeur Atmega 168 (3.3V, 8Mhz) qui a moins de mémoire de programmation mais qui est moins chers.

Le programme se téléverse classiquement via l'onglet « Televersement ».

VI. Programmation du microcontrôleur (NodeMcu)

La programmation du microcontrôleur se déroule en deux phases :

- **téléversement du programme principal :**

Le programme se téléverse classiquement via l'onglet « Téléversement ». Le fichier du programme se trouve dans: *smart radio/ Microcontrôleur NodeMcu/ Esp_radio/ Esp_radio.ino*

- **Téléversement de la configuration :**

Pour que le microcontrôleur se configure automatiquement au démarrage (connexion au wifi, station radio pré configuré, améliorations audio...), il faut téléverser un fichier d'initialisation ayant les paramètres configurés par l'utilisateur.

Le fichier se trouve dans : *smart radio/ Microcontrôleur NodeMcu/ Esp_radio/ data/radio.ini*.

Voici le fichier d'initialisation :

```
###Connexion au broker Liveobjects
mqttbroker = liveobjects.orange-business.com # Broker to connect with
mqttport = 1883 # Portnumber (1883 is default)
mqttuser = json+device # (No) username for broker
mqttpasswd = 7ef7949fb37c454aa3087f63c276573d # (No) password for broker
mqtttopic = dev/data # Topic for receiving commands
mqttpubtopic = dev/data # IP will be published here
# api : 7ef7949fb37c454aa3087f63c276573d

### Configuration de la connexion WIFI (sélectionne automatiquement dans la liste ci-dessous le signal Wifi le plus fort)
wifi_00 = Livbox-1ffd/0123456789
wifi_01 = lb124bis/0123456789
wifi_02 = Livebox-LB115/0123456789
#wifi_03 = Livebox-43B1/0123456789
#wifi_04 = Livbox-2C2E/0123456789
#wifi_05 = Livebox-Reunion/0123456789

### Configurations Audios
volume = 72
toneha = 0
tonehf = 0
tonela = 0
tonelf = 0

### Stations Radios présélectionnées
preset = 0 # Start with preset 0
preset_00 = 213.136.90.169:8034 # 0 - Radio nova
preset_01 = 200.91.44.242:80 # 0 - PLAY FM
preset_02 = s3.imgradio.pro:80/RusHit48 # 0 - RUSSIAN HIT
preset_03 = 176.31.224.211:8720 # 0 - Nostalgie
preset_04 = 193.200.42.211:80/pulseHD.mp3 # 0 - Pulse radio
preset_05 = 91.197.138.3:80 # 0 - Allouette
```

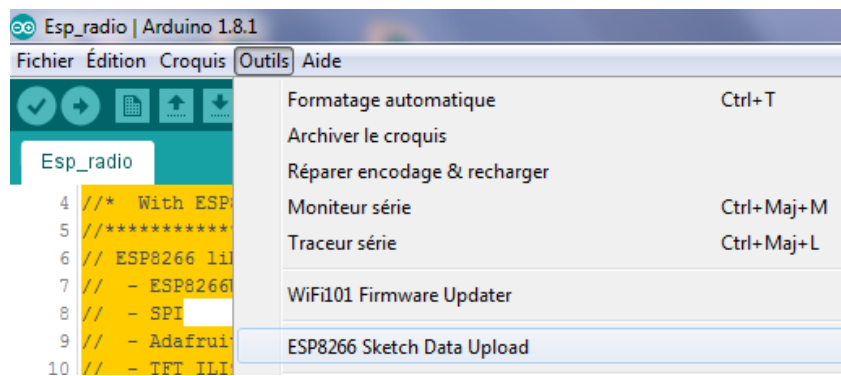
Pour se connecter à LiveObjects, il ne faut pas changer la configuration préconfiguré comme ci-dessous, sauf pour la clé API, nommé dans l'exemple : « mqttpasswd »

```
###Connexion au broker Liveobjects
mqttbroker = liveobjects.orange-business.com # Broker to connect with
mqttport = 1883 # Portnumber (1883 is default)
mqttuser = json+device # (No) username for broker
mqttpasswd = 7ef7949fb37c454aa3087f63c276573d # (No) password for broker
mqtttopic = dev/data # Topic for receiving commands
mqttpubtopic = dev/data # IP will be published here
```

Si l'objet n'arrive pas à se connecter au réseau, il se mettra en mode point d'accès et sera visible sur le réseau wifi avec le nom : « ESP_radio »

Il est donc possible de se connecter à ce réseau via le navigateur web à l'adresse « 192.168.1.4 ». Il est possible de commander ou configurer la radio via ce mode.

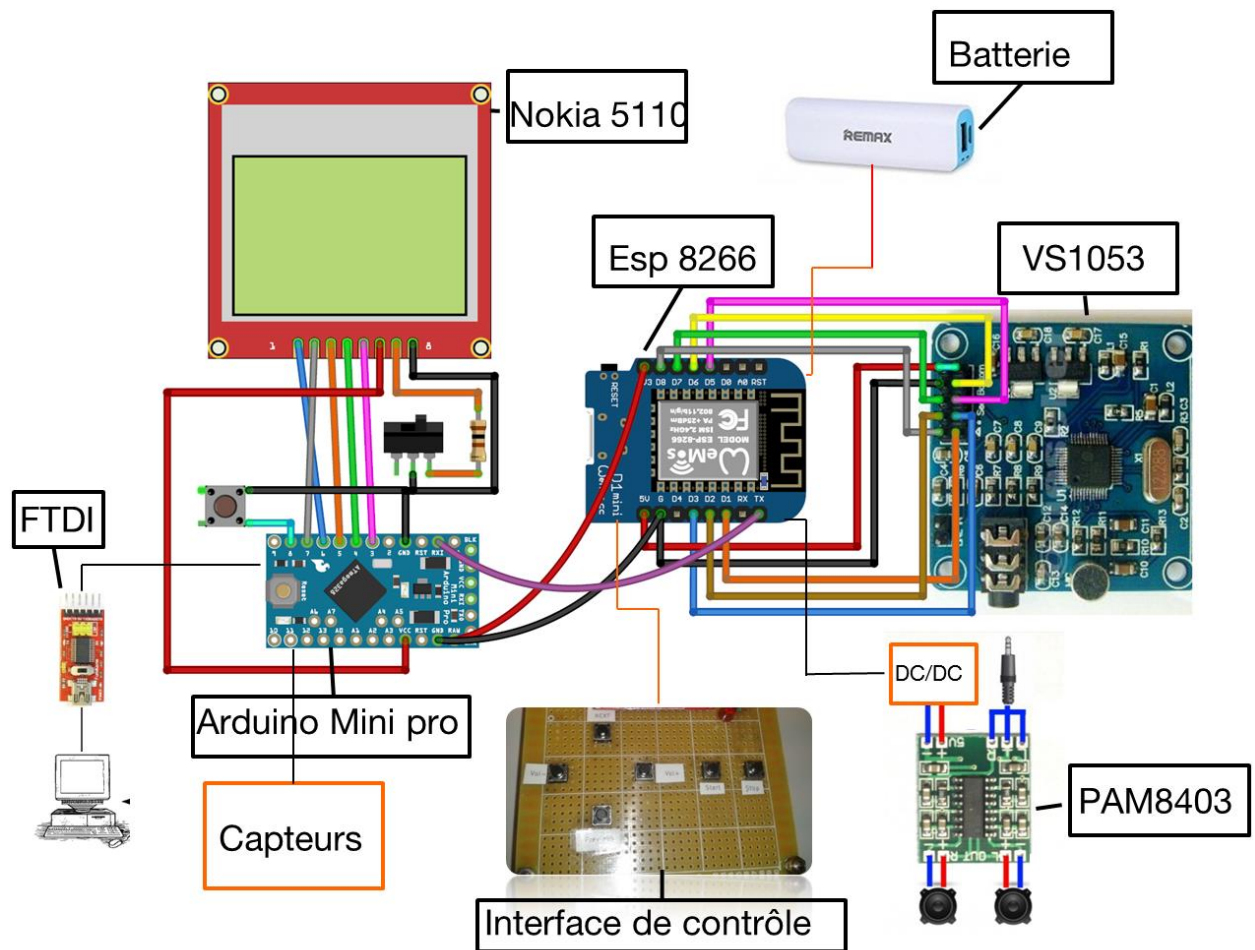
Pour téléverser il faut cliquer sur « *ESP8266 data upload* » dans l'onglet « *outils* » :



Attention :

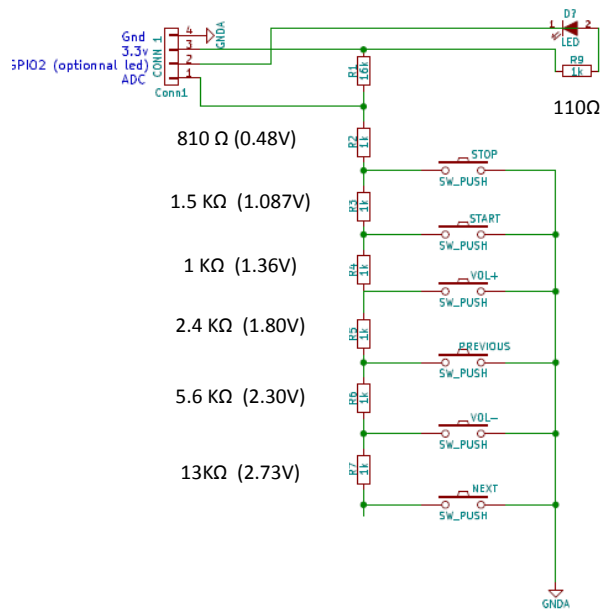
- Il faut s'assurer que le type de carte sélectionnée soit : **NodeMCU 1.0**
- Il faut s'assurer qu'il n'y a pas de monitoring du bus série durant le téléversement

VII. Schéma PCB principale



Dans l'alimentation de l'amplificateur audio il a du être ajouté un coupleur DC /DC pour réduire le bruit perturbateur sur le signal sonore.

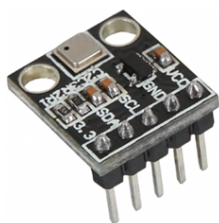
VIII. Schéma PCB commande



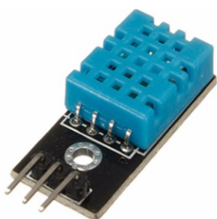
IX. Matériel

Materiel	Fonction	Cout
Arduino Mini pro	Microcontrôleur	1,82 €
Esp8266	Microcontrôleur WIFI	2,87 €
VS1053	Décodeur de format audio (Ogg vorbis, MP3, MP1, MP2, MPEG4, WMA, FLAC, WAW, MIDI) en signal analogique	5,85 €
Pam8403	Amplificateur Audio 2 * 3W, classe D	1,69 €
Nokia 5110	Ecran LCD	1,22 €
Convertisseur DC/DC	Isoler électriquement un composant	1,13 €
Interface FTDI	Convertisseur USB → UART (Tx, Rx...)	3,60 €
Consommables	Plaque à pastille, « Cavaliers duponts », connecteurs mâles et femelles, condensateurs, résistance, LED, fils	5,26 €
Boitier en bois		5,55 €
Haut-parleurs * 2		3,67 €
Total	projet sans capteur, PCB dédié et boitier	32,66 €
PCB dédié	30 € pour 5 PCB → Il faut 2 PCB pour notre projet	12 €

Materiel	Désignation	Fonction	Cout
KY-052 - BMP180	1	Capteur d'humidité et température	0,97 €
KY-015 – DHT11	2	Capteur de pression atmosphérique et température	0,82 €
KY-037 - Microphone	3	capteur de bruit	0,47 €
KY-018 - Photorésistance	4	Capteur de luminosité	0,34 €
Total		Capteurs uniquement	2,60 €
Total		Capteurs + matériel de base	35,26 €



1



2



3



4