

Projet 2 (2025) : Construction d'un système Linux embarqué complet pour Raspberry Pi 5 avec WiFi, MQTT, SSH, application PyQt5 et splash screen en utilisant Buildroot *ou* Yocto

Description générale

Dans ce projet, vous devez construire, **à partir de zéro**, un système Linux embarqué complet pour le **Raspberry Pi 5**. Vous êtes libres d'utiliser :

- **Buildroot**, ou
- **Yocto Project**.

Le système final devra intégrer :

- une interface graphique PyQt5 (application fournie),
- un point d'accès WiFi fonctionnel,
- un service DHCP/DNS,
- un broker MQTT local,
- un accès SSH,
- un splash screen personnel (créé par vous),
- un mécanisme de démarrage automatique via scripts ou services.

Vous devez concevoir toute la structure Buildroot/Yocto nécessaire (board directory, layers, post-build, configuration du rootfs, etc.). Seuls deux répertoires vous sont fournis : **etc/** et **usr/bin/**. Le reste doit être entièrement construit par vous.

Fichiers fournis

Seuls les répertoires suivants vous sont fournis :

```
overlay/  
  etc/  
    init.d/  
      S40network  
      S57app  
    hostapd/hostapd.config  
    mosquitto/mosquitto.conf  
    dnsmasq.conf  
    network/interfaces  
  usr/  
    bin/  
      TankApp
```

Remarques importantes :

- Aucun répertoire `board/raspberrypi/` n'est fourni.
- Aucun fichier `post-build.sh`, `post-image.sh`, `cmdline.txt` ou `config.txt` n'est fourni.
- **Aucun splash screen n'est fourni** : vous devez créer le vôtre.

Vous devez donc :

- créer la structure nécessaire pour intégrer cet overlay,
- décider où l'inclure dans Buildroot ou Yocto,
- générer les scripts et fichiers nécessaires au boot du Raspberry Pi 5.

Configurations fournies (extraits)

WiFi AP

```
interface=wlan0
driver=nl80211
ssid=TankControl
wpa_passphrase=12345678
hw_mode=g
channel=6
```

DHCP/DNS

```
interface=wlan0
dhcp-range=192.168.4.10,192.168.4.50,255.255.255.0,24h
```

MQTT

```
listener 1883
allow_anonymous true
log_dest stdout
```

Interfaces réseau

```
auto wlan0
iface wlan0 inet static
    address 192.168.4.1
    netmask 255.255.255.0
    pre-up iw dev wlan0 set type __ap
```

À vous d'interpréter ces configurations pour déterminer les services, bibliothèques et outils nécessaires.

Objectifs du projet

1. Construire une image bootable pour Raspberry Pi 5

Le système doit :

- booter,
- activer WiFi + DHCP/DNS + MQTT + SSH,
- lancer automatiquement l'application PyQt5.

2. Intégrer l'overlay fourni

Vous devez construire :

- pour Buildroot : un répertoire `board/` et un mécanisme d'overlay,
- pour Yocto : un layer, une recette, ou un mécanisme équivalent.

3. Créer et afficher votre splash screen

Vous devez :

- créer une image (PNG, etc.),
- l'intégrer dans le système,
- afficher l'image au tout début du boot (avant l'application PyQt5).

Méthode totalement libre (framebuffer, pspplash, plymouth, fbv, script, service, etc.).

4. Déduire les dépendances

Vous devez identifier par vous-même :

- les paquets nécessaires,
- les services obligatoires,
- les bibliothèques Python et Qt,
- les options du noyau,
- les utilitaires réseau requis.

Aucune liste n'est fournie.

Tests et validation

Le système final doit permettre :

- connexion au WiFi AP,
- obtention d'une IP via DHCP,
- publication/réception MQTT,
- accès SSH,
- affichage automatique de votre splash screen,
- lancement automatique de l'application PyQt5.

Livrables attendus

1. **Image complète du système** (sdcard.img ou équivalent).
2. **Fichiers de configuration du système de construction :**
 - **Buildroot :**
 - defconfig,
 - .config,
 - répertoires créés (board/, scripts/, overlays/, etc.).
 - **Yocto :**
 - layer complet,
 - fichiers local.conf, bblayers.conf,
 - recettes et bbappends créés.
3. **Votre splash screen personnel :**
 - image créée par vous,
 - méthode permettant l'affichage au démarrage.
4. **Application PyQt5** (si modifiée).
5. **Rapport (6–10 pages)** comprenant :
 - justification du choix Buildroot/Yocto,
 - intégration de l'overlay,
 - organisation du rootfs,
 - dépendances identifiées,
 - débogage et erreurs rencontrées,
 - preuves du bon fonctionnement : WiFi, DHCP, MQTT, SSH, GUI, splash screen.

Conseils

- Commencez par le boot minimal.
- Ajoutez les services un par un.
- Utilisez la console série pour déboguer.
- Travaillez de façon incrémentale.