

Fiche technique

Buildroot



Table des matières

1. Présentation	3
2. Matériel utilisé	3
3. Installation logicielle.....	4
3.1. Installation de l'environnement de travail	4
3.2. Installation et utilisation de Buildroot.....	4
3.2.1. Installation	4
3.2.2. Configuration	4
3.2.3. Package personnalisé.....	5
4. Utilisation de Tensorflow Lite sur Buildroot.....	6
4.1. Création du package contenant le projet	6
4.1.1. Model.tflite	6
4.1.2. Main.cpp	6
4.2. Lancement sur le Raspberry.....	6

1. Présentation

Cette fiche technique présente l'utilisation faite de « Buildroot » pendant ce projet, le but étant de construire un OS pour le Raspberry Pi 5 qui serait plus léger qu'un OS Raspberry normalement utilisé sur ce type de carte.

Cet OS généré par Buildroot permettra de choisir uniquement les dépendances, les bibliothèques et autres programmes nécessaires pour le fonctionnement de la carte Raspberry Pi 5 et de son accélérateur de programme d'intelligence artificielle Hailo-8L AI Kit.

L'objectif final de cette partie du projet est de lancer un algorithme d'intelligence artificielle depuis le Buildroot de la carte Raspberry avec l'accélérateur Hailo-8L AI Kit.

2. Matériel utilisé

- Raspberry Pi 5
- Hailo-8L AI Kit
- Dernière version stable du Buildroot (02/2025 au temps du projet)
- Une caméra USB
- Une alimentation adaptée (USB-C 5V/5A)

3. Installation logicielle

3.1. Installation de l'environnement de travail

L'installation et l'utilisation de Buildroot n'est possible que sur un OS Linux.

Si vous n'avez pas encore d'environnement Linux à disposition, nous recommandons l'utilisation d'une machine virtuelle Ubuntu avec 25-50 Go de mémoire étant donné que l'image construite fera environ 15 Go.

3.2. Installation et utilisation de Buildroot

3.2.1. Installation

- Se rendre sur le site officiel de Buildroot (cf. [\[1\]](#)) et télécharger la dernière version stable ou la version utilisée pendant ce projet qui est celle du 02/2025.
- Extraire le fichier puis l'ouvrir dans le terminal : Vous avez maintenant accès aux commandes Buildroot (make, make menuconfig, etc.)

3.2.2. Configuration

- Installer la configuration de Buildroot pour un Raspberry Pi 5 : **'make raspberrypi5_defconfig'**

Si un autre Raspberry Pi est utilisé, d'autres configurations de base existent sur Buildroot, on peut les lister avec **'make list-defconfigs'** (à combiner avec un 'grep' pour n'avoir que les résultats qui nous intéressent)

Pour ne pas avoir de problème ultérieurement, il est conseillé de tout de suite modifier la taille de l'image générée par Buildroot : **'make menuconfig'** → Filesystem Images → Exact size : mettre 15000M pour 15 Go par exemple.

Vous pouvez déjà commencer à lancer la compilation qui est assez longue avec la commande 'make' (~2-3h).

- En plus de la configuration de base, quelques outils doivent être inclus dans Buildroot.

Pour sélectionner ces outils, il faut tout d'abord se rendre dans le menu de Buildroot avec **'make menuconfig'**

Nous devons d'une part sélectionner les outils nécessaires pour le fonctionnement du modèle d'IA fait avec Tensorflow et d'autre part nous devons sélectionner les outils nécessaires pour le fonctionnement de l'Hailo-8L AI Kit, **les deux parties peuvent être testées séparément pour résoudre plus facilement les erreurs.**

Outils à sélectionner pour la partie 'utilisation du modèle d'IA' :

- Toolchain → Enable Fortran support
- Target Packages → Libraries → Other → Tensorflow Lite
- Target Packages → Libraries → Graphics → opencv4
- Target Packages → tensorflow-projects (cf. [\[2\]](#))

Outils à sélectionner pour la partie 'Hailo-8L AI Kit' :

- Packages personnalisés Hailort, Hailo-pci et Hailo8-firmware (cf. [\[2\]](#))

3.2.3. Package personnalisé

Dans cette partie, nous allons voir comment créer un package qui n'existe pas sur Buildroot.

Tout d'abord, nous pouvons voir la liste des packages existants dans le sous dossier 'package' situé à la racine du projet. Ce dossier regroupe tous les packages sélectionnables dans le menu '**make menuconfig**'

Pour créer un package personnalisé il faudra suivre la même structure que les packages déjà présent et donc créer manuellement un dossier 'meson' par exemple et ajouter à l'intérieur de ce dossier un fichier Config.in ainsi qu'un makefile 'meson.mk'

Une fois que le package est créé, il faut appeler son Config.in dans le fichier package/Config.in avec la ligne '**source "package/meson/Config.in"**' (On peut par exemple ajouter la ligne dans Target Packages)

Une fois que c'est fait, il ne reste plus qu'à le sélectionner dans le menu de Buildroot (make menuconfig) et à compiler le package en faisant make <nom du package>

Les packages créés pendant ce projet sont disponibles sur github (cf. [\[2\]](#)).

4. Utilisation de Tensorflow Lite sur Buildroot

Tensorflow Lite étant présent sur la version officielle de Buildroot, il est donc plus facile de commencer l'utilisation d'intelligence artificielle sur Buildroot avec un modèle entraîné à partir de cette librairie.

Attention :

- Il faut faire attention aux modules présents dans Tensorflow Lite (pas de Keras par exemple)
- Si aucun compilateur supplémentaire n'est ajouté le compilateur utilisé pour construire le modèle sera le compilateur C++, présent par défaut dans Buildroot

Les étapes suivantes montrent la méthode à suivre pour utiliser un modèle basique créé avec tensorflow lite.

4.1. Création du package contenant le projet

Le package « tensorflow-projects » présent sur le GitHub (cf. ^[3]) contient un exemple basique de l'utilisation de tensorflow sur Buildroot

4.1.1. Model.tflite

Ce premier fichier est le modèle qui sera utilisé dans le projet.

Pour obtenir ce fichier, il faut entraîner une IA sur l'ordinateur hôte et récupérer le modèle généré sous la forme d'un .tflite

Il suffira ensuite de placer ce binaire dans le package du projet dans Buildroot.

4.1.2. Main.cpp

C'est dans ce fichier que l'on utilisera le modèle .tflite, ce programme n'utilisera que les outils et librairies présents sur Buildroot, il faudra donc penser à les cocher dans le 'make menuconfig' (ex : opencv).

La compilation par Buildroot génèrera un exécutable qui sera installé sur la cible de Buildroot

4.2. Lancement sur le Raspberry

Tout d'abord se donner les permissions d'exécuter le fichier avec chmod

Ensuite charger manuellement les modules noyau nécessaires, dans l'exemple sur le GitHub une caméra USB est utilisée, il faudra donc charger le module 'uvcvideo'

5. Liens

^[1] : <https://buildroot.org/>

^[2] : <https://github.com/florianVit/Projet-Raspberry-PI-Buildroot-IA-Hailo-8L/tree/main/Package-Buildroot>

^[3] : <https://github.com/florianVit/Projet-Raspberry-PI-Buildroot-IA-Hailo-8L/tree/main/Package-Buildroot/tensorflow-projects>