Création d'un environnement autonome sur SDCARD

1) Problématique

Il est intéressant dans un environnement embarqué de disposer d'une carte qui soit tout à fait autonome sans à proximité un PC avec la suite logiciel VIVADO.
On se propose lors de cette séance de placer sur la carte SD, les éléments logiciels nécessaires à un démarrage autonome de la carte

2) Mécanisme de démarrage de la carte ZYBO

Pour rendre autonome la carte le processeur CPU0 charge et execute une image du boot du ZYNQ qui inclut :

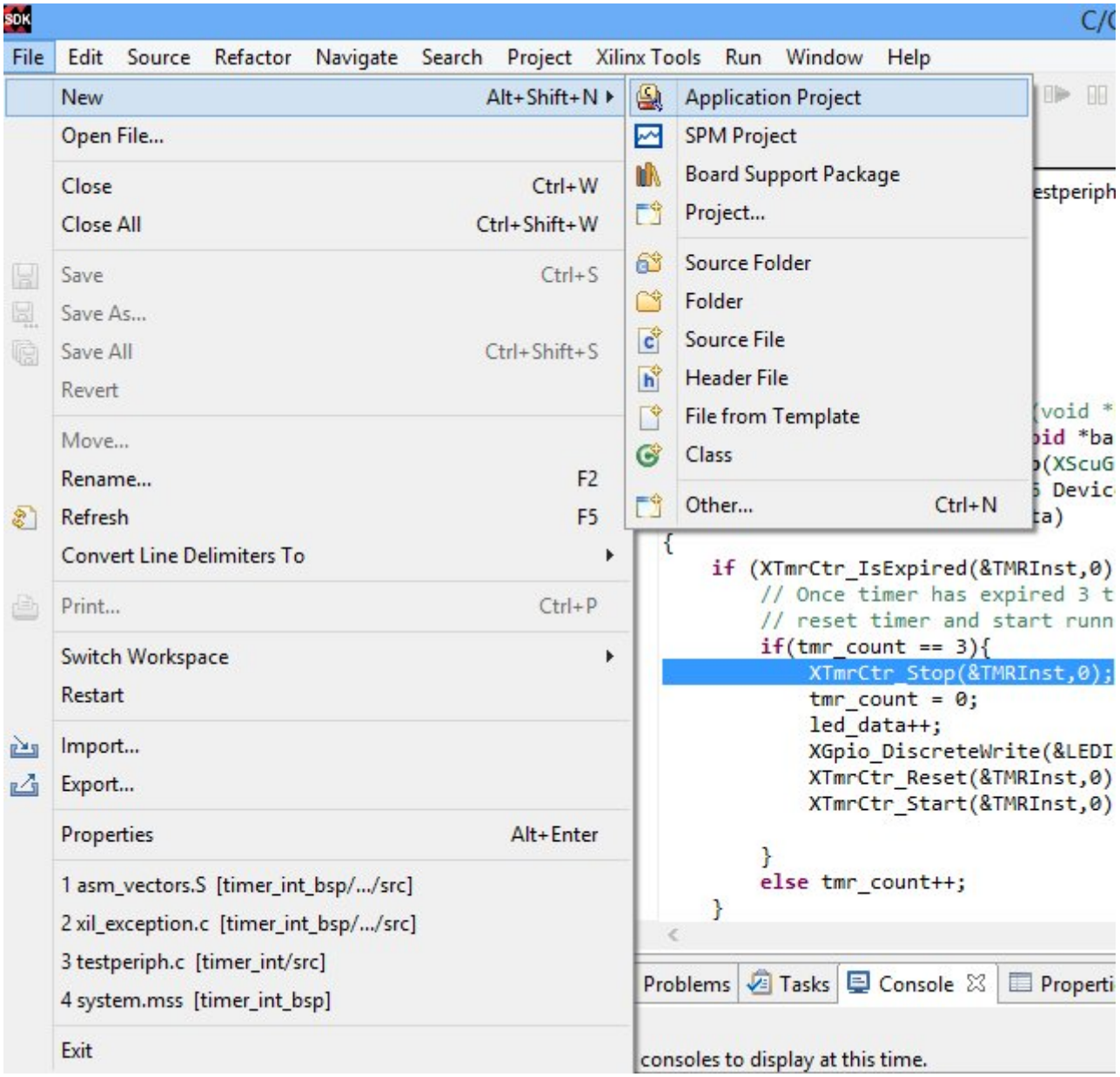
- ✓ Le FSBL (First Stage Bootloader) premiere étage du bootloader
- ✓ Un bitstream pour la configuration de la logique programmable (optionnel)
- ✓ Les applications utilisateur

Après que la carte ZYBO soit alimentée et que le RESET du ZYNQ soit effectif, le processeur CPU0 commence à exécuter un code interne en lecture seul appelé le BOOTROM. Le bootrom va vérifier l'état des cavaliers du connecteur JP5.
Ce connecteur JP5 permet de sélectionner le type démarrage de la carte ZYNQ suivant plusieurs modes de démarrage:

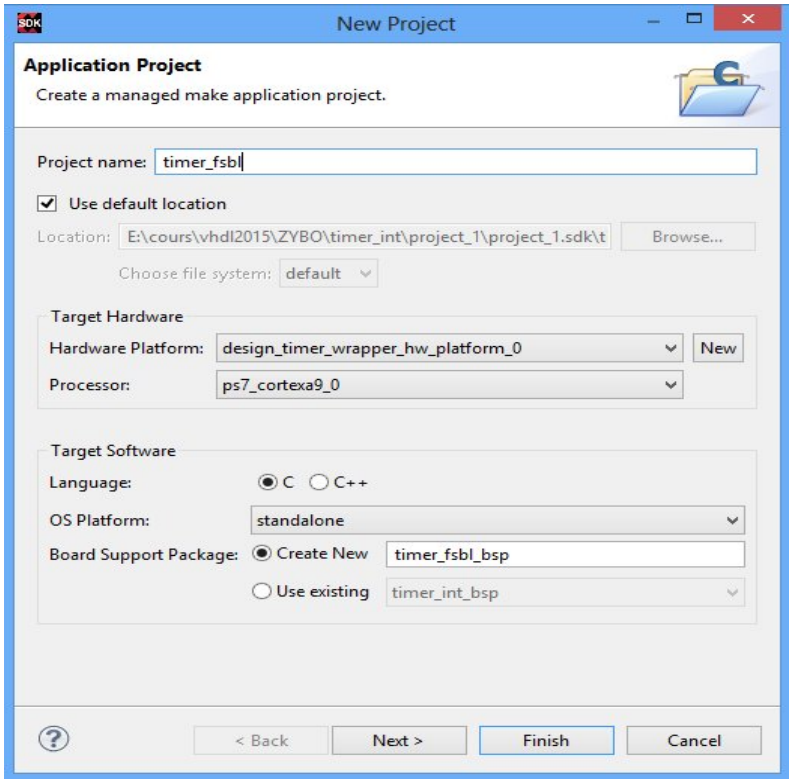
- ✓ SDCARD
- ✓ QSPI grâce à de la mémoire flash implanté sur la carte ZYBO
- ✓ JTAG ou la carte ZYBO attendra que la programmation de la carte se fasse par un PC

Si le bootrom est exécuté suite à l'événement RESET l'état précédent (SDCAR, QSPI, JTAG) est pris en compte. Par la suite le BOOTROM copie le FSBL à partir de la SDCARD ou de la flash vers la RAM interne OCM de l'APU
Le FSBL s'exécute et configure le PS. Si un bitstream est présent dans l'image de boot du ZYNQ il est lu et permet de configurer la logique programmable PL. Au final l'application utilisateur est chargée dans la mémoire à partir de l'image de boot du ZYNQ.
L'applicatif est alors exécuté. Cet applicatif peut être un simple programme comme un second chargeur de boot utilisé par un système d'exploitation

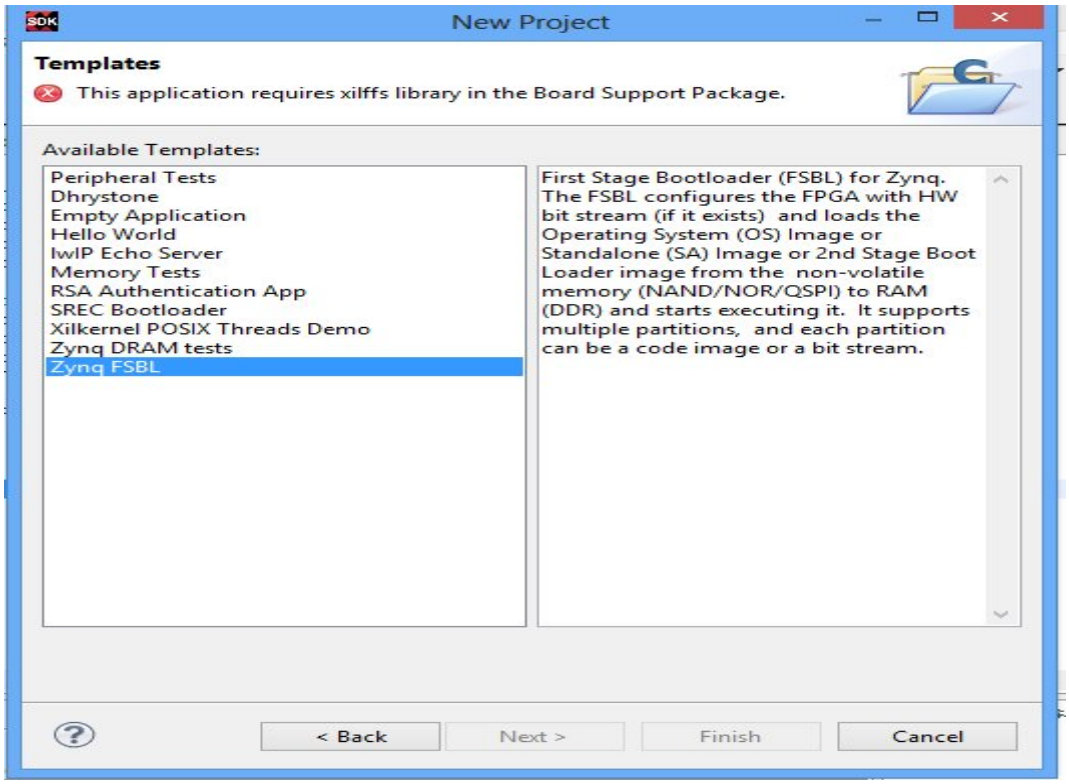
La première partie consiste à créer un bootloader nommé FSBL



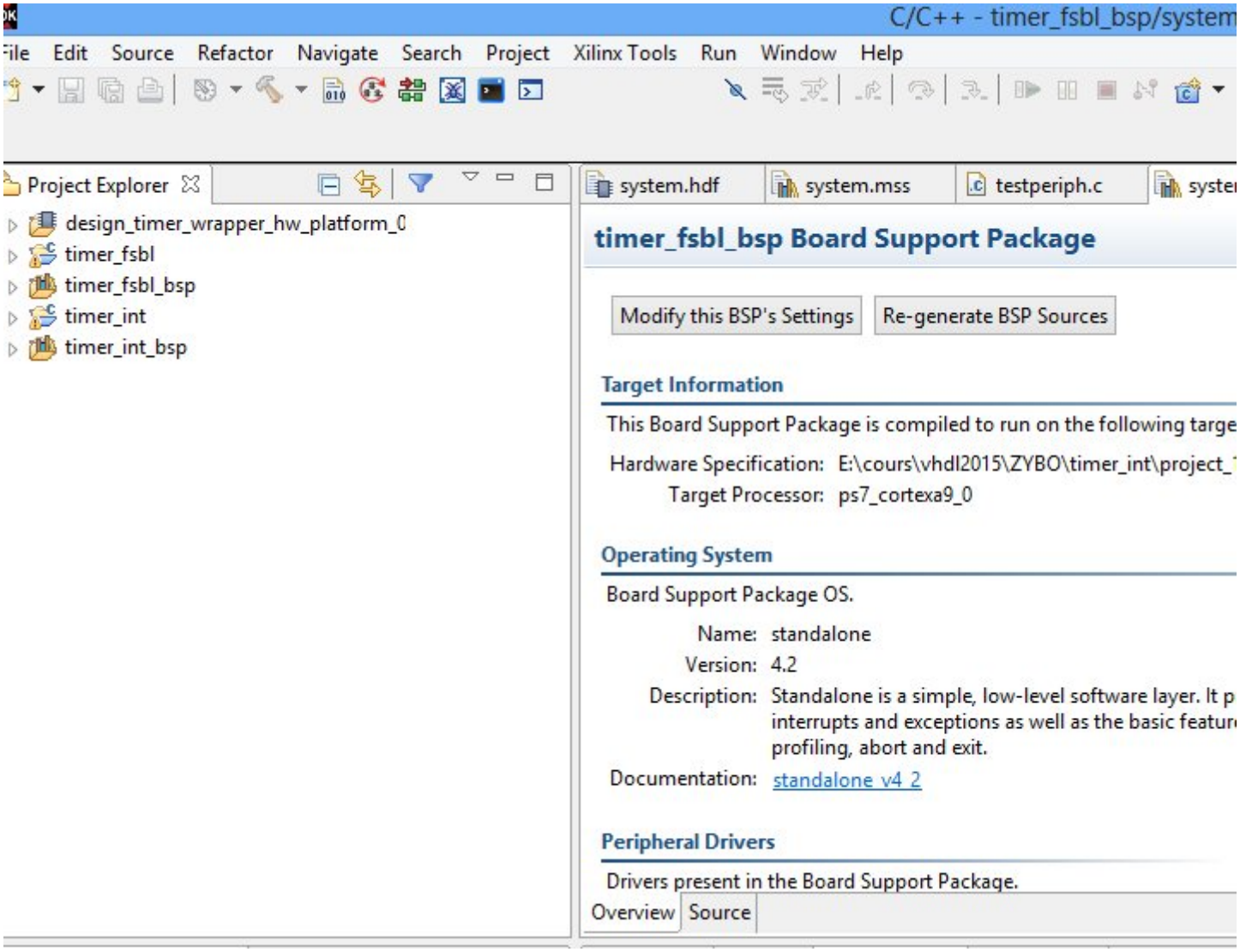
Le bootloader sera nommé timer_fsbl, on choisira la création d'un nouveau board support package adapté à la mise en place du bootloader



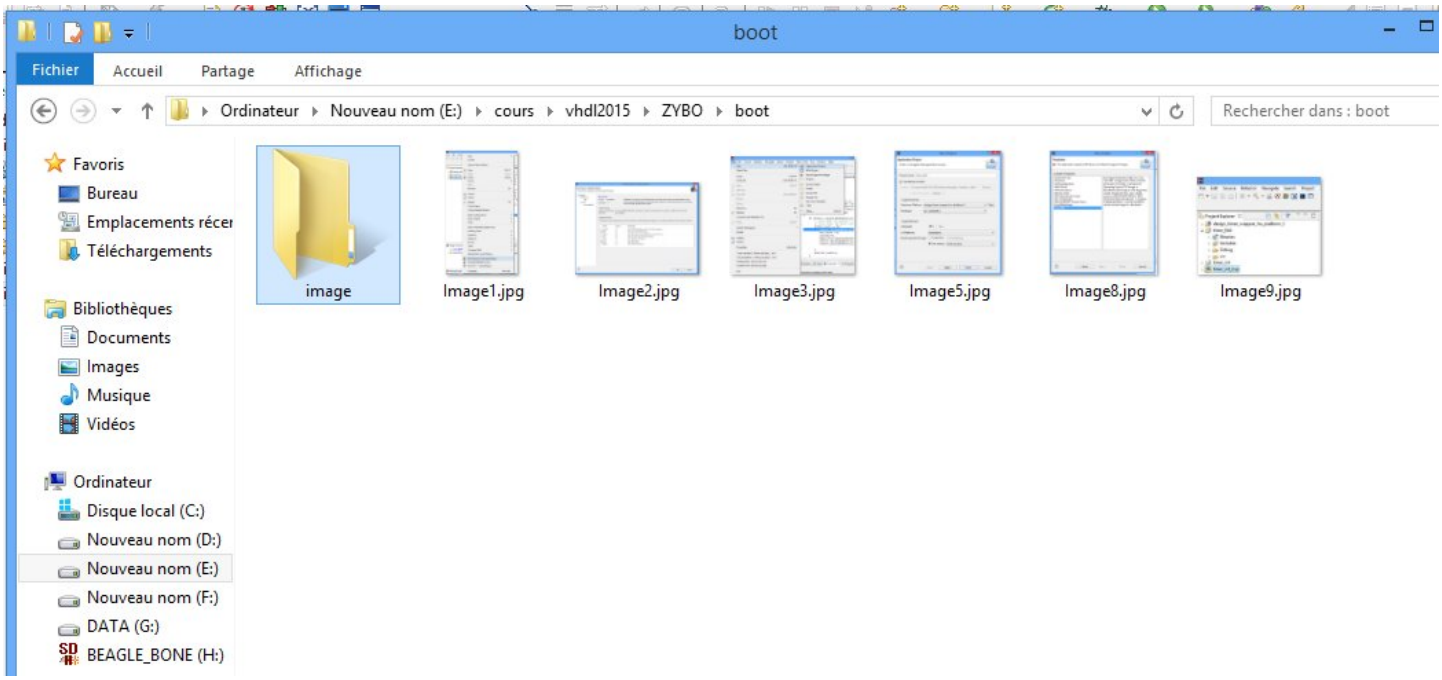
On choisit une application du type ZYNQ FSBL



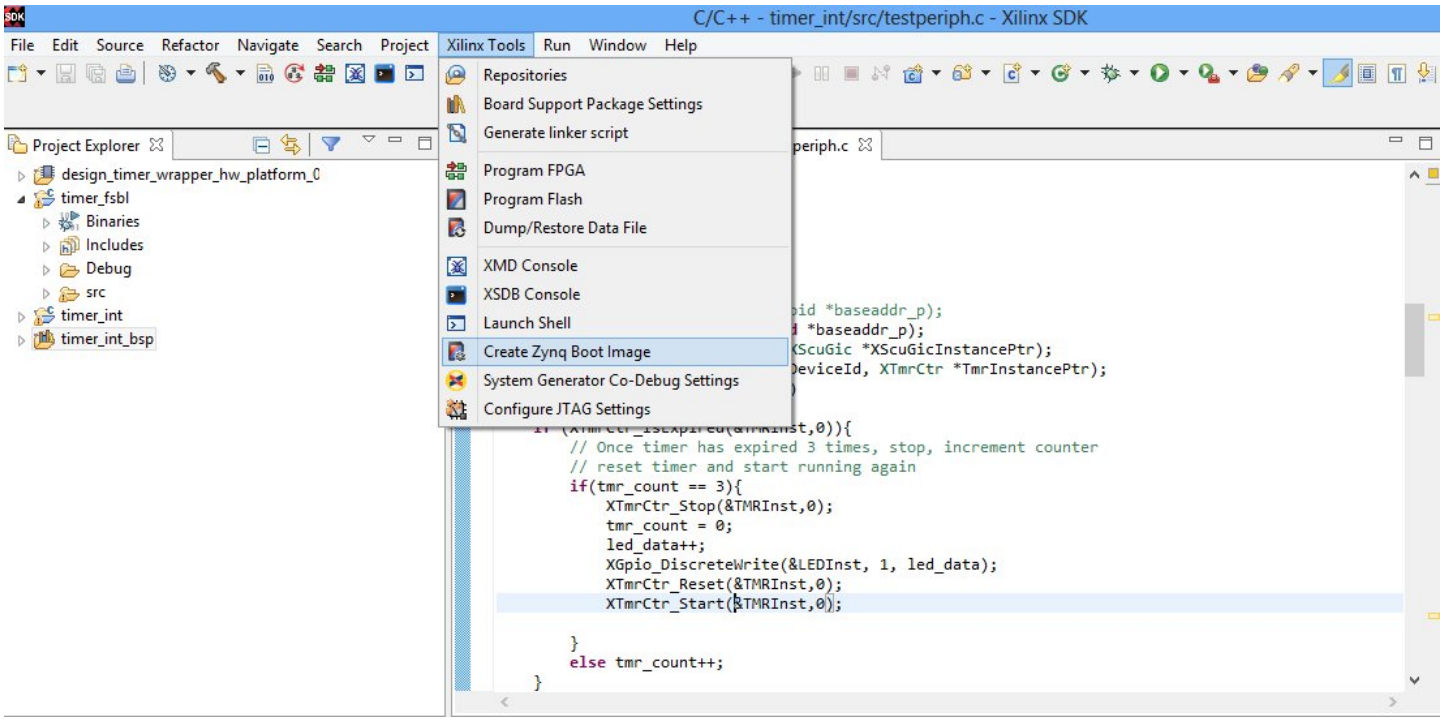
L'explorateur de projet s'agrémente d'un dossier timer_fsbl_bsp et timer_fsbl



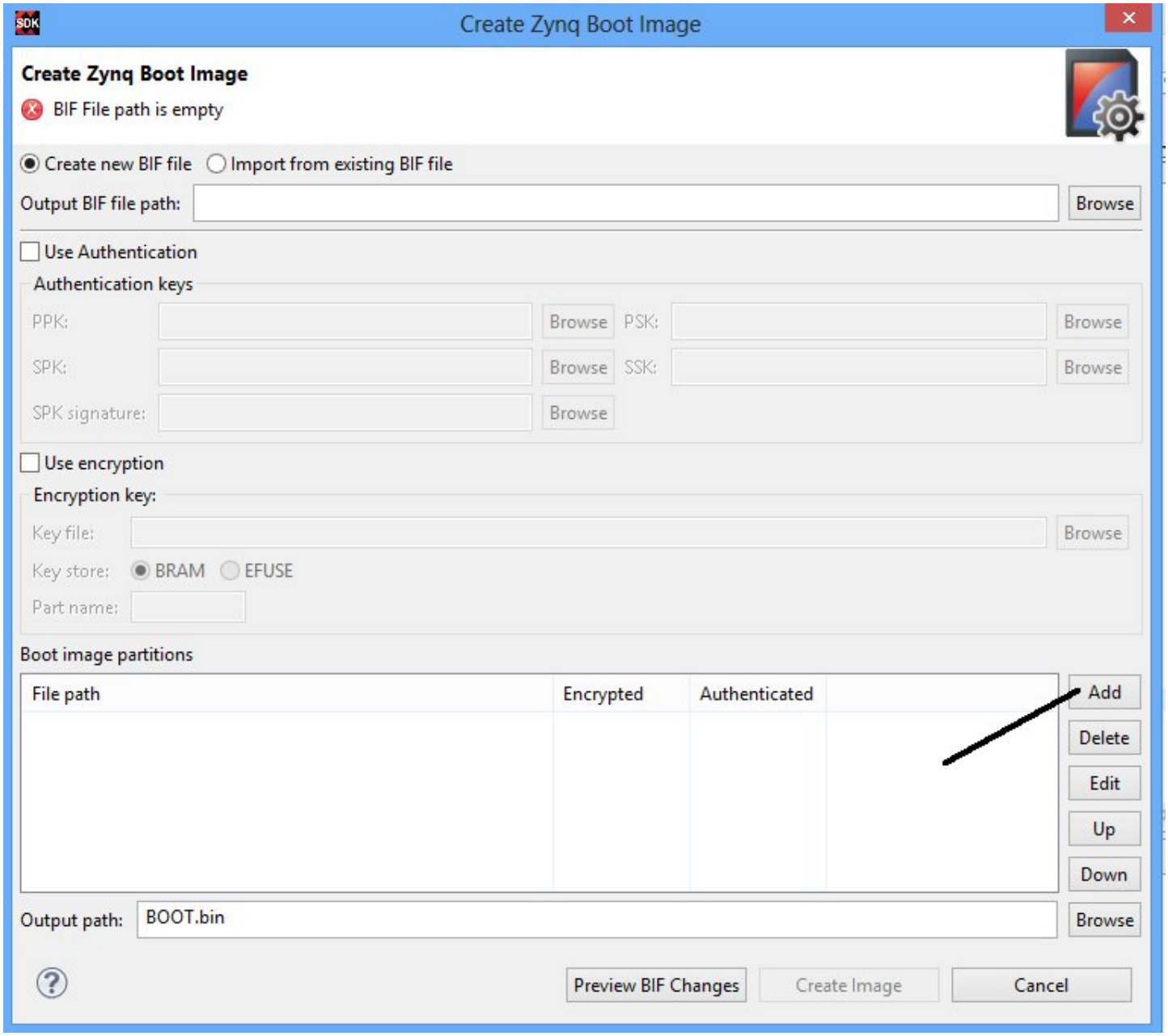
Créer un répertoire qui nous permettra d'accueillir le fichier BOOT.bin



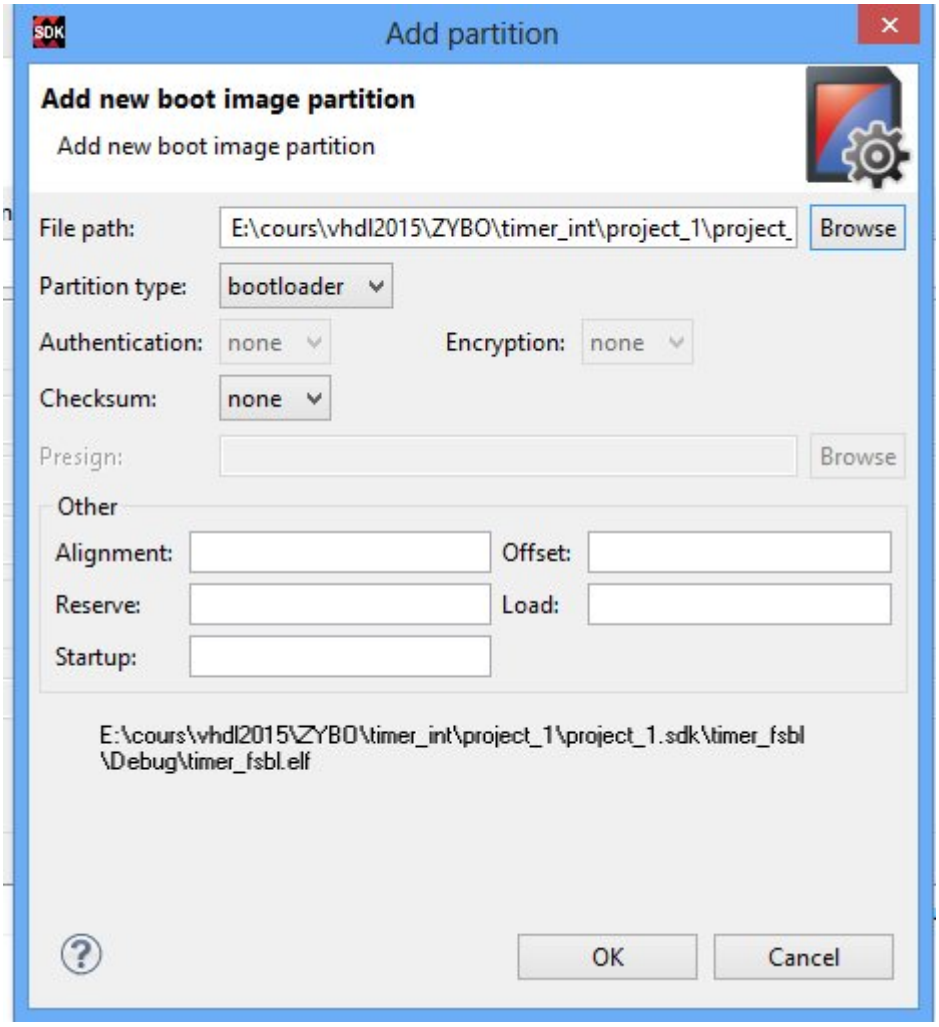
On va créer le fichier BOOT.bin pour la SDCARD



Le fichier BOOT.bin doit intégrer d'abord le fichier bootloader FSBL , cliquez sur ADD



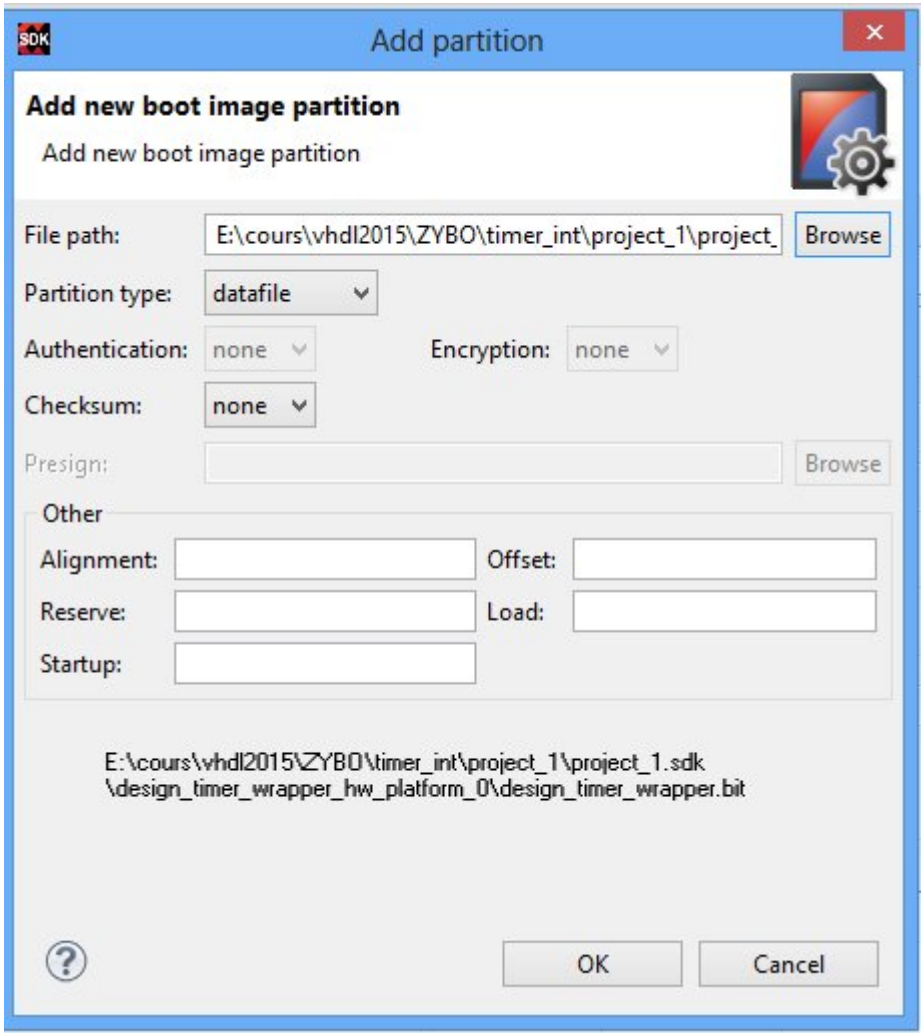
Donnez le chemin du fichier bootloader timer_fsbl.elf. Choisir comme type de partition bootloader



Cliquez à nouveau sur ADD



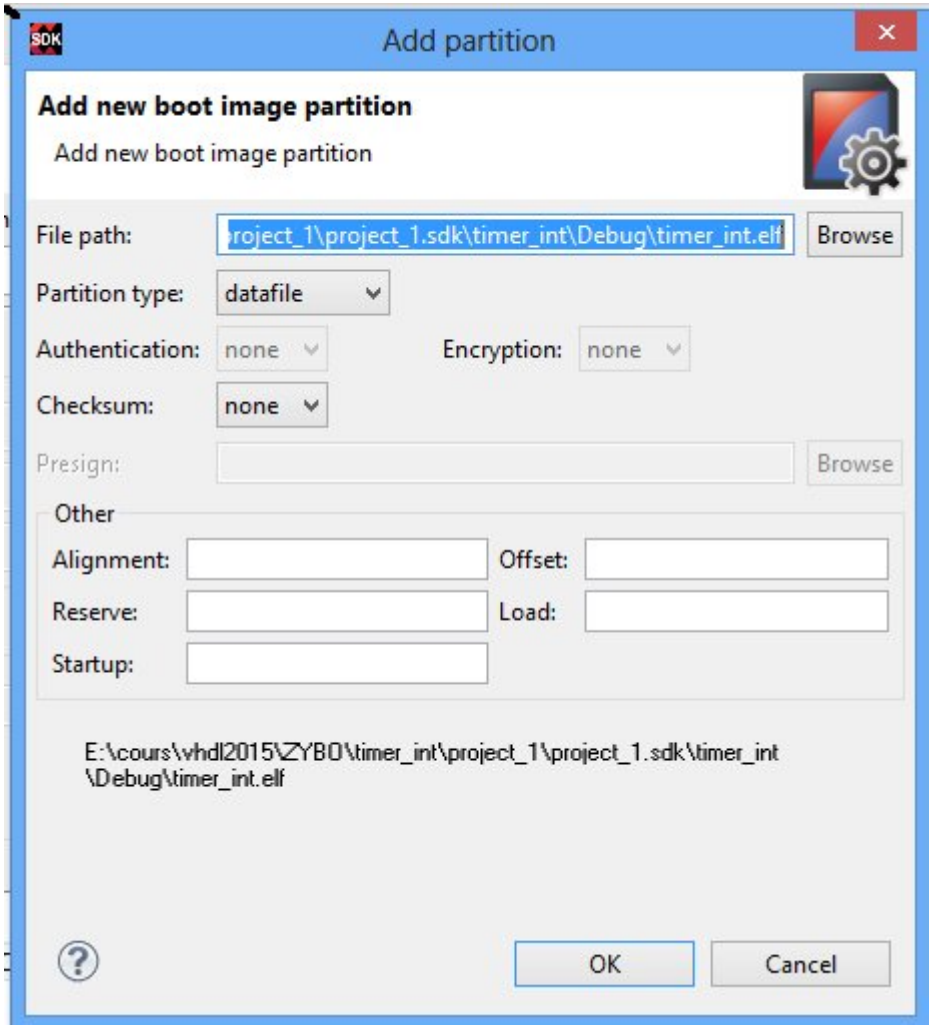
Le fichier BOOT.bin intègre sur sa seconde partie le fichier bitstream qui permettra de programmer la partie logique programmable PL du FPGA. Choisir comme type de partition datafile. Cette étape n'est pas forcément obligatoire dans le cas où vous n'avez pas de fichier Bitstream et que vous n'utilisez que la partie PS du FPGA



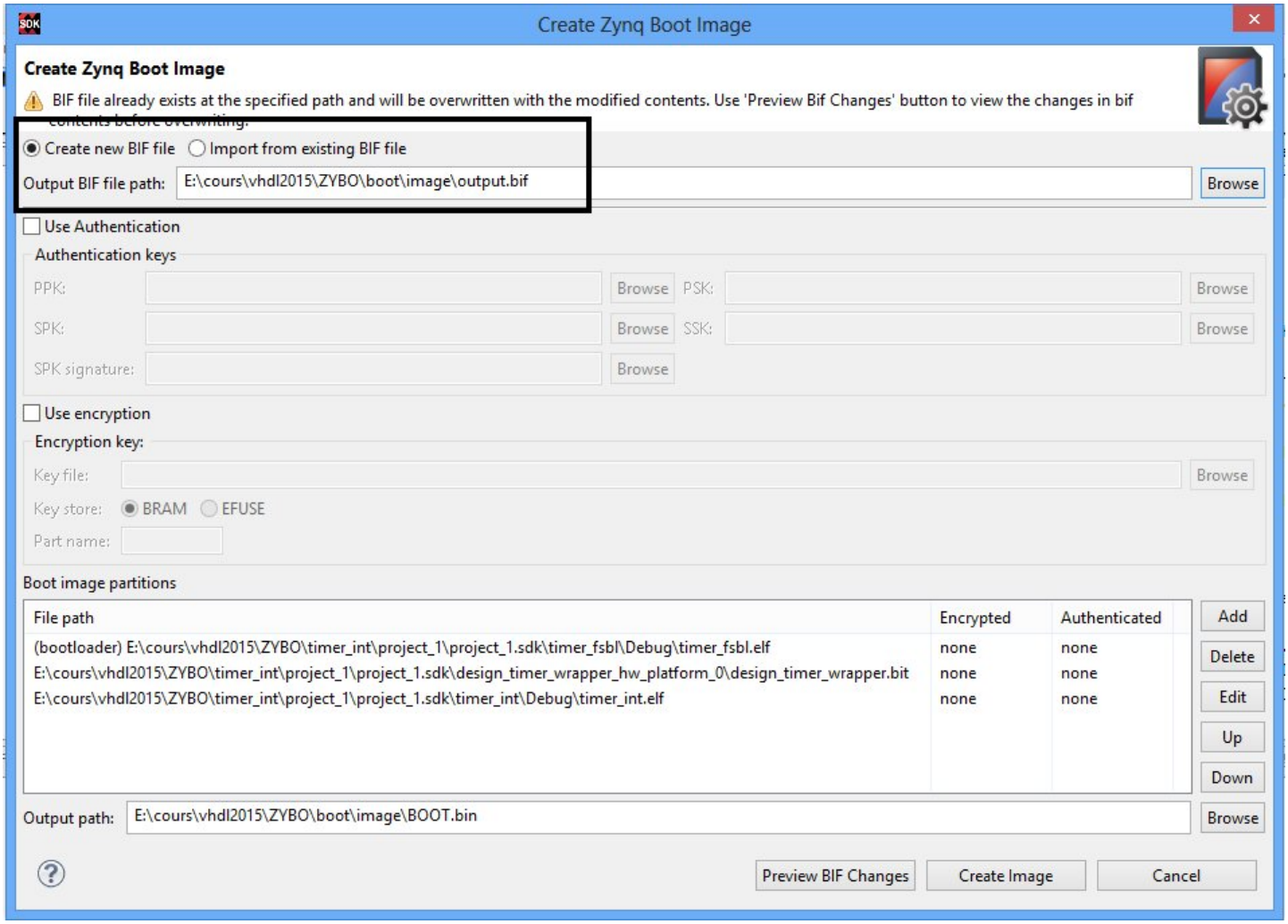
Cliquez sur add



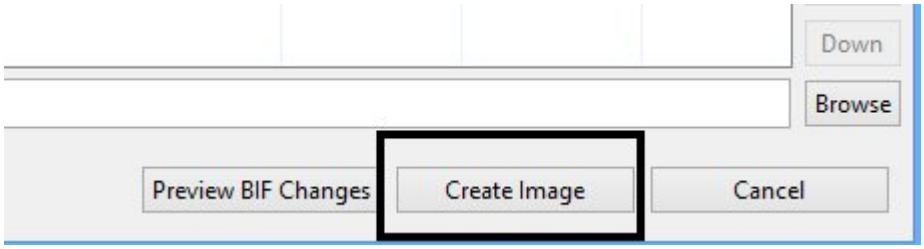
Le fichier BOOT.bin contient dans sa dernière partie notre applicatif
Donnez le chemin de l'applicatif timer_int.elf


Add partition
Add new boot image partition
Add new boot image partition
File path:
project_1\project_1.sdk\timer_int\Debug\timer_int.elf
Browse
Partition type:
datafile
Authentication:
none
Encryption:
none
Checksum:
none
Presign:
Browse
Other
Alignment:
Offset:
Reserve:
Load:
Startup:
E:\cours\vhdl2015\ZYBO\timer_int\project_1\project_1.sdk\timer_int
\Debug\timer_int.elf
OK
Cancel

Créez un fichier output.bif qui contient les données essentielles pour l'élaboration du fichier BOOT.bin



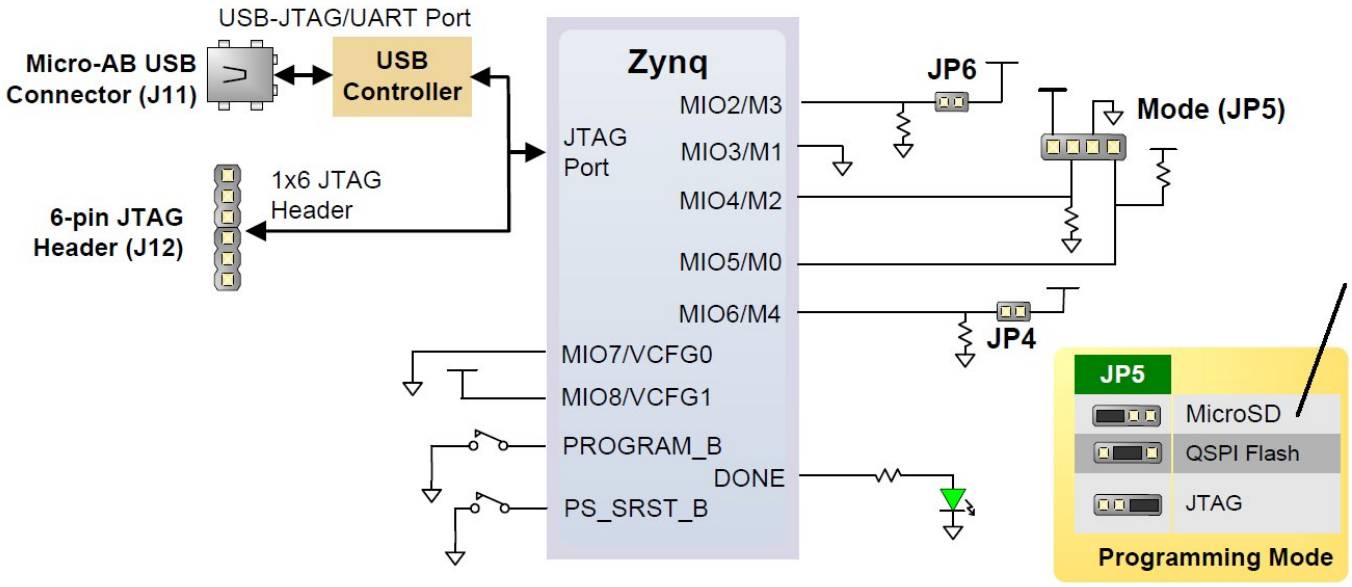
Cliquez sur create image pour générer le fichier BOOT.bin



Transférez le fichier BOOT.bin sur la racine de la SDCARD. La SDCARD doit être formatée en FAT 32



Placez un cavalier sur les deux broches identifiées SD de JP5



Placez votre carte SD dans le lecteur et alimentez la carte ZYBO. Cette carte est devenue totalement autonome