Analyse détaillée des schémas pour préparer le BSP dans le bootloader (config RAM/ROM, PIO, chipset périphériques) Modifier les sources du bootloader, le compiler et le flasher sur la Board

--------------------------------------------------------------------------------------------------------------------------------------

Dans le BSP du bootloader on configure :

- timings de RAM / ROM,

- taille RAM/ROM

- différentes PIO disponibles (numéro de port et direction)

- les chipsets extérieurs présents (Chip ethernet, contrôleur USB extérieur, Real Time Clock, etc...) - différents bus utilisés par le CPU (SATA, IDE, USB, I2C, etc...)

On initialise tout ce qui est indispensable au démarrage : RAM, etc

Et tout ce qui est nécessaire au chargement du kernel : HDD, NAND, … Sans oublier les PIO minimale (alimentation, LED, ventilateur, etc...)

On définit la cross-compilation. On explique que l'intérêt principal est la facilité de développement offerte par la puissance du PC comparativement à la cible

Le mot Anglais bootloader pourrait se traduire par "chargement du code de démarrage". Pour faire une analogie informatique, c'est un peu comme le BIOS de votre ordinateur. C'est un mini programme basique utile pour faire des actions simples.  
  
Le **bootloader** est un petit programme qui "réveille" la puce STM de votre radio. C'est un bout de code qui permet de démarrer votre radio.  
Ce bootloader de démarrage permet la communication avec la puce STM et permet également de faire des actions de maintenances spéciales, telles que l'accès USB, l'accès à la carte SD, mettre à jour OpenTX depuis la carte SD et sauvegarder/restaurer l'EEPROM sans la nécessité d'utiliser un ordinateur.

Cross Compilateur :

La plupart du temps, le processeur utilisé sur votre cible n'est pas d'architecture x86. Vous ne pouvez donc pas utiliser le compilateur de votre PC car il générerait du binaire non exécutable sur votre cible. Cependant, un PC étant généralement bien plus puissant que les cibles, il est préférable de compiler sur ce dernier. Nous allons donc utiliser un compilateur qui s'exécutera sur le PC, mais qui génèrera des binaires qui seront eux exécutables uniquement sur la cible. C'est cette étape qu'on appelle la cross-compilation.—

Most of the time, the processor used on your target is not of x86 architecture. You cannot therefore use your PC's builder because it generates the binary that cannot be executed on your target. However, as a PC is generally much more powerful than the targets, it is preferable to build on the latter. We will therefore use a compiler which will run on the PC, but which will generate binaries which will be executable only on the target. This step is called cross compilation.

The B.S.P or Board Support Package is a generic term that can be summed up by "providing everything necessary for software to run on particular hardware". Typically, the B.S.P starts with primordial configurations, like those of RAM and ROM. It continues with the configuration of the various PIOs. For example, to control all the food necessary for the proper functioning of the other components of the electronic card, but also the various communication inputs and outputs on the various BUSes. Finally, more complex initializations will take place by registering registers of various peripheral components to add advanced functionalities (Ethernet, SATA, cryptography, etc.) These operations must be done partly in the bootloader, and partly in the Kernel. We can also say that the BSP consists of adding support for new electronic cards in the bootloader and the Linux kernel