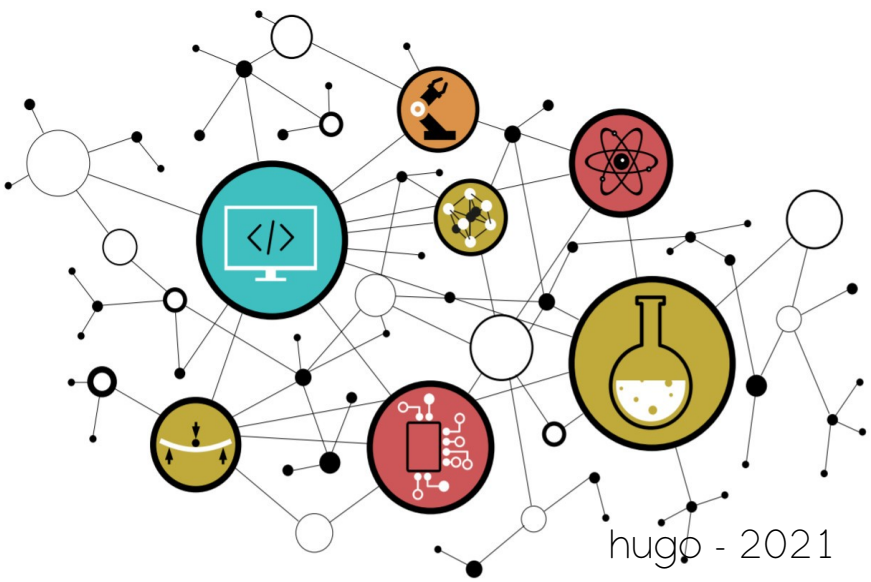
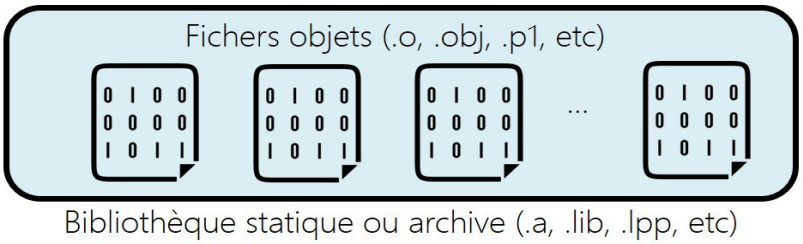
TUTORIEL

CRÉATION DE BIBLIOTHÈQUE STATIQUE SOUS IDE MPLABX



hugo - 2021

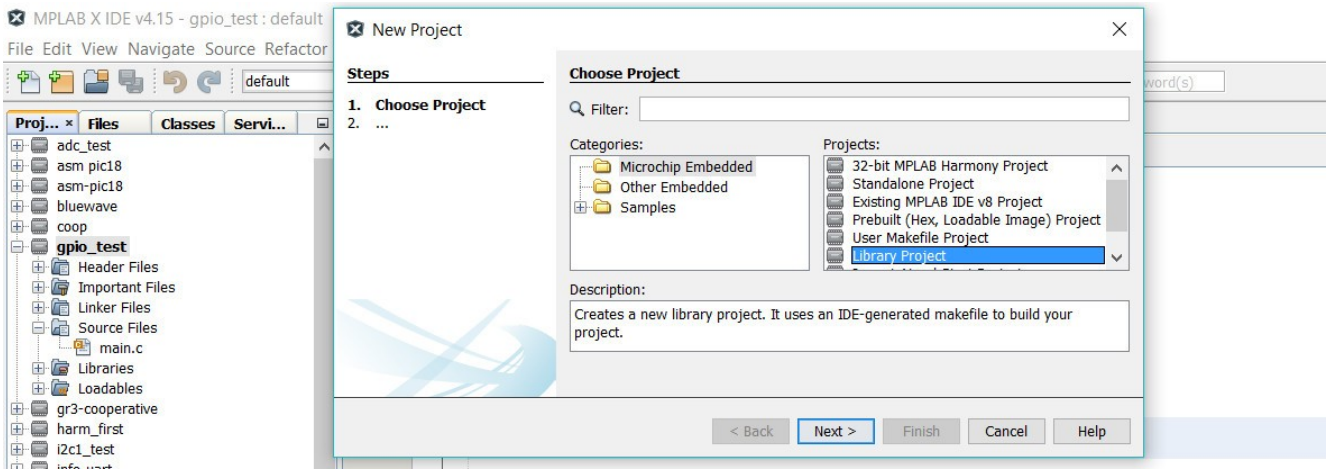
CRÉATION D'UNE BIBLIOTHÈQUE STATIQUE SOUS MPLABX



Une bibliothèque statique est une archive de fichiers objets pré-compilés. En informatique, une archive (non compressée) est un simple ensemble d'éléments joints bout à bout. Il s'agit d'ailleurs de l'outil *ar* (archiver) sur système GNU\Linux. Une bibliothèque statique ne comporte pas de fonction *main*, elle est un agglomérat de fonctions présentes dans des fichiers au format binaire. La fonction *main* est quant à elle le point d'entrée d'une application, utilisant potentiellement des fonctions présentent dans une bibliothèque statique voire dynamique sur système disposant d'un système d'exploitation évolué.
Attention, les outils de développement Microchip sont sujets à un BUG connu et non résolu. Il vous faudra donc générer 2 bibliothèques statiques, une avec tous les fichiers C et l'autre avec tous les fichiers ASM. La procédure qui suit sera donc à réaliser 2 fois (2 x 5mn) et vous obtiendrez en résultat 2 bibliothèques.

Sélection du générateur de bibliothèque statique

- *Ouvrir MPLABX → File → New Project...*
- *Microchip Embedded → Library Project → Next*



Sélection du processeur cible

- *Family → Advanced 8-bits MCUs (PIC18)*
- *Device → PIC18F27K40 → Next*

Sélection de la sonde de programmation ou du starter kit

- *Hardware Tool > Simulator > Apply > OK*

Sélection de la chaîne de compilation

- *XC8 (v1.45) → Next*

Sélection de l'emplacement du projet

- Toujours donner à vos projet un nom en lien avec les développements en cours. Dans le cas présent, une bibliothèque ou library. Par exemple, bsp_lib, adclib, lib_uart et donc éviter des nommages ne permettant pas de connaître le contenu du projet, par exemple lib, tp1, exo3, etc. Le projet MPLABX pour la génération de la bibliothèque statique intégrant tous les binaires du BSP (Board Support Package) développés durant les TP sera placé dans le répertoire *disco/bsp/lib/pjct*.
- *Project Name → <choisir_un_nom_court_sans_accent_sans_espace_ayant_un_sens>* *par exemple bsplib_<your_name>*
- *Project Location → <your_parth>/mcu/tp/disco/bsp/lib/pjct*
- *[x] Use project location as the project folder → Finish*

Ajouter les fichiers sources de la bibliothèque

- *Clic droit sur le nom du projet → Set as Main Project*
- *Clic droit sur Source Files → Add Existing Items... → <add_all_needed_sources> → Select* . Pour rappel, une bibliothèque de fonctions n'est pas une application. La fonction *main* et le code de supervision des applications ne doivent pas être ajoutés. Dans notre cas, seules les fonctions du BSP doivent être insérées au projet.
- Compiler le projet. *Clic droit sur le nom du projet → Build*

Configurer les outils de compilation

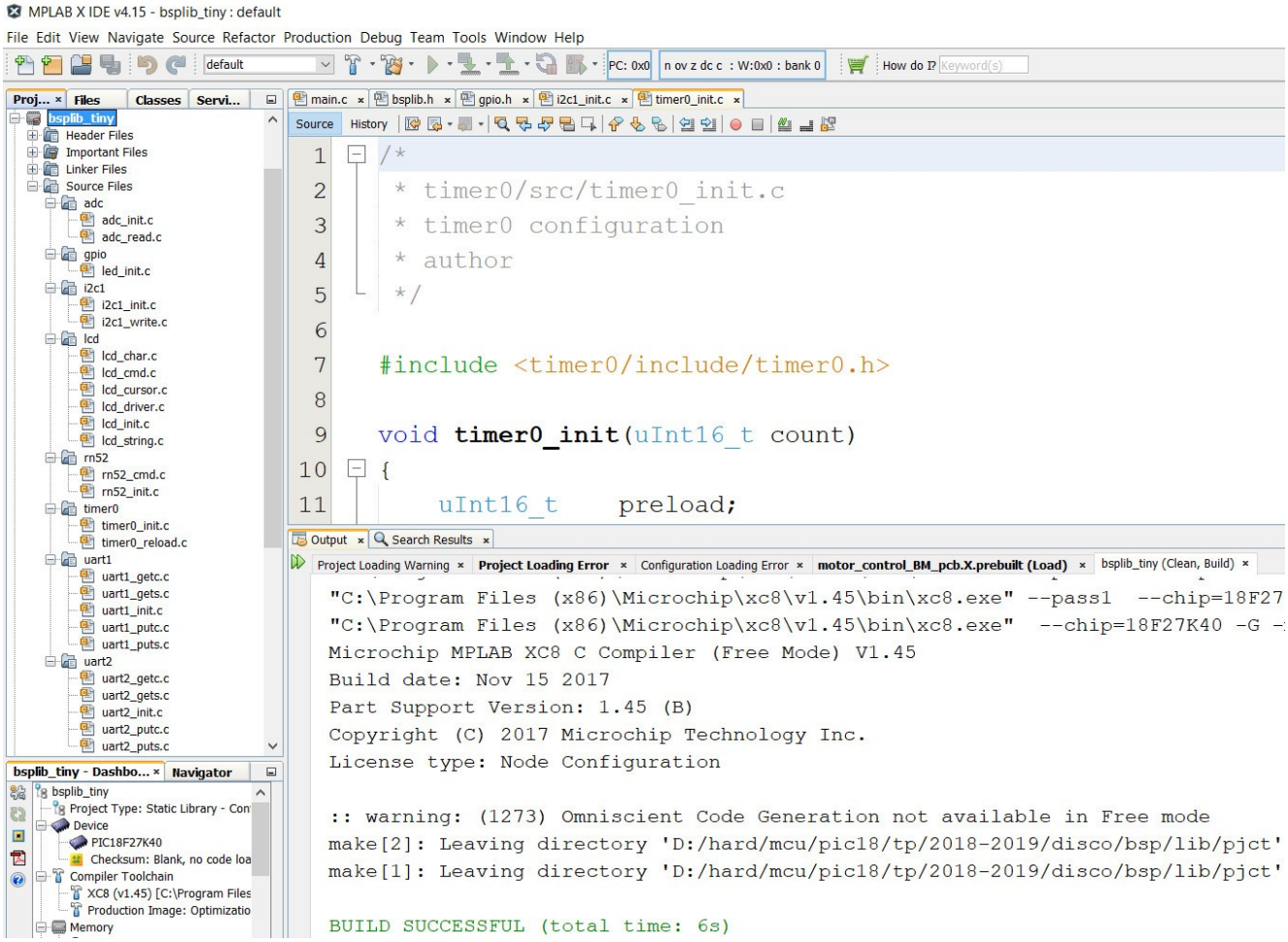
- *Clic droit sur le nom du projet → Properties*
- *Cliquer sur XC8 compiler*
- *Option Categories → Preprocessing and message*
- *Include directories → ... → Browse... → <your_parth>/bsp/ → Apply*

Génération de la bibliothèque statique

- *Clic droit sur Source Files → New Logical Folder* afin de générer une arborescence logique permettant un usage efficace de l'environnement graphique proposé par l'IDE. Bien s'assurer que tous les sources C nécessaires soient bien présents (cf. capture d'écran ci-dessous). Ne pas inclure les fichiers assembleur (fonctions delays et switch_init)

- *<u>BUG :</u> Pour rappel, il vous faudra générer 2 bibliothèques statiques distinctes et donc créer 2 projets séparés. Un n'incluant que les fichiers sources C, puis l'autre que les fichiers sources assembleurs ASM. Il ne s'agit pas d'une procédure naturelle pour un chaîne de compilation et un générateur de bibliothèque statique. Ce bug est connu et est documenté. L'exemple ci-dessous présente la génération de la bibliothèque statique du BSP intégrant les fonctions pilotes (drivers) développées en langage C. Lire et suivre les directives du document disco/bsp/lib/bsplib_asm/README.txt afin de générer la bibliothèque ASM en lignes de commandes*

- *Clic droit sur le nom du projet → Build*

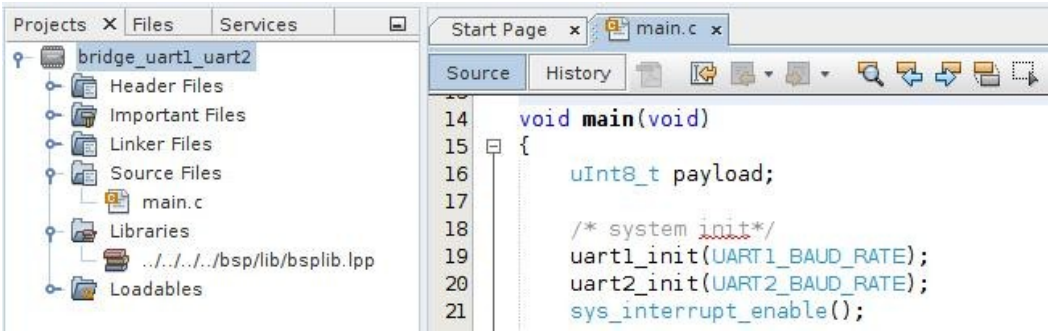


- Après génération, la bibliothèque sera rangée par défaut dans le répertoire *pjct/dist/default/production/<lib_name>.lpp*

- A copier voire renommer dans *bsp/lib*. Par exemple, différencier les deux bibliothèques de la sorte : *bsplib_<your_name>.lpp* et *bsplib_asm_<your_name>.lpp*

Utilisation de la bibliothèque statique

- Il ne reste maintenant plus qu'à ajouter votre bibliothèque à vos applications. Les applications de test *disco/apps/adc_uart1* et *disco/apps/bridge_uart1_uart2* peuvent par exemple être testées avec ces bibliothèques (bibliothèque fournie par défaut dans *disco/bsp/lib/bsplib.lpp*)
  - *Si ce n'est pas déjà fait, bien penser à retirer les fichiers sources des fonctions pilotes afin d'éviter des définitions multiples à l'édition des liens*
  - *Clic droit sur le répertoire logique du projet Libraries (cf. ci-dessous)*
  - *Add Library/Object File...*
  - *Ajouter votre bibliothèque statique ou celle par défaut pour test*
  - *Compiler et tester l'application de test*



- La bibliothèque complète générée en fin d'enseignement sera le fruit de plusieurs mois de travail. Elle pourra maintenant être utilisée dans une infinité d'applications potentielles … il ne reste plus qu'à imaginer des projets … et les réaliser !