



## Fichiers inclus sous Arduino

Une bonne pratique de programmation, quand les programmes deviennent longs et que des parties de programmes sont utilisables dans différents programmes, est de mettre les définitions, fonctions, modules dans des fichiers séparés et les appeler avec l'ordre #include.

Le mécanisme est bien connu pour les bibliothèques Arduino, qui sont dans une zone mémoire réservée et connue du compilateur.

Prenons l'exemple de l'ensemble des définitions pour les exercices du cours Microcontrôleur. On crée un fichier LcDef.h qui les contient, et dans le programme, on note #include "LcDef.h" pour que le préprocesseur ajoute ce fichier avant d'envoyer le programme au compilateur.

Pour le set-up, on préfère définir une fonction qui est appelée dans le setup. On ajoute dans le setup les initialisations éventuellement nécessaires, comme Serial.begin(9600);

De cette façon, on a dans le fichier LcDef.h tout ce qui concerne l'aspect matériel de l'application.

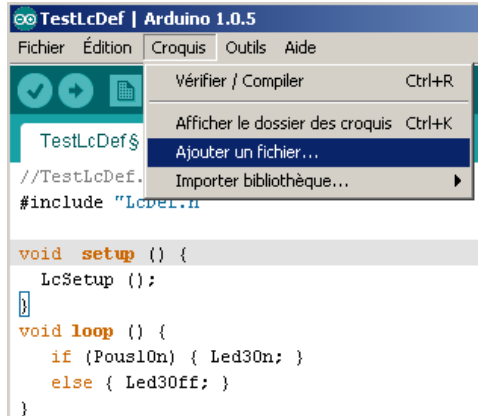
On peut parler de bibliothèques propres (elles sont entre "" alors que les bibliothèques communes sont entre <>). La contrainte pour les bibliothèques propres est que les fichiers insérés doivent se trouver dans le même dossier (sketch, croquis) que le programme. Le sketch du programme contient le fichier .ino et les fichiers .h associés.

Dans l'index "Sketch" on peut aller chercher des fichiers à ajouter dans d'autres dossiers, mais on ne peut pas les enlever, il faut intervenir avec le gestionnaire de fichiers.

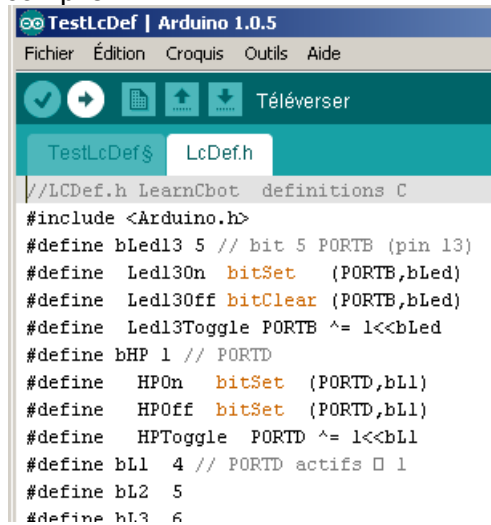
Exemple; vous voulez clignoter la Led3. Il faut se référer avant la boucle de programme aux deux éléments prédéfinis:

```
#include "LcDef.h"
et LcSetup ();
```

Ensuite, il faut ajouter le fichier LcDef.h, préparé quelque part sur le disque, dans le dossier du programme.



On peut ensuite observer les fichiers importés, et il n'est pas nécessaire de sélectionner la première fenêtre pour recompiler.



On peut naturellement faire des fichiers importés avec des fonctions. La contrainte est que les variables utilisées doivent être définies dans un module précédemment appelé.

La coupure en morceaux d'un programme existant n'est pas évidente. Il faut parfois passer par un éditeur comme NotePad++.

Sous l'onglet "sketch" le menu "add file" permet d'ajouter un fichier, mais apparemment on ne peut pas ajouter un fichier vide que l'on nomme à ce moment. Dans Energia, l'option "New Tab" crée un fichier vide.

Pour des modifications successives de programmes, c'est facile, les fichiers inclus sont automatiquement transportés. Ne pas oublier de sauver à chaque étape.

A noter que si le fichier inclus comporte des noms de registres ou des fonctions Arduino, il faut ajouter la ligne #include <Arduino.h>.

### Programmation sous C

Le changement est mineur si vous utilisez un compilateur C à la place d'Arduino.

```
#include "LcDef.h"
int main () {
    LcSetup ();
    while (1) { // remplace void loop () {
        . . .
    }
}
```