

École Polytechnique de Montréal

Département de génie informatique et génie logiciel

Cours INF1995

Projet initial en génie informatique et travail en équipe

Travail pratique 8

Makefile et production de bibliothèques statiques

Par l'équipe

No 1620

Noms:

David Mainville
Yannick Ouellet
Mario Saliby
Youssef Zemmahi

Date:

28 octobre 2013

Partie 1 : Description de la librairie

Une librairie est un fichier compilé regroupant un ensemble de méthodes prêtes à être utilisées par un programme. Dans le cas présent, le programme sera en mesure d'exécuter certaines fonctions propres au robot.

À partir des codes écrits au cours des sept travaux pratiques précédents, nous avons créé trois méthodes qui, selon nous, semblaient les plus pertinentes et les plus significatives. Elles sont toutes étiquetées comme étant statiques puisqu'un seul exécutable est chargé dans la puce du microcontrôleur. Si la librairie était utilisée par plusieurs executables à la fois, elle aurait été décrite comme dynamique.

La première méthode constituant l'interface de la classe `LibRobot` dans le fichier source `libRobot.h` est `tournerRoueDroite()`. Comme son nom l'indique, cette méthode permet de faire tourner la roue droite du robot en générant un signal PWM. Les trois paramètres de la fonction caractérisent ce signal : la variable `dureeAllumee` de type `uint16_t` représente la durée, en millisecondes, durant laquelle l'onde est dans sa période active, la variable `periode`, également de type `uint16_t`, représente la période totale de l'onde en millisecondes et `duree`, de type `int`, la durée du roulement total en millisecondes. Aucune valeur n'est retournée à la fin de l'exécution.

La deuxième méthode retenue est `attenuerDEL()` qui, aussi, ne retourne rien une fois exécutée. Cette fonction permet l'atténuation de l'intensité de la DEL, encore une fois grâce à un signal PWM. Le seul paramètre, `rouge`, est une variable booléenne contrôlant la couleur initiale de la DEL avant son atténuation. Si la variable vaut 1, la DEL est rouge. Autrement, elle est verte.

La dernière fonction de l'interface de `LibRobot` est `BoutonEstEnfoncé()` qui sert à vérifier l'état du bouton-poussoir en tenant compte de l'anti-rebond. Aucun paramètre n'est pris en charge. La méthode retourne un `uint8_t`. La valeur 0 est retournée lorsque le bouton est relâché et la valeur 1, grâce à l'opérateur `!!`, lorsqu'il est appuyé.

Évidemment, `libRobot.h` contient un constructeur par défaut ainsi qu'un destructeur. Le constructeur, le destructeur et les trois méthodes sont tous implémentés dans le fichier `libRobot.cpp`. Celui-ci fait donc office d'unité de compilation lors de la création de l'archive.

Partie 2 : Décrire les modifications apportées au Makefile de départ

Makefile de la librairie

Pour ce qui est du Makefile de la librairie, la librairie a été nommée `libRobot` via la variable `PROJECTNAME` et l'unité de compilation `libRobot.cpp` a été affectée à `PRJSRC`. De plus, dans la section `automatic target names`, l'extension par défaut `.out` de la variable `TRG` a été changée pour `.a` afin d'obtenir une librairie `libRobot.a` comme produit final.

Deux nouvelles variables ont été créées dans la section `executables`. La première, `CCAR`, est une variable pour le nom de la commande `avr-ar` tandis que la deuxième, `CCAROP`, désigne les options de cette commande, `-crs`. Ces deux variables sont utilisées dans la section `Make targets` à l'appel de la règle `TRG` où elles substituent `$(CC)` `$(LDFLAGS)` dans le Makefile original. En temps normal, le Makefile appelle la commande `avr-gcc` ainsi que ses options. Cependant, lorsque l'on désire produire ou mettre à jour une librairie statique, il est indispensable d'appeler la commande `avr-ar` avec les options `c`, `r` et `s` qui, respectivement, créent l'archive, y insèrent les fichiers objets nécessaires puis indexent ceux-ci.

Il est à noter que dans la section `additional includes`, aucun chemin n'est attribué à `INC` puisqu'aucun des fichiers sources dont le Makefile a besoin ne se trouve dans un autre répertoire. Aucun chemin n'a également été donné à la variable `LIBS` dans la section `libraries to link in` puisque celle créée par le Makefile n'en requiert aucune autre comme référence.

Makefile de l'application utilisant la librairie

En ce qui concerne le Makefile de l'application utilisant la librairie construite, l'exécutable a été nommé `tp8` dans la variable `PROJECTNAME` et l'unité de compilation `testLib.cpp`, testant de manière rudimentaire les méthodes de la librairie, sert d'unique fichier source dans la variable `PRJSRC`.

Contrairement au Makefile précédent, il a fallu modifier les sections `additional includes` et `libraries to link in`. Dans la première de ces sections, le chemin `-I ./Lib/` est affecté à `INC`. En effet, dans ce cas-ci, les fichiers « include » ne se trouvent pas dans le même répertoire que le Makefile mais plutôt dans le répertoire `Lib/`. Il faut donc aller les chercher grâce à l'option `-I`. Une variable supplémentaire, `LINC`, a

aussi été ajoutée dans cette même section. `LINC` permet de prendre en référence la librairie créée afin de permettre son utilisation lors de la formation de l'exécutable. Encore une fois, puisque cette librairie ne se trouve pas dans le même répertoire que le Makefile, il faut préciser son chemin, soit `-L ./Lib/`. L'option `-L` est cette fois-ci employée au lieu de l'option `-I` puisqu'il s'agit du chemin d'une librairie et non celui d'un fichier « include ». `LINC` est ensuite introduite dans la variable `LDFLAGS`, contenant toutes les options passant par l'éditeur de liens, dans la section `linker`.

Pour pouvoir avoir recours à la librairie lors de la formation de l'application, il ne faut pas oublier d'également donner son nom. Par conséquent, `-lRobot` est affecté à la variable `LIBS` dans la section `librairies to link in`. Il est ici important de faire la distinction entre l'option `-l` et l'option `-L`. Alors que l'option `-L` annonce le chemin du répertoire dans lequel se trouve une librairie, l'option `-l` sert tout simplement à nommer celle-ci.