

Polytechnique 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 librairie statique

2
2
1.5
1
6.5

Par l'équipe

No 0889

Noms:

Guillaume Renaud
Yann-Joel D. Tessier
Loic Bellemare-Alford
Julien Bergeron

Date:
13 Mars 2017

Partie 1 : Description de la librairie

Décrire la librairie construite et formée (définitions, fonctions ou classes, utilité, etc...) pour que cette partie du travail soient bien documentées pour la suite du projet pour le bénéfice de tous les membres de l'équipe.

Partie 2 : Décrire les modifications apportées au Makefile de départ

Décrire les quelques modifications apportées au Makefile de la librairie pour démontrer votre compréhension de la formation des fichiers. Faire de même pour les modifications apportées au Makefile du code (bidon) de test qui utilise cette librairie.

Le rapport total ne doit pas dépasser 7 pages incluant la page couverture.

Barème: vous serez jugé sur:

- *La qualité et le choix de vos portions de code choisies (5 points sur 20)*
- *La qualité de vos modifications aux Makefiles (5 points sur 20)*
- *Le rapport (7 points sur 20)*
 - *Explications cohérentes par rapport au code retenu pour former la librairie (2 points)*
 - *Explications cohérentes par rapport aux Makefiles modifiés (2 points)*
 - *Explications claires avec un bon niveau de détails (2 points)*
 - *Bon français (1 point)*
- *Bonne soumission de l'ensemble du code (compilation sans erreurs, ...) et du rapport selon le format demandé (3 points sur 20)*

Partie 1 – Description de notre librairie

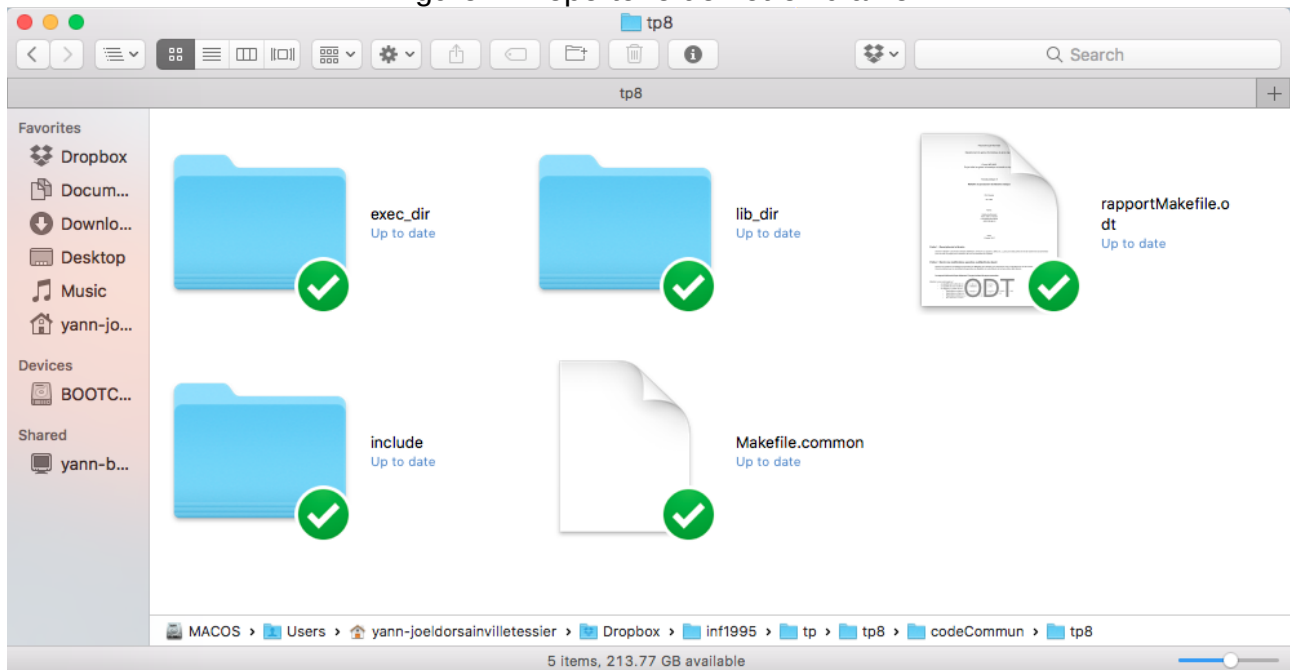
Notre librairie est constituée des fichiers sources contenant les prototypes des principales fonctionnalités utilisées par notre robot. L'utilité est d'être en mesure d'utiliser ce code dans n'importe quel futur travail pratique qui en aurait besoin. De cette façon, nous n'avons pas à réécrire le même code plusieurs fois. Pour avoir accès à ces fonctionnalités, il faut les inclure dans le code que l'on souhaite implémenter sur la carte. Ces fonctionnalités sont des classes ayant différentes méthodes parmi lesquelles nous pouvons choisir l'action à effectuer. Dans notre librairie, nous avons les fonctionnalités suivantes :

- Manipulation des deux moteurs du robot. Les deux moteurs fonctionnent indépendamment l'un de l'autre afin de pouvoir envoyer des fréquences de PWM différentes pour avoir des vitesses de rotation différentes et faire ainsi tourner le robot. Nous pouvons le faire avancer, reculer et arrêter.
- Manipulation du convertisseur analogique numérique. Nous pouvons initialiser le convertisseur, le rendre inactif et faire des lectures qui seront retournées sur 16 bits.
- Ajustement du compteur 1 de 16 bits. Nous pouvons choisir le mode du compteur ainsi que la vitesse de l'horloge.
- Manipulation de la DEL libre sur la carte mère. Nous pouvons allumer la DEL et ajuster la couleur pour qu'elle soit verte ou rouge et l'éteindre.

UART? Mémoire Int externe

Nous avons mis les fichiers à inclure (.h) dans un dossier nommé « include ».

Figure 1. Répertoire de notre librairie



Partie 2 – Description des Makefiles

Pour ce qui est des Makefiles, nous avons fait de la ^{re}factorisation pour les Makefiles locaux, ceux que l'on retrouve dans les répertoires `exec_dir` et `lib_dir`, puisqu'ils ne font que définir des variables, la cible (la librairie) et la commande de compilation qui sont toutes spécifiques à l'exécutable ou à la librairie. Ensuite, ils font appel au Makefile commun qui compilera, créera les liens et fera le « install » dans le cas de l'exécutable. ✓

✓ Dans le Makefile de la librairie, nous avons créé une variable `SRC` qui contient tous les fichiers « .cpp » des fonctionnalités décrites plus haut. La cible devient le nom de la librairie avec l'extension « .a ». Nous avons ajouté le ~~compilateur~~ `avr-ar` pour la création de l'archive à partir des fichiers. La commande de compilation spécifique à la librairie est écrite dans ce Makefile et est passée au Makefile commun afin que celui-ci puisse exécuter des commandes différentes. ✓ *archiveur*

Le Makefile de l'exécutable est très similaire au Makefile de la librairie puisque les seuls éléments qui changent sont les variables, la cible (l'exécutable) et la commande de compilation spécifique à ce Makefile. La variable `SRC` ne contient que le code à implémenter sur la carte. La cible devient le nom de l'exécutable avec l'extension « .out ». La commande de compilation diffère légèrement puisqu'elle doit lier les librairies (dans notre cas une seule) avec l'option « `-lm` ». ? Nous avons ajouté une variable `INCL` qui contient une référence au dossier « include » des fichiers « .h » ainsi qu'une variable `LIBS` pour avoir accès à la librairie. ✓

- statique

plus de détails

Le Makefile commun ne fait que compiler la cible avec les bonnes dépendances et la bonne commande et exécute le « install » et le « clean » si nécessaire.