Polytechnique Montréal

Département de génie informatique et génie logiciel

Cours INF1900:
Projet initial de système embarqué

Travail pratique 7

Makefile et production de librairie statique

Par l'équipe

No 3236

Noms:

LACHIEZE-REY Hugo
TASSAUX Lucas
JABIR Yassine
HOMSI Yoseph

Date :
20 Octobre 2019

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 par dépasser 7 pages incluant la page couverture.

Barème: vous serez jugé sur:

- *La qualités et le choix de vos portions de code choisies (5 points sur 20)*
- *La qualités 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)*

1/ Choix de codes retenus pour notre librairie

Nous avons choisi de garder 5 portions de codes pour notre librairie :

- DEL : pour allumer notre DEL selon la couleur demandée et le port utilisé ;

Méthode : *void AllumerDEL(uint8_t couleur, char port);*

- AntiRebond : lorsqu'on presse notre pouton poussoir, il est nécessaire d'avoir un anti-rebond afin de linéariser le signal envoyé ;

Méthode : *bool BoutonPresse();*

- PWM : le signal d'horloge doit être ajuster en fonction de la vitesse désirée pour faire fonctionner les moteurs ;

Méthode :
void ajustementPWM(uint8_t parametre, uint16_t maskTCCRnA, uint16_t maskTCCRnB);

- Mémoire : d'après la documentation Atmel fournie, il est nécessaire d'initialiser certains registres pour avoir accès à la mémoire de notre robot. De plus, nous aurons potentielement besoin d'afficher ce qui se trouve en mémoire.

Méthodes :
void transmissionUART (uint8_t donnee);
void initialisationUART (void);
void affichageMemoire(uint16_t adresse, uint8_t caractereLu, uint8_t dernierCaractereLu);

- Minuterie : contrôle de la minuterie au besoin.

Méthodes :
void partirMinuterie(uint16_t duree, uint16_t masqueTCCRnB, uint16_t masqueTCCRnC);
void arreterMinuterie();

Nous avons aussi ajouté les fichiers *mémoire_24* et *can* à notre librairie statique. En effet, nous en avons besoin de *mémoire_24* pour la lecture et l'écriture de données dans la mémoire externe du robot. La classe *can* permet d'utiliser un capteur de luminosité en transformant le signal analogique en signal numérique.

2/ Modifications Makefiles

i/ Makefile de notre librairie

Nous utilisons maintenant notre propre librairie statique. Nous avons donc du modifier le makefile de départ.

Nommer notre librairie :
(l. 11) PROJECTNAME = lib_statique

Utiliser tous les fichiers cpp du répertoire de notre librairie statique
(l.17) PRJSRC= $(wildcard *.cpp)

Produire une libraire statique
(l. 52) AR = avr-ar

Extension d'un fichier pour une librairie statique afin d'avoir la bonne cible par défaut
(l.84) TRG=$(PROJECTNAME).a

Implémenter notre cible. Notre cible est liée à son flag -crs soit *création de l'archive* puis *insertion des fichiers membresdan l'archive* et enfin *écriture de l'objet dans l'archive* selon la documentation ar fournie.
(l. 123-124) $(TRG) : $(OBJDEPS)
$ (AR) -crs $(TRG) $(OBJDEPS)

ii/ Makefile pour l'éxécutable

Nous voulons maitenant créer une makefile afin de pouvoir utiliser notre librairie statique.

Nommer notre librairie :
(l. 26) PROJECTNAME = executable

Récupérer tous les fichiers cpp dans le répertoire dans notre librairie statique.
(l.32) PRJSRC= $(wildcard *.cpp)

Inclusion de notre librairie statique en spécifiant le chemin
(l. 35) INC= -I ../librairie_statique

Liaison de la librairie en spécfiant le chemin
(l. 38) LIBS=-L ../librairie_statique/-lstatique