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

Par l'équipe

No 1024

Noms:

Sébastien Chagnon
Éric Li
Charles Girard
Félix Brunet

Date:
1er novembre 2016

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.

Enums

Roue
Défini les deux roues du robot. Les valeurs qui leurs sont attribués sont pour aider lorsque la direction de rotation des roue est fixée.

- Roue_Gauche
  - Valeur de 4
- Roue_Droite
  - Valeur de 5

Prescaler
Défini les differentes valeurs possible du prescaler. Les valeurs qui leurs sont attribués sont pour aider lorsque le prescaler est fixé.

- Prescaler_1
  - Valeur de 1
- Prescaler_8
  - Valeur de 2
- Prescaler_64
  - Valeur de 3
- Prescaler_256
  - Valeur de 4
- Prescaler_1024
  - Valeur de 5

Direction
Défini les 2 différents sens de rotation possible de la roue

- Direction_Avancer
- Direction_Reculer

CouleurDell
Défini les differentes couleurs possibles de la dell. Les valeurs qui leurs sont attribués sont pour aider lorsque la couleur est fixée.

- CouleurDell_Eteint
  - Valeur de 0
- CouleurDell_Vert
  - Valeur de 1
- CouleurDell_Rouge
  - Valeur de 2

Fonctions

void debounce()

brief	Effectue un delais de 10 ms qui est utile pour l'anti-rebond

bool isButtonPressed()

brief	Verifie si le bouton est pese avec système d'anti-rebond
return	true si le boutton est pese, false sinon

bool isButtonReleased()

brief	Vérifie si le bouton n'est pas pese
return	true si le boutton n'est pas pese, false sinon

void setLight(CouleurDell couleur)

brief	Change la couleur de la dell à la couleur desiree lorsque la dell est branchée aux ports PC0 et PC1	
params	couleur	Couleur desiree de la dell

void initialiasation(uint8_t portA, uint8_t portB, uint8_t portC, uint8_t portD)

brief	Initialise les entrees et sorties des differents ports de la carte mère et active les interruptions	
params	portA	Valeurs d'entree ou de sortie du port A
	portB	Valeurs d'entree ou de sortie du port B
	portC	Valeurs d'entree ou de sortie du port C
	portD	Valeurs d'entree ou de sortie du port D

void ajusterPWM(uint8_t ratio, Direction direction)

brief	Ajuste le PWM au ratio désire par le compteur 2 donc output PWM pour la roue gauche au PD6 et roue droite au PD7. Output de la direction pour la roue gauche au PD5 et roue droite au PD6.	
params	ratio	Valeur du ratio désiré
	direction	Sens de la rotation des roues désiré

void ajusterPWM(uint8_t ratioRoueGauche, uint8_t ratioRoueDroite, Direction direction)

brief	Ajuste le PWM au ratio désire par le compteur 2 donc output PWM pour la roue gauche au PD6 et roue droite au PD7. Output de la direction pour la roue gauche au PD5 et roue droite au PD6.	
params	ratioRoueGauche	Valeur du ratio désiré pour la roue gauche
	ratioRoueDroite	Valeur du ratio désiré pour la roue droite
	direction	Sens de la rotation des roues désiré

void ajusterPWM(uint8_t ratioRoueGauche, uint8_t ratioRoueDroite, Direction directionRoueGauche, Direction directionRoueGauche)

brief	Ajuste le PWM au ratio désire par le compteur 2 donc output PWM pour la roue gauche au PD6 et roue droite au PD7. Output de la direction pour la roue gauche au PD5 et roue droite au PD6.	
params	ratioRoueGauche	Valeur du ratio désiré pour la roue gauche
	ratioRoueDroite	Valeur du ratio désiré pour la roue droite
	directionRoueGauche	Sens de la rotation de la roue gauche désiré
	directionRoueDroite	Sens de la rotation de la roue droite désiré

void ajusterDirection(Roue roue, Direction direction)

brief	Ajuste le sens de rotation de la roue. Output de la direction pour la roue gauche au PD5 et roue droite au PD6.	
params	roue	Roue à laquelle appliquer le sens de rotation
	direction	Sens de la rotation de la roue désiré

void clairInterruptions()

brief	Retire les interruptions en queue d'execution

void initialiserMinuterie(uint16_t ms)

brief	Initialise le compteur 1 du microcontrôleur affin qu'il produise une interruption (TIMER1_COMPA_vect) à chaque nombre demandé de ms (entre 1 et 8388)	
params	ms	Nombre de milli-secondes de la minuterie

void initialiserCompteur(uint16_t duree)

brief	Ajuste le compteur 1 en mode CTC avec un prescaler de 1 et une durée demandée	
params	duree	Valeur que prendra le registre OCR1A

void initialiserCompteur(uint16_t duree, lib::Prescaler prescaler)

brief	Ajuste le compteur 1 en mode CTC avec un prescaler une durée demandée	
params	duree	Valeur que prendra le registre OCR1A
	prescaler	Prescaler du compteur 1

void definirPrescaler(lib::Prescaler prescaler)

brief	Ajuste le prescaler du compteur 1	
params	prescaler	Prescaler à défénir au compteur1

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.

Makefile de la librairie

- PROJECTNAME
  Le nom du projet a été modifié pour être représentatif de la librairie. Dans notre cas, nous y sommes allé général et l'avons appellé "mylib".

- PROJECTSRC
  Les fichiers sources de la librairie ont dû être changés dans cette section. Dans notre cas, nous n'avons qu'un seul fichier source (.cpp). Ainsi, seulement "Librairie.cpp" est inscrit dans cette variable

- $(TRG)
  La méthode de mise en commun des fichiers de compilation (.o) a été modifiée pour ne pas faire un executable mais plutôt une librairie. Ainsi, la commande

  ```
  $(CC) $(LDFLAGS) -o $(TRG) $(OBJDEPS) \
  -lm $(LIBS)
  ```

  dans la section $(TRG) a été remplacée par

  ```
  ar rsv -o $(TRG) $(OBJDEPS)
  ```

  Ainsi, unr archive est crée (ar) avec les arguments r (ajoute les fichiers), s (ajoute un index à l'archive) et v (verbose). Par la suite, on identifie où mettre l'archive (-o $(TRG)) puis finallement les fichiers concernés ($(OBJDEPS)).

Makefile de bidon des tests

- PROJECTNAME
  Le nom du projet a été modifié pour être représentatif du projet. Puisque ce projet servait seulement à faire des tests, le nom du projet est "test"

- PROJECTSRC
  Les fichiers sources des tests ont dû être changés dans cette section. Dans notre cas, nous avons fait des tests avec la mémoire et le convertisseur analogique-numérique donc les fichiers dans cette section sont "test.cpp", "can.cpp" et "Memoire24CXXX.cpp".

- LIBS
  La librairie crée par le Makefile de la librairie doit être joint au projet. Ainsi, dans cette section, nous avons écrit l'emplacement du fichier archive (.a) précédement crée "lib_dir/mylib.a".

- all
  Ajout de l'instruction
  ```
  make -C lib_dir
  ```
  qui appelle le makefile (make) de la librairie dans le fichier "lib_dir" (-C lib_dir).

- clean
  Ajout de l'instruction
  ```
  make -C lib_dir clean
  ```
  qui appelle dans le makefile (make) de la librairie dans le fichier "lib_dir" (-C lib_dir) l'instruction "clean" (clean).