

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 4081

Noms:

Philippe Désilets
Maximiliano Falicoff
Guillaume Nadeau
Alexis Thibeault

Date:
9 mars 2020

Partie 1 : Description de la librairie

Notre librairie contient les fichiers suivants (.cpp et .h):

- led
- ifpressed
- partirMinuterie
- PWM_roues
- connect_pc
- memoire_24
- Memoire
- can
- Photovolt

LED:

Définition: Contrôle de la couleur de la led sur le dessus du robot.

Fonctions:

clignoterLed: Elle prend une couleur (uint8_t couleur), une durée (uint16_t duree), un nombre de fois à répéter (uint8_t repeterXfois) et (bool minuterieExpiree) en paramètre. La couleur peut être verte ou rouge selon la valeur de couleur ainsi que le port relié à la led, la durée indique le temps que la lumière est allumée, mais aussi le temps qu'elle est éteinte et finalement on peut choisir le nombre de fois que l'on fait le clignotement avec repeterXfois.

ambre: Elle prend une durée (uint16_t) en paramètre. La led prendra la couleur ambre pour la durée indiquée qui doit être en milliseconde.

Utilité: Ces fonctions sont pratiques à avoir puisqu'elles permettent de faire un débogage rapide lorsque l'on travaille sur le code du robot et permettront de savoir dans quel état est le robot en programmant un switch case.

Ifpressed:

Définition: Permet de vérifier si une interruption est survenue auprès du bouton poussoir.

Fonctions:

ifpressed: La fonction vérifie que le bouton soit poussé, puis une minuterie de millisecondes se déclenche. Après, une deuxième vérification est faite, puis la fonction retourne un booléen, vrai si le bouton est encore enfoncé ou faux si il ne l'est pas.

Utilité: Ce fichier nous permet de savoir si le bouton poussoir a été poussé. On pourrait se servir de cette fonctionnalité pour faire allumer la DEL par exemple.

PartirMinuterie:

Définition: Active la minuterie (Timer 1) jusqu'à ce qu'il atteigne sa durée prédéfinie. (N'inclut pas l'ISR, le main.cpp s'en chargera)

Fonctions:

partirMinuterie: Cette fonction initialise les bits nécessaire à un, selon le mode de minuterie nécessaire.

Utilité:

- Ce fichier nous permet de produire une interruption qui peut amener un changement d'état de notre système.

PWM roues:

Définition: Émet un PWM sur deux broches d'un port.

Fonctions:

ajustementPWM: Elle prend un pourcentage (uint8_t pourcentage) qui va déterminer la moyenne de puissance générée sur les broches (soit le rapport a/b).

Utilité: Il est essentiel d'avoir un PWM de ce genre sur le robot pour le faire avancer tout étant capable d'effectuer d'autres opérations.

Connect_pc:

Définition: Gere la connexion entre le microcontrôleur et le PC.

Fonctions:

initialisationUART: Ajuste les registres pour permettre la réception et la transmission par l'UART0.

transmissionUART: Transmet des données (uint8_t donnee) vers le PC.

afficher: Affiche une donnée (unsigned char mots[]) sur le PC.

Utilité: Ses fonctions sont d'autres façons pour nous permettre le débogage de notre code lors du projet.

ControleMem:

Définition: Contrôle la lecture et l'écriture sur la mémoire.

Fonctions:

lecture: Permet de lire la mémoire d'une adresse (uint16_t addr) à une autre (uint16_t fin_lect) en plaçant les données dans une variable (uint8_t mot), sans oublier l'objet mémoire en paramètre pour permettre la lecture.

ecriture: Permet de lire la mémoire d'une adresse (uint16_t addr) à une autre (uint16_t fin_lect) en plaçant les données dans une variable (uint8_t mot), sans oublier l'objet mémoire en paramètre pour permettre l'écriture.

Utilité: Permet l'écriture en mémoire du microcontrôleur, cela peut nous être utile dans le débogage de notre code, on pourra savoir quelles sont les valeurs des variables en un certain moment.

Mémoire 24:

Définition: Comprend la classe mémoire fournie en classe.

Utilité: Permet d'utiliser les méthodes reliées à la mémoire.

Can:

Définition: Comprend la can mémoire fournie en classe.

Utilité: Permet d'utiliser les méthodes reliées au convertisseur analogique numérique.

Photovolt:

Définition: Donne une valeur sur 8 bits à la lecture de la photorésistance.

Fonctions:

photo: Retourne la valeur (uint8_t) d'une photorésistance sur un port et la met sur 8 bits avec un objet can (& photo) en paramètre.

Utilité: Permet de lire la photorésistance ce qui nous sera utile pour guider notre robot à travers la piste.

Partie 2 : Décrire les modifications apportées au Makefile de départ

Dans notre Makefile pour la librairie on a réalisé les changements suivants:

- PROJECTNAME = libstatic
- PRJSRC = \$(wildcard *.cpp) afin que tous les fichiers .cpp soient mis comme source.
- Pour le TRG: on change le type d'extension pour un .a
- Dans \$(TRG), on utilise pas avr-gcc pour la compilation de notre librairie, on utilise a la place avr-ar. De plus on a supprimé -lm \$(libs) parce que c'est la librairie en soit que l'on en a pas besoin.
- On supprime le make install car on en a pas besoin de l'installer sur le robot.

Dans le Makefile de notre exécutable main, on a repris le Makefile des TP précédents et on ajoute les paths vers la librairie.

- INC = -I ../libstatic
- LIBS = -l static -L ../libstatic (on émet le lib dans pour le nom libstatic car il est automatiquement mis a l'avant du nom)