

Programme de tso

**PROJET FIN DE SESSION 4**

**Véhicule télécommandé**

**Travail présenté**

**A**

**Daniel Breton**

**Alain Champagne**

**Yves Roy**

**Par**

**William Bordeleau**

**Samuel Duclos**

**Cégep Limoilou**

**Québec, 15/05/2017**

Table des matières

[Fonctionnement général : 3](#_Toc482621210)

[Schéma bloc : 3](#_Toc482621211)

[Fonctionnement des modes : 4](#_Toc482621212)

[Mode Automatique : 4](#_Toc482621213)

[Mode Manuel : 4](#_Toc482621214)

[Mode Inclinomètre : 4](#_Toc482621215)

[Mode Évitement : 4](#_Toc482621216)

[Schéma de câblage : 5](#_Toc482621217)

[Tableau descriptif du câblage : 6](#_Toc482621218)

[Explication du fonctionnement de la manette : 6](#_Toc482621219)

[Explication du fonctionnement Dallas : 7](#_Toc482621220)

[Procédure de vérification : 10](#_Toc482621221)

[Vérification matérielle : 10](#_Toc482621222)

[Vérification logicielle : 10](#_Toc482621223)

# Fonctionnement général :

Le véhicule est téléguidé et est donc commandé par une télécommande. La manette permet de choisir le mode voulu en appuyant sur les bouts suivant sur le clavier : A pour le mode suiveur de ligne, B pour le mode contrôle par clavier, C pour le mode évitement par capteur de distance et 9 pour le mode inclinomètre. La manette envoie des trames en Bluetooth à la carte Dallas qui va agir selon le mode et la commande. Le fonctionnement des modes est décrit plus bas.

# Schéma bloc :



# Fonctionnement des modes :

## Mode Automatique :

Le mode automatique est un mode où le véhicule suit une ligne noire au sol. Le suiveur de ligne est muni de 5 Del qui s’éteignent s’il détecte une ligne noire. Un PCF envoi en I2C une valeur 8 bit indiquant quelle Del est éteinte. Par la suite, l’octet est masqué puis rentré dans un switch case pour savoir comment il faut réagir. Ensuite, une dérivé améliore la réaction et la rend plus fluide. Les valeurs de vitesses sont envoyées aux roues.

## Mode Manuel :

Le mode manuel est un mode où le véhicule est contrôlé par le clavier I2C de la manette. Cinq boutons permettent de contrôler le véhicule : 2, 4, 5, 6 & 8. Ils servent respectivement à faire avancer, faire tourner à gauche sur place, arrêter le véhicule, faire tourner à droite sur place et faire reculer. Le STM32 envoie par la suite une trame au dallas par Bluetooth. La trame est analysée et la bonne commande est envoyée aux roues.

## Mode Inclinomètre :

Le mode inclinomètre est un mode où le véhicule est contrôlé par l’inclinomètre. Le programme de la manette lit la valeur des deux axes, celle des x et des y lorsque l’utilisateur fait bouger la manette. Un range minimum et maximum est inclue pour que l’arrêt, la marche avant et la marche arrière soit plus facile à faire. La valeur en y est convertie en vitesse allant de 0x0000 à 0x03FF et celle des x en différence de vitesse pour aller en diagonale. Les valeurs de vitesse sont envoyées par Bluetooth dans une trame de 12. Le Dallas reçoit la trame et converti les caractères en Int qui sont envoyés au roues.

## Mode Évitement :

Le mode évitement est un mode où le véhicule avance en ligne droite mais tourne lorsque l’un des capteurs détecte un objet. Dans ce mode, on lit les capteurs en I2C et lorsque la valeur de l’un des capteurs augmente, on tourne dans la direction opposé à l’objet.

# Schéma de câblage :

DS89C450

Suiveur

Carte I2C

Drive avant

Drive arrière

Capteur droit

Capteur gauche

Batterie

Roue avant/droite

Roue arrière/droite

Roue avant/gauche

Roue arrière/gauche

1

2

3

4

5

7

6

9

8

10

11

12

# Tableau descriptif du câblage :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Numéro** | **Rôle** | **Couleurs** | **Origine** | **Destination** |
| 1 | Alimentation 12V | Rouge/Noir | Batterie | DS89C450 |
| 2 | Alimentation 12V | Rouge/Noir | DS89C450 | Drive avant |
| 3 | Alimentation 12V | Rouge/Noir | Drive avant | Drive arrière |
| 4 | Alimentation 12V | Rouge/Jaune | Drive avant | Roue avant/gauche |
| 5 | Alimentation 12V | Rouge/Jaune | Drive avant | Roue avant/droite |
| 6 | Alimentation 12V | Rouge/Jaune | Drive arrière | Roue arrière/gauche |
| 7 | Alimentation 12V | Rouge/Jaune | Drive arrière | Roue arrière/droite |
| 8 | Signal drive | Bleu/Blanc | Carte I2C | Drive avant |
| 9 | Signal drive | Bleu/Blanc | Carte I2C | Drive arrière |
| 10 | Capteur distance gauche | Rouge/Bleu/Noir | Capteur distance gauche | Carte I2C |
| 11 | Capteur distance droite | Rouge/Bleu/Noir | Capteur distance droite | Carte I2C |
| 12 | Capteurs suiveur de ligne | Mauve/Blanc/Bleu/Jaune | Capteurs suiveur de ligne | Carte I2C |

# Explication du fonctionnement de la manette :

Le programme de la manette se divise en deux parties : La gestion du clavier et la gestion de l’inclinomètre. Pour la gestion du clavier, on fonctionne par interruption et dans ce cas-ci lorsqu’un bouton est appuyé. Lorsqu’un bouton est appuyé on lit le quel est appuyer et on met les valeurs de la trames correspondante dans un tableau. Par la suite, un checksum est calculé dans une fonction et le place dans les deux derniers bits de la trame. Si le bouton appuyé est un bouton de changement de mode ou de vitesse, il est automatiquement envoyé. Si c’est une touche de commande de direction du mode manuel soit les boutons 2, 4, 5, 6 ou 8, on vérifie si l’on est en mode clavier et si c’est le cas on envoie le tableau par le Bluetooth. Le programme gère uniquement l’inclinomètre si l’on est en mode inclinomètre puisque toute la gestion se déroule à l’intérieur d’une condition. La première opération est la lecture de l’axe de la valeur en x qui est par la suite afficher sur l’écran et ensuite on li la valeur en y qui est aussi afficher à l’écran. Ensuite, un treshold minimale de 0.25 sur les deux axes permet d’avoir une zone limite de lecture pour éviter que le véhicule n’arrête pas ou ne puisse pas aller en ligne droite. Ensuite, une fonction de conversion converti la valeur en y en vitesse allant de 0x000 à 0x03FF. Pour les diagonales, la valeur en x est convertie en valeur de vitesse à soustraire d’un côté du véhicule pour avoir des diagonales. La vitesse de chaque roue est mise dans un tableau en ascii qui va servir de trame et qui est envoyé par Bluetooth au dallas.

# Explication du fonctionnement Dallas :

Le programme décrit permet de contrôler le véhicule téléguide. Les différents modes sont

Fichiers d'entêtes prérequis lors de la compilation:

\_DeclarationGenerale.h

Ce fichier fut complètement réécrit pour accommoder la hiérarchie de fichiers de déclarations d'entêtes, définir de nouveaux types et macros.

Test.h

Ce fichier introduit de la compilation conditionnelle pour (dé)sélectionner les différents modes de test et modifier des paramètres.

ConversionKeilToIAR.h

Ce fichier fut modifié pour uniformiser la syntaxe avec la notation de Keil pour les SFR.

TypeCible.h

Ce fichier, lègue de l'ère de l'UPSD, fut éliminé.

ioDS89C450.h

Ce fichier fut modifié, surtout pour compresser le nombre de lignes de moitié.

Classes déjà réalisées:

CLInOutBase:

Cette classe permet de lire et d'écrire dans et sur les entrées/sorties de la Dallas (par défaut a partir de l'adresse 0xF800).

CLEcran:

Cette classe contient les fonctions pour afficher sur l'écran de la Dallas.

CLI2CBase:

Implémente la base des fonctions du protocole I2C permettant de manipuler des bits et des octets pour lire et écrire.

CLI2CPort:

Implémentation du protocole I2C pour la lecture/écriture simple ou séquentielle sur n'importe quel appareil supportant l'I2C.

CLCom2:

Cette classe permet de lire et d'écrire une ou plusieurs trames RS-232.

Classes pilotes de périphériques I2C:

CLDAC6574:

Cette classe gère l'écriture au convertisseur digital -> analogue. Nécessaire pour contrôler les roues.

CLIOPCF8574:

Cette classe gère la lecture et l'écriture des entrées/sorties du véhicule. Nécessaire pour lire les capteurs du suiveur de ligne.

CLADCMax1236:

Cette classe gère la lecture du convertisseur analogue -> digital. Nécessaire pour lire les capteurs de distance et la tension de la pile.

Classes utilitaires du véhicule:

CLRoue:

Cette classe hérite de CLDAC6574 et permet de contrôler l'écriture directe aux moteurs des roues.

Elle contient aussi des fonctions pour l'écriture a une vitesse différentielle d'une roue par rapport à la vitesse moyenne du véhicule.

CLCaptDist:

Cette classe hérite de CLADCMax1236 et permet de détecter un objet à distance d'un capteur situe à l'avant du véhicule.

CLPile:

Cette classe hérite de CLADCMax1236 et permet de calculer le pourcentage de charge de la pile d'alimentation du véhicule.

CLControleMoteurs:

Cette classe hérite de CLRoue et gère le déplacement d'un véhicule a quatre roues.

CLSuiveurLigne:

Cette classe hérite de CLControleMoteurs et de CLPCF8574 et implémente un contrôleur PID pour régulariser le véhicule lorsqu'il suit une ligne.

Classe composée et programme principal:

CLVehicule:

Cette classe principale hérite de CLControleMoteurs et est composée des capteurs de distance, du suiveur de ligne, de CLCom2 et de CLPile.

Elle implémente la compilation conditionnelle pour séparer le mode technicien (tests) des autres modes.

Cette compilation conditionnelle permet d'alléger le fichier projet.hex et d'empêcher le reverse engineering des procédures de test.

Les modes suivants sont implémentés: TECHNICIEN, AUTOMATIQUE, INCLINOMETRE, CLAVIER et EVITEMENT.

Le programme principal débute l'exécution dans la routine principale de contrôle.

Le port série est vérifié et lu en entrée.

Le mode choisi est sélectionné et exécute.

La gestion de la pile est appelée.

Puis l'exécution revient à la boucle du programme principal.

MainVehicule:

Le programme principal hérite de CLVehicule et ne fait qu'appeler la fonction principale de contrôle du véhicule, en boucle.

Explication des différents modes:

TECHNICIEN:

Le mode TECHNICIEN est accessible seulement par compilation conditionnelle.

Cela a pour effet d'alléger la taille du code et d'empêcher le désassemblage des procédures de test.

Il permet de tester tous les autres modes en contournant la gestion du programme principal.

Il s'active et se configure dans le fichier Test.h.

# Procédure de vérification :

## Vérification matérielle :

Si le véhicule ne marche pas, regarder la fuse. Ensuite vérifie si tous les fils sont bien connectés. Si les moteurs ne tournent pas assez forts, changez la batterie. Si les roues ne tourne pas dans le bon sens, allumer la batterie avec le dallas en mode programmation, si les quatre roues tourne à 100% vers l’arrière, le problème est logiciel, sinon intervertir les fils roues et jaune correspondant à la roue.

## Vérification logicielle :

Le programme est muni de plusieurs fonctions de test utilisable avec la compilation conditionnelle. En cas de doute sur la survie des capteurs, le test de capteur affiche à l’écran la valeur lu par les capteurs. En cas de dysfonctionnement des roues, une fonction de test individuel permet de tester chaque roue individuellement. Si c’est la trame, une fonction pour tester le port série est inclue afin de voir si le problème est logiciel ou matériel.