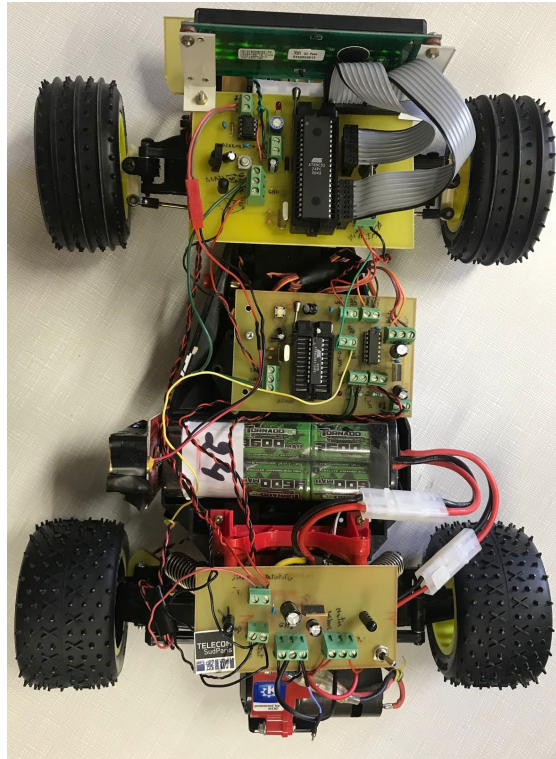
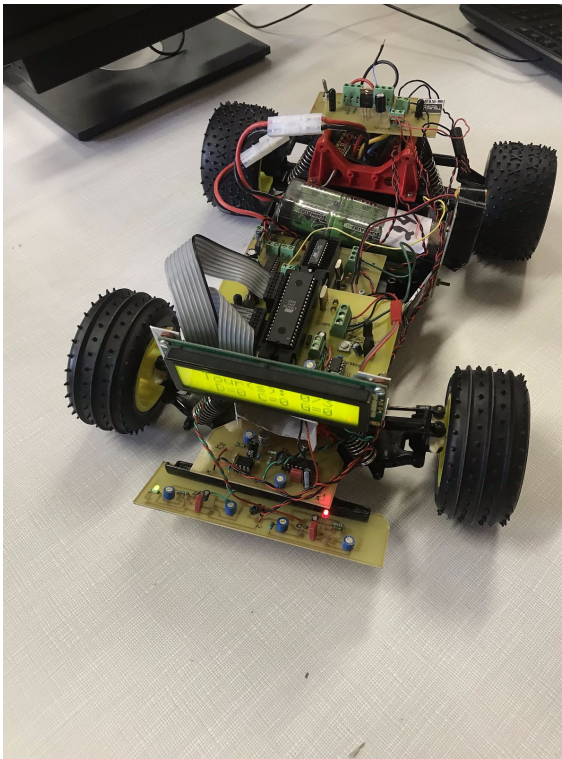


Intégration des Systèmes Électroniques

Rapport groupe 2

SOTON Dylan VÉDIE Victor DUBOC Aurélien
COUPRIE DIAZ Téo



Introduction

Ce rapport a pour objectif de vous présenter les choix électroniques réalisées ainsi que le contexte de programmation des cartes maîtres et esclaves de la voiture répondant au cahier des charges suivant:

- La voiture doit réaliser trois tours de piste en adaptant automatiquement sa direction
- Elle doit démarrer sous l'ordre d'une balise de départ qui lui permet aussi de définir son nombre de tour accomplis
- A l'approche d'une deuxième balise située à mi-chemin, elle doit à l'aide d'un laser envoyer un signal modulé à la seconde balise qui dispose de 3 cibles, le but étant de toucher les cibles avec le laser.
- Tout en effectuant ses 3 tours, la voiture doit afficher à l'aide de son écran LCD les informations jugées importantes, telles que le nombre de tours révolus ainsi que le nombre de cible touchées

Carte maître

La carte maître a deux fonctions principales : celle de gérer la propulsion et la direction du véhicule de façon fluide et celle de réagir aux commandes lancées par la carte esclave : arrêt et départ du véhicule.

Sur la carte maître on trouve le microcontrôleur AT89C2051, deux boutons libres, un bouton de reset, les sorties de contrôle de direction et de propulsion ainsi que les entrées de détection de la piste.

En début de projet des règles de programmations ont été décidées pour permettre de coder à plusieurs sans empiéter sur les ressources utilisées par d'autre partie du programme. Par exemple la banque 0 est réservée pour le passage et retour de paramètres des routines, la banque 3 est utilisée comme banque pour les résultats temporaires.

Direction

Détection de la piste

La piste qui doit être suivie est constituée d'une bande noire entourée de deux bandes blanches. Nous disposons de deux couples capteur IS471F/diode infrarouge SFH485. Chaque capteur module la lumière de la diode et traite la lumière réfléchi afin de détecter les zones les moins réfléchissantes. La sortie de détection de chaque capteur est ensuite connecté au microcontrôleur. On résume l'état du capteur au dessus d'une bande blanche ou noire dans le tableau suivant :

	Blanche	Noire
État LED témoin	Allumée	Eteinte
État DETECT	0	1

Il est possible de régler l'intensité des diodes à l'aide de potentiomètres. Cela permet d'ajuster la sensibilité de détection du système, pour une utilisation dans les

environnements très lumineux par exemple. De plus, l'orientation des DEL ainsi que des capteurs doit être ajustée afin d'être correctement positionnés vis à vis de la piste et d'être aussi réactif que possible quand la voiture s'éloigne de son trajet.

Nous avons fait le choix de baser notre direction sur la détection de la piste noire centrale. Ainsi, lorsque la voiture est décalée sur la gauche par exemple, le capteur de droite détectera la piste noire. Nous aurons donc la pin correspondant à la détection de la piste à droite à l'état haut, signalant qu'il faut corriger le déplacement à droite. Cette logique de détecter la direction dans laquelle le mouvement doit être corrigé est la motivation de ce choix.

Commande de la direction

La direction est gérée par un servomoteur contrôlé à l'aide d'un PWM de période 20ms dont l'état haut varie, en théorie, entre 1ms (Direction à gauche) et 2ms (Direction à droite), centré autour de 1.5ms (Direction alignée avec la voiture). Ce PWM est généré de manière logicielle par le microcontrôleur de la carte maître. De plus, il est contrôlé de façon à ne pas donner d'à-coups en faisant évoluer la direction par petits incréments jusqu'à la nouvelle position. Ainsi, la direction est plus fluide et moins susceptible de faire fortement osciller le véhicule ou même de le faire sortir de piste par une suite de corrections trop rapides et brutales.

Afin de pouvoir contrôler correctement la voiture, il est nécessaire de s'assurer des valeurs théoriques de ce PWM. En effet, il n'est pas garanti que la position "roue droites" corresponde exactement à 1.5ms de temps haut. De plus, les roues sont bloquées dans leur rotation par le châssis du véhicule. Il faut donc connaître les valeurs maximales dans chaque direction. On les obtient grâce au protocole suivant:

- Relier la pin de direction du moteur à un générateur de fonction électronique grâce auquel on générera un PWM
- Brancher la batterie de la partie moteur/direction de la voiture
- Émettre un PWM à 50Hz avec un temps haut de 1.5ms et examiner l'alignement des roues
 - Si les roues sont droites, la valeur centrale est bien 1.5ms

- Si les roues ne sont pas droites, il faut changer la valeur de temps haut du PWM jusqu'à trouver la valeur à laquelle elles le sont en l'augmentant si elles sont trop à gauche et en la diminuant si elles sont trop à droite
- À partir de la valeur centrale, incrémenter la durée du temps haut jusqu'à ce que les roues ne puissent plus tourner à gauche. On obtient ainsi la valeur limite pour tourner à gauche.
- Faire de même en décrémentant pour obtenir la valeur limite pour tourner à droite.

Les valeurs que nous avons obtenu sur notre véhicule sont les suivantes :

- Roues droites : 1.500ms
- Rotation maximale à gauche : 1.833 ms
- Rotation maximale à droite : 1.088 ms

Nous observons donc que la valeur centrale est bien celle prévue théoriquement. Cependant, on observe aussi qu'on a plus de valeurs accessibles à droite qu'à gauche : 0,412 ms pour l'un et 0.333ms pour l'autre. Le véhicule aura donc plus de difficulté à tourner à gauche ce qui va impliquer une vitesse maximale plus faible.

Propulsion

Commande du moteur

La vitesse et le sens de rotation du moteur sont gérés par un contrôleur commandé via un PWM de période 20ms, avec un état haut pour l'arrêt de 1.273ms et un état haut pour la vitesse de marche déterminé par l'asservissement.

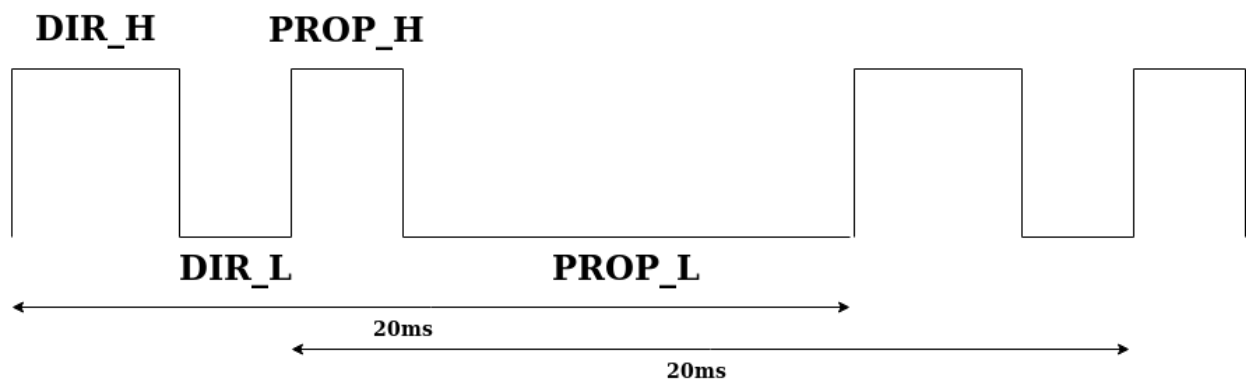
Pour déterminer la vitesse actuelle de la voiture, nous utilisons un encodeur optique rotatif composé de 16 bandes noires et blanches alternées à l'intérieur de la roue arrière gauche. Suivant le même principe que la détection de la piste pour la direction, nous utilisons une LED qui éclaire le disque de l'encodeur. Nous comptons, via interruption sur front descendant, le nombre de transitions entre une bande blanche et une bande noire pendant une période. Nous avons choisi une période de 500ms (25 périodes de PWM) car c'est un juste milieu entre le temps de mise à jour et le nombre de fronts descendants que nous pouvons compter. Nous pouvons ainsi déterminer la vitesse de la voiture, et décider d'augmenter, diminuer, ou stabiliser la valeur du PWM à envoyer au moteur. Ce système

permet d'avoir des accélérations douces et de s'affranchir de l'état de charge des batteries qui joue grandement sur les performances du moteur.

Implémentation

Du fait des limitations du AT89C2051 qui possède seulement 2 timers, dont un que nous avons alloué à la communication série, il nous est obligatoire d'implémenter un PWM multiplexé pour gérer les PWM de la direction et de la propulsion via le timer restant. Le timer PWM déclenchant une interruption lors de son overflow, le changement de la valeur de chargement se faisant dans cette interruption

Le principe du PWM multiplexé est de faire varier la valeur de chargement du timer pour la durée du niveau logique suivant en rapport avec le PWM à générer. Il peut être visualisé sur le schéma ci-dessous.



DIR_H : valeur de chargement correspondante au temps du niveau logique HAUT pour le PWM de la direction (varie suivant la commande de direction)

DIR_L : valeur de chargement correspondante à l'écart temporel CONSTANT entre DIR_H et PROP_H (et non pas le temps du niveau logique BAS pour le PWM de la direction, qui se calcul en faisant $20\text{ms} - \text{DIR}_H$)

PROP_H : valeur de chargement correspondante au temps du niveau logique HAUT pour le PWM de la propulsion (varie suivant la commande de propulsion)

PROP_L : valeur de chargement correspondante à l'écart temporel VARIABLE entre PROP_H et le prochain DIR_H (et non pas le temps du niveau BAS pour le PWM propulsion, qui se calcul avec 20ms-PROP_H).

Nous avons le choix entre deux méthodes pour implémenter le PWM multiplexé, en faisant varier DIR_L et PROP_L pour compenser l'incrément ou le décrément de DIR_H et PROP_H, ou bien en laissant DIR_L constant et appliquer les compensations seulement sur PROP_L. Cette dernière solution a été retenue. Nous avons en revanche des inquiétudes sur le fait que cette méthode puisse créer des imprécisions sur la propulsion lors d'un changement de direction (l'écart entre deux DIR_H ne faisant plus exactement 20ms pendant un court laps de temps). Pendant les tests sur piste, aucun problème de ce type ne fut détecté, et cette solution fût donc conservée.

La valeur de PROP_L est calculée suivant la relation suivante :

$$\text{PROP_L} = (0xFFFF - 0xB1E0) + (0xFFFF - \text{DIR_H}) + (0xFFFF - \text{DIR_L}) + (0xFFFF - \text{PROP_H})$$

Carte esclave

Dialogue avec l'écran LCD

Le rôle majoritaire de la carte esclave est le traitement des informations et son utilisation à travers un affichage compréhensible. Voici les informations affichées par l'écran LCD que commande la carte esclave:

- Affichage du nombre de tours de piste effectuées
- Affichage du nombre de message correspondant aux valeurs ascii D/C/G reçus.

Une première initialisation de l'écran LCD au démarrage de la carte esclave va mettre en place cet affichage. Lors des différents tours, seuls certains caractères seront mis à jour ce qui facilite grandement le traitement de donnée ainsi que la complexité du code.

On a parfois besoin de modifier un seul caractère affiché sur l'écran, comme c'est le cas lorsqu'un tour de piste vient d'être effectué ou que l'on vient de recevoir sur le capteur un message correspondant aux valeurs ascii D, C ou G. En se référant à la documentation de l'écran LCD, il a fallu récupérer les adresses correspondant aux caractères que nous souhaitions modifier. Ainsi nous pouvions remplacer un unique caractère à la fois, ceci nous évitant de réécrire l'intégralité du message à chaque mise à jour des informations.

Réception des données infrarouges

Tout au long de son périple, notre voiture va recevoir des informations provenant de 2 balises, celle de départ et la balise cible. Ces informations sont transmises par modulation d'un signal infrarouge à 38kHz. Ainsi la voiture peut recevoir 4 caractères utiles, qui une fois démodulés par le TSOP1738 (connecté en série à l'AT89C51 de la carte esclave) sont les caractères ASCII suivant: 0,4,D,C et G.

Le 0 est transmit par la balise de départ, il va nous permettre de mettre en route la voiture ainsi que de mettre à jour le nombre de tours effectués.

Le 4 est envoyé par la balise cible lorsqu'aucune des cibles n'est activée. Servant de signalisation à la voiture, la détection d'un 4 implique l'approche de la balise cible, et le LASER est alors activé, ainsi que la sirène pour des raisons évidentes de sécurité...

Dès que le laser touche une cible (photodiode) la balise cible cesse d'émettre un 4 et va le remplacer par un caractère ASCII correspondant à la cible touchée. D pour la balise droite, C pour la centre et G pour la gauche.

Après réception du G, notre LASER est éteint et la voiture est alors en attente d'un 0 pour mettre son compteur de tour à jour.

Comme les balises émettent en permanence et à une fréquence élevée, nous avons dû mettre en place un système de sélection des données. Aussi, nous ne prenons en compte un caractère ASCII que si il est différent du précédent reçu. Par exemple, ceci évite à la voiture de toucher plusieurs fois la balise droite sur un seul tour.

Communication avec la carte maître

Envoi des données

La communication avec la carte maître est réalisée par une liaison série, nous avons établi la méthode suivante: la carte maître transmet un message sur 8 bit correspond à un certain caractère ASCII:

- S (= stop): consigne d'arrêt du moteur
- G (=go): consigne de redémarrage du moteur

Lors des deux premiers tours, les consignes d'arrêt et de redémarrage du moteur sont successives espacées de quelques secondes à l'aide de la fonction "timer 50ms" répétées plusieurs fois. Lors du dernier tour, la consigne de redémarrage.

Mise en place de la liaison série

Pour la communication en liaison série on attribue 01010000 au SCON. On rappellera que SCON est le registre permettant de contrôler la liaison série. Dans notre cas le mode 1

est sélectionné c'est à dire que l'on peut réaliser une transmission full-duplex asynchrone avec 10 bits de transmission par donnée:

- 1 bit de début à 0
- 8 bits de données (particularité: le bit de poids faible est en premier)
- le bit de fin à 1

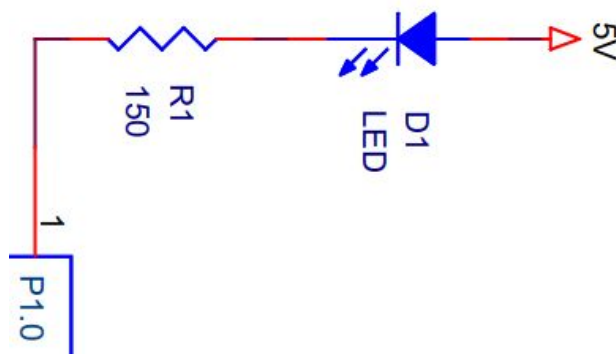
On valide la réception (REN=1)



FIGURE 17 – Registre de fonction spéciale SCON

Debugging

Une LED connectée à P1.0 était utilisée pour du debugging. Nous l'avons majoritairement utilisée pour vérifier si les capteurs ont bien reçus les messages de valeurs ascii 0, 4, D, C ou G. Sur le schéma de la carte esclave on remarque que cette LED est connectée au 5V, c'est donc l'instruction CLR qui permettra d'allumer cette LED (il faut initialiser avec un SETB donc). En effet, on a besoin de réaliser une différence de potentiel de 5V pour allumer cette LED.



Annexe

A.4. CARTE MAÎTRE

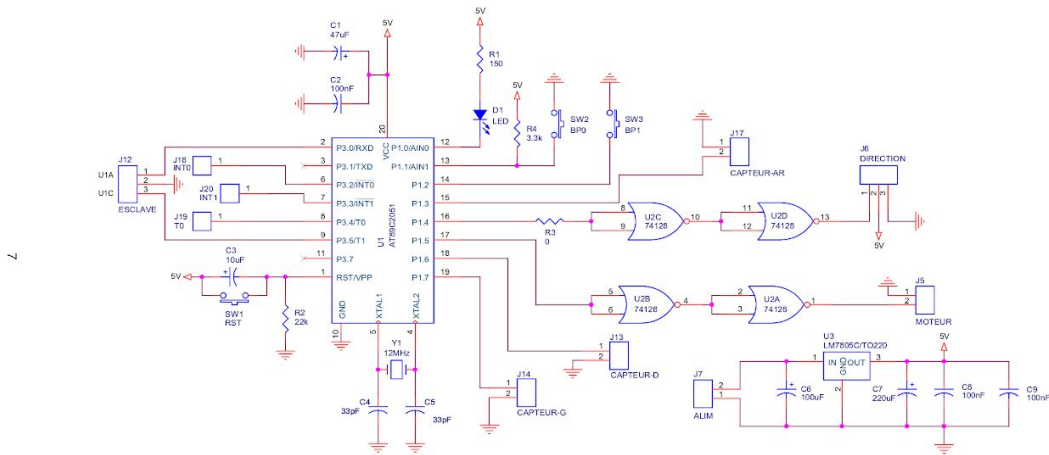


FIGURE A.4 – Schéma électrique de la carte maître

A.5. CARTE ESCLAVE

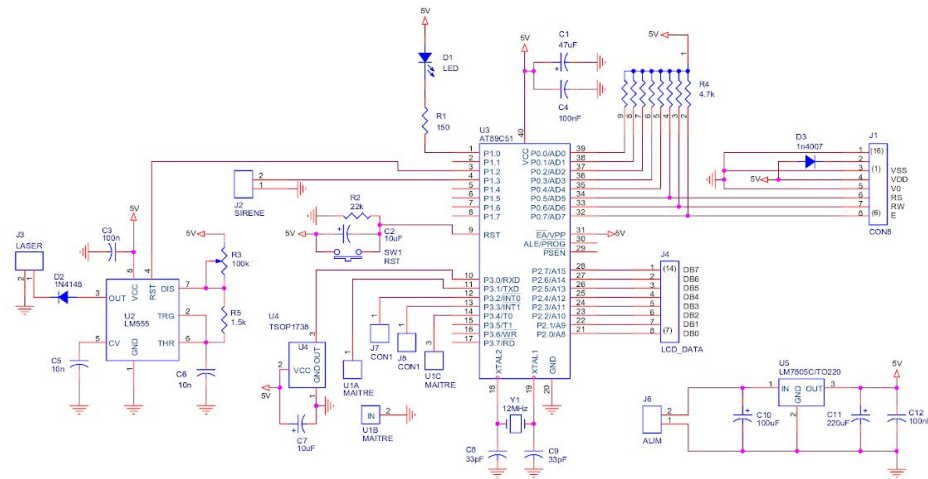


FIGURE A.5 – Schéma électrique de la carte esclave