Projet d’Ingénierie Electronique pour le Traitement de l’Information

Dossier Technique Voiture

Autonome 2022

Descrition du projet

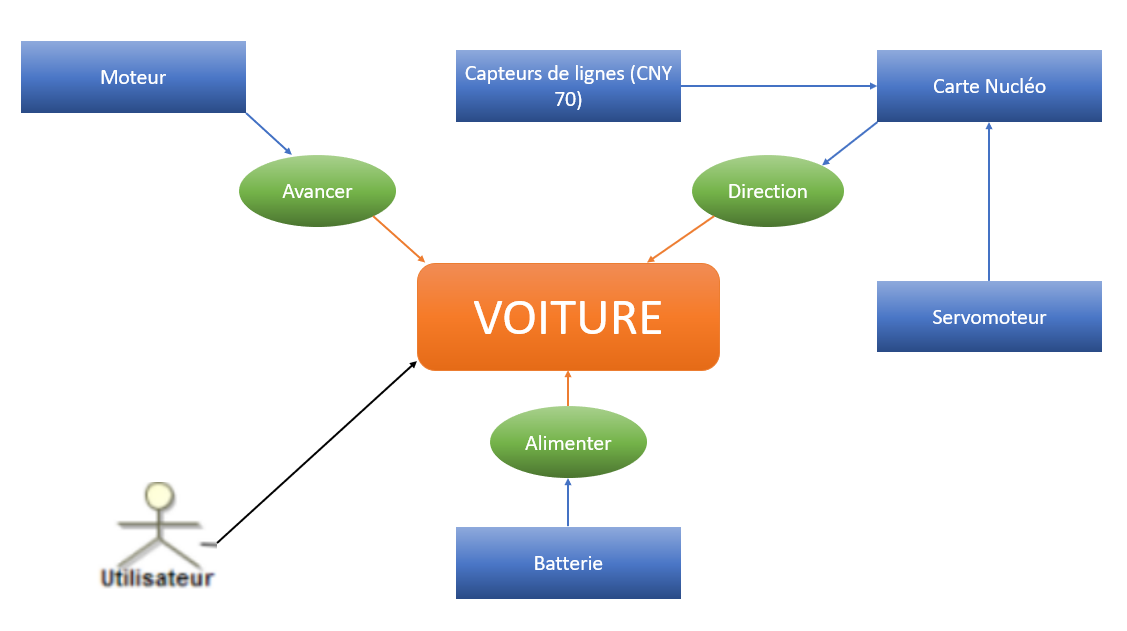
Objectif

Notre projet est une voiture autonome qui suit une ligne sur terrain plat. Celle-ci doit être autonome énergétiquement.

Matériel

Pour cela, nous utilisons un châssis Tamiya Lancia Delta de voiture téléguidée qui est fournit avec une batteries et un servomoteur. On a équipé le châssis de six capteurs CNY70, d’une carte nucleo, d’une breadboard afin de réaliser le branchement des capteurs et de faire fonctionner la voiture grâce à une carte embarquée.

Découpage fonctionnel



Une image contenant table

Description générée automatiquementCahier des charges

Capteurs

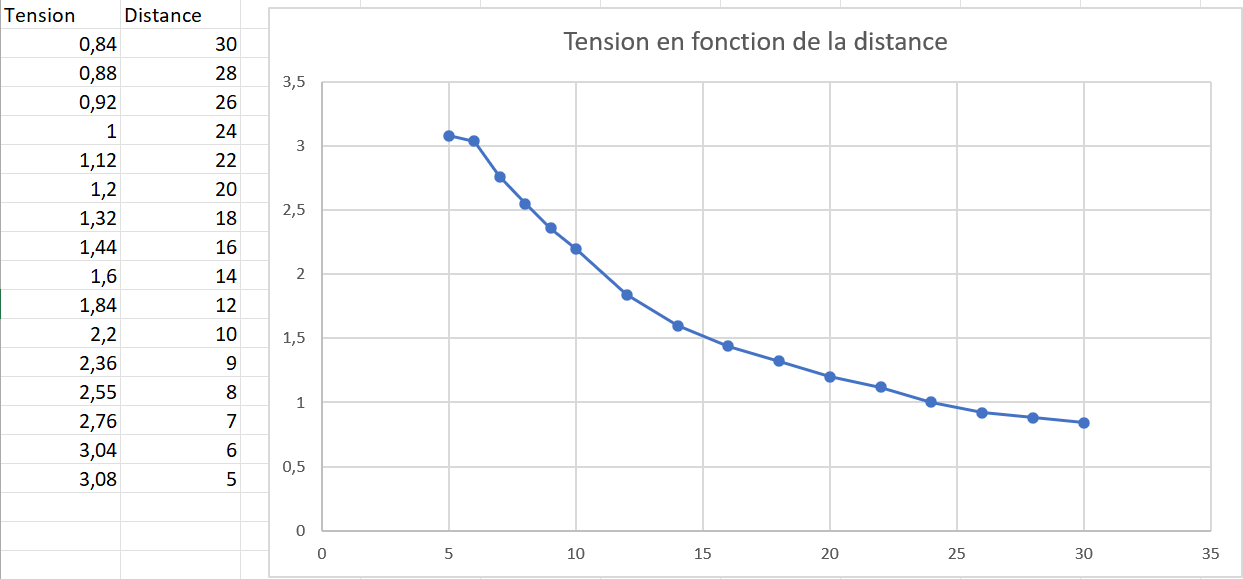
Capteur Sharp

Pour pouvoir s’assurer que la voiture ne s’endommage pas, il est nécessaire de lui permettre de détecter les obstacles sur sa route. Pour se faire nous avons voulu installer un capteur Sharp en avant de la voiture. Ce capteur mesure par effet Doppler la distance qui le sépare de l’obstacle.

Une image contenant texte, intérieur, encombré, guichet

Description générée automatiquement

Figure 1 : montage permettant de déterminer la loi entrée-sortie

Par ailleurs, pour savoir si l’objet est très proche de la voiture et donc savoir s’il faut déclencher une condition d'arrêt, il faut déterminer la loi entrée- sortie du capteur U(x) qui donne la tension aux bornes du capteur en fonction de sa distance à un objet.

Capteur CNY 70

**Principe de fonctionnement du capteur**

Le capteur CNY70 est un capteur composé d’un émetteur infrarouge et d’un récepteur aligné. Nous avons décidé de placer six capteurs sous le châssis de la voiture. Pour détecter la ligne nous avons décidé de comparer les intensités lumineuses issues de la réflexion de l’onde infrarouge sur le sol. En effet le capteur envoie une onde infrarouge sur le sol, cette onde est réfléchie par celui-ci, cette onde est alors captée par le récepteur qui nous renvoie une tension. Selon la tension renvoyée par le capteur on en déduit la couleur du sol en dessous du châssis, les tensions faibles correspondent à du noir et les tensions élevées à du blanc, et donc si la voiture suit bien la ligne ou non.

**Problèmes rencontrés et solution proposée**

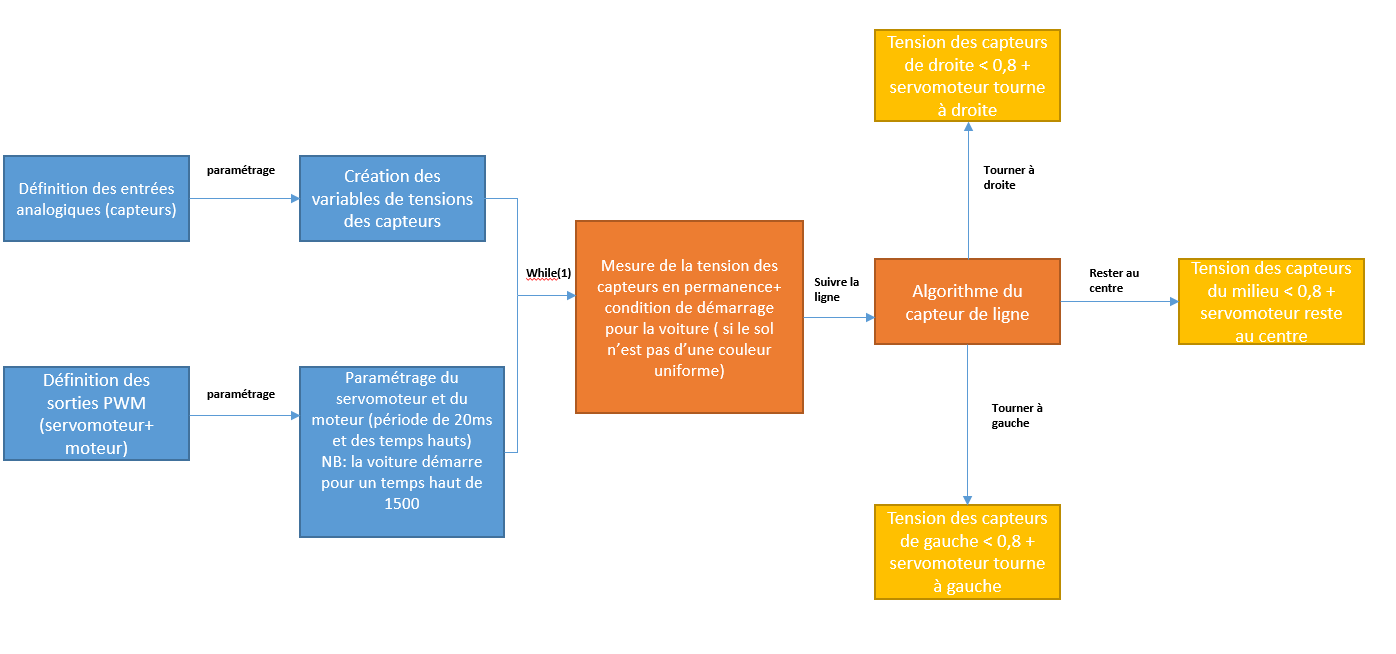
Au départ nous n’avions placé que trois capteurs CNY70 en dessous de la voiture. Nous nous sommes rendu compte que deux problèmes se posaient à nous.

* Le circuit n’était pas adapté au capteur, en effet nous avions choisis de construire le circuit avec seulement une ligne noire. La différence de tension émise par le capteur pour les lignes noires et celle pour le sol de la salle vert était trop faible. Ce qui engendrait des erreurs de directions de rotation de la voiture sur le circuit.
* La voiture tournait trop abruptement, car il était impossible de conditionner précisément la direction de rotation, ce qui la faisait sortir du circuit.

La solution à ce problème était donc d’encadrer la ligne noire de deux lignes blanches et de placer six capteurs cote à cote sous la voiture.

Programme embarqué

Vue d’ensemble du code



Nos capteurs sont placés de telle sorte que lorsque la voiture se trouve sur la ligne, différents cas se posent à notre algorithme. Ceux-ci sont décrit par le tableau ci-dessous :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Capteur | Gauche 1 | Centre 1 | Droite 1 | Gauche2 |  | Centre2 | Droite 2 |
| Placement initial |  |  |  |  |  |  |  |
| Condition de rotation  (V en volts) | V<0,8 | V<0,8 | Vdroite1<0,8 et Vgauche1<0,8 | Vdroite1<0,8 et Vgauche1<0,8 |  | V<0,8 | V<0,8 |
| Direction de rotation | Gauche | Gauche | Tout droit | Tout droit |  | Droite | Droite |
| Amplitude de rotation | Forte | Faible | Aucune | Aucune |  | Faible | Forte |
| Variation de la vitesse de la voiture | Diminue | Diminue | Constante | Constante |  | Diminue | Diminue |

**Limites techniques**

Par ailleurs, certaines limites mécaniques et électroniques restent présentes pour cette voiture. Le servomoteur à une limite d’angle de rotation, le parcours ne doit donc pas contenir de virages trop serrés.

De plus, le temps de réponse des capteurs du servomoteurs et de la carte nucléo limitent la vitesse de la voiture, en effet si la voiture va trop vite et, lorsque la voiture doit tourner, le capteur n’a pas le temps d’informer qu’il y’a eu un changement de couleur perçue et la voiture dévie de sa trajectoire.

Pour permettre à la voiture d’aller plus vite, il aurait donc fallu rajouter à notre système un correcteur PID qui aurait permis au système d'être plus rapide et précis.

Il aurait été possible de rajouter un capteur Sharp qui permet de détecter des obstacles présents sur le circuit de la voiture. On aurait pu rajouter une télécommande qui nous permettrait d’éviter cette obstacle et de remettre la voiture sur la ligne.

Annexes

#include "mbed.h"

#include "platform/mbed\_thread.h"

AnalogIn droite1(A0);

AnalogIn centre1(A2);

AnalogIn gauche1(A4);

AnalogIn gauche2(A5);

AnalogIn centre2(A3);

AnalogIn droite2(A1);

PwmOut moteur\_cc(PC\_9);

DigitalOut myled(LED1);

PwmOut servo\_mot(PB\_13);

Serial pc(USBTX, USBRX);

int main()

{

double rc = 1500;

int meas\_gauche1, meas\_centre1, meas\_droite1,meas\_gauche2, meas\_centre2, meas\_droite2;

double tensiongauche1, tensioncentre1, tensiondroite1, tensiongauche2, tensioncentre2, tensiondroite2;

double a =1300 ;

servo\_mot.period\_ms(20);

servo\_mot.pulsewidth\_us(a);

moteur\_cc.pulsewidth\_us(rc);

wait\_us (3000000);

meas\_gauche1 = gauche1.read\_u16(); // sur 12 bits MSB

tensiongauche1 = meas\_gauche1 / 65536.0 \* 3.3; // en V

meas\_centre1 = centre1.read\_u16(); // sur 12 bits MSB

tensioncentre1 = meas\_centre1 / 65536.0 \* 3.3; // en V

meas\_droite1 = droite1.read\_u16(); // sur 12 bits MSB

tensiondroite1 = meas\_droite1 / 65536.0 \* 3.3; // en V

meas\_gauche2 = gauche2.read\_u16(); // sur 12 bits MSB

tensiongauche2 = meas\_gauche2 / 65536.0 \* 3.3; // en V

meas\_centre2 = centre2.read\_u16(); // sur 12 bits MSB

tensioncentre2 = meas\_centre2 / 65536.0 \* 3.3; // en V

meas\_droite2 = droite2.read\_u16(); // sur 12 bits MSB

tensiondroite2 = meas\_droite2 / 65536.0 \* 3.3; // en V

while (1){

meas\_gauche1 = gauche1.read\_u16(); // sur 12 bits MSB

tensiongauche1 = meas\_gauche1 / 65536.0 \* 3.3; // en V

meas\_centre1 = centre1.read\_u16(); // sur 12 bits MSB

tensioncentre1 = meas\_centre1 / 65536.0 \* 3.3; // en V

meas\_droite1 = droite1.read\_u16(); // sur 12 bits MSB

tensiondroite1 = meas\_droite1 / 65536.0 \* 3.3; // en V

meas\_gauche2 = gauche2.read\_u16(); // sur 12 bits MSB

tensiongauche2 = meas\_gauche2 / 65536.0 \* 3.3; // en V

meas\_centre2 = centre2.read\_u16(); // sur 12 bits MSB

tensioncentre2 = meas\_centre2 / 65536.0 \* 3.3; // en V

meas\_droite2 = droite2.read\_u16(); // sur 12 bits MSB

tensiondroite2 = meas\_droite2 / 65536.0 \* 3.3; // en V

pc.printf("G1=%lf - C1=%lf, D1=%lf, G2=%lf - C2=%lf, D2=%lf\r\n", tensiongauche1, tensioncentre1, tensiondroite1,tensiongauche2, tensioncentre2, tensiondroite2);

if ( (tensiongauche1 >= 0.5) && (tensioncentre1 >= 0.5) && (tensiondroite1 >= 0.5) && (tensiongauche2 >= 0.5) && (tensioncentre2 >= 0.5) && (tensiondroite2 >= 0.5) ) {

myled = 1; // La LED est allumée

wait\_us(20000); // Attente de 200 ms

myled = 0; // La LED est allumée

wait\_us(20000); // Attente de 200 ms

rc = 1500 ;

moteur\_cc.pulsewidth\_us(rc);

}

else {

if (rc<= 1610){

rc= rc+1 ;

moteur\_cc.pulsewidth\_us(rc);

if (tensiongauche1 <= 0.8){

rc=1605;

wait\_us(1000);

a=a-100;

moteur\_cc.pulsewidth\_us(rc);

servo\_mot.pulsewidth\_us(a);

}

else{

if (tensioncentre1 <= 0.8){

rc=1605;

wait\_us(1000);

a=a-100;

moteur\_cc.pulsewidth\_us(rc);

servo\_mot.pulsewidth\_us(a);

}

else{

if (tensiondroite2 <= 0.8){

rc=1605;

wait\_us(1000);

a=a+100;

servo\_mot.pulsewidth\_us(a);

moteur\_cc.pulsewidth\_us(rc);

}

else{

if (tensioncentre2 <= 0.8){

rc=1605;

wait\_us(1000);

a=a+100;

servo\_mot.pulsewidth\_us(a);

moteur\_cc.pulsewidth\_us(rc);

}

else{

if ((tensiongauche2 <= 0.8) && (tensiondroite1 <=0.8) ) {

a=1300 ;

servo\_mot.pulsewidth\_us(a);

}

}

}

}

}

}

}}}