



Suivi de ligne et de lumière

1. Introduction

Une application robotique traditionnelle est le suivi d'une ligne tracée sur le sol. Les premières tortues de Grey Walter avaient un comportement photovore et photophobe facile à programmer. Enfin, une jolie application pour une classe est d'avoir des robots qui se suivent, chaque robot s'accrochant à l'ampoule portée par le robot précédent.

Il faut maîtriser plusieurs niveaux de problèmes:

- 1) **mécanique:** fixer les composants sur le robot au bon endroit, de façon stable mais avec des possibilités de réglage
- 2) **optique:** définir la zone de vision, protéger de la lumière parasite, ajouter éventuellement une optique améliorant l'efficacité
- 3) **électrique:** choisir le capteur, définir son interface vers le processeur, obtenir des niveaux mesurables compatibles avec le E/S du contrôleur
- 4) **interface:** choisir sur quelles lignes du processeur il faut se câbler, et mettre le port dans le bon mode
- 5) **test:** définir un environnement de test proche des conditions d'utilisations et écrire un programme de test permettant de vérifier que les composants et leur mise en œuvre sont corrects, et que l'on sait comment ajuster les paramètres si les conditions changent
- 6) **programmation:** choisir l'algorithme du comportement, définir la structure du programme, programmer
- 7) **améliorations:** tester, modifier, prévoir le nouveau comportement, tester et vérifier son raisonnement, etc..

Il est évident que ces étapes ne peuvent pas être toutes effectuées dans un environnement scolaire, elles prendraient trop de temps et les jeunes veulent commencer par le point 6 et après quelques essais aléatoires, soit cela marche, soit ils abandonnent.

Une démarche de développement minimale est par exemple:

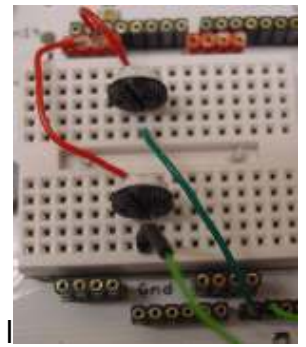
- 1) prendre une solution testée
- 2) idem
- 3) étudier le capteur sur le bloc d'expérimentation, pour comprendre son comportement électronique
- 4) voir comment lire le signal électronique du capteur, afficher sa valeur, prendre une décision selon cette valeur
- 5) tester une solution qui présente mieux l'information, pour 2 capteurs, permet d'ajuster des paramètres. Se concentrer sur la qualité des signaux.
- 6) programmer des versions successives facilitant le déverminage et l'observation et aller vers la complexité par étapes bien maîtrisées. Documenter l'objectif de chaque étape et naturellement conserver le programme correspondant
- 7) tester le plus loin possible en gardant le robot connecté, en déplaçant la piste, la source de lumière par rapport au robot et pas l'inverse.

Naturellement, ces dernières étapes sont plus rapides si la technologie est bien connue. On laisse quelques points dans l'incertitude pour que les élèves réfléchissent.

2. Suivi de ligne avec le Ddr1

Le Ddr1 voulait faciliter l'application du suivi de ligne, et la solution proposée est de fixer "en l'air" deux LDR dessous. Une bande foncée large doit être utilisée et si l'éclairage donne une bonne variation de tension avec le potentiomètre proche du centre, c'est jouable.

Les potentiomètres sur la carte doivent être désactivés pour un bon résultat. Pour permettre un exercice simple de lecture analogique, ces potentiomètres sont câblés au + et -. Ceci réduit la gamme de réglage. De plus, leur valeur de 22k semble être trop faible dans des éclairages normaux. Couper les cavaliers et mettre deux pots de 100k (?) sur le bloc d'expérimentation, câblés en résistance variable entre le +5V et les pins 14 et 15.



3. Suivi de ligne avec le Ddr2

3.1 Nouveau schéma

Les potentiomètres et connecteurs ont été supprimés sur le Ddr2, et un circuit imprimé (ou comme précédemment mettre les pots sur le bloc d'expérimentation) résout de façon plus fiable le problème du suivi de ligne. Trois circuits ont été développés pour le suivi de piste et le suivi de lumière – voir www.didel.com/diduino/Ddr2PisteEtSuivi.pdf

Ces circuits utilisent des composants discrets, avec l'idée qu'après avoir compris l'électronique et la programmation avec un capteur (étapes 3,4), les élèves soudent le circuit et continuent les tests. Les circuits sont aussi disponibles montés.

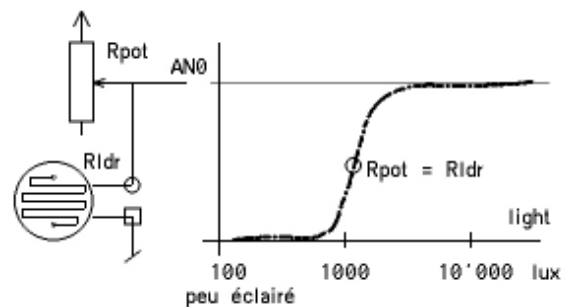
3.2 Circuit de suivi de piste

Le circuit Ddr2Suivi se fixe sur le connecteur avant. Dans la version minimale, l'orientation des capteurs, le cache optique éventuel sont laissés à l'initiative des élèves. Le choix de la bande, réfléchissante ou absorbante, la calibration des capteurs, est la partie intéressante du projet, en plus de la programmation.



3.3 Schéma et afficheur

Le capteur LDR est l'élément supérieur d'un diviseur de tension qui est lu sur une entrée analogique de processeur. Le problème, voir sous <http://www.didel.com/doc/sens/Doclr.pdf> est que la sensibilité est maximale si les résistances de le LDR et du pot sont proches. La tension est alors de ~2.5V et le terminal affiche une valeur proche de 500 (max1023).



A noter qu'il ne faut pas se contenter d'une variation de 10; elle signifie 1% de la valeur, dans la marge des erreurs et fluctuations. Il faut des variations de 100 et plus selon la position de la piste..

Le programme pour afficher la valeur sur le terminal est rappelé ci-contre.

```
void setup()
{
  Serial.begin(9600);
  pinMode(14, INPUT);
}

void loop()
{
  Serial.println(analogRead(14));
  delay(1000);
}
```

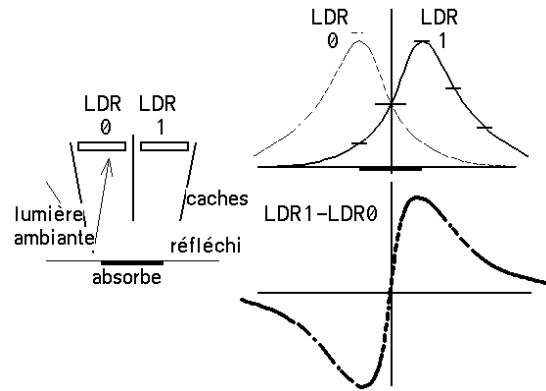
Lorsque le robot est autonome, cette valeur serait fort utile. Elle est remplacée par un affichage approximatif sur deux Leds, l'intensité lumineuse dépendant de la tension envoyée au processeur.

Le schéma a l'avantage d'être compréhensible par un élève. Un transistor bipolaire a un courant collecteur proportionnel au courant de base. On s'arrange pour que ce courant augmente entre 1.5 et 3.5V en mettant une diode et une résistance tenant compte du gain du transistor et de l'efficacité de la Led.

3.4 Algorithme

Si la piste est foncée sur un sol clair, on doit prévoir des caches qui font que les LDR voient la moitié de la piste lorsque le capteur est centré. Un écart augmente la lumière d'un côté et diminue de l'autre. La commande de la direction résulte de la différence des ces deux valeurs.

Le test des algorithmes d'asservissement peut commencer.



4. Aller vers une lumière

Le problème pour viser la lumière est le même que pour le suivi de ligne. Une fonction intéressante peut être ajoutée: s'arrêter à une certaine distance, en tenant compte de l'intensité reçue. À noter que cette intensité dépend du carré de la distance, et que le réglage du potentiomètre est fixe.

5. Procession de robots

Le robot se dirige vers une lumière et adapte sa vitesse pour que l'intensité, donc la distance, soit constante. Le potentiomètre peut être réglé pour cette distance, ce qui améliore la précision tant de l'orientation que de la distance.

Le problème nouveau est l'éclairage à mettre sur le robot qui doit être suivi. Un ampoule 5V 30-50mA à un spectre très large comme les LDR. Une LED verte correspond au maximum de sensibilité de la LDR (560nm) et il existe des Leds vertes d'un excellent rendement.



Cyclopes en 1994, processeur HC11 et caméra linéaire

6. Autres solutions

L'utilisation d'un phototransistor plutôt que la LDR ne change pas l'électronique de lecture, si ce n'est que le potentiomètre doit avoir une valeur plus élevée. L'avantage est une directivité meilleure et le travail en infrarouge, qui demande d'éclairer la piste (dont la réflectivité ne peut pas être estimée par l'œil humain) et est très favorable pour une procession, la lumière ambiante jouant nettement moins de rôle.

Par contre, l'utilisation d'un capteur lumière-fréquence comme le TSL245 a l'avantage d'une dynamique de trois décades que l'on convertit en logarithme pour avoir une valeur qui correspond à notre perception visuelle, et qui fonctionne dans une large gamme d'éclairage ambiant.

Enfin une caméra linéaire de 100 pixels (TSL201) donne précisément le maximum de lumière et sa valeur linéaire.

La mesure par capteur ultrasonique est facile. Un circuit facilitant la programmation est prévu.

7 Etapes suivantes

Le PWM permet le contrôle de la vitesse. Associé à un capteur de distance précis, des mouvements plus souples peuvent être atteints et on peut réfléchir aux algorithmes de réglage.