

DOSSIER PROJET N°4 :

Robot Ramon

Coéquipier n°4 : Guillaume BONNAMOUR

Gestion du déplacement du robot dans le conduit.

SOMMAIRE

I / Conception Préliminaire

- 1.1) Explorer des solutions permettant de mesurer la tension de la batterie et de détecter un seuil batterie trop faible.
- 1.2) Explorer des solutions permettant de détecter la fin du conduit.
- 1.3) Explorer des solutions permettant d'inverser automatiquement le sens de déplacement du robot aux extrémités du conduit.
- 1.4) Explorer des solutions permettant de piloter les roues du robot pour se déplacer dans le conduit.

II / Solutions gardées

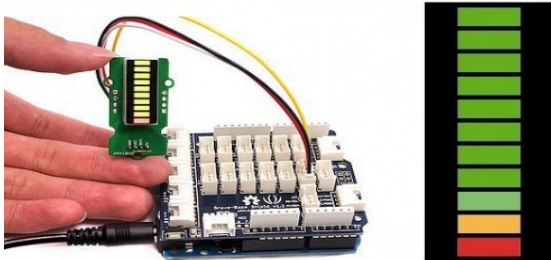
- 1.1) Mesurer la tension de la batterie et de détecter un seuil batterie trop faible.
- 1.2) Détecter la fin du conduit.
- 1.3) Inverser automatiquement le sens de déplacement du robot aux extrémités du conduit.
- 1.4) Piloter les roues du robot pour se déplacer dans le conduit.

III / Conclusion

Notre, projet, le robot ramoneur, est composé de plusieurs parties, toutes assignées à certains membres. La partie que nous avons traitée est celle de la gestion du déplacement du robot dans le conduit, ainsi que la mesure de la tension de la batterie. Voici le devoir, plus en détails :

I / Conception préliminaire

1.1 Explorer des solutions permettant de mesurer la tension de la batterie et de détecter un seuil batterie trop faible.



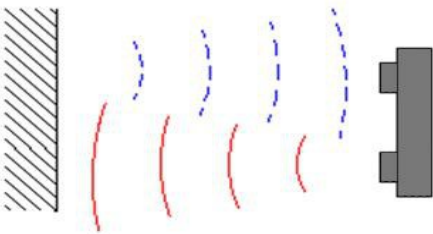
Pour cette partie, nous pensons utiliser un bargraph 10 LEDS pour afficher l'état de la batterie.

Les 10 LEDs ne sont pas toutes de la même couleur, ce qui permet de visualiser plus rapidement l'état de la batterie.

La première LED, celle du bas, représente la tension la plus faible de la batterie. Elle est donc de couleur rouge, pour indiquer un niveau critique. Juste au dessus se trouve une LED orange, servant à prévenir l'utilisateur. Puis vient 8 LEDs vertes, montrant que la batterie délivre encore assez de tension pour alimenter le robot.

1.2 Explorer des solutions permettant de détecter la fin du conduit.

Dans cette partie, nous nous sommes rajouté un but : mettre un capteur à l'arrière. En effet, ce capteur est crucial. Il servira à empêcher la chute du robot dans la cheminée, lorsqu'il redescendra. Etant donné que la cheminée est beaucoup plus large, plusieurs problèmes au niveau du déploiement des roues s'imposent. 1 : La taille. Le robot n'est simplement pas assez large pour rouler dans la cheminée. Les roues ne pourront jamais atteindre les parois. 2 : La vitesse de déploiement des roues n'est pas assez élevée pour que le robot puisse se rattraper aux parois de la cheminée en tombant. Le robot ferait donc une chute et tomberait directement au fond de la cheminée. Le capteur que nous avons ajouté permet d'éviter tout cela. Il fera en sorte que le robot s'arrête avant la fin du conduit.



Tout d'abord, parlons des capteurs. Il existe différents types de capteurs. Nous en avons étudié 4, pour comparer leurs avantages et inconvénients. Le premier était un capteur à Ultrasons. Ce capteur est composé d'un émetteur et d'un récepteur. Son fonctionnement est simple. L'émetteur, envoie des salves de sons inaudibles pour l'Homme, à intervalles réguliers. Le son se propage à 330m/s. Si il ne rencontre aucune paroi, il continue sa route, et dans ce cas, le récepteur ne captera rien. Si, au contraire, un objet se trouve dans son champ, ce dernier renverra les vibrations qui seront captés par le récepteur. Grâce à cela, on peut savoir si un objet se trouve devant et même, en connaissant la vitesse du son, et le temps que la vibration a mis pour revenir, savoir à quelle distance il se trouve.

Ce capteur est très pratique en général, pour, par exemple, connaître la quantité restante d'eau dans une cuve. Malheureusement, dans le conduit, les risques d'écho sont très élevés. Cela pourrait brouiller les signaux et ainsi les informations seraient faussées.

Ensuite, vient le capteur infrarouge. Ce capteur fonctionne exactement de la même manière que le capteur à ultrasons. Il est composé d'un émetteur et d'un récepteur. La seule différence réside en le fait qu'il n'émet pas de son, mais une lumière dite "infrarouge". Elle se trouve dans le spectre invisible. C'est à dire que sa longueur d'onde est supérieure à celle de la lumière visible. L'être humain ne peut donc pas la voir. La lumière, de la même manière que le son, sera renvoyée lorsqu'elle rencontre un objet, et captée par le récepteur. Ainsi, on peut détecter la présence d'un objet. Encore une fois, un problème est très vite rencontré. La suie, qui recouvre les parois, est d'un noir sombre. La couleur noire a la capacité de retenir les rayons de lumières. Le capteur ne fonctionnera pas aussi bien que voulu.

Le troisième capteur est un capteur à contact. Celui-ci est différent des autres, car, comme son nom l'indique, il a besoin d'une pression contre un objet pour détecter celui-ci. Celui étudié possédait une roulette. Deux possibilités d'utilisations s'offraient à nous avec ce capteur. Soit, nous le laissons collé contre une paroi, le laissant ainsi capter la présence du mur, et lorsque le robot arrive tout en haut, il

détecte la fin du mur, soit nous le mettons au dessus, et lorsqu'il touche le chapeau, le robot s'arrête et fait demi-tour. Seulement, toutes les cheminées ne possèdent pas de chapeau, et surtout, la suie se coincerait dans la petite roulette, l'empêchant de bien tourner et pouvant casser la tige. Ce capteur ne peut pas être utilisé pour le bas non plus, à cause de sa trop petite taille. Il a besoin d'un contact direct avec l'objet à détecter. Hors, le fond de la cheminée est bien trop éloigné du début du conduit.

Puis enfin, le dernier capteur est un capteur LDR. Ce capteur, tout comme le capteur infrarouge, détecte la présence de lumière, à la seule différence que cette lumière peut-être vue par l'être humain. En somme, toutes celles se trouvant dans le spectre visible, entre autre la lumière du soleil. Ce capteur serait pratique des deux côtés du robot. Le conduit sera assez sombre pour que le capteur ne détecte rien, jusqu'à l'arrivée à l'un des deux bouts.

1.3 Explorer des solutions permettant d'inverser automatiquement le sens du robot aux extrémités du conduit.

Pour cette partie, le robot doit pouvoir s'arrêter et faire demi-tour lorsqu'il arrive à l'extrémité. Pour cela, les capteurs sont indispensables : le robot sait quand il arrive à la fin. Une fois que le robot sait qu'il a atteint le bout, il doit pouvoir faire demi-tour. Pour cela, nous avons écrit un programme, dont voici l'explication.

Tout d'abord, nous n'avons eu qu'à reprendre le programme qui sert à faire avancer ou reculer le robot, puis je l'ai modifié pour qu'il fonctionne avec les capteurs.

Si le robot est en marche avant, alors c'est le capteur du haut qui détectera la lumière arrivant du bout du conduit. Tant que cette lumière ne dépasse pas un certain seuil, le robot continue sa route. Une fois ce seuil atteint, le programme réagit comme prévu : le robot fait demi-tour. Idem pour lorsque le robot descend, et se trouve en bas. Cette fois-ci, c'est le deuxième capteur, celui du dessous qui entre en jeu. Une fois le seuil de luminosité atteint, le robot fait demi-tour, et remonte.

1.4 Explorer des solutions permettant de piloter les roues du robot pour se déplacer dans le conduit.

Le robot peut nettoyer la cheminée, en étant piloté par deux modes différents :

- Le mode automatique, où le robot avance et recule tout seul, sans que l'utilisateur n'ait à s'en charger
- Le mode manuel, où l'utilisateur décide de quand arrêter et faire changer de sens le robot.

Ici, le robot se trouve dans un conduit. Ses possibilités de direction sont donc réduites. Il ne possède

```
case 'a':           // si on recoit le caractere 'a'
Arret_robot();      // Arrêter le robot
break;

case 'm':           // si on recoit le caractere 'm'
Marche_montee();    // Actionner la montée du robot
break;

case 'd':           // si on recoit le caractere 'd'
Marche_descente();  // Actionner la descente du robot
break;
```

plus qu'un axe : l'axe verticale. C'est à dire que le robot ne peut faire que monter et descendre dans le tuyau. Ensuite, pour que la personne puisse le guider, elle doit disposer d'un pupitre, composé de boutons de direction. 3 boutons sont utiles pour la guider.

Le programme doit répondre aux demandes de l'utilisateur. Ces dernières sont :

- Faire avancer le robot lorsque l'utilisateur appuie sur la touche "m"
- Faire reculer le robot lorsque l'utilisateur appuie sur la touche "d"
- Faire arrêter le robot lorsque l'utilisateur appuie sur la touche "a"

Pour cela, le programme doit lire l'octet saisi, c'est à dire le bouton sur lequel l'utilisateur appuie. Une fois cet octet lu, l'information est traitée et l'ordre correspondant est envoyé aux moteurs.

II / Solutions gardées

1.1 Mesurer la tension de la batterie et de détecter un seuil batterie trop faible.

Pour cette partie, la seule solution imaginée a été gardée. Elle permet une visualisation rapide et précise du niveau de la batterie

Pour faire en sorte que les LEDs s'allument à un niveau exact, nous avons rédigé un programme, nous aidant de programmes tests existant déjà. Nous avons établi différents paliers, pour chaque LEDs. Si la valeur reçue se trouve dans une des plages, toutes les LEDs en dessous s'allumeront. Par exemple, si la batterie est à 750 mV, la 6ème LED s'allumera, ainsi que toutes les LEDs en dessous d'elle.

Les plages ont été très simplement établies : la LED rouge correspond à une valeur comprise entre 620 et 610 mV, la LED orange, à une valeur comprise entre 640 et 620 mV, ainsi de suite jusqu'à la dernière LED, verte qui correspond à une plage de 840 mV à 810 mV. Le bargraph affichera donc l'état de la batterie.

Un problème s'est imposé : la tension délivrée par la batterie est bien supérieur à celle que peut supporter la carte Arduino à laquelle le bargraph est relié. En effet, la batterie a une tension maximale de 8,4 V, alors que la carte ne peut en supporter que 5. Pour cela, nous avons utilisé un système appelé Pont diviseur de tension. Ce système consiste à réduire une tension, en la faisant passer dans des résistances. La tension ainsi réduite, sera adaptée à l'usage qui lui sera réservé. Ici, un calcul a été réalisé pour connaître la valeur des résistances à utiliser.

$$V_{imgb} = (R2 * V_{max}) / (R1 + R2)$$

Vmax étant la tension maximale de la batterie, et Vimgb la tension maximale que peut recevoir la carte Arduino.

Nous obtenons donc 3 valeurs différentes : $R2 = 15K\Omega$ et $R1 = 10K\Omega + 12K\Omega$.

Nous avons donc nos trois valeurs : 15k Ω , 10K Ω ET 12K Ω . Maintenant que nous avons les valeurs des résistances, la carte arduino peut-être reliée à la batterie sans être endommagée.

1.2 Détecter la fin du conduit.

Pour cette partie, nous avons choisie les capteur LDR. Ils sont beaucoup plus adaptés aux contraintes qu'imposent le conduit et la suie présente à l'intérieur. Les capteurs ne peuvent pas être salis, puisque celui du haut se trouve bien au dessus de la brosse, et celui du bas est orienté de telle façon que la suie ne peut pas lui tomber dessus. Ainsi, ils capteront la lumière tout au long du nettoyage.

De plus, contrairement au capteur infrarouge, la lumière viendra directement de l'extérieur. Le capteur n'aura pas besoin que celle-ci soit réfléchi sur les parois. Donc elle sera bien moins absorbée à cause de la couleur sombre de la suie.

Ensuite, le capteur LDR possède un inconvénient en moins par rapport au capteur à contact. Ce dernier nécessite un toit au dessus de la cheminée, pour pouvoir être actionné et ainsi le détecter. Or, toutes les cheminées n'en possèdent pas une. Ce capteur aurait aussi pu être utilisé contre la paroi directement, comme expliqué en haut, la suie aurait rapidement bloquée la roulette, et l'aurait endommagée au bout de plusieurs minutes. Le capteur LDR n'a besoin d'aucun contact, il est donc protégé des problèmes des capteurs à contact.

1.3 Inverser automatiquement le sens du robot aux extrémités du conduit.

Pour cette partie, la solution retenue est celle qui a été pensée au tout début. Le robot, grâce aux capteurs saura quand s'arrêter, et l'information sera traitée par le programme qui inversera le sens de rotation des moteurs.

1.4 Piloter les roues du robot pour se déplacer dans le conduit.

La solution gardée ici est la même que l'idée pensée du début. Le programme écrit permet de choisir le sens de rotation des roues, la direction du robot donc, ainsi que de l'arrêter quand l'utilisateur le veut.

III / Conclusion

Grâce à cette partie, le robot sera tout à fait capable de se déplacer dans le conduit, faire marche avant et marche arrière manuellement, ou automatiquement, ainsi que de détecter lorsqu'il arrive à une des deux extrémités. Merci aux ITEC, pour la fabrication des différentes parties du robot.