Rapport du projet de système embarqués et robotique

Photo

* Introduction

Dans le cadre de ce projet, nous avons essayé de mettre en place le logiciel d’un robot aspirateur avec les contraintes que vous avions sur l’E-puck2. Nous avons donc implémenté cette fonction en quelques étapes. La premère étant le nettoyage de la pièce. Lorsque le nettoyage est fini, nous avons une phase de recherche de la station de recharge représentée par une ligne noire sur le mur. Pour terminer le robot se gare devant cette ligne noire pour simuler la charge.

* Principe de fonctionnement
* Déplacement dans l’environnement

Lors de la phase de nettoyage, le robot navigue aléatoirement dans une pièce en évitant les murs. Pour cela nous utilisons les 8 capteurs de proximité disponibles sur le contour du robot afin de controller la vitesse des moteurs pour avancer et tourner. Le fonctionnement est assez simple, en effet il suffit que les deux moteurs tournent dans des sens opposés lorsque le mur est détecté à l'avant droit ou gauche du robot. Dans le cas où le robot arrive de face à un mur (donc le mur est détecté par les deux capteurs à l'avant du robot) il tourne sur lui-même pendant un certain temps avec une vitesse différente à chaque fois pour que les déplacements soient aléatoires. Il y a un dernier cas dans lequel le robot avancerait avec une trajectoire quasi-parrallèle à un mur et qu'il s'en approche petit à petit. Cette situation est à éviter et donc si un mur est trop proche d'un des côtés du robot, ce dernier corrige sa trajectoire pour ne pas foncer dans le mur.

* Recherche de la ligne

Après un certain temps de nettoyage (par exemple si la batterie est bientot vide) le robot doit retourner à sa station de charge représentée par une ligne noire verticale contre un des murs. Il faut tout d'abord que le robot trouve la ligne, pour cela nous utilisons la caméra. Le robot tourne assez lentement sur lui-même pendant qu'un thread d'acquisition de l'image et un thread de traitement de l'image tournent. Ces deux threads permettent de déterminer si une ligne de largeur suffisante se trouve dans le champ de vision du robot. Une fois la ligne trouvée le robot commence la phase d'approche. Si le robot ne trouve pas de ligne après un tour sur lui-même il revient à une phase de nettoyage avec déplacements aléatoires pour un cetain temps.

* Phase d'approche de la station
* Organisation du code
* Organisation générale
* FSM
* Interaction entre les threads
* Résultats
* Conclusion
* Référence