

RAPPORT TECHNIQUE

I - Introduction

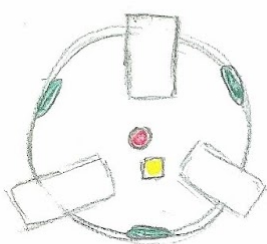
1) L'objectif

L'objectif de notre projet est de créer une aide aux guides de musées robotisée: En nous aidant d'un robot HOLONOME, d'un lecteur mp3 et d'un capteur de distance, le but est de créer un dispositif capable de suivre un parcours donné, de lire un fichier audio une fois arrivé devant un tableau, et de s'arrêter s'il est confronté à un obstacle.

2) Notice d'utilisation

L'aide d'un guide de musée est nécessaire au préalable pour enregistrer les différents messages audios de description des tableaux souhaités. Une fois le trajet de la visite entré dans le code de pilotage du robot, il est recommandé de baliser le trajet dans le musée pour éviter les obstacles au maximum.

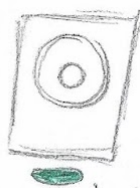
Il suffit alors de lancer le démarrage du robot: celui-ci avancera de salle en salle tout en décrivant les tableaux souhaités sur son chemin. Si une personne lui bloque la route, elle sera gentiment priée de se décaler par celui-ci.



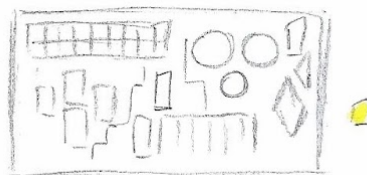
un moteur par roue,
programmé pour avancer
dans 4 directions.



Un haut-parleur programmé pour
parler à des endroits précis (devant
les tableaux), ou en cas d'obstacle



3 capteurs infra-rouges
détectant les obstacles.



Carte Nucléo + Carte SD (mémoire
fichier audio et trajet).

I- Cahier des charges

| | | |
|---|--|----------------|
| Encombrement | Forme du robot | Circulaire |
| | Poids | 1 kg |
| | Dimensions | 13 cm de rayon |
| Vitesse du robot | 0.08 m/s | |
| Distance de détection | 90 cm | |
| Temps d'arrêt après disparition de l'obstacle | 5 sec | |
| Usage | Aide à la visite de musées | |
| Design | Sobre et élégant | |
| Alimentation | Batterie | |
| Nombre de moteurs / roues | 3 | |
| Schéma de déplacement | Parcours prédéfini, balisé au préalable | |
| Type de déplacements | Translation : gauche, droite, haut, bas | |
| Système pour véhiculer l'information | Haut-parleur, lecture de fichiers audio préenregistrés | |
| Tension d'alimentation par moteur | Entre 6 et 12 V | |
| Volume des messages audio | ~ 60 dB | |

II- Description des fonctionnalités souhaitées

- Avancer de manière autonome dans 4 directions
- Suivre fidèlement un trajet programmé
- Stopper la rotation des roues lors de la rencontre d'un obstacle à moins de 90 cm
- Avoir une distance d'arrêt inférieure à 90 cm
- Pouvoir lire un fichier audio (mp3 ou autre)
- Restituer le message audio à l'aide du haut-parleur
- Prévenir poliment la personne située sur son passage
- Après l'arrêt devant un obstacle, reprendre le trajet initial là où il s'était arrêté

Fonctions réalisées au final :

Avancer de manière autonome dans 4 directions :

Le robot doit être capable de se déplacer dans toutes les directions. Cela était réalisable grâce aux trois roues motrices commandées indépendamment. En modifiant le rapport

cyclique de chacune des roues, à l'aide de l'inertie, on peut contrôler la vitesse de rotation de celles-ci, et ainsi par un simple calcul de mécanique, choisir la direction de déplacement.

Nous avons utilisé la fonction "write" de *mbed.h* pour contrôler ce rapport cyclique.

Pour vérifier le bon fonctionnement de notre code, nous avons branché une sonde à l'oscilloscope et constaté le créneau qui apparaissait aux bornes du moteur. Nous avons fait plusieurs essais avec différents rapports cycliques pour observer à l'oeil nu les effets sur la vitesse du moteur.

Suivre fidèlement un trajet programmé :

Le robot doit être capable de savoir où il se trouve afin de permettre une programmation du trajet. Pour cela, on utilise un codeur placé sur chaque moteur et on compte les impulsions fournies par celui-ci. On peut ainsi prévoir un certain nombre de tours de roues et programmer un déplacement. Nous avons compté les tours de roues pour vérifier qu'il correspondait bien à celui prévu

Stopper les roues en cas de présence d'un obstacle :

On utilise pour cela un capteur de distance. En cas de présence d'un objet à moins de 1.20m environ, la tension à ses bornes augmente (tension mesurée à l'oscillo pour déterminer expérimentalement la tension seuil). On utilise donc une fonction d'interruption qui vient vérifier que la tension aux bornes du capteur infrarouge n'a pas dépassé une tension seuil, et, si c'est le cas, on arrête tous les moteurs. L'utilisation d'une fonction d'interruption permet également de remplir la fonction "**reprenre le trajet initial là où il s'était arrêté après un obstacle**". Le bon fonctionnement du système (capteur et code) a été testé en "visuel".

Lire les fichiers audio :

Le fichier lu est placé sur une carte SD, et la carte Nucléo lit le flux audio quand on le lui demande, en fonction du code global (soit dans le main, soit dans la fonction d'interruption). Les tests ont d'abord été réalisés à part, en appuyant sur un bouton, puis en programmant la lecture autonome du fichier, et enfin en intégrant le montage au code réalisé contrôlant les moteurs. Cela correspondait à l'achèvement de notre projet.

Compte-rendu des séances :

Séance 1 : Nous avons choisi notre projet et développé l'univers autour de celui-ci. Les calculs de vitesse des roues du robot ont été effectués ainsi que la compréhension du fonctionnement de la mobilité du robot.

Séance 2 : Nous avons réparti les différentes tâches à faire : Clémence et Adam s'occupent de programmer le moteur, le déplacement et les arrêts, tandis que Clara et Nicolas s'occupent de la partie liée au haut-parleur et aux discours du robot.

Nous avons imaginé un dispositif permettant au robot de reprendre sa course là où il s'était arrêté après la détection d'un obstacle, à l'aide d'un compteur d'impulsions.

Séance 3 : Nous avons étudié le fonctionnement du capteur, et choisi une tension seuil signifiant l'arrêt du robot dans le code. Le capteur a été testé, branché, il était fonctionnel à la fin de la séance. Clémence et Adam ont cherché la tension seuil correspondant à un obstacle situé à 1m pour la suite du projet. Un code fonctionnel a été réalisé afin d'allumer une LED lorsqu'un objet s'approchait à moins d'un mètre.

Clara et Nicolas ont enregistré les différents messages audio (4 au total) et ont changé leur format pour les mettre dans la carte SD à notre disposition.

Séance 4 : Clara et Nicolas ont récupéré les codes créés pour nous par Julien Villemejeane pour connecter le haut-parleur et les avons copiés sur Keil studio. Une fois les branchements effectués, nous n'entendions pas de son en sortie. Avec l'aide de Fabienne Bernard, nous avons cherché où était l'erreur. Après avoir rajouté plusieurs fonctions « printf », nous avons pu localiser la boucle qui posait problème grâce au logiciel Teraterm.

De leur côté, Adam et Clémence se sont attelés à la programmation du parcours avec le robot. La première étape était de réaliser une sorte de cartographie du robot en identifiant les moteurs et les branchements associés à chaque roue. En fin de séance, le robot était en mesure d'avancer et un programme potentiel du parcours.

Séance 5 : Clara et Nicolas ont continué à essayer de faire marcher le haut-parleur : nous avons branché un amplificateur LM386 pour entendre quelque chose sans succès : le problème vient du codage. Nous sommes cependant très reconnaissants envers Gabriel Granveau qui a passé plus d'une heure à nous aider.

Adam et Clémence ont tenté de mettre en forme le programme pour que le robot réalise le parcours. Après plusieurs échecs liés à des défaillances du matériel, nous avons codé le capteur pour qu'il déclenche une interruption du programme. Même en simplifiant le programme au maximum, les roues ne suivaient pas les consignes, il nous a fallu longtemps avant d'identifier le matériel défaillant qui bloquait le bon fonctionnement du système.

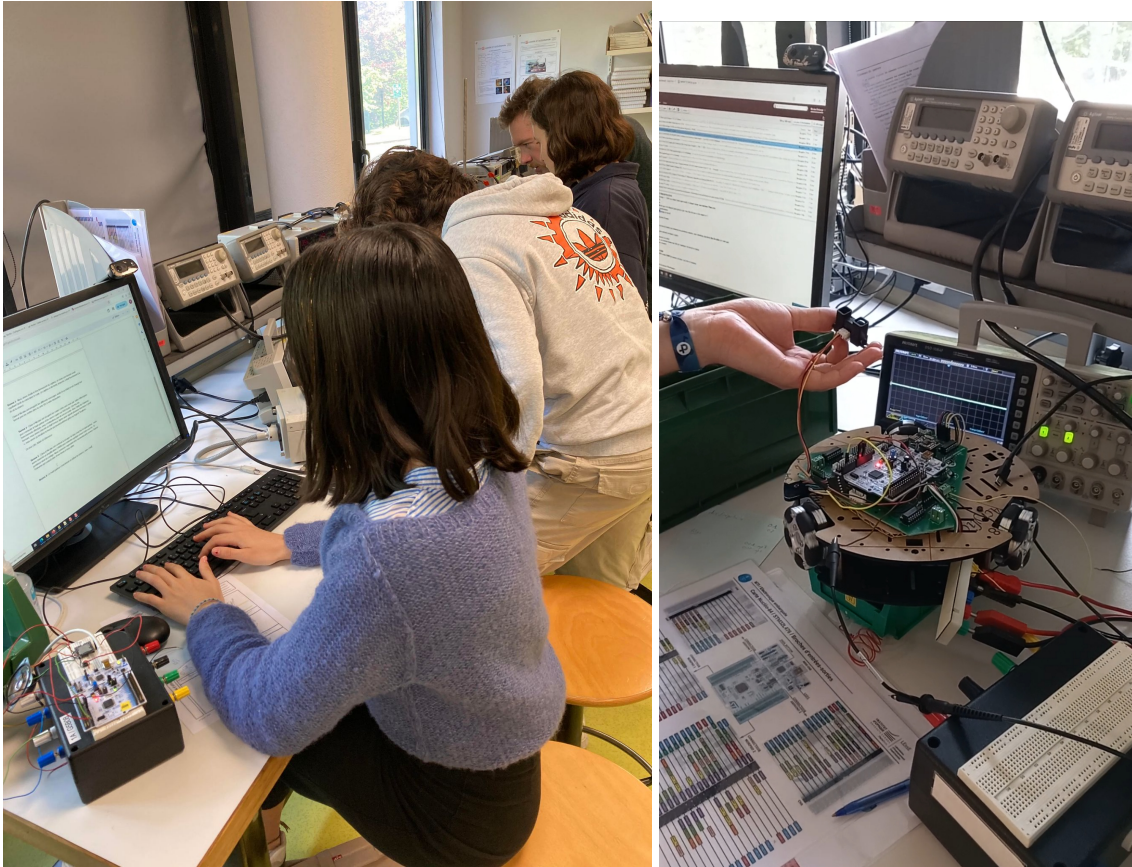
Séance 6 : Le haut-parleur a fonctionné en début de séance. Un bouton permettant de passer à la piste suivante a ensuite été câblé. Clara et Nicolas ont donc pu s'intéresser à la fonction qui permet de jouer un titre choisi parmi les répertoires présents sur la carte SD.

Adam et Clémence ont décidé de simplifier l'objectif du parcours car la mise en place d'un asservissement avec interruption compliquait la tâche. Ils ont décidé de se restreindre à une ligne droite au cours de laquelle le robot s'arrête s'il rencontre un obstacle, selon une unique fonction d'interruption. La bibliothèque QEI ayant assez mal fonctionné, ils ont, à l'aide de M. Villemejeane, remplacé l'ancien code par un programme comptant les impulsions du capteurs suivant le principe des fronts montants.

Bilan :

En conclusion, ce projet nous a permis d'apprendre à coder davantage en C++, ainsi qu'à manier les outils électroniques de manière autonome en mettant à profit nos connaissances

dans la matière. Si on excepte le fait que notre parcours ait été transformé en simple ligne droite puisqu'une roue n'était pas fonctionnelle, notre objectif était globalement atteint. Nous sommes fiers d'avoir pu terminer ce projet et prêts à commercialiser d'ici peu.



Petites photos prises pendant notre travail :