Projet SPI3A : Rapport

C.Doublet
B.Gouneaud
C.Diehl
Y.Dusoleil

Professeur : A.Houzé

L2 STS IE3-03
2021/2022

Table des matières

Introduction

Ce rapport fait l'objet du rendu final du projet de SPI3A. Le but de ce projet est de faire avancer un « Arduino Robot », d'afficher la distance parcourue par celui-ci sur son écran intégré, de réaliser et d'afficher des mesures d'humidité et de température sur ce même écran.

Lors de la réalisation de ce projet nous avons été confrontés à plusieurs problèmes qui ont conduit à un aménagement du sujet initial par Mr Houzé. Les problèmes rencontrés et l'aménagement du sujet seront détaillés dans la première partie de ce rapport. Dans un second temps nous expliquerons comment nous avons implémenté ce nouveau sujet.

Avant de conclure nous expliquerons comment nous aurions pu réaliser le sujet initial si nous n'avions pas rencontrés les problèmes expliqués dans la première partie.

I Aménagement du sujet

I.1 Problèmes rencontré

Comme dit dans l'introduction, nous avons rencontrés plusieurs problèmes qui ont induit un aménagement du sujet. Le premier problème est lié à l'import de la librairie ArduinoRobot qui ne se faisait pas de manière attendue. Une fois ce problème réglé nous avons constaté que notre robot ne fonctionnait pas. Heureusement, nous avions un deuxième robot qui lui, était fonctionnel. Nous avons eu un autre problème certainement dû à la calibration des roues du robot. Ce qui ne nous a pas permit d'effectuer le calcul de la distance parcourue par le robot (notre méthode de calcul de cette distance est expliquée dans la dernière partie de ce rapport). En effet, la mauvaise calibration des roues du robot ne permettait pas de faire un déplacement rectiligne et donc nous ne pouvions pas appliquer notre méthode de calcul de distance. Finalement, tous ces problèmes nous ont fait perdre un temps considérable lors de la réalisation du projet.

I.2 Nouveau sujet

Les problèmes que nous avons rencontrés, nous ont amenés à une diminution considérable du temps de réalisation de ce projet. De ce fait Mr Houzé, s'est permit d'aménager les objectifs du projet. Les nouveaux objectifs sont, l'affichage d'une image sur l'écran LCD du robot, le déplacement de celui-ci, la modification de la vitesse du robot à l'aide du potentiomètre intégré et pour finir l'affichage de cette valeur de vitesse sur l'écran LCD.

II Réalisation du nouveau projet

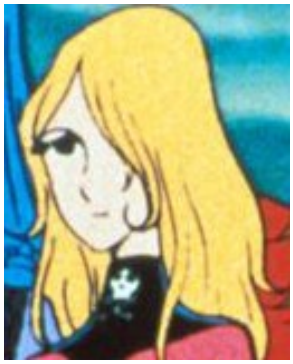
Pour réaliser ce projet nous avons dans un premier temps déclaré et initialisé deux variables. La première variable « deplacement » est initialisé à false elle vaudra true lorsque le robot sera en déplacement. La seconde variable « relevePot » est initialisée à 0 et permettra de stocker la valeur de vitesse de notre robot définie par notre potentiomètre.

Dans la fonction setup(), nous initialisions le robot, l'afficheur et le port SD permettant d'accéder à l'image que nous voulons afficher. Ensuite, toujours dans le setup nous affichons l'image albator2.bmp pendant 5 secondes.

albator2.bmp

Dans la fonction loop(), si la variable « deplacement » vaut false, nous arrêtons le déplacement du robot. Nous récupérons ensuite la valeur du potentiomètre servant à régler la vitesse du robot. Nous calculons la vitesse du robot à partir de la valeur du potentiomètre entre 0 et 1023 avec la fonction map(). La vitesse est une valeur entière entre -255 et 255, si la valeur est négative, le robot recule, sinon il avance.

Afin d'afficher la vitesse sur l'afficheur LCD, nous définissons la couleur du texte avec la couleur noir grâce à la fonction stroke() puis nous affichons la vitesse à la coordonnée (0, 0) de l'afficheur avec la fonction text(). Après un délai de 32 millisecondes, la vitesse est effacée en réécrivant la même valeur en blanc et à la même coordonnée. Pour terminer, si le bouton du milieu intégré au robot est appuyé. Nous passons la valeur de « deplacement » à true et nous affichons l'image albator.bmp.



albator.bmp

Si « deplacement » vaut true. Nous activons le déplacement de notre robot avec la fonction motorsWrite() avec en paramètre la vitesse du robot pour le moteur gauche et droit. Si le bouton du milieu intégré au robot est appuyé, « deplacement » est mis à false et l'affichage est effacé après un délai de 10 millisecondes.

III Réalisation du projet initial

Comme expliqué dans l'introduction, la réalisation du projet initial devait permettre à notre robot de se déplacer, de réaliser et d'afficher des mesures de température et d'humidité ainsi que d'afficher sa distance parcourue. Nous avons détaillé comment fonctionne le déplacement de notre robot dans la partie précédente ainsi que l'affichage sur l'écran intégré. C'est pourquoi, dans cette partie, nous ne détaillerons que la façon dont nous aurions procédé pour réaliser les mesures d'humidité et de température ainsi que le calcul de la distance parcourue.

III.1 Mesures d'humidité et de température

Les mesures de température et d'humidité s'effectuent à l'aide d'un capteur DHT11, avec l'importation de la librairie DHT. La définition de la broche du capteur s'effectue avec l'instruction #define DHTPIN 5. La définition du type de capteur utilisé est faite avec l'instruction #define DHTTYPE DHT11. Ensuite, l'objet DHT qui permet d'effectuer les mesures est créé à l'aide de son constructeur dht() avec DHTPIN et DHTTYPE en paramètres. Ce capteur est initialisé dans la fonction setup avec la méthode begin(). Pour terminer il suffit de lire les valeurs de température et d'humidité avec les méthodes readTemperature() et readHumidity().

III.2 Calcul de la distance parcourue

Comme expliqué dans la partie précédente, la valeur de vitesse de notre robot est située entre -255 et 255. Pour calculer la distance parcourue par le robot nous avons eu l'idée de déterminer la distance parcourue par celui-ci en ligne droite pour un pas de vitesse (pour la valeur de vitesse à 1 ou -1). Pour se faire, nous avons pensé à faire avancer le robot pendant 1 seconde à sa vitesse maximale. Ce qui nous aurait permis de mesurer sa distance parcourue en centimètres en 1 seconde. En divisant cette valeur par 1000 nous obtenons alors la distance que parcours le robot en 1 millisecondes à sa vitesse maximale. Il nous suffit alors de diviser cette valeur par 255 ce qui nous permet d'obtenir la distance parcourue par notre robot en 1 milliseconde à sa vitesse minimale.

Connaissant cette valeur constante. Pour calculer la distance parcourue en centimètre par notre robot, il aurait suffi de multiplier cette valeur constante par le délai en millisecondes d'un tour de boucle et de multiplier cette valeur obtenue par la valeur absolue de la vitesse du robot et d'ajouter le résultat à la distance totale parcourue.

Comme expliqué dans la première partie du rapport, nous n'avons malheureusement pas pu mettre en pratique cette méthode à cause de la mauvaise calibration des roues du robot. En effet celui-ci n'avance pas en ligne droite, nous ne pouvons donc pas calculer la distance qu'il parcourt en 1 seconde à sa vitesse maximale.

Conclusion

Afin de conclure la rédaction de ce rapport et la fin de ce projet, nous aimerions faire part de ce que nous a apporté ce projet dans sa conception. Tout d'abord, la réalisation de ce projet nous a permis d'apprendre à travailler efficacement en groupe. D'apprendre à gérer la réalisation d'un projet et à se répartir les tâches de production entre nous. Ce projet nous a aussi permis d'apprendre à nous adapter à la situation et aux différents problèmes rencontrés lors de la réalisation d'un projet. Nous souhaitons finalement remercier Be-Bop pour sa coopération.