

RAPPORT TECHNIQUE

ROBOT VÉRONICA - EN ROUTE POUR MARS

Projet PROTIS 2023

Yassir AMAZOUZ, Camille BASA, Aglaé BRIDONNEAU, Alexandre MERSCH

1 Introduction

Le but de ce projet est de développer un robot guidé à distance permettant d'explorer le sol particulier de Mars, pour détecter la présence d'eau...

Les principaux objectifs du robot retenus sont :

- Se déplacer en ligne droite et effectuer des rotations sur lui-même de manière autonome.
- Relever des données de température et d'humidité.
- Transmettre ces données à la base située sur Terre.

De plus, les données envoyées sur Terre devront être affichées de manière lisible et compréhensible pour tout le monde.

Les liens entre les différents acteurs du projet sont décrits sur le schéma de principe Figure 1.

Les contraintes et performances attendues sont les suivantes :

- **Suivre une direction donnée** : le robot doit pouvoir être commandé en direction. Sur la position, l'erreur maximal est de 2 cm, et sur l'angle, une erreur maximale de 5° à 10° est tolérée.
- **Avancer** : le robot doit pouvoir être commandé en vitesse. Il devra avoir une vitesse comprise entre 10 cm/s et 30 cm/s.
- **Être autonome** : le robot devra être sur batterie et devra tenir sur 1 km, soit une autonomie d'environ 2h45. Une erreur de +/- 10 min est acceptée.
- **Capter la température et l'humidité** : Le robot doit pouvoir relever la température et l'humidité toutes les secondes.
- **Afficher les données** : L'interface communiquant avec le robot doit pouvoir afficher une courbe de la température en fonction du temps (et si possible de la position). Il faudra pour cela recevoir les données par communication sans fil.

Les principales fonctions sont résumées dans le schéma fonctionnel Figure 2.

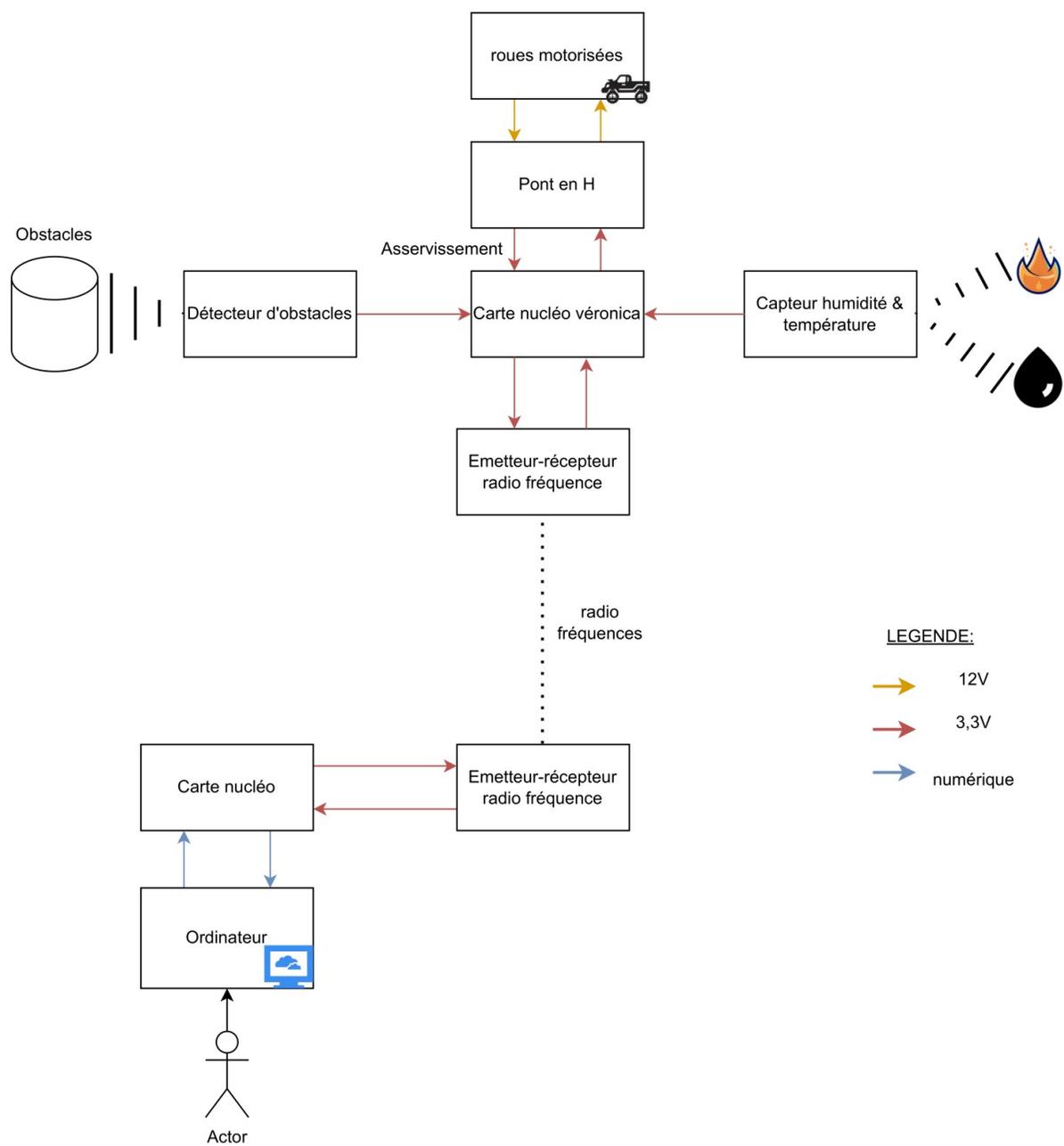


FIGURE 1 – Schéma de principe du fonctionnement du robot

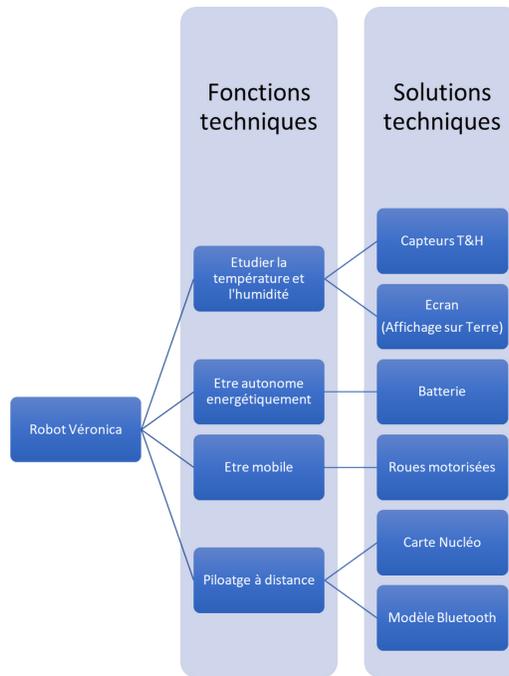


FIGURE 2 – Schéma fonctionnel du robot

2 Description

2.1 Être mobile

L'objectif de cette fonction est de permettre à la voiture de se déplacer d'un point A à un point B. La voiture pourra ainsi relever des mesures au point B ou sur le trajet entre A et B. La voiture est dotée de deux roues motorisées indépendantes qui suivent le modèle de voiture autonome et d'une roue folle. La fonction mobilité est donc étroitement liée aux commandes envoyées aux deux roues. Il s'agit notamment de la donnée de la direction et du rapport cyclique, en plus de l'alimentation. Une première étape a été de le déplacement engendré par une commande envoyé à la roue. Nous avons décidé d'asservir la roue en position à l'aide des capteurs présents sur les roues. Ainsi la distance à parcourir est convertie en un nombre d'encoches à passer et on adapte la vitesse de la roue en fonction du nombre d'encoches à parcourir. On utilise pour cela une fonction qui définit le rapport cyclique de la roue en fonction de la distance parcourue et la distance à parcourir :

$$rc = 1 + k - \exp \left[\frac{1 - dist_{restante}}{(dist_{àparcourir})^{(4dist_{àparcourir}/1000)\log(1+k)}} \right]$$

Avec $dist_{restante}$ le nombre d'encoches qu'il reste à passer ; $dist_{àparcourir}$ le nombre d'encoches à parcourir initialement et k le rc maximal voulu pour la voiture.

Une fois la notion de distance instaurée, il reste à définir les deux fonctions principales liées à la mobilité : avancer et tourner.

- Pour avancer on fait tourner les deux roues en même temps à la même vitesse. Mais comme les deux roues sont indépendantes, il est nécessaire d'asservir une des roues sur l'autre. On applique la commande pour asservir en position la roue gauche. Et une autre commande force la roue droite à suivre l'allure de la roue gauche. Notamment la roue droite accélère si elle est en retard et ralentit si elle est en avance au nombre d'encoches parcourues.
- Pour tourner, on fait tourner qu'une seule des deux roues. L'angle de la rotation dépend de la distance parcourue par la roue qui tourne. On associe un nombre d'encoches à la réalisation d'un tour. On

rajoute un offset à cette valeur, pour prendre en compte le fait que si le rc est trop faible, le moteur ne tourne pas même s'il lui reste des encoches à passer.

On remarquera que l'asservissement ne fonctionne bien que lorsque la vitesse de la voiture varie lentement. Ainsi on applique des gains différents lors du freinage de la voiture.

2.2 Être autonome

Afin que le robot puisse se déplacer seul sans source d'énergie fixe (ex : GBF), il lui fallait une source d'énergie qu'il puisse transporter avec lui. Cette fonction là a été réalisée assez rapidement. Nous utilisons donc une batterie de 12 V maximum. Pour utiliser la batterie, il faut se placer en mode E5V.

2.3 Capter la température

L'objectif de cette fonction étant de capter la température et l'humidité, nous avons choisi d'utiliser le capteur *TempHum 14 Click*. Ce capteur fonctionne sur le protocole I2C. Pour faire fonctionner ce capteur, nous avons eu besoin d'utiliser une nouvelle bibliothèque nommée `TEMPHUM_14_CLICK`.

Nous avons besoin de l'alimenter à l'aide de la carte NUCLEO-L476RG à 3,3 V. Les broches utilisées sont la broche SDA reliée à l'entrée A4 de la nucléo, et la broche SCL reliée à l'entrée A5. Ces choix de broches ont été faits en fonction des broches compatibles avec le protocole I2C, ainsi qu'en fonction des broches déjà utilisées par les moteurs du robot. En effet, les broches de base utilisées, D14 et D15, étaient déjà occupées. Il a donc fallu souder les broches PC1 et PC0 pour pouvoir les utiliser avec A4 et A5 en I2C (cf. Figure 3).

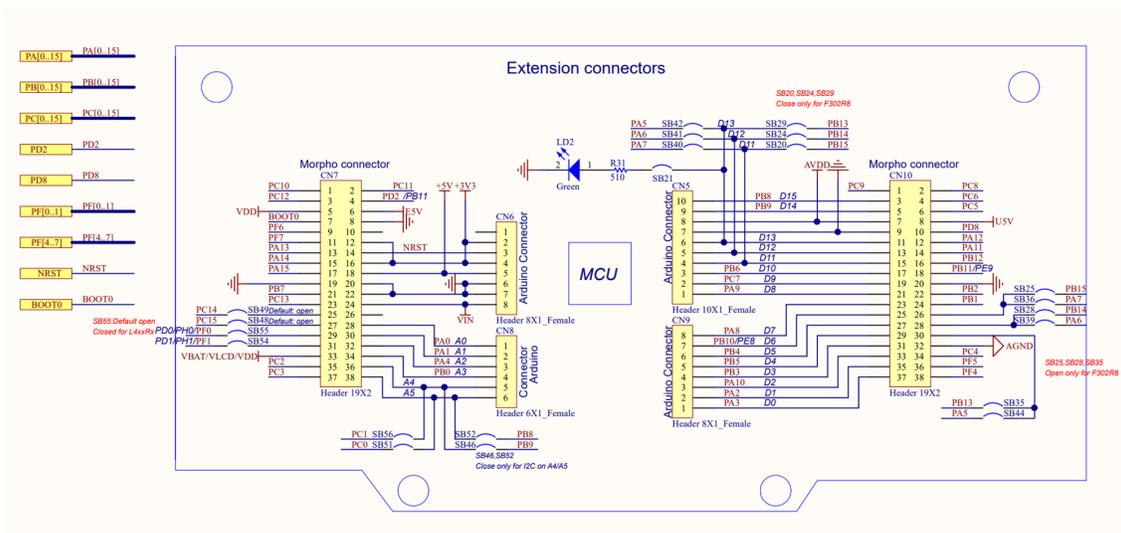


FIGURE 3 – Schéma des connecteurs de la carte NUCLEO-L476RG

Les tests effectués pour vérifier le bon fonctionnement du capteur ont été réalisés sur le logiciel TeraTerm. Les températures étaient affichées en degrés et l'humidité relative en pourcentage. On pouvait ainsi vérifier que le capteur donnait bien des valeurs cohérentes, comme par exemple une température de l'ordre de la température ambiante et qui augmentait si on posait un doigt sur le capteur. La conversion de la température en degré et celle de l'humidité relative en pourcentage est donnée par la datasheet du capteur, et peut ainsi être exploitée dans le code avec $T(^{\circ}C) = -40 + 165 \times \frac{S_T}{2^{16}-1}$ et $H(\%) = 100 \times \frac{S_H}{2^{16}-1}$, avec S la sortie en décimal.

2.4 Communiquer à distance et afficher les données

Nous voulons maintenant transférer les données récupérées par le capteur sur l'ordinateur, pour cela on utilise deux émetteurs/récepteurs qui sont placés sur la voiture et sur une carte reliée à l'ordinateur. Ce sont des émetteurs/récepteurs NRF24L01 qui fonctionnent par bluetooth.

On a donc créé deux algorithmes "envoie" et "recevoir", afin de pouvoir envoyer les données sous la bonne forme depuis le robot et les recevoir de la bonne façon "sur Terre". Afin de suivre l'évolution de la température et de l'humidité nous avons récupéré les données à l'aide de la deuxième carte qui est connectée sur l'ordinateur et de python (avec la bibliothèque "Serial"). Les données reçues sont sous forme de bit qu'on transforme en chaîne de caractère. Ensuite on récupère de cette chaîne que la valeur de la température afin de pouvoir l'injecter dans une liste de nombre réel. (Exemple : on reçoit une chaîne de caractère de la forme suivante "la température est T=24" on prend que le caractère "24" qu'on transforme en nombre réel).

Après un temps donné, on arrête cette opération pour pouvoir traiter les données (10 minutes dans notre cahier de charge), puis on représente tout cela dans un graphique grâce à la bibliothèque matplotlib.

De même qu'avec le capteur, les premiers tests ont été réalisés avec TeraTerm afin de savoir si la carte Nucléo représentant l'interface "sur Terre" recevait bien les données sous la bonne forme. Après validation seulement, nous sommes passés sur Python pour pouvoir tracer la courbe voulue. Les tests ont été très concluants comme on peut l'observer Figure 4 sur un temps de 30 secondes. L'augmentation de température correspond au moment où l'on a posé un doigt sur le capteur, et la diminution correspond au moment où l'on a enlevé ce même doigt du capteur.

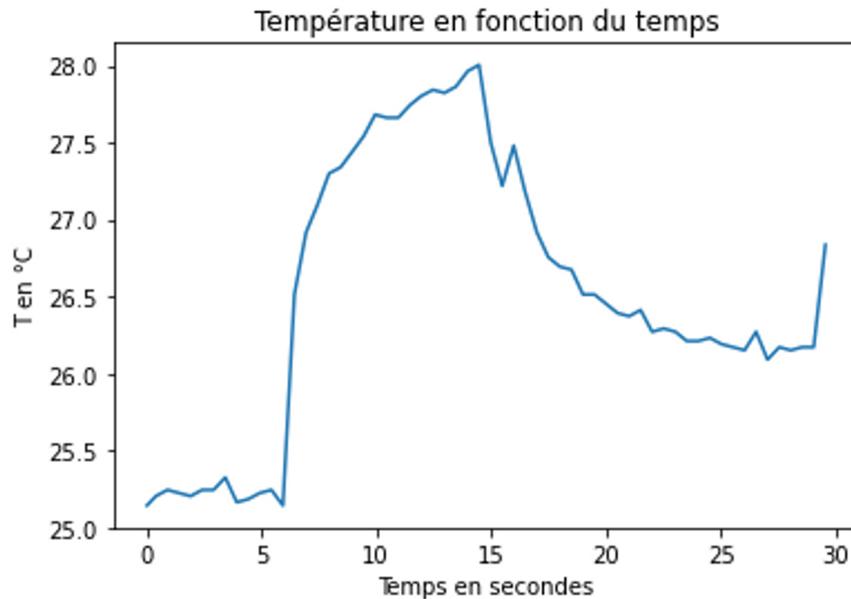


FIGURE 4 – Courbe réalisée sur Python de la température relevée par le capteur et envoyé par bluetooth au récepteur, en fonction du temps

3 Bilan

3.1 Bilan technique

Le bilan de ce projet est à nos yeux très positif.

Tout d'abord, le capteur de température et la transmission de données fonctionnent très bien, de même que l'affichage de la courbe. Ensuite le déplacement du robot est bien contrôlé. Grâce à l'asservissement qui marche très bien, le robot roule droit. De plus, la réunion des codes des deux parties du projet a bien été réalisée. Le robot peut rouler en même temps que de capter la température et de l'envoyer sur l'ordinateur.

Les points que l'on pourrait améliorer sont dans un premier temps la partie sur le capteur d'humidité. Nous avons vérifié que nous captions bien les bonnes valeurs d'humidité mais nous n'avons pas encore coder la partie sur l'envoi des données par bluetooth. Dans un deuxième temps, nous aurions pu améliorer si nous avions eu plus de temps la partie déplacement du robot. Nous aurions voulu créer un code qui puisse envoyer les coordonnées d'une position directement au robot par bluetooth. Ainsi, il aurait été possible de changer la destination du robot en cours de route en fonction des données de température reçues et analysées. Pour l'instant, le trajet de notre robot doit être défini à l'avance (c'est-à-dire avant son "envoi sur Mars").

3.2 Bilan humain

D'un point de vue humain, le projet est aussi une réussite. Nous avons pu nous répartir les tâches avec un diagramme de Gantt. Les échéances ont été tenues, même s'il reste encore des pistes d'amélioration.

Nous avons aussi bien réussi à partager nos informations, grâce à l'utilisation d'un notion. Nous avons pu ainsi partager nos avancées et la documentation que nous trouvions. Cela était aussi très pratique pour conserver des codes fonctionnels.

Dans la réalisation des différentes missions, nous avons principalement travaillé en groupe de 2, c'était stimulant et instructif. Nous avons ainsi pu garder un bon rythme de progression au cours des séances.

Enfin chaque membre de l'équipe est content d'avoir participé à ce projet et se réjouit du résultat final. Ce projet est donc une réussite même si notre robot n'est pas encore prêt à aller sur Mars.