

# ProTis - Robot Véronica : Rapport technique

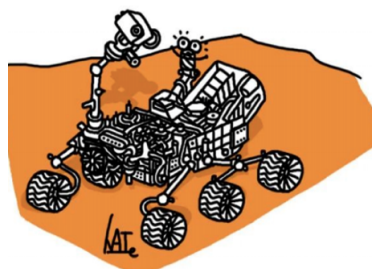
EL FERGOUGUI Hamza – RJIBA Eya  
PASTERNAK Stanislas – SALDUCCI Clément  
Institut d'Optique Graduate School

7 mai 2020

---

## Table des matières

<b>1</b>	<b>Description du projet</b>	<b>2</b>
<b>2</b>	<b>Schéma fonctionnel</b>	<b>2</b>
<b>3</b>	<b>Développement technique</b>	<b>3</b>
3.1	Faire avancer le robot . . . . .	3
3.1.1	Choix du moteur . . . . .	3
3.1.2	Fonctionnement du moteur pas à pas . . . . .	4
3.1.3	Alimentation du moteur pas à pas . . . . .	5
3.1.4	Pilotage du moteur . . . . .	5
3.2	Acquérir et gérer les données . . . . .	7
3.2.1	Capteur de température . . . . .	7
3.2.2	Capteur d'humidité . . . . .	8
3.2.3	Récupération des données par la Nucléo . . . . .	9
3.3	Communiquer avec le robot depuis la base . . . . .	10
<b>4</b>	<b>Conclusion générale</b>	<b>11</b>
<b>5</b>	<b>Bilan des compétences</b>	<b>11</b>



# 1 Description du projet

Notre projet est la conception et le développement d'un robot guidé à distance afin d'effectuer des relevés de température et d'humidité sur le sol martien.

La société SOLEC pour laquelle nous avons travaillé nous a préalablement communiqué les contraintes suivantes :

- Le robot doit être commandé par un système à distance.
- Il doit pouvoir se déplacer en ligne droite selon une distance transmise par l'utilisateur.
- Il doit pouvoir tourner sur lui même selon un angle transmis par l'utilisateur.
- Il doit être composé de trois roues : deux sont motorisées et une doit être libre de se déplacer.
- Les relevés de température et d'humidité doivent être effectués tous les 10 cm et communiqués à la base toutes les 10 minutes.

D'autre part, la société SOLEC souhaite que son robot puisse répondre à certains critères de performances qui sont les suivants :

- Le robot doit pouvoir avancer à une vitesse comprise entre 10 et 30 cm/s.
- Une erreur maximale de 2 cm sur la position et de 3° sur l'angle est tolérée.
- Le robot doit pouvoir se déplacer sur 1km sans que ses batteries ne soient rechargées.
- L'interface Homme-Machine, permettant de transmettre les ordres de parcours, doit pouvoir être utilisée sans formation préalable et les données doivent pouvoir être affichées en fonction de la distance ou du temps.

## 2 Schéma fonctionnel

Pour commencer, voici ci-dessous l'approche que nous avons souhaiter adopter dans la conception du robot résumée sous forme d'un schéma bloc :

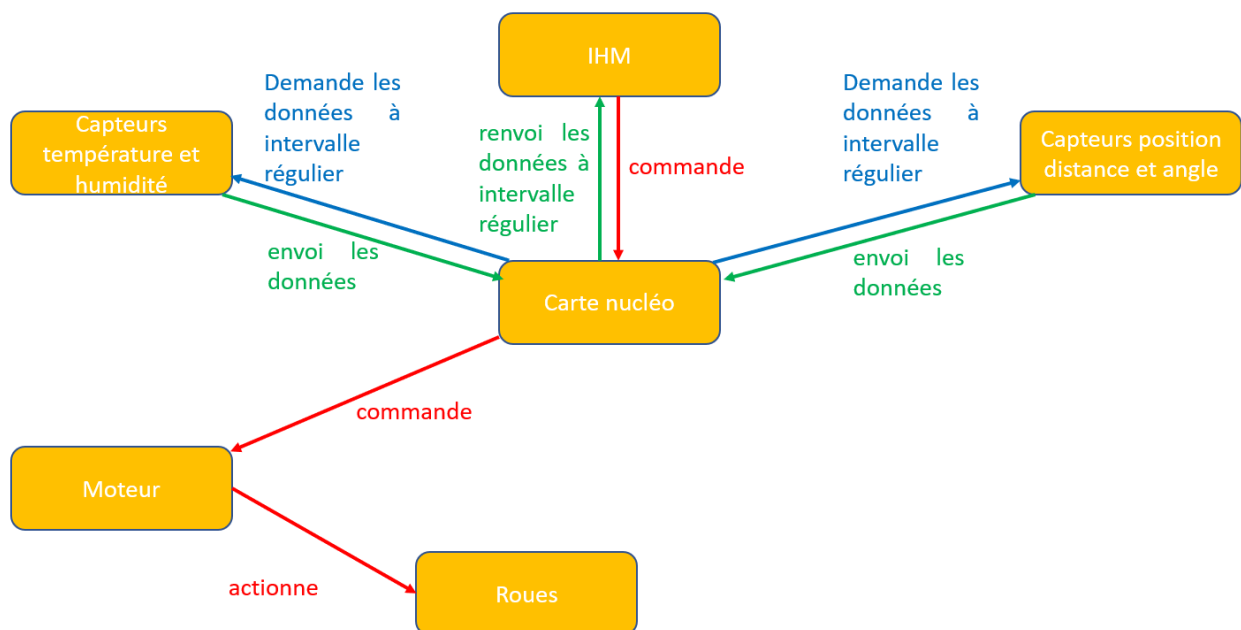


FIGURE 1 – Schéma fonctionnel du robot Veronica

La carte Nucléo L476RG fournie par l'école est l'élément central d'échange des informations :

elle fait le lien entre l'utilisateur et les différents composants du système.

Un moteur pas à pas permet de faire avancer le robot. La distance parcourue par le robot est contrôlée par des capteurs de position et d'angle. Côté relevés, ils sont assurés par deux capteurs indépendants, un pour la température et l'autre pour l'humidité.

Le tout est alimenté par une batterie fixée au robot afin que celui-ci puisse être commandé à distance.

## 3 Développement technique

### 3.1 Faire avancer le robot

#### 3.1.1 Choix du moteur

On compte mettre 3 roues dans le moteur dont 2 roues motrices commandées par 2 moteurs indépendants et deux roues libres.

En effet, les 2 roues motrices vont nous permettre de faire avancer le robot en ligne droite lorsque celles-ci tournent dans le même sens à la même vitesse.

Elles permettront aussi de faire tourner le robot sur lui-même, en fixant l'une de ces deux roues et en tournant l'autre. Le robot va donc effectuer une rotation d'un cercle dont le centre est le centre roue fixée. L'angle de rotation du robot dépendra du nombre de pas effectués par le moteur de la roue motrice en mouvement.

On commence par choisir quelques valeurs pour les caractéristiques de notre robot :

Masse totale du robot	1 kg
Accélération	$0,1 \text{ m.s}^{-2}$
Vitesse de déplacement $v$	$20 \text{ cm.s}^{-1}$
Rayon de la roue $r$	2 cm
Distance entre les 2 roues	30 cm

A l'aide de ces données, on essaiera de faire d'une manière très simple quelques calculs pour trouver les critères à prendre en compte pour le choix des moteurs.

Ceci nous permet de calculer la vitesse de rotation des roues  $\omega = \frac{v}{r} = 10 \text{ cm.s}^{-1}$

La force pesante d'un seul moteur se calcule à l'aide de la formule  $F = \frac{mg}{2} = 4,9 \text{ N}$  ce qui nous permet de calculer puissance du moteur  $P = F.v = 0,98 \text{ W}$  et son couple maximal  $C = \frac{P}{\omega} = 0,098 \text{ N.m}$ .

Le cahier de charges exige une erreur maximale de 2 cm sur la position et une erreur maximale de  $3^\circ$  sur l'angle. On a donc besoin d'un retour précis sur la position et l'angle afin de vérifier les bonnes performances de notre robot.

L'utilisation des moteurs pas à pas s'avère très utile dans ce cas grâce à sa très bonne réponse pour le démarrage et l'arrêt et son positionnement très précis. En effet, les moteurs pas à pas typiques ont une précision de 3 à 5% pour un pas et cette erreur est non cumulative d'un pas à l'autre. En outre, ils **ne nécessitent pas de boucle d'asservissement**, contrairement aux moteurs à courant continu. Ainsi, ils sont plus simples à commander.

Le rotor d'un moteur pas à pas est constitué d'un aimant permanent et le stator comporte deux paires de bobines. On souhaite tourner le champ magnétique en agissant sur les bobines alimentées pour faire tourner l'aimant fixé au rotor qui va se placer à chaque fois sur l'un des axes des bobines.

Il y a plusieurs types de moteurs pas à pas et on choisit d'utiliser des **moteurs pas à pas bipolaires** car ils ont un fonctionnement très simple à comprendre.

### 3.1.2 Fonctionnement du moteur pas à pas

Les moteurs pas à pas bipolaires comportent 2 paires de bobines et 4 fils (2 fils par bobine). Les bobines sont reliées par paire en série et sont commandées ensemble. Chaque bobine est commandée en courant positivement ou négativement de façon effectuer 4 pas par tour. L'aimant permanent du rotor est piloté en fonction du placement de ces bobines. En voici le schéma de principe :

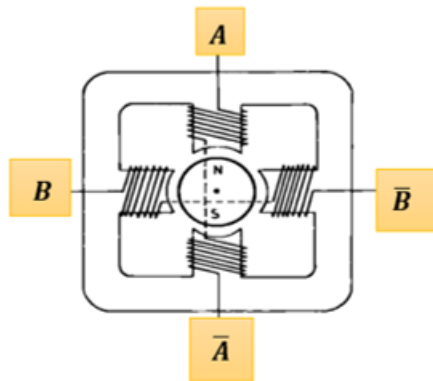
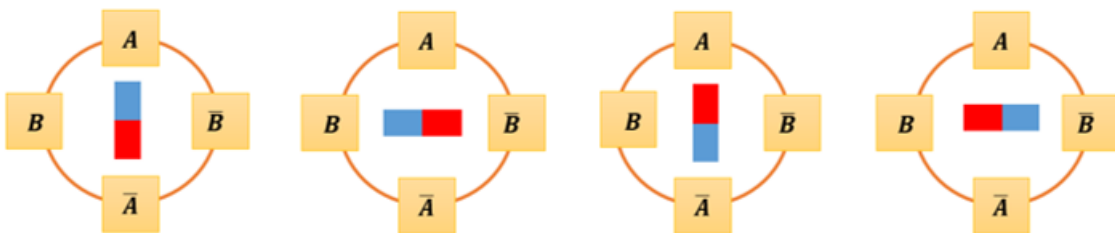


FIGURE 2 – Principe du fonctionnement d'un moteur pas à pas

Le schéma ci-dessous présente le principe d'un tour du moteur pas à pas, où l'aimant représente la polarisation des bobines (N = Nord et S = Sud) :



temps	A	$\bar{A}$	B	$\bar{B}$
$t_k$	N	S		
$t_{k+1}$			N	S
$t_{k+2}$	S	N		
$t_{k+3}$			S	N

FIGURE 3 – Modélisation d'un tour

Si on fait passer un courant entre A et  $\bar{A}$  pour alimenter la paire de bobine verticale, un champ électromagnétique va apparaître de part et d'autre du rotor qui va donc tourner sur lui-même pour se placer de façon à ce que son pôle Nord soit en face du pôle Sud du champ magnétique créé dans la première bobine et vice-versa pour la deuxième bobine.

En alimentant la paire de bobines horizontale B et  $\bar{B}$ , le rotor va, comme le cas précédent,

tourner et s'aligner vers les pôles Nord/Sud et Sud/Nord.

En changeant le signe de courant entre A et  $\bar{A}$  (*resp.* B et  $\bar{B}$ ), le moteur fait un autre quart du tour et le rotor s'oriente dans la direction opposée à celle avant le changement du signe du courant entre A et  $\bar{A}$  (*resp.* B et  $\bar{B}$ ).

### 3.1.3 Alimentation du moteur pas à pas

Le moteur bipolaire a besoin d'une inversion de courant pour pouvoir utiliser les bobines et changer les pas. Cette inversion de courant se fait à l'aide d'un pont en H.

Un pont en H ne peut piloter qu'une seule paire de bobines. Or le moteur bipolaire contient deux paires de bobines, donc on doit utiliser un composant qui contient deux paires de bobines. On va utiliser dans notre projet le composant L298. Pour le câblage, il suffit de relier les 4 fils de bobines aux sorties de la L298. Le circuit ci-dessous (Figure 4) représente le montage global. Il est inspiré d'un autre projet trouvé sur Internet, réalisé avec une carte Arduino.

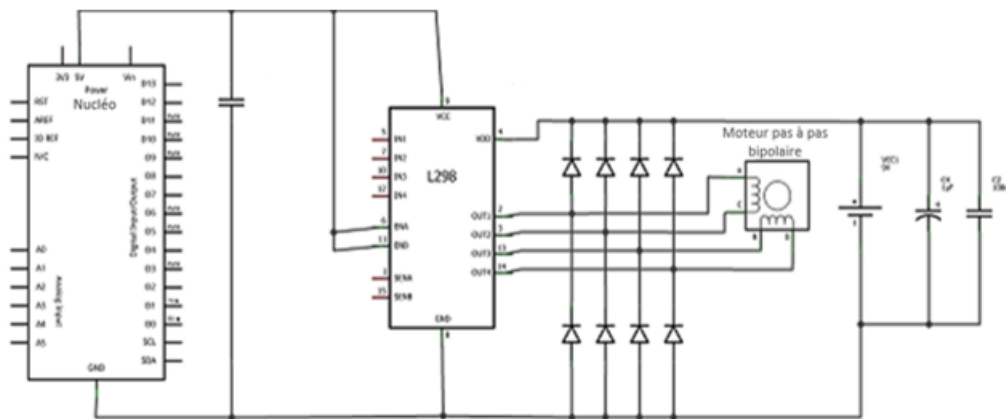


FIGURE 4 – Montage global de l'alimentation du moteur

Les condensateurs sur le montage aident l'alimentation en cas de forte demande de courant et filtrent les parasites engendrés par les bobines du moteur.

### 3.1.4 Pilotage du moteur

Le pilotage du robot se fait par deux mouvements : Une translation droite et une rotation du robot autour de lui-même. Le robot est équipé de 4 roues : 2 roues motrices liées à deux moteurs pas à pas et 2 roues libres qui suivent le mouvement des deux roues motrices et maintiennent l'équilibre du robot. Pour piloter nos deux moteurs nous utilisons des pins dans notre programme C, chaque pins va correspondre à une borne de bobine du moteur pas à pas : 2 Bobines - 4 Pins.

#### Translation

Pour piloter le moteur en translation nous avons tourné les deux roues motrices à la même vitesse et dans le même sens, ce qui permet au robot de bouger en avant et en arrière avec une vitesse fixe et en ligne droite. La vitesse du robot étant fixé à  $v = 20 \text{ cm.s}^{-1}$ , on a donc une vitesse de rotation  $\omega = 1,6 \text{ tr.s}^{-1}$ , ainsi le temps d'attente entre deux pas du moteur est  $T_{attente} = 0.157\text{s}$ .

Nous négligeons le temps d'accélération et d'arrêt dans notre modèle de moteur.

Le nombre de pas total permet de fixer le nombre total de steps que le moteur ne doit pas dépasser. Un bouton poussoir attaché à un ticker permet de commander quand on sort du programme et donc quand on arrête le robot.

Il existe deux façon pour arrêter l'acquisition des données, la première consiste à calculer le nombre de pas total permettant au robot de s'arrêter au bout de 10 minutes. Ce nombre de pas  $N$  dans le cas d'un translation est égal à  $N = \frac{10 \times 60}{0,157} = 3822 \text{ pas}$ .

La deuxième méthode est manuelle : L'utilisateur appui sur un bouton poussoir au bout de 10 minutes d'acquisition ce qui permet de sortir de la boucle et d'arrêter le programme.

La précision sur la position est donnée pour un angle  $\Delta\theta_{pas} = \frac{\pi}{2}$  par :

$$\Delta x_{propre} = r \times \Delta\theta_{pas} = 3,14 \text{ cm} \quad (1)$$

### Rotation

Le robot effectue une rotation autour de lui-même (autour du centre  $O'$  sur la figure ci contre) en tournant les deux roues dans deux sens différent avec la même vitesse.

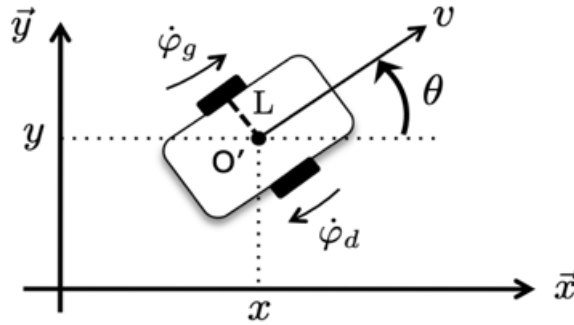


FIGURE 5 – Mécanisme de la rotation propre

La vitesse de rotation du robot autour de lui-même  $\omega_{propre}$  est donnée par :

$$\omega_{propre} = \frac{r}{2L}(\dot{\phi}_d + \dot{\phi}_g) \quad (2)$$

En faisant tourner les deux roues à la même vitesse et dans deux sens différents, on atteint une précision angulaire de  $12^\circ$  car  $\Delta\theta_{propre} = \frac{r}{2L} \times 2\Delta\theta_{pas}$ . Pour réduire cette précision angulaire, nous avons imaginé une nouvelle configuration où l'une des roues est bloqué et l'autre roue tourne.

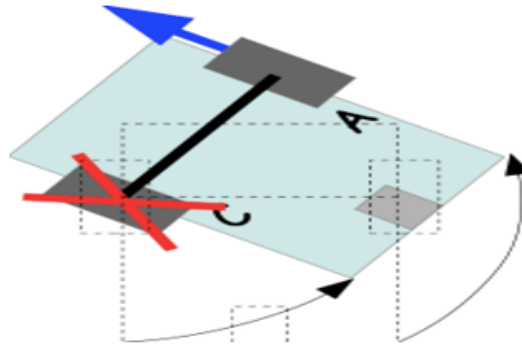


FIGURE 6 – Rotation propre autour de l'une des roues

La nouvelle vitesse propre est donnée par :

$$\omega_{propre} = \frac{r}{2L} \dot{\varphi}_A = 5,7 \cdot 10^{-2} \text{ tr.s}^{-1} \quad (3)$$

Ainsi la précision angulaire est réduite par 2, c'est-à-dire atteint  $6^\circ$ .

## 3.2 Acquérir et gérer les données

Premièrement afin d'acquérir les données de température et d'humidité, nous avons dû choisir des capteurs adaptés aux conditions martiennes. Cette étape qui ne semble pas demander de compétences techniques particulière n'a pas été la plus aisée pour autant car il nous a fallu choisir parmi des centaines de capteurs différents qui se ressemblent beaucoup du point de vue de personnes qui sont pas des experts en électronique.

### 3.2.1 Capteur de température

Pour relever la température, nous avons choisi une résistance en platine de  $100\Omega$  (modèle XF-988-FAR). Ce choix a été motivé par la grande plage de mesure de cette résistance ( $-200^\circ\text{C}$  à  $+650^\circ\text{C}$ ) puisque sur Mars la température moyenne est d'environ  $-60^\circ\text{C}$  et peut descendre jusqu'à  $-120^\circ\text{C}$ . Sur le plan technique, les deux grands points forts de ce capteur sont :

- La linéarité : la résistance varie selon une loi affine de la température.
- Le fait qu'il n'y ait pas besoin de calibration : le modèle de Pt $100\Omega$  est connu.

La Datasheet, donne la loi de variation de la résistance en fonction de la température :

$$R(T) = 100 + 0,4 \times T \quad (4)$$

Lorsque l'on branche ce capteur à la carte nucléo, il est facile de relever la tension au borne de celui-ci mais on ne peut pas accéder à une mesure directe du courant pour ensuite calculer la résistance. Pour palier à cela, on utilise le pont diviseur de tension suivant :

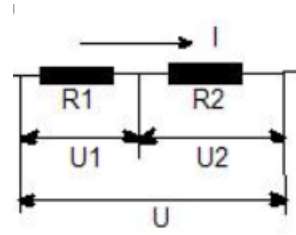


FIGURE 7 – Mesure du courant par pont diviseur de tension

Ici  $R_1$  représente le capteur et  $R_2$  est une résistance à choisir. Nous avons décidé de choisir  $R_2$  telle que  $U_1 = \frac{U}{2}$  pour  $T = 60^\circ\text{C}$  la température moyenne sur Mars. Ceci donne alors  $R_2 = 76 \Omega$ .

La valeur de la résistance, donc de la température, peut se calculer ensuite sous la forme :

$$R_1(T) = \frac{R_2 U}{U - U_1} \text{ soit } T = \frac{\frac{R_2 U}{U - U_1} - 100}{0,4} \quad (5)$$

D'autre part, on vérifie qu'avec cette valeur de  $R_2$ , la tension aux bornes du capteur aux températures extrêmes n'est pas nul. On trouve pour  $T = -150^\circ\text{C}$ ,  $U_1(t = -150^\circ\text{C}) = 1,14 \text{ V}$ .

La valeur de  $U$  est connue et celle de  $U_1$  est lue par la nucléo à l'aide d'une entrée analogique.

Ce que nous avons ensuite vérifier est de vérifier si les variations de la résistance sont détectables par la carte nucléo. Par exemple, pour une variation  $dT = 0,1^\circ\text{C}$ , la tension  $U_1$  mesurée par la nucléo au borne du capteur varie de :

$$dU_1 = U \times \frac{2R_1 + R_2}{(R_1 + R_2)^2} dR_1 = 3,3 \times \frac{2R_1 + 76}{(R_1 + 76)^2} 0,04 \quad (6)$$

car pour  $dT = 0,1^\circ\text{C}$  on a  $dR_1 = 0,4 \times dT = 0,04 \Omega$ .

Le rapport  $\frac{2R_1 + R_2}{(R_1 + R_2)^2}$  dépend de la valeur de la résistance du capteur donc de la température. Ce rapport décroît quand  $R_1$  augmente, donc si la variation du signal est détectable à  $T = 20^\circ\text{C}$  (température maximale à la surface du sol martien), alors elle sera détectable sur toute la plage de température ( $-150^\circ\text{C}$  à  $+20^\circ\text{C}$ ).

A une telle température, ceci donne  $dU_1 = 1,13 \text{ mV}$ . Comme la carte nucléo a une résolution de 12 bits (4096 valeurs possibles), ceci permet de détecter des variations de tension d'environ 1 mV, ce qui convient parfaitement avec le critère énoncé précédemment, sous réserve que le bruit thermique de la résistance ne soit pas supérieur à 1 mV.

Le code en annexe présente la méthode pour récupérer les données de températures sur la carte nucléo.

### 3.2.2 Capteur d'humidité

Pour relever le pourcentage d'humidité au sol, nous avons choisi un capteur 3 broches (modèle HIH-5030-001) dont le schéma de principe est donné ci-dessous :



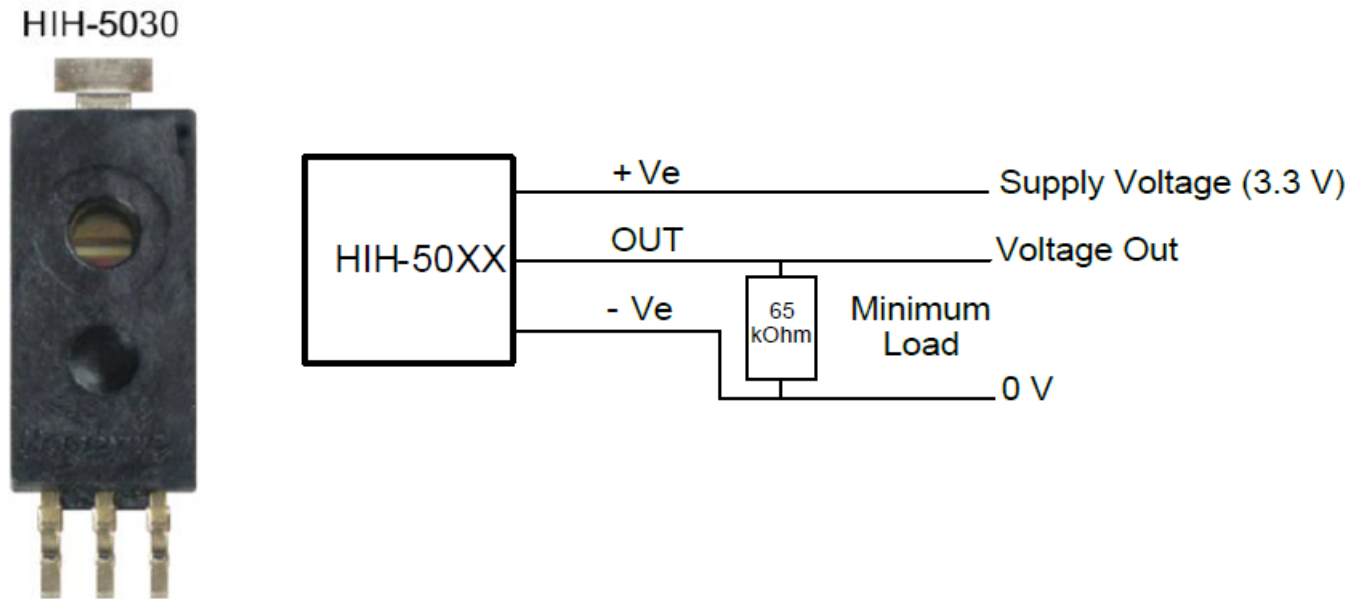


FIGURE 8 – Image (gauche) et schéma de branchement (droite) du capteur d'humidité

Les points forts et critères qui ont fait que nous avons sélectionné ce capteur sont les suivants :

- Une alimentation entre 0 et 3,3 V donc possible avec une carte Nucléo.
- Un comportement quasiment linéaire, qui présente un léger hystérésis d'environ 1% sur le taux d'humidité.

Pour ce qui est du branchement, on utilise une résistance de 65 k $\Omega$  entre la masse et la broche de sortie comme dit sur la datasheet et on alimente avec le 3,3 V de la Nucléo. On connecte la broche de sortie à une entrée analogique de la carte nucléo.

La datasheet indique aussi que le rapport entre la tension en sortie  $V_{out}$  et le pourcentage d'humidité varie en fonction de la température. La solution à laquelle nous pensons serait donc de placer un système de chauffage asservi en température afin de garder une température constante autour du capteur d'humidité.

A 25°C le rapport  $\frac{\% \text{ d'humidité}}{V_{out}}$  est environ :  $\frac{96}{2} = 48 \% \text{ d'humidité}/V$ . D'autre part, pour 0% d'humidité, la datasheet montre que la tension en sortie du capteur est  $V_{out} = 0,5 V$ . Ceci nous permet donc d'avoir la loi de variation du pourcentage d'humidité (que l'on note "%H") en fonction de  $V_{out}$  :

$$\%H = (V_{out} - 0,5) \times \frac{2}{96} = -1,04 + 2,08 \times V_{out} \quad (7)$$

D'autre part, cette équation montre que la variation de tension sur la broche de sortie est de 2 mV pour une variation de 0,1 % d'humidité. Comme pour le capteur de température, ceci devrait suffire si le bruit du circuit est inférieur à cette valeur.

Le code en annexe présente la méthode pour récupérer les données d'humidité sur la carte nucléo.

### 3.2.3 Récupération des données par la Nucléo

Le but de ce code est d'acquérir les données renvoyées par les capteurs, puis de les envoyer à la base toutes les 10 minutes. Le cahier des charges impose que les données soient collectées

tous les 10 cm. L'idée est d'utiliser un fichier texte que l'on va remplir au fur et à mesure que le robot avance, puis que l'on va envoyer à la base et effacer au bout de 10 minutes.

Pour acquérir les données, on utilise un pointeur sur une donnée spécifique dont le type est **FILE**. Puis à l'aide de la fonction **append**, on complète le fichier.

Il faut ensuite faire cette action tous les 10 cm. La phase d'accélération du robot lorsqu'il est à l'arrêt est négligée. L'angle de rotation appelé "**theta**" dans le code est celui donné par l'interface Homme-Machine (IHM). On n'a pas eu le temps d'étudier la connexion entre l'IHM et la Nucléo, donc cette variable est juste utilisée ici pour montrer comment on aurait fait.

Ensuite pour réaliser l'envoi toutes les 10 minutes, on écrit à l'aide du tutoriel sur le LenSE « faire une action à intervalles régulier ». On utilise ici un **ticker**.

L'envoi des fichiers se fait par liaison radio fréquence. C'est une partie que nous n'avons réellement eu le temps d'aborder, mais nous pensions utiliser le protocole RS232 en suivant le tutoriel publié sur le LEnSe.

### 3.3 Communiquer avec le robot depuis la base

Pour communiquer entre l'utilisateur et le robot on a créé une interface graphique codée en C++ en utilisant la bibliothèque graphique Qt. La structure choisie pour cette interface est la suivante :

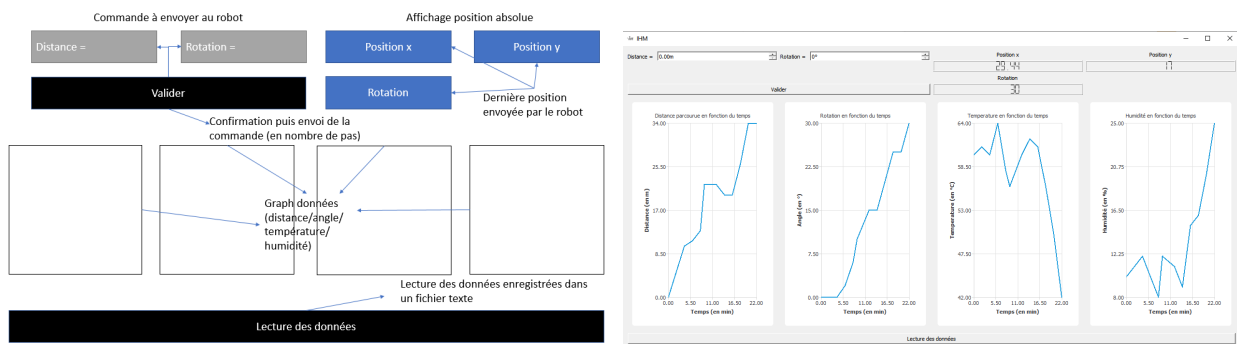


FIGURE 9 – Structure schématique et capture de l'interface graphique

L'interface possède deux champs permettant de remplir la commande à envoyer au robot et un bouton valider (qui peut être activé en appuyant sur la touche Entrée). La position et l'angle de rotation du robot sur un plan dont l'origine est sa position initiale est donnée par un affichage en haut à gauche de l'interface. Les données relevées par le capteur (température, humidité, distance et angle) sont affichées par quatre graphiques donnant chaque mesure en fonction du temps et dont les axes sont automatiquement calculés pour correspondre aux valeurs limites des données. Le bouton en bas permet d'actualiser les données affichées par les graphiques et les affichages avec les dernières valeurs envoyées par le robot. Si le fichier de donnée n'est pas lisible un message d'erreur s'affiche.

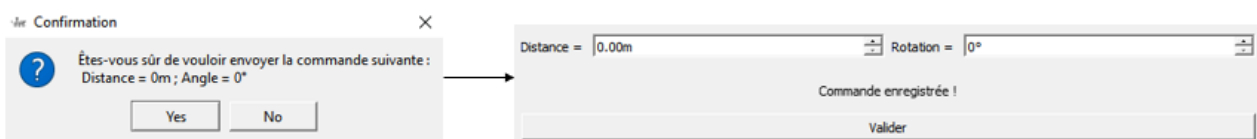


FIGURE 10 – Confirmation de l'envoi de la commande

Pour éviter qu'une mauvaise commande soit envoyée par erreur une boîte de dialogue demandant confirmation apparaît quand on appuie sur le bouton valider. Si l'on confirme la commande est écrite dans le fichier texte envoyé au robot et un texte apparaît un court instant pour informer que la commande a bien été envoyée.

## 4 Conclusion générale

Notre projet n'a certes pas abouti mais il respecte d'ores et déjà un certain nombre de points du cahier des charges. D'autre part, le robot est capable d'avancer en ligne droite et de tourner.

Le robot respecte les points suivants :

- Il avance à une vitesse de 20 cm/s donc bien entre 10 et 30 cm/s et peut se déplacer en ligne droite.
- Il effectue des relevés tous les 10 cm et est capable de les envoyer toutes les 10 minutes.
- L'interface Homme-Machine est faite telle que n'importe qui peut commander le robot.

En revanche les points suivants sont à améliorer :

- Il ne respecte pas la tolérance sur la position et en angle respectivement de 2 cm et 3°. Pour palier à cela, nous avons imaginé un système robot avec deux moteurs continus commandés par des bornes PWR (Power Removal) qui garantissent l'arrêt du moteur et tout redémarrage intempestif ainsi qu'un programme de pilotage, en direction et puissance par un GBF.

Grâce à un signal horloge et en modifiant le rapport cyclique de ce signal nous pouvons piloter la puissance et le pas de notre moteur ce qui nous permettra d'optimiser encore la précision angulaire et la précision sur la position de notre Robot Véronica.

- La connexion entre la carte nucléo et l'interface Homme-Machine n'a pas été étudiée, par conséquent il ne peut être piloté à distance. Nous avons pensé à utiliser une liaison radio-fréquence et une connexion de type RS232.
- La batterie n'a pas été étudiée donc on ne peut garantir une autonomie de 1 km.

## 5 Bilan des compétences

Durant de projet, nous avons pu développer de nouvelles compétences. Voici un retour de chaque membre de l'équipe.

### **Eya Rjiba**

Le projet ne m'a pas seulement aidée à développer des nouvelles compétences techniques en électronique, il m'a aussi amené à communiquer avec tous les membres de l'équipe. Ceci a permis de faire avancer le travail et a permis à toute l'équipe d'être très efficace.

En plus, vu que le projet était fait en télétravail, l'organisation était plus indispensable que jamais. Toute l'équipe a fait preuve d'une très bonne organisation au niveau de la gestion des documents et des tâches. Ceci a été réalisé en partageant quotidiennement l'avancement du travail sur la plateforme unique Basecamp et en faisant chaque semaine le point sur l'avancement du travail.

Le projet m'a aidé aussi à améliorer ma capacité à prioriser les tâches, et m'a montré qu'une auto-formation, dans un cadre guidé, est une très bonne méthode d'apprentissage.

### **Hamza El Fergougui**

Le projet Protis a été une occasion pour moi pour acquérir plusieurs savoirs faire pendant les

quatre phases allant du «recrutement et validation des acquis» jusqu'à «la présentation du prototype virtuel». La compétence essentiel que j'ai appris lors de ce projet est **l'adaptabilité** au nouveau contexte de travail et aux nouveaux défis que nous devons relever : entre communication, gestion de travail et recherche de solutions.

Ce projet a été un premier pas pour moi vers l'espace du travail virtuel en équipe, et ça m'a permis de bien saisir les enjeux derrière ce travail et comment surmonter les difficultés liées au travail à distance.

Outre les nombreuses compétences techniques acquises au cours de ce projet grâce à l'accompagnement des professeurs ainsi que les ressources en ligne mis à disposition, j'ai pu développer de nouvelles compétences en relationnelles parmi lesquels a été mon rôle essentiel sur la **gestion d'équipe**.

### Clément Salducci

J'ai travaillé sur les capteurs et le traitement des données. Ce travail m'a permis de développer plusieurs compétences importantes selon moi.

Premièrement, il a fallu chercher les bons capteurs et ceci n'a pas été aussi facile qu'escompter car je n'y connaissais pas grand chose dans ce domaine. J'ai donc appris à **cibler les composants faciles d'utilisation et adaptés à notre robot**.

Il a ensuite fallu faire l'étude de ces capteurs. Ce fut simplement une étude théorique et ceci a posé quelques problèmes car je ne savais pas si la voie que je suivais était la bonne. Je me suis aperçu un peu tardivement que j'avais loupé certains éléments. Cependant, cette phase m'a tout de même appris à **développer des solutions en autonomie** tout en **m'adaptant aux demandes** des autres membres de l'équipe.

D'un point de vue technique, le traitement des données m'a permis de découvrir de nouveaux aspects de la programmation tel que l'utilisation de **tickers**.

### Stanislas Pasternak

Ce projet m'a permis de découvrir la programmation d'interface graphique et la programmation orientée objet mais surtout à rechercher à l'aide d'une documentation les classes/fonctions utiles pour réaliser le programme voulu dans une bibliothèque extrêmement complexe et fournie qu'est Qt.

J'ai aussi appris à adapter mon travail au demande faites par les autres membres de l'équipe. L'interface graphique devant communiquer avec les autres programmes du robots j'ai dû modifier certaines de mes idées initial pour qu'elle s'adapte au projet dans sa globalité.

La situation du télétravail m'a permis d'apprendre à gérer un projet, l'organisation et la communication étant essentiel dans cette configuration.