Compte rendu SAE 1

Semestre 3

Groupe C7

TOUAMI Nedjoua

PONNOU Stévane

Robot en milieu confiné

Sommaire :

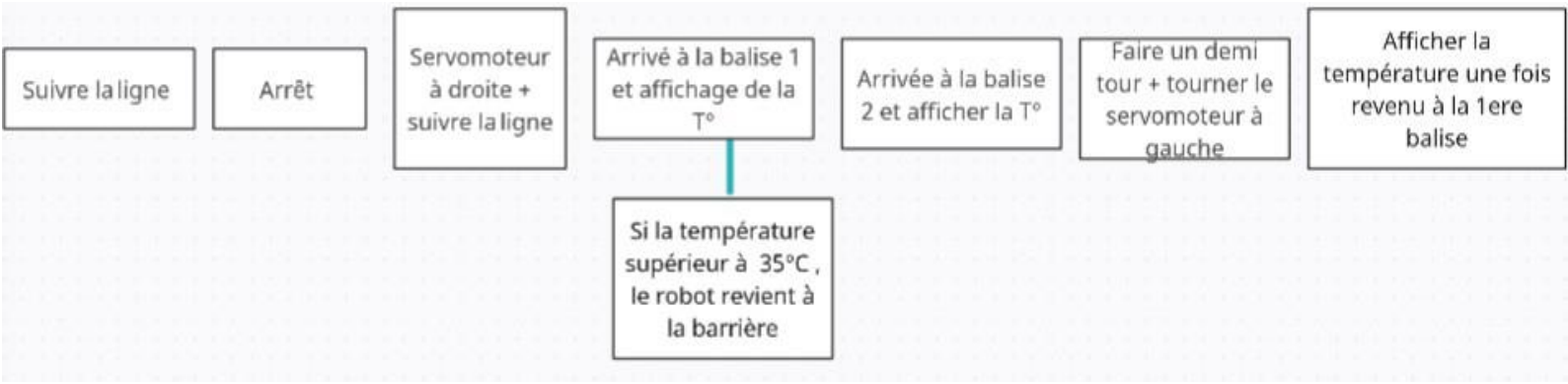
Présentation du sujet :

Le but de cet SAE est de programmer un robot qui servira à parvenir aux endroits où l'homme ne peut pas accéder.
Le robot aura pour but de détecter une barrière et s'arrêter, une fois la barrière levée, il devra avancer et déplacer son servomoteur à droite afin de détecter une 1ère balise, arrivé à cette balise il devra afficher la température (intégré au robot). Il devra ensuite avancer pour détecter une 2nd balise, une fois détecté, le robot effectuera un retour et affichera la température dès qu'il atteindra la 1ère balise.
Si le robot affiche une température supérieure à un seuil à l'aller à la 1ere balise, le robot reviendra automatiquement à la barrière

Cahier des charges :
- Se déplacer à l'aide des moteurs
- Suivre une ligne
- Détecter un obstacle à l'aide du télémètre

Afficher la température, ALERTE si le seuil est dépassé sur un écran LC

Schéma fonctionnel :



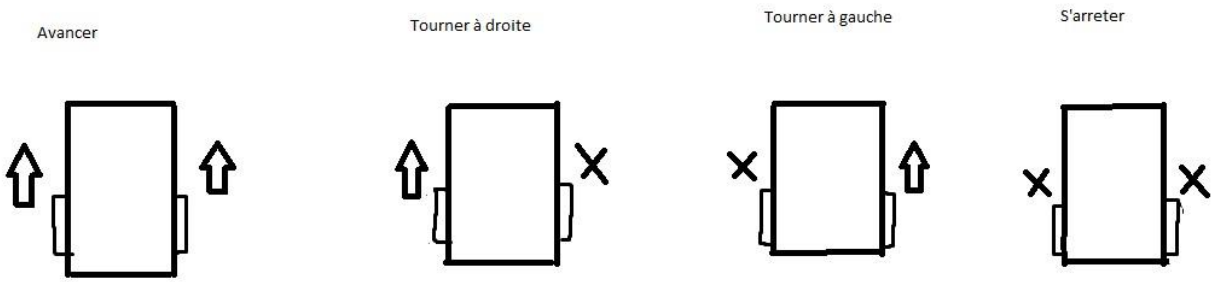
I) Mouvement du robot

Fonctionnement / câblage :
Les moteurs des roues sont alimentés en 5V et sont relié avec deux fils

Pour mettre en mouvement le robot, nous avons mis en place 4 fonctions :

- Avancer
- Tourner à droite
- Tourner à gauche
- Arrêter
- Demi-tour

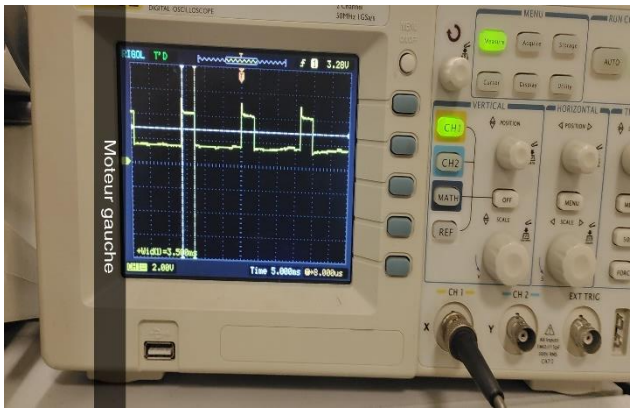
Dans la logique : (robot vue du dessus)

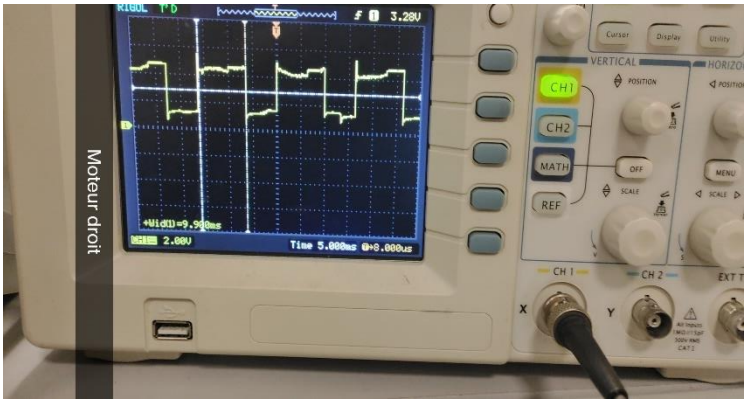


⇧ : La roue est en marche (1) ✖ : La roue ne tourne pas (0)

Pour appliquer ces fonctions dans les moteurs nous avons mis en place le Timer0, ce Timer permet de compter de 0 à 255, il est un compteur à 8 bits
Ce qui nous permettra de régler la vitesse est le sens des roues
Pour cela nous devons configurer 2 registres pour les 2 moteurs en question :
TCCR0A = 0x6A
TCCR0B = 0x05
Le PORT D est déclaré en sortie
Nous avons une période de T= 15 ms
Ces registres sont paramétrés de sorte à avoir une pré-division de 1024, un PWM rapide

Une fois les registres configurés, nous avons ces courbes :





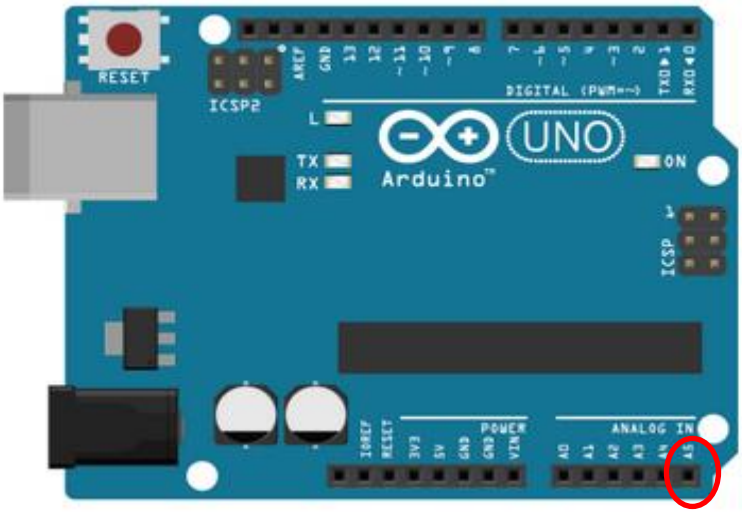
Pour créer les fonctions nous avons besoins de 2 registres :
- OCR0A (Moteur droit)
- OCR0B (Moteur gauche)

Ces registres permettent de définir la vitesse et le sens des moteurs, ainsi nous réalisons les fonctions :

- Si OCR0A = OCR0B = 1➔ Le robot avance
- Si OCR0A = OCR0B =0 ➔ Le robot s'arrête
- Si OCR0A =0 et OCR0B = 1 ➔ Le robot tourne à droite
- Si OCR0A = 1 et OCR0B = 0 ➔ Le robot tourne à gauche
- Si OCR0A = 1 (pendant un petit temps) et OCR0B = 0 ➔ Demi-tour

II) Télémètre

Le télémètre est mis en place afin de détecter les obstacles, il détecte dans un premier temps la barrière puis les deux balises.

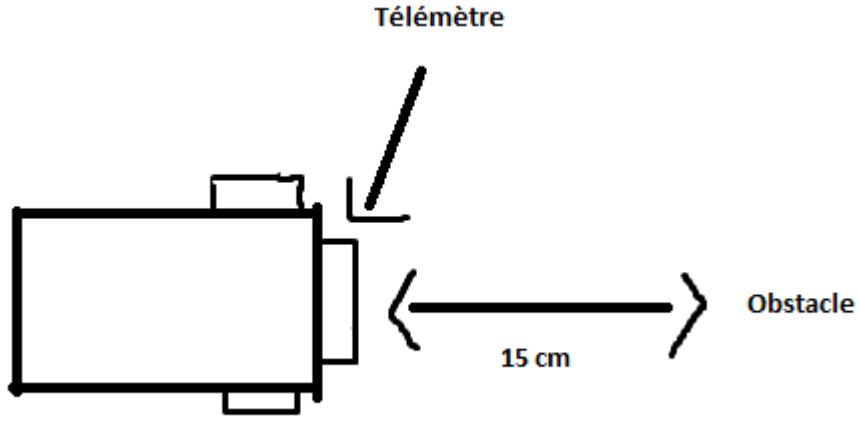


Le télémètre est alimenté en 5V et est branché sur le PIN A5 de l'Arduino, 3 fils proviennent du télémètre :

- Fils rouge ➔5V
- Fil noir ➔ GND
- Fil bleu ➔ A5

Nous avons programmé le télémètre de telle sorte à afficher le mot « OBSTACLE » lorsqu'il détecte un obstacle à moins de 15 cm

Schéma vue du dessus :



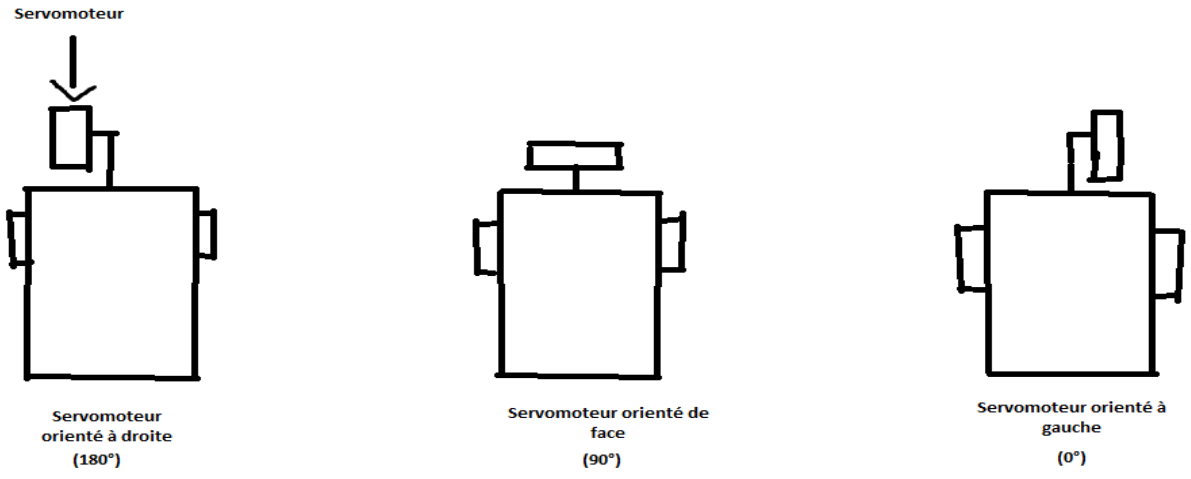
Nous avons converti les valeurs que nous trouvons en centimètre, en valeur de 0 à 255 , nous obtenons donc des valeurs numériques qui varient de 0 à 255 grâce au CAN .
En paramétrant les registres ADMUX et ADCSRA nous avons :
ADMUX = 0x25 ➔ Connexion du capteur sur la broche A5
ADCSRA = 0x87

Donc avec la conversion analogique-numérique, nous avons 15cm = 100

Si distance> 100 (Détection d'obstacle)
➔la variable TELE =1 , affichage « OBSTACLE »
Sinon TELE = 0 (Absence d'obstacle)

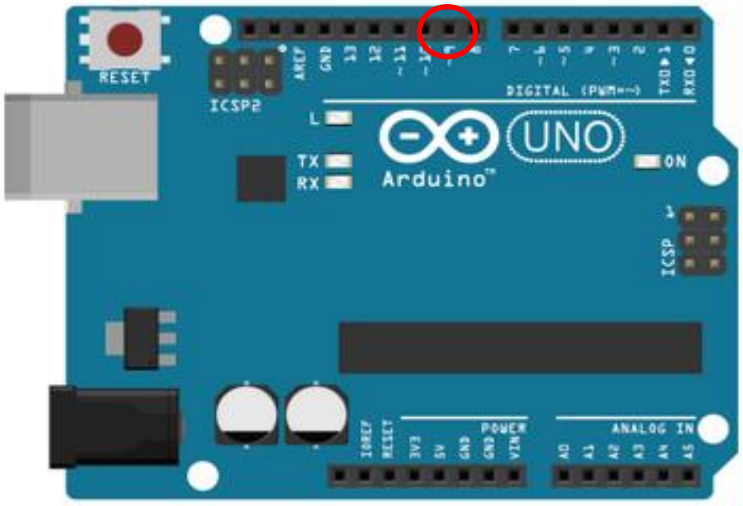
III) Servomoteur

Le servomoteur a pour but de contrôler l'orientation de télémètre
Schéma vue du dessus :



Au début du scenario, le servomoteur prend sa position initiale à 90° après la détection de la barrière et une fois qu'elle n'est plus détectée, le servomoteur

tourne à droite ce qui veut dire qu'il se positionne à 180° pour permettre la détection des deux balises.
Lorsque le robot fera le demi-tour, le servomoteur tournera à gauche donc à 0°.



Le servomoteur est alimenté en 5V et est branché sur la broche 9 de l'Arduino.
De la même manière que le télémètre, 3 fils proviennent du télémètre :

- Fil rouge ➔ 5V
- Fil noir ➔ GND
- Fil Vert ➔ Broche 9

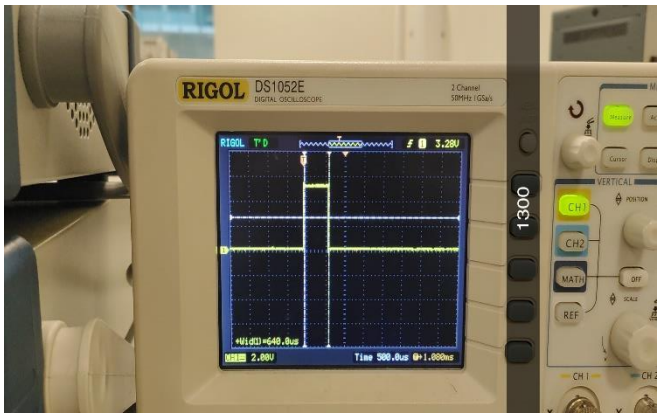
Pour programmer le servomoteur sur l'Arduino, nous avons utilisé le Timer1 qui compte jusqu'à 65535.
Nous avons donc besoin des registres suivant :
TCCR1A = 0x82
TCCR1B = 0x1A

Ces registres ont été configuré de façon à avoir un PWM correct et une pré-division de 8,
Sachant que nous avons une période T = 40ms

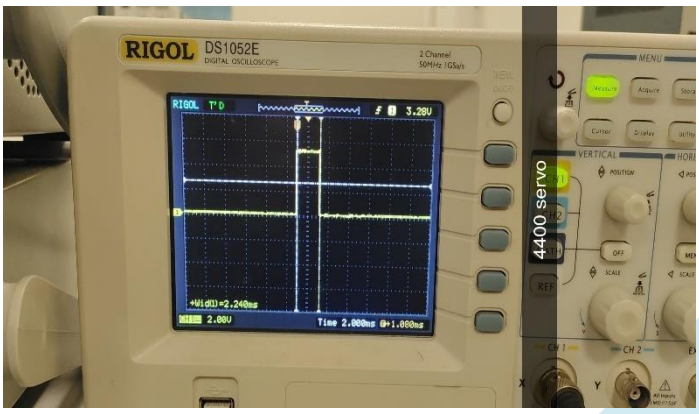
Il faut donc créer 3 fonctions :

- Servomoteur à droite
- Servomoteur au milieu
- Servomoteur à gauche

A l'aide de l'oscilloscope nous avons les courbes suivantes en fonction de la position du télémètre :



Servomoteur à droite



Servomoteur à gauche

Nous obtenons donc 3 valeurs en fonction des Temps haut de chaque position

- Pour la position droite ➔ Th = 544 µs
- Pour la position milieu (face) ➔ Th = 1480 µs
- Pour la position gauche ➔ Th = 4920 µs

Ce qui nous donne des valeurs de OCR1A étant le double des temps haut de chaque position

- OCR1A (droite) = 544 * 2 =1088 ➔1300 (Pratique)
- OCR1A (milieu) = 1480 * 2 = 2960 ➔ 3000 (Pratique)
- OCR1A (gauche) = 2410 * 2 = 4920 ➔ 4400 (Pratique)

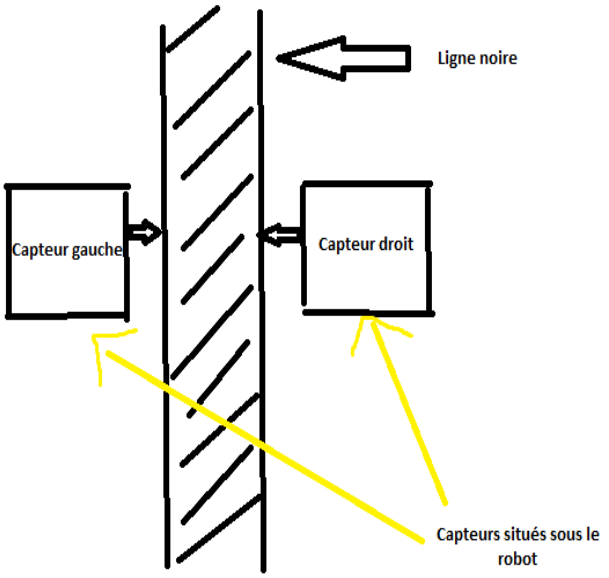
Les valeurs que nous avons trouvé à la pratique sont dues à des décalages du servomoteur

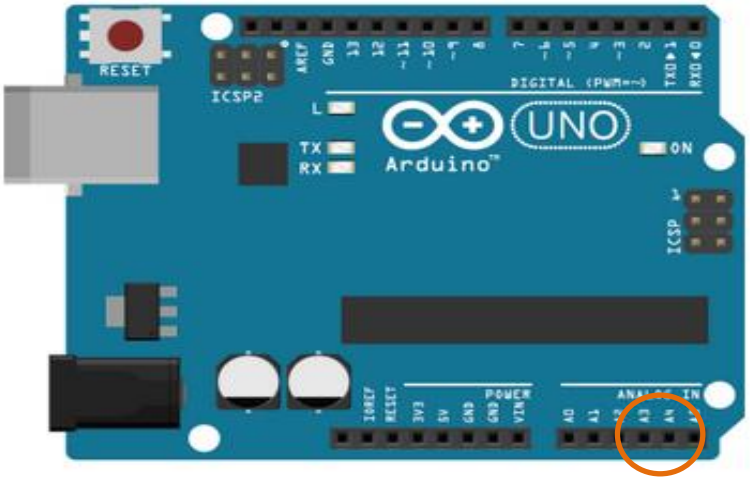
Comme mentionné plus tôt dans cette partie, il suffira de créer 3 fonctions qui illustreront les 3 positions :
ServoDR ➔ OCR1A = 1300
ServoML ➔ OCR1A = 3000
ServoGH ➔ OCR1A = 4400

IV) Suiveur de ligne

<u>Schéma vue du dessus, sous le robot :</u>

Le robot suit une ligne noire à l'aide des capteurs infrarouge
placer en dessous de celui-ci





Les capteurs sont chacun alimenté entre 0 et 5V et sont branché respectivement dans les PIN A2 ET A3 :

- Capteur droit ➔Fil vert (A2)
- Capteur gauche ➔ Fil jaune (A3)

Les capteurs sont programmés de façon à détecter une zone noire et une zone blanche
Par exemple, sur le capteur gauche, lorsque le noir est détecté, une valeur de 200est affichée sur le moniteur série de l'Arduino, tandis que quand le blanc est détecté une valeur bien plus petite est affichée (<20).

En testant les capteurs dans la zone noir et blanc nous trouvons ces valeurs :

- Capteur droit ➔ blanc = 16 ; noir = 200 ;
- Capteur gauche ➔ blanc = 16 ; noir = 200 ;

Le but étant que la ligne soit au milieu des 2 capteurs

Pour programmer le suiveur de ligne nous ferons appel aux fonctions du mouvement du robot :

<u>Pseudo-code du suiveur de ligne :</u>

Si capteur droit > 160 et capteur gauche < 16
➔ Robot tourne à gauche

Sinon si capteur gauche > 160 et capteur droite < 16
➔ Robot tourne à droite

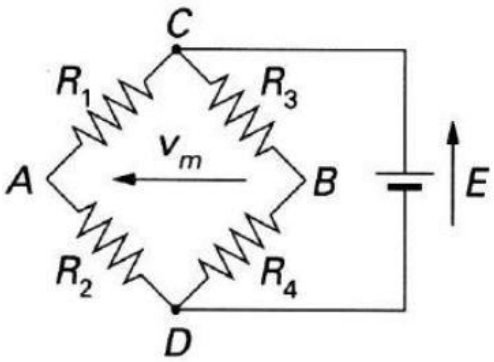
Sinon
➔Avancer

Nous devons aussi convertir ces valeurs en valeurs numérique avec le CAN et les registres suivant :
ADMUX = 0x22 ➔ Connexion à la broche A2
ADMUX = 0x21 ➔ Connexion à la broche A3
ADCSRA = 0x87

V) Capteur de température

Le capteur de température servira donc à afficher la température à partir de la balise 1

Schéma de câblage du pont de Wheatstone :



R2=R4= 4,7 K Ω R1=100 Ω
R3 est la sonde PT100

Calcul de Vm :

Rappelons la formule :

$$Vm = 5 * \frac{\mathrm{R2} * \mathrm{R3} - \mathrm{R1} * \mathrm{R4}}{(R1 + R2) * (R3 + R4)}$$

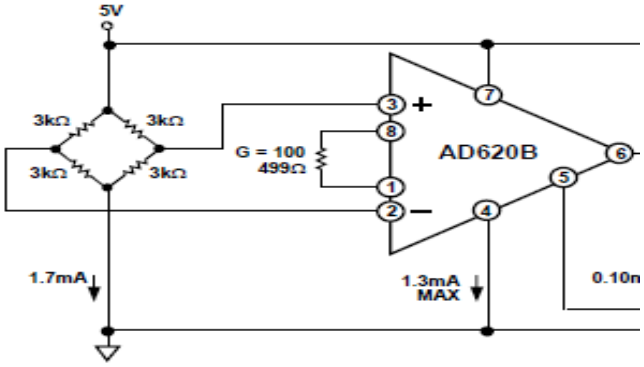
R3=PT100= 138.5 Ω➔100°C

D'après la datasheet du capteur PT100 :

Pour 0°C :
PT100=100 Ω

Pour 100°C :
PT100=138.5 Ω ➔ Vm=39 mV

Gardons le principe de ce câblage :



G= vs/Vm = 5/0.039 = 128

Rg = 49400 / 128-1 =389 ➔ 390 Ω

La valeur en sortie du pont de Wheatstone doit être amplifié avec un AD620
Nous avons donc une amplification de 128

- R2 = R4 = 4,7 kΩ
- Rg = 390 Ω
- R3 (sonde PT100)= 138,5 Ω

Vérification pour 100°C ➔ Vs=G*Vm= 128*0.039=4,98 ≈ 5V

-OK ➔ Temp < 22 °C
-Surveillance➔ 22°C < Temp <25°C
-Alerte➔ Temp >25 °C

Comme pour les capteurs de suiveur de ligne et le télémètre, pour afficher la température nous aurons besoin de convertir ces valeurs en tension, image de la grandeur en °C
A l'aide de ces registres :
ADMUX=0x24 ➔ Connexion avec la broche A4 de l'Arduino
ADCSRA=0x87

VI) Assemblage de toutes les parties

Pour réaliser le scénario final, nous avons assemblé toutes les parties dans un même programme
Nous avons donc intégré une variable t, qui représente les étapes du scénario
, le scénario débute avec t=0(condition initiales), cette variable s'incrémente en fonction les étapes validées

Tant que t=0 et distance < 100 **(Absence d'obstacle)**

➔Avancer
➔Suiveur de ligne
➔OCR1A = 3000 (Servomoteur position face)
➔t=1 (1ère étape)

Tant que t=1 et distance > 100 **(Détection de la barrière)**
➔Affichage du mot « obstacle »
➔Arrêter
➔t=2 (2ème étape)

Tant que t=2 et distance < 100 **(Barrière levée, avancer avec le servomoteur à droite)**
➔Avancer
➔Suivre la ligne
➔OCR1A = 1300 (Servomoteur à droite)
➔t=3 (3ème étape)

Tant que t=3 et distance > 100 **(Détection de la 1ère balise et affichage de la température)**
➔Avancer
➔Afficher le mot « Balise 1 » et afficher la température
➔t=4 (4ème étape)

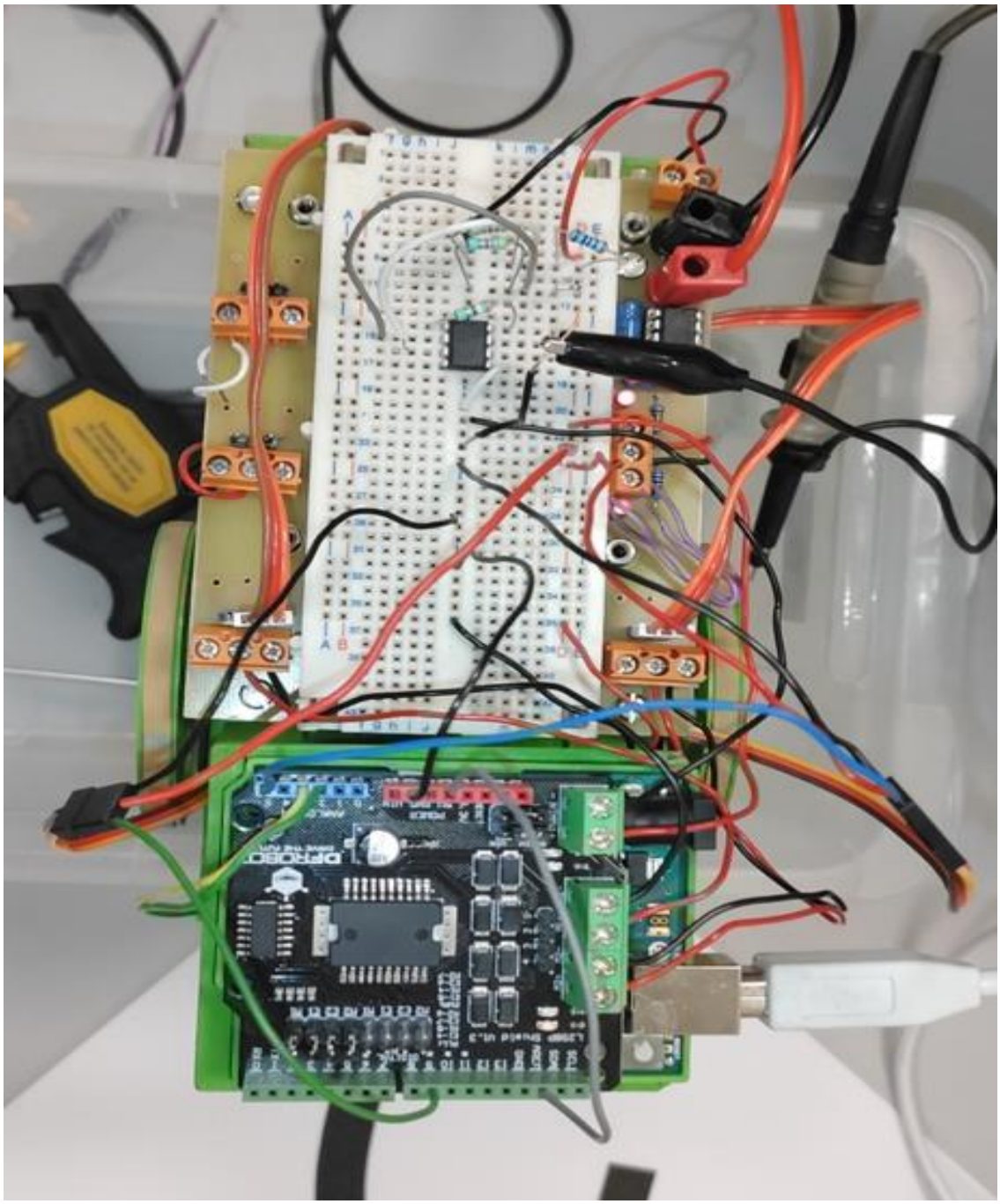
Tant que t=4 et distance < 100 **(Absence d'obstacle , avancer normalement)**
➔Avancer
➔t=5 (5ème étape)

Tant que t=5 et distance > 100 **(Détection de la 2ème balise et affichage de la température)**
➔Affichage du mot « Balise 2 » et afficher la température
➔Avancer
➔t=6 (6ème étape)

Tant que t=6 et distance < 100 **(Absence d'obstacle , effectuer le demi-tour et orienter le servomoteur à gauche)**
➔Afficher le mot « Demi-tour »
➔Demi-tour
➔OCR1A = 4400 (Servomoteur position gauche)
➔t=7 (7ème étape)

Tant que t=7 et distance > 100 **(Détection de la 1ère balise et réafficher la température)**
➔Avancer
➔t=8 (8ème étape)

Schéma de câblage de l’ensemble, Moteurs + Télémètre + Servomoteur + Capteur de ligne + Température :



VII) Conclusion

Nomenclature :

Dans ce tableau nous retrouvons tous les composants nécessaires à la réalisation du projet ainsi que leurs prix

Composant	Quantité	Prix unitaire HT (€)	Total (€)
Arduino	1	22,82 €	22,82 €
ARD Shield L298P	1	11,70 €	11,70 €
AD620B	1	11,78 €	11,78 €
Capteur de température PT100	1	2,53 €	2,53 €
Résistance (R1,R2,R4,Rg)	4	0,01 €	0,04 €
Servomoteur DF9GMS	1	4,44 €	4,44 €
Télémètre SHARP GP2Y0A21YK0F	1	10,80 €	10,80 €
Total (HT)	57,14 €		
TVA	19,60%		
Total (TTC)	68,34 €		

D'après le tableau ci-dessus pour réaliser le projet nous en avons pour un total de 68,34 €

Conclusion finale :

Malgré le bon fonctionnement de toutes les parties, nous avons eu quelques soucis au niveau de l'assemblage des parties pour mettre en évidence le scénario du sujet.
Nous n'avons aussi pas pu afficher la température sur un écran LCD car nous n'avons pas eu le temps

Tout au long du projet nous avons rencontré plusieurs problèmes liés au matériel, comme par exemple la carte Arduino ou le Shield que nous avons dû changer au moins 3 fois durant la réalisation du projet.

Cependant, grâce à cet SAE nous avons appris beaucoup de choses concernant la partie commande et la partie puissance d'un système et ainsi comment associer ces 2 parties