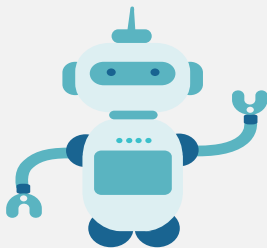




Présentation du projet CAO

Self balancing robot



Realisé par : MSEIBI Souha

CHABAANI Elaa

JABRI Chaima

JLASSI Wejden

Professeur : Mme. BOUBAKR Olfa

A person is riding a motorized wheelchair on a concrete sidewalk in a suburban neighborhood. A large, light-colored dog is walking alongside the wheelchair. The background features several houses with lawns and trees with autumn foliage under a clear blue sky. The text "Standing Mode" is overlaid in a large, white, serif font across the center of the image.

Standing Mode

clideo.com

Plan

Modelisation
Mathématique



Conception
électrique



Conception
mécanique



Programmation

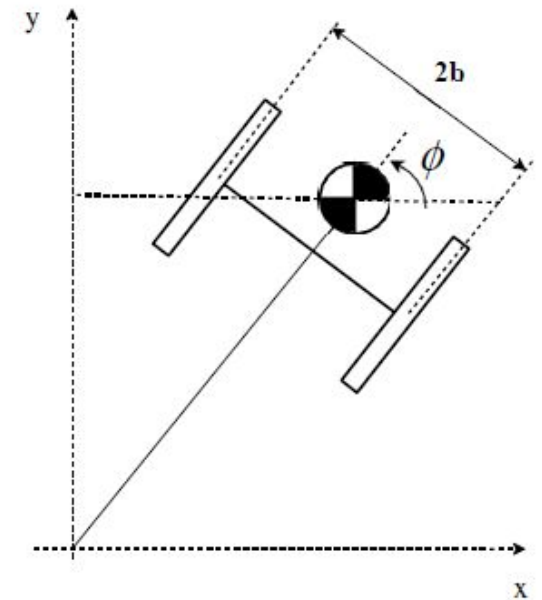
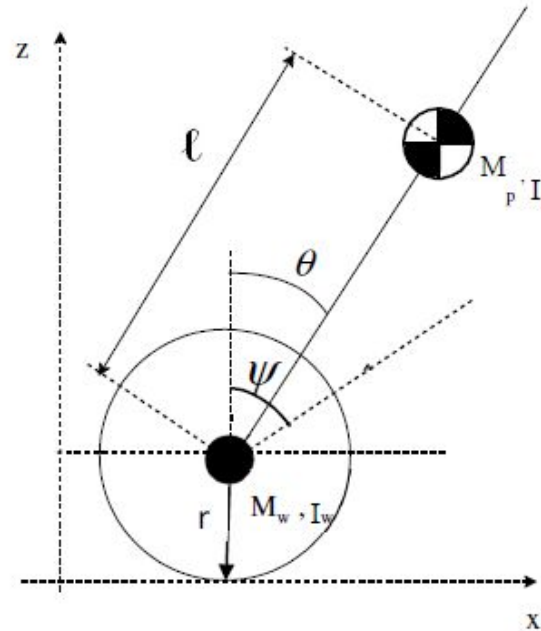


Conclusion et
perspectives

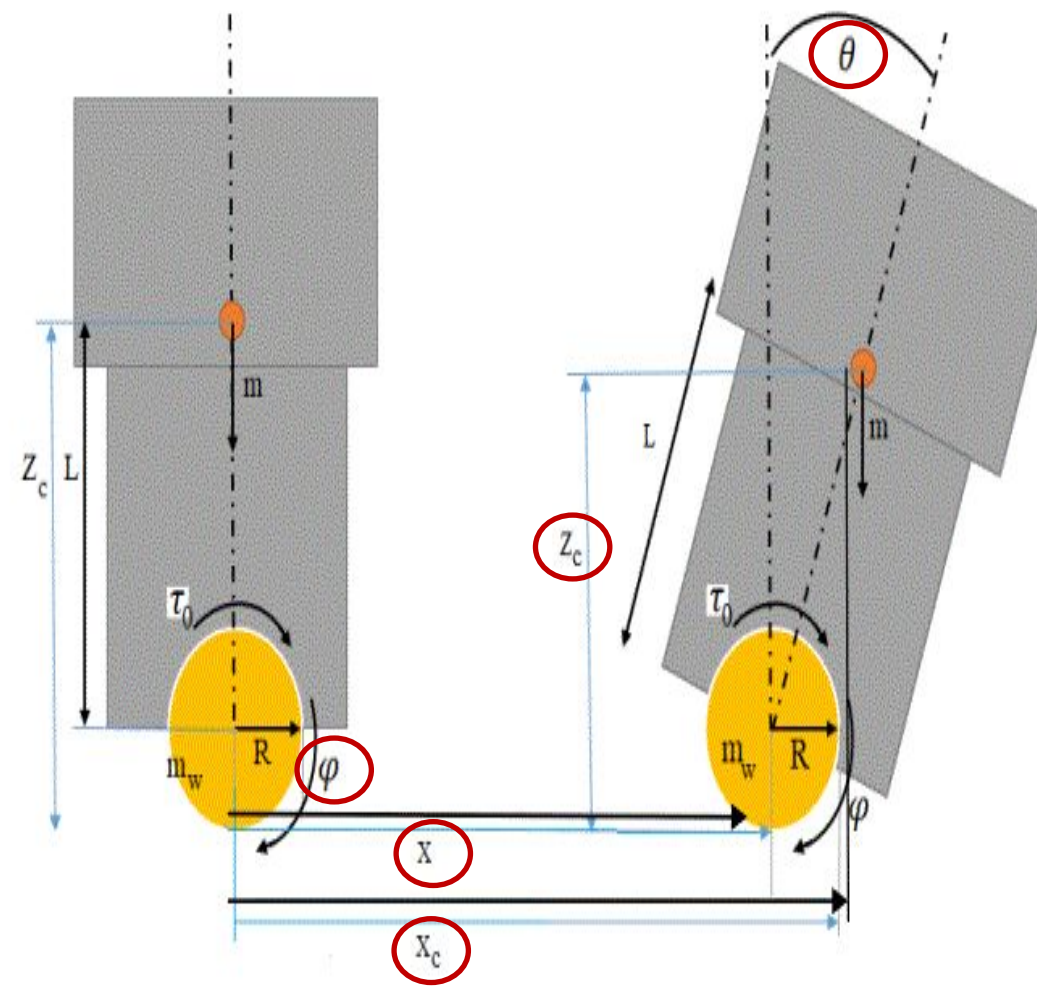


Modélisation et équations mathématiques :

- on peut modéliser le robot en un sous-système mécanique qui se compose d corps du robot et de deux roues.
- Le corps peut être modélisé comme un pendule inversé avec la masse concentrée au centre de gravité et l'axe de rotation au-dessus de l'axe de roues.

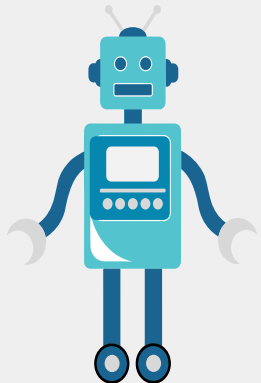


Assignations des variables



X	position horizontale du centre de la roue par rapport à une origine définie
Xc	position horizontale du point m de la partie du corps par rapport à une origine définie
φ	angle de rotation de la roue dans le sens des aiguilles d'une montre par rapport à l'axe horizontal à $t = 0$
z_c	position verticale du corps de la partie du corps du sol
θ	angle de rotation dans le sens des aiguilles d'une montre de la partie du corps à partir de la position verticale
M	masse de la partie du corps
m_w	masse de la roue
R	Rayon de la roue
L	longueur entre le corps et le centre de la roue
τ_0	couple appliqué
I	l'inertie de la partie du corps
I_w	inertie de la roue

- Dans ce système, nous supposons que le robot se déplace horizontalement sans glisser entre le sol et les roues
- La principale différence entre ce système et le pendule inversé est l'inertie et la position du centre de masse de la partie du corps, car nous cherchons un modèle plus détaillé du système plutôt que de l'approcher comme une masse ponctuelle.



Les équations dynamique

$$x = R \phi$$

$$x_c = R \phi + L \sin \theta$$

$$z_c = R + L \cos \theta$$

Dérivation du lagrangien $L=E_c-E_p$



$$L=E_c-E_p$$

$$= \frac{1}{2} (I_w + m_w R^2 + m R^2) \dot{\phi}^2 + m R L \cos \theta \dot{\phi} \dot{\theta} + \frac{1}{2} (I + m L^2) \dot{\theta}^2 - mgL (\cos \theta - 1)$$

Nous pouvons réécrire ces équations dynamiques non linéaires dans un style de matrice de second ordre comme suit :

$$\begin{pmatrix} I_w + m_w R^2 + m R^2 & m R L \cos \theta \\ m R L \cos \theta & I + m L^2 \end{pmatrix} \begin{pmatrix} \ddot{\phi} \\ \ddot{\theta} \end{pmatrix} + \begin{pmatrix} 0 & -m R L \sin \theta \dot{\theta} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \end{pmatrix} + \begin{pmatrix} 0 \\ -m g L \sin \theta \end{pmatrix} = \begin{pmatrix} \mu \\ \chi \end{pmatrix}$$

01: Modélisation
mathématique

02: Conception
électrique

03: Conception
mécanique

04: Programmation

05: Conclusion et
perspectives

Conception électrique :

Dimensionnement Matériels

composants électronique et
dispositifs nécessaires pour la
réalisation et la conception de
notre projet

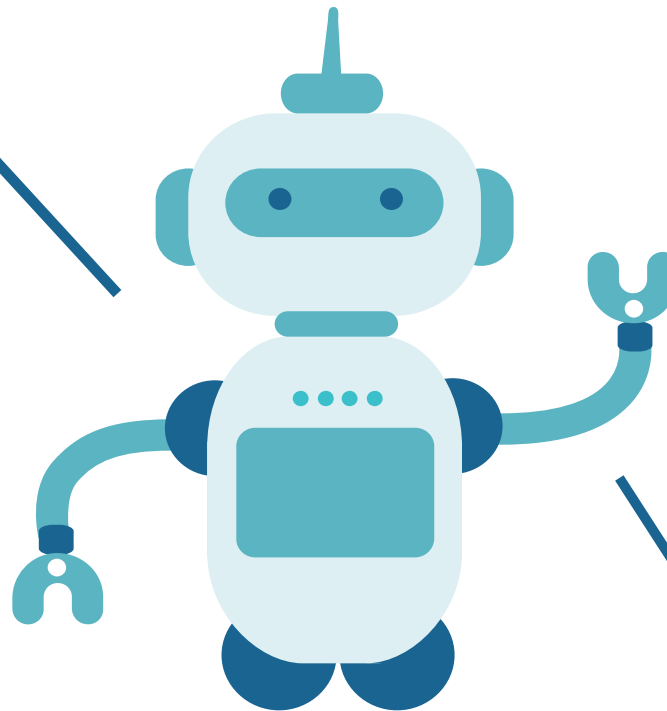


Schéma de câblage

comment relier les composants entre
eux

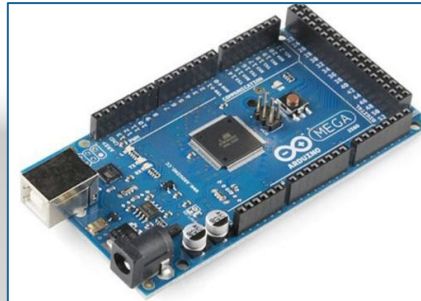
Dimensionnement Matériels

Microcontrôleurs



- Moins couteuse
- Disponibilité matériel
- Possède beaucoup de modules permettant de simplifier le travail

Arduino Uno



- Un grand nombre d'entrée/sortie
- Puissance de calcul faible
- Encombrement

Arduino Mega



- Plus professionnel
- Plus puissante
- elle peut fonctionner en temps réel avec un OS embarqué STM32 F4

STM 32

Dimensionnement Matériels

Moteurs



- Cout faible
- Petite taille et installation facile
- La complexité de mettre en œuvre (asservissement ,réglage PID)

Moteur DC 6V



- contrôle de vitesse précis
- Plus de précision
- Garantir la stabilité suite a une perturbation externe
- Cout élevé

Moteur DC avec encodeur

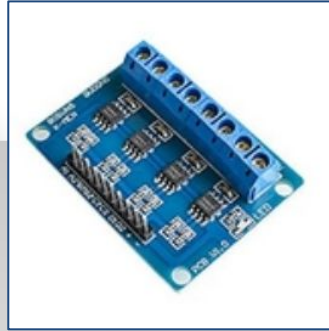


- Plus précis
- plus lourd,
- plus volumineux
- il consommera plus d'énergie

Moteur pas à pas

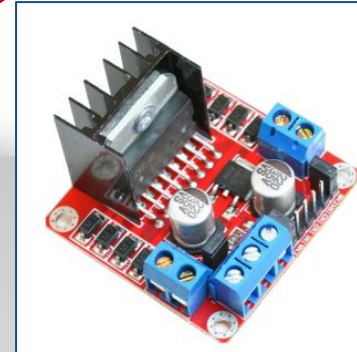
Dimensionnement Matériels

Carte de puissance



- Courant de commutation max. : 10 A
- Tension de coupure 250 V/AC, 30 V/DC max.

Carte relais



- Cout faible
- Il peut délivrer jusqu'à 2A en pointe et 20W en continu.

Pont en H

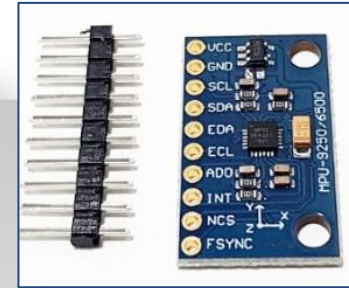
Dimensionnement Matériels

Capteur



- un gyroscope 3 axes et un accéléromètre 3 axes
- Cout faible

MPU 6050

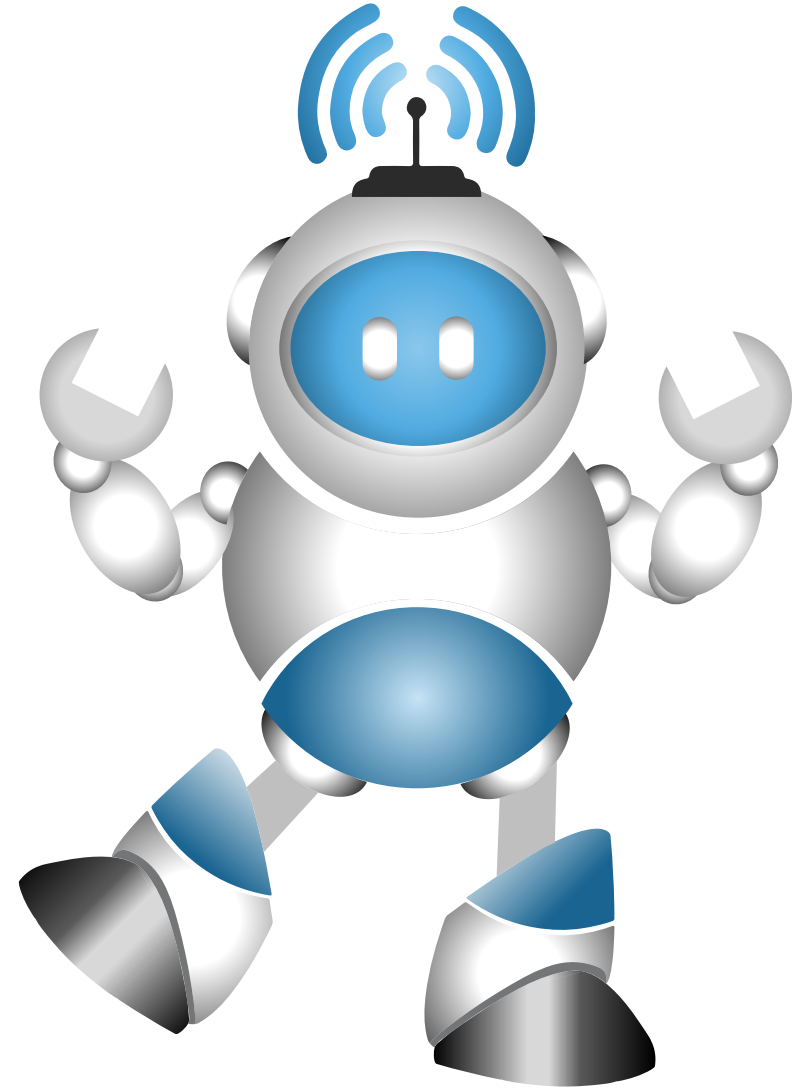


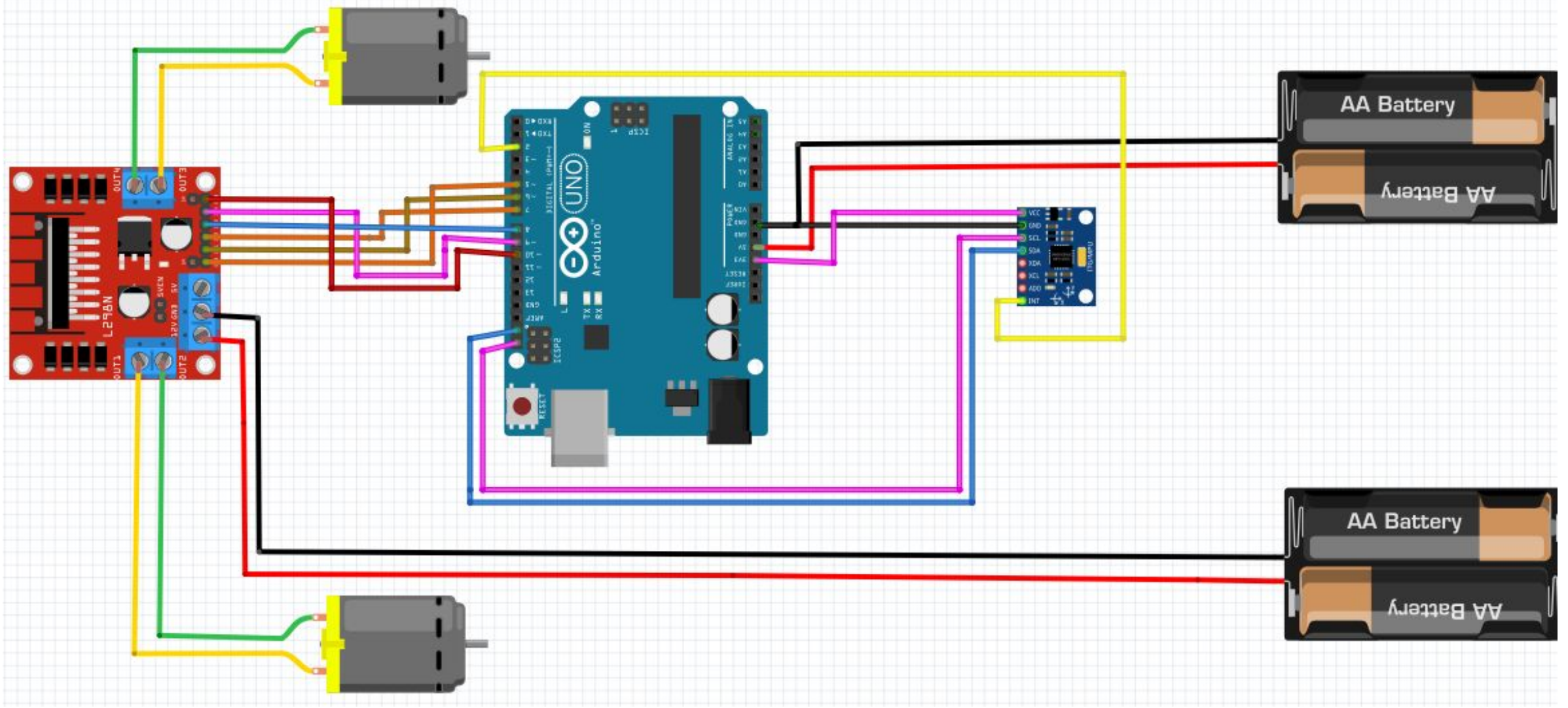
- 9 axes qui combine un gyroscope à 3 axes, un accéléromètre à 3 axes, un magnétomètre à 3 axes

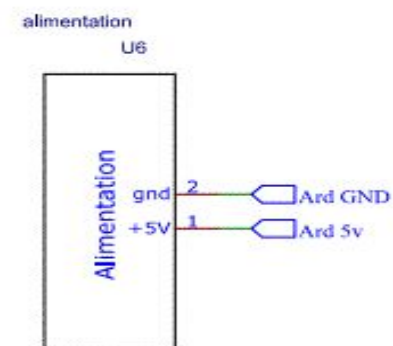
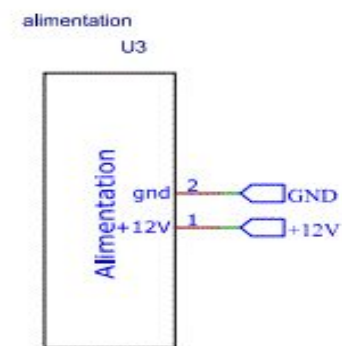
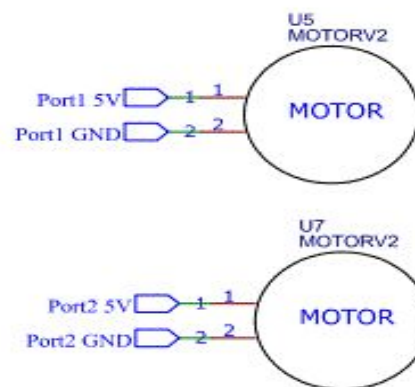
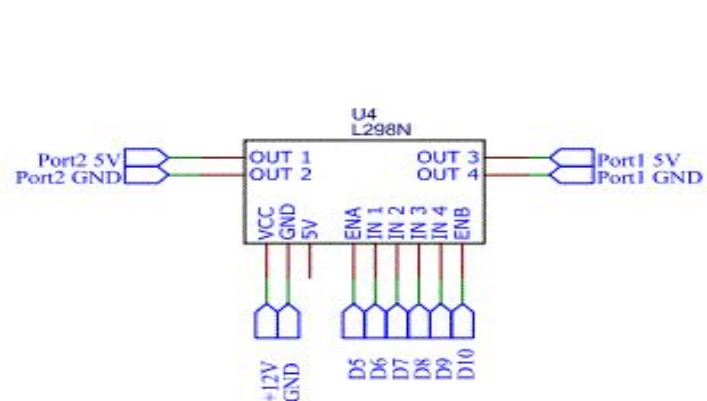
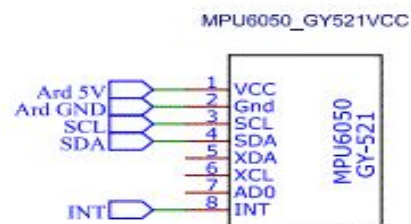
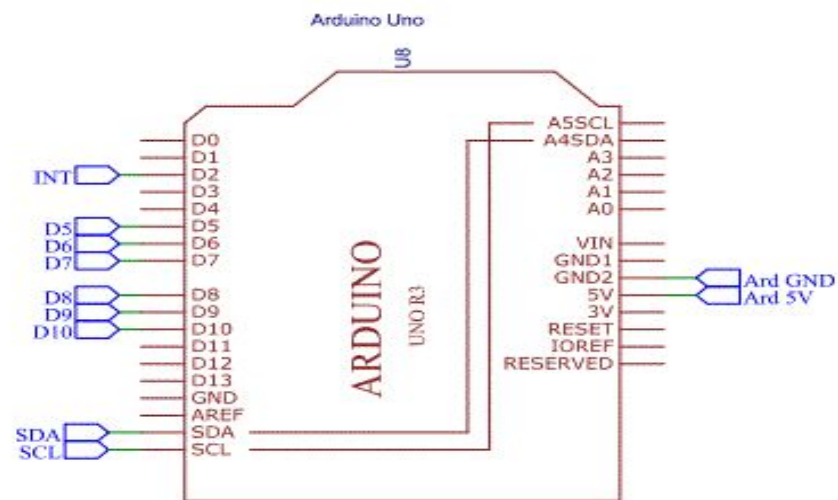
MPU 9250

Schéma de câblage

- Conception avec Fritizing
- Schéma synoptique







01: Modélisation
mathématique

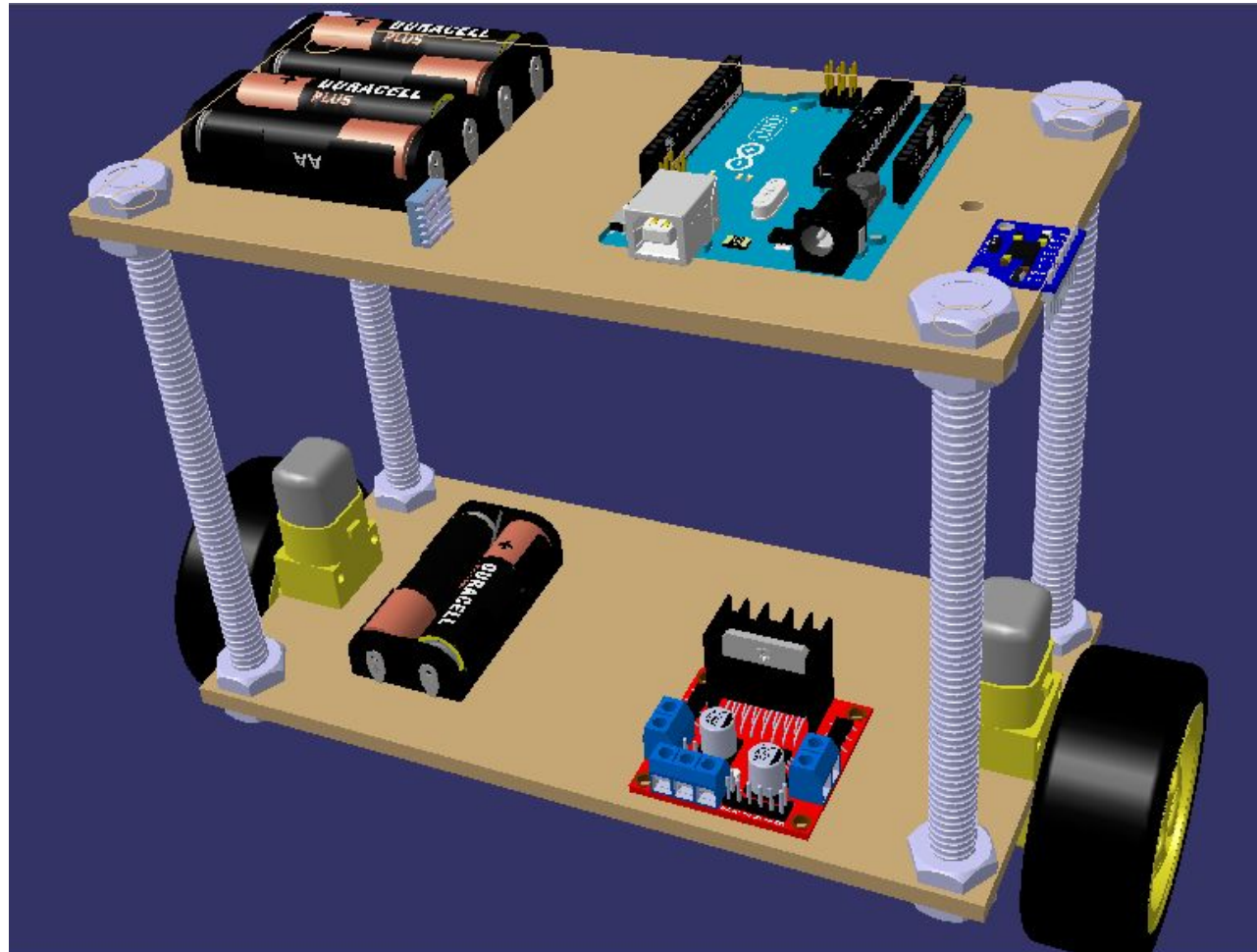
02: Conception
électrique

03: Conception
mécanique

04: Programmation

05: Conclusion et
perspectives

Conception Mécanique de la Version finale de robot :



Vue éclatée

Vue isométrique
Echelle : 1:4

Nomenclature :
Product1

N°	Référence	Type
1	Part1	Pièce
2	Tige	Pièce
3	Part2	Pièce
4	2 x AA Battery Holder	Assemblage
5	Arduino Uno	Assemblage
6	motoréducteur	Pièce
7	roue	Pièce
8	L298N Driver DC Motor, Stepper Motor	Pièce
9	Ecrou	Pièce
10	ITG-MPU	Assemblage
11	regulateur	Pièce

DESIGNED BY:
Chaima Jabri

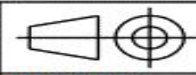
DATE:
24/04/2022

CHECKED BY:
XXX

DATE:
XXX

SIZE
A4

SCALE
1:1



WEIGHT (kg)
XXX

DRAWING NUMBER

Assembly

SHEET

1 / 1

This drawing is our property; it can't be reproduced or communicated without our written agreement.

Self Balancing Robot

DASSAULT SYSTEMES

Vue isométrique
Echelle : 1:4

**01: Modélisation
mathématique**

**02: Conception
électrique**

**03: Conception
mécanique**

**04:
Programmation**

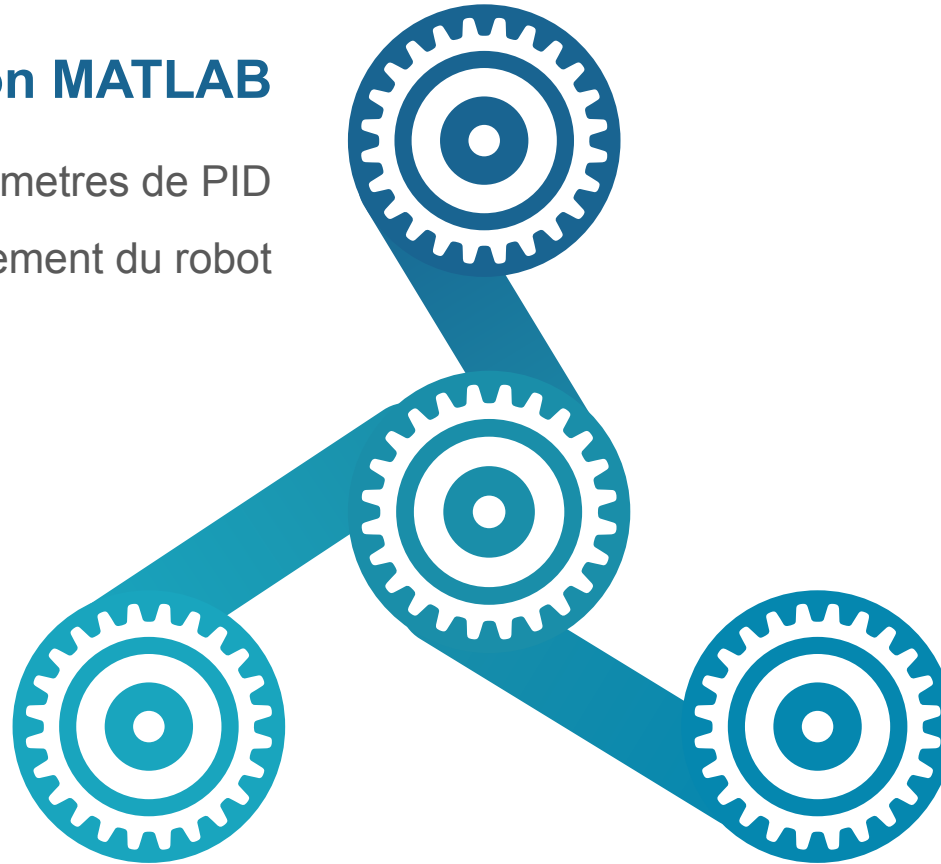
**05: Conclusion et
perspectives**

1- Simulation MATLAB

- Déterminer les paramètres de PID
- Simuler le fonctionnement du robot

2- Code ARDUINO

- Organigramme de
fonctionnement du robot

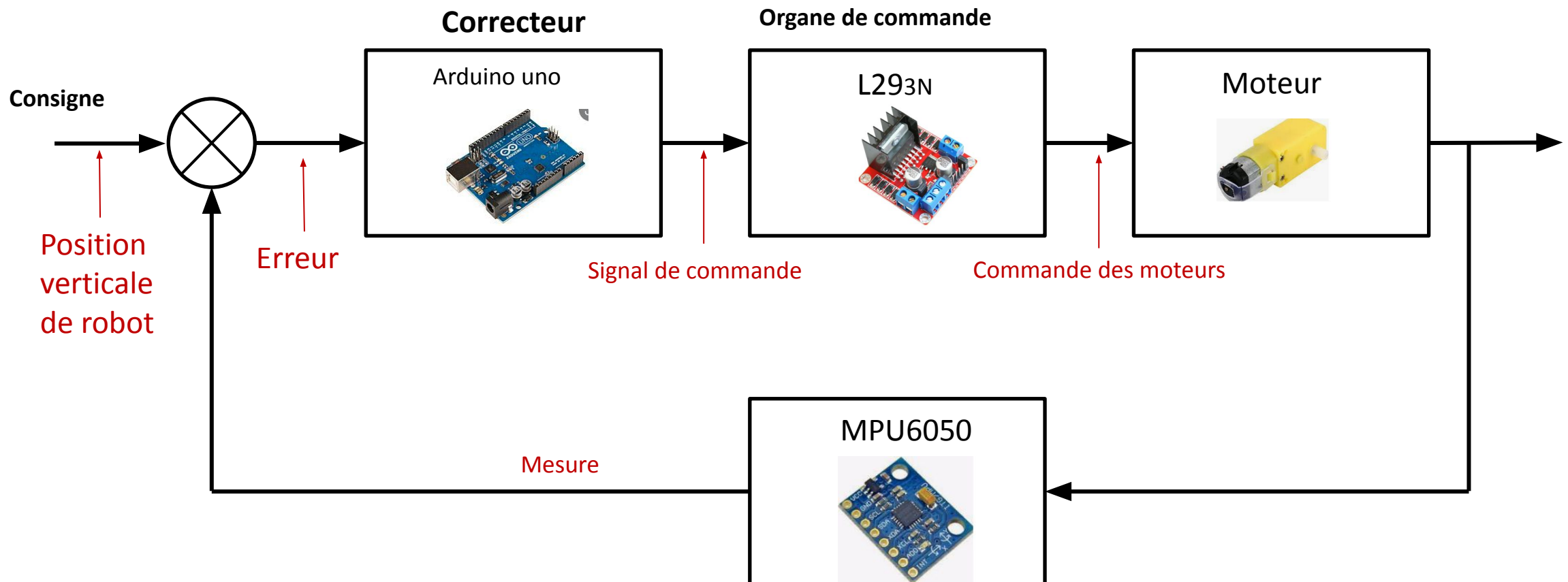


3- Interface LABVIEW

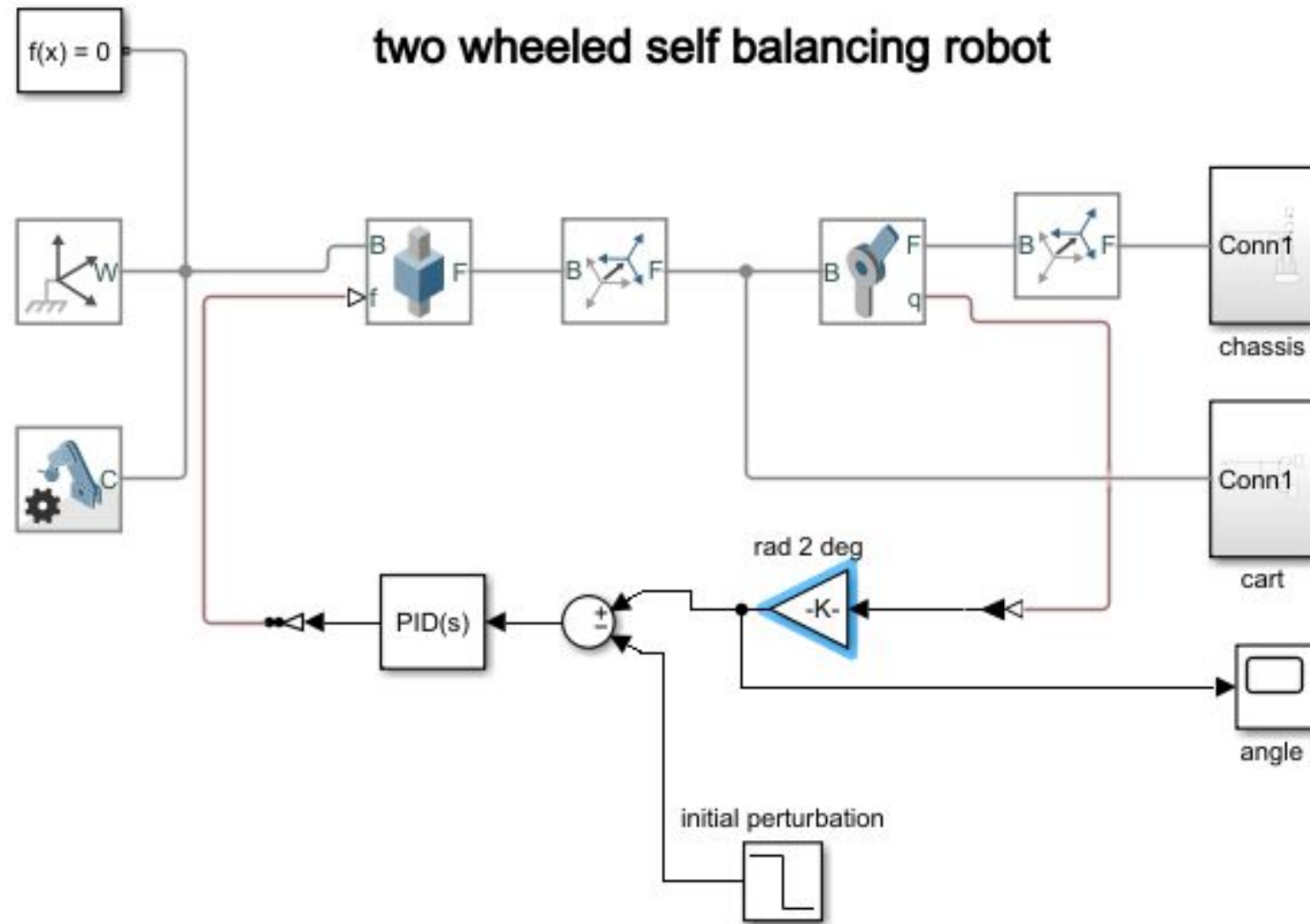
- Afficher les valeurs des
positions et de l'accélération
par rapport aux 3 axes

Régulateur PID

Schéma bloc Régulateur PID :



1- Simulation MATLAB



File Explorer Simulation View Tools Help



MENU AND TOOLBARS

D: > Polyspace > R2021a > bin >

Mechanics Explorers - Mechanics Explorer-SBRSIMULATION

Mechanics Explorer-SBRSIMULATION

- SBRSIMULATION
 - cart
 - chassis
 - Mechanism_Configuration
 - World_Frame
 - Prismatic_Joint
 - Revolute_Joint
 - Rigid_Transform1
 - Rigid_Transform2
 - Connection Frames

Simulation controls: Play, Stop, Step Back, Step Forward, Time range: T = [0,10], Scale: 1X, Time: 10

Command Window

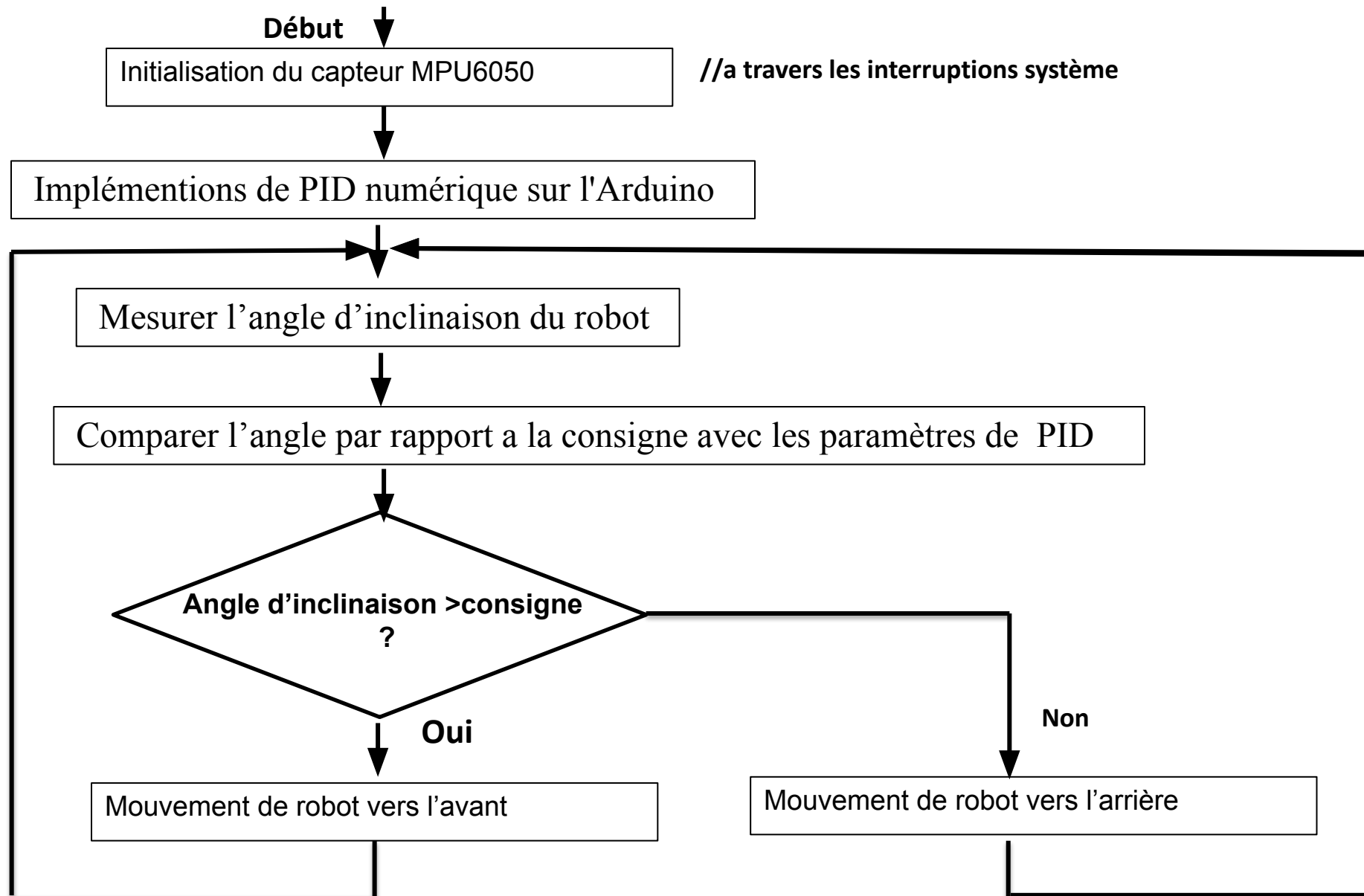
New to MATLAB? See resources for [Getting Started](#).

```
fx >>
```

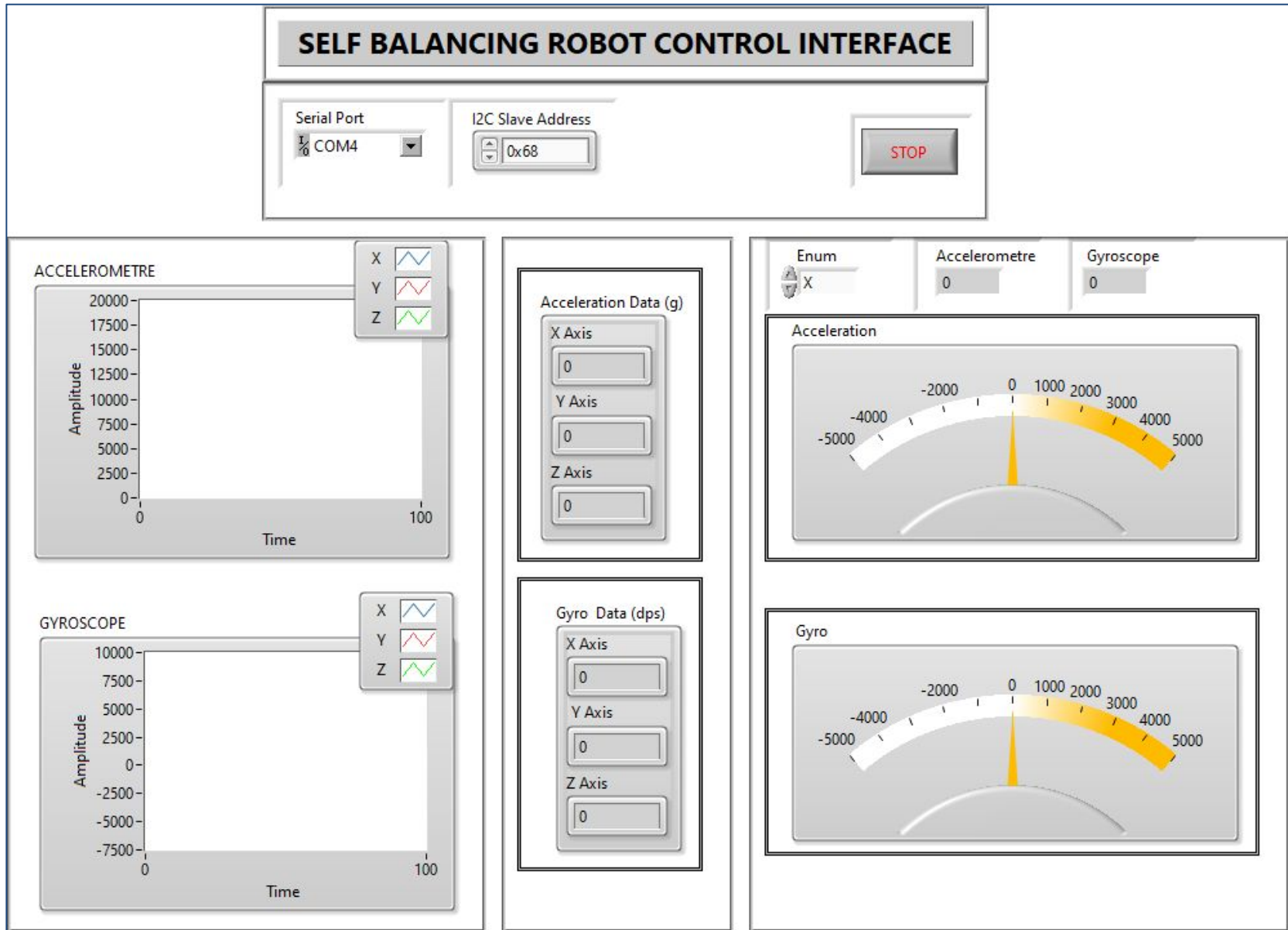
0:00:03 0:00:12

Ready

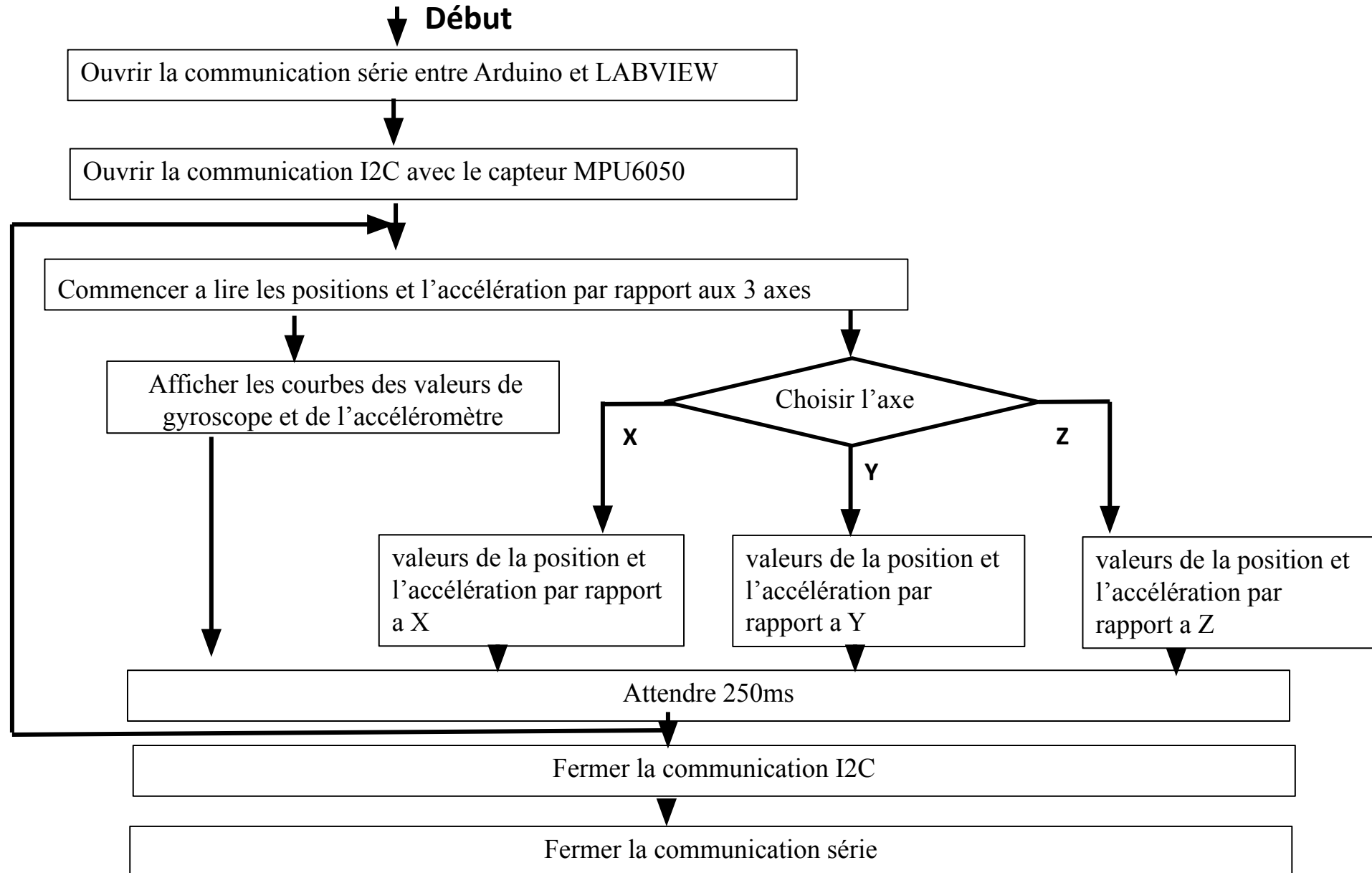
2- Code ARDUINO: organigramme de fonctionnement de robot auto balance



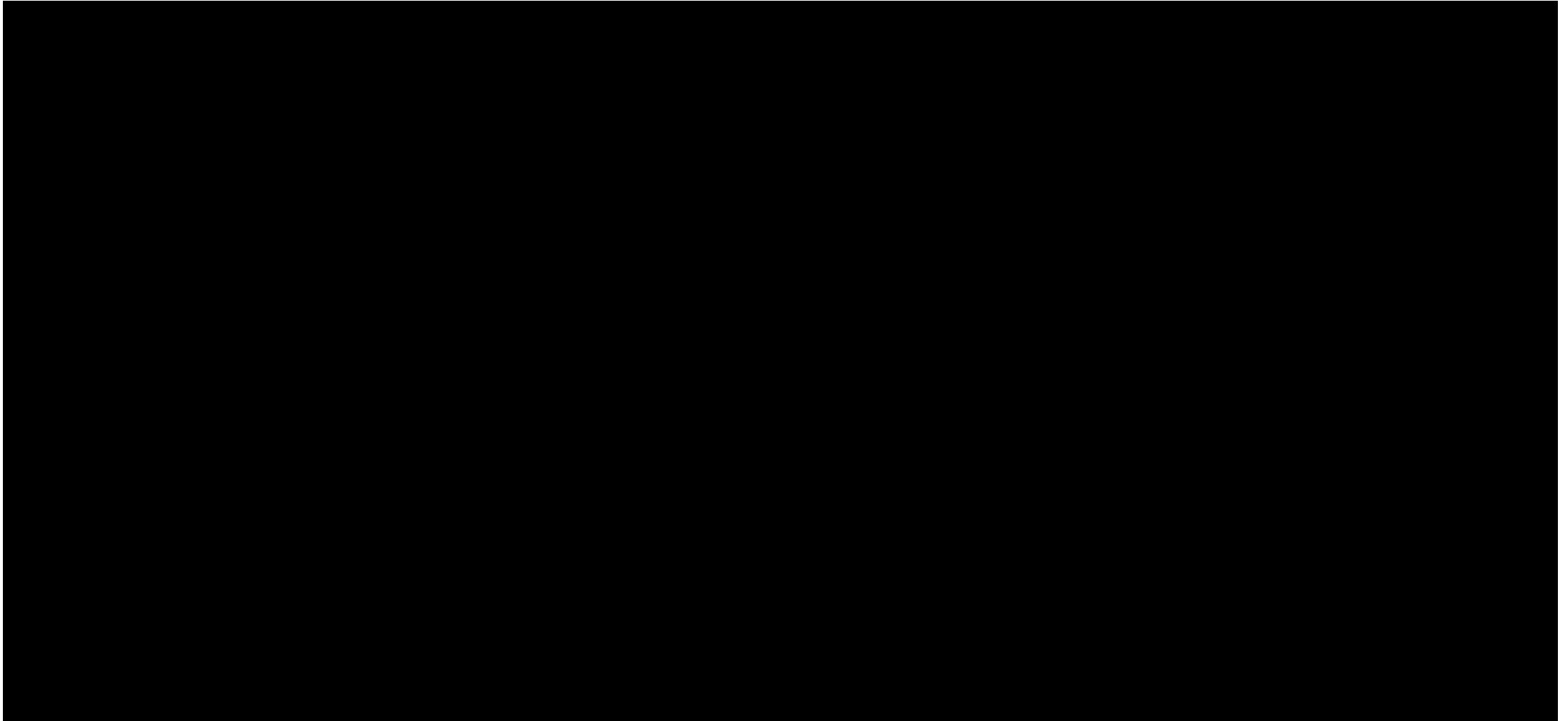
3- Interface LABVIEW



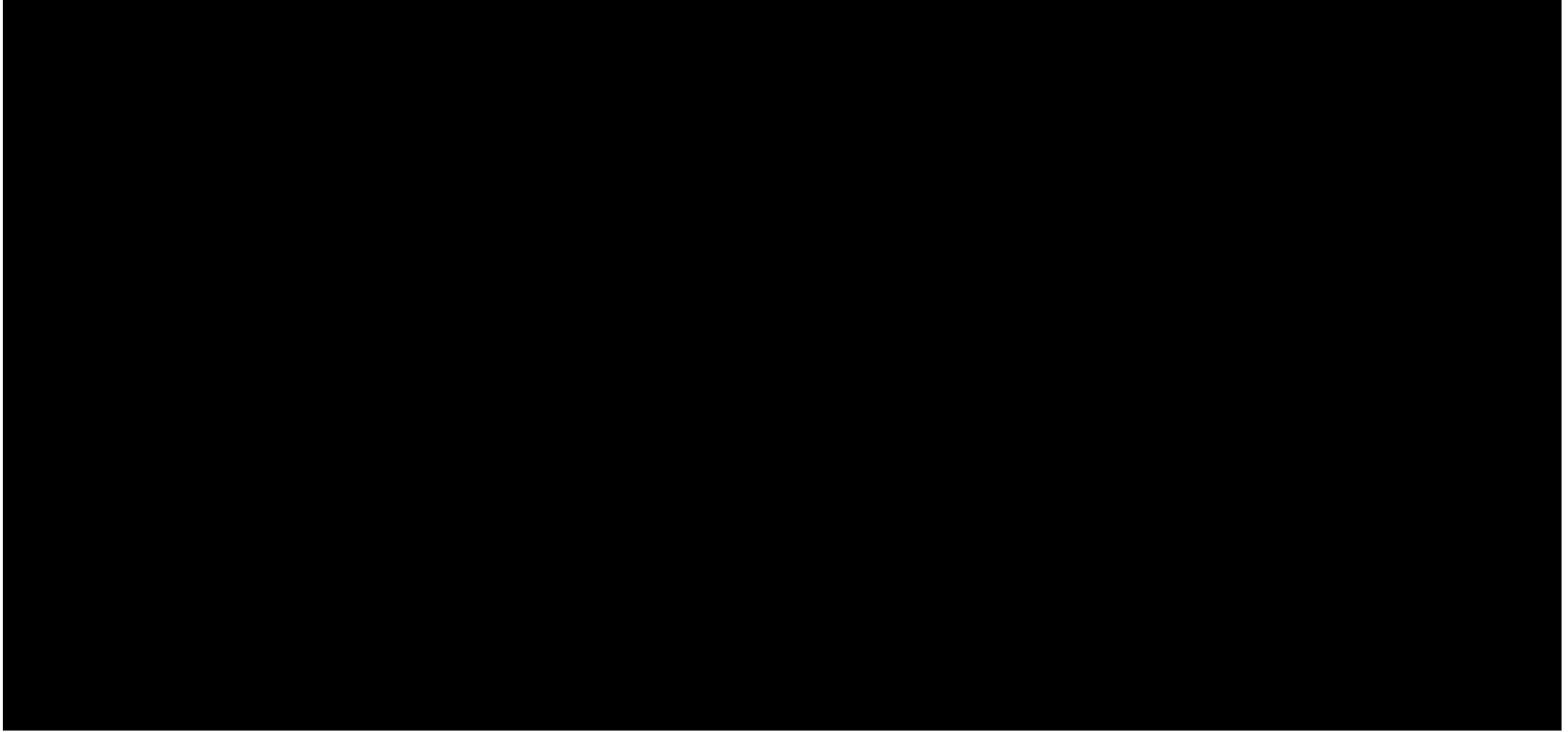
Code :organigramme interface LABVIEW



Vidéo simulation de fonctionnement



Vidéo simulation de fonctionnement



Etude technico économique

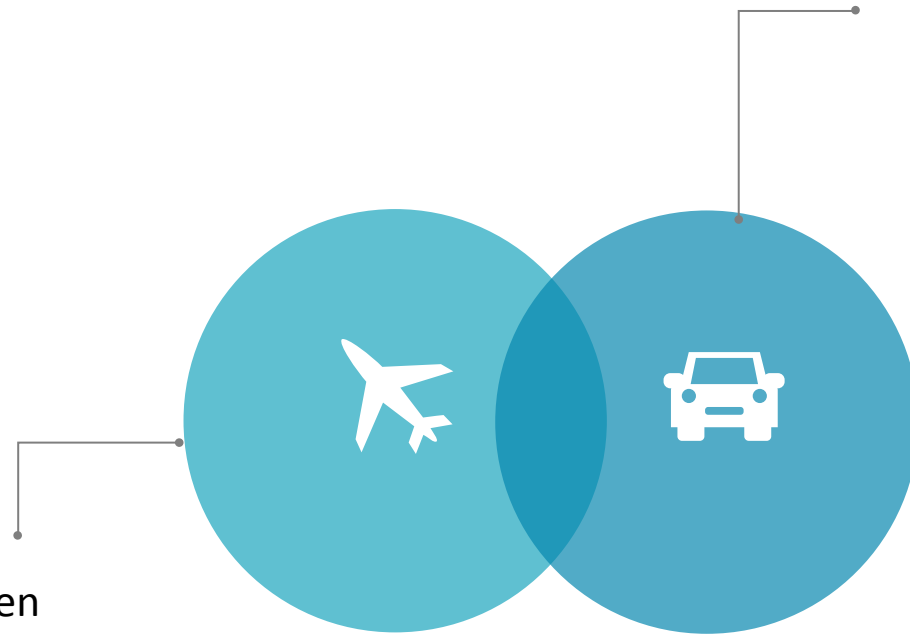
Matériels	Référence	Cout
ARUIONO UNO	0	0
Moteurs+Roues	6VMOTEUR DC 3-	9 DT*2 =18 DT
Module Gyroscope accéléromètre	MPU6050 - GY521 ACCELEROMETRE ET GYROSCOPE 3 AXES	14.000 DT
Pont en H	L298N DC Motor Driver Module	12.800 DT
Batteries	Batterie alimentation moteurs	0
	1 PACK 4 PILES alimentation Arduino	4.500 DT
Coupleur-pile	COUPLEUR PILES 4*AA AVEC CONNECTEUR JACK 5.5	4.800 DT
Cout total	54100 DT	
Cout individuel	13.525 DT	



Conclusion

Nous avons commencé notre étude par une modélisation mathématique du système , par la suite nous avons travaillé sur la conception électrique du robot : dimensionnement matériels et schéma de câblage .
nous avons étudié la conception mécanique : les dimensions ainsi que le positionnement des matériels afin de garantir la stabilité du robot et finalement nous avons développé le code nécessaire au fonctionnement et l'interface de contrôle avec Labview

Perspective



Cette idée peut être développer en ajoutant un capteur de détection d'obstacle comme le capteur sonore , un contrôle par application android

Durant la réalisation de ce projet , on n'a pas réussi à bien équilibré le robot et le réguler , à cause de mauvais fonctionnement de l'accéléromètre MPU , même avec la calibration de la sensibilité du capteur on n'a pas réussi à établir la connection entre la carte arduino et le capteur , du coup , des valeurs erronées étant rendu par le MPU . ça peut être expliqué par un défaut au pin de clock ou serial data .