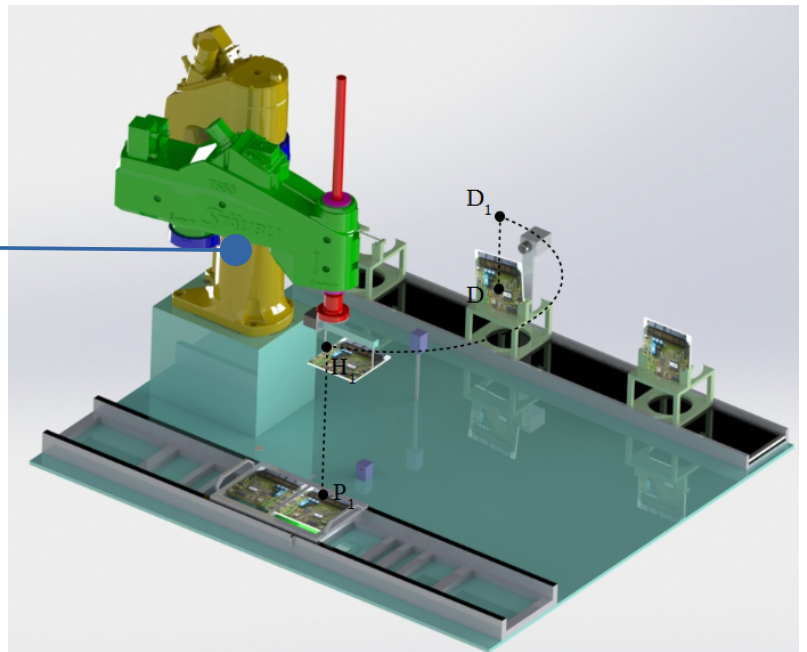


Document ressource : Expérimentation Robot SCARA

0. Contextualisation

Dans beaucoup de chaînes de production de nombreuses tâches de manutention de composants sont assurées par des robots. Par exemple sur la chaîne de production de l'entreprise Bosch chargée de la réalisation des calculateurs d'injection (EPA) une tâche de transfert de composant est assurée par un robot de type SCARA ()



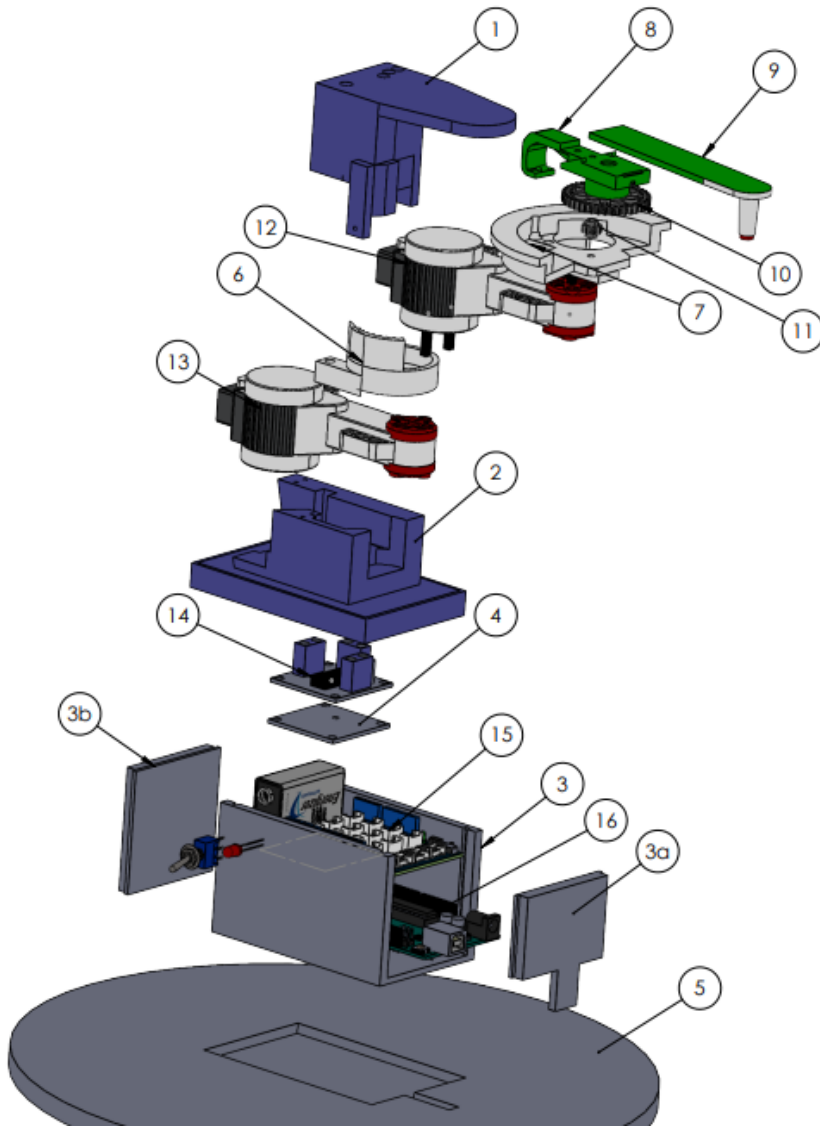
doc1 : Schéma d'ensemble de la chaîne de production

Afin que le takt time de la ligne de production soit respecté il est impératif que le robot effectue cette tâche de transfert avec une rapidité et une précision suffisante. En effet la vitesse de déplacement moyenne doit être à minima de $0,225 \text{ m.s}^{-1}$ et la zone de préhension est comprise entre deux cercles concentriques de rayon 10cm et 256cm centrés sur le robot. Evidemment il sera nécessaire de permettre des opérations de maintenance hebdomadaire et respecter les normes de sécurité en dotant le robot d'une cage.

Le but de cette activité est d'étudier la cinématique et la commande d'une maquette didactisée de ce système.

1. Description de la maquette

1.A) Les éléments constitutifs de la maquette sont les suivants :



Nomenclature :

1:Châssis haut
 2:Châssis bas
 3:Boîtier de contrôle
 3a:Couvercle 1
 3b:Couvercle 2
 4:Support de circuit
 5:Base
 6:connecteur
 7:Indexeur
 8:Bras arrière
 9:bras avant

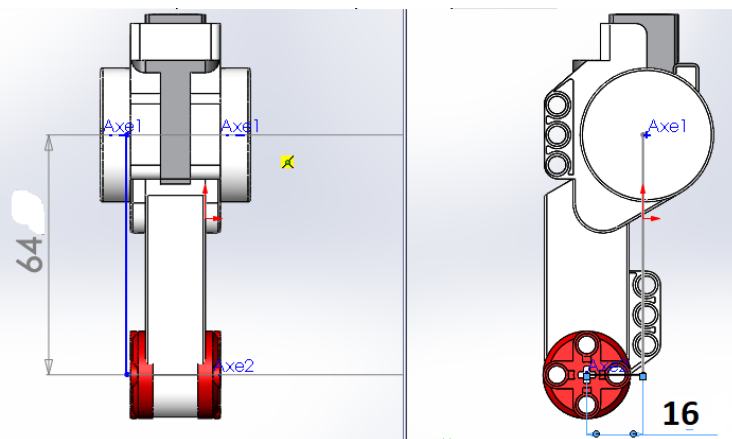
10:Roue dentée 40 dents
 11:Roue dentée 8 dents

12:Moteur2 NTX
 13:Moteur1 NTX

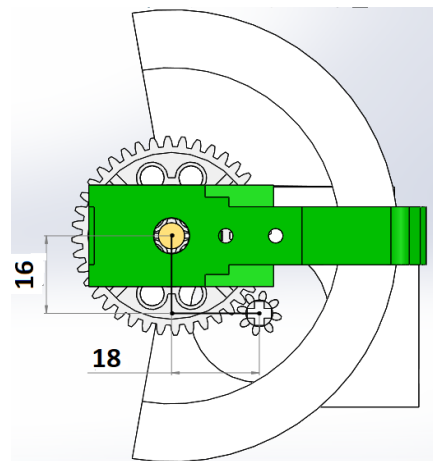
14:Carte l298d
 15:Arduino grove
 16:Arduino uno

doc2 : Schéma éclaté de la maquette

Le système est donc identifiable à un bras bi-articulé :

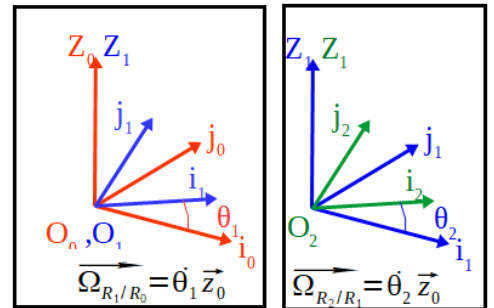
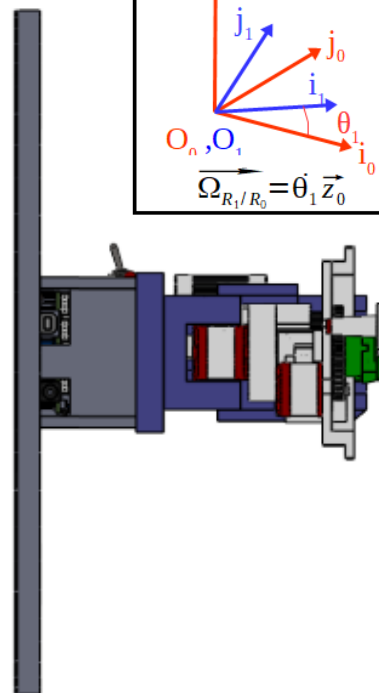
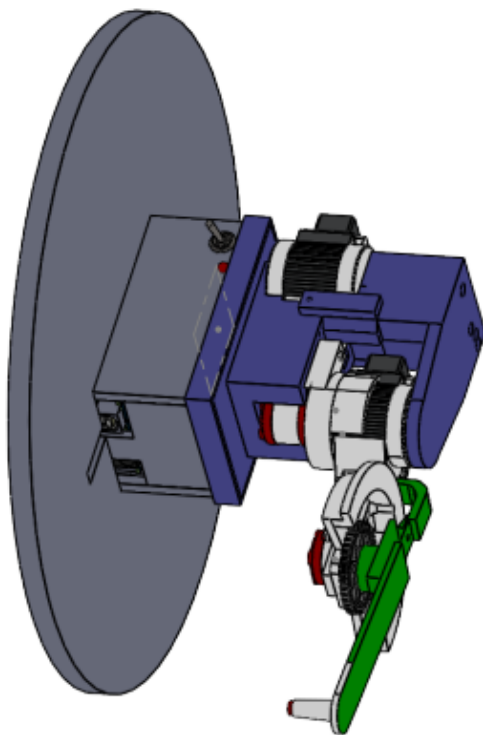
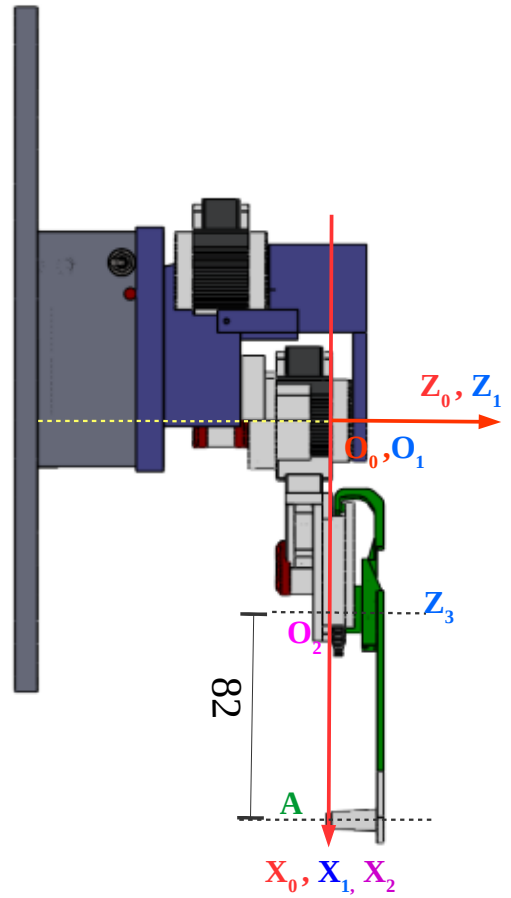
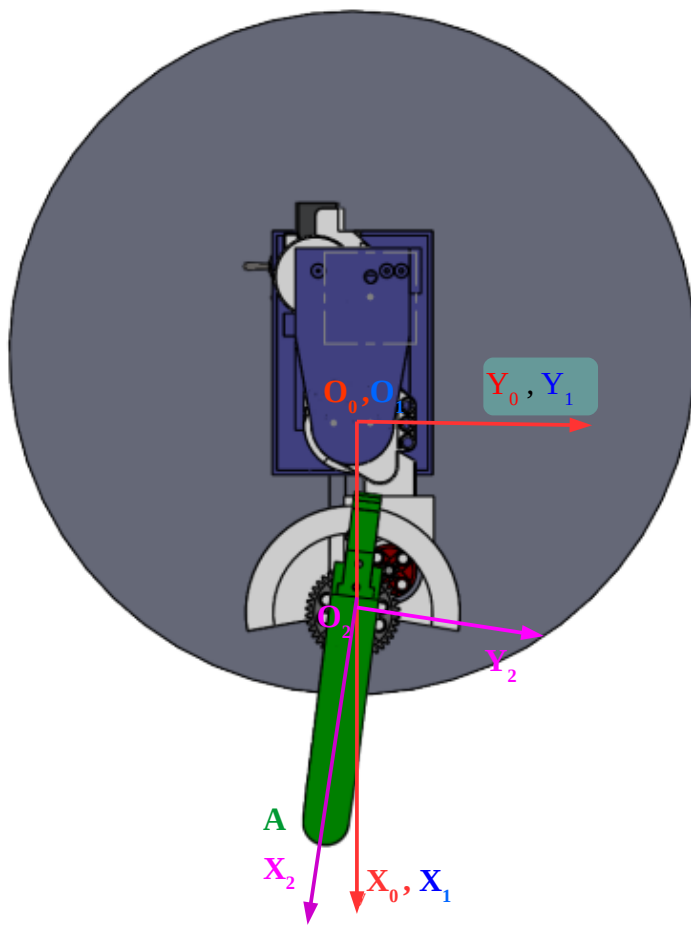


Et d'un indexeur associé à un pignon décentré



doc3 : Schéma du moteur et de la transmission

1.B) la maquette se paramètre géométriquement de la manière suivante :

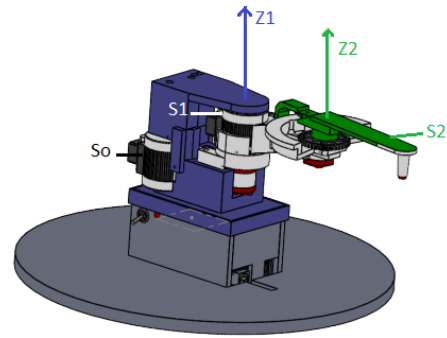


doc4 : Repérage cinématique

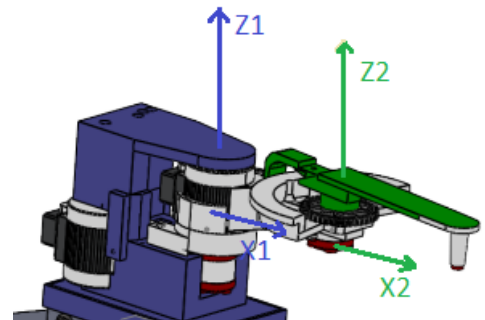
1.C) Rappel sur le repérage lié à l'écriture des matrices de changement de Base dans la convention de Denavit Hartenberg modifié (convention Khalil Kleinfinger)

1.C.a) MGD (Modèle Géométrique Direct)

1) On dessine les axes Z_i
 définissant les axes
 de liaisons entre les solide (i) et (i+1)



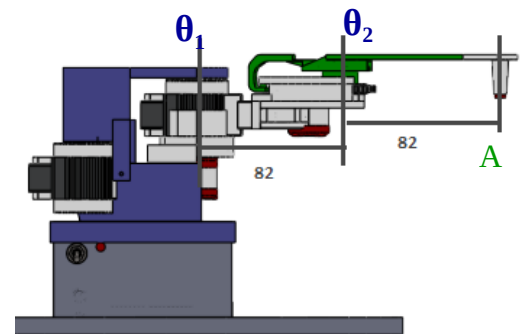
2) On dessine les axes X_i portés par les perpendiculaires communes entre Z_i et Z_{i+1} orientés de Z_i à Z_{i+1}
 Si les Z_i et Z_{i+1} sont concourants alors $\vec{X}_i = \vec{Z}_i \wedge \vec{Z}_{i+1}$



3) les axes Y se déterminent par : $\vec{Y}_i = \vec{Z}_i \wedge \vec{X}_i$

4) Positionnement des paramètres

Distance entre	Rotation autour
X_{i-1} et X_i suivant Axe Z_i r_i	Axe Z_i entre X_{i-1} et X_i θ_i
Z_{i-1} et Z_i suivant Axe X_{i-1} d_i	Axe X_{i-1} entre Z_{i-1} et Z_i α_i



5) Paramètres de Denavit Hartenberg modifié (convention Khalil Kleinfinger)

i	α_i	r_i	θ_i	d_i
1				
2				
3				

- $\alpha_1 : \mathcal{A}(z_{ref}, z_1)$ autour de x_{ref} ,
- $d_1 : \mathcal{D}(z_{ref}, z_1)$ suivant l'axe x_{ref} ,
- $\theta_1 : \mathcal{A}(x_{ref}, x_1)$ autour de z_1 ,
- $r_1 : \mathcal{D}(x_{ref}, x_1)$ suivant l'axe z_1 ,
- $\alpha_2 : \mathcal{A}(z_1, z_2)$ autour de x_1 ,
- $d_2 : \mathcal{D}(z_1, z_2)$ suivant l'axe x_1 ,
- $\theta_2 : \mathcal{A}(x_1, x_2)$ autour de z_2 ,
- $r_2 : \mathcal{D}(x_1, x_2)$ suivant l'axe z_2 ,

On remarquera que l'on cherche l'espace des poses correspondant au point A et qu'il est donc nécessaire de renseigner la ligne 3, en considérant le point A comme fixe dans R2

6) Matrice de Denavit Hartenberg modifié (convention Khalil Kleinfinger)

$$\mathfrak{P}_{n-1,n} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & d \\ \cos(\alpha) \sin(\theta) & \cos(\alpha) \cos(\theta) & -\sin(\alpha) & -r \sin(\alpha) \\ \sin(\alpha) \sin(\theta) & \sin(\alpha) \cos(\theta) & \cos(\alpha) & r \cos(\alpha) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

1.C.b) Détermination du MGI (Modèle Géométrique Inverse): Méthode

Soit un robot dont la matrice de transformation d'un repère R_0 à R_n se décompose en :

$$B_{0,n} = B_{0,1}(q_1)B_{1,2}(q_2)\dots B_{n-1,n}(q_n) \quad \text{dont la situation désirée est } U_0 = \begin{vmatrix} S_x & n_x & a_x & P_x \\ S_y & n_y & a_y & P_y \\ S_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

et on cherche à résoudre $U_0 = B_{0,1}(q_1)B_{1,2}(q_2)\dots B_{n-1,n}(q_n)$

La technique consiste à multiplier l'équation ci-dessus successivement par des matrices de passage inverse afin de se ramener au système suivant :

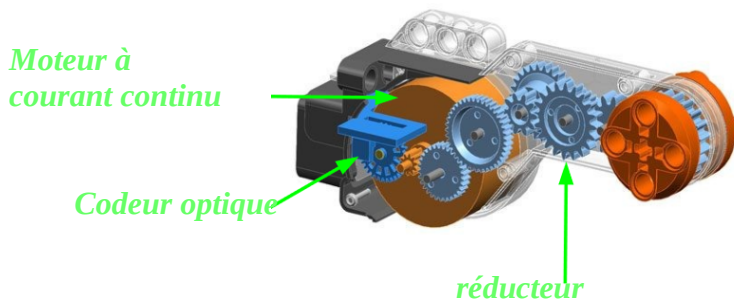
$$\begin{aligned} U_0 &= B_{0,1} B_{1,2} \dots B_{n-1,n} \\ B_{1,0} U_0 &= B_{1,2} \dots B_{n-1,n} \quad \text{terme } B_{1,0} U_0 \text{ ne dépend que de } U_0 \text{ et de } q_1, \text{ on peut donc expliciter } q_1 \\ B_{2,1} \cdot B_{1,0} U_0 &= \dots B_{n-1,n} \quad \text{terme } B_{2,1} \cdot B_{1,0} U_0 \text{ permet d'explicitier } q_2 \\ B_{n-1,n-2} \dots B_{2,1} \cdot B_{1,0} U_0 &= B_{n-1,n} \quad \text{ainsi de suite jusqu'à } q_n \end{aligned}$$

Rq : Il pourra être utile d'utiliser la matrice inverse

$$\mathfrak{P}_{n-1,n}^{-1} = \mathfrak{P}_{n,n-1} = \begin{pmatrix} \cos(\theta) & \cos(\alpha) \sin(\theta) & \sin(\alpha) \sin(\theta) & -d \cos(\theta) \\ -\sin(\theta) & \cos(\alpha) \cos(\theta) & \sin(\alpha) \cos(\theta) & d \sin(\theta) \\ 0 & -\sin(\alpha) & \cos(\alpha) & -r \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2. Motricité NXT

Chaque motoréducteur est composé de :

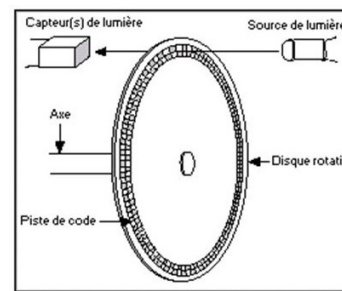
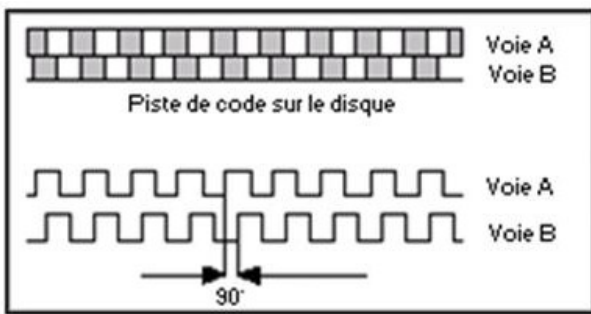


Donnée électrique du moteur cc
 $R = 7.2 \Omega$
 $L = 0.0048 \text{ H}$
 $K_e = 0.46$
 $K_m = 0.32$

doc5 : Moteur NXT

2..A) L'encodeur optique en quadrature :

L'encodeur en quadrature comporte deux pistes de code dont les secteurs sont décalés de 90 degrés d'une piste à l'autre; ceci permet d'avoir deux sorties de voie qui indiquent la position et la direction de la rotation. Si A devance B, par exemple, le disque tourne dans le sens des aiguilles d'une montre. Si B devance A, le disque tourne dans le sens contraire. Par conséquent, en surveillant à la fois le nombre d'impulsions et les phases relatives des signaux A et B, vous pouvez suivre la position et la direction de la rotation.



doc6 : Moteur NXT (source

Codeuse 24 imp/tr
En quadrature

la roue du codeur
 $Z_r = 32$ dents.



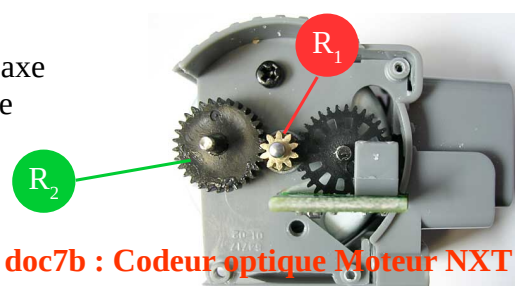
Le pignon moteur
 $Z_p = 10$ dents



doc7a : Codeur optique Moteur NXT

2.B) Le train d'engrenage revêt l'allure suivante :

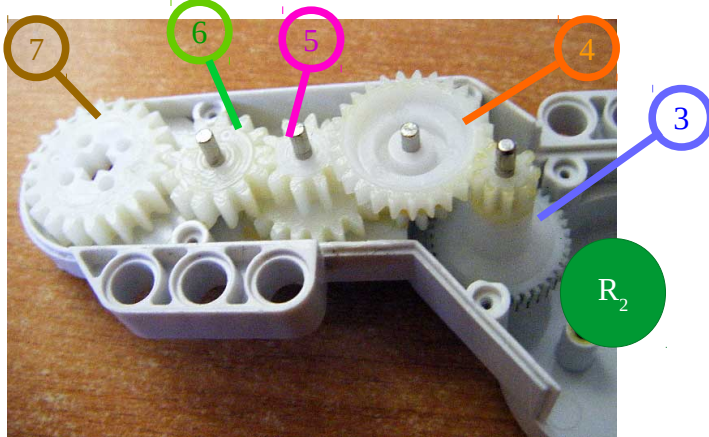
Un pignon R_1 de 10 dents est monté sur l'axe du moteur qui engrène sur une roue dentée R_2 de 30 dents



doc7b : Codeur optique Moteur NXT

La roue R_2 engrène sur la la denture Z_{3a} de la roue dentée 3

Caractéristiques

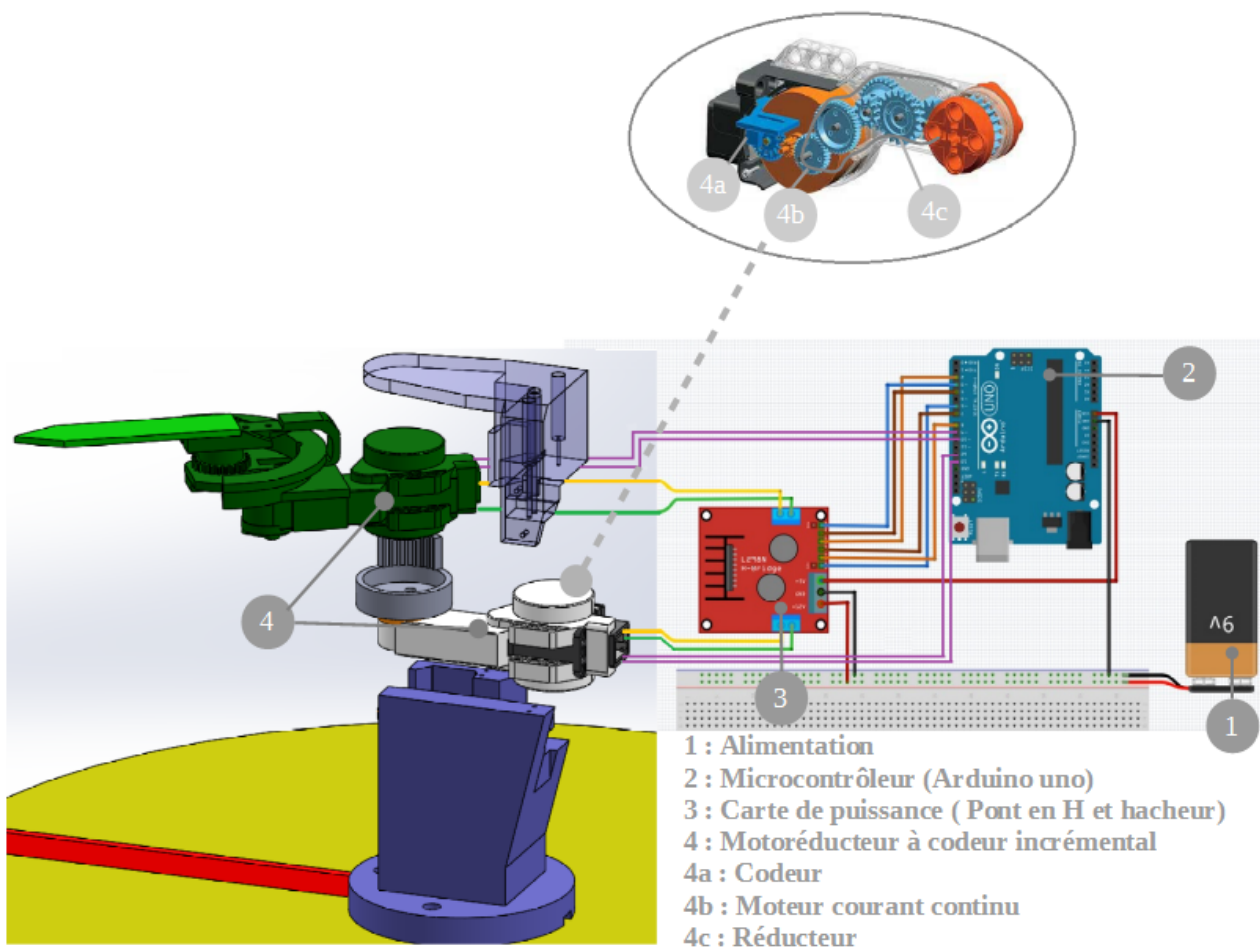


- 3 : $Z_{3a} = 40$
 $Z_{3b} = 9$
- 4 : $Z_{4a} = 27$
 $Z_{4b} = 10$
- 5 : $Z_{5a} = 20$
 $Z_{5b} = 10$
- 6 : $Z_6 = 13$
- 7 : $Z_7 = 20$

doc8 : Train engrenages Moteur NXT

3. Commande

La partie pilotage est quant à elle assurée par un arduino couplé à un interface de puissance permettant de gérer la vitesse et le sens de rotation d'un moteur à courant continu. Un programme « Scara » permet un pilotage direct ou par boucle de rétroaction



doc9 : Schéma de pilotage de la maquette

3.A) Les caractéristiques de l'interface de puissance sont :

Channel 1		Channel 2	
Name	Function	Name	Function
1A	Input A of ch.1 (TTL input)	2A	Input A of ch.2 (TTL input)
1B	Input B of ch.1 (TTL input)	2B	Input B of ch.2 (TTL input)
E1	Enable ch.1 (TTL input)	E2	Enable ch.2 (TTL input)

Tab.1 - Connections

To understand the meaning of these signals and their use you can read the following table (Tab.2), where all conditions are reported. Note that there are reported only the conditions for channel 1 because conditions for channel 2 are just alike them.

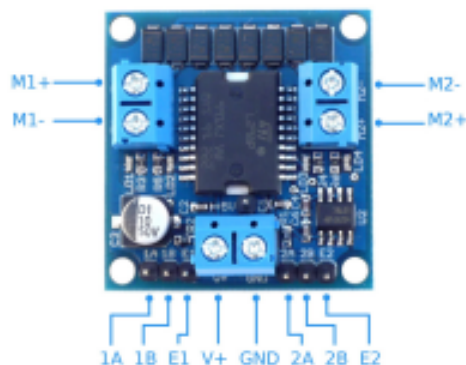
Inputs			M1+ and M1- output
E1	1A	1B	
1	1	1	HIGH state for both output (motor stopped)
1	0	0	LOW state for both output (motor stopped)
1	1	0	Current flows from M1+ to M1- (direction 1)
1	0	1	Current flows from M1- to M1+ (direction 2)
0	X	X	High impedance (motor is in free running)

Tab.2 - Conditions

About the *E1* and *E2* signals, they have pull-up resistors so in some applications you don't need to drive them and you need just 2 signals to control each motor.

SPECIFICATIONS

Supply voltage	7 - 30V
Supply current (logic)	24mA typ. (36mA max.)
Output current	4A (2A for each channel)
Data I/O voltage	TTL standard
Dimensions	37x36x13mm (connectors included)
Weight	12.5g / 0.44oz
Operating temperature	-25 - 130°C



doc10 : Datasheet contrôleur double DC L298N

3.B) Les principes de commande

En commande directe la consigne est l'angle d'arrivé

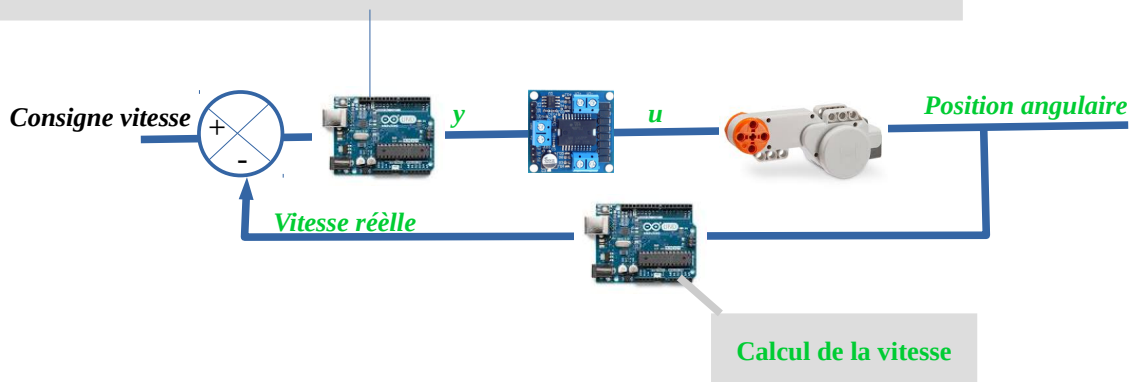


doc11 : Commande directe

Le problème de ce type de pilotage est qu'il ne tient pas compte de la réalité du déplacement, et a peu de chance de réaliser un déplacement précis. On lui préfère un pilotage rétroactif

En commande rétroactive la consigne est la différence entre l'angle d'arrivé et la position réelle

Saisir la consigne angulaire
Calcul de l'erreur



doc12 : Commande par rétroaction

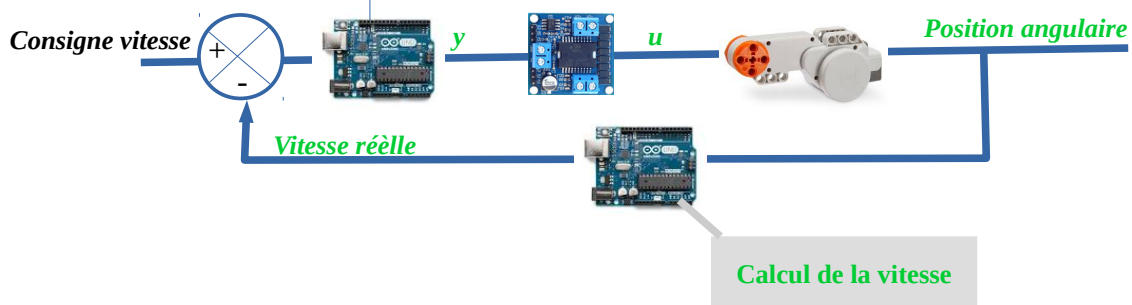
On peut améliorer cette commande en lui ajoutant un correcteur qui analysera les erreurs au fur et à mesure du déplacement. Ce correcteur se nomme PID :

P proportionnel : il s'agit d'analyser la somme des erreurs

I intégral : il s'agit d'analyser les erreurs du déplacement passé

D dérivateur : il s'agit d'analyser les erreurs du déplacement futur

Saisir la consigne angulaire
Calcul de l'écart ϵ , de la somme des écarts $\sum \epsilon$ de la différence des écarts $\Delta \epsilon$
Calcul de la consigne : $\text{consigne} = K_p \cdot \epsilon$
Calcul de y : $y = \text{map}[\text{consigne}, 0, 255]$



doc13: Commande et correction

