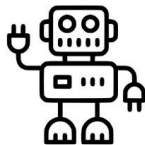
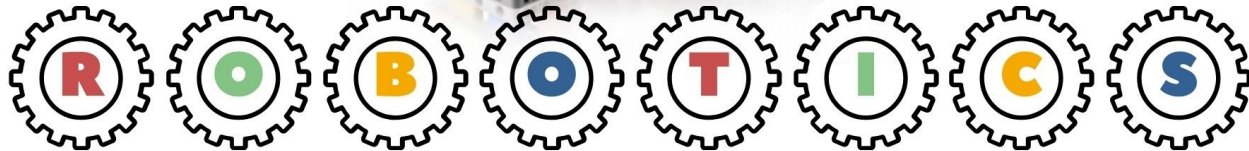
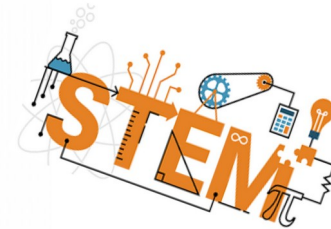


LIVRET DE ROBOTIQUE

ce livret appartient à :

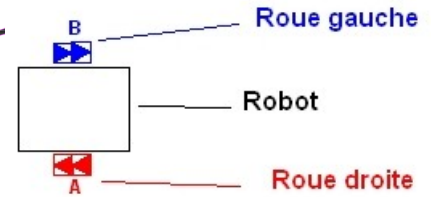
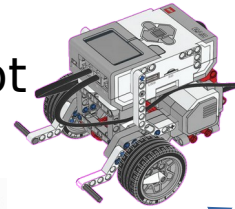


Collège Colette - 2023-2024





Déplacement d'un robot > à 2 roues motorisées



au lancement du programme

définir les moteurs de déplacement pour B et A

définir la vitesse de déplacement à 50 %

avancer tout droit: 0 pendant 360 degrés

Virage à DROITE	Virage à GAUCHE	AVANCE	RECULE
B marche AVANT A marche ARRIERE	B marche ARRIERE A marche AVANT	B marche AVANT A marche AVANT	B marche ARRIERE A marche ARRIERE

avancer tout droit: 0 pendant 360 degrés

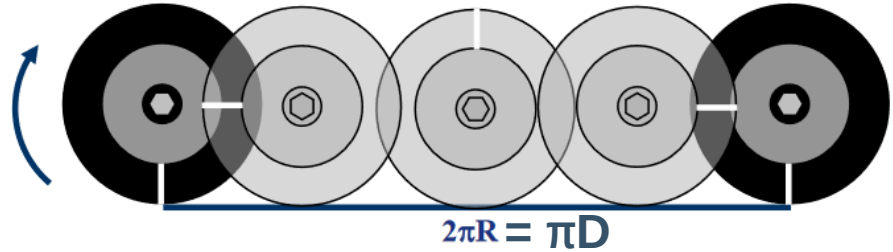
rotations
✓ degrés
secondes

3 façons de programmer les rotations de roues motorisées





Actionneur > GROS Moteur



Si on prend comme exemple une roue LEGO de Diamètre 56 mm :
Périmètre = $\pi D = 3,14 \times 56 = 175,929$ mm

> En un tour la roue avance de 175,9 mm

Pour programmer un déplacement précis du robot, on peut **mesurer avec un mètre** la distance à parcourir et **calculer le nombre de degrés de rotation de roue à programmer** pour faire avancer le robot de cette longueur exacte.

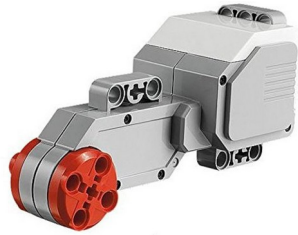
Ce calcul se fait en utilisant la règle de proportionnalité (ou produit en croix)



Produit en croix

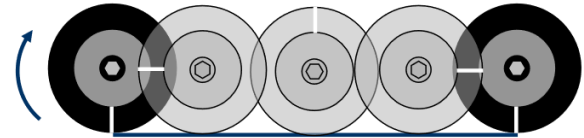
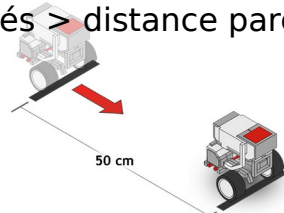
degrés	mm
360	πD
X ?	500

Les calculs sont détaillés et expliqués page suivante



Actionneur (Gros moteur)

> Degrés > distance parcourue > vitesse de déplacement robot



$$\text{Périmètre} = 2 \pi R = \pi D$$

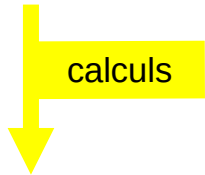
1- Comment programmer un déplacement précis du robot ?

> Il suffit de programmer les rotations de roues en degrés ($360^\circ = 1 \text{ tour de Roue} = \pi D = 2 \times \pi R$)

2- Comment connaître le nombre précis de degrés à programmer pour avancer de 50 cm (500mm) ?

> on utilise la règle de proportionnalité en faisant un **produit en croix** :
on sait que pour un tour de roue (360°) la roue se déplace de la valeur du périmètre (image ci-dessus).
Prenons comme exemple une roue du Robot de Diamètre $D = 56 \text{ mm}$

> Calculer x ? le nombre de degrés recherché



Produit en croix

degrés	mm
360	πD
$x ?$	500

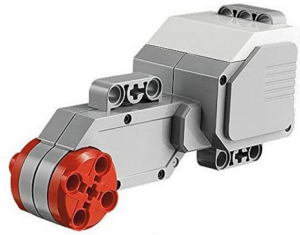
Produit en croix

$$x ? \times \pi D = 360 \times 500 \quad \Rightarrow \quad x ? = (360 \times 500) / \pi D = 180000 / (\pi \times 56) = 1023 \text{ degrés}$$

Il faudra donc programmer un déplacement en marche avant pendant 1023 degrés

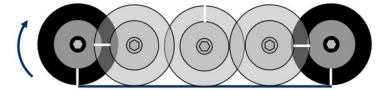


avancer vers l'avant pendant 1023 degrés

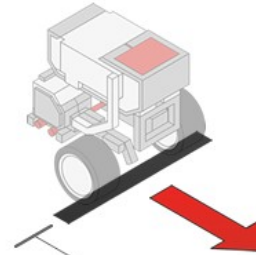


Actionneur (Gros moteur)

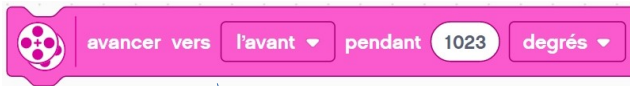
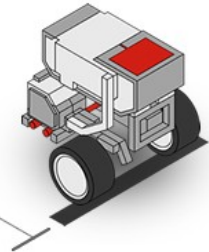
> Degrés > distance parcourue > vitesse de déplacement robot



Rotation pendant 1023 degrés



50 cm



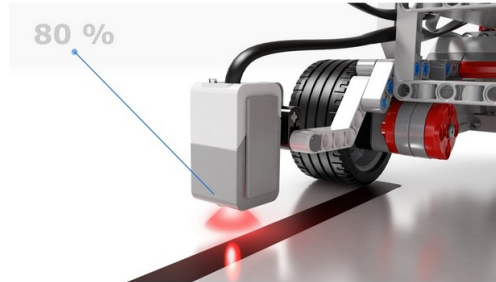
```
au lancement du programme
définir les moteurs de déplacement pour B et C
définir la vitesse de déplacement à 50 %
avancer vers l'avant pendant 1023 degrés
```





Détecteur de ligne

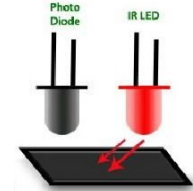
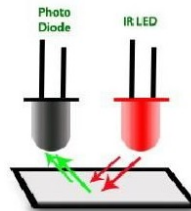
> Mode réflexion lumineuse



surface blanche = 80% de lumière réfléchiée



surface noire = 20% de lumière réfléchiée

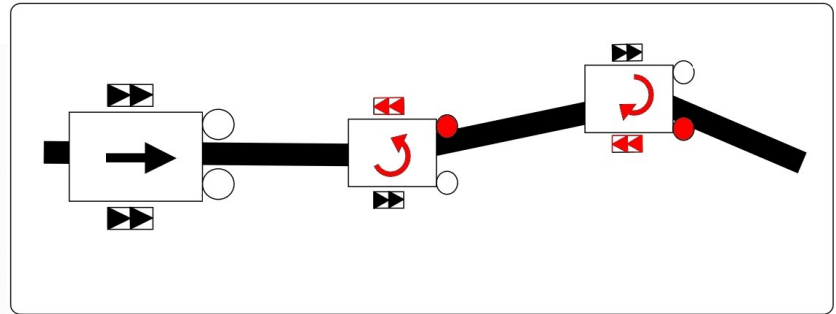
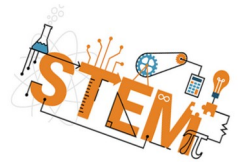


Le **Détecteur de ligne** se compose d'une LED IR et d'une photodiode IR. La lumière InfraRouge émise par la LED frappe la surface et est réfléchiée vers la photodiode IR. La photodiode donne alors une tension de sortie proportionnelle à la réflectance de la surface (haute valeur pour la surface claire / faible valeur pour surface foncée).



Détecter et suivre une ligne

> Mode réflexion lumineuse > Suivi de ligne **ZIG ZAG**



```
graph TD
    Start[au lancement du programme] --> DefineMotors[définir les moteurs de déplacement pour B et C]
    DefineMotors --> DefineSpeed[définir la vitesse de déplacement à 70 %]
    DefineSpeed --> Loop[ répéter indéfiniment ]
    Loop --> If1[si intensité de la lumière réfléchie < 25 % ?]
    If1 -- alors --> MoveLeft[commencer le déplacement gauche: -100]
    MoveLeft --> Else1[sinon]
    Else1 --> If2[si intensité de la lumière réfléchie < 25 % ?]
    If2 -- alors --> MoveRight[commencer le déplacement droite: 100]
    MoveRight --> Else2[sinon]
    Else2 --> MoveStraight[commencer le déplacement tout droit: 0]
    MoveStraight --> Loop
```

Suiveur de ligne ZIGZAG ?

Principe de fonctionnement :

Pour suivre la ligne, le robot tourne à Gauche si le capteur gauche touche la ligne et inversement ...il avancera donc en ZIGZAGpour rester sur la ligne.

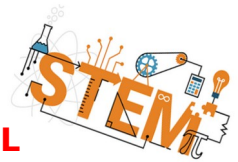
L'avance en zigzag est très saccadée en raison des brusques changement de direction du robotcela peut-être amélioré ...en programmant autrement le robot....l'explication se trouve page suivante.....





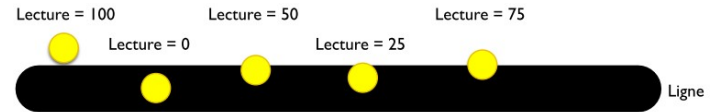
Détecter et suivre une ligne

> Mode réflexion lumineuse > Suivi de ligne **PROPORTIONNEL**

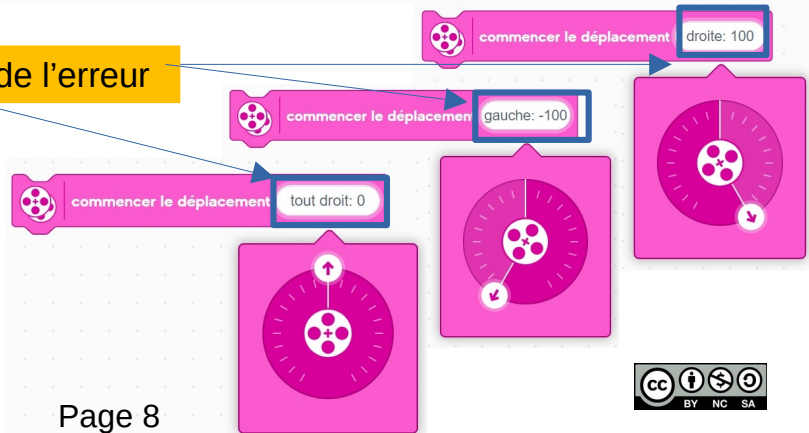
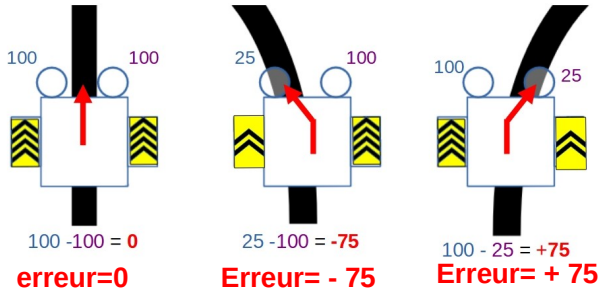


- > Les relevés des capteurs de ligne indiquent le niveau de lumière réfléchi : de 0 à 100 (0= noir > 100= blanc)
- > Dans le programme on va calculer la différence entre le capteur gauche et le droit : c'est l'**ERREUR**
- > Plus l'erreur est un nombre grand, plus le taux de virage sera fort.....si C'est un nombre Positif = virage à DROITE ...Si c'est un nombre négatif = virage à GAUCHE

Zone mesurée par un capteur de lumière : ●



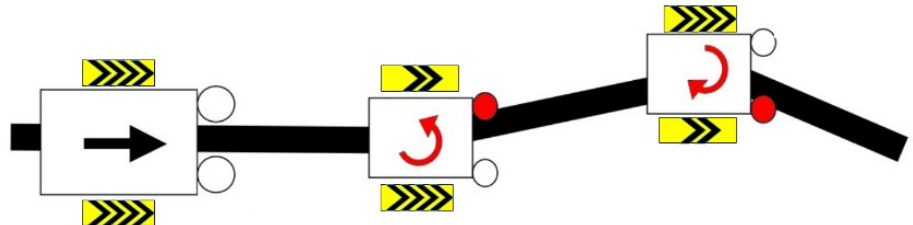
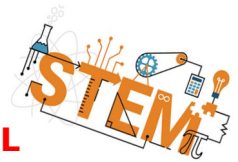
Le robot tournera de la valeur de l'erreur





Détecter et suivre une ligne

> Mode réflexion lumineuse > Suivi de ligne **PROPORTIONNEL**



au lancement du programme

définir les moteurs de déplacement pour B et C

définir la vitesse de déplacement à 50 %

répéter indéfiniment

mettre erreur à $1 \times \text{intensité de la lumière réfléchie} - 2 \times \text{intensité de la lumière réfléchie}$

mettre gain à 3

mettre TAUX-de-VIRAGE à $\text{erreur} \times \text{gain}$

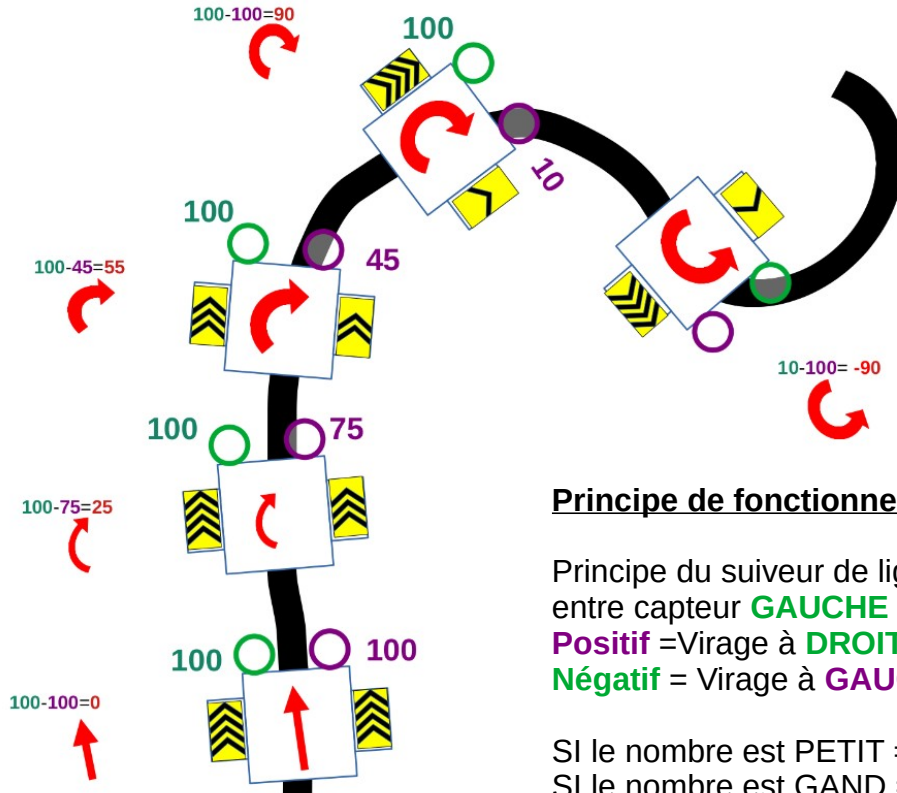
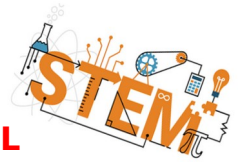
commencer le déplacement TAUX-de-VIRAGE

Le taux de virage sera proportionnel à la différence de lumière réfléchie sur les 2 capteurs ligne



Détecter et suivre une ligne

> Mode réflexion lumineuse > Suivi de ligne **PROPORTIONNEL**



Principe de fonctionnement :

Principe du suiveur de ligne Proportionnel : Si la différence entre capteur **GAUCHE** et **DROITE** est un nombre

Positif = Virage à **DROITE**

Négatif = Virage à **GAUCHE**

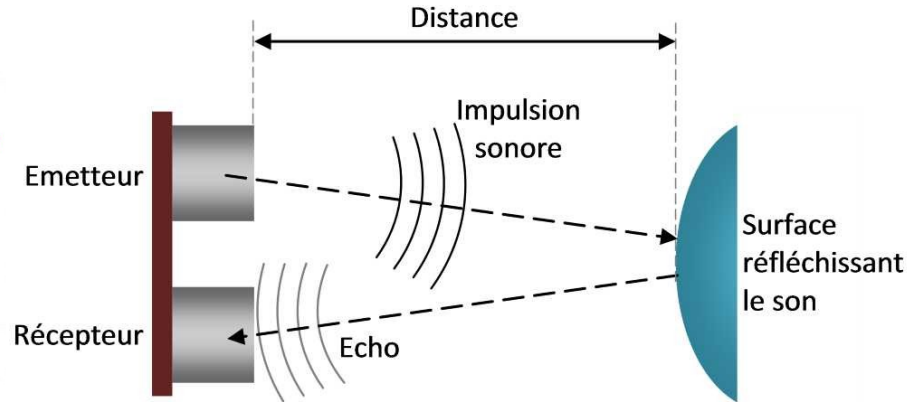
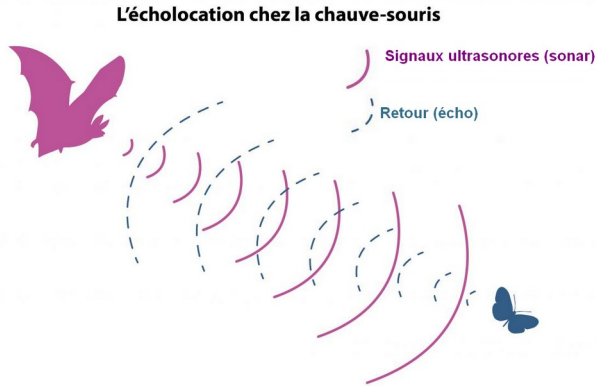
Si le nombre est PETIT = virage DOUX

Si le nombre est GRAND = virage FORT





Détecteur d'obstacle > à Ultrasons



$$D = (T \times C) / 2$$

*D : Distance (mètres)
T : temps (sec)
C : vitesse du son 343 m/s*

Principe de fonctionnement :

Le **Détecteur d'obstacle à ultrasons** émet à intervalles réguliers de courtes impulsions sonores à haute fréquence. Ces impulsions se propagent dans l'air à la vitesse du son. Lorsqu'elles rencontrent un objet, elles se réfléchissent et reviennent sous forme d'écho au capteur. Celui-ci calcule alors la distance le séparant de la cible sur la base du temps écoulé entre l'émission du signal et la réception de l'écho avec la formule de calcul $t = (2d/V)$.

* *ultrasons* : C'est un onde sonore (onde mécanique = vibration de l'air) dans une fréquence trop élevée pour être perçue par l'oreille humaine

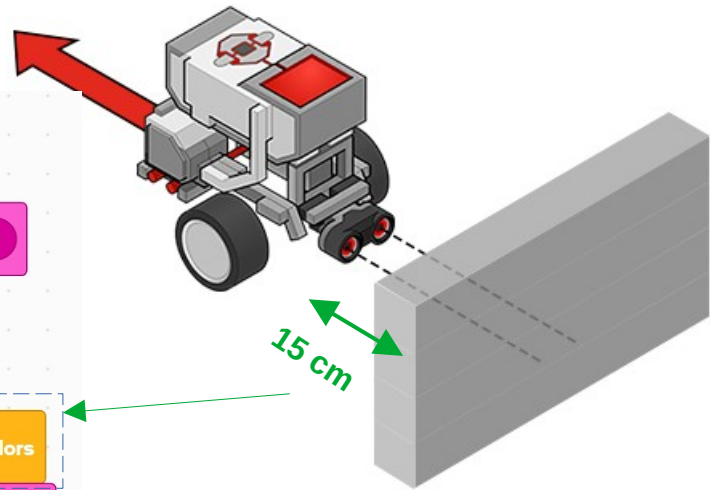


Détecteur d'obstacle (à Ultrasons)

> programmation du fonctionnement

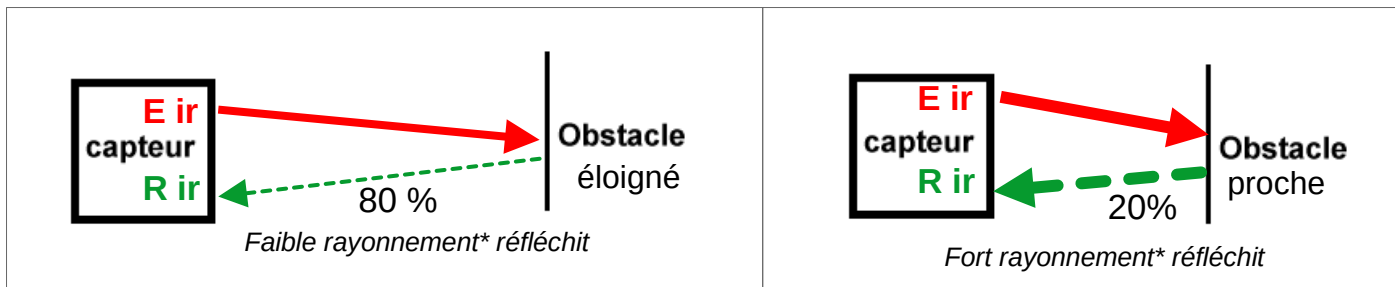
```
graph TD
    Start[au lancement du programme] --> DefineMotors[définir les moteurs de déplacement pour B et C]
    DefineMotors --> DefineSpeed[définir la vitesse de déplacement à 50 %]
    DefineSpeed --> Repeat[ répéter indéfiniment ]
    Repeat --> If[ si distance < 15 cm ? alors ]
    If --> MoveBack[ avancer vers l'arrière pendant 4 rotations ]
    MoveBack --> Else[ sinon ]
    Else --> MoveForward[ commencer le déplacement tout droit: 0 ]
    Else --> Repeat
```

The code is written in a Scratch-style block-based language. It starts with a yellow 'au lancement du programme' block. This is followed by two pink blocks: 'définir les moteurs de déplacement pour B et C' and 'définir la vitesse de déplacement à 50 %'. A yellow 'répéter indéfiniment' loop block contains an 'if' block. The 'if' block has a condition 'distance < 15 cm ?'. If true, it executes 'avancer vers l'arrière pendant 4 rotations'. If false, it executes 'commencer le déplacement tout droit: 0'.





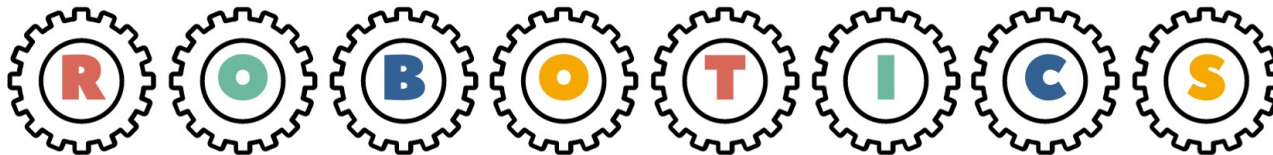
Détecteur d'obstacle > à InfraRouges (IR)



Principe de fonctionnement :

Le **Détecteur d'obstacle** à Infra-Rouge est constitué d'un **Récepteur** qui détecte l'intensité lumineuse dans la gamme des lumières InfraRouge et d'un **Emetteur** de lumière InfraRouge. Un **Emetteur IR** envoie de la lumière Infra-Rouge, si aucun obstacle proche n'est détecté, la valeur lue par le **Récepteur IR** est nulle (et indique 100 %). Si un obstacle proche est détecté, le capteur renvoie par exemple une valeur de 20 % ...(voir image ci-dessus). La lumière réfléchi sera proportionnelle à la distance de l'obstacle (20%=proche.....60 %= plus éloigné...)...

* *rayonnement infrarouge (IR) : C'est un rayonnement électromagnétique de même nature que la lumière visible....mais dont la longueur d'onde est supérieure à celle du spectre visibledonc nos yeux ne voient pas les IR*

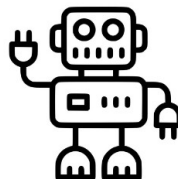


CERTIFICAT DE PARTICIPATION

au stage de robotique remis à :



OUISEM MOUSSI



24 AU 26
OCTOBRE 2023



VALENTIN COUNTRY
MAISON QUARTIER C. FARRERE