Comment orienter un véhicule de façon autonome ?

DOCUMENT PROFESSEUR (AVEC CORRECTION)

Table des matières

Ι.	LES CAPTEURS, ELEMENTS ESSENTIELS POUR ACQUERIR DES INFORMATIONS 2
н.	L'UNITE DE TRAITEMENT : LE MICROCONTROLEUR ARDUINO7
III.	LA DEMARCHE DE REALISATION D'UN PROGRAMME8
IV.	PRISE EN MAIN DE LA CARTE ARDUINO9
Α.	CLIGNOTEMENT D'UNE LED
в.	SON AVEC UN BUZZER
V .	MESURE DE DISTANCE AVEC UN CAPTEUR A ULTRASONS
Α.	CAPTEUR A ULTRASONS (ET LED)
В.	UTILISATION DU CAPTEUR A ULTRASONS (ET BUZZER)
VI.	VERIFICATION DES DONNEES CONSTRUCTEURS POUR LE CAPTEUR A ULTRASONS 14
VII.	FAIRE NAVIGUER LE ROBOT MBOT DANS UN PARCOURS
Α.	ALLUMER DES LEDS DU ROBOT MBOT
в.	EMETTRE DES SONS
C.	COMMANDE AVEC TELECOMMANDE/CLAVIER
D.	DETECTION ET EVITEMEMENT D'OBSTACLES
Ε.	SUIVI DE LIGNES
VIII. COMP	POUR ALLER PLUS LOIN : UTILISATION D'INTELLIGENCE ARTIFICIELLE POUR AMELIORER LE ORTEMENT DU ROBOT
Α.	DEFINITIONS
B. PAR	AMELIORATION DE LA DETECTION DE CONTACT AVEC DE L'INTELLIGENCE ARTIFICIELLE LA METHODE D'APPRENTISSAGE PAR RENFORCEMENT
C. MET	AMELIORATION DU SUIVI DE LIGNE AVEC DE L'INTELLIGENCE ARTIFICIELLE PAR LA HODE D'APPRENTISSAGE PAR RENFORCEMENT

Les véhicules autonomes circulent sur les routes sans l'intervention d'un être humain. Ils sont équipés d'un système de pilotage automatique qui leurs permettent de rouler dans des conditions réelles de circulation.

L'industrie fait également appel à des robots autonomes pour le déplacement de marchandises dans les entrepôts (<u>https://www.youtube.com/watch?v=wxbi-ErDqPk</u> ou <u>https://www.youtube.com/watch?v=LDhJ5I89H_I</u>). Ces robots autonomes sont pourvus de capteurs leur permettant de s'orienter et d'éviter toute collision avec le personnel, les structures fixes et les autres robots.

De nombreux systèmes industriels sont automatiques : ils exécutent des tâches plus ou moins complexes de manière autonome. Chacune de ces tâches peuvent très souvent être décomposées en tâches élémentaires (que l'on peut décrire avec la chaîne d'information/puissance).

Dans cette activité, nous nous intéresserons plus particulièrement aux composants de la chaîne d'information, et notamment à la manière dont l'unité de traitement gère les ordres et informations.

<u>Problématique :</u> Comment orienter un robot de marchandise ou un véhicule en évitant tout contact avec des obstacles ?

I. LES CAPTEURS, ELEMENTS ESSENTIELS POUR ACQUERIR DES INFORMATIONS

Pour se déplacer et détecter la présence d'obstacles sur son chemin, le robot a besoin de capteurs pour obtenir des informations sur son milieu extérieur.

Les capteurs font partie de la **chaîne d'acquisition** de données, située dans la chaîne d'Information d'un système. Un capteur est un dispositif permettant de capter un phénomène physique (distance, température, luminosité, ...) et de le restituer sous forme de signal. Pour notre activité, les capteurs de distance prélèvent une distance entre 2 objets et la renvoient sous forme d'un signal électrique.



Chaine d'acquisition des données représentée dans la chaîne d'Information/Puissance d'un système

La grandeur de sortie du capteur peut varier :

• de manière binaire (information vraie ou fausse), c'est le capteur **Tout Ou Rien** (TOR)

- de façon progressive (variation continue), c'est le capteur **analogique**
- d'échelon de tension ou de courant, c'est le capteur numérique

La détection du phénomène physique par le capteur peut s'effectuer :

- de manière directe par contact
- à une certaine distance sans contact

Parmi les informations de toutes natures issues de notre environnement, il existe les grandeurs physiques associées à des événements climatiques, géométriques ou encore lumineux ou temporels. Le rôle du capteur est de rendre exploitable ces différentes grandeurs physiques en vue de leur traitement ultérieur.

Phénomènes physiques	Applications
Mesure de présence	indique la présence d'un "objet" à proximité immédiate : radar de recul.
Mesure de position, de déplacement ou de niveau	indique la position courante d'un objet animé
Mesure d'un mouvement de rotation ou de translation;	indique la vitesse linéaire ou angulaire d'un "objet"
Mesure d'accélération, de vibrations ou de chocs	Drone, véhicules
Mesure de débit, de force, de couples, de pressions	Machine à laver, station météo
Mesure de température, d'humidité	Station météo, installation de chauffage
Mesure de vitesse	Véhicules

Exemples de capteurs usuels d'après Ressources Capteurs (<u>https://eduscol.education.fr/sti/technologie-</u>

college/ressources pedagogiques/le-fonctionnement-dune-ecluse#fichiers-liens) :

Capteurs	Aspect	Symbole	Classification et applications
Photo résistance L.D.R			Capteur de luminosité, de type analogique (résistance), sans contact.
Contacteur fin de course		F	Capteur TOR, détection avec contact de la position d' un objet. Capteur logique.
Roue codeuse			Capteur numérique, détection d' un mouvement de rotation sans contact.
Potentiomètre rotatif			Capteur permettant de détecter la position angulaire d'un objet avec contact. Capteur analogique (résistance).





Organigramme d'après Ressources Capteurs (<u>https://eduscol.education.fr/sti/technologie-</u> college/ressources_pedagogiques/le-fonctionnement-dune-ecluse#fichiers-liens)

<u>Activité 1 :</u> En utilisant l'organigramme, retrouver le type de capteur qu'il est possible d'utiliser pour notre problématique ?



A travers les prochaines activités nous allons appréhender le fonctionnement réel de capteurs.

<u>Activité 2 :</u> Quels types de capteurs peuvent être utilisés pour détecter des obstacles ? Quels sont les avantages et inconvénients principaux de chacun d'eux ?

Il est possible de s'aider des informations acquises sur les capteurs présents dans les systèmes du laboratoire de SI et de s'aider des ressources disponibles sur https://eduscol.education.fr/sti/technologie-college/ressources_pedagogiques/le-fonctionnement-dune-ecluse#fichiers-liens

Capteurs	Avantages	Inconvénients
Capteur mécanique de fin de course	Prix	Contact obligatoire avec l'objet → usure de l'objet
D2F-01L		
Capteur inductif	Pas de contact	Prix
	Très précis	Détection uniquement d'objets métalliques. Détection de faibles distances
Capteur capacitif	Pas de contact	Prix.
F	Très précis	Détection de faibles distances

Capteur à ultrasons	Prix Précision des mesures	Efficace uniquement pour distances moyennes (de l'ordre de la dizaine de cm)
Capteur de temps de vol – LIDAR	Très précis	Très cher Possibilité de problèmes avec la luminosité (intense) pour le LASER.

(images d'après www.gotronic.fr)

Les capteurs ne peuvent pas fonctionner seuls. Il faut les relier à une carte d'acquisition qui va acquérir les données provenant des capteurs. Ces données sont ensuite analysées par les unités de traitement (microcontrôleur) afin, par exemple, d'analyser la distance entre l'objet et le capteur et de prendre une décision sur les mouvements possibles du robot. Cette prise de décision n'est pas aléatoire, elle est scrupuleusement et méthodiquement définie dans un programme informatique global qui régit le fonctionnement du robot.

II. L'UNITE DE TRAITEMENT : LE MICROCONTROLEUR ARDUINO

Les unités de traitement existent sous différentes formes. Les microcontrôleurs en font partie. Leurs couts et les fonctionnalités qu'ils proposent dépendent du cahier des charge du système à concevoir. Pour des raisons de coûts et de simplicité de manipulation, nous utiliserons une carte de type Arduino. Le site officiel <u>www.arduino.cc</u> propose de multiples ressources sur les produits Arduino. La carte de « base » est la carte Arduino Uno. Cette carte est fondée sur un microcontrôleur ATMega328 cadencé à 16 MHz. Les connecteurs situés sur les bords extérieurs du circuit imprimé permettent d'enficher une série de modules complémentaires. Cette carte peut se programmer avec le logiciel Arduino, grâce à un cordon de connexion USB. Les entrées/sorties de la carte Arduino sont détaillées sur la figure suivante.



Cette carte Arduino Uno dispose :

- d'un microcontrôleur ATMega328
- de 14 broches numériques d'entrées/sorties (dont 6 peuvent être utilisées en sorties PWM 2)
- de 6 entrées analogiques (qui peuvent également être utilisées en broches entrées/sorties numériques)
- d'un quartz 16 MHz (horloge de cadencement)
- d'une connexion USB (avec son contrôleur associé) qui permet la programmation du microcontrôleur ainsi que l'alimentation de la carte
- d'un connecteur d'alimentation jack (nécessaire si le cordon USB est déconnecté après programmation)
- d'un connecteur ICSP (programmation "in-circuit")
- d'un bouton de réinitialisation (reset)

III. LA DEMARCHE DE REALISATION D'UN PROGRAMME

La carte Arduino est une carte programmable, à volonté, à cela près que le langage de base n'est malheureusement pas Python, mais le langage Arduino (un mix entre le langage C et C++). La syntaxe est bien plus lourde que celle offerte par Python, mais aussi bien plus rigoureuse.

Pour simplifier la mise en place des programmes informatiques, la programmation s'effectuera ici en **Scratch** via le logiciel **mBlock** dont la notice d'utilisation est en annexe « **NOTICE LOGICIEL MBLOCK** *avec ARDUINO* ».

<u>Activité 3 :</u> En utilisant l'annexe « **NOTICE LOGICIEL MBLOCK avec ARDUINO** », connectez la carte Arduino Uno au PC via le logiciel mBlock.

La démarche de conception d'un programme est donnée ci-dessous :

- 1. On utilise le logiciel mBlock pour écrire son programme.
- 2. On télécharge (téléverse) le programme dans la carte (il y a une étape de compilation intégrée).
- Le programme étant téléchargé, la carte fonctionne en autonomie si elle a sa propre source d'énergie, ou reste connectée au PC sinon. Il est également possible d'avoir une communication avec le PC pour renvoyer des données mesurées (via la liaison série par exemple).

IV. PRISE EN MAIN DE LA CARTE ARDUINO

A. <u>CLIGNOTEMENT D'UNE LED</u>



<u>CODE</u>

<u>Activité 4 :</u> Réaliser le montage puis réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans la carte Arduino Uno afin de tester son fonctionnement.

Remarque : Les blocs sont à sélectionner puis à faire glisser dans la feuille blanche du logiciel mBlock. Un « clic » est entendu lorsque les blocs sont correctement emboités entre eux.

		aei	nal												
our toujours	1														
attendre 1	secs														
∞ régler la so	rtie d	le l	a b	roc	he	nur	nér	iqu	ie (9) •	sur	h	aut	•
attendre 1	secs		•	•		1	1	1	:		•	•	:	:	:

Vérification visuelle du fonctionnement (clignotement de la LED toutes les secondes).





B. SON AVEC UN BUZZER

BUZZER :

- Fil noir sur GND
- Fil rouge sur D8





<u>CODE</u>

<u>Activité 5</u>: Réaliser le montage puis réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans la carte Arduino Uno afin de tester son fonctionnement.

our toujours	· ·		•													
∞ jouer la note	C4	•	pe	enda	ant	0	.25		tem	ıps	sur	le	ро	rt (8	
attendre 1 se	cs .			•		:	•	•	•			1	•	-	•	
∞ jouer la note	A2	•	pe	enda	ant	0	.25		ten	nps	sur	le	ро	rt	8	
attendre 1 se	cs :		•	•	•			•	•	•		•	•	•		
☉ jouer la note	G7	•	pe	enda	ant	0	.25		ten	nps	sur	le	ро	rt	8)
attendre 1 se	cs .		-	•		•	•	•	-	•	•	•	•		-	
و																

Vérification auditive du fonctionnement (le buzzer joue un son différent (3 en tout) chaque seconde).

V. MESURE DE DISTANCE AVEC UN CAPTEUR A ULTRASONS

Le capteur à ultrasons est parfait pour tous les projets nécessitant de faire de la mesure de distances, en évitement d'obstacles par exemple. Le HC-SR04 (ou HC-SR05) est un produit peu onéreux et facile à utiliser.

Selon les informations fournies par le constructeur, le capteur à ultrasons permet une mesure d'un objet situé à une **distance allant de 2 cm à 4 m** et fourni une mesure avec une **précision de 3 mm**. Son cône de propagation des ondes est de **15** °.

Principe de fonctionnement :

- 1. Un signal de déclenchement (trigger) est émis pendant 10 µs.
- 2. Le module émet 8 signaux (minimum) à 40 kHz.
- 3. Le module se met à l'écoute d'un signal de retour.
- 4. Le temps entre l'émission et la réception est le temps nécessaire au signal pour faire l'aller et le retour vers l'objet qui a réfléchi celui-ci.



Illustration du signal TRIGGER (émetteur) et ECHO (réception) d'après www.generationrobots.com

Pour résumer cela, le capteur à ultrasons envoie, via son émetteur, un ultrason (son inaudible pour l'oreille humaine) : *Impulsion sonore*. Lorsqu'un objet est détecté, le son est rebondi sur l'objet. Le récepteur reçoit le signal sonore de retour (*Echo*). Le capteur mesure ensuite le temps mis entre l'émission et la réception du signal. Cela lui permet de déterminer sa distance à l'objet.



On a donc **Distance = (Temps_mesuré * Vitesse_du_son)/2** (340 m/s pour la vitesse du son)

<u>Remarque</u>: Le divisé par 2 correspond à l'aller-retour du son entre le capteur et l'objet (émetteur-objet +objet-récepteur)

<u>Attention</u>: La vitesse du son dans l'air dépend de la température. Pour nos activités, on supposera que la température ambiante est constante.

Pratiquement tous les matériaux qui reflètent le son peuvent être détectés, peu importe leur couleur.

A. <u>CAPTEUR A ULTRASONS (ET LED)</u>

<u>LED :</u>

- Pate courte sur GND
- Pate longue sur D9

CAPTEUR US :

- GND sur GND
- VCC sur 5V
- TRIG sur **D12**
- ECHO sur D13
- OUT non connecté







CODE

<u>Activité 6</u>: Réaliser le montage puis réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans la carte Arduino Uno afin de tester son fonctionnement.

<u>NB</u> : Les distances sont indiquées en cm.



Vérification du fonctionnement avec un obstacle placé à moins de 10 cm du capteur à ultrasons.



B. UTILISATION DU CAPTEUR A ULTRASONS (ET BUZZER)

BUZZER :

- Fil noir sur GND
- Fil rouge sur D8

CAPTEUR US :

- GND sur GND
- VCC sur **5V**
- TRIG sur D12
- ECHO sur D13
- OUT non connecté







<u>CODE</u>

<u>Activité 7</u>: Réaliser le montage puis réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans la carte Arduino Uno afin de tester son fonctionnement.

<u>NB</u> : Les distances sont indiquées en cm.



VI. VERIFICATION DES DONNEES CONSTRUCTEURS POUR LE CAPTEUR A ULTRASONS

<u>Activité 8 :</u> Détailler un protocole expérimental permettant de vérifier les caractéristiques du capteur à ultrasons annoncées par le constructeur (Mesure d'une **distance allant de 2 cm à 4 m** avec une **précision de 3 mm.** Le **cône de propagation des on**des est de **15** °).

DISTANCE MESURABLE PAR LE CAPTEUR :

Pour évaluer la plage de mesure et la précision du capteur, il est nécessaire de faire des mesures de distance à l'aide du capteur, en les comparant à des mesures effectuées avec un mètre.

Matériel nécessaire :

- Montage avec « Capteur à ultrasons et LED » ou « Capteur à ultrasons et Buzzer »
- 1 Règle graduée
- 1 Mètre
- Une plaque de bois ou de carton pour réfléchir l'onde ultrasonore

Protocole :

- Dans le code Scratch, modifier la valeur dans la boucle « *si* » (partie verte du code). Téléverser le code dans la carte Arduino
- Eloigner la plaque du capteur puis la rapprocher doucement jusqu'à ce que la LED s'allume (ou le buzzer émette un bruit). Mesurer et noter la distance entre le capteur et la plaque à l'aide du mètre ou de la règle graduée.
- Recommencer les étapes précédentes en ajoutant 1 cm (ou tous les 0,1 cm si besoin de plus de précision) à la valeur dans la boucle « *si* » du code Scratch.

<u>NB</u>: Les mesures étant longues, il est possible de diviser la classe en plusieurs groupes dont chaque groupe effectue ces mesures sur une plage différente (groupe 1 de 0 à 20 cm, groupe 2 de 20 à 40 cm, groupe 3 de 40 à 60 cm, etc...).

Remarques : Il faudra veiller à ce que la surface sur laquelle les ondes sonores vont se réfléchir soit la plus grande, plane et homogène possible. Il faudra également faire attention de ne pas se positionner sur le sol car les ondes se réfléchissent sur le sol et peuvent perturber les mesures.

Pour un montage optimal (montage en extérieur, pas d'obstacles à proximité ou de surface en dessous du capteur, une surface dure et lisse de bonne taille en face), le capteur et l'obstacle sont alignés, le capteur est centré sur le centre de la surface de l'obstacle et les deux sont bien perpendiculaires.

Photo du protocole expérimental (exemple pour une distance de 30 cm) :



Mesure de la distance entre la position de l'objet avec le logiciel et avec le mètre. horizontal.

CONE DE PROPAGATION DES ONDES :

Pour évaluer le cône de propagation des ondes du capteur, il est nécessaire de faire des mesures d'angles pour une distance fixée (ici 30 cm par exemple).

Matériel nécessaire

- Montage avec « Capteur à ultrasons et LED » ou « Capteur à ultrasons et Buzzer »
- 1 Règle graduée
- 1 Rapporteur
- 1 Mètre
- Une plaque de bois ou de carton pour réfléchir l'onde ultrasonore

Protocole :

- Dans le code Scratch, modifier la valeur dans la boucle « *si* » et mettre *30 cm* (partie verte du code). Téléverser le code dans la carte Arduino
- Pour une distance fixée à 30 cm, décaler la plaque de 1°. Déplacer doucement la plaque vers le capteur jusqu'à ce que la LED s'allume (ou le buzzer émette un bruit). Mesurer et noter la distance entre le capteur et la plaque à l'aide du mètre ou de la règle graduée.
- Recommencer les étapes précédentes en répétant le protocole en ajoutant 1°. Et cela d'un côté puis de l'autre (sens positif et sens négatif)

Photo du protocole expérimental pour la détection du cône de propagation :



Mesure de l'angle entre la position initiale et celle pour une détection de l'objet lors d'un décalage horizontal.

<u>Activité 9 :</u> Mettre en place ce protocole expérimental et comparer les résultats obtenus à ceux donnés par le constructeur (Mesure d'une **distance allant de 2 cm à 4 m** avec une **précision de 3 mm.** Le **cône de propagation des on**des est de **15** °).



Tests pour des distances allant de 1 à 50 cm en intérieur :





Tests pour des distances allant de 100 à 150 cm en intérieur :







Tests pour des distances allant de 200 à 250 cm en intérieur :





Test global pour des distances allant de 1 à 250 cm en intérieur :

L'erreur moyenne sur la plage complète de 1 à 250 centimètres est de 38 millimètres. La précision donnée par le constructeur (3 mm) n'est pas atteinte même si les résultats expérimentaux sont proches. Cependant, pour des projets ne nécessitant pas des mesures au millimètre près, ce type de capteur convient très bien.

La distance maximale atteinte est de 384 cm, inférieure aux 400 cm annoncés par le constructeur.

Une étude sur le cône de propagation des ondes peut également être menée de la même manière pour vérifier les données du constructeur. Le cône de propagation est d'environ 30° (de +15° à -15°)



Courbe pour le cône de propagation du capteur à ultrasons

VII. FAIRE NAVIGUER LE ROBOT MBOT DANS UN PARCOURS



Pour simplifier la mise en place des programmes informatiques, la programmation du robot mBot s'effectuera ici en **Scratch** via le logiciel **mBlock** dont la notice d'utilisation est en annexe « **NOTICE LOGICIEL MBLOCK avec MBOT** ».

<u>Activité 10 :</u> En utilisant l'annexe « *NOTICE LOGICIEL MBLOCK avec MBOT* », connectez la carte le robot mBot au PC via le logiciel mBlock.

A. ALLUMER DES LEDS DU ROBOT MBOT

<u>Activité 11 :</u> Réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans le robot mBot afin de tester son fonctionnement.

CODE



Activité 12 : Expliquer les différentes étapes du code précédent.

- Lorsque le robot est allumé :
 - Le robot effectue les actions suivantes sans arrêt (pour toujours)
 - Définir la variable RED à 1

- Pendant 50 fois :
 - Allumer la LED avec une couleur rouge dont la valeur dépend de RED
 - Incrémenter de 1 la valeur de RED
 - Attendre 0.05 sec
 - Recommencer à allumer la LED avec une couleur rouge ...

La boucle qui se répète 50 fois permet de faire varier graduellement l'intensité de la LED rouge, de plus en plus intense.



B. EMETTRE DES SONS

<u>Activité 13 :</u> Réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans le robot mBot afin de tester son fonctionnement.

CODE

<u> </u>	
Lorsque le mBot(mcore) démarre	
attendre jusqu'à 🦀 sur appui du bouton Carte pr	ressé 🔻 ?
jouer la note C4 ▼ pendant 0.5 pulsations	
🤓 jouer la note C4 ▼ pendant 0.5 pulsations	
🤓 jouer la note G4 🔻 pendant 0.5 pulsations	
iouer la note G4 ▼ pendant 0.5 pulsations	
🥰 jouer la note A4 🔻 pendant 0.5 pulsations	
i jouer la note A4 ▼ pendant 0.5 pulsations	
i jouer la note G4 ▼ pendant 0.5 pulsations	

Activité 14 : Vérifier le bon fonctionnement du code en appuyant sur le bouton du mBot.

Une fois le bouton appuyé, le robot émet 7 sons les uns à la suite des autres.

C. COMMANDE AVEC TELECOMMANDE/CLAVIER

<u>Activité 15 :</u> Réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans le robot mBot afin de tester son fonctionnement.

CODE

· ·	quand la touche flèch	ne du haut ▼	est pressée					
	avancer 🔻 à	50 % de pui	ssance					
	attendre 1 secs							
	avancer 🔻 à	0 % de puis	ssance					
	quand la touche espa	ice 🔻 est pre	essée					
	allumer la lumière	tout ▼ av	ec la couleu	r rouge(0	vert 6	50 bleu	
	allumer la lumière	tout v av	ec la couleu	r rouge(0	vert	bleu	
· ·	allumer la lumière attendre 1 secs	tout ▼ av	ec la couleu	r rouge(0	vert	50 bleu	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs attendre la lumière allumer la lumière	tout av	ec la couleu	r rouge(r rouge(0 0 0 0 0	vert (bleu bleu bleu bleu	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs allumer la lumière	tout av tout av tout av tout av	ec la couleu ec la couleu ec la couleu	r rouge(· · · · · · · · · · · · · · · · · · ·		vert	bleu bleu bleu bleu	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs	tout av av tout av tout av	ec la couleu ec la couleu 	r rouge (r rouge (r rouge (vert	bleu bleu bleu bleu bleu	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs	tout av · · · · tout · tout av · ·	ec la couleu ec la couleu 	r rouge (r rouge (0 	vert vert vert	Image: Non-State Image: Non-State Image: Non-State	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs allumer la lumière	tout ■ · · · · · · tout ■ av tout ■ av · · · av · · · · · · · · · · · · · · · · · · ·	ec la couleu contra contra	r rouge (vert vert vert vert	Image: organization of the sector of the	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs attendre 1 secs allumer la lumière	tout av tout tout tout tout tout tout tout tou	ec la couleu contra couleu ec la couleu ec la couleu contra contra co	r rouge (r rouge (r rouge (r rouge (· · · · · · · · · · · · · · · · · · ·		vert () vert () vert () vert () vert () vert () vert ()	Image: Network Image: Network Image: Network <td< th=""><th></th></td<>	
· · · · · · · · · · · · · · · · · · ·	allumer la lumière attendre 1 secs attendre 1 secs allumer la lumière	tout ■ • • • • • • tout ■ • tout ■ • tout ■ • tout ■ • est pressée ■	ec la couleu ec la couleu	r rouge (r rouge (r rouge (r rouge (vert vert vert vert	Delet Delet </th <th></th>	

<u>Activité 16</u>: Vérifier le bon fonctionnement du robot en le faisant se déplacer, allumer ses LED et émettre un son.

Le fonctionnement est vérifié avec la télécommande ou en appuyant sur le clavier avec les touches « flèche du haut » ou « 1 » ou « espace ». Il faut que le câble USB soit connecté entre le PC et le robot mBot pour utiliser le clavier du PC.

D. DETECTION ET EVITEMEMENT D'OBSTACLES

On souhaite faire fonctionner le robot mBot sur une piste avec des obstacles. On souhaite que le robot puisse détecter les obstacles devant lui grâce à son capteur à ultrasons. Lorsque le robot avance, si un obstacle est à moins de 10 cm, le robot doit tourner à droite pendant 1 seconde pour le contourner. Sinon, le robot continue d'avancer.

Activité 17 : Expliquer les différentes étapes de fonctionnement du robot sous forme d'algorigramme.

Première version avec distance inférieure à 10 cm :



Autre version possible (inversion des conditions SI et SINON).



<u>Activité 18 :</u> Réaliser le code décrit précédemment sur le logiciel mBlock puis le téléverser dans le robot mBot.

<u>Remarque :</u> Il faudra veiller à ce que le port du capteur à ultrasons soit le même lors du branchement sur le robot et dans le code.

CODE







Activité 19 : Tester son fonctionnement en positionnant des obstacles sur le trajet du robot.



Lorsque le robot détecte un obstacle à moins de 10 cm devant lui, il tourne à droite puis continue d'avancer.

E. SUIVI DE LIGNES

Le robot mBot comporte un module de suiveur de ligne. Il s'agit de 2 capteurs composés d'une LED infrarouge et d'un photo-transistor qui recevra l'information lumineuse. Ces 2 capteurs (situés à droite et à gauche de l'axe central du robot) permettent d'envoyer un signal au robot lorsqu'ils détectent une ligne noire sur un fond blanc.

La LED émettrice envoie une lumière infrarouge qui est réfléchie par le sol en direction du phototransistor qui capte ainsi la quantité de lumière en retour.

Les couleurs foncées réfléchissent moins bien la lumière que les couleurs claires, le capteur peut ainsi définir s'il se trouve au-dessus d'une couleur claire (lorsqu'il reçoit beaucoup de lumière), il prend alors la valeur 1, ou s'il se trouve au-dessus d'une couleur foncée (lorsqu'il reçoit peu de lumière), il prend alors la valeur 0.

Etat du robot mBot	Capteur gauche	Capteur droit
0		
1		
2		
3		

<u>Activité 20 :</u> Selon les différents états du robot définis dans le tableau ci-dessous, que doit faire le robot pour continuer à suivre la ligne noire ?

0 : Le robot continue à avancer.

- 1 : Le robot doit tourner vers la gauche.
- 2 : Le robot doit tourner vers la droite.
- 3 : Le robot doit reculer pour recouper la piste noire .

<u>Remarque</u>: Pour la suite, nous garderons les états du robot (0/1/2/3) pour les différentes actions du robot déterminées précédemment

Pour tester le fonctionnement du robot avec le suivi de ligne, il est possible de réaliser soi-même un circuit noir sur des feuilles blanches (largeur du trait noir supérieur à 2 cm). Il est également possible d'utiliser le circuit fourni dans la boite du robot mBot.



Activité 21 : Expliquer les différentes étapes de fonctionnement du robot sous forme d'algorigramme.



Organigramme du fonctionnement du robot en mode suivi de ligne :

<u>Activité 22</u>: Réaliser le code sur le logiciel mBlock et le téléverser dans le robot mBot. Tester son fonctionnement en positionnant le robot sur la piste noire.

<u>Remarque importante :</u> Si les branchements et la construction du robot ont été faits en suivant les instructions dans la boite du mBot, le moteur droit est situé à droite dans le sens de marche, le gauche à gauche. Les moteurs ont également leur sens d'avance dans le sens de marche.

<u>Remarque :</u> Il faudra veiller à ce que le port du suiveur de ligne soit le même lors du branchement sur le robot et dans le code.

CODE



<u>Activité 23 :</u> Lors du fonctionnement du robot mBot sur le circuit, identifier les différentes phases du code selon si les capteurs du robot détectent du noir ou du blanc.



Détection du blanc pour le capteur droit et de noir pour le capteur gauche. Le robot doit retourner vers la gauche pour revenir sur la ligne noire.



Détection du blanc pour le capteur gauche et de noir pour le capteur droit. Le robot doit retourner vers la droite pour revenir sur la ligne noire



Détection de la ligne noire par les deux capteurs du robot. Le robot suit bien la ligne. Il continue d'avancer.



Détection du blanc par les deux capteurs du robot. Le robot est en dehors du circuit et doit reculer pour retourner sur la ligne noire.

VIII. POUR ALLER PLUS LOIN : UTILISATION D'INTELLIGENCE ARTIFICIELLE POUR AMELIORER LE COMPORTEMENT DU ROBOT

Nous allons mettre en place des solutions utilisant de l'Intelligence Artificielle (IA) afin d'améliorer les réactions et le comportement global du robot. Il existe de nombreuses méthodes d'IA pour améliorer les systèmes (Méthode par apprentissage, Méthode des plus proches voisins, Réseaux de neurones, ...). Nous allons ici utiliser uniquement une méthode d'IA très répandue : **l'apprentissage par renforcement** (ou *machine learning* par renforcement).

Tous les codes d'IA se feront sur le logiciel mBlock en Scratch, sans la nécessité d'utiliser d'autres capteurs que ceux présents sur notre robot, ni d'autres logiciels annexes que Scratch.

A. **DEFINITIONS**

Définition de l'apprentissage (ou machine learning) :

La méthode par apprentissage (ou *machine learning*) permet au système d'avoir la **capacité** d' « apprendre » à partir de données, en utilisant des modèles mathématiques. Le système va tirer des enseignements des données et à s'améliorer avec l'expérience, au lieu d'être explicitement programmées pour le faire.

Il existe 3 méthodes par apprentissage : Apprentissage supervisé, non-supervisé et par renforcement.

Les méthodes d'apprentissage supervisé et non-supervisé consiste à utiliser des jeux de données pour entraîner les algorithmes à classer les données ou à prédire les résultats avec précision, avec ou non une supervision de l'utilisateur.

Définition de l'Apprentissage par renforcement (ou machine-learning par renforcement) :

L'apprentissage par renforcement est **une méthode d'apprentissage automatique qui entraîne les systèmes à effectuer certaines actions souhaitées**. L'apprentissage par renforcement repose sur la récompense des comportements souhaités et la punition des comportements indésirables.

B. <u>AMELIORATION DE LA DETECTION DE CONTACT AVEC DE L'INTELLIGENCE</u> <u>ARTIFICIELLE PAR LA METHODE D'APPRENTISSAGE PAR RENFORCEMENT</u>

Nous allons améliorer le système de **détection de contact** en utilisant uniquement le **capteur à ultrasons du robot du mBot en utilisant une IA adaptative** pour modifier le comportement du robot en fonction des obstacles rencontrés. Nous allons o**ptimiser l'évitement** en analysant les collisions passées. La mise en place d'un **score** permet au robot d'apprendre de ses choix et d'optimiser son comportement pour obtenir le meilleur score, c'est-à-dire, de réagir au mieux aux obstacles. <u>Activité 24 :</u> Réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans le robot mBot afin de tester son fonctionnement.

CODE

Créer les variables suivantes (onglet Variables puis cliquer sur « Créer une variable ») :

<u>Attention :</u> Selon les versions du logiciel et les versions des mBot, les variables peuvent ne pas être affectées lors du lancement du code. Il vous faudra dans ce cas remplacer les variables par des valeurs fixes.



Activité 25 : Analyser le programme fourni et définir le rôle des variables suivantes :

"Distance_obstacle" "Distance_de_détection" "Nombre_Obstacles" "Score"

"Distance_obstacle" → Stocke la dernière distance mesurée par le capteur US.
 "Distance_de_détection" → Stocke la dernière distance de détection avant mouvement.
 "Nombre_Obstacles" → Compte combien de fois un obstacle a été détecté.
 "Score" → Stocke les scores du robot, ce sont des récompenses à son bon comportement.

<u>Activité 26 :</u> Après analyse du code fourni ainsi que du comportement du robot, déterminer l'intérêt de l'IA dans la détection du contact étudié ici.

- Le robot détecte les obstacles AVANT le contact grâce au capteur à ultrasons
- Il ajuste sa vitesse selon la proximité de l'obstacle.
- Il apprend à reculer et tourner plus efficacement en analysant les collisions passées.
- Il devient plus fluide et rapide en optimisant ses réactions après plusieurs obstacles.
- Les scores acquis lui permettent de s'améliorer en essayant d'obtenir les meilleures réactions.

<u>Activité 27 :</u> Pour chaque étape du code présenté précédemment, expliquer pourquoi le système reçoit ou perd des points à son score.

- Si le chemin est dégagé (> 10 cm) → Avancer normalement
 > Ajouter 1 à Score // Récompense pour une action normale et sûre
- Si un obstacle est détecté (5 cm < Distance < 10 cm) → Anticipation
 - > Ajouter 0.5 à Score // Récompense pour la bonne anticipation des obstacles
- Si l'obstacle est trop proche (< 5 cm) \rightarrow Évitement
 - diminuer Score de 2 // Pénalité pour ne pas avoir anticipé à temps
- Optimisation avec apprentissage : ajuster les actions en fonction du score
 Diminuer Score de 1 // Pénalité pour trop d'obstacles
- Evaluation finale : ajuster les actions selon le score global
 - > Ajouter 1 à Score // Récompense pour la réinitialisation et l'amélioration

C. <u>AMELIORATION DU SUIVI DE LIGNE AVEC DE L'INTELLIGENCE ARTIFICIELLE PAR</u> LA METHODE D'APPRENTISSAGE PAR RENFORCEMENT

Nous allons améliorer le suivi de ligne grâce à une approche simplifiée d'IA, en appliquant un modèle d'apprentissage par renforcement. Nous allons utiliser uniquement les capteurs de suivi de ligne du robot mBot, une logique d'IA simplifiée basée sur l'apprentissage par renforcement, un contrôle intelligent avec un correcteur PID et un programme en Scratch qui « apprend » à ajuster les moteurs en fonction des capteurs.

Le programme va suivre la logique simplifiée suivante :

- 1. Lire la valeur du capteur pour savoir où se trouve la ligne.
- 2. **Corriger progressivement** la trajectoire du mBot pour suivre la ligne.
- 3. Si le robot est perdu trop longtemps, il effectue une action spéciale pour retrouver la ligne.

Activité 28 : Justifier pourquoi la logique d'apprentissage décrite précédemment est bien de l'IA.

- Apprentissage par renforcement : Le robot apprend progressivement quelle correction fonctionne.
- Adaptation automatique : Si le circuit change, il s'adapte après quelques essais.

<u>Activité 29 :</u> Réaliser le code ci-après sur l'interface du logiciel mBlock. Téléverser/télécharger le code dans le robot mBot afin de tester son fonctionnement.

CODE

Créer les variables suivantes (onglet Variables puis cliquer sur « Créer une variable ») :





Activité 30 : Analyser le code et compléter l'algorigramme ci-dessous :





Activité 31 : Après avoir analysé le programme précédent ainsi que le comportement du robot. Conclure sur le fonctionnement du robot avec IA pour le suivi de ligne. Mettre en avant les points apportés par l'IA.

- Le mBot apprend en ajustant l'intensité de correction progressivement.
- Il garde une mémoire des erreurs passées pour réagir plus vite.
- Il ne réagit pas brutalement mais ajuste progressivement ses moteurs.
- Ajustement dynamique des paramètres PID en fonction du score, permettant une meilleure convergence vers un comportement optimal.
- Il évite de tourner trop violemment, ce qui réduit l'oscillation.
- Le mBot va progressivement s'adapter aux virages et suivre la ligne plus naturellement.
- Équilibre entre exploration et exploitation : Le robot continue d'explorer de nouvelles valeurs de PID tout en optimisant ses actions à partir des expériences passées.