

Cycle 4

Classe de 4ème

Séquence 5

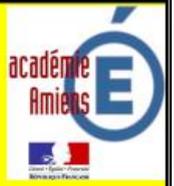
Notion de capteurs / actionneurs

- TP1 - Capteur Infra-rouge
 - TP2 - Capteur à ultra son
 - TP3 - Programmer un servomoteur
 - TP4 - piloter un robot (2 moteurs électriques)
- ⇒ Eviter les obstacles (capteur à ultrason)

S5		Thème de séquence		Problématique	
		Environnement Arduino		P5_1 : Qu'est-ce q'un capteur infrarouge ?	
Compétences		Thématiques du programme		Connaissances	
CT 1.2	► Mesurer des grandeurs de manière directe ou indirecte.	MSOST .1.6	Mesurer des grandeurs de manière directe ou indirecte.	Instruments de mesure usuels. Principe de fonctionnement d'un capteur, d'un codeur, d'un détecteur. Nature du signal	
CS 5,7	Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de				
CT 4.2	► Appliquer les principes élémentaires de l'algorithmique et du codage à la résolution d'un problème simple.				
Présentation de la séquence			Situation déclenchante possible		
Activités permettant de revenir dans l'environnement Arduino découvert en 5ème. Utilisation de maquettes électroniques à base de carte Arduino Uno (infrarouge, ultrason, robot Arduino).			La sécurité est l'un des piliers de la domotique, comment détecter une intrusion ou un obstacle ?		
Éléments pour la synthèse de la séquence (objectifs)			Piste d'évaluation		
Programmation de maquettes réelles permettant de comprendre la notion de capteur/actionneur (détection/action). Importance de suivre l'organigramme donnée.			Chaque élève tire au sort un tp déjà réalisé en binôme et doit le refaire (montage + programme). Contrôle papier sur les éléments électroniques rencontrés dans la séquence (nom et rôle).		
Positionnement dans le cycle 4 Milieu de cycle			Liens possibles avec les EPI ou les parcours (Avenir, Citogen, PEAC)		
Prérequis					
Proposition de déroulé					
	Séance 1	Séance 2		Séance 3	
Question directrice	Que permettent un capteur infrarouge et un capteur à ultrason ?	Comment ouvrir ou fermer une barrière ?		Comment programmer le déplacement d'un robot ?	
Activités	<u>Etude de la plaquette infrarouge</u> équipée de deux capteurs infrarouges : le 1er détecte un faisceau infrarouge produit par un émetteur placé en face, le 2ème détecte une télécommande. Une led devra s'allumer lorsque l'un d'eux détectera un signal (ou pas). 3 programmes à faire. <u>Etude de la plaquette à ultrason</u> permettant de mesurer la distance entre le capteur et un obstacle. 1 programme à faire.	Découverte de la <u>maquette servomoteur</u> , programmation de l'ouverture et la fermeture d'une barrière, 1 programme à faire en suivant l'organigramme.		Programmation d'un robot équipé de deux moteurs électriques connectés à une carte Arduino via un shield moteur. Avancer, reculer, tourner à droite, tourner à gauche et stopper.	
Démarche pédagogique	Identifier et comprendre le fonctionnement des capteurs infrarouge et ultrason. Suivre l'organigramme, créer les programmes.	Intégrer la notion d'angle de 0° à 180° pour la programmation et utiliser le bloc spécifique.		Bien intégrer la programmation des deux moteurs à l'aide de 4 blocs. 1 moteur = 2 blocs : 1 pour programmer le sens, et l'autre pour la puissance.	
Conclusion / bilan	Circulation des informations entre les éléments capteurs, carte Arduino et Led. Parallèle avec les alarmes de maison	Faire le parallèle avec l'ouverture et la fermeture d'une barrière, d'une porte, d'un portail automatique.		Utilisation des connaissances sur les capteurs infrarouge et ultrason afin d'en équiper le robot Arduino et de la rendre autonome (détecteur d'obstacle, suiveur de ligne)	
Ressources	Maquette électronique +Logiciel Mblock + Doc de cours : Séquence 4.pdf	Maquette électronique +Logiciel Mblock + Doc de cours : Séquence 4.pdf		Maquette électronique +Logiciel Mblock + Doc de cours : Séquence 4.pdf	

Séquence 5 : Notion de capteurs / actionneurs

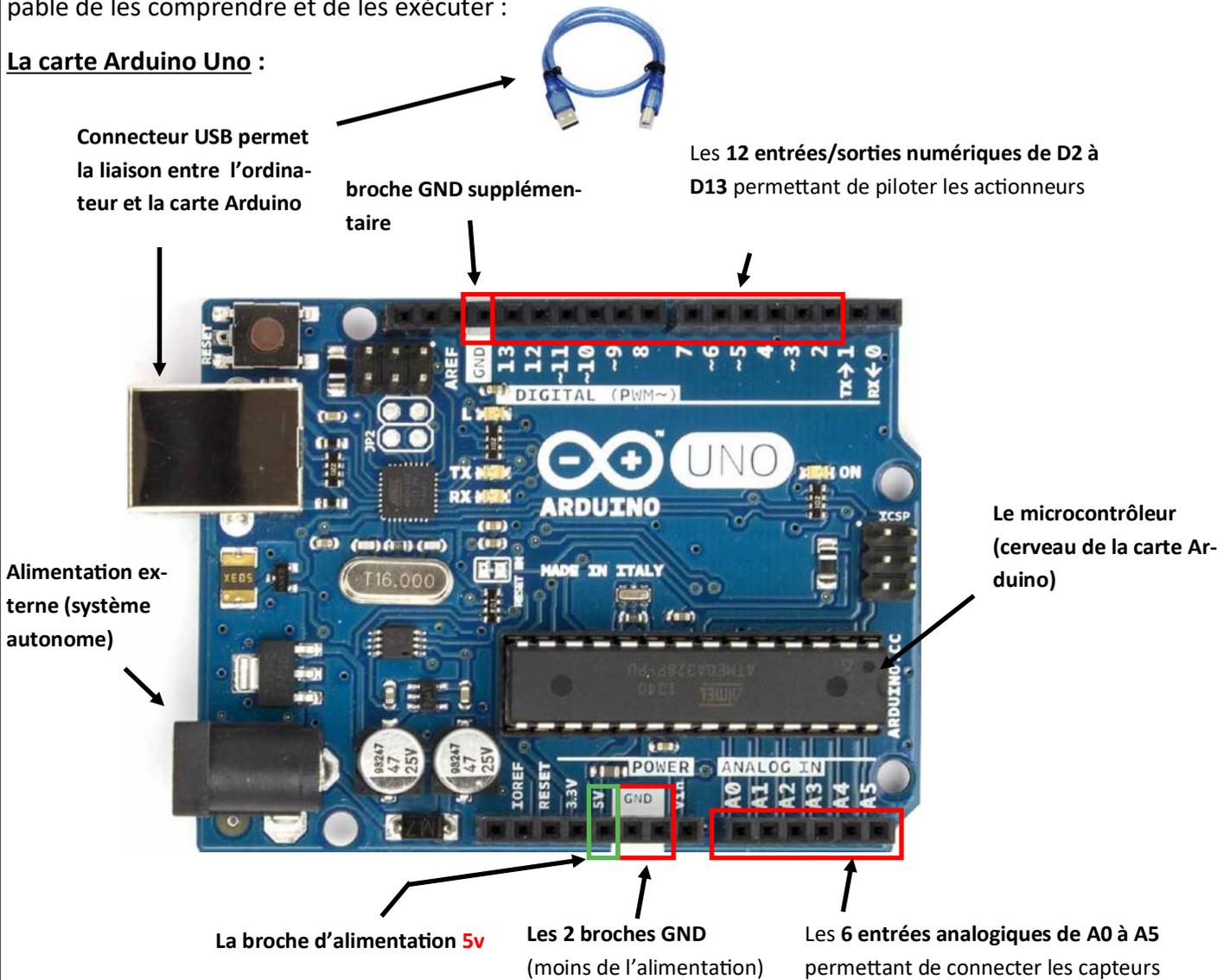
Rappels sur la carte programmable Arduino Uno



Jusqu'à maintenant, les programmes que vous avez créés s'exécutaient sur votre ordinateur et nous pouvions vérifier son fonctionnement par le déplacement des objets programmés sur la scène de Mblock.

Maintenant, les programmes que vous allez faire seront transférés dans une carte électronique qui sera capable de les comprendre et de les exécuter :

La carte Arduino Uno :

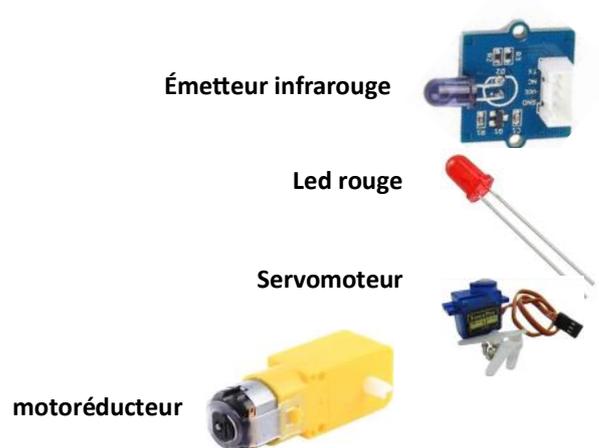


Quelques exemples de capteurs et d'actionneurs que l'on va utiliser dans cette séquence :

Capteurs



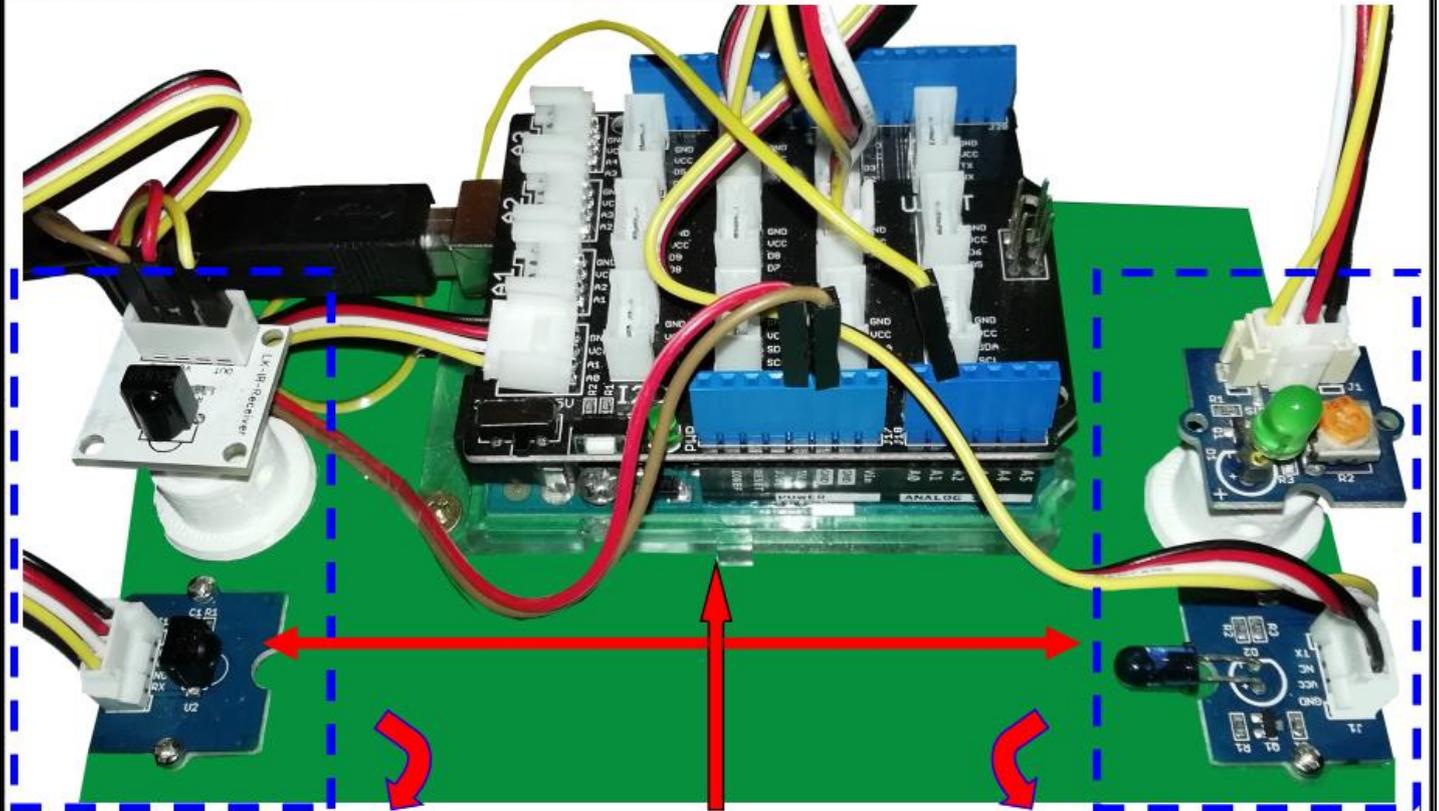
Actionneurs



Séquence 5 : Notion de capteurs / actionneurs

TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge

Description de la maquette capteurs infrarouges :



2 capteurs infrarouges :

- le **blanc** détectera lorsqu'on appuiera sur une touche de la télécommande.

 Il est connecté à la broche analogique **A1** de la carte Arduino et renverra une valeur < 800 quand on appuiera sur une touche de la télécommande.

- le **bleu** détectera lorsque le faisceau infrarouge émis par l'émetteur placé en face de lui sera coupé.

Il est connecté à la broche analogique **A0** de la carte Arduino et renverra une valeur > 988 lorsque le faisceau sera coupé.

La carte Arduino pourra comprendre les signaux des capteurs infrarouges en entrée et envoyer des signaux de sortie aux leds.

Sur la carte Arduino, un **shield Grove** est installé permettant des connexions plus faciles entre les capteurs, les actionneurs et la carte Arduino.

Travail à faire :

- 1 programme permettant d'allumer la led lorsque le faisceau entre l'émetteur et le récepteur infrarouge est coupé
- 1 programme permettant d'allumer la led lorsqu'on appuie sur une touche de la télécommande
- 1 programme permettant d'allumer la led soit en appuyant sur la télécommande ou en coupant le faisceau infrarouge.

2 actionneurs :

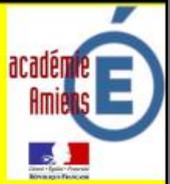
- La led verte :

elle est connectée à la sortie **D6** de la carte Arduino et pour l'allumer il faut mettre l'état logique de la broche D6 au niveau « haut »

- L'émetteur infrarouge enverra un signal sur le récepteur infrarouge placé en face de lui. Il est connecté à la broche digitale **D3** et pour l'activer, il faut mettre l'état logique de la broche D6 au niveau « haut ».

Séquence 4 : Notion de capteurs / actionneurs

TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge



Bilan du TP n°1 Questionnaire à remplir avant de passer à la programmation :

1. Quels sont les capteurs de cette maquette ?
2. Quels sont les actionneurs de cette maquette ?
3. Sur quelle broche de la carte Arduino est connectée le capteur infrarouge qui est placé en face d'un émetteur infrarouge ?
4. Sur quelle broche de la carte Arduino est connecté le capteur infrarouge qui reçoit le signal d'une télécommande ?

Détailler les réponses aux questions suivantes :

5. Que se passe t-il lorsque j'active une télécommande ?
6. Que se passe t-il lorsque le faisceau infrarouge est coupé (avec le doigt par exemple) ?
7. Sur quel broche de la carte Arduino est connectée la Led ?
8. Quand s'allumera t-elle ?
9. Sur quel broche de la carte Arduino est connecté l'émetteur infrarouge ?
10. Quel est son rôle ?
11. Comment s'appelle la carte fixée sur la carte Arduino ?
12. Que permet-elle ?

Séquence 5 : Notion de capteurs / actionneurs

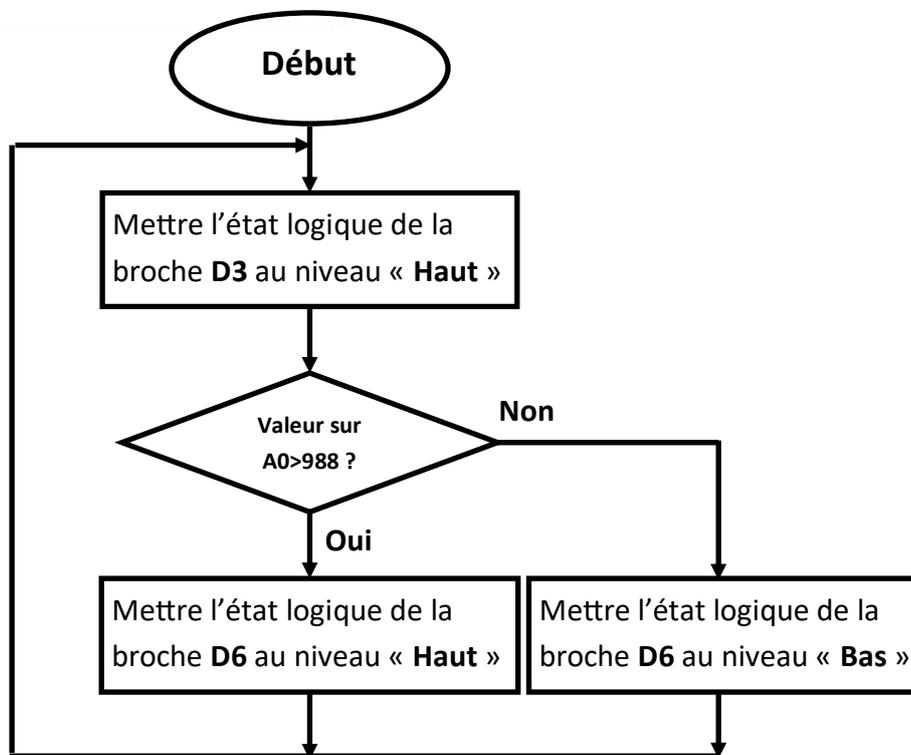
TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge

Programme 1 : allumer la led lorsque le faisceau infrarouge est coupé

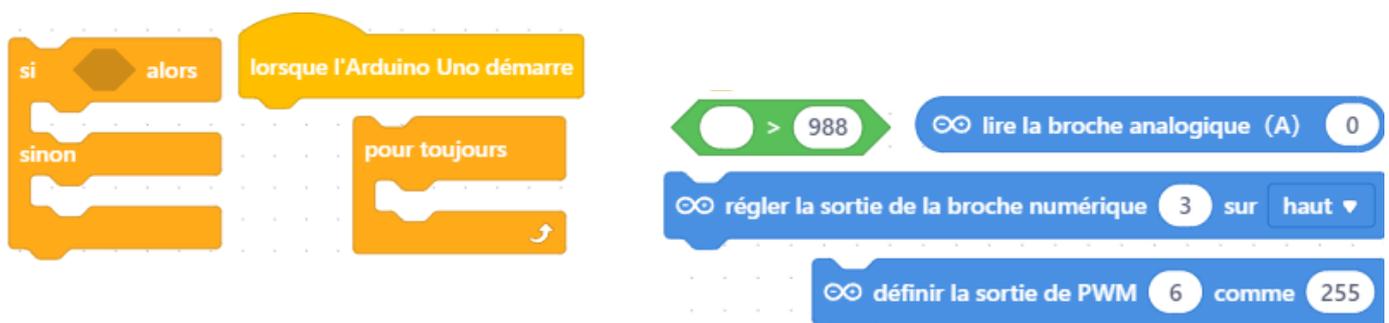
Explications :

1. Votre programme devra commencer par activer l'émetteur infra rouge afin qu'il envoie un signal continue en direction du récepteur : c'est-à-dire, **mettre l'état logique de la broche D3 au niveau « haut »**
2. Ensuite, le programme devra effectuer un test sur l'entrée analogique **A0** pour savoir si le faisceau infrarouge est coupé : **si la valeur sur A0 est > 988**, le faisceau est coupé, **il faudra allumer la led.**
3. Pour allumer la led, il suffit de **mettre l'état logique de la broche D6 au niveau « haut »**.
4. Pour éteindre la led il suffit de **mettre l'état logique de la broche D6 au niveau « Bas »**.

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Dans Mblock, n'oubliez pas d'ajouter la carte Arduino Uno comme appareil. Blocs à utiliser dans le logiciel Mblock :



Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel

Séquence 5 : Notion de capteurs / actionneurs

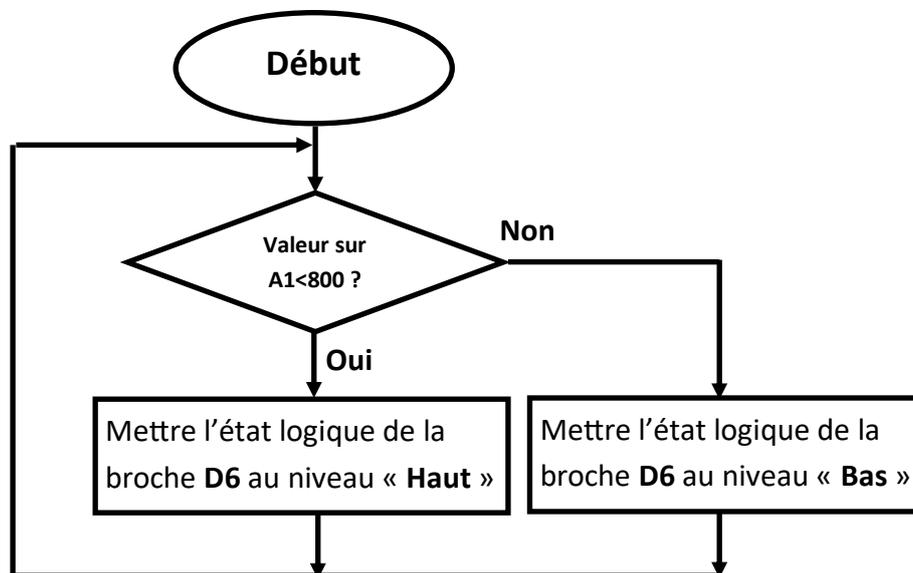
TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge

Programme 2 : allumer la led lorsqu'on appuie sur un bouton de la télécommande

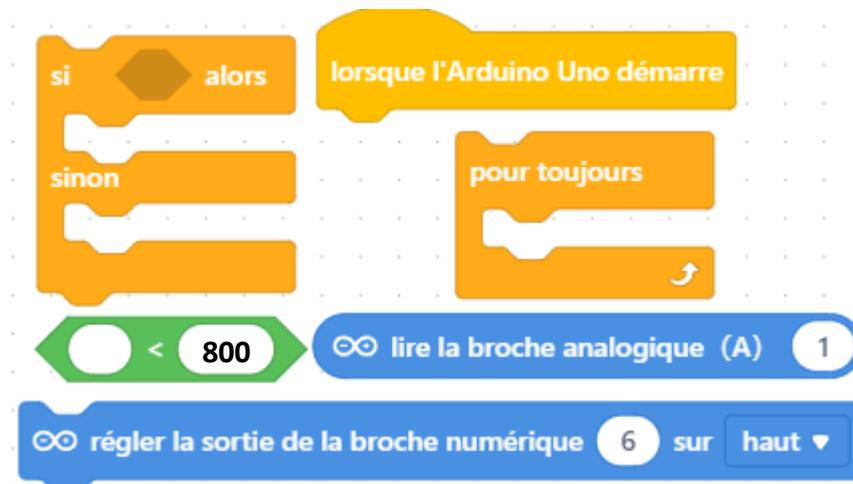
Explications :

1. Votre programme devra commencer par effectuer un test de la valeur sur l'entrée analogique A1 : **si A1 < 800** alors il faudra **allumer la led**, sinon il faudra l'éteindre.
2. **Pour allumer** la led il faut mettre l'état logique de la broche **D6 au niveau « Haut »**
3. **Pour éteindre** la led il faut mettre l'état logique de la broche **D6 au niveau « bas »**

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Blocs à utiliser dans le logiciel Mblock :



Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel

Séquence 5 : Notion de capteurs / actionneurs

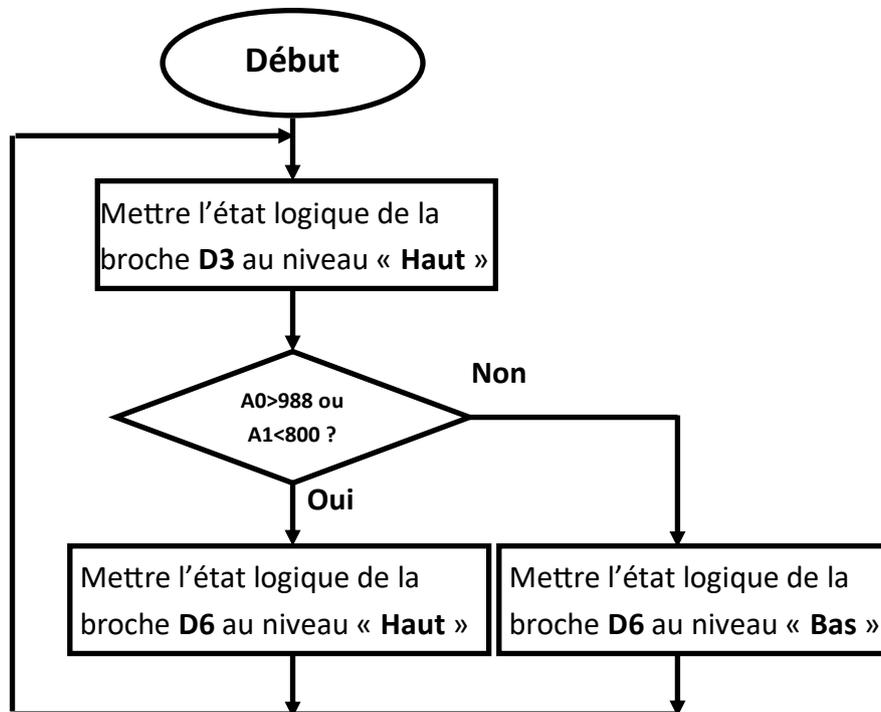
TP1 : capteur infrarouge - allumage d'une led si présence

Programme 3 : allumer la led lorsqu'on appuie sur un bouton de la télécommande ou si le faisceau infrarouge est coupé

Explications :

1. Votre programme devra commencer par activer l'émetteur infra rouge afin qu'il envoie un signal continue en direction du récepteur : c'est-à-dire, **mettre l'état logique de la broche D3 au niveau « haut »**
2. Ensuite, il faut effectuer un test pour savoir **si un bouton de la télécommande a été appuyé ou si le faisceau infrarouge est coupé** (les mêmes que précédemment)
3. Pour allumer la led il faut **mettre l'état logique de la broche D6 au niveau « Haut »**
4. Pour éteindre la led il faut **mettre l'état logique de la broche D6 au niveau « bas »**

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Blocs à utiliser dans le logiciel Mblock :



si alors lorsque l'Arduino Uno démarre

sinon pour toujours

< 800 lire la broche analogique (A) 1

réglage la sortie de la broche numérique 6 sur bas

> 988 ou

réglage la sortie de la broche numérique 3 sur haut

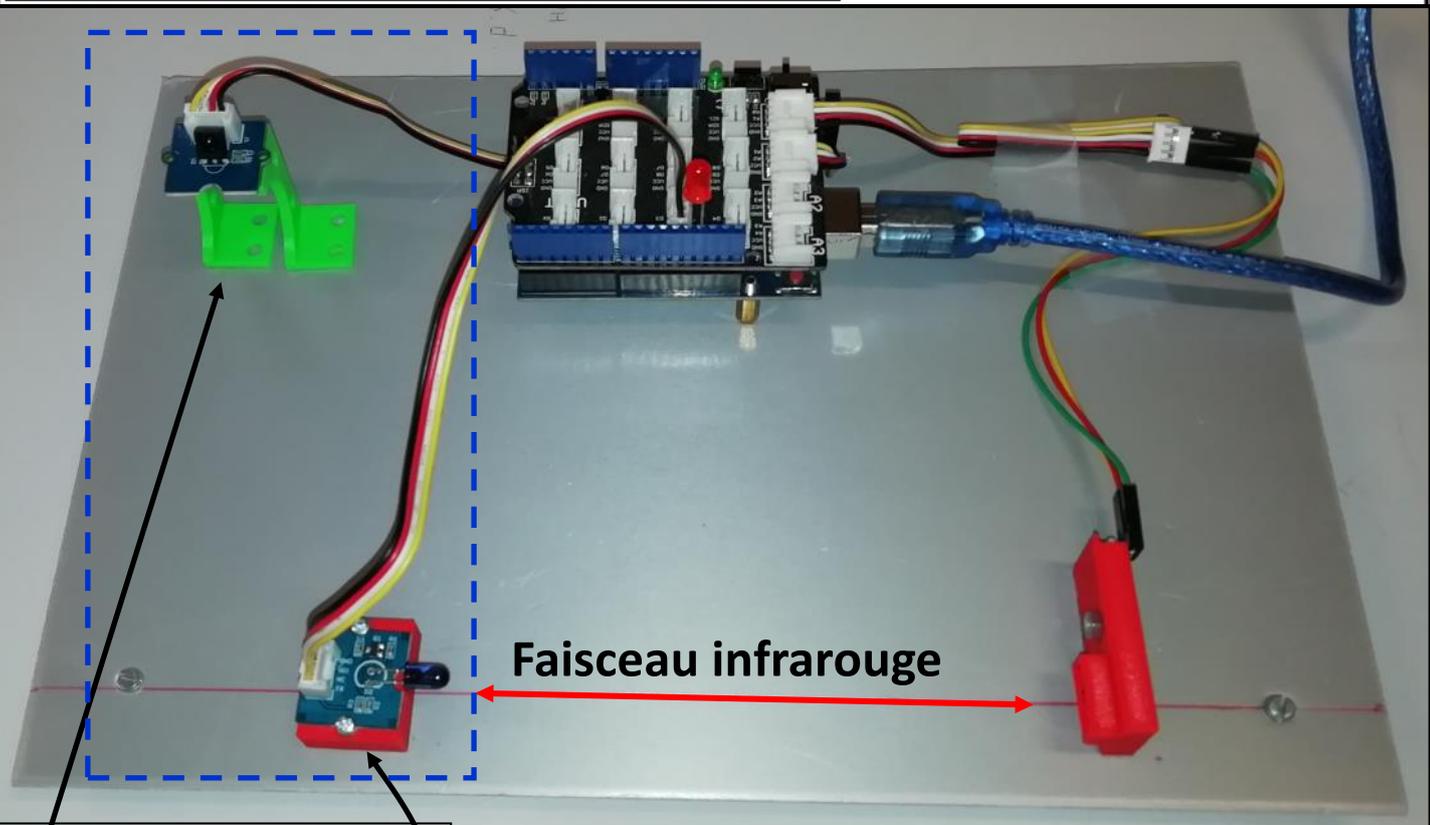
lire la broche analogique (A) 0

Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel

Séquence 5 : Notion de capteurs / actionneurs

TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge

Description de la maquette capteurs infrarouges :



Capteur infrarouge du haut détectera lorsqu'on appuiera sur une touche de la télécommande.



Il est connecté sur l'entrée analogique A1 de la carte Arduino.

Il renverra une valeur 0 quand on appuiera sur une touche de la télécommande.

Capteur infrarouge du bas détectera lorsque le faisceau infrarouge émis par l'émetteur placé en face de lui sera coupé.

Il est connecté sur l'entrée numérique D8 de la carte Arduino.

Il renverra une valeur 0 lorsque le faisceau sera coupé.

La carte Arduino pourra comprendre les signaux des capteurs infrarouges en entrée et envoyer des signaux de sortie aux leds.

Sur la carte Arduino, un **shield Grove** est installé permettant des connexions plus faciles entre les capteurs, les actionneurs et la carte Arduino.

Les 2 actionneurs :

La LED rouge connectée sur la sortie 13 de la carte Arduino.

Elle s'allumera lorsqu'un des deux capteurs aura détecté quelque chose... (télécommande ou faisceau coupé).

L'émetteur infrarouge enverra en permanence un signal infrarouge sur la capteur infrarouge du bas placé en face de lui.

Il est connecté sur la sortie D3 de la carte Arduino. I ne faut pas oublier de l'alimenter !

Travail à faire :

- 1 programme permettant d'allumer la led lorsque le faisceau entre l'émetteur et le récepteur infrarouge est coupé
- 1 programme permettant d'allumer la led lorsqu'on appuie sur une touche de la télécommande
- 1 programme permettant d'allumer la led soit en appuyant sur la télécommande ou en coupant le faisceau infrarouge.

Séquence 4 : Notion de capteurs / actionneurs

TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge



Bilan du TP n°1 Questionnaire à remplir avant de passer à la programmation :

1. Quels sont les capteurs de cette maquette ?
2. Quels sont les actionneurs de cette maquette ?
3. Sur quelle broche de la carte Arduino est connectée le capteur infrarouge qui est placé en face d'un émetteur infrarouge ?
4. Sur quelle broche de la carte Arduino est connecté le capteur infrarouge qui reçoit le signal d'une télécommande ?

Détailler les réponses aux questions suivantes :

5. Que se passe t-il lorsque j'active une télécommande ?
6. Que se passe t-il lorsque le faisceau infrarouge est coupé (avec le doigt par exemple) ?
7. Sur quel broche de la carte Arduino est connectée la Led ?
8. Quand s'allumera t-elle ?
9. Sur quel broche de la carte Arduino est connecté l'émetteur infrarouge ?
10. Quel est son rôle ?
11. Comment s'appelle la carte fixée sur la carte Arduino ?
12. Que permet-elle ?

Séquence 5 : Notion de capteurs / actionneurs

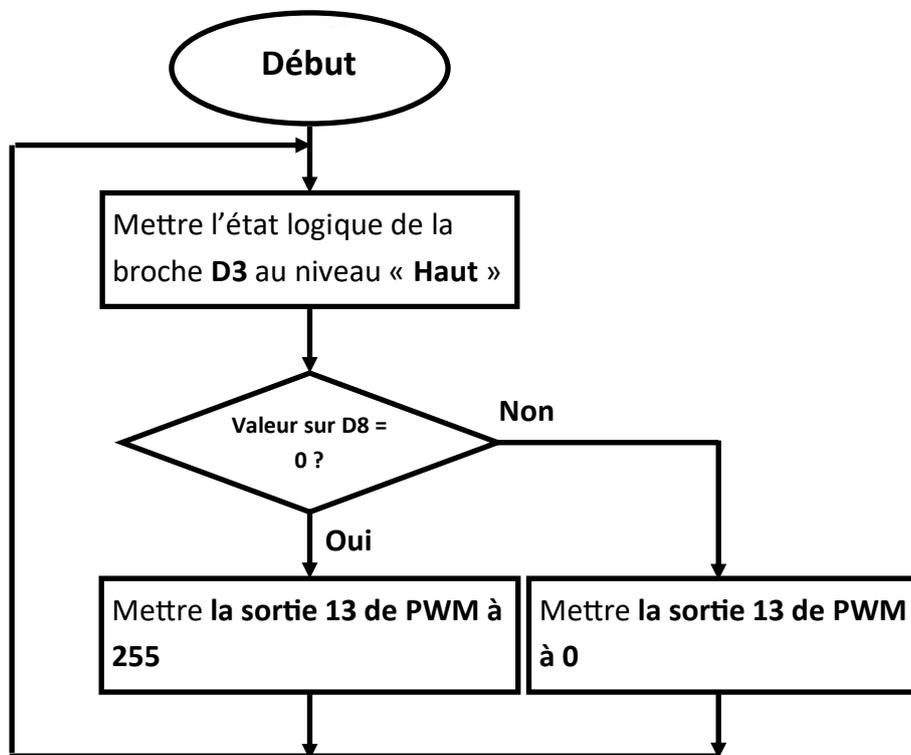
TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge

Programme 1 : allumer la led lorsque le faisceau infrarouge est coupé

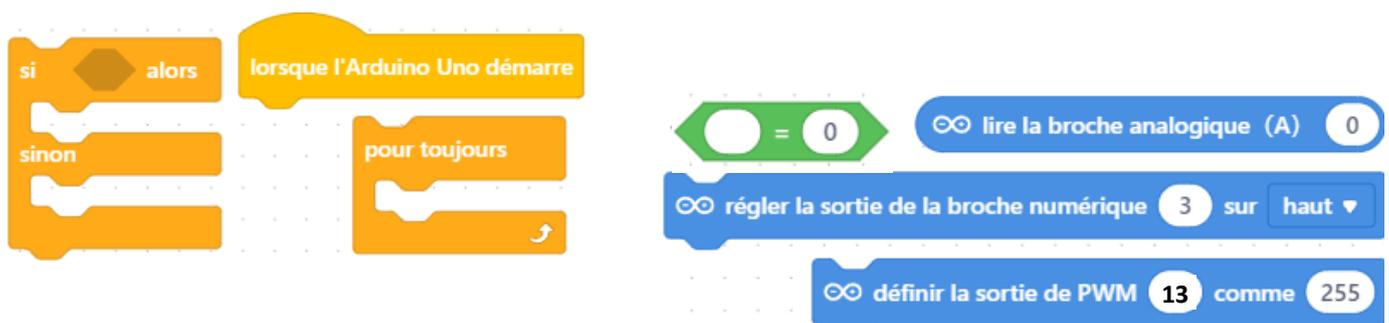
Explications :

1. Votre programme devra commencer par activer l'émetteur infra rouge afin qu'il envoie un signal continue en direction du récepteur : c'est-à-dire, **mettre l'état logique de la broche D3 au niveau « haut »**
2. Ensuite, le programme devra effectuer un test sur **l'entrée numérique D8** pour savoir si le faisceau infrarouge est coupé : **si la valeur de D8 = 0, le faisceau est coupé, il faudra allumer la led.**
3. Pour allumer la led, il suffit de **Définir la sortie 13 de PWM à 255**
4. Pour éteindre la led il suffit de **Définir la sortie 13 de PWM à 0**

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Dans Mblock, n'oubliez pas d'ajouter la carte Arduino Uno comme appareil. Blocs à utiliser dans le logiciel Mblock :



Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel

Séquence 5 : Notion de capteurs / actionneurs

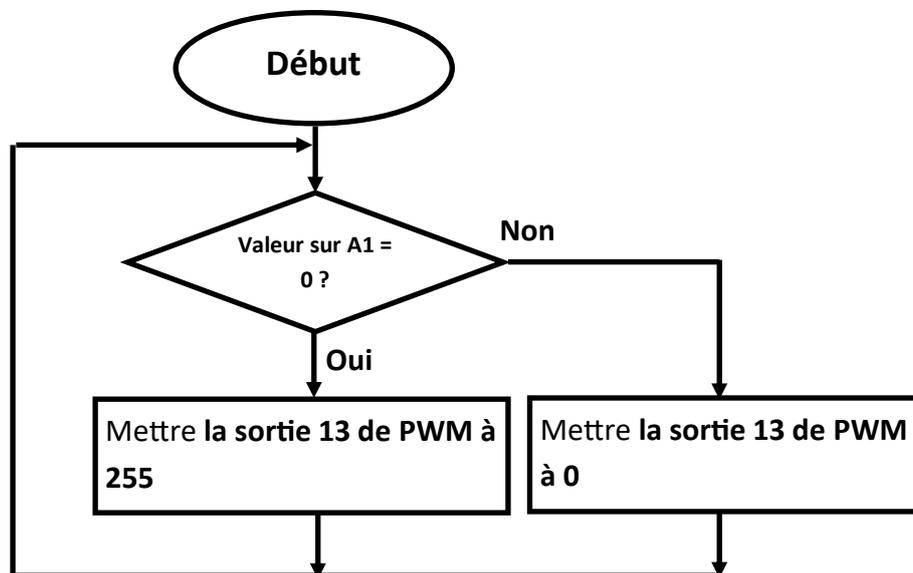
TP1 : capteur infrarouge - allumage d'une led si présence d'un signal infrarouge

Programme 2 : allumer la led lorsqu'on appuie sur un bouton de la télécommande

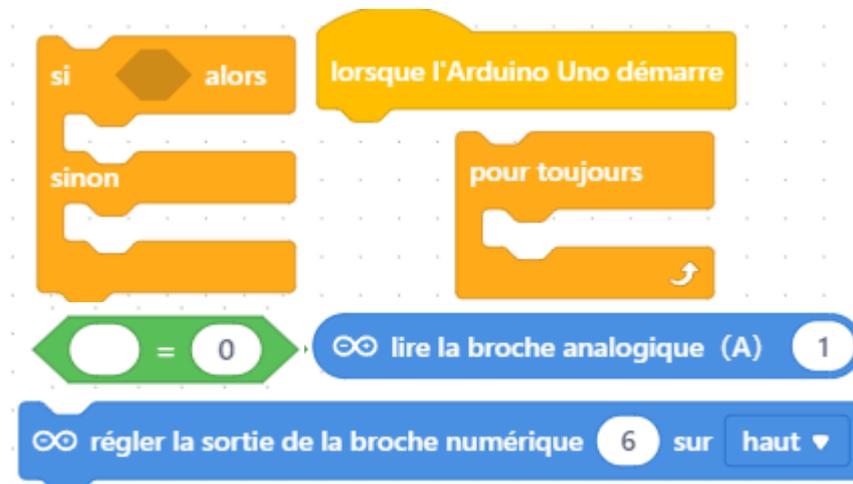
Explications :

1. Votre programme devra commencer par effectuer un test de la valeur sur l'entrée analogique A1 : **Si A1 = 0** alors il faudra **allumer la led**, sinon il faudra l'éteindre.
2. **Pour allumer** la led il faut **définir la sortie 13 de PWM à 255**
3. **Pour éteindre** la led il faut **définir la sortie 13 de PWM à 0**

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Blocs à utiliser dans le logiciel Mblock :



Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel

Séquence 5 : Notion de capteurs / actionneurs

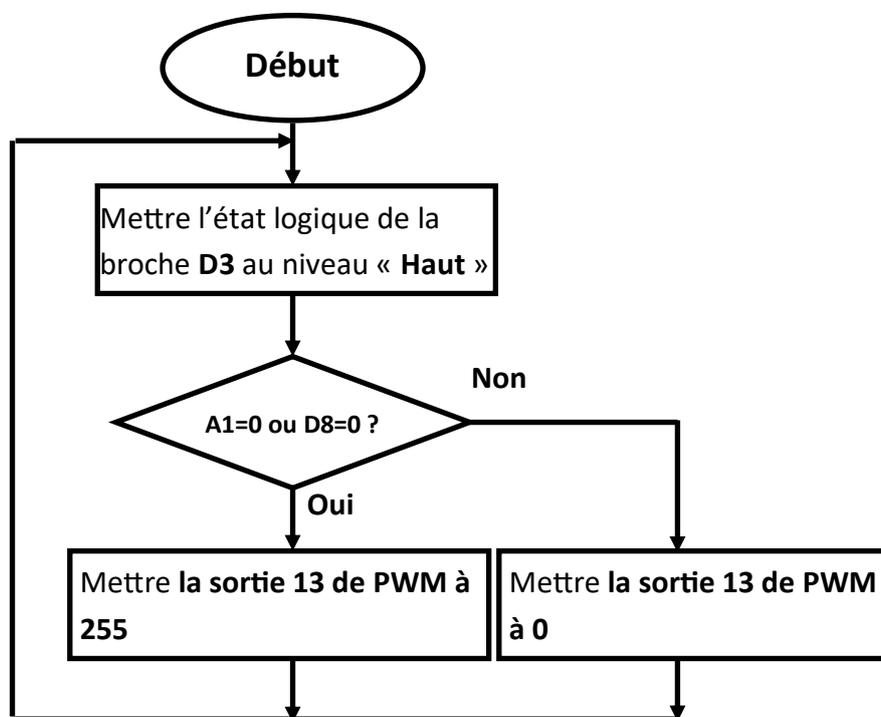
TP1 : capteur infrarouge - allumage d'une led si présence

Programme 3 : allumer la led lorsqu'on appuie sur un bouton de la télécommande ou si le faisceau infrarouge est coupé

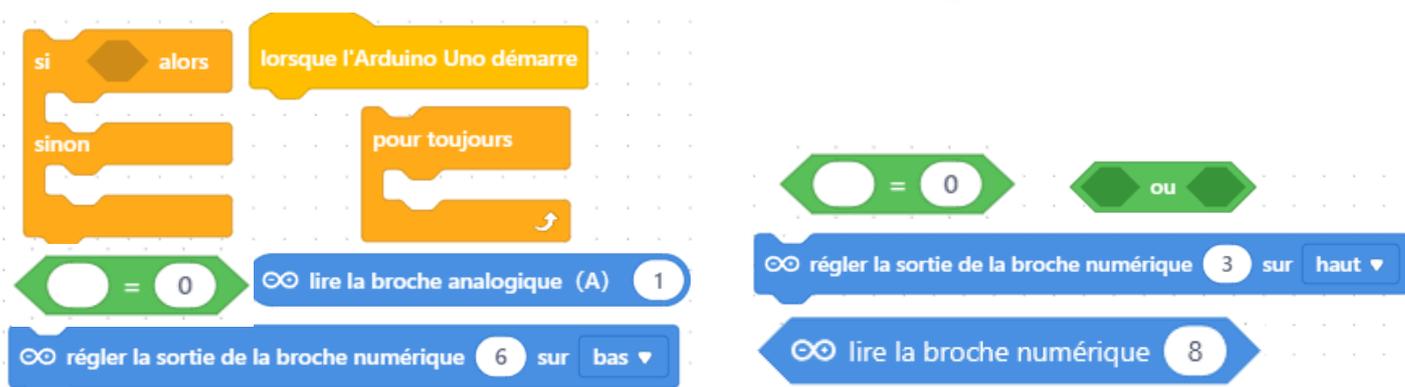
Explications :

1. Votre programme devra commencer par activer l'émetteur infra rouge afin qu'il envoie un signal continue en direction du récepteur : c'est-à-dire, **mettre l'état logique de la broche D3 au niveau « haut »**
2. Ensuite, il faut effectuer un test pour savoir **si un bouton de la télécommande a été appuyé ou si le faisceau infrarouge est coupé** (les mêmes que précédemment)
3. Pour allumer la led il faut **définir la sortie 13 de PWM à 255**
4. Pour éteindre la led il faut **définir la sortie 13 de PWM à 0**

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Blocs à utiliser dans le logiciel Mblock :



The screenshot shows several blocks from the Mblock software:

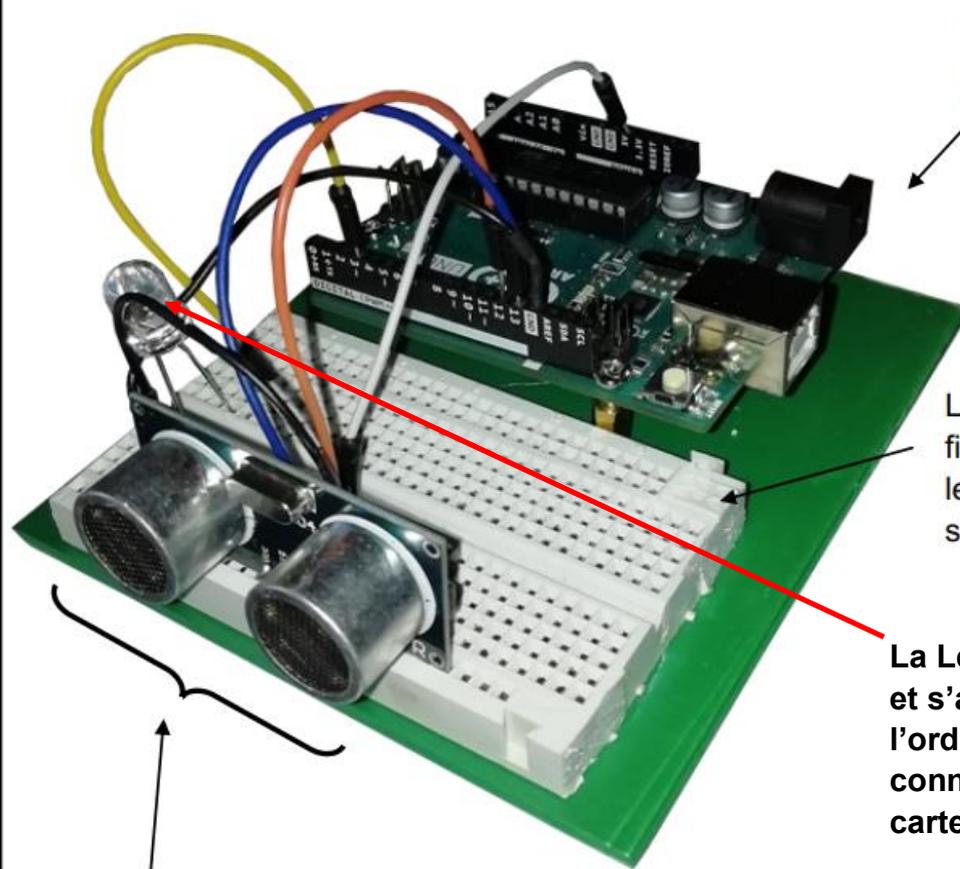
- Control blocks: "si... alors... sinon" (if... then... else), "lorsque l'Arduino Uno démarre" (when Arduino Uno starts), "pour toujours" (forever loop).
- Logic blocks: Comparison " $= 0$ ", Logical operator "ou" (or).
- Action blocks: "régler la sortie de la broche numérique 6 sur bas" (set digital pin 6 to low), "régler la sortie de la broche numérique 3 sur haut" (set digital pin 3 to high), "lire la broche analogique (A) 1" (read analog pin 1), "lire la broche numérique 8" (read digital pin 8).

Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel

Séquence 5 : Notion de capteurs / actionneurs

TP2 : capteur à ultrason - mesure d'une distance

Description de la maquette capteur à ultrason :



La carte Arduino sur laquelle sont connectés les fils qui proviennent de la plaquette Lab

La plaquette Lab sur laquelle sont fixés la led, le capteur à ultrason et les fils permettant le transfert des signaux à la carte Arduino

La Led est l'actionneur du système et s'allumera lorsqu'elle en recevra l'ordre de la carte Arduino. Elle est connectée à la broche **D3** de la carte Arduino

Le capteur à ultrason qui permet de mesurer la distance entre lui et un obstacle devant lui...

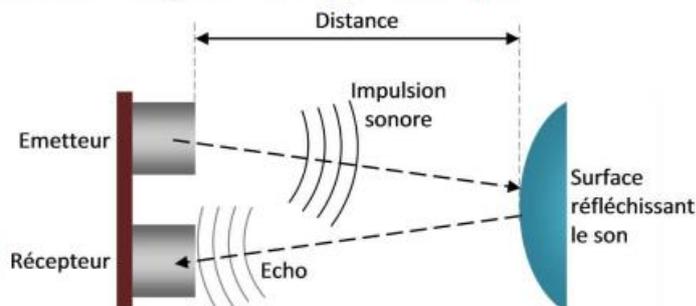
Connexions des broches (de bas en haut) :

- GND sur la broche GND
- TRIG sur la broche 12
- ECHO sur la broche 11
- VCC sur 5V

Principe d'un capteur à ultrason :

Ils utilisent le principe de l'**écho** pour déterminer la distance à laquelle se trouve un objet :

1. Un court signal sonore est envoyé (inaudible car dans le domaine des ultrasons)
2. Le son est réfléchi par une surface et repart en direction du capteur
3. Ce dernier le détecte, une fois revenu à son point de départ



C'est en mesurant la durée entre l'instant d'émission et l'instant de la réception de l'écho du signal que le programme pourra connaître la distance entre le capteur et la surface réfléchissant.

Séquence 4 : Notion de capteurs / actionneurs

TP2 : capteur à ultrason - mesure d'une distance



Bilan du TP n°2

Questionnaire à remplir avant de passer à la programmation :

1. Quel est le capteur de cette maquette ?
2. Sur quelles broches de la carte Arduino est connecté le capteur de cette maquette ?
3. Quel est l'actionneur de cette maquette ?
4. Sur quelle broche de la carte Arduino est il connecté ?

Détailler les réponses aux questions suivantes :

5. Que se passe t-il lorsque je place ma main devant le capteur ?
6. Expliquer en détails ce qu'il se passe quand la led s'allume :
7. Que peut permettre ce capteur placé à l'avant d'un robot ?
8. Donner d'autres applications que peut permettre ce capteur :
9. Rappeler rapidement le fonctionnement de ce capteur :

Séquence 5 : Notion de capteurs / actionneurs

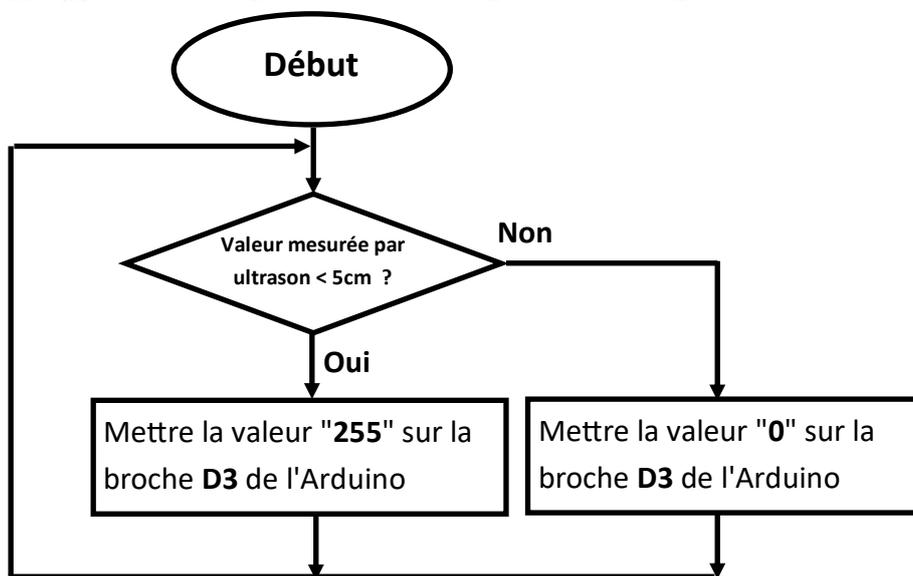
TP2 : capteur à ultrason - mesure d'une distance

Programme 4 : allumer la led lorsqu'un obstacle s'approche à moins de 5 cm du capteur

Explications :

1. Votre programme devra commencer par **afficher sur le port série la « distance mesurée par ultrason »**
2. Ensuite il effectuera un test de la distance mesurée par le capteur à ultrason
3. Pour allumer la led en mettant l'état logique de la broche D3 au niveau « Haut » si < 5 cm
4. **Ou** pour éteindre la led en mettant l'état logique de la broche D3 au niveau « bas » si > 5 cm

Organigramme du programme : il permet de suivre, dans l'ordre, les différentes étapes du programme



Blocs à utiliser dans le logiciel Mblock :

Quand le programme est terminé sur Mblock, enregistrez-le dans votre espace personnel



1/ Présentation :

Un servomoteur, contrairement à un moteur électrique classique, sera capable de faire tourner son axe selon un angle précis (de 0° à 180°). Ainsi, dans notre exemple, il pourra faire monter ou descendre la barrière, suivant les conditions suivantes :



180° : barrière fermée

90° : barrière ouverte

2/ Découverte et connexion du servomoteur :

Ce servomoteur dispose de 3 fils de connexion sur la carte Arduino (marron, rouge et jaune) :

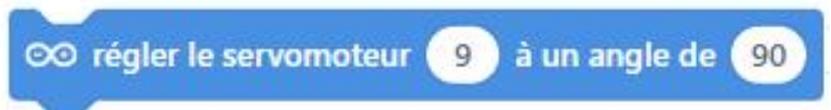
- Le **marron** est connecté sur la broche **GND** de la carte Arduino, il représente le **moins de l'alimentation électrique (-)**.
- Le **rouge** est connecté sur la broche **VCC** de la carte Arduino, il représente le **plus de l'alimentation électrique (+)**.
- Le **jaune** est connecté sur la broche **D9** de la carte Arduino, c'est le fil par lequel la carte Arduino pourra **commander** le servomoteur.

2/ Bloc de programmation du servomoteur :

voici le bloc qu'il faut utiliser dans le logiciel

Mblock pour piloter le servomoteur en pré-

sant bien le n° de la broche sur laquelle on a relié le fil jaune du servomoteur (**D9**), ainsi que l'angle souhaité (90° par exemple pour faire monter la barrière). Vous trouverez ce bloc dans la catégorie « **Broches** » lorsque la carte Arduino est installée comme « **Appareil** »



Problème : si on utilise ce bloc seul, le servomoteur effectuera un mouvement trop rapide risquant d'endommager la barrière fixée dessus.



Solution : faire en sorte que le servomoteur soit plus lent dans le déplacement de son axe comme par exemple faire monter ou descendre la barrière d'un degré à la fois...

Mais pour lui faire faire une rotation de 90° il faudra trouver une méthode plus rapide pour le programme... En utilisant une variable par exemple...



Bilan du TP n°3

Questionnaire à remplir avant de passer à la programmation :

1. Quel est l'actionneur de cette maquette ?

2. Sur quelles broches de la carte Arduino est connecté l'actionneur de cette maquette ?

3. Quel est le rôle de cet actionneur ?

4. Donner d'autres applications que peut permettre cet actionneur :

5. Quelle est la broche de la carte Arduino qui permettra la commande de cet actionneur ?

6. Quel est le problème si on utilise ce bloc seul ?



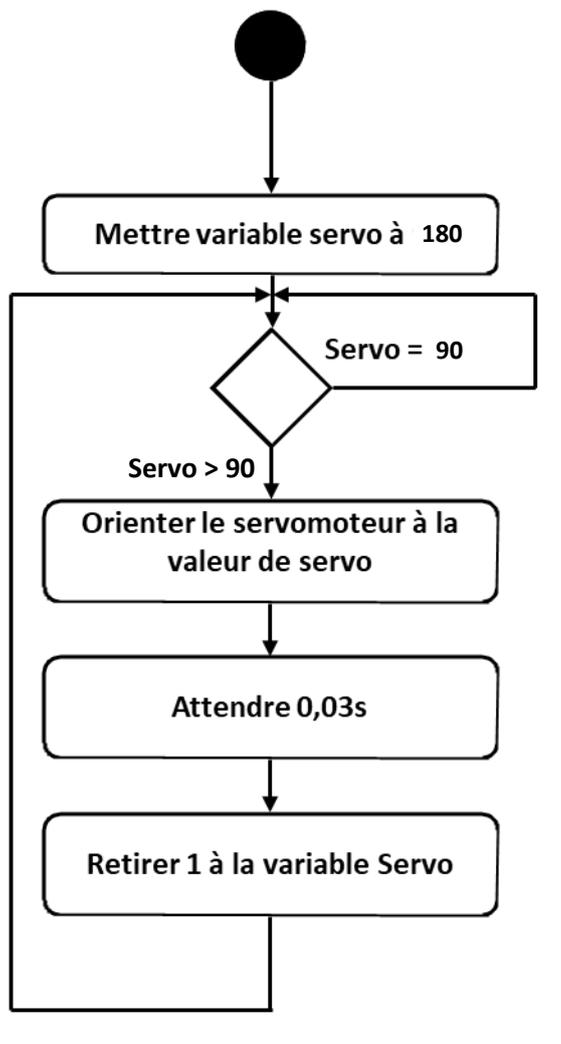
∞ régler le servomoteur 9 à un angle de 90

7. Quelle serait la solution pour lever une barrière à 90° ?

3/ Programmons la montée de la barrière :

Ouvrez le logiciel **Mblock** et remettez dans l'ordre les blocs fournis conformément au diagramme d'activité (en suivant l'ordre des flèches) afin de réaliser le programme :

Diagramme d'activité « monter barrière »



- 1 rectangle = une action
- 1 losange = test (répéter jusqu'à)
- Le rond noir indique le début du programme

Le bloc « **servo** » est une **variable**, il faut la créer, pour cela, ouvrez la catégorie « **Variables** » et appuyez sur le bouton « **Créer une variable** ». La fenêtre ci-dessous s'ouvre, tapez le nom « **servo** » et



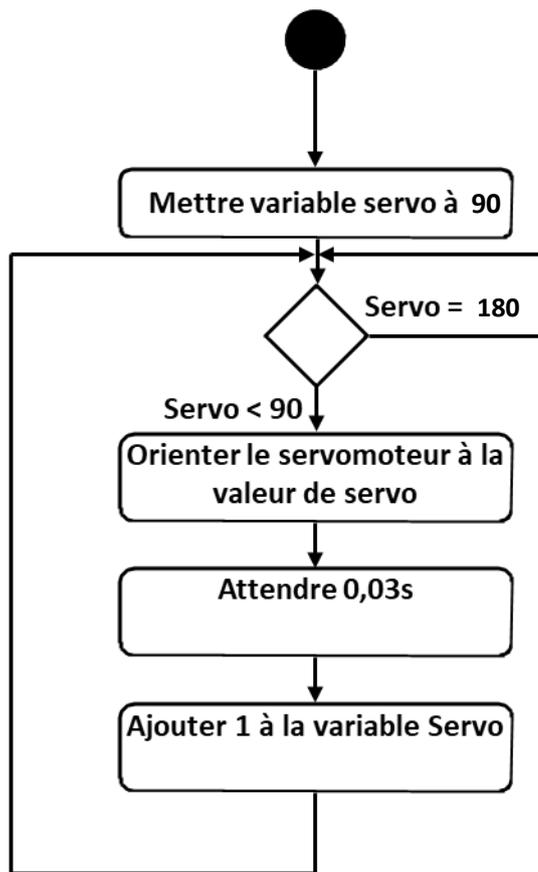
Lorsque vous avez terminé le programme, enregistrez-le dans votre espace



3/ Programmons la descente de la barrière :

Ouvrez le logiciel **Mblock** et remettez dans l'ordre les blocs fournis conformément au diagramme d'activité (en suivant l'ordre des flèches) afin de réaliser le programme :

Diagramme d'activité « descendre barrière »



- 1 rectangle = une action
- 1 losange = test (répéter jusqu'à)
- Le rond noir indique le début du programme

Le bloc « **servo** » est une **variable**, il faut la créer, pour cela, ouvrez la catégorie « **Variables** » et appuyez sur le bouton « **Créer une variable** ». La fenêtre ci-dessous s'ouvre, tapez le nom « **servo** »

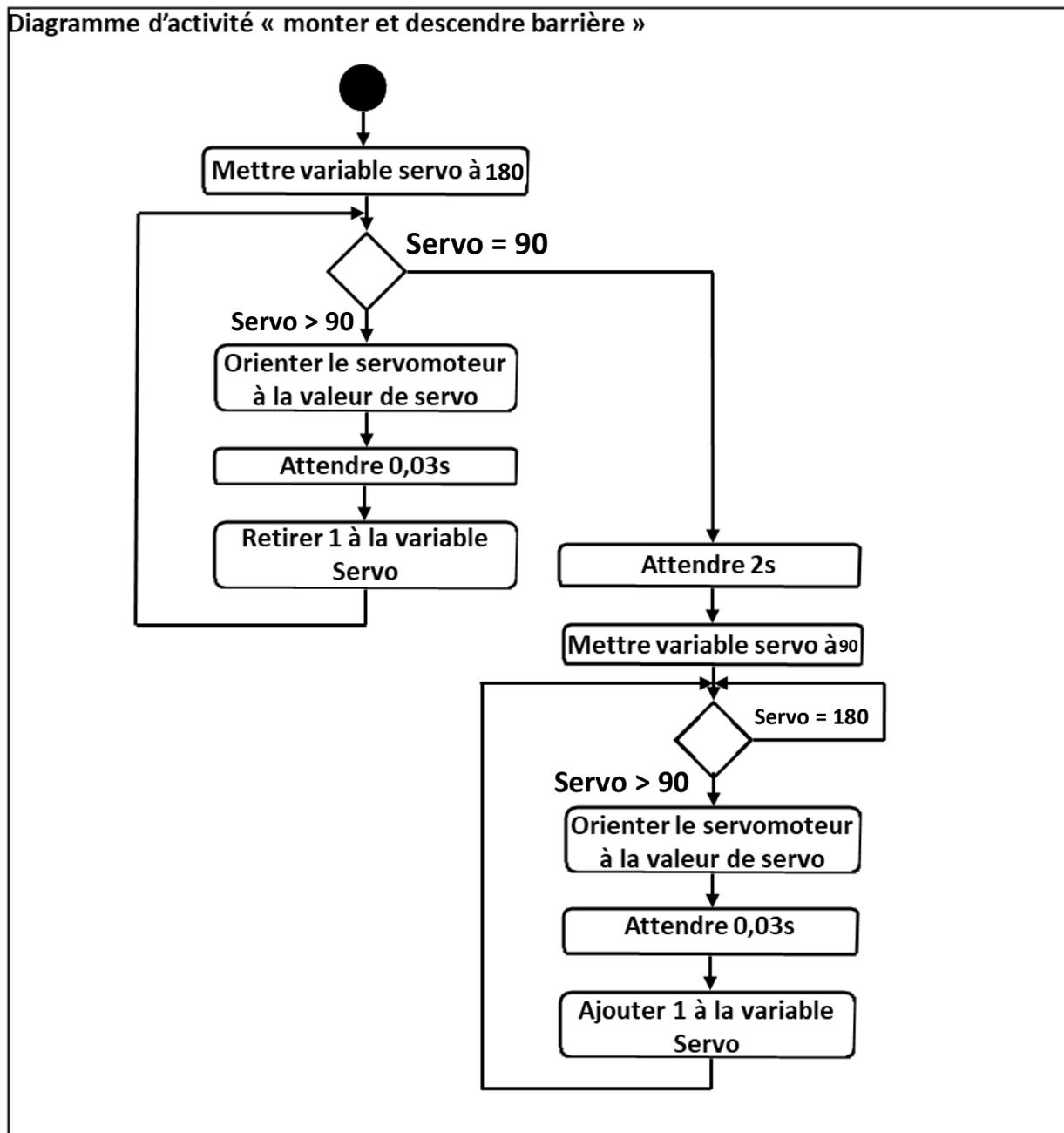


Lorsque vous avez terminé le programme, enregistrez-le dans votre espace.



5/ Programmons la montée et, 2s plus tard, la descente de la barrière : utilisez les mêmes blocs...

Diagramme d'activité « monter et descendre barrière »



lorsque l'Arduino Uno démarre

servo

répéter jusqu'à

régler le servomoteur 9 à un angle de servo

attendre 0.03 secs

définir servo à 0

ajouter 1 à servo

90

servo

répéter jusqu'à

régler le servomoteur 9 à un angle de servo

attendre 0.03 secs

définir servo à 90

ajouter -1 à servo

180

Séquence 5 : Notion de programmation de systèmes - TP4

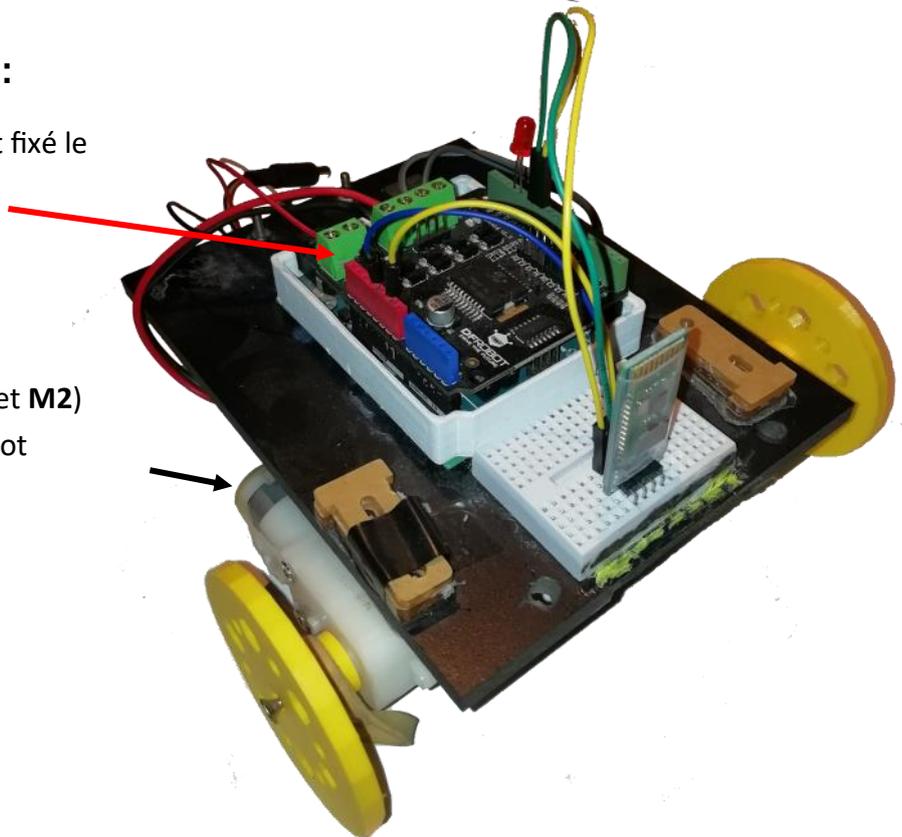
Programmer des actionneurs et faire faire des mouvements à un robot



Description de la maquette :

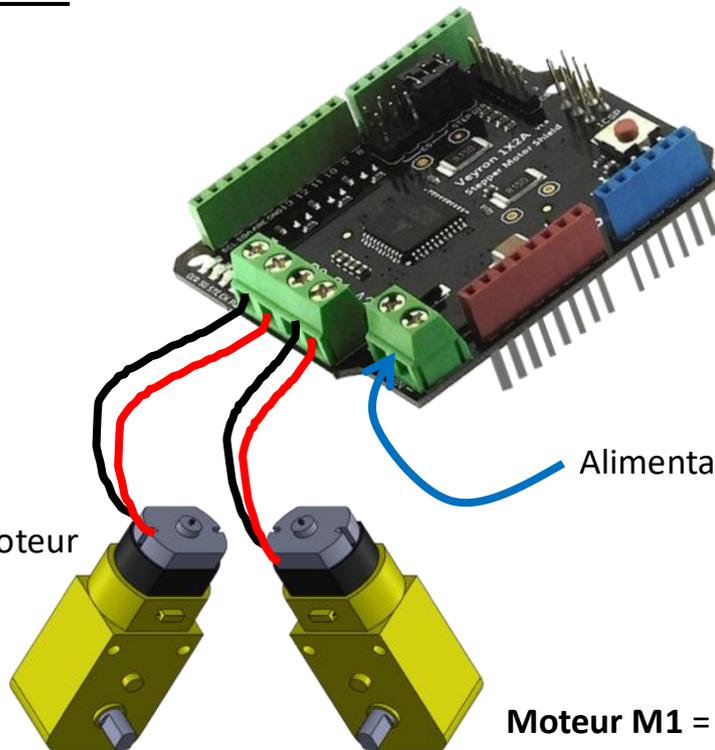
La carte **Arduino Uno** sur laquelle est fixé le **shield motor** (commande moteurs)

Les deux actionneurs (moteurs **M1** et **M2**) qui permettront de propulser le robot



Connexions du shield mo- teur :

Moteur M2 = moteur
gauche



Alimentation externe

Moteur M1 = moteur
droit

Séquence 5 : Notion de programmation de systèmes - TP4

Programmer des actionneurs et faire faire des mouvements à un robot



Bilan du TP n°4

Questionnaire à remplir avant de passer à la programmation :

1. Quels sont les actionneurs de cette maquette ?

2. Quel est leur rôle ?

3. Donner d'autres applications que peuvent permettre ces actionneurs :

4. Quelles sont les broches de la carte Arduino qui permettront la commande de ces actionneurs

Lire la page suivante avant de répondre aux questions 5 et 6 :

5. Quelles sont les broches qui permettront la commande du sens de la rotation des moteurs ?

6. Quelles sont les broches qui permettront la commande de la vitesse des moteurs ?

7. Que permet le shield fixé sur la carte Arduino ?

Séquence 5 : Notion de programmation de systèmes - TP4

Programmer des actionneurs et faire faire des mouvements à un robot



Programme 1 : faire faire des mouvements à un robot

Pour programmer un moteur électrique avec une carte arduino, il est nécessaire d'employer deux blocs dans le logiciel Mblock :

- Un bloc pour programmer la vitesse de rotation (de 0 à 255)
- Un bloc pour programmer le sens de rotation (0 = sens horaire ; 1 = sens antihoraire)

Moteur M2 gauche	
Broche D7	Broche 6 (PWM)
Haut = sens droite (Horaire)	De 0 à 255 avec : 0 = Arrêt
Bas = sens gauche (anti Horaire)	255 = vitesse maximale

Moteur M1 droit	
Broche D4	Broche 5 (PWM)
Haut = sens droite (Horaire)	De 0 à 255 avec : 0 = Arrêt
Bas = sens gauche (anti Horaire)	255 = vitesse maximale

Donc d'après le tableau ci-dessus :

Le moteur M1 (moteur droit) se pilote en programmant :

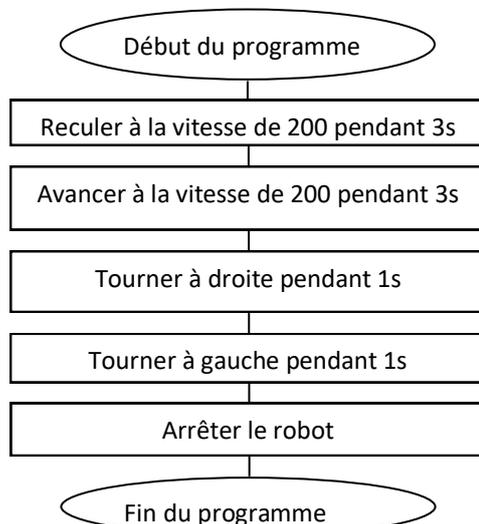
- La broche D4 de la carte Arduino pour le sens de rotation («Haut » = sens horaire ; « Bas » = sens anti horaire)
- La broche 5 (PWM) de la carte Arduino pour la vitesse de rotation (de 0 à 255)

Le moteur M2 (moteur gauche) se pilote en programmant :

- La broche D7 de la carte Arduino pour le sens de rotation («Haut » = sens horaire ; « Bas » = sens anti horaire)
- La broche 6 (PWM) de la carte Arduino pour la vitesse de rotation (de 0 à 255)

Algorithme :

Début du programme
 Reculer tout droit pendant 3 secondes (vitesse 150)
 Avancer tout droit pendant 3 secondes (vitesse 150)
 Tourner à droit pendant 1 seconde
 Tourner à gauche pendant 1 seconde
 Arrêter le robot



Faites le programme en utilisant les blocs suivants (certains seront à dupliquer)

☞ définir la sortie de PWM 5 comme 255

lorsque l'Arduino Uno démarre

☞ régler la sortie de la broche numérique 4 sur haut

attendre 1 secs

☞ définir la sortie de PWM 6 comme 0

☞ régler la sortie de la broche numérique 7 sur bas

attendre 3 secs

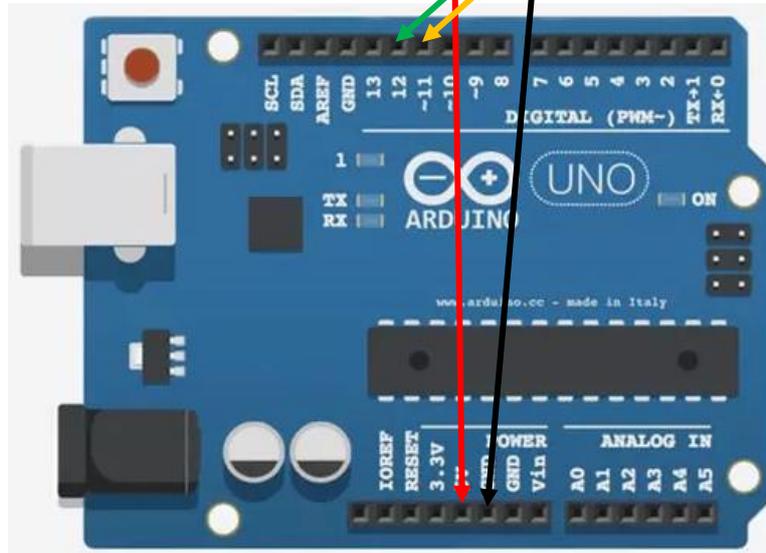
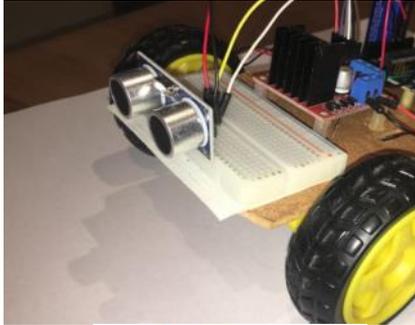
Séquence 5 : Notion de programmation de systèmes - TP4

Programmer des actionneurs et faire faire des mouvements à un robot

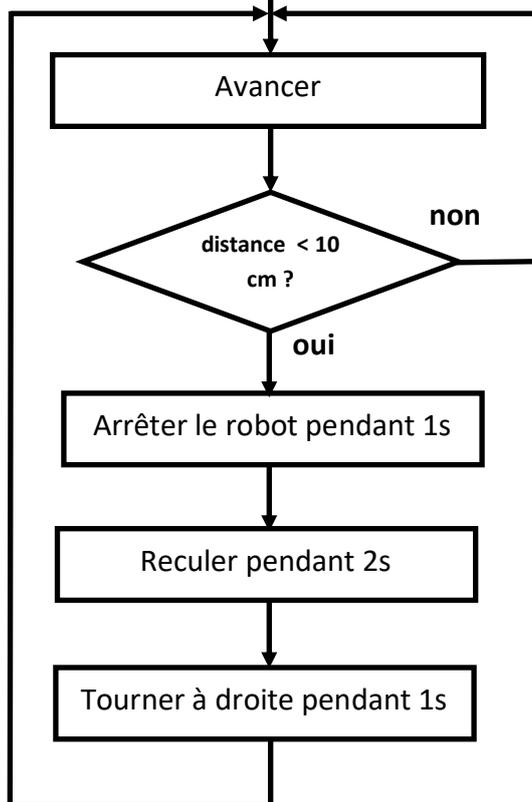


Programme 2 : robot détecteur d'obstacle

A l'aide de fils de connexion arduino, relier un capteur ultrason à la carte Arduino qui pilote les moteurs du robot comme indiqué ci-dessous :



Début



lire les broches TRIG 12 et ECHO 11 du capteur ultrasonique

définir la sortie de PWM 5 comme 255

lorsque l'Arduino Uno démarre

< 10

régler la sortie de la broche numérique 4 sur haut

attendre 1 secs

attendre 2 secs

définir la sortie de PWM 6 comme 0

régler la sortie de la broche numérique 7 sur bas