

Cycle 4

Classe de 5^{ème}

Séquence 5

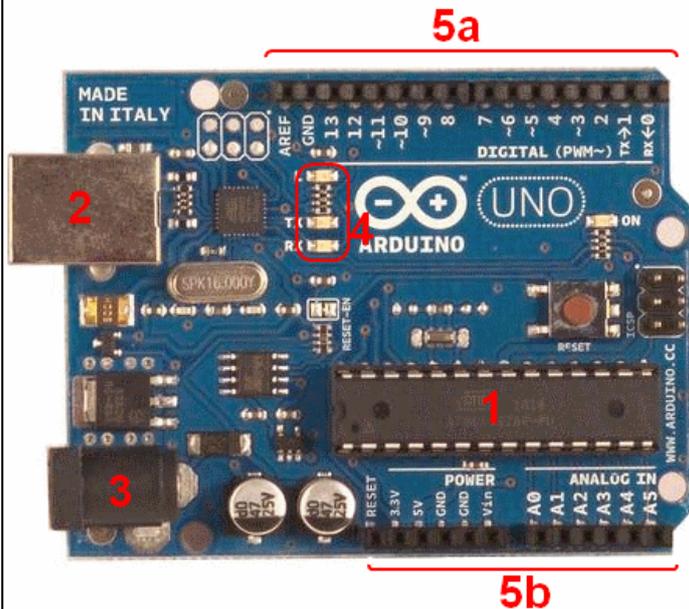
Comment commander un éclairage automatique ?

- Initiation à la programmation de carte Arduino Uno
- Clignotement d'une led
- Clignotement de 3 leds (Chenillard)
- Commander un éclairage en fonction de la luminosité

TECH5.4 - Piloter un système connecté localement ou à distance.	
TECH5.5 - Modifier ou paramétrer le fonctionnement d'un objet communicant.	IP2.2
TECH8.5 - Analyser le comportement attendu d'un système réel et décomposer le problème posé en sous-problèmes afin de structurer un programme de commande	IP2.1
TECH8.6 - Écrire, mettre au point (tester, corriger) et exécuter un programme commandant un système réel et vérifier le comportement attendu.	IP2.2
TECH8.7 - Écrire un programme dans lequel des actions sont déclenchées par des événements extérieurs.	IP2.3

Qu'est-ce qu'une carte Arduino Uno ?

D'abord, La carte Arduino, comme un ordinateur, est une carte électronique *qui ne sait rien faire* sans qu'on lui dise *quoi faire*, elle a besoin d'un **programme** pour fonctionner. Elle dispose de connecteurs (**5a** et **5b**) qui lui permettront de recevoir des informations provenant de capteurs (capteur de luminosité par exemple) ou d'envoyer des informations vers des actionneurs (Led, Buzzer, Moteur).



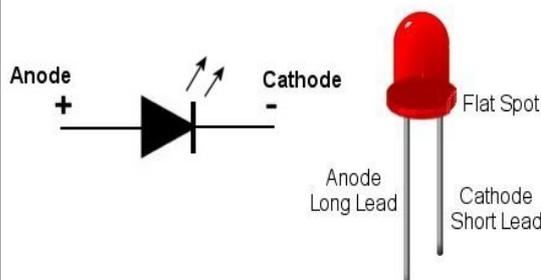
L'élément essentiel de cette carte est le **microcontrôleur (1)**, c'est le composant qui pourra interpréter les informations reçues sur les connecteurs et qui exécutera le programme que la carte recevra par le connecteur USB (**2**).

Le connecteur (**3**) permettra de brancher une alimentation externe afin de rendre l'ensemble autonome.

Cette carte, comme le logiciel qui permet de la programmer, est « Open Source », c'est-à-dire, que les plans électroniques de la carte, les composants utilisés comme le code source du programme sont disponibles et téléchargeables gratuitement. De ce fait, des logiciels comme scratch, Mblock, ArduBlock, etc... sont apparus permettant ainsi la programmation de cette carte très facilement par l'imbrication de blocs.

TP n°1 : faire clignoter une Led : Une Led (*diode électroluminescente*) est un composant électronique. Elle s'allume lorsqu'elle est traversée par un courant électrique. C'est un composant polarisé c'est-à-dire que l'on doit respecter un sens de branchement :

- Le **plus** de l'alimentation (une pile par exemple) doit être relié à l'**anode** de la Led



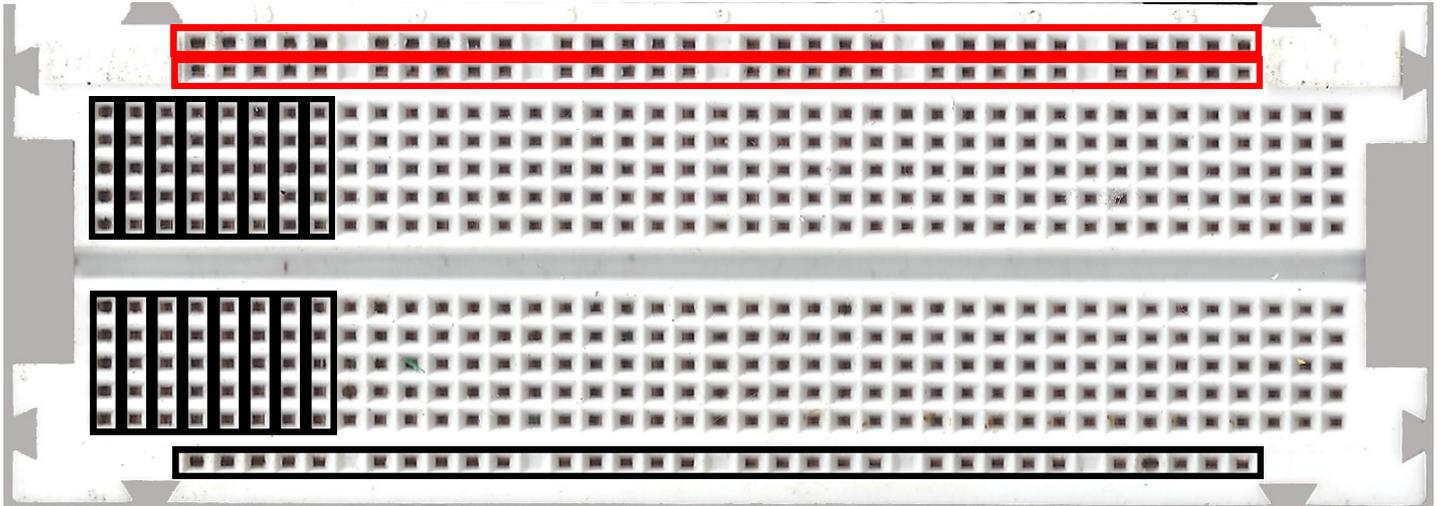
- Le **moins** de l'alimentation doit être relié à la **cathode** de la Led.

La cathode se reconnaît par un méplat sur le côté de la Led et généralement, l'anode est la patte de la led la plus longue.

Pour notre montage, nous utiliserons aussi une résistance qui aura pour rôle de limiter l'intensité du courant dans la Led. Ce composant n'a pas de sens de branchement.

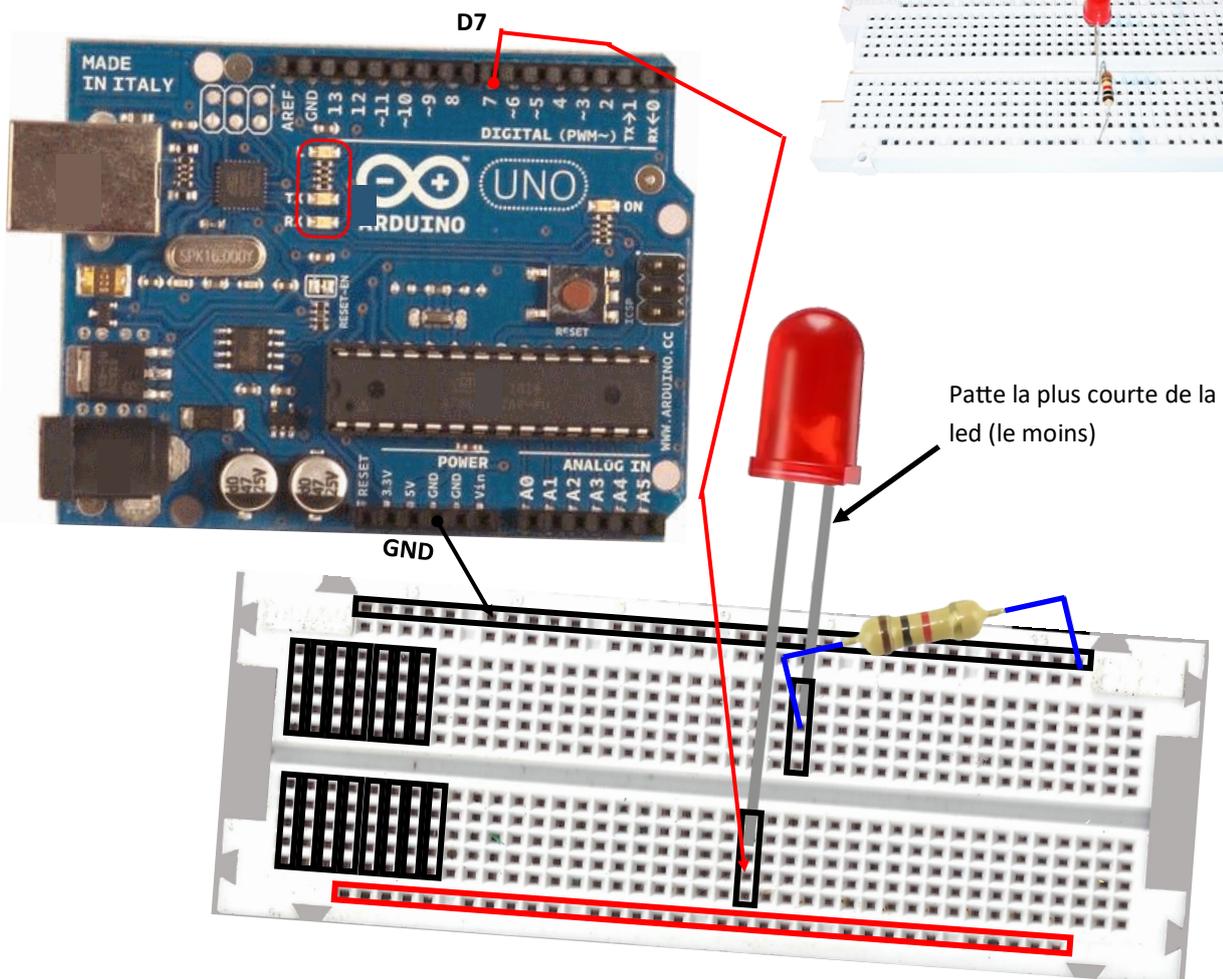
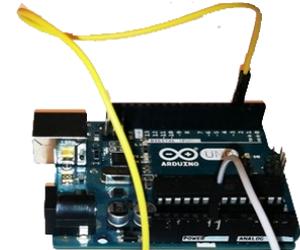


Nous utiliserons, pour relier les composants entre-eux, une plaquette Lab. Elle permet de réaliser des prototypes de montages électroniques. On s'en sert lorsque l'on veut tester des nouveau circuit électroniques avant de réaliser les cartes définitives.



- Tous les trous de chaque cadre rouge horizontal sont en contact
- Tous les trous du cadre noir horizontal sont en contact.
- Tous les trous de chaque cadre noir vertical sont en contact

Montage du TP n°1 : Placer la Led et la résistance sur la plaquette Lab de la façon suivante puis brancher la carte Arduino :



Programme du TP n°1 :

La Led est reliée à l'entrée digitale **D3** (PWM~ 3) de la carte Arduino. Pour faire clignoter la Led, notre programme doit donc effectuer **indéfiniment** la boucle suivante :

1. envoyer un signal électrique sur la broche PWM~ 3

définir la sortie de PWM 3 comme 255

2. attendre un certain temps (assez court : 0,3 seconde par exemple)

attendre 0.3 secondes

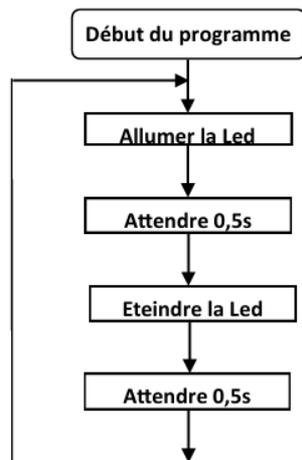
3. éteindre la Led.

définir la sortie de PWM 3 comme 0

pour toujours

Ces blocs « **Arduino** » se trouvent dans la catégorie **Pilotage** du logiciel Mblock.

Organigramme :

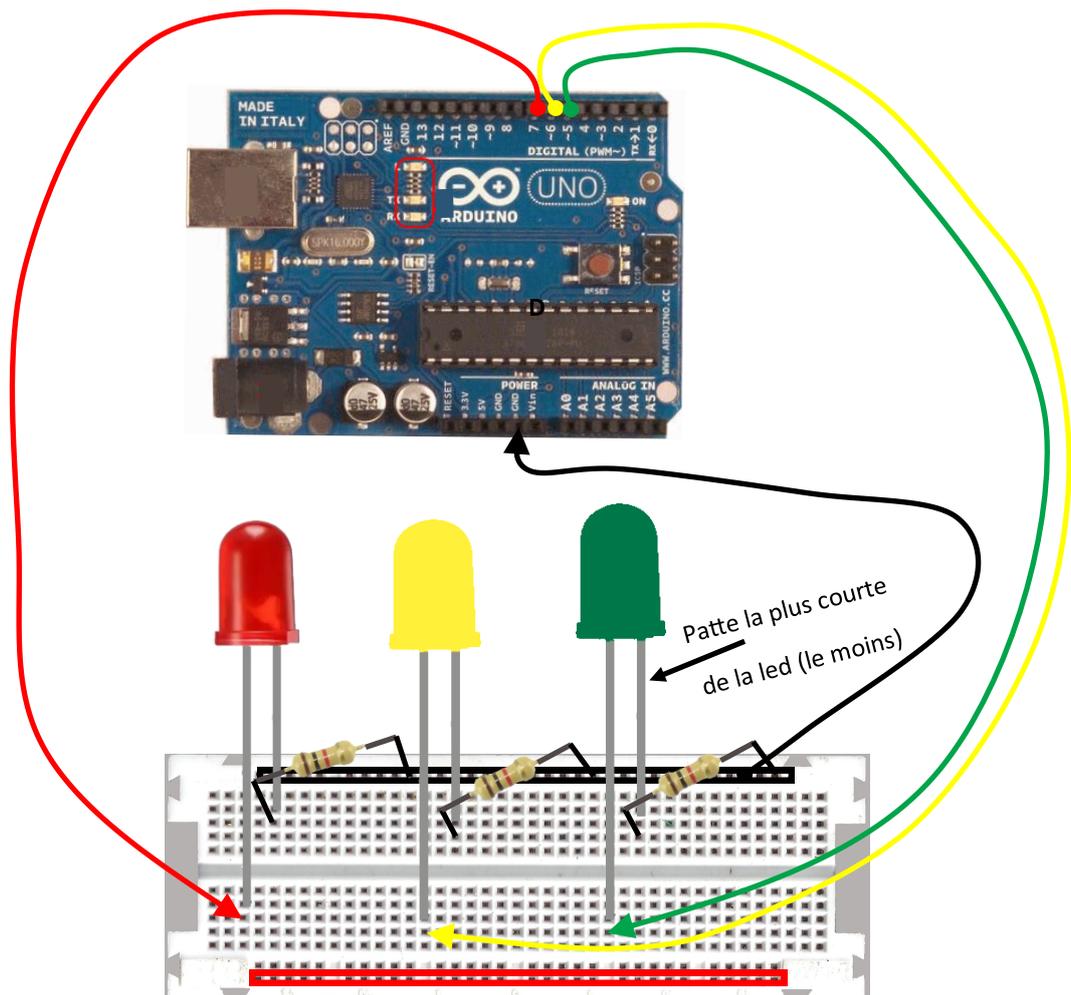


Créez le programme avec Mblock et enregistrez le dans votre espace perso

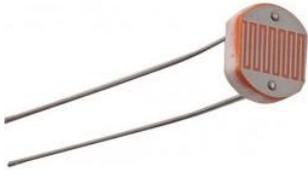
lorsque l'Arduino Uno démarre

TP n°2 : faire clignoter plusieurs Leds :

1. Débrancher les éléments du premier TP.
2. Effectuer le montage avec 3 Leds
3. Faire le programme permettant de faire clignoter les 3 Leds



TP n°3 : Utilisation d'une photorésistance : lorsque la luminosité diminue, il est temps d'allumer la lumière...

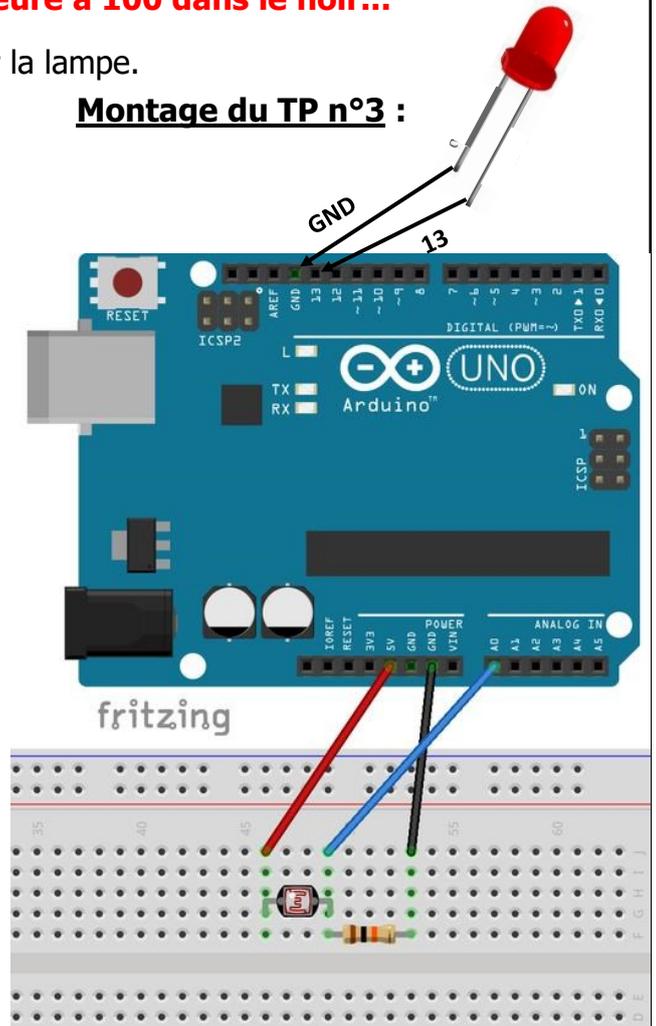
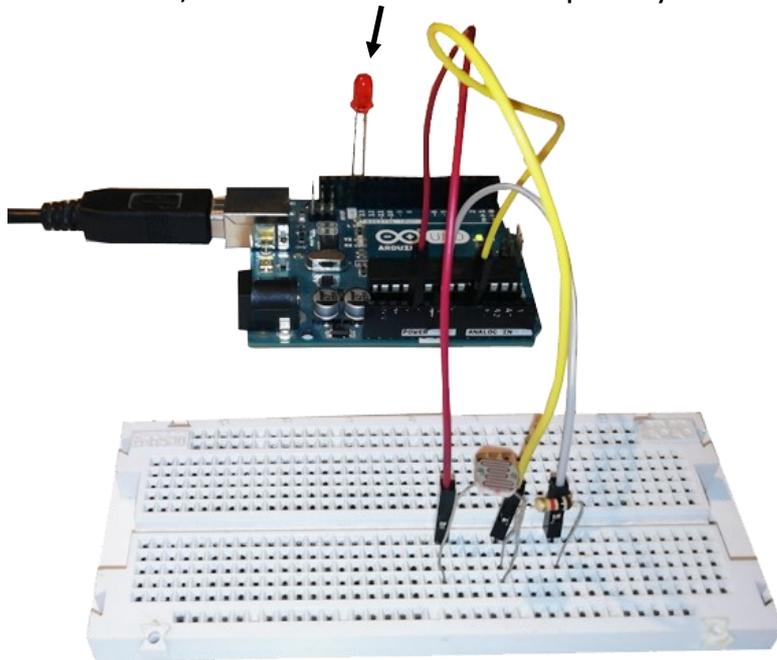


Une **photorésistance** est un composant dont la résistivité dépend de la luminosité ambiante. Pour faire simple, c'est une résistance dont la **valeur change en fonction de la lumière qu'elle reçoit**.

La photorésistance que nous utiliserons enverra une valeur supérieure à 400 en pleine lumière et une valeur inférieure à 100 dans le noir...

Dans notre TP, nous utiliserons une **LED** pour symboliser la lampe.

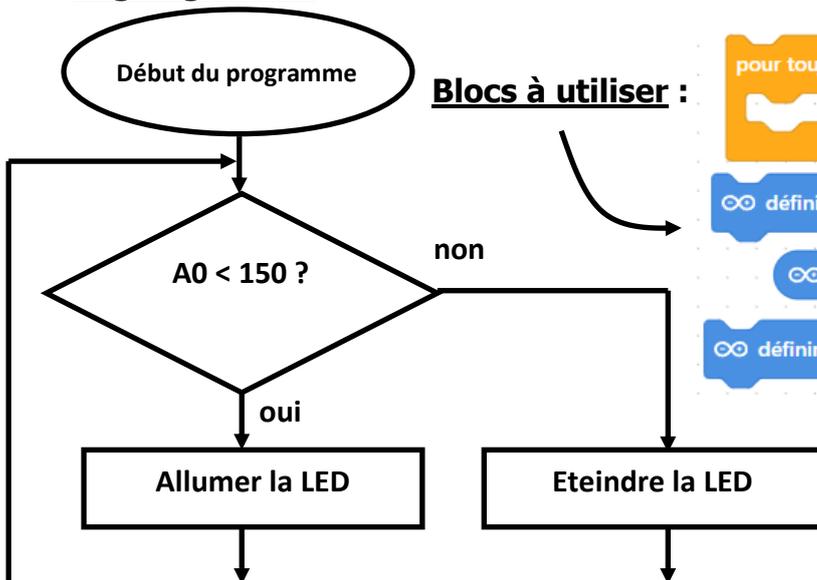
Montage du TP n°3 :



Le programme devra respecter les consignes :

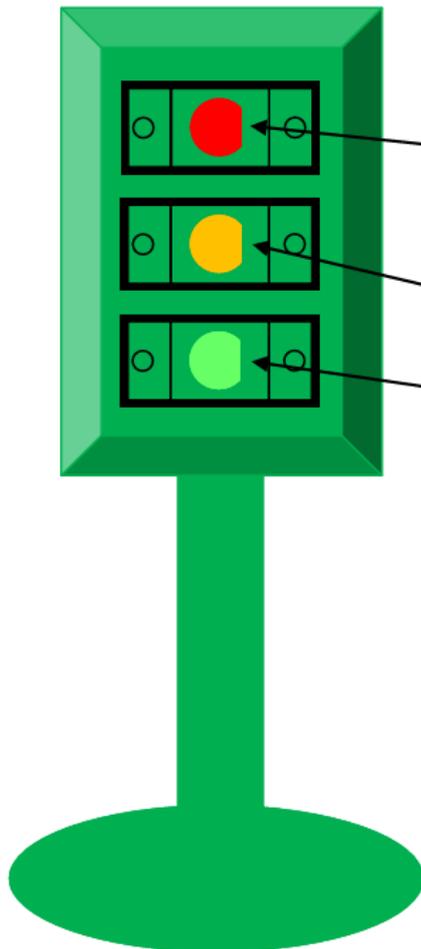
- Lorsque la photorésistance est dans le noir, la LED doit être allumée (valeur au-dessous de 150).
- Lorsque la photorésistance est dans la lumière, la LED doit s'éteindre (valeur au-dessus de 150).

Organigramme :



Blocs à utiliser :

1/ Présentation de la maquette feux de circulation : elle est composée de deux parties



Les feux pour les voitures :

Led rouge = « Feu rouge voiture » connectée sur la broche **D5** de la carte Arduino

Led orange = « Feu orange voiture » connectée sur la broche **D6** de la carte Arduino

Led verte = Feu vert voiture connectée sur la broche **D7** de la carte Arduino

Pour allumer une Led (ou pour l'éteindre), vous devez mettre l'état logique de la broche correspondante de la carte Arduino à « Haut » en utilisant le bloc suivant :

Numéro de la broche Haut = Led allumée

réglage de la sortie de la broche numérique 5 sur haut

Numéro de la broche Bas = Led éteinte

réglage de la sortie de la broche numérique 5 sur bas

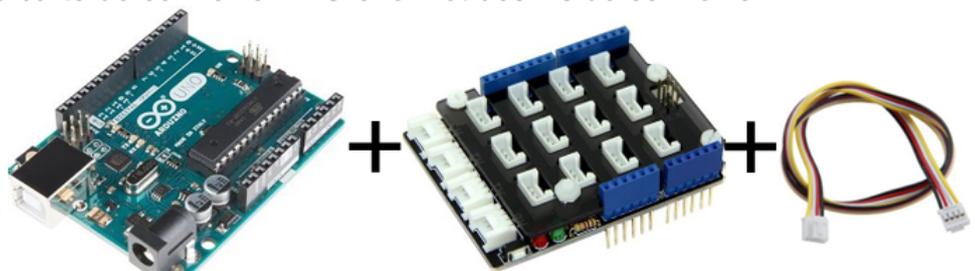
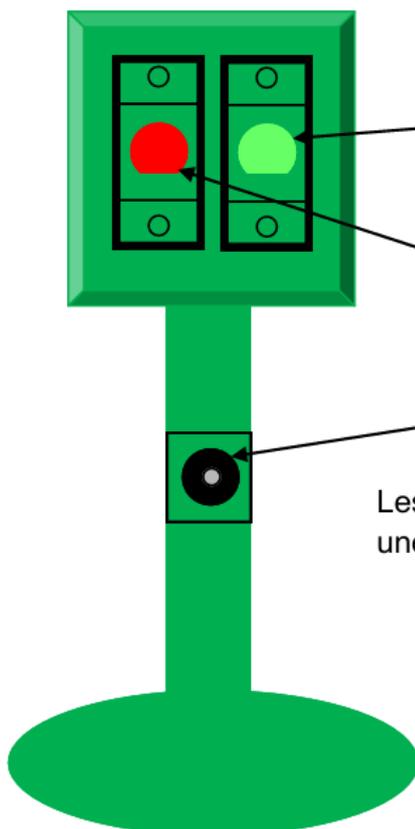
Les feux pour les piétons :

Led verte = « Feu vert piéton » connectée sur la broche **D4** de la carte Arduino

Led rouge = « Feu rouge piéton » connectée sur la broche **D3** de la carte Arduino

Bouton Poussoir (BP) = capteur permettant d'activer le passage piéton connecté sur la broche **D2** de la carte Arduino

Les 5 Leds et le bouton poussoir sont connectés à la carte Arduino via une carte de connexion « Grove » et des fils de connexion



2/ Programme n°1 : feux de circulation routier classique, on gère juste les 3 feux pour les voitures, on ne tient pas compte des feux piétons et du bouton poussoir.

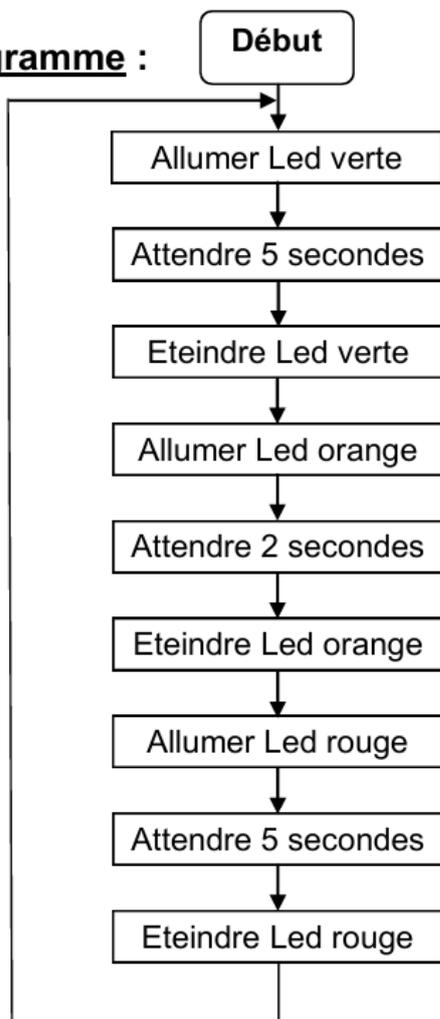
Algorithme :

répéter indéfiniment :

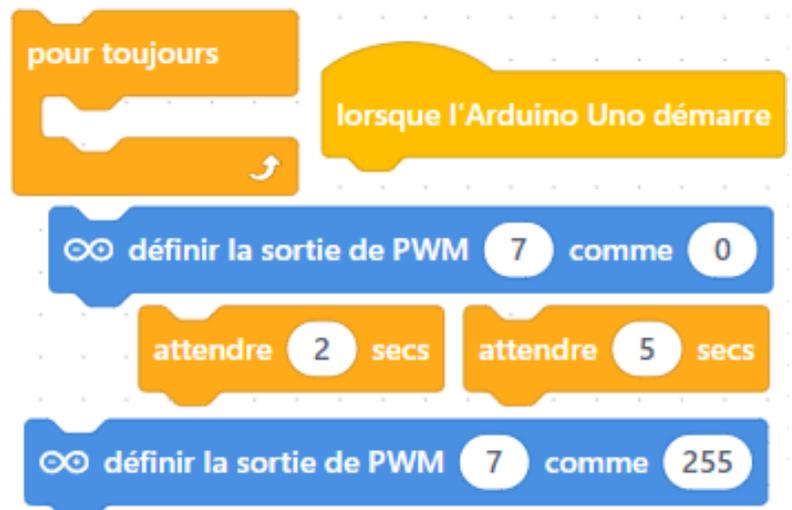
1. feu vert voiture 5 secondes ; broche D7 (haut pour allumer la led - bas pour l'éteindre)
2. Feu orange voiture 2 secondes ; broche D6 (haut pour allumer - bas pour éteindre)
3. Feu rouge voiture 5 secondes ; broche D5 (haut pour allumer - bas pour éteindre)



Organigramme :

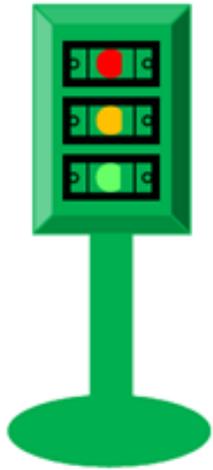


3/ Faire le programme en vous aidant de l'organigramme et des blocs suivants (qu'il faudra utiliser plusieurs fois...) :



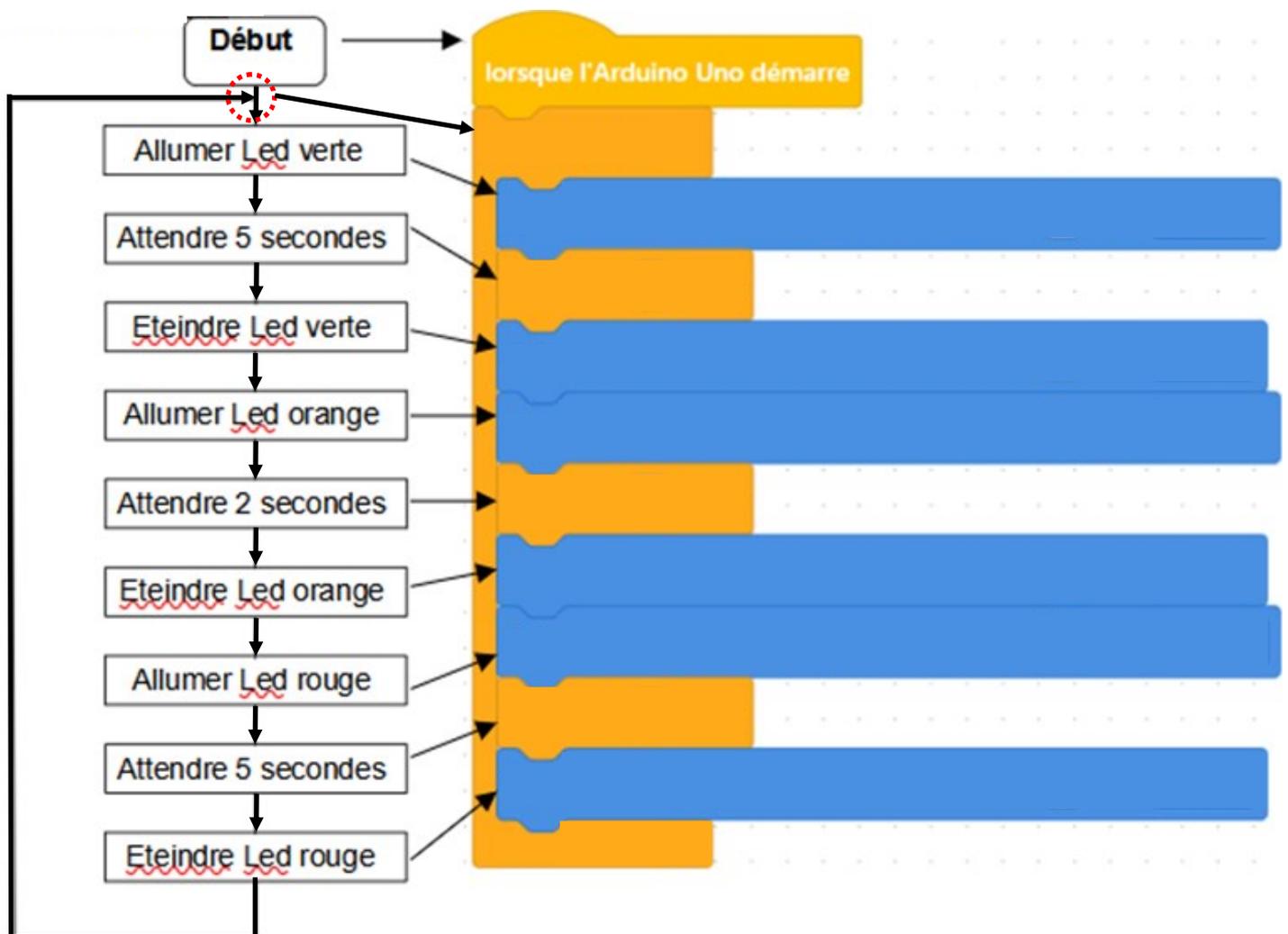
Programme n°1 : feux de circulation routier classique, on gère juste les 3 feux pour les voitures, on ne tient pas compte des feux piétons et du bouton poussoir.

Algorithme :



répéter indéfiniment :

1. feu vert voiture 5 secondes ; broche D7 (haut pour allumer la led - bas pour l'éteindre)
2. Feu orange voiture 2 secondes ; broche D6 (haut pour allumer - bas pour éteindre)
3. Feu rouge voiture 5 secondes ; broche D5 (haut pour allumer - bas pour éteindre)



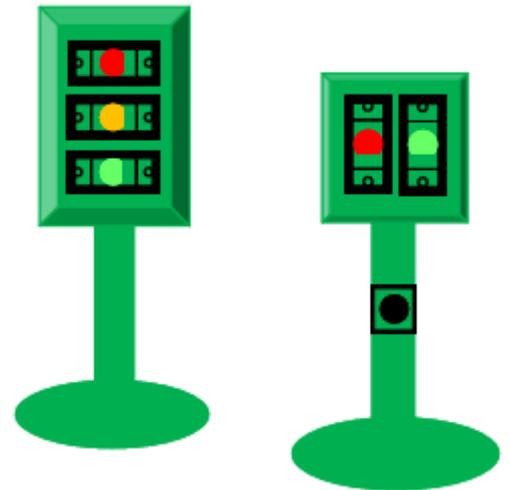
Séquence 5 : **activité** : **maquette 1** : feux de circulation routier + passage piéton

page 7

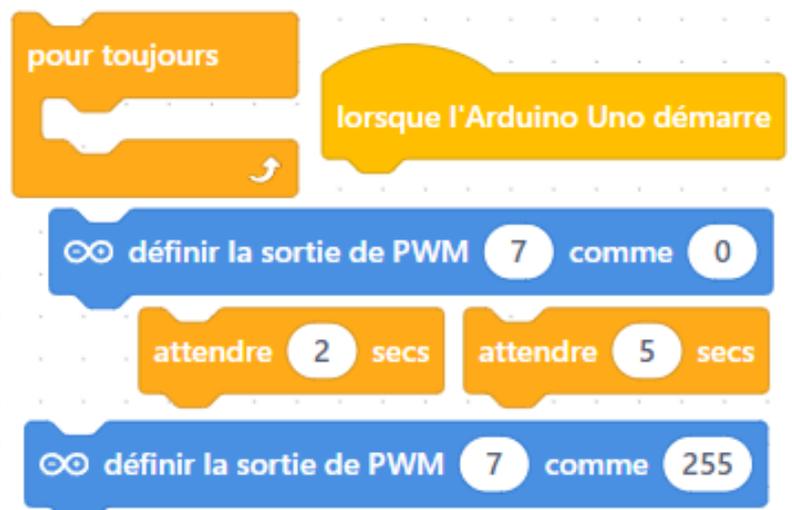
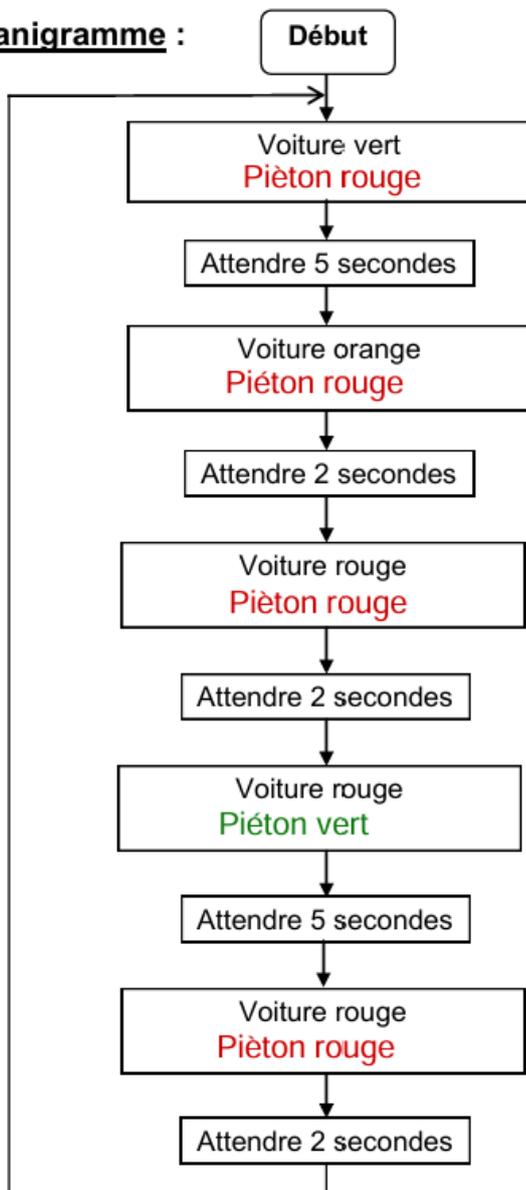
2/ **Programme n°2** : feux de circulation routier + passage piéton, on gère les 3 feux pour les voitures, les deux feux pour les piétons et on ne tient pas compte du bouton poussoir.

Algorithme : répéter indéfiniment :

1. feu vert voiture (D7) ; feu rouge piéton (D3) 5 secondes
2. Feu orange voiture (D6) ; feu rouge piéton (D3) 2 secondes
3. Feu rouge voiture (D5) ; feu rouge piéton (D3) 2 secondes
4. Feu rouge voiture (D5) ; feu vert piéton (D4) 5 secondes
5. Feu rouge voiture (D5) ; feu rouge piéton (D3) 2 secondes



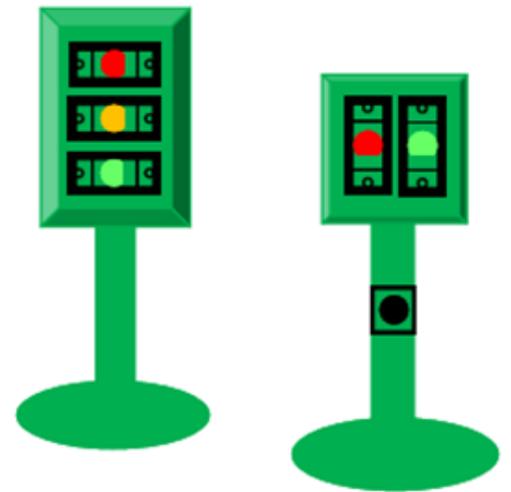
Organigramme :



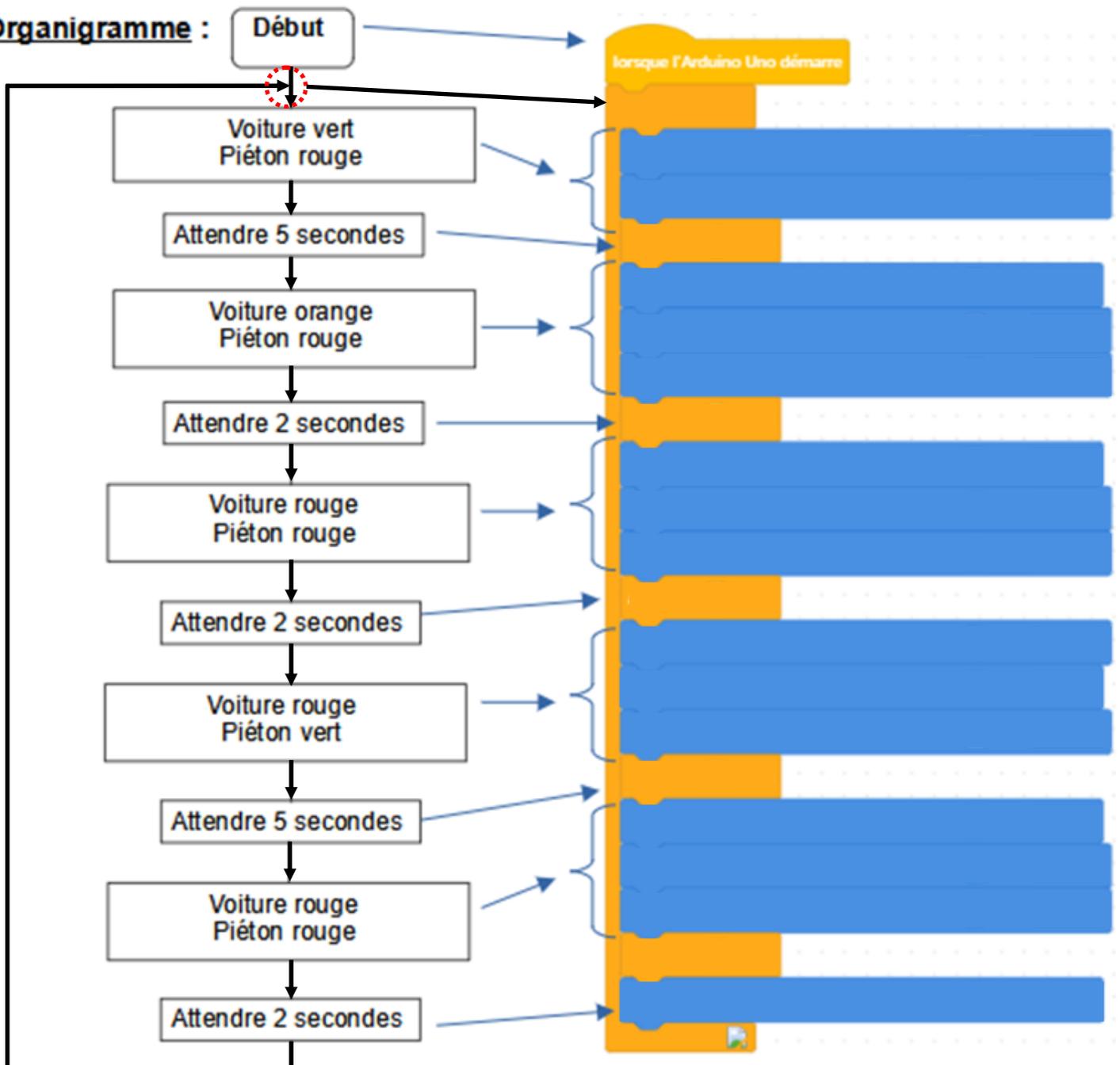
Programme n°2 : feux de circulation routier + passage piéton, on gère les 3 feux pour les voitures, les deux feux pour les piétons et on ne tient pas compte du bouton poussoir.

Algorithme : répéter indéfiniment :

1. feu vert voiture (D7) ; feu rouge piéton (D3) 5 secondes
2. Feu orange voiture (D6) ; feu rouge piéton (D3) 2 secondes
3. Feu rouge voiture (D5) ; feu rouge piéton (D3) 2 secondes
4. Feu rouge voiture (D5) ; feu vert piéton (D4) 5 secondes
5. Feu rouge voiture (D5) ; feu rouge piéton (D3) 2 secondes



Organigramme :



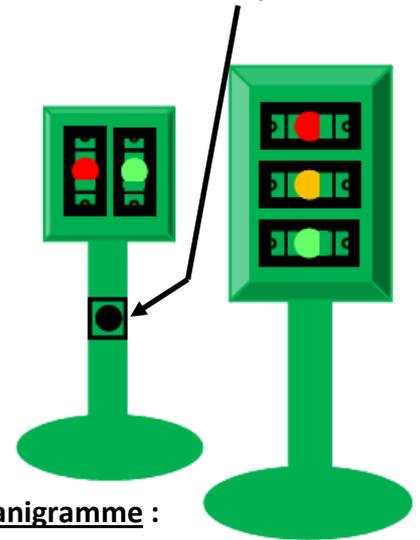
Séquence 5 : activité : **maquette 1** : feux de circulation routier + passage piéton avec déclenchement par bouton poussoir

2/ Programme n°3 : feux de circulation routier + passage piéton avec déclenchement par bouton poussoir, on gère les 3 feux pour les voitures, les deux feux pour les piétons et le bouton poussoir. Le **bouton poussoir** est connecté à la broche **D2** et renverra une valeur égale à 1 si on appuie dessus.

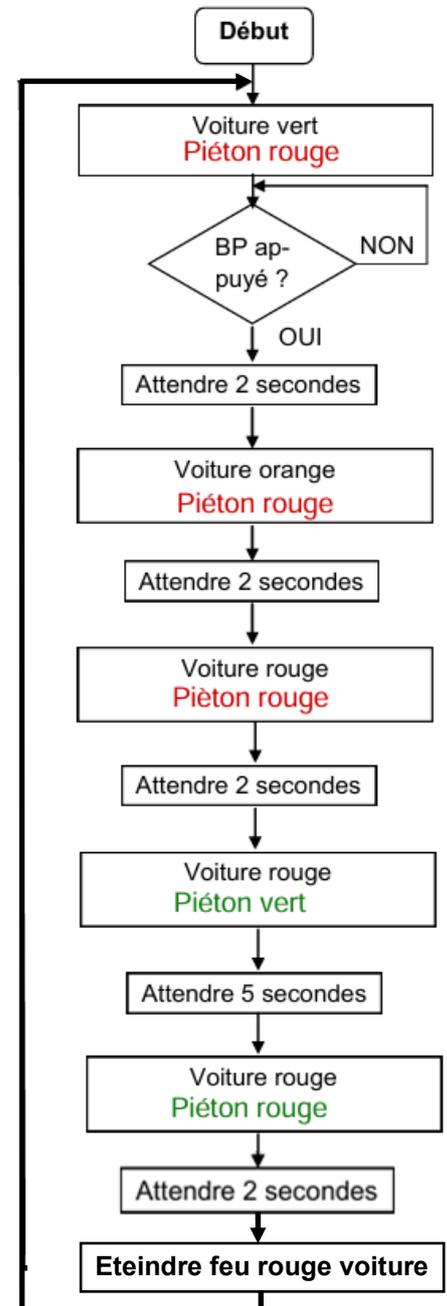
Algorithme :

répéter indéfiniment :

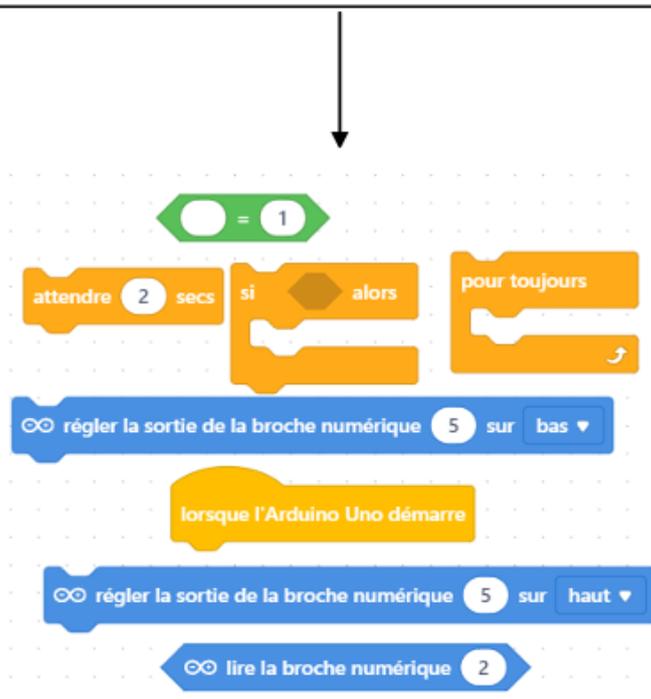
1. Feu vert voiture ; feu rouge piéton
2. **Si** Bouton Poussoir appuyé **alors**
3. Attendre 2 secondes
4. Feu orange voiture ; feu rouge piéton 2 secondes
5. Feu rouge voiture ; feu rouge piéton 2 secondes
6. Feu rouge voiture ; feu vert piéton 5 secondes
7. Feu rouge voiture ; feu rouge piéton 2 secondes
8. Eteindre feu rouge voiture



Organigramme :



3/ Faire le programme en suivant l'organigramme et avec les blocs suivants (qu'il faudra dupliquer) :



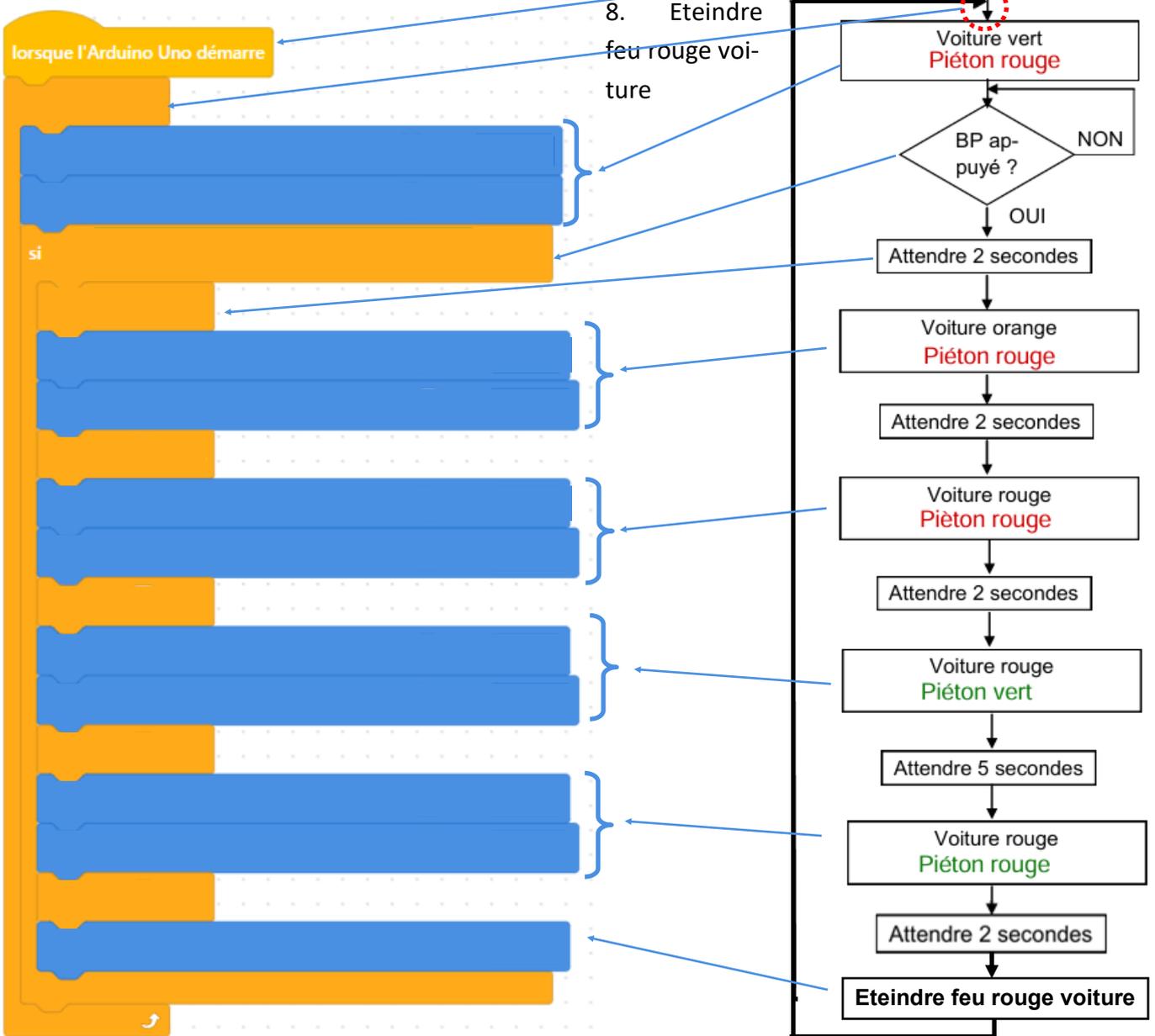
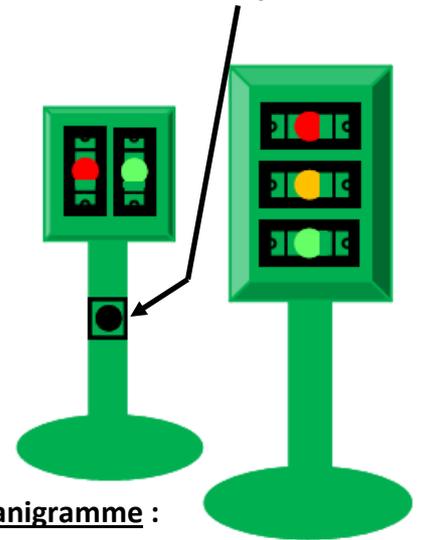
Séquence 5 : activité : **maquette 1** : feux de circulation routier + passage piéton avec déclenchement par bouton poussoir

2/ Programme n°3 : feux de circulation routier + passage piéton avec déclenchement par bouton poussoir, on gère les 3 feux pour les voitures, les deux feux pour les piétons et le bouton poussoir. Le **bouton poussoir** est connecté à la broche **D2** et renverra une valeur égale à 1 si on appuie dessus.

Algorithme :

répéter indéfiniment :

1. Feu vert voiture ; feu rouge piéton
2. **Si** Bouton Poussoir appuyé **alors**
3. Attendre 2 secondes
4. Feu orange voiture ; feu rouge piéton 2 secondes
5. Feu rouge voiture ; feu rouge piéton 2 secondes
6. Feu rouge voiture ; feu vert piéton 5 secondes
7. Feu rouge voiture ; feu rouge piéton 2 secondes
8. Eteindre feu rouge voiture



lorsque l'Arduino Uno démarre

si