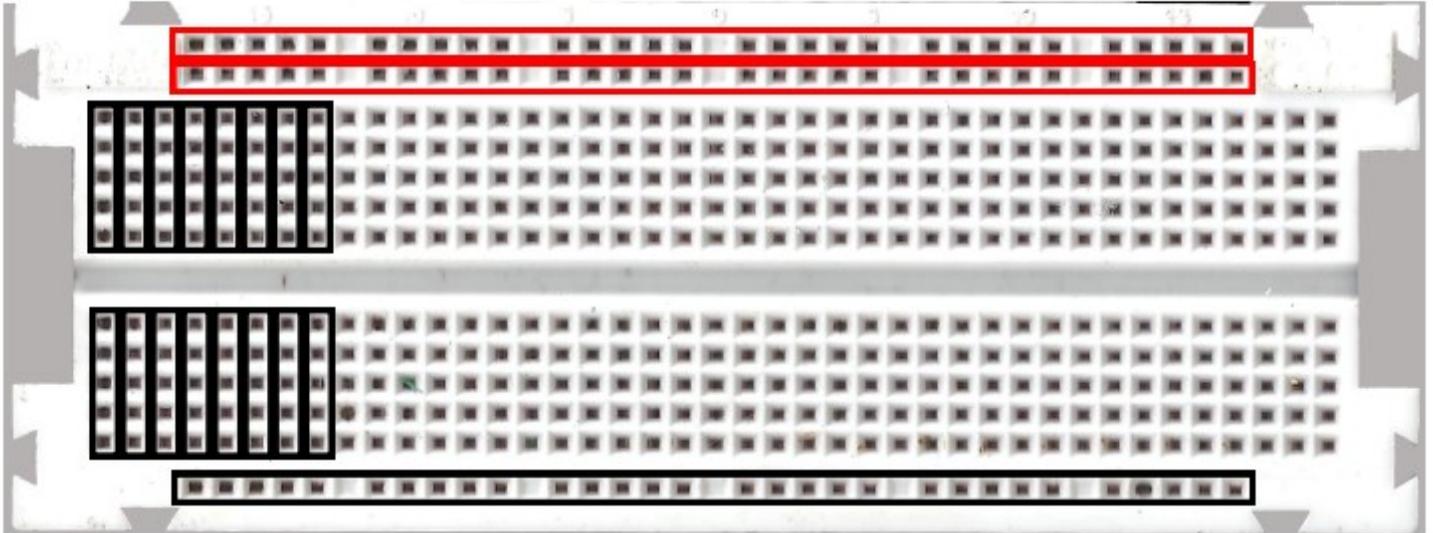


Séquence 5 - Correction

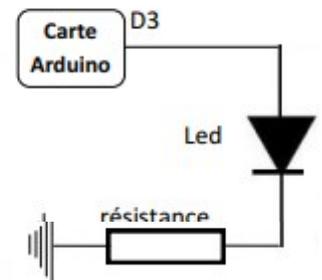
programmation d'une carte Arduino Uno
page 1

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.

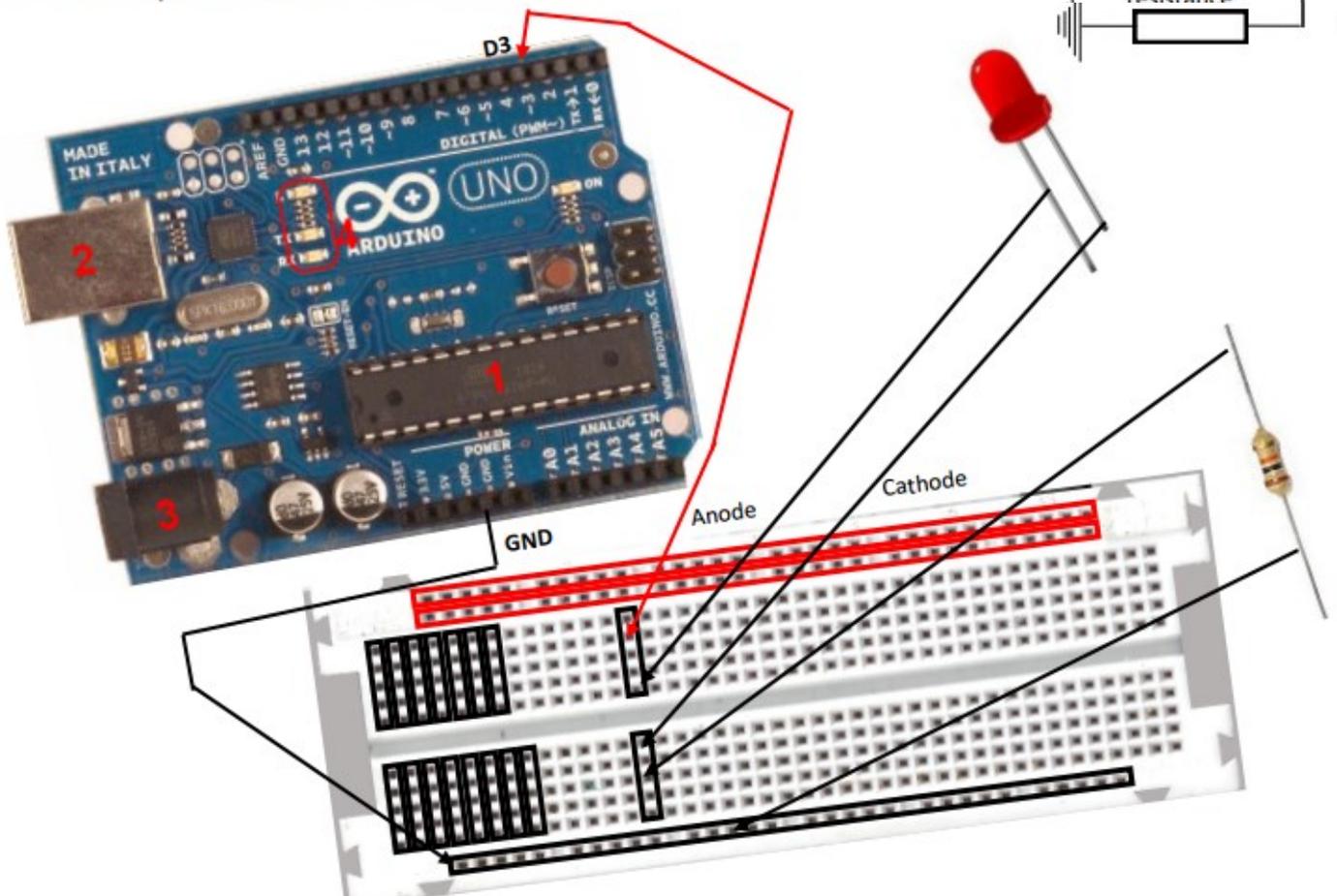


- Tous les trous de chaque cadre rouge horizontal sont en contact
- Tous les trous du cadre noir horizontal sont en contact.
- Tous les trous de chaque cadre noir vertical sont en contact

Schéma électronique



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



Séquence 5 - Correction

programmation d'une carte Arduino Uno
page 2

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

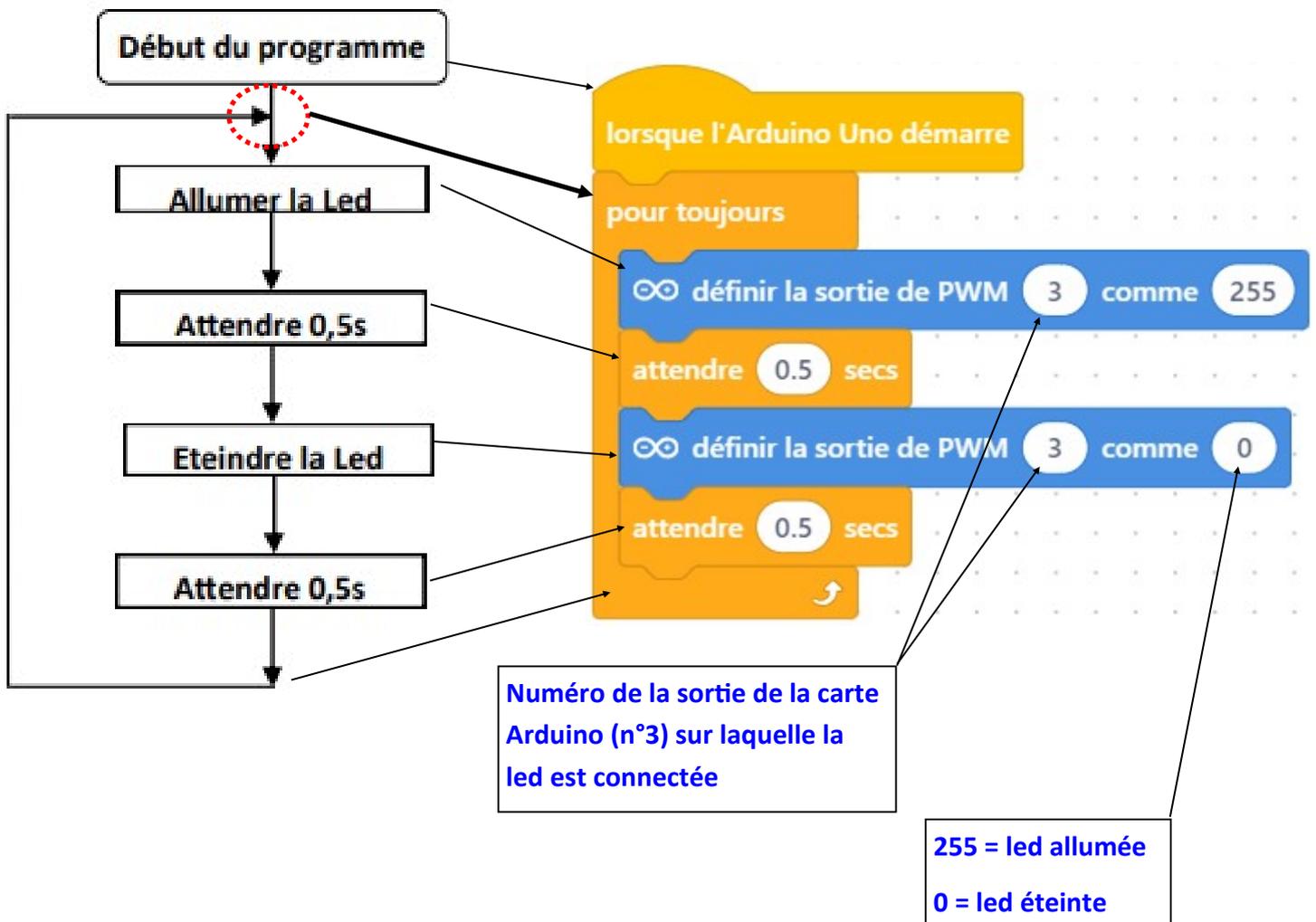
Ces blocs « **Arduino** » se trouvent dans la catégorie **Pilotage** du logiciel Mblock.

Organigramme :

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

lorsque l'Arduino Uno démarre

Correction du programme :



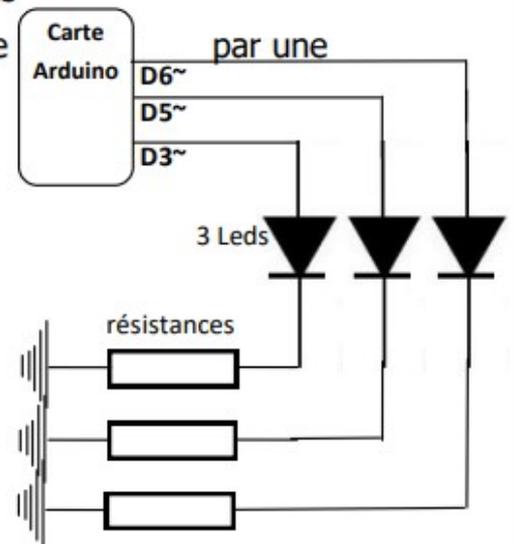
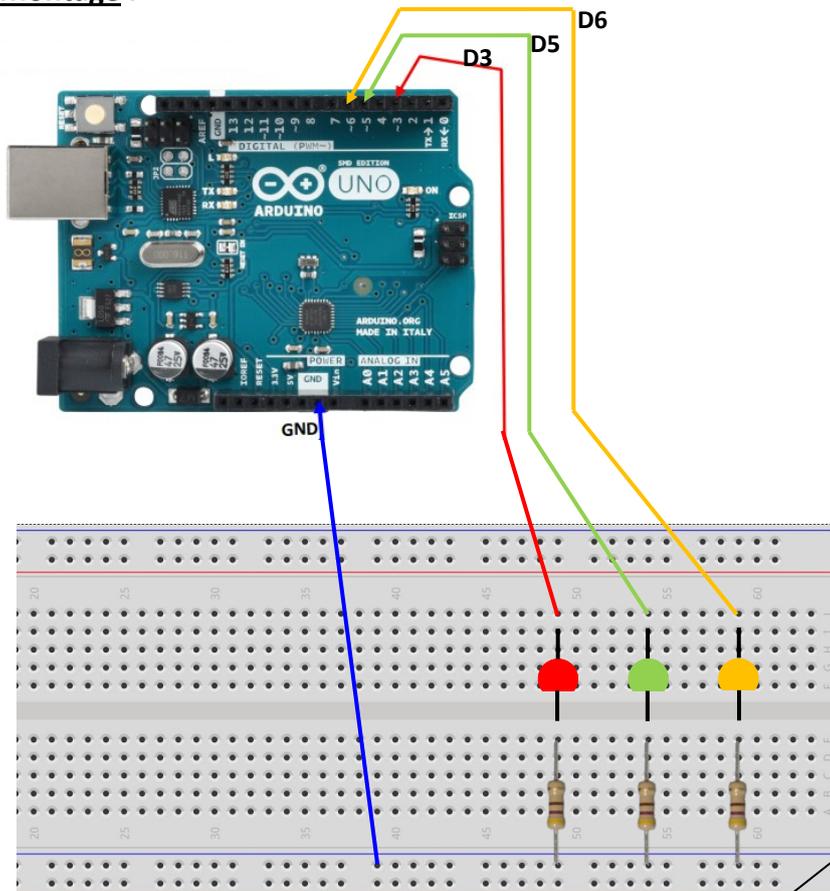
Séquence 5 - Correction

programmation d'une carte Arduino Uno
page 3

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
4. Faire le programme permettant d'allumer les 3 leds une

Montage :



Correction du programme :

Algorithmes :

Les 3 leds clignotent l'une après l'autre

Début du programme

Répétez indéfiniment

- Envoyer sur la broche PWM~3 la valeur 255
- Attendre 0.5 secondes
- Envoyer sur la broche PWM~3 la valeur 0
- Attendre 0.5 secondes
- Envoyer sur la broche PWM~5 la valeur 255
- Attendre 0.5 secondes
- Envoyer sur la broche PWM~5 la valeur 0
- Attendre 0.5 secondes
- Envoyer sur la broche PWM~6 la valeur 255
- Attendre 0.5 secondes
- Envoyer sur la broche PWM~6 la valeur 0
- Attendre 0.5 secondes



Séquence 5 - Correction

programmation d'une carte Arduino Uno
page 4

TP3 : créer un éclairage automatique

Objectif : allumer la led rouge lorsqu'il fait nuit et l'éteindre lorsqu'il fait jour. Nous utiliserons une **photorésistance** sensible à la luminosité.

- **Le jour**, la photorésistance enverra une valeur **supérieure à 100**.
- **La nuit** elle enverra une valeur **inférieure à 100**.
- La photorésistance est connectée sur l'**entrée analogique A0** de la carte Arduino
- La led est connectée sur la **broche 13** de la carte Arduino :
si la broche 13 reçoit la valeur 255 alors la led s'allume
si la broche 13 reçoit la valeur 0 alors la led s'éteint

Algorithme :

Début du programme

Répétez indéfiniment

Si valeur de la broche Analogique $0 < 100$

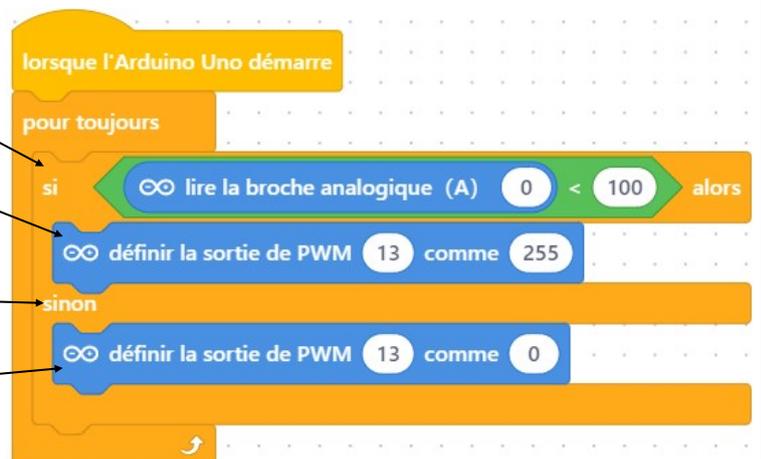
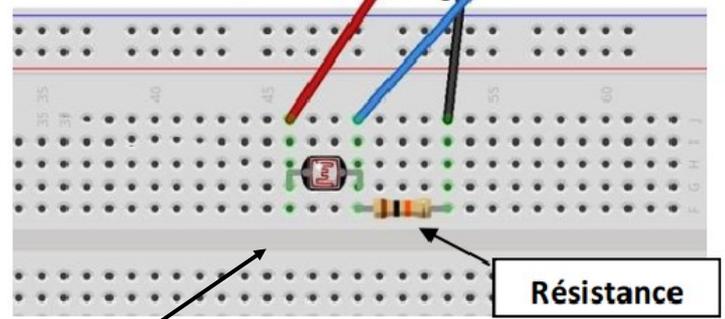
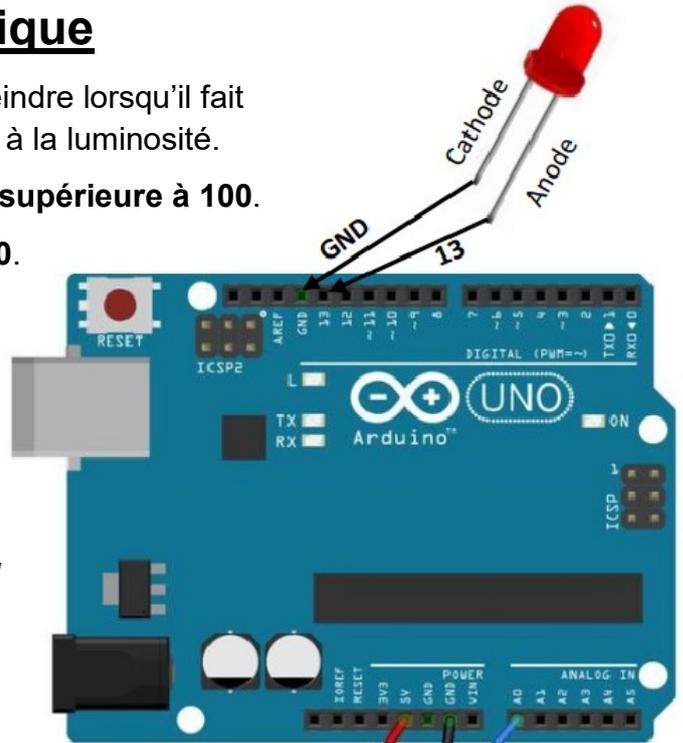
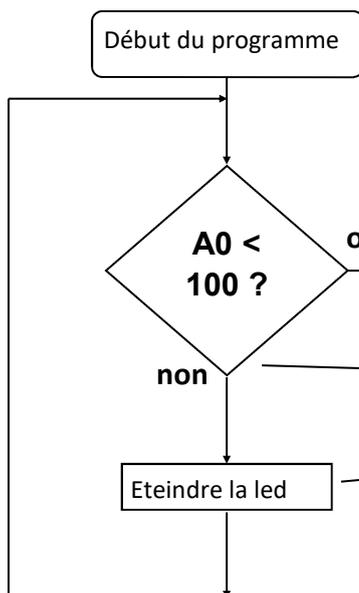
Alors

Envoyer sur la broche PWM~13 la valeur 255

Sinon

Envoyer sur la broche PWM~13 la valeur 0

Organigramme :

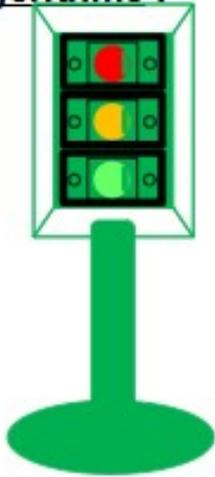


Séquence 5 cycle 4 / MSOST - IP2 - CORRIGE

Découverte de l'environnement Arduino
Programmation d'un système de feux de croisement routier

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 :

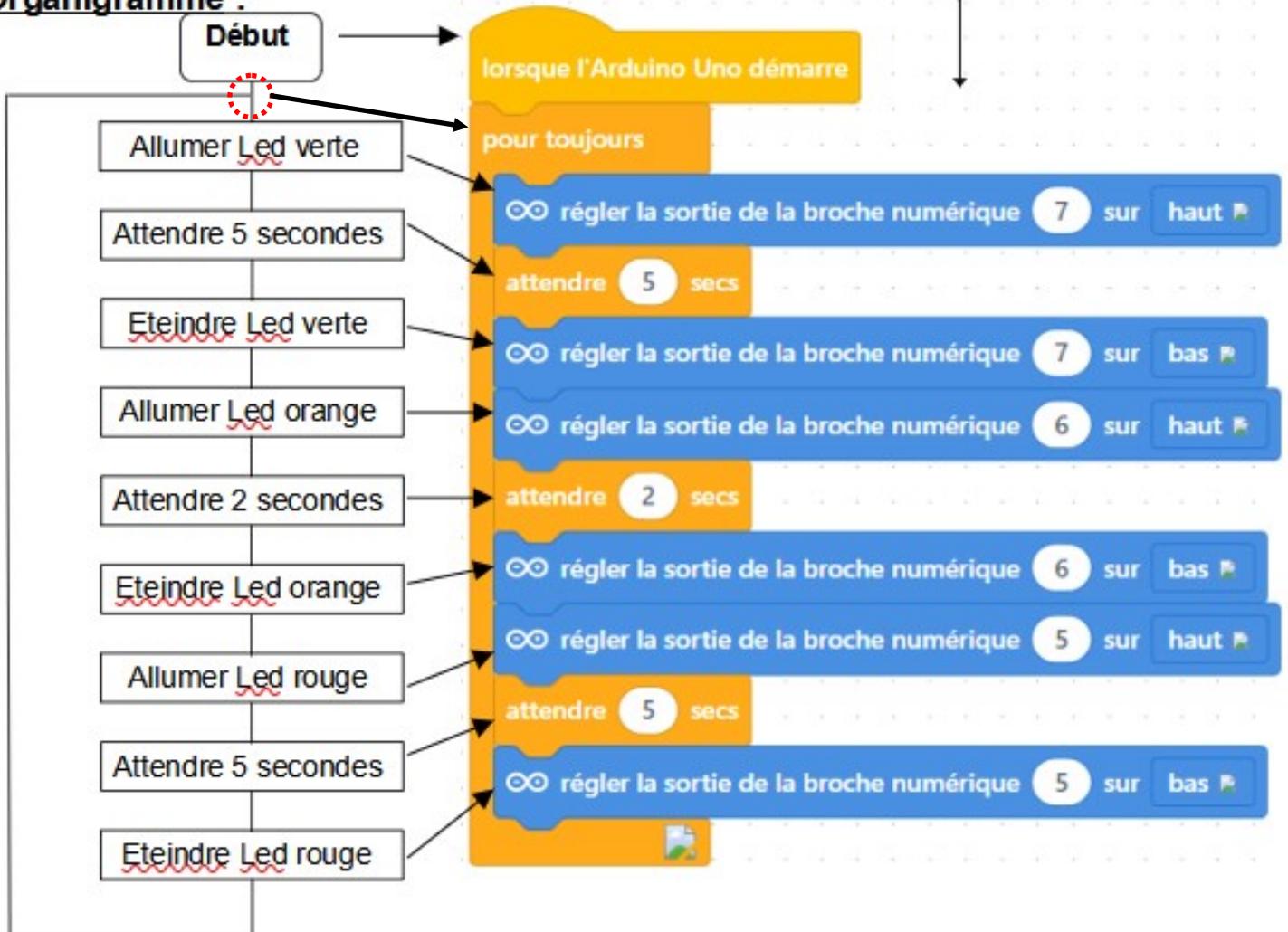
feu vert voiture 5 secondes ; broche **D7** (haut pour allumer la led - bas pour l'éteindre)

Feu orange voiture 2 secondes ; broche **D6** (haut pour allumer - bas pour éteindre)

Feu rouge voiture 5 secondes ; broche **D5** (haut pour allumer - bas pour éteindre)

PROGRAMME

Organigramme :



Séquence 5 cycle 4 / MSOST - IP2 - CORRIGE

Découverte de l'environnement Arduino

Programmation d'un système de feux de croisement routier

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 :

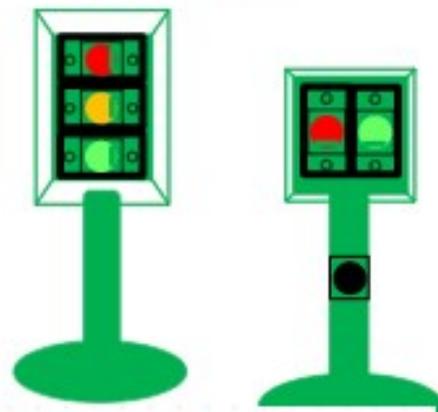
feu vert voiture ; feu rouge piéton 5 secondes

Feu orange voiture ; feu rouge piéton 2 secondes

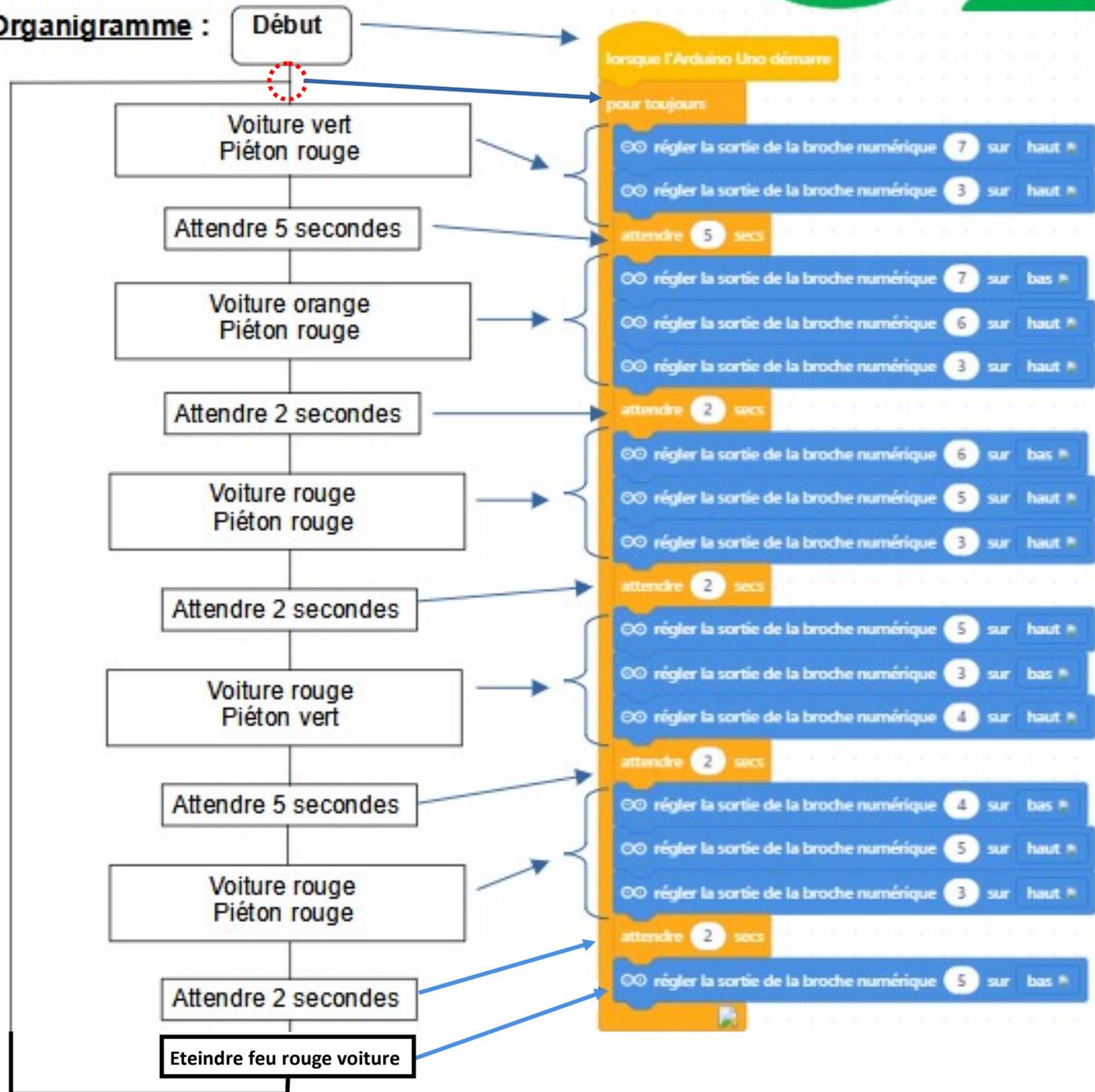
Feu rouge voiture ; feu rouge piéton 2 secondes

Feu rouge voiture ; feu vert piéton 5 secondes

Feu rouge voiture ; feu rouge piéton 2 secondes



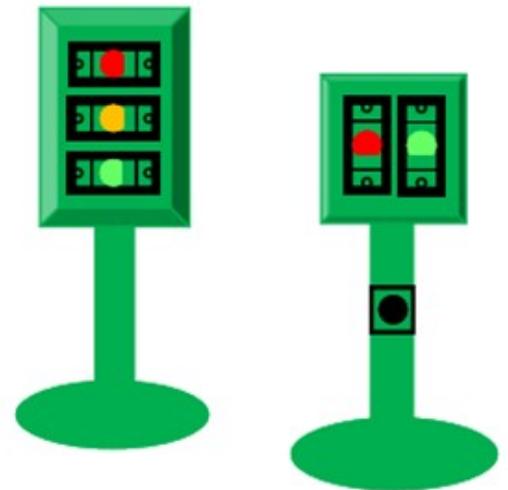
Organigramme :



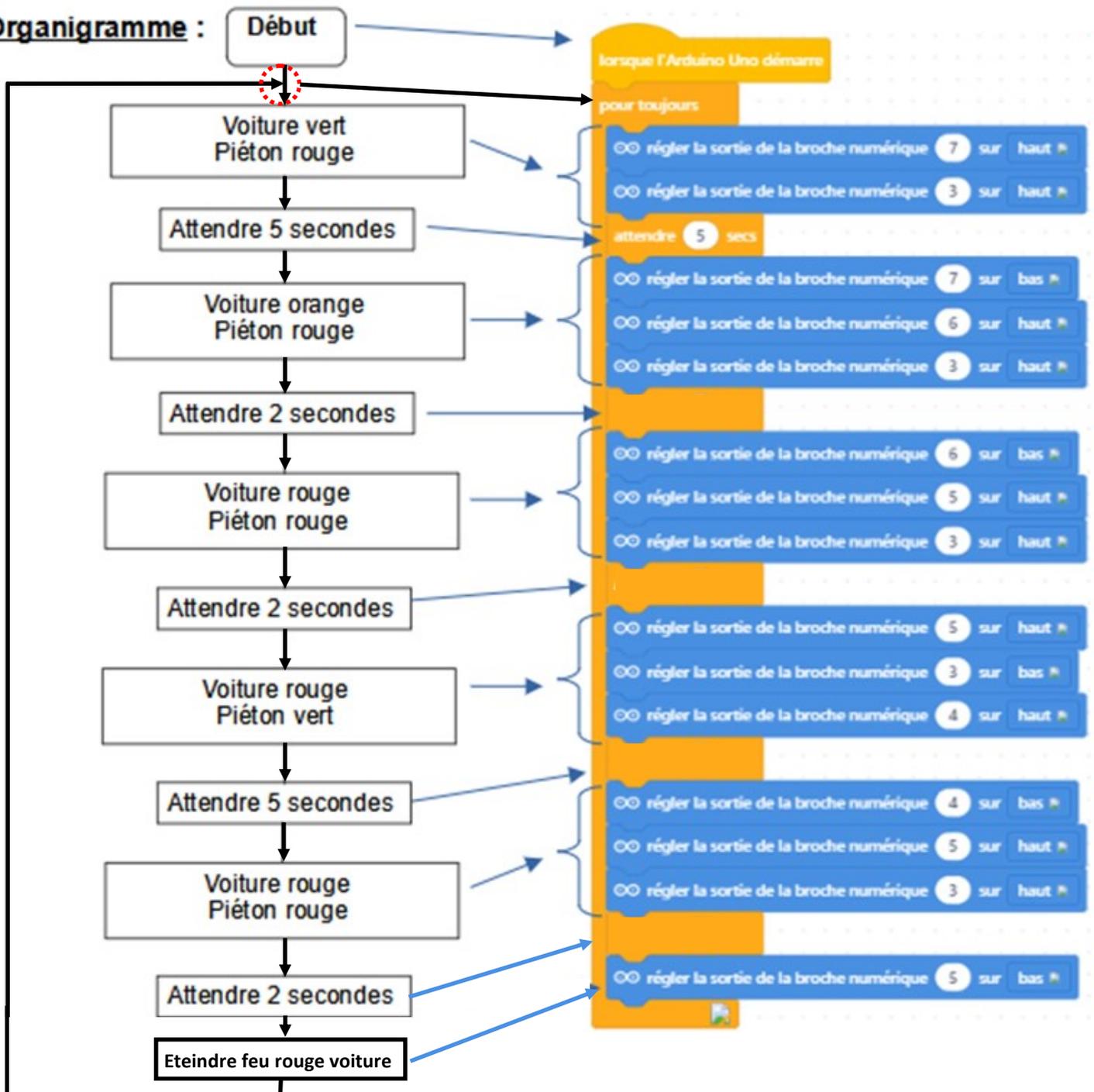
Programme n°2 : feux de circulation routier + passage piéton, on gère les 3 feux pour les voitures, les deux feux pour les piétons et on ne tient pas compte du bouton poussoir.

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 :



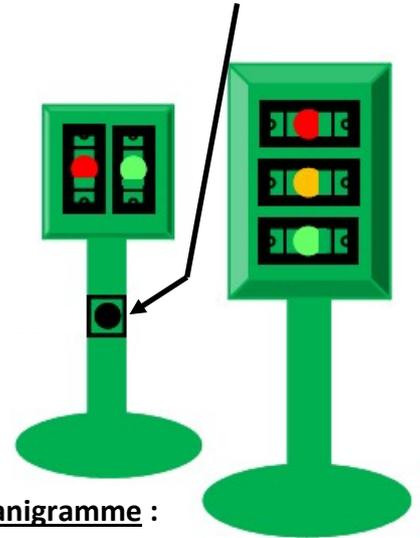
Séquence 5 : activité : **maquette 1** : feux de circulation routier + passage piéton avec déclenchement par bouton poussoir

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 :

