

Nom : .....

## OBJECTIF : SÉQUENCEMENT ET CONTROLE DU TEMPS

Ce système va vous apprendre à gérer des séquences d'instruction, le contrôle du temps et de bouton à l'aide d'un microcontrôleur. Ultimement, il simule les deux feux rouges situés à une intersection.

### MATERIEL

- Arduino board \*1
- USB cable \*1
- Red M5 LED\*2
- Yellow M5 LED\*2
- Green M5 LED\*2
- 220Ω resistor \*6
- Breadboard\*1

### UN SEUL FEU ROUGE

D'abord, prenez en main le cablage des LED et leur programmation avec le microcontrôleur en faisant fonctionner le montage de la figure 1.

- 1 Quel courant faut-il dans chaque LED ?

.....

.....

.....

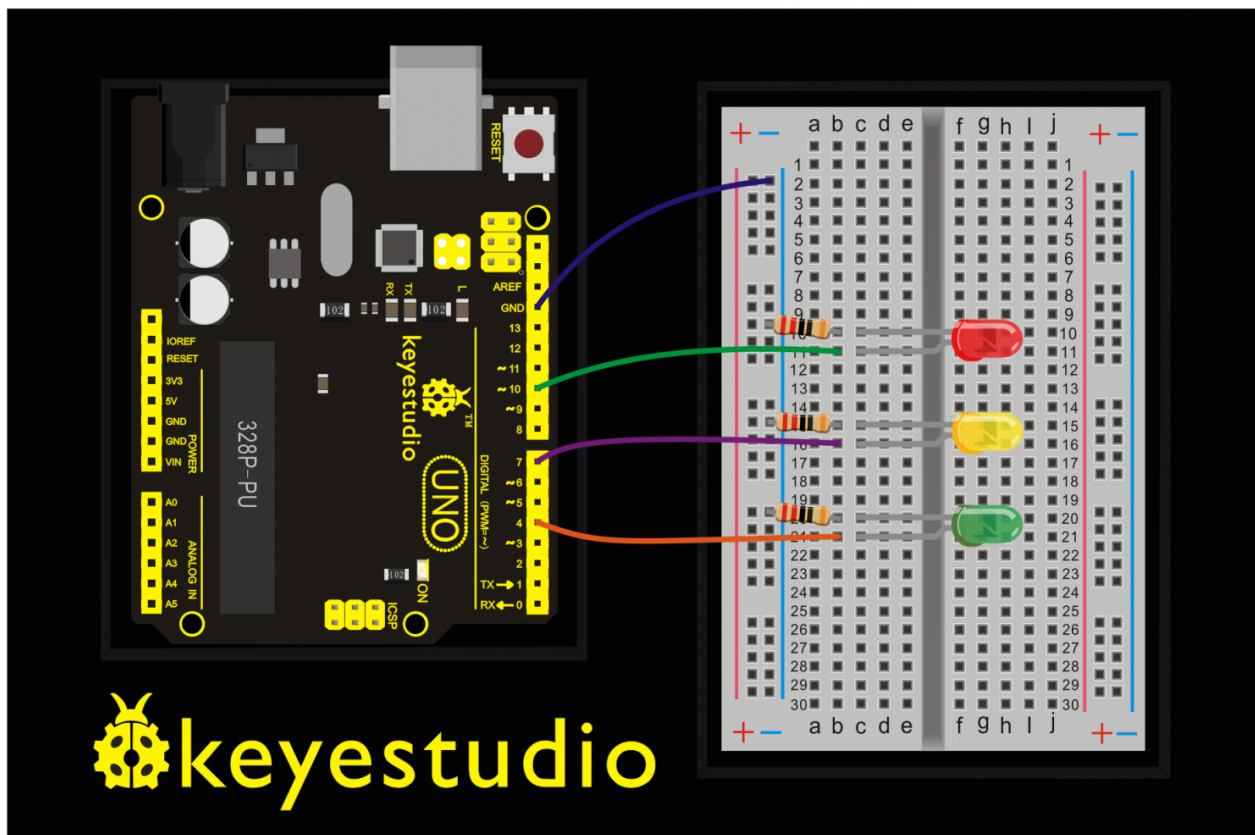


Figure 1: Cablage d'un seul feu rouge

2 Quel tension chute aux bornes des trois LED ?

.....

.....

.....

3 En déduire la valeur de chaque résistance en série avec les LED ?

.....

.....

.....

4 Quelles résistances  $R_R, R_V, R_J$  du matériel prenons-nous effectivement ?

.....

.....

.....

5 Implémenter le code suivant et résumer d'une phrase le comportement observé.

```
1 int redled =10;           // initialize digital pin 10
2 int yellowled =7;         // initialize digital pin 7
3 int greenled =4;          // initialize digital pin 4
4 void setup() {
5     pinMode(redled , OUTPUT); // pin with red LED as output
6     pinMode(yellowled , OUTPUT); // pin with yellow LED as output
7     pinMode(greenled , OUTPUT); // pin with green LED as output
8 }
9 void loop()
10 {
11     digitalWrite(greenled , HIGH); // turn on green LED
12     delay(5000); // wait 5 seconds
13     digitalWrite(greenled , LOW); // turn off green LED
14     for(int i=0;i<3;i++) { // blinks for 3 times
15         delay(500); // wait 0.5 second
16         digitalWrite(yellowled , HIGH); // turn on yellow LED
17         delay(500); // wait 0.5 second
18         digitalWrite(yellowled , LOW); // turn off yellow LED
19     }
20     delay(500); // wait 0.5 second
21     digitalWrite(redled , HIGH); // turn on red LED
22     delay(5000); // wait 5 second
23     digitalWrite(redled , LOW); // turn off red LED
24 }
```

**Constatation professeur :**

6 Décrire l'algorithme quasiment ligne à ligne.

.....

.....

.....

.....

.....

## UN SEUL FEU ROUGE ET UN BOUTON

7 Programmer un simple feu rouge avec un bouton.

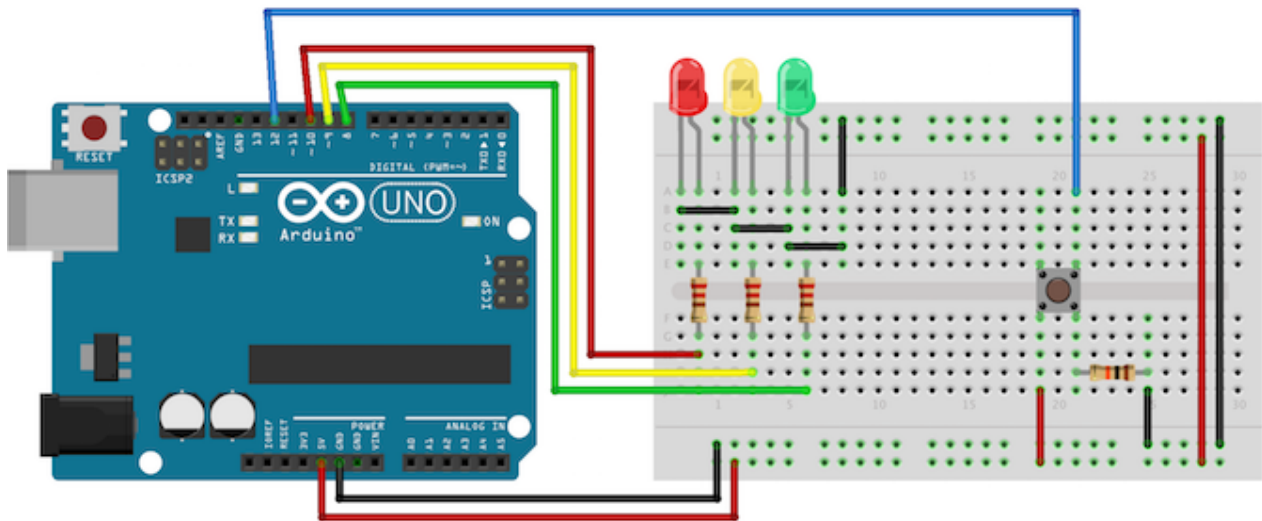


Figure 2: Feu de signalisation avec un bouton

```

1 int red = 10;
2 int yellow = 9;
3 int green = 8;
4 int button = 12;
5
6 void setup(){
7     pinMode(red, OUTPUT);
8     pinMode(yellow, OUTPUT);
9     pinMode(green, OUTPUT);
10    pinMode(button, INPUT);
11    digitalWrite(green, HIGH);
12 }
13
14 void loop() {
15     if (digitalRead(button) == HIGH){
16         delay(15); // software debounce
17         if (digitalRead(button) == HIGH) {

```

```
18         changeLights();
19         delay(15000); // wait for 15 seconds
20     }
21 }
22 }
23
24 void changeLights(){
25     digitalWrite(green, LOW);
26     digitalWrite(yellow, HIGH);
27     delay(3000);
28
29     digitalWrite(yellow, LOW);
30     digitalWrite(red, HIGH);
31     delay(5000);
32
33     digitalWrite(red, LOW);
34     digitalWrite(green, HIGH);
35     delay(3000);
36 }
```

**Constatation professeur :**

8 A la fin de la fonction `setup()`, dans quels états sont les DEL verte, rouge et orange ?

9 Combien de temps doit rester appuyé le bouton pour permettre l'instruction à la ligne 18 ?

10 Que fait l'instruction de la ligne 18 ?

11 Combien de temps restent allumées les lumières orange et rouge avant de changer ?

12 A la ligne 16, il y a écrit `software debounce`. Pourquoi `software` ? Pourquoi `debounce` ?

13 Quelle valeur retourne la fonction `changeLights()` ?

## DEUX FEUX ROUGES

14 Pour l'exercice, implémentons des feux de croisement avec le séquençage qu'ils ont aux états-unis. Ce séquençage est précisé dans la figure 4. Décrire ce qui change par rapport aux feux rouges français.

15 Implémenter ce code et le câblage de la figure 3.

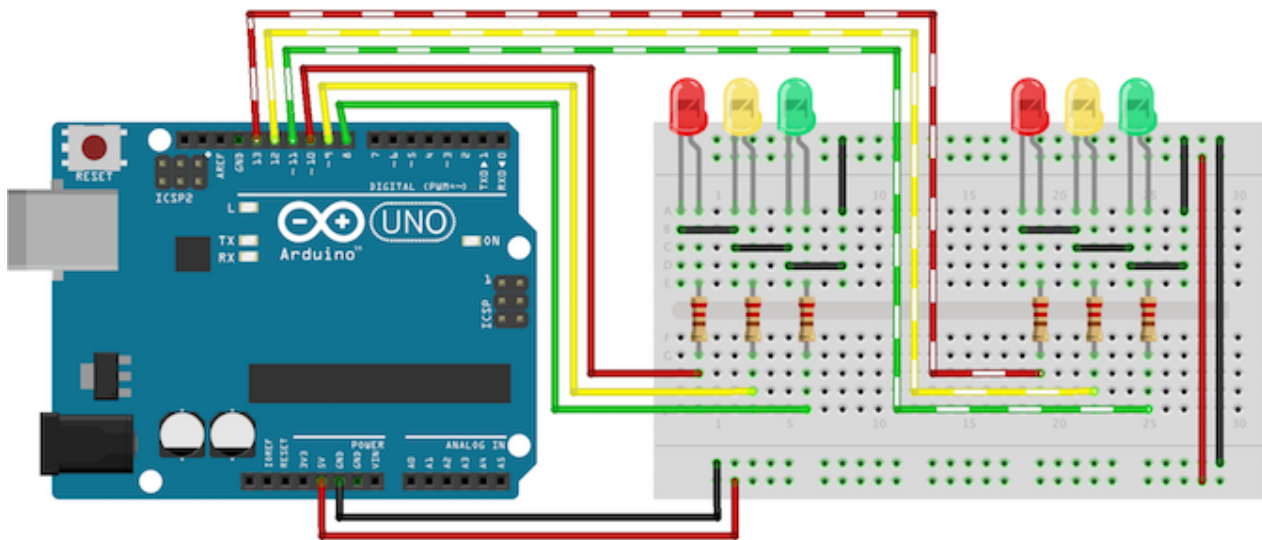


Figure 3: Cablage de deux feux rouges à une intersection

```

1  int red1 = 10;
2  int yellow1 = 9;
3  int green1 = 8;
4  int red2 = 13;
5  int yellow2 = 12;
6  int green2 = 11;
7
8  void setup(){
9      pinMode(red1, OUTPUT);
10     pinMode(yellow1, OUTPUT);
11     pinMode(green1, OUTPUT);
12
13     pinMode(red2, OUTPUT);
14     pinMode(yellow2, OUTPUT);
15     pinMode(green2, OUTPUT);
16 }
17
18 void loop(){
19     changeLights();
20     delay(15000);
21 }
22
23 void changeLights(){
24     // turn both yellows on
25     digitalWrite(green1, LOW);
26     digitalWrite(yellow1, HIGH);
27     digitalWrite(yellow2, HIGH);
28     delay(5000);
29
30     // turn both yellows off, and opposite green and red
31     digitalWrite(yellow1, LOW);
32     digitalWrite(red1, HIGH);

```

| Timeline (seconds) | Busy Bunny Lane Cycle | Lazy Tortoise Ave Cycle | Comments   |
|--------------------|-----------------------|-------------------------|--|
| 0                  | GREEN                 | RED                     | Busy Bunny Lane has its traffic flowing. Lazy Tortoise Ave has its traffic waiting for its turn.     |
| 1                  | GREEN                 | RED                     |  |
| 2                  | GREEN                 | RED                     |  |
| 3                  | GREEN                 | RED                     |  |
| 4                  | GREEN                 | RED                     |  |
| 5                  | GREEN                 | RED                     |  |
| 6                  | GREEN                 | RED                     |  |
| 7                  | GREEN                 | RED                     |  |
| 8                  | GREEN                 | RED                     |  |
| 9                  | GREEN                 | RED                     |  |
| 10                 | GREEN                 | RED                     |  |
| 11                 | GREEN                 | RED                     |  |
| 12                 | AMBER                 | RED                     | Busy Bunny Lane is about to stop its traffic. Lazy Tortoise Ave still waits for its turn.            |
| 13                 | AMBER                 | RED                     |  |
| 14                 | AMBER                 | RED                     |  |
| 15                 | RED                   | RED                     | Dead period for both streets (prevents accidents).   |
| 16                 | RED                   | RED                     |  |
| 17                 | RED                   | GREEN                   | Lazy Tortoise Ave has its traffic flowing. Busy Bunny Lane Ave has its traffic waiting for its turn. |
| 18                 | RED                   | GREEN                   |  |
| 19                 | RED                   | GREEN                   |  |
| 20                 | RED                   | GREEN                   |  |
| 21                 | RED                   | AMBER                   | Lazy Tortoise Ave is about to stop its traffic. Busy Bunny Lane still waits for its turn.            |
| 22                 | RED                   | AMBER                   |  |
| 23                 | RED                   | AMBER                   |  |
| 24                 | RED                   | RED                     | Dead period for both streets (prevents accidents).   |
| 25                 | RED                   | RED                     |  |

Figure 4: séquencement des feux rouges aux états-unis.

```

33 digitalWrite(yellow2, LOW);
34 digitalWrite(red2, LOW);
35 digitalWrite(green2, HIGH);
36 delay(5000);
37
38 // both yellows on again
39 digitalWrite(yellow1, HIGH);
40 digitalWrite(yellow2, HIGH);
41 digitalWrite(green2, LOW);
42 delay(3000);
43
44 // turn both yellows off, and opposite green and red
45 digitalWrite(green1, HIGH);
46 digitalWrite(yellow1, LOW);
47 digitalWrite(red1, LOW);
48 digitalWrite(yellow2, LOW);
49 digitalWrite(red2, HIGH);
50 delay(5000);
51 }

```

**Constatation professeur :**