# ousseynou seck

# SIMULATION FEU DE SIGNALISATION

# CCNB

# bathurst

# 2021

# Simulation de quatre feux rouges

Table des matières

[**SIMULATION FEU DE SIGNALISATION** 1](#_Toc89377593)

[**Simulation de quatre feux rouges** 2](#_Toc89377597)

[**Introduction** 3](#_Toc89377598)

[**Définition et Rôles** 4](#_Toc89377599)

[**Fonctionnement** 5](#_Toc89377600)

[Simulation du feu 6](#_Toc89377601)

[Étape : Câblage 6](#_Toc89377602)

[**Le Schéma** 7](#_Toc89377603)

[**Le Code** 8](#_Toc89377604)

[**Amélioration Du Code** 9](#_Toc89377605)

[**Le Code Amélioré** 11](#_Toc89377606)

[**Conclusion** 13](#_Toc89377607)

# **Introduction**

De nos jours, de nombreux travaux de recherche à travers le monde portent sur le transport routier. La motivation vient du fait que ce secteur nécessite de nouvelles applications de méthodes permettant de contrecarrer l'augmentation accrue des problèmes issus du trafic routier, que ce soit, en vue de réduire le nombre d'accidents, de diminuer le trafic dans les grandes agglomérations ou de réduire la pollution. En effet, l'un des facteurs favorables à cette problématique est dû à l'accroissement du nombre exponentiel de véhicules sur la route. Cependant pour réguler ou fluidifier le trafic routier plusieurs solutions sont mises en place dont on peut citer les feux de signalisation entre autres. On va essayer de définir leur rôle et leur fonctionnement et enfin simuler nos propres feux de signalisation.

# **Définition et Rôles**

feu tricolore, est un dispositif permettant la régulation du trafic routier entre les usagers de la route, les véhicules et les piétons. Ces feux, le plus souvent disposés dans les carrefours, sont commandés par un contrôleur de feux. Le contrôleur de feux désigne donc cet appareil électronique de contrôle/ commande servant à la gestion des feux de circulation en régulation de trafic ; c’est-à-dire qu’il permet d’assurer la commande des feux en respectant les contraintes de sécurité définies entre les lignes de feux antagonistes. Lorsque le volume de la circulation excède la capacité d’un carrefour à quatre panneaux d’arrêt, il faut songer à installer des feux. Les critères qui gouvernent l’installation de feux sont basés sur :  Le volume des voitures et des piétons traversant un carrefour ;  Les retards subis par les véhicules et les piétons en provenance des rues transversales ;  Le nombre de collisions à ce carrefour. Un système de feux bien ajusté peut augmenter considérablement le nombre de véhicules traversant un carrefour ou améliorer la sécurité des piétons et des voitures. Les feux sont généralement déclinés à partir de deux couleurs de base : le rouge pour fermer, le vert (ou encore le bleu plus rarement) pour ouvrir. Le jaune-orangé est également utilisé et sert à signaler le passage du feu vert au feu rouge (et aussi, dans certains pays, du feu rouge au feu vert)

# **Fonctionnement**

Aux intersections, un feu tricolore de couleur Rouge indique que toutes les usagères et tous les usagers de cet axe doivent s’arrêter à proximité immédiate du feu, en général sur une ligne, tracée au sol du feu de signalisation pour les automobilistes et les cyclistes ou matérialisée par un trottoir pour les piétons. Ceci laisse le temps à différents usagers de passer en évitant de bloquer certaines voies de circulation. Il existe trois principales séquences de feux.  Feux de signalisation à trois états  Rouge : « Tout conducteur doit marquer l'arrêt absolu devant un feu de signalisation rouge, fixe ou clignotant. »  Vert : « Les feux de signalisation verts autorisent le passage des véhicules » ;  Orange ou Jaune : « Tout conducteur doit marquer l'arrêt devant un feu de signalisation Orange ou Jaune fixe, sauf dans le cas où, lors de l'allumage dudit feu, le conducteur ne peut plus arrêter son véhicule dans des conditions de sécurité suffisantes. ».  Feux de signalisation à quatre états : Le deuxième état, Orange ou Jaune bref sans extinction du Rouge, permet aux conducteurs de se préparer à démarrer mais n'autorise pas le passage.  Feux de signalisation à cinq états : L'état supplémentaire, Vert clignotant entre les états du Vert et Orange ou Jaune, prévient la fin de l'état Vert. 1.3. Gestion du trafic, sécurité et environnement En voirie urbaine, les feux tricolores permettent de réguler et sécuriser les flux denses de véhicules rapides, à condition d'être utilisés et réglés avec pertinence.

# Simulation du feu

Pour la simulation de notre projet, nous avons utilisé le logiciel de Tinkercad

Étape 1: Pièces et outils

Liste des pieces

Tout d’abord vous devez collecter les pièces.

4 leds vert

4 jaune Led

4 rouge Led

6 resistance 220 ohms

Une plaquette de prototypage

Un microcontrôleur

Des fils multicolores

## Étape 2: Câblage (1 de 2)

**Câblage:**

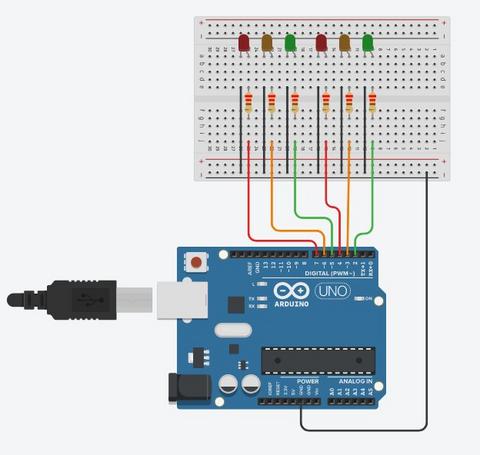
* Broche Digital **2**- Feu **Vert 1**
* Broche Digital **3**- Feu**Orange 1**
* Broche Digital **4** - Feu **Rouge 1**
* Broche Digital **5** - Feu **Vert 2**
* Broche Digital **6** - Feu **Orange 2**
* Broche Digital **7** - Feu **Rouge 2**

Chaque **Cathode**des leds est reliée au **GND**

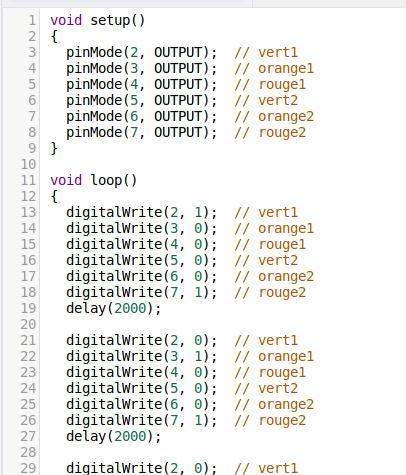
Chaque **Anode**des leds est reliée à une patte d'une résistance de **220 Ohms**

L'autre patte de la résistance est reliée à une **Pin Digital** comme indiquée ci dessous

## **Le Schéma**

* [](http://emery.claude.free.fr/_media/img/medium/feux-tricolores-cablage.jpg)

## **Le Code**

[](http://emery.claude.free.fr/_media/img/small/feux-tricolore-code.jpg)

void setup()  
{  
  pinMode(2, OUTPUT);  // vert1  
  pinMode(3, OUTPUT);  // orange1  
  pinMode(4, OUTPUT);  // rouge1  
  pinMode(5, OUTPUT);  // vert2  
  pinMode(6, OUTPUT);  // orange2  
  pinMode(7, OUTPUT);  // rouge2  
}  
void loop()  
{  
  digitalWrite(2, 1);  // vert1  
  digitalWrite(3, 0);  // orange1  
  digitalWrite(4, 0);  // rouge1  
  digitalWrite(5, 0);  // vert2  
  digitalWrite(6, 0);  // orange2  
  digitalWrite(7, 1);  // rouge2  
  delay(2000);  
    
  digitalWrite(2, 0);  // vert1  
  digitalWrite(3, 1);  // orange1  
  digitalWrite(4, 0);  // rouge1  
  digitalWrite(5, 0);  // vert2  
  digitalWrite(6, 0);  // orange2  
  digitalWrite(7, 1);  // rouge2  
  delay(2000);  
    
  digitalWrite(2, 0);  // vert1  
  digitalWrite(3, 0);  // orange1  
  digitalWrite(4, 1);  // rouge1  
  digitalWrite(5, 1);  // vert2  
  digitalWrite(6, 0);  // orange2  
  digitalWrite(7, 0);  // rouge2  
  delay(2000);  
    
  digitalWrite(2, 0);  // vert1  
  digitalWrite(3, 0);  // orange1  
  digitalWrite(4, 1);  // rouge1  
  digitalWrite(5, 0);  // vert2  
  digitalWrite(6, 1);  // orange2  
  digitalWrite(7, 0);  // rouge2  
  delay(2000);  
    
}

## **Amélioration Du Code**

Nous allons améliorer le code :

* Utilisation de constante pour les feux
* Utilisation de temporisation entre chaque phase

A l'initialisation, les leds sont dans les états suivants:

* Vert1 = allumé
* Orange1 = éteint
* Rouge1 = éteint
* Vert2 = éteint
* Orange2 = éteint
* Rouge2 = allumé

**Les étapes:**

* Phase 1 - Rouge1 éteint, Vert1 et Rouge2 allumés  // Début du cycle
* Phase 2 - Temporisation
* Phase 3 - Vert1 éteint , Orange1 et Rouge2 allumés
* Phase 3 - Temporisation
* Phase 4 - Orange1 éteint, Rouge1 et Rouge2 allumés  // les 2 feux rouges sont allumés = SECURITE
* Phase 5 - Temporisation
* Phase 6 - Rouge2 éteint, Rouge1 et Vert2 allumés
* Phase 7 - Temporisation
* Phase 8 - Vert2 éteint, Rouge1 et Orange2 allumés
* Phase 9 - Temporisation
* Phase 10 - Orange2 éteint, Rouge1 et Rouge2 allumés  // les 2 feux rouges sont allumés = SECURITE
* Phase 11 - Temporisation
* On boucle .... // Retour à la phase 1

## **Le Code Amélioré**

[](http://emery.claude.free.fr/_media/img/medium/feu-tricolore-code-ameliore.jpg)

  // Initialisation des variables  
    const byte Vert1 = 2 ;  
    const byte Orange1 = 3 ;  
    const byte Rouge1 = 4 ;  
    const byte Vert2 = 5 ;  
    const byte Orange2 = 6 ;  
    const byte Rouge2 = 7 ;  
       
    //Tempo  
    const long TempoRougeSeul = 2000;  // Tempo feu rouge seul  
    const long TempoVert = 2000;  // Tempo feu vert  
    const long TempoOrange = 2000;  // Tempo feu orange  
       
    // Initialisation des pin 2 à  7 en OUTPUT  
    void setup () {  
      pinMode (Vert1, OUTPUT) ;  
      pinMode (Orange1, OUTPUT) ;  
      pinMode (Rouge1, OUTPUT) ;  
      pinMode (Vert2, OUTPUT) ;  
      pinMode (Orange2, OUTPUT) ;  
      pinMode (Rouge2, OUTPUT) ;  
    }  
       
    void loop () {  
      // Etat des leds au départ  
      digitalWrite (Vert1, 1) ;  // Si Vert1 = 0 et Rouge2 = 0,   
      digitalWrite (Orange1, 0) ;  // ajouter les deux lignes  
      digitalWrite (Rouge1, 0) ;  // Allumage de Vert1 et Rouge2  
      digitalWrite (Vert2, 0) ;  
      digitalWrite (Orange2, 0) ;  
      digitalWrite (Rouge2, 1) ;  // .........  
        
      // Allumage de Vert1 et Rouge2  
      //digitalWrite (Vert1, 1) ;  
      //digitalWrite (Rouge2, 1) ;  
       
      // Début du cycle  
      // Feu n°1  
      delay (TempoVert) ; // Tempo  
      digitalWrite (Vert1, 0) ; // Vert1 éteint  
      digitalWrite (Orange1, 1) ; // Orange1 allumé  
      delay (TempoOrange) ;  // Tempo  
      digitalWrite (Orange1, 0) ; // Orange1 éteint  
      digitalWrite (Rouge1, 1) ; // Rouge1 allumé  
      delay (TempoRougeSeul) ; // Tempo  
       
      // Feu n°2  
      digitalWrite (Rouge2, 0) ; // Rouge2 éteint  
      digitalWrite (Vert2, 1) ; // Vert2 allumé  
      delay (TempoVert) ; // Tempo  
      digitalWrite (Vert2, 0) ; // Vert2 éteint  
      digitalWrite (Orange2, 1) ; // Orange2 allumé  
      delay (TempoOrange) ; // Tempo  
      digitalWrite (Orange2, 0) ; // Orange2 éteint  
      digitalWrite (Rouge2, 1) ; // Rouge2 allumé  
      delay (TempoRougeSeul) ; // Tempo  
      digitalWrite (Vert1, 1) ; // Vert1 allumé  
       
      // On boucle ....  
    }

EXPLICATION DU CODE

###### L’organigramme

Cette fois, l’organigramme a changé de forme, c’est une liste. Comment le lire ? De haut en bas ! Le premier élément du programme commence après le début, le deuxième élément, après le premier, etc.

* **DEBUT**
* /\* première partie du programme, on s'occupe principalement du deuxième feu \*/
* Allumer led\_rouge\_feux\_1
* Allumer led\_verte\_feux\_2
* Attendre 3 secondes
* Éteindre led\_verte\_feux\_2
* Allumer led\_jaune\_feux\_2
* Attendre 1 seconde
* Éteindre led\_jaune\_feux\_2
* Allumer led\_rouge\_feux\_2
* /\* deuxième partie du programme, pour l'instant : led\_rouge\_feux\_1 et led\_rouge\_feux\_2 sont allumées; on éteint donc la led\_rouge\_feux\_1 pour allumer la led\_verte\_feux\_1 \*/
* Attendre 3 secondes
* Éteindre led\_rouge\_feux\_1
* Allumer led\_verte\_feux\_1
* Attendre 3 secondes
* Éteindre led\_verte\_feux\_1
* Allumer led\_jaune\_feux\_1
* Attendre 1 seconde
* Éteindre led\_jaune\_feux\_1
* Allumer led\_rouge\_feux\_1
* **FIN**

# **Conclusion**

Parvenu au terme de notre travail nous pouvons dire sans risque de nous tromper que notre projet dans presque toute son intégralité a été réussi car il satisfait tous les objectifs que nous nous sommes fixés dès le début du travail, celui de simuler des feux de circulation sur vscode.

SOURCE : <https://zestedesavoir.com/tutoriels/686/arduino-premiers-pas-en-informatique-embarquee/743_gestion-des-entrees-sorties/3422_tp-feux-de-signalisation-routiere/>

<https://www.tubefr.com/simulateur-de-feux-tricolores-arduino_7.html#title>

https://www.researchgate.net/publication/335928612\_ETUDE\_ET\_SIMULATION\_DES\_FEUX\_DE\_CIRCULATION\_CAS\_DU\_CARREFOUR\_POSTE\_CENTRALE\_DE\_YAOUNDE\_CAMEROUN