

Gestion de feux tricolores avec Arduino

Auteur :
Alain

SON OBJECTIF

Pouvoir initier le jeune public présent à La Fabrique aux joies de la programmation avec Arduino

FABRICATION À LA DÉCOUPEUSE LASER

Toutes les pièces ont été découpées avec la découpeuse laser, dans du médium de 3mm

Les fichiers de découpe sont disponibles ici : [Documentation feux carrefour](#)

PROGRAMMATION ARDUINO

Bon, maintenant il faut se creuser un peu les méninges ...

Structure globale du programme

Ce programme simple comprend :

- un entête déclaratif
- une partie « configuration » qui ne sera exécutée qu'une fois (fonction setup())
- une partie constituée d'une boucle sans fin que le programme répètera à l'infini (fonction loop()) : c'est le cœur du programme.

Déroulement du programme

Le programme se déroule de la façon suivante :

- Après avoir pris en compte les instructions de la partie déclarative,
- puis après avoir exécuté la partie configuration (fonction setup()),
- le programme bouclera sans fin (fonction loop ()), exécutant de façon répétée le code compris dans la boucle sans fin.

Au niveau de la partie déclarative :

Déclaration des constantes de broches

- Déclaration des constantes pour les broches utilisées dans le programme : on renomme les 6 broches utilisées avec les LEDs.

```
const int ROUTEROUGE_1/3=7;      //déclaration constante de broche 7
const int ROUTEORANGE_1/3=6;     //déclaration constante de broche 6
const int ROUTEVERT_1/3=5;       //déclaration constante de broche 5
const int PIETONROUGE_1/3=8;     //déclaration constante de broche 8
const int PIETONVERT_1/3=9;      //déclaration constante de broche 9
const int ROUTEROUGE_2/4=13;     //déclaration constante de broche 13
const int ROUTEORANGE_2/4=12;    //déclaration constante de broche 12
const int ROUTEVERT_2/4=11;      //déclaration constante de broche 11
const int PIETONROUGE_2/4=2;     //déclaration constante de broche 2
```

```
const int PIETONVERT_2/4=3;      //déclaration constante de broche 3
```

Initialisation des fonctionnalités utilisées :

- On initialise les LEDs avec le rouge du feu 1 allumé et le vert du feu 2 allumé à l'aide de l'instruction [digitalWrite](#) suivi d'une pause avec l'instruction [delay](#) :

```
//—— initialisation des LEDs au début = feu route 1/3 au rouge et feu route 2/4 au vert
```

```
digitalWrite(ROUTEROUGE_1/3,HIGH);      // met la broche 7 au niveau HAUT
```

```
digitalWrite(PIETONVERT_2/4,HIGH);      // met la broche 3 au niveau HAUT
```

```
digitalWrite(VERT_1/3,LOW);              // met la broche 5 au niveau BAS
```

```
digitalWrite(ORANGE_1/3,LOW);            // met la broche 6 au niveau BAS
```

```
digitalWrite(ROUGE_2/4,LOW);              // met la broche 13 au niveau BAS
```

```
digitalWrite(PIETONROUGE_1/3,HIGH);      met la broche 8 au niveau HAUT
```

```
digitalWrite(VERT_2/4,HIGH);              // met la broche 11 au niveau HAUT
```

```
digitalWrite(ORANGE_2/4,LOW);            // met la broche 12 au niveau BAS
```

```
delay(5000); // pause de 5 secondes
```

Au niveau de la boucle principale, la fonction loop () :

- on réalise la séquence d'allumage / extinction des 6 LEDs de façon à obtenir la simulation du fonctionnement de 2 feux tricolores :

```
//— le feu route 2/4 passe à l'orange
```

```
digitalWrite(ROUTEVERT_2/4,LOW);          // éteint vert feu route 2/4
```

```
digitalWrite(PIETONVERT1/3,LOW);          // éteint vert feu piéton 1/3
```

```
digitalWrite(ROUTEORANGE_2/4,HIGH);        // allume orange feu route 2/4
```

```
digitalWrite(PIETONROUGE1/3,HIGH);         // allume rouge feu piéton 1/3
```

```
delay (2000);                             // pause 2 secondes
```

```
//— le feu route 2/4 passe au rouge
```

```
digitalWrite(ROUTEORANGE_2/4,LOW);         // éteint orange feu route 2/4
```

```
digitalWrite(ROUTEROUGE_2/4,HIGH);         // allume rouge feu route 2/4
```

```
delay (1000);                             // pause 1 seconde
```

```
//— le feu route 1/3 passe au vert
```

```
digitalWrite(ROUTEROUGE_1/3,LOW);          // éteint rouge feu route 1/3
```

```
digitalWrite(PIETONVERT_2/4,HIGH);         // allume vert feu piéton 2/4
```

```
digitalWrite(ROUTEVERT_1/3,HIGH);         // allume vert feu route 1/3
```

```
delay (5000);                             // pause 5 secondes
```

```
//— le feu 1/3 passe à l'orange
```

```
digitalWrite(ROUTEVERT_1/3,LOW);          // éteint vert feu 1/3
```

```
digitalWrite(PIETONVERT2/4,LOW);          // éteint vert feu piéton 2/4
```

```
digitalWrite(PIETONROUGE2/4,HIGH);         // allume rouge feu piéton 2/4
```

```
digitalWrite(ROUTEORANGE_1/3,HIGH);       // allume orange feu route 1/3
```

```
delay (2000);                             // pause 2 secondes
```

```

//— le feu 1/3 passe au rouge
digitalWrite(ROUTEORANGE_1/3,LOW);      // éteint orange feu route 1/3
digitalWrite(ROUTEROUGE_1/3,HIGH);      // allume rouge feu 1/3
delay (1000); // pause courte 1 seconde

//— le feu 2 passe au vert
digitalWrite(ROUTEROUGE_2/4,LOW);      // éteint rouge feu route 2/4
digitalWrite(PIETONVERT_1/3,HIGH);      // allume vert feu piéton 1/3
digitalWrite(ROUTEVERT_2/4,HIGH);      // allume vert feu route 2/4
delay (5000);                          // pause longue 5 secondes

```

- Le programme recommence en boucle les instructions de la fonction loop().

Mise en œuvre du programme

Préparation du montage et programmation de la carte Arduino :

- Commencer par réaliser le montage sur plaque d'expérimentation
- Ensuite, programmer la carte Arduino avec ce programme selon la procédure habituelle

Fonctionnement

- Les LEDs s'allument successivement réalisant la séquence des feux de circulation.

Le programme complet en langage Arduino :

```

// Connecter sur la broche 2 (configurée en sortie) une LED rouge connectée en série avec sa résistance
au 0V

// Connecter sur la broche 3 (configurée en sortie) une LED jaune connectée en série avec sa résistance
au 0V

// Connecter sur la broche 4 (configurée en sortie) une LED verte connectée en série avec sa résistance
au 0V

// Connecter sur la broche 5 (configurée en sortie) une LED rouge connectée en série avec sa résistance
au 0V

// Connecter sur la broche 6 (configurée en sortie) une LED jaune connectée en série avec sa résistance
au 0V

// Connecter sur la broche 7 (configurée en sortie) une LED verte connectée en série avec sa résistance
au 0V

// * Entête déclarative A ce niveau sont déclarées les librairies incluses, les constantes, les variables...

// — Inclusion des librairies utilisées —

// — Déclaration des constantes —

// — constantes des broches —

const int ROUGE_1=2;    //déclaration constante de broche 2
const int ORANGE_1=3;  //déclaration constante de broche 3
const int VERT_1=4;     //déclaration constante de broche 4
const int ROUGE_2=5;    //déclaration constante de broche 5
const int ORANGE_2=6;   //déclaration constante de broche 6
const int VERT_2=7;     //déclaration constante de broche 7

```

// — Déclaration des variables globales —

/ FONCTION SETUP = Code d'initialisation La fonction setup() est exécutée en premier et 1 seule fois, au démarrage du programme*/*

Void **setup()**

{ // début de la fonction setup() - ici instructions à exécuter au démarrage —

// ——— Broches en sortie ———

pinMode(ROUGE_1, OUTPUT); //met la broche 2 en sortie

pinMode(VERT_1, OUTPUT); //met la broche 4 en sortie

pinMode(ORANGE_1, OUTPUT); //met la broche 3 en sortie

pinMode(ROUGE_2, OUTPUT); //met la broche 5 en sortie

pinMode(VERT_2, OUTPUT); //met la broche 7 en sortie

pinMode(ORANGE_2, OUTPUT); //met la broche 6 en sortie

//- initialisation des LEDs au début = feu 1 au rouge et feu 2 au vert

digitalWrite(ROUGE_1,HIGH); // met la broche 2 au niveau HAUT

digitalWrite(VERT_1,LOW); // met la broche 4 au niveau BAS

digitalWrite(ORANGE_1,LOW); // met la broche 3 au niveau BAS

digitalWrite(ROUGE_2,LOW); // met la broche 5 au niveau BAS

digitalWrite(VERT_2,HIGH); // met la broche 7 au niveau HAUT

digitalWrite(ORANGE_2,LOW); // met la broche 6 au niveau BAS

delay(5000); // pause de 5 secondes

}// fin de la fonction setup()

/ FONCTION LOOP = Boucle sans fin = coeur du programme la fonction loop() s'exécute sans fin en boucle aussi longtemps que l'Arduino est sous tension*/*

void **loop()**

{// debut de la fonction loop()

//— le feu 2 passe à l'orange

digitalWrite(VERT_2,LOW); // éteint vert feu 2

digitalWrite(ORANGE_2,HIGH); // allume orange feu 2

delay (2000); // pause 2 secondes

//— le feu 2 passe au rouge

digitalWrite(ORANGE_2,LOW); // éteint orange feu 2

digitalWrite(ROUGE_2,HIGH); // allume rouge feu 2

delay (1000); // pause courte 1 seconde

//— le feu 1 passe au vert

digitalWrite(ROUGE_1,LOW); // éteint rouge feu 1

digitalWrite(VERT_1,HIGH); // allume vert feu 1

delay (5000); // pause longue 5 secondes

//— le feu 1 passe à l'orange

```
digitalWrite(VERT_1,LOW); // éteint vert feu 1
digitalWrite(ORANGE_1,HIGH); // allume orange feu 1
delay (2000); // pause 2 secondes
//— le feu 1 passe au rouge
digitalWrite(ORANGE_1,LOW); // éteint orange feu 1
digitalWrite(ROUGE_1,HIGH); // allume rouge feu 1
delay (1000); // pause courte 1 seconde
//— le feu 2 passe au vert
digitalWrite(ROUGE_2,LOW); // éteint rouge feu 2
digitalWrite(VERT_2,HIGH); // allume vert feu 2
delay (5000); // pause longue 5 secondes
} // fin de la fonction loop() – le programme recommence au début de la fonction loop sans fin
```

Publié le 31 août 2019