



```

const int brochePulsante = 2; // Broche du bouton poussoir

const int brocheLED[] = {3, 4, 5, 6, 7, 8, 9, 10}; // Broches des LEDs

const int nombreLED = sizeof(brocheLED) / sizeof(brocheLED[0]); // Nombre total de LEDs

const int brocheLEDModalite[] = {11, 12, 13}; // Broches pour les LEDs de mode

int ledCourante = 0; // LED actuellement allumée

int mode = 0; // Mode actuel

unsigned long tempsAppuiPulsante = 0; // Temps de l'appui sur le bouton

const unsigned long tempsAppuiLong = 1000; // 1 seconde pour un appui long

boolean boutonPresse = false; // Indicateur de l'appui sur le bouton

boolean enSelectionMode = false; // Indicateur de la sélection de mode

int modeSelectionne = 0; // Mode sélectionné

void setup() {
    pinMode(brochePulsante, INPUT_PULLUP);

    // Configuration des broches des LEDs
    for (int i = 0; i < nombreLED; i++) {
        pinMode(brocheLED[i], OUTPUT);
        digitalWrite(brocheLED[i], LOW);
    }

    // Configuration des broches des LEDs de mode

```

```

for (int i = 0; i < 3; i++) {
    pinMode(brocheLEDModalite[i], OUTPUT);
    digitalWrite(brocheLEDModalite[i], LOW);
}
}

void loop() {
    int etatPulsante = digitalRead(brochePulsante);

    // Gestion de l'appui sur le bouton
    if (etatPulsante == LOW) {
        if (!boutonPresse) {
            boutonPresse = true;
            tempsAppuiPulsante = millis();
        }
        if (millis() - tempsAppuiPulsante > tempsAppuiLong && !enSelectionMode) {
            enSelectionMode = true;
            tempsAppuiPulsante = millis(); // Réinitialiser le temps pour la sélection de mode
        }
    } else if (boutonPresse) {
        boutonPresse = false;
        if (!enSelectionMode) {
            // Appui court - changer la LED en mode 1 ou sélectionner le mode
            if (mode == 0) { // Mode 1
                digitalWrite(brocheLED[ledCourante], LOW);
                ledCourante = (ledCourante + 1) % nombreLED;
                digitalWrite(brocheLED[ledCourante], HIGH);
            } else { // Sélectionner le mode
                mode = modeSelectionne;
                executerMode();
            }
        }
    }
}

```

```

} else {

    // Relâchement après un appui long - sélectionner le mode

    enSelectionMode = false;

    mode = modeSelectionne;

    executerMode();

}

}

// Gestion de la sélection de mode

if (enSelectionMode) {

    unsigned long tempsActuel = millis();

    if (tempsActuel - tempsAppuiPulsante >= 500) {

        tempsAppuiPulsante = tempsActuel;

        modeSelectionne = (modeSelectionne + 1) % 3; // Passer au mode suivant

    }

    // Mise à jour des LEDs de mode en fonction du mode sélectionné

    if (modeSelectionne == 0){

        digitalWrite(11, LOW);

        digitalWrite(12,HIGH);

        digitalWrite(13, HIGH);

    }

    if (modeSelectionne == 1){

        digitalWrite(11, HIGH);

        digitalWrite(12,LOW);

        digitalWrite(13, HIGH);

    }

    if (modeSelectionne == 2){

        digitalWrite(11, HIGH);

        digitalWrite(12,HIGH);

        digitalWrite(13, LOW);

    }

}

```

```

    }
}

// Mode de fonctionnement
if (!enSelectionMode) {
    executerMode();
}
}

void executerMode() {
    for (int i = 0; i < 3; i++) {
        digitalWrite(brocheLEDModalite[i], LOW);
    }
    digitalWrite(brocheLEDModalite[mode], HIGH);

    // Exécution des différentes modalités
    switch (mode) {
        case 0:
            // La modalité 0 est gérée directement dans la boucle lors de l'appui sur le bouton
            digitalWrite(11, LOW);
            digitalWrite(12, HIGH);
            digitalWrite(13, HIGH);
            break;

        case 1: // Modalité 2 dans votre code original - Allumage progressif
            digitalWrite(11, HIGH);
            digitalWrite(12, LOW);
            digitalWrite(13, HIGH);
            for (int i = 0; i < nombreLED; i++) {
                digitalWrite(brocheLED[i], HIGH);
                delay(500);
                digitalWrite(brocheLED[i], LOW);
            }
        }
    }
}

```

```
}  
break;  
case 2: // Modalité 3 dans votre code original - Allumage continu  
    digitalWrite(11, HIGH);  
    digitalWrite(12,HIGH);  
    digitalWrite(13, LOW);  
    for (int i = 0; i < nombreLED; i++) {  
        digitalWrite(brocheLED[i], HIGH);  
    }  
    delay(100); // Léger délai pour voir l'effet  
    for (int i = 0; i < nombreLED; i++) {  
        digitalWrite(brocheLED[i], LOW);  
    }  
    break;  
}  
}
```

<https://www.tinkercad.com/things/i62QU447znX-arnaud-cour-1-projet>