

Tp: Multitâche avec Interruptions

Kennouche Abderrahmane

Réaliser un système multitâche sur Arduino Uno en utilisant des **interruptions** pour gérer plusieurs tâches simultanément. Les tâches sont :

1. Faire clignoter une LED toutes les 1 seconde.
2. Allumer une LED lorsqu'un bouton est pressé.
3. Contrôler la luminosité d'une LED via le moniteur série.
4. Afficher la luminosité de la LED toutes les 5 secondes.

1 Matériel utilisé

- Arduino Uno (ATmega328P)
- 3 LEDs
- 3 résistances ($220\ \Omega$)
- 1 bouton-poussoir
- Câbles de connexion
- Simulateur SimulIDE

2 Principe de fonctionnement

2.1 Timers et interruptions

- **Timer** : compteur interne du microcontrôleur qui permet de mesurer le temps avec précision.
- **Interruption (ISR)** : événement qui interrompt le programme principal pour exécuter un code spécifique.

2.2 Tâches

1. LED1 : clignote toutes les secondes via Timer1.
2. LED2 : allumée tant que le bouton est pressé.
3. LED3 : luminosité contrôlée par l'utilisateur via Serial.
4. Affichage brightness : toutes les 5 secondes via Timer2 et un drapeau.

3 Description du code

3.1 Configuration des broches

```

1 pinMode(LED1, OUTPUT);
2 pinMode(LED2, OUTPUT);
3 pinMode(LED3, OUTPUT);
4 pinMode(BTN, INPUT_PULLUP);

```

- LEDs → sorties - Bouton → entrée avec résistance Pull-Up interne

3.2 Timer1 pour LED1

```

1 TCCR1A = 0;
2 TCCR1B = 0;
3 TCNT1 = 0;
4 OCR1A = 15625;
5 TCCR1B |= (1 << WGM12);
6 TCCR1B |= (1 << CS12) | (1 << CS10);
7 TIMSK1 |= (1 << OCIE1A);

```

- Mode CTC : remise à zéro à chaque comparaison - Prescaler 1024 pour atteindre 1 seconde - Activation de l'interruption Compare A

3.3 Timer2 pour impression luminosité

```

1 TCCR2A = 0;
2 TCCR2B = 0;
3 TCNT2 = 0;
4 TCCR2B |= (1 << CS22) | (1 << CS21) | (1 << CS20);
5 TIMSK2 |= (1 << TOIE2);

```

- Timer2 déborde pour déclencher un flag toutes les 5 secondes

3.4 Lecture du bouton

```

1 if (digitalRead(BTN) == LOW) {
2     digitalWrite(LED2, HIGH);
3 } else {
4     digitalWrite(LED2, LOW);
5 }

```

- LED2 allumée tant que le bouton est pressé

3.5 Lecture Serial pour LED3

```

1 if (Serial.available()) {
2     String input = Serial.readStringUntil('\n');
3     int val = input.toInt();
4     if (val >= 0 && val <= 255) {
5         brightness = val;
6         analogWrite(LED3, brightness);
7     }
8 }

```

- PWM de LED3 contrôlé via moniteur série

3.6 ISR Timer1 pour LED1

```
1 ISR(TIMER1_COMPA_vect) {  
2     LED1_state = !LED1_state;  
3     digitalWrite(LED1, LED1_state);  
4 }
```

- Inversion de l'état de LED1 toutes les secondes

3.7 ISR Timer2 pour affichage luminosité

```
1 ISR(TIMER2_OVF_vect) {  
2     timer2_count++;  
3     if (timer2_count >= 312) {  
4         print_flag = true;  
5         timer2_count = 0;  
6     }  
7 }
```

- Compte les overflow de Timer2 pour afficher la luminosité toutes les 5 secondes

4 Schémas et captures d'écran

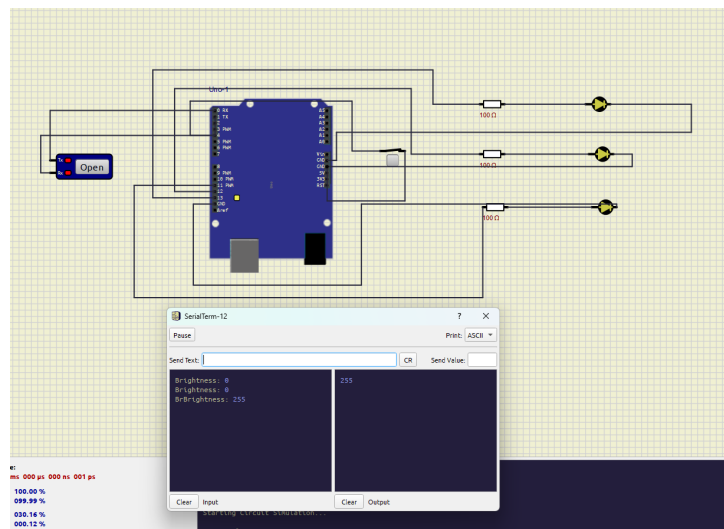


FIGURE 1 – Schéma du montage réalisé sur SimulIDE