

TP4 : Commande d'un moteur pas-à-pas via Timer1 en mode CTC

Objectifs

- Comprendre le fonctionnement d'un moteur pas-à-pas
- Utiliser Timer1 en mode CTC pour générer des impulsions précises
- Gérer vitesse, accélération et position

Matériel requis

- Arduino UNO
- Moteur pas-à-pas (unipolaire 28BYJ-48 ou bipolaire NEMA17)
- Driver moteur (ULN2003 pour unipolaire, A4988 pour bipolaire)
- Alimentation adaptée

Contenu détaillé

Partie 1 : Principe du moteur pas-à-pas

1. Séquence de pas :
 - a. Plein pas : 4 phases
 - b. Demi-pas : 8 phases pour plus de précision
2. Calcul de l'angle par pas (1.8° pour NEMA17)

Partie 2 : Configuration du Timer1 en mode CTC

1. Initialisation sans interruption :

```
void initTimer1CTC(unsigned int compareValue) {  
  noInterrupts();  
  TCCR1A = 0;  
  TCCR1B = 0;  
  TCNT1 = 0;  
  OCR1A = compareValue;  
  TCCR1B |= (1 << WGM12); // Mode CTC  
  TCCR1B |= (1 << CS10); // Prescaler 1  
  TIMSK1 |= (1 << OCIE1A); // Activer interruption  
  interrupts();  
}
```

}

2. Calcul de la valeur de comparaison selon la fréquence de pas souhaitée

Partie 3 : ISR pour la génération de pas

```
ISR(TIMER1_COMPA_vect) {  
    // Avancer d'un pas  
    faireUnPas(sens);  
    compteurPas++;  
  
    // Gestion de l'accélération  
    if (compteurPas < seuilAccel) {  
        // Augmenter fréquence (diminuer OCR1A)  
        ajusterFrequence(++frequence);  
    }  
}
```

Partie 4 : Profil de mouvement

1. Accélération progressive (rampe)
2. Vitesse constante
3. Décélération progressive
4. Gestion de position absolue et relative

Partie 5 : Commande par moniteur série

1. Syntaxe de commandes : POS 2000 (2000 pas), SPD 500 (500 pas/s)
2. Retour d'état : position actuelle, en mouvement/arrêt