Université Djilali BOUNAAMA, Khemis Miliana
مليانة خميس بونعامة جيلالي جامعة
Faculté des Sciences et de la Technologie
والتكنولوجيا العلوم كلية

Série d'exercices 5 : Programmation des interruptions et des temporisations

Exercice 1 : Configuration d'un port parallèle en sortie

On utilise un microcontrôleur 8 bits (ex : PIC16F877A) avec un port B (RB0 à RB7). On veut piloter 8 LEDs connectées à ce port.

1. Configurez les registres de direction (TRISB) pour que toutes les broches soient en sortie.
2. Écrivez une fonction allume_leds() qui allume les LEDs sur les 4 bits de poids faible et éteint les 4 bits de poids fort.
3. Quelle est la valeur hexadécimale à écrire dans le registre PORTB ?

Exercice 2 : Temporisation simple avec un Timer (mode 8 bits)

On utilise un Timer0 (8 bits, prédiviseur 1:256). L'horloge système est à 4 MHz (cycle d'instruction = 1 µs).

1. Calculez la durée d'un débordement du Timer0 (valeur initiale 0).
2. Programmez le Timer0 pour générer une temporisation de 50 ms en utilisant une valeur de préchargement.
3. Donnez la valeur à écrire dans TMR0.

Exercice 3 : Mesure de durée d'une impulsion externe

Énoncé
Un signal carré est appliqué sur l'entrée T0CKI (Timer0 en mode compteur d'événements). On mesure une largeur d'impulsion haute de 200 µs. Sachant que le prédiviseur est à 1:4 et que le timer est incrémenté sur chaque front montant, combien d'incréments sont comptés ? Quelle est la valeur lue dans TMR0 si on part de 0 ?

Exercice 4 : Interruption sur débordement de Timer

On veut générer une interruption toutes les 10 ms avec Timer2 (8 bits, postdiviseur 1:1, prédiviseur 1:16). Horloge 4 MHz.

1. Calculez la valeur de période PR2.
2. Écrivez la routine d'interruption qui bascule une LED à chaque interruption.

Exercice 5 : Configuration et utilisation de l'ADC

Énoncé
Un capteur de température délivre 0–5V. On utilise un ADC 10 bits sur le microcontrôleur avec Vref = 5V.

1. Calculez la résolution en mV/bit.
2. Quelle valeur numérique lira-t-on pour 2,3 V ?
3. Configurez l'ADC pour un canal AN0, horloge Fosc/8 (Fosc=8 MHz), et lancez une conversion.

Exercice 6 : Interruption externe et PWM par timer

Énoncé
On veut générer un signal PWM de 1 kHz, rapport cyclique 30%, sur une broche de sortie, à l'aide d'un timer (PWM hardware).

1. Si l'horloge système est 8 MHz, calculez la valeur de période et de rapport cyclique pour le timer.
2. Expliquez comment une interruption externe (bouton poussoir sur INT0) pourrait augmenter le rapport cyclique de 10% à chaque appui.

.

Université Djilali BOUNAAMA, Khemis Miliana
مليانة خميس بونعامة جيلالي جامعة
Faculté des Sciences et de la Technologie
والتكنولوجيا العلوم كلية

Solution Série 5

Exercice 1 :

Ordre correct :

1. Création du projet
2. Écriture du code source
3. Compilation / assemblage
4. Édition de liens
5. Exécution / débogage

Exercice 2 :

IDE → C
Linker → B
Hex file → A
Programmeur → E
Breakpoint → D

Exercice 3 :

- Prédiviseur 1:4 → une incrémentation tous les 4 fronts.

- Pour mesurer la durée haute, on compte les fronts montants pendant 200 µs.
- Si la période du signal externe est inconnue, on suppose que le timer compte directement les fronts :
  En réalité, on ne peut pas connaître le nombre sans la fréquence du signal. Reformulons :
  Hypothèse : Le timer est incrémenté par l'horloge système avec prédiviseur.
  Si l'impulsion est mesurée en mode capture, on utilise plutôt Timer1. Mais ici, avec T0CKI, on compte les fronts du signal externe.
  On donne une solution générique :
  Le nombre de fronts = (durée) × (fréquence du signal).
  Donc pour 200 µs, si le signal externe a une période T, nombre de fronts = 200 µs / T.
  Sans T, on ne peut pas calculer.

- **Correction** : On choisit une fréquence externe de 10 kHz (période 100 µs) → fronts montants tous les 100 µs.
  Pendant 200 µs → 2 fronts.
  Prédiviseur 1:4 → le timer s'incrémente tous les 4 fronts → 0 incrément. Donc TMR0 = 0.

- (Exercice volontairement ambigu pour discussion, mais la solution propre consiste à dire que le mode compteur nécessite la connaissance de la fréquence externe.)

Exercice 4 :

1. Pas d'incrémentation timer = 1 µs × 16 = 16 µs.
   Période Timer2 = (PR2 + 1) × 16 µs.
   On veut (PR2 + 1) × 16 µs = 10 ms = 10000 µs
   → PR2 + 1 = 10000 / 16 = 625
   → PR2 = 624 (valeur > 255, impossible en 8 bits !)
   → Il faut utiliser un postdiviseur ou prédiviseur plus grand.
   Avec prédiviseur 1:64 : pas = 64 µs → PR2+1 = 10000/64 = 156,25 → pas entier.
   Avec prédiviseur 1:256 : pas = 256 µs → 10000/256 ≈ 39,06 → PR2=38, période = 39×256 µs = 9984 µs (≈10 ms).
   Donc PR2 = 38 (0x26).
2. Code interruption (PIC C) :

```
void interrupt timer2_isr() {
   if (TMR2IF) {
      TMR2IF = 0;
      RB0 = !RB0;  // Bascule LED sur RB0
   }
}
```

Exercice 5 :

1. Résolution = 5V / 1024 = 0,0048828125 V ≈ 4,88 mV/bit.
2. Valeur = (2,3 / 5) × 1024 = 0,46 × 1024 = 471,04 → 471 (décimal).
3. Configuration :

```
ADCON0 = 0x01;  // Canal AN0, ADC on
ADCON1 = 0x80;  // Vref = Vdd, juste justification droite (si PIC)
ADCON2 = 0x82;  // Horloge Fosc/8, résultat justifié à droite, temps d'acquisition 2Tad (exemple)
GO_DONE = 1;
while(GO_DONE);
valeur = ADRESH << 8 | ADRESL;
```

Exercice 6 :

1. PWM 1 kHz → période = 1 ms.
   Avec Timer2 en mode PWM (ex : module CCP en PIC), la période = (PR2+1) × 4 × Tosc × prédiviseur.
   Soit prédiviseur 1:1, Tosc = 1/8 MHz = 0,125 µs, 4×Tosc = 0,5 µs.
   (PR2+1) × 0,5 µs = 1000 µs → PR2+1 = 2000 → PR2 = 1999 (impossible en 8 bits).
   Il faut augmenter le prédiviseur : 1:16 → 4×Tosc×16 = 8 µs → PR2+1 = 1000/8 = 125 → PR2 = 124 (possible).
   Rapport cyclique 30% → duty cycle = 30% de 125 = 37,5 → 37 (entier).
   Registre CCPR1L = 37 >> 2 (si résolution 10 bits) ou configuration simplifiée.
2. Dans l'ISR de INT0, on incrémente le rapport cyclique de 10% de la période, avec saturation à 100% :

```
void interrupt int0_isr() {
```

```
    if (INT0IF) {
        INT0IF = 0;
        duty += 12;  // 10% de 125 ≈ 12
        if (duty > 124) duty = 124;
        CCPR1L = duty >> 2;  // ajustement pour bits de poids faible
    }
}
```