

Série d'exercices 1: Architecture Des Systèmes Embarqués A Base De Microcontrôleur

Exercice 1 : Choix du microcontrôleur et estimation mémoire

On souhaite concevoir un système embarqué qui acquiert une température via un capteur numérique (12 bits toutes les 100 ms) et envoie le résultat par liaison série (UART). Le code programme occupe 8 Ko, les variables globales 2 Ko, la pile 1 Ko.

- Quelle taille de mémoire Flash et RAM devez-vous choisir ?
- Le microcontrôleur doit-il intégrer un convertisseur ADC ? Justifier.

Exercice 2 : Schéma bloc d'un microcontrôleur

Énoncé :

Dessiner le schéma bloc interne d'un microcontrôleur type (CPU, mémoire, périphériques, bus). Placer sur ce schéma les éléments suivants :

- Unités de calcul (ALU)
- Registres
- Mémoire programme (Flash)
- Mémoire données (RAM)
- Ports d'entrée/sortie (GPIO)
- Timer
- UART
- Bus système (adresse/données)

Exercice 3 : Gestion des interruptions

Un microcontrôleur reçoit une interruption externe sur la broche INT0 à chaque front montant d'un signal externe. Le programme principal affiche un compteur toutes les secondes.

- Expliquez ce que fait le processeur quand l'interruption survient.
- Quelle est la priorité si plusieurs interruptions arrivent en même temps (INT0, timer, UART) ?
- Donnez un extrait de code (pseudo-C) pour l'ISR (Interrupt Service Routine) qui incrémente un compteur global cpt.

Exercice 4 : Configuration d'un port GPIO

On utilise un microcontrôleur 8 bits (type AVR, PIC, STM8). Le port B est configuré ainsi :

- PB0 et PB1 en sortie
- PB2 en entrée avec pull-up activé
- PB3 en entrée sans pull-up
- Donner les valeurs des registres de configuration pour :
 - Direction (DDR)
 - Pull-up (si séparé)

Exercice 5 : Timer et génération d'un délai

Microcontrôleur cadencé à 16 MHz. On utilise un timer 16 bits avec prédiviseur /256.

- Quelle est la résolution du timer (temps par tick) ?
- Quelle valeur de registre (TOP) faut-il pour obtenir un débordement toutes les 50 ms ?
- Donner la formule.

Exercice 6 : Communication UART

On veut envoyer le caractère 'A' (0x41) à 9600 bauds, 8 bits, 1 stop, sans parité.

- Donner le chronogramme sur la broche TX (start bit, data bits LSB first).
- Calculer le temps total d'un caractère.
- Quel registre configure la vitesse en bauds (exemple pour STM32 ou AVR) ?

Solution Série 5

Exercice 1 :

- Flash nécessaire = code = 8 Ko
- RAM nécessaire = variables globales (2 Ko) + pile (1 Ko) = 3 Ko
- Choix : Flash \geq 8 Ko (souvent 16 Ko par sécurité), RAM \geq 3 Ko (ex: 4 Ko).
- ADC non obligatoire car capteur numérique (ex: DS18B20) \rightarrow communication 1-Wire ou I2C.

Exercice 2 : Réception USART avec interruption

[CPU]

├─ ALU

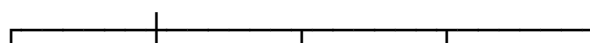
├─ Registres

├─ Contrôleur

|

├─ Bus système (adresse + données)

|



| | | | |
Flash RAM GPIO Timer UART

Exercice 3 :

1. Quand INT0 arrive :

- Le CPU finit l'instruction en cours.
- Sauvegarde le contexte (PC, registres).
- Va chercher l'adresse de l'ISR dans le vecteur d'interruption.
- Exécute l'ISR.
- Restaure le contexte et retourne au programme principal.

2. Priorité (selon config typique) : INT0 > Timer > UART (ou configurable).

3. Code :

```
volatile int cpt = 0;
```

```
void INT0_ISR(void) {
```

```
    cpt++;
```

```
}
```

```
// Dans main : initialisation IT
```

```
// IT rising edge, global interrupt enable
```

Exercice 4 :

(exemple AVR) :

- DDRB (0=entrée, 1=sortie) : PB0=1, PB1=1, PB2=0, PB3=0 → DDRB = 0b00000011 (0x03)
- PORTB (pull-up activé si DDR=0) : PB2=1, PB3=0 → PORTB = 0b00000100 (0x04)
- PB3 est en haute impédance.

Exercice 5 :

1. Fréquence timer = 16 MHz / 256 = 62 500 Hz
Période par tick = 1 / 62 500 = 16 µs.
2. Délai souhaité = 50 ms = 50 000 µs
Nombre de ticks = 50 000 / 16 = 3125 ticks.
3. Valeur TOP = 3125 - 1 = 3124 (car compteur de 0 à TOP).
En hexa : 0xC34.

Formule générale :

$$TOP = (\text{Delay} \times \text{FCPU} / \text{Prediviseur}) - 1$$

Exercice 6 :

1. Chronogramme (niveaux logiques) :

- Repos = 1
- Start = 0 pendant 1 bit
- LSB first : 'A' = 0x41 = b01000001 → bits dans l'ordre : 1,0,0,0,0,0,1,0
- Stop = 1 pendant 1 bit

Dessin :

 (approximatif)

2. Temps par bit = $1 / 9600 \approx 104,17 \mu\text{s}$
Trame = 1 start + 8 data + 1 stop = 10 bits
Temps total = $10 \times 104,17 \mu\text{s} = 1,0417 \text{ ms}$.
3. Registre :

- AVR : $UBRR = F_CPU / (16 \times \text{BAUD}) - 1$
- STM32 : $USART_BRR$

Exemple AVR avec $F_CPU=16 \text{ MHz}$, $\text{BAUD}=9600$:
 $UBRR = 16000000 / (16 \times 9600) - 1 = 104,16 \rightarrow 104 (0x68)$