

NOM :**Prénom :****Groupe :****Feuille réponse n°1****Exercice 1** (Expressions en C)

1°) Si une variable p1 de type char (8 bits signés) est déclarée écrire les expressions en C utilisant des masques (avec opérateur de décalage) permettant de :

- mettre à 1 le bit b3 et b2
- mettre à 0 le bit b4 et b1
- inverser le bit b6 (se fait facilement avec un ou exclusif)
- mettre à 1 le bit b7 et à 0 le bit b3

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

Réponses :

2°) Construire des expressions booléennes sur les tests suivants : expression vraie si

- le bit b7 est à 1 et le bit b4 est à 0 (de la variable p1)
- le bit b6 est à 1 ou le bit b2 est à 0 (de la variable p1)

Réponses :

3°) Donner toutes les valeurs possibles de l'expression :

$$((p1 \& 0x40) \gg 6) \wedge ((p1 \& 0x04) \gg 2)$$

pour toutes les valeurs de p1. Il y a 256 valeurs possibles pour p1, si vous ne trouvez aucune astuce de calcul vous y passerez quelques heures, soit bien plus que la durée du DS !

Réponses :

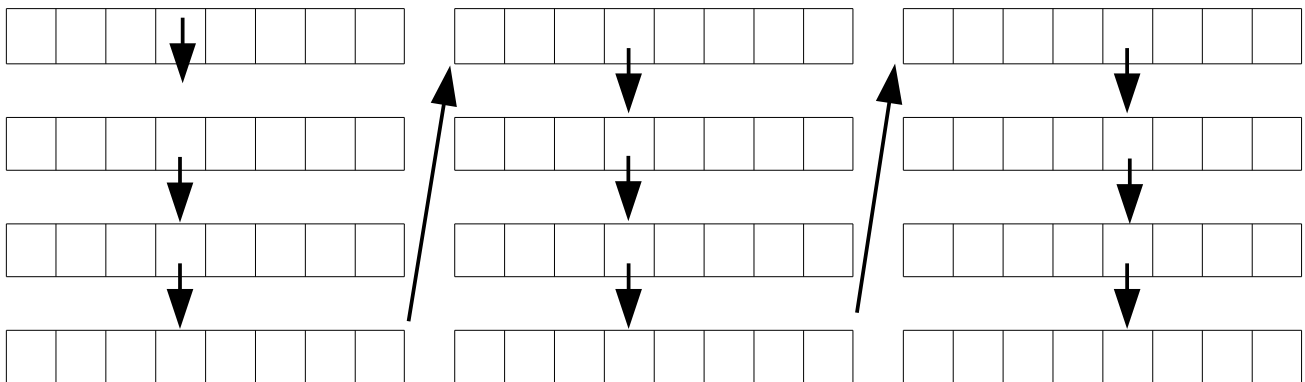
Exercice 2

Le PORTB est relié à un ensemble de LEDs permettant la réalisation de chenillards. Un chenillard est donné par le programme suivant :

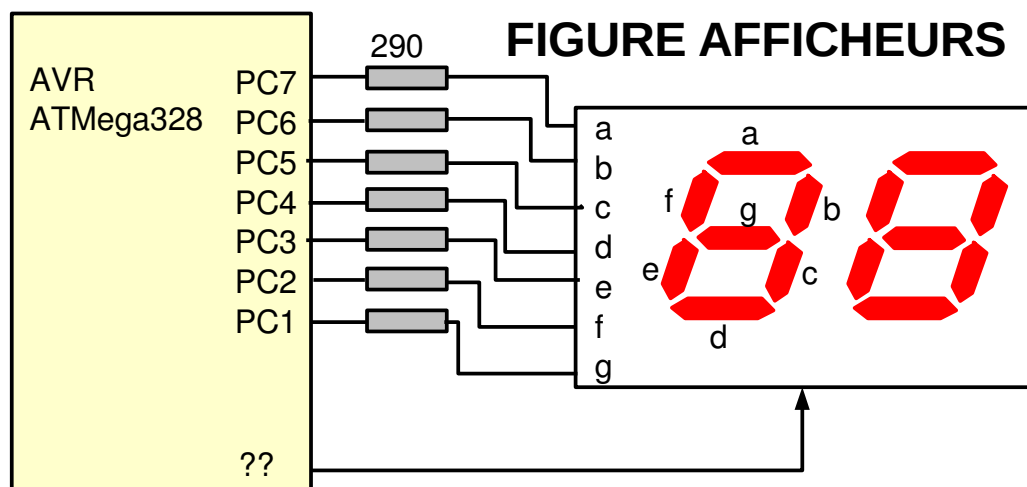
```
// ATmega328
main() {
    int i;
    DDRB=0xFF;
    while(1) {
        for(i=0;i<6;i++) {
            PORTB=(0x03<<i|0xC0>>i);
            _delay_ms(1000);
        }
        PORTB=(0xFF);
        _delay_ms(1000);
        PORTB=(0x00);
        _delay_ms(1000);
    }
}
```

1°) Compléter la suite des états de ce chenillard (ceux où une ou plusieurs leds sont allumées).

Réponse : (noircir les cases correspondantes quand la led est allumée)



On abandonne les chenillards pour le schéma simplifié suivant :



où PC1 à PC7 désignent les bits du PORTC. L'afficheur est à anode commune, c'est à dire que pour allumer le segment "a" il faut y mettre un '0' logique (vrai aussi pour les autres segments).

NOM :**Feuille réponse n°3****Prénom :**

On désire afficher la suite suivante :



sur un seul afficheur.

2°) Calculer les valeurs correspondantes à envoyer à l'afficheur sept segments pour chacune des lettres. On donnera le résultat en binaire et en hexadécimal.

Réponses :

3°) Écrire l'instruction C qui met l'ensemble de ces valeurs dans un tableau de huit cases.

Réponse :

4°) Écrire un programme C complet qui envoie toutes les 0,5 s l'ensemble de ces valeurs dans l'afficheur en vous aidant du tableau de la question 3.

Réponse :

5°) Écrire une fonction qui vous retourne la valeur correcte à envoyer sur le PORTC pour l'afficheur si on lui passe les caractères 'G', 'E', 'I', '2', '0', '1' et '5' et un tiret dans tous les autres cas.

Réponse : (à compléter)

```
// Avr-gcc
unsigned char texte7seg(unsigned char lettre){
    switch (lettre){
        case          : return          ;
```

case

6°) On désire maintenant utiliser le mode comparaison du timer 1 pour la commutation de l'affichage sur les deux afficheurs. A partir de la documentation, pouvez-vous dire à quoi correspondent les ?? de la **FIGURE AFFICHEURS (Feuille réponse 2)**?

Réponse sur figure correspondante

7°) On peut trouver sur Internet le sous-programme suivant :

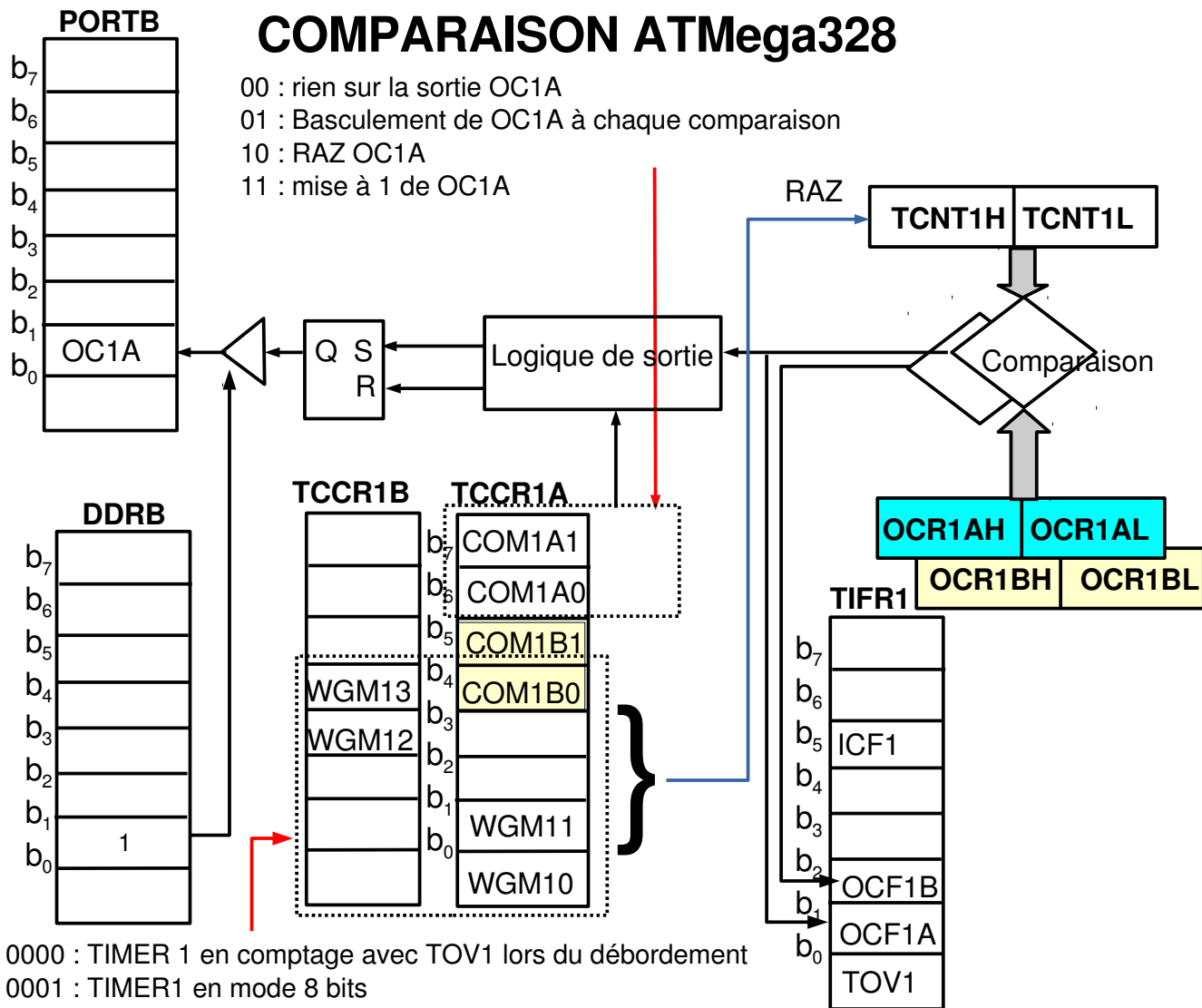
```
void timer1_setup (byte mode, int prescale, byte outmode_A, byte
outmode_B, byte capture_mode) {
    // enforce field widths for sanity
    mode &= 15 ;
    outmode_A &= 3 ;
    outmode_B &= 3 ;
    capture_mode &= 3 ;

    byte clock_mode = 0 ; // 0 means no clocking
    switch (prescale){
        case 1: clock_mode = 1 ; break ;
        case 8: clock_mode = 2 ; break ;
        case 64: clock_mode = 3 ; break ;
        case 256: clock_mode = 4 ; break ;
        case 1024: clock_mode = 5 ; break ;
        default:
            if (prescale < 0)
                clock_mode = 7 ; // external clock
    }
    TCCR1A = (outmode_A << 6) | (outmode_B << 4) | (mode & 3) ;
    TCCR1B = (capture_mode << 6) | ((mode & 0xC) << 1) | clock_mode ;
}
```

Écrire l'appel à ce sous-programme pour réaliser une commutation des afficheurs à 30 Hz

Réponse :

COMPARAISON ATmega328



- 00 : rien sur la sortie OC1A
- 01 : Basculement de OC1A à chaque comparaison
- 10 : RAZ OC1A
- 11 : mise à 1 de OC1A

- 0000 : TIMER 1 en comptage avec TOV1 lors du débordement
- 0001 : TIMER1 en mode 8 bits
- 0100 : RAZ du timer 1 quand comparaison avec OCR1A (CTC)
- 1100 : RAZ du timer 1 quand comparaison avec ICR1