

Test écrit

Nom.....

Lisez bien les questions ! Rendez les réponses sur cette feuille (recto et verso) et sur une feuille annexe si nécessaire. Le test est individuel, documentation personnelle autorisée, ordinateurs exclus. Durée prévue : 15h20 à 17h00 **Bon test !**

1) Vrai-Faux (cochez vrai ou faux)

- a) Un MSP430G consomme environ 5 mA sous 3 V en fonctionnement normal. [] vrai [] faux
- b) Avec un microcontrôleur, on peut choisir le rôle d'une patte (entrée ou sortie) au début du programme et le changer à n'importe quel moment [] vrai [] faux
- c) Le signal produit sur P1.4 a une fréquence de 6 kHz lorsque le programme suivant s'exécute sur un MSP430G cadencé à 8 MHz: `while (1) {P1OUT^=(1<<4);} [] vrai [] faux`
- d) Les instructions suivantes permettent de mettre la patte P2.5 en sortie avec résistance de tirage pull-down : `P2DIR &=~(1<<5); P2REN |= (1<<5); P2OUT &=~(1<<5); [] vrai [] faux`

2) Complétez le programme

Complétez le programme ci-dessous à la ligne indiquée pour qu'il réagisse aux flancs montant et descendants du signal sur P1.4.

```
int main() {
    P1IE |= (1<<4); // P1.4 interrupt enabled
    P1IES |= (1<<4); // P1.4 Hi/lo edge
    P1IFG &=~(1<<4); // P1.4 flag cleared
    _BIS_SR(GIE); // general interrupt enable
    while (1) {
        }
    }
#pragma vector=PORT1_VECTOR // Port 1 interrupt service routine
__interrupt void Port_1(void) {
    P1IFG &= ~0x10; // P1.4 flag cleared
    .....
}
```

3) Serrure codée

Une serrure codée est munie de trois touches A, B et C. Elle doit s'ouvrir uniquement lorsqu'on exécute la séquence suivante : presser C, relâcher C, presser A, presser B, relâcher B, relâcher A.

Ecrivez un programme qui réalise cette fonctionnalité, en utilisant P1.0 pour la sortie et P1.3 à P1.5 pour les entrées.

Suggestion : vous pouvez décrire le problème par un graphe d'état et programmer la machine d'état correspondante.

4) Signal triphasé par PWM

On souhaite produire trois signaux sinusoïdaux déphasés l'un par rapport à l'autre de 120 degrés avec un MSP430G, à une fréquence de 1 Hz.

Ecrivez un programme pour produire ces trois signaux PWM sur les pattes P1.3, P1.4 et P1.5.

Vous pouvez utiliser une table avec les sinus entre 0 et 89° :

`unsigned char Sinus [90] = {... // valeur de 256 * sinus (x), avec x en degré`

