

Exercices pour la préparation de l'examen final

1. Les interruptions sont masquées par défaut durant l'exécution d'un ISR (sous-programme de service d'interruption).

- a) Vrai
- b) Faux

Réponse a) Vrai



2. Qu'est ce qui indique à l'UCT HCS12 où est stocké un ISR (sous-programme de service d'interruption) en mémoire.

- a) La broche d'entrée IRQ
- b) Le vecteur Reset
- c) Le compteur de programme (PC)
- d) Le vecteur d'interruption

Réponse d) Le vecteur d'interruption

3. Quelle est la différence entre les entrées IRQ et XIRQ?

Réponse :

IRQ est une interruption masquable tandis XIRQ est un non-masquable.

4. Sélectionnez l'instruction qui met à zéro le bit drapeau C7F (le bit le plus significatif du registre d'état TFLG1 de la minuterie) :

- a) bclr TFLG1,\$80
- b) movb #127, TFLG1
- c) bclr TFLG1,127
- d) bset TFLG1,\$80

Réponse a) bclr TFLG1,\$80

Partie 1 Réponses courtes

5. Considérons un signal, contenant des fréquences de 10 kHz à 23 kHz (10000 à 23000 Hz), appliqué à une broche du port ATD du HCS12. En supposant que ce signal est converti en une valeur binaire de 8 bits à l'aide du mode SCAN et que 8 cycles sont utilisés pour la 2e phase du temps d'échantillonnage, déterminez et sélectionnez une fréquence d'horloge ATD disponible dans le module ATD du HCS12 pour traiter ce signal. Supposez une horloge système de 24 MHz.

Réponse :

Fréquence d'échantillonnage = 46 kHz, temps d'échantillonnage = temps de conversion= 21,7 µsec. Nombre de cycles de conversion/conversion = $2 + 8 + 8 = 18$ cycles d'horloge ATD/conversion. Cycle d'horloge ATD = 22,7 µsec/conversion divisé par 18 cycles d'horloge ATD/conversion = 1,21 µsec/cycles d'horloge ATD. Fréquence ATD minimale = 829 kHz. Utilisez l'horloge de 860 kHz disponible dans le module ATD.

5. Quelle est la plus grande valeur qui peut être atteinte par le registre TCNT du timer?

Réponse :

$2^{16}-1$ or 65535

6. Est-il possible de lire le contenu du registre de données de transmission SCI? Expliquez votre réponse?

Réponse:

Non, le registre de données d'émission et le registre de données de réception occupent la même adresse, donc lors de la lecture de l'adresse, c'est le contenu du registre de données de réception qui est lu, pas le registre de données d'émission.

Question) Application: Le système d'alarme

Cette partie de l'examen porte sur l'émulation d'un système d'alarme à l'aide de la carte Dragon-12. La banque de 8 interrupteurs DIP est utilisée pour émuler les contacts de portes et de fenêtres. Des LEDs et un interrupteur DIP supplémentaire sont utilisés pour émuler le panneau d'alarme. Enfin, un haut-parleur est utilisé pour émuler la sirène d'alarme. Le développement du système d'alarme est complété par trois questions distinctes qui reflètent une conception modulaire.

Application: Le système d'alarme

Question: Module de surveillance des portes et fenêtres

La figure 1 montre comment 8 interrupteurs DIP, utilisés pour émuler les contacts de porte et de fenêtre, sont connectés au microcontrôleur HCS12 Port H de la carte Dragon-12. La figure montre aussi comment les 8 LEDs sont connectés au Port B.

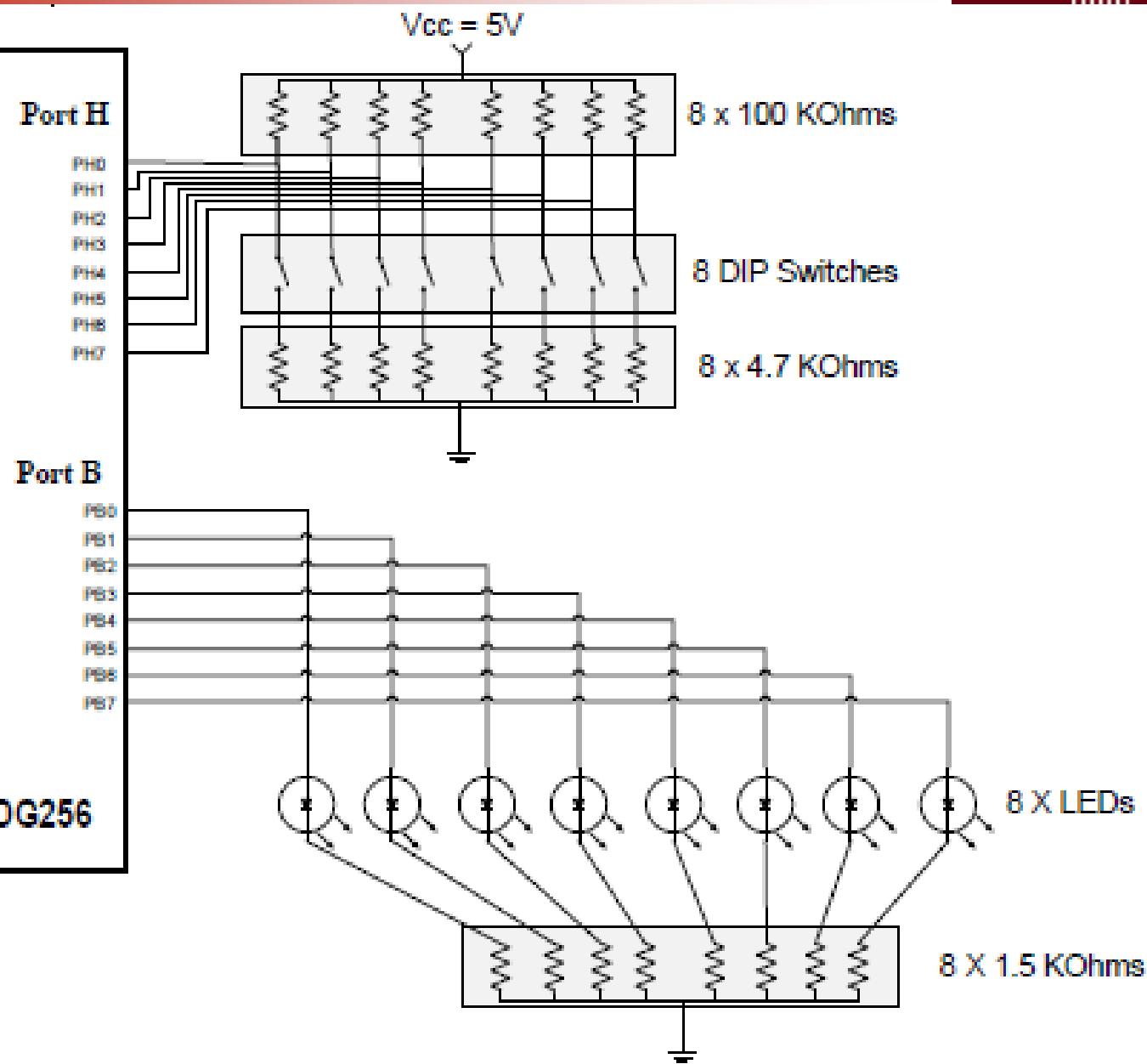


Figure 1.

Application: Le système d'alarme

Module de surveillance des portes et fenêtres

- A) Expliquez brièvement l'utilité des résistances du circuit utilisé pour connecter les interrupteurs DIP.

Réponse : Règle la tension sur la broche à 0 lorsque l'interrupteur est fermé et à 5V lorsqu'il est ouvert. Une résistance de 100 Kohms agit comme une résistance pullup.

- A) Expliquez brièvement l'utilité des résistances dans le circuit utilisé pour connecter les DELs.

Réponse : Limite le courant lorsque 5 V est appliqué à la DEL.

Module de surveillance des portes et fenêtres

C) Conception du logiciel : Module de surveillance de porte/fenêtre.

L'objectif de ce module est de

- Surveiller les portes et les fenêtres (c'est-à-dire les interrupteurs DIP) à l'aide d'un ISR déclenché par le canal 0 de la minuterie toutes les 100 ms.
- Refléter l'état des contacts des portes/fenêtres sur les LEDs ; quand un interrupteur est ouvert, une LED correspondante est allumée. Supposons que la LED connectée à PB0 reflète l'état du commutateur connecté à PH0, la LED connectée à PB1 reflète l'état du commutateur connecté à PH1, etc.
- Définir une variable globale, alarmStatus, à TRUE (valeur 1) lorsqu'un des interrupteurs est ouvert (c'est-à-dire qu'une porte ou une fenêtre est ouverte) et à FALSE (valeur 0) lorsque tous les interrupteurs sont fermés.

C)

- a) Développez une fonction d'initialisation, `initDoorWindowMonitor()`, qui configure le PORT B, le PORT H et le canal 0 de la minuterie (notez que pour le canal 0 de la minuterie, configurez uniquement le canal, et non les fonctions générales de la minuterie; vous pouvez supposer une fréquence d'horloge de la minuterie de 187,5 KHz, c'est-à-dire un cycle d'horloge de 5 1/3 μ sec).
- b) Développez ISR, `doorWindowMonitorISR()`.

a) Réponse

```
#define ONE_HUNDRED_MS 18750 // Number of 5 1/3 micro-sec ticks for 100 ms
void initDoorWindowMonitor()
{
    // Setup Port B
    DDRB = 0xFF; // Setup all pins as output
    PUCR &= 0b11111101; // Deactivate pull-up registers
    // Setup Port H
    DDRH = 0x00; // Setup as input pins – no pullups for port H
    PORTH = PORTB; // bit=0 when switches are closed
    // Setup Timer Channel 0
    // Assume 5 1/3 micro-second ticks
    TIOS |= 0b00000001; // Set TC0 as output-compare
    TIE |= 0b00000001; // Enable TC0 interrupts
    TC0 = TCNT + ONE_HUNDRED_MS; // For first interrupt in 100 ms
}
```

b) Réponse :

```
#define TRUE 1
#define FALSE 0
int alarmStatus;
void interrupt VectorNumber_Vtimch0 doorWindowMonitorISR()
{
    PORTH = PORTB; // reflect status of switches on LEDs
    if(PORTB != 0x0) // Are all Switches closed?
        alarmStatus = FALSE
    else
        alarmStatus = TRUE;

    TC0 = TC0 + ONE_HUNDRED_MS;
}
```