

Merci d'indiquer sur chaque page votre nom et votre prénom

Questions relatives au cahier des charges : « Conception d'un poste de pesée et de mesure de niveau sur une ligne de production »

Remarque : Les questions peuvent être traitées indépendamment les unes par rapport aux autres.

Précision sur la configuration du 8051F020 : Pour traiter l'ensemble des questions, on considérera que l'horloge SYSCLK fonctionne à la fréquence d'un quartz externe à 22,1184 MHz.

Précisions sur l'évaluation :

Sur l'ensemble des questions, 26 points sont attribués. Votre total de points obtenus sera directement considéré comme une note sur 20.

Certains exercices peuvent être traités de différentes manières. Toutefois l'évaluation sera maximale pour des solutions qui sollicitent le moins possible le processeur et utilisent au mieux les périphériques. Les solutions à base de temporisation et qui monopolisent le processeur sont à éviter.

1. Question – Configuration globale du microcontrôleur 8051F020 – 1 point. 2,5/4

Citez quelles sont les opérations de configuration globale du microcontrôleur à faire avant d'appeler les fonctions de configuration des divers périphériques.

- Configuration de l'horloge ✓ *watchdog*
- Configuration des interruptions timer
- Configuration du xB1E ✓
- Configuration du GPIO ✓
- Configuration des interruptions externes
- gestion des sources de réinitialisation
- Activation globale des interruptions si nécessaire

2. Question – Identification des périphériques – 1,5 points. 5/6

En première approche et compte tenu du cahier des charges, identifiez les périphériques que vous devrez forcément mettre en œuvre, justifiez leur emploi.

- Le GPIO n'est pas forcément considérable comme un périphérique, mais est nécessaire pour différents signaux
- L'UART est nécessaire pour la transmission des mesures au poste de contrôle (L'UART0 est ici choisi)
- Une interruption externe devra aussi être paramétrée avec le cronbar. ceci afin de mesurer l'écho ultrasonique.
- Un ADC (ADCO) devra être utilisé pour effectuer des mesures sur la balance.

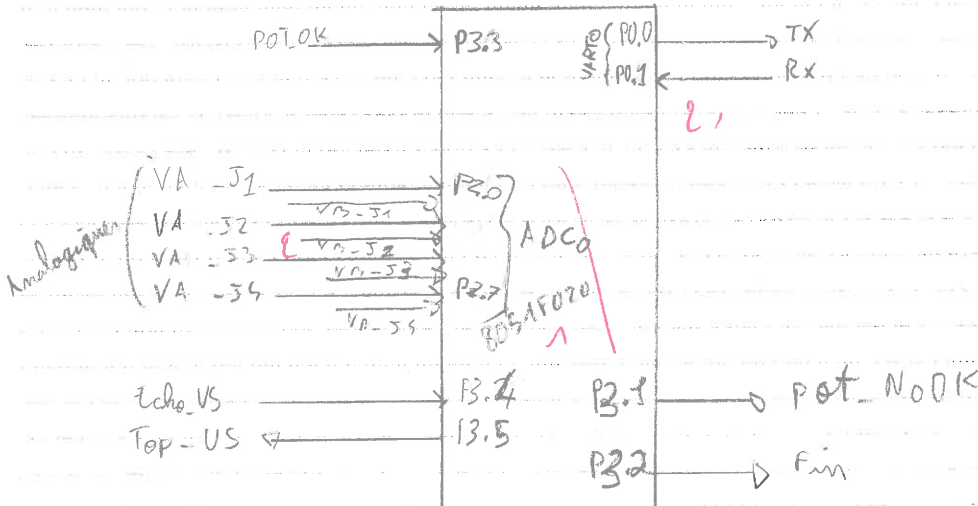
Timer?

3. Question – Identification des entrées-sorties du système – 1,5 points.

1/6

Dessinez sur un schéma tous les signaux échangés entre le 8051F020 et les différents éléments du dispositif. Indiquez le sens des signaux (entrée ?, sortie ?), leur nature (numérique ? analogique ?). Identifiez, côté 8051F020, les GPIOs, et les ports utilisés pour véhiculer les signaux de périphériques.

→ Sauf mention contraire, les signaux sont numériques



4. Question – Datation des mesures – 2 points

4,5/8

Proposez une solution logicielle pour la gestion des informations Heures, Minutes et Secondes.

Précisez : Quel(s) périphérique(s) ? Quelle(s) interruption(s) ? Quel(s) mode(s) ? Quel(s) registre(s), Quelle(s) broches(s) ?

Nous ne vous demandons pas le détailler la configuration des registres, mais de montrer que votre solution est faisable, compte tenu de l'ensemble des contraintes.

L'utilisation d'un Timer en mode auto-rechargement semble être la solution indiquée. N'importe quelle période inférieure à 1 seconde devrait suffire ; de préférence une fréquence entière.

Par exemple, le Timer 2 peut être utilisé, en mode 16 bit avec auto-rechargement.

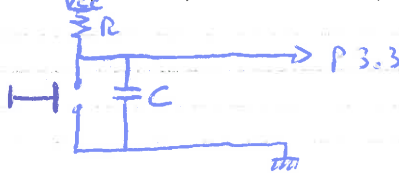
Il se configure à l'aide de CKCON, T2CON, RCAP2L/H.

Il s'agit du vecteur d'interruption n°5.

5. Question – Connexion du signal Pot_OK sur le 8051F020– 1,5 points. 6/6

Pour connecter le signal **Pot_OK** sur le 8051F020, on utilisera la broche GPIO P3.3. On veut que lors du passage du pot, le signal vu par le 8051F020 soit au niveau bas.

- Dessinez le schéma de branchement de l'interrupteur **Pot_OK** sur le microcontrôleur.
- Indiquez les opérations à faire (quels registres ?) pour configurer la broche P3.3 du microcontrôleur. Expliquez.



$R = 20\text{ k}\Omega$ ou similaire (pull-up).

Une constante de temps $\tau = RC$ ^{de 1 ms} devrait suffire pour l'anti-rebond matériel. Elle peut être augmentée jusqu'à 10 ms.

On veut configurer P3.3 en tant qu'entrée : Il faut donc mettre

P3MOUT.3 au niveau logique 0 et P3.3 au niveau logique 1
Note : P3MOUT n'est pas adressable bit à bit et requiert donc l'envoi d'un masque. $P3MOUT \&= \sim(1 \ll 3);$ devrait suffire

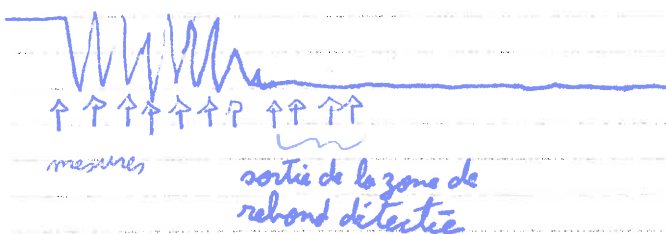
Note 2 : Une pull-up de 100k est connectée par défaut à ce port.

Note 3 : Il n'y a pas besoin du XBAR pour l'instant (entrées).

6. Question – Gestion anti-rebond de Pot_OK– 1,5 points. 6/6

Le capteur **Pot_OK** étant un simple interrupteur, on aura donc des phénomènes de rebond durant les changements d'état de l'interrupteur. Proposez une solution logique de traitement de ces rebonds. (On ne vous demande pas les détails de la réalisation, mais simplement le principe).

Une machine d'état comme celle vue lors du TP peut permettre de corriger l'effet de rebond. Le principe est de vérifier l'état de l'interrupteur sur plusieurs mesures effectuées à des instants différents, de manière à confirmer l'établissement du contact et la sortie de la zone de rebond.



En pratique, dans ce cas, une faible fréquence de consultation suffit, et deux lectures successives identiques séparées de 5 ms permettraient de confirmer le changement d'état.

7. Question – Opération de Pesée – 4 points

Proposez et justifiez une solution de gestion de la pesée en donnant le squelette de 2 fonctions. On cherchera à obtenir la meilleure résolution possible.

- Fonction 1 : `Config_Pesee()` - Fonction de configuration de la pesée, appelée une seule fois lors des initialisations du microcontrôleur.
- Fonction 2 : `Execution_Pesee()` - Fonction chargée d'effectuer la pesée sur les 4 jauges. Cette fonction renvoie un poids en gramme (somme des poids mesurés par chaque jauge).

Précisez : Quel(s) périphérique(s) ? Quelle(s) interruption(s) ? Quel(s) mode(s) ? Quel(s) registre(s), Quelle(s) broches(s) ?

- Compte tenu du dispositif employé, quelle est la résolution de la mesure sur chaque jauge de contrainte ?
- Donnez un ordre de grandeur de la durée d'exécution de la fonction : `Execution_Pesee()`, à défaut de valeur précise, expliquez comment vous la calculeriez.

void Config_Pesee(void) {

- configure la référence de tension (2,4V)
- Active un multiplicateur par 4 (ADLOCF)
- Activation sur AD0BUSY (AD0CM de AD0CN)
- AD0EN à 1 (AD0CN)
- AD0TM à 0 pour tracking continu.
- AD0LJST en fonction des codes.
- AD0SC réglé pour une fréquence de 2,5 MHz (ADLOCF)
- AIN 0-7 configurés en différentiels (AMKOCF)

long int Execution_Pesee() {

↳ 4 capteurs pouvant aller jusqu'à 20 kg, soit 80 000g max > 65535

→ choix du capteur (boucle sur les capteurs) avec AMXOSL (voir tableau p. 48 de la datasheet).

↳ Mise à 1 du bit de conversion AD0BUSY

↳ Attente (avec Timeout) de la fin de la conversion (par mutation de AD0INT (préalablement mis à 0 au début de la fonction)).

↳ ajout dans une variable locale initialisée à 0 (long int ou similaire) la valeur mesurée, avec conversion.

↳ boucle sur les capteurs suivants

→ retour de la somme.

suite au verso.

Résolution de 12 bits avec un multiplicateur par 4:

$0 \rightarrow 4095$

$0V \rightarrow 0,6V$

soit une résolution de environ $147 \mu V$

En supposant l'évolution du capteur linéaire, cela correspond à une résolution de $5,88 g$ (soit environ 6 grammes).

(Les imprécisions dues aux arrondis n'ont pas été prises en compte afin de gagner du temps).

La vitesse d'exécution maximale est limitée par le temps de conversion (16 coups de clock SAR à $2,5 MHz$, soit $6,4 \mu s$ par capteur, ou $25,6 \mu s$ pour l'ensemble). À cela vient s'ajouter le coût des opérations de conversion en grammes, et des additions sur un type de données > 8 bits, qui peut être important. (Si l'on prend comme estimation 100 coups d'horloge à $22 MHz$, on obtient 5 μs).

Le temps total de conversion est donc de l'ordre de la trentaine de μs .

8. Question – Mesure de niveau – 4 points 9/16

Proposez et justifiez une solution de gestion de la mesure de niveau en donnant le squelette de 2 fonctions. On cherchera à obtenir la meilleure résolution possible (Résolution de l'ordre du millimètre) dans la mesure du temps d'aller-retour.

- Fonction 1 : `Config_MesureNiveau()` - Fonction de configuration de la mesure de niveau, appelée une seule fois lors des initialisations du microcontrôleur.
- Fonction 2 : `Execution_MesureNiveau()` - Fonction de chargée d'effectuer la mesure de niveau. Cette fonction renvoie une mesure de distance capteur US - Surface de liquide en mm.

Précisez : Quel(s) périphérique(s) ? Quelle(s) interruption(s) ? Quel(s) mode(s) ? Quel(s) registre(s), Quelle(s) broches(s) ? Expliquez votre raisonnement.

Indice : un timer peut fonctionner autrement qu'en mode auto-rechargement....

`void Config_MesureNiveau()`

→ ~~activation~~ d'un Timer en mode « counter », par exemple configuration

Timer 3, à la fréquence maximale.

→ appel de la configuration d'INT 1 sur front descendant.

}

`long int Execution_MesureNiveau()`

{ → mise à 0 d'une variable globale "x" (long int) et du compteur

→ sortie du signal `TOP-US`

→ activation du compteur

→ désactivation du signal `TOP-US`

→ attente sur la variable globale "x"

→ conversion de la variable globale en mm

retour de la nouvelle variable

}

`void ISR_INT_interrupt2(void) {`

désactivation compteur

copie du registre du Timer dans x

}

`void ISR_Timer3_interrupt79 {`

Mise de la variable globale à une valeur "Timerout"

}



Nom :
Préno

CPE Lyon – 4ETI

Ver : 09/11/2015 15:25

Devoir du module « Bases
des systèmes embarqués »
13/11/2015

9. Question – Transmission du résultat de mesure– 2,5 points

Les résultats de mesure seront transmis en utilisant le périphérique UART1. Coder, en les commentant les 3 fonctions suivantes :

```
void Config_Clock_UART1(void) ; // Configuration de l'horloge pilotant l'UART1  
void Config_UART1(void) ; // Configuration de l'UART1  
char Puchar(char c) ; // Envoi d'un caractère sur l'UART1 - La fonction retourne le caractère transmis. Pas de gestion de Timeout requise
```

void Config - Clock - UART1 (void) ;

résultat du flag de transmission

10. Question – Signalement Pot non conforme – 1 points

Ce signal sera connecté sur la broche P3.4. Quelle configuration devez-vous adopter sur cette broche et pourquoi ?

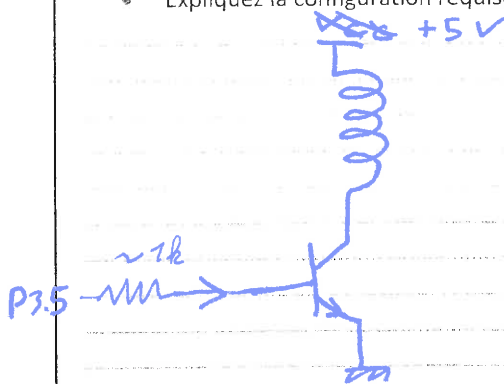
Une configuration push-pull avec activation à l'état bas est préférable, car la bobine consomme 66 mA. En réalité, un transistor est préférable pour la piloter confondu avec P3.5

Il faut configurer cette sortie en drain ouvert afin d'avoir par défaut un état haute impédance (ou haut, avec la pull-up). On réalise ainsi un π en câblage π , où l'une des sorties passant à 0 entraîne le passage à 0 de toute la ligne

11. Question – Pilotage de l'éjection – 1.5 points

C'est la broche P3.5 qui pilotera le relais chargé de commander le poussoir d'éjection. Pour fermer le circuit alimentant le poussoir, il suffit d'assurer une tension d'au moins 3V aux bornes B1/B2 de la bobine (voir schéma). Le relais sera activé (contact fermé, bobine alimentée) sur un niveau haut sur P3.5.

- Proposez un schéma électrique de câblage entre P3.5 et le relais.
- Expliquez la configuration requise sur P3.5.



P3.5 doit être configurée en push-pull



Nom : _____

Prénom : _____

CPE Lyon – 4ETI

Ver : 09/11/2015 15:25

**Devoir du module « Bases
des systèmes embarqués »
13/11/2015**

12. Question – Conception complète– 4 points.

Proposez un squelette de solution pour ce cahier des charges. Y apparaitront essentiellement des noms de fonctions. Chaque fonction sera explicitée si nécessaire.

Précisez les fonctions exécutées dans le « main » et les fonctions exécutées dans les routines d'interruption. Précisez la nature des événements produisant les interruptions.

Un schéma d'illustration sera le bienvenu.