

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 SYSCCLK 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.

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

Avant de configurer les périphériques du système, il est nécessaire de configurer l'horloge du système, ainsi que la crossbar si elle est nécessaire.

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

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

Compte tenu du cahier des charges, il est nécessaire de mettre en œuvre la conversion analogique/numérique, l'UART et des timers.

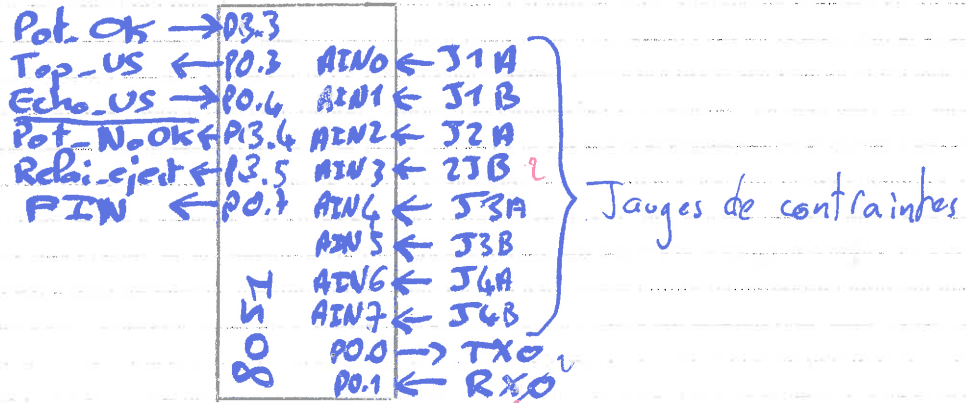
La conversion A/N servira à lire le résultat de la pesée du pot.

L'UART servira à envoyer des informations au poste de commande via une liaison série asynchrone.

Les timers serviront d'horloge pour la liaison série et de source d'interruption pour la servation.

3. Question – Identification des entrées-sorties du système – 1,5 points. 5/6

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.



Tous les ports sauf AIN* sont numériques.

4. Question – Datation des mesures – 2 points 6/8

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) ?

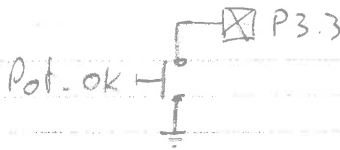
Nous ne vous demandons pas le détailler la configuration des registres, mais de montrer que votre solution est faisable, compte tenu de l'ensemble des contraintes.

Afin de gérer l'horloge nous nous appuierons sur le timer 2, ce timer 16bits nous permet de générer une interruption à chaque débordement. De plus ce timer gère l'auto-reload, configuré pour compter à une fréquence de $\text{SYSCLK}/12$, il nous permet de générer une interruption toutes les 5 μs . Il suffit ensuite de mettre des variables à jour à chaque interruption pour savoir le nombre d'Heures/Minutes/Secondes écoulées depuis l'allumage du système.

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.



Pour configurer ce port en entrée il est nécessaire de le configurer en drain-ouvert et à un niveau HAUT.
Pour que le niveau soit HAUT en l'absence de pot, il est nécessaire d'utiliser la résistance de pull-up. Oui!!

Configuration en drain-ouvert : $P3MDOUT = xxxx0xxx$
Mise à '1' de P3.3 : $P3 = xxxx1xxx$
Activation des pull-ups : $XBR2 = 0xxx xxxx$

x : Ce bit ne doit pas être changé.

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

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 logicielle de traitement de ces rebonds. (On ne vous demande pas les détails de la réalisation, mais simplement le principe).

Afin de traiter les rebonds, il faut vérifier que l'état du pin ne change pas durant 5ms (temps maximal des rebonds).

Il est possible pour cela d'utiliser le timer 2 (configuré pour interrompre le processus principal toutes les 5ms) et de vérifier que l'état actuel est le même que l'état précédent. Dans ces conditions, nous sommes bien en régime établi. L'impulsion du passage d'un pot étant supérieure ou égale à 5ms, il n'est pas possible de la rater à cause d'un retard de 5ms.

7. Question – Opération de Pesée– 4 points

12/16

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.

`Config_Pesee()` {
`REFO CN = --- 0x10` : référence de tension
`ANXOCF = ---- 1111` : entrées par paires +/-
`ADCO CN = 10--00-0` : activation ADC
 + tracking continu, démarrage de conversion via `ADOBUSY` et
 registres justifiés à gauche.
`ADCO CF = 1111010` : gain de 4

`Execution_Pesee()` : `ANUXO = 00000000` : sélection `AIN0-AIN1`
`ADOBUSY = 1` : démarrage de la conversion
 attente de la fin de la conversion
 résultat total += `ADC0L + (ADC0H << 4)`
 puis on passe `ANUXO` à `0x02` et on recommence,
 on réitère ensuite avec `ANUXO = 0x04` et on finit
 avec `ANUXO = 0x0C`.

On obtient un résultat total qu'il faut convertir de
 multiplier par 5,85 pour obtenir le poids en
 grammes.

On a donc une résolution de 5,85g
 par jauge grâce à l'ADC 12 bits.

8. Question – Mesure de niveau– 4 points

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....

Config_MesureNiveau(): On configure le timer 0 en compteur 16 bits avec une fréquence de $SYSCLOCK/12$.

Execution_MesureNiveau(): On active Top-US qu'on désactive ensuite (et sensible au front). A cet instant, on active le compteur. On attend ensuite un niveau bas sur Echo-US (il est possible de gérer un timeout). A son passage à l'état bas, on stoppe le compteur. Le compteur effectuant des pas de 567ns, on en déduit le temps mis au signal pour parcourir la distance voulue. On divise ce temps par 2 (aller-retour), puis on le multiplie par la vitesse du son pour obtenir la distance.

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
```

2

```
Config_Clock_UART1() {
    M0D8 = ~(0x00); // Mode 2 : timer 8bits
    M0D1 = 0x20; // avec auto-reload
    TL1 = TH1 = -(3600/SYSCLOCK/16);
    TR1 = 1; // CLOCK
}
```

1,5

```
Config_UART1() {
    SCON0 = 0x72;
    UARTGEN = 1; // XBAR config
    XBAR = 1; // XBAR enable
}
```

1

```
char Puchar(char c) {
    while (TIO != 1);
    SBUF0 = c;
    return c;
}
```

10. Question – Signalement Pot non conforme – 1 points

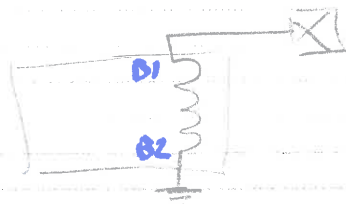
Ce signal sera connecté sur la broche P3.4. Quelle configuration devez-vous adopter sur cette broche et pourquoi ?

Ce pin doit être configuré en drain ouvert afin de réaliser une fonction câblée. Quel que soit le système qui passe le pin à 0, l'état général sera 0 et les sorties des autres systèmes ne seront pas endommagées.

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 configuré en push-pull.
les courants de sorties sont de
100 mA max. Ici, le courant est
de 73 mA, donc inférieur à la
limite maximale.*



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 évènements produisant les interruptions.

Un schéma d'illustration sera le bienvenu.