

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 considèrera 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. 4/4

Citez quelles sont les opérations de configuration globale du microcontrôleur à faire avant d'appeler les fonctions de configuration des divers périphériques. 1/1

Configuration des sources de reset :

- du watchdog
- des mémoires
- de l'oscillateur
- gestion puissance
- du cristal et des GPIO

puis des périphériques internes.

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

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

- Watchdog non utilisé.
- Pour la RTC : timer2 en auto-reload.

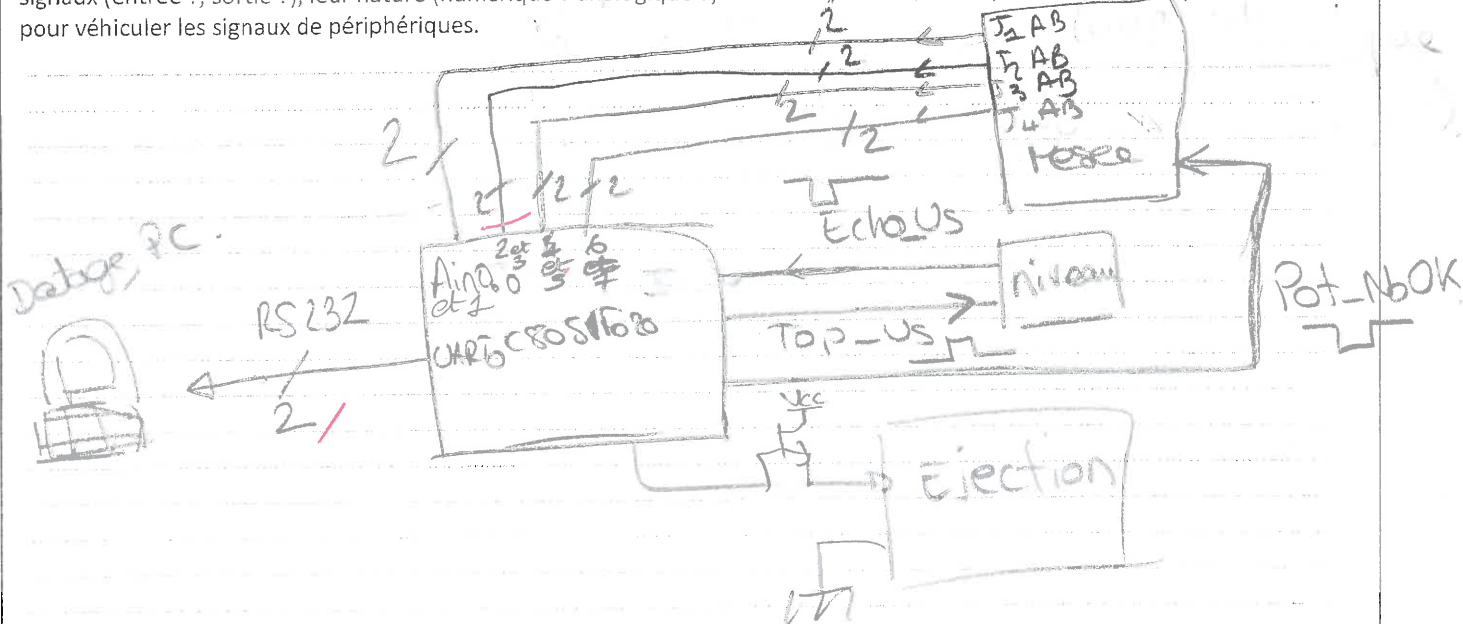
Base de tps à 5ms.

- Pousée : Sclatation sur Pot-OK (GPIO) et attente de 2 interruption timer 2 pour être sûr pas de rebond.
- Pesée : ADC 0 + timing grâce à RTC.
- Mesure du niveau de liquide : { Int ext
Timer0 en 16bit pour connaître distance.
- Liaison UART : Timer 1 en Baudrate generator (Mode 1)

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

4/6

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.



4. Question – Datation des mesures – 2 points

6,5/8

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.

Utilisation de Timer 2 en mode 16 bits avec auto-reload. Période 5ms.

⇒ interruption Timer 2 active, incrémentation de la base de temps de la RTC.

Nisc en oeuvre : • CKCON = TMR2 à SYSCLK

• TZCON : Timer2 utilise et n'est pas utilisé pour UART0.

• RCAP2L et RCAP2H : [RCAP2H, RCAP2L] = -9216.

↳ 5ms. $\frac{SYSCLK}{12}$

• T2L et T2H : [T2H, T2L] = FFFF (initia).

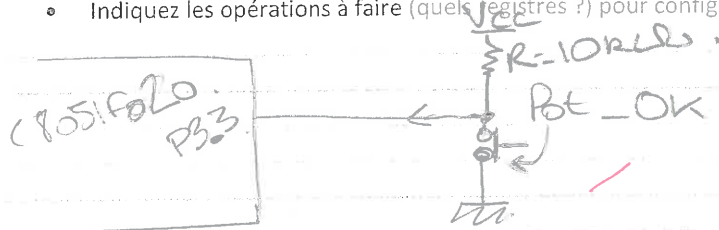
• IE : Activer int TMR2

• IP : priorité

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.



On veut P3.3 en entrée \Rightarrow $\left\{ \begin{array}{l} \text{open drain.} \\ \text{mettre le bit} \\ \text{à 1} \end{array} \right.$ (H.Z).

- enable XBAR (XBR2.6 à 1).
- dans P3NDOUT : bit 3 à 0 (P3.3 en OD)
- dans P3 : bit 3 à 1 (P3.3 en haute impédance)

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

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).

Anti-rebond logiciel : on lit P3.3 à chaque ISR Timer 2 (toutes les 5ms). On stocke cette valeur. Si la valeur actuelle est différente, le bouton a été poussé ou relâché. On met une autre variable à 1 ou 2, et on attend 1 ou 2 (5 ou 10ms) interruption TMR2 avant de considérer que le bouton a changé d'état pour de bon.

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()

{

AMXOCF = 0x0F; // différentiel.

AMXOSL = 0x00; // pour commencer (J1).

ADLOCF = 0xFA; // SAR CLK la plus lente possible

// gain de 4 → 500mV devient 2V
↳ meilleure précision

ADCON = 0x80; // activé

// continuous tracking

// lancement de conversion par écriture sur ADOBUSY.

// justification à droite

}

void execution_Pesee()

{

unsigned char i, j;

for(i=0; i<8; i+=2)

AMXOSL = i; // sélection J_i

for(j=0; j<100; ++j); // attente (peut-être facultative)

ADOBUSY = 1;

while(!ADOINT) { // s'il faudrait un compteur ici }

pes_result += (ADCON & 0x0F) << 8 + ADCOL;

pes_oler = 1;

Résolution: 47 grammes sur chaque

```
int Execution_Pesee()
```

```
{ int total = 0; ADC_word_12b = 0;
```

```
  unsigned char i = 0, j = 0;
```

```
  for( i = 0; i < 4; ++i)
```

```
  { ANXOSL = 2 * i; // sélection Ji
```

```
    for( j = 0; j < 200; ++j); // délai mise on place.
```

```
    ADO_BUSY = 1;
```

```
    while(!AD0INT) { // compteur ici
```

```
    }
```

```
    // conversion i terminée.
```

```
    ADC_word_12b = ((AD0H & 0x07) << 8) + AD0L;
```

```
    if(AD0H & (0x08))
```

```
    { ADC_word_12b *= -1;
```

```
    }
```

```
    total += 10000 + (296 * ADC_word_12b) / 100;
```

entre 0 et 2000g.
par pas de 39.

```
}
```

```
  return total;
```

```
}
```

8. Question – Mesure de niveau– 4 points 10/16

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_Mesure_Niveau()`

```
{
    // config INTO pour detection echo...
    TCON |= 0x01;
    EX0 = 1; PX0 = 1;
```

```
    // config TMR0 en 16 bits sur SysCLK
    TCON &= ~0x10; CKCON |= 0x08;
    TMOD |= 0x01; TMOD &= 0xF1;
    TLO = 0xFF;
    TH0 = 0xFF;
```

expliquez!

`int Execution_MesureNiveau()`

```
{
    int TMR; dist = 0;
```

```
    Top_US = 1; TopUS=0;
```

```
    TLO = 0x00;
    TH0 = 0x00;
```

`TCON |= 0x10;`
/ démarrage

```
    while( Echo_US );
```

```
    TCON &= ~0x10; //arrêt TMR0.
```

```
    TMR = TLO + ((int)TH0) << 8;
```

```
    dist = (TMR * 154) / 10000; => resolution 15µm.
```

```
    return dist;
```

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

2/10

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 Putchar(char c) ; // Envoi d'un caractère sur l'UART1 - La fonction retourne le caractère transmis. Pas de gestion de Timeout requise
```

void Config ———— UART1()

{

// config

2

~~SNOD1 = 1;~~
~~SSTAT1 = 1;~~
~~/SCON1 = 0x70;~~

}

void Config - Clock - UART1()

{

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 ?

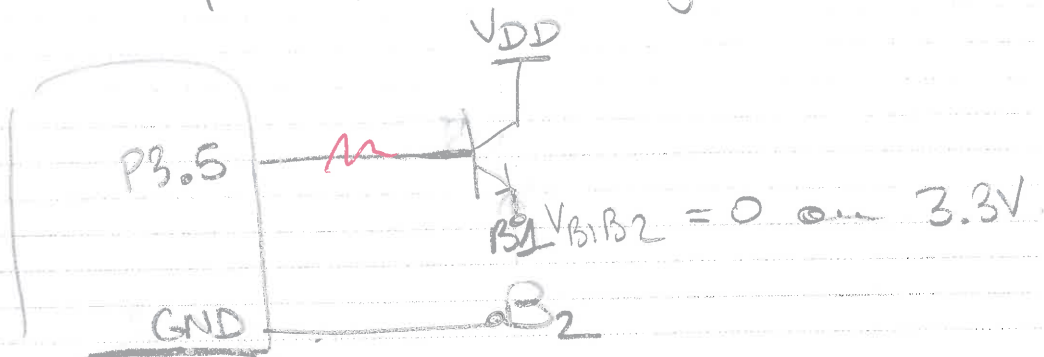
Open drain, en haute impédance.
Permet à tous les systèmes de pesée d'écrire 0.

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 en push-pull. Actif niveau haut.



Ces C8051f020 ne pourrait pas donner assez, de courant



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 apparaîtront 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.