EXAMEN
Systèmes Embarqués

NOM : PRENOM : N° PLACE:

ENSICAEN - année 2021-2022 – semestre 6
1ière année Génie Physique et Systèmes Embarqués
11/05/2022 - durée 2h30 – hugo descoubes

Documents autorisés

- Pages manuscrites

Conseils

Ce document tient lieu à la fois de sujet et de copie d'examen. Attention à ne pas recopier directement des phrases du polycopié. Utilisez vos propres mots afin de répondre aux questions. Si nécessaire, vous trouverez un espace de prise de notes en deuxième page de ce document. Les pondérations de chaque question étant données, il est conseillé de parcourir une première fois le sujet dans son ensemble avant de débuter l'épreuve. Une attention toute particulière sera apportée à la clarté, la qualité des schémas et la simplicité des solutions proposées. Ce sujet d'examen est découpé en trois parties indépendantes :

1. CONNAÎTRE – 6pts : Questions de culture générale pouvant traiter sur tout point abordé en séance de cours présentiel ou présent dans le support de travail. *Connaissances fondamentales et culture scientifique de l'ingénieur en Systèmes Embarqués*

2. COMPRENDRE – 10pts : Exercice de traduction d'un programme C vers un équivalent en assembleur PIC18. Niveau d'exigence proche de l'exercice réalisé en cours. *Comprendre le processus de traduction réalisé par les outils de compilation. Comprendre le processus d'exécution d'un programme.sur processeur*

3. INTERPRÉTER – 4pts : Analyse d'un programme réalisant une application simple sur une architecture processeur non découverte en enseignement. *Adaptabilité de l'ingénieur aux concepts étudiés sur de nouvelles technologies*


notes

1. CONNAÎTRE

Cet exercice peut être vu comme un échange d'ingénieur à ingénieur, la communication étant l'une des principales clés d'une collaboration saine en milieu professionnel. A travers la lecture de vos réponses, je dois comprendre que vous m'avez compris. Nous devons donc nous comprendre mutuellement. Utiliser obligatoirement un schéma exhaustif (utilisation d'un code couleur approprié) commenté afin d'illustrer vos propos. Bonne épreuve à vous ...

hugo

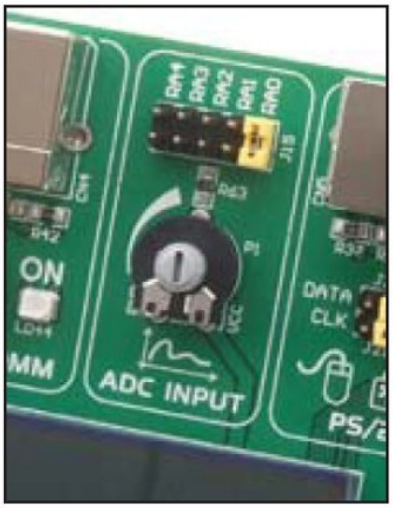
1. (*2pts*) Qu'est-ce qu'un Système Embarqué ? *Prendre le temps de représenter l'architecture puis d'expliquer le rôle et fonctionnement de chaque sous élément structurel voire d'illustrer avec des technologies existantes*

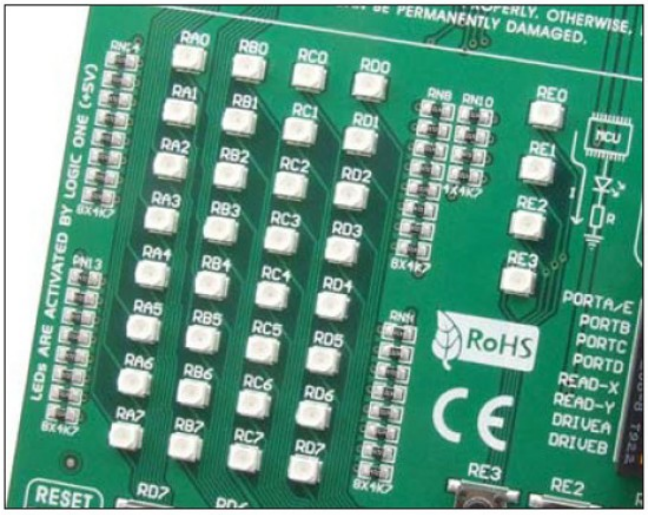
2. (*2pts*) Question Surprise !

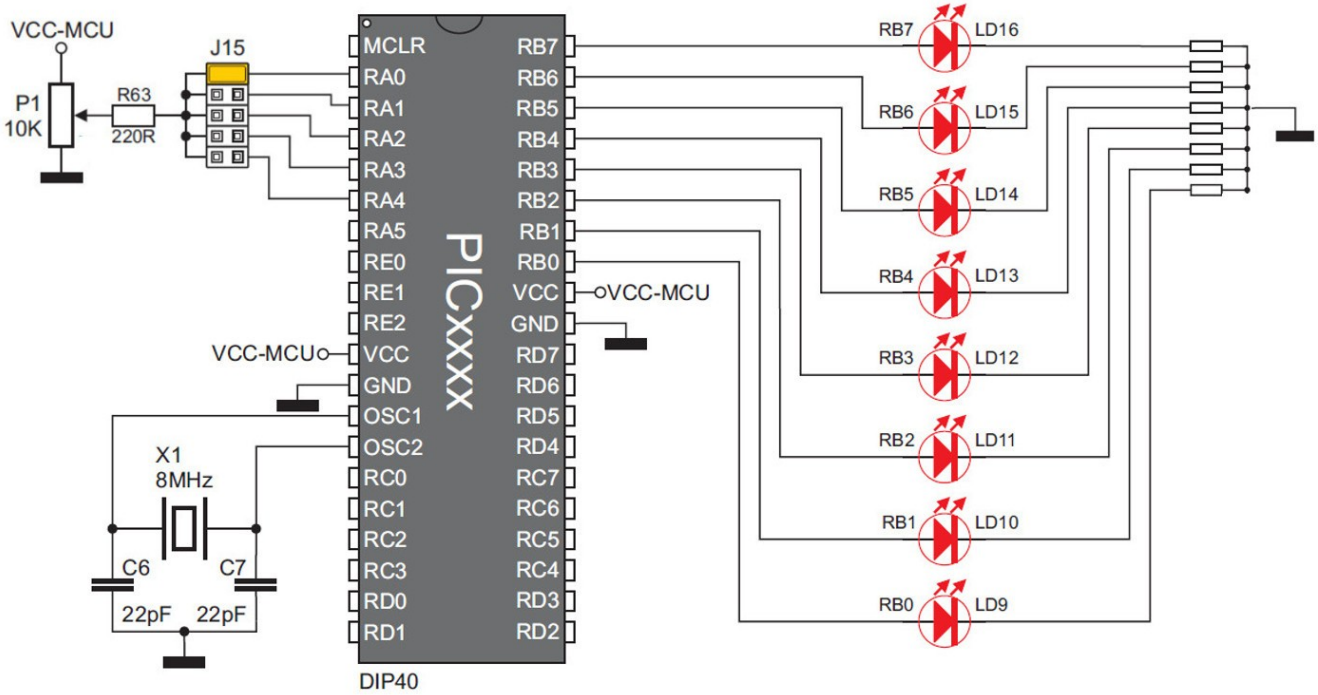
3. *(2pts)* Question Surprise !

2. COMPRENDRE

Les pages suivantes présentent un programme applicatif de test C exploitant une carte de développement PIC18 du marché (carte EASYPIC6 de Mikroelektronika). Observons sur les photos et le schéma électrique ci-dessous les interfaces matérielles utilisées par l'application. Sur cette carte de développement, le processeur est alimenté en 5V (VCC = 5V) :







Cet exercice consiste à traduire un programme développé en langage C vers un équivalent potentiel en langage assembleur pour PIC18. Plusieurs solutions sont possibles. En d'autres termes, vous allez devoir jouer le rôle du compilateur C suite à l'analyse du programme (*parsing*), celle-ci ayant été validée. Vous trouverez ci-dessous les consignes et limitations à respecter :

- Si un paramètre est passé à une fonction ou est retourné par une fonction, le passage du paramètre se fera dans les deux cas par le registre de travail W (Working register) présent dans le CPU. *Nous n'aborderons sur PIC18 pas les passages et récupérations par la pile de paramètres aux fonctions*

- Toutes les variables du programme sont des variables statiques globales. *Les allocations statiques facilitent l'implémentation du programme en assembleur et évitent la gestion de variables locales sur une pile ou stack par adressage relatif aux pointeurs SP (Stack Pointer) ou BP (Base Pointeur ou FP Frame Pointer)*

- Nous supposerons que toutes les variables de notre programme seront situées en ''Access Bank'' ou banque n°0. *Solution plus simple à implémenter sur PIC18 et permettant d'éviter la gestion de l'opérande* ''a'' ou ''access'' *présente dans grand nombre d'instructions assembleur sur PIC18*

Présentation du programme. Les questions suivent le code :

```
/*
* Application mystère !
*/

#include <p18f4550.h>


#pragma config   PLLDIV=2, CPUDIV=OSC1_PLL2, FOSC=HSPLL_HS
#pragma config   WDT=OFF, MCLRE=ON, BOR=OFF


#define        TMROHVALUE        0xFD
#define        TMROLVALUE        0x11


/* shared ressources */
unsigned char  convData, adcFlag = 0;
```

```
void __interrupt(high_priority)  isr_timer (void) {
        //  Acquittement flag d'interruption du timer
        INTCONbits.TMR0IF = 0;

        // pré-chargement de TMR0L + chargement automatique de TMR0H
        TMR0L = TMR0LVALUE;

        //  Démarrer une conversion analogique - numérique sur la broche RA0/AN0
        ADCON0bits.GO = 1;

        // Attendre la fin de conversion analogique – numérique sur 8bits
        while(ADCON0bits.GO);

        // Récupération de la donnée convertie dans le registre 8bits ADRESH
        // Valeur entière non signée sur 8bits
        convData = ADRESH;
        adcFlag = 1;
}



void hardware_config (void) {
       // Configuration des GPIO
       TRISB = 0x00;
       TRISAbits.TRISA0 = 1;     // RA0 en entrée pour l'ADC

       // Configuration Timer 0, période 20ms
        T0CON = 0x02 ;
        TMR0H = TMR0HVALUE;
        TMR0L = TMR0LVALUE;

        // Configuration ADC – RA0/AN0 configurée en entrée analogique
        // Plage analogique 0-5V    <---->    Plage numérique 0-255 (mode 8bits)
        ADCON0 = 0x01;
        ADCON1 = 0x0E;
        ADCON2 = 0x2C;

        //  Configuration des interruptions
        INTCON2bits.TMR0IP = 1;
        INTCONbits.TMR0IE = 1;
        RCONbits.IPEN = 1;
        INTCONbits.GIEH = 1;
}



char do_this (void) {
    return  convData
}



void  do_that ( char displayData) {
   LATB = displayData;
}
```

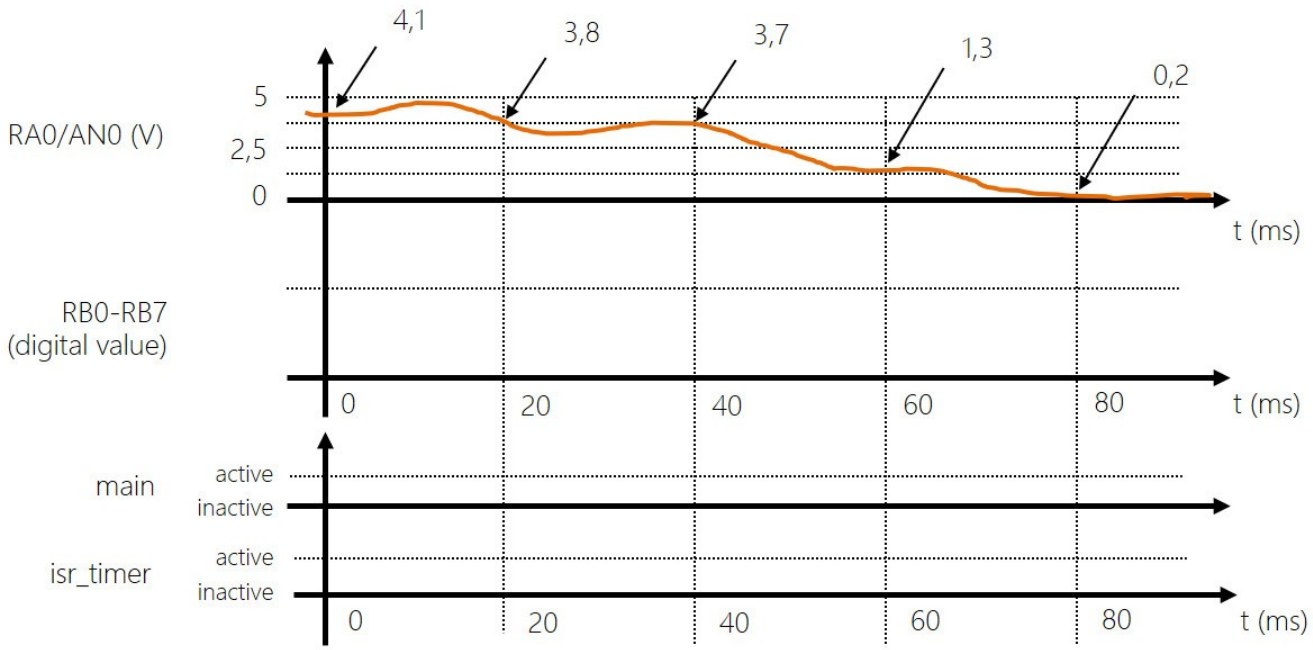
```
void main ( void ) {


   hardware_config();

    // application mystère
   while (1) {

      if ( adcFlag ) {
          adcFlag = 0;

          // traitement de la donnée convertie
         if ( convData < 42  ) {
                do_that( 42 );
          }  else {
                 LATB = do_this () ;
          }
      }
   }

}  // Fin main
```

Ne rien écrire ici !

Les deux questions qui suivent sont indépendantes de la suite de l'exercice de traduction. Il s'agit de questions d'analyse du programme à traduire.

4. (*0.5pt*) En analysant le programme, en dessous de quel seuil de tension analogique d'entrée sur la broche RA0 l'affichage des LED présentent sur le port B sera statique à la valeur 42_{10} ou $2A_{16}$ ou 00101010_2 ?*Expliquer et* ***justifier impérativement votre résultat par le calcul***

5. (*1.5pt*) Que fait ce programme ? *Expliquer le fonctionnement du programme et compléter* ***impérativement le chronogramme*** *ci-dessous.*



6. (*1pt*) Proposer une solution en assembleur PIC18 afin d'implémenter la fonction *hdwConfig*

```
void hardware_config (void) {
        // Configuration des GPIO
        TRISB = 0x00;
        TRISAbits.TRISA0 = 1;     // RA0 en entrée pour l'ADC

        // Configuration Timer 0, période 20ms
         T0CON = 0x02 ;
         TMR0H = TMROHVALUE;
         TMR0L = TMROLVALUE;

         // Configuration ADC – RA0/AN0 configurée en entrée analogique
         // Plage analogique 0-5V    <---->    Plage numérique 0-255 (mode 8bits)
         ADCON0 = 0x01;
         ADCON1 = 0x0E;
         ADCON2 = 0x2C;

         //  Configuration des interruptions
         INTCON2bits.TMR0IP = 1;
         INTCONbits.TMR0IE = 1;
         RCONbits.IPEN = 1;
         INTCONbits.GIEH = 1;
}
```

7. (*1pt*) Proposer une solution en assembleur PIC18 afin d'implémenter les fonctions *do_this* et *do_that*

```
char do_this (void) {
    return  convData
}



void  do_that ( char displayData) {
   LATB = displayData;
}
```

8. (*3pts*) Proposer une solution en assembleur PIC18 afin d'implémenter la fonction *main*

```
void main ( void ) {


   hardware_config();

    // application mystère
   while (1) {

      if ( adcFlag ) {
          adcFlag = 0;

          // traitement de la donnée convertie
         if ( convData < 42  ) {
                do_that( 42 );
          }  else {
                 LATB = do_this () ;
          }
      }
   }

}  // Fin main
```

9. (*3pts*) Proposer une solution en assembleur PIC18 afin d'implémenter l'ISR. *Attention, pour rappel les ISR ne sont pas des fonctions conventionnelles. La traduction est donc spéciale.*

```
void __interrupt(high_priority)  isr_timer (void) {
        //  Acquittement flag d'interruption du timer
        INTCONbits.TMR0IF = 0;

        // pré-chargement de TMR0L + chargement automatique de TMR0H
        TMR0L = TMR0LVALUE;

        //  Démarrer une conversion analogique - numérique sur la broche RA0/AN0
        ADCON0bits.GO = 1;

        // Attendre la fin de conversion analogique – numérique sur 8bits
        while( ADCON0bits.GO );

        // Récupération de la donnée convertie dans le registre 8bits ADRESH
        // Valeur entière non signée sur 8bits
        convData = ADRESH;
        adcFlag = 1;
}
```

3. INTERPRÉTER

L'exercice qui suit consiste à analyser un programme implémentant une application simple développée par un ingénieur d'une société tierce. Ce programme de test pour la carte XXXXXXX a été trouvé suite à une recherche sur internet. Le processeur utilisé est un MCU XXXXXXX proposé par la société XXXXXXX. Observons le programme à analyser :

(Bonus) En complément de l'épreuve, qu'avez-vous retenu de l'enseignement dans son ensemble et notamment du fonctionnement des MCU ? Avez-vous des recommandations à me faire pour améliorer l'enseignement ? *Merci par avance. Votre réponse me permettra une meilleure prise de recul sur l'enseignement. Merci également à vous pour les bons moments partagés en CM et en TP pour mon groupe ...*

hugo