

# TD1 - Codage des données numériques

## 1 Entiers relatifs

1. On considère le mot code  $A$  défini sur 8 bits par :  $A = 1111\ 1111$ . Quel est l'entier codé en considérant que  $A$  représente :
  - un entier naturel,  $\mathcal{N}(A)$
  - un entier relatif selon un codage signe et valeur absolue,  $\mathcal{S}(A)$
  - un entier relatif selon un codage en complément à 2,  $\mathcal{Z}(A)$
  - un entier relatif selon un codage par excédent,  $\mathcal{X}_{128}(A)$
2. Donner les représentations en signe et valeur absolue, codage par excédent ( ou binaire décalé de  $2^{n-1}$ ,  $n$  : nombre de bits) et en complément à 2 sur un octet des nombres 0, 1, -1, 127, -128.
3. Peut-on en déduire une règle simple de passage du binaire décalé au complément à 2 et inversement ?

## 2 Réels en virgule fixe

On veut coder des nombres réels, selon la technique du complément à 2, en notation virgule fixe sur 16 bits avec partie fractionnaire sur 7 bits. Le mot code  $A$  représente le réel noté  $\Phi_7(A)$ .

1. Représenter 12.75, 255.5, -255.5, 256, 1.007.
2. Conclure sur les intérêts et les limites du codage virgule fixe.
3. Que représente avec cette convention le mot code  $A = F8\ 10$  ?

## 3 Réels en virgule flottante (format IEEE P754)

Le format IEEE P754 est un standard pour la représentation des réels en virgule flottante. On rappelle que sur 32 bits (simple précision) un réel sera représenté par un bit de signe  $S$ , un exposant  $E$  sur 8 bits (codage par excédent), une mantisse de partie fractionnaire  $F$  sur 23 bits.

1. Représenter les nombres réels 1, 256, 0.125, -6.25, 5315.84375 en virgule flottante.
2. Inversement exprimer en décimal les nombres réels donnés en notation hexadécimale par :  $3F\ C0\ 00\ 00$  et  $C1\ 56\ 00\ 00$ .

## 4 Extension de signe

Soit  $a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0$  la représentation  $A$  d'un nombre en complément à 2 sur  $n$  bits.

1. donner la valeur numérique de  $\mathcal{Z}(A)$ .
2. montrer que le code représentant ce nombre sur  $n+1$  bits est  $a_{n-1}\ a_{n-1}\ a_{n-2}\ \dots\ a_1\ a_0$

## 5 Complément à la base, dépassement de capacité

1. Calculer en complément à 2 sur un octet la somme des nombres décimaux 96 et 45, 96 et -45, -96 et -45.  
Conclure sur le débordement.
2. Une unité arithmétique et logique (UAL) est munie de différents indicateurs mis à jour par les opérations arithmétiques et logiques,  
 $C$  : retenue finale,  $Z$  : zéro,  $N$  : signe,  $V$  : overflow.  
Comment sont définis ces indicateurs par rapport aux éléments du registre de sortie de l'UAL et aux retenues ?

3. Trouver les conditions logiques à tester pour effectuer un branchement conditionnel si  $A > B$  dans un contexte non signé et signé, ces conditions correspondent à `cc_UGT` (non signé plus grand que) et `cc_SGT` (signé plus grand que)

## TD2 - Entrées/Sorties séries

L'unité ASC0 (Asynchronous Synchronous Controller 0) du micro-contrôleur Siemens C167 permet de transmettre et recevoir des données sur une ligne série.

On souhaite écrire une application permettant de faire communiquer le C167 avec un PC par l'intermédiaire de cette ligne série. La communication consiste à envoyer des requêtes depuis le PC (demande de résultats de mesure) auxquelles le C167 doit répondre.

### 1 Paramètres de transmission

On souhaite effectuer les transmissions en mode asynchrone. Les paramètres choisis sont : 7 bits de données, parité paire, un seul bit de stop, vitesse de transmission : 57600 bauds.

1. Quelle valeur doit recevoir le registre de commande `SOC0N` sachant que l'on veut activer la détection des erreurs de réception éventuelles et que l'initialisation doit fonctionner aussi bien en émission qu'en réception ?
2. Le registre `S0BG` permet de régler la vitesse de transmission. Quelle valeur doit recevoir `S0BG` pour obtenir une vitesse de transmission de 57600 bauds ? On calculera cette valeur suivant la valeur fixée pour `S0BRS`.
3. Quel est l'erreur relative selon que l'on tronque ou que l'on arrondit la valeur calculée pour `S0BG` ?

### 2 Réception des données

Lors de la réception d'une donnée, un drapeau (bit `SORIR` du registre `SORIC`) passe à un indiquant qu'une donnée est disponible dans le registre de réception `SORBUF`.

Comment peut-on utiliser ce drapeau ?

En dehors des requêtes du PC auxquelles il doit répondre, le C167 a une tâche de fond à effectuer. Les requêtes pouvant apparaître n'importe quand, on choisit de les traiter par interruption de la tâche en cours. Le passage à un d'un drapeau peut-être utilisée par le micro-contrôleur pour générer une interruption à condition que l'utilisateur l'autorise. .

1. Quel bit de quel registre doit être programmé pour autoriser cette interruption ?
2. A quoi correspond le niveau d'une interruption ? Comment initialiser le registre de contrôle d'interruption pour une IT de niveau 3 (groupe 0) ?
3. Ecrire et mettre en place la fonction d'interruption permettant de recevoir une donnée de la ligne série.

### 3 Détection des erreurs

Si la détection des erreurs est activée, le drapeau `SOEIR` passe à un lorsqu'une erreur de réception survient.

1. On souhaite allumer une led suivant le type d'erreur détectée. Pour cela on relie trois leds aux bits 0 à 2 du port P7. Comment doit-t-on initialiser les registres associés au port P7 ?
2. Ecrire une fonction `void alarme_erreur()`, appelée si le drapeau `SOEIR` est levé, qui commande les leds en fonction du type d'erreur.

## 4 Emission des données

1. Représenter le chronogramme du signal émis si la donnée à transmettre est : \$56
2. Combien de bits sont effectivement transmis et par conséquent quelle est la durée d'émission d'une donnée? Combien de cycles d'horloge du C167 cela représente t'il?
3. L'émission d'une donnée consiste à la copier dans le registre de transmission SOTBUF. Que se passe t'il si on émet les données les unes après les autres sans précaution avec une ligne de programme du genre :

```
for (i=0; i<N; i++)
    SOTBUF=data[i];
```

4. Les drapeaux SOTIR et SOTBIR passent à un quand, respectivement, le registre de transmission SOTBUF et le registre à décalage SOSHR sont vides. Comment doit-on modifier l'extrait de programme précédent en utilisant un de ces drapeaux pour que l'émission se fasse correctement?

## 5 Analyse d'un programme

Le programme suivant permet d'envoyer les 26 lettres de l'alphabet sur la ligne série. Malheureusement, il n'a été ni documenté, ni commenté :

```
#include<c167.h>

void main()
{
    int i;

    SOCON=0xB08F;
    SOBG=0x56;
    SET_SFRBIT(DP3.10);
    SET_SFRBIT(P3.10);
    for (i=0; i<26; i++)
    {
        SOTBUF='a'+i;
        WAIT_UNTIL_BIT_SET(SOTBIR);
        CLR_SFRBIT(SOTBIR);
    }
    while(1);
}
```

Pour pouvoir écrire un programme de réception sur le PC permettant de recevoir ces données, retrouvez :

- le format de transmission des données
- la cadence de transmission.

# TD3 - Conversion Analogique/Numérique et gestion du temps

## 1 Lecture d'une voie analogique

Le convertisseur analogique numérique du micro-contrôleur Siemens C167, utilise les registres suivants :

registres de données	registre de contrôle	registres d'interruptions	port
ADDAT	ADCON	ADCIC ADEIC	P5

On souhaite lire une tension en mode conversion unique sur l'entrée analogique 3,

1. quelle valeur doit recevoir le registre de contrôle ADCON ?
2. si la tension appliquée est de 3V, quelle valeur contient le registre ADDAT en fin de conversion ?
3. **lecture en mode programmé** : écrire la fonction de lecture du signal qui renvoie le résultat de la conversion, la fin de conversion est détectée par scrutation du bit ADCIR. La fonction doit renvoyer la valeur en volts de la tension.

## 2 Echantillonnage de l'entrée analogique

On souhaite lire la tension en mode conversion unique à intervalle régulier. La période d'échantillonnage choisie est d'environ 25 ms. Pour cela on va utiliser un timer et on opte pour le timer T3 de l'unité GPT1. On rappelle les registres de cette unité :

registre. de données	registre de contrôle	registre d'interruptions	port
T2	T2CON	T2IC	P3
T3	T3CON	T3IC	DP3
T4	T4CON	T4IC	ODP3 P5

1. initialiser le registre T3CON
2. initialiser T3IC afin que la lecture de la voie se fasse sur interruption (de niveau 1) provoquée par un dépassement du temporisateur (événement T3IR).
3. Ecrire la fonction d'interruption, le numéro de trap lié à l'interruption sur dépassement de T3 est 0x23.

### 3 Mesure de fréquence

On souhaite mesurer la fréquence de signaux carrés en utilisant l'unité de gestion du temps (General Purpose Timer) GPT1 du micro-contrôleur C167.

Pour cela on va compter le nombre de fronts (montants ou descendants) du signal dont on veut connaître la fréquence pendant une durée fixée, durée correspondant à la période entre deux exécutions d'une fonction d'interruption

La fréquence du signal sera finalement obtenue directement en divisant le nombre de fronts par la durée ou période de comptage.

#### 3.1 Période de comptage

1. La durée de comptage est fixée à 1s. Quel temporisateur de GPT1 doit-on utiliser pour obtenir une interruption (provoquée par le débordement du temporisateur) toutes les secondes? Faire la liste des registres à initialiser.
2. Déterminer la programmation des registres permettant le fonctionnement souhaité. Le niveau d'interruption est fixé à 4.

#### 3.2 Comptage

Le temporisateur restant est utilisé pour compter les fronts montants du signal.

1. Sur quelle broche doit-on connecter le signal?
2. Pour obtenir une précision suffisante (de l'ordre de 1%), on veut mesurer au moins 100 fronts pendant la période de mesure, quelle est la fréquence minimale du signal que l'on pourra mesurer?
3. Donner l'initialisation du registre de contrôle du temporisateur nécessaire au comptage.
4. Ecrire la fonction d'interruption permettant de déterminer la fréquence du signal. Celle-ci devra par ailleurs préparer la mesure suivante.

# TD4 - Unités capture/comparaison et PWM

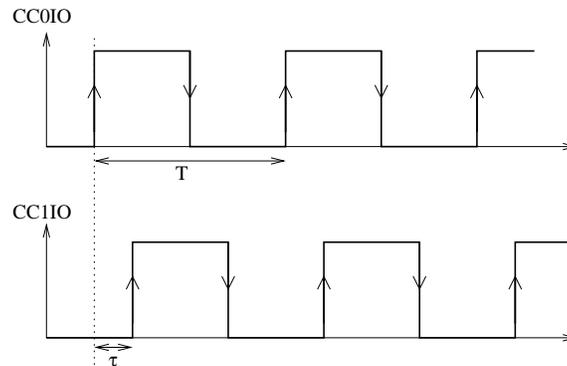
## 1 Mesure d'un déphasage angulaire

On souhaite mesurer le déphasage entre deux signaux carrés de même fréquence (signaux TTL) en utilisant les possibilités de l'unité CAPCOM1 du micro-contrôleur C167.

Le signal de référence est relié à la broche CC0IO. Le second signal est relié à la broche CC1IO. Les registres CC0, CC1 sont associés au temporisateur T0 et programmés en mode capture.

Le principe de mesure est le suivant (illustré par la figure) :

- Le front montant du signal de référence permet la capture de T0 qui est ensuite réinitialisé à zéro.
- Le front montant du second signal (signal déphasé) effectue lui aussi la capture de T0.



1. Sachant que l'on souhaite ne pas avoir de débordement de T0 au cours d'une période des signaux et que celle-ci sera toujours inférieure à 1 s (fréquence supérieure à 1 Hertz), quelle résolution doit-t'on choisir pour T0? En déduire la programmation du registre de contrôle de T0.
2. La valeur du déphasage en radians est donné par  $\varphi = 2\pi(\tau/T)$ . Déterminer l'expression permettant de le calculer en fonction des valeurs de T0 capturées.
3. La mesure du déphasage peut se faire en mettant en place une seule interruption. En vous aidant du chronogramme, indiquer quel est l'événement qui permet d'effectuer la mesure et qui doit donc être traité par interruption.
4. Donnez la programmation du registre de contrôle CCM0 et précisez quels sont le registre de contrôle d'interruption à initialiser.
5. Ecrire le code de la fonction d'interruption.

## 2 Génération de signaux

On souhaite utiliser l'unité CAPCOM1 pour générer deux signaux déphasés d'un angle  $\varphi$ .

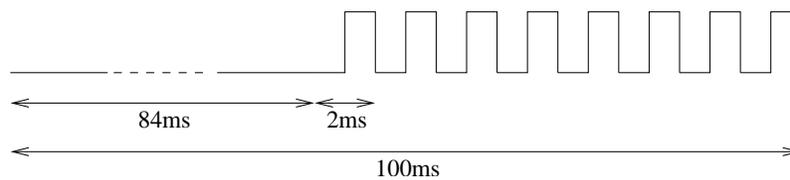
Le premier signal (signal de référence) est généré sur la broche CC0IO et le second sur la broche CC1IO.

1. Quel mode de comparaison doit-on choisir de préférence pour générer ces deux signaux sachant que leur rapport cyclique doit être égal à 0,5?
2. Dessinez le chronogramme de T0 et du premier signal généré pour le mode de comparaison de la question précédente, sachant que la valeur de comparaison CC0 est fixée à  $(T0REL+65536)/2$ .
3. Donner la relation permettant de calculer la période du signal en fonction de la résolution  $t_r$  du timer T0 et de la valeur de rechargement T0REL.

4. Rajouter au chronogramme le second signal (pour une valeur de comparaison CC1 supérieure à CC0).
  - Déterminer la relation fixant le déphasage  $\varphi = 2\pi(\tau/T)$  en fonction de la valeur de comparaison CC1.
  - Déterminer l'intervalle de réglage du déphasage possible si CC1 varie entre sa valeur minimale et sa valeur maximale.
5. Déterminer les initialisations des registres de contrôle CCM0 et T01CON ainsi que celles des registres de données T0REL, CC0 et CC1 pour une période de 50ms et un déphasage de  $+\pi/4$
6. Certaines initialisations ne doivent pas être omises. Lesquelles ?

### 3 Signal PWM

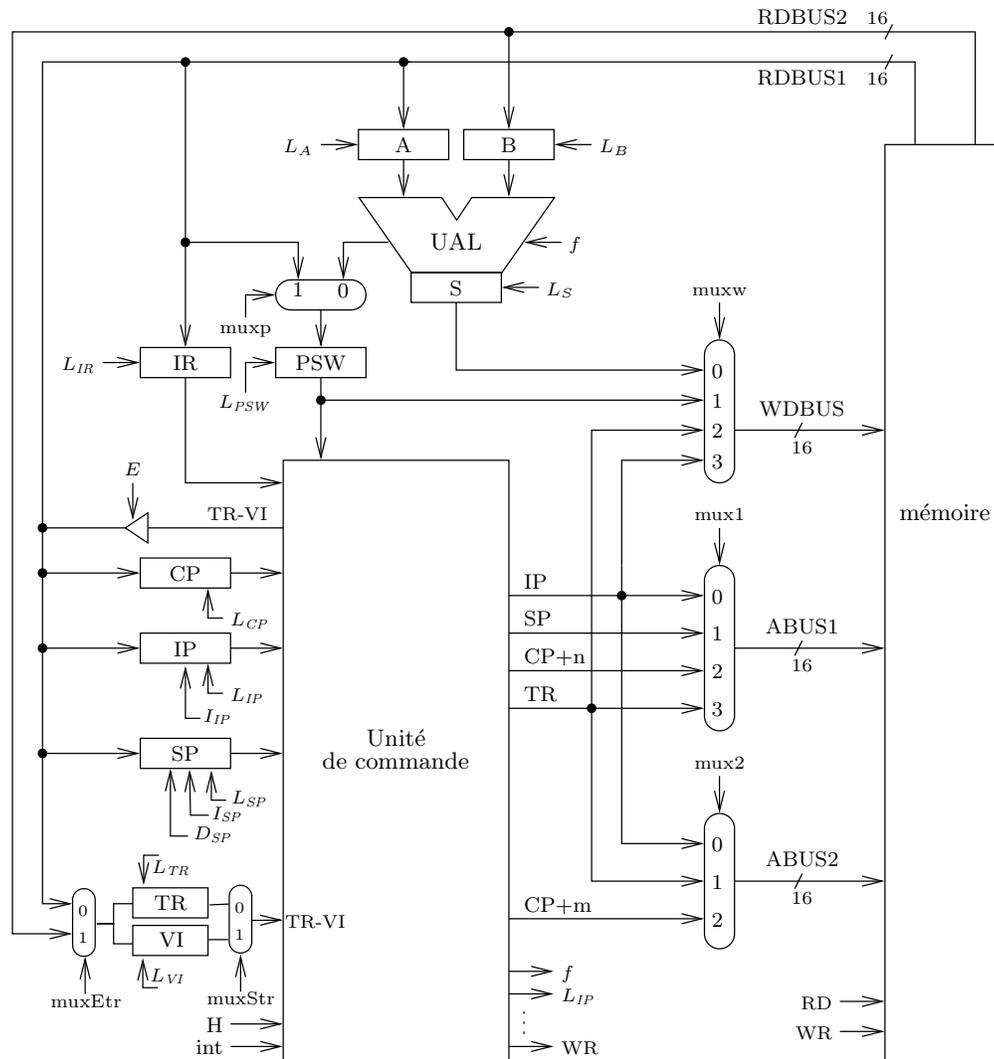
On souhaite générer un signal périodique, dont une période est représentée sur la figure suivante, en utilisant l'unité PWM du C167.



1. Quel est le mode de fonctionnement de PWM qui permet de générer un signal de ce type ? Sur quel bit de quel port peut-on récupérer le signal ?
2. Donner la programmation des registres de contrôle PWMCON0 et PWMCON1 de l'unité PWM permettant de générer ce signal.
3. Déterminer les valeurs à donner aux différents registres PPx et PWx utilisés. Justifier vos réponses.

## TD5 - Unité centrale du micro-contrôleur

La figure représente une version simplifiée du processeur qui est au cœur du microcontrôleur Siemens C167. Cette architecture a été présentée dans le chapitre 3 du polycopié de cours et les éléments qui la composent y ont été décrits.



Sur la figure les signaux de commande sont représentés par des petites flèches et par un symbole indiquant quelle est la fonction assurée :

- $L_x$  chargement du registre  $x$
- $I_x$  incrémentation du compteur  $x$
- $D_x$  décrémentation du compteur  $x$
- $f$  sélection de la fonction de l'UAL
- $muxz$  sélection d'une entrée du multiplexeur  $z$
- $E$  activation des circuits à logique 3 états
- $RD$  ordre de lecture en mémoire
- $WR$  ordre d'écriture en mémoire

## 1 Détermination de signaux

L'instruction `push Rn` (sauvegarde d'une donnée sur la pile) est exécutable en 4 phases :

1. *recherche de l'instruction* :  $IR \leftarrow M[IP]$ ,  $IP \leftarrow IP+1$
2. *décodage et extraction de la donnée* :  $A \leftarrow M[SP]$
3. *incrémentement du pointeur de pile* :  $SP \leftarrow SP+1$ ,  $S \leftarrow A$
4. *rangement de la donnée dans le registre* :  $M[CP+n] \leftarrow S$

Déterminer les signaux à générer pour chaque phase.

## 2 Analyse de signaux

Une instruction est réalisée en générant les signaux suivants :

1.  $mux1=0$ ;  $RD=1$ ;  $L_{IR} = 1$ ;  $I_{IP} = 1$
2.  $RD=1$ ;  $mux1=2$ ;  $mux2=0$ ;  $L_A = 1$ ;  $L_B = 1$ ;  $I_{IP} = 1$
3.  $L_S = 1$ ;  $L_{PSW} = 1$ ;  $f=00$ ;  $muxp=0$
4.  $mux1=2$ ;  $muxw=0$ ;  $WR=1$

Déterminer les transferts réalisés (en langage RTL) à chaque phase. En déduire l'instruction correspondante. Sur combien de bits cette instruction est-elle codée ?

## 3 Analyse d'instructions

Pour les instructions suivantes

- faire l'analyse RTL (Détermination des transferts à effectuer pour les chaque phase de l'instruction)
- associer aux différentes phases les signaux ouvrant le chemin des données pour assurer les transferts entre registres nécessaires à la réalisation de l'instruction.

### 3.1 Transfert de registre à registre

instruction	code instruction
<code>mov Rn, Rm</code>	F0 nm

### 3.2 Chargement en adressage immédiat

instruction	code instruction
<code>mov Rn, #data</code>	E6 Fn #####

la donnée sur 16 bits (#####) suit immédiatement l'instruction.

### 3.3 Rangement en adressage direct

instruction	code instruction
<code>mov mem, Rn</code>	F6 Fn MMMM

L'adresse sur 16 bits (MMMM) suit immédiatement l'instruction.