

Examen Électronique Programmable 2

Ordinateurs, téléphones (portables) et communication interdits, supports papiers autorisés.

1. Combien d'octets sont occupés en mémoire par une variable de type `short` en C ?

`int main() {printf("%d\n",sizeof(short));}` indique la réponse 2

2. Pouvons nous stocker la valeur $4 \cdot 10^4$ dans une variable de type `signed short` ? Justifier

$\pm 2^{15} - 1$ limite à ± 32768 donc il n'est **pas** possible de stocker $4 \cdot 10^4$

3. Combien de symboles (caractères) ASCII sont nécessaires pour afficher à l'écran le contenu d'une variable de type `short` ?

chaque quartet est transmis sur un caractère (0:F) pour les valeurs de 0 à $2^4 - 1$ donc il faut quatre symboles en hexadécimal, ou 5 en décimal (solution que j'accepte sans y avoir pensé initialement).

4. Combien de temps faut-il pour communiquer une variable de type `short` au travers d'une liaison asynchrone compatible RS232 cadencée à 9600 bauds ? Est-ce que cette question définit tous les paramètres du problème ? Si oui quelle est votre réponse ? Si non, proposez les valeurs les plus courantes des paramètres manquants et proposez votre réponse. Dans tous le cas, justifier le calcul.

Le problème est incomplet en l'absence du nombre de bits de parité et de stop. Le mode le plus classique est 8N1 donc 10 symboles par octet transmis. Un `short` nécessite deux octets donc $2/960 \simeq 2/1000 \simeq 2$ ms.

5. Quelle séquence d'instructions en C permettent d'extraire le quartet de poids le plus fort du `short s` sous forme de caractère ASCII stocké dans `char c` ?

`char c=(s>>12); if (c<10) c+='0'; else b+='A'-10;`

6. Sur un bus de communication SPI, combien de signaux de communication excluant les alimentations faut-il pour connecter le microcontrôleur avec 42 capteurs ? Est-ce possible avec un Atmega32U4 ? Justifier.

Trois signaux communs – MOSI, MISO et CLK – et un signal par capteur, CS#. Il n'y a pas 32 GPIO dans un Atmega32U4 donc cette configuration n'est pas possible.

7. Sur un bus de communication I²C, combien de signaux de communication excluant les alimentations faut-il pour connecter le microcontrôleur avec 42 capteurs ? Est-ce possible avec un Atmega32U4 ? Justifier.

Deux signaux communs uniquement – CLK et DAT. L'adresse est sur 7 bits donc peut encoder 42 interlocuteurs.

8. Nous avons déclaré une variable de type `char` initialisée à 3 et l'avons multipliée par une certaine valeur. Le code résultant prêt à être exécuté par l'unité arithmétique et logique du processeur est le suivant :

```
ldi r24,lo8(3)
mov r24,r25
lsl r24
lsl r24
lsl r24
sub r24,r25
```

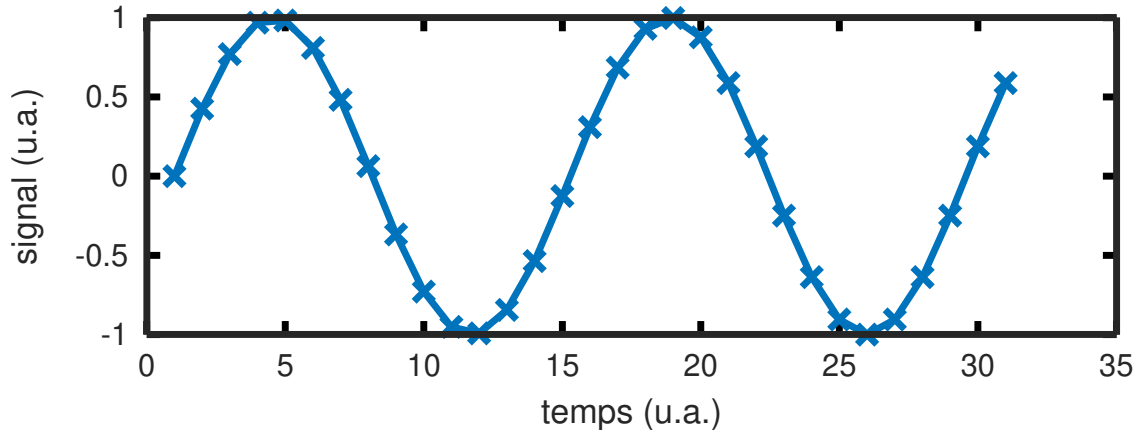
Quel est le facteur de multiplication appliqué ? Justifier le raisonnement qui mène à cette réponse. On rappelle qu'en assembleur AVR, `lsl` est le décalage à gauche du contenu du registre, `ldi` est une affectation de valeur immédiate, `mov` est une copie de valeur entre registres ou mémoires, et `sub` est une soustraction.

Décalle trois fois à gauche donc multiplie par 8 puis soustrait la valeur initiale qui a été stockée dans le registre R25 donc multiplication par 7 : le résultat est 21.

9. Quelle instruction en C permet d'affecter la valeur 0x24 à l'emplacement mémoire d'adresse 0x42 interprétée comme contenant une valeur codée sur 8 bits ?

`*(unsigned char*)0x42=0x24;`

10. Un signal continu émis à 1 kHz a été échantillonné par un convertisseur analogique numérique pour donner la mesure suivante



Quelle est la fréquence d'échantillonnage ? Justifier la réponse et son incertitude.

15 ± 1 points/période donc 15 ± 1 kHz si on ne compte que 1 période, mais 30 ± 1 points/2 périodes donc $30/2 \pm 0,5$ kHz si on compte sur 2 périodes.

11. Quelle est l'incertitude sur la fréquence d'un signal acquis pendant T secondes à f_s échantillons/seconde par un convertisseur analogique-numérique codé sur N bits et de tension de référence V_{ref} ? Justifier.

seule la durée de la mesure importe : la résolution spectrale est $1/T$ puisque N et f_s s'élimineront lors du calcul.

12. Quelle est la résolution en tension de la mesure fournie par le convertisseur mentionné ci-dessus ?

$V_{ref}/2^N$

13. Comment se nomme l'exécutable du compilateur libre qui permet de générer un code à destination d'un microcontrôleur Atmega32U4 ?

avr-gcc

14. Quelle option de ce compilateur permet d'ajouter des répertoires contenant les fichiers d'entête (d'extension .h) lors de la compilation ?

-I

Nombre de problèmes s'expriment sous forme de nombres complexes. Nous avons par exemple vu qu'en radio logicielle, la nature des données traitées est naturellement défini de cette façon. Cependant, les complexes ne sont pas une structure de données nativement supportée par le C.

15. Proposer une fonction qui calcule la somme de deux complexes et renvoie le résultat. On pourra investir un peu de temps à définir la structure de données adéquate, mais ne ferons en aucun cas appel à une bibliothèque externe.

L'approche la plus simple est de passer quatre paramètres que sont la partie réelle et imaginaire des deux arguments ainsi que *le pointeur* vers la partie réelle et *le pointeur* vers la partie imaginaire du résultat. Sans pointeur, la valeur ne sera pas retournée modifiée. Alternativement on peut construire une `struct` contenant deux éléments, partie réelle et partie imaginaire. Dans ce cas, la fonction somme peut renvoyer un `struct` contenant le résultat du calcul.

16. Proposer une fonction qui calcule le module d'un complexe et renvoie le résultat.

Le problème est plus simple puisque cette fois nous renvoyons un scalaire qui est la racine de la somme du carré de la partie imaginaire avec le carré de la partie réelle de l'argument fourni.

17. Proposer une fonction qui calcule le produit deux complexes et renvoie le résultat.

Le problème est le même que pour la somme sauf que la partie réelle du résultat est le produit des parties réelles des arguments moins le produit des parties imaginaires, et la partie imaginaire est la somme des produits croisés des parties réelle et imaginaire des arguments.

18. Supposons que nous ayons placé ces fonctions dans un fichier `complexe.c` afin de proposer une bibliothèque de calcul sur les nombres complexes : proposer le fichier d'entête (extension `.h`) correspondant.

Le fichier d'entête ne contient que la déclaration des fonctions (arguments d'entrée et de sortie) suivi de “;”

19. Supposons que nous voulions exploiter ces fonctions dans un programme contenu dans `main.c`. Quel mécanisme appelé par quelle instruction vous permettrait d'automatiser la compilation de ces fichiers ?

Makefile appelé par la commande `make` automatise la séquence de compilation.

20. Proposer le fichier de configuration de cet outil pour automatiser la compilation de `main.c` et `complexe.c` afin de générer l'exécutable `exec`.

```
all: exec

exec: main.o complexe.o
    gcc -o exec main.o complexe.o

main.o: main.c
    gcc -c main.c

complexe.o: complexe.o
    gcc -c complexe.c
```