

Licence SPI ingénierie L2 S3

TP n°2 Architecture des ordinateurs

Nous allons travailler aujourd'hui sur les méthodes d'entrée-sortie. A la fin de la première moitié du TP (2h), déposez sur Moodle les programmes que vous avez écrits, dans leur état (html, py, pdf, ipynb), inutile ici de couper ou mettre en forme. Vous me rendrez un rapport, m'expliquant ce que vous avez compris, mais aussi les difficultés que vous avez rencontrées, dès la fin de la seconde moitié du TP (les 2 autres heures). Pour le rapport, vous utiliserez votre traitement de texte préféré, me ferez une belle page d'entête, une introduction, une conclusion, quelques références internet si vous en avez trouvé des utiles... et me l'exporterez en pdf. Je vous donne tout de suite le sujet complet, sachant que j'avais prévu le découpage : Partie 1 : Q1 et 2, partie 2 : Q3 et peaufiner le rapport.

1) décalages et masquages

Supposons disposer d'un dispositif d'entrée-sortie tout ou rien (c'est à dire quelque chose qui nous permette d'allumer ou éteindre des sorties, par exemple votre chauffage, les moteurs des volets... et qui permette de lire des entrées, par exemple pour savoir si oui ou non un volet est fermé). Comme je ne peux pas vous en fournir en nombre suffisant, nous saisissons donc au clavier les valeurs que devraient nous donner les dispositifs d'acquisition, et afficherons à l'écran les informations que nous aimerions transmettre en sortie. Nous nous limiterons à 8 entrées (une variable entière) et 8 sorties (là aussi une seule variable, entière).

a) écrivez une fonction (def) qui affiche l'état (allumé ou éteint) de 8 bits (contenus dans un int donné en argument, pas de saisie clavier ici). Vous pouvez bien vous rappeler de ce que vous avez fait au premier TP. J'appelle bit 0 celui de poids faible. L'ordre d'affichage n'a aucune importance.

b) pour tester, le programme principal demandera au clavier la valeur de l'entier et appellera la fonction précédente.

c) écrivez désormais 5 fonctions qui reçoivent en argument un entier, ainsi que le n° du bit nous intéresse, retourne la valeur modifiée (mais n'affichent rien)

- affiche la valeur de ce bit
- met le bit à 0, sans changer les 7 autres (vérifiez avec la fonction d'affichage du a)
- met le bit à 1, sans changer les 7 autres
- inverse ce bit, sans changer les 7 autres
- fait une rotation sur 8 bits (chaque bit est décalé d'un cran, mais de plus le dernier est placé en premier).

il est important de **changer le contenu de la mémoire**, et pas seulement de changer l'affichage !

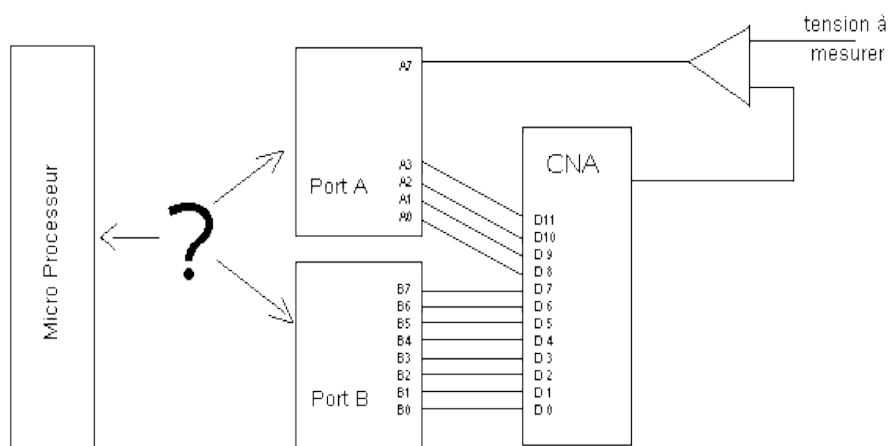
d) testez avec un programme qui demande la valeur et le n° du bit, et teste les 5 fonctions en affichant le résultat à chaque fois.

2) fabrication d'un voltmètre

Supposons disposer de deux ports comme traités dans la question précédente (8 bits chacun).

Branchons-y un Convertisseur Numérique - Analogique 12 bits (CNA), comme le montre la figure ci-contre. Le CNA reçoit (depuis l'ordinateur) un nombre en binaire sur 12 chiffres, et émet à sa sortie (à droite) une tension proportionnelle à ce nombre.

Pour simplifier, mettez ces 12 bits dans une seule variable entière.



Nous branchons cette sortie à un comparateur, qui la comparera à la tension que l'on désire mesurer. Le comparateur envoie 1 si la tension à mesurer est supérieure ou égale à celle donnée par le CNA, 0 sinon. Nous récupérons cette information sur le port A. (je vous rappelle que n'ayant pas physiquement ce matériel, c'est VOUS qui répondrez, au clavier, à chaque fois 0 ou 1).

a) programmez le voltmètre de manière incrémentale : on envoie 1, 2, 3... et ainsi de suite, jusqu'à ce qu'on atteigne la valeur à mesurer. On connaît alors la valeur de la tension, puisqu'elle est égale à celle qu'on a envoyée aux ports. Dans cette question vous avez le droit de faire des calculs (ajouter 1).

b) Si vous avez le temps (et n'êtes pas trop nuls en programmation), programmez le même voltmètre, mais avec une méthode bien plus efficace : la dichotomie. Ne cherchez pas sur internet, car vous devez utiliser exclusivement cet algorithme : On connaît le minimum (0) et le maximum (douze 1) que peut traiter le CNA. On essaye d'abord la valeur du milieu de cette plage (1 et onze 0). On regarde ce que donne le comparateur. Et on en déduit s'il faut continuer dans la première moitié (entre mini et max/2) ou la seconde (entre max/2 et max). On divise la plage par 2 par un **décalage à droite**, pas un calcul ! On recommence (comparer avec le milieu de l'intervalle) jusqu'à ce que l'intervalle n'ait plus qu'une taille de 1. N'utilisez **que** des masquages, décalages et fonctions logiques. Si vous avez déjà utilisé un voltmètre d'entrée de gamme, vous avez vu qu'avant qu'il ne se stabilise, il affiche les différentes valeurs qu'il teste, et qui vous semblaient jusqu'à aujourd'hui aléatoires.

3) remplissage de vérins

Ce sujet correspond à un besoin réel, dans une entreprise de la région. Sa production nécessite le remplissage d'azote, sous une pression précise, d'une enceinte hyperbare (en clair : récipient supportant les hautes pressions). Auparavant, la pression était réglée manuellement, en actionnant la molette d'un régulateur (dit plus simplement : on ouvre un robinet). L'opérateur tourne la molette (dans le sens trigo) tout en surveillant un manomètre à cadran, jusqu'à ce que la pression désirée soit atteinte. Les produits dans l'enceinte comportant un clapet anti-retour, tout dépassement de la pression prévue entraîne la mise au rebut du lot de produits. Ceci nécessite donc une attention soutenue de l'opérateur, qui pour éviter de dépasser la pression de consigne, ajuste lentement la pression, ce qui ralentit le cycle de production.

Nous avons donc envisagé d'automatiser cette tâche. L'outil de production (la "machine") étant satisfaisant, et vu qu'en recréer une nouvelle, même plus performante, coûterait largement plus que le gain de productivité escompté, il a été décidé de ne pas modifier la "machine" dans son fonctionnement, mais que seule la commande du régulateur se ferait automatiquement. Pour cela, il a suffi d'y installer un moteur pas à pas (et un réducteur), de rajouter un capteur de pression sur l'enceinte, et d'intercaler entre les deux un dispositif de commande : on a choisi d'utiliser un ordinateur (PC), pour son prix (récupéré dans le parc machines de l'entreprise, un modèle un peu ancien suffisant largement), mais aussi pour ses possibilités de traitement (base de données de production).

Le capteur de pression envoie un signal à un CAN. Celui-ci nous donne un entier (12 bits) proportionnel à la pression mesurée. Nous avons étalonné et calibré l'ensemble, et après plusieurs essais déterminé que

$$\text{pression réelle} = (\text{valeur_can} - 412) / 6,6$$

Nous avons également remarqué que la valeur mesurée fluctuait légèrement autour de la valeur réelle, le problème a été résolu en faisant 10 lectures d'affilée et en prenant la moyenne.

3a) écrivez une fonction qui permette de faire une lecture du capteur (pour la suite vous ne vous en servirez pas car ça fait trop de valeurs à saisir au clavier, puisque vous ne disposez pas de la machine).

3b) faites le programme de commande, d'après le Cahier des Charges suivant :

L'opérateur entre la pression désirée (input). Dès qu'il appuie sur la touche "entrée", le cycle démarre. On ouvre le régulateur (en envoyant des impulsions : une milliseconde allumé, une ms éteint...) jusqu'à atteindre 90 % de la pression désirée. Puis on continue à ouvrir, mais plus lentement, (1ms allumé, 3ms éteint) jusqu'à 97 % de la pression désirée, puis encore plus lentement (1ms allumé, 10ms éteint) jusqu'à 100% de la pression désirée. Puis il faut remettre le régulateur en position initiale, en renvoyant, sur une seconde sortie, exactement le même nombre d'impulsions (1ms allumé, 1ms éteint).

PS : Pour faire attendre l'ordinateur pendant 1ms, il faut appeler la fonction `time.sleep(0.001)`. Pour avoir le temps de voir ce qui se passe, vous pouvez mettre cette valeur dans une variable flottante, que vous fixerez plus grande (1.0 par ex). Et à chaque fois que vous modifiez la sortie, appelez la fonction de la Q1 pour afficher vos sorties (on a une carte 8 bits, utilisez les 2 bits de poids faible, les autres servent pour une vanne de sécurité et divers voyants).