Ex 2.1 Programmer Fibonacci sur le Dauphin – Assembleur

Indications pour la programmation de Fibonacci

On a à disposition les registres A, B, X, Y et la pile. A et B sont utilisés pour l'affichage. On va donc utiliser X et Y et comme l'addition se fait entre 2 registres et pas trois, il faut trouver une astuce en se disant que le résultat de l'addition X+Y va dans Y.

il faudrait copier Y dans X. La solution est de copier Y sur A ou sur la pile et le mettre dans X après l'addition.
Pour la partie affichage, on copie Y dans A et on mets B=0.
Pour arrêter, il faut surveiller le Carry et utiliser l'instruction Jump,CS ou Jump,CC

Pour afficher le résultat, il faut utiliser la routine _DisplayHexaByte.
Sa documentation se trouve sous ROM en haut à droite de l'écran. Quelques routines ont été mise dans la zone de mémoire morte, à partir de l'adresse 0x800.

Avant d'appeler _DisplayHexaByte il faut préparer deux paramètres dans A et B.
Le registre A: contient le nombre à afficher. C'est un nombre 8 bits, donc il va allumer 2 digits. Si B contient 0 on affiche sur les 2 premières positions.

Pour tester cet appel, écrire le programme :

```
    Move   #H'3F,A
    Move   #0,B   ; Essayer #1 #2 #3 #4
    Call   _DisplayHexaByte
    Halt
```

Cliquer sur CODE, puis vitesse 10, puis RUN - ou STEP

Différence entre l'assembleur du Dauphin et le C

- Les commentaires commencent par un ; et pas par //
- Les nombres hexa utilisent la notation H'C3 (et non 0xC3)
- Les minuscules et majuscules sont identifiées avec l'assembleur Dauphin
- Utiliser des tabs et non pas des espaces pour la mise en page. En C on préfère 2-3 espaces.

Ex 2.2 Microprocesseur et microcontrôleur

Question : Quelle est la différence entre un microcontrôleur et un microprocesseur ?

Ex 2.3 Microcontrôleurs 8-16-32 bits

Dans une application qui nécessite une bonne précision de calcul, on doit effectuer plusieurs fois par seconde quelques calculs arithmétiques avec des nombre de 32 bits.

Question : Est-ce qu'on est obligé alors de choisir un microcontrôleur 32 bits ?

Ex 2.4 Fréquence et puissance de calcul

Question : Si la fréquence d'un microcontrôleur est supérieure à celle d'une autre microcontrôleur, peut-on dire que sa puissance de calcul est supérieure ?

Ex 2.5 Programmation en C/Arduino : clignoter pendant que l'on presse

Réaliser un programme qui permet d'utiliser un bouton poussoir actif à zéro, câblé sur la pin 4 ainsi qu'une Led câblée sur la pin 5.

Si on presse et que l'on maintient le touche enfoncé, la Led doit clignoter à 2 Hz (période 500 ms).

Ex 2.6 Programmation en C/Arduino : allumer 2 secondes quand on presse

Réaliser un programme qui permet d'allumer une led pendant 2 secondes, si on presse (impusion) sur un bouton – pin 4 (par exemple)

Ex 2.7 Programmation en C/Arduino : Un poussoir pour 2 leds

Réaliser un programme qui permet à l'aide d'un bouton poussoir et deux leds d'allumer ou d'éteindre la led1 ; la led2 ne doit changer d'état que si la led1 est allumé et le bouton poussoir est activé.

Ex 2.8 Programmation en C/Arduino : Deux poussoirs pour une led

Réaliser un programme qui permet d'allumer une led si le bouton poussoir P1 est activé et d'éteindre celle-ci si par le bouton P2

Ex 2.9 Programmation d'un compteur up/down

On a une variable `byte compteur;` dont on va modifier la valeur avec deux poussoirs qui agissent statiquement. On ne sait pas encore comment afficher cette valeur, mais pour avoir une idée, on allume la led pour les valeurs impaires. Un clignotement de la led indique que le compteur augmente ou diminue (rajouter des leds si vous voulez).
Quand on pèse sur Pous1 et on maintien pressé (action statique), le compteur augmente chaque 0.2 seconde, mais se bloque à 7, led allumée puisque c'est impair.
Quand on pèse sur Pous2 et maintien pressé, le compteur décompte et se bloque à zéro, led éteinte.

Ex 2.10 Estimation de la durée de la fonction Arduino Map ()

La fonction Arduino map() doit faire des multiplications et division. Quel est son temps de calcul ?
On met cette fonction dans une boucle exécutée 10'000 fois (max 30000 en 16 bits signés). On mesure la période entre deux basculements de la led. Si la fonction dure 100 microsecondes, la période est de 1 secondes. Ecrire le programme et dire quelle est la durée de la fonction map().