

Ex 3.1 Variables et arithmétique

Nous avons les variables suivantes : byte x=8; byte y=3; byte z=250;

Rappels:

- le type byte (ou unsigned char) est un nombre entier de 8 bits, capable de prendre des valeurs de 0 à 255.
- l'opération / est la division entière
- l'opération % est le reste de la division entière

Question : Calculez :

- | | | | | |
|----------|----------|----------|----------|----------|
| 1) x + y | 2) y + z | 3) x + z | 4) y - x | 5) x / y |
| 6) y / x | 7) z / y | 8) z % y | | |

Ex 3.2 Autre manière d'écrire la boucle for

Quand on écrit une boucle : for (i=0; i<10; i++) {instructions}

Ecrivez cette même action sans le mot "for" !

Ex 3.3 Utilisation de l'instruction break

L'instruction `break;` permet de sortir d'un { bloc d'instructions } avant sa fin. Il est souvent utilisé comme une sortie "en catastrophe". Par exemple, on peut lancer un clignotement dans une boucle `while ()` et en sortir avec un `break;` après le 3e clignotement :

// extrait du programme Solution310.ino

```
byte i=0;
```

```
while (1)
```

```
    Led1On; delay (500);
```

```
    Led1Off; delay (500);
```

```
    if (i >= 3) { break; }
```

```
    i++;
```

```
}
```

```
delay (1000); // on continue avec d'autres instructions ou on recommence dans la boucle principale.
```

Question : Ecrire un programme qui attend une pression sur Pous1 et sort de l'attente en clignotant si on ne presse pas dans les 5 secondes. On doit donc presser régulièrement pour réinitialiser un compteur qui augmente avec une période de 50ms.

Ex 3.4 Test de tous les ports d'un microcontrôleur

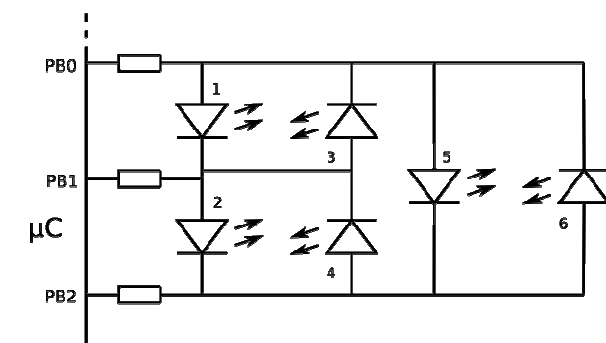
Ecrivez un programme qui teste toutes les sorties d'un microcontrôleur en les faisant clignoter : tous les bits pairs allumés, puis tous les bits impairs.

Ex 3.5 Charlie-plexing

Comment faire lorsqu'on a 6 LED, qu'on doit pouvoir en allumer une parmi les 6 et qu'on ne dispose que de 3 pattes sur un microcontrôleur ?

Il existe une solution astucieuse, connue sous le nom de Charlieplexing !

Voici le schéma :



Ecrivez les deux instructions qui allume la LED 5 sans en allumer d'autre.

Ex 3.6 Champs de bits

Ecrivez d'une manière plus lisible l'instruction suivante : `DDRC = 0b01000001;`

Ex 3.7 Maximum dans une table

Dans une table de bytes, on doit trouver où se trouve la valeur maximale. Par exemple, si la table est `byte table [6] {23,43,66,51,11,62};` la réponse est 3, puisque 66 est le 3e élément, et pour le test on va clignoter 3 fois.

Attention c'est l'élément 2 de la table puisque les index sont comptés à partir de 0.

L'algorithme doit comparer les éléments successifs, en mettant à jour si nécessaire le maximum et sa position.

Ex 3.8 Combien de 1 ?

On doit compter le nombre de 1 dans un mot de 16 bits. L'algorithme consiste à tester si le bit en extrémité vaut 1 ou 0, puis à décaler.

Indiquez le résultat par des clignotements. Tester des données en :

- binaire : mot = 0b0000 1000 0100 1100; résultat 4
 - hexadécimal : mot = 0x 3C41; résultat 6
 - décimal : mot = 1025; résultat 2 // le compilateur converti en binaire, pas en BCD.
-

Ex 3.9 Clignoter toujours plus

On vient de voir une boucle for pour clignoter un certain nombre de fois. Ecrivez une fonction qui permettra d'écrire dans le programme :

```
ClignoterNfois (nombre de fois);
```

Utiliser cette fonction dans un programme qui clignote une fois, deux fois, etc et s'arrête à 5 fois.

Ex 3.10 Clignoter davantage en accélérant

Cette fonction a un 2e paramètre qui est la vitesse. Testez avec un programme qui à la fois augmente le nombre de clignotement et la fréquence, avec une durée de salve constante. On voit donc le produit de la période par le nombre d'impulsions est constant.

Ex 3.11 Mesurer une durée

On veut une fonction qui mesure la durée pendant laquelle on presse sur un poussoir. La durée sera en unité de 50ms, mais on essayera des valeurs plus faibles jusqu'à être perturbé par les rebonds de contact. Pour tester, écrire le programme qui allume la Led1 si on est très rapide, et la Led2 si on est lent.