

TP N°1

Exercice 1 (15 minutes) : réécrivez le code ci-dessous dans un éditeur de texte (typiquement Dev-C++ sous windows) et compilez-le. En conséquence, le compilateur renvoie des erreurs en indiquant qu'il y a quelques choses qui manquent. Veuillez lire les erreurs (en anglais) et essayez de les corriger pour réussir la compilation.

```
#include<iostream>

int main()
{
    cout<<"introduction aux systèmes embarqués"<<endl ;
    system(« pause ») ;
    return 0 ;
}
```

Exercice 2 (15 minutes) : on veut lire la valeur donnée par un capteur analogique (potentiomètre) et la convertir en valeur numérique en degré. Sachant que la valeur d'entrée est comprise entre 0-1024, et le potentiomètre peut tourner entre -90° et +90°.



Figure 1. Photo réelle d'un potentiomètre.

1024	512	0
90°	0	-90°

Ecrivez un programme pour simuler le scénario décrit plus haut, où la valeur d'entrée (du capteur) peut être lue au clavier avec la fonction c++ dédiée. Le programme doit afficher l'angle de rotation en degré.

Exercice 3 (20 minutes) : on a un robot mobile et on veut mesurer la distance entre la position courante donnée par un capteur GPS et la position de destination. Ecrivez un programme pour calculer la distance Euclidienne entre les deux positions, sachant que les coordonnées de la position de destination sont fixes (constantes) et les coordonnées du robot sont variables (longitude et latitude). *Attention au choix de types.*

Exercice 4 (15 minutes) : on reprend l'énoncé du deuxième exercice, et on extrait le signe de l'angle soit positif soit négatif. Sachant que si le 8^{ème} bit est mis à 1 ça veut dire que la valeur est négative.

Exercice 5 (25 minutes) : supposant qu'on a créé un protocole de communication dédié pour transmettre un byte à travers un module radio. Le byte transmit contient trois informations relatives (trois états) à trois actionneurs comme illustré dans la figure ci-dessous (Figure 2), où les trois premiers

bits contiennent la valeur du 1^{er} actionneur, les trois seconds contiennent la valeur du 2^{ème} actionneur et les deux derniers contiennent la valeur du 3^{ème} actionneur.

bit	0	1	2	3	4	5	6	7
act	Actionneur 1			Actionneur 2			Actionneur 3	
val	0..6			0..8			0..2	

Figure 2. Structure du byte envoyé à travers le réseau.

Ecrivez un programme pour extraire les trois valeurs des trois états à partir du byte transmit et mettez chaque valeur dans une variable (et affichez le contenu). On peut lire le byte transmit au clavier pour simuler le processus (exemple : 210).

NB. *Le but de ce protocole consiste à optimiser le temps de transmission et le tampon (mémoire) (nombre de bytes).*

Exercice 6 (30 minutes) : contrairement à l'exercice N° 5, on veut remplir le byte à transmettre avec les trois valeurs des trois actionneurs (lues au clavier). Donc, écrivez le programme dédié pour construire le byte à partir des trois valeurs des trois actionneurs (attention à l'ordre des bits), puis affichez le byte final.

Remarque : *chaque étudiant est noté entre A-D sur la participation et la dynamique durant la séance du TP. Cette note sera accumulée avec l'assiduité et le test final.*