

Instructions C++ les plus utiles avec la carte ARDUINO

LES VARIABLES

Les variables sont désignées par un nom. Elles doivent être déclarées en précisant le type de donnée qu'elles vont stocker. Lors de la déclaration, les variables peuvent être initialisées, mais ce n'est pas obligatoire. Quelques exemples :

```
int var1 : déclare la variable "var1" de type "entier"
```

```
int var2=0 : déclare "var2" de type "entier" et initialise sa valeur à 0
```

```
float mesure1 = 0.0 ; // déclare et initialise mesure1 à un nombre réel
```

Les variables doivent être déclarées en début de programme, avant la procédure "setup()"

ENTREES ET SORTIES

Les broches numériques utilisées dans le programme doivent être déclarées comme entrée ou sortie dans la procédure setup() :

```
pinMode(3, OUTPUT) : déclare la broche numérique 3 comme sortie
```

```
pinMode(4, INPUT): déclare la broche numérique 4 comme entrée
```

Pour définir l'état d'une sortie numérique :

```
digitalWrite(3, HIGH): impose la valeur HIGH (5 volt) à la broche 3
```

```
digitalWrite(3, LOW): impose la valeur LOW (0 volt) à la broche 3
```

Pour lire une entrée numérique :

```
flag = digitalRead(4) : lit la valeur HIGH ou LOW de la broche numérique 4 et l'attribue à la variable flag (déclarée comme un entier au préalable)
```

Pour une sortie pseudo-analogique (numérique en mode PWM) :

```
analogWrite(6,0) : impose la valeur 0 à la broche 6 (0 volt)
```

```
analogWrite(6,255) : impose la valeur 255 à la broche 6 (5 volt)
```

```
analogWrite(6,100) : impose la valeur 100 à la broche 6 (2 volt en moyenne)
```

Pour lire une entrée analogique :

```
mesure1 = digitalRead(A5) : mesure la tension sur la broche A5 (un nombre entier entre 0 et 1023)
```

COMMUNICATION PAR LE CABLE USB

Il faut définir la vitesse de transfert dans la fonction setup() :

`Serial.begin(9600)` : initialise la broche série, vitesse de 9600 bauds

La carte peut alors envoyer des informations à l'ordinateur :

`Serial.print("Mesure")` : la carte envoie du texte

`Serial.print(lecture)` : la carte envoie la valeur de la variable lecture

`Serial.println("toto")` : la carte envoie du texte et un saut de ligne.

AUTRES INSTRUCTIONS TRES UTILES

Temporisation

`delay(100)` : attente en millisecondes, ici 100 ms

`delayMicroseconds(10)` : attente en microseconde, ici 10 µs

La boucle for

```
for (index=0; index < 100; index=index+1){  
// instructions de la boucle à exécuter 100 fois  
}
```

Cette boucle s'exécutera 100 fois ; la variable index (déclarée comme un entier au préalable) variera de 0 à 99, augmentant d'une unité à chaque itération).

La boucle while

```
index=0;  
while (index < 3) {  
// instructions de la boucle à exécuter tant que index < 3  
index=index+1;  
}
```

La boucle s'exécutera tant que la valeur de la variable index initialisée avant la boucle à 0 sera plus petite que 3.

Test conditionnel if

```
if (varTest==2) {  
// instructions à écrire ici si varTest vaut 2 ;  
}  
else {  
// instructions à écrire ici si varTest est différent de 2;  
}
```

Cette liste est loin d'être exhaustive, consulter
<https://www.arduino.cc/reference/en/>