Ecrire dans le port série

Application : visualiser et transmettre des données en cours d'acquisition (acquisitions réelles à suivre)

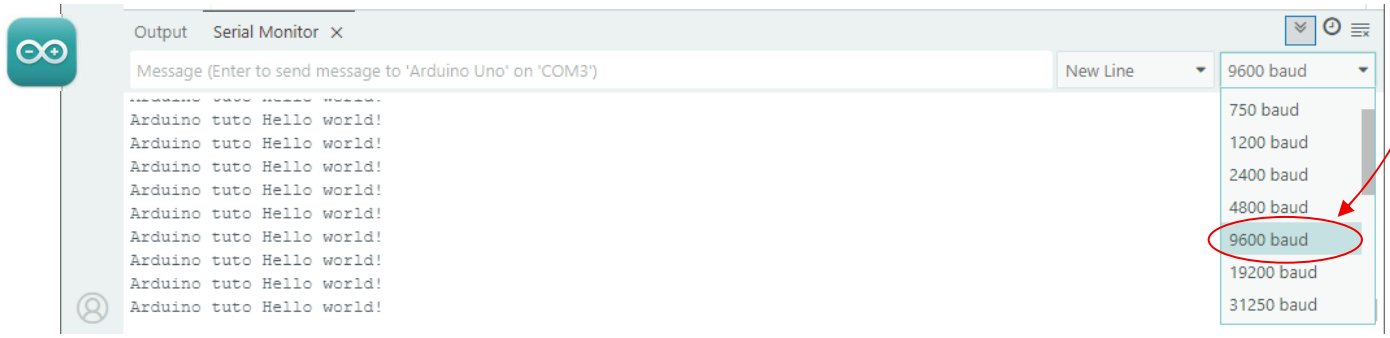
Le sketch suivant illustre la communication entre la carte Arduino et le ***port série*** de l'ordinateur (physiquement, via le câble USB).

Instructions

`Serial.begin``(v)`	Ouvrir le port série et fixer la vitesse *v* de transmission (valeurs prédéfinies, cf. ci-dessous).
`Serial.print``(s)`	Ecrire la chaîne *s* (ligne courante si existante ou nouvelle ligne sinon) sans retour à la ligne final.
`Serial.println``(s)`	Ecrire la chaîne *s* (ligne courante si existante ou nouvelle ligne sinon) puis retour à la ligne.
`delay``(n)`	Attendre *n* ms (millisecondes).

```
1  /*
2                    Exemple 1 - Ecrire dans le port série
3  */
4
5  void setup() {
6  // Ouverture port série - vitesse transmission (en baud = bit/s)
7  Serial.begin(9600);
8  // Tant que le port série n'est pas prêt, attendre 100 ms
9  while (!Serial) {
10       delay(100);     // Attendre 100 ms
11     }
12 }
13
14 void loop() {
15 Serial.print("Arduino "); // Affichages les uns …
16 Serial.print("tuto "); // … à la suite des autres (même ligne)
17 Serial.println("Hello world!"); // Affichage puis retour à la ligne
18 delay(1000); // Attendre 1000 ms
19 }
```

⚠ **Vérifier que la vitesse de transmission indiquée dans le port série correspond à la vitesse indiquée dans le sketch sous peine de ne rien voir s'afficher**.



Remarque : la boucle étant infinie, le flux dans le port série est ininterrompu.

Que signifie l'instruction **`!Serial`** (ligne 9) ?
Quelle est la différence entre les instructions **`print()`** et **`println()`** ? Ne pas hésiter à modifier le code pour effectuer des tests.