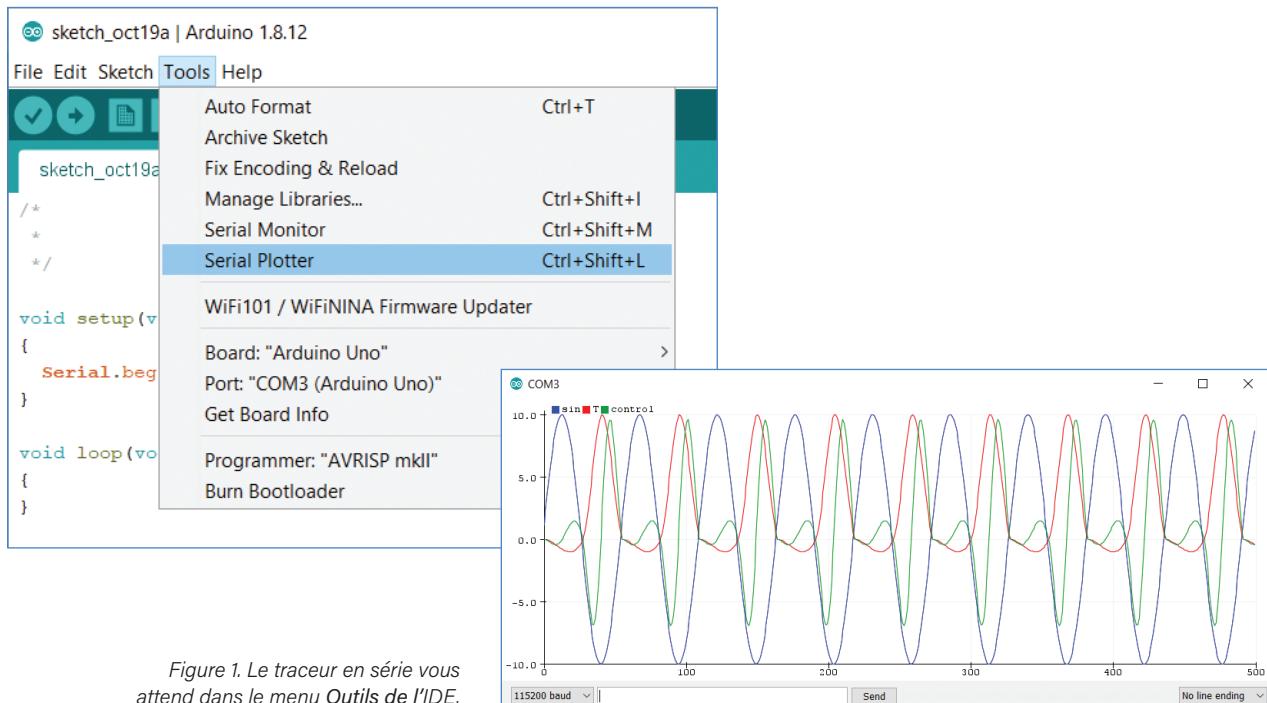


tracer des graphiques avec Arduino

C'est facile avec le traceur série d'Arduino



Clemens Valens (Elektor)

L'un des points forts d'Arduino est son port série bien pratique pour exporter des données, recevoir des commandes ou pour déboguer. Il existe un moniteur série et un traceur série pour visualiser ces données. Le premier marche tout seul, et voici comment utiliser le *Serial Plotter*.

La plupart des utilisateurs d'Arduino connaissent le *Serial Monitor* intégré à l'IDE (dans le menu *Outils*, ou avec *Ctrl+Maj+M*). Ce modeste terminal série affiche toutes les données reçues sur le port série. Il permet d'envoyer des chaînes de caractères (p.ex. des commandes ou des données) de l'ordinateur à la carte Arduino. Pendant le débogage, utilisez le *Serial Monitor* pour faire envoyer par le sketch des messages texte sur ce qu'il fait ou ce qu'il est censé faire, c'est pratique.

Peloteur en série

Le *Serial Monitor* peut aussi afficher des valeurs numériques, bien sûr. Vous avez déjà essayé de suivre l'évolution d'un paramètre sous forme d'une liste de valeurs imprimées ? C'est plutôt soporifique...

L'EDI Arduino comporte un *Serial Plotter* (dans le menu *Outils*, ou avec *Ctrl+Maj+L*, **fig. 1**).

À partir des données numériques que reçoit ce traceur en série, il crée des graphiques et les affiche. Ce n'est pas la cybernétique avec plein d'options, mais c'est facile à utiliser. Seulement il n'y a pas de doc à son sujet. Seul le code source Java [1] peut vous donner une idée de ses capacités. Essayez et vous découvrirez les fonctions suivantes :

- nombre illimité de graphiques ;
- mise à l'échelle automatique de l'axe vertical ou Y ;
- 500 points sur l'axe horizontal ou X ;
- afficher les 500 points de données les plus récents.

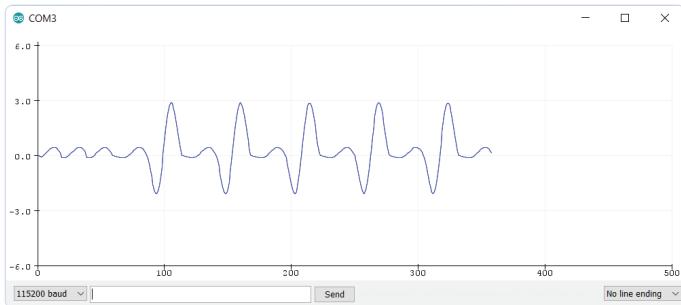


Figure 2. L'axe horizontal peut accueillir jusqu'à 500 points. Si vous en envoyez plus, il se mettra à défiler.

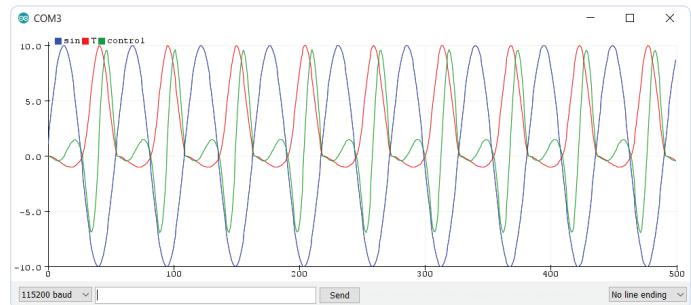


Figure 3. En théorie, le nombre de graphes est illimité. Utilisez des libellés pour savoir quel graphique représente quelles données.

Peloter un graphique

Après avoir ouvert le traceur en série, choisissez la même vitesse de transmission que celle qui est spécifiée dans `Serial.begin` dans le *sketch*. La case voisine de la boîte de sélection de la vitesse est destinée à renvoyer les données vers le *sketch*, exactement comme dans le *Serial Monitor*. Cela vous permet de contrôler le croquis dans une certaine mesure.

Vous n'aurez besoin d'apprendre ni le grec ni aucune commande spéciale pour utiliser le traceur série, ce sont les mêmes commandes de port série que le moniteur série. Peloter en série, c'est juste une autre façon de visualiser les données envoyées par le port série. C'est seulement le formatage des données qui importe. Pour tracer le graphique d'un seul paramètre (fig. 2), une valeur de capteur p.ex., il suffit de séparer les valeurs successives par un caractère de saut de ligne. Autrement dit, pour envoyer les valeurs, utilisez

```
Serial.println(sensor_value);
```

Les échelles s'adaptent à la plage

Le traceur utilisera ces valeurs pour l'axe vertical, l'axe Y ; il incrémentera lui-même l'axe horizontal (X). Autrement dit, la première valeur envoyée sera considérée comme la valeur «y» pour $x = 0$, la deuxième valeur est la valeur «y» pour $x = 1$, et ainsi de suite. Lorsque x atteint 500, le graphique commence à défiler horizontalement ; à mesure que les points les plus anciens sont supprimés au début, de nouveaux points sont ajoutés à la fin .

Lorsque la *valeur y* sort des limites, l'échelle de l'axe vertical s'adapte automatiquement pour rester dans les limites verticales. Inversement, si toutes les valeurs tiennent entre dans des limites resserrées (lorsque le graphique défile et qu'une valeur élevée a été supprimée au début), l'échelle s'adapte à la nouvelle plage de valeurs.

Peloter plusieurs graphiques

Pour tracer deux paramètres ou plus, envoyez les différentes valeurs sous forme d'une liste de valeurs séparées par une virgule, un espace ou un caractère de tabulation (\t) et fermez la liste avec un caractère de retour à la ligne :

```
Serial.print(temperature);
Serial.print(',');
Serial.print(humidity);
```

```
Serial.print(humidity);
Serial.print(',');
Serial.print(pressure);
Serial.println();
```

Le format qui sépare les données par une virgule est dit CSV, pour *comma-separated values* (valeurs séparées par des virgules). La plupart des tableurs reconnaissent ce format, certains vous permettent de choisir vous-même le caractère séparateur. Une fois que le résultat vous plaît dans le *Serial Plotter*, exportez les valeurs avec le *Serial Monitor* pour les copier-coller dans un tableau et obtenir de jolis graphiques et faire d'autres traitements.

La liste des valeurs des données est lue comme des valeurs y différentes pour une valeur x invariable, de sorte que tous les graphiques sont actualisés au même moment et à la même vitesse. Le traceur affectera une couleur différente à chaque graphique sans demander l'avis de l'utilisateur. La seule façon d'influer un peu sur ce choix est de jouer sur l'ordre des données.

Libellés

Ce peloteur ne manque pas de fantaisie, puisqu'il permet de spécifier un libellé pour chaque graphique (fig. 3). Il y a deux façons de le faire :

- Avant d'envoyer des valeurs de données, envoyez une liste de libellés séparés par une virgule, un espace ou une tabulation et terminez avec un caractère de retour à la ligne. Ces caractères réservés ne peuvent donc pas figurer dans les libellés. Pour qu'elles correspondent, les données doivent être envoyées dans le même ordre que les libellés. Exemple :

```
Serial.println(<<temperature,humidity,pressure>>)
```

- Envoyez étiquettes et valeurs par paires et séparez-les par deux points (:), par exemple :

```
Serial.println(<<temperature:21,humidity:76,
pressure:1007>>)
```

Envoyez ces paires toujours dans le même ordre pour que le traceur ne mélange pas les données des graphiques. Une liste de paires de valeurs de libellés doit être séparée à son tour soit par une virgule, un espace ou une tabulation et se terminer par un caractère de retour à la ligne.

Toute chaîne de texte sera considérée comme un libellé, et vous ne manquerez pas d'occasions de vous fourvoyer, par exemple en oubliant les délimiteurs, ce qui fait que les données se changeront en libellé.

Caprices et limites

- L'axe horizontal se prolonge à l'infini. Pas moyen de boucler le tracé par programmation. À moins de fermer le traceur et de le rouvrir. Parfois, il faut s'y reprendre à deux fois pour se débarrasser des libellés et/ou des données périmés.
 - Vous ne pourrez pas tracer un point à une position (x,y) de votre choix sans que de nouveaux points soient ajoutés derrière.
 - Que l'axe vertical se mette à l'échelle automatiquement, c'est bien, mais ça devient très ennuyeux lorsque des aberrations ou des pépins forcent l'échelle à augmenter et rendent difficile, voire impossible, la visualisation des données intéressantes beaucoup plus petites. Cela peut également se produire lorsqu'une série de données se situe dans la plage de $[-1,+1]$ par exemple tandis qu'une autre est dans la plage $[-100,+100]$. Dans ce cas, l'échelle sera de -100 à $+100$ pour les deux séries.
- Des petits malins essaient de ruser en ajoutant quelque chose comme `min:-100,max:100` à la liste des valeurs de données pour tenter d'arrêter la mise à l'échelle automatique, mais cela ne fonctionne que si toutes les valeurs restent dans ces limites (**fig. 4**).
- Seul le papa du peloteur en série sait pourquoi l'échelle minimale est généralement de -6 à $+6$ mais parfois de -5 à $+5$.

Fonction sparadrap

Voici un petit sparadrap à insérer à vos sketches pour améliorer un peu la convivialité du traceur en série tout en gardant vos croquis lisibles :

```
void plot(String label, float value, bool last)
{
  Serial.print(label); // May be an empty string.
  if (label!="") Serial.print(":");
  Serial.print(value);
  if (last==false) Serial.print(",");
  else Serial.println();
}
```

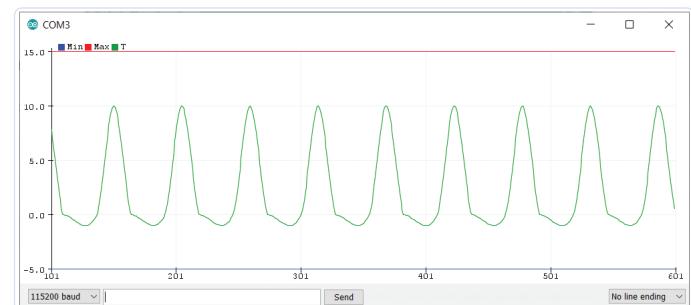


Figure 4. L'envoi de valeurs minimales et maximales constantes évite la mise à l'échelle automatique de l'axe vertical, mais seulement tant que toutes les autres valeurs restent dans ces limites.

Si le libellé de la chaîne est spécifié, cette fonction imprimera une paire `label:value`. Si `label` est une chaîne vide, alors elle n'imprimera que la valeur. Cela lui permet de sortir des données au format CSV pur dans le Serial Monitor, utilisable avec des tableurs.

Vous noterez que vous ne pouvez pas avoir le *Serial Monitor* et le *Serial Plotter* ouverts en même temps.

Après avoir ouvert le port série, appelez-le ainsi :

```
// Graph1, more graphs follow.
plot("Temperature",temperature_value,false);
// Graph2, more graphs follow.
plot("Humidity",humidity_value,false);
// Graph3, last graph.
plot("Pressure",pressure_value,true);
```

N'oubliez pas de fermer et rouvrir le traceur en série chaque fois que vous modifiez le croquis pour vous assurer que données et libellés périmés sont tous supprimés. En cas de doute, recommencez. 

200540-03

Votre avis, s'il vous plaît ?

Questions et commentaires sur cet article bienvenus par courriel à l'auteur clemens.valens@elektor.com ou à la redaction@elektor.fr

Ont contribué

Auteur : **Clemens Valens**
Rédaction : **C.J. Abate**

Maquette : **Giel Dols**
Traduction : **Rémy Bécarre**

LIEN

- [1] **Code source du traceur série de l'IDE Arduino :**
<https://github.com/arduino/Arduino/blob/master/app/src/processing/app/SerialPlotter.java>