



MyViz

Tutoriel de communication avec une carte Arduino

07/04/2021

Version 1.0

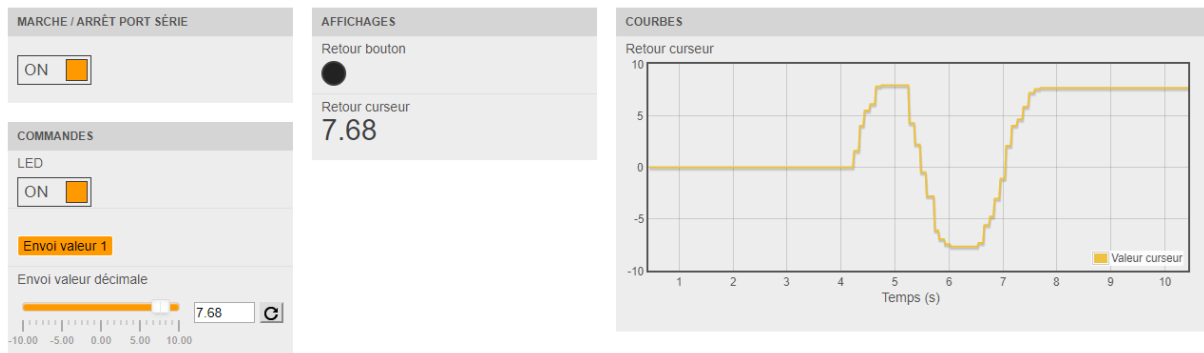


Table des matières

1 Introduction.....	3
2 Pré-requis.....	4
2.1 Installation de l'environnement de développement Arduino.....	4
2.1.1 Installation principale.....	4
2.1.2 Installation de la bibliothèque complémentaire StringUtil.....	4
2.1.3 Installation de la bibliothèque complémentaire MyViz.....	5
2.2 Installation de MyViz.....	6
3 Création d'un tableau de bord de communication sur port série avec une carte Arduino.....	7
4 Utilisation du tableau de bord.....	19
5 Support technique.....	20

1 Introduction

Ce document décrit un exemple simple d'utilisation de MyViz pour échanger de façon bidirectionnelle des données avec une carte Arduino via le port série de cette dernière.

Noter que seule la carte Arduino Uno a été testée, mais normalement toutes les autres devraient fonctionner sans problème.

2 Pré-requis

Ce tutoriel nécessite de réaliser au préalable un certain nombre de téléchargements et d'installations.

2.1 Installation de l'environnement de développement Arduino

Cette section présente les étapes à mettre en œuvre (une fois pour toute) pour mettre en place sur votre ordinateur l'environnement de programmation d'une carte Arduino.

2.1.1 Installation principale

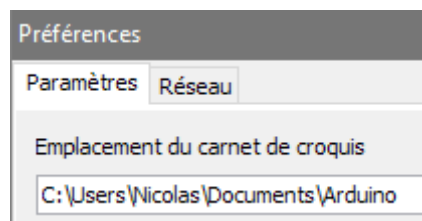
L'IDE Arduino en version supérieure ou égale à 1.8 doit tout d'abord être téléchargé (<http://arduino.cc/en/Main/Software>) et installé (<http://arduino.cc/en/Guide/HomePage>).

2.1.2 Installation de la bibliothèque complémentaire StringUtil

Cette bibliothèque apporte des utilitaires de gestion de chaînes de caractères permettant de contourner certaines limitations de la carte Arduino Uno.

Vous pouvez la télécharger à l'adresse suivante: <https://www.3sigma.fr/telechargements/StringUtil.zip>.

Une fois téléchargée, décompressez-là dans le répertoire des librairies de votre installation Arduino (typiquement, Documents\Arduino\libraries sur Windows). Pour identifier ce répertoire sur votre ordinateur, lancez l'IDE Arduino et sélectionnez **Fichier → Préférences**, l'information recherchée se trouve en haut de la fenêtre qui s'ouvre alors (« Emplacement du carnet de croquis »). Par exemple :



2.1.3 Installation de la bibliothèque complémentaire MyViz

Cette bibliothèque inclut les fonctions de communication via le port série avec MyViz.

Vous pouvez la télécharger à l'adresse suivante:

<https://www.3sigma.fr/telechargements/MyViz.zip>.

Une fois téléchargée, décompressez-là dans le répertoire des librairies de votre installation Arduino (voir les explications détaillées ci-dessus)

ATTENTION !

L'environnement Arduino doit être redémarré après l'installation d'une bibliothèque complémentaire.

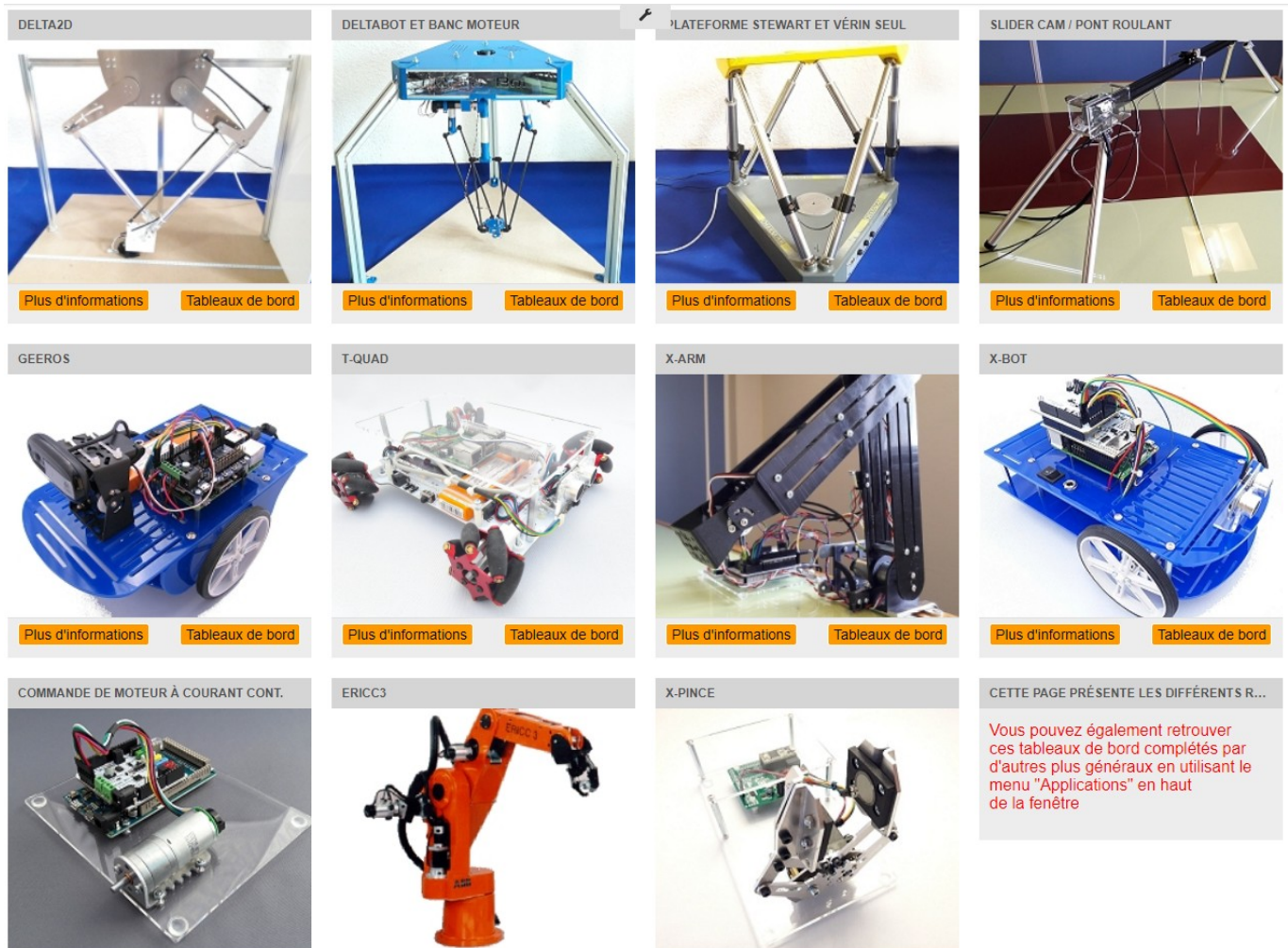
2.2 Installation de MyViz

MyViz se télécharge depuis la page suivante :

<https://www.3sigma.fr/myviz/MyViz.zip>

Il n'y a pas à proprement parler d'installateur. Il suffit de décompresser l'archive dans le répertoire de votre choix (mais si possible dans C: et jamais dans une arborescence profonde à cause de la limitation sur la longueur des chemins de fichiers dans Windows) et de lancer le programme **MyViz.exe**.

Le tableau de bord initialement affiché sera similaire à la capture d'écran ci-dessous :



Notez que si une ancienne version de MyViz (antérieure à la 0.8.5) a été utilisée précédemment, l'écran de démarrage ou le passage d'une application à l'autre pourraient ne pas fonctionner correctement. Dans ce cas, il faut supprimer le répertoire C:\Utilisateurs\nom_d_utilisateur\AppData\local\MyViz, ou équivalent.

Cette opération doit être faite une seule et unique fois pour un compte donné.

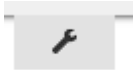
3 Création d'un tableau de bord de communication sur port série avec une carte Arduino

La version finale du tableau de bord que nous allons créer peut se télécharger à l'adresse suivante :

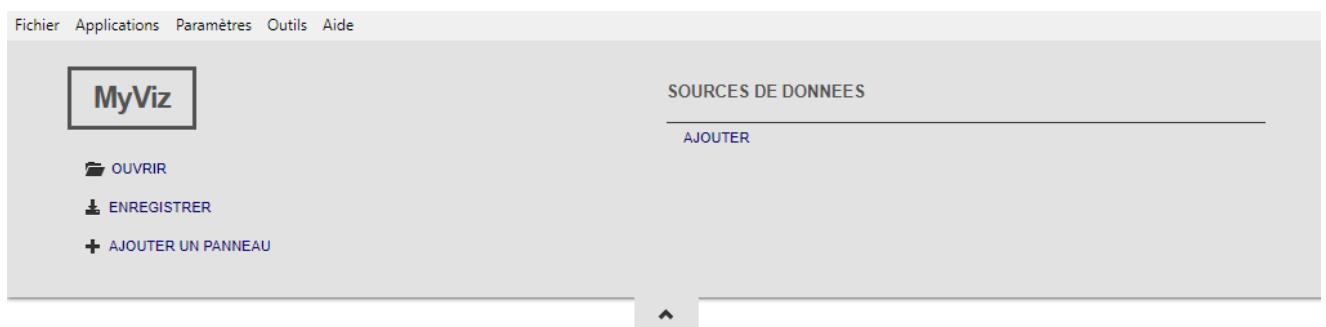
https://www.3sigma.fr/myviz/tutoriels/Tutoriel_PortSerie_1.zip

Sa création se fait en suivant scrupuleusement un certain nombre d'étapes :

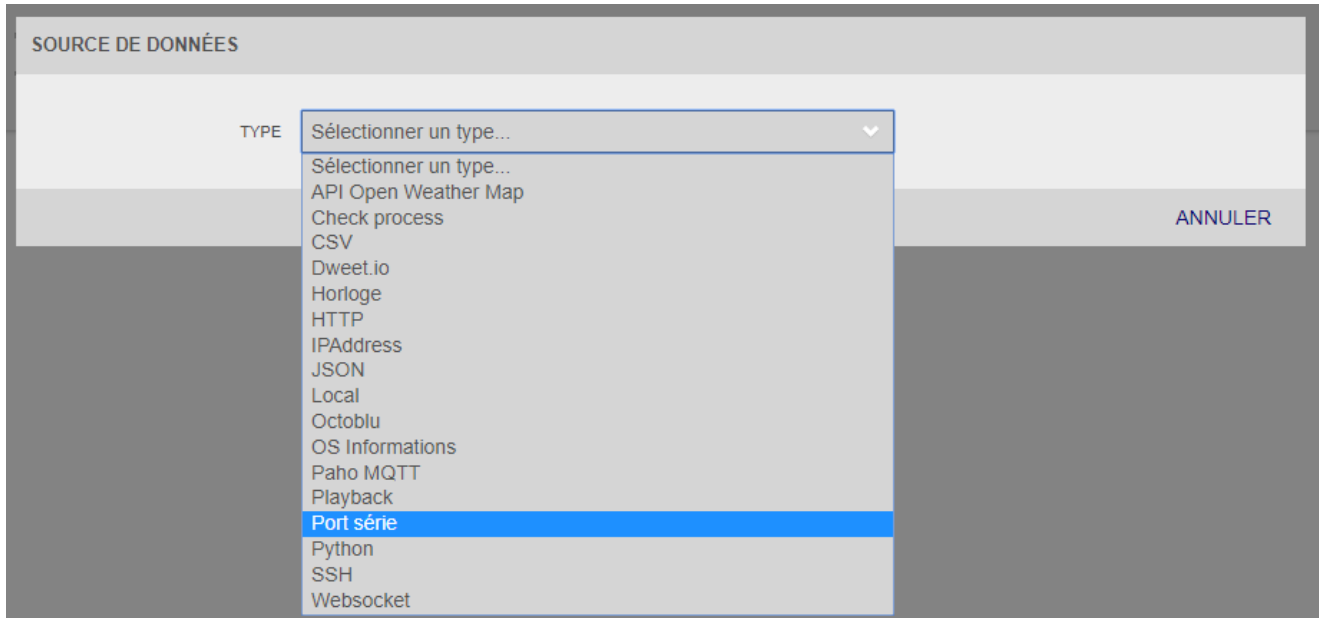
- Passer MyViz en mode développeur :
 - Fermer MyViz s'il est ouvert
 - Exécuter le programme « SwitchToDev.exe », situé dans le répertoire « MyVizFiles »
- Créer un tableau de bord vierge : Fichier → Nouveau
- Mettre la fenêtre de MyViz en plein écran et enregistrer ce tableau de bord vierge dans le répertoire de votre choix : Fichier → Enregistrer.
Remarque : l'enregistrement d'un tableau de bord ferme le mode « Édition ». Pour le réactiver, cliquer sur la clé située en haut du tableau de bord :



- Repasser en mode édition (en cliquant sur la clé, si ça n'a pas été fait à l'étape précédente). Ce mode déplie le haut du tableau de bord pour faire apparaître ceci :



- La première étape consiste à créer une source de données, qui sera la port série de la carte Arduino. Pour cela, cliquer sur le lien « Ajouter » (sous « Sources de données »). Ceci ouvre la boîte de dialogue suivante, invitant à choisir un type de source de données. Sélectionner « Port série » :



- La fenêtre de paramétrage du port série s'ouvre tout de suite après. Voici une capture du haut de cette fenêtre, avec les valeurs par défaut :

SOURCE DE DONNÉES

Flux de données temps-réel provenant du port série.

TYPE: Port série

NOM:

UTILISER LE PORT PAR DÉFAUT: ☒ YES

DÉFAUT: Détermine si on utilise le port par défaut, spécifié dans le menu Paramètres -> Port série par défaut

PORT: COM9 - Arduino LLC (www.arduino.cc)

NOMBRE DE BAUDS:

VARIABLES À LIRE:

Nom des variables à lire, séparées par des virgules

VARIABLES À ENVOYER:

Nom des variables à envoyer, séparées par des virgules

VARIABLES TO SEND 2:

Name of the second list of variables to send, separated by comma. If they exist, these variables and the ones from the first list will be sent alternatively

DELAY BEFORE SENDING FIRST DATA: 0 MILLISECONDES

Delay before sending first data. The dashboard must be reloaded if this delay is modified.

TAUX DE RAFFRAÎCHISSEMENT POUR L'ENVOI DES DONNÉES: 2000 MILLISECONDES

Taux de rafraîchissement pour l'envoi des données. Les données seront envoyées même si la valeur des contrôles auxquelles elles sont connectées sur le tableau de bord est inchangé. Le tableau de bord doit être rechargé si ce taux de rafraîchissement est modifié.

RECEIVE SEPARATOR:

Separator character between received values

END OF LINE RECEIVE SEPARATOR:

Separator character between lines of received values. If not set, a newline is used by default

- Modifier les paramètres suivants, permettant de définir une liaison série se nommant « sp », de vitesse 115200 baud, envoyant toutes les 100 ms les variables « LED, envoiBouton et envoiCurseur » séparées par une virgule avec l'ajout d'une somme de contrôle, recevant les variables « temps, lectureBouton et lectureCurseur » séparées par une virgule, avec un démarrage non immédiat (le démarrage se fera en utilisant un interrupteur, que nous allons définir plus loin) :
 - Nom : **sp**
 - Nombre de bauds : **115200**
 - Variables à lire : **temps,lectureBouton,lectureCurseur**
 - Variables à envoyer : **LED,envoiBouton,envoiCurseur**
 - Taux de rafraîchissement pour l'envoi des données : **100**
 - Receive separator : **,**
 - Méthode de somme de contrôle : **Somme**
 - Démarrage immédiat : **NO**
- Cliquer sur « Enregistrer » pour fermer la fenêtre de paramétrage du port série
- Enregistrer le tableau de bord (Fichier → Enregistrer). Ceci n'est pas rigoureusement nécessaire à cet instant précis mais c'est bien d'avoir le réflexe d'enregistrer régulièrement son travail.
Remarque : le raccourci clavier « CTRL S » n'est pas actif dans MyViz
- Nous allons maintenant ajouter des composants sur le tableau de bord. Ceci se fait en créant tout d'abord des panneaux, puis en intégrant dans ceux-ci des « widgets ».
Repasser le tableau de bord en mode « Édition » (en cliquant sur la clé, voir plus haut), si ce n'est pas déjà fait, et cliquer sur le lien « Ajouter un panneau » en bas à gauche de la zone d'édition :

MyViz

OUVRIR

ENREGISTRER

AJOUTER UN PANNEAU

SOURCES DE DONNEES

Nom	Dernière mise à jour	
sp	11:53:56	■ ↺ 🗑
AJOUTER		

- Ceci crée immédiatement un panneau vierge en haut à gauche de l'écran :



- Cliquer sur la clé en haut du panneau nouvellement créé pour accéder à ses propriétés et entrer le titre « Marche / arrêt port série » :

PANNEAU

TITRE

COLONNES

ENREGISTRER ANNULER

- Cliquer sur « Enregistrer ». Le titre apparaît maintenant sur ce panneau :



- Ajouter un widget à ce panneau en cliquant sur le « + », à gauche de la clé précédemment cliquée. Ceci ouvre la zone de dialogue de sélection de widget :

WIDGET

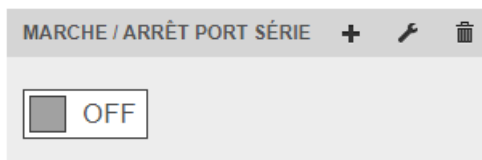
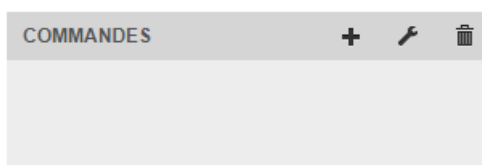
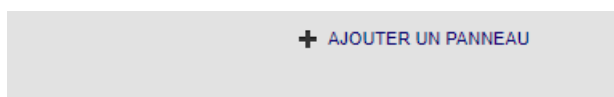
TYPE

ANNULER

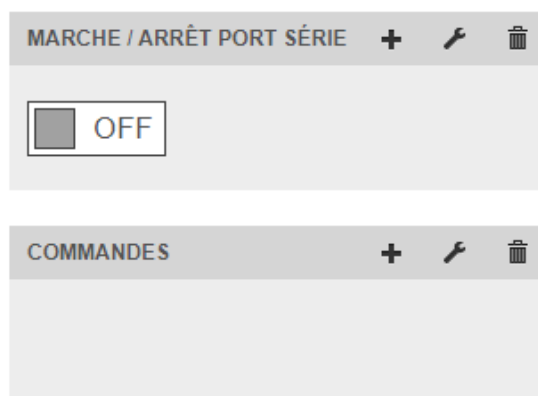
- Choisir « Interrupteur Port Série », ce qui ouvre immédiatement les propriétés de ce widget, dont voici un extrait :

The screenshot shows a configuration window titled 'WIDGET'. Inside, there's a subtitle 'Contrôle de type interrupteur pour les communications série'. The configuration includes several fields: 'TYPE' is a dropdown menu set to 'Interrupteur Port Série'; 'TITRE' is an empty text box; 'SOURCE DE DONNÉES SÉRIES' is an empty text box with a small explanatory text below it: 'Source de données correspondant au port série à commuter. Vous "devez" créer tout d'abord une source de données avec ce nom'; 'TEXTE CORRESPONDANT À "ON"' is a text box containing 'ON'; 'TEXTE CORRESPONDANT À "OFF"' is a text box containing 'OFF'; and 'ETAT INITIAL' is a checkbox labeled 'NO' which is currently unchecked.

- Ce widget permet d'ouvrir ou de fermer la communication série via un interrupteur. Modifier le paramètre suivant afin de lier cet interrupteur avec la source de données « Série » définie précédemment en lui donnant impérativement le même nom :
 - Source de données séries : **sp**
- Cliquer sur « Enregistrer » pour fermer la fenêtre de paramétrage de l'interrupteur port série
- Créer un nouveau panneau et définir son titre en « Commandes » :



- Glisser-déposer ce panneau pour le placer sous le panneau « Marche / arrêt port série » :



- Ajouter un widget à ce panneau « Commandes » en cliquant sur le « + » et choisir « Interrupteur », ce qui ouvre immédiatement les propriétés de ce widget, dont voici un extrait :

WIDGET

Contrôle de type interrupteur pour les communications série, socket ou http

TYPE	Interrupteur		
TITRE			
VARIABLE		+ DATASOURCE	JS EDITOR
+ ADD			
TEXTE CORRESPONDANT À "YES"	YES		
La valeur correspondante est définie ci-dessous			
VALEUR "YES"	1		
Valeur correspondant à la position "YES"			
TEXTE CORRESPONDANT À "NO"	NO		
La valeur correspondante est définie ci-dessous			
VALEUR "NO"	0		
Valeur correspondant à la position "NO"			

- Nous allons maintenant lier ce widget avec une des variables envoyées à la carte Arduino, en l'occurrence « LED ». Pour cela, cliquer sur le lien « Datasource » à droite du champ « Variable » :

VARIABLE `datasources[\"']` + DATASOURCE JS EDITOR

+ ADD

RESPONDANT À "YES"

VALEUR "YES"

RESPONDANT À "NO"

sp

- Cliquer sur « sp » pour sélectionner la source de données « Série », puis choisir « LED » dans la liste des variables :

VARIABLE `datasources[\"sp\"]` + DATASOURCE JS EDITOR

+ ADD

ONDANT À "YES"

EUR "YES"

ONDANT À "NO"

LED

_rawdata

envoiBouton

envoiCurseur

lectureBouton

lectureCurseur

- Ceci permet de lier l'interrupteur au premier champ des données envoyées à la carte Arduino :

WIDGET

Contrôle de type interrupteur pour les communications série, socket ou http

TYPE Interrupteur

TITRE

VARIABLE `datasources[\"sp\"][\"LED\"]` + DATASOURCE JS EDITOR

- Compléter le paramétrage de ce widget en modifiant ces autres propriétés :
 - Titre : **LED**
 - Texte correspondant à « Yes » : **ON**
 - Texte correspondant à « No » : **OFF**
 - État initial : **NO**
- Une fois paramétré, ce widget ressemble à ceci :



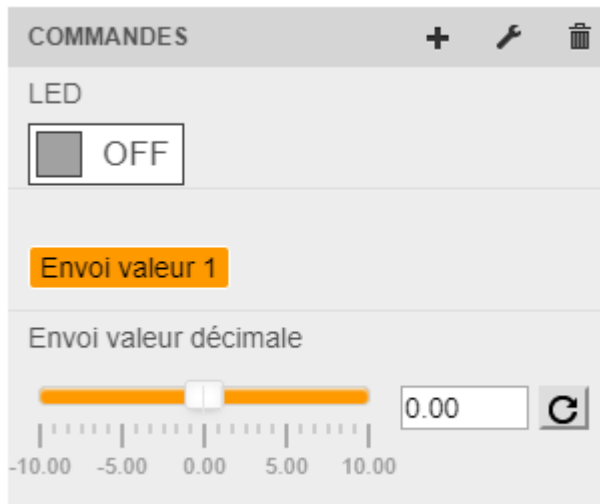
- Noter que pour modifier les propriétés d'un widget donné, il faut passer la souris en haut à droite du widget pour faire apparaître la clé spécifique à ce composant, puis cliquer dessus :



- En procédant comme précédemment, ajouter à ce panneau « Commandes » un widget de type « Bouton » et modifier les propriétés suivantes pour envoyer 1 au lieu de 0 dans le deuxième champs des données envoyées à la carte Arduino :
 - Variable : **datasources["sp"]["envoiBouton"]**
 - Texte sur le bouton : **Envoi valeur 1**
- Le panneau « Commandes » ressemble maintenant à ceci :

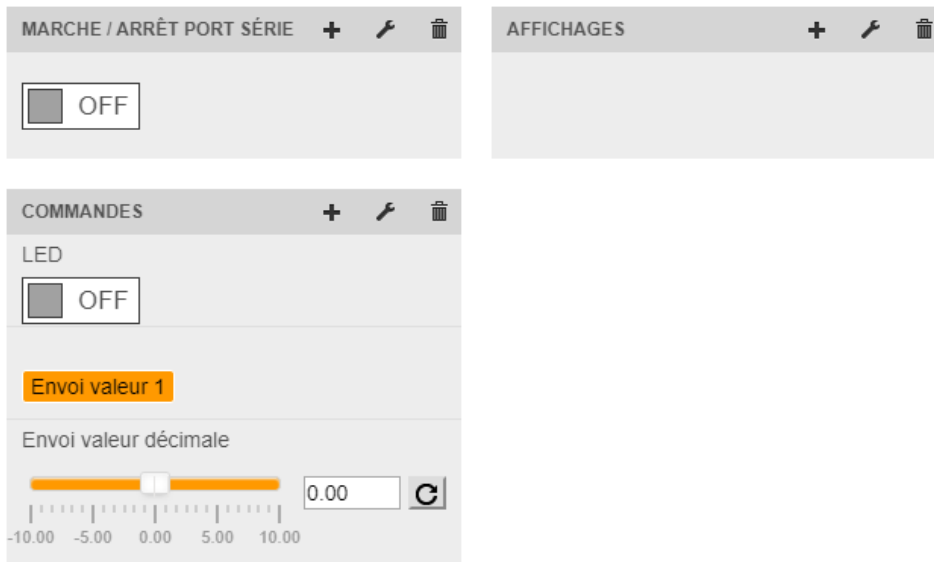


- Ajouter enfin un widget de type « Curseur » avec les propriétés suivantes :
 - Titre : **Envoi valeur décimale**
 - Variable : **datasources["sp"]["envoiCurseur"]**
 - Formule : **(100*x).toFixed(0)**
- Le panneau « Commande » intègre maintenant ce widget :

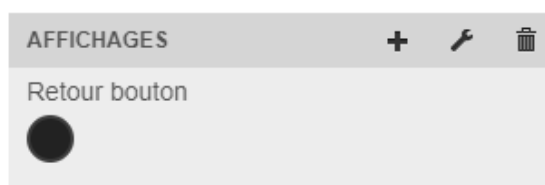


Il permet de placer dans le troisième champ des données envoyées à la carte Arduino la valeur du curseur multipliée par 100, entière (sans chiffre après la virgule), grâce à la formule **(100*x).toFixed(0)**. Ce point est très important : même s'il est possible de faire autrement, telle que la communication série est configurée entre MyViz et la carte Arduino dans ce tutoriel, il est impératif d'envoyer une suite d'entiers, sans partie décimale.

- Nous allons maintenant afficher le retour (passant par la carte Arduino) des valeurs envoyées par l'appui bouton et par le curseur. Pour cela, créer un nouveau panneau « Affichages » positionné à droite des deux autres, comme ceci :

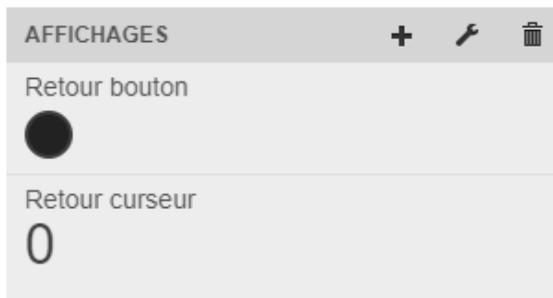


- Ajouter tout d'abord dans ce panneau un widget « Indicateur lumineux » possédant les propriétés suivantes :
 - Titre : **Retour bouton**
 - Valeur : `datasources["sp"]["lectureBouton"]`
- Ceci donne ce résultat :

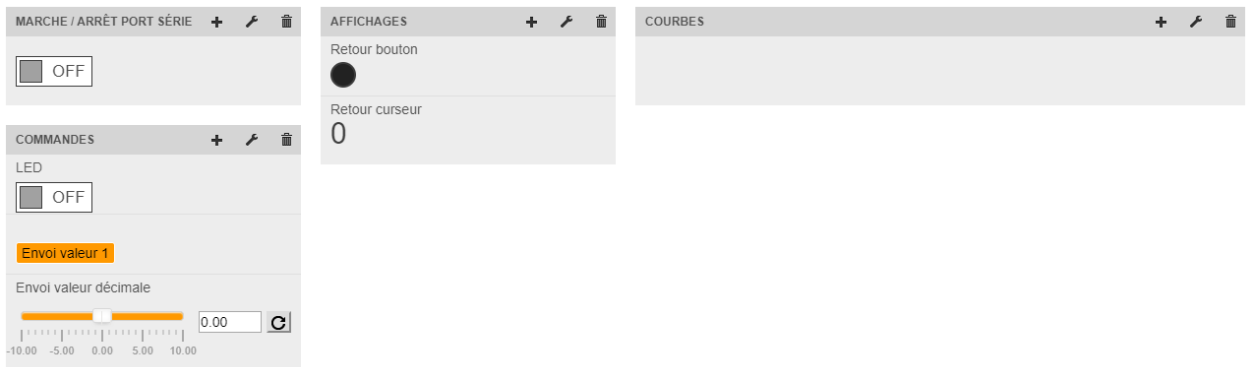


- Ajouter ensuite un widget de type « Texte » avec ces paramètres :
 - Titre : **Retour curseur**
 - Valeur : `datasources["sp"]["lectureCurseur"]`
 - Animer les changements de valeur : **NO**

- Ce nouveau panneau ressemble alors à ceci :

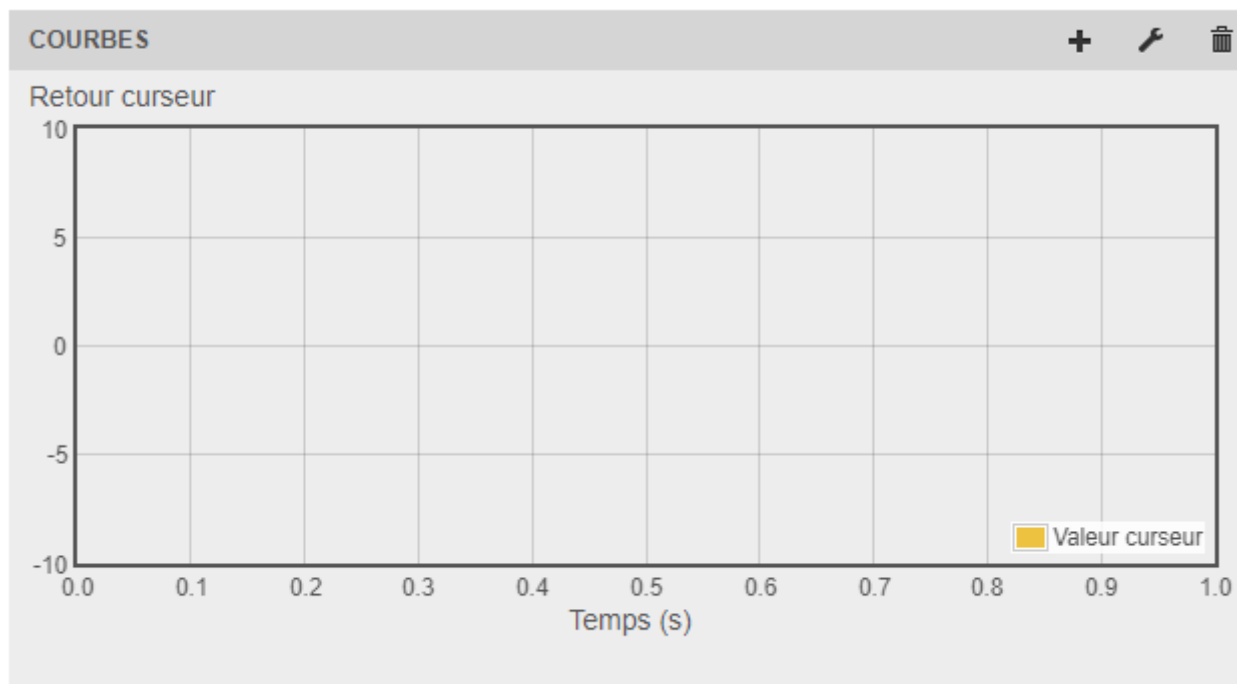


- Nous allons pour finir ajouter un dernier panneau, à droite des autres, d'une largeur de deux colonnes, afin d'afficher la valeur du curseur (lue depuis la carte Arduino) sur une courbe. Pour cela, en plus de définir le titre du panneau (« Courbes »), il faut spécifier le nombre de colonnes (2). Le résultat est le suivant :

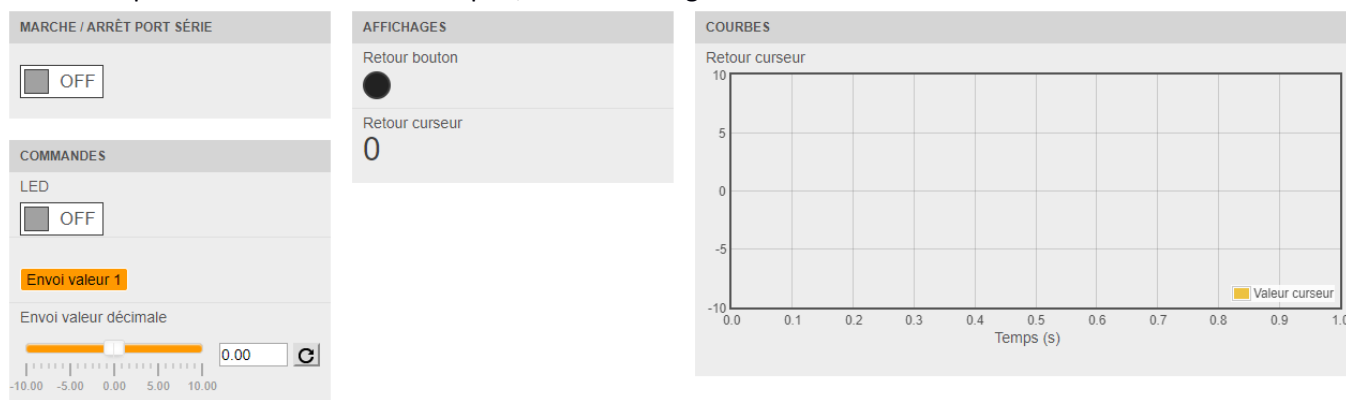


- Ajouter sur ce panneau un widget « Tracé » possédant ces propriétés :
 - Titre : **Retour curseur**
 - Axe x : **Colonne de la source de données**
 - Temps (axe x) : **datasources["sp"]["temps"]**
 - Légende sur l'axe des abscisses : **Temps (s)**
 - Valeur de l'axe y gauche : **datasources["sp"]["lectureCurseur"]**
 - Inclure la légende : **YES**
 - Légende : **Valeur curseur**
 - Hauteur en nombre de blocs : **5**

- Ceci donne pour ce panneau :



- Et pour le tableau de bord complet, une fois enregistré :



4 Utilisation du tableau de bord

La mise en œuvre du tableau de bord nécessite de réaliser les opérations suivantes :

- Connecter la carte Arduino sur l'ordinateur via son câble USB
- Ouvrir l'IDE Arduino et sélectionner la carte que vous souhaitez utiliser (par exemple : « Outils → Type de carte → Arduino Uno » ou « Outil → Type de carte → Arduino AVR Boards → Arduino Uno »)
- Choisir le port sur lequel est connectée la carte (« Outils → Port »)
- Ouvrir le programme Arduino contenant le premier exemple de communication série avec MyViz (Fichier → Exemples → MyViz → Example_SerialPort₁)
- Téléverser ce programme sur la carte Arduino
- Ouvrir MyViz et sélectionner le port série sur lequel est connectée la carte Arduino (Paramètres → Port série par défaut)
- Positionner sur « ON » l'interrupteur du panneau « Marche / arrêt port série ». Quand le tracé de courbe démarre, il est possible d'interagir avec les éléments de pilotage du tableau de bord



5 Support technique

Pour toute question ou problème, veuillez nous contacter à l'adresse support@3sigma.fr.