

ACTIVITE

Ecrire dans un fichier CSV

Qu'est-ce qu'un fichier CSV

Un fichier CSV (Comma-Separated Values) est un format de fichier texte simple utilisé pour stocker des données. Voici quelques caractéristiques clés des fichiers CSV :

Structure : Les données sont organisées en lignes et en colonnes. Chaque ligne représente un enregistrement, et chaque colonne représente un champ ou une valeur de cet enregistrement.

Séparateur : Les valeurs sont généralement séparées par des virgules, des points-virgules ou des tabulations.

En-tête : La première ligne d'un fichier CSV contient souvent les noms des colonnes, qui servent d'en-têtes pour les données.

Utilisation : Ils sont couramment utilisés pour l'importation et l'exportation de données entre différents systèmes, car ils sont faciles à lire et à écrire.

Exemple simple du contenu d'un fichier CSV :

```
Nom,Âge,Ville
Alice,30,Paris
Bob,25,Lyon
Charlie,35,Marseille
```

Récupération des données

Cas pratique :

Un capteur (potentiomètre) connecté à une carte Arduino envoie toutes les 500 ms la valeur du potentiomètre vers Node-Red en utilisant la liaison série.

✍ 1 – Concevez un programme qui envoie sur la liaison série la valeur du potentiomètre toutes les 500 ms. Vous transmettez que la valeur numérique comprise entre 0 et 1023, pas de texte ou nom de la variable.

Votre programme :

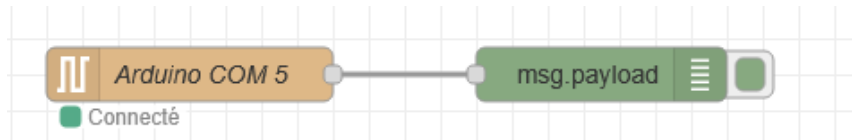
```
// ----- Programme 1 - Test communication -----
//----- liste des includes -----
//----- liste des define -----
#define POTA A0
//----- liste des constantes -----
//----- liste des variables globales -----
int valeur;
//----- liste des fonctions -----
// ----- SETUP -----
void setup()
{
    Serial.begin(9600); // Initialise la communication série à 9600 bauds
    pinMode(POTA, INPUT);
}
// ----- LOOP -----
void loop()
{
    valeur = analogRead(POTA); // Lecture de la valeur du potentiomètre
```

```

Serial.println(valeur); // Envoie la valeur de i via la communication
série
delay(500); // Attente de 500 ms
}

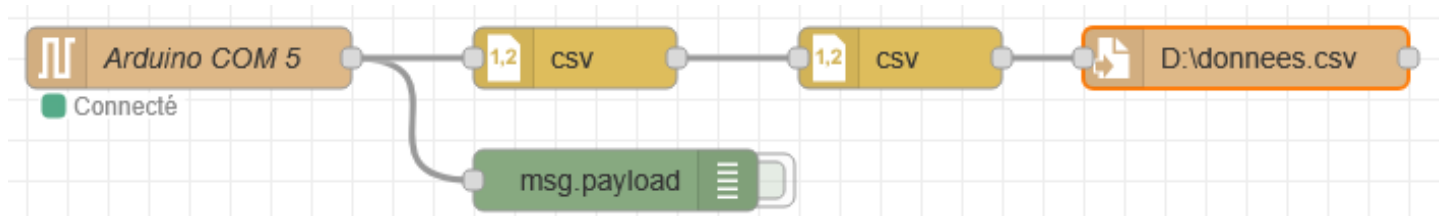
```

2 – Récupérez sur Node-Red le flux présent sur la liaison série en provenance de la carte Arduino. Vous utiliserez un nœud "debug" pour afficher le flux de la liaison série dans le débogueur.



Ecriture dans un fichier CSV

3 – Utilisez la structure suivante pour préparer les données, puis les transmettre au fichier CSV



Configuration :

1 ^{er} nœud CSV	2 ^{ème} nœud CSV
Modifier le nœud csv Supprimer Annuler Terminer Propriétés Colonnes: a Séparateur: point-virgule Nom: Nom Options CSV vers objet Input: Passer en premier 0 lignes ☐ La première ligne contient les noms des col ☒ Analyser les valeurs numériques ☐ Inclure les chaînes vides ☒ Inclure les valeurs nulles Sortie: un message par ligne Options Objet vers CSV Sortie: ne jamais envoyer d'en-têtes de colon Nouvelle ligne: Windows (\r\n)	Modifier le nœud csv Supprimer Annuler Terminer Propriétés Colonnes: a Séparateur: virgule Nom: Nom Options CSV vers objet Input: Passer en premier 0 lignes ☒ La première ligne contient les noms des co ☒ Analyser les valeurs numériques ☐ Inclure les chaînes vides ☐ Inclure les valeurs nulles Sortie: un message par ligne Options Objet vers CSV Sortie: ne jamais envoyer d'en-têtes de colon Nouvelle ligne: Windows (\r\n)

Nœud "Ecrire le fichier"

Modifier le nœud Écrire le fichier

Supprimer

Annuler

Terminer

Propriétés

Nom du fichier

chemin D:\donnees.csv

Action

ajouter au fichier

☐ Ajouter une nouvelle ligne (\n) à chaque charge ?

☐ Créer un répertoire s'il n'existe pas ?

Encodage

Par défaut

Nom

Astuce : Le nom du fichier doit être un chemin absolu, sinon il sera relatif au répertoire de travail du processus Node-RED.

Données multiples

En général on ne transmet pas qu'une seule valeur mais plusieurs. Par exemple, la valeur du potentiomètre, l'état d'un interrupteur, la valeur d'un capteur de température, ...

- ✍ 4 – Modifiez votre programme Arduino pour transmettre trois valeurs en même temps sur la liaison série :
- Valeur du potentiomètre.
 - Etat d'un interrupteur 0 (ouvert) ou 1 (fermé).
 - Valeur aléatoire comprise entre 0 et 40 qui représente par exemple, la valeur mesurée par un capteur de température.

Format du message envoyé (chaîne de caractère) : valeurPotentiometre;etatBouton;valeurAleatoire

Votre programme :

```
// ----- Programme 2 - Envoi multiple -----
//----- liste des includes -----
//----- liste des define -----
#define POTA A0
#define BOUTON 4
//----- liste des constantes -----
//----- liste des variables globales -----
int valeur;
int valeurAlea = 0;
bool etatBouton = false;
char resultat[50];

//----- liste des fonctions -----
// ----- SETUP -----
void setup()
{
```

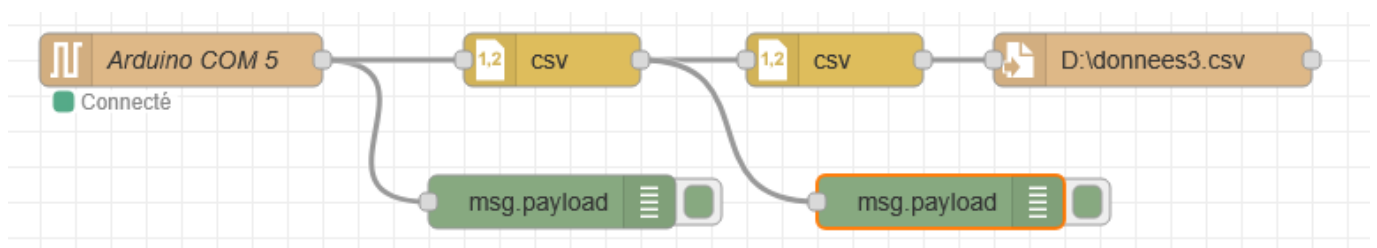
```

Serial.begin(9600); // Initialise la communication série à 9600 bauds
pinMode(POTA,INPUT);
pinMode(BOUTON,INPUT);
}
// ----- LOOP -----
void loop()
{
    valeur = analogRead(POTA); // Lecture de la valeur du potentiomètre
    etatBouton = digitalRead(BOUTON); // Lecture de l'état du bouton
    valeurAlea = random(0, 41); //valeur aléatoire comprise entre 0 et 40
    sprintf(resultat, "%d;%d;%d", valeur, etatBouton, valeurAlea); // constitution
    du message
    Serial.println(resultat); // Envoie du message via la liaison série
    delay(500); // Attente de 500 ms
}

```

5 – Récupérez sur node-red le flux présent sur la liaison série en provenance de la carte Arduino. Sauvegardez les dans un fichier CSV.

Vous pourrez utiliser les nœuds CSV pour réorganiser les valeurs et les sauvegarder correctement dans le fichier CSV.



Configuration :

1 ^{er} nœud CSV	2 ^{ème} nœud CSV
Modifier le nœud csv Supprimer Annuler Terminer Propriétés Colonnes: valeurPota,etatBoutonb,valeurAleatoire Séparateur: point-virgule Nom: Nom Options CSV vers objet Input: Passer en premier 0 lignes ☐ La première ligne contient les noms des colonnes ☒ Analyser les valeurs numériques ☐ Inclure les chaînes vides ☒ Inclure les valeurs nulles Sortie: un message par ligne Options Objet vers CSV Sortie: ne jamais envoyer d'en-têtes de colonnes Nouvelle ligne: Windows (\r\n)	Modifier le nœud csv Supprimer Annuler Terminer Propriétés Colonnes: valeurPota,etatBoutonb,valeurAleatoire Séparateur: point-virgule Nom: Nom Options CSV vers objet Input: Passer en premier 0 lignes ☒ La première ligne contient les noms des colonnes ☒ Analyser les valeurs numériques ☐ Inclure les chaînes vides ☐ Inclure les valeurs nulles Sortie: un message par ligne Options Objet vers CSV Sortie: ne jamais envoyer d'en-têtes de colonnes Nouvelle ligne: Windows (\r\n)

Modifier le noeud Écrire le fichier

Supprimer Annuler

Propriétés

Nom du fichier:

Action:

☐ Ajouter une nouvelle ligne (\n) à chaque charge

☐ Créer un répertoire s'il n'existe pas ?

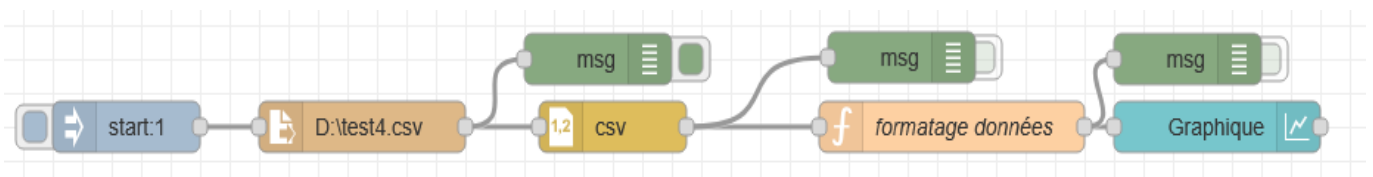
Encodage:

ATTENTION

Surveillez la taille du fichier de données, déconnectez l'Arduino lorsque vous ne faites pas de mesures pour éviter d'ajouter des lignes inutiles à votre fichier CSV.

Affichage du contenu d'un fichier CSV

6 – Faites un nouveau schéma Node-Red pour lire et afficher le contenu d'un fichier CSV.



L'injecteur "start" permet d'activer la lecture du fichier "tesr4.csv".

Le nœud "CSV" permet de récupérer les données au format CSV, n'oubliez pas de sélectionner le type de séparateur utilisé par le fichier CSV : virgule, point-virgule, ...

Le nœud "formatage données" permet de formater les données (matrice [x], [y]) qui sera transmise au nœud "graphique" pour affichage.

Modifier le noeud Lire le fichier

Supprimer

Propriétés

Nom du fichier:

Sortie:

Encodage:

Nom:

Modifier le noeud csv

Supprimer Annuler

Propriétés

Colonnes:

Séparateur:

Nom:

Options CSV vers objet

Input:

☒ La première ligne contient les noms de colonnes

☒ Analyser les valeurs numériques

☐ Inclure les chaînes vides

☐ Inclure les valeurs nulles

Sortie:

Nœud "formatage données" : (onglet message reçu)

```
let payload=msg.payload;
let data = [[]];

for(let i=0;i<payload.length;i++){
  data[0].push({ "x": payload[i].TIME, "y": payload[i].IA});
}

let labels = msg.columns.split(",").slice(1);
msg.payload=[{
  "series": labels,
  "data":data,
  "labels": labels
}]];
return msg;
```

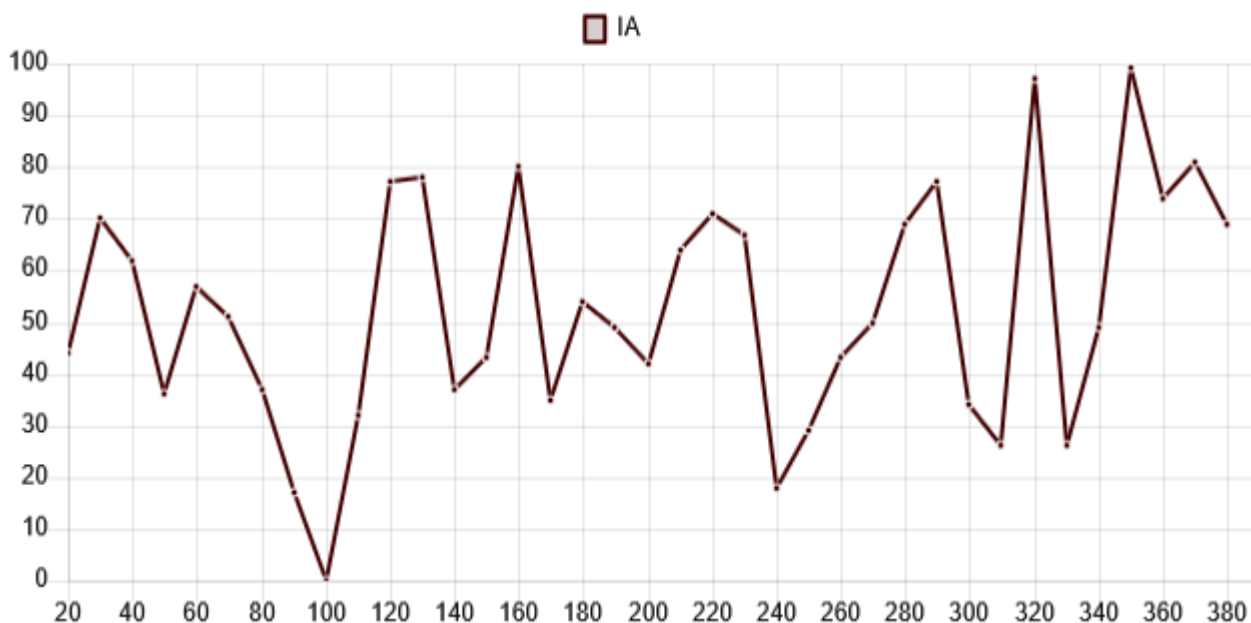
Récupération des coordonnées x et y pour chaque ligne du tableau envoyé par le nœud "CSV".

Tableau = payload
i = indice de ligne
payload.length = taille du tableau

x= payload[i].TIME
y= payload[i].IA

data[] = tableau de coordonnées (x,y) transmis au nœud "Graphique"

Graphique



Affichage multiple

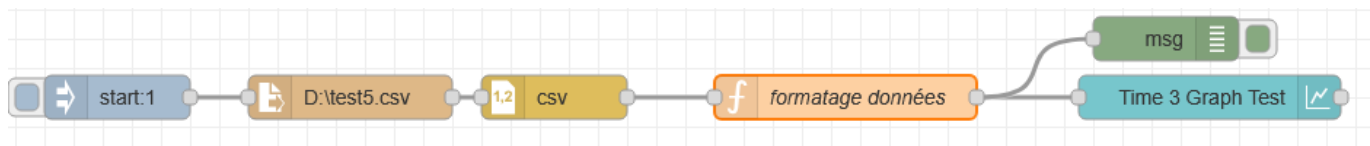
Cette fois ci, on souhaite afficher plusieurs courbes, IA(t), VA(t) et IB(t) dont les données sont enregistrées dans le fichier CSV "test5.CSV".

Exemple du contenu du fichier CSV :

	A
1	TIME,IA,VA,IB
2	20,44,120,60
3	30,7,150,40
4	40,62,130,25
5	50,36,126,30
6	60,57,140,53
7	70,51,90,60

Attention :

Dans cet exemple on utilise la virgule pour séparer les données. Il faudra vérifier que le nœud CSV est bien configuré avec l'option "virgule" pour le séparateur.



Nœud "formatage données" : (onglet message reçu)

```

let payload=msg.payload;
let data = [[],[],[[]];
for(let i=0;i<payload.length;i++){
  data[0].push({"x": payload[i].TIME, "y": payload[i].IA});
  data[1].push({"x": payload[i].TIME, "y": payload[i].VA});
  data[2].push({"x": payload[i].TIME, "y": payload[i].IB});
}
let labels = msg.columns.split(",").slice(1);
msg.payload=[{
  "series": labels,
  "data":data,
  "labels": labels
}];
return msg;

```

Modifier le noeud csv

Supprimer

Annuler

Propriétés

Colonnes

TIME,IA,VA,IB

Séparateur

virgule

Nom

Nom

Options CSV vers objet

Input

Passer en premier

0

lignes

☒ La première ligne contient les non
 ☒ Analyser les valeurs numériques
 ☐ Inclure les chaînes vides
 ☐ Inclure les valeurs nulles

Modifier le noeud chart

Supprimer

Annuler

Propriétés

Group

[Home] Default

Size

auto

Label

Time 3 Graph Test

Type

Line chart

enlarge po

X-axis

last

1

hours

OR

1000

pc

X-axis Label

x

ε

Y-axis

min

max

Legend

Show

Interpolate

linear

Series Colours

