Termux, Termux:API & Node-RED

Termux

Sources

- https://termux.dev/en/
- https://wiki.termux.com/wiki/Main_Page

Termux est un émulateur de terminal Android et une application d'environnement Linux qui fonctionne directement, sans les droits root. Un système de base minimal est installé automatiquement, des paquets supplémentaires sont disponibles en utilisant le gestionnaire de paquets.

Installation du paquet termux

- Télécharger le module complémentaire Termux depuis F-Droid

Termux possède quelques fonctionnalités supplémentaires. Vous pouvez les ajouter en installant des modules complémentaires :

Termux:API
: Access Android.

Termux:Boot
: Run script(s) when your device boots.

…….

Termux:API

Installation du paquet termux-api

- Télécharger le module complémentaire Termux:API depuis F-Droid
- au départ de termux , utiliser la commande : pkg install termux-api

Important : Ne pas mélanger les installations de Termux et Addons entre Google Play et F-Droid.

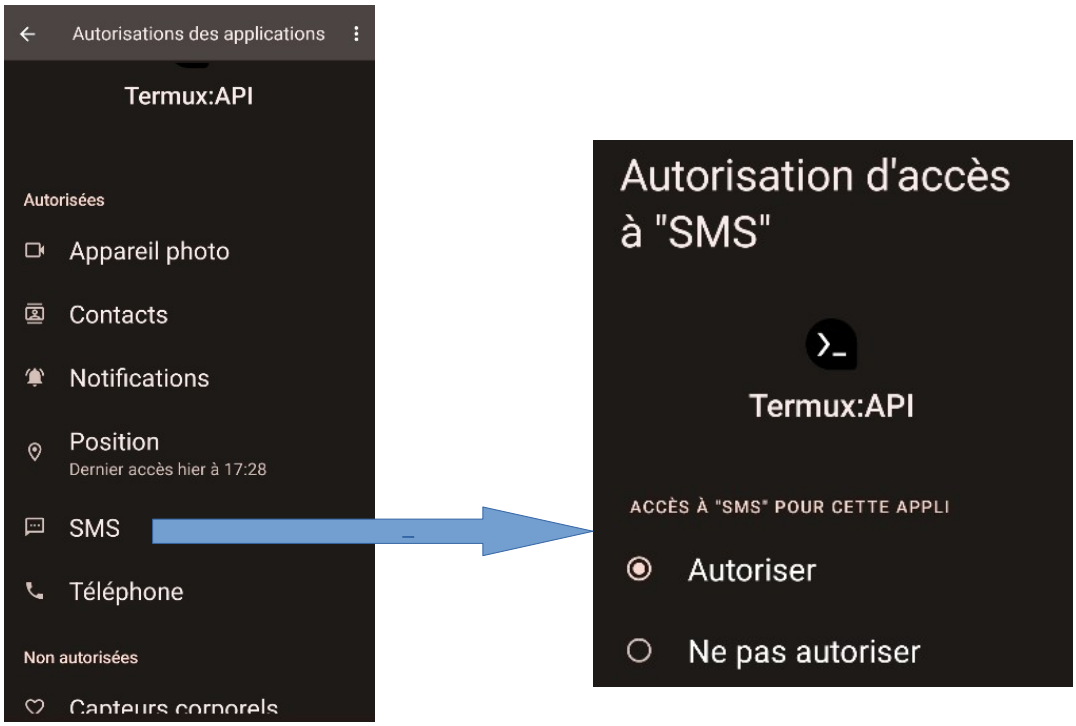
Implémentations actuelles de l'API

termux-battery-status
    Get the status of the device battery.
……..
termux-sensor
    Get information about types of sensors as well as live data.
……..
termux-sms-send
    Send a SMS message to the specified recipient number(s).

!!! Activer les autorisations des API !!!

pour utiliser ces implémentations, il est nécessaire d'activer les autorisations d'accès à SMS dans le smartphone !

Paramètres > Applications > Termux:API > Autorisations >



Node-RED sur Android

Source : https://nodered.org/docs/getting-started/android

Installation via Termux

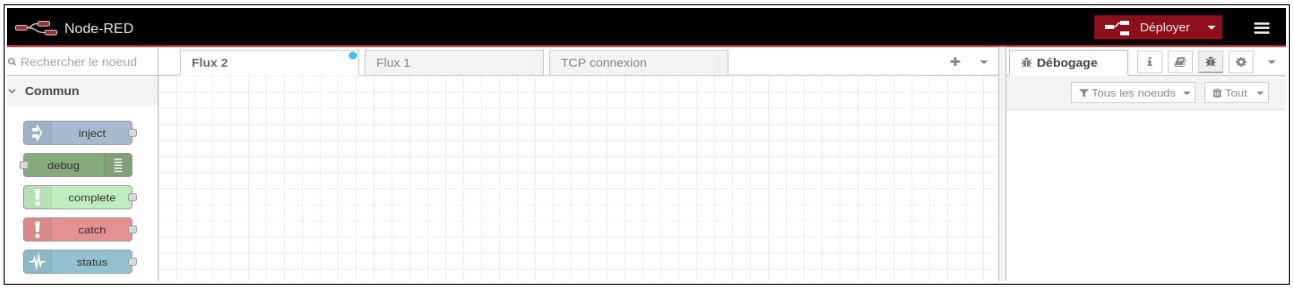
```
apt update
apt upgrade
apt install coreutils nano nodejs
npm i -g --unsafe-perm node-red
node-red
```

Visualiser l’éditeur de Node-RED sur un PC

Récupérer l’IP du SmartPhone

Ouvrir le navigateur d’un PC connecté au même réseau WiFi du SmartPhone

Taper le lien : http://IP_Android:1880

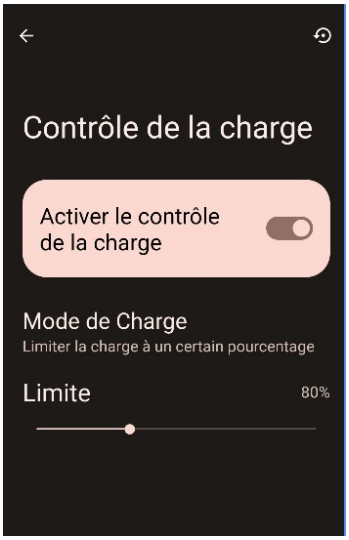


Cas pratiques :

1) Gérer la charge d’un Smartphone S10e

Comment activer le contrôle de la charge

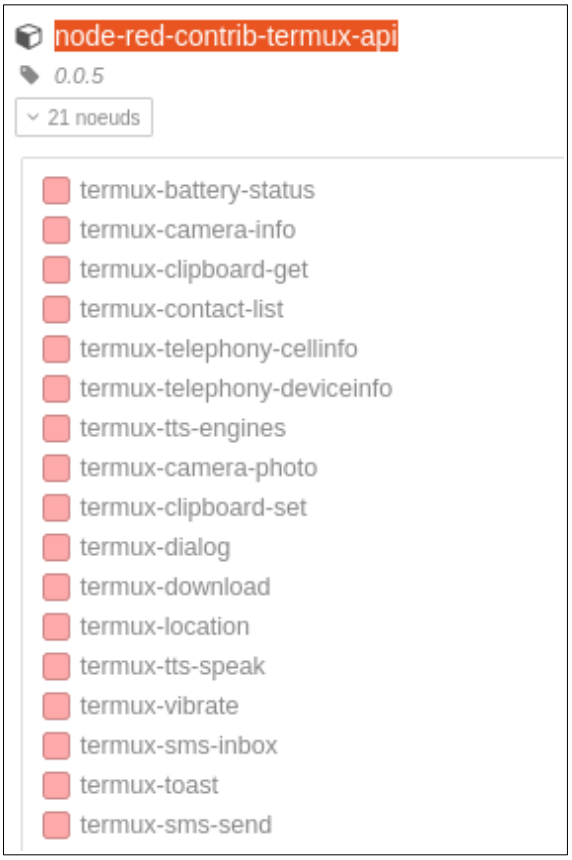
Paramètres > batterie > contrôle de la charge



Au dela de ce seuil, le courant de charge mesuré à la sortie du chargeur est nul !
Le courant reste nul jusqu’à ce que le % restant descend à 69 %, ensuite le courant est rétabli !

2) Envoyer un SMS sur son smartphone en cas de panne de courant à son domicile

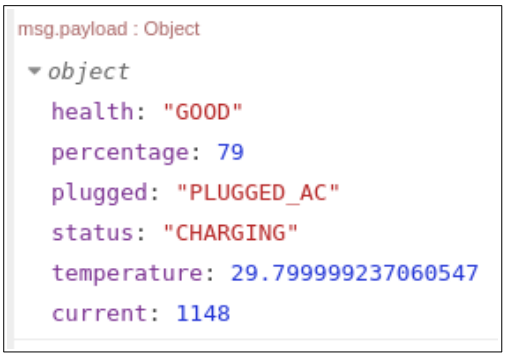
Dans Node-RED, installer la palette « node-red-contrib-termux-api »



Limite à l'utilisation de la propriété « msg.payload.plugged » et du contrôle de charge d'un smartphone !!

<table>
<tr><td colspan="2">Dans Node-RED on observe :</td></tr>
<tr><td>

- un courant de charge de 1148 mA à 79 %

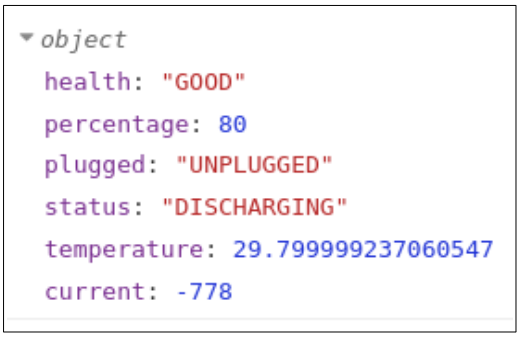


- la propriété msg.payload.plugged

  affiche « PLUGGED_AC »

</td><td>

- un courant négatif -778 mA à 80 %

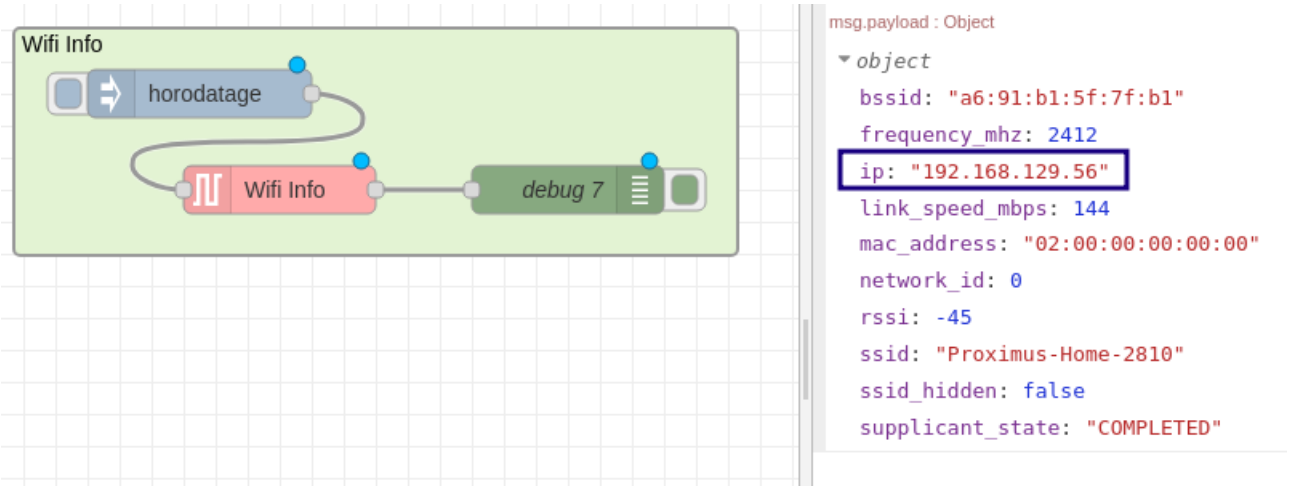


- la propriété msg.payload.plugged

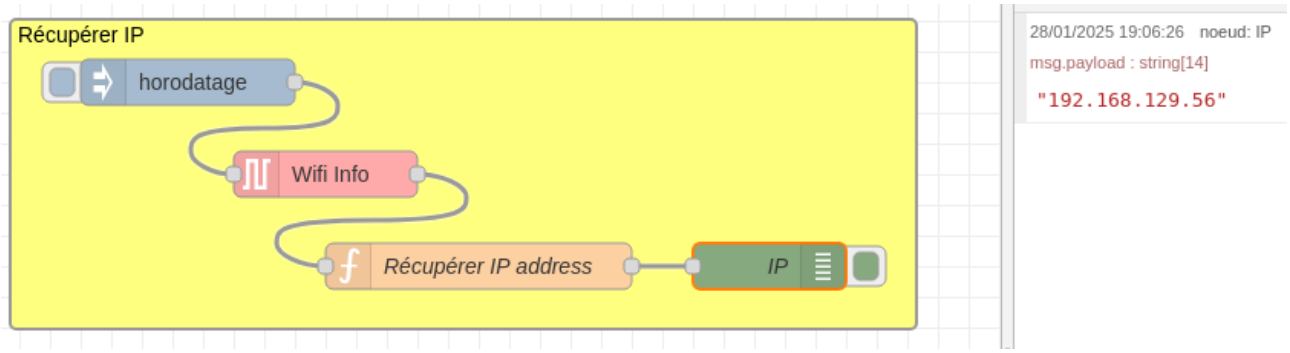
  affiche « UNPLUGGED »

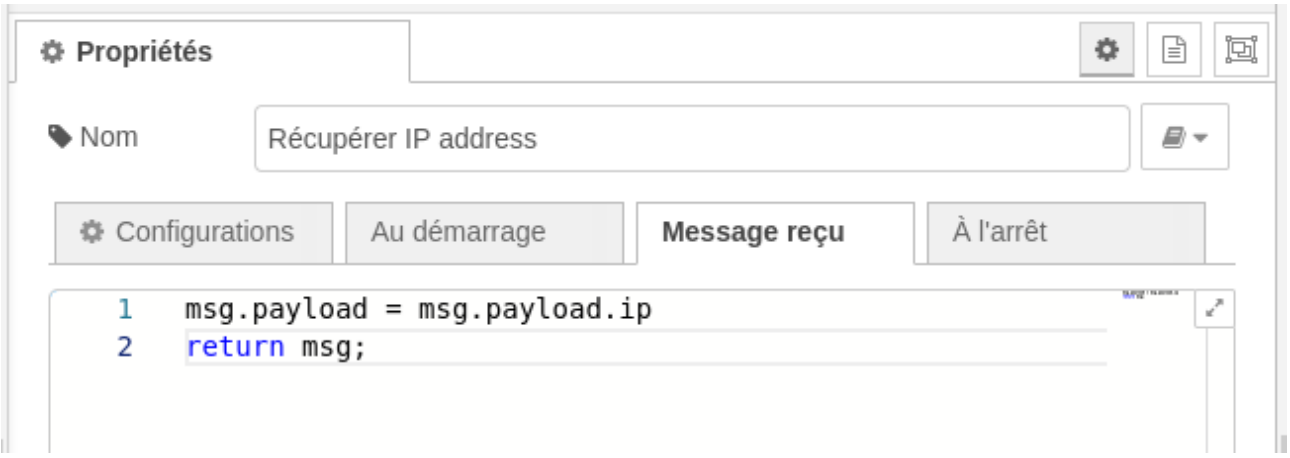
</td></tr>
<tr><td colspan="2">La propriété « msg.payload.plugged » utilisée dans le flow Node-RED ne permet pas à elle seule de distinguer une panne de courant d'un dépassement de seuil !!</td></tr>
</table>

Utilisation de la propriété « Wifi Info » indépendante de la valeur du courant lors d'un contrôle de charge



Récupérer l'adresse IP du smartphone





Utilisation de « exec » & « termux-sms-send »

termux-sms-send

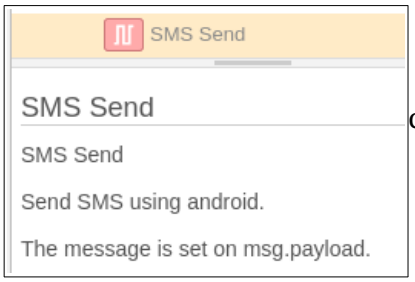
Send a SMS message to the specified recipient number(s).

Usage

```
termux-sms-send -n number[,number2,number3,...][text]
```

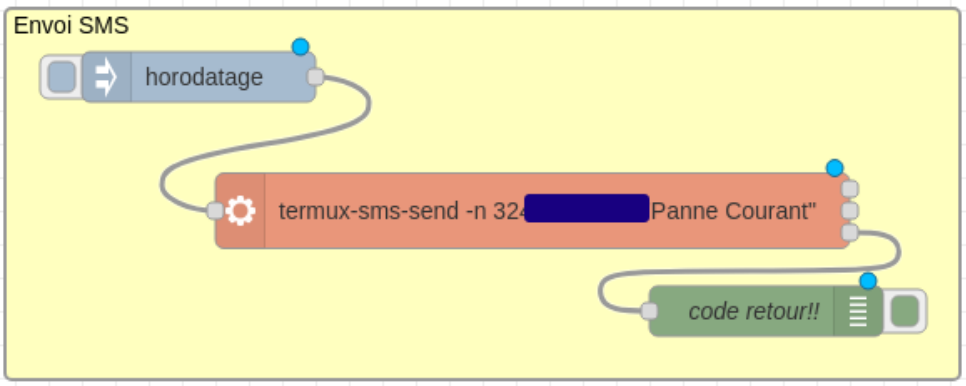
The text to send is either supplied as arguments or read from stdin if no arguments are given.

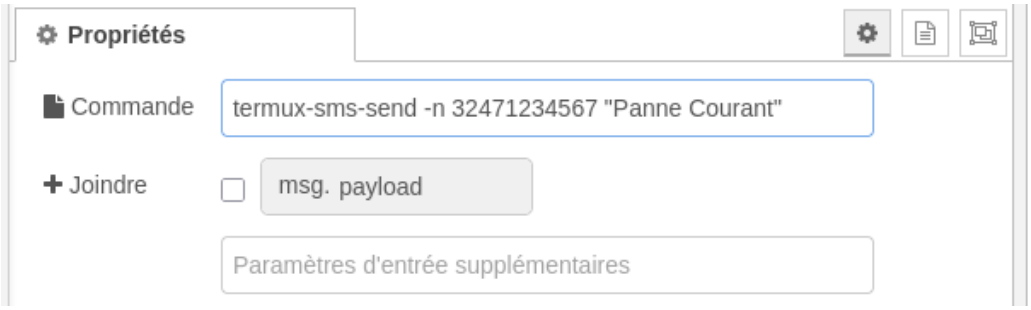
Pourquoi utiliser « termux-sms-send » de Termux-API, au lieu de « sms-send » de la palette «node-red-contrib-termux-api» ??



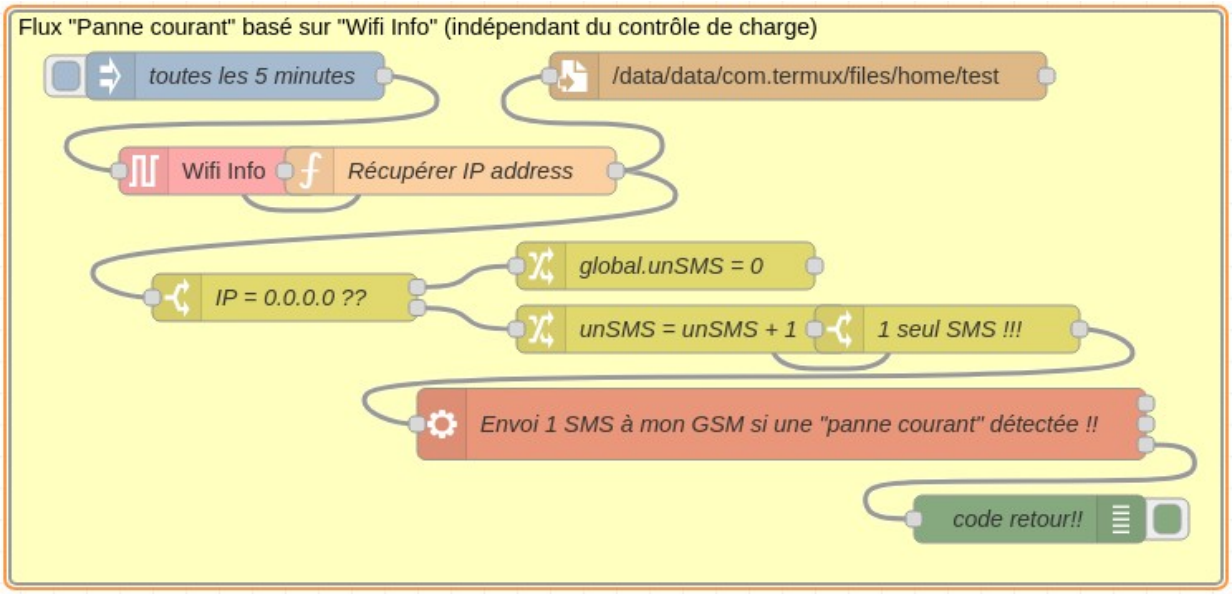
La documentation de « sms-send » ne précise pas comment l'utiliser….

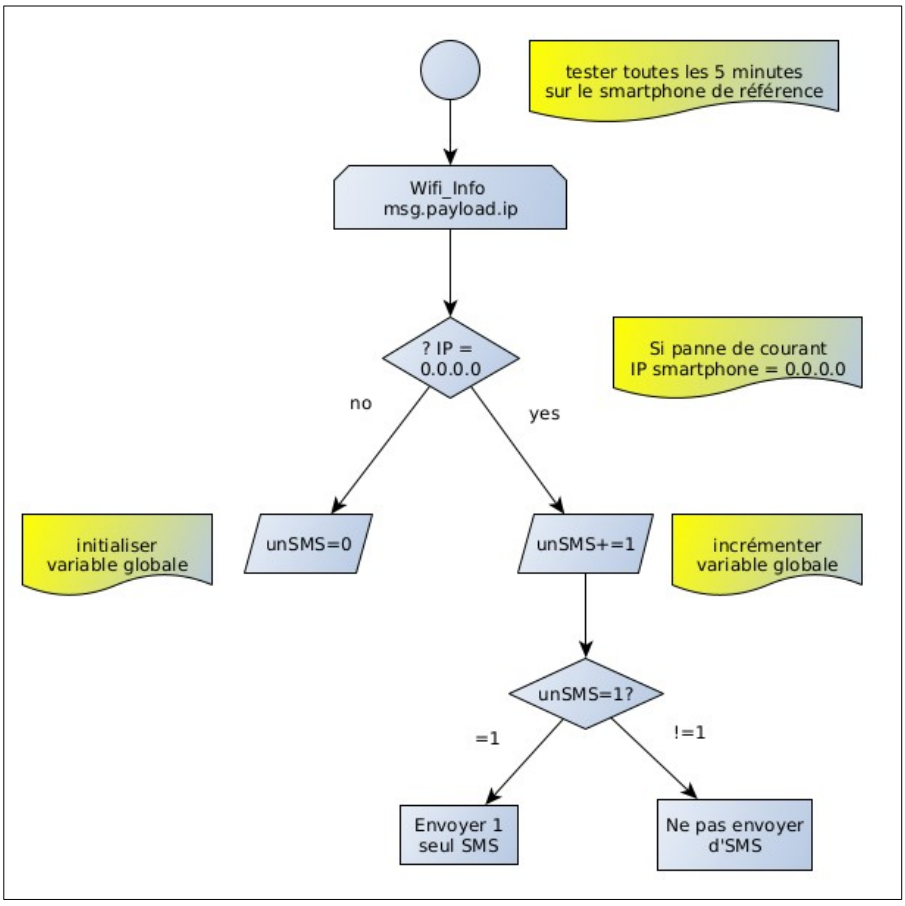
Utilisation termux-send-sms





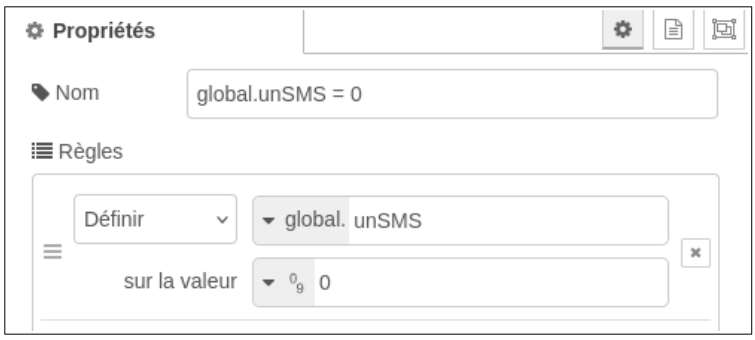
Utilisation conjointe de « Wifi Info » et « termux-send-sms » pour détecter une panne de courant et envoyer un seul SMS



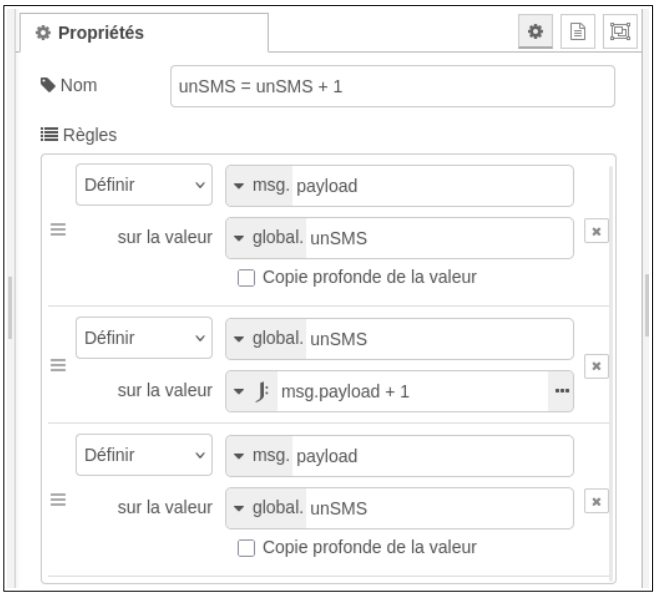


Modifier le nœud switch

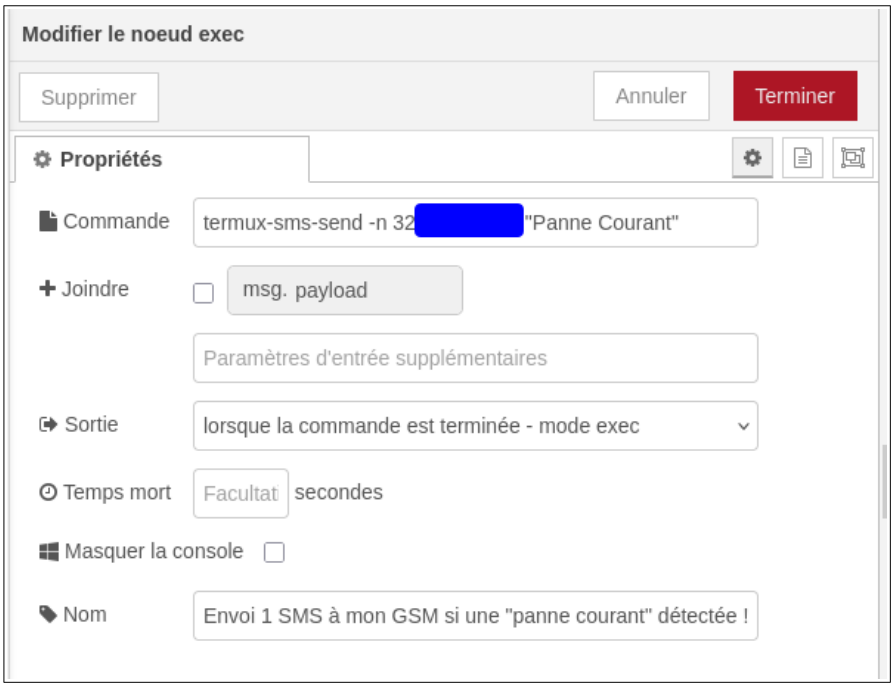
Modifier le nœud change



Modifier le nœud change



Modifier le nœud switch



3) Envoyer un SMS à son smartphone lors d'une panne de courant et lors du rétablissement de celui-ci

Utilisation d'un fichier alimenté ttes les 5 minutes par l' IP smartphone pour envoyer un seul SMS en cas de panne et un SMS lors du rétablissement du courant

Via script bash créé dans le smartphone

….

ne faire ce test que si le fichier « test » existe et qu'il contient >=2 lignes

d=$(tail -n1 fic_ip | cut -d'.' -f4) # recupere derniere ligne du fichier fic_ip et la derniere colonne

ad=$(tail -n2 fic_ip | head -n1 | cut -d'.' -f4) recupere avant derniere ligne du fichier fic_ip et la derniere colonne

echo $(($ad - $d)) # calcule la différence

si le résultat est positif ⇒ panne courant

si résultat nul ⇒ ne rien faire

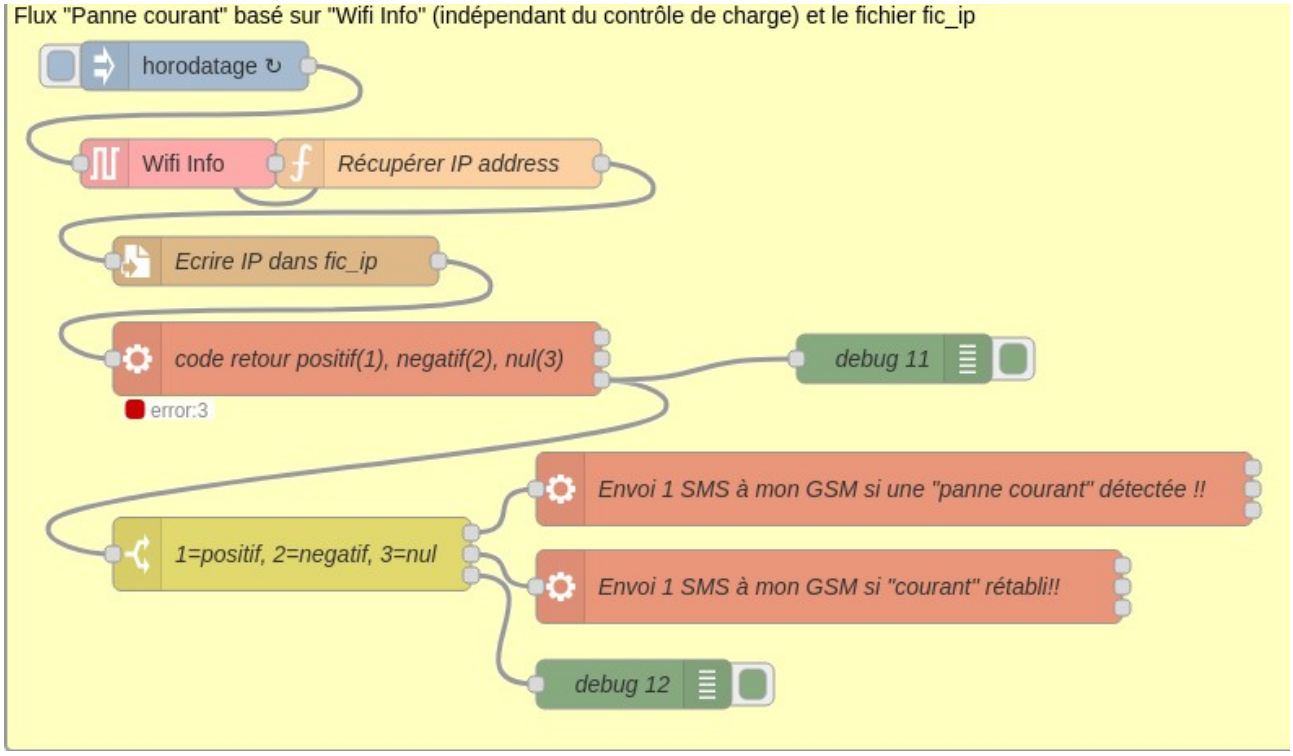
si résultat négatif ⇒ courant rétabli

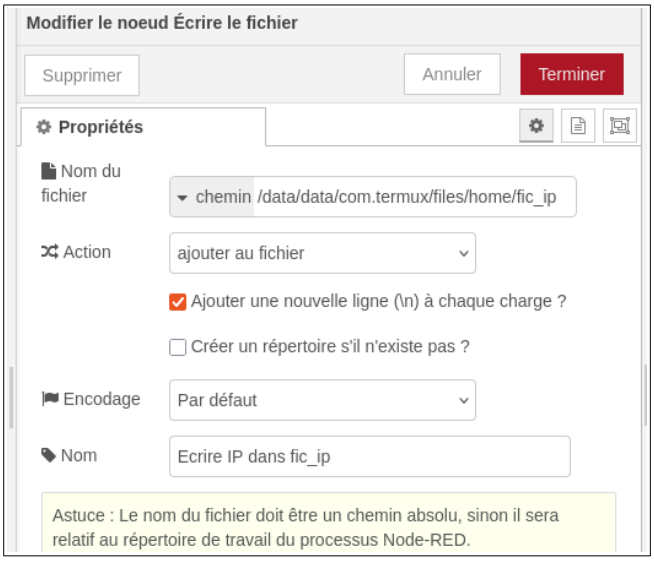
script EnvoiSMS.sh avec argument fic_ip ⇒ ./EnvoiSMS fic_ip

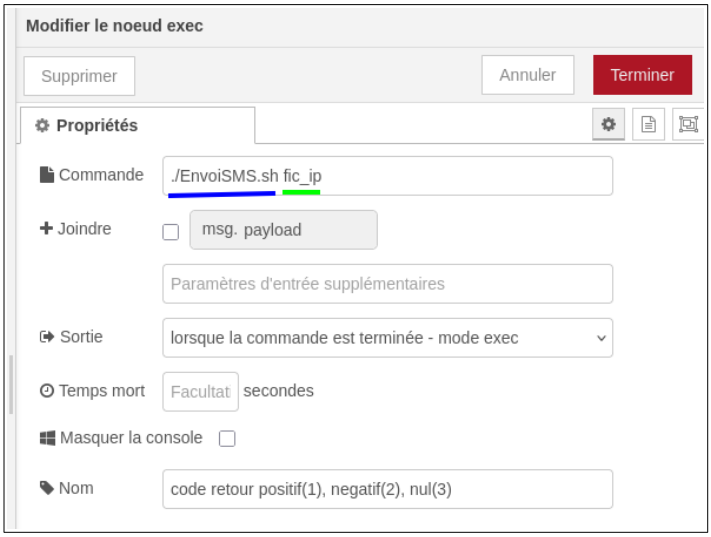
```
#! /bin/bash

if [[ -f "$1" && $(cat "$1"|wc -l) -gt 1 ]]
        then
                d=$(tail -n1 "$1" | cut -d'.' -f4)
                ad=$(tail -n2 "$1" | head -n1 | cut -d'.' -f4)

                if [[ $((ad - d)) -gt 0 ]]
                        then
                                #echo positif
                                exit 1
                        elif [[ $((ad - d)) -lt 0 ]]
                                then
                                        #echo negatif
                                        exit 2
                        else
                                #echo nul
                                exit 3
                fi
        else
                echo "pas de fichier"
fi
```


Flux "Panne courant" basé sur "Wifi Info" (indépendant du contrôle de charge) et le fichier fic_ip
horodatage
Wifi Info
Récupérer IP address
Ecrire IP dans fic_ip
code retour positif(1), negatif(2), nul(3)
error:3
debug 11
Envoi 1 SMS à mon GSM si une "panne courant" détectée !!
1=positif, 2=negatif, 3=nul
Envoi 1 SMS à mon GSM si "courant" rétabli!!
debug 12


Modifier le noeud Écrire le fichier
Supprimer
Annuler
Terminer
Propriétés
Nom du fichier
chemin
/data/data/com.termux/files/home/fic_ip
Action
ajouter au fichier
Ajouter une nouvelle ligne (\n) à chaque charge ?
Créer un répertoire s'il n'existe pas ?
Encodage
Par défaut
Nom
Ecrire IP dans fic_ip
Astuce : Le nom du fichier doit être un chemin absolu, sinon il sera relatif au répertoire de travail du processus Node-RED.


Modifier le noeud exec
Supprimer
Annuler
Terminer
Propriétés
Commande
./EnvoiSMS.sh fic_ip
Joindre
msg. payload
Paramètres d'entrée supplémentaires
Sortie
lorsque la commande est terminée - mode exec
Temps mort
Facultat
secondes
Masquer la console
Nom
code retour positif(1), negatif(2), nul(3)

!!!Remarque importante à propos de « termux-wake-(un)lock » !!!

« termux-wake-lock » doit être utilisé pour empêcher Android de se mettre en veille et permettre ainsi un fonctionnement continu de vos flows Node-RED !

La commande « termux-wake-lock » sera donc lancée dans termux avant la commande « node-red » !

Documentation

https://termuxtools.com/termux-commands-list-for-basic/