

Bouton MOB dans OpenPlotter

Merci à_Kinefou pour le travail que j'ai (honteusement !) récupéré et adapté à mon idée. Voici le lien sur son article :

<https://www.hisse-et-oh.com/articles/2494-ajouter-un-bouton-physique-mob-a-opencpn>

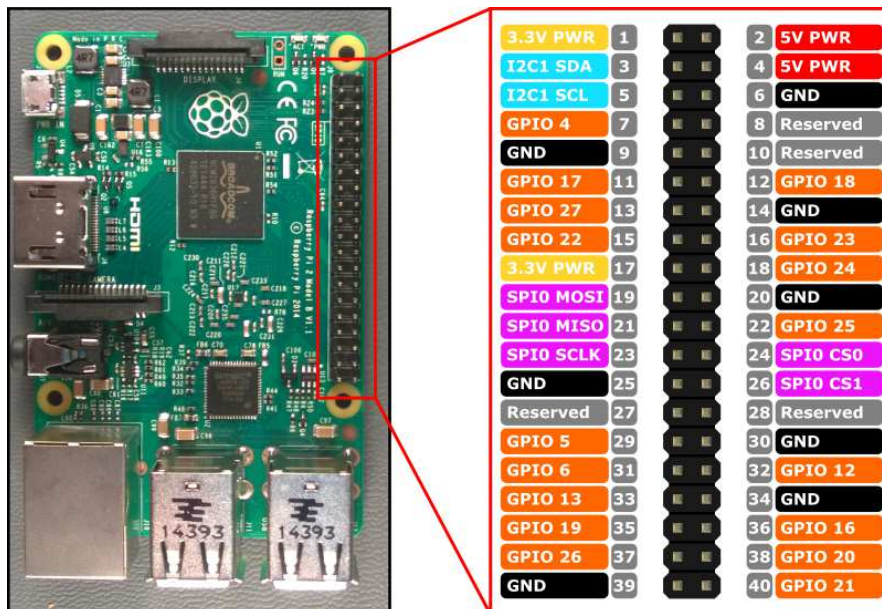
Dans OpenPlotter il est possible de générer une action Signal K (MOB dans OpenCPN et alarme) par la pression sur le bouton MOB.

1. Câblage

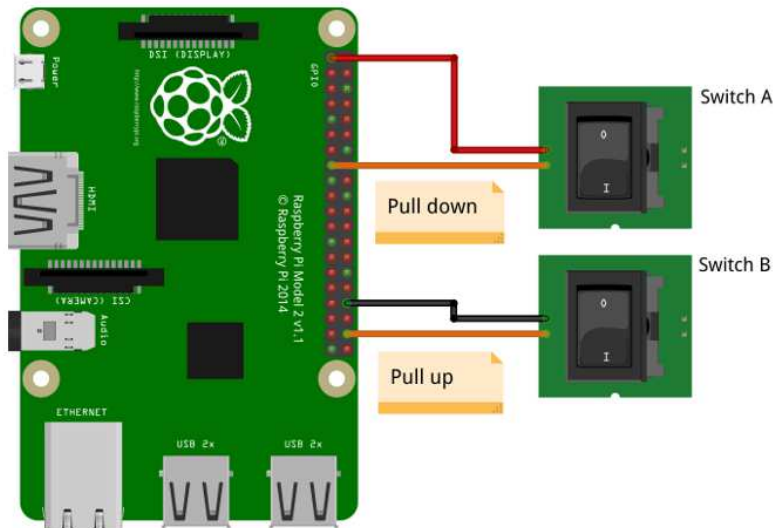
Il nous faut un bouton poussoir étanche et solide qui sera branché d'un côté sur le 3,3V du raspberry et de l'autre au GPIO 17 (pin 11). Le port GPIO 17(dans mon exemple) sera configuré en Input et attendra de se trouver à l'état haut pour déclencher le script.

Le site « Framboise » donne les informations de câblage :

<https://www.framboise314.fr/le-bouton-poussoir-un-composant-banal-o-combien-etonnant/>



Digital sensors are driven through the **GPIO** (General Purpose Input/Output) pins. These pins are copper connected to the CPU chip. A short circuit with 0 V or +3.3 V is not final, but any **direct contact with the + 5 V (or higher) can KILL** the Raspberry Pi.



For a normal digital sensor (open by default), you have to configure it as "Pull down" and to connect it between selected GPIO pin and +3.3v pin (DANGER, NEVER TO +5v).

It is not a problem if you make a mistake connecting to GND or +3.3v but **be careful and avoid the +5v pin**. It isn't a bad idea to add a 1 kOhm resistor into the circuit.

Câblage réalisé pour ce prototypage:

```
+3,3v ----R 1 K ---- BP--|
                          |----GPIO 17 (pin 11)
GND ----R 4,7K-----|
```

Le BP est fermé lorsque appuyé. La résistance de 4,7k positionne l'entrée à 0v au repos

2. Installation xdotool

Le logiciel Kflex de OpenPlotter en appelle le script 'mob.sh' qui enverra la commande « Ctrl Espace » à OpenCPN

Ce script bash fonctionne grâce à xdotool qu'il vous faudra installer sur le pi :

```
pi@:~ $ sudo apt-get install xdotool
```

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances

Lecture des informations d'état... Fait

Les paquets supplémentaires suivants seront installés :

libxdo3

Les NOUVEAUX paquets suivants seront installés :

libxdo3 xdotool

0 mis à jour, 2 nouvellement installés, 0 à enlever et 0 non mis à jour.

Il est nécessaire de prendre 77,8 ko dans les archives.

Après cette opération, 147 ko d'espace disque supplémentaires seront utilisés.

Souhaitez-vous continuer ? [O/n] o

```
Réception de : 1 http://mirrordirector.raspbian.org/raspbian/jessie/main libxdo3 armhf
```

```
1:3.20140805.1-2 [29,2 kB].....
```

```
...Traitement des actions différées (« triggers ») pour libc-bin (2.19-18+deb8u10) ...
```

```
pi@:~ $
```

3. script d'appel xdotool

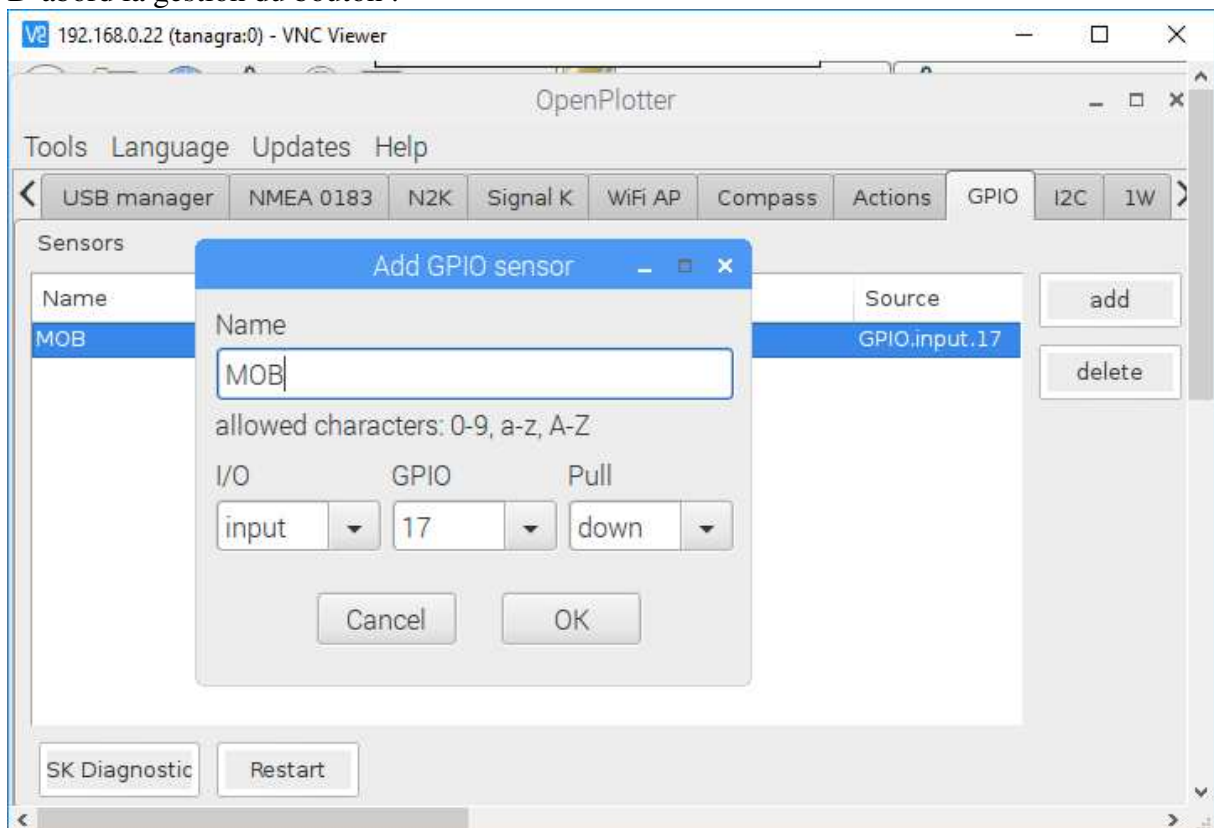
Le contenu du script mob.sh est le suivant :

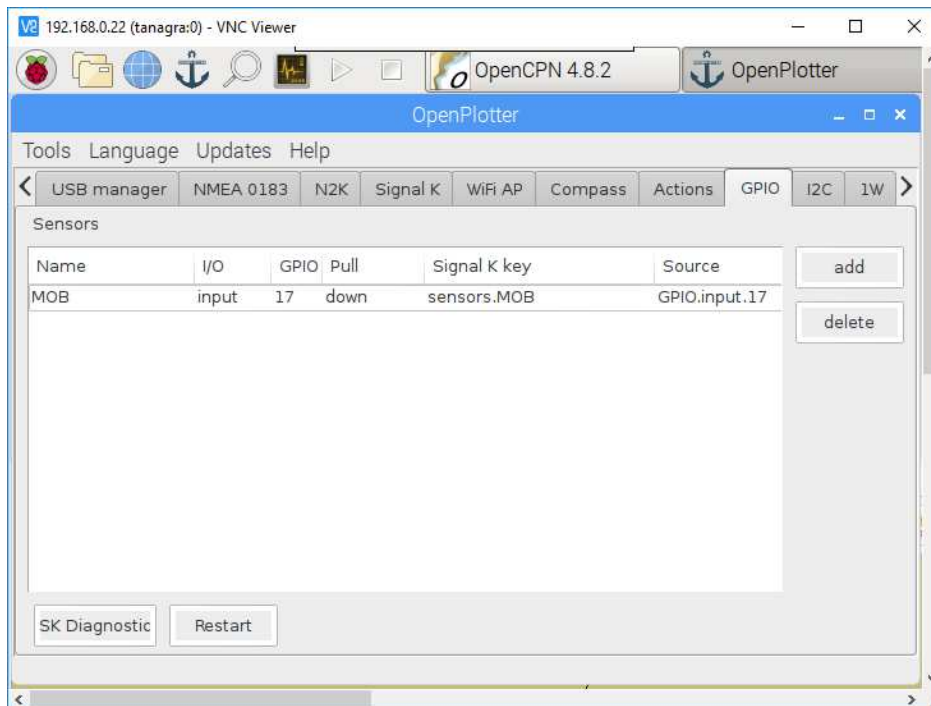
```
#!/bin/bash
# la commande a lancer depuis linux en mode commande:
#sh /home/pi/mob.sh
#####
# on cherche la fenetre (attention au changement de nom/version)
WID=`xdotool search --name "OpenCPN 4.8.2" `
# on active la fenetre
xdotool windowfocus $WID
# on envoie le raccourci clavier MOB
xdotool key ctrl+space
```

4. MOB par OpenPlotter

Ci-dessous la configuration dans OpenPlotter.

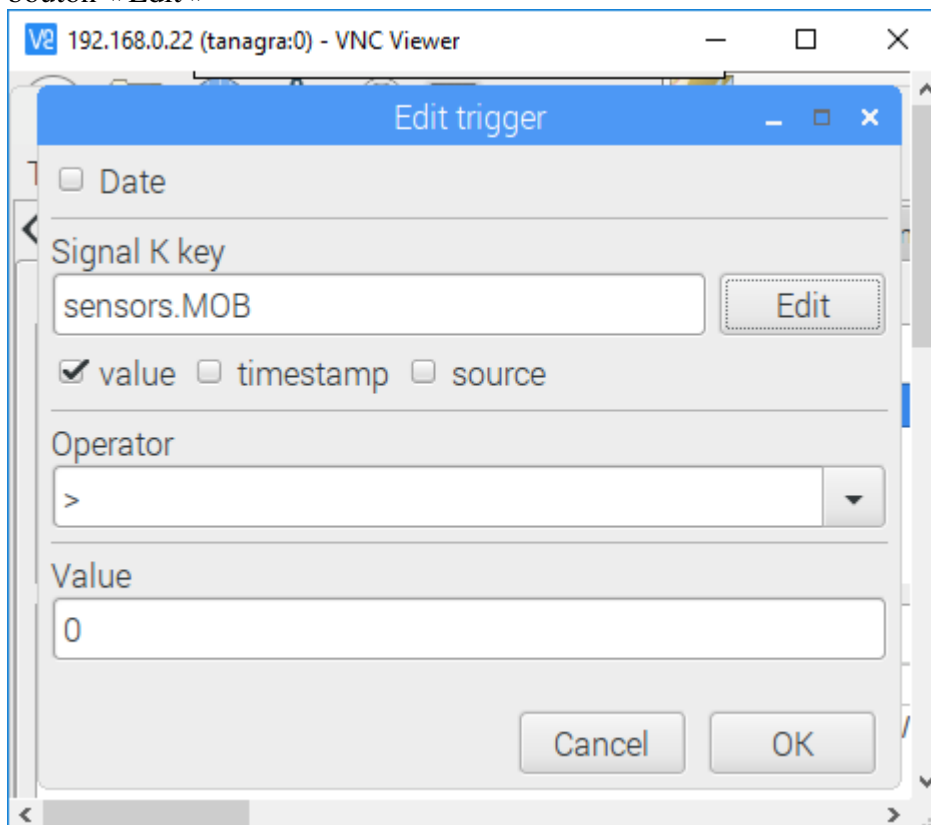
D'abord la gestion du bouton :

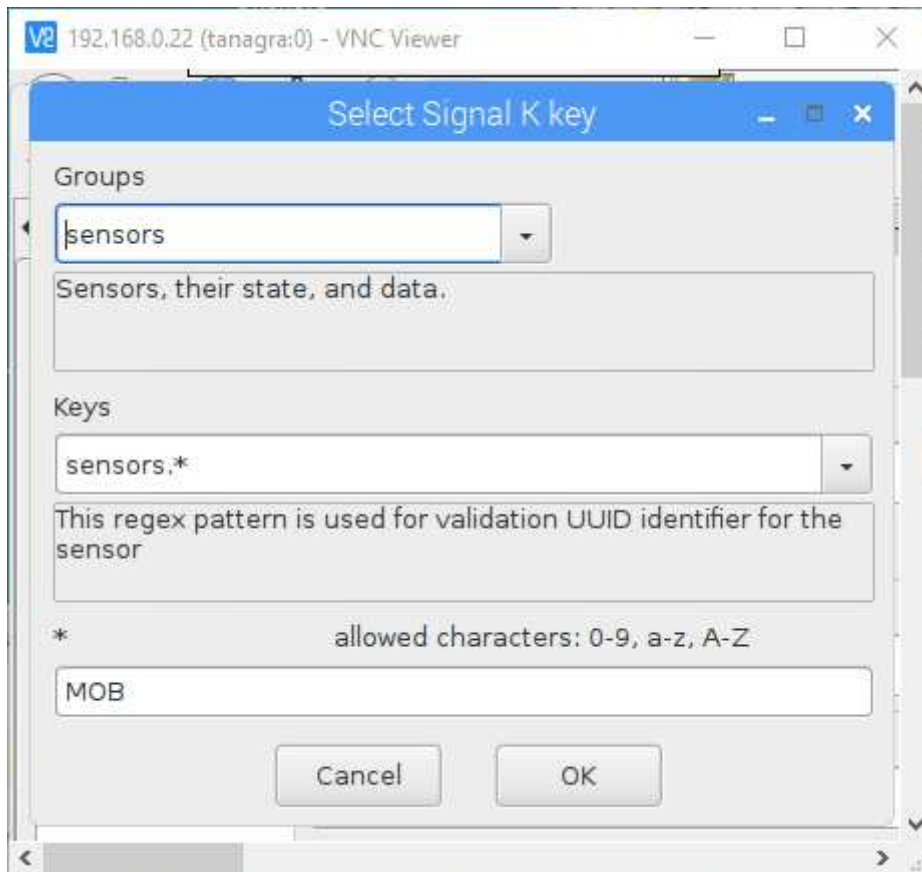




Le lien entre le capteur GPIO (onglet GPIO) et le trigger (onglet Actions) est fait au travers du nom (MOB dans cet exemple).

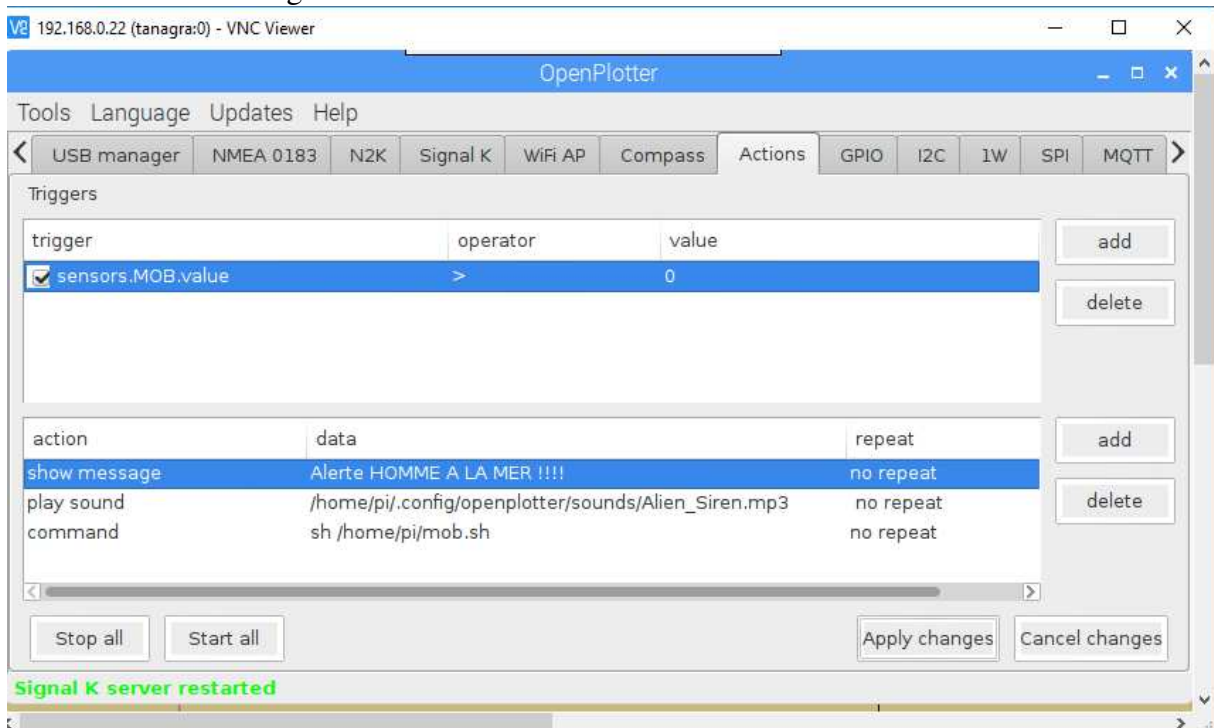
Il faut cliquer sur le bouton « Add » de la fenêtre trigger et définir le nom signal K par le bouton « Edit »





Ce trigger défini dans la fenêtre « Triggers » déclenche les actions définies dans la fenêtre « Actions » par le bouton « add » du bas.

Le résultat de la configuration doit ressembler à la fenêtre ci-dessous :

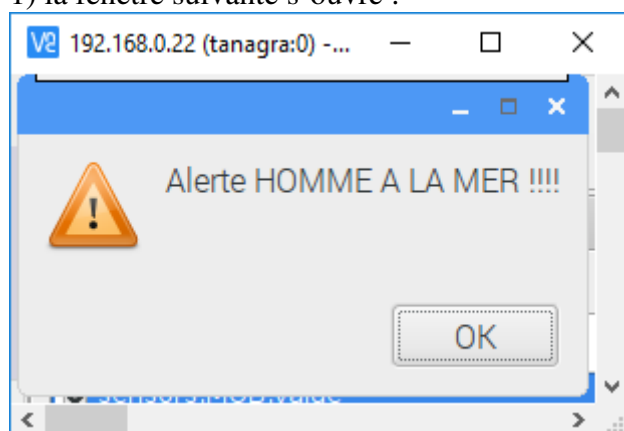


Ne pas oublier d'enregistrer la configuration par « Apply changes » et « Start all »

Le test :

Sur une action sur le bouton poussoir :

1) la fenêtre suivante s'ouvre :



Elle a été définie dans l'action « show message »

2) L'alarme sonore se fait entendre

(action « play sound »)

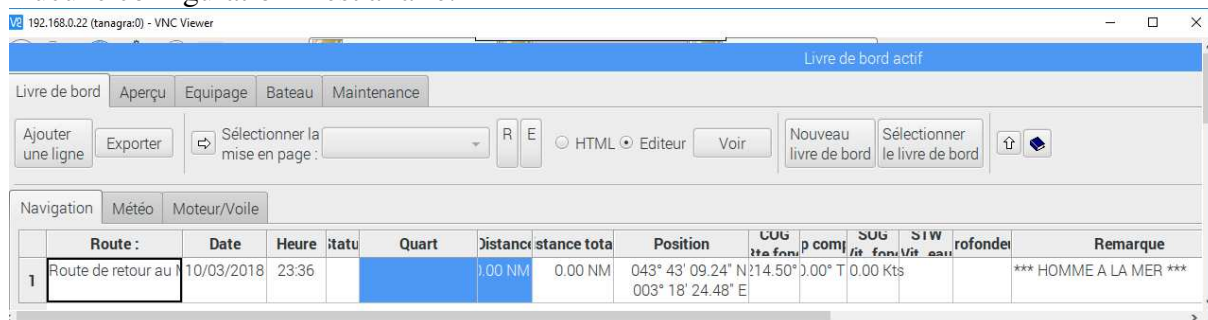
3) Le script mob.sh s'exécute et enregistre le MOB dans OpenCPN

(action « command »)

5. Enregistrement MOB dans le livre de bord

La génération du MOB dans OpenCPN est automatiquement complété par une ligne dans le livre de bord « Logbook Koni », si ce plugin OpenCPN est installé et activé au préalable !

Aucune configuration n'est à faire.

A screenshot of the OpenCPN Logbook Koni interface. The window title is "192.168.0.22 (tanagra:0) - VNC Viewer". The interface shows a "Livres de bord" section with tabs for "Aperçu", "Equipage", "Bateau", and "Maintenance". Below these are buttons for "Ajouter une ligne", "Exporter", and "Sélectionner la mise en page:". A table displays logbook entries. The first entry is highlighted in blue and contains the following data:

	Route :	Date	Heure	Statu	Quart	Distance	Distance tota	Position	COG	Drift	SOG	STW	Profondeur	Remarque
1	Route de retour au	10/03/2018	23:36			0.00 NM	0.00 NM	043° 43' 09.24" N 003° 18' 24.48" E	14.50°	0.00°	0.00 Kts			*** HOMME A LA MER ***