Enregistrement de vidéos dans la solution domotique **WebThings**.

Retour d’expérience.

WebThings : un petit rappel.

RPI >=1

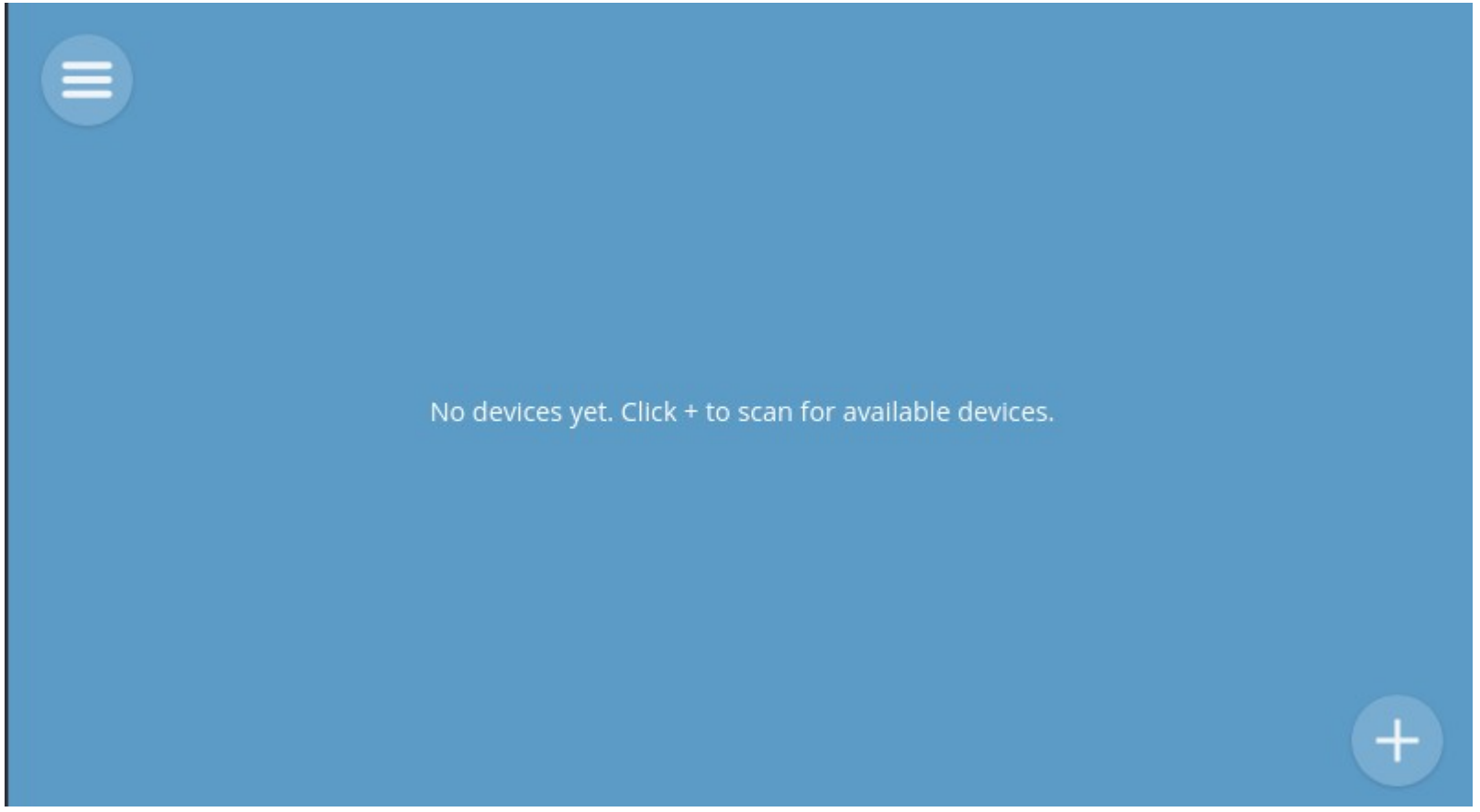
conbee 2 zigbee
https://phoscon.de/en/conbee2

https://phoscon.de/en/conbee2/compatible

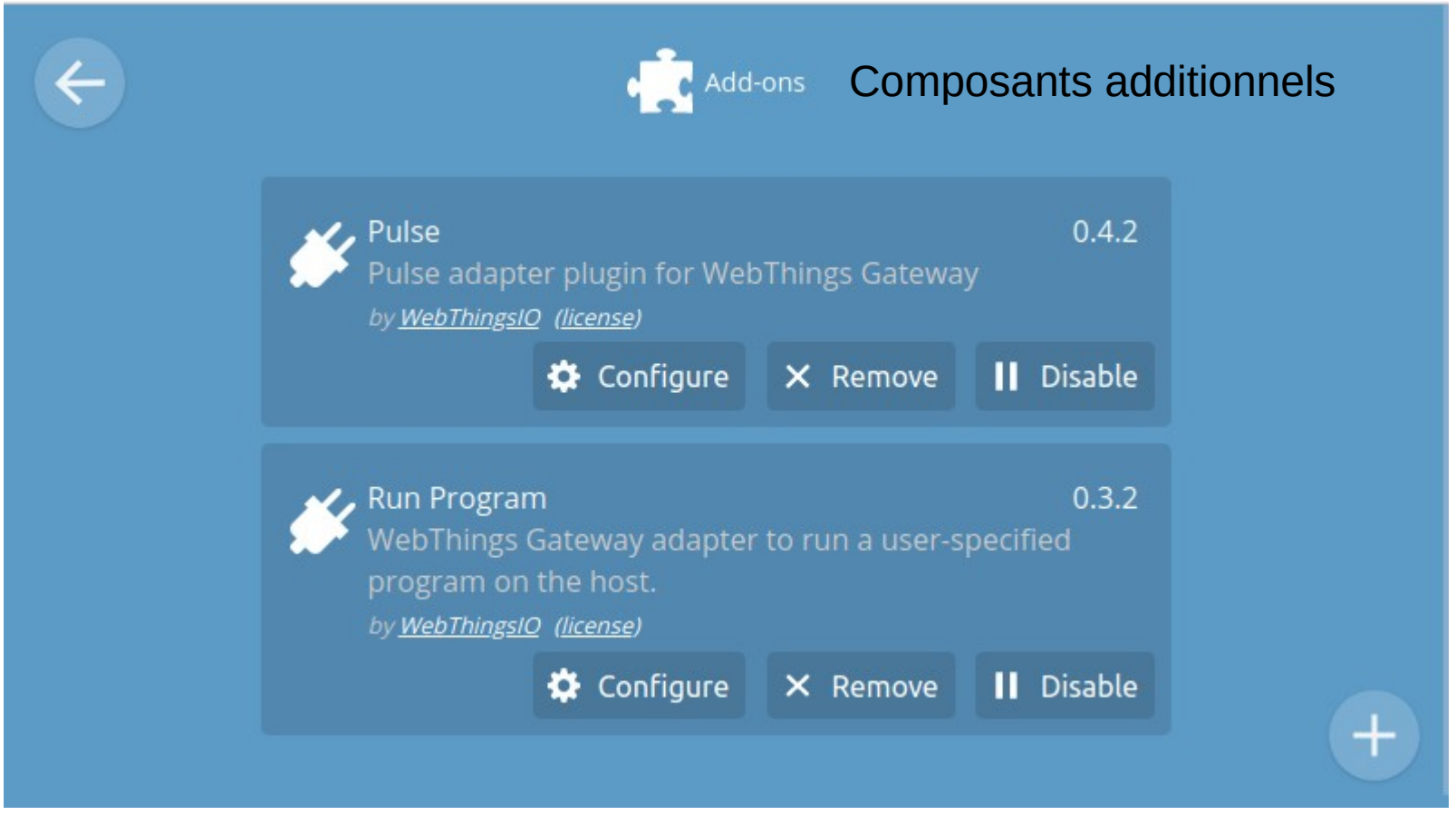
Détecteur de Mouvements

Motion sensor with temperature- and light measurement

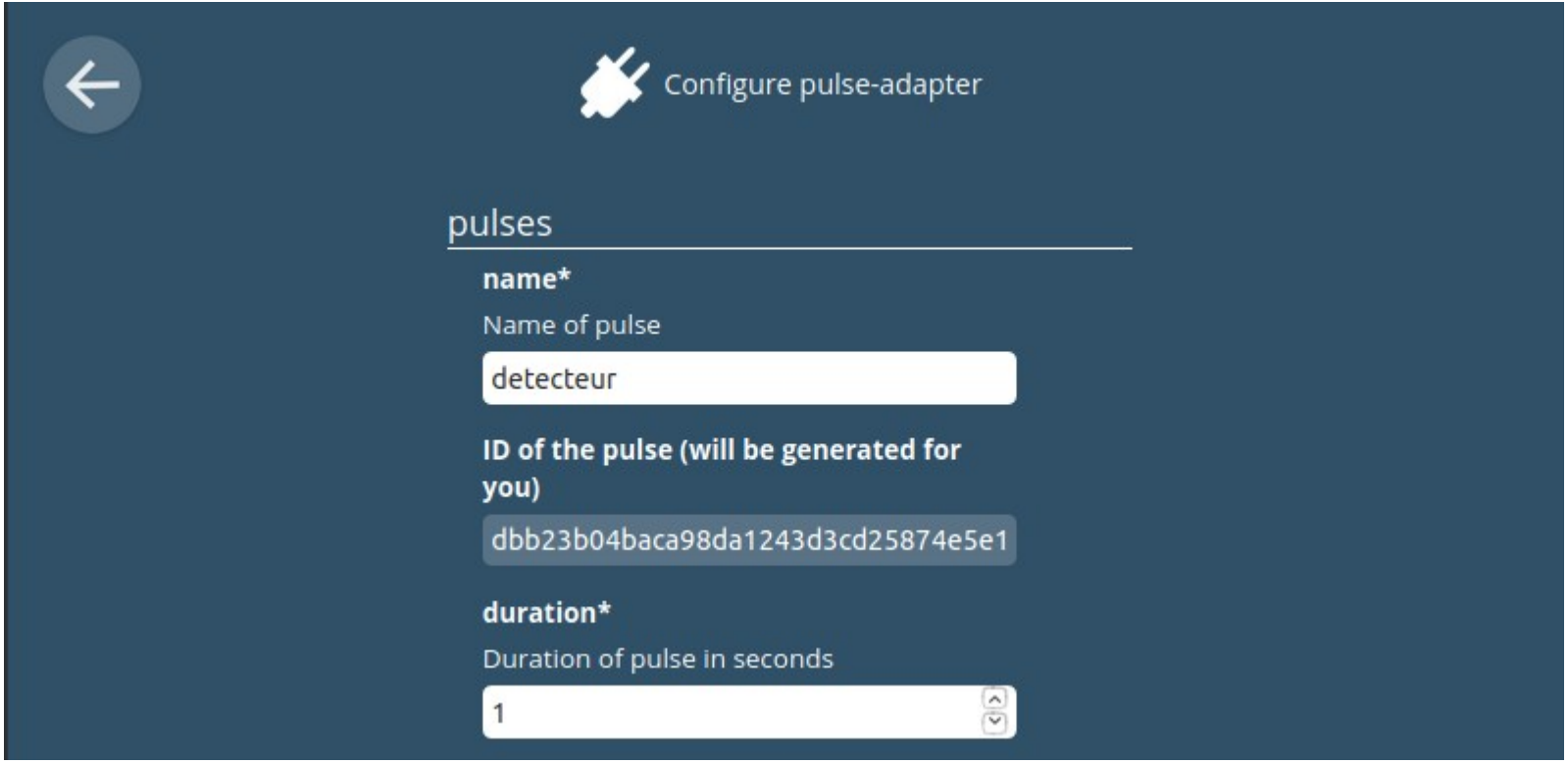
WebThings.



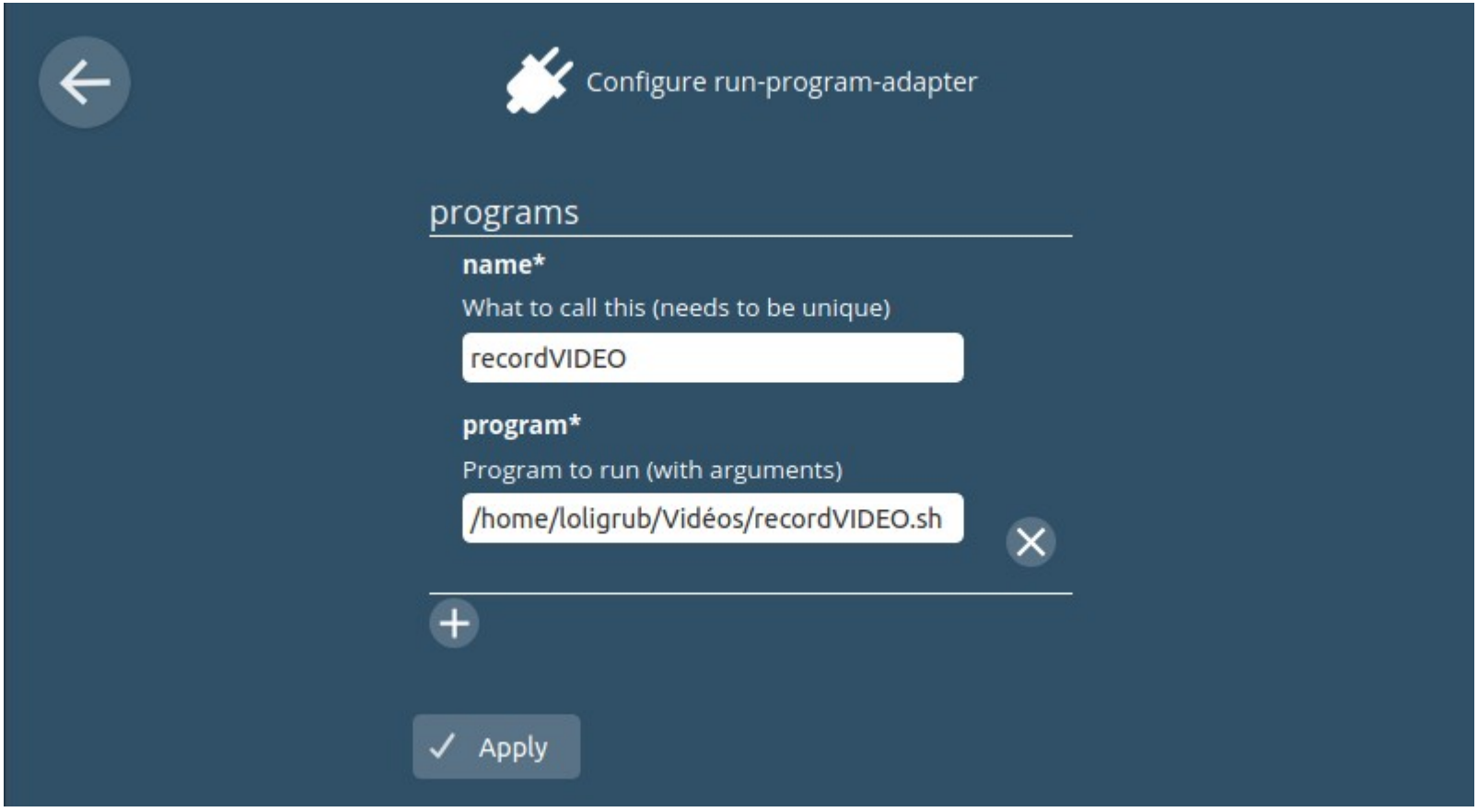
WebThings.


Add-ons
Composants additionnels
Pulse
0.4.2
Pulse adapter plugin for WebThings Gateway
by WebThingsIO (license)
Configure
Remove
Disable
Run Program
0.3.2
WebThings Gateway adapter to run a user-specified program on the host.
by WebThingsIO (license)
Configure
Remove
Disable

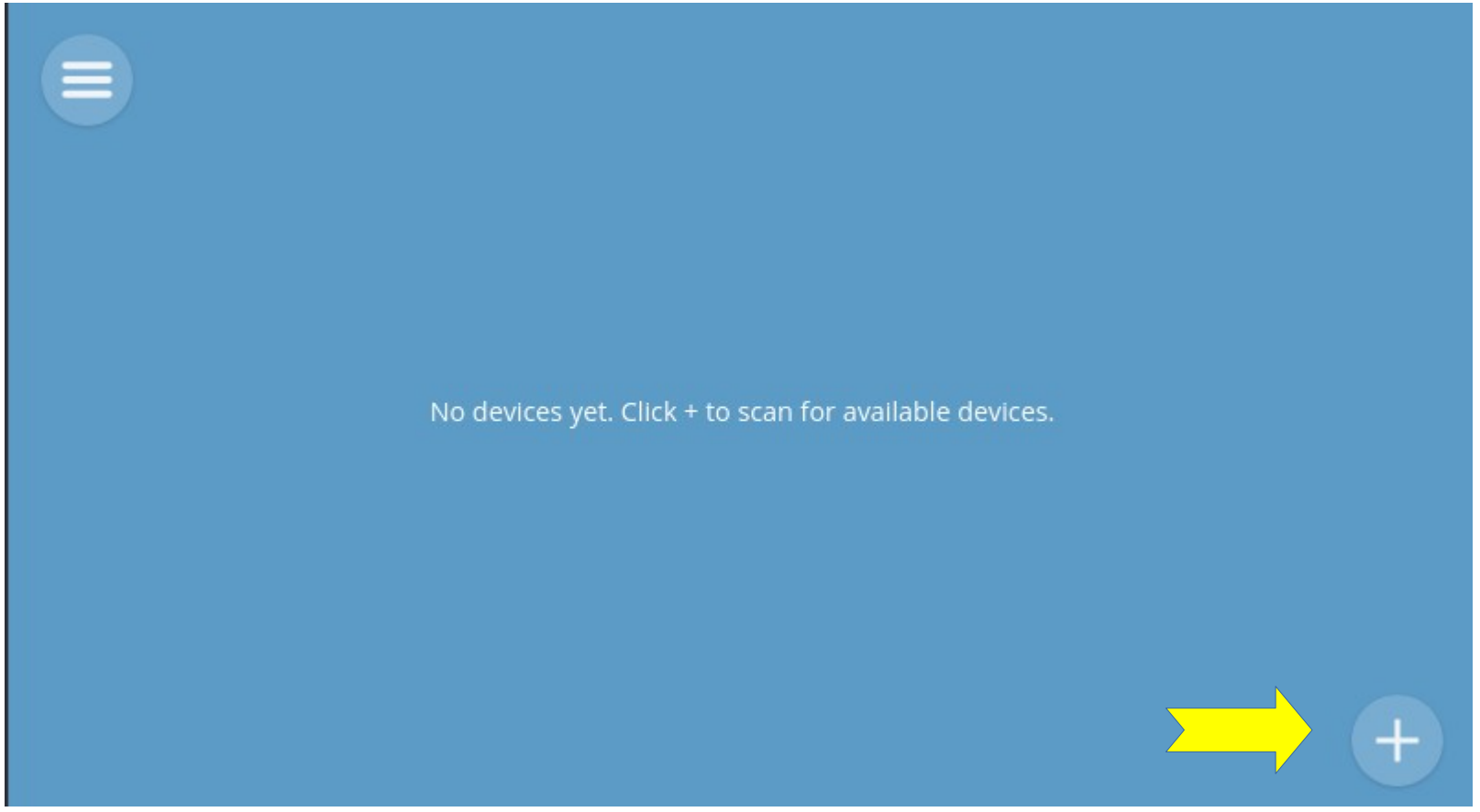
WebThings.


Configure pulse-adapter
pulses
name*
Name of pulse
detecteur
ID of the pulse (will be generated for you)
dbb23b04baca98da1243d3cd25874e5e1
duration*
Duration of pulse in seconds
1

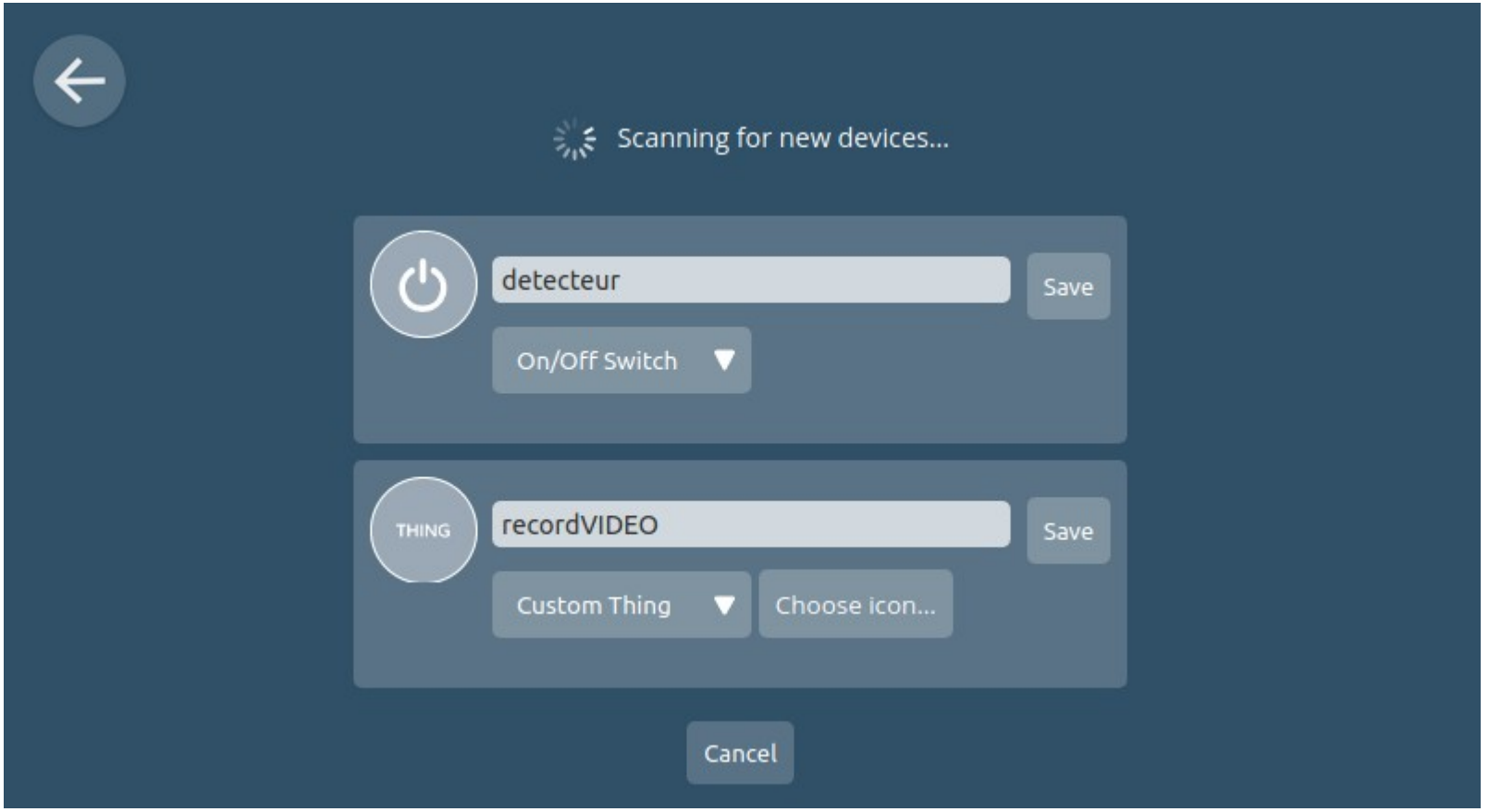
WebThings.


Configure run-program-adapter
programs
name*
What to call this (needs to be unique)
recordVIDEO
program*
Program to run (with arguments)
/home/loligrub/Vidéos/recordVIDEO.sh
Apply

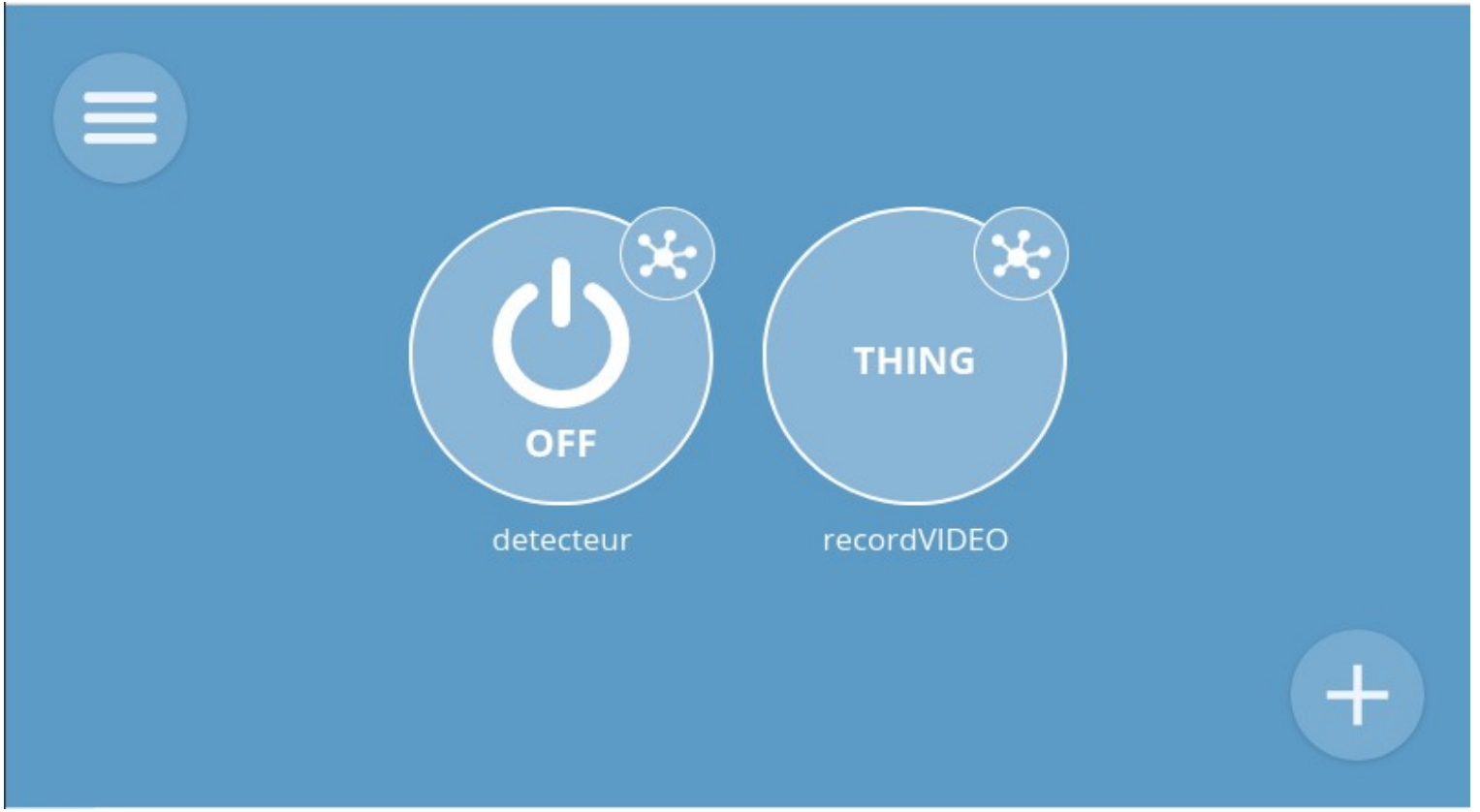
WebThings.



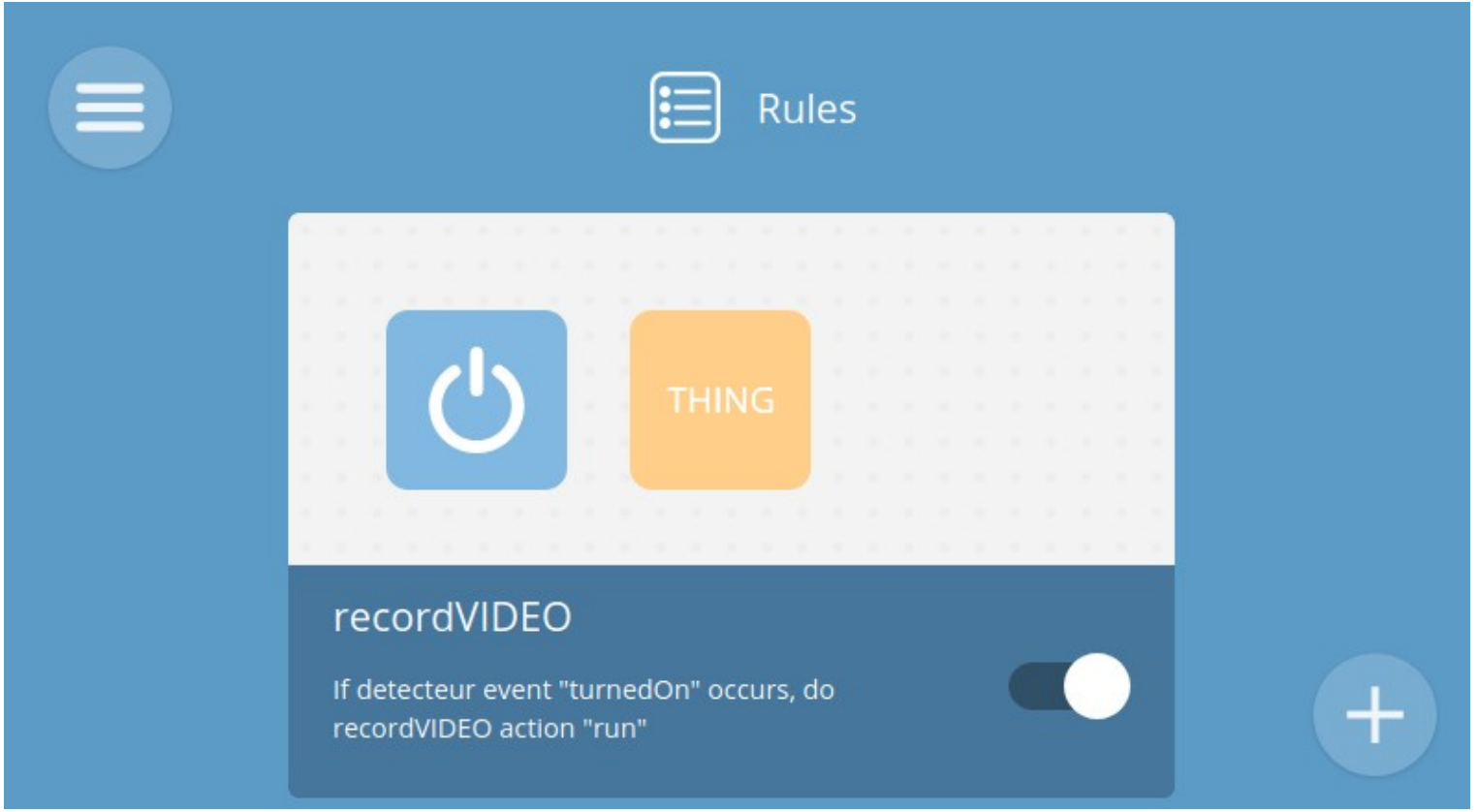
WebThings.


Scanning for new devices...
detecteur
Save
On/Off Switch
THING
recordVIDEO
Save
Custom Thing
Choose icon...
Cancel

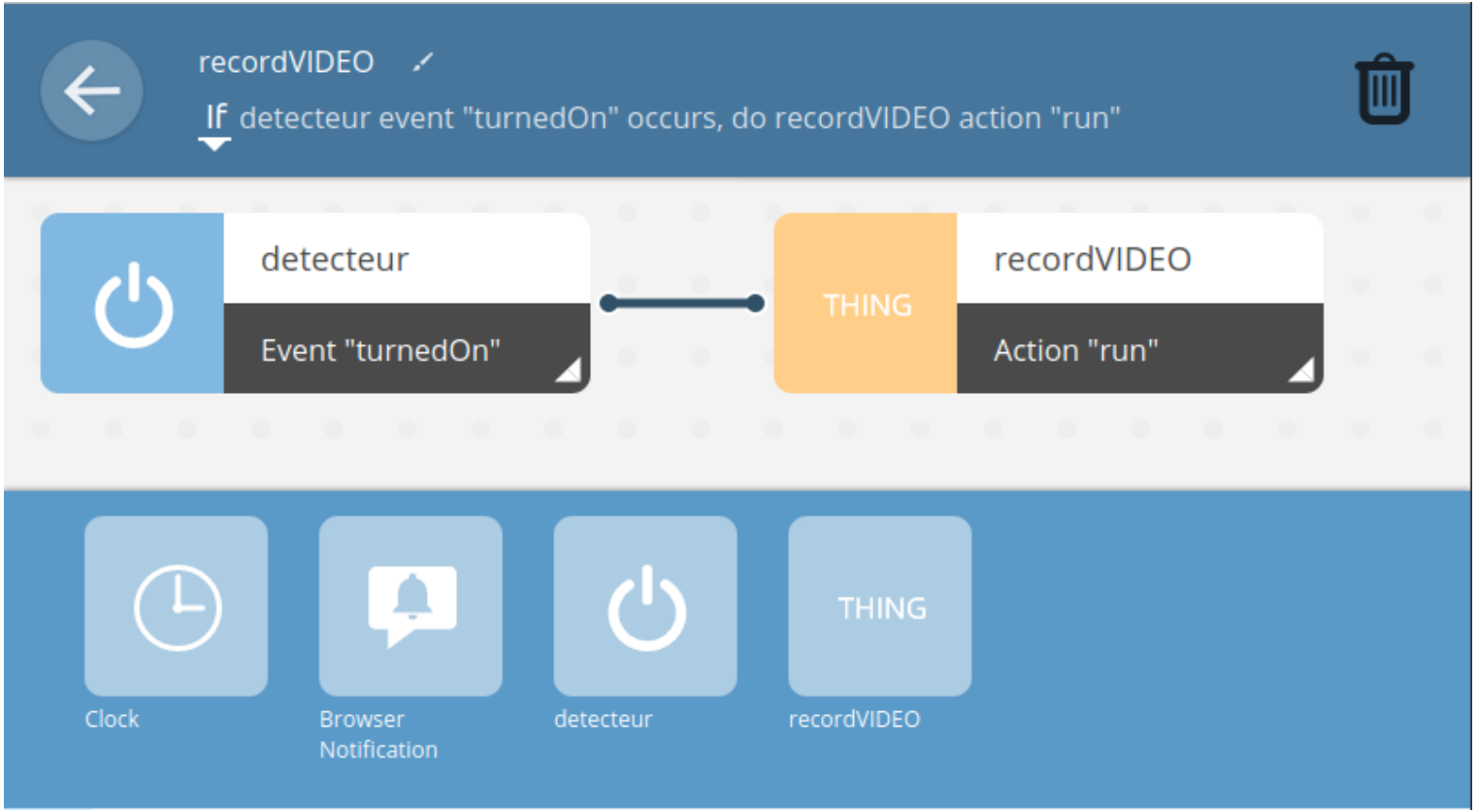
WebThings.



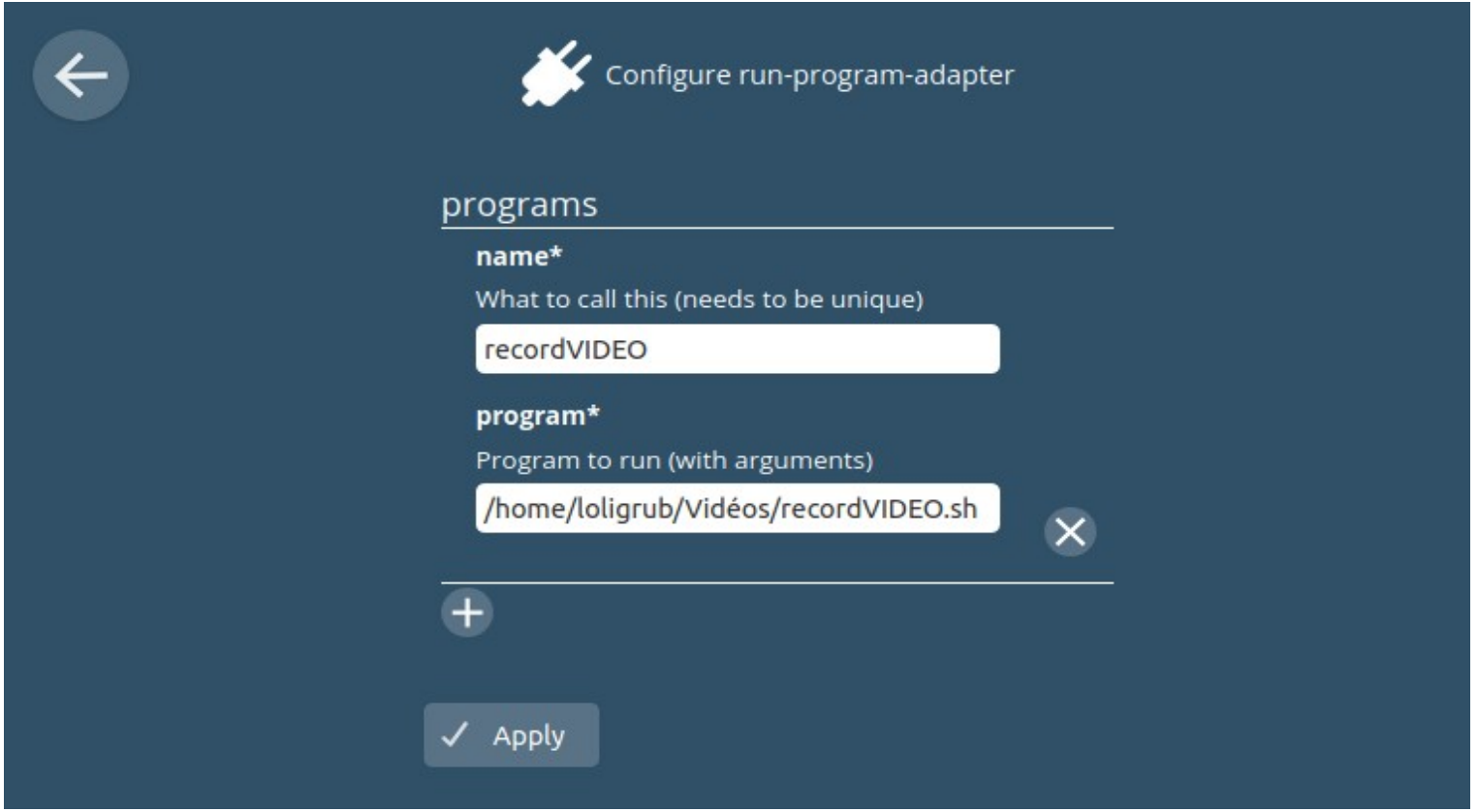
WebThings.


Rules
THING
recordVIDEO
If detecteur event "turnedOn" occurs, do
recordVIDEO action "run"

WebThings.


recordVIDEO
If detecteur event "turnedOn" occurs, do recordVIDEO action "run"
detecteur
Event "turnedOn"
THING
recordVIDEO
Action "run"
Clock
Browser Notification
detecteur
THING
recordVIDEO

WebThings.

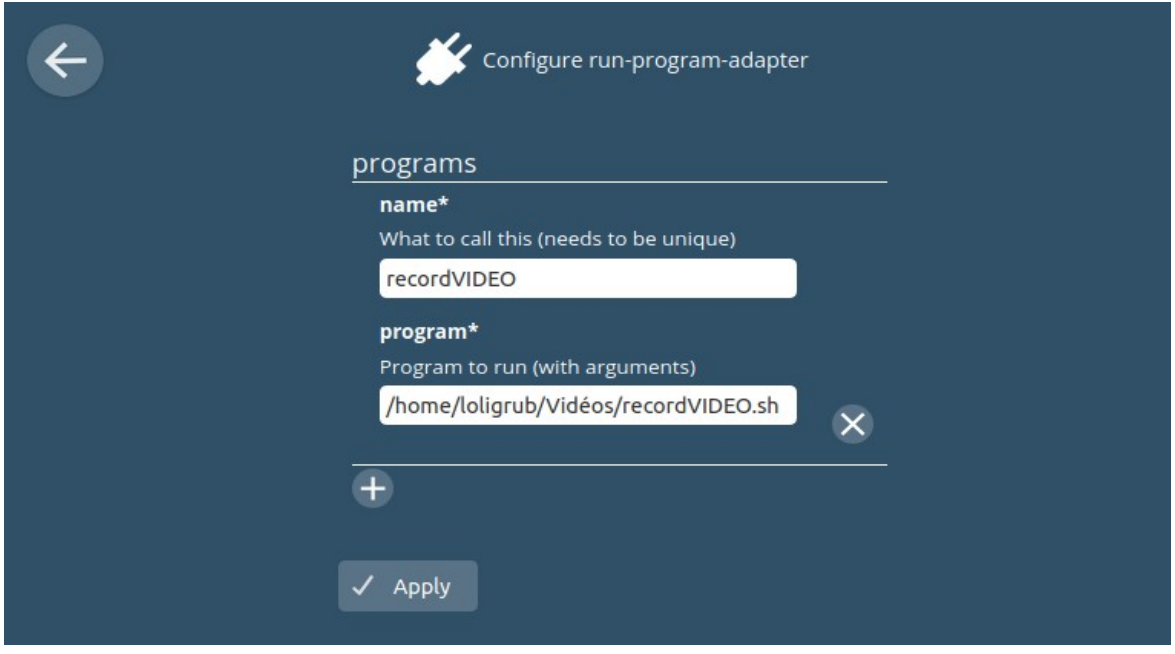

Configure run-program-adapter
programs
name*
What to call this (needs to be unique)
recordVIDEO
program*
Program to run (with arguments)
/home/loligrub/Vidéos/recordVIDEO.sh
Apply

WebThings.

Visualisons l’interface;-)

WebThings.

Quelle(s) commande(s) va contenir le script recordVIDEO.sh pour enregistrer un flux vidéo?



Serveur centralisé de vidéo surveillance avec des RaspberryPi

V4l2rtspserver , permet de visualiser des caméras de RPI via un flux réseau sans surcharger le CPU d'un RPI-1 !

Protocol RTSP : https://fr.wikipedia.org/wiki/Real_Time_Streaming_Protocol

Un des auteurs mentionne également la possibilité d'utiliser MotionEye (RPI >=3) pour visualiser les différentes caméras de façon centralisée.

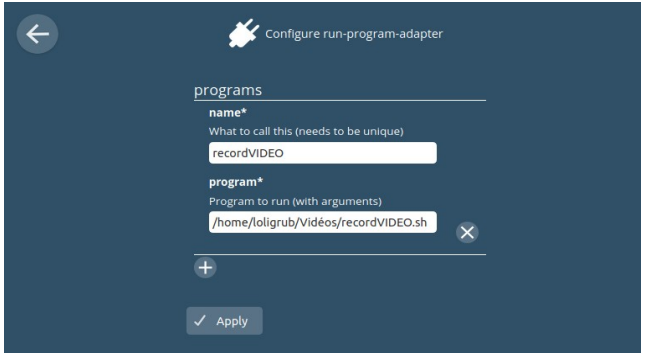
Sources des articles :

- **https://ouafnico.shivaserv.fr/posts/geek-raspberry-motioneye/ (MotionEye sur RPI)**
- **https://sylvaindurand.fr/surveillance-camera-with-raspberry-pi/**

Documentation v4l2rtspserver : https://github.com/mpromonet/v4l2rtspserver

v4l2rtspserver

- **Schéma général**
- **Installation**
- **Lancement et tests de paramètres avec VLC**
- **configurer le service**
- **activer et démarrer le service**
- **FFMPEG**
- **Script : recordVIDEO.sh**



v4l2rtspserver

- **Schéma général**

On va tester le script sur un PC et une fois le résultat vérifié, on pourra le transférer dans le système domotique ;-)

v4l2rtspserver

- Schéma général
- **Installation**
  - Suivre les instructions d’un des deux sites
    - **https://ouafnico.shivaserv.fr/posts/geek-raspberry-motioneye/**
    - **https://sylvaindurand.fr/surveillance-camera-with-raspberry-pi/**

sudo apt install cmake liblog4cpp5-dev libv4l-dev ⇒ outils pour compiler

git clone https://github.com/mpromonet/v4l2rtspserver.git ⇒ cloner le dépôt

cd v4l2rtspserver/ ⇒ se placer dans le réperetoire

cmake .

make

compiler

sudo make install ⇒ installation

v4l2rtspserver

- Schéma général
- Installation
- **Lancement et tests de paramètres**

dans le RPI lancer la commande :

v4l2rtspserver -H 600 -W 800 -F 10 -P 8888 -U loli:grub /dev/video0

dans VLC , lire le flux réseau « rtsp://loli:grub@192.168.1.3:8888/unicast »

v4l2rtspserver

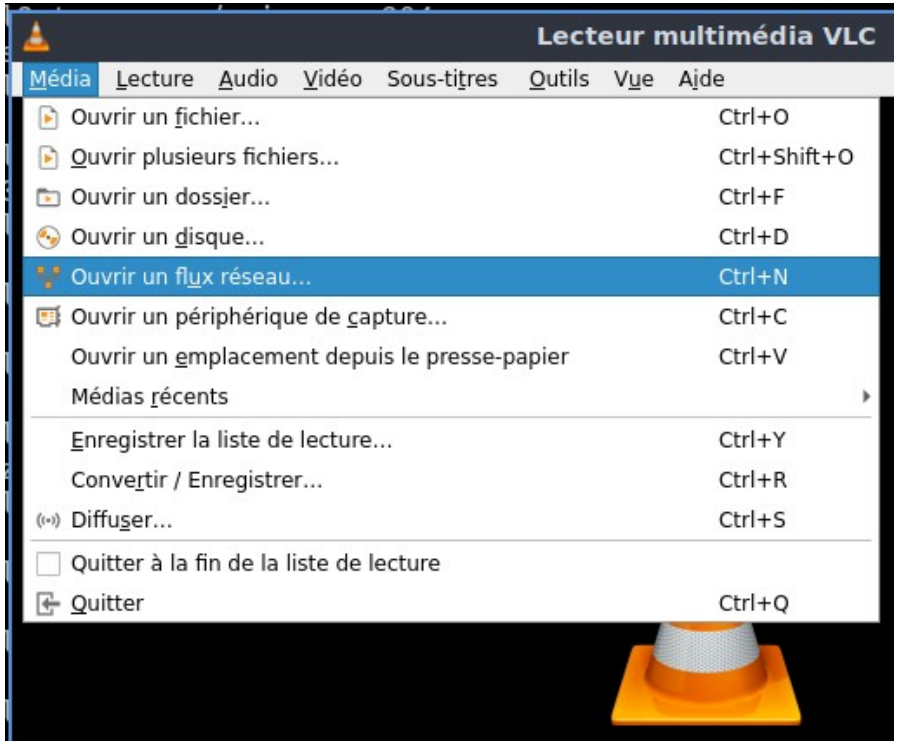
labojurbise@labojurbise-2522g33:~$ ssh pi@192.168.1.3

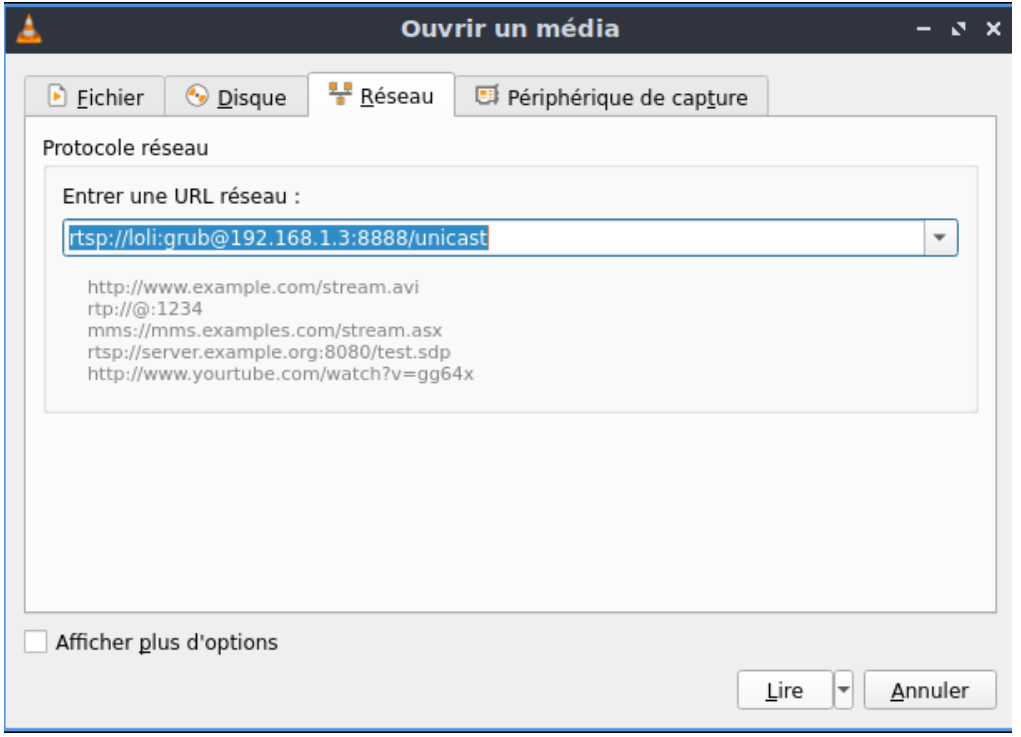
pi@raspberrypi:~ $ v4l2rtspserver -H 600 -W 800 -F 10 -P 8888 -U loli:grub /dev/video0

```
pi@raspberrypi:~ $ v4l2rtspserver -H 600 -W 800 -F 10 -P 8888 -U loli:grub /dev/video0
2022-03-10 19:46:08,742 [NOTICE] - /home/pi/v4l2rtspserver/main.cpp:294
        Version: 0.2.3-34-g1f0c768 live555 version:2021.10.31
2022-03-10 19:46:08,751 [NOTICE] - /home/pi/v4l2rtspserver/src/V4l2RTSPServer.cpp:36
        Create V4L2 Source.../dev/video0
.......
.......
2022-03-10 19:46:09,217 [NOTICE] - /home/pi/v4l2rtspserver/inc/V4l2RTSPServer.h:81
        Play this stream using the URL "rtsp://192.168.1.3:8888/unicast"
2022-03-10 19:46:09,219 [NOTICE] - /home/pi/v4l2rtspserver/src/DeviceSource.cpp:93
        begin thread
```

v4l2rtspserver

Dans VLC(version 3.0.16)

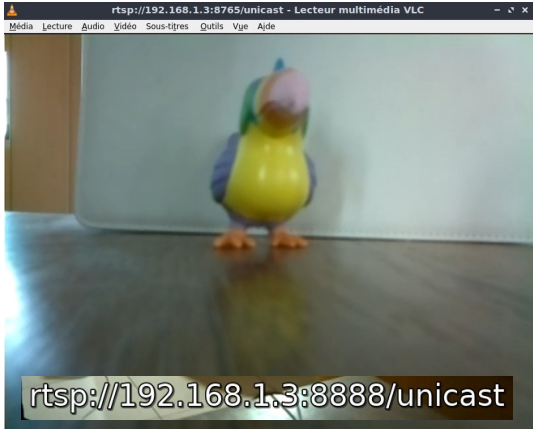




v4l2rtspserver

v4l2rtspserver -H 600 -W 800 -F 10 -P 8888 -U loli:grub /dev/video0

Dans VLC



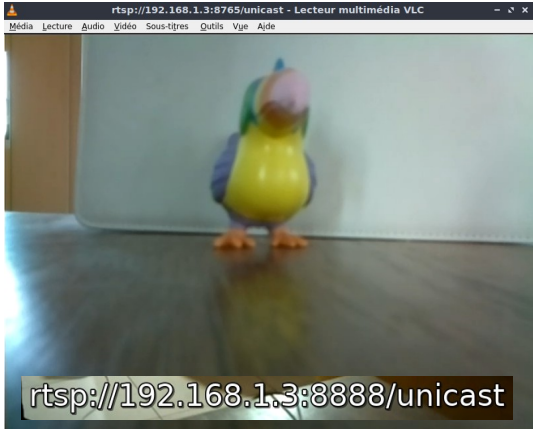

```
top - 17:23:35 up 21:58,  1 user,  load average: 0.23, 0.16, 0.10
Tasks:  77 total,   1 running,  76 sleeping,   0 stopped,   0 zombie
%Cpu(s):  3.0 us,  6.4 sy,  0.0 ni, 90.0 id,  0.0 wa,  0.0 hi,  0.7 si,  0.0 st
MiB Mem :    366.6 total,    211.1 free,     35.8 used,    119.7 buff/cache
MiB Swap:    100.0 total,    100.0 free,      0.0 used.    277.1 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
  958 root      20   0   25012   9248   8468 S   7.8   2.5  36:29.00 v4l2rtspserver
```

v4l2rtspserver

v4l2rtspserver -H 1200 -W 1600 -F 10 -P 8888 -U loli:grub /dev/video0

Dans VLC




```
top - 20:01:29 up 36 min,  2 users,  load average: 0.26, 0.16, 0.11
Tasks:  79 total,   1 running,  78 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.5 us, 10.9 sy,  0.0 ni, 77.6 id,  0.0 wa,  0.0 hi,  4.1 si,  0.0 st
MiB Mem :    366.6 total,    240.1 free,     51.0 used,     75.5 buff/cache
MiB Swap:    100.0 total,    100.0 free,      0.0 used.    264.6 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR    %CPU  %MEM     TIME+ COMMAND
  849 pi        20   0   37172  22676  22356   22.1   6.0   0:15.09 v4l2rtspserver
```

WebThings.

Testons ;-)

v4l2rtspserver

- Installation
- Lancement et tests de paramètres
- **configurer le service**
  - éditer le service systemd : /usr/lib/systemd/system/v4l2rtspserver.service
  - Modifiez la ligne du “ExecStart” :

    pi@raspberrypi:~ $ sudo nano /usr/lib/systemd/system/v4l2rtspserver.service

    ExecStart=/usr/local/bin/v4l2rtspserver -H 600 -W 800 -F 15 -P 8888 -U loli:grub /dev/video0

v4l2rtspserver

- Installation
- Lancement et tests de paramètres
- configurer le service
- **activer et démarrer le service :**
  - `systemctl enable v4l2rtspserver`
  - `systemctl start v4l2rtspserver`

v4l2rtspserver

activer et démarrer le service :

- `systemctl enable v4l2rtspserver`
- `systemctl start v4l2rtspserver`
- `pi@raspberrypi:~ $ systemctl status v4l2rtspserver`

```
Warning: The unit file, source configuration file or drop-ins of v4l2rtspserver.service
changed on disk. Run 'systemctl daemon-reload' to rel
● v4l2rtspserver.service - V4L2 RTSP server
   Loaded: loaded (/lib/systemd/system/v4l2rtspserver.service; enabled; vendor preset:
enabled)
   Active: active (running) since Tue 2022-03-15 16:51:59 GMT; 31min ago
 Main PID: 1830 (v4l2rtspserver)
    Tasks: 2 (limit: 725)
   CGroup: /system.slice/v4l2rtspserver.service
           └─1830 /usr/local/bin/v4l2rtspserver -H 600 -W 800 -F 15 -P 8888 -U loli:grub
/dev/video0

raspberrypi v4l2rtspserver[1830]:   Play this stream using the URL
"rtsp://192.168.1.3:8888/unicast"
```

WebThings.

Nous avons vu comment :

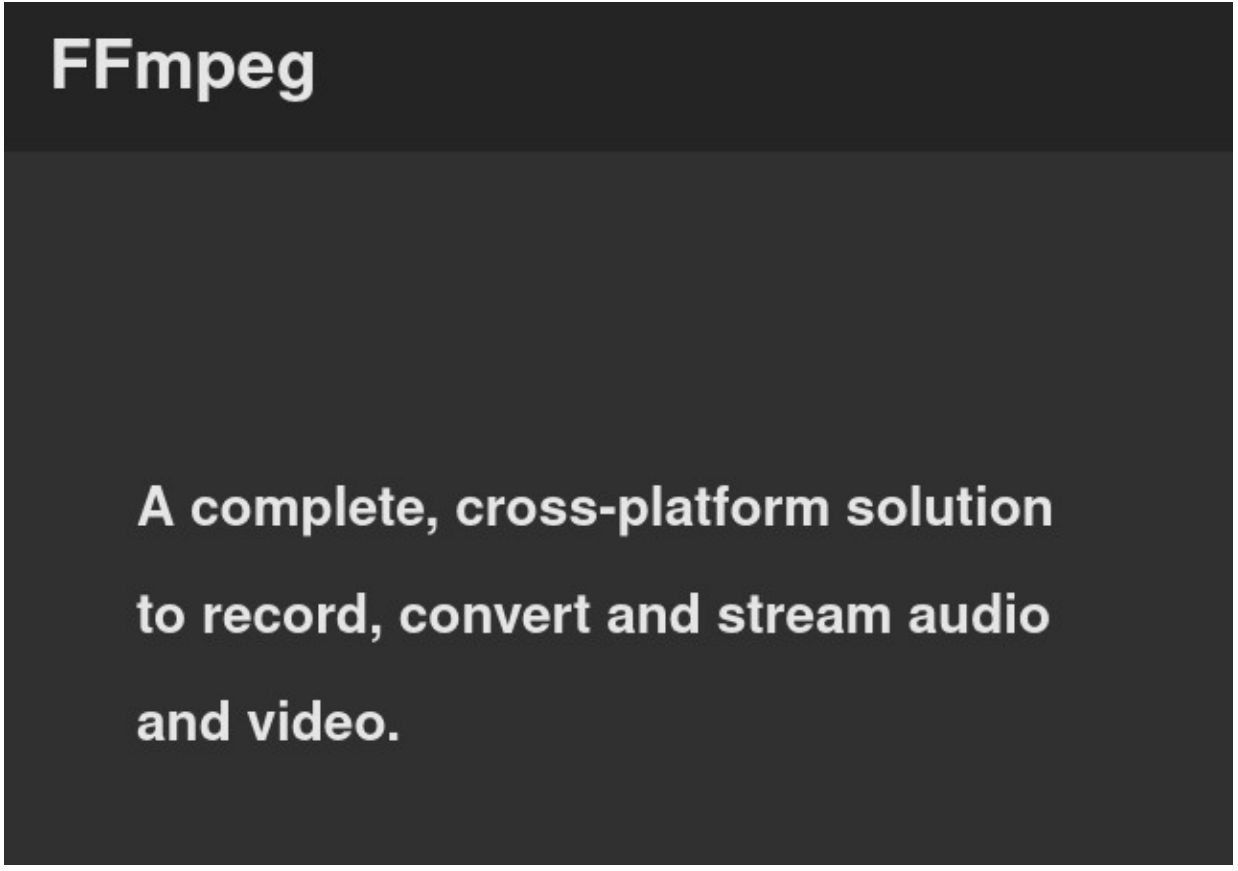
- Configurer le serveur v4lrtspserver
- l’activer en permanence sur le RPI-1
- Visualiser le flux via vlc

Il nous reste à découvrir comment enregistrer un flux vidéo de x secondes

FFmpeg

FFmpeg : https://ffmpeg.org/

- Une solution complète et multiplateforme pour **enregistrer**, convertir et diffuser des fichiers audio et **vidéo**.



FFmpeg

labojurbise@labojurbise-2522g33:~$

ffmpeg -rtsp_transport tcp -i rtsp://loli:grub@192.168.1.3:8888/unicast -vcodec copy -an -t 60 /home/labojurbise/Vidéos/"Video".mp4

- -i rtsp://loli:grub@192.168.1.3:8888/unicast ⇒ fichier source que l’on a déjà utilisé dans VLC
- -vcodec copy ⇒ copie le flux sans réencodage
- -an ⇒ supprime l’audio
- -t 60 ⇒ durée de 60 secondes
- /home/labojurbise/Vidéos/"Video".mp4 ⇒ fichier de destination
- -rtsp_transport tcp ⇒ utilise TCP comme protocole de transport

WebThings.

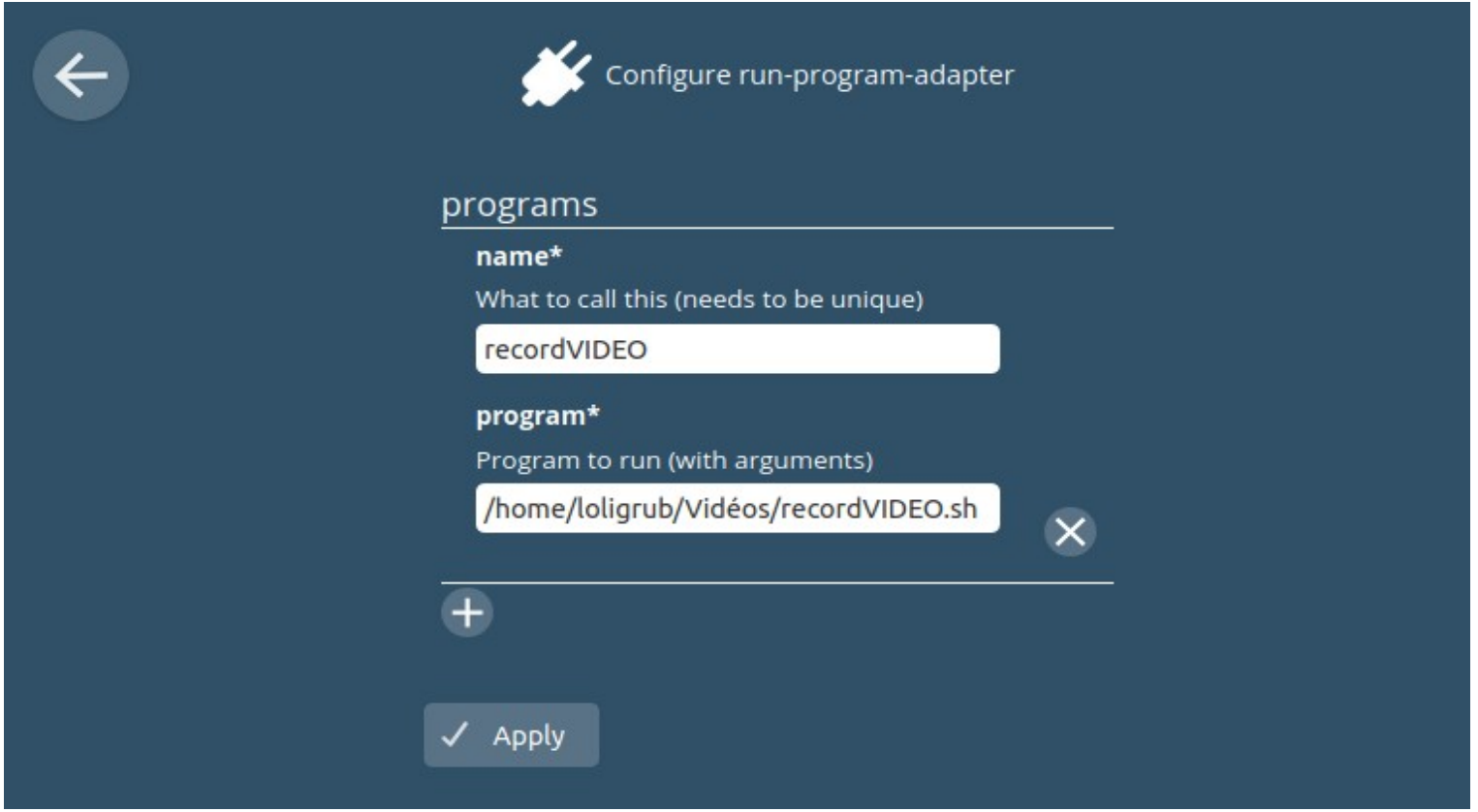
Testons ;-)

WebThings.

- **Script : recordVIDEO.sh**

```
#!/bin/sh
heure=$(date +%H%M)
jour=$(date +%Y%m%d)
ffmpeg -rtsp_transport tcp -i rtsp://loli:grub@192.168.1.3:8888/unicast -vcodec copy -an -t 60
/home/pi/video/$jour$heure"_Video".mp4
# -rtsp_transport tcp :: cela résout le probleme d'images corrompues
# -i :: input file
# -vcodec copy :: copie directe du flux vidéo d'entrée ==> transcodage désactivé
# -an :: désactive le son
# -t 60 :: durée du flux vidéo en sortie
```

WebThings.


Configure run-program-adapter
programs
name*
What to call this (needs to be unique)
recordVIDEO
program*
Program to run (with arguments)
/home/loligrub/Vidéos/recordVIDEO.sh
Apply

WebThings.

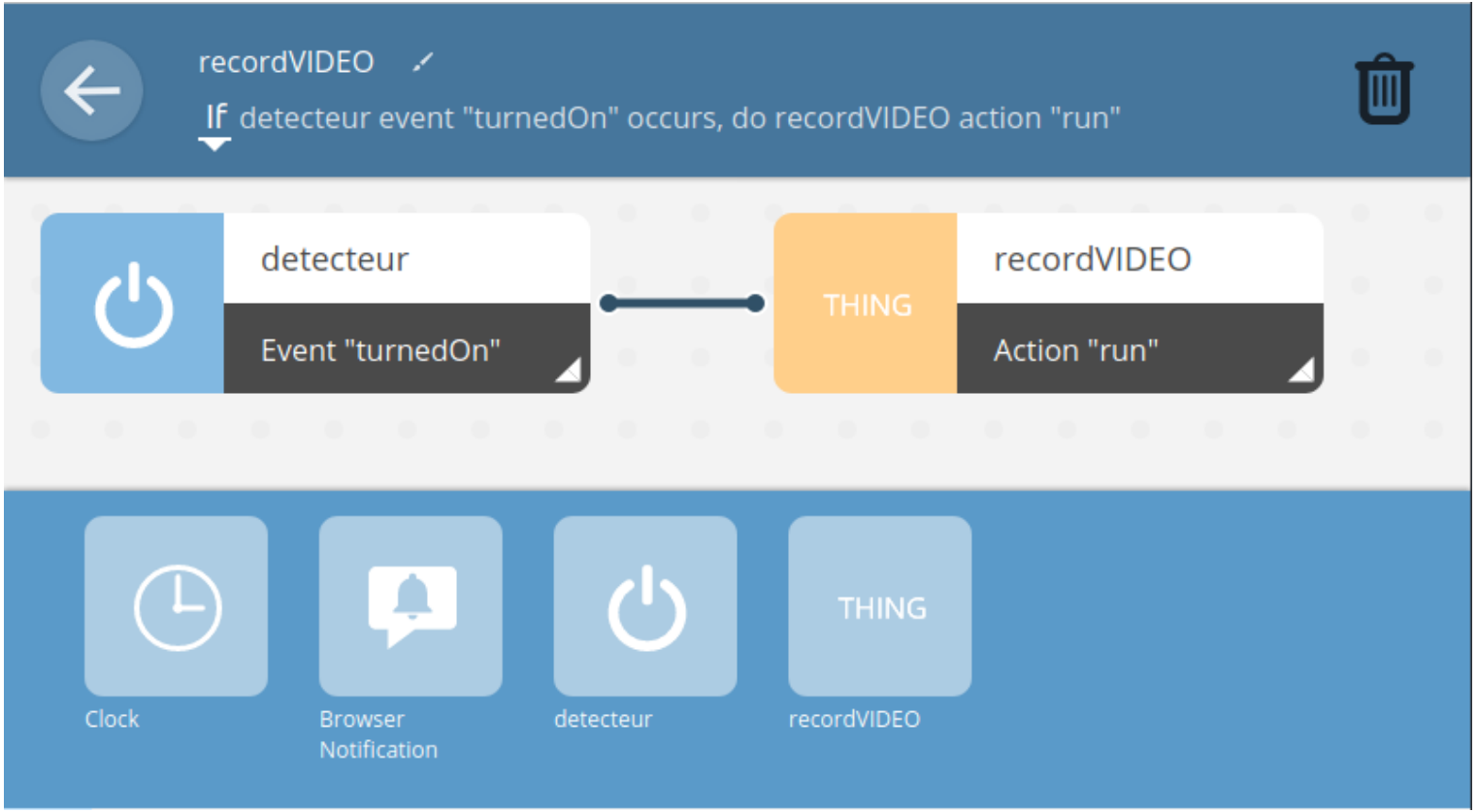

recordVIDEO
If detecteur event "turnedOn" occurs, do recordVIDEO action "run"
detecteur
Event "turnedOn"
THING
recordVIDEO
Action "run"
Clock
Browser Notification
detecteur
THING
recordVIDEO

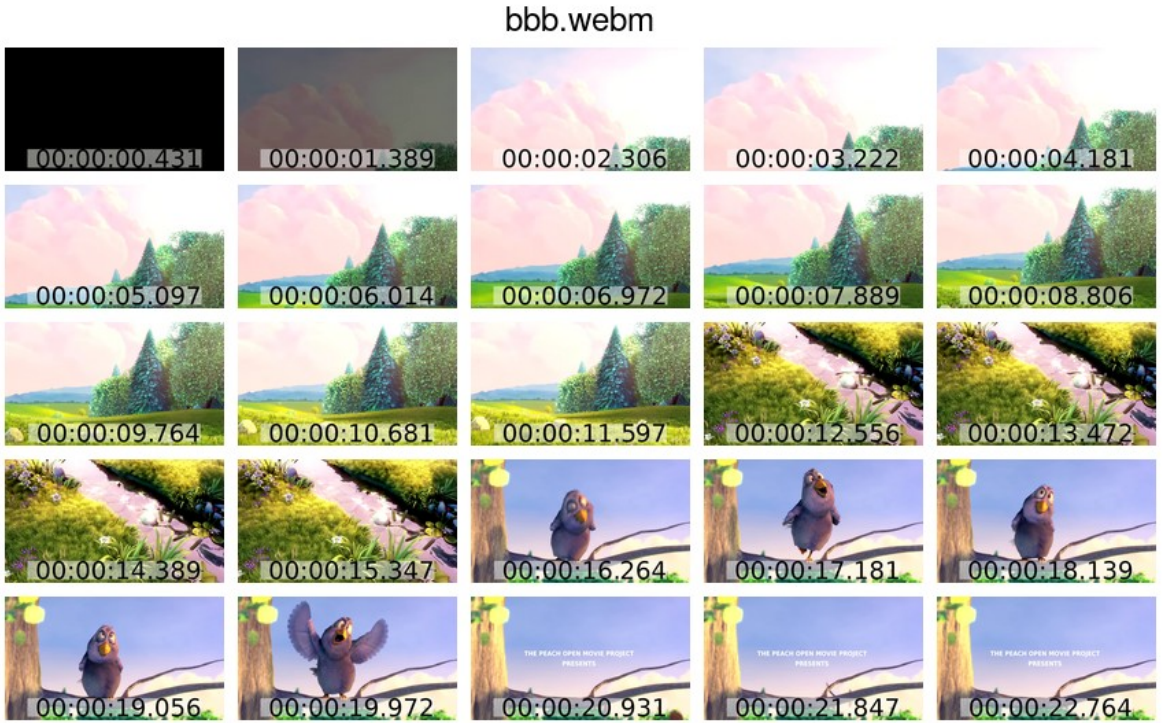
Planche-contact.

Créer une planche-contact d'une vidéo sous Linux :

https://www.arsouyes.org/blog/2021/2021-06-14_Miniatures_videos

```
sudo apt install ffmpeg imagemagick
```

./miniatures.sh bbb.webm 5 5



miniatures.sh

```
#!/bin/sh

if [ $# -lt 3 ]
then
    echo "Introduire 3 paramètres:le fichier vidéo, le nbr de colonnes et le nbr de lignes"
    exit
fi

length=`ffprobe -i $1 -show_entries format=duration -v quiet -of csv="p=0"`

freq=`echo "(($2*$3/$length))" | bc -l`

ffmpeg  -i $1 -vf drawtext="text='timestamp: %{pts \: hms}':x=(w-text_w)/2:y=h-th-
10:box=1:fontcolor=black:boxcolor=white@0.5:fontsize=(h/5)",fps=$freq -vcodec png /tmp/capture-%03d.png

for i in  /tmp/capture-*.png; do name=`basename $i`; convert $i -resize 160x120  /tmp/resized-$name; done

montage -title "$1" -tile $2x$3 -geometry +$2+$3  /tmp/resized-*.png $1.png

rm  /tmp/capture-*  /tmp/resized-*
```

Webthings

Fin de l’exposé!

Le bar va bientôt ouvrir !! Cool ;-)