

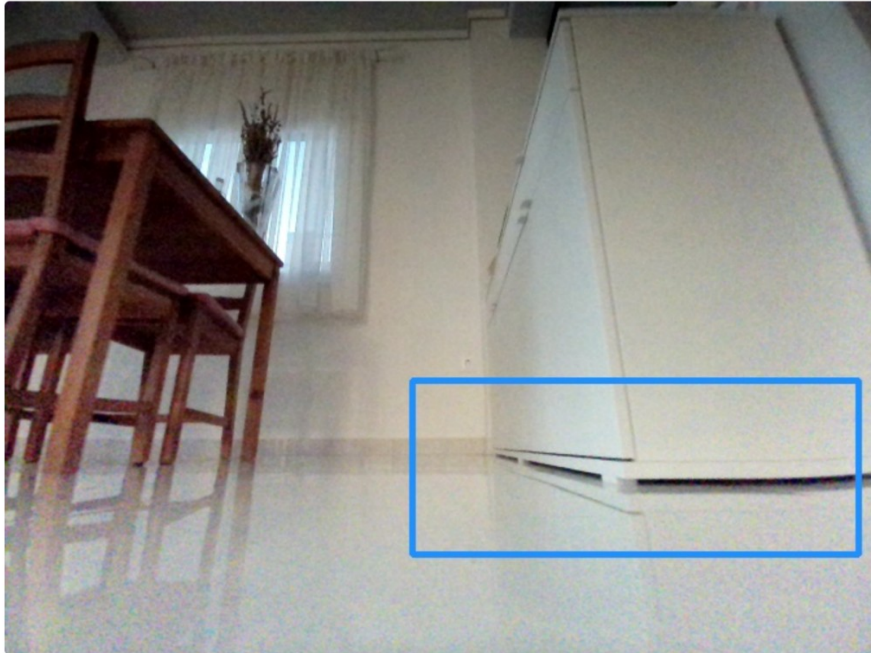
Reading v4l2 data from MR813
devices in a way that doesn't suck

Vivia Nikolaidou
2025-10-23

So, I got a new robot vacuum...

With a camera, for AI obstacle recognition

Obstacle Information



Bathroom Scale (67%)

Obstacle Information



Cable (82%)

... and of course I rooted it

Apparently the robots use video4linux and expose /dev/videoX devices?

(2023-08-13 DEFCON 31 – Dennis Giese)

https://dontvacuum.me/talks/DEFCON31/DEFCON31-Vacuum_robot_security_and_privacy-Prevent_your_robot_from_sucking_your_data.html



What would a GStreamer developer think?

First investigation

Trying to run `v4l2-ctl -all`

```
<3>[ 78.905482] [VIN_ERR]v4l2 sub device scaler get_selection  
error!  
<4>[ 78.905509] [VIN_WARN]v4l2 sub device g_parm fail!  
<1>[ 78.905820] Unable to handle kernel NULL pointer dereference at  
virtual address 000000b8
```

... Off to a good start, I see.

Trying GStreamer's v4l2src

```
[root@p2114_release:/data/gstreamer/install]# gst-launch-1.0 v4l2src ! video/x-raw,width=1632,height=1224,format=NV21,framerate=15/1 ! fakesink

[...]
```

0:00:00.049841417 3308 0x1ca97630
ERROR v4l2
gstv4l2object.c:2009:gst_v4l2_object_get_interlace_mode: Driver bug detected - check driver with v4l2-compliance from <http://git.linuxtv.org/v4l-utils.git>
[several lines of driver bug detected]

Patching GStreamer's v4l2src

```
[root@282d9bee8aa1 v4l2]# git diff --stat .
subprojects/gst-plugins-good/sys/v4l2/ext/videodev2.h      |    4 + -
subprojects/gst-plugins-good/sys/v4l2/gstv4l2allocator.c   |   20 +++--
subprojects/gst-plugins-good/sys/v4l2/gstv4l2object.c      |  285 ++++++
+++++
+-----
subprojects/gst-plugins-good/sys/v4l2/meson.build          |    2 + -
subprojects/gst-plugins-good/sys/v4l2/v4l2_calls.c         |   13 +++-
5 files changed, 193 insertions(+), 131 deletions(-)
```

... And it was still far from working...

Why not write a new element?

Reference implementation camerademo.c
existed in upstream TinaLinux repositories

https://github.com/CrealityTech/sonic_pad_os/blob/7f37a7fbf4ad59907034715e3d967b9177de1cd4/package/allwinner/camerademo/src/camerademo.c

Traced back the exact v4l2 call
sequence

... And dustsrc was born!



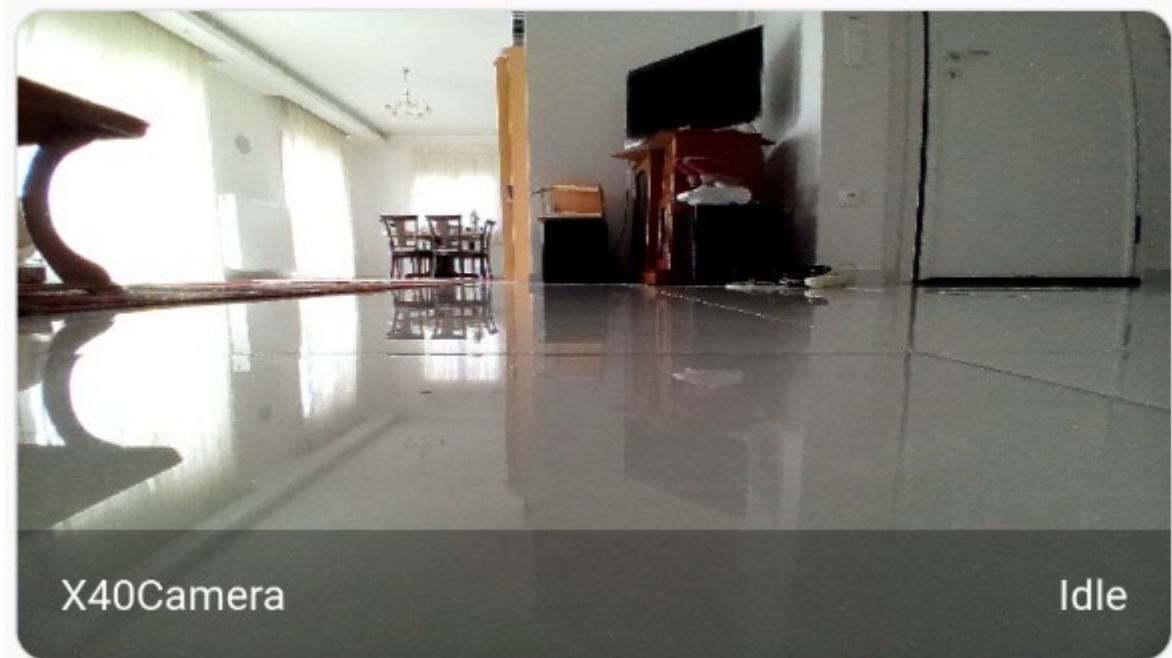
Debugging remotely

Protect against
kernel crashes



It works!

- x264 and test-launch RTSP pipeline
- 640x480 @ 2fps, ~7% CPU usage
- Home Assistant integration
- Internal obstacle detection still works!



Code still WIP

<https://gitlab.freedesktop.org/vivia/gstreamer/-/commits/dustsrc>

