Sony IMX296 on Horizon Robotics RDK X5 - Current Integration Status

1. Project

Integration of the Sony IMX296 Global Shutter sensor on a Sunrise Horizon RDK X5.

Target configuration:

- Sony IMX296
- Global Shutter
- Bayer RGGB
- RAW10
- 1456x1088
- 60 fps
- 1-lane MIPI CSI-2
- I2C6
- I2C address `0x1A`
- MIPI HOST 0 / PHY 0
- Camera connected to `cam0`
- `mipi_cam 15pin -> 22pin` adapter

The camera module has an external 54 MHz oscillator.

---

2. What already works

Hardware

Working chain:

```
IMX296
  -> I2C6 / 0x1A
  -> MIPI HOST0 / PHY0
  -> VIN
  -> RAW
```

I2C, MIPI and VIN have been verified by an actual frame capture.

Sensor utility

The SDK contains:

`hbre/camsys/libcam/src/sensor/imx296/`

with:

```
Makefile
version.mk
```

imx296_utility.c
inc/imx296_setting.h

Register tables

imx296_setting.h contains:

imx296_init_setting[]
imx296_stream_on_setting[]
imx296_stream_off_setting[]
imx296_inck_54mhz_setting[]
imx296_inck_37p125mhz_setting[]

imx296_init_setting[] contains:

0x308D
0x308E
0x308F
0x3204
0x3205

which cover SHS1 and Gain registers.

sensor_module_t

The end of imx296_utility.c contains a valid sensor_module_t with:

module
init
start
stop
deinit
power_on
power_off
aexp_gain_control
aexp_line_control
userspace_control

The structure matches the interface used by the existing working sensors.

.identify has NOT been added because the current API and the working reference sensor do not establish that it is required.

---

3. libimx296.so builds successfully

Generated:

out/deploy/hbre/lib/sensor/libimx296.so.1.0.0

Installed on the RDK X5:

```
/usr/hobot/lib/sensor/libimx296.so
/usr/hobot/lib/sensor/libimx296.so.1
/usr/hobot/lib/sensor/libimx296.so.1.0.0
```

SHA256

WSL:

```
e1afa84c2fb20bc6b789abef597b56dcd6d11518e40afc9ed008977d0c6c3ebd
```

RDK:

```
e1afa84c2fb20bc6b789abef597b56dcd6d11518e40afc9ed008977d0c6c3ebd
```

Therefore the installed library is byte-for-byte identical to the SDK build output.

Exports

The command:

```
nm -D /usr/hobot/lib/sensor/libimx296.so.1.0.0 | grep -i imx296
```

returns:

```
0000000000012c98 D cammod_imx296
0000000000013288 D imx296
```

`readelf` also shows `cammod_imx296`, `imx296`, `imx296_utility.c`, and `imx296_init_setting`.

Dependencies

`ldd /usr/hobot/lib/sensor/libimx296.so.1.0.0` contains no `not found` entries.

Conclusion: the `.so` is built, installed, ARM64, exports the expected sensor objects, and has resolvable dependencies.

---

4. `vp_sensors` is already integrated

The following exists:

```
app/samples/platform_samples/vp_sensors/imx296/
    linear_1456x1088_raw10_60fps_1lane.c
```

and the corresponding configuration exists under:

```
app/samples/platform_samples/sunrise_camera/Platform/x5/vp_sensors/imx296/
```

Mode parameters:

```
SENSOR_FPS = 60
mipiclk = 1188
linelenth = 1760
framelenth = 1125
config_file = linear_1456x1088_raw10_60fps_1lane.c
```

`vp_sensors.c` contains the corresponding `extern` declaration and list entry.

---

5. get_vin_data works

On the RDK:

```
/app/multimedia_samples/sample_vin/get_vin_data/get_vin_data
```

shows:

```
index: 42  sensor_name: imx296
config_file:linear_1456x1088_raw10_60fps_1lane.c
```

Running:

```
get_vin_data -s 42
```

produces:

```
Using index:42 sensor_name:imx296
Searching camera sensor on device:
i2c bus: 6
mipi rx phy: 0

INFO: Found sensor_name:imx296
on mipi rx csi 0
i2c addr 0x1a
config_file:linear_1456x1088_raw10_60fps_1lane.c
```

A real RAW frame was then captured:

```
resolution: 1456x1088
stride: 2912
size: 3168256
```

This proves:

```
I2C detection       OK
Sensor detection    OK
MIPI                OK
VIN                 OK
RAW capture         OK
```

---

6. RAW frame

Captured file:

`handle_34661_chn0_1456x1088_stride_2912_frameid_1_ts_9818472889056.raw`

Size:

`3168256 bytes`

The RAW frame was successfully extracted from the RDK and transferred for further processing.

---

7. The remaining problem is only in Python Camera / libsrcampy

The RDK uses:

`/usr/local/lib/python3.10/dist-packages/hobot_vio/libsrcampy.so`

and:

`/usr/lib/libsrcampy.so`

`ldd` of `libsrcampy.so` shows dependencies including:

```
libcam.so.1
libvpf.so.1
libhbmem.so.1
libmultimedia.so.1
libgdcbin.so
```

The package owner of `libcam.so.1` is:

`hobot-multimedia`

verified with:

```
dpkg -S /usr/hobot/lib/libcam.so.1
```

---

8. Python autodetection does not see IMX296

When using:

```
cam.open_cam(0, -1, 60, [1456, disp_w], [1088, disp_h])
```

runtime reports:

```
Searching camera sensor on device:
/proc/device-tree/soc/cam/vcon@0
```

```
i2c bus: 6
mipi rx phy: 0
```

It then probes a list of supported sensors such as:

```
sc1330t
irs2875-tof
sc230ai
sc132gs
sc035hgs
ov5640
f37
imx415
sc202cs
imx219
imx477
ov50h40
ox05b1s
sc850sl
shw3g
...
```

But `imx296` never appears in the sensor probe output.

After the final candidates, runtime reports:

```
No camera sensor found
```

This is the key symptom.

---

9. `video_idx` does not solve the problem

Trying different `video_idx` values still results in failure, for example:

```
video_idx=2 ret=-1
video_idx=3 ret=-1
```

The Camera API still performs the same sensor probing sequence.

Therefore the `get_vin_data` sensor index `42` should not be treated as the same thing as the Python `video_idx`.

---

10. Manual replacement of `libcam.so` was tested

The built `libcam.so.1.1.0` was copied to the RDK and the symlinks were recreated.

The runtime links became:

```
/usr/hobot/lib/libcam.so
    -> libcam.so.1
    -> libcam.so.1.1.0
```

Python behavior did not change.

Because the SHA256 of the WSL and RDK copies is identical, reinstalling this same `libcam.so` does not solve the issue.

---

11. Sensor Makefile

`hbre/camsys/libcam/src/sensor/Makefile` automatically includes sensor directories through a construct equivalent to:

```
SUB_ALL = $(filter-out $(SUB_IGNORE),$(notdir $(shell find . -mindepth 1 -maxdepth 1 -type d
```

Therefore directories such as:

```
sensor/sc1330t/
sensor/imx219/
sensor/imx477/
sensor/imx296/
```

are handled by the same build framework unless explicitly excluded.

The Makefile also installs sensor shared libraries into:

```
/usr/hobot/lib/sensor/
```

Therefore the hypothesis that IMX296 was simply forgotten in the common sensor Makefile is not supported by the current evidence.

---

12. `libimx296.so` is physically present in the runtime sensor directory

On the RDK:

```
/usr/hobot/lib/sensor/libimx296.so
/usr/hobot/lib/sensor/libimx296.so.1
/usr/hobot/lib/sensor/libimx296.so.1.0.0
```

The exported objects include:

```
cammod_imx296
imx296
```

So the problem is not simply a missing file or a missing `sensor_module_t` object.

---

13. Camera Framework dynamic loading was found in source

The Camera Framework contains a dynamic module loader using:

```
dlopen(so_name, RTLD_LAZY);
dlsym(so_fd, sym_name);
```

Therefore the framework is capable of dynamically loading sensor modules.

The unresolved questions are now:

```
Who creates so_name?
Who creates sym_name?
Where is the runtime sensor-module list generated?
Why is libimx296.so not becoming a Python Camera detection candidate?
```

---

14. `strace` investigation

A runtime trace was performed with `strace`.

It confirmed that `/usr/hobot/lib/sensor` is part of the dynamic loader search path and that many dependencies are searched there.

However, the filtered trace did not show explicit access to:

```
/usr/hobot/lib/sensor/libimx296.so
/usr/hobot/lib/sensor/libimx219.so
/usr/hobot/lib/sensor/libsc1330t.so
```

Therefore ordinary `openat` tracing did not provide a definitive answer about internal sensor-module `dlopen()` operations.

---

15. `ltrace` is not available

On the RDK:

```
which ltrace
```

returns nothing.

Therefore `ltrace -e dlopen,dlsym` could not be used without installing an additional package.

---

16. `LD_DEBUG` investigation

The following was executed:

```
LD_DEBUG=libs,files \
python3 test_cam.py > /tmp/lddebug.log 2>&1
```

The output confirms that:

`/usr/hobot/lib/sensor`

is present in the `LD_LIBRARY_PATH` search path.

The loader searches that directory for dependencies such as:

```
libm.so.6
libc.so.6
libhbspdev.so
libavformat.so.58
libavcodec.so.58
...
```

However:

```
grep -n "imx296" /tmp/lddebug.log
```

returns nothing.

This is an important clue but is not, by itself, final proof that `libimx296.so` is never passed to an internal `dlopen()` call.

---

17. Current non-blockers

Do NOT change the following without evidence:

```
DTS
I2C6
I2C address 0x1A
MIPI HOST0 / PHY0
X5 MCLK 24 MHz
external 54 MHz clock
vp_sensors.c
get_vin_data
IMX296 register tables
```

Do not add `.identify` unless the actual SDK ABI proves that this field is required.

Do not keep reinstalling the same `libcam.so`.

Do not start ISP tuning before the Python Camera detection problem is solved.

---

18. MCLK / 54 MHz

The IMX296 camera module has an external 54 MHz oscillator.

The sensor settings already contain:

`imx296_inck_54mhz_setting[]`

The 54 MHz clock should therefore be treated as the sensor's external clock.

There is currently no evidence that the RDK's base 24 MHz MCLK must be changed to 54 MHz.

The fact that `get_vin_data` already captures RAW successfully strongly suggests that the existing physical clock arrangement is working.

---

19. ISP is not the current blocker

An `imx296_tuning.json` file is not yet available.

However, the absence of the tuning JSON does not prevent:

```
I2C
MIPI
VIN
RAW capture
```

ISP tuning is a later stage:

```
RAW
 -> ISP
 -> Bayer
 -> AWB
 -> CCM
 -> NR
 -> NV12
 -> YOLO
```

---

20. Exact current project stage

The project is currently blocked at:

`RUNTIME CAMERA MODULE REGISTRATION / AUTO-DETECTION`

The project is NOT blocked at:

```
hardware
I2C
MIPI
```

```
VIN
sensor utility
RAW capture
```

Those stages are already working.

---

21. Core problem

The current behavior is:

```
get_vin_data
    -> vp_sensor_config_list
    -> IMX296 detected
    -> RAW works

BUT

Python libsrcampy
    -> Camera Framework
    -> runtime sensor module set
    -> IMX296 is not a detection candidate
```

This remains true even though:

```
libimx296.so exists
cammod_imx296 exists
imx296 sensor_module_t exists
libimx296.so is in /usr/hobot/lib/sensor/
all dependencies resolve
```

---

22. Exact next investigation

The next task is to find the runtime registration mechanism around `camera_module.c`.

Search the source tree for:

```
grep -RniE \
"so_name|sym_name|cammod_|dlopen|dlsym|sensor.*module|module.*sensor" \
 hbre/camsys/libcam/src
```

The target is to find:

```
module_list[]
sensor_module_list[]
camera_module_list[]
```

or any code that generates:

```
lib<sensor>.so
cammod_<sensor>
```

---

23. Desired successful runtime sequence

After the missing registration issue is fixed, the target runtime is:

```
Python
  -> Camera 0
  -> MIPI HOST0
  -> I2C6 / 0x1A
  -> Sensor Name: imx296
  -> chip-ID check
  -> sensor_init()
  -> MIPI stream
  -> VIN
  -> 1456x1088
  -> 60 fps
  -> get_img()
```

Then move on to:

```
ISP tuning
 -> NV12
 -> YOLOv5s
```

---

24. Status table

Stage	Status
Hardware connection	DONE
I2C6 / 0x1A	DONE
MIPI HOST0 / PHY0	DONE
External 54 MHz clock	DONE
IMX296 sensor utility	DONE
`sensor_module_t`	DONE
`libimx296.so`	DONE
`cammod_imx296`	DONE
`vp_sensors`	DONE
`get_vin_data`	DONE
RAW 1456x1088	DONE
Python `libsrcampy`	BLOCKED
IMX296 in Python autodetection	BLOCKED

Stage	Status
`open_cam()`	BLOCKED
ISP tuning	PENDING
NV12	PENDING
YOLO	PENDING

---

25. Explicit request to the RDK developer

Please help identify the exact runtime registration/list mechanism used by the Python Camera Framework (`libsrcampy` / `libcam`) and explain why an already-installed sensor module:

`/usr/hobot/lib/sensor/libimx296.so`

with exported objects:

```
cammod_imx296
imx296
```

is not included in the automatic sensor-detection candidates.

Specifically, please identify:

1. The source file that defines the runtime sensor-module list.
2. The source or configuration that generates `so_name` and `sym_name` for `dlopen()` / `dlsym()`.
3. Whether an additional registration table, package, configuration file, or build step is required for a new sensor to appear in Python `srcampy.Camera.open_cam()` autodetection.
4. The exact files/package that must be modified and rebuilt to complete IMX296 support.

Please do not treat the already-working `get_vin_data` path as the unresolved part; it already detects and captures IMX296 successfully.

---

26. Important rule for further investigation

Do not repeat already-tested assumptions such as:

```
"the problem is .identify"
"the problem is 24 MHz"
"the problem is DTS"
"the problem is board_config.json"
"libimx296.so is missing"
"sensor_module_t is missing"
"the entire autodetect must be a closed HBN binary"
```

These are either disproven or not established by the current evidence.

The confirmed fact is:

> The low-level IMX296 integration is working, but the Camera Framework used by Python `libsrcampy` does not expose IMX296 as an automatic sensor-detection candidate, even though the corresponding sensor shared library is present and exports `cammod_imx296` and `imx296`.

That runtime registration/autodetection boundary is the exact point that must now be resolved.