

NAO V6 hints

LoLA

- The signs of the IMU measurements now follow a sign convention that matches the robot coordinate system and a right-hand axis convention (which was not the case before, e.g. the gyro around the z axis was flipped).

Camera

- The camera is a Leopard Imaging OV5640 that is connected via USB. Therefore, the uvcvideo driver is used.
- A datasheet of the camera sensor exists here:
https://cdn.sparkfun.com/datasheets/Sensors/LightImaging/OV5640_datasheet.pdf
- The camera firmware supports UVC extension units in addition to the standard USB control settings. These have to be used to control some settings, especially flipping the image of the upper camera.
 - With Video4Linux, this is the documentation on how to get/set XU controls:
<https://linuxtv.org/downloads/v4l-dvb-apis/v4l-drivers/uvcvideo.html>
 - The extension unit ID is always 3.
 - The selector can have the following values: 1, 2, 3, 7, 8, 9, 10, 11, 12, 13, 14. They have the following meanings:

Selector	Size	Minimum	Maximum	Meaning
1	2	0	255	unknown
2	2	0	255	unknown
3	2	22	22	Version of the firmware. Read-only.
7	20			unknown
8	2	0	65536	Average luminance.
9	17			Weight table for the auto exposure mechanism. See below for the meaning of the 9 bytes.
10	1	0	1	Enables a test pattern.
11	6			unknown
12	2	0	1	Horizontal flip.
13	2	0	1	Vertical flip.
14	5			Interface to directly access camera registers. See below for the meaning of the 5 bytes.

- The auto exposure weight table is represented as follows (see figure 4-6 on page 47 of this [datasheet](#)):

Index	Meaning
0	Enable. Should always be 1.
1	Upper four bits of the x coordinate of the auto exposure window start.
2	Lower eight bits of the x coordinate of the auto exposure window start.

Index**Meaning**

- 3 Upper three bits of the y coordinate of the auto exposure window start.
 - 4 Lower eight bits of the y coordinate of the auto exposure window start.
 - 5 Upper four bits of the x coordinate of the auto exposure window end.
 - 6 Lower eight bits of the x coordinate of the auto exposure window end.
 - 7 Upper three bits of the y coordinate of the auto exposure window end.
 - 8 Lower eight bits of the y coordinate of the auto exposure window end.
 - 9 The upper four bits contain the weight of window 01, the lower four bits window 00.
 - 10 The upper four bits contain the weight of window 03, the lower four bits window 02.
 - 11 The upper four bits contain the weight of window 11, the lower four bits window 10.
 - 12 The upper four bits contain the weight of window 13, the lower four bits window 12.
 - 13 The upper four bits contain the weight of window 21, the lower four bits window 20.
 - 14 The upper four bits contain the weight of window 23, the lower four bits window 22.
 - 15 The upper four bits contain the weight of window 31, the lower four bits window 30.
 - 16 The upper four bits contain the weight of window 33, the lower four bits window 32.
- The direct register access control layout is as follows:

Index**Meaning**

- 0 Read (0) or write (1).
- 1 Upper eight bits of register address.
- 2 Lower eight bits of register address.
- 3 Upper eight bits of register value.
- 4 Lower eight bits of register value.

To read a register, one has to set the control first with the first byte set to 0 and the address in bytes 1 and 2 (bytes 3 and 4 can be left uninitialized). After that, the value of the control can be obtained using `UVC_GET_CUR` and the register value should be at bytes 3 and 4 of the data structure. It might be necessary to wait some time in between setting and getting the control.

- Example code for setting a extension unit control:

```
#include <linux/usb/video.h>
#include <linux/uvccvideo.h>
#include <sys/ioctl.h>
#include <stdint.h>

int set_uvc_xu(int device_fd, uint8_t extension_unit_id, uint8_t control_selector,
uint16_t size, uint8_t* data)
{
    struct uvc_xu_control_query query;
    query.unit = extension_unit_id;
    query.selector = control_selector;
    query.query = UVC_SET_CUR;
    query.size = size;
    query.data = data;
    return ioctl(device_fd, UVCIOC_CTRL_QUERY, &query);
}
```

- By default, timestamps use the monotonic clock instead of the realtime clock. This can be

changed using a module parameter (`/sys/module/uvccvideo/parameters/clock`). The timestamps denote the start of image exposure.

- Also note that hardware timestamps are disabled by default, but can be enabled (`/sys/module/uvccvideo/parameters/hwtimestamps`). It seems, although, that this has no effects.
- Sometimes it is not possible to open a camera device until rebooting. It might be possible that combinations of delaying and retrying work.
- Sometimes (very rarely) images are scrambled (i.e. rows of the image are swapped with others).