Hazardous Gas Leakage Detection using Nanosaur and Jetson Nano

Electronic Design Lab

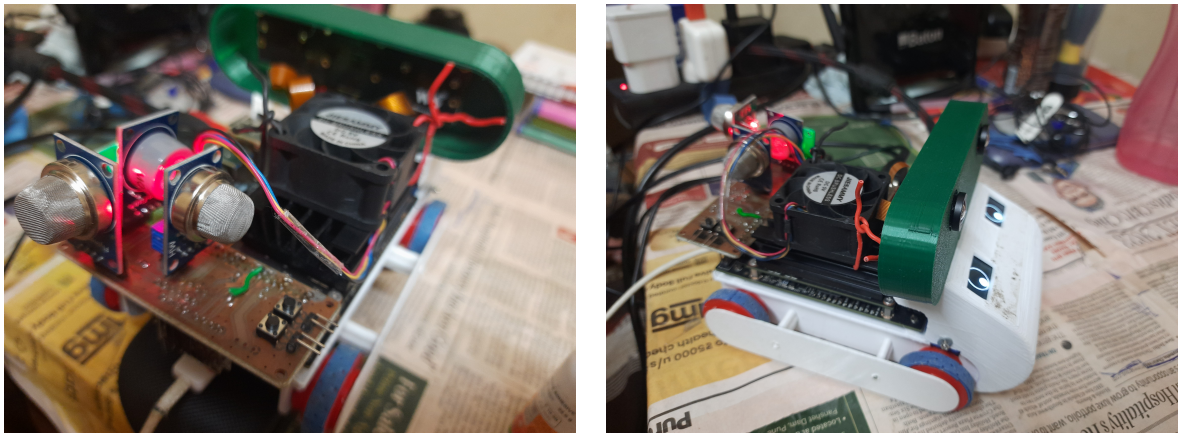
By

Tanmay Dokania : 200070083
Aziz Shameem : 20d070020
Navneet : 200070048

Under the guidance of
Prof. Siddharth Tallur
Prof. V Rajbabu

Department of Electrical Engineering
Indian Institute of Technology Bombay
April, 2023

Contents



(a) Back View (b) Front View

Figure 1: Final Functioning Nanosaur

1 Description of Project Design

Nanosaur is a simple open-source robot based on *NVIDIA Jetson*. The robot is fully 3D printable, able to wander on your desk autonomously and uses a simple camera and two OLEDs — these act as a pair of eyes.

In this project, we aim to integrate three gas sensors with the bot. These should be able to collect data in real time, and transmit them (wirelessly) to a remote device capable to plotting them (in real time).
Further, we aim to implement Visual Simulataneou Localisation and Mapping (VSLAM), using an onboard camera module, which is capable to a 3D reconstruction of its environment.

These two technologies can be combined to make the bot useful in a wide range of applications.

We managed to complete the desired objective satisfactorily, the obtained results and experiments, as well as information about the bot itself, is collected in this report.
Several video demonstrations are created. They are kept together for ease of access.

- Dual Camera Feed on the Nanosaur
- Inertial Measurement Unit data on the Nanosaur
- Demonstration of Keyboard Control and Remote Camera feed
- Modified ROSboard: Used to remotely view Camera feed and Sensor data
- Final Submission video with Subtitles
- Final Presentation slides

2 Bill of Materials

The bill of materials can found in Table 1.

Table 1: Bill of Materials

Item	Price
NVIDIA Jetson Nano	-
Wi-Fi Dongle 5Ghz	-
Pololu Micro Gearbox	-
Adafruit motor control	-
Ball bearings F686ZZ	-
IMX219-83 Stereo Camera, 8MP Binocular Camera Module For Depth Vision	5999/-
MQ135 - Air Quality Gas Sensor Module	103/-
MQ9 Carbon Monoxide, Methane and LPG Gas Sensor Module	129/-
MQ4 Gas Sensor Module	95/-
16 Bit I2C 4 Channel ADS1115 Module	479/-

3 Choice of Sensors

3.1 MQ135 - Air Quality Gas Sensor

The MQ-135 Gas sensor can detect gases like Ammonia (NH_3), sulfur (S), Benzene (C_6H_6), Carbon Dioxide (CO_2), and other harmful gases and smoke. Similar to other MQ series gas sensor, this sensor also has a digital and analog output pin. When the level of these gases go beyond a threshold limit in the air the digital pin goes high. This threshold value can be set by using the on-board potentiometer. The analog output pin, outputs an analog voltage which can be used to approximate the level of these gases in the atmosphere.

The choice of this sensor was motivated by the low power consumption(750mW) of the module, a decent supply voltage range (2.5V-5V), and the larger number of gases that the module can detect.

3.2 MQ9 - Carbon Monoxide, Methane and LPG Gas Sensor Module

MQ9 Gas Sensor is a Metal Oxide Semiconductor (MOS) type Gas Sensor of MQ Gas Sensors family involving MQ 2, MQ 4, MQ 3, MQ 135, etc. It is mainly used to detect gases like Carbon Monoxide, Methane, and Propane, etc. This MQ9 Smoke Sensor contains a sensing element, mainly aluminum-oxide based ceramic, coated with Tin dioxide (SnO2), enclosed in a stainless steel mesh. Whenever gas comes into contact with the sensing element, the resistivity of the element changes. The change is then measured to get the concentration of the gases present. It has a small heating element present which is needed to preheat the sensor to get it in the working window.

It can detect LPG, Propene, Hydrogen, Carbon Monoxide, and Methane Gas concentrations. It finds uses in detecting gas leakage in pipelines.

We decided to go ahead with this sensor since it has a power consumption of 750mW, which is quite low. The analog supply voltage requirement is +5V, and like the MQ135 sensor discussed before, it can detect a collection of gases.

3.3 MQ4 - Methane Gas Sensor Module

MQ4 Gas sensor is a Metal Oxide Semiconductor (MOS) type Gas Sensor mainly used to detect the Methane (CNG) gas concentration in the air either at home or in industry. This sensor contains a sensing element, mainly aluminum-oxide based ceramic, coated with Tin dioxide, enclosed in a stainless-steel mesh. Whenever gas comes into contact with the sensing element, the resistivity of the element changes. The change is then measured to get the concentration of the gases present. Its sensing range of 300-10000 PPM is suitable for gas leak detection. The combustion of Methane is highly exothermal means it will release a vast amount of heat if ignited which when used in a controlled way is beneficial but if an accident occurs it will be devastating.

The sensor has a power consumption of less than 750mW, and works at an operating voltage of 5V.

3.4 IMX219-83 Stereo Camera

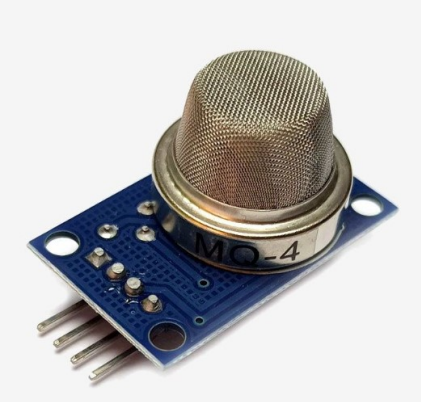
This is a binocular camera module which features dual IMX219 cameras onboard, 8Megapixels of each camera. It is suitable for AI vision applications like depth vision and stereo vision. The module supports NVIDIA Jetson Nano Developer Kit B01 version.

Since our deliverable needs a mechanism for localisation and mapping (SLAM), we decided to go ahead with a minimal stereo camera, which enables a good enough field of vision as well as depth perception. The proposed module contains two Sony IMX219 cameras, each having a resolution of 3280×2464 pixels. Further, the module also contains an Accelerometer, Gyroscope and Magnetometer. Being compatible with Jetson Nano(the board used inside Nanosaur), it will be easy to integrate this into the bot, and use it to implement V-SLAM techniques.

4 Block Diagram

Please find attached an overview of our deliverable as a block diagram. This is followed by relevant brief explanations of existing subsystems.

(a) MQ135



(b) MQ4

(c) MQ9

(d) IMX219-83 Stereo Camera

Figure 2: Sensors

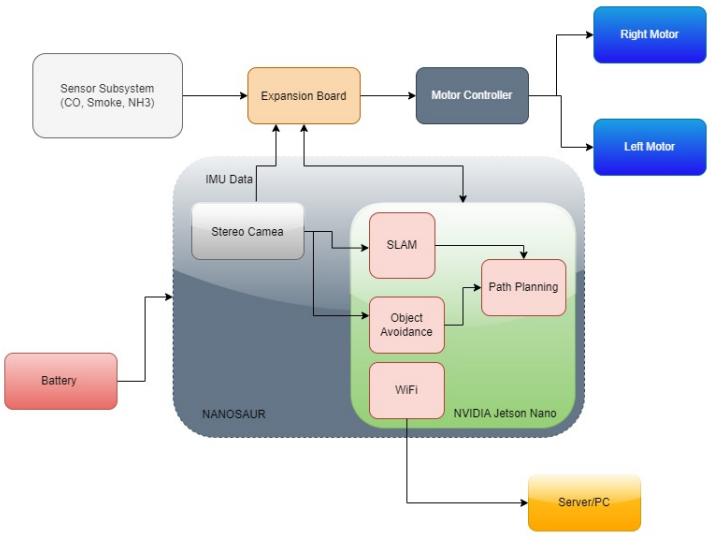


Figure 3: Block Diagram of the design

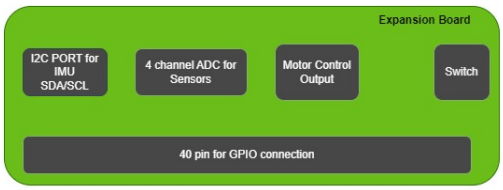


Figure 4: Block Diagram of the Expansion Board

5 Components of the design

5.1 Expansion Board

An expansion board or an expansion card is a circuit board or card that can be added to a computer adding new functionality. Expansion cards plug directly into a slot on the computer motherboard. Computer motherboards can have different types of expansion card slots such as PCI, AGP and PCI-Express slots. Mainboards are designed with a different number of expansion card slots; most computers can accommodate three or more expansion cards.

To run the base circuit of Nanosaur, we need an expansion board. We have designed a PCB that includes connections to and from Jetson Nano and the various sensors/ADC. The buzzer interfacing circuit is also held in the PCB. Further, the PCB holds two push-button switches that can be used as power and reset buttons, to run the bot.

The sensors and the Analog-Digital Converter are mounted on the PCB using connectors. The buzzer is also mounted in a similar way. However, the interfacing circuit consisting of a MOSFET and resistor are soldered. The camera/IMU feed is connected to Jetson Nano through the PCB.

5.2 Motor Control

The control and navigation block generates commands for the motor to rotate. A high-level controller generates the path-planning commands. This command will then be sent to the motor driver circuit. The motor driver generates the command for the brushed DC motor which has a metal gearbox.

5.3 Printed Circuit Board : Design and Testing

The PCB schematic is attached at the end of the report.
The general outline is as follows : The circuit enables attaching the three sensors and adc through pin-headers. Jetson nano is also attached, and the relevant connections required for the to and fro transfer of data and clock signals are made. The circuit also incorporates two push-button switches, which will be used for power and reset purposes.

The design is on a two-layer board, with the minimum trace width being 16 mils, and was printed by chemical etching at PCB Lab, IIT-B. The components were soldered, and the connections checked by probing the connectors using DMMs. It was made sure that there are no shorted connections between the ground and the power supply. Lastly, the set-up was tried on the ESP-12E board, and the obtained data confirmed the functioning of the three sensors, ADC, and the relevant portion of the PCB.

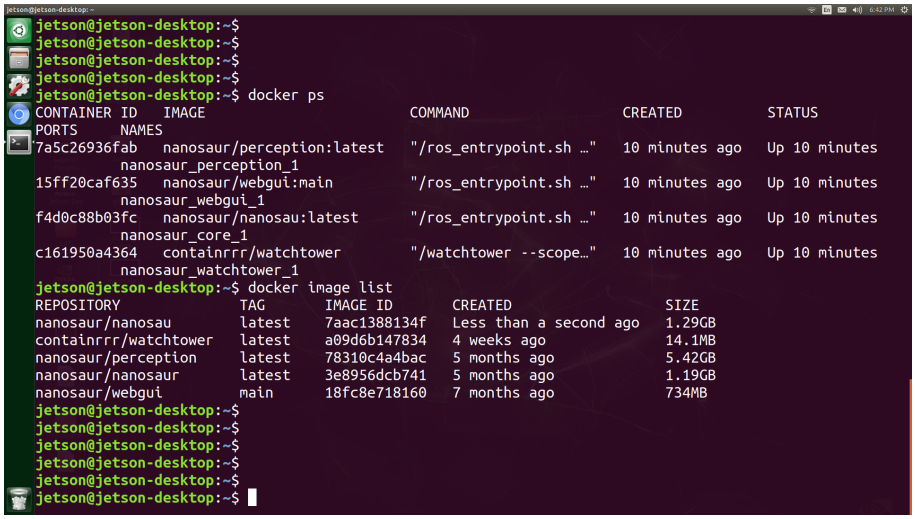


Figure 5: Basic Docker Commands on Nanosaur

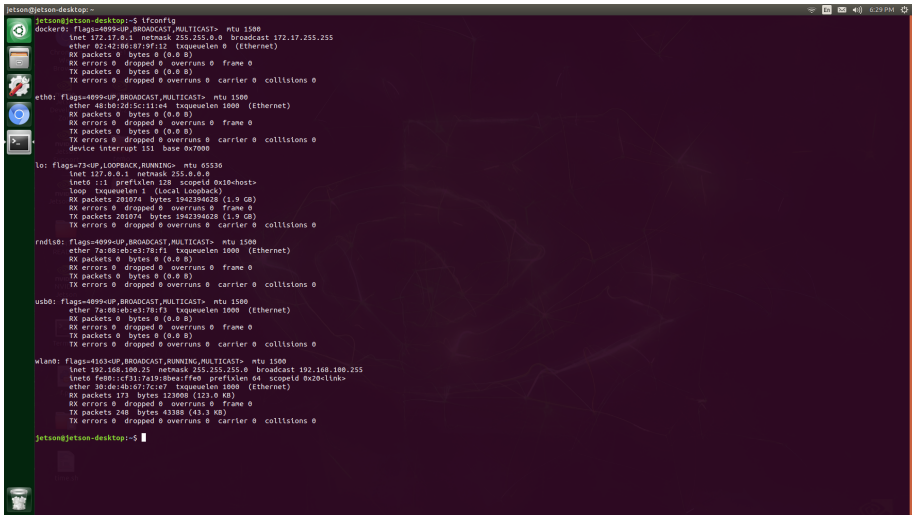


Figure 6: Command to find ip address of Nanosaur

5.4 ROS

We have successfully set up Nanosaur ROS2 environment on software, and have had it interfaced with Jetson Nano. Keyboard control has also been incorporated and tested via a remote PC on the assembled Nanosaur using the PCB.
One needs to learn about how `docker` and `ROS` function to be able to make modifications and implement them. To turn on the Nanosaur, one needs to make docker containers from the docker images, this is done via the command `nanosaur wakeup`. To close the Nanosaur's operations, one needs to stop the containers. This is done by `nanosaur down`.

5.5 Sensor Interfacing

For the Nanosaur, we initially decided to incorporate the sensors along with the Analog-to-Digital Converter on a separate PCB since these components could not find adequate space on the design of the expansion board. The ADC was to be de-soldered,

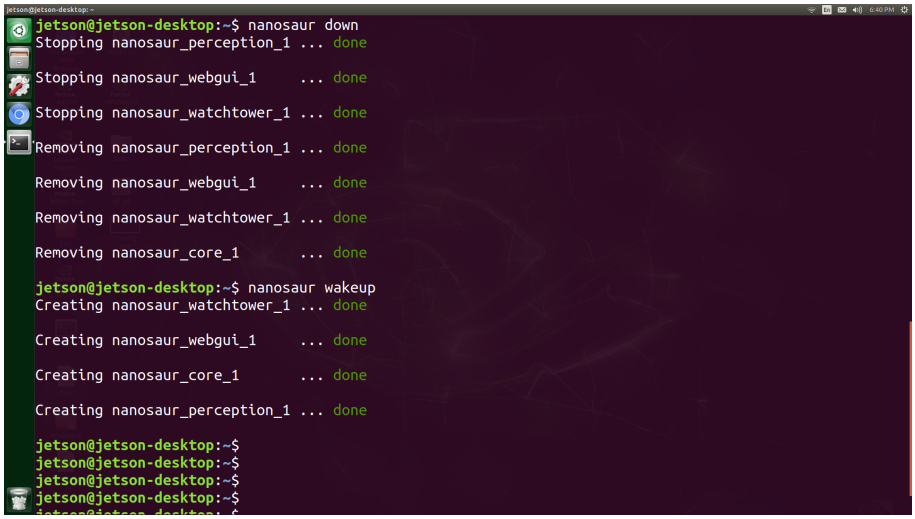


Figure 7: Turning Nanosaur On and Off

and the Integrated Circuit interfaced directly on the board. The sensors were to be used directly, along with their modules, and attached to the board via an array of connectors. However, after a discussion with Prof. Tallur, we are advised to have the expansion PCB itself handle the load of the three sensors and ADC. This is what we have implemented. The interfacing circuits and the components are operational and have been tested subsequently.
Currently, the sensor data (one at a time) is collected via the ADC and published by a ros node, which can be viewed on ros board. Further modifications will be required to obtain data from all three sensors simultaneously, as well as to make the visualization more readable.
To achieve this feat, we created a ROS Node to publish data on a ROS Topic with the INT64 datatype. Then this data was picked up by ROSBoard to view this time series plot remotely and wirelessly. To access ROSBoard, one needs to be on the same WiFi network as Nanosaur and go to the website: `"<ip-address>:8888"`. To find the IP address of Jetson Nano, one can run the command `"ifconfig"` on the terminal.

5.6 CAD

5.6.1 Migration to Solidworks

The available CAD of the Nanosaur was in .STL format, which cannot be modified easily. Hence, the files were converted to .SLDPRT, which is used in Solidworks. The CAD was assembled in Solidworks to enable better design and guarantee that the new parts would be compatible. Appropriate mates were added to assemble the converted CADs. Assembling the CAD gave insight into what modifications could be done to achieve the purpose better.

5.6.2 Wheel and Sprocket Design

The wheels in the initial design had a track, which is not possible for us to implement. Therefore, the wheels and the sprockets had to be redesigned by increasing the central diameter.

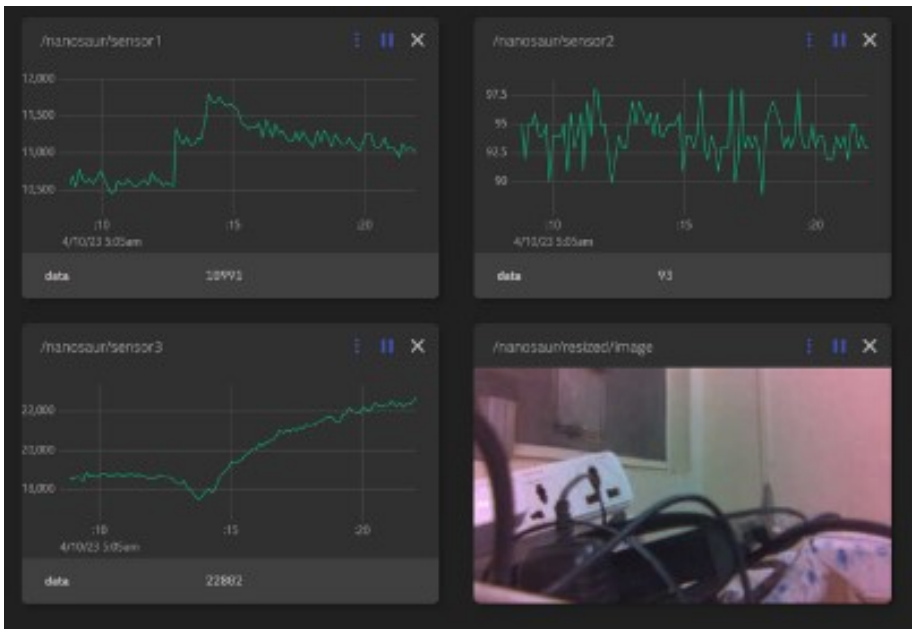
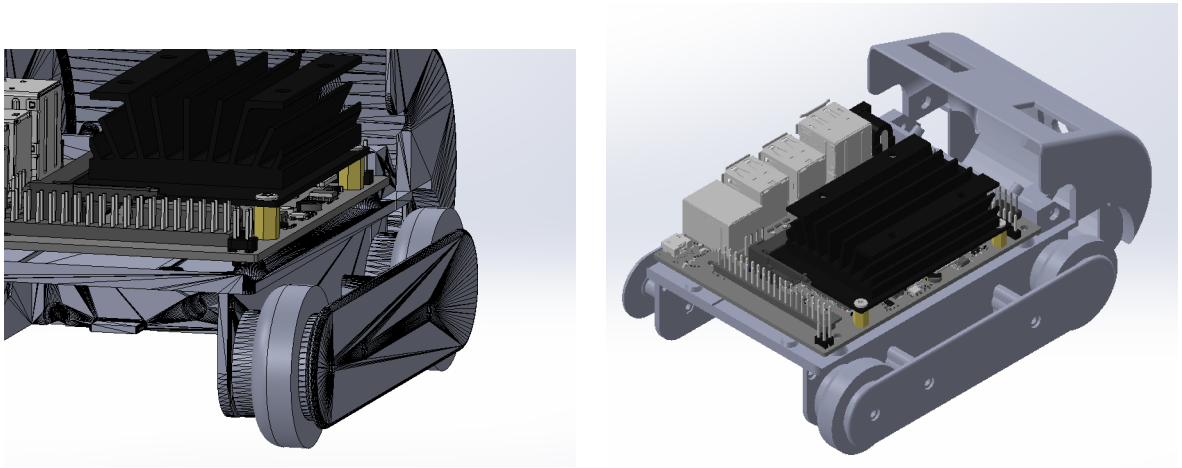


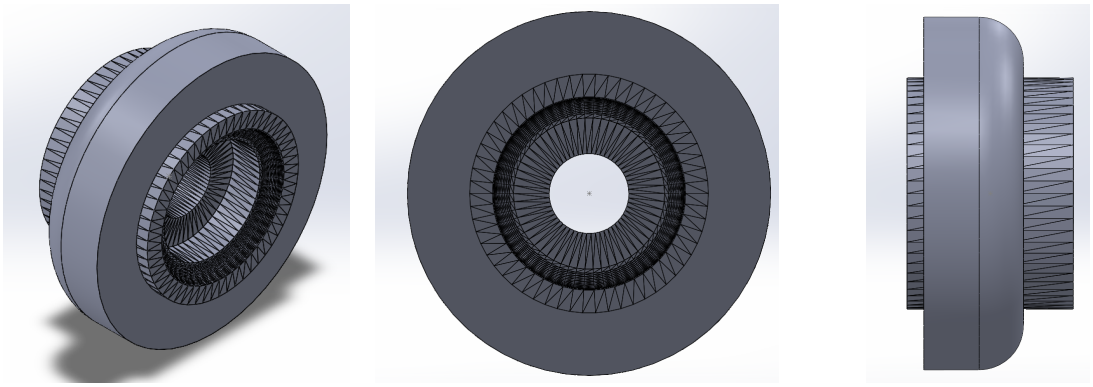
Figure 8: Remote Visualization of sensor data on ROSBoard(top and bottom left). Camera Feed (bottom right)



(a) Wheel Assembly View

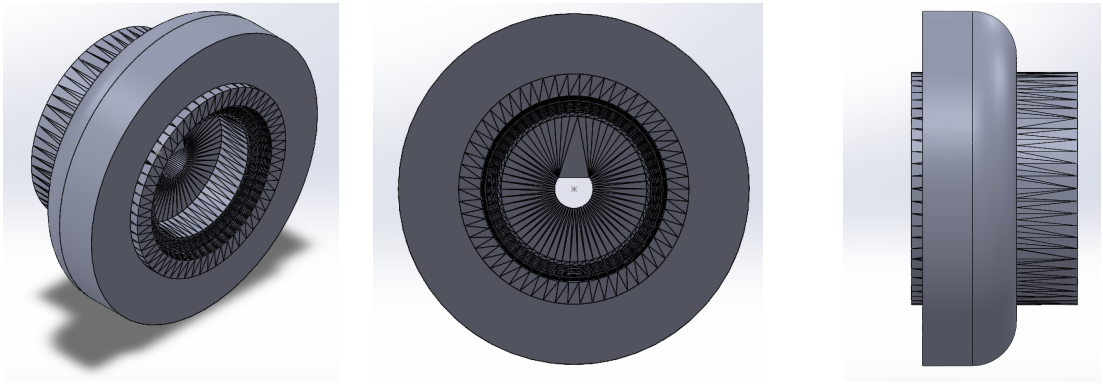
(b) Full Assembly Angled View

Figure 9: Assembly in Solidworks



(a) Angled View (b) Front View (c) Side View

Figure 10: Wheel Design



(a) Angled View (b) Front View (c) Side View

Figure 11: Sprocket Design

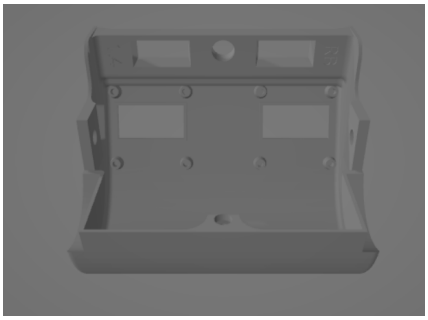
The design issue faced here was to increase the diameter larger enough and make its width larger, to ensure that the wheel could sustain the load and have ground clearance. But it causes the wheel to brush against the body. Hence, the wheel was designed with a fillet.
As the wheel is 3D-printed, the traction is less. Therefore, the wheel will now be filed to make it rough, or a rubber track will be added.

5.6.3 Camera Cover Design

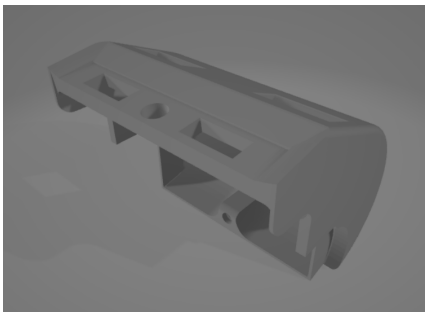
To mount the camera module, we created a CAD design after measuring the dimensions of the Nanosaur and the module. The main issue we faced was fixating the camera location and having a robust enough cover to hold it fixed.
To achieve this, the Nanosaur cover designed for use with Real-Sense was printed, and then a camera cover was designed to fit the pre-designed cover.

5.7 VSLAM

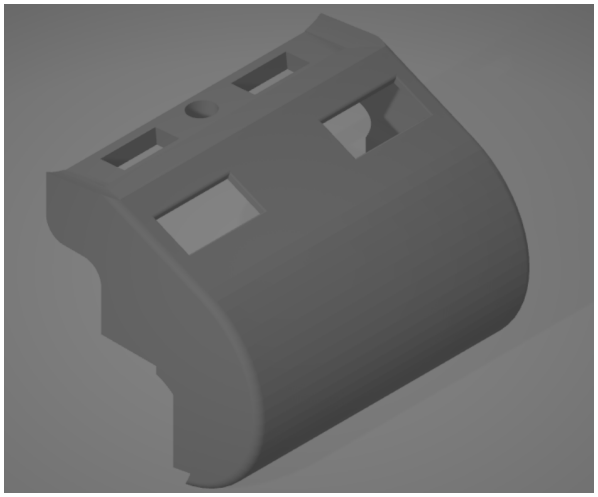
Visual simultaneous localization and mapping (vSLAM) refers to the process of calculating the position and orientation of a camera, with respect to its surroundings, while simultaneously mapping the environment. The process uses only visual inputs



(a) Bottom View

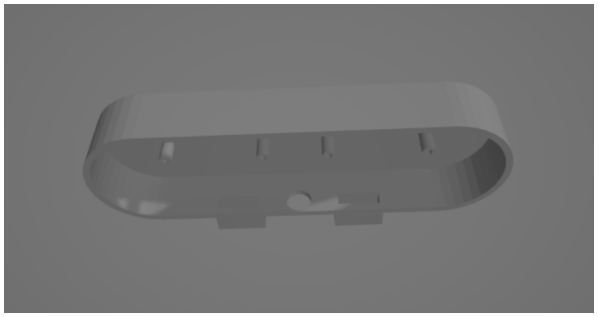


(b) Front View

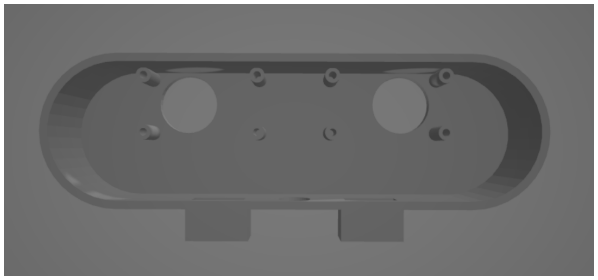


(c) Side View

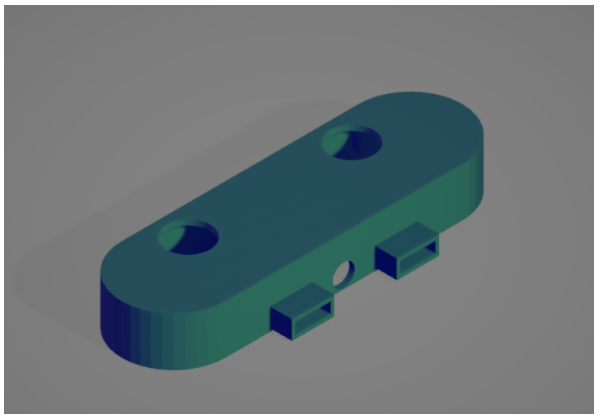
Figure 12: Cover Designed for Intel Real-sense



(a) Bottom View



(b) Front View



(c) Side View

Figure 13: Camera Cover Designed for IMX-219 Stereo Camera

from the camera.

In this project, we have attempted at obtaining a live map of the environment, uisng the feed from on-board cameras. This is accomplished using a framework – ORB-SLAM2 – which is build on OpenCV and uses images to identify *features* of its environment, which are then marked on a 3D plot (point-cloud map) to obtain a reconstruction.
To run ORBSLAM2, the following needs to be done:

1. To run the camera node, run the following command on terminal:
   `cd ~catkin_ws && catkin_make && source ~catkin_wsdevelsetup.bash`
2. Followed by, `roslaunch jetbot_pro csi_camera.launch`
3. Now, the camera node is running and publishing data on the corresponding ROS topic
4. Followed by,
   `export ROS_PACKAGE_PATH=${ROS_PACKAGE_PATH}` `pathtoORB_SLAM2_CUDAExamplesROS`
5. Followed by,
   `roslaunch pathtoORB_SLAM2_CUDAExamplesROSORB_SLAM2_CUDAlaunch/ros_mono.launch [bUseViewer:=(false by default)] [bEnablePublishROSTopic:=(true by default)]`

To install ORBSLAM2, the references used are:

- Waveshare Wiki: Install ROS System on Jetson Nano Environment Configuration
- GitHub Repository for ORBSLAM2 CUDA, enabling it to run on the GPU

5.8 Keyboard Control

The keyboard control was set up wirelessly. This has been tried and tested, and the bot seems to respond adequately to the keyboard command on the remote pc.
The motion of the bot was slightly hindered due to a lack of friction on the wheels, which led to slipping. This was rectified by adding rubber on the wheels, thereby increasing the grip.
To run keyboard control, one needs to do the following:

1. Follow the instructions present for "Install on Desktop."
2. Connect Nanosaur to a desired network; one might need a monitor for initial setup.
3. Make sure your Nanosaur is working by running `nanosaur wakeup`.
4. Connect your PC to the same network as Nanosaur.
5. Run the command `sudo source nanosaur` on your PC; using this one enters the Nanosaur environment.

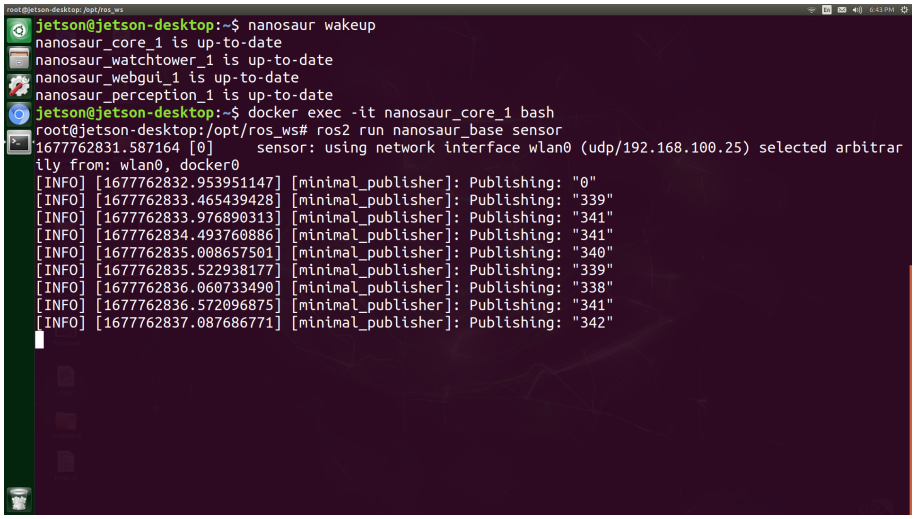


Figure 14: Running commands to enable the script written to publish data

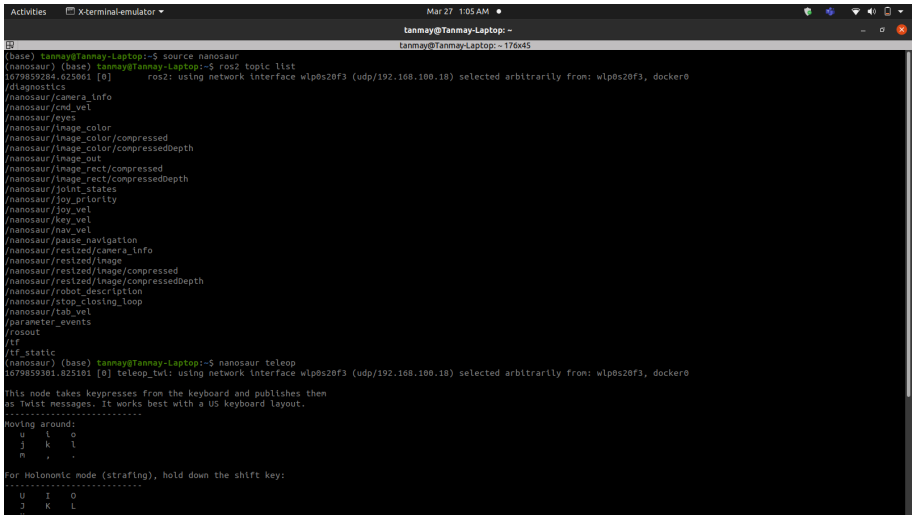


Figure 15: Running commands to achieve remote keyboard control

6. Run the command `ros2 topic list` on your PC; this prints all the ROS topics available on Nanosaur.
7. Run the command `nanosaur teleop` on your PC.
8. Follow the instructions on the screen to control Nanosaur.

The outcome will be similar to this : Demonstration of Keyboard Control and Remote Camera feed

6 Challenges Faced and Solutions

Some of the major challenges/roadblocks that we encountered during the course of the project are listed here, along with the correspond solutions employed :

1. **Sensor functioning and data collection** : Since all of the sensors are connected to Jetson Nano through a single Analog-Digital Converter (ADC), taking data

from all three at the same time was difficult. Finally, we ended up increasing the frequency of collection to it maximum allowable limit, and obtaining the data sequentially from the three sensors.
Further, plotting of data remotely and in real time posed an issue, since secure shell (ssh) does not allow data transfer as a continuous bit stream (ready for plotting). This was overcome by making use of **RosBoard**, a dashboard hosted on Jetson Nano and accessible from any device connected to the common network.

2. **ORBSLAM3 integration** : Due to software issues, particularly pertaining to the available RAM in the micro controller, integration of ORBSLAM was difficult. Several adjustments had to be made to get the package running, and obtain the point cloud.

3. **CAD Modifications** : It was not clear how we were going to place the sensors, and what modifications to the structure was required. Further, the camera module also had to be housed. Finally, it was decided to place the sensors directly, along with their modules, on the PCB. Careful design of the PCB was required to ensure optimal placements of the three modules.
Modifications were made to the CAD to mount the camera module at the front of the bot.

4. **Wheels** : The design of the wheels was such that, due to lack of traction, there was a lot of slipping while moving the bot, and the bot was not able to move or turn adequately. Rubber sheet was placed on the wheels to improve grip.

7 Future Work

Nanosaur, being a mobile robot capable of being controlled remotely, finds applications in several domains were human labour cannot reach, either because of the hazards involved or other forms of inaccessibility constraints. Further, we have shown that a group of sensors can easily be integrated and read from remotely, by modification to the Printed Circuit Board mounted on the bot. These traits make the robot use full in several areas. We identify some broad areas here :

1. **Gas Leakage Detector** : Identification of gas leakage in industrial areas. Can prove to be beneficial in providing constant surveillance, as well as mitigate the risks of toxic gas intake. The SLAM techniques implemented can be used to obtain a map of the environment, potentially localising the areas of leakage.

2. **Mobile sensor setup** : Any sensor, capable of providing measurements of a continuous quantity such as temperature, heat, pressure, signal strength etc. can be integrated with the bot and thus, avail all the functionalities the bot is capable of providing.

3. **Environment Mapping** : Particularly in inaccessible areas, the bot, being portable, light-weight and relatively small, can be used as a tool for mapping the environment.

4. **Remote Surveillance** : Can be used to provide constant, mobile surveillance using the on-board cameras, where the feed can be observed and/or recorded on a remote device.

8 Acknowledgements

We would like to extend our heartful gratitude to the instructor of the course, **Prof. Siddharth Tallur**, for giving us this oppurtunity and pushing us to out limits, without which none of what we managed to achieve would have been possible. We would like to thank all other instructors : **Prof. Joseph John**, **Prof. P.C. Pandey**, **Prof. Laxmeesha**, **Prof. V. Rajbabu** and **Prof. Anil Kottantharayil** for their time and efforts.

Secondly, we would like to thank the WEL staff, including **Maheshwar sir**, **Ankur sir**, and everyone else working behind the scenes to make all of this possible.

Lastly, thank you to **Sagar** and our TAs : **Robin** and **Ruchi**, for their constant support and guidance throughout the semester.

9 References

The following websites were referred for making the report and the design :

- https://nanosaur.ai/
- https://www.waveshare.com/wiki/IMX219-83_Stereo_Camera
- https://quartzcomponents.com/
- https://robocraze.com/blogs/post/mq-series-gas-sensor
- https://docs.nvidia.com/isaac/packages/visual_slam/doc/elbrus_visual_slam.html
- https://robocraze.com/products/16-bit-i2c-4-channel-ads1115-module
- https://www.mathworks.com/help/vision/ug/visual-simultaneous-localization-and-mapping-slam-overview.html