Robotic Exploration

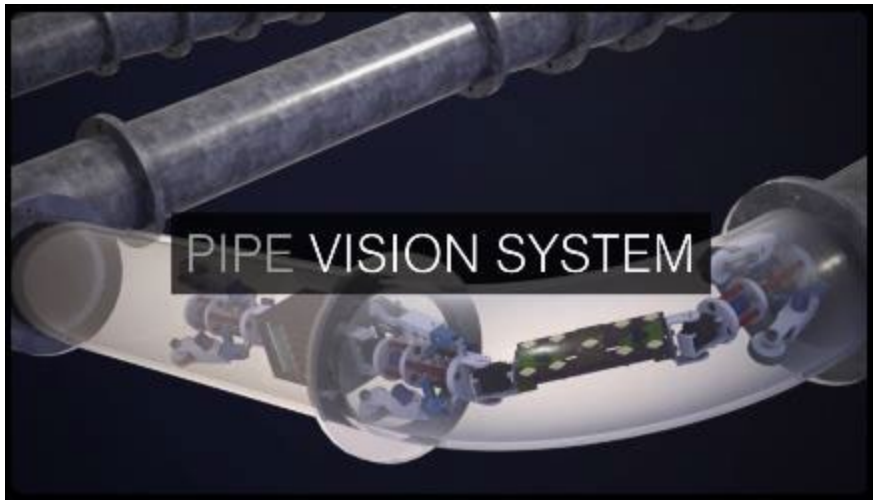
With Camera, Distance Sensor & Data Analysis

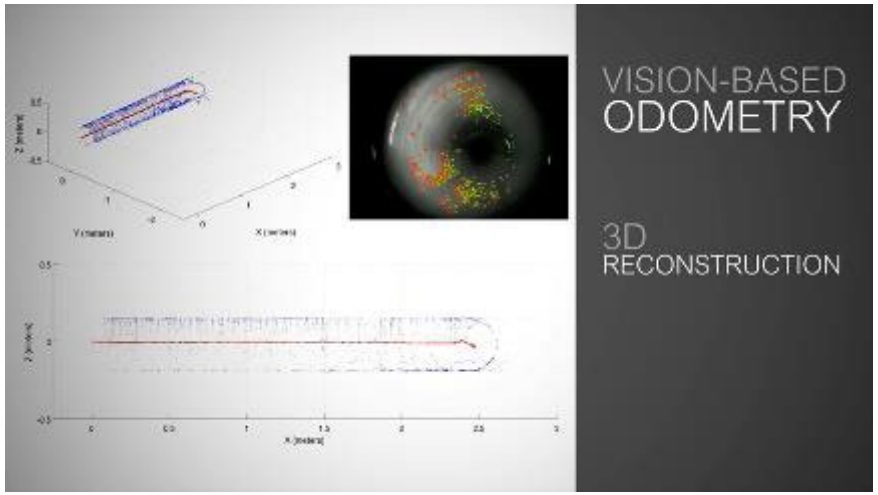
Jetson Nano with Jetbot

Carnegie Mellon Robotics Academy

Robotic Exploration

- Sometimes robots are sent into environments that are impossible or unsafe for humans.
- For example, this Pipe Exploration robot can travel through gas pipelines and collect data on the integrity of the pipe infrastructure: https://vimeo.com/215521157
- This data is then used to identify necessary repairs while avoiding excessive costs of digging up sections in need of no repair.
- In this exercise, you will use a robot to collect visual data from an environment with limited human visibility.





Carnegie Mellon Robotics Academy

Exercise Overview

- In the Autonomous Exploration exercise, you will configure your robot and its payloads (camera, light, distance sensor) to collect camera and distance sensor data from two environments with limited-or-no human visibility.
  - **Environment 1: Small area, adequate lighting**
  - **Environment 2: Large area, poor lighting**
- The robot will move forward into the environment and then collect snapshots from its camera and distance sensor as it spins in place.
- You will analyze the camera and distance sensor data through several processes and visualizations to ultimately generate a map of object locations.

Carnegie Mellon Robotics Academy

Setup

Robotic Exploration

Carnegie Mellon Robotics Academy

Materials Required

Flexible USB Lamp

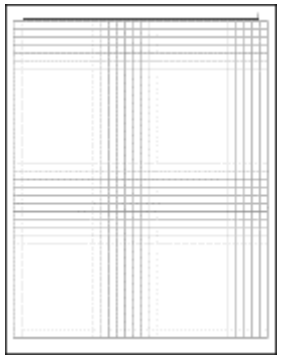
Flexible Qwiic Cable - Female Jumper (4-pin) and Breadboard Jumper Cables

Adafruit VL53L1X Time of Flight Distance Sensor

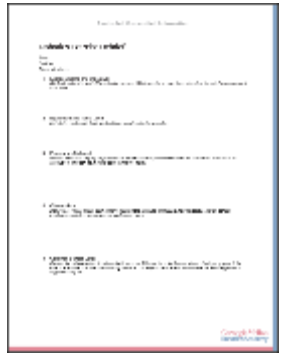
Jetson Nano Jetbot

Writing Utensils

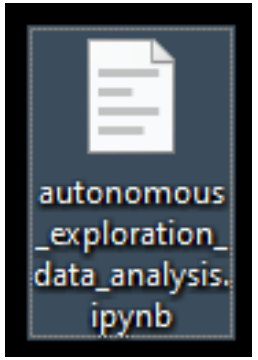
Assorted Objects



Graph Paper



Robot Exercise Debrief

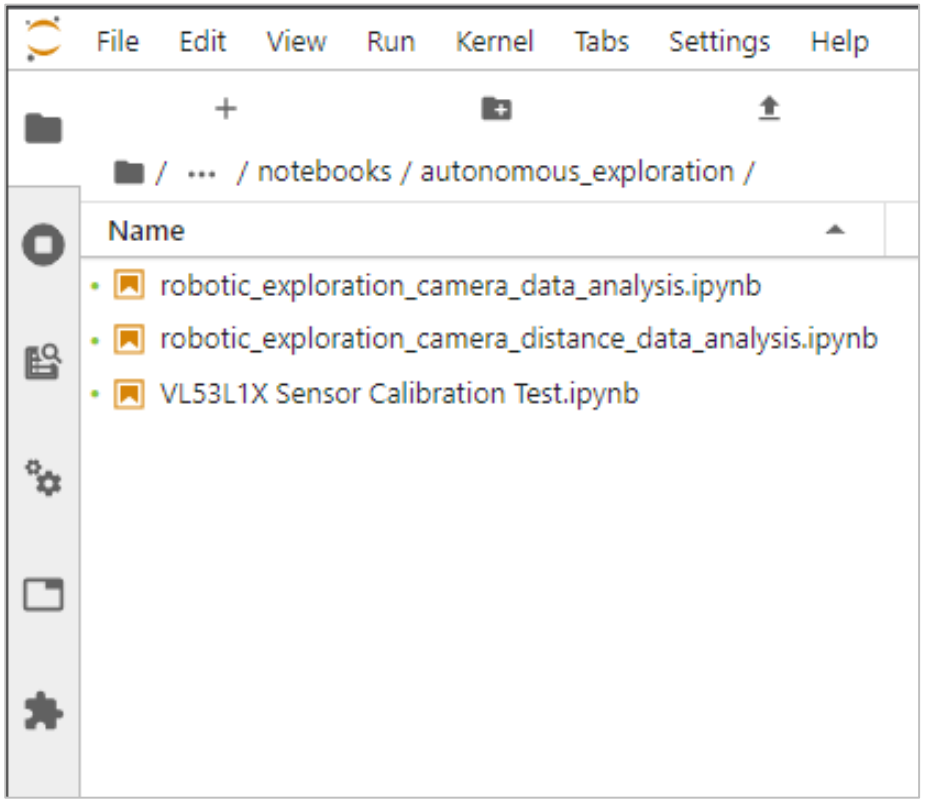


Jupyter Notebook

Carnegie Mellon Robotics Academy

Software Setup

- Connect to your Jetson Nano through Jupyter Notebooks.
- If the ***VL53L1X Sensor Calibration Test*** and ***Robotic Exploration Investigation (with Camera, Distance Sensor and Enhanced Analysis)*** Jupyter Notebook is not already installed on your Jetson Nano, create an “autonomous_exploration” folder in your “notebooks” directory and upload them there.



Carnegie Mellon Robotics Academy

Software Setup

- The ***Robotic Exploration Investigation (with Camera, Distance Sensor and Enhanced Analysis)*** Jupyter Notebook requires several Python Packages to be installed.
- Ensure that you are using a Jetson image that contains these, or separately install them using ***pip3*** commands.
- Libraries:
  - Matplotlib
  - VL53L1X
  - Smbus2
  - Jetson_inference
  - Jetson_utils

```
import time
import threading
import uuid
import subprocess
from smbus2 import SMBus
from VL53L1X import VL53L1X
import cv2
import ipywidgets as widgets
import traitlets
from IPython.display import display
from jetbot import Camera, Robot, bgr8_to_jpeg
import os
import shutil
import zipfile
import csv
import jetson_inference
import jetson_utils
import numpy as np
import matplotlib.pyplot as plt
```

Carnegie Mellon Robotics Academy

Payloads in Robotic Applications

Navigational Elements vs. Payloads:

- **Navigational Elements:** Components that help the robot sense its environment or navigate through it. They directly impact the robot's movement and decision-making.
  - Examples: Motors, wheels.
- **Payloads:** Components that add specific capabilities or functionalities to a robot but don't directly influence its navigation.
  - **Camera:** Captures snapshots of the robot's environment. Used for data collection and post-mission analysis, not for real-time navigation.
  - **USB Light:** Enhances the camera's visibility in dim environments. Ensures clarity of images but doesn't guide the robot's path.
  - **Distance Sensor:** Time of Flight sensor collects distance readings between the robot and a surface using a small laser

Carnegie Mellon Robotics Academy

IMX219-160 Camera Specifications

- Compact and efficient camera module that enhances the visual capabilities of the Jetson Nano.
- Camera Specifications:
  - Sensor: Sony IMX219.
  - Resolution: 8 Megapixels (3280 × 2464).
  - Field of View (FOV): 160◦.
  - Lens: CMOS size of 1/4 inch with an aperture of F2.35 and a focal length of 3.15mm.
  - Distortion: <14.3%1.

Carnegie Mellon Robotics Academy

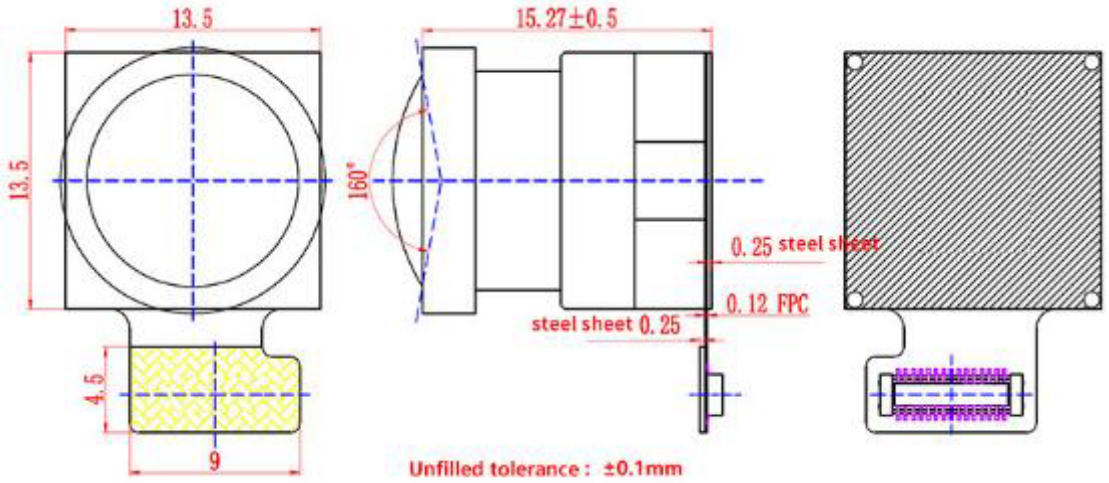
IMX219-160 Operating Conditions & Limitations

Optimal Conditions:

- Temperature: 0◦C to 50◦C (Stable Image)
- Adequate Lighting for High-Quality Images

Limitations:

- Temperature Range: -20◦C to 70◦C
- Low/Bright Lighting Affects Image Quality
- Distortion: <14.3%
- IR Sensitivity: Visible Light Only



Carnegie Mellon Robotics Academy

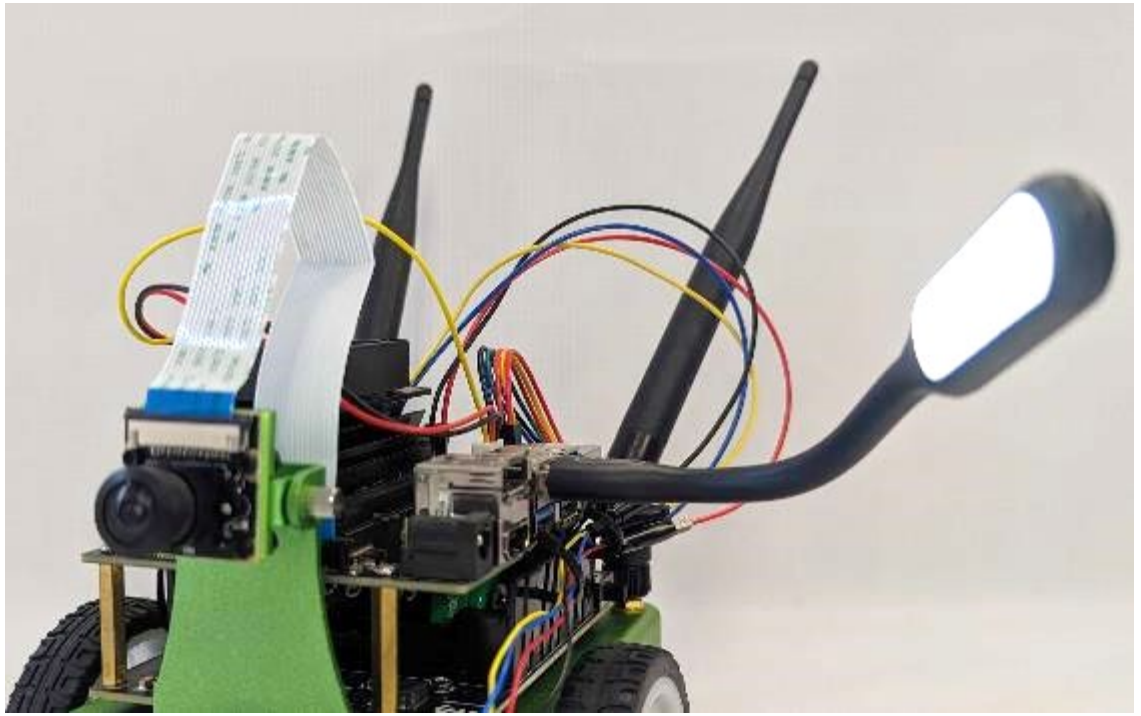
Flexible USB LED Lamp

- Flexible body, the lamp may be twisted 360 degrees to adjust light direction.
- Lamp Specifications:
  - Light Source Type: LED
  - Connectivity: USB
  - Wattage: 1 watts
  - Voltage: 5 Volts (DC)
  - Brightness: 80 Lumen
  - Color Temperature: 5500 Kelvin

Carnegie Mellon Robotics Academy

Hardware Setup

- Ensure that the Camera on the Jetson Nano is facing outward, such that it can capture images of the robot's surroundings.
- Connect the USB Light to the Jetson Nano and orient it to illuminate the camera's field of view.



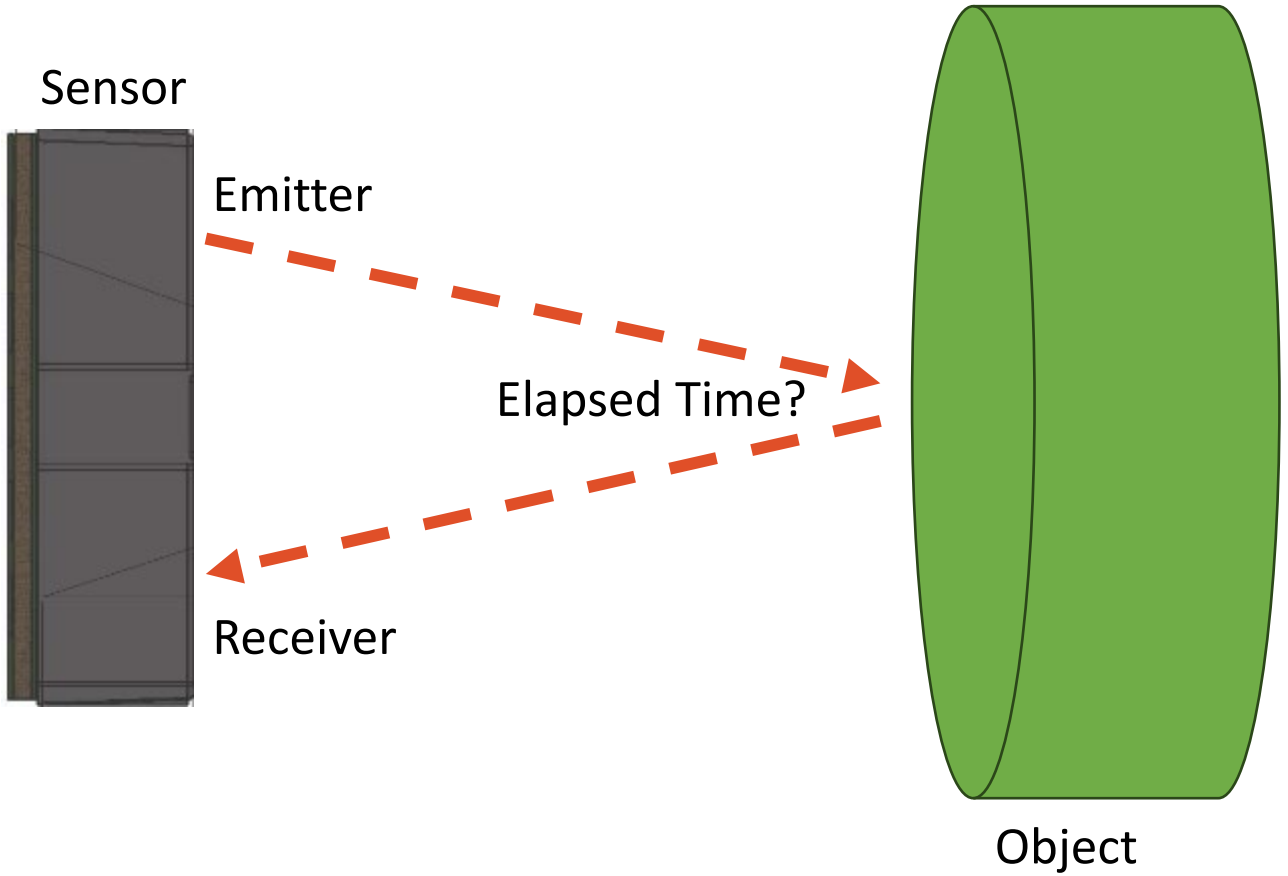
Carnegie Mellon Robotics Academy

VL53L1X Time-of-Flight Distance Sensor

- A state-of-the-art distance sensor utilizing Time-of-Flight (ToF) technology.
  - Works by measuring the time it takes for emitted light to reflect off an object and return.
- Measures distances from 40mm to 4m with high accuracy.
- 27-degree Field-of-View

Carnegie Mellon Robotics Academy

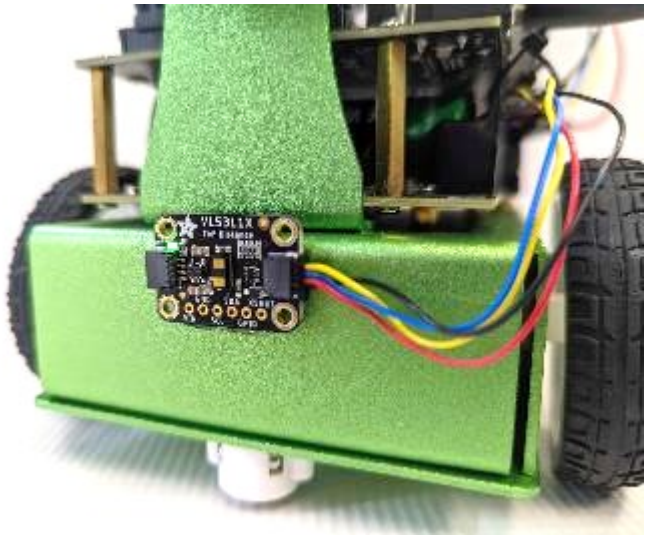
VL53L1X Time-of-Flight Distance Sensor

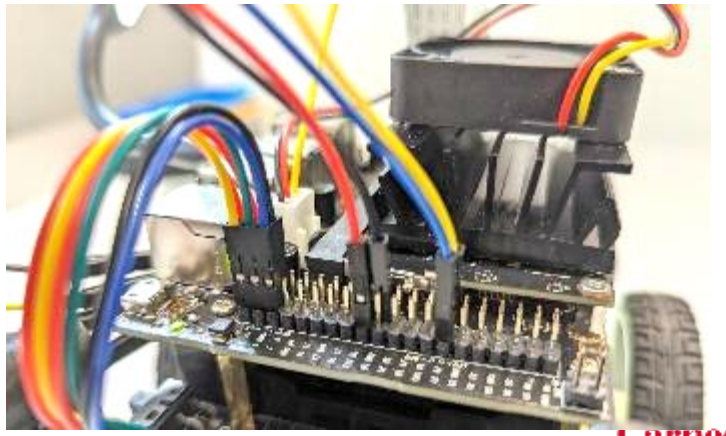


- The sensor emits a pulse of light.
- Measures the time taken for the pulse to bounce off an object and return.
- Uses this time measurement to calculate distance.

Connecting VL53L1X to Jetson Nano

- Secure the VL53L1X Distance Sensor to the front of the robot, with the sensor name facing up.
- Use the Qwiic Cable to connect the sensor to the Jetson GPIO. Breadboard Jumper cables may be used to extend the length.
  - **Black Wire (GND):** Connects to any GND pin on the Jetson Nano.
  - **Red Wire (3.3V):** Power the sensor by connecting to a 3.3V pin on the Nano.
  - **Blue Wire (SDA):** Data line, connects to Pin 27 on the Nano.
  - **Yellow Wire (SCL):** Clock line, connects to Pin 28 on the Nano.

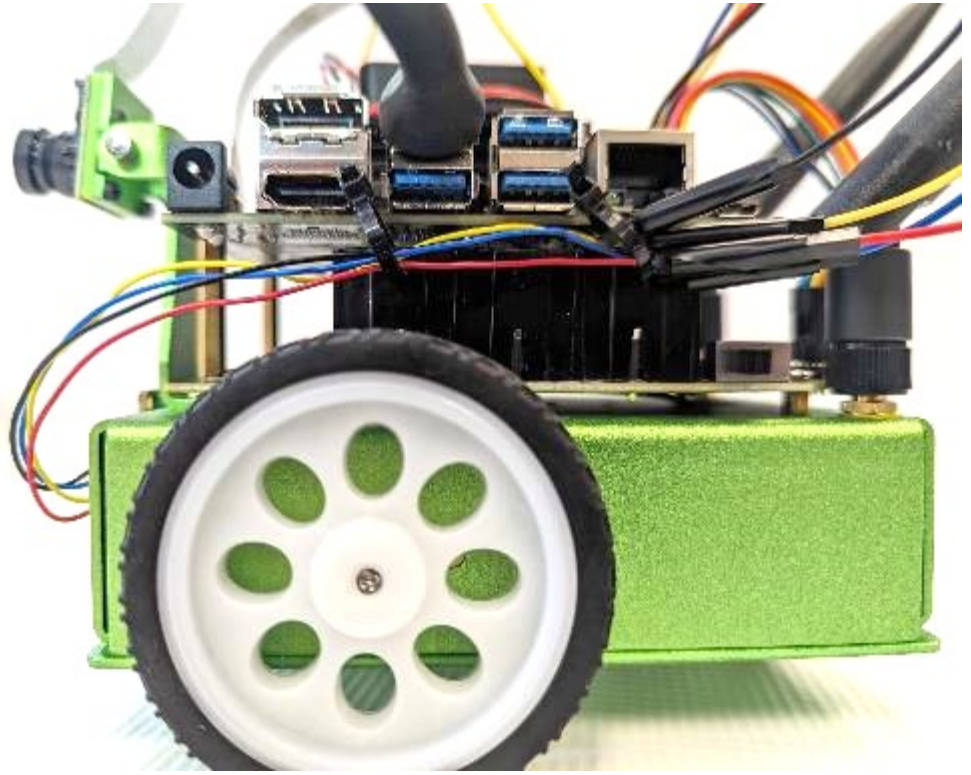




Carnegie Mellon Robotics Academy

Connecting VL53L1X to Jetson Nano

- If you used Breadboard Jumper Cables to extend the length the length of the cables, electrical tape may be used to ensure the connection holds as the robot moves.
- Ensure that the cables connecting the sensor to the Jetson GPIO are not at risk of entanglement with the wheels or other components.



Carnegie Mellon Robotics Academy

Sensor Range and Accuracy Factors

Optimal Conditions:

- Indoors with controlled lighting.
- Away from reflective surfaces.
- Stable power supply.

Limitations:

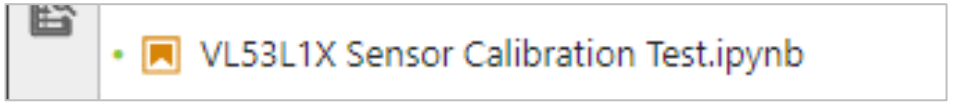
- Accuracy decreases with increased ambient light.
- Reflective surfaces can cause false readings.
- Limited to a maximum distance of ~4 meters.

Distance mode	Max. distance in dark (cm)	Max. distance under strong ambient light (cm)
Short	136	135
Medium	290	76
Long	360	73

Carnegie Mellon Robotics Academy

VL53L1X Sensor Calibration

- Open the ***VL53L1X Sensor Calibration Test*** Jupyter Notebook.
- The code included initializes the VL53L1X Distance Sensor and outputs 10 sensor readings (1 per second for 10 seconds). Run the code to ensure that the sensor is working and providing accurate distance readings.
- The value in the “start_ranging()” command controls whether the distance sensor is calibrated for short, medium, or long-distance ranges and may be modified based on your environment and testing.




```
Starting the program...
Initializing the VL53L1X sensor...
Starting to read sensor values...
Reading 1: Distance = 250mm
Reading 2: Distance = 249mm
Reading 3: Distance = 254mm
Reading 4: Distance = 9mm
Reading 5: Distance = 117mm
Reading 6: Distance = 136mm
Reading 7: Distance = 109mm
Reading 8: Distance = 133mm
Reading 9: Distance = 136mm
Reading 10: Distance = 123mm
Program finished.
```

```
# 0 = Unchanged
# 1 = Short Range
# 2 = Medium Range
# 3 = Long Range
tof.start_ranging(3)
```

Carnegie Mellon Robotics Academy

Procedures

Robotic Exploration

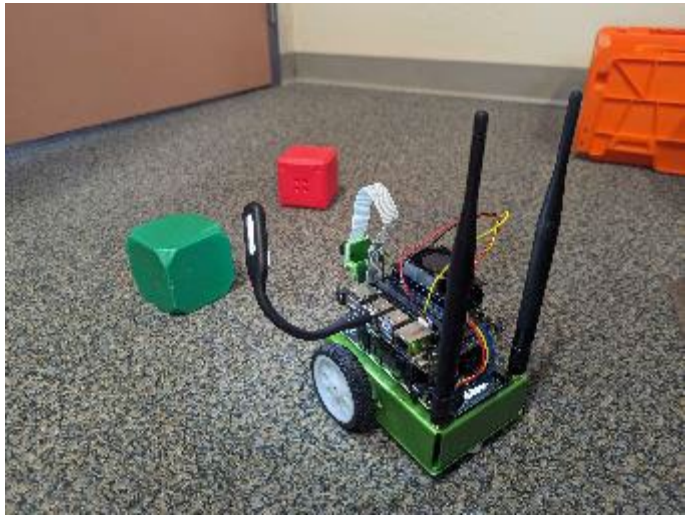
Carnegie Mellon Robotics Academy

Investigation: Environment 1

- In this Investigation, you will collect camera and distance sensor data from two separate environments with limited-or-no human visibility.
  - **Environment 1: Small area, adequate lighting**
  - **Environment 2: Large area, poor lighting**
- This process will help you to better understand how the different sensors perform under these conditions, allowing you to generate plans that leverage the strengths of the sensors and system.
- Begin by placing your robot in Environment 1.

Carnegie Mellon Robotics Academy

Robot and Camera Calibration



- Begin running ***Robotic Exploration Investigation (with Camera, Distance Sensor and Enhanced Analysis)*** Jupyter Notebook
- Place the robot in a known location with objects in view.
- Run the steps through the "*3. Raw Data Visualization*" section of the notebook to set up the distance sensor, robot and camera.
  - Verify that the view in the camera widget is working, properly angled and illuminated to capture the robot's surroundings.
  - Adjust the camera and USB light as necessary.

Carnegie Mellon Robotics Academy

Robot Turn Calibration

- Run the "*4. Robot Movement Calibration*" section of the notebook to find the appropriate values for robot_speed and time_to_360.
- Adjust the robot_speed and time_to_360 variables until the robot very slowly creates a full 360 degree turn.

```
# Calibrate robot speed and duration for 360 degree turn
robot_speed = 0.125
time_to_360 = 5.5

robot.left(robot_speed)
time.sleep(time_to_360)
robot.stop()
```

Carnegie Mellon Robotics Academy

Adjust Time Forward Parameter

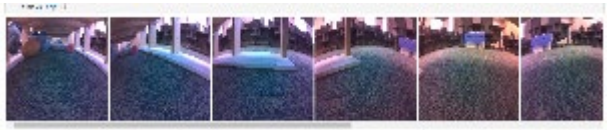
The code block under the "*5. Raw Data Collection*" section of the notebook contains a variable that allows you to rapidly adjust how far the robot should travel before collecting data.

- Adjust this parameter so that the robot travels the correct duration forward.

```
# Variable to control
time_forward = 0
```

Run Exploration Routine

- Place the robot in the environment with limited visibility.
- Run the code block in the Jupyter Notebook under "*5. Raw Data Collection*".
  - Snapshot images from the camera at each interval will appear in a "snapshots" folder.
- Run the code block in the Jupyter Notebook under "*6. Raw Data Display*" to inspect the frames and corresponding distances captured
- Step 7 stops the camera to free up resources and Step 8 generates a video of the frames.



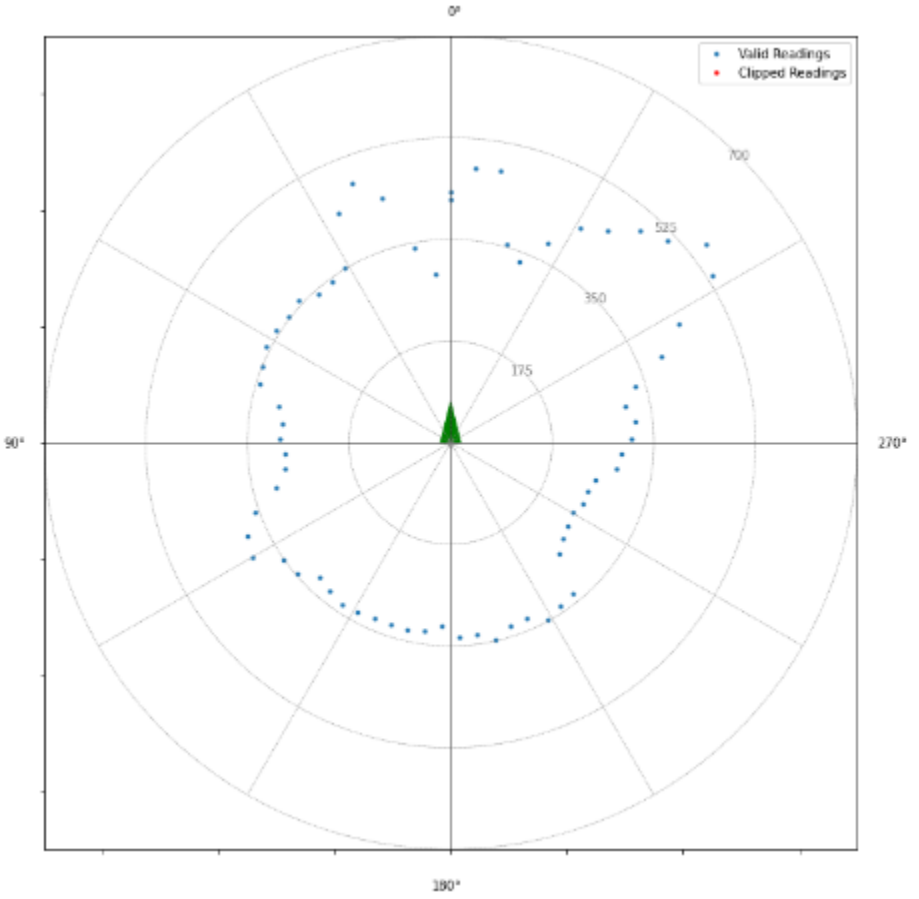
Carnegie Mellon Robotics Academy

Analysis: Generate 2D Polar Grid

- Steps 9 and 10 generate a 2D Polar Grid of the Distance Sensor readings. The green triangle in the center represents the starting orientation of the robot, and the dots around it represent the distance sensor readings taken as the robot spins.
- The “cutoff_threshold” variable in Step 10 may be adjusted to set an appropriate range for the grid and “clip” far-out values (represented as red dots).

```
cutoff_threshold = 800

# Plotting the distance values
display_distance_values_polar_view(distance_values, cutoff_threshold)
```



Carnegie Mellon Robotics Academy

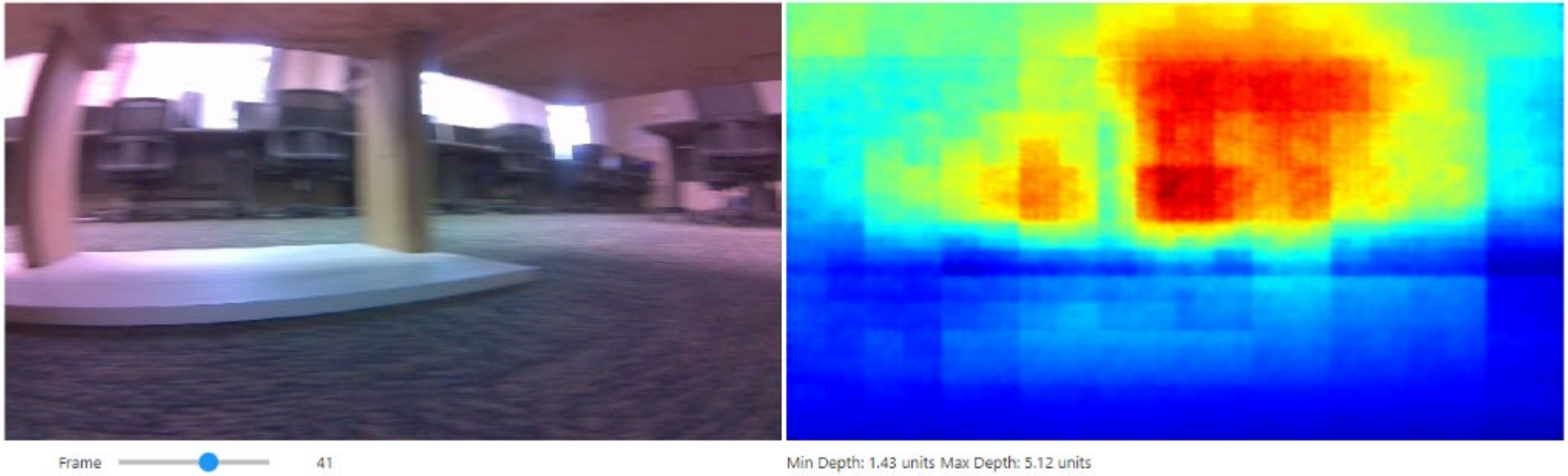
Analysis: Generate Panoramic Image

- The "*11. Image Analysis Panorama*" step of the notebook generates a panoramic image from the snapshot images collected. The number of frames used is configurable and will take some time to complete. After processing, the image should appear in the notebook.

Carnegie Mellon Robotics Academy

Analysis: Generate Depth Maps

- Steps 12 – 14 load the DepthNet Model and run an algorithm to process each frame for estimated minimum and maximum distances, represented as a Depth Map, similar to a Heat Map.



Carnegie Mellon Robotics Academy

Analysis: Perform Object Recognition

- Steps 15 – 17 load the MobileNet Model and run an algorithm to process each frame for identifiable objects. The “threshold” value in Step 15 may be adjusted to increase or decrease the necessary confidence score for recognition.

Carnegie Mellon Robotics Academy

Analysis: Generate Data Log

12. Data Log text

autonomous_exploration_da | data_log.csv

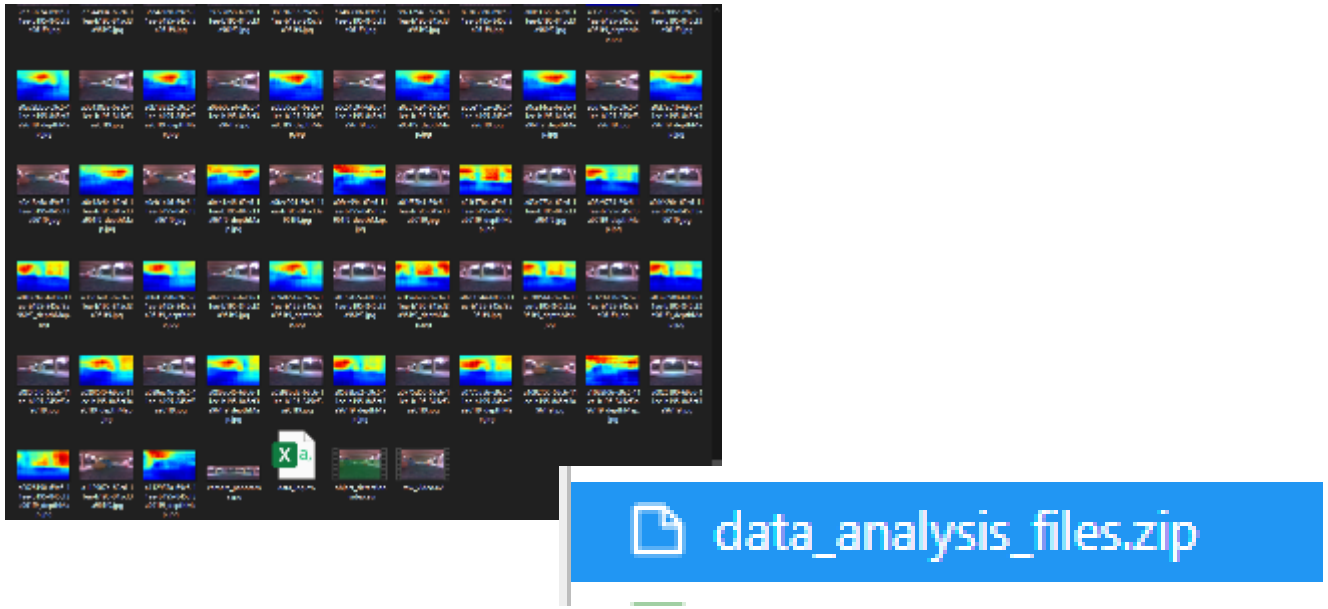
Delimiter: ,

	File Name	Distance	Depth Map Name	Min Depth Value	Max Depth Value	d Objects and Confidence
1	ee-b195-845cf3a961f9.jpg	431	5cf3a961f9_depthMap.jpg	1.0234375	4.4492188	bed (0.37)
2	ee-b195-845cf3a961f9.jpg	292	5cf3a961f9_depthMap.jpg	1.0927734	5.4453125	
3	ee-b195-845cf3a961f9.jpg	341	5cf3a961f9_depthMap.jpg	1.0771484	2.3085938	person (0.41)
4	ee-b195-845cf3a961f9.jpg	437	5cf3a961f9_depthMap.jpg	1.1425781	2.5507812	
5	ee-b195-845cf3a961f9.jpg	478	5cf3a961f9_depthMap.jpg	1.1464844	4.0195312	
6	ee-b195-845cf3a961f9.jpg	440	5cf3a961f9_depthMap.jpg	0.9980469	4.2929688	
7	ee-b195-845cf3a961f9.jpg	352	5cf3a961f9_depthMap.jpg	1.1425781	4.0859375	
8	ee-b195-845cf3a961f9.jpg	344	5cf3a961f9_depthMap.jpg	0.97314453	4.734375	sports ball (0.36)
9	ee-b195-845cf3a961f9.jpg	342	5cf3a961f9_depthMap.jpg	0.9692383	4.96875	sports ball (0.37)
10	ee-b195-845cf3a961f9.jpg	357	5cf3a961f9_depthMap.jpg	0.8828125	4.890625	

Carnegie Mellon Robotics Academy

Analysis: Generate Data Archive

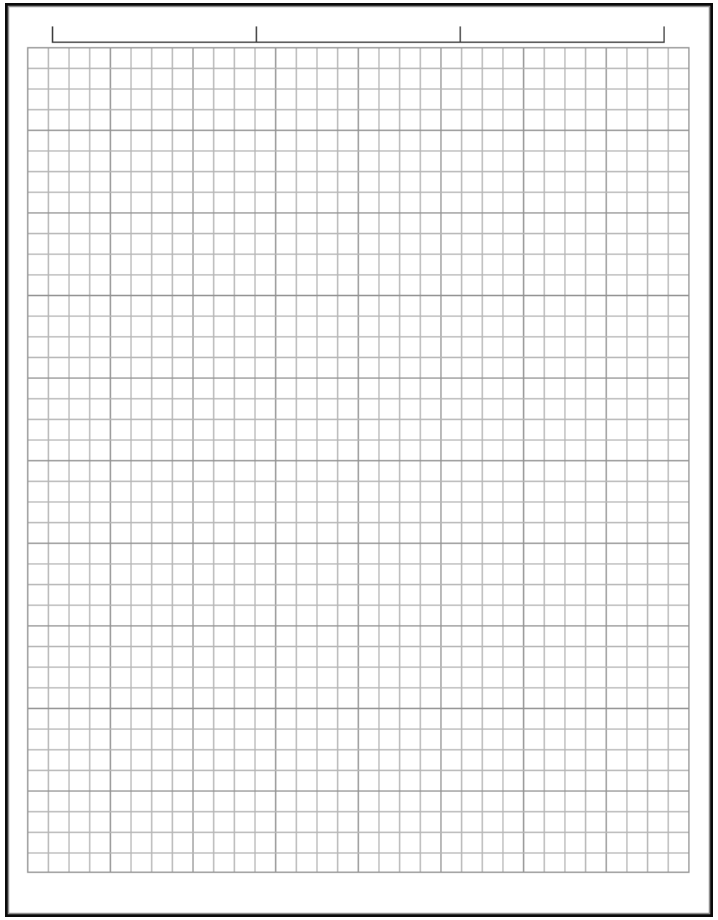
- Step 18 generates a CSV log of all data gathered and processed during the investigation, and Step 19 compresses the log and other data into a single ZIP file for convenient download.



Carnegie Mellon Robotics Academy

Manual Analysis: Generate Map

- Choose a scale that allows your environment to fit on the graph paper.
- Use writing instruments to add landmarks to the graph paper that provide reference points.
- Represent the path that the robot took with an arrow.
- Use the data you captured and analyzed to map and label objects from the environment.



Carnegie Mellon Robotics Academy

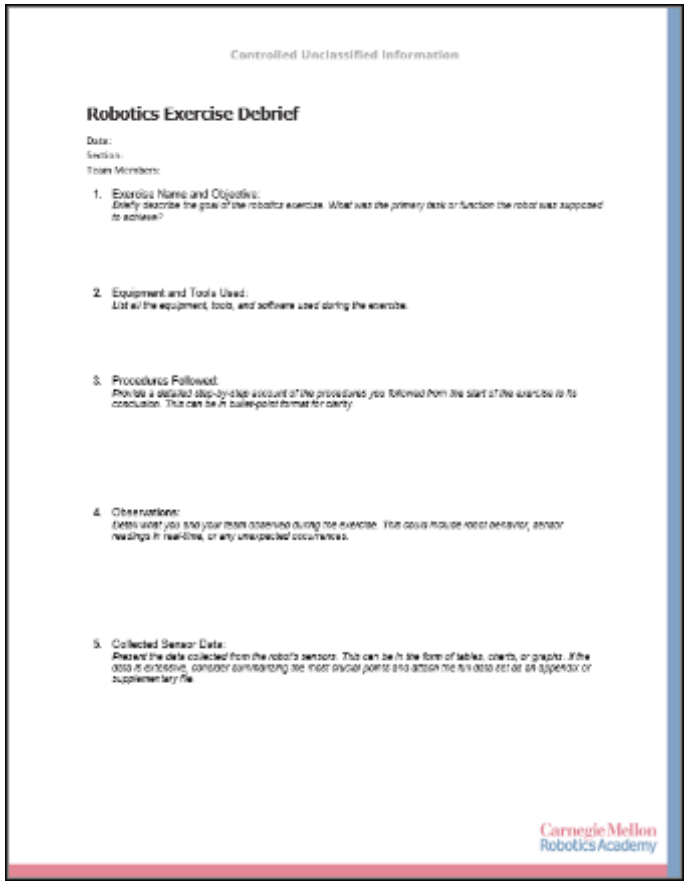
Investigation: Environment 2

- After processing all data from Environment 1, it is time to move on to Environment 2:
  - **Environment 1: Small area, adequate lighting**
  - **Environment 2: Large area, poor lighting**
- Move your robot to Environment 2.
- Complete all the procedures to capture and analyze data in Environment 2.

Carnegie Mellon Robotics Academy

Store and Disseminate Data

- Complete the Robotics Exercise Debrief. Make sure to include all data and the map you generated. Use appendices as necessary.
- Upload the Robotics Exercise Debrief, along with the data to the classroom server.
- E-mail the instructors to share this data.


Controlled Unclassified Information

Robotics Exercise Debrief

Date:

Section:

Team Members:

1. Exercise Name and Objective:
   Briefly describe the goal of the robotics exercise. What was the primary task or function the robot was supposed to achieve?

2. Equipment and Tools Used:
   List all the equipment, tools, and software used during the exercise.

3. Procedures Followed:
   Provide a detailed step-by-step account of the procedures you followed from the start of the exercise to its conclusion. This can be in bullet-point format for clarity.

4. Observations:
   Detail what you and your team observed during the exercise. This could include robot behavior, sensor readings in real-time, or any unexpected occurrences.

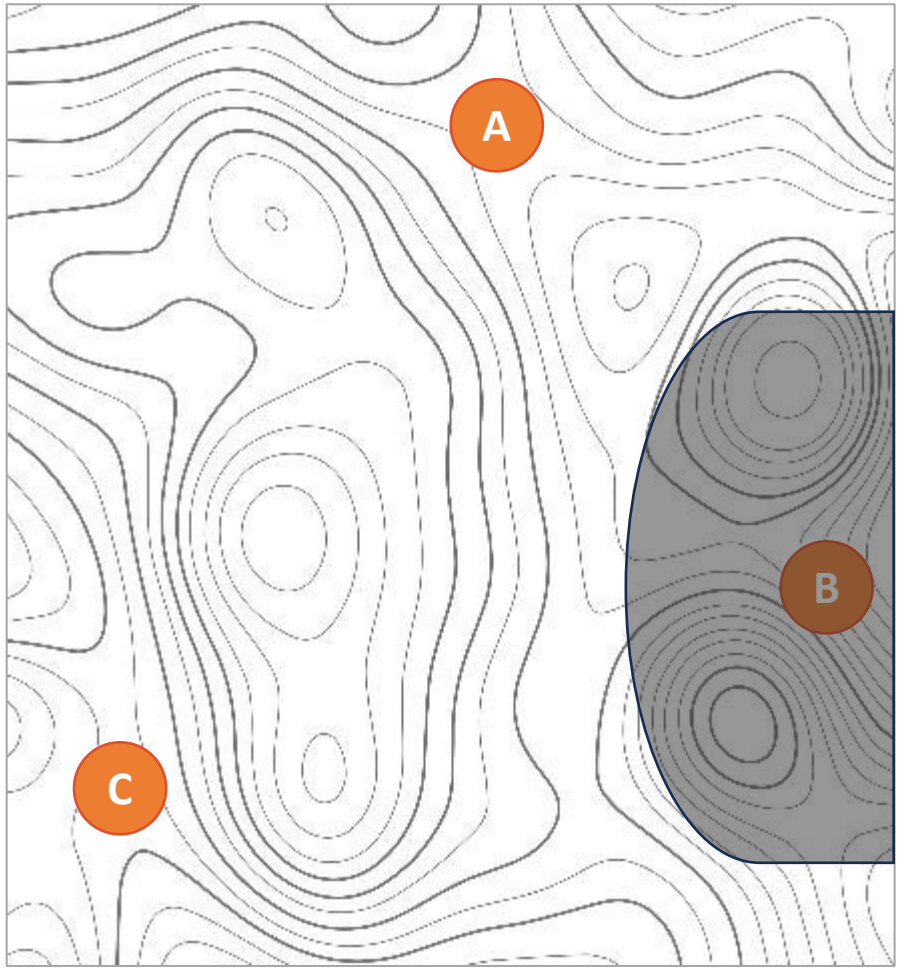
5. Collected Sensor Data:
   Present the data collected from the robot's sensors. This can be in the form of tables, charts, or graphs. If the data is extensive, consider summarizing the most crucial points and attach the full data set as an appendix or supplementary file.

Carnegie Mellon Robotics Academy

Carnegie Mellon Robotics Academy

Sensor Mapping Challenge

- In this challenge, participants must generate a map by collecting and analyzing sensor data from 3 connected areas.
  - Some areas of the map have lighting limitations.
  - Participants **must generate a plan** for which sensors and analysis they will use at each area, as well as the necessary procedures to carry out their plan.
  - Joystick Control with Camera may be used to navigate the robot from one area to the next, but direct visibility is not permitted.
- Turn in the resulting map and all data collected during the challenge.



Carnegie Mellon Robotics Academy