International Journal
Information Technology Applications (ITA)
Volume 14, Number 2, December 2025

PRELIMINARY EVALUATION OF AN VL53L8CX 8X8 TIME-OF-FLIGHT ZONE DISTANCE SENSOR MODULE FOR A LINE-FOLLOWING ROBOT APPLICATION

Samuel Ladislav Šindler

Abstract:

This paper describes preliminarily evaluation of an 8x8 Time-of-Flight obstacle detection module for a line-following mobile robot. The module is based on the VL53L8CX multi-zone distance sensor mounted on a Teensy 4.1 robot platform. The method uses a low-resolution depth matrix, simple validity filtering, and rule-based interpretation of obstacle patterns. Static measurements were taken for three obstacle types relevant to line-following competitions: a brick, a curtain, and a tunnel. The results show that the sensor can detect obstacles in the practical 10-30 cm range and that a distance of approximately 25 cm is especially useful for distinguishing obstacle patterns. A brick appears as a compact surface with background visible around it, a curtain appears as a wide non-homogeneous surface, and a tunnel shows a hollow center. The conclusion is that an 8x8 ToF matrix is not a perfect universal classifier, but it gives enough spatial information to replace simpler single-distance detection in many practical line-following scenarios.

Keywords:

Line-following robot, obstacle detection, time-of-flight sensor, VL53L8CX, depth map, embedded robotics.

Introduction

Line-following robots are often understood as simple autonomous mobile robots that only need to track a black line on a bright surface. In practice, competition tracks are not always that simple. The robot can meet a tunnel, a brick, a curtain, interrupted line segments, lighting changes, color changes, or a track edge. Some of these situations can be solved by the line sensor itself, but some of them require additional front perception. The problem is not only to see that something is in front of the robot. The more useful question is what kind of obstacle it is and what the robot should do next. Older solutions often use sensors which only give one distance value. It can tell that there is something in front of the robot and the distance of that something, but it cannot describe the shape of the object. This becomes a serious limitation when the same robot should handle several obstacle types differently.

The motivation of this work is therefore practical. I am building a multi-purpose line-follower robot that should run in more than one competition scenario without physically changing the hardware. The front sensor should help the robot understand the obstacle before it is too close. The selected sensor, VL53L8CX, provides an 8x8 matrix of distance readings.

This matrix is enough to create a low-resolution heat map of the space in front of the robot. Combining this information with a simple rule-based logic allows the robot to autonomously deal with the obstacle it is facing. First, the robot has to determine that an obstacle is present.

Region around 15 cm before the obstacle, is the last useful region where the robot can still engage obstacle avoidance maneuver without immediately engaging brakes or reversing motors. For classification purpose and according to the data around 25 cm is a more suitable region for decision making, because the brick, curtain, and tunnel produce different and distinguishable patterns in the 8x8 matrix. This paper describes the implementation, measured patterns, and the limits of this approach.

1 Practical Background and Competition Requirements

Line-follower competitions usually define a bright playing field with a dark guiding line, but the exact obstacle set depends on the competition and the year. The ISTROBOT line follower rules, for example, state that different obstacles may appear on the track in different order. The listed obstacle set includes a tunnel, interrupted line, spotlight, split line, bridge, brick, color change, curtain, oil patch, and return loop [2]. The rules also state that the track has no barriers, so sensor interpretation near the edge of the table becomes a safety issue [2].

The Robotic day in Prague line follower category is also based on following a black line and overcoming obstacles [3]. These rule sets are not identical, but they show the same practical problem: an environment containing several obstacle types. A robot that only knows that an object is nearby remains limited, because it still depends on a predefined obstacle order. True autonomous behavior requires not only detecting the presence of an obstacle, but also interpreting its type and position, since different placements may require different reactions.

The hardest practical situation is not a single obstacle placed in the middle of a straight line. The difficult situation is an obstacle after a tight turn, inside a tight turn, or close to another feature. In that case the robot may have less time to react reliably. For example, a single ultrasonic sensor often forces the robot to slow down prematurely, because the returned sound wave and processing delay reduce usable reaction time. A multi-zone ToF sensor can read a spatial pattern using light-based distance measurement and can provide more information about the obstacle at a higher rate.

For this article the practical obstacle set was reduced to: brick, curtain, and tunnel. This is enough to show whether an 8x8 matrix adds meaningful value. The expected robot behavior is different for each class. A brick is treated as a solid obstacle and should be bypassed. A curtain is treated as a passable obstacle, the sensor detection can be disabled for a short time after entry. A tunnel follows similar principle, but it should be recognized as separate object because the required timeout of sensor detection is longer so that the robot does not start an avoidance maneuver inside the tunnel.

2 Hardware and Sensor Integration

The tested robot platform is a custom line-follower based on Teensy 4.1. The robot uses a QTRX-MD-13A analog reflectance sensor array for line detection, a TB67H420FTG motor driver, magnetic wheel encoders, an OLED display, SD logging, and a front VL53L8CX ToF sensor. The line-following part is intentionally custom and does not depend on the Pololu QTR library, because the library doesn’t support short gaps, loops and colored line sections. The obstacle-detection module which uses VL53L8CX described here is one subsystem of the same robot.

The VL53L8CX sensor was mounted at a vertical height of 60 mm from the ground. This was selected as a practical compromise. It is high enough not to be mechanically awkward and low enough to observe the playing field. In the current interpretation, the bottom two rows of the 8x8 matrix are allowed to see the floor, while the upper six rows are used mainly for obstacle detection.

This is useful because many line-follower tracks are placed on tables roughly 1.0-1.4 m above the ground, without safety nets. Seeing the ground in the bottom rows can be used for an emergency stop if the robot approaches the edge of the playing field.

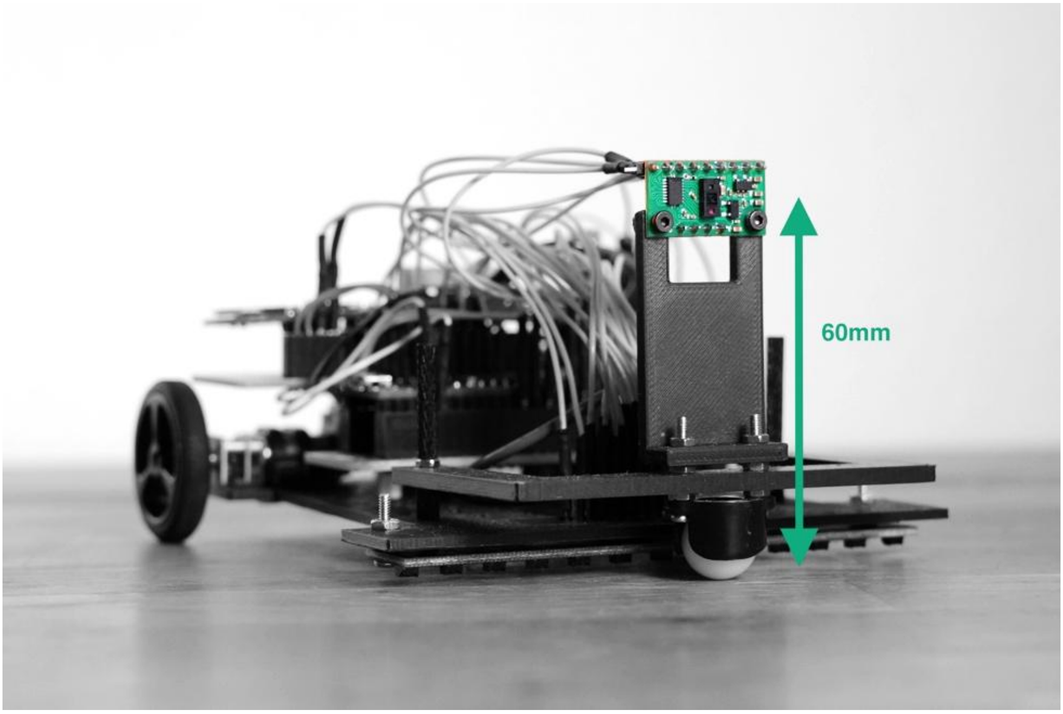

Fig.1. Robot platform with the front VL53L8CX sensor mounted above the line sensor.

The VL53L8CX is a multi-zone Time-of-Flight distance sensor. According to Pololu, the carrier board provides an ST VL53L8CX sensor capable of fast and accurate ranging up to 4 m through I2C or SPI. It can measure multiple targets across multiple zones and produce enough data for a depth map with up to 8x8 resolution. Pololu describes this sensor as a basic 3D lidar in the practical sense. The sensor can be configured in 4x4 or 8x8 mode. For this work the 8x8 mode was used because the main goal is not only distance detection but also shape interpretation. The carrier board includes voltage regulators and level shifters, which simplifies the connection to a 3.3 V. The supported speed of the SPI communication according to Pololu is up to 3 MHz [1].

A practical implementation issue appeared during testing. The I2C scanner detected the sensor address, but sensor initialization over I2C failed. The LP pin jumper shorting was tested and did not fix the issue. Address-related attempts also did not produce a stable result. Because of that, the final implementation used SPI. With SPI, the sensor initialized and produced an 8x8 matrix.
The code rotated the printed matrix, so that the terminal view matched the physical left and right side of the robot. Invalid cells without a detected target were printed as -1.

The cause of the I2C initialization failure was not conclusively identified. Since only one sensor board was available, it was not possible to determine whether the issue was caused by the particular board, wiring sensitivity, signal integrity, or bus configuration. Further investigation with additional sensor units and oscilloscope measurements would be required.

3 Measurement and Processing Method

3.1 Matrix interpretation

The sensor output was interpreted as an 8x8 matrix, where each cell contains the measured distance in centimeters. A corresponding status matrix was also printed. Values with no detected target were stored as -1. In the current practical processing, invalid values are excluded from the decision instead of being treated as very short or very long distances. This is necessary because direct sunlight, heavy vibration, and very short distances can produce missing readings.

The measurement window was divided into two functional parts. The upper six rows are used for obstacle interpretation. The lower two rows are allowed to see the floor and can later be used for line-following field edge detection.

$$A(r,c) = 1 \text{ if } \mathrm{D(r,c)} \leq T \text{ and } D(r,c) \text{ is valid}, \text{otherwise } A(r,c) = 0 \tag{1}$$

In Equation (1), T is the distance threshold. A threshold around 15 cm is useful as a late safety threshold. However, for obstacle classification this work found that approximately 25 cm is more informative. At 25 cm the object is close enough to occupy meaningful matrix space, but still far enough that the 8x8 resolution can show differences between a solid brick, a wide curtain, and a tunnel entrance.

3.2 Rule-based classification

The classification is intentionally simple. The reason is that the robot is an embedded line-follower with limited time to react, and the rules of the target obstacles are known. The goal is to detect a practical pattern, not to recognize arbitrary objects in the world.

A brick is expected to appear as a compact solid object. At 25-30 cm the side columns and some top cells can still see the track or the background, so these values are higher than the center. A curtain is wider, so it fills most of the visible frame with roughly similar numbers. Because the test curtain is made from vertical paper strips, the surface is not perfectly uniform; the matrix shows small vertical differences between strips. This was more visible than expected at only 8x8 resolution. A tunnel produces the most interesting pattern: the entrance frame is close, but the center appears farther away because the sensor sees into the hollow inside.

$$\text{obstacle class} = \text{f (width of active region, center distance,} \\ \text{side distance, vertical non} - \text{uniformity)} \tag{2}$$

Equation (2) describes the classification logic at a high level. Each feature is needed for a different reason. The width of the active region is useful for distinguishing a narrow brick from a wider curtain. The center distance is important for tunnel detection, because the entrance frame is close, but the middle of the tunnel appears farther away. The side distance is useful for recognizing whether the object fills the whole frame or only the central part of it. The vertical non-uniformity is useful for curtain detection, because the paper strips do not form a perfectly flat surface and therefore create small but visible distance differences in the matrix.

The result of the classification is not only the obstacle label, but also the driving reaction. For a brick, the robot should perform an avoidance maneuver and return to the line. For a curtain, the robot should continue forward and temporarily ignore the front sensor, because the curtain itself will remain close while the robot passes through it. For a tunnel, the robot should continue through the hollow region and avoid interpreting the tunnel walls as an obstacle that needs to be bypassed.

3.3 Test procedure

The measurements were taken in static conditions first. The robot or sensor was placed in front of the tested object and a full 8x8 distance matrix was recorded. The tested distances were 30 cm, 25 cm, 20 cm, 15 cm, 10 cm, and in some generic-distance tests also 5 cm. The main classification comparison used 10 - 30 cm, with special attention to 25 cm because it gave the clearest separation between the obstacle classes.

The three target objects were a tunnel, a curtain, and a brick. The tunnel corresponds to a passable object with an entrance at least 20x20 cm in the referenced competition rules. The makeshift curtain had a size of approximately 21x4x21 cm and was made from vertical paper strips. The makeshift brick had a size of approximately 7x4x15 cm. The exact dimensions are not the main point; the important point is that the proportions imitate the different obstacle types. Actual competition dimensions can be calibrated before a run.

4 Experimental Results

4.1 Reference frame and floor visibility

The reference scenario was measured with a wall approximately 92-93 cm from the robot. The upper rows reported the wall distance, while the lower rows reported much shorter values because they saw the floor of the track. This confirmed that the chosen 60 mm mounting height gives useful information about both the obstacle region and the playing field.

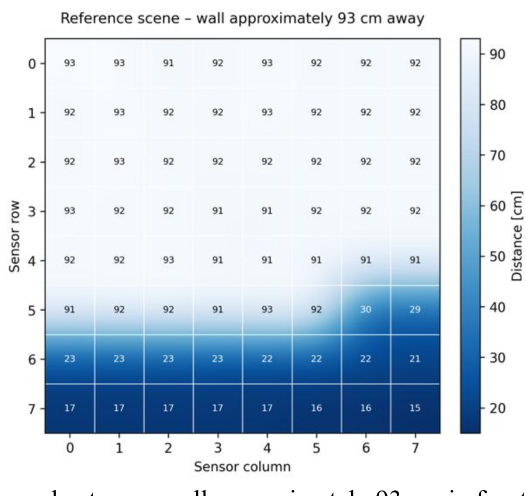

Fig.2. Reference heat map: wall approximately 93 cm in front of the robot.

4.2 Practical distance range

A dummy obstacle was measured at 30 cm, 25 cm, 20 cm, 15 cm, 10 cm, and 5 cm. In static conditions the sensor measured these distances accurately enough for the intended use. At 5 cm the readings already started to show small deviations, which is acceptable for distance detection but not ideal for detailed classification. The official minimum ranging distance is 2 cm, but in the moving robot use case such distance is too late for a useful maneuver [1].

For a line-follower, the useful range is not the same as the technical minimum range. If the robot detects a solid obstacle only at 5 cm, it may already need to brake hard, reverse, or accept wheel drift. A better strategy is to detect and classify the obstacle in a range between 10 cm and 30 cm. Higher distances such as 80-120 cm can still indicate that something is in front of the robot, but the 8x8 resolution is not sufficient to classify the object shape reliably at such a distance. This can still be used to slow the robot down before it reaches the obstacle zone.

Table 1. Practical meaning of tested obstacle distances.

Distance	Observation	Practical meaning
30 cm	Object is visible, but fine class differences are weaker.	Good for early warning or speed reduction.
25 cm	Brick, curtain, and tunnel produce clearly different patterns.	Best distance in this dataset for classification.
20 cm	Objects are still identifiable, tunnel hollow shape remains visible.	Good for confirmation before action.
15 cm	Late but still useful for detection and braking decision.	Minimum practical maneuver threshold.
10 cm	No classification possible and reaction time is getting too short.	Useful as last resort at slower speeds.
5 cm	Static values exist but moving tests produce many invalid readings.	Too late for obstacle avoidance without applying reverse motor command.

4.3 Sunlight behavior

Sunlight is a practical issue for optical and infrared sensors. Some ToF sensors produce false obstacle detections in bright venues. The VL53L8CX product documentation claims improvement in ambient light performance [1]. The robot was tested outside on a sunny day. Some of the top-row values in 8x8 matric occasionally became invalid, but the lower and central usable regions still provided enough usable data. When the sensor was pointed directly into the sun, many top and centre values became unusable. This result is not surprising, because direct sunlight is very harsh. The important practical conclusion is that invalid readings must be filtered and that the robot should not base a decision on a single captured frame.

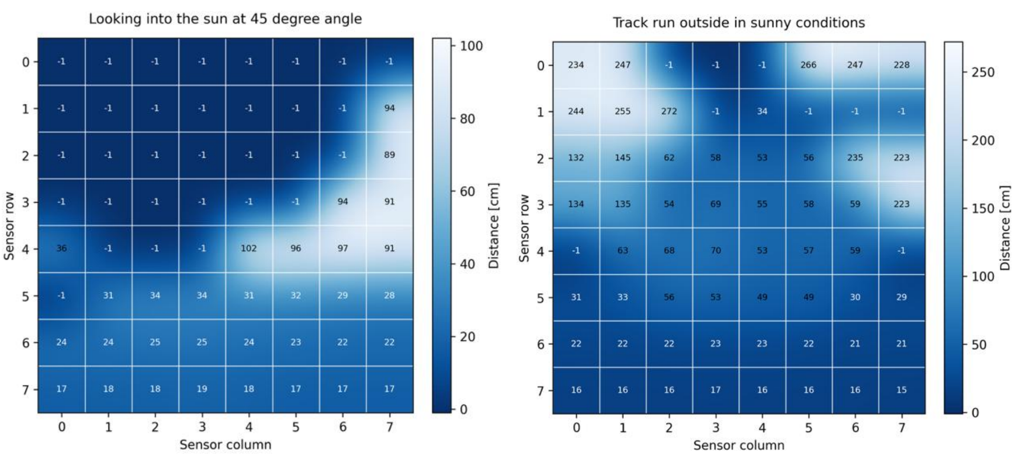

Fig.3. Outdoor sunlight comparison: normal track-facing reading versus direct sun-facing reading.

4.4 Tunnel pattern

The tunnel was measured from 30 cm down to 10 cm. At 25 cm, the matrix clearly showed the tunnel frame and a further center. The center cells reached roughly 33-42 cm. That means the sensor was seeing into the hollow part of the tunnel. At 20 cm, the tunnel pattern was still visible. At 15 cm and under, the sensor started to see more of the inside of the tunnel rather than only the entrance. This is expected because the robot is close enough that the entrance frame no longer fills the same part of the field of view.

A moving test was performed with the robot instructed to stop when it recognized a hollow tunnel-like object. The robot stopped successfully at about 17 cm from the entrance. The difference between detection distance and stop distance is probably caused by speed, wheel drift, and mechanical inertia.

4.5 Curtain pattern

The curtain was the widest tested object. At 25 cm it filled the whole useful frame with similar values, mostly around 23-27 cm. This is different from the brick, where the side columns can still see background, and different from the tunnel, where the centre is further away. The curtain is made of vertical paper strips which also produced small differences between columns and rows. Some strips were hanging slightly closer and some further. This is an interesting because it shows that even an 8x8 matrix can reveal that the surface is not homogeneous.

4.6 Brick pattern

The brick was measured as a smaller solid object. At 25 cm, the centre cells showed the close object, while some side and top cells showed much higher distances or invalid values because the object did not fill the whole field of view. This is the expected pattern for a compact obstacle. Compared with the curtain, the brick is narrower. Compared with the tunnel, the centre is not hollow. Therefore, the correct action is to perform an avoidance maneuver and return to the line after the obstacle.

At 15 cm and 10 cm, the brick occupied much more of the matrix and was less likely to be distinguished from the curtain. This confirms why 25 cm is better for classification purpose. Too close to the obstacle, several object types can start to look alike. The robot can still detect that something is present, but it loses part of the shape information. Therefore, the first classification should happen earlier, and the closer readings should be used for confirmation and safe maneuver timing.

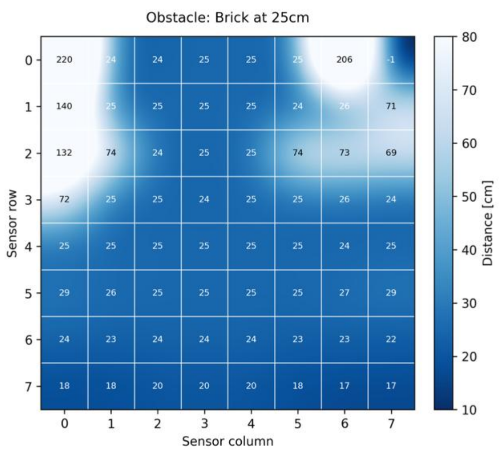

Fig.4. Heat map of a brick at distance of 25 cm.

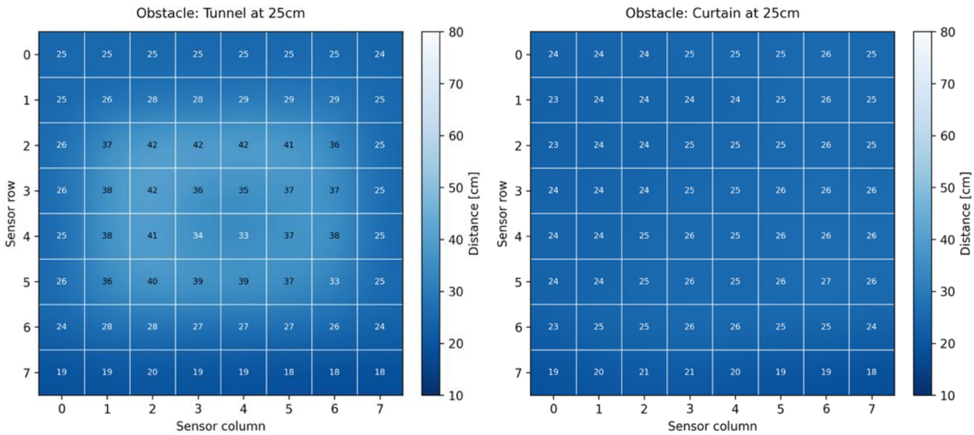

Fig.5. Heat map of a tunnel and curtain at distance of 25 cm before entrance.

5 Discussion

The results support the basic assumption of this work: an 8x8 ToF matrix contains more useful information than a single distance measurement. It is not only possible to detect that an obstacle is present, but also to reason about its spatial pattern. This is especially visible at 25 cm, where the measured data show distinguishable differences between brick, curtain, and tunnel. The tunnel has a hollow centre, the curtain is wide and non-homogeneous, and the brick is compact.

The approach is still not universal. If the field is densely populated with several obstacles placed close to each other, for example with only 10 cm between them, the robot may not have enough time or enough spatial resolution to separate them reliably. In that case, the order of obstacles would need to be hard-coded or the robot would require additional sensing.

Another limitation is motion. Static tests are cleaner than moving tests. When the robot moved, the amount of invalid -1 readings increased at very short distances. My current theory is that vibrations from the motors and chassis propagate into the VL53L8CX board and disturb the measurement. Soft mounting with rubber dampers reduced the number of failed readings, but this needs proper testing before a strong conclusion can be made. A future experiment should compare rigid mounting and soft mounting at several vibration levels and robot speeds.

Latency is also important. The sensor was run over SPI, and Pololu specifies SPI operation up to 3 MHz [1]. In this implementation, SPI communication was reliable and much more practical than the failed I2C attempt. However, the full robot reaction time is not only sensor communication time. It also includes ranging update time, firmware processing, control-loop timing, motor response, wheel slip, and the physical stopping distance. For a fast robot, the useful question is not only how fast the sensor can read, but how far the robot travels between the first valid obstacle pattern and the completed maneuver.

For practical competition use, the sensor could support two stages. At longer distance, the robot can slow down when it sees that something is present in front of it. At approximately 25 cm, it can classify the pattern. At 15 cm and below, it should already be executing or confirming the selected maneuver. This strategy is more robust than simply one value obstacle detection.

Conclusion

This article presented the implementation and preliminary evaluation of an 8x8 Time-of-Flight obstacle detection module for a line-following robot. The VL53L8CX sensor was integrated with a Teensy 4.1 platform over SPI. The sensor produced an 8x8 distance matrix that can used for rule-based obstacle interpretation.

The measured data show that the practical working range for this application is approximately 10-30 cm. The most useful classification distance in this dataset is about 25 cm. At this distance, the three tested obstacle types show different spatial patterns: a brick is compact, a curtain is wide and slightly non-homogeneous, and a tunnel has a hollow centre. These differences are large enough to support practical robot decisions: go around the brick, pass through the curtain, and continue through the tunnel.

The solution is not perfect and should not be presented as a universal vision system. Closely spaced obstacles, direct sunlight, vibration, and very short distances can still create problems. However, for common line-follower obstacle scenarios, the method provides a useful middle ground between a simple ultrasonic distance sensor and a full camera-based solution.

Acknowledgement

This contribution was prepared as part of bachelor-thesis work focused on control algorithms of a line-following robot with obstacle detection. The author thanks his bachelor-thesis supervisor, Ing. Juraj Štefanovič, PhD. for his guidance and the faculty for the opportunity to connect practical embedded robotics development with experimental evaluation.

References

[1] Pololu Corporation. (2025). VL53L8CX Time-of-Flight 8x8-Zone Distance Sensor Carrier with Voltage Regulators, 400cm Max. Retrieved online, May 11, 2026, from https://www.pololu.com/product/3419

[2] Robotika.SK. (2019). Pravidla kategorie Stopar / Line-follower category rules. Retrieved online, May 11, 2026, from https://robotika.sk/contest/2019/index.php?page=rules&type=follower

[3] Roboticky den. (2026). Roboticky den 2026 and Line Follower rules. Retrieved online, May 11, 2026, from https://robotickyden.cz/2026/

[4] STMicroelectronics. (2024). VL53L8CX multizone Time-of-Flight ranging sensor documentation and API resources. Retrieved online, May 11, 2026, from https://www.st.com/

Authors

Samuel Ladislav Šindler
Faculty of Informatics, Pan-European University, Bratislava, Slovakia
xsindler@paneurouni.com
Samuel studies applied informatics and works on a bachelor thesis focused on control algorithms for a line-following robot with 3D obstacle detection. His practical background includes embedded systems, robotics, Linux-based development, custom kernel development, electronics, and experimental work with drones, camera, sensor, and mobile-robot platforms.