MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE
NATIONAL AVIATION UNIVERSITY
Faculty of Aeronautics, Electronics and Telecommunications, Department of Aviation Computer-Integrated Complexes

ACCEPT TO PROTECTION
Head of Department
_____ Viktor SINEGLAZOV
“_”_____2023 y.

QUALIFICATION PAPER (EXPLANATORY NOTE)

HIGHER EDUCATION STUDY

“MASTER”

Specialty 151 "Automation and computer-integrated technologies"
Educational and professional program "Information support and engineering of aviation computer systems"

Subject: Obstacle detection system for ground robotic complex

Performer: student of the group І3-225M Oleksandr Pokatilov
Supervisor: Associate professor, Mykola Vasylenko

Consultant of the "Environmental Protection" section_______ Radomska M.M.
Consultant of the "Occupational safety and health" section________Кazhan.K.I.
Norm control:___________Fylashkin M.К.

Kyiv 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій
Кафедра авіаційних комп’ютерно-інтегрованих комплексів

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
________Віктор СИНЄГЛАЗОВ
“___”_______________2023 р.

КВАЛІФІКАЦІЙНА РОБОТА (ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ

“МАГІСТР”

Спеціальність 151 «Автоматизація та компю’терно-інтегровані технології»
Освітньо-професійна програма «Інформаційне забезпечення та інженерія авіаційних комп’ютерних система»

Тема: Система виявлення перешкод для наземного робототехнічного комплексу

Виконавець: студент групи ІЗ-225М Покатілов Олександр Володимирович
Керівник: доцент, Василенко Микола Павлович

Консультант розділу «Охорона навколишнього середовища»_____ Радомська М.М.
Консультант розділу «Охорона праці»_______Кажан.К.І.
Нормоконтролер:________Філяшкін М.К.

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
Факультет аеронавігації, електроніки та телекомунікацій

Кафедра авіаційних комп’ютерно-інтегрованих комплексів

Освітній ступінь: Магістр

Спеціальність: 151 " Автоматизація та комп'ютерно-інтегровані технології"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віктор СИНЄГЛАЗОВ

“ ____ ” __________2023 р.

ЗАВДАННЯ

на виконання дипломної роботи студента

Покатілова Олександра Володимировича

1. **Тема проекту (роботи):** “Система виявлення перешкод для наземного робототехнічного комплексу”
2. **Термін виконання проекту (роботи):** з 01.12.2023 р. до 27.12.2023 р.
3. **Вихідні данні до проекту (роботи):** спосіб визначення дистанції до об'єкта, оточення Arduino IDE та Proteus VSM.
4. **Зміст пояснювальної записки (перелік питань, що підлягають розробці):** 1. Актуальність системи виявлення перешкод для наземного робототехнічного комплексу;

   2. Аналіз існуючих рішень для вирішення проблеми;

   3. Розробка системи виявлення перешкод для наземного робототехнічного комплексу з використанням технології лідар;

   4. Моделювання і тестування з порівнянням для розробленої системи.

   Перелік обов’язкового графічного матеріалу: 1. Структурна схема системи виявлення перешкод; 2. Схема роботи системи; 3. Структурна схема лідара; 4. Таблиця перевірки точності, порівняння лідара з ультразвуковим та радіо альтімертрами;

5. Календарний план-графік

№ пор.	Завдання	Термін виконання	Відмітка про виконання
1.	Отримання завдання	02.10.2023 – 03.10.2023	
2.	Формування мети та основних завдань дослідження	03.10.2023 – 05.10.2023	
3.	Аналіз існуючих рішень	07.10.2023 – 15.10.2023	
4.	Теоретичний розгляд рішення задачі	17.10.2023 – 01.11.2023	
5.	Розробка схеми системи виявлення перешкод для для наземного робототехнічного комплексу	01.11.2023 – 15.11.2023	
6.	Розробка програмного та апаратного забезпечення	20.11.2023 – 05.12.2023	
7.	Оформлення пояснювальної записки	07.12.2023 – 10.12.2023	
8.	Підготовка презентації та роздаткового матеріалу	12.12.2023– 17.12.2023	

6. Консультанти з окремих розділів

Розділ	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Охорона праці	Доцент Катерина КАЖАН		
Охорона навколишнього середовища	Доцент Маргарита РАДОМСЬКА		

7. Дата видачі завдання: “2” жовтня 2023 р.

Керівник дипломної роботи ______________ Микола ВАСИЛЕНКО

Завдання прийняв до виконання _________________ Олександр ПОКАТІЛОВ

NATIONAL AVIATION UNIVERSITY

Faculty of aeronavigation, electronics and telecommunications

Department of Aviation Computer Integrated Complexes

Educational level: Master

Specialty: 151 "Automation and computer-integrated technologies"

APPROVED

Head of Department

Viktor SINEGLAZOV

"____" __________2023

TASK

For the student's thesis

Pokatilov Oleksandr Volodymyrovych

1. **Theme of the project:** "Obstacle detection system for ground robotic complex"
2. **The term of the project (work):** from December 01, 2023 until December 27, 2023
3. **Output data to the project (work):** method of determining the distance to the object, the environment of Arduino IDE and Proteus VSM.
4. **Contents of the explanatory note (list of questions to be developed):**
   1. The relevance of the obstacle detection system for the ground robotics complex;
   2. Analysis of existing solutions to solve the problem;
   3. Development of an obstacle detection system for the ground robotic complex using lidar technology;
   4. Modeling and testing with comparison for the developed system.

   List of compulsory graphic material: 1. Structural diagram of the obstacle detection system; 2. Scheme of system operation; 3. Structural diagram of the lidar; 4. Accuracy check table, comparison of lidar with ultrasonic and radio altimeters;

5. **Planned schedule:**

№	**Task**	**Execution term**	**Execution mark**
1.	Task	02.10.2023 – 03.10.2023	
2.	Purpose formation and describing the main research tasks	03.10.2023 – 05.10.2023	
3.	Analysis of existing solutions	07.10.2023 – 15.10.2023	
4.	Analysis of existing systems	17.10.2023 – 01.11.2023	
5.	Development of the obstacle detection system scheme for the ground robotic complex	01.11.2023 – 15.11.2023	
6.	Software and hardware development	20.11.2023 – 05.12.2023	
7.	Making an explanatory note	07.12.2023 – 10.12.2023	
8.	Preparation of presentation and handouts	12.12.2023– 17.12.2023	

6. **Consultants from individual sections**

Section	Consultant	Date, signature	
		Issued the task	Accepted the task
Occupational safety and health	Associate Professor Kateryna KAZHAN		
Environmental protection	Associate Professor Margarita RADOMSKA		

7. **Date of task receiving:** “2” October 2023

Diploma thesis supervisor Mykola VASYLENKO
(signature)

Issued task accepted Oleksandr POKATILOV
(signature)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи « Система виявлення перешкод для наземного робототехнічного комплексу»: 80 с., 34 рис., 3 табл., 26 літературних джерела.

Об’єкт дослідження: датчик Лідар.

Мета роботи: розробка системи виявлення перешкод для наземного робототехнічного комплексу.

Для досягнення цієї мети необхідно розв’язати наступні завдання:

- проаналізувати існуючі види альтиметрів та їх недоліки;
- проаналізувати основні елементи системи виявлення перешкод для наземного робототехнічного комплексу;
- розробити структуру системи виявлення перешкод для наземного робототехнічного комплексу;
- розробити програмне та апаратне забезпечення реалізації обраного рішення;
- провести експериментальне дослідження роботи на базі розробленої системи.

Предмет дослідження: розробка системи виявлення перешкод для наземного робототехнічного комплексу на базі технології Lidar.

Методи дослідження: теоретична фізика, теоретична електроніка, теорія визначення висоти на основі світла.

СИСТЕМИ ВИЯВЛЕННЯ ПЕРЕШКОД; ДАТЧИК ЛІДАР; МІКРОЕЛЕКТРОМЕХАНІЧНІ СИСТЕМИ.

ABSTRACT

Explanatory note to the thesis "Obstacle detection system for the ground robotic complex": 80 pages, 34 figures, 3 tables, 26 literary sources.

Research object: Lidar sensor.

The purpose of the work: development of an obstacle detection system for the ground robotics complex.

To achieve this goal, the following tasks must be solved:

- analyze the existing types of altimeters and their shortcomings;
- analyze the main elements of the obstacle detection system for the ground robotics complex;
- to develop the structure of the obstacle detection system for the ground robotics complex;
- develop software and hardware for implementing the chosen solution;
- conduct an experimental study of work based on the developed system.

Research subject: development of an obstacle detection system for a ground robotic complex based on Lidar technology.

Research methods: theoretical physics, theoretical electronics, theory of height determination based on light.

OBSTACLE DETECTION SYSTEMS; LIDAR SENSOR; MICROELECTROMECHANICAL SYSTEMS.

CONTENT

LIST OF ABBREVIATIONS

MEMS – micro-electromechanical systems.

IDE – integrated development environment.

LIDAR – light detecting and ranging.

Introduction

In the rapidly evolving landscape of robotics and automation, the ability to navigate complex terrains and environments is paramount for ground-based robotic systems. These robots, whether deployed for industrial, agricultural, or exploratory purposes, often encounter a myriad of obstacles in their operational environments. The presence of such obstacles not only poses a risk to the robot's integrity but also to its surrounding environment and any humans in proximity.

An effective obstacle detection system is, therefore, a critical component for ensuring the safe and efficient operation of these robotic complexes. Such systems employ a combination of sensors, algorithms, and computational techniques to identify and circumvent potential hindrances, allowing the robot to execute its tasks seamlessly. The significance of these systems is further underscored by the increasing integration of robots in urban settings, where the density of obstacles is considerably higher.

This paper delves into the intricacies of designing and implementing an obstacle detection system tailored for ground-based robotic complexes. We will explore the current state-of-the-art technologies, challenges faced in real-world scenarios, and potential solutions to enhance the robustness and accuracy of these systems.

1. RELEVANCE OF THE WORK

Setting the Context for Obstacle Detection in Ground-Based Robotic Systems. The dawn of the 21st century has witnessed an unprecedented surge in technological advancements, particularly in the field of robotics. Ground-based robotic systems, once a figment of science fiction, have now become a tangible reality, permeating various sectors of our global economy. From automated warehouses and precision agriculture to urban delivery systems and disaster response units, these robots are reshaping the way we perceive automation.

However, as with any technological leap, the integration of robots into our daily lives and workspaces brings forth a set of challenges. One of the most pressing of these challenges is ensuring that these robots can navigate their operational environments safely and efficiently. This is not merely a technical challenge but a fundamental requirement to ensure the harmonious coexistence of robots, humans, and the environments they operate in.

Obstacle detection, in this context, emerges as a critical area of study and development. It serves as the eyes and ears of these robotic systems, allowing them to perceive, understand, and respond to their surroundings. The efficacy of an obstacle detection system can be the difference between a robot seamlessly performing its task or causing a catastrophic failure.

Fig. 1.1 Automated warehouse work

This chapter delves deep into the relevance of this work, exploring the multifaceted implications of obstacle detection, from safety and economic perspectives to technological and environmental considerations.

1.1 The Evolution and Significance of Ground-Based Robotic Systems in Modern Society

The trajectory of technological progress has always been marked by innovations that redefine the boundaries of what's possible. Ground-based robotic systems stand as a testament to this relentless pursuit of advancement. Their evolution is not just a story of technical achievements but also of societal adaptation and the reshaping of industries.

Historical Context

The origins of ground-based robotic systems can be traced back to the early experiments in automation and mechanization. From rudimentary automated carts to the sophisticated robots of today, the journey has been marked by continuous innovation. Each iteration brought forth improvements in mobility, autonomy, and adaptability, making these systems more aligned with the needs of the times.

Industry Transformation

As these robotic systems evolved, so did their applications. Industries that were once labor-intensive began to see the potential of automation. Manufacturing units started employing robots for tasks that were repetitive, hazardous, or required precision. Agriculture saw the introduction of robots for tasks like planting, harvesting, and monitoring. The logistics sector employed them for warehousing and delivery. The list goes on, with each industry finding unique applications for these versatile machines.

Fig. 1.2 Autonomous robots of the hand type

Societal Implications

The rise of ground-based robotic systems has had profound societal implications. On one hand, they have led to increased efficiency, reduced costs, and the possibility of operations in environments that were previously deemed hazardous for humans. On the other hand, they have raised questions about job displacement, ethics, and the need for regulations.

Challenges and the Need for Obstacle Detection

In the age of automation, where robots are no longer confined to isolated labs but are actively integrated into shared spaces with humans, safety emerges as a paramount concern. The coexistence of humans and robots in the same environment brings forth a unique set of challenges and considerations.

1.2 Safety Concerns: The Imperative of Ensuring Human-Robot Coexistence in Shared Environments

In the age of automation, where robots are no longer confined to isolated labs but are actively integrated into shared spaces with humans, safety emerges as a paramount concern. The coexistence of humans and robots in the same environment brings forth a unique set of challenges and considerations.

Human-Robot Interaction

The dynamics of human-robot interaction are complex. While robots are programmed to follow specific instructions, humans are unpredictable. A child might suddenly run in front of a delivery robot, or a worker might inadvertently step into the path of an industrial robot. Ensuring that robots can anticipate and respond to such unpredictable behaviors is crucial.

Physical Safety

Physical safety pertains to preventing collisions and accidents. A robot's inability to detect an obstacle or a human can lead to collisions, causing potential harm or even fatalities. This is especially concerning in high-speed scenarios or where robots carry heavy loads.

Psychological Safety

Beyond physical safety, there's the aspect of psychological safety. Humans need to feel safe around robots. If a robot moves unpredictably or invades personal spaces, it can lead to discomfort and anxiety. Ensuring that robots move in a manner that's predictable and respectful of human boundaries is essential.

Operational Safety

Operational safety concerns the safe functioning of robots in their designated tasks. For instance, a robot tasked with transporting hazardous materials needs to ensure it doesn't spill or drop its load. An obstacle detection system plays a crucial role in ensuring the robot can navigate its path without disturbances.

The Role of Obstacle Detection

In the context of these safety concerns, obstacle detection emerges as a foundational component. It serves as the primary mechanism through which robots can be aware of their surroundings, detect potential threats, and take corrective actions. Whether it's avoiding a pedestrian on the sidewalk, stopping for a worker in a factory, or navigating around a fallen tree in a forest, the ability to detect obstacles is the first line of defense in ensuring safety.

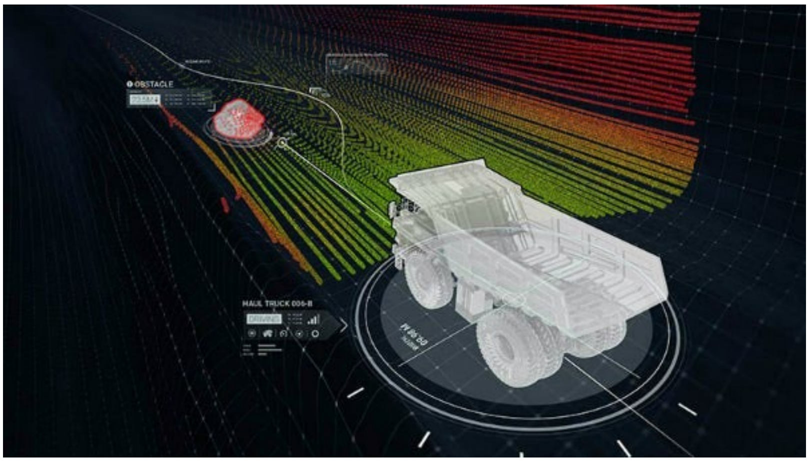


Fig. 1.3 Obstacles illustration

1.4Economic Implications: The Financial Impact of Ground-Based Robotic Systems and the Role of Obstacle Detection

The integration of ground-based robotic systems into various sectors has not only transformed operational methodologies but has also brought about significant economic implications. The financial impact of these systems, both positive and negative, is influenced by their efficiency, reliability, and the effectiveness of their obstacle detection capabilities.

Cost Savings and Efficiency

One of the primary motivations behind the adoption of robotic systems in industries is the potential for cost savings. Robots, when functioning optimally, can operate continuously without the need for breaks, leading to increased productivity. Moreover, they can perform tasks with precision, reducing errors and wastage. The effective detection of obstacles ensures that these robots can operate without interruptions, further enhancing efficiency and leading to direct economic benefits.

Investment and Maintenance Costs

While robotic systems offer potential cost savings in operations, they also come with significant upfront investment costs. Advanced robots equipped with state-of-the-art obstacle detection systems can be expensive. Additionally, maintenance, software updates, and potential repairs add to the ongoing costs. The reliability of the obstacle detection system plays a crucial role here; frequent malfunctions or failures can lead to increased maintenance costs and downtime.

Job Displacement and Retraining

The economic implications of robotic systems extend beyond direct operational costs. As robots take on tasks traditionally performed by humans, there’s a potential for job displacement. This has broader economic ramifications, including the need for retraining workers and potential unemployment challenges. Effective obstacle detection systems, by ensuring the smooth operation of robots, might accelerate this trend, making it imperative for industries and governments to address the socio-economic challenges associated.

Market Growth and Innovation

The demand for advanced obstacle detection systems has spurred growth in the tech industry. Companies specializing in sensors, AI algorithms, and robotics have seen increased investment and growth. This not only leads to economic benefits in terms of market growth but also drives innovation, as companies compete to develop more advanced and reliable systems.

Liabilities and Insurance

Accidents or malfunctions involving robots can lead to significant liabilities, especially if they result in harm to humans or damage to property. Insurance premiums for companies deploying robotic systems might be influenced by the reliability of their obstacle detection systems. A robust and reliable system could lead to reduced insurance costs, while a system prone to failures might result in higher premiums.

1.4 Technological Advancements and Their Relevance: A Deep Dive into the Innovations in Obstacle Detection for Ground-Based Robotic Systems

The world of robotics has been a hotbed of innovation, with continuous advancements pushing the boundaries of what’s achievable. Within this domain, the field of obstacle detection has seen some of the most groundbreaking developments. These technological leaps are not just academic achievements; they have direct implications for the real-world performance and capabilities of robotic systems.

Sensor Evolution

The journey of obstacle detection began with rudimentary sensors that could only detect immediate obstacles. Over time, these sensors evolved:

- **Ultrasonic Sensors:** These sensors use sound waves to detect obstacles. Their range and accuracy have improved significantly over the years, making them suitable for close-range detections.
- **Infrared Sensors:** Using infrared light, these sensors can detect heat and proximity. They are particularly useful in low-light conditions.
- **LiDAR:** A more recent innovation, LiDAR uses laser beams to map surroundings in 3D. It offers high precision and is becoming increasingly popular in autonomous vehicles.

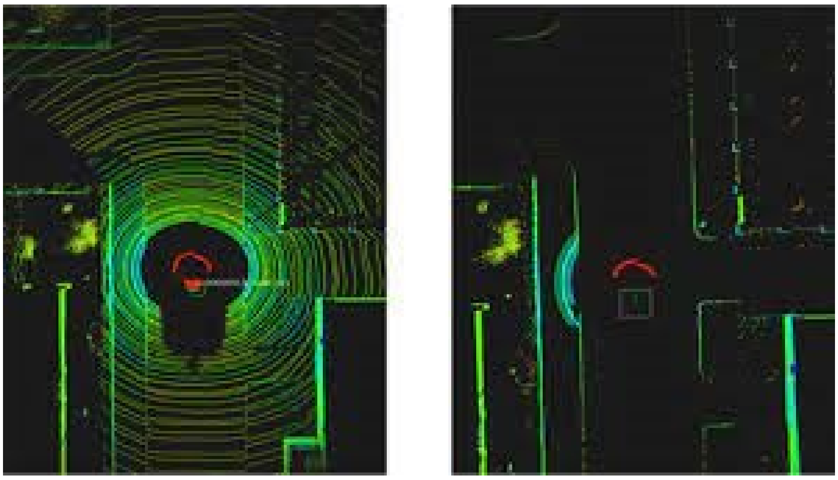


Fig. 1.4 Circular lidar work

Integration and Fusion

A single sensor or algorithm is rarely sufficient. Modern robotic systems integrate multiple sensors and data sources, fusing the information to get a comprehensive understanding of their surroundings.

Relevance to Real-World Applications

The advancements in obstacle detection technology have direct implications for real-world applications:

- **Autonomous Vehicles:** The safety and reliability of self-driving cars hinge on their ability to detect and navigate around obstacles.
- **Industrial Robots:** In factories, robots need to operate alongside human workers, making accurate obstacle detection crucial.
- **Agricultural Robots:** In fields, robots encounter varied obstacles, from plants to animals. Advanced detection systems ensure they can operate without causing damage.

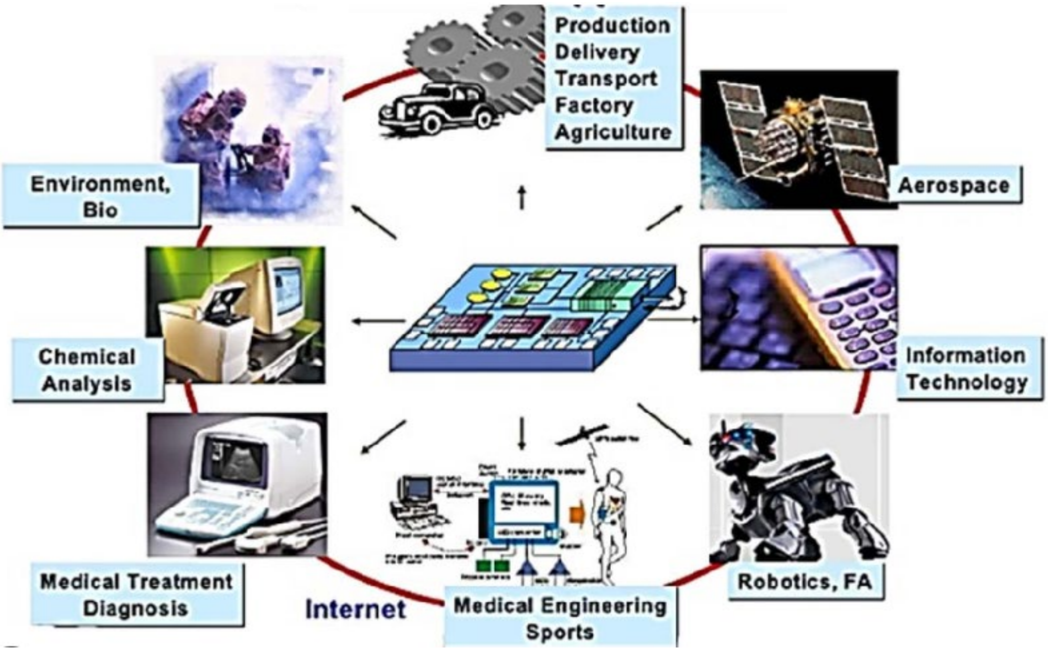


Fig. 1.5 Fields of application of robots

1.5 The Environmental Aspect: Exploring the Interactions of Ground-Based Robotic Systems with Natural and Urban Ecosystems

In the discourse surrounding robotics, the focus often gravitates towards technological prowess, economic implications, and safety concerns. However, an equally important dimension is the environmental aspect. As ground-based robotic systems become more prevalent, their interactions with both natural and urban ecosystems become crucial. This section delves into the environmental considerations associated with these robotic systems and the role of obstacle detection in mitigating potential ecological impacts.

Natural Ecosystems

Robotic systems are increasingly being deployed in natural environments for various purposes:

- **Conservation Efforts:** Robots are used for monitoring wildlife, tracking endangered species, and even replanting forests. Their ability to navigate dense forests or rugged terrains without disturbing the natural habitat is vital.
- **Research and Exploration:** In remote or inaccessible areas, robots serve as the eyes and ears of researchers, collecting data and samples. Their non-intrusive presence ensures minimal disruption to the environment.

Urban Ecosystems

In urban settings, the environmental considerations are different but equally significant:

- **Waste Management:** Robots are being employed for tasks like waste sorting and recycling. Effective obstacle detection ensures they can operate in cluttered environments without causing spillages or mishandling materials.
- **Green Infrastructure:** Robots are used in maintaining urban green spaces, gardens, and vertical farms. Their interactions with plants, soil, and water systems need to be precise to avoid damages.

Environmental Footprint

The manufacturing, operation, and disposal of robotic systems have environmental implications:

- **Resource Consumption:** The materials used in building robots, especially rare metals for sensors, have environmental extraction costs.
- **Energy Use:** The energy consumption patterns of robots, especially those operating continuously, contribute to their carbon footprint.

Role of Obstacle Detection

In the context of these environmental interactions, obstacle detection plays a pivotal role:

- **Minimizing Disturbances:** In natural habitats, the ability to detect and avoid animals, plants, or water bodies ensures that the robot doesn't cause disturbances or harm.
- **Efficient Operations:** In urban settings, avoiding obstacles means less rerouting and more efficient operations, leading to energy savings.

1.6 Synthesizing the Multifaceted Implications of Obstacle Detection in Ground-Based Robotic Systems

As we navigate through the intricate tapestry of technological advancements, economic considerations, safety protocols, and environmental interactions associated with ground-based robotic systems, one element consistently emerges as a linchpin: obstacle detection. This conclusion aims to synthesize the multifaceted implications of this critical component, drawing connections between the various dimensions explored in the preceding sections.

Technological Significance

The technological landscape of robotics has been in a state of perpetual evolution. From rudimentary sensors to sophisticated AI-driven algorithms, the journey of obstacle detection mirrors the broader trajectory of robotic advancements. It stands as a testament to the relentless pursuit of innovation, constantly pushing the boundaries of what's achievable.

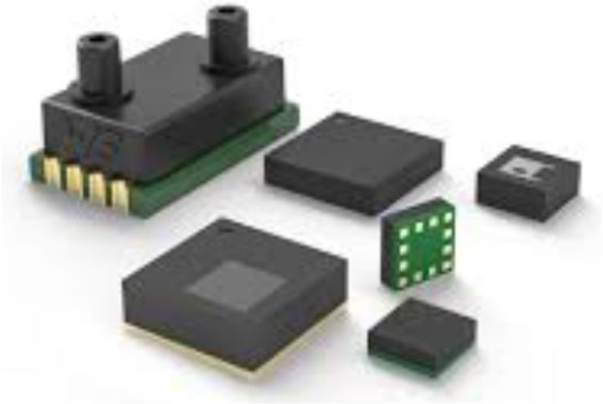


Fig. 1.6 MEMS components

Economic Ramifications

Beyond the realm of technology, the economic implications of effective obstacle detection are profound. By ensuring seamless operations, reducing downtimes, and minimizing accidents, obstacle detection systems directly influence the financial viability of robotic integrations in various sectors.

Safety: A Non-Negotiable Priority

At the intersection of technology and economics lies the paramount concern of safety. The coexistence of humans and robots in shared spaces underscores the need for impeccable obstacle detection. It's not just about preventing collisions; it's about fostering trust and ensuring that robots can be integrated into our lives without reservations.

Environmental Stewardship

In an age where environmental considerations are becoming central to all technological deployments, the role of obstacle detection in ensuring that robots operate with minimal ecological impact cannot be overlooked. Whether it's navigating dense forests without disturbing wildlife or maintaining urban green spaces, the environmental stewardship of robotic systems is intrinsically linked to their obstacle detection capabilities.

The Road Ahead

As we look to the future, the relevance of obstacle detection in ground-based robotic systems is poised to grow. With robots becoming more integrated into our daily lives, industries, and ecosystems, the challenges they face will become more complex. The obstacle detection systems of tomorrow will need to be more advanced, more reliable, and more adaptable to ensure that robots can meet these challenges head-on.

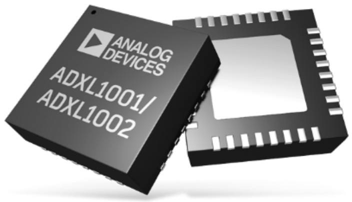


Fig. 1.7 MEMS microcontroller SOC

2. ANALYSIS OF EXISTING SOLUTIONS FOR SOLVING THE PROBLEM

The realm of obstacle detection for ground-based robotic systems has witnessed a plethora of algorithmic advancements over the years. These algorithms, each with its unique approach and methodology, aim to address the challenges of detecting and navigating around obstacles. This chapter delves into a comprehensive analysis of the existing algorithms, evaluating their strengths, weaknesses, and overall effectiveness in solving the problem at hand.

2.1 Background on Micro Electro Mechanical System (MEMS)

Micro Electro Mechanical Systems, commonly known as MEMS, represent a convergence of multiple disciplines, bringing together the principles of mechanics, electronics, and microfabrication. This section aims to provide a foundational understanding of MEMS, exploring their definitions, characteristics, fabrication processes, and applications.

2.1.1 Definition of MEMS

MEMS are miniaturized devices or systems that combine both electrical and mechanical components. They are typically smaller than the width of a human hair, ranging in size from a few micrometers to millimeters.

2.1.2 Characteristics of MEMS

- **Miniaturization:** The most defining characteristic of MEMS is their small size. This miniaturization allows for faster response times and reduced material consumption.
- **Integration:** MEMS devices often integrate multiple functions (sensing, actuation, and computation) into a single chip, enhancing their functionality.
- **Low Power Consumption:** Due to their small size, MEMS devices typically consume less power, making them ideal for battery-operated applications.

2.1.3 Fabrication of MEMS

MEMS are fabricated using microfabrication techniques, which are similar to the processes used in the semiconductor industry. The primary steps include:

- **Deposition:** Layers of materials are deposited on a substrate.
- **Patterning:** Photolithography techniques are used to define patterns for each layer.
- **Etching:** Unwanted material is removed to create the desired structures.

2.1.4 Types of MEMS Devices

MEMS technology has given rise to a variety of devices, including:

- **Sensors:** Devices that detect changes in their environment, such as pressure, temperature, or motion.
- **Actuators:** Devices that perform a specific action in response to an external signal, such as moving a micro-mirror or opening a valve.
- **Microfluidic Devices:** Systems designed to handle small volumes of fluids, often used in biomedical applications.

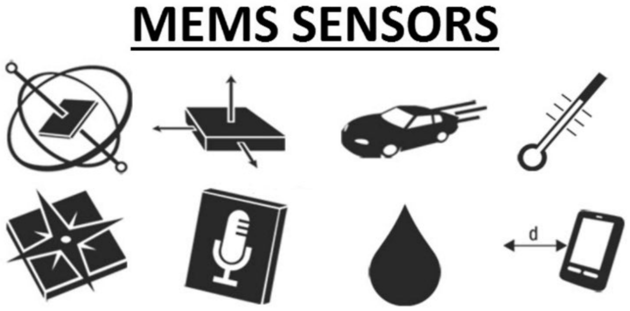


Fig. 2.1 MEMS sensors types

2.1.5 Applications of MEMS

MEMS have found applications across a wide range of industries:

- **Consumer Electronics:** Used in smartphones for functions like motion sensing and image stabilization.
- **Automotive:** Employed in car safety systems, such as airbag deployment sensors.

- **Healthcare:** Utilized in devices like hearing aids and lab-on-a-chip diagnostic tools.
- **Aerospace:** Integrated into navigation systems and pressure sensors in aircraft.

2.2 MEMS in Obstacle Detection: Harnessing Miniaturized Systems for Enhanced Sensing Capabilities

The integration of MEMS technology into obstacle detection systems has revolutionized the way robots perceive and interact with their environment. This section aims to elucidate the role of MEMS in obstacle detection, exploring their inherent advantages, types of MEMS-based sensors used, and their real-world applications.

2.2.1 Why MEMS for Obstacle Detection?

The choice of MEMS for obstacle detection stems from several inherent advantages:

- **High Sensitivity:** Due to their miniaturized nature, MEMS sensors can detect minute changes in their environment, offering high-resolution data.
- **Low Power Consumption:** MEMS devices, given their small size, typically consume less power, making them ideal for battery-operated robotic systems.
- **Compactness:** Their small form factor allows for easy integration into robotic systems without adding significant weight or bulk.

2.2.2 Types of MEMS Sensors in Obstacle Detection

Several MEMS-based sensors have found applications in obstacle detection:

- **MEMS Accelerometers:** These sensors measure changes in velocity, helping robots detect movements and adjust their paths accordingly.
- **MEMS Gyroscopes:** Used to measure angular velocity, aiding in maintaining the robot's orientation and balance.
- **MEMS Proximity Sensors:** These sensors detect the presence of nearby objects without any physical contact, making them ideal for obstacle detection.

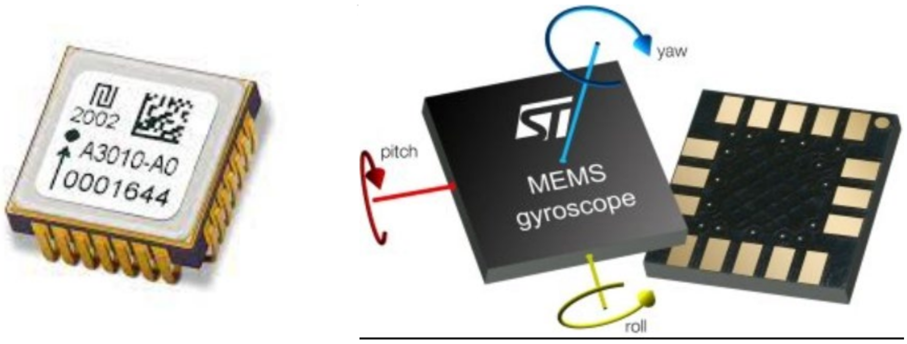


Fig. 2.2 MEMS gyroscope

2.2.3 Integration with Algorithms

The raw data from MEMS sensors is processed using algorithms to make it actionable:

- **Filtering Algorithms:** Used to remove noise from the sensor data, ensuring accurate obstacle detection.
- **Pattern Recognition:** Algorithms analyze the data patterns to identify and categorize obstacles.
- **Predictive Algorithms:** Based on the data, these algorithms predict the movement of obstacles, allowing robots to plan their paths proactively.

2.2.4 Real-World Applications of MEMS in Obstacle Detection

MEMS-based obstacle detection systems have been integrated into various robotic applications:

- **Autonomous Vehicles:** MEMS sensors help these vehicles detect pedestrians, other vehicles, and obstacles in real-time, ensuring safe navigation.
- **Drones:** MEMS sensors aid drones in avoiding collisions with buildings, trees, and other aerial obstacles.
- **Industrial Robots:** In factories, MEMS-based systems ensure that robots can operate safely alongside human workers, detecting and avoiding any potential collisions.

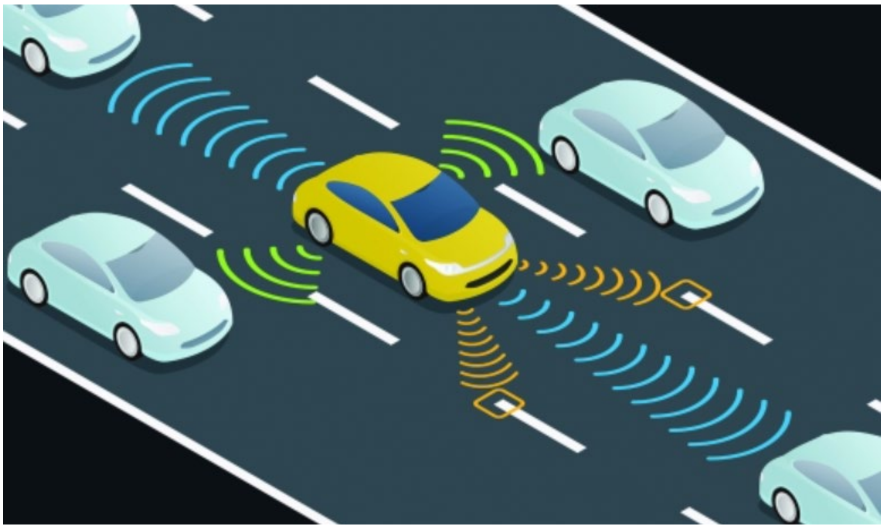


Fig. 2.3 Detection of potential obstacles in the traffic flow

2.3 Existing Algorithms Leveraging MEMS

As MEMS technology has matured, it has become an integral component in the development of algorithms for obstacle detection. These algorithms, tailored to harness the unique capabilities of MEMS sensors, offer enhanced accuracy and responsiveness. This section delves into the various algorithms that have been developed to leverage MEMS in the realm of obstacle detection.

2.3.1 Threshold-Based Detection

- This is a rudimentary algorithm that involves setting a predetermined threshold value for the sensor readings. When the reading from a MEMS sensor exceeds or falls below this threshold, an obstacle is detected.
- Often used in proximity sensors where a sudden change in readings indicates the presence of an obstacle.

2.3.2 Temporal Analysis

- This algorithm analyzes the temporal (time-based) data from MEMS sensors. By studying the patterns and sequences of readings over time, it can predict the movement and speed of obstacles.

- Useful in dynamic environments where obstacles are in motion, such as autonomous vehicles navigating traffic.

2.3.3 Spatial Analysis

- Spatial analysis focuses on the data from multiple MEMS sensors to create a comprehensive map of the environment. By comparing readings from different sensors, it can determine the position, size, and shape of obstacles.
- Employed in robotic systems that require a 3D understanding of their surroundings, such as drones or robots navigating complex terrains.

2.3.4 Data Fusion Techniques

- Given that a single MEMS sensor type might not provide a complete picture, data fusion techniques combine information from multiple sensor types to enhance detection accuracy.
- Common in advanced robotic systems where multiple sensors (e.g., accelerometers, gyroscopes, proximity sensors) work in tandem to ensure comprehensive obstacle detection.

2.3.5 Machine Learning-Driven Approaches

- With the advent of AI and machine learning, algorithms have been developed that can learn and adapt based on the data they receive from MEMS sensors. These algorithms can improve their detection accuracy over time.
- Used in scenarios where the environment is unpredictable and constantly changing, requiring the algorithm to adapt and learn from new data.

2.4 Comparative Analysis: MEMS-Based Algorithms vs. Traditional Methods

In the evolving landscape of obstacle detection, both MEMS-based algorithms and traditional methods have their unique strengths and limitations. This section aims to provide a comparative analysis, shedding light on the nuances of each approach and their implications for ground-based robotic systems.

2.4.1 Advantages of MEMS-Based Algorithms

- **High Sensitivity:** MEMS sensors, due to their miniaturized nature, can detect minute changes in their environment, offering high-resolution data. This sensitivity translates to more accurate obstacle detection when paired with suitable algorithms.
- **Compactness:** The small form factor of MEMS devices allows for easy integration into robotic systems, facilitating more streamlined designs without compromising on detection capabilities.
- **Adaptability:** MEMS-based systems can be easily calibrated and adjusted, allowing them to adapt to various environments and conditions.

2.4.2 Challenges of MEMS-Based Algorithms

- **Environmental Susceptibility:** MEMS sensors can be sensitive to external factors like temperature, humidity, and physical shocks, which might affect their readings.
- **Complex Calibration:** While MEMS devices offer adaptability, they often require intricate calibration processes to ensure accuracy.
- **Cost Implications:** Advanced MEMS sensors and the algorithms tailored for them can sometimes be more expensive than traditional methods.

2.4.3 Traditional Methods: Strengths and Limitations

- **Proven Track Record:** Traditional obstacle detection methods have been in use for longer and have a proven track record in specific applications.
- **Simplicity:** Some traditional methods are simpler and easier to implement, especially in less complex robotic systems.
- **Limitations in Sensitivity:** Compared to MEMS-based systems, some traditional methods might not offer the same level of sensitivity and resolution.

2.4.4 Implications for Robotic Systems

- **Choice Based on Application:** While MEMS-based algorithms offer advanced capabilities, the choice between MEMS and traditional methods should be based on the specific application and requirements of the robotic system.

- **Integration Challenges:** Integrating MEMS-based algorithms into existing systems that rely on traditional methods can pose challenges and might require significant overhauls.
- **Future Trends:** With the rapid advancements in MEMS technology and the increasing complexity of robotic applications, it's likely that MEMS-based algorithms will gain even more prominence in the future.

2.5 Case Studies: Real-World Applications of MEMS-Based Obstacle Detection Algorithms

The theoretical and technical aspects of MEMS-based algorithms for obstacle detection are best understood when contextualized within real-world applications. This section delves into specific case studies that showcase the integration and efficacy of these algorithms in diverse environments.

2.5.1 Autonomous Vehicles

- Modern autonomous vehicles are equipped with a myriad of sensors to navigate complex urban environments safely.
- MEMS-based sensors, such as accelerometers and gyroscopes, are integrated into these vehicles to detect sudden changes in motion or orientation. Combined with advanced algorithms, these sensors help vehicles detect pedestrians, other vehicles, and obstacles in real-time, ensuring safe navigation.
- Enhanced safety, reduced accidents, and increased trust in autonomous vehicle technology.

2.5.2 Drones for Agricultural Monitoring

- Drones are increasingly being used for precision agriculture, monitoring crop health, and assessing field conditions.
- MEMS-based proximity sensors help drones detect and avoid obstacles like trees or power lines. Additionally, MEMS accelerometers ensure stable flight, even in windy conditions.

- Efficient and accurate agricultural monitoring, leading to optimized crop yields and reduced operational costs.

Fig. 2.4 Air-based drone

2.5.3 Industrial Robots in Manufacturing

- In modern manufacturing units, robots are employed for tasks ranging from assembly to quality control.
- MEMS sensors integrated into these robots detect nearby objects or workers, ensuring safe operations. Algorithms process this data in real-time to prevent collisions and ensure smooth workflow.
- Enhanced safety in industrial settings, increased productivity, and reduced downtime.

2.5.4 Search and Rescue Robots

- In disaster-stricken areas, robots are deployed for search and rescue operations where human intervention might be risky.
- These robots, equipped with MEMS-based sensors, can navigate through debris, detect survivors by sensing minute vibrations or sounds, and relay information back to rescue teams.
- Faster rescue operations, increased chances of finding survivors, and reduced risks for rescue personnel.

2.5.5 Healthcare: Surgical Robots

- Surgical robots are used in minimally invasive procedures, requiring precision and stability.
- MEMS sensors in these robots detect the slightest of movements, ensuring precision during surgeries. Algorithms process this data to guide the robot's movements accurately.
- Increased surgical accuracy, reduced recovery times for patients, and minimized surgical complications.

3. DEVELOPMENT OF AN OBSTACLE DETECTION SYSTEM FOR A GROUND ROBOTIC COMPLEX USING LIDAR TECHNOLOGY

Lidar (Light Detection And Ranging) is an abbreviation for light detection and ranging. Lidar is similar to radar except that it uses light instead of radio waves. The light source is a laser (Fig. 3.1):

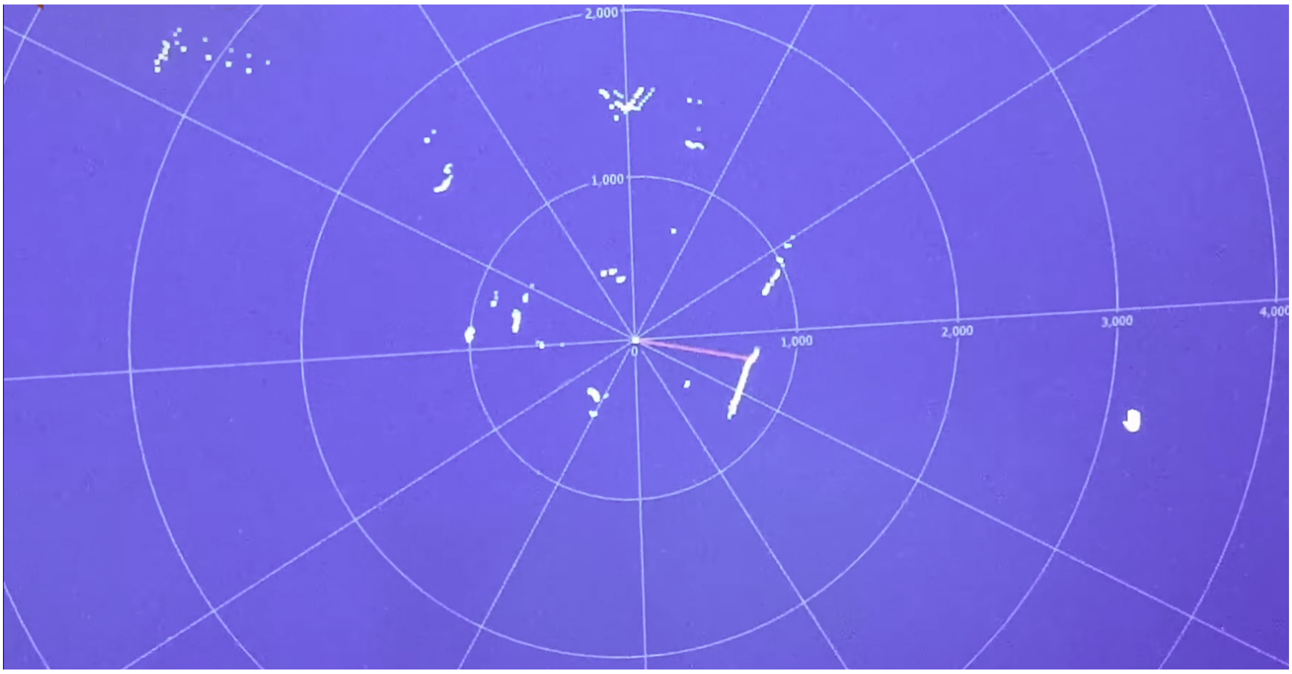


Fig. 3.1 Principle of operation of LIDAR

The lidar sends light pulses and measures the time it takes for the wave to bounce off a distant object and return to the device..

3.1 Calculation of the distance to the object using LIDAR

Since the speed of light is a known constant, the distance to the object can be calculated by the time of passage of the light pulse (Fig. 3.2):

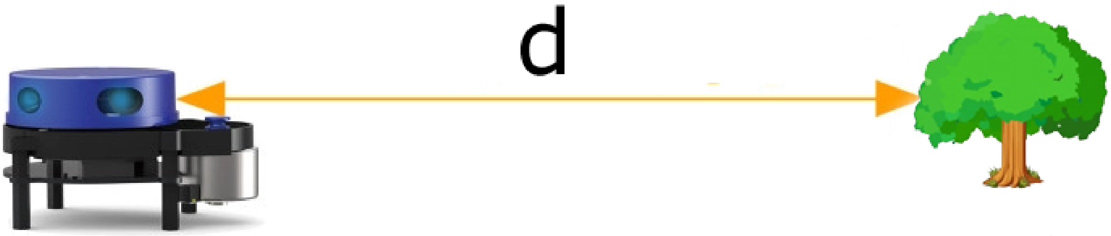


Fig. 3.2 Calculation of the distance from LIDAR to the object

The distance from the LIDAR to the object is calculated according to formula 3.1:

$$\boldsymbol{d = k \cdot t \cdot c,} \tag{3.1}$$

where: $\boldsymbol{t}$ – the transit time of the light pulse, s;

c - speed of light, m/s;

$\boldsymbol{k} = 0,5$ - coefficient.

The coefficient k corrects for the fact that the laser pulse must travel to the object and back. Accordingly, go twice the distance.

Lidar can be either one-dimensional, such as, for example, laser rangefinders, or two-dimensional, similar to the radars used on ships or in control towers at airports.

3D lidar also exists and is used, for example, by airplanes to create a three-dimensional image of the surface of the earth below them (Fig. 3.3):

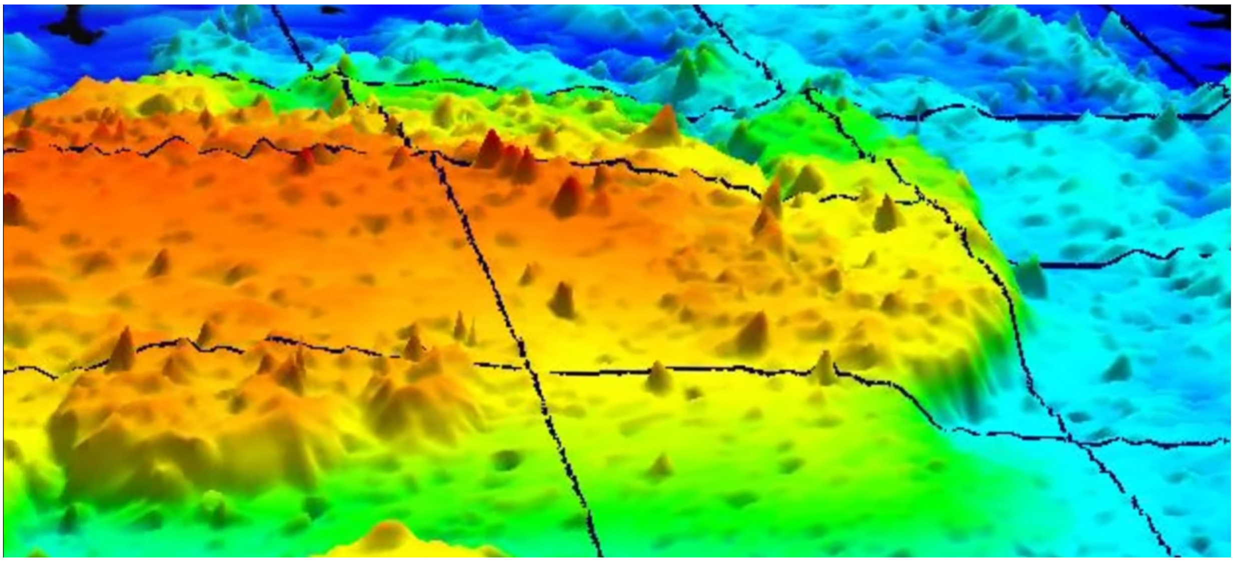


Fig. 3.3 Three-dimensional image of the earth's surface, made with the help of lidar

In the development of an obstacle detection system for the ground robotics complex, we will use lidar to scan X, Y and Z coordinates.

3.2 2D LIDAR background image

In essence, a two-dimensional lidar or 2D lidar is nothing more than a rotating 1D lidar. Instead of rotating the laser and detector, it is often easier to shine the laser onto a rotating mirror (Fig. 3.4):

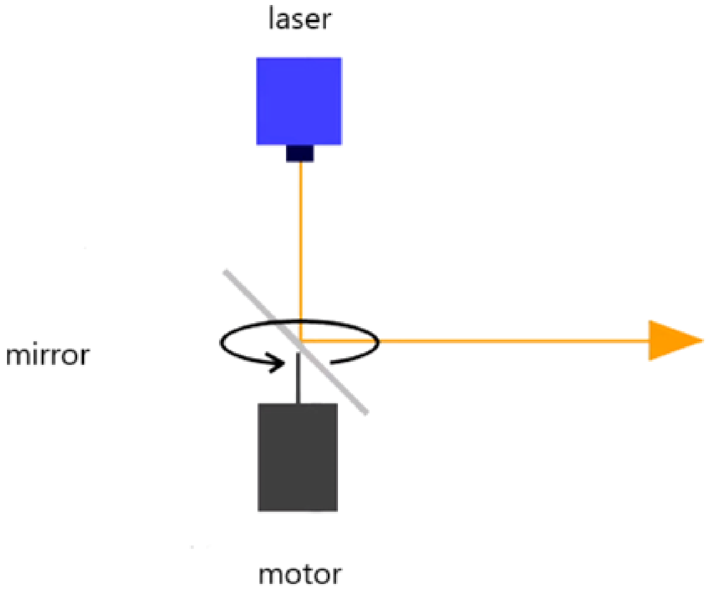


Fig. 3.4 Scheme of using MEMS lidar

By sending periodic pulses of light, it is possible to cover 360 degrees and create a distance map with the lidar in the center. At the same time, the reflectivity of the object is important. A perfect blackbody cannot be seen on lidar because it does not reflect light at all.

For development, we will use LIDAR-Lite 3 Laser Rangefinder (Fig. 3.5):

Fig. 3.5 Appearance of LIDAR-Lite 3 Laser Rangefinder

3.3 Description and main characteristics of LIDAR-Lite 3

This type of lidar has the following characteristics [1] (Table 3.1):

Table 3.1 Features of LIDAR-Lite 3

Parameter	Value
Sizes, mm	20x48x40 mm (0,8x1,9x1,6 inch)
Weight, g	22
Range of measurements, cm-m	5cm - 40 m
Accuracy at a distance of more than 1 m, cm	+/- 2,5
DC power supply, V	4.75-5
Current consumption in standby mode, mA	105
Current consumption during continuous operation, mA	130
Refresh rate, Hz	Up to 500
Interface	I2C or PWM

Laser wavelength/peak power, nm/W	905 /1,3
Beam divergence, milliradians	4x2
Aperture, mm	12,5
Operating temperature range, °C	-20 ... 60

The selected LIDAR is a highly efficient optical distance sensor:

- reliable and powerful range and proximity sensor for drones, robots or unmanned vehicles;
- compact, light with low power consumption;
- is user-configurable, allowing adjustment of accuracy, operating range, and measurement time;
- have communication through I2C and PWM interfaces;
- requires a power supply and an external microcontroller running the program.

When size and weight requirements are limited, the LIDAR-Lite v3 wins. It is the ideal compact, high performance optical remote sensing sensor solution for drones, robots or unmanned vehicles. By using a single-chip microcontroller for signal processing together with minimal hardware, this highly configurable sensor can be used as a basic unit for applications where small size, light weight, low power consumption and high performance are key criteria when designing an obstacle detection system for the ground robotics complex.

Featuring all the main features of the popular LIDAR-Lite v3, this easy-to-use 40m laser sensor consumes around 130mA during data acquisition. It is user-configurable, so you can choose between accuracy, operating range and measurement time [2].

You also need a power supply and an external microcontroller on which the program will run.

3.4 Description of the Arduino IDE platform

The Arduino Integrated Development Environment (IDE) is a cross-platform program (for Microsoft Windows, macOS, and Linux) written in the Java programming language. It is derived from IDEs for Processing and Wiring languages. It includes a code editor with features such as cut and paste text, find and replace text, auto-indent, brace matching, and syntax highlighting, and provides simple mechanisms for compiling and uploading programs to the Arduino platform. It also contains a message area, a text console, a toolbar with buttons for common functions, and a hierarchy of work menus. The source code for the IDE is released under the GNU General Public License, version 2 [3].

After the platform was completed, lighter and less expensive versions were distributed to the open source community. It is estimated that over 300,000 official Arduino boards were commercially produced in mid-2011 [4] and 700,000 official boards were in the hands of users in 2013 [5, 6].

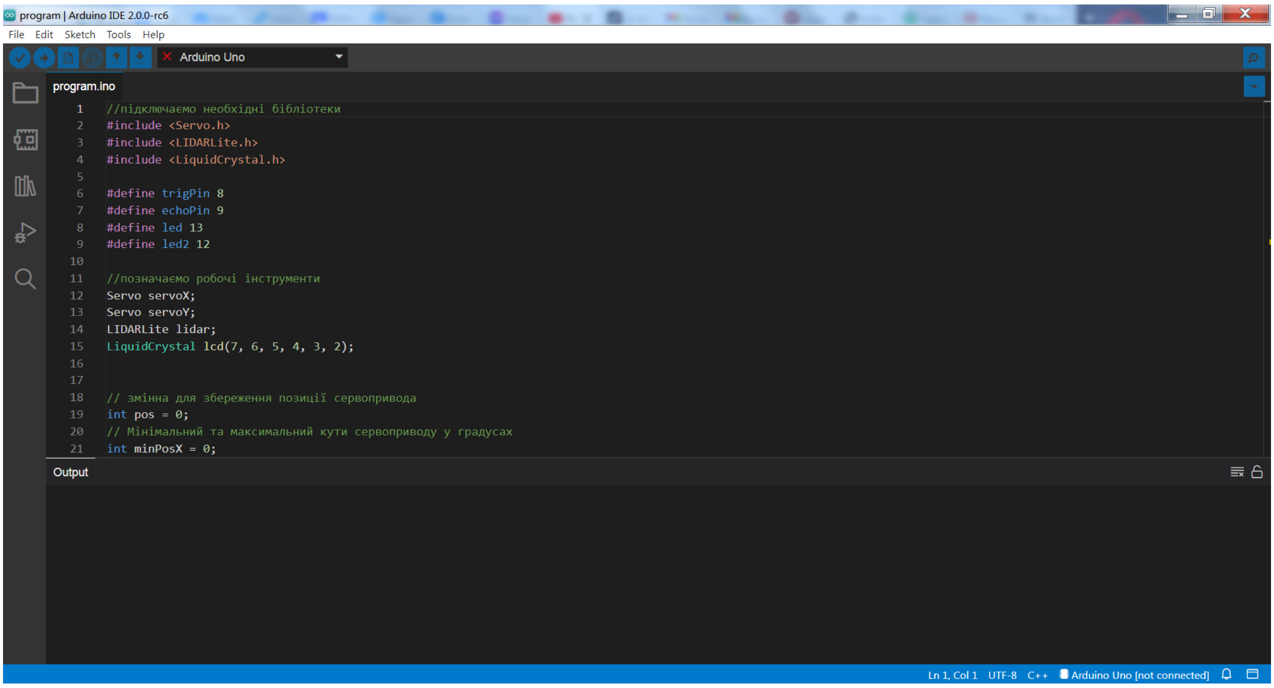
Arduino and Arduino-compatible platforms use printed circuit board expansion boards, called shields, that connect to standard Arduino pin connectors [7]. Shields can provide motor control for 3D printing and other applications, GNSS (navigation satellite), Ethernet, liquid crystal display (LCD), or mock-up (prototyping)[8].

There are many Arduino-compatible and Arduino-derived boards. Some of them are functionally equivalent to Arduino and can be used interchangeably. Most often, the basic Arduino is improved by adding output drivers [9, 10] to simplify the creation of systems and small robots. Some variants use different processors with different compatibility.

The Arduino Uno has a resettable fuse that protects the computer's USB ports from short circuits and overcurrents. While most computers provide their own internal protection, a fuse provides an extra layer of protection. If more than 500mA is applied to the USB port, the fuse automatically disconnects until the short circuit or overload is removed.

3.5 Writing code in the Arduino IDE environment

A .hex file must be obtained for successful simulation in the Proteus environment. To obtain it (compile), we will use the Arduino IDE environment (Fig. 3.6).

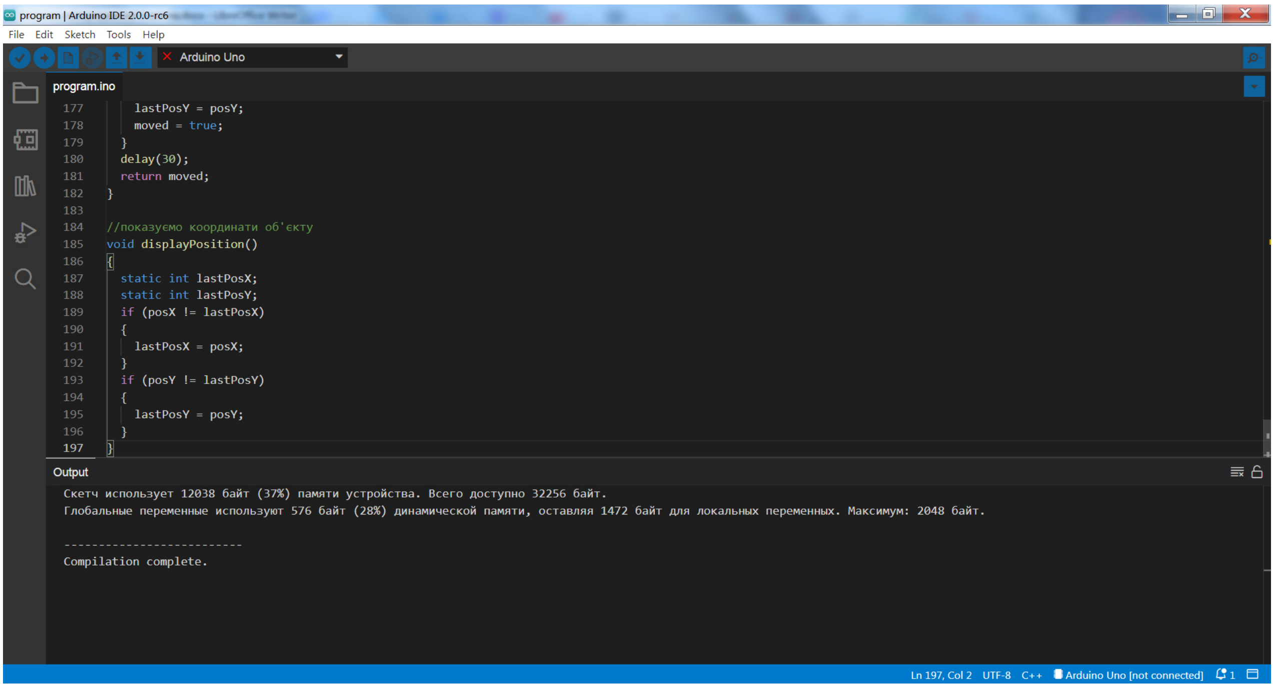


```
//підключаємо необхідні бібліотеки
#include <Servo.h>
#include <LIDARLite.h>
#include <LiquidCrystal.h>

#define trigPin 8
#define echoPin 9
#define led 13
#define led2 12

//позначаємо робочі інструменти
Servo servoX;
Servo servoY;
LIDARLite lidar;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);


// змінна для збереження позиції сервопривода
int pos = 0;
// Мінімальний та максимальний кути сервоприводу у градусах
int minPosX = 0;
```

Fig. 3.6 Arduino IDE environment

Next, we will sequentially write the code, the listing of which is presented in Appendix A.

After writing the code, compile it to obtain a .hex file (Fig. 3.7):



```
    lastPosY = posY;
    moved = true;
  }
  delay(30);
  return moved;
}

//показуємо координати об'єкту
void displayPosition()
{
  static int lastPosX;
  static int lastPosY;
  if (posX != lastPosX)
  {
    lastPosX = posX;
  }
  if (posY != lastPosY)
  {
    lastPosY = posY;
  }
}
```

```
Скетч использует 12038 байт (37%) памяти устройства. Всего доступно 32256 байт.
Глобальные переменные используют 576 байт (28%) динамической памяти, оставляя 1472 байт для локальных переменных. Максимум: 2048 байт.

--------------------------
Compilation complete.
```

Fig. 3.7 Writing and compiling source code

Conclusions:

So, to detect obstacles, we determine the distance to the nearest object in front of the lidar. The lidar sends light pulses and measures the time it takes for the wave to bounce off a distant object and return to the device.

The Arduino board sends a short pulse to initiate detection and then registers the pulse on the same pin using the pulseIn() function. The duration of this second pulse is equal to the time it takes the laser to travel to the object and back to the lidar. Using the speed of sound, we convert this time into distance.

Chapter 3 shows how LIDAR-Lite works and how it can be used in conjunction with an Arduino board. For our project, we are creating a system for fixing the distance to an obstacle.

In this section, the program code for the obstacle detection system is also written and compiled in the Arduino IDE environment.

4. IMPLEMENTATION OF THE OBSTACLE DETECTION SYSTEM FOR THE GROUND ROBOTIC COMPLEX AND TESTING RESULTS

4.1 Development of the scheme of the principle system of obstacle detection for the ground robotic complex

Development of the scheme of the principle system of obstacle detection for the ground robotic complex:

- Arduino Uno;

- лідар LIDAR-Lite 3;

- LCD display;

- variable resistance resistor;

- electrolytic capacitors 100 μF, 25 V;
- Power Supply;
- wires for connection.

As a result, we will get the developed scheme (Fig. 4.1):

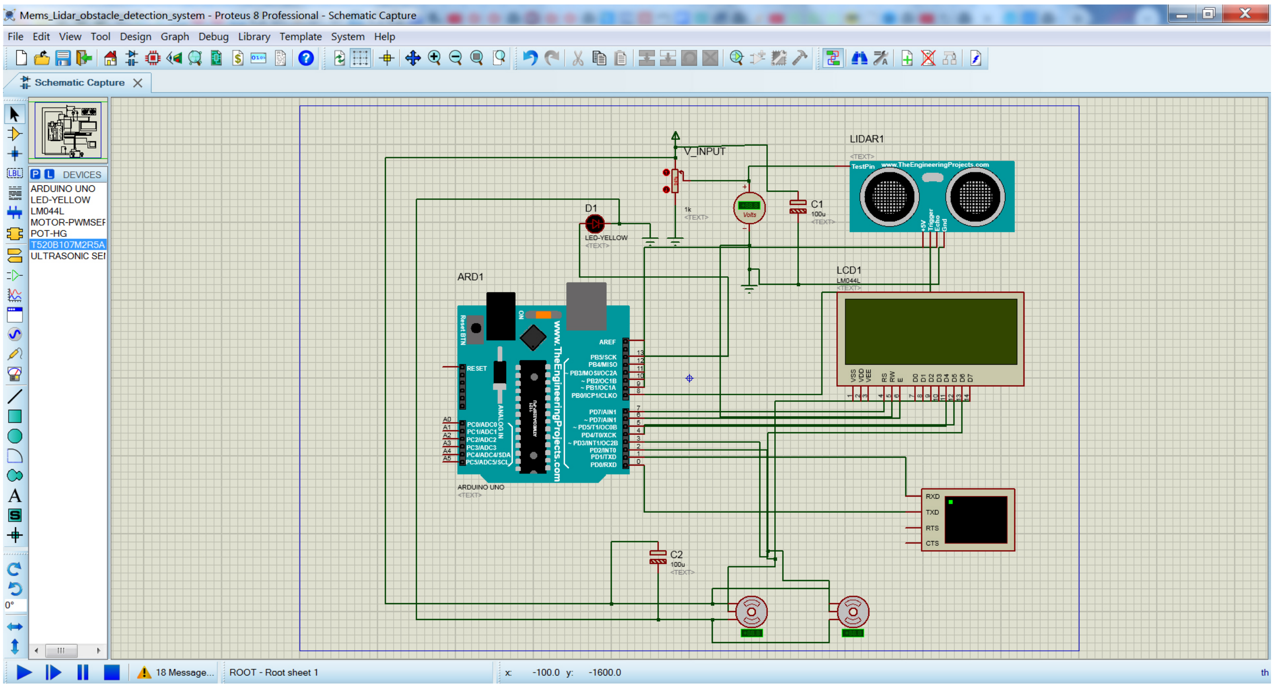


Fig. 4.1 Scheme of the obstacle detection system for the ground robotic complex

4.2 Ways and means of virtual modeling of electrical systems based on Arduino

Arduino is an open source electronic platform that can be used to create interactive projects. To develop a real device, you need a breadboard, electrical system components and an Arduino.

It is often necessary to experiment with the Arduino simulator before committing to real development. A good simulator should allow several aspects of the process to be reproduced digitally:

- create your own components and schemes or import from the library;
- create programs (sketches) in the Arduino IDE;
- simulate the interaction between the Arduino, the I/O interfaces and the program;
- export boards and circuits for the production of printed circuit boards [11].

Let's consider the currently existing means of virtual modeling of electrical systems based on Arduino.

4.2.1 Microsoft Maker Code

In Microsoft Maker Code (Fig. 4.2), creating simulations (simulations) using various boards, including Arduino models, is done using visual blocks, which makes it accessible even to those who have no prior programming experience. You can also choose programming in Python or JavaScript. Everything in this environment is very intuitive.

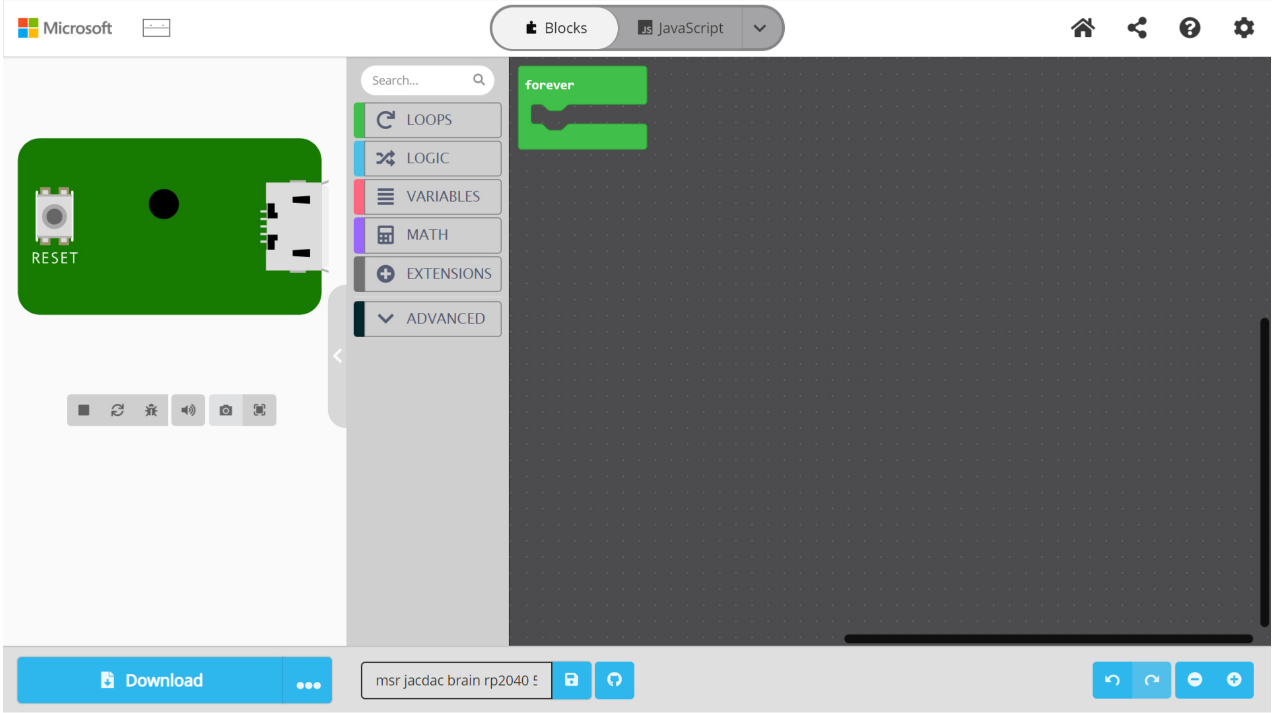


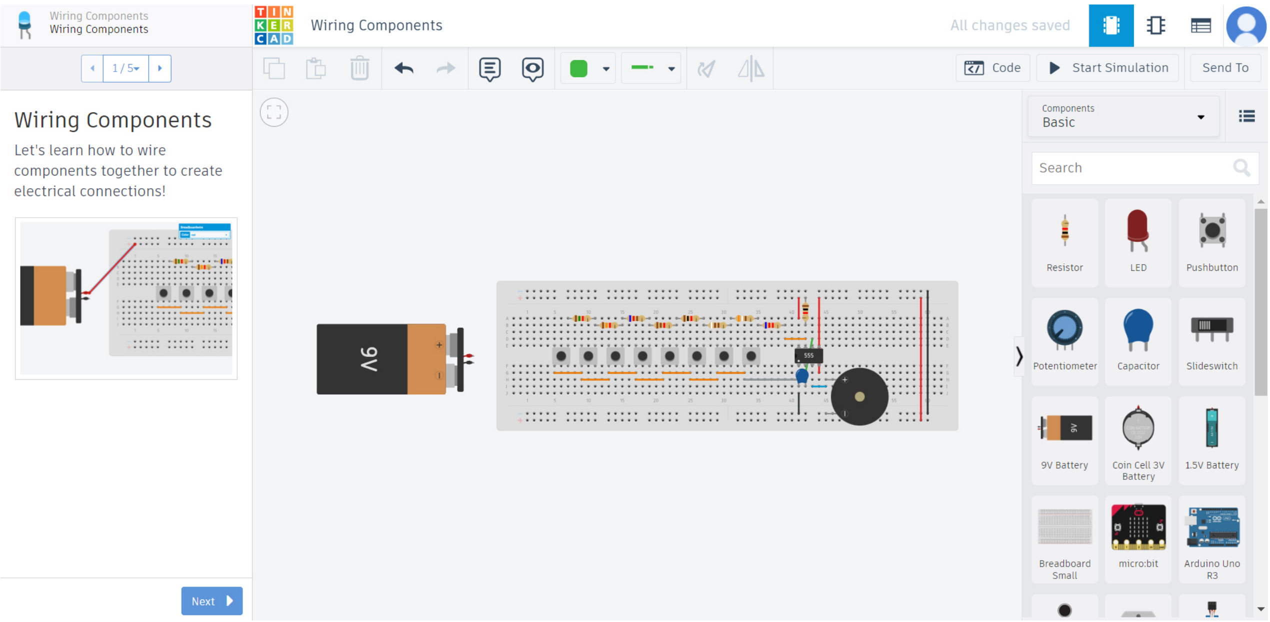
Fig. 4.2 Microsoft Maker Code virtual simulation tool

Although it is an online platform, physical devices can also be connected. There is access to extensions for sensors and other components that are great simulation aids. In addition to the most basic functions, there are more powerful extensions that can be found further down the panel to provide additional functions such as a joystick, sensors or even commands to support USB flash drive and flash memory.

The user interface is quite simplistic, displaying an illustration panel with animations on the left and a block programming panel on the right. Some output commands display results even without an Arduino or component such as sound connected [12].

4.2.2 Tinkercad Circuits

Tinkercad Circuits is a free online service from Autodesk (Fig. 6.3), which was launched in 2017 and is a convenient Arduino simulator where you can easily design your own circuits, create a program in block or text format, and then debug it.



Wiring Components

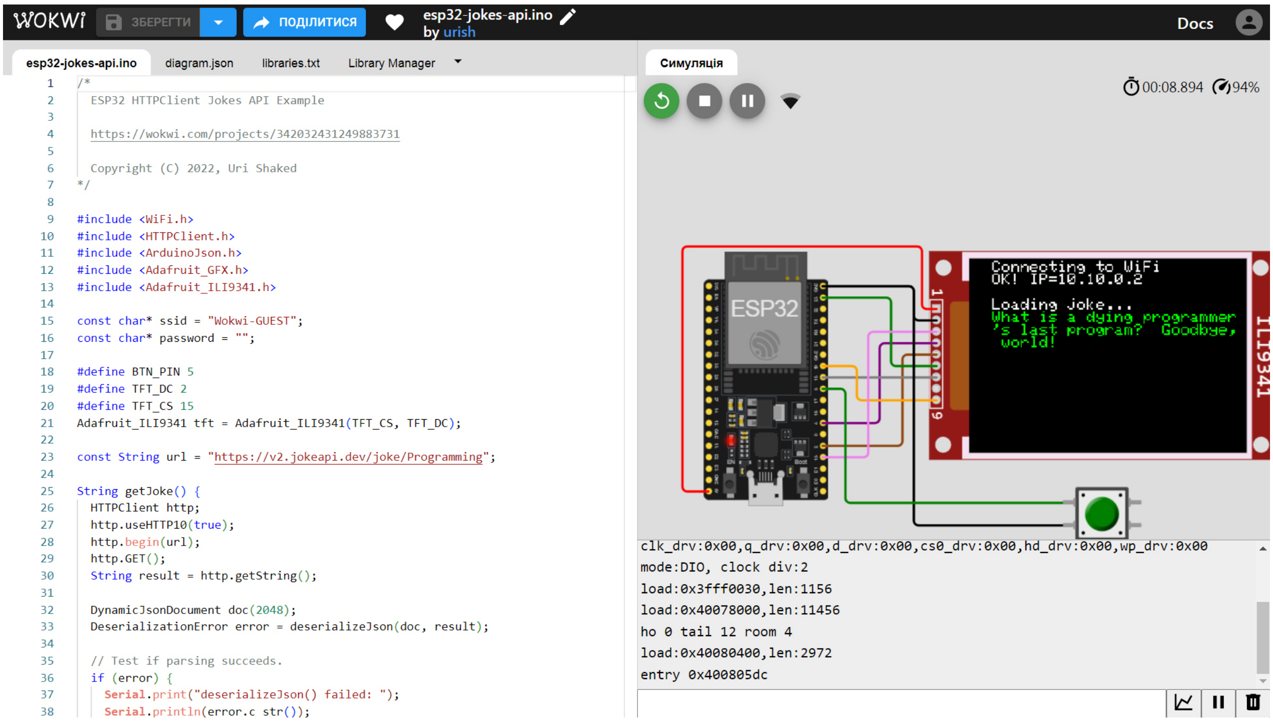
Let's learn how to wire components together to create electrical connections!

Fig. 4.3 Tinkercad Circuits virtual modeling tool

Tinkercad allows you to use any element from its library, but does not allow you to add new component options (modules, different board models, sensors) to the library. Some basic components such as resistors can be parameterized, but this is not an option for microcontrollers, and for Arduino there is only the Arduino Uno R3 model [13].

4.2.3 Wokwi

Wokwi is based on AVR8js, a JavaScript implementation of the 8-bit AVR architecture (Figure 4.4). Here you can explore and simulate the examples, as well as modify the sketch and appearance of the diagram using the diagram.json file.



```
/*
  ESP32 HTTPClient Jokes API Example

  https://wokwi.com/projects/342032431249883731

  Copyright (C) 2022, Uri Shaked
*/

#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>
#include <Adafruit_GFX.h>
#include <Adafruit_ILI9341.h>

const char* ssid = "Wokwi-GUEST";
const char* password = "";

#define BTN_PIN 5
#define TFT_DC 2
#define TFT_CS 15
Adafruit_ILI9341 tft = Adafruit_ILI9341(TFT_CS, TFT_DC);

const String url = "https://v2.jokeapi.dev/joke/Programming";

String getJoke() {
  HTTPClient http;
  http.useHTTP10(true);
  http.begin(url);
  http.GET();
  String result = http.getString();

  DynamicJsonDocument doc(2048);
  DeserializationError error = deserializeJson(doc, result);

  // Test if parsing succeeds.
  if (error) {
    Serial.print("deserializeJson() failed: ");
    Serial.println(error.c str());
```

```
Connecting to WiFi
OK! IP=10.10.0.2

Loading joke...
What is a dying programmer
's last program?  Goodbye,
 world!
```

```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:2
load:0x3fff0030,len:1156
load:0x40078000,len:11456
ho 0 tail 12 room 4
load:0x40080400,len:2972
entry 0x400805dc
```

Fig. 4.4 Wokwi virtual simulation tool

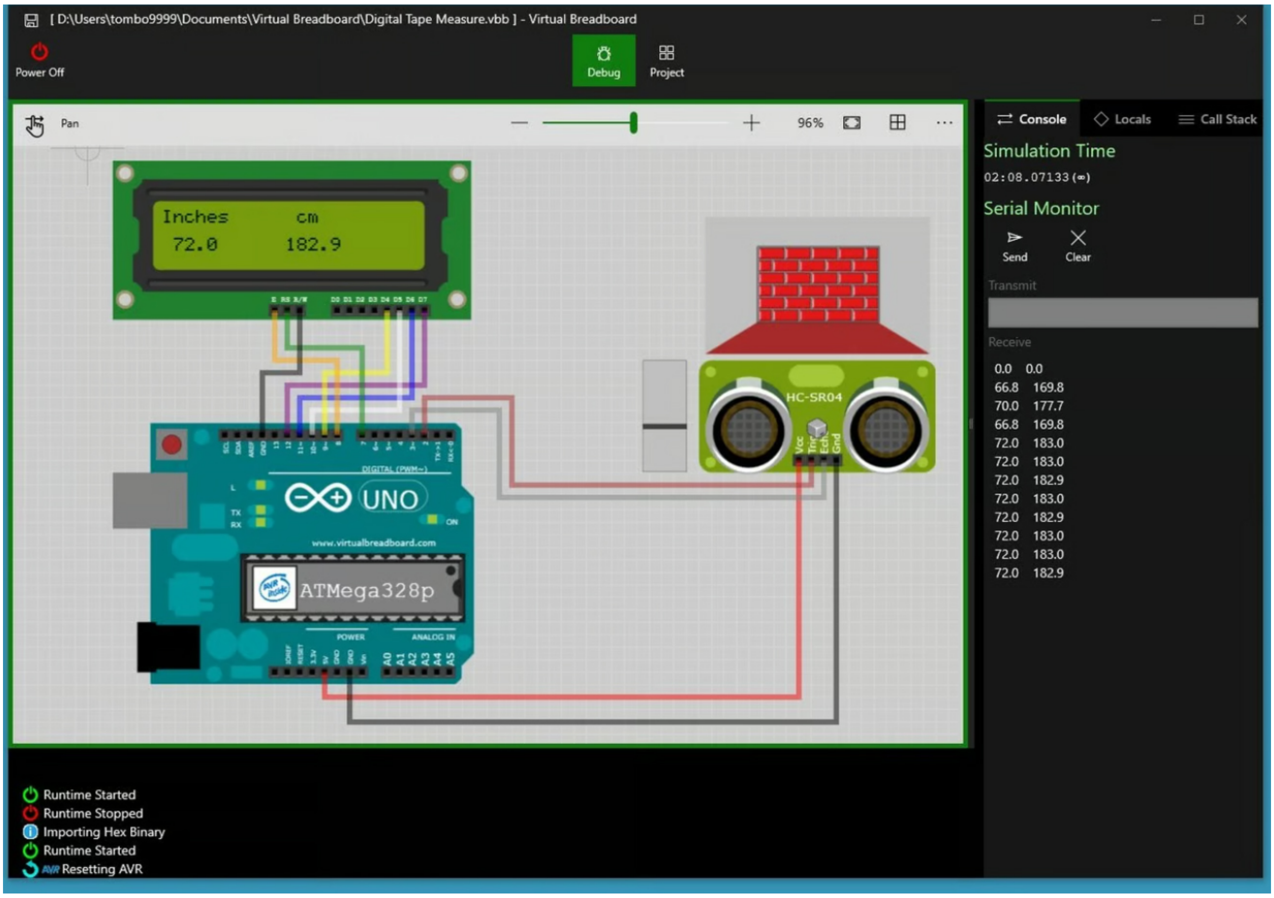
This is not a drag-and-drop interface, so you need to look at existing examples, copy them, modify them, and test the results yourself. After doing this, you can create your own simulation. To create your own diagram, you need to modify the diagram.json file.

Advanced users can create or add their own parts and components, as well as add Arduino libraries. The only limitation is that you cannot export boards and circuits for PCB manufacturing.

The developer and community are very active and Wokwi has grown rapidly. In the Discord channel, you can ask questions and get support, including from the developer [14].

4.2.4 Virtual breadboard

The virtual layout is a program from the Windows application (Fig. 4.5). The interface is user-friendly and you can easily create your own schemes using the drag-and-drop tools. You can also upload your sketches in HEX format from Arduino IDE, Arduino Create, PlatformIO, Visual Studio, etc.



```
0.0   0.0
66.8  169.8
70.0  177.7
66.8  169.8
72.0  183.0
72.0  183.0
72.0  182.9
72.0  183.0
72.0  182.9
72.0  183.0
72.0  183.0
72.0  182.9
```

```
Runtime Started
Runtime Stopped
Importing Hex Binary
Runtime Started
AVR Resetting AVR
```

Fig. 4.5 Virtual breadboard virtual simulation tool

Its scope of functionality includes modeling the Arduino board, I/O interfaces, and program interactions. A special feature of Virtual Breadboard is the inclusion of virtual mixed reality hardware and the ability to modify the hardware.

The simulator is somewhat limited in that you cannot create or add your own parts and components, or create your own programs from the software interface [15].

4.2.5 FlowCode

Flowcode - a graphical programming platform with an integrated IDE for various controllers such as PIC, AVR, ARM, ESP or Raspberry Pi (Fig. 4.6). In addition to easy

Arduino simulation, it offers a wide range of test components and 3D or 2D visualization of the environment. With simple click and drag, editable command blocks can be included in the block diagram to test the Arduino.

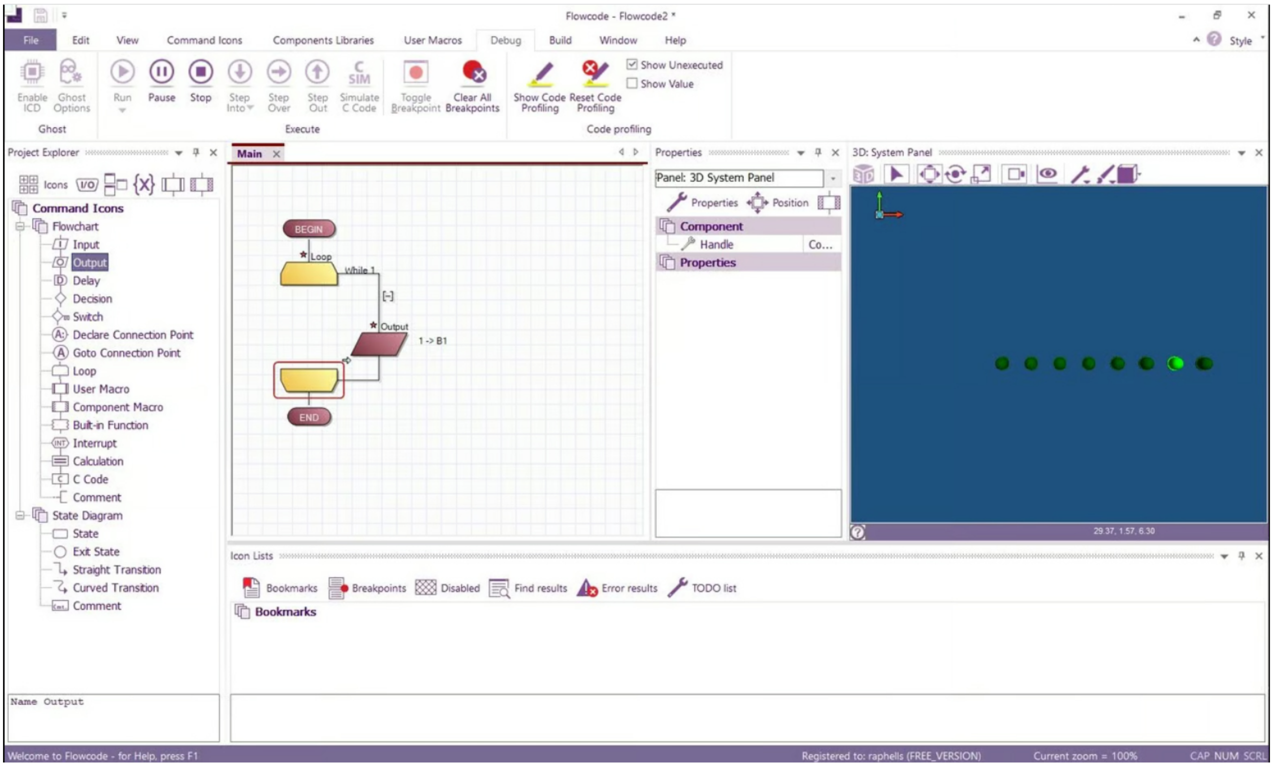


Fig. 4.6 FlowCode virtual simulation tool

Flowcode is quite intuitive and suitable for both beginners and experienced users. It has the unique ability to import 3D models in MESH, STEP and OBJ file formats, making it ideal for experiments with electromechanical systems where motors, servos and actuators can be integrated into a digital representation [16].

4.2.6 Virtronics

Virtronics is a simulator that does not offer many virtual components for assembling circuit designs from wires, modules and symbolic parts (Fig. 4.7). Instead, you can do a test run of how an Arduino board or model might behave in a certain situation in a more technical and modern way.

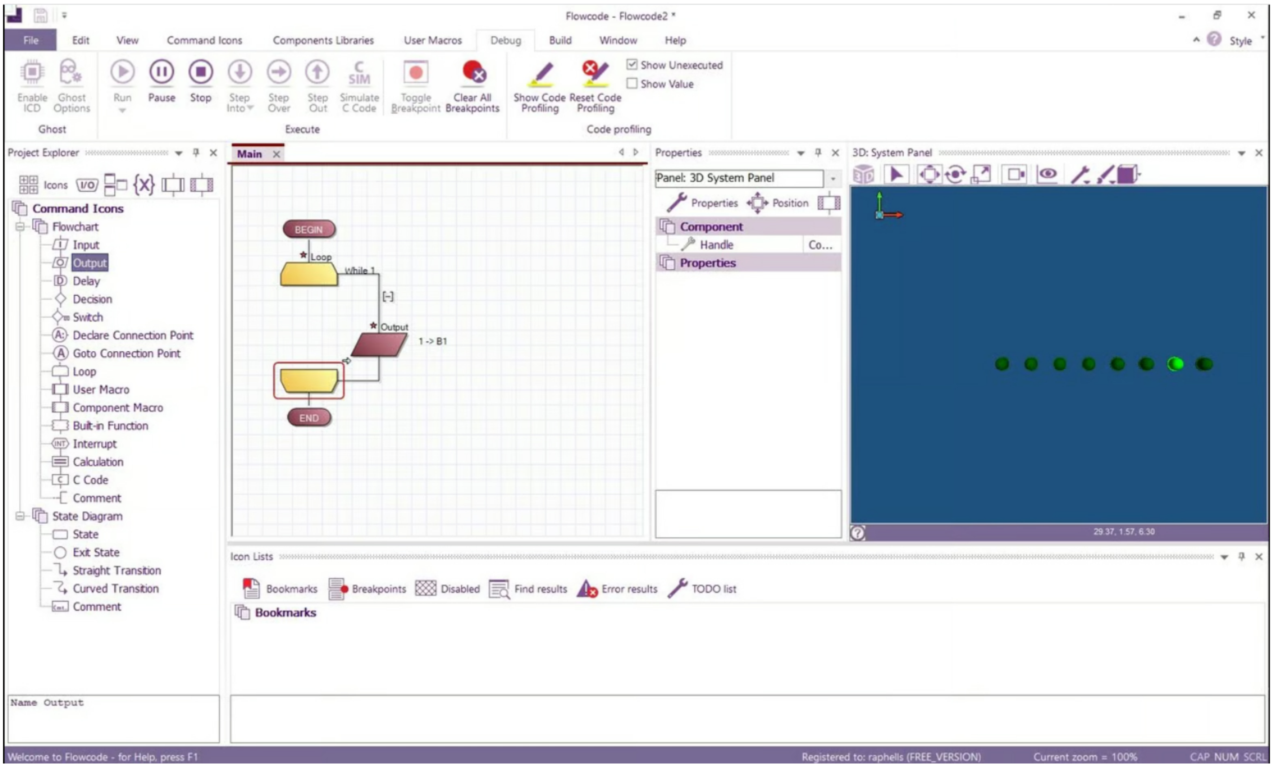


Fig. 4.7 Virtronics virtual simulation tool

Inputs and outputs can be manipulated to simulate circuit components and behavior, and monitors can be used to observe potential errors and understand why they occur. There is also a serial monitor and many other tools such as ASCII table, barcode generator, calculator, digital pin logic analyzer, USB, Ethernet and 2.1mm jack etc.

The interface displays variables in real-time, and the View menu provides other options such as SubRoutines, EEPROM, Class Explorer, and more. All of this can be used with 10 different board models: Uno, Due, Papilio, Esplora, Leonardo, Lily Pad, Arduino Mega, Arduino Nano, Uno32 and Yun. Other hardware such as terminals and LCD monitors can also be tested and simulated.

All of these features allow you to both test code before purchasing hardware, as well as debug, demonstrate and teach how the board works, and are a great application for sketching out more complex code and testing faster and easier than using the physical board itself [17].

4.2.7 Proteus VSM

The Proteus VSM for Arduino AVR is by far the best option on the list (Figure 4.8). With that in mind, in addition to all the features and capabilities listed above, it offers some special things that other simulators don't. For example, there is access to thousands of peripheral models, as well as dragging dozens of components into your simulations [18].

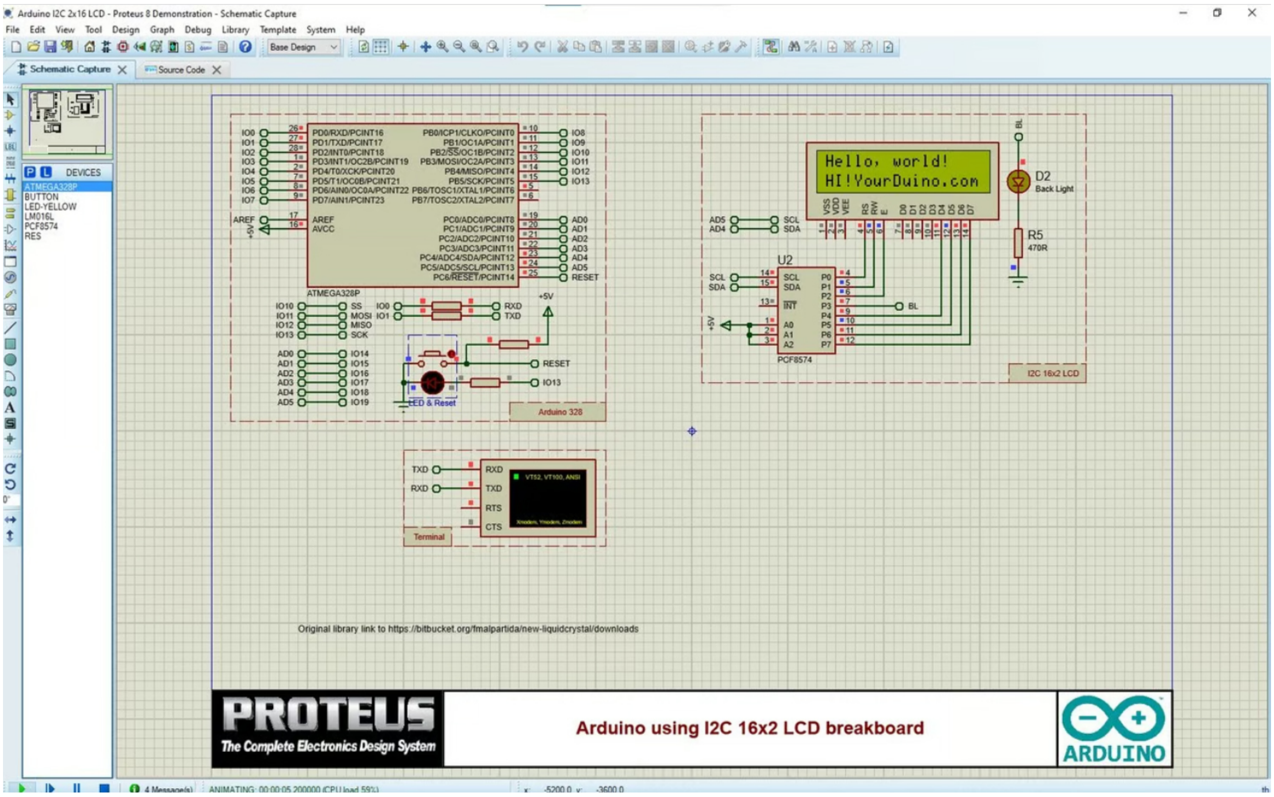


Fig. 4.8 Proteus VSM virtual simulation tool

4.3 Modeling the obstacle detection system in the Proteus VSM environment

4.3.1 System Structure and Connection Diagram

The obstacle detection system is based on the Arduino Uno as the main controller, to which the LIDAR-Lite 3 is connected to determine distances, and an LCD display for visual display of data. Additionally, a power supply and connecting wires are used.

All system components are connected to the Arduino Uno: LIDAR-Lite 3 is connected via I2C interface, LCD display via GPIO ports. Stable power supply to all components is provided.

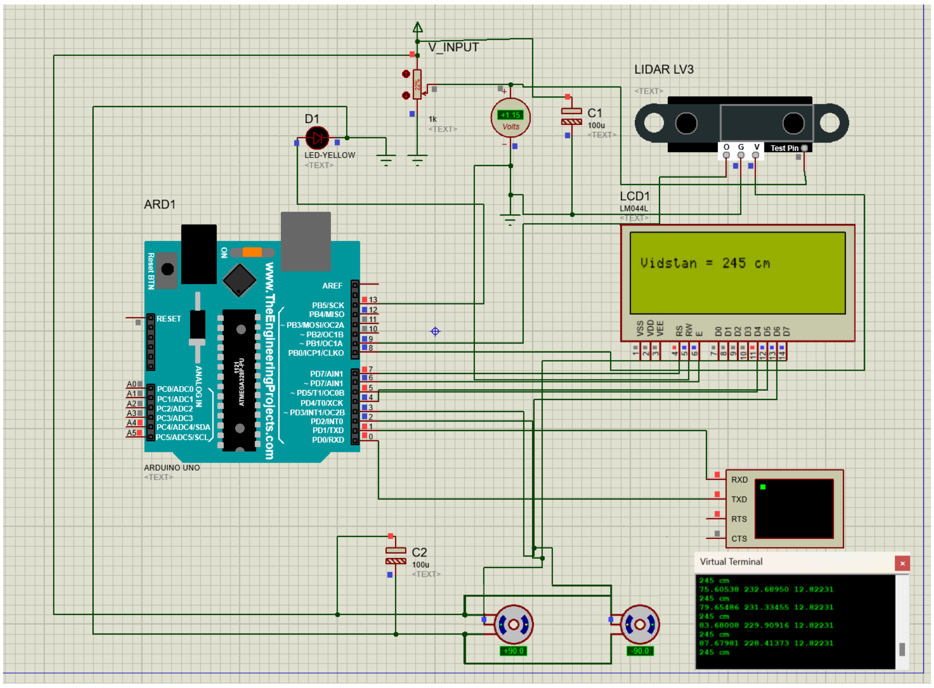
4.3.2. Proteus Simulation

Let's open the Proteus VSM environment and simulate the obstacle detection system to measure the distance to the obstacle. The distance in centimeters is calculated, the numerical value is displayed on the LCD display, as well as on the virtual terminal [19].

To begin with, by double-clicking on ARDUINO NANO V3, we will see the "Edit component" window, after which we will specify the path to the file program.ino.hex, which was compiled earlier.

The Play, Stop, Pause buttons are located at the bottom of the Proteus window screen. Pressing the Play button will start the simulation (Fig. 4.9). Remember that the Arduino IDE will not compile the code if Simulation is running and will not show any error [20-24].

Figure 4.9 Modeling of the obstacle detection system for the ground robotics complex



```
Vidstan = 245 cm
```

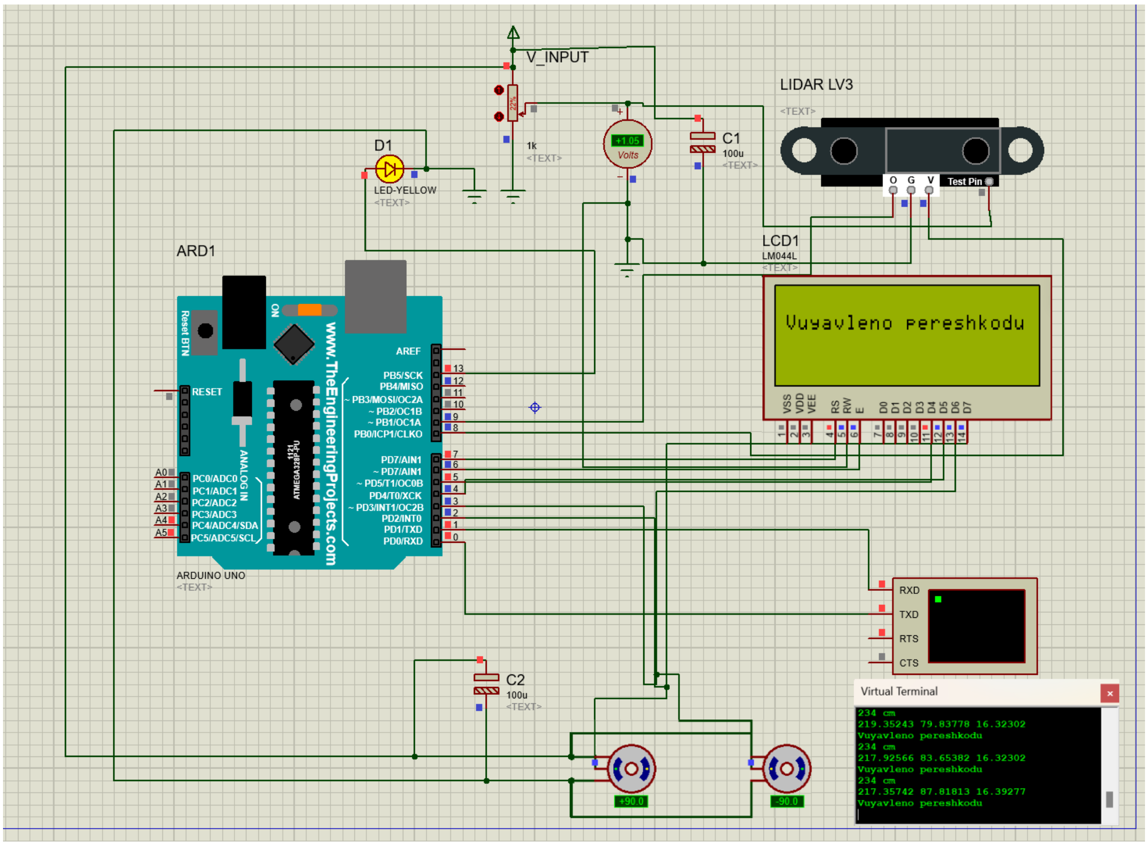
```
245 cm
75.60538 232.68950 12.82231
245 cm
79.65486 231.33455 12.82231
245 cm
83.68008 229.90916 12.82231
245 cm
87.67981 228.41373 12.82231
245 cm
```

Fig. 4.9 Modeling the system when work

As you can see, the display shows the value of the distance to the detected object in centimeters, and the terminal shows the distance in cm and the result of scanning the X, Y and Z coordinates.

Since distance cannot be physically measured in the simulation environment, a variable resistance resistor must be used.

We measure the distance between the lidar and the object, then display the measured distance on the serial monitor, and then perform the test at this distance if it is less than or equal to 240 cm. Change the resistor parameters. This means that the object is in range and the LED is on, or if the distance is more than 240 cm, the LED turns off alternately, as shown in Fig. 4.10:



```
Vuyavleno pereshkodu
```

```
234 cm
219.35243 79.83778 16.32302
Vuyavleno pereshkodu
234 cm
217.92566 83.65382 16.32302
Vuyavleno pereshkodu
234 cm
217.35742 87.81813 16.39277
Vuyavleno pereshkodu
```

Fig. 4.10 Modeling the system when an obstacle is detected

As you can see, the display first shows the value of 211 cm, and then the message "Obstacle detected".

4.3.3. Experimental Testing and Error Determination

Testing was conducted in real-world conditions to determine the system's efficiency and accuracy. Measurements were made at different distances, with different types of objects, and in different lighting conditions.

- Maximum conditions: Distance of 40 meters, daylight.
- Minimum conditions: Distance of 5 cm, dark environment.

During the test, conditional and absolute measurement errors were determined.

- Conditional Error: Variation in measurement accuracy with distance.
- Absolute Error: +/- 2.3 cm at a distance of more than 1 m.

4.3.4. Comparison with other types of sensors

To determine the average error per 1 meter for each sensor, you need to take into account the errors for hard, soft, and moving objects. The average error value can be calculated by averaging these three values. Here is an updated table with the revised average error value:

Parameter	LIDAR-Lite 3	Ultrasonic Sensor	Radios Sensor
Max distance	40 m	4 m	10 m
Min distance	5 cm	2 cm	5 cm
Accuracy	High	Middle	High
Impact Lighting	Minimum	Zero	Minimum
Impact of the Objects	Low	High	Middle
Average Error per 1 m	+/- 2.3 cm	+/- 2.7 cm	+/- 4.7 cm
Error on solid objects	+/- 1.5 cm	+/- 1 cm	+/- 3 cm
Error on Soft Objects	+/- 2.5 cm	+/- 3 cm	+/- 4 cm
Error on moving objects	+/- 3 cm	+/- 4 cm	+/- 6 cm

Calculating the Average Error:

- LIDAR-Lite 3: (1.5 cm + 2.5 cm + 3 cm) / 3 = 2.3 cm
- Ultrasonic Sensor: (1 cm + 3 cm + 4 cm) / 3 = 2.7 cm
- Radios Sensor: (3 cm + 4 cm + 6 cm) / 3 = 4.7 cm

To determine how much more accurate and long-range the LIDAR-Lite 3 is than the ultrasonic and radio sensors (in percentage terms), we can compare the maximum measurement distance and average error for each sensor. Below is a table with these percentage comparisons:

Parameter	LIDAR-Lite 3	Ultrasonic Sensor	Radios Sensor	**LIDAR vs Ultrasonic**	**LIDAR vs Radio**
Max distance	40 m	4 m	10 m	+900%	+300%
Average Error per 1 m	+/- 2.3 cm	+/- 2.7 cm	+/- 4.7 cm	+14.8%	+51.1%

Range (Maximum Distance):

- LIDAR-Lite 3 has a 900% longer range than an ultrasonic sensor (40 m vs. 4 m).
- LIDAR-Lite 3 has a 300% longer range than a radio sensor (40 m vs. 10 m).

Accuracy (Average Error per 1 m):

- LIDAR-Lite 3 is 14.8% more accurate than an ultrasonic sensor (2.3 cm vs. 2.7 cm).
- - LIDAR-Lite 3 is 51.1% more accurate than a radio sensor (2.3 cm vs. 4.7 cm).

Conclusions:

In Chapter 4, a schematic diagram is developed, analysis is performed, and the methods and means of virtual modeling of electrical systems based on Arduino are substantiated. Modeling of the obstacle detection system for the ground robotic complex when detecting an obstacle in the Proteus VSM environment was performed.

Summing up, the concept of an obstacle detection system for a ground robotic complex has been developed. This autonomous system uses low-cost lidar to detect obstacles and avoid objects.

There are many prospects for improving this solution, e.g:

- minimization of the error in determining distances to objects and as a result of the movement of the ground robotic complex in the direction of the greatest distance;

- writing an application for controlling a robotic complex and displaying a 3D landscape graph with detected obstacles;

- sending a push reminder to the operator's smartphone in case of critical obstacles, etc.

LIDAR-Lite 3 significantly outperforms ultrasonic and radio sensors in terms of measurement range and accuracy. The average error per 1 meter for LIDAR-Lite 3 is +/- 2.3 cm, which shows its high accuracy on different types of objects. The ultrasonic sensor has an average error of +/- 2.7 cm, which indicates its relatively high accuracy, but only on hard, stationary objects. The radio sensor, with the highest average error of +/- 4.7 cm, may be less accurate but is useful in scenarios where greater distances are required.

5. ENVIRONMENTAL PROTECTION

5.1: An Overview of Life Cycle Assessment (LCA)

Life Cycle Assessment (LCA) is a fundamental approach to analyzing the environmental impact of a product or system from its inception to the end of its life cycle. This process covers everything from raw material extraction and production to use, maintenance and, ultimately, recycling or disposal. LCA serves as a key tool in determining the environmental footprint of products and helps to make informed decisions to reduce their environmental impact.

The main goal of LCA is to provide a comprehensive view of the environmental aspects associated with a product or system. It takes into account various elements such as resource depletion, energy use, air, water and soil emissions, and waste generation. This method contributes to a deeper understanding of the environmental impacts at each stage of the product life cycle.

Key Elements of a Life Cycle Assessment

Defining the Purpose and Scope of the Study: This involves defining the assessment objectives and establishing the study parameters that cover the system boundaries, unit of operation, and impact categories to be considered.

Inventory Analysis: This step identifies and quantifies all inputs (e.g., materials, energy) and outputs (e.g., emissions, waste) at each stage of the life cycle. This involves compiling a detailed inventory of the processes involved.

Impact Assessment: This stage assesses the potential environmental impacts of the identified inputs and outputs. The analysis includes examining effects on categories such as global warming, acidification, eutrophication, and resource depletion.

Analysis and Interpretation: The final step involves analyzing the results to draw conclusions and recommendations based on the environmental impacts identified. It includes weighing trade-offs and identifying areas for improvement.

Benefits of Life Cycle Assessment (LCA)

Product Management / Research and Development:

Compliance with Regulatory Requirements: LCA is essential for product managers and R&D teams to ensure that their products meet environmental regulations and standards. Understanding the environmental impacts of products throughout their life cycle allows for decisions to be made that meet legal requirements and prevent legal complications.

Creating a Sustainable Product:

Developing a Sustainable Product: LCA is key in developing a sustainable, innovative product. It helps to identify ways to improve existing products or develop new ones with a lower environmental impact, meeting the growing consumer demand for eco-friendly options and improving the company's image.

Supply Chain and Procurement Management:

Supplier Selection: LCA helps in the evaluation and selection of suppliers based on their environmental impact. It allows supply chain and procurement professionals to evaluate the sustainability of raw materials and components, ensuring that the supply chain is aligned with the company's environmental goals.

Communication and Marketing:

Environmental Marketing and Reporting: An LCA provides valuable information for marketing communications, helping to shape messages about sustainability and environmental responsibility. This can include information for sustainability reports, eco-labeling, and stakeholder communications.

Risk and Reputation Management:

Reducing Risks and Improving Reputation: The use of LCA can help to avoid risks associated with environmental change and its impacts, as well as to improve the overall corporate reputation through a commitment to sustainability.

Practical Application of Life Cycle Assessment

Product development:

LCA is used in the development of products to determine their overall environmental impact. This may include an analysis of materials, production processes, packaging, transportation, use and disposal.

Policy and Strategy:

LCA provides information for the development of environmental policies and strategies, including reducing the environmental impact of products, optimizing supply chains, and improving resource efficiency.

Regulatory Requirements and Legislation:

LCA can be used to comply with legal requirements by ensuring that products meet environmental standards and regulations.

Methodology and Challenges of Life Cycle Assessment

Standards and Guidelines:

LCA is performed in accordance with international standards such as ISO 14040 and ISO 14044, which define the principles and structure of LCA.

Complexity and Scope:

Performing an LCA can be challenging due to the large amount of data required for analysis and the need for accurate data collection and processing.

Interpretation of Results:

Interpreting the results of an LCA requires a thorough understanding of environmental processes and their impacts, as well as the ability to weigh and compare different types of environmental impacts.

Life cycle assessment is an important tool for understanding and managing the environmental impact of products and systems. Its application contributes to sustainable development, efficient resource management, and environmental responsibility at the level of companies and organizations. With its ability to identify and analyze environmental impacts at different stages of the life cycle, LCA provides a reliable basis for making informed, responsible and effective decisions aimed at reducing environmental impact.

5.2. Equipment Description: Arduino-Based Altimeter with LIDAR-Lite 3

The Arduino-Based Altimeter, incorporating LIDAR-Lite 3, is an advanced instrument engineered to measure altitude or elevation above a predetermined reference point. This device leverages the precision of LIDAR technology, an Arduino Uno microcontroller, and various other electronic components to provide accurate altitude readings.

LIDAR-Lite 3 Sensor

The LIDAR-Lite 3 sensor is integral to the altimeter, using light detection and ranging (LIDAR) technology to measure altitude with high precision.

Materials: Constructed from robust materials compatible with electronic components, including plastic for the casing and glass or transparent polymers for the optical components.

Arduino Uno Microcontroller

Acts as the central processing unit of the altimeter. It receives data from the LIDAR-Lite 3 sensor, processes it, and delivers accurate altitude readings.

Materials: Built on a PCB with a microchip made of semiconductor materials like silicon. The board also includes metal connections and various electronic components like resistors and capacitors.

Printed Circuit Board (PCB)

The Arduino Uno's PCB serves as the foundation, connecting and supporting all electronic components, including the LIDAR-Lite 3, LCD display, and other peripherals.

Materials: Made using a substrate such as fiberglass-reinforced epoxy, with conductive copper pathways. Additional components like electrolytic capacitors (100 μF, 25 V) are also mounted on the board.

LCD Display

Provides a user-friendly interface, displaying altitude information in a readable format.

Materials: Typically uses glass for the display panel, with a mixture of organic compounds for the actual display technology. The frame and casing around the display can be made from plastics or metal alloys.

Housing/Enclosure

Protects the internal components from external factors like moisture, dust, and impact.

Materials: Usually made of durable materials like ABS or polycarbonate plastics, or lightweight metals such as aluminum.

Buttons/Controls

Includes input controls like buttons or switches for navigating settings and calibrating the device.

Materials: Often made of plastics or elastomers, with internal metal contacts. A variable resistance resistor is included for adjusting settings.

Connectivity

Function: The Arduino Uno allows for connectivity options such as USB, enabling programming, data transfer, and firmware updates.

Materials: USB connectors are typically made from metals like nickel-plated brass. The Arduino also includes headers and sockets for wire connections.

Memory

Function: The Arduino Uno includes onboard memory for code storage and can be used to store altitude data or user preferences temporarily.

Materials: The memory is an integrated circuit on the PCB, typically made of silicon. Protective casing may include plastic or metal.

Altitude Reference System

Function: The system may include an altitude reference feature, allowing users to set a specific reference point or adjust for variations, enhancing measurement accuracy.

Materials: Made using materials like silicon and durable polymers for the sensor components and adjustment mechanisms.

This chapter describes the structure and materials of an Arduino-based altimeter using the LIDAR-Lite 3 sensor, focusing on the functionality and composition of each key component. The integration of these components creates a versatile and accurate altimeter system suitable for various applications.

5.3. Description of the environmental impact of each stage of the life cycle for each piece of equipment

In the journey of an Arduino and LIDAR-Lite 3 Altimeter from concept to disposal, each stage of its life cycle carries distinct environmental implications. By understanding and addressing these impacts, we can steer this technological marvel towards a more sustainable path.

5.3.1 Environmental impact: LIDAR-Lite 3 Sensor

Extraction of raw materials: The production of the LIDAR-Lite 3 begins with the extraction of raw materials such as silicone for optical components and metals for electronic parts. This process can have a negative impact on the environment due to mining, which is often accompanied by CO2 emissions, water pollution, and landscape alteration.

Manufacturing and processing: The process of manufacturing a sensor involves the use of energy and production materials, which can lead to greenhouse gas emissions. Using energy-efficient technologies and resources that can be reused or recycled can reduce this impact.

Transportation: The transportation of components and finished sensors is another important factor that affects the environment due to the carbon emissions associated with logistics.

Operation and sales: During use, the LIDAR-Lite 3 consumes energy, which can come from a variety of sources, including non-renewable ones. The energy efficiency of the device can reduce this impact.

Disposal and recycling: At the end of its life cycle, improper disposal of the LIDAR-Lite 3 may result in environmental pollution. Reusing or recycling components may be a more sustainable option.

5.3.2 Impact on the Environment: Arduino Uno Microcontroller

Extraction of raw materials: The production of Arduino Uno requires silicone for the chips and various metals for the electronics. These processes can involve resource extraction, which impacts ecosystems.

Manufacturing and processing: Creating Arduino Uno chips and boards requires significant use of energy and chemicals, which can contribute to air and water pollution.

Transportation: Transporting Arduino Uno from the manufacturer to the end user generates a carbon footprint due to the fuel burned in vehicles.

Operating and selling: Using the Arduino Uno requires electricity, which can come from non-renewable sources. Reducing energy consumption through efficient programming and optimization can reduce the environmental impact.

Disposal and recycling: Improper handling of e-waste can cause pollution. Environmentally sound disposal and recycling of components is key to reducing environmental impact.

5.3.3 Environmental Impact: Other Components

Extraction of raw materials: Mining materials for additional components such as PCBs, displays, and enclosures also impacts the environment through emissions and changes to ecosystems.

Manufacturing and processing: Manufacturing these components requires energy and resources, including chemicals that can cause pollution.

Transportation: The transportation of components and finished products contributes to the overall carbon footprint through the use of fuel.

Operation and sales: The operation of these components can include the consumption of electricity, as well as the potential generation of waste during the sale phase.

Disposal and recycling: At this stage, it is important to ensure that components are either recycled or disposed of responsibly to reduce their environmental impact.

In general, each stage of the life cycle of the Arduino-based Altimeter with LIDAR-Lite 3 sensor has its own specific environmental impact. From raw material extraction to disposal, each component of the equipment contributes to the overall environmental footprint of the device. It is clear that to reduce this impact, it is important to use sustainable materials, efficient production techniques, and ensure that components are recyclable at every stage of the life cycle.

5.4. Comparison of the environmental impact with other similar devices

When evaluating the Arduino and LIDAR-Lite 3 Altimeter's role in the realm of altitude measurement tools, it's vital to consider how it stacks up against similar devices in terms of environmental impact. This comparative analysis spans various dimensions, from materials used to energy efficiency and end-of-life considerations.

The Altimeter uses a LIDAR-Lite 3 sensor that consumes less power and is smaller than traditional LIDAR systems. This reduces the environmental impact in the production and use phases. Traditional LIDAR systems require more raw materials and energy to produce, which increases their carbon footprint. Compared to larger LIDAR systems, the LIDAR-Lite 3 sensor requires fewer raw materials, which reduces the impact on natural ecosystems and lowers the carbon footprint. The compactness and efficiency of the sensor's design contribute to less waste in disposal compared to larger, less efficient LIDAR systems.

The PCB in the altimeter is designed with material and size efficiency in mind. Compared to larger and less optimized boards in other devices, the altimeter's PCB minimizes the use of harmful chemicals and reduces waste. The compact PCB of the altimeter reduces the need for raw materials, especially copper, compared to larger boards used in other devices. Due to the compact size of the altimeter PCB, its recyclability and environmental footprint is higher than that of larger boards, which often contain larger amounts of toxic materials.

The Arduino Uno used in the altimeter has an advantage due to its energy efficiency and compact size. In similar systems that use larger and less efficient microcontrollers, the environmental impact is higher due to increased energy consumption and production waste. A traditional microcontroller is larger and has higher power consumption. The Arduino Uno consumes less power and requires fewer rare earth metals to produce, reducing the environmental impact compared to larger microcontrollers.

Comparison of the altimeter with similar devices, such as traditional altimeters, shows that the Arduino-based altimeter with LIDAR-Lite 3 sensor has better environmental performance. This is due to its energy efficiency, compactness, choice of materials, and design, which are aimed at reducing the environmental impact at all stages of the life cycle - from production to disposal. Overall, the altimeter demonstrates significant advantages in terms of environmental impact compared to similar products.

5.5. Recommendations for Minimizing Environmental Impact

To minimize our environmental footprint, we need to implement a number of environmental initiatives and strategies. These recommendations cover all aspects of the product life cycle, from production to disposal.

Use of environmentally friendly materials. It is important to use less toxic materials such as lead-free solder alloys and brominated refractory compounds in the production of PCBs and other components.

Using biodegradable plastics and recycled materials for the altimeter housing can significantly reduce its environmental impact.

Optimizing manufacturing processes to reduce energy consumption. This may include the use of renewable energy sources such as solar or wind power.

Minimizing waste in the production process by recycling production waste and reducing production volumes.

Designing with the ability to easily replace or upgrade key components to ensure longevity and reduce the need for frequent replacements.

Minimizing the size and weight of components to reduce environmental impact during transportation.

Applying efficient logistics solutions, such as increasing the efficiency of transportation routes and using low-emission vehicles.

Developing collection and recycling programs for old components to reduce waste and promote the reuse of materials.

Provide consumers with information on the proper disposal of the altimeter and its components.

The application of these recommendations will help to reduce the environmental impact of the altimeter with LIDAR-Lite 3 sensor, ensuring its sustainable development and reducing its environmental impact. These measures are key to achieving environmental sustainability and responsibility in the modern world.

Conclusion

In this review of the environmental aspects of the life cycle of the Arduino-based Altimeter with LIDAR-Lite 3 sensor, we have identified the importance of a thorough understanding of the environmental impacts at every stage, from production to disposal. Using the Life Cycle Assessment (LCA) methodology, we assessed the environmental impact of each component and identified key challenges and opportunities for sustainable development.

The analysis showed that the production, use and disposal of each piece of equipment has its own unique environmental impact. We faced challenges related to the extraction of raw materials, energy consumption during production, and transportation. The issue of component recycling also proved to be important, especially given the toxic materials and complexity of recycling.

A comparative analysis with similar devices revealed the advantages of the altimeter in terms of energy efficiency and environmental sustainability. However, to achieve greater sustainability, some aspects, such as material selection and component design, need to be improved.

The recommendations presented in the document are key to minimizing environmental impact. They include the introduction of environmentally friendly materials, optimization of production, transportation efficiency, and responsible disposal. Particular attention should be paid to consumer education and involvement in the sustainable use and disposal of equipment.

In general, the consideration of the Arduino-based Altimeter with LIDAR-Lite 3 sensor emphasizes the importance of an integrated approach to environmental sustainability in the development of electronic devices. We are faced with the task of not only improving technologies, but also ensuring their sustainable development, which includes responsible use of resources, minimizing environmental impact, and extending the life cycle of equipment.

6. OCCUPATIONAL SAFETY AND HEALTH

The subject of this document is the comprehensive safety management in the workplace for the development of LiDAR and MEMS-based Obstacle Detection Systems for Ground Robotic Systems. This encompasses a detailed analysis and implementation of various safety measures tailored to the unique requirements and challenges posed by the development and testing of such advanced technological systems. The focus extends beyond conventional workplace safety, delving into specifics such as microclimate control, fire safety, and ergonomic considerations, ensuring a secure and efficient environment for specialist-developers engaged in this cutting-edge field. Our objective is to establish a standard of safety that not only complies with regulatory requirements but also fosters innovation and productivity in the realm of robotic technology development.

6.1. Analysis of Hazardous Factors in the Workplace

6.1.1. Microclimatic Conditions

Elaborate on the importance of monitoring and controlling temperature and humidity levels in the workspace. Highlight the significance of maintaining a stable microclimate for the optimal functioning of sensitive LiDAR and MEMS components.

6.1.2. Electrical Safety

Tailor safety measures to the specific electrical hazards associated with LiDAR and MEMS component development.

6.1.3. Noise Exposure

Address noise challenges specific to the machinery and equipment used in the development and testing of your system.

6.1.4. Handling Harmful Substances

Include safety protocols for any chemicals or hazardous materials involved in LiDAR and MEMS development.

6.1.5. Adequacy of Lighting

Emphasize the need for adequate lighting to ensure precision in assembling and testing LiDAR and MEMS components.

6.1.6. Vibration

Analyze the impact of vibrations on the functionality of LiDAR and MEMS devices, and the well-being of the development team.

6.1.7. Radiation

Consider any radiation risks, particularly from electronic components and testing equipment.

6.1.8. Ergonomic Challenges

Adapt ergonomic recommendations to the specific tasks involved in developing your system, such as assembling small components or extended periods of computer work.

6.1.9. Chemical Exposure

Assess the exposure to any chemicals used in the development process, including adhesives or solvents.

6.2. Analysis of working conditions and development of protective measures

Classification of Physical Work Intensity

As per ДСН 3.3.6.042-99, physical labor is classified into three intensity levels: light (categories Ia and Ib), medium (categories IIa and IIb), and heavy (category III). This categorization is crucial in determining the appropriate microclimate for various work activities. In the case of a specialist-developer working on LiDAR and MEMS-based systems, the majority of tasks are sedentary and intellectual in nature, fitting primarily into category Ia. This category typically involves minimal physical exertion, highlighting the importance of a well-regulated microclimate for both comfort and cognitive efficiency.

General Provisions of the Microclimate

The standard, ДСН 3.3.6.042-99, is comprehensive in its scope, applying to a wide range of production premises. It sets the baseline for microclimate conditions, focusing on environments where precise control is both possible and necessary. Excluded from its purview are environments like underground mines and mobile transports, where microclimate control is more challenging. This standard is particularly relevant for workplaces like ours, where technological development and precision are paramount.

Microclimate Parameters Requirements

Under the guidelines of ДСН 3.3.6.042-99, the microclimate in production areas is characterized by several key parameters: air temperature, relative air humidity, air movement speed, the intensity of thermal radiation, and surface temperature. These parameters are vital in creating a conducive work environment. For a workspace dedicated to the development of LiDAR and MEMS-based systems, maintaining optimal microclimate conditions is not just a matter of comfort but also crucial for the proper functioning and longevity of sensitive equipment and the overall quality of the developmental work.

Various types of microclimate norms

- Air temperature
- Relative humidity,
- Speed of air movement,
- The intensity of thermal (infrared) radiation,
- Surface temperature.

According to the degree of influence on the thermal condition of a person, microclimatic conditions are divided into optimal and permissible.

Optimum and permissible microclimatic conditions are established for the working area of industrial premises, taking into account the difficulty of the work performed and the period of the year. When simultaneous execution of works of different categories of difficulty in the work area, the levels of microclimate indicators should be set taking into account the largest group of workers.

Optimal Microclimate Conditions

In aligning with ДСН 3.3.6.042-99, optimal microclimate conditions are outlined for permanent workstations. This encompasses maintaining a consistent air temperature across the workspace, ensuring no significant variations occur that could impact both worker comfort and equipment performance. The standard also stipulates that the temperature of internal surfaces like walls, floors, and ceilings, along with the surfaces of technological equipment, should not deviate from the normative air temperature suited to the category of work by more than 2 degrees Celsius. This precision in temperature control is essential in environments where technological development is sensitive to even minor fluctuations in environmental conditions.

The standard specifies that for light work categories like Ia, the optimal conditions include maintaining an air temperature of 22-24°C during the colder season and 23-25°C during the warmer season. Relative air humidity should be kept between 40-60%, and air movement speed should not exceed 0.1 m/s. These conditions ensure both worker comfort and the stability of sensitive technological equipment used in the development process.

Comparison and Compliance Assessment:

In the current setup of the workplace for LiDAR and MEMS development, these norms are largely adhered to. The air temperature is maintained within the optimal range, considering both seasonal variations and the nature of the work.

Relative air humidity is managed within the recommended parameters, ensuring comfort and equipment safety.

Air movement speed is controlled to prevent disruptive drafts, while also ensuring sufficient ventilation.

The thermal radiation intensity and surface temperature are monitored to maintain a conducive environment for both staff and technological processes.

6.3. Fire Safety

6.3.1. Risk Assessment:

- Identify Specific Risks: Evaluate risks associated with electrical circuits, batteries, and any solvents or chemicals used in LiDAR and MEMS development.
- Assess Material Flammability: Assess the flammability of materials used in the development and testing of your systems.
- Regular Risk Reviews: Conduct periodic reviews to identify new risks as technology and workplace practices evolve.

6.3.2. Fire Prevention Measures:

- Electrical Safety Compliance: Ensure all electrical equipment complies with safety standards to prevent overheating and short circuits.
- Control of Flammable Substances: Store flammable materials in designated, safe areas with proper ventilation.
- Maintenance of Fire Prevention Systems: Regularly test and maintain fire alarms, sprinkler systems, and other fire prevention technologies.

6.3.3. Emergency Exits and Routes:

- Clear and Accessible Exits: Maintain clear pathways to all emergency exits.
- Signage and Lighting: Ensure exits are well-marked and illuminated.
- Evacuation Drills: Conduct regular evacuation drills to familiarize staff with emergency exit routes.

6.3.4. Fire Detection and Alarm Systems:

- Installation of Detectors: Install smoke and heat detectors in key areas, especially where high-risk activities occur.

- Regular Testing: Test these systems regularly to ensure they are operational.
- Integration with Emergency Services: Systems should be capable of alerting local fire services automatically in case of a fire.

6.3.5. Fire Suppression Equipment:

- Appropriate Extinguishers: Equip the workspace with fire extinguishers suitable for electrical and chemical fires.
- Accessibility: Place fire extinguishers in easily accessible locations.
- Training in Use: Regularly train employees on the correct usage of fire extinguishers and other fire suppression tools.

6.3.6. Training and Awareness:

- Regular Fire Safety Training: Educate employees about fire risks and the correct response to a fire incident.
- Hazard Awareness: Highlight common fire hazards in the workplace.
- Mock Drills: Conduct mock fire drills to ensure employees are prepared in case of an emergency.

6.3.7. Safe Handling of Chemicals and Materials:

- Material Safety Data Sheets (MSDS): Keep MSDS for all chemicals used in the workplace.
- Proper Storage: Store chemicals according to their fire risk and compatibility.
- Spill Response Plan: Have a plan in place for dealing with chemical spills.

6.3.8. Regular Inspections and Audits:

- Internal Audits: Conduct routine internal audits to identify potential fire hazards.

- External Inspections: Engage external experts for comprehensive fire safety inspections.
- Continuous Improvement: Use audit findings to continuously improve fire safety measures.

6.3.9. Documentation and Record Keeping:

- Safety Records: Maintain records of all fire safety equipment inspections and maintenance activities.
- Training Logs: Keep detailed logs of all fire safety training sessions.
- Incident Reports: Document any fire-related incidents, no matter how small, for analysis and improvement.

6.3.10. Coordination with Local Fire Services:

- Consultation and Collaboration: Establish a line of communication with local fire services for advice and support.
- Emergency Response Planning: Work with fire services to develop a site-specific emergency response plan.
- Joint Drills: Conduct joint emergency drills with local fire services to test and improve response strategies.

Conclusion

In conclusion, this document underscores the vital importance of rigorous safety practices in the development of LiDAR and MEMS-based obstacle detection systems for ground robotic applications. It highlights the necessity for continuous vigilance, regular updates to safety protocols, and an unwavering commitment to the well-being of the development team. The document also serves as a reminder that safety in such high-tech environments is a dynamic, ongoing process, requiring active participation and adaptation to technological advancements and regulatory changes. This commitment to safety is integral not only for compliance but also for fostering a culture of innovation and excellence in the field of robotic system development.

LITERATURE

1. Laser rangefinder LIDAR-Lite v3 [Electronic resource] - Retrieved on November 20, 2023: https://arduino.ua/prod5288-lazernii-dalnomer-lidar-lite-v3.

2. High-performance Optical Distance Measurement Sensor [Electronic resource] - Retrieved on November 20, 2023: https://www.garmin.com/en-US/p/557294#overview.

3. Announcing the Arduino IDE 2.0 (beta) | Arduino Blog [Electronic resource] - Retrieved on November 20, 2023: https://blog.arduino.cc/2021/03/01/announcing-the-arduino-ide-2-0-beta/.

5. Arduino is an Italian open-source hardware and software company [Electronic resource] - Retrieved on November 20, 2023: https://en.wikipedia.org/wiki/Arduino.

6. Arduino Uno Rev3 - Arduino Online Shop [Electronic resource] - Retrieved on November 20, 2023: https://store-usa.arduino.cc/products/arduino-uno-rev3.

7. Can I force the Arduino IDE to recompile everything? https://forum.arduino.cc/t/can-i-force-the-arduino-ide-to-recompile-everything/866547.

8. Cannot find Arduino IDE VS Code [Electronic resource] - Retrieved on November 20, 2023: https://stackoverflow.com/questions/64188588/cannot-find-arduino-ide-vs-code.

9. Uninstall for a fresh re-install [Electronic resource] - Retrieved on November 20, 2023: https://forum.arduino.cc/t/uninstall-for-a-fresh-re-install/349503.

10. Program Your Arduino Like a Pro with the Eclipse C/C++ IDE [Electronic resource] - Retrieved on November 20, 2023: https://www.eclipse.org/community/eclipse_newsletter/2017/april/article4.php.

11. The Best Arduino Simulators of 2023 (Online & Offline) [Electronic resource] - Retrieved on November 20, 2023: https://all3dp.com/2/best-arduino-simulators-online-offline/.

12. MakeCode Maker - Blocks / Javascript editor [Electronic resource] - Retrieved on November 20, 2023: https://maker.makecode.com/.

13. Learn how to use Tinkercad [Electronic resource] - Retrieved on November 20, 2023: https://www.tinkercad.com/learn/circuits.

14. Wokwi - Online ESP32, STM32, Arduino Simulator [Electronic resource] - Retrieved on November 20, 2023: https://wokwi.com/.

15. Virtual Breadboard [Electronic resource] - Retrieved on November 20, 2023: https://www.virtualbreadboard.com/.

16. Flowcode - Visual Programming Language [Electronic resource] - Retrieved on November 20, 2023: https://www.flowcode.co.uk/.

17. Simulator for Arduino [Electronic resource] - Retrieved on November 20, 2023: https://virtronics.com.au/Simulator-for-Arduino.html.

18. Virtual System Modelling Software - VSM MCU Simulation - Proteus [Electronic resource] - Retrieved on November 20, 2023: https://www.labcenter.com/whyvsm/.

19. I2C LCD on Arduino [Electronic resource] - Retrieved on November 20, 2023: https://www.make-it.ca/i2c-lcd-display-on-arduino/.

20. Raspberry Pi or Arduino Uno? One Simple Rule to Choose the Right [Electronic resource] - Retrieved on November 20, 2023: https://makezine.com/2015/12/04/admittedly-simplistic-guide-raspberry-pi-vs-arduino/.

21. The SIM800 Cellular Module and Arduino: A Powerful IoT Combo [Electronic resource] - Retrieved on November 20, 2023: https://predictabledesigns.com/the-sim800-cellular-module-and-arduino-a-powerful-iot-combo/.

22. Third Eye for The Blind [Electronic resource] - Retrieved on November 20, 2023: https://create.arduino.cc/projecthub/muhammedazhar/third-eye-for-the-blind-8c246d.

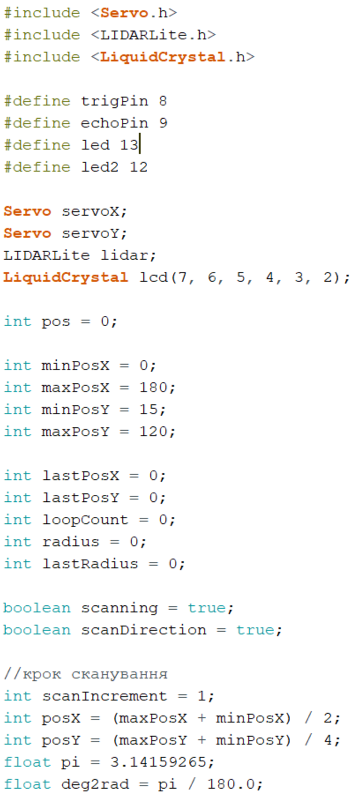
23. What is Arduino? http://www.arduino.cc/en/Guide/Introduction.

24. What is an Arduino? https://learn.sparkfun.com/tutorials/what-is-an-arduino/all.

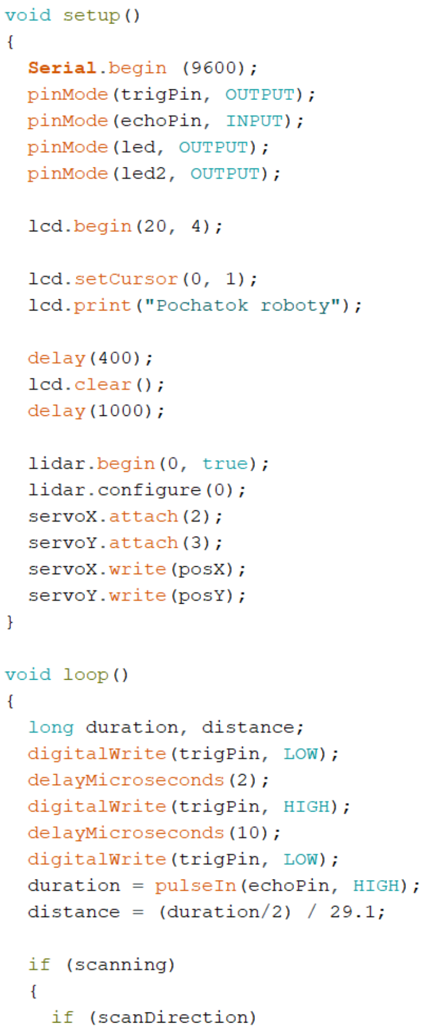
25. State sanitary standards for the microclimate of industrial premises DSN 3.3.6.042-99: Resolution of the Ministry of Health of Ukraine of December 01, 1999, No. 42.

26. DNAP - Normative legal library of instructions and documents. DNAP - Normative legal library of instructions and documents. URL: https://dnaop.com/ (accessed on December 13, 2023).

APPENDIX A



```
#include <Servo.h>
#include <LIDARLite.h>
#include <LiquidCrystal.h>

#define trigPin 8
#define echoPin 9
#define led 13
#define led2 12

Servo servoX;
Servo servoY;
LIDARLite lidar;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

int pos = 0;

int minPosX = 0;
int maxPosX = 180;
int minPosY = 15;
int maxPosY = 120;

int lastPosX = 0;
int lastPosY = 0;
int loopCount = 0;
int radius = 0;
int lastRadius = 0;

boolean scanning = true;
boolean scanDirection = true;

//крок сканування
int scanIncrement = 1;
int posX = (maxPosX + minPosX) / 2;
int posY = (maxPosY + minPosY) / 4;
float pi = 3.14159265;
float deg2rad = pi / 180.0;
```



```
void setup()
{
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led, OUTPUT);
  pinMode(led2, OUTPUT);

  lcd.begin(20, 4);

  lcd.setCursor(0, 1);
  lcd.print("Pochatok roboty");

  delay(400);
  lcd.clear();
  delay(1000);

  lidar.begin(0, true);
  lidar.configure(0);
  servoX.attach(2);
  servoY.attach(3);
  servoX.write(posX);
  servoY.write(posY);
}

void loop()
{
  long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (scanning)
  {
    if (scanDirection)
```

```
    if (scanDirection)
    {
      posX += scanIncrement;
    }
    else
    {
      posX -= scanIncrement;
    }
    if (posX > maxPosX || posX < minPosX)
    {
      scanDirection = !scanDirection;
      posY += scanIncrement;
    }
  }

  posX = min(max(posX, minPosX), maxPosX);
  posY = min(max(posY, minPosY), maxPosY);
  bool moved = moveServos();
  displayPosition();

  loopCount += 1;
  if (loopCount % 100 == 0) {
    radius = lidar.distance();
  } else {
    radius = lidar.distance(false);
  }
  if (abs(radius - lastRadius) > 2)
  {
    lastRadius = radius;
    lcd.setCursor(8, 0);
  }
  if (scanning || moved) {
    float azimuth = posX * deg2rad;
    float elevation = (180 - maxPosY + posY) * deg2rad;
    double x = radius * sin(elevation) * cos(azimuth);
    double y = radius * sin(elevation) * sin(azimuth);
    double z = radius * cos(elevation);
    Serial.println(String(-x, 5) + " " + String(y, 5) + " " + String(-z, 5));
  }



  if (distance < 240) {
    digitalWrite(led,HIGH);

  lcd.clear();
  lcd.setCursor(0, 1);
  lcd.print("Vuyavleno pereshkodu");
    Serial.println("Vuyavleno pereshkodu");

  delay(400);
  lcd.clear();
  digitalWrite(led,LOW);
  delay(200);
  }
  else {
    digitalWrite(led,LOW);
  }
  lcd.setCursor(1, 0);
  lcd.print("Vidstan = ");
  lcd.print(distance);
  lcd.println(" cm");

  Serial.print(distance);
  Serial.println(" cm");

  delay(500);
}
```

```
bool moveServos()
{
  bool moved = false;
  static int lastPosX;
  static int lastPosY;
  int delta = 0;
  if (posX != lastPosX)
  {
    delta += abs(posX - lastPosX);
    servoX.write(posX);
    lastPosX = posX;
    moved = true;
  }
  if (posY != lastPosY)
  {
    delta += abs(posY - lastPosY);
    servoY.write(posY);
    lastPosY = posY;
    moved = true;
  }
  delay(30);
  return moved;
}

void displayPosition()
{
  static int lastPosX;
  static int lastPosY;
  if (posX != lastPosX)
  {
    lastPosX = posX;
  }
  if (posY != lastPosY)
  {
    lastPosY = posY;
  }
}
```