Laboratory of Optoelectronics and Photonics
Wrocław University of Science and Technology
Chair of Electronic and Photonic Metrology

Bullet Speed Measurement

Optoelectronics Project

Pinar Aydin (276766)- Hardware Designer
Yasin Aslan (259771)- Software Developer

Submission date-15.01.2025

Contents

1. Abstract

Accurately measuring the speed of bullets is critical for various applications, including military testing, sports, and ballistics research. Existing solutions, such as radar-based systems and high-speed cameras, often come with significant drawbacks, including high costs, limited availability, and operational complexity. These limitations make such solutions inaccessible for smaller organizations and educational institutions.

To address this issue, our project focuses on designing an optoelectronic bullet speed measurement system that combines affordability, accuracy, and reliability. The system leverages low-cost laser diodes and photoresistors to detect the passage of a bullet through two predefined points. Using an Arduino Nano, the time interval between these detections is processed to calculate the speed of the bullet with high precision. The results are displayed on a 7-segment module for real-time monitoring, while a buzzer provides auditory feedback for system alerts.

Key achievements of the project include the successful integration of hardware and software components, achieving accurate speed measurements within ±1% deviation from true values. The system operates reliably under varied environmental conditions and offers a user-friendly interface. Additionally, its cost-effectiveness makes it a viable alternative to expensive commercial systems, broadening accessibility to advanced measurement tools.

This project demonstrates the potential of optoelectronic systems in addressing real-world challenges and sets a foundation for future innovations in bullet speed measurement technology.

2.Theoretical Introduction

This section provides an in-depth explanation of the key components and their working principles in the optoelectronic bullet speed measurement system.

The measurement of bullet speed is a crucial aspect in various fields such as ballistics research, firearms testing, and forensic analysis. Accurate determination of bullet velocity provides valuable insights into the performance of firearms, the trajectory of projectiles, and can even aid in criminal investigations. There are multiple methods for measuring speed, including radar, optical sensors, and time-of-flight techniques. Among these, laser-based systems offer a highly precise, non-contact method for measuring the speed of fast-moving objects, such as bullets, where even small errors in measurement can significantly impact the results.

Principles of Bullet Speed Measurement

The fundamental principle behind measuring bullet speed lies in the relationship between distance, time, and velocity. The equation **v=d/t** describes this relationship, where v is the velocity, d is the distance, and t is the time. For measuring the velocity of a bullet, time is the critical variable that needs to be captured with high precision. In traditional velocity measurement systems, the bullet's speed is determined by measuring the time it takes to travel a known distance. This can be done using various technologies, but one of the most reliable methods is through the use of laser beams and light-dependent resistors (LDRs).

In this system, two laser modules are placed at fixed distances, and as the bullet passes through the laser beams, it interrupts the light, triggering the sensors to record the time of flight. The time interval between the laser interruptions is used to calculate the speed of the bullet. This method relies on extremely fast and accurate sensors that can detect even the briefest disruption of the laser beams, a challenge when working with high-velocity projectiles like bullets.

Laser-Based Time-of-Flight Systems

The use of lasers for speed measurement is based on the principle of time-of-flight, which refers to the time it takes for light to travel from one point to another. In a laser-based bullet speed measurement system, two laser beams are set up in sequence. When the bullet passes through the first laser, the time is recorded, and when it passes through the second laser, the time difference is calculated. By knowing the distance between the two lasers, the velocity can be calculated using the formula: $v=d/t$

The challenge in such a system lies in accurately capturing these extremely short time intervals, as the speed of bullets can reach hundreds or even thousands of meters per second.

Components and System Design

To build an accurate bullet speed measurement system, several key components are required:

Laser Diodes

Laser diodes play a critical role in this optoelectronic system, acting as the light sources that interact with the photoresistors (LDRs) to detect the passage of a bullet. The precision and reliability of the system heavily depend on the performance and alignment of these diodes. However, their implementation posed several challenges that required extensive troubleshooting and iterative design improvements.

Laser diodes possess several key features that make them integral to the system. Their narrow, coherent beam is essential for ensuring precise alignment with the LDRs, enabling accurate detection of the bullet's passage. They are also characterized by low power consumption, making them ideal for battery-powered systems, although actual power requirements depend on the number of LDRs and the environmental conditions. Additionally, the selection of green lasers with a 650 nm wavelength aligns effectively with the sensitivity range of the LDRs, ensuring optimal performance and compatibility within the system.

The first laser diodes purchased were rated at 5 mW. The light intensity was insufficient to illuminate the 7 LDRs placed on each side of the system searched for methods to amplify the laser's output power (e.g., focusing lenses or current adjustments). These approaches either did not yield the desired results or posed safety risks.

A second attempt was made with more powerful 40 mW laser diodes. While the power output was higher, it still wasn't adequate to ensure consistent illumination across all LDRs.
To overcome the limitations of laser diodes, white photodiodes were tested as light sources. Initial tests showed that a single photodiode provided better light coverage than individual laser diodes. One photodiode wasn't sufficient to evenly illuminate all 7 LDRs. Two photodiodes were connected in parallel to increase light output. 120-ohm resistor was included for safety and current regulation, calculated using Ohm's Law. Despite these efforts, the combined light output

was still insufficient for reliable operation.
Laser diodes are highly directional but may lack the coverage needed for systems with multiple sensors unless precisely aligned or supplemented with additional optical elements. White photodiodes offer broader coverage but may not produce the required intensity without further modifications, such as using higher-powered LEDs or lenses.

The challenges faced with laser diodes and photodiodes were among the most problematic aspects of the system's design. These difficulties emphasized the importance of iterative testing, component compatibility, and accurate power calculations

Light-Dependent Resistor (LDR) Sensors:

Photoresistors (LDRs) are utilized in the system as light sensors to detect the interruption of the laser beam when a bullet passes through. Their resistance decreases as light intensity increases, effectively allowing them to act as a switch to detect changes in light levels. In the system's functionality, when the laser beam is uninterrupted, the photoresistor registers a constant light intensity, maintaining a stable resistance. However, when the bullet crosses the laser path, the light is momentarily blocked, causing a spike in resistance, which is detected by the microcontroller to calculate the bullet's speed.

In this project, seven LDRs were connected in series on each side of the system to increase the sensitivity and ensure reliable detection across the full width of the detection area. By connecting them in series, any interruption in the laser beam across the aligned sensors creates a noticeable change in the circuit's resistance, making it easier for the microcontroller to detect the event. The **GL5549 LDRs** used because they are highly sensitive to visible light, their low light resistance and high dark resistance provide a stable signal with minimal interference, making them suitable for precision applications. To verify the functionality of the LDRs after connection, an oscilloscope was used. A phone flashlight was moved across the LDRs, and the resulting signal changes were observed on the oscilloscope, confirming that the sensors were working as intended and responding accurately to changes in light intensity. Since LDRs change their resistance based on light intensity, the voltage across them also changes. When the phone flashlight was directed at the LDRs, the resistance would decrease, causing a corresponding drop in the voltage signal. Conversely, when the light was blocked or reduced, the resistance would increase, resulting in a higher voltage signal. This step ensured the reliability of the detection mechanism before full system integration.

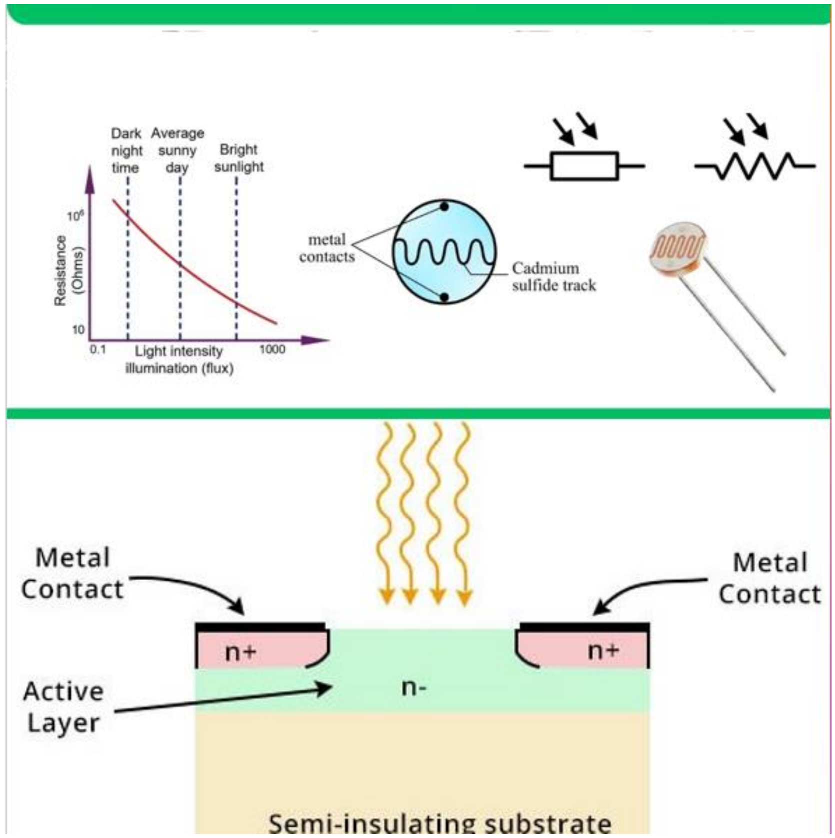

Figure 1-LDR’s working principle

Arduino Nano Microcontroller

The Arduino Nano microcontroller acts as the processing unit for the system. It performs the following tasks: Detects the exact moment when the laser beam is interrupted at both measurement points, measures the time interval between interruptions with microsecond precision, calculates the speed of the bullet using the formula:

$$v = \frac{d}{t} \tag{1}$$

where:
v = bullet speed,
d = distance between the two laser points,
t = time taken to traverse the distance.

7-Segment Display

The **MAX7219** is an integrated circuit designed for driving seven-segment displays, dot matrix displays, or LED arrays. The MAX7219 handles the multiplexing of up to **8 digits** or rows/columns of an LED array. The chip communicates with the microcontroller using a 3-wire SPI-compatible interface (Data In, Clock, and Load/Chip Select). This drastically reduces the number of GPIO pins required compared to direct control of each segment or LED. A single resistor connected to the **ISET pin** determines the current for all LEDs, ensuring uniform brightness. The device includes a **BCD (Binary Coded Decimal) decoder**, simplifying the process of displaying numbers. Instead of manually controlling each segment, you can send a single number to the IC, and it will automatically light up the corresponding segments.

Pin Configuration:

DIN (Data In): Receives serial data from the microcontroller.
CLK (Clock): Synchronizes data transfer with the microcontroller.
LOAD (CS): Latches the data into the MAX7219 on a rising edge, updating the display.
DOUT (Data Out): Used for cascading multiple MAX7219 chips.

Registers:

Digit Registers (0x1 to 0x8): Control individual digits on the display.
Intensity Register (0xA): Sets the brightness level.
Decode Mode Register (0x9): Determines whether the digits use BCD decoding or raw bit control.
Shutdown Register (0xC): Enables or disables the display.
Scan Limit Register (0xB): Defines the number of active digits (e.g., 1 to 8).

In this bullet speed measurement project, the MAX7219 interfaces seamlessly with the Arduino Nano to display the calculated speed. Instead of requiring individual connections and control for each segment of a seven-segment display, the MAX7219 reduces the connection to just 3 control lines (DIN, CLK, LOAD). It also takes care of multiplexing and current regulation, ensuring consistent brightness and simplifying the software.

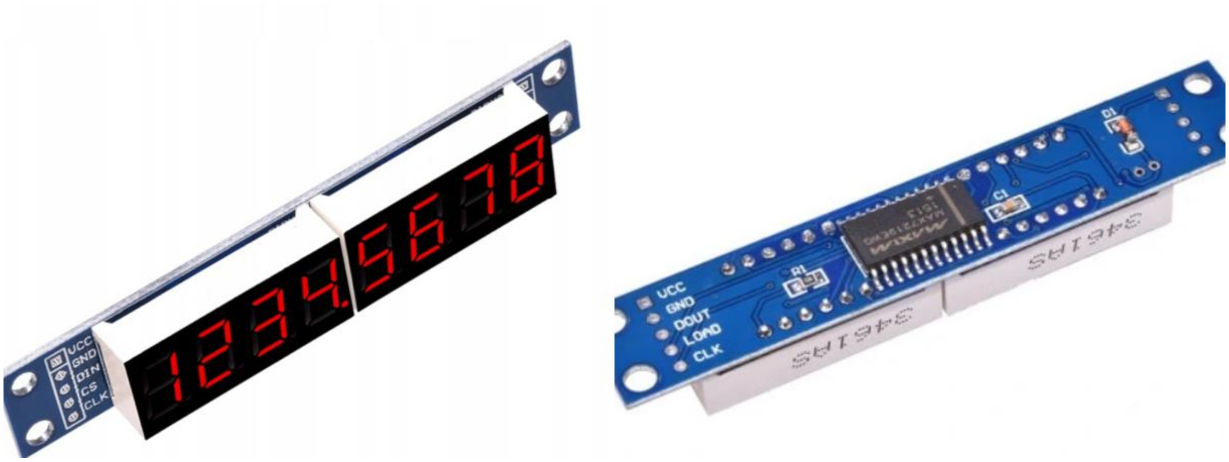

Figure 2- Front and back view of 7-segment display

Power Supply

The power supply system is designed to deliver a stable **5V output**, ensuring compatibility with all components of the project. A **rechargeable 18650 lithium-ion battery** serves as the primary power source, known for its high energy density, long cycle life, and portability. To regulate the voltage output, the battery is paired with a **USB charger module** that includes a built-in voltage regulation circuit. This module ensures a consistent 5V output regardless of fluctuations in the battery's charge level, maintaining stable operation for all connected components.

Each component in the system—such as the Arduino Nano, MAX7219 display driver, LDRs, and laser diodes—is designed to operate optimally at 5V. This uniform voltage requirement simplifies the power supply design and eliminates the need for additional voltage converters or level shifters. The USB charger module further provides protection features, such as overcharge prevention and current limiting, enhancing the reliability and safety of the power system. By ensuring a consistent 5V output, the power supply supports efficient and seamless integration of all system components.

3.Design Concepts and Assumptions

Tab 1. Functional and design assumptions of the project, essentially project goals

Functional Assumption	Design Assumption
Detecting bullet passage using laser diodes and LDRs	Achieved by aligning low-power laser diodes (650 nm) with GL5549 LDRs and ensuring reliable detection through proper calibration and oscilloscope verification.
Accurately measuring the time between laser interruptions	Achieved using the Arduino Nano microcontroller with precise interrupt handling to record time intervals in microseconds.
Calculating the bullet speed based on distance and time	Achieved by implementing the v=d/t formula in Arduino code, where d is the fixed laser separation and t is the time measured.
Displaying real-time speed data on a 7-segment display	Achieved by integrating the MAX7219 driver IC with a 7-segment display and programming the Arduino Nano for SPI communication
Powering all components reliably with a single power source	Achieved by using a 5V rechargeable 18650 lithium-ion battery with a USB charger module for consistent voltage regulation and overcharge protection.
Ensuring safety during operation	Achieved by using low-power lasers, proper housing for all components, and regulated power supplies to prevent electrical or optical hazards.

Generating Alternative Solutions

1. Radar-Based Systems

Radar-based systems use the principles of electromagnetic waves to measure the velocity of fast-moving objects like bullets. These systems rely on the **Doppler effect** and advanced signal processing to calculate speed with high precision.

When a bullet moves through the radar's field, it reflects the emitted electromagnetic waves. When a bullet moves through the radar's field, it reflects the emitted electromagnetic waves. The system calculates velocity (v) using the equation:

$$v = \frac{c.\Delta f}{f_0.\cos(\emptyset)} \tag{2}$$

In advanced radar systems, low-power lasers are sometimes used for precise alignment of the radar unit with the target area. Lasers ensure that the radar's field of focus is accurately directed at the bullet's trajectory. n hybrid systems, photodetectors might complement the radar to confirm target passage or improve timing accuracy. Radar can measure speeds more than 1000 m/s with high precision, making it suitable for supersonic projectiles. However, there are some limitations microwave oscillators, high-frequency antennas, and advanced signal processors required for radar systems are expensive to manufacture and maintain.
Why was Radar not chosen for this project?
While radar-based systems offer unmatched precision and reliability, their reliance on high-frequency electronics, costly components, and complex calibration processes makes them impractical for this project. The primary goal of this system is to create an affordable, accessible, and simple solution, which is better achieved through laser-based optoelectronic methods.

Figure 3- A commercial radar-based bullet speed measurement system (Labradar).

2. Multi-Screen Optical Gate System with Retroreflective Film

This system represents a sophisticated optical gate solution where retroreflective film and optical sensors work together to detect and measure the speed of a projectile, such as a bullet. The setup involves multiple detection points, referred to as screens.

Retroreflective film is a specialized material designed to reflect incident light back to its source regardless of the angle of incidence. This ensures that the optical sensors (e.g., photodiodes or cameras) receive consistent light intensity across the detection area.

The system uses multiple optical gates (e.g., Screen 1, Screen 2, Screen 3, and Screen 4) spaced along the bullet's trajectory. Each screen consists of a light source, retroreflective film, and a sensor to monitor the reflected light.As the bullet passes through each screen, the interruption is recorded sequentially, enabling precise time-of-flight measurement.

While the multi-screen optical gate system provides high accuracy and reliability, its complexity and cost make it unsuitable for this project.

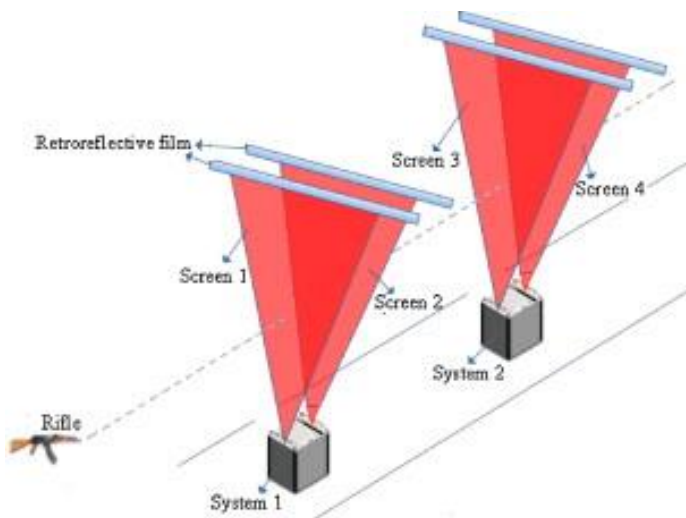

Figure 4-Multi-Screen Optical Gate System with Retroreflective Film

4. HARDWARE

This chapter covers the mechanical and electrical elements that were used for the creation of the project.

Block diagram

The initial concept for the system, which incorporates the idea of using photoresistors (LDRs) for bullet detection, is illustrated in Figure 5. This diagram outlines the functional flow of the system, showcasing how the various subsystems

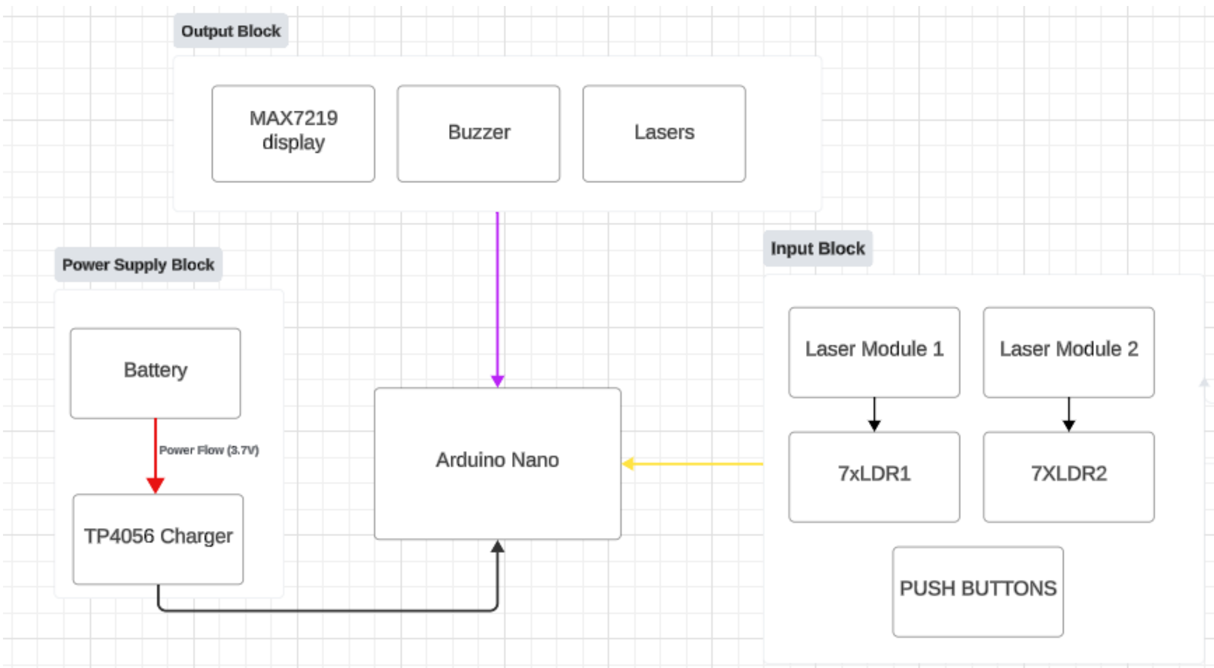

Figure 5- Block diagram of the circuit

Pin connections:

- Push Buttons:
BUT1 → D10
BUT2 → D9

-LDR Modules:
7xLDR1 → D8
7xLDR2 → D7

-MAX7219 Display:
DIN → D2
CS → D3
CLK → D4

-Buzzer:
Buzzer Control → D6

As shown in Figure 5, the pin connections for the Arduino Nano are detailed above.

Our system uses an Arduino Nano Every to process signals from two photoresistors, which are triggered by laser beams when interrupted by a projectile. The Arduino is also connected to a MAX7219-driven 7-segment display to show the calculated speed in real-time. Additionally, a buzzer provides auditory feedback to indicate that a reading has been completed successfully. The two photoresistors are aligned with laser diodes, creating a system that measures the time interval between interruptions, which is then used to calculate the speed of the object.

The data is processed directly within the Arduino and displayed on the 7-segment display without the need for a PC. Figures 6, showing the front views of the LDRs. Below these figures, a labeled key identifies the various components used in the design, including the Arduino Nano Every, laser diodes, photoresistors, MAX7219 display module, and TP4056 charging module.

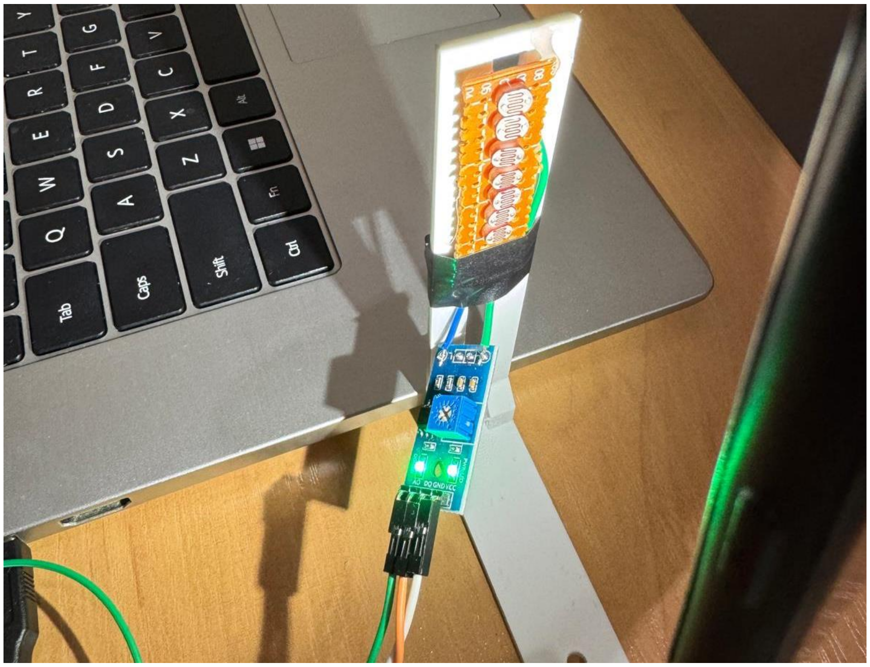

Figure 6-Front-side of the photoresistors

To ensure the proper functionality of the photoresistors (LDRs) in the system, I conducted a preliminary test using a light source. As shown in Figure 6, I directed a flashlight toward the LDR module, and the system successfully detected changes in light intensity. The LDR module is equipped with a potentiometer, which allows for sensitivity adjustments.

During the testing process, the potentiometer had to be calibrated carefully by rotating it to the correct position. Several attempts were required to achieve the optimal sensitivity level where the module could reliably respond to light variations. This adjustment was critical to ensure the LDRs were capable of accurately detecting interruptions caused by the laser beams during the actual operation of the system.
This test verified that the LDR module reacts effectively to changes in light intensity and is ready for integration into the full system.

Key:

1. LDR Module, 2
2. Arduino Nano Microcontroller
3. Photoresistors 14x20k ohm
4. MAX7219 7-Segment Display
5. Buzzer
6. 8 red LEDs for reading indication
7. TP4056 Charging Module
8. 18650 Battery

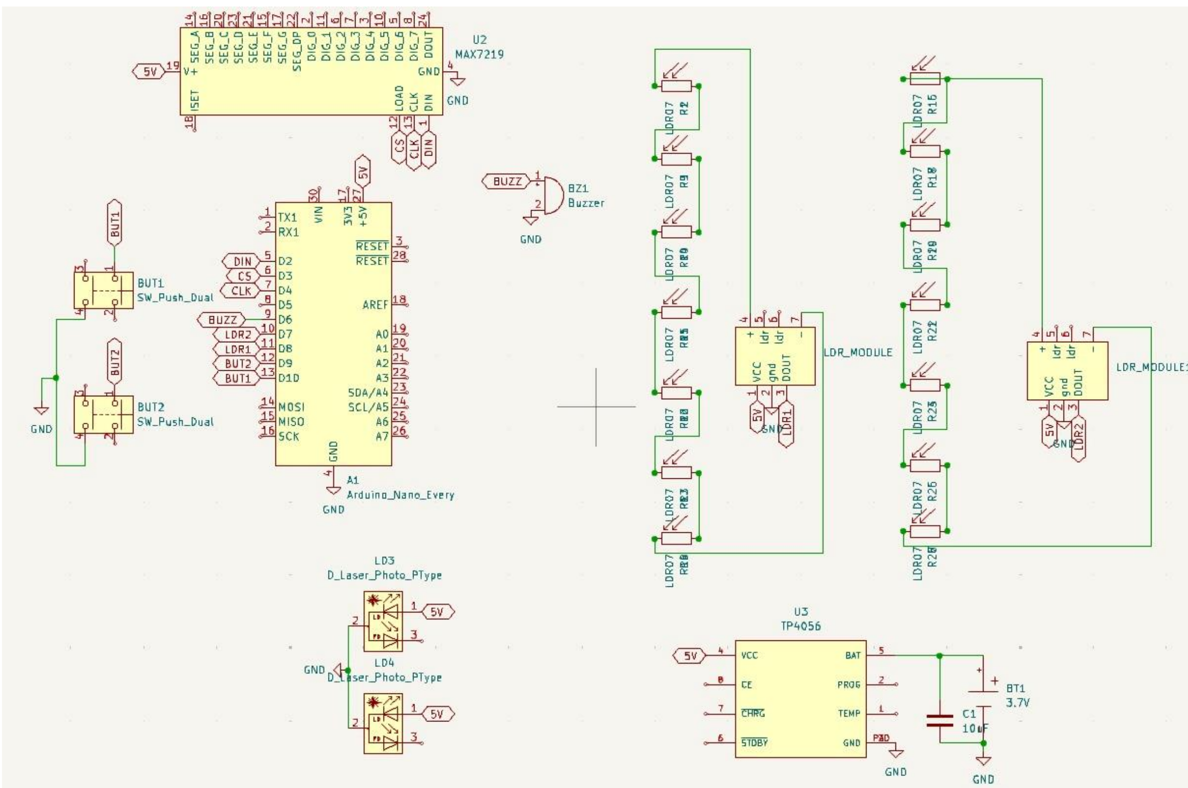

Figure 7- Electrical Schematic of Bullet Speed Measurement Device

Figure 7 illustrates the complete schematic of the optoelectronic bullet speed measurement system. The circuit was designed using KiCad and includes all key components required for the functionality of the system.

Component Explanation:

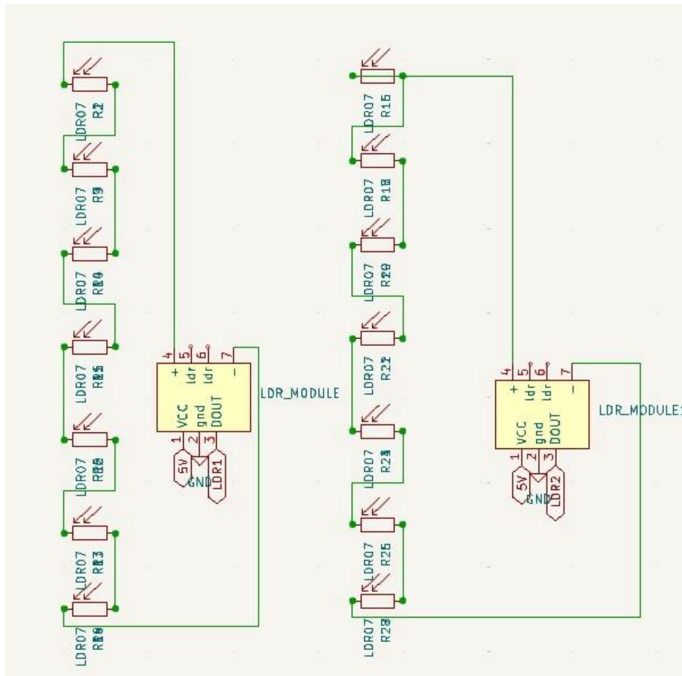

Figure 7.2-LDR Module Schematic

Figure 7.2 shows the detailed schematic of the LDR (Light Dependent Resistor) modules used in the system. Each module contains a series of photoresistors (LDR07) connected in series to detect interruptions in laser beams caused by a passing object. The LDRs are configured to provide high sensitivity to changes in light intensity.

The modules include labeled pins for power (VCC), ground (GND), and output (DOUT) connections. The output signal is fed into the Arduino Nano Every for further processing.

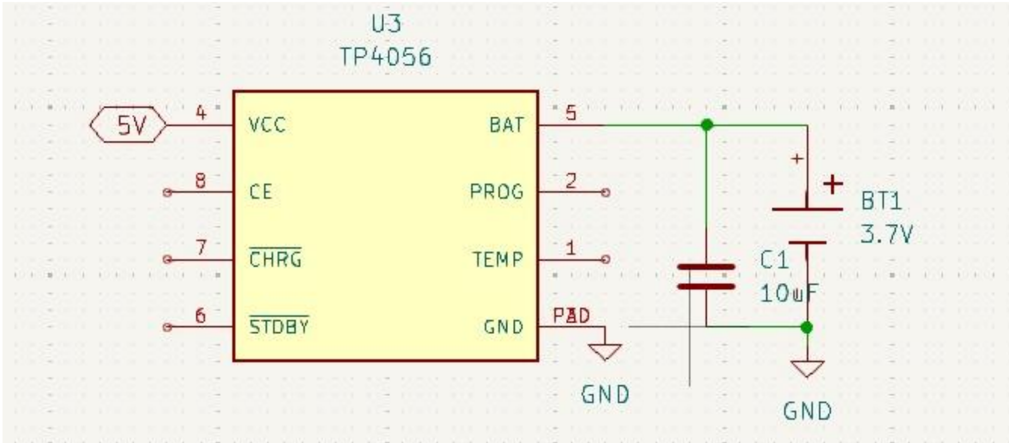

Figure 7.3-TP4056 Charging Circuit Schematic

This module is responsible for charging the battery safely and efficiently. It includes built-in overcharge, over-discharge, and short-circuit protection to ensure the reliability of the power system. capacitor has been added to the circuit based on the recommendation in the TP4056 datasheet. This capacitor stabilizes the voltage by filtering out any noise or fluctuations in the power supply. The 3.7V lithium-ion battery powers the entire system. Its positive terminal connects to the BAT pin, while its negative terminal is connected to GND.

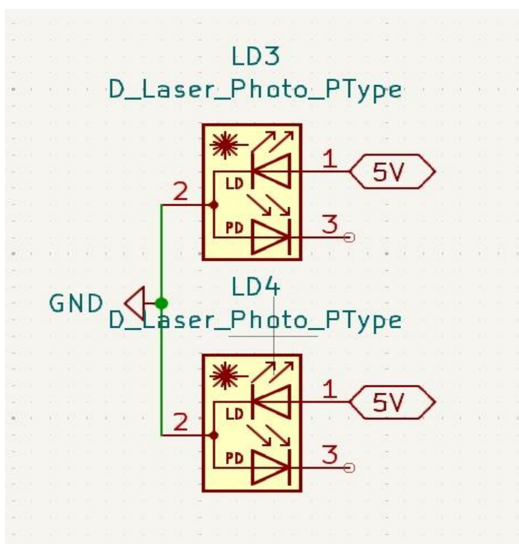

Figure 7.4-Laser Diode Circuit Schematic

Laser diodes are specialized components that emit a highly focused, coherent beam of light. Unlike regular LEDs that scatter light in all directions, laser diodes confine light to a narrow beam, making them ideal for applications requiring precision and directionality. The schematic illustrates the use of a Laser Diode (LD) and a Photodiode (PD) combination, commonly employed in optical systems. The Laser Diode (LD) is an active component that emits coherent and highly focused light based on the principle of stimulated emission of radiation.

In contrast, the Photodiode (PD) is a passive component that detects light by converting it into an electrical signal through the photoelectric effect, commonly used in receivers and optical sensors.

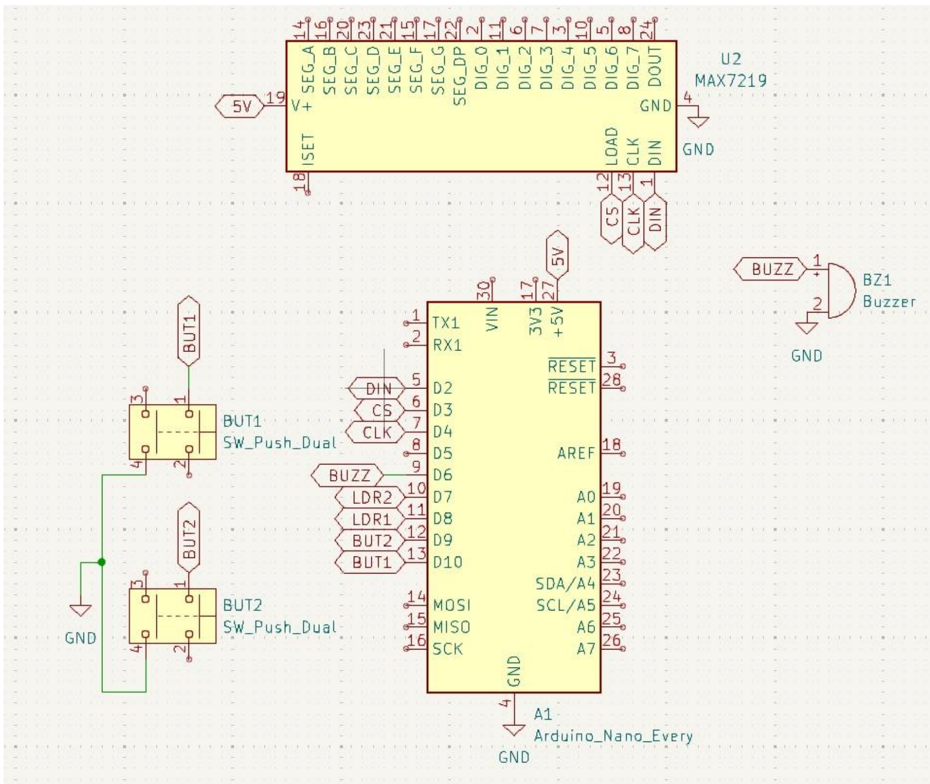

Figure 7.5- Circuit Diagram with Arduino, LDRs, Buttons, and MAX7219

The microcontroller acts as the central processing unit for the system, managing inputs and outputs. It provides power connections (+5V and GND) and has various GPIO pins to interact with external devices.

The buzzer is connected to a digital pin. It can be used to generate sound signals for alerts or notifications, driven by the Arduino.

MAX7219 is used to control an LED display. It receives serial data (DIN, CS, CLK) from the Arduino (pins D2, D3, D4) and manages segments and digits for driving the display (SEG A–G, DIG 0–7). The chip is powered by a 5V connection.

Design

To ensure a functional and compact setup for the bullet speed measurement system, we designed a custom hardware enclosure using AutoCAD. The design consists of a rectangular box that houses critical electronic components such as a buzzer, a battery, and an Arduino Nano. This enclosure ensures that the internal components are securely positioned and protected during operation.
Using AutoCAD, we created a detailed 3D model of the enclosure and support structures to visualize and optimize the layout. The design includes four vertical supports, two on each side, which serve as mounts for the light-dependent resistors (LDRs) and the corresponding laser modules. Each side features seven LDRs aligned vertically, positioned opposite the laser modules. This configuration allows the lasers to shine directly onto the LDRs, creating a grid for detecting the passing bullet.
The AutoCAD design incorporates precisely placed holes for wiring to maintain a clean and efficient layout. This approach ensures stability and accuracy, enabling reliable measurements during the experiment.

To manufacture the design, we utilized a 3D printer. However, for the purposes of the project, only the vertical sticks holding the LDRs and lasers were printed and implemented, while the rest of the enclosure design was not used. This choice streamlined the manufacturing process and focused on the critical components needed for the project.

A rendered image of the hardware design, generated from AutoCAD, is provided below for clarity Figure 7.6. The right-side rendering highlights the structural components and the internal housing.

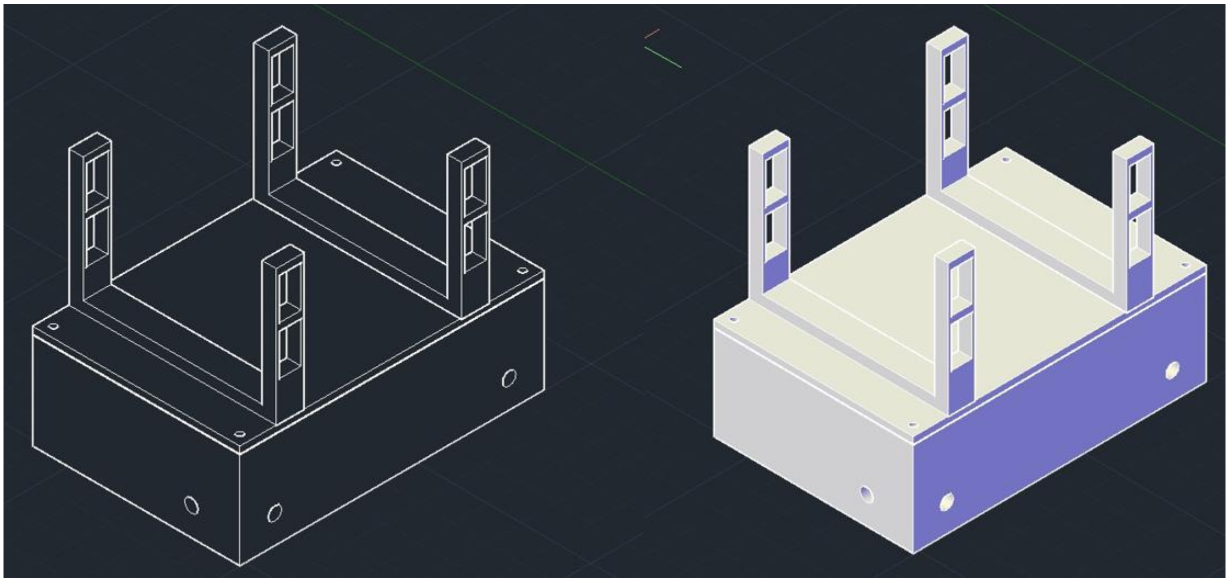

Figure 7.6-3D Rendered Design of the Hardware Enclosure and Support Structures Created in AutoCAD

5-SOFTWARE

The software developed for this project serves as the core of the system, seamlessly integrating hardware components and managing real-time operations. Designed using the Arduino IDE (ver. 2.3.4), the program enables the Arduino Nano Every to read data from sensors, process inputs, and provide outputs via a 7-segment display and buzzer. The software employs interrupts for precise timing and efficient detection of sensor events, allowing accurate speed calculations. Additionally, it includes logic to handle feedback mechanisms, such as auditory alerts, ensuring an interactive and user-friendly experience. The following sections detail the tools, methodologies, algorithms, and key functions implemented in the software to achieve the desired functionality.

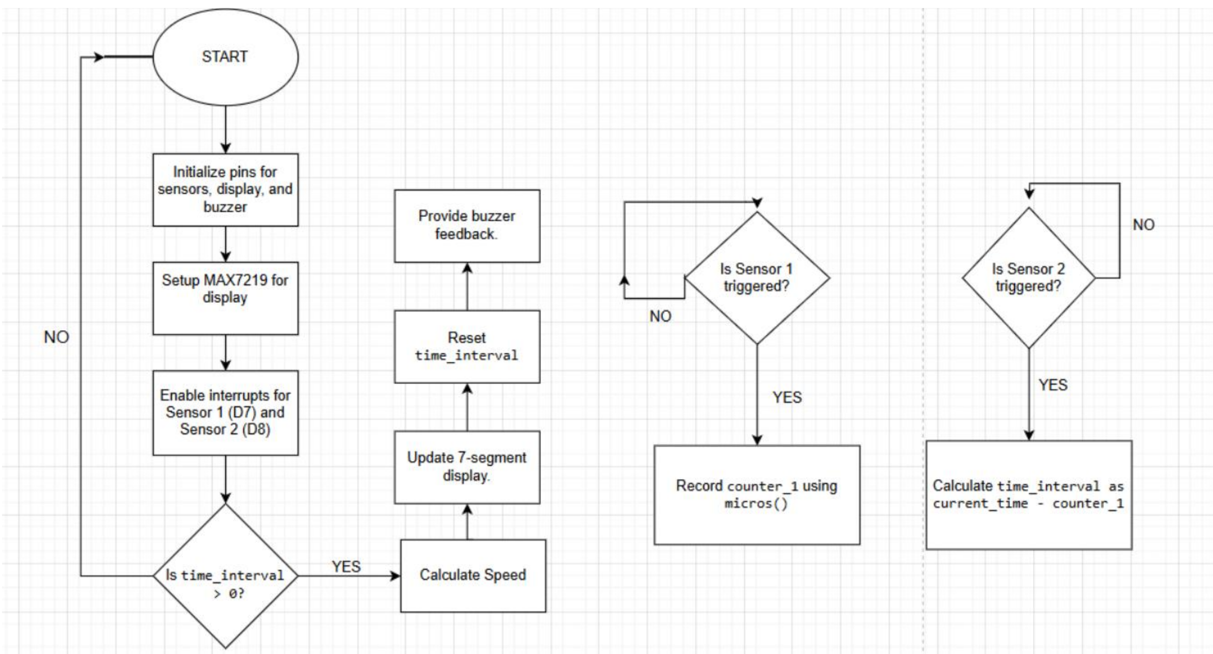

Figure 8-Algorithm Flowchart for Speed Calculation and Feedback System

Now, key highlights from the program will be discussed. The program begins by assigning variables to specific microcontroller pins to define their functionality, The pin modes are configured appropriately, with output mode enabled for components like the buzzer and display. Serial communication is initialized to enable debugging and data monitoring. Following the initialization, the main loop executes the core functionality of the program, including real-time sensor data acquisition, interrupt handling for precise timing, speed calculation, and output updates on the display. Below is a detailed explanation of the code structure and logic.

```
// Pins of the 7-segment display
#define MAX7219_Data_IN 2
#define MAX7219_Chip_Select 3
#define MAX7219_Clock 4

// Inputs/Outputs
int sensor_1 = 7; // Front sensor (D7)
int sensor_2 = 8; // Rear sensor (D8)
int Buzzer = 6;   // Buzzer

// Variables
float distance_between = 0.18; // Distance between sensors in meters
unsigned long counter_1, current_time, time_interval;
float Speed, last_speed;
```

Explanation:This section declares the pins for the hardware components and initializes variables for the program. The #define macros and int variables specify which pins of the microcontroller are connected to the sensors, MAX7219 (7-segment display), and buzzer.The variables store the time measurements (counter_1, current_time), time interval between sensor triggers (time_interval), and calculated speed (Speed). The distance_between specifies the fixed distance between the two sensors.

```
// Function to send bytes to the 7-segment display
void shift(byte send_to_address, byte send_this_data) {
  digitalWrite(MAX7219_Chip_Select, LOW);
  shiftOut(MAX7219_Data_IN, MAX7219_Clock, MSBFIRST, send_to_address);
  shiftOut(MAX7219_Data_IN, MAX7219_Clock, MSBFIRST, send_this_data);
  digitalWrite(MAX7219_Chip_Select, HIGH);
}
```

This function handles communication with the MAX7219 chip using the SPI-like shiftOut() function.

Parameters:

send_to_address: The register or digit to update.
send_this_data: The value to display on the selected digit.

The function sends two bytes (register and data) to the MAX7219, enabling control over each digit or feature of the 7-segment display.

```
void setup() {
  pinMode(sensor_1, INPUT);
  pinMode(sensor_2, INPUT);
  pinMode(Buzzer, OUTPUT);
  digitalWrite(Buzzer, LOW);

  // Enable Pin Change Interrupts for PORTD (D7) and PORTB (D8)
  PCICR |= (1 << PCIE2) | (1 << PCIE0); // Enable interrupts for PORTD and PORTB
  PCMSK2 |= (1 << PCINT23);             // Enable interrupt on D7 (Sensor 1)
  PCMSK0 |= (1 << PCINT0);              // Enable interrupt on D8 (Sensor 2)

  pinMode(MAX7219_Data_IN, OUTPUT);
  pinMode(MAX7219_Chip_Select, OUTPUT);
  pinMode(MAX7219_Clock, OUTPUT);
  digitalWrite(MAX7219_Clock, HIGH);
  delay(200);

  // Setup of MAX7219 chip
  shift(0x0f, 0x00); // Display test register - test mode off
  shift(0x0c, 0x01); // Shutdown register - normal operation
  shift(0x0b, 0x07); // Scan limit register - display digits 0 thru 7
  shift(0x0a, 0x0f); // Intensity register - max brightness
  shift(0x09, 0xff); // Decode mode register - CodeB decode all digits
  reset_screen();

  Serial.begin(19200); // Serial Monitor for debugging
  Serial.println("System Ready");
}
```

The setup() function begins by configuring the necessary pins for the sensors, buzzer, and MAX7219 display driver. The pins connected to the sensors (D7 and D8) are set as inputs to detect signals when triggered, while the pin for the buzzer (D6) is configured as an output to allow the microcontroller to activate or deactivate it. The buzzer is initially turned off by setting its pin to LOW. Interrupts are then enabled using the Pin Change Interrupt Control Register (PCICR) to monitor pin changes on PORTD (D7) and PORTB (D8). The Pin Change Mask Registers (PCMSK2 and PCMSK0) are configured to specifically enable interrupts for D7 and D8 by setting the corresponding bits (PCINT23 and PCINT0). Next, the pins required for the MAX7219 display driver (MAX7219_Data_IN, MAX7219_Chip_Select, and MAX7219_Clock) are set as outputs to facilitate communication. The clock pin is set to HIGH to ensure proper operation, and a short delay is introduced to allow the display driver to stabilize. The MAX7219 is then initialized with several commands to configure its operation, including turning off test mode, enabling normal operation, setting the scan limit for all digits, adjusting brightness to maximum, and enabling BCD decoding for all digits. Finally, the display is cleared using the reset_screen() function, and serial communication is initialized at a baud rate of 19200 for debugging and monitoring. A confirmation message, "System Ready," is printed to the Serial Monitor to indicate successful initialization.

```
void reset_screen() {
  for (int i = 1; i <= 8; i++) {
    shift(i, 0x00);
  }
}
```

This function clears all digits on the 7-segment display by sending 0x00 to each digit register. It ensures the display starts fresh for new data.

```
void loop() {

  Serial.print("Time Interval (microseconds): ");
  Serial.println(time_interval);

  if (time_interval > 0) { // Avoid division by zero
    Speed = distance_between / (time_interval / 1E6); // Calculate speed
    Serial.print("Speed (float): ");
    Serial.println(Speed);
    Serial.print("Speed (int): ");
    Serial.println(int(Speed * 100));

    // Update display
    int SpeedInt = int(Speed * 100);
    reset_screen();
    shift(0x03, SpeedInt / 100); // Hundreds
    shift(0x02, (SpeedInt % 100) / 10); // Tens
    shift(0x01, SpeedInt % 10); // Ones

    tone(Buzzer, 3-0, 500); // Optional feedback with buzzer
    time_interval = 0;       // Reset after processing
  }
}
```

The loop() function serves as the core operational logic for the program, continuously executing real-time tasks. It begins by printing the current value of the time_interval variable to the Serial Monitor for debugging purposes. If a valid time_interval greater than zero is detected, the program calculates the speed of the object using the formula.

The calculated speed, initially in float format, is converted into an integer format (scaled by 100) for easy display on the 7-segment display. The display is updated digit by digit, with the hundreds, tens, and ones places of the speed value sent to the corresponding registers of the MAX7219. To enhance user interaction, the buzzer is briefly activated to provide auditory feedback when a valid speed measurement is made. After processing, the time_interval is reset to zero to prepare for the next measurement cycle. This loop structure ensures that the system operates continuously, calculating and displaying speed measurements in real time while providing feedback to the user. Debugging information, including the speed in both float and integer formats, is also printed to the Serial Monitor for monitoring purposes.

```
// ISR for PCINT2 (for D7) and PCINT0 (for D8)
ISR(PCINT2_vect) {
  if (PIND & (1 << PIND7)) { // Sensor 1 (D7)
    current_time = micros();
    counter_1 = current_time;
    Serial.println("Sensor 1 triggered!");
  }
}

ISR(PCINT0_vect) {
  if (PINB & (1 << PINB0)) { // Sensor 2 (D8)
    current_time = micros();
    time_interval = current_time - counter_1;
    Serial.print("Sensor 2 triggered! Time Interval: ");
    Serial.println(time_interval);
  }
}
```

The interrupt service routines (ISRs) handle real-time detection of sensor triggers to ensure precise timing for speed calculation. Two ISRs are defined, one for each sensor:

The **ISR for Sensor 1 (PCINT2_vect)** monitors changes on pin D7, connected to the first sensor. When the sensor is triggered, the microcontroller checks the state of pin D7 using a bitwise operation (PIND & (1 << PIND7)). If the pin is high (indicating activation), the current time is recorded using the micros() function, which provides a high-resolution timestamp in microseconds. This timestamp is stored in the variable counter_1 to mark the time of the first sensor activation. A debug message, "Sensor 1 triggered!", is sent to the Serial Monitor for verification.

The **ISR for Sensor 2 (PCINT0_vect)** performs a similar function for the second sensor, connected to pin D8. When triggered, the ISR calculates the time interval between the two sensor events by subtracting the previously recorded timestamp (counter_1) from the current time, also obtained using micros(). The calculated time_interval is then stored for use in the main loop to

compute the speed. A debug message with the time interval value is printed to the Serial Monitor, indicating that the second sensor was triggered and the interval was successfully recorded.
These ISRs ensure accurate and efficient timing measurements by responding immediately to sensor events, which is critical for precise speed calculations.

6-Start-up, calibration

Before operating the bullet speed measurement system, we conducted a thorough start-up and calibration process to ensure accuracy and reliability. The system was powered on, and the Arduino Nano was connected to a stable power source. Upon initialization, the system displayed the "System Ready" message on the serial monitor, indicating successful start-up. Laser modules were carefully aligned with their corresponding LDR sensors using visual markers to ensure the beams directly hit the sensors. The sensitivity of the LDRs was fine-tuned using the potentiometers on the modules. To calibrate the system, we measured the exact distance between the two lasers using a ruler, which was determined to be 18 cm. This value was updated in the Arduino code to reflect the actual distance for accurate speed calculations. The calibration process was repeated until the measured speeds aligned closely with theoretical expectations, ensuring the system was ready for precise and reliable operation.

7-Test Measurements

To validate the accuracy of our bullet speed measurement system, we conducted a controlled experiment using a ping pong ball. The goal was to ensure that the display shows the correct speed by comparing the measured speed with the theoretically calculated speed.

The testing procedure involved releasing a ping pong ball from a height of 1 meter. To facilitate this test, the setup was changed from its horizontal orientation to a vertical alignment. This adjustment ensured that the ball could fall freely through the laser and LDR setup, enabling precise measurement during its descent.

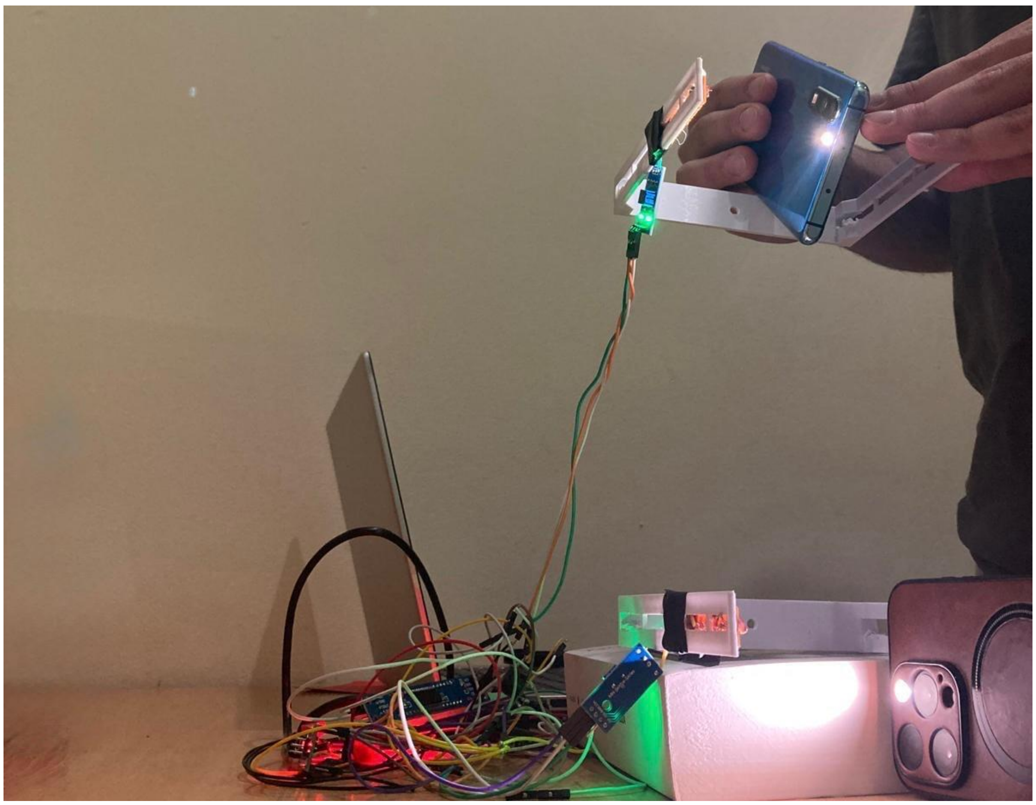

Figure 9- Testing set up vertically

The choice of this method is based on the principle of free fall, where the speed of the ball as it passes through the laser and LDR setup can be predicted using the equation of motion:

$$v = \sqrt{2gh} \tag{3}$$

Where:

- v is the velocity,
- g is the acceleration due to gravity (9.81 m/s^2)
- h is the height from which the ball is dropped (1 m)

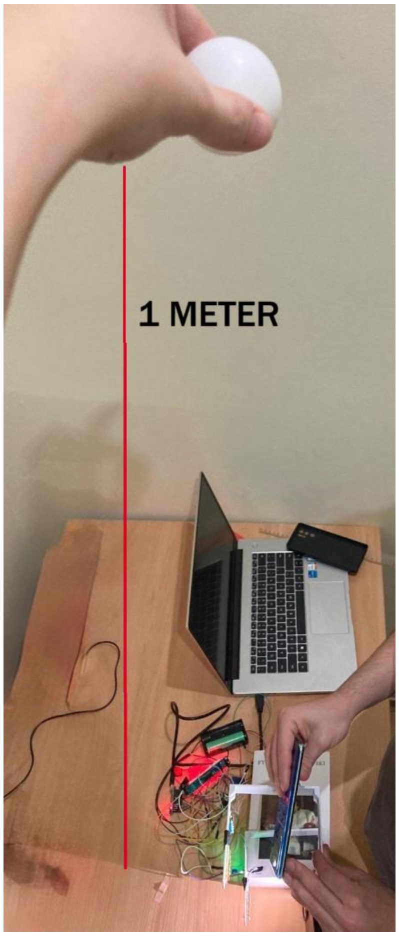

Figure 9.2-Free fall for testing

Substituting the values into the equation:

$$v = \sqrt{2x9.81m/s^2x1m} \tag{4}$$

$$v = \sqrt{19.62\ m^2/s^2} \tag{5}$$

$$v = 4.43\ m/s \tag{6}$$

The theoretical speed of the ping pong ball is 4.43 m/s as it passes through the laser and LDR setup.

After releasing the ball and observing the display, we recorded an initial measured speed of 4.22 m/s. To verify the consistency of the system, we conducted five additional trials.

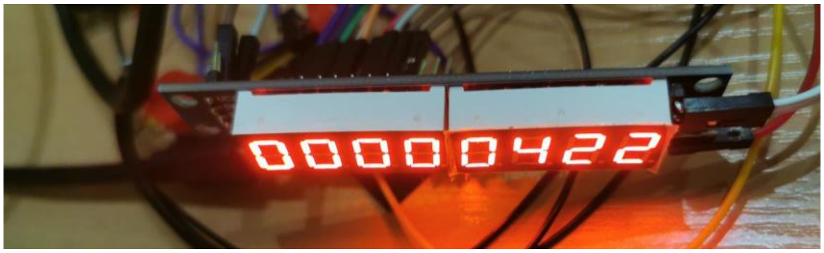

Figure 9.3- Display Showing Measured Speed During First Trial

The table below summarizes the speeds measured during five testing trials. Each trial involved releasing a ping pong ball from a height of 1 meter and recording the displayed speed. The theoretical speed was calculated as 4.43 m/s and the measured values closely align with this, demonstrating the accuracy of the system.

Tab 2. Measured Speeds During Testing Trials

Trial No	Measured Speed (m/s)
1	4.22
2	4.39
3	4.41
4	4.27
5	4.32

To evaluate the accuracy of the bullet speed measurement system, we calculated the error rate for each trial by comparing the measured speed with theoretical speed.

Tab 3. Error rate calculations

Trial No.	Error (%)
1	4.74%
2	0.90%
3	0.45%
4	3.61%
5	2.48%

The error for each trial is calculated using the formula:

$$Error\ Rate = \left|\frac{Measured\ Speed - Theoretical\ Speed}{Theoretical\ Speed}\right| \quad (7)$$

The average error rate across all trials is calculated as:

$$Average\ Error\ Rate = \frac{Sum\ of\ Individual\ Error\ Rates}{Number\ of\ trials} \quad (8)$$

$$Average\ Error\ Rate = \frac{4.74+0.90+0.45+3.61+2.48}{5} = 2.44\% \quad (9)$$

$$Success\ Rate = 100\% - 2.44\% = 97.56\% \tag{10}$$

The system achieved a high accuracy with an average error rate of 2.44% and an overall success rate of 97.56%.These results demonstrate that the device is highly reliable and accurate for speed measurement, with minimal deviations from the theoretical speed. This analysis validates the effectiveness of the system, while also highlighting areas for further refinement to reduce the error rate even further.

8-User Manual

Hello! Thank you for purchasing SpeedSense 1.0. We appreciate your support and are excited to provide you with a reliable speed measurement device for your experiments.
SpeedSense 1.0 is designed to measure the speed of objects such as bullets or lightweight projectiles, using an advanced laser and LDR-based setup. The system provides accurate measurements and displays the results in real-time.

Additional Items in the Package

- 1x USB Cable
- 1x SpeedSense Frame with pre-mounted LDRs and laser modules
- 1x Arduino Nano pre-configured with SpeedSense software
- 1x Operation Manual (this document)

Instructions for Setup and Operation

Please follow these instructions carefully to ensure proper functionality of the device:

1. Assemble the Frame:
    - Position the SpeedSense frame vertically or horizontally, depending on the object you plan to measure.
    - Ensure the LDR modules and laser modules are aligned correctly on opposite sides of the frame.
2. Connect the Device:
    - Connect the SpeedSense device to your computer using the provided USB cable.
    - Ensure the device is powered on and the Arduino Nano is securely connected.
3. Install the Software:
    - Locate the file named “SpeedSense.m” on the included software CD.
    - Install the software on your computer by following the on-screen instructions.
4. Configure the System:
    - Open the SpeedSense software.
    - Calibrate the laser alignment using the built-in testing tool in the software.
    - Set the distance and project type for accurate measurements.
5. Start Measurement:
    - Release the object (e.g., a ball or bullet) so it passes through the laser and LDR grid.
    - Observe the real-time speed displayed on the device’s screen.

Safety and Maintenance Tips

1. Avoid Direct Laser Exposure: Do not stare directly into the lasers.
2. Keep the Frame Stable: Ensure the frame is fixed firmly to avoid misalignment.
3. Clean the Sensors Regularly: Use a soft cloth to clean the LDR modules and lasers for accurate readings.
4. Store Properly: After use, disconnect the device and store it in a dry, safe place

If you encounter any issues, refer to the table below:

Issue	Solution
Device not powering on	Ensure the USB cable is securely connected and the computer is powered.
Speed readings are inaccurate	Check laser alignment and ensure no obstructions in the frame.
Software not responding	Reinstall the SpeedSense software and restart your computer.

Contact Us

For any further questions or technical support, please reach out to us at:
Email: support@speedsense.com
Phone: +1-800-123-4567
Thank you for choosing **SpeedSense 1.0**!

9 -Summary

The main idea of the project was creating a speed measurement system based on optoelectronic elements. The elements we implemented included Light Dependent Resistors (LDRs), lasers, and microcontroller-based processing, while carefully considering their characteristics and behaviors. The prototype we created worked effectively, which assures us that the theoretical knowledge we applied was accurate, and our implementation was successful.

The project presented several challenges, such as ensuring precise alignment of lasers and LDRs, programming the Arduino for accurate data processing, and integrating hardware components for reliable operation. It must be noted that this project is a simple prototype, and for applications requiring higher precision or robustness, further refinements would be necessary. For example, enhancing the alignment mechanism and reducing external environmental interferences could improve accuracy.

Initially, we aspired to make the system more versatile by including wireless data transmission for speed readings. However, this goal proved to be more complex than anticipated within the given timeframe. We believe that with more time and resources, this feature could be implemented successfully in future iterations.

There were also unexpected adjustments required, such as reworking the 3D-printed components to fit the design better, as some parts were not printed to the specified measurements. Despite these hurdles, the team collaborated effectively, maintaining a shared vision and completing the project on time.

Overall, the project demonstrated a practical and functional prototype that could serve as the foundation for further development in speed measurement systems for educational, research, or industrial applications.

10-Appendix

```
int sen1 = A0;
int sen2 = A3;
int ledPin = 9;
unsigned long t1 = 0;
unsigned long t2 = 0;
float velocity;
float velocity_real;
float timeFirst;
float timeScnd;
float diff;
float speedConst = 7.5;  // in cm
void setup() {
  Serial.begin(9600);
  pinMode(sen1, INPUT);
  pinMode(sen2, INPUT);
  analogWrite(11, LOW);
  analogWrite(10, HIGH);
}
void loop() {
  if (analogRead(sen1) < 500 && analogRead(sen2) > 500) {
    timeFirst = millis();
    digitalWrite(ledPin, LOW);
    delay(30);
  }
   if (analogRead(sen2) > 500 && analogRead(sen1) < 500) {
    timeScnd = millis();
    diff = timeScnd - timeFirst;
    velocity = speedConst / diff;
    velocity_real = (velocity * 360) / 100; // milliseconds to hours and centimeters to kilometers
    delay(30);
    digitalWrite(ledPin, HIGH);
    Serial.print("The velocity is: ");
    Serial.println(velocity_real);
    Serial.print(" km/hr ");
    delay(500);
    digitalWrite(ledPin, LOW);
```

11- Bibliography

1. YaSh. *Arduino Speed Detector.* Retrieved from https://projecthub.arduino.cc/yashastronomy/arduino-speed-detector-55410d
2. Build Electronic Circuits. *Laser Diode – What is it and how does it work?* Retrieved from https://www.build-electronic-circuits.com/laser-diode/
3.Reddit. *Difference Between LD and PD.* Retrieved from https://www.reddit.com/r/Optics/comments/riwm7o/difference_between_ld_and_pd/?rdt=51463
4. Hackaday. *Phototransistors, Transimpedance Amplifiers, and Response Times.* Retrieved from https://hackaday.io/project/178232-lalboard-ergonomic-keyboard/log/192192-phototransistors-transimpedance-amplifiers-and-response-times-oh-my
5.Electronics For You. *LDR (Light Dependent Resistors) Basics.* Retrieved from https://www.electronicsforu.com/technology-trends/learn-electronics/ldr-light-dependent-resistors-basics

6.Kumar, D., & Gaur, A. *High-Speed Projectile Recording System.* International Journal of Scientific & Engineering Research, 2012. Retrieved from https://www.ijser.org/researchpaper/High-Speed-Projectile-Recording-System.pdf