IT & BUSINESS COLLEGE

DEPARTMENT: COMPUTER SCIENCE

PROJECT TITLE: 7-SEGMENT WITH BUZZER MODEL

GROUP NAME: VANTERRANPAL

Members:

Bolotov Beknazar - 238715071

Satiev Atai - 238715061

Rakhmatillaev Adilet – 238715055

Abdullaev Mardonbek

Group: SCA-23B

Instructor: Nurbekov Mirlan

Table of Contents

List of figures

List of tables

Introduction

Today, issues of urban accessibility are becoming increasingly important, especially when it comes to helping people with visual impairments. One of the key problems faced by blind and visually impaired people at bus stops is the inability to identify the number of the arriving route. Visual information is unavailable, which means that people must rely on the help of others or act at random. To demonstrate how even simple electronic devices can alleviate this situation, a small group of students decided to create a simplified model of an audio notification system capable of indicating the number of the arriving vehicle using a sequence of sound signals.

An Arduino microcontroller was chosen for the project as a convenient and intuitive platform for learning the basic principles of automation. The main goal was to show how basic electronic components such as a seven-segment indicator, buzzer, resistors, and connecting wires can work together to form a clear and reproducible signal. In the prototype, the seven-segment display generates a random number from 0 to 10, which in this educational project represents the bus number. After that, the connected buzzer emits a series of short sounds, the number of which corresponds exactly to the displayed number. Thus, a person who cannot see the display can “hear” the value shown on the indicator.

The operation of such a system is based on two key elements: a module that generates a numerical value and a sound execution component that transmits it to the user. The general structure of the device is shown in Figure 1, which clearly shows how the microcontroller receives a random value, outputs it to the indicator, and then triggers the corresponding number of sound signals. The developed prototype demonstrates the basic principle of simple sound indication and provides a better understanding of how such devices can be used to improve accessibility in real-life situations, even in their most simplified form.

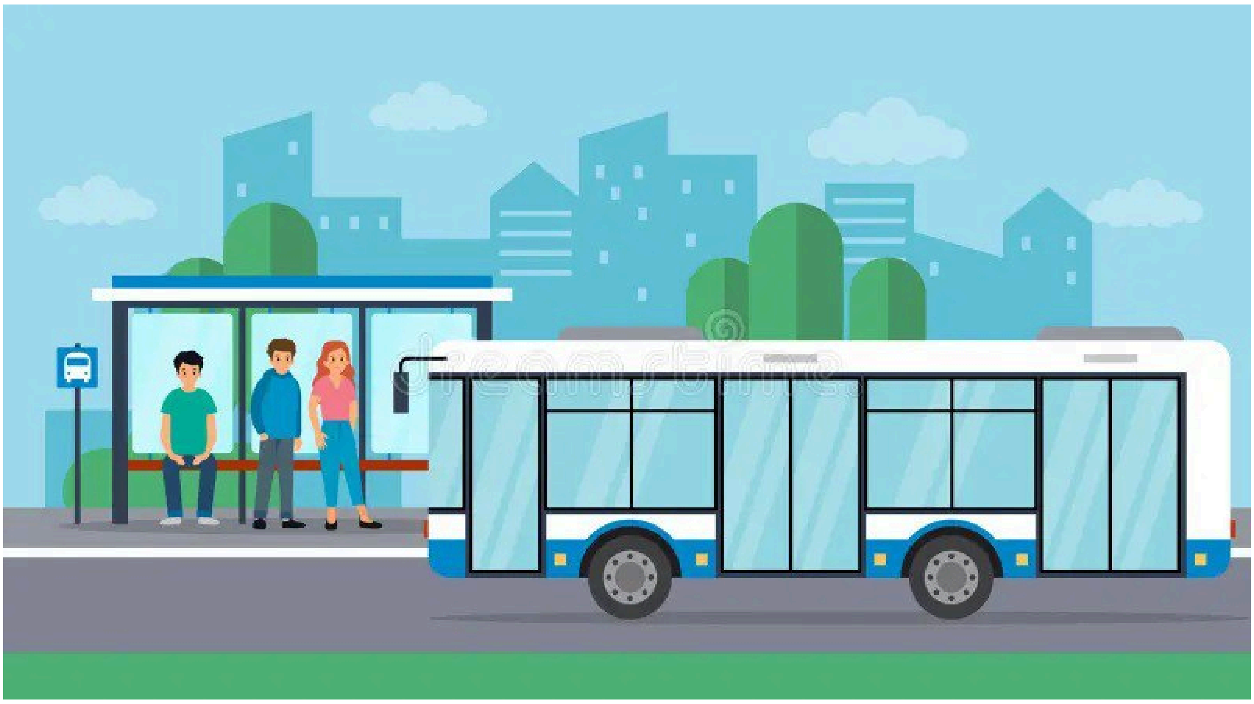

Figure 1

Motivation

For this project, we kind of just picked something we knew we could handle. We weren't trying to invent anything big or complicated, and honestly, we just needed a project that wouldn't take forever to understand. That's how we ended up with the 7 - segment display and the buzzer. We all saw these parts before on different devices, so it didn't feel scary to work with them.

At the start it looked super easy, but when we plugged everything in, things didn't go the way we expected. The display sometimes showed strange lines instead of numbers, and the buzzer just stayed silent even though we were sure we connected it right. So, we kept rechecking wires, fixing weird mistakes, rewriting tiny parts of the code, and just trying things until it finally worked. It wasn't clean work at all, but that's pretty much how we learned what each part was doing.

We didn't have a perfect plan or anything. One of us kept moving wires around. someone else just typed code and hoped it compiled, and another person tested the display by turning things on and off again. It was kind of chaotic, but we understood more from this messy process than from just reading theory.

So really our motivation was just to get some hands - on practice and understand how basic components react when we control them. We wanted a project that shows results right away - like a number.

Lighting up or a beep - because that helps us see if we're doing things right. It's simple, but it helped us get more comfortable before trying bigger projects.

Abstract

This project presents a compact assistive system for individuals with visual impairments. It was developed to provide an alternative method for perceiving numerical information. The system uses an arduino uno, a 7 - segment display, and a buzzer to show a randomly generated number (0-10) visually and through corresponding buzzer beeps. Users can identify numbers without relying on sight. Testing confirmed that the system reliably delivers accurate visual and audio feedback, demonstrating the potential of simple electronics for accessible applications.

Benefits of such a system

1. **Easy to understand** - The 7 - segment display shows numbers clearly, and the buzzer gives immediate audio feedback. This makes it simple to see what the system is doing, even for beginners.
2. **Hands-on learning** - Working with both the display and the buzzer helps understand basic electronics, like wiring, voltage, and controlling components with a microcontroller. It's a good way to practice without complicated circuits.
3. **Interactive output** - The combination of visual and sound signals makes the project interactive. You can see numbers change and hear alerts, which makes testing and debugging more engaging.
4. **Practical applications** - This system is the basis for real life devices like timers, counters, alarms, or simple notification systems. It shows how simple components are used in everyday electronics.
5. **Confidence building** - Because the results are immediate, it helps build confidence in coding and wiring. When the number lights up correctly and the buzzer works, it's easy to understand the cause and effect in electronics.
6. **Teamwork-friendly** - The project can be divided into tasks: one person works on wiring, another on coding, someone else on testing. This helps practice collaboration while learning practical skills.

Problem statement

In modern urban environments, people with visual impairments face serious difficulties when using public transport. At bus stops, passengers need to visually identify the number of the approaching bus, which is impossible for people who cannot rely on their sight. Traditional orientation methods offer limited assistance, often requiring help from others or putting the person at risk of missing the bus.

Despite growing attention to inclusive technologies, there is still a lack of simple, accessible, and easy to understand systems that could indicate the number of the arriving vehicle with an audible signal. This creates a clear need for a small prototype that can demonstrate how microcontroller based logic and audible alerts can be used to support people with visual impairments in everyday transportation situations.

Objectives:

1) Develop a simplified assistive system using an Arduino microcontroller that helps people with visual impairments to identify approaching buses.

2) Integrate a seven-segment display capable of generating a random bus number for demonstration purposes.

Desired Needs

1. **Clear display of information** - The system should show numbers or simple data clearly on the 7 - segment display so anyone can understand it easily.

2. **Reliable audio feedback** - The buzzer should work consistently to alert or signal actions, making the system more interactive and easier to test.

3. **Simple wiring and setup** - The connections between the microcontroller, display, and buzzer should be straightforward, so beginners can assemble the project without too much difficulty.

4. **Ease of programming** - The code should be simple enough to modify and understand, allowing anyone to experiment with numbers, timings, or buzzer patterns.

5. **Immediate response** - Both display and buzzer should react right away to changes in input or code, so testing and troubleshooting are easier.

6. **Expandable system** - The project should allow adding extra components later, like buttons, sensors, or more displays, without needing to redesign everything from scratch.

7. **Team-friendly tasks** - The system should allow dividing work among group members, like one person's wiring, another coding, and another testing outputs, so learning happens together.

System requirements & deliverables

The system is designed to generate numerical values and convert them into audible signals, enabling visually impaired users to identify the represented number.

4.1 Requirements

1. **Numerical output** - The system must generate numbers accurately and display them on the 7-segment display.

2. **Audible feedback** - The system must convert the numbers into audible signals using a buzzer to ensure accessibility for visually impaired users.

3. **Component utilization** - The system must use the components listed in Table 1.

4. **Indoor operation** - The system should function reliably in an indoor environment.

5. **Simple sound patterns** - The system must produce distinguishable sound pulses corresponding to the numerical values.

Specifications

The system specifications define its technical details and operational limits:

1. **7 - Segment display**
    - o Type: Common cathode/anode
    - o Displays: Single-digit numbers (0-9)
    - o Input: Digital signals from Arduino pins
2. **Buzzer**
    - o Type: Buzzer Model
    - o Output: Short beeps corresponding to numerical values
    - o Frequency: ~2 kHz (adjustable in code)
3. **Microcontroller (Arduino uno)**
    - o Operating Voltage: 5V
    - o I/O Pins: Digital pins for controlling display and buzzer
    - o Programming: Arduino IDE
4. **Power supply**
    - o Voltage: 5V DC
    - o Source: USB or battery
5. **Prototype**
    - o Platform: Breadboard assembly
    - o Environment: Indoor use only

4.2 Deliverables

The project deliverables include:

1. **Functional 7 - segment display** - Correctly shows generated numbers.
2. **Buzzer output** - Converts numbers into audible beeps for identification.f
3. **Prototype hardware** - Arduino-based indoor prototype with wiring, display, and buzzer.
4. **Source code** - Arduino code for number generation, display control, and buzzer signaling.
5. **Demonstration** - A working demo showing how numbers are displayed and sounded for users.
6. **Documentation** - Project report including system design, requirements, specifications, and testing results.

No.	Component	Quantity	Description
1	Arduino Uno	1	Microcontroller for control
2	7-Segment Display	1	Visual number output
3	Buzzer	1	Audible feedback
4	Breadboard	1	Prototype assembly
5	Jumper Wires	As needed	Connections between components

Table 1

4.2.1 Application

The system has several practical applications, especially for **learning and basic accessibility projects**:

1. **Educational tool** - It helps beginners understand how microcontrollers, displays, and buzzers work together in a simple system.
2. **Accessibility aid** - The buzzer provides audio feedback so visually impaired users can identify numbers without relying on sight.
3. **Timers and counters -** This setup can be used as a basic counter, timer, or numeric indicator for small devices.
4. **Prototyping base -** The project can serve as a foundation for more advanced systems, such as interactive displays, alarms, or notifications.
5. **Team learning** - The project allows multiple users to work together on wiring, coding, and testing components.

Algorithm

The algorithm describes how the system works step by step:

1. **Start the system** - Power up the Arduino and initialize the pins for the 7 - segment display and buzzer.
2. **Generate number** - Create a random number between 0 and 9 (or receive input if extended).
3. **Display number** - Send the appropriate signals to the 7 - segment display to show the generated number.
4. **Convert to sound** - Use the buzzer to generate a series of short beeps corresponding to the number. For example, if the number is 3, the buzzer beeps three times.
5. **Pause** - Add a small delay to allow the user to read or hear the output.
6. **Repeat** - Loop back to generate the next number and repeat steps 3-5 continuously.

Constraints

6.1 Technical constraints

1. **7-segment display limitations** - Can only display single-digit numbers (0-9) Can not show letters or complex symbols.
2. **Buzzer output** - Produces only short beeps. Can not generate complex sounds or multiple tones simultaneously.
3. **Power supply** - Requires stable 5V DC. fluctuations may cause the display or buzzer to malfunction.
4. **Microcontroller capacity** - Arduino has limited I/O pins. Adding more components may require additional circuitry.
5. **Prototype platform** - Breadboard and jumper wires are not durable for long-term use or outdoor environments.
6. **Random number generation** - Without additional input, the system cannot represent real time data like actual bus routes.

6.2 Non-technical constraints

1. **Indoor usage** - The system is designed for indoor use only; exposure to moisture, dust, or extreme temperatures can damage components.
2. **Accessibility limitation** - While the buzzer helps visually impaired users, the 7-segment display alone cannot be accessed without sight.
3. **Noise interference** - The buzzer may be hard to hear in noisy environments.
4. **User knowledge requirement** - Users need a basic understanding of electronics to assemble and troubleshoot the prototype.
5. **Time and resources** - Limited budget and time may restrict adding advanced features or more components.

List of Figures

7.1 - 7-segment display module

This figure illustrates the 7-segment display module used in the project. It shows the individual LED segments (A-G) and the decimal point, all arranged to form numeric characters. The diagram highlights the pin configuration, segment labeling, and electrical connections that allow the display to show digits from 0 to 9 based on the input signals.

7.2 - Buzzer module

This figure presents the buzzer module integrated into the system. It includes the internal structure of the buzzer, its two-pin or three-pin interface, and the signal input mechanism that produces sound. The diagram demonstrates how the buzzer is activated through electrical pulses to generate alerts or notifications within the project.

Components

The system has several limitations:

1. 7 - segment display visibility. The display still requires sight to read, so without the buzzer it does not provide accessibility for visually impaired users.

2. Buzzer output - The buzzer produces short beeps that may be difficult to hear in noisy outdoor environments, limiting practical use.

3. Random number generation - The prototype generates numbers randomly and therefore cannot represent real bus routes without receiving actual input data.

4. Indoor prototype - The setup is intended for indoor use. Breadboards, wires, and Arduino components are not protected against moisture, dust, or temperature changes.

5. Limited sound patterns - The buzzer can only produce a fixed type of sound (short beeps), so the system cannot encode more complex information beyond countable pulses.

You can see the components and their purpose below in Table 1.

No.	Component	Quantity	Description
1	Arduino Uno	1	Microcontroller for control
2	7-Segment Display	1	Visual number output
3	Buzzer	1	Audible feedback
4	Breadboard	1	Prototype assembly
5	Jumper Wires	As needed	Connections between components

Table 2

System design (hardware and software)

The assistive audio indication system is designed as an educational prototype that demonstrates how a combination of basic hardware components and simple software logic can be used to support users with visual impairments. The system integrates a microcontroller, a numerical display, and an audio output device to provide information through both visual and auditory channels.

From the hardware perspective, the core of the system is the Arduino Uno microcontroller, which controls all connected components and coordinates their interaction. A 7-segment display is used to visually present numerical information, while a buzzer serves as the audio output device. These components are chosen due to their simplicity, low cost, and suitability for educational and prototyping purposes. The wiring is kept minimal to ensure reliability and ease of understanding for beginners.

On the software side, the Arduino is programmed to generate a random numerical value within a predefined range. This number is then displayed on the 7-segment display, allowing sighted users or developers to verify system operation. At the same time, the microcontroller activates the buzzer to produce a sequence of audio signals. Each beep represents a single unit of the generated number, enabling visually impaired users to determine the value by counting the sound pulses. Short delays between beeps are implemented to ensure clarity and prevent signal overlap.

The system workflow follows a clear and repetitive cycle: number generation, visual display, audio conversion, and reset. This structure makes the system easy to understand, modify, and expand. For example, future improvements could include variable sound patterns, voice output, or user input controls. Overall, this prototype demonstrates how simple hardware and software integration can be effectively applied to create accessible assistive technologies.

Component selection

7-segment display (Figure 2):

- Displays a single digit, representing the “bus number” in this prototype.
- Simple and widely used visual numeric indicator.
- Compatible with Arduino through direct pin control.
- Can represent values from 0 to 9 (or 0-10 in software simulation).

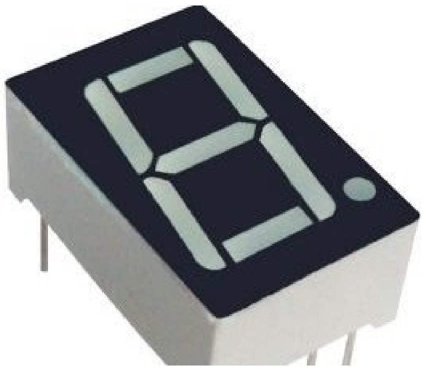

Figure 2. 7-Segment Display Module

Buzzer module (Figure 3):

- Produces short audible beeps for numerical representation.
- Low power consumption and easy integration with Arduino.
- Generates distinct sound pulses that can be counted by the user.
- Suitable for simple alert or indicator systems.

Figure 3. Buzzer module

Arduino uno board (Figure 4):

- Central microcontrollers responsible for generating random numbers.
- Controls both the 7-segment display and the buzzer.
- Widely used in educational electronics due to simplicity and reliability.
- Provides stable power and programming using USB.

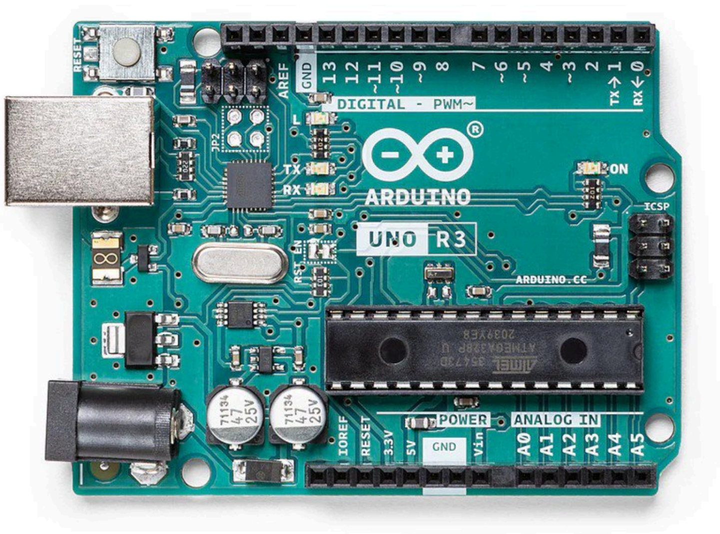

Figure 4. Arduino Uno Board

Breadboard & Jumper Wires (Figure 5):

- Used for assembling the circuit without soldering.
- Provides flexible wiring for the 7-segment display, resistors, and buzzer.
- Ideal for prototyping and testing educational electronics systems.

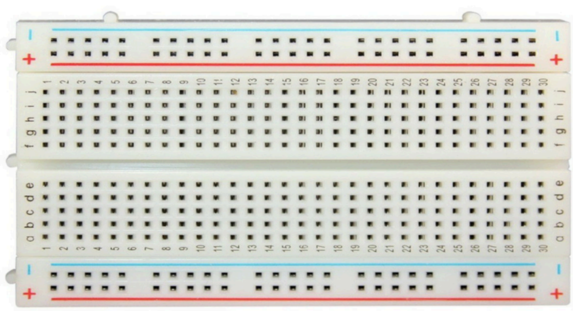

Figure 5. Breadboard

Prototype versions and progress

Prototype 0000:

At the earliest stage of development, a simple test setup was assembled to verify the basic operation of the buzzer and button using the Arduino Uno. The objective of this prototype was to ensure that the system could successfully detect a button press and generate an audible signal in response. This initial functionality is essential, as it forms the foundation for the later logic where the buzzer will need to produce a specific number of beeps corresponding to the displayed bus number.

In this prototype, a push button was connected to a digital input pin of the Arduino, while the buzzer was connected to an output pin. When the button was pressed, the Arduino activated the buzzer, confirming that the device could correctly interpret the input and trigger the sound output. This basic response validated the core interaction between the components and demonstrated that the buzzer functions reliably under simple control.

The test version of this prototype is shown in Figure 6.

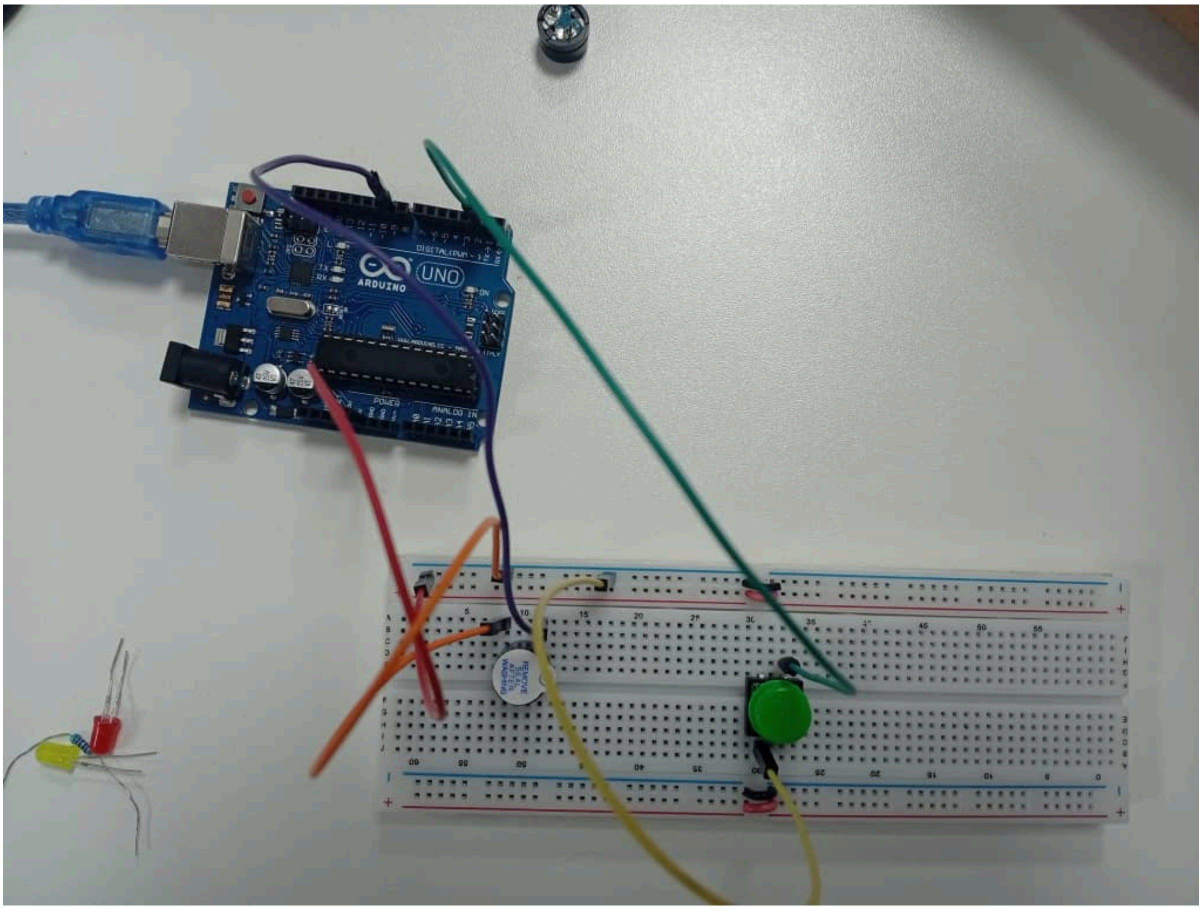

Figure 6. Prototype 0000

Prototype 0001 description

Prototype 0001 is an early functional model designed to demonstrate the basic interaction between three core components: a button, an LED, and a buzzer. The purpose of this prototype is to validate the input–output behavior of a simple user-triggered signaling system.

The prototype operates using a straightforward mechanism. When the user presses the button, the system begins with a counting sequence from 0 to 10. During this sequence, the LED blinks once for each count, resulting in a total of 10 LED flashes by the end of the cycle. Each time the LED lights up, the buzzer simultaneously produces a sound, ensuring that the number of audible beeps matches the number of LED flashes.

You can see the prototype below in Figure 7.

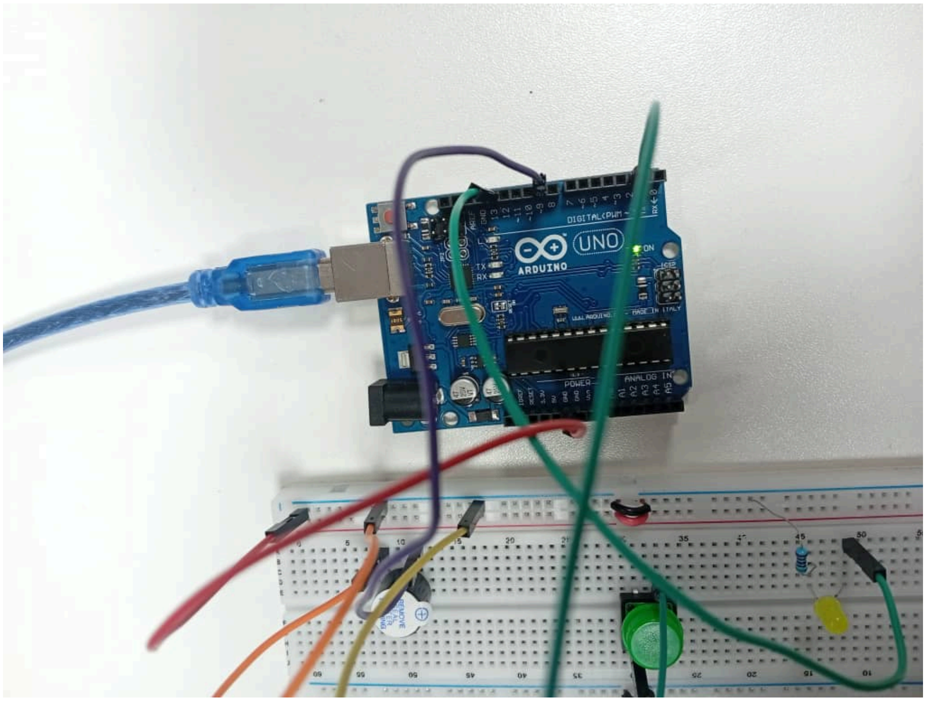

Figure 7. Prototype 0001

Final Prototype

The Final Prototype represents the completed functional system integrating a 7-segment display, buzzer, and microcontroller to assist visually impaired users by providing both visual and audible numerical output.

In this prototype, the system automatically generates a random number from 0 to 10. Once the number is selected, it is immediately shown on the 7-segment display, allowing sighted users to see the result visually. To ensure accessibility for visually impaired users, the buzzer emits a sequence of beeps equal to the number displayed. For example, if the 7-segment shows “4”, the buzzer will produce four beeps.

This dual-output system creates synchronized feedback:

- **Visual output:** A numerical digit from 0–10 shown on the 7-segment display
- **Audio output:** A corresponding number of beeps produced by the buzzer

The final prototype demonstrates full functionality of the core concept, including:

- Random number generation using the microcontroller
- Digital output handling for 7-segment display control
- Timed audio signaling through the buzzer
- Synchronization between displayed number and audible output

This prototype illustrates how the system can assist visually impaired users in identifying numerical values through sound patterns and confirms successful integration of hardware and software components.

You can see it in figure 8 below.

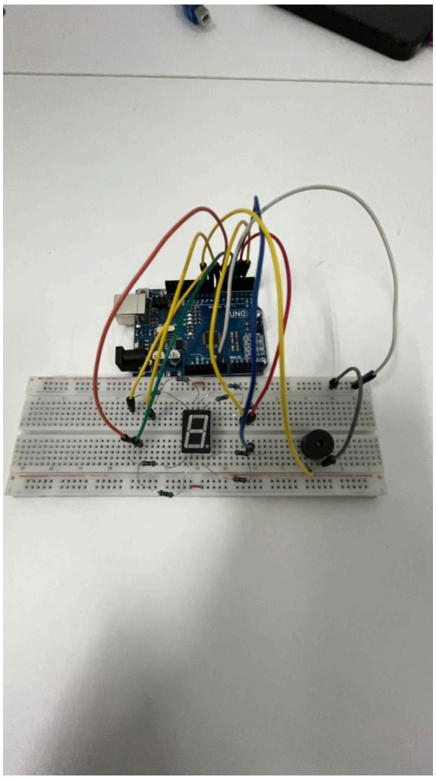

Figure 8. Final Prototype

Implementation

This section describes how the system was developed, including both the programming stage and the physical assembly of the hardware. The goal was to create a functional prototype capable of generating a random number, displaying it on a 7-segment module, and producing an equivalent number of audible beeps through a buzzer.

Code implementation
The program is designed to demonstrate the coordinated operation of a 7-segment display and a buzzer by repeatedly showing random digits and producing corresponding audio signals. Its functionality is organized into several key components:
1. Random number generation
The microcontroller generates a random integer from 0 to 9 using random(0, 10). To enhance randomness, the program seeds the generator with an analog reading from an unused input pin.
2. Display control using a segment table
A lookup table (digits) defines which of the seven display segments must be lit to represent each digit from 0 to 9.
The showDigit() function reads the table and sets the output pins (segA-segG) accordingly, ensuring the correct visual representation of each number on the common-cathode 7-segment display.
3. Audio signaling with a buzzer
For each unit of the generated number, the buzzer produces a short beep:
- A 1000 Hz tone is activated for 200 ms,
- Followed by 200 ms of silence.
  This results in a sequence of audio pulses whose count matches the digit shown on the display.

4. Synchronized feedback
The program ensures that the display updates first, followed immediately by the beeping sequence. This provides users with consistent and synchronized visual and auditory feedback for every randomly generated value.
5. Continuous loop execution
All operations run inside the loop() function. After completing the display and beep sequence, the program waits 2 seconds before repeating the cycle, creating a continuous demonstration useful for testing or presentation.
6. Program organization
The code uses clear structure, modular functions, and a segment-mapping table, making it readable and easy to modify for future expansion such as adding multi-digit displays, varying beep patterns, or introducing user input.

You can see the code below in Figure 9.

```
// ----- Функция вывода цифры -----
void showDigit(int n) {
  digitalWrite(segA, digits[n][0]);
  digitalWrite(segB, digits[n][1]);
  digitalWrite(segC, digits[n][2]);
  digitalWrite(segD, digits[n][3]);
  digitalWrite(segE, digits[n][4]);
  digitalWrite(segF, digits[n][5]);
  digitalWrite(segG, digits[n][6]);
}

void setup() {
  // Устанавливаем пины сегментов как выходы
  pinMode(segA, OUTPUT);
  pinMode(segB, OUTPUT);
  pinMode(segC, OUTPUT);
  pinMode(segD, OUTPUT);
  pinMode(segE, OUTPUT);
  pinMode(segF, OUTPUT);
  pinMode(segG, OUTPUT);

  pinMode(buzzer, OUTPUT);

  randomSeed(analogRead(A0)); // Для случайности
}

void loop() {
  int number = random(0, 10); // Рандом от 0 до 9 (10 невозможна на индикаторе)

  showDigit(number); // Вывод цифры

  // Сигналы бузяка
  for (int i = 0; i < number; i++) {
    tone(buzzer, 1000);
    delay(200);
    noTone(buzzer);
    delay(200);
  }

  delay(2000); // Пауза перед следующим числом
}
```

Figure 9. Code

The most essential parts of the code are:

1) The segment table and the digit display function. This is the whole logic that controls which segments of the 7-segment display should light up (volid showDigit()).

2) Initialization of pins and random generator. All segment pins and the buzzer pin are configured as outputs and randomness is initialized.

3) Buzzer audio signaling. The buzzer beeps as many times as 7-segment shows.(volid loop()).

4) Delay before repeating the cycle. Adds a short pause for clear visual and audio understanding.(delay(2000))

Building the Project

The physical assembly required connecting and configuring all components on a breadboard. The main steps in constructing the prototype were:

1. **Microcontroller setup:**
   The Arduino board was positioned as the central control unit, supplying digital output signals to both the 7-segment display and buzzer.
2. **7-segment display wiring:**
   Each segment (A-G) was linked to a designated Arduino pin using jumper wires. Resistors were added where necessary to limit current and protect the LEDs inside the display.
3. **Buzzer integration:**
   The buzzer was connected to a digital pin, with the ground tied to the common GND rail. A series resistor or transistor driver can be used depending on buzzer type.
4. **Power distribution:**
   The breadboard's power rails were connected to the Arduino's 5V and GND outputs to ensure consistent voltage across all components.
5. **Structural arrangement:**
   All components including the display, buzzer, and supporting wires were positioned in a compact layout to maintain stability while allowing easy access for testing.
6. **Testing and calibration:**
   After wiring, the system was powered and the code uploaded. Each number from 0 to 10 was checked individually to verify:
   - Correct segment illumination
   - Accurate beep count
   - Stable operation with no wiring conflicts

The final assembly produced a working model capable of demonstrating both visual and audible numeric feedback, supporting the intended use case for visually impaired users.

Testing and Results

The system was tested by checking each component individually and then evaluating the full prototype. The 7-segment display was verified by showing every number from 0 to 10 to ensure correct segment illumination. The buzzer was tested by generating simple beep sequences to confirm it responded properly to digital pulses. Random number generation was also checked and consistently produced values within the required range.

When the full system was operated, each randomly displayed number matched the exact number of beeps produced by the buzzer. The system ran reliably over multiple test cycles, and no major errors or inconsistencies were observed. The results show that the prototype performs as intended, providing accurate visual and audible numerical output.

Conclusion

The project successfully demonstrated a simple yet effective system capable of presenting numerical information through both visual and audible outputs. The 7-segment display accurately showed the generated numbers, while the buzzer produced corresponding beep counts, allowing the information to be understood both visually and audibly. This dual-mode output ensures that the system is accessible to users with visual impairments, highlighting the potential of combining basic electronic components to create practical assistive technology solutions.

During the development and testing process, the prototype proved to be reliable and easy to use. Each component of the 7-segment display, buzzer, and microcontroller worked as intended, and the system correctly synchronized the visual and audio signals. Testing confirmed that the random number of generations remained within the expected range and that the number of buzzer beeps always matched the number displayed. This demonstrates that even a simple system can provide consistent and meaningful output for users.

The project also provided valuable learning opportunities in electronics, programming, and system integration. It helped reinforce key concepts such as digital output control, timing, and component interfacing. Moreover, it showed that functional prototypes do not always require complex or expensive components; with proper planning and coding, simple modules can achieve practical results.

Finally, while the system successfully met its objectives, there is room for future improvements. For instance, more advanced audio patterns could make the system even more intuitive, real input data could replace random number generation, and the hardware could be made more durable for different environments. Overall, this project demonstrates that combining a 7-segment display with a buzzer provides an effective, low-cost solution for presenting numerical information in an accessible way, forming a strong foundation for more complex assistive devices in the future.

References

1. Arduino official documentation. Arduino platform reference and tutorials. Available at: https://www.arduino.cc

2. Electronics Tutorials. 7-segment display working principle and wiring guide. Electronics-Tutorials.ws.

3. Arduino Project Hub. Using passive and active buzzers with Arduino. Project tutorials and practical examples.

4. Arduino IDE language reference. Random() and randomSeed() functions for microcontroller applications.

5. 7-segment LED display datasheet. Manufacturer documentation describing pin configuration, operating voltage, and current limitations.

6. Piezo buzzer module datasheet. Technical specifications including operating frequency, sound pressure level, and input requirements.

7. Arduino Uno R3 datasheet. Detailed technical specifications covering microcontroller performance, input/output pins, and electrical characteristics.

8. Embedded systems textbook. Fundamental concepts of digital output control and timing in simple hardware prototypes.

9. Maker community forums. Online discussions and shared best practices related to wiring, debugging, and optimization of Arduino-based projects.

Appendices

YouTube Video Link: https://youtu.be/s_JecqIpyVA?si=Tnyz_3JZZpSYrXz8

YouTube Video QR code:

GitHub Repository Link: https://github.com/BeknazarBolotov/7SegmentProject

GitHub Repository QR code: