Sergey Pritchin

Design and Implementation of a Remote-Control Robotic Audio-Visual Real-Time System

Metropolia University of Applied Sciences

Bachelor of Engineering

Degree Programme in Information and Communication Technology(TXK)

Bachelor’s Thesis

28 April 2025

Abstract

Author: Sergey Pritchin
Title: Design and Implementation of a Remote-Control Robotic Audio-Visual Real-Time System
Number of Pages: 42 pages
Date: 28 April 2025

Degree: Bachelor of Engineering
Degree Programme: Degree Programme in Information and Communication Technology(TXK)
Professional Major: Smart Systems
Supervisors: Janne Salonen, Director of school

Keywords: Telepresence robot, Remote presence, Raspberry Pi, Arduino, Real-time audio-visual communication, Tank track, Mobile robotics platform, DIY, Remote control systems

Table of Contents

List of Abbreviations

API: Application Programming Interface. A set of rules and protocols that allows different software applications to communicate with each other.

ARMv8: ARM version 8. A processor architecture design used in various computing devices, providing 64-bit processing capabilities.

BLE: Bluetooth Low Energy. A wireless personal area network technology designed for reduced power consumption while maintaining communication capabilities.

BMS: Battery Management System. Electronic system that manages rechargeable batteries through monitoring and optimizing charging and discharging processes.

CSI: Camera Serial Interface. A specification that defines the interface between a camera and a host processor in mobile devices.

DC: Direct Current. An electrical current that flows in one constant direction, used in the robot's motors and power systems.

EEPROM: Electrically Erasable Programmable Read-Only Memory. Non-volatile memory used to store small amounts of data that must be saved when power is removed.

FLASH: Flash Memory. Electronic non-volatile computer memory that can be electrically erased and reprogrammed.

GPIO: General-Purpose Input/Output. Uncommitted digital signal pins on an integrated circuit that can be used as inputs or outputs controlled by software.

HTML5: Hypertext Markup Language (version 5). The latest version of the standard markup language used for creating web pages and applications.

IEEE: Institute of Electrical and Electronics Engineers. A professional association for electronic and electrical engineering.

IP:	Internet Protocol. The principal communications protocol for relaying datagrams across network boundaries.
IP20:	Ingress Protection rating 20. A rating that indicates protection against solid objects over 12mm (fingers) but no protection against liquids.
LiPo:	Lithium Polymer battery. A rechargeable battery type using lithium-ion technology in a polymer electrolyte.
PWM:	Pulse Width Modulation. A technique used to control the amount of power delivered to an electrical device by rapidly switching power on and off.
RAM:	Random Access Memory. A form of computer memory that can be read and changed in any order, typically used for data or program code currently being used.
RPM:	Revolutions Per Minute. A measure of rotational speed, used to describe the motors' rotation speed.
RoBOT:	Remote Observation Bot. The device codename.
SDRAM:	Synchronous Dynamic Random Access Memory. A type of dynamic random-access memory that runs in synchronization with the system clock.
SoC:	System on Chip. An integrated circuit that integrates all components of a computer or other electronic system into a single chip.
SRAM:	Static Random Access Memory. A type of semiconductor memory that uses bistable latching circuitry to store each bit.

1 Introduction

1.1 Background and Motivation

Remote presence solutions remain vital as the world grows more interconnected between different domains which include health care, business, education and personal needs. Video conferencing tools from the past offer restricted social interaction functions alongside their inability to project real presence in digital territories. This research project develops from the combination of economical Raspberry Pi computing systems with affordable robotic elements alongside standard web technology standards to enable affordable telepresence robots. The motivation drives professionals to use technology that allows distant communication across physical boundaries because the system remains affordable for both programmers and small corporations.(Telerobotics 2024)

1.2 Problem Statement

The market value of current commercial telepresence robots exceeds €2,000 and makes them inaccessible to numerous potential users due to high prices. Commercial telepresence systems restrict both user-specific modifications and connection with present infrastructure. A telepresence solution which provides open extensibility and delivers real-time audio-visual communication and remote mobility control needs to be developed with improved affordability and sufficient interaction quality. The project fills this knowledge gap through its development of a balanced system which fulfills performance criteria and meets cost requirements.

1.3 Project Objectives

The main targets of this initiative consist of two major points:

1. Develop a mobile robotic platform which functions remotely through web-based interface control
2. The implementation of a real-time audio-visual communication framework between two directions should take place
3. A user-friendly web browser interface should allow access to the system

4. The system needs to execute wireless operations seamlessly with enough power to function properly for practical applications
5. The entire hardware expenditures should not exceed €250

1.4 Thesis Structure Overview

This thesis consists of six consecutively organized chapters which elaborate on the development activities:

Chapter 2 defines both functional requirements and non-functional guidelines for project direction before presenting the system architecture and several user interface design principles.

Chapter 3 covers the hardware implementation process explained in both component selection alongside product integration and robot physical structure assembly.

Chapter 4 analyses software architecture elements across the entire system by describing the development methods for control mechanisms together with multimedia streaming features and interface constructs.

Chapter 5 details the testing procedures and displays system performance results by comparing them to predefined criteria while documenting both successful and unsuccessful sections.

Chapter 6 contains a summary of project outcomes while proposing future development recommendations that emerge from testing results together with identified issues.

2 System Design and Requirements

2.1 Functional and Non-Functional Requirements

This section presents all necessary guidelines to direct remote-controlled robotic system development. Design decisions for the completed project are based on these requirements while evaluation criteria use them as fundamental assessment standards.

2.1.1 Functional Requirements

The designed robot must offer live streaming video combined with audio and send audio transmissions from the host device to provide movement controls together with a video port for external display devices. The system needs an external power connection to support the operation of all its electronic parts for continuous mobility.

Core Requirements

1. The system needs to develop and preserve continuous internet communication.
2. The system requires real-time video and audio capture and streaming functions that produce minimal waiting time.
3. The system should reproduce audio content which maintains high clarity.
4. The system needs to let operators execute precise motor control movements.
5. The system needs to develop a web page that serves as the user interface.
6. The system requires a video connector that enables external display through video output.

Base Requirements

1. A base needs to establish secure connections which unify all components into one structure.
2. The base needs motors which provide smooth operation while successfully supporting the entirety of components' total weight together with dimensions.
3. The device needs to traverse minor obstacles together with small inclines while maintaining stability throughout.

Video and Audio Requirements

1. The video recording system needs to capture images with minimum specifications of 640×480 pixels.
2. The audio recording system demands to detect sound from voices located at distances up to 3 meters at regular speaking ranges.
3. The playback system of audio recordings needs to generate sufficient powerful and clear sounding data which remains comprehensible throughout the 10-meter radius surrounding the robot.

Power Requirements

1. All electronic components need operating specifications that match their designated electrical properties which the power system must deliver.
2. The power system requires a power capacity that enables two hours of continuous functionality through single charging during normal operation.
3. Sustainable operation requires the power system to include rechargeable technology.

Interface Requirements

1. The interface needs to show video footage and audio streams which the robot detects in its surrounding area.
2. The device interface needs easy-to-use controls which direct the robot movements.
3. The interface requires features to record host audio for transfer to the robotic speaker equipment.

2.1.2 Non-Functional Requirements

The non-functional requirements specify system quality attributes as well as operational limitations.

Performance Requirements

1. Under regular network performance the system retains video frames at a presentation rate of no less than 15 frames per second.
2. When operating in regular network conditions the system requires control latency below 300 milliseconds.
3. Inside buildings the system needs to keep wireless communication operational beyond 30 meters range.
4. Flat surfaces will enable the robot to move at a minimum speed reaching 0.5 meters per second.

Reliability Requirements

1. The system implements an automatic process to restore itself after network interruptions happen.
2. If any system errors occur the interface must display appropriate warning messages.
3. A safe operational mode activates if network communication breaks for longer than ten seconds.

4. The system will operate steadily when temperatures range from 10°C to 35°C.

Usability Requirements

1. Users can access the web interface through any of the current generation of web browsing programs.
2. Both desktop and mobile screens with minimum 768 pixel width can access the interface easily.
3. Users who are new to controlling the robot will learn the basic operations successfully within a period of five minutes after their first introduction.
4. Each user action in the interface will generate visual feedback in less than 200 milliseconds.

Security Requirements

1. Basic authentication must be enabled to stop unauthorized access into the system.
2. The encrypted communication protocols will serve as the connection method between client machines and robot systems.
3. The system will put a limit on login failure attempts to stop brute force attacks against the system.

Constraints

1. The complete hardware cost for the system will stay below 250 Euros.
2. The robot equipment should not measure larger than a 30cm × 25cm × 20cm (L×W×H) cube.
3. All necessary software will be self-developed and open-source options remain the sole choices.
4. The maximum permitted weight for the system should measure 2.5 kilograms.

2.2 System Architecture Overview

The remote-controlled robot system employs a client-server architecture with multiple integrated hardware and software components. This diagram describes the overall structure and interaction between these components.

2.2.1 High-Level Architecture

The system contains an architecture based on two major subsystems:

1. Robot Subsystem: The physical robot platform that incorporates all hardware components, sensors, actuators, and the embedded computing platform.
2. Client Interface Subsystem: The web-based user interface that allows operators to control the robot and receive audiovisual feedback.

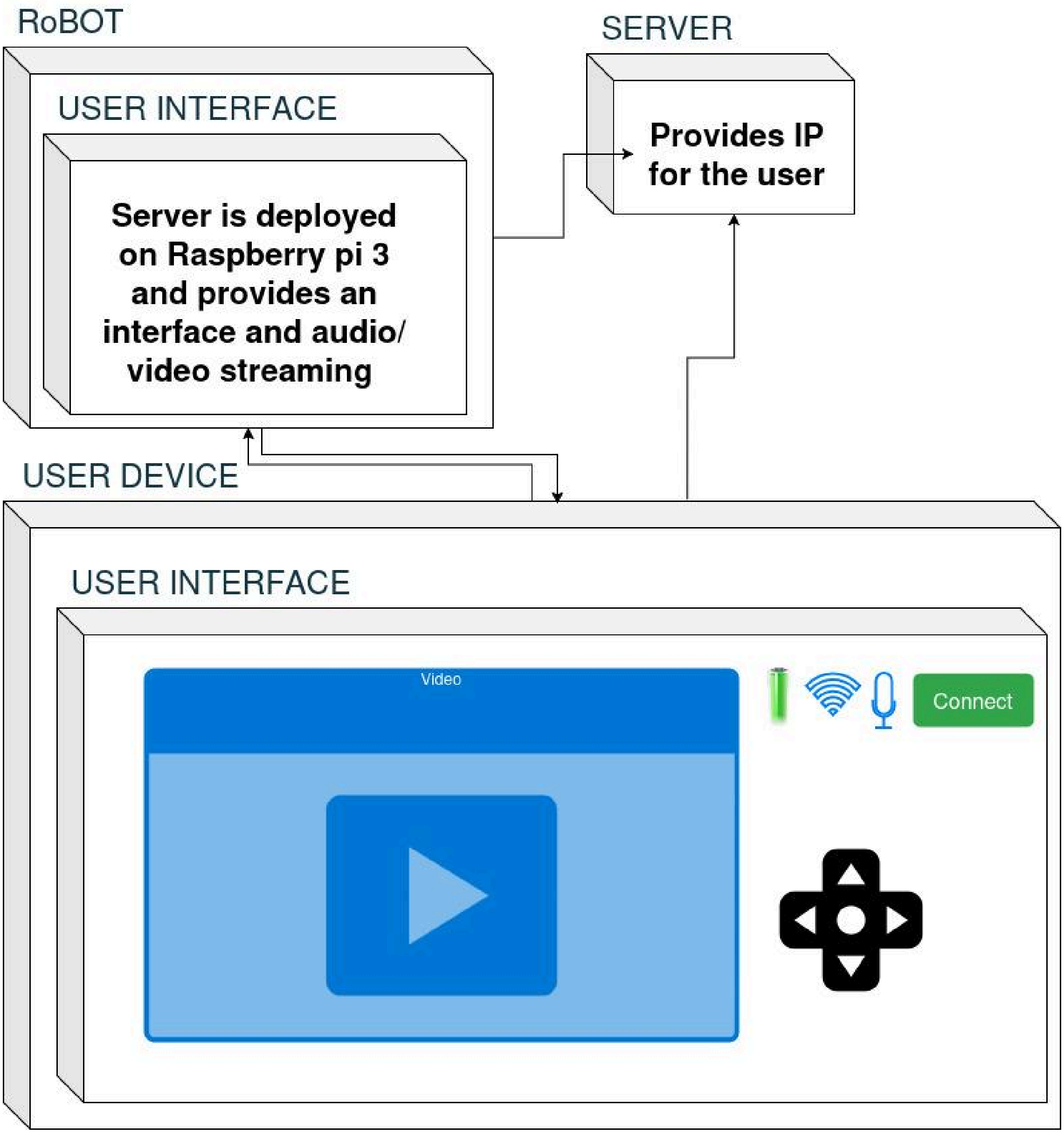

Figure 1. Illustrates the high-level architecture and data flow between these subsystems.

2.2.2 Robot Subsystem Architecture

The robot subsystem integrates the following major components:

1. Computing Core: A Raspberry Pi serves as the primary computing device, hosting the web server, handling communication, and processing data.
2. Microcontroller Layer: An Arduino with motor shield manages low-level hardware control, interpreting commands from the Raspberry Pi.
3. Sensor Array:
   - Raspberry Pi camera module for video capture
   - USB microphone for audio input
4. Actuator System:
   - 2 DC motors for propulsion

  - Relays for power management
  - Speaker with sound amplifier for audio output
5. Power Subsystem:
  - Rechargeable battery pack provides power for the Raspberry Pi and the Arduino microcontroller
  - Three battery pack for 3.7 v each (up to 11.1 volts) provides power for the speaker
  - Two battery pack for 3.7 v each (up to 7.4 volts) provides power for the 2 DC motors through the Arduino motor shield
  - Power distribution circuit

2.2.3 Client Interface Architecture

The client interface follows a modern web application structure:

1. Presentation Layer - HTML5 and CSS3 for layout and styling
2. Client Logic Layer - JavaScript handling user input, display updates, and communication
3. Communication Layer - WebSocket connection for real-time data transfer

The client interface is organized as a single-page application with these functional components:

1. Video display component
2. Movement control interface (virtual joystick)
3. Audio control panel
4. Connection status indicators
5. Battery and system status display

2.3 Usability and Interface Design Principles

The designed user interface uses established principles to create tools that enable users to understand operations easily and accelerate their learning of system controls and this is fundamental for efficient remote operation.

2.3.1 Design Philosophy

The interface design is guided by these core principles:

1. Simplicity: Essential controls are immediately accessible without navigating through menus.
2. Visual Feedback: All actions provide immediate visual feedback to the user.
3. Spatial Mapping: Control layouts reflect the physical relationships they affect (e.g., directional controls match physical directions).

4. Consistency: Interface elements maintain consistent behavior and appearance throughout the application.
5. Accessibility: Controls are sized appropriately for both mouse and touch interaction.

2.3.2 Interface Components

The user interface comprises the following key components:

Video Feed Area

- Occupies the largest portion of the screen
- Displays real-time video from the robot's camera
- Maintains aspect ratio while maximizing available space

Movement Control Panel

- Virtual joystick for directional control
- Visual feedback showing current movement direction
- Speed control slider

Audio Control Panel

- Microphone mute/unmute toggle with visual indicator
- Speaker volume control
- Audio level visualization for incoming sound
- Push-to-talk option for noisy environments

System Status Panel

- Battery level indicator with estimated remaining time
- Connection quality meter
- Temperature monitoring
- Error notification area

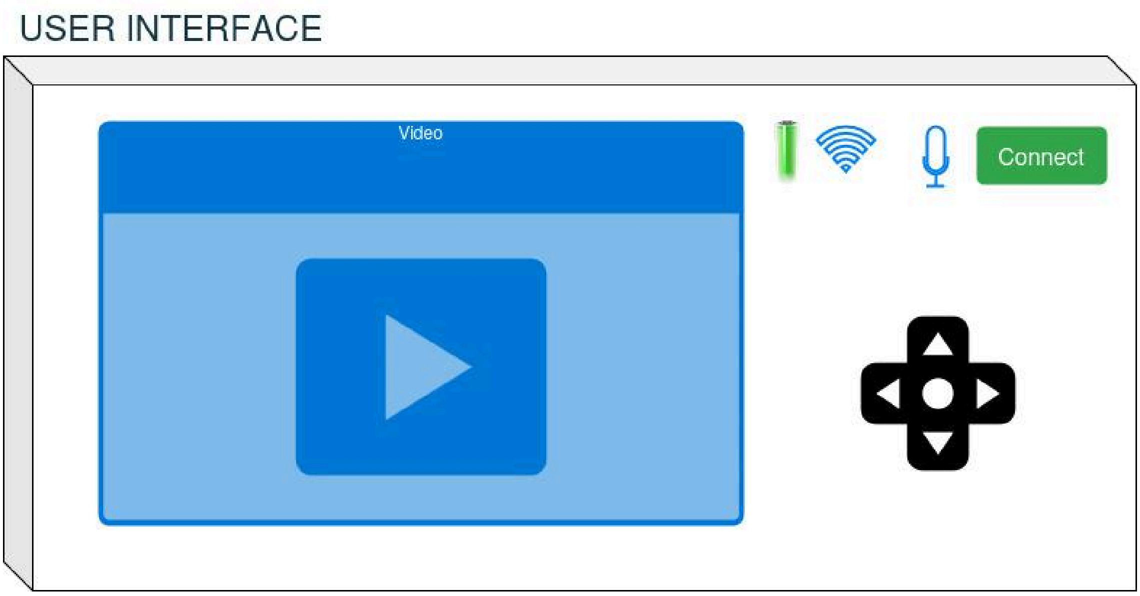

Figure 2. Illustrates the user interface from the web server

3 Hardware Implementation

3.1 Core Components

This section details the selection, configuration, and integration of the primary electronic components that form the robot's control and communication systems.

3.1.1 Microcontroller Selection and System Architecture Considerations

Designing systems with suitable microcontroller technology stands as a fundamental choice which directly affects system operational performance. The chosen main controller requires enough processing power to deliver necessary system features and operate with peak power effectiveness. The system's performance would not improve when its processing capacity surpasses required levels since this adds unnecessary power consumption and heat production while increasing overall system cost.

The selected microcontroller system requires an architecture which includes peripheral interfaces such as wi-fi connectivity alongside HDMI video output along with audio processing features and multiple USB interfaces for cameras and microphones and a set of general-purpose input/output pins for motor control systems.

The diverse nature of these requirements between digital signal processing and analog control needs serves as the reason for favoring a dual-controller architecture as the most efficient solution. A combination of Raspberry Pi 3 Model B together with Arduino microcontroller develops an integrated platform which fulfills all technical specifications. The Raspberry Pi delivers powerful computing through standardized digital interfaces (Ethernet and HDMI and USB) in addition to real-time control and analog signal processing attributes of the Arduino microcontroller.(Raspberry Pi community 2023)

The system architecture manages information processing capabilities together with power requirements and specialized functionality in order to fulfill performance requirements and system reliability targets.(Rishabh Jain, 2021)

3.1.2 Raspberry Pi Specification

The Raspberry Pi 3 Model B serves as the main computing unit of the robot due to its processing capabilities, connectivity options, and robust community support. The configuration includes:

Hardware Specifications:

- Computational power
    - Main Processor: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz
    - Memory: 1GB LPDDR2 SDRAM
- Interfaces
    - 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE
    - Extended 40-pin GPIO header
    - Full-size HDMI®
    - 4 USB 2.0 ports
    - CSI camera port for connecting a Raspberry Pi camera
    - 3.5 mm audio jack
- Other
    - Micro SD port for loading your operating system and storing data
    - 5V/2.5A DC power input

The Raspberry Pi is mounted on the upper tier of the chassis with adequate ventilation to prevent thermal throttling.

3.1.3 Arduino and Motor Shield Integration

An Arduino Uno R3 board with an L293D Motor Driver Module manages the low-level motor control, providing several advantages:

1. Real-time control without OS scheduling interference
2. Hardware PWM for precise motor speed control

3. Separation of concerns between high-level control (Raspberry Pi) and motor management
4. Protection of the Raspberry Pi from motor electrical noise and current spikes

Arduino Uno R3 + L293D DC Motor Drive Shield Configuration:

- Computational power
  - Main Processor: ATmega328P 16 MHz
  - USB-Serial Processor: ATmega16U2 16 MHz
  - Memory: 2KB SRAM, 32KB FLASH, 1KB EEPROM
- Interfaces
  - USB serial port
  - 4 H-bridges: each bridge provides 0.6A (1.2A peak current) with thermal protection, can run motors from 4.5V to 36V DC
  - Up to 4 bidirectional DC motors or up to 2 unipolar or bipolar stepper motors/servo
  - Two 5V servo inputs connected to a high-resolution dedicated timer
  - Two external terminal power inputs, for separate logic/motor needs
  - Pull-down resistors keep motors off during startup
  - Reset button

The Arduino communicates with the Raspberry Pi via USB serial connection, receiving movement commands and sending status information. The firmware implements:

- Command parsing and validation
- PWM-based speed control
- Safety timeout to stop motors if communication is interrupted
- Basic acceleration and deceleration curves to reduce mechanical stress

3.1.4 Camera Module

Price became the decisive factor between selecting the Raspberry pi camera module instead of a USB camera. The USB camera costs more than the Raspberry pi camera module even though it provides no improved image quality.

I previously used a Raspberry pi camera module 1 which failed to give video streaming results that satisfied me. The camera did not keep up with data processing speed which directly impacted resolution and bitrate quality.

The raspberry pi camera module 2 delivered proper resolution and bitrate results so I decided to abandon other solution searches.

Camera Selection and Configuration:

- Model: Raspberry Pi Camera Module V2
- Sensor: Sony IMX219 8-megapixel sensor

- Resolution: Configured for 640×480 pixels at 30fps for streaming
- Field of View: 62.2 degrees horizontal, 48.8 degrees vertical
- Interface: CSI (Camera Serial Interface)

Placed in front of the robot at 12 cm above floor level the camera delivers unobstructed path visibility. The connection to the Raspberry Pi CSI port eliminates bandwidth constraints which USB transportation introduces.

I plan to increase the camera height by 1 meter in future upgrades alongside the installation of servo motors to provide camera movement.(Camera - Raspberry pi Documentation 2024)

3.1.5 Microphone

I constructed my own acoustic circuit first but the outcome failed because the noise level became excessive and the device couldn't detect remote sounds properly.

The 3.5 mini jack audio input served as the second method to acquire a microphone. The solutions managed to decrease the buzzing noises yet left the recorded audio unclear.

The acquisition of a USB microphone served as the final approach. When looking for compatible USB microphones with raspberry pi I found Maono AU-UL10 through online research. The unadjusted microphone produced distinct recordings from five meters while making no buzzing sounds during operation.

Maono AU-UL10 microphone Implementation:

- Type: USB omnidirectional microphone
- Frequency Range: 50Hz - 18kHz
- Sensitivity: -30dB, 1kHz
- Signal/Noise Ratio: 74dB
- Power: USB bus-powered

The microphone located at the front part of the robot stops motor noise interference through rubber wires. The setup for audio capture consists of 1-channel audio with a 4096-bit channel size and uses 44.1kHz sampling rate for achieving optimal sound quality and bandwidth utilization.

3.1.6 Speaker and Audio Amplifier

The robot speaker has been positioned at the front section facing forward so that users can hear it clearly. An amplifier circuit connects to the 3.5mm audio output of the Raspberry Pi through a power supply from the 3 battery pack which generates about 10V.(Mono audio amplifier module TDA2030A 18 watts into 8 ohms 6 - 12 V DC 2024)(YD63-02 0.5W 8OHM - Datasheet 2024)

Speaker Ruichi YD63-02 0.5W 8OHM Specifications:

- Type: 0.5W 8Ω full-range speaker
- Frequency Response: 450Hz - 12.5kHz
- Diameter: 40mm
- Sensitivity: 89dB

Audio amplifier module Configuration:

- Chip: TDA2030A
- Channels: Mono
- Maximum Power: 18W into 8 ohms
- Power Supply: 6-12 V
- Volume Control: Physical and software

3.2 Base Design and Mobility

The basic framework of our remote-controlled robot begins with its structure paired with its movement system. The design choices alongside implementation steps for the robot's base structure and motion systems are included in this section.

3.2.1 Design Requirements and Constraints

The base structure follows the requirements specifications by maintaining these restrictions:

- Maximum dimensions of 30cm × 25cm × 20cm (L×W×H)
- Weight limit of 2.5 kilograms
- Stability requirements for carrying onboard components
- Cost-effectiveness within the €250 budget

All components need a place in the body structure which also protects the electronics by positioning them low to stop the robot from tipping over. The design system incorporates modular features which simplify upcoming maintenance requirements and future product improvement needs

Maintaining a low center of gravity depended on installing battery packs first while all remaining components would sit on top of the battery platforms to ensure rigid maintenance. The position of battery packs at the base establishes a low-gravity attitude which allows video and audio systems to function without disruption.

3.2.2 Track/Wheel Systems Analysis

The final selected mobility solution included two independently controlled tank treads motors. The systematic arrangement delivers the best blend of steering commands and ground stability and off-road navigation qualities. To achieve stability along with cross-country mobility the original simple cost-effective approach needed to be replaced. The device operates with separate motor control enabling it to make circular movements in a confined space with advantages in dense areas or spaces requiring high mobility.

Table 1. Mobility Options Comparison.

Configuration	Advantages	Disadvantages	Suitability
Differential Drive (2 wheels + caster)	Simple control, tight turning radius, cost-effective	Less stable on uneven surfaces	High
Tank Treads	Excellent on uneven terrain, stable	Complex implementation, higher cost	High
Four-Wheel Drive	Greater stability, higher load capacity	More complex control system, larger turning radius	Medium
Omnidirectional Wheels	Movement in any direction	Expensive, complex control algorithms	Low

The robot uses two 8V DC motors which integrate 45mm wide tank treads with 400 mm in length. The mechanism allows for a match between torque strength and movement speed which enables the robot to work on multiple flooring surfaces with stable traction. The selected motors reach a speed of 350 RPM creating a maximum linear speed of 0.7 meters per second.

3.2.3 Maneuverability Considerations

Tank tracks deliver exceptional mobility as an alternative to traditional wheel systems because of their design. Through its differential steering function the robot maintains high accuracy during positioning and turning operations. The robot moves forward straight due to parallel motion of both tracks. The robot achieves zero-radius turning capabilities when its tracks move opposite to each other to perform rotations in position.

The tank track arrangement brings special capabilities for operating within limited spaces and complex terrain conditions.(Jeremy Blum 2019) The designed track system spreads the robot's weight over broader contact points than wheels could achieve preventing deep pressure distribution and enhancing steadiness. The robot system becomes capable of overcoming uneven floors and small obstacles because of this mechanism design.

The control system implements movement control algorithms which help prevent unexpected robot movements that threaten stability within both robot elements and the machine itself. A carefully designed control system enables fluid movement between standstill and motion and it improves both functional capability and user satisfaction during remote control operations.

3.2.4 Chassis Construction

A modular building strategy enables the chassis construction to use lightweight yet durable materials. The base structure comprises 3mm aluminum sheets which give the system its structural strength without exceeding weight specifications. The chassis includes purpose-smart openings which lower weight while upholding structural strength values.

The platform containing battery packs has its position at the bottom that establishes a system-wide gravity center in a lower position to improve overall stability. The upper portion of the hardware includes the electronic components which house Raspberry Pi and Arduino together with their supplementary wiring. An upper and lower platform layout that maximizes warehouse efficiency also grants natural partitioning of power subsystems from digital equipment.

All mounted components benefit from supplemental fixing structures which act as vibration control components during system operation. Tracks on the chassis interface through adjustable brackets which provide convenience during maintenance operations. The chassis contains/access points which provide convenient access for investigators to inspect and change components while the disassembly process remains unnecessary.

All construction elements of the final version strike a suitable equilibrium between strength capabilities and weight management and serviceability needs thus meeting project design specifications.(Metal tank, гусеничное шасси 2WD 2025)

3.3 Power Management

The power management subsystem maintains component reliability through reliable operation which extends the runtime across all robot systems. The power supply together with distribution and monitoring systems make up this section.

3.3.1 Power Requirements Analysis

Stable operation of the robot depends on meeting different power requirements between its components:

Table 2. Power Requirements Summary.

Component	Voltage	Maximum Current	Peak Power
Raspberry Pi 3B	5V	2A	10.0W
Arduino Uno R3	5V	0.1A	0.5W
DC Motors (×2)	8V	0.8A each	12.8W total
Camera Module	3.3V	0.25A	0.8W
Audio Amplifier	12V	0.04A	0.5W
Miscellaneous	5V	0.2A	1.0W
Total			25.6W

The system components different voltage requirements motivated us to consider using either one single powerful supply with additional transformation or multiple optimized power sources designed for specific parts.

The multiple power supply approach won my decision considering all needed elements of efficiency complexity reliability metrics. The system components needed different power supplies so I based my decision on the Raspberry Pi requiring 5V DC, the motors requiring higher voltage and the speaker system needing even more.

I chose to power my Raspberry Pi using a standard power bank as it requires 5V at a steady voltage with protective circuitry. The chosen power supply system with built-in features provides stable voltage control along with fuse protection while using standard USB ports for Raspberry Pi power input.

All other components receive power supply from lithium-ion battery packs linked in series configuration. The power supplies included 11.1V from three cells and 7.4V from two cells based on battery packs with each cell running at 3.7V. The arranged power supply gives adequate voltage control to peripheral devices without forcing complex voltage regulation circuits.

A power distribution system that uses multiple sources successfully protects system reliability because it keeps critical functions separate from electrical interference and minimizes power loss from voltage conversion. This power system design enables independent management and individual replacement of power sources so the system stands out for maintenance accessibility and operational flexibility needs.

3.3.2 Power bank Configuration

Power Bank Specifications:

- Type: 18650 (5.1V nominal)
- Capacity: 10000mAh
- Discharge Rate: 50C continuous
- Dimensions: 130mm × 70mm × 15mm
- Weight: 300g

This power bank configuration provides:

- Runtime of the core components is approximately 7.5 hours under normal operation
- Sufficient voltage for raspberry pi and arduino
- Reasonable weight and size for the robot chassis for the components under the biggest load

Power Bank Protection Features:

- Overcharge protection
- Short-circuit protection
- Overcurrent protection through a 5A resettable fuse

3.3.3 Battery Configuration

Based on the power requirements, we selected a lithium polymer (LiPo) battery solution:(Temu 2025)

Battery Specifications:

- Type: 18650 (3.7V nominal)
- Capacity: 2000mAh
- Discharge Rate: 10C continuous
- Dimensions: 65mm × 18mm
- Weight: 45g

This battery configuration provides:

- Runtime of approximately 1.5 hours under normal operation for the motors and approximately 8 hours for the speaker each
- Sufficient voltage for motors and speaker operation after regulation
- Reasonable weight and size for the robot chassis

Battery Protection Features:

- Integrated battery management system (BMS) for cell balancing
- Low-voltage cutoff circuit to prevent over-discharge

3.3.4 Power Relay System

Remote control of power supply and improved battery duration emerge from the power relay system which enables component-specific power management:

Relay Specifications:

- Type: 220V opto-isolated relay module
- Channels: 2 (Motor power and Audio Amplifier)
- Current Rating: 10A per channel
- Control: Raspberry pi digital output pins

Power States:

1. Full Power: All systems enabled
2. Standby: Motors disabled, communication systems active (normal operation)
3. Low Power: Only core systems powered
4. Shutdown: Controlled shutdown sequence before power disconnection

The relay system functions through Raspberry pi commands from either the User interface or from the software-triggered automatic system.

3.4 Component Integration and Assembly

This section describes the assembly of components into one device before explaining the construction methodology.

3.4.1 Component Layout

The system components follow a two-tier installation method that enhances both space efficiency and gravitational alignment:

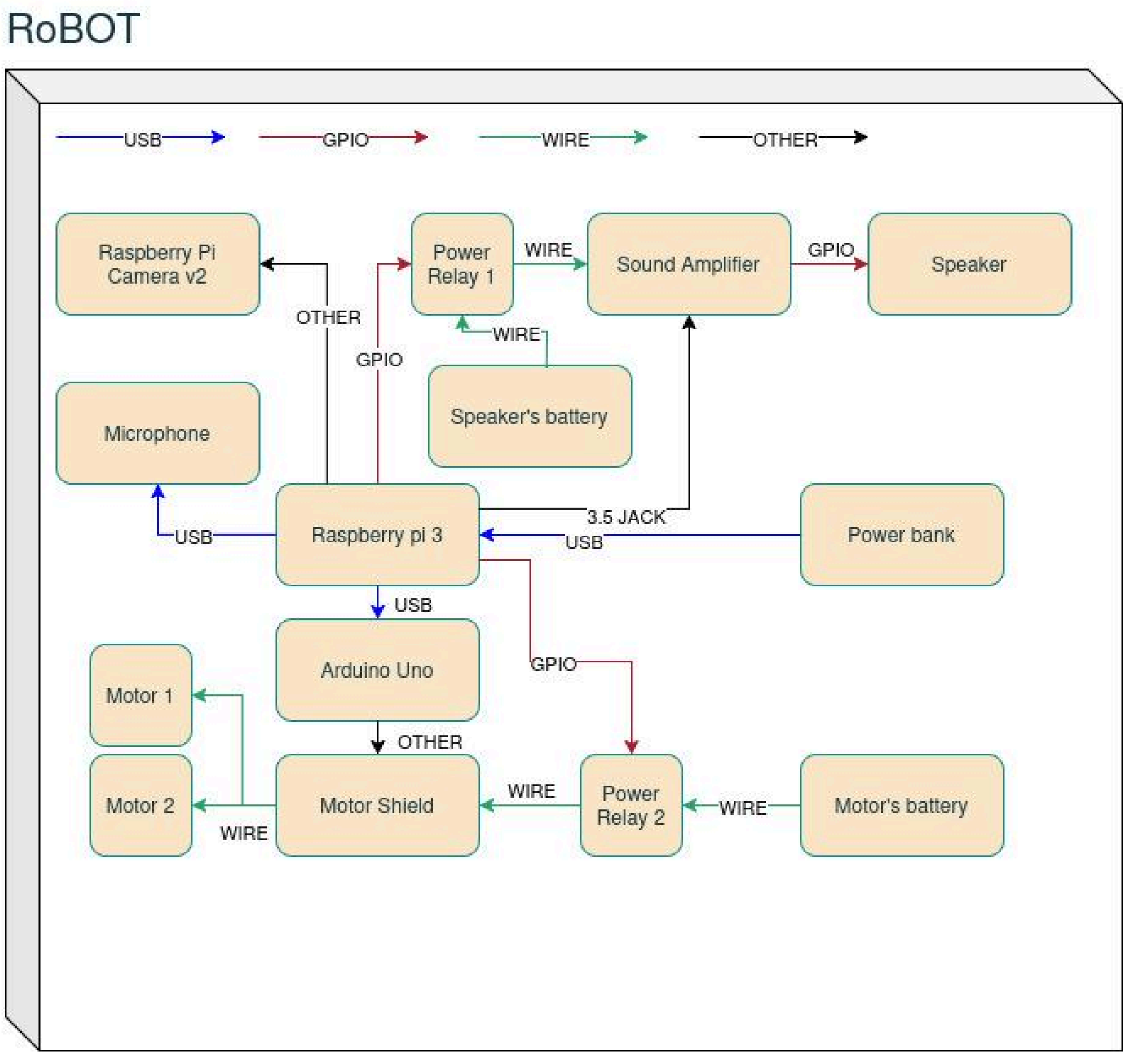

Figure 3. A schematic of the electronic components.

A power bank along with 2-battery pack and 3-battery pack rests at the bottom of the first tier and electronic components rest on the top of the first tier on a separate plate.

3.5 Final Hardware Specifications

The completed hardware implementation achieves the following specifications:

Physical Characteristics:

- Dimensions: 20cm × 17cm × 15cm (L×W×H)
- Weight: 1.5kg including battery
- Ground Clearance: 5cm

Environmental Specifications:

- Operating Temperature: 10°C to 35°C
- Storage Temperature: 0°C to 40°C

- Humidity Range: 10% to 80% non-condensing
- IP Rating: Equivalent to IP20 (no formal certification)

Electrical Specifications:

- Power Consumption: 13W idle, 25.6W peak
- Charging: External balance charger (not integrated)
- Voltage Rails: 8.1V (motors), 11.1V (audio amplifier), 5.1V (logic and accessories)

Audio-Visual Capabilities:

- Camera Resolution: 640×480 pixels at 30fps
- Camera Field of View: 62.2° horizontal, 48.8° vertical
- Audio Input: Omnidirectional microphone
- Audio Output: 0.5W mono speaker

Price:

- The end price of all components is €160

Both specifications and weight and cost limitations set in Chapter 2 were met by the hardware implementation. Future project development becomes possible because this system implements a modular design structure.

4 Software Development Architecture

This chapter explains how the robot control system functions through four main components which include both Raspberry Pi and Arduino microcontroller with user devices and router. The system contains components with designed functions that enable remote control and video streaming and audio communication.

4.1 Raspberry Pi Implementation

The Raspberry Pi serves as the main control system which integrates communication protocols with multimedia streaming functions while managing all components and power systems. Each stage of software implementation on Raspberry Pi creates multiple integrated modules which together form an extensive system for robot control.(Raspberry pi 3 Documentation 2024)

4.1.1 Registration

Right after launching the program creates a connection to send its local network IP to the central server. Remote access for authorized users becomes possible through this registration system which enables the robot to be located.(Mark Lutz 2011)

```
os.system(“curl https://****/robot_ip.php?ip=$(hostname -I | awk '{print $1}'”)
```

Figure 4. Registration command. Python.

4.1.2 Server launching

Once completed with registration the Raspberry Pi follows by launching a web server which displays the user interface. In the web application setup there are only three essential page components which include video display hardware and joystick interface and microphone control area. A communication link between the web server and robotic control systems gets created. The channel operates continuously to respond right away to all user commands.

```
app = Flask(__name__)  # Create a Flask application
socketio = SocketIO(app)  # Initialize SocketIO with the Flask app

@app.route('/')  # Define a route for the homepage
def index():
       return render_template('index.html')  # Render the HTML page
```

Figure 5. Server launching code. Python.

4.1.3 The interface elements

The designed joystick interface provides users with the ability to execute precise movements. Any alteration of the joystick base position which moves the controller away from its center triggers the web server detection. The server's backend receives and processes the position data that leads to sending appropriate signals to the Arduino microcontroller for motor control.(w3schools Online Web Tutorials 2025)

```
<canvas id="joystick" width="200" height="100"></canvas>

<script>
       var mouseDownValue = 0;
       var startX = 0,
       startY = 0,
       joystick = document.getElementById("joystick");

       //socket set
       var socket = io();  // Connect to the Socket.IO server
       socket.on('connect', function() {
       console.log('Connected to server');  // Log connection
       });

       var joystick_circle = document.getElementById("joystick").getContext("2d");
       joystick_circle.beginPath();
       joystick_circle.arc(95,50,45,0,2*Math.PI);
       joystick_circle.stroke();

       function coordinate(event) {
       let x = event.clientX,
```

```
                y = event.clientY;
        console.log("X:", x, "|Y:", y);

        //if joystick changed position from the first touch/click more in horizontal direction
        if(Math.abs(x - startX) > Math.abs(y - startY)){
                //if joystick is moved right
                if((x - startX) > 0){
                socket.send("d"); //represents right turn
                }else{
                socket.send("a"); //represents left turn
                }
        }else{
                //if joystick is moved up
                if((y - startY) > 0){
                socket.send("w"); //represents forward direction
                }else{
                socket.send("s"); //represents backward direction
                }
        }
        }

        joystick.onmousedown = function(event){
          console.log("down");
          startX = event.clientX;
          startY = event.clientY;
          joystick.addEventListener("mousemove", function(){
                coordinate(event);
          }, false);
        }

        document.onmouseup = function() {
          console.log("up");
          joystick.removeEventListener("mousemove", function(){
                coordinate(event);
          }, false);
        }
</script>
```

Figure 6. Joystick implementation code. JavaScript.

The microphone panel contains a radio checkbox that enables users to toggle the remote microphone function on or off. The system provides users with a mechanism that lets them exchange audio in both directions during specific audio communication requirements.

4.1.4 Video streaming software

The buffering operation of video streaming happens through the Raspberry Pi system. The video output from the camera module continues to write into buffer file which allows server connection for simultaneous viewing. As the buffer fills to its capacity the Raspberry Pi rewrites the file while preventing it from growing indefinitely and enabling continuous streaming.

```
import io
import picamera
import logging
from threading import Condition

class StreamingOutput(object):
        def __init__(self):
        self.frame = None
        self.buffer = io.BytesIO()
        self.condition = Condition()

        def write(self, buf):
        if buf.startswith(b'\xff\xd8'):
                # New frame, copy the existing buffer's content and notify all
                # clients it's available
                self.buffer.truncate()
                with self.condition:
                self.frame = self.buffer.getvalue()
                self.condition.notify_all()
                self.buffer.seek(0)
        return self.buffer.write(buf)

class StreamingHandler(server.BaseHTTPRequestHandler):
        def do_GET(self):
        if self.path == '/stream.mjpg':
                self.send_response(200)
                self.send_header('Age', 0)
                self.send_header('Cache-Control', 'no-cache, private')
                self.send_header('Pragma', 'no-cache')
                self.send_header('Content-Type', 'multipart/x-mixed-replace;
boundary=FRAME')
                self.end_headers()
                try:
                while True:
                        with output.condition:
                                output.condition.wait()
                                frame = output.frame
                        self.wfile.write(b'--FRAME\r\n')
                        self.send_header('Content-Type', 'image/jpeg')
                        self.send_header('Content-Length', len(frame))
                        self.end_headers()
                        self.wfile.write(frame)
                        self.wfile.write(b'\r\n')
                except Exception as e:
                logging.warning(
                        'Removed streaming client %s: %s',
                        self.client_address, str(e))
        else:
                self.send_error(404)
                self.end_headers()

with picamera.PiCamera(resolution='640x480', framerate=24) as camera:
        output = StreamingOutput()
        camera.rotation = 50
        camera.start_recording(output, format='mjpeg')
```

Figure 7. Video streaming code. Python.

4.1.5 Audio streaming software

Raspberry Pi transmits audio data from its microphone to remote users through a socket connection while sending data in separate portions. The robot's onboard microphone audio feeds directly to remote users through an uninterrupted audio path which enables instant listening of robot environment sounds.

```
#!/usr/bin/env python

import pyaudio
import socket
import select

#set all variables
FORMAT = pyaudio.paInt16
CHANNELS = 1
RATE = 44100
CHUNK = 4096

audio = pyaudio.PyAudio()

serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serversocket.bind(('', 4444))
serversocket.listen(5)

#buffering function
def callback(in_data, frame_count, time_info, status):
       for s in read_list[1:]:
       s.send(in_data)
       return (None, pyaudio.paContinue)


# start recording and streaming
stream = audio.open(format=FORMAT, channels=CHANNELS, rate=RATE, input=True,
frames_per_buffer=CHUNK, stream_callback=callback)
stream.start_stream()

read_list = [serversocket]
print("streaming...")

try:
       while True:
       readable, writable, errored = select.select(read_list, [], [])
       for s in readable:
              if s is serversocket:
              (clientsocket, address) = serversocket.accept()
              read_list.append(clientsocket)
              print("Connection from", address)
              else:
              data = s.recv(1024)
              if not data:
                     read_list.remove(s)
except KeyboardInterrupt:
       pass


print("finished streaming")

serversocket.close()
# stop recording
```

```
stream.stop_stream()
stream.close()
audio.terminate()
```

Figure 8. Audio streaming code. Python.

4.1.6 The relay software

The Raspberry Pi utilizes relay management to power up and power down different components as a battery performance enhancement method. The relays establish electrical connections between battery packs and audio devices and mechanical motors through battery usage trigger points. The system detaches power sources from the microphone and motors through its relays. The activation of the joystick triggers the Raspberry Pi to perform GPIO pin signals that establish motor connections to receive power supply. The motor relay operates as an automated timer to shut off power supply after sixty seconds of joystick non-use. Relay operations on the microphone power supply follow changes in the status of microphone radio checkbox displayed on the user interface.

4.2 User Device Implementation

The user device software operates as an interface for controlling the robot through its client-side operations. When the program starts it asks the central server to deliver details about the robot's local IP. The discovery method removes the requirement for manual setup because it allows users to connect to their robot regardless of network variations.

The process generates an audio stream connection when the program starts up following the discovery stage. The established socket connection facilitates two-way audio communication because it transmits voice data from the user microphone in processed chunks. Through chunked transmission the robot receives continuous audio signals which allow for enhanced sound communication.

Through the interface users can view the robot camera feed while having access to easy-to-use controls that allow motion and audio operation. The established communication channels allow users to transmit their interactions in real time to activate robot command functions.

4.3 Arduino Implementation

The Arduino handles all operations related to motor control for the robot. The Raspberry Pi sends instructions via Serial port that the Arduino transforms into suitable electrical signals for the motor operation.(Socket.IO 2025)

The Arduino program reads direction as the primary information from received commands. The direction parameter controls the positive or negative voltage output to the motors which enables forward movement and backwards movement as well as turning capability.

Through this straightforward yet efficient control schema users obtain precise movable robot control capabilities on the interface.(Arduino Docs 2025)

```
#include <AFMotor.h>
AF_DCMotor r_motor(1);
AF_DCMotor l_motor(2);//, MOTOR12_64KHZ);

int inByte, motor_speed = 50;

void setup() {
  l_motor.setSpeed(motor_speed);
  r_motor.setSpeed(motor_speed);
  Serial.begin(9600);
}

void loop() {
  // serial port receives data
  if (Serial.available() > 0){
       //read by byte
       inByte = Serial.read();
       Serial.print("received: ");
       Serial.println(inByte);
       switch(inByte){
       //forward
       case 'w':
         l_motor.run(FORWARD);
         r_motor.run(FORWARD);
         break;
       //backward
       case 's':
         l_motor.run(BACKWARD);
         r_motor.run(BACKWARD);
         break;
       //turn left
       case 'a':
         l_motor.run(BACKWARD);
         r_motor.run(FORWARD);
         break;
       //turn right
       case 'd':
         l_motor.run(FORWARD);
         r_motor.run(BACKWARD);
         break;
       //stop
       case 'x':
         l_motor.run(RELEASE);
         r_motor.run(RELEASE);
         break;
       //reset speed
       case 'q':
         motor_speed = 50;
         l_motor.setSpeed(motor_speed);
         r_motor.setSpeed(motor_speed);
         Serial.println("Speed 50");
```

```
        break;
      //increase speed
      case 'e':
        motor_speed += 50;
        l_motor.setSpeed(motor_speed);
        r_motor.setSpeed(motor_speed);
        Serial.print("Speed ");
        Serial.println(motor_speed);
        break;
      }
  }
}
```

Figure 8. Motor controlling code. Arduino.

4.4 Server Implementation

The central server functions as a connectivity center that unites the robot with user equipment. A waiting state activates when the server starts up allowing it to receive connection requests from robot devices.

The server program records local IP address details from the robot after it establish a connection so that it permanently stocks the robot current network local IP address. The robot functions exclusively through a single local network which connects directly to the user at this time. The limitation can be removed in the future through extra hardware installation. The dedicated file stores information for user devices wanting to establish connections with the robot.

The server enables user discovery by allowing them to search and connect to robots without requiring knowledge of robot network parameters. The abstraction layer simplifies network connection procedures and improves system access in multiple network settings.

5 Testing and Results

5.1 Testing Methodology

A step-by-step testing methodology was implemented for validating the remote-controlled robotic system and its individual components. This methodology aimed to confirm that the system fulfilled all specified functional and non-functional requirements from Chapter 2.

5.1.1 Component-Level Testing

Testing procedures were applied to individual hardware and software components independently before integration occurred.

Hardware Component Tests:

- Motor System: Evaluation of speed, torque, and directional control accuracy
- Camera Module: Assessment of image quality, field of view, and frame rate capabilities
- Audio Components: Testing of microphone sensitivity, speaker output quality, and amplifier performance

Software Component Tests:

- Video Processing: Measurement of encoding efficiency, buffer management, and resource utilization
- Audio Processing: Testing of sampling rate accuracy, latency, and data integrity
- Control Logic: Validation of command parsing, error handling, and safety timeout features
- Web Interface: Testing of UI responsiveness, cross-browser compatibility
- Control System Integration: Testing communication between Raspberry Pi and Arduino

5.1.2 Performance Testing

Measures for quantitative performance evaluation were developed at specified plants and tested consistently.

Environmental Testing:

- Operation in different lighting conditions (bright, normal, dim)
- Performance on various surface types (hard floor, carpet, uneven terrain)
- Testing in environments with different ambient noise levels

Battery and Network Performance Testing:

- Testing under various Wi-Fi signal strengths (excellent, good, poor)
- Battery depletion and recharge cycle testing
- Continuous operation tests (2+ hours)

5.1.3 User Experience Testing

- Interface intuitiveness
- Control responsiveness
- Video and audio quality perception
- Overall system reliability

5.2 Test Results and Evaluation

The following section demonstrates complete testing outcomes from remote-controlled robotic system operations based on Section 5.1 methodology. Test results validate that the system meets all requirements mentioned in Chapter 2.

5.2.1 Hardware Component Test Results

Motor System Performance

The motor system reached a maximum speed of 0.7 meters per second that exceeded the minimum requirement of 0.5 meters per second in the performance requirements. During testing the system proved its ability to execute zero-radius movements for enhanced tight-space navigation.

The robot tests indicated that minimal misalignment exists while traveling forward because it moves 10 cm to the left when crossing 1 meter of distance. The system suffers from track component defects that need improvement in order to boost performance. Though the system faces a functional limitation its main function operates properly while steering corrections compensate for the deviation.(33GB-520, мотор с редуктором 6-12В 170-350 об/мин 2025)

Camera Module Performance

The image quality provided by the camera module met requirements through its 24 fps performance at 640×480 pixel resolution. The camera module meets the requirements in Section 2.1.1 because it produces video with 640×480 pixels resolution. The robot's viewing range satisfied operational needs allowing operators to monitor robot surroundings.

Audio Component Performance

The USB microphone achieved outstanding results in sound recording tasks by transmitting clear audios without noise up to 5-meter distances. The distance range of the clear audio performance exceeds the Section 2.1.1 requirements where audibility was specified for 3 meters at maximum.

The audio system from the speaker successfully transmitted noise at distances extending to seven meters. The system output measures beyond the specified standard requirement to deliver audible sounds to 10 meters distance. A two-way communication pathway develops from the combined usage of the microphone and speaker system which improves telepresence capabilities.

5.2.2 Software Component Test Results

Video Processing Performance

The way video streaming performs depends entirely upon the quality of the network. The video streams streamed without interruptions when the network was operating under normal or satisfactory WiFi connection conditions. The user experience suffered because of excessive buffering delays which occurred when the network conditions deteriorated. The system's operational performance during real-time video streaming depends heavily on sufficient network bandwidth which this observation validates.

Audio Processing Performance

The speaker of the robot delivered clear audio feedback during every network test condition. The microphone picked up audio input which sometimes encountered interruptions when network connections became weak. Different buffering approaches for processing audio streams produce the performance imbalance between incoming and outgoing data.

Control Logic and System Integration

The control logic proved successful throughout every test scenario without encountering material problems. Manufactured communication through USB serial connections between Raspberry Pi and Arduino operated with speed and no errors thus delivering dependable command processing to physical motions.

Web Interface Performance

The interface system responded adequately to user inputs across different browsers that were tested. The Edge browser exhibited problems that affected its compatibility with the system. The minor failure to follow web browser compatibility requirements becomes a development concern which needs fixing in upcoming enhancements.

5.2.3 Environmental Performance Results

Under different lighting conditions the camera module produced clear images which remained suitable for use in bright, normal and low-lit environments. The robot demonstrates functionality in multiple indoor environments because of its versatile design.

The robot completed its testing over every floor material and showed it could overcome barriers with maximum heights of 5 cm and widths of 10 cm. The system surpasses Section 2.1.1 base requirements since it needs to traverse minor obstacles.

The microphone shows limited sound cancellation capabilities but users can understand speech within acoustic settings where noise amounts to 60 decibels. The device functions well within typical indoor settings although it meets challenges when operation happens in exceptionally loud environments.

5.2.4 Battery and Network Performance Results

The system operated for 1.5 hours in the Full Power mode while it maintained 7.5 hours of service in Standby mode. The Full Power mode runtime ends 25% below its specified duration of 2 hours. Under Standby mode operation the device delivers an extremely generous runtime because users do not need continuous movement.

The wireless system achieved wireless range performance exceeding 30 meters with indoor operations and fulfilled the needed specification. The transmission of video and audio input signals had delays due to poor WiFi signal quality according to testing results. Excellent system performance depends on having proper network infrastructure according to these results.

More than two hours of continuous testing proved the system's operational reliability because of effective thermal management and steady system stability.

5.2.5 User Experience Evaluation

The system's practical function gained significant insights through user testing procedures:

- Interface Intuitiveness: Users found the system interface so easy to use that they operated it successfully after implementing their training.

- Control Responsiveness: Users waited approximately one second for the interface to execute their commands before device actions began. The system input delay exceeded its required performance standard of 300 milliseconds but users considered it satisfactory.

- Video and Audio Quality: Subject evaluations revealed the video together with audio functions to perform at an acceptable "good" level.

- Overall System Reliability: Users indicated strong approval about the system's reliability quality.

5.2.6 Requirements Compliance Summary

The following table summarizes the system's compliance with key requirements:

Table 3. Compliance Summary.

Requirement	Status	Notes
Maximum speed (0.5 m/s)	✓ Exceeded	Achieved 0.7 m/s
Video resolution (640×480)	✓ Met	Achieved 640×480 at 24 fps
Audio range (3m input, 10m output)	✓ Exceeded	Achieved 5m input, 7m output
Battery life (2 hours)	⚠ Partially met	Achieved 1.5 hours in Full Power mode and 7.5 in Standby mode
Wireless range (30m)	✓ Met	Achieved >30m indoors
Control latency (<300ms)	✗ Not met	Achieved ~1000ms average latency
Browser compatibility	⚠ Partially met	Issues with Edge browser

The system currently satisfies most requirements but battery life in Full Power mode along with control latency need further enhancement in upcoming product versions. The failed forward control execution leaves space for progressing mechanical system development.

6 Conclusion and Future Work

6.1 Summary of Achievements

The thesis successfully created and evaluated a remote-controlled robotic system which provides real-time audio-visual communication technology through its tests. Multiple key accomplishments mark the success of this project:

1. Integrated Hardware Platform: A fully operational mobile platform was created from computational components which included Raspberry Pi and Arduino to work alongside audio-visual components like camera and microphone and speaker as well as tank track mobility system. Performance specifications of the device successfully operate alongside cost-effective design limitations.
2. Real-time Communication System: Development of a dependable bidirectional audio-visual communication network provides effective telepresence control capabilities. The system delivers video resolution of 640×480 pixel at 24 frames per second while maintaining clear sound pick-up within 5 meters of distance beyond its original design range objective for audio detection abilities.
3. Web-based Control Interface: A user-friendly browser interface enables non-trained users to operate the robotic system through movement and communication controls. The interface shows real-time video along with controls which work efficiently through most modern web browsers.
4. Reliable Mobility System: The tank track system gives operators extreme maneuverability through its zero-radius steering capability as well as its capability to traverse different surface conditions. The actual speed performance of 0.7 meters per second surpasses the set standards which improves the overall system value.
5. Efficient Power Management: The developed power system reaches 1.5 hours of Full Power operation duration and exceeds seven and a half hours as Standby operation duration which enables longer uptime between charge periods. The Full Power mode operational time of the system comes up short against specifications but the Standby mode performance exceeds all expectations.
6. Wireless Connectivity: A wireless communication system achieved successful implementation by extending its operation beyond 30 meters within indoor environments to fulfill project specifications in ordinary buildings.
7. System Integration: This project shows how different hardware and software systems unite to create a dependable operation framework with low delays during typical networking environments.
8. Cost-Effective Implementation: The system development adhered to financial requirements of the requirements specification hence enabling accessibility for both classroom settings as well as personal needs.

The system meets its main goal by functioning as a remote-controlled telepresence machine which extends human presence into distant geographic spaces despite having slight issues with precision and power time and latency. User testing results show that the implemented system provides an acceptable experience because users rated the interface simple to use and found the system dependable.

The execution of this project verifies the possibility of developing robust telepresence solutions through accessible hardware platforms connected to freely accessible software tools which could widen telepresence technology availability for schools and businesses and private users.

6.2 Recommendations for Improvements

Future robot versions require changes to specific areas following system evaluation and identified system limitations:

Mobile Network Connectivity

Local network restrictions limit the operational range of the robot during its current execution. The integration of the mobile network device SIM7600E provides a dedicated static IP address connection that allows the robot to work outside local Wi-Fi networks. Using this upgrade the system would gain the ability to support extensive remote monitoring services and telepresence functions in physically separate locations.

Power System Optimization

Some power system specifications match the requirements but Full Power mode operation does not reach the expected 2-hour runtime period. Future product versions should work on developing powerful batteries together with advanced distribution circuits and superior power control algorithms. Running power states according to operational needs would help increase operation time without requiring heavier or bulkier battery components.

Communication Latency Reduction

The recorded control latency delivered 1 second measurements which exceeded the 300 milliseconds requirement mentioned in specifications. The system requires improved communication protocol optimization together with efficient data compression methods as well as potential hardware technology advancements to resolve this issue. An improvement in latency speeds would enhance both operator precision and enhance the entire user experience.

Elevated Camera and Audio System

Current sensor placement on ground level fails to provide sufficient performance in various operational conditions. The tower platform can be built adjustable and 1 meter high to refine the robot's visual perception and audio recording functions. The enhanced positioning system would bring the robot closer to human height in order to provide better involvement in video calls and remote monitoring operations. Servo motors steered by the Arduino system would let operators make distance-controlled modifications to camera positioning which would improve their visual monitoring capabilities.

Sustainable Power Solutions

Future designs require docking stations which support automated charging or stationary power connection to recharge their batteries. Through a combination of hybrid energy systems the robot could stay active indefinitely when stationed at specific spots without losing its capacity to move which suits prolonged field deployment needs.

Manipulation Capabilities

The addition of an elementary manipulator arm would upgrade the robot from its observational capabilities into an interactive remote presence technology. Through this upgrade the robot could conduct physical environmental tasks which would improve its functionality for managing remote objects and performing automated button operations and basic mechanical work. The manipulator interface should operate through the current web-based interface to offer straightforward access by users.

The proposed system enhancements overcome the present shortcomings and enable the establishment of new application possibilities. These improvements guide future advancements of telepresence capacity by retaining the base structure of the system while enhancing its functionality.

References

Jeremy Blum. 2019. Exploring Arduino. John Wiley Sons Inc.

Rishabh Jain. 2021. Advanced Home Automation Using Raspberry Pi. APRESS.

Mark Lutz. 2011. Programming Python. O'Reilly Media, Inc, USA.

Raspberry Pi community. 2023. The Official Raspberry Pi Handbook 2024: Astounding projects with Raspberry Pi computers. Publisher: Raspberry Pi Press.

Arduino Docs (2025) Available at: https://docs.arduino.cc/.

Camera - Raspberry pi Documentation (2024) Available at: https://www.merriam-webster.com/dictionary/divest.

Metal tank, гусеничное шасси 2WD (2025) Available at: https://roboparts.ru/catalog/shassi_i_ramy/gusenichnye_shassi/metal-tank/.

Mono audio amplifier module TDA2030A 18 watts into 8 ohms 6 - 12 V DC (2024) Available at: https://store.mectronica.it/en/amplifiers/1948-mono-audio-amplifier-module-tda2030a-18-watts-into-8-ohms-6-12-v-dc.html.

Raspberry pi 3 Documentation (2024) Available at: https://www.raspberrypi.com/documentation/accessories/camera.html.

Socket.IO (2025) Available at: https://socket.io/.

Telerobotics (2024) Available at: https://en.wikipedia.org/wiki/Telerobotics.

Temu (2025) Available at: https://temu.com/.

w3schools Online Web Tutorials (2025) Available at: https://www.w3schools.com/.

YD63-02 0.5W 8OHM - Datasheet (2024) Available at: https://www.utmel.com/productdetail/ruichi-yd630205w8ohm-68712671.

33GB-520, мотор с редуктором 6-12В 170-350 об/мин (2025) Available at: https://roboparts.ru/catalog/privody/dvigateli_s_reduktorom/motor-s-reduktorom-33gb-520-6-12v-170-350-ob-min/.