IN1901
Microcontroller Based Application Development Project

Final Report

RYTHMOBOT

Supervisor :- Mr. B. H. Sudantha

Group No : 22 **Group Name :** Circuit Builders

Index Number	Name
234063G	W.C.G.H.Fernando
2342219R	T.M.Warsavithana
234088L	H.Jannan
234051T	G.A.Dewduni
234135F	T.V.Namasena

Bachelor of Science Honours in Information Technology (BScHons (IT))
Faculty of Information Technology
University of Moratuwa

Table Of Content

1. Introduction

Music and movement have always shared a powerful bond—from classical performances to contemporary choreography. Today, as technology blends with creativity, robotics is becoming a new medium for artistic expression. Our project, **Dancing Robot**, explores this intersection by developing a robot that performs physical dance-like movements in response to **clap sounds** detected through a real-time sound sensor.

Unlike traditional robots that rely on pre-programmed movement sequences or complex audio processing, our robot introduces a **clap-controlled response system**. It listens for sharp sound pulses—like hand claps— through a **sound sensor module**, and clap triggers a predefined movement or dance step.

The **ESP32 microcontroller** serves as the central processing unit, controlling **13 servo motors (SG90 and MG995) and 4 DC motors** that move the robot's limbs in sync with detected claps. Visually, the robot becomes more captivating through the use of **user-controllable RGB LED strips**, which respond to movement cues **and allow real-time pattern selection through the web interface**, and an **LCD display**, which provides real-time feedback (such as motion status, song name)

An ultrasonic sensor, placed at the robot's neck like a pair of eyes, allows the robot to sense obstacles in front of it—adding a touch of personality and environmental awareness. Meanwhile, an IR sensor is attached to the base to detect nearby objects and prevent collisions during movement. These sensors help the base react to its environment, but do not affect the upper dancing robot. The integration of an **I2C module** simplifies hardware wiring, making the setup efficient and modular. To add motion and make the performance more dynamic, the robot is placed on a moving base. This base allows simple forward and backward movement, helping the robot appear more lifelike during dances.

A **JavaScript-based frontend interface** enables wireless connectivity, allowing users to control system behavior or trigger additional functions via Wi-Fi. Altogether, this system showcases how simple sound cues and smart hardware can create an interactive and entertaining robot that bridges learning, engineering, and fun.

2. Literacy survey

Dancing robots have been developed in various forms over the years, mainly for entertainment, educational projects, or competitions. Based on a study of existing implementations, most designs fall into two major categories:

2.1. Features of previous systems

1. Pre-Programmed Motion Robots :

These robots follow a fixed sequence of dance moves that are hardcoded into the system. They do not respond to their environment or external input. While simple, they offer little interactivity or adaptability.

2. Basic Sound Detection Robots :

These robots use microphones or basic sound sensors to detect loud sounds or music. However, their responses are generally based on volume thresholds and do not involve any interpretation of rhythm or tempo. As a result, their movement may appear random or out of sync.

2.2. Limitations in Previous Systems

- Hardware Limitations: Most prior designs were built using Arduino Uno or Mega boards, which have limited memory, processing speed, and GPIO pins. This restricts multitasking, servo control, and sensor integration.

- Limited Sound Interpretation: Sound detection is usually simplistic, reacting only to volume without identifying meaningful audio cues like claps.

- Lack of Interactivity: Few systems incorporate sensors to detect human presence or gestures.
- No Remote Access: Earlier designs rarely included web-based control systems or real-time feedback mechanisms.

3. Problem in Brief

Modern robotics often focuses on industrial efficiency and automation, with limited emphasis on its potential in creative domains such as entertainment and performance art. Most robotic systems rely on rigid, pre-programmed routines that lack real-time adaptability or human-like responsiveness— making them less engaging for interactive settings like exhibitions or stage shows. Many robots lack the ability to engage with human emotions through dynamic expressions such as music and dance, often relying on preprogrammed routines rather than adapting to real-time stimuli.

To address this, our project introduces a **Dancing Robot** that performs synchronized dance steps based on structured timing rather than complex audio analysis. we divide a selected song into **10 equal segments** and sends a signal via web socket **every 5 seconds**, aligning the robot's motions with the general rhythm of the music in a predictable loop. This approach simplifies implementation while still creating the illusion of musical responsiveness.

Each step is powered by precise control of **servo motors (SG90 and MG995) and DC motors** using an **ESP32 microcontroller**, delivering expressive limb movements. The dance sequence is enhanced with **RGB LED strip effects** to boost visual engagement.

Additionally, most previous dancing robots are stationary, which limits their expressiveness and stage presence. Without a mobile base, robots appear restricted and less dynamic. However, enabling mobility is challenging due to stability and balance issues, especially when multiple limbs are moving simultaneously. Our approach resolves this by fixing the robot to a wheeled base using a support pole, giving it movement while maintaining structural balance.

This solution demonstrates a cost-effective and imaginative way to integrate robotics into artistic experiences, making it ideal for educational demos, tech showcases, and creative events that aim to captivate and inspire audiences.

4. Proposed Solution

The proposed system is an interactive dancing robot that performs preprogrammed dance movements in response to environmental and use triggered inputs. Instead of complex audio or music analysis, the system relies on simple clap detection for activation, supported by sensors, visual effects, and feedback systems to enhance user engagement- showcasing how robotics can creatively blend technology with art to inspire and captivate audiences.

4.1. Features of the Proposed Solution

- **Sound-Responsive Dance Robot**
  Utilizes a sound sensor to detect clap inputs and automatically executes dance movements. When a clap is detected, the robot performs a choreographed dance routine, providing immediate visual feedback to user interaction.

- **Time-Synchronized Dance Using WebSocket Control**
  Instead of relying on advanced audio processing, the system divides a selected song into 10 equal time segments and sends signals via WebSocket every 5 seconds. This structured timing approach aligns the robot's movements with the

rhythm of the music in a repeatable, predictable loop—creating the illusion of beat-based dancing without complex analysis.

- **Remote Control via Web Interface**

  Users can control and monitor the robot through a browser-based dashboard, making it accessible from any device on the same Wi-Fi network.

- **Dynamic Song Upload and Management**

  The web interface features an intuitive drag-and-drop functionality that allows users to upload custom MP3 files directly into the system. Users can simply drag MP3 files from their computer into the designated upload zone on the web interface, and the system automatically processes and adds them to the available song list. Once uploaded, the robot can immediately dance to these custom songs using the same 10-segment timing synchronization system, providing unlimited musical variety for performances.

- **Visual and Audio Feedback**

  Combines customizable RGB lighting and buzzer tones for immersive feedback. Users can select from multiple LED patterns through the web interface, including rainbow cycles, breathing effects, strobe patterns, and solid colors. The selected lighting pattern is applied instantly via WebSocket communication. The buzzer sounds at the start and end of routines, as well as during each step transition, helping users stay in sync with the robot's movements while the personalized LED effects enhance the visual spectacle.

- **Interactive Presence Detection and Collision Avoidance**

  An ultrasonic sensor placed on the robot's neck acts like "eyes," helping it detect objects in front and react to its surroundings. When the sensor detects an object within range, the robot's base automatically stops moving to prevent collisions.

This gives the robot a more lifelike, interactive feel while ensuring safe operation by making it aware of nearby obstacles

- **Customizable Dance Routines**
  Pre-programmed moves can be easily modified or expanded, allowing flexibility in choreography based on user preference or music type.

- **Creative & Entertaining Use of Tech**
  Unlike many robots focused on utility, this project blends tech and creativity—perfect for performances, education, or showcasing engineering in a fun way.

- **Wheeled Base with Stability Support**
  Since humanoid robots often struggle with balance during movement, our design secures the dancing robot to a wheeled base using a vertical support pole. The base is powered by DC motors and allows forward and backward motion. This not only provides dynamic range to the performance but also ensures structural integrity during dance routines. The use of battery-powered mobility ensures that the robot is fully untethered and performance-ready.

5. Aim and Objectives

5.1. Aim

The primary aim of this project is to design and build an interactive Dancing Robot that performs synchronized movements based on timed control signals, rather than complex audio processing. By integrating mechanical actuation, sensor-based interaction, and a Wi-Fi-enabled web interface, the project seeks to blend creativity with embedded systems engineering, delivering an entertaining and engaging robotic performance experience.

5.2. Objectives

- **Develop a robot controlled by an ESP32 microcontroller** that can manage multiple hardware components including servo motors, motor drivers, LEDs, sensors, and a buzzer.

- **Implement time-based dance synchronization** by dividing songs into equal time segments and using WebSocket communication to control movement in real-time.

- **Enable hands-free activation** of the robot using a sound sensor to detect claps, allowing users to trigger performances intuitively.

- **Design a responsive web-based interface** that allows users to start/stop dances, monitor system status, and view live sensor data remotely over Wi-Fi.

- **Integrate RGB LEDs and an LCD display** to enhance the visual appeal and provide real-time status messages throughout the performance.

- **Uses an ultrasonic sensor** positioned at the robot's neck as robot "eyes" to detect obstacles ahead. When an obstacle is detected, the robot automatically stops its base movement to avoid collisions, enhancing both interaction and safety.

- **Provide clear & unique audio feedback** using a buzzer to signal the start and end of dance routines, as well as step transitions, so users can track movement timing even without visual cues.

- **Build a motorized base** using DC motors and a support pole to enable safe, forward and backward movement during performances, enhancing mobility without compromising robot balance.

- **Enable dynamic song expansion** through a user-friendly drag-and-drop interface that allows real-time MP3 file uploads, expanding the robot's musical repertoire beyond pre-programmed songs and providing unlimited creative possibilities for performances.

5.2.1. Hardware Development

1. Mechanical Assembly:

- Design and construct a stable robot chassis using **PLA (Polylactic Acid)** material, providing lightweight yet durable construction suitable for dynamic movement.
- Mount **12 servo motors** at key joints: 6 servos for arms (shoulders, biceps, forearms) and 6 servos for legs (hips, knees, ankles) to allow smooth and precise movement during dance routines.
- Integrate a **stabilizing pole** into the design to enhance structural stability and ensure balanced operation during dance movements, particularly when responding to motion commands or operating on various surfaces.
- Mount an **ultrasonic sensor** on the robot's neck to act as "eyes" for obstacle detection.

2. Electronic Integration:

- Wire 12 servo motors, RGB LEDs, ultrasonic sensor, sound sensor, motor drivers, and buzzer to the **ESP32 microcontroller** using a systematic layout.
- Use **Adafruit PWM Servo Driver (PCA9685)** with I2C communication (address 0x40) to control all 12 servos efficiently.
- Implement efficient power distribution using an LM2596 DC-DC buck converter to regulate 12V battery power to stable 5V for PCB components, ensuring optimal power efficiency and thermal management.

- Integrate **16x2 LCD display** with I2C communication (address 0x27) for real-time status display.
- The robot operates with a **5V 20A power supply** system, providing sufficient power for all servo motors, LEDs, and electronic components during continuous operation.

3. Sensor Integration:

- **Ultrasonic Sensor (HC-SR04)**: Positioned on robot's neck for obstacle detection within 0-20cm range, automatically stopping base movement when obstacles detected.
- **Sound Sensor**: Implements clap detection with edge detection (HIGH to LOW transition) and 500ms debounce delay to trigger special dance response movements.
- **Motor Driver**: Controls base movement with forward/backward cycles (5 cycles each direction) with 2-second intervals.

5.2.2. Software Implementation

1. ESP32 Firmware Architecture:

- **Non-blocking state machine** design for efficient multitasking
- **WebSocket server** running on port 81 for real-time communication with web interface
- **Multi-sensor integration** with continuous monitoring (ultrasonic: 100ms intervals, sound sensor: continuous)
- **PWM servo control** with smooth movement interpolation and home position management

2. Dance Move Algorithms:

- **8 predefined dance steps** with coordinated arm and leg movements
- **Modular step functions** (step1() through step8()) each with specific servo angle configurations
- **Smooth servo transitions** with configurable step delays and interpolation
- **Home position reset** after each dance step for consistency
- **Special clap response sequence** with 5-movement choreography independent of main dance routine

3. Music Synchronization System:

- **Simplified rhythm synchronization** that divides selected songs into **10 equal segments**
- **WebSocket communication** sends step signals **every 5 seconds**
- **Predictable loop system** cycling through 8 dance steps repeatedly throughout song duration
- **State management** for play/pause/stop/resume operations with appropriate servo responses

4. Safety and Interaction Features:

- **Obstacle avoidance**: Ultrasonic sensor stops base movement and head motor when objects detected within 20cm
- **Clap detection**: Sound sensor triggers special 5-step dance sequence with unique sound effects
- **Multi-modal feedback**: Buzzer plays different melodies for WiFi connection, clap response, dance steps, and system states

- **LCD status display**: Real-time updates showing song name, playback status, and obstacle warnings

5. Frontend Web Application:

- **Pure JavaScript implementation**
- **Drag-and-drop MP3 upload interface allowing users to add custom songs by simply dragging files into the upload zone**
- **Dynamic song management system that automatically processes uploaded MP3 files and adds them to the selection dropdown**
- **Song selection dropdown displaying both pre-loaded and user-uploaded song names**
- **LED pattern control interface with dropdown menus for lighting effects**
- **HTML5 audio player with play/pause/stop controls supporting both default and uploaded songs**
- **WebSocket client connecting to ESP32 on port 81**
- **Real-time status display showing current playback state and active LED pattern**

6. Communication Protocol:

- **WebSocket-based real-time communication** between web app and ESP32
- **Command structure**:
    - 1-8: Trigger specific dance steps
    - song:[name]: Start new song with name display on LCD
    - paused: Pause dance and display pause status
    - resume: Resume dance maintaining song context

    - stop: Stop dance and reset all servos to home positions

7. Timing and Control System:

- **5-second interval timing** for step signals (matching song segmentation)
- **Non-blocking loop architecture** allowing simultaneous:
    - WebSocket message handling
    - Continuous clap detection
    - Obstacle monitoring (100ms intervals)
    - Base movement control (2-second cycles)
    - Head motor rotation (1-second intervals, 45°↔0° oscillation)
    - LCD animation updates (400ms intervals)

8. Audio Feedback System:

- **Unique sound signatures** for different events:
    - WiFi connection: Victory fanfare melody
    - Clap detection: Ascending celebration melody
    - Dance steps: Musical scale tones (C4 to C5)
    - System states: Distinct melodies for pause/resume/stop
    - Obstacle detection: Warning beeps

9. Error Handling and State Management:

- **Debounced clap detection** preventing multiple triggers within 500ms
- **Obstacle detection override** pausing base and head movement during detection

- **Movement conflict prevention** stopping normal operations during clap response sequence
- **WebSocket connection monitoring** with appropriate error responses
- **Servo position tracking** maintaining current angles for smooth transitions

10. Technology Stack:

- **Hardware**: ESP32 microcontroller, PCA9685 PWM driver, HC-SR04 ultrasonic sensor, sound sensor
- **Communication**: WebSocket protocol for real-time bidirectional communication
- **Frontend**: Pure JavaScript with HTML5 audio API
- **Servo Control**: 12-channel PWM with I2C communication
- **Display**: I2C LCD with animated status updates

11. LED Pattern Management System:

- **Real-time pattern switching** via WebSocket commands from web interface
- **Multiple predefined patterns**: Rainbow cycle, breathing effect, strobe, solid color modes
- **Smooth pattern transitions** without interrupting dance movements
- **Pattern synchronization** with dance steps and music timing
- **WebSocket command processing** for instant LED pattern updates
- **Memory-efficient pattern storage** using FastLED library functions

5.2.3. Testing and Optimization

1. Stability Testing:

- Ensure the robot remains balanced on various surfaces during complex moves using the integrated stabilizing pole.
- Test structural integrity of the PLA chassis under different movement conditions and servo loads.
- Validate base movement cycles (forward/backward) maintain stability during dance sequences.

2. Fluidity of Movements:

- Fine-tune servo speeds with configurable step delays (default 10ms) for smooth, natural motions.
- Test smooth movement interpolation between current and target servo positions.
- Validate home position resets maintain consistent starting points for each dance step.

3. Synchronization Accuracy:

- Test 5-second interval timing system across different song durations and genres.
- Ensure WebSocket communication latency doesn't affect dance step timing.
- Validate the 8-step cycle repeats correctly throughout song playback.

4. Sensor Performance Testing:

- **Ultrasonic sensor**: Test obstacle detection accuracy within 0-20cm range across different surface materials.
- **Sound sensor**: Validate clap detection with various clap intensities and background noise levels.

- **Response time testing**: Ensure sensors respond within acceptable timeframes (ultrasonic: 100ms, clap: immediate).

5. Communication Reliability:

- Test WebSocket connection stability during extended dance sessions.
- Validate command processing for all message types (steps 1-8, song, paused, resume, stop).
- Test error handling for disconnection and reconnection scenarios.

6. Code Optimization:

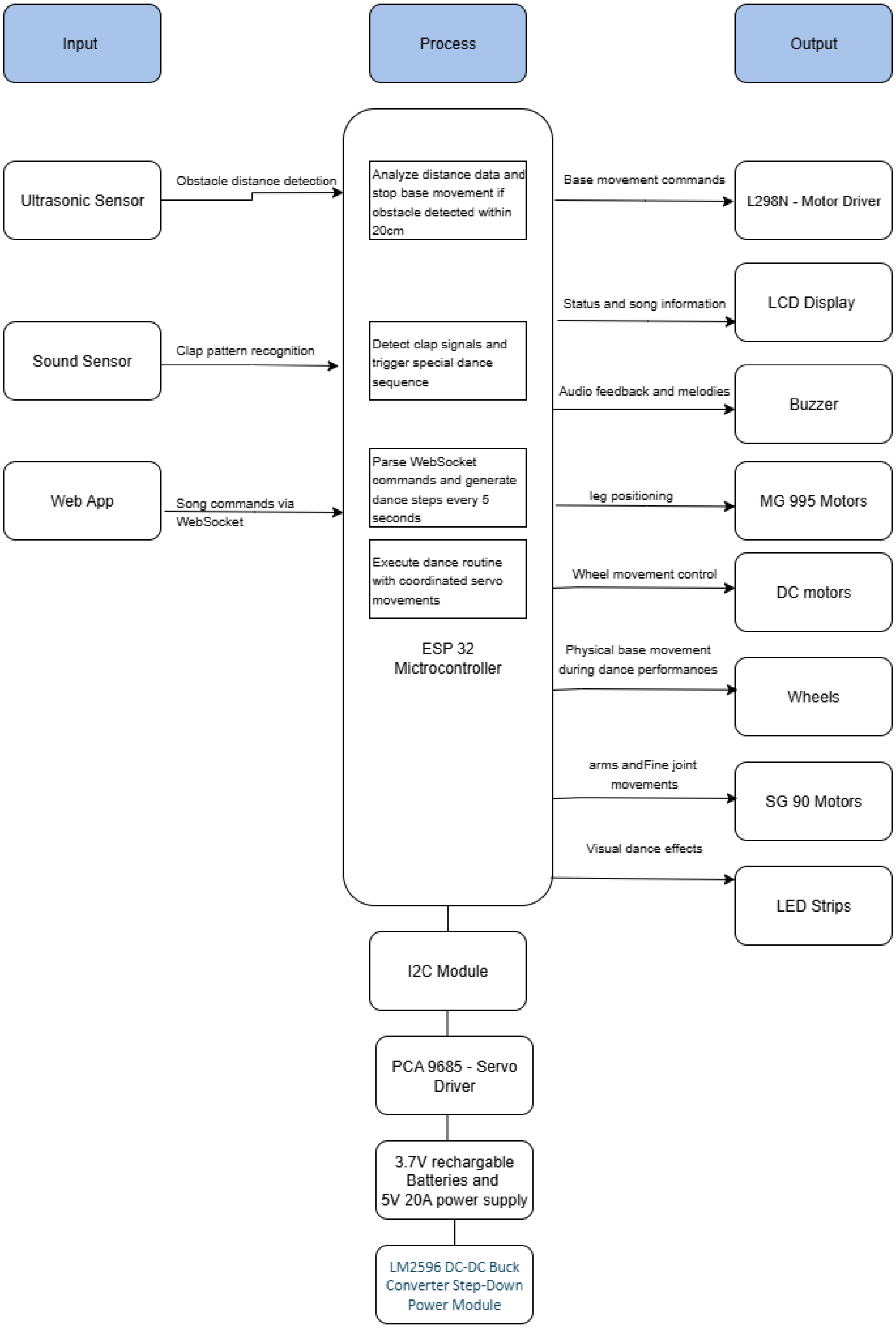
- Optimize non-blocking loop architecture to prevent delays in sensor monitoring.
- Enhance WebSocket message processing efficiency for real-time responsiveness.
- Validate memory usage and prevent potential crashes during extended operation.

7. Iterative Improvements:

- Gather feedback on dance movement quality and timing synchronization.
- Test durability of PLA chassis and servo mounting points under repeated use.
- Refine clap detection sensitivity and obstacle detection thresholds based on real-world testing.

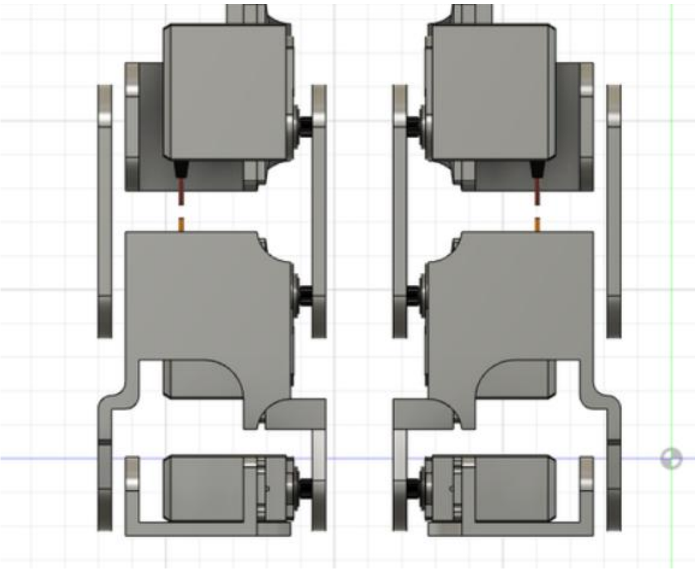
6. Analysis and Design

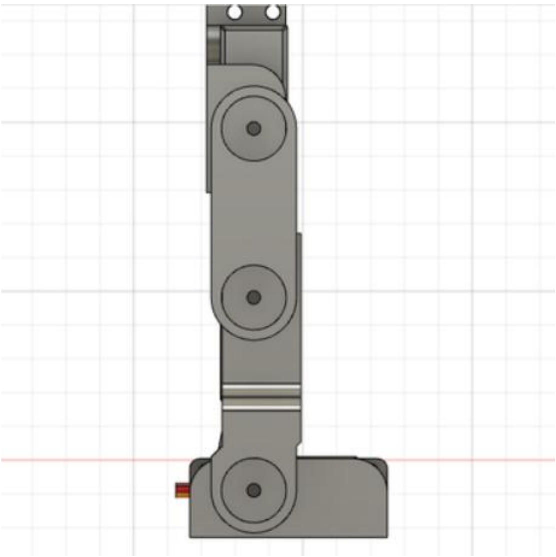
6.1. The basic block diagram

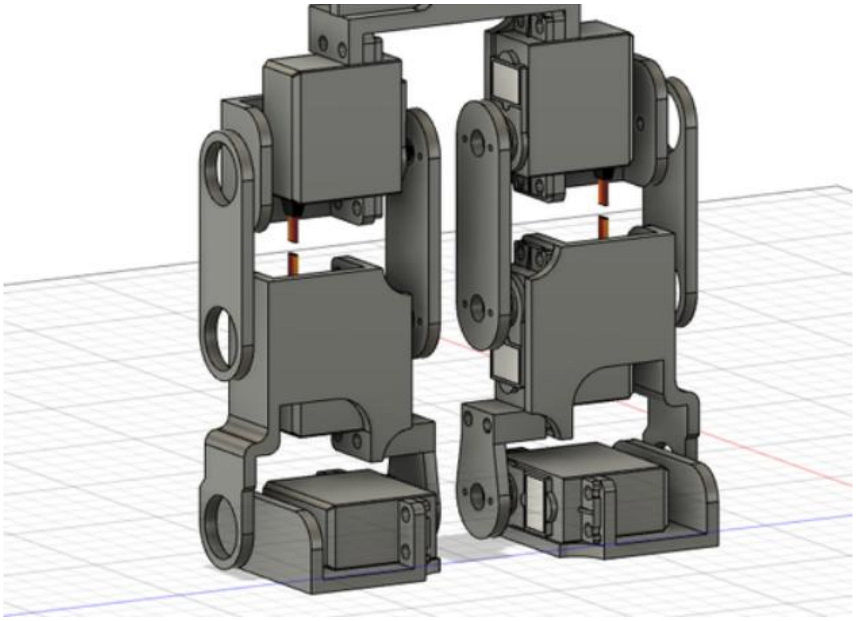

Input
Process
Output
Ultrasonic Sensor
Obstacle distance detection
Analyze distance data and stop base movement if obstacle detected within 20cm
Base movement commands
L298N - Motor Driver
Status and song information
LCD Display
Sound Sensor
Clap pattern recognition
Detect clap signals and trigger special dance sequence
Audio feedback and melodies
Buzzer
Web App
Song commands via WebSocket
Parse WebSocket commands and generate dance steps every 5 seconds
leg positioning
MG 995 Motors
Execute dance routine with coordinated servo movements
Wheel movement control
DC motors
ESP 32 Mictrocontroller
Physical base movement during dance performances
Wheels
arms andFine joint movements
SG 90 Motors
Visual dance effects
LED Strips
I2C Module
PCA 9685 - Servo Driver
3.7V rechargable Batteries and 5V 20A power supply
LM2596 DC-DC Buck Converter Step-Down Power Module

6.2. The 3D design

Legs

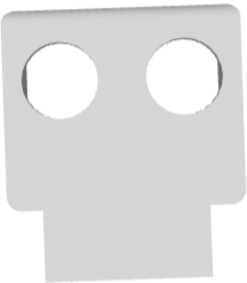




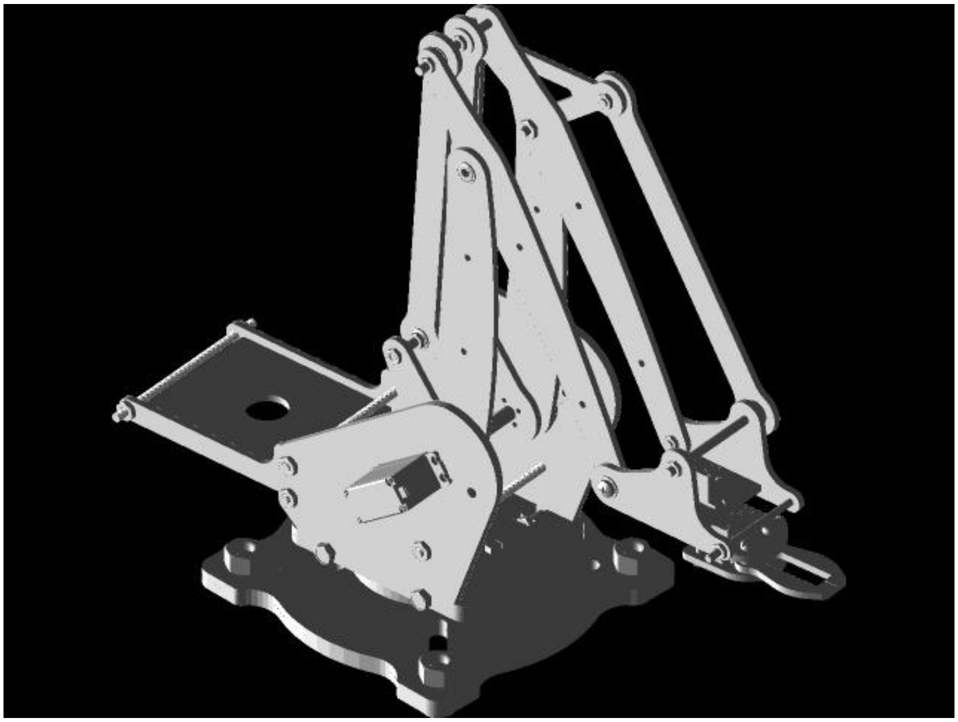


Body

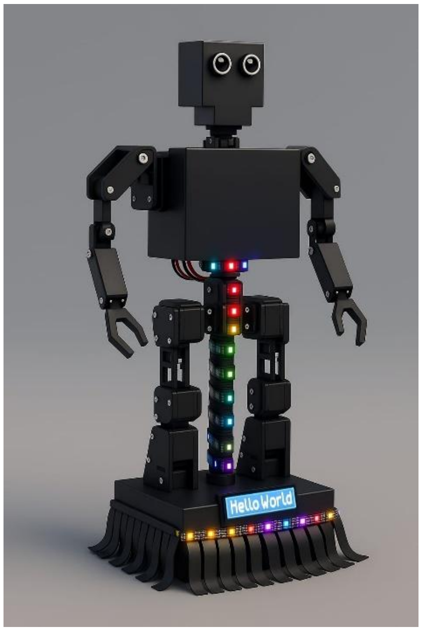
Head



Arms (2D)



Final Design

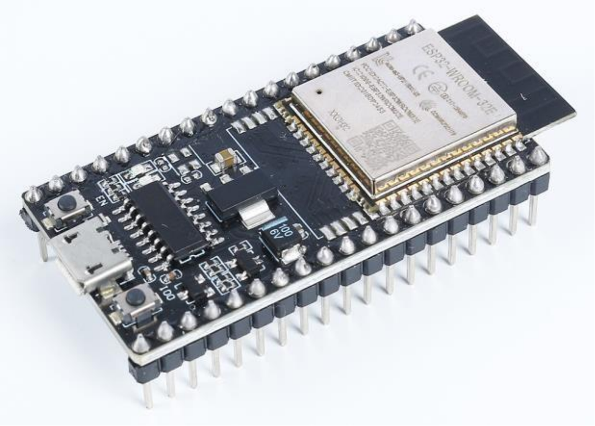


7. Components

7.1. ESP32 Microcontroller

The ESP32 is a powerful and versatile microcontroller used as the brain of the robot. It handles all the logic, sensor readings, WebSocket communication, and motor control. Its built-in Wi-Fi capability enables the web-based control interface, while its dual-core processor allows real-time multitasking.

- **Processor:** Tensilica Xtensa Dual-Core 32-bit LX6
- **Clock Speed:** Up to 240 MHz
- **Memory:** 520 KB SRAM + 4MB Flash
- **Connectivity:** Wi-Fi (802.11 b/g/n), Bluetooth 4.2 (Classic + BLE)
- **GPIO Pins:** 34 GPIOs (varies by board model)
- **Operating Voltage:** 3.3V
- **Power Supply:** 5V via USB or battery input (e.g., VIN pin)



7.2. SG90 & MG995 Servo Motors

The robot uses a combination of SG90 and MG995 servo motors to control limb movements like the arms, legs, and head. SG90s are ideal for lightweight, fast movement, while MG995s offer higher torque for more stable or heavier motion.

• **SG90 Servo:**

❖ **Weight:** 9 g

❖ **Dimension:** 22.2 × 11.8 × 31 mm (approx.)

❖ **Stall Torque:** 1.8 kgf.cm

❖ **Operating Speed:** 0.1 s/60°

❖ **Operating Voltage:** 4.8 V (~5V)

❖ **Dead Band Width:** 10 µs

❖ **Temperature Range:** 0 ºC – 55 ºC o **Pulse Positioning:**

"0" Position: 1.5 ms pulse (center)

"+90°" Position: ~2 ms pulse (rightmost)

"–90°" Position: ~1 ms pulse (leftmost)

• **MG995 Servo:**

❖ **Weight:** 55 g

❖ **Dimension:** 40.7 × 19.7 × 42.9 mm (approx.)

❖ **Stall Torque:** 8.5 kgf·cm (4.8 V), 10 kgf·cm (6 V)

❖ **Operating Speed:** 0.2 s/60º (4.8 V), 0.16 s/60º (6 V)

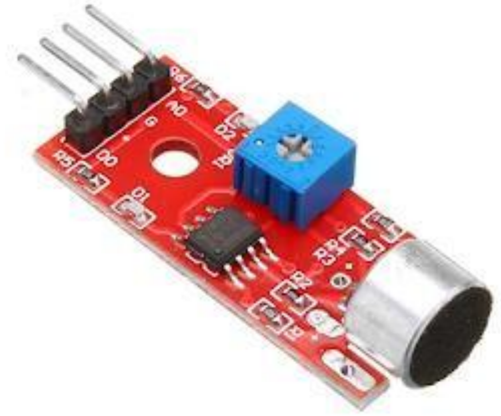
- **Operating Voltage:** 4.8 V to 7.2 V
- **Dead Band Width:** 5 µs
- **Design:** Stable and shock-proof double ball bearing
- **Temperature Range:** 0 ºC – 55 ºC
- **Model Number:** 31150-MP
- **Rotation Angle:** 120° (±60° from center)
- **Gears:** Metal gears for longer life

These motors execute timed movement patterns based on incoming control signals.

7.3. Sound Sensor (Clap Detection Module)

The sound sensor detects loud sounds like claps and converts them into electrical signals sent to the ESP32 for processing. When a clap is detected, it triggers a special short movement sequence on the robot.

- **Sensitivity:** Adjustable via onboard potentiometer
- **Output:** Analog(AO)
- **Operating Voltage:** 3.3V–5V



7.4. Ultrasonic Sensor (HC-SR05)

Used to detect obstacles within a 20cm range in front of the robot. When an obstacle is detected, the robot automatically stops its base movement and head motor rotation to prevent collisions, ensuring safe operation during dance performances.

- **Trigger Pin Format:** 10 uS digital pulse
- **Sound Frequency:** 40 kHz
- **Echo Pin Output:** 0-Vcc
- **Echo Pin Format:** output is DIGITAL and directly proportional with range. See our conversion formula above.
- **Measurement Range:** 2cm to ~4.5m
- **Measurement Resolution:** 0.3cm
- **Measurement Angle:** up to 15 deg
- **Measurement Rate:** 40 Hz
- **Supply Voltage:** 4.5V to 5.5V
- **Supply Current:** 10 to 40mA
- **Connector:** standard 5-pin male connector which can plug directly into breadboards.
- **Static current :** less than 2mA
- **Detection distance:** 2cm-450cm

7.5. Buzzer

The buzzer provides audio cues to users by emitting a short beep at the start and end of a dance routine and when a step changes. It offers quick auditory feedback about the system’s state.

- **Operating Voltage:** 4V - 8V
- **Type:** Active
- **Rated Voltage:** 6V DC
- **Operating Voltage:** 4 to 8V DC
- **Rated Current:** ≤30mA
- **Sound Output at 10cm:** ≥85dB
- **Resonant Frequency:** 2300 ±300Hz
- **Tone:** Continuous
- **Operating Temperature:** -25°C to +80°C
- **Storage Temperature:** -30°C to +85°C
- **Weight:** 2g
- **Use Case:** Dance routine start/end and steps changing signals

7.6. RGB LED Strips (WS2812 or Similar)

These add visual flair to the robot’s performance by flashing and changing colors during the dance. The LEDs can be programmed to match the timing segments.

- **Type:** Addressable RGB (e.g., WS2812)
- **Power Supply:** 5V
- **Control:** Single digital pin using libraries like FastLED
- **Colors:** Full 24-bit color support (16.7 million colors)

7.7. LCD Display with I2C Module (16x2 LCD)

The LCD provides textual feedback like current song , “playing”, “paused” etc. The I2C interface reduces wiring complexity and pin usage on the ESP32.

- **Compatible with** Arduino Board or other controller boards with I2C bus
- **Operating Voltage:** 5V
- **Display Type:** Negative white on Blue backlight
- **I2C Address:** 0x38–0x3F (default 0x3F)
- **Interface:** I2C to 4-bit LCD data and control lines
- **Contrast Adjustment:** Built-in potentiometer
- **Backlight Control:** Firmware or jumper wire
- **Board Size:** 80 × 36 mm
- **Use Case:** Real-time system status display

7.8. Power Supply

In this project, a 5V 20A SMPS (Switched Mode Power Supply) with a metal casing was used to power the entire dancing robot system. This power supply delivers stable 5V with current capability up to 20A, which is sufficient to simultaneously operate multiple components including the ESP32 microcontroller, 12 servo motors, ultrasonic sensor, sound sensor, motor drivers for base movement, I2C LCD display, and buzzer system, LED strips. The SMPS is efficient, compact, and energy-efficient, making it ideal for continuous dance performances. The metal casing provides effective heat dissipation and protection against overheating, ensuring reliable and safe operation during extended dancing sessions across various performance conditions

7.9. DC Motors

DC motors are used to move the base of the robot. They help the robot go forward, backward, and turn. A motor driver connects them to the ESP32 and controls the speed and direction. Wheels are attached to help the robot move smoothly.

- **Operating Voltage:** 4.5V - 9V
- **Weight:** 17.5g
- **Operating Temperature:** -10°C ~ +60°C
- **Rated Voltage:** 6.0VDC
- **Rated Load:** 10 g*cm
- **No-load Current:** 70 mA max
- **No-load Speed:** 9100 ±1800 rpm
- **Loaded Current:** 250 mA max
- **Loaded Speed:** 4500 ±1500 rpm

- **Starting Torque:** 20 g*cm
- **Starting Voltage:** 2.0
- **Stall Current:** 500mA max
- **Body Size:** 27.5mm x 20mm x 15mm
- **Shaft Size:** 8mm x 2mm diameter

7.10. L298N Motor Driver

The L298N motor driver module connects the DC motors to the ESP32 for controlling the robot's wheeled base movement system. It manages both the speed and direction of the wheels by controlling the motor phases through digital signals (IN1, IN2, IN3, IN4). The L298N also provides electrical isolation and protection for the ESP32 by handling the higher current requirements of the motors, preventing damage to the microcontroller's GPIO pins. The driver enables the robot to perform coordinated base movements during dance routines, moving the wheeled base forward for 5 cycles and backward for 5 cycles with 2-second intervals, adding dynamic movement to the dance performance.

- **Input Voltage:** 3.2V ~ 40V DC

Brief Data:

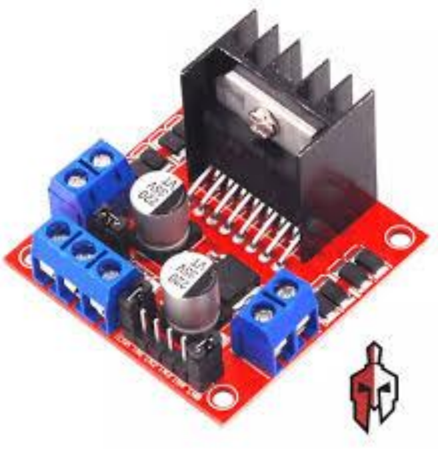
- **Driver:** L298N Dual H Bridge DC Motor Driver
- **Power Supply:** DC 5V - 35V
- **Peak Current:** 2 Amp
- **Operating Current Range:** 0 ~ 36mA
- **Control Signal Input Voltage Range:**

  Low: -0.3V ≤ Vin ≤ 1.5V

  High: 2.3V ≤ Vin ≤ Vss
- **Enable Signal Input Voltage Range:**

  Low: -0.3V ≤ Vin ≤ 1.5V (control signal invalid)

  High: 2.3V ≤ Vin ≤ Vss (control signal active)
- **Maximum Power Consumption:** 20W (when the temperature T = 75 ℃)
- **Storage Temperature:** -25 ℃ ~ +130 ℃
- **On-board +5V Regulated Output Supply:** Supplies controller board (e.g., Arduino)
- **Size:** 3.4 cm × 4.3 cm × 2.7 cm



7.11. PCA9685 Servo Driver

The PCA9685 helps the ESP32 control many servo motors at once. It uses only two wires (I²C) to send signals to all the motors. This makes it easy to move different parts of the robot at the same time.

- **Controller IC:** PCA9685
- **Power Supply Voltage (VCC):** 3.3V – 5V (logic level)
- **Servo/Motor Power (V+):** 4.8V – 6V (external power source for servos)
- **Communication Interface:** I²C (2-wire: SDA & SCL)
- **PWM Output Channels:** 16 independent channels
- **PWM Resolution:** 12-bit (4096 steps per cycle)
- **Frequency Range:** Adjustable from 24 Hz to 1526 Hz
- **Output Drive Capability:** Can directly drive up to 16 servos or LEDs
- **Current per I/O pin:** 10 mA (sink or source)
- **I²C Address:** Default 0x40 (can be changed via address pins up to 62 devices on the same bus)
- **Onboard Oscillator Frequency:** 25 MHz
- **Connection Type:** Standard 4-pin I²C header (GND, VCC, SDA, SCL)
- **Dimensions:** Approx. 62 mm × 25 mm × 12 mm
- **Mounting Holes:** Yes (typically 4)
- **Extra Features:**
  - **Onboard power LED**
  - **Terminal block or header for external servo power**
  - **Stackable design** with I²C address jumpers

7.12. 3.7V Rechargeable Batteries

The 3.7V rechargeable battery powers the l298N motor driver and DC motors at the robot’s base. It supplies the necessary energy to drive the wheels while keeping the ESP32 safe from high current demands.

- **Chemistry:** Lithium-Ion
- **Nominal Voltage:** 3.7 V
- **Capacity:** 2.6 Ah (2600 mAh)
- **Max Charge Voltage:** 4.25 V
- **Max Continuous Discharge Current:** 2.5 A
- **Overcharge Protection:** Yes (detects at 4.25 V)
- **Over discharge Protection:** Yes (detects at 2.5 V)
- **Internal Resistance:** ≤ 65 mΩ
- **Size:** 19 mm diameter × 69 mm length
- **Weight:** 48 g
- **Charge Temp Range:** 0°C to 52°C
- **Discharge Temp Range:** -20°C to 60°C

+ DE 18650 1800mAh 3.7V -
+ DE 18650 1800mAh 3.7V -
+ DE 18650 1800mAh 3.7V -

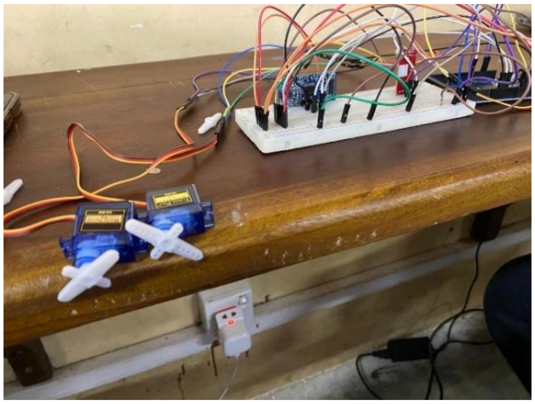
7.13. LM2596 DC-DC Buck Converter Step-Down Power Module

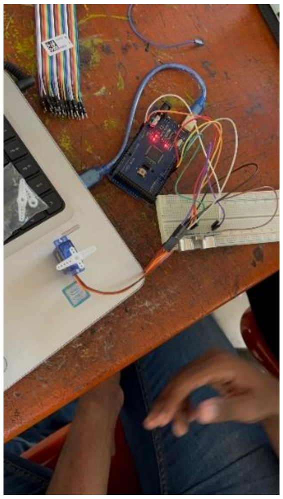
The LM2596 DC-DC Buck Converter Step-Down Power Module serves as the primary power regulation stage in the system, efficiently converting the 12V battery pack voltage down to a stable 5V supply for the PCB and other electronic components. This switching regulator topology offers high efficiency (typically 85-92%) compared to linear regulators, minimizing power loss and heat generation while maximizing battery life. The LM2596 can handle input voltages from 4V to 40V and deliver up to 3A of output current, providing ample margin for the system's power requirements. Its built-in thermal shutdown and current limiting features ensure reliable operation under varying load conditions, making it well-suited for portable battery-powered applications where power efficiency and thermal management are critical considerations.

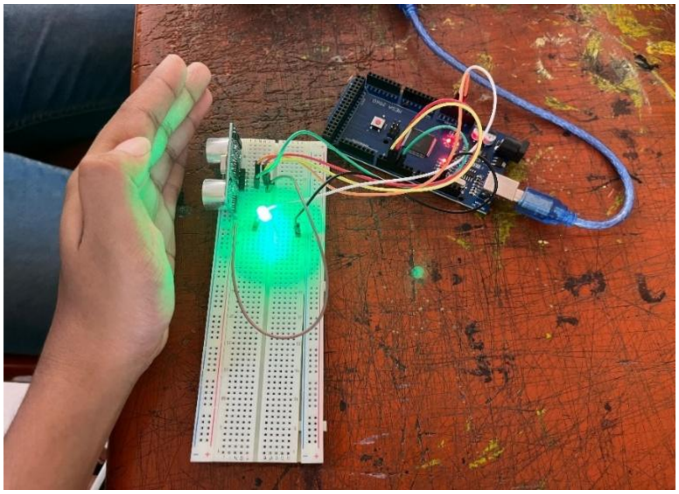
8. Testing and Implementation

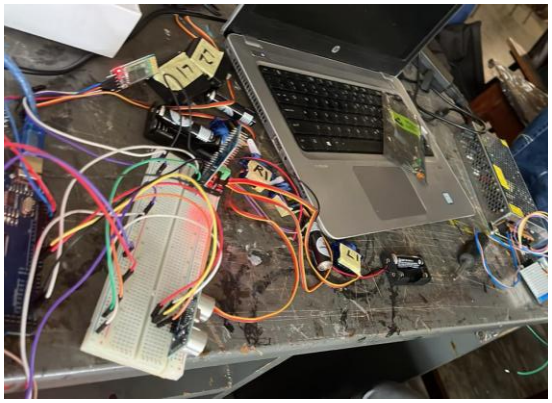
connecting to the power supply and software development

Checking ultrasonic sensor

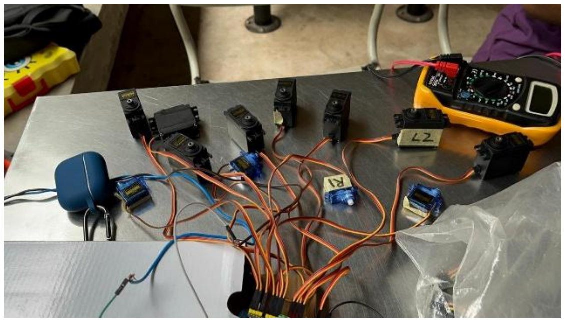






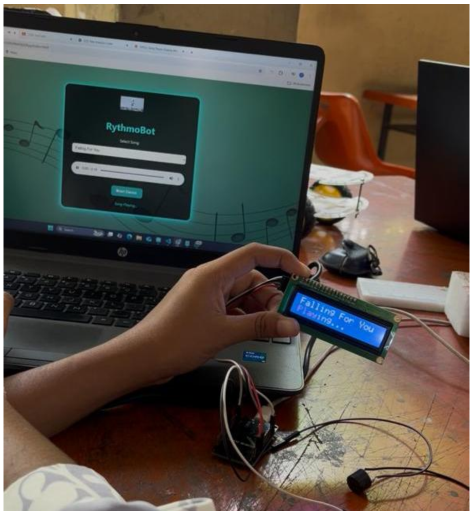


Connecting all motors to motor driver

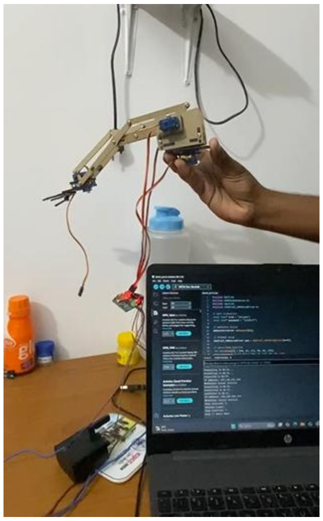


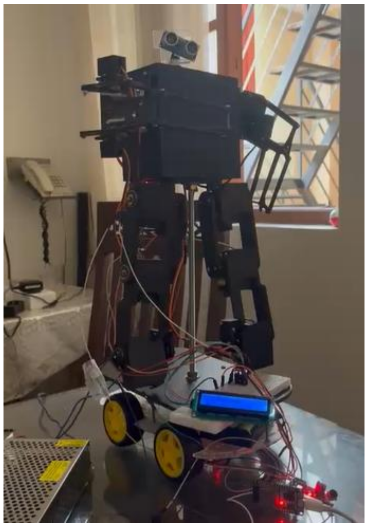
Base with wheels and motor driver connections

Connecting LCD screen and checking buzzer

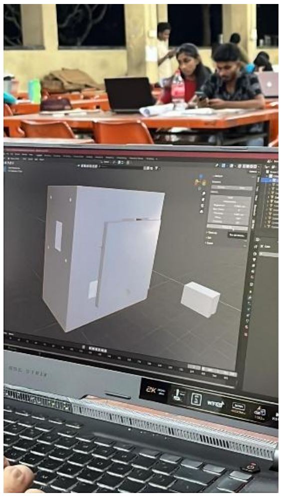


Assembling the arm mechanism

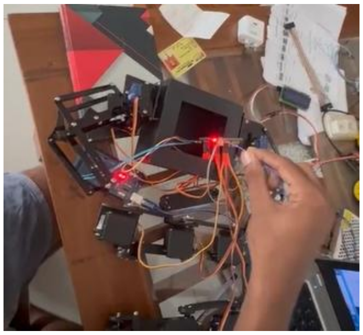




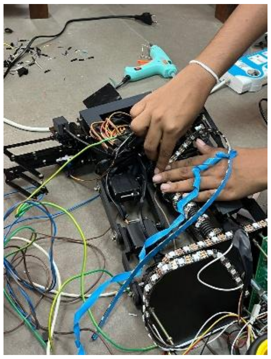
3D Designing Part and assembling

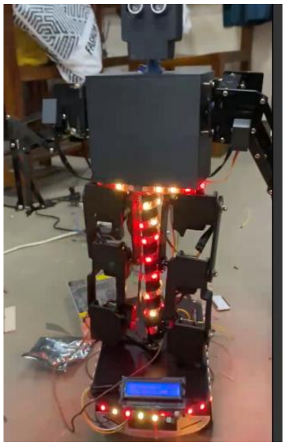


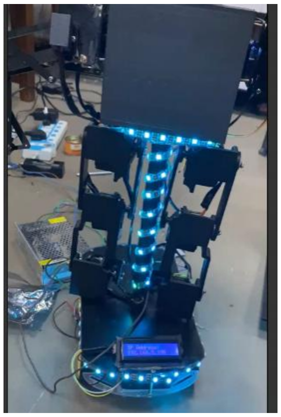
Programming dance movements



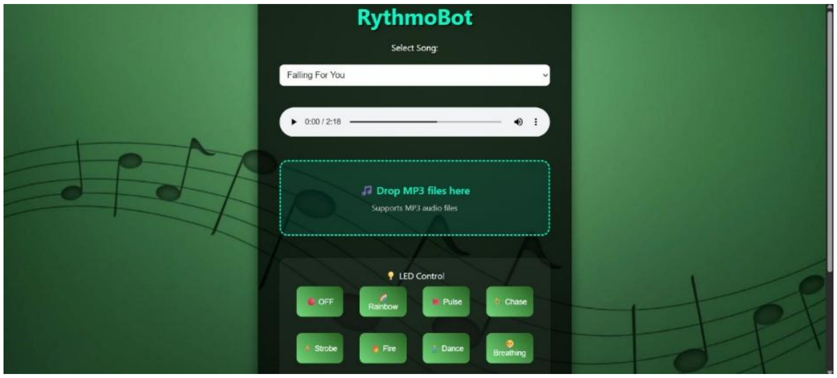
LED patterns checking and arranging



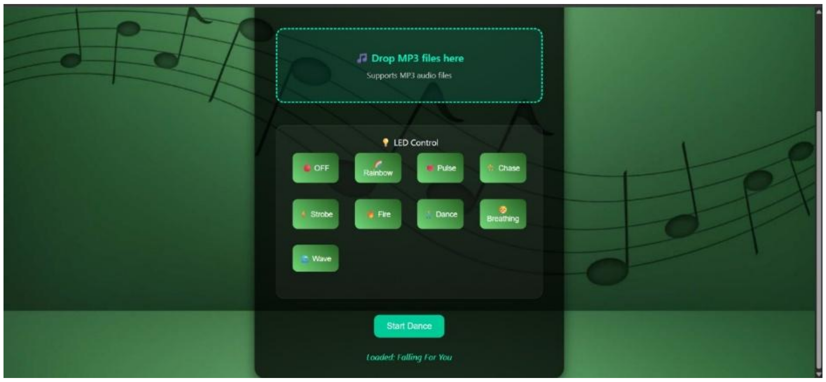




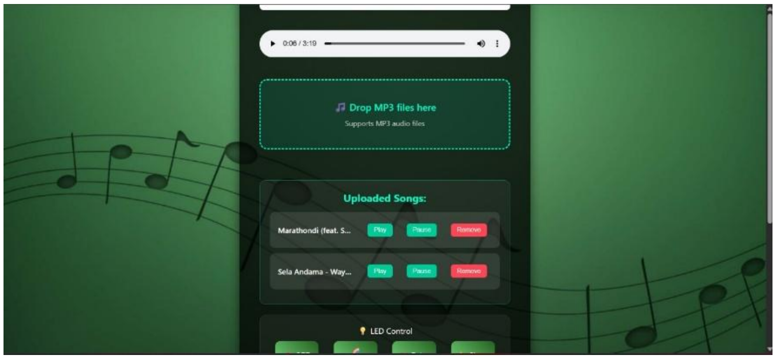
Implemented web interface



Select Song: Falling For You



Loaded: Falling For You



0:06 / 3:19

Drop MP3 files here

Supports MP3 audio files

Uploaded Songs:

Marathondi (feat. S...

Sela Andama - Way...

LED Control

9.Future Enhancements

1. Full-Body Balance & Walking

Upgrade from a fixed base to a fully mobile bipedal system that can actually walk and dance without external support using advanced motion control and balance algorithms.

2. Gesture-Based Controls

Use hand gestures or body movements (via a camera or IR sensors) to control the robot’s dance or trigger reactions instead of just relying on claps or the web interface.

3. AI-Based Beat Detection

Integrate real-time music analysis using AI/ML to detect tempo, rhythm, and beat patterns—so the robot can actually dance to live music .

4. Voice Control Integration

Add voice assistant compatibility (like Google Assistant or Alexa) so users can say “Start dancing!” or “Change style!” to control the bot hands-free.

5. Dance Style Selection

Add different dance modes—hip hop, ballet, K-pop, etc.—with unique pre-programmed moves for each. Users can choose through the web UI.

6. Face Tracking & Interaction

Use a camera module to recognize and follow people’s faces, making the robot respond to viewers with movements, expressions, or light shows.

7. Self-Charging Dock

Build a return-to-charging station feature (like robot vacuums) so the robot can recharge itself automatically when low on battery.

8. Mobile App Control

Develop a mobile app for iOS/Android that lets users control the robot remotely, customize moves, and even upload new dance routines.

9. Sound-Responsive LED Visuals

Let the RGB LEDs respond dynamically to music frequencies—like a mini light show synced to the beat.

10.Individual Contribution

234063G :

PCB Design using EasyEDA:

My main task was to design the custom printed circuit board (PCB) using the EasyEDA software. This board had to connect and support all the important electronic parts like the ESP32 microcontroller (the brain), the PCA9685 servo controller (to manage the servo motors), the L298 motor driver (to control the base motors), sensors, buzzer, LCD display, and the DC motors. I planned the layout carefully to make sure that power was supplied correctly to all components, signals between parts were clean and didn’t interfere with each other, and that everything fit neatly inside the robot’s body. This involved arranging parts logically and creating proper paths for the electrical connections.

L298 Motor Driver Setup and Testing:

The L298 motor driver controls the two DC motors located in the robot’s base. These motors help the robot move around and keep balanced while dancing. I set up this motor driver circuit and wrote tests to check that it could control the motors smoothly without jerky or uneven movements. I also made sure the motor driver did not overheat and that it supplied steady power to avoid sudden stops or power loss during operation. This step was important for the robot to move reliably without hardware issues.

Final Assembly and Wiring Management:

In the last stage, I took charge of putting everything together—mounting the PCB and all the hardware components firmly inside the robot frame. I carefully arranged the wires and cables so they were neat, safe from damage, and did not cause electrical interference between parts. Good cable management also made it easier to troubleshoot and maintain the robot later. This careful assembly helped improve the overall durability and performance of the robot, ensuring it worked well during testing and demonstrations.

2342219R :

Web Application Development:

I developed the web application using Node.js, HTML, and CSS to create a simple and easy-to-use interface for controlling the robot. The app lets users select music tracks and send commands wirelessly over Wi-Fi to start or change the robot’s dance moves. This remote control feature made interacting with the robot much smoother and more flexible. I focused on designing a clean layout and ensuring the app responded quickly to user inputs, so the robot would react almost instantly when commands were sent.

I implemented a WebSocket-based system to enable instant, bidirectional communication between the browser and ESP32, ensuring zero-latency control similar to platforms like WhatsApp and Slack. The robot’s dance moves are driven by intelligent step signals, with adaptive patterns for different songs, random or sequential moves, and precise 5-second timing for perfect synchronization. A modern file management system allows drag-and-drop uploads with automatic MP3 validation and clear error messages for a smooth user experience.

Buzzer Integration and Testing:

I integrated the buzzer into the robot's system and wrote the code to trigger sounds during specific actions, such as starting, stopping, or signaling errors. The buzzer adds an audio layer to the robot's interaction, making it more engaging and easier to understand what the robot is doing. I conducted thorough testing to make sure the buzzer sounds happened exactly at the right moments and didn't interfere with the robot's movements or overall performance.

I2C Communication Management:

A large part of my work focused on managing the I2C communication protocol between the ESP32 and its connected devices, especially the PCA9685 servo controller and the LCD display. I designed and implemented communication routines to transfer commands and data efficiently over the I2C bus, which uses just two wires to handle multiple devices. This required careful testing to ensure that data was sent and received correctly without errors or delays. Proper I2C communication was crucial because any delay or failure could cause the servos to move out of sync or the display to show wrong information, which would disrupt the robot's performance.

234088L :

ESP32 Microcontroller Programming:

I was in charge of programming the ESP32 microcontroller, which is the brain of the robot. My main task was to develop the firmware that reads input from various sensors, processes this data, and sends control signals to the servo motors. This allows the robot to perform coordinated dance movements with precise timing. I wrote code to handle multiple servos simultaneously, ensuring that each motor moved exactly when needed to create smooth and natural motions. This involved careful planning of timing and synchronization so the robot's movements would not lag or become jerky.

PCA9685 Servo Controller Management:

A significant aspect of my responsibility involved managing the communication between the ESP32 microcontroller and the PCA9685 servo controller via the I2C protocol. I developed and implemented the necessary code to transmit precise control signals to the PCA9685, which is capable of operating up to 16 servo motors concurrently. This setup enabled the robot to execute complex and synchronized movements, as the PCA9685 facilitated smooth and simultaneous control of multiple servos. I ensured that the servos responded accurately to the control commands, thereby contributing to the robot's fluid and natural dance performance.

Buck Converter:

As part of my individual contribution, I integrated the LM2596 DC-DC Buck Converter Step-Down Power Module into the system to provide stable and efficient power delivery. I selected this module to step down the 12V battery pack voltage to a regulated 5V supply for the PCB and other electronic components. Its high efficiency (85–92%) reduced heat generation and extended battery life, making it ideal for our portable, battery-powered design. I also ensured the module was configured to meet the system's maximum current requirements (up to 3A) and verified its built-in thermal shutdown and current limiting features to guarantee safe operation under varying loads.

234135F :

Sound Sensor (Clap Detection and Movement Trigger):

I implemented a comprehensive clap detection system using a digital sound sensor connected to pin 34. The system continuously monitors the sensor state and detects claps by identifying the transition from HIGH to LOW signals. To ensure reliable operation, I implemented a debounce mechanism with a 500ms delay to prevent multiple false detections. When a clap is detected, the robot executes a sophisticated 5-phase response sequence including enthusiastic arm waving, synchronized leg movements, arm crossing patterns, asymmetric leg positioning, and a final victory pose. The system also integrates with the LED Pattern and LCD display to display a “Clap detected” Additionally, I programmed a unique ascending musical melody that plays during the clap response, making the interaction more engaging and dynamic.

LED Strips (Advanced Visual Effects System):

I developed a comprehensive LED control system managing 60 WS2812B LEDs connected to pin D2 using the FastLED library. The system features 11 distinct lighting patterns including rainbow cycling, breathing effects, chase patterns, fire simulation, dance synchronization, obstacle warnings, and wifi success sequences. Each pattern is optimized with specific update intervals (ranging from 30ms to 200ms) for smooth animations. The LED system is fully integrated with the robot's state machine, automatically switching patterns based on whether the robot is dancing, paused, clap detection or detecting obstacles. I also added WebSocket commands for remote LED control, allowing users to change patterns wirelessly.

Web Application Development (Full-Stack Interface):

I created a complete web-based control system with both frontend and backend components. The frontend features a responsive HTML interface with drag-and-drop MP3 file upload functionality, real-time song selection, LED pattern control buttons, and live status updates. I Make a drag and drop section for add music files mp3 for users. The interface supports both pre-loaded songs and user- uploaded MP3 files, with automatic file validation and duplicate detection. I styled the typography with modern font stacks, text shadows, and responsive sizing, while creating gradient LED buttons with smooth CSS transitions for a professional, contemporary appearance.

234051T :

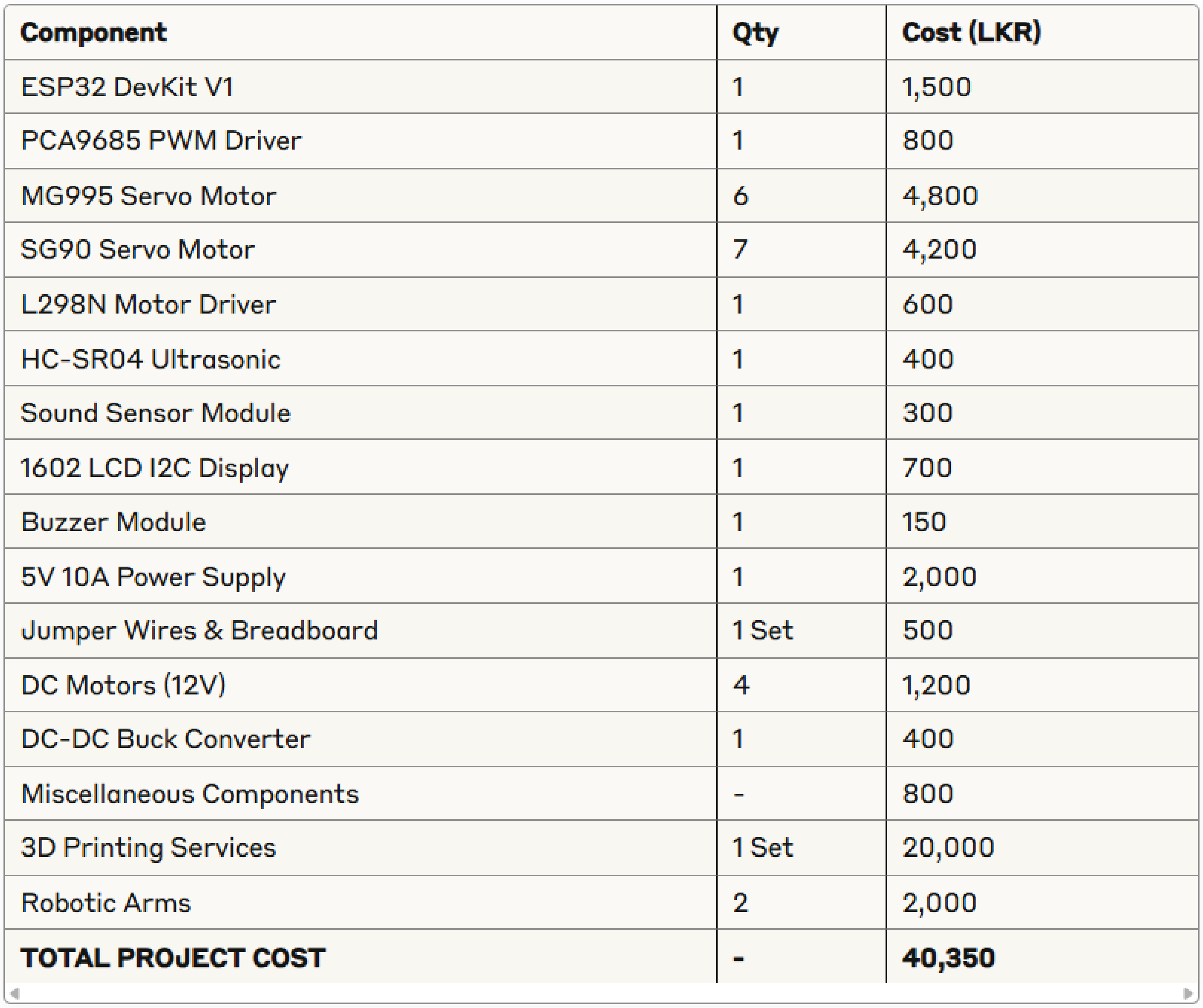
Ultrasonic Sensor Integration:

I was responsible for integrating the ultrasonic sensor with the ESP32 microcontroller, which allowed the robot to measure distances and detect obstacles in real time. The sensor works by sending high-frequency sound waves from the TRIG pin, which bounce back from objects and are received by the ECHO pin. By calculating the time it takes for the echo to return, the sensor determines the distance to obstacles. In our robot, the sensor performed multiple functions. it prevented collisions by controlling the base motors to stop or redirect movement, stopped the head motor and adjusted its orientation when obstacles were detected, activated LEDs to provide visual alerts, and sounded the buzzer as an audible warning. I wrote the code to process these readings accurately, implemented calibration routines for consistent performance in different environments, and conducted multiple rounds of testing to minimize errors caused by surface texture, ambient noise, or lighting conditions.

3D Model Design:

In addition to integrating the sensors, I designed the robot’s physical structure using SolidWorks, employing parametric modeling to ensure precise dimensions and accurate alignment of all electronic components. The 3D model was carefully developed to maintain mechanical stability, provide adequate support for motors and sensors, and enable smooth, coordinated movements of all limbs. Bearings were added at key joints to reduce friction and ensure smoother motion. The design process involved multiple iterations, during which the dimensions, balance, and structural integrity of the robot were refined to optimize performance, facilitate assembly, and ensure reliable operation during dynamic motion.

11.Cost Estimate



Component	Qty	Cost (LKR)
ESP32 DevKit V1	1	1,500
PCA9685 PWM Driver	1	800
MG995 Servo Motor	6	4,800
SG90 Servo Motor	7	4,200
L298N Motor Driver	1	600
HC-SR04 Ultrasonic	1	400
Sound Sensor Module	1	300
1602 LCD I2C Display	1	700
Buzzer Module	1	150
5V 10A Power Supply	1	2,000
Jumper Wires & Breadboard	1 Set	500
DC Motors (12V)	4	1,200
DC-DC Buck Converter	1	400
Miscellaneous Components	-	800
3D Printing Services	1 Set	20,000
Robotic Arms	2	2,000
TOTAL PROJECT COST	**-**	**40,350**

12.References

- ElectronicWings. (n.d.). HC-05 Bluetooth Module. Retrieved from https://www.electronicwings.com/sensors-modules/bluetooth-module-hc-05-
- Espressif Systems. (n.d.). ESP32-WROOM-32. Retrieved from https://www.espressif.com/en/products/modules/esp32-wroom-32
- Texas Instruments. (n.d.). L298N Dual H-Bridge Motor Driver. Retrieved from https://www.ti.com/lit/ds/symlink/l298.pdf
- Samsung SDI. (n.d.). INR18650-26J. Retrieved from https://www.samsungsdi.com/lithium-ion-battery/inr18650-26j.html
- Texas Instruments. (n.d.). LM2596 Adjustable Output Regulator. Retrieved from https://www.ti.com/lit/ds/symlink/lm2596.pdf
- Futaba S3003 Servo. (n.d.). Retrieved from https://www.futabausa.com/products/s3003/
- Parallax Inc. (n.d.). Ping))) Ultrasonic Distance Sensor. Retrieved from https://www.parallax.com/product/ping-ultrasonic-distance-sensor/
- Adafruit Industries. (n.d.). NeoPixel Digital RGBW LED Strip. Retrieved from https://www.adafruit.com/product/284
- Newhaven Display. (n.d.). 16x2 LCD Module. Retrieved from https://www.newhavendisplay.com/
- https://www.micros.com.pl/mediaserver/M_HY-SRF05_0003.pdf
- https://www.datasheethub.com/ky-038-lm393-sound-detection-module/?utm_source=chatgpt.com
- https://www.alldatasheet.com/datasheet-pdf/pdf/1179113/WORLDSEMI/WS2812B.html
- https://www.waveshare.com/w/upload/e/ea/PCA9685.pdf
- https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
- https://www.handsontec.com/dataspecs/L298N%20Motor%20Driver.pdf
- https://www.handsontec.com/dataspecs/module/I2C_1602_LCD.pdf
- https://www.farnell.com/datasheets/2171929.pdf

- https://www.theengineeringprojects.com/2020/09/lm2596-buck-converter-datasheet-pinout-features-applications.html