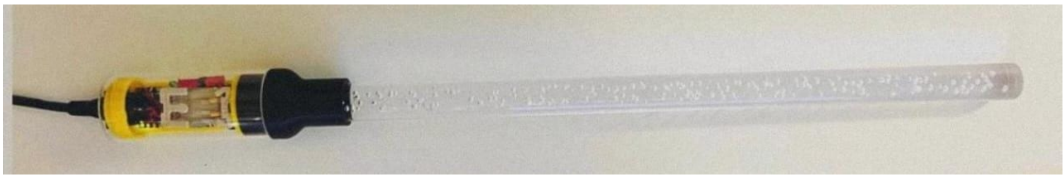

Manufacturing and Assembly

Magic Wand Using Arduino Nano 33 BLE Sense

Author:	Adhiraj Walse
Matriculation No.:	7025711
Author:	Sudeshna Nanda
Matriculation No.:	7026003
Author:	Srikanth Nanda
Matriculation No.:	7026002
Course of Studies:	Business Intelligence and Data Analytics

First examiner:	Prof. Dr. Elmar Wings
Submission date:	January 30, 2025

University of Applied Sciences Emden/Leer · Faculty of Technology · Mechanical Engineering Department
Constantiaplatz 4 · 26723 Emden · http://www.hs-emden-leer.de

Inhaltsverzeichnis

1. Acronyms 5

Acronyms 5

I. Manufacturing and Assembly 7

2. Main Function 9
- 2.1. Structural Design and Fabrication 9
  - 2.1.1. Wand Body Design 9
  - 2.1.2. Material Selection 9
  - 2.1.3. Fabrication Process 9
- 2.2. Component Integration and Circuit Assembly 10
  - 2.2.1. Electronic Components 10
  - 2.2.2. Circuit Assembly 10
- 2.3. Software and Functional Testing 10
  - 2.3.1. Code Upload and Gesture Recognition 10
  - 2.3.2. LED Control 11
  - 2.3.3. Debugging and Optimization 11
- 2.4. Final Assembly and Enclosure Sealing 11
  - 2.4.1. Component Placement 11
  - 2.4.2. Securing Components 12
  - 2.4.3. Sealing the Wand 12
- 2.5. Aesthetic Finishing and Quality Control 12
  - 2.5.1. Painting and Decoration 12
  - 2.5.2. Quality Checks 12

3. Sketch 13
- 3.1. Components Required 13
- 3.2. Assembly Plan 13
- 3.3. Circuit Diagram 13

4. List Of Parts 15
- 4.1. List of Parts 15
  - 4.1.1. Assembled Parts 15
  - 4.1.2. Manufactured Parts 15
- 4.2. List of Items 15

5. Set Up 17
5.1. Detailed Description of Assembly 17
5.1.1. Prepare the Board 17
5.1.2. Upload the Firmware 17
5.1.3. Integrate the Sensors and Components 18
5.1.4. Testing the Gesture Recognition 18
5.1.5. Final Assembly 18

6. Functions 19
6.1. Assembly Functions 19
6.1.1. Microcontroller Integration 19
6.1.2. Firmware Installation 19
6.1.3. Sensor Configuration 19
6.1.4. External Component Assembly 19
6.1.5. Final Casing Assembly 19
6.2. Manufacturing Functions 20
6.2.1. PCB and Component Sourcing 20
6.2.2. Casing Fabrication 20
6.2.3. Circuit Integration 20
6.2.4. Testing and Calibration 20
6.2.5. Quality Control and Packaging 20

7. Maintenance 21
7.1. Hardware Maintenance 21
7.1.1. Avoiding Physical Damage 21
7.1.2. Cleaning and Dust Protection 21
7.1.3. Power Management 21
7.2. Sensor Calibration and Maintenance 21
7.2.1. IMU (Inertial Measurement Unit) Calibration 22
7.2.2. Temperature and Humidity Sensor Care 22
7.3. Software and Firmware Maintenance 22

8. Safety Guidelines 23

9. Troubleshooting 25
9.1. Troubleshooting 25
9.1.1. The Magic Wand can't turn on (i.e. Power LED or Orange Programmable LED are not working) 25
9.1.2. I am not getting the gesture which I waved 26
9.1.3. Is it wireless, Can I connect it with Bluetooth? 26
9.1.4. How to turn on/turn off the magic wand? 26

10.Add Ons 27
10.1. Project Folder Structure 27
10.2. File Descriptions 27

1. Acronyms

Teil I.

Manufacturing and Assembly

2. Main Function

This document provides a detailed guide to the manufacturing and assembly of a Magic Wand using the Arduino Nano 33 BLE Sense. The project leverages the onboard IMU (Inertial Measurement Unit) and machine learning to detect motion gestures, which can be mapped to different magical spells or actions.

The Main Functions of the Assembly and Manufacturing Process for the Magic Wand.

2.1. Structural Design and Fabrication

2.1.1. Wand Body Design

- Use CAD software (Fusion 360, SolidWorks, Tinkercad) for modeling.
- Design includes:
  - Hollow tube to house components.
  - Openings for LEDs, buttons, and charging ports.
  - Secure space for wiring and batteries.

2.1.2. Material Selection

- **3D Printing**: PLA or ABS plastic for lightweight, durable structure.
- **Woodworking**: Carved wooden body for an authentic look.
- **Acrylic/PVC**: Tubes for a transparent or modern design.

2.1.3. Fabrication Process

- 3D print in two halves for easy component integration.
- If using wood, drill a hollow channel for internal components.
- Sand and smooth edges for comfort and aesthetics.

2.2. Component Integration and Circuit Assembly

2.2.1. Electronic Components

- **Microcontroller**: Arduino Nano 33 BLE Sense.
- **Motion Sensor**: Built-in LSM9DS1 (9-axis IMU).
- **Power Source**: Li-Po battery (3.7V, 500mAh+) or AAA with step-up converter.
- **Lighting**: WS2812 (Neopixel) LED strip or single RGB LED.
- **Switch/Button**: Tactile push-button or capacitive touch sensor.
- **Vibration Motor (Optional)**: For haptic feedback.

2.2.2. Circuit Assembly

- Wire the LED strip to the Arduino (D6 for data, 5V power, GND).
- Connect the button to a digital input pin with a pull-up resistor.
- Integrate the power supply (Li-Po with charging module or step-up circuit).
- Solder and insulate connections with heat shrink tubing.

2.3. Software and Functional Testing

2.3.1. Code Upload and Gesture Recognition

Upload firmware to Arduino using the Arduino IDE. The IMU reads motion data and triggers LED effects.

```
#include <Arduino_LSM9DS1.h>

void setup() {
    Serial.begin(115200);
    if (!IMU.begin()) {
        Serial.println("IMU␣initialization␣failed!");
        while (1);
    }
}

void loop() {
    float ax, ay, az;
    if (IMU.accelerationAvailable()) {
        IMU.readAcceleration(ax, ay, az);
        if (ax > 1.5) {
            Serial.println("Right␣Swipe␣Detected!");
        }
    }
    delay(100);
}
```

Listing 2.1: Gesture Detection Code

Reference to the code: See Listing 2.1.

2.3.2. LED Control

Different spells are represented by different colors.

```
#include <Adafruit_NeoPixel.h>

#define LED_PIN 6
#define NUM_LEDS 5

Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800)
   ;

void setup() {
    strip.begin();
    strip.show();
}

void loop() {
    strip.fill(strip.Color(255, 0, 0), 0, NUM_LEDS);
    strip.show();
    delay(500);

    strip.fill(strip.Color(0, 0, 255), 0, NUM_LEDS);
    strip.show();
    delay(500);
}
```

Listing 2.2: LED Animation Code

Reference to the code: See Listing 2.2.

2.3.3. Debugging and Optimization

- Use Serial Monitor to check real-time IMU data.
- Adjust motion detection thresholds.
- Optimize LED animations for smooth transitions.

2.4. Final Assembly and Enclosure Sealing

2.4.1. Component Placement

- Insert Arduino and battery securely into the wand body.

- Align buttons and LEDs with pre-cut openings.
- Neatly route wires to prevent obstruction.

2.4.2. Securing Components

- Use hot glue, foam padding, or brackets to hold parts.
- Ensure easy battery access for charging or replacement.

2.4.3. Sealing the Wand

- Attach final cap using screws, magnets, or snap-fit design.
- Glue wooden or acrylic parts carefully for a secure fit.

2.5. Aesthetic Finishing and Quality Control

2.5.1. Painting and Decoration

- Apply wood stain, metallic paint, or matte finish.
- Use engravings, decals, or resin casting for aesthetics.
- Wrap handle in leather or grip tape for better handling.

2.5.2. Quality Checks

- Verify gesture recognition accuracy.
- Test LED animations and haptic feedback.
- Check battery life and power efficiency.
- Perform durability tests.

3. Sketch

3.1. Components Required

- Arduino Nano 33 BLE Sense
- LiPo Battery (or AAA with Boost Converter)
- RGB LED / NeoPixel
- Vibration Motor (optional)
- Push Button (optional)
- Thin Copper Wire
- Wand Casing (Wooden or 3D-Printed)

3.2. Assembly Plan

1. Mount the Arduino securely inside the wand near the grip.
2. Connect the battery and voltage regulator (if needed).
3. Place RGB LEDs at the tip and route wires through the wand.
4. Attach a vibration motor near the grip (optional for haptic feedback).
5. Install a button on the side for user interaction (optional).
6. Manage the wires using thin insulated cables for a compact fit.
7. Secure the components with glue or 3D-printed holders inside the wand.

3.3. Circuit Diagram

The basic wiring schematic is shown below:

The magic wand is assembled with motion-based interaction and LED effects using the Arduino Nano 33 BLE Sense. Further programming can be done to integrate spell recognition through gesture sensing.

4. List Of Parts

4.1. List of Parts

4.1.1. Assembled Parts

Pre-manufactured components that are integrated into the wand:

- Arduino Nano 33 BLE Sense – Microcontroller with built-in sensors for motion and gesture recognition.
- LiPo Battery / Battery Pack – Power source (3.7V LiPo or AAA with a boost converter).
- RGB LED / NeoPixel – For magical lighting effects at the tip of the wand.
- Vibration Motor (Optional) – Provides haptic feedback for spellcasting.
- Push Button (Optional) – Allows user interaction for specific spell triggers.
- Thin Copper Wires – Connects components within the wand.
- Resistors & Voltage Regulator (if needed) – Ensures correct voltage supply to components.

4.1.2. Manufactured Parts

Custom-made or externally fabricated components:

- Wand Casing – Made of wood, plastic, or 3D-printed material to house all components.
- Internal Mounts / Holders – 3D-printed or handcrafted supports for securing components inside the wand.
- Custom PCB (Optional) – If a compact circuit board is designed for better integration.
- Enclosure for Battery – A secure holder for battery placement within the wand.

4.2. List of Items

Item	**Description**	**Price**	**Source**
Arduino Nano 33 BLE Sense	Microcontroller with built-in IMU & BLE	$35	Amazon
USB Cable	Micro USB cable for power & programming	$5	Amazon
3D-Printed Case	Custom wand enclosure	$10	Custom
RGB LED	Optional lighting effect	$3	Amazon
Vibration Motor	Optional haptic feedback	$4	Amazon

Tabelle 4.1.: List of necessary components for assembly

5. Set Up

5.1. Detailed Description of Assembly

5.1.1. Prepare the Board

1. Connect the **Arduino Nano 33 BLE Sense** to your computer using a **micro USB cable**.

2. Open the **Arduino IDE** and ensure the board is recognized:
   - Go to `Tools` → `Board` → Select **Arduino Nano 33 BLE Sense**.
   - Go to `Tools` → `Port` → Select the correct COM port.

3. If required, install the board package:
   - Open **Boards Manager** in Arduino IDE.
   - Search for `Arduino mbed OS Nano Boards` and install it.

5.1.2. Upload the Firmware

1. Install necessary libraries via the **Arduino Library Manager**:
   - `TensorFlowLite` (for ML model execution)
   - `Arduino_LSM9DS1` (for IMU sensor integration)

2. Open the firmware source file:
   - Navigate to `src/main.cpp`.
   - Load the firmware sketch into the Arduino IDE.

3. Verify the code by clicking the **Checkmark icon**.

4. Upload the firmware by clicking the **Upload (→) icon**.

5. Open the **Serial Monitor** (`Tools` → `Serial Monitor`) and set baud rate to `115200`.

5.1.3. Integrate the Sensors and Components

- **Onboard IMU (LSM9DS1) Setup**
  - The **Inertial Measurement Unit (IMU)** detects motion and gestures.
  - No additional wiring is needed as it is built into the board.
- **Optional External Components**
  - **RGB LEDs** for visual feedback:
    - Connect Red, Green, and Blue (RGB) pins to digital PWM-capable pins on the Arduino.
    - Use a common ground (GND).
  - **Vibration Motor** for haptic feedback:
    - Connect the positive terminal to a PWM digital pin.
    - Use a transistor and a diode for safe operation.
    - Connect the negative terminal to GND.

5.1.4. Testing the Gesture Recognition

1. Open the **Serial Monitor** to observe real-time sensor data.
2. Run the command:

```
edge−impulse−run−impulse
```

3. Observe classification results and verify correct gesture recognition.
4. If necessary, retrain or fine-tune the machine learning model.

5.1.5. Final Assembly

1. Carefully place the **Arduino Nano 33 BLE Sense** inside the wand casing.
2. Ensure the **USB port remains accessible** for firmware updates and charging.
3. Securely route any external component wiring inside the casing.
4. Seal the casing while allowing necessary access points for LEDs or buttons.
5. Perform a final functionality test to confirm all components work as expected.

6. Functions

6.1. Assembly Functions

6.1.1. Microcontroller Integration

- Mount the Arduino Nano 33 BLE Sense inside the wand casing.
- Ensure the USB port remains accessible for firmware updates and charging.

6.1.2. Firmware Installation

- Load the gesture recognition firmware onto the microcontroller.
- Install necessary libraries (TensorFlowLite, Arduino_LSM9DS1, etc.).

6.1.3. Sensor Configuration

- Utilize the onboard IMU (LSM9DS1) for motion detection.
- Calibrate the IMU to detect specific gestures accurately.

6.1.4. External Component Assembly

- Install RGB LED for visual feedback.
- Set up the vibration motor for haptic feedback.
- Secure wiring inside the casing.

6.1.5. Final Casing Assembly

- 3D-print or fabricate a custom wand enclosure.
- Assemble and seal the casing while keeping essential ports accessible.

6.2. Manufacturing Functions

6.2.1. PCB and Component Sourcing

- Procure Arduino Nano 33 BLE Sense and other electronic components.
- Source RGB LEDs, vibration motors, and connectors from reliable suppliers.

6.2.2. Casing Fabrication

- Use 3D printing or injection molding to create the wand shell.
- Choose durable and lightweight materials for the enclosure.

6.2.3. Circuit Integration

- Design and assemble the wiring layout for components.
- Ensure proper soldering and insulation to prevent short circuits.

6.2.4. Testing and Calibration

- Run diagnostics on the IMU sensor to verify gesture detection accuracy.
- Test LED and vibration feedback to confirm responsiveness.

6.2.5. Quality Control and Packaging

- Conduct functional tests on each assembled unit.
- Package the wand with necessary documentation and accessories (USB cable, instructions).

7. Maintenance

7.1. Hardware Maintenance

7.1.1. Avoiding Physical Damage

- **Use a protective casing:** Shield the Arduino board and sensors from dust and physical impacts.

- **Secure all connections:** Ensure wires, sensors, and external modules are firmly connected to prevent loose wiring issues.

7.1.2. Cleaning and Dust Protection

- **Compressed air:** Use compressed air to remove dust from the board and sensors.

- **Antistatic brush:** Gently brush dirt away from the PCB and sensor components.

- **Isopropyl alcohol (IPA 99%):** Dampen a microfiber cloth with IPA and gently clean the PCB, avoiding direct contact with sensors.

7.1.3. Power Management

- **Use a quality power source:** Avoid using cheap USB adapters to prevent voltage fluctuations.

- **Battery care:** If using a Li-Po battery, ensure it is not overcharged or discharged below 3.3V to prolong its lifespan.

7.2. Sensor Calibration and Maintenance

The Arduino Nano 33 BLE Sense includes various sensors such as the IMU, microphone, and temperature sensor. Periodically recalibrating the sensors ensures accurate readings.

7.2.1. IMU (Inertial Measurement Unit) Calibration

- **Run the self-test:** Use the self-test function available in most IMU libraries.
- **Allow stabilization:** Keep the wand stationary for a few seconds after powering it on to allow the IMU to stabilize.
- **Manual offset calibration:** Log raw accelerometer and gyroscope values, and adjust offsets in the code.

7.2.2. Temperature and Humidity Sensor Care

- **Avoid moisture exposure:** High humidity may affect sensor accuracy.
- **Reset the sensor:** Restart the board and reinitialize the sensor if readings drift.

7.3. Software and Firmware Maintenance

- **Update firmware and libraries:** Ensure you are using the latest version of Arduino firmware and sensor libraries.
- **Run diagnostic tests:** Periodically check sensor outputs using diagnostic scripts.
- **Log and analyze data:** Store sensor logs to detect any unusual trends or malfunctioning over time.

8. Safety Guidelines

Important Safety Instructions

1. Please read the Safety Instructions before using your Magic Wand.
2. Heed all warnings.
3. Clean only with hygienic towel.
4. Use attachments and accessories only from reputable manufacturers.
5. Avoid placing near any heat sources, including stoves, heat registers, radiators, and other appliances that emit heat, including amplifiers.
6. **Object or liquid entry WARNING:** Avoid exposing the apparatus to rain or water, including near bathtubs, sinks, or pools. Do not place liquid-filled objects on it, as this may cause electric shock.
7. This apparatus uses power banks. Check with local authorities for proper disposal or recycling guidelines, as there may be environmental regulations.
8. Disconnect power banks when not in use, especially if children, elderly, or disabled individuals are present. This helps prevent damage from battery leakage.
9. **Climate:** The gadget has been engineered for application in moderate climates and in domestic situations.
10. **Abnormal smell or sound:** If any foul smell or sound is detected from the equipment, unplug the cable from the Micro-USB Port and Power banks. Contact your authorized dealer or service center and do not reconnect the equipment.
11. **Damage requiring services:** The equipment should be serviced by qualified service personnel when:
    - The power-supply or the cable has been damaged, or
    - Objects have fallen, or liquid has spilled into the equipment, or
    - The equipment has been exposed to rain, or
    - The hardware does not appear to operate normally or exhibits a marked change in performance, or
    - The equipment has been dropped or the enclosure damaged.

12. The figures and illustrations in this User Manual may not exactly match the appearance of the product; they are only meant to be used as a guide. Product specs and design are subject to change at any time.

9. Troubleshooting

9.1. Troubleshooting

If the Magic Wand seems to have a problem, first review this list of possible problems and solutions. If none of these troubleshooting tips apply, please visit www.magic-wand.com and click on Support.

9.1.1. The Magic Wand can't turn on (i.e. Power LED or Orange Programmable LED are not working)

Common Issues

- Ensure you have **FULLY** inserted the USB cables.
- Check you have a **COMMON GROUND** where required.
- Where possible use USB 2.0 ports or a USB 2.0 **POWERED HUB** to rule out USB 3.0 issues.
- Try other computers where possible.
- Try other USB leads where possible.
- You may not have the correct driver installed. `CH340/341 12`, `CP2102 9`, or `FT232 VCP Drivers- FTDI 11`.
- There may be a problem with the board, check or remove your wiring first.
- Remove any items connected to pins 0 and 1.

Computer Related

- Close any other serial programs before opening the IDE. Ensure you turn off any additional security/antivirus just to test.
- There may be a problem with the PC, try **RESTARTING** it.
- You may have selected the wrong COM port.
- Avoid cloud/network-based installations where possible OR ensure your Network/Cloud software is **RUNNING**.

- Clear your browser's **CACHE**.
- Close the IDE before using any other serial programs. Preferably install IDE's as **ADMINISTRATOR** or your OS equivalent.

9.1.2. I am not getting the gesture which I waved

- **Review Technique**: Revisit the gesture instructions to confirm you are performing the movements correctly.
- **Clear Space**: Make sure you have ample space around you, free from obstructions that could interfere with gesture detection.
- **Re-hold**: Hold the wand in a manner properly. If the gesture still cannot be detected, please contact the authorized dealers or service centers for further assistance.

9.1.3. Is it wireless, Can I connect it with Bluetooth?

It can be connected with Bluetooth, but as there is no inbuilt battery, it should be powered by connecting it with the Micro-B USB cable to the computer.

9.1.4. How to turn on/turn off the magic wand?

The board is powered via USB connector. Once turned on, the Power green LED lights up. Removing the cable from the USB port can turn off the Magic Wand.

10. Add Ons

10.1. Project Folder Structure

```
Project Root
|- src/
|- main.cpp
|- imu_processing.cpp
|- ml_model.cpp
|- gesture_recognition.cpp
|- include/
|- imu_processing.h
|- ml_model.h
|- gesture_recognition.h
|- models/
|- magic_gestures.tflite
|- scripts/
|- data_collection.py
|- model_training.py
```

Listing 10.1: Project Directory Structure

Reference to the structure: See Listing 10.1.

10.2. File Descriptions

`src/`

Contains the main application code.

- `main.cpp`: Serves as the entry point of the program, initializing and orchestrating the overall flow.

- `imu_processing.cpp`: Handles data collection from IMU (Inertial Measurement Unit) sensors, including filtering or preprocessing data.

- `ml_model.cpp`: Loads and interfaces with the pre-trained TensorFlow Lite model, running inference on the processed IMU data.

- `gesture_recognition.cpp`: Contains the logic for recognizing specific gestures based on model predictions.

`include/`

Header files for the source files, defining necessary interfaces and dependencies.

- `imu_processing.h`: Declares the functions and classes for IMU processing.
- `ml_model.h`: Declares the model loading and inference functions.
- `gesture_recognition.h`: Declares the gesture recognition logic functions.

`models/`

Stores machine learning models.

- `magic_gestures.tflite`: The pre-trained TensorFlow Lite model used for gesture classification.

`scripts/`

Python scripts for data collection and model training.

- `data_collection.py`: Used to collect motion data, typically in the form of raw IMU sensor readings.
- `model_training.py`: Handles the training of the machine learning model, potentially generating `magic_gestures.tflite`.