Requirment specification

Project Title	Smart-Wand
Project Leader	Toferer Thomas
Created on	12.12
Last changed on	23.12
state	Finished
Current Version	2

Change History

Nr.	Date	Version	Changed Chapter	Type of Change	Author	State
1	12.12.2025	1	1,2,3,4,5,6	Creation	Durkowitsch, Codoban, Toferer	Finished
2	13.12.2025	1.5	2,7	Update	Durkowitsch	Finished
3	23.12.2025	2	1,2,3,4,5,6,7	Update	Toferer	Finished
4	-	-	-	-		-
5	-	-	-	-		-

Table of Contents

1 Introduction

This Requirement Specification document defines the complete set of functional and non-functional requirements for the product to be developed. It serves as the foundation for the project tender and contract formulation, thereby establishing the mandatory basis for proposal preparation.

Should a contract be concluded between the contractor and the client, this document becomes legally binding. Unless explicitly stated otherwise herein, all prior agreements and understandings between the client and the contractor are superseded by this Requirement Specification.

By defining these requirements, the framework for the development is established, which the contractor shall implement in detail as specified in this document.

2 General

2.1 Aim and purpose of the document

This requirement specification defines the functional and non-functional requirements for the "Smart Wand" project. It serves as the foundation for development, testing, and final acceptance, and is legally binding upon agreement between the client and the project team.

2.2 Initial situation

This project is executed under the educational framework of **HTL Wels**, a technical college that provides the necessary institutional backing, supervision, and resources. It is conducted as a hands-on proof-of-concept by a student team, focusing on the practical implementation of a gesture-based smart home controller prototype.

The key stakeholders and partners involved are defined as follows:

Role	Entity / Name	Description & Responsibility
Client	**Susanne Loidl**	The interested party in the project's concept and outcomes.
Supplier / Contractor	**HTL Wels**	The educational institution under which the project is executed, providing the framework and resources.
Project Lead	**Thomas Toferer**	Holds overall responsibility for project management, coordination, and delivery.
Project Team	**Laurenz Durkowitsch, Simon Codoban**	Responsible for the design, development, integration, and testing of the "Smart Wand" prototype.
Technology Partner (Hardware)	**Espressif Systems**	Producer of the core microcontroller unit (ESP32) and associated development frameworks.
Technology Partner (Platform)	**Home Assistant (open-source project)**	The target home automation platform for integration.

Technology Partner (Actuators)	**Shelly (Allterco Robotics)**	Producer of the target smart home devices (e.g., relays, plugs) to be controlled.
End-User	*Advanced Smart Home Enthusiasts / Tech Hobbyists*	The ultimate beneficiary. Their interaction patterns are a key measure of success.

2.3 Abbreviations

The present project is an independent project. It involves a documentation ("Smart Wand") for a feasibility study of gesture-based smart home control.

2.4 Teams and Interfaces

Rolle(n)	Name	Telefon	E-Mail	Team
Project-Owner	HTL-Wels		Office@htl-wels.at	
Project-Leader	Thomas Toferer		Thomas,toferer@htl-wels.at	Smart-Wand
Project-Teammember	Laurenz Durkowitsch		Laurenz.durkowitsch@htl-wels.at	Smart-Wand
Project-Teammember	Simon Codoban		simon.codoban@htl-wels.at	Smart-Wand
Client	Susanne Loidl		Susanne.loid@bildung.gv.at	HTL Wels
Hardware Advisor	Andreas Wögerbauer		Andreas.woegerbauer@bildung.gv.at	HTL Wels

3 Concept

3.1 Concept

The Smart-Home Wand is a handheld, battery-powered device designed to control smart-home devices through intuitive “spell-casting” gestures. Equipped with Wi-Fi connectivity, the wand communicates directly with Home Assistant to toggle for example lights on and off. The device emphasizes reliable motion and spell detection while offering an appealing, design suitable for demonstrations and playful interactions.

3.2 Goal(s) of the provider specification

- Define a feasible, reliable, and easy-to-build smart-home wand concept.
- Ensure stable gesture/spell detection for consistent performance.
- Integrate Wi-Fi communication compatible with Home Assistant.
- Outline charging and power requirements to keep the device practical and maintainable.

- Provide a clear standard for creating a demonstration-ready gimmick suitable for events, makerspaces, or tech showcases.

3.3 Specifications Goals and benefits for the user

- Create a **simple, user-friendly wand** that is easy to assemble, configure, and operate.
- Provide a **stable wireless connection** to smart-home systems.
- Ensure **gesture/spell detection accuracy** even with novice users.
- Make the wand visually appealing with optional customization.
- Design the wand to be **portable, lightweight, and reliable** for repeated handling.

User Benefits

- **Magical interaction experience:** Users can toggle lights with a “spell” instead of a switch.
- **Reliable performance:** Responsive spell detection enhances the user experience.
- **Rechargeable:** No need to constantly replace batteries; USB-C or magnetic charging. (May Goal)
- **Demonstration-ready:** An engaging tool for presentations, workshops, and smart-home demos.
- **Customizable housing:** Users can personalize the look to fit fantasy or sci-fi themes.

3.4 Target group(s)

- **Smart-home enthusiasts** who enjoy incorporating playful and experimental automation devices.
- **Tech hobbyists and DIY makers** looking for fun projects involving motion sensors, Wi-Fi microcontrollers, and 3D-printed parts.
- **Educators and STEM presenters** who want a visually appealing demonstration tool to spark interest in IoT concepts.
- **Event hosts, exhibitors, or demonstrators** needing an engaging gimmick for smart-home showcases.
- **Cosplayers and fantasy fans** seeking a functional, interactive magic-themed prop.
- **Beginner electronics learners** who want an approachable project that delivers satisfying results.

4 Functional Requirements

4.1 Requirement 1:

Spell Detection

- The wand must detect at least 5 different spell movements using the MPU6050 gyroscope sensor
- Detection accuracy must be at least 60% (3 out of 5 attempts successful)
- Response time for spell detection must be under 1000ms

4.2 Requirement 2:

Communication with Home Assistant

- The ESP32 must establish a stable connection to the Home Assistant server
- Commands must be sent via REST API to control smart home devices
- The system must support controlling lights, switches, and other Home Assistant entities

4.3 Requirement 3:

Movement Recognition

- The system must distinguish between different wand movements (up, down, circular, zigzag, etc.)
- Each spell pattern must trigger a specific smart home action
- The wand must provide feedback (LED) when a spell is recognized

4.4 Requirement 4:

Configuration

- Users must be able to assign different spells to different smart home actions
- The system must store spell configurations persistently

5 Non-Functional Requirements

5.1 General Requirements

Performance:

- Maximum latency of 800ms between spell execution and smart home device response
- Battery life of at least 8 hours
- The system must support simultaneous use by one user

Usability:

- Intuitive spell gestures that are easy to learn and remember
- Clear feedback for successful/failed spell detection
- Simple setup process

Reliability:

- 85% uptime during operation
- Stable WiFi connection with automatic reconnection after connection loss
- Error handling for failed commands

Maintainability:

- Modular code structure for easy updates
- Documented API endpoints
- Version control using Git

5.2 Legal Requirements

Data Protection:

- No personal data is collected or stored beyond local network configuration
- GDPR compliance for any user data handling
- Local processing of sensor data (no cloud dependency)

Safety:

- Low voltage operation (3.3V/5V)
- Safe battery charging with overcharge protection

Intellectual Property:

- Open-source libraries used under their respective licenses (Arduino Framework, ESP32 libraries)
- Home Assistant API used according to their terms of service

5.3 Technical Requirements

Hardware:

- ESP32 microcontroller (WiFi enabled)
- MPU6050 6-axis gyroscope/accelerometer sensor
- Power supply: LiPo battery or USB power
- Optional: LED indicator, buzzer for feedback

Software:

- Arduino Framework for ESP32 programming

- Home Assistant REST API integration
- WiFi connectivity using ESP32 WiFi library
- Sensor data processing using MPU6050 library

Network:

- Local Wifi LAN (Using a old router)
- Home Assistant server accessible on local network
- Static IP support for reliable connection

6 Framework conditions

6.1 Schedule

Project Start: 02.12.2025
Project End: End of May 2026

Milestones:

1. **15.01.2026** - Setup of Home Assistant and additional hardware complete
2. **01.02.2026** - Spell Detection working reliably
3. **20.02.2026** - Communication from wand to Home Assistant established
4. **01.03.2026** - Preparation for first demo complete
5. **End of May** - Final presentation and project completion

6.2 Technical Requirements

Hardware Requirements

Minimum	Recommended
ESP32 Development Board (1x) - 7€	ESP32 Development Board (1x) - 7€
MPU6050 Sensor Module (1x) - 7,55€	MPU6050 Sensor Module (1x) - 7,55€
Wand housing/3D printed case - ~5€	Wand housing/3D printed case - ~5€
Raspberry Pi 5 for Home Assistant (1x) - 75€	Raspberry Pi 5 for Home Assistant (1x) - 75€
LiPo Battery 3.7V (1x) - ~10€	LiPo Battery 3.7V (1x) - ~10€
	LED, buzzer, cables - ~5€
SD Card 32 GB - ~10€	SD Card 32GB - ~10€
Total minimum: ~105€	Total recommended: ~110€

Software Recommended Requirements

- Arduino IDE or PlatformIO
- Home Assistant OS/Supervised installation
- Git for version control
- Python 3.x (for potential additional scripting)

6.3 Problem Analysis

Expected Problems and Solutions:

Problem 1: Spell Detection Accuracy

- Risk: Low accuracy in distinguishing different movements
- Solution: Implement machine learning algorithm or threshold-based detection with extensive calibration and testing phase

Problem 2: WiFi Connection Stability

- Risk: Connection drops causing commands to fail
- Solution: Implement automatic reconnection logic, command queue system, and local feedback for connection status

Problem 3: Power Consumption

- Risk: Battery drains too quickly during use
- Solution: Implement sleep modes, optimize sensor polling rate, use efficient code practices

Problem 4: Home Assistant API Changes

- Risk: API updates breaking compatibility
- Solution: Use stable API version, implement error handling, document API version used

Problem 5: Sensor Calibration

- Risk: Different users need different sensitivity settings
- Solution: Implement calibration routine at startup, allow user-adjustable sensitivity settings

Problem 6: Integration with Different Smart Home Devices

- Risk: Not all devices respond the same way
- Solution: Test with multiple device types, implement device-specific handlers if needed

7 Delivery and Acceptance conditions

7.1 Delivery Scope

The supplier (project team) will deliver:

- A demonstration of the project's intern prototype
- Complete software (source code) and configuration files
- STL Files for a wand design
- Documentation for building (user manual, technical documentation, setup guide)

- An image file of the raspberry pi’s filesystem

7.2 Delivery Date

The final delivery shall be completed by the end of May 2026.

7.3 Acceptance Criteria

The project will be considered accepted when:

- All functional requirements (Section 4) are fulfilled
- All non-functional requirements (Section 5) are met
- A successful live demonstration is performed for the client
- All documentation is submitted and reviewed

8 Appendix

Project Handbook