TED UNIVERSITY

CMPE 491 – Senior Project I

Fall 2025

AKKE – Analysis Report

(SMART COMMAND AND CONTROL GLOVE)

Webpage: https://ilter-akke.github.io/website/

Name Surname Department

Abdullah ESİN (CMPE)

Berk ÇAKMAK (CMPE)

Ömer Efe DİKİCİ (CMPE)

Şevval KURTULMUŞ (EEE)

Supervisors: Ali BERKOL and Hüseyin Uğur YILDIZ

Jury Members: Hakkı Gökhan İLK and Mehmet Evren COŞKUN

Table of Contents

1. Introduction

In modern defense and security operations, reliable and disciplined communication is crucial for mission – success and personnel safety. However, traditional communication methods such as voice commands, radio communication and visual hand signals become insufficient or risky in environments where silence, speed and confidentiality are critical. Noise, limited visibility, electronic warfare threats and line-of-sight constraints may lead to misunderstandings, delays or exposure of the team.

The AKKE project addresses these limitations by introducing a smart command and control glove that enables users to transmit predefined commands silently through hand gestures. The system integrates sensor-based data collection, gesture recognition and wireless audio communication into a wearable platform. Flex sensors and IMU modules capture the user's hand posture and motion, while an ESP32-S3 microcontroller processes this data and maps recognized gestures to corresponding audio commands. These commands are then broadcast over a radio channel to multiple receiver units, enabling silent, fast and coordinated communication without the need for speech or line-of-sight contact.

This document presents the analysis of the AKKE system. It refines and structures the previously defined requirements, models the problem domain using object-oriented analysis techniques, and clarifies how users and the system will interact. The goal of this analysis report is to provide a correct, consistent and verifiable model of the system that will serve as a contract between the development team and the stakeholders, and as a solid foundation for the upcoming high-level design activities.

1.1 Document Organization

The remainder of this report is organized as follows:

- **Section 2** describes the current communication practices and limitations in the targeted application domain.
- **Section 3** introduces the proposed AKKE system, summarizes its functional and non-functional requirements, and presents the main system models, including scenarios, use case model, object/class model, dynamic models and user interface concepts.
- **Section 4** provides a glossary of important terms used in the project.
- **Section 5** lists the references that guided the requirements and analysis activities.

2. Current system

2.1 Communication Practices in the Field

In many tactical and security-oriented missions, teams rely on a combination of voice communication, radio systems and visual hand signals. Squad leaders often issue spoken commands over handheld radios or directly shout verbal orders to nearby team members. In quieter conditions or when close contact is possible, conventional hand signals are used to indicate actions such as moving, stopping, regrouping or attacking.

These methods, although well-established, assume that either speech is safe and feasible or that team members have a clear line of sight to the leader. Moreover, they require continuous attention and may be affected by environmental conditions such as terrain, obstacles, poor lighting and weather.

2.2 Limitations of the Current System

The existing communication approaches face several limitations in the environments targeted by AKKE:

- **Silence and secrecy requirements:** In covert or high-risk missions, speaking over the radio or shouting commands may reveal the team's position or intent. Even encrypted radios do not fully mitigate the risks of acoustic detection.
- **Line-of-sight dependency:** Hand signals require that team members visually spot the leader. In smoke, darkness, complex urban environments or forested areas, this is often not possible.

As a result, the current system may fail to provide the required level of reliability, speed and discipline in command-control, especially under harsh operational conditions.

2.3 Stakeholders and Users

The main stakeholders of the AKKE system are:

- **Primary users – Field personnel:** Squad leaders and team members who will use the glove and receiver units during missions. They require a system that is fast, reliable, ergonomic and easy to learn.
- **Development and maintenance teams:** Engineers and technicians responsible for implementing, testing, deploying and maintaining the hardware and software components. They require clear requirements, maintainable design and suitable documentation.

Understanding the needs and constraints of these stakeholders is essential for shaping the proposed solution and for validating the analysis model.

3. Proposed system

3.1 Overview

The proposed system, AKKE, is a smart wearable solution designed to convert predefined hand gestures into audio-based commands that can be transmitted wirelessly to multiple receivers. The core of the system is a glove equipped with flex sensors, which measure finger bending, and IMU sensors, which capture 3-axis acceleration and angular velocity. These sensors provide a continuous stream of raw data that characterizes the user's hand posture and motion.

An ESP32-S3 microcontroller embedded in the glove processes the sensor data in real time. The system applies preprocessing and a gesture recognition algorithm to detect which predefined gesture is being performed. Once a gesture is recognized, it is mapped to a corresponding command from a predefined command set (e.g., "Advance", "Hold Position", "Retreat").

The recognized command is then converted into an audio representation. Pre-recorded audio files stored on a microSD card are played through a DFPlayer Mini audio module, and the resulting audio signal is transmitted by an RF-based radio module. Multiple receiver units (e.g., radios worn by team members) can listen to the broadcast and instantly hear the command, enabling silent but highly synchronized communication without voice or visual contact.

3.2 Functional Requirements

Based on the specifications report, the functional requirements of AKKE can be grouped into four main categories: sensor and data collection, gesture recognition, audio communication and command processing.

Sensor and Data Collection

The glove must continuously collect data from flex sensors and IMU sensors while the system is active. Each flex sensor should capture finger bending from approximately 0° to 180°, and the IMU must provide tri-axial acceleration and gyroscope readings. The sampling rate must be sufficient to detect rapid hand movements and transitions between gestures in real time.

Gesture Recognition

The system must recognize at least five distinct hand gestures that correspond to tactical commands. The gesture recognition process should combine multiple sensor channels to distinguish gestures that differ in both finger configuration and overall hand orientation. The target recognition accuracy is at least 75–80% under realistic operating conditions, which implies that the algorithms must handle sensor noise, small user-specific variations and environmental disturbances.

Audio Communication

For each recognized gesture, the system must trigger the playback of a specific audio file that describes the corresponding command. Audio files are stored on a microSD card and played through the DFPlayer Mini module. The audio output must then be broadcast via an RF transmitter so that multiple receivers can simultaneously obtain the same command message.

Command Processing

The system must implement the full pipeline from sensor data to high-level commands. This includes converting sensor readings to feature representations, feeding them to the gesture recognition module, mapping recognized gestures to command identifiers, retrieving the correct audio file and ensuring that the broadcast is completed. The system must work without requiring voice input or line-of-sight communication, thereby enabling silent operation.

3.3 Nonfunctional Requirements

The nonfunctional requirements specify performance, hardware, reliability, usability and maintainability expectations for AKKE.

Performance

The end-to-end response time from gesture completion to audio command transmission must be less than one second. The microcontroller must be capable of processing data streams from multiple sensors simultaneously without noticeable delays.

Hardware and Sensor Specifications

The system will be built around an ESP32-S3 microcontroller for sensor processing and classification, a DFPlayer Mini for audio playback, an RF-based transmitter for wireless audio communication, and a microSD card for storage. Flex sensors must be flexible and comfortable, while IMU modules must offer reliable 3-axis accelerometer and gyroscope data. Power will be provided by a rechargeable battery, and appropriate module will be used for RF transmission.

Reliability and Environmental Conditions

The glove must operate in harsh military environments, including dust, moisture and varying temperatures. The system should maintain a minimum communication range (e.g., around 10 meters) and must incorporate methods to mitigate sensor noise.

Usability and Ergonomics

The glove must be easy to put on and take off, and calibration and setup procedures must be simple and repeatable. Users should not require deep technical knowledge to operate the system. Visual or haptic feedback should clearly indicate the system state, such as successful gesture recognition or error conditions, to build user confidence.

Maintainability and Sustainability

The system should support rechargeable or replaceable batteries and allow for hardware access in case of maintenance or component replacement. Software should be modular to facilitate updates to gesture sets, classification models and communication parameters. Low power consumption is essential for long mission durations and sustainability in field operations.

3.4 Pseudo requirements

In addition to the formal functional and nonfunctional requirements, several design choices and constraints influence the implementation of AKKE. These pseudo requirements are not strictly imposed by the problem domain but are selected to align with project scope, available resources and modern engineering practices:

- **Programming languages and platforms:**
    - Embedded firmware on the ESP32-S3 will be developed using C/C++, enabling direct control over sensors, peripherals and timing.
    - Gesture recognition models may be initially developed and evaluated in Python using machine learning libraries and then translated or compressed into a form suitable for embedded deployment (e.g., TinyML).

- **Hardware selection:**
    - ESP32-S3 is preferred due to its sufficient processing power, available memory, integrated communication features and potential AI acceleration capabilities.
    - Off-the-shelf DFPlayer Mini and RF modules are chosen for rapid prototyping and to keep costs within the project budget.

- **Project constraints:**
    - The prototype is designed under a limited budget and within a single academic year. Therefore, component choices, complexity of the classification model and the number of supported gestures is selected to be feasible for implementation and evaluation during this period.

These pseudo requirements document the rationale behind implementation choices and will guide the subsequent design and prototyping activities.

3.5 System models

3.5.1 Scenarios

Scenario 1 – Sending a Command in a Noisy Environment

A squad leader is equipped with the AKKE glove during an operation in a noisy urban area. The leader performs a predefined “Advance” gesture. The glove collects sensor data, recognizes the gesture and immediately plays the corresponding “Advance” audio command through the audio module. The command is broadcast via the RF transmitter and received by the radios of all team members, who then advance according to the order. No verbal communication is required, reducing the risk of detection or misunderstanding.

Scenario 2 – Calibrating the Glove Before a Mission

Before a mission, a user powers on the glove and enters calibration mode. The system guides the user to perform a set of reference poses to adapt to their hand size and natural movement pattern. Sensor baselines are updated, and the recognition thresholds are tuned accordingly. Once calibration completes successfully, the glove switches to normal operation.

Scenario 3 – Handling Misrecognized Gestures

During a mission, the user performs a gesture, but sensor noise or an incomplete movement leads to low confidence in recognition. Instead of issuing a potentially wrong command, the system detects the low confidence and activates a visual indicator (e.g., a red LED). The user understands that the gesture was not correctly recognized and repeats it more clearly. Only when the recognition confidence is above a threshold does the system transmit the corresponding command.

3.5.2 Use case model

The main actors and use cases of the system are:

- **Actors:**
  - *Glove User (Squad Leader)* – primary actor who performs gestures and issues commands.
  - *Receiver Unit* – external hardware that receives and plays the audio commands (passive actor).
  - *Maintenance Operator* – optional actor responsible for configuring and updating the system.
- **Use Cases:**
  - **Perform Gesture Command:** User performs a predefined gesture, the system recognizes it, maps it to a command and broadcasts the corresponding audio message.
  - **Calibrate Glove:** User or maintenance operator initiates calibration, and the system collects reference data to normalize sensor readings.
  - **Update Gesture Set:** Maintenance operator updates the gesture definitions or associated commands and redeploys the configuration.
  - **Monitor System Status:** User checks indicators (LEDs, battery level, error notifications) to ensure that the glove is operational.

In the corresponding UML Use Case Diagram, the Glove User is linked to "Perform Gesture Command", "Calibrate Glove" and "Monitor System Status", while the Maintenance Operator is additionally linked to "Update Gesture Set".

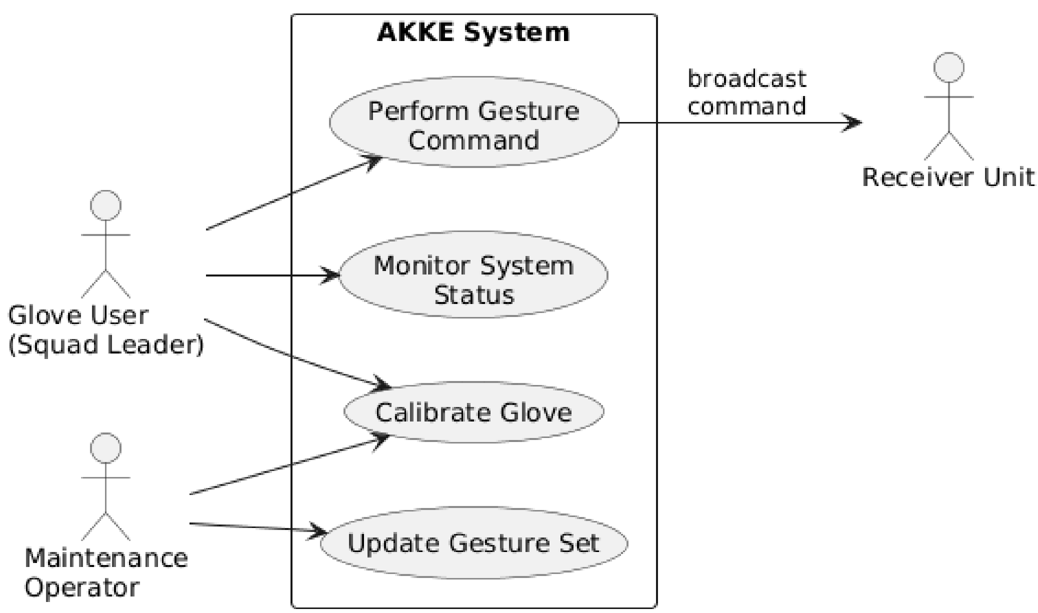

Figure 1: Use case diagram illustrating the interactions between the user, the AKKE system, and external receivers.

3.5.3 Object and class model

The object and class model describes the main concepts in the application domain and their relationships. Key classes include:

- **GloveDevice:** Represents the overall glove system, coordinating sensors, recognition and communication.
- **Sensor (abstract):** Generic interface for all sensors.
    - **FlexSensor:** Specialization that measures finger bending.
    - **IMUSensor:** Specialization that provides accelerometer and gyroscope data.
- **SensorData:** Encapsulates raw and preprocessed readings from sensors at a given time.
- **Gesture:** Represents a predefined pattern of sensor data corresponding to a command.
- **GestureClassifier:** Implements the recognition algorithm, mapping SensorData sequences to Gesture instances.
- **Command:** Represents a high-level command (e.g., "Advance"), associated with a Gesture and an audio file identifier.
- **AudioPlayer:** Controls the DFPlayer Mini to play the appropriate audio file.
- **RFTransmitter:** Handles transmission of the audio signal to receiver units.
- **ReceiverUnit:** Represents external receivers that play the transmitted command to team members.
- **User:** Represents the person wearing and interacting with the glove (for logging and calibration).

Associations among these classes reflect the data flow: GloveDevice owns multiple Sensor objects, collects SensorData, requests recognition from GestureClassifier, retrieves a Command, and uses AudioPlayer and RFTransmitter to deliver the command. This model captures the core structure that will later be refined during high-level design.

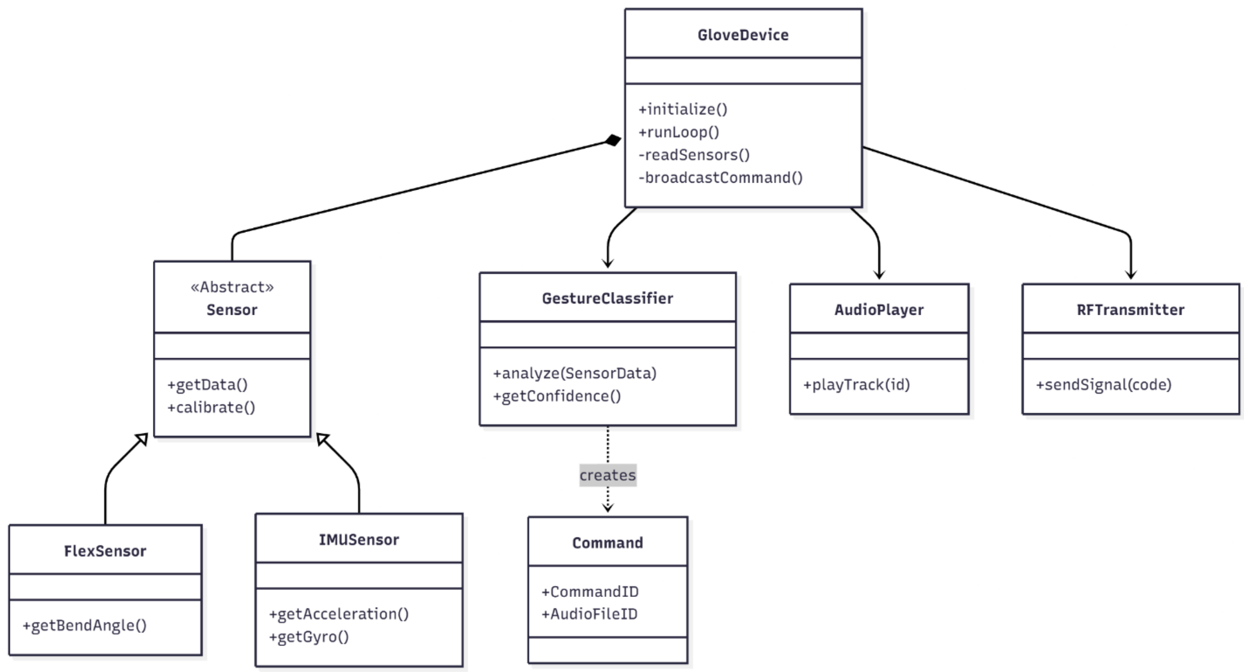

Figure 2: Object and class diagram depicting the software architecture and main components of the AKKE system.

3.5.4 Dynamic models

Dynamic models describe how system objects interact over time.

Sequence of "Perform Gesture Command"

A typical sequence for issuing a command is as follows:

1. The User performs a gesture.
2. GloveDevice requests new readings from FlexSensor and IMUSensor objects.
3. The sensors return raw data, which GloveDevice wraps into a SensorData structure.
4. GloveDevice sends SensorData to GestureClassifier.
5. GestureClassifier analyzes the data and returns a recognized Gesture with a confidence value.
6. GloveDevice maps this Gesture to a Command.
7. GloveDevice instructs AudioPlayer to play the audio file associated with the Command.
8. AudioPlayer outputs the audio signal, which RFTransmitter broadcasts to all ReceiverUnit instances.

If the recognition confidence is below a threshold, the sequence is altered so that instead of broadcasting, the system activates an error indicator and waits for a new gesture.

Glove State Machine

The glove can be described by a state machine with states such as:

- *PowerOff* – the device is not active.
- *Idle* – the device is powered on, waiting for gestures.
- *Calibrating* – the device is collecting baseline data.
- *CollectingData* – the device is reading sensors for a potential gesture.
- *RecognizingGesture* – the classifier is processing the data.
- *TransmittingCommand* – the audio command is being played and broadcast.
- *ErrorState* – a failure or low confidence is detected; the user is notified.

Transitions occur based on events such as power on/off, calibration request, gesture detection or error conditions.

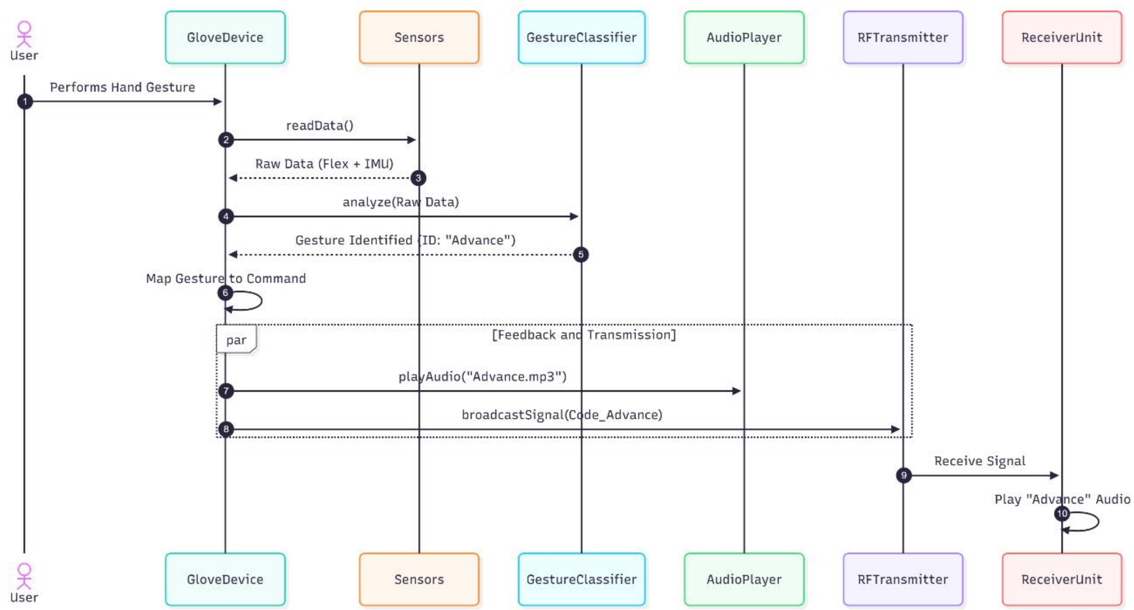

Figure 3: Sequence diagram for the "Perform Gesture Command" scenario, showing the data flow from sensor reading to audio transmission.

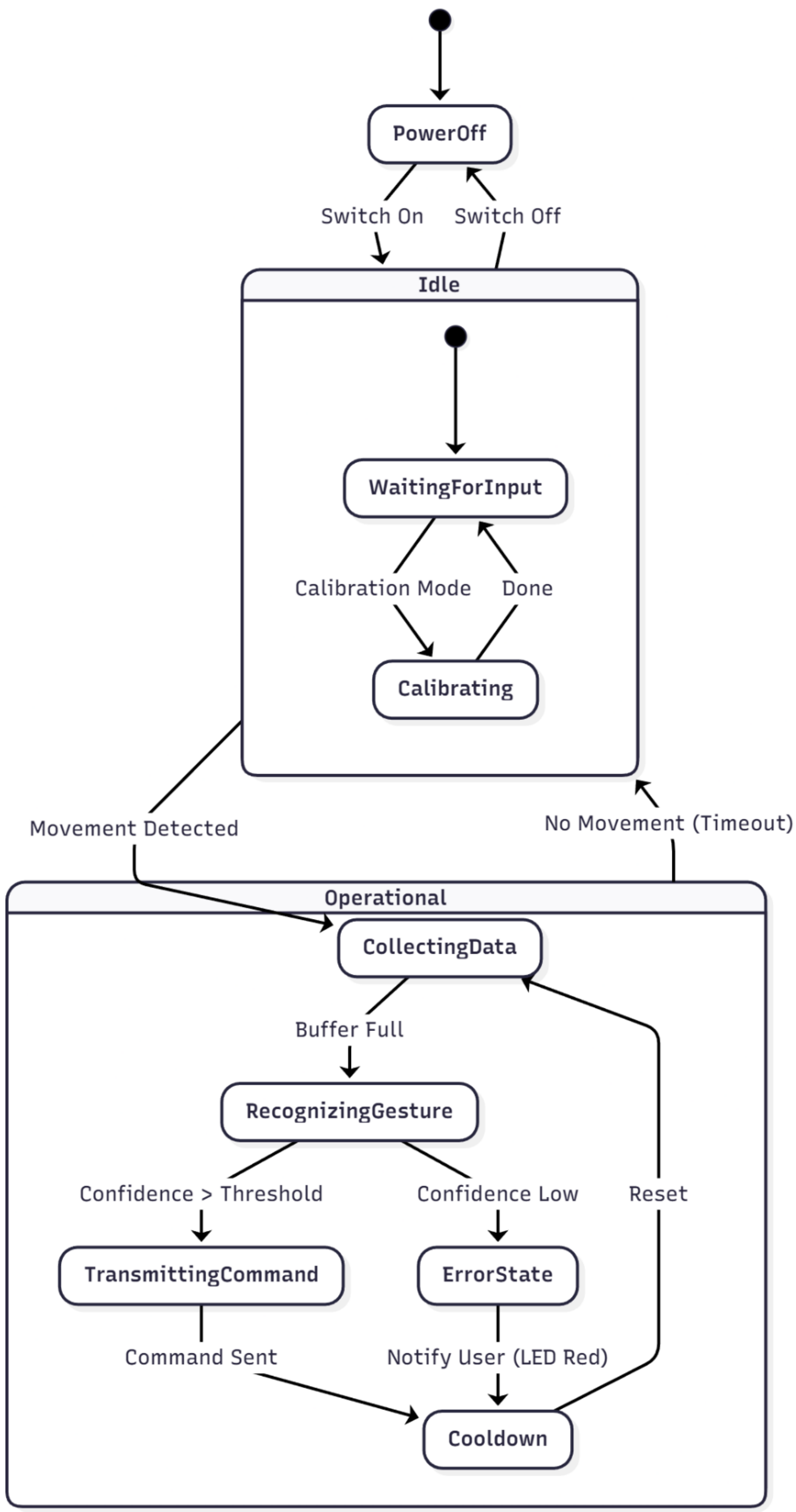

Figure 4: State machine diagram illustrating the operational states and transitions of the AKKE glove.

3.5.5 User interface – navigational paths and screen mock-ups

Since the AKKE system is designed for covert military operations, it intentionally excludes traditional graphical user interfaces (screens) to minimize light emission and cognitive load. instead, the system employs a **"Headless" Physical User Interface**, relying on tactile inputs and minimalist visual/haptic feedback.

1. Physical Layout: The user interface consists of the wearable glove and a wrist-mounted control unit. The control unit houses the following interface elements:

- **Power Switch:** A mechanical toggle switch to physically disconnect the battery.
- **Status LED (RGB):** A single multi-color LED used to communicate the system's operational state to the user without creating significant light pollution.
- **Calibration Button:** A physical button to trigger the sensor calibration routine.

2. Feedback Mechanisms: The system communicates with the user through color-coded light signals. The interactions are designed to be understood briefly:

System State	LED Color / Behavior	Meaning
Booting	Blinking White	System is initializing sensors.
Idle / Ready	Solid Blue (Dimmed)	Ready to detect gestures.
Processing	Solid Yellow	Gesture detected, analyzing data.
Success	Blink Green (Once)	Command recognized and transmitted.
Error / Low Confidence	Blink Red (Twice)	Gesture not recognized, repeat action.
Low Battery	Blinking Red (Continuous)	Battery needs charging.

3. Navigational Concept: Unlike software navigation, the "navigation" in AKKE is state-based. The user transitions between states (Idle => Active) solely through hand movements, removing the need for menu navigation.

4. Glossary

- **AKKE:** Smart command and control glove project enabling silent communication via hand gestures.
- **Flex Sensor:** A bend-sensitive resistor attached to fingers that measures the degree of finger bending.
- **IMU (Inertial Measurement Unit):** A sensor module that combines accelerometers and gyroscopes to measure linear acceleration and angular velocity.
- **Gesture:** A specific pattern of hand posture and motion captured by sensors and mapped to a predefined command.
- **Command:** A high-level instruction (e.g., “Advance”, “Hold Position”) associated with a gesture and an audio message.
- **ESP32-S3:** A microcontroller platform used for sensor data processing, gesture recognition and overall system control.
- **DFPlayer Mini:** A small audio module that plays pre-recorded audio files from a microSD card.
- **RF (Radio Frequency) Transmitter:** A module that wirelessly broadcasts the audio commands to multiple receivers.
- **Receiver Unit:** A device (e.g., radio) that receives and plays the audio command transmitted by the glove.
- **Calibration:** The process of adapting the system to a specific user’s hand characteristics and sensor baselines.
- **Recognition Accuracy:** The proportion of gestures correctly classified by the system.

5. References

Belcamino, V., Carfì, A., & Mastrogiovanni, F. (2024). A systematic review on custom data gloves. IEEE Transactions on Human-Machine Systems, 1–16. https://doi.org/10.1109/THMS.2024.3394674

Bruegge, B., & Dutoit, A. H. (2004). Object-oriented software engineering: Using UML, patterns, and Java (2nd ed.). Prentice Hall.

DFRobot. (n.d.). DFPlayer Mini MP3 player for Arduino (SKU: DFR0299). DFRobot Wiki.

Espressif Systems. (n.d.). ESP32-S3 series datasheet: 2.4 GHz Wi-Fi + Bluetooth LE SoC (Version 2.0). Espressif Systems Documentation.

Filipowska, A., Filipowski, W., Raif, P., Pieniążek, M., Bodak, J., Ferst, P., Pilarski, K., Sieciński, S., Doniec, R. J., & Mieszczanin, J. (2024). Machine learning-based gesture recognition glove: Design and implementation. Sensors, 24(18), 6157. https://doi.org/10.3390/s24186157

Fowler, M. (2003). UML distilled: A brief guide to the standard object modeling language (3rd ed.). Addison-Wesley.

Leite, C., Byvshev, P., Mauranen, H., & Xiao, Y. (2024). Simulation-driven design of smart gloves for gesture recognition. Scientific Reports, 14, 14873. https://doi.org/10.1038/s41598-024-65069-2

Ong, J. H. (2017). Data-glove-based hand gesture recognition system using flex sensors and an IMU sensor (Project report). Universiti Sains Malaysia, School of Electrical & Electronic Engineering.