Motion Play Data Collection System - Requirements Document

Project: Motion Play Cloud Platform
Version: 1.0
Date: November 7, 2025
Author: Marc
Status: Draft

1. Executive Summary

This document defines requirements for the Motion Play Data Collection System, a cloud-based platform enabling sensor data capture, labeling, and analysis for the Motion Play basketball hoop detection system. The platform supports development and tuning of detection algorithms through structured data collection and provides remote control capabilities for the ESP32-S3 based hardware.

2. Project Objectives

2.1 Primary Goal

Develop a cloud infrastructure and web interface that enables:

- Remote control of Motion Play device operation modes
- Collection and storage of labeled sensor data for algorithm development
- Analysis and visualization of sensor readings
- Foundation for future AI/ML-based detection systems

2.2 Success Criteria

- ESP32-S3 device successfully communicates bidirectionally with cloud backend
- Web interface can trigger data collection sessions and receive results
- Sensor data is stored with associated labels for future analysis
- System supports transition between debug and play modes
- Platform architecture supports future multi-device expansion

3. Scope

3.1 In Scope - Phase 1

- Single device communication infrastructure
- Data collection session management
- Web-based control interface
- Sensor data storage and retrieval
- Session labeling and metadata management
- Basic data visualization and export capabilities
- Debug mode and Play mode operation

3.2 Out of Scope - Phase 1

- Multi-device coordination and clustering
- Automated ML model training
- On-device AI inference
- Mobile application interface
- User authentication and multi-user support
- Real-time data streaming visualization
- Production game modes and scoring logic

3.3 Future Phases

- Phase 2: Multi-device coordination and tournament infrastructure
- Phase 3: ML model training and deployment pipeline
- Phase 4: Consumer-facing application and production features

4. Functional Requirements

4.1 Device Communication

REQ-COM-001: Device shall establish WiFi connection and maintain MQTT connection to AWS IoT Core
REQ-COM-002: Device shall display connection status on T-Display screen
REQ-COM-003: Device shall receive command messages from cloud backend
REQ-COM-004: Device shall transmit sensor data batches to cloud backend
REQ-COM-005: Device shall buffer sensor data locally during collection sessions

4.2 Operating Modes

REQ-MODE-001: Device shall support Debug Mode for data collection
REQ-MODE-002: Device shall support Play Mode for game operation
REQ-MODE-003: Device shall support Idle/Standby state when not actively collecting or playing
REQ-MODE-004: Device shall accept mode change commands from web interface
REQ-MODE-005: Device shall display current mode on T-Display screen

4.3 Data Collection (Debug Mode)

REQ-DATA-001: System shall support manual start/stop of data collection sessions via web interface
REQ-DATA-002: Device shall collect readings from all 6 VCNL4040 sensors (3 positions × 2 sensors)
REQ-DATA-003: Device shall sample sensors at up to 1000 Hz during collection
REQ-DATA-004: Device shall buffer complete session data locally before transmission
REQ-DATA-005: Collection sessions shall support duration from 5-30 seconds
REQ-DATA-006: Device shall transmit buffered data to backend upon collection stop command
REQ-DATA-007: Each data point shall include timestamp, sensor ID, proximity values, and ambient values

4.4 Session Management

REQ-SESS-001: System shall create unique session identifier for each collection
REQ-SESS-002: System shall store session metadata (timestamp, duration, mode, device ID)
REQ-SESS-003: User shall be able to add text labels to completed sessions
REQ-SESS-004: User shall be able to create, edit, and delete label categories
REQ-SESS-005: System shall support multiple labels per session
REQ-SESS-006: User shall be able to delete sessions
REQ-SESS-007: System shall persist sessions until manually deleted

4.5 Web Interface

REQ-UI-001: Interface shall display device connection status
REQ-UI-002: Interface shall provide controls to start/stop data collection
REQ-UI-003: Interface shall allow mode switching (Debug/Play)
REQ-UI-004: Interface shall display list of collected sessions
REQ-UI-005: Interface shall allow user to view session details and sensor data
REQ-UI-006: Interface shall provide session labeling capabilities
REQ-UI-007: Interface shall support data export (JSON/CSV format)
REQ-UI-008: Interface shall work on desktop browsers (Chrome, Firefox, Safari)

4.6 Device Control Commands

REQ-CMD-001: System shall support "start_collection" command
REQ-CMD-002: System shall support "stop_collection" command
REQ-CMD-003: System shall support "set_mode" command (debug/play)
REQ-CMD-004: System shall support "restart" command
REQ-CMD-005: System shall support "get_status" command
REQ-CMD-006: Device shall acknowledge received commands

5. Non-Functional Requirements

5.1 Performance

REQ-PERF-001: Command latency shall be <2 seconds from web UI to device acknowledgment
REQ-PERF-002: Data transmission shall complete within 10 seconds for 30-second collection session
REQ-PERF-003: Web interface shall load session list in <3 seconds
REQ-PERF-004: System shall support up to 50 collection sessions per day

5.2 Reliability

REQ-REL-001: System shall gracefully handle WiFi disconnection during collection (buffer and retry)
REQ-REL-002: Device shall attempt automatic reconnection on connection loss
REQ-REL-003: Data transmission failures shall not result in data loss (retry mechanism)

5.3 Usability

REQ-USE-001: User shall be able to complete full collection workflow in <60 seconds
REQ-USE-002: Interface shall provide clear feedback for all user actions
REQ-USE-003: Error messages shall be descriptive and actionable

5.4 Maintainability

REQ-MAINT-001: System architecture shall support future multi-device expansion without major refactoring
REQ-MAINT-002: Data format shall be extensible for future sensor types or metadata
REQ-MAINT-003: MQTT topic structure shall accommodate future device clustering

---

6. Data Requirements

6.1 Data Models

Sensor Reading Data Point:

- Timestamp (milliseconds since session start)
- Sensor Position ID (1-3)
- Sensor Index (A/B)
- Proximity value (uint16)
- Ambient light value (uint16)

Session Metadata:

- Session ID (UUID)
- Device ID
- Start timestamp (ISO 8601)
- End timestamp (ISO 8601)
- Duration (seconds)
- Mode (debug/play)
- Sample count
- Labels (array of strings)
- Notes (free text)

6.2 Data Volume Estimates

- Maximum data points per session: 180,000 (30s × 1000 Hz × 6 sensors)
- Estimated sessions per day: 10-20
- Estimated monthly data volume: 15-30 sessions × 30 days = 450-900 sessions
- Storage requirement per session: ~3-5 MB (uncompressed JSON)

---

7. System Architecture Requirements

7.1 ESP32-S3 Firmware

REQ-ARCH-001: Firmware shall use dual-core architecture (Core 0: sensors, Core 1: networking)
REQ-ARCH-002: Firmware shall implement FreeRTOS queues for inter-core data transfer
REQ-ARCH-003: Firmware shall use MQTT protocol for cloud communication
REQ-ARCH-004: Firmware shall persist WiFi credentials in NVMe
REQ-ARCH-005: Firmware shall implement connection retry with exponential backoff

7.2 Cloud Infrastructure

REQ-ARCH-006: System shall use AWS IoT Core for MQTT broker
REQ-ARCH-007: System shall use AWS Lambda for backend logic
REQ-ARCH-008: System shall use DynamoDB for data storage
REQ-ARCH-009: System shall use API Gateway for web interface backend
REQ-ARCH-010: System shall use device certificates for IoT Core authentication

7.3 MQTT Topic Structure

REQ-ARCH-011: Device-to-cloud topics: `motionplay/{device_id}/data`, `motionplay/{device_id}/status`
REQ-ARCH-012: Cloud-to-device topics: `motionplay/{device_id}/commands`
REQ-ARCH-013: Topic structure shall support future multi-device grouping

---

8. Security Requirements

REQ-SEC-001: Device authentication shall use X.509 certificates
REQ-SEC-002: All data transmission shall use TLS encryption
REQ-SEC-003: Web interface authentication not required for Phase 1 (local development use)
REQ-SEC-004: Device certificates shall be securely provisioned during manufacturing/setup

---

9. Constraints

9.1 Technical Constraints

- Device: ESP32-S3 with T-Display module
- Network: Home WiFi environment (reliable connectivity assumed)
- Power: USB powered during Phase 1 development
- Development environment: macOS, VSCode, AWS CLI
- Frontend technologies: TypeScript/JavaScript frameworks

9.2 Timeline Constraints

- Target completion: 1-2 months
- Development time availability: 2-8 hours per week
- Phased implementation approach required

9.3 Budget Constraints

- AWS monthly cost target: $15-20/month acceptable
- No hard budget ceiling for Phase 1

10. Assumptions

1. Single device operation for Phase 1
2. Reliable home WiFi connectivity during testing
3. USB power available during all testing sessions
4. Developer has AWS account with appropriate permissions
5. No production users or external access during Phase 1
6. Data retention indefinite (manual cleanup only)
7. Desktop browser access sufficient (no mobile requirement)

11. Dependencies

1. Existing Motion Play hardware (main PCB v5.1, sensor PCB v5.4)
2. Functional sensor reading firmware on ESP32-S3
3. AWS account with IoT Core, Lambda, DynamoDB, API Gateway access
4. Development environment with Node.js/npm for frontend
5. KiCad project files and component datasheets for reference

12. Acceptance Criteria

The system shall be considered complete for Phase 1 when:

1. ✓ ESP32-S3 connects to AWS IoT Core and maintains stable connection
2. ✓ Web interface can send commands that device receives and executes
3. ✓ User can initiate data collection session from web interface
4. ✓ Device collects sensor data and transmits complete batch to backend
5. ✓ Web interface displays received sensor data
6. ✓ User can add labels to completed sessions
7. ✓ User can view list of historical sessions
8. ✓ User can export session data for external analysis
9. ✓ Device displays current mode and connection status on screen
10. ✓ System handles at least 20 collection sessions per day without issues

13. Future Considerations

Items documented for future phases but not implemented in Phase 1:

- Multi-device coordination via MQTT pub/sub
- Automated ML model training pipeline
- On-device inference capabilities
- Real-time streaming data visualization
- Mobile application interface
- User authentication and access control
- Game mode implementations and scoring logic
- Tournament coordination features

Document Approval:

Developer: Marc (Self-approved for hobby project)
Date: November 7, 2025

Revision History:

Version	Date	Author	Changes
1.0	Nov 7, 2025	Marc	Initial requirements document