**Université Mohammed 6 Polytechnique**
Digital Innovation Center of Excellence (DICE)

# Specifications document

**11 Feb 2025**

**Interactive Educational Robot**

**for Teaching Programming**

**Prepared By:** Mohamed Ayman Ouchker

**Supervised By:** Mr. Said Amharech

**Version:** 1.0

## Table of contents

## 1. Introduction

This document outlines the requirements for the development and design of an interactive educational robot aimed at teaching programming and basic robotics concepts to children. The purpose of this requirements document is to provide a clear, comprehensive, and agreed-upon description of the system's functionalities, performance targets, and technical constraints. It will serve as the roadmap for both the development and evaluation phases of the project.

---

## 2. Context and Pedagogical Objectives

**Project Context:**
The project is designed within an educational framework where the goal is to introduce children (aged 6 and above) to the fundamentals of programming and robotics. This interactive robot will be used in classroom settings and at home to stimulate logical thinking, creativity, and problem-solving skills. The robot's design will focus on simplicity and intuitiveness, ensuring it is accessible for young learners.

**Pedagogical Objectives:**

- **Introduction to Programming Concepts:**
  Teach basic programming constructs such as sequencing, loops, and conditional statements through interactive activities.
- **Development of Logical Thinking:**
  Enhance children's ability to think logically and systematically through problem-solving tasks.

- **Fostering Creativity:**
  Allow learners to design and modify movement sequences, thereby encouraging creativity and experimentation.
- **Interactive Learning:**
  Provide immediate feedback (visual and auditory) to reinforce learning and motivate students.

## 3. Functional Requirements

The system shall meet the following functional requirements:

- **User Interface:**
  - A graphical interface (mobile or web-based) that allows users to select basic commands (e.g., move forward, turn left, move backward, etc.).
  - A drag-and-drop or block-based programming environment for assembling command sequences.
- **Command Sequencing:**
  - Ability to create, store, and edit a sequence of commands that define the robot's path.
  - Display a preview of the programmed path before execution.
- **Real-Time Communication:**
  - Enable real-time transmission of commands from the application to the robot using wireless protocols (Bluetooth or WiFi).
  - Implement a simple, robust communication protocol ensuring that each command (e.g., "MOVE_FORWARD", "TURN_LEFT") is executed accurately.
- **Feedback Mechanism:**
  - The robot shall provide immediate visual (e.g., LED indicators) and/or auditory feedback for each executed command.
  - The system should notify the user in case of errors or unexpected behaviors.
- **Data Logging and Reporting:**
  - Record the sequence of commands executed along with performance data to support troubleshooting and future improvements.

---

## 4. Non-Functional Requirements

In addition to the functional requirements, the system must satisfy the following quality attributes:

- **Performance and Responsiveness:**
  - The robot should respond to commands with minimal latency to ensure a smooth and interactive learning experience.

- **Safety:**
  - All materials and components used in the robot must be child-safe, avoiding sharp edges or toxic substances.
  - The design should comply with applicable safety standards for educational toys.
- **Usability and Accessibility:**
  - The interface should be highly intuitive with clear icons and an attractive design tailored for young users.
  - Documentation (manuals, tutorials) should be provided in simple language for educators, parents, and children.
- **Reliability and Robustness:**
  - The robot should withstand frequent handling and be durable enough for repeated use in classroom environments.
  - Ensure robustness in wireless communication to avoid interruptions during operation.
- **Scalability:**
  - The system should allow future enhancements, such as adding new commands, sensors, or additional programming environments (e.g., switching from a visual to a text-based language).

---

# 5. Technical Constraints and Material Choices

**Hardware Platform:**

- **Main Controller & Power Management**
  - Raspberry Pi 4 Model B - Acts as the main processing unit (runs your Python code, sensor inputs, and motor control).

- 7.4v Airsoft LiPO Batteries (2200mAh 30C Mini)
    - DC-DC Buck Converter 7-24V to 5V 4A (SKU: DFR0831) - Steps down battery voltage to a stable 5V for the Raspberry Pi.
- **Motor & Motor Driver Components**
    - 2× DC Gear Motors with Encoders Small, geared DC motors (often “TT motors”) for differential drive; encoders enable precise movement control.
    - L298N Dual H-Bridge Motor Driver Module - Drives two DC motors with PWM speed control and directional switching.
- **Sensor Suite**
    - 10x Infrared sensors TCRT5000
    - Inertial Measurement Unit (IMU) - An MPU6050 or BNO055 module for measuring orientation and acceleration.
    - Ambient Light Sensor (LDR Module) - Light Dependent Resistor (LDR) with a fixed resistor to form a voltage divider.
- **User Interface & Feedback**
    - OLED Display Module (e.g., 0.96" I2C OLED) - Provides text or simple graphics to show status, instructions, or sensor readings.
    - RGB LEDs - For visual feedback—indicating status, errors, or programmed sequences.
    - Push Buttons - Allow for manual inputs (e.g., start/stop, mode selection).
    - Piezo Buzzer - For audible alerts or confirmation tones.
- **Additional Components & Accessories**
    - 3D Printed Chassis Components - Custom-designed parts for a durable, child-safe robot body.
    - Breadboard and Jumper Wires - For prototyping and connecting components.

**Software Environment:**

- **Programming Environment:**
    - A visual programming language (similar to Scratch) should be used for ease of use by beginners, with an option to switch to text-based programming (e.g., Python) for advanced users.
- **Application Development:**
    - Develop the control application for mobile and/or web platforms with a user-friendly graphical interface.
- **Communication Protocol:**
    - Design a simple protocol for translating UI commands into actionable instructions for the robot.

**Budget and Timeline Constraints:**

- **Cost:**
    - Define an estimated budget covering the procurement of components, development of the application, and prototyping.
- **Timeframe:**
    - Establish a project timeline with milestones for prototype development, user testing, and final deployment. Include iterative testing phases with feedback from educators and children.

**Chassis & Structure Example:**

| | |
|---|---|
| | |

| | |
|---|---|
| | |
| | |

## 6. Planning

| Phase | Task | Description | Start Week | End Week | Status | Notes |
|---|---|---|---|---|---|---|
| Planning & Research | Define Requirements | Finalize hardware & features | 1 | 2 | In Progress | Requirements document |
| Planning & Research | Order Components | Purchase electronic and mechanical parts | 1 | 2 | Not Started | - |
| Planning & Research | Set Up Development Environment | Install Raspberry Pi OS libraries and 3D modeling tools | 1 | 2 | Not Started | - |
| Mechanical Design & Prototyping | Chassis Design & Fabrication | Design and 3D print a modular chassis | 3 | 6 | Not Started | - |
| Mechanical Design & Prototyping | Motor Integration | Mount and test motors with L298N driver | 3 | 6 | Not Started | - |
| Electronics & Sensor Integration | Power System Setup | Integrate LiPo battery & voltage regulators | 7 | 10 | Not Started | - |
| Electronics & Sensor Integration | Sensor Integration | Implement IR proximity line tracking IMU and light sensors | 7 | 10 | Not Started | - |
| Electronics & Sensor Integration | User Feedback Components | Program OLED display LEDs buzzer and buttons | 7 | 10 | Not Started | - |
| Software & Wireless Communication | Path-Execution Algorithm | Develop movement processing and logic | 11 | 14 | Not Started | - |
| Software & Wireless Communication | Wireless Communication | Enable WiFi-based remote control | 11 | 14 | Not Started | - |
| Software & Wireless Communication | Basic Control App | Develop initial web/mobile interface | 11 | 14 | Not Started | - |
| Application & Final Integration | Software Refinements | Improve error handling and sensor fusion | 15 | 18 | Not Started | - |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| Application & Final Integration | GUI Enhancement | Upgrade UI for user-friendly programming | 15 | 18 | Not Started | - |
| Testing & Optimization | Full System Testing | Test real-world use cases and optimize movement | 19 | 22 | Not Started | - |
| Testing & Optimization | Performance Improvements | Enhance battery efficiency and response time | 19 | 22 | Not Started | - |
| Testing & Optimization | Final Documentation | Create guides diagrams and educational exercises | 19 | 22 | Not Started | - |
| Final Adjustments & Deployment | Final Iterations | Apply refinements based on user feedback | 23 | 24 | Not Started | - |
| Final Adjustments & Deployment | Demo & Release | Prepare presentation demo and next steps | 23 | 24 | Not Started | - |