
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: 2.0

Table of contents

1. Introduction.....	
2. Context and Pedagogical Objectives.....	
3. Functional Requirements.....	
4. Non-Functional Requirements.....	
5. Requirements & Materials.....	
6. Planning.....	

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.

Overview:

- Build a simple, 3D printed robot (using the established LittleBots design) controlled by an Arduino Nano.
- Develop a custom Android app where kids can visually build a sequence of commands to define the robot's path and progress through increasingly complex programming levels.

Objectives:

- Teach foundational programming concepts (sequencing, debugging, modularity) in a fun, interactive way.

-
- Combine mechanical design, electronics, firmware development, and app programming into one interdisciplinary project.
 - Provide an engaging tool to introduce children to coding and autonomous robotics.

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.
 - **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.

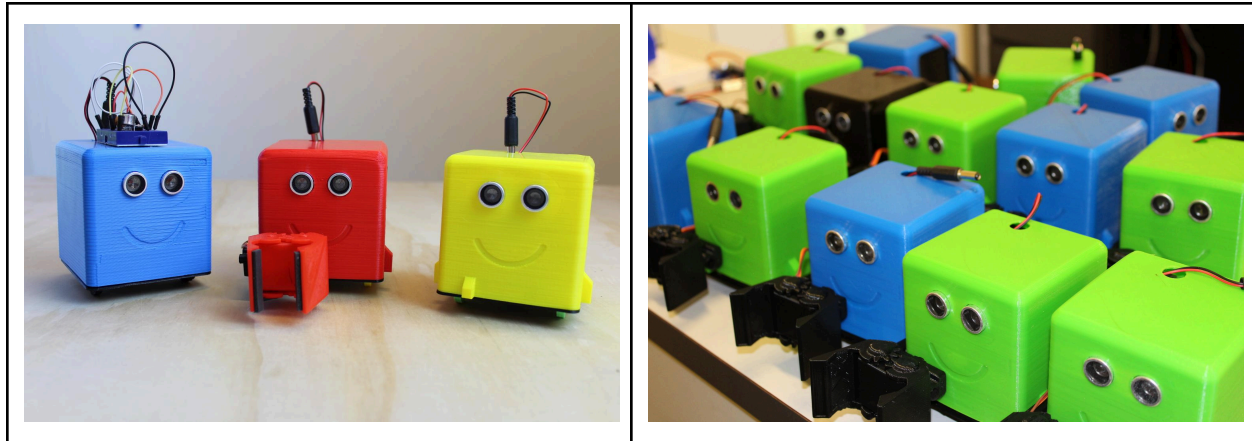
-
- **Safety:**
 - All materials and components used in the robot must be child-safe, avoiding sharp edges or toxic substances.
 - The design should comply with applicable safety standards for educational toys.
 - **Usability and Accessibility:**
 - The interface should be highly intuitive with clear 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. Requirements & Materials

Hardware:

- Arduino Nano
- Continuous rotation servos
- HC-06 Bluetooth module
- Ultrasonic sensor
- LiPo battery (3.7V)
- TP4056 Module charger

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	7	10	Not Started	-

Electronics & Sensor Integration	User Feedback Components	light sensors 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	-