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

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).

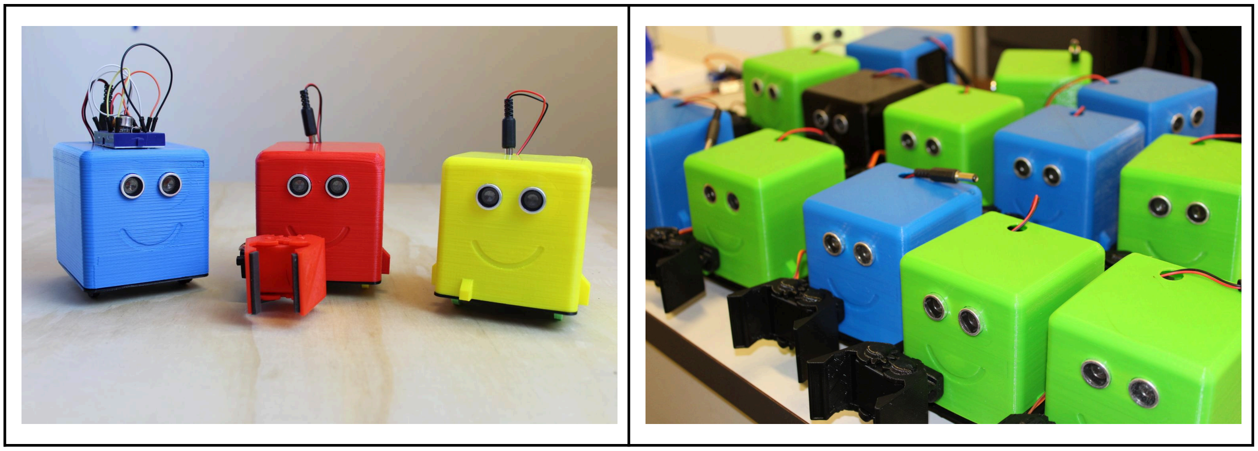
---

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. Programming Levels & Features

Level 1: Basic Movement

- **Objective:** Introduce simple movement commands.
- **Blocks Available:** Forward, Backward, Left, Right, Stop.
- **Focus:** Understanding sequencing and immediate motor responses.

Level 2: Path Planning

- **Objective:** Teach kids to design and sequence multiple moves to form a complete path.
- **Blocks Available:** Basic movement blocks arranged to form a route.
- **Focus:** Spatial reasoning, planning, and visualizing the robot's path.

Level 3: Sensor Integration (Ultrasonic Sensor)

- **Objective:** Incorporate obstacle detection.
- **New Blocks:** Sensor blocks that read distance values; conditional blocks (if/else) to handle obstacle detection.
- **Focus:** Understanding how sensor input influences decisions and integrating simple logic.

Level 4: Manipulation (Gripper Integration)

- **Objective:** Expand functionality to include object manipulation.
- **New Blocks:** Gripper controls (Open, Close).
- **Focus:** Combining movement, sensor input, and manipulation to perform tasks (e.g., picking up and delivering objects).

Level 5: Auto Mode (Autonomous Navigation)

- **Objective:** Enable the robot to memorize and autonomously navigate a path without direct sequential commands.
- **New Blocks:**
  - **Auto Mode Block:** Initiates an autonomous routine using preprogrammed navigation algorithms.
  - **Path Memory & Correction Blocks:** Allow the robot to store a learned path and adjust based on real-time sensor data (from ultrasonic sensors or line sensors).
- **Features:**
  - The robot uses previously learned commands and sensor feedback to autonomously follow a path and avoid obstacles.
  - Introduces more advanced algorithms like PID control or even simple path memorization techniques inspired by research.

- **Focus:**

  - Teaching autonomous decision-making and error correction.

  - Allowing the robot to operate with minimal intervention, enhancing its real-world application potential.