INNOVATIONS
MACHERIN

ENTWICKLERDOKUMENTATION

RoBo SDG JUNIOR

netidee Call 19 · Projekt 7392

Wien 2024–2026

RoboSDG Junior
Developer Documentation

Table of Content

1. Introduction

RoboSDG Junior is a browser-based educational robotics platform designed to introduce children to programming, robotics and creative problem solving through highly accessible, hands-on learning experiences.

The platform was developed as part of the RoboSDG Junior project by InnovationsMacherIN and builds upon earlier Robo4Girls and Robo4Earth concepts. While the original systems focused mainly on Blockly-based programming environments, RoboSDG Junior introduces a modern React-based architecture with improved scalability, accessibility and future learning possibilities.

The software is designed for use in workshops, schools, MINT regions and informal educational settings. Learners use the interface to program robots based on the BBC micro:bit microcontroller through visual block programming and text-based code views.

A central goal of RoboSDG Junior is lowering barriers to entry into programming and robotics while simultaneously creating future pathways toward advanced topics such as AI, physical computing and creative prototyping.

2. Educational Philosophy

RoboSDG Junior follows several core educational principles:

- **Hands-on learning** through tangible robotics experiences
- **Low-threshold programming access** for beginners
- **Inclusive interface design**
- **Creative experimentation**
- **Purpose-driven technology education**
- **Bridge learning** between visual and text-based programming
- **Future-oriented digital literacy**, including AI concepts

The platform intentionally supports both beginner and advanced learners through a dual-view approach:

- visual block-based programming
- text-based JavaScript/Python code representation

This enables gradual progression instead of forcing learners into one fixed programming model.

3. Technical Overview

The RoboSDG Junior programming interface is a web-based application built with:

- React
- Vite
- JavaScript
- Web Bluetooth API
- BBC micro:bit

The platform communicates directly with the robot via Bluetooth and allows learners to create programs through drag-and-drop programming blocks.

Compared to earlier Blockly-only systems, the React-based architecture provides:

- improved modularity
- better maintainability
- easier scalability
- cleaner UI management
- future compatibility for advanced learning modules

The current architecture was designed with future educational expansion in mind, including:

- gamification features
- AI learning modules
- advanced robotics projects
- drawing robots
- neural network demonstrations

4. Repository Structure

GitHub Repository: [RoboSDG Junior GitHub Repository]

The project structure is organized as follows:

```
├── src
│   ├── assets
│   │   ├── icons/
│   │   └── fonts/
│   ├── components
│   │   ├── ProgrammingInterface.jsx   # Main application component
│   │   ├── blocks/
│   │   ├── bluetooth/
│   │   ├── navigation/
│   │   └── programming/
│   ├── config
│   │   ├── blocksConfig.js            # Configuration for block categories and properties
│   │   └── blockIconConfig.jsx        # Maps block types to icons
│   ├── locales
│   │   └── translations.js            # Language translations
│   ├── styles
│   │   ├── components/
│   │   ├── global.css
│   │   └── variables.css
│   ├── utils
│   │   ├── blocksConverter.js         # Converts blocks to robot commands
│   │   ├── blockStorage.js            # Session storage helpers
│   │   └── useTouchDrag.js            # Custom hook for touch-based drag and drop
│   ├── i18n.js                        # i18next configuration
│   └── main.jsx                       # Application entry point
├── index.html                         # Main HTML file
├── package.json                       # Project dependencies and scripts
└── vite.config                        # Vite configuration
```

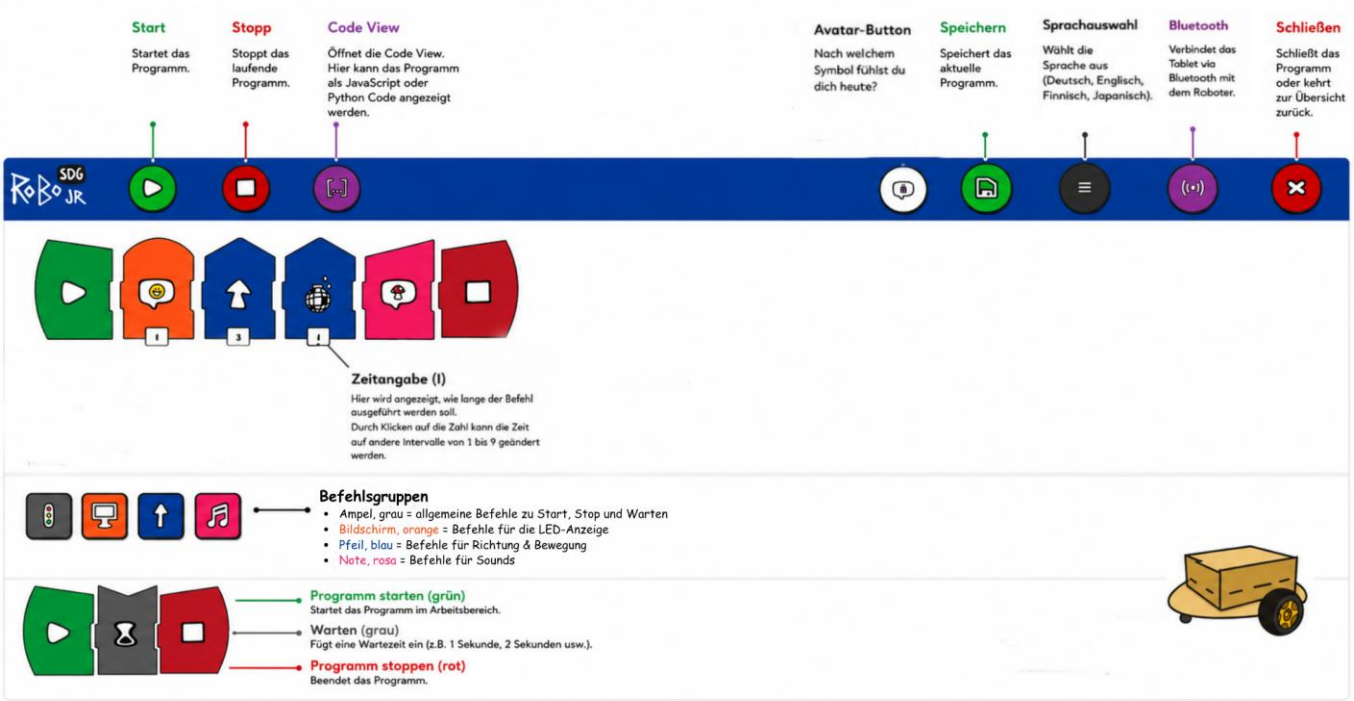
5. Core Features

5.1 Visual Programming Interface

The platform provides a drag-and-drop visual programming system where users arrange command blocks to create robot behaviors.

Features include:

- block reordering
- adjustable parameters
- categorized commands
- zoom and pan navigation
- touch support for tablets



5.2 Dual-View System

One of the most important architectural changes was the introduction of a dual-view system.

Users can switch between:

- block-based programming
- textual code preview

This creates a bridge between beginner-friendly learning and advanced programming understanding.

The code preview currently supports:

- JavaScript
- Python

5.3 Bluetooth Communication

The platform uses the Web Bluetooth API to connect directly with BBC micro:bit devices.

Compared to earlier implementations, the Bluetooth communication layer was redesigned to:

- improve reliability
- reduce communication errors
- increase responsiveness
- simplify classroom usage

The platform requires:

- HTTPS hosting or localhost
- browser support for Web Bluetooth
- micro:bit advertising Nordic UART Service

Chrome-based browsers are recommended.

5.4 Session Storage

Programs are automatically saved within the browser session using session storage utilities.

This allows learners to continue unfinished work, recover from accidental reloads, work more independently in workshops.

5.5 Multilingual Support

The platform includes multilingual support via i18next. Supported languages currently include: English, German, Finnish, Japanese

Additional languages can be added through: src/locales/translations.js

6. Installation & Local Development

Prerequisites

Install the following software:

- Node.js
- npm
- Git
- Visual Studio Code (recommended)

Recommended VS Code extensions:

- GitHub Pull Requests
- ESLint
- Prettier
- JavaScript ES6 Snippets

Clone Repository

git clone https://github.com/InnovationsMacherIN/robosdgjunior.git

Install Dependencies

npm install

Start Development Server

npm run dev

The local server will usually run on: http://localhost:5173

7. Building for Production

To create a production-ready build: npm run build

The optimized files will be generated inside: dist/

These files can be deployed on any static hosting provider.

8. Micro:bit Integration

RoboSDG Junior communicates with the BBC micro:bit via Bluetooth.

Key hardware components:

- BBC micro:bit
- motor driver board
- servo/motor setup
- battery pack
- optional sensors and extensions

The platform was intentionally designed to remain flexible for future robotics projects.

9. Future Extensions

The architecture was designed for scalability and future educational experimentation.

10. Troubleshooting

Bluetooth Connection Problems

Check:

- HTTPS usage
- browser compatibility
- Bluetooth permissions
- Nordic UART Service support

Missing Dependencies

Reinstall packages: npm install

Session Storage Issues

Ensure the browser allows:

- local storage
- session storage

Translation Errors

Verify:src/locales/translations.js and: src/i18n.js

11. Maintenance Recommendations

For stable workshop operation:

- regularly test Bluetooth connections
- update dependencies carefully
- maintain consistent folder structure
- test micro:bit firmware compatibility
- avoid changing critical communication protocols without full testing

Before educational deployments:

- test on tablets and laptops
- test touch interactions
- test browser compatibility
- test accessibility features

12. Contributors

The RoboSDG Junior platform was developed collaboratively by:

Sonja Macher (Educational concept and workshop integration)

Anniina Pohjola (Graphic Design)

Atte Haataja (Programming architecture and React implementation)

13. License

This project follows the licensing structure of the upstream Robo4Earth repository.

Please refer to the GitHub repository for the current license details.