System Requirements Specification

for

EGR101 Simulation

Version 3.0 approved

Prepared by:   
Emily Connearney  
Luke Crump  
Vivian Dang  
Keely Mashburn  
Daniel Khalil

Embry Riddle Aeronautical University

3/31/2022

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 3

2.4 Operating Environment 3

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 4

3. External Interface Requirements 4

3.1 User Interfaces 4

3.2 Software Interfaces 7

3.3 Communications Interfaces 7

4. System Features 8

4.1 Wiring and Design Interface 8

4.2 Arduino IDE 9

4.3 Bot Simulation 10

4.4 Arduino Emulation 11

5. Nonfunctional Requirements 11

5.1 Performance Requirements 11

5.2 Safety Requirements 12

5.3 Security Requirements 12

5.4 Software Quality Attributes 12

5.5 Business Rules 12

Appendix A: Glossary 12

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Daniel Khalil  Vivian Dang | 9/18/21 | Deliverable | 1.0 |
| Daniel Khalil  Vivian Dang | 10/26/21 | Deliverable | 2.0 |
| Luke Crump | 12/4/21 | Edited Format | 2.1 |
| Luke Crump | 1/31/22 | Correcting document for deliverable | 2.2 |
| Luke Crump | 2/2/2022-2/3/2022 | Adding pictures to the User Interface section and adding page descriptions | 2.3 |
| Luke Crump | 3/2/2022 | Adding suggestions made by Dr. Akbas and TA. | 2.4 |
| Luke Crump | 3/31/2022 | Changed document to current tense. Changed awkward grammar throughout document and added some requirements. Edited older requirements to be current. Added new photos. | 3.0 |

# Introduction

## Purpose

The EGR 101 Simulator is designed to supplement learning for remote students in the course EGR101. The course utilizes an expensive Parallax Boe-Bot Robot Kit. EGR 101’s main project deliverables include grades based on performance in four Boe-Bot courses built to challenge students on forming solutions to: basic line following, line following corrected for noise, object avoidance, and resource management. Normally students are split into groups of three, with each group receiving a Boe-Bot kit, which includes its respective sensors, LEDs, and resistors. Due to COVID-19, the role of the course has changed due to variability of student in-person attendance.

## Document Conventions

This document intends to define all requirements and conditions associated with the EGR101-Simulation System. This document covers the product itself, its interaction with the user, and the requirement associated with bringing the system in compliance with the EGR101 course vision.

## Intended Audience and Reading Suggestions

The intended users of this product include students, and the instructor of the EGR101 course. This document contains information on product functionality, requirements, standards, and regulations given by the current EGR101 instructor. Section 2 describes the intent of the project. Section 3 describes the technical aspect most pertinent to the other software developers maintaining or enhancing the project. Section 4 and 5 describe the features and standards the project follows and be of interest to instructors interested in understanding the scope of this tool. Along with reading this document, the references outlined in section 1.5 are important to know for the proper operation and testing of the project.

## Product Scope

The scope of this application is to reduce the cost of eventual replacement of Boe-Bot kits, allow for remote learning through testing electronic based solutions in a sandbox environment, and ease of grading said electronic-based solutions. The proposed project allows students to program Arduino sketches, design a virtual bot through adding components and wire connections, test their virtual bot on the 4 deliverable courses, and provide a sandbox environment to improve understanding of basic circuitry and imperative programming.

## References

*[1]* “Language reference,” *Arduino Reference - Arduino Reference*. Available: https://www.arduino.cc/reference/en/. [Accessed: 02-Mar-2022].

*[2]* “Boe-Bot Robot Kit - USB,” *Parallax*. Available: https://www.parallax.com/product/boe-bot-robot-kit-usb/. [Accessed: 02-Mar-2022].

# Overall Description

Section 2 covers purpose of the product in 2.1, the product functions in 2.2 associated user information in 2.3, and the operating environment in 2.4, Sections 2.5, 2.6 and 2.7 contain additional information on the implementation of the functions outlined in section 2.2

## Product Perspective

The EGR101 Simulation System acts as a sandbox to test Arduino code and wiring configurations from the Arduino to its individual components on the simulated bot. The application is controlled from the core Java application which initializes a Unity application displaying 3-Dimensional Bot customization and simulation display. This would put both the core application thread of execution and the Unity thread of execution in parallel. These threads communicate with each other over TCP.

## Product Functions

### Simulate an Arduino IDE to allow users to program virtual bots.

### Emulate Arduino and allow IDE to compile and change Arduino behavior at runtime.

### Allow components to be added to the bot configuration.

Allow components to be added to the bot configuration.

Allow for wired connections between components and Arduino to be specified.

Simulate component functionality based off wired connections.

Display 3-Dimensional autonomous bot course simulation.

Produce recording of 3-Dimensional course simulation.

Produce zipped configuration files for wired connections to components, component list, and Arduino sketches.

## User Classes and Characteristics

Figure 1. A Conceptual Diagram showing the basic relationship between the major subsystems.

### Student

The student is the default user intended to interact with this system. The student is not expected to understand how the Arduino API functions, or how to properly wire Arduino components. Students interact with the Arduino IDE to produce code, update the bot configurations through adding/removing components, and wiring the components to power, ground, and IO ports on the Arduino, students can also test their configured bots on the 4 deliverable courses. This application is mainly focused on providing a user-friendly experience for these students.

### Instructor

The instructor is a secondary user who intends to interact with this system. The instructor can create new maps and components. The instructor can download a student’s code and run it for grading purposes.

## Operating Environment

The system requires Java to be installed on the machine to run the interface. Design constraints dictate that this application must work with low-end hardware as it might be used on school computers.

## Design and Implementation Constraints

* System must only have the components in the kit provided to the EGR101 students.
* The instructor of EGR 101 is responsible for maintaining the delivered code.
* Cannot have functionality or components that the actual robot and kit do not have.
* System must operate on school computers, which have the following specifications:
  + Processor – Intel i7-10700
  + GPU – Nvidia Quadro P1000
  + Memory – 16GB DDR4
  + Storage – 1TB NVMe

## User Documentation

This system will be provided to the product owner with the documentation listed below:

* Software Requirement Specification Document
* Software Design Specification Document
* Testing Plan
* Arduino Documentation

## Assumptions and Dependencies

* Students are familiar with the kit and its components.
* Students understand assigned task, given by the instructor.
* Only the professor can add additional components and maps to the simulator.

# External Interface Requirements

Section 3 covers information about the software interface. Sections 3.1 covers the three different interfaces. Section 3.2 covers the third-party software that must be downloaded. Section 3.3 covers how data is sent between the three interfaces that are shown in section 3.1.

## User Interfaces

There will be 3 interfaces that the users of the system can interact with:

Bot Customization/Wiring GUI

The bot customization and wiring GUI will used to virtually apply hardware changes to the bot. Components, and mounts can be selected, moved, and added to the virtual bot. The wiring interface will allow for the user to add, modify, or delete connections between the components and the Arduino sitting atop the bot.

A picture containing text, projector

Description automatically generatedGraphical user interface

Description automatically generated3D Simulation

Figure 3. Wiring scene where users can add wires between components

Figure 2. The scene components can be added to the bot.

The 3D simulation interface will provide a display for the user to view behavior of the Boe-bot after successfully compiling Arduino code, adding components, and wiring those components. There shall be a play, pause, and replay button for the user to have control over the viewing of the simulation. There will also be the option to rotate the camera as the user to see multiple perspectives of the Boe-bot running the course.

### A picture containing text, way, road, scene Description automatically generatedArduino IDE

Figure 4. Simulation scene with first project board.

The Arduino IDE will be the main hub that connects each of the subsystem interfaces together. The user will be able to code and build and Arduino sketch file, launch the bot design and wiring interface, and execute on a selected course.

Figure 5. Arduino interface with code example.

Graphical user interface, text

Description automatically generated

#### Arduino Page: This page shows the coding interface for the user. This is the main interface that will be interacted with by the user.

#### File Tab Page: This page shows the open, save, and new file button.

#### Sketch Tab Page: This page shows the user two buttons to upload the text in the editor and to verify and compile the code.

#### Toggle Tab Page: This page shows the serial monitor.

#### Button Bar: The buttons in the darker green color will have the following abilities:

##### Open Simulation Window

##### Open Bot Customization Window

##### Create, Save, and Open a file

## Software Interfaces

External Software Interfaces used in this project include Java JDK 1.8.0\_301 and Unity v2020.3.18f.

## Communications Interfaces

There shall be a system of communication between two major subsystems in this product. It will be required that the Unity executable can communicate with the Java application at runtime. This will be a local host socket connection between the two applications in which both applications can send packets to each other via a TCP connection. There will be 3 main scenarios in which information will be passed between the two running applications.

1. **Bot Customization**: If a component is added to the 3D Bot in the Bot-Customization GUI a packet shall be sent to prompt the java application to add that component to the list of components saved in the virtual Arduino.
2. **Bot Wiring**: if a wiring connection between Arduino pins and a component pin is present a packet shall be sent to prompt the java application to add the connection to the list of connections saved in the virtual Arduino.
3. **3D Simulation:** on execution of the 3D simulation of Boe-bot on a course a bidirectional communication shall occur between both applications as the Emulated Arduino will execute on the java application and send the Simulation (in Unity) component behavior information. Likewise, simulated sensors in Unity shall pick up data from the environment and update the sensor readings in the Arduino Emulation.

# System Features

## Wiring and Design Interface

Section 4.1 defines a list of requirements for the wiring and design interface.

Description and Priority

The Wiring and Design Interface will allow the user to design a robot in 3-dimensional space. This would include adding components, mounts, and cosmetic changes. The wiring and design GUI will also allow the user to connect components to the built in Arduino to make functional circuits.

Functional Requirements

#### The system shall allow the user to select a pin in the wiring interface.

#### The system shall allow the user to connect a wire between two pins in the wiring interface.

#### The system shall record the selected pin locations that are connected by a wire in a text file.

#### The system shall be able to connect wires between a resistor and a pin.

#### The system shall disable the components from being placed on the bot when the user is in the Wiring interface.

#### The system shall save the current wire configuration when the user exits the Wiring interface.

#### The system shall have a button to exit the Wiring and Design Interface.

#### The system shall exit to the Arduino interface from the Bot Customization interface.

#### The system shall exit to the Arduino interface from the Wiring interface.

#### The system shall have a button to add a resistor to the board while the user is in the Wiring interface.

#### The system shall enable requirements 4.1.2.1-4.1.2.6 when the scene changes to the Wiring interface.

#### The system shall move components that are attached to the bot with the bot when it is rotated in the Wiring interface.

#### The system shall allow the user to move a component after it has been placed on a bot.

#### The system shall allow the user to place a component on the bot at a predetermined location.

## Arduino IDE

Section 4.2 defines a list of requirements for the Arduino IDE.

Description and Priority

The Arduino IDE will allow for the user to program and build Arduino code. The IDE will also be a hub for traversing the application as the interface can connect with the Wiring and Design interface as well as the 3D Simulation.

Functional Requirements

#### The system shall have a File button that shows file operations when clicked on.

#### The system shall have a save button appear when the File button is clicked

#### The system shall save the Arduino script when the save button is clicked.

#### The system shall have an open button appear when the File Button is clicked.

#### The system shall open a new Arduino script when the New File button is clicked.

#### The system shall have a “save configuration” button appear when the File button is clicked.

#### The system shall save a configuration file when the “save configuration” button is clicked.

#### The system shall produce a configuration file that contains the wiring setup and the Arduino code when the “save configuration” button is pressed.

#### The system shall open the Serial window when the Serial window button is pressed.

#### The system shall print in console the status of the compilation of the current file.

#### The system shall have a line counter on the left side of the Arduino interface.

#### The system shall build the code when the Check Mark button is pressed.

#### The system shall print to console “Building…” when the Check Mark button is pressed.

#### The system shall open the simulation when the Run button is pressed.

#### The system shall create a TCP connection to the Unity simulation to send commands to the bot when the Run button is pressed.

#### The system shall open a new Arduino interface when the user clicks the Create New Interface button.

## Bot Simulation

Section 4.3 defines a list of requirements for the Bot Simulation.

Description and Priority

Bot Simulation will be used to view the behavior of the Arduino based on changes done in the bot design and wiring, as well as emulated Arduino behavior changes due to code inserted into the IDE.

Functional Requirements

#### The system shall prompt the user to select a course, the course selected will be displayed and executed on.

#### The emulation thread shall send each components behavior to the simulation UI via TCP connection per cycle.

#### The simulation UI shall send each components sensor data to the Arduino Emulation via TCP connection per update iteration.

#### The system shall have a “save configuration” button appear when the File button is clicked.

#### The simulation UI shall update the Boe-Bot’s rotation if one wheel is rotating

#### The simulation UI shall update the Boe-Bots position if both wheels rotate

#### The system shall display console logs in the Arduino IDE during simulation execution if logging is present within Arduino code.

#### The system shall restart the simulation if the restart button is selected

#### The system shall end the simulation if the end button is selected.

#### The system shall pause the simulation if the pause button is selected.

#### The system shall play the simulation if the play button is selected.

#### The system shall not allow the Boe-bot to change position if it is colliding with an obstacle in the course.

#### The system shall have a save button appear when the File button is clicked.

## Arduino Emulation

Section 4.4 defines a list of requirements for the Arduino Emulation.

Description and Priority

This Feature simulates Arduino pin connections and component behavior. It monitors, modifies, and decides whether components will actually work as intended through providing power and ground connections. The User can update delays and send power through specific IO pins.

Functional Requirements

#### The system shall allow for pin connections between the Arduino pins and components.

#### The system shall simulate the Arduino clock by counting in microseconds after executing.

#### The system shall allow for calls from Arduino code to delay programmable interaction with the components.

#### The system shall match the behavior defined from compiled Arduino code.

#### The system shall allow for digital writing to pins by providing voltage via simulated Pulse Width Modulation (PWM).

#### The system shall allow for analog writing to pins by providing a voltage.

#### The system shall provide traditionally used libraries like Servo.h and Serial.h to interface with.

#### On execution the system shall push voltage to each pin sequentially beginning at the IO and power ports of the Arduino.

#### On execution the components shall work if and only if they have adequate ground connection, and enough voltage to satisfy the potential of the component.

# Nonfunctional Requirements

## Performance Requirements

The System Simulation Menu shall execute at 30 FPS on low-end hardware PCs.

The System Bot Customization Menu shall execute at 30 FPS on low-end hardware PCs.

The system shall be less than 300 MB.

## Safety Requirements

Safety Requirements will not be necessary as all subsystems composing this project are software based.

## Security Requirements

Security Requirements will not be necessary as no subsystems composing this project handle personal/private information.

## Software Quality Attributes

**Readability, Maintainability, Reusability**

This software package is designed with the intention of being easily modified due to the changing nature of the EGR101 course. If the instructor or any other party will modify the code or add program additions the existing code base shall be concise, atomic, and unambiguous. There shall be java-doc comment notation for code within the EGR101 code base. Abstracted objects used by the code base IE: inline Arduino APIs and Components shall be modular and easily manufactured.

**Portability**

This software package is designed to work on Windows 10 & MAC OSX both software executables will be packaged within the same repository.

## Business Rules

Section 5.5 Business Rules explains how different users will interact with the system to describe functionality utilized by both individuals. No functionality is disabled for either user, but both users will interact with the system in different ways.

**Instructor**

The instructor will interact with the system to upload configuration files, view the code, bot configuration, and simulation results.

**Student**

The student will interact the system to create code, add and remove components via the bot design menu, and run simulations of the virtual bot, and generate configuration files for the instructor.

Appendix A: Glossary

**Arduino**- Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

**EGR-101**- A Course Offered by Embry-Riddle Aeronautical University called Engineering 101, which gives students experience with robotics, and programming.

**Boe-Bot**- A robot configuration that contains a frame and a caster wheel with two continuous servo motors, and an Arduino.

**Transmission Control Protocol (TCP)** – A messaging protocol that provides reliable, ordered, and error-checked delivery of a stream of bytes between the host and the application.