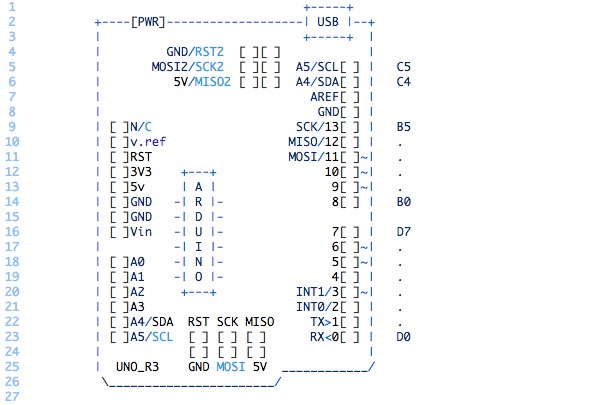
System Requirements Specification for

EGR101 Simulation System

Emily Connearney  
Luke Crump  
Vivian Dang  
Keely Mashburn  
Daniel Khalil



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Date | Reason for Change | Initials | Version |
| Initial Release | 9/22/21 | Initial Release | ALL | 1.0 |
|  |  |  |  |  |
|  |  |  |  |  |

2

**Contents**

[1 Introduction 2](#_Toc83888446)

[1.1 Purpose 3](#_Toc83888447)

[1.2 Intended Audience and Reading Suggestions 3](#_Toc83888448)

[1.3 Product Scope 3](#_Toc83888449)

[1.4 References 3](#_Toc83888450)

[2 Overall Description 3](#_Toc83888451)

[2.1 Product Perspective 4](#_Toc83888452)

[2.2 Product Functions 4](#_Toc83888453)

[2.3 User Classes and Characteristics 5](#_Toc83888454)

[2.3.1 EGR101 Student: 5](#_Toc83888455)

[2.3.2 EGR101 Instructor: 5](#_Toc83888456)

[2.4 Operating Environment 5](#_Toc83888457)

[2.4.1 User Interface: 5](#_Toc83888458)

[2.4.2 Data Collection: 5](#_Toc83888459)

[2.4.3 Data Transmission: 5](#_Toc83888460)

[2.5 Design and Implementation Constraints 5](#_Toc83888461)

[2.6 User Documentation 5](#_Toc83888462)

[2.7 Assumptions and Dependencies 5](#_Toc83888463)

[2.8 User Interfaces (Software) 5](#_Toc83888464)

[2.9 Hardware Interfaces 5](#_Toc83888465)

[2.10 Software Interfaces 5](#_Toc83888466)

[2.11 Communications Interfaces 5](#_Toc83888467)

[3 System Features 5](#_Toc83888468)

[3.1 Wiring and Design Interface 6](#_Toc83888469)

[3.1.1 Description and Priority 6](#_Toc83888470)

[3.1.2 Stimulus/Response Sequences 6](#_Toc83888471)

[3.1.3 Functional Requirements 6](#_Toc83888472)

[3.1.3.1 The system shall allow the user to select a wire. 6](#_Toc83888473)

[3.1.3.2 The system shall allow the user to move the wire to connect two pin locations. 6](#_Toc83888474)

[3.2 Arduino IDE 6](#_Toc83888475)

[3.2.1 Description and Priority 6](#_Toc83888476)

[3.2.2 Stimulus/Response Sequences 6](#_Toc83888477)

[3.2.3 Functional Requirements 6](#_Toc83888478)

[3.3 PIGEONS Mission Data Viewer 7](#_Toc83888479)

[3.3.1 Description and Priority 7](#_Toc83888480)

[3.3.2 Stimulus/Response Sequences 7](#_Toc83888481)

[3.3.3 Functional Requirements 7](#_Toc83888482)

[4 Other Nonfunctional Requirements 10](#_Toc83888483)

[4.1 Performance Requirements 10](#_Toc83888484)

[4.2 Safety Requirements 11](#_Toc83888485)

[4.3 Security Requirements 11](#_Toc83888486)

[4.4 System Quality Attributes 11](#_Toc83888487)

[4.5 Business Rules 12](#_Toc83888488)

[5 Appendix A: Glossary 12](#_Toc83888489)

[6 Appendix B: Analysis Models 13](#_Toc83888490)

# Introduction

This Product is designed to supplement learning for remote students in the course EGR101. The course utilizes a $229.00 Parallax Boe-Bot Robot Kit to allow students to design a functionally autonomous robot. EGR101’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 the recent pandemic, the role of the course has changed due to variability of student in-person attendance. The current solution to this problem was to make students pay $85 for their own kits and perform the required deliverables remotely.

## Purpose

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 should 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 would be 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 would Allow 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. This product could be used in applications far beyond the scope of this course as a virtual electronics test environment could be in-valuable to autonomous vehicle testing

## References

*[1] Language reference*. Arduino Reference - Arduino Reference. (n.d.). Retrieved September 30, 2021, from <https://www.arduino.cc/reference/en/>.

# 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 will act as a sandbox to test Arduino code and wiring configurations from the Arduino to its individual components on the simulated bot. The application should be controlled out of the core Java application which will initialize 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 will communicate with each other over TCP.

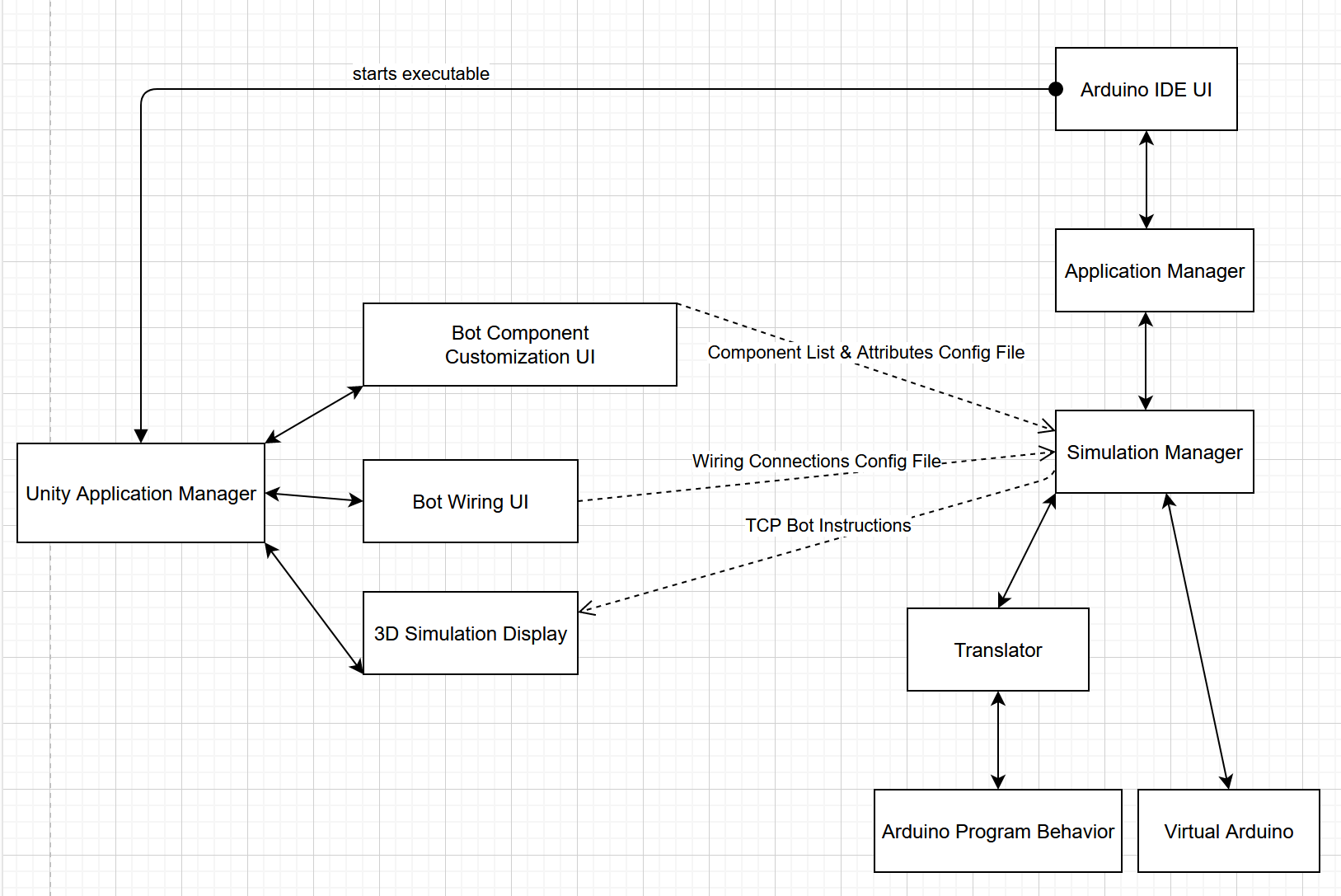


Figure 1: System Configuration Diagram

## Product Functions

2.2.1 Simulate an Arduino IDE to allow users to program virtual bots

2.2.2 Emulate Arduino and allow IDE to compile and change Arduino behavior at runtime

2.2.3 Provide Arduino API that will modify the emulated Arduino.

2.2.4 Allow components to be added to the bot configuration

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

2.2.6 Simulate component functionality based off wired connections.

2.2.7 Display 3-Dimensional autonomous bot course simulation

2.2.8 Produce recording of 3-Dimensional course simulation

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

## User Classes and Characteristics

### 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 will 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 will be 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. This individual will most likely have proficient knowledge in the systems this application is emulating. This user will require students to have configuration files exported and imported to help ease of grading. There will also be a system in which students must fill out a profile which will be injected into the configuration file which will give the instructor information about student name, student ID, and any other information needed for the identity of the student.

## Operating Environment

### User Interface:

### Data Collection:

### Data Transmission:

## Design and Implementation Constraints

1. Design Constraints:
2. Implementation Constraints:

## User Documentation

## Assumptions and Dependencies

## User Interfaces (Software)

## Hardware Interfaces

## Software Interfaces

## Communications Interfaces

# System Features

## Wiring and Design Interface

### Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

### 3.1.3.1 The system shall allow the user to select a wire.

### 3.1.3.2 The system shall allow the user to move the wire to connect two pin locations.

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

3.1.3.4 The system shall have a button to save the current wire configuration.

3.1.3.5 The system shall save the current configuration of wires when the “save” button is pressed.

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

3.1.3.7 The system shall exit to the main view screen when the exit button is pressed.

3.1.3.8 The system shall have a button to add a resistor to the board.

3.1.3.9 The system shall be able to connect resistors to pins.

3.1.3.10 The system shall be able to connect resistors to wires.

## Arduino IDE

### Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

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

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

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

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

3.2.3.5 The system shall open a new Arduino script when selected

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

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

* + - 1. The system shall produce a configuration file that contains the wiring setup and the Arduino code when the “save configuration” button is pressed.
  1. **Bot Simulation**

## Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

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

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

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

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

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

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

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

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

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

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

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

* 1. Arduino Emulation

### Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

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

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

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

3.4.3.4 The system shall match the behavior defined from compiled Arduino code

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

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

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

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

3.4.3.9 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.

# Other Nonfunctional Requirements

Section 5 details the current known requirements and regulations associated with this product as of the Initial Release, Version 1.0.

## Performance Requirements

## Safety Requirements

There are no safety requirement to document regarding this application as it does not handle, distribute, or save critical information, and is completely software based.

## Security Requirements

There are no safety requirement to document regarding this application as it does not handle, distribute, or save critical information, and is completely software based.

## System Quality Attributes

## Business Rules

# Appendix A: Glossary