



Winter Internship Report

On

Bug Fixes and UI Enhancements

Submitted by

Mishtee Gandhi

K. J. Somaiya College of Engineering, Mumbai

Under the guidance of

Prof. Kannan Moudgalya

Chemical Engineering Department
IIT Bombay

Mentors

Mr. Nagesh Karmali

Ms. Firuza Aibara

January 2024

Acknowledgement

I, **Mishtee Gandhi**, winter intern of **FOSSEE – Arduino on Cloud, 2023-24** am overwhelmed in all humbleness and gratefulness to acknowledge my deep gratitude to all those who have helped me put our ideas to perfection and have assigned tasks well above the level of simplicity and into something concrete and unique.

I wholeheartedly thank **Prof. Kannan M. Moudgalya** for having faith in me, selecting me to be a part of his valuable project and for constantly motivating me to do better. I thank **Mr. Nagesh Karmali** and **Ms. Firuza Aibara** for providing me the opportunity to work on this project. I am also very thankful to our mentors for their valuable suggestions. They were and are always there to show me the right track when needed help. With help of their brilliant guidance and encouragement, I was able to complete my tasks properly and was up to the mark in all the tasks assigned. During the process, I got a chance to see the stronger side of my technical and nontechnical aspects and also strengthen my concepts. Last but not the least, I sincerely thank all our other colleagues working in different projects under **Prof. Kannan M. Moudgalya** for helping me evolve better with their critical advice.

Declaration

I declare that this written submission represents my ideas in my own words and whenever others' ideas or words have been included, I have adequately cited and referenced the original sources. I declare that I have properly and accurately acknowledged all sources used in the production of this thesis.

I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources which have not been properly cited or from whom proper permission has not been taken when needed.

Mishtee Gandhi

Index

1. Introduction	5
1.1. Overview	5
2. Error message on empty code	6
2.1. Introduction	6
2.2. Implementation	6
2.3. Output	6
3. Resize font in editor	8
3.1. Introduction	8
3.2. Implementation	8
3.3. Output	8
4. Save Message	10
4.1. Introduction	10
4.2. Implementation	10
5. References	11

Chapter 1 - Introduction

1. Introduction

Arduino on cloud is a cloud-based simulator, which can be used by students and researchers to test and run the simulation on the web, before actual execution. This system allows the users to drag and drop Arduino components from the left pane onto the working space on the right. The pins of the Arduino board can be connected to various input/ output devices like LED, motor, push-button, etc. using wires. There is also a facility to change the color of wires, LEDs, and such components, to differentiate easily. The users can then proceed to write their code in the code window which is then simulated. There is an option for the users to print or save it in pdf format for documentation purposes. The basic ERC check enables the users to find out errors if any.^[1]

1.1. Project Overview

During this fellowship, I worked on the existing project which was already completed to some extent, for the most part, I did bug fixes, UI improvements and features added as well as improvements on certain features.

Chapter 2 – Error message on empty code

2.1. Introduction

The task deals with a bug fix in the compiler. Initially, the compiler would compile regardless of there being any written code or not. While this scenario is parallel to the real world one, this website is meant primarily for students, therefore, in order to prevent errors that can be easily avoided, an error message upon the entering of an empty code rather than it compiling is a prudent measure.

2.2 Implementation:

The thought process was to check if the compiler returned an empty code when the compile function is called and show an error message if the compiler is empty rather than going to the compilation program.

The changes were made in the `ArduinoFrontend/src/app/Libs/Workspace.ts` file in the `CompileCode` function.

<https://github.com/frg-fossee/eSim-Cloud/pull/534>

2.3. Output:

The code testing is seen in figure 2.1 and 2.2 as seen below:

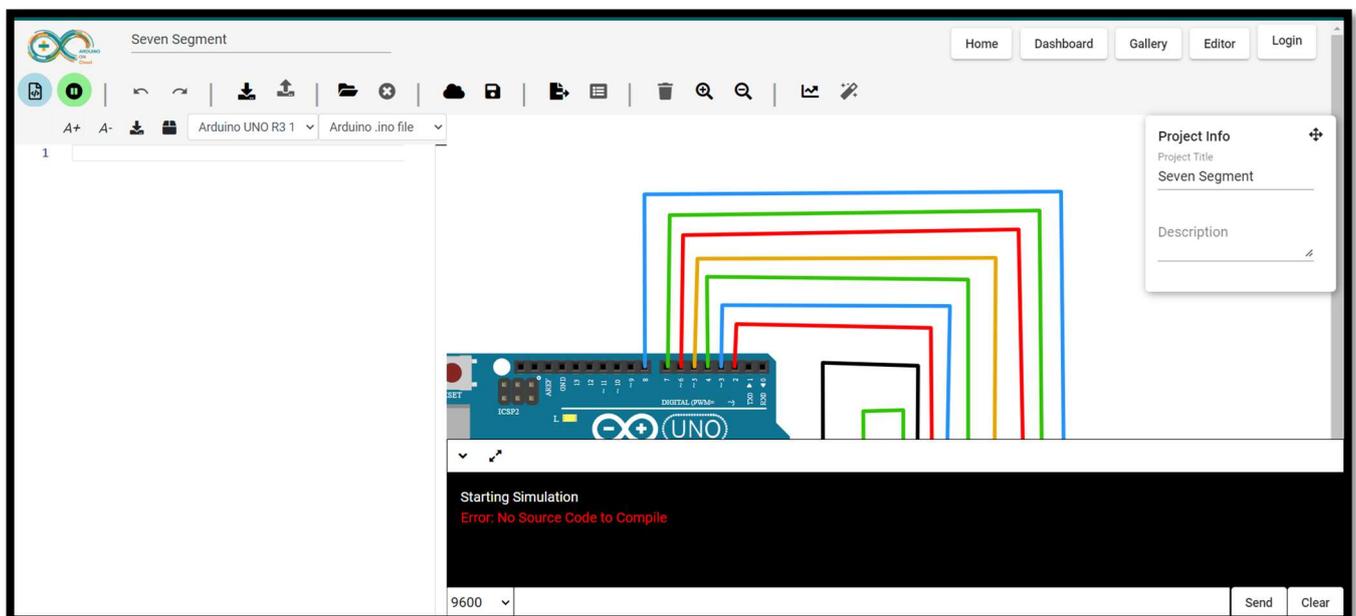


Figure 2.1

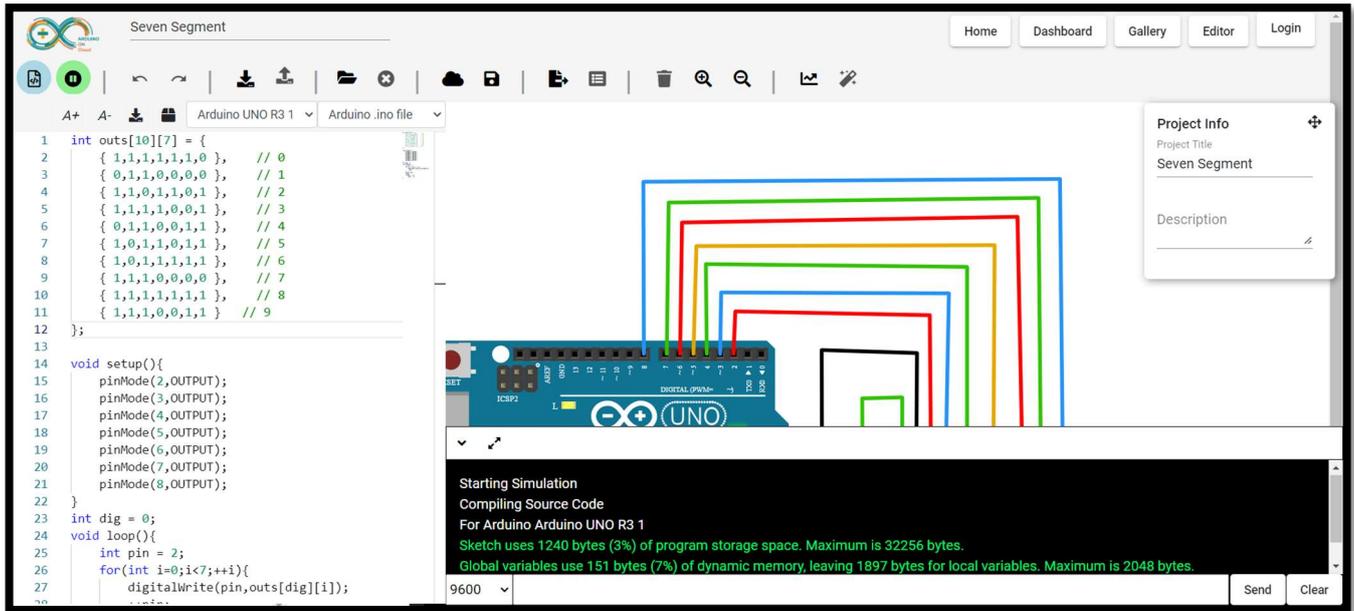


Figure 2.2

Chapter 3 – Resize Font in Editor

3.1. Introduction

The task deals with a UI enhancement in the compiler. Initially, there was no provision for font resizing. This helps the interface to be user friendly.

3.2. Implementation:

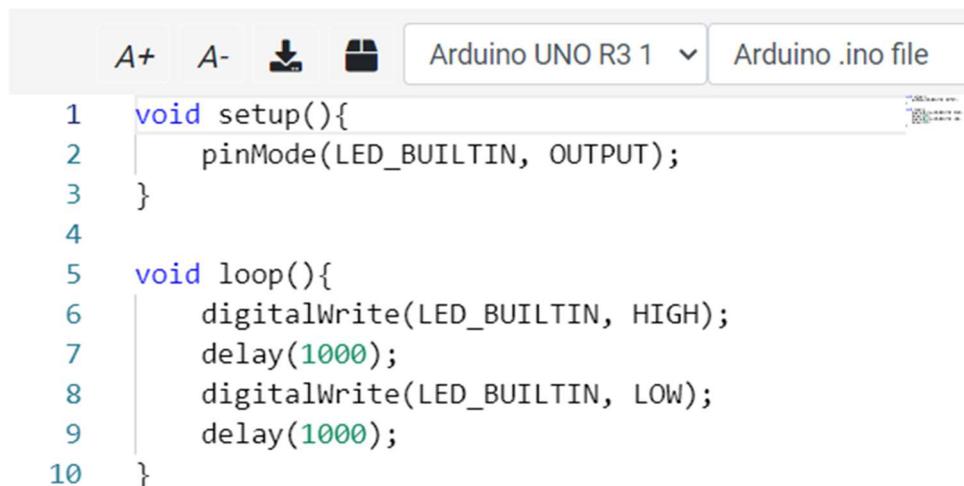
The thought process was to learn more about the editor and the UI. Upon going through the code, I found the file wherein the code editor could be edited.

The changes were made in the `ArduinoFrontend/src/app/code-editor/code-editor.component.ts` file as well as `ArduinoFrontend/src/app/code-editor/code-editor.component.html` in order to add the required options to the UI.

<https://github.com/frg-fossee/eSim-Cloud/pull/535>

3.3. Output:

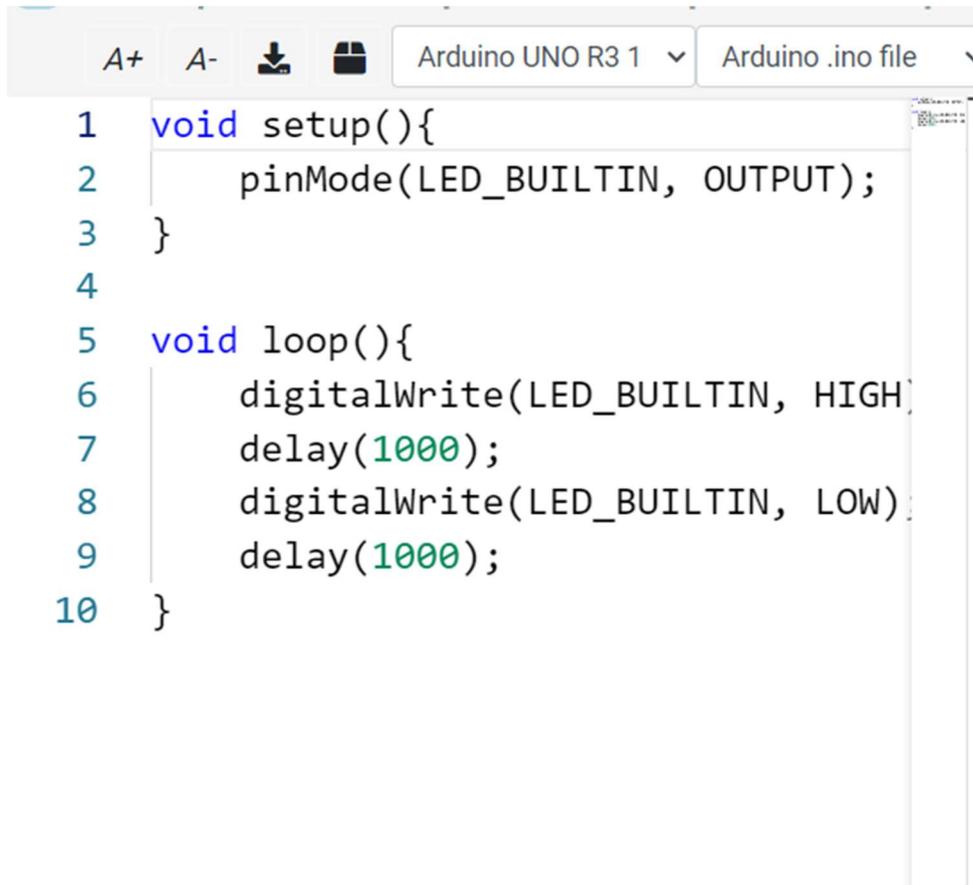
The code testing is seen in figure 3.1, 3.2 and 3.3 as seen below



The screenshot shows the Arduino IDE interface. At the top, there are icons for font size adjustment (A+ and A-), a download icon, and a file icon. To the right, there are dropdown menus for the board type, set to 'Arduino UNO R3 1', and the file type, set to 'Arduino .ino file'. The main area displays the following code:

```
1 void setup(){
2     pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop(){
6     digitalWrite(LED_BUILTIN, HIGH);
7     delay(1000);
8     digitalWrite(LED_BUILTIN, LOW);
9     delay(1000);
10 }
```

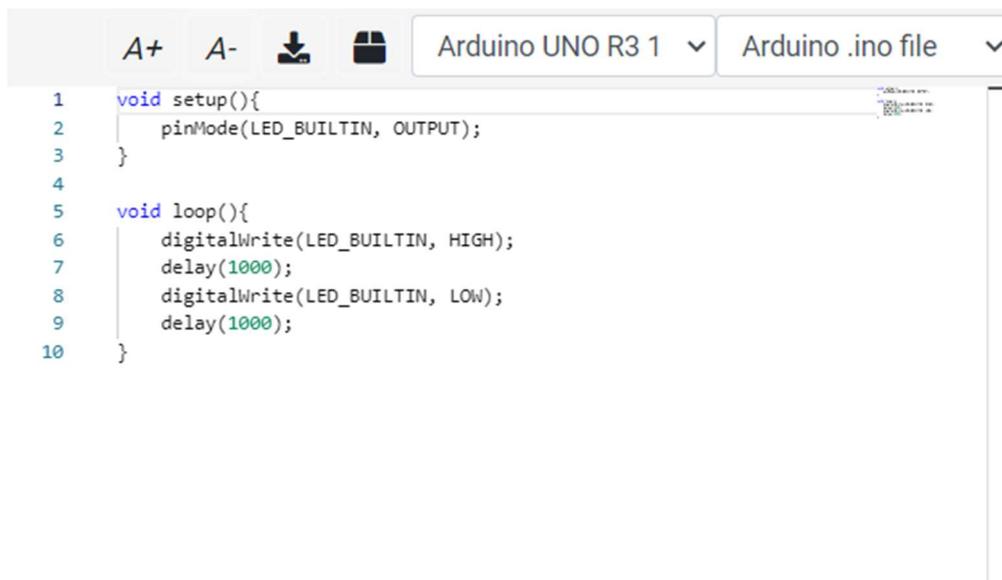
Fig 3.1 – Original font size

A screenshot of an IDE window showing a code editor. The window title is "Arduino UNO R3 1" and the file type is "Arduino .ino file". The code is as follows:

```
1 void setup(){
2     pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop(){
6     digitalWrite(LED_BUILTIN, HIGH);
7     delay(1000);
8     digitalWrite(LED_BUILTIN, LOW);
9     delay(1000);
10 }
```

The font size is noticeably larger than in the previous figure.

Fig 3.2 – Increase in font size

A screenshot of an IDE window showing a code editor. The window title is "Arduino UNO R3 1" and the file type is "Arduino .ino file". The code is as follows:

```
1 void setup(){
2     pinMode(LED_BUILTIN, OUTPUT);
3 }
4
5 void loop(){
6     digitalWrite(LED_BUILTIN, HIGH);
7     delay(1000);
8     digitalWrite(LED_BUILTIN, LOW);
9     delay(1000);
10 }
```

The font size is noticeably smaller than in the previous figure.

Fig 3.3 – Decrease in font size

Chapter 4 – Save Message

4.1. Introduction

The task deals with a UI. Initially, the “Save Changes” pop-up would be triggered regardless of whether changes were made to be saved or if the circuit has already been saved. This is something that wastes time and the pop-up should be seen only when required, i.e. only when changes are made and not saved.

4.2. Implementation:

The thought process was to create version control so that each time a new version is created and saved it is recorded and based on that the pop-up is shown. Changes in the code were made to `ArduinoFrontend/src/app/Libs/Workspace.ts`, `ArduinoFrontend/src/app/Libs/Simulator.ts` and `ArduinoFrontend/src/app/Libs/save-project-dialog-component.html`:

<https://github.com/frg-fossee/eSim-Cloud/pull/536>

References

[1] <https://esim-cloud.readthedocs.io/en/latest/overview/index.html#arduino-on-cloud>