



Summer Fellowship Report

On

Arduino on Cloud - Bug fixing and Further Improvements

Submitted by
Pratham Ingawale
Mumbai University Maharashtra

Under the guidance of
Prof. Kannan Moudgalya Chemical
Engineering Department IIT Bombay

Mentors
Mr. Nagesh Karmali
Ms. Firuza Aibara
March 2023

Acknowledgement

I would like to take this opportunity to express my heartfelt gratitude to all those who have been instrumental in making my summer internship with FOSSEE - Arduino On Cloud a truly enriching and memorable experience. First and foremost, I extend my deepest thanks to Prof. Nagesh Karmali and Prof. Firuza Aibara for believing in my abilities and selecting me to be a part of this invaluable project. Their unwavering support and constant encouragement have been a driving force behind my progress. I owe a debt of gratitude to my mentors, whose guidance and wisdom have been invaluable throughout my internship journey. Their expert advice and timely assistance have not only helped me complete tasks successfully but have also sharpened my technical and non-technical skills. I would also like to extend my appreciation to my colleagues working on various projects under the leadership of Prof. Kannan M. Moudgalya. Their constructive feedback and critical insights have contributed significantly to my growth and development. In conclusion, I am deeply thankful to everyone involved in this endeavor for their support, encouragement, and the opportunity to learn and grow. This experience has been a transformative one, and I am grateful for the chance to have been a part of it.

Pratham Ingawale

Declaration

I declare that this written submission represents our ideas in my own words and whenever other's 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 We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our 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.

Pratham Ingawale

Index

1. Introduction

1.1. Project Overview

2. Feature Additions

2.1. Automatic Scrolling in Console

3. Bug Fixes

3.1. Disabling unwanted toolbar buttons during simulation

3.2. Disabling dragging capability during simulation

3.3. Removed Instances of Breadboard Persist After Deletion Attempt

3.4. Disabling Addition of New Components During Simulation

3.5. Disabling Drawing New Wires During Simulation

3.6. Bug in Breadboard (simulation goes into infinite loop)

3.7. Resolving RTL Pass Combine Error

3.8. Fixing Resistor Value Reset to Default Upon Data Import, Ignoring Previous Changes

4. Conclusion

5. Future Work

6. References

List Of Figures

Figure 2.1: Before Automatic scrolling

Figure 2.2: After automatic scrolling feature

Figure 3.1: Disabled Editing Buttons

Figure 3.2: Drag is restricted

Figure 3.3: REMAINING NODES

Figure 3.4: Adding new components while simulation

Figure 3.5: Adding new wire while simulation

Figure 3.6: APP is not Responding

Figure 3.7: Stack size exceeded

Figure 3.8: RTL pass combine error

Figure 3.9: Error in Arduino Console

Figure 3.10: Resistor Default Value

Chapter 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, My main job was to find and fix any bugs or errors in the project and to add the features that were needed.

Chapter 2

Feature Additions

During the internship, I added a few features to the project, most of these features aimed at improving the accessibility of the website and facilitate some stuff in the website.

2.1 Automatic Scrolling in Console

In this section, I will discuss the implementation of a code snippet that manages automatic scrolling and output display within a web application. The code focuses on enhancing user experience by ensuring that new content remains visible, particularly when dealing with continuous data stream

1.Scroll Control Setup:

The code initializes a boolean variable `shouldScrollToBottom` and retrieves an HTML element with the ID "msg", which is intended to be a container for displaying messages.

2.Scroll Listener:

A scroll event listener is attached to the msg container. When the user scrolls within this container, the listener determines whether the container is scrolled to the bottom. This is achieved by comparing the sum of the current scroll position and visible height with the total scrollable height. If the container is almost at the bottom (within a margin of 1), the `shouldScrollToBottom` flag is set to true.

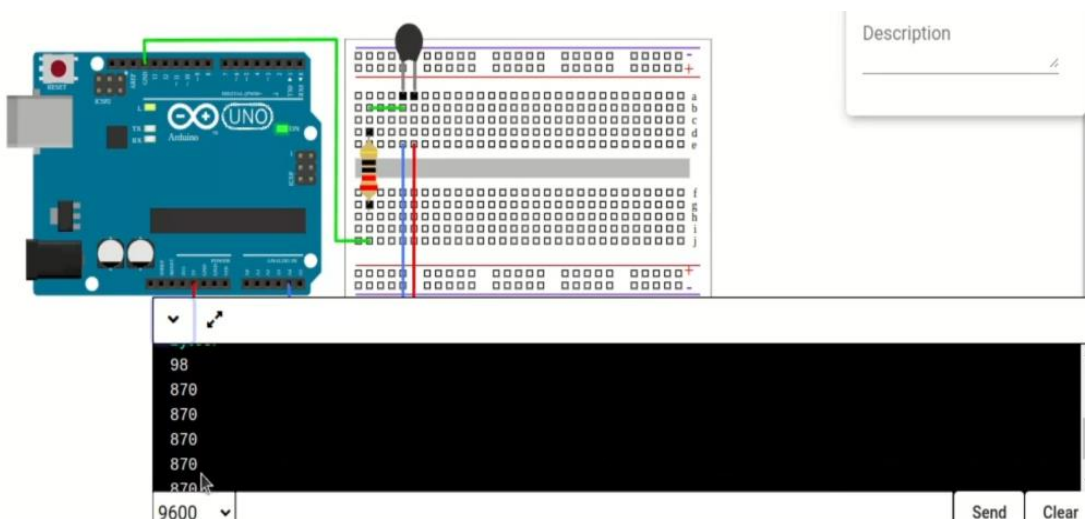


Figure 2.1: Before Automatic scrolling

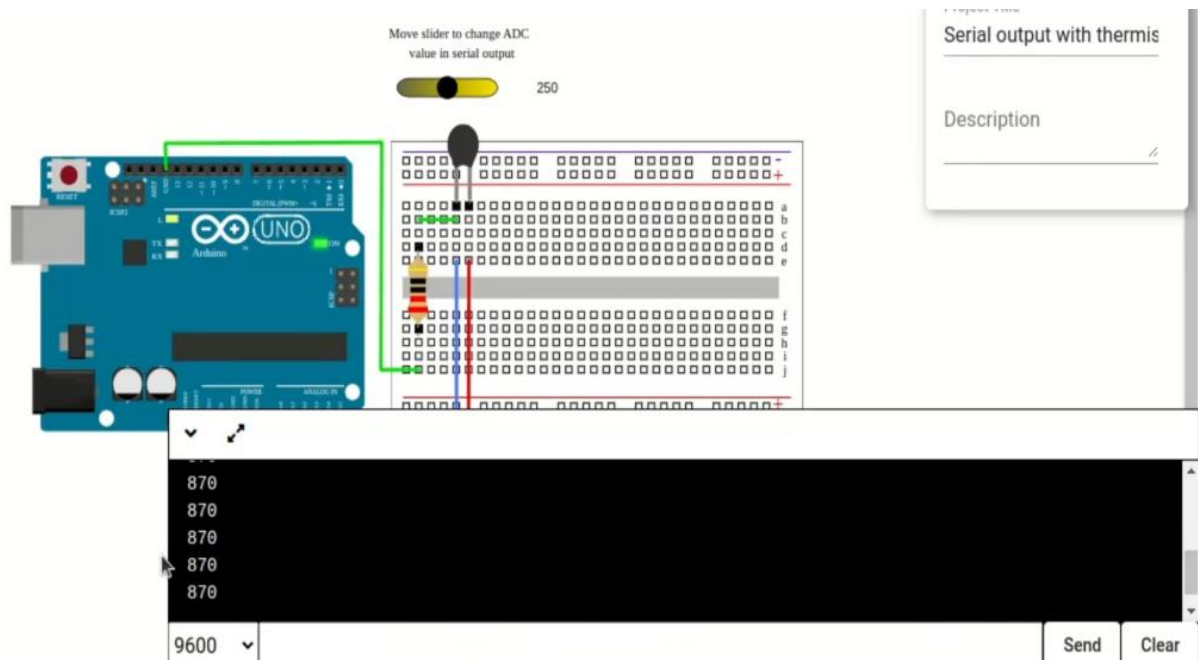


Figure 2.2: After automatic scrolling feature

The advantages of automatic scrolling include:

1. **Seamless Data Stream Viewing:** Automatic scrolling ensures that users can effortlessly keep up with incoming data streams, such as chat messages or real-time updates. This eliminates the need for manual scrolling, allowing users to focus on the content itself.
2. **Continuous Engagement:** By automatically displaying new content as it arrives, the application maintains user engagement and prevents them from missing out on important information. This is particularly valuable for time-sensitive data or live events.
3. **Reduced User Effort:** Users don't have to manually scroll down to view new content, making the experience more convenient and efficient. This is especially beneficial when dealing with lengthy discussions or constant updates.
4. **Real-time Visibility:** Automatic scrolling ensures that the latest content is always visible at the bottom of the container. This is advantageous for applications that prioritize showing the most recent information without requiring users to manually scroll down.
5. **Improved User Experience:** By anticipating user needs and automatically adjusting the content display, the application provides a smoother and more intuitive experience. Users don't have to worry about missing content or interrupting their flow to scroll manually.

These advantages collectively enhance user interaction, content consumption, and overall satisfaction within the web application.

Chapter 3

Bug Fixes

Apart from feature additions, component and examples additions provided above, I even fixed some bugs in it that were already there, some of which were critical enough to stop the normal functioning of simulator. Below I have provided description of all of them in a bit detail.

3.1. Disabling unwanted toolbar buttons during simulation

During a simulation, there are buttons that allow editing. If we use these buttons to change or remove parts while the simulation is running, it can cause unexpected errors. This might lead to problems we didn't predict.

OS tested on: Ubuntu(Linux)

Solution Proposed:

Deactivate or disable all the buttons that are typically used for making edits or modifications. By doing this, users will not be able to utilize these buttons to change or manipulate any elements while the simulation is in progress. This action is taken to prevent unintended consequences or errors that could arise from altering components or settings during the simulation. Disabling the editing buttons ensures that the simulation remains stable and accurate, providing reliable results without the risk of interference due to real-time modifications.

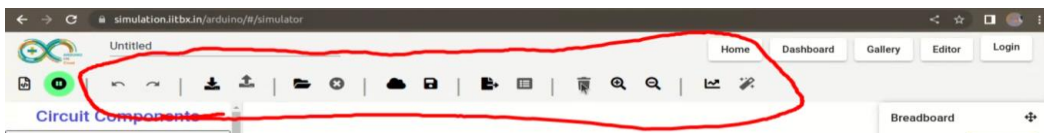


Figure 3.1: Disabled Editing Buttons

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/504>

3.2. Disabling dragging capability during simulation

Due to this bug users move things around during the simulation. This wasn't supposed to happen, as things are usually locked during simulation. The mistake made it possible for users to change things while the simulation was going on. This caused some problems and made the simulation less reliable. We need to fix this so the simulation works correctly.

Tested on: Ubuntu(Linux)

Solution Proposed:

To fix this issue, I've added a feature that stops people from moving parts around once the simulation begins. This helps avoid problems that could happen if things are changed while the simulation is going on. Now, during the simulation, everything stays in its place, making sure the results are correct and dependable.

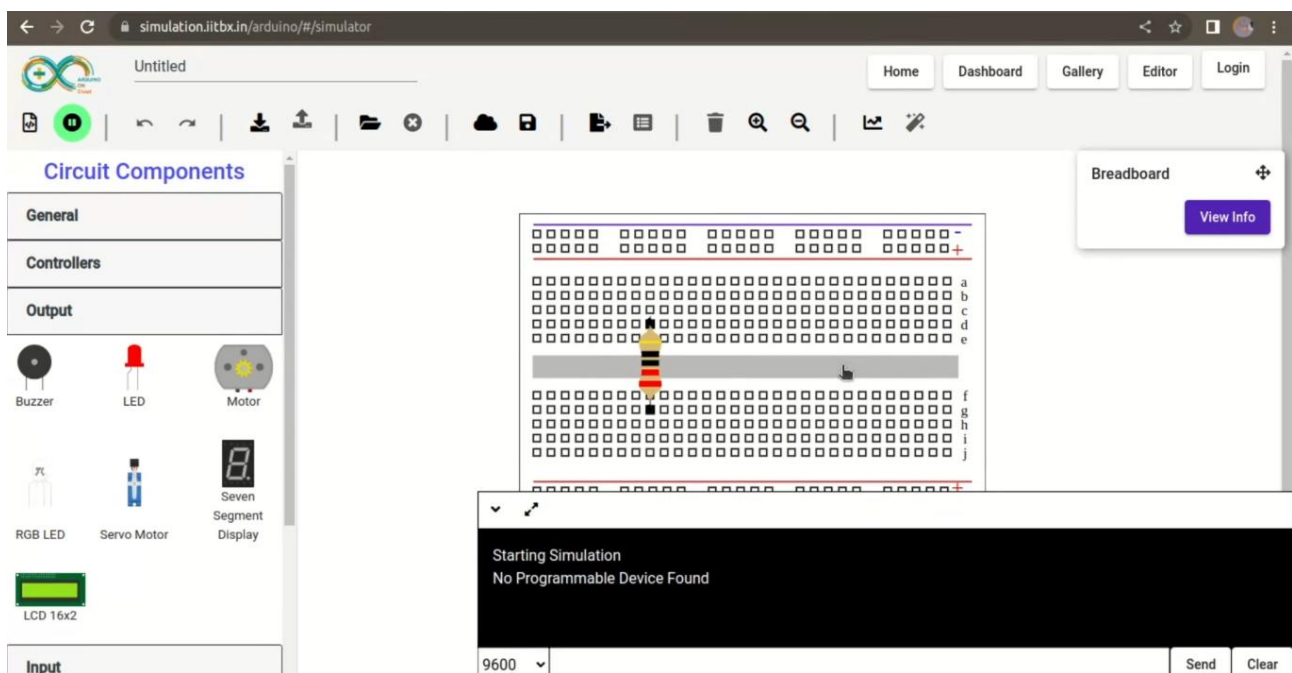


Figure 3.2: Drag is restricted.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/50>

3.3. Removed Instances of Breadboard Persist After Deletion Attempt

Description: BreadBoard has some issue in it, After Deleting the breadboard some nodes of breadboard remain as it is on the workspace.

How to reproduce the error:

- Delete the BREADBOARD after attaching any component on it.

Issue:

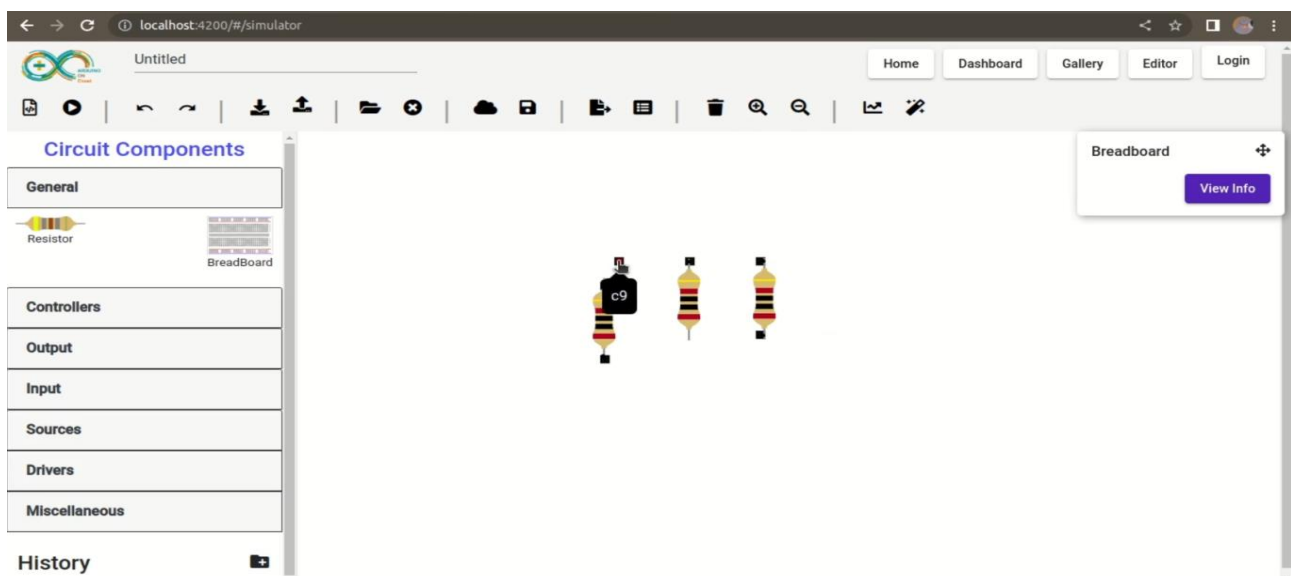


Figure 3.3: REMAINING NODES

Tested on: Ubuntu(Linux)

Solution Proposed:

To address the issue at hand, we recommend implementing the following approach:

Code:-

```
const selectedBreadboard = window['Selected'];
selectedBreadboard.maybeUnsolderElement(selectedBreadboard);
```

This proposed solution involves integrating a ts code snippet into the simulation system. The provided code snippet aims to interact with the selected breadboard element. Specifically, it utilizes the `maybeUnsolderElement` method associated with the `selectedBreadboard` object. This method is intended to manage the process of unsoldering or disconnecting elements on the breadboard during the simulation.

By incorporating this solution into the existing system, it is anticipated that the identified issue will be effectively mitigated. This will ensure the desired control over component behavior within the simulation environment.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/520>

3.4. Disabling Addition of New Components During Simulation

The current problem is that new components can be added during a simulation, which is causing unexpected behavior. Currently, users have the ability to introduce new components while the simulation is in progress. This feature, however, is leading to disruptions in the simulation's intended flow and outcomes. This unexpected functionality goes against the standard simulation practice where components should remain constant once the simulation starts.

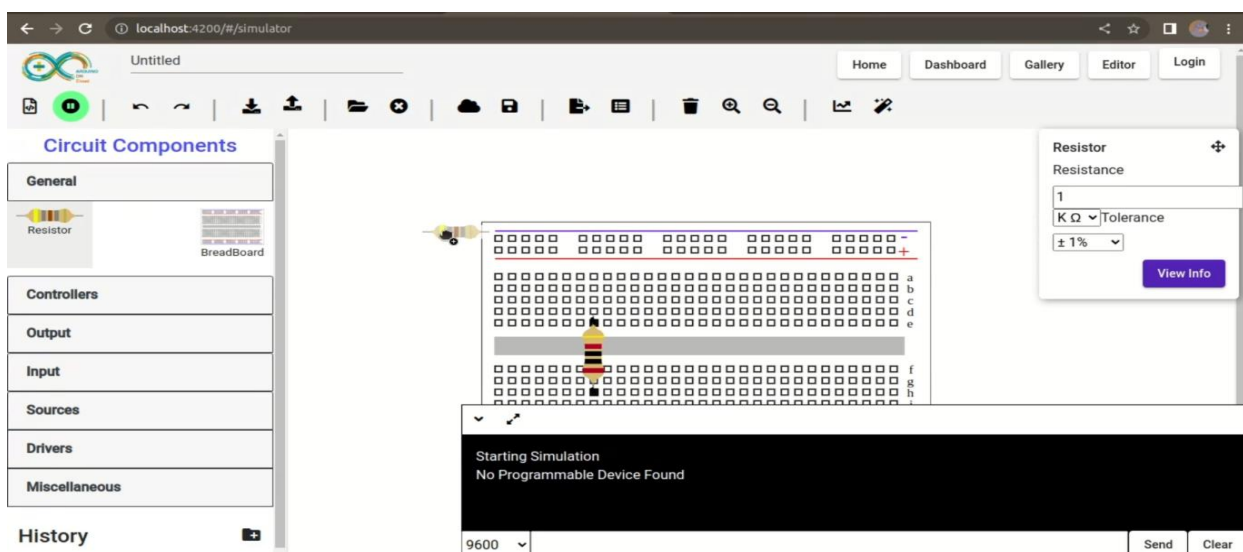


Figure 3.4: Adding new components while simulation

Tested on: Live website & develop branch

OS: Ubuntu

SOLUTION PROPOSED:

I have introduced a functionality that stops users from adding new components once the simulation has started. This ensures that once the simulation is in progress, no new elements can be added. This change is aimed at maintaining the consistency and integrity of the simulation, as any additions after the start could disrupt the accuracy of the results. By preventing the addition of new components during the simulation, we ensure that the simulation remains focused on the initial conditions, leading to more reliable and meaningful outcomes.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/506>

3.5. Disabling Drawing New Wires During Simulation

The problem we're facing is that during the simulation, people can draw new wires, which is causing unexpected things to happen. Right now, while the simulation is running, users can create new wire connections between things. However, this can mess up how the simulation is supposed to work.

This is a problem because these new wires might make things happen that weren't supposed to happen. The simulation was designed with the idea that once it starts, no new wires should be added. But right now, people can add wires and that's causing issues with the results we get from the simulation.

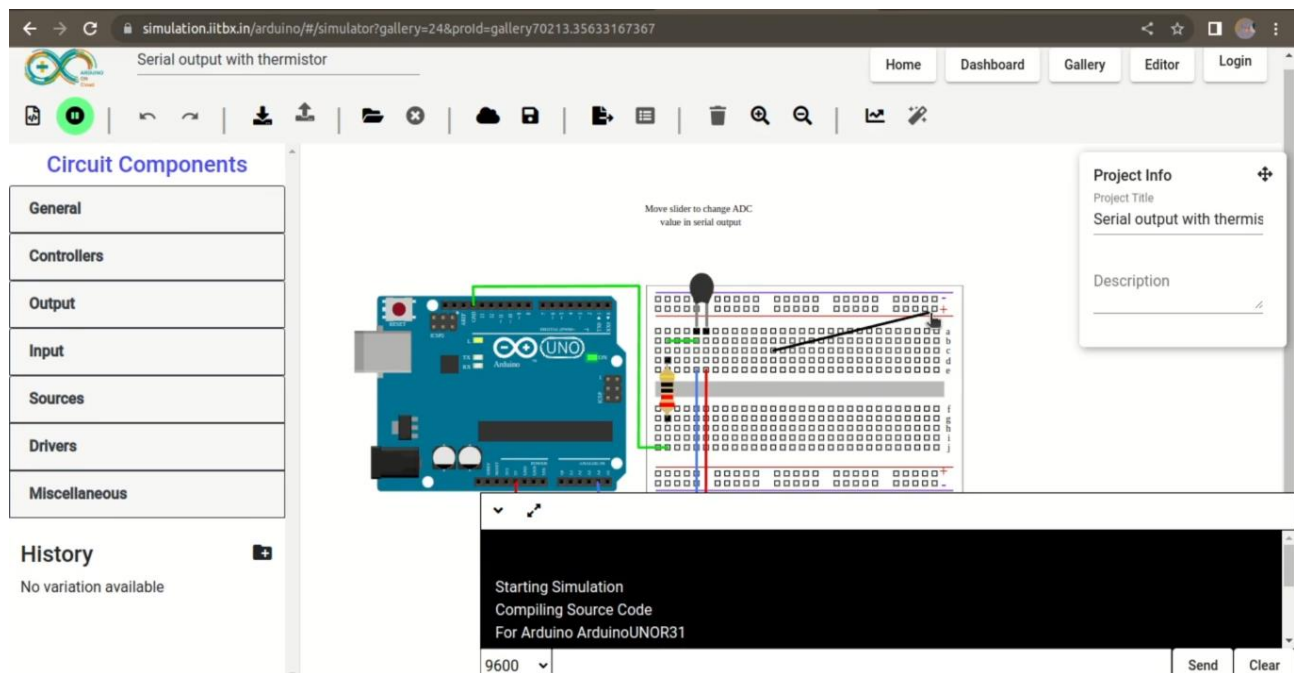


Figure 3.5: Adding new wire while simulation

Tested on: Live website & develop branch

OS: Ubuntu

SOLUTION PROPOSED:

I have successfully addressed the issue by implementing a solution that disables the functionality of drawing new wires during the simulation. This adjustment ensures that users are no longer able to create additional wire connections once the simulation is in progress. By implementing this solution, we have resolved the problem of unexpected interactions and disruptions that were previously caused by the introduction of new wires during simulation. This enhancement contributes to maintaining the accuracy and reliability of the simulation results, aligning with the intended behavior of the simulation system.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/515>

3.6. Bug in Breadboard (simulation goes into infinite loop)

This scenario involves recursion, a programming concept where a function calls itself, leading to an infinite loop within the context of the breadboard simulation. In this situation, a function within the simulation keeps calling itself repeatedly without an exit condition, causing the simulation to get stuck in an endless loop. This can lead to a significant issue as the simulation becomes unresponsive and consumes excessive resources, potentially causing the application to crash or hang.

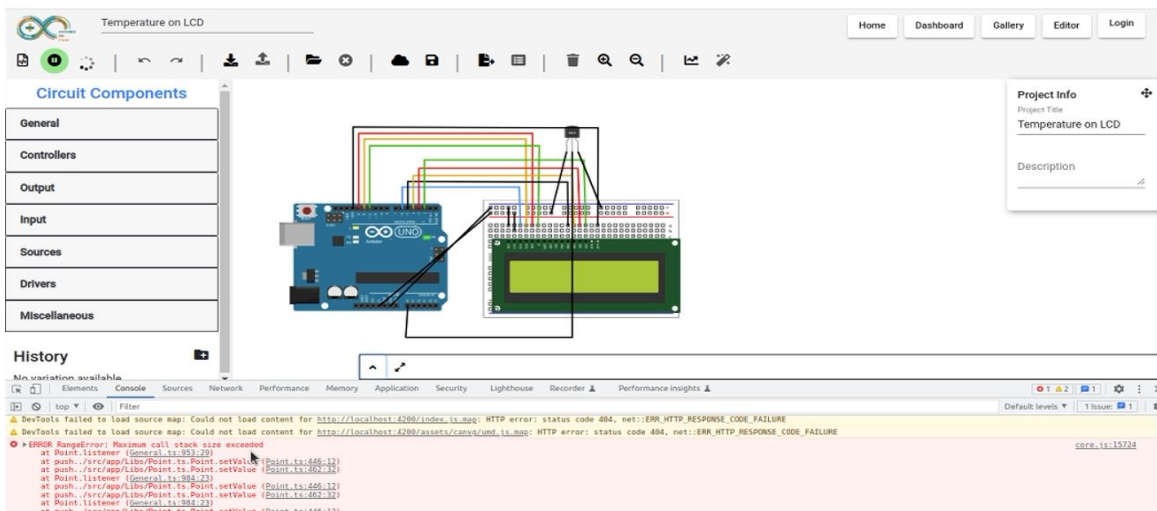


Figure 3.6: APP is not Responding

Error in console:

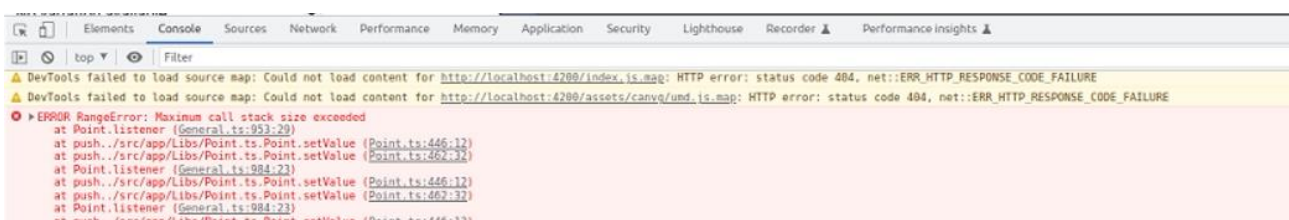


Figure 3.7: Stack size exceeded

Tested on: Live Website & Develop branch

OS Used: Ubuntu(Linux)

TEMPORARY SOLUTION :

I made an adjustment in the JSON file of the breadboard simulation by modifying the X coordinate value, increasing it by 0.1. This was done to prevent the occurrence of an infinite loop. By making this change, I ensured that the condition necessary for terminating the loop is always satisfied. This alteration should help maintain the simulation's functionality and prevent it from getting stuck in an endless loop, contributing to a smoother and more reliable simulation experience.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/527>

3.7. Resolving RTL Pass Combine Error

When this error occurs, it typically indicates a problem during the process of optimizing or combining certain operations in the code. The compiler or tool encounters difficulties while trying to simplify or optimize specific code segments for better performance. This particular error often occurs when using the "print" function within the Arduino Integrated Development Environment (IDE).

In simpler terms, it's like the software is having trouble rearranging or streamlining parts of code. This issue can arise due to complex or unusual operations within the code.

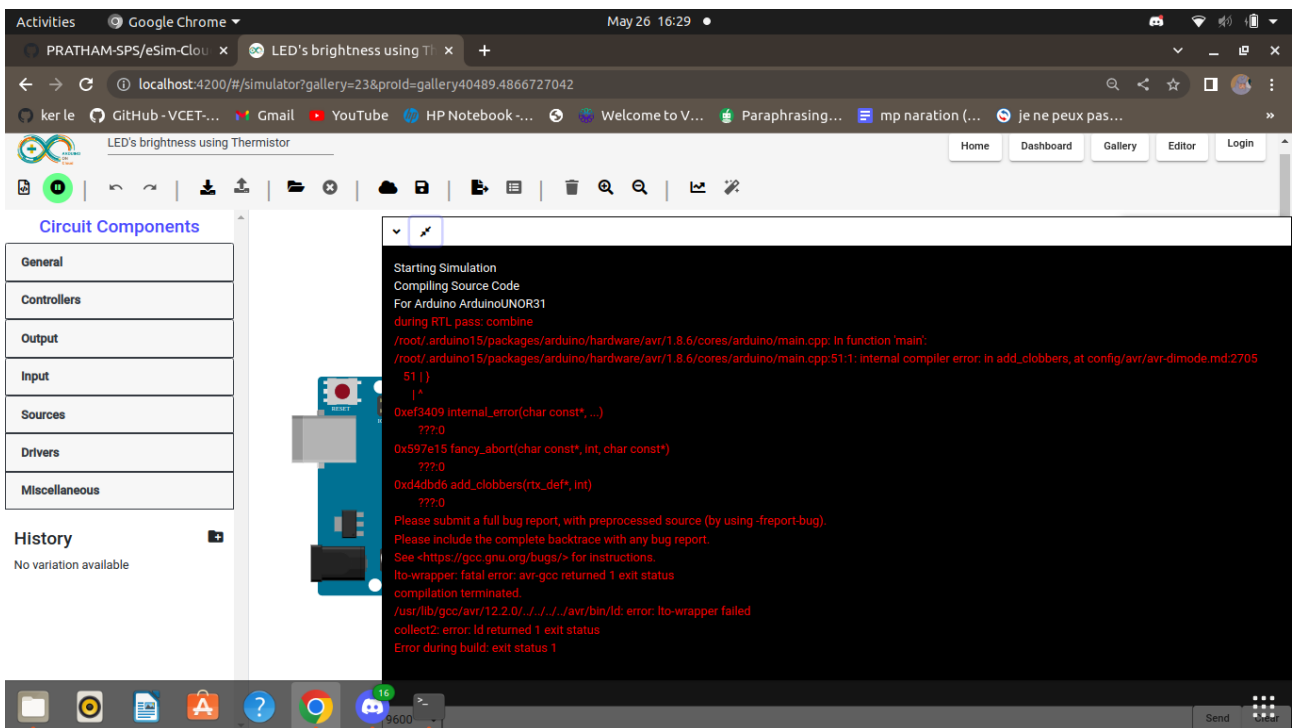


Figure 3.8: RTL pass combine error

Error in Arduino Console :

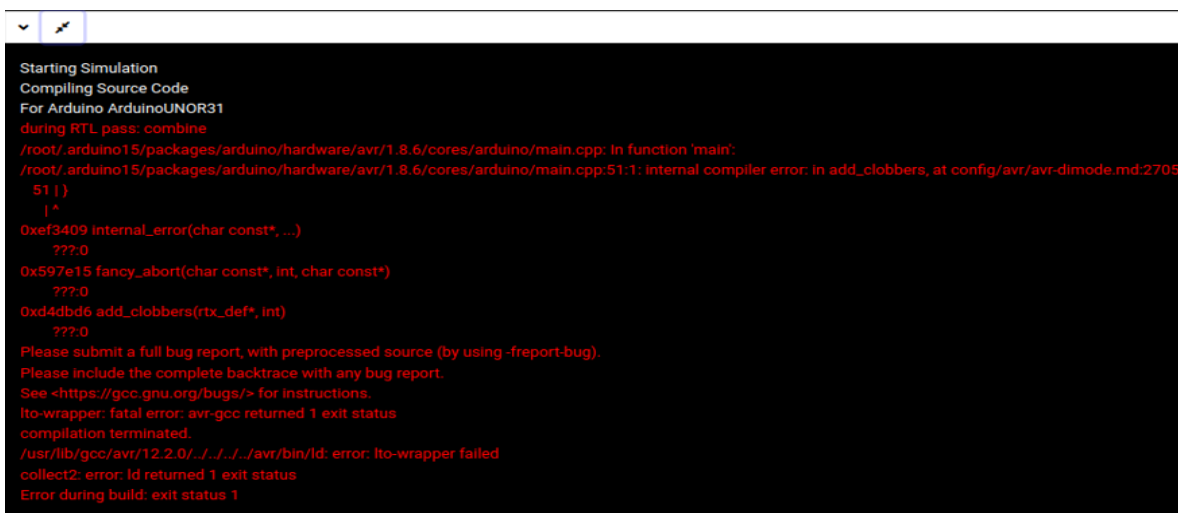


Figure 3.9: Error in Arduino Console

Tested on: Live Website & Develop branch
OS Used: Ubuntu(Linux)

SOLUTION PROPOSED:

Upon investigation of the Docker containers, I found an issue with the compatibility between the GNU Compiler Collection (GCC) version and the current Arduino Command Line Interface (CLI) version. To tackle this, I made an adjustment by upgrading the Alpine version to 3.13 in the Docker containers. This newer Alpine version included a GCC version that was in line with the needs of the current CLI version.

As a result of this change, the previously mismatched GCC version was replaced with a suitable one. This ensured that the Arduino CLI could execute its compilation processes smoothly without encountering compatibility hiccups. By ensuring that the versions aligned, I successfully resolved the challenge that was hindering the proper functioning of the Arduino CLI within the Docker environment. This demonstrates the significance of keeping software components in sync for a seamless workflow.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/522>

3.8. Fixing Resistor Value Reset to Default Upon Data Import, Ignoring Previous Changes

The problem we're encountering relates to a repetitive behavior where, upon importing a JSON file, the software consistently reverts resistor values to their original settings(The default value is set to 220). This happens regardless of any previous adjustments made to these resistor values.

Think of it like adjusting the temperature on your air conditioner. Every time you turn it on, the temperature goes back to the default setting, even if you had set it differently before. Similarly, in this case, we're changing resistor values, but when we import a JSON file, the software resets them to their initial state.

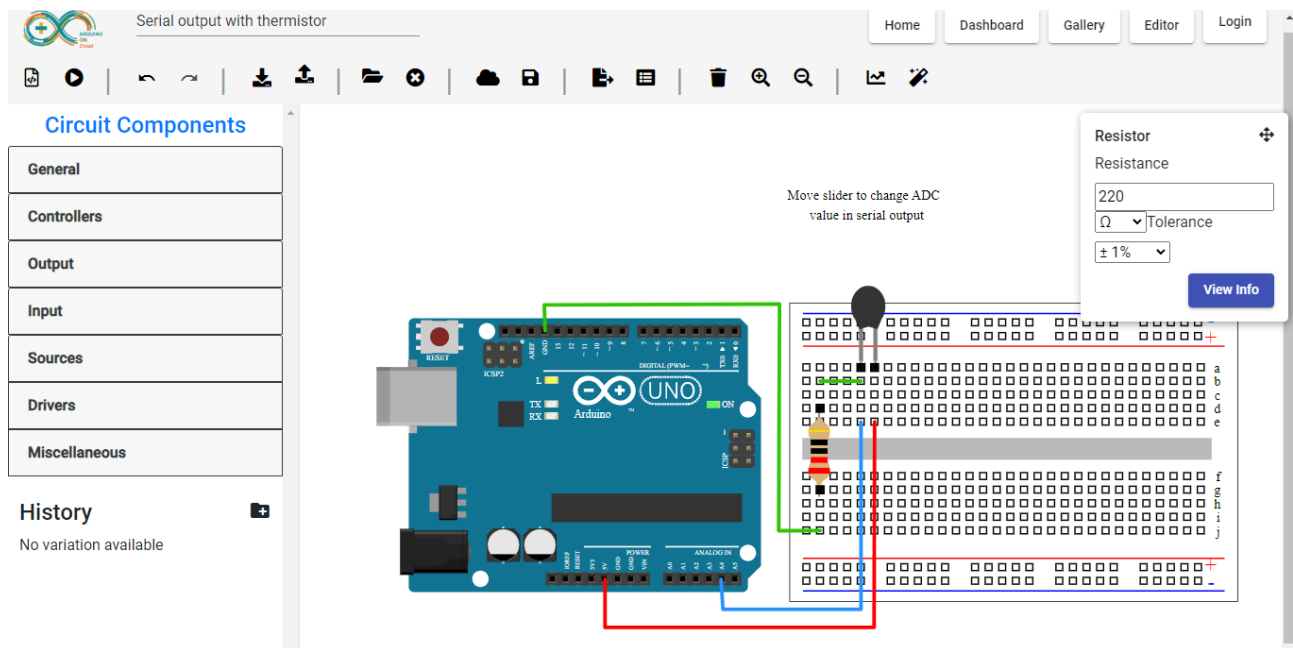


Figure 3.10: Resistor Default Value

Tested on: Live Website & Develop branch
OS Used: Ubuntu(Linux)

SOLUTION PROPOSED:

I've added a special signal (flag) to monitor whether a file has been imported. If a file is imported, the resistor value will be fetched from the JSON file. However, if the resistor is being initialized for the first time, it will be set to the default value of 220. This approach ensures that the correct value is used based on whether a file has been imported or not.

PR: <https://github.com/frg-fossee/eSim-Cloud/pull/524>

Chapter 4

Conclusion

This project was truly special, like a work of art that I thoroughly enjoyed creating. Working on it and adding different features was a lot of fun, and I got to learn so many new things along the way. I'm really thankful to my mentors for their amazing guidance throughout the project.

I believe this project has so much potential. It could be a great help to students and researchers who are exploring Arduino. All the effort and creativity that went into this project make it a valuable tool for learning and researching. I'm excited about the impact this project could have and how it might make a difference for others.

Chapter 5

Future Work

This project can be improved further by adding new features and fixing already present bugs.

- Adding more sensors and components.
- Fixing bug in motor, due to which motor's rotor don't rotate in if PWM is attached.
- Adding support for more programming languages.
- Optimisation of application.
- Adding collaboration support in it, example GitHub.
- Resizing the Code Window and Adjusting Font Size for Code Display

References

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

[2] <https://stackblitz.com/>

[3] <https://www.npmjs.com/>

[4] <https://arduino.github.io/arduino-cli/0.33/>

[5] <https://docs.docker.com/>

[6] <https://stackoverflow.com/>